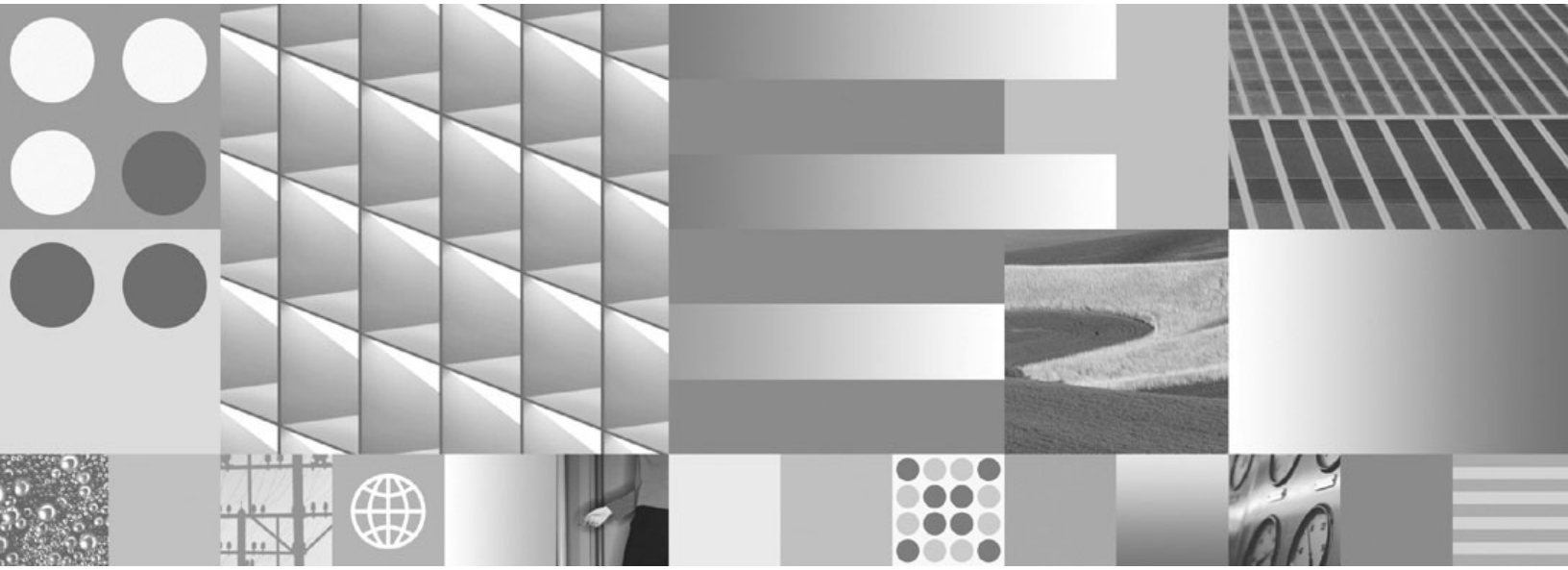
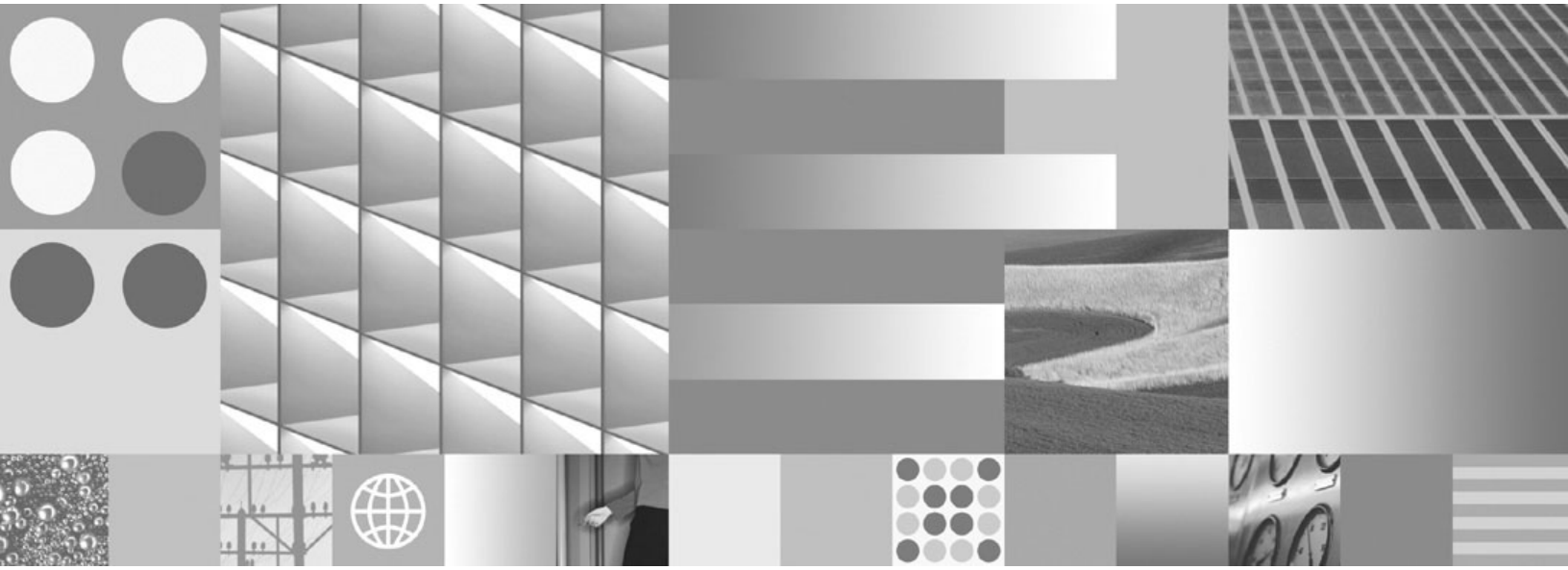
DB2 Version 9.5
für Linux, UNIX und Windows



Datenbanksicherheit

SC12-3914-01

DB2 Version 9.5
für Linux, UNIX und Windows



Datenbanksicherheit

SC12-3914-01

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter Anhang B, „Bemerkungen", auf Seite 295 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Version 9.5 for Linux, UNIX, and Windows Database Security Guide,
IBM Form SC23-5850-01,
herausgegeben von International Business Machines Corporation, USA

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
März 2008

Inhaltsverzeichnis

Zu diesem Handbuch

Das Handbuch 'Datenbanksicherheit' beschreibt, wie DB2-Sicherheitsfunktionen zur Implementierung und Verwaltung der Stufe von Sicherheit eingesetzt werden, die für Ihre Datenbankinstallation erforderlich ist.

Das Handbuch 'Datenbanksicherheit' stellt detaillierte Informationen zu folgenden Themenbereichen bereit:

- Verwalten der Authentifizierung von Benutzern, die auf DB2-Datenbank zugreifen können
- Einrichten der Berechtigungsfunktion zur Steuerung des Benutzerzugriffs auf Objekte und Daten in Datenbanken

Kapitel 1. DB2-Sicherheitsmodell

Zwei Sicherheitsmodi steuern den Zugriff auf die Daten und Funktionen des DB2-Datenbanksystems. Der Zugriff auf das DB2-Datenbanksystem wird durch Einrichtungen verwaltet, die sich außerhalb des DB2-Datenbanksystems (Authentifizierung) befinden, während der Zugriff innerhalb des DB2-Datenbanksystems durch den Datenbankmanager (Berechtigungen) verwaltet wird.

Authentifizierung

Die Authentifizierung ist ein Prozess, bei dem ein System die Identität eines Benutzers prüft. Die Benutzerauthentifizierung wird durch eine Sicherheitseinrichtung außerhalb des DB2-Datenbanksystems mithilfe eines Sicherheits-Plug-in-Moduls zur Authentifizierung ausgeführt. Ein Standard-Sicherheits-Plug-in-Modul zur Authentifizierung, das auf die betriebssystembasierte Authentifizierung zurückgreift, wird bereitgestellt, wenn Sie das DB2-Datenbanksystem installieren. Zur Erhöhung der Flexibilität bei der Umsetzung Ihrer speziellen Authentifizierungsanforderungen können Sie ein eigenes Sicherheits-Plug-in-Modul zur Authentifizierung erstellen.

Der Authentifizierungsprozess generiert eine DB2-Berechtigungs-ID. Informationen zur Gruppenzugehörigkeit des Benutzers werden ebenfalls bei der Authentifizierung abgerufen. Das Standardverfahren zum Abrufen von Gruppeninformationen greift auf ein Plug-in-Modul zum Abrufen von Gruppenzugehörigkeiten auf Betriebssystemebene zurück, das zur Verfügung gestellt wird, wenn Sie das DB2-Datenbanksystem installieren. Wenn Sie dies bevorzugen, können Sie Informationen zur Gruppenzugehörigkeit auch abrufen, indem Sie ein spezielles Plug-in-Modul für Gruppenzugehörigkeiten verwenden, wie zum Beispiel ein LDAP-Modul (Lightweight Directory Access Protocol).

Berechtigung

Wenn ein Benutzer authentifiziert wurde, stellt der Datenbankmanager fest, ob diesem Benutzer Zugriff auf Daten oder Ressourcen von DB2 gewährt wird. Die Berechtigung ist ein Prozess, bei dem der DB2-Datenbankmanager Informationen zum authentifizierten Benutzer abruft, die angeben, welche Datenbankoperationen der Benutzer ausführen und auf welche Datenobjekte er zugreifen darf.

Die folgenden unterschiedlichen Quellen von Berechtigungen sind für eine Berechtigungs-ID verfügbar:

1. Primäre Berechtigungen: die Berechtigungen, die der Berechtigungs-ID direkt erteilt werden.
2. Sekundäre Berechtigungen: die Berechtigungen, die den Gruppen und Rollen erteilt sind, zu denen die Berechtigungs-ID gehört.
3. Allgemein zugängliche Berechtigungen: die Berechtigungen, die der speziellen Gruppe PUBLIC erteilt werden.
4. Kontextabhängige Berechtigungen: die Berechtigungen, die einer Rolle in einem gesicherten Kontext erteilt werden.

Berechtigungen können Benutzern in den folgenden Kategorien erteilt werden:

- Berechtigungen der Systemebene

Die Berechtigungen Systemadministrator (SYSADM), Systemsteuerung (SYSCTRL), Systempflege (SYSMAINT) und Systemmonitor (SYSMON) stellen verschiedene Grade an Steuerungsmöglichkeiten für Funktionen auf Instanzebene zur Verfügung. Berechtigungen bieten die Möglichkeit, Zugriffsrechte zu gruppieren und Verwaltungs- und Dienstprogrammoperationen für Instanzen, Datenbanken und Datenbankobjekte zu steuern.

- Berechtigungen der Datenbankebene

  Die Berechtigungen Sicherheitsadministrator (SECADM) und Datenbankadministrator (DBADM) erteilen Steuerungsmöglichkeiten innerhalb der Datenbank. Weitere Datenbankberechtigungen sind LOAD (die Möglichkeit zum Laden von Daten in eine Tabelle) und CONNECT (die Möglichkeit, eine Verbindung zu einer Datenbank herzustellen).
- Berechtigungen der Objektebene

  Zu den Berechtigungen der Objektebene gehört das Überprüfen von Zugriffsrechten, wenn eine Operation an einem Objekt ausgeführt wird. Zum Beispiel muss ein Benutzer (mindestens) das Zugriffsrecht SELECT für eine Tabelle besitzen, um Daten aus dieser Tabelle auswählen zu können.
- Kontextabhängige Berechtigungen

  Mithilfe von Sichten lässt sich steuern, welche Spalten oder Zeilen einer Tabelle bestimmte Benutzer lesen können. Die kennsatzbasierte Zugriffssteuerung (LBAC, Label-Based Access Control) bestimmt, welche Benutzer Lese- und Schreibzugriff auf einzelne Zeilen und einzelne Spalten haben.

Sie können diese Funktionen in Verbindung mit der DB2-Prüffunktion zur Zugriffsüberwachung verwenden, um die Sicherheitsstufe zu definieren und zu verwalten, die für Ihre Datenbankinstallation erforderlich ist.

Authentifizierung

Die Authentifizierung eines Benutzers wird mithilfe einer Sicherheitseinrichtung außerhalb des DB2-Datenbanksystems ausgeführt. Die Sicherheitseinrichtung kann Teil des Betriebssystems oder ein separates Produkt sein.

Die Sicherheitseinrichtung benötigt zwei Elemente zur Authentifizierung eines Benutzers: eine Benutzer-ID und ein Kennwort. Die Benutzer-ID identifiziert den Benutzer der Sicherheitseinrichtung gegenüber. Durch die Eingabe des richtigen Kennworts (eine Information, die nur dem Benutzer und der Sicherheitseinrichtung bekannt ist) wird die Identität des Benutzers (entsprechend der Benutzer-ID) bestätigt.

Anmerkung: In Nichtrootinstallationen muss die betriebssystembasierte Authentifizierung durch Ausführen des Befehls db2rfe aktiviert werden.

Nach erfolgreicher Authentifizierung:

- Der Benutzer muss für DB2 mithilfe eines SQL-Berechtigungsnamens oder einer Berechtigungs-ID (*authid*) identifiziert werden. Der Name kann mit der Benutzer-ID übereinstimmen oder ein zugeordneter Wert sein. Beispielsweise wird unter UNIX-Betriebssystemen bei Verwendung des Standard-Sicherheits-Plug-in-Moduls eine DB2-Berechtigungs-ID (*authid*) abgeleitet, indem eine UNIX-Benutzer-ID, die den DB2-Namenskonventionen entspricht, in Großbuchstaben umgesetzt wird.
- Eine Liste von Gruppen, zu denen der Benutzer gehört, wird abgerufen. Die Gruppenzugehörigkeit kann bei der Berechtigung des Benutzers verwendet werden. Gruppen sind Entitäten von Sicherheitseinrichtungen, die auch DB2-

Berechtigungsnamen zugeordnet sein müssen. Diese Zuordnung erfolgt mit einer ähnlichen Methode wie für Benutzer-IDs.

Der DB2-Datenbankmanager verwendet die Sicherheitseinrichtung zur Authentifizierung von Benutzern auf eine von zwei Arten:

- Eine erfolgreiche Anmeldung am Sicherheitssystem dient als Nachweis der Identität und lässt Folgendes zu:
  - Verwendung lokaler Befehle zum Zugriff auf lokale Daten
  - Verwendung von Remoteverbindungen, wenn der Server die Clientauthentifizierung akzeptiert
- Eine erfolgreiche Prüfung einer Benutzer-ID mit Kennwort durch die Sicherheitseinrichtung dient als Nachweis der Identität und lässt Folgendes zu:
  - Verwendung von Remoteverbindungen, bei denen der Server einen Nachweis der Authentifizierung verlangt
  - Verwendung von Operationen, bei denen der Benutzer einen Befehl unter einer anderen als der zur Anmeldung verwendeten Identität ausführen will

Anmerkung: Auf einigen UNIX-Systemen kann der DB2-Datenbankmanager fehlgeschlagene Versuche der Kennworteingabe über das Betriebssystem protokollieren und erkennen, wann ein Client die Anzahl zulässiger Anmeldeversuche überschritten hat, die mit dem Parameter LOGINRETRIES festgelegt wird.

Berechtigung

Die Berechtigung wird mithilfe von DB2-Funktionen durchgeführt. DB2-Tabellen und -Konfigurationsdateien werden zum Aufzeichnen der Berechtigungen verwendet, die jedem Berechtigungsnamen zugeordnet sind.

Wenn ein authentifizierter Benutzer versucht, auf Daten zuzugreifen, werden der Berechtigungsname des Benutzers, die Namen der Gruppen, zu denen der Benutzer gehört, und die Namen der Rollen, die dem Benutzer direkt oder indirekt durch eine Gruppe oder Rolle erteilt wurden, mit den gespeicherten Berechtigungen verglichen. Anhand dieses Vergleichs entscheidet der DB2-Server, ob der angeforderte Zugriff gewährt wird.

Die Typen von Berechtigungen, die aufgezeichnet werden, sind Zugriffsrechte, Berechtigungsstufen und LBAC-Berechtigungsnachweise.

Ein *Zugriffsrecht* definiert eine einzelne Berechtigung für einen Berechtigungsnamen, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Zugriffsrechte werden in den Datenbankkatalogen gespeichert.

Berechtigungsstufen bilden eine Methode, Zugriffsrechte und die Kontrolle über Operationen zur Wartung des Datenbankmanagers und über Dienstprogrammoperationen auf einer höheren Stufe in Gruppen zusammenzufassen. Datenbankspezifische Berechtigungen werden in den Datenbankkatalogen gespeichert, während Systemberechtigungen durch Gruppenzugehörigkeit zugeordnet und die den Berechtigungsstufen zugeordneten Gruppennamen in der Konfigurationsdatei des Datenbankmanagers für die jeweilige Instanz gespeichert werden.

LBAC-Berechtigungsnachweise sind LBAC-Sicherheitskennsätze und Freistellungen von LBAC-Regeln, die ein Zugreifen auf Daten zulassen, die durch LBAC (Label-Based Access Control) geschützt sind. LBAC-Berechtigungsnachweise werden in den Datenbankkatalogen gespeichert.

Gruppen stellen eine zweckmäßige Methode dar, für einen Benutzerverbund Berechtigungen durchzuführen, ohne jedem Benutzer einzeln Zugriffsrechte erteilen bzw. entziehen zu müssen. Sofern nicht anders angegeben, können Gruppenberechtigungsnamen überall dort verwendet werden, wo Berechtigungsnamen zu Berechtigungszwecken verwendet werden. Im Allgemeinen wird die Gruppenzugehörigkeit für dynamisches SQL und Berechtigungen für Nichtdatenbankobjekte (z. B. Befehle auf Instanzebene und Dienstprogramme), jedoch nicht für statisches SQL berücksichtigt. Eine Ausnahme von diesem allgemeinen Fall ist das Erteilen von Zugriffsrechten für PUBLIC: Diese werden bei der Verarbeitung von statischem SQL berücksichtigt. Bestimmte Fälle, in denen die Gruppenzugehörigkeit nicht gültig ist, werden in der DB2-Dokumentation entsprechend vermerkt.

Eine Rolle ist ein Datenbankobjekt, das ein oder mehrere Zugriffsrechte zusammenfasst und Benutzern, Gruppen, der Gruppe PUBLIC oder anderen Rollen durch eine Anweisung GRANT oder einem gesicherten Kontext durch eine Anweisung CREATE TRUSTED CONTEXT oder ALTER TRUSTED CONTEXT erteilt werden kann. Eine Rolle kann für das Verbindungsattribut SESSION_USER ROLE in einer Auslastungsdefinition (Workloaddefinition) angegeben werden. Bei der Verwendung von Rollen ordnen Sie Zugriffsberechtigungen für Datenbankobjekte über die Rollen zu. Benutzer, die diese Rollen haben, verfügen über die Zugriffsrechte, die für die Rolle definiert sind und mit denen sie auf Datenbankobjekte zugreifen können.

Rollen stellen eine ähnliche Funktionalität wie Gruppen zur Verfügung. Sie führen die Berechtigung für einen Benutzerverbund aus, ohne dass Zugriffsrechte jedem Benutzer einzeln erteilt oder entzogen werden müssen. Ein Vorteil von Rollen besteht darin, dass sie vom DB2-Datenbanksystem verwaltet werden. Die den Rollen erteilten Berechtigungen werden während des Berechtigungsprozesses für Sichten, Trigger, MQTs (Materialized Query Tables), Pakete und SQL-Routinen geprüft, während die Berechtigungen, die Gruppen erteilt wurden, nicht geprüft werden. Die Berechtigungen, die Gruppen erteilt sind, werden während des Berechtigungsprozesses für Sichten, Trigger, MQTs, Pakete und SQL-Routinen deshalb nicht geprüft, weil das DB2-Datenbanksystem nicht erkennen kann, wenn sich die Zugehörigkeit zu einer Gruppe ändert. Daher kann es die genannten Objekte bei Bedarf nicht ungültig machen.

Anmerkung: Berechtigungen, die Rollen erteilt sind, die wiederum Gruppen erteilt sind, werden während des Berechtigungsprozesses für Sichten, Trigger, MQTs, Pakete und SQL-Routinen nicht geprüft.

Während der Verarbeitung von SQL-Anweisungen werden vom DB2-Berechtigungsmodell alle folgenden Berechtigungen geprüft:

1. Die Berechtigungen, die der primären Berechtigungs-ID der SQL-Anweisung erteilt sind
2. Die Berechtigungen, die den Rollen erteilt sind, die der primären Berechtigungs-ID der SQL-Anweisung erteilt sind
3. Die Berechtigungen, die den sekundären Berechtigungs-IDs (Gruppen oder Rollen) der SQL-Anweisung erteilt sind
4. Die Berechtigungen, die den Rollen erteilt sind, die den sekundären Berechtigungs-IDs (Gruppen oder Rollen) der SQL-Anweisung erteilt sind
5. Die Berechtigungen, die der Gruppe PUBLIC erteilt sind, einschließlich der Rollen, die der Gruppe PUBLIC direkt oder indirekt durch andere Rollen erteilt sind
6. Die Berechtigungen, die der Rolle des gesicherten Kontexts erteilt sind, falls zutreffend.

Sicherheitsaspekte bei Installation und Verwendung des DB2-Datenbankmanagers

Sicherheitsaspekte spielen für den DB2-Administrator von dem Moment der Installation des Produkts an eine wichtige Rolle.

Zur Ausführung der Installation des DB2-Datenbankmanagers sind eine Benutzer-ID, ein Gruppenname und ein Kennwort erforderlich. Das grafisch orientierte Installationsprogramm des DB2-Datenbankmanagers erstellt Standardwerte für verschiedene Benutzer-IDs und die Gruppe. Abhängig davon, ob Sie auf Linux und UNIX-Plattformen oder Windows-Plattformen installieren, werden unterschiedliche Standardwerte erstellt:

- Auf UNIX- und Linux-Plattformen erstellt das DB2-Datenbankinstallationsprogramm, falls Sie eine DB2-Instanz im Fenster für die Instanzinstallation erstellen möchten, standardmäßig verschiedene Benutzer für den DB2-Verwaltungsserver (`dasusr`), den Instanzeigner (`db2inst`) und den abgeschirmten Benutzer (`db2fenc`). Sie können optional verschiedene Benutzernamen angeben.

  Das DB2-Datenbankinstallationsprogramm hängt eine Nummer von 1 bis 99 an den Standardbenutzernamen an, bis eine Benutzer-ID erstellt wird, die nicht bereits vorhanden ist. Wenn zum Beispiel die Benutzer `db2inst1` und `db2inst2` bereits vorhanden sind, erstellt das DB2-Datenbankinstallationsprogramm den Benutzer `db2inst3`. Wenn eine Nummer über 10 verwendet wird, wird der Buchstabenteil des Namens in der Standardbenutzer-ID abgeschnitten. Wenn zum Beispiel die Benutzer-ID `db2fenc9` bereits vorhanden ist, schneidet das DB2-Datenbankinstallationsprogramm das `c` der Benutzer-ID ab und hängt dann den Wert `10` an (also `db2fen10`). Dieses Abschneiden findet nicht statt, wenn der numerische Wert an die Standard-DAS-Benutzer-ID angehängt wird (z. B. `dasusr24`).
- Auf Windows-Plattformen erstellt das DB2-Datenbankinstallationsprogramm standardmäßig die Benutzer-ID `db2admin` für den DAS-Benutzer, den Instanzeigner und abgeschirmte Benutzer. (Falls erwünscht, können Sie bei der Konfiguration einen anderen Benutzernamen angeben.) Im Unterschied zu Linux- und UNIX-Plattformen, wird kein numerischer Wert an die Benutzer-ID angehängt.

Zur Minimierung des Risikos, dass ein anderer Benutzer als der Administrator von den Standardwerten erfährt und sie in unlauterer Weise in Datenbanken und Instanzen verwendet, sollten Sie die Standardwerte bei der Installation in eine neue oder vorhandene Benutzer-ID Ihrer Wahl ändern.

Anmerkung: Installationen mit Antwortdateien arbeiten nicht mit Standardwerten für Benutzer-IDs oder Gruppennamen. Diese Werte müssen in der Antwortdatei angegeben werden.

Kennwörter spielen bei der Authentifizierung von Benutzern eine wichtige Rolle. Wenn auf der Ebene des Betriebssystems keine Anforderungen an die Authentifizierung festgelegt wurden und die Datenbank das Betriebssystem zur Authentifizierung von Benutzern verwendet, wird den Benutzern die Berechtigung zur Verbindung erteilt. Unter Linux- und UNIX-Betriebssystemen werden nicht definierte Kennwörter beispielsweise wie NULL behandelt. In diesem Fall wird jeder Benutzer ohne definiertes Kennwort wie ein Benutzer mit dem Kennwort NULL behandelt. Aus der Perspektive des Betriebssystems ist dies eine Übereinstimmung, und der Benutzer erhält eine gültige Berechtigung und kann eine Verbindung zur Datenbank herstellen.

Verwenden Sie Kennwörter auf Betriebssystemebene, wenn das Betriebssystem für die Authentifizierung von Benutzern Ihrer Datenbank zuständig sein soll.

Bei der Verwendung von DB2 Data Partitioning Feature (DPF) in Linux- und UNIX-Betriebssystemumgebungen nutzt der DB2-Datenbankmanager standardmäßig das Dienstprogramm 'rsh' ('remsh' unter HP-UX) für die Ausführung einiger Befehle auf fernen Knoten. Das Dienstprogramm 'rsh' überträgt Kennwörter unverschlüsselt über das Netzwerk, was ein Sicherheitsrisiko darstellen kann, wenn sich der DB2-Server nicht in einem gesicherten Netzwerk befindet. Sie können mithilfe der Registrierdatenbankvariablen DB2RSHCMD eine sicherere Alternative für das Programm für die ferne Shell definieren, durch die dieses Sicherheitsrisiko vermieden wird. Ein Beispiel für eine sicherere Alternative ist 'ssh'. Informationen zu den Einschränkungen bei der Konfiguration ferner Shells finden Sie in der Dokumentation zur Registrierdatenbankvariablen DB2RSHCMD.

Nach der Installation des DB2-Datenbankmanagers, sollten Sie darüber hinaus die Standardzugriffsrechte, die Benutzern erteilt wurden, prüfen und (falls erforderlich) ändern. Standardmäßig erteilt der Installationsprozess den folgenden Benutzern unter den einzelnen Betriebssystemen die Zugriffsrechte eines Systemadministrators (SYSADM):

Windows-Umgebungen
: Einem gültigen DB2-Datenbankbenutzernamen, der zur Administratorgruppe gehört

Linux- und UNIX-Plattformen
: Einem gültigen DB2-Datenbankbenutzernamen, der zur Primärgruppe des Instanzeigners gehört

SYSADM-Zugriffsrechte bilden die mächtigste Gruppe von Zugriffsrechten, die innerhalb des DB2-Datenbankmanagers zur Verfügung steht. Es ist daher vielleicht nicht sinnvoll, dass alle diese Benutzer standardmäßig über SYSADM-Zugriffsrechte verfügen. Der DB2-Datenbankmanager gibt dem Administrator die Möglichkeit, Gruppen und einzelnen Benutzer-IDs Zugriffsrechte zu erteilen und zu entziehen.

Durch Aktualisieren des Konfigurationsparameters *sysadm_group* des Datenbankmanagers kann der Administrator steuern, welche Gruppe von Benutzern SYSADM-Zugriffsrechte besitzt. Sie müssen die im Folgenden beschriebenen Richtlinien befolgen, um die Sicherheitsanforderungen für die DB2-Datenbankinstallation sowie die nachfolgende Erstellung von Instanzen und Datenbanken zu erfüllen.

Jede Gruppe, die (durch Aktualisieren des Parameters *sysadm_group*) als Systemadministratorgruppe definiert wird, muss vorhanden sein. Der Name sollte diese Gruppe leicht als die Gruppe zu erkennen geben, die für Instanzeigner erstellt wurde. Die Benutzer-IDs und Gruppen, die dieser Gruppe angehören, verfügen über die Systemadministratorberechtigung für ihre jeweiligen Instanzen.

Der Administrator sollte in Betracht ziehen, eine Benutzer-ID für den Instanzeigner zu erstellen, die sich leicht als zu einer bestimmten Instanz gehörig erkennen lässt. Diese Benutzer-ID sollte als eine ihrer Gruppen den Namen der oben erstellten SYSADM-Gruppe enthalten. Eine weitere Empfehlung ist, diese Benutzer-ID des Instanzeigners nur als Mitglied der Instanzeignergruppe und in keiner anderen Gruppe zu verwenden. Dadurch wird die Verbreitung von Benutzer-IDs und Gruppen begrenzt, die die Instanz oder ein Objekt innerhalb der Instanz modifizieren können.

Der erstellten Benutzer-ID muss ein Kennwort zugewiesen werden, um eine Authentifizierung vor dem Zugriff auf die Daten und Datenbanken der Instanz zu ermöglichen. Es wird empfohlen, bei der Erstellung des Kennworts die Kennwortrichtlinien des jeweiligen Unternehmens einzuhalten.

Anmerkung: Zur Vermeidung eines unbeabsichtigten Löschens oder Überschreibens von Instanzkonfigurationsdateien und anderen Dateien, sollten Administratoren die Verwendung eines anderen Benutzerkontos, das nicht zur Primärgruppe als Instanzeigner gehört, für die täglichen, direkt auf dem Server auszuführenden Verwaltungsaufgaben in Betracht ziehen.

Vorausgesetzte Dateiberechtigungen für die Instanz- und Datenbankverzeichnisse

Das DB2-Datenbanksystem setzt voraus, dass Ihr Instanzverzeichnis und Ihr Datenbankverzeichnis mindestens folgende Berechtigungen haben.

Anmerkung: Wenn das Instanzverzeichnis und das Datenbankverzeichnis durch den DB2-Datenbankmanager erstellt werden, sind die Berechtigungen korrekt und sollten nicht geändert werden.

Auf UNIX- und Linux-Systemen müssen die folgenden Mindestberechtigungen für das Instanzverzeichnis und das Verzeichnis NODE000*x*/sqldbdir festgelegt werden: `u=rwx` und `go=rx`. Die Bedeutung der Buchstaben wird in folgender Tabelle erläutert:

Zeichen	**Stellt dar:**
u	Benutzer (Eigner)
g	Gruppe
o	Andere Benutzer
r	Lesen
w	Schreiben
x	Ausführen

Zum Beispiel gelten für die Instanz db2inst1 in /home folgende Berechtigungen:

```
drwxr-xr-x  36 db2inst1  db2grp1         4096 Jun 15 11:13 db2inst1
```

Verzeichnisse, die die Datenbanken enthalten, benötigen auf allen Verzeichnisebenen bis einschließlich NODE000*x* die folgenden Berechtigungen:

```
drwxrwxr-x  11 db2inst1  db2grp1         4096 Jun 14 15:53 NODE0000/
```

Wenn sich eine Datenbank in /db2/data/db2inst1/db2inst1/NODE0000 befindet, müssen die Verzeichnisse /db2, /db2/data, /db2/data/db2inst1, /db2/data/db2inst1/db2inst1 und /db2/data/db2inst1/db2inst1/NODE0000 die Berechtigungen `drwxrwxr-x` haben.

Innerhalb des Verzeichnisses NODE000x erfordert das Verzeichnis sqldbdir die Berechtigungen `drwxrwxr-x`. Beispiel:

```
drwx------   5 db2inst1  db2grp1          256 Jun 14 14:17 SAMPLE/
drwxr-x---   7 db2inst1  db2grp1         4096 Jun 14 13:26 SQL00001/
drwxrwxr-x   2 db2inst1  db2grp1          256 Jun 14 13:02 sqldbdir/
```

Achtung:
Zur Aufrechterhaltung der Sicherheit Ihrer Dateien sollten Sie die Berechtigungen für die Verzeichnisse in *DBNAME* (z. B. SAMPLE) und die Verzeichnisse SQL*xxxx*, die vom DB2-Datenbankmanager bei der Erstellung der Verzeichnisse zugeordnet werden, nicht ändern.

Details zur Authentifizierung

Authentifizierungsmethoden für den Server

Der Zugriff auf eine Instanz oder eine Datenbank erfordert zunächst, dass der Benutzer *authentifiziert* wird. Der *Authentifizierungstyp* für die jeweilige Instanz bestimmt, wie und wo der Benutzer überprüft wird. Der Authentifizierungstyp wird in der Konfigurationsdatei auf dem Server gespeichert. Sie wird erstmalig bei der Erstellung der Instanz definiert. Es gibt einen Authentifizierungstyp pro Instanz, der für den Zugriff auf den zugehörigen Datenbankserver und alle Datenbanken unter seiner Steuerung gilt.

Wenn Sie von einer föderierten Datenbank aus auf Datenquellen zugreifen wollen, müssen Sie die Authentifizierungsverarbeitung der Datenquellen und die Definitionen für die Authentifizierungstypen föderierter Datenbanken beachten.

Anmerkung: Auf der folgenden Website können Sie die Zertifizierungsinformationen zu den kryptografischen Routinen nachlesen, die vom DB2-Datenbankverwaltungssystem verwendet werden, um beim Authentifizierungstyp SERVER_ENCRYPT Benutzer-IDs und Kennwörter bzw. beim Authentifizierungstyp DATA_ENCRYPT Benutzer-IDs, Kennwörter und Benutzerdaten zu verschlüsseln: http://www.ibm.com/security/standards/st_evaluations.shtml.

Wechseln des Benutzers bei einer expliziten gesicherten Verbindung

Bei CLI/ODBC- und XA CLI/ODBC-Anwendungen stimmt das Authentifizierungsverfahren, das bei der Verarbeitung einer Anforderung zum Wechseln des Benutzers mit Authentifizierung verwendet wird, mit dem Verfahren überein, das auch bei der Herstellung der gesicherten Verbindung selbst angewendet wird. Aus diesem Grund wird davon ausgegangen, dass alle anderen variablen Sicherheitsattribute (z. B. Verschlüsselungsalgorithmus, Verschlüsselungsschlüssel und Plug-in-Namen), die während der Herstellung der expliziten gesicherten Verbindung verwendet wurden, bei allen Authentifizierungen gleich sind, die für eine Benutzerwechselanforderung auf dieser gesicherten Verbindung erforderlich sind. Bei Java-Anwendungen kann das Authentifizierungsverfahren bei einer Benutzerwechselanforderung (mithilfe eines Datenquellenmerkmals) geändert werden.

Da ein Objekt eines gesicherten Kontextes so definiert werden kann, dass beim Benutzerwechsel auf einer gesicherten Verbindung *keine* Authentifizierung erforderlich ist, müssen die benutzerdefinierten Sicherheits-Plug-ins folgende Bedingungen erfüllen, um die Vorteile der Funktion für den Benutzerwechsel auf einer expliziten gesicherten Verbindung voll nutzen zu können:

- Sie müssen ein reines Benutzer-ID-Token akzeptieren.
- Sie müssen eine gültige DB2-Berechtigungs-ID für diese Benutzer-ID zurückgeben.

Anmerkung: Eine explizite gesicherte Verbindung kann nicht hergestellt werden, wenn der Authentifizierungstyp CLIENT aktiv ist.

Bereitgestellte Authentifizierungstypen

Die folgenden Authentifizierungstypen stehen zur Verfügung:

SERVER
Gibt an, dass die Authentifizierung auf dem Server durch den Sicherheitsmechanismus erfolgt, der für diese Konfiguration eingerichtet ist, zum Beispiel durch ein Sicherheits-Plug-in-Modul. Der Standardsicherheitsmechanismus besteht darin, dass bei Angabe einer Benutzer-ID und eines Kennworts beim Versuch, eine Verbindung herzustellen, diese mit den gültigen Kombinationen aus Benutzer-ID und Kennwort auf dem Server verglichen werden, um festzustellen, ob dem Benutzer Zugriff auf die Instanz gewährt wird.

Anmerkung: Der Server-Code erkennt, ob eine Verbindung lokal oder fern ist. Für lokale Verbindungen sind beim Authentifizierungstyp SERVER zur erfolgreichen Authentifizierung keine Benutzer-ID und kein Kennwort erforderlich.

SERVER_ENCRYPT
Gibt an, dass der Server verschlüsselte SERVER-Authentifizierungsschemata akzeptiert. Wenn die Clientauthentifizierung nicht angegeben ist, wird der Client mit der Methode überprüft, die auf dem Server ausgewählt ist.

CLIENT
Gibt an, dass die Authentifizierung mithilfe der Sicherheitseinrichtungen des Betriebssystems in der Datenbankpartition erfolgt, in der die Anwendung aufgerufen wird. Die bei der Herstellung einer CONNECT- oder ATTACH-Verbindung angegebene Benutzer-ID und das zugehörige Kennwort werden mit den zulässigen Kombinationen aus Benutzer-ID und Kennwort verglichen, die auf dem Clientknoten hinterlegt sind. Auf diese Weise wird festgestellt, ob die Benutzer-ID über die Zugriffsberechtigung für die Instanz verfügt. Es findet keine weitere Authentifizierung auf dem Datenbankserver statt. Diese Vorgehensweise wird auch als einmalige Anmeldung oder Einzelanmeldung (Single Sign-on) bezeichnet.

Wenn der Benutzer eine lokale Anmeldung bzw. Anmeldung auf dem Client vornimmt, ist der Benutzer nur dieser lokalen Client-Workstation bekannt.

Wenn für die ferne Instanz der Authentifizierungstyp CLIENT definiert ist, bestimmen zwei weitere Parameter den endgültigen Authentifizierungstyp: *trust_allclnts* und *trust_clntauth*.

Sicherheit auf CLIENT-Ebene nur für gesicherte Clients (TRUSTED):

Gesicherte Clients sind Clients, die über ein zuverlässiges lokales Sicherheitssystem verfügen.

Wenn der Authentifizierungstyp CLIENT ausgewählt wurde, kann eine zusätzliche Option zum Schutz vor Clients ausgewählt werden, deren Betriebsumgebung keine eigenen Sicherheitssysteme hat.

Zum Schutz gegen ungesicherte Clients kann der Administrator die Authentifizierung für gesicherte Clients auswählen, indem er den Parameter *trust_allclnts* auf den Wert NO setzt. Dies impliziert, dass alle gesicherten Plattformen den Benutzer im Namen des Servers authentifizieren können. Nicht gesicherte Clients werden auf dem Server authentifiziert und müssen eine Benutzer-ID und ein Kennwort bereitstellen.

Der Konfigurationsparameter *trust_allclnts* wird verwendet, um anzugeben, ob Sie Clients vertrauen. Der Standardwert für diesen Parameter ist YES.

Anmerkung: Es ist möglich, alle Clients zu sichern (in dem Sie *trust_allclnts* auf YES setzen), auch wenn Sie einige solcher Clients haben, die über kein eigenes Sicherheitssystem zur Authentifizierung verfügen.

Vielleicht erscheint es Ihnen außerdem wünschenswert, die Authentifizierung auch für gesicherte Clients auf dem Server durchzuführen. Zur Festlegung, wo gesicherte Clients überprüft werden sollen, verwenden Sie den Konfigurationsparameter *trust_clntauth*. Der Standardwert für diesen Parameter ist CLIENT.

Anmerkung: Nur für gesicherte Clients gilt, dass wenn bei einem Verbindungsversuch mit CONNECT oder ATTACH keine Benutzer-ID oder kein Kennwort explizit angegeben wird, die Gültigkeitsprüfung auf dem Client stattfindet. Der Parameter *trust_clntauth* wird nur dazu verwendet, festzulegen, wo die Informationen, die in den Klauseln USER bzw. USING angegeben werden, zu überprüfen sind.

Zum Schutz gegen alle Clients außer DRDA-Clients von DB2 für OS/390 und z/OS, DB2 für VM und VSE sowie DB2 für System i setzen Sie die Parameter *trust_allclnts* auf DRDAONLY. Nur diese Clients dürfen die Authentifizierung auf der Clientseite ausführen. Alle anderen Clients müssen eine Benutzer-ID und ein Kennwort bereitstellen, wobei die Authentifizierung vom Server durchgeführt wird.

Der Parameter *trust_clntauth* wird verwendet, um zu bestimmen, wo die oben genannten Clients authentifiziert werden: Wenn *trust_clntauth* auf den Wert CLIENT gesetzt ist, findet die Authentifizierung auf dem Client statt. Wenn *trust_clntauth* auf den Wert SERVER gesetzt ist, wird die Authentifizierung auf dem Client ausgeführt, wenn keine Benutzer-ID und kein Kennwort angegeben werden, und auf dem Server, wenn eine Benutzer-ID und ein Kennwort angegeben werden.

Tabelle 1. Authentifizierungsmodi mit einer Kombination der Parameter TRUST_ALLCLNTS und TRUST_CLNTAUTH

TRUST_ ALLCLNTS	TRUST_ CLNTAUTH	Ungesicherte Nicht-DRDA-Clientauthentifizierung (ohne Benutzer-ID & ohne Kennwort)	Ungesicherte Nicht-DRDA-Clientauthentifizierung (mit Benutzer-ID & ohne Kennwort)	Gesicherte Nicht-DRDA-Clientauthentifizierung (ohne Benutzer-ID & ohne Kennwort)	Gesicherte Nicht-DRDA-Clientauthentifizierung (mit Benutzer-ID & ohne Kennwort)	DRDA-Clientauthentifizierung (ohne Benutzer-ID & ohne Kennwort)	DRDA-Clientauthentifizierung (mit Benutzer-ID & ohne Kennwort)
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

KERBEROS

Wird verwendet, wenn sowohl der DB2-Client als auch der DB2-Server unter Betriebssystemen arbeiten, die das Kerberos-Sicherheitsprotokoll unterstützen. Das Kerberos-Sicherheitsprotokoll führt die Authentifizierung mithilfe von Services anderer Hersteller aus und verwendet ein konventionelles Verschlüsselungsverfahren, um einen gemeinsam genutzten, geheimen Schlüssel zu erstellen. Dieser Schlüssel wird als Identitätsnachweis für den Benutzer definiert und zur Prüfung der Identität von Benutzern während aller Anforderungen für lokale Services oder Netzservices verwendet. Der Schlüssel beseitigt die Notwendigkeit, den Benutzernamen sowie das zugehörige Kennwort unverschlüsselt über das Netz zu übertragen. Mit dem Kerberos-Sicherheitsprotokoll kann eine einmalige Anmeldung (Single Sign-on) an einem fernen DB2-Datenbankserver durchgeführt werden. Der Kerberos-Authentifizierungstyp wird unter verschiedenen Betriebssystemen unterstützt; weitere Informationen finden Sie im Abschnitt mit den zugehörigen Informationen.

Die Kerberos-Authentifizierung funktioniert folgendermaßen:

1. Ein Benutzer, der sich an der Clientmaschine mit einem Domänenkonto anmeldet, wird in der Kerberos-Schlüsselverteilungszentrale (KDC - Key Distribution Center) auf dem Domänencontroller authentifiziert. Die Schlüsselverteilungszentrale gibt ein so genanntes Ticket-granting Ticket (TGT) an den Client aus.
2. Während der ersten Phase des Verbindungsaufbaus sendet der Server den Ziel-Principal-Namen, bei dem es sich um den Servicekontonamen für den Service des DB2-Datenbankservers handelt, an den Client. Mithilfe des Ziel-Principal-Namens des Servers und dem TGT fordert der Client ein Service-Ticket vom TGS (Ticket erteilenden Service, Ticket-Granting Service) an, der sich auf dem Domänencontroller befindet. Wenn sowohl das Ticket-granting Ticket des Clients als auch der Ziel-Principal-Name des Servers gültig sind, gibt der TGS ein Service-Ticket an den Client aus. Der Name des Principals, der im Datenbankverzeichnis aufgezeichnet wird, kann im Format 'name/instanz@REALM' angegeben werden. (Dieses Format ist zusätzlich zu den Formaten 'DOMÄNE\benutzerID' und 'benutzerID@xxx.xxx.xxx.com' möglich, die unter Windows akzeptiert werden.)
3. Der Client sendet dieses Service-Ticket über den Kommunikationskanal (bei dem es sich beispielsweise um TCP/IP handeln kann) an den Server.
4. Der Server prüft die Gültigkeit des Server-Tickets des Clients. Wenn das Service-Ticket des Clients gültig ist, wird die Authentifizierung abgeschlossen.

Es ist möglich, die Datenbanken auf der Clientmaschine zu katalogisieren und den Authentifizierungstyp Kerberos explizit zusammen mit dem Ziel-Principal-Namen des Servers anzugeben. Auf diese Weise lässt sich die erste Phase des Verbindungsaufbaus überspringen.

Wenn eine Benutzer-ID und ein Kennwort angegeben werden, fordert der Client das Ticket-granting Ticket für dieses Benutzerkonto an und verwendet es zur Authentifizierung.

KRB_SERVER_ENCRYPT

Gibt an, dass der Server Schemata der KERBEROS-Authentifizierung oder einer verschlüsselten SERVER-Authentifizierung akzeptiert. Wenn als Clientauthentifizierung KERBEROS angegeben wurde, wird der Client mithilfe des Kerberos-Sicherheitssystems authentifiziert. Wenn als Clientauthentifizierung SERVER_ENCRYPT angegeben ist, wird der Client mithilfe

einer Benutzer-ID und eines verschlüsselten Kennworts authentifiziert. Wenn die Clientauthentifizierung nicht angegeben ist, arbeitet der Client mit Kerberos, falls verfügbar, oder mit Kennwortverschlüsselung. Für andere Clientauthentifizierungstypen wird ein Authentifizierungsfehler zurückgegeben. Als Authentifizierungstyp des Clients darf nicht KRB_SERVER_ENCRYPT angegeben werden.

Anmerkung: Die Kerberos-Authentifizierungstypen werden auf Clients und Servern unter verschiedenen Betriebssystemen unterstützt; weitere Informationen finden Sie im Abschnitt mit den zugehörigen Informationen. Bei Windows-Betriebssystemen müssen sowohl die Client- als auch die Servermaschine entweder zur gleichen Windows-Domäne oder zu gesicherten Domänen gehören. Dieser Authentifizierungstyp sollte verwendet werden, wenn der Server Kerberos unterstützt und einige, jedoch nicht alle, Clientmaschinen die Kerberos-Authentifizierung unterstützen.

DATA_ENCRYPT
Der Server akzeptiert verschlüsselte SERVER-Authentifizierungsschemata und die Verschlüsselung von Benutzerdaten. Die Authentifizierung funktioniert in exakt gleicher Weise wie für den Authentifizierungstyp SERVER_ENCRYPT erläutert. Weitere Informationen finden Sie dort.

Die folgenden Benutzerdaten werden bei Verwendung dieses Authentifizierungstyps verschlüsselt:
- SQL- und XQuery-Anweisungen
- Daten von SQL-Programmvariablen
- Ausgabedaten aus der Serververarbeitung einer SQL- oder XQuery-Anweisung, einschließlich einer Beschreibung der Daten
- Einige oder alle Antwortgruppendaten, die aus einer Abfrage resultieren
- LOB-Datenströme (große Objekte)
- SQLDA-Deskriptoren

DATA_ENCRYPT_CMP
Der Server akzeptiert verschlüsselte SERVER-Authentifizierungsschemata und die Verschlüsselung von Benutzerdaten. Darüber hinaus bietet dieser Authentifizierungstyp Kompatibilität mit Produkten früherer Versionen, die den Authentifizierungstyp DATA_ENCRYPT nicht unterstützen. Diese Produkte erhalten die Möglichkeit, die Verbindung mit dem Authentifizierungstyp SERVER_ENCRYPT und ohne Verschlüsselung von Benutzerdaten herzustellen. Produkte, die den neuen Authentifizierungstyp unterstützen, müssen ihn verwenden. Dieser Authentifizierungstyp ist nur in der Konfigurationsdatei des Datenbankmanagers des Servers, jedoch nicht im Befehl CATALOG DATABASE gültig.

GSSPLUGIN
Gibt an, dass der Server ein GSS-API-Plug-in zur Ausführung der Authentifizierung verwendet. Wenn die Clientauthentifizierung nicht angegeben ist, gibt der Server eine Liste der vom Server unterstützten Plug-ins, einschließlich aller Kerberos-Plug-ins, die im Konfigurationsparameter *srvcon_gssplugin_list* des Datenbankmanagers aufgelistet sind, an den Client zurück. Der Client wählt das erste Plug-in in der Liste aus, das im Plug-in-Verzeichnis des Clients zu finden ist. Wenn der Client kein Plug-in der Liste unterstützt, wird der Client mit dem Kerberos-Authentifizierungsschema authentifiziert (falls dies zurückgegeben wird). Wenn der Clientauthentifizierungstyp das Authentifizierungsschema GSSPLUGIN angibt, wird der Client mit dem ersten unterstützten Plug-in der Liste authentifiziert.

GSS_SERVER_ENCRYPT

Gibt an, dass der Server Plug-in-Authentifizierungsschemata oder verschlüsselte SERVER-Authentifizierungsschemata akzeptiert. Wenn die Clientauthentifizierung durch ein Plug-in erfolgt, wird der Client mit dem ersten vom Client unterstützten Plug-in in der Liste der vom Server unterstützten Plug-ins authentifiziert.

Wenn die Clientauthentifizierung nicht angegeben ist und eine implizite Verbindung hergestellt wird (d. h., der Client gibt bei der Herstellung der Verbindung keine Benutzer-ID und kein Kennwort an), gibt der Server eine Liste der vom Server unterstützten Plug-ins, das Kerberos-Authentifizierungsschema (wenn eines der Plug-ins in der Liste auf Kerberos basiert) und das verschlüsselte Serverauthentifizierungsschema zurück. Der Client wird mit dem ersten unterstützten Plug-in in der Liste authentifiziert, das im Plug-in-Verzeichnis des Clients zu finden ist. Wenn der Client keines der Plug-ins der Liste unterstützt, wird der Client mit dem Kerberos-Authentifizierungsschema authentifiziert. Wenn der Client das Kerberos-Authentifizierungsschema nicht unterstützt, wird der Client mit dem verschlüsselten SERVER-Authentifizierungsschema authentifiziert und die Verbindung schlägt fehl, weil das Kennwort fehlt. Ein Client unterstützt das Kerberos-Authentifizierungsschema, wenn für das Betriebssystem ein von DB2 bereitgestelltes Kerberos-Plug-in vorhanden ist oder ein Kerberos-basiertes Plug-in im Konfigurationsparameter *srvcon_gssplugin_list* des Datenbankmanagers angegeben ist.

Wenn die Clientauthentifizierung nicht angegeben ist und eine explizite Verbindung hergestellt wird (d. h., die Benutzer-ID und das Kennwort werden angegeben), ist der Authentifizierungstyp mit SERVER_ENCRYPT äquivalent.

Anmerkung:

1. Sperren Sie sich nicht versehentlich von Ihrer Instanz aus, wenn Sie Berechtigungsinformationen ändern, da der Zugriff auf die Konfigurationsdatei selbst durch Informationen in der Konfigurationsdatei geschützt wird. Mit den folgenden Parametern in der Konfigurationsdatei des Datenbankmanagers wird der Zugriff auf die Instanz gesteuert:
   - AUTHENTICATION *
   - SYSADM_GROUP *
   - TRUST_ALLCLNTS
   - TRUST_CLNTAUTH
   - SYSCTRL_GROUP
   - SYSMAINT_GROUP

   * kennzeichnet die beiden wichtigsten Parameter, die am ehesten Probleme verursachen können.

   Es gibt einige Maßnahmen, um dies zu verhindern: Wenn Sie sich versehentlich aus dem DB2-Datenbanksystem aussperren, gibt es auf allen Plattformen eine Sicherheitsoption, die Ihnen ermöglicht, die normalen Sicherheitsprüfungen des DB2-Datenbanksystems außer Kraft zu setzen, um die Konfigurationsdatei des Datenbankmanagers über einen Sicherheitsbenutzer des lokalen Betriebssystems, der über Zugriffsrechte einer hohen Berechtigungsstufe verfügt, zu aktualisieren. Dieser Benutzer hat *immer* das Zugriffsrecht zur Aktualisierung der Konfigurationsdatei des Datenbankmanagers und kann daher das Problem beheben. Diese Umgehung der Sicherheit ist jedoch auf eine lokale Aktualisierung der Konfigurationsdatei des Datenbankmanagers beschränkt. Sie können

einen Sicherheitsbenutzer nicht über Remotezugriff oder für irgendeinen anderen DB2-Datenbankbefehl verwenden. Dieser Benutzer mit Sonderberechtigung wird folgendermaßen identifiziert:

- UNIX-Plattformen: der Instanzeigner.
- Windows-Plattform: Eine zur lokalen Gruppe der „Administratoren" gehörende Person.
- Andere Plattformen: Da es auf anderen Plattformen keine lokalen Sicherheitseinrichtungen gibt, bestehen alle Benutzer lokale Sicherheitsprüfungen ohnehin.

Authentifizierungsaspekte für ferne Clients

Bei der Katalogisierung einer Datenbank für den fernen Zugriff, kann der Authentifizierungstyp im Datenbankverzeichniseintrag angegeben werden.

Der Authentifizierungstyp ist nicht erforderlich. Wenn er nicht angegeben ist, nimmt der Client den Standardwert SERVER_ENCRYPT an. Wenn der Server jedoch den Authentifizierungstyp SERVER_ENCRYPT nicht unterstützt, versucht es der Client mit einem vom Server unterstützten Wert erneut. Wenn der Server mehrere Authentifizierungstypen unterstützt, trifft der Client keine Auswahl unter diesen, sondern gibt stattdessen einen Fehler zurück. Die Rückgabe eines Fehlers soll sicherstellen, dass der korrekte Authentifizierungstyp verwendet wird. In diesem Fall muss der Client die Datenbank mit einem unterstützten Authentifizierungstyp katalogisieren. Wenn ein Authentifizierungstyp angegeben wird, kann die Authentifizierung unverzüglich beginnen, sofern der angegebene Wert mit dem auf dem Server übereinstimmt. Wenn keine Übereinstimmung erkannt wird, unternimmt die DB2-Datenbank einen Behebungsversuch. Dieser Behebungsversuch kann zum Ausgleich des Unterschieds zu mehr Übertragungen führen oder einen Fehler verursachen, wenn keine Behebung durch die DB2-Datenbank möglich ist. Im Fall einer fehlenden Übereinstimmung wird angenommen, dass der Wert auf dem Server korrekt ist.

Mit dem Authentifizierungstyp DATA_ENCRYPT_CMP können Clients von einem Vorgängerrelease, das keine Datenverschlüsselung unterstützt, eine Verbindung zu einem Server mithilfe der Authentifizierung SERVER_ENCRYPT anstelle von DATA_ENCRYPT herstellen. Diese Authentifizierung funktioniert nicht, wenn Folgendes gilt:

- Die Version des Clients ist Version 7.2.
- Die Version des Gateways ist Version 8 FixPak 7 oder höher.
- Der Server ist Version 8 FixPak 7 oder höher.

Wenn diese Bedingungen alle zutreffen, kann der Client keine Verbindung zum Server herstellen. Damit die Verbindung hergestellt werden kann, müssen Sie entweder für den Client einen Upgrade auf Version 8 ausführen, oder das Gateway darf höchstens die Version 8 FixPak 6 besitzen.

Die Festlegung des zur Herstellung einer Verbindung verwendeten Authentifizierungstyps erfolgt durch die Angabe des geeigneten Authentifizierungstyps in Form eines Datenbankkatalogeintrags auf dem Gateway. Dies gilt sowohl für DB2 Connect-Szenarios als auch für Clients und Server in einer Umgebung mit partitionierten Datenbanken, in der für den Client die Registrierdatenbankvariable DB2NODE definiert ist. Sie katalogisieren den Authentifizierungstyp in der Katalogpartition mit der Absicht, zur entsprechenden Partition zu „springen" (engl. „hop"). In einem solchen Szenario wird der auf dem Gateway katalogisierte Authentifizierungstyp nicht verwendet, weil die Vereinbarung ausschließlich zwischen dem Client und dem Server stattfindet.

Sie müssen eventuell mehrere Datenbankaliasnamen auf dem Gateway mit unterschiedlichen Authentifizierungstypen katalogisieren, wenn Clients betrieben werden müssen, die unterschiedliche Authentifizierungstypen verwenden. Bei der Entscheidung, welcher Authentifizierungstyp auf einem Gateway zu katalogisieren ist, können Sie den gleichen Authentifizierungstyp verwenden wie auf dem Client und dem Server. Alternativ können Sie auch den Authentifizierungstyp NOTSPEC verwenden, wobei dieser Wert standardmäßig als SERVER interpretiert wird.

Authentifizierungsaspekte bei partitionierten Datenbanken

In einer partitionierten Datenbank muss jede Partition der Datenbank über dieselbe Menge definierter Benutzer und Gruppen verfügen. Wenn die Definitionen nicht übereinstimmen, kann der Benutzer in verschiedenen Partitionen zu verschiedenen Operationen berechtigt sein. Konsistenz über alle Partitionen hinweg ist zu empfehlen.

Kerberos-Authentifizierung - Details

Das DB2-Datenbanksystem bietet Unterstützung für das Kerberos-Authentifizierungsprotokoll unter den Betriebssystemen AIX, Solaris, Linux IA32 und AMD64 sowie unter Windows.

Die Kerberos-Unterstützung wird in Form eines GSS-API-Sicherheits-Plug-ins mit dem Namen „IBMkrb5" zur Verfügung gestellt, das als Authentifizierungs-Plug-in sowohl auf dem Server als auch auf dem Client verwendet wird. Die Bibliothek befindet sich unter UNIX und Linux in den Verzeichnissen sqllib/security{32|64}/plugin/IBM/{client|server} und unter Windows in den Verzeichnissen sqllib/security/plugin/IBM{client|server}.

Anmerkung: Bei 64-Bit-Windows-Systemen hat die Plug-in-Bibliothek den Namen IBMkrb564.dll. Darüber hinaus ist der tatsächliche Plug-in-Quellcode IBMkrb5.C für das UNIX- und Linux-Plug-in im Verzeichnis sqllib/samples/security/plugins verfügbar.

Es wird dringend empfohlen, sich gute Kenntnisse über die Verwendung und Konfiguration von Kerberos anzueignen, bevor Sie versuchen, die Kerberos-Authentifizierung mit dem DB2-Datenbanksystem zu verwenden.

Kerberos - Beschreibung und Einführung

Kerberos ist ein Netzwerkauthentifizierungsprotokoll eines Fremdanbieters, das mit einem System gemeinsamer geheimer Schlüssel (Shared Secret Keys) zur sicheren Authentifizierung eines Benutzers in einer ungesicherten Netzwerkumgebung arbeitet. Kerberos implementiert ein dreistufiges System, in dem verschlüsselte Tickets (die von einem getrennten Server bereitgestellt werden, der als Kerberos Key Distribution Center (Kerberos-Schlüsselverteilungszentrale) bzw. kurz KDC bezeichnet wird) zwischen dem Anwendungsserver und dem Client anstelle von Paaren aus Benutzer-ID und Kennwort im Textformat ausgetauscht werden. Diese verschlüsselten Service-Tickets (auch als *Credentials* - Berechtigungsnachweise bezeichnet) haben eine endliche Gültigkeitsdauer und werden nur vom Client und dem Server verstanden. Dies verringert das Sicherheitsrisiko, selbst wenn das Ticket im Netzwerk abgefangen wird. Jeder Benutzer, bzw. in der Kerberos-Terminologie *Principal*, besitzt einen privaten Chiffrierschlüssel, der mit dem KDC gemeinsam verwendet wird. Zusammengenommen werden die Gruppe von Principals und Computern, die bei einem KDC registriert sind, als *Realm* bezeichnet.

Eine Schlüsselfunktion von Kerberos besteht darin, eine Einzelanmeldungsumgebung (Single Sign-on) zu ermöglichen, in der ein Benutzer seine Identität den Ressourcen innerhalb des Kerberos-Realms gegenüber nur einmal nachweisen muss. Für die Arbeit mit der DB2-Datenbank bedeutet dies, dass ein Benutzer in der Lage ist, eine Verbindung zu einem DB2-Datenbankserver herzustellen (CONNECT oder ATTACH), ohne eine Benutzer-ID und ein Kennwort anzugeben. Ein weiterer Vorteil ist der, dass die Verwaltung von Benutzer-IDs vereinfacht wird, da ein zentrales Repository für Principals verwendet wird. Und schließlich unterstützt Kerberos eine gegenseitige Authentifizierung, sodass der Client auch die Identität des Servers überprüfen kann.

Kerberos-Einrichtung

Das DB2-Datenbanksystem und die implementierte Unterstützung von Kerberos setzen voraus, dass die Kerberos-Schicht auf allen beteiligten Maschinen vor der Einbindung des DB2-Datenbanksystems ordnungsgemäß installiert und konfiguriert wurde. Dazu gehört unter anderem die Erfüllung der folgenden Voraussetzungen:

1. Die Client- und Servermaschinen sowie die Principals müssen zum gleichen Realm oder ansonsten zu vertrauten (gesicherten) Realms (bzw. vertrauten Domänen in der Windows-Terminologie) gehören.
2. Es müssen geeignete Principals erstellt werden.
3. Es müssen Serverchiffrierschlüsseldateien (keytab) nach Bedarf erstellt werden.
4. Auf allen beteiligten Maschinen müssen die Systemuhren synchronisiert werden (Kerberos lässt in der Regel einen Zeitunterschied von fünf Minuten zu, ansonsten kann ein Vorauthentifizierungsfehler beim Empfang des Berechtigungsnachweises auftreten).

Detaillierte Informationen zur Installation und Konfiguration von Kerberos finden Sie in der Dokumentation des installierten Kerberos-Produkts.

Für das DB2-Datenbanksystem ist allein wichtig, ob der Kerberos-Sicherheitskontext auf der Basis der von der Anwendung, die die Verbindung herstellt, bereitgestellten Berechtigungsnachweise erfolgreich erstellt wird (d. h., ob die Authentifizierung erfolgreich ist). Andere Kerberos-Funktionen, wie zum Beispiel die Signierung oder Verschlüsselung von Nachrichten, werden nicht verwendet. Darüber hinaus wird, sofern verfügbar, die gegenseitige Authentifizierung unterstützt.

Für Kerberos gelten folgende Voraussetzungen:

- Für die AIX-, Solaris-Betriebsumgebungs- und Linux-Plattformen ist IBM Network Authentication Service (NAS) Toolkit v1.4 oder höher erforderlich. Sie können NAS-Toolkits unter https://www6.software.ibm.com/dl/dm/dm-nasherunterladen.
- Für die Windows-Plattformen gibt es keine Voraussetzungen.

Kerberos und Client-Principals

Der Principal kann entweder in einem zweiteiligen oder mehrteiligen Format vorliegen (d. h. in der Form *name@REALM* oder *name/instanz@REALM*). Da der Wert für „name" in der Zuordnung der Berechtigungs-ID (AUTHID) verwendet wird, muss der Name den Namenskonventionen des DB2-Datenbanksystems entsprechen. Dies bedeutet, dass der Name bis zu 30 Zeichen lang sein darf und die vorhandenen Einschränkungen bei der Auswahl der Zeichen berücksichtigt. (Die AUTHID-Zuordnung wird in einem späteren Abschnitt behandelt.)

Anmerkung: Windows ordnet den Kerberos-Principal direkt einem Domänenbenutzer zu. Daraus folgt, dass die Kerberos-Authentifizierung für Windows-Maschinen, die nicht zu einer Domäne oder einem Realm gehören, nicht zur Verfügung steht. Darüber hinaus unterstützt Windows lediglich zweiteilige Namen (d. h. im Format *name@domäne*).

Der Principal selbst muss in der Lage sein, abgehende Berechtigungsnachweise abzurufen, mit denen er Service-Tickets für die Zieldatenbank anfordern und empfangen kann. Dies geschieht normalerweise mithilfe des Befehls kinit unter UNIX oder Linux und erfolgt unter Windows implizit bei der Anmeldung.

Kerberos und Zuordnung von Berechtigungs-IDs

Im Unterschied zu Benutzer-IDs von Betriebssystemen, deren Existenzbereich im Normalfall auf eine einzige Maschine beschränkt ist, bieten Kerberos-Principals die Möglichkeit einer Authentifizierung auch in anderen Realms als dem eigenen. Das potenzielle Problem doppelter Namen von Principals wird dadurch vermieden, dass der Principal mit dem Realmnamen vollständig qualifiziert wird. In Kerberos hat ein vollständig qualifizierter Principal die Form name/instanz@REALM. Dabei kann das Instanzfeld tatsächlich sogar mehrere, durch „/" getrennte Instanzen enthalten, d. h. name/instanz1/instanz2@REALM, oder es kann ganz weggelassen werden. Dies unterliegt natürlich der Einschränkung, dass der Realmname innerhalb aller im Netzwerk definierten Realms eindeutig sein muss. Das Problem für das DB2-Datenbanksystem besteht darin, dass zur Bereitstellung einer einfachen Zuordnung zwischen Principal und Berechtigungs-ID (AUTHID) eine Eins-zu-eins-Zuordnung zwischen dem Namen des Principals, das heißt dem Wert für „name" im vollständig qualifizierten Principal, und der Berechtigungs-ID wünschenswert ist. Eine einfache Zuordnung ist erforderlich, da die Berechtigungs-ID als Standardschema im DB2-Datenbanksystem verwendet wird und daher einfach und logisch abgeleitet werden sollte. Infolgedessen muss der Datenbankadministrator mit den folgenden potenziellen Problemen rechnen:

- Principals aus verschiedenen Realms, jedoch mit dem gleichen Namen, werden derselben Berechtigungs-ID (AUTHID) zugeordnet.
- Principals mit dem gleichen Namen, jedoch verschiedenen Instanzen, werden derselben Berechtigungs-ID (AUTHID) zugeordnet.

In Anbetracht der oben genannten Umstände ergeben sich die folgenden Empfehlungen:

- Pflegen Sie einen eindeutigen Namensbereich (Namespace) für den Namen innerhalb aller vertrauten (gesicherten) Realms, von denen aus auf den DB2-Datenbankserver zugegriffen wird.
- Alle Principals mit dem gleichen Namen sollten unabhängig von der Instanz demselben Benutzer gehören.

Kerberos und Server-Principals

Unter UNIX oder Linux wird als Name des Server-Principals für die DB2-Datenbankinstanz ein Name mit dem Format <instanzname>/<vollständig qualifizierter hostname>@REALM angenommen. Dieser Principal muss in der Lage sein, Kerberos-Sicherheitskontexte zu akzeptieren. Außerdem muss er vorhanden sein, bevor die DB2-Datenbankinstanz gestartet wird, da der Servername durch das Plug-in bei der Initialisierung an das DB2-Datenbanksystem gemeldet wird.

Unter Windows wird als Principal für den Server das Domänenkonto angenommen, unter dem der DB2-Datenbankservice gestartet wurde. Eine Ausnahme bildet der Fall, dass die Instanz durch das lokale Systemkonto gestartet wird. In diesem Fall wird der Name des Server-Principals im Format host/<hostname> gemeldet. Dies ist nur gültig, wenn sowohl der Client als auch der Server zu Windows-Domänen gehören.

Windows unterstützt keine Namen, die aus mehr als zwei Teilen bestehen. Dies stellt ein Problem dar, wenn ein Windows-Client versucht, eine Verbindung zu einem UNIX-Server aufzubauen. Daher muss eventuell eine Zuordnung zwischen dem Kerberos-Principal und dem Windows-Konto in der Windows-Domäne definiert werden, wenn Interoperabilität mit UNIX-Kerberos erforderlich ist. (Informationen finden Sie in der entsprechenden Microsoft-Dokumentation.)

Sie können den Namen des Kerberos-Server-Principals, der vom DB2-Server unter den Betriebssystemen UNIX und Linux verwendet wird, überschreiben. Setzen Sie die Umgebungsvariable DB2_KRB5_PRINCIPAL auf den vollständig qualifizierten Namen des Server-Principals. Die Instanz muss erneut gestartet werden, da der Name des Server-Principals vom DB2-Datenbanksystem nur erkannt wird, wenn der Befehl **db2start** ausgeführt wird.

Kerberos-Chiffrierschlüsseldateien

Jeder Kerberos-Service unter UNIX oder Linux, der Sicherheitskontextanforderungen akzeptieren soll, muss seine Berechtigungsnachweise in einer Chiffrierschlüsseldatei (*keytab*) hinterlegen. Dies gilt für die Principals, die als Server-Principals vom DB2-Datenbanksystem verwendet werden. Der Schlüssel des Servers wird nur in der Standardchiffrierschlüsseldatei gesucht. Anweisungen zum Hinzufügen eines Schlüssels zur Chiffrierschlüsseldatei finden Sie in der Dokumentation zu Ihrem Kerberos-Produkt.

Das Konzept einer Chiffrierschlüsseldatei ist unter Windows unbekannt. Das System sorgt automatisch für das Speichern und Abrufen der Kennungen für Berechtigungsnachweise für einen Principal.

Kerberos und Gruppen

Kerberos ist ein Authentifizierungsprotokoll, das kein Gruppenkonzept besitzt. Infolgedessen greift das DB2-Datenbanksystem auf das lokale Betriebssystem zurück, um eine Gruppenliste für den Kerberos-Principal abzurufen. Für UNIX oder Linux setzt dies voraus, dass ein äquivalentes Systemkonto für jeden Principal vorhanden sein muss. Für den Principal name@REALM sammelt das DB2-Datenbanksystem zum Beispiel Gruppeninformationen, indem das lokale Betriebssystem nach den Namen aller Gruppen abgefragt wird, zu denen der Betriebssystembenutzer *name* gehört. Wenn kein Betriebssystembenutzer vorhanden ist, wird die Berechtigungs-ID (AUTHID) nur der Gruppe PUBLIC zugeordnet. Windows hingegen ordnet einem Kerberos-Principal automatisch ein Domänenkonto zu, und der zusätzliche Schritt zur Erstellung eines getrennten Betriebssystemkontos ist nicht erforderlich.

Aktivieren der Kerberos-Authentifizierung auf dem Client

Der Konfigurationsparameter *clnt_krb_plugin* des Datenbankmanagers muss mit dem Namen des verwendeten Kerberos-Plug-ins aktualisiert werden. Auf den unterstützten Plattformen muss dieser Parameter auf den Wert IBMkrb5 gesetzt werden. Dieser Parameter teilt dem DB2-Datenbanksystem mit, dass Kerberos zum Verbindungsaufbau und zu lokalen Aktionen auf Instanzebene verwendet werden kann, wenn der Parameter AUTHENTICATION auf den Wert KERBEROS oder KRB_SERVER_ENCRYPT gesetzt ist. Ansonsten wird keine clientseitige Kerberos-Unterstützung angenommen.

Anmerkung: Es werden keine Prüfungen durchgeführt, um festzustellen, ob die Kerberos-Unterstützung verfügbar ist.

Optional kann beim Katalogisieren einer Datenbank auf dem Client der Authentifizierungstyp wie folgt angegeben werden:

```
db2 catalog db testdb at node testnode authentication kerberos target
   principal service/host@REALM
```

Wenn die Authentifizierungsinformationen jedoch nicht angegeben werden, sendet der Server dem Client den Namen des Server-Principals.

Aktivieren der Kerberos-Authentifizierung auf dem Server

Der Konfigurationsparameter *srvcon_gssplugin_list* des Datenbankmanagers muss mit dem Namen des Kerberos-Plug-ins des Servers aktualisiert werden. Obwohl dieser Parameter eine Liste von unterstützten Plug-ins enthalten kann, darf nur ein Kerberos-Plug-in angegeben werden. Wenn dieses Feld jedoch leer ist und AUTHENTICATION auf den Wert KERBEROS oder KRB_SERVER_ENCRYPT gesetzt ist, wird das Kerberos-Standard-Plug-in (IBMkrb5) angenommen und verwendet. Es muss entweder der Parameter AUTHENTICATION oder der Parameter SVRCON_AUTH auf den Wert KERBEROS oder KRB_SERVER_ENCRYPT gesetzt werden, wenn die Kerberos-Authentifizierung abhängig davon verwendet werden soll, ob sie für alle Verbindungen oder nur für ankommende Verbindungen zu verwenden ist.

Erstellen eines Kerberos-Plug-ins

Für die Erstellung eines Kerberos-Plug-ins sind verschiedene Aspekte zu beachten:

- Schreiben Sie ein Kerberos-Plug-in als GSS-API-Plug-in mit der besonderen Ausnahme, dass der Parameter *plugintype* im Funktionszeigerfeld, das an das DB2-Datenbanksystem in der Initialisierungsfunktion zurückgegeben wird, auf den Wert DB2SEC_PLUGIN_TYPE_KERBEROS gesetzt werden muss.
- Unter bestimmten Bedingungen kann der Name des Server-Principals durch den Server an den Client gemeldet werden. Als solcher sollte der Name des Principals nicht im GSS_C_NT_HOSTBASED_SERVICE-Format (service@host) angegeben werden, da DRDA festlegt, dass der Name des Principals im GSS_C_NT_USER_NAME-Format (server/host@REALM) anzugeben ist.
- In einer typischen Situation kann die Standardchiffrierschlüsseldatei (keytab) durch die Umgebungsvariable KRB5_KTNAME angegeben werden. Da das Server-Plug-in jedoch innerhalb eines Prozesses der DB2-Datenbanksteuerkomponente ausgeführt wird, ist vielleicht kein Zugriff auf diese Umgebungsvariable möglich.

Kompatibilität mit zSeries und System i

Für Verbindungen zu zSeries und System i muss die Datenbank mit dem Parameter AUTHENTICATION KERBEROS katalogisiert werden, und der Parametername TARGET PRINCIPAL muss explizit angegeben werden.

Weder zSeries noch System i unterstützen eine gegenseitige Authentifizierung.

Windows-Aspekte

Wenn Sie Kerberos auf Windows-Plattformen verwenden, müssen Sie Folgendes beachten:

- Aufgrund der Art und Weise, wie Windows Fehler erkennt und meldet, können die folgenden Bedingungen zu einem unerwarteten Fehler (SQL30082N, RC=36) des Sicherheitsplug-ins des Clients führen:
  - Abgelaufenes Konto
  - Ungültiges Kennwort
  - Abgelaufenes Kennwort
  - Vom Administrator erzwungene Kennwortänderung
  - Inaktiviertes Konto

  Darüber hinaus gibt das DB2-Verwaltungsprotokoll bzw. die Datei db2diag.log in allen Fällen einen Fehler "Anmeldung fehlgeschlagen" bzw. "Anmeldung verweigert" an.
- Wenn ein Domänenkontoname auch lokal definiert ist, schlagen Verbindungsanforderungen, die den Domänennamen und das Kennwort explizit angeben, mit dem folgenden Fehler fehl: `Die lokale Sicherheitsautorität (LSA) ist nicht erreichbar`.

  Der Fehler wird dadurch verursacht, dass Windows den lokalen Benutzer zuerst lokalisiert. Die Lösung besteht darin, den Benutzer in der Verbindungszeichenfolge vollständig qualifiziert anzugeben. Beispiel: `name@DOMÄNE.IBM.COM`
- Windows-Konten dürfen in ihren Namen das Zeichen @ nicht enthalten, da es vom DB2-Kerberos-Plug-in als Domänentrennzeichen interpretiert wird.
- Bei der Arbeit mit einer Nicht-Windows-Plattform müssen Sie sicherstellen, dass alle Konten des Windows-Domänenservers und alle Konten von Windows-Clients für die Verwendung der DES-Verschlüsselung konfiguriert sind. Wenn das Konto, das zum Starten des DB2-Service verwendet wird, nicht zur Verwendung der DES-Verschlüsselung konfiguriert ist, ist der DB2-Server nicht in der Lage, Kerberos-Kontexte zu akzeptieren. Insbesondere empfängt DB2 einen unerwarteten Server-Plug-in-Fehler und protokolliert, dass die API AcceptSecurityContext den Code SEC_I_CONTINUE_NEEDED (0x00090312L) zurückgegeben hat.

  Um festzustellen, ob Windows-Konten zur Verwendung der DES-Verschlüsselung konfiguriert sind, prüfen Sie die Informationen unter den Kontoeigenschaften im **Active Directory**. Wenn die Kontoeigenschaften geändert werden, ist eventuell ein Neustart erforderlich.
- Wenn sowohl der Client als auch der Server unter Windows ausgeführt werden, kann der DB2-Service unter dem lokalen Systemkonto gestartet werden. Wenn sich der Client und der Server jedoch in unterschiedlichen Domänen befinden, kann die Herstellung der Verbindung mit einem Fehler wegen eines ungültigen Namens für den Zielprincipal fehlschlagen. Dieses Problem lässt sich umgehen, indem der Name des Zielprincipals auf dem Client explizit mit dem vollständig qualifizierten Hostnamen und dem vollständig qualifizierten Domänennamen des Servers im folgenden Format katalogisiert wird: host/*hostname des servers@domänenname des servers*.

Beispiel: `host/meinhost.domäne.ibm.com@DOMÄNE.IBM.COM`

Ansonsten müssen Sie den DB2-Service unter einem gültigen Domänenkonto starten.

Kennwortverwaltung auf Servern

Möglicherweise müssen Sie Tasks für die Kennwortverwaltung ausführen. Da derartige Tasks im Allgemeinen auf dem Server ausgeführt werden, viele Benutzer jedoch mit der Arbeit in der Serverumgebung nicht vertraut sind, stellt das Ausführen dieser Tasks teilweise eine schwer zu bewältigende Herausforderung dar.

Das DB2-Datenbanksystem stellt eine Möglichkeit zur Verfügung, mit der Kennwörter aktualisiert und überprüft werden können; dabei ist die Existenz auf dem Server nicht erforderlich. Sie können mit den DB2-Produkten Kennwörter unter AIX-, Linux- und Windows-Betriebssystemen ändern.

Wird z. B. die Fehlernachricht SQL1404N „Das Kennwort ist abgelaufen" oder die Fehlernachricht SQL30082N „Die Sicherheitsverarbeitung ist mit Ursachencode 1 (PASSWORD EXPIRED) fehlgeschlagen" empfangen, können Sie das Kennwort wie folgt mit der Anweisung CONNECT ändern:

```
CONNECT TO datenbank USER benutzer-id USING kennwort NEW neues_kennwort
CONFIRM neues_kennwort
```

Sie können das Kennwort ebenso mit dem Befehl ATTACH oder über das Dialogfenster des DB2-Konfigurationsassistenten zum Ändern von Kennwörtern ändern.

Berechtigungen, Zugriffsrechte und Objekteigentumsrecht

Benutzer (angegeben durch eine Berechtigungs-ID) können SQL- oder XQuery-Anweisungen nur erfolgreich ausführen, wenn sie über die Berechtigung zum Ausführen der angegebenen Funktion verfügen. Zur Erstellung einer Tabelle muss ein Benutzer berechtigt sein, Tabellen zu erstellen, zur Änderung einer Tabelle muss ein Benutzer berechtigt sein, die Tabelle zu ändern, usw.

Es gibt drei Formen der Berechtigung: *Administratorberechtigung*, *Zugriffsrechte* und *LBAC-Berechtigungsnachweise*, die nachfolgend behandelt werden.

Der Datenbankmanager setzt voraus, dass jeder Benutzer entweder implizit oder explizit zur Verwendung der jeweiligen Datenbankfunktion speziell berechtigt wird, um eine bestimmte Task auszuführen. *Explizite* Berechtigungen oder Zugriffsrechte werden dem Benutzer erteilt (GRANTEETYPE hat den Wert U in den Datenbankkatalogen). *Implizite* Berechtigungen oder Zugriffsrechte werden einer Gruppe, der der Benutzer angehört (GRANTEETYPE hat den Wert G in den Datenbankkatalogen), oder einer Rolle, zu der der Benutzer, die Gruppe oder eine andere Rolle gehören (GRANTEETYPE hat den Wert R in den Datenbankkatalogen), erteilt.

Administratorberechtigung

Die Person bzw. die Personen mit Administratorberechtigung muss/müssen den Datenbankmanager steuern und ist/sind für die Sicherheit und Integrität der Daten verantwortlich. Diejenigen mit den Administratorberechtigungsstufen SYSADM und DBADM verfügen implizit über alle Zugriffsrechte für alle Objekte, mit Ausnahme der Objekte, die zur Datenbanksicherheit gehören, und steuern, welche Personen Zugriff auf den Datenbankmanager und den Speicherbereich dieses Zugriffs haben.

Berechtigungsstufen bilden eine Methode, Zugriffsrechte sowie Operationen zur Wartung des Datenbankmanagers und Dienstprogrammoperationen auf einer höheren Stufe in Gruppen zusammenzufassen. *Datenbankberechtigungen* ermöglichen Benutzern die Ausführung von Aktivitäten auf der Datenbankebene. Ein Benutzer, eine Gruppe oder eine Rolle kann über eine oder mehrere der nachfolgend aufgeführten Berechtigungen verfügen:

- Administratorberechtigungsstufe, die auf Instanzebene operiert, SYSADM (Systemadministrator):

  Die Berechtigungsstufe SYSADM berechtigt zur Steuerung aller Ressourcen, die vom Datenbankmanager erstellt und verwaltet werden. Der Systemadministrator besitzt die Berechtigungen DBADM, SYSCTRL, SYSMAINT und SYSMON sowie die Berechtigung, die Berechtigungen DBADM und SECADM zu erteilen oder zu widerrufen.

  Der Benutzer, der die Berechtigung SYSADM besitzt, ist sowohl für die Steuerung des Datenbankmanagers als auch für die Gewährleistung der Sicherheit und Integrität der Daten zuständig. Die Berechtigung SYSADM stellt die implizite Berechtigung DBADM in einer Datenbank bereit, jedoch nicht die implizite Berechtigung SECADM in einer Datenbank.
- Administratorberechtigungsstufen, die auf Datenbankebene operieren:
  - DBADM (Datenbankadministrator)

    Die DBADM-Berechtigungsebene gilt auf der Datenbankebene und stellt die Verwaltungsberechtigung über eine einzige Datenbank bereit. Dieser Datenbankadministrator besitzt die Zugriffsrechte, die zur Erstellung von Objekten, zur Ausführung von Datenbankbefehlen und zum Zugriff auf Tabellendaten erforderlich sind. Der Datenbankadministrator kann außerdem das Zugriffsrecht CONTROL und Einzelzugriffsrechte erteilen oder widerrufen.
  - SECADM (Sicherheitsadministrator)

    Die Berechtigungsstufe SECADM ist auf der Datenbankebene gültig. Dabei handelt es sich um die Berechtigung, die zum Erstellen, Ändern (sofern anwendbar) und Löschen von Rollen, gesicherten Kontexten, Prüfrichtlinien, Sicherheitskennsatzkomponenten, Sicherheitsrichtlinien und Sicherheitskennsätzen erforderlich ist, die zum Schutz von Tabellen verwendet werden. Diese Berechtigung ist auch zum Erteilen und Widerrufen von Rollen, Sicherheitskennsätzen und Freistellungen erforderlich sowie zum Erteilen und Widerrufen des Zugriffsrechts SETSESSIONUSER. Ein Benutzer mit der Berechtigung SECADM kann das Eigentumsrecht von Objekten übertragen, die ihm nicht gehören. Er kann darüber hinaus mit der Anweisung AUDIT einer bestimmten Datenbank bzw. einem bestimmten Datenbankobjekt auf dem Server eine Prüfrichtlinie zuordnen.

    Die Berechtigung SECADM verfügt über kein eigenes Zugriffsrecht zum Zugreifen auf Daten, die in Tabellen gespeichert werden, und verfügt auch über kein anderes eigenes Zugriffsrecht. Es kann nur einem Benutzer mit der Berechtigung SYSADM erteilt werden. Die Berechtigung SECADM kann einem Benutzer, aber keiner Gruppe, Rolle oder PUBLIC erteilt werden.
- Systemsteuerungsberechtigungsstufen, die auf Instanzebene operieren:
  - SYSCTRL (Systemsteuerung)

    Die Berechtigungsstufe SYSCTRL berechtigt zur Steuerung von Operationen, die sich auf Systemressourcen beziehen. Zum Beispiel kann ein Benutzer mit der Berechtigung SYSCTRL eine Datenbank erstellen, aktualisieren, starten, stoppen oder löschen. Ein solcher Benutzer kann darüber hinaus eine Instanz starten oder stoppen, jedoch nicht auf Tabellendaten zugreifen. Benutzer mit der Berechtigung SYSCTRL verfügen auch über die Berechtigung SYSMON.

– SYSMAINT (Systempflege)

  Die Berechtigungsstufe SYSMAINT erteilt die Berechtigung, die zur Ausführung von Pflegeoperationen an allen zu einer Instanz gehörenden Datenbanken erforderlich ist. Ein Benutzer mit der Berechtigung SYSMAINT kann die Datenbankkonfiguration aktualisieren, eine Datenbank oder einen Tabellenbereich sichern, eine vorhandene Datenbank wiederherstellen und eine Datenbank überwachen. Ebenso wie SYSCTRL erteilt SYSMAINT keinen Zugriff auf Tabellendaten. Benutzer mit der Berechtigung SYSMAINT verfügen auch über die Berechtigung SYSMON.

- Berechtigungsebene SYSMON (Systemmonitor)

  SYSMON erteilt die Berechtigung, die zur Verwendung des Datenbanksystemmonitors erforderlich ist. Sie funktioniert auf Instanzebene.

- Datenbankberechtigungen

  Zur Ausführung solcher Aktivitäten wie das Erstellen einer Tabelle oder einer Routine oder das Laden von Daten in eine Tabelle sind bestimmte Datenbankberechtigungen erforderlich. Die Datenbankberechtigung LOAD ist beispielsweise zur Verwendung des Dienstprogramms LOAD zum Laden von Daten in Tabellen erforderlich (ein Benutzer muss auch über das Zugriffsrecht INSERT in der Tabelle verfügen).

In Abb. 1 sind die Abhängigkeiten zwischen Berechtigungen und deren Gültigkeitsbereichen (Datenbank, Datenbankmanager) dargestellt.

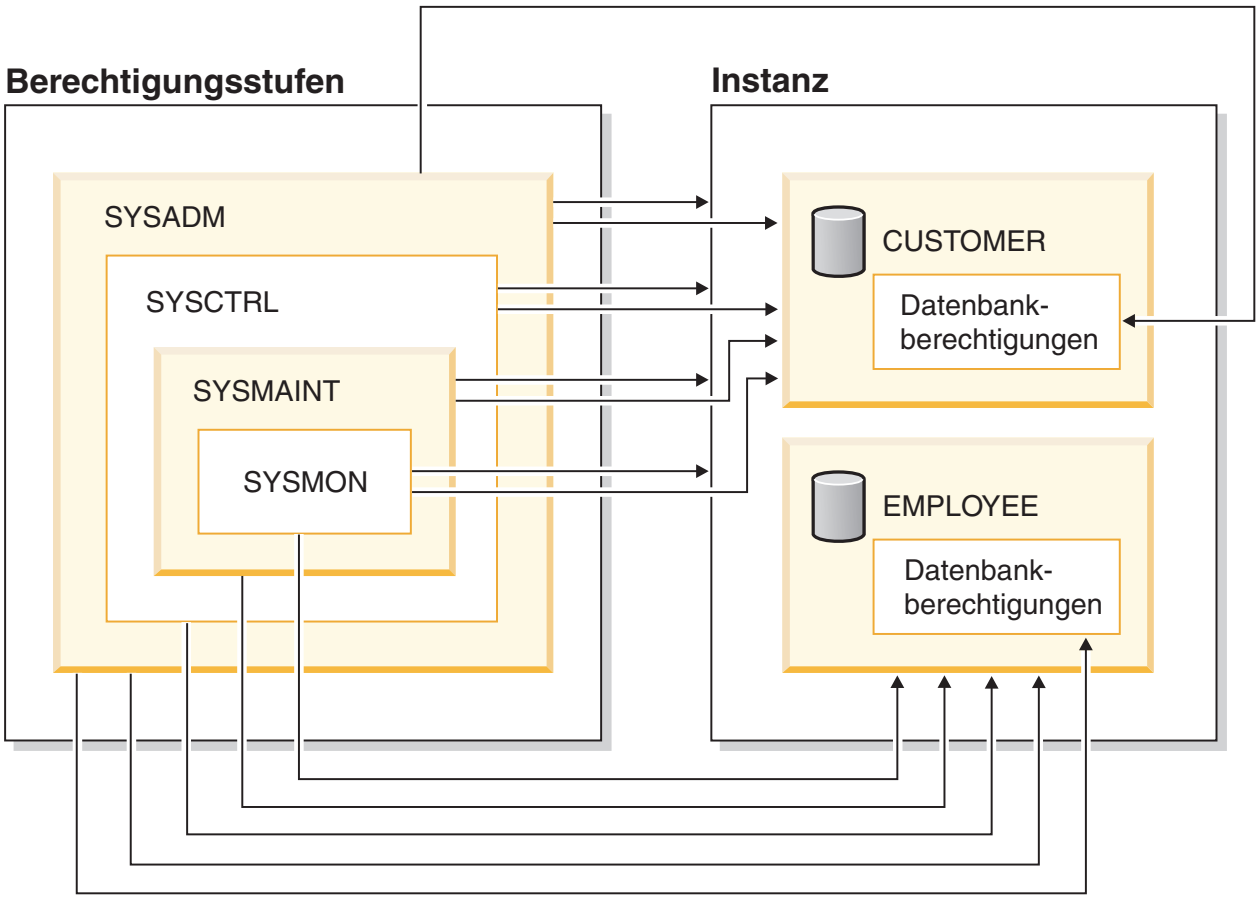


Abbildung 1. Hierarchie der Berechtigungen

Zugriffsrechte

Zugriffsrechte sind die Aktivitäten, die ein Benutzer ausführen darf. Berechtigte Benutzer können Objekte erstellen, auf ihre eigenen Objekte zugreifen und Zugriffsrechte für ihre eigenen Objekte an andere Benutzer mithilfe der Anweisung GRANT weitergeben.

Zugriffsrechte können einzelnen Benutzern, Gruppen oder PUBLIC erteilt werden. PUBLIC ist eine spezielle Gruppe, die aus allen Benutzern, einschließlich zukünftigen Benutzern, besteht. Benutzer, die Mitglieder einer Gruppe sind, können die der Gruppe erteilten Zugriffsrechte indirekt an den Stellen nutzen, an denen Gruppen unterstützt werden.

Zugriffsrecht CONTROL: Der Besitz des Zugriffsrechts CONTROL für ein Objekt berechtigt einen Benutzer zum Zugriff auf dieses Datenbankobjekt sowie zum Erteilen und Widerrufen von Zugriffsrechten für andere Benutzer in Bezug auf dieses Objekt.

Anmerkung: Das Zugriffsrecht CONTROL gilt nur für Tabellen, Sichten, Kurznamen und Pakete.

Wenn ein anderer Benutzer das Zugriffsrecht CONTROL für dieses Objekt benötigt, kann ein Benutzer mit der Berechtigung SYSADM oder DBADM das Zugriffsrecht CONTROL für dieses Objekt erteilen. Das Zugriffsrecht CONTROL kann dem Objekteigner nicht entzogen werden; allerdings kann der Objekteigner mit der Anweisung TRANSFER OWNERSHIP geändert werden.

In einigen Fällen erhält der Ersteller eines Objekts automatisch das Zugriffsrecht CONTROL für das Objekt.

Einzelzugriffsrecht: Einzelzugriffsrechte können erteilt werden, um einem Benutzer die Ausführung bestimmter Tasks für bestimmte Objekte zu erlauben. Benutzer mit einer Administratorberechtigung (SYSADM oder DBADM) oder dem Zugriffsrecht CONTROL können anderen Benutzern Zugriffsrechte erteilen und entziehen.

Einzelne Zugriffsrechte und Datenbankberechtigungen erlauben eine bestimmte Funktion, jedoch beinhalten sie nicht das Recht, die gleichen Zugriffsrechte oder Berechtigungen anderen Benutzern zu erteilen. Das Recht, Zugriffsrechte für Tabellen, Sichten, Schemata, Pakete, Routinen und Sequenzen zu erteilen kann durch die Klausel WITH GRANT OPTION in der Anweisung GRANT auf andere Benutzer übertragen werden. Jedoch erlaubt die Klausel WITH GRANT OPTION der Person, die das Zugriffsrecht erteilt, nicht, das erteilte Zugriffsrecht zu widerrufen. Sie müssen die Berechtigung SYSADM bzw. DBADM oder das Zugriffsrecht CONTROL haben, um das Zugriffsrecht widerrufen zu können.

Zugriffsrechte für Objekte in einem Paket oder einer Routine: Wenn ein Benutzer über das Zugriffsrecht zum Ausführen eines Pakets oder einer Routine verfügt, sind nicht notwendigerweise bestimmte Zugriffsrechte für die Objekte erforderlich, die in dem Paket oder der Routine verwendet werden. Wenn das Paket oder die Routine statische SQL- oder XQuery-Anweisungen enthält, werden die Zugriffsrechte des Eigners des Pakets für diese Anweisungen verwendet. Wenn das Paket oder die Routine dynamische SQL- oder XQuery-Anweisungen enthält, ist die Berechtigungs-ID für die Zugriffsrechtsüberprüfung von der Einstellung der Bindeoption DYNAMICRULES des Pakets abhängig, das die dynamischen Abfrageanweisungen absetzt; außerdem ist sie davon abhängig, ob diese Anweisungen abgesetzt werden, wenn das Paket im Kontext einer Routine verwendet wird.

Ein Benutzer oder eine Gruppe kann für jede Kombination individueller Zugriffsrechte oder Berechtigungen berechtigt werden. Wenn ein Zugriffsrecht einem Objekt zugeordnet wird, muss dieses Objekt vorhanden sein. Zum Beispiel kann einem Benutzer nur dann das Zugriffsrecht SELECT für eine Tabelle erteilt werden, wenn die Tabelle zuvor erstellt wurde.

Anmerkung: Wenn einem Berechtigungsnamen, der einen Benutzer oder eine Gruppe darstellt, Berechtigungen und Zugriffsrechte erteilt werden, und es keinen Benutzer oder keine Gruppe gibt, die mit diesem Namen erstellt wurden, ist Vorsicht geboten. Später kann ein Benutzer oder eine Gruppe mit diesem Namen erstellt werden und automatisch alle diesem Berechtigungsnamen zugeordneten Berechtigungen und Zugriffsrechte erhalten.

Die Anweisung REVOKE dient zum Widerrufen zuvor erteilter Zugriffsrechte. Durch das Widerrufen eines Zugriffsrechts für einen Berechtigungsnamen wird das von allen Berechtigungsnamen erteilte Zugriffsrecht widerrufen.

Durch das Widerrufen eines Zugriffsrechts für einen Berechtigungsnamen wird das gleiche Zugriffsrecht nicht für andere Berechtigungsnamen widerrufen, die das Zugriffsrecht von diesem Berechtigungsnamen erhalten haben. Nehmen Sie zum Beispiel an, dass die Benutzerin CLAIRE dem Benutzer RICK das Zugriffsrecht SELECT WITH GRANT OPTION erteilt, und anschließend der Benutzer RICK das Zugriffsrecht SELECT den Benutzern BOBBY und CHRIS erteilt. Wenn nun die Benutzerin CLAIRE das Zugriffsrecht SELECT für RICK widerruft, behalten die Benutzer BOBBY und CHRIS das Zugriffsrecht SELECT.

LBAC-Berechtigungsnachweise

Mit der kennsatzbasierten Zugriffssteuerung (LBAC - Label-Based Access Control) kann der Sicherheitsadministrator exakt festlegen, wer Schreibzugriff und wer Lesezugriff auf einzelne Zeilen und einzelne Spalten erhält. Der Sicherheitsadministrator konfiguriert das LBAC-System, indem er Sicherheitsrichtlinien erstellt. Eine Sicherheitsrichtlinie (Security Policy) beschreibt die Bedingungen, die bestimmen, wer auf welche Daten Zugriff besitzt. Eine Tabelle kann nur durch eine Sicherheitsrichtlinie geschützt werden, jedoch können verschiedene Tabellen durch verschiedene Sicherheitsrichtlinien geschützt werden.

Nach der Erstellung einer Sicherheitsrichtlinie erstellt der Sicherheitsadministrator Datenbankobjekte, die als Sicherheitskennsätze (Security Labels) und Freistellungen (Exemptions) bezeichnet werden und die Teil der jeweiligen Richtlinie sind. Ein Sicherheitskennsatz beschreibt eine bestimmte Gruppe von Sicherheitsbedingungen. Eine Freistellung bietet die Möglichkeit, dass eine Regel zum Vergleich von Sicherheitskennsätzen für den Benutzer nicht angewendet wird, der die Freistellung besitzt, wenn er auf Daten zugreift, die durch die entsprechende Sicherheitsrichtlinie geschützt werden.

Nach der Erstellung kann ein Sicherheitskennsatz einzelnen Spalten und Zeilen in einer Tabelle zugeordnet werden, um die dort gespeicherten Daten zu schützen. Daten, die durch einen Sicherheitskennsatz geschützt sind, werden als geschützte Daten bezeichnet. Ein Sicherheitsadministrator berechtigt Benutzer zum Zugriff auf geschützte Daten, indem er ihnen Sicherheitskennsätze erteilt. Wenn ein Benutzer versucht, auf geschützte Daten zuzugreifen, wird sein Sicherheitskennsatz mit dem Sicherheitskennsatz verglichen, der die Daten schützt. Der schützende Kennsatz blockiert einige Sicherheitskennsätze und andere nicht.

Objekteigentumsrecht

Wenn ein Objekt erstellt wird, wird einer Berechtigungs-ID das *Eigentumsrecht* an diesem Objekt zugeordnet. Eigentumsrecht bedeutet, dass der Benutzer berechtigt ist, in jeder beliebigen gültigen SQL- oder XQuery-Anweisung auf das Objekt zu verweisen.

Wenn ein Objekt innerhalb eines Schemas erstellt wird, muss die Berechtigungs-ID der Anweisung über das erforderliche Zugriffsrecht verfügen, um Objekte in dem implizit oder explizit angegebenen Schema erstellen zu können. Das bedeutet, der Berechtigungsname muss entweder der Eigner des Schemas sein oder das Zugriffsrecht CREATEIN für das Schema besitzen.

Anmerkung: Diese Voraussetzung gilt nicht für die Erstellung von Tabellenbereichen, Pufferpools oder Datenbankpartitionsgruppen. Solche Objekte werden nicht in Schemata erstellt.

Wenn ein Objekt erstellt wird, ist die Berechtigungs-ID der Anweisung der definierende Benutzer dieses Objekts und wird standardmäßig zum Eigner des Objekts, wenn es erstellt ist.

Anmerkung: Es gibt eine Ausnahme. Wenn die Option AUTHORIZATION in der Anweisung CREATE SCHEMA angegeben wird, ist jedes andere Objekt, das im Rahmen der CREATE SCHEMA-Operation erstellt wird, der Berechtigungs-ID eigen, die in der Option AUTHORIZATION angegeben wird. Alle Objekte, die in dem Schema nach der ersten CREATE SCHEMA-Operation erstellt werden, sind jedoch der Berechtigungs-ID eigen, die der bestimmten CREATE-Anweisung zugeordnet wird.

Zum Beispiel erstellt die Anweisung `CREATE SCHEMA SCOTTSTUFF AUTHORIZATION SCOTT CREATE TABLE T1 (C1 INT)` das Schema `SCOTTSTUFF` und die Tabelle `SCOTTSTUFF.T1`, deren beider Eigner SCOTT ist. Nehmen Sie an, der Benutzer BOBBY erhält das Zugriffsrecht CREATEIN für das Schema `SCOTTSTUFF` und erstellt einen Index für die Tabelle `SCOTTSTUFF.T1`. Da der Index nach dem Schema erstellt wird, ist BOBBY der Eigner des Index für die Tabelle `SCOTTSTUFF.T1`.

Zugriffsrechte werden dem Objekteigner auf der Grundlage des Typs des erstellten Objekts erteilt:

- Das Zugriffsrecht CONTROL wird für neu erstellte Tabellen, Indizes und Pakete implizit erteilt. Dieses Zugriffsrecht berechtigt den Objektersteller zum Zugriff auf das Datenbankobjekt sowie zum Erteilen und Widerrufen von Zugriffsrechten für andere Benutzer in Bezug auf dieses Objekt. Wenn ein anderer Benutzer das Zugriffsrecht CONTROL für dieses Objekt benötigt, muss ein Benutzer mit der Berechtigung SYSADM oder DBADM das Zugriffsrecht CONTROL für dieses Objekt erteilen. Das Zugriffsrecht CONTROL kann nicht durch den Objekteigner entzogen werden.
- Das Zugriffsrecht CONTROL wird implizit für neu erstellte Sichten erteilt, wenn der Objekteigner über das Zugriffsrecht CONTROL für alle Tabellen, Sichten und Kurznamen verfügt, die in der Sichtdefinition angegeben sind.
- Anderen Objekten wie Triggern, Routinen, Sequenzen, Tabellenbereichen und Pufferpools ist kein Zugriffsrecht CONTROL zugeordnet. Der Objekteigner erhält jedoch alle einem Objekt zugeordneten Zugriffsrechte automatisch (und kann diese Zugriffsrechte mithilfe der Klausel WITH GRANT OPTION der Anweisung GRANT anderen Benutzern erteilen, wo dies unterstützt wird). Darüber hinaus kann der Objekteigner das Objekt ändern und löschen oder ihm

einen Kommentar hinzufügen. Diese Berechtigungen gelten für den Objekteigner implizit und können nicht widerrufen werden.

Bestimmte Zugriffsrechte für das Objekt, z. B. zum Ändern einer Tabelle, können durch den Eigner erteilt werden und dem Eigner durch einen Benutzer entzogen werden, der über die Berechtigung SYSADM oder DBADM verfügt. Bestimmte Zugriffsrechte für das Objekt, z. B. zum Kommentieren einer Tabelle, können durch den Eigner nicht erteilt und dem Eigner auch nicht entzogen werden. Verwenden Sie die Anweisung TRANSFER OWNERSHIP, um diese Zugriffsrechte einem anderen Benutzer zu erteilen. Wenn ein Objekt erstellt wird, ist die Berechtigungs-ID der Anweisung der definierende Benutzer dieses Objekts und wird standardmäßig zum Eigner des Objekts, wenn es erstellt ist. Wenn allerdings ein Paket erstellt und die Bindeoption OWNER angegeben wird, entspricht der Eigner der Objekte, die mit den statischen SQL-Anweisungen in dem Paket erstellt werden, dem Wert der Bindeoption OWNER. Außerdem handelt es sich, wenn die Klausel AUTHORIZATION in einer CREATE SCHEMA-Anweisung angegeben wird, bei dem Berechtigungsnamen, der nach dem Schlüsselwort AUTHORIZATION angegeben wird, um den Eigner des Schemas.

Ein Sicherheitsadministrator oder der Objekteigner kann zum Ändern des Eigentumsrechts eines Datenbankobjekts die Anweisung TRANSFER OWNERSHIP verwenden. Daher kann ein Administrator im Namen einer Berechtigungs-ID ein Objekt erstellen, indem das Objekt mit der Berechtigungs-ID als Qualifikationsmerkmal erstellt und anschließend die Anweisung TRANSFER OWNERSHIP zum Übertragen des Eigentumsrechts, über das der Administrator für das Objekt verfügt, an die Berechtigungs-ID verwendet wird.

Berechtigungs-IDs in verschiedenen Kontexten

Eine Berechtigungs-ID dient zwei Zwecken: Identifikation und Berechtigungsprüfung. Zum Beispiel wird die Sitzungsberechtigungs-ID für die einleitende Berechtigungsprüfung verwendet.

Wenn auf die Verwendung einer Berechtigungs-ID in einem bestimmten Kontext Bezug genommen wird, ist der Verweis auf die Berechtigung dazu geeignet, den Kontext zu identifizieren, wie nachfolgend gezeigt.

Kontextbezogener Verweis auf die Berechtigungs-ID
Definition

Systemberechtigungs-ID
Die Berechtigungs-ID, die für jede einleitende Berechtigungsprüfung verwendet wird, zum Beispiel zum Überprüfen des Zugriffsrechts CONNECT bei der CONNECT-Verarbeitung. Als Teil des Authentifizierungsprozesses während der CONNECT-Verarbeitung wird eine Berechtigungs-ID generiert, die mit den DB2-Namenskonventionen kompatibel ist und die externe Benutzer-ID innerhalb des DB2-Datenbanksystems repräsentiert. Die Systemberechtigungs-ID stellt den Benutzer dar, der die Verbindung hergestellt hat. Mithilfe des Sonderregisters SYSTEM_USER können Sie den aktuellen Wert der Systemberechtigungs-ID ermitteln. Die Systemberechtigungs-ID kann für eine Verbindung nicht geändert werden.

Sitzungsberechtigungs-ID
Die Berechtigungs-ID, die für jede Sitzungsberechtigungsprüfung nach den einleitenden Überprüfungen der CONNECT-Verarbeitung verwendet wird. Der Standardwert der Sitzungsberechtigungs-ID ist der Wert der Systemberechtigungs-ID. Mithilfe des Sonderregisters SESSION_USER können Sie den aktuellen Wert der Sitzungsberechtigungs-ID feststellen. Das Sonder-

register USER ist ein Synonym für das Sonderregister SESSION_USER. Die Sitzungsberechtigungs-ID kann mithilfe der Anweisung SET SESSION AUTHORIZATION geändert werden.

Paketberechtigungs-ID
Die Berechtigungs-ID, die zum Binden (BIND) eines Pakets an die Datenbank verwendet wird. Diese Berechtigungs-ID wird aus dem Wert der Bindeoption OWNER ermittelt. Die Paketberechtigungs-ID wird manchmal als Paketbinder oder Paketeigner bezeichnet.

Routineneignerberechtigungs-ID
Die Berechtigungs-ID, die in den Systemkatalogen als Eigner der SQL-Routine eingetragen ist, die aufgerufen wurde.

Routinenaufruferberechtigungs-ID
Die Berechtigungs-ID, die die Anweisungsberechtigungs-ID für die Anweisung ist, die eine SQL-Routine aufgerufen hat.

Anweisungsberechtigungs-ID
Die Berechtigungs-ID, die einer bestimmten SQL-Anweisung zugeordnet ist und die für alle Berechtigungsanforderungen sowie zur Bestimmung des Objekteigentumsrechts (sofern zutreffend) zu verwenden ist. Ihr Wert leitet sich abhängig vom Typ der SQL-Anweisung aus der entsprechenden Quellenberechtigungs-ID ab:

- Statisches SQL

  Die Paketberechtigungs-ID wird verwendet.
- Dynamisches SQL (aus Nichtroutinenkontext)

  Die Tabelle zeigt, welche Berechtigungs-ID in den einzelnen Fällen verwendet wird:

Wert der Option DYNAMICRULES zum Ausführen des Pakets	**Verwendete Berechtigungs-ID**
RUN	Sitzungsberechtigungs-ID
BIND	Paketberechtigungs-ID
DEFINERUN, INVOKERUN	Sitzungsberechtigungs-ID
DEFINEBIND, INVOKEBIND	Paketberechtigungs-ID

- Dynamisches SQL (aus Routinenkontext)

  Die Tabelle zeigt, welche Berechtigungs-ID in den einzelnen Fällen verwendet wird:

Wert der Option DYNAMICRULES zum Ausführen des Pakets	**Verwendete Berechtigungs-ID**
DEFINERUN, DEFINEBIND	Routineneignerberechtigungs-ID
INVOKERUN, INVOKEBIND	Routinenaufruferberechtigungs-ID

Mithilfe des Sonderregisters CURRENT_USER können Sie den aktuellen Wert der Anweisungsberechtigungs-ID ermitteln. Die Anweisungsberechtigungs-ID kann nicht direkt geändert werden. Sie wird vom DB2-Datenbanksystem abhängig vom Typ der jeweiligen SQL-Anweisung automatisch geändert.

Berechtigungen der Instanzebene

Systemverwaltungsberechtigung (SYSADM)

Die Berechtigung SYSADM ist die höchste Stufe der Administratorberechtigung. Benutzer mit der Berechtigung SYSADM können Dienstprogramme ausführen, Befehle der Datenbank und des Datenbankmanagers ausführen und auf die Daten aller Tabellen in allen Datenbanken innerhalb der Datenbankmanagerinstanz zugreifen, die nicht LBAC-geschützt sind. Sie erteilt die Berechtigung, alle Datenbankobjekte in der Instanz zu steuern, einschließlich Datenbanken, Tabellen, Sichten, Indizes, Paketen, Schemata, Servern, Aliasnamen, Datentypen, Funktionen, Prozeduren, Triggern, Tabellenbereichen, Datenbankpartitionsgruppen, Pufferpools und Ereignismonitoren.

Die Berechtigung SYSADM wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysadm_group* angegeben ist. Die Zugehörigkeit zu dieser Gruppe wird außerhalb des Datenbankmanagers über die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer mit der Berechtigung SYSADM kann folgende Funktionen ausführen:

- Migrieren einer Datenbank
- Ändern der Konfigurationsdatei des Datenbankmanagers (einschließlich Angabe der Gruppen, die über die Berechtigung SYSCTRL, SYSMAINT oder SYSMON verfügen)
- Erteilen und Entziehen der Berechtigung DBADM
- Erteilen und Entziehen der Berechtigung SECADM

Zwar hat man mit der Berechtigung SYSADM alle Möglichkeiten, die auch mit den meisten anderen Berechtigungen möglich sind, allerdings stellt die Berechtigung SECADM weitere Möglichkeiten zur Verfügung, die mit der Berechtigung SYSADM nicht möglich sind. Die Möglichkeiten der Berechtigung SECADM stehen mit keiner anderen Berechtigung zur Verfügung. Mit der Berechtigung SYSADM können Sie auch nicht auf LBAC-geschützte Daten zugreifen.

Anmerkung: Wenn Benutzer mit der Berechtigung SYSADM eine Datenbank erstellt, wird diesem Benutzer automatisch die explizite Berechtigung DBADM für die Datenbank erteilt. Wenn der Datenbankersteller aus der SYSADM-Gruppe entfernt wird und Sie diesen Benutzer auch daran hindern möchten, mit der Berechtigung DBADM auf die Datenbank zuzugreifen, müssen Sie die Berechtigung DBADM des Benutzers explizit widerrufen.

Systemsteuerungsberechtigung (SYSCTRL)

Die Berechtigung SYSCTRL ist die höchste Stufe der Systemsteuerungsberechtigung. Diese Berechtigung bietet die Möglichkeit, Pflege- und Dienstprogrammoperationen für die Datenbankmanagerinstanz und ihre Datenbanken auszuführen. Diese Operationen können die Systemressourcen beeinflussen, aber sie ermöglichen keinen direkten Zugriff auf Daten in den Datenbanken. Die Systemsteuerungsberechtigung ist für Benutzer gedacht, die eine Instanz des Datenbankmanagers mit sensiblen Daten verwalten.

Die Berechtigung SYSCTRL wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysctrl_group* angegeben ist. Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer, der mindestens über die Berechtigung SYSCTRL verfügt, kann folgende Operationen ausführen:

- Aktualisieren einer Datenbank, eines Knotens oder eines DCS-Verzeichnisses (Distributed Connection Services)
- Benutzer zwangsweise aus dem System entfernen
- Erstellen oder Löschen einer Datenbank
- Löschen, Erstellen oder Ändern eines Tabellenbereichs
- Restore in eine neue Datenbank

Zusätzlich kann ein Benutzer mit der Berechtigung SYSCTRL die Funktionen von Benutzern mit Systempflegeberechtigung (SYSMAINT) und Systemmonitorberechtigung (SYSMON) ausführen.

Benutzer mit der Berechtigung SYSCTRL verfügen auch über das implizite Zugriffsrecht, eine Verbindung zu einer Datenbank herzustellen.

Anmerkung: Wenn Benutzer mit der Berechtigung SYSCTRL Datenbanken erstellen, wird ihnen automatisch die explizite Berechtigung DBADM für die Datenbank erteilt. Wenn der Datenbankersteller aus der SYSCTRL-Gruppe gelöscht wird und Sie ihn auch daran hindern möchten, mit der Berechtigung DBADM auf die Datenbank zuzugreifen, müssen Sie diese Berechtigung DBADM explizit widerrufen.

Systempflegeberechtigung (SYSMAINT)

Die Berechtigung SYSMAINT ist die zweite Stufe der Systemsteuerungsberechtigung. Diese Berechtigung bietet die Möglichkeit, Pflege- und Dienstprogrammoperationen für die Datenbankmanagerinstanz und ihre Datenbanken auszuführen. Diese Operationen können die Systemressourcen beeinflussen, aber sie ermöglichen keinen direkten Zugriff auf Daten in den Datenbanken. Die Systempflegeberechtigung ist für Benutzer konzipiert, die Datenbanken innerhalb einer Datenbankmanagerinstanz pflegen, die sensible Daten enthält.

Die Berechtigung SYSMAINT wird der Gruppe zugeordnet, die im Konfigurationsparameter *sysmaint_group* angegeben ist. Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Nur ein Benutzer mit der Berechtigung SYSMAINT oder einer höheren Systemberechtigung kann folgende Aktionen ausführen:

- Aktualisieren der Datenbankkonfigurationsdateien
- Backup einer Datenbank oder eines Tabellenbereichs
- Restore in eine existierende Datenbank
- Ausführen einer aktualisierenden Recovery
- Starten oder Stoppen einer Instanz
- Restore eines Tabellenbereichs
- Ausführen eines Trace
- Erstellen von Momentaufnahmen einer Datenbankmanagerinstanz oder ihrer Datenbanken mithilfe des Datenbanksystemmonitors.

Ein Benutzer mit der Berechtigung SYSMAINT, DBADM oder einer höheren Berechtigung kann Folgendes ausführen:

- Abfragen des Status eines Tabellenbereichs
- Aktualisieren von Protokolldateien

- Versetzen eines Tabellenbereichs in den Quiescemodus
- Neuorganisieren einer Tabelle
- Sammeln von Katalogstatistiken mit dem Dienstprogramm RUNSTATS

Benutzer mit der Berechtigung SYSMAINT verfügen auch über das implizite Zugriffsrecht, eine Verbindung zu einer Datenbank herzustellen, und sie können die Funktionen von Benutzern mit Systemmonitorberechtigung (SYSMON) ausführen.

Systemmonitorberechtigung (SYSMON)

Die Berechtigung SYSMON ermöglicht die Verwendung des Datenbanksystemmonitors zur Erstellung von Momentaufnahmen (Snapshots) einer Datenbankmanagerinstanz oder seiner Datenbanken. Die Berechtigung SYSMON wird der Gruppe zugeordnet, die im Konfigurationsparameter **sysmon_group** angegeben ist. Wenn eine Gruppe angegeben wird, wird die Zugehörigkeit zu dieser Gruppe außerhalb des Datenbankmanagers durch die auf Ihrer Plattform verwendete Sicherheitseinrichtung gesteuert.

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden Befehle auszuführen:
- GET DATABASE MANAGER MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET SNAPSHOT
- LIST ACTIVE DATABASES
- LIST APPLICATIONS
- LIST DCS APPLICATIONS
- RESET MONITOR
- UPDATE MONITOR SWITCHES

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden APIs zu verwenden:
- db2GetSnapshot - Momentaufnahme erstellen
- db2GetSnapshotSize - Größe des erforderlichen Ausgabepuffers für db2GetSnapshot() schätzen
- db2MonitorSwitches - Monitorschalter abrufen/aktualisieren
- db2ResetMonitor - Monitor zurücksetzen

Die Berechtigung SYSMON gibt dem Benutzer die Möglichkeit, die folgenden SQL-Tabellenfunktionen zu verwenden:

- Alle Tabellenfunktionen zur Erstellung von Momentaufnahmen, ohne zuvor SYSPROC.SNAP_WRITE_FILE auszuführen.

  SYSPROC.SNAP_WRITE_FILE erfasst eine Momentaufnahme und speichert ihren Inhalt in einer Datei. Wenn eine der Tabellenfunktionen für Momentaufnahmen mit null Eingabeparametern aufgerufen wird, wird der Inhalt der Datei anstelle einer Echtzeitmomentaufnahme des Systems zurückgegeben.

Berechtigungen der Datenbankebene

Sicherheitsverwaltungsberechtigung (SECADM)

Die Berechtigung SECADM (Sicherheitsadministrator) ist die Berechtigung, die zum Erstellen, Ändern (sofern zutreffend) und Löschen von Rollen, gesicherten Kontexten, Prüfrichtlinien, Sicherheitskennsatzkomponenten, Sicherheitsrichtlinien und Sicherheitskennsätzen erforderlich ist. Diese Berechtigung ist auch zum Erteilen und Widerrufen von Rollen, Sicherheitskennsätzen und Freistellungen sowie

des Zugriffsrechts SETSESSIONUSER erforderlich. Die Berechtigung SECADM beinhaltet kein Zugriffsrecht für den Zugriff auf Daten, die in Tabellen gespeichert werden.

Die Berechtigung SECADM kann durch den Systemadministrator (mit der Berechtigung SYSADM) erteilt und nur einem Benutzer, jedoch nicht einer Gruppe oder einer Rolle erteilt werden. Sie berechtigt ausschließlich zu den folgenden Aktionen:

- Erstellen, Ändern, Kommentieren und Löschen folgender Elemente:
  - Prüfrichtlinien
  - Sicherheitkennsatzkomponenten
  - Sicherheitsrichtlinien
  - Gesicherte Kontexte
- Erstellen, Kommentieren und Löschen folender Elemente:
  - Rollen
  - Sicherheitskennsätze
- Erteilen und Entziehen folgender Elemente:
  - Rollen
  - Freistellungen
  - Sicherheitskennsätze
  - SETSESSIONUSER-Zugriffsrechte
- Verwenden der gespeicherten Prozeduren und der Tabellenfunktion für das Prüfsystem: SYSPROC.AUDIT_ARCHIVE, SYSPROC.AUDIT_LIST_LOGS und SYSPROC.AUDIT_DELIM_EXTRACT. Diese können nur durch den Sicherheitsadministrator aufgerufen werden.
- Verwenden der Anweisung AUDIT, um einer bestimmten Datenbank bzw. einem Datenbankobjekt auf dem Server eine Prüfrichtlinie zuzuordnen.
- Ausführen der SQL-Anweisung TRANSFER OWNERSHIP an Objekten, deren Eigner nicht die Berechtigungs-ID der Anweisung ist.

Keine andere Berechtigung berechtigt zu diesen Aktionen, nicht einmal SYSADM.

Der Instanzeigner verfügt standardmäßig nicht über die Sicherheitsverwaltungsberechtigung (SECADM). Der Systemadministrator (SYSADM) kann die Berechtigung SECADM anderen Benutzern mit Grant erteilen. Allerdings ist der Systemadministrator nicht in der Lage, sich selbst die Berechtigung SECADM zu erteilen. Jedes Mitglied der Gruppe SYSADM_GROUP besitzt die Systemverwaltungsberechtigung (SYSADM) und kann daher die Berechtigung SECADM einem beliebigen Benutzer seiner Wahl erteilen.

Datenbankadministratorberechtigung (DBADM)

Die Berechtigung DBADM ist eine Administratorberechtigung für eine bestimmte Datenbank; sie ermöglicht dem Benutzer die Durchführung bestimmter Aktionen und das Absetzen von Datenbankbefehlen für diese Datenbank. Mit der Berechtigung DBADM ist ein Zugreifen auf die Daten in beliebigen Tabellen der Datenbank möglich, solange die Daten durch LBAC geschützt sind. Sie müssen über die entsprechenden LBAC-Berechtigungsnachweise verfügen, um auf LBAC-geschützte Daten zugreifen zu können.

Wenn die Berechtigung DBADM erteilt wurde, werden auch die folgenden Datenbankberechtigungen explizit für dieselbe Datenbank erteilt (und auch nicht automatisch widerrufen, wenn die Berechtigung DBADM später widerrufen wird):

- BINDADD

- CONNECT
- CREATETAB
- CREATE_EXTERNAL_ROUTINE
- CREATE_NOT_FENCED_ROUTINE
- IMPLICIT_SCHEMA
- QUIESCE_CONNECT
- LOAD

Nur ein Benutzer mit der Berechtigung SYSADM kann die Berechtigung DBADM erteilen oder widerrufen. Benutzer mit der Berechtigung DBADM können anderen Benutzern Zugriffsrechte für die Datenbank erteilen und entziehen, unabhängig davon, wer diese Berechtigungen erteilt hat.

Mit der Berechtigung DBADM oder einer umfassenderen Berechtigung für eine Datenbank kann ein Benutzer die folgenden Aktionen für diese Datenbank durchführen:

- Lesen von Protokolldateien
- Erstellen, Aktivieren und Löschen von Ereignismonitoren

Ein Benutzer mit der Berechtigung DBADM für eine Datenbank, mit der Berechtigung SYSMAINT oder einer höheren Berechtigung kann die folgenden Aktionen für die Datenbank durchführen:

- Abfragen des Status eines Tabellenbereichs
- Aktualisieren von Protokolldateien
- Versetzen eines Tabellenbereichs in den Quiescemodus
- Neuorganisieren einer Tabelle
- Sammeln von Katalogstatistiken mit dem Dienstprogramm RUNSTATS

Zwar hat man mit der Berechtigung DBADM einige Möglichkeiten, die auch mit den anderen Berechtigungen möglich sind, allerdings stellt die Berechtigung SECADM noch weitere Möglichkeiten zur Verfügung, die mit der Berechtigung DBADM nicht möglich sind. Die Möglichkeiten der Berechtigung SECADM stehen mit keiner anderen Berechtigung zur Verfügung.

Berechtigung LOAD

Benutzer, die über LOAD-Berechtigung auf Datenbankebene sowie über das Zugriffsrecht INSERT für eine Tabelle verfügen, können den Befehl LOAD verwenden, um Daten in eine Tabelle zu laden.

Benutzer, die über die Berechtigung LOAD auf Datenbankebene sowie über das Zugriffsrecht INSERT auf eine Tabelle verfügen, können den Befehl LOAD RESTART oder LOAD TERMINATE verwenden, wenn bei der vorangegangenen Ladeoperation Daten eingefügt (LOAD INSERT) wurden.

Benutzer, die über LOAD-Berechtigung auf Datenbankebene sowie über die Zugriffsrechte INSERT und DELETE für eine Tabelle verfügen, können den Befehl LOAD REPLACE verwenden.

Wenn bei der vorangegangenen Ladeoperation Daten ersetzt (LOAD REPLACE) wurden, muss diesem Benutzer auch das Zugriffsrecht DELETE erteilt werden, damit dieser den Befehl LOAD RESTART oder LOAD TERMINATE verwenden kann.

Wenn die Ausnahmetabellen im Rahmen der LOAD-Operation verwendet werden, muss der Benutzer über das Zugriffsrechte INSERT für die Ausnahmetabellen verfügen.

Der Benutzer mit dieser Berechtigung kann die Befehle QUIESCE TABLESPACES FOR TABLE, RUNSTATS und LIST TABLESPACES ausführen.

Datenbankberechtigungen

Jede Datenbankberechtigung verleiht der Berechtigungs-ID, der sie erteilt ist, das Recht, einen bestimmten Typ von Aktion an der Datenbank als Ganzer auszuführen. Datenbankberechtigungen unterscheiden sich von Zugriffsrechten, die zur Ausführung einer bestimmten Aktion an einem bestimmten Datenbankobjekt, zum Beispiel einer Tabelle oder einem Index, berechtigen.

Die folgenden Datenbankberechtigungen sind verfügbar:

SECADM
Gibt dem Inhaber die Möglichkeit, als Sicherheitsadministrator zu fungieren und Sicherheitsobjekte zu erstellen und löschen sowie Berechtigungen und Zugriffsrechte für Sicherheitsobjekte zu erteilen und zu widerrufen und Eigentumsrechte an Objekten zu übertragen. Der Sicherheitsadministrator verwaltet gesicherte Kontexte, Prüfrichtlinien, Datenbankrollen und den Datenschutz durch die kennsatzbasierte Zugriffssteuerung (LBAC).

DBADM
Erteilt dem Inhaber die Berechtigung, als Datenbankadministrator zu fungieren. Insbesondere erteilt diese Berechtigung dem Inhaber alle anderen Datenbankberechtigungen mit Ausnahme von SECADM.

CONNECT
Berechtigt den Inhaber zur Herstellung einer Verbindung zur Datenbank.

BINDADD
Berechtigt den Inhaber zur Erstellung neuer Pakete in der Datenbank.

CREATETAB
Berechtigt den Inhaber zur Erstellung neuer Tabellen in der Datenbank.

CREATE_EXTERNAL_ROUTINE
Berechtigt den Inhaber zur Erstellung einer Prozedur zur Verwendung durch Anwendungen und andere Benutzer der Datenbank.

CREATE_NOT_FENCED_ROUTINE
Berechtigt den Inhaber zur Erstellung einer benutzerdefinierten Funktion (UDF) oder Prozedur, die nicht abgeschirmt („Not Fenced") ist. CREATE_EXTERNAL_ROUTINE wird jedem Benutzer automatisch erteilt, dem CREATE_NOT_FENCED_ROUTINE erteilt wird.

Achtung: Der Datenbankmanager schützt seinen Speicher und seine Steuerblöcke nicht vor UDFs oder Prozeduren, die „nicht abgeschirmt" sind. Ein Benutzer mit dieser Berechtigung muss daher eine UDF mit äußerster Sorgfalt testen, bevor er sie als „nicht abgeschirmt" registriert.

IMPLICIT_SCHEMA
Berechtigt jeden Benutzer zur impliziten Erstellung eines Schemas, indem er ein Objekt mit der Anweisung CREATE und einem Schemanamen, der noch nicht existiert, erstellt. SYSIBM wird zum Eigner des implizit erstellten Schemas, und die Gruppe PUBLIC erhält das Zugriffsrecht zur Erstellung von Objekten in diesem Schema.

LOAD
Berechtigt den Inhaber zum Laden von Daten in eine Tabelle.

QUIESCE_CONNECT
Berechtigt den Inhaber zum Zugriff auf die Datenbank, während sie im Quiescemodus ist.

Nur Berechtigungs-IDs mit der Berechtigung SYSADM können die Berechtigungen SECADM und DBADM erteilen. Alle anderen Berechtigungen können durch Berechtigungs-IDs erteilt werden, die über die Berechtigungen SYSADM oder DBADM verfügen.

Wenn eine Datenbank erstellt wird, werden der Gruppe PUBLIC automatisch die folgenden Datenbankberechtigungen für die neue Datenbank erteilt:

- CREATETAB
- BINDADD
- CONNECT
- IMPLICIT_SCHEMA

Zusätzlich werden die folgenden Zugriffsrechte erteilt:

- Zugriffsrecht USE für den Tabellenbereich USERSPACE1
- Zugriffsrecht SELECT für die Systemkatalogsichten

Wenn der Gruppe PUBLIC eine Datenbankberechtigung entzogen werden soll, muss eine Berechtigungs-ID mit der Berechtigung DBADM oder SYSADM diese explizit widerrufen (REVOKE).

Hinweise zur Berechtigung IMPLICIT_SCHEMA

Wenn eine neue Datenbank erstellt wird, erhält die Gruppe PUBLIC die Datenbankberechtigung IMPLICIT_SCHEMA. Mit dieser Berechtigung kann jeder Benutzer ein Schema erstellen, indem er ein Objekt erstellt und dabei einen Schemanamen angibt, der noch nicht existiert. SYSIBM wird zum Eigner des implizit erstellten Schemas, und die Gruppe PUBLIC erhält das Zugriffsrecht zur Erstellung von Objekten in diesem Schema.

Wenn für die Datenbank eine Kontrolle darüber erforderlich ist, wer implizit Schemaobjekte erstellen kann, sollte die Berechtigung IMPLICIT_SCHEMA der Gruppe PUBLIC entzogen werden. Wenn dies geschehen ist, gibt es nur drei Möglichkeiten zur Erstellung eines Schemenobjekts:

- Jeder Benutzer kann ein Schema erstellen, indem er seinen eigenen Berechtigungsnamen in der Anweisung CREATE SCHEMA verwendet.
- Jeder Benutzer mit der Berechtigung DBADM kann ein beliebiges, noch nicht vorhandenes Schema explizit erstellen und optional einen anderen Benutzer als Eigner des Schemas angeben.
- Jeder Benutzer mit der Berechtigung DBADM hat die Berechtigung IMPLICIT_SCHEMA (unabhängig von PUBLIC), sodass er implizit ein Schema mit einem beliebigen Namen bei der Erstellung anderer Datenbankobjekte erstellen kann. SYSIBM wird zum Eigner des implizit erstellten Schemas, und die Gruppe PUBLIC erhält das Zugriffsrecht zur Erstellung von Objekten in dem Schema.

Zugriffsrechte

Zugriffsrechte der Berechtigungs-IDs

Zugriffsrechte der Berechtigungs-IDs beinhalten Aktionen, die an Berechtigungs-IDs ausgeführt werden können. Gegenwärtig ist nur ein solches Zugriffsrecht vorhanden: SETSESSIONUSER.

Das Zugriffsrecht SETSESSIONUSER kann einem Benutzer oder einer Gruppe erteilt werden und gibt seinem Besitzer die Möglichkeit, seine Identität in eine beliebige der Berechtigungs-IDs zu wechseln, für die das Zugriffsrecht erteilt wurde. Der Identitätswechsel wird mithilfe der SQL-Anweisung SET SESSION AUTHORIZATION durchgeführt. Das Zugriffsrecht SETSESSIONUSER kann nur einem Benutzer mit der Berechtigung SECADM erteilt werden.

Anmerkung: Wenn Sie eine Datenbank der Version 8 auf Version 9.1 (oder eine spätere Version) migrieren, wird den Berechtigungs-IDs mit expliziter DBADM-Berechtigung für diese Datenbank automatisch das Zugriffsrecht SETSESSIONUSER für die Gruppe PUBLIC erteilt. Dadurch wird verhindert, dass Anwendungen fehlschlagen, die davon abhängig sind, dass Berechtigungs-IDs mit DBADM-Berechtigung die Sitzungsberechtigungs-ID auf eine beliebige Berechtigungs-ID setzen zu können. Dies geschieht nicht, wenn die Berechtigungs-ID die Berechtigung SYSADM besitzt, ihr jedoch die Berechtigung DBADM nicht explizit erteilt wurde.

Schemazugriffsrechte

Zugriffsrechte für Schemata gehören zur Kategorie der Zugriffsrechte für Objekte. Die Zugriffsrechte für Objekte sind in Abb. 2 auf Seite 37 dargestellt.

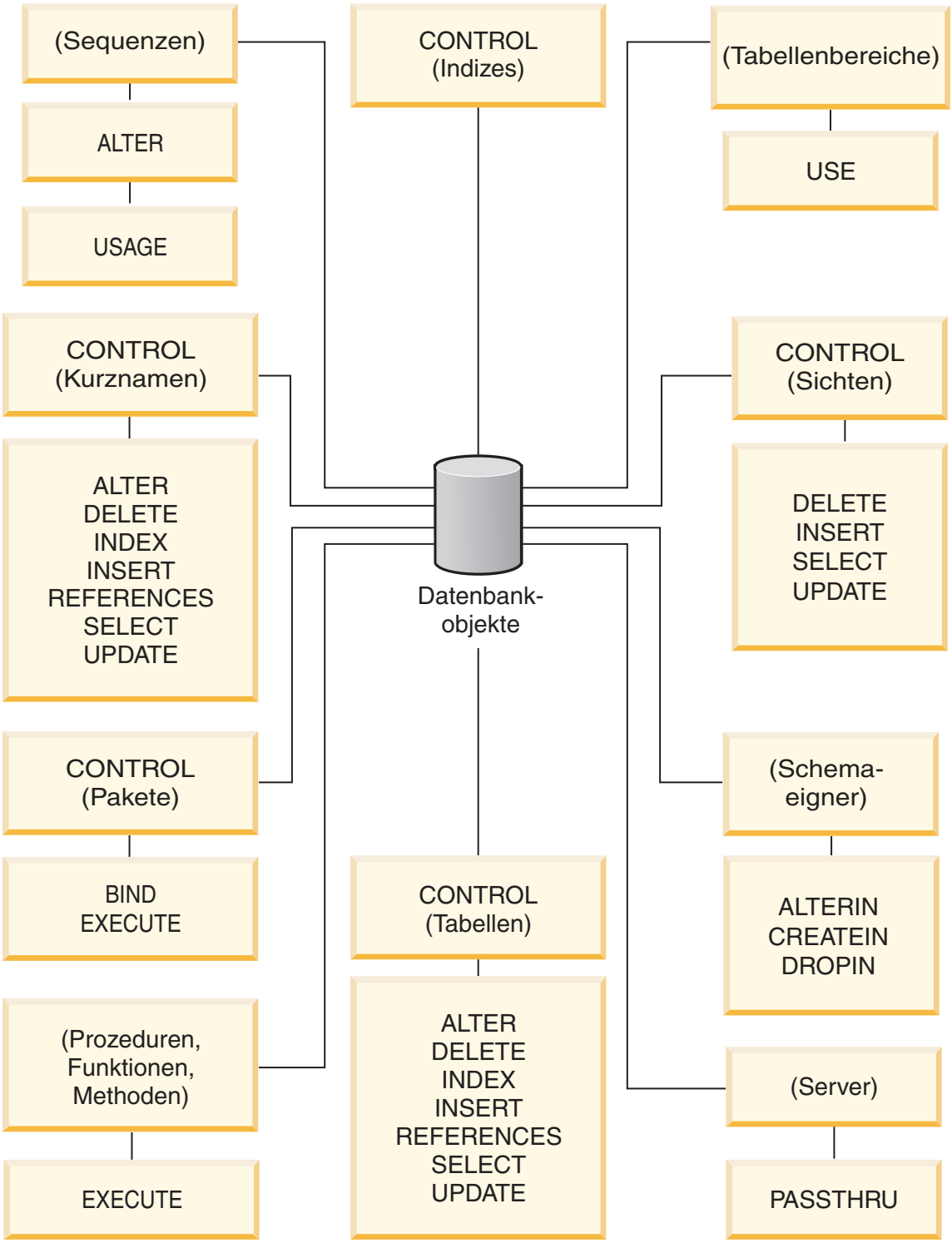


Abbildung 2. Zugriffsrechte für Objekte

Zugriffsrechte für Schemata berechtigen zu Aktionen an Schemata in einer Datenbank. Einem Benutzer kann jedes der folgenden Zugriffsrechte erteilt werden:

- CREATEIN erlaubt dem Benutzer, Objekte innerhalb des Schemas zu erstellen.
- ALTERIN erlaubt dem Benutzer, Objekte innerhalb des Schemas zu ändern.
- DROPIN erlaubt dem Benutzer, Objekte aus dem Schema zu löschen.

Der Eigner des Schemas besitzt alle diese Zugriffsrechte und die Möglichkeit, diese Zugriffsrechte anderen Benutzern zu erteilen. Zu den Objekten, die innerhalb des Schemenobjekts bearbeitet werden, gehören: Tabellen, Sichten, Indizes, Pakete, Datentypen, Funktionen, Trigger, Prozeduren und Aliasnamen.

Zugriffsrechte für Tabellenbereiche

Mit den Zugriffsrechten für Tabellenbereiche können Aktionen für die Tabellenbereiche einer Datenbank ausgeführt werden. Wenn einem Benutzer das Zugriffsrecht USE für einen Tabellenbereich zugeordnet wird, ist er zum Erstellen von Tabellen innerhalb des Tabellenbereichs berechtigt.

Der Eigner des Tabellenbereichs (normalerweise der Ersteller mit der Berechtigung SYSADM oder SYSCTRL) verfügt über das Zugriffsrecht USE und die Möglichkeit, dieses Zugriffsrecht auch anderen Benutzern zu erteilen. Standardmäßig wird beim Erstellen der Datenbank das Zugriffsrecht USE für den Tabellenbereich USERSPACE1 der Gruppe PUBLIC erteilt. Dieses Zugriffsrecht kann jedoch auch widerrufen werden.

Das Zugriffsrecht USE kann bei SYSCATSPACE oder anderen temporären Systemtabellenbereichen nicht verwendet werden.

Zugriffsrechte für Tabellen und Sichten

Zugriffsrechte für Tabellen und Sichten umfassen Aktionen für Tabellen oder Sichten in einer Datenbank.

Ein Benutzer muss über die Berechtigung CONNECT für die Datenbank verfügen, um eines der folgenden Zugriffsrechte verwenden zu können:

- CONTROL erteilt dem Benutzer alle Zugriffsrechte für eine Tabelle oder Sicht, einschließlich der Erlaubnis, sie zu löschen und einzelne Zugriffsrechte für Tabellen zu erteilen oder zu widerrufen. Zur Erteilung des Zugriffsrechts CONTROL müssen Sie über die Berechtigung SYSADM oder DBADM verfügen. Der Ersteller einer Tabelle erhält automatisch das Zugriffsrecht CONTROL für die Tabelle. Der Ersteller einer Sicht erhält nur dann automatisch das Zugriffsrecht CONTROL, wenn er über das Zugriffsrecht CONTROL für alle Tabellen, Sichten und Kurznamen verfügt, auf die in der Sichtdefinition verwiesen wird, oder wenn er über die Berechtigung SYSADM oder DBADM verfügt.
- Das Zugriffsrecht ALTER ermöglicht dem Benutzer, eine Tabelle zu modifizieren, zum Beispiel, der Tabelle Spalten oder eine eindeutige Integritätsbedingung hinzuzufügen. Ein Benutzer mit dem Zugriffsrecht ALTER kann außerdem die Anweisung COMMENT ON für eine Tabelle bzw. für Spalten der Tabelle angeben. Informationen zu den möglichen Modifikationen, die an einer Tabelle vorgenommen werden können, finden Sie in den Beschreibungen der Anweisungen ALTER TABLE und COMMENT.
- DELETE erlaubt dem Benutzer das Löschen von Zeilen aus einer Tabelle oder Sicht.
- INDEX erlaubt dem Benutzer das Erstellen eines Index für eine Tabelle. Die Ersteller von Indizes verfügen automatisch über das Zugriffsrecht CONTROL für den Index.
- INSERT erlaubt dem Benutzer das Einfügen einer Zeile in eine Tabelle oder Sicht und das Ausführen des Dienstprogramms IMPORT.
- REFERENCES erlaubt dem Benutzer das Erstellen und Löschen eines Fremdschlüssels unter Angabe der Tabelle als die übergeordnete Tabelle in einer Abhängigkeitsbeziehung. Der Benutzer hat dieses Zugriffsrecht möglicherweise nur für bestimmte Spalten.

- SELECT erlaubt dem Benutzer das Abrufen von Zeilen aus einer Tabelle oder Sicht, das Erstellen einer Sicht für eine Tabelle und das Ausführen des Dienstprogramms EXPORT.
- UPDATE erlaubt dem Benutzer das Ändern eines Eintrags in einer Tabelle oder Sicht bzw. für eine oder mehrere Spalten in einer Tabelle oder Sicht. Der Benutzer hat dieses Zugriffsrecht eventuell nur für bestimmte Spalten.

Das Zugriffsrecht zum Erteilen dieser Zugriffsrechte an andere Benutzer kann auch mithilfe der Klausel WITH GRANT OPTION in der Anweisung GRANT erteilt werden.

Anmerkung: Wenn einem Benutzer oder einer Gruppe das Zugriffsrecht CONTROL für eine Tabelle erteilt wird, werden alle anderen Zugriffsrechte für diese Tabelle automatisch mit der Klausel WITH GRANT OPTION erteilt. Wenn Sie anschließend das Zugriffsrecht CONTROL eines Benutzers für die Tabelle widerrufen, behält der Benutzer weiterhin die anderen Zugriffsrechte, die ihm automatisch erteilt wurden. Wenn Sie alle Zugriffsrechte widerrufen wollen, die zusammen mit dem Zugriffsrecht CONTROL erteilt wurden, müssen Sie entweder jedes einzelne Zugriffsrecht explizit widerrufen oder das Schlüsselwort ALL in der Anweisung REVOKE angeben wie im folgenden Beispiel:

```
REVOKE ALL
  ON EMPLOYEE FROM USER HERON
```

Bei der Arbeit mit typisierten Tabellen sind folgende Punkte für die Tabellen- und Sichtzugriffsrechte zu berücksichtigen.

Anmerkung: Zugriffsrechte können auf jeder Stufe einer Tabellenhierarchie unabhängig erteilt werden. Daher kann ein Benutzer, dem ein Zugriffsrecht für eine übergeordnete Tabelle innerhalb einer Hierarchie typisierter Tabellen erteilt wurde, indirekt auch beliebige untergeordnete Tabellen beeinflussen. Ein Benutzer kann jedoch nur direkt auf eine untergeordnete Tabelle einwirken, wenn er über das nötige Zugriffsrecht für diese untergeordnete Tabelle verfügt.

Die Abhängigkeiten von über- und untergeordneten Tabellen in einer Tabellenhierarchie bedeuten, dass Operationen wie SELECT, UPDATE und DELETE die Zeilen der Zieltabelle für die Operation und alle ihre untergeordneten Tabellen (sofern vorhanden) beeinflussen. Dieses Verhalten wird als *Substitutionsfähigkeit* bezeichnet. Nehmen Sie zum Beispiel an, Sie haben eine Mitarbeitertabelle des Typs Employee_t erstellt, zu der eine untergeordnete Tabelle des Typs Manager_t gehört. Ein Manager ist eine (besondere) Art von Mitarbeiter, wie durch die Abhängigkeit von Typ und untergeordnetem Typ für die strukturierten Typen Employee_t und Manager_t und durch die entsprechende Abhängigkeit von Tabelle und untergeordneter Tabelle für die Tabellen Employee und Manager angegeben wird. Die SQL-Abfrage

```
SELECT * FROM Employee
```

gibt daher als Ergebnis dieser Abhängigkeit die Objektkennung und die Employee_t-Attribute für Mitarbeiter und Manager zurück. Gleichermaßen definiert die Aktualisierungsoperation

```
UPDATE Employee SET Salary = Salary + 1000
```

eine Gehaltserhöhung von 1000 € sowohl für Manager als auch für reguläre Mitarbeiter.

Ein Benutzer mit dem Zugriffsrecht SELECT für die Tabelle Employee kann diese SELECT-Operation ausführen, selbst wenn er kein explizites Zugriffsrecht SELECT für die Tabelle Manager hat. Einem Benutzer wird jedoch nicht erlaubt, eine SELECT-Operation direkt für die untergeordnete Tabelle Manager auszuführen. Daher kann er nicht auf die nicht übernommenen Spalten der Tabelle Manager zugreifen.

Gleichermaßen kann ein Benutzer mit dem Zugriffsrecht UPDATE für die Tabelle Employee eine UPDATE-Operation für die Tabelle Manager ausführen, was sich auf gewöhnliche Mitarbeiter und Manager auswirkt, selbst wenn er nicht über das explizite Zugriffsrecht UPDATE für die Tabelle Manager verfügt. Einem Benutzer wird jedoch nicht erlaubt, UPDATE-Operationen direkt für die untergeordnete Tabelle Manager auszuführen, und daher kann er die nicht übernommenen Spalten der Tabelle Manager nicht aktualisieren.

Paketzugriffsrechte

Ein Paket ist ein Datenbankobjekt, das die Informationen enthält, die vom Datenbankmanager zum effizientesten Zugriff auf Daten für ein bestimmtes Anwendungsprogramm benötigt werden. Zugriffsrechte für Pakete erlauben einem Benutzer das Erstellen und Bearbeiten von Paketen.

Der Benutzer muss über die Berechtigung CONNECT für die Datenbank verfügen, um eines der folgenden Zugriffsrechte verwenden zu können:

- CONTROL gibt dem Benutzer die Möglichkeit, ein Paket erneut zu binden, zu löschen oder auszuführen sowie die Möglichkeit, die Zugriffsrechte auf andere Benutzer zu erweitern. Der Ersteller eines Pakets erhält dieses Zugriffsrecht automatisch. Einem Benutzer mit dem Zugriffsrecht CONTROL werden die Zugriffsrechte BIND und EXECUTE erteilt, und er kann auch anderen Benutzern diese Zugriffsrechte mithilfe der Anweisung GRANT erteilen. (Wenn ein Zugriffsrecht mit der Klausel WITH GRANT OPTION erteilt wird, kann ein Benutzer, der das Zugriffsrecht BIND oder EXECUTE erhält, seinerseits dieses Zugriffsrecht anderen Benutzern erteilen.) Um das Zugriffsrecht CONTROL erteilen zu können, muss der Benutzer über die Berechtigung SYSADM oder DBADM verfügen.
- Das Zugriffsrecht BIND für ein Paket ermöglicht dem Benutzer, dieses Paket zu binden oder erneut zu binden sowie neue Paketversionen des gleichen Paketnamens mit dem gleichen Ersteller hinzuzufügen.
- EXECUTE ermöglicht dem Benutzer, ein Paket auszuführen.

Anmerkung: Alle Paketzugriffsrechte gelten für sämtliche VERSIONen, die den gleichen Paketnamen und den gleichen Ersteller haben.

Zusätzlich zu diesen Zugriffsrechten für Pakete erlaubt das Zugriffsrecht BINDADD für Datenbanken Benutzern das Erstellen neuer Pakete oder das Durchführen eines Rebinds für ein vorhandenes Paket in der Datenbank.

Objekte, auf die über Kurznamen verwiesen wird, müssen Authentifizierungsprüfungen an den Datenquellen durchlaufen, die die Objekte enthalten. Außerdem müssen Paketbenutzer über geeignete Zugriffsrechte oder eine geeignete Berechtigungsstufe für die Datenquellenobjekte an der Datenquelle verfügen.

Für Pakete mit Kurznamen sind möglicherweise weitere Berechtigungsschritte erforderlich, da die DB2-Datenbank dynamische Abfragen verwendet, um mit Datenquellen der DB2-Produktfamilie Daten auszutauschen. Die Berechtigungs-ID,

die das Paket an der Datenquelle ausführt, muss mit der entsprechenden Berechtigung ausgestattet sein, um das Paket an dieser Datenquelle dynamisch ausführen zu können.

Indexzugriffsrechte

Der Ersteller eines Index oder einer Indexspezifikation erhält automatisch das Zugriffsrecht CONTROL für den Index. Das Zugriffsrecht CONTROL für einen Index entspricht effektiv der Berechtigung zum Löschen des Index. Zum Erteilen eines Zugriffsrechts CONTROL für einen Index muss ein Benutzer über die Berechtigung SYSADM oder DBADM verfügen.

Das Zugriffsrecht INDEX auf Tabellenebene ermöglicht einem Benutzer das Erstellen eines Index für diese Tabelle.

Das Zugriffsrecht INDEX auf Kurznamenebene ermöglicht einem Benutzer das Erstellen eines Index für diesen Kurznamen.

Zugriffsrechte für Sequenzen

Der Ersteller einer Sequenz erhält automatisch die Zugriffsrechte USAGE und ALTER für die Sequenz. Das Zugriffsrecht USAGE ist zur Verwendung der Ausdrücke NEXT VALUE und PREVIOUS VALUE für die Sequenz erforderlich. Um anderen Benutzern die Verwendung der Ausdrücke NEXT VALUE und PREVIOUS VALUE zu erlauben, müssen die Sequenzzugriffsrechte der Gruppe PUBLIC erteilt werden. Dadurch erhalten alle Benutzer die Erlaubnis die Ausdrücke mit der angegebenen Sequenz zu verwenden.

Das Zugriffsrecht ALTER für die Sequenz erlaubt dem Benutzer solche Tasks wie das erneute Starten der Sequenz oder das Ändern des Inkrements für zukünftige Sequenzwerte auszuführen. Der Ersteller der Sequenz kann das Zugriffsrecht ALTER anderen Benutzern erteilen. Wenn dazu die Klausel WITH GRANT OPTION verwendet wird, können diese Benutzer selbst diese Zugriffsrechte wieder anderen Benutzern erteilen.

Zugriffsrechte für Routinen

Zugriffsrechte zum Ausführen (EXECUTE) beinhalten Aktionen an allen Arten von Routinen, wie zum Beispiel Funktionen, Prozeduren und Methoden, innerhalb einer Datenbank. Wenn ein Benutzer das Zugriffsrecht EXECUTE besitzt, kann er die entsprechende Routine aufrufen, eine Quellenfunktion erstellen, die von dieser Routine abgeleitet ist (gilt nur für Funktionen), und auf die Routine in einer beliebigen DDL-Anweisung wie CREATE VIEW oder CREATE TRIGGER verweisen.

Der Benutzer, der die extern gespeicherte Prozedur, Funktion oder Methode definiert, erhält das Zugriffsrecht EXECUTE WITH GRANT. Wenn das Zugriffsrecht EXECUTE einem anderen Benutzer über die Klausel WITH GRANT OPTION erteilt wird, kann dieser Benutzer selbst das Zugriffsrecht EXECUTE wieder einem anderen Benutzer erteilen.

Zugriffsrecht USAGE für Auslastungen

Zur Verwendung einer Auslastung (Workload) kann der Datenbankadministrator einem Benutzer, einer Gruppe oder einer Rolle das Zugriffsrecht USAGE für die betreffende Auslastung mithilfe der Anweisung GRANT USAGE ON WORKLOAD erteilen.

Wenn das DB2-Datenbanksystem eine entsprechende Auslastung findet, überprüft es, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für diese Auslastung besitzt. Wenn der Sitzungsbenutzer das Zugriffsrecht USAGE für diese Auslastung nicht

besitzt, sucht das DB2-Datenbanksystem nach der nächsten entsprechenden Auslastung in der geordneten Liste. Das heißt, die Auslastungen, für die der Sitzungsbenutzer das Zugriffsrecht USAGE nicht hat, werden so behandelt, als wären sie nicht vorhanden.

Die Informationen zum Zugriffsrecht USAGE werden in den Katalogen gespeichert und können über die Sicht SYSCAT.WORKLOADAUTH angezeigt werden.

Das Zugriffsrecht USAGE kann mithilfe der Anweisung REVOKE USAGE ON WORKLOAD widerrufen werden.

Ein Benutzer mit der Berechtigung SYSADM oder DBADM kann eine beliebige Auslastung verwenden, die im Katalog vorhanden ist, solange die Auslastung den Verbindungsattributen entspricht.

Die Auslastung SYSDEFAULTUSERWORKLOAD und das Zugriffsrecht USAGE

Das Zugriffsrecht USAGE für die Auslastung SYSDEFAULTUSERWORKLOAD wird der speziellen Gruppe PUBLIC bei der Erstellung einer Datenbank erteilt, wenn die Datenbank ohne die Option RESTRICT erstellt wird. Anderenfalls muss das Zugriffsrecht USAGE durch einen Benutzer mit der Berechtigung SYSADM oder DBADM explizit erteilt werden.

Wenn der Sitzungsbenutzer das Zugriffsrecht USAGE für keine der Auslastungen (einschließlich SYSDEFAULTUSERWORKLOAD) besitzt, wird ein SQL-Fehler zurückgegeben.

Die Auslastung SYSDEFAULTADMWORKLOAD und das Zugriffsrecht USAGE

Das Zugriffsrecht USAGE für die Auslastung SYSDEFAULTADMWORKLOAD kann keinem Benutzer explizit erteilt werden. Nur Benutzer, die den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD ausführen und deren Sitzungsberechtigungs-ID über die Berechtigung SYSADM oder DBADM verfügt, haben die Berechtigung, diese Auslastung zu verwenden.

Die Anweisungen GRANT USAGE ON WORKLOAD und REVOKE USAGE ON WORKLOAD haben auf die Auslastung SYSDEFAULTADMWORKLOAD keine Wirkung.

Aufgaben und erforderliche Berechtigungen

In den verschiedenen Unternehmen und Behörden sind die einzelnen Aufgabengebiete nicht immer in gleicher Weise verteilt. In der folgenden Tabelle werden allgemein verwendete Jobbezeichnungen, die Aufgaben, die diesen normalerweise zufallen, und die Berechtigungen bzw. Zugriffsrechte, die zum Ausführen dieser Aufgaben erforderlich sind, aufgeführt.

Tabelle 2. Allgemeine Jobbezeichnungen, Aufgaben und erforderliche Berechtigungen

JOBBEZEICHNUNG	AUFGABEN	ERFORDERLICHE BERECHTIGUNG
Abteilungsadministrator	Überwacht das Abteilungssystem; erstellt Datenbanken.	Berechtigung SYSCTRL. Berechtigung SYSADM, wenn die Abteilung über eine eigene Instanz verfügt.

Tabelle 2. Allgemeine Jobbezeichnungen, Aufgaben und erforderliche Berechtigungen (Forts.)

JOBBEZEICHNUNG	AUFGABEN	ERFORDERLICHE BERECHTIGUNG
Sicherheitsadministrator	Verwaltet die Sicherheit innerhalb einer oder mehrerer Datenbanken.	Berechtigung SECADM
Datenbankadministrator	Entwirft, entwickelt, betreibt und pflegt eine oder mehrere Datenbanken.	Berechtigungen DBADM und SYSMAINT für eine oder mehrere Datenbanken. In einigen Fällen Berechtigung SYSCTRL.
Systembediener	Überwacht die Datenbank und führt Backups aus.	Berechtigung SYSMAINT
Anwendungsprogrammierer	Entwickelt und testet die Anwendungsprogramme für den Datenbankmanager; kann auch Tabellen mit Testdaten erstellen.	BINDADD, BIND für ein vorhandenes Paket, CONNECT und CREATETAB für eine oder mehrere Datenbanken, einige spezielle Zugriffsrechte für Schemata und eine Liste von Zugriffsrechten für einige Tabellen. CREATE_EXTERNAL_ROUTINE kann ebenfalls erforderlich sein.
Benutzeranalytiker	Definiert die Datenerfordernisse für ein Anwendungsprogramm durch Überprüfen der Systemkatalogsichten.	SELECT für die Katalogsichten; CONNECT für eine oder mehrere Datenbanken
Programmendbenutzer	Führt ein Anwendungsprogramm aus.	EXECUTE für das Paket; CONNECT für eine oder mehrere Datenbanken. Siehe Anmerkung im Anschluss an diese Tabelle.
Berater im Informationszentrum	Definiert die Datenerfordernisse für einen Abfragebenutzer; stellt die Daten zur Verfügung, indem er Tabellen und Sichten erstellt und den Zugriff auf Datenbankobjekte erteilt.	Berechtigung DBADM für eine oder mehrere Datenbanken
Abfragebenutzer	Führt SQL-Anweisungen zum Abrufen, Hinzufügen, Löschen oder Ändern von Daten aus; kann Ergebnisse in Form von Tabellen speichern.	CONNECT für eine oder mehrere Datenbanken; CREATEIN für das Schema der Tabellen und Sichten, die erstellt werden; und SELECT, INSERT, UPDATE, DELETE für einige Tabellen und Sichten

Anmerkung: Wenn ein Anwendungsprogramm dynamische SQL-Anweisungen enthält, muss der Programmendbenutzer möglicherweise neben EXECUTE und CONNECT über weitere Zugriffsrechte (wie SELECT, INSERT, DELETE und UPDATE) verfügen.

Erteilen, Widerrufen und Überwachen von Zugriff

Erteilen von Zugriffsrechten

Damit ein Benutzer Zugriffsrechte für die meisten Datenbankobjekte erteilen kann, muss er über die Berechtigung SYSADM, die Berechtigung DBADM oder das Zugriffsrecht CONTROL für das betreffende Objekt verfügen. Oder der Benutzer muss das jeweilige Zugriffsrecht über die Klausel WITH GRANT OPTION besitzen. Zugriffsrechte können nur für existierende Objekte erteilt werden.

Damit ein Benutzer das Zugriffsrecht CONTROL an einen anderen Benutzer erteilen kann, muss er über die Berechtigung SYSADM oder DBADM verfügen. Zum Erteilen der Berechtigung DBADM muss der Benutzer die Berechtigung SYSADM haben.

Mit der Anweisung GRANT kann ein berechtigter Benutzer Zugriffsrechte erteilen. Ein Zugriffsrecht kann einem oder mehreren Berechtigungsnamen in einer Anweisung erteilt werden. Das Zugriffsrecht kann auch der Gruppe PUBLIC erteilt werden, wodurch es allen Benutzern zur Verfügung gestellt wird. Bei einem Berechtigungsnamen kann es sich um einen Einzelbenutzer oder um eine Gruppe handeln.

Bei Betriebssystemen, unter denen Benutzer denselben Namen wie Gruppen haben können, sollten Sie angeben, ob Sie das Zugriffsrecht dem Benutzer oder der Gruppe erteilen. Die Anweisungen GRANT und REVOKE unterstützen beide die Schlüsselwörter USER und GROUP. Wenn diese optionalen Schlüsselwörter nicht verwendet werden, überprüft der Datenbankmanager die Sicherheitseinrichtung des Betriebssystems, um zu ermitteln, ob der Berechtigungsname einen Benutzer oder eine Gruppe bezeichnet. Wenn es sich bei dem Berechtigungsnamen sowohl um einen Benutzer als auch um eine Gruppe handeln könnte, wird ein Fehler zurückgegeben. Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE dem Benutzer HERON erteilt:

```
GRANT SELECT
       ON EMPLOYEE TO USER HERON
```

Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE der Gruppe HERON erteilt:

```
GRANT SELECT
        ON EMPLOYEE TO GROUP HERON
```

In der Steuerzentrale können Sie zum Erteilen und Widerrufen von Zugriffsrechten für diese Datenbankobjekte die Notizbücher **Schemazugriffsrechte**, **Zugriffsrechte für Tabellenbereiche** und **Sichtzugriffsrechte** verwenden. Führen Sie folgende Schritte aus, um eines dieser Notizbücher zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis der Ordner angezeigt wird, in dem die Objekte enthalten sind, mit denen Sie arbeiten möchten, zum Beispiel der Ordner **Sichten**.
2. Klicken Sie den Ordner an.

   Alle vorhandenen Datenbankobjekte in diesem Ordner werden im Inhaltsteilfenster angezeigt.
3. Klicken Sie das gewünschte Objekt mit der rechten Maustaste im Inhaltsteilfenster an, und wählen Sie **Zugriffsrechte** im Kontextmenü aus.

   Das Notizbuch mit den entsprechenden Zugriffsrechten wird angezeigt.

Widerrufen von Zugriffsrechten

Die Anweisung REVOKE erlaubt es berechtigten Benutzern, Zugriffsrechte zu widerrufen, die anderen Benutzern zuvor erteilt wurden.

Zum Widerrufen von Zugriffsrechten für Datenbankobjekte müssen Sie über die Berechtigung DBADM, die Berechtigung SYSADM oder das Zugriffsrecht CONTROL für das betreffende Objekt verfügen. Beachten Sie, dass es nicht ausreicht, ein Zugriffsrecht mit der Klausel WITH GRANT OPTION zu besitzen, um es widerrufen zu können. Zum Widerrufen des Zugriffsrechts CONTROL eines anderen Benutzers müssen Sie die Berechtigung SYSADM oder DBADM haben. Zum

Widerrufen der Berechtigung DBADM müssen Sie über die Berechtigung SYSADM verfügen. Zugriffsrechte können nur für existierende Objekte widerrufen werden.

Anmerkung: Ein Benutzer ohne die Berechtigung DBADM oder das Zugriffsrecht CONTROL kann kein Zugriffsrecht widerrufen, das er durch Verwendung der Klausel WITH GRANT OPTION erteilt hat. Auch gibt es kein benutzerübergreifendes Widerrufen für die Benutzer, die Zugriffsrechte besitzen, die ihnen von der Person, deren Zugriffsrechte widerrufen werden, erteilt wurden.

Wenn ein explizit erteiltes Zugriffsrecht auf eine Tabelle (oder eine Sicht) für einen Benutzer mit der Berechtigung DBADM widerrufen wird, werden die Zugriffsrechte für andere Sichten, die für diese Tabelle definiert sind, **nicht** widerrufen. Der Grund hierfür ist, dass die Zugriffsrechte für die Sichten auf der Berechtigung DBADM basieren und nicht von expliziten Zugriffsrechten auf die entsprechenden Tabellen abhängig sind.

Wenn ein Zugriffsrecht einem Benutzer und einer Gruppe mit demselben Namen erteilt wurde, müssen Sie beim Widerrufen des Zugriffsrechts das Schlüsselwort GROUP oder USER angeben. Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE des Benutzers HERON widerrufen:

```
REVOKE SELECT
     ON EMPLOYEE FROM USER HERON
```

Im folgenden Beispiel wird das Zugriffsrecht SELECT für die Tabelle EMPLOYEE der Gruppe HERON widerrufen:

```
REVOKE SELECT
      ON EMPLOYEE FROM GROUP HERON
```

Zu beachten ist, dass beim Widerrufen eines Zugriffsrechts für eine Gruppe nicht notwendigerweise das Zugriffsrecht jedes einzelnen Angehörigen dieser Gruppe widerrufen wird. Wenn einem einzelnen Namen ein Zugriffsrecht direkt erteilt wurde, bleibt dieses Zugriffsrecht erhalten, bis es direkt widerrufen wird.

Wenn ein Zugriffsrecht auf eine Tabelle für einen Benutzer widerrufen wird, werden auch Zugriffsrechte für alle Sichten widerrufen, die von dem betreffenden Benutzer erstellt wurden und die von dem widerrufenen Zugriffsrecht auf die Tabelle abhängen. Es werden jedoch nur die vom System implizit erteilten Zugriffsrechte widerrufen. Wenn ein Zugriffsrecht für die Sicht direkt von einem anderen Benutzer erteilt wurde, bleibt dieses erhalten.

Wenn ein Zugriffsrecht auf eine Tabelle für einen Benutzer widerrufen wird, werden auch Zugriffsrechte für alle Sichten widerrufen, die von dem betreffenden Benutzer erstellt wurden und die von dem widerrufenen Zugriffsrecht auf die Tabelle abhängen. Es werden jedoch nur die vom System implizit erteilten Zugriffsrechte widerrufen. Wenn ein Zugriffsrecht für die Sicht direkt von einem anderen Benutzer erteilt wurde, bleibt dieses erhalten.

Es kann zu Situationen kommen, in denen Sie einer Gruppe ein Zugriffsrecht erteilen (GRANT) und nur für ein Mitglied dieser Gruppe widerrufen (REVOKE) möchten. Es gibt nur wenige Möglichkeiten, dies zu tun, ohne die Fehlernachricht SQL0556N zu erhalten:

- Sie können das Mitglied aus der Gruppe entfernen oder eine neue Gruppe mit weniger Mitgliedern erstellen und dieser neuen Gruppe das Zugriffsrecht erteilen (GRANT).

- Sie können das Zugriffsrecht für die Gruppe widerrufen (REVOKE) und es anschließend einzelnen Benutzern (Berechtigungs-IDs) erteilen (GRANT).

Anmerkung: Wenn einem Benutzer das Zugriffsrecht CONTROL für eine Tabelle oder Sicht entzogen wird, behält der Benutzer die Möglichkeit, Zugriffsrechte anderen Benutzern zu erteilen. Durch Erhalt des Zugriffsrechts CONTROL empfängt der Benutzer auch alle anderen Zugriffsrechte mit der Klausel WITH GRANT OPTION. Wird das Zugriffsrecht CONTROL widerrufen, bleiben alle anderen Zugriffsrechte mit der Klausel WITH GRANT OPTION erhalten, bis sie explizit widerrufen werden.

Alle Pakete, die von widerrufenen Zugriffsrechten abhängig sind, werden als ungültig markiert, können jedoch wieder zu gültigen Paketen werden, wenn ein Benutzer mit der entsprechenden Berechtigung einen Rebind für sie durchführt. Pakete können auch wiederhergestellt werden, wenn die Zugriffsrechte später dem Benutzer, der die Anwendung gebunden hat, wieder erteilt werden. Durch Ausführen der Anwendung wird ein erfolgreicher impliziter Rebind ausgelöst. Wenn Zugriffsrechte der Gruppe PUBLIC widerrufen werden, werden alle Pakete ungültig gemacht, die von Benutzern gebunden wurden, die dies nur aufgrund der Zugriffsrechte der Gruppe PUBLIC tun konnten. Wenn die Berechtigung DBADM einem Benutzer entzogen wird, werden alle von diesem Benutzer gebundenen Pakete ungültig gemacht, einschließlich der Pakete, die Dienstprogrammen für Datenbanken zugeordnet sind. Wenn versucht wird, ein Paket zu verwenden, das als ungültig markiert wurde, versucht das System, einen Rebind für dieses Paket auszuführen. Wenn dieser Rebind fehlschlägt, tritt ein Fehler auf (SQLCODE -727). In diesem Fall müssen die Pakete von einem Benutzer explizit erneut gebunden werden, der über Folgendes verfügt:

- Die Berechtigung für den Rebind der Pakete
- Entsprechende Berechtigungen für die Objekte innerhalb der Pakete

Für diese Pakete sollte zu dem Zeitpunkt, an dem die Zugriffsrechte widerrufen werden, ein Rebind durchgeführt werden.

Wenn Sie einen Trigger oder eine SQL-Funktion in Abhängigkeit von einem oder mehreren Zugriffsrechten definieren und Sie eines oder mehrere dieser Zugriffsrechte verlieren, kann der Trigger bzw. die SQL-Funktion nicht verwendet werden.

Verwalten impliziter Berechtigungen durch Erstellen und Löschen von Objekten

Der Datenbankmanager erteilt einem Benutzer, der ein Datenbankobjekt, wie zum Beispiel eine Tabelle oder ein Paket, erstellt, implizit bestimmte Zugriffsrechte. Es werden auch Zugriffsrechte erteilt, wenn Objekte von Benutzern mit der Berechtigung SYSADM oder DBADM erstellt werden. In ähnlicher Weise werden Zugriffsrechte entfernt, wenn ein Objekt gelöscht wird.

Wenn das erstellte Objekt eine Tabelle, ein Kurzname, ein Index oder ein Paket ist, erhält der Benutzer das Zugriffsrecht CONTROL für das Objekt. Wenn das Objekt eine Sicht ist, wird das Zugriffsrecht CONTROL für die Sicht nur dann implizit erteilt, wenn der Benutzer über das Zugriffsrecht CONTROL für alle Tabellen, Sichten und Kurznamen verfügt, auf die in der Sichtdefinition verwiesen wird.

Wenn das explizit erstellte Objekt ein Schema ist, erhält der Schemaeigner die Zugriffsrechte ALTERIN, CREATEIN und DROPIN mit der Klausel WITH GRANT OPTION. Für ein implizit erstelltes Schema wird das Zugriffsrecht CREATEIN der Gruppe PUBLIC erteilt.

Einrichten des Eigentumsrechts für ein Paket

Mit den Befehlen BIND und PRECOMPILE kann ein Anwendungspaket erstellt oder geändert werden. Für jeden der Befehle können Sie die Option OWNER angeben, um den Eigner des generierten Pakets zu benennen.

Für die Benennung des Paketeigners gibt es die folgenden einfachen Regeln:

- Jeder Benutzer kann sich selbst als Eigner benennen. Dies ist die Standardeinstellung, falls die Option OWNER nicht angegeben wird.
- Eine ID mit der Berechtigung SYSADM oder DBADM darf eine beliebige Berechtigungs-ID mit der Option OWNER als Eigner benennen.

Nicht alle Betriebssysteme, die mit DB2-Datenbankprodukten ein Paket binden können, unterstützen die Option OWNER.

Implizite Zugriffsrechte durch ein Paket

Der Zugriff auf Daten in einer Datenbank kann durch Anwendungsprogramme sowie durch Personen, die an einer interaktiven Workstation-Sitzung teilnehmen, angefordert werden. Ein Paket enthält Anweisungen, die es Benutzern ermöglichen, eine Vielzahl von Aktionen für viele Datenbankobjekte auszuführen. Für jede dieser Aktionen ist mindestens ein Zugriffsrecht erforderlich.

Zugriffsrechte, die Einzelbenutzern, die das Paket binden, und der Gruppe PUBLIC erteilt werden, sowie die Rollen, die den Einzelbenutzern und der Gruppe PUBLIC erteilt werden, werden beim Binden von statischen SQL- und XQuery-Anweisungen zur Berechtigungsprüfung verwendet. Zugriffsrechte, die über Gruppen erteilt wurden, und die Rollen, die Gruppen erteilt wurden, werden beim Binden von statischen SQL- und XQuery-Anweisungen *nicht* zur Berechtigungsprüfung verwendet. Dem Benutzer mit einer gültigen *authID*, der ein Paket bindet, müssen entweder alle zum Ausführen der statischen SQL- oder XQuery-Anweisungen im Paket erforderlichen Zugriffsrechte explizit oder implizit die erforderlichen Zugriffsrechte durch PUBLIC, durch die Rollen, die der Gruppe PUBLIC erteilt wurden, oder durch die Rollen, die dem Benutzer erteilt wurden, erteilt worden sein, sofern beim Binden des Pakets nicht VALIDATE RUN angegeben wurde. Wurde während der Ausführung von BIND die Option VALIDATE RUN angegeben, schlägt BIND beim Auftreten von Berechtigungsfehlern für statische SQL- oder XQuery-Anweisungen innerhalb des Pakets nicht fehl. Die entsprechenden SQL- oder XQuery-Anweisungen werden dann während der Laufzeit nochmals geprüft. Zugriffsrechte über PUBLIC sowie Gruppen-, Rollen- und Einzelbenutzerzugriffsrechte werden *alle* bei der Überprüfung verwendet, um sicherzustellen, dass der Benutzer die nötige Berechtigung (Zugriffsrecht BIND oder BINDADD) zum Binden des Pakets besitzt.

Pakete können sowohl statische als auch dynamische SQL- und XQuery-Anweisungen umfassen. Zum Verarbeiten eines Pakets mit statischen Abfragen muss ein Benutzer lediglich über das Zugriffsrecht EXECUTE für das Paket verfügen. Der Benutzer kann in diesem Fall implizit die Zugriffsrechte des Paketbinders für statische Abfragen im Paket erhalten. Dies ist allerdings nur im Rahmen der durch das Paket festgelegten Einschränkungen möglich.

Wenn das Paket dynamische SQL- oder XQuery-Anweisungen enthält, hängen die erforderlichen Zugriffsrechte von dem Wert ab, der beim Vorkompilieren (PRECOMPILE) oder Binden (BIND) des Pakets für DYNAMICRULES angegeben wurde. Weitere Informationen finden Sie im Abschnitt zu den Auswirkungen von DYNAMICRULES auf dynamische Abfragen.

Indirekte Zugriffsrechte durch ein Paket mit Kurznamen

Wenn ein Paket Verweise auf Kurznamen enthält, ist die Berechtigungsverarbeitung für Ersteller und Benutzer von Paketen etwas komplexer. Wenn ein Paketersteller Pakete, die Kurznamen enthalten, erfolgreich bindet, muss der Paketersteller keine Authentifizierung oder Überprüfung von Zugriffsrechten für Tabellen und Sichten durchlaufen, auf die an der Datenquelle mit den Kurznamen verwiesen wird. Wer jedoch das Paket ausführt, muss eine Authentifizierung und eine Berechtigungsprüfung an den Datenquellen durchlaufen.

Nehmen Sie z. B. an, dass eine Datei mit der Endung .SQC eines Paketerstellers mehrere SQL- oder XQuery-Anweisungen enthält. Eine statische Anweisung verweist auf eine lokale Tabelle. Eine weitere dynamische Anweisung verweist auf einen Kurznamen. Wenn das Paket gebunden wird, wird die Berechtigungs-ID des Paketerstellers verwendet, um Zugriffsrechte für die lokale Tabelle und den Kurznamen zu überprüfen. Allerdings findet für die Datenquellenobjekte, die durch den Kurznamen angegeben werden, keine Überprüfung statt. Wenn ein anderer Benutzer das Paket ausführt und das Zugriffsrecht EXECUTE für dieses Paket gegeben ist, muss dieser Benutzer für die Anweisung, die auf die Tabelle verweist, keinerlei weitere Überprüfung seiner Zugriffsrechte durchlaufen. Allerdings muss der Benutzer, der das Paket ausführt, für die Anweisung mit dem Kurznamen eine Authentifizierung und eine Überprüfung der Zugriffsrechte an der Datenquelle durchlaufen.

Wenn die Datei mit der Endung .SQC nur dynamische SQL- und XQuery-Anweisungen und eine Kombination aus Tabellen- und Kurznamenverweisen enthält, ist die Berechtigungsprüfung durch die DB2-Datenbank für lokale Objekte und Kurznamen ähnlich. Paketbenutzer müssen die Überprüfung der Zugriffsrechte für beliebige lokale Objekte (Tabellen, Sichten), die sich in der Anweisung befinden, durchlaufen. Außerdem werden die Zugriffsrechte für Kurznamenobjekte überprüft. (Paketbenutzer müssen eine Authentifizierung und eine Überprüfung der Zugriffsrechte an der Datenquelle durchlaufen, die die durch die Kurznamen angegebenen Objekte enthält.) In beiden Fällen müssen die Benutzer des Pakets über das Zugriffsrecht EXECUTE verfügen.

Die ID und das Kennwort des Benutzers, der das Paket ausführt, werden für die gesamte Verarbeitung der Authentifizierungen und Zugriffsrechte verwendet. Diese Daten können geändert werden, indem eine Benutzerzuordnung erstellt wird.

Anmerkung: In statischen SQL- und XQuery-Anweisungen dürfen Kurznamen nicht angegeben werden. Verwenden Sie mit Paketen, die Kurznamen enthalten, nicht die Option DYNAMICRULES (auf BIND gesetzt).

Für Pakete mit Kurznamen sind möglicherweise weitere Berechtigungsschritte erforderlich, da die DB2-Datenbank dynamisches SQL verwendet, um mit Datenquellen der DB2-Produktfamilie Daten auszutauschen. Die Berechtigungs-ID, die das Paket an der Datenquelle ausführt, muss mit der entsprechenden Berechtigung ausgestattet sein, um das Paket an dieser Datenquelle dynamisch ausführen zu können.

Steuern des Zugriffs auf Daten mit Sichten

Eine Sicht bietet eine Möglichkeit, den Zugriff auf eine Tabelle zu beschränken bzw. die Zugriffsrechte für eine Tabelle zu erweitern.

Die Verwendung einer Sicht bietet die folgenden Steuerungsmöglichkeiten für den Zugriff auf eine Tabelle:

- Es wird der Zugriff nur auf bestimmte Spalten der Tabelle ermöglicht.

Für Benutzer und Anwendungsprogramme, die nur einen Zugriff auf bestimmte Spalten einer Tabelle benötigen, kann ein berechtigter Benutzer eine Sicht erstellen, um den Zugriff auf die benötigten Spalten zu beschränken.

- Es wird der Zugriff nur auf eine Teilmenge der Tabellenzeilen ermöglicht.

  Durch Angabe einer Klausel WHERE in der Unterabfrage einer Sichtdefinition kann ein berechtigter Benutzer die Zeilen beschränken, auf die über eine Sicht zugegriffen wird.
- Es wird der Zugriff nur auf eine Teilmenge der Tabellenzeilen oder -spalten an der Datenquelle ermöglicht. Wenn Sie durch Kurznamen auf Datenquellen zugreifen, können Sie lokale DB2-Datenbanksichten erstellen, die auf Kurznamen verweisen. Diese Sichten können auf Kurznamen aus einer oder mehreren Datenquellen verweisen.

  Anmerkung: Da Sie eine Sicht erstellen können, die Verweise auf Kurznamen aus mehreren Datenquellen enthalten kann, können die Benutzer von einer Sicht aus auf Daten mehrerer Datenquellen zugreifen. Diese Sichten sind *Sichten mehrerer Speicherpositionen*. Solche Sichten sind nützlich, wenn Sie Daten von Spalten mit sensiblen Tabellendaten in einer verteilten Umgebung verknüpfen oder wenn einzelne Benutzer für spezifische Objekte nicht über die erforderlichen Zugriffsrechte auf Datenquellen verfügen.

Zum Erstellen einer Sicht muss ein Benutzer über die Berechtigung SYSADM, die Berechtigung DBADM oder das Zugriffsrecht CONTROL oder SELECT für jede Tabelle, jede Sicht oder jeden Kurznamen verfügen, auf die in der Sichtdefinition verwiesen wird. Der Benutzer muss darüber hinaus in der Lage sein, ein Objekt in dem für die Sicht angegebenen Schema zu erstellen. Das heißt, er muss über das Zugriffsrecht CREATEIN für ein vorhandenes Schema oder die Berechtigung IMPLICIT_SCHEMA für die Datenbank haben, wenn das Schema noch nicht existiert.

Wenn Sie Sichten mit Verweisen auf Kurznamen erstellen, müssen Sie über keine weitere Berechtigung auf die Datenquellen (Tabellen oder Sichten) verfügen, auf die mit den Kurznamen in der Sicht verwiesen wird. Allerdings müssen die Benutzer der Sicht die Berechtigung SELECT oder eine gleichwertige Berechtigungsstufe für die zugrunde liegenden Datenquellenobjekte haben, wenn sie auf die Sicht zugreifen.

Wenn die Benutzer an der Datenquelle nicht die entsprechende Berechtigung für die zugrunde liegenden Objekte (Tabellen und Sichten) haben, können Sie wie folgt vorgehen:

1. Erstellen Sie eine Sicht auf die Datenquelle für diejenigen Spalten in der Tabelle der Datenquelle, auf die der Benutzer zureifen darf.
2. Erteilen Sie den Benutzern das Zugriffsrecht SELECT auf diese Sicht.
3. Erstellen Sie einen Kurznamen, der auf diese Sicht verweist.

Anschließend können die Benutzer auf die Spalten zugreifen, indem Sie eine SELECT-Anweisung absetzen, die den neuen Kurznamen als Verweis enthält.

Im folgenden Szenario erhalten Sie ein genaueres Beispiel, wie Sichten zur Einschränkung des Datenzugriffs verwendet werden können.

Viele Benutzer benötigen vielleicht aus unterschiedlichen Gründen Zugriff auf Informationen der Tabelle STAFF. Zum Beispiel:

- Die Personalabteilung muss in der Lage sein, die gesamte Tabelle anzuzeigen und zu aktualisieren.

Diese Anforderung kann leicht dadurch erfüllt werden, dass der Gruppe PERSONNL die Zugriffsrechte SELECT und UPDATE für die Tabelle STAFF erteilt werden:

```
GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
```

- Die einzelnen Abteilungsleiter müssen in der Lage sein, Gehaltsdaten für ihre Mitarbeiter anzuzeigen.

  Diese Anforderung kann erfüllt werden, indem eine Sicht für jeden Abteilungsleiter erstellt wird. Zum Beispiel kann folgende Sicht für den Leiter der Abteilung Nr. 51 erstellt werden:

  ```
  CREATE VIEW EMP051 AS
          SELECT NAME,SALARY,JOB FROM STAFF
          WHERE DEPT=51
     GRANT SELECT ON TABLE EMP051 TO JANE
  ```

  Der Abteilungsleiter mit dem Berechtigungsnamen JANE würde die Sicht EMP051 genauso wie die Tabelle STAFF abfragen. Wenn er auf die Sicht EMP051 der Tabelle STAFF zugreift, werden dem Abteilungsleiter folgende Informationen angezeigt:

NAME	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- Alle Benutzer müssen in der Lage sein, andere Mitarbeiter zu finden. Diese Anforderung kann erfüllt werden, indem eine Sicht auf die Spalte NAME der Tabelle STAFF und auf die Spalte LOCATION der Tabelle ORG erstellt wird und die beiden Tabellen über ihre jeweiligen Spalten DEPT und DEPTNUMB verknüpft werden:

  ```
  CREATE VIEW EMPLOCS AS
          SELECT NAME, LOCATION FROM STAFF, ORG
          WHERE STAFF.DEPT=ORG.DEPTNUMB
     GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
  ```

  Wenn Benutzer auf die Sicht EMPLOCS über die Standorte der Mitarbeiter zugreifen, werden folgende Informationen angezeigt:

NAME	LOCATION
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington
Pernal	Washington
James	Washington
Sneider	Washington

NAME	LOCATION
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

Steuern des Zugriffs für Benutzer mit SYSADM- und DBADM-Berechtigung

Es kann sinnvoll sein, den Zugriff auf Daten durch Benutzer, die über die Berechtigungen SYSADM und DBADM verfügen, zu überwachen oder zu steuern.

Führen Sie die folgenden Schritte aus, wenn Sie den Zugriff von Systemadministratoren und Datenbankadministratoren überwachen und steuern wollen:

1. Erstellen Sie eine Prüfrichtlinie, die die Ereignisse überwacht, die für Benutzer mit den Berechtigungen SYSADM und DBADM erfasst werden sollen.
2. Ordnen Sie diese Prüfrichtlinie der Berechtigung SYSADM und der Berechtigung DBADM zu.
3. Erstellen Sie eine Rolle, und erteilen Sie dieser Rolle die Berechtigung DBADM.
4. Definieren Sie einen gesicherten Kontext, und machen Sie die Rolle zur Standardrolle für diesen gesicherten Kontext.

   Erteilen Sie die Zugehörigkeit zu dieser Rolle keiner Berechtigungs-ID explizit. Auf diese Weise ist die Rolle nur über diesen gesicherten Kontext verfügbar, und Benutzer erhalten DBADM-Berechtigungen nur, wenn sie sich innerhalb der Grenzen des gesicherten Kontexts befinden.

Anmerkung: Diese Option ist kein Schutz gegen Benutzer mit der Berechtigung SYSADM, weil solche Benutzer über eine implizite Berechtigung DBADM verfügen.

5. Es gibt zwei Verfahren zur Steuerung, wie Benutzer auf den gesicherten Kontext zugreifen:
   - Impliziter Zugriff: Erstellen Sie einen eindeutigen gesicherten Kontext für jeden Benutzer. Wenn der Benutzer eine reguläre Verbindung herstellt, die den Attributen des gesicherten Kontexts entspricht, wird er als implizit gesicherter Benutzer erkannt und erhält Zugriff auf die Rolle.
   - Expliziter Zugriff: Erstellen Sie einen gesicherten Kontext mit der Klausel WITH USE FOR, indem Sie alle Benutzer definieren, die Zugriff haben sollen. Erstellen Sie eine Anwendung, über die diese Benutzer Datenbankanforderungen absetzen können. Die Anwendung stellt eine explizite gesicherte Verbindung her. Wenn ein Benutzer eine Anforderung absetzt, wechselt die Anwendung zur entsprechenden Benutzer-ID und führt die Anforderung als dieser Benutzer in der Datenbank aus.
6. Erstellen Sie eine Prüfrichtlinie, die die Ereignisse überwacht, die Sie für Benutzer des gesicherten Kontexts erfassen wollen, und ordnen Sie sie dem gesicherten Kontext zu.
7. Wenn Sie hochsensible Daten haben, erstellen Sie eine Prüfrichtlinie, die die Kategorie EXECUTE überwacht, und ordnen Sie diese Richtlinie den Tabellen zu, in denen die zu überwachenden sensiblen Daten enthalten sind. Durch die Kategorie EXECUTE werden alle Abfragen unabhängig vom ausführenden Benutzer erfasst, die auf diese Tabellen zugreifen.

Anmerkung: Wenn Sie Benutzern mit den Berechtigungen SYSADM und DBADM explizit den Zugriff auf Daten in Tabellen verwehren wollen, ziehen Sie in Betracht, für die schutzwürdigen Tabellen den Sicherheitsmechanismus der kennsatzbasierten Zugriffssteuerung (LBAC) zu verwenden.

Datenverschlüsselung

Zur Verschlüsselung von Daten im Speicher (in diesem Abschnitt beschrieben) können Sie die folgenden integrierten Funktionen für Verschlüsselung und Entschlüsselung verwenden: ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR und GETHINT. Zur Verschlüsselung von Daten bei der Übertragung zwischen Clients und DB2-Datenbanken können Sie die Authentifizierungstypen DATA_ENCRYPT und SERVER_ENCRYPT oder die Unterstützung des DB2-Datenbanksystems für Secure Socket Layer (SSL) nutzen.

Die integrierte Funktion ENCRYPT verschlüsselt Daten mithilfe eines kennwortbasierten Verschlüsselungsverfahrens. Diese Funktionen ermöglichen ferner das Einbinden eines Kennworthinweises. Der Kennworthinweis ist in die verschlüsselten Daten eingebunden. Nach der Verschlüsselung können die Daten nur unter Verwendung des richtigen Kennworts wieder entschlüsselt werden. Anwendungsentwickler, die diese Funktionen verwenden möchten, sollten die Verwaltung vergessener Kennwörter und unbenutzbarer Daten in Ihre Planung mit einbeziehen.

Das Ergebnis der ENCRYPT-Funktionen sind Daten des Typs VARCHAR FOR BIT DATA (mit einer Begrenzung von 32.631).

Nur Daten der Typen CHAR, VARCHAR und FOR BIT DATA können verschlüsselt werden.

Mit den Funktionen DECRYPT_BIN und DECRYPT_CHAR werden Daten mit der Entschlüsselung auf Kennwortbasis entschlüsselt.

DECRYPT_BIN gibt immer Daten des Typs VARCHAR FOR BIT DATA zurück, während DECRYPT_CHAR immer Daten des Typs VARCHAR zurückgibt. Da das erste Argument CHAR FOR BIT DATA oder VARCHAR FOR BIT DATA sein kann, gibt es Fälle, in denen das Ergebnis nicht mit dem ersten Argument übereinstimmt.

Die Länge des Ergebnisses hängt von den Byte bis zur nächsten 8-Byte-Grenze ab. Die Länge des Ergebnisses könnte die Länge des Datenarguments plus 40 plus die Anzahl von Byte bis zur nächsten 8-Byte-Grenze sein, wenn der optionale Hinweisparameter angegeben wird. Oder die Länge des Ergebnisses könnte die Länge des Datenarguments plus 8 plus die Anzahl von Byte bis zur nächsten 8-Byte-Grenze sein, wenn der optionale Hinweisparameter nicht angegeben wird.

Mit der Funktion GETHINT wird ein eingebundener Kennworthinweis wieder zurückgegeben. Ein Kennworthinweis ist ein Ausdruck, mit dessen Hilfe die Eigner von Daten sich wieder an die Kennwörter erinnern können. Das Wort „Ozean" kann zum Beispiel als Hinweis für das Kennwort "Pazifik" verwendet werden.

Das Kennwort, das zum Verschlüsseln der Daten verwendet wird, wird mit einer der folgenden Methoden ermittelt:

- Kennwortargument. Das Kennwort ist eine Zeichenfolge, die beim Aufruf der Funktion ENCRYPT explizit übergeben wird. Die Daten werden mit dem angegebenen Kennwort verschlüsselt und entschlüsselt.
- Sonderregister für Verschlüsselungskennwort. Die Anweisung SET ENCRYPTION PASSWORD verschlüsselt den Kennwortwert und sendet das verschlüsselte Kennwort an den Datenbankmanager, der das Kennwort in einem Sonderregister speichert. Werden die Funktionen ENCRYPT, DECRYPT_BIN und DECRYPT_CHAR ohne Kennwortparameter aufgerufen, wird der Wert im Sonderregister ENCRYPTION PASSWORD verwendet. Das Sonderregister ENCRYPTION PASSWORD wird nur in verschlüsselter Form gespeichert.

  Der Anfangs- oder Standardwert für das Sonderregister ist eine leere Zeichenfolge.

Gültige Längen für Kennwörter liegen zwischen 6 und 127 einschließlich dieser Längen selbst. Gültige Längen für Hinweise liegen zwischen 0 und 32 einschließlich.

Konfigurieren der SSL-Unterstützung in einer DB2-Instanz

Das DB2-Datenbanksystem unterstützt SSL (Secure Socket Layer). Dies bedeutet, dass eine Clientanwendung, die IBM Data Server Driver for JDBC and SQLJ verwendet, die Verbindung zu einer DB2-Datenbank über einen SSL-Socket herstellen kann. Zur Aktivierung der SSL-Unterstützung in einer DB2-Instanz setzen Sie die Registrierdatenbankvariable **DB2COMM** auf den Wert SSL, erstellen eine SSL-Konfigurationsdatei und starten die Instanz erneut.

Vorbereitung vor der Konfiguration der SSL-Unterstützung:

- Stellen Sie sicher, dass der Pfad zu den GSKit-Bibliotheken in der Umgebungsvariablen **PATH** unter Windows bzw. in den Umgebungsvariablen **LIBPATH**, **SHLIB_PATH** oder **LD_LIBRARY_PATH** unter Linux und UNIX angegeben ist.
- Stellen Sie sicher, dass der Verbindungskonzentrator nicht aktiviert ist. Die SSL-Unterstützung wird in der DB2-Instanz nicht aktiviert, wenn der Verbindungskonzentrator aktiv ist.

Wenn Sie feststellen wollen, ob der Verbindungskonzentrator aktiviert ist, führen Sie den Befehl GET DATABASE MANAGER CONFIGURATION aus. Wenn der Konfigurationsparameter **MAX_CONNECTIONS** einen größeren Wert als der Konfigurationsparameter **MAX_COORDAGENTS** hat, ist der Verbindungskonzentrator aktiviert.

SSL wird zur Kommunikation zwischen IBM Data Server Driver for JDBC and SQLJ (Verbindungen des Typs 4) und DB2-Datenbankprodukten unterstützt.

Die folgenden unterstützten Plattformen enthalten SSL-Unterstützung für den DB2-Datenserver:

- AIX
- HP-UX auf Itanium-basierten HP Integrity Series-Systemen (IA-64)
- Linux auf x86, x64, IA64, POWER-Servern (64 Bit) und zSeries (64 Bit) oder System z9
- Solaris auf x64
- Windows auf 32-Bit-, x64- und Itanium-basierten Systemen

Die SSL-Kommunikation findet immer im FIPS-Modus statt.

Wenn Sie DB2 Connect für System i, DB2 Connect für System z oder DB2 Enterprise Edition auf einem zwischengeschalteten Serversystem verwenden, um DB2-Clients mit einer Hostdatenbank oder einer System i-Datenbank zu verbinden, ist die SSL-Unterstützung zwischen dem als Gateway fungierenden DB2-Datenbankprodukt und der Hostdatenbank bzw. der System i-Datenbank nicht verfügbar. In diesem Szenario steht jedoch die SSL-Unterstützung zwischen IBM Data Server Driver for JDBC and SQLJ (Typ-4-Konnektivität) auf dem DB2-Client und dem als Gateway eingesetzten DB2-Datenbankprodukt zur Verfügung.

Gehen Sie wie folgt vor, um die SSL-Unterstützung in einer DB2-Instanz zu konfigurieren:

1. Melden Sie sich als DB2-Instanzeigner an.
2. Erstellen Sie eine SSL-Konfigurationsdatei:
   - Linux und UNIX: *INSTHOME*/cfg/SSLconfig.ini
   - Windows: *INSTHOME*/SSLconfig.ini

   Dabei ist *INSTHOME* das Ausgangsverzeichnis der Instanz.

   Es wird empfohlen, über die Dateiberechtigungen den Zugriff auf die Datei 'SSLconfig.ini' einzuschränken, da diese Datei sensible Daten enthalten kann. Beschränken Sie zum Beispiel die Lese- und Schreibberechtigung für die Datei auf Mitglieder der Gruppe SYSADM, wenn die Datei das Kennwort für den Keystore (Schlüsselspeicher) enthält.
3. Fügen Sie SSL-Parameter in die SSL-Konfigurationsdatei ein. Die Datei SSLconfig.ini enthält die SSL-Parameter, die zum Laden und Starten von SSL verwendet werden. Nachfolgend werden die SSL-Parameter aufgelistet:

Tabelle 3. SSL-Parameter in der SSL-Konfigurationsdatei

Name des SSL-Parameters	Beschreibung
DB2_SSL_KEYSTORE_FILE	Der vollständig qualifizierte Dateiname des Schlüsselspeichers (KeyStore), in dem das Serverzertifikat gespeichert ist.
DB2_SSL_KEYSTORE_PW	Das Kennwort für den Keystore, in dem das Serverzertifikat gespeichert ist.

Tabelle 3. SSL-Parameter in der SSL-Konfigurationsdatei (Forts.)

Name des SSL-Parameters	Beschreibung
DB2_SSL_KEYSTORE_LABEL	Die Kennzeichnung des Serverzertifikats.
DB2_SSL_LISTENER	Der Servicename oder die Portnummer für den SSL-Listener.

Anmerkung:

- Der Parameter **DB2_SSL_KEYSTORE_PW** kann einen Nullwert enthalten und kann weggelassen werden, wenn kein Schlüsselwort für die Keystore-Datei erforderlich ist.
- Wenn der Parameter **DB2_SSL_KEYSTORE_LABEL** weggelassen wird, wird das Standardserverzertifikat verwendet. Wenn das Standardserverzertifikat nicht vorhanden ist, schlägt die SSL-Konfiguration fehl.
- Der für den Parameter **DB2_SSL_LISTENER** verwendete Wert muss sich von dem Wert unterscheiden, der im Konfigurationsparameter **SVCENAME** des Datenbankmanagers verwendet wird. Wenn Sie versuchen, die DB2-Instanz zu starten, und sowohl SSL als auch TCP/IP an derselben Portnummer empfangsbereit sind, tritt ein Fehler SQL5043N auf.

Das folgende Beispiel zeigt eine Datei 'SSLconfig.ini':

```
DB2_SSL_KEYSTORE_FILE=/home/test1/GSKit/Keystore/key.kdb
DB2_SSL_LISTENER=20397
DB2_SSL_KEYSTORE_PW=aaa111
```

4. Fügen Sie den Wert SSL der Registrierdatenbankvariablen **DB2COMM** hinzu. Beispiel:

   ```
   db2set -i db2inst1 DB2COMM=SSL
   ```

   Dabei ist *db2inst1* der Name der DB2-Instanz. Der Datenbankmanager kann mehrere Protokolle gleichzeitig unterstützen. Wenn Sie zum Beispiel die beiden Kommunikationsprotokolle TCP/IP und SSL aktivieren wollen, geben Sie den folgenden Befehl ein:

   ```
   db2set -i db2inst1 DB2COMM=SSL,TCPIP
   ```

5. Starten Sie die DB2-Instanz erneut. Beispiel:

   ```
   db2stop
   db2start
   ```

Prüfen von DB2-Aktivitäten

Einführung in die DB2-Prüffunktion

Zur Verwaltung des Zugriffs auf schutzwürdige Daten können Sie eine Reihe von Authentifizierungs- und Zugriffssteuerungsmechanismen verwenden, um Regeln und Steuerelemente für bekanntes und akzeptables Zugriffsverhalten einzurichten. Zum Schutz gegen unbekanntes bzw. nicht akzeptables Zugriffsverhalten und zur Erkennung eines solchen Verhaltens müssen Sie den Zugriff auf die Daten außerdem überwachen. Das DB2-Datenbanksystem stellt eine Prüffunktion bereit, die Sie bei dieser Aufgabe unterstützt.

Die erfolgreiche Überwachung unerwünschter Datenzugriffe mit anschließender Analyse dieser Vorgänge kann zu verbesserter Datenzugriffssteuerung und damit zur Verhinderung böswilliger oder unbedachter Datenzugriffe beitragen. Durch

Überwachung von Anwendungs- und Benutzerzugriffen, einschließlich der Systemverwaltungsaktionen, kann ein fortlaufendes Protokoll aller an Ihrer Datenbank ausgeführten Aktivitäten erstellt werden.

Die DB2-Prüffunktion generiert ein Prüfprotokoll für eine Reihe vordefinierter Datenbankereignisse und ermöglicht Ihnen die Verwaltung dieses Protokolls. Die von dieser Funktion generierten Datensätze werden in einer Prüfprotokolldatei aufgezeichnet. Die Analyse dieser Datensätze macht Nutzungsmuster deutlich, an denen Systemmissbrauch erkennbar wird. Sobald der Systemmissbrauch identifiziert ist, können entsprechende Gegenmaßnahmen eingeleitet werden.

Die Prüffunktion bietet die Möglichkeit einer Prüfung sowohl auf Instanzebene auch auf der Ebene einzelner Datenbanken und zeichnet alle Aktivitäten auf Instanz- und Datenbankebene unabhängig voneinander jeweils in separaten Protokollen auf. Der Systemadministrator (der die Berechtigung SYSADM auf Instanzebene besitzt) kann das Tool 'db2audit' verwenden, um eine Prüfung auf Instanzebene zu konfigurieren und zu steuern, wann die entsprechenden Prüfdaten erfasst werden. Der Systemadministrator kann mithilfe des Tools 'db2audit' Prüfprotokolle der Instanz- und der Datenbankebene archivieren sowie Prüfdaten aus archivierten Protokollen beider Typen extrahieren.

Der Sicherheitsadministrator (der die Berechtigung SECADM auf der Datenbankebene besitzt) kann Prüfrichtlinien in Verbindung mit der SQL-Anweisung AUDIT verwenden, um die Prüfvorschriften für eine einzelne Datenbank zu konfigurieren und zu steuern. Mithilfe der gespeicherten Prozedur SYSPROC.AUDIT_ARCHIVE, der Tabellenfunktion SYSPROC.AUDIT_LIST_LOGS und der gespeicherten Prozedur SYSPROC.AUDIT_DELIM_EXTRACT kann der Sicherheitsadministrator Prüfprotokolle archivieren, interessante Protokolle lokalisieren und Daten in Dateien mit begrenzter Satzlänge zur Analyse extrahieren.

Beim Arbeiten in einer Umgebung mit partitionierten Datenbanken treten viele der zu protokollierenden Ereignisse in der Datenbankpartition auf, mit der der Benutzer verbunden ist (Koordinatorpartition), oder in der Katalogpartition (sofern diese beiden Datenbankpartitionen nicht identisch sind). Dies bedeutet, dass Prüfsätze von mehreren Datenbankpartitionen generiert werden können. Ein Teil jedes Prüfsatzes enthält Informationen, die die Koordinatorpartition und die Ursprungsdatenbankpartition (die Partition, aus der der Prüfsatz generiert wurde), angeben.

Auf der Instanzebene muss die Prüffunktion explizit mit den Befehlen db2audit start und db2audit stop gestartet bzw. gestoppt werden. Wenn Sie die Prüfung auf Instanzebene starten, arbeitet die Prüffunktion mit vorhandenen Prüfkonfigurationsdaten. Da die Prüffunktion vom DB2-Datenbankserver unabhängig ist, bleibt sie auch dann aktiv, wenn die Instanz gestoppt wird. Beim Stoppen der Instanz kann sogar ein Prüfsatz im Prüfprotokoll generiert werden. Zum Starten einer Prüfung auf der Datenbankebene ordnen Sie einem beliebigen zu prüfenden Objekt eine Prüfrichtlinie zu.

Kategorien von Prüfsätzen

Verschiedene Kategorien von Prüfsätzen können generiert werden. Beachten Sie in der folgenden Beschreibung der Kategorien möglicher Prüfereignisse, dass auf jeden Kategorienamen ein aus einem Wort bestehendes Schlüsselwort folgt, das den Typ der Kategorie identifiziert. Folgende Kategorien von Prüfereignissen stehen zur Verfügung:

- Prüfung (AUDIT). Generiert Prüfsätze, wenn Prüfeinstellungen geändert werden oder auf das Prüfprotokoll zugegriffen wird.

- Berechtigungsprüfung (CHECKING). Generiert Prüfsätze während der Berechtigungsprüfung, wenn versucht wird, auf DB2-Datenbankobjekte oder -funktionen zuzugreifen oder sie zu manipulieren.
- Objektpflege (OBJMAINT). Generiert Prüfsätze beim Erstellen oder Löschen von Datenobjekten sowie beim Ändern bestimmter Objekte.
- Sicherheitspflege (SECMAINT). Generiert Prüfsätze bei folgenden Aktivitäten:
  - Erteilen oder Widerrufen von Datenbankzugriffsrechten oder Datenbankberechtigungen
  - Erteilen oder Widerrufen von Sicherheitskennsätzen oder Freistellungen
  - Ändern der Gruppenberechtigung, Rollenberechtigung oder Überschreiben bzw. Einschränken von Attributen einer LBAC-Sicherheitsrichtlinie
  - Erteilen oder Widerrufen des Zugriffsrechts SETSESSIONUSER
  - Erteilen oder Widerrufen von Berechtigungen DBADM oder SECADM
  - Ändern der Konfigurationsparameter SYSADM_GROUP, SYSCTRL_GROUP, SYSMAINT_GROUP oder SYSMON_GROUP
- Systemverwaltung (SYSADMIN). Generiert Prüfsätze, wenn Operationen ausgeführt werden, für die die Berechtigung SYSADM, SYSMAINT oder SYSCTRL erforderlich ist.
- Benutzergültigkeitsprüfung (VALIDATE). Generiert Prüfsätze bei der Vergabe von Benutzerberechtigungen und beim Abrufen von Systemsicherheitsdaten.
- Operationskontext (CONTEXT). Generiert Prüfsätze, die den Operationskontext darstellen, wenn eine Datenbankoperation ausgeführt wird. Diese Prüfkategorie erleichtert die Interpretation der Prüfprotokolldatei. Bei Verwendung zusammen mit dem Ereigniskorrelationsfeld des Protokolls kann eine Gruppe von Ereignissen auf eine einzige Datenbankoperation zurückgeführt werden. Beispiel: Eine Abfrageanweisung für dynamische Abfragen, eine Paketkennung für statische Abfragen oder ein Indikator des ausgeführten Operationstyps (z. B. CONNECT) kann den erforderlichen Kontext zum Analysieren der Prüfergebnisse liefern.

  Anmerkung: Die SQL- oder XQuery-Anweisung, die den Operationskontext bereitstellt, kann sehr lang sein. Sie wird jedoch im CONTEXT-Prüfsatz vollständig angegeben. Dadurch kann der CONTEXT-Prüfsatz sehr umfangreich werden.
- Ausführen (EXECUTE). Generiert Prüfsätze während der Ausführung von SQL-Anweisungen.

Für alle oben aufgeführten Kategorien können Sie auf fehlgeschlagene oder erfolgreich ausgeführte Operationen oder beide Arten prüfen.

Alle Operationen auf dem Datenbankserver können mehrere Prüfsätze generieren. Die tatsächliche Anzahl der im Prüfprotokoll generierten Prüfsätze hängt von der Anzahl der aufzuzeichnenden Ereigniskategorien ab, die in der Konfiguration der Prüffunktion angegeben werden. Sie ist außerdem davon abhängig, ob erfolgreiche oder fehlgeschlagene Operationen oder beide geprüft werden. Deshalb sollte gezielt ausgewählt werden, welche Ereignisse zu protokollieren sind.

Prüfrichtlinien

Der Sicherheitsadministrator kann mithilfe von Prüfrichtlinien das Prüfsystem so konfigurieren, dass nur Informationen zu den Daten und Objekten erfasst werden, die benötigt werden.

Der Sicherheitsadministrator kann Prüfrichtlinien erstellen, um zu steuern, welche Elemente in einer einzelnen Datenbank zu prüfen sind. Den folgenden Objekten kann eine Prüfrichtlinie zugeordnet werden:

- Gesamte Datenbank

  Alle prüfbaren Ereignisse, die in der Datenbank auftreten, werden der Prüfrichtlinie entsprechend protokolliert.
- Tabellen

  Alle Zugriffe durch die Datenbearbeitungssprache (DML) und XQUERY-Zugriffe auf die Tabelle (nicht typisiert), die MQT (Materialized Query Table) oder den Kurznamen werden protokolliert. Nur Prüfereignisse der Kategorie EXECUTE werden mit oder ohne Daten generiert, wenn auf die Tabelle zugegriffen wird, selbst wenn die Richtlinie angibt, dass andere Kategorien protokolliert werden sollen.
- Gesicherte Kontexte

  Alle prüfbaren Ereignisse, die in einer gesicherten Verbindung auftreten, die durch den bestimmten gesicherten Kontext definiert wird, werden der Prüfrichtlinie entsprechend protokolliert.
- Berechtigungs-IDs von Benutzern, Gruppen oder Rollen

  Alle prüfbaren Ereignisse, die von dem angegebenen Benutzer eingeleitet werden, werden der Prüfrichtlinie entsprechend protokolliert.

  Alle prüfbaren Ereignisse, die von Benutzern eingeleitet werden, die zu einer Gruppe oder Rolle gehören, werden der Prüfrichtlinie entsprechend protokolliert. Dies schließt indirekte Rollenzugehörigkeiten zum Beispiel durch andere Rollen und Gruppen mit ein.

  Sie können ähnliche Daten mithilfe der Workload Management-Ereignismonitore, erfassen, indem Sie eine Auslastung (Workload) für eine Gruppe definieren und die Aktivitätsdetails erfassen. Sie sollten sich jedoch darüber im Klaren sein, dass die Zuordnung von Auslastungen neben der Berechtigungs-ID auch auf Attribute zurückgreift, die dazu führen können, dass nicht die gewünschte Granularität beim Prüfen erzielt wird oder, falls diese anderen Attribute geändert werden, dass Verbindungen möglicherweise anderen (eventuell nicht überwachten) Auslastungen zugeordnet werden. Die Lösung der Prüffunktion bietet hingegen die Garantie, dass ein Benutzer, eine Gruppe oder eine Rolle geprüft wird.
- Berechtigungen (SYSADM, SECADM, DBADM, SYSCTRL, SYSMAINT, SYSMON)

  Alle prüfbaren Ereignisse, die von einem Benutzer eingeleitet werden, der die angegebene Berechtigung hat, selbst wenn diese Berechtigung für das Ereignis nicht erforderlich ist, werden der Prüfrichtlinie entsprechend protokolliert. Wenn eine Prüfrichtlinie einer Berechtigung DBADM zugeordnet wird, wird auch jeder Benutzer mit der Berechtigung SYSADM dieser Richtlinie entsprechend geprüft, weil solche Benutzer auch als Besitzer der Berechtigung DBADM betrachtet werden.

Der Sicherheitsadministrator kann mehrere Prüfrichtlinien erstellen. Zum Beispiel kann es für ein Unternehmen sinnvoll sein, eine Richtlinie zur Prüfung schutzwürdiger Daten und eine Richtlinie zur Prüfung der Aktivitäten von Benutzern mit der Berechtigung DBADM zu haben. Wenn mehrere Prüfrichtlinien für eine Anweisung verwendet werden, werden alle Ereignisse, die für jede der Prüfrichtlinien geprüft werden müssen, geprüft (jedoch nur einmal). Wenn zum Beispiel die Prüfrichtlinie der Datenbank eine Prüfung erfolgreicher EXECUTE-Ereignisse für eine bestimmte Tabelle erfordert und die Prüfrichtlinie des Benutzers eine Prüfung fehlgeschlagener EXECUTE-Ereignisse für dieselbe Tabelle erfordert, werden sowohl erfolgreiche als auch fehlgeschlagene Zugriffsversuche für diese Tabelle protokolliert.

Für ein bestimmtes Objekt kann nur eine Prüfrichtlinie in Kraft sein. Es ist zum Beispiel nicht möglich, mehrere Prüfrichtlinien gleichzeitig derselben Tabelle zuzuordnen.

Eine Prüfrichtlinie kann nicht einer Sicht oder einer typisierten Tabelle zugeordnet werden. Sichten, die auf eine Tabelle zugreifen, der eine Prüfrichtlinie zugeordnet ist, werden entsprechend der Prüfrichtlinie der zugrunde liegenden Tabelle geprüft.

Eine Prüfrichtlinie, die für eine Tabelle gilt, gilt nicht automatisch auch für eine MQT (Materialized Query Table), die auf dieser Tabelle basiert. Wenn Sie eine Prüfrichtlinie einer Tabelle zuordnen, sollten Sie dieselbe Richtlinie auch jeder MQT zuordnen, die auf dieser Tabelle basiert.

Eine Prüfung, die während einer Transaktion ausgeführt wird, erfolgt auf der Basis der Prüfrichtlinien und ihrer Zuordnungen zu Beginn der Transaktion. Wenn zum Beispiel der Sicherheitsadministrator eine Prüfrichtlinie einem Benutzer zuordnet und sich dieser Benutzer zu diesem Zeitpunkt in einer Transaktion befindet, wirkt sich die Prüfrichtlinie nicht auf die verbleibenden Anweisungen aus, die in dieser Transaktion ausgeführt werden. Auch Änderungen an einer Prüfrichtlinie treten erst in Kraft, wenn sie festgeschrieben (COMMIT) werden. Wenn ein Sicherheitsadministrator eine Anweisung ALTER AUDIT POLICY ausführt, wird sie erst wirksam, wenn die Anweisung mit COMMIT festgeschrieben wird.

Der Sicherheitsadministrator verwendet die Anweisung CREATE AUDIT POLICY zum Erstellen einer Prüfrichtlinie und die Anweisung ALTER AUDIT POLICY zum Ändern einer Prüfrichtlinie. In diesen Anweisungen können folgende Spezifikationen angegeben werden:

- Die Statuswerte für zu prüfende Ereignisse: NONE (Kein), SUCCESS (Erfolg), FAILURE (Fehlschlag) oder BOTH (Beides).

  Nur prüfbare Ereignisse, die dem angegebenen Statuswert entsprechen, werden protokolliert.
- Das Serververhalten, wenn Fehler während der Prüfung auftreten.

Der Sicherheitsadministrator verwendet die Anweisung AUDIT, um eine Prüfrichtlinie der aktuellen Datenbank oder einem Datenbankobjekt auf dem aktuellen Server zuzuordnen. Bei jeder Verwendung dieses Objekts wird es der zugeordneten Prüfrichtlinie entsprechend geprüft.

Der Sicherheitsadministrator verwendet die Anweisung DROP zum Löschen einer Prüfrichtlinie. Eine Prüfrichtlinie kann nicht gelöscht werden, wenn sie einem Objekt zugeordnet ist. Mit der Anweisung AUDIT REMOVE kann jede verbliebene Zuordnung zu einem Objekt entfernt werden. Wenn einer Prüfrichtlinie Metadaten hinzugefügt werden sollen, kann der Sicherheitsadministrator dazu die Anweisung COMMENT verwenden.

Vor Einrichtung einer vollständigen Verbindung generierte Ereignisse

Für einige Ereignisse, die während eines Verbindungsaufbaus und einer Benutzerwechseloperation generiert werden, sind nur die Informationen der Prüfrichtlinie verfügbar, die der Datenbank zugeordnet ist. Diese Ereignisse sind in der folgenden Tabelle aufgeführt:

Tabelle 4. Verbindungsereignisse

Ereignis	Prüfkategorie	Kommentar
CONNECT	CONTEXT	
CONNECT_RESET	CONTEXT	
AUTHENTICATION	VALIDATE	Dieses Ereignis erfasst die Authentifizierung sowohl beim Verbindungsaufbau als auch beim Benutzerwechsel in einer gesicherten Verbindung.
CHECKING_FUNC	CHECKING	Der versuchte Zugriff ist SWITCH_USER.

Diese Ereignisse werden nur auf der Basis der Prüfrichtlinie, die der Datenbank zugeordnet ist, und nicht durch Prüfrichtlinien, die anderen Objekten, wie Benutzern, deren Gruppen oder Berechtigungen, zugeordnet sind, protokolliert. Für die Ereignisse CONNECT und AUTHENTICATION, die während des Verbindungsaufbaus auftreten, werden die Prüfeinstellungen auf der Instanzebene verwendet, bis die Datenbank aktiviert wird. Die Datenbank wird aktiviert, wenn die erste Verbindung zu ihr hergestellt wird oder wenn der Befehl ACTIVATE DATABASE ausgeführt wird.

Wirkung des Benutzerwechsels

Wenn ein Benutzer innerhalb einer gesicherten Verbindung gewechselt wird, bleiben keine Spuren des ursprünglichen Benutzers zurück. In diesem Fall wird eine Prüfrichtlinie, die dem ursprünglichen Benutzer zugeordnet ist, nicht mehr berücksichtigt, und die anzuwendenden Prüfrichtlinien werden auf der Basis des neuen Benutzers erneut ausgewertet. Eine der gesicherten Verbindung zugeordnete Prüfrichtlinie bleibt weiterhin in Kraft.

Wenn die Anweisung SET SESSION USER verwendet wird, wird nur die Sitzungsberechtigungs-ID gewechselt. Die Prüfrichtlinie der Berechtigungs-ID des ursprünglichen Benutzers (die Systemberechtigungs-ID) bleibt in Kraft und die Prüfrichtlinie des neuen Benutzers wird zusätzlich verwendet. Wenn mehrere Anweisungen SET SESSION USER innerhalb einer Sitzung ausgeführt werden, werden nur die Prüfrichtlinie, die dem ursprünglichen Benutzer (die Systemberechtigungs-ID) zugeordnet ist, und die Prüfrichtlinie, die dem aktuellen Benutzer (die Sitzungsberechtigungs-ID) zugeordnet ist, berücksichtigt.

Einschränkungen für die Datendefinitionssprache (DDL)

Die folgenden Anweisungen der Datendefinitionssprache (DDL-Anweisungen) werden als exklusive SQL-Anweisungen der Prüffunktion (exklusive AUDIT-Anweisungen) bezeichnet:

- AUDIT
- CREATE AUDIT POLICY, ALTER AUDIT POLICY und DROP AUDIT POLICY
- DROP ROLE und DROP TRUSTED CONTEXT, wenn die Rolle bzw. der gesicherte Kontext, die bzw. der gelöscht wird, einer Prüfrichtlinie zugeordnet ist

Exklusive AUDIT-Anweisungen unterliegen in ihrer Verwendung bestimmten Einschränkungen:

- Auf jede Anweisung muss eine Anweisung COMMIT oder ROLLBACK folgen.
- Diese Anweisungen können nicht innerhalb einer globalen Transaktion, zum Beispiel einer XA-Transaktion, ausgeführt werden.

Nur eine nicht festgeschriebene exklusive DDL-Anweisung der Prüffunktion ist für alle Partitionen gleichzeitig zulässig. Wenn eine nicht festgeschriebene exklusive DDL-Anweisung der Prüffunktion ausgeführt wird, warten nachfolgende exklusive DDL-Anweisungen der Prüffunktion, bis die aktuelle Anweisung festgeschrieben (COMMIT) oder rückgängig gemacht (ROLLBACK) wird.

Anmerkung: Änderungen werden in den Katalog geschrieben, werden jedoch erst wirksam, wenn sie festgeschrieben werden. Dies gilt auch für die Verbindung, die die Anweisung ausführt.

Beispiel für das Prüfen eines beliebigen Zugriffs auf eine bestimmte Tabelle

Betrachten Sie zum Beispiel ein Unternehmen, bei dem die Tabelle EMPLOYEE extrem sensible Informationen enthält, sodass das Unternehmen sämtliche SQL-Zugriffe auf die Daten in dieser Tabelle prüfen möchte. Zur Verfolgung aller Zugriffe auf eine Tabelle kann die Kategorie EXECUTE verwendet werden. Sie prüft die SQL-Anweisung und optional den für diese Anweisung bei der Ausführung angegebenen Eingabedatenwert.

Die Verfolgung der Aktivitäten an der Tabelle erfolgt in zwei Schritten. Zunächst erstellt der Sicherheitsadministrator eine Prüfrichtlinie, die die Kategorie EXECUTE angibt. Anschließend ordnet der Sicherheitsadministrator diese Richtlinie der Tabelle zu:

```
CREATE AUDIT POLICY SENSITIVEDATAPOLICY
    CATEGORIES EXECUTE STATUS BOTH ERROR TYPE AUDIT
COMMIT

AUDIT TABLE EMPLOYEE USING POLICY SENSITIVEDATAPOLICY
COMMIT
```

Beispiel für das Prüfen beliebiger Aktionen durch SYSADM oder DBADM

Ein Unternehmen muss zur Erfüllung einer Sicherheitszertifizierung nachweisen, dass alle Aktivitäten in der Datenbank durch Personen, die die Berechtigung SYSADM (Systemverwaltung) oder DBADM (Datenbankverwaltung) besitzen, überwacht werden können.

Zur Erfassung aller Aktionen in der Datenbank sollten sowohl die Kategorie EXECUTE als auch die Kategorie SYSADMIN geprüft werden. Der Sicherheitsadministrator erstellt eine Prüfrichtlinie, die diese beiden Kategorie prüft. Der Sicherheitsadministrator kann diese Prüfrichtlinie mit der Anweisung AUDIT den Berechtigungen SYSADM und DBADM zuordnen. Für jeden Benutzer, der entweder über die Berechtigung SYSADM oder die Berechtigung DBADM verfügt, werden anschließend alle prüfbaren Ereignisse protokolliert. Das folgende Beispiel zeigt, wie eine solche Prüfrichtlinie erstellt und den Berechtigungen SYSADM und DBADM zugeordnet wird:

```
CREATE AUDIT POLICY ADMINSPOLICY CATEGORIES EXECUTE STATUS BOTH,
   SYSADMIN STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT SYSADM, DBADM USING POLICY ADMINSPOLICY
COMMIT
```

Beispiel für das Prüfen eines beliebigen Zugriffs durch eine bestimmte Rolle

In einem Unternehmen können die Webanwendungen des Unternehmens auf die Unternehmensdatenbank zugreifen. Die genauen Personen, die mit den Webanwendungen arbeiten, sind nicht bekannt. Bekannt sind nur die verwendeten Rollen, die daher zur Verwaltung der Datenbankberechtigungen verwendet werden. Das Unternehmen möchte die Aktionen aller Benutzer überwachen, die zu dieser Rolle gehören, um die Anforderungen zu untersuchen, die sie an die Datenbank übergeben. Dadurch soll sichergestellt werden, dass sie nur über die Webanwendungen auf die Datenbank zugreifen.

Die Kategorie EXECUTE enthält die erforderliche Prüfstufe, um die Aktivitäten der Benutzer in diesem Fall zu verfolgen. Der erste Schritt besteht darin, die entsprechende Prüfrichtlinie zu erstellen und sie den Rollen zuzuordnen, die von den Webanwendungen verwendet werden (in diesem Beispiel sind dies die Rollen TELLER und CLERK):

```
CREATE AUDIT POLICY WEBAPPPOLICY CATEGORIES EXECUTE WITH DATA
   STATUS BOTH ERROR TYPE AUDIT
COMMIT
AUDIT ROLE TELLER, ROLE CLERK USING POLICY WEBAPPPOLICY
COMMIT
```

Speicherung und Analyse von Prüfprotokollen

Der Systemadministrator kann den Pfad für das aktive Prüfprotokoll und das archivierte Prüfprotokoll mithilfe des Befehls db2audit configure konfigurieren. Die Archivierung des Prüfprotokolls versetzt das aktive Prüfprotokoll in ein Archivverzeichnis, während der Server gleichzeitig mit dem Schreiben in ein neues aktives Prüfprotokoll beginnt. Durch dieses Verfahren kann das Prüfprotokoll offline gespeichert werden, ohne die Daten extrahieren zu müssen. (Dies kann bei Bedarf geschehen.) Wenn der Sicherheitsadministrator oder der Systemadministrator ein Protokoll archiviert hat, können sie Daten aus dem Protokoll in Dateien mit begrenzter Satzlänge extrahieren. Die Daten in den Dateien mit begrenzter Satzlänge können zur Analyse in DB2-Datenbanktabellen geladen werden.

Die Konfiguration der Position für die Prüfprotokolle bietet die Möglichkeit, die Prüfprotokolle auf einem großen Hochgeschwindigkeitsdatenträger zu platzieren, wobei für jeden Knoten in einer DPF-Installation (DPF - Database Partitioning Feature, Datenbankpartitionierungsfunktion) optional separate Platten verwendet werden können. In einer DPF-Umgebung kann der Pfad für das aktive Prüfprotokoll ein Verzeichnis sein, das für jeden Knoten eindeutig ist. Ein eindeutiges Verzeichnis für jeden Knoten hilft bei der Vermeidung von Dateizugriffskonflikten, da jeder Knoten auf eine andere Platte schreibt.

Der Standardpfad für die Prüfprotokolle unter Windows-Betriebssystemen ist *instanz*\security\auditdata. Unter Linux- und UNIX-Betriebssystemen ist dieser Pfad *instanz*/security/auditdata. Wenn Sie die Standardposition nicht verwenden wollen, können Sie andere Verzeichnisse auswählen. (Sie können neue Verzeichnisse auf Ihrem System zur Verwendung als alternative Positionen erstellen, falls die gewünschten Verzeichnisse noch nicht vorhanden sind.) Zum Festlegen des Pfads der Position für das aktive Prüfprotokoll und der Position für die archivierten Prüfprotokolle verwenden Sie den Befehl db2audit configure mit den Parametern datapath und archivepath wie im folgenden Beispiel:

```
db2audit configure datapath /auditlog archivepath /auditarchive
```

Die Speicherpositionen für Prüfprotokolle, die Sie mit dem Befehl db2audit festlegen, gelten für alle Datenbanken in der Instanz.

Anmerkung: Wenn mehrere Instanzen auf dem Server vorhanden sind, sollte jede Instanz separate Daten- und Archivpfade haben.

Pfad für aktive Prüfprotokolle (datapath) in einer DPF-Umgebung

In einer DPF-Umgebung muss in jeder Partition die gleiche Position für das aktive Prüfprotokoll (durch den Parameter **datapath** festgelegt) verwendet werden. Dies lässt sich mit zwei Methoden erreichen:

1. Verwenden Sie Datenbankpartitionsausdrücke, wenn Sie den Parameter **datapath** angeben. Durch die Verwendung von Datenbankpartitionsausdrücken kann die Partitionsnummer in den Pfad für die Prüfprotokolldateien eingefügt werden, sodass sich in jeder Datenbankpartition ein anderer Pfad angeben lässt.
2. Verwenden Sie ein gemeinsam genutztes Laufwerk, das auf allen Knoten identisch ist.

Sie können Datenbankpartitionsausdrücke an einer beliebigen Stelle innerhalb des Werts verwenden, den Sie für den Parameter **datapath** angeben. Auf einem System mit drei Knoten, auf dem die Datenbankpartitionsnummer 10 ist, kann zum Beispiel der folgende Befehl verwendet werden:

```
db2audit configure datapath '/pfadFuerKnoten $N'
```

Dieser Befehl erstellt die folgenden Dateien:

- /pfadFuerKnoten10
- /pfadFuerKnoten20
- /pfadFuerKnoten30

Anmerkung: Zur Angabe des Archivpfads (Parameter **archivepath**) für Protokolldateien können keine Datenbankpartitionsausdrücke verwendet werden.

Archivieren aktiver Prüfprotokolle

Der Systemadministrator kann das Tool db2audit zum Archivieren von Prüfprotokollen sowohl der Instanz als auch der Datenbank sowie zum Extrahieren von Prüfdaten aus den archivierten Protokollen beider Typen verwenden. Zum Archivieren des aktiven Prüfprotokolls kann der Sicherheitsadministrator die gespeicherte Prozedur SYSPROC.AUDIT_ARCHIVE verwenden. Zum Extrahieren von Daten aus dem Protokoll und zum Laden dieser Daten in Dateien mit begrenzter Satzlänge kann der Sicherheitsadministrator die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT verwenden.

Zum Archivieren der Prüfprotokolle und zum Extrahieren von Daten aus den Prüfprotokollen muss der Sicherheitsadministrator die folgenden Schritte ausführen:

1. Er muss eine Anwendung terminieren, die regelmäßige Archivierungen des aktiven Prüfprotokolls mithilfe der gespeicherten Prozedur SYSPROC.AUDIT_ARCHIVE ausführt.
2. Er muss feststellen, welche archivierten Protokolldateien von Interesse sind. Dabei kann er die Tabellenfunktion SYSPROC.AUDIT_LIST_LOGS verwenden, um alle archivierten Prüfprotokolle aufzulisten.
3. Er muss den ermittelten Dateinamen als Parameter an die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT übergeben, um Daten aus dem Protokoll zu extrahieren und in Dateien mit begrenzter Satzlänge zu laden.

4. Er muss die Prüfdaten zur Analyse in DB2-Datenbanktabellen laden.

Die archivierten Protokolldateien müssen nicht sofort zur Analyse in Tabellen geladen werden. Sie können für eine spätere Analyse gespeichert werden. Zum Beispiel ist es möglich, dass sie überhaupt nur untersucht werden, wenn eine Unternehmensprüfung stattfindet.

Falls während der Archivierung ein Problem auftritt, zum Beispiel, wenn der Plattenspeicherplatz im Archivpfad nicht ausreicht oder der Archivpfad nicht vorhanden ist, schlägt der Archivierungsprozess fehl. In diesem Fall wird eine vorläufige Protokolldatei mit der Dateierweiterung .bk im Datenpfad für das Prüfprotokoll generiert. Beispiel: db2audit.instance.log.0.20070508172043640941.bk. Wenn das Problem beseitigt ist (durch Zuordnen eines ausreichenden Plattenspeicherbereichs im Archivpfad bzw. durch Erstellen des Archivpfads) müssen Sie diese vorläufige Protokolldatei in den Archivpfad versetzen. Anschließend können Sie diese Datei genauso wie ein erfolgreich archiviertes Protokoll behandeln.

Archivieren aktiver Prüfprotokolle in einer DPF-Umgebung

Wenn in einer DPF-Umgebung der Archivierungsbefehl ausgeführt wird, während die Instanz aktiv ist, wird der Archivierungsprozess automatisch auf jedem Knoten ausgeführt. Auf allen Knoten wird dieselbe Zeitmarke im Dateinamen der archivierten Protokolldatei verwendet. Auf einem System mit drei Knoten, auf dem die Datenbankpartitionsnummer 10 ist, kann zum Beispiel der folgende Befehl verwendet werden:

```
db2audit archive to /auditarchive
```

Dieser Befehl erstellt die folgenden Dateien:

- /auditarchive/db2audit.log.10.*zeitmarke*
- /auditarchive/db2audit.log.20.*zeitmarke*
- /auditarchive/db2audit.log.30.*zeitmarke*

Wenn der Archivierungsbefehl ausgeführt wird, wenn die Instanz nicht aktiv ist, können Sie mit einer der folgenden Methoden steuern, auf welchem Knoten die Archivierung ausgeführt wird:

- Sie können die Option node im Befehl db2audit verwenden, um die Archivierung nur für den aktuellen Knoten auszuführen.
- Sie können den Befehl db2_all verwenden, um die Archivierung auf allen Knoten auszuführen.

  Beispiel:

  ```
  db2_all db2audit archive node to /auditarchive
  ```

  Dieser Befehl definiert die Umgebungsvariable DB2NODE, um anzugeben, auf welchen Knoten der Befehl aufgerufen wird.

Alternativ können Sie auf jedem Knoten einen separaten Archivierungsbefehl absetzen. Beispiel:

- Auf Knoten 10:

  ```
  db2audit archive node 10 to /auditarchive
  ```

- Auf Knoten 20:

  ```
  db2audit archive node 20 to /auditarchive
  ```

- Auf Knoten 30:

  ```
  db2audit archive node 30 to /auditarchive
  ```

Anmerkung: Wenn die Instanz nicht aktiv ist, werden in die Dateinamen der archivierten Prüfprotokolle nicht dieselben Zeitmarken auf allen Knoten eingefügt.

Anmerkung: Es wird empfohlen, denselben Archivpfad über alle Knoten hinweg gemeinsam zu nutzen, jedoch ist dies keine Voraussetzung.

Anmerkung: Die gespeicherte Prozedur AUDIT_DELIM_EXTRACT und die Tabellenfunktion AUDIT_LIST_LOGS können nur auf die archivierten Protokolldateien zugreifen, die vom aktuellen Knoten (Koordinatorknoten) aus sichtbar sind.

Beispiel für das Archivieren eines Protokolls und das Extrahieren von Daten in eine Tabelle

Zur Sicherstellung, dass die Prüfdaten erfasst und zur zukünftigen Verwendung gespeichert werden, muss ein Unternehmen alle sechs Stunden ein neues Prüfprotokoll erstellen und das aktuelle Prüfprotokoll auf einem Laufwerk mit WORM-Funktionalität archivieren. Das Unternehmen terminiert den folgenden Aufruf an die gespeicherte Prozedur SYSPROC.AUDIT_ARCHIVE, die vom Sicherheitsadministrator alle sechs Stunden auszuführen ist. Der Pfad zu den archivierten Protokollen ist der Standardarchivpfad /auditarchive, und die Archivierung wird auf allen Knoten ausgeführt:

```
CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
```

Im Rahmen seiner Sicherheitsprozeduren hat das Unternehmen eine Reihe verdächtiger Verhaltensweisen bzw. unzulässiger Aktivitäten erkannt und definiert, nach denen in den Prüfdaten zu suchen ist. Das Verfahren sieht vor, alle Daten aus einem oder mehreren Prüfprotokollen zu extrahieren, diese Daten in eine relationale Tabelle zu laden und anschließend mithilfe von SQL-Abfragen nach solchen Aktivitäten zu suchen. Das Unternehmen hat die geeigneten Kategorien für die Prüfung festgelegt und der Datenbank oder anderen Datenbankobjekten die erforderlichen Prüfrichtlinien zugeordnet.

Nun kann zum Beispiel die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT aufgerufen werden, um die Daten aus den archivierten Prüfprotokollen, die mit einer Zeitmarke vom April 2006 erstellt wurden, für alle Kategorien aus allen Knoten unter Verwendung des Standardbegrenzers zu extrahieren:

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT(
     '', '', '/auditarchive',  'db2audit.%.200604%', '' )
```

In einem weiteren Beispiel kann die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT aufgerufen werden, um die archivierten Prüfsätze mit erfolgreichen Ereignissen der Kategorie EXECUTE und fehlgeschlagenen Ereignissen der Kategorie CHECKING aus einer Datei mit der gewünschten Zeitmarke zu extrahieren:

```
CALL SYSPROC.AUDIT_DELIM_EXTRACT( '', '', '/auditarchive',
    'db2audit.%.20060419034937', 'categories
     execute status success, checking status failure );
```

Namen der Prüfprotokolldateien:

Die Prüfprotokolldateien haben Namen, die zu erkennen geben, ob sie Prüfprotokolle auf Instanzebene oder Datenbankebene sind und aus welcher Partition in einer DPF-Umgebung (Database Partitioning Feature, Datenbankpartitionierungs-

funktion) sie stammen. Archivierte Prüfprotokolle haben die an ihren Dateinamen angehängte Zeitmarke für den Zeitpunkt, zu dem der Archivierungsbefehl ausgeführt wurde.

Dateinamen für das aktive Prüfprotokoll

In einer DPF-Umgebung kann der Pfad für das aktive Prüfprotokoll ein Verzeichnis sein, das für jede Partition eindeutig ist, sodass jede Partition in eine eigene Datei schreibt. Um die Herkunft der Prüfsätze exakt verfolgen zu können, wird die Partitionsnummer in die Dateinamen der Prüfprotokolle eingefügt. Für Partition 20 lautet der Dateiname des Prüfprotokolls auf Instanzebene zum Beispiel db2audit.instance.log.20. Für eine Datenbank mit dem Namen 'testdb' heißt die Prüfprotokolldatei db2audit.db.testdb.log.20.

In einer Nicht-DPF-Umgebung wird als Partitionsnummer der Wert 0 (null) angenommen. In diesem Fall lautet der Prüfprotokolldateiname auf Instanzebene db2audit.instance.log.0. Für eine Datenbank mit dem Namen 'testdb' in dieser Instanz ist der Name des Prüfprotokolls db2audit.db.testdb.log.0.

Dateinamen für archivierte Prüfprotokolle

Wenn das aktive Prüfprotokoll archiviert wird, wird die aktuelle Zeitmarke im folgenden Format an den Dateinamen angehängt: *JJJJMMTTHHMMSS*. (Dabei geben JJJJ das Jahr, MM den Monat, TT den Tag, HH die Stunde, MM die Minuten und SS die Sekunden an.)

Das Dateinamenformat für ein archiviertes Prüfprotokoll hängt von der Ebene des Prüfprotokolls ab:

Archiviertes Prüfprotokoll auf Instanzebene
Der Dateiname für das archivierte Prüfprotokoll auf Instanzebene ist: db2audit.instance.log.*partition.JJJJMMTTHHMMSS*.

Archiviertes Prüfprotokoll auf Datenbankebene
Der Dateiname für das archivierte Prüfprotokoll auf Datenbankebene ist: db2audit.db*datenbank*.log.*partition.JJJJMMTTHHMMSS*.

In einer Nicht-DPF-Umgebung hat *partition* den Wert 0 (null).

Die Zeitmarke stellt den Zeitpunkt dar, zu dem der Archivierungsbefehl ausgeführt wurde. Daher ist dieser Zeitpunkt nicht immer mit dem Zeitpunkt des letzten Prüfsatzes im Protokoll identisch. Die archivierte Prüfprotokolldatei kann Prüfsätze mit Zeitmarken enthalten, die einen um einige Sekunden späteren Zeitpunkt angeben als die Zeitmarke im Namen der Protokolldatei. Dies kann folgende Gründe haben:

- Wenn der Archivierungsbefehl abgesetzt wird, wartet die Prüffunktion darauf, dass laufende Schreiboperationen für Prüfsätze abgeschlossen werden, bevor die archivierte Protokolldatei erstellt wird.
- In einer Umgebung mit mehreren Systemen ist möglicherweise die Systemzeit auf einem fernen System nicht mit der Systemzeit des Systems synchron, auf dem der Archivierungsbefehl abgesetzt wird.

Wenn in einer DPF-Umgebung der Server aktiv ist, wenn die Archivierung ausgeführt wird, ist die Zeitmarke über alle Partitionen hinweg konsistent und stellt die Zeitmarke dar, die in der Partition generiert wurde, in der die Archivierung ausgeführt wurde.

Erstellen von Tabellen zur Aufnahme der DB2-Prüfdaten:

Bevor Sie mit Prüfdaten in Datenbanktabellen arbeiten können, müssen Sie die Tabellen erstellen, in denen die Daten gespeichert werden sollen. Sie sollten in Betracht ziehen, diese Tabelle in einem separaten Schema zu erstellen, um die Daten in den Tabellen von unbefugten Benutzern zu isolieren.

- Die zur Erstellung eines Schemas erforderlichen Berechtigungen und Zugriffsrechte finden Sie in den Informationen zur Anweisung CREATE SCHEMA.
- Die zur Erstellung einer Tabelle erforderlichen Berechtigungen und Zugriffsrechte finden Sie in den Informationen zur Anweisung CREATE TABLE.
- Legen Sie fest, welcher Tabellenbereich zum Speichern der Tabellen verwendet werden soll. (Dieser Abschnitt behandelt keine Erstellung von Tabellenbereichen.)

Anmerkung: Das Format der Tabellen, die Sie zur Aufnahme der Prüfdaten erstellen müssen, kann sich von Release zu Release ändern. Möglicherweise werden neue Spalten hinzugefügt oder die Größe einer vorhandenen Spalte geändert. Das Script db2audit.ddl erstellt Tabellen im korrekten Format, um die Prüfsätze aufzunehmen.

In den folgenden Beispielen wird gezeigt, wie die Tabellen erstellt werden, die die Datensätze aus den Dateien mit begrenzter Satzlänge aufnehmen können. Falls erwünscht, können Sie ein separates Schema zum Speichern dieser Tabellen erstellen.

Wenn Sie nicht alle Daten verwenden wollen, die in den Dateien enthalten sind, können Sie je nach Bedarf Spalten aus den Tabellendefinitionen weglassen oder die Erstellung bestimmter Tabellen übergehen. Wenn Sie Spalten aus den Tabellendefinitionen weglassen, müssen Sie die Befehle modifizieren, die Sie zum Laden von Daten in diese Tabellen verwenden.

1. Führen Sie den Befehl db2 aus, um ein DB2-Befehlsfenster zu öffnen.
2. Optional. Erstellen Sie ein Schema zum Speichern der Tabellen. In diesem Beispiel erhält das Schema den Namen AUDIT:
   ```
   CREATE SCHEMA AUDIT
   ```
3. Optional. Wenn Sie das Schema AUDIT erstellt haben, wechseln Sie zu diesem Schema, bevor Sie Tabellen erstellen:
   ```
   SET CURRENT SCHEMA = 'AUDIT'
   ```
4. Führen Sie das Script db2audit.ddl aus, um die Tabellen zu erstellen, die die Prüfsätze aufnehmen sollen.

   Das Script db2audit.ddl befindet sich im Verzeichnis sqllib/misc (sqllib\misc unter Windows). Das Script setzt voraus, dass eine Verbindung zu der Datenbank besteht und dass ein 8K-Tabellenbereich verfügbar ist. Der Befehl zum Ausführen des Scripts sieht wie folgt aus: db2 +o -tf sqllib/misc/db2audit.ddl. Die folgenden Tabellen werden durch das Script erstellt: AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN, VALIDATE, CONTEXT und EXECUTE.
5. Nach der Erstellung der Tabellen kann der Sicherheitsadministrator die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT bzw. der Systemadministrator den Befehl db2audit extract verwenden, um die Prüfsätze aus den archivierten Prüfprotokolldateien in die Dateien mit begrenzter Satzlänge zu extrahieren. Sie können die Prüfdaten aus den Dateien mit begrenzter Satzlänge in die Datenbanktabellen laden, die Sie zuvor erstellt haben.

Laden von DB2-Prüfdaten in Tabellen:

Nachdem Sie die Prüfprotokolldatei archiviert und in Dateien mit begrenzter Satzlänge extrahiert haben und die Datenbanktabellen zur Aufnahme der Prüfdaten erstellt wurden, können Sie die Prüfdaten aus den Dateien mit begrenzter Satzlänge zur Analyse in die Datenbanktabellen laden.

Die Prüfdaten werden mit dem Dienstprogramm LOAD in die Tabellen geladen. Führen Sie für jede Tabelle einen getrennten LOAD-Befehl aus. Wenn Sie eine oder mehrere Spalten aus den Tabellendefinitionen weggelassen haben, müssen Sie die Version des LOAD-Befehls modifizieren, um die Daten erfolgreich zu laden. Wenn Sie ein anderes Begrenzungszeichen als das Standardzeichen beim Extrahieren der Prüfdaten angegeben haben, müssen Sie die von Ihnen verwendete Version des LOAD-Befehls ebenfalls modifizieren.

1. Führen Sie den Befehl db2 aus, um ein DB2-Befehlsfenster zu öffnen.
2. Zum Laden von Daten in die Tabelle AUDIT führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM audit.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.AUDIT
   ```

   Anmerkung: Geben Sie den Modifikator DELPRIORITYCHAR an, um ein ordnungsgemäßes Parsing der Binärdaten sicherzustellen.

   Anmerkung: Geben Sie die Option LOBSINFILE des Befehls LOAD an (wegen der Einschränkung, dass integrierte Daten für große Objekte auf 32 K begrenzt werden müssen). In einen Fällen kann es auch erforderlich sein, die Option LOBS FROM zu verwenden.

   Anmerkung: Verwenden Sie zur Angabe des Dateinamens den vollständig qualifizierten Pfadnamen. Wenn das DB2-Datenbanksystem zum Beispiel auf dem Laufwerk C: eines Windows-basierten Computers installiert ist, müsste `C:\Program Files\IBM\SQLLIB\instanz\security\audit.del` als vollständig qualifizierter Dateiname für die Datei audit.del angegeben werden.
3. Zum Laden von Daten in die Tabelle CHECKING führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM checking.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.CHECKING
   ```
4. Zum Laden von Daten in die Tabelle OBJMAINT führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM objmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.OBJMAINT
   ```
5. Zum Laden von Daten in die Tabelle SECMAINT führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM secmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.SECMAINT
   ```
6. Zum Laden von Daten in die Tabelle SYSADMIN führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM sysadmin.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.SYSADMIN
   ```
7. Zum Laden von Daten in die Tabelle VALIDATE führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM validate.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.VALIDATE
   ```

8. Zum Laden von Daten in die Tabelle CONTEXT führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM context.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.CONTEXT
   ```

9. Zum Laden von Daten in die Tabelle EXECUTE führen Sie den folgenden Befehl aus:

   ```
   LOAD FROM execute.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
            INSERT INTO schema.EXECUTE
   ```

10. Nach Abschluss des Ladens der Daten in die Tabellen, löschen Sie die .del-Dateien aus dem Unterverzeichnis security/auditdata des Verzeichnisses sql-lib.
11. Wenn Sie die Prüfdaten in die Tabellen geladen haben, können Sie die Daten aus diesen Tabellen zur Analyse auswählen.

Wenn Sie die Tabellen bereits ein erstes Mal gefüllt haben und dies wiederholen wollen, verwenden Sie die Option INSERT, um die neuen Tabellendaten den vorhandenen Tabellendaten hinzuzufügen. Wenn Sie die Datensätze der vorigen Operation db2audit extract aus den Tabellen entfernen wollen, laden Sie die neuen Daten mit der Option REPLACE in die Tabellen.

Gespeicherte Prozeduren zum Archivieren und Extrahieren von Prüfprotokollen:

Der Sicherheitsadministrator kann mithilfe der gespeicherten Prozeduren SYSPROC.AUDIT_ARCHIVE und SYSPROC.AUDIT_DELIM_EXTRACT sowie mit der Tabellenfunktion SYSPROC.AUDIT_LIST_LOGS Prüfprotokolle der Datenbank, mit der er momentan verbunden ist, archivieren und Daten aus den Prüfprotokollen in Dateien mit begrenzter Satzlänge extrahieren.

Der Sicherheitsadministrator muss mit einer Datenbank verbunden sein, um die Prüfprotokolle der Datenbank mit diesen gespeicherten Prozeduren und der Tabellenfunktion zu archivieren bzw. aufzulisten.

Wenn Sie die archivierten Dateien in ein anderes Datenbanksystem kopieren und die gespeicherten Prozeduren für den Zugriff auf sie verwenden wollen, müssen Sie sicherstellen, dass der Datenbankname identisch ist oder die Dateien umbenennen, sodass sie den gleichen Datenbanknamen enthalten.

Diese gespeicherten Prozeduren und diese Tabellenfunktion dienen nicht zum Archivieren bzw. Auflisten des Prüfprotokolls der Instanzebene. Das Prüfprotokoll der Instanzebene muss vom Systemadministrator mit dem Befehl db2audit archiviert und extrahiert werden.

Der Sicherheitsadministrator kann diese gespeicherten Prozeduren und diese Tabellenfunktion zur Ausführung der folgenden Operationen verwenden:

Tabelle 5. Gespeicherte Prozeduren für das Prüfsystem

Gespeicherte Prozedur bzw. Tabellenfunktion	**Operation**	**Kommentare**
AUDIT_ARCHIVE	Archiviert das aktuelle Prüfprotokoll.	Akzeptiert den Archivpfad als Eingabe. Wenn der Archivpfad nicht angegeben wird, entnimmt diese gespeicherte Prozedur den Archivpfad der Prüfkonfigurationsdatei. Die Archivierung wird auf jedem Knoten ausgeführt. An den Namen der Prüfprotokolldatei wird eine synchronisierte Zeitmarke angehängt.
AUDIT_LIST_LOGS	Gibt eine Liste der archivierten Prüfprotokolle im angegebenen Pfad für die aktuelle Datenbank zurück.	

Tabelle 5. Gespeicherte Prozeduren für das Prüfsystem (Forts.)

Gespeicherte Prozedur bzw. Tabellenfunktion	Operation	Kommentare
AUDIT_ DELIM_EXTRACT	Extrahiert Daten aus den archivierten Binärprotokollen und lädt sie in Dateien mit begrenzter Satzlänge.	Die extrahierten Prüfsätze werden ein begrenztes Format umgesetzt, das sich zum Laden in DB2-Datenbanktabellen eignet. Die Ausgabe wird in separaten Dateien, jeweils eine pro Kategorie, erstellt. Darüber hinaus wird die Datei **auditlobs** erstellt, in der alle großen Objekte (LOBs) gespeichert werden, die in den Prüfdaten enthalten sind. Die Dateien haben die folgenden Dateinamen: • **audit.del** • **checking.del** • **objmaint.del** • **secmaint.del** • **sysadmin.del** • **validate.del** • **context.del** • **execute.del** • **auditlobs** Wenn die Dateien bereits vorhanden sind, wird die Ausgabe an sie angehängt. Die Datei **auditlobs** wird erstellt, wenn Daten der Kategorien CONTEXT oder EXECUTE extrahiert werden. Es können nur Daten der archivierten Prüfprotokolle für die aktuelle Datenbank extrahiert werden. Daten werden nur aus Dateien extrahiert, die für den Koordinatorknoten sichtbar sind. Nur der Instanzeigner kann archivierte Prüfprotokolle löschen.

Kategorie EXECUTE zum Prüfen von SQL-Anweisungen

Die Kategorie EXECUTE bietet die Möglichkeit, die von einem Benutzer ausgeführten SQL-Anweisungen präzise zu verfolgen. (Vor Version 9.5 musste die Kategorie CONTEXT zur Erfassung solcher Informationen verwendet werden.)

Die Kategorie EXECUTE erfasst den SQL-Anweisungstext sowie die Kompilierungsumgebung und andere Werte, die erforderlich sind, um die Anweisung zu einem späteren Zeitpunkt zu wiederholen. Zum Beispiel kann eine Wiederholung der Anweisung zeigen, welche Zeilen genau durch eine Anweisung SELECT zurückgegeben wurden.

Zur Wiederholung einer Anweisung müssen die Datenbanktabellen zunächst in den Status zurückversetzt werden, den sie hatten, als die Anweisung ursprünglich ausgeführt wurde.

Wenn Sie eine Prüfung mit der Kategorie EXECUTE ausführen, wird der Anweisungstext für statisches und dynamisches SQL aufgezeichnet, ebenso wie Eingabeparametermarken und Hostvariablen. Sie können die Kategorie EXECUTE zur Prüfung mit oder ohne Eingabewerte konfigurieren.

Anmerkung: Globale Variablen werden nicht geprüft.

Die Prüfung von EXECUTE-Ereignissen erfolgt am Ende des Ereignisses (bei SELECT-Anweisungen ist dies nach dem Schließen des Cursors). Der Status, mit dem das Ereignis abgeschlossen wurde, wird ebenfalls gespeichert. Da EXECUTE-Ereignisse bei Abschluss geprüft werden, werden Abfragen mit langer Laufzeit nicht sofort in das Prüfprotokoll eingetragen.

Anmerkung: Die Vorbereitung einer Anweisung wird nicht als Teil der Ausführung betrachtet. Die meisten Berechtigungsprüfungen werden bei der Vorbereitung durchgeführt (z. B. für das Zugriffsrecht SELECT). Dies bedeutet, dass Anweisungen, die während der Vorbereitung aufgrund von Berechtigungsfehlern fehlschlagen, keine EXECUTE-Ereignisse generieren.

Die Felder 'Statement Value Index' (Anweisungswertindex), 'Statement Value Type' (Anweisungswerttyp) und 'Statement Value Data' (Anweisungswertdaten) können für einen gegebenen EXECUTE-Prüfsatz mehrfach erfasst werden. Bei dem durch die Extraktion generierten Berichtsformat listet jeder Prüfsatz mehrere Werte auf. Beim Dateiformat mit begrenzter Satzlänge werden mehrere Zeilen verwendet. Die erste Zeile enthält den Ereignistyp STATEMENT und keine Werte. Die nachfolgenden Zeilen enthalten den Ereignistyp DATA, wobei für jeden Datenwert, der der SQL-Anweisung zugeordnet ist, eine Zeile verwendet wird. Über die Felder für den Ereigniskorrelator und die Anwendungs-ID können Sie STATEMENT-Zeilen und DATA-Zeilen korrelieren. Die Spalten 'Statement Text' (Anweisungstext), 'Statement Isolation Level' (Isolationsstufe der Anweisung) und 'Compilation Environment Description' (Beschreibung der Kompilierungsumgebung) sind in den DATA-Ereignissen nicht enthalten.

Der Anweisungstext und die Eingabedatenwerte, die geprüft werden, werden in die Codepage der Datenbank konvertiert, wenn sie auf Platte gespeichert werden. (Alle geprüften Felder werden in der Codepage der Datenbank gespeichert.) Es wird kein Fehler zurückgemeldet, wenn die Codepage der Eingabedaten nicht mit der Codepage der Datenbank kompatibel ist. Stattdessen werden die nicht konvertierten Daten protokolliert. Da jede Datenbank über ein eigenes Prüfprotokoll verfügt, stellen Datenbanken mit unterschiedlichen Codepages kein Problem dar.

Die Anweisungen ROLLBACK und COMMIT werden geprüft, wenn sie von der Anwendung ausgeführt werden und ebenso wenn sie implizit als Teil eines anderen Befehls, zum Beispiel BIND, ausgeführt werden.

Wenn ein EXECUTE-Ereignis aufgrund eines Zugriffs auf eine geprüfte Tabelle protokolliert wurde, werden alle Anweisungen geprüft, die sich darauf auswirken, welche anderen Anweisungen innerhalb einer UOW ausgeführt werden. Diese Anweisungen sind COMMIT, ROLLBACK, ROLLBACK TO SAVEPOINT und SAVEPOINT.

Feld 'Savepoint ID' (Sicherungspunkt-ID)

Mithilfe des Feldes 'Savepoint ID' (Sicherungspunkt-ID) können Sie verfolgen, welche Anweisungen von einer Anweisung ROLLBACK TO SAVEPOINT betroffen waren. Bei einer normalen DML-Anweisung (z. B. SELECT, INSERT usw.) wird die aktuelle Sicherungspunkt-ID protokolliert. Bei der Anweisung ROLLBACK TO SAVEPOINT wird hingegen die Sicherungspunkt-ID protokolliert, bis zu der die ROLLBACK-Operation ausgeführt wird. Das heißt, jede Anweisung mit einer Sicherungspunkt-ID, die größer oder gleich dieser ID ist, wird durch ROLLBACK rückgängig gemacht, wie im folgenden Beispiel veranschaulicht. Die Tabelle zeigt die Abfolge der ausgeführten Anweisungen. Alle Ereignisse mit einer Sicherungspunkt-ID größer oder gleich 2 werden rückgängig gemacht. Nur der Wert 3 (aus der ersten Anweisung INSERT) wird in die Tabelle T1 eingefügt.

Tabelle 6. Abfolge von Anweisungen zur Veranschaulichung der Wirkung einer Anweisung ROLLBACK TO SAVEPOINT

Anweisung	Sicherungspunkt-ID
INSERT INTO T1 VALUES (3)	1
SAVEPOINT A	2
INSERT INTO T1 VALUES (5)	2
SAVEPOINT B	3
INSERT INTO T1 VALUES (6)	3
ROLLBACK TO SAVEPOINT A	2
COMMIT	

Option WITH DATA

Nicht alle Eingabewerte werden geprüft, wenn Sie die Option WITH DATA angeben. LOB-, LONG- und XML-Parameter sowie Parameter strukturierter Typen werden als NULL erfasst.

Datums-, Zeit- und Zeitmarkenfelder werden im ISO-Format aufgezeichnet.

Wenn in einer Richtlinie, die Objekten zugeordnet ist, die von der Ausführung der SQL-Anweisung betroffen sind, die Option WITH DATA angegeben wird, und in einer anderen, ebensolchen Objekten zugeordneten Richtlinie die Option WITHOUT DATA angegeben wird, erhält die Option WITH DATA Vorrang, sodass für diese bestimmte Anweisung Daten erfasst werden. Wenn zum Beispiel die einem Benutzer zugeordnete Prüfrichtlinie die Option WITHOUT DATA angibt, die einer Tabelle zugeordnete Prüfrichtlinie jedoch die Option WITH DATA, werden bei einem Zugriff dieses Benutzers auf diese Tabelle die für die Anweisung verwendeten Eingabedaten protokolliert.

Sie haben keine Möglichkeit, die Zeilen zu ermitteln, die durch eine Anweisung zur positionierten Aktualisierung oder positionierten Löschung geändert wurden. Es wird nur die Ausführung der zugrunde liegenden Anweisung SELECT protokolliert, nicht jedoch die einzelne FETCH-Operation. Es ist nicht möglich, anhand eines EXECUTE-Prüfsatzes festzustellen, auf welcher Zeile sich der Cursor befand, als die Anweisung ausgeführt wurde. Wenn die Anweisung zu einem späteren Zeitpunkt wiederholt wird, ist es nur möglich, die Anweisung SELECT auszuführen, um festzustellen, welcher Bereich von Zeilen möglicherweise betroffen war.

Beispiel für die Wiederholung früherer Aktivitäten

Betrachten Sie das folgende Beispiel: Im Rahmen einer umfassenden Sicherheitsrichtlinie ist es für ein Unternehmen erforderlich, die Möglichkeit zu behalten, bis zu sieben Jahren rückwirkend die Auswirkungen einer bestimmten Anforderung für bestimmte Tabellen in der Datenbank zu analysieren. Zu diesem Zweck wird eine Richtlinie zur Archivierung der wöchentlichen Backups und der zugehörigen Protokolldateien eingerichtet, sodass sich die Datenbank für jeden ausgewählten Zeitpunkt rekonstruieren lässt. Es ist erforderlich, dass die Datenbankprüfung ausreichend Informationen zu jeder Anforderung erfasst, die an der Datenbank ausgeführt wird, sodass es möglich ist, jede beliebige Anforderung an der relevanten, für den Zeitpunkt wiederhergestellten Datenbank zu wiederholen. Diese Voraussetzung schließt statische und dynamische SQL-Anweisungen mit ein.

Dieses Beispiel zeigt die Prüfrichtlinie, die eingerichtet sein muss, wenn die SQL-Anweisung ausgeführt wird, und die Schritte, die zur Archivierung der Prüfprotokolle sowie zur späteren Extraktion und Analyse von Daten erforderlich sind.

1. Es muss eine Prüfrichtlinie mit der Kategorie EXECUTE erstellt und der Datenbank zugeordnet werden:

   ```
   CREATE AUDIT POLICY STATEMENTS CATEGORIES EXECUTE WITH DATA
      STATUS BOTH ERROR TYPE AUDIT
   COMMIT

   AUDIT DATABASE USING POLICY STATEMENTS
   COMMIT
   ```

2. Das Prüfprotokoll muss regelmäßig archiviert werden, um eine Archivkopie zu erstellen.

   Die folgende Anweisung sollte vom Sicherheitsadministrator regelmäßig ausgeführt werden, zum Beispiel einmal pro Woche oder einmal pro Tag, je nachdem, wie viel Daten protokolliert werden. Diese archivierten Dateien können nach Bedarf über einen beliebig langen Zeitraum aufbewahrt werden. Die Prozedur SYSPROC.AUDIT_ARCHIVE wird mit zwei Eingabeparametern aufgerufen: dem Pfad zum Archivverzeichnis und dem Wert '-2', um anzugeben, dass die Archivierung auf allen Knoten ausgeführt werden soll:

   ```
   CALL SYSPROC.AUDIT_ARCHIVE( '/auditarchive', -2 )
   ```

3. Der Sicherheitsadministrator verwendet die Tabellenfunktion SYSPROC.AUDIT_LIST_LOGS, um alle verfügbaren Prüfprotokolle zum Beispiel von April 2006 zu untersuchen und festzustellen, welche Protokolle die benötigten Daten enthalten:

   ```
   SELECT FILE FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('/auditarchive'))
      AS T WHERE FILE LIKE 'db2audit.dbname.log.0.200604%'
   FILE
   ------------------------------------
   ...
   db2audit.dbname.log.0.20060418235612
   db2audit.dbname.log.0.20060419234937
   db2audit.dbname.log.0.20060420235128
   ```

4. Dieser Ausgabe entnimmt der Sicherheitsadministrator, dass sich die benötigten Dateien in nur einer Datei befinden sollten: db2audit.dbname.log.20060419234937. Die Zeitmarke zeigt, dass diese Datei am Ende des Tages an dem Tag, den die Prüfer sehen möchten, archiviert wurde.

   Der Sicherheitsadministrator verwendet diesen Dateinamen als Eingabe für die gespeicherte Prozedur SYSPROC.AUDIT_DELIM_EXTRACT, um die Prüfdaten in Dateien mit begrenzter Satzlänge zu extrahieren. Die Prüfdaten in diesen Dateien können in DB2-Datenbanktabellen geladen werden, wo sie analysiert werden können, um die bestimmte Anweisung zu ermitteln, für die sich die

Prüfer interessieren. Obwohl die Prüfer nur an einer einzigen SQL-Anweisung interessiert sind, müssen möglicherweise mehrere Anweisungen aus dieser UOW untersucht werden, falls sie eine Auswirkung auf die fragliche Anweisung haben.

5. Zur Wiederholung der Anweisung muss der Sicherheitsadministrator die folgenden Aktionen ausführen:
   - Ermitteln der genauen Anweisung, die aus diesem Prüfsatz ausgeführt werden muss
   - Ermitteln des Benutzers, der die Anweisung aus diesem Prüfsatz ausgeführt hat
   - Erneutes Erstellen der genauen Berechtigungen (einschließlich LBAC-Schutz) des Benutzers zu dem Zeitpunkt, als er die Anweisung ausführte
   - Reproduzieren der Kompilierungsumgebung unter Verwendung der Spalte mit der Beschreibung der Kompilierungsumgebung im Prüfsatz in Kombination mit der Anweisung SET COMPILATION ENVIRONMENT
   - Wiederherstellen der Datenbank in exakt dem Status, den sie zum Zeitpunkt der Ausführung der Anweisung hatte

   Zur Vermeidung von Störungen des Produktionssystem sollten alle Wiederherstellungen der Datenbank und Wiederholungen von Anweisungen auf einem zweiten Datenbanksystem durchgeführt werden. Der Sicherheitsadministrator kann unter der Benutzer-ID des Benutzers, der die Anweisung ausgeführt hat, die Anweisung, wie sie im Anweisungstext vorliegt, mit den in den Anweisungsdatenelementen bereitgestellten Eingabevariablen erneut ausführen.

Verwalten der Prüffunktion

Funktionsweise der Prüffunktion

Dieser Abschnitt enthält Hintergrundinformationen, die Ihnen beim Verständnis helfen sollen, wie der zeitliche Ablauf beim Schreiben von Prüfsätzen in das Protokoll die Datenbankleistung beeinflussen kann, wie Fehler behandelt werden können, die in der Prüffunktion auftreten, und wie Prüfsätze in den verschiedenen Situationen generiert werden.

Steuern des Zeitpunkts für das Schreiben von Prüfsätzen in das aktive Protokoll

Das Schreiben der Prüfsätze in das aktive Protokoll kann synchron oder asynchron zum Auftreten der Ereignisse erfolgen, die die Generierung dieser Prüfsätze auslösen. Der Wert des Konfigurationsparameters *audit_buf_sz* des Datenbankmanagers legt fest, wann das Schreiben der Prüfsätze stattfindet.

Wenn der Parameter *audit_buf_sz* den Wert null (0) hat, erfolgt das Schreiben synchron. Das den Prüfsatz auslösende Ereignis bleibt im Wartestatus bis der Prüfsatz auf die Platte geschrieben ist. Die Wartezeit beim Schreiben der einzelnen Prüfsätze wirkt sich nachteilig auf die Verarbeitungsleistung der DB2-Datenbank aus.

Wenn der Wert des Parameters *audit_buf_sz* größer als null ist, wird der Schreibvorgang asynchron ausgeführt. Wenn der Parameterwert für *audit_buf_sz* größer als null ist, entspricht er der Anzahl an 4-KB-Seiten, die zum Erstellen eines internen Puffers verwendet werden. In dem internen Puffer werden Prüfsätze zwischengespeichert, bis sie gruppenweise auf die Platte geschrieben werden. Die Anweisung, die den aus einem Prüfereignis resultierenden Prüfsatz generiert, wartet nicht, bis der Prüfsatz auf die Platte geschrieben ist, sondern kann die Verarbeitung ohne Verzögerung fortsetzen.

Beim asynchronen Schreiben verbleiben die Prüfsätze gegebenenfalls für einige Zeit in einem nur teilweise gefüllten Puffer. Damit dies nicht über zu lange Zeiträume der Fall ist, erzwingt der Datenbankmanager das regelmäßige Schreiben der Prüfsätze auf die Platte. Ein berechtigter Benutzer der Prüffunktion kann den Prüfpuffer auch mit einer expliziten Anforderung (FLUSH) leeren. Die Puffer werden zudem bei einer Archivierungsoperation automatisch geleert.

Je nachdem, ob der Schreibvorgang synchron oder asynchron erfolgt, ergeben sich Unterschiede beim Auftreten von Fehlern. Im asynchronen Modus gehen möglicherweise einige Datensätze verloren, da die Prüfsätze gepuffert werden, bevor sie auf die Platte geschrieben werden. Im synchronen Modus geht, wenn überhaupt, ein Datensatz verloren, da durch den Fehler höchstens ein Prüfsatz nicht geschrieben werden kann.

Verwalten von Fehlern der Prüffunktion

Die Einstellung des Prüffunktionsparameters ERRORTYPE legt fest, wie auftretende Fehler zwischen dem DB2-Datenbanksystem und der Prüffunktion verwaltet werden. Wenn die Prüffunktion aktiv und der Prüffunktionsparameter ERRORTYPE auf `AUDIT` gesetzt ist, wird die Prüffunktion genau so behandelt wie jede andere Komponente der DB2-Datenbank. Ein Prüfsatz muss geschrieben werden (im synchronen Modus auf die Platte, im asynchronen Modus in den Prüfpuffer), damit ein Prüfereignis, das sich auf eine Anweisung bezieht, als erfolgreich eingestuft wird. Bei jedem in diesem Modus festgestellten Fehler wird für die Anweisung, die einen Prüfsatz generiert hat, ein negativer SQLCODE-Wert an die Anwendung zurückgegeben.

Ist der Parameter ERRORTYPE auf `NORMAL` gesetzt, werden alle durch db2audit verursachten Fehler ignoriert und der SQLCODE-Wert der Operation wird zurückgegeben.

In verschiedenen Situationen generierte Prüfsätze

In Abhängigkeit von der API bzw. Abfrageanweisung und den Prüfeinstellungen können für ein bestimmtes Ereignis kein Prüfsatz, ein Prüfsatz oder mehrere Prüfsätze generiert werden. Beispiel: Eine SQL-Anweisung UPDATE mit einer Unterabfrage SELECT kann zur Generierung eines Prüfsatzes mit den Ergebnissen der Berechtigungsprüfung für das Zugriffsrecht UPDATE einer Tabelle und zur Generierung eines weiteren Prüfsatzes mit den Ergebnissen der Berechtigungsprüfung für das Zugriffsrecht SELECT einer Tabelle führen.

Für dynamische DML-Anweisungen (DML = Data Manipulation Language) werden Prüfsätze für alle Berechtigungsprüfungen zum Zeitpunkt der Anweisungsvorbereitung (Prepare) generiert. Die erneute Verwendung dieser Anweisungen durch denselben Benutzer wird nicht erneut überprüft, weil zu diesem Zeitpunkt keine Berechtigungsprüfung stattfindet. Wenn jedoch eine der Katalogtabellen mit Berechtigungsinformationen geändert wurde, werden in der nächsten UOW die Berechtigungen für die im Cache gespeicherten dynamischen SQL- oder XQuery-Anweisungen erneut geprüft und mindestens ein neuer Prüfsatz erstellt.

Für ein Paket, das ausschließlich statische DML-Anweisungen enthält, ist die Berechtigungsprüfung, mit der geprüft wird, ob ein Benutzer die erforderliche Berechtigung zum Ausführen des Pakets hat, das einzige prüfbare Ereignis, das einen Prüfsatz generieren könnte. Die Berechtigungsprüfung und die gegebenenfalls erforderliche Prüfsatzerstellung für die statischen SQL- oder XQuery-Anweisungen des Pakets werden beim Vorkompilieren oder Binden des Pakets durchge-

führt. Die Ausführung der statischen SQL- bzw. XQuery-Anweisungen in dem Paket kann mit der Kategorie EXECUTE (Ausführen) geprüft werden. Wird für ein Paket entweder explizit vom Benutzer oder implizit vom System ein Rebind durchgeführt, werden Prüfsätze für die von den statischen SQL- oder XQuery-Anweisungen benötigten Berechtigungsprüfungen generiert.

Für Anweisungen, bei denen die Berechtigungsprüfung zum Zeitpunkt der Ausführung erfolgt (z. B. Data Definition Language (DDL) oder GRANT- und REVOKE-Anweisungen), werden bei jeder Verwendung dieser Anweisungen Prüfsätze generiert.

Anmerkung: Bei der Ausführung von DDL ist die aufgezeichnete Abschnittsnummer für alle Ereignisse (ausgenommen Kontextereignisse) im Prüfsatz null (0), und zwar unabhängig von der tatsächlichen Abschnittsnummer der Anweisung.

Tipps und Techniken für die Prüffunktion

Archivieren des Prüfprotokolls

Sie sollten das Prüfprotokoll regelmäßig archivieren. Die Archivierung des Prüfprotokolls versetzt das aktuelle Prüfprotokoll in ein Archivverzeichnis, während der Server gleichzeitig mit dem Schreiben in ein neues, aktives Prüfprotokoll beginnt. Der Name jeder archivierten Protokolldatei enthält eine Zeitmarke, an der Sie interessante Protokolldateien bei einer späteren Analyse identifizieren können.

Für die langfristige Speicherung ist möglicherweise eine Komprimierung von Gruppen archivierter Dateien in ZIP-Dateien sinnvoll.

Für archivierte Prüfprotokolle, die für Sie von keinem Interesse mehr sind, kann der Instanzeigner einfach die Dateien aus dem Betriebssystem löschen.

Fehlerbehandlung

Wenn Sie eine Prüfrichtlinie erstellen, sollten Sie den Fehlertyp AUDIT verwenden, sofern Sie nicht nur eine Testprüfrichtlinie erstellen. Wenn zum Beispiel der Fehlertyp auf AUDIT gesetzt ist und ein Fehler auftritt, zum Beispiel durch nicht ausreichenden Plattenspeicherplatz, wird eine Fehlernachricht zurückgegeben. Die Fehlerbedingung muss behoben werden, bevor alle weiteren, von der Prüffunktion überwachten Aktionen fortgesetzt werden können. Wenn der Fehlertyp jedoch auf NORMAL gesetzt war, würde die Protokollierung einfach fehlschlagen, ohne einen Fehler an den Benutzer zurückzumelden. Der Betrieb wird so fortgesetzt, als wäre der Fehler nicht aufgetreten.

Falls während der Archivierung ein Problem auftritt, zum Beispiel, wenn der Plattenspeicherplatz im Archivpfad nicht ausreicht oder der Archivpfad nicht vorhanden ist, schlägt der Archivierungsprozess fehl. In diesem Fall wird eine vorläufige Protokolldatei mit der Dateierweiterung .bk im Datenpfad für das Prüfprotokoll generiert. Beispiel: db2audit.instance.log.0.20070508172043640941.bk. Wenn das Problem beseitigt ist (durch Zuordnen eines ausreichenden Plattenspeicherbereichs im Archivpfad bzw. durch Erstellen des Archivpfads) müssen Sie diese vorläufige Protokolldatei in den Archivpfad versetzen. Anschließend können Sie diese Datei genauso wie ein erfolgreich archiviertes Protokoll behandeln.

Einschränkungen für DDL-Anweisungen

Einige Anweisungen der Datendefinitionssprache (DDL-Anweisungen), die als exklusive SQL-Anweisungen der Prüffunktion (exklusive AUDIT-Anweisungen) bezeichnet werden, werden erst bei der nächsten UOW (Unit of Work, Arbeitseinheit) wirksam. Daher gilt die Empfehlung, die Anweisung COMMIT unmittelbar nach jeder dieser Anweisungen zu verwenden.

Bei den exklusiven SQL-Anweisungen der Prüffunktion handelt es sich um folgende Anweisungen:

- AUDIT
- CREATE AUDIT POLICY, ALTER AUDIT POLICY und DROP AUDIT POLICY
- DROP ROLE und DROP TRUSTED CONTEXT, wenn die Rolle bzw. der gesicherte Kontext, die bzw. der gelöscht wird, einer Prüfrichtlinie zugeordnet ist

Mögliche Änderung des Tabellenformats zum Speichern archivierter Daten

Mit der gespeicherten Prozedur SYSPROC.AUDIT_DEL_EXTRACT kann der Sicherheitsadministrator, bzw. der Systemadministrator mit dem Befehl db2audit extract, Prüfsätze aus den archivierten Prüfprotokolldateien in Dateien mit begrenzter Satzlänge extrahieren. Sie können die Prüfdaten aus den Dateien mit begrenzter Satzlänge zur Analyse in DB2-Datenbanktabellen laden. Das Format der Tabellen, die Sie zur Aufnahme der Prüfdaten erstellen müssen, kann sich von Release zu Release ändern.

Wichtig: Das Script db2audit.ddl erstellt Tabellen im korrekten Format, um die Prüfsätze aufzunehmen. Sie sollten die Ausführung des Scripts db2audit.ddl für jedes Release vorsehen, da möglicherweise neue Spalten hinzugefügt oder die Größe einer vorhandenen Spalte geändert werden.

Verwenden von CHECKING-Ereignissen

Beim Arbeiten mit CHECKING-Ereignissen wird meist das Feld für den Objekttyp im Prüfsatz überprüft, um festzustellen, ob die Benutzer-ID, die auf das Objekt zuzugreifen versucht, über das erforderliche Zugriffsrecht bzw. die erforderliche Berechtigung verfügt. Beispiel: Wenn ein Benutzer versucht, mit ALTER eine Tabellenspalte hinzuzufügen, weist der Prüfsatz des CHECKING-Ereignisses darauf hin, dass ein Zugriff mit „ALTER" auf ein Objekt des Typs „TABLE" (Tabelle) versucht wurde (nicht Spalte, weil in diesem Fall die Tabellenzugriffsrechte geprüft werden).

Muss beim Überprüfen jedoch festgestellt werden, ob die erforderliche Datenbankberechtigung vorhanden ist, mit der eine Benutzer-ID eine CREATE- oder BIND-Operation an einem Objekt ausführen oder ein Objekt löschen kann, dann erfolgt zwar eine Prüfung in der Datenbank, aber im Feld für den Objekttyp wird das zu erstellende, zu bindende oder zu löschende Objekt angegeben (und nicht die Datenbank).

Bei der Indexerstellung für eine Tabelle ist das Zugriffsrecht zum Erstellen eines Index erforderlich, deshalb wird im Prüfsatz des CHECKING-Ereignisses der Zugriffstyp „index" und nicht „create" angegeben.

Für das Binden eines Pakets erstellte Prüfsätze

Beim Binden eines bereits vorhandenen Pakets wird ein OBJMAINT-Prüfsatz für die DROP-Operation des Pakets erstellt und danach ein weiterer OBJMAINT-Prüfsatz für die CREATE-Operation der neuen Paketkopie.

Verwenden von CONTEXT-Ereignisdaten nach ROLLBACK

Die Datendefinitionssprache (Data Definition Language, DDL) kann OBJMAINT- oder SECMAINT-Ereignisse generieren, die als erfolgreich protokolliert werden. Es kann jedoch vorkommen, dass nach der Protokollierung des Ereignisses durch einen nachfolgenden Fehler ein Rollback ausgelöst wird. Dadurch würde das Objekt als nicht erstellt bzw. die GRANT- oder REVOKE-Operation als unvollständig ausgewiesen. In diesem Fall erweist sich die Verwendung von CONTEXT-Ereignissen als hilfreich. Die Prüfsätze von CONTEXT-Ereignissen, insbesondere die abschließende Anweisung des Ereignisses, geben an, wie die versuchte Operation beendet wurde.

Ladebegrenzer

Beim Extrahieren von Prüfsätzen in begrenztem Format, das sich zum Laden in eine DB2-Datenbanktabelle eignet, ist darauf zu achten, dass im Textfeld der Anweisung der richtige Begrenzer verwendet wird. Dies kann beim Extrahieren der Datei mit begrenzter Satzlänge zum Beispiel mit folgendem Befehl erreicht werden:

```
db2audit extract delasc delimiter <ladebegrenzer>
```

Dabei kann *ladebegrenzer* ein Einzelzeichen (z. B. ″) oder ein Vierbytezeichen sein, das eine Hexadezimalzahl darstellt (z. B. „0xff”). Folgende Beispiele zeigen gültige Befehle:

```
db2audit extract delasc
db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

Wenn beim Extrahieren nicht der Standardladebegrenzer (″) verwendet wurde, sollte die Option MODIFIED BY im Befehl LOAD verwendet werden. Das folgende Beispiel zeigt einen abgekürzten Befehl LOAD mit dem Begrenzer „0xff”:

```
db2 load from context.del of del modified by chardel0xff replace into ...
```

Dadurch wird der Standardzeichenfolgebegrenzer ″ (Anführungszeichen) für LOAD außer Kraft gesetzt.

Kapitel 2. Rollen

Rollen vereinfachen die Verwaltung und das Management von Zugriffsrechten, indem sie eine äquivalente Funktionalität zu Gruppen bieten, jedoch ohne deren Einschränkungen. Eine Rolle ist ein Datenbankobjekt, das ein oder mehrere Zugriffsrechte zusammenfasst und Benutzern, Gruppen, PUBLIC oder anderen Rollen durch eine Anweisung GRANT oder einem gesicherten Kontext durch eine Anweisung CREATE TRUSTED CONTEXT oder ALTER TRUSTED CONTEXT erteilt werden kann. Eine Rolle kann für das Verbindungsattribut SESSION_USER ROLE in einer Auslastungsdefinition (Workloaddefinition) angegeben werden.

Rollen bieten verschiedene Vorteile, die das Management von Zugriffsrechten in einem Datenbanksystem vereinfachen:

- Sicherheitsadministratoren können den Zugriff auf ihre Datenbanken auf eine Weise steuern, die die Struktur ihrer Unternehmen widerspiegelt. (Sie können Rollen in der Datenbank erstellen, die die Aufgabenbereiche in ihren Unternehmen direkt abbilden.)
- Benutzern wird die Zugehörigkeit zu den Rollen erteilt, die ihren Zuständigkeiten entsprechen. Wenn sich Zuständigkeiten ändern, lässt sich die Rollenzugehörigkeit problemlos neu erteilen und entziehen.
- Die Zuordnung von Zugriffsrechten wird vereinfacht. Anstatt die gleiche Gruppe von Zugriffsrechten jedem einzelnen Benutzer mit einem bestimmten Aufgabenbereich zu erteilen, kann der Administrator diese Gruppe von Zugriffsrechten einer Rolle erteilen, die diesen Aufgabenbereich darstellt, und anschließend diese Rolle jedem Benutzer mit diesem Aufgabenbereich erteilen.
- Die Zugriffsrechte einer Rolle können aktualisiert werden, und alle Benutzer, denen diese Rolle erteilt wurde, empfangen die Aktualisierung. Der Administrator braucht die Zugriffsrechte nicht für jeden Benutzer einzeln zu aktualisieren.
- Die Zugriffsrechte und Berechtigungen, die Rollen erteilt sind, werden bei jeder Erstellung von Sichten, Triggern, MQTs (Materialized Query Tables), statischem SQL und SQL-Routinen verwendet, während die Zugriffsrechte und Berechtigungen, die Gruppen (direkt oder indirekt) erteilt sind, nicht verwendet werden.

  Dies liegt daran, dass das DB2-Datenbanksystem nicht feststellen kann, wenn sich die Zugehörigkeit zu einer Gruppe ändert, da die Gruppe von der Software eines anderen Anbieters (z. B. dem Betriebssystem oder einem LDAP-Verzeichnis) verwaltet wird. Da Rollen innerhalb der Datenbank verwaltet werden, kann das DB2-Datenbanksystem feststellen, wenn sich eine Berechtigung ändert, und sich entsprechend verhalten. Rollen, die Gruppen erteilt sind, werden aus demselben Grund nicht berücksichtigt, aus dem Gruppen nicht berücksichtigt werden.
- Alle Rollen, die einem Benutzer zugeordnet sind, werden aktiviert, wenn dieser Benutzer eine Verbindung herstellt, sodass alle Zugriffsrechte und Berechtigungen, die Rollen erteilt sind, bei der Herstellung einer Verbindung durch einen Benutzer ausgewertet werden. Rollen können nicht explizit aktiviert oder inaktiviert werden.
- Der Sicherheitsadministrator kann das Management einer Rolle an andere Personen delegieren.

Alle DB2-Zugriffsrechte und -Berechtigungen, die innerhalb einer Datenbank erteilt werden können, können einer Rolle erteilt werden. Ausgenommen von dieser

Regel ist die Berechtigung SECADM (Sicherheitsadministrator). Zum Beispiel können einer Rolle beliebige der folgenden Berechtigungen und Zugriffsrechte erteilt werden:

- Die Datenbankberechtigungen DBADM, LOAD und IMPLICIT_SCHEMA
- Die Datenbankberechtigungen CONNECT, CREATETAB, CREATE_NOT-_FENCED, BINDADD CREATE_EXTERNAL_ROUTINE oder QUIESCE_CONNECT
- Alle Zugriffsrechte für Datenbankobjekte (einschließlich CONTROL)

Die Rollen eines Benutzers werden automatisch aktiviert und bei der Berechtigungsverarbeitung berücksichtigt, wenn der Benutzer eine Verbindung zur Datenbank herstellt. Sie brauchen eine Rolle nicht durch die Anweisung SET ROLE zu aktivieren. Wenn Sie zum Beispiel eine Sicht, eine MQT, einen Trigger, ein Paket oder eine SQL-Routine erstellen, werden die Zugriffsrechte, über die Sie durch Rollen verfügen, angewendet. Die Zugriffsrechte, über die Sie durch Rollen verfügen, die Gruppen erteilt sind, deren Mitglied Sie sind, werden hingegen nicht angewendet.

Eine Rolle hat keinen Eigner. Der Sicherheitsadministrator kann mithilfe der Klausel WITH ADMIN OPTION der Anweisung GRANT das Management der Rolle an einen anderen Benutzer delegieren, sodass der andere Benutzer die Rollenzugehörigkeit steuern kann.

Einschränkungen

Bei der Verwendung von Rollen sind einige Einschränkungen zu beachten:

- Eine Rolle kann nicht Eigner von Datenbankobjekten sein.
- Einer Rolle kann nicht die Berechtigung SECADM (Sicherheitsadministrator) erteilt werden.
- Berechtigungen und Rollen, die Gruppen erteilt sind, werden beim Erstellen der folgenden Datenbankobjekte nicht geprüft:
  - Pakete mit statischem SQL
  - Sichten
  - MQTs (Materialized Query Tables)
  - Trigger
  - SQL-Routinen

  Beim Erstellen dieser Objekte werden nur die Rollen geprüft, die dem Benutzer, der das Objekt erstellt, oder PUBLIC direkt oder indirekt (z. B. durch eine Rollenhierarchie) erteilt sind.

Erstellen von Rollen und Erteilen der Zugehörigkeit

Der Sicherheitsadministrator hat die Berechtigung, eine Rolle zu erstellen, zu löschen, zu erteilen und zu entziehen sowie einer Rolle einen Kommentar hinzuzufügen. Der Sicherheitsadministrator verwendet die Anweisung GRANT (Rolle), um einer Berechtigungs-ID die Zugehörigkeit zu einer Rolle zu erteilen, und die Anweisung REVOKE (Rolle), um einer Berechtigungs-ID die Zugehörigkeit zu einer Rolle zu entziehen.

Der Sicherheitsadministrator kann das Management der Zugehörigkeit zu einer Rolle an eine Berechtigungs-ID delegieren, indem er der Berechtigungs-ID die Zugehörigkeit zu dieser Rolle mit der Klausel WITH ADMIN OPTION in der

Anweisung GRANT erteilt. Die Klausel WITH ADMIN OPTION der Anweisung GRANT (Rolle) gibt einem anderen Benutzer folgende Möglichkeiten:

- Er kann Rollen anderen Benutzern erteilen.
- Er kann Rollen anderen Benutzern entziehen.
- Er kann eine Rolle mit Kommentaren versehen.

Die Klausel WITH ADMIN OPTION schränkt den berechtigten Benutzer jedoch wie folgt ein:

- Er kann die Rolle nicht löschen.
- Er kann einer Berechtigungs-ID die Berechtigung mit der Klausel WITH ADMIN OPTION für die Rolle nicht entziehen.
- Er kann einem anderen Benutzer die Rolle nicht mit der Klausel WITH ADMIN OPTION erteilen (sofern er nicht die Berechtigung SECADM besitzt).

Wenn der Sicherheitsadministrator eine Rolle erstellt hat, kann der Datenbankadministrator der Rolle mithilfe der Anweisung GRANT Berechtigungen und Zugriffsrechte zuordnen. Einer Rolle können alle DB2-Zugriffsrechte und -Berechtigungen, die innerhalb einer Datenbank erteilt werden können, mit Ausnahme der Berechtigung SECADM, erteilt werden. Berechtigungen auf der Instanzebene, wie zum Beispiel die Berechtigung SYSADM (Systemadministrator), können einer Rolle nicht zugeordnet werden.

Der Sicherheitsadministrator bzw. jeder Benutzer, dem der Sicherheitsadministrator die Zugehörigkeit zu einer Rolle mit der Klausel WITH ADMIN OPTION erteilt hat, kann die Anweisung GRANT (Rolle) verwenden, um die Zugehörigkeit zu dieser Rolle anderen Benutzern, Gruppen, der speziellen Gruppe PUBLIC oder Rollen zu erteilen. Einem Benutzer kann die Zugehörigkeit zu einer Rolle mit der Klausel WITH ADMIN OPTION entweder direkt oder indirekt über PUBLIC, eine Gruppe oder eine Rolle erteilt werden.

Alle Rollen, die einem Benutzer zugeordnet sind, werden aktiviert, wenn dieser Benutzer eine Sitzung einrichtet. Alle Zugriffsrechte und Berechtigungen, die den Rollen eines Benutzers zugeordnet sind, werden bei der Berechtigungsprüfung durch das DB2-Datenbanksystem berücksichtigt. Einige Datenbanksysteme verwenden die Anweisung SET ROLE, um eine bestimmte Rolle zu aktivieren. Das DB2-Datenbanksystem unterstützt die Anweisung SET ROLE aus Gründen der Kompatibilität mit anderen Produkten, die mit der Anweisung SET ROLE arbeiten. In einem DB2-Datenbanksystem überprüft die Anweisung SET ROLE, ob dem Sitzungsbenutzer die Rolle erteilt ist, und gibt einen Fehler zurück, wenn dies nicht der Fall ist.

Zum Entziehen der Rollenzugehörigkeit eines Benutzers verwendet der Sicherheitsadministrator bzw. ein Benutzer, der das Zugriffsrecht WITH ADMIN OPTION für die Rolle besitzt, die Anweisung REVOKE (Rolle).

Beispiel

Eine Rolle besitzt eine Reihe von Zugriffsrechten, und ein Benutzer, dem diese Rolle erteilt wird, übernimmt diese Zugriffsrechte. Diese Übernahme von Zugriffsrechten vermeidet die Verwaltung einzelner Zugriffsrechte, wenn Zugriffsrechte eines Benutzers auf einen anderen Benutzer übertragen werden. Die einzigen Operationen, die bei Verwendung von Rollen dazu erforderlich sind, bestehen darin, die Zugehörigkeit zu der Rolle dem einen Benutzer zu entziehen und sie dem anderen Benutzer zu erteilen.

Betrachten Sie ein Beispiel: Die Mitarbeiter BOB und ALICE, die in der Abteilung DEV ('Entwicklung') arbeiten, haben das Zugriffsrecht SELECT für die Tabellen SERVER, CLIENT und TOOLS. Eines Tages entscheidet das Management, sie in eine andere Abteilung, zum Beispiel QA, zu versetzen. Der Datenbankmanager muss ihnen das Zugriffsrecht SELECT für die Tabellen SERVER, CLIENT und TOOLS entziehen. Die Abteilung DEV stellt nachfolgend einen neuen Mitarbeiter TOM ein, sodass der Datenbankadministrator dem Benutzer TOM das Zugriffsrecht SELECT für die Tabellen SERVER, CLIENT und TOOLS erteilen muss.

Bei Verwendung von Rollen sind in diesem Fall die folgenden Schritte erforderlich:

1. Der Sicherheitsadministrator erstellt die Rolle DEVELOPER ('Entwickler'):
   ```
   CREATE ROLE DEVELOPER
   ```
2. Der Datenbankadministrator (mit der Berechtigung DBADM) erteilt der Rolle DEVELOPER das Zugriffsrecht SELECT für die Tabellen SERVER, CLIENT und TOOLS:
   ```
   GRANT  SELECT ON TABLE SERVER TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE CLIENT TO ROLE DEVELOPER
   GRANT  SELECT ON TABLE TOOLS TO ROLE DEVELOPER
   ```
3. Der Sicherheitsadministrator erteilt den Benutzern BOB und ALICE in der Abteilung DEV die Rolle DEVELOPER:
   ```
   GRANT ROLE DEVELOPER TO USER BOB, USER ALICE
   ```
4. Wenn BOB und ALICE die Abteilung DEV verlassen, entzieht der Sicherheitsadministrator den Benutzern BOB und ALICE die Rolle DEVELOPER:
   ```
   REVOKE ROLE DEVELOPER FROM USER BOB, USER ALICE
   ```
5. Wenn TOM in der Abteilung DEV angestellt wird, erteilt der Sicherheitsadministrator dem Benutzer TOM die Rolle DEVELOPER:
   ```
   GRANT ROLE DEVELOPER TO USER TOM
   ```

Rollenhierarchien

Eine Rollenhierarchie entsteht, wenn einer Rolle die Zugehörigkeit zu einer anderen Rolle erteilt wird.

Eine Rolle *enthält* eine andere Rolle, wenn die andere Rolle der ersten Rolle erteilt wird. Die andere Rolle übernimmt alle Zugriffsrechte der ersten Rolle. Wenn zum Beispiel die Rolle DOCTOR der Rolle SURGEON (Chirurg) erteilt wird, dann lässt sich sagen, dass die Rolle SURGEON die Rolle DOCTOR enthält. Die Rolle SURGEON übernimmt alle Zugriffsrechte der Rolle DOCTOR.

Zyklen in Rollenhierarchien sind nicht zulässig. Ein *Zyklus* entsteht, wenn erteilte Rollen einen abgeschlossenen Kreis bilden, indem zum Beispiel eine Rolle einer anderen Rolle erteilt wird und diese andere Rolle wiederum der ersten Rolle erteilt wird. Ein Beispiel für einen Zyklus wäre, wenn die Rolle DOCTOR der Rolle SURGEON erteilt würde und anschließend die Rolle SURGEON wieder der Rolle DOCTOR erteilt würde. Wenn Sie einen Zyklus in einer Rollenhierarchie erstellen, wird ein Fehler zurückgegeben (SQLSTATE-Wert 428GF).

Beispiel für den Aufbau einer Rollenhierarchie

Das folgende Beispiel zeigt, wie eine Rollenhierarchie aufgebaut wird, um die medizinischen Ebenen in einem Krankenhaus darzustellen.

Betrachten Sie die folgenden Rollen: DOCTOR (Arzt), SPECIALIST (Facharzt) und SURGEON (Chirurg). Eine Rollenhierarchie wird aufgebaut, indem eine Rolle einer

anderen Rolle erteilt wird, jedoch ohne Zyklen zu bilden. Die Rolle DOCTOR wird der Rolle SPECIALIST, die Rolle SPECIALIST der Rolle SURGEON erteilt.

Wenn die Rolle SURGEON der Rolle DOCTOR erteilt würde, entstünde ein Zyklus, der nicht zulässig wäre.

Der Sicherheitsadministrator führt die folgenden SQL-Anweisungen aus, um die Rollenhierarchie aufzubauen:

```
CREATE ROLE DOCTOR
CREATE ROLE SPECIALIST
CREATE ROLE SURGEON

GRANT ROLE DOCTOR TO ROLE SPECIALIST

GRANT ROLE SPECIALIST TO ROLE SURGEON
```

Wirkung des Widerrufens von Zugriffsrechten für Rollen

Wenn Zugriffsrechte widerrufen werden, kann dies in manchen Fällen dazu führen, dass abhängige Datenbankobjekte, wie zum Beispiel Sichten, Pakete oder Trigger, ungültig oder unbrauchbar werden.

Die folgenden Beispiele zeigen, was mit einem Datenbankobjekt geschieht, wenn einige Zugriffsrechte einer Berechtigungs-ID widerrufen werden und die Zugriffsrechte durch eine Rolle oder andere Methoden erteilt wurden.

Beispiel für das Widerrufen von Zugriffsrechten von Rollen

1. Der Sicherheitsadministrator erstellt die Rolle DEVELOPER und erteilt dem Benutzer BOB die Zugehörigkeit zu dieser Rolle:

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB
   ```

2. Die Benutzerin ALICE erstellt die Tabelle WORKITEM:

   ```
   CREATE TABLE WORKITEM (x int)
   ```

3. Der Datenbankadministrator erteilt PUBLIC und der Rolle DEVELOPER die Zugriffsrechte SELECT und INSERT für die Tabelle WORKITEM:

   ```
   GRANT SELECT ON TABLE ALICE.WORKITEM TO PUBLIC
   GRANT INSERT ON TABLE ALICE.WORKITEM TO PUBLIC
   GRANT SELECT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
   GRANT INSERT ON TABLE ALICE.WORKITEM TO ROLE DEVELOPER
   ```

4. Der Benutzer BOB erstellt die Sicht PROJECT, die auf der Tabelle WORKITEM basiert, und das Paket PKG1, das von der Tabelle WORKITEM abhängig ist:

   ```
   CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
   PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
   ```

5. Wenn der Datenbankadministrator der Gruppe PUBLIC das Zugriffsrecht SELECT für die Tabelle ALICE.WORKITEM entzieht, bleiben die Sicht BOB-.PROJECT brauchbar und das Paket PKG1 gültig, weil der Benutzer BOB, der die Sicht und das Paket definiert hat, weiterhin die erforderlichen Zugriffsrechte durch seine Zugehörigkeit zur Rolle DEVELOPER besitzt:

   ```
   REVOKE SELECT ON TABLE ALICE.WORKITEM FROM PUBLIC
   ```

6. Wenn der Datenbankadministrator der Rolle DEVELOPER das Zugriffsrecht SELECT für die Tabelle ALICE.WORKITEM entzieht, werden die Sicht BOB-.PROJECT unbrauchbar und das Paket PKG1 ungültig, weil der Benutzer BOB, der die Sicht und das Paket definiert hat, die erforderlichen Zugriffsrechte jetzt nicht mehr durch andere Methoden besitzt:

   ```
   REVOKE SELECT ON TABLE ALICE.WORKITEM FROM ROLE DEVELOPER
   ```

Beispiel für das Widerrufen der Berechtigung DBADM

In diesem Beispiel erhält die Rolle DEVELOPER die Berechtigung DBADM und wird dem Benutzer BOB erteilt.

1. Der Sicherheitsadministrator erstellt die Rolle DEVELOPER:
   ```
   CREATE ROLE DEVELOPER
   ```
2. Der Systemadministrator erteilt der Rolle DEVELOPER die Berechtigung DBADM:
   ```
   GRANT DBADM ON DATABASE TO ROLE DEVELOPER
   ```
3. Der Sicherheitsadministrator erteilt dem Benutzer BOB die Zugehörigkeit zu dieser Rolle:
   ```
   GRANT ROLE DEVELOPER TO USER BOB
   ```
4. Die Benutzerin ALICE erstellt die Tabelle WORKITEM:
   ```
   CREATE TABLE WORKITEM (x int)
   ```
5. Der Benutzer BOB erstellt die Sicht PROJECT, die auf der Tabelle WORKITEM basiert, das Paket PKG1, das von der Tabelle WORKITEM abhängig ist, und den Trigger TRG1, der ebenfalls von der Tabelle WORKITEM abhängig ist:
   ```
   CREATE VIEW PROJECT AS SELECT * FROM ALICE.WORKITEM
   PREP emb001.sqc BINDFILE PACKAGE USING PKG1 VERSION 1
   CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
                FOR EACH STATEMENT MODE DB2SQL
         INSERT INTO ALICE.WORKITEM VALUES (1)
   ```
6. Der Sicherheitsadministrator entzieht dem Benutzer BOB die Rolle DEVELOPER:
   ```
   REVOKE ROLE DEVELOPER FROM USER BOB
   ```
   Durch das Entziehen der Rolle DEVELOPER verliert der Benutzer BOB die Berechtigung DBADM, weil ihm die Rolle, die diese Berechtigung besitzt, entzogen wird. Die Sicht, das Paket und der Trigger werden davon wie folgt betroffen:
   - Die Sicht BOB.PROJECT ist weiterhin gültig.
   - Das Paket PKG1 wird ungültig.
   - Der Trigger BOB.TRG1 ist weiterhin gültig.

   Die Sicht BOB.PROJECT und der Trigger BOB.TRG1 bleiben verwendbar, während das Paket PKG1 nicht mehr verwendet werden kann. Sicht- und Triggerobjekte, die durch eine Berechtigungs-ID mit der Berechtigung DBADM erstellt werden, werden nicht betroffen, wenn die Berechtigung DBADM widerrufen wird.

Delegieren der Rollenverwaltung mit der Klausel WITH ADMIN OPTION

Mithilfe der Klausel WITH ADMIN OPTION der SQL-Anweisung GRANT (Rolle) kann ein Sicherheitsadministrator das Management und die Steuerung der Zugehörigkeit zu einer Rolle an eine andere Person delegieren. Die Klausel WITH ADMIN OPTION gibt einem anderen Benutzer die Berechtigung, die Zugehörigkeit zur jeweiligen Rolle anderen Benutzern zu erteilen, anderen Benutzern, die diese Rolle haben, diese Rolle zu entziehen und diese Rolle mit Kommentaren zu versehen. Jedoch kann er die Rolle nicht löschen.

Die Klausel WITH ADMIN OPTION gibt einem anderen Benutzer nicht die Berechtigung, die Rolle wiederum einem anderen Benutzer ebenfalls mit der Klausel WITH ADMIN OPTION zu erteilen.

Darüber hinaus gibt sie nicht die Berechtigung, einer anderen Berechtigungs-ID, die diese Rolle mit dem Zugriffsrecht WITH ADMIN OPTION besitzt, das Zugriffsrecht WITH ADMIN OPTION für diese Rolle zu entziehen.

Beispiel für die Verwendung der Klausel WITH ADMIN OPTION

1. Ein Sicherheitsadministrator erstellt die Rolle DEVELOPER und erteilt die neue Rolle dem Benutzer BOB mithilfe der Anweisung GRANT mit der Klausel WITH ADMIN OPTION:

   ```
   CREATE ROLE DEVELOPER
   GRANT ROLE DEVELOPER TO USER BOB WITH ADMIN OPTION
   ```

2. Der Benutzer BOB kann jetzt die Zugehörigkeit zu dieser Rolle anderen Benutzern, zum Beispiel ALICE, erteilen und entziehen:

   ```
   GRANT ROLE DEVELOPER TO USER ALICE
   REVOKE ROLE DEVELOPER FROM USER ALICE
   ```

3. Der Benutzer BOB kann die Rolle jedoch nicht löschen oder sie einem anderen Benutzer mit der Klausel WITH ADMIN OPTION erteilen. (Diese beiden Operationen können nur vom Sicherheitsadministrator ausgeführt werden.) Die folgenden Befehle schlagen fehl, wenn sie vom Benutzer BOB ausgeführt werden:

   ```
   DROP ROLE DEVELOPER - FEHLER!
                       - Nur ein Sicherheitsadministrator darf die Rolle löschen.
   GRANT ROLE DEVELOPER TO USER ALICE WITH ADMIN OPTION - FEHLER!
                       - Nur ein Sicherheitsadministrator kann die Rolle mit der
                         Klausel WITH ADMIN OPTION erteilen.
   ```

4. Der Benutzer BOB kann die Verwaltungszugriffsrechte für die Rolle (die durch WITH ADMIN OPTION erteilt wurden) Benutzern mit der Rolle DEVELOPER nicht entziehen, weil er nicht die Berechtigung SECADM (Sicherheitsadministrator) besitzt. Wenn BOB den folgenden Befehl eingibt, schlägt dieser fehl:

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER SANJAY - FEHLER!
   ```

5. Ein Sicherheitsadministrator kann die Rollenverwaltungszugriffsrechte für die Rolle DEVELOPER (die mit der Klausel WITH ADMIN OPTION übertragen wurden) dem Benutzer BOB entziehen, sodass ihm weiterhin die Rolle DEVELOPER erteilt bleibt:

   ```
   REVOKE ADMIN OPTION FOR ROLE DEVELOPER FROM USER BOB
   ```

   Wenn ein Sicherheitsadministrator dem Benutzer BOB andererseits die Rolle DEVELOPER entzieht, verliert BOB alle Zugriffsrechte, die er durch die Zugehörigkeit zur Rolle DEVELOPER besaß, und die Berechtigung für die Rolle, die er durch die Klausel WITH ADMIN OPTION erhalten hatte:

   ```
   REVOKE ROLE DEVELOPER FROM USER BOB
   ```

Rollen im Vergleich zu Gruppen

Zugriffsrechte und Berechtigungen, die Gruppen erteilt sind, werden bei der Erstellung von Sichten, MQTs (Materialized Query Tables), SQL-Routinen, Triggern und Paketen mit statischem SQL nicht berücksichtigt. Diese Einschränkung kann durch die Verwendung von Rollen anstelle von Gruppen überwunden werden.

Durch Rollen können Benutzer Datenbankobjekte erstellen und dabei ihre durch Rollen erhaltenen Zugriffsrechte verwenden, die durch das DB2-Datenbanksystem gesteuert werden. Gruppen und Benutzer werden außerhalb des DB2-Datenbanksystems, zum Beispiel durch das Betriebssystem oder einen LDAP-Server, gesteuert.

Beispiel für das Ersetzen von Gruppen durch Rollen

Das folgende Beispiel zeigt, wie Gruppen durch die Verwendung von Rollen ersetzt werden können.

Nehmen Sie an, es seien drei Gruppen vorhanden: DEVELOPER_G, TESTER_G und SALES_G. Die Benutzer BOB, ALICE und TOM sind Mitglieder dieser Gruppen, wie in der folgenden Tabelle dargestellt:

Tabelle 7. Beispiel für Gruppen und Benutzer

Gruppe	Mitglieder der Gruppe
DEVELOPER_G	BOB
TESTER_G	ALICE, TOM
SALES_G	ALICE, BOB

1. Der Sicherheitsadministrator erstellt die Rollen DEVELOPER, TESTER und SALES, die anstelle der Gruppen verwendet werden sollen.
   ```
   CREATE ROLE DEVELOPER
   CREATE ROLE TESTER
   CREATE ROLE SALES
   ```
2. Der Sicherheitsadministrator erteilt Benutzern die Zugehörigkeit zu diesen Rollen (die Festlegung der Mitgliedschaft von Benutzern in Gruppen lag in der Zuständigkeit des Systemadministrators):
   ```
   GRANT  ROLE DEVELOPER TO USER BOB
   GRANT  ROLE TESTER TO USER ALICE, USER TOM
   GRANT  ROLE SALES TO USER BOB, USER ALICE
   ```
3. Der Datenbankadministrator kann den Rollen ähnliche Zugriffsrechte oder Berechtigungen erteilen, wie sie auch die Gruppen hatten. Beispiel:
   ```
   GRANT <zugriffsrecht> ON <objekt> TO ROLE DEVELOPER
   ```

   Der Datenbankadministrator kann anschließend den Gruppen diese Zugriffsrechte entziehen und den Systemadministrator bitten, die Gruppen aus dem System zu entfernen.

Beispiel für das Erstellen eines Triggers mit den durch eine Rolle erteilten Zugriffsrechten

Dieses Beispiel zeigt, dass der Benutzer BOB einen Trigger (TRG1) erfolgreich erstellen kann, wenn er das erforderliche Zugriffsrecht durch die Rolle DEVELOPER besitzt.

1. Zunächst erstellt die Benutzerin ALICE die Tabelle WORKITEM:
   ```
   CREATE TABLE WORKITEM (x int)
   ```
2. Anschließend wird das Zugriffsrecht ALTER zum Ändern der Tabelle der Benutzerin ALICE vom Datenbankadministrator der Rolle DEVELOPER erteilt.
   ```
   GRANT ALTER ON ALICE.WORKITEM TO ROLE DEVELOPER
   ```
3. Der Benutzer BOB kann den Trigger TRG1 erfolgreich erstellen, weil er die Zugehörigkeit zur Rolle DEVELOPER besitzt.
   ```
   CREATE TRIGGER TRG1 AFTER DELETE ON ALICE.WORKITEM
              FOR EACH STATEMENT MODE DB2SQL INSERT INTO ALICE.WORKITEM VALUES (1)
   ```

Verwenden von Rollen nach der Migration von IBM Informix Dynamic Server

Wenn Sie eine Migration von IBM Informix Dynamic Server auf das DB2-Datenbanksystem ausgeführt haben und Rollen verwenden, müssen Sie einige Gesichtspunkte beachten.

Die SQL-Anweisung GRANT ROLE von Informix Dynamic Server (IDS) ermöglicht die Verwendung der Klausel WITH GRANT OPTION. Die Anweisung GRANT ROLE des DB2-Datenbanksystems stellt die Klausel WITH ADMIN OPTION (dies entspricht dem SQL-Standard) zur Verfügung, die dieselbe Funktionalität besitzt. Während einer Migration von IDS auf das DB2-Datenbanksystem generiert das Tool dbschema Anweisungen CREATE ROLE und GRANT ROLE. Anschließend ersetzt das Tool dbschema alle Vorkommen der Klausel WITH GRANT OPTION durch WITH ADMIN OPTION.

In einem IDS-Datenbanksystem aktiviert die Anweisung SET ROLE eine bestimmte Rolle. Das DB2-Datenbanksystem unterstützt die Anweisung SET ROLE, jedoch nur, um Kompatibilität mit anderen Produkten herzustellen, die diese SQL-Anweisung verwenden. Die Anweisung SET ROLE überprüft, ob der Sitzungsbenutzer zu der Rolle gehört und gibt einen Fehler zurück, wenn er dies nicht tut.

Beispielausgabe für das Tool 'dbschema'

Nehmen Sie an, dass eine IDS-Datenbank die Rollen DEVELOPER, TESTER und SALES enthält. Den Benutzern BOB, ALICE und TOM wurden jeweils verschiedene Rollen erteilt. Die Rolle DEVELOPER wurde dem Benutzer BOB erteilt, die Rolle TESTER der Benutzerin ALICE und die Rollen TESTER und SALES dem Benutzer TOM. Zur Migration auf das DB2-Datenbanksystem verwenden Sie das Tool dbschema, um die Anweisungen CREATE ROLE und GRANT ROLE für die Datenbank zu generieren:

```
CREATE ROLE DEVELOPER
CREATE ROLE TESTER
CREATE ROLE SALES

GRANT  DEVELOPER TO BOB
GRANT  TESTER TO ALICE, TOM
GRANT  SALES TO TOM
```

Sie müssen die Datenbank im DB2-Datenbanksystem erstellen und anschließend die oben gezeigten Anweisungen in dieser Datenbank ausführen, um die Rollen und die Zuordnung der Rollen erneut zu erstellen.

Kapitel 3. Verwenden gesicherter Kontexte und gesicherter Verbindungen

Sie können eine explizite gesicherte Verbindung herstellen, indem Sie in einer Anwendung eine Anforderung absetzen, wenn eine Verbindung zu einer DB2-Datenbank hergestellt wird. Der Sicherheitsadministrator muss zuvor mithilfe der Anweisung CREATE TRUSTED CONTEXT einen gesicherten Kontext definiert haben, dessen Attribute mit denen der Verbindung übereinstimmen, die Sie herstellen (siehe Schritt 1 unten).

Die Anwendungsprogrammierschnittstelle (API), die Sie bei der Verbindungsherstellung zum Anfordern einer expliziten gesicherten Verbindung verwenden, hängt von dem Typ der Anwendung ab, den Sie verwenden (siehe Tabelle in Schritt 2).

Nachdem Sie eine explizite gesicherte Verbindung hergestellt haben, kann die Anwendung die Benutzer-ID der Verbindung mithilfe der entsprechenden API für den Typ von Anwendung (siehe Tabelle in Schritt 3) in eine andere Benutzer-ID ändern.

1. Der Sicherheitsadministrator definiert einen gesicherten Kontext auf dem Server mit der Anweisung CREATE TRUSTED CONTEXT. Beispiel:

```
CREATE TRUSTED CONTEXT MYTCX
       BASED UPON CONNECTION USING SYSTEM AUTHID NEWTON
       ATTRIBUTES (ADDRESS   '192.0.2.1')
       WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
       ENABLE
```

2. Zum Herstellen einer gesicherten Verbindung müssen Sie in Ihrer Anwendung eine der folgenden APIs verwenden:

Option	**Bezeichnung**
Anwendung	**API**
CLI/ODBC	SQLConnect, SQLSetConnectAttr
XA CLI/ODBC	Xa_open
JAVA	getDB2TrustedPooledConnection, getDB2TrustedXAConnection

3. Für den Wechsel zu einem anderen Benutzer mit oder ohne Authentifizierung verwenden Sie in Ihrer Anwendung eine der folgenden APIs:

Option	**Bezeichnung**
Anwendung	**API**
CLI/ODBC	SQLSetConnectAttr
XA CLI/ODBC	SQLSetConnectAttr
JAVA	getDB2Connection, reuseDB2Connection
.NET	DB2Connection.ConnectionString Schlüsselwörter: TrustedContextSystemUserID und TrustedContextSystemPassword

   Der Wechsel kann mit oder ohne Authentifizierung der neuen Benutzer-ID erfolgen. Dies hängt von der Definition des Objekts für den gesicherten Kontext

ab, das der expliziten gesicherten Verbindung zugeordnet wird. Nehmen Sie zum Beispiel an, dass der Sicherheitsadministrator das folgende Objekt für einen gesicherten Kontext erstellt:

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID BENUTZER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR BENUTZER2 WITH AUTHENTICATION,
               BENUTZER3 WITHOUT AUTHENTICATION
  ENABLE
```

Nehmen Sie weiterhin an, dass eine explizite gesicherte Verbindung hergestellt wird. In diesem Fall ist ein Wechsel der Benutzer-ID für die gesicherte Verbindung zu BENUTZER3 ohne die Bereitstellung von Authentifizierungsdaten zulässig, weil BENUTZER3 als ein Benutzer des gesicherten Kontextes CTX1 definiert wurde, für den keine Authentifizierung erforderlich ist. Eine Anforderung zum Wechseln der Benutzer-ID für die gesicherte Verbindung zu BENUTZER2 schlägt dagegen fehl, wenn keine Authentifizierungsdaten angegeben werden, weil BENUTZER2 als ein Benutzer des gesicherten Kontextes CTX1 definiert ist, für den Authentifizierungsdaten angegeben werden müssen.

Beispiel für das Herstellen einer expliziten gesicherten Verbindung und den Wechsel des Benutzers

Im folgenden Beispiel muss ein Server der Mittelschicht Anforderungen für einen Endbenutzer absetzen, verfügt jedoch über keinen Zugriff auf die Berechtigungsnachweise des Endbenutzers, um eine Datenbankverbindung für diesen Endbenutzer herzustellen.

Sie können ein Objekt für einen gesicherten Kontext auf dem Datenbankserver erstellen, mit dem der Server der Mittelschicht eine explizite gesicherte Verbindung zur Datenbank herstellen kann. Nach der Herstellung einer expliziten gesicherten Verbindung kann der Server der Mittelschicht die aktuelle Benutzer-ID der Verbindung wechseln und eine neue Benutzer-ID angeben, ohne dass die neue Benutzer-ID auf dem Datenbankserver authentifiziert werden muss. Das folgende CLI-Codefragment zeigt, wie eine gesicherte Verbindung mit dem gesicherten Kontext MYTCX hergestellt werden kann, der in Schritt 1 definiert wurde, und wie der Benutzer der gesicherten Verbindung ohne Authentifizierung gewechselt werden kann.

```
int main(int argc, char *argv[])
{
 SQLHANDLE henv;         /* Umgebungskennung */
 SQLHANDLE hdbc1;       /* Verbindungskennung */
 char origUserid[10] = "newton";
  char password[10] = "test";
   char switchUserid[10] = "zurbie";
 char dbName[10] = "testdb";

// Kennungen zuordnen
SQLAllocHandle( SQL_HANDLE_ENV, &henv );
SQLAllocHandle( SQL_HANDLE_DBC, &hdbc1 );

// Attribut für gesicherte Verbindung festlegen
SQLSetConnectAttr( hdbc1, SQL_ATTR_USE_TRUSTED_CONTEXT,
SQL_TRUE, SQL_IS_INTEGER );

// Gesicherte Verbindung einrichten
SQLConnect( hdbc1, dbName, SQL_NTS, origUserid, SQL_NTS,
password, SQL_NTS );

// Einige Arbeitsoperationen unter Benutzer-ID "newton" ausführen
```

```
. . . . . .

 // Commit durchführen
SQLEndTran(SQL_HANDLE_DBC, hdbc1, SQL_COMMIT);


// Benutzer-ID der gesicherten wechseln
SQLSetConnectAttr( hdbc1,
SQL_ATTR_TRUSTED_CONTEXT_USERID, switchUserid,
SQL_IS_POINTER
);

//Neue Arbeitsoperationen unter Benutzer-ID "zurbie" ausführen
. . . . . .

// Commit durchführen
SQLEndTranSQL_HANDLE_DBC, hdbc1, SQL_COMMIT);

// Datenbankverbindung trennen
SQLDisconnect( hdbc1 );

 return 0;

}  /* Ende von 'main' */
```

Tatsächlicher Zeitpunkt des Wechsels der Benutzer-ID

Nachdem der Befehl zum Wechseln des Benutzers auf der gesicherten Verbindung abgesetzt wurde, wird die Benutzerwechselanforderung erst ausgeführt, wenn die nächste Anweisung an den Server gesendet wird. Dies wird im folgenden Beispiel veranschaulicht, in dem der Befehl list applications die ursprüngliche Benutzer-ID anzeigt, bis die nächste Anweisung abgesetzt wird.

1. Stellen Sie eine explizite gesicherte Verbindung mit BENUTZERID1 her.
2. Setzen Sie den Befehl zum Wechseln des Benutzers ab, wie zum Beispiel getDB2Connection für BENUTZERID2.
3. Führen Sie den Befehl `db2 list applications` aus. Die Ausgabe des Befehls zeigt weiterhin an, dass BENUTZERID1 verbunden ist.
4. Setzen Sie eine Anweisung über die gesicherte Verbindung, wie zum Beispiel `executeQuery("values current sqlid")` ab, sodass die Benutzerwechselanforderung auf dem Server ausgeführt wird.
5. Führen Sie den Befehl `db2 list applications` erneut aus. Die Ausgabe des Befehls zeigt jetzt an, dass BENUTZERID2 verbunden ist.

Gesicherte Kontexte und Verbindungen

Ein gesicherter Kontext ist ein Datenbankobjekt, das eine Vertrauensbeziehung für eine Verbindung zwischen der Datenbank und einer externen Entität wie beispielsweise einem Anwendungsserver definiert.

Die Vertrauensbeziehung basiert auf der folgenden Gruppe von Attributen:

- Systemberechtigungs-ID: Ist dem Benutzer zugeordnet, der eine Datenbankverbindung herstellt.
- IP-Adresse (oder Domänenname): Ist dem Host zugeordnet, über den eine Datenbankverbindung hergestellt wird.
- Datenstromverschlüsselung: Stellt die Verschlüsselungseinstellung (sofern vorhanden) für die Datenübertragung zwischen dem Datenbankserver und dem Datenbankclient dar.

Wenn ein Benutzer eine Datenbankverbindung herstellt, überprüft das DB2-Datenbanksystem, ob die Verbindung mit der Definition eines Objekts für einen gesicherten Kontext in der Datenbank übereinstimmt. Wenn eine Übereinstimmung vorhanden ist, dann wird die Datenbankverbindung als gesichert eingestuft.

Eine gesicherte Verbindung ermöglicht dem Initiator dieser Verbindung das Anfordern weiterer Funktionen, die außerhalb der gesicherten Verbindung möglicherweise nicht verfügbar sind. Diese zusätzlichen Funktionen können abhängig davon, ob es sich um eine explizite oder implizite gesicherte Verbindung handelt, variieren.

Der Initiator einer expliziten gesicherten Verbindung verfügt über folgende Möglichkeiten:

- Wechseln der aktuellen Benutzer-ID der Verbindung und Angabe einer anderen Benutzer-ID mit oder ohne Authentifizierung.
- Anfordern zusätzlicher Zugriffsrechte über die Funktion für die Rollenvererbung in gesicherten Kontexten.

Eine implizite gesicherte Verbindung stellt eine gesicherte Verbindung dar, die nicht explizit angefordert wurde. Die implizite gesicherte Verbindung resultiert aus einer normalen Verbindungsanforderung und nicht aus der Anforderung einer expliziten gesicherten Verbindung. Zum Anfordern einer impliziten Verbindung sind keine Änderungen am Anwendungscode erforderlich. Darüber hinaus hat die Entscheidung zum Anfordern einer impliziten gesicherten Verbindung oder eines anderen Verbindungstyps keine Auswirkungen auf den Verbindungsrückkehrcode (wenn Sie eine explizite gesicherte Verbindung anfordern, gibt der Verbindungsrückkehrcode an, ob die Anforderung erfolgreich ausgeführt werden konnte oder nicht). Der Initiator einer impliziten gesicherten Verbindung kann zusätzliche Zugriffsrechte nur über die Funktion für die Rollenvererbung in gesicherten Kontexten anfordern. Ein Wechsel der Benutzer-ID ist hingegen nicht möglich.

Verbesserte Sicherheit durch Verwendung gesicherter Kontexte

Das auf drei Schichten basierende Anwendungsmodell erweitert das standardmäßige zweischichtige Client- und Servermodell durch eine Mittelschicht, die zwischen der Clientanwendung und dem Datenbankserver eingefügt wird. In den letzten Jahren fand dieses Modell mit dem verstärkten Einsatz webbasierter Technologien und der J2EE-Plattform (J2EE = Java 2 Enterprise Edition) zunehmend Verbreitung. Als Beispiel für ein Softwareprodukt, das das dreischichtige Anwendungsmodell unterstützt, kann IBM WebSphere Application Server (WAS) aufgeführt werden.

In einem dreischichtigen Anwendungsmodell ist die Mittelschicht für die Authentifizierung der Benutzer verantwortlich, die die Clientanwendungen ausführen, und darüber hinaus auch für die Verwaltung der Interaktion mit dem Datenbankserver. Traditionell erfolgt die gesamte Interaktion mit dem Datenbankserver über eine Datenbankverbindung, die von der Mittelschicht mithilfe einer Kombination aus Benutzer-ID und einem Berechtigungsnachweis hergestellt wird, mit der die Mittelschicht gegenüber dem Datenbankserver identifiziert wird. Dies bedeutet, dass der Datenbankserver für alle Berechtigungsprüfungs- und Protokollierungsoperationen, die für den Datenbankzugriff erforderlich sind, die Datenbankzugriffsrechte verwendet, die der Benutzer-ID der Mittelschicht zugeordnet sind. Dies gilt auch für die Zugriffsoperationen der Mittelschicht für einen Benutzer.

Obwohl das dreischichtige Anwendungsmodell zahlreiche Vorteile aufweist, verursacht die Ausführung der gesamten Interaktion mit dem Datenbankserver (z. B.

eine Benutzeranforderung) unter der Berechtigungs-ID der Mittelschicht jedoch verschiedene Sicherheitsrisiken, die wie folgt zusammengefasst werden können:

- Verlust der Benutzeridentität

  Für bestimmte Unternehmen ist es aus Gründen der Zugriffskontrolle wichtig, die Identität des Benutzers zu kennen, der auf die Datenbank zugreift.

- Reduzierte Benutzerverantwortlichkeit

  Die Möglichkeit zur Feststellung der Verantwortlichkeit mithilfe entsprechender Protokollierungsoperationen ist ein Grundprinzip der Datenbanksicherheit. Dadurch, dass die Benutzeridentität nicht bekannt ist, wird die Unterscheidung von Transaktionen der Mittelschicht für eigene Zwecke und Transaktionen der Mittelschicht für einen Benutzer erheblich erschwert.

- Gewährung nicht erforderlicher Zugriffsrechte für die Berechtigungs-ID der Mittelschicht

  Die Berechtigungs-ID der Mittelschicht muss über alle Zugriffsrechte verfügen, die zur Ausführung sämtlicher Anforderungen aller Benutzer erforderlich sind. Hierdurch entsteht das Sicherheitsproblem, dass Benutzern, die keinen Zugriff auf bestimmte Informationen benötigen, dieser Zugriff dennoch gewährt wird.

- Beeinträchtigte Sicherheit

  Zusätzlich zum Problem der Zugriffsrechte, das im vorherigen Listenpunkt erläutert wurde, ist es bei diesem Ansatz erforderlich, dass die von der Mittelschicht für die Herstellung von Verbindungen verwendete Berechtigungs-ID über Zugriffsrechte für alle Ressourcen verfügen muss, auf die mit Benutzeranforderungen zugegriffen werden kann. Wenn diese Berechtigungs-ID der Mittelschicht in ihrer Sicherheit beeinträchtigt wird, dann sind alle diese Ressourcen einem erhöhten Sicherheitsrisiko ausgesetzt.

- ″Überlauf″ zwischen Benutzern derselben Verbindung

  Änderungen eines vorherigen Benutzers können sich auf den aktuellen Benutzer auswirken.

Es besteht die Notwendigkeit zur Einrichtung von Verfahren, durch die die Identität und die Datenbankzugriffsrechte des aktuellen Benutzers für Datenbankanforderungen verwendet werden können, die von der Mittelschicht für diesen Benutzer ausgeführt werden. Der einfachste Ansatz zur Erreichung dieser Zielsetzung besteht darin, dass die Mittelschicht mithilfe der Benutzer-ID und des zugehörigen Kennworts eine neue Verbindung herstellt und die Benutzeranforderungen dann über diese Verbindung leitet. Obwohl dieser Ansatz einfach ist, birgt er verschiedene Nachteile, die im Folgenden aufgeführt sind:

- Nichtanwendbarkeit für bestimmte Mittelschichten. Zahlreiche Server der Mittelschicht verfügen nicht über die Berechtigungsnachweise für die Benutzerauthentifizierung, die für die Herstellung einer Verbindung erforderlich sind.
- Leistungseinbußen. Mit der Herstellung einer neuen physischen Verbindung und der erneuten Authentifizierung des Benutzers auf dem Datenbankserver sind gewisse Leistungseinbußen verbunden.
- Verwaltungsaufwand. In bestimmten Fällen, in denen keine zentrale Sicherheitsfunktion definiert und eingesetzt wird oder in denen die einmalige Anmeldung (Single Sign-on) nicht verwendet wird, entsteht durch das Vorhandensein von zwei Benutzerdefinitionen (in der Mittelschicht und auf dem Server) ein erhöhter Verwaltungsaufwand. Hierdurch ist es erforderlich, das Kennwort an unterschiedlichen Stellen zu ändern.

Dieses Problem kann mit der Funktion für gesicherte Kontexte behoben werden. Der Sicherheitsadministrator kann in der Datenbank ein Objekt für einen gesicherten Kontext erstellen, mit dem eine Vertrauensbeziehung zwischen der Datenbank und der Mittelschicht definiert wird. Die Mittelschicht kann dann eine explizite

gesicherte Verbindung zur Datenbank herstellen, wodurch sie die Möglichkeit erhält, die aktuelle Benutzer-ID für die Verbindung mit oder ohne Authentifizierung zu wechseln und eine andere Benutzer-ID zu verwenden. Zusätzlich zur Behebung des Problems mit der Identitätsprüfung für den Endbenutzer bieten gesicherte Kontexte einen weiteren Vorteil. Mit gesicherten Kontexten können Sie steuern, wann einem Datenbankbenutzer ein Zugriffsrecht gewährt wird. Wenn diese Möglichkeit nicht besteht, dann kann dies das gesamte Sicherheitskonzept des Systems beeinträchtigen. Zugriffsrechte können z. B. zu anderen Zwecken als den ursprünglich vorgesehenen verwendet werden. Der Sicherheitsadministrator kann einer Rolle ein oder auch mehrere Zugriffsrechte zuordnen und diese Rolle dann einem Objekt für einen gesicherten Kontext zuordnen. Nur gesicherte Datenbankverbindungen (explizit oder implizit), die mit der Definition dieses gesicherten Kontextes übereinstimmen, können die Zugriffsrechte, die dieser Rolle zugeordnet sind, nutzen.

Verbessern der Leistung

Bei Verwendung gesicherter Verbindungen können Sie die Leistung aufgrund der folgenden Vorteile maximieren:

- Beim Wechsel der aktuellen Benutzer-ID der Verbindung wird keine neue Verbindung hergestellt.
- Wenn in der Definition des gesicherten Kontextes die Authentifizierung der Benutzer-ID, zu der gewechselt werden soll, nicht als erforderlich definiert ist, dann entsteht der mit der Authentifizierung eines neuen Benutzers auf dem Datenbankserver verbundene Systemaufwand nicht.

Beispiel zum Erstellen eines gesicherten Kontextes

Der Sicherheitsadministrator erstellt das folgende Objekt eines gesicherten Kontextes:

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID BENUTZER2
  ATTRIBUTES (ADDRESS   '192.0.2.1')
  DEFAULT ROLE managerRole
   ENABLE
```

Wenn Benutzer *benutzer1* eine gesicherte Verbindung über die IP-Adresse 192.0.2.1 anfordert, gibt das DB2-Datenbanksystem eine Warnung (SQLSTATE 01679, SQLCODE +20360) zurück, um anzuzeigen, dass keine gesicherte Verbindung hergestellt werden konnte und dass dem Benutzer *benutzer1* nun eine nicht gesicherte Verbindung zugeordnet wurde. Wenn hingegen der Benutzer *benutzer2* eine gesicherte Verbindung über die IP-Adresse 192.0.2.1 anfordert, wird diese Anforderung ausgeführt, weil die Verbindungsattribute vom gesicherten Kontext CTX1 erfüllt werden. Nachdem der Benutzer *benutzer2* nun eine gesicherte Verbindung hergestellt hat, kann er nun alle Zugriffsrechte und Berechtigungen anfordern, die der Rolle managerRole des gesicherten Kontextes zugeordnet sind. Diese Zugriffsrechte und Berechtigungen stehen dem Benutzer *benutzer2* außerhalb des Geltungsbereichs dieser gesicherten Verbindung möglicherweise nicht zur Verfügung.

Übernahme der Rollenzugehörigkeit über einen gesicherten Kontext

Der aktuelle Benutzer einer gesicherten Verbindung kann mithilfe der automatischen Rollenübernahme über den gesicherten Kontext zusätzliche Zugriffsrechte anfordern, wenn diese Möglichkeit vom Sicherheitsadministrator in der Definition des relevanten gesicherten Kontextes so festgelegt wurde.

Eine Rolle kann standardmäßig von allen Benutzern einer gesicherten Verbindung übernommen werden. Der Sicherheitsadministrator kann die Definition des gesicherten Kontextes auch verwenden, um eine Rolle anzugeben, die nur von ganz bestimmten Benutzern übernommen werden soll.

Die aktiven Rollen, über die eine Sitzungsberechtigungs-ID verfügen kann, während sie über eine gesicherte Verbindung verfügt, lauten wie folgt:

- Die Rollen, die der Sitzungsberechtigungs-ID normalerweise zugeordnet sind, sowie
- entweder die Standardrolle des gesicherten Kontextes oder die Rolle eines speziellen Benutzers des gesicherten Kontextes, sofern diese definiert sind.

Anmerkung:

- Wenn Sie die Benutzerauthentifizierung mithilfe eines angepassten Sicherheits-Plug-ins konfigurieren, das so konzipiert ist, dass die Systemberechtigungs-ID und die Sitzungsberechtigungs-ID, die bei erfolgreicher Verbindungsherstellung von diesem Plug-in erzeugt werden, nicht identisch sind, kann die Rolle eines gesicherten Kontextes nicht über diese Verbindung übernommen werden. Dies gilt auch dann, wenn es sich hierbei um eine gesicherte Verbindung handelt.
- Zugriffsrechte gesicherter Kontexte, die über eine Rolle angefordert wurden, gelten ausschließlich für dynamische DML-Operationen. Sie gelten hingegen nicht für die folgenden Komponenten:
  - DDL-Operationen
  - Operationen mit nicht dynamischem SQL (Operationen mit statischen SQL-Anweisungen wie z. B. BIND und REBIND sowie implizite Rebinds, inkrementelle Bindungen usw.)

Anfordern benutzerspezifischer Zugriffsrechte im gesicherten Kontext

Der Sicherheitsadministrator kann die Definition des gesicherten Kontextes verwenden, um dem gesicherten Kontext bestimmte Rollen zuzuordnen, sodass Folgendes gilt:

- Alle Benutzer der gesicherten Verbindung können standardmäßig eine angegebene Rolle übernehmen.
- Bestimmte Benutzer der gesicherten Verbindung können eine angegebene Rolle übernehmen.

Wenn beim Benutzer einer gesicherten Verbindung zu einer neuen Berechtigungs-ID gewechselt wird und für diese neue Berechtigungs-ID eine benutzerspezifische Rolle des gesicherten Kontextes vorhanden ist, dann setzt die benutzerspezifische Rolle die Standardrolle des gesicherten Kontextes außer Kraft, sofern eine solche Rolle vorhanden ist. Diese Vorgehensweise wird im folgenden Beispiel dargestellt.

Beispiel zum Erstellen eines gesicherten Kontextes, in dem eine Standardrolle und eine benutzerspezifische Rolle zugeordnet wird

Der Sicherheitsadministrator erstellt das folgende Objekt eines gesicherten Kontextes:

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID BENUTZER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
```

```
  WITH USE FOR BENUTZER2 WITH AUTHENTICATION,
               BENUTZER3 WITHOUT AUTHENTICATION
  DEFAULT ROLE AUDITOR
  ENABLE
```

Wenn BENUTZER1 eine gesicherte Verbindung herstellt, werden die Zugriffsrechte, die der Rolle AUDITOR gewährt wurden, von dieser Berechtigungs-ID übernommen. Diese Zugriffsrechte werden auch von BENUTZER3 übernommen, wenn die aktuelle Berechtigungs-ID für die gesicherte Verbindung gewechselt und dann die Benutzer-ID von BENUTZER3 verwendet wird. (Wenn die Benutzer-ID der Verbindung zu einem bestimmten Zeitpunkt gewechselt und dann die Benutzter-ID von BENUTZER2 verwendet wird, dann übernimmt BENUTZER2 auch die Standardrolle des gesicherten Kontextes, AUDITOR.) Der Sicherheitsadministrator kann angeben, dass BENUTZER3 eine andere Rolle als die Standardrolle des gesicherten Kontextes übernehmen soll. Hierzu kann diesem Benutzer wie folgt eine bestimmte Rolle zugeordnet werden:

```
CREATE TRUSTED CONTEXT CTX1
 BASED UPON CONNECTION USING SYSTEM AUTHID BENUTZER1
 ATTRIBUTES (ADDRESS   '192.0.2.1')
  WITH USE FOR BENUTZER2 WITH AUTHENTICATION,
               BENUTZER3 WITHOUT AUTHENTICATION ROLE OTHER_ROLE
  DEFAULT ROLE AUDITOR
  ENABLE
```

Wenn die aktuelle Benutzer-ID der gesicherten Verbindung gewechselt und dann die Benutzer-ID von BENUTZER3 verwendet wird, übernimmt dieser Benutzer nicht mehr die Standardrolle des gesicherten Kontextes. Stattdessen wird die spezielle Rolle OTHER_ROLE übernommen, die dem Benutzer vom Sicherheitsadministrator zugeordnet wurde.

Regeln zum Wechseln der Benutzer-ID bei einer expliziten gesicherten Verbindung

Bei einer expliziten gesicherten Verbindung können Sie die Benutzer-ID der Verbindung wechseln. Es sind bestimmte Regeln zu beachten.

1. Wenn die Anforderung zum Wechseln nicht über eine explizite gesicherte Verbindung abgesetzt wurde und die Anforderung zum Wechseln zur Verarbeitung an den Server gesendet wird, dann wird die Verbindung geschlossen und das System gibt eine Fehlernachricht (SQLSTATE 08001, SQLCODE -30082 mit Ursachencode 41) zurück.
2. Wenn die Anforderung zum Wechseln nicht an einer Transaktionsgrenze abgesetzt wird, wird die Transaktion mit einem Rollback zurückgesetzt und die Anforderung zum Wechseln wird zur Verarbeitung an den Server gesendet. Die Verbindung wird in den Status 'Nicht verbunden' versetzt und das System gibt eine Fehlernachricht (SQLSTATE 58009, SQLCODE -30020) zurück.
3. Wenn die Anforderung zum Wechseln innerhalb einer gespeicherten Prozedur abgesetzt wird, gibt das System eine Fehlernachricht (SQLCODE -30090, Ursachencode 29) zurück, in der darauf hingewiesen wird, dass diese Operation in der vorliegenden Umgebung nicht zulässig ist. Der Verbindungsstatus wird beibehalten und die Verbindung wird nicht in den Status 'Nicht verbunden' versetzt. Nachfolgende Anforderungen können verarbeitet werden.
4. Wenn die Anforderung zum Wechseln über eine Instanzverbindung (und nicht über eine Datenbankverbindung) geleitet wird, dann wird diese Verbindung geschlossen und das System gibt eine Fehlernachricht (SQLCODE -30005) zurück.

5. Wenn die Anforderung zum Wechseln mit einer Berechtigungs-ID ausgeführt wird, die auf der gesicherten Verbindung nicht zulässig ist, wird der Fehler (SQLSTATE 42517, SQLCODE -20361) zurückgegeben und die Verbindung wird in den Status 'Nicht verbunden' versetzt.
6. Wenn die Anforderung zum Wechseln mit einer Berechtigungs-ID ausgeführt wird, die auf der gesicherten Verbindung mit der Einstellung WITH AUTHENTICATION zulässig ist, das erforderliche Authentifizierungstoken jedoch nicht bereitgestellt wird, wird ein Fehler (SQLSTATE 42517, SQLCODE -20361) zurückgegeben und die Verbindung wird in den Status 'Nicht verbunden' versetzt.
7. Wenn das Objekt für den gesicherten Kontext, das der gesicherten Verbindung zugeordnet ist, inaktiviert wurde und eine Anforderung zum Wechseln für diese gesicherte Verbindung abgesetzt wird, dann gibt das System einen Fehler (SQLSTATE 42517, SQLCODE -20361) zurück und die Verbindung wird in den Status 'Nicht verbunden' versetzt.

   In diesem Fall werden nur Benutzerwechselanforderungen akzeptiert, in denen die Benutzer-ID, über die die gesicherte Verbindung hergestellt wurde, oder die Benutzer-ID NULL angegeben ist. Wenn zu der Benutzer-ID gewechselt wird, mit der die gesicherte Verbindung hergestellt wurde, dann übernimmt diese Benutzer-ID keine der für den gesicherten Kontext definierten Rollen (d. h. weder die Standardrolle des gesicherten Kontextes noch seine benutzerspezifische Rolle).
8. Wenn das Attribut für die Systemberechtigungs-ID des Objekts für den gesicherten Kontext, das der gesicherten Verbindung zugeordnet ist, geändert wurde und eine Anforderung zum Wechseln für diese gesicherte Verbindung abgesetzt wird, dann gibt das System einen Fehler (SQLSTATE 42517, SQLCODE -20361) zurück und die Verbindung wird in den Status 'Nicht verbunden' versetzt.

   In diesem Fall werden nur Benutzerwechselanforderungen akzeptiert, in denen die Benutzer-ID, über die die gesicherte Verbindung hergestellt wurde, oder die Benutzer-ID NULL angegeben ist. Wenn zu der Benutzer-ID gewechselt wird, mit der die gesicherte Verbindung hergestellt wurde, dann übernimmt diese Benutzer-ID keine der für den gesicherten Kontext definierten Rollen (d. h. weder die Standardrolle des gesicherten Kontextes noch seine benutzerspezifische Rolle).
9. Wenn das Objekt für den gesicherten Kontext, das der gesicherten Verbindung zugeordnet ist, gelöscht wird und eine Anforderung zum Wechseln für diese gesicherte Verbindung abgesetzt wird, dann gibt das System einen Fehler (SQLSTATE 42517, SQLCODE -20361) zurück und die Verbindung wird in den Status 'Nicht verbunden' versetzt.

   In diesem Fall werden nur Benutzerwechselanforderungen akzeptiert, in denen die Benutzer-ID, über die die gesicherte Verbindung hergestellt wurde, oder die Benutzer-ID NULL angegeben ist. Wenn zu der Benutzer-ID gewechselt wird, mit der die gesicherte Verbindung hergestellt wurde, dann übernimmt diese Benutzer-ID keine der für den gesicherten Kontext definierten Rollen (d. h. weder die Standardrolle des gesicherten Kontextes noch seine benutzerspezifische Rolle).
10. Wenn die Anforderung zum Wechseln mit einer Benutzer-ID ausgeführt wird, die auf der gesicherten Verbindung zulässig ist, diese Benutzer-ID jedoch nicht über das Zugriffsrecht CONNECT für die Datenbank verfügt, wird die Verbindung in den Status 'Nicht verbunden' versetzt und das System gibt eine Fehlernachricht (SQLSTATE 08004, SQLCODE -1060) zurück.

11. Wenn die Systemberechtigungs-ID des gesicherten Kontextes in der Klausel WITH USE FOR aufgeführt ist, dann berücksichtigt das DB2-Datenbanksystem die Authentifizierungseinstellung für die Systemberechtigungs-ID bei Benutzerwechselanforderungen, mit denen zurück zu dieser Systemberechtigungs-ID gewechselt werden soll. Wenn die Systemberechtigungs-ID des gesicherten Kontextes in der Klausel WITH USE FOR nicht aufgeführt wird, dann ist die Benutzerwechselanforderung zum Zurückwechseln zur Systemberechtigungs-ID immer zulässig. Dies gilt auch dann, wenn keine Authentifizierung durchgeführt wird.

Anmerkung: Wenn die Verbindung in den Status 'Nicht verbunden' versetzt wird, dann werden nur die folgenden Anforderungen vom System akzeptiert und führen nicht zur Ausgabe des Fehlers "Der Anwendungsstatus ist fehlerhaft. Die Verbindung zur Datenbank ging verloren." (SQLCODE -900):

- Benutzerwechselanforderung
- COMMIT- oder ROLLBACK-Anweisung
- DISCONNECT-, CONNECT RESET- oder CONNECT-Anforderung

Anmerkung: Wenn die Benutzer-ID der gesicherten Verbindung gewechselt wird, dann gehen alle Traces für die Verbindungsumgebung, die zu der alten Benutzer-ID gehören, verloren. Dies bedeutet, dass der Wechsel von Benutzer-IDs zu einer Umgebung führt, die einer neuen Verbindungsumgebung entspricht. Wenn die alte Benutzer-ID der Verbindung beispielsweise über temporäre Tabellen oder geöffnete WITH HOLD-Cursor verfügte, gehen diese Objekte vollständig verloren, wenn die Benutzer-ID für diese Verbindung gewechselt und eine neue Benutzer-ID definiert wird.

Fehlerbestimmung für gesicherte Kontexte

Eine explizite gesicherte Verbindung ist eine Verbindung, die mithilfe einer bestimmten, expliziten Anforderung für eine gesicherte Verbindung erfolgreich hergestellt werden kann. Wenn Sie eine explizite gesicherte Verbindung anfordern und nicht über die hierfür erforderlichen Berechtigungen verfügen, wird Ihnen eine reguläre Verbindung zugeteilt und das System gibt eine Warnung (+20360) aus. Um festzustellen, warum ein Benutzer keine gesicherte Verbindung herstellen konnte, muss der Sicherheitsadministrator die Definition des gesicherten Kontextes in den Systemkatalogen und die Verbindungsattribute prüfen.

Dabei müssen insbesondere die IP-Adresse, über die die Verbindung hergestellt werden soll, die Verschlüsselungsstufe des Datenstroms oder Netzes sowie die Systemberechtigungs-ID berücksichtigt werden, die zur Herstellung der Verbindung verwendet wird. Die Option -application des Dienstprogramms db2pd gibt diese sowie die folgenden zusätzlichen Informationen zurück:

- Sicherungstyp der Verbindung: Dieser Typ gibt an, ob die Verbindung gesichert ist oder nicht. Bei gesicherten Verbindungen enthält diese Angabe auch die Information, ob es sich um eine explizite oder implizite gesicherte Verbindung handelt.
- Name des gesicherten Kontextes: Der Name des gesicherten Kontextes, der der gesicherten Verbindung zugeordnet ist.
- Übernommene Rolle: Die Rolle, die durch die gesicherte Verbindung übernommen wurde.

Die folgenden Ursachen für Fehler beim Abrufen einer expliziten gesicherten Verbindung treten am häufigsten auf:

- Die Clientanwendung verwendet nicht TCP/IP zur Kommunikation mit dem DB2-Server. TCP/IP ist das einzige unterstützte Protokoll für eine Clientanwendung, über das mit dem DB2-Server kommuniziert werden kann und das zur Herstellung einer gesicherten Verbindung (explizit oder implizit) genutzt werden kann.
- Der Authentifizierungstyp des Datenbankservers ist auf den Wert CLIENT eingestellt.
- Der Datenbankserver hat kein aktiviertes Objekt für einen gesicherten Kontext. Die Definition eines Objekts für einen gesicherten Kontext muss explizit ENABLE (Aktivieren) angeben, damit der entsprechende gesicherte Kontext beim Abgleichen der Attribute einer eingehenden Verbindung in Betracht gezogen wird.
- Die Objekte für gesicherte Kontexte auf dem Datenbankserver stimmen nicht mit den entsprechenden Attributen für die gesicherte Verbindung überein, die dargestellt werden. Es kann zum Beispiel eine der folgenden Situationen vorliegen:
  - Die Systemberechtigungs-ID der Verbindung entspricht keiner Systemberechtigungs-ID eines Objekts für einen gesicherten Objekt.
  - Die IP-Adresse, von der die Verbindung ausgeht, stimmt mit keiner IP-Adresse in dem Objekt für den gesicherten Kontext überein, das für die Verbindung in Betracht kommt.
  - Die von der Verbindung verwendete Datenstromverschlüsselung stimmt nicht mit dem Wert des Attributs ENCRYPTION in dem Objekt für den gesicherten Kontext überein, das für die Verbindung in Betracht gezogen wird.

  Mithilfe des Tools db2pd können Sie die IP-Adresse, von der aus die Verbindung hergestellt wurde, die Verschlüsselungsstufe des Datenstroms oder des Netzes, das von der Verbindung verwendet wird, und die Systemberechtigungs-ID, unter der die Verbindung hergestellt wird, ermitteln. Sie können die Katalogsichten SYSCAT.CONTEXTS und SYSCAT.CONTEXTATTRIBUTES abfragen, um die Definition eines bestimmten Objekts für einen gesicherten Kontext, wie zum Beispiel die Systemberechtigungs-ID, die Gruppe der zulässigen IP-Adressen und den Wert des Attributs ENCRYPTION des Objekts, zu ermitteln.

Die folgenden Ursachen für einen Benutzerwechselfehler treten am häufigsten auf:

- Die Benutzer-ID, zu der gewechselt werden soll, besitzt keine CONNECT-Zugriffsrechte für die Datenbank. In diesem Fall wird der Fehler SQL1060N zurückgegeben.
- Die Benutzer-ID, zu der gewechselt werden soll, oder die Gruppe PUBLIC ist in der Klausel WITH USE FOR des Objekts des gesicherten Kontexts, das der explizit gesicherten Verbindung zugeordnet ist, nicht definiert.
- Das Wechseln des Benutzers ist mit Authentifizierung zulässig, jedoch gibt der Benutzer keine Berechtigungsnachweise bzw. falsche Berechtigungsnachweise an.
- Eine Anforderung zum Benutzerwechsel wird nicht an einer Transaktionsgrenze abgesetzt.
- Der gesicherte Kontext, der einer gesicherten Verbindung zugeordnet ist, wurde inaktiviert, gelöscht oder geändert. In diesem Fall ist nur ein Wechsel zu der Benutzer-ID zulässig, unter der die gesicherte Verbindung hergestellt wurde.

Kapitel 4. Kennsatzbasierte Zugriffssteuerung (LBAC)

LBAC (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) bietet wesentlich bessere Kontrollmöglichkeiten für die Steuerung des Zugriffs auf Ihre Daten. Mit LBAC können Sie exakt festlegen, wer Schreibzugriff und wer Lesezugriff auf einzelne Zeilen und einzelne Spalten erhält.

Funktionsweise von LBAC

Die LBAC-Funktionalität ist umfassend konfigurierbar und lässt sich auf Ihre individuelle Sicherheitsumgebung abstimmen. Alle LBAC-Konfigurationsaktivitäten werden durch einen *Sicherheitsadministrator* ausgeführt. Dies ist ein Benutzer, dem vom Systemadministrator die Berechtigung SECADM erteilt wurde.

Ein Sicherheitsadministrator konfiguriert das LBAC-System, indem er Sicherheitskennsatzkomponenten erstellt. Eine *Sicherheitskennsatzkomponente* ist ein Datenbankobjekt, das eine Bedingung darstellt, mit dem Sie feststellen wollen, ob ein Benutzer auf bestimmte Daten zugreifen darf. Die Bedingung kann z. B. sein, dass der Benutzer einer bestimmten Abteilung angehört oder dass er an einem bestimmten Projekt mitarbeitet. Eine *Sicherheitsrichtlinie* (Security Policy) beschreibt die Bedingungen, die zur Entscheidung herangezogen werden, wer auf welche Daten Zugriff besitzt. Eine Sicherheitsrichtlinie enthält mindestens eine Sicherheitskennsatzkomponente. Eine Tabelle kann nur durch eine Sicherheitsrichtlinie geschützt werden, jedoch können verschiedene Tabellen durch verschiedene Sicherheitsrichtlinien geschützt werden.

Nach der Erstellung einer Sicherheitsrichtlinie erstellt ein Sicherheitsadministrator Objekte, die als *Sicherheitskennsätze* (Security Labels) bezeichnet werden und die Teil der jeweiligen Richtlinie sind. Sicherheitskennsätze enthalten Sicherheitskennsatzkomponenten. Die exakten Bestandteile eines Sicherheitskennsatzes werden durch die Sicherheitsrichtlinie bestimmt und lassen sich so konfigurieren, dass sie die Bedingungen darstellen, auf die sich in Ihrem Unternehmen die Entscheidung gründet, wer auf welche bestimmten Datenelemente Zugriff haben soll. Wenn Sie zum Beispiel festlegen, dass die Entscheidung, welche Daten für eine Person zugänglich sein sollen, von ihrer Position im Unternehmen sowie von den Projekten, an denen sie beteiligt ist, abhängen soll, können Sie Ihre Sicherheitskennsätze so konfigurieren, dass jeder Kennsatz diese Informationen beinhaltet. LBAC bietet die Flexibilität, alle Arten von Kriterien zu konfigurieren, welche die gesamte Bandbreite von sehr komplexen bis hin zu sehr einfachen Systemen, bei denen jeder Kennsatz nur entweder einen ″hohen″ oder einen ″niedrigen″ Vertraulichkeitsgrad darstellt, abdecken.

Nach der Erstellung kann ein Sicherheitskennsatz einzelnen Spalten und Zeilen in einer Tabelle zugeordnet werden, um die dort gespeicherten Daten zu schützen. Daten, die durch einen Sicherheitskennsatz geschützt sind, werden als *geschützte Daten* bezeichnet. Ein Sicherheitsadministrator berechtigt Benutzer zum Zugriff auf geschützte Daten, indem er ihnen Sicherheitskennsätze erteilt. Wenn ein Benutzer versucht, auf geschützte Daten zuzugreifen, wird sein Sicherheitskennsatz mit dem Sicherheitskennsatz verglichen, der die Daten schützt. Der schützende Kennsatz blockiert einige Sicherheitskennsätze und andere nicht.

Ein Benutzer, eine Rolle oder eine Gruppe kann Sicherheitskennsätze für mehrere Sicherheitsrichtlinien besitzen. Für eine bestimmte Sicherheitsrichtlinie kann ein

Benutzer, eine Rolle oder eine Gruppe jedoch höchstens einen Kennsatz für den Lesezugriff und einen Kennsatz für den Schreibzugriff besitzen.

Ein Sicherheitsadministrator kann Benutzern außerdem Freistellungen erteilen. Eine *Freistellung* (engl. exemption) erlaubt einem Benutzer den Zugriff auf geschützte Daten, der ihm ansonsten durch seine Sicherheitskennsätze verwehrt wäre. Sicherheitskennsätze und Freistellungen werden zusammengenommen als *LBAC-Berechtigungsnachweise* (engl. LBAC credentials) eines Benutzers bezeichnet.

Wenn Sie versuchen, auf eine geschützte Spalte zuzugreifen, auf die der Zugriff durch Ihre LBAC-Berechtigungsnachweise nicht zugelassen wird, schlägt der Zugriffsversuch fehl und Sie empfangen eine Fehlernachricht.

Wenn Sie versuchen, geschützte Zeilen zu lesen, auf die ihre LBAC-Berechtigungsnachweise keinen Lesezugriff zulassen, reagiert DB2 so, als ob diese Zeilen nicht vorhanden wären. Diese Zeilen können durch keine von Ihnen ausgeführte SQL-Anweisung (d. h. Anweisungen SELECT, UPDATE oder DELETE) ausgewählt werden. Auch Spaltenfunktion ignorieren Zeilen, auf die Ihre LBAC-Berechtigungsnachweise keinen Lesezugriff zulassen. Die Funktion COUNT(*) gibt zum Beispiel nur die Anzahl der Zeilen zurück, auf die Sie Lesezugriff haben.

Sichten und LBAC

Sie können eine Sicht für eine geschützte Tabelle in der gleichen Weise definieren wie für eine nicht geschützte Tabelle. Beim Zugriff auf eine solche Sicht wird der LBAC-Schutz für die zugrunde liegende Tabelle wirksam. Dazu werden die LBAC-Berechtigungsnachweise der jeweiligen Sitzungsberechtigungs-ID verwendet. Zwei Benutzern, die auf die gleiche Sicht zugreifen, werden abhängig von ihren LBAC-Berechtigungsnachweisen möglicherweise verschiedene Zeilen angezeigt.

Referenzielle Integritätsbedingungen und LBAC

Die folgenden Regeln erläutern, wie LBAC-Regeln im Hinblick auf vorhandene referenzielle Integritätsbedingungen umgesetzt werden:

- **Regel 1:** Die LBAC-Regeln für den Lesezugriff werden auf intern generierte Suchen in untergeordneten Tabellen NICHT angewendet. Dadurch soll verhindert werden, dass auf untergeordnete Tabellen nicht mehr zugegriffen werden kann (verwaiste Tabelle).
- **Regel 2:** Die Regeln für den Lesezugriff werden auf intern generierte Suchen in übergeordneten Tabellen NICHT angewendet.
- **Regeln 3:** Die LBAC-Regeln für den Schreibzugriff werden angewendet, wenn eine kaskadierende Operation an untergeordneten Tabellen ausgeführt wird. Wenn ein Benutzer zum Beispiel ein übergeordnetes Element löscht, jedoch keines der untergeordneten Elemente wegen Verstoßes gegen die LBAC-Schreibzugriffsregel löschen kann, sollte die Löschung rückgängig gemacht und ein Fehler zurückgegeben werden.

Speichersystemaufwand bei der Verwendung von LBAC

Wenn Sie LBAC zum Schutz einer Tabelle auf Zeilenebene verwenden, fällt zusätzlicher Speicheraufwand in Form von Kosten für die Spalte der Zeilensicherheitskennsätze an. Diese Kosten sind vom Typ des ausgewählten Sicherheitskennsatzes abhängig. Wenn Sie zum Beispiel eine Sicherheitsrichtlinie mit zwei Komponenten zum Schutz einer Tabelle erstellen, belegt ein Sicherheitskennsatz aus dieser Sicherheitsrichtlinie 16 Byte (8 Byte für jede Komponente).

Da die Spalte für die Zeilensicherheitskennsätze als VARCHAR-Spalte, die keine Nullwerte enthalten darf (NOT NULLABLE) behandelt wird, beträgt der Gesamtaufwand in diesem Fall 20 Byte pro Zeile. Im Allgemeinen beträgt der Gesamtaufwand pro Zeile (*N**8 + 4) Byte. Datei ist *N* die Anzahl der Komponenten in der Sicherheitsrichtlinie, die die Tabelle schützt.

Wenn Sie LBAC zum Schutz einer Tabelle auf Spaltenebene verwenden, gehört der Spaltensicherheitskennsatz zu den Metadaten (d. h. er wird zusammen mit den Metadaten der Spalte in der Katalogtabelle SYSCOLUMNS gespeichert). Diese Metadaten bestehen einfach aus der ID des Sicherheitskennsatzes, der die Spalte schützt. In diesem Fall fällt für die Benutzertabelle kein zusätzlicher Speicheraufwand an.

Von LBAC nicht ausgeführte Aktionen

- LBAC wird nie einen Zugriff auf Daten zulassen, der durch eine eignerdefinierte Zugriffssteuerung verboten ist.

  Beispiel: Wenn Sie keine Berechtigung zum Lesen einer Tabelle haben, wird Ihnen das Lesen von Daten aus dieser Tabelle, selbst für Zeilen und Spalten, auf die Ihnen LBAC ansonsten Zugriff gewähren würde, nicht erlaubt.
- Ihre LBAC-Berechtigungsnachweise schränken nur Ihren Zugriff auf geschützte Daten ein. Sie wirken sich nicht auf Ihren Zugriff auf ungeschützte Daten aus.
- LBAC-Berechtigungsnachweise werden nicht überprüft, wenn Sie eine Tabelle oder eine Datenbank löschen, selbst wenn die Tabelle bzw. die Datenbank geschützte Daten enthält.
- LBAC-Berechtigungsnachweise werden nicht überprüft, wenn Sie ein Backup Ihrer Daten erstellen. Wenn Sie ein Backup für eine Tabelle ausführen können, unterliegen die durch das Backup gesicherten Zeilen keinerlei Einschränkung durch den LBAC-Schutz der Daten. Darüber hinaus ist zu beachten, dass die Daten auf dem Backupdatenträger nicht durch LBAC geschützt sind. Nur Daten in der Datenbank werden geschützt.
- LBAC kann nicht zum Schutz der folgenden Typen von Tabellen verwendet werden:
  - Eine MQT (Materialized Query Table, gespeicherte Abfragetabelle)
  - Eine Tabelle, von der eine MQT abhängig ist
  - Eine Zwischenspeichertabelle
  - Eine Tabelle, von der eine Zwischenspeichertabelle abhängig ist
  - Eine typisierte Tabelle
- Der LBAC-Zugriffsschutz kann nicht auf einen Kurznamen angewendet werden.

LBAC-Lernprogramm

Ein Lernprogramm, das Sie in die Grundlagen der Verwendung von LBAC einführt, ist online verfügbar. Das Lernprogramm ist Teil der IBM developerWorks-Website (http://www.ibm.com/developerworks/db2) und trägt den Titel DB2 Label-Based Access Control, a practical guide.

LBAC-Sicherheitsrichtlinien

Der Sicherheitsadministrator definiert in einer Sicherheitsrichtlinie die Bedingungen, die bestimmen, wer Schreibzugriff und wer Lesezugriff auf einzelne Zeilen und einzelne Spalten von Tabellen erhält.

Eine Sicherheitsrichtlinie enthält folgende Informationen:

- Welche Sicherheitskennsatzkomponenten in den Sicherheitskennsätzen verwendet werden, die Teil der Richtlinie sind.
- Welche Regeln beim Vergleichen dieser Sicherheitskennsatzkomponenten verwendet werden.
- Welche von bestimmten optionalen Verhaltensweisen beim Zugriff auf Daten angewendet werden, die durch die Richtlinie geschützt werden.
- Welche zusätzlichen Sicherheitskennsätze und Freistellungen zu prüfen sind, wenn der Zugriff auf Daten, die durch die Sicherheitsrichtlinie geschützt werden, umgesetzt wird. Zum Beispiel wird die Option, ob die Sicherheitskennsätze zu prüfen sind, die Rollen und Gruppen erteilt wurden, durch die Sicherheitsrichtlinie gesteuert.

Jeder geschützten Tabelle darf eine und nur eine Sicherheitsrichtlinie zugeordnet werden. Zeilen und Spalten in einer solchen Tabelle können nur mit Sicherheitskennsätzen geschützt werden, die Teil dieser Sicherheitsrichtlinie sind. Jeglicher Zugriff auf geschützte Daten unterliegt den Regeln dieser Richtlinie. Sie können in einer Datenbank mehrere Sicherheitsrichtlinien haben, jedoch kann eine bestimmte Tabelle jeweils nur durch eine einzige Sicherheitsrichtlinie geschützt werden.

Erstellen einer Sicherheitsrichtlinie

Zur Erstellung einer Sicherheitsrichtlinie müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Eine Sicherheitsrichtlinie wird mithilfe der SQL-Anweisung CREATE SECURITY POLICY erstellt. Die in einer Sicherheitsrichtlinie aufgeführten Sicherheitskennsatzkomponenten müssen erstellt werden, bevor die Anweisung CREATE SECURITY POLICY ausgeführt wird. Die Reihenfolge, in der die Komponenten bei der Erstellung einer Sicherheitsrichtlinie aufgelistet werden, definiert keinerlei Art von Vorrangstellung oder irgendeine andere Beziehung zwischen den Komponenten. Es ist jedoch wichtig, diese Reihenfolge zu kennen, wenn Sicherheitskennsätze mit integrierten Funktionen wie SECLABEL erstellt werden.

Sie können von der Sicherheitsrichtlinie, die Sie erstellt haben, Sicherheitskennsätze erstellen, um Ihre Daten zu schützen.

Ändern einer Sicherheitsrichtlinie

Ein Sicherheitsadministrator kann eine Sicherheitsrichtlinie mithilfe der Anweisung ALTER SECURITY POLICY ändern.

Löschen einer Sicherheitsrichtlinie

Zum Löschen einer Sicherheitsrichtlinie müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Eine Sicherheitsrichtlinie wird mithilfe der SQL-Anweisung DROP gelöscht.

Sie können eine Sicherheitsrichtlinie nicht löschen, wenn Sie einer Tabelle zugeordnet (bzw. hinzugefügt) ist.

Komponenten von LBAC-Sicherheitskennsätzen - Übersicht

Eine *Sicherheitskennsatzkomponente* ist ein Datenbankobjekt, das Teil der LBAC-Funktionalität (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) ist. Mithilfe von Sicherheitskennsatzkomponenten können Sie die Sicherheitsstruktur Ihres Unternehmens modellieren.

Ein Sicherheitskennsatzkomponente kann ein beliebiges Kriterium darstellen, dass Sie möglicherweise zur Entscheidung heranziehen möchten, ob ein Benutzer Zugriff auf eine bestimmte Dateninformation haben soll. Typische Beispiele für solche Kriterien sind u. a.:

- Wie vertrauenswürdig der Benutzer ist.
- In welcher Abteilung der Benutzer tätig ist.
- Ob der Benutzer an einem bestimmten Projekt beteiligt ist.

Beispiel: Wenn Sie wollen, dass die Abteilung, der ein Benutzer angehört, Auswirkung auf die Daten hat, auf die der Benutzer zugreifen kann, können Sie eine Komponente mit dem Namen 'dept' erstellen und Elemente für diese Komponente definieren, die die verschiedenen Abteilungen in Ihrem Unternehmen benennen. Anschließend können Sie die Komponente 'dept' in Ihre Sicherheitsrichtlinie aufnehmen.

Ein *Element* einer Sicherheitskennsatzkomponente ist eine bestimmte "Einstellung", die für diese Komponente zulässig ist.

Beispiel: Eine Sicherheitskennsatzkomponente, die eine Vertraulichkeitsstufe darstellt, könnte zum Beispiel die folgenden vier Elemente haben: 'Top Secret' (Streng geheim), 'Secret' (Geheim), 'Classified' (Vertraulich) und 'Unclassified' (Nicht eingestuft).

Erstellen einer Sicherheitskennsatzkomponente

Zur Erstellung einer Sicherheitskennsatzkomponente müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Sicherheitskennsatzkomponenten werden mithilfe der SQL-Anweisung CREATE SECURITY LABEL COMPONENT erstellt.

Bei der Erstellung einer Sicherheitskennsatzkomponente müssen Sie folgende Informationen angeben:

- Einen Namen für die Komponente
- Den Typ der Komponente (ARRAY, TREE oder SET)
- Eine vollständige Liste aller zulässigen Elemente
- Für die Typen ARRAY und TREE: Eine Beschreibung, wie sich jedes einzelne Element in die Struktur der Komponente einfügt

Nachdem Sie Ihre Sicherheitskennsatzkomponenten erstellt haben, können Sie eine Sicherheitsrichtlinie auf der Basis dieser Komponenten erstellen. Sie können von dieser Sicherheitsrichtlinie Sicherheitskennsätze erstellen, um Ihre Daten zu schützen.

Typen von Komponenten

Es gibt drei Typen von Sicherheitskennsatzkomponenten:

- TREE: Jedes Element stellt einen Knoten in einer Baumstruktur dar.

- ARRAY: Jedes Element stellt einen Punkt auf einer linearen Skala dar.
- SET: Jedes Element stellt ein Element einer Menge dar.

Die Typen werden zur Modellierung der verschiedenen Möglichkeiten von Beziehungen zwischen den Elementen verwendet. Wenn Sie zum Beispiel eine Komponente erstellen, um eine oder mehrere Abteilungen in einem Unternehmen zu beschreiben, bietet sich wahrscheinlich der Komponententyp TREE an, da die meisten Unternehmensstrukturen die Form einer Baumstruktur besitzen. Wenn Sie eine Komponente erstellen, die die Vertraulichkeitsstufe darstellt, den eine Person besitzt, würden Sie wahrscheinlich den Komponententyp ARRAY verwenden, da in der Regel für je zwei Vertraulichkeitsgrade stets einer höher als der andere ist.

Detaillierte Informationen zu den einzelnen Typen, einschließlich ausführlicher Beschreibungen der Beziehungen, die zwischen den Elementen bestehen können, finden Sie in den jeweiligen Beschreibungen zu den Typen.

Ändern der Komponenten von Sicherheitskennsätzen

Der Sicherheitsadministrator kann eine Sicherheitskennsatzkomponente mithilfe der Anweisung ALTER SECURITY LABEL COMPONENT ändern.

Löschen einer Sicherheitskennsatzkomponente

Zum Löschen einer Sicherheitskennsatzkomponente müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Eine Sicherheitskennsatzkomponente wird mithilfe der SQL-Anweisung DROP gelöscht.

Typ der LBAC-Sicherheitskennsatzkomponente: SET

SET ist ein Typ von Sicherheitskennsatzkomponente, der in einer LBAC-Sicherheitsrichtlinie (LBAC - kennsatzbasierte Zugriffssteuerung) verwendet werden kann.

Komponenten des Typs SET sind ungeordnete Listen von Elementen. Die einzige Vergleichsoperation, die für Elemente dieses Komponententyps durchgeführt werden kann, ist die Prüfung, ob ein bestimmtes Element in der Liste enthalten ist oder nicht.

Typ der LBAC-Sicherheitskennsatzkomponente: ARRAY

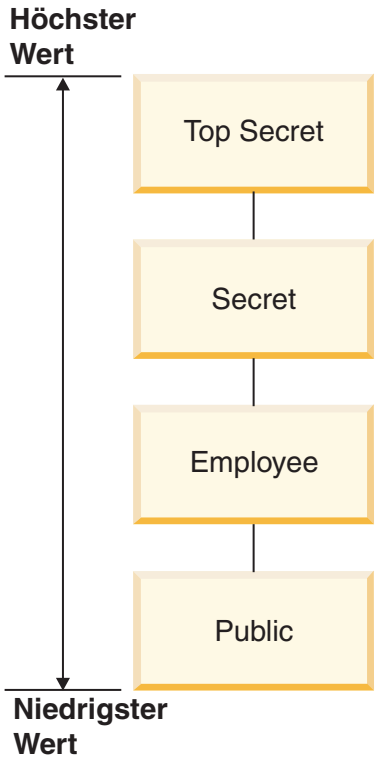
ARRAY ist ein Typ von Sicherheitskennsatzkomponente.

Bei dem Komponententyp ARRAY definiert die Reihenfolge, in der die Elemente beim Erstellen der Komponente aufgelistet werden, eine Skala, in der das erste aufgeführte Element den höchsten Wert und das zuletzt aufgeführte Element den niedrigsten Wert besitzt.

Beispiel: Eine Komponente 'mycomp' wird zum Beispiel auf folgende Weise definiert:

```
CREATE SECURITY LABEL COMPONENT mycomp
  ARRAY [ 'Top Secret', 'Secret', 'Employee', 'Public' ]
```

In diesem Fall werden die Elemente so behandelt, als wären sie in einer Struktur wie der folgenden angeordnet:



In einer Komponente des Typs ARRAY können Elemente die folgenden Arten von Beziehungen zueinander haben:

Höher als
Element A ist höher als Element B, wenn Element A in der Klausel ARRAY früher aufgeführt ist als Element B.

Niedriger als
Element A ist niedriger als Element B, wenn Element A in der Klausel ARRAY später aufgeführt ist als Element B.

Typ der LBAC-Sicherheitskennsatzkomponente: TREE

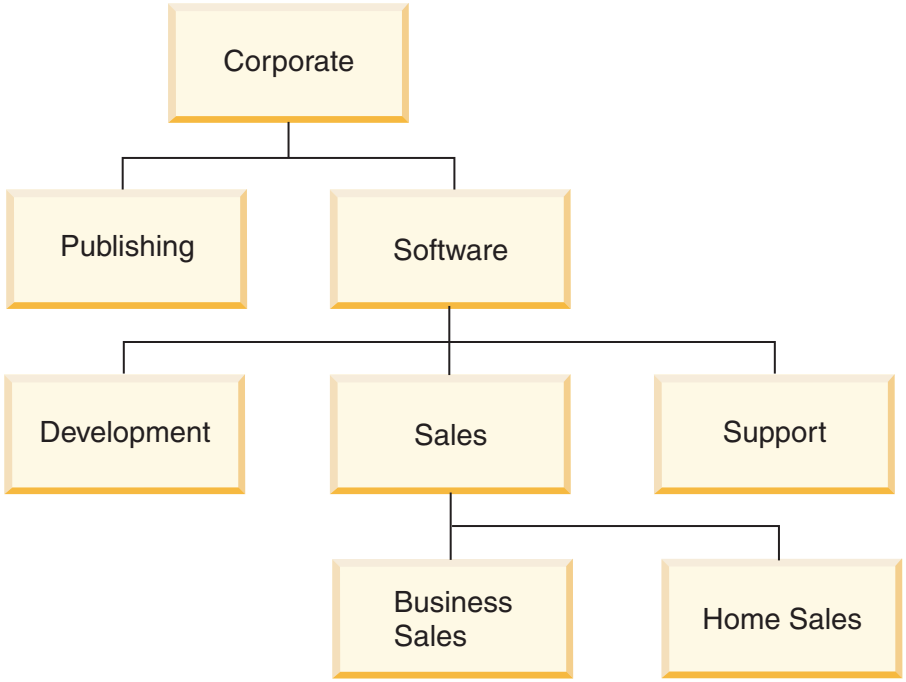
TREE ist ein Typ von Sicherheitskennsatzkomponente, der in einer LBAC-Sicherheitsrichtlinie (LBAC - kennsatzbasierte Zugriffssteuerung) verwendet werden kann.

Bei dem Komponententyp TREE werden die Elemente so behandelt, als wären sie in einer Baumstruktur angeordnet. Wenn Sie ein Element angeben, das Teil einer Komponente des Typs TREE ist, müssen Sie auch angeben, unter welchem anderen Element es sich befindet. Einzige Ausnahme ist das erste Element, das als Stammelement (ROOT) der Baumstruktur angegeben werden muss. Auf diese Weise können die Elemente in einer Baumstruktur verwaltet werden.

Beispiel: Eine Komponente 'mycomp' wird zum Beispiel auf folgende Weise definiert:

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
    'Corporate'      ROOT,
    'Publishing'     UNDER 'Corporate',
    'Software'       UNDER 'Corporate',
    'Development'    UNDER 'Software',
    'Sales'          UNDER 'Software',
    'Support'        UNDER 'Software'
    'Business Sales' UNDER 'Sales'
    'Home Sales'     UNDER 'Sales'
)
```

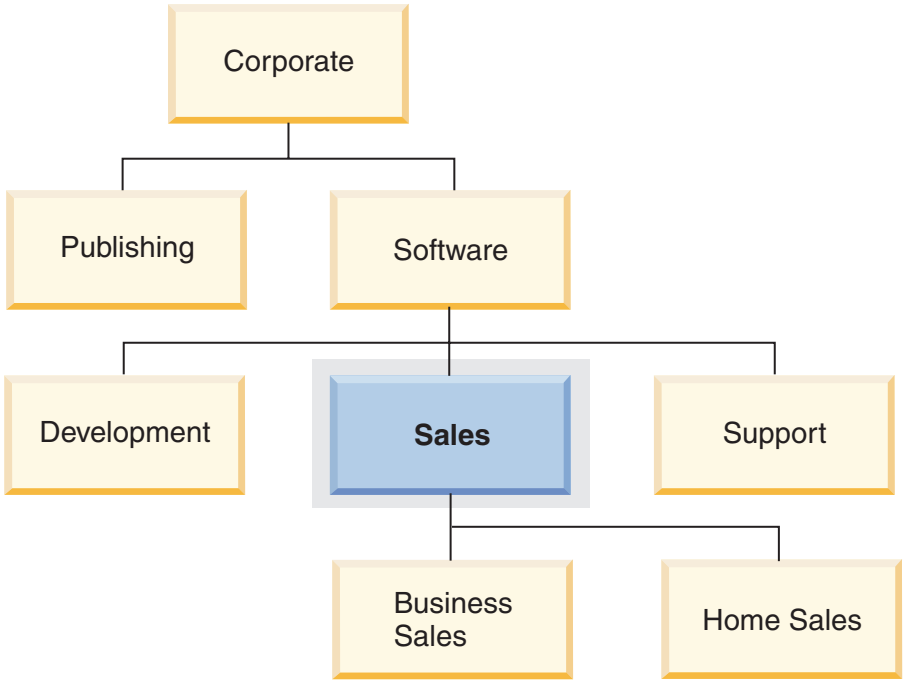
In diesem Fall werden die Elemente so behandelt, als wären sie in einer Baumstruktur wie der folgenden angeordnet:



In einer Komponente des Typs TREE können Elemente die folgenden Typen von Beziehungen zueinander haben:

Direkt übergeordnetes Element (Elter)
Element A ist ein direkt übergeordnetes Element von Element B, wenn sich Element B *direkt unter* Element A befindet.

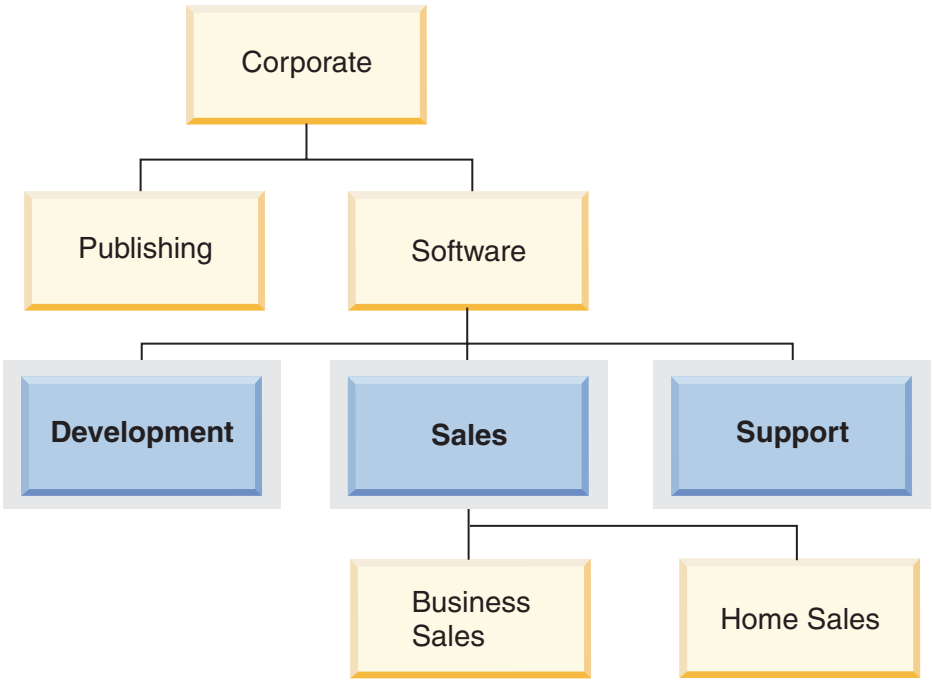
Beispiel: Das folgende Diagramm zeigt das direkt übergeordnete Element des Elements 'Business Sales':



Direkt untergeordnetes Element (Kind)

Element A ist ein direkt untergeordnetes Element von Element B, wenn sich Element A direkt unter Element B befindet.

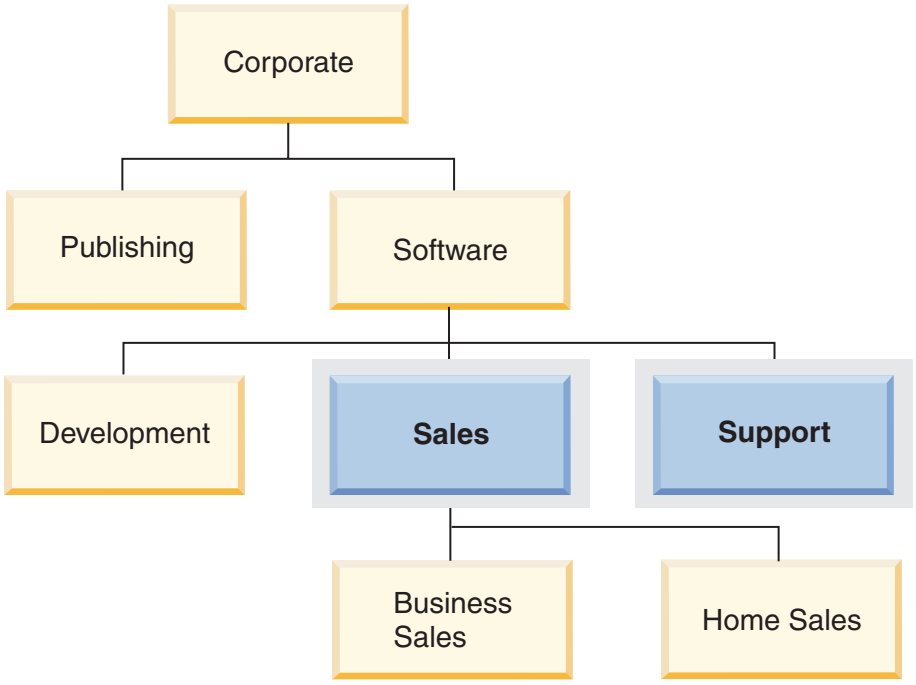
Beispiel: Das folgende Diagramm zeigt die direkt untergeordneten Elemente des Elements 'Software':



Gleichgeordnetes Element

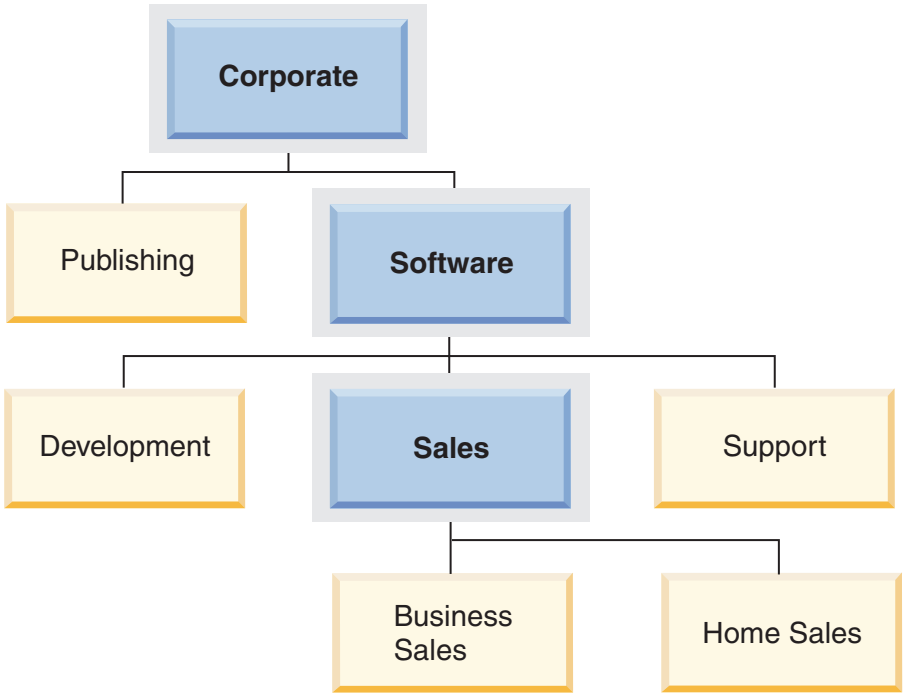
Zwei Elemente sind einander gleichgeordnet, wenn sie dasselbe direkt übergeordnete Element besitzen.

Beispiel: Das folgende Diagramm zeigt die gleichgeordneten Elemente des Elements 'Development':



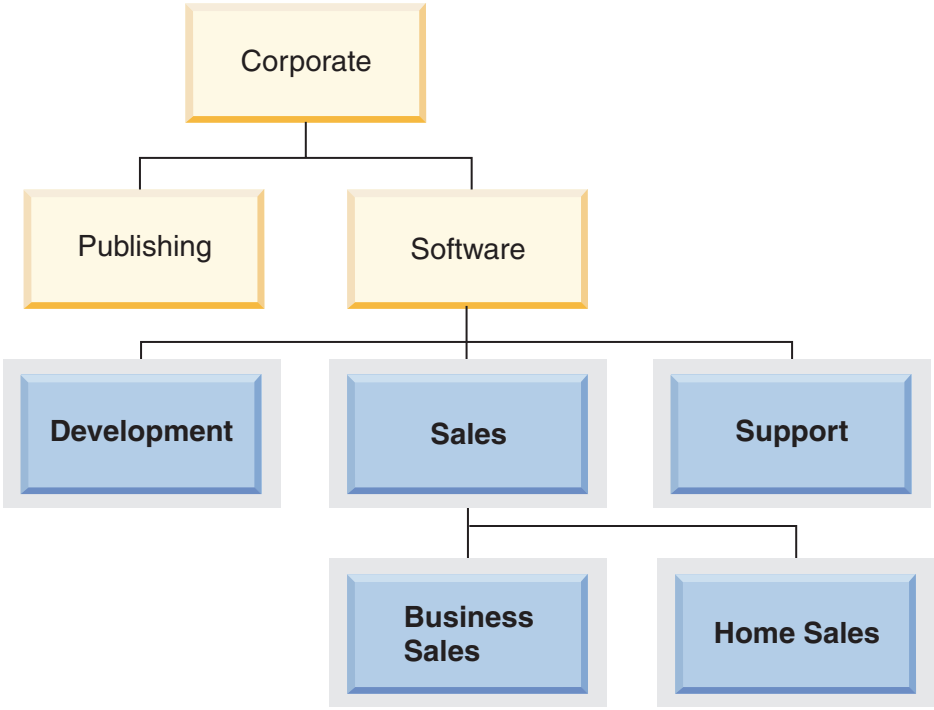
Übergeordnetes Element (Vorfahre)
Element A ist ein übergeordnetes Element (Vorfahre) von Element B, wenn es das direkt übergeordnete Element von B oder ein entfernter übergeordnetes Element von B ist. Das Stammelement ist ein Vorfahre aller anderen Elemente in der Baumstruktur.

Beispiel: Das folgende Diagramm zeigt die Vorfahren des Elements 'Home Sales':



Untergeordnetes Element
Element A ist ein untergeordnetes Element von B, wenn es ein direkt oder entfernter untergeordnetes Element von B ist.

Beispiel: Das folgende Diagramm zeigt die untergeordneten Elemente des Elements 'Software':



LBAC-Sicherheitskennsätze

In LBAC (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) ist ein *Sicherheitskennsatz* ein Datenbankobjekt, das eine bestimmte Gruppe von Sicherheitskriterien beschreibt. Sicherheitskennsätze werden auf Daten angewendet, um die Daten zu schützen. Sie werden Benutzern erteilt, um Ihnen den Zugriff auf geschützte Daten zu ermöglichen.

Wenn ein Benutzer versucht, auf geschützte Daten zuzugreifen, wird sein Sicherheitskennsatz mit dem Sicherheitskennsatz verglichen, der die Daten schützt. Der schützende Sicherheitskennsatz blockiert einige Sicherheitskennsätze und andere nicht. Wenn der Sicherheitskennsatz eines Benutzers blockiert wird, erhält der Benutzer keinen Zugriff auf die Daten.

Jeder Sicherheitskennsatz ist Teil genau einer Sicherheitsrichtlinie und enthält einen Wert für jede Komponente in dieser Sicherheitsrichtlinie. Im Kontext einer Sicherheitskennsatzkomponente ist ein *Wert* eine Liste von null oder mehr Elementen, die durch diese Komponente zugelassen werden. Werte für Komponenten des Typs ARRAY können null oder ein Element enthalten, Werte für andere Typen können null oder mehrere Elemente enthalten. Ein Wert, der keine Elemente enthält, wird als *leerer Wert* bezeichnet.

Beispiel: Wenn eine Komponente des Typs TREE die drei Elemente 'Human Resources' (Personal), 'Sales' (Vertrieb) und 'Shipping' (Versand) enthält, stellen die folgenden Möglichkeiten einige der gültigen Werte für diese Komponente dar:

- Human Resources (oder ein beliebiges der Elemente allein)
- Human Resources, Shipping (oder eine beliebige andere Kombination der Elemente, solange kein Element mehr als einmal vorkommt)
- *Ein leerer Wert*

Ob ein bestimmter Sicherheitskennsatz einen anderen blockiert, wird durch die Werte der einzelnen Komponenten in den Kennsätzen sowie durch den LBAC-Regelsatz bestimmt, der in der Sicherheitsrichtlinie der Tabelle angegeben ist. Eine detaillierte Beschreibung dieses Vergleichs finden Sie in dem Abschnitt, der den Vergleich von LBAC-Sicherheitskennsätzen behandelt.

Wenn Sicherheitskennsätze in eine Textzeichenfolge konvertiert werden, verwenden sie das Format, das in dem Abschnitt beschrieben wird, der das Format für Sicherheitskennsatzwerte behandelt.

Erstellen von Sicherheitskennsätzen

Zur Erstellung eines Sicherheitskennsatzes müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Ein Sicherheitskennsatz wird mithilfe der SQL-Anweisung CREATE SECURITY LABEL erstellt. Bei der Erstellung eines Sicherheitskennsatzes müssen Sie folgende Informationen angeben:

- Einen Namen für den Kennsatz
- Die Sicherheitsrichtlinie, zu der der Kennsatz gehört
- Werte für eine oder mehrere der Komponenten, die in der Sicherheitsrichtlinie enthalten sind

Alle Komponenten, für die kein Wert angegeben wird, werden so interpretiert, als ob sie einen leeren Wert enthielten. Ein Sicherheitskennsatz muss mindestens einen nicht leeren Wert besitzen.

Ändern von Sicherheitskennsätzen

Sicherheitskennsätze können nicht geändert werden. Die einzige Möglichkeit zur Änderung eines Sicherheitskennsatzes besteht darin, diesen zu löschen und erneut zu erstellen. Die *Komponenten* eines Sicherheitskennsatzes können hingegen von einem Sicherheitsadministrator (mit der Anweisung ALTER SECURITY LABEL COMPONENT) geändert werden.

Löschen von Sicherheitskennsätzen

Zum Löschen eines Sicherheitskennsatzes müssen Sie über die Berechtigung eines Sicherheitsadministrators verfügen. Ein Sicherheitskennsatz wird mithilfe der SQL-Anweisung DROP gelöscht. Ein Sicherheitskennsatz, der in der Datenbank zum Schutz von Daten verwendet wird oder zurzeit mindestens einem Benutzer erteilt ist, kann nicht gelöscht werden.

Erteilen von Sicherheitskennsätzen

Sie müssen über die Berechtigung eines Sicherheitsadministrators verfügen, um einem Benutzer, einer Gruppe oder einer Rolle einen Sicherheitskennsatz erteilen zu können. Ein Sicherheitskennsatz wird mithilfe der SQL-Anweisung GRANT SECURITY LABEL erteilt. Beim Erteilten eines Sicherheitskennsatzes können Sie ihn für den Lesezugriff, für den Schreibzugriff oder für den kombinierten Lese- und Schreibzugriff erteilen. Ein Benutzer, eine Gruppe bzw. eine Rolle kann nur einen Sicherheitskennsatz aus derselben Sicherheitsrichtlinie für denselben Typ von Zugriff besitzen.

Entziehen von Sicherheitskennsätzen

Sie müssen über die Berechtigung eines Sicherheitsadministrators verfügen, um einem Benutzer, einer Gruppe oder einer Rolle einen Sicherheitskennsatz entziehen zu können. Ein Sicherheitskennsatz wird mithilfe der SQL-Anweisung REVOKE SECURITY LABEL entzogen.

Mit Sicherheitskennsätzen kompatible Datentypen

Sicherheitskennsätze besitzen den Datentyp SYSPROC.DB2SECURITYLABEL. Zwischen den Datentypen SYSPROC.DB2SECURITYLABEL und VARCHAR(128) FOR BIT DATA wird eine Datenkonvertierung unterstützt.

Ermitteln der Benutzern erteilten Sicherheitskennsätze

Mithilfe der folgenden Abfrage können Sie die Sicherheitskennsätze ermitteln, die Benutzern erteilt wurden:

```
SELECT A.grantee, B.secpolicyname, c.seclabelname
FROM syscat.securitylabelaccess A, syscat.securitypolicies B, syscat.securitylabels C
WHERE A.seclabelid = C.seclabelid and B.secpolicyid = C.secpolicyid
```

Format für Werte von Sicherheitskennsätzen

Manchmal werden die Werte in einem Sicherheitskennsatz in Form einer Zeichenfolge dargestellt, zum Beispiel wenn die integrierte Funktion SECLABEL verwendet wird. Zur Darstellung der Werte eines Sicherheitskennsatzes als Zeichenfolge wird das folgende Format verwendet.

- Die Werte der Komponenten werden von links nach rechts in der gleichen Reihenfolge aufgelistet, in der die Komponenten in der Anweisung CREATE SECURITY POLICY für die Sicherheitsrichtlinie aufgelistet werden.
- Ein Element wird durch den Namen dieses Elements dargestellt.
- Elemente für verschiedene Komponenten werden durch einen Doppelpunkt (:) getrennt.
- Wenn mehr als ein Element für die gleiche Komponente angegeben wird, werden die Elemente in Klammern (()) angegeben und durch Kommata (,) getrennt.
- Leere Werte werden durch ein Paar leerer Klammern (()) dargestellt.

Beispiel: Ein Sicherheitskennsatz ist Teil einer Sicherheitsrichtlinie, welche die folgenden drei Komponenten in dieser Reihenfolge enthält: 'Level', 'Department' und 'Projects'. Der Sicherheitskennsatz hat die folgenden Werte:

Tabelle 8.

Komponente	**Werte**
Level	Secret
Department	*Leerer Wert*
Projects	• Epsilon 37 • Megaphone • Cloverleaf

Die Werte dieses Sicherheitskennsatzes werden als Zeichenfolge wie folgt dargestellt:

```
'Secret:():(Epsilon 37,Megaphone,Cloverleaf)'
```

Vergleichen von LBAC-Sicherheitskennsätzen

Wenn Sie versuchen, auf Daten zuzugreifen, die dem LBAC-Schutz (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) unterliegen, werden Ihre LBAC-Berechtigungsnachweise mit einem oder mehreren Sicherheitskennsätzen verglichen, um zu prüfen, ob der Zugriff blockiert ist. Ihre LBAC-Berechtigungsnachweise bestehen aus allen Sicherheitskennsätzen und allen Freistellungen, die Sie besitzen.

Nur zwei Typen von Vergleichen können ausgeführt werden. Ihre LBAC-Berechtigungsnachweise können mit einem einzelnen Sicherheitskennsatz für Lesezugriff verglichen werden. Oder sie können mit einem einzelnen Sicherheitskennsatz für Schreibzugriff verglichen werden. Aktualisieren und Löschen werden wie ein Lesevorgang gefolgt von einem Schreibvorgang behandelt. Wenn eine Operation mehrere Vergleiche erfordert, wird jeder Vergleich separat ausgeführt.

Welcher Ihrer Sicherheitskennsätze wird verwendet?

Auch wenn Sie vielleicht mehrere Sicherheitskennsätze besitzen, wird nur einer mit dem schützenden Sicherheitskennsatz verglichen. Verwendet wird der Kennsatz, der folgende Bedingungen erfüllt:

- Er ist Teil der Sicherheitsrichtlinie, die die Tabelle schützt, auf die zugegriffen werden soll.
- Er wurde für den beabsichtigten Typ von Zugriff (Lese- oder Schreibzugriff) erteilt.

Wenn Sie keinen Sicherheitskennsatz besitzen, der diese Bedingungen erfüllt, wird ein Standardsicherheitskennsatz angenommen, der leere Werte für alle Komponenten enthält.

Ausführung des Vergleichs

Sicherheitskennsätze werden komponentenweise verglichen. Wenn ein Sicherheitskennsatz keinen Wert für eine der Komponenten enthält, wird ein leerer Wert angenommen. Bei der Prüfung der einzelnen Komponenten wird anhand der relevanten Regeln des LBAC-Regelsatzes entschieden, ob die Elemente in Ihrem Wert für die jeweilige Komponente durch Elemente im Wert für die gleiche Komponente im schützenden Kennsatz blockiert werden sollen. Wenn irgendeiner Ihrer Werte blockiert wird, werden Ihre LBAC-Berechtigungsnachweise durch den schützenden Sicherheitskennsatz blockiert.

Der LBAC-Regelsatz, der für den Vergleich herangezogen wird, wird in der Sicherheitsrichtlinie festgelegt. Informationen zu den Regeln sowie zu ihrer Anwendung finden Sie in der Beschreibung des jeweiligen Regelsatzes.

Wirkung von Freistellungen auf den Vergleich

Wenn Sie eine Freistellung für die Regel besitzen, die zum Vergleichen zweier Werte verwendet wird, wird dieser Vergleich nicht ausgeführt und der schützende Wert wird so interpretiert, dass er den Wert in Ihrem Sicherheitskennsatz nicht blockiert.

Beispiel: Es wird der LBAC-Regelsatz DB2LBACRULES verwendet, und die Sicherheitsrichtlinie hat zwei Komponenten. Eine Komponente hat den Typ ARRAY, die andere den Typ TREE. Dem Benutzer wurde eine Freistellung für die Regel DB2LBACREADTREE erteilt. Diese Regel wird auf den Lesezugriff angewendet, wenn Komponentenwerte des Typs TREE verglichen werden. Wenn der Benutzer versucht, geschützte Daten zu lesen, blockiert jeder beliebige Wert, den der Benutzer für die TREE-Komponente besitzt, selbst wenn es sich um einen leeren Wert handelt, den Zugriff nicht, weil die Regel nicht angewendet wird. Ob der Benutzer die Daten lesen kann, hängt ganz von den Werten der ARRAY-Komponente der Kennsätze ab.

LBAC-Regelsätze - Übersicht

Ein LBAC-Regelsatz ist ein vordefinierter Satz von Regeln, die beim Vergleichen von Sicherheitskennsätzen verwendet werden. Wenn die Werte zweier Sicherheitskennsätze verglichen werden, wird mindestens eine der Regeln des Regelsatzes angewendet, um zu bestimmen, ob ein Wert einen anderen blockiert.

Jeder LBAC-Regelsatz wird durch einen eindeutigen Namen identifiziert. Beim Erstellen einer Sicherheitsrichtlinie müssen Sie den LBAC-Regelsatz angeben, der mit dieser Richtlinie verwendet werden soll. Für alle Vergleiche von Sicherheitskennsätzen, die Teil dieser Richtlinie sind, wird dann der angegebene LBAC-Regelsatz verwendet.

Jede einzelne Regel in einem Regelsatz wird ebenfalls durch einen eindeutigen Namen gekennzeichnet. Der Name einer Regel wird verwendet, wenn eine Freistellung von der betreffenden Regel erteilt wird.

Die Anzahl von Regeln in einem Satz und der Zeitpunkt, zu dem eine Regel verwendet wird, können je nach Regelsatz unterschiedlich sein.

Gegenwärtig wird nur ein LBAC-Regelsatz unterstützt. Der Name dieses Regelsatzes lautet DB2LBACRULES.

LBAC-Regelsatz: DB2LBACRULES

Der LBAC-Regelsatz DB2LBACRULES stellt einen traditionellen Satz von Regeln zum Vergleich der Werte von Sicherheitskennsatzkomponenten bereit. Er schützt gegen Write-up- und Write-down-Vorgänge.

Was bedeuten Write-up und Write-down?

Write-up und Write-down bezieht sich nur auf Komponenten des Typs ARRAY und nur auf den Schreibzugriff. Ein Write-up tritt auf, wenn der Wert, der Daten schützt, in die Sie schreiben wollen, höher ist als Ihr Wert. Wenn der Wert, der die Daten schützt, niedriger als Ihr Wert ist, wird dies als Write-down bezeichnet. Standardmäßig werden weder Write-up- noch Write-down-Vorgänge zugelassen. Das bedeutet, dass Sie nur in Daten schreiben können, die durch den gleichen Wert geschützt werden, den auch Sie besitzen.

Wenn zwei Werte für dieselbe Komponente verglichen werden, hängen die Regeln, die dazu verwendet werden, vom Typ der Komponente (ARRAY, SET oder TREE) sowie vom Typ des versuchten Zugriffs (Schreib- oder Lesezugriff) ab. In der folgenden Tabelle werden die Regeln aufgelistet und beschrieben, wann sie jeweils verwendet werden und wie sie bestimmen, ob der Zugriff blockiert wird.

Tabelle 9. Zusammenfassung der DB2LBACRULES-Regeln

Regelname	Typ von Komponente, für deren Wertevergleich die Regel gilt	Typ von Zugriffsversuch, für den die Regel gilt	Bedingung, unter der der Zugriff blockiert wird
DB2LBACREADARRAY	ARRAY	Lesen	Der Wert des Benutzers ist niedriger als der schützende Wert.
DB2LBACREADSET	SET	Lesen	Es gibt mindestens einen schützenden Wert, den der Benutzer nicht besitzt.
DB2LBACREADTREE	TREE	Lesen	Keiner der Werte des Benutzers ist gleich einem der schützenden Werte oder einem der übergeordneten Werte (Vorfahren) der schützenden Werte.
DB2LBACWRITEARRAY	ARRAY	Schreiben	Der Wert des Benutzers ist höher oder niedriger als der schützende Wert.[1]
DB2LBACWRITESET	SET	Schreiben	Es gibt mindestens einen schützenden Wert, den der Benutzer nicht besitzt.
DB2LBACWRITETREE	TREE	Schreiben	Keiner der Werte des Benutzers ist gleich einem der schützenden Werte oder einem der übergeordneten Werte (Vorfahren) der schützenden Werte.

Anmerkung:

1. Die Regel DB2LBACWRITEARRAY kann als Kombination aus zwei verschiedenen Regeln betrachtet werden. Die eine verhindert das Schreiben in Daten, die sich auf einer höheren als Ihre Stufe befinden (Write-up), während die andere das Schreiben in Daten verhindert, die sich auf einer niedrigeren als Ihre Stufe befinden (Write-down). Wenn Sie eine Freistellung von dieser Regel erteilen, können Sie den Benutzer von der einen oder der anderen oder auch von beiden Regeln freistellen.

Behandlung leerer Werte durch die Regeln

Alle Regeln behandeln leere Werte in gleicher Weise. Ein leerer Wert blockiert keine anderen Werte und wird selbst von jedem nicht leeren Wert blockiert.

Beispiele für DB2LBACREADSET und DB2LBACWRITESET

Diese Beispiele gelten für einen Benutzer, der versucht, geschützte Daten zu lesen oder in geschützte Daten zu schreiben. Für sie wird angenommen, dass die Werte für eine Komponente des Typs SET definiert sind, die die folgenden Elemente besitzt: eins, zwei, drei, vier.

Tabelle 10. Beispiele für die Anwendung der Regeln DB2LBACREADSET und DB2LBACWRITESET

Wert des Benutzers	Schützender Wert	Zugriff blockiert?
'eins'	'eins'	Nicht blockiert. Die Werte sind identisch.
'(eins,zwei,drei)'	'eins'	Nicht blockiert. Der Wert des Benutzers enthält das Element 'eins'.
'(eins,zwei)'	'(eins,zwei,vier)'	Blockiert. Das Element 'vier' ist im schützenden Wert, jedoch nicht im Wert des Benutzers enthalten.
'()'	'eins'	Blockiert. Ein leerer Wert wird von jedem beliebigen nicht leeren Wert blockiert.
'eins'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.
'()'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.

DB2LBACREADTREE und DB2LBACWRITETREE

Die folgenden Beispiele gelten sowohl für den Lese- als auch für den Schreibzugriff. Für sie wird angenommen, dass die Werte für eine Komponente des Typs TREE gelten, die auf folgende Weise definiert ist:

```
CREATE SECURITY LABEL COMPONENT mycomp
TREE (
    'Corporate'      ROOT,
    'Publishing'     UNDER 'Corporate',
    'Software'       UNDER 'Corporate',
    'Development'    UNDER 'Software',
    'Sales'          UNDER 'Software',
    'Support'        UNDER 'Software'
    'Business Sales' UNDER 'Sales'
    'Home Sales'     UNDER 'Sales'
)
```

Dies bedeutet, dass die Elemente wie folgt angeordnet sind:

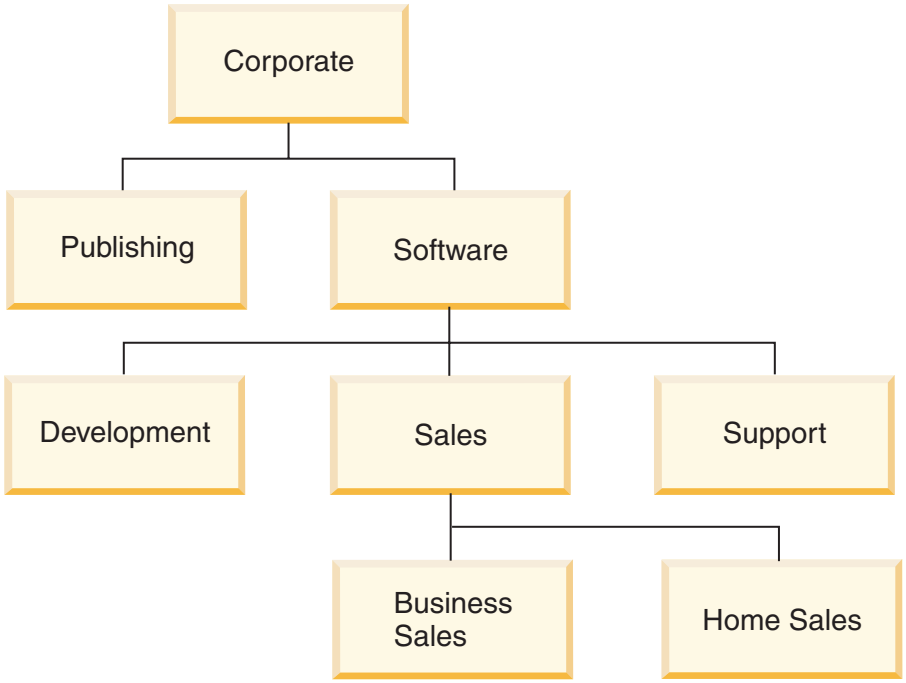


Tabelle 11. Beispiel für die Anwendung der Regeln DB2LBACREADTREE und DB2LBACWRITETREE

Wert des Benutzers	Schützender Wert	Zugriff blockiert?
'(Support,Sales)'	'Development'	Blockiert. Das Element 'Development' ist nicht in den Werten des Benutzers enthalten und weder 'Support' noch 'Sales' ist ein übergeordnetes Element (Vorfahre) von 'Development'.
'(Development,Software)'	'(Business Sales,Publishing)'	Nicht blockiert. Das Element 'Software' ist ein übergeordnetes Element (Vorfahre) des Elements 'Business Sales'.
'(Publishing,Sales)'	'(Publishing,Support)'	Nicht blockiert. Das Element 'Publishing' ist in beiden Gruppen von Werten enthalten.
'Corporate'	'Development'	Nicht blockiert. Das Stammelement ist ein übergeordnetes Element (Vorfahre) aller anderen Werte.
'()'	'Sales'	Blockiert. Ein leerer Wert wird von jedem beliebigen nicht leeren Wert blockiert.
'Home Sales'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.
'()'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.

Beispiele für DB2LBACREADARRAY

Diese Beispiele gelten nur für den Lesezugriff. Für sie wird angenommen, dass die Werte für eine Komponente des Typs ARRAY definiert sind, die die folgenden Elemente in der angegebenen Anordnung enthält:

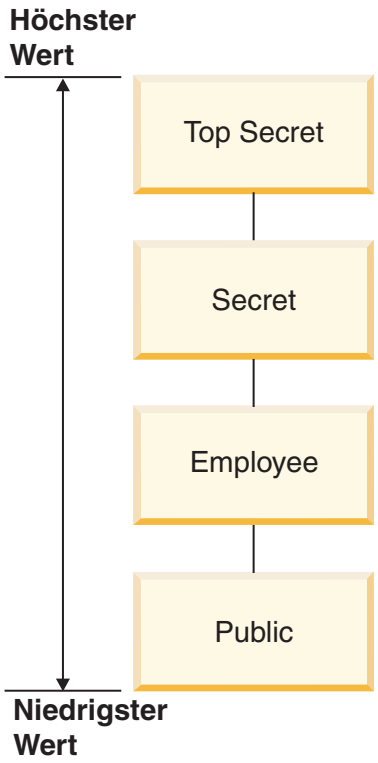


Tabelle 12. Beispiele für die Anwendung der Regel DB2LBACREADARRAY

Wert des Benutzers	Schützender Wert	Lesezugriff blockiert?
'Secret'	'Employee'	Nicht blockiert. Das Element 'Secret' ist höher als das Element 'Employee'.
'Secret'	'Secret'	Nicht blockiert. Die Werte sind identisch.
'Secret'	'Top Secret'	Blockiert. Das Element 'Top Secret' ist höher als das Element 'Secret'.
'()'	'Public'	Blockiert. Ein leerer Wert wird von jedem beliebigen nicht leeren Wert blockiert.
'Public'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.
'()'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.

Beispiele für DB2LBACWRITEARRAY

Diese Beispiele gelten nur für den Schreibzugriff. Für sie wird angenommen, dass die Werte für eine Komponente des Typs ARRAY definiert sind, die die folgenden Elemente in der angegebenen Anordnung enthält:

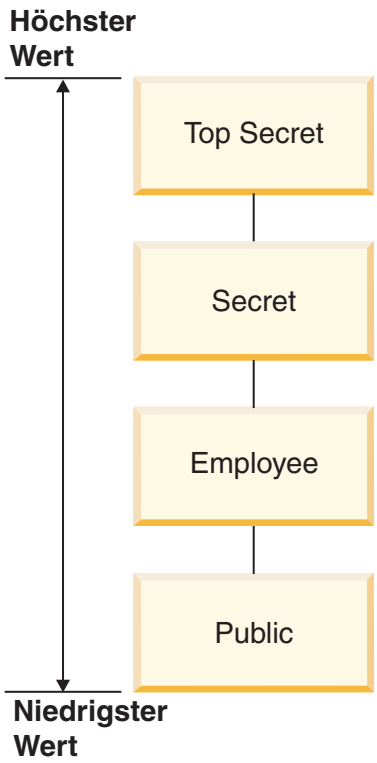


Tabelle 13. Beispiele für die Anwendung der Regel DB2LBACWRITEARRAY

Wert des Benutzers	Schützender Wert	Schreibzugriff blockiert?
'Secret'	'Employee'	Blockiert. Das Element 'Employee' ist niedriger als das Element 'Secret'.
'Secret'	'Secret'	Nicht blockiert. Die Werte sind identisch.
'Secret'	'Top Secret'	Blockiert. Das Element 'Top Secret' ist höher als das Element 'Secret'.
'()'	'Public'	Blockiert. Ein leerer Wert wird von jedem beliebigen nicht leeren Wert blockiert.
'Public'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.
'()'	'()'	Nicht blockiert. Kein Wert wird durch einen leeren Wert blockiert.

Freistellungen von LBAC-Regeln

Wenn Sie eine LBAC-Regelfreistellung für eine bestimmte Regel einer bestimmten Sicherheitsrichtlinie haben, wird diese Regel nicht angewendet, wenn Sie versuchen, auf die Daten zuzugreifen, die durch diese Sicherheitsrichtlinie geschützt werden.

Eine Freistellung hat keine Wirkung, wenn Sicherheitskennsätze einer anderen Sicherheitsrichtlinie als die, für die sie erteilt wurde, verglichen werden.

Beispiel:

Es sind zwei Tabellen vorhanden: T1 und T2. T1 wird durch die Sicherheitsrichtlinie P1, T2 durch die Sicherheitsrichtlinie P2 geschützt. Beide Sicherheitsrichtlinien enthalten eine Komponente. Die Komponenten sind jeweils vom Typ ARRAY. T1 und T2 enthalten jeweils nur eine Zeile von Daten. Der Sicherheitskennsatz, den Sie für den Schreibzugriff unter der Sicherheitsrichtlinie P1 besitzen, erlaubt Ihnen keinen Zugriff auf die Zeile in T1. Der Sicherheitskennsatz, den Sie für den Lesezugriff unter der Sicherheitsrichtlinie P2 besitzen, erlaubt Ihnen keinen Zugriff auf die Zeile in T2. In dieser Situation wird Ihnen eine Freistellung für DB2LBACREADARRAY unter P1 erteilt. Jetzt können Sie die Zeile in T1 lesen, die Zeile in T2 jedoch nicht, weil T2 von einer anderen Sicherheitsrichtlinie geschützt wird und Sie keine Freistellung für die Regel DB2LBACREADARRAY in dieser Richtlinie haben.

Sie können mehrere Freistellungen besitzen. Wenn Sie eine Freistellung für jede Regel besitzen, die von einer Sicherheitsrichtlinie verwendet wird, haben Sie vollständigen Zugriff auf alle Daten, die durch diese Sicherheitsrichtlinie geschützt werden.

Erteilen von Freistellungen für LBAC-Regeln

Zum Erteilen einer Freistellung für eine LBAC-Regel müssen Sie ein Sicherheitsadministrator sein. Eine Freistellung für eine LBAC-Regel wird mithilfe der SQL-Anweisung GRANT EXEMPTION ON RULE erteilt.

Beim Erteilen einer Freistellung für eine LBAC-Regel geben Sie die folgenden Informationen an:
- Die Regel bzw. die Regeln, für die die Freistellung gilt
- Die Sicherheitsrichtlinie, für die die Freistellung gilt
- Den Benutzer, die Gruppe oder die Rolle, für die Sie die Freistellung erteilen

Wichtig: Freistellungen von LBAC-Regeln gewähren einen sehr mächtigen Zugriff. Erteilen Sie Freistellungen nur nach reiflicher Überlegung.

Entziehen von Freistellungen für LBAC-Regeln

Zum Entziehen einer Freistellung für eine LBAC-Regel müssen Sie ein Sicherheitsadministrator sein. Eine Freistellung für eine LBAC-Regel wird mithilfe der SQL-Anweisung REVOKE EXEMPTION ON RULE entzogen.

Ermitteln der Benutzern erteilten Regelfreistellungen

Mithilfe der folgenden Abfrage können Sie die Regelfreistellungen ermitteln, die Benutzern erteilt wurden:

```
SELECT A.grantee, A.accessrulename, B.secpolicyname
FROM syscat.securitypolicyexemptions A, syscat.securitypolicies B
WHERE A.secpolicyid = B.secpolicyid
```

Integrierte Funktionen zum Verwalten von LBAC-Sicherheitskennsätzen

Die integrierten Funktionen SECLABEL, SECLABEL_BY_NAME und SECLABEL_TO_CHAR dienen zur Verwaltung von Sicherheitskennsätzen der kennsatzbasierten Zugriffssteuerung (LBAC, Label-Based Access Control).

Jede von ihnen wird im Folgenden kurz beschrieben. Eine detaillierte Beschreibung finden Sie in *SQL Reference*.

SECLABEL

Diese integrierte Funktion dient zum Aufbau eines Sicherheitskennsatzes. Dazu werden eine Sicherheitsrichtlinie und Werte für die einzelnen Komponenten in diesem Kennsatz angegeben. Der zurückgegebene Wert besitzt den Datentyp DB2SECURITYLABEL und ist ein Sicherheitskennsatz, der zur angegebenen Sicherheitsrichtlinie gehört und die angegebenen Werte für die Komponenten besitzt. Es ist nicht erforderlich, dass ein Sicherheitskennsatz mit den angegebenen Werten bereits vorhanden ist.

Beispiel: Tabelle T1 hat zwei Spalten. Die erste hat den Datentyp DB2SECURITYLABEL und die zweite den Datentyp INTEGER. T1 wird durch die Sicherheitsrichtlinie P1 geschützt, die drei Sicherheitskennsatzkomponenten enthält: 'level', 'departments' und 'groups'. Wenn 'UNCLASSIFIED' ein Element der Komponente 'level', 'ALPHA' und 'SIGMA' beides Elemente der Komponente 'departments' und 'G2' ein Element der Komponente 'groups' sind, kann zum Beispiel ein Sicherheitskennsatz wie der folgende eingefügt werden:

```
INSERT INTO T1 VALUES
   ( SECLABEL( 'P1', 'UNCLASSIFIED:(ALPHA,SIGMA):G2' ), 22 )
```

SECLABEL_BY_NAME

Diese integrierte Funktion akzeptiert den Namen einer Sicherheitsrichtlinie und den Namen eines Sicherheitskennsatzes, der Teil dieser Sicherheitsrichtlinie ist. Dann gibt Sie den angegebenen Sicherheitskennsatz als Wert des Typs DB2SECURITYLABEL zurück. Diese Funktion muss verwendet werden, wenn ein vorhandener Sicherheitskennsatz in eine Spalte eingefügt wird, die den Datentyp DB2SECURITYLABEL besitzt.

Beispiel: Tabelle T1 hat zwei Spalten. Die erste hat den Datentyp DB2SECURITYLABEL und die zweite den Datentyp INTEGER. Der Sicherheitskennsatz mit dem Namen L1 ist Teil der Sicherheitsrichtlinie P1. Die folgende SQL-Anweisung kann in diesem Fall zum Einfügen des Sicherheitskennsatzes verwendet werden:

```
INSERT INTO T1 VALUES ( SECLABEL_BY_NAME( 'P1', 'L1' ), 22 )
```

Die folgende SQL-Anweisung funktioniert hingegen nicht:

```
INSERT INTO T1 VALUES ( P1.L1, 22 )     // Syntaxfehler!
```

SECLABEL_TO_CHAR

Diese integrierte Funktion liefert eine Zeichenfolgedarstellung der Werte, die einen Sicherheitskennsatz bilden.

Beispiel: Spalte C1 in Tabelle T1 hat den Datentyp DB2SECURITYLABEL. T1 wird durch die Sicherheitsrichtlinie P1 geschützt, die drei Sicherheitskennsatzkomponenten enthält: 'level', 'departments' und 'groups'. Die Tabelle T1 enthält eine Zeile, wobei der Wert in Spalte C1 die folgenden Elemente für die einzelnen Komponenten enthält:

Komponente	Elemente
'level'	SECRET

Komponente	Elemente
'departments'	DELTA und SIGMA
'groups'	G3

Ein Benutzer, der über LBAC-Berechtigungsnachweise verfügt, die einen Lesezugriff auf die Zeilen zulassen, kann zum Beispiel die folgende SQL-Anweisung ausführen:

```
SELECT SECLABEL_TO_CHAR( 'P1', C1 ) AS C1 FROM T1
```

Die Ausgabe sähe wie folgt aus:

```
C1

'SECRET:(DELTA,SIGMA):G3'
```

Schützen von Daten mit LBAC

Die LBAC-Funktionalität (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) kann zum Schutz von Datenzeilen, Datenspalten oder beidem verwendet werden. Daten in einer Tabelle können nur durch Sicherheitskennsätze geschützt werden, die Teil der Sicherheitsrichtlinie sind, die die Tabelle schützt. Der Datenschutz, einschließlich des Hinzufügens einer Sicherheitsrichtlinie, kann beim Erstellen oder später beim Ändern einer Tabelle eingefügt werden.

Sie können innerhalb einer Anweisung CREATE TABLE oder ALTER TABLE einer Tabelle eine Sicherheitsrichtlinie hinzufügen und Daten in dieser Tabelle schützen.

Als allgemeine Regel gilt, dass es nicht zulässig ist, Daten so zu schützen, dass die eigenen aktuellen LBAC-Berechtigungsnachweise keinen Schreibzugriff auf diese Daten erlauben.

Hinzufügen einer Sicherheitsrichtlinie zu einer Tabelle

Sie können einer zu erstellenden Tabelle eine Sicherheitsrichtlinie hinzufügen, indem Sie die Klausel SECURITY POLICY in der Anweisung CREATE TABLE verwenden. Sie können einer vorhandenen Tabelle eine Sicherheitsrichtlinie hinzufügen, indem Sie die Klausel ADD SECURITY POLICY der Anweisung ALTER TABLE verwenden. Zum Hinzufügen einer Sicherheitsrichtlinie zu einer Tabelle sind weder eine Berechtigung SECADM noch LBAC-Berechtigungsnachweise erforderlich.

Sicherheitsrichtlinien können keinen Typen von Tabellen hinzugefügt werden, die nicht durch LBAC geschützt werden können. In der Übersicht zu LBAC finden Sie eine Liste von Tabellentypen, die nicht durch LBAC geschützt werden können.

Jeder Tabelle kann nur eine Sicherheitsrichtlinie hinzugefügt werden.

Schützen von Zeilen

Sie können geschützte Zeilen in einer neuen Tabelle zulassen, indem Sie eine Spalte mit dem Datentyp DB2SECURITYLABEL beim Erstellen der Tabelle mit einfügen. Die Anweisung CREATE TABLE muss der Tabelle außerdem eine Sicherheitsrichtlinie hinzufügen. Zur Erstellung einer solchen Tabelle sind weder eine Berechtigung SECADM noch LBAC-Berechtigungsnachweise erforderlich.

Sie können geschützte Zeilen in einer vorhandenen Tabelle zulassen, indem Sie eine Spalte mit dem Datentyp DB2SECURITYLABEL hinzufügen. Um eine solche Spalte hinzufügen zu können, muss die Tabelle entweder bereits durch eine Sicherheitsrichtlinie geschützt sein oder die Anweisung ALTER TABLE, die die Spalte hinzufügt, muss der Tabelle auch eine Sicherheitsrichtlinie hinzufügen. Wenn die Spalte hinzugefügt ist, wird der Sicherheitskennsatz, den Sie für den Schreibzugriff besitzen, zum Schutz aller vorhandenen Zeilen verwendet. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff besitzen, der Teil der Sicherheitsrichtlinie ist, die die Tabelle schützt, können Sie keine Spalte mit dem Datentyp DB2SECURITYLABEL hinzufügen.

Wenn eine Spalte des Typs DB2SECURITYLABEL in eine Tabelle eingefügt ist, schützen Sie jede neue Zeile von Daten, indem Sie einen Sicherheitskennsatz in dieser Spalte speichern. Detaillierte Informationen dazu, wie dies funktioniert, enthalten die Themenabschnitte zum Einfügen und Aktualisieren von LBAC-geschützten Daten. Sie müssen über LBAC-Berechtigungsnachweise verfügen, um Zeilen in eine Tabelle einzufügen, die eine Spalte des Typs DB2SECURITYLABEL besitzt.

Eine Spalte, die den Datentyp DB2SECURITYLABEL besitzt, kann weder gelöscht noch in einen anderen Datentyp geändert werden.

Schützen von Spalten

Sie können bei der Erstellung der Tabelle eine Spalte schützen, indem Sie die Spaltenoption SECURED WITH der Anweisung CREATE TABLE verwenden. Sie können einer vorhandenen Spalte einen Schutz hinzufügen, indem Sie die Option SECURED WITH in einer Anweisung ALTER TABLE verwenden.

Zum Schützen einer Spalte durch einen bestimmten Sicherheitskennsatz müssen Sie über LBAC-Berechtigungsnachweise verfügen, die Ihnen einen Schreibzugriff auf die durch diesen Sicherheitskennsatz geschützten Daten erlauben. Die Berechtigung SECADM ist nicht erforderlich.

Spalten können nur durch Sicherheitskennsätze geschützt werden, die Teil der Sicherheitsrichtlinie sind, die die Tabelle schützt. Spalten in einer Tabelle, die keine Sicherheitsrichtlinie besitzt, können nicht geschützt werden. Es ist zulässig, eine Tabelle durch eine Sicherheitsrichtlinie und gleichzeitig eine oder mehrere Spalten in derselben Anweisung zu schützen.

Sie können eine beliebige Anzahl von Spalten in einer Tabelle schützen, jedoch kann eine Spalte nur durch einen einzigen Sicherheitskennsatz geschützt werden.

Lesen von LBAC-geschützten Daten

Wenn Sie versuchen, Daten zu lesen, die einem LBAC-Schutz (Label-Based Access Control, kennsatzbasierte Zugriffssteuerung) unterliegen, werden Ihre LBAC-Berechtigungsnachweise für den Lesezugriff mit dem Sicherheitskennsatz verglichen, der die Daten schützt. Wenn der schützende Kennsatz Ihre Berechtigungsnachweise nicht blockiert, wird Ihr Lesezugriff auf die Daten zugelassen.

Bei einer geschützten Spalte ist der schützende Sicherheitskennsatz im Schema der Tabelle definiert. Der schützende Sicherheitskennsatz für diese Spalte ist für jede Zeile in der Tabelle der gleiche. Bei einer geschützten Zeile wird der schützende Sicherheitskennsatz der Zeile in einer Spalte des Typs DB2SECURITYLABEL gespeichert. Der Wert dieser Spalte kann für jede Zeile in der Tabelle unterschiedlich sein.

Detaillierte Informationen dazu, wie Ihre LBAC-Berechtigungsnachweise mit einem Sicherheitskennsatz verglichen werden, finden Sie in dem Abschnitt über den Vergleich von LBAC-Sicherheitskennsätzen.

Lesen geschützter Spalten

Wenn Sie versuchen, Daten aus einer geschützten Spalte zu lesen, werden Ihre LBAC-Berechtigungsnachweise mit dem Sicherheitskennsatz verglichen, der die Spalte schützt. Abhängig vom Ergebnis dieses Vergleichs wird der Zugriff entweder blockiert oder zugelassen. Wenn der Zugriff blockiert wird, wird ein Fehler zurückgegeben und die Ausführung der Anweisung schlägt fehl. Anderenfalls wird die Ausführung der Anweisung normal fortgesetzt.

Ein Versuch, eine Spalte zu lesen, auf die Ihre LBAC-Berechtigungsnachweise keinen Lesezugriff erlauben, bewirkt, dass die gesamte Anweisung fehlschlägt.

Beispiel:

Die Tabelle T1 enthält zwei geschützte Spalten. Die Spalte C1 wird durch den Sicherheitskennsatz L1 geschützt. Die Spalte C2 wird durch den Sicherheitskennsatz L2 geschützt.

Nehmen Sie an, die Benutzerin Jyoti verfügt über LBAC-Berechtigungsnachweise für einen Lesezugriff, der durch den Sicherheitskennsatz L1 zugelassen wird, durch den Sicherheitskennsatz L2 jedoch nicht. Wenn Jyoti die folgende SQL-Anweisung absetzt, schlägt die Ausführung der Anweisung fehl:

```
SELECT * FROM T1
```

Die Ausführung der Anweisung schlägt fehl, weil Spalte C2 durch das Platzhalterzeichen (*) in die SELECT-Klausel mit eingeschlossen ist.

Wenn Jyoti die folgende SQL-Anweisung absetzt, ist die Ausführung erfolgreich:

```
SELECT C1 FROM T1
```

Die einzige geschützte Spalte in der SELECT-Klausel ist C1, und die LBAC-Berechtigungsnachweise von Jyoti lassen den Lesezugriff auf diese Spalte zu.

Lesen geschützter Zeilen

Wenn Sie keine LBAC-Berechtigungsnachweise besitzen, die Ihnen einen Lesezugriff auf eine Zeile erlauben, wird Ihr Lesezugriff so verarbeitet, als ob diese Zeile für Sie nicht vorhanden wäre.

Wenn Sie geschützte Zeilen lesen, werden nur solche Zeilen zurückgegeben, auf die Ihre LBAC-Berechtigungsnachweise einen Lesezugriff zulassen. Dies gilt auch dann, wenn die Spalte des Typs DB2SECURITYLABEL nicht in die SELECT-Klausel eingeschlossen ist.

Abhängig von den LBAC-Berechtigungsnachweisen werden verschiedenen Benutzern möglicherweise verschiedene Zeilen aus einer Tabelle angezeigt, die geschützte Zeilen enthält. Zum Beispiel empfangen zwei Benutzer, die die Anweisung `SELECT COUNT(*) FROM T1` ausführen, vielleicht unterschiedliche Ergebnisse, wenn die Tabelle T1 geschützte Spalten enthält und die Benutzer über unterschiedliche LBAC-Berechtigungsnachweise verfügen.

Ihre LBAC-Berechtigungsnachweise gelten nicht nur für SELECT-Anweisungen, sondern auch für andere SQL-Anweisungen wie UPDATE und DELETE. Wenn Sie keine LBAC-Berechtigungsnachweise besitzen, die Ihnen einen Lesezugriff auf eine Zeile erlauben, können Sie diese Zeile nicht bearbeiten.

Beispiel:

Die Tabelle 1 enthält die folgenden Zeilen und Spalten. Die Spalte ROWSECURITYLABEL hat den Datentyp DB2SECURITYLABEL.

Tabelle 14.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3
Bird	55	L2

Nehmen Sie an, dass der Benutzer Dan über LBAC-Berechtigungsnachweise verfügt, die ihm einen Lesezugriff auf Daten erlauben, die durch den Sicherheitskennsatz L1 geschützt werden, jedoch keinen Lesezugriff auf Daten, die durch die Sicherheitskennsätze L2 oder L3 geschützt werden.

Dan setzt die folgende SQL-Anweisung ab:

```
SELECT * FROM T1
```

Die SELECT-Anweisung liefert nur die Zeile für Miller. Es werden keine Nachrichten oder Warnungen zurückgegeben.

Dan erhält die folgende Sicht auf die Tabelle T1:

Tabelle 15.

LASTNAME	DEPTNO	ROWSECURITYLABEL
Miller	77	L1

Die Zeilen für Rjaibi, Fielding und Bird werden nicht zurückgegeben, weil der Lesezugriff durch die zugehörigen Sicherheitskennsätze blockiert wird. Dan kann diese Zeile nicht löschen oder aktualisieren. Diese Zeile werden darüber hinaus auch aus allen Spaltenfunktionen ausgeklammert. Für Dan verhält es sich so, als ob diese Zeilen nicht vorhanden wären.

Dan setzt die folgende SQL-Anweisung ab:

```
SELECT COUNT(*) FROM T1
```

Die Anweisung liefert den Wert 1, da nur die Zeile für Miller durch den Benutzer Dan gelesen werden kann.

Lesen geschützter Zeilen, die geschützte Spalten enthalten

Der Spaltenzugriff wird vor dem Zeilenzugriff überprüft. Wenn Ihre LBAC-Berechtigungsnachweise für den Lesezugriff durch den Sicherheitskennsatz blockiert werden, der eine der Spalten schützt, die Sie auswählen, schlägt die Ausführung der gesamten Anweisung fehl. Ist dies nicht der Fall, wird die Ausführung

der Anweisung fortgesetzt, wobei nur die Zeilen zurückgegeben werden, die durch Sicherheitskennsätze geschützt werden, die Ihren LBAC-Berechtigungsnachweisen einen Lesezugriff auf die Zeilen gestatten.

Beispiel

Die Spalte LASTNAME der Tabelle T1 wird durch den Sicherheitskennsatz L1 geschützt. Die Spalte DEPTNO wird durch den Sicherheitskennsatz L2 geschützt. Die Spalte ROWSECURITYLABEL hat den Datentyp DB2SECURITYLABEL. Die Tabelle T1 sieht einschließlich Daten wie folgt aus:

Tabelle 16.

LASTNAME *geschützt durch L1*	**DEPTNO** *geschützt durch L2*	**ROWSECURITYLABEL**
Rjaibi	55	L2
Miller	77	L1
Fielding	11	L3

Nehmen Sie an, dass die Benutzerin Sakari über LBAC-Berechtigungsnachweise verfügt, die ihr einen Lesezugriff auf Daten erlauben, die durch den Sicherheitskennsatz L1 geschützt werden, jedoch keinen Lesezugriff auf Daten, die durch die Sicherheitskennsätze L2 oder L3 geschützt werden.

Sakari setzt die folgende SQL-Anweisung ab:

```
SELECT * FROM T1
```

Die Ausführung der Anweisung schlägt fehl, weil in der SELECT-Klausel das Platzhalterzeichen (*) verwendet wird, das auch die Spalte DEPTNO mit einschließt. Die Spalte DEPTNO wird durch den Sicherheitskennsatz L2 geschützt, auf den die LBAC-Berechtigungsnachweise von Sakari keinen Lesezugriff zulassen.

Sakari setzt als Nächstes die folgende SQL-Anweisung ab:

```
SELECT LASTNAME, ROWSECURITYLABEL FROM T1
```

Die SELECT-Klausel schließt keine Spalten mit ein, die von Sakari nicht gelesen werden können, sodass die Ausführung der Anweisung fortgesetzt wird. Es wird jedoch nur eine Zeile zurückgegeben, da jede der anderen Zeilen durch den Sicherheitskennsatz L2 bzw. L3 geschützt wird.

Tabelle 17.

LASTNAME	**ROWSECURITYLABEL**
Miller	L1

Einfügen von LBAC-geschützten Daten

Einfügen in geschützte Spalten

Wenn Sie versuchen, Daten explizit in eine geschützte Spalte einzufügen, werden Ihre LBAC-Berechtigungsnachweise für den Schreibzugriff mit dem Sicherheitskennsatz verglichen, der diese Spalte schützt. Abhängig vom Ergebnis dieses Vergleichs wird der Zugriff entweder blockiert oder zugelassen.

Detaillierte Informationen dazu, wie zwei Sicherheitskennsätze verglichen werden, finden Sie in dem Abschnitt über den Vergleich von LBAC-Sicherheitskennsätzen.

Wenn der Zugriff zugelassen wird, wird die Ausführung der Anweisung normal fortgesetzt. Wenn der Zugriff blockiert wird, schlägt das Einfügen fehl und ein Fehler wird zurückgegeben.

Wenn Sie eine Zeile einfügen, jedoch keinen Wert für eine geschützte Spalte angeben, wird der Standardwert, sofern verfügbar, eingefügt. Dies geschieht auch dann, wenn Ihre LBAC-Berechtigungsnachweise keinen Schreibzugriff auf diese Spalte zulassen. Ein Standardwert ist in folgenden Fällen verfügbar:

- Die Spalte wurde mit der Option WITH DEFAULT deklariert.
- Die Spalte ist eine generierte Spalte.
- Die Spalte besitzt einen Standardwert, der durch einen Vortrigger eingefügt wird.
- Die Spalte hat den Datentyp DB2SECURITYLABEL, wobei in diesem Fall der Standardwert der Sicherheitskennsatz ist, den Sie für den Schreibzugriff besitzen.

Einfügen in geschützte Zeilen

Wenn Sie eine neue Zeile in eine Tabelle mit geschützten Zeilen einfügen, brauchen Sie keinen Wert für die Spalte des Typs DB2SECURITYLABEL anzugeben. Wenn Sie keinen Wert für diese Spalte angeben, wird die Spalte automatisch mit dem Sicherheitskennsatz gefüllt, der Ihnen für den Schreibzugriff erteilt wurde. Wenn Ihnen kein Sicherheitskennsatz für den Schreibzugriff erteilt wurde, wird ein Fehler zurückgegeben und die Einfügung schlägt fehl.

Durch die Verwendung solcher integrierter Funktionen wie SECLABEL können Sie explizit einen Sicherheitskennsatz angeben, der in eine Spalte des Typs DB2SECURITYLABEL einzufügen ist. Der angegebene Sicherheitskennsatz wird jedoch nur verwendet, wenn Ihre LBAC-Berechtigungsnachweise einen Schreibzugriff auf die Daten zulassen würden, die durch den Sicherheitskennsatz geschützt werden, den Sie einzufügen versuchen.

Wenn Sie einen Sicherheitskennsatz angeben, der Ihnen keinen Schreibzugriff gewähren würde, hängt die weitere Verarbeitung von der Sicherheitsrichtlinie ab, welche die Tabelle schützt. Wenn die Sicherheitsrichtlinie die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL enthält, schlägt die Einfügung fehl und es wird ein Fehler zurückgegeben. Wenn die Sicherheitsrichtlinie die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL nicht enthält oder stattdessen die Option OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL enthält, wird der Sicherheitskennsatz, den Sie angeben, ignoriert und der Sicherheitskennsatz für den Schreibzugriff verwendet, sofern Sie einen haben. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird ein Fehler zurückgegeben.

Beispiele

Die Tabelle T1 wird durch eine Sicherheitsrichtlinie mit dem Namen P1 geschützt, die ohne die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL erstellt wurde. Die Tabelle T1 enthält zwei Spalten, jedoch keine Zeilen. Die Spalten haben die Bezeichnungen LASTNAME und LABEL. Die Spalte LABEL hat den Datentyp DB2SECURITYLABEL.

Der Benutzer Joe besitzt den Sicherheitskennsatz L2 für den Schreibzugriff. Nehmen Sie an, dass ihm der Sicherheitskennsatz L2 den Schreibzugriff auf Daten erlaubt, die durch den Sicherheitskennsatz L2 geschützt werden, jedoch keinen Schreibzugriff auf Daten, die durch die Sicherheitskennsätze L1 oder L3 geschützt werden.

Joe setzt die folgende SQL-Anweisung ab:

```
INSERT INTO T1 (LASTNAME, DEPTNO) VALUES ('Rjaibi', 11)
```

Weil kein Sicherheitskennsatz in der INSERT-Anweisung angegeben wurde, wird Joes Sicherheitskennsatz für den Schreibzugriff in die Spalte LABEL der Zeile eingefügt.

Die Tabelle T1 sieht nun folgendermaßen aus:

Tabelle 18.

LASTNAME	LABEL
Rjaibi	L2

Joe setzt die folgende SQL-Anweisung ab, in der er den Sicherheitskennsatz explizit angibt, der in die Spalte LABEL einzufügen ist:

```
INSERT INTO T1 VALUES ('Miller', SECLABEL_BY_NAME('P1', 'L1') )
```

Die Funktion SECLABEL_BY_NAME in der Anweisung gibt einen Sicherheitskennsatz zurück, der Teil der Sicherheitsrichtlinie P1 ist und den Namen L1 hat. Joe ist jedoch nicht berechtigt, in Daten zu schreiben, die durch den Sicherheitskennsatz L1 geschützt werden. Daher darf er den Wert L1 nicht in die Spalte LABEL einfügen.

Da die Sicherheitsrichtlinie, welche die Tabelle T1 schützt, ohne die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL erstellt wurde, wird stattdessen der Sicherheitskennsatz eingefügt, den Joe für den Schreibzugriff besitzt. Es wird weder ein Fehler noch eine Nachricht zurückgegeben.

Die Tabelle sieht nun folgendermaßen aus:

Tabelle 19.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2

Wenn die Sicherheitsrichtlinie, welche die Tabelle schützt, mit der Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL erstellt worden wäre, wäre die Einfügung unter Rückgabe eines Fehlers fehlgeschlagen.

Als Nächstes wird Joe eine Freistellung für eine der LBAC-Regeln erteilt. Nehmen Sie an, dass seine neuen LBAC-Berechtigungsnachweise ihm nun einen Schreibzugriff auf Daten gestatten, die durch die Sicherheitskennsätze L1 und L2 geschützt werden. Der Sicherheitskennsatz, der Joe für den Schreibzugriff erteilt ist, bleibt unverändert L2.

Joe setzt die folgende SQL-Anweisung ab:

```
INSERT INTO T1 VALUES ('Bird', SECLABEL_BY_NAME('P1', 'L1') )
```

Aufgrund der neuen LBAC-Berechtigungsnachweise hat Joe nun Schreibzugriff auf Daten, die durch den Sicherheitskennsatz L1 geschützt werden. Die Einfügung des Werts L1 wird daher zugelassen. Die Tabelle sieht nun folgendermaßen aus:

Tabelle 20.

LASTNAME	LABEL
Rjaibi	L2
Miller	L2
Bird	L1

Aktualisieren von LBAC-geschützten Daten

Ihre LBAC-Berechtigungsnachweise müssen Ihnen Schreibzugriff auf Daten erlauben, damit Sie sie aktualisieren können. Für die Aktualisierung einer geschützten Zeile müssen Ihnen Ihre LBAC-Berechtigungsnachweise außerdem den Lesezugriff auf die Zeile erlauben.

Aktualisieren geschützter Spalten

Wenn Sie versuchen, Daten in einer geschützten Spalte zu aktualisieren, werden Ihre LBAC-Berechtigungsnachweise mit dem Sicherheitskennsatz verglichen, der die Spalte schützt. Der Vergleich, der vorgenommen wird, bezieht sich auf den Schreibzugriff. Wenn der Schreibzugriff blockiert wird, wird ein Fehler zurückgegeben und die Ausführung der Anweisung schlägt fehl. Anderenfalls wird die Aktualisierung fortgesetzt.

Detaillierte Informationen dazu, wie Ihre LBAC-Berechtigungsnachweise mit einem Sicherheitskennsatz verglichen werden, finden Sie in dem Abschnitt über den Vergleich von LBAC-Sicherheitskennsätzen.

Beispiel:

Nehmen Sie an, in einer Tabelle T1 werden die Spalte DEPTNO durch einen Sicherheitskennsatz L2 und die Spalte PAYSCALE durch einen Sicherheitskennsatz L3 geschützt. Die Tabelle T1 sieht einschließlich ihrer Daten wie folgt aus:

Tabelle 21. Tabelle T1

EMPNO	LASTNAME	DEPTNO *geschützt durch* L2	PAYSCALE *geschützt durch* L3
1	Rjaibi	11	4
2	Miller	11	7
3	Bird	11	9

Der Benutzer Lhakpa besitzt keine LBAC-Berechtigungsnachweise. Er setzt die folgende SQL-Anweisung ab:

```
UPDATE T1 SET EMPNO = 4
   WHERE LASTNAME = "Bird"
```

Diese Anweisung wird ohne Fehler ausgeführt, da sie keine geschützten Spalten aktualisiert. Die Tabelle T1 sieht nun folgendermaßen aus:

Tabelle 22. Tabelle T1 nach der Aktualisierung

EMPNO	**LASTNAME**	**DEPTNO** *geschützt durch* **L2**	**PAYSCALE** *geschützt durch* **L3**
1	Rjaibi	11	4
2	Miller	11	7
4	Bird	11	9

Als Nächstes setzt Lhakpa die folgende SQL-Anweisung ab:

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

Die Ausführung dieser Anweisung schlägt unter Rückgabe eines Fehlers fehl, weil die Spalte DEPTNO geschützt wird und der Benutzer Lhakpa keine LBAC-Berechtigungsnachweise besitzt.

Nehmen Sie an, dem Benutzer Lhakpa werden LBAC-Berechtigungsnachweise erteilt, die ihm die in der folgenden Tabelle zusammengefassten Zugriffsmöglichkeiten gewähren. Wie diese Berechtigungsnachweise im Einzelnen aussehen und welche Elemente sich in den Sicherheitskennsätzen befinden, spielt für dieses Beispiel keine wichtige Rolle.

Schützender Sicherheitskennsatz der Daten	**Lesezugriff?**	**Schreibzugriff?**
L2	Nein	Ja
L3	Nein	Nein

Lhakpa setzt die folgende SQL-Anweisung erneut ab:

```
UPDATE T1 SET DEPTNO = 55
   WHERE LASTNAME = "Miller"
```

Diesmal wird die Anweisung ohne Fehler ausgeführt, weil die LBAC-Berechtigungsnachweise des Benutzers Lhakpa ihm einen Schreibzugriff auf Daten erlauben, die durch den Sicherheitskennsatz geschützt werden, der die Spalte DEPTNO schützt. Dabei spielt es keine Rolle, dass er nicht berechtigt ist, Daten eben dieser Spalte zu lesen. Die Daten der Tabelle T1 sehen nun folgendermaßen aus:

Tabelle 23. Tabelle T1 nach der zweiten Aktualisierung

EMPNO	**LASTNAME**	**DEPTNO** *geschützt durch* **L2**	**PAYSCALE** *geschützt durch* **L3**
1	Rjaibi	11	4
2	Miller	55	7
4	Bird	11	9

Als Nächstes setzt Lhakpa die folgende SQL-Anweisung ab:

```
UPDATE T1 SET DEPTNO = 55, PAYSCALE = 4
   WHERE LASTNAME = "Bird"
```

Die Spalte PAYSCALE wird durch den Sicherheitskennsatz L3 geschützt, und die LBAC-Berechtigungsnachweise des Benutzers Lhakpa erlauben ihm keinen Schreibzugriff auf sie. Da Lhakpa nicht in der Lage, Daten in die Spalte zu schreiben, schlägt die Aktualisierung fehl, sodass keine Daten geändert werden.

Aktualisieren geschützter Zeilen

Wenn Ihre LBAC-Berechtigungsnachweise keinen Lesezugriff auf eine Zeile zulassen, dann verhält sich das System so, als wäre diese Zeile für Sie nicht vorhanden. Daher haben Sie auch keine Möglichkeit, diese Zeile zu aktualisieren. Für Zeilen, die Sie lesen können, benötigen Sie zusätzlich einen Schreibzugriff, um die Zeile aktualisieren zu können.

Wenn Sie versuchen, eine Zeile zu aktualisieren, werden Ihre LBAC-Berechtigungsnachweise für den Schreibzugriff mit dem Sicherheitskennsatz verglichen, der die Zeile schützt. Wenn der Schreibzugriff blockiert wird, schlägt die Aktualisierung fehl und ein Fehler wird zurückgegeben. Wenn der Schreibzugriff nicht blockiert wird, wird die Aktualisierung fortgesetzt.

Die Aktualisierung wird abgesehen von der Behandlung der Spalte mit dem Datentyp DB2SECURITYLABEL auf die gleiche Weise wie eine Aktualisierung an einer nicht geschützten Zeile ausgeführt. Wenn Sie den Wert für diese Spalte nicht explizit festlegen, wird er automatisch auf den Sicherheitskennsatz gesetzt, den Sie für den Schreibzugriff besitzen. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird ein Fehler zurückgegeben und die Ausführung der Anweisung schlägt fehl.

Wenn die Aktualisierung explizit einen Wert für die Spalte mit dem Datentyp DB2SECURITYLABEL angibt, werden Ihre LBAC-Berechtigungsnachweise erneut überprüft. Wenn die Aktualisierung, die Sie auszuführen versuchen, eine Zeile erstellen würde, auf die Ihre LBAC-Berechtigungsnachweise keinen Schreibzugriff zuließen, hängt die weitere Verarbeitung von der Sicherheitsrichtlinie ab, welche die Tabelle schützt. Wenn die Sicherheitsrichtlinie die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL enthält, schlägt die Aktualisierung fehl und es wird ein Fehler zurückgegeben. Wenn die Sicherheitsrichtlinie die Option RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL nicht enthält oder stattdessen die Option OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL enthält, wird der Sicherheitskennsatz, den Sie angeben, ignoriert und der Sicherheitskennsatz für den Schreibzugriff verwendet, sofern Sie einen haben. Wenn Sie keinen Sicherheitskennsatz für den Schreibzugriff haben, wird ein Fehler zurückgegeben.

Beispiel:

Nehmen Sie an, die Tabelle T1 wird durch eine Sicherheitsrichtlinie mit dem Namen P1 geschützt und enthält eine Spalte mit dem Namen LABEL, die den Datentyp DB2SECURITYLABEL hat.

Die Tabelle T1 sieht einschließlich ihrer Daten wie folgt aus:

Tabelle 24. Tabelle T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1
2	Miller	11	L2
3	Bird	11	L3

Nehmen Sie an, die Benutzerin Jenni besitzt LBAC-Berechtigungsnachweise, die ihr einen Lese- und einen Schreibzugriff auf Daten erlauben, die durch die Sicherheitskennsätze L0 und L1 geschützt werden, jedoch nicht auf Daten, die durch beliebige andere Sicherheitskennsätze geschützt werden. Der Sicherheitskennsatz, den Sie für den Schreib- und den Lesezugriff besitzt, ist jeweils L0. Die vollständigen Details ihrer Berechtigungsnachweise sowie die einzelnen Elemente, die in den Kennsätzen enthalten sind, spielen für dieses Beispiel keine wichtige Rolle.

Jenni setzt die folgende SQL-Anweisung ab:

```
SELECT * FROM T1
```

Jenni wird nur eine Zeile aus der Tabelle angezeigt:

Tabelle 25. Ergebnis der SELECT-Abfrage von Jenni

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	11	L1

Die Zeilen, die durch die Kennsätze L2 und L3 geschützt werden, gehören nicht zur Ergebnismenge, weil Jennis LBAC-Berechtigungsnachweise keinen Lesezugriff auf diese Zeilen zulassen. Für Jenni verhält es sich so, als ob diese Zeilen nicht vorhanden wären.

Jenni setzt die folgenden SQL-Anweisungen ab:

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11;
SELECT * FROM T1;
```

Die von der Abfrage zurückgegebene Ergebnismenge sieht folgendermaßen aus:

Tabelle 26. Ergebnis der UPDATE-Anweisung und der SELECT-Abfrage von Jenni

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0

Tatsächlich sehen die vollständigen Daten der Tabelle wie folgt aus:

Tabelle 27. Tabelle T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L0
2	Miller	11	L2
3	Bird	11	L3

Die Anweisung wurde ohne Fehler ausgeführt, jedoch hat sie nur die erste Zeile geändert. Die zweite und dritte Zeile sind für Jenni nicht lesbar, sodass sie nicht für die Aktualisierung durch die Anweisung ausgewählt wurden, obwohl sie die Bedingung der WHERE-Klausel erfüllen.

Beachten Sie, dass der Wert der Spalte LABEL in der aktualisierten Zeile geändert wurde, obwohl diese Spalte in der UPDATE-Anweisung nicht explizit angegeben war. Die Spalte wurde auf den Sicherheitskennsatz gesetzt, den Jenni für den Schreibzugriff besaß.

Nehmen Sie jetzt an, dass Jenni LBAC-Berechtigungsnachweise für den Lesezugriff auf Daten erteilt werden, die durch einen beliebigen Sicherheitskennsatz geschützt werden. Ihre LBAC-Berechtigungsnachweise für den Schreibzugriff werden nicht geändert. Sie kann auch weiterhin nur Daten schreiben, die durch die Sicherheitskennsätze L0 und L1 geschützt werden.

Jenni setzt die folgende SQL-Anweisung erneut ab:

```
UPDATE T1 SET DEPTNO = 44 WHERE DEPTNO = 11
```

Diesmal schlägt die Aktualisierung wegen der zweiten und der dritten Zeile fehl. Jenni kann diese Zeilen lesen, sodass sie für die Aktualisierung durch die Anweisung ausgewählt werden. Sie ist jedoch nicht berechtigt, in diese Zeilen zu schreiben, weil sie durch die Sicherheitskennsätze L2 und L3 geschützt werden. Die Aktualisierung findet nicht statt, und es wird ein Fehler zurückgegeben.

Jenni setzt nun die folgende SQL-Anweisung ab:

```
UPDATE T1
 SET DEPTNO = 55, LABEL = SECLABEL_BY_NAME( 'P1', 'L2' )
 WHERE LASTNAME = "Rjaibi"
```

Die Funktion SECLABEL_BY_NAME in der Anweisung gibt den Sicherheitskennsatz mit dem Namen L2 zurück. Jenni versucht, den Sicherheitskennsatz, der die erste Zeile schützt, explizit festzulegen. Jennis LBAC-Berechtigungsnachweise erlauben ihr, die erste Zeile zu lesen, sodass diese für die Aktualisierung ausgewählt wird. Ihre LBAC-Berechtigungsnachweise erlauben ihr einen Schreibzugriff auf Zeilen, die durch den Sicherheitskennsatz L0 geschützt werden, sodass sie die Zeile aktualisieren darf. Allerdings würden ihre LBAC-Berechtigungsnachweise ihr keinen Schreibzugriff auf eine Zeile erlauben, die durch den Sicherheitskennsatz L2 geschützt wird. Daher ist sie nicht berechtigt, die Spalte LABEL auf diesen Wert zu setzen. Die Ausführung der Anweisung schlägt unter Rückgabe eines Fehlers fehl. Es werden keine Spalten in der Zeile aktualisiert.

Jenni setzt nun die folgende SQL-Anweisung ab:

```
UPDATE T1 SET LABEL = SECLABEL_BY_NAME( 'P1', 'L1' ) WHERE LASTNAME = "Rjaibi"
```

Die Anweisung wird erfolgreich ausgeführt, weil sie in eine Zeile schreiben könnte, die durch den Sicherheitskennsatz L1 geschützt wird.

Die Tabelle T1 sieht nun folgendermaßen aus:

Tabelle 28. Tabelle T1

EMPNO	LASTNAME	DEPTNO	LABEL
1	Rjaibi	44	L1
2	Miller	11	L2
3	Bird	11	L3

Aktualisieren geschützter Zeilen, die geschützte Spalten enthalten

Wenn Sie versuchen, geschützte Spalten in einer Tabelle mit geschützten Zeilen zu aktualisieren, müssen Ihre LBAC-Berechtigungsnachweise einen Schreibzugriff auf alle geschützten Spalten zulassen, die von der Aktualisierung betroffen sind. Ansonsten schlägt die Aktualisierung fehl und ein Fehler wird zurückgegeben.

Dieser Sachverhalt wird im vorausgehenden Abschnitt über das Aktualisieren geschützter Spalten beschrieben. Wenn Sie berechtigt sind, alle geschützten Spalten, die von der Aktualisierung betroffen sind, zu aktualisieren, können Sie trotzdem nur die Zeilen aktualisieren, auf die Ihnen Ihre LBAC-Berechtigungsnachweise sowohl einen Lesezugriff als auch einen Schreibzugriff ermöglichen. Dieser Sachverhalt wird im vorausgehenden Abschnitt über das Aktualisieren geschützter Zeilen beschrieben. Die Behandlung einer Spalte mit dem Datentyp DB2SECURITYLABEL bleibt gleich, unabhängig davon, ob die Aktualisierung geschützte Spalten betrifft oder nicht.

Wenn die Spalte, die den Datentyp DB2SECURITYLABEL hat, selbst eine geschützte Spalte ist, müssen Ihre LBAC-Berechtigungsnachweise einen Schreibzugriff auf diese Spalte zulassen. Ansonsten können Sie keine der Zeilen in der Tabelle aktualisieren.

Löschen oder Entfernen von LBAC-geschützten Daten

Wenn Ihre LBAC-Berechtigungsnachweise keinen Lesezugriff auf eine Zeile zulassen, dann verhält sich das System so, als wäre diese Zeile für Sie nicht vorhanden. Daher haben Sie auch keine Möglichkeit, sie zu löschen. Zum Löschen einer Zeile, die Sie lesen können, müssen Ihnen Ihre LBAC-Berechtigungsnachweise auch den Schreibzugriff auf diese Zeile erlauben. Zum Löschen einer Zeile in einer Tabelle, die geschützte Spalten enthält, müssen Sie über LBAC-Berechtigungsnachweise verfügen, die Ihnen einen Schreibzugriff auf alle geschützten Spalten in der Tabelle erlauben.

Löschen geschützter Zeilen

Wenn Sie versuchen, eine Zeile zu löschen, werden Ihre LBAC-Berechtigungsnachweise für den Schreibzugriff mit dem Sicherheitskennsatz verglichen, der die Zeile schützt. Wenn der schützende Sicherheitskennsatz den Schreibzugriff durch Ihre LBAC-Berechtigungsnachweise blockiert, schlägt die Ausführung der DELETE-Anweisung fehl. Darüber hinaus wird ein Fehler zurückgegeben, und keine Zeile wird gelöscht.

Beispiel

Die geschützte Tabelle T1 enthält folgende Zeilen:

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

Nehmen Sie an, die Benutzerin Pat besitzt LBAC-Berechtigungsnachweise, die ihr den in der folgenden Tabelle dargestellten Zugriff erlauben:

Sicherheitskennsatz	Lesezugriff?	Schreibzugriff?
L1	Ja	Ja
L2	Ja	Nein
L3	Nein	Nein

Die exakten Details ihrer LBAC-Berechtigungsnachweise und der Sicherheitskennsätze sind für dieses Beispiel nicht relevant.

Pat setzt die folgende SQL-Anweisung ab:

```
SELECT * FROM T1 WHERE DEPTNO != 999
```

Die Anweisung wird ausgeführt und liefert die folgende Ergebnismenge:

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2

Die letzte Zeile von T1 ist nicht in den Ergebnissen enthalten, weil Pat keinen Lesezugriff auf diese Zeile besitzt. Es verhält sich so, als ob diese Zeile für Pat nicht vorhanden wäre.

Pat setzt die folgende SQL-Anweisung ab:

```
DELETE FROM T1 WHERE DEPTNO != 999
```

Pat hat keinen Schreibzugriff auf die erste und die dritte Zeile, die beide durch den Sicherheitskennsatz L2 geschützt werden. Obwohl sie also die Zeilen lesen kann, ist sie nicht berechtigt, sie zu löschen. Die Ausführung der DELETE-Anweisung schlägt fehl, und keine Zeilen werden gelöscht.

Pat setzt die folgende SQL-Anweisung ab:

```
DELETE FROM T1 WHERE DEPTNO = 77;
```

Diese Anweisung wird erfolgreich ausgeführt, weil Pat Schreibzugriff auf die Zeile mit dem Wert 'Miller' in der Spalte LASTNAME hat. Dies ist die einzige Zeile, die durch die Anweisung ausgewählt wird. Die Zeile mit dem Wert 'Fielding' in der Spalte LASTNAME wird nicht ausgewählt, weil Pats LBAC-Berechtigungsnachweise keinen Lesezugriff auf diese Zeile zulassen. Diese Zeile wird für die Löschung nie in Betracht gezogen, sodass kein Fehler auftritt.

Die tatsächlichen Zeilen der Tabelle sehen nun folgendermaßen aus:

LASTNAME	DEPTNO	LABEL
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

Löschen von Zeilen, die geschützte Spalten enthalten

Zum Löschen einer Zeile in einer Tabelle, die geschützte Spalten enthält, müssen Sie über LBAC-Berechtigungsnachweise verfügen, die Ihnen einen Schreibzugriff auf alle geschützten Spalten in der Tabelle erlauben. Wenn sich in der Tabelle eine Zeile befindet, auf die Ihnen Ihre LBAC-Berechtigungsnachweise keinen Schreibzugriff gewähren, schlägt das Löschen fehl und es wird ein Fehler zurückgegeben.

Wenn die Tabelle sowohl geschützte Spalten als auch geschützte Zeilen enthält, müssen Sie zum Löschen einer bestimmten Zeile über LBAC-Berechtigungsnachweise verfügen, die Ihnen einen Schreibzugriff auf jede geschützte Spalte in der Tabelle und außerdem einen Lese- und Schreibzugriff auf die Zeile erlauben, die Sie löschen möchten.

Beispiel

In der geschützten Tabelle T1 wird die Spalte DEPTNO durch den Sicherheitskennsatz L2 geschützt. Die Tabelle T1 enthält die folgenden Zeilen:

LASTNAME	**DEPTNO** *geschützt durch* **L2**	**LABEL**
Rjaibi	55	L2
Miller	77	L1
Bird	55	L2
Fielding	77	L3

Nehmen Sie an, der Benutzer Benny verfügt über LBAC-Berechtigungsnachweise, die ihm den in der folgenden Tabelle dargestellten Zugriff erlauben:

Sicherheitskennsatz	**Lesezugriff?**	**Schreibzugriff?**
L1	Ja	Ja
L2	Ja	Nein
L3	Nein	Nein

Die exakten Details seiner LBAC-Berechtigungsnachweise und der Sicherheitskennsätze sind für dieses Beispiel nicht relevant.

Benny setzt die folgende SQL-Anweisung ab:

```
DELETE FROM T1 WHERE DEPTNO = 77
```

Die Ausführung der Anweisung schlägt fehl, weil Benny keinen Schreibzugriff auf die Spalte DEPTNO besitzt.

Nun werden LBAC-Berechtigungsnachweise des Benutzers Benny geändert, sodass er über den in der folgenden Tabelle dargestellten Zugriff verfügt:

Sicherheitskennsatz	**Lesezugriff?**	**Schreibzugriff?**
L1	Ja	Ja
L2	Ja	Ja
L3	Ja	Nein

Benny setzt die folgende SQL-Anweisung erneut ab:

```
DELETE FROM T1 WHERE DEPTNO = 77
```

Diesmal hat Benny Schreibzugriff auf die Spalte DEPTNO, sodass die Ausführung der DELETE-Anweisung fortgesetzt wird. Die DELETE-Anweisung wählt nur die Zeile aus, die den Wert 'Miller' in der Spalte LASTNAME enthält. Die Zeile mit dem Wert 'Fielding' in der Spalte LASTNAME wird nicht ausgewählt, weil Bennys LBAC-Berechtigungsnachweise keinen Lesezugriff auf diese Zeile zulassen. Da die Zeile von der Anweisung zum Löschen nicht ausgewählt wird, spielt es keine Rolle, dass Benny keinen Schreibzugriff auf die Zeile besitzt.

Die eine ausgewählte Zeile wird durch den Sicherheitskennsatz L1 geschützt. Bennys LBAC-Berechtigungsnachweise erlauben ihm einen Schreibzugriff auf Daten, die durch den Sicherheitskennsatz L1 geschützt werden, sodass die Löschung erfolgreich ausgeführt wird.

Die tatsächlichen Zeilen in der Tabelle T1 sehen nun folgendermaßen aus:

LASTNAME	**DEPTNO** *geschützt durch* **L2**	**LABEL**
Rjaibi	55	L2
Bird	55	L2
Fielding	77	L3

Löschen geschützter Daten

Sie können eine Spalte, die durch einen Sicherheitskennsatz geschützt wird, nur löschen, wenn Ihre LBAC-Berechtigungsnachweise einen Schreibzugriff auf diese Spalte zulassen.

Eine Spalte mit dem Datentyp DB2SECURITYLABEL kann nicht aus einer Tabelle gelöscht werden. Um diese Spalte zu entfernen, müssen Sie zuerst die Sicherheitsrichtlinie der Tabelle löschen. Wenn Sie die Sicherheitsrichtlinie löschen, ist die Tabelle nicht mehr durch LBAC geschützt und der Datentyp der Spalte wird automatisch von DB2SECURITYLABEL in VARCHAR(128) FOR BIT DATA geändert. Anschließend kann die Spalte gelöscht werden.

Ihre LBAC-Berechtigungsnachweise hindern Sie nicht daran, gesamte Tabellen oder Datenbanken zu löschen, die geschützte Daten enthalten. Wenn Sie unter normalen Umständen die Berechtigung besitzen, eine Tabelle oder Datenbank zu löschen, benötigen Sie dazu keine LBAC-Berechtigungsnachweise, selbst wenn die Datenbank geschützte Daten enthält.

Entfernen des LBAC-Schutzes von Daten

Sie müssen über die Berechtigung SECADM verfügen, um die Sicherheitsrichtlinie von einer Tabelle entfernen zu können. Verwenden Sie die Klausel DROP SECURITY POLICY der Anweisung ALTER TABLE, um die Sicherheitsrichtlinie von einer Tabelle zu entfernen. Dadurch wird automatisch auch der Schutz von allen Zeilen und allen Spalten der Tabelle entfernt.

Entfernen des Schutzes von Zeilen

In einer Tabelle, die geschützte Zeilen enthält, muss jede Zeile durch einen Sicherheitskennsatz geschützt werden. Es gibt keine Möglichkeit, den LBAC-Schutz von einzelnen Zeilen zu entfernen.

Eine Spalte des Typs DB2SECURITYLABEL kann nicht geändert oder entfernt werden, ohne zuvor die Sicherheitsrichtlinie von der Tabelle zu entfernen.

Entfernen des Schutzes von Spalten

Der Schutz einer Spalte kann mithilfe der Klausel DROP COLUMN SECURITY der SQL-Anweisung ALTER TABLE entfernt werden. Um den Schutz von einer Spalte entfernen zu können, müssen Sie neben den normalen Zugriffsrechten und Berechtigungen zum Ändern einer Tabelle auch die LBAC-Berechtigungsnachweise für einen Lese- und Schreibzugriff auf die Spalte besitzen.

Kapitel 5. Verwenden des Systemkatalogs für Sicherheitsinformationen

Informationen über die einzelnen Datenbanken werden automatisch in einer Gruppe von Sichten, dem so genannten Systemkatalog, gepflegt, der beim Erstellen der Datenbank erstellt werden. Dieser Systemkatalog beschreibt Tabellen, Spalten, Indizes, Programme, Zugriffsrechte und andere Objekte.

In den folgenden Sichten und Tabellenfunktionen werden Informationen über Zugriffsrechte für Benutzer, Kennungen für Benutzer, die Zugriffsrechte erteilen sowie Informationen zu Objekteigentumsrechten aufgeführt:

SYSCAT.DBAUTH
Listet die Zugriffsrechte für Datenbanken auf.

SYSCAT.TABAUTH
Listet die Zugriffsrechte für Tabellen und Sichten auf.

SYSCAT.COLAUTH
Listet die Zugriffsrechte für Spalten auf.

SYSCAT.PACKAGEAUTH
Listet die Zugriffsrechte für Pakete auf.

SYSCAT.INDEXAUTH
Listet die Zugriffsrechte für Indizes auf.

SYSCAT.SCHEMAAUTH
Listet die Zugriffsrechte für Schemata auf.

SYSCAT.PASSTHRUAUTH
Listet das Zugriffsrecht für Server auf.

SYSCAT.ROUTINEAUTH
Listet die Zugriffsrechte für Routinen (Funktionen, Methoden und gespeicherte Prozeduren) auf.

SYSCAT.SURROGATEAUTHIDS
Listet die Berechtigungs-IDs auf, für die eine andere Berechtigungs-ID als Ersatz fungieren kann.

Zugriffsrechte, die den Benutzern vom System verliehen wurden, weisen in der Spalte GRANTOR den Wert SYSIBM auf. SYSADM, SYSMAINT SYSCTRL und SYSMON werden nicht im Systemkatalog aufgelistet.

Durch die Anweisungen CREATE und GRANT werden Zugriffsrechte in den Systemkatalog eingetragen. Benutzer mit den Berechtigungen SYSADM und DBADM können Zugriffsrechte SELECT für die Systemkatalogsichten erteilen und widerrufen.

Abrufen von Berechtigungsnamen mit erteilten Zugriffsrechten

Über die Verwaltungssicht PRIVILEGES und andere Verwaltungssichten können Sie Informationen über die Berechtigungsnamen abrufen, denen Zugriffsrechte in einer Datenbank erteilt wurden.

Beispiel: Mit der folgenden Abfrage werden alle expliziten Zugriffsrechte und Berechtigungs-IDs, denen sie erteilt wurden, sowie weitere Informationen über die Verwaltungssicht PRIVILEGES abgerufen:

```
SELECT AUTHID, PRIVILEGE, OBJECTNAME, OBJECTSCHEMA, OBJECTTYPE FROM SYSIBMADM.PRIVILEGES
```

In der folgenden Abfrage wird die Verwaltungssicht AUTHORIZATIONIDS verwendet, um nach allen Berechtigungs-IDs zu suchen, denen Zugriffsrechte oder Berechtigungen erteilt wurden, und um die entsprechenden Typen anzuzeigen:

```
SELECT AUTHID, AUTHIDTYPE FROM SYSIBMADM.AUTHORIZATIONIDS
```

Mit der Verwaltungssicht SYSIBMADM.OBJECTOWNERS und der Tabellenfunktion SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID können Sie nach sicherheitsbezogenen Informationen suchen.

Vor Version 9.1 enthielt keine einzelne Systemkatalogsicht Informationen zu allen Zugriffsrechten. Bei Releases vor Version 9.1 werden mithilfe der folgenden Anweisung sämtliche Berechtigungsnamen mit Zugriffsrechten abgerufen:

```
 SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE   ' FROM SYSCAT.TABAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX   ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN  ' FROM SYSCAT.COLAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA  ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER  ' FROM SYSCAT.PASSTHRUAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3
```

Die durch diese Anweisung abgerufene Liste sollte in regelmäßigen Abständen mit Listen von Benutzer- und Gruppennamen verglichen werden, die in der Sicherheitseinrichtung des Systems definiert sind. Auf diese Weise können die Berechtigungsnamen ermittelt werden, die nicht mehr gültig sind.

Anmerkung: Wenn Sie ferne Datenbankclients unterstützen, ist der Berechtigungsname möglicherweise nur auf dem fernen Client und nicht auf Ihrem Datenbankserversystem definiert.

Abrufen aller Namen mit der Berechtigung DBADM

Die folgende Anweisung ruft alle Berechtigungsnamen ab, denen die Berechtigung DBADM direkt erteilt wurde:

```
SELECT DISTINCT GRANTEE, GRANTEETYPE FROM SYSCAT.DBAUTH
   WHERE DBADMAUTH = 'Y'
```

Anmerkung: Diese Abfrage gibt keine Informationen über Berechtigungsnamen zurück, die die DBADM-Berechtigung implizit durch die Berechtigung SYSADM erworben haben.

Abrufen der Namen mit Zugriffsberechtigung für eine Tabelle

Über die Verwaltungssicht PRIVILEGES und andere Verwaltungssichten können Sie Informationen über die Berechtigungsnamen abrufen, denen Zugriffsrechte in einer Datenbank erteilt wurden.

Die folgende Anweisung ruft alle Berechtigungsnamen (und deren Typen) ab, die direkt zum Zugriff auf die Tabelle EMPLOYEE mit dem Qualifikationsmerkmal JAMES berechtigt sind:

```
SELECT DISTINCT AUTHID, AUTHIDTYPE FROM SYSIBMADM.PRIVILEGES
    WHERE OBJECTNAME = 'EMPLOYEE' AND OBJECTSCHEMA = 'JAMES'
```

Bei Releases vor Version 9.1 werden mithilfe der folgenden Abfrage dieselben Informationen abgerufen:

```
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
   WHERE TABNAME = 'EMPLOYEE'
     AND TABSCHEMA = 'JAMES'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
   WHERE TABNAME = 'EMPLOYEE'
     AND TABSCHEMA = 'JAMES'
```

Mithilfe der folgenden Anweisung können Sie herausfinden, wer die Tabelle EMPLOYEE mit dem Qualifikationsmerkmal JAMES aktualisieren kann:

```
 SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
   WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
      (CONTROLAUTH  = 'Y' OR
       UPDATEAUTH IN ('G','Y'))
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
   WHERE DBADMAUTH = 'Y'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
   WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
   PRIVTYPE = 'U'
```

Diese Anweisung ruft alle Berechtigungsnamen mit der Berechtigung DBADM sowie die Namen ab, denen die Zugriffsrechte CONTROL oder UPDATE direkt erteilt wurden. Allerdings werden die Berechtigungsnamen von Benutzern, die nur über die Berechtigung SYSADM verfügen, nicht zurückgegeben.

Es ist zu beachten, dass einige Berechtigungsnamen Gruppen sein können, nicht nur Einzelbenutzer.

Abrufen aller Benutzern erteilter Zugriffsrechte

Durch Abfragen der Systemkatalogsichten können Benutzer eine Liste der Zugriffsrechte, über die sie verfügen, und eine Liste der Zugriffsrechte, die sie anderen Benutzern erteilt haben, abrufen.

Über die Verwaltungssicht PRIVILEGES und andere Verwaltungssichten können Sie Informationen über die Berechtigungsnamen abrufen, denen Zugriffsrechte in einer Datenbank erteilt wurden. Beispiel: Mit der folgenden Abfrage werden alle Zugriffsrechte abgerufen, die der Berechtigungs-ID der aktuellen Sitzung erteilt wurden:

```
SELECT * FROM SYSIBMADM.PRIVILEGES
 WHERE AUTHID = SESSION_USER AND AUTHIDTYPE = 'U'
```

Das Schlüsselwort SESSION_USER in dieser Anweisung ist ein Sonderregister, das dem Wert des Berechtigungsnamens des aktuellen Benutzers entspricht.

Bei Releases vor Version 9.1 werden mithilfe der Anweisungen in folgenden Beispielen dieselben Informationen bereitgestellt. Zum Beispiel wird durch die fol-

gende Anweisung eine Liste der Datenbankzugriffsrechte abgerufen, die direkt dem einzelnen Berechtigungsnamen JAMES erteilt wurden:

```
SELECT * FROM SYSCAT.DBAUTH
 WHERE GRANTEE = 'JAMES' AND GRANTEETYPE = 'U'
```

Mit der folgenden Anweisung wird eine Liste der Tabellenzugriffsrechte abgerufen, die direkt vom Benutzer JAMES erteilt wurden:

```
SELECT * FROM SYSCAT.TABAUTH
 WHERE GRANTOR  = 'JAMES'
```

Mit der folgenden Anweisung wird eine Liste der einzelnen Spaltenzugriffsrechte abgerufen, die direkt vom Benutzer JAMES erteilt wurden:

```
SELECT * FROM SYSCAT.COLAUTH
   WHERE GRANTOR  = 'JAMES'
```

Schützen der Systemkatalogsicht

Da in den Systemkatalogsichten jedes Objekt in der Datenbank beschrieben wird, möchten Sie möglicherweise, falls Sie über sensible Daten verfügen, den Zugriff auf diese Daten einschränken.

Sie können den Befehl CREATE DATABASE ... RESTRICTIVE zum Erstellen einer Datenbank verwenden, in der keine Zugriffsrechte automatisch der speziellen Gruppe PUBLIC erteilt werden. In diesem Fall kommt es zu keiner der folgenden normalen Standard-GRANT-Aktionen:

- CREATETAB
- BINDADD
- CONNECT
- IMPLSCHEMA
- EXECUTE with GRANT für alle Prozeduren im Schema SQLJ
- EXECUTE with GRANT für alle Funktionen und Prozeduren im Schema SYSPROC
- BIND für alle im Schema NULLID erstellten Pakete
- EXECUTE für alle im Schema NULLID erstellten Pakete
- CREATEIN im Schema SQLJ
- CREATEIN im Schema NULLID
- USE für den Tabellenbereich USERSPACE1
- SELECT-Zugriff auf die SYSIBM-Katalogtabellen
- SELECT-Zugriff auf die SYSCAT-Katalogsichten
- SELECT-Zugriff auf die SYSIBMADM-Verwaltungssichten
- SELECT-Zugriff auf die SYSSTAT-Katalogsichten
- UPDATE-Zugriff auf die SYSSTAT-Katalogsichten

Wenn Sie eine Datenbank mit der Option RESTRICTIVE erstellt haben und Sie überprüfen möchten, ob die Berechtigungen eingeschränkt sind, die Sie PUBLIC erteilt haben, können Sie die folgende Abfrage absetzen, um zu prüfen, auf welche Schemata PUBLIC zugreifen kann:

```
SELECT DISTINCT OBJECTSCHEMA FROM SYSIBMADM.PRIVILEGES WHERE AUTHID='PUBLIC'
```

```
OBJECTSCHEMA
------------
SYSFUN
SYSIBM
SYSPROC
```

Wenn Sie wissen möchten, welchen Zugriff PUBLIC noch auf SYSIBM hat, können Sie die folgende Abfrage absetzen, um zu überprüfen, welche Zugriffsrechte für SYSIBM erteilt wurden. Die Ergebnisse zeigen, dass nur EXECUTE für bestimmte Prozeduren und Funktionen erteilt wurde.

```
SELECT * FROM SYSIBMADM.PRIVILEGES WHERE OBJECTSCHEMA = 'SYSIBM'
AUTHID       AUTHIDTYPE  PRIVILEGE   GRANTABLE  OBJECTNAME         OBJECTSCHEMA    OBJECTTYPE
---------... ----------  ----------  ---------  ---------------... ------------... ----------
PUBLIC       G           EXECUTE     N          SQL060207192129400 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129700 SYSPROC         FUNCTION
PUBLIC       G           EXECUTE     N          SQL060207192129701 SYSPROC
...
PUBLIC       G           EXECUTE     Y          TABLES             SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          TABLEPRIVILEGES    SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          STATISTICS         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SPECIALCOLUMNS     SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURES         SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PROCEDURECOLS      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          PRIMARYKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          FOREIGNKEYS        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLUMNS            SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          COLPRIVILEGES      SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          UDTS               SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          GETTYPEINFO        SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGE       SYSIBM          PROCEDURE
PUBLIC       G           EXECUTE     Y          SQLCAMESSAGECCSID  SYSIBM          PROCEDURE
```

Anmerkung: Die Verwaltungssicht SYSIBMADM.PRIVILEGES ist ab Version 9.1 des DB2-Datenbankmanagers verfügbar.

Bei Releases vor Version 9.1 des DB2-Datenbankmanagers wird während der Datenbankerstellung das Zugriffsrecht SELECT für die Systemkatalogsichten der Gruppe PUBLIC erteilt. In den meisten Fällen stellt dies kein Sicherheitsproblem dar. Bei sehr sensiblen Daten kann es jedoch zu Problemen kommen, da in diesen Tabellen jedes Objekt der Datenbank beschrieben wird. Wenn dies der Fall ist, kann das Zugriffsrecht SELECT für PUBLIC widerrufen werden. Anschließend kann das Zugriffsrecht SELECT nach Bedarf bestimmten Benutzern erteilt werden. Das Erteilen und Widerrufen des Zugriffsrechts SELECT für die Systemkatalogsichten erfolgt genauso wie für jede andere Sicht, jedoch benötigen Sie dafür die Berechtigung SYSADM oder DBADM.

Sie sollten zumindest in Betracht ziehen, falls kein Benutzer wissen soll, auf welche Objekte andere Benutzer Zugriff haben, den Zugriff auf die folgenden Katalog- und Verwaltungssichten einzuschränken:

- SYSCAT.COLAUTH
- SYSCAT.DBAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.PASSTHRUAUTH
- SYSCAT.ROUTINEAUTH
- SYSCAT.SCHEMAAUTH
- SYSCAT.SECURITYLABELACCESS

- SYSCAT.SECURITYPOLICYEXEMPTIONS
- SYSCAT.SEQUENCEAUTH
- SYSCAT.SURROGATEAUTHIDS
- SYSCAT.TABAUTH
- SYSCAT.TBSPACEAUTH
- SYSCAT.XSROBJECTAUTH
- SYSIBMADM.AUTHORIZATIONIDS
- SYSIBMADM.OBJECTOWNERS
- SYSIBMADM.PRIVILEGES

Dadurch wird verhindert, dass Informationen über Zugriffsrechte von Benutzern für alle Benutzer mit Zugriff auf die Datenbank verfügbar werden.

Sie sollten auch die Spalten überprüfen, für die die Statistikdaten gesammelt wurden. Einige der im Systemkatalog aufgezeichneten Statistikdaten könnten Datenwerte enthalten, die in Ihrer Umgebung sensible Informationen darstellen. Wenn diese Statistikdaten sensible Daten enthalten, können Sie das Zugriffsrecht SELECT auf die Katalogsichten SYSCAT.COLUMNS und SYSCAT.COLDIST für PUBLIC widerrufen.

Wenn Sie den Zugriff auf die Systemkatalogsichten beschränken möchten, können Sie Sichten definieren, damit jeder Berechtigungsname Informationen über seine eigenen Zugriffsrechte abrufen kann.

Zum Beispiel enthält die folgende Sicht MYSELECTS den Eigner und den Namen jeder Tabelle, für die dem Berechtigungsnamen eines Benutzers direkt das Zugriffsrecht SELECT erteilt wurde:

```
CREATE VIEW MYSELECTS AS
   SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
   WHERE GRANTEETYPE = 'U'
     AND GRANTEE = USER
     AND SELECTAUTH = 'Y'
```

Das Schlüsselwort USER in dieser Anweisung ist gleich dem Wert des Berechtigungsnamens der aktuellen Sitzung.

Mit der folgenden Anweisung wird die Sicht für jeden Berechtigungsnamen verfügbar gemacht:

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

Außerdem sollten Sie nicht vergessen, das Zugriffsrecht SELECT für die Sicht- und Basistabelle durch Absetzen der folgenden beiden Anweisungen zu widerrufen:

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC
REVOKE SELECT ON TABLE SYSIBM.SYSTABAUTH FROM PUBLIC
```

Sicherheitsaspekte

Für eine erfolgreiche Verwaltung der Sicherheit müssen Sie mit den indirekten Methoden vertraut sein, mit denen sich Benutzer Zugriff auf Daten verschaffen können. Darüber hinaus müssen Sie die Standardzugriffsrechte für bestimmte Systemtabellen kennen, die bei der Erstellung einer Datenbank erteilt werden.

Indirekte Methoden des Zugriffs auf Daten

Beachten Sie die folgenden indirekten Methoden, durch die sich Benutzer Zugriff auf Daten verschaffen können, für die sie möglicherweise nicht berechtigt sind:

- **Katalogsichten:** In den Katalogsichten des DB2-Datenbanksystems werden Metadaten und Statistiken über Datenbankobjekte gespeichert. Benutzer mit dem Zugriffsrecht SELECT für die Katalogsichten können Erkenntnisse über Daten gewinnen, für die sie möglicherweise nicht qualifiziert sind. Zur besseren Sicherheit müssen Sie sicherstellen, dass nur qualifizierte Benutzer Zugriff auf die Katalogsichten haben.

  Anmerkung: In DB2 Universal Database Version 8 und früheren Versionen wurde das Zugriffsrecht SELECT standardmäßig der Gruppe PUBLIC erteilt. In Datenbanksystemen von DB2 ab Version 9.1 können Benutzer wählen, ob das Zugriffsrecht SELECT für die Katalogtabellen der Gruppe PUBLIC erteilt werden soll oder nicht, indem sie die neue Option RESTRICTIVE im Befehl CREATE DATABASE verwenden.
- **Visual Explain:** Visual Explain zeigt den vom Abfrageoptimierungsprogramm ausgewählten Zugriffsplan für eine bestimmte Abfrage. Die Informationen von Visual Explain enthalten außerdem Statistiken zu Spalten, die in der Abfrage angegeben sind. Diese Statistiken können Aufschluss über den Inhalt einer Tabelle geben.
- **EXPLAIN-Momentaufnahme:** Die EXPLAIN-Momentaufnahme enthält komprimierte Informationen, die erfasst werden, wenn eine SQL- oder XQuery-Anweisung mit EXPLAIN bearbeitet wird. Sie wird als großes Binärobjekt (BLOB) in der Tabelle EXPLAIN_STATEMENT gespeichert und enthält Spaltenstatistiken, die Aufschluss über Tabellendaten geben können. Zur besseren Sicherheit sollte der Zugriff auf die EXPLAIN-Tabellen nur qualifizierten Benutzern erteilt werden.
- **Protokolllesefunktionen:** Ein Benutzer, der zur Ausführung einer Funktion berechtigt ist, die die Protokolle liest, kann sich Zugriff auf Daten verschaffen, für die er nicht berechtigt ist, wenn er das Format eines Protokollsatzes versteht. Die folgenden Funktionen lesen die Protokolle:

Funktion	Zum Ausführen der Funktion erforderliche Berechtigung
db2ReadLog	SYSADM oder DBADM
db2ReadLogNoConn	Keine

- **Replikation:** Wenn Sie Daten replizieren, werden auch die geschützten Daten an der Zielposition reproduziert. Zur besseren Sicherheit müssen Sie sicherstellen, dass die Zielposition mindestens ebenso sicher ist wie die Quellenposition.
- **Ausnahmetabellen:** Wenn Sie eine Ausnahmetabelle angeben, wenn Sie Daten in eine Tabelle laden, können Benutzer mit Zugriff auf die Ausnahmetabelle Informationen erlangen, für die sie möglicherweise nicht berechtigt sind. Zur besseren Sicherheit sollten Sie den Zugriff auf die Ausnahmetabelle nur berechtigten Benutzern erteilen und die Ausnahmetabelle sofort löschen, wenn Sie sie nicht mehr benötigen.
- **Backups von Tabellenbereichen oder Datenbanken:** Benutzer mit der Berechtigung zur Ausführung des BACKUP-Befehls können ein Backup einer Datenbank oder eines Tabellenbereichs, einschließlich aller geschützten Daten, erstellen und die Daten an einer beliebigen anderen Position wiederherstellen. Das Backup kann Daten umfassen, auf die der Benutzer ansonsten möglicherweise keinen Zugriff hat.

Der Befehl BACKUP kann von Benutzern mit der Berechtigung SYSADM, SYSCTRL oder SYSMAINT ausgeführt werden.

- **SQL-Anweisung SET SESSION AUTHORIZATION:** In DB2 Universal Database Version 8 und früheren Versionen konnte ein Benutzer mit der Berechtigung DBADM mithilfe der SQL-Anweisung SET SESSION AUTHORIZATION die Sitzungsberechtigungs-ID auf einen beliebigen Datenbankbenutzer setzen. In Datenbanksystemen von DB2 ab Version 9.1 muss ein Benutzer explizit durch die Anweisung GRANT SETSESSIONUSER berechtigt werden, bevor er die Sitzungsberechtigungs-ID festlegen kann.

  Bei der Migration einer vorhandenen Datenbank der Version 8 auf ein Datenbanksystem von DB2 Version 9.1 (oder höhere Version) behält ein Benutzer mit expliziter Berechtigung DBADM (z. B. in SYSCAT.DBAUTH erteilt) die Möglichkeit, die Sitzungsberechtigungs-ID auf einen beliebigen Datenbankbenutzer zu setzen. Dies wird zugelassen, damit vorhandene Anwendungen auch weiterhin funktionieren. Die Fähigkeit, die Sitzungsberechtigung festzulegen, bietet potenziell die Möglichkeit, auf alle geschützten Daten zuzugreifen. Zur Implementierung einer restriktiveren Sicherheitsregelung können Sie diese Einstellung überschreiben, indem Sie die SQL-Anweisung REVOKE SETSESSIONUSER ausführen.
- **Überwachung von Anweisungen und Deadlocks:** Im Rahmen der Aktivitäten zur Deadlocküberwachung von DB2-Datenbankmanagementsystemen werden Werte mit Parametermarken in die Überwachungsausgabe geschrieben, wenn die Klausel WITH VALUES angegeben wird. Ein Benutzer mit Zugriff auf die Überwachungsausgabe kann möglicherweise Zugriff auf Informationen erhalten, für die er nicht berechtigt ist.
- **Traces:** Ein Trace kann Tabellendaten enthalten. Ein Benutzer mit Zugriff auf einen solchen Trace kann Zugriff auf Informationen erhalten, für die er möglicherweise nicht berechtigt ist.
- **Speicherauszugsdateien:** Als Debughilfe für bestimmte Probleme generieren DB2-Datenbankprodukte möglicherweise Speicherauszugsdateien im Verzeichnis `sqllib\db2dump`. Diese Speicherauszugsdateien können Tabellendaten enthalten. Wenn dies der Fall ist, können Benutzer mit Zugriff auf die Dateien Zugriff auf Informationen erhalten, für die sie möglicherweise nicht berechtigt sind. Zur besseren Sicherheit sollten Sie den Zugriff auf das Verzeichnis `sqllib\db2dump` einschränken.
- **Tool 'db2dart':** Das Tool 'db2dart' untersucht eine Datenbank und meldet alle architekturbezogenen Fehler, die es feststellt. Das Tool kann auf Tabellendaten zugreifen, und DB2 schränkt diesen Zugriff durch keinerlei Zugriffssteuerung ein. Ein Benutzer mit der Berechtigung zur Ausführung des Tools 'db2dart' bzw. mit Zugriff auf die Ausgabe von 'db2dart' kann Zugriff auf Informationen erhalten, für die er möglicherweise nicht berechtigt ist.
- **Bindeoption REOPT:** Wenn die Bindeoption REOPT angegeben ist, werden die EXPLAIN-Momentaufnahmeninformationen für jede SQL-Anweisung zum reoptimierbaren inkrementellen Binden bei der Ausführung in die EXPLAIN-Tabellen eingefügt. Die EXPLAIN-Informationen zeigen auch Eingabedatenwerte.
- **Tool 'db2cat':** Das Tool 'db2cat' dient zum Extrahieren des gepackten Deskriptors einer Tabelle. Der gepackte Deskriptor einer Tabelle enthält Statistiken, die Informationen zum Inhalt der Tabelle enthüllen können. Ein Benutzer, der das Tool 'db2cat' ausführt oder Zugriff auf die Ausgabe hat, kann Zugriff auf Informationen erhalten, für die er möglicherweise nicht berechtigt ist.

Beim Erstellen einer Datenbank erteilte Standardzugriffsrechte

Die folgende Liste enthält die Standardzugriffsrechte für bestimmte Systemtabellen, die bei der Erstellung einer Datenbank erteilt werden:

1. SYSIBM.SYSDBAUTH
   - Dem Datenbankersteller werden die folgenden Zugriffsrechte erteilt:
     - DBADM
     - CREATETAB
     - CREATEROLE
     - BINDADD
     - CONNECT
     - NOFENCE
     - IMPLSCHEMA
     - LOAD
     - EXTERNALROUTINE
     - QUIESCECONNECT
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - CREATETAB
     - BINDADD
     - CONNECT
     - IMPLSCHEMA
2. SYSIBM.SYSTABAUTH
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - SELECT für alle SYSCAT- und SYSIBM-Tabellen
     - SELECT und UPDATE für alle SYSSTAT-Tabellen
3. SYSIBM.SYSROUTINEAUTH
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - EXECUTE with GRANT für alle Prozeduren im Schema
     - SQLJ EXECUTE with GRANT für alle Funktionen und Prozeduren im Schema SYSFUN
     - EXECUTE with GRANT für alle Funktionen und Prozeduren im Schema SYSPROC
     - EXECUTE für alle Tabellenfunktionen im Schema SYSIBM
     - EXECUTE für alle anderen Prozeduren im Schema SYSIBM
4. SYSIBM.SYSPACKAGEAUTH
   - Dem Datenbankersteller werden die folgenden Zugriffsrechte erteilt:
     - CONTROL für alle im Schema NULLID erstellten Pakete
     - BIND with GRANT für alle im Schema NULLID erstellten Pakete
     - EXECUTE with GRANT für alle im Schema NULLID erstellten Pakete
     -
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - BIND für alle im Schema NULLID erstellten Pakete
     - EXECUTE für alle im Schema NULLID erstellten Pakete

5. SYSIBM.SCHEMAAUTH
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - CREATEIN im Schema SQLJ
     - CREATEIN im Schema NULLID
6. SYSIBM.TBSPACEAUTH
   - Der speziellen Gruppe PUBLIC werden die folgenden Zugriffsrechte erteilt:
     - USE für den Tabellenbereich USERSPACE1

Kapitel 6. Firewallunterstützung

Eine *Firewall* ist eine Gruppe zusammengehöriger Programme, die sich auf einem Gateway-Netzwerkserver befinden und dazu dienen, unbefugten Zugriff auf ein System oder ein Netzwerk zu verhindern.

Es gibt vier Arten von Firewalls:

1. Paketfilterungsfirewalls, Screening Router Firewalls oder Firewalls der Netzwerkebene
2. Klassische Proxy-Firewalls der Anwendungsebene
3. Transparente Proxy- oder Circuit-Level-Firewalls
4. SMLI-Firewalls (Stateful Multi-Layer Inspection)

Es sind Firewallprodukte verfügbar, die eine der oben genannten Firewallarten implementieren. Viele andere Firewallprodukte implementieren jedoch eine Kombination aus diesen Arten.

Screening-Router-Firewalls

Die Screening-Router-Firewall wird auch als Netzwerk- oder Paketfilterungsfirewall bezeichnet. Die Funktionsweise einer solchen Firewall beruht auf der Überwachung (Screening) ankommender Datenpakete und der Prüfung von Protokollattributen. Zu den getesteten Protokollattributen gehören die Quellen- und Zieladresse, der Typ des Protokolls, der Quellen- und Zielport sowie einige andere protokollspezifische Attribute.

Für alle Firewall-Lösungen (außer SOCKS) müssen Sie sicherstellen, dass alle von der DB2-Datenbank verwendeten Ports für ankommende und abgehende Pakete geöffnet sind. Die DB2-Datenbank verwendet den Port 523 für den DB2-Verwaltungsserver (DAS), der von den DB2-Datenbanktools verwendet wird. Bestimmen Sie die Ports, die von allen Ihren Serverinstanzen verwendet werden, indem Sie mithilfe der Datei 'services' den Servicenamen in der Konfigurationsdatei des Datenbankmanagers auf dem Server mit seiner Portnummer abgleichen.

Proxy-Firewalls der Anwendungsebene

Ein Proxy bzw. ein Proxy-Server ist eine Technik, die als Vermittlungsstelle zwischen einem Web-Client und einem Web-Server fungiert. Eine Proxy-Firewall fungiert als Gateway für Anforderungen, die von Clients ankommen.

Wenn Clientanforderungen durch die Firewall empfangen werden, wird die endgültige Zieladresse durch die Proxysoftware ermittelt. Der Anwendungsproxy übersetzt die Adresse, führt weitere Prüfungen zur Zugriffssteuerung und Protokollfunktionen nach Bedarf aus und stellt die Verbindung zum Server für den Client her.

Das Produkt DB2 Connect auf einer Firewallmaschine kann als Proxy zum Zielserver eingesetzt werden. Darüber hinaus funktioniert ein DB2-Datenbankserver auf der Firewallmaschine, der als Hop-Server zum endgültigen Zielserver eingesetzt wird, wie ein Anwendungsproxy.

Circuit-Level-Firewalls

Die Circuit-Level-Firewall wird auch als transparente Proxy-Firewall bezeichnet.

Eine transparente Proxy-Firewall modifiziert die Anforderung bzw. die Antwort nicht über das hinaus, was für die Proxy-Authentifizierung und -Identifikation erforderlich ist. Ein Bespiel für eine transparente Proxy-Firewall ist SOCKS.

Das DB2-Datenbanksystem unterstützt SOCKS Version 4.

SMLI-Firewalls (Stateful Multi-Layer Inspection)

Die SMLI-Firewall bietet eine fortgeschrittene Form der Paketfilterung, bei der alle sieben Schichten des OSI-Modells (Open System Interconnection) untersucht werden.

Jedes Paket wird untersucht und mit bekannten Status freundlicher Pakete verglichen. Während Screening-Router-Firewalls nur den Paketheader untersuchen, prüfen SMLI-Firewalls das gesamte Paket einschließlich der Daten.

Kapitel 7. Sicherheits-Plug-ins

Die Authentifizierung für das DB2-Datenbanksystem erfolgt mithilfe von *Sicherheits-Plug-ins*. Ein Sicherheits-Plug-in ist eine dynamisch ladbare Bibliothek, die Authentifizierungssicherheitsservices bereitstellt.

Das DB2-Datenbanksystem stellt die folgenden Typen von Plug-ins bereit:

- Plug-in zum Abrufen von Gruppen: Ruft Informationen zur Gruppenzugehörigkeit für einen bestimmten Benutzer ab.
- Plug-in zur Clientauthentifizierung: Verwaltet die Authentifizierung auf einem DB2-Client.
- Plug-in zur Serverauthentifizierung: Verwaltet die Authentifizierung auf einem DB2-Server.

DB2 unterstützt zwei Mechanismen zur Authentifizierung durch Plug-ins:

Benutzer-ID/Kennwort-Authentifizierung
Dieses Authentifizierungsverfahren arbeitet mit einer Benutzer-ID und zugehörigem Kennwort. Die folgenden Authentifizierungstypen sind mithilfe von Plug-ins zur Authentifizierung mit Benutzer-ID und Kennwort implementiert:

- CLIENT
- SERVER
- SERVER_ENCRYPT
- DATA_ENCRYPT
- DATA_ENCRYPT_CMP

Diese Authentifizierungstypen bestimmen, wie und wo die Authentifizierung eines Benutzers erfolgt. Der verwendete Authentifizierungstyp hängt von dem Authentifizierungstyp ab, der durch den Konfigurationsparameter des Datenbankmanagers *authentication* angegeben wird. Wenn der Parameter SRVCON_AUTH angegeben wird, hat er bei der Verarbeitung von CONNECT- und ATTACH-Operationen Vorrang vor AUTHENTICATION.

GSS-API-Authentifizierung
GSS-API ist formal als *Generic Security Service Application Program Interface Version 2* (IETF RFC2743) und *Generic Security Service API Version 2: C-Bindings* (IETF RFC2744) bekannt. Die Kerberos-Authentifizierung wird ebenfalls mithilfe von GSS-API implementiert. Die folgenden Authentifizierungstypen werden mithilfe der Plug-ins der GSS-API-Authentifizierung implementiert:

- KERBEROS
- GSSPLUGIN
- KRB_SERVER_ENCRYPT
- GSS_SERVER_ENCRYPT

KRB_SERVER_ENCRYPT und GSS_SERVER_ENCRYPT unterstützen sowohl die GSS-API-Authentifizierung als auch die Benutzer-ID/Kennwort-Authentifizierung. Allerdings wird der GSS-API-Authentifizierungstyp bevorzugt.

Anmerkung: Authentifizierungstypen bestimmen, wie und wo ein Benutzer authentifiziert wird. Zur Verwendung eines bestimmten Authentifizierungstyps aktualisieren Sie den Konfigurationsparameter 'authentication' des Datenbankmanagers.

Jedes der Plug-ins kann unabhängig oder in Verbindung mit einem oder mehreren der anderen Plug-ins verwendet werden. Sie können zum Beispiel nur ein Plug-in zur Serverauthentifizierung verwenden und die DB2-Standards für die Client- und die Gruppenauthentifizierung voraussetzen. Alternativ könnten Sie auch nur ein Plug-in für die Gruppen- oder Clientauthentifizierung haben. Der einzige Fall, in dem sowohl ein Client-Plug-in als auch ein Server-Plug-in erforderlich sind, betrifft die Plug-ins für die GSS-API-Authentifizierung.

Das Standardverfahren ist die Verwendung eines Benutzer-ID/Kennwort-Plug-ins, das ein Authentifizierungsverfahren auf Betriebssystemebene implementiert. In früheren Releases bestand das Standardverfahren darin, direkt die Authentifizierung der Betriebssystemebene ohne Implementierung eines Plug-ins zu nutzen. Die clientseitige Kerberos-Unterstützung ist unter den Betriebssystemen Solaris, AIX, Windows und Linux verfügbar. Auf Windows-Plattformen ist die Kerberos-Unterstützung standardmäßig aktiviert.

DB2-Datenbanksysteme enthalten jeweils Plug-ins zum Abrufen von Gruppen, zur Benutzer-ID/Kennwort-Authentifizierung und zur Kerberos-Authentifizierung. Aufgrund der Sicherheits-Plug-in-Architektur können Sie das Verhalten für die Client- und Serverauthentifizierung von DB2 anpassen, indem Sie entweder eigene Plug-ins entwickeln oder Plug-ins von einem Fremdanbieter erwerben.

Implementierung von Sicherheits-Plug-ins auf DB2-Clients

DB2-Clients können ein Plug-in zum Abrufen von Gruppen und ein Plug-in zur Benutzer-ID/Kennwort-Authentifizierung unterstützen sowie mit dem DB2-Server ein bestimmtes GSS-API-Plug-in vereinbaren. Diese Vereinbarung wird getroffen, indem der Client die Liste der implementierten GSS-API-Plug-ins des DB2-Servers nach dem Namen des ersten Authentifizierungs-Plug-ins durchsucht, das einem auf dem Client implementierten Authentifizierungs-Plug-in entspricht. Die Liste der Plug-ins des Servers wird durch den Wert des Konfigurationsmanagers *srvcon_gssplugin_list* des Datenbankmanagers angegeben, der die Namen aller Plug-ins enthält, die auf dem Server implementiert sind. Die folgende Abbildung veranschaulicht die Plug-in-Infrastruktur auf einem DB2-Client.

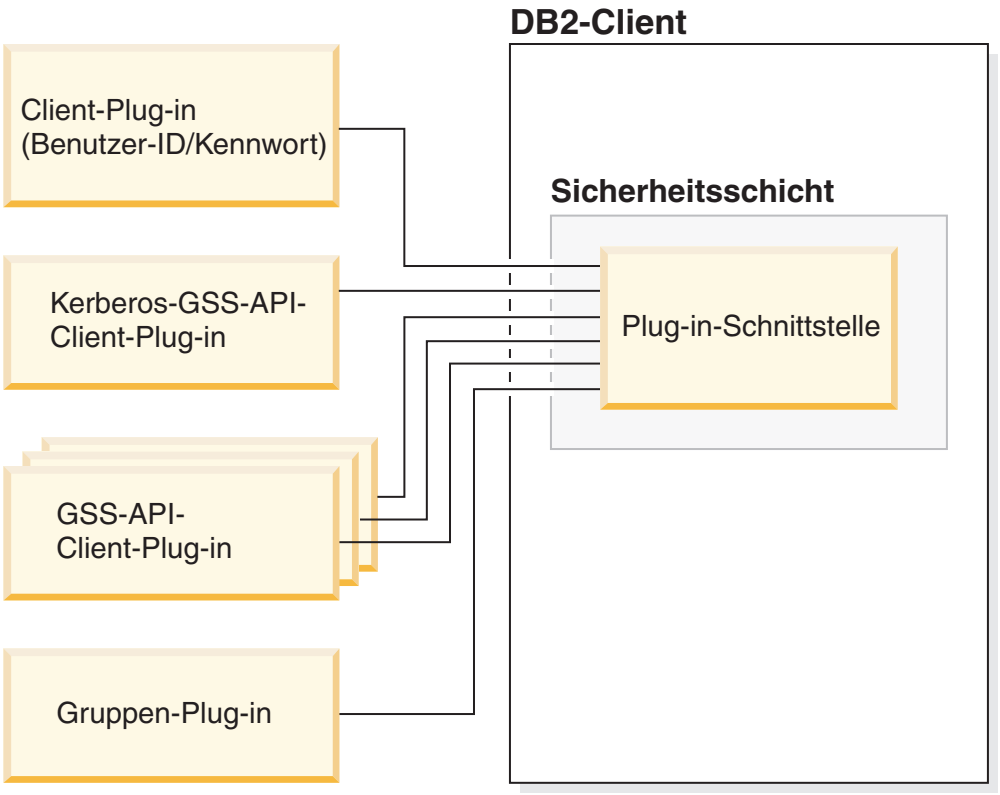


Abbildung 3. Implementierung von Sicherheits-Plug-ins auf DB2-Clients

Implementierung von Sicherheits-Plug-ins auf DB2-Servern

DB2-Server können ein Plug-in zum Abrufen von Gruppen, ein Plug-in zur Benutzer-ID/Kennwort-Authentifizierung und mehrere GSS-API-Plug-ins unterstützen. Die verschiedenen GSS-API-Plug-ins werden im Wert des Konfigurationsparameters *srvcon_gssplugin_list* des Datenbankmanagers in Form einer Liste angegeben. Nur ein GSS-API-Plug-in in dieser Liste darf ein Kerberos-Plug-in sein.

Neben den serverseitigen Sicherheits-Plug-ins müssen Sie möglicherweise auch Plug-ins für die Clientauthentifizierung auf Ihrem Datenbankserver implementieren. Wenn Sie Operationen auf Instanzebene wie db2start und db2trc ausführen, führt der DB2-Datenbankmanager Berechtigungsprüfungen für diese Operationen mithilfe der Plug-ins zur Clientauthentifizierung durch. Aus diesem Grund sollten Sie das Clientauthentifizierungs-Plug-in installieren, das dem Server-Plug-in entspricht, das durch den Konfigurationsparameter *authentication* des Datenbankmanagers angegeben wird. Es gibt einen Hauptunterschied zwischen *authentication* und *srvcon_auth*. Insbesondere können sie auf verschiedene Werte gesetzt werden, um zu veranlassen, dass der eine Mechanismus zur Authentifizierung von Datenbankverbindungen und der andere Mechanismus zur lokalen Authentifizierung verwendet wird. Das gängigste Verfahren besteht darin, den Parameter *srvcon_auth* auf den Wert GSSPLUGIN und den Parameter *authentication* auf den Wert SERVER zu setzen. Wenn Sie keine Plug-ins zur Clientauthentifizierung auf dem Datenbankserver verwenden, schlagen Operationen auf Instanzebene wie zum Beispiel db2start fehl. Wenn der Authentifizierungstyp zum Beispiel SERVER ist und kein benutzerdefiniertes Client-Plug-in verwendet wird, verwendet das DB2-Datenbanksystem das von IBM gelieferte Standard-Plug-in für die Clientauthentifizierung auf Betriebssystemebene.

Die folgende Abbildung veranschaulicht die Plug-in-Infrastruktur auf einem DB2-Server.

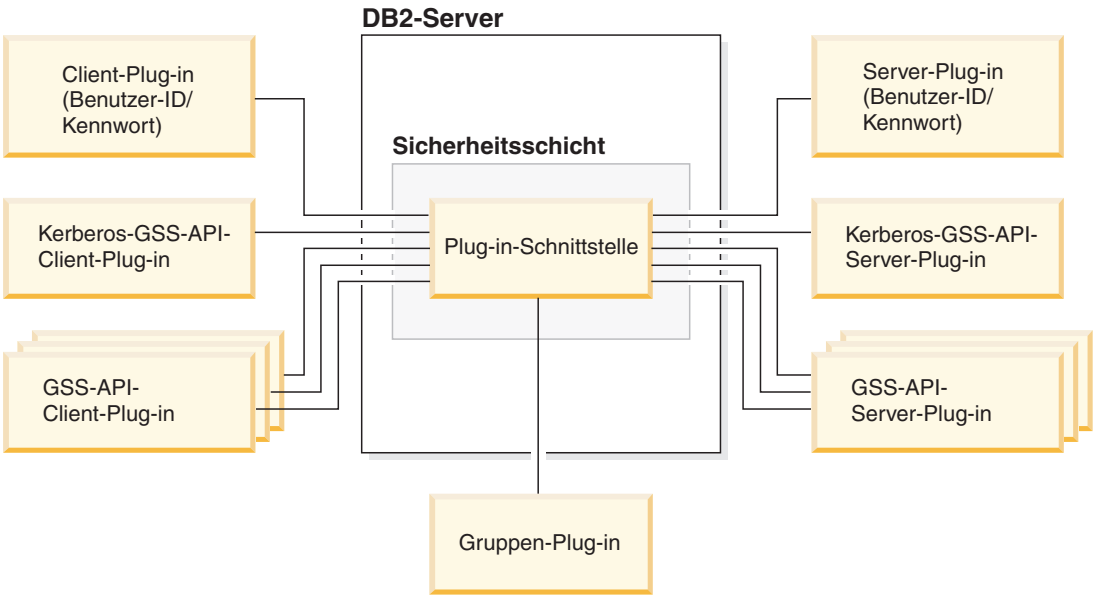


Abbildung 4. Implementierung von Sicherheits-Plug-ins auf DB2-Servern

Anmerkung: Die Integrität der Installation Ihres DB2-Datenbanksystems kann beeinträchtigt werden, wenn die Implementierung von Sicherheits-Plug-ins nicht angemessen codiert, überprüft und getestet wird. Das DB2-Datenbanksystem enthält Vorkehrungen gegen zahlreiche gängige Typen von Fehlern, kann jedoch keine vollständige Integrität garantieren, wenn von Benutzern geschriebene Plug-ins implementiert werden.

Aktivieren von Sicherheits-Plug-ins

Der Systemadministrator kann die Namen der für die einzelnen Authentifizierungsverfahren zu verwendenden Plug-ins angeben, indem er bestimmte, für Plug-ins relevante Konfigurationsparameter des Datenbankmanagers aktualisiert. Wenn diese Parameter den Wert NULL haben, werden standardmäßig die von DB2 bereitgestellten Plug-ins zum Abrufen von Gruppen, zum Benutzer-ID/Kennwort-Management oder für Kerberos (falls der Parameter 'authentication' auf dem Server auf 'Kerberos' gesetzt ist) verwendet. DB2 stellt kein Standard-GSS-API-Plug-in zur Verfügung. Das bedeutet, dass Systemadministratoren, wenn sie den Authentifizierungstyp GSSPLUGIN im Parameter *authentication* angeben, auch ein GSS-API-Authentifizierungs-Plug-in im Parameter *srvcon_gssplugin_list* angeben müssen.

Laden von Sicherheits-Plug-ins in DB2

Alle unterstützten Plug-ins, die durch die Konfigurationsparameter des Datenbankmanagers angegeben werden, werden geladen, wenn der Datenbankmanager gestartet wird.

Der DB2-Client lädt ein Plug-in, das dem mit dem Server bei CONNECT- oder ATTACH-Operationen vereinbarten Sicherheitsmechanismus entspricht. Es ist möglich, dass eine Clientanwendung veranlasst, dass mehrere Sicherheits-Plug-ins gleichzeitig geladen und verwendet werden. Diese Situation kann zum Beispiel bei

einem Multithread-Programm auftreten, das gleichzeitige Verbindungen zu verschiedenen Datenbanken aus verschiedenen Instanzen unterhält.

Für andere Aktionen als CONNECT- oder ATTACH-Operationen (z. B. Aktualisieren der Datenbankmanagerkonfiguration, Starten und Stoppen des Datenbankmanagers, Aktivieren und Inaktivieren der DB2-Tracefunktion) ist ebenfalls eine Berechtigung erforderlich. Für solche Aktionen lädt das DB2-Clientprogramm ein Plug-in, das in einem anderen Konfigurationsparameter des Datenbankmanagers angegeben ist. Wenn der Konfigurationsparameter *authentication* auf den Wert GSSPLUGIN gesetzt ist, verwendet der DB2-Datenbankmanager das Plug-in, das im Konfigurationsparameter *local_gssplugin* angegeben ist. Wenn der Konfigurationsparameter *authentication* auf den Wert KERBEROS gesetzt ist, verwendet der DB2-Datenbankmanager das Plug-in, das im Konfigurationsparameter *clnt_krb_plugin* angegeben ist. Ansonsten verwendet der DB2-Datenbankmanager das Plug-in, das im Konfigurationsparameter *clnt_pw_plugin* angegeben ist.

APIs von Sicherheits-Plug-ins können von einer IPv4-Plattform und von einer IPv6-Plattform aus aufgerufen werden. Eine IPv4-Adresse ist eine 32-Bit-Adresse der lesbaren Form a.b.c.d, bei der jeder Buchstabe von a bis d eine Dezimalzahl zwischen 0 und 255 darstellt. Eine IPv6-Adresse ist eine 128-Bit-Adresse der Form a:b:c:d:e:f:g:h, wobei die Buchstaben von a bis h jeweils 4 Hexadezimalstellen darstellen.

Entwickeln von Sicherheits-Plug-ins

Wenn Sie ein Sicherheits-Plug-in entwickeln, müssen Sie die Standardauthentifizierungsfunktionen implementieren, die der DB2-Datenbankmanager verwenden soll. Wenn Sie ein eigenes, angepasstes Sicherheits-Plug-in verwenden, können Sie eine Benutzer-ID mit einer maximalen Länge von bis zu 255 Zeichen in einer CONNECT-Anweisung verwenden, die über den Befehlszeilenprozessor (CLP) oder durch eine dynamische SQL-Anweisung abgesetzt wird. Für die verfügbaren Typen von Plug-ins müssen Sie die folgende Funktionalität implementieren:

Abrufen von Gruppen
Ruft eine Liste der Gruppen ab, zu denen ein Benutzer gehört.

Benutzer-ID/Kennwort-Authentifizierung
- Identifiziert den Standardsicherheitskontext (nur Client).
- Prüft und ändert (optional) ein Kennwort.
- Stellt fest, ob eine angegebene Zeichenfolge einen gültigen Benutzer darstellt (nur Server).
- Ändert die Benutzer-ID oder das Kennwort, die bzw. das auf dem Client angegeben wurde, bevor sie/es an den Server gesendet wird (nur Client).
- Gibt die DB2-Berechtigungs-ID zurück, die einem angegebenen Benutzer zugeordnet ist.

GSS-API-Authentifizierung
- Implementiert die erforderlichen GSS-API-Funktionen.
- Identifiziert den Standardsicherheitskontext (nur Client).
- Generiert die Anfangsberechtigungsnachweise auf der Basis von Benutzer-ID und Kennwort und ändert (optional) das Kennwort (nur Client).
- Erstellt und akzeptiert Sicherheitstickets.
- Gibt die DB2-Berechtigungs-ID zurück, die einem angegebenen GSS-API-Sicherheitskontext zugeordnet ist.

Speicherpositionen für Sicherheits-Plug-in-Bibliotheken

Wenn Sie Sicherheits-Plug-ins (entweder durch Eigenentwicklung oder durch Kauf bei einem Fremdanbieter) erwerben, kopieren Sie sie an bestimmte Speicherpositionen auf Ihrem Datenbankserver.

DB2-Clients suchen clientseitige Plug-ins zur Benutzerauthentifizierung im folgenden Verzeichnis:

- UNIX (32 Bit): `$DB2PATH/security32/plugin/client`
- UNIX (64 Bit): `$DB2PATH/security64/plugin/client`
- WINDOWS (32 und 64 Bit): `$DB2PATH\security\plugin\`*`instanzname`*`\client`

Anmerkung: Auf Windows-basierten Plattformen werden die Unterverzeichnisse *instanzname* und *client* nicht automatisch erstellt. Der Instanzeigner muss diese Verzeichnisse manuell erstellen.

Der DB2-Datenbankmanager sucht serverseitige Plug-ins zur Benutzerauthentifizierung im folgenden Verzeichnis:

- UNIX (32 Bit): `$DB2PATH/security32/plugin/server`
- UNIX (64 Bit): `$DB2PATH/security64/plugin/server`
- WINDOWS (32 und 64 Bit): `$DB2PATH\security\plugin\`*`instanzname`*`\server`

Anmerkung: Auf Windows-basierten Plattformen werden die Unterverzeichnisse *instanzname* und *server* nicht automatisch erstellt. Der Instanzeigner muss diese Verzeichnisse manuell erstellen.

Der DB2-Datenbankmanager sucht nach Plug-ins für Gruppen im folgenden Verzeichnis:

- UNIX (32 Bit): `$DB2PATH/security32/plugin/group`
- UNIX (64 Bit): `$DB2PATH/security64/plugin/group`
- WINDOWS (32 und 64 Bit): `$DB2PATH\security\plugin\`*`instanzname`*`\group`

Anmerkung: Auf Windows-basierten Plattformen werden die Unterverzeichnisse *instanzname* und *group* nicht automatisch erstellt. Der Instanzeigner muss diese Verzeichnisse manuell erstellen.

Namenskonventionen für Sicherheits-Plug-ins

Sicherheits-Plug-in-Bibliotheken müssen eine plattformspezifische Dateinamenerweiterung haben. Sicherheits-Plug-in-Bibliotheken, die in C oder C++ geschrieben sind, müssen eine plattformspezifische Dateinamenerweiterung haben:

- Windows: `.dll`.
- AIX: `.a` `oder` `.so`. Falls beide Erweiterungen vorhanden sind, wird die Erweiterung `.a` verwendet.
- Linux, HP IPF und Solaris: `.so`.
- HPUX auf PA-RISC: `.sl` `oder` `.so`. Falls beide Erweiterungen vorhanden sind, wird die Erweiterung `.sl` verwendet.

Anmerkung: Benutzer können Sicherheits-Plug-ins auch mit DB2 Universal JDBC Driver entwickeln.

Nehmen Sie zum Beispiel an, Sie haben eine Sicherheits-Plug-in-Bibliothek mit dem Namen `MyPlugin`. Für die einzelnen unterstützten Betriebssysteme müsste der korrekte Dateiname der Bibliothek wie folgt aussehen:

- Windows (32 Bit): MyPlugin.dll
- Windows (64 Bit): MyPlugin64.dll
- AIX (32 oder 64 Bit): MyPlugin.a oder MyPlugin.so
- SUN (32 oder 64 Bit), Linux (32 oder 64 Bit), HP (32 oder 64 Bit) auf IPF: MyPlugin.so
- HP-UX (32 oder 64 Bit) auf PA-RISC: MyPlugin.sl oder MyPlugin.so

Anmerkung: Das Suffix ″64″ ist nur bei Bibliotheksnamen für Sicherheits-Plug-ins auf 64-Bit-Windows-Plattformen erforderlich.

Wenn Sie die Datenbankmanagerkonfiguration mit dem Namen eines Sicherheits-Plug-ins aktualisieren, verwenden Sie den vollen Namen der Bibliothek ohne das Suffix ″64″ und lassen sowohl die Dateierweiterung als auch den qualifizierten Pfadteil des Namens weg. Unabhängig vom Betriebssystem wird ein Sicherheits-Plug-in mit dem Namen `MyPlugin` wie folgt registriert:

```
UPDATE DBM CFG USING CLNT_PW_PLUGIN MyPlugin
```

Der Sicherheits-Plug-in-Name ist von der Groß-/Kleinschreibung abhängig und muss mit dem Bibliotheksnamen exakt übereinstimmen. DB2-Datenbanksysteme verwenden den Wert aus dem relevanten Konfigurationsparameter des Datenbankmanagers, um den Bibliothekspfad zusammenzusetzten, und verwenden anschließend den Bibliothekspfad, um die Sicherheits-Plug-in-Bibliothek zu laden.

Zur Vermeidung von Plug-in-Namenskonflikten sollten Sie das Plug-in unter Angabe der verwendeten Authentifizierungsmethode und eines kennzeichnenden Symbols für das Unternehmen, von dem das Plug-in geschrieben wurde, benennen. Wenn zum Beispiel das Plug-in, das die Authentifizierungsmethode `einemethode` implementiert, vom Unternehmen `Foo, Inc.` geschrieben wurde, könnte es einen Namen wie `FOOeinemethode.dll` erhalten.

Die maximale Länge eines Plug-in-Namens (ohne Dateierweiterung und das Suffix ″64″) ist auf 32 Byte begrenzt. Es gibt keine maximale Anzahl von Plug-ins, die vom Datenbankserver unterstützt werden, jedoch ist die maximale Länge der durch Kommata getrennten Plug-in-Liste in der Datenbankmanagerkonfiguration auf 255 Byte begrenzt. Zwei Define-Anweisungen in der Kopfdatendatei `sqlenv.h` geben diese beiden Grenzwerte an:

```
#define SQL_PLUGIN_NAME_SZ     32     /* Plug-in-Name */
#define SQL_SRVCON_GSSPLUGIN_LIST_SZ 255 /* GSS-API-Plug-in-Liste */
```

Die Bibliotheksdateien eines Plug-ins müssen die folgenden Dateiberechtigungen besitzen:

- Eigentum des Instanzeigners
- Lesbar durch alle Benutzer auf dem System
- Ausführbar durch alle Benutzer auf dem System

Unterstützung zweiteiliger Benutzer-IDs in Sicherheits-Plug-ins

Der DB2-Datenbankmanager unter Windows unterstützt die Verwendung zweiteiliger Benutzer-IDs sowie die Zuordnung zwischen zweiteiligen Benutzer-IDs und zweiteiligen Berechtigungs-IDs.

Betrachten Sie zum Beispiel eine zweiteilige Benutzer-ID unter dem Windows-Betriebssystem, die sich aus einer Domäne und einer Benutzer-ID zusammensetzt: `MEDWAY\pieter`. In diesem Beispiel ist `MEDWAY` eine Domäne und `pieter` der Benutzername. In DB2-Datenbanksystemen können Sie angeben, ob diese zweiteilige Benutzer-ID einer einteiligen oder einer zweiteiligen Berechtigungs-ID zugeordnet werden soll.

Die Zuordnung einer zweiteiligen Benutzer-ID zu einer zweiteiligen Berechtigungs-ID wird unterstützt, ist jedoch nicht das Standardverhalten. Standardmäßig werden sowohl einteilige Benutzer-IDs als auch zweiteilige Benutzer-IDs einteiligen Berechtigungs-IDs zugeordnet.

Die Standardzuordnung einer zweiteiligen Benutzer-ID zu einer einteiligen Benutzer-ID ermöglicht einem Benutzer, eine Verbindung zu der Datenbank mit folgenden Befehl herzustellen:

```
db2 connect to db user MEDWAY\pieter using pw
```

Beim Standardverhalten wird die Benutzer-ID `MEDWAY\pieter` in diesem Fall in die Berechtigungs-ID `PIETER` aufgelöst. Wenn die Unterstützung für die Zuordnung einer zweiteiligen Benutzer-ID zu einer zweiteiligen Berechtigungs-ID aktiviert würde, hieße die Berechtigungs-ID `MEDWAY\PIETER`.

Zur Aktivierung der Zuordnung zweiteiliger Benutzer-IDs zu zweiteiligen Berechtigungs-IDs in DB2 stellt DB2 zwei Gruppen von Authentifizierungs-Plug-ins zur Verfügung:

- Bei der einen Gruppe werden einteilige Benutzer-IDs und zweiteilige Benutzer-IDs ausschließlich einteiligen Berechtigungs-IDs zuordnet.
- Bei der anderen Gruppe werden einteilige Benutzer-IDs und zweiteilige Benutzer-IDs zweiteiligen Berechtigungs-IDs zugeordnet.

Wenn sich ein Benutzername in Ihrer Arbeitsumgebung mehreren Konten zuordnen lässt, die an verschiedenen Positionen (z. B. als lokales Konto, Domänenkonto und vertraute Domänenkonten) definiert sind, können Sie das Plug-in angeben, das die Zuordnung zweiteiliger Berechtigungs-IDs ermöglicht.

Es wichtig zu beachten, dass eine einteilige Berechtigungs-ID wie zum Beispiel `PIETER` und eine zweiteilige Berechtigungs-ID, die eine Domäne und eine Benutzer-ID wie zum Beispiel `MEDWAY\pieter` kombiniert, funktionell verschiedene Berechtigungs-IDs sind. Die Gruppe der Zugriffsrechte, die einer dieser Berechtigungs-IDs erteilt ist, kann sich von der Gruppe der Zugriffsrechte, die der anderen Berechtigungs-ID erteilt sind, völlig unterscheiden. Daher muss mit einteiligen und zweiteiligen Berechtigungs-IDs sorgfältig gearbeitet werden.

In der folgenden Tabelle sind die Arten der von DB2-Datenbanksystemen bereitgestellten Plug-ins sowie die Plug-in-Namen für die jeweiligen Authentifizierungsimplementierungen aufgeführt.

Tabelle 29. DB2-Sicherheits-Plug-ins

Authentifizierungstyp	Name des Plug-ins für einteilige Benutzer-IDs	Name des Plug-ins für zweiteilige Benutzer-IDs
Benutzer-ID/Kennwort (Client)	IBMOSauthclient	IBMOSauthclientTwoPart
Benutzer-ID/Kennwort (Server)	IBMOSauthserver	IBMOSauthserverTwoPart

Tabelle 29. DB2-Sicherheits-Plug-ins (Forts.)

Authentifizierungstyp	Name des Plug-ins für einteilige Benutzer-IDs	Name des Plug-ins für zweiteilige Benutzer-IDs
Kerberos	IBMkrb5	IBMkrb5TwoPart

Anmerkung: Auf 64-Bit-Windows-Plattformen werden die Zeichen ″64″ an die hier aufgeführten Plug-in-Namen angehängt.

Wenn Sie einen Authentifizierungstyp angeben, der ein Benutzer-ID/Kennwort- oder Kerberos-Plug-in erfordert, werden standardmäßig die Plug-ins verwendet, die in der Spalte ″Name des Plug-ins für einteilige Benutzer-IDs″ der obigen Tabelle aufgeführt sind.

Wenn zweiteilige Benutzer-IDs zweiteiligen Berechtigungs-IDs zugeordnet werden sollen, müssen Sie angeben, dass das Plug-in für zweiteilige Benutzer-IDs, das nicht das Standard-Plug-in ist, verwendet wird. Sicherheits-Plug-ins werden auf der Instanzebene angegeben, indem die sicherheitsbezogenen Konfigurationsparameter des Datenbankmanagers wie folgt eingestellt werden:

Für die Serverauthentifizierung, bei der zweiteilige Benutzer-IDs zweiteiligen Berechtigungs-IDs zugeordnet werden, müssen Sie die folgenden Parametereinstellungen definieren:

- `srvcon_pw_plugin` mit dem Wert `IBMOSauthserverTwoPart`
- `clnt_pw_plugin` mit dem Wert `IBMOSauthclientTwoPart`

Für die Clientauthentifizierung, bei der zweiteilige Benutzer-IDs zweiteiligen Berechtigungs-IDs zugeordnet werden, müssen Sie die folgenden Parametereinstellungen definieren:

- `srvcon_pw_plugin` mit dem Wert `IBMOSauthserverTwoPart`
- `clnt_pw_plugin` mit dem Wert `IBMOSauthclientTwoPart`

Für die Kerberos-Authentifizierung, bei der zweiteilige Benutzer-IDs zweiteiligen Berechtigungs-IDs zugeordnet werden, müssen Sie die folgenden Parametereinstellungen definieren:

- `srvcon_gssplugin_list` mit dem Wert `IBMOSkrb5TwoPart`
- `clnt_krb_plugin` mit dem Wert `IBMkrb5TwoPart`

Die Bibliotheken der Sicherheits-Plug-ins akzeptieren zweiteilige Benutzer-IDs, die in einem mit der Microsoft Windows-SAM-Datenbank (Security Account Manager) kompatiblen Format angegeben werden, wie zum Beispiel das Format: *domäne**benutzer-id*. Sowohl der Domänenname als auch die Benutzer-ID werden von der DB2-Authentifizierung und den Authentifizierungsprozessen beim Verbindungsaufbau verwendet.

Sie sollten in Betracht ziehen, die Plug-ins mit zweiteiligen Berechtigungs-IDs zu implementieren, wenn Sie neue Datenbanken erstellen, um Konflikte mit einteiligen Berechtigungs-IDs in vorhandenen Datenbanken zu vermeiden. Neue Datenbanken, die zweiteilige Berechtigungs-IDs verwenden, müssen in einer Instanz erstellt werden, die von den Datenbanken mit einteiligen Berechtigungs-IDs getrennt ist.

Versionssteuerung für Sicherheits-Plug-in-APIs

Das DB2-Datenbanksystem unterstützt eine Versionsnummerierung der Sicherheits-Plug-in-APIs. Solche Versionsnummern sind ganzzahlige Werte, die bei 1 für DB2 UDB Version 8.2 beginnen.

Die Versionsnummer, die DB2 an die Sicherheits-Plug-in-APIs übergibt, ist die höchste Versionsnummer der API, die DB2 unterstützen kann. Sie entspricht der Versionsnummer der Struktur. Wenn das Plug-in eine höhere API-Version unterstützen kann, muss sie Funktionszeiger für die Version zurückgeben, die DB2 angefordert hat. Wenn das Plug-in nur eine niedrigere Version der API unterstützt, sollte das Plug-in die Funktionszeiger für die niedrigere Version einfügen. In beiden Fällen sollten die Sicherheits-Plug-in-APIs die unterstützte Versionsnummer für die API im Feld 'version' der Funktionsstruktur zurückgeben.

Für DB2 ändern sich die Versionsnummern von Sicherheits-Plug-ins nur, wenn dies erforderlich ist (z. B. wenn es Änderungen an den Parametern der APIs gibt). Versionsnummern werden nicht automatisch mit DB2-Releasenummern geändert.

Hinweise zu 32- und 64-Bit-Sicherheits-Plug-ins

Im Allgemeinen verwendet eine 32-Bit-DB2-Instanz das 32-Bit-Sicherheits-Plug-in und eine 64-Bit-DB2-Instanz das 64-Bit-Sicherheits-Plug-in. Allerdings unterstützt DB2 in einer 64-Bit-Instanz 32-Bit-Anwendungen, die eine 32-Bit-Plug-in-Bibliothek erfordern.

Eine Datenbankinstanz, in der sowohl 32-Bit-Anwendungen als auch 64-Bit-Anwendungen ausgeführt werden können, wird als Hybridinstanz bezeichnet. Wenn Sie eine Hybridinstanz haben und 32-Bit-Anwendungen ausführen wollen, stellen Sie sicher, dass die erforderlichen 32-Bit-Sicherheits-Plug-ins im Verzeichnis für 32-Bit-Plug-ins verfügbar sind. In 64-Bit-DB2-Instanzen unter Linux- und UNIX-Betriebssystemen (ausgenommen Linux auf IPF) sind die Verzeichnisse `security32` und `security64` vorhanden. In einer 64-Bit-DB2-Instanz unter Windows auf X64 oder IPF befinden sich die 32- und die 64-Bit-Sicherheits-Plug-ins im selben Verzeichnis, jedoch sind die Namen der 64-Bit-Plug-ins durch das Suffix ″64″ gekennzeichnet.

Wenn Sie von einer 32-Bit-Instanz auf eine 64-Bit-Instanz migrieren wollen, sollten Sie Versionen Ihrer Sicherheits-Plug-ins erwerben, die für 64-Bit-Systeme rekompiliert sind.

Wenn Sie Ihre Sicherheits-Plug-ins bei einem Anbieter erworben haben, der keine 64-Bit-Plug-in-Bibliotheken zur Verfügung stellt, können Sie ein 64-Bit-Stub implementieren, das eine 32-Bit-Anwendung ausführt. In diesem Fall ist das Sicherheits-Plug-in eher ein externes Programm und keine Bibliothek mehr.

Fehlerbestimmung für Sicherheits-Plug-ins

Probleme mit Sicherheits-Plug-ins werden auf zwei Arten zurückgemeldet: durch SQL-Fehler und durch das Protokoll mit Benachrichtigungen für die Systemverwaltung.

Die folgenden SQLCODE-Werte beziehen sich auf Sicherheits-Plug-ins:

- SQLCODE -1365 wird zurückgegeben, wenn ein Plug-in-Fehler bei der Verarbeitung von db2start oder db2stop auftritt.

- SQLCODE -1366 wird zurückgegeben, wenn ein Problem mit der lokalen Berechtigung vorliegt.
- SQLCODE -30082 wird für alle verbindungsbezogenen Plug-in-Fehler zurückgegeben.

Das Protokoll mit Benachrichtigungen für die Systemverwaltung ist eine gute Ressource für das Debugging und Verwalten von Sicherheits-Plug-ins. Zum Anzeigen des Protokolls mit Benachrichtigungen für die Systemverwaltung unter UNIX prüfen Sie die Datei `sqllib/db2dump/`*`instanzname`*`.nfy`. Zum Anzeigen des Protokolls mit Benachrichtigungen für die Systemverwaltung unter Windows-Betriebssystemen verwenden Sie das Tool 'Ereignisanzeige'. Das Tool 'Ereignisanzeige' finden Sie, indem Sie von der Schaltfläche 'Start' des Windows-Betriebssystems aus die Optionen `Einstellungen -> Systemsteuerung -> Verwaltung -> Ereignisanzeige` auswählen. Die folgenden Werte im Protokoll mit Benachrichtigungen für die Systemverwaltung beziehen sich auf Sicherheits-Plug-ins:

- 13000 gibt an, dass ein Aufruf der API für das GSS-API-Sicherheits-Plug-in mit einem Fehler fehlgeschlagen ist und eine optionale Fehlernachricht zurückgegeben hat.

  ```
  SQLT_ADMIN_GSS_API_ERROR (13000)
  Das Plug-in "plug-in-name" hat den Fehlercode "fehlercode"
  von der GSS-API (Generic Security Service) "name-der-gss-api" mit der
  Fehlernachricht "fehlernachricht" erhalten.
  ```

- 13001 gibt an, dass ein Aufruf der API für das DB2-Sicherheits-Plug-in mit einem Fehler fehlgeschlagen ist und eine optionale Fehlernachricht zurückgegeben hat.

  ```
  SQLT_ADMIN_PLUGIN_API_ERROR(13001)
  Das Plug-in "plug-in-name" hat den Fehlercode "fehlercode"
  von der API des DB2-Sicherheits-Plug-in "name-der-gss-api" mit der
  Fehlernachricht "fehlernachricht" erhalten.
  ```

- 13002 gibt an, dass DB2 ein Plug-in nicht entladen konnte.

  ```
  SQLT_ADMIN_PLUGIN_UNLOAD_ERROR (13002)
  Das Plug-in "plug-in-name" konnte nicht entladen werden.
  Es sind keine weiteren Schritte erforderlich.
  ```

- 13003 weist auf einen falschen Namen des Principals hin.

  ```
  SQLT_ADMIN_INVALID_PRIN_NAME (13003)
  Der für "plug-in-name" verwendete Name des Principals "name-des-principals"
  ist ungültig. Stellen Sie sicher, dass der Name des Principals gültig ist und
  dass er ein Format verwendet, das vom Sicherheits-Plug-in erkannt wird.
  ```

- 13004 gibt an, dass der Plug-in-Name nicht gültig ist. Pfadtrennzeichen (unter UNIX "/", unter Windows "\") sind im Plug-in-Namen nicht zulässig.

  ```
  SQLT_ADMIN_INVALID_PLGN_NAME (13004)
  Der Plug-in-Name "plug-in-name" ist ungültig. Stellen Sie sicher,
  dass ein gültiger Plug-in-Name angegeben wurde.
  ```

- 13005 gibt an, dass das Sicherheits-Plug-in nicht geladen werden konnte. Stellen Sie sicher, dass sich das Plug-in im richtigen Verzeichnis befindet und dass die entsprechenden Konfigurationsparameter des Datenbankmanagers aktualisiert wurden.

  ```
  SQLT_ADMIN_PLUGIN_LOAD_ERROR (13005)
  Das Plug-in "plug-in-name" konnte nicht geladen werden. Prüfen Sie, ob das Plug-in
  vorhanden ist und ob die Verzeichnisposition und die Dateiberechtigungen gültig sind.
  ```

- 13006 gibt an, dass von einem Sicherheits-Plug-in ein unerwarteter Fehler festgestellt wurde. Sammeln Sie alle Unterstützungsinformationen (db2support), zeichnen Sie einen Trace auf (db2trc), falls möglich, und rufen Sie anschließend die IBM Unterstützungsfunktion an, um weitere Hilfe zu erhalten.

  ```
  SQLT_ADMIN_PLUGIN_UNEXP_ERROR (13006)
  Das Plug-in "plug-in-name" hat einen unerwarteten Fehler festgestellt.
  Hilfe erhalten Sie bei der IBM Unterstützungsfunktion.
  ```

Anmerkung: Wenn Sie Sicherheits-Plug-ins auf einem 64-Bit-Windows-Datenbankserver verwenden und einen Ladefehler für ein Sicherheits-Plug-in empfangen, lesen Sie den Abschnitt mit den Hinweisen zu 32- und 64-Bit-Sicherheits-Plug-ins sowie den Abschnitt zu den Namenskonventionen für Sicherheits-Plug-ins. Die Plug-in-Bibliothek für die 64-Bit-Plattform erfordert das Suffix ″64″ am Ende des Namens der Bibliothek, während der Eintrag in den Konfigurationsparametern des Datenbankmanagers für Sicherheits-Plug-ins dieses Suffix nicht angeben darf.

Aktivieren von Plug-ins

Implementieren eines Plug-ins zum Abrufen von Gruppen

Zur Anpassung des Verhaltens des DB2-Sicherheitssystems beim Abrufen von Gruppen können Sie ein eigenes Plug-ins zum Abrufen von Gruppen entwickeln oder ein Plug-in bei einem Fremdanbieter erwerben.

Nachdem Sie ein Plug-in zum Abrufen von Gruppen erworben haben, das für Ihr Datenbankverwaltungssystem geeignet ist, können Sie es implementieren.

- Führen Sie die folgenden Schritte aus, um ein Plug-in zum Abrufen von Gruppen auf dem Datenbankserver zu implementieren:
  1. Kopieren Sie die Bibliothek des Plug-ins zum Abrufen von Gruppen in das Gruppen-Plug-in-Verzeichnis des Servers (group).
  2. Aktualisieren Sie den Konfigurationsparameter *group_plugin* des Datenbankmanagers mit dem Namen des Plug-ins.
- Führen Sie die folgenden Schritte aus, um ein Plug-in zum Abrufen von Gruppen auf Datenbankclients zu implementieren:
  1. Kopieren Sie die Bibliothek des Plug-ins zum Abrufen von Gruppen in das Gruppen-Plug-in-Verzeichnis des Clients (group).
  2. Aktualisieren Sie auf dem Datenbankclient den Konfigurationsparameter *group_plugin* des Datenbankmanagers mit dem Namen des Plug-ins.

Implementieren eines Benutzer-ID/Kennwort-Plug-ins

Zur Anpassung des Benutzer-ID/Kennwort-Authentifizierungsverhaltens des DB2-Sicherheitssystems können Sie eigene Plug-ins für die Benutzer-ID/Kennwort-Authentifizierung entwickeln oder ein Plug-in bei einem Fremdanbieter erwerben.

Abhängig von ihrer geplanten Verwendung müssen alle Benutzer-ID/Kennwort-Authentifizierungs-Plug-ins entweder im Client-Plug-in-Verzeichnis oder im Server-Plug-in-Verzeichnis platziert werden. Wenn ein Plug-in im Client-Plug-in-Verzeichnis platziert wird, wird es für die Überprüfung bei der lokalen Authentifizierung sowie zur Prüfung des Clients verwendet, wenn dieser versucht, eine Verbindung zum Server herzustellen. Wenn das Plug-in im Server-Plug-in-Verzeichnis platziert wird, wird es verwendet, um eingehende Verbindungen zum Server zu verarbeiten und zu überprüfen, ob eine Berechtigungs-ID vorhanden und gültig ist, wenn die Anweisung GRANT ohne Angabe des Schlüsselworts USER oder GROUP abgesetzt wird. In den meisten Fällen erfordert die Benutzer-ID/Kennwort-Authentifizierung nur ein serverseitiges Plug-in. Es ist möglich, auch wenn dies allgemein für weniger sinnvoll erachtet wird, nur ein clientseitiges Plug-in zur Benutzer-ID/Kennwort-Authentifizierung zu haben. Es ist möglich, wenn auch recht ungewöhnlich, übereinstimmende Benutzer-ID/Kennwort-Plug-ins auf dem Client und auf dem Server vorauszusetzen.

Anmerkung: Sie müssen den DB2-Server und alle Anwendungen, die die Plug-ins verwenden, stoppen, bevor Sie eine *neue* Version eines *vorhandenen* Plug-ins imple-

mentieren. Undefiniertes Verhalten, einschließlich Traps, kann auftreten, wenn ein Prozess ein Plug-in zu dem Zeitpunkt verwendet, zu dem eine neue Version (mit demselben Namen) über das Plug-in kopiert wird. Diese Einschränkung gilt nicht, wenn Sie ein Plug-in zum ersten Mal implementieren oder wenn das Plug-in nicht im Gebrauch ist.

Nachdem Sie Benutzer-ID/Kennwort-Authentifizierungs-Plug-ins erworben haben, die für Ihr Datenbankverwaltungssystem geeignet sind, können Sie sie implementieren.

- Führen Sie die folgenden Schritte auf dem Datenbankserver aus, um ein Benutzer-ID/Kennwort-Authentifizierungs-Plug-in auf dem Datenbankserver zu implementieren:
  1. Kopieren Sie die Bibliothek des Benutzer-ID/Kennwort-Authentifizierungs-Plug-ins in das Server-Plug-in-Verzeichnis.
  2. Aktualisieren Sie den Konfigurationsparameter *srvcon_pw_plugin* des Datenbankmanagers mit dem Namen des Server-Plug-ins. Dieses Plug-in wird vom Server verwendet, wenn er CONNECT- und ATTACH-Anforderungen verarbeitet.
  3. Anschließend haben Sie zwei Möglichkeiten:
     - Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf den Authentifizierungstyp CLIENT, SERVER, SERVER_ ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP. Alternativ:
     - Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf den Wert NOT_SPECIFIED und den Parameter *authentication* auf den Authentifizierungstyp CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP.
- Führen Sie die folgenden Schritte auf jedem Client aus, um ein Benutzer-ID/Kennwort-Authentifizierungs-Plug-in auf Datenbankclients zu implementieren:
  1. Kopieren Sie die Bibliothek des Benutzer-ID/Kennwort-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis.
  2. Aktualisieren Sie den Konfigurationsparameter *clnt_pw_plugin* des Datenbankmanagers mit dem Namen des Client-Plug-ins. Dieses Plug-in wird unabhängig davon, wo die Authentifizierung stattfindet, geladen und aufgerufen, nicht nur wenn der Datenbankkonfigurationsparameter *authentication* auf den Wert CLIENT gesetzt ist.
- Führen Sie die folgenden Schritte zur Implementierung der Berechtigung auf einem Client, Server oder Gateway durch ein Benutzer-ID/Kennwort-Authentifizierungs-Plug-in auf jedem Client, Server oder Gateway aus:
  1. Kopieren Sie die Bibliothek des Benutzer-ID/Kennwort-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis auf dem Client, Server oder Gateway.
  2. Aktualisieren Sie den Konfigurationsparameter *clnt_pw_plugin* des Datenbankmanagers mit dem Namen des Plug-ins.
  3. Setzen Sie den Konfigurationsparameter *authentication* des Datenbankmanagers auf den Wert CLIENT, SERVER, SERVER_ENCRYPT, DATA_ ENCRYPT oder DATA_ENCRYPT_CMP.

Implementieren eines GSS-API-Plug-ins

Zur Anpassung des Authentifizierungsverhaltens des DB2-Sicherheitssystems können Sie eigene Plug-ins für die Authentifizierung mithilfe der GSS-API entwickeln oder ein Plug-in bei einem Fremdanbieter erwerben.

Bei anderen Plug-in-Typen als Kerberos müssen die Plug-in-Namen auf dem Client und dem Server sowie der Plug-in-Typ übereinstimmen. Die Plug-ins auf dem Client und dem Server brauchen nicht von demselben Anbieter zu stammen, jedoch müssen sie kompatible GSS-API-Token generieren und verarbeiten. Jede Kombination von Kerberos-Plug-ins, die auf dem Client und auf dem Server implementiert sind, ist akzeptabel, da Kerberos-Plug-ins standardisiert sind. Verschiedene Implementierungen weniger standardisierter GSS-API-Mechanismen, wie zum Beispiel *x.509*-Zertifikate, sind jedoch möglicherweise nur teilweise mit DB2-Datenbanksystemen kompatibel. Abhängig von ihrer geplanten Verwendung müssen alle GSS-API-Authentifizierungs-Plug-ins entweder im Client-Plug-in-Verzeichnis oder im Server-Plug-in-Verzeichnis platziert werden. Wenn ein Plug-in im Client-Plug-in-Verzeichnis platziert wird, wird es für die Überprüfung bei der lokalen Authentifizierung sowie bei Versuchen eines Clients, eine Verbindung mit dem Server herzustellen, verwendet. Wenn das Plug-in im Server-Plug-in-Verzeichnis platziert wird, wird es verwendet, um eingehende Verbindungen zum Server zu verarbeiten und zu überprüfen, ob eine Berechtigungs-ID vorhanden und gültig ist, wenn die Anweisung GRANT ohne Angabe des Schlüsselworts USER oder GROUP abgesetzt wird.

Anmerkung: Sie müssen den DB2-Server und alle Anwendungen, die die Plug-ins verwenden, stoppen, bevor Sie eine *neue* Version eines *vorhandenen* Plug-ins implementieren. Undefiniertes Verhalten, einschließlich Traps, kann auftreten, wenn ein Prozess ein Plug-in zu dem Zeitpunkt verwendet, zu dem eine neue Version (mit demselben Namen) über das Plug-in kopiert wird. Diese Einschränkung gilt nicht, wenn Sie ein Plug-in zum ersten Mal implementieren oder wenn das Plug-in nicht im Gebrauch ist.

Nachdem Sie GSS-API-Authentifizierungs-Plug-ins erworben haben, die für Ihr Datenbankverwaltungssystem geeignet sind, können Sie sie implementieren.

- Führen Sie die folgenden Schritte auf dem Server aus, um ein GSS-API-Authentifizierungs-Plug-in auf dem Datenbankserver zu implementieren:
  1. Kopieren Sie die Bibliothek des GSS-API-Authentifizierungs-Plug-ins in das Server-Plug-in-Verzeichnis. Sie können viele GSS-API-Plug-ins in dieses Verzeichnis kopieren.
  2. Aktualisieren Sie den Konfigurationsparameter *srvcon_gssplugin_list* des Datenbankmanagers mit einer geordneten, durch Kommata begrenzten Liste der Namen der im GSS-API-Plug-in-Verzeichnis installierten Plug-ins.
  3. Anschließend haben Sie zwei Möglichkeiten:
     - Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf den Wert GSSPLUGIN oder GSS_SERVER_ENCRYPT, um den Server zur Verwendung der Authentifizierungsmethode GSSAPI PLUGIN zu konfigurieren. Alternativ:
     - Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf den Wert NOT_SPECIFIED und den Konfigurationsparameter *authentication* auf den Wert GSSPLUGIN oder GSS_SERVER_ENCRYPT, um den Server zur Verwendung der Authentifizierungsmethode GSSAPI PLUGIN zu konfigurieren.
- Führen Sie die folgenden Schritte auf jedem Client aus, um ein GSS-API-Authentifizierungs-Plug-in auf Datenbankclients zu implementieren:
  1. Kopieren Sie die Bibliothek des GSS-API-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis. Sie können viele GSS-API-Plug-ins in dieses Verzeichnis kopieren.

Der Client wählt ein GSS-API-Plug-in zur Authentifizierung während einer CONNECT- oder ATTACH-Operation aus, indem er das erste, in der Plug-in-Liste des Servers enthaltene GSS-API-Plug-in verwendet, das auf dem Client verfügbar ist.

2. Optional: Katalogisieren Sie die Datenbanken, auf die der Client zugreifen soll, indem Sie angeben, dass der Client nur ein GSS-API-Authentifizierungs-Plug-in als Authentifizierungsverfahren akzeptiert. Beispiel:

   ```
   CATALOG DB testdb AT NODE testnode AUTHENTICATION GSSPLUGIN
   ```

- Führen Sie die folgenden Schritte zur Implementierung der Berechtigung auf einem Client, Server oder Gateway durch ein GSS-API-Authentifizierungs-Plug-in aus:
  1. Kopieren Sie die Bibliothek des GSS-API-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis auf dem Client, Server oder Gateway.
  2. Aktualisieren Sie den Konfigurationsparameter *local_gssplugin* des Datenbankmanagers mit dem Namen des Plug-ins.
  3. Setzen Sie den Konfigurationsparameter *authentication* des Datenbankmanagers auf den Wert GSSPLUGIN oder GSS_SERVER_ENCRYPT.

Implementieren eines Kerberos-Plug-ins

Zur Anpassung des Kerberos-Authentifizierungsverhaltens des DB2-Sicherheitssystems können Sie eigene Plug-ins für die Kerberos-Authentifizierung entwickeln oder ein Plug-in bei einem Fremdanbieter erwerben. Beachten Sie, dass das Kerberos-Sicherheits-Plug-in IPv6 nicht unterstützt.

Anmerkung: Sie müssen den DB2-Server und alle Anwendungen, die die Plug-ins verwenden, stoppen, bevor Sie eine *neue* Version eines *vorhandenen* Plug-ins implementieren. Undefiniertes Verhalten, einschließlich Traps, kann auftreten, wenn ein Prozess ein Plug-in zu dem Zeitpunkt verwendet, zu dem eine neue Version (mit demselben Namen) über das Plug-in kopiert wird. Diese Einschränkung gilt nicht, wenn Sie ein Plug-in zum ersten Mal implementieren oder wenn das Plug-in nicht im Gebrauch ist.

Nachdem Sie Kerberos-Authentifizierungs-Plug-ins erworben haben, die für Ihr Datenbankverwaltungssystem geeignet sind, können Sie sie implementieren.

- Führen Sie die folgenden Schritte auf dem Server aus, um ein Kerberos-Authentifizierungs-Plug-in auf dem Datenbankserver zu implementieren:
  1. Kopieren Sie die Bibliothek des Kerberos-Authentifizierungs-Plug-ins in das Server-Plug-in-Verzeichnis.
  2. Aktualisieren Sie den Konfigurationsparameter *srvcon_gssplugin_list* des Datenbankmanagers, der die Form einer geordneten, durch Kommata begrenzten Liste hat, um den Namen des Kerberos-Server-Plug-ins hinzuzufügen. Nur ein Plug-in in dieser Liste darf ein Kerberos-Plug-in sein. Wenn diese Liste leer ist und der Konfigurationsparameter *authentication* auf den Wert KERBEROS oder KRB_SVR_ENCRYPT gesetzt ist, wird das Standard-Plug-in IBMkrb5 von DB2 verwendet.
  3. Zum Implementieren und Verwenden eines Kerberos-Plug-ins gibt es zwei Optionen, mit denen der Konfigurationsparameter *srvcon_auth* des Datenbankmanagers angegeben werden kann:
     - Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf einen der folgenden Authentifizierungstypen:
       - KERBEROS
       - KRB_SERVER_ENCRYPT

- GSSPLUGIN
- GSS_SERVER_ENCRYPT

– Setzen Sie den Konfigurationsparameter *srvcon_auth* des Datenbankmanagers auf den Wert NOT_SPECIFIED. In diesem Fall verwendet DB2 den Wert von *authentication*, der auf einen der folgenden Authentifizierungstypen gesetzt werden könnte:
  - KERBEROS
  - KRB_SERVER_ENCRYPT
  - GSSPLUGIN
  - GSS_SERVER_ENCRYPT

- Führen Sie die folgenden Schritte auf jedem Client aus, um ein Kerberos-Authentifizierungs-Plug-in auf Datenbankclients zu implementieren:
  1. Kopieren Sie die Bibliothek des Kerberos-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis.
  2. Aktualisieren Sie den Konfigurationsparameter *clnt_krb_plugin* des Datenbankmanagers mit dem Namen des Kerberos-Plug-ins. Wenn der Parameter *clnt_krb_plugin* leer ist, nimmt DB2 an, dass der Client die Kerberos-Authentifizierung nicht verwenden kann. Diese Einstellung ist nur korrekt, wenn der Server Plug-ins nicht unterstützen kann. Wenn der Server und der Client Sicherheits-Plug-ins unterstützen, würde das Standard-Server-Plug-in *IBMkrb5* anstelle des Clientwerts *clnt_krb_plugin* verwendet. Führen Sie die folgenden Schritte zur Implementierung der Berechtigung auf einem Client, Server oder Gateway durch ein Kerberos-Authentifizierungs-Plug-in aus:
     a. Kopieren Sie die Bibliothek des Kerberos-Authentifizierungs-Plug-ins in das Client-Plug-in-Verzeichnis auf dem Client, Server oder Gateway.
     b. Aktualisieren Sie den Konfigurationsparameter *clnt_krb_plugin* des Datenbankmanagers mit dem Namen des Plug-ins.
     c. Setzen Sie den Konfigurationsparameter *authentication* des Datenbankmanagers auf den Wert KERBEROS oder KRB_SERVER_ENCRYPT.
  3. Optional: Katalogisieren Sie die Datenbanken, auf die der Client zugreifen soll, indem Sie angeben, dass der Client nur ein Kerberos-Authentifizierungs-Plug-in akzeptiert. Beispiel:

```
CATALOG DB testdb AT NODE testnode AUTHENTICATION KERBEROS
        TARGET PRINCIPAL service/host@REALM
```

Anmerkung: Auf Plattformen, die Kerberos unterstützen, befindet sich die Bibliothek des Plug-ins `IBMkrb5` im Client-Plug-in-Verzeichnis. DB2 erkennt diese Bibliothek als gültiges GSS-API-Plug-in, da Kerberos-Plug-ins mithilfe von GSS-API-Plug-ins implementiert werden.

Unterstützung für LDAP-basierte Authentifizierung und Gruppensuchfunktion

Der DB2-Datenbankmanager und DB2 Connect unterstützen die LDAP-basierte Funktionalität zur Authentifizierung und Gruppensuche durch den Einsatz von LDAP-Sicherheits-Plug-in-Modulen (LDAP - Lightweight Directory Access Protocol).

LDAP-Sicherheits-Plug-in-Module bieten dem DB2-Datenbankmanager die Möglichkeit, Benutzer, die in einem LDAP-Verzeichnis definiert sind, zu authentifizieren, ohne dass die Benutzer und Gruppen dazu im Betriebssystem definiert sein müssen.

Unterstützte Plattformen sind AIX, Linux auf IA32, Linux auf x64, Linux auf zSeries, Solaris und Windows. Kompilierte binäre Plug-in-Module für diese unterstützten Plattformen sind in den entsprechenden Verzeichnissen zu finden (z. B. aix64, win32 usw.).

Die folgenden LDAP-Server werden zur Verwendung mit Sicherheits-Plug-in-Modulen unterstützt:

- IBM Tivoli Directory Server (ITDS) Version 5.2, 6.0 und spätere Versionen
- Microsoft Active Directory (MSAD) Version 2000, 2003 und spätere Versionen
- Sun Java System Directory Server Enterprise Edition Version 5.2 und spätere Versionen
- Novell eDirectory Version 8.7 und spätere Versionen
- IBM Lotus Domino LDAP Server Version 7.0 und spätere Versionen
- z/OS Integrated Security Services LDAP Server Version V1R6 und spätere Versionen

Anmerkung: Wenn Sie die LDAP-Plug-in-Module verwenden, müssen alle Benutzer, die der Datenbank zugeordnet sind, im LDAP-Server definiert sein. Dies schließt sowohl die DB2-Instanzeigner-ID als auch den abgeschirmten Benutzer mit ein. (Diese Benutzer werden in der Regel im Betriebssystem definiert, müssen jedoch auch in LDAP definiert werden.) In ähnlicher Weise gilt, dass bei Verwendung des LDAP-Gruppen-Plug-in-Moduls alle Gruppen, die zur Authentifizierung erforderlich sind, im LDAP-Server definiert sein müssen. Dies schließt die Gruppen SYSADM, SYSMAINT, SYSCTRL und SYSMON mit ein, die in der Datenbankmanagerkonfiguration definiert sind.

DB2-Sicherheits-Plug-in-Module sind für die serverseitige Authentifizierung, die clientseitige Authentifizierung und die Gruppensuche verfügbar. Diese Funktionen werden nachfolgend beschrieben. Abhängig von Ihrer speziellen Umgebung müssen Sie einen, zwei oder alle drei Typen von Plug-in verwenden.

Gehen Sie zur Verwendung von DB2-Sicherheits-Plug-in-Modulen wie folgt vor:

1. Entscheiden Sie, ob Sie Server-, Client- oder Gruppen-Plug-in-Module oder eine Kombination dieser Module benötigen.
2. Konfigurieren Sie die Plug-in-Module, indem Sie Werte in der Konfigurationsdatei für IBM LDAP-Sicherheits-Plug-ins (Standardname: IBMLDAPSecurity.ini) definieren. Sie müssen sich wahrscheinlich bei Ihrem LDAP-Administrator nach den geeigneten Werten erkundigen.
3. Aktivieren Sie die Plug-in-Module.
4. Testen Sie den Verbindungsaufbau mit verschiedenen LDAP-Benutzer-IDs.

Plug-in zur Serverauthentifizierung

Das Plug-in-Modul zur Serverauthentifizierung führt eine serverseitige Prüfung von Benutzer-IDs und Kennwörtern aus, die von Clients in CONNECT- und ATTACH-Anweisungen angegeben werden. Es stellt außerdem eine Methode bereit, LDAP-Benutzer-IDs DB2-Berechtigungs-IDs zuzuordnen, wenn dies erforderlich ist. Das Server-Plug-in-Modul ist im Allgemeinen erforderlich, wenn Benutzer für den DB2-Datenbankmanager mit ihrer LDAP-Benutzer-ID und dem zugehörigen Kennwort authentifiziert werden sollen.

Plug-in zur Clientauthentifizierung

Das Plug-in-Modul zur Clientauthentifizierung wird verwendet, wenn die Prüfung von Benutzer-ID und Kennwort auf dem Clientsystem stattfindet. Das heißt, wenn der DB2-Server mit der Einstellung CLIENT des Parameters SRVCON_AUTH oder AUTHENTICATION konfiguriert ist. Der Client prüft alle Benutzer-IDs und Kennwörter, die in CONNECT- oder ATTACH-Anweisungen angegeben werden, und sendet die Benutzer-ID an den DB2-Server. Beachten Sie, dass sich die Clientauthentifizierung nur schwer schützen lässt und im Allgemeinen nicht empfohlen wird.

Das Plug-in-Modul zur Clientauthentifizierung kann außerdem erforderlich sein, wenn sich die Benutzer-IDs des lokalen Betriebssystems auf dem Datenbankserver von den DB2-Berechtigungs-IDs, die diesen Benutzern zugeordnet sind, unterscheiden. Mithilfe des clientseitigen Plug-ins können Sie Benutzer-IDs des lokalen Betriebssystems DB2-Berechtigungs-IDs zuordnen, bevor Sie Berechtigungsprüfungen für lokale Befehle auf dem Datenbankserver (z. B. für `db2start`) durchführen.

Plug-in zum Abrufen von Gruppen

Das Plug-in-Modul zum Abrufen von Gruppen ruft Informationen zur Gruppenzugehörigkeit für einen bestimmten Benutzer aus dem LDAP-Server ab. Es ist erforderlich, wenn Sie Ihre Gruppendefinitionen in LDAP speichern wollen. Das gängigste Szenario weist folgende Merkmale auf:

- Alle Benutzer und Gruppen sind im LDAP-Server definiert.
- Alle Benutzer, die lokal auf dem Datenbankserver definiert sind, sind mit derselben Benutzer-ID im LDAP-Server definiert (einschließlich des Instanzeigners und des abgeschirmten Benutzers).
- Die Kennwortprüfung findet auf dem DB2-Server statt (d. h. in der Konfigurationsdatei des Datenbankmanagers des Servers ist der Parameter AUTHENTICATION oder SRVCON_AUTH auf den Wert SERVER, SERVER_ENCRYPT oder DATA_ENCRYPT gesetzt).

Im Regelfall ist es ausreichend, nur das Plug-in-Modul zur Serverauthentifizierung und das Plug-in-Modul zum Abrufen von Gruppen auf dem Server zu installieren. DB2-Clients benötigen im Normalfall keine Installation des LDAP-Plug-in-Moduls.

Es ist möglich, nur das Plug-in-Modul zum Abrufen von LDAP-Gruppen in Kombination mit einer anderen Form von Authentifizierungs-Plug-in (z. B. Kerberos) zu verwenden. In diesem Fall werden dem Plug-in-Modul zum Abrufen von LDAP-Gruppen die DB2-Berechtigungs-IDs übergeben, die einem Benutzer zugeordnet sind. Das Plug-in-Modul durchsucht das LDAP-Verzeichnis nach einem Benutzer mit einem entsprechenden Attribut AUTHID_ATTRIBUTE und ruft anschließend die Gruppen ab, die diesem Benutzerobjekt zugeordnet sind.

Konfigurieren der LDAP-Plug-in-Module

Zur Konfiguration der LDAP-Plug-in-Module (LDAP, Lightweight Directory Access Protocol) müssen Sie Ihre Konfigurationsdatei für IBM LDAP-Sicherheits-Plug-ins aktualisieren, um sie an Ihre Umgebung anzupassen. In den meisten Fällen müssen Sie sich bei Ihrem LDAP-Administrator über die erforderlichen Konfigurationswerte informieren.

Die Konfigurationsdatei für IBM LDAP-Sicherheits-Plug-ins hat standardmäßig folgenden Namen und folgende Position:

- Unter UNIX: INSTHOME/sqllib/cfg/IBMLDAPSecurity.ini
- Unter Windows: %DB2PATH%\cfg\IBMLDAPSecurity.ini

Optional können Sie die Position dieser Datei mit der Umgebungsvariablen 'DB2LDAPSecurityConfig' angeben. Unter Windows sollten Sie die Variable 'DB2LDAPSecurityConfig' in der globalen Systemumgebung definieren, um sicherzustellen, dass sie vom DB2-Service berücksichtigt wird.

Die folgenden Tabellen enthalten Informationen, die Ihnen bei der Bestimmung geeigneter Konfigurationswerte helfen.

Tabelle 30. Serverbezogene Werte

Parameter	Beschreibung
LDAP_HOST	Der Name Ihres bzw. Ihrer LDAP-Server. Dieser Wert ist eine durch Leerzeichen getrennte Liste von Hostnamen oder IP-Adressen (mit optionaler Portnummer) für LDAP-Server. Beispiel: `host1[:port] [host2:[port2] ... ]` Die Standardportnummer ist 389, oder 636, wenn SSL aktiviert ist.
ENABLE_SSL	Zum Aktivieren der SSL-Unterstützung setzen Sie ENABLE_SSL auf TRUE. (Das GSKit muss installiert sein.) Dieser Parameter ist optional; sein Standardwert ist FALSE (keine SSL-Unterstützung).
SSL_KEYFILE	Der Pfad für die SSL-Schlüsselringdatei. Eine Schlüsseldatei (keyfile) ist nur erforderlich, wenn der LDAP-Server ein Zertifikat verwendet, dem von Ihrer GSKit-Installation nicht automatisch vertraut wird. Beispiel: `SSL_KEYFILE = /home/db2inst1/IBMLDAPSecurity.kdb`
SSL_PW	Das Kennwort für die SSL-Schlüsselringdatei. Beispiel: `SSL_PW = kennwort-für-schlüsselringdatei`

Tabelle 31. Benutzerbezogene Werte

Parameter	Beschreibung
USER_ OBJECTCLASS	Die LDAP-Objektklasse für Benutzer. Setzen Sie USER_OBJECTCLASS im Allgemeinen auf `inetOrgPerson` (den Benutzer für Microsoft Active Directory) Beispiel: `USER_OBJECTCLASS = inetOrgPerson`
USER_BASEDN	Der LDAP-Basis-DN für die Suche nach Benutzern. Wenn nicht angegeben, beginnen Benutzersuchen im Stammelement des LDAP-Verzeichnisses. Einige LDAP-Server setzen voraus, dass Sie einen Wert für diesen Parameter angeben. Beispiel: `USER_BASEDN = o=ibm`
USERID_ ATTRIBUTE	Das LDAP-Benutzerattribut, das die Benutzer-ID darstellt. Das Attribut USERID_ATTRIBUTE wird mit USER_OBJECTCLASS und USER_BASEDN kombiniert (falls angegeben), um einen LDAP-Suchfilter zu bilden, wenn ein Benutzer eine DB2-Anweisung CONNECT mit einer nicht qualifizierten Benutzer-ID absetzt. Beispiel: Bei `USERID_ATTRIBUTE = uid` wird die folgende Anweisung abgesetzt: `db2 connect to MYDB user bob using bobpass` In diesem Fall wird der folgende Suchfilter gebildet: `&(objectClass=inetOrgPerson)(uid=bob)`

Tabelle 31. Benutzerbezogene Werte (Forts.)

Parameter	Beschreibung
AUTHID_ ATTRIBUTE	Das LDAP-Benutzerattribut, das die DB2-Berechtigungs-ID darstellt. Dieser Wert ist in der Regel gleich USERID_ATTRIBUTE. Beispiel: `AUTHID_ATTRIBUTE = uid`

Tabelle 32. Gruppenbezogene Werte

Parameter	Beschreibung
GROUP_ OBJECTCLASS	Die LDAP-Objektklasse für Gruppen. Dieser Wert ist in der Regel `groupOfNames` oder `groupOfUniqueNames` (für Microsoft Active Directory ist dies `group`). Beispiel: `GROUP_OBJECTCLASS = groupOfNames`
GROUP_BASEDN	Der LDAP-Basis-DN für die Suche nach Gruppen. Wenn nicht angegeben, beginnen Gruppensuchen im Stammelement des LDAP-Verzeichnisses. Einige LDAP-Server setzen voraus, dass Sie einen Wert für diesen Parameter angeben. Beispiel: `GROUP_BASEDN = o=ibm`
GROUPNAME_ ATTRIBUTE	Das LDAP-Gruppenattribut, das den Namen der Gruppe darstellt. Beispiel: `GROUPNAME_ATTRIBUTE = cn`

GROUP_LOOKUP_ METHOD	Bestimmt die Methode zur Suche der Gruppenzugehörigkeiten für einen Benutzer. Mögliche Werte:
	• SEARCH_BY_DN: Gibt an, dass nach Gruppen zu suchen ist, die den Benutzer als Mitglied auflisten. Die Zugehörigkeit wird durch das Gruppenattribut angegeben, das als GROUP_LOOKUP_ATTRIBUTE definiert ist (in der Regel `member` oder `uniqueMember`).
	• USER_ATTRIBUTE: In diesem Fall werden die Gruppen eines Benutzers als Attribute des Benutzerobjekts selbst aufgelistet. Diese Einstellung gibt an, dass nach dem Benutzerattribut zu suchen ist, das als GROUP_LOOKUP_ATTRIBUTE definiert ist, um die Gruppen des Benutzers abzurufen (in der Regel `memberOf` für Microsoft Active Directory oder `ibm-allGroups` für IBM Tivoli Directory Server).
	Zum Beispiel:
	`GROUP_LOOKUP_METHOD = SEARCH_BY_DN`
	`GROUP_LOOKUP_METHOD = USER_ATTRIBUTE`

GROUP_LOOKUP_ ATTRIBUTE	Der Name des Attributs, das zur Ermittlung der Gruppenzugehörigkeit wie für GROUP_LOOKUP_METHOD beschrieben verwendet wird.
	
	Zum Beispiel:
	`GROUP_LOOKUP_ATTRIBUTE = member`
	`GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups`

NESTED_GROUPS	Wenn NESTED_GROUPS auf TRUE gesetzt ist, sucht der DB2-Datenbankmanager rekursiv nach Gruppenzugehörigkeiten, indem er versucht, die Gruppenzugehörigkeiten für jede gefundene Gruppe zu ermitteln. Zyklen (Beispiel: A gehört zu B und B gehört zu A) werden ordnungsgemäß verarbeitet. Dieser Parameter ist optional und hat den Standardwert FALSE.

Tabelle 33. Verschiedene Werte

Parameter	Beschreibung

SEARCH_DN, SEARCH_PW	Wenn Ihr LDAP-Server keinen anonymen Zugriff unterstützt oder wenn ein anonymer Zugriff für die Suche nach Benutzern oder Gruppen nicht ausreicht, können Sie optional einen definierten Namen (DN) und ein Kennwort definieren, die zur Ausführung von Suchen verwendet werden.
	
	Zum Beispiel:
	`SEARCH_DN = cn=root`
	`SEARCH_PW = rootkennwort`

DEBUG	Setzen Sie DEBUG auf den Wert TRUE, wenn zusätzliche Informationen in die Datei db2diag.log geschrieben werden sollen, um ein Debugging für LDAP zu unterstützen. Die meisten Zusatzinformationen werden bei folgender Stufe protokolliert: DIAGLEVEL 4 (INFO). DEBUG hat den Standardwert FALSE.

Aktivieren der LDAP-Plug-in-Module

Die folgenden Tabellen zeigen, wo sich die LDAP-Plug-in-Module in Ihrer DB2-Instanz befinden.

Tabelle 34. Für UNIX- und Linux-Systeme (64 Bit)

Typ des Plug-in-Moduls	Position
Server	/sqllib/security64/plugin/IBM/server
Client	/sqllib/security64/plugin/IBM/client
Gruppe	/sqllib/security64/plugin/IBM/group

Tabelle 35. Für UNIX- und Linux-Systeme (32 Bit)

Typ des Plug-in-Moduls	Position
Server	/sqllib/security32/plugin/IBM/server
Client	/sqllib/security32/plugin/IBM/client
Gruppe	/sqllib/security32/plugin/IBM/group

Tabelle 36. Für Windows-Systeme (64- und 32 Bit)

Typ des Plug-in-Moduls	Position
Server	%DB2PATH%\security\plugin\IBM*instanzname*\server
Client	%DB2PATH%\security\plugin\IBM*instanzname*\client
Gruppe	%DB2PATH%\security\plugin\IBM*instanzname*\group

Anmerkung: Die Plug-in-Module für 64-Bit-Windows-Systeme haben die Ziffern ″64″ im Dateinamen.

Verwenden Sie den DB2-Befehlszeilenprozessor, um die Datenbankmanagerkonfiguration zur Aktivierung der erforderlichen Plug-in-Module zu aktualisieren:

- Für das Server-Plug-in-Modul:

  ```
  UPDATE DBM CFG USING SRVCON_PW_PLUGIN IBMLDAPauthserver
  ```

- Für das Client-Plug-in-Modul:

  ```
  UPDATE DBM CFG USING CLNT_PW_PLUGIN IBMLDAPauthclient
  ```

- Für das Gruppen-Plug-in-Modul:

  ```
  UPDATE DBM CFG USING GROUP_PLUGIN IBMLDAPgroups
  ```

Beenden Sie alle laufenden Back-End-Prozesse des DB2-Befehlszeilenprozessors, indem Sie den Befehl db2 terminate ausführen. Führen Sie anschließend die Befehle db2stop und db2start aus, um die Instanz zu stoppen und erneut zu starten.

Verbindungsaufbau mit einer LDAP-Benutzer-ID

Die Position eines Objekts in einem LDAP-Verzeichnis (Lightweight Directory Access Protocol) wird durch den definierten Namen (DN) festgelegt.

Ein definierter Name (DN) besteht in der Regel aus mehreren Teilen, die eine Art von Hierarchie darstellen. Beispiel:

```
cn=John Smith, ou=Sales, o=WidgetCorp
```

Nach dem Aktivieren und Konfigurieren eines LDAP-Plug-in-Moduls kann ein Benutzer eine Verbindung zu einer DB2-Datenbank mithilfe einer Reihe verschiedener Zeichenfolgen herstellen:

- Ein vollständiger definierter Name (DN). Beispiel:

  ```
  connect to MYDB user 'cn=John Smith, ou=Sales, o=WidgetCorp'
  ```

- Ein Teil eines definierten Namens (Teil-DN), vorausgesetzt, dass eine Suche im LDAP-Verzeichnis mit dem Teil-DN und dem entsprechenden Basis-DN (falls definiert) genau eine Übereinstimmung liefert. Beispiel:

  ```
  connect to MYDB user 'cn=John Smith'
  connect to MYDB user uid=jsmith
  ```

- Eine einfache Zeichenfolge (ohne Gleichheitszeichen). Die Zeichenfolge wird mit dem Wert des Attributs USERID_ATTRIBUTE qualifiziert und wie ein Teil-DN behandelt. Beispiel:

  ```
  connect to MYDB user jsmith
  ```

Anmerkung: Jede Zeichenfolge, die in einer CONNECT- oder ATTACH-Anweisung angegeben wird, muss in einfache Anführungszeichen gesetzt werden, wenn sie Leerzeichen oder Sonderzeichen enthält.

Benutzer-IDs und DB2-Berechtigungs-IDs

Die *Benutzer-ID* eines Benutzers wird durch ein Attribut definiert, das dem Benutzerobjekt zugeordnet ist (in der Regel das Attribut uid). Dabei kann es sich um eine einfache Zeichenfolge (z. B. jsmith) oder um eine Zeichenfolge nach Art einer E-Mail-Adresse (z. B. jsmith@sales.widgetcorp.com), die einen Teil der Organisationshierarchie widerspiegelt, handeln.

Die *Berechtigungs-ID* eines Benutzers in DB2 ist der Name, der diesem Benutzer innerhalb der DB2-Datenbank zugeordnet wird.

In der Vergangenheit wurden Benutzer in der Regel im Hostbetriebssystem des Servers definiert, und die Benutzer-ID und die Berechtigungs-ID stimmten überein (obwohl für die Berechtigungs-ID normalerweise Großbuchstaben verwendet werden). Die DB2-LDAP-Plug-in-Module geben Ihnen die Möglichkeit, der Benutzer-ID und der Berechtigungs-ID verschiedene Attribute des LDAP-Benutzerobjekts zuzuordnen. In den meisten Fällen können die Benutzer-ID und die Berechtigungs-ID dieselbe Zeichenfolge sein, und Sie können denselben Attributnamen für USERID_ATTRIBUTE und AUTHID_ATTRIBUTE verwenden. Wenn das Benutzer-ID-Attribut in Ihrer Umgebung in der Regel zusätzliche Informationen enthält, die nicht auf die Berechtigungs-ID übertragen werden sollen, können Sie ein anderes Attribut AUTHID_ATTRIBUTE in der Plug-in-Initialisierungsdatei konfigurieren. Der Wert des Attributs AUTHID_ATTRIBUTE wird aus dem Server abgerufen und als interne DB2-Darstellung des Benutzers verwendet.

Wenn Ihre LDAP-Benutzer-IDs zum Beispiel wie E-Mail-Adressen (z. B. jsmith@sales.widgetcorp.com) aussehen und Sie die Verwendung nur des Benutzerteils (jsmith) als DB2-Berechtigungs-ID vorziehen, können Sie wie folgt vorgehen:

1. Ordnen Sie allen Benutzerobjekten auf Ihrem LDAP-Server ein neues Attribut zu, das nur den kürzeren Namen enthält.
2. Konfigurieren Sie das Attribut AUTHID_ATTRIBUTE mit dem Namen dieses neuen Attributs.

Benutzer können anschließend eine Verbindung zu einer DB2-Datenbank herstellen, indem sie ihre vollständige LDAP-Benutzer-ID und das Kennwort angeben. Beispiel:

```
db2 connect to MYDB user 'jsmith@sales.widgetcorp.com' using 'kennwort'
```

Intern verwendet der DB2-Datenbankmanager für den Benutzer jedoch den kurzen Namen, der mithilfe des Attributs AUTHID_ATTRIBUTE abgerufen wird (in diesem Beispiel: jsmith).

Hinweise zur Gruppensuchfunktion

Informationen zu Gruppenzugehörigkeiten werden in einem LDAP-Server in der Regel als Attribut des Benutzerobjekts oder als Attribut des Gruppenobjekts dargestellt:

- Als Attribut des Benutzerobjekts

  Jedes Benutzerobjekt hat ein Attribut mit dem Namen GROUP_LOOKUP_ATTRIBUTE, das Sie abfragen können, um alle Gruppenzugehörigkeiten für den bestimmten Benutzer abzurufen.

- Als Attribut des Gruppenobjekts

  Jedes Gruppenobjekt hat ein Attribut, das ebenfalls den Namen GROUP_LOOKUP_ATTRIBUTE hat, mit dessen Hilfe Sie alle Benutzerobjekte auflisten können, die Mitglieder der Gruppe sind. Sie können die Gruppen für einen bestimmten Benutzer aufzählen, indem Sie nach allen Gruppen suchen, die das Benutzerobjekt als Mitglied auflisten.

Viele LDAP-Server können auf beide Arten konfiguriert werden, und manche unterstützen beide Methoden gleichzeitig. Wenden Sie sich an Ihren LDAP-Administrator, um zu erfahren, wie Ihr LDAP-Server konfiguriert ist.

Bei der Konfiguration der LDAP-Plug-in-Module können Sie mit dem Parameter GROUP_LOOKUP_METHOD angeben, wie die Gruppensuche ausgeführt werden soll:

- Wenn Sie das Attribut GROUP_LOOKUP_ATTRIBUTE des Benutzerobjekts zum Ermitteln von Gruppenzugehörigkeiten verwenden müssen, konfigurieren Sie GROUP_LOOKUP_METHOD = USER_ATTRIBUTE.
- Wenn Sie das Attribut GROUP_LOOKUP_ATTRIBUTE des Gruppenobjekts zum Ermitteln von Gruppenzugehörigkeiten verwenden müssen, konfigurieren Sie GROUP_LOOKUP_METHOD = SEARCH_BY_DN.

Viele LDAP-Server verwenden das Attribut GROUP_LOOKUP_ATTRIBUTE des Gruppenobjekts zur Ermittlung von Zugehörigkeiten. Sie können wie im folgenden Beispiel gezeigt konfiguriert werden:

```
GROUP_LOOKUP_METHOD = SEARCH_BY_DN
GROUP_LOOKUP_ATTRIBUTE = groupOfNames
```

Microsoft Active Directory speichert Gruppenzugehörigkeiten in der Regel als Benutzerattribut und kann daher wie im folgenden Beispiel gezeigt konfiguriert werden:

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = memberOf
```

IBM Tivoli Directory Server unterstützt beide Methoden gleichzeitig. Zum Abfragen der Gruppenzugehörigkeiten für einen Benutzer können Sie das spezielle Benutzerattribut ibm-allGroups wie im folgenden Beispiel verwenden:

```
GROUP_LOOKUP_METHOD = USER_ATTRIBUTE
GROUP_LOOKUP_ATTRIBUTE = ibm-allGroups
```

Andere LDAP-Server können ähnliche spezielle Attribute zur Verfügung stellen, um das Abrufen von Gruppenzugehörigkeiten zu unterstützen. Im Allgemeinen ist das Abrufen von Zugehörigkeiten über ein Benutzerattribut schneller als die Suche nach Gruppen, die den gewünschten Benutzer als Mitglied auflisten.

Fehlerbehebung beim Authentifizieren von LDAP-Benutzern oder beim Abrufen von Gruppen

Wenn Sie Probleme bei der Authentifizierung von LDAP-Benutzern oder beim Abrufen der Gruppen von LDAP-Benutzern feststellen, sind das DB2-Diagnoseprotokoll db2diag.log und das Verwaltungsprotokoll gute Informationsquellen, die Sie bei der Fehlerbehebung unterstützen.

Die LDAP-Plug-in-Module protokollieren LDAP-Rückkehrcodes, Suchfilter und andere nützliche Daten in der Regel dann, wenn ein Fehler auftritt. Wenn Sie die Option DEBUG in der LDAP-Plug-in-Konfigurationsdatei aktivieren, protokollieren die Plug-in-Module darüber hinaus weitere Informationen in der Datei db2diag.log. Obwohl diese Option bei der Fehlerbehebung eine Hilfe sein kann, wird sie aufgrund des Aufwands, der mit dem Schreiben der zusätzlichen Daten in eine Datei verbunden ist, nicht zur dauerhaften Verwendung auf Produktionssystemen empfohlen.

Stellen Sie sicher, dass der Konfigurationsparameter DIAGLEVEL im Datenbankmanager auf den Wert 4 gesetzt ist, sodass alle Nachrichten aus den LDAP-Plug-in-Modulen erfasst werden.

Schreiben von Sicherheits-Plug-ins

Laden von Sicherheits-Plug-ins in DB2

Jede Plug-in-Bibliothek muss eine Initialisierungsfunktion mit einem bestimmten Namen enthalten, der durch den Plug-in-Typ festgelegt wird:

- Serverseitiges Authentifizierungs-Plug-in: `db2secServerAuthPluginInit()`
- Clientseitiges Authentifizierungs-Plug-in: `db2secClientAuthPluginInit()`
- Gruppen-Plug-in: `db2secGroupPluginInit()`

Diese Funktion wird als Plug-in-Initialisierungsfunktion bezeichnet. Die Plug-in-Initialisierungsfunktion initialisiert das angegebene Plug-in und stellt Informationen bereit, die DB2 zum Aufrufen der Funktionen des Plug-ins benötigt. Die Plug-in-Initialisierungsfunktion akzeptiert die folgenden Parameter:

- Die höchste Versionsnummer der Funktionszeigerstruktur, die von der DB2-Instanz, die das Plug-in aufruft, unterstützt werden kann
- Ein Zeiger auf eine Struktur, die Zeiger auf alle APIs enthält, die implementiert werden müssen
- Ein Zeiger auf eine Funktion, die Protokollnachrichten in die Datei db2diag.log einfügt
- Ein Zeiger auf eine Fehlernachrichtenzeichenfolge
- Die Länge der Fehlernachricht

Die Funktionskennung für die Initialisierungsfunktion eines Plug-ins zum Abrufen von Gruppen sieht wie folgt aus:

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit(
  db2int32 version,
  void *group_fns,
  db2secLogMessage *logMessage_fn,
  char **errormsg,
  db2int32  *errormsglen);
```

Anmerkung: Wenn die Plug-in-Bibliothek als C++ kompiliert wird, müssen alle Funktionen mit `extern "C"` deklariert werden. DB2 greift auf das dynamische Ladeprogramm des zugrunde liegenden Betriebssystems zurück, um die C++-Konstruktoren und -Destruktoren zu verarbeiten, die in einer benutzerdefinierten C++-Plug-in-Bibliothek verwendet werden.

Die Initialisierungsfunktion ist die einzige Funktion in der Plug-in-Bibliothek, die einen vorgeschriebenen Funktionsnamen verwendet. Die anderen Plug-in-Funktionen werden über Funktionszeiger angegeben, die aus der Initialisierungsfunktion zurückgegeben werden. Server-Plug-ins werden geladen, wenn der DB2-Server gestartet wird. Client-Plug-ins werden geladen, wenn sie auf dem Client erforderlich sind. Unmittelbar nach dem Laden einer Plug-in-Bibliothek löst DB2 die Position dieser Initialisierungsfunktion auf und ruft sie auf. Diese Funktion hat die folgende spezielle Aufgabe:

- Sie setzt den Funktionszeiger in einen Zeiger auf eine entsprechende Funktionsstruktur um.
- Sie fügt Werte in die Zeiger auf die anderen Funktionen in der Bibliothek ein.
- Sie fügt die Versionsnummer der Funktionszeigerstruktur ein, die zurückgegeben wird.

DB2 kann die Plug-in-Initialisierungsfunktion potenziell mehrmals aufrufen. Dieser Fall kann eintreten, wenn eine Anwendung die DB2-Clientbibliothek dynamisch

lädt, sie wieder entlädt und erneut lädt und anschließend Authentifizierungsfunktionen aus einem Plug-in sowohl vor als auch nach dem erneuten Laden ausführt. In dieser Situation wird die Plug-in-Bibliothek möglicherweise nicht entladen und dann erneut geladen. Allerdings ist dieses Verhalten je nach Betriebssystem unterschiedlich.

Ein weiteres Beispiel für einen Fall, in dem DB2 mehrere Aufrufe an eine Plug-in-Initialisierungsfunktion absetzt, ist die Ausführung von gespeicherten Prozeduren oder Systemaufrufen in einer Umgebung mit föderierten Datenbanken, in denen der Datenbankserver selbst als Client fungieren kann. Wenn sich die Client- und Server-Plug-ins auf dem Datenbankserver in derselben Datei befinden, könnte DB2 die Plug-in-Initialisierungsfunktion zweimal aufrufen.

Wenn das Plug-in erkennt, dass die Funktion `db2secGroupPluginInit` mehr als einmal aufgerufen wurde, sollte es dieses Ereignis so behandeln, als wäre es angewiesen worden, die Verarbeitung zu beenden und die Plug-in-Bibliothek zu reinitialisieren. Die Plug-in-Initialisierungsfunktion sollte selbst alle Bereinigungsaufgaben ausführen, die von einem Aufruf von `db2secPluginTerm` ausgeführt würden, bevor sie erneut eine Gruppe von Funktionszeigern zurückgibt.

Auf einem DB2-Server, der unter einem UNIX- oder Linux-basierten Betriebssystem ausgeführt wird, kann DB2 Plug-in-Bibliotheken potenziell mehr als einmal in verschiedenen Prozessen laden und initialisieren.

Einschränkungen für die Entwicklung von Sicherheits-Plug-in-Bibliotheken

Die folgenden Einschränkungen sind bei der Entwicklung von Plug-in-Bibliotheken zu beachten:

C-Verlinkung
Plug-in-Bibliotheken müssen mit der C-Verlinkung (C-linkage) verlinkt werden. Headerdateien, die Prototypen definieren, Datenstrukturen, die die Plug-ins implementieren, sowie Fehlercodedefinitionen werden nur für C/C++ zur Verfügung gestellt. Funktionen, die von DB2 zur Ladezeit aufgelöst werden, müssen mit `extern "C"` deklariert werden, wenn die Plug-in-Bibliothek als C++ kompiliert wird.

Keine Unterstützung von .NET Common Language Runtime
.NET Common Language Runtime (CLR) wird für die Kompilierung und Verlinkung von Quellcode für Plug-in-Bibliotheken nicht unterstützt.

Signalroutinen
Plug-in-Bibliotheken dürfen keine Signalroutinen (Signal Handler) installieren oder die Signalmaske ändern, da dies zu einer Kollision mit den Signalroutinen von DB2 führt. Eine Kollision der DB2-Signalroutinen könnte ernste Folgen für die Fähigkeit von DB2 haben, Fehler zu melden und zu beheben, was auch Traps im Plug-in-Code selbst betrifft. Plug-in-Bibliotheken sollten nie C++-Ausnahmebedingungen auslösen, da dies ebenfalls mit der Fehlerbehandlung durch DB2 kollidieren könnte.

Threadsicherheit
Plug-in-Bibliotheken müssen threadsicher und simultan verwendbar sein. Die Plug-in-Initialisierungsfunktion ist die einzige API, die nicht simultan verwendbar sein muss. Die Plug-in-Initialisierungsfunktion könnte potenziell von verschiedenen Prozessen aufgerufen werden. In diesem Fall bereinigt das Plug-in alle genutzten Ressourcen und reinitialisiert sich selbst.

Exitsteuerroutinen und Überschreiben von Aufrufen der C-Standardbibliotheken und von Betriebssystemaufrufen

Plug-in-Bibliotheken dürfen Aufrufe der C-Standardbibliotheken und Betriebssystemaufrufe nicht überschreiben. Plug-in-Bibliotheken dürfen außerdem keine Exitsteuerroutinen (Exit Handler) oder Steuerroutinen `pthread_atfork` installieren. Die Verwendung von Exitsteuerroutinen wird nicht empfohlen, weil sie aus dem Speicher entladen werden können, bevor das Programm beendet wird.

Bibliotheksabhängigkeiten

Unter Linux oder UNIX können die Prozesse, die die Plug-in-Bibliotheken laden, `setuid` oder `setgid` sein. Dies bedeutet, dass sie nicht auf die Umgebungsvariablen `$LD_LIBRARY_PATH`, `$SHLIB_PATH` oder `$LIBPATH` zurückgreifen können, um abhängige Bibliotheken zu finden. Daher sollten die Plug-in-Bibliotheken nicht von weiteren Bibliotheken abhängig sein, sofern auf solche abhängigen Bibliotheken nicht aufgrund anderer Bedingungen wie zum Beispiel der folgenden zugegriffen werden kann:

- Die Bibliotheken befinden sich im Verzeichnis `/lib` oder `/usr/lib`.
- Die Verzeichnisse, in denen sich die Bibliotheken befinden, sind betriebssystemweit bekannt (z. B. durch Angabe in der Datei `ld.so.conf` unter Linux).
- Die Bibliotheken sind in `RPATH` in der Plug-in-Bibliothek selbst angegeben.

Diese Einschränkung gilt nicht für Windows-Systeme.

Symbolkollisionen

Plug-in-Bibliotheken sollten nach Möglichkeit mit allen verfügbaren Optionen kompiliert und verlinkt werden, die die Wahrscheinlichkeit von Symbolkollisionen verringern, wie zum Beispiel mit den Optionen, die nicht gebundene externe symbolische Referenzen reduzieren. Zum Beispiel kann die Verwendung der Linkeroption "-Bsymbolic" unter HP, Solaris und Linux dabei helfen, Probleme im Zusammenhang mit Symbolkollisionen zu vermeiden. Allerdings darf für Plug-ins, die unter AIX geschrieben werden, die Linkeroption `"-brtl"` weder explizit noch implizit verwendet werden.

32- und 64-Bit-Anwendungen

32-Bit-Anwendungen müssen mit 32-Bit-Plug-ins arbeiten. 64-Bit-Anwendungen müssen mit 64-Bit-Plug-ins arbeiten. Weitere Details finden Sie in den Hinweisen zu 32- und 64-Bit-Sicherheits-Plug-ins.

Textzeichenfolgen

Es ist nicht garantiert, dass Eingabetextzeichenfolgen auf Null enden, und es ist nicht erforderlich, dass Ausgabezeichenfolgen auf Null enden. Stattdessen werden ganzzahlige Längenwerte für alle Eingabezeichenfolgen und Zeiger auf ganzzahlige Werte für zurückzugebende Längen angegeben.

Übergeben von Parametern für die Berechtigungs-ID

Ein Parameter für die Berechtigungs-ID (authid), der von DB2 an ein Plug-in (als Eingabeparameter 'authid') übergeben wird, enthält eine Berechtigungs-ID in Großbuchstaben ohne auffüllende Leerzeichen. Ein Parameter 'authid', den ein Plug-in an DB2 (als Ausgabeparameter 'authid') zurückgibt, erfordert keine besondere Behandlung. Die Berechtigungs-ID wird von DB2 entsprechend dem internen DB2-Standard in Großbuchstaben umgesetzt und mit Leerzeichen aufgefüllt.

Größenbegrenzungen für Parameter

Die Plug-in-APIs arbeiten mit folgenden Werten als Längenbegrenzungen für Parameter:

```
#define DB2SEC_MAX_AUTHID_LENGTH 255
#define DB2SEC_MAX_USERID_LENGTH 255
#define DB2SEC_MAX_USERNAMESPACE_LENGTH 255
#define DB2SEC_MAX_PASSWORD_LENGTH 255
#define DB2SEC_MAX_DBNAME_LENGTH 128
```

Eine bestimmte Plug-in-Implementierung erfordert oder erzwingt möglicherweise kleinere Maximallängen für die Berechtigungs-IDs, Benutzer-IDs und Kennwörter. Insbesondere sind die Plug-ins für die Betriebssystemauthentifizierung, die mit DB2-Datenbanksystemen geliefert werden, auf die Begrenzungen für die Maximallängen von Benutzernamen, Gruppennamen und Namensbereichen beschränkt, die durch das Betriebssystem festgelegt sind, wenn diese Betriebssystembegrenzungen niedriger sind als die oben angegebenen Werte.

Dateinamenerweiterungen für Sicherheits-Plug-in-Bibliotheken unter AIX

Auf AIX-Systemen können Sicherheits-Plug-in-Bibliotheken die Dateinamenerweiterung *.a* oder *.so* haben. Der Mechanismus, der zum Laden der Plug-in-Bibliothek verwendet wird, hängt von der Dateinamenerweiterung ab:

- Plug-in-Bibliotheken mit der Dateinamenerweiterung *.a* werden als Archive betrachtet, die gemeinsam genutzte Objektteildateien (Member) enthalten. Diese Teildateien müssen den Namen *shr.o* (32 Bit) oder *shr64.o* (64 Bit) haben. Ein einzelnes Archiv kann sowohl die 32-Bit-Teildateien als auch die 64-Bit-Teildateien enthalten, sodass es auf beiden Typen von Plattformen implementiert werden kann.

  Zum Beispiel kann eine Plug-in-Bibliothek im Stil eines 32-Bit-Archivs wie folgt erstellt werden:

  ```
  xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
  ar rv MyPlugin.a shr.o
  ```

- Plug-in-Bibliotheken mit der Dateinamenerweiterung *.so* werden als dynamisch ladbare gemeinsam genutzte Objekte betrachtet. Ein solches Objekt ist entweder ein 32- oder ein 64-Bit-Objekt, je nachdem, welche Compiler- und Linkeroptionen bei der Erstellung verwendet werden. Eine 32-Bit-Plug-in-Bibliothek kann zum Beispiel wie folgt erstellt werden:

  ```
  xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
  ```

Auf allen anderen Plattformen als AIX werden Sicherheits-Plug-in-Bibliotheken immer als dynamisch ladbare gemeinsam genutzte Objekte aufgefasst.

Einschränkungen für Sicherheits-Plug-ins

Für die Verwendung von Sicherheits-Plug-ins gelten die folgenden Einschränkungen:

Einschränkungen für die Unterstützung in der DB2-Datenbankfamilie

Sie können kein GSS-API-Plug-in zur Authentifizierung von Verbindungen zwischen DB2-Clients unter Linux, UNIX und Windows und einem anderen Server der DB2-Produktfamilie, wie zum Beispiel DB2 für z/OS, verwenden.

Darüber hinaus können auch keine Verbindungen von einem anderen Produkt der DB2-Datenbankfamilie, das als Client fungiert, zu einem DB2-Server unter Linux, UNIX oder Windows authentifizieren.

Wenn Sie einen DB2-Client unter Linux, UNIX oder Windows verwenden, um eine Verbindung zu anderen Servern der DB2-Datenbankfamilie herzustellen, können Sie clientseitige Benutzer-ID/Kennwort-Plug-ins (z. B. das von IBM bereitgestellte Plug-in zur betriebssystembasierten Authentifizierung) verwenden oder Sie können ein eigenes Benutzer-ID/Kennwort-Plug-in schreiben. Sie können auch die integrierten Kerberos-Plug-ins verwenden oder eigene implementieren.

Mit einem DB2-Client unter Linux, UNIX oder Windows sollten Sie eine Datenbank nicht mit dem Authentifizierungstyp GSSPLUGIN katalogisieren.

Einschränkungen für die Berechtigungs-ID. Ab Version 9.5 des DB2-Datenbanksystems können Sie eine 128-Byte-Berechtigungs-ID verwenden. Wenn die Berechtigungs-ID jedoch als Benutzer-ID oder Gruppenname des Betriebssystems interpretiert wird, gelten die Einschränkungen des Betriebssystems (z. B. eine Begrenzung auf 8 oder 30 Zeichen für Benutzer-IDs und auf 30 Zeichen für Gruppennamen). Daher können Sie zwar eine 128-Byte-Berechtigungs-ID erteilen, jedoch ist es nicht möglich, als Benutzer, der diese Berechtigungs-ID hat, eine Verbindung herzustellen. Wenn Sie ein eigenes Sicherheits-Plug-in schreiben, sind Sie in der Lage, die erweiterten Größen für die Berechtigungs-ID voll auszunutzen. Zum Beispiel können Sie an Ihr Sicherheits-Plug-in eine 30-Byte-Benutzer-ID übergeben, und das Plug-in kann eine 128-Byte-Berechtigungs-ID während der Authentifizierung zurückgeben, mit der Sie eine Verbindung herstellen können.

Einschränkungen der Unterstützung durch WebSphere Federation Server

DB2 II unterstützt die Verwendung delegierter Berechtigungsnachweise aus einem GSS_API-Plug-in nicht, um abgehende Verbindungen zu Datenquellen herzustellen. Verbindungen zu Datenquellen müssen weiterhin den Befehl CREATE USER MAPPING verwenden.

Einschränkungen der Unterstützung durch den Datenbankverwaltungsserver (DAS)

Der DB2-Verwaltungsserver (DAS) unterstützt Plug-ins nicht. Der DAS unterstützt nur das Authentifizierungsverfahren über das Betriebssystem.

Sicherheits-Plug-in-Problem und Einschränkung für DB2-Clients (Windows)

Wenn Sie Sicherheits-Plug-ins entwickeln, die in DB2-Clients unter Windows-Betriebssystemen implementiert werden sollen, führen Sie das Entladen Sie von Hilfsbibliotheken aus dem Speicher nicht in der Beendigungsfunktion des Plug-ins aus. Diese Einschränkung gilt für alle Typen von Client-Sicherheits-Plug-ins, einschließlich Gruppen-, Benutzer-ID/Kennwort-, Kerberos- und GSS-API-Plug-ins. Da diese Beendigungs-API wie zum Beispiel 'db2secPluginTerm', 'db2secClientAuthPluginTerm' und 'db2secServerAuthPluginTerm' auf keiner Windows-Plattform aufgerufen werden, müssen Sie die entsprechende Ressourcenbereinigung selbst durchführen.

Diese Einschränkung steht im Zusammenhang mit Bereinigungsproblemen, die mit dem Entladen von DDL-Dateien unter Windows verbunden sind.

Laden von Plug-in-Bibliotheken mit der Erweiterung .a oder .so unter AIX

Unter AIX können Sicherheits-Plug-in-Bibliotheken die Dateinamenerweiterung `.a` oder `.so` haben. Der Mechanismus, der zum Laden der Plug-in-Bibliothek verwendet wird, hängt von der Dateinamenerweiterung ab:

- Plug-in-Bibliotheken mit der Dateinamenerweiterung `.a`

  Plug-in-Bibliotheken mit der Dateinamenerweiterung .a werden als Archive betrachtet, die gemeinsam genutzte Objektteildateien (Member) enthalten. Diese Teildateien müssen den Namen `shr.o` (32 Bit) oder `shr64.o` (64 Bit) haben. Ein einzelnes Archiv kann sowohl die 32-Bit-Teildateien als auch die 64-Bit-Teildateien enthalten, sodass es auf beiden Typen von Plattformen implementiert werden kann.

  Zum Beispiel kann eine Plug-in-Bibliothek im Stil eines 32-Bit-Archivs wie folgt erstellt werden:

  ```
  xlc_r -qmkshrobj -o shr.o MyPlugin.c -bE:MyPlugin.exp
  ar rv MyPlugin.a shr.o
  ```

- Plug-in-Bibliotheken mit der Dateinamenerweiterung `.so`

  Plug-in-Bibliotheken mit der Dateinamenerweiterung .so werden als dynamisch ladbare gemeinsam genutzte Objekte betrachtet. Ein solches Objekt ist entweder ein 32- oder ein 64-Bit-Objekt, je nachdem, welche Compiler- und Linkeroptionen bei der Erstellung verwendet werden. Eine 32-Bit-Plug-in-Bibliothek kann zum Beispiel wie folgt erstellt werden:

  ```
  xlc_r -qmkshrobj -o MyPlugin.so MyPlugin.c -bE:MyPlugin.exp
  ```

Auf allen anderen Plattformen als AIX werden Sicherheits-Plug-in-Bibliotheken immer als dynamisch ladbare gemeinsam genutzte Objekte aufgefasst.

GSS-API-Sicherheits-Plug-ins unterstützen keine Verschlüsselung und Signierung von Nachrichten

Funktionen zur Verschlüsselung und Signierung von Nachrichten sind in GSS-API-Sicherheits-Plug-ins nicht verfügbar.

Rückkehrcodes für Sicherheits-Plug-ins

Alle Sicherheits-Plug-in-AIPs (API - Anwendungsprogrammierschnittstelle) müssen einen ganzzahligen Wert (Integer) zurückgeben, um Erfolg oder Fehler bei der Ausführung der API anzuzeigen. Der Rückkehrcodewert 0 gibt an, dass die API erfolgreich ausgeführt wurde. Alle negativen Rückkehrcodes mit Ausnahme von -3, -4 und -5 geben an, dass die API einen Fehler festgestellt hat.

Alle negativen Rückkehrcodes, die von den Sicherheits-Plug-in-APIs zurückgegeben werden, werden dem SQLCODE-Wert -1365, SQLCODE-Wert -1366 oder SQLCODE-Wert -30082 zugeordnet. Davon ausgenommen sind die Rückkehrcodes -3, -4 und -5. Die Werte -3, -4 und -5 dienen zur Angabe, ob eine Berechtigungs-ID einen gültigen Benutzer oder eine gültige Gruppe darstellt.

Alle Rückkehrcodes von Sicherheits-Plug-in-APIs sind in der Datei db2secPlugin.h definiert, die sich im DB2-INCLUDE-Verzeichnis befindet: SQLLIB/include.

Details in Bezug auf alle Rückkehrcodes von Sicherheits-Plug-ins werden in der folgenden Tabelle dargestellt:

Tabelle 37. Rückkehrcodes von Sicherheits-Plug-ins

Rückkehr-code	DEFINE-Wert	Bedeutung	Relevante APIs
0	DB2SEC_PLUGIN_OK	Die Plug-in-API wurde erfolgreich ausgeführt.	Alle
-1	DB2SEC_PLUGIN_UNKNOWNERROR	Die Plug-in-API hat einen unerwarteten Fehler festgestellt.	Alle
-2	DB2SEC_PLUGIN_BADUSER	Die als Eingabe übergebene Benutzer-ID ist nicht definiert.	db2secGenerateInitialCred db2secValidatePassword db2secRemapUserid db2secGetGroupsForUser
-3	DB2SEC_PLUGIN_INVALIDUSERORGROUP	Kein solcher Benutzer bzw. keine solche Gruppe vorhanden.	db2secDoesAuthIDExist db2secDoesGroupExist
-4	DB2SEC_PLUGIN_USERSTATUSNOTKNOWN	Unbekannter Benutzerstatus. Dieser Wert wird von DB2 nicht als Fehler behandelt, sondern dient bei der Verarbeitung einer Anweisung GRANT zur Ermittlung, ob eine Berechtigungs-ID (authid) einen Benutzer oder eine Betriebssystemgruppe darstellt.	db2secDoesAuthIDExist
-5	DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN	Unbekannter Gruppenstatus. Dieser Wert wird von DB2 nicht als Fehler behandelt, sondern dient bei der Verarbeitung einer Anweisung GRANT zur Ermittlung, ob eine Berechtigungs-ID (authid) einen Benutzer oder eine Betriebssystemgruppe darstellt.	db2secDoesGroupExist
-6	DB2SEC_PLUGIN_UID_EXPIRED	Benutzer-ID abgelaufen.	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-7	DB2SEC_PLUGIN_PWD_EXPIRED	Kennwort abgelaufen.	db2secValidatePassword db2GetGroupsForUser db2secGenerateInitialCred
-8	DB2SEC_PLUGIN_USER_REVOKED	Benutzer widerrufen.	db2secValidatePassword db2GetGroupsForUser
-9	DB2SEC_PLUGIN_USER_SUSPENDED	Benutzer ausgesetzt (gesperrt).	db2secValidatePassword db2GetGroupsForUser
-10	DB2SEC_PLUGIN_BADPWD	Falsches Kennwort.	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred
-11	DB2SEC_PLUGIN_BAD_NEWPASSWORD	Falsches neues Kennwort.	db2secValidatePassword db2secRemapUserid

Tabelle 37. Rückkehrcodes von Sicherheits-Plug-ins (Forts.)

Rückkehr-code	DEFINE-Wert	Bedeutung	Relevante APIs
-12	DB2SEC_PLUGIN _CHANGEPASSWORD _NOTSUPPORTED	Kennwortänderung nicht unterstützt.	db2secValidatePassword db2secRemapUserid db2secGenerateInitialCred
-13	DB2SEC_PLUGIN_NOMEM	Speicherzuordnungsversuch des Plug-ins wegen unzureichender Speicherkapazität fehlgeschlagen.	Alle
-14	DB2SEC_PLUGIN_DISKERROR	Plug-in hat einen Plattenfehler festgestellt.	Alle
-15	DB2SEC_PLUGIN_NOPERM	Versuch des Zugriffs auf eine Datei durch das Plug-in wegen falscher Berechtigungen für die Datei fehlgeschlagen.	Alle
-16	DB2SEC_PLUGIN_NETWORKERROR	Plug-in hat einen Netzwerkfehler festgestellt.	Alle
-17	DB2SEC_PLUGIN _CANTLOADLIBRARY	Plug-in kann die erforderliche Bibliothek nicht laden.	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-18	DB2SEC_PLUGIN_CANT _OPEN_FILE	Plug-in kann eine Datei nicht öffnen und lesen, wobei die Ursache weder eine fehlende Datei noch eine falsche Dateiberechtigung ist.	Alle
-19	DB2SEC_PLUGIN_FILENOTFOUND	Plug-in kann eine Datei nicht öffnen und lesen, weil die Datei im Dateisystem fehlt.	Alle
-20	DB2SEC_PLUGIN _CONNECTION_DISALLOWED	Das Plug-in verweigert den Verbindungsaufbau wegen der Einschränkung, zu welcher Datenbank eine Verbindung zulässig ist, oder die TCP/IP-Adresse kann keine Verbindung zu einer bestimmten Datenbank herstellen.	Alle serverseitigen Plug-in-APIs
-21	DB2SEC_PLUGIN_NO_CRED	Nur GSS-API-Plug-in: Anfangsberechtigungsnachweis für Client fehlt.	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-22	DB2SEC_PLUGIN_CRED_EXPIRED	Nur GSS-API-Plug-in: Clientberechtigungsnachweis ist abgelaufen.	db2secGetDefaultLoginContext db2secServerAuthPluginInit
-23	DB2SEC_PLUGIN _BAD_PRINCIPAL_NAME	Nur GSS-API-Plug-in: Der Name des Principals ist ungültig.	db2secProcessServer PrincipalName
-24	DB2SEC_PLUGIN _NO_CON_DETAILS	Dieser Rückkehrcode wird vom db2secGetConDetails-Callback (z. B. von DB2 an das Plug-in) zurückgegeben, um mitzuteilen, dass DB2 die TCP/IP-Adresse des Clients nicht ermitteln kann.	db2secGetConDetails

Tabelle 37. Rückkehrcodes von Sicherheits-Plug-ins (Forts.)

Rückkehr-code	DEFINE-Wert	Bedeutung	Relevante APIs
-25	DB2SEC_PLUGIN _BAD_INPUT_PARAMETERS	Beim Aufruf der Plug-in-API sind einige Parameter nicht gültig oder fehlen.	Alle
-26	DB2SEC_PLUGIN _INCOMPATIBLE_VER	Die vom Plug-in gemeldete Version der APIs ist nicht mit DB2 kompatibel.	db2secGroupPluginInit db2secClientAuthPluginInit db2secServerAuthPluginInit
-27	DB2SEC_PLUGIN_PROCESS_LIMIT	Nicht genügend Ressourcen für das Plug-in zur Erstellung eines neuen Prozesses verfügbar.	Alle
-28	DB2SEC_PLUGIN_NO_LICENSES	Das Plug-in hat ein Benutzerlizenzproblem festgestellt. Es besteht die Möglichkeit, dass die Lizenz des zugrunde liegenden Mechanismus ihre Obergrenze erreicht hat.	Alle

Fehlernachrichtenbehandlung für Sicherheits-Plug-ins

Wenn in einer Sicherheits-Plug-in-API ein Fehler auftritt, kann die API eine ASCII-Textzeichenfolge im Feld errormsg zurückgeben, um eine genauere Beschreibung des Problems als den Rückkehrcode bereitzustellen.

Die Zeichenfolge im Feld errormsg könnte zum Beispiel den Text "Datei /home/db2inst1/mypasswd.txt ist nicht vorhanden." enthalten. DB2 schreibt die gesamte Zeichenfolge in das DB2-Protokoll mit Benachrichtigungen für die Systemverwaltung und fügt darüber hinaus eine abgeschnittene Version als Token in einige SQL-Nachrichten ein. Da Token in SQL-Nachrichten nur eine begrenzte Länge haben können, sollten diese Nachrichten kurz gehalten werden, und wichtige Informationen dieser Nachrichten sollten im Anfang der Zeichenfolge enthalten sein. Als Debughilfe sollten Sie in Betracht ziehen, den Namen des Sicherheits-Plug-ins in die Fehlernachricht aufzunehmen.

Bei nicht dringenden Fehlern, zum Beispiel Fehlern aufgrund abgelaufener Kennwörter, wird die Zeichenfolge in errormsg nur dann in das Protokoll geschrieben, wenn der Konfigurationsparameter DIAGLEVEL des Datenbankmanagers auf den Wert 4 gesetzt ist.

Der Speicher für diese Fehlernachrichten muss vom Sicherheits-Plug-in zugeordnet werden. Daher müssen die Plug-ins auch eine API zur Freigabe dieses Speichers verwenden: db2secFreeErrormsg.

Das Feld errormsg wird von DB2 nur überprüft, wenn eine API einen Fehler ungleich null zurückgibt. Daher sollte das Plug-in keinen Speicher für diese zurückgegebene Fehlernachricht zuordnen, wenn kein Fehler aufgetreten ist.

Bei der Initialisierung wird ein Zeiger (logMessage_fn) für die Protokollierungsfunktion ('db2secLogMessage') an das Gruppen-, Client- oder Server-Plug-in übergeben. Die Plug-ins können die Funktion zum Protokollieren beliebiger Debugging-Informationen in der Datei 'db2diag.log' verwenden. Beispiel:

```
// Protokollieren einer Nachricht 'Init erfolgreich'
(*(logMessage_fn))(DB2SEC_LOG_CRITICAL,
                   "db2secGroupPluginInit erfolgreich",
                    strlen("db2secGroupPluginInit erfolgreich"));
```

Weitere detaillierte Informationen zu den Parametern für die Funktion `db2secLogMessage` finden Sie in Beschreibungen der Initialisierungs-APIs für die einzelnen Plug-in-Typen.

Aufrufreihenfolgen für die APIs der Sicherheits-Plug-ins

Im Folgenden werden die Hauptszenarios aufgeführt, in denen der DB2-Datenbankmanager APIs für Sicherheits-Plug-ins aufruft:

- Auf einem Client zur Herstellung einer Datenbankverbindung (implizit und explizit)
  - CLIENT
  - Serverbasiert (SERVER, SERVER_ENCRYPT, DATA_ENCRYPT)
  - GSSAPI und Kerberos
- Auf einem Client, Server oder Gateway zur lokalen Autorisierung
- Auf einem Server zur Herstellung einer Datenbankverbindung
- Auf einem Server für eine Anweisung GRANT
- Auf einem Server zum Abrufen einer Liste der Gruppen, zu denen eine Berechtigungs-ID gehört

Anmerkung: Die DB2-Datenbankserver behandeln Datenbankaktionen, die lokale Berechtigungen erfordern, wie zum Beispiel db2start, db2stop und db2trc, wie Clientanwendungen.

Für jede dieser Operationen ist die Reihenfolge unterschiedlich, in der der DB2-Datenbankmanager die Sicherheit-Plug-in-APIs aufruft. Im Folgenden werden die Reihenfolgen der APIs, die vom DB2-Datenbankmanager aufgerufen werden, für jedes dieser Szenarios aufgeführt.

CLIENT - implizit

Wenn der vom Benutzer konfigurierte Authentifizierungstyp CLIENT ist, ruft die DB2-Clientanwendung die folgenden Sicherheits-Plug-in-APIs auf:

- `db2secGetDefaultLoginContext();`
- `db2secValidatePassword();`
- `db2secFreetoken();`

Bei einer impliziten Authentifizierung, d. h. wenn Sie die Verbindung herstellen, ohne eine bestimmte Benutzer-ID oder Kennwort anzugeben, wird die API `db2secValidatePassword` aufgerufen, wenn Sie ein Plug-in zur Benutzer-ID/Kennwort-Authentifizierung verwenden. Diese API ermöglicht Plug-in-Entwicklern, die implizite Authentifizierung zu unterbinden, falls dies erforderlich ist.

CLIENT - explizit

Bei einer expliziten Authentifizierung, d. h., wenn Sie die Verbindung zu einer Datenbank unter Angabe der Benutzer-ID und des Kennworts herstellen, ruft die DB2-Clientanwendung, wenn der Konfigurationsparameter *authentication* des Datenbankmanagers auf den Wert CLIENT gesetzt ist, die folgenden Plug-in-APIs mehrere Male auf, falls die Implementierung dies erfordert:

- `db2secRemapUserid();`

- db2secValidatePassword();
- db2secFreeToken();

Serverbasierte Authentifizierung (SERVER, SERVER_ENCRYPT, DATA_ENCRYPT) - implizit

Bei einer impliziten Authentifizierung, wenn der Client und der Server eine Benutzer-ID/Kennwort-Authentifizierung vereinbart haben (z. B. bei Einstellung des Parameters *srvcon_auth* auf dem Server auf den Wert SERVER, SERVER_ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP), ruft die Clientanwendung die folgenden Sicherheits-Plug-in-APIs auf:

- db2secGetDefaultLoginContext();
- db2secFreeToken();

Serverbasierte Authentifizierung (SERVER, SERVER_ENCRYPT, DATA_ENCRYPT) - explizit

Bei einer expliziten Authentifizierung, wenn der Client und der Server eine Benutzer-ID/Kennwort-Authentifizierung vereinbart haben (z. B. bei Einstellung des Parameters *srvcon_auth* auf dem Server auf den Wert SERVER, SERVER_ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP), ruft die Clientanwendung die folgenden Sicherheits-Plug-in-APIs auf:

- db2secRemapUserid();

GSSAPI und Kerberos - implizit

Bei einer impliziten Authentifizierung, wenn der Client und der Server eine GSS-API- oder Kerberos-Authentifizierung vereinbart haben (z. B. bei Einstellung des Parameters *srvcon_auth* auf dem Server auf den Wert KERBEROS, KRB_SERVER_ENCRYPT, GSSPLUGIN oder GSS_SERVER_ENCRYPT), ruft die Clientanwendung die folgenden Sicherheits-Plug-in-APIs auf. (Der Aufruf der API 'gss_init_sec_context()' verwendet GSS_C_NO_CREDENTIAL als Eingabeberechtigungsnachweise.)

- db2secGetDefaultLoginContext();
- db2secProcessServerPrincipalName();
- gss_init_sec_context();
- gss_release_buffer();
- gss_release_name();
- gss_delete_sec_context();
- db2secFreeToken();

Mit der Unterstützung für mehrere GSS-API-Abläufe kann die API gss_init_sec_context() mehrfach aufgerufen werden, wenn die Implementierung dies erfordert.

GSSAPI und Kerberos - explizit

Wenn der vereinbarte Authentifizierungstyp GSS-API oder Kerberos ist, ruft die Clientanwendung die folgenden Sicherheits-Plug-in-APIs für GSS-API-Plug-ins in der angegebenen Reihenfolge auf. Diese APIs gelten für die implizite und die explizite Authentifizierung, sofern nicht anders angegeben.

- db2secProcessServerPrincipalName();
- db2secGenerateInitialCred(); (nur für explizite Authentifizierung)
- gss_init_sec_context();
- gss_release_buffer ();
- gss_release_name();
- gss_release_cred();

- db2secFreeInitInfo();
- gss_delete_sec_context();
- db2secFreeToken();

Die API gss_init_sec_context() kann mehrmals aufgerufen werden, wenn ein Token zur gegenseitigen Authentifizierung vom Server zurückgegeben wird und die Implementierung dies erfordert.

Auf einem Client, Server oder Gateway zur lokalen Autorisierung

Für eine lokale Berechtigung ruft der verwendete DB2-Befehl die folgenden Sicherheits-Plug-in-APIs auf:

- db2secGetDefaultLoginContext();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

Diese APIs werden für beide Verfahren, das der Benutzer-ID/Kennwort-Authentifizierung und das der GSS-API-Authentifzierung, aufgerufen.

Auf einem Server zur Herstellung einer Datenbankverbindung

Für eine Datenbankverbindung auf dem Datenbankserver ruft der Prozess oder Thread des DB2-Agenten die folgenden Sicherheits-Plug-in-APIs für das Benutzer-ID/Kennwort-Authentifizierungsverfahren auf:

- db2secValidatePassword(); (nur wenn der Datenbankkonfigurationsparameter *authentication* nicht den Wert CLIENT hat)
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- db2secFreeToken();
- db2secFreeGroupList();

Für eine CONNECT-Verbindung zu einer Datenbank ruft der Prozess oder Thread des DB2-Agenten die folgenden Sicherheits-Plug-in-APIs für das GSS-API-Authentifizierungsverfahren auf:

- gss_accept_sec_context();
- gss_release_buffer();
- db2secGetAuthIDs();
- db2secGetGroupsForUser();
- gss_delete_sec_context();
- db2secFreeGroupListMemory();

Auf einem Server für eine Anweisung GRANT

Für eine Anweisung GRANT, in der das Schlüsselwort USER oder GROUP nicht angegeben ist (z. B. "GRANT CONNECT ON DATABASE TO benutzer1") muss der Prozess oder Thread des DB2-Agenten feststellen können, ob es sich bei benutzer1 um einen Benutzer, um eine Gruppe oder um beides handelt. Daher ruft der Prozess bzw. Thread des DB2-Agenten die folgenden Sicherheits-Plug-in-APIs auf:

- db2secDoesGroupExist();
- db2secDoesAuthIDExist();

Auf einem Server zum Abrufen einer Liste der Gruppen, zu denen eine Berechtigungs-ID gehört

Wenn Sie von Ihrem Datenbankserver eine Liste der Gruppen abrufen müssen, zu denen eine Berechtigungs-ID gehört, ruft der Prozess oder Thread des DB2-Agenten die folgende Sicherheits-Plug-in-API nur mit der Berechtigungs-ID als Eingabe auf:

- `db2secGetGroupsForUser();`

Token aus anderen Sicherheits-Plug-ins sind nicht beteiligt.

Kapitel 8. Sicherheits-Plug-in-APIs

Das DB2-Datenbanksystem stellt APIs bereit, mit denen Sie vorhandene Plug-in-Module modifizieren bzw. neue Sicherheits-Plug-in-Module erstellen können, um die Funktionalität zur Authentifizierung und Gruppensuche Ihres DB2-Datenbanksystems an Ihre Verhältnisse anzupassen.

Wenn Sie ein Sicherheits-Plug-in-Modul entwickeln, müssen Sie die Standardfunktionen zur Authentifizierung bzw. zur Ermittlung von Gruppenzugehörigkeiten implementieren, die vom DB2 aufgerufen werden sollen. Für die drei verfügbaren Typen von Plug-in-Modulen müssen Sie die folgende Funktionalität implementieren:

Abrufen von Gruppen
: Ruft Informationen zur Gruppenzugehörigkeit für einen angegebenen Benutzer ab und bestimmt, ob eine angegebene Zeichenfolge einen gültigen Gruppennamen darstellt.

Benutzer-ID/Kennwort-Authentifizierung
: Ein Authentifizierungsverfahren, das den Standardsicherheitskontext angibt (nur Client), ein Kennwort prüft und optional ändert, feststellt, ob eine angegebene Zeichenfolge einen gültigen Benutzer darstellt (nur Server), die auf dem Client angegebene Benutzer-ID bzw. das Kennwort vor dem Senden an den Server ändert (nur Client) sowie die einem angegebenen Benutzer zugeordnete DB2-Berechtigungs-ID zurückgibt.

GSS-API-Authentifizierung
: Ein Authentifizierungsverfahren, das die erforderlichen GSS-API-Funktionen implementiert, den Standardsicherheitskontext angibt (nur Clientseite), die Anfangsberechtigungsnachweise auf der Basis der Benutzer-ID und des Kennworts generiert und optional das Kennwort ändert (nur Clientseite), Sicherheitstickets erstellt und akzeptiert sowie die einem angegebenen GSS-API-Sicherheitskontext zugeordnete DB2-Berechtigungs-ID zurückgibt.

Nachfolgend werden einige der in den Beschreibungen der Plug-in-APIs verwendeten Begriffe definiert.

Plug-in
: Eine dynamisch ladbare Bibliothek, die von DB2 geladen wird, um auf benutzerdefinierte Funktionen zur Authentifizierung und Gruppenzugehörigkeitsermittlung zuzugreifen.

Implizite Authentifizierung
: Eine Verbindung zu einer Datenbank, bei der keine Benutzer-ID bzw. kein Kennwort angegeben wird.

Explizite Authentifizierung
: Eine Verbindung zu einer Datenbank, bei der sowohl die Benutzer-ID als auch das Kennwort angegeben werden.

Berechtigungs-ID (authid)
: Eine interne Kennung, die einen Einzelbenutzer oder eine Gruppe darstellt, dem bzw. der Berechtigungen und Zugriffsrechte innerhalb der Datenbank erteilt sind. Intern wird eine DB2-Berechtigungs-ID durchgehend in Großbuchstaben umgesetzt und ist mindestens acht Zeichen lang (ggf. mit Leerzeichen auf die Länge von acht Zeichen aufgefüllt). Gegenwärtig setzt DB2

Berechtigungs-IDs, Benutzer-IDs, Kennwörter, Gruppennamen, Namensbereiche und Domänennamen voraus, die im 7-Bit-ASCII-Code darstellbar sind.

Lokale Berechtigung
Eine Berechtigungsfunktion, die in Bezug auf den Server oder Client, der sie implementiert, lokal ist und prüft, ob ein Benutzer zur Ausführung einer Aktion (abgesehen von der Herstellung einer Verbindung zur Datenbank) berechtigt ist, wie zum Beispiel zum Starten und Stoppen des Datenbankmanagers, zum Aktivieren und Inaktivieren der DB2-Tracefunktion oder zum Aktualisieren der Datenbankmanagerkonfiguration.

Namensbereich
Eine Sammlung bzw. Gruppierung von Benutzern, innerhalb deren die Kennungen für einzelne Benutzer eindeutig sein müssen. Allgemeine Beispiele sind Windows-Domänen und Kerberos-Realms. Zum Beispiel müssen innerhalb der Windows-Domäne ″usa.company.com″ alle Benutzernamen eindeutig sein. Beispiel: ″benutzer1@usa.company.com″. Dieselbe Benutzer-ID in einer anderen Domäne, zum Beispiel ″benutzer1@canada.company.com″, bezieht sich hingegen auf eine andere Person. Eine vollständig qualifizierte Benutzerkennung besteht aus einem Paar aus Benutzer-ID und Namensbereich. Beispiel: ″benutzer@domäne.name″ oder ″domäne\benutzer″.

Eingabe
Gibt an, dass DB2 den Wert für den jeweiligen API-Parameter des Sicherheits-Plug-ins bereitstellt.

Ausgabe
Gibt an, dass die Sicherheits-Plug-in-API den Wert für den API-Parameter bereitstellt.

APIs für Plug-ins zum Abrufen von Gruppen

Für das Plug-in-Modul zum Abrufen von Gruppen müssen Sie die folgenden APIs implementieren:

- `db2secGroupPluginInit`

  Anmerkung: Die Anwendungsprogrammierschnittstelle (API) 'db2secGroupPluginInit' empfängt als Eingabe den Zeiger `*logMessage_fn` auf eine API mit dem folgenden Prototyp:

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );
  ```

  Die API 'db2secLogMessage' ermöglicht dem Plug-in, Nachrichten in der Datei db2diag.log zu Debug- oder Informationszwecken zu protokollieren. Diese API wird vom DB2-Datenbanksystem bereitgestellt, sodass Sie sie nicht zu implementieren brauchen.
- `db2secPluginTerm`
- `db2secGetGroupsForUser`
- `db2secDoesGroupExist`
- `db2secFreeGroupListMemory`
- `db2secFreeErrormsg`

- Die einzige API, die extern auflösbar sein muss, ist die API db2secGroupPluginInit. Diese API empfängt den Parameter void *, der in den folgenden Typ umgesetzt werden muss:

```
typedef struct db2secGroupFunctions_1
{
db2int32 version;
db2int32 plugintype;
SQL_API_RC (SQL_API_FN * db2secGetGroupsForUser)
(
const char *authid,
db2int32    authidlen,
const char *userid,
db2int32    useridlen,
const char *usernamespace,
db2int32    usernamespacelen,
db2int32    usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
const void *token,
db2int32    tokentype,
db2int32    location,
const char *authpluginname,
db2int32    authpluginnamelen,
void      **grouplist,
db2int32   *numgroups,
char    **errormsg,
db2int32 *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secDoesGroupExist)
(
const char *groupname,
db2int32    groupnamelen,
char    **errormsg,
db2int32 *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeGroupListMemory)
(
void     *ptr,
char    **errormsg,
db2int32 *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *msgtobefree
);

SQL_API_RC (SQL_API_FN * db2secPluginTerm)
(
char    **errormsg,
db2int32 *errormsglen
);

} db2secGroupFunctions_1;
```

Die API db2secGroupPluginInit weist die Adressen für die übrigen extern verfügbaren Funktionen zu.

Anmerkung: Die Erweiterung _1 gibt an, dass dies die Struktur ist, die der Version 1 der API entspricht. Nachfolgende Schnittstellenversionen werden die Erweiterungen _2, _3 usw. haben.

db2secDoesGroupExist (API) - Vorhandensein einer Gruppe überprüfen

Stellt fest, ob eine Berechtigungs-ID (authid) eine Gruppe darstellt.

Wenn der Gruppenname vorhanden ist, muss die API den Wert DB2SEC_PLUGIN_OK zurückgeben können, um den Erfolg zu melden. Sie muss außerdem den Wert DB2SEC_PLUGIN_INVALIDUSERORGROUP zurückgeben können, wenn der Gruppenname nicht gültig ist. Es ist zulässig, dass die API den Wert DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN zurückgibt, wenn sich nicht feststellen lässt, ob die Eingabe eine gültige Gruppe angibt. Wenn der Wert für ungültige Gruppe (DB2SEC_PLUGIN_INVALIDUSERORGROUP) oder für unbekannte Gruppe (DB2SEC_PLUGIN_GROUPSTATUSNOTKNOWN) zurückgegeben wird, kann DB2 möglicherweise nicht feststellen, ob die Berechtigungs-ID eine Gruppe oder einen Benutzer darstellt, wenn die Anweisung GRANT ohne das Schlüsselwort USER oder GROUP abgesetzt wird. In einem solchen Fall wird ein Fehler mit dem SQLCODE-Wert -569 und dem SQLSTATE-Wert 56092 an den Benutzer zurückgegeben.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secDoesGroupExist)
                      ( const char *groupname,
                        db2int32 groupnamelen,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secDoesGroupExist'

groupname
Eingabe. Eine in Großbuchstaben umgesetzte Berechtigungs-ID (authid) ohne folgende Leerzeichen.

groupnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'groupname'.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secDoesGroupExist' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secFreeErrormsg (API) - Speicher für Fehlernachricht freigeben

Gibt den Speicher frei, der zur Aufnahme einer Fehlernachricht aus einem früheren API-Aufruf verwendet wurde. Dies ist die einzige API, die keine Fehlernachricht zurückgibt. Wenn diese API einen Fehler zurückgibt, protokolliert DB2 diesen Fehler und setzt die Verarbeitung fort.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secFreeErrormsg)
                      ( char *errormsg );
```

Parameter der API 'db2secFreeErrormsg'

msgtofree
Eingabe. Ein Zeiger auf den Speicher der Fehlernachricht, der von einem früheren API-Aufruf zugeordnet wurde.

db2secFreeGroupListMemory (API) - Speicher für Gruppenliste freigeben

Gibt den Speicher frei, der zur Aufnahme der Liste von Gruppen aus einem früheren Aufruf der API 'db2secGetGroupsForUser' verwendet wurde.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secFreeGroupListMemory)
                      ( void *ptr,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secFreeGroupListMemory'

ptr Eingabe. Zeiger auf den freizugebenden Speicher.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secFreeGroupListMemory' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secGetGroupsForUser (API) - Liste von Gruppen für Benutzer abrufen

Gibt eine Liste von Gruppen zurück, zu denen ein Benutzer gehört.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secGetGroupsForUser)
                      ( const char *authid,
                        db2int32 authidlen,
                        const char *userid,
                        db2int32 useridlen,
                        const char *usernamespace,
                        db2int32 usernamespacelen,
                        db2int32 usernamespacetype,
                        const char *dbname,
                        db2int32 dbnamelen,
                        void *token,
                        db2int32 tokentype,
                        db2int32 location,
                        const char *authpluginname,
                        db2int32 authpluginnamelen,
                        void      **grouplist,
                        db2int32 *numgroups,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secGetGroupsForUser'

authid Eingabe. Dieser Parameterwert ist eine SQL-Berechtigungs-ID. Dies bedeutet, dass sie von DB2 in eine Zeichenfolge in Großbuchstaben ohne folgende Leerzeichen konvertiert wird. DB2 stellt immer einen Nichtnullwert für den Parameter 'authid' bereit. Die API muss eine Liste von Gruppen zurückgeben können, zu denen die Berechtigungs-ID gehört, ohne von den anderen Eingabeparametern abhängig zu sein. Es ist zulässig, eine verkürzte oder leere Liste zuzurückzugeben, wenn sich keine Gruppen bestimmen lassen.

Wenn ein Benutzer nicht vorhanden ist, muss die API den Rückkehrcode DB2SEC_PLUGIN_BADUSER zurückgeben. DB2 behandelt den Fall eines nicht vorhandenen Benutzers nicht als Fehler, da es zulässig ist, wenn eine Berechtigungs-ID keine zugeordneten Gruppen besitzt. Zum Beispiel kann die API 'db2secGetAuthids' eine Berechtigungs-ID zurückgeben, die im Betriebssystem nicht vorhanden ist. Die Berechtigungs-ID ist zwar keinen Gruppen zugeordnet, dennoch können ihr Zugriffsrechte direkt erteilt werden.

Wenn die API nur anhand der Berechtigungs-ID keine vollständige Liste der Gruppen zurückgeben kann, gelten einige Einschränkungen für bestimmte SQL-Funktionen in Bezug auf die Unterstützung von Gruppen. Eine Liste der möglichen Problemszenarios finden Sie unter der Überschrift 'Hinweise' in diesem Abschnitt.

authidlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'authid'. Der DB2-Datenbankmanager stellt immer einen Nichtnullwert für den Parameter 'authidlen' bereit.

userid Eingabe. Dieser Wert gibt die Benutzer-ID an, die der Berechtigungs-ID (authid) entspricht. Wenn diese API auf dem Server in einem Szenario ohne Verbindung aufgerufen wird, wird dieser Parameter von DB2 nicht gefüllt.

useridlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

usernamespace
Eingabe. Der Namensbereich, aus dem die Benutzer-ID abgerufen wurde. Wenn die Benutzer-ID nicht verfügbar ist, wird dieser Parameter vom DB2-Datenbankmanager nicht gefüllt.

usernamespacelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'.

usernamespacetype
Eingabe. Der Typ des Namensbereichs. Folgende Werte für den Parameter 'usernamespacetype' (in der Datei 'db2secPlugin.h' definiert) sind gültig:
- DB2SEC_NAMESPACE_SAM_COMPATIBLE: Entspricht einer Benutzernamensdarstellung im Format 'domäne\meinname'.
- DB2SEC_NAMESPACE_USER_PRINCIPAL: Entspricht einer Benutzernamensdarstellung im Format 'meinname@domäne.ibm.com'.

Gegenwärtig unterstützt das DB2-Datenbanksystem nur den Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE. Wenn die Benutzer-ID nicht verfügbar ist, wird der Parameter 'usernamespacetype' auf den Wert DB2SEC_USER_NAMESPACE_UNDEFINED (in der Datei 'db2secPlugin.h' definiert) gesetzt.

dbname
Eingabe. Der Name der Datenbank, zu der die Verbindung hergestellt wird. Dieser Parameter kann in einem Szenario ohne Verbindung den Wert NULL haben.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'. Dieser Parameter wird auf den Wert 0 gesetzt, wenn der Parameter 'dbname' den Wert NULL in einem Szenario ohne Verbindung hat.

token Eingabe. Ein Zeiger auf Daten, die vom Authentifizierungs-Plug-in bereitgestellt werden. Dieser Parameter wird von DB2 nicht verwendet. Er bietet dem Autor des Plug-ins die Möglichkeit, Benutzer- und Gruppeninformationen zu koordinieren. Dieser Parameter wird möglicherweise nicht in allen Fällen bereitgestellt (z. B. in einem Szenario ohne Verbindung). In einem solchen Fall hat er den Wert NULL. Wenn das verwendete Authentifizierungs-Plug-in auf der GSS-API basiert, wird das Token auf die Kontextkennung (gss_ctx_id_t) der GSS-API gesetzt.

tokentype
Eingabe. Gibt den Typ von Daten an, die vom Authentifizierungs-Plug-in bereitgestellt werden. Wenn das verwendete Authentifizierungs-Plug-in auf der GSS-API basiert, wird das Token auf die Kontextkennung (gss_ctx_id_t) der GSS-API gesetzt. Wenn das verwendete Authentifizierungs-Plug-in auf Benutzer-ID und Kennwort basiert, gibt dieser Parameter einen generischen Typ an. Folgende Werte für den Parameter 'tokentype' (in der Datei 'db2secPlugin.h' definiert) sind gültig:

- `DB2SEC_GENERIC`: Gibt an, dass das Token von einem Plug-in auf Benutzer-ID/Kennwort-Basis stammt.
- `DB2SEC_GSSAPI_CTX_HANDLE`: Gibt an, dass das Token von einem Plug-in auf GSS-API-Basis (einschließlich Kerberos) stammt.

location
Eingabe. Gibt an, ob DB2 diese API auf der Clientseite oder der Serverseite aufruft. Folgende Werte für den Parameter 'location' (in der Datei 'db2secPlugin.h' definiert) sind gültig:

- `DB2SEC_SERVER_SIDE`: Die API ist auf dem Datenbankserver aufzurufen.
- `DB2SEC_CLIENT_SIDE`: Die API ist auf einem Client aufzurufen.

authpluginname
Eingabe. Der Name des Authentifizierungs-Plug-ins, das die Daten im Token bereitgestellt hat. Die API 'db2secGetGroupsForUser' könnte diese Informationen zur Feststellung der korrekten Gruppenzugehörigkeiten verwenden. Dieser Parameter wird von DB2 möglicherweise nicht gefüllt, wenn die Berechtigungs-ID (authid) nicht authentifiziert wurde (z. B. wenn die Berechtigungs-ID nicht dem momentan verbundenen Benutzer entspricht).

authpluginnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'authpluginname'.

grouplist
Ausgabe. Die Liste der Gruppen, zu denen der Benutzer gehört. Die Liste der Gruppen muss als Zeiger auf eine Speichersektion zurückgegeben werden, die vom Plug-in zugeordnet wird und verkettete VARCHAR-Werte enthält. (Ein VARCHAR-Wert ist eine Zeichenfeldgruppe, in der das erste Byte die Anzahl der darauf folgenden Byte angibt.) Die Länge ist ein Zeichen (CHAR) ohne Vorzeichen (1 Byte). Dies begrenzt die maximale Länge

eines Gruppennamens auf 255 Zeichen. Beispiel: "\006GROUP1\
007MYGROUP\008MYGROUP3". Jeder Gruppenname muss eine gültige DB2-Berechtigungs-ID (authid) sein. Der Speicher für diese Feldgruppe muss vom Plug-in zugeordnet werden. Das Plug-in muss deshalb eine API angeben, zum Beispiel die API 'db2secFreeGroupListMemory', die DB2 aufruft, um den Speicher freizugeben.

numgroups
Ausgabe. Die Anzahl der Gruppen, die im Parameter 'grouplist' enthalten sind.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secGetGroupsForUser' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

Hinweise

In der folgenden Liste werden Problemszenarios aufgeführt, die auftreten können, wenn von dieser API eine unvollständige Liste an DB2 zurückgegeben wird:

- Anwendung mit eingebettetem SQL und DYNAMICRULES BIND (bzw. DEFINEDBIND oder INVOKEDBIND, wenn die Pakete als eigenständige Anwendung ausgeführt werden): DB2 prüft auf SYSADM-Zugehörigkeit, und die Anwendung schlägt fehl, wenn sie von der impliziten Erteilung der Berechtigung DBADM durch die Mitgliedschaft in der Gruppe SYSADM abhängig ist.
- Durch die Anweisung CREATE SCHEMA bereitgestellte alternative Berechtigung: Die Gruppensuchfunktion wird am Parameter für den Berechtigungsnamen (AUTHORIZATION NAME) ausgeführt, wenn verschachtelte Anweisungen CREATE in der Anweisung CREATE SCHEMA enthalten sind.
- Anwendungen mit eingebettetem SQL mit DYNAMICRULES DEFINERUN/DEFINEBIND, wobei die Pakete in einem Routinenkontext ausgeführt werden: DB2 prüft auf SYSADM-Zugehörigkeit des Benutzers, der die Routine definiert hat, und die Anwendung schlägt fehl, wenn sie von der impliziten Erteilung der Berechtigung DBADM durch die Mitgliedschaft in der Gruppe SYSADM abhängig ist.
- Verarbeitung einer JAR-Datei in einer Umgebung mit exklusiver Parallelverarbeitung (MPP): In einer MPP-Umgebung werden die Anforderungen zur JAR-Verarbeitung vom Koordinatorknoten mit der Sitzungsberechtigungs-ID gesendet. Der Katalogknoten empfängt die Anforderungen und verarbeitet die JAR-Dateien auf der Basis der Berechtigung der Sitzungsberechtigungs-ID (d. h. des Benutzers, der die Anforderungen zur JAR-Verarbeitung ausführt).
  - Installieren einer JAR-Datei: Die Sitzungsberechtigungs-ID muss über eine der folgenden Berechtigungen verfügen: SYSADM, DBADM oder CREATEIN (implizit oder explizit für das JAR-Schema). Die Operation schlägt fehl, wenn die oben genannten Berechtigungen einer Gruppe erteilt sind, die die Sitzungsberechtigungs-ID enthält, jedoch nicht explizit der Sitzungsberechtigungs-ID, oder wenn nur SYSADM erteilt ist, da die SYSADM-Zugehörigkeit durch die Mitgliedschaft in der Gruppe bestimmt wird, die durch einen Datenbankkonfigurationsparameter definiert wird.

– Entfernen einer JAR-Datei: Die Sitzungsberechtigungs-ID muss über eine der folgenden Berechtigungen verfügen: SYSADM, DBADM oder DROPIN (implizit oder explizit für das JAR-Schema), oder muss der Benutzer sein, der die JAR-Datei definiert hat. Die Operation schlägt fehl, wenn die oben genannten Berechtigungen einer Gruppe erteilt sind, die die Sitzungsberechtigungs-ID enthält, jedoch nicht explizit der Sitzungsberechtigungs-ID, und die Sitzungsberechtigungs-ID nicht der Benutzer ist, der die JAR-Datei definiert hat, oder wenn nur SYSADM erteilt ist, da die SYSADM-Zugehörigkeit durch die Mitgliedschaft in der Gruppe bestimmt wird, die durch einen Datenbankkonfigurationsparameter definiert wird.
– Ersetzen einer JAR-Datei: Dies ist dasselbe wie das Entfernen einer JAR-Datei mit anschließendem Installieren einer JAR-Datei. Beide Hinweise sind zu beachten.

- Erneutes Generieren von Sichten: Diese Operation wird durch die Anweisungen ALTER TABLE, ALTER COLUMN, SET DATA TYPE VARCHAR/VARGRAPHIC oder während einer Migration ausgelöst. Der DB2-Datenbankmanager prüft auf SYSADM-Zugehörigkeit des Benutzers, der die Sicht definiert. Die Anwendung schlägt fehl, wenn sie von der impliziten Erteilung der Berechtigung DBADM durch die Mitgliedschaft in der Gruppe SYSADM abhängig ist.
- Absetzen der Anweisung SET SESSION_USER: Nachfolgende DB2-Operationen werden unter dem Kontext der durch diese Anweisung angegebenen Berechtigungs-ID ausgeführt. Diese Operationen schlagen fehl, wenn die erforderlichen Berechtigungen einer der Gruppen des Sitzungsbenutzers (SESSION_USER) und nicht explizit der Berechtigungs-ID des Sitzungsbenutzers erteilt sind.

db2secGroupPluginInit (API) - Gruppen-Plug-in initialisieren

Initialisierungs-API für das Plug-in zum Abrufen von Gruppen, das der DB2-Datenbankmanager unmittelbar nach dem Laden des Plug-ins aufruft.

API- und Datenstruktursyntax

```
SQL_API_RC SQL_API_FN db2secGroupPluginInit
                              ( db2int32 version,
                                void *group_fns,
                                db2secLogMessage *logMessage_fn,
                                char     **errormsg,
                                db2int32 *errormsglen );
```

Parameter der API 'db2secGroupPluginInit'

version

Eingabe. Die höchste Version der API, die von der Instanz unterstützt wird, die dieses Plug-in lädt. Der Wert DB2SEC_API_VERSION (in 'db2secPlugin.h' definiert) enthält die aktuellste Versionsnummer der API, die vom DB2-Datenbankmanager gegenwärtig unterstützt wird.

group_fns

Ausgabe. Ein Zeiger auf die Struktur 'db2secGroupFunctions_<versionsnummer> (auch als 'group_functions_<versionsnummer>' bezeichnet). Die Struktur 'db2secGroupFunctions_<versionsnummer>' enthält Zeiger auf die APIs, die für das Plug-in zum Abrufen von Gruppen implementiert sind. In Zukunft kann es andere Versionen der APIs geben (z. B. 'db2secGroupFunctions_<versionsnummer>'), sodass der Parameter 'group_fns' als Zeiger auf die Struktur 'db2secGroupFunctions_<versionsnummer>' umgesetzt wird, die der Version entspricht, die vom Plug-in implementiert wird. Der erste Parameter

der Struktur 'group_functions_<versionsnummer>' teilt DB2 die Version der APIs mit, die vom Plug-in implementiert werden. Hinweis: Die Umsetzung (CAST) erfolgt nur, wenn die DB2-Version höher oder gleich der Version der APIs ist, die vom Plug-in implementiert werden. Die Versionsnummer stellt die Version der APIs dar, die von dem Plug-in implementiert werden. Der Parameter 'pluginType' sollte auf den Wert DB2SEC_PLUGIN_TYPE_GROUP gesetzt werden.

logMessage_fn
Eingabe. Ein Zeiger auf die API 'db2secLogMessage', die vom DB2-Datenbanksystem implementiert wird. Die API 'db2secGroupPluginInit' kann die API 'db2secLogMessage' aufrufen, um Nachrichten in der Datei 'db2diag.log' zu Debug- oder Informationszwecken zu protokollieren. Der erste Parameter ('level') der API 'db2secLogMessage' gibt den Typ der zu diagnostizierenden Fehler an, die in der Datei 'db2diag.log' aufgezeichnet werden. Die beiden letzten Parameter geben die Nachrichtenzeichenfolge bzw. ihre Länge an. Die folgenden Werte sind für den ersten Parameter der API 'dbesecLogMessage' (in der Datei 'db2secPlugin.h' definiert) gültig:

- `DB2SEC_LOG_NONE`: (0) - Keine Protokollierung
- `DB2SEC_LOG_CRITICAL`: (1) - Schwer wiegender Fehler
- `DB2SEC_LOG_ERROR`: (2) - Fehler
- `DB2SEC_LOG_WARNING`: (3) - Warnung
- `DB2SEC_LOG_INFO`: (4) Informativ

Der Nachrichtentext wird in der Datei 'db2diag.log' nur gezeigt, wenn der Wert des Parameters 'level' der API 'db2secLogMessage' kleiner oder gleich dem Wert des Konfigurationsparameters 'diaglevel' des Datenbankmanagers ist. Wenn Sie zum Beispiel den Wert `DB2SEC_LOG_INFO` verwenden, wird der Nachrichtentext in der Datei 'db2diag.log' nur gezeigt, wenn der Konfigurationsparameter 'diaglevel' des Datenbankmanagers auf den Wert 4 gesetzt ist.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secGroupPluginInit' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secPluginTerm - Gruppen-Plug-in-Ressourcen bereinigen

Gibt die Ressourcen frei, die vom Plug-in zum Abrufen der Gruppen verwendet wurden.

Diese Anwendungsprogrammierschnittstelle (API) wird vom DB2-Datenbankmanager kurz vor dem Entladen des Plug-ins zum Gruppenabruf aufgerufen. Sie sollte in einer Weise implementiert werden, in der sie eine ordnungsgemäße Bereinigung aller Ressourcen ausführt, die von der Bibliothek des Plug-ins genutzt werden. Sie sollte zum Beispiel den vom Plug-in zugeordneten Speicher freigeben, Dateien schließen, die noch geöffnet sind, und Netzwertverbindungen schließen. Es liegt in der Zuständigkeit des Plug-ins, diese Ressourcen zu verfolgen, um sie freigeben zu können. Diese API wird auf keiner Windows-Plattform aufgerufen.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secPluginTerm)
                      ( char      **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secPluginTerm'

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secPluginTerm' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

APIs für Plug-ins zur Benutzer-ID/Kennwort-Authentifizierung

Für das Plug-in-Modul zur Benutzer-ID/Kennwort-Authentifizierung müssen Sie die folgenden clientseitigen Anwendungsprogrammierschnittstellen (APIs) implementieren:

- `db2secClientAuthPluginInit`

  Anmerkung: Die API db2secClientAuthPluginInit API empfängt als Eingabe den Zeiger `*logMessage_fn` auf eine API mit dem folgenden Prototyp:

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );
  ```

  Die API 'db2secLogMessage' ermöglicht dem Plug-in, Nachrichten in der Datei db2diag.log zu Debug- oder Informationszwecken zu protokollieren. Diese API wird vom DB2-Datenbanksystem bereitgestellt, sodass Sie sie nicht zu implementieren brauchen.
- `db2secClientAuthPluginTerm`
- `db2secGenerateInitialCred` (nur für 'gssapi' verwendet)
- `db2secRemapUserid` (optional)
- `db2secGetDefaultLoginContext`
- `db2secValidatePassword`
- `db2secProcessServerPrincipalName` (nur für GSS-API)
- `db2secFreeToken` (Funktionen zur Freigabe von Speicher, der von der DLL-Datei belegt wird)
- `db2secFreeErrormsg`
- `db2secFreeInitInfo`
- Die einzige API, die extern auflösbar sein muss, ist die API `db2secClientAuthPluginInit`. Diese API empfängt den Parameter `void *`, der auf eine der folgenden Arten umgesetzt werden muss:

  ```
  typedef struct db2secUseridPasswordClientAuthFunctions_1
  {
  db2int32 version;
  db2int32 plugintype;
  ```

```
SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void     **token,
char         **errormsg,
db2int32      *errormsglen
);
/* Optional */
SQL_API_RC (SQL_API_FN * db2secRemapUserid)
(
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
char        password[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *passwordlen,
char        newpassword[DB2SEC_MAX_PASSWORD_LENGTH],
db2int32   *newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secValidatePassword)
(
const char *userid,
db2int32       useridlen,
const char    *usernamespace,
db2int32       usernamespacelen,
db2int32    usernamespacetype,
const char    *password,
db2int32       passwordlen,
const char *newpassword,
db2int32       newpasswordlen,
const char *dbname,
db2int32    dbnamelen,
db2Uint32   connection_details,
void     **token,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void     **token,
char     **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *errormsg
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
```

```
(
char     **errormsg,
db2int32      *errormsglen
);
}
```

oder

```
typedef struct db2secGssapiClientAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

SQL_API_RC (SQL_API_FN * db2secGetDefaultLoginContext)
(
char        authid[DB2SEC_MAX_AUTHID_LENGTH],
db2int32   *authidlen,
char        userid[DB2SEC_MAX_USERID_LENGTH],
db2int32   *useridlen,
db2int32    useridtype,
char        usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
db2int32   *usernamespacelen,
db2int32   *usernamespacetype,
const char *dbname,
db2int32    dbnamelen,
void     **token,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secProcessServerPrincipalName)
(
const void *data,
gss_name_t *gssName,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secGenerateInitialCred)
(
const char    *userid,
db2int32       useridlen,
const char    *usernamespace,
db2int32       usernamespacelen,
db2int32       usernamespacetype,
const char    *password,
db2int32       passwordlen,
const char    *newpassword,
db2int32       newpasswordlen,
const char    *dbname,
db2int32       dbnamelen,
gss_cred_id_t *pGSSCredHandle,
void         **initInfo,
char         **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeToken)
(
void      *token,
char     **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)
(
char *errormsg
```

```
);


SQL_API_RC (SQL_API_FN * db2secFreeInitInfo)
(
void      *initInfo,
char     **errormsg,
db2int32      *errormsglen
);

SQL_API_RC (SQL_API_FN * db2secClientAuthPluginTerm)
(
char     **errormsg,
db2int32      *errormsglen
);

/* GSS-API-spezifische Funktionen
   Parameterliste: siehe db2secPlugin.h */
OM_uint32 (SQL_API_FN * gss_init_sec_context )(<parameterliste>);
  OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameterliste>);
  OM_uint32 (SQL_API_FN * gss_display_status )(<parameterliste>);
  OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameterliste>);
  OM_uint32 (SQL_API_FN * gss_release_cred )(<parameterliste>);
  OM_uint32 (SQL_API_FN * gss_release_name )(<parameterliste>);
  }
```

Sie sollten die Struktur db2secUseridPasswordClientAuthFunctions_1 verwenden, wenn Sie ein Plug-in zur Benutzer-ID/Kennwort-Authentifizierung schreiben. Wenn Sie ein GSS-API-Plug-in (einschließlich Kerberos) schreiben, sollten Sie die Struktur db2secGssapiClientAuthFunctions_1 verwenden.

Für die Plug-in-Bibliothek zur Benutzer-ID/Kennwort-Authentifizierung müssen Sie die folgenden serverseitigen APIs implementieren:

- db2secServerAuthPluginInit

  Die API 'db2secServerAuthPluginInit' empfängt als Eingabe den Zeiger *logMessage_fn auf die API 'db2secLogMessage' sowie den Zeiger *getConDetails_fn auf die API 'db2secGetConDetails' mit den folgenden Prototypen:

  ```
  SQL_API_RC (SQL_API_FN db2secLogMessage)
  (
  db2int32 level,
  void    *data,
  db2int32 length
  );

  SQL_API_RC (SQL_API_FN db2secGetConDetails)
  (
  db2int32    conDetailsVersion,
  const void *pConDetails
  );
  ```

  Die API 'db2secLogMessage' ermöglicht dem Plug-in, Nachrichten in der Datei db2diag.log zu Debug- oder Informationszwecken zu protokollieren. Die API 'db2secGetConDetails' ermöglicht dem Plug-in das Abrufen von Details zu dem Client, der versucht, eine Datenbankverbindung herzustellen. Sowohl die API 'db2secLogMessage' als auch die API 'db2secGetConDetails' werden vom DB2-Datenbanksystem bereitgestellt, sodass Sie sie nicht zu implementieren brauchen. Die API 'db2secGetConDetails' empfängt als zweiten Parameter (pConDetails) wiederum einen Zeiger auf eine der folgenden Strukturen:

  db2sec_con_details_1:

```
typedef struct db2sec_con_details_1
{
  db2int32  clientProtocol;
  db2Uint32 clientIPAddress;
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char      dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
} db2sec_con_details_1;
```

db2sec_con_details_2:

```
typedef struct db2sec_con_details_2
{
  db2int32  clientProtocol;     /* Siehe SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress;    /* Gesetzt, wenn Protokoll = TCPIP4 */
  db2Uint32 connect_info_bitmap;
  db2int32  dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Gesetzt, wenn Protokoll = TCPIP6 */
} db2sec_con_details_2;
```

db2sec_con_details_3:

```
typedef struct db2sec_con_details_3
{
  db2int32  clientProtocol;     /* Siehe SQL_PROTOCOL_ in sqlenv.h */
  db2Uint32 clientIPAddress; /* Gesetzt, wenn Protokoll = TCPIP4 */
  db2Uint32 connect_info_bitmap;
  db2int32 dbnameLen;
  char dbname[DB2SEC_MAX_DBNAME_LENGTH + 1];
  db2Uint32 clientIP6Address[4];/* Gesetzt, wenn Protokoll = TCPIP6 */
  db2Uint32 clientPlatform; /* SQLM_PLATFORM_* aus sqlmon.h */
  db2Uint32 _reserved[16];
} db2sec_con_details_3;
```

Die möglichen Werte für `conDetailsVersion` sind DB2SEC_CON_DETAILS_VERSION_1, DB2SEC_CON_DETAILS_VERSION_2 und DB2SEC_CON_DETAILS_VERSION_3, die die Version der API darstellen.

Anmerkung: Wenn Sie 'db2sec_con_details_1', 'db2sec_con_details_2' oder 'db2sec_con_details_3' verwenden, beachten Sie folgende Hinweise:

– Vorhandene Plug-ins, die die Struktur 'db2sec_con_details_1' und den Wert DB2SEC_CON_DETAILS_VERSION_1 verwenden, funktionieren weiterhin wie mit Version 8.2, wenn sie die API 'db2GetConDetails' aufrufen. Wenn diese API auf einer IPv4-Plattform aufgerufen wird, wird die IP-Adresse des Clients im Feld 'clientIPAddress' der Struktur zurückgegeben. Wenn diese API auf einer IPv6-Plattform aufgerufen wird, wird im Feld 'clientIPAddress' der Wert 0 zurückgegeben. Zum Abrufen der IP-Adresse des Clients auf einer IPv6-Plattform muss der Code des Sicherheits-Plug-ins so geändert werden, dass er entweder die Struktur 'db2sec_con_details_2' und den Wert DB2SEC_CON_DETAILS_VERSION_2 oder die Struktur 'db2sec_con_details_3' und den Wert DB2SEC_CON_DETAILS_VERSION_3 verwendet.

– Neue Plug-ins sollten die Struktur 'db2sec_con_details_3' und den Wert DB2SEC_CON_DETAILS_VERSION_3 verwenden. Wenn die API 'db2secGetConDetails' auf einer IPv4-Plattform aufgerufen wird, wird die IP-Adresse des Clients im Feld 'clientIPAddress' der Struktur 'db2sec_con_details_3' zurückgegeben. Wenn die API auf einer IPv6-Plattform aufgerufen wird, wird
die IP-Adresse des Clients im Feld 'clientIP6Address' der Struktur 'db2sec_con_details_3' zurückgegeben. Das Feld *clientProtocol*
der Struktur für die Verbindungsdetails wird auf den Wert SQL_PROTOCOL_TCPIP (für IPv4 bei Version 1 der Struktur), SQL_PROTOCOL_TCPIP4

(für IPv4 bei Version 2 der Struktur) oder SQL_PROTOCOL_TCPIP6 (für IPv6 bei Version 2 oder Version 3 der Struktur) gesetzt.

– Die Struktur 'db2sec_con_details_3' stimmt mit der Struktur 'db2sec_con_details_2' überein, enthält jedoch ein zusätzliches Feld (*clientPlatform*), das den Typ der Clientplattform (wie er durch die Übertragungsschicht gemeldet wird) mithilfe der Plattformtypkonstanten angibt, die in 'sqlmon.h' definiert sind (z. B. SQLM_PLATFORM_AIX).

- `db2secServerAuthPluginTerm`
- `db2secValidatePassword`
- `db2secGetAuthIDs`
- `db2secDoesAuthIDExist`
- `db2secFreeToken`
- `db2secFreeErrormsg`
- Die einzige API, die extern auflösbar sein muss, ist die API `db2secServerAuthPluginInit`. Diese API empfängt den Parameter `void *`, der auf eine der folgenden Arten umgesetzt werden muss:

```
typedef struct db2secUseridPasswordServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;

  /* Parameterlisten aus Gründen der Lesbarkeit nicht angegeben.
     Parameter: siehe oben. */
SQL_API_RC (SQL_API_FN * db2secValidatePassword)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameterliste);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secFreeToken)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();
} userid_password_server_auth_functions;
```

oder

```
typedef struct db2secGssapiServerAuthFunctions_1
{
db2int32 version;
db2int32 plugintype;
gss_buffer_desc serverPrincipalName;
gss_cred_id_t ServerCredHandle;
SQL_API_RC (SQL_API_FN * db2secGetAuthIDs)(<parameterliste);
SQL_API_RC (SQL_API_FN * db2secDoesAuthIDExist)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secFreeErrormsg)(<parameterliste>);
SQL_API_RC (SQL_API_FN * db2secServerAuthPluginTerm)();

/* GSS-API-spezifische Funktionen:
   Parameterliste: siehe db2secPlugin.h */
OM_uint32 (SQL_API_FN * gss_accept_sec_context )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_display_name )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_delete_sec_context )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_display_status )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_release_buffer )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_release_cred )(<parameterliste>);
OM_uint32 (SQL_API_FN * gss_release_name )(<parameterliste>);

} gssapi_server_auth_functions;
```

Sie sollten die Struktur `db2secUseridPasswordServerAuthFunctions_1` verwenden, wenn Sie ein Plug-in zur Benutzer-ID/Kennwort-Authentifizierung schreiben. Wenn Sie ein GSS-API-Plug-in (einschließlich Kerberos) schreiben, sollten Sie die Struktur `db2secGssapiServerAuthFunctions_1` verwenden.

db2secClientAuthPluginInit (API) - Plug-in zur Clientauthentifizierung initialisieren

Initialisierungs-API für das Plug-in zur Clientauthentifizierung, das der DB2-Datenbankmanager unmittelbar nach dem Laden des Plug-ins aufruft.

API- und Datenstruktursyntax

```
SQL_API_RC SQL_API_FN db2secClientAuthPluginInit
                              ( db2int32 version,
                                void *client_fns,
                                db2secLogMessage *logMessage_fn,
                                char    **errormsg,
                                db2int32 *errormsglen );
```

Parameter der API 'db2secClientAuthPluginInit'

version

Eingabe. Die höchste Versionsnummer der API, die vom DB2-Datenbankmanager gegenwärtig unterstützt wird. Der Wert DB2SEC_API_VERSION (in 'db2secPlugin.h' definiert) enthält die aktuellste Versionsnummer der API, die von DB2 gegenwärtig unterstützt wird.

client_fns

Ausgabe. Ein Zeiger auf Speicher, der vom DB2-Datenbankmanager für eine Struktur 'db2secGssapiClientAuthFunctions_<versionsnummer>' (auch als 'gssapi_client_auth_functions_<versionsnummer>' bekannt), wenn die GSS-API-Authentifizierung verwendet wird, oder für eine Struktur 'db2secUseridPasswordClientAuthFunctions_<versionsnummer>' (auch als 'userid_password_client_auth_functions_<versionsnummer>' bekannt), wenn die Benutzer-ID/Kennwort-Authentifizierung verwendet wird, bereitgestellt wird. Die Struktur 'db2secGssapiClientAuthFunctions_<versionsnummer>' und die Struktur 'db2secUseridPasswordClientAuthFunctions_<versionsnummer>' enthalten wiederum Zeiger auf die APIs, die für die GSS-API-Authentifizierung bzw. die Benutzer-ID/Kennwort-Authentifizierung implementiert sind. In zukünftigen Versionen von DB2 wird es möglicherweise andere Versionen der APIs geben, sodass der Parameter 'client_fns' als Zeiger auf die Struktur 'gssapi_client_auth_functions_<versionsnummer>' umgesetzt wird, die der Version entspricht, die vom Plug-in implementiert wird.

Der erste Parameter der Struktur 'gssapi_client_auth_functions_<versionsnummer>' bzw. der Struktur 'userid_password_client_auth_functions_<versionsnummer>' teilt dem DB2-Datenbankmanager die Version der APIs mit, die vom Plug-in implementiert werden.

Anmerkung: Die Umsetzung (CAST) erfolgt nur, wenn die DB2-Version höher oder gleich der Version der APIs ist, die vom Plug-in implementiert werden.

In der Struktur 'gssapi_server_auth_functions_<versionsnummer>' bzw. 'userid_password_server_auth_functions_<versionsnummer>' sollte der Parameter 'plugintype' auf einen der Werte DB2SEC_PLUGIN_TYPE_USERID_PASSWORD, DB2SEC_PLUGIN_TYPE_GSSAPI oder DB2SEC_PLUGIN_TYPE_KERBEROS gesetzt werden. Andere Werte können in zukünftigen Versionen der API definiert werden.

logMessage_fn
Eingabe. Ein Zeiger auf die API 'db2secLogMessage', die vom DB2-Datenbankmanager implementiert wird. Die API 'db2secClientAuthPluginInit' kann die API 'db2secLogMessage' aufrufen, um Nachrichten in der Datei db2diag.log zu Debug- oder Informationszwecken zu protokollieren. Der erste Parameter ('level') der API 'db2secLogMessage' gibt den Typ der zu diagnostizierenden Fehler an, die in der Datei db2diag.log aufgezeichnet werden. Die beiden letzten Parameter geben die Nachrichtenzeichenfolge bzw. ihre Länge an. Die folgenden Werte sind für den ersten Parameter der API 'dbesecLogMessage' (in der Datei db2secPlugin.h definiert) gültig:

- DB2SEC_LOG_NONE (0) - Keine Protokollierung
- DB2SEC_LOG_CRITICAL (1) - Schwer wiegender Fehler
- DB2SEC_LOG_ERROR (2) - Fehler
- DB2SEC_LOG_WARNING (3) - Warnung
- DB2SEC_LOG_INFO (4) - Informativ

Der Nachrichtentext wird in der Datei db2diag.log nur gezeigt, wenn der Wert des Parameters 'level' der API 'db2secLogMessage' kleiner oder gleich dem Wert des Konfigurationsparameters 'diaglevel' des Datenbankmanagers ist. Wenn Sie zum Beispiel den Wert DB2SEC_LOG_INFO verwenden, wird der Nachrichtentext in der Datei db2diag.log nur gezeigt, wenn der Konfigurationsparameter 'diaglevel' des Datenbankmanagers auf den Wert 4 gesetzt ist.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secClientAuthPluginInit' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secClientAuthPluginTerm (API) - Ressourcen für Plug-in zur Clientauthentifizierung bereinigen

Gibt Ressourcen frei, die vom Plug-in zur Clientauthentifizierung verwendet wurden.

Diese Anwendungsprogrammierschnittstelle (API) wird vom DB2-Datenbankmanager kurz vor dem Entladen des Plug-ins zur Clientauthentifizierung aufgerufen. Sie sollte in einer Weise implementiert werden, in der sie eine ordnungsgemäße Bereinigung aller Ressourcen ausführt, die von der Bibliothek des Plug-ins genutzt werden. Sie sollte zum Beispiel den vom Plug-in zugeordneten Speicher freigeben, Dateien schließen, die noch geöffnet sind, und Netzwertverbindungen schließen. Es liegt in der Zuständigkeit des Plug-ins, diese Ressourcen zu verfolgen, um sie freigeben zu können. Diese API wird auf keiner Windows-Plattform aufgerufen.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secClientAuthPluginTerm)
                      ( char     **errormsg,
                        db2int32 *errormsglen);
```

Parameter der API 'db2secClientAuthPluginTerm'

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secClientAuthPluginTerm' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secDoesAuthIDExist - Vorhandensein der Berechtigungs-ID überprüfen

Ermittelt, ob die Berechtigungs-ID (authid) einen einzelnen Benutzer darstellt (z. B. ob die API die Berechtigungs-ID einer externen Benutzer-ID zuordnen kann).

Die Anwendungsprogrammierschnittstelle (API) sollte den Wert DB2SEC_PLUGIN_OK zurückgeben, wenn sie erfolgreich ist, d. h. wenn die Berechtigungs-ID gültig ist. Ist die Berechtigungs-ID nicht gültig, sollte sie den Wert DB2SEC_PLUGIN_INVALID_USERORGROUP zurückgeben. Lässt sich das Vorhandensein der Berechtigungs-ID nicht feststellen, sollte der Wert DB2SEC_PLUGIN_USERSTATUSNOTKNOWN zurückgegeben werden.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secDoesAuthIDExist)
                        ( const char *authid,
                          db2int32 authidlen,
                          char       **errormsg,
                          db2int32 *errormsglen );
```

Parameter der API 'db2secDoesAuthIDExist'

authid Eingabe. Die zu überprüfende Berechtigungs-ID. Dieser Wert liegt in Großbuchstaben und ohne folgende Leerzeichen vor.

authidlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'authid'.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secDoesAuthIDExist' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secFreeInitInfo (API) - Von 'db2secGenerateInitialCred' genutzte Ressourcen bereinigen

Gibt alle Ressourcen frei, die durch die API 'db2secGenerateInitialCred' zugeordnet wurden. Dazu können zum Beispiel interne Kennungen (Handles) für Kontexte zugrunde liegender Mechanismen oder ein für die GSS-API erstellter Cache für Berechtigungsnachweise gehören.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secFreeInitInfo)
                     ( void *initinfo,
                       char     **errormsg,
                       db2int32 *errormsglen);
```

Parameter der API 'db2secFreeInitInfo'

initinfo
Eingabe. Ein Zeiger auf Daten, die dem DB2-Datenbankmanager nicht bekannt sind. Das Plug-in kann diesen Speicher verwenden, um eine Liste von Ressourcen zu verwalten, die im Prozess der Generierung der Berechtigungsnachweiskennung zugeordnet werden. Diese Ressourcen werden durch Aufrufen dieser API freigegeben.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secFreeInitInfo' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secFreeToken (API) - Vom Token belegten Speicher freigeben

Gibt den Speicher frei, der von einem Token belegt wird. Diese Anwendungsprogrammierschnittstelle (API) wird vom DB2-Datenbankmanager aufgerufen, wenn er den Speicher, der von dem Parameter 'token' belegt wird, nicht mehr benötigt.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secFreeToken)
                     ( void *token,
                       char       **errormsg,
                       db2int32 *errormsglen );
```

Parameter der API 'db2secFreeToken'

token Eingabe. Zeiger auf den freizugebenden Speicher.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secFreeToken' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secGenerateInitialCred (API) - Anfangsberechtigungsnachweise generieren

Ruft die GSS-API-Anfangsberechtigungsnachweise auf der Basis der übergebenen Benutzer-ID und des Kennworts ab. Für Kerberos ist dies das Ticket-Granting-Ticket (TGT). Die Berechtigungsnachweiskennung, die im Parameter 'pGSSCred-

Handle' zurückgegeben wird, ist die Kennung, die mit der API 'gss_init_sec_context' verwendet wird. Dabei muss es sich entweder um INITIATE-Berechtigungsnachweise oder um BOTH-Berechtigungsnachweise handeln. Die API 'db2secGenerateInitialCred' wird nur aufgerufen, wenn eine Benutzer-ID und möglicherweise ein Kennwort angegeben werden. Ansonsten gibt der DB2-Datenbankmanager den Wert GSS_C_NO_CREDENTIAL an, wenn er die API 'gss_init_sec_context' aufruft, um anzuzeigen, dass die aus dem aktuellen Anmeldekontext abgerufenen Standardberechtigungsnachweise zu verwenden sind.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secGenerateInitialCred)
                      ( const char *userid,
                        db2int32 useridlen,
                        const char *usernamespace,
                        db2int32 usernamespacelen,
                        db2int32 usernamespacetype,
                        const char *password,
                        db2int32 passwordlen,
                        const char *newpassword,
                        db2int32 newpasswordlen,
                        const char *dbname,
                        db2int32 dbnamelen,
                        gss_cred_id_t *pGSSCredHandle,
                        void          **InitInfo,
                        char          **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secGenerateInitialCred'

userid Eingabe. Die Benutzer-ID, deren Kennwort auf dem Datenbankserver zu prüfen ist.

useridlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

usernamespace
Eingabe. Der Namensbereich, aus dem die Benutzer-ID abgerufen wurde.

usernamespacelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'.

usernamespacetype
Eingabe. Der Typ des Namensbereichs.

password
Eingabe. Das zu prüfende Kennwort.

passwordlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'password'.

newpassword
Eingabe. Ein neues Kennwort, wenn das Kennwort geändert werden soll. Wenn keine Änderung angefordert wird, ist der Parameter 'newpassword' auf den Wert NULL gesetzt. Wenn er nicht NULL ist, sollte die API das alte Kennwort prüfen, bevor sie das Kennwort auf den neuen Wert setzt. Die API muss die Anforderung zum Ändern des Kennworts nicht ausführen. Wenn sie es jedoch nicht tut, sollte sie sofort mit dem Rückgabewert DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED zurückkehren, ohne das alte Kennwort zu prüfen.

newpasswordlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'newpassword'.

dbname
Eingabe. Der Name der Datenbank, zu der die Verbindung hergestellt wird. Die API hat die Freiheit, diesen Parameter zu ignorieren. Alternativ kann die API auch den Wert DB2SEC_PLUGIN_CONNECTION_DISALLOWED zurückgeben, wenn sie der Richtlinie unterliegt, den Zugriff auf bestimmte Datenbanken für Benutzer einzuschränken, die ansonsten gültige Kennwörter besitzen.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'.

pGSSCredHandle
Ausgabe. Ein Zeiger auf die GSS-API-Berechtigungsnachweiskennung.

InitInfo
Ausgabe. Ein Zeiger auf Daten, die DB2 nicht bekannt sind. Das Plug-in kann diesen Speicher verwenden, um eine Liste von Ressourcen zu verwalten, die im Prozess der Generierung der Berechtigungsnachweiskennung zugeordnet werden. Der DB2-Datenbankmanager ruft die API 'db2secFreeInitInfo' am Ende des Authentifizierungsprozesses auf, sodass diese Ressourcen anschließend freigegeben werden. Wenn die API 'db2secGenerateInitialCred' keine solche Liste zu verwalten braucht, sollte der Wert NULL zurückgegeben werden.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secGenerateInitialCred' nicht erfolgreich ausgeführt wird.

Anmerkung: Für diese API sollten keine Fehlernachrichten erstellt werden, wenn der Rückgabewert auf eine falsche Benutzer-ID oder ein falsches Kennwort hinweist. Es sollte nur eine Fehlernachricht zurückgegeben werden, wenn ein interner Fehler in der API aufgetreten ist, der die ordnungsgemäße Ausführung der API verhindert hat.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secGetAuthIDs (API) - Berechtigungs-IDs abrufen

Gibt eine SQL-Berechtigungs-ID (authid) für einen authentifizierten Benutzer zurück. Diese Anwendungsprogrammierschnittstelle (API) wird während der Herstellung von Datenbankverbindungen sowohl beim Verfahren der Benutzer-ID/Kennwort-Authentifizierung als auch beim Verfahren der GSS-API-Authentifizierung aufgerufen.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secGetAuthIDs)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *dbname,
                         db2int32 dbnamelen,
                         void      **token,
                         char SystemAuthID[DB2SEC_MAX_AUTHID_LENGTH],
```

```
db2int32 *SystemAuthIDlen,
char InitialSessionAuthID[DB2SEC_MAX_AUTHID_LENGTH],
db2int32 *InitialSessionAuthIDlen,
char username[DB2SEC_MAX_USERID_LENGTH],
db2int32 *usernamelen,
db2int32 *initsessionidtype,
char     **errormsg,
db2int32 *errormsglen );
```

Parameter der API 'db2secGetAuthIDs'

userid Eingabe. Der authentifizierte Benutzer. Dieser Wert wird in der Regel nicht für die GSS-API-Authentifizierung verwendet, es sei denn, es ist ein gesicherter Kontext definiert, der Operationen zum Wechseln von Benutzern ohne Authentifizierung zulässt. In solchen Fällen wird der für die Benutzerwechselanforderung bereitgestellte Benutzername in diesem Parameter übergeben.

useridlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

usernamespace
Eingabe. Der Namensbereich, aus dem die Benutzer-ID abgerufen wurde.

usernamespacelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'.

usernamespacetype
Eingabe. Der Wert für den Typ des Namensbereichs. Gegenwärtig wird nur der Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE für den Typ des Namensbereichs unterstützt (entspricht einem Benutzernamen des Formats 'domäne\meinname').

dbname
Eingabe. Der Name der Datenbank, zu der die Verbindung hergestellt wird. Die API kann diesen Parameter ignorieren oder unterschiedliche Berechtigungs-IDs zurückgeben, wenn derselbe Benutzer Verbindungen zu verschiedenen Datenbanken herstellt. Dieser Parameter kann den Wert NULL haben.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'. Dieser Parameter wird auf den Wert 0 gesetzt, wenn der Parameter 'dbname' den Wert NULL hat.

token Eingabe oder Ausgabe. Daten, die das Plug-in an die API 'db2secGetGroupsForUser' übergeben kann. Für GSS-API ist dies eine Kontextkennung (gss_ctx_id_t). Normalerweise ist der Parameter 'token' ein reiner Eingabeparameter, dessen Wert aus der API 'db2secValidatePassword' empfangen wird. Er kann auch ein Ausgabeparameter sein, wenn die Authentifizierung auf dem Client erfolgt und dementsprechend die API 'db2secValidatePassword' nicht aufgerufen wird. In Umgebungen, in denen ein gesicherter Kontext definiert wird, der Operationen zum Benutzerwechsel ohne Authentifizierung zulässt, muss die API 'db2secGetAuthIDs' in der Lage sein, einen NULL-Wert für diesen Parameter 'token' zu empfangen und eine Systemberechtigungs-ID auf der Basis der oben beschriebenen Eingabeparameter 'userid' und 'useridlen' abzuleiten.

SystemAuthID
Ausgabe. Die Systemberechtigungs-ID, die der ID des authentifizierten Benutzers entspricht. Die Größe beträgt 255 Byte, jedoch verwendet der DB2-Datenbankmanager gegenwärtig nur bis zu 30 Byte (einschließlich).

SystemAuthIDlen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'SystemAuthID'.

InitialSessionAuthID
Ausgabe. Die Berechtigungs-ID, die für diese Verbindungssitzung verwendet wird. Dieser Wert stimmt in der Regel mit dem Wert des Parameters 'SystemAuthID' überein. Er kann sich in einigen Fällen jedoch unterscheiden, zum Beispiel, wenn eine Anweisung SET SESSION AUTHORIZATION ausgeführt wird. Die Größe beträgt 255 Byte, jedoch verwendet der DB2-Datenbankmanager gegenwärtig nur bis zu 30 Byte (einschließlich).

InitialSessionAuthIDlen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'InitialSessionAuthID'.

username
Ausgabe. Ein Benutzername, der dem authentifizierten Benutzer und der Berechtigungs-ID (authid) entspricht. Dieser Wert wird nur für die Prüffunktion verwendet und im Feld für die Benutzer-ID in Prüfsatz für die CONNECT-Anweisung protokolliert. Wenn die API den Parameter 'username' nicht mit einem Wert füllt, kopiert der DB2-Datenbankmanager den Wert aus dem Feld 'userid'.

usernamelen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'username'.

initsessionidtype
Ausgabe. Der Typ der Sitzungsberechtigung, der angibt, ob der Parameter 'InitialSessionAuthid' eine Rolle oder eine Berechtigungs-ID (authid) angibt. Die API sollte einen der folgenden Werte (in der Datei 'db2secPlugin.h' definiert) zurückgeben:
- DB2SEC_ID_TYPE_AUTHID (0)
- DB2SEC_ID_TYPE_ROLE (1)

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secGetAuthIDs' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secGetDefaultLoginContext (API) - Standardanmeldekontext abrufen

Diese API ermittelt den Benutzer, der dem Standardanmeldekontext zugeordnet ist. Das heißt, sie ermittelt die DB2-Berechtigungs-ID (authid) des Benutzers, der einen DB2-Befehl aufruft, ohne explizit eine Benutzer-ID anzugeben (entweder durch eine implizite Authentifizierung für eine Datenbank oder durch eine lokale Authentifizierung). Diese API muss sowohl eine Berechtigungs-ID (authid) als auch eine Benutzer-ID zurückgeben.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secGetDefaultLoginContext)
                      ( char authid[DB2SEC_MAX_AUTHID_LENGTH],
                        db2int32 *authidlen,
                        char userid[DB2SEC_MAX_USERID_LENGTH],
                        db2int32 *useridlen,
                        db2int32 useridtype,
                        char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
                        db2int32 *usernamespacelen,
                        db2int32 *usernamespacetype,
                        const char *dbname,
                        db2int32 dbnamelen,
                        void      **token,
                        char      **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secGetDefaultLoginContext'

authid Ausgabe. Der Parameter, in dem die Berechtigungs-ID zurückgegeben werden sollte. Der zurückgegebene Wert muss den DB2-Namensregeln für Berechtigungs-IDs entsprechen. Ansonsten wird der Benutzer nicht zur Ausführung der angeforderten Aktion berechtigt.

authidlen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'authid'.

userid Ausgabe. Der Parameter, in dem die dem Standardanmeldekontext zugeordnete Benutzer-ID zurückgegeben werden sollte.

useridlen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

useridtype
Eingabe. Gibt an, ob die reale oder die effektive Benutzer-ID des Prozesses angegeben wird. Unter Windows ist nur die reale Benutzer-ID vorhanden. Unter UNIX und Linux können sich die reale Benutzer-ID und die effektive Benutzer-ID unterscheiden, wenn die Benutzer-ID `uid` für die Anwendung eine andere ID ist als die ID des Benutzers, der den Prozess ausführt. Die folgenden Werte für den Parameter 'userid' (in der Datei 'db2secPlugin.h' definiert) sind gültig:

DB2SEC_PLUGIN_REAL_USER_NAME
Gibt an, dass die reale Benutzer-ID angegeben wird.

DB2SEC_PLUGIN_EFFECTIVE_USER_NAME
Gibt an, dass die effektive (aktuelle) Benutzer-ID angegeben wird.

Anmerkung: Bei einigen Plug-in-Implementierungen wird möglicherweise nicht zwischen der realen und der effektiven Benutzer-ID unterschieden. Insbesondere kann ein Plug-in diesen Unterschied gefahrlos ignorieren, das nicht mit der UNIX- oder Linux-Identität des Benutzers arbeitet, um die DB2-Berechtigungs-ID einzurichten.

usernamespace
Ausgabe. Der Namensbereich der Benutzer-ID.

usernamespacelen
Ausgabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'. Unter der Einschränkung, dass der Parameter 'usernamespacetype' auf den

Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE (in der Datei 'db2secPlugin.h' definiert) gesetzt sein muss, beträgt die gegenwärtig unterstützte Maximallänge 15 Byte.

usernamespacetype
Ausgabe. Der Wert für den Typ des Namensbereichs. Gegenwärtig wird nur der Typ DB2SEC_NAMESPACE_SAM_COMPATIBLE für den Namensbereich unterstützt (entspricht einem Benutzernamen des Formats 'domäne\meinname').

dbname
Eingabe. Enthält den Namen der Datenbank, zu der die Verbindung hergestellt wird, wenn dieser Aufruf im Kontext einer Datenbankverbindung verwendet wird. Für lokale Berechtigungsaktionen oder Instanzverbindungen (ATTACH) wird dieser Parameter auf NULL gesetzt.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'.

token Ausgabe. Dies ist ein Zeiger auf vom Plug-in zugeordnete Daten, die vom Plug-in an nachfolgende Authentifizierungsaufrufe im Plug-in übergeben werden können oder potenziell an das Plug-in zum Abrufen von Gruppen übergeben werden können. Die Struktur dieser Daten wird durch den Autor des Plug-ins festgelegt.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secGetDefaultLoginContext' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secProcessServerPrincipalName (API) - Vom Server zurückgegebenen Service-Principal-Namen verarbeiten

Verarbeitet den Namen des Service-Principals, der vom Server zurückgegeben wurde, und gibt den Namen des Principals im internen Format 'gss_name_t' zur Verwendung mit der API 'gss_init_sec_context' zurück. Die API 'db2secProcessServerPrincipalName' verarbeitet auch den Namen des Service-Principals, der im Datenbankverzeichnis katalogisiert ist, wenn die Kerberos-Authentifizierung verwendet wird. Normalerweise verwendet diese Konvertierung die API 'gss_import_name'. Wenn der Kontext eingerichtet ist, wird das Objekt 'gss_name_t' durch einen Aufruf der API 'gss_release_name' freigegeben. Die API 'db2secProcessServerPrincipalName' gibt den Wert DB2SEC_PLUGIN_OK zurück, wenn der Parameter 'gssName' auf einen gültigen GSS-Namen verweist. Ist der Name des Principals ungültig, wird der Fehlercode DB2SEC_PLUGIN_BAD_PRINCIPAL_NAME zurückgegeben.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secProcessServerPrincipalName)
                      ( const char *name,
                        db2int32 namelen,
                        gss_name_t *gssName,
                        char       **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secProcessServerPrincipalName'

name Eingabe. Der Textname des Service-Principals im Format GSS_C_NT_USER_NAME, zum Beispiel service/host@REALM.

namelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'name'.

gssName
Ausgabe. Ein Zeiger auf den Ausgabenamen des Service-Principals im internen GSS-API-Format.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secProcessServerPrincipalName' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secRemapUserid (API) - Zuordnung von Benutzer-ID und Kennwort ändern

Diese Anwendungsprogrammierschnittstelle (API) wird vom DB2-Datenbankmanager auf der Clientseite aufgerufen, um eine gegebene Benutzer-ID mit zugehörigem Kennwort (und möglicherweise ein neues Kennwort und einen neuen Benutzernamensbereich) anderen als den beim Verbindungsaufbau angegebenen Werten zuzuordnen. Der DB2-Datenbankmanager ruft diese API nur auf, wenn beim Verbindungsaufbau eine Benutzer-ID und ein Kennwort angegeben werden. Dies verhindert, dass ein Plug-in eine Benutzer-ID selbsttätig einem anderen Paar aus Benutzer-ID und Kennwort zuordnet. Diese API ist optional und wird nicht aufgerufen, wenn dies nicht vorgesehen ist oder sie vom Sicherheits-Plug-in nicht implementiert wird.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secRemapUserid)
                       ( char userid[DB2SEC_MAX_USERID_LENGTH],
                         db2int32 *useridlen,
                         char usernamespace[DB2SEC_MAX_USERNAMESPACE_LENGTH],
                         db2int32 *usernamespacelen,
                         db2int32 *usernamespacetype,
                         char password[DB2SEC_MAX_PASSWORD_LENGTH],
                         db2int32 *passwordlen,
                         char newpasswd[DB2SEC_MAX_PASSWORD_LENGTH],
                         db2int32 *newpasswdlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         char      **errormsg,
                         db2int32 *errormsglen);
```

Parameter der API 'db2secRemapUserid'

userid Eingabe oder Ausgabe. Die Benutzer-ID, deren Zuordnung zu ändern ist. Wenn ein Eingabewert für die Benutzer-ID angegeben ist, muss die API einen Ausgabewert für die Benutzer-ID bereitstellen, der mit dem Eingabewert für die Benutzer-ID identisch oder ein anderer Wert sein kann. Wenn kein Eingabewert für die Benutzer-ID angegeben ist, sollte die API keinen Ausgabewert für die Benutzer-ID zurückgeben.

useridlen
Eingabe oder Ausgabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

usernamespace
Eingabe oder Ausgabe. Der Namensbereich der Benutzer-ID. Dieser Wert kann optional anders zugeordnet werden. Wenn kein Eingabeparameterwert angegeben wird, jedoch ein Ausgabewert zurückgegeben wird, wird der Parameter 'usernamespace' vom DB2-Datenbankmanager nur für den Authentifizierungstyp CLIENT verwendet und für andere Authentifizierungstypen ignoriert.

usernamespacelen
Eingabe oder Ausgabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'. Unter der Einschränkung, dass der Parameter 'usernamespacetype' auf den Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE (in der Datei 'db2secPlugin.h' definiert) gesetzt sein muss, beträgt die gegenwärtig unterstützte Maximallänge 15 Byte.

usernamespacetype
Eingabe oder Ausgabe. Der alte und der neue Wert für den Typ des Namensbereichs. Gegenwärtig wird nur der Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE für den Typ des Namensbereichs unterstützt (entspricht einem Benutzernamen des Formats 'domäne\meinname').

password
Eingabe oder Ausgabe. Als Eingabe ist dieser Wert das Kennwort, das anders zuzuordnen ist. Als Ausgabe ist dieser Wert das anders zugeordnete Kennwort. Wenn ein Eingabewert für diesen Parameter angegeben wird, muss die API einen Ausgabewert zurückgeben können, der sich vom Eingabewert unterscheidet. Wenn kein Eingabeparameter angegeben wird, darf die API keinen Ausgabewert für das Kennwort zurückgeben.

passwordlen
Eingabe oder Ausgabe. Die Länge (in Byte) des Werts des Parameters 'password'.

newpasswd
Eingabe oder Ausgabe. Als Eingabe ist dieser Wert das neue Kennwort, das festzulegen ist. Als Ausgabe ist dieser Wert das bestätigte neue Kennwort.

Anmerkung: Dies ist das neue Kennwort, das vom DB2-Datenbankmanager in den Parameter 'newpassword' der API 'db2secValidatePassword' auf dem Client oder dem Server (je nach dem Wert des Konfigurationsparameters 'authentication' des Datenbankmanagers) eingefügt wird. Wenn ein neues Kennwort als Eingabe übergeben wurde, muss die API einen Ausgabewert zurückgeben können, bei dem es sich um ein anderes neues Kennwort handeln kann. Wenn kein neues Kennwort als Eingabe übergeben wird, sollte die API kein neues Kennwort als Ausgabe zurückgeben.

newpasswdlen
Eingabe oder Ausgabe. Die Länge (in Byte) des Werts des Parameters 'newpasswd'.

dbname
Eingabe. Der Name der Datenbank, zu der der Client die Verbindung herstellt.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secRemapUserid' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secServerAuthPluginInit - Plug-in zur Serverauthentifizierung initialisieren

Initialisierungs-API für das Plug-in zur Serverauthentifizierung, das der DB2-Datenbankmanager unmittelbar nach dem Laden des Plug-ins aufruft. Im Fall der GSS-API ist das Plug-in dafür zuständig, den Namen des Principals im Parameter 'serverPrincipalName' in der Struktur 'gssapi_server_auth_functions' bei der Initialisierung auszufüllen und die Berechtigungsnachweiskennung des Servers im Parameter 'serverCredHandle' in der Struktur 'gssapi_server_auth_functions' anzugeben. Die Freigabe des Speichers, der zur Aufnahme des Namens des Principals und der Berechtigungsnachweiskennung zugeordnet wird, muss von der API 'db2secServerAuthPluginTerm' durch Aufrufen der APIs 'gss_release_name' und 'gss_release_cred' ausgeführt werden.

API- und Datenstruktursyntax

```
SQL_API_RC SQL_API_FN db2secServerAuthPluginInit
                              ( db2int32 version,
                                void *server_fns,
                                db2secGetConDetails *getConDetails_fn,
                                db2secLogMessage *logMessage_fn,
                                char               **errormsg,
                                db2int32 *errormsglen );
```

Parameter der API 'db2secServerAuthPluginInit'

version
Eingabe. Die höchste Versionsnummer der API, die vom DB2-Datenbankmanager gegenwärtig unterstützt wird. Der Wert DB2SEC_API_VERSION (in db2secPlugin.h) enthält die aktuellste Versionsnummer der API, die vom DB2-Datenbankmanager gegenwärtig unterstützt wird.

server_fns
Ausgabe. Ein Zeiger auf Speicher, der vom DB2-Datenbankmanager für eine Struktur 'db2secGssapiServerAuthFunctions_<versionsnummer>' (auch als 'gssapi_server_auth_functions_<versionsnummer>' bekannt), wenn die GSS-API-Authentifizierung verwendet wird, oder für eine Struktur 'db2secUseridPasswordServerAuthFunctions_<versionsnummer>' (auch als 'userid_password_server_auth_functions_<versionsnummer>' bekannt), wenn die Benutzer-ID/Kennwort-Authentifizierung verwendet wird, bereitgestellt wird. Die Struktur 'db2secGssapiServerAuthFunctions_<versionsnummer>' und die Struktur 'db2secUseridPasswordServerAuthFunctions_<versionsnummer>' enthalten wiederum Zeiger auf die APIs, die für die GSS-API-Authentifizierung bzw. die Benutzer-ID/Kennwort-Authentifizierung implementiert sind.

Der Parameter 'server_fns' wird als Zeiger auf die Struktur 'gssapi_server_auth_functions_<versionsnummer>' umgesetzt, die der Version entspricht, die vom Plug-in implementiert wird. Der erste Parameter der Struktur 'gssapi_server_auth_functions_<versionsnummer>' bzw. der Struktur 'userid_password_server_auth_functions_<versionsnummer>' teilt dem DB2-Datenbankmanager die Version der APIs mit, die vom Plug-in implementiert werden.

Anmerkung: Die Umsetzung (CAST) erfolgt nur, wenn die DB2-Version höher oder gleich der Version der APIs ist, die vom Plug-in implementiert werden.

In der Struktur 'gssapi_server_auth_functions_<versionsnummer>' bzw. 'userid_password_server_auth_functions_<versionsnummer>' sollte der Parameter 'plugintype' auf einen der Werte DB2SEC_PLUGIN_TYPE_USERID_PASSWORD, DB2SEC_PLUGIN_TYPE_GSSAPI oder DB2SEC_PLUGIN_TYPE_KERBEROS gesetzt werden. Andere Werte können in zukünftigen Versionen der API definiert werden.

getConDetails_fn

Eingabe. Ein Zeiger auf die API 'db2secGetConDetails', die von DB2 implementiert wird. Die API 'db2secServerAuthPluginInit' kann die API 'db2secGetConDetails' in einer beliebigen der anderen Authentifizierungs-APIs aufrufen, um Details zur Datenbankverbindung abzurufen. Zu diesen Details gehören Informationen zu dem Kommunikationsmechanismus, der der Verbindung zugeordnet ist (z.B. die IP-Adresse im Fall von TCP/IP), die der Autor des Plug-ins möglicherweise benötigt, wenn er Entscheidungen in Bezug auf die Authentifizierung trifft. Zum Beispiel könnte das Plug-in eine Verbindung für einen bestimmten Benutzer nicht zulassen, wenn dieser Benutzer die Verbindung nicht über eine bestimmte IP-Adresse herstellt. Die Verwendung der API 'db2secGetConDetails' ist optional.

Wenn die API 'db2secGetConDetails' in einer Situation aufgerufen wird, in der keine Datenbankverbindung besteht, gibt sie den Wert DB2SEC_PLUGIN_NO_CON_DETAILS zurück. Anderenfalls gibt sie bei Erfolg den Wert 0 zurück.

Die API 'db2secGetConDetails' empfängt zwei Eingabeparameter: den Parameter 'pConDetails', bei dem es sich um einen Zeiger auf die Struktur 'db2sec_con_details_<versionsnummer>' handelt, und den Parameter 'conDetailsVersion', bei dem es sich um eine Versionsnummer handelt, die angibt, welche Struktur 'db2sec_con_detailsx' zu verwenden ist. Mögliche Werte sind DB2SEC_CON_DETAILS_VERSION_1, wenn 'db2sec_con_details1' verwendet wird, oder DB2SEC_CON_DETAILS_VERSION_2, wenn 'db2sec_con_details2' verwendet wird. Die zur Verwendung empfohlene Versionsnummer ist DB2SEC_CON_DETAILS_VERSION_2.

Bei einer erfolgreichen Rückgabe enthält die Struktur 'db2sec_con_detailsx' (entweder 'db2sec_con_details1' oder 'db2sec_con_details2') die folgenden Informationen:

- Das für die Verbindung zum Server verwendete Protokoll. Die Liste der Protokolldefinitionen (SQL_PROTOCOL_*) befindet sich in der Datei 'sqlenv.h' (im include-Verzeichnis). Diese Information wird in den Parameter 'clientProtocol' eingefügt.

- Die TCP/IP-Adresse der eingehenden Verbindung zum Server, wenn der Parameter 'clientProtocol' den Wert SQL_PROTOCOL_TCPIP oder SQL-_PROTOCOL_TCPIP4 hat. Diese Information wird in den Parameter 'clientIPAddress' eingefügt.
- Der Name der Datenbank, zu der der Client eine Verbindung aufbauen will. Dies wird für Instanzverbindungen (mit ATTACH) nicht angegeben. Diese Information wird in die Parameter 'dbname' und 'dbnameLen' eingefügt.
- Eine Bitzuordnung mit Verbindungsinformationen, die dieselben Details wie die enthält, die im Parameter 'connection_details' der API 'db2secValidatePassword' dokumentiert werden. Diese Information wird in den Parameter 'connect_info_bitmap' eingefügt.
- Die TCP/IP-Adresse der eingehenden Verbindung zum Server, wenn der Parameter 'clientProtocol' den Wert SQL_PROTOCOL_TCPIP6 hat. Diese Information wird in den Parameter 'clientIP6Address' eingefügt und ist nur verfügbar, wenn für den Aufruf der API 'db2secGetConDetails' der Wert DB2SEC_CON_DETAILS_VERSION_2 verwendet wird.

logMessage_fn

Eingabe. Ein Zeiger auf die API 'db2secLogMessage', die vom DB2-Datenbankmanager implementiert wird. Die API 'db2secClientAuthPluginInit' kann die API 'db2secLogMessage' aufrufen, um Nachrichten in der Datei db2diag.log zu Debug- oder Informationszwecken zu protokollieren. Der erste Parameter ('level') der API 'db2secLogMessage' gibt den Typ der zu diagnostizierenden Fehler an, die in der Datei db2diag.log aufgezeichnet werden. Die beiden letzten Parameter geben die Nachrichtenzeichenfolge bzw. ihre Länge an. Die folgenden Werte sind für den ersten Parameter der API 'dbesecLogMessage' (in der Datei db2secPlugin.h definiert) gültig:

DB2SEC_LOG_NONE (0)
Keine Protokollierung

DB2SEC_LOG_CRITICAL (1)
Schwer wiegende Fehler

DB2SEC_LOG_ERROR (2)
Fehler

DB2SEC_LOG_WARNING (3)
Warnung

DB2SEC_LOG_INFO (4)
Informativ

Der Nachrichtentext wird in der Datei db2diag.log nur gezeigt, wenn der Wert des Parameters 'level' der API 'db2secLogMessage' kleiner oder gleich dem Wert des Konfigurationsparameters 'diaglevel' des Datenbankmanagers ist.

Wenn Sie zum Beispiel den Wert DB2SEC_LOG_INFO verwenden, wird der Nachrichtentext in der Datei 'db2diag.log' nur gezeigt, wenn der Konfigurationsparameter 'diaglevel' des Datenbankmanagers auf den Wert 4 gesetzt ist.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten

ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secServerAuthPluginInit' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secServerAuthPluginTerm (API) - Ressourcen für Plug-in zur Serverauthentifizierung bereinigen

Gibt Ressourcen frei, die vom Plug-in zur Serverauthentifizierung verwendet wurden. Diese Anwendungsprogrammierschnittstelle (API) wird vom DB2-Datenbankmanager kurz vor dem Entladen des Plug-ins zur Serverauthentifizierung aufgerufen. Sie sollte in einer Weise implementiert werden, in der sie eine ordnungsgemäße Bereinigung aller Ressourcen ausführt, die von der Bibliothek des Plug-ins genutzt werden. Sie sollte zum Beispiel den vom Plug-in zugeordneten Speicher freigeben, Dateien schließen, die noch geöffnet sind, und Netzwertverbindungen schließen. Es liegt in der Zuständigkeit des Plug-ins, diese Ressourcen zu verfolgen, um sie freigeben zu können. Diese API wird auf keiner Windows-Plattform aufgerufen.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secServerAuthPluginTerm)
                      ( char     **errormsg,
                        db2int32 *errormsglen );
```

Parameter der API 'db2secServerAuthPluginTerm'

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secServerAuthPluginTerm' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

db2secValidatePassword (API) - Kennwort prüfen

Stellt eine Methode zur Ausführung einer Authentifizierung durch Benutzer-ID und Kennwort während einer Operation zur Herstellung einer Datenbankverbindung bereit.

Anmerkung: Wenn die Anwendungsprogrammierschnittstelle (API) auf der Clientseite ausgeführt wird, wird der API-Code mit den Berechtigungen des Benutzers ausgeführt, der die Anweisung CONNECT ausführt. Diese API wird auf der Clientseite nur aufgerufen, wenn der Konfigurationsparameter 'authentication' auf den Wert CLIENT gesetzt ist.

Wenn die API auf der Serverseite ausgeführt wird, wird der API-Code mit den Berechtigungen des Instanzeigners ausgeführt.

Der Plug-in-Autor sollte die oben gegebenen Hinweise beachten, wenn die Authentifizierung besondere Berechtigungen (z. B. Systemzugriff auf Rootebene unter UNIX) erfordert.

Diese API muss den Wert DB2SEC_PLUGIN_OK (Erfolg) zurückgeben, wenn das Kennwort gültig ist. Wenn das Kennwort ungültig ist, muss sie einen Fehlercode wie zum Beispiel DB2SEC_PLUGIN_BADPWD zurückgeben.

API- und Datenstruktursyntax

```
SQL_API_RC ( SQL_API_FN *db2secValidatePassword)
                       ( const char *userid,
                         db2int32 useridlen,
                         const char *usernamespace,
                         db2int32 usernamespacelen,
                         db2int32 usernamespacetype,
                         const char *password,
                         db2int32 passwordlen,
                         const char *newpasswd,
                         db2int32 newpasswdlen,
                         const char *dbname,
                         db2int32 dbnamelen,
                         db2Uint32 connection_details,
                         void      **token,
                         char      **errormsg,
                         db2int32 *errormsglen );
```

Parameter der API 'db2secValidatePassword'

userid Eingabe. Die Benutzer-ID, deren Kennwort zu prüfen ist.

useridlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'userid'.

usernamespace
Eingabe. Der Namensbereich, aus dem die Benutzer-ID abgerufen wurde.

usernamespacelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'usernamespace'.

usernamespacetype
Eingabe. Der Typ des Namensbereichs. Folgende Werte für den Parameter 'usernamespacetype' (in der Datei db2secPlugin.h definiert) sind gültig:

- `DB2SEC_NAMESPACE_SAM_COMPATIBLE`: Entspricht einer Benutzernamensdarstellung des Formats 'domäne\meinname'.
- `DB2SEC_NAMESPACE_USER_PRINCIPAL`: Entspricht einer Benutzernamensdarstellung des Formats 'meinname@domäne.ibm.com'.

Gegenwärtig unterstützt das DB2-Datenbanksystem nur den Wert DB2SEC_NAMESPACE_SAM_COMPATIBLE. Wenn die Benutzer-ID nicht verfügbar ist, wird der Parameter 'usernamespacetype' auf den Wert DB2SEC_USER_NAMESPACE_UNDEFINED (in der Datei db2secPlugin.h definiert) gesetzt.

password
Eingabe. Das zu prüfende Kennwort.

passwordlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'password'.

newpasswd
Eingabe. Ein neues Kennwort, wenn das Kennwort geändert werden soll. Wenn keine Änderung angefordert wird, ist dieser Parameter auf den Wert

NULL gesetzt. Wenn dieser Parameter nicht NULL ist, sollte die API das alte Kennwort prüfen, bevor sie es in das neue Kennwort ändert. Die API muss die Anforderung zum Ändern des Kennworts nicht erfüllen. Wenn sie es jedoch nicht tut, sollte sie sofort mit dem Rückgabewert DB2SEC_PLUGIN_CHANGEPASSWORD_NOTSUPPORTED zurückkehren, ohne das alte Kennwort zu prüfen.

newpasswdlen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'newpasswd'.

dbname
Eingabe. Der Name der Datenbank, zu der die Verbindung hergestellt wird. Die API hat die Freiheit, den Parameter 'dbname' zu ignorieren. Alternativ kann die API auch den Wert DB2SEC_PLUGIN_CONNECTIONREFUSED zurückgeben, wenn sie der Richtlinie unterliegt, den Zugriff auf bestimmte Datenbanken für Benutzer einzuschränken, die ansonsten gültige Kennwörter besitzen. Dieser Parameter kann den Wert NULL haben.

dbnamelen
Eingabe. Die Länge (in Byte) des Werts des Parameters 'dbname'. Dieser Parameter wird auf den Wert 0 gesetzt, wenn der Parameter 'dbname' den Wert NULL hat.

connection_details
Eingabe. Ein 32-Bit-Parameter, in dem 3 Bit gegenwärtig zum Speichern der folgenden Informationen verwendet werden:

- Das äußerste rechte Bit gibt an, ob die Quelle der Benutzer-ID der Standardwert auf der API 'db2secGetDefaultLoginContext' ist oder während des Verbindungsaufbaus explizit angegeben wurde.
- Das zweite Bit von rechts gibt an, ob die Verbindung lokal (durch Interprozesskommunikation (IPC) oder durch eine Verbindung von einem der Knoten in der Datei db2nodes.cfg in der Umgebung mit partitionierten Datenbanken) oder fern (über ein Netzwerk oder durch Loopback) ist. Anhand dieser Information kann die API entscheiden, ob Clients auf demselben System die Verbindung zu dem DB2Server ohne Kennwort herstellen können. Durch das standardmäßig verwendete, betriebssystembasierte Benutzer-ID/Kennwort-Plug-in werden lokale Verbindungen ohne Kennwort von Clients auf demselben System zugelassen (vorausgesetzt, der Benutzer hat Verbindungsberechtigungen).
- Das dritte Bit von rechts gibt an, ob der DB2-Datenbankmanager die API von der Serverseite oder der Clientseite aus aufruft.

Die Bit-Werte sind in der Datei db2secPlugin.h definiert:

- `DB2SEC_USERID_FROM_OS` (0x00000001): Gibt an, dass die Benutzer-ID aus dem Betriebssystem abgerufen und nicht explizit in der CONNECT-Anweisung angegeben wurde.
- `DB2SEC_CONNECTION_ISLOCAL` (0x00000002): Gibt eine lokale Verbindung an.
- `DB2SEC_VALIDATING_ON_SERVER_SIDE` (0x0000004): Gibt an, ob der DB2-Datenbankmanager von der Serverseite oder der Clientseite zur Kennwortprüfung aufruft. Wenn dieser Bit-Wert gesetzt ist, erfolgt der Aufruf des DB2-Datenbankmanagers von der Serverseite aus. Anderenfalls erfolgt er von der Clientseite aus.

Das Standardverhalten des DB2-Datenbanksystems bei einer impliziten Authentifizierung ist, die Verbindung ohne Kennwortprüfung zuzulassen.

Plug-in-Entwickler haben jedoch die Option, die implizite Authentifizierung unter Rückgabe des Fehlers DB2SEC_PLUGIN_BADPASSWORD zu verweigern.

token Eingabe. Ein Zeiger auf Daten, die während der aktuellen Verbindung als Parameter an nachfolgende API-Aufrufe übergeben werden können. Zu möglichen APIs, die aufgerufen werden können, gehören die API 'db2secGetAuthIDs' und die API 'db2secGetGroupsForUser'.

errormsg
Ausgabe. Ein Zeiger auf die Adresse einer vom Plug-in zugeordneten ASCII-Fehlernachrichtenzeichenfolge, die in diesem Parameter zurückgegeben werden kann, wenn die API 'db2secValidatePassword' nicht erfolgreich ausgeführt wird.

errormsglen
Ausgabe. Ein Zeiger auf einen ganzzahligen Wert (Integer), der die Länge (in Byte) der Fehlernachrichtenzeichenfolge im Parameter 'errormsg' angibt.

Erforderliche APIs und Definitionen für Plug-ins zur GSS-API-Authentifizierung

Nachfolgend finden Sie eine vollständige Liste der GSS-APIs, die für die DB2-Sicherheits-Plug-in-Schnittstelle erforderlich sind.

Die unterstützten APIs entsprechend den folgenden Spezifikationen: *Generic Security Service Application Program Interface, Version 2* (IETF RFC2743) und *Generic Security Service API Version 2: C-Bindings* (IETF RFC2744). Vor der Implementierung eines GSS-API-basierten Plug-ins sollten Sie sich mit diesen Spezifikationen eingehend vertraut machen.

Tabelle 38. Erforderliche APIs und Definitionen für Plug-ins zur GSS-API-Authentifizierung

Name		Beschreibung
Clientseitige APIs	`gss_init_sec_context`	Initialisiert einen Sicherheitskontext mit einer Peeranwendung.
Serverseitige APIs	`gss_accept_sec_context`	Akzeptiert einen von einer Peeranwendung initialisierten Sicherheitskontext.
Serverseitige APIs	`gss_display_name`	Konvertiert einen Namen aus dem internen Format in Text.
Allgemeine APIs	`gss_delete_sec_context`	Löscht einen eingerichteten Sicherheitskontext.
Allgemeine APIs	`gss_display_status`	Ruft die einem GSS-API-Statuscode zugeordnete Textfehlernachricht ab.
Allgemeine APIs	`gss_release_buffer`	Löscht einen Puffer.
Allgemeine APIs	`gss_release_cred`	Gibt lokale Datenstrukturen frei, die einem GSS-API-Berechtigungsnachweis zugeordnet sind.
Allgemeine APIs	`gss_release_name`	Löscht den Namen im internen Format.
Erforderliche Definitionen	`GSS_C_DELEG_FLAG`	Fordert eine Delegierung an.
Erforderliche Definitionen	`GSS_C_EMPTY_BUFFER`	Signalisiert, dass `gss_buffer_desc` keine Daten enthält.
Erforderliche Definitionen	`GSS_C_GSS_CODE`	Gibt einen übergeordneten GSS-Statuscode an.

Tabelle 38. Erforderliche APIs und Definitionen für Plug-ins zur GSS-API-Authentifizierung (Forts.)

Name		Beschreibung
Erforderliche Definitionen	GSS_C_INDEFINITE	Gibt an, dass der Mechanismus keinen Gültigkeitsverfall des Kontexts unterstützt.
Erforderliche Definitionen	GSS_C_MECH_CODE	Gibt einen untergeordneten GSS-Statuscode an.
Erforderliche Definitionen	GSS_C_MUTUAL_FLAG	Gegenseitige Authentifizierung angefordert.
Erforderliche Definitionen	GSS_C_NO_BUFFER	Bedeutet, dass die Variable gss_buffer_t nicht auf eine gültige Struktur gss_buffer_desc zeigt.
Erforderliche Definitionen	GSS_C_NO_CHANNEL_BINDINGS	Keine Kommunikationskanalbindungen.
Erforderliche Definitionen	GSS_C_NO_CONTEXT	Bedeutet, dass die Variable gss_ctx_id_t nicht auf einen gültigen Kontext zeigt.
Erforderliche Definitionen	GSS_C_NO_CREDENTIAL	Bedeutet, dass die Variable gss_cred_id_t nicht auf eine gültige Berechtigungsnachweiskennung zeigt.
Erforderliche Definitionen	GSS_C_NO_NAME	Bedeutet, dass die Variable gss_name_t nicht auf einen gültigen internen Namen zeigt.
Erforderliche Definitionen	GSS_C_NO_OID	Standardauthentifizierungsverfahren verwenden.
Erforderliche Definitionen	GSS_C_NULL_OID_SET	Standardverfahren verwenden.
Erforderliche Definitionen	GSS_S_COMPLETE	API erfolgreich abgeschlossen.
Erforderliche Definitionen	GSS_S_CONTINUE_NEEDED	Die Verarbeitung ist nicht abgeschlossen, und die API muss erneut mit dem Antworttoken aus dem Peer aufgerufen werden.

Einschränkungen für Plug-ins zur GSS-API-Authentifizierung

In der nachfolgenden Liste werden Einschränkungen für Plug-ins zur GSS-API-Authentifizierung aufgeführt.

- Es wird immer der Standardsicherheitsmechanismus angenommen. Daher sind keine besonderen OID-Aspekte zu beachten.
- Die einzigen GSS-Services, die in gss_init_sec_context() angefordert werden, sind die Services für gegenseitige Authentifizierung und Delegierung. Der DB2-Datenbankmanager fordert immer ein Ticket für die Delegierung an, verwendet dieses Ticket jedoch nicht zur Generierung eines neuen Tickets.
- Nur die Standardkontextzeit wird angefordert.
- Es werden keine Kontexttoken aus gss_delete_sec_context() vom Client an den Server und umgekehrt gesendet.
- Anonymität wird nicht unterstützt.
- Eine Kanalbindung (Channel Binding) wird nicht unterstützt.
- Wenn die Gültigkeit der Anfangsberechtigungsnachweise abläuft, werden sie vom DB2-Datenbankmanager nicht automatisch erneuert.
- Die GSS-API-Spezifikation legt fest, dass auch wenn gss_init_sec_context() oder gss_accept_sec_context() fehlschlagen, beide Funktionen ein Token an den Peer zurückgeben müssen. Aufgrund der DRDA-Begrenzungen sendet der DB2-Datenbankmanager nur ein Token, wenn gss_init_sec_context() fehlschlägt, und generiert ein Token beim ersten Aufruf.

Kapitel 9. Aufbau der Datensätze der Prüffunktion

Jeder Prüfsatz, der aus dem Prüfprotokoll extrahiert wird, hat eines der in den folgenden Tabellen dargestellten Formate. Jeder Tabelle ist ein Beispieldatensatz vorangestellt.

Die einzelnen Einträge in den Prüfsätzen werden in der dazugehörigen Tabelle Zeile für Zeile beschrieben. Jeder Eintrag wird in der Tabelle in derselben Reihenfolge dargestellt, in der er auch in der Datei mit begrenzter Satzlänge nach der Extrahierungsoperation ausgegeben wird.

Anmerkung:

1. Nicht in allen Feldern der Beispielprüfsätze sind Werte angegeben.
2. Einige Felder, wie z. B. „Access Attempted", werden als Bitmaps in begrenztem ASCII-Format gespeichert. In der hier gezeigten, unstrukturierten Berichtsdatei werden diese Felder jedoch durch Zeichenfolgen dargestellt, die die Bitmapwerte darstellen.

Prüfsatzobjekttypen

In der folgenden Tabelle wird für jeden Prüfsatzobjekttyp angegeben, ob er CHECKING-, OBJMAINT- und SECMAINT-Ereignisse generieren kann.

Tabelle 39. Prüfsatzobjekttypen auf der Basis der Prüfereignisse

Objekttyp	CHECKING-Ereignisse	OBJMAINT-Ereignisse	SECMAINT-Ereignisse
ACCESS_RULE			X
ALIAS	X	X	
ALL	X		
AUDIT_POLICY	X	X	
BUFFERPOOL	X	X	
CHECK_CONSTRAINT		X	
DATABASE	X		X
DATA TYPE		X	
EVENT_MONITOR	X	X	
FOREIGN_KEY		X	
FUNCTION	X	X	X
FUNCTION MAPPING	X	X	
GLOBAL_VARIABLE	X	X	X
HISTOGRAM TEMPLATE	X	X	
INDEX	X	X	X
INDEX EXTENSION		X	
INSTANCE	X		
JAR_FILE		X	
METHOD_BODY	X	X	X
NICKNAME	X	X	X

Tabelle 39. Prüfsatzobjekttypen auf der Basis der Prüfereignisse (Forts.)

Objekttyp	CHECKING-Ereignisse	OBJMAINT-Ereignisse	SECMAINT-Ereignisse
NODEGROUP	X	X	
NONE	X	X	X
OPTIMIZATION PROFILE	X		
PACKAGE	X	X	X
PACKAGE CACHE	X		
PRIMARY_KEY		X	
REOPT_VALUES	X		
ROLE	X	X	X
SCHEMA	X	X	X
SECURITY LABEL		X	X
SECURITY LABEL COMPONENT		X	
SECURITY POLICY		X	X
SEQUENCE	X	X	
SERVER	X	X	X
SERVER OPTION	X	X	
SERVICE CLASS	X	X	
STORED_PROCEDURE	X	X	X
SUMMARY TABLES	X	X	X
TABLE	X	X	X
TABLESPACE	X	X	X
THRESHOLD	X	X	
TRIGGER		X	
TRUSTED CONTEXT	X	X	X
TYPE MAPPING	X	X	
TYPE&TRANSFORM	X	X	
UNIQUE_CONSTRAINT		X	
USER MAPPING	X	X	
VIEW	X	X	X
WORK ACTION SET	X	X	
WORK CLASS SET	X	X	
WORKLOAD	X	X	X
WRAPPER	X	X	
XSR-Objekt	X	X	X

Prüfsatzaufbau für AUDIT-Ereignisse

Die folgende Tabelle zeigt den Prüfsatzaufbau für AUDIT-Ereignisse.

Beispiel für einen Prüfsatz:

```
timestamp=2007-04-10-08.29.52.000001;
category=AUDIT;
audit event=START;
event correlator=0;
event status=0;
userid=newton;
authid=NEWTON;
application id=*LOCAL_APPLICATION;
application name=db2audit.exe;
```

Tabelle 40. Prüfsatzaufbau für AUDIT-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: AUDIT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie AUDIT unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 40. Prüfsatzaufbau für AUDIT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Package Section (Paketabschnitt)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.
Policy Name (Name der Richtlinie)	VARCHAR(128)	Der Name der Prüfrichtlinie.
Policy Association Object Type (Objekttyp der Richtlinienzuordnung)	CHAR(1)	Der Typ des Objekts, dem die Prüfrichtlinie zugeordnet ist. Mögliche Werte: • N = Kurzname (Nickname) • S = MQT (Materialized Query Table) • T = Tabelle (nicht typisiert) • i = Berechtigungs-ID • g = Berechtigung • x = Gesicherter Kontext • leer = Datenbank
Policy Association Subobject Type (Subobjekttyp der Richtlinienzuordnung)	CHAR(1)	Der Typ des Subobjekts, dem die Prüfrichtlinie zugeordnet ist. Wenn der Objekttyp ? (Berechtigungs-ID) ist, sind folgende Werte möglich: • U = Benutzer (User) • G = Gruppe (Group) • R = Rolle (Role)

Tabelle 40. Prüfsatzaufbau für AUDIT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Policy Association Object Name (Objektname der Richtlinienzuordnung)	VARCHAR(128)	Der Name des Objekts, dem die Prüfrichtlinie zugeordnet ist.
Policy Association Object Schema (Objektschema der Richtlinienzuordnung)	VARCHAR(128)	Der Schemaname des Objekts, dem die Prüfrichtlinie zugeordnet ist. Dieser Wert ist NULL, wenn der Objekttyp der Richtlinienzuordnung ein Objekt angibt, für das kein Schema verwendet wird.
Audit Status (AUDIT-Status)	CHAR(1)	Der Status der Kategorie AUDIT in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Checking Status (CHECKING-Status)	CHAR(1)	Der Status der Kategorie CHECKING in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Context Status (CONTEXT-Status)	CHAR(1)	Der Status der Kategorie CONTEXT in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Execute Status (EXECUTE-Status)	CHAR(1)	Der Status der Kategorie EXECUTE in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Execute With Data (Ausführen mit Daten)	CHAR(1)	Die Option WITH DATA der Kategorie EXECUTE in der Prüfrichtlinie. Mögliche Werte: • Y-WITH DATA (mit Daten) • N-WITHOUT DATA (ohne Daten)
Objmaint Status (OBJMAINT-Status)	CHAR(1)	Der Status der Kategorie OBJMAINT in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Secmaint Status (SECMAINT-Status)	CHAR(1)	Der Status der Kategorie SECMAINT in einer Prüfrichtlinie. Mögliche Werte: siehe Feld 'AUDIT-Status'.

Tabelle 40. Prüfsatzaufbau für AUDIT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Sysadmin Status (SYSADMIN-Status)	CHAR(1)	Der Status der Kategorie SYSADMIN in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Validate Status (VALIDATE-Status)	CHAR(1)	Der Status der Kategorie VALIDATE in einer Prüfrichtlinie. Mögliche Werte: • B-Beide (Both) • F-Fehler (Failure) • N-Kein (None) • S-Erfolg (Success)
Error Type (Fehlertyp)	CHAR(8)	Der Fehlertyp in einer Prüfrichtlinie. Mögliche Werte: AUDIT und NORMAL.
Data Path (Datenpfad)	VARCHAR(1024)	Der Pfad zu den aktiven Prüfprotokollen, der im Befehl db2audit configure angegeben wurde.
Archive Path (Archivpfad)	VARCHAR(1024)	Der Pfad zu den archivierten Prüfprotokollen, der im Befehl db2audit configure angegeben wurde.

Prüfsatzaufbau für CHECKING-Ereignisse

Das Format des Prüfsatzes für CHECKING-Ereignisse wird in der folgenden Tabelle gezeigt.

Beispiel für einen Prüfsatz:

```
timestamp=1998-06-24-08.42.11.622984;
category=CHECKING;
audit event=CHECKING_OBJECT;
event correlator=2;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SYSSH200;
package section=0;
object schema=GSTAGER;
object name=NONE;
object type=REOPT_VALUES;
access approval reason=DBADM;
access attempted=STORE;
```

Tabelle 41. Prüfsatzaufbau für CHECKING-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: CHECKING
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie CHECKING unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde.
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde.
Object Type (Objekttyp)	VARCHAR(32)	Der Typ des Objekts, für das das Prüfereignis generiert wurde. Mögliche Werte: siehe Abschnitt „Prüfsatzobjekttypen".
Access Approval Reason (Grund für Zugriffgewährung)	CHAR(18)	Gibt an, warum der Zugriff für dieses Prüfereignis gewährt wurde. Mögliche Werte: siehe Abschnitt „Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen".

Tabelle 41. Prüfsatzaufbau für CHECKING-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Access Attempted (Zugriffsversuch)	CHAR(18)	Gibt an, welche Art von Zugriff versucht wurde. Gültige Werte: siehe Abschnitt „Typen von Zugriffsversuchen bei CHECKING-Ereignissen".
Package Version (Paketversion)	VARCHAR (64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Checked Authorization ID (Überprüfte Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID wird überprüft, wenn sie sich von der Berechtigungs-ID beim Prüfereignis unterscheidet. Dabei kann es sich zum Beispiel um den Zieleigner in der Anweisung TRANSFER OWNERSHIP handeln. Wenn das Prüfereignis SWITCH_USER ist, stellt dieses Feld die Berechtigungs-ID dar, zu der gewechselt wird.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.

Gründe für die Zugriffsgewährung bei CHECKING-Ereignissen

Die folgende Liste zeigt die möglichen Gründe der Zugriffsgewährung für CHECKING-Ereignisse.

0x0000000000000001 ACCESS DENIED
Der Zugriff wird nicht gewährt, sondern verweigert.

0x0000000000000002 SYSADM
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SYSADM.

0x0000000000000004 SYSCTRL
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SYSCTRL.

0x0000000000000008 SYSMAINT
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SYSMAINT.

0x0000000000000010 DBADM
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung DBADM.

0x0000000000000020 DATABASE PRIVILEGE
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über ein explizites Zugriffsrecht für die Datenbank.

0x0000000000000040 OBJECT PRIVILEGE
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über ein Zugriffsrecht für das Objekt oder die Funktion.

0x0000000000000080 DEFINER
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer ist der definierende Benutzer bzw. die definierende Anwendung für das Objekt oder die Funktion.

0x0000000000000100 OWNER
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer ist der Eigner des Objekts oder der Funktion.

0x0000000000000200 CONTROL
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über das Zugriffsrecht CONTROL für das Objekt oder die Funktion.

0x0000000000000400 BIND
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über das Zugriffsrecht BIND für das Paket.

0x0000000000000800 SYSQUIESCE
Der Zugriff wird gewährt. Wenn sich die Instanz oder die Datenbank im Quiescemodus befindet, kann die Anwendung bzw. der Benutzer eine Verbindung (CONNECT oder ATTACH) herstellen.

0x0000000000001000 SYSMON
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SYSMON.

0x0000000000002000 SECADM
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SECADM.

0x0000000000004000 SETSESSIONUSER
Der Zugriff wird gewährt. Die Anwendung oder der Benutzer verfügt über die Berechtigung SETSESSIONUSER.

0x0000000000008000 TRUSTED_CONTEXT_MATCH
Verbindungsattribute stimmten mit den Attributen eines eindeutigen gesicherten Kontexts auf dem DB2-Server überein.

0x0000000000010000 TRUSTED_CONTEXT_USE
Zugriff wird zur Verwendung des gesicherten Kontexts gewährt.

Typen von Zugriffsversuchen bei CHECKING-Ereignissen

Die folgende Liste zeigt die möglichen Typen von Zugriffsversuchen für CHECKING-Ereignisse.

Wenn das Prüfereignis CHECKING_TRANSFER ist, gibt der Prüfeintrag an, ob ein Zugriffsrecht enthalten ist oder nicht.

0x0000000000000001 CONTROL
Es wurde versucht, zu prüfen, ob das Zugriffsrecht CONTROL erteilt wurde.

0x0000000000000002 ALTER
Es wurde versucht, ein Objekt zu ändern bzw. zu prüfen, ob das Zugriffsrecht ALTER erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000004 DELETE
Es wurde versucht, ein Objekt zu löschen bzw. zu prüfen, ob das Zugriffsrecht DELETE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000008 INDEX
Es wurde versucht, einen Index zu verwenden bzw. zu prüfen, ob das Zugriffsrecht INDEX erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000010 INSERT
Es wurde ein Objekteinfügevorgang versucht bzw. es wurde versucht, zu prüfen, ob das Zugriffsrecht INSERT erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000020 SELECT
Es wurde versucht, eine Tabelle oder eine Sicht abzufragen bzw. zu prüfen, ob das Zugriffsrecht SELECT erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000040 UPDATE
Es wurde versucht, Daten in einem Objekt zu aktualisieren bzw. zu prüfen, ob das Zugriffsrecht UPDATE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000080 REFERENCE
Es wurde versucht, referenzielle Integritätsbedingungen zwischen Objekten zu erstellen bzw. zu prüfen, ob das Zugriffsrecht REFERENCE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000000100 CREATE
Es wurde versucht, ein Objekt zu erstellen.

0x0000000000000200 DROP
Es wurde versucht, ein Objekt zu löschen.

0x0000000000000400 CREATEIN
Es wurde versucht, ein Objekt in einem anderen Schema zu erstellen.

0x0000000000000800 DROPIN
Es wurde versucht, ein in einem anderen Schema gefundenes Objekt zu löschen.

0x0000000000001000 ALTERIN
Es wurde versucht, ein in einem anderen Schema gefundenes Objekt zu ändern.

0x0000000000002000 EXECUTE
Es wurde versucht, eine Anwendung auszuführen oder eine Routine aufzurufen, eine Funktion aus der Routine als Quelle (gilt nur für Funktionen) zu erstellen oder auf eine Routine in einer DDL-Anweisung zu verweisen bzw. zu prüfen, ob das Zugriffsrecht EXECUTE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000000004000 BIND
Es wurde versucht, eine Anwendung zu binden oder vorzubereiten.

0x0000000000008000 SET EVENT MONITOR
Es wurde versucht, Ereignismonitorschalter zu setzen.

0x0000000000010000 SET CONSTRAINTS
Es wurde versucht, Integritätsbedingungen für ein Objekt zu definieren.

0x0000000000020000 COMMENT ON
Es wurde versucht, Kommentare zu einem Objekt zu erstellen.

0x0000000000040000 GRANT
Es wurde versucht, einer anderen Berechtigungs-ID Zugriffsrechte oder Rollen für ein Objekt zu erteilen.

0x0000000000080000 REVOKE
Es wurde versucht, einer Berechtigungs-ID Zugriffsrechte oder Rollen für ein Objekt zu entziehen.

0x0000000000100000 LOCK
Es wurde versucht, ein Objekt zu sperren.

0x0000000000200000 RENAME
Es wurde versucht, ein Objekt umzubenennen.

0x0000000000400000 CONNECT
Es wurde versucht, eine Verbindung zu einem Objekt herzustellen.

0x0000000000800000 Member of SYS Group
Es wurde versucht, auf ein Mitglied der SYS-Gruppe zuzugreifen oder ein Mitglied der SYS-Gruppe zu verwenden.

0x0000000001000000 Access All
Es wurde versucht, eine Anweisung mit allen erforderlichen Zugriffsrechten für gespeicherte Objekte auszuführen (wird nur für DBADM/SYSADM verwendet).

0x0000000002000000 Drop All
Es wurde versucht, mehrere Objekte zu löschen.

0x0000000004000000 LOAD
Es wurde versucht, eine Tabelle in einen Tabellenbereich zu laden.

0x0000000008000000 USE
Es wurde versucht, eine Tabelle in einem Tabellenbereich zu erstellen bzw. zu prüfen, ob das Zugriffsrecht USE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER lautet.

0x0000000010000000 SET SESSION_USER
Es wurde versucht, eine Anweisung SET SESSION_USER auszuführen.

0x0000000020000000 FLUSH
Es wurde versucht, eine Anweisung FLUSH auszuführen.

0x0000000040000000 STORE
Es wurde versucht, die Werte einer reoptimierten Anweisung in der Tabelle EXPLAIN_PREDICATE anzuzeigen.

0x0000000400000000 TRANSFER
Es wurde versucht, ein Objekt zu übertragen.

0x0000000800000000 ALTER_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht ALTER with GRANT erteilt wurde.

0x0000001000000000 DELETE_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht DELETE with GRANT erteilt wurde.

0x0000002000000000 INDEX_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht INDEX with GRANT erteilt wurde.

0x0000004000000000 INSERT_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht INSERT with GRANT erteilt wurde.

0x0000008000000000 SELECT_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht SELECT with GRANT erteilt wurde.

0x0000010000000000 UPDATE_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht UPDATE with GRANT erteilt wurde.

0x0000020000000000 REFERENCE_WITH_GRANT
Es wurde versucht, zu prüfen, ob das Zugriffsrecht REFERENCE with GRANT erteilt wurde.

0x0000040000000000 USAGE
Es wurde versucht, eine Reihenfolge oder ein XSR-Objekt zu verwenden bzw. zu prüfen, ob das Zugriffsrecht USAGE erteilt wurde, wenn das Prüfereignis CHECKING_TRANSFER ist.

0x0000080000000000 SET ROLE
Es wurde versucht, eine Rolle festzulegen.

0x0000100000000000 EXPLICIT_TRUSTED_CONNECTION
Es wurde versucht, eine explizite gesicherte Verbindung herzustellen.

0x0000200000000000 IMPLICIT_TRUSTED_CONNECTION
Es wurde versucht, eine implizite gesicherte Verbindung herzustellen.

0x0000400000000000 READ
Es wurde versucht, eine globale Variable zu lesen.

0x0000800000000000 WRITE
Es wurde versucht, eine globale Variable zu schreiben.

0x0001000000000000 SWITCH_USER
Es wurde versucht, eine Benutzer-ID über eine explizite gesicherte Verbindung zu wechseln.

0x0002000000000000 AUDIT_USING
Es wurde versucht, einem Objekt eine Prüfrichtlinie zuzuordnen.

0x0004000000000000 AUDIT_REPLACE
Es wurde versucht, eine Prüfrichtlinienzuordnung eines Objekts zu ersetzen.

0x0008000000000000 AUDIT_REMOVE
Es wurde versucht, eine Prüfrichtlinienzuordnung eines Objekts zu entfernen.

0x0010000000000000 AUDIT_ARCHIVE
Es wurde versucht, das Prüfprotokoll zu archivieren.

0x0020000000000000 AUDIT_EXTRACT
Es wurde versucht, das Prüfprotokoll zu extrahieren.

0x0040000000000000 AUDIT_LIST_LOGS
Es wurde versucht, die Prüfprotokolle aufzulisten.

Prüfsatzaufbau für OBJMAINT-Ereignisse

Das Format des Prüfsatzes für OBJMAINT-Ereignisse wird in der folgenden Tabelle gezeigt.

Beispiel für einen Prüfsatz:

```
timestamp=1998-06-24-08.42.41.957524;
category=OBJMAINT;
audit event=CREATE_OBJECT;
event correlator=3;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=AUDIT;
object type=TABLE;
```

Tabelle 42. Prüfsatzaufbau für OBJMAINT-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: OBJMAINT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie OBJMAINT unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(256)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde.
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde.
Object Type (Objekttyp)	VARCHAR(32)	Der Typ des Objekts, für das das Prüfereignis generiert wurde. Mögliche Werte: siehe Abschnitt „Prüfsatzobjekttypen".
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 42. Prüfsatzaufbau für OBJMAINT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Security Policy Name (Name der Sicherheitsrichtlinie)	VARCHAR(128)	Der Name der Sicherheitsrichtlinie, wenn der Objekttyp TABLE ist und dieser Tabelle eine Sicherheitsrichtlinie zugeordnet ist.
Alter Action (ALTER-Aktion)	VARCHAR(32)	Die spezielle ALTER-Operation. Mögliche Werte: • ADD_PROTECTED_COLUMN (Geschützte Spalte hinzufügen) • ADD_COLUMN_PROTECTION (Spaltenschutz hinzufügen) • DROP_COLUMN_PROTECTION (Spaltenschutz löschen) • ADD_ROW_PROTECTION (Zeilenschutz hinzufügen) • ADD_SECURITY_POLICY (Sicherheitsrichtlinie hinzufügen) • ADD_ELEMENT (Element hinzufügen) • ADD COMPONENT (Komponente hinzufügen) • USE GROUP AUTHORIZATIONS (Gruppenberechtigungen verwenden) • IGNORE GROUP AUTHORIZATIONS (Gruppenberechtigungen ignorieren) • USE ROLE AUTHORIZATIONS (Rollenberechtigungen verwenden) • IGNORE ROLE AUTHORIZATIONS (Rollenberechtigungen ignorieren) • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL (Nicht berechtigten Sicherheitskennsatz überschreiben) • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL (Nicht berechtigten Sicherheitskennsatz einschränken)
Protected Column Name (Name der geschützten Spalte)	VARCHAR(128)	Wenn 'Alter Action' den Wert ADD_COLUMN_PROTECTION oder DROP_COLUMN_PROTECTION hat, ist dies der Name der betroffenen Spalte.
Column Security Label (Sicherheitskennsatz der Spalte)	VARCHAR(128)	Der Sicherheitskennsatz, der die Spalte schützt, die im Feld 'Column Name' (Spaltenname) angegeben ist.
Security Label Column Name (Name der Spalte mit dem Sicherheitskennsatz)	VARCHAR(128)	Der Name der Spalte, die den Sicherheitskennsatz enthält, der die Zeile schützt.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.

Tabelle 42. Prüfsatzaufbau für OBJMAINT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Client Accounting String (Abrechnungs-zeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.

Prüfsatzaufbau für SECMAINT-Ereignisse

Das Format des Prüfsatzes für SECMAINT-Ereignisse wird in der folgenden Tabelle gezeigt.

Beispiel für einen Prüfsatz:

```
timestamp=1998-06-24-11.57.45.188101;
category=SECMAINT;
audit event=GRANT;
event correlator=4;
event status=0;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.boss.980624155728;
application name=db2bp;
package schema=NULLID;
package name=SQLC28A1;
package section=0;
object schema=BOSS;
object name=T1;
object type=TABLE;
grantor=BOSS;
grantee=WORKER;
grantee type=USER;
privilege=SELECT;
```

Tabelle 43. Prüfsatzaufbau für SECMAINT-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: SECMAINT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie SECMAINT unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Object Schema (Objektschema)	VARCHAR(128)	Das Schema des Objekts, für das das Prüfereignis generiert wurde. Wenn das Objekttypfeld den Wert ACCESS_RULE hat, enthält dieses Feld den Namen der Sicherheitsrichtlinie, die der Regel zugeordnet ist. Der Name der Regel wird im Feld für den Objektnamen gespeichert. Wenn das Objekttypfeld den Wert SECURITY_LABEL hat, enthält dieses Feld den Namen der Sicherheitsrichtlinie, zu der der Sicherheitskennsatz gehört. Der Name des Sicherheitskennsatzes wird im Feld für den Objektnamen gespeichert.

Tabelle 43. Prüfsatzaufbau für SECMAINT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Object Name (Objektname)	VARCHAR(128)	Der Name des Objekts, für das das Prüfereignis generiert wurde. Stellt einen Rollennamen dar, wenn das Prüfereignis eines der folgenden ist: ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE, ALTER_DEFAULT_ROLE, ADD_USER, DROP_USER, ALTER_USER_ADD_ROLE, ALTER_USER_DROP_ROLE oder ALTER_USER_AUTHENTICATION. Wenn das Objekttypfeld den Wert ACCESS_RULE hat, enthält dieses Feld den Namen der Regel. Der Name der Sicherheitsrichtlinie, die der Regel zugeordnet ist, wird im Feld für das Objektschema gespeichert. Wenn das Objekttypfeld den Wert SECURITY_LABEL hat, enthält dieses Feld den Namen des Sicherheitskennsatzes. Der Name der Sicherheitsrichtlinie, der er zugeordnet ist, wird im Feld für das Objektschema gespeichert.
Object Type (Objekttyp)	VARCHAR(32)	Der Typ des Objekts, für das das Prüfereignis generiert wurde. Mögliche Werte: siehe Abschnitt „Prüfsatzobjekttypen". Der Wert ist ROLE, wenn das Prüfereignis eines der folgenden ist: ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE, ALTER_DEFAULT_ROLE, ADD_USER, DROP_USER, ALTER_USER_ADD_ROLE, ALTER_USER_DROP_ROLE oder ALTER_USER_AUTHENTICATION.
Grantor (Berechtigungsgeber)	VARCHAR(128)	Die ID des Benutzers, der das Zugriffsrecht bzw. die Berechtigung erteilt oder entzogen hat.
Grantee (Berechtigter)	VARCHAR(128)	Die ID des Berechtigten, dem ein Zugriffsrecht oder eine Berechtigung erteilt oder entzogen wurde. Stellt ein Objekt für einen gesicherten Kontext dar, wenn das Prüfereignis eines der folgenden ist: ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE, ALTER_DEFAULT_ROLE, ADD_USER, DROP_USER, ALTER_USER_ADD_ROLE, ALTER_USER_DROP_ROLE oder ALTER_USER_AUTHENTICATION.
Grantee Type (Typ des Berechtigten)	VARCHAR(32)	Der Typ des Berechtigten, dem eine Berechtigung erteilt oder entzogen wurde. Mögliche Werte: USER, GROUP, ROLE, AMBIGUOUS. Oder TRUSTED_CONTEXT, wenn das Prüfereignis eines der folgenden ist: ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE, ALTER_DEFAULT_ROLE, ADD_USER, DROP_USER, ALTER_USER_ADD_ROLE, ALTER_USER_DROP_ROLE oder ALTER_USER_AUTHENTICATION.
Privilege or Authority (Zugriffsrecht oder Berechtigung)	CHAR(18)	Gibt an, welcher Typ von Zugriffsrecht oder Berechtigung erteilt oder entzogen wurde. Mögliche Werte: siehe Abschnitt „SECMAINT-Zugriffsrechte bzw. -Berechtigungen". Der Wert ist ROLE MEMBERSHIP, wenn das Prüfereignis eines der folgenden ist: ADD_DEFAULT_ROLE, DROP_DEFAULT_ROLE, ALTER_DEFAULT_ROLE, ADD_USER, DROP_USER, ALTER_USER_ADD_ROLE, ALTER_USER_DROP_ROLE oder ALTER_USER_AUTHENTICATION.
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 43. Prüfsatzaufbau für SECMAINT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Access Type (Zugriffstyp)	VARCHAR(32)	Der Zugriffstyp, dem ein Sicherheitskennsatz erteilt wurde. Gültige Werte: • READ • WRITE • ALL Der Zugriffstyp, für den eine Sicherheitsrichtlinie geändert wurde. Gültige Werte: • USE GROUP AUTHORIZATIONS • IGNORE GROUP AUTHORIZATIONS • USE ROLE AUTHORIZATIONS • IGNORE ROLE AUTHORIZATIONS • OVERRIDE NOT AUTHORIZED WRITE SECURITY LABEL • RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL
Assumable Authid (Anzunehmende Berechtigungs-ID)	VARCHAR(128)	Wenn es sich bei dem erteilten Zugriffsrecht um das Zugriffsrecht SETSESSIONUSER handelt, ist dies die Berechtigungs-ID, die der Berechtigte als Sitzungsbenutzer definieren darf.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Grantor Type (Typ des Berechtigungsgebers)	VARCHAR(32)	Der Typ des Berechtigungsgebers. Mögliche Werte: USER.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context User (Benutzer des gesicherten Kontexts)	VARCHAR(128)	Gibt einen Benutzer des gesicherten Kontexts an, wenn das Prüfereignis ADD_USER oder DROP_USER ist.
Trusted Context User Authentication (Benutzerauthentifizierung für gesicherten Kontext)	INTEGER	Gibt die Authentifizierungseinstellung für einen Benutzer des gesicherten Kontexts an, wenn das Prüfereignis ADD_USER, DROP_USER oder ALTER_USER_AUTHENTICATION ist. 1 : Authentifizierung erforderlich 0 : Authentifizierung nicht erforderlich
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.

Tabelle 43. Prüfsatzaufbau für SECMAINT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.

SECMAINT-Zugriffsrechte bzw. -Berechtigungen

Die folgende Liste zeigt die möglichen SECMAINT-Zugriffsrechte bzw. -Berechtigungen.

0x0000000000000001 Control Table
Zugriffsrecht CONTROL einer Tabelle oder Sicht erteilt oder entzogen.

0x0000000000000002 ALTER
Zugriffsrecht zum Ändern einer Tabelle oder Sequenz erteilt oder widerrufen.

0x0000000000000004 ALTER with GRANT
Zugriffsrecht zum Ändern einer Tabelle oder Sequenz mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder widerrufen.

0x0000000000000008 DELETE TABLE
Zugriffsrecht DELETE zum Löschen einer Tabelle oder Sicht erteilt oder entzogen.

0x0000000000000010 DELETE TABLE with GRANT
Zugriffsrecht DELETE zum Löschen einer Tabelle mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000000020 Table Index
Zugriffsrecht einem Index erteilt oder entzogen.

0x0000000000000040 Table Index with GRANT
Zugriffsrecht mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten einem Index erteilt oder entzogen.

0x0000000000000080 Table INSERT
Zugriffsrecht INSERT zum Einfügen für eine Tabelle oder Sicht erteilt oder entzogen.

0x0000000000000100 Table INSERT with GRANT
Zugriffsrecht INSERT zum Einfügen für eine Tabelle oder Sicht mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000000200 Table SELECT
Zugriffsrecht SELECT zum Auswählen in einer Tabelle erteilt oder entzogen.

0x0000000000000400 Table SELECT with GRANT
Zugriffsreicht SELECT zum Auswählen in einer Tabelle mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000000800 Table UPDATE
Zugriffsrecht UPDATE zum Aktualisieren für eine Tabelle oder Sicht erteilt oder entzogen.

0x0000000000001000 Table UPDATE with GRANT
Zugriffsrecht UPDATE zum Aktualisieren für eine Tabelle oder Sicht mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000002000 Table REFERENCE
Zugriffsrecht REFERENCE zum Verweisen auf eine Tabelle erteilt oder entzogen.

j0x0000000000004000 Table REFERENCE with GRANT
Zugriffsrecht REFERENCE zum Verweisen auf eine Tabelle mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000020000 CREATEIN Schema
Zugriffsrecht CREATEIN zum Erstellen in einem Schema erteilt oder entzogen.

0x0000000000040000 CREATEIN Schema with GRANT
Zugriffsrecht CREATEIN zum Erstellen in einem Schema mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000080000 DROPIN Schema
Zugriffsrecht DROPIN zum Löschen in einem Schema erteilt oder entzogen.

0x0000000000100000 DROPIN Schema with GRANT
Zugriffsrecht DROPIN zum Löschen in einem Schema mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000200000 ALTERIN Schema
Zugriffsrecht ALTERIN zum Ändern in einem Schema erteilt oder entzogen.

0x0000000000400000 ALTERIN Schema with GRANT
Zugriffsrecht ALTERIN zum Ändern in einem Schema mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000000800000 DBADM Authority
Berechtigung DBADM erteilt oder entzogen.

0x0000000001000000 CREATETAB Authority
CREATETAB-Berechtigung erteilt oder entzogen.

0x0000000002000000 BINDADD Authority
Berechtigung BINDADD erteilt oder entzogen.

0x0000000004000000 CONNECT Authority
Berechtigung CONNECT erteilt oder entzogen.

0x0000000008000000 Create not fenced Authority
Berechtigung 'Create not fenced' (nicht abgeschirmte erstellen) erteilt oder entzogen.

0x0000000010000000 Implicit Schema Authority
Berechtigung 'Implicit schema' erteilt oder entzogen.

0x0000000020000000 Server PASSTHRU
Zugriffsrecht für die Verwendung der Durchgriffsfunktion für diesen Server (Datenquelle föderierter Datenbanken) erteilt oder entzogen.

0x0000000040000000 ESTABLISH TRUSTED CONNECTION
Gesicherte Verbindung wurde erstellt.

0x0000000100000000 Table Space USE
Zugriffsrecht USE zum Erstellen einer Tabelle in einem Tabellenbereich erteilt oder entzogen.

0x0000000200000000 Table Space USE with GRANT
Zugriffsrecht USE zum Erstellen einer Tabelle in einem Tabellenbereich mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000000400000000 Column UPDATE
Zugriffsrecht UPDATE zum Aktualisieren einer oder mehrerer bestimmter Spalten einer Tabelle erteilt oder entzogen.

0x0000000800000000 Column UPDATE with GRANT
Zugriffsrecht UPDATE zum Aktualisieren einer oder mehrerer bestimmter Spalten einer Tabelle mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000001000000000 Column REFERENCE
Zugriffsrecht REFERENCE zum Verweisen auf eine oder mehrere bestimmte Spalten einer Tabelle erteilt oder entzogen.

0x0000002000000000 Column REFERENCE with GRANT
Zugriffsrecht REFERENCE zum Verweisen auf eine oder mehrere bestimmte Spalten einer Tabelle mit der Möglichkeit zum Erteilen von Zugriffsrechten (GRANT) erteilt oder entzogen.

0x0000004000000000 LOAD Authority
Berechtigung LOAD erteilt oder entzogen.

0x0000008000000000 Package BIND
Zugriffsrecht BIND für ein Paket erteilt oder entzogen.

0x0000010000000000 Package BIND with GRANT
Zugriffsrecht BIND für ein Paket mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000020000000000 EXECUTE
Zugriffsrecht EXECUTE für ein Paket oder eine Routine erteilt oder entzogen.

0x0000040000000000 EXECUTE with GRANT
Zugriffsrecht EXECUTE für ein Paket oder eine Routine mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000080000000000 EXECUTE IN SCHEMA
Zugriffsrecht EXECUTE für alle Routinen in einem Schema erteilt oder entzogen.

0x0000100000000000 EXECUTE IN SCHEMA with GRANT
Zugriffsrecht EXECUTE für alle Routinen in einem Schema mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000200000000000 EXECUTE IN TYPE
Zugriffsrecht EXECUTE für alle Routinen in einem Typ erteilt oder entzogen.

0x0000400000000000 EXECUTE IN TYPE with GRANT
Zugriffsrecht EXECUTE für alle Routinen in einem Typ mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0000800000000000 CREATE EXTERNAL ROUTINE
Zugriffsrecht CREATE EXTERNAL ROUTINE erteilt oder entzogen.

0x0001000000000000 QUIESCE_CONNECT
Zugriffsrecht QUIESCE_CONNECT erteilt oder entzogen.

0x0004000000000000 SECADM Authority
Berechtigung SECADM erteilt oder entzogen.

0x0008000000000000 USAGE Authority
Zugriffsrecht USAGE für eine Sequenz erteilt oder entzogen.

0x0010000000000000 USAGE with GRANT Authority
Zugriffsrecht USAGE für eine Sequenz mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0020000000000000 WITH ADMIN Option
Zugriffsrecht WITH ADMIN OPTION für eine Rolle erteilt oder entzogen.

0x0040000000000000 SETSESSIONUSER Privilege
Zugriffsrecht SETSESSIONUSER erteilt oder entzogen.

0x0080000000000000 Exemption
Freistellung erteilt oder entzogen.

0x0100000000000000 Security label
Sicherheitskennsatz erteilt oder entzogen.

0x0200000000000000 WRITE with GRANT
Zugriffsrecht WRITE zum Schreiben einer globalen Variablen mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x0400000000000000 Role Membership
Rollenzugehörigkeit, die erteilt oder entzogen wurde.

0x0800000000000000 Role Membership with ADMIN Option
Rollenzugehörigkeit mit dem Zugriffsrecht WITH ADMIN OPTION, die erteilt oder entzogen wurde.

0x1000000000000000 READ
Zugriffsrecht READ zum Lesen einer globalen Variablen erteilt oder entzogen.

0x2000000000000000 READ with GRANT
Zugriffsrecht READ zum Lesen einer globalen Variablen mit der Möglichkeit zum Erteilen (GRANT) von Zugriffsrechten erteilt oder entzogen.

0x4000000000000000 WRITE
Zugriffsrecht WRITE zum Schreiben einer globalen Variablen erteilt oder entzogen.

Prüfsatzaufbau für SYSADMIN-Ereignisse

Die folgende Tabelle zeigt den Prüfsatzaufbau für SYSADMIN-Ereignisse.

Beispiel für einen Prüfsatz:

```
timestamp=1998-06-24-11.54.04.129923;
category=SYSADMIN;
audit event=DB2AUDIT;
event correlator=1;
event status=0;
userid=boss;authid=BOSS;
application id=*LOCAL.boss.980624155404;
application name=db2audit;
```

Tabelle 44. Prüfsatzaufbau für SYSADMIN-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: SYSADMIN
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie SYSADMIN unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 44. Prüfsatzaufbau für SYSADMIN-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.

Prüfsatzaufbau für VALIDATE-Ereignisse

Das Format des Prüfsatzes für VALIDATE-Ereignisse wird in der folgenden Tabelle gezeigt.

Beispiel für einen Prüfsatz:

```
timestamp=2007-05-07-10.30.51.585626;
category=VALIDATE;
audit event=AUTHENTICATION;
event correlator=1;
event status=0;
userid=newton;
authid=NEWTON;
execution id=gstager;
application id=*LOCAL.gstager.070507143051;
application name=db2bp;
auth type=SERVER;
plugin name=IBMOSauthserver;
```

Tabelle 45. Prüfsatzaufbau für VALIDATE-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: VALIDATE
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie VALIDATE unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses.
Execution ID (Ausführungs-ID)	VARCHAR(1024)	Die Ausführungs-ID, die beim Auftreten des Prüfereignisses verwendet wurde.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 45. Prüfsatzaufbau für VALIDATE-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Authentication Type (Authentifizierungstyp)	VARCHAR(32)	Der Authentifizierungstyp beim Auftreten des Prüfereignisses.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Plug-in Name (Plug-in-Name)	VARCHAR(32)	Der Name des Plug-ins, das beim Auftreten des Prüfereignisses verwendet wurde.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Der Name der Rolle, die durch den gesicherten Kontext übernommen wurde.

Prüfsatzaufbau für CONTEXT-Ereignisse

Die folgende Tabelle zeigt den Prüfsatzaufbau für CONTEXT-Ereignisse.

Beispiel für einen Prüfsatz:

```
timestamp=1998-06-24-08.42.41.476840;
category=CONTEXT;
audit event=EXECUTE_IMMEDIATE;
event correlator=3;
database=FOO;
userid=boss;
authid=BOSS;
application id=*LOCAL.newton.980624124210;
application name=testapp;
package schema=NULLID;
package name=SQLC28A1;
package section=203;
text=create table audit(c1 char(10), c2 integer);
```

Tabelle 46. Prüfsatzaufbau für CONTEXT-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: CONTEXT
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie CONTEXT unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses. Wenn das Prüfereignis SWITCH_USER ist, stellt dieses Feld die Benutzer-ID dar, zu der gewechselt wird.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID zum Zeitpunkt des Prüfereignisses. Wenn das Prüfereignis SWITCH_USER ist, stellt dieses Feld die Berechtigungs-ID dar, zu der gewechselt wird.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 46. Prüfsatzaufbau für CONTEXT-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section Number (Paketabschnittsnummer)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Statement Text (Anweisungstext)	CLOB(8M)	Text der SQL- oder XQuery-Anweisung, falls vorhanden. Null, wenn kein Text für die SQL- oder XQuery-Anweisung verfügbar ist.
Package Version (Paketversion)	VARCHAR(64)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust Type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_CONNECTION EXPLICIT_TRUSTED_CONNECTION
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.

Prüfsatzaufbau für EXECUTE-Ereignisse

Die folgende Tabelle beschreibt alle Felder, die von der Kategorie EXECUTE der Prüffunktion geprüft werden.

Beispiel für einen Prüfsatz:

Anmerkung: Im Unterschied zu anderen Prüfkategorien können bei der Kategorie EXECUTE, wenn das Prüfprotokoll in einem Tabellenformat angezeigt wird, mehrere Zeilen für die Beschreibung ein und desselben Ereignisses angezeigt werden. Der erste Prüfsatz beschreibt das Hauptereignis, und die Ereignisspalte enthält das Schlüsselwort STATEMENT. Die übrigen Zeilen beschreiben die Parametermarken oder Hostvariablen (jeweils in einer Zeile), und die entsprechende Ereignisspalte enthält das Schlüsselwort DATA. Wenn das Prüfprotokoll im Berichtsformat angezeigt wird, ist jeweils ein Prüfsatz vorhanden, jedoch mit mehreren Einträgen für den Anweisungswert ('Statement Value'). Das Schlüsselwort DATA wird nur im Tabellenformat verwendet.

```
timestamp=2006-04-10-13.20.51.029203;
  category=EXECUTE;
  audit event=STATEMENT;
  event correlator=1;
  event status=0;
  database=SAMPLE;
  userid=smith;
  authid=SMITH;
  session authid=SMITH;
  application id=*LOCAL.prodrig.060410172044;
  application name=myapp;
  package schema=NULLID;
  package name=SQLC2F0A;
  package section=201;
  uow id=2;
  activity id=3;
  statement invocation id=0;
  statement nesting level=0;
  statement text=SELECT * FROM DEPARTMENT WHERE DEPTNO = ? AND DEPTNAME = ?;
  statement isolation level=CS;
  compilation environment=
    isolation level=CS
    query optimization=5
    min_dec_div_3=NO
    degree=1
    sqlrules=DB2
    refresh age=+00000000000000.000000
    schema=SMITH
    maintained table type=SYSTEM
    resolution timestamp=2006-06-29-20.32.13.000000
    federated asynchrony=0;
  value index=0;
  value type=CHAR;
  value data=C01;
  value index=1;
  value type=VARCHAR;
  value index=INFORMATION CENTER;
```

Tabelle 47. Prüfsatzaufbau für EXECUTE-Ereignisse

NAME	FORMAT	BESCHREIBUNG
Timestamp (Zeitmarke)	CHAR(26)	Datum und Uhrzeit des Prüfereignisses
Category (Kategorie)	CHAR(8)	Die Kategorie des Prüfereignisses. Mögliche Werte: EXECUTE
Audit Event (Prüfereignis)	VARCHAR(32)	Spezifisches Prüfereignis. Eine Liste gültiger Werte finden Sie im Abschnitt über die Kategorie EXECUTE unter „Prüfereignisse" auf Seite 263.
Event Correlator (Ereigniskorrelationswert)	INTEGER	Korrelationskennung für die geprüfte Operation. Kann zur Erkennung verwendet werden, welche Prüfsätze einem einzelnen Ereignis zugeordnet sind.
Event Status (Ereignisstatus)	INTEGER	Status des Prüfereignisses, dargestellt durch einen SQLCODE, für den Folgendes gilt: Erfolgreiches Ereignis > = 0 und Fehlgeschlagenes Ereignis < 0
Database Name (Datenbankname)	CHAR(8)	Name der Datenbank, für die das Ereignis generiert wurde. Nicht belegt, wenn es sich um ein Prüfereignis auf Instanzebene handelt.
User ID (Benutzer-ID)	VARCHAR(1024)	Die Benutzer-ID zum Zeitpunkt des Prüfereignisses.
Authorization ID (Berechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID der Anweisung zum Zeitpunkt des Prüfereignisses.
Session Authorization ID (Sitzungsberechtigungs-ID)	VARCHAR(128)	Die Berechtigungs-ID der Sitzung zum Zeitpunkt des Prüfereignisses.
Origin Node Number (Nummer des Ursprungsknotens)	SMALLINT	Nummer des Knotens, auf dem das Prüfereignis aufgetreten ist.
Coordinator Node Number (Nummer des Koordinatorknotens)	SMALLINT	Die Knotennummer des Koordinatorknotens.
Application ID (Anwendungs-ID)	VARCHAR(255)	Die ID der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.
Application Name (Anwendungsname)	VARCHAR(1024)	Der Name der Anwendung, die beim Auftreten des Prüfereignisses verwendet wurde.

Tabelle 47. Prüfsatzaufbau für EXECUTE-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Client User ID (Clientbenutzer-ID)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT USERID zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Accounting String (Abrechnungszeichenfolge für Client)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_ACCTNG zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Workstation Name (Name der Client-Workstation)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_WRKSTNNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Client Application Name (Name der Clientanwendung)	VARCHAR(255)	Der Wert des Sonderregisters CURRENT CLIENT_APPLNAME zu dem Zeitpunkt, als das Prüfereignis auftrat.
Trusted Context Name (Name des gesicherten Kontexts)	VARCHAR(128)	Der Name des gesicherten Kontexts, der der gesicherten Verbindung zugeordnet ist.
Connection Trust type (Typ der gesicherten Verbindung)	INTEGER	Mögliche Werte: IMPLICIT_TRUSTED_ CONNECTION und EXPLICIT_TRUSTED_ CONNECTION.
Role Inherited (Übernommene Rolle)	VARCHAR(128)	Die Rolle, die durch eine gesicherte Verbindung übernommen wurde.
Package Schema (Paketschema)	VARCHAR(128)	Das Schema des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Name (Paketname)	VARCHAR(128)	Der Name des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Section (Paketabschnitt)	SMALLINT	Die Abschnittsnummer innerhalb des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Package Version (Paketversion)	VARCHAR(164)	Die Version des Pakets, das beim Auftreten des Prüfereignisses verwendet wurde.
Local Transaction ID (Lokale Transaktions-ID)	VARCHAR(10) FOR BIT DATA	Die lokale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist die Struktur SQLU_TID, die Teil der Transaktionsprotokolle ist.

Tabelle 47. Prüfsatzaufbau für EXECUTE-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Global Transaction ID (Globale Transaktions-ID)	VARCHAR(30) FOR BIT DATA	Die globale Transaktions-ID, die beim Auftreten des Prüfereignisses verwendet wurde. Dies ist das Datenfeld in der Struktur SQLP_GXID, die Teil der Transaktionsprotokolle ist.
UOW ID (Kennung der UOW)	BIGINT	Die Kennung der UOW (Unit of Work, Arbeitseinheit), aus der eine Aktivität stammt. Dieser Wert ist innerhalb einer Anwendungs-ID für jede UOW eindeutig.
Activity ID (Aktivitäts-ID)	BIGINT	Die eindeutige Aktivitäts-ID innerhalb der UOW.
Statement Invocation ID (Aufruf-ID der Anweisung)	BIGINT	Die Kennung (ID) des Routinenaufrufs, in dem die SQL-Anweisung ausgeführt wurde. Der Wert gibt die Anzahl der Routinenaufrufe auf der aktuellen Verschachtelungsebene an, die erfolgt sind, während diese Ebene in der Anwendung aktiv war. Mithilfe dieses Elements und der Verschachtelungsebene der Anweisung können Sie den Aufruf einer bestimmten SQL-Anweisung eindeutig bestimmen.
Statement Nesting Level (Verschachtelungsebene der Anweisung)	BIGINT	Die aktive Verschachtelungs- bzw. Rekursionsebene, als die Anweisung ausgeführt wurde. Jede Verschachtelungsebene entspricht dem verschachtelten oder rekursivem Aufruf einer gespeicherten Prozedur oder benutzerdefinierten Funktion (UDF).
Activity Type (Aktivitätstyp)	VARCHAR(32)	Der Typ der Aktivität. Mögliche Werte: • READ_DML • WRITE_DML • DDL • CALL • NONE
Statement Text (Anweisungstext)	CLOB(8M)	Text der SQL- oder XQuery-Anweisung, falls vorhanden.

Tabelle 47. Prüfsatzaufbau für EXECUTE-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Statement Isolation Level (Isolationsstufe der Anweisung)	CHAR(8)	Der aktive Isolationswert für die Anweisung während der Ausführung. Mögliche Werte: • NONE (keine Isolation angegeben) • UR (Nicht festgeschriebener Lesevorgang) • CS (Cursorstabilität) • RS (Lesestabilität) • RR (Wiederholtes Lesen)
Compilation Environment Description (Beschreibung der Kompilierungsumgebung)	BLOB(8K)	Die Kompilierungsumgebung, die beim Kompilieren der SQL-Anweisung verwendet wurde. Sie können dieses Element als Eingabe für die Tabellenfunktion COMPILATION_ENV oder als Angabe in der SQL-Anweisung SET COMPILATION ENVIRONMENT verwenden.
Rows Modified (Geänderte Zeilen)	INTEGER	Enthält die Gesamtanzahl der gelöschten, eingefügten oder aktualisierten Zeilen, die das Ergebnis folgender Aktivitäten sind: • Umsetzung von Integritätsbedingungen nach einer erfolgreichen DELETE-Operation • Verarbeitung von ausgelösten SQL-Anweisungen aus aktivierten Triggern Wenn Compound-SQL-Anweisungen aufgerufen werden, enthält dieses Feld die summierte Anzahl solcher Zeilen für alle Unteranweisungen. Wenn in einigen Fällen ein Fehler festgestellt wird, enthält dieses Feld einen negativen Wert, der einen Zeiger für einen internen Fehler darstellt. Dieser Wert ist äquivalent mit dem Feld 'sqlerrd(5)' des SQL-Kommunikationsbereichs (SQLCA).

Tabelle 47. Prüfsatzaufbau für EXECUTE-Ereignisse (Forts.)

NAME	FORMAT	BESCHREIBUNG
Rows Returned (Zurückgegebene Zeilen)	BIGINT	Enthält die Gesamtanzahl der Zeilen, die durch die Anweisung zurückgegeben wurden.
Savepoint ID (Sicherungspunkt-ID)	BIGINT	Die aktive Sicherungspunkt-ID für die Anweisung während der Ausführung. Wenn das Prüfereignis SAVEPOINT, RELEASE_SAVEPOINT oder ROLLBACK_SAVEPOINT ist, stellt die Sicherungspunkt-ID den Sicherungspunkt dar, der gesetzt, freigegeben bzw. rückgängig gemacht wird.
Statement Value Index (Anweisungswertindex)	INTEGER	Die Position der in der SQL-Anweisung verwendeten Eingabeparametermarke bzw. Hostvariablen.
Statement Value Type (Anweisungswerttyp)	CHAR(16)	Eine Zeichenfolgedarstellung des Typs eines Datenwerts, der der SQL-Anweisung zugeordnet ist. Beispiele für mögliche Werte sind INTEGER oder CHAR.
Statement Value Data (Anweisungswertdaten)	CLOB(128K)	Eine Zeichenfolgedarstellung eines Datenwerts für die SQL-Anweisung. Parameter für LOB-, LONG- und XML-Datentypen sowie strukturierte Datentypen sind nicht enthalten. Felder für Datum, Zeit und Zeitmarken werden im ISO-Format erfasst.

Prüfereignisse

Für jede Prüfkategorie können bestimmte Ereignistypen Prüfsätze erstellen.

Ereignisse für die Kategorie AUDIT

- ALTER_AUDIT_POLICY
- ARCHIVE
- AUDIT_REMOVE
- AUDIT_REPLACE
- AUDIT_USING
- CONFIGURE
- CREATE_AUDIT_POLICY
- DB2AUD
- DROP_AUDIT_POLICY
- EXTRACT
- FLUSH

- LIST_LOGS
- PRUNE (ab Version 9.5 nicht mehr generiert)
- START
- STOP
- UPDATE_ADMIN_CFG

Ereignisse für die Kategorie CHECKING

- CHECKING_FUNCTION
- CHECKING_MEMBERSHIP_IN_ROLES
- CHECKING_OBJECT
- CHECKING_TRANSFER

Ereignisse für die Kategorie CONTEXT

Tabelle 48. Ereignisse für die Kategorie CONTEXT

CONNECT	SET_APPL_PRIORITY
CONNECT_RESET	RESET_DB_CFG
ATTACH	GET_DB_CFG
DETACH	GET_DFLT_CFG
DARI_START	UPDATE_DBM_CFG
DARI_STOP	SET_MONITOR
BACKUP_DB	GET_SNAPSHOT
RESTORE_DB	ESTIMATE_SNAPSHOT_SIZE
ROLLFORWARD_DB	RESET_MONITOR
OPEN_TABLESPACE_QUERY	OPEN_HISTORY_FILE
FETCH_TABLESPACE	CLOSE_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	FETCH_HISTORY_FILE
OPEN_CONTAINER_QUERY	SET_RUNTIME_DEGREE
CLOSE_CONTAINER_QUERY	UPDATE_AUDIT
FETCH_CONTAINER_QUERY	DBM_CFG_OPERATION
SET_TABLESPACE_CONTAINERS	DISCOVER
GET_TABLESPACE_STATISTIC	OPEN_CURSOR
READ_ASYNC_LOG_RECORD	CLOSE_CURSOR
QUIESCE_TABLESPACE	FETCH_CURSOR
LOAD_TABLE	EXECUTE
UNLOAD_TABLE	EXECUTE_IMMEDIATE
UPDATE_RECOVERY_HISTORY	PREPARE
PRUNE_RECOVERY_HISTORY	DESCRIBE
SINGLE_TABLESPACE_QUERY	BIND
LOAD_MSG_FILE	REBIND
UNQUIESCE_TABLESPACE	RUNSTATS
ENABLE_MULTIPAGE	REORG
DESCRIBE_DATABASE	REDISTRIBUTE
DROP_DATABASE	COMMIT
CREATE_DATABASE	ROLLBACK
ADD_NODE	REQUEST_ROLLBACK
FORCE_APPLICATION	IMPLICIT_REBIND
	EXTERNAL_CANCEL
	SWITCH_USER

Ereignisse für die Kategorie EXECUTE

- COMMIT: Ausführung einer Anweisung COMMIT
- CONNECT: Herstellung einer Datenbankverbindung
- CONNECT RESET: Beendigung einer Datenbankverbindung
- DATA: Werte von Hostvariablen oder Parametermarken für die Anweisung

  Dieses Ereignis wird für jede Hostvariable oder Parametermarke in der Anweisung wiederholt. Es ist nur in den Daten enthalten, die aus dem Prüfprotokoll in Dateien mit begrenzter Satzlänge extrahiert wurden.
- GLOBAL COMMIT: Ausführung einer Anweisung COMMIT innerhalb einer globalen Transaktion
- GLOBAL ROLLBACK: Ausführung einer Anweisung ROLLBACK innerhalb einer globalen Transaktion
- RELEASE SAVEPOINT: Ausführung einer Anweisung RELEASE SAVEPOINT
- ROLLBACK: Ausführung einer Anweisung ROLLBACK
- SAVEPOINT: Ausführung einer Anweisung SAVEPOINT
- STATEMENT: Ausführung einer SQL-Anweisung
- SWITCH USER: Wechseln eines Benutzers innerhalb einer gesicherten Verbindung

Ereignisse für die Kategorie OBJMAINT

- ALTER_OBJECT (werden nur generiert, wenn geschützte Tabellen geändert werden)
- CREATE_OBJECT
- DROP_OBJECT
- RENAME_OBJECT

Ereignisse für die Kategorie SECMAINT

- ADD_DEFAULT_ROLE
- ADD_USER
- ALTER_DEFAULT_ROLE
- ALTER SECURITY POLICY
- ALTER_USER_ADD_ROLE
- ALTER_USER_AUTHENTICATION
- ALTER_USER_DROP_ROLE
- DROP_DEFAULT_ROLE
- DROP_USER
- GRANT
- IMPLICIT_GRANT
- IMPLICIT_REVOKE
- REVOKE
- SET_SESSION_USER
- TRANSFER_OWNERSHIP
- UPDATE_DBM_CFG

Ereignisse für die Kategorie SYSADMIN

Tabelle 49. Ereignisse für die Kategorie SYSADMIN

START_DB2
STOP_DB2
CREATE_DATABASE
ALTER_DATABASE
DROP_DATABASE
UPDATE_DBM_CFG
UPDATE_DB_CFG
CREATE_TABLESPACE
DROP_TABLESPACE
ALTER_TABLESPACE
RENAME_TABLESPACE
CREATE_NODEGROUP
DROP_NODEGROUP
ALTER_NODEGROUP
CREATE_BUFFERPOOL
DROP_BUFFERPOOL
ALTER_BUFFERPOOL
CREATE_EVENT_MONITOR
DROP_EVENT_MONITOR
ENABLE_MULTIPAGE
MIGRATE_DB_DIR
DB2TRC
DB2SET
ACTIVATE_DB
ADD_NODE
BACKUP_DB
CATALOG_NODE
CATALOG_DB
CATALOG_DCS_DB
CHANGE_DB_COMMENT
DEACTIVATE_DB
DROP_NODE_VERIFY
FORCE_APPLICATION
GET_SNAPSHOT
LIST_DRDA_INDOUBT_TRANSACTIONS
MIGRATE_DB
RESET_ADMIN_CFG
RESET_DB_CFG
RESET_DBM_CFG
RESET_MONITOR
RESTORE_DB
ROLLFORWARD_DB
SET_RUNTIME_DEGREE
SET_TABLESPACE_CONTAINERS
UNCATALOG_DB
UNCATALOG_DCS_DB
UNCATALOG_NODE
UPDATE_ADMIN_CFG
UPDATE_MON_SWITCHES
LOAD_TABLE
DB2AUDIT
SET_APPL_PRIORITY
CREATE_DB_AT_NODE
KILLDBM
MIGRATE_SYSTEM_DIRECTORY
DB2REMOT
DB2AUD
MERGE_DBM_CONFIG_FILE
UPDATE_CLI_CONFIGURATION
OPEN_TABLESPACE_QUERY
SINGLE_TABLESPACE_QUERY
CLOSE_TABLESPACE_QUERY
FETCH_TABLESPACE
OPEN_CONTAINER_QUERY
FETCH_CONTAINER_QUERY
CLOSE_CONTAINER_QUERY
GET_TABLESPACE_STATISTICS
DESCRIBE_DATABASE
ESTIMATE_SNAPSHOT_SIZE
READ_ASYNC_LOG_RECORD
PRUNE_RECOVERY_HISTORY
UPDATE_RECOVERY_HISTORY
QUIESCE_TABLESPACE
UNLOAD_TABLE
UPDATE_DATABASE_VERSION
CREATE_INSTANCE
DELETE_INSTANCE
SET_EVENT_MONITOR
GRANT_DBADM
REVOKE_DBADM
GRANT_DB_AUTHORITIES
REVOKE_DB_AUTHORITIES
REDISTRIBUTE_NODEGROUP

Ereignisse für die Kategorie VALIDATE

- AUTHENTICATE
- CHECK_GROUP_MEMBERSHIP (ab Version 9.5 nicht mehr generiert)
- GET_USERMAPPING_FROM_PLUGIN
- GET_GROUPS (ab Version 9.5 nicht mehr generiert)
- GET_USERID (ab Version 9.5 nicht mehr generiert)

Kapitel 10. Arbeiten mit der Betriebssystemsicherheit

Betriebssysteme stellen Sicherheitseinrichtungen bereit, die Sie zur Unterstützung der Sicherheit für Ihre Datenbankinstallation verwenden können.

DB2- und Windows-Sicherheit

Bei einer Windows-Domäne handelt es sich um eine Zusammenstellung von Client- und Servercomputern, auf die mit einem bestimmten, eindeutigen Namen verwiesen wird und die eine Benutzerkontendatenbank gemeinsam nutzen, die als Security Access Manager (SAM) bezeichnet wird. Einer der Computer in der Domäne ist der Domänencontroller. Der Domänencontroller verwaltet alle Aspekte der Interaktionen zwischen den Benutzern und der Domäne.

Mithilfe der Informationen in der Benutzerkontendatenbank der Domäne authentifiziert der Domänencontroller Benutzer, die sich über Domänenkonten anmelden. Für jede Domäne ist ein Domänencontroller der primäre Domänencontroller (PDC - Primary Domain Controller). Innerhalb der Domäne kann es außerdem Backup-Domänen-Controller (BDC) geben, welche die Authentifizierung von Benutzerkonten durchführen, wenn kein primärer Domänencontroller vorhanden oder der primäre Domänencontroller nicht verfügbar ist. Backup-Domänen-Controller enthalten eine Kopie der Windows Security Account Manager-Datenbank (SAM), die regelmäßig mit der Masterkopie auf dem primären Domänencontroller synchronisiert wird.

Benutzerkonten, Benutzer-IDs und Kennwörter müssen nur auf dem primären Domänencontroller definiert werden, um auf die Ressourcen der Domäne zugreifen zu können.

Anmerkung: Zweiteilige Benutzer-IDs werden von der Anweisung CONNECT und dem Befehl ATTACH unterstützt. Das Qualifikationsmerkmal der SAM-kompatiblen Benutzer-ID ist ein Name der Form 'Domäne\Benutzer', der eine Länge von maximal 15 Zeichen besitzen kann.

Beim Setup wird bei Installation eines Windows-Servers das Erstellen folgender Einheiten zur Auswahl gestellt:

- Primärer Domänencontroller in einer neuen Domäne
- Backup-Domänen-Controller in einer bekannten Domäne
- Alleinstehender Server in einer bekannten Domäne

Durch die Auswahl „Controller" in einer neuen Domäne wird dieser Server als primärer Domänencontroller eingerichtet.

Der Benutzer kann sich an der lokalen Maschine oder, wenn die betreffende Maschine in einer Windows-Domäne installiert ist, bei der Domäne anmelden. Zur Authentifizierung des Benutzers überprüft DB2 zunächst die lokale Maschine, dann den Domänencontroller der aktuellen Domäne und zuletzt sonstige vertrauenswürdige Domänen, die dem Domänencontroller bekannt sind.

Zur Veranschaulichung dazu ein Beispiel: Angenommen, die DB2-Instanz erfordert eine Serverauthentifizierung. Die Konfiguration sieht folgendermaßen aus:

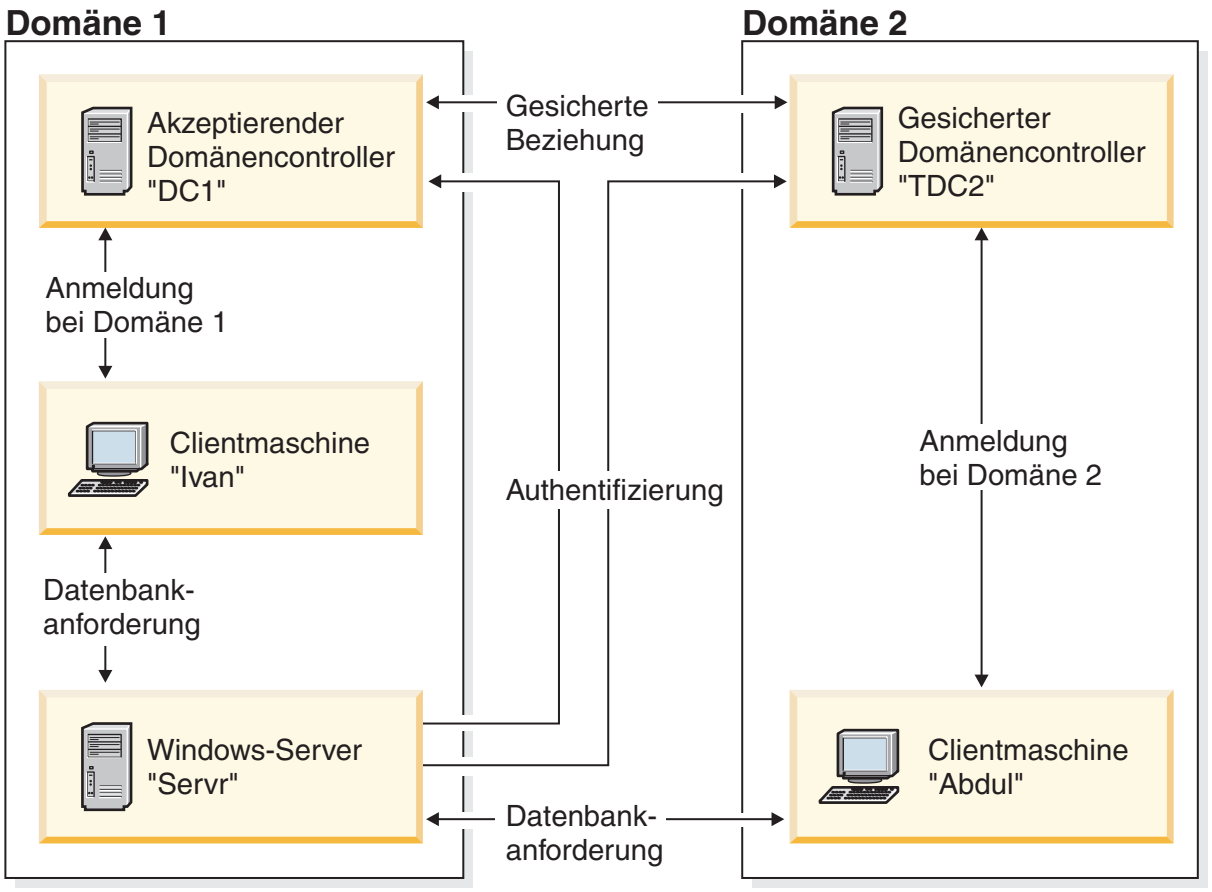


Abbildung 5. Authentifizierung in Windows-Domänen

Jede Maschine verfügt über eine SAM-Sicherheitsdatenbank (Security Access Management). 'DC1' ist der Domänencontroller, auf dem die Clientmaschine 'Ivan' und der DB2-Server 'Servr' registriert sind. 'TDC2' ist für 'DC1' eine sichere Domäne, und die Clientmaschine 'Abdul' ist ein Mitglied der Domäne von 'TDC2'.

Authentifizierungsszenarios

Szenario mit Serverauthentifizierung (Windows)

1. Adbul meldet sich an der Domäne TDC2 an (d. h. dieser Client ist in der SAM-Datenbank von TDC2 registriert).
2. Abdul stellt dann die Verbindung zu einer DB2-Datenbank her, die laut Katalog auf SRV3 gespeichert ist:

   ```
   db2 connect to remotedb user Abdul using fredpw
   ```

3. SRV3 stellt fest, wo Abdul bekannt ist. Die API, die zum Abrufen dieser Informationen eingesetzt wird, durchsucht zunächst die lokale Maschine (SRV3) und anschließend den Domänencontroller (DC1), bevor sie mit anderen gesicherten Domänen fortfährt. Der Benutzername Abdul wird auf TDC2 gefunden. Diese Suchreihenfolge setzt einen einzigen Namensbereich (Namespace) für Benutzer und Gruppen voraus.
4. SRV3 unternimmt dann folgende Schritte:
   a. Er überprüft den Benutzernamen und das Kennwort mithilfe von TDC2.
   b. Er ermittelt, ob Abdul ein Administrator ist, indem er TDC2 befragt.
   c. Er zählt alle Gruppen von Abdul auf, indem er TDC2 befragt.

Szenario mit Clientauthentifizierung und einer Windows-Clientmaschine

1. Dale, der Administrator, meldet sich an SRV3 an und ändert den Authentifizierungstyp für die Datenbankinstanz in CLIENT:

   ```
   db2 update dbm cfg using authentication client
   db2stop
   db2start
   ```

2. Ivan meldet sich von der Windows-Clientmaschine aus an der Domäne DC1 an (d. h., er ist in der SAM-Datenbank von DC1 registriert).
3. Ivan stellt dann die Verbindung zu einer DB2-Datenbank her, die laut Katalog auf SRV3 gespeichert ist:

   ```
   DB2 CONNECT to remotedb user Ivan using johnpw
   ```

4. Ivans Maschine überprüft den Benutzernamen und das Kennwort. Die API, die zum Abrufen dieser Informationen eingesetzt wird, durchsucht zunächst die lokale Maschine (Ivan) und anschließend die Domänensteuereinheit (DC1), bevor sie mit anderen gesicherten Domänen fortfährt. Der Benutzername Ivan wird auf DC1 gefunden.
5. Ivans Maschine überprüft dann den Benutzernamen und das Kennwort mithilfe von DC1.
6. SRV3 unternimmt dann folgende Schritte:
   a. Er stellt fest, wo Ivan registriert ist.
   b. Er ermittelt, ob Ivan ein Administrator ist, indem er DC1 befragt.
   c. Er zählt alle Gruppen von Ivan auf, indem er DC1 befragt.

Anmerkung: Stellen Sie sicher, dass der DB2-Sicherheitsservice gestartet wurde, bevor Sie versuchen, eine Verbindung zur DB2-Datenbank herzustellen. Der Sicherheitsservice wird im Rahmen der Windows-Installation installiert. DB2 wird anschließend installiert und als Windows-Dienst (Service) „registriert", jedoch nicht automatisch gestartet. Geben Sie den Befehl `NET START DB2NTSECSERVER` ein, um den DB2-Sicherheitsservice zu starten.

Unterstützung globaler Gruppen (unter Windows)

Das DB2-Datenbanksystem unterstützt globale Gruppen.

Zur Nutzung globaler Gruppen müssen Sie die gewünschten globalen Gruppen als Mitglieder in eine lokale Gruppe einfügen. Wenn der DB2-Datenbankmanager alle Gruppen aufzählt, deren Mitglied eine Person ist, führt er auch die lokalen Gruppen auf, deren Mitglied der Benutzer indirekt ist (dadurch, dass er in einer globalen Gruppe ist, die selbst wiederum zu einer oder mehreren lokalen Gruppen gehört).

Globale Gruppen können auf zwei Arten verwendet werden:

- Als Mitglieder in einer lokalen Gruppe. Berechtigungen müssen dieser lokalen Gruppe erteilt werden.
- Als eigenständige Gruppe auf einem Domänencontroller. Berechtigungen müssen der globalen Gruppe erteilt werden.

Benutzerauthentifizierung mit DB2 unter Windows

Einschränkungen für Benutzer- und Gruppennamen (Windows)

Es gibt einige wenige Einschränkungen, die speziell für die Windows-Umgebung zu berücksichtigen sind. Beachten Sie, dass die allgemeinen Namensregeln für DB2-Objekte ebenfalls gelten.

- Bei unter Windows verwendeten Benutzernamen muss die Groß-/Kleinschreibung nicht beachtet werden, jedoch bei den zugehörigen Kennwörtern.
- Benutzernamen und Gruppennamen können aus einer Kombination von Groß- und Kleinbuchstaben eingegeben werden. Allerdings werden sie für die Verwendung innerhalb des DB2-Datenbanksystems normalerweise in Großbuchstaben umgesetzt. Wenn Sie z. B. eine Verbindung zu einer Datenbank herstellen und die Tabelle schema1.tabelle1 erstellen, wird diese Tabelle unter dem Namen SCHEMA1.TABELLE1 in der Datenbank gespeichert. (Wenn Sie Objektnamen in Kleinbuchstaben verwenden wollen, müssen Sie über den Befehlszeilenprozessor entsprechende Befehle absetzen. Dabei müssen die Objektnamen in Anführungszeichen eingeschlossen werden. Alternativ können Sie auch ODBC-Front-End-Tools anderer Hersteller verwenden.)
- Ein Benutzer kann nicht mehr als 64 Gruppen angehören.
- Der DB2-Datenbankmanager unterstützt nur einen Namensbereich. Das heißt für eine Umgebung mit gesicherten (vertrauten) Domänen, dass Sie kein Benutzerkonto eines Namens haben sollten, der in mehreren Domänen vorhanden ist oder der in der lokalen SAM-Datenbank der Servermaschine und in einer anderen Domäne vorhanden ist.

Gruppen- und Benutzerauthentifizierung unter Windows

Benutzer werden unter Windows definiert, indem Benutzerkonten über ein Windows-Verwaltungstool mit dem Namen „Benutzer-Manager" erstellt werden. Eine Gruppe ist ein Konto, das andere Konten enthält, die in diesem Fall als Mitglieder bezeichnet werden.

Gruppen bieten Windows-Administratoren die Möglichkeit, Benutzern einer Gruppe gleichzeitig Zugriffsrechte und Berechtigungen zu erteilen, ohne jeden Benutzer einzeln verwalten zu müssen. Gruppen werden ebenso wie Benutzerkonten in der SAM-Datenbank (SAM - Security Access Manager) definiert und gepflegt.

Es gibt zwei Arten von Gruppen:

- Lokale Gruppen. Eine lokale Gruppe kann Benutzerkonten enthalten, die in der lokalen Kontendatenbank erstellt wurden. Wenn sich die lokale Gruppe auf einer Maschine befindet, die zu einer Domäne gehört, kann die lokale Gruppe außerdem Domänenkonten und Gruppen aus der Windows-Domäne enthalten. Wenn die lokale Gruppe auf einer Workstation erstellt wird, ist sie für diese Workstation spezifisch.
- Globale Gruppen. Eine globale Gruppe ist nur auf einem Domänencontroller vorhanden und enthält Benutzerkonten der SAM-Datenbank der Domäne. Das heißt, eine globale Gruppe kann nur Benutzerkonten aus der Domäne enthalten, in der sie erstellt wurde. Sie kann keine anderen Gruppen als Mitglieder enthalten. Eine globale Gruppe kann in Servern und Workstations der eigenen Domäne und auch in akzeptierenden (vertrauenden) Domänen verwendet werden.

Vertrauensstellungen zwischen Domänen unter Windows

Vertrauensstellungen stellen eine Verwaltungs- und Kommunikationsverbindung zwischen zwei Domänen dar. Eine Vertrauensstellung zwischen zwei Domänen stellt die Möglichkeit bereit, Benutzerkonten und globale Gruppen in einer anderen Domäne zu verwenden als der, in der die Konten definiert sind.

Konteninformationen werden gemeinsam verwendet, um die Gültigkeit von Zugriffsrechten und Berechtigungen von Benutzerkonten und globalen Gruppen ohne erneute Authentifizierung zu bestätigen. Vertrauensstellungen vereinfachen die Benutzerverwaltung, indem sie zwei oder mehr Domänen zu einer administrativen Einheit verbinden.

In einer Vertrauensstellung befinden sich zwei Domänen:

- Die vertrauende Domäne. Diese Domäne vertraut der anderen Domäne in Bezug auf die Authentifizierung von Benutzern für beide Domänen.
- Die vertraute (gesicherte) Domäne. Diese Domäne authentifiziert Benutzer im Auftrag (im Vertrauen) einer anderen Domäne.

Vertrauensstellungen sind nicht transitiv. Dies bedeutet, dass in beide Richtungen zwischen den Domänen explizite Vertrauensstellungen eingerichtet werden müssen. Zum Beispiel muss die vertrauende Domäne nicht unbedingt auch eine vertraute Domäne sein.

DB2-Datenbanksystem und Sicherheitsservice (Windows-Dienst)

Für das DB2-Datenbanksystem wurde die Authentifizierung von Benutzernamen und Kennwörtern in den DB2-Systemcontroller integriert.

Der Sicherheitsservice wird nur für die Verbindung eines Clients zu einem Server benötigt, für den der Authentifizierungstyp CLIENT konfiguriert wurde.

Authentifizierung mit Gruppen und Domänensicherheit (Windows)

Das DB2-Datenbanksystem gibt Ihnen die Möglichkeit, beim Erteilen von Zugriffsrechten oder Definieren von Berechtigungsstufen eine lokale Gruppe oder eine globale Gruppe anzugeben.

Ein Benutzer wird als Mitglied einer Gruppe erkannt, wenn das Konto des Benutzers explizit in der lokalen bzw. globalen Gruppe oder implizit durch seine Mitgliedschaft in einer globalen Gruppe definiert ist, die wiederum als Mitglied einer lokalen Gruppe definiert ist.

Der DB2-Datenbankmanager unterstützt die folgenden Gruppenarten:

- Lokale Gruppen
- Globale Gruppen
- Globale Gruppen als Mitglieder einer lokalen Gruppe

  Der DB2-Datenbankmanager zählt die lokalen und globalen Gruppen auf, zu denen der Benutzer gehört, und verwendet dazu die Sicherheitsdatenbank, in der der Benutzer lokalisiert wurde. Das DB2-Datenbanksystem stellt eine Überschreibungsfunktion zur Verfügung, durch die die Ausführung der Gruppenaufzählung auf dem lokalen Windows-Server erzwungen wird, auf dem die DB2-Datenbank installiert ist, unabhängig davon, wo das Benutzerkonto lokalisiert wurde. Diese Überschreibung kann mit den folgenden Befehlen veranlasst werden:

  – Für globale Einstellungen:

```
db2set -g DB2_GRP_LOOKUP=local
```

– Für Instanzeinstellungen:

```
db2set -i <instanzname> DB2_GRP_LOOKUP=local
```

Nach der Ausführung dieses Befehls müssen Sie die DB2-Datenbankinstanz stoppen und erneut starten, um die Änderung in Kraft zu setzen. Erstellen Sie anschließend lokale Gruppen und fügen Sie Domänenkonten oder globale Gruppen in die lokale Gruppe als Mitglieder ein.

Geben Sie Folgendes ein, um alle definierten Variablen der DB2-Profilregistrierdatenbank anzuzeigen:

```
db2set -all
```

Wenn die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank auf den Wert 'local' gesetzt ist, versucht die DB2-Datenbank, die Gruppen des Benutzers nur der lokalen Maschine aufzuzählen. Wenn der Benutzer nicht als Mitglied einer lokalen oder globalen Gruppe definiert ist, schlägt die Aufzählung der Gruppen fehl. DB2 versucht **nicht**, die Gruppen des Benutzers auf einer anderen Maschine in der Domäne oder auf dem Domänencontroller aufzuzählen.

Wenn die Variable DB2_GRP_LOOKUP der Profilregistrierdatenbank nicht definiert ist, gilt Folgendes:

1. Das DB2-Datenbanksystem versucht zunächst, den Benutzer auf derselben Maschine zu finden.
2. Ist der Benutzername lokal definiert, wird auch die Authentifizierung lokal ausgeführt.
3. Wird der Benutzer auf dem lokalen System nicht gefunden, versucht das DB2-Datenbanksystem, den Benutzernamen in der zugehörigen Domäne und anschließend in den sicheren Domänen zu lokalisieren.

Wenn der DB2-Datenbankmanager auf einer Maschine ausgeführt wird, die als primärer Domänencontroller (PDC) oder Backup-Domänen-Controller (BDC) in der Ressourcendomäne eingesetzt ist, kann er jeden Domänencontroller in jeder beliebigen sicheren Domäne lokalisieren. Der Grund dafür ist der, dass die Namen der Domänen von Backup-Domänen-Controllern in gesicherten Domänen einer Maschine nur bekannt sind, wenn sie ein Domänencontroller ist.

Wenn der DB2-Datenbankmanager nicht auf einem Domänencontroller ausgeführt wird, geben Sie folgenden Befehl ein:

```
db2set -g DB2_GRP_LOOKUP=DOMAIN
```

Dieser Befehl weist das DB2-Datenbanksystem an, einen Domänencontroller in der eigenen Domäne zu verwenden, um den Namen eines Domänencontrollers in der Kontendomäne ausfindig zu machen. Das heißt, wenn ein DB2-Datenbanksystem ermittelt, dass ein bestimmtes Benutzerkonto in Domäne x definiert ist, versucht es nicht, einen Domänencontroller für Domäne x zu finden, sondern sendet die entsprechende Anforderung an einen Domänencontroller in der eigenen Domäne. Der Name des Domänencontrollers in der Kontendomäne wird gefunden und an die Maschine zurückgegeben, auf der die DB2-Datenbank ausgeführt wird. Diese Methode bietet zwei Vorteile:

1. Es wird der nächste Domänencontroller gefunden, wenn der primäre Domänencontroller nicht verfügbar ist.
2. Es wird der nächste Domänencontroller gefunden, wenn sich der primäre Domänencontroller an einem geographisch entfernen Standort befindet.

Authentifizierung mit einer geordneten Domänenliste

Benutzer-IDs können mehr als einmal in einer gesicherten (vertrauten) Domänengesamtstruktur definiert sein. Eine gesicherte Domänengesamtstruktur ist eine Gruppe von Domänen, die durch ein Netzwerk miteinander verbunden sind.

Es ist möglich, dass ein Benutzer in einer Domäne die gleiche Benutzer-ID wie ein anderer Benutzer in einer anderen Domäne besitzt. Dies kann beim Versuch einer der folgenden Aktionen zu Schwierigkeiten führen:

- Authentifizierung mehrerer Benutzer, die die gleiche Benutzer-ID, jedoch in verschiedenen Domänen haben
- Gruppensuche zum Zweck des Erteilens und Widerrufens von Zugriffsrechten für Gruppen
- Überprüfung von Kennwörtern
- Steuerung des Datenaustauschs im Netzwerk

Zur Vermeidung von Schwierigkeiten aufgrund der möglichen Existenz mehrerer Benutzer mit der gleichen Benutzer-ID in einer Domänengesamtstruktur sollten Sie eine geordnete Domänenliste verwenden, wie sie in der Registrierdatenbankvariablen DB2DOMAINLIST mithilfe des Befehls db2set definiert werden kann. Bei der Definition der Reihenfolge müssen die Domänen, die in die Liste aufgenommen werden sollen, durch Kommas voneinander getrennt werden. Sie müssen eine bewusste Entscheidung im Hinblick auf die Reihenfolge treffen, in der die Domänen bei der Authentifizierung von Benutzern durchsucht werden.

Die Benutzer-IDs, die sich in Domänen weiter hinten in der Liste befinden, müssen von Ihnen umbenannt werden, wenn Sie für den Zugriff authentifiziert werden sollen.

Eine Steuerung des Zugriffs kann über die Domänenliste erfolgen. Wenn zum Beispiel die Domäne eines Benutzers nicht in der Liste enthalten ist, erhält der Benutzer keine Berechtigung zur Herstellung einer Verbindung.

Anmerkung: Die Registrierdatenbankvariable DB2DOMAINLIST ist nur wirksam, wenn in der Konfiguration des Datenbankmanagers eine CLIENT-Authentifizierung definiert ist, und wird benötigt, wenn eine einmalige Anmeldung (single signon on) über einen Windows-Desktop in einer Windows-Domänenumgebung erforderlich ist. Die Registrierdatenbankvariable DB2DOMAINLIST wird von einigen Versionen von DB2-Servern unterstützt. Allerdings wird die Registrierdatenbankvariable DB2DOMAINLIST nicht umgesetzt, wenn sich weder der Client noch der Server in einer Windows-Umgebung befinden.

Unterstützung der Domänensicherheit (Windows)

Das folgende Beispiel erläutert, wie das DB2-Datenbankmanagementsystem die Windows-Domänensicherheit unterstützten kann. Die Verbindung funktioniert, weil sich der Benutzername und die lokale Gruppe in derselben Domäne befinden.

Die Verbindung funktioniert im folgenden Szenario, weil sich der Benutzername und die lokale oder globale Gruppe in derselben Domäne befinden.

Beachten Sie hierbei, dass der Benutzername und die lokale oder globale Gruppe nicht in der Domäne definiert sein müssen, in der der Datenbankserver ausgeführt wird. Sie müssen sich jedoch in derselben Domäne befinden.

Tabelle 50. Erfolgreiche Verbindung mit einem Domänencontroller

Domäne1	**Domäne2**
Zu Domäne2 besteht eine Vertrauensstellung.	• Zu Domäne1 besteht eine Vertrauensstellung. • Die lokale oder globale Gruppe `grp2` wurde definiert. • Der Benutzername `id2` wurde definiert. • Der Benutzername `id2` gehört zu `grp2`.

Der DB2-Server wird in dieser Domäne ausgeführt. Die folgenden DB2-Befehle werden über dieses System abgesetzt:
`REVOKE CONNECT ON db FROM public`
`GRANT CONNECT ON db TO GROUP grp2`
`CONNECT TO db USER id2`	

Die lokale oder globale Domäne wird durchsucht, `id2` kann jedoch nicht gefunden werden. Die Einrichtung für die Domänensicherheit wird durchsucht.	
	Der Benutzername `id2` wird in dieser Domäne gefunden. DB2 ruft zusätzliche Informationen zu diesem Benutzernamen (der zu der Gruppe `grp2` gehört) ab.
Die Verbindung funktioniert, weil sich der Benutzername und die lokale oder globale Gruppe in derselben Domäne befinden.	

Abrufen von Windows-Benutzergruppeninformationen mit einem Zugriffstoken

Ein Zugriffstoken ist ein Objekt, dass den Sicherheitskontext eines Prozesses oder Threads beschreibt. Die Informationen in einem Zugriffstoken enthalten die Identität und die Zugriffsrechte des Benutzerkontos, das dem Prozess oder Thread zugeordnet ist.

Wenn Sie sich anmelden, überprüft das System Ihr Kennwort, indem es dieses mit den in einer Sicherheitsdatenbank gespeicherten Informationen vergleicht. Wenn das Kennwort authentifiziert wird, erstellt das System ein Zugriffstoken. Jeder Prozess, der in Ihrem Namen ausgeführt wird, verwendet eine Kopie dieses Zugriffstokens.

Ein Zugriffstoken kann außerdem auf der Grundlage eines im Cache gespeicherten Berechtigungsnachweises empfangen werden. Wenn Sie für das System authentifiziert wurden, werden Ihre Berechtigungsnachweise vom Betriebssystem im Cache gespeichert. Auf das Zugriffstoken der letzten Anmeldung kann im Cache zurückgegriffen werden, wenn es nicht möglich ist, den Domänencontroller zu kontaktieren.

Das Zugriffstoken enthält Informationen über alle Gruppen, denen Sie angehören: lokale Gruppen und verschiedene Domänengruppen (globale Gruppen, lokale Domänengruppen, universelle Gruppen).

Anmerkung: Eine Gruppensuche (Lookup) unter Verwendung der Clientauthentifizierung wird über eine Remoteverbindung nicht unterstützt, selbst wenn die Unterstützung für Zugriffstoken aktiviert ist.

Zur Aktivierung der Unterstützung für Zugriffstoken müssen Sie die Registrierdatenbankvariable DB2_GRP_LOOKUP mithilfe des Befehls db2set aktualisieren. Folgende Möglichkeiten stehen zur Aktualisierung dieser Registrierdatenbankvariablen zur Verfügung:

- TOKEN

  Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen Gruppen, zu denen der Benutzer gehört, und zwar sowohl auf lokalen Systemen als auch an der Position, an der das Benutzerkonto definiert ist (wenn das Konto auf der Domänenebene definiert ist).
- TOKENLOCAL

  Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen lokalen Gruppen, zu denen der Benutzer gehört, auf dem DB2-Datenbankserver.
- TOKENDOMAIN

  Diese Auswahl aktiviert die Unterstützung für Zugriffstoken zur Suche nach allen Gruppen, zu denen der Benutzer gehört, an der Position, an der das Benutzerkonto definiert ist. Dabei handelt es sich in der Regel um ein Verzeichnis in der Domäne oder ein lokales Verzeichnis auf dem DB2-Datenbankserver.

Es empfiehlt sich im Allgemeinen, die Registrierdatenbankvariable DB2_GRP_LOOKUP zu verwenden und die Position für die Gruppensuchfunktion anzugeben, um dem DB2-Datenbanksystem anzugeben, wo nach Gruppen mit der herkömmlichen Gruppenaufzählungsmethode gesucht werden soll. Beispiel:

```
db2set DB2_GRP_LOOKUP=LOCAL,TOKENLOCAL
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung lokaler Gruppen.

```
db2set DB2_GRP_LOOKUP=,TOKEN
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung von Gruppen sowohl auf den lokalen Systemen als auch an der Position, an der das Benutzerkonto definiert ist (wenn das Konto auf der Domänenebene definiert ist).

```
db2set DB2_GRP_LOOKUP=DOMAIN,TOKENDOMAIN
```

Dieser Befehl aktiviert die Unterstützung für Zugriffstoken zur Aufzählung von Domänengruppen an der Position, an der die Benutzer-ID definiert ist.

Die Unterstützung für Zugriffstoken kann bei allen Authentifizierungstypen außer der Clientauthentifizierung (CLIENT) aktiviert werden.

Windows-Plattform: Sicherheitsaspekte für Benutzer

Die Berechtigung SYSADM für die Systemverwaltung wird allen gültigen DB2-Datenbankbenutzerkonten erteilt, die auf der Maschine, auf der sie definiert sind, zur lokalen Gruppe der Administratoren gehören.

Standardmäßig verfügen in einer Windows-Domänenumgebung nur Domänenbenutzer, die zur Administratorengruppe des Domänencontrollers gehören, über die Berechtigung SYSADM für eine Instanz. Da DB2 die Erteilung von Berechtigungen immer auf der Maschine ausführt, auf der das jeweilige Konto definiert ist, wird einem Domänenbenutzer durch das Hinzufügen zur lokalen Administratorgruppe auf dem Server nicht die Berechtigung SYSADM für die Gruppe erteilt.

Anmerkung: In einer Domänenumgebung wie unter Windows authentifiziert DB2 nur die ersten 64 Gruppen, die den Anforderungen und Einschränkungen entsprechen und eine Benutzer-ID besitzen. Sie können jedoch über mehr als 64 Gruppen verfügen.

Um das Hinzufügen eines Domänenbenutzers zur Administratorgruppe auf dem primären Domänencontroller (PDC) zu vermeiden, sollten Sie eine globale Gruppe erstellen und dieser sowohl die Domänenbenutzer als auch die lokalen Benutzer hinzufügen, denen die Berechtigung SYSADM erteilt werden soll. Geben Sie dazu folgende Befehle ein:

```
DB2STOP
DB2 UPDATE DBM CFG USING SYSADM_GROUP globale_gruppe
DB2START
```

Unterstützung für das lokale Systemkonto unter Windows

Auf Windows-Plattformen (außer Windows ME) unterstützt das DB2-Datenbanksystem Anwendungen, die unter dem Kontext des lokalen Systemkontos (LSA) mit impliziter lokaler Verbindung ausgeführt werden.

Entwickler, die Anwendungen zur Ausführung unter diesem Konto schreiben, müssen sich über die Einschränkungen im DB2-Datenbanksystem für Objekte mit Schemanamen, die mit „SYS" beginnen, im Klaren sein. Wenn Ihre Anwendungen DDL-Anweisungen enthalten, durch die DB2-Datenbankobjekte erstellt werden, sollten diese daher so geschrieben sein, dass sie folgende Voraussetzungen erfüllen:

- Für statische Abfragen sollten sie mit einem anderen Wert für die QUALIFIER-Optionen als dem Standardwert gebunden werden.
- Für dynamische Abfragen sollten die zu erstellenden Objekte explizit mit einem vom DB2-Datenbankmanager unterstützten Schemanamen qualifiziert werden, oder das Register CURRENT SCHEMA muss auf einen Schemanamen gesetzt werden, der vom DB2-Datenbankmanager unterstützt wird.

Gruppeninformationen für das lokale Systemkonto (LSA) werden beim ersten Suchen nach Gruppen (Lookup) nach dem Start der DB2-Datenbankinstanz erfasst und erst aktualisiert, wenn die Instanz erneut gestartet wird.

Anmerkung: Anwendungen, die unter dem Kontext des lokalen Systemkontos (LSA) ausgeführt werden, werden auf allen Windows-Plattformen mit Ausnahme von Windows ME unterstützt.

Erweiterte Windows-Sicherheit mithilfe der Gruppen DB2ADMNS und DB2USERS

Bei der Serverversion des DB2-Datenbankmanagers ist die erweiterte Sicherheit (DB2 Extended Security) standardmäßig implizit *aktiviert*. Bei der Clientversion ist die erweiterte Sicherheit hingegen standardmäßig *inaktiviert*. Sie *müssen* die erweiterte Sicherheit bei der Installation daher explizit auswählen, damit sie aktiviert wird.

Zur Aktivierung der erweiterten Sicherheit wählen Sie bei der Installation von DB2 auf einem Client das Markierungsfeld **Betriebssystemsicherheit aktivieren** im Fenster **Betriebssystemsicherheit für DB2-Objekte aktivieren** aus. Das Installationsprogramm erstellt die beiden neuen Gruppen DB2ADMNS und DB2USERS. Die Namen DB2ADMNS und DB2USERS sind die Standardnamen der Gruppen. Bei der Installation können Sie optional andere Namen für diese Gruppen angeben (wenn Sie eine nicht überwachte Installation auswählen, können Sie diese Namen

in der Antwortdatei für die Installation ändern). Wenn Sie sich entscheiden, Gruppen zu verwenden, die in Ihrem System bereits vorhanden sind, müssen Sie beachten, dass die Zugriffsrechte für diese Gruppen modifiziert werden. Ihnen werden nach Bedarf die Zugriffsrechte erteilt, die in der Tabelle weiter unten aufgeführt sind. Es wichtig zu beachten, dass diese Gruppen für den Schutz auf der *Betriebssystemebene* verwendet werden und in keiner Weise den DB2-Berechtigungsstufen (z. B. SYSADM, SYSMAINT und SYSCTRL) zugeordnet sind. Allerdings kann Ihr Datenbankadministrator anstelle der Standardgruppe für Administratoren die Gruppe DB2ADMNS für eine oder auch alle DB2-Berechtigungsstufen verwenden. Dies liegt im Ermessen des Installationsverantwortlichen oder des Administrators. Wenn Sie eine SYSADM-Gruppe angeben, sollte dies die Gruppe DB2ADMNS sein. Dies kann während der Installation oder nachfolgend durch einen Administrator eingerichtet werden.

Anmerkung: Sie können Ihre DB2-Administratorengruppe (DB2ADMNS oder den Namen, den Sie bei der Installation auswählen) und Ihre DB2-Benutzergruppe (DB2USERS oder den Namen, den Sie bei der Installation auswählen) entweder als lokale Gruppen oder Domänengruppen angeben. Beide Gruppen müssen vom selben Typ sein, das heißt, beide müssen entweder lokale Gruppen oder Domänengruppen sein.

Wenn Sie den Computernamen ändern und die Computergruppen DB2ADMNS und DB2USERS lokale Computergruppen sind, müssen Sie die globalen Registrierdatenbanken DB2_ADMINGROUP und DB2_USERSGROUP aktualisieren. Zum Aktualisieren der Registrierdatenbankvariablen nach dem Umbenennen und erneuten Starten des Computers führen Sie den folgenden Befehl aus:

1. Öffnen Sie eine Eingabeaufforderung.
2. Führen Sie den Befehl db2extsec zum Aktualisieren der Sicherheitseinstellungen:

   ```
   db2extsec -a neuer computername\DB2ADMNS -u neuer computername\DB2USERS
   ```

Anmerkung: Wenn die erweiterte Sicherheit für DB2-Datenbankprodukte unter Windows Vista aktiviert wird, können nur Benutzer, die Mitglied der Gruppe DB2ADMNS sind, die grafischen DB2-Verwaltungstools ausführen. Darüber hinaus müssen die Mitglieder der Gruppe DB2ADMNS die Tools mit den vollständigen Administratorrechten starten. Hierzu müssen Sie mit der rechten Maustaste auf den Direktaufruf klicken und dann die Option zur Ausführung als Administrator auswählen.

Durch die Gruppen DB2ADMNS und DB2USERS erteilte Berechtigungen

Die Gruppen DB2ADMNS und DB2USERS statten Mitglieder mit den folgenden Berechtigungen aus:

- DB2ADMNS

  Vollständige Steuerung über alle DB2-Objekte (siehe Liste der geschützten Objekte unten)
- DB2USERS

  Lese- und Ausführungszugriff auf alle DB2-Objekte, die sich im Installationsverzeichnis und im Instanzverzeichnis befinden, jedoch keinen Zugriff auf Objekte unter dem Datenbanksystemverzeichnis sowie begrenzten Zugriff auf IPC-Ressourcen

Für bestimmte Objekte können je nach Bedarf zusätzliche Zugriffsrechte verfügbar sein (z. B. Zugriffsrechte zum Schreiben, Hinzufügen und Aktualisieren usw.). Mitglieder dieser Gruppe haben keinen Zugriff auf Objekte unter dem Datenbanksystemverzeichnis.

Anmerkung: Die Bedeutung des Ausführungszugriffs hängt von der Art des Objekts ab. Bei Dateien mit den Erweiterungen **.dll** oder **.exe** bedeutet Ausführungszugriff, dass Sie berechtigt sind, die jeweilige Datei auszuführen. Bei einem Verzeichnis bedeutet dieser Zugriff, dass Sie berechtigt sind, das Verzeichnis zu durchqueren.

Im Idealfall sollten alle DB2-Administratoren Mitglieder der Gruppe DB2ADMNS (sowie Mitglieder der lokalen Administratorengruppe) sein, jedoch ist dies keine strikte Voraussetzung. Alle anderen Personen, die Zugriff auf das DB2-Datenbanksystem benötigen, *müssen* Mitglied der Gruppe DB2USERS sein. Gehen Sie wie folgt vor, um einer dieser Gruppen einen Benutzer hinzuzufügen:

1. Starten Sie das Verwaltungstool für Benutzer und Kennwörter.
2. Wählen Sie den hinzuzufügenden Benutzernamen in der Liste aus.
3. Klicken Sie **Eigenschaften** an. Klicken Sie im Eigenschaftenfenster die Indexzunge für Gruppenmitgliedschaft an.
4. Wählen Sie den Radioknopf **Andere** aus.
5. Wählen Sie die gewünschte Gruppe in der Dropdown-Liste aus.

Hinzufügen der erweiterten Sicherheit nach der Installation (Befehl 'db2extsec')

Wenn das DB2-Datenbanksystem ohne Aktivierung der erweiterten Sicherheit (Extended Security) installiert wurde, können Sie diese aktivieren, indem Sie den Befehl **db2extsec** (in früheren Releases **db2secv82**) ausführen. Zur Ausführung des Befehls **db2extsec** müssen Sie Mitglied der lokalen Administratorgruppe sein, sodass Sie berechtigt sind, die Zugriffssteuerungsliste (ACL) der geschützten Objekte zu modifizieren.

Sie können den Befehl **db2extsec** mehrfach ausführen, falls erforderlich. Wenn Sie dies jedoch tun, können Sie die erweiterte Sicherheit nur dann wieder inaktivieren, wenn Sie den Befehl **db2extsec –r** direkt im Anschluss an *jede* Ausführung des Befehls **db2extsec** ausführen.

Entfernen der erweiterten Sicherheit

Achtung:
Entfernen Sie die erweiterte Sicherheit nicht, nachdem sie aktiviert wurde, sofern dies nicht absolut notwendig ist.

Sie können die erweiterte Sicherheit entfernen, indem Sie den Befehl **db2extsec -r** ausführen. Dies ist jedoch nur dann erfolgreich, wenn keine anderen Datenbankoperationen (z. B. Erstellen einer Datenbank, Erstellen einer neuen Instanz, Hinzufügen von Tabellenbereichen usw.) nach der Aktivierung der erweiterten Sicherheit ausgeführt wurden. Die sicherste Methode zum Entfernen der Option für die erweiterte Sicherheit besteht darin, das DB2-Datenbanksystem zu deinstallieren, alle relevanten DB2-Verzeichnisse (einschließlich der Datenbankverzeichnisse) zu löschen und das DB2-Datenbanksystem anschließend erneut zu installieren, ohne die erweiterte Sicherheit zu aktivieren.

Geschützte Objekte

Die folgenden *statischen* Objekte können mithilfe der Gruppen DB2ADMNS und DB2USERS geschützt werden:

- Dateisystem
  - Datei
  - Verzeichnis
- Services (Dienste)
- Registrierungsschlüssel

Die folgenden *dynamischen* Objekte können mithilfe der Gruppen DB2ADMNS und DB2USERS geschützt werden:

- Ressourcen der Interprozesskommunikation (IPC-Ressourcen):
  - Pipes
  - Semaphore
  - Ereignisse
- Gemeinsam genutzter Speicher

Zugriffsrechte der Gruppen DB2ADMNS und DB2USERS

In der folgenden Tabelle sind die Zugriffsrichte aufgelistet, die den Gruppen DB2ADMNS und DB2USERS zugewiesen sind:

Tabelle 51. Zugriffsrechte für die Gruppen DB2ADMNS und DB2USERS

Zugriffsrecht	DB2ADMNS	DB2USERS	Erläuterung
Tokenobjekt erstellen (SeCreateTokenPrivilege)	J	N	Tokenbearbeitung (für bestimmte Operationen zur Tokenbearbeitung erforderlich und bei der Authentifizierung und Autorisierung verwendet)
Token auf Prozessebene ersetzen (SeAssignPrimaryTokenPrivilege)	J	N	Prozesserstellung als anderer Benutzer
Quoten anheben (SeIncreaseQuotaPrivilege)	J	N	Prozesserstellung als anderer Benutzer
Als Teil des Betriebssystems handeln (SeTcbPrivilege)	J	N	LogonUser (vor Windows XP zur Ausführung der API 'LogonUser' zu Authentifizierungszwecken erforderlich)
Sicherheitsprotokolle generieren (SeSecurityPrivilege)	J	N	Bearbeitung von Prüf- und Sicherheitsprotokollen
Eigentumsrecht an Dateien und anderen Objekten übernehmen (SeTakeOwnershipPrivilege)	J	N	Modifizieren von Zugriffssteuerlisten (ACLs) von Objekten
Ausführungspriorität anheben (SeIncreaseBasePriorityPrivilege)	J	N	Modifizieren des Prozessarbeitsbereichs
Backup von Dateien und Verzeichnissen durchführen (SeBackupPrivilege)	J	N	Bearbeitung von Profilen/ Registrierdatenbank (zur Ausführung bestimmter Bearbeitungsroutinen für Benutzerprofile und die Registrierung erforderlich: LoadUserProfile, RegSaveKey(Ex), RegRestoreKey, RegReplaceKey, RegLoadKey(Ex))

Tabelle 51. Zugriffsrechte für die Gruppen DB2ADMNS und DB2USERS (Forts.)

Zugriffsrecht	DB2ADMNS	DB2USERS	Erläuterung
Restore von Dateien und Verzeichnissen ausführen (SeRestorePrivilege)	J	N	Bearbeitung von Profilen/ Registrierdatenbank (zur Ausführung bestimmter Bearbeitungsroutinen für Benutzerprofile und die Registrierung erforderlich: LoadUserProfile, RegSaveKey(Ex), RegRestoreKey, RegReplaceKey, RegLoadKey(Ex))
Programme debuggen (SeDebugPrivilege)	J	N	Tokenbearbeitung (für bestimmte Operationen zur Tokenbearbeitung erforderlich und bei der Authentifizierung und Autorisierung verwendet)
Prüf- und Sicherheitsprotokoll verwalten (SeAuditPrivilege)	J	N	Generierung von Prüfprotokolleinträgen
Als Service anmelden (SeServiceLogonRight)	J	N	Ausführung von DB2 als Service (Dienst)
Auf diesen Computer über das Netzwerk zugreifen (SeNetworkLogonRight)	J	J	Zulassen von Netzwerkberechtigungsnachweisen (ermöglicht dem DB2-Datenbankmanager die Verwendung der Option LOGON32_LOGON_NETWORK zur Authentifizierung, die Auswirkungen auf die Leistung hat)
Identität eines Clients nach Authentifizierung vortäuschen (SeImpersonatePrivilege)	J	N	Vortäuschen der Identität eines Clients (für Windows erforderlich, damit bestimmte APIs die Identität von DB2-Clients vortäuschen können: ImpersonateLoggedOnUser, ImpersonateSelf, RevertToSelf usw.)
Seiten im Speicher sperren (SeLockMemoryPrivilege)	J	N	Unterstützung großer Seiten
Globale Objekte erstellen (SeCreateGlobalPrivilege)	J	J	Terminal-Server-Unterstützung (unter Windows erforderlich)

Hinweise zu Vista: Benutzerzugriffssteuerungsfunktion

Die Benutzerzugriffssteuerungsfunktion (User Access Control, UAC) von Windows Vista hat die nachfolgend erläuterten Auswirkungen auf das DB2-Datenbanksystem.

Starten von Anwendungen mit vollständigen Administratorberechtigungen

Unter Vista werden Anwendungen standardmäßig nur mit den Standardbenutzerrechten gestartet, auch wenn der Benutzer ein lokaler Administrator ist. Wenn Sie eine Anwendung mit weiter reichenden Zugriffsrechten starten wollen, müssen Sie den Befehl über ein Befehlsfenster ausführen, das mit den vollständigen Administratorberechtigungen ausgeführt wird. Der DB2-Installationsprozess erstellt eine Verknüpfung mit dem Namen "Befehlsfenster - Administrator" speziell für Vista-Benutzer. Es wird empfohlen, das Fenster über diese Verknüpfung zu starten, wenn Sie Administratorbefehle ausführen möchten.

Wenn Sie nicht über die vollständigen Administratorberechtigungen verfügen und versuchen, DB2-Verwaltungstasks über eine Eingabeaufforderung oder ein grafisch orientiertes Tool unter Windows Vista auszuführen, können verschiedene Fehlernachrichten ausgegeben werden, die implizieren, dass Ihnen der Zugriff verweigert wurde und die Tasks nicht erfolgreich ausgeführt werden können.

Um festzustellen, ob die Aktion, die Sie ausführen wollen, als Verwaltungstask betrachtet wird, überprüfen Sie, ob Folgendes auf sie zutrifft:

- Sie erfordert die Berechtigung SYSADM, SYSCTRL oder SYSMAINT.
- Sie ändert Registrierungsschlüssel unter dem Zweig HKLM in der Registrierdatenbank.
- Sie schreibt in die Verzeichnisse unter dem Verzeichnis für Programmdateien (Program Files).

Zum Beispiel werden die folgenden Aktionen alle als Verwaltungstasks betrachtet:

- Erstellen und Löschen von DB2-Instanzen
- Starten und Stoppen von DB2-Instanzen
- Erstellen von Datenbanken
- Aktualisieren von Konfigurationsparametern des Datenbankmanagers oder Konfigurationsparametern des DB2-Verwaltungsservers (DAS)
- Aktualisieren von CLI-Konfigurationsparametern und Konfigurieren von Systemdatenquellennamen (DSN)
- Starten der DB2-Tracefunktion
- Ausführen des Dienstprogramms db2pd
- Ändern von Variablen der DB2-Profilregistrierdatenbank

Zur Lösung des Problems müssen Sie DB2-Verwaltungstasks über eine Eingabeaufforderung bzw. ein grafisch orientiertes Tool ausführen, die bzw. das mit den vollständigen Administratorberechtigungen ausgeführt wird. Zum Starten einer Eingabeaufforderung oder eines grafisch orientierten Tools mit vollständigen Administratorberechtigungen klicken Sie die Verknüpfung mit der rechten Maustaste an und wählen die Option **Als Administrator ausführen** aus.

Anmerkung: Wenn die erweiterte Sicherheit aktiviert ist, müssen Sie darüber hinaus Mitglied der Gruppe DB2ADMNS sein, um grafische Verwaltungstools (z. B. den Befehlseditor oder die Steuerzentrale) starten zu können.

Speicherposition von Benutzerdaten

Benutzerdaten (z. B. Dateien unter den Instanzverzeichnissen) werden in ProgramData\IBM\DB2*name_der_kopie* gespeichert. Dabei ist *name_der_kopie* der Name der DB2-Kopie. (Standardmäßig hat die erste installierte Kopie den Namen DB2COPY1.) Unter anderen Windows-Versionen als Vista werden Benutzerdaten im Verzeichnis Dokumente und Einstellungen\All Users\Application Data\IBM\DB2*name_der_kopie* gespeichert.

DB2- und UNIX-Sicherheit

UNIX-Plattform: Sicherheitsaspekte für Benutzer

Die DB2-Datenbank unterstützt keine direkte Funktion der Rootberechtigung als Datenbankadministrator. Sie sollten den Befehl su - <instanzeigner> als Datenbankadministrator verwenden.

Aus Sicherheitsgründen sollten Sie den Instanznamen nicht als Namen der Gruppe für abgeschirmte Funktionen (Fenced ID) verwenden. Wenn Sie jedoch nicht beabsichtigen, abgeschirmte UDFs oder gespeicherte Prozeduren zu verwenden, können Sie die Fenced ID auf den Instanznamen setzen, anstatt eine weitere Benutzer-ID zu erstellen.

Es wird empfohlen, eine Benutzer-ID zu erstellen, die als dieser Gruppe zugeordnet erkannt wird. Der Benutzer für abgeschirmte UDFs und gespeicherte Prozeduren wird als Parameter der Prozedur zur Instanzerstellung angegeben (db2icrt ... -u <AbgeschirmtID>). Dies ist nicht erforderlich, wenn Sie die DB2-Clients oder das DB2 Software Developer's Kit installieren.

Position des Instanzverzeichnisses

Für Rootinstallationen unter Linux und UNIX erstellt der Befehl db2icrt das Hauptverzeichnis für die SQL-Bibliothek (sqllib) unter dem Ausgangsverzeichnis des Instanzeigners.

Unter Windows-Betriebssystemen befindet sich das Instanzverzeichnis im Unterverzeichnis /sqllib des Verzeichnisses, in dem das DB2-Datenbanksystem installiert wurde.

DB2- und Linux-Sicherheit

Unterstützung für Kennwortänderung (Linux)

DB2-Datenbankprodukte stellen eine Unterstützung für die Änderung von Kennwörtern unter Linux-Betriebssystemen bereit.

Diese Unterstützung wird durch die Verwendung der Sicherheits-Plug-in-Bibliotheken mit den Namen IBMOSchgpwdclient.so und IBMOSchgpwdserver.so implementiert.

Zur Aktivierung der Unterstützung für die Kennwortänderung unter Linux setzen Sie den Konfigurationsparameter **CLNT_PW_PLUGIN** des Datenbankmanagers auf den Wert IBMOSchgpwdclient und den Konfigurationsparameter **SRVCON_PW_PLUGIN** des Datenbankmanagers auf den Wert IBMOSchgpwdserver.

Darüber hinaus müssen Sie eine PAM-Konfigurationsdatei mit dem Namen ″db2″ im Verzeichnis /etc/pam.d erstellen.

Implementieren eines Plug-ins zur Kennwortänderung (Linux)

Zur Aktivierung der Unterstützung für das Ändern von Kennwörtern in DB2-Datenbankprodukten unter Linux müssen Sie die DB2-Instanz zur Verwendung der Sicherheits-Plug-ins IBMOSchgpwdclient und IBMOSchgpwdserver konfigurieren.

Die Plug-in-Bibliotheken befinden sich in den folgenden Verzeichnissen:
- *INSTHOME*/sqllib/security*XX*/plugin/IBM/client/IBMOSchgpwdclient.so
- *INSTHOME*/sqllib/security*XX*/plugin/IBM/server/IBMOSchgpwdserver.so

Dabei ist *INSTHOME* das Ausgangsverzeichnis des Instanzeigners, und *securityXX* ist abhängig von der Bit-Breite der Instanz entweder security32 oder security64.

Führen Sie die folgenden Schritte aus, um die Sicherheits-Plug-ins in einer DB2-Instanz zu implementieren:

1. Melden Sie sich als Benutzer mit Rootberechtigung an.
2. Erstellen Sie eine PAM-Konfigurationsdatei: /etc/pam.d/db2

   Stellen Sie sicher, dass die Datei die richtige Gruppe von Regeln enthält, wie dies durch Ihren Systemadministrator definiert wurde. Zum Beispiel können unter SLES 9 folgende Angaben verwendet werden:

   ```
   auth     required pam_unix2.so    nullok
   account  required pam_unix2.so
   password required pam_pwcheck.so  nullok tries=1
   password required pam_unix2.so    nullok use_authtok use_first_pass
   session  required pam_unix2.so
   ```

   Unter RHEL 4 können folgende Angaben verwendet werden:

   ```
   #%PAM-1.0
   auth     required   /lib/security/$ISA/pam_env.so
   auth     sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
   auth     required   /lib/security/$ISA/pam_deny.so
   account  required   /lib/security/$ISA/pam_unix.so
   account  sufficient /lib/security/$ISA/pam_succeed_if.so uid < 100 quiet
   account  required   /lib/security/$ISA/pam_permit.so
   password requisite  /lib/security/$ISA/pam_cracklib.so retry=3 dcredit=-1
                        ucredit=-1
   password sufficient /lib/security/$ISA/pam_unix.so nullok use_authtok
                        md5 shadow remember=3
   password required   /lib/security/$ISA/pam_deny.so
   session  required   /lib/security/$ISA/pam_limits.so
   session  required   /lib/security/$ISA/pam_unix.so
   ```

3. Aktivieren Sie die Sicherheits-Plug-ins in der DB2-Instanz:
   a. Aktualisieren Sie den Konfigurationsparameter **SRVCON_PW_PLUGIN** des Datenbankmanagers mit dem Wert IBMOSchgpwdserver:

      ```
      db2 update dbm cfg using srvcon_pw_plugin IBMOSchgpwdserver
      ```

   b. Aktualisieren Sie den Konfigurationsparameter **CLNT_PW_PLUGIN** des Datenbankmanagers mit dem Wert IBMOSchgpwdclient:

      ```
      db2 update dbm cfg using CLNT_PW_PLUGIN IBMOSchgpwdclient
      ```

   c. Stellen Sie sicher, dass entweder der Konfigurationsparameter **SRVCON_AUTH** des Datenbankmanagers auf den Wert CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP gesetzt wird oder der Konfigurationsparameter **SRVCON_AUTH** des Datenbankmanagers auf den Wert NOT_SPECIFIED und der Parameter **AUTHENTICATION** auf den Wert CLIENT, SERVER, SERVER_ENCRYPT, DATA_ENCRYPT oder DATA_ENCRYPT_CMP gesetzt werden.

Anhang A. Übersicht über die technischen Informationen zu DB2

Die technischen Informationen zu DB2 stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2-Informationszentrale
  - Themen (zu Tasks, Konzepten und Referenzinformationen)
  - Hilfe für DB2-Tools
  - Beispielprogramme
  - Lernprogramme
- DB2-Bücher
  - PDF-Dateien (für den Download verfügbar)
  - PDF-Dateien (auf der DB2-PDF-DVD)
  - Gedruckte Bücher
- Befehlszeilenhilfe
  - Hilfe für Befehle
  - Hilfe für Nachrichten

Anmerkung: Die Themen der DB2-Informationszentrale werden häufiger aktualisiert als die PDF- und Hardcopybücher. Um stets die neuesten Informationen zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald diese verfügbar sind, oder die DB2-Informationszentrale unter ibm.com aufrufen.

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2, wie beispielsweise technische Hinweise (Technotes), White Papers und IBM Redbooks, online über ibm.com zugreifen. Rufen Sie die Website 'DB2 Information Management - Software - Library' unter http://www.ibm.com/software/data/sw-library/ auf.

Feedback zur Dokumentation

Senden Sie uns Ihr Feedback zur DB2-Dokumentation! Wenn Sie Anregungen zur Verbesserung der DB2-Dokumentation haben, senden Sie eine E-Mail an db2docs@ca.ibm.com. Das DB2-Dokumentationsteam bearbeitet das gesamte Feedback, kann jedoch nicht im Einzelnen auf Ihre E-Mails antworten. Nennen Sie uns, wenn möglich, konkrete Beispiele, sodass wir die Problemstellung besser beurteilen können. Wenn Sie uns Feedback zu einem bestimmten Thema oder einer bestimmten Hilfedatei senden, geben Sie den entsprechenden Titel sowie die URL an.

Verwenden Sie diese E-Mail-Adresse nicht, wenn Sie sich an die DB2-Kundenunterstützung wenden möchten. Wenn ein technisches Problem bei DB2 vorliegt, das Sie mithilfe der Dokumentation nicht beheben können, fordern Sie beim zuständigen IBM Service-Center Unterstützung an.

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format

Die folgenden Tabellen enthalten eine Beschreibung der DB2-Bibliothek, die im IBM Publications Center unter www.ibm.com/shop/publications/order zur Verfügung steht. Über die folgende Adresse können Sie englische Handbücher im PDF-Format sowie übersetzte Versionen zu DB2 Version 9.5 herunterladen: www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

In den Tabellen sind die Bücher, die in gedruckter Form zur Verfügung stehen, gekennzeichnet; möglicherweise sind diese in Ihrem Land oder Ihrer Region jedoch nicht verfügbar.

Die Formnummer wird bei jeder Aktualisierung eines Handbuchs erhöht. Anhand der nachfolgenden Liste können Sie sicherstellen, dass Sie die jeweils neueste Version des Handbuchs lesen.

Anmerkung: Die DB2-Informationszentrale wird häufiger aktualisiert als die PDF- und Hardcopybücher.

Tabelle 52. Technische Informationen zu DB2

Name	IBM Form	In gedruckter Form verfügbar
Administrative API Reference	SC23-5842-01	Ja
Administrative Routines and Views	SC23-5843-01	Nein
Call Level Interface Guide and Reference, Volume 1	SC23-5844-01	Ja
Call Level Interface Guide and Reference, Volume 2	SC23-5845-01	Ja
Command Reference	SC23-5846-01	Ja
Dienstprogramme für das Versetzen von Daten Handbuch und Referenz	SC12-3917-01	Ja
Datenrecovery und hohe Verfügbarkeit Handbuch und Referenz	SC12-3919-01	Ja
Datenserver, Datenbanken und Datenbankobjekte	SC12-3912-001	Ja
Datenbanksicherheit	SC12-3914-01	Ja
Developing ADO.NET and OLE DB Applications	SC23-5851-01	Ja
Developing Embedded SQL Applications	SC23-5852-01	Ja
Developing Java Applications	SC23-5853-01	Ja
Developing Perl and PHP Applications	SC23-5854-01	Nein
Developing User-defined Routines (SQL and External)	SC23-5855-01	Ja
Getting Started with Database Application Development	GC23-5856-01	Ja

Tabelle 52. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar
Installation und Verwaltung von DB2 unter Linux und Windows - Erste Schritte	GC12-3922-01	Ja
Internationalisierung	SC12-3916-01	Ja
Fehlernachrichten, Band 1	GI11-3098-00	Nein
Fehlernachrichten, Band 2	GI11-3099-00	Nein
Migration	GC12-3921-01	Ja
Net Search Extender Verwaltung und Benutzerhandbuch	SC12-3979-01	Ja
Partitionierung und Clustering	SC12-3915-01	Ja
Query Patroller Verwaltung und Benutzerhandbuch	SC12-3977-00	Ja
IBM Data Server-Clients - Einstieg	GC12-3924-01	Nein
DB2-Server - Einstieg	GC12-3923-01	Ja
Spatial Extender und Geodetic Data Management Feature Benutzer- und Referenzhandbuch	SC12-3978-01	Ja
SQL Reference, Volume 1	SC23-5861-01	Ja
SQL Reference, Volume 2	SC23-5862-01	Ja
Systemmonitor Handbuch und Referenz	SC12-3918-01	Ja
Fehlerbehebung	GI11-3097-01	Nein
Optimieren der Datenbankleistung	SC12-3913-01	Ja
Lernprogramm für Visual Explain	SC12-3932-00	Nein
Neue Funktionen	SC12-3928-01	Ja
Workload-Manager Handbuch und Referenz	SC12-3929-01	Ja
pureXML - Handbuch	SC12-3930-01	Ja
XQuery - Referenz	SC12-3931-01	Nein

Tabelle 53. Technische Informationen zu DB2 Connect

Name	IBM Form	In gedruckter Form verfügbar
DB2 Connect Personal Edition - Einstieg	GC12-3926-01	Ja
DB2 Connect-Server - Einstieg	GC12-3927-01	Ja
DB2 Connect Benutzerhandbuch	SC12-3925-01	Ja

Tabelle 54. Technische Informationen zu Information Integration

Name	IBM Form	I In gedruckter Form verfügbar
Information Integration: Föderierte Systeme - Verwaltung	SC12-3759-01	Ja
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-02	Ja
Information Integration: Konfiguration föderierter Datenquellen	SC12-3777-01	Nein
Information Integration: SQL Replication - Handbuch und Referenz	SC12-3782-01	Ja
Information Integration: Replikation und Event-Publishing - Einführung	GC12-3779-01	Ja

Bestellen gedruckter DB2-Bücher

Gedruckte DB2-Bücher können Sie in den meisten Ländern oder Regionen online bestellen. Das Bestellen gedruckter DB2-Bücher ist stets über den zuständigen IBM Ansprechpartner möglich. Beachten Sie hierbei bitte, dass einige Softcopybücher auf der DVD mit der *DB2-PDF-Dokumentation* nicht in gedruckter Form verfügbar sind. So sind beispielsweise die beiden Bände des Handbuchs *DB2 Fehlernachrichten* nicht in gedruckter Form erhältlich.

Gedruckte Versionen vieler DB2-Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können gegen eine Gebühr bei IBM bestellt werden. Abhängig vom jeweiligen Land bzw. der jeweiligen Region können Sie Bücher möglicherweise online über das IBM Publications Center bestellen. Ist im jeweiligen Land bzw. der jeweiligen Region keine Onlinebestellung möglich, können Sie gedruckte DB2-Bücher stets über den zuständigen IBM Ansprechpartner bestellen. Nicht alle Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können in gedruckter Form bestellt werden.

Anmerkung: Über http://publib.boulder.ibm.com/infocenter/db2luw/v9r5 haben Sie Zugriff auf die DB2-Informationszentrale, wo Sie die neueste und umfassendste DB2-Dokumentation finden.

Gehen Sie wie folgt vor, um gedruckte DB2-Bücher zu bestellen:

- Informationen dazu, ob in Ihrem Land oder Ihrer Region die Bestellung von gedruckten DB2-Büchern möglich ist, finden Sie auf der Website mit dem IBM Publications Center unter http://www.ibm.com/shop/publications/order. Wählen Sie ein Land, eine Region oder eine Sprache aus, um die Bestellinformationen für Veröffentlichungen aufzurufen, und führen Sie dann die entsprechenden Schritte des Bestellverfahrens für Ihr Land bzw. Ihre Region aus.
- Gehen Sie wie folgt vor, um gedruckte DB2-Bücher beim zuständigen IBM Ansprechpartner zu bestellen:
  1. Kontaktinformationen zum zuständigen Ansprechpartner finden Sie auf einer der folgenden Websites:
     – IBM Verzeichnis weltweiter Kontakte unter www.ibm.com/planetwide.

– Website mit IBM Veröffentlichungen unter http://www.ibm.com/shop/publications/order. Wählen Sie das gewünschte Land, die gewünschte Region oder die gewünschte Sprache aus, um auf die entsprechende Homepage mit Veröffentlichungen Ihres Landes bzw. Ihrer Region zuzugreifen. Folgen Sie auf dieser Seite dem Link für Informationen zu dieser Site ("About this Site").

2. Geben Sie bei Ihrem Anruf an, dass Sie eine DB2-Veröffentlichung bestellen möchten.
3. Teilen Sie dem zuständigen Ansprechpartner die Titel und Formularnummern der Bücher mit, die Sie bestellen möchten. Titel und Formularnummern finden Sie unter „Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format" auf Seite 286.

Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2 gibt für Bedingungen, die aufgrund einer SQL-Anweisung generiert werden können, einen SQLSTATE-Wert zurück. Die SQLSTATE-Hilfe erläutert die Bedeutung der SQL-Statuswerte und der SQL-Statusklassencodes.

Zum Aufrufen der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

```
? sqlstate oder ? klassencode
```

Hierbei steht *sqlstate* für einen gültigen fünfstelligen SQL-Statuswert und *klassencode* für die ersten beiden Ziffern dieses Statuswertes.

So kann beispielsweise durch die Eingabe von `? 08003` Hilfe für den SQL-Statuswert 08003 angezeigt werden, durch die Eingabe von `? 08` Hilfe für den Klassencode 08.

Zugriff auf verschiedene Versionen der DB2-Informationszentrale

Für Themen aus DB2 Version 9.5 lautet die URL der DB2-Informationszentrale http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/.

Für Themen aus DB2 Version 9 lautet die URL der DB2-Informationszentrale http://publib.boulder.ibm.com/infocenter/db2luw/v9/.

Für Themen aus DB2 Version 8 lautet die URL der Informationszentrale (Version 8, 'Information - Unterstützung') http://publib.boulder.ibm.com/infocenter/db2luw/v8/.

Anzeigen von Themen in der gewünschten Sprache in der DB2-Informationszentrale

In der DB2-Informationszentrale werden Themen, wenn möglich, in der Sprache angezeigt, die in den Vorgaben Ihres Browsers angegeben ist. Falls ein Thema nicht in die gewünschte Sprache übersetzt wurde, wird es in der DB2-Informationszentrale in Englisch angezeigt.

- Um Themen in der gewünschten Sprache im Browser 'Internet Explorer' anzuzeigen, gehen Sie wie folgt vor:
  1. Klicken Sie im Internet Explorer **Extras** —> **Internetoptionen...** —> **Sprachen...** an. Das Fenster **Spracheinstellung** wird geöffnet.

2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
   – Klicken Sie den Knopf **Hinzufügen...** an, um eine neue Sprache zur Liste hinzuzufügen.

     Anmerkung: Das Hinzufügen einer Sprache bedeutet nicht zwangsläufig, dass der Computer über die erforderlichen Schriftarten verfügt, um die Themen in der gewünschten Sprache anzuzeigen.
   – Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Löschen Sie den Inhalt des Browser-Cache, und aktualisieren Sie anschließend die Seite, um die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.

- Um Themen in der gewünschten Sprache in einem Firefox- oder Mozilla-Browser anzuzeigen, gehen Sie wie folgt vor:
  1. Wählen Sie den Knopf im Bereich **Languages** des Dialogfensters **Tools** —> **Options** —> **Advanced** aus. Die Anzeige für die Auswahl der Sprache wird im Fenster mit den Einstellungen aufgerufen.
  2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
     – Wenn Sie eine neue Sprache zur Liste hinzufügen möchten, klicken Sie den Knopf **Add...** an, um eine Sprache im entsprechenden Fenster auszuwählen.
     – Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Move Up** aus, bis die Sprache an erster Stelle in der Liste steht.
  3. Löschen Sie den Inhalt des Browser-Cache, und aktualisieren Sie anschließend die Seite, um die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.

Bei einigen Kombinationen aus Browser und Betriebssystem müssen Sie möglicherweise auch die Ländereinstellungen des Betriebssystems in die gewünschte Locale und Sprache ändern.

Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale

Wenn Sie die DB2-Informationszentrale lokal installiert haben, können Sie Dokumentationsaktualisierungen von IBM abrufen und installieren.

Zur Aktualisierung der lokal installierten DB2-Informationszentrale sind die folgenden Schritte erforderlich:

1. Stoppen Sie die DB2-Informationszentrale auf Ihrem Computer, und starten Sie die Informationszentrale im Standalone-Modus erneut. Die Ausführung der Informationszentrale im Standalone-Modus verhindert, dass andere Benutzer in Ihrem Netz auf die Informationszentrale zugreifen, und ermöglicht das Anwenden von Aktualisierungen. DB2-Informationszentralen, deren Installation nicht als Administrator oder Root ausgeführt wurde, werden stets im Standalone-Modus ausgeführt.

2. Verwenden Sie die Aktualisierungsfunktion, um zu prüfen, welche Aktualisierungen verfügbar sind. Falls Aktualisierungen verfügbar sind, die Sie installieren möchten, können Sie die Aktualisierungsfunktion verwenden, um diese abzurufen und zu installieren.

   Anmerkung: Wenn es in der verwendeten Umgebung erforderlich ist, die Aktualisierungen für die DB2-Informationszentrale auf einer Maschine zu installieren, die nicht über ein Verbindung zum Internet verfügt, müssen Sie die Aktualisierungssite auf ein lokales Dateisystem spiegeln und dabei eine Maschine verwenden, die mit dem Internet verbunden ist und auf der die DB2-Informationszentrale installiert ist. Wenn viele Benutzer Ihres Netzes die Dokumentationsaktualisierungen installieren sollen, können Sie die Zeit, die jeder einzelne Benutzer für die Aktualisierungen benötigt, reduzieren, indem Sie die Aktualisierungssite lokal spiegeln und ein Proxy dafür erstellen. Ist dies der Fall, verwenden Sie die Aktualisierungsfunktion, um die Pakete abzurufen. Die Aktualisierungsfunktion ist jedoch nur im Standalone-Modus verfügbar.
3. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale, und starten Sie die DB2-Informationszentrale auf Ihrem Computer erneut.

Anmerkung: Unter Windows Vista müssen Sie zur Ausführung der nachfolgend aufgeführten Befehle über Administratorberechtigung verfügen. Zum Starten einer Eingabeaufforderung oder eines Grafiktools mit vollen Administratorberechtigungen klicken Sie mit der rechten Maustaste auf die Verknüpfung, und wählen Sie **Als Administrator ausführen** aus.

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte DB2-Informationszentrale zu aktualisieren:

1. Stoppen Sie die DB2-Informationszentrale.
   - Unter Windows klicken Sie **Start** → **Einstellungen** → **Systemsteuerung** → **Verwaltung** → **Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an, und wählen Sie **Stoppen** aus.
   - Unter Linux: Geben Sie den folgenden Befehl ein:
     ```
     /etc/init.d/db2icdv95 stop
     ```
2. Starten Sie die Informationszentrale im Standalone-Modus.
   - Unter Windows:
     a. Öffnen Sie ein Befehlsfenster.
     b. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis <Programme>\IBM\DB2 Information Center\Version 9.5 installiert, wobei <Programme> das Verzeichnis der Programmdateien (Program Files) angibt.
     c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis doc\bin.
     d. Führen Sie die Datei help_start.bat aus:
        ```
        help_start.bat
        ```
   - Unter Linux:
     a. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis /opt/ibm/db2ic/V9.5 installiert.
     b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis doc/bin.
     c. Führen Sie das Script help_start aus:
        ```
        help_start
        ```

Der standardmäßig auf dem System verwendete Web-Browser wird aufgerufen und zeigt die Standalone-Informationszentrale an.

3. Klicken Sie den Aktualisierungsknopf ( ) an. Klicken Sie im rechten Fenster der Informationszentrale den Knopf für die Suche nach Aktualisierungen an. Eine Liste der Aktualisierungen für die vorhandene Dokumentation wird angezeigt.
4. Wählen Sie zum Initiieren des Installationsprozesses die gewünschten Aktualisierungen aus, und klicken Sie anschließend den Knopf für die Installation der Aktualisierungen an.
5. Klicken Sie nach Abschluss des Installationsprozesses **Fertig stellen** an.
6. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale:
   - Unter Windows: Navigieren Sie in das Verzeichnis doc\bin des Installationsverzeichnisses, und führen Sie die Datei help_end.bat aus:
     ```
     help_end.bat
     ```
     Anmerkung: Die Stapeldatei help_end enthält die Befehle, die erforderlich sind, um die Prozesse, die mit der Stapeldatei help_start gestartet wurden, ordnungsgemäß zu beenden. Verwenden Sie nicht die Tastenkombination `Strg+C` oder eine andere Methode, um help_start.bat zu beenden.
   - Unter Linux: Navigieren Sie in das Verzeichnis doc/bin des Installationsverzeichnisses, und führen Sie das Script help_end aus:
     ```
     help_end
     ```
     Anmerkung: Das Script help_end enthält die Befehle, die erforderlich sind, um die Prozesse, die mit dem Script help_start gestartet wurden, ordnungsgemäß zu beenden. Verwenden Sie keine andere Methode, um das Script help_start zu beenden.
7. Starten Sie die DB2-Informationszentrale erneut.
   - Unter Windows klicken Sie **Start** → **Einstellungen** → **Systemsteuerung** → **Verwaltung** → **Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an, und wählen Sie **Start** aus.
   - Unter Linux: Geben Sie den folgenden Befehl ein:
     ```
     /etc/init.d/db2icdv95 start
     ```

In der aktualisierten DB2-Informationszentrale werden die neuen und aktualisierten Themen angezeigt.

DB2-Lernprogramme

Die DB2-Lernprogramme unterstützen Sie dabei, sich mit den unterschiedlichen Aspekten der DB2-Produkte vertraut zu machen. Die Lerneinheiten bieten eine in einzelne Schritte unterteilte Anleitung.

Vorbereitungen

Die XHTML-Version des Lernprogramms kann über die Informationszentrale unter http://publib.boulder.ibm.com/infocenter/db2help/ angezeigt werden.

In einigen der Lerneinheiten werden Beispieldaten und Codebeispiele verwendet. Informationen zu bestimmten Voraussetzungen für die Ausführung der Tasks finden Sie in der Beschreibung des Lernprogramms.

DB2-Lernprogramme

Klicken Sie zum Anzeigen des Lernprogramms den Titel an.

„pureXML" in *pureXML - Handbuch*
Einrichten einer DB2-Datenbank, um XML-Daten zu speichern und Basisoperationen mit dem nativen XML-Datenspeicher auszuführen.

„Visual Explain" in *Lernprogramm für Visual Explain*
Analysieren, Optimieren und Anpassen von SQL-Anweisungen zur Leistungsverbesserung mithilfe von Visual Explain.

Informationen zur Fehlerbehebung in DB2

Eine breite Palette verschiedener Informationen zur Fehlerbestimmung und Fehlerbehebung steht zur Verfügung, um Sie bei der Verwendung von DB2-Produkten zu unterstützen.

DB2-Dokumentation
Informationen zur Fehlerbehebung stehen im Handbuch DB2-Fehlerbehebung oder im Abschnitt zur Unterstützung und Fehlerbehebung der DB2-Informationszentrale zur Verfügung. Dort finden Sie Informationen dazu, wie Sie Probleme mithilfe der DB2-Diagnosetools und -Dienstprogramme eingrenzen und identifizieren können, Lösungen für einige der häufigsten Probleme sowie weitere Hinweise zur Behebung von Fehlern und Problemen, die bei der Verwendung der DB2-Produkte auftreten können.

DB2-Website mit technischer Unterstützung
Auf der DB2-Website mit technischer Unterstützung finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die Website mit technischer Unterstützung enthält Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports) und Fehlerkorrekturen, Fixpacks sowie weiteren Ressourcen. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Rufen Sie die DB2-Website mit technischer Unterstützung unter http://www.ibm.com/software/data/db2/udb/support.html auf.

Bedingungen

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

Persönliche Nutzung: Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Kommerzielle Nutzung: Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens nicht vervielfältigen, weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Abgesehen von den hier gewährten Berechtigungen erhalten Sie keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum.

IBM behält sich das Recht vor, die in diesem Dokument gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Informationen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf ″as-is″-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit oder die Freiheit der Rechte Dritter zur Verfügung gestellt.

Anhang B. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der Produkte, Programme oder Services können auch andere ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Dieses Dokument enthält möglicherweise Links oder Verweise auf Websites und Ressourcen anderer Anbieter. Es bestehen keine Zusicherungen, Gewährleistungen oder Verpflichtungen von IBM hinsichtlich der Websites oder Ressourcen anderer Anbieter, auf die im vorliegenden Dokument verwiesen wird, Zugriff besteht oder Links vorhanden sind. Ein Link auf eine Website eines anderen Anbieters bedeutet nicht, dass IBM den Inhalt und die Verwendung dieser Website billigt oder deren Eigentümer anerkennt. Darüber hinaus ist IBM nicht an Transaktionen beteiligt und übernimmt keine Verantwortung für Transaktionen zwischen Ihnen und anderen Anbietern, auch wenn die Informationen (oder Links) zu diesen Anbietern auf einer IBM Website zur Verfügung stehen. IBM ist nicht für die Verfügbarkeit solcher externen Sites oder Ressourcen verantwortlich und übernimmt keine Verantwortung oder Haftung für Inhalte, Services, Produkte oder sonstiges Material, die bzw. das auf diesen oder über diese Sites oder Ressourcen verfügbar sind. Die Software anderer Anbieter unterliegt den Lizenzbedingungen der jeweiligen Software.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung sowie der Allgemeinen Geschäftsbedingungen von IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichen Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (*Name Ihrer Firma*) (*Jahr*). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *_Jahr/Jahre angeben_*. Alle Rechte vorbehalten.

Marken

Folgende Namen sind Marken oder eingetragene Marken der International Business Machines Corporation in den USA und/oder anderen Ländern.

pureXML	OS/390
DB2 Connect	DB2 Universal Database
Redbooks	z/OS
developerWorks	System i
Informix	IBM
DB2	zSeries
AIX	System z9
Lotus	Tivoli
DRDA	System z
Domino	ibm.com
POWER	WebSphere

Folgende Namen sind Marken oder eingetragene Marken anderer Unternehmen.

- Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.
- Java und alle Java-basierten Marken sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- UNIX ist eine eingetragene Marke von The Open Group in den USA und/oder anderen Ländern.
- Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

SC12-3914-01