

IBM DB2 Universal Database



Initiation à SQL

Version 7

IBM DB2 Universal Database



Initiation à SQL

Version 7

Important

Avant d'utiliser ce manuel et le produit associé, prenez connaissance des informations générales figurant à l'«Annexe C. Remarques» à la page 123.

Réf. US : SC09-2973-00

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2000. Tous droits réservés.

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

Table des matières

Avis aux lecteurs canadiens	v	Utilisation d'une clause WHERE avec une clause GROUP BY	32
Bienvenue	vii	Utilisation de la clause HAVING après la clause GROUP BY	32
Bibliographie	vii		
Conventions de mise en évidence	viii		
Chapitre 1. SQL et les bases de données relationnelles	1		
Chapitre 2. Organisation des données	3	Chapitre 5. Expressions et sous-requêtes	35
Tables	3	Instructions FULLSELECT scalaires	35
Vues	4	Spécifications de transtypage de données	36
Schémas	4	Expressions CASE	36
Types de données	5	Expressions de table	38
		Expressions de table imbriquées	38
		Expressions de table communes	39
		Noms de corrélation	40
		Sous-requêtes corrélées	41
		Implémentation d'une sous-requête corrélée	43
Chapitre 3. Création de tables et de vues	9		
Création de tables	9	Chapitre 6. Utilisation d'opérateurs et de prédicats dans des requêtes	47
Insertion de données	10	Association de requêtes à l'aide d'opérateurs ensemblistes	47
Modification de données	12	Opérateur UNION	47
Suppression de données	13	Opérateur EXCEPT	48
Création de vues	13	Opérateur INTERSECT	49
Utilisation de vues pour manipuler des données	15	Prédicats	50
		Utilisation du prédicat IN	50
		Utilisation du prédicat BETWEEN	51
		Utilisation du prédicat LIKE	51
		Utilisation du prédicat EXISTS	51
		Prédicats quantifiés	52
Chapitre 4. Utilisation d'instructions SQL pour accéder aux données	17		
Connexion à une base de données	18	Chapitre 7. Fonctions SQL évoluées	55
Localisation des erreurs	18	Respect des règles de gestion à l'aide de contraintes et de déclencheurs	55
Sélection de colonnes	19	Clés	56
Sélection de lignes	20	Contraintes d'unicité	56
Classement des lignes	23	Contraintes d'intégrité référentielle	56
Élimination des lignes en double	24	Contraintes de vérification de table	57
Ordre des opérations	25	Déclencheurs	58
Utilisation d'expressions pour calculer des valeurs	25	Jointures	62
Dénomination des expressions	26	Requêtes complexes	67
Sélection de données dans plusieurs tables	26	Requêtes ROLLUP et CUBE	67
Utilisation d'une sous-requête	28	Requêtes récursives	67
Utilisation de fonctions	28	Fonctions OLAP	68
Fonctions de colonne	29		
Fonctions scalaires	30		
Fonctions de table	31		
Groupage	31		

Chapitre 8. Personnalisation et amélioration de la manipulation des données	71
Types définis par l'utilisateur	71
Fonctions définies par l'utilisateur (UDF)	72
Objets LOB	73
Manipulation des objets LOB	74
Registres spéciaux	74
Présentation des vues de catalogue	75
Sélection de lignes des catalogues système	75

Annexe A. Tables de la base de données SAMPLE	77
Base de données SAMPLE	78
Création de la base de données SAMPLE	78
Effacement de la base de données SAMPLE	79
Table CL_SCHED	79
Table DEPARTMENT	79
Table EMPLOYEE	80
Table EMP_ACT	83
Table EMP_PHOTO	85
Table EMP_RESUME	85
Table IN_TRAY	86
Table ORG	86
Table PROJECT	87
Table SALES	88
Table STAFF	89
Table STAFFG	90
Modèles de fichiers utilisant les types d'objet BLOB et CLOB	92

Photo de Dolores Quintana	92
Fiche descriptive de Dolores Quintana	92
Photo d'Heather A. Nicholls	93
Fiche descriptive d'Heather A. Nicholls	93
Photo de Bruce Adamson	95
Fiche descriptive de Bruce Adamson	95
Photo de James H. Walker	96
Fiche descriptive de James H. Walker	96

Annexe B. Utilisation de la Bibliothèque DB2	99
Manuels imprimés et fichiers au format PDF DB2.	99
Informations sur DB2	99
Impression des manuels au format PDF	112
Commande des manuels imprimés	112
Documentation en ligne DB2	114
Accès à l'aide en ligne	114
Affichage des informations en ligne	116
Assistants DB2	118
Configuration d'un serveur de documents	120
Recherche d'informations en ligne	120

Annexe C. Remarques	123
Marques	126

Index	129
Comment prendre contact avec IBM.	133
Infos produit	133

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens








Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire

correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Bienvenue

Le présent ouvrage vous est destiné si vous désirez vous familiariser avec le langage SQL (Structured Query Language) et les bases de données relationnelles. Vous y trouverez :

- Les concepts de base de SQL utilisé avec DB2.
- Des explications sur la façon de manipuler des bases de données.
- Des exemples de tâches simples.

Si vous êtes administrateur, avant d'étudier les exemples du présent ouvrage :

- Installez et configurez le serveur comme l'explique le document *Mise en route* de votre système d'exploitation. Créez la base de données SAMPLE à l'aide de l'option "Premiers pas". La base de données SAMPLE peut également être créée à partir de la ligne de commande. Pour plus de détails, reportez-vous au manuel *SQL Reference*. Remarque : Évitez de mettre vos propres données dans la base de données SAMPLE de DB2.
- Créez votre ID utilisateur d'administrateur DB2 selon les instructions du document *Mise en route*.

Si vous n'êtes pas administrateur système, assurez-vous que vous avez un ID utilisateur admis ainsi que les droits d'accès et les privilèges requis.

Bibliographie

Vous trouverez des informations utiles dans les publications suivantes :

<i>Mise en route</i>	Informations nécessaires à l'installation et à l'utilisation du gestionnaire de bases de données.
<i>SQL Reference</i>	Informations de référence sur SQL.
<i>Administration Guide</i>	Informations nécessaires à la conception, la mise en oeuvre et la mise à jour d'une base de données, soit dans le cadre d'un accès local, soit dans un environnement client-serveur.
<i>Application Development Guide</i>	Processus de développement des applications et méthodes à utiliser pour coder, compiler et exécuter des programmes d'application utilisant du SQL imbriqué pour accéder à la base de données, ou exécuter des procédures mémorisées à l'aide du langage de procédure SQL (ou d'autres langages de programmation pris en charge).

Conventions de mise en évidence

Le présent document utilise les conventions suivantes :

Gras	Dans les exemples, indique les commandes et les mots clés prédéfinis par le système.
<i>Italiques</i>	Indique : <ul style="list-style-type: none">• Soit l'introduction d'un terme nouveau• Soit une référence à une autre source d'information
MAJUSCULES	Indique : <ul style="list-style-type: none">• Soit des commandes et des mots clés prédéfinis par le système• Soit des exemples de valeurs ou de noms de colonnes spécifiques

Chapitre 1. SQL et les bases de données relationnelles

Dans une *base de données relationnelle*, les données sont conservées dans des *tables*. Une table se compose d'un ensemble de *lignes* et de *colonnes*. Reportez-vous à la figure 1 à la page 4. Les colonnes et les lignes ont été indiquées dans cette figure. Le *langage SQL (Langage d'interrogation structuré)* sert à extraire et à mettre à jour des données, en précisant les colonnes et les tables concernées ainsi que les diverses relations qui existent entre elles.

Le langage SQL est un ensemble d'instructions standard, permettant de définir et manipuler des données dans une base de données relationnelle. Les instructions SQL sont exécutées par un *gestionnaire de bases de données*. Il s'agit d'un programme informatique chargé de gérer les données.

Une base de données relationnelle est *partitionnée* si ses données sont gérées sur plusieurs partitions (également appelées *noeuds*). Il existe un moyen simple de se représenter les partitions en considérant chacune d'entre elles comme un ordinateur physique. Le présent document traite des bases de données à partition unique.

Vous pouvez accéder à la base de données SAMPLE et vous exercer à utiliser le langage SQL en mode *interactif* grâce aux exemples du présent document. Comme interface, utilisez soit l'Interpréteur de commandes, soit le Centre de commande.

Chapitre 2. Organisation des données

Le présent chapitre décrit les concepts importants que sont les *tables*, les *vues* et les *schémas*. Il présente une vue d'ensemble de l'articulation d'une base de données relationnelle. La dernière section présente brièvement certains des types de données les plus importants et les plus courants.

Tables

Les tables sont des structures logiques constituées d'un nombre défini de *colonnes* et d'un nombre variable de *lignes*. Une colonne est un ensemble de valeurs attribuées à des données de même type. Une ligne se compose d'une séquence de valeurs qui composent un enregistrement dans une table. Dans une table, les lignes ne sont pas nécessairement classées. Pour obtenir un classement du résultat, il faut le demander explicitement dans l'instruction SQL qui sélectionne les données de la table. A l'intersection entre chacune des colonnes et lignes se trouve une donnée, appelée *valeur*. Dans la table de la figure 1 à la page 4, "Sanders" est un exemple de valeur.

Une *table de base* créée à l'aide de l'instruction CREATE TABLE reçoit les données de l'utilisateur. Une *table résultat* est un ensemble de lignes sélectionnées par le gestionnaire de bases de données, ou générées à partir d'une ou de plusieurs tables de base, en réponse à une requête.

La figure 1 à la page 4 illustre une partie de table. Les légendes indiquent l'emplacement des lignes et des colonnes.

		Colonne			
Ligne		ID	NAME	DEPT	J
	10		Sanders	20	Mg
	20		Pernal	20	Sa
	30		Marenghi	38	Mg
	40		O'Brien	38	Sa
	50		Hanes	15	Mg
	60		Quigley	38	Sa
				15	Sa

Figure 1. Exemple de table

Vues

Une *vue* est une autre façon de visualiser les données d'une ou de plusieurs tables. Elle ouvre une fenêtre dynamique sur les tables.

Elle permet d'obtenir une présentation différente des mêmes données selon l'utilisateur auquel elle est destinée. Par exemple, plusieurs utilisateurs ont accès à une table de données sur les employés. Un directeur, par exemple, ne voit que les données relatives à ses propres employés. Le responsable du recrutement voit les dates d'embauche de tous les salariés, mais pas leurs salaires, et le responsable financier voit les salaires mais pas les dates d'embauche. Chacun de ces utilisateurs travaille sur des vues dérivées de la table réelle et chaque vue prend l'apparence d'une table et possède un nom.

L'utilisation de vues présente un avantage. Elle permet de contrôler l'accès à des données confidentielles. Ainsi, des personnes différentes n'ont pas accès aux mêmes colonnes ni aux mêmes lignes.

Schémas

Un *schéma* est une collection d'objets nommés (par exemple, des tables et des vues). Il fournit une classification logique des objets de la base de données.

Il est créé implicitement en même temps que la table, la vue ou tout autre objet nommé. Vous pouvez aussi le créer explicitement avec l'instruction CREATE SCHEMA.

Lors de la création d'un objet nommé, vous pouvez le *qualifier* par (lui associer) un nom de schéma particulier. Les noms des objets nommés sont constitués de deux parties. La première est le nom du schéma auquel l'objet est affecté. Si vous ne précisez pas de nom de schéma, l'objet est affecté au schéma par défaut portant comme nom *l'ID autorisation* de la personne exécutant l'instruction.

La méthode adoptée pour exécuter les exemples du présent document utilise le SQL en interactif. Dans ce cas, l'ID autorisation est l'ID utilisateur qui a été précisé avec l'instruction CONNECT. Par exemple, dans une table STAFF, si l'ID utilisateur précisé est USERXYZ, le nom qualifié de la table devient USERXYZ.STAFF. Pour plus de détails sur l'instruction CONNECT, consultez la section «Connexion à une base de données» à la page 18.

Certains noms de schémas sont réservés. Par exemple, les *fonctions intégrées* se trouvent dans le schéma SYSIBM et les *fonctions UDF* appartiennent au schéma SYFUN. Pour plus de détails sur l'instruction CREATE SCHEMA, consultez le document *SQL Reference*.

Types de données

Les types de données définissent les valeurs que peuvent accepter les constantes, les colonnes, les variables SQL, les fonctions, les expressions et les registres spéciaux. Cette section décrit les types de données auxquels les exemples font référence. Pour obtenir une liste complète et une description détaillée des autres types de données, consultez le document *SQL Reference*.

Chaîne de caractères

Une *chaîne de caractères* est une suite d'octets. La longueur de la chaîne est le nombre d'octets de la suite. Si la longueur est égale à zéro, la valeur est appelée *chaîne vide*.

Chaîne de caractères de longueur fixe

CHAR(x) est une chaîne de longueur fixe. La valeur de x, attribut de longueur, doit être comprise entre 1 et 254 inclus.

Chaîne de caractères de longueur variable

Il existe trois types de chaînes de caractères de longueur variable : VARCHAR, LONG VARCHAR et CLOB.

Les types VARCHAR(x) désignent des chaînes de longueur variable. C'est pourquoi une chaîne dont la longueur est égale à 9 peut être insérée dans VARCHAR(15), tout en gardant une longueur de 9 caractères.

Pour plus de détails sur CLOB, consultez la section «Objets LOB» à la page 73.

Chaîne graphique

Une *chaîne graphique* est une séquence de caractères double octet.

Chaîne graphique de longueur fixe

GRAPHIC(x) est une chaîne de longueur fixe. La valeur de x, attribut de longueur, doit être comprise entre 1 et 127 inclus.

Chaîne graphique de longueur variable

Il existe trois types de chaînes graphiques de longueur variable : VARGRAPHIC, LONG VARGRAPHIC et DBCLOB. Pour plus de détails sur DBCLOB, consultez la section «Objets LOB» à la page 73.

Chaîne binaire

Une *chaîne binaire* est une suite d'octets. Elle accueille des données non traditionnelles, comme des illustrations. Un objet BLOB (Binary Large Object) est une chaîne binaire. Pour plus de détails, consultez la section «Objets LOB» à la page 73.

Nombres

Tous les nombres sont accompagnés d'un signe et d'une *précision*. La précision est le nombre de chiffres binaires ou décimaux, signe non compris.

SMALLINT

Le type *SMALLINT* comprend les nombres entiers sur deux octets, avec une précision de 5 chiffres.

INTEGER

Le type *INTEGER* comprend les nombres entiers sur quatre octets, avec une précision de 10 chiffres.

BIGINT

Le type *BIGINT* comprend les nombres entiers sur 8 octets, avec une précision de 19 chiffres.

REAL Le type *REAL* (virgule flottante simple précision) est une approximation sur 32 bits d'un nombre réel.

DOUBLE

Le type *DOUBLE* (virgule flottante double précision) est une approximation sur 64 bits d'un nombre réel. Le type *DOUBLE* est également appelé type *FLOAT*.

DECIMAL(p,e)

Le type *DECIMAL* comprend les nombres décimaux. La position du séparateur décimal est déterminée par la *précision* (*p*) et par *l'échelle* (*e*) du nombre. La précision est le nombre

total de chiffres, qui doit être inférieur à 32. L'échelle est le nombre de chiffres dans la partie fractionnaire d'un nombre. Sa valeur est toujours inférieure ou égale à la valeur de la précision. Une valeur décimale *a*, par défaut, une précision égale à 5 et une échelle égale à 0.

Valeurs de date et/ou d'heure

Les dates, les heures et l'horodatage sont représentés par des valeurs sous la forme d'une chaîne de caractères de 14 chiffres : *aaaaxxjjhhmmss*. Ces valeurs peuvent être utilisées dans certaines opérations arithmétiques et de chaîne. Elles sont compatibles avec certaines chaînes, bien qu'elles ne soient elles-mêmes ni des chaînes ni des nombres.¹

Date Une *date* se compose de trois parties (année, mois et jour).

Heure Une *heure* se compose de trois parties (heure, minute et seconde) et est indiquée au format 24 heures.

Horodatage

L'*horodatage* utilise une valeur composée de sept parties (année, mois, jour, heure, minute, seconde et microseconde) désignant une date et une heure.

Valeur NULL

Une valeur *NULL* est une valeur spéciale, distincte de toutes les autres valeurs. Elle signifie l'absence de toute autre valeur dans la cellule située à l'intersection de la colonne et de la ligne. Tous les types de données peuvent avoir une valeur *NULL*.

Le tableau suivant présente les caractéristiques des types de données utilisés dans les exemples. Tous les types numériques se situent dans une certaine plage définie, que ce tableau rappelle. Vous pouvez vous reporter rapidement à ces informations pour vérifier que vous utilisez correctement un type de données.

1. Dans le présent document, les représentations ISO des dates et des heures sont utilisées.

Type de données	Type	Caractéristiques	Exemple ou plage
CHAR(15)	Chaîne de caractères de longueur fixe	Longueur maximum : 254	'Jour de soleil'
VARCHAR(15)	Chaîne de caractères de longueur variable	Longueur maximum : 32672	'Jour de soleil'
SMALLINT	Numérique	Longueur de 2 octets Précision de 5 chiffres	Entre -32768 et 32767
INTEGER	Numérique	Longueur de 4 octets Précision de 10 chiffres	Entre -2147483648 et 2147483647
BIGINT	Numérique	Longueur de 8 octets Précision de 19 chiffres	Entre -9223372036854775808 et 9223372036854775807
REAL	Numérique	Virgule flottante simple précision - Approximation sur 32 bits	Plage comprise entre -3.402E+38 et -1.175E-37 ou 1.175E-37 et -3.402E+38 ou encore, égale à zéro
DOUBLE	Numérique	Virgule flottante double précision - Approximation sur 64 bits	Plage comprise entre -1.79769E+308 et -2.225E-307 ou 2.225E-307 et 1.79769E+308 ou encore, égale à zéro
DECIMAL(5,2)	Numérique	Précision : 5 Echelle : 2	Plage comprise entre -10**31+1 et 10**31-1
DATE	Date et/ou heure	Valeur en trois parties	1997-10-27
TIME	Date et/ou heure	Valeur en trois parties	13.30.05
TIMESTAMP	Date et/ou heure	Valeur en sept parties	1997-10-27-13.30.05.000000

Pour plus d'informations, voir la table des compatibilités entre types de données dans le document *SQL Reference*.

Chapitre 3. Création de tables et de vues

Le présent chapitre explique comment créer et manipuler des tables et des vues dans DB2 Universal Database. Différents diagrammes et exemples illustrent les relations entre les tables et les vues.

Les sujets suivants sont abordés :

- Création de tables et Création de vues
- Insertion de données
- Modification de données
- Suppression de données
- Utilisation de vues pour manipuler des données

Création de tables

Créez vos tables à l'aide de l'instruction `CREATE TABLE`, en précisant les noms et les types des colonnes, ainsi que les *contraintes*. Vous trouverez une description des contraintes dans la section «Respect des règles de gestion à l'aide de contraintes et de déclencheurs» à la page 55.

L'instruction suivante crée la table `PERS`, similaire à la table `STAFF`, mais comprenant une colonne supplémentaire pour la date de naissance.

```
CREATE TABLE PERS
  (ID          SMALLINT NOT NULL,
   NAME        VARCHAR(9),
   DEPT        SMALLINT WITH DEFAULT 10,
   JOB         CHAR(5),
   YEARS       SMALLINT,
   SALARY      DECIMAL(7,2),
   COMM        DECIMAL(7,2),
   BIRTH_DATE  DATE)
```

Cette instruction crée une table sans données. La section suivante décrit comment insérer des données dans une nouvelle table.

Comme le montre l'exemple, il faut préciser le nom et le type de chacune des colonnes. Les types de données sont décrits dans la section «Types de données» à la page 5. `NOT NULL`, facultatif, indique que des valeurs `NULL` ne sont pas admises dans cette colonne. Les valeurs par défaut sont également facultatives.

De nombreuses autres options peuvent être précisées dans une instruction `CREATE TABLE`, comme les *contraintes d'unicité* ou les *contraintes référentielles*.

Pour plus d'informations sur toutes les options, consultez l'instruction CREATE TABLE dans le document *SQL Reference*.

Insertion de données

A sa création, une table ne contient aucune donnée. Pour y insérer des lignes, utilisez l'instruction INSERT. Cette instruction se présente généralement sous deux formats :

- Vous pouvez utiliser une clause VALUES précisant les valeurs des colonnes d'une ou de plusieurs lignes. Les trois exemples suivants utilisent ce format général pour insérer des données dans des tables.
- Sinon, au lieu de préciser des valeurs à l'aide de la clause VALUES, vous pouvez indiquer une instruction FULLSELECT qui identifiera les colonnes venant d'autres tables et/ou vues.

FULLSELECT fait partie des instructions de sélection utilisées dans les instructions INSERT ou CREATE VIEW, ou qui suivent un prédicat. Une instruction FULLSELECT entre parenthèses est souvent appelée *sous-requête*.

Selon les options par défaut que vous avez choisies lors de la création de la table, dans chaque ligne que vous insérez, soit vous fournissez une valeur pour chaque colonne, soit vous acceptez une valeur par défaut. Les valeurs par défaut des différents types de données sont décrites dans le document *SQL Reference*.

L'instruction suivante utilise une clause VALUES pour insérer une ligne de données dans la table PERS :

```
INSERT INTO PERS  
VALUES (12, 'Harris', 20, 'Sales', 5, 18000, 1000, '1950-1-1')
```

L'instruction suivante utilise la clause VALUES pour insérer trois lignes dans la table PERS, où seuls les ID, les noms et les postes sont connus. Si une colonne est de type NOT NULL sans valeur par défaut, il faut lui donner une valeur.

Dans une instruction CREATE TABLE, la clause NOT NULL d'une définition de colonne peut être suivie des mots WITH DEFAULT. Si la colonne est de type NOT NULL WITH DEFAULT, ou si une constante lui a été attribuée par défaut, comme WITH DEFAULT 10, et si vous ne la citez pas dans la liste des colonnes, la valeur par défaut s'inscrit automatiquement dans cette colonne de la ligne insérée. Par exemple, dans l'instruction CREATE TABLE, seule la colonne DEPT a été définie, et elle a une valeur égale à 10. Par conséquent, le numéro de service 10 s'inscrit dans la colonne DEPT et toute colonne à laquelle il n'a pas été explicitement attribué de valeur prend la valeur NULL.

```

INSERT INTO PERS (NAME, JOB, ID)
VALUES ('Swagerman', 'Prgmr', 500),
       ('Limoges', 'Prgmr', 510),
       ('Li', 'Prgmr', 520)

```

L'instruction suivante renvoie le résultat des insertions :

```

SELECT *
FROM PERS

```

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM	BIRTH_DATE
12	Harris	20	Sales	5	18000.00	1000.00	01/01/1950
500	Swagerman	10	Prgmr	-	-	-	-
510	Limoges	10	Prgmr	-	-	-	-
520	Li	10	Prgmr	-	-	-	-

Dans ce cas, les colonnes ne sont pas toujours alimentées. Les valeurs NULL sont signalées par un tiret (-). Il faut que la liste des noms de colonnes suive le même ordre et que les données soient du même type que dans la clause VALUES. Si la liste des noms de colonnes est absente (c'est le cas du premier exemple), la liste des valeurs citées après VALUES doit être dans le même ordre que les colonnes de la table dans laquelle elles ont été insérées et le nombre de valeurs doit être égal au nombre de colonnes de la table.

Chaque valeur doit être compatible avec le type de données attribué à la colonne dans laquelle elle est insérée. Si une colonne accepte une valeur NULL, et si aucune valeur n'est précisée, la ligne insérée prend la valeur NULL.

L'exemple suivant insère la valeur NULL dans les colonnes YEARS, COMM et BIRTH_DATE, puisque ces colonnes n'ont pas reçu de valeur spécifique pour la ligne.

```

INSERT INTO PERS (ID, NAME, JOB, DEPT, SALARY)
VALUES (410, 'Perna', 'Sales', 20, 20000)

```

L'autre format de l'instruction INSERT est très pratique pour alimenter une table de valeurs provenant des lignes d'une autre table. Comme nous l'avons déjà mentionné, c'est l'instruction FULLSELECT qui est utilisée au lieu de la clause VALUES. Elle identifie les colonnes de lignes contenues dans d'autres tables et/ou vues.

L'exemple suivant sélectionne, dans la table STAFF, des données sur les membres du service 38 et les insère dans la table PERS :

```
INSERT INTO PERS (ID, NAME, DEPT, JOB, YEARS, SALARY)
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM STAFF
WHERE DEPT = 38
```

Après cette insertion, l'instruction SELECT ci-après produit un résultat identique à l'instruction FULLSELECT de l'instruction INSERT.

```
SELECT ID, NAME, DEPT, JOB, YEARS, SALARY
FROM PERS
WHERE DEPT = 38
```

Voici le résultat obtenu :

ID	NAME	DEPT	JOB	YEARS	SALARY
30	Marenghi	38	Mgr	5	17506.75
40	O'Brien	38	Sales	6	18006.00
60	Quigley	38	Sales	-	16808.30
120	Naughton	38	Clerk	-	12954.75
180	Abrahams	38	Clerk	3	12009.75

Modification de données

Pour modifier les données d'une table, utilisez l'instruction UPDATE. Cette instruction permet de modifier la valeur d'une ou de plusieurs colonnes, dans toutes les lignes répondant aux conditions de recherche de la clause WHERE.

L'exemple suivant met à jour les informations concernant l'employé dont l'ID est 410 :

```
UPDATE PERS
SET JOB='Prgmr', SALARY = SALARY + 300
WHERE ID = 410
```

La clause SET précise les colonnes à mettre à jour et fournit les valeurs.

La clause WHERE, facultative, précise les lignes à mettre à jour. Si elle est omise, le gestionnaire de bases de données remplace les anciennes valeurs par celles que vous fournissez dans toutes les lignes de la table ou de la vue.

L'exemple indique d'abord la table à mettre à jour (PERS), puis précise la condition à respecter pour qu'une ligne soit mise à jour. Les informations concernant l'employé 410 ont changé. Il occupe maintenant un poste de programmeur (Prgmr) et son salaire a augmenté de 300 F.

Pour modifier les données de plusieurs lignes, il est possible d'inclure une clause WHERE désignant les lignes à prendre en compte. Dans l'exemple suivant, le salaire de tous les vendeurs augmente de 15 % :

```
UPDATE PERS
  SET SALARY = SALARY * 1,15
  WHERE JOB = 'Sales'
```

Suppression de données

Utilisez l'instruction DELETE pour supprimer des lignes de données d'une table, sur la base d'une condition de recherche précisée dans la clause WHERE. L'exemple suivant supprime une ligne contenant la valeur 120 comme ID d'employé :

```
DELETE FROM PERS
  WHERE ID = 120
```

La clause WHERE, facultative, précise les lignes à supprimer. Si elle est omise, le gestionnaire de bases de données supprime toutes les lignes de la table ou de la vue.

Pour supprimer plusieurs lignes, utilisez l'instruction DELETE. L'exemple suivant supprime toutes les lignes contenant la valeur 20 comme numéro de service (DEPT) :

```
DELETE FROM PERS
  WHERE DEPT = 20
```

L'instruction DELETE ne supprime pas seulement les valeurs des colonnes d'une ligne, mais retire celle-ci en entier.

Pour supprimer la définition d'une table ainsi que son contenu, utilisez l'instruction DROP TABLE (voir le document *SQL Reference*).

Création de vues

Comme nous l'avons déjà décrit dans la section «Vues» à la page 4, une vue offre une autre possibilité de visualiser les données d'une ou de plusieurs tables. La création de vues permet de choisir les informations à afficher selon les utilisateurs auxquelles elles sont destinées. Le diagramme suivant montre les relations entre les vues et les tables.

Dans la figure 2 à la page 14, Vue_A limite l'accès aux colonnes AC1 et AC2 de la TABLE_A.

Vue_AB autorise l'accès à la colonne AC3 de la TABLE_A et à la colonne BC2 de la TABLE_B.

La création de la Vue_A restreint l'accès à la TABLE_A. La création de la Vue_AB limite l'accès à certaines colonnes des deux tables.

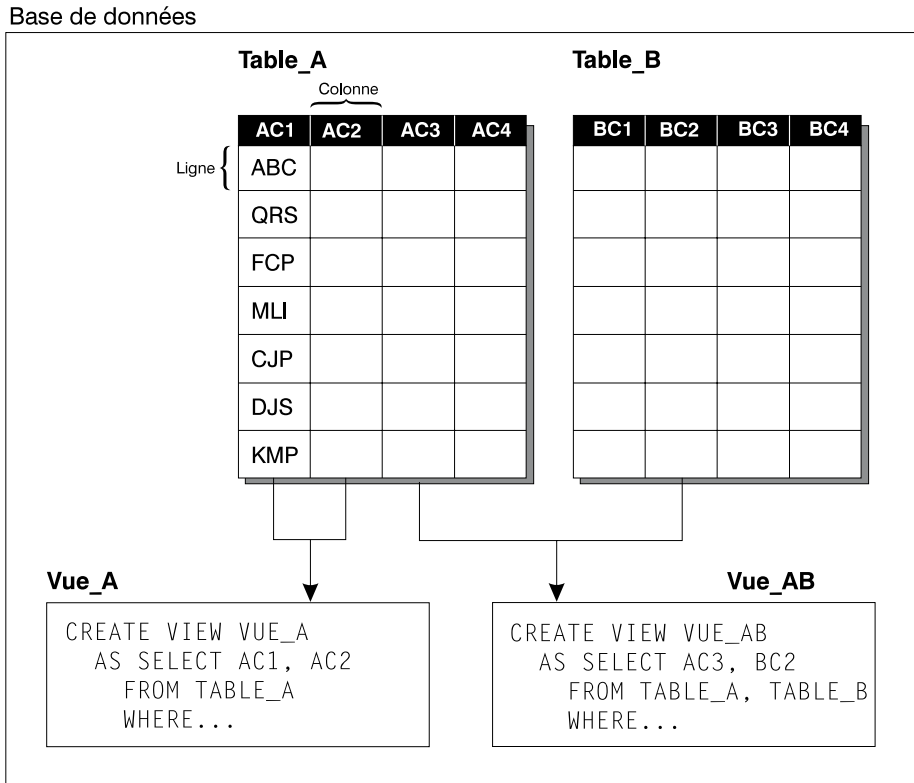


Figure 2. Relations entre les tables et les vues

L'instruction suivante crée une vue des employés du service 20 de la table STAFF qui ne sont pas des directeurs. Leur salaire et leur commission, présents dans la table de base, n'apparaissent pas dans la vue.

```

CREATE VIEW STAFF_ONLY
  AS SELECT ID, NAME, DEPT, JOB, YEARS
  FROM STAFF
  WHERE JOB <> 'Mgr' AND DEPT=20
  
```

Après la création de la vue, l'instruction suivante affiche son contenu :

```

SELECT *
FROM STAFF_ONLY
  
```


Cette instruction donne le résultat suivant :

ID	NAME	DEPT	JOB	YEARS
20	Pernal	20	Sales	8
80	James	20	Clerk	-
190	Sneider	20	Clerk	8

Nous pourrons ensuite utiliser les tables STAFF et ORG pour créer une vue qui liste les noms de tous les services accompagnés du nom de leur directeur. L'instruction suivante crée la vue ci-après :

```
CREATE VIEW DEPARTMENT_MGRS
AS SELECT NAME, DEPTNAME
FROM STAFF, ORG
WHERE MANAGER = ID
```

Lors de la création d'une vue, la clause WITH CHECK OPTION permet d'ajouter des contraintes aux insertions et aux mises à jour d'une table. Si cette clause est présente, le gestionnaire de bases de données vérifie que les mises à jour ou les insertions apportées à la vue sont conformes à la définition de la vue et refuse celles qui ne le sont pas. En l'absence de cette clause, les insertions et les mises à jour ne sont pas comparées à la définition de la vue. Pour plus de détails sur la clause WITH CHECK OPTION, reportez-vous à la description de l'instruction CREATE VIEW dans le document *SQL Reference*.

Utilisation de vues pour manipuler des données

Comme l'instruction SELECT, les instructions INSERT, DELETE et UPDATE sont appliquées à une vue, comme s'il s'agissait d'une table réelle. Les instructions manipulent les données de la ou des tables sous-jacentes. Ainsi, lors de l'accès suivant à la vue, cette dernière est comparée à la dernière table de base, ou aux plus récentes. En l'absence de la clause WITH CHECK OPTION, les données que vous modifiez en passant par une vue risquent de ne pas apparaître lors des accès suivants à cette dernière, puisqu'il est possible qu'elles ne correspondent plus à la définition d'origine de la vue.

Voici un exemple de mise à jour s'appliquant à la vue FIXED_INCOME :

```
CREATE VIEW FIXED_INCOME (LNAME, DEPART, JOBTITLE, NEWSALARY)
AS SELECT NAME, DEPT, JOB, SALARY
FROM PERS
WHERE JOB <> 'Sales' WITH CHECK OPTION

UPDATE FIXED_INCOME
SET NEWSALARY = SALARY * 1.10
WHERE LNAME = 'Li'
```

La mise à jour de la vue précédente équivaut à la mise à jour de la table de base PERS, sans l'option de vérification (la clause WITH CHECK OPTION est absente) :

```
UPDATE PERS
  SET SALARY = SALARY * 1.10
  WHERE NAME = 'Li'
  AND JOB <> 'Sales'
```

La vue étant créée avec la clause WITH CHECK OPTION pour la contrainte JOB <> 'Sales' dans la vue CREATE VIEW FIXED_INCOME, la mise à jour suivante n'est pas autorisée si Limoges devient vendeur :

```
UPDATE FIXED_INCOME
  SET JOBTITLE = 'Sales'
  WHERE LNAME = 'Limoges'
```

Les colonnes définies par des expressions telles que SALARY + COMM ou SALARY * 1,25 ne peuvent pas être mises à jour. Si vous définissez une vue contenant une ou plusieurs colonnes de ce type, le privilège UPDATE sur ces colonnes n'est pas accordé au propriétaire. Les instructions INSERT ne sont pas autorisées sur les vues contenant de telles colonnes. Par contre, les instructions DELETE sont autorisées.

Etudions une table PERS, dans laquelle aucune des colonnes n'est de type NOT NULL. Il est possible de passer par la vue FIXED_INCOME pour insérer des lignes dans la table PERS, même si la vue ne contient pas les colonnes ID, YEARS, COMM et BIRTHDATE de la table PERS sous-jacente. Les colonnes qui n'apparaissent pas dans la vue prennent la valeur NULL ou la valeur par défaut, selon le cas.

Cependant, dans la table PERS, la colonne ID est de type NOT NULL. Si vous passez par la vue FIXED_INCOME pour tenter d'insérer une ligne, le système essaie d'insérer des valeurs NULL dans toutes les colonnes de la table PERS «invisibles» dans la vue. Comme la colonne ID n'est pas incluse dans la vue et n'accepte pas les valeurs NULL, le système ne vous permet pas d'insérer une ligne en passant par la vue.

Pour connaître les règles et les restrictions s'appliquant à la modification des vues, reportez-vous à la description de l'instruction CREATE VIEW dans le document *SQL Reference*.

Chapitre 4. Utilisation d'instructions SQL pour accéder aux données

La présente section explique comment utiliser des instructions SQL pour se connecter à une base de données et extraire des données.

Les exemples présentent l'instruction à utiliser, suivie (dans la plupart des cas) des résultats qui s'affichent lorsque cette instruction est exécutée sur la base de données SAMPLE. Bien que les instructions apparaissent ici en majuscules, vous pouvez utiliser n'importe quelle combinaison de minuscules et de majuscules, sauf lorsqu'elles figurent entre apostrophes (') ou entre guillemets (").

La base de données SAMPLE, incluse dans DB2 Universal Database, se compose de plusieurs tables, dont vous trouverez la liste en «Annexe A. Tables de la base de données SAMPLE» à la page 77. Cette base de données SAMPLE peut être créée à l'aide de l'option "Premiers pas" ou à partir de la ligne de commande. Pour plus de détails, consultez le document *SQL Reference*.

Des modèles de bases de données supplémentaires ont été inclus dans DB2 Universal Database pour mettre en lumière les fonctionnalités de Data Warehouse Center et de OLAP Starter Kit. Les exemples du présent document utilisent uniquement la base de données générique appelée SAMPLE.

Selon la façon dont votre base de données a été installée, vous aurez peut-être à faire précéder les noms de table utilisés du nom du schéma et d'un point. Dans les exemples du présent document, le schéma par défaut a été appelé IDUTIL. C'est pourquoi vous pouvez utiliser le nom IDUTIL.ORG pour désigner la table ORG. Demandez à votre administrateur si ceci est nécessaire.

Les sujets suivants sont abordés :

- Connexion à une base de données
- Localisation des erreurs
- Sélection de colonnes et Sélection de lignes
- Classement des lignes et Elimination des lignes en double
- Ordre des opérations
- Utilisation d'expressions pour calculer des valeurs
- Dénomination des expressions
- Sélection de données dans plusieurs tables

- Utilisation d'une sous-requête
- Utilisation de fonctions
- Groupage

Connexion à une base de données

Vous devez vous connecter à la base de données avant de pouvoir l'interroger et la manipuler avec des instructions SQL. L'instruction CONNECT associe la connexion de la base de données à un nom d'utilisateur.

Par exemple, pour vous connecter à la base de données SAMPLE, tapez la commande suivante dans l'Interpréteur de commandes DB2 :

```
CONNECT TO SAMPLE USER IDUTIL USING MOT_DE_PASSE
```

(Veillez à attribuer à IDUTIL et à MOT_DE_PASSE des valeurs reconnues sur le système serveur.)

Dans cet exemple, USER a pour valeur IDUTIL et USING a pour valeur MOT_DE_PASSE.

Le message suivant indique que la connexion a été établie :

Informations de connexion à la base de données

Produit de gestion de bases de données	= DB2/NT 7.1.0
ID autorisation SQL	= IDUTIL
Alias de la base de données locale	= SAMPLE

Une fois connecté, vous pouvez commencer à manipuler la base de données. Pour plus de détails sur les connexions, reportez-vous à la description de l'instruction CONNECT dans le document *SQL Reference*.

Localisation des erreurs

Chaque fois que vous faites une faute de frappe dans l'un des exemples ou qu'une erreur survient pendant l'exécution d'une instruction SQL, le gestionnaire de bases de données renvoie un message d'erreur. Ce dernier se compose d'un identificateur de message, d'une brève explication et d'un code SQLSTATE.

Les codes SQLSTATE sont des codes d'erreur communs à toute la gamme de produits DB2. Ils respectent la norme ISO/ANSI SQL92.

Par exemple, si l'ID utilisateur ou le mot de passe avait été incorrect dans l'instruction CONNECT, le gestionnaire de bases de données aurait renvoyé l'identificateur de message SQL1403N et le code SQLSTATE 08004. Voici comment le message se présente :

```
SQL1403N Le nom d'utilisateur et/ou le mot de passe sont
          incorrects.  SQLSTATE=08004
```

Pour obtenir plus d'informations sur le message d'erreur, tapez un point d'interrogation (?), puis l'identificateur du message ou le code SQLSTATE :

```
? SQL1403N
OU
? SQL1403
OU
? 08004
```

Dans la description de l'erreur SQL1403N, l'avant-dernière ligne indique que le code SQLCODE a pour valeur -1403. SQLCODE est un code d'erreur spécifique au produit. Les identificateurs de message se terminant par N (Notification) ou C (Critique) représentent une erreur et leur code SQLCODE est négatif. Les identificateurs de message se terminant par W (ou avertissement) ont un code SQLCODE positif.

Sélection de colonnes

L'instruction SELECT permet de sélectionner des colonnes spécifiques dans une table. Dans l'instruction, indiquez une liste de noms de colonne, séparés par des virgules. Cette liste est appelée *liste de sélection*.

L'instruction suivante sélectionne les colonnes DEPTNAME et DEPTNUMB de la table ORG, dans la base de données SAMPLE :

```
SELECT DEPTNAME, DEPTNUMB
FROM ORG
```

L'instruction ci-dessus fournit le résultat suivant :

DEPTNAME	DEPTNUMB
-----	-----
Head Office	10
New England	15
Mid Atlantic	20
South Atlantic	38
Great Lakes	42
Plains	51
Pacific	66
Mountain	84

Un astérisque (*) permet de sélectionner toutes les colonnes de la table. L'exemple suivant recense toutes les colonnes et toutes les lignes de la table ORG :

```
SELECT *
FROM ORG
```

Cette instruction donne le résultat suivant :

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	Head Office	160	Corporate	New York
15	New England	50	Eastern	Boston
20	Mid Atlantic	10	Eastern	Washington
38	South Atlantic	30	Eastern	Atlanta
42	Great Lakes	100	Midwest	Chicago
51	Plains	140	Midwest	Dallas
66	Pacific	270	Western	San Francisco
84	Mountain	290	Western	Denver

Sélection de lignes

Pour préciser la ou les conditions que les lignes doivent respecter pour être sélectionnées dans la table, faites suivre l'instruction SELECT de la clause WHERE. Un critère de sélection de lignes dans une table est appelé *condition de recherche*.

Une condition de recherche se compose d'un ou de plusieurs *prédicats*. Un prédicat demande que la ligne respecte une condition vraie ou fausse (ou encore inconnue). Les prédicats de base suivants peuvent être utilisés dans une clause WHERE pour préciser les conditions à respecter :

Prédicat	Fonction
x = y	x est égal à y
x <> y	x est différent de y
x < y	x est inférieur à y
x > y	x est supérieur à y
x <= y	x est inférieur ou égal à y
x >= y	x est supérieur ou égal à y
IS NULL/IS NOT NULL	Recherche les valeurs NULL

Lorsque vous élaborez les conditions de recherche, veillez à ne demander des opérations arithmétiques que sur les données de type numérique et à ne comparer que des données de types compatibles. Par exemple, il est impossible de comparer des chaînes de texte à des nombres.

Si vous sélectionnez des lignes en fonction d'une valeur alphanumérique, pensez à placer cette dernière entre apostrophes (par exemple, WHERE JOB = 'Clerk') et à la taper telle qu'elle existe dans la base de données. Si la valeur est en minuscules dans la base de données et que vous la tapez en majuscules, aucune ligne n'est sélectionnée. Si vous sélectionnez des lignes en fonction d'une valeur numérique, ne mettez-pas celle-ci entre apostrophes (par exemple, WHERE DEPT = 20).

L'exemple suivant ne sélectionne, dans la table STAFF, que les lignes portant le numéro de service 20 :

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE DEPT = 20
```

Cette instruction donne le résultat suivant :

DEPT	NAME	JOB
20	Sanders	Mgr
20	Pernal	Sales
20	James	Clerk
20	Sneider	Clerk

L'exemple suivant utilise une fonction AND pour préciser plusieurs conditions. Il est possible de préciser autant de conditions que nécessaire. L'exemple sélectionne, dans la table STAFF, les employés de production (Clerk) du service 20 :

```
SELECT DEPT, NAME, JOB
FROM STAFF
WHERE JOB = 'Clerk'
AND DEPT = 20
```

Cette instruction donne le résultat suivant :

DEPT	NAME	JOB
20	James	Clerk
20	Sneider	Clerk

Si la colonne n'a pas de valeur par défaut et si vous ne précisez aucune valeur, elle prend la valeur NULL. C'est également le cas si vous avez spécifiquement demandé la valeur NULL. Les colonnes n'acceptent la valeur NULL que si leur définition le prévoit. La définition et la prise en charge des valeurs NULL sont abordées dans la section «Création de tables» à la page 9.

Pour rechercher une valeur NULL, utilisez les prédicats IS NULL et IS NOT NULL.

L'instruction suivante établit la liste des employés dont la commission est inconnue :

```
SELECT ID, NAME
FROM STAFF
WHERE COMM IS NULL
```

Cette instruction donne le résultat suivant :

ID	NAME
10	Sanders
30	Marenghi
50	Hanes
100	Plotz
140	Fraye
160	Molinare
210	Lu
240	Daniels
260	Jones
270	Lea
290	Quill

La valeur zéro n'est pas identique à la valeur NULL. L'instruction suivante sélectionne tous les employés de la table dont la commission est égale à zéro :

```
SELECT ID, NAME
FROM STAFF
WHERE COMM = 0
```

Comme il n'y a pas de valeur égale à zéro dans la colonne COMM du modèle de table, le résultat est vide.

L'exemple suivant sélectionne toutes les lignes de la table STAFF où la colonne YEARS a une valeur supérieure à 9 :

```
SELECT NAME, SALARY, YEARS
FROM STAFF
WHERE YEARS > 9
```

Cette instruction donne le résultat suivant :

NAME	SALARY	YEARS
Hanes	20659.80	10
Lu	20010.00	10
Jones	21234.00	12
Quill	19818.00	10
Graham	21000.00	13

Classement des lignes

Si vous désirez que les informations apparaissent dans un ordre spécifique, utilisez la clause ORDER BY, qui permet de trier les informations selon la valeur d'une ou de plusieurs colonnes.

L'instruction suivante affiche les employés du service 84, classés selon leur nombre d'années de présence dans la société :

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS
```

Cette instruction donne le résultat suivant :

NAME	JOB	YEARS
Davis	Sales	5
Gafney	Clerk	5
Edwards	Sales	7
Quill	Mgr	10

La clause ORDER BY doit être la dernière de l'instruction SELECT. Les colonnes citées dans cette clause peuvent être des expressions ou n'importe quelle colonne de la table. Il est inutile de préciser dans la liste de sélection les noms des colonnes de la clause ORDER BY.

Dans la clause ORDER BY, les mots clés ASC ou DESC indiquent explicitement et respectivement l'ordre de tri croissant ou décroissant. Si aucun des deux n'est présent, les lignes sont automatiquement classées par ordre croissant. L'instruction suivante affiche les employés du service 84 par ordre décroissant selon le nombre d'années de présence dans l'entreprise :

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY YEARS DESC
```

Cette instruction donne le résultat suivant :

NAME	JOB	YEARS
Quill	Mgr	10
Edwards	Sales	7
Davis	Sales	5
Gafney	Clerk	5

Les lignes peuvent être classées selon des données de type alphanumérique ou numérique. L'instruction suivante affiche les employés du service 84 par ordre alphabétique :

```
SELECT NAME, JOB, YEARS
FROM STAFF
WHERE DEPT = 84
ORDER BY NAME
```

Cette instruction donne le résultat suivant :

NAME	JOB	YEARS
Davis	Sales	5
Edwards	Sales	7
Gafney	Clerk	5
Quill	Mgr	10

Élimination des lignes en double

Dans l'instruction SELECT, il est possible de demander l'élimination des informations en double. Par exemple, dans la table STAFF, plusieurs numéros de la colonne DEPT et plusieurs descriptions de la colonne JOB sont cités plusieurs fois.

Pour éliminer les lignes en double, ajoutez, dans l'instruction SELECT, l'option DISTINCT. Dans l'exemple suivant, l'ajout de DISTINCT dans l'instruction évite les doublons dans la liste des postes :

```
SELECT DISTINCT DEPT, JOB
FROM STAFF
WHERE DEPT < 30
ORDER BY DEPT, JOB
```

Cette instruction donne le résultat suivant :

DEPT	JOB
10	Mgr
15	Clerk
15	Mgr
15	Sales
20	Clerk
20	Mgr
20	Sales

DISTINCT a éliminé toutes les lignes contenant des données en double de l'ensemble de colonnes précisé dans l'instruction SELECT.

Ordre des opérations

Il est important de prendre en compte l'ordre dans lequel les opérations se déroulent. Le résultat d'une clause sert de données en entrée dans la clause suivante, comme le montre la liste ci-dessous. La section «Dénomination des expressions» à la page 26, donne un exemple où l'ordre des opérations a son importance.

La séquence d'opérations suivantes ne reflète pas nécessairement la façon dont les opérations sont effectuées dans le code DB2. Cette explication est simplement une approche plus intuitive des requêtes. Voici la séquence des opérations :

1. Clause FROM
2. Clause WHERE
3. Clause GROUP BY
4. Clause HAVING
5. Clause SELECT
6. Clause ORDER BY

Utilisation d'expressions pour calculer des valeurs

Une *expression* est un calcul ou une fonction intégrée dans une instruction. L'instruction suivante calcule les salaires des employés du service 38 s'ils recevaient chacun une prime de 500 F :

```
SELECT DEPT, NAME, SALARY + 500
FROM STAFF
WHERE SERV = 38
ORDER BY 3
```

Voici le résultat obtenu :

DEPT	NAME	3
38	Abrahams	12509.75
38	Naughton	13454.75
38	Quigley	17308.30
38	Marenghi	18006.75
38	O'Brien	18506.00

A la place du titre de la troisième colonne apparaît un numéro. Il s'agit d'un numéro généré par le système, SALARY+500 n'étant pas un nom de colonne. Dans un exemple ultérieur, ce numéro est utilisé dans la clause ORDER BY pour faire référence à la troisième colonne. La section «Dénomination des expressions» à la page 26 indique comment donner aux expressions des noms plus significatifs.

Les expressions arithmétiques utilisent les opérateurs de base de l'addition (+), la soustraction (-), la multiplication (*) et la division (/).

Les opérateurs peuvent être utilisés avec divers types d'opérandes, tels que :

- Des noms de colonnes (par exemple, RATE * HOURS)
- Des constantes (par exemple, RATE * 1,07)
- Des fonctions scalaires (par exemple LENGTH(NAME) + 1)

Dénomination des expressions

Le fait d'attribuer un nom à une expression avec la clause AS, facultative, permet de se référer plus facilement à cette expression. Avec la clause AS, tout élément de la liste de sélection peut recevoir un nom.

L'instruction suivante affiche tous les employés dont le total du salaire et de la commission est inférieur à 13000 F. Le nom PAY a été attribué à l'expression SALARY + COMM :

```
SELECT NAME, JOB, SALARY + COMM AS PAY
FROM STAFF
WHERE (SALARY + COMM) < 13000
ORDER BY PAY
```

Cette instruction donne le résultat suivant :

NAME	JOB	PAY
Yamaguchi	Clerk	10581.50
Burke	Clerk	11043.50
Scoutten	Clerk	11592.80
Abrahams	Clerk	12246.25
Kermisch	Clerk	12368.60
Ngan	Clerk	12714.80

L'utilisation de la clause AS permet de faire référence à une colonne par son nom au lieu d'utiliser le numéro généré par le système dans la clause ORDER BY. Dans cet exemple, dans la clause WHERE, nous comparons (SALARY + COMM) à 13000, au lieu d'utiliser le nom PAY. C'est le résultat de l'ordre des opérations. Il y a évaluation de la clause WHERE avant de donner à (SALARY + COMM) le nom PAY, car la clause SELECT est exécutée après la clause WHERE. Par conséquent, PAY ne peut pas être utilisé dans le prédicat.

Sélection de données dans plusieurs tables

L'instruction SELECT peut servir à produire des états sur plusieurs tables. Cette opération est couramment appelée *jointure*. Par exemple, une jointure des tables STAFF et ORG permet d'obtenir une nouvelle table. Pour joindre deux tables, précisez, dans la clause SELECT, les colonnes à afficher, dans la

clause FROM, les noms des tables et dans la clause WHERE, la condition de recherche. La clause WHERE est facultative.

Dans l'exemple suivant, le résultat doit associer les noms de tous les directeurs à leur service. Les informations viennent de deux tables, les données sur les employés (table STAFF) et sur les services (table ORG) étant conservées séparément. La requête suivante sélectionne la colonne NAME dans la table STAFF et la colonne DEPTNAME dans la table ORG. La condition de recherche limite la sélection aux lignes contenant, dans la colonne MANAGER, la même valeur que la colonne ID :

```
SELECT DEPTNAME, NAME
FROM ORG, STAFF
WHERE MANAGER = ID
```

La figure 3 montre comment sont comparées des colonnes de deux tables différentes. Les valeurs encadrées indiquent qu'il y a correspondance et que, par conséquent, la condition de recherche est remplie.

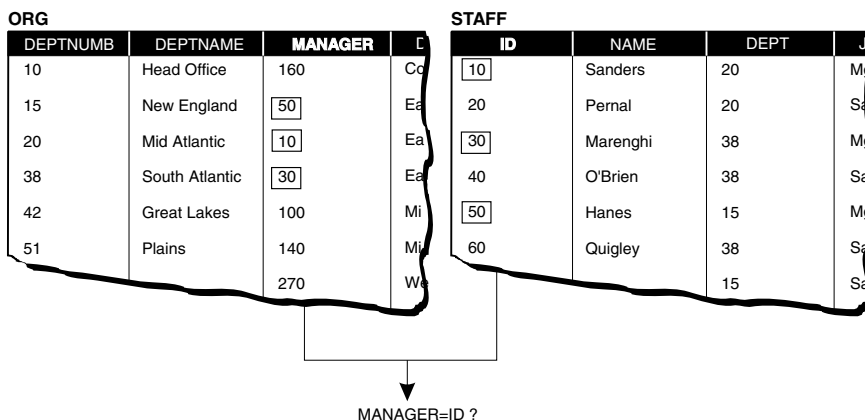


Figure 3. Sélection dans les tables STAFF et ORG

L'instruction SELECT donne le résultat suivant :

```
DEPTNAME      NAME
-----
Mid Atlantic   Sanders
South Atlantic Marenghi
New England    Hanes
Great Lakes    Plotz
Plains         Fraye
Head Office    Molinare
Pacific        Lea
Mountain       Quill
```

Le résultat établit la liste des directeurs et de leur service.

Utilisation d'une sous-requête

Des instructions SELECT supplémentaires peuvent être imbriquées dans une instruction SQL SELECT, dans la clause WHERE. Chaque nouvelle instruction SELECT lance une sous-requête.

Une sous-requête peut alors, à son tour, inclure une autre sous-requête dont le résultat est remplacé par la clause WHERE de la sous-requête d'origine. De plus, plusieurs conditions de recherche d'une clause WHERE peuvent inclure des sous-requêtes. La sous-requête peut faire référence à des tables et à des colonnes différentes de celles qu'utilise la requête principale.

L'instruction suivante sélectionne, dans la table ORG, la région et la ville de l'employé correspondant à l'ID 280 dans la table STAFF :

```
SELECT DIVISION, LOCATION
FROM ORG
WHERE DEPTNUMB = (SELECT DEPT
FROM STAFF
WHERE ID = 280)
```

Quand il traite une instruction, DB2 commence par déterminer le résultat de la sous-requête. Le résultat de cette instruction est 66, l'employé correspondant à l'ID 280 se trouvant dans le service 66. Ensuite, pour obtenir le résultat final, il prend dans la table ORG la ligne dont la colonne DEPTNUMB a la valeur 66. Voici le résultat obtenu :

```
DIVISION  LOCATION
-----  -
Western   San Francisco
```

Si vous utilisez une sous-requête, le gestionnaire de bases de données l'évalue et remplace directement la valeur résultant de la sous-requête dans la clause WHERE.

Pour plus de détails sur les sous-requêtes, consultez la section «Sous-requêtes corrélées» à la page 41.

Utilisation de fonctions

Vous trouverez ci-après une présentation rapide des fonctions utilisées dans les exemples du présent document. Une *fonction de base de données* est une relation entre un ensemble de valeurs fournies en entrée et une valeur résultat.

Les fonctions peuvent être *intégrées* ou *définies par l'utilisateur*. Vous trouverez dans DB2 Universal Database de nombreuses fonctions intégrées et définies par l'utilisateur et déjà installées.

Les fonctions intégrées se trouvent dans le schéma SYSIBM et les fonctions définies par l'utilisateur et déjà installées figurent dans le schéma SYSPFUN. SYSIBM et SYSPFUN sont des schémas réservés.

Les fonctions intégrées et définies par l'utilisateur et déjà installées ne suffisent pas à remplir tous les besoins de l'utilisateur. C'est pourquoi les développeurs ont quelquefois besoin d'ajouter leur propre ensemble de fonctions, spécifique de leurs applications. Il est possible de le faire à partir des fonctions définies par l'utilisateur, qui élargissent la portée de DB2 Universal Database par l'inclusion, par exemple, de fonctions scientifiques ou commerciales personnalisées. Vous trouverez plus de détails sur ce sujet à la section «Fonctions définies par l'utilisateur (UDF)» à la page 72.

Fonctions de colonne

Les *fonctions de colonne* donnent un résultat unique à partir d'un ensemble de valeurs présentes dans une colonne. Vous trouverez ci-dessous quelques exemples de fonctions de colonne. Le document *SQL Reference* contient la liste complète de ces fonctions de colonne.

AVG	Renvoie la somme des valeurs d'un ensemble, divisée par le nombre de valeurs de l'ensemble.
COUNT	Renvoie le nombre de lignes ou de valeurs, dans un ensemble de lignes ou de valeurs.
MAX	Renvoie la valeur la plus élevée d'un ensemble de valeurs.
MIN	Renvoie la valeur la plus basse d'un ensemble de valeurs.

L'instruction suivante sélectionne le salaire maximum de la table STAFF :

```
SELECT MAX(SALARY)
FROM STAFF
```

Cette instruction renvoie de la table STAFF la valeur 22959,20.

L'exemple suivant sélectionne les noms et les salaires des employés qui ont un revenu plus élevé que la moyenne des revenus constatés dans la société et un temps de présence inférieur au temps de présence moyen.

```
SELECT NAME, SALARY
FROM STAFF
WHERE SALARY > (SELECT AVG(SALARY) FROM STAFF)
AND YEARS < (SELECT AVG(YEARS) FROM STAFF)
```

Cette instruction donne le résultat suivant :

NAME	SALARY
-----	-----
Marengi	17506.75
Daniels	19260.25
Gonzales	16858.20

Dans la clause WHERE de l'exemple ci-dessus, la fonction de colonne est déclarée dans une *sous-requête* au lieu d'être implémentée directement (par exemple, WHERE SALARY > AVG(SALARY)). Les fonctions de colonne ne peuvent pas être déclarées dans la clause WHERE. Ceci est dû à l'ordre des opérations. On peut considérer que la clause WHERE est évaluée avant la clause SELECT. Par conséquent, lors de l'évaluation de la clause WHERE, la fonction de colonne n'a pas accès à l'ensemble de valeurs. Cet ensemble est sélectionné ultérieurement, avec la clause SELECT.

L'élément DISTINCT, utilisé dans l'argument d'une fonction de colonne, peut servir à éliminer les valeurs en double avant d'appliquer la fonction. Ainsi, COUNT(DISTINCT WORKDEPT) calcule le nombre de services différents.

Fonctions scalaires

Une *fonction scalaire* effectue une opération sur une valeur, pour renvoyer une autre valeur. Voici quelques exemples de fonctions scalaires fournies par DB2 Universal Database.

ABS	Renvoie la valeur absolue d'un nombre.
HEX	Renvoie la représentation hexadécimale d'une valeur.
LENGTH	Renvoie le nombre d'octets d'un argument (dans le cas d'une chaîne graphique, renvoie le nombre de caractères double octet).
YEAR	Extrait la partie année d'une valeur indiquant la date et/ou l'heure.

Le document *SQL Reference* contient une liste détaillée des fonctions scalaires, accompagnée de leur description.

L'instruction suivante renvoie les noms des services de la table ORG ainsi que la longueur de chacun des noms :

```
SELECT DEPTNAME, LENGTH(DEPTNAME)
FROM ORG
```

Cette instruction donne le résultat suivant :

```
DEPTNAME          2
-----
Head Office              11
New England              11
Mid Atlantic             12
South Atlantic           14
Great Lakes              11
Plains                   6
Pacific                  7
Mountain                 8
```


Remarque : Comme la clause AS n'a pas été utilisée pour donner un nom significatif à LENGTH(DEPTNAME), un numéro généré par le système apparaît en titre de la deuxième colonne.

Fonctions de table

Les *fonctions de table* renvoient les colonnes d'une table qui ressemblent à la table créée par une instruction CREATE TABLE.

Une fonction de table peut être utilisée uniquement dans la clause FROM d'une instruction SQL.

La seule fonction de table prise en charge dans DB2 Universal Database est SQLCACHE_SNAPSHOT.

SQLCACHE_SNAPSHOT

Renvoie les résultats d'une image instantanée de la mémoire cache d'une instruction SQL dynamique en tant que table.

Groupage

Pour l'analyse de données, DB2 Universal Database peut se baser sur certaines colonnes d'une table.

Il est possible de rassembler plusieurs lignes selon la structure de groupage définie par la clause GROUP BY. Dans sa forme la plus simple, un *groupe* est constitué de lignes, chacune ayant des valeurs identiques dans les colonnes "GROUP BY". Dans une clause SELECT, les noms des colonnes doivent correspondre à des colonnes de groupage ou à des fonctions de colonne. Ces dernières renvoient une valeur pour chaque groupe défini par la clause GROUP BY. Chaque groupe est représenté par une seule ligne dans l'ensemble des résultats. Dans l'exemple suivant, le résultat produit la liste des salaires les plus élevés par numéro de service :

```
SELECT DEPT, MAX(SALARY) AS MAXIMUM
FROM STAFF
GROUP BY DEPT
```

Cette instruction donne le résultat suivant :

DEPT	MAXIMUM
10	22959,20
15	20659,80
20	18357,50
38	18006,00
42	18352,80
51	21150,00
66	21000,00
84	19818,00

Notez que le salaire le plus élevé (MAX(SALARY)) est calculé par service (groupe défini par la clause GROUP BY), et non pour toute la société.

Utilisation d'une clause WHERE avec une clause GROUP BY

Une requête de groupage peut contenir une clause WHERE standard qui élimine les lignes ne répondant pas à un ou plusieurs critères avant de former les groupes et d'exécuter les fonctions de colonne. Il faut préciser la clause WHERE *avant* la clause GROUP BY. Par exemple :

```
SELECT WORKDEPT, EDLEVEL, MAX(SALARY) AS MAXIMUM
      FROM EMPLOYEE
 WHERE HIREDATE > '1979-01-01'
  GROUP BY WORKDEPT, EDLEVEL
  ORDER BY WORKDEPT, EDLEVEL
```

Voici le résultat obtenu :

WORKDEPT	EDLEVEL	MAXIMUM
D11	17	18270,00
D21	15	27380,00
D21	16	36170,00
D21	17	28760,00
E11	12	15340,00
E21	14	26150,00

Tous les noms de colonne précisés dans l'instruction SELECT sont également mentionnés dans la clause GROUP BY. L'absence de ces noms de colonne dans les deux clauses entraîne une erreur. La clause GROUP BY renvoie une ligne par combinaison unique de WORKDEPT et EDLEVEL.

Utilisation de la clause HAVING après la clause GROUP BY

Il est possible d'appliquer aux groupes une condition qualifiante pour que DB2 ne renvoie que les résultats concernant les groupes qui respectent cette condition. Pour ce faire, incluez une clause HAVING *après* la clause GROUP BY. Une clause HAVING peut contenir un ou plusieurs prédicats, reliés par des AND et des OR. Chaque prédicat compare une propriété du groupe (comme AVG(SALARY)) à l'un des éléments suivants :

- Une autre propriété du groupe

Par exemple :

```
HAVING AVG(SALARY) > 2 * MIN(SALARY)
```

- Une constante

Par exemple :

```
HAVING AVG(SALARY) > 20000
```

Par exemple, la requête suivante recherche le salaire le plus élevé et le salaire le moins élevé dans les services comprenant plus de 4 employés :

```
SELECT WORKDEPT, MAX(SALARY) AS MAXIMUM, MIN(SALARY) AS MINIMUM
FROM EMPLOYEE
GROUP BY WORKDEPT
HAVING COUNT(*) > 4
ORDER BY WORKDEPT
```

Cette instruction donne le résultat suivant :

WORKDEPT	MAXIMUM	MINIMUM
D11	32250,00	18270,00
D21	36170,00	17250,00
E11	29750,00	15340,00

Dans une requête, il est possible mais rare d'avoir une clause HAVING sans clause GROUP BY. Dans ce cas, DB2 traite la table en entier en tant que groupe. Comme la table est considérée comme un seul groupe, le résultat ne donnera, au plus, qu'une ligne de résultat. Si une condition HAVING est vraie sur l'ensemble de la table, le résultat sélectionné est renvoyé (il ne se compose que de fonctions de colonne) ; sinon, aucune ligne n'est renvoyée.

Chapitre 5. Expressions et sous-requêtes

DB2 permet d'exprimer les requêtes avec beaucoup de souplesse. Le présent chapitre décrit quelques-unes des principales méthodes dont vous disposez pour rédiger des requêtes complexes.

Les éléments suivants sont décrits en détail :

- Instructions FULLSELECT scalaires
- Spécifications de transtypage de données
- Expressions CASE
- Expressions de table
- Noms de corrélation

Instructions FULLSELECT scalaires

Une instruction FULLSELECT est un type de requête qui peut être utilisé dans les instructions SQL. Une instruction FULLSELECT scalaire est une instruction FULLSELECT qui renvoie une ligne contenant une seule valeur. Ces instructions permettent d'extraire des valeurs d'une base de données pour les utiliser dans une expression.

- L'exemple suivant permet d'obtenir la liste de l'ensemble du personnel dont le salaire est supérieur au salaire moyen de la société. L'instruction FULLSELECT scalaire dans la requête correspond à l'instruction SELECT entre parenthèses.

```
SELECT LASTNAME, FIRSTNAME
FROM EMPLOYEE
WHERE SALARY > (SELECT AVG(SALARY)
FROM EMPLOYEE)
```

- L'exemple suivant permet d'obtenir le salaire moyen des employés de la table STAFF et le salaire moyen des employés de la table EMPLOYEE.

```
SELECT AVG(SALARY)
AS "Average_Employee",
(SELECT AVG(SALARY) AS "Average_Staff" FROM STAFF)
FROM EMPLOYEE
```

Spécifications de transtypage de données

Vous aurez quelquefois à convertir des valeurs d'un type dans un autre. Une spécification de transtypage (CAST) permet, par exemple, de convertir une valeur numérique en chaîne de caractères. Pour convertir une valeur dans un type différent, utilisez la spécification CAST.

Elle permet aussi de tronquer une chaîne de caractères trop longue. La colonne RESUME est de type CLOB(5K). Pour n'afficher que les 370 premiers caractères au format ASCII de la fiche du candidat, à partir de la table EMP_RESUME, utilisez la requête suivante :

```
SELECT EMPNO, CAST(RESUME AS VARCHAR(370))
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'ascii'
```

Un avertissement est renvoyé pour les valeurs supérieures à 370 caractères, pour vous prévenir que des informations ont été perdues.

Il est possible de convertir des valeurs NULL en d'autres types de données, plus faciles à manipuler dans une requête. La section «Expressions de table communes» à la page 39, est un exemple de l'utilisation d'une spécification de transtypage (CAST) dans ce but.

Expressions CASE

Les expressions CASE peuvent être utilisées dans les instructions SQL pour faciliter la manipulation de la représentation des données d'une table. Il s'agit d'expressions conditionnelles puissantes de même nature que les instructions CASE de certains langages de programmation.

- Dans la table ORG, pour convertir les numéros de la colonne DEPTNAME en mots significatifs, entrez la requête suivante :

```
SELECT DEPTNAME,
CASE DEPTNUMB
WHEN 10 THEN 'Marketing'
WHEN 15 THEN 'Research'
WHEN 20 THEN 'Development'
WHEN 38 THEN 'Accounting'
ELSE 'Sales'
END AS FUNCTION
FROM ORG
```

Voici le résultat obtenu :

DEPTNAME	FUNCTION
Head Office	Marketing
New England	Research
Mid Atlantic	Development

South Atlantic	Accounting
Great Lakes	Sales
Plains	Sales
Pacific	Sales
Mountain	Sales

- Les expressions CASE peuvent également constituer une protection contre les opérations impossibles telles que la division par zéro. Dans l'exemple suivant, si l'employé n'a ni prime ni commission, la condition évite l'erreur en empêchant la division par zéro :

```

SELECT LASTNAME, WORKDEPT FROM EMPLOYEE
WHERE (CASE
      WHEN BONUS+COMM=0 THEN NULL
      ELSE SALARY/(BONUS+COMM)
      END ) > 10

```

- Vous pouvez utiliser une expression CASE pour obtenir le ratio de la somme des valeurs d'un sous-ensemble de valeurs sélectionnées dans une colonne par rapport à la somme de toutes les valeurs de cette colonne. Ce ratio peut figurer dans une instruction utilisant une expression CASE qui permet d'effectuer un seul passage dans les données. Sans expression CASE, deux passages au moins seraient nécessaires pour effectuer le même calcul.

Dans l'exemple suivant, le calcul du ratio des salaires du service 20 par rapport au total de tous les salaires est effectué à l'aide d'une expression CASE :

```

SELECT CAST(CAST (SUM(CASE
      WHEN DEPT = 20 THEN SALARY
      ELSE 0
      END) AS DECIMAL(7,2))/
      SUM(SALARY) AS DECIMAL (3,2))
FROM STAFF

```

Le résultat obtenu est 0,11. Les fonctions de transtypage (CAST) permettent de garder la précision du résultat.

- Vous pouvez utiliser une expression CASE pour évaluer une fonction simple, au lieu d'appeler la fonction elle-même, ce qui pourrait entraîner une surcharge du temps système. Par exemple :

```

CASE
  WHEN X<0 THEN -1
  WHEN X=0 THEN 0
  WHEN X>0 THEN 1
END

```

L'expression précédente donne le même résultat que la fonction SIGN définie par l'utilisateur dans le schéma SYSFUN.

Expressions de table

Une *expression de table* permet de définir une vue qui n'est nécessaire que dans une seule requête.

Les expressions de table sont provisoires. Elles ne sont valables que pour la durée de l'instruction SQL considérée ; elles ne peuvent pas être partagées comme des vues, mais leur utilisation est cependant plus souple que celle des vues.

Vous trouverez ci-dessous des exemples d'utilisation, dans des requêtes, d'expressions de table, imbriquées ou non imbriquées.

Expressions de table imbriquées

Une expression de table imbriquée est une vue temporaire dont la définition est *imbriquée* (définie directement) dans la clause FROM de la requête principale.

La requête ci-après utilise une expression de table imbriquée pour générer une réponse à la question suivante : Quel est le salaire total moyen, ainsi que le niveau d'études et l'ancienneté, des employés dont le niveau d'études est supérieur à 16 :

```
SELECT EDLEVEL, HIREYEAR, DECIMAL(AVG(TOTAL_PAY), 7,2)
FROM (SELECT EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
            SALARY+BONUS+COMM AS TOTAL_PAY
      FROM EMPLOYEE
      WHERE EDLEVEL > 16 ) AS PAY_LEVEL
GROUP BY EDLEVEL, HIREYEAR
ORDER BY EDLEVEL, HIREYEAR
```

Voici le résultat obtenu :

EDLEVEL	HIREYEAR	3
17	1967	28850,00
17	1973	23547,00
17	1977	24430,00
17	1979	25896,50
18	1965	57970,00
18	1968	32827,00
18	1973	45350,00
18	1976	31294,00
19	1958	51120,00
20	1975	42110,00

Cette requête nécessite l'utilisation d'une expression de table imbriquée pour extraire l'année d'embauche de la colonne HIREDATE. Cette année sera ensuite utilisée dans la clause GROUP BY. Vous pouvez choisir de ne pas

générer l'expression en tant que vue, si vous avez l'intention d'exécuter les mêmes requêtes à l'aide de valeurs différentes de EDLEVEL.

L'exemple utilise la fonction scalaire intégrée DECIMAL qui renvoie la représentation décimale d'un nombre ou d'une chaîne de caractères. Pour plus de détails sur les fonctions, consultez le document *SQL Reference*.

Expressions de table communes

Une *expression de table commune* est une expression de table que vous créez parce que vous avez l'intention de l'utiliser plusieurs fois dans une requête complexe. Définissez-la et attribuez-lui un nom dès le début de la requête, à l'aide d'une clause WITH. Les utilisations répétées d'une expression de table commune se serviront à chaque fois du même ensemble de résultats. Par comparaison, si vous utilisiez une expression de table imbriquée ou une vue imbriquée, l'ensemble de résultats serait généré à chaque fois, avec des résultats éventuellement différents.

L'exemple suivant permet de répertorier tous les employés de la société dont le niveau d'études est supérieur à 16, dont le salaire moyen est inférieur à celui des employés de même niveau d'études embauchés la même année. Les éléments de la requête sont décrits de façon plus détaillée à la suite de la requête.

1

```
WITH
  PAYLEVEL AS
    (SELECT EMPNO, EDLEVEL, YEAR(HIREDATE) AS HIREYEAR,
     SALARY+BONUS+COMM AS TOTAL_PAY
     FROM EMPLOYEE
     WHERE EDLEVEL > 16),
```

2

```
  PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS
    (SELECT EDLEVEL, HIREYEAR, AVG(TOTAL_PAY)
     FROM PAYLEVEL
     GROUP BY EDLEVEL, HIREYEAR)
```

3

```
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, DECIMAL(AVG_TOTAL_PAY,7,2)
FROM PAYLEVEL, PAYBYED
WHERE EDLEVEL = EDUC_LEVEL
AND HIREYEAR = YEAR_OF_HIRE
AND TOTAL_PAY < AVG_TOTAL_PAY
```

1

Il s'agit d'une expression de table commune appelée PAYLEVEL. Cette table résultat comporte le matricule, l'année d'embauche et le salaire

total de l'employé ainsi que son niveau d'études. Elle ne contient que des lignes concernant des employés dont le niveau d'études est supérieur à 16.

- 2** Il s'agit d'une expression de table commune appelée PAYBYED (pour paie selon études). Elle utilise la table PAYLEVEL générée dans l'expression de table commune précédente pour obtenir le niveau d'études, l'année d'embauche et le salaire moyen des employés, par niveau d'études, embauchés la même année. Les noms des colonnes renvoyées par cette table sont différents (par exemple, EDUC_LEVEL) des noms de colonnes utilisés dans la liste de sélection. Elle produit un ensemble de résultats appelé PAYBYED qui est identique au résultat obtenu par l'exemple d'expression de Table imbriquée.
- 3** Voici enfin la requête qui donne le résultat souhaité. Les deux tables (PAYLEVEL, PAYBYED) sont jointes pour déterminer les employés dont le salaire total est inférieur au salaire moyen des employés embauchés la même année qu'eux. Notez que PAYBYED est basé sur PAYLEVEL. Ainsi, la requête accède deux fois à PAYLEVEL dans l'ensemble de l'instruction. Les deux fois, l'évaluation de la requête utilise le même ensemble de lignes.

Voici le résultat obtenu :

EMPNO	EDLEVEL	YEAR_OF_HIRE	TOTAL_PAY	5
000210	17	1979	20132,00	25896,50

Noms de corrélation

Un *nom de corrélation* est un identificateur servant à distinguer différentes utilisations d'un même objet. Il peut être défini dans la clause FROM d'une requête et dans la première clause d'une instruction UPDATE ou DELETE. Il peut être associé à une table, à une vue ou à une expression de table imbriquée, mais uniquement dans le contexte dans lequel il a été défini.

Par exemple, la clause FROM STAFF S, ORG O définit que S et O sont les noms de corrélation de STAFF et ORG.

```
SELECT NAME, DEPTNAME
FROM STAFF S, ORG O
WHERE O.MANAGER = S.ID
```

Une fois défini, *seul* le nom de corrélation peut être utilisé pour qualifier l'objet. Par exemple, si, dans l'exemple ci-dessus, nous avions déclaré ORG.MANAGER=STAFF.ID, l'instruction aurait échoué.

Il est aussi possible d'utiliser un nom de corrélation comme abréviation, pour se référer à un objet de base de données. Le fait de taper S uniquement est plus rapide et plus facile que d'entrer STAFF.

L'utilisation de noms de corrélation permet de **dupliquer** un objet. Ceci s'avère utile pour comparer les entrées d'une table avec la table elle-même. Dans l'exemple ci-dessous, la table EMPLOYEE est comparée à l'une de ses instances, pour rechercher les directeurs de tous les employés. Il affiche les noms des employés autres que des concepteurs, le nom de leur directeur et le numéro de leur service.

```

SELECT E2.FIRSTNME, E2.LASTNAME, E2.JOB, E1.FIRSTNME AS MGR_FIRSTNAME,
       E1.LASTNAME AS MGR_LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1, EMPLOYEE E2
WHERE E1.WORKDEPT = E2.WORKDEPT
      AND E1.JOB = 'MANAGER'
      AND E2.JOB <> 'MANAGER'
      AND E2.JOB <> 'DESIGNER'

```

Cette instruction donne le résultat suivant :

FIRSTNME	LASTNAME	JOB	MGR_FIRSTNAME	MGR_LASTNAME	WORKDEPT
DOLORES	QUINTANA	ANALYST	SALLY	KWAN	C01
HEATHER	NICHOLLS	ANALYST	SALLY	KWAN	C01
JAMES	JEFFERSON	CLERK	EVA	PULASKI	D21
SALVATORE	MARINO	CLERK	EVA	PULASKI	D21
DANIEL	SMITH	CLERK	EVA	PULASKI	D21
SYBIL	JOHNSON	CLERK	EVA	PULASKI	D21
MARIA	PEREZ	CLERK	EVA	PULASKI	D21
ETHEL	SCHNEIDER	OPERATOR	EILEEN	HENDERSON	E11
JOHN	PARKER	OPERATOR	EILEEN	HENDERSON	E11
PHILIP	SMITH	OPERATOR	EILEEN	HENDERSON	E11
MAUDE	SETRIGHT	OPERATOR	EILEEN	HENDERSON	E11
RAMLAL	MEHTA	FIELDREP	THEODORE	SPENSER	E21
WING	LEE	FIELDREP	THEODORE	SPENSER	E21
JASON	GOUNOT	FIELDREP	THEODORE	SPENSER	E21

Sous-requêtes corrélées

Une sous-requête autorisée à se référer à l'une quelconque des tables déjà mentionnées est appelée *sous-requête corrélée*. Il est également possible de dire que la sous-requête contient une *référence corrélée* à une table utilisée dans la requête principale.

Dans l'exemple suivant, une sous-requête non corrélée demande la liste des matricules et des noms des employés du service 'A00' dont le salaire est supérieur au salaire moyen du service.

```

SELECT EMPNO, LASTNAME
FROM EMPLOYEE
WHERE WORKDEPT = 'A00'
AND SALARY > (SELECT AVG(SALARY)
              FROM EMPLOYEE
              WHERE WORKDEPT = 'A00')

```

Cette instruction donne le résultat suivant :

```

EMPNO  LASTNAME
-----
000010 HAAS
000110 LUCCHESI

```

Pour connaître le salaire moyen par service, il faut évaluer la sous-requête une fois par service. L'aptitude de SQL à établir des corrélations permet de le faire : il suffit d'écrire une sous-requête à exécuter de manière répétitive, une fois par ligne de la table identifiée dans la requête de niveau supérieur.

L'exemple suivant recense, via une sous-requête corrélée, tous les employés dont le salaire est supérieur au salaire moyen en vigueur dans leur service :

```

SELECT E1.EMPNO, E1.LASTNAME, E1.WORKDEPT
FROM EMPLOYEE E1
WHERE SALARY > (SELECT AVG(SALARY)
               FROM EMPLOYEE E2
               WHERE E2.WORKDEPT = E1.WORKDEPT)
ORDER BY E1.WORKDEPT

```

Dans cette requête, la sous-requête est évaluée une fois par service. Voici le résultat obtenu :

```

EMPNO  LASTNAME          WORKDEPT
-----
000010 HAAS              A00
000110 LUCCHESI        A00
000030 KWAN           C01
000060 STERN           D11
000150 ADAMSON         D11
000170 YOSHIMURA       D11
000200 BROWN              D11
000220 LUTZ                 D11
000070 PULASKI            D21
000240 MARINO              D21
000270 PEREZ                D21
000090 HENDERSON           E11
000280 SCHNEIDER           E11
000100 SPENSER             E21
000330 LEE                  E21
000340 GOUNOT              E21

```

Pour écrire une requête avec une sous-requête corrélée, utilisez le format de base d'une requête externe ordinaire avec sous-requête. Cependant, dans la clause FROM de la requête externe, juste après le nom de la table, placez un nom de corrélation. Ensuite, la sous-requête peut contenir des références qualifiées par le nom de corrélation. Par exemple, si E1 est un nom de corrélation, E1.WORKDEPT désigne la valeur WORKDEPT de la ligne en cours, dans la table de la requête externe. La sous-requête est ré-évaluée à chaque ligne de la table de la requête externe.

Lorsque vous utilisez une sous-requête corrélée, le système fait le travail pour vous. Ainsi, vous avez moins de code à écrire dans l'application.

DB2 autorise les références corrélées non qualifiées. Par exemple, une colonne de la table EMPLOYEE est appelée LASTNAME et une colonne de la table SALES est appelée SALES_PERSON, mais aucune colonne n'est appelée LASTNAME.

```
SELECT LASTNAME, FIRSTNAME, COMM
FROM EMPLOYEE
WHERE 3 > (SELECT AVG(SALES)
FROM SALES
WHERE LASTNAME = SALES_PERSON)
```

Dans cet exemple, le système cherche une colonne LASTNAME dans la clause FROM la plus imbriquée. Comme il n'en trouve pas, il recherche dans la clause FROM de niveau immédiatement supérieur (qui, dans ce cas, est la clause FROM externe). Bien que cela ne soit pas toujours nécessaire, il est recommandé de qualifier des références corrélées pour rendre la requête plus lisible et obtenir le résultat attendu.

Implémentation d'une sous-requête corrélée

Quand utiliser une sous-requête corrélée ? Quelquefois, l'utilisation d'une fonction de colonne relève de l'énigme.

Par exemple, vous désirez obtenir la liste des employés ayant un niveau d'études supérieur au niveau d'études moyen en vigueur dans leur service.

Il faut d'abord déterminer les éléments de la liste de sélection. Le but de la requête est d'obtenir une «liste des employés». Ceci implique que LASTNAME de la table EMPLOYEE devrait être suffisant pour identifier les employés sans risque de doublon. La liste doit énumérer les employés en fonction de leur niveau d'études (EDLEVEL) et du service (WORKDEPT) auquel ils appartiennent. Bien que le but de la requête ne soit pas l'affichage des colonnes, le fait de les inclure dans la liste de sélection aidera à illustrer le résultat. Il devient possible de construire une partie de la requête :

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE
```

Ensuite, il faut prévoir une condition de recherche (clause WHERE). Voici la condition déclarée par l'instruction : «...qui ont un niveau d'études supérieur au niveau d'études moyen des employés de ce service». Ceci signifie qu'il faut calculer, pour chaque employé, le niveau d'études moyen du service auquel il appartient. Cette instruction illustre le cas où une sous-requête corrélée conviendrait très bien. Sur chaque ligne est calculée une propriété inconnue (en l'occurrence, le niveau d'études moyen du service auquel l'employé appartient). Il faut donner à la table EMPLOYEE un nom de corrélation :

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
```

La sous-requête nécessaire est simple. Elle calcule le niveau d'études moyen par service. Voici l'instruction SQL complète :

```
SELECT LASTNAME, WORKDEPT, EDLEVEL
FROM EMPLOYEE E1
WHERE EDLEVEL > (SELECT AVG(EDLEVEL)
FROM EMPLOYEE E2
WHERE E2.WORKDEPT = E1.WORKDEPT)
```

Voici le résultat obtenu :

LASTNAME	WORKDEPT	EDLEVEL
HAAS	A00	18
KWAN	C01	20
PULASKI	D21	16
HENDERSON	E11	16
LUCCHESSI	A00	19
PIANKA	D11	17
SCOUTTEN	D11	17
JONES	D11	17
LUTZ	D11	18
MARINO	D21	17
JOHNSON	D21	16
SCHNEIDER	E11	17
MEHTA	E21	16
GOUNOT	E21	16

Vous voulez obtenir les noms, et non les numéros, des services auxquels les employés appartiennent. Les informations à utiliser (DEPTNAME) se trouvent dans une autre table (DEPARTMENT). La requête externe définissant une variable de corrélation peut également être une requête de jointure (voir la section «Sélection de données dans plusieurs tables» à la page 26, pour plus de détails).

Lorsque vous utilisez des jointures dans une requête externe, dans la clause FROM, indiquez les tables à joindre et placez le nom de corrélation à côté du nom de table approprié.

Pour que la requête recherche le nom du service, et non son numéro, remplacez WORKDEPT par DEPTNAME dans la liste de sélection. La clause FROM doit également inclure la table DEPARTMENT et la clause WHERE doit exprimer la condition de jointure appropriée.

Voici la requête modifiée :

```

SELECT LASTNAME, DEPTNAME, EDLEVEL
  FROM EMPLOYEE E1, DEPARTMENT
  WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
  AND EDLEVEL > (SELECT AVG(EDLEVEL)
                  FROM EMPLOYEE E2
                  WHERE E2.WORKDEPT = E1.WORKDEPT)

```

Cette instruction donne le résultat suivant :

LASTNAME	DEPTNAME	EDLEVEL
HAAS	SPIFFY COMPUTER SERVICE DIV.	18
LUCCHESSI	SPIFFY COMPUTER SERVICE DIV.	19
KWAN	INFORMATION CENTER	20
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17
MEHTA	SOFTWARE SUPPORT	16
GOUNOT	SOFTWARE SUPPORT	16

Les exemples ci-dessous montrent que le nom de corrélation utilisé dans une sous-requête doit être défini dans la clause FROM d'une requête contenant la sous-requête corrélée. Cependant, ceci peut impliquer plusieurs niveaux d'imbrication.

Si certains services ont un effectif restreint, le niveau d'études moyen obtenu risque de prêter à confusion. Pour que la comparaison entre le niveau d'études moyen d'un service et celui d'un employé donne un résultat significatif, vous pouvez préciser qu'il faut au moins cinq employés par service. Il nous faut alors recenser les employés dont le niveau d'études est supérieur à la moyenne de leur service, et ne prendre en compte que les services ayant un effectif supérieur ou égal à cinq employés.

Il faut prévoir une autre sous-requête afin de déterminer le nombre total d'employés du service auquel appartient chaque employé indiqué dans la requête externe :

```
SELECT COUNT(*)
  FROM EMPLOYEE E3
 WHERE E3.WORKDEPT = E1.WORKDEPT
```

La moyenne ne sera calculée que si le résultat est égal ou supérieur à 5 :

```
SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <= (SELECT COUNT(*)
           FROM EMPLOYEE E3
           WHERE E3.WORKDEPT = E1.WORKDEPT)
```

Enfin, seuls les employés ayant un niveau d'études supérieur à la moyenne du service seront inclus :

```
SELECT LASTNAME, DEPTNAME, EDLEVEL
  FROM EMPLOYEE E1, DEPARTMENT
 WHERE E1.WORKDEPT = DEPARTMENT.DEPTNO
 AND EDLEVEL >
 (SELECT AVG(EDLEVEL)
  FROM EMPLOYEE E2
 WHERE E2.WORKDEPT = E1.WORKDEPT
 AND 5 <=
 (SELECT COUNT(*)
  FROM EMPLOYEE E3
 WHERE E3.WORKDEPT = E1.WORKDEPT))
```

Cette instruction donne le résultat suivant :

LASTNAME	DEPTNAME	EDLEVEL
PIANKA	MANUFACTURING SYSTEMS	17
SCOUTTEN	MANUFACTURING SYSTEMS	17
JONES	MANUFACTURING SYSTEMS	17
LUTZ	MANUFACTURING SYSTEMS	18
PULASKI	ADMINISTRATION SYSTEMS	16
MARINO	ADMINISTRATION SYSTEMS	17
JOHNSON	ADMINISTRATION SYSTEMS	16
HENDERSON	OPERATIONS	16
SCHNEIDER	OPERATIONS	17

Chapitre 6. Utilisation d'opérateurs et de prédicats dans des requêtes

Dans DB2 Universal Database, il est possible d'associer des requêtes à différents *opérateurs ensemblistes* et de construire des instructions conditionnelles complexes avec des *prédicats quantifiés*.

Le présent chapitre explique comment :

- Associer des tables différentes à l'aide des opérateurs ensemblistes UNION, EXCEPT et INTERSECT.
- Mettre en place des conditions complexes, à utiliser dans des requêtes à prédicats quantifiés. Vous trouverez une brève présentation des prédicats à la section «Sélection de lignes» à la page 20.

Association de requêtes à l'aide d'opérateurs ensemblistes

Les opérateurs ensemblistes UNION, EXCEPT et INTERSECT permettent de fusionner dans une même requête plusieurs requêtes externes. Toutes les requêtes réunies par ces opérateurs ensemblistes sont exécutées et leurs résultats respectifs sont associés. Le résultat dépend de l'opérateur utilisé.

Opérateur UNION

Pour dériver une table résultat, l'opérateur UNION associe deux autres tables résultat (par exemple, TABLE1 et TABLE2) et élimine de ces tables les lignes en double. Lorsque ALL est utilisé avec UNION (UNION ALL), les lignes en double ne sont pas éliminées. Dans un cas comme dans l'autre, toutes les lignes de la table dérivée proviennent soit de TABLE1, soit de TABLE2.

Dans l'exemple suivant, la requête utilise l'opérateur UNION pour renvoyer les noms de toutes les personnes ayant un salaire supérieur à 21 000 F, ou qui ont des responsabilités de directeur et moins de 8 ans de présence dans la société :

1

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000  
UNION
```

2

```
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8  
ORDER BY ID
```

Voici les résultats des différentes requêtes :

1

ID	NAME
140	Fraye
160	Molinare
260	Jones

2

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels

Le gestionnaire de bases de données associe les résultats des deux requêtes, élimine les doublons et renvoie le résultat final trié par ordre croissant.

ID	NAME
10	Sanders
30	Marenghi
100	Plotz
140	Fraye
160	Molinare
240	Daniels
260	Jones

Si, dans une requête, vous devez utiliser la clause `ORDER BY` avec un opérateur ensembliste, écrivez-la après la dernière requête. Le système effectue le classement de l'ensemble des réponses obtenues.

Si le nom d'une colonne diffère entre les deux tables, la table associant tous les résultats ne trouve pas la correspondance. Les colonnes sont alors numérotées dans l'ordre où elles apparaissent. Dans ce cas, pour que les résultats soient classés, il faut préciser les numéros de colonne dans une clause `ORDER BY`.

Opérateur EXCEPT

Pour dériver une table résultat, l'opérateur `EXCEPT` inclut toutes les lignes se trouvant dans `TABLE1` mais non dans `TABLE2` et il élimine les lignes en double. Si `ALL` est utilisé avec `EXCEPT` (`EXCEPT ALL`), les lignes en double ne sont pas éliminées.

Dans l'exemple suivant, la requête utilise l'opérateur EXCEPT pour renvoyer les noms de toutes les personnes gagnant plus de 21000 F, mais n'ayant pas le titre de directeur et justifiant de 8 ans de présence, ou plus, dans la société :

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
EXCEPT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

Les résultats des différentes requêtes se trouvent dans la section réservée à l'opérateur UNION. L'instruction ci-dessus donne le résultat suivant :

```
ID      NAME
-----
260 Jones
```

Opérateur INTERSECT

Pour dériver une table résultat, l'opérateur INTERSECT n'inclut que les lignes existant à la fois dans TABLE1 et TABLE2 et élimine les lignes en double. Si ALL est utilisé avec INTERSECT (INTERSECT ALL), les lignes en double ne sont pas éliminées.

Dans l'exemple suivant, la requête utilise l'opérateur INTERSECT pour renvoyer les noms et les ID de toutes les personnes ayant un salaire supérieur à 21000 F, des responsabilités de directeur et moins de 8 ans de présence dans la société :

```
SELECT ID, NAME FROM STAFF WHERE SALARY > 21000
INTERSECT
SELECT ID, NAME FROM STAFF WHERE JOB='Mgr' AND YEARS < 8
```

Les résultats des différentes requêtes se trouvent dans la section réservée à l'opérateur UNION. Voici le résultat de deux requêtes sur lesquelles l'opérateur INTERSECT a été appliqué :

```
ID      NAME
-----
140 Fraye
160 Molinare
```

Lorsque vous utilisez les opérateurs UNION, EXCEPT et INTERSECT, gardez à l'esprit que :

- Tous les éléments des listes de sélection des requêtes utilisant ces opérateurs doivent être compatibles. Pour plus d'informations, voir la table des compatibilités entre types de données dans le document *SQL Reference*.
- Si elle est utilisée, la clause ORDER BY doit être placée après la dernière requête faisant appel à un opérateur ensembliste. Un nom de colonne ne peut être utilisé dans la clause ORDER BY que si le nom de colonne est identique pour les éléments correspondants dans la liste de sélection des requêtes pour chaque opérateur.

- Les opérations impliquant des colonnes de même type et de même longueur produisent une colonne de ce type et de cette longueur. Le document *SQL Reference* énonce les règles respectées pour attribuer un type de données aux résultats des opérateurs ensemblistes UNION, EXCEPT et INTERSECT.

Prédicats

Les prédicats permettent de créer des conditions telles que seules les lignes qui les respectent seront traitées. Vous trouverez une description des prédicats à la section «Sélection de lignes» à la page 20. La présente section décrit les prédicats IN, BETWEEN, LIKE, EXISTS et les prédicats quantifiés.

Utilisation du prédicat IN

Utilisez un prédicat IN pour comparer une valeur à plusieurs autres. Par exemple :

```
SELECT NAME
FROM STAFF
WHERE DEPT IN (20, 15)
```

Cet exemple équivaut à :

```
SELECT NAME
FROM STAFF
WHERE DEPT = 20 OR DEPT = 15
```

Utilisez les prédicats IN et NOT IN quand une sous-requête renvoie un ensemble de valeurs. Par exemple, la requête suivante permet de recenser les noms de tous les employés responsables des projets MA2100 et OP2012 :

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE EMPNO IN
  (SELECT RESPEMP
   FROM PROJECT
   WHERE PROJNO = 'MA2100'
   OR PROJNO = 'OP2012')
```

La sous-requête n'est évaluée qu'une fois, et le résultat se place directement dans la requête externe, sous forme de liste. Par exemple, la sous-requête ci-dessous sélectionne les employés ayant les matricules 10 et 330.

L'évaluation de la requête externe donne le même résultat que si elle contenait la clause WHERE suivante :

```
WHERE EMPNO IN (10, 330)
```

La liste renvoyée par la sous-requête peut ne contenir aucune valeur ou en comporter une ou plusieurs.

Utilisation du prédicat BETWEEN

Utilisez un prédicat BETWEEN pour comparer une valeur à une plage de valeurs inclusives (nommées dans le prédicat BETWEEN).

L'exemple suivant recherche les noms des employés gagnant entre 10000 et 20000 F :

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY BETWEEN 10000 AND 20000
```

Ceci équivaut à :

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY >= 10000 AND SALARY <= 20000
```

L'exemple suivant recherche les noms des employés gagnant moins de 10000 ou plus de 20000 F :

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE SALARY NOT BETWEEN 10000 AND 20000
```

Utilisation du prédicat LIKE

Utilisez le prédicat LIKE pour rechercher les chaînes correspondant à certains modèles. Le modèle est précisé à l'aide du signe % et du trait de soulignement.

- Le trait de soulignement () représente n'importe quel caractère individuel.
- Le symbole du pourcentage (%) représente une chaîne de zéro ou plusieurs caractères.
- Tous les autres caractères se représentent eux-mêmes.

L'exemple suivant sélectionne les noms des employés comportant 7 caractères et commençant par la lettre 'S' :

```
SELECT NAME
FROM STAFF
WHERE NAME LIKE 'S _ _ _ _ _ _ _'
```

L'exemple suivant sélectionne les noms des employés commençant par une lettre autre que 'S' :

```
SELECT NAME
FROM STAFF
WHERE NAME NOT LIKE 'S%'
```

Utilisation du prédicat EXISTS

Une sous-requête peut être utilisée pour vérifier l'existence d'une ligne répondant à une condition. Dans ce cas, la sous-requête est liée à la requête externe par le prédicat EXISTS ou NOT EXISTS.

Si vous liez une sous-requête à une requête externe par le prédicat EXISTS, cette sous-requête ne renvoie pas de valeur. Le prédicat EXISTS est vrai si l'ensemble des réponses de la sous-requête contient une ligne ou plus, et il est faux s'il n'en contient aucune.

Le prédicat EXISTS est souvent utilisé avec les sous-requêtes corrélées. L'exemple ci-dessous recense les services ne possédant aucune entrée dans la table PROJECT :

```
SELECT DEPTNO, DEPTNAME
FROM DEPARTMENT X
WHERE NOT EXISTS
      (SELECT *
       FROM PROJECT
        WHERE DEPTNO = X.DEPTNO)
ORDER BY DEPTNO
```

Il est possible de lier les prédicats EXISTS et NOT EXISTS à d'autres prédicats, en utilisant AND et OR dans la clause WHERE de la requête externe.

Prédicats quantifiés

Un prédicat quantifié compare une valeur à une collection de valeurs. Si une instruction FULLSELECT renvoie plusieurs valeurs, vous devez modifier les opérateurs de comparaison du prédicat, en ajoutant le suffixe ALL, ANY ou SOME. Ces suffixes déterminent le traitement dans le prédicat externe de l'ensemble de valeurs renvoyées. Nous prenons ci-dessous l'opérateur de comparaison > comme exemple, mais les remarques s'appliquent également aux autres opérateurs :

expression > ALL (FULLSELECT)

Le prédicat est vrai si l'expression est supérieure à chacune des valeurs renvoyées par l'instruction FULLSELECT. Si l'instruction FULLSELECT ne renvoie aucune valeur, le prédicat est vrai. Le résultat est faux si la relation précisée est fausse pour au moins une valeur. Notez que le prédicat quantifié <>ALL équivaut au prédicat NOT IN.

L'exemple suivant utilise une sous-requête et une comparaison avec > ALL pour rechercher les noms et les professions des employés gagnant plus que tous les directeurs :

```
SELECT LASTNAME, JOB
FROM EMPLOYEE
WHERE SALARY > ALL
      (SELECT SALARY
       FROM EMPLOYEE
        WHERE JOB='MANAGER')
```

expression > ANY (FULLSELECT)

Le prédicat est vrai si l'expression est supérieure à au moins l'une des valeurs renvoyées par l'instruction FULLSELECT. Si l'instruction FULLSELECT ne renvoie aucune valeur, le prédicat est faux. Notez que l'opérateur quantifié =ANY équivaut au prédicat IN.

expression > SOME (FULLSELECT)

SOME est synonyme de ANY.

Pour de plus amples informations sur les prédicats et les opérateurs, consultez le document *SQL Reference*.

Chapitre 7. Fonctions SQL évoluées

Le présent chapitre couvre plusieurs caractéristiques de DB2 Universal Database qui apportent plus d'efficacité à la conception des requêtes et permettent de les personnaliser. Il est indispensable que vous ayez une connaissance approfondie des fonctions SQL simples.

Les sujets suivants sont abordés :

- Respect des règles de gestion à l'aide de contraintes et de déclencheurs
- Jointures
- Requêtes ROLLUP et CUBE et Requêtes récursives
- Fonctions OLAP

Respect des règles de gestion à l'aide de contraintes et de déclencheurs

Dans le monde de l'entreprise, nous devons nous assurer que certaines règles sont toujours respectées. Par exemple, un employé travaillant sur un projet doit être inscrit au registre du personnel. Ou, peut-être, certains événements doivent se produire de manière systématique. Par exemple, si un vendeur réalise une vente, sa commission doit être augmentée.

DB2 Universal Database propose un ensemble de méthodes très utiles pour garantir le respect de ces règles :

- Les *contraintes d'unicité* empêchent qu'une ou plusieurs colonnes d'une table puissent contenir des valeurs en double.
- Les *contraintes d'intégrité référentielle* garantissent la cohérence des données dans les différentes tables précisées.
- Les *contraintes de vérification de table* sont des conditions qui limitent les valeurs utilisées dans une colonne. Les insertions et les mises à jour n'aboutissent pas si la valeur affectée à une colonne ne satisfait pas les contraintes de vérification pour cette colonne.
- Les *déclencheurs* définissent un ensemble d'actions exécutées, ou déclenchées, par une suppression, une insertion ou une mise à jour dans une table précisée. Ils peuvent être utilisés pour écrire dans d'autres tables, modifier des valeurs d'entrée et émettre des messages d'alerte.

Cette première section présente une vue d'ensemble des clés. Dans les sections qui suivent, plusieurs exemples et diagrammes permettent d'étudier les possibilités offertes par l'intégrité référentielle, les contraintes et les déclencheurs.

Clés

Une *clé* est un ensemble de colonnes à utiliser pour identifier une ou plusieurs lignes ou pour y accéder.

Une clé composée de plusieurs colonnes est appelée *clé composée*. Dans une table à clé composée, l'ordre des colonnes dans la clé ne correspond pas nécessairement à leur ordre dans la table.

Clés uniques

Une *clé unique* peut être définie comme une colonne (ou un ensemble de colonnes) qui ne peut pas contenir de valeurs en double. Les colonnes auxquelles s'applique une clé unique ne peuvent pas contenir de valeur NULL. Le respect de la contrainte est assuré par le gestionnaire de bases de données, pendant l'exécution des instructions INSERT et UPDATE. Plusieurs clés uniques peuvent être appliquées à une même table. Facultatifs, les clés uniques peuvent être définies dans les instructions CREATE TABLE et ALTER TABLE.

Clés primaires

Une *clé primaire* est une clé unique faisant partie de la définition de la table. Une table ne peut pas avoir plusieurs clés primaires, et les colonnes concernées par une clé primaire ne peuvent pas contenir de valeur NULL. Facultatifs, les clés primaires peuvent être définies dans les instructions CREATE TABLE et ALTER TABLE.

Clés associées

Une *clé associée* fait partie de la définition d'une contrainte référentielle. Aucune, une ou plusieurs clés associées peuvent être appliquées à une table. La valeur de la clé associée composée est NULL si un des composants de la valeur est NULL. Facultatifs, les clés associées peuvent être définies dans les instructions CREATE TABLE et ALTER TABLE.

Contraintes d'unicité

Une contrainte d'unicité garantit que les valeurs d'une clé sont uniques dans une même table. Facultatifs, les contraintes d'unicité peuvent être définies avec les instructions CREATE TABLE ou ALTER TABLE, en précisant une clause PRIMARY KEY ou UNIQUE. Par exemple, il est possible de définir une contrainte d'unicité sur la colonne EMPNO (matricules) d'une table, pour garantir que chaque numéro d'employé sera unique.

Contraintes d'intégrité référentielle

Les contraintes d'unicité et les clés associées permettent de définir les relations entre les tables et, par conséquent, de garantir le respect de certaines règles de gestion. Les contraintes utilisant des clés uniques et des clés associées sont couramment appelées contraintes d'intégrité référentielle. Une contrainte d'unicité référencée par une clé associée est appelée *clé parente*. Une clé associée se réfère à une clé parente spécifique, ou elle lui est reliée. Par

exemple, une règle peut fixer que tous les employés (table EMPLOYEE) doivent appartenir à un service existant (table DEPARTMENT). Ainsi, le numéro de service défini dans la table EMPLOYEE sera utilisé comme clé associée et le numéro de service défini dans la table DEPARTMENT sera utilisé comme clé primaire. Le diagramme suivant illustre les contraintes d'intégrité référentielle :

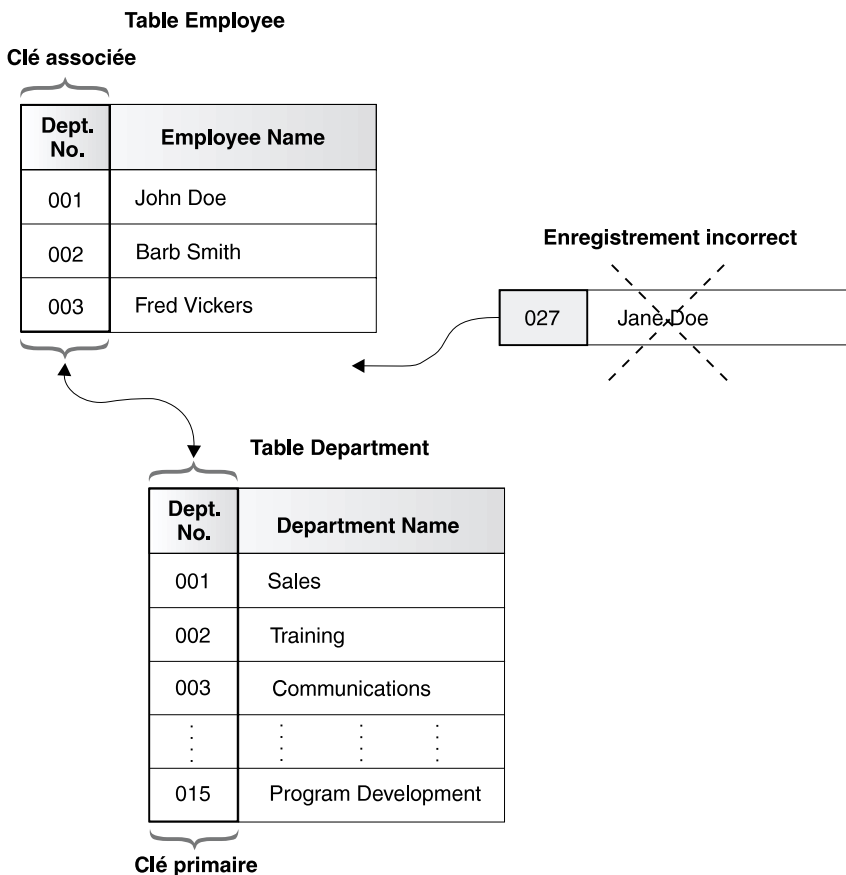


Figure 4. Les contraintes primaires et associées définissent les relations entre les données et assurent leur protection

Contraintes de vérification de table

Les *contraintes de vérification de table* indiquent les conditions qui sont vérifiées pour chaque ligne d'une table. Il est possible d'indiquer des contraintes de vérification de table individuelles. Elles peuvent être ajoutées au moyen des instructions CREATE ou ALTER TABLE.

L'instruction ci-dessous crée une table devant respecter les contraintes suivantes :

- Les valeurs du numéro de service doivent être comprises entre 10 et 100.
- Un employé ne peut avoir qu'un des postes suivants : "Sales", "Mgr" ou "Clerk".
- Tous les employés embauchés avant 1986 doivent gagner plus de 40500 F.

```
CREATE TABLE EMP
  (ID          SMALLINT NOT NULL,
   NAME        VARCHAR(9),
   DEPT        SMALLINT CHECK (DEPT BETWEEN 10 AND 100),
   JOB         CHAR(5)   CHECK (JOB IN ('Sales', 'Mgr', 'Clerk')),
   HIREDATE    DATE,
   SALARY      DECIMAL(7,2),
   COMM        DECIMAL(7,2),
   CLÉ PRIMAIRE (ID),
   CONSTRAINT YEARSAL CHECK (YEAR(HIREDATE) >= 1986 OR SALARY > 40500) )
```

Il n'y a violation de contrainte que si l'évaluation donne un résultat faux. Par exemple, si DEPT a une valeur NULL sur une ligne insérée, l'insertion se fait sans erreur, bien que la contrainte demande que les numéros de service soient compris entre 10 et 100.

L'instruction suivante indique pour la table EMPLOYEE une contrainte COMP selon laquelle le salaire compensé total de l'employé doit dépasser 15000 F :

```
ALTER TABLE EMP
  ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)
```

Le respect de la nouvelle contrainte doit également être vérifié dans les lignes existantes de la table. Il est possible de différer cette vérification, en utilisant l'instruction SET CONSTRAINTS de la manière suivante :

```
SET CONSTRAINTS FOR EMP OFF
ALTER TABLE EMP ADD CONSTRAINT COMP CHECK (SALARY + COMM > 15000)
SET CONSTRAINTS FOR EMP IMMEDIATE CHECKED
```

Tout d'abord, l'instruction SET CONSTRAINTS permet de différer la vérification des contraintes de la table. Ensuite, une ou plusieurs contraintes peuvent être ajoutées à la table sans être vérifiées. Enfin, l'instruction SET CONSTRAINTS, émise une nouvelle fois, réactive la vérification de la table, pour que les vérifications différées, s'il y en a, aient lieu.

Déclencheurs

Un *déclencheur* définit un ensemble d'actions activé par une opération qui modifie les données dans la table de base précisée.

Les déclencheurs peuvent être utilisés pour :

- valider les données en entrée ;

- générer automatiquement une valeur pour une nouvelle ligne insérée ;
- rechercher des références croisées dans d'autres tables ;
- écrire dans d'autres tables pour assurer la trace de contrôle ;
- prendre en charge des alertes via une messagerie électronique.

Les déclencheurs permettent de développer plus rapidement des applications, d'imposer des règles de gestion globales et de faciliter la maintenance des applications et des données.

DB2 Universal Database accepte plusieurs types de déclencheurs. La définition des déclencheurs peut prévoir qu'ils soient activés avant ou après une opération DELETE, INSERT ou UPDATE. Chaque déclencheur comporte un ensemble d'instructions SQL, appelé *action déclenchée*, qui peut inclure une condition de recherche facultative.

La définition des *déclencheurs AFTER* peut également préciser que l'action doit être déclenchée à chaque ligne ou une seule fois pour l'instruction entière, tandis que les *déclencheurs BEFORE* appliquent toujours l'action déclenchée à chacune des lignes.

Un déclencheur placé avant une instruction INSERT, UPDATE ou DELETE permet de vérifier que certaines conditions sont remplies avant d'exécuter une opération de déclenchement ou de modifier des valeurs avant de les stocker dans la table.

Un déclencheur AFTER permet de transmettre des valeurs en fonction des besoins ou d'exécuter d'autres tâches nécessaires au cours de l'opération de déclenchement, comme l'envoi d'un message.

L'exemple suivant décrit l'utilisation de déclencheurs BEFORE et AFTER. Prenons le cas d'une application qui enregistre et suit les modifications des cours de la Bourse. La base de données contient deux tables, CURRENTQUOTE et QUOTEHISTORY :

```
CREATE TABLE CURRENTQUOTE
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
STATUS VARCHAR(9))
```

```
CREATE TABLE QUOTEHISTORY
(SYMBOL VARCHAR(10),
QUOTE DECIMAL(5,2),
TIMESTAMP TIMESTAMP)
```

Lorsque la colonne QUOTE de CURRENTQUOTE est mise à jour à l'aide d'une instruction telle que :

```
UPDATE CURRENTQUOTE
SET QUOTE = 68.5
WHERE SYMBOLE = 'IBM'
```

La colonne STATUS de CURRENTQUOTE doit être mise à jour pour indiquer si la valeur de l'action :

- est en hausse
- est à son niveau le plus haut pour l'année
- est en baisse
- est à son niveau le plus bas pour l'année
- est stable

Cette mise à jour s'effectue à l'aide du déclencheur BEFORE ci-dessous :

1

```
CREATE TRIGGER STOCK_STATUS
NO CASCADE BEFORE UPDATE OF QUOTE ON CURRENTQUOTE
REFERENCING NEW AS NEWQUOTE OLD AS OLDQUOTE
FOR EACH ROW MODE DB2SQL
```

2

```
SET NEWQUOTE.STATUS =
```

3

```
CASE
```

4

```
WHEN NEWQUOTE.QUOTE >=
  (SELECT MAX(QUOTE)
   FROM QUOTEHISTORY
   WHERE SYMBOL = NEWQUOTE.SYMBOL
   AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
THEN 'En hausse'
```

5

```
WHEN NEWQUOTE.QUOTE <=
  (SELECT MIN(QUOTE)
   FROM QUOTEHISTORY
   WHERE SYMBOL = NEWQUOTE.SYMBOL
   AND YEAR(TIMESTAMP) = YEAR(CURRENT DATE) )
THEN 'En baisse'
```

6

```
WHEN NEWQUOTE.QUOTE > OLDQUOTE.QUOTE
  THEN 'En hausse'
WHEN NEWQUOTE.QUOTE < OLDQUOTE.QUOTE
  THEN 'En baisse'
WHEN NEWQUOTE.QUOTE = OLDQUOTE.QUOTE
  THEN 'Stable'
```

END

1

Ce bloc de syntaxe définit STOCK_STATUS en tant que déclencheur avant la mise à jour de la colonne QUOTE de la table CURRENTQUOTE. La deuxième ligne indique que l'action doit être déclenchée avant l'application, dans la base de données, des modifications qu'entraînerait la mise à jour de la table CURRENTQUOTE. La clause NO CASCADE signifie que l'action déclenchée n'activera pas elle-même d'autres déclencheurs. La troisième ligne précise les noms à utiliser comme qualificatifs de nom de colonne pour les nouvelles valeurs (NEWQUOTE) et les anciennes (OLDQUOTE). Les noms de colonne que NEWQUOTE et OLDQUOTE (noms de corrélation) qualifient sont appelés *variables de transition*. La quatrième ligne indique que l'action déclenchée doit s'exécuter pour chaque ligne.

2

Ce bloc indique le début de la première et unique instruction SQL de l'action déclenchée par ce déclencheur. Sur la ligne mise à jour par l'instruction ayant activé le déclencheur, l'instruction SET variable-transition affecte une valeur à une colonne. Dans notre exemple, l'instruction affecte une valeur à la colonne STATUS de la table CURRENTQUOTE.

3

L'expression utilisée à droite de l'affectation est une expression CASE. Elle se termine par le mot clé END.

4

Cette première expression CASE vérifie si la nouvelle cotation (NEWQUOTE.QUOTE) dépasse la valeur maximale des actions boursières fixée pour l'année en cours. La sous-requête utilise la table QUOTEHISTORY mise à jour par le déclencheur AFTER qui suit.

5

Cette deuxième expression CASE vérifie si la nouvelle cotation (NEWQUOTE.QUOTE) est inférieure à la valeur minimale des actions boursières fixée pour l'année en cours. La sous-requête utilise la table QUOTEHISTORY mise à jour par le déclencheur AFTER qui suit.

6

Ces trois dernières expressions CASE déterminent si la nouvelle cotation (NEWQUOTE.QUOTE) est supérieure, inférieure ou égale à la cotation (OLDQUOTE.QUOTE) comprise dans la table. L'instruction SET variable-transition se termine ici.

Outre la mise à jour de l'entrée dans la table `CURRENTQUOTE`, vous devez créer un enregistrement d'audit en copiant la nouvelle cotation ainsi que l'horodatage, dans la table `QUOTEHISTORY`. Cette mise à jour s'effectue à l'aide du déclencheur `AFTER` ci-dessous :

1

```
CREATE TRIGGER RECORD_HISTORY
AFTER UPDATE OF QUOTE ON CURRENTQUOTE
REFERENCING NEW AS NEWQUOTE
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
```

2

```
INSERT INTO QUOTEHISTORY
VALUES (NEWQUOTE.SYMBOL, NEWQUOTE.QUOTE, CURRENT TIMESTAMP);
END
```

1

Ce bloc de syntaxe définit que le déclencheur `RECORD_HISTORY` devra être activé après la mise à jour de la colonne `QUOTE` de la table `CURRENTQUOTE`. La troisième ligne indique le nom qui doit qualifier le nom de colonne de la nouvelle valeur (`NEWQUOTE`). La quatrième ligne indique que l'action déclenchée doit s'exécuter pour chaque ligne.

2

L'action entraînée par ce déclencheur comprend une seule instruction SQL qui insère une ligne dans la table `QUOTEHISTORY` à l'aide des données de la ligne mise à jour (`NEWQUOTE.SYMBOL` et `NEWQUOTE.QUOTE`) et de l'horodatage en cours.

`CURRENT DATESTAMP` est un registre spécial contenant l'horodatage. Vous trouverez une liste et une explication des registres à la section «Registres spéciaux» à la page 74.

Jointures

Le processus qui consiste à combiner des données provenant de plusieurs tables est appelé jointure. Avec le gestionnaire de bases de données, il est possible d'obtenir toutes sortes de combinaisons de lignes à partir des tables précisées. Il teste la *condition de jointure* de chaque combinaison. Une condition de jointure est une condition de recherche, avec quelques restrictions. Consultez le document *SQL Reference* qui recense ces restrictions.

Il est à noter que les colonnes impliquées dans la condition de jointure ne doivent pas forcément contenir des données de même type. Cependant, les données doivent être de types compatibles. La condition de jointure est évaluée de la même façon que n'importe quelle autre condition de recherche, et les mêmes règles de comparaison s'y appliquent.

Si vous ne précisez pas de condition de jointure, toutes les combinaisons de lignes possibles à partir des tables citées dans la clause FROM sont renvoyées, même si les lignes n'ont aucune relation entre elles. Le résultat peut être considéré comme un *produit croisé* des deux tables.

Les exemples de cette section sont basés sur les deux tables présentées ci-après. Il s'agit de versions simplifiées des tables de la base de données SAMPLE, mais elles n'existent pas dans la base. Elles illustrent quelques points intéressants sur les jointures en général. SAMP_STAFF dresse la liste des employés qui ne sont pas sous contrat, accompagnée des descriptions de poste. SAMP_PROJECT liste les employés (sous contrat et à plein temps) et les projets sur lesquels ils travaillent.

Voici comment se présentent les tables :

NAME	PROJ
Haas	AD3100
Thompson	PL2100
Walker	MA2112
Lutz	MA2111

Figure 5. TABLE SAMP_PROJECT

NAME	JOB
Haas	PRES
Thompson	MANAGER
Lucchessi	SALESREP
Nicholls	ANALYST

Figure 6. TABLE SAMP_STAFF

L'exemple suivant permet d'obtenir un résultat issu du produit croisé des deux tables. Aucune condition de jointure n'étant précisée, toutes les combinaisons de lignes possibles ont été trouvées :

```
SELECT SAMP_PROJECT.NAME,  
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT, SAMP_STAFF
```

Cette instruction donne le résultat suivant :

NAME	PROJ	NAME	JOB
-----	-----	-----	-----
Haas	AD3100	Haas	PRES
Thompson	PL2100	Haas	PRES
Walker	MA2112	Haas	PRES
Lutz	MA2111	Haas	PRES
Haas	AD3100	Thompson	MANAGER
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	Thompson	MANAGER
Lutz	MA2111	Thompson	MANAGER
Haas	AD3100	Lucchessi	SALESREP
Thompson	PL2100	Lucchessi	SALESREP
Walker	MA2112	Lucchessi	SALESREP
Lutz	MA2111	Lucchessi	SALESREP
Haas	AD3100	Nicholls	ANALYST
Thompson	PL2100	Nicholls	ANALYST
Walker	MA2112	Nicholls	ANALYST
Lutz	MA2111	Nicholls	ANALYST

Il y a deux principaux types de jointure : les *jointures internes* et les *jointures externes*. Jusqu'à présent, dans tous nos exemples, nous avons utilisé une jointure interne. Ce type de jointure ne garde du produit croisé que les lignes remplissant la condition de jointure. Si une ligne existe dans une table et pas dans l'autre, ses informations n'apparaissent pas dans la table résultat.

L'exemple suivant assure la jointure interne de deux tables. La jointure interne dresse la liste de tous les employés à plein temps affectés à un projet :

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT, SAMP_STAFF
WHERE SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

Il serait aussi possible de préciser la jointure interne de la manière suivante :

```

SELECT SAMP_PROJECT.NAME,
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT INNER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME

```

Voici le résultat obtenu :

NAME	PROJ	NAME	JOB
-----	-----	-----	-----
Haas	AD3100	Haas	PRES
Thompson	PL2100	Thompson	MANAGER

Le résultat de la jointure externe se compose de lignes dont les valeurs de la colonne NAME correspondent dans les tables de droite et de gauche - 'Haas' et 'Thompson' sont des noms que l'on retrouve dans la table SAMP_STAFF

contenant la liste de tous les employés à plein temps, et dans la table SAMP_PROJECT contenant la liste des employés à plein temps et sous contrat affectés à un projet.

Les jointures externes sont une concaténation de la jointure interne et des lignes de la table de gauche, de la table de droite, ou des deux tables absentes de la jointure interne. Lorsque vous exécutez une jointure externe sur deux tables, vous affectez arbitrairement l'une en tant que table de gauche et l'autre en tant que table de droite. Il existe trois types de jointure :

1. Une *jointure externe gauche*, qui inclut la jointure interne et les lignes de la table de gauche non comprises dans la jointure interne.
2. Une *jointure externe droite*, qui inclut la jointure interne et les lignes de la table de droite non comprises dans la jointure interne.
3. Une *jointure externe complète*, qui inclut la jointure interne et les lignes des tables de gauche et de droite non comprises dans la jointure interne.

Utilisez l'instruction SELECT pour préciser les colonnes à afficher. Dans la clause FROM, citez le nom de la table, suivi des mots clés LEFT OUTER JOIN, RIGHT OUTER JOIN ou FULL OUTER JOIN. Ensuite, précisez le nom de la seconde table, suivi du mot clé ON. Après le mot clé ON, précisez la condition de jointure, qui exprimera la relation entre les tables à joindre.

Dans l'exemple suivant, SAMP_STAFF est la table de droite et SAMP_PROJECT est la table de gauche. LEFT OUTER JOIN permet de dresser la liste des noms et des numéros de projet de tous les employés, à plein temps et sous contrat (liste de la table SAMP_PROJECT) ainsi que leur poste s'ils sont à plein temps (liste de la table SAMP_STAFF) :

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,
       SAMP_STAFF.NAME, SAMP_STAFF.JOB
FROM SAMP_PROJECT LEFT OUTER JOIN SAMP_STAFF
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Cette instruction donne le résultat suivant :

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
Lutz	MA2111	-	-
Thompson	PL2100	Thompson	MANAGER
Walker	MA2112	-	-

Les lignes comportant des valeurs dans toutes les colonnes sont le résultat de la jointure interne. Elles remplissent la condition de jointure : 'Haas' et 'Thompson' sont présents dans les deux tables, SAMP_PROJECT (table de gauche) et SAMP_STAFF (table de droite). Sur les lignes ne remplissant pas la condition de jointure, la valeur NULL apparaît dans les colonnes de la table de droite : 'Lutz' et 'Walker' sont des employés sous contrat, présents dans la

table SAMP_PROJECT mais absents de la table SAMP_STAFF. Toutes les lignes de la table de gauche sont incluses dans l'ensemble de résultats.

Dans l'exemple suivant, SAMP_STAFF est la table de droite et SAMP_PROJECT est la table de gauche. RIGHT OUTER JOIN permet de dresser la liste des noms et des postes de tous les employés à plein temps (liste de la table SAMP_STAFF), et d'indiquer également leur numéro de projet, s'ils ont été affectés à un projet (liste de la table SAMP_PROJECT) :

```
SELECT SAMP_PROJECT.NAME,  
       SAMP_PROJECT.PROJ, SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT RIGHT OUTER JOIN SAMP_STAFF  
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Voici le résultat obtenu :

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER

Comme dans le cas de la jointure externe gauche, les lignes comportant des valeurs dans toutes les colonnes sont le résultat de la jointure interne. Elles remplissent la condition de jointure : 'Haas' et 'Thompson' sont présents dans les deux tables, SAMP_PROJECT (table de gauche) et SAMP_STAFF (table de droite). Sur les lignes ne remplissant pas la condition de jointure, la valeur NULL apparaît dans les colonnes de la table de droite : 'Lucchessi' et 'Nicholls' sont des employés à plein temps non affectés à un projet. Ils font partie de la liste du personnel de la table SAMP_STAFF, mais sont absents de la table SAMP_PROJECT. Toutes les lignes de la table de droite sont incluses dans l'ensemble de résultats.

L'exemple suivant utilise FULL OUTER JOIN sur les tables SAMP_PROJECT et SAMP_STAFF. Il donne les noms de tous les employés à plein temps, y compris ceux qui ne sont pas affectés à un projet, et de tous les employés sous contrat :

```
SELECT SAMP_PROJECT.NAME, SAMP_PROJECT.PROJ,  
       SAMP_STAFF.NAME, SAMP_STAFF.JOB  
FROM SAMP_PROJECT FULL OUTER JOIN SAMP_STAFF  
ON SAMP_STAFF.NAME = SAMP_PROJECT.NAME
```

Voici le résultat obtenu :

NAME	PROJ	NAME	JOB
Haas	AD3100	Haas	PRES
-	-	Lucchessi	SALESREP
-	-	Nicholls	ANALYST
Thompson	PL2100	Thompson	MANAGER
Lutz	MA2111	-	-
Walker	MA2112	-	-

Le résultat inclut la jointure externe gauche, la jointure externe droite et la jointure interne. La liste inclut tous les employés à plein temps et sous contrat. Comme dans le cas d'une jointure externe gauche ou droite, la valeur NULL apparaît lorsque la condition de jointure n'est pas remplie. Toutes les lignes des tables SAMP_STAFF et SAMP_PROJECT sont incluses dans l'ensemble de résultats.

Requêtes complexes

DB2 Universal Database permet de grouper, consolider et visualiser plusieurs colonnes dans un même ensemble de résultats, via les requêtes ROLLUP et CUBE. Cette nouvelle fonction, par sa puissance, améliore et simplifie l'analyse des données SQL.

Plusieurs méthodes permettent d'extraire des informations utiles de la base de données. Il est possible, par exemple, d'implémenter des requêtes récursives, qui produiront des tables résultat à partir d'ensembles de données existants.

Requêtes ROLLUP et CUBE

Les opérations ROLLUP et CUBE sont définies dans la clause GROUP BY d'une requête.

Un groupage par ROLLUP permet d'obtenir, comme ensemble de résultats, les lignes groupées normales et les *lignes de sous-totaux*. Un groupage par CUBE permet d'obtenir, comme ensemble de résultats, les lignes trouvées par ROLLUP et les lignes de tabulation croisée.

Ainsi, ROLLUP permettra d'obtenir les ventes par personne et par mois, ainsi que le total des ventes du mois et un total général. CUBE ajoutera des lignes au total des ventes par personne.

Pour plus de détails, consultez le manuel *SQL Reference*.

Requêtes récursives

Une *requête récursive* est une requête qui utilise les résultats de manière itérative, pour déterminer de nouveaux résultats. Elle pourrait être représentée comme une arborescence ou un graphique.

Parmi les exemples pratiques, citons les systèmes de réservation et la planification de réseau.

Une requête récursive utilise une expression de table commune incluant une référence à son propre nom.

Le document *SQL Reference* contient des exemples de requêtes récursives.

Fonctions OLAP

Les fonctions de traitement analytique en ligne (OLAP) permettent d'effectuer des opérations de fonction de colonne sur une *fenêtre* de données. Cette fenêtre peut définir un partitionnement de lignes, un classement de lignes dans des partitions ou un *groupe d'agrégation*. Le groupe d'agrégation permet à l'utilisateur d'indiquer quelles sont les lignes, en relation avec la ligne en cours, qui font l'objet d'un calcul. L'utilisation de cette fenêtre permet d'effectuer des opérations telles que le calcul des sommes cumulées et des moyennes mobiles.

Outre la possibilité de fournir à l'utilisateur une fenêtre pour les fonctions de colonne existantes, (telles que SUM et AVG), les fonctions OLAP peuvent exécuter des opérations de classement (RANK et DENSE_RANK) et fournissent la numérotation des lignes (ROW_NUMBER), en fonction du partitionnement et de la numérotation des lignes.

L'exemple de requête suivant illustre le classement d'un employé dans un service en fonction de son salaire et affiche la somme cumulée des salaires du service (pour les services 15 et 38) :

```
SELECT NAME, DEPT,
       RANK () OVER (PARTITION BY DEPT ORDER BY SALARY DESC) AS RANK,
       SUM (SALARY) OVER (PARTITION BY DEPT
                        ORDER BY SALARY DESC
                        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
       AS CUMULATIVE_SUM
FROM STAFF
WHERE DEPT IN (15,38)
ORDER BY DEPT, RANK
```

Cette instruction donne le résultat suivant :

NAME	DEPT	RANK	CUMULATIVE_SUM
Hanes	15	1	20659.80
Rothman	15	2	37162.63
Ngan	15	3	49670.83
Kermisch	15	4	61929.33
O'Brien	38	1	18006.00
Marenghi	38	2	35512.75

Quigley	38	3	52321.05
Naughton	38	4	65275.80
Abrahams	38	5	77285.55

Chapitre 8. Personnalisation et amélioration de la manipulation des données

Le présent chapitre contient une courte présentation des *extensions orientées objet* de DB2 Universal Database. Il y a de nombreux avantages à utiliser des extensions orientées objet. Les *types définis par l'utilisateur (UDT)* augmentent l'ensemble de types de données dont vos applications disposent et les *fonctions définies par l'utilisateur (UDF)* permettent la création de fonctions spécifiques à une application. Les fonctions UDF servent de *méthodes*, qui assurent un comportement et un encapsulage cohérents entre les différents types UDT.

Vous y trouverez ensuite une description des *registres spéciaux* et des *catalogues système*. Un registre spécial est une zone de mémoire réservée à la connexion par le gestionnaire de bases de données et conservant des informations pouvant être référencées dans des instructions SQL. Les registres spéciaux sont établis au moment de la connexion et dépendent du traitement de l'application. Les catalogues système vous renseignent sur la structure logique et physique des objets de base de données.

Les sujets suivants sont abordés :

- Types définis par l'utilisateur
- Fonctions définies par l'utilisateur (UDF)
- Objets LOB
- Registres spéciaux
- Présentation des vues de catalogue

Les sujets ci-dessus font l'objet d'explications détaillées dans les documents *SQL Reference* et *Administration Guide*.

Types définis par l'utilisateur

Un *type distinct* est un type de données défini par l'utilisateur dont la représentation interne est commune avec celle d'un type existant (son type «source»), mais considéré comme distinct et incompatible pour la plupart des opérations. Par exemple, il est possible de définir un type Age, un type Poids et un type Taille, utilisant chacun une sémantique différente, mais partageant le même type de données intégré, INTEGER, pour leurs représentations internes.

L'exemple suivant illustre la création d'un type distinct, nommé PAY :

```
CREATE DISTINCT TYPE PAY AS DECIMAL(9,2) WITH COMPARISONS
```

Bien que PAY ait la même représentation que le type de données intégré DECIMAL(9,2), il est considéré comme un type distinct non compatible avec un type DECIMAL(9,2) ou tout autre type. Il ne peut être comparé qu'au même type distinct. En outre, les opérateurs et fonctions qui fonctionneraient avec le type DECIMAL ne s'appliquent pas ici. Par exemple, une valeur de type PAY ne peut pas être multipliée par une valeur de type INTEGER. Il faut donc dans ce cas écrire des fonctions ne s'appliquant qu'au type PAY.

L'utilisation de types de données distincts limite les erreurs accidentelles. Par exemple, si la colonne SALARY de la table EMPLOYEE a été définie comme type de données PAY, elle ne peut pas être ajoutée à COMM, même si les types source sont les mêmes.

Les types de données distincts prennent en charge le transtypage. Un type source peut être converti en type de données distinct et un type de données distinct en type source. Par exemple, si la colonne SALARY de la table EMPLOYEE a été définie comme étant de type PAY, il n'y aura pas, dans l'exemple suivant, d'erreur sur l'opérateur de comparaison.

```
SELECT * FROM EMPLOYEE  
WHERE DECIMAL(SALARY) = 41250
```

DECIMAL(SALARY) renvoie un type de données décimal. Inversement, un type de données numérique peut être converti en type PAY. Par exemple, il est possible d'utiliser PAY(41250) pour convertir le nombre 41250.

Fonctions définies par l'utilisateur (UDF)

Comme le mentionne la section «Utilisation de fonctions» à la page 28, DB2 Universal Database fournit des fonctions intégrées et des fonctions UDF (définies par l'utilisateur). Cependant, cet ensemble de fonctions ne répond jamais à toutes les exigences. Souvent, il faut créer des fonctions personnalisées pour certaines tâches. Les fonctions UDF permettent de créer des fonctions personnalisées.

Il existe quatre types de fonctions UDF : la fonction *dérivée* (ou modèle), la fonction *scalaire externe*, la fonction de *table externe* et la fonction de *table externe OLE DB*.

La présente section traite des types de fonctions dérivées ou scalaires externes. Pour plus de détails sur les types de fonctions de table externe et de table OLE DB, reportez-vous au manuel *SQL Reference*.

Les fonctions UDF dérivées permettent aux UDT de faire une référence sélective à une autre fonction intégrée ou UDF, déjà connue de la base de données. Il est possible d'utiliser les fonctions scalaires comme les fonctions de colonne.

Dans l'exemple suivant, la nouvelle fonction UDF (MAX) est basée sur la fonction de colonne intégrée MAX, qui prend en entrée les données de type DECIMAL. La fonction UDF MAX prend PAY comme type d'entrée et de sortie.

```
CREATE FUNCTION MAX(PAY) RETURNS PAY  
SOURCE MAX(DECIMAL)
```

Les fonctions UDF externes sont écrites en langage de programmation. Il peut s'agir de *fonctions scalaires externes* et de *fonctions de table externes*. Ces deux types sont décrits dans le document *SQL Reference*.

Prenons le cas où vous avez déjà écrit une fonction destinée à compter le nombre de mots d'une chaîne. Il est possible de l'enregistrer avec la base de données, sous le nom WORDCOUNT, avec l'instruction CREATE FUNCTION. Ensuite, cette fonction peut être utilisée dans les instructions SQL.

L'instruction suivante renvoie les numéros des employés et le nombre de mots que contient leur fiche sous forme ASCII. WORDCOUNT est une fonction scalaire externe enregistrée avec la base de données et utilisée dans l'instruction.

```
SELECT EMPNO, WORDCOUNT(RESUME)  
FROM EMP_RESUME  
WHERE RESUME_FORMAT = 'ascii'
```

Pour plus de détails sur les fonctions UDF, consultez le document *Application Development Guide*.

Objets LOB

Le terme *LOB* (pour Large Object) fait référence à trois types de données : BLOB, CLOB et DBCLOB. Il désigne des types d'objets peuvent contenir beaucoup de données, comme des documents, des photos et des fichiers audio.

Un *objet BLOB* (Binary Large Object) est une chaîne de longueur variable mesurée en octets, qui peut atteindre 2 giga-octets. Un objet BLOB est principalement destiné aux données non traditionnelles comme les images, les données vocales et les supports mixtes.

Un *objet CLOB* (Character-Large Object) est une chaîne de longueur variable mesurée en octets, qui peut atteindre 2 giga-octets. Un objet CLOB sert à conserver de gros fichiers de caractères à un octet, comme des documents. Un objet CLOB est considéré comme une chaîne de caractères.

Un *objet DBCLOB* (Double-Byte Character Large Object) est une chaîne de longueur variable, composée de caractères double octet, qui peut atteindre 2 giga-octets (1 073 741 823 caractères double octet). Un objet DBCLOB sert à stocker de gros fichiers de caractères à deux octets, comme des documents. Un objet DBCLOB est considéré comme une chaîne graphique.

Manipulation des objets LOB

Comme les valeurs LOB représentent souvent de gros volumes de données, le fait de les transférer du serveur de bases de données vers un programme d'application client risque de prendre beaucoup de temps. Cependant, les valeurs LOB sont traitées une par une et non comme une entité. Lorsqu'il n'est pas nécessaire que la valeur LOB en entier soit stockée dans la mémoire de l'application, il est possible, pour faire référence à cette valeur, d'utiliser une variable *releveur de coordonnées LOB*.

Les instructions suivantes peuvent utiliser ces releveurs pour effectuer des opérations sur les données sans avoir à extraire l'objet LOB en entier. Ces variables permettent de réduire le flot de données entre le client et le serveur, et limitent donc les besoins en mémoire des applications et améliorent les performances.

Les *variables de référence à un fichier* constituent un autre mécanisme. Ces variables servent à extraire directement un fichier un objet LOB ou à mettre directement à jour un objet LOB dans une table, à partir d'un fichier. Les variables de référence à un fichier réduisent les besoins en mémoire des applications, qui n'ont pas à stocker les données des objets LOB. Pour de plus amples informations, consultez les documents *Application Development Guide* et *SQL Reference*.

Registres spéciaux

Un *registre spécial* est une zone de mémoire réservée à la connexion par le gestionnaire de bases de données et conservant des informations pouvant être référencées dans des instructions SQL. Vous trouverez ci-après quelques exemples des registres spéciaux les plus utilisés. Le document *SQL Reference* contient plus de détails sur les registres spéciaux et fournit leur liste complète.

- CURRENT DATE : Date de l'horloge système au moment de l'exécution de l'instruction SQL.
- CURRENT FUNCTION PATH : Chemin d'accès utilisé pour résoudre les références aux fonctions et aux types de données.

- **CURRENT SERVER** : Serveur d'applications utilisé.
- **CURRENT TIME** : Heure de l'horloge système au moment de l'exécution de l'instruction SQL.
- **CURRENT TIMESTAMP** : Horodatage de l'horloge système au moment de l'exécution de l'instruction SQL.
- **CURRENT TIMEZONE** : Différence entre l'heure TUC (Temps Universel Coordonné) et l'heure locale du serveur d'applications.
- **USER** : Précise l'ID autorisation de l'exécution.

L'instruction **VALUES** permet d'afficher le contenu d'un registre spécial. Par exemple :

```
VALUES (CURRENT TIMESTAMP)
```

Vous pouvez également spécifier :

```
SELECT CURRENT TIMESTAMP FROM ORG
```

L'horodatage sera renvoyé pour toutes les lignes de la table.

Présentation des vues de catalogue

DB2 crée et assure la maintenance de plusieurs tables par base de données, destinées aux catalogues système. Ces tables contiennent des informations sur la structure logique et physique des objets de bases de données, comme les tables, les vues, les modules, les relations d'intégrité référentielle, les fonctions, les types distincts et les déclencheurs. Elles sont créées en même temps que la base de données et mises à jour dans le cadre d'opérations normales. Il n'est pas possible de les créer ou de les supprimer explicitement, mais vous pouvez les interroger et en visualiser le contenu.

Pour de plus amples informations, consultez le document *SQL Reference*.

Sélection de lignes des catalogues système

Les vues de catalogues ressemblent à n'importe quelle autre vue de base de données. Il est possible d'utiliser les instructions SQL pour visualiser leurs données, exactement comme vous le feriez pour n'importe quelle autre vue du système.

Le catalogue SYSCAT.TABLES contient des informations très utiles sur les tables. Pour rechercher les noms des tables que vous avez déjà créées, utilisez une instruction semblable à celle présentée ci-après :

```
SELECT TABNAME, TYPE, CREATE_TIME
FROM SYSCAT.TABLES
WHERE DEFINER = USER
```

Cette instruction donne le résultat suivant :

TABNAME	TYPE	CREATE_TIME
ORG	T	1999-07-21-13.42.55.128005
STAFF	T	1999-07-21-13.42.55.609001
DEPARTMENT	T	1999-07-21-13.42.56.069001
EMPLOYEE	T	1999-07-21-13.42.56.310001
EMP_ACT	T	1999-07-21-13.42.56.710001
PROJECT	T	1999-07-21-13.42.57.051001
EMP_PHOTO	T	1999-07-21-13.42.57.361001
EMP_RESUME	T	1999-07-21-13.42.59.154001
SALÉS	T	1999-07-21-13.42.59.855001
CL_SCHED	T	1999-07-21-13.43.00.025002
IN_TRAY	T	1999-07-21-13.43.00.055001

Vous trouverez ci-dessous la liste des vues de catalogues liées aux sujets traités dans le présent manuel. Il en existe de nombreuses autres, dont vous trouverez une liste détaillée dans les documents *SQL Reference* et *Administration Guide*.

Description	Vue de catalogue
Contraintes de vérification	SYSCAT.CHECKS
Colonnes	SYSCAT.COLUMNS
Colonnes référencées par des contraintes de vérification	SYSCAT.COLCHECKS
Colonnes utilisées dans des clés	SYSCAT.KEYCOLUSE
Types de données	SYSCAT.DATATYPES
Paramètres de fonction ou résultat d'une fonction	SYSCAT.FUNCPARMS
Contraintes référentielles	SYSCAT.REFERENCES
Schémas	SYSCAT.SCHEMATA
Contraintes de table	SYSCAT.TABCONST
Tables	SYSCAT.TABLES
Déclencheurs	SYSCAT.TRIGGERS
Fonctions définies par l'utilisateur	SYSCAT.FUNCTIONS
Vues	SYSCAT.VIEWS

Annexe A. Tables de la base de données SAMPLE

Vous trouverez ci-après les informations que contiennent les modèles de tables de la base de données SAMPLE, ainsi que la manière de créer et de retirer ces derniers.

Des modèles de bases de données supplémentaires sont fournis avec DB2 Universal Database à des fins de démonstration des fonctions d'intelligence économique ; ils sont utilisés avec le tutoriel correspondant. Seul le contenu de la base de données SAMPLE est décrit dans la présente annexe. Pour plus de détails sur les bases de données exemple concernant l'intelligence économique, reportez-vous au manuel *Data Warehouse Center Administration Guide*.

Ces modèles, utilisés dans les exemples du présent manuel, apparaissent également dans d'autres manuels de la même bibliothèque. Vous trouverez également dans la présente annexe les modèles de fichiers utilisant des données de types BLOB et CLOB.

L'annexe se compose des sections suivantes :

- «Base de données SAMPLE» à la page 78
- «Création de la base de données SAMPLE» à la page 78
- «Effacement de la base de données SAMPLE» à la page 79
- «Table CL_SCHED» à la page 79
- «Table DEPARTMENT» à la page 79
- «Table EMPLOYEE» à la page 80
- «Table EMP_ACT» à la page 83
- «Table EMP_PHOTO» à la page 85
- «Table EMP_RESUME» à la page 85
- «Table IN_TRAY» à la page 86
- «Table ORG» à la page 86
- «Table PROJECT» à la page 87
- «Table SALES» à la page 88
- «Table STAFF» à la page 89
- «Table STAFFG» à la page 90
- «Modèles de fichiers utilisant les types d'objet BLOB et CLOB» à la page 92
- «Photo de Dolores Quintana» à la page 92
- «Fiche descriptive de Dolores Quintana» à la page 92
- «Photo d'Heather A. Nicholls» à la page 93
- «Fiche descriptive d'Heather A. Nicholls» à la page 93
- «Photo de Bruce Adamson» à la page 95
- «Fiche descriptive de Bruce Adamson» à la page 95
- «Photo de James H. Walker» à la page 96

Tables de la base de données SAMPLE

«Fiche descriptive de James H. Walker» à la page 96.

Dans les modèles de tables, un tiret (-) représente une valeur nulle.

Base de données SAMPLE

Les exemples du présent document utilisent un modèle de base de données, appelé SAMPLE, que vous devez créer. Vérifiez aussi que le gestionnaire de bases de données est installé.

Création de la base de données SAMPLE

La base de données SAMPLE est créée par un fichier exécutable.² Pour créer une base de données, vous devez disposer du droit d'accès SYSADM.

- **Installation sur des plateformes UNIX**

Si vous tapez votre commande à l'invite du système d'exploitation, utilisez la syntaxe suivante :

```
sql1lib/bin/db2saml <chemin_d'accès>
```

dans le répertoire privé du propriétaire de l'instance du gestionnaire de bases de données. *chemin_d'accès* est un paramètre facultatif qui précise l'emplacement où doit être créée la base de données SAMPLE. Appuyez sur Entrée.³ Le schéma correspondant à DB2SAMPL est la valeur de registre spéciale CURRENT SCHEMA.

- **Installation sur des plateformes OS/2 ou Windows**

Si vous tapez votre commande à l'invite du système d'exploitation, utilisez la syntaxe suivante :

```
db2saml e
```

où *e* est un paramètre facultatif qui précise l'unité sur laquelle créer la base de données. Appuyez sur Entrée.⁴

Si vous n'êtes pas connecté à votre poste de travail par Gestion des profils utilisateur, un message vous invite à le faire.

2. Pour plus de détails sur la commande DB2SAMPL, consultez le manuel *Command Reference*.

3. Si le paramètre PATH n'est pas indiqué, la base de données SAMPLE est créée à l'emplacement indiqué par le paramètre DFTDBPATH (chemin d'accès par défaut) dans le fichier de configuration du gestionnaire de bases de données.

4. Si le paramètre d'unité n'est pas précisé, la base de données SAMPLE est créée sur la même unité que DB2.

Effacement de la base de données SAMPLE

Si vous n'utilisez pas la base de données SAMPLE, il vous est possible de l'effacer par la commande DROP DATABASE :

```
db2 drop database sample
```

Table CL_SCHED

Nom :	CLASS_CODE	DAY	STARTING	ENDING
Type :	char(7)	smallint	time	time
Desc :	Code classe (salle:enseignant)	Jour # du planning sur 4 jours	Heure début cours	Heure fin cours

Table DEPARTMENT

Nom :	DEPTNO	DEPTNAME	MGRNO	ADMNDEPT	LOCATION
Type :	char(3) not null	varchar(29) not null	char(6)	char(3) not null	char(16)
Desc :	Numéro du service	Description générale des activités du service	Matricule (EMPNO) du directeur de service	Service (DEPTNO) dont ce service dépend	Nom de la ville
Valeurs :	A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
	B01	PLANNING	000020	A00	-
	C01	INFORMATION CENTER	000030	A00	-
	D01	DEVELOPMENT CENTER	-	A00	-
	D11	MANUFACTURING SYSTEMS	000060	D01	-
	D21	ADMINISTRATION SYSTEMS	000070	D01	-
	E01	SUPPORT SERVICES	000050	A00	-
	E11	OPERATIONS	000090	E01	-
	E21	SOFTWARE SUPPORT	000100	E01	-

Tables de la base de données SAMPLE

Table EMPLOYEE

Noms :	EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE
Type :	char(6) not null	varchar(12) not null	char(1) not null	varchar(15) not null	char(3)	char(4)	date
Desc :	Matricule de l'employé	Premier prénom	Initiale du deuxième prénom	Nom	Service (DEPTNO) dans lequel travaille l'employé	Numéro de téléphone	Date d'embauche

JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
Poste	Nombre d'années d'études	Sexe (M masculin, F féminin)	Date de naissance	Salaire annuel	Prime annuelle	Commission annuelle

La page suivante présente les différentes valeurs de la table EMPLOYEE.

EMPNO	FIRSTNAME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
char(6) not null	vchar(12) not null	char(1) not null	vchar(15) not null	char(3)	char(4)	date	char(8)	smallint not null	char(1)	date	dec(9,2)	dec(9,2)	dec(9,2)
000010	CHRISTINE		HAAS	A00	3978	1965-01-01	PRES	18	F	1933-08-24	52750,00	1000,00	4220,00
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER	18	M	1948-02-02	41250,00	800,00	3300,00
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER	20	F	1941-05-11	38250,00	800,00	3060,00
000050	JOHN	B	CHEYER	E01	6789	1949-08-17	MANAGER	16	M	1952-09-15	40175,00	800,00	3214,00
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER	16	M	1945-07-07	32250,00	500,00	2580,00
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36170,00	700,00	2893,00
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29750,00	600,00	2380,00
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26150,00	500,00	2092,00
000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46500,00	900,00	3720,00
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M	1942-10-18	29250,00	600,00	2340,00
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1952-09-15	23800,00	500,00	1904,00
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28420,00	600,00	2274,00
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M	1947-05-17	25280,00	500,00	2022,00
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22250,00	400,00	1780,00
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER	16	M	1951-01-05	24680,00	500,00	1974
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21340,00	500,00	1707,00
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20450,00	400,00	1636,00
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M	1941-05-29	27740,00	600,00	2217,00
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18270,00	400,00	1462,00
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29840,00	600,00	2387,00
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK	14	M	1935-05-30	22180,00	400,00	1774,00
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M	1954-03-31	28760,00	600,00	2301,00
000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M	1939-11-12	19180,00	400,00	1534,00
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F	1936-10-05	17250,00	300	1380,00
000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F	1953-05-26	27380,00	500,00	2190,00
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F	1936-03-28	26250,00	500,00	2100,00
000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M	1946-07-09	15340,00	300	1227,00
000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M	1936-10-27	17750,00	400,00	1420,00
000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F	1931-04-21	15900,00	300	1272,00
000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M	1932-08-11	19950,00	400,00	1596,00

Tables de la base de données SAMPLE

EMPNO	FIRSTNME	MID INIT	LASTNAME	WORK DEPT	PHONE NO	HIREDATE	JOB	ED LEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000330	WING		LEE	E21	2103	1976-02-23	FIELDREP	14	M	1941-07-18	25370,00	500,00	2030,00
000340	JASON	R	GOUNOT	E21	5698	1947-05-05	FIELDREP	16	M	1926-05-17	23840,00	500,00	1907,00

Table EMP_ACT

Nom :	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
Type :	char(6) not null	char(6) not null	smallint not null	dec(5,2)	date	date
Desc :	Matricule de l'employé	Numéro du projet	Numéro de l'activité	Temps consacré par l'employé à travailler sur le projet	Date de début de l'activité	Date de fin de l'activité
Valeurs :	000010	AD3100	10	0,50	1982-01-01	1982-07-01
	000070	AD3110	10	1,00	1982-01-01	1983-02-01
	000230	AD3111	60	1,00	1982-01-01	1982-03-15
	000230	AD3111	60	0,50	1982-03-15	1982-04-15
	000230	AD3111	70	0,50	1982-03-15	1982-10-15
	000230	AD3111	80	0,50	1982-04-15	1982-10-15
	000230	AD3111	180	1,00	1982-10-15	1983-01-01
	000240	AD3111	70	1,00	1982-02-15	1982-09-15
	000240	AD3111	80	1,00	1982-09-15	1983-01-01
	000250	AD3112	60	1,00	1982-01-01	1982-02-01
	000250	AD3112	60	0,50	1982-02-01	1982-03-15
	000250	AD3112	60	0,50	1982-12-01	1983-01-01
	000250	AD3112	60	1,00	1983-01-01	1983-02-01
	000250	AD3112	70	0,50	1982-02-01	1982-03-15
	000250	AD3112	70	1,00	1982-03-15	1982-08-15
	000250	AD3112	70	0,25	1982-08-15	1982-10-15
	000250	AD3112	80	0,25	1982-08-15	1982-10-15
	000250	AD3112	80	0,50	1982-10-15	1982-12-01
	000250	AD3112	180	0,50	1982-08-15	1983-01-01
	000260	AD3113	70	0,50	1982-06-15	1982-07-01
	000260	AD3113	70	1,00	1982-07-01	1983-02-01
	000260	AD3113	80	1,00	1982-01-01	1982-03-01
	000260	AD3113	80	0,50	1982-03-01	1982-04-15
	000260	AD3113	180	0,50	1982-03-01	1982-04-15
	000260	AD3113	180	1,00	1982-04-15	1982-06-01
	000260	AD3113	180	0,50	1982-06-01	1982-07-01
	000270	AD3113	60	0,50	1982-03-01	1982-04-01
	000270	AD3113	60	1,00	1982-04-01	1982-09-01
	000270	AD3113	60	0,25	1982-09-01	1982-10-15
	000270	AD3113	70	0,75	1982-09-01	1982-10-15

Tables de la base de données SAMPLE

Nom :	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000270	AD3113	70	1,00	1982-10-15	1983-02-01
	000270	AD3113	80	1,00	1982-01-01	1982-03-01
	000270	AD3113	80	0,50	1982-03-01	1982-04-01
	000030	IF1000	10	0,50	1982-06-01	1983-01-01
	000130	IF1000	90	1,00	1982-01-01	1982-10-01
	000130	IF1000	100	0,50	1982-10-01	1983-01-01
	000140	IF1000	90	0,50	1982-10-01	1983-01-01
	000030	IF2000	10	0,50	1982-01-01	1983-01-01
	000140	IF2000	100	1,00	1982-01-01	1982-03-01
	000140	IF2000	100	0,50	1982-03-01	1982-07-01
	000140	IF2000	110	0,50	1982-03-01	1982-07-01
	000140	IF2000	110	0,50	1982-10-01	1983-01-01
	000010	MA2100	10	0,50	1982-01-01	1982-11-01
	000110	MA2100	20	1,00	1982-01-01	1982-03-01
	000010	MA2110	10	1,00	1982-01-01	1983-02-01
	000200	MA2111	50	1,00	1982-01-01	1982-06-15
	000200	MA2111	60	1,00	1982-06-15	1983-02-01
	000220	MA2111	40	1,00	1982-01-01	1983-02-01
	000150	MA2112	60	1,00	1982-01-01	1982-07-15
	000150	MA2112	180	1,00	1982-07-15	1983-02-01
	000170	MA2112	60	1,00	1982-01-01	1983-06-01
	000170	MA2112	70	1,00	1982-06-01	1983-02-01
	000190	MA2112	70	1,00	1982-02-01	1982-10-01
	000190	MA2112	80	1,00	1982-10-01	1983-10-01
	000160	MA2113	60	1,00	1982-07-15	1983-02-01
	000170	MA2113	80	1,00	1982-01-01	1983-02-01
	000180	MA2113	70	1,00	1982-04-01	1982-06-15
	000210	MA2113	80	0,50	1982-10-01	1983-02-01
	000210	MA2113	180	0,50	1982-10-01	1983-02-01
	000050	OP1000	10	0,25	1982-01-01	1983-02-01
	000090	OP1010	10	1,00	1982-01-01	1983-02-01
	000280	OP1010	130	1,00	1982-01-01	1983-02-01
	000290	OP1010	130	1,00	1982-01-01	1983-02-01
	000300	OP1010	130	1,00	1982-01-01	1983-02-01
	000310	OP1010	130	1,00	1982-01-01	1983-02-01
	000050	OP2010	10	0,75	1982-01-01	1983-02-01
	000100	OP2010	10	1,00	1982-01-01	1983-02-01

Tables de la base de données SAMPLE

Nom :	EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	000320	OP2011	140	0,75	1982-01-01	1983-02-01
	000320	OP2011	150	0,25	1982-01-01	1983-02-01
	000330	OP2012	140	0,25	1982-01-01	1983-02-01
	000330	OP2012	160	0,75	1982-01-01	1983-02-01
	000340	OP2013	140	0,50	1982-01-01	1983-02-01
	000340	OP2013	170	0,50	1982-01-01	1983-02-01
	000020	PL2100	30	1,00	1982-01-01	1982-09-15

Table EMP_PHOTO

Nom :	EMPNO	PHOTO_FORMAT	PICTURE
Type :	char(6) not null	varchar(10) not null	blob(100k)
Desc :	Matricule de l'employé	Format de la photo	Photo de l'employé
Valeurs :	000130	bitmap	db200130.bmp
	000130	gif	db200130.gif
	000130	xwd	db200130.xwd
	000140	bitmap	db200140.bmp
	000140	gif	db200140.gif
	000140	xwd	db200140.xwd
	000150	bitmap	db200150.bmp
	000150	gif	db200150.gif
	000150	xwd	db200150.xwd
	000190	bitmap	db200190.bmp
	000190	gif	db200190.gif
	000190	xwd	db200190.xwd

- La «Photo de Dolores Quintana» à la page 92, représente l'employée Dolores Quintana.
- La «Photo d'Heather A. Nicholls» à la page 93, représente l'employée Heather Nicholls.
- La «Photo de Bruce Adamson» à la page 95, représente l'employé Bruce Adamson.
- La «Photo de James H. Walker» à la page 96, représente l'employé James Walker.

Table EMP_RESUME

Nom :	EMPNO	RESUME_FORMAT	RESUME
Type :	char(6) not null	varchar(10) not null	clob(5k)
Desc :	Matricule de l'employé	Format de la fiche	Fiche de l'employé

Tables de la base de données SAMPLE

Nom :	EMPNO	RESUME_FORMAT	RESUME
Valeurs :	000130	ascii	db200130.asc
	000130	script	db200130.scr
	000140	ascii	db200140.asc
	000140	script	db200140.scr
	000150	ascii	db200150.asc
	000150	script	db200150.scr
	000190	ascii	db200190.asc
	000190	script	db200190.scr

- La «Fiche descriptive de Dolores Quintana» à la page 92, concerne l'employée Dolores Quintana.
- La «Fiche descriptive d'Heather A. Nicholls» à la page 93, concerne l'employée Heather Nicholls.
- La «Fiche descriptive de Bruce Adamson» à la page 95, concerne l'employé Bruce Adamson.
- La «Fiche descriptive de James H. Walker» à la page 96, concerne l'employé James Walker.

Table IN_TRAY

Nom :	RECEIVED	SOURCE	SUBJECT	NOTE_TEXT
Type :	timestamp	char(8)	char(64)	varchar(3000)
Desc :	Date et heure de réception	ID de l'expéditeur de la note	Commentaire	Note

Table ORG

Nom :	DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
Type :	smallint not null	varchar(14)	smallint	varchar(10)	varchar(13)
Desc :	Numéro du service	Nom du service	Matricule du directeur	Région où se trouve la société	Ville
Valeurs :	10	Siège	160	Corporate	New York
	15	New England	50	Eastern	Boston
	20	Mid Atlantic	10	Eastern	Washington
	38	South Atlantic	30	Eastern	Atlanta
	42	Great Lakes	100	Midwest	Chicago
	51	Plains	140	Midwest	Dallas
	66	Pacific	270	Western	San Francisco
	84	Mountain	290	Western	Denver

Table PROJECT

Nom :	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
Type :	char(6) not null	varchar(24) not null	char(3) not null	char(6) not null	dec(5,2)	date	date	char(6)
Desc :	Numéro du projet	Nom du projet	Responsable du service	Responsable ressources humaines	Effectif moyen estimé	Date de début estimée	Date de fin estimée	Projet principal, s'il s'agit d'un sous-projet
Valeurs :	AD3100	ADMIN SERVICES	D01	000010	6.5	1982-01-01	1983-02-01	-
	AD3110	GENERAL ADMIN SYSTEMS	D21	000070	6	1982-01-01	1983-02-01	AD3100
	AD3111	PAYROLL PROGRAMMING	D21	000230	2	1982-01-01	1983-02-01	AD3110
	AD3112	PERSONNEL PROGRAMMING	D21	000250	1	1982-01-01	1983-02-01	AD3110
	AD3113	ACCOUNT PROGRAMMING	D21	000270	2	1982-01-01	1983-02-01	AD3110
	IF1000	QUERY SERVICES	C01	000030	2	1982-01-01	1983-02-01	-
	IF2000	USER EDUCATION	C01	000030	1	1982-01-01	1983-02-01	-
	MA2100	WELD LINE AUTOMATION	D01	000010	12	1982-01-01	1983-02-01	-
	MA2110	W L PROGRAMMING	D11	000060	9	1982-01-01	1983-02-01	MA2100
	MA2111	W L PROGRAM DESIGN	D11	000220	2	1982-01-01	1982-12-01	MA2110
	MA2112	W L ROBOT DESIGN	D11	000150	3	1982-01-01	1982-12-01	MA2110
	MA2113	W L PROD CONT PROGS	D11	000160	3	1982-02-15	1982-12-01	MA2110
	OP1000	OPERATION SUPPORT	E01	000050	6	1982-01-01	1983-02-01	-
	OP1010	OPERATION	E11	000090	5	1982-01-01	1983-02-01	OP1000
	OP2000	GEN SYSTEMS SERVICES	E01	000050	5	1982-01-01	1983-02-01	-
	OP2010	SYSTEMS SUPPORT	E21	000100	4	1982-01-01	1983-02-01	OP2000
	OP2011	SCP SYSTEMS SUPPORT	E21	000320	1	1982-01-01	1983-02-01	OP2010
	OP2012	APPLICATIONS SUPPORT	E21	000330	1	1982-01-01	1983-02-01	OP2010
	OP2013	DB/DC SUPPORT	E21	000340	1	1982-01-01	1983-02-01	OP2010

Tables de la base de données SAMPLE

Nom :	PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE	MAJPROJ
	PL2100	WELD LINE PLANNING	B01	000020	1	1982-01-01	1982-09-15	MA2100

Table SALES

Nom :	SALES_DATE	SALES_PERSON	REGION	SALES
Type :	date	varchar(15)	varchar(15)	int
Desc :	Date de la vente	Nom de l'employé	Région de la vente	Nombre de ventes
Valeurs :	1995-12-31	LUCCHESSI	Ontario-South	1
	1995-12-31	LEE	Ontario-South	3
	1995-12-31	LEE	Quebec	1
	1995-12-31	LEE	Manitoba	2
	1995-12-31	GOUNOT	Quebec	1
	1996-03-29	LUCCHESSI	Ontario-South	3
	1996-03-29	LUCCHESSI	Quebec	1
	1996-03-29	LEE	Ontario-South	2
	1996-03-29	LEE	Ontario-North	2
	1996-03-29	LEE	Quebec	3
	1996-03-29	LEE	Manitoba	5
	1996-03-29	GOUNOT	Ontario-South	3
	1996-03-29	GOUNOT	Quebec	1
	1996-03-29	GOUNOT	Manitoba	7
	1996-03-30	LUCCHESSI	Ontario-South	1
	1996-03-30	LUCCHESSI	Quebec	2
	1996-03-30	LUCCHESSI	Manitoba	1
	1996-03-30	LEE	Ontario-South	7
	1996-03-30	LEE	Ontario-North	3
	1996-03-30	LEE	Quebec	7
	1996-03-30	LEE	Manitoba	4
	1996-03-30	GOUNOT	Ontario-South	2
	1996-03-30	GOUNOT	Quebec	18
	1996-03-30	GOUNOT	Manitoba	1
	1996-03-31	LUCCHESSI	Manitoba	1
	1996-03-31	LEE	Ontario-South	14
	1996-03-31	LEE	Ontario-North	3
	1996-03-31	LEE	Quebec	7
	1996-03-31	LEE	Manitoba	3
	1996-03-31	GOUNOT	Ontario-South	2
	1996-03-31	GOUNOT	Quebec	1
	1996-04-01	LUCCHESSI	Ontario-South	3
	1996-04-01	LUCCHESSI	Manitoba	1
	1996-04-01	LEE	Ontario-South	8

Tables de la base de données SAMPLE

Nom :	SALES_DATE	SALES_PERSON	REGION	SALES
	1996-04-01	LEE	Ontario-North	-
	1996-04-01	LEE	Quebec	8
	1996-04-01	LEE	Manitoba	9
	1996-04-01	GOUNOT	Ontario-South	3
	1996-04-01	GOUNOT	Ontario-North	1
	1996-04-01	GOUNOT	Quebec	3
	1996-04-01	GOUNOT	Manitoba	7

Table STAFF

Nom :	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type :	smallint not null	varchar(9)	smallint	char(5)	smallint	dec(7,2)	dec(7,2)
Desc :	Matricule de l'employé	Nom de l'employé	Numéro du service	Poste	Années de présence	Salaire actuel	Commission
Valeurs :	10	Sanders	20	Mgr	7	18357,50	-
	20	Pernal	20	Sales	8	18171,25	612,45
	30	Marenghi	38	Mgr	5	17506,75	-
	40	O'Brien	38	Sales	6	18006,00	846,55
	50	Hanes	15	Mgr	10	20659,80	-
	60	Quigley	38	Sales	-	16808,30	650,25
	70	Rothman	15	Sales	7	16502,83	1152,00
	80	James	20	Clerk	-	13504,60	128,20
	90	Koonitz	42	Sales	6	18001,75	1386,70
	100	Plotz	42	Mgr	7	18352,80	-
	110	Ngan	15	Clerk	5	12508,20	206,60
	120	Naughton	38	Clerk	-	12954,75	180,00
	130	Yamaguchi	42	Clerk	6	10505,90	75,60
	140	Fraye	51	Mgr	6	21150,00	-
	150	Williams	51	Sales	6	19456,50	637,65
	160	Molinare	10	Mgr	7	22959,20	-
	170	Kermisch	15	Clerk	4	12258,50	110,10
	180	Abrahams	38	Clerk	3	12009,75	236,50
	190	Sneider	20	Clerk	8	14252,75	126,50
	200	Scoutten	42	Clerk	-	11508,60	84,20
	210	Lu	10	Mgr	10	20010,00	-
	220	Smith	51	Sales	7	17654,50	992,80
	230	Lundquist	51	Clerk	3	13369,80	189,65
	240	Daniels	10	Mgr	5	19260,25	-

Tables de la base de données SAMPLE

Nom :	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	250	Wheeler	51	Clerk	6	14460,00	513,30
	260	Jones	10	Mgr	12	21234,00	-
	270	Lea	66	Mgr	9	18555,50	-
	280	Wilson	66	Sales	9	18674,50	811,50
	290	Quill	84	Mgr	10	19818,00	-
	300	Davis	84	Sales	5	15454,50	806,10
	310	Graham	66	Sales	13	21000,00	200,30
	320	Gonzales	66	Sales	4	16858,20	844,00
	330	Burke	66	Clerk	1	10988,00	55,50
	340	Edwards	84	Sales	7	17844,00	1285,00
	350	Gafney	84	Clerk	5	13030,50	188,00

Table STAFFG

Remarque : La table STAFFG ne concerne que les pages de codes double octet.

Nom :	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
Type :	smallint not null	varchar(9)	smallint	graphic(5)	smallint	dec(9,0)	dec(9,0)
Desc :	Matricule de l'employé	Nom de l'employé	Numéro du service	Poste	Années de présence	Salaire actuel	Commission
Valeurs :	10	Sanders	20	Mgr	7	18357,50	-
	20	Pernal	20	Sales	8	18171,25	612,45
	30	Marengi	38	Mgr	5	17506,75	-
	40	O'Brien	38	Sales	6	18006,00	846,55
	50	Hanes	15	Mgr	10	20659,80	-
	60	Quigley	38	Sales	-	16808,30	650,25
	70	Rothman	15	Sales	7	16502,83	1152,00
	80	James	20	Clerk	-	13504,60	128,20
	90	Koonitz	42	Sales	6	18001,75	1386,70
	100	Plotz	42	Mgr	7	18352,80	-
	110	Ngan	15	Clerk	5	12508,20	206,60
	120	Naughton	38	Clerk	-	12954,75	180,00
	130	Yamaguchi	42	Clerk	6	10505,90	75,60
	140	Fraye	51	Mgr	6	21150,00	-
	150	Williams	51	Sales	6	19456,50	637,65
	160	Molinare	10	Mgr	7	22959,20	-
	170	Kermisch	15	Clerk	4	12258,50	110,10

Tables de la base de données SAMPLE

Nom :	ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
	180	Abrahams	38	Clerk	3	12009,75	236,50
	190	Sneider	20	Clerk	8	14252,75	126,50
	200	Scoutten	42	Clerk	-	11508,60	84,20
	210	Lu	10	Mgr	10	20010,00	-
	220	Smith	51	Sales	7	17654,50	992,80
	230	Lundquist	51	Clerk	3	13369,80	189,65
	240	Daniels	10	Mgr	5	19260,25	-
	250	Wheeler	51	Clerk	6	14460,00	513,30
	260	Jones	10	Mgr	12	21234,00	-
	270	Lea	66	Mgr	9	18555,50	-
	280	Wilson	66	Sales	9	18674,50	811,50
	290	Quill	84	Mgr	10	19818,00	-
	300	Davis	84	Sales	5	15454,50	806,10
	310	Graham	66	Sales	13	21000,00	200,30
	320	Gonzales	66	Sales	4	16858,20	844,00
	330	Burke	66	Clerk	1	10988,00	55,50
	340	Edwards	84	Sales	7	17844,00	1285,00
	350	Gafney	84	Clerk	5	13030,50	188,00

Modèles de fichiers utilisant les types d'objet BLOB et CLOB

Vous trouverez dans cette section les données contenues dans les fichiers de EMP_PHOTO (photographies des employés) et de EMP_RESUME (fiches des employés).

Photo de Dolores Quintana



Figure 7. Dolores M. Quintana

Fiche descriptive de Dolores Quintana

Le texte suivant se trouve dans les fichiers db200130.asc et db200130.scr.

Fiche descriptive : Dolores M. Quintana

Personal Information

Address:	1150 Eglinton Ave Mellonville, Idaho 83725
Phone:	(208) 555-9933
Birthdate:	September 15, 1952
Sex:	Female
Marital Status:	Married
Height:	5'2"
Weight:	120 lbs.

Department Information

Employee Number:	000130
Dept Number:	C01
Manager:	Sally Kwan
Position:	Analyst
Phone:	(208) 555-4578
Hire Date:	1971-07-28

Education

1965	Math and English, B.A. Adelphi University
-------------	---

1960

Dental Technician Florida Institute of Technology

Work History

10/91 - present

Advisory Systems Analyst Producing documentation tools for engineering department.

12/85 - 9/91

Technical Writer Writer, text programmer, and planner.

1/79 - 11/85

COBOL Payroll Programmer Writing payroll programs for a diesel fuel company.

Interests

- Cooking
- Reading
- Sewing
- Remodeling

Photo d'Heather A. Nicholls



Figure 8. Heather A. Nicholls

Fiche descriptive d'Heather A. Nicholls

Le texte suivant se trouve dans les fichiers db200140.asc et db200140.scr.

Resume: Heather A. Nicholls

Personal Information

Address:

844 Don Mills Ave Mellonville, Idaho 83734

Phone:

(208) 555-2310

Birthdate:

January 19, 1946

Tables de la base de données SAMPLE

Sex: Female
Marital Status: Single
Height: 5'8"
Weight: 130 lbs.

Department Information

Employee Number: 000140
Dept Number: C01
Manager: Sally Kwan
Position: Analyst
Phone: (208) 555-1793
Hire Date: 1976-12-15

Education

1972 Computer Engineering, Ph.D. University of Washington

1969 Music and Physics, M.A. Vassar College

Work History

2/83 - present Architect, OCR Development Designing the architecture of OCR products.

12/76 - 1/83 Text Programmer Optical character recognition (OCR) programming in PL/I.

9/72 - 11/76 Punch Card Quality Analyst Checking punch cards met quality specifications.

Interests

- Model railroading
- Interior decorating
- Embroidery
- Knitting

Photo de Bruce Adamson

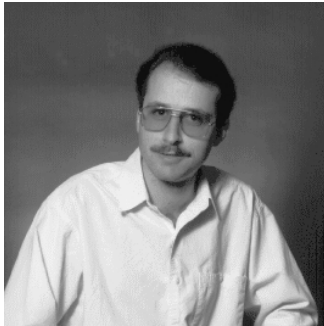


Figure 9. Bruce Adamson

Fiche descriptive de Bruce Adamson

Le texte suivant se trouve dans les fichiers db200150.asc et db200150.scr.

Resume: Bruce Adamson

Personal Information

Address:	3600 Steeles Ave Mellonville, Idaho 83757
Phone:	(208) 555-4489
Birthdate:	May 17, 1947
Sex:	Male
Marital Status:	Married
Height:	6'0"
Weight:	175 lbs.

Department Information

Employee Number:	000150
Dept Number:	D11
Manager:	Irving Stern
Position:	Designer
Phone:	(208) 555-4510
Hire Date:	1972-02-12

Education

1971	Environmental Engineering, M.Sc. Johns Hopkins University
1968	American History, B.A. Northwestern University

Tables de la base de données SAMPLE

Work History

8/79 - present

Neural Network Design Developing neural networks for machine intelligence products.

2/72 - 7/79

Robot Vision Development Developing rule-based systems to emulate sight.

9/71 - 1/72

Numerical Integration Specialist Helping bank systems communicate with each other.

Interests

- Racing motorcycles
- Building loudspeakers
- Assembling personal computers
- Sketching

Photo de James H. Walker

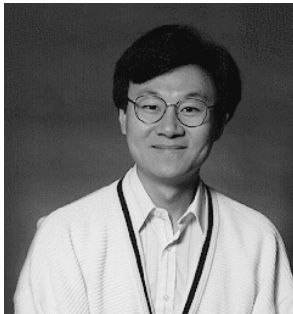


Figure 10. James H. Walker

Fiche descriptive de James H. Walker

Le texte suivant se trouve dans les fichiers db200190.asc et db200190.scr.

Resume: James H. Walker

Personal Information

Address:	3500 Steeles Ave Mellonville, Idaho 83757
Phone:	(208) 555-7325
Birthdate:	June 25, 1952
Sex:	Male
Marital Status:	Single
Height:	5'11"
Weight:	166 lbs.

Tables de la base de données SAMPLE

Department Information

Employee Number: 000190
Dept Number: D11
Manager: Irving Stern
Position: Designer
Phone: (208) 555-2986
Hire Date: 1974-07-26

Education

1974 Computer Studies, B.Sc. University of Massachusetts
1972 Linguistic Anthropology, B.A. University of Toronto

Work History

6/87 - present Microcode Design Optimizing algorithms for mathematical functions.
4/77 - 5/87 Printer Technical Support Installing and supporting laser printers.
9/74 - 3/77 Maintenance Programming Patching assembly language compiler for mainframes.

Interests

- Wine tasting
- Skiing
- Swimming
- Dancing

Tables de la base de données SAMPLE

Annexe B. Utilisation de la Bibliothèque DB2

La bibliothèque DB2 Universal Database est constituée de rubriques d'aide en ligne, de manuels au format HTML et PDF, et de programmes exemples au format HTML. La présente annexe décrit les informations disponibles et indique comment y accéder.

Le Centre d'aide et d'information vous aide à accéder aux informations en ligne relatives au produit. Pour plus de détails, reportez-vous à «Recherche d'informations à l'aide du Centre d'aide et d'information» à la page 117. Vous pouvez consulter des manuels DB2, afficher des informations sur les tâches et sur la résolution des incidents, visualiser des programmes exemples et avoir accès aux informations DB2 disponibles sur le Web.

Manuels imprimés et fichiers au format PDF DB2

Informations sur DB2

Le tableau suivant répartit les manuels db2 comme suit :

Manuels d'utilisation et de référence de DB2

Ces manuels contiennent les informations communes relatives à l'utilisation de DB2 sur toutes les plateformes.

Manuels d'installation et de configuration de DB2

Ces manuels concernent l'utilisation de DB2 sur une plateforme spécifique. Par exemple, il existe des manuels *Mise en route* distincts pour DB2 sur des plateformes OS/2, Windows et UNIX.

Exemples de programmes multiplateformes au format HTML

Ces exemples de programmes au format HTML sont installés avec le composant Application Development Client. Ils n'ont qu'une vocation informative et ne remplacent pas les programmes réels.

Documents "Release Notes"

Ces fichiers contiennent les informations de dernière minute n'ayant pas pu être intégrées dans les manuels DB2.

Les manuels d'installation, les remarques sur le produit et les tutoriels sont directement consultables en format HTML à partir du CD-ROM produit. La plupart des manuels existent au format HTML à des fins de consultation et au format Adobe Acrobat (PDF) sur le CD-ROM DB2 Universal Database publications pour consultation et impression. Pour commander une copie

papier des manuels, reportez-vous à «Commande des manuels imprimés» à la page 112 . Le tableau ci-après fournit la liste des manuels pouvant faire l'objet d'une commande.

Sur les plateformes OS/2 et Windows, vous pouvez installer les fichiers HTML dans le répertoire sqllib\doc\html. Les informations sur DB2 sont traduites en différentes langues mais pas nécessairement dans leur totalité. Lorsque des informations n'existent pas dans une langue déterminée, elles sont fournies en anglais.

Sur les plateformes UNIX, vous pouvez installer des versions multilingues des fichiers HTML dans les sous-répertoires doc/%L/html, %L représentant l'environnement local. Pour plus de détails, reportez-vous au manuel *Mise en route* approprié.

Vous pouvez vous procurer des manuels DB2 et accéder aux informations de différentes manières :

- «Affichage des informations en ligne» à la page 116
- «Recherche d'informations en ligne» à la page 120
- «Commande des manuels imprimés» à la page 112
- «Impression des manuels au format PDF» à la page 112

Tableau 1. Informations sur DB2

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
Manuels d'utilisation et de référence de DB2			
<i>Administration Guide</i>	<p data-bbox="397 343 825 543"><i>Administration Guide: Planning</i> présente les concepts mis en oeuvre dans les bases de données, fournit des informations sur les enjeux liés à la conception (conception logique et physique de base de données) et traite des fonctions de haute disponibilité.</p> <p data-bbox="397 569 825 716"><i>Administration Guide: Implementation</i> fournit des informations sur la mise en oeuvre de votre structure, de l'accès aux bases de données, du contrôle, de la sauvegarde et de la reprise, entre autres.</p> <p data-bbox="397 743 825 881"><i>Administration Guide: Performance</i> fournit des informations sur l'environnement de base de données, ainsi que sur l'évaluation et l'adaptation des performances.</p> <p data-bbox="397 907 825 1020">Les trois volumes du manuel <i>Administration Guide</i> en anglais peuvent être commandés sous la référence SBOF-8922.</p>	<p data-bbox="858 343 969 395">SC09-2946 db2d1x70</p> <p data-bbox="858 517 969 569">SC09-2944 db2d2x70</p> <p data-bbox="858 664 969 716">SC09-2945 db2d3x70</p>	db2d0
<i>Administrative API Reference</i>	<p data-bbox="397 1043 825 1156">Décrit les API et les structures de données DB2 utilisées pour gérer des bases de données. Explique comment appeler les API à partir des applications.</p>	<p data-bbox="858 1043 969 1121">SC09-2947 db2b0x70</p>	db2b0
<i>Application Building Guide</i>	<p data-bbox="397 1173 825 1373">Fournit des informations relatives à la configuration de l'environnement et présente, étape par étape, les instructions nécessaires à la compilation, à la définition des accès et à l'exécution d'applications DB2 sur les plateformes Windows, OS/2 et UNIX.</p>	<p data-bbox="858 1173 969 1251">SC09-2948 db2axx70</p>	db2ax

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>APPC, CPI-C, and SNA Sense Codes</i>	Fournit des informations générales relatives aux codes de détection APPC, CPI-C et SNA pouvant être rencontrés lors de l'utilisation des produits DB2 Universal Database.	Aucun numéro de référence	db2ap
	Ce manuel est disponible au format HTML uniquement.	db2apx70	
<i>Application Development Guide</i>	Fournit des informations relatives au développement d'applications accédant à des bases de données DB2 à l'aide d'instructions SQL imbriquées ou Java (JDBC et SQLJ). Ce manuel traite, entre autres, de l'écriture de procédures mémorisées et de fonctions UDF, de la création de types UDT, de l'utilisation des déclencheurs et du développement d'applications dans des environnements partitionnés ou avec des systèmes fédérés.	SC09-2949	db2a0
		db2a0x70	
<i>CLI Guide and Reference</i>	Décrit comment développer des applications permettant d'accéder à des bases de données DB2 à l'aide de l'interface DB2 CLI (interface SQL d'appel compatible avec le système ODBC de Microsoft).	SC09-2950	db2i0
		db2i0x70	
<i>Command Reference</i>	Explique comment utiliser l'interpréteur de commandes et fournit une description des commandes de gestion des bases de données.	SC09-2951	db2n0
		db2n0x70	

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>DB2 Connectivité - Informations complémentaires</i>	<p>Fournit des informations de référence et de configuration utiles à toute personne souhaitant utiliser DB2 pour AS/400, DB2 pour OS/390, DB2 pour MVS ou DB2 pour VM en tant que demandeurs d'application DRDA avec les serveurs DB2 Universal Database et aux personnes souhaitant utiliser les serveurs d'applications DRDA avec les demandeurs d'application DB2 Connect. Ce manuel détaille également l'utilisation de serveurs d'applications DRDA avec les demandeurs d'application DB2 Connect.</p> <p>Ce manuel est disponible au format HTML ou PDF uniquement.</p>	Aucun numéro de référence db2h1x70	db2h1
<i>Data Movement Utilities Guide and Reference</i>	Explique comment utiliser les utilitaires DB2 UDB qui simplifient le déplacement des données, tels que import, export, load, AutoLoader et DPROF.	SC09-2955 db2dmx70	db2dm
<i>Data Warehouse Center Administration Guide</i>	Fournit des informations sur la création et la gestion d'un entrepôt de données au moyen de Data Warehouse Center.	SC26-9993 db2ddx70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	Fournit des informations permettant aux programmeurs d'intégrer des applications via Data Warehouse Center et via Information Catalog Manager.	SC26-9994 db2adx70	db2ad
<i>DB2 Connect User's Guide</i>	Présente les concepts, ainsi que des informations générales et de programmation sur les produits DB2 Connect.	SC09-2954 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Fournit des indications générales sur le fonctionnement du système DB2 Query Patroller, des informations fonctionnelles et administratives spécifiques ainsi que des informations sur les utilitaires graphiques d'administration.	SC09-2958 db2dwx70	db2dw
<i>DB2 Query Patroller User's Guide</i>	Décrit l'utilisation des outils et des fonctions de DB2 Query Patroller.	SC09-2960 db2wwx70	db2ww

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>Glossaire</i>	Fournit des définitions de termes utilisés dans DB2 et ses composants. Ce glossaire est disponible au format HTML et dans le manuel <i>SQL Reference</i> .	Aucun numéro de référence db2t0x70	db2t0
<i>Extensions Image, Audio et Vidéo - Administration et programmation</i>	Fournit des informations générales sur les extensions DB2, ainsi que des informations sur l'administration et la configuration des extensions Image, Audio et Vidéo et la programmation via ces extensions. Il comporte des informations de référence, de diagnostic (avec des messages) et des exemples.	SC11-1682 dmbu7x70	dmbu7
<i>Information Catalog Manager Administration Guide</i>	Fournit des directives sur la gestion des catalogues d'informations.	SC26-9995 db2dix70	db2di
<i>Information Catalog Manager Programming Guide and Reference</i>	Fournit des définitions relatives aux interfaces structurées pour Information Catalog Manager.	SC26-9997 db2bix70	db2bi
<i>Information Catalog Manager - Guide de l'utilisateur</i>	Fournit des informations sur la mise en oeuvre de l'interface utilisateur d'Information Catalog Manager.	SC11-1678 db2aix70	db2ai
<i>Installation et configuration - Informations complémentaires</i>	Aide à la planification, l'installation et la configuration de clients DB2 en fonction de la plateforme utilisée. Ce supplément contient des informations sur la définition des accès et la configuration des communications client et serveur, l'interface graphique DB2 GUI, DRDA AS, l'installation répartie et présente également la configuration de requêtes réparties et de méthodes d'accès aux sources de données hétérogènes.	GC11-1641 db2iyx70	db2iy

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>Guide des messages</i>	Contient une liste des messages et des codes renvoyés par DB2, Information Catalog Manager et Data Warehouse Center, et décrit les opérations correctives à effectuer, le cas échéant. Les deux volumes du manuel Guide des messages en anglais peuvent être commandés sous la référence SBOF-8922.	Volume 1 GC11-1653 db2m1x70 Volume 2 GC11-1654	db2m0
<i>OLAP Integration Server Administration Guide</i>	Explique l'utilisation du composant Administration Manager de OLAP Integration Server.	SC27-0787 db2dpx70	n/a
<i>OLAP Integration Server Metaoutline User's Guide</i>	Explique comment créer et peupler des métastructures OLAP via l'interface standard OLAP Metaoutline (et non via l'Assistant Metaoutline).	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	Explique comment créer des modèles OLAP via l'interface standard OLAP Model Interface (et non via l'Assistant de modélisation).	SC27-0783 db2lpx70	n/a
<i>OLAP - Installation et utilisation</i>	Fournit des informations de configuration pour OLAP Starter Kit.	SC11-1700 db2ipx70	db2ip
<i>OLAP Spreadsheet Add-in User's Guide for Excel</i>	Décrit l'utilisation du tableur Excel pour analyser les données OLAP.	SC27-0786 db2epx70	db2ep
<i>OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</i>	Décrit l'utilisation du tableur Lotus 1-2-3 pour analyser les données OLAP.	SC27-0785 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	Fournit des informations de planification, de configuration, d'administration et d'utilisation sur les outils de réplication IBM livrés avec DB2.	SC26-9920 db2e0x70	db2e0

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>Extension Spatiale - Guide d'utilisation et de référence</i>	Fournit des informations d'installation, de configuration, d'administration, de programmation et d'identification et résolution des incidents pour l'extension Spatiale. Ce manuel contient également des descriptions détaillées des concepts de données spatiales et des informations de référence (messages et SQL) propres à l'extension Spatiale.	SC11-1684 db2sbx70	db2sb
<i>Initiation à SQL</i>	Présente les concepts SQL et fournit des exemples de structures et de tâches.	SC11-1655 db2y0x70	db2y0
<i>SQL Reference, Volume 1 et Volume 2</i>	Décrit la syntaxe des instructions SQL, les règles sémantiques et celles liées au langage. Fournit également des informations sur les incompatibilités entre versions, sur les limites des produits et les vues de catalogue. Les deux volumes du manuel <i>SQL Reference</i> en anglais peuvent être commandés sous la référence SBOF-8933.	Volume 1 SC09-2974 db2s1x70 Volume 2 SC09-2975 db2s2x70	db2s0
<i>System Monitor Guide and Reference</i>	Décrit comment collecter différents types d'informations relatives aux bases de données et au gestionnaire de bases de données. Décrit également comment exploiter les informations pour analyser l'activité de la base de données, améliorer les performances du système et déterminer l'origine des incidents.	SC09-2956 db2f0x70	db2f0
<i>Extension Texte - Administration et programmation</i>	Fournit des informations générales sur les extensions DB2, ainsi que des informations sur l'administration et la configuration de l'extension Texte et la programmation via cette extension. Il comporte des informations de référence, de diagnostic (avec des messages) et des exemples.	SC11-1683 desu9x70	desu9

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>Troubleshooting Guide</i>	Fournit des informations d'aide pour déterminer la source des erreurs, effectuer une récupération après incident et utiliser les outils de diagnostic mis à disposition par le service d'assistance DB2.	GC09-2850 db2p0x70	db2p0
<i>Nouveautés</i>	Décrit les nouvelles fonctions et améliorations apportées à DB2 Universal Database, version 7.	SC11-1656 db2q0x70	db2q0
Manuels d'installation et de configuration de DB2			
<i>DB2 Connect Enterprise Edition pour OS/2 et Windows - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Connect Enterprise Edition sous OS/2 et systèmes Windows 32 bits. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1640 db2c6x70	db2c6
<i>DB2 Connect Enterprise Edition pour UNIX - Mise en route</i>	Contient des informations sur la planification, l'installation, la configuration et les fonctions de DB2 Connect Enterprise Edition sous UNIX. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1639 db2cyx70	db2cy
<i>DB2 Connect Personal Edition - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et les fonctions de DB2 Connect Personal Edition sous OS/2 et systèmes Windows 32 bits. Contient également des informations d'installation et de configuration relatives à tous les clients pris en charge.	GC11-1647 db2c1x70	db2c1
DB2 Connect Personal Edition pour Linux - Mise en route	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Connect Personal Edition sur toutes les plateformes Linux.	GC11-1642 db2c4x70	db2c4

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>DB2 Data Links Manager - Mise en route</i>	Contient des informations sur la planification, l'installation et les fonctions de DB2 Data Links Manager sous AIX et Windows 32 bits.	GC11-1646 db2z6x70	db2z6
<i>DB2 Enterprise Extended Edition pour UNIX - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Enterprise - Extended Edition sous UNIX. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1644 db2v3x70	db2v3
<i>DB2 Enterprise - Extended Edition pour Windows - Mise en route</i>	Contient des informations sur la planification, l'installation et la configuration de DB2 Enterprise - Extended Edition sous Windows 32 bits. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1643 db2v6x70	db2v6
<i>DB2 pour OS/2 - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Universal Database sous OS/2. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1648 db2i2x70	db2i2
<i>DB2 pour UNIX - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Universal Database sous UNIX. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1650 db2ixx70	db2ix
<i>DB2 pour Windows - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Universal Database sous systèmes Windows 32 bits. Contient également des informations d'installation et de configuration relatives à la plupart des clients pris en charge.	GC11-1651 db2i6x70	db2i6

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
		Nom de fichier PDF	
<i>DB2 Personal Edition - Mise en route</i>	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Universal Database Personal Edition sous OS/2 et systèmes Windows 32 bits.	GC11-1649 db2i1x70	db2i1
DB2 Personal Edition pour Linux - Mise en route	Contient des informations sur la planification, la migration, l'installation et la configuration de DB2 Universal Database Personal Edition sur toutes les plateformes Linux.	GC11-1652 db2i4x70	db2i4
<i>DB2 Query Patroller Installation Guide</i>	Contient des informations relatives à l'installation de DB2 Query Patroller.	GC09-2959 db2iwx70	db2iw
DB2 Warehouse Manager - Installation	Fournit des informations sur l'installation d'agents d'entrepôt, des fonctions Transformation correspondantes et du produit Information Catalog Manager.	GC11-1681 db2idx70	db2id

Tableau 1. Informations sur DB2 (suite)

Nom	Description	Références	Répertoire HTML
Exemples de programmes multiplateformes au format HTML			
Nom de fichier PDF			
Exemples de programmes au format HTML	<p>Fournit les exemples de programmes au format HTML des langages de programmation sur toutes les plateformes prises en charges par DB2. Ces programmes sont fournis à titre informatif. Certains exemples de programmes ne sont pas disponibles dans tous les langages. Ces exemples de programmes ne sont disponibles que si le composant DB2 Application Development Client est installé.</p> <p>Pour plus de détails sur ces programmes, reportez-vous au manuel <i>Application Building Guide</i>.</p>	Aucun numéro de référence	db2hs
Remarques sur le produit			
<i>DB2 Connect Release Notes</i>	Ces documents contiennent les informations de dernière minute n'ayant pas pu être intégrées dans les manuels DB2 Connect.	Voir remarque #2.	db2cr
<i>Remarques sur l'installation de DB2</i>	Ces documents contiennent des informations de dernière minute relatives à l'installation et qui n'ont pas pu être intégrées dans les manuels DB2.	Ces documents sont disponibles sur le CD-ROM produit uniquement.	
<i>DB2 Release Notes</i>	Ces documents contiennent des informations de dernière minute relatives à l'ensemble des produits et fonctions DB2 et qui n'ont pas pu être intégrées dans les manuels DB2.	Voir remarque #2.	db2ir

Remarques :

1. Le caractère *x* se trouvant en sixième position dans le nom du fichier indique dans quelle langue est fourni le manuel. Par exemple, le nom de fichier *db2d0e70* identifie la version anglaise du manuel *Administration Guide* et *db2d0f70*, la version française. Les lettres ci-dessous sont utilisées dans les noms de fichier pour indiquer la langue dans laquelle les manuels sont disponibles.

Langue	Identificateur
Portugais (Brésil)	b
Bulgare	u
Tchèque	x
Danois	d
Néerlandais	q
Anglais	e
Finnois	y
Français	f
Allemand	g
Grec	a
Hongrois	h
Italien	i
Japonais	j
Coréen	k
Norvégien	n
Polonais	p
Portugais	v
Russe	r
Chinois simplifié	c
Slovène	l
Espagnol	z
Suédois	s
Chinois traditionnel	t
Turc	m

2. Les informations de dernière minute qui n'ont pas pu être intégrées dans les manuels DB2 sont disponibles dans les Remarques sur le produit au format HTML et sous forme de fichier ASCII. La version HTML peut être obtenue à partir du Centre d'aide et d'information et des CD-ROM produit. Pour consulter un fichier ASCII :
 - Pour les plateformes UNIX, reportez-vous au fichier Release.Notes qui se trouve dans le répertoire DB2DIR/Readme/%L où %L représente le nom d'environnement local et DB2DIR :
 - /usr/lpp/db2_07_01 sous AIX
 - /opt/IBMdb2/V7.1 sous HP-UX, PTX, Solaris et Silicon Graphics IRIX
 - /usr/IBMdb2/V7.1 sous Linux.
 - Pour les autres plateformes, reportez-vous au fichier RELEASE.TXT situé dans le répertoire d'installation du produit. Sur les plateformes OS/2, cliquez deux fois sur le dossier **DB2**, puis sur l'icône **Remarques sur le produit**.

Impression des manuels au format PDF

Si vous préférez disposer de documents imprimés, vous pouvez décompacter et imprimer les fichiers contenus sur le CD-ROM des publications DB2. Adobe Acrobat Reader vous permet d'imprimer la totalité d'un manuel ou un ensemble de pages déterminé. Pour connaître le nom de fichier correspondant à chaque manuel, reportez-vous au tableau tableau 1 à la page 101.

Vous pouvez obtenir la dernière version d'Adobe Acrobat Reader à partir du site Web Adobe en vous connectant à l'adresse <http://www.adobe.com>.

Les fichiers PDF se trouvent sur le CD-ROM des publications DB2 et sont dotés du suffixe PDF. Pour accéder à ces fichiers, procédez comme suit :

1. Insérez le CD-ROM des publications DB2. Sur les plateformes UNIX, montez-le. Pour connaître les procédures de montage du CD-ROM, reportez-vous au manuel *Mise en route*.
2. Démarrez Acrobat Reader.
3. Ouvrez le fichier PDF de votre choix dans un des répertoires suivants :
 - Sur les plateformes OS/2 et Windows :
répertoire `x:\doc\langue`, où `x` désigne l'unité de CD-ROM et `langue` le code pays à deux caractères correspondant à votre langue (par exemple, FR pour le français).
 - Sur des plateformes UNIX :
Répertoire `/cdrom/doc/%L` du CD-ROM, où `/cdrom` désigne le point de montage du CD-ROM et `%L` le nom de l'environnement local souhaité.

Vous pouvez également copier les fichiers PDF du CD-ROM sur une unité locale ou réseau, et les y consulter.

Commande des manuels imprimés

Vous pouvez commander les manuels DB2 imprimés séparément ou sous forme de jeu (pour l'Amérique du Nord uniquement) en utilisant une référence SBOF. Pour commander des manuels, contactez votre distributeur agréé ou votre partenaire commercial, ou composez le 1-800-879-2755 aux États-Unis ou le 1-800-IBM-4YOU au Canada. Vous pouvez aussi les commander sur la page Web Publications en vous connectant à l'adresse <http://www.elink.ibm.com/pbl/pbl>.

Il existe deux jeux de manuels disponibles. Le jeu SBOF-8935 fournit des informations d'utilisation et de référence sur DB2 Warehouse Manager, et le jeu SBOF-8931, des informations de même type concernant tous les autres produits et fonctions de DB2 Universal Database. Le contenu de chaque jeu est répertorié dans le tableau ci-après.

Tableau 2. Commande de manuels imprimés

Référence SBOF	Manuels
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center - Administration • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation et configuration - Informations complémentaires • Extensions Image, Audio et Vidéo : Administration et programmation • Guide des messages, Volumes 1 et 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP - Installation et utilisation • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Extension Spatiale : Administration et programmation • Initiation à SQL • SQL Reference, Volumes 1 et 2 • System Monitor Guide and Reference • Extension Texte : Administration et programmation • Troubleshooting Guide • Nouveautés
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager - Guide de l'utilisateur • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

Documentation en ligne DB2

Accès à l'aide en ligne

Des rubriques d'aide en ligne sont disponibles avec tous les composants DB2. Le tableau ci-après décrit les différents types d'aide disponibles.

Type d'aide	Contenu	Mode d'accès
Aide sur les commandes	Explique la syntaxe des commandes de l'interpréteur de commandes.	À partir de l'interpréteur de commandes, en mode interactif, entrez : <i>? commande</i> où <i>commande</i> correspond à un mot clé ou à la commande complète. Par exemple, <i>? catalog</i> permet d'afficher l'aide sur les commandes CATALOG, alors que <i>? catalog database</i> permet d'afficher l'aide sur la commande CATALOG DATABASE.
Aide sur l'Assistant de configuration client	Décrit les tâches à effectuer dans une fenêtre ou un bloc-notes. L'aide fournit les informations qu'il est	À partir d'une fenêtre ou d'un bloc-notes, cliquez sur le bouton de fonction Aide ou appuyez sur la touche F1 .
Aide sur le Centre de commande	nécessaire de connaître et décrit comment utiliser les	
Aide sur le Centre de contrôle	éléments de contrôle de fenêtre ou de bloc-notes.	
Aide sur Data Warehouse Center		
Aide sur l'Analyseur d'événements		
Aide sur Information Catalog Manager		
Aide sur le Centre d'administration des satellites		
Aide sur le Centre de gestion des scripts		

Type d'aide	Contenu	Mode d'accès
Aide sur les messages	Décrit l'origine d'un message et indique les opérations correctives à effectuer, le cas échéant.	<p>À partir de l'interpréteur de commandes, en mode interactif, entrez :</p> <p style="padding-left: 40px;"><i>? XXXnnnnn</i></p> <p>où <i>XXXnnnnn</i> correspond à un numéro de message correct.</p> <p>Par exemple, <i>? SQL30081</i> permet d'afficher l'aide sur le message SQL30081.</p> <p>Pour afficher l'aide écran par écran, entrez :</p> <p style="padding-left: 40px;"><i>? XXXnnnnn more</i></p> <p>Pour sauvegarder l'aide sur un message dans un fichier, entrez :</p> <p style="padding-left: 40px;"><i>? XXXnnnnn > nomfichier.ext</i></p> <p>où <i>nomfichier.ext</i> correspond au fichier dans lequel vous souhaitez sauvegarder l'aide sur un message.</p>
Aide sur le SQL	Décrit la syntaxe des instructions SQL.	<p>À partir de l'interpréteur de commandes, en mode interactif, entrez :</p> <p style="padding-left: 40px;"><i>help instruction</i></p> <p>où <i>instruction</i> correspond à une instruction SQL.</p> <p>Par exemple, <i>help SELECT</i> permet d'afficher l'aide sur l'instruction SELECT.</p> <p>Remarque : L'aide SQL n'est pas disponible sur les plateformes UNIX.</p>
Aide sur SQLSTATE	Décrit les codes SQLSTATE et de classe.	<p>À partir de l'interpréteur de commandes, en mode interactif, entrez :</p> <p style="padding-left: 40px;"><i>? sqlstate or ? code-classe</i></p> <p>où <i>sqlstate</i> correspond à un code d'état SQL correct composé de cinq chiffres et <i>code-classe</i> aux deux premiers chiffres du code d'état SQL.</p> <p>Par exemple, <i>? 08003</i> permet d'afficher l'aide sur l'état SQL 08003, alors que <i>? 08</i> permet de visualiser l'aide sur le code de classe 08.</p>

Affichage des informations en ligne

Les manuels livrés avec ce produit sont au format électronique HTML, ce qui vous permet de rechercher et de consulter aisément les informations, à l'aide de liens hypertexte. Cela permet également de partager plus efficacement la bibliothèque électronique entre les différents utilisateurs du site.

La visualisation des manuels en ligne et des exemples de programmes peut être effectuée à l'aide de tout navigateur compatible avec la version 3.2 de HTML.

Pour visualiser les manuels en ligne ou les exemples de programmes :

- Si vous utilisez les outils d'administration DB2, utilisez le Centre d'aide et d'information.
- Dans un navigateur, cliquez sur **Fichier** → **Ouvrir une page**. La page qui s'affiche contient des descriptions des manuels DB2 et les liens correspondants :

- Pour les plateformes UNIX, ouvrez la page suivante :

```
INSTHOME /sqllib/doc/%L/html/index.htm
```

où %L est le nom de l'environnement local.

- Pour les autres plateformes, ouvrez la page suivante :

```
sqllib\doc\html\index.htm
```

Cette page est disponible à partir de l'unité sur laquelle DB2 est installé.

Si le Centre d'aide et d'information n'est pas installé, vous pouvez également l'ouvrir en cliquant deux fois sur l'icône **Informations DB2**. Selon le système utilisé, cette icône est disponible à partir du dossier principal du produit ou du menu Démarrer de Windows.

Installation du navigateur Netscape

Si vous ne disposez pas encore d'un navigateur Web, vous pouvez installer Netscape à partir du CD-ROM correspondant fourni avec les produits DB2. Pour obtenir plus de détails sur la procédure d'installation, procédez comme suit :

1. Insérez le CD-ROM Netscape.
2. Montez le CD-ROM (uniquement sur les plateformes UNIX). Pour connaître les procédures de montage du CD-ROM, reportez-vous au manuel *Mise en route*.
3. Pour la procédure d'installation, reportez-vous au fichier CDNAV nn .txt, où nn désigne l'identificateur de langue à deux caractères. Ce fichier se trouve dans le répertoire principal du CD-ROM.

Recherche d'informations à l'aide du Centre d'aide et d'information

Le Centre d'aide et d'information permet d'accéder rapidement aux informations relatives à DB2. Le Centre d'aide et d'information est disponible sur toutes les plateformes sur lesquelles les outils d'administration DB2 sont installés.

Vous pouvez ouvrir le Centre d'aide et d'information en cliquant deux fois sur l'icône correspondante. Selon le système utilisé, cette icône est disponible à partir du dossier principal du produit ou du menu **Démarrer** de Windows.

Vous pouvez aussi accéder au Centre d'aide et d'information en utilisant la barre d'outils et le menu **Aide** sur la plateforme DB2 Windows.

Le Centre d'aide et d'information fournit six types d'informations. Cliquez sur l'onglet approprié pour afficher les informations.

Procédures Affiche la liste des tâches pouvant être exécutées à l'aide de DB2.

Référence Affiche la liste des informations de référence sur DB2 (mots clés, commandes, API, etc.).

Manuels Affiche la liste des manuels DB2.

Résolution des incidents

Affiche la liste des catégories de messages d'erreur et les actions correctives correspondantes.

Programmes exemples

Affiche la liste des exemples de programmes livrés avec l'application DB2 Application Development Client. Si cette application n'est pas installée, aucun onglet ne s'affiche.

Web Affiche la liste des informations DB2 disponibles sur le Web. Pour pouvoir accéder à ces informations, une connexion doit être établie entre votre système et le Web.

Lorsque vous sélectionnez un élément apparaissant dans l'une des listes, le Centre d'aide et d'information lance un programme d'affichage de sorte que vous puissiez consulter les informations correspondantes. Selon le type d'information sélectionné, il peut s'agir du programme système d'affichage de l'aide, d'un éditeur de texte ou d'un logiciel de navigation Web.

Le Centre d'aide et d'information comporte une fonction de recherche qui vous permet de localiser une rubrique déterminée sans consulter les listes.

Pour une recherche en texte intégral, suivez le lien hypertexte allant du Centre d'aide et d'information au formulaire **Recherche dans la documentation DB2**.

Normalement, le serveur de recherche HTML démarre automatiquement. Si une recherche effectuée dans les informations HTML est infructueuse, il peut être nécessaire de démarrer le serveur de recherche en procédant comme suit :

Sous Windows

Cliquez sur **Démarrer**, puis sélectionnez **Programmes** → **DB2** → **Informations** → **Démarrage du serveur de recherche HTML**.

Sous OS/2

Cliquez deux fois sur le dossier **DB2 pour OS/2**, puis à nouveau deux fois sur l'icône **Démarrage du serveur de recherche HTML**.

Si vous rencontrez des difficultés lors de vos recherches dans les informations HTML, reportez-vous aux remarques sur le produit.

Remarque : La fonction de recherche n'est pas disponible dans les environnements Linux, PTX et Silicon Graphics IRIX.

Assistants DB2

Les assistants vous guident dans l'exécution de certaines tâches d'administration en vous indiquant les étapes à effectuer les unes après les autres. Vous pouvez disposer des assistants via le Centre de contrôle et l'Assistant de configuration client. Le tableau suivant fournit la liste des assistants et en détaille les fonctions :

Remarque : Les assistants Création de base de données, Index et Mise à jour multisite sont disponibles pour l'environnement de bases de données partitionnées.

Assistant	Opération concernée	Mode d'accès
Ajout d'une base de données	Catalogage d'une base de données sur un poste de travail client.	À partir de l'Assistant de configuration client, cliquez sur Ajout .
Sauvegarde de base de données	Détermination, création et planification d'un plan de sauvegarde.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur la base de données que vous souhaitez sauvegarder et sélectionnez Sauvegarde → Base de données - Assistant .
Configuration de mise à jour multisite	Configuration d'une mise à jour multisite, d'une transaction répartie ou d'une validation en deux phases.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur le dossier Bases de données et sélectionnez Mise à jour multisite .

Assistant	Opération concernée	Mode d'accès
Création d'une base de données	Création d'une base de données et exécution de certaines tâches élémentaires de configuration.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur le dossier Bases de données et sélectionnez Création —> Base de données — Assistant .
Création d'une table	Sélection des types de données de base et création d'une clé primaire pour la table.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur l'icône Tables et sélectionnez Création —> Table — Assistant .
Création d'un espace table	Création d'un espace table.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur l'icône Espaces table et sélectionnez Création —> Espace table — Assistant .
Création d'index	Détermination des index à créer et à supprimer pour toutes vos requêtes.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur l'icône Index et sélectionnez Création —> Index — Assistant .
Configuration des performances	Ajustement des performances d'une base de données avec mise à jour des paramètres de configuration en fonction de vos besoins.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur la base de données dont vous voulez ajuster les performances et sélectionnez Configuration des performances - Assistant . Dans un environnement de bases de données partitionnées, dans l'écran Partitions de base de données, cliquez à l'aide du bouton droit de la souris sur la première partition dont vous voulez ajuster les performances, puis sélectionnez Configuration des performances - Assistant .
Restauration de la base de données	Récupération d'une base de données après un incident. Cet assistant vous aide à déterminer la copie de sauvegarde et les journaux à utiliser.	À partir du Centre de contrôle, cliquez à l'aide du bouton droit de la souris sur la base de données que vous souhaitez restaurer et sélectionnez Restauration —> Base de données - Assistant .

Configuration d'un serveur de documents

Les informations relatives à DB2 sont installées par défaut sur votre système local. Cela implique que les fichiers correspondants doivent être installés sur le poste de chaque utilisateur. Afin de l'éviter et de ne stocker les informations DB2 qu'à un seul emplacement, procédez comme suit :

1. Copiez tous les fichiers et sous-répertoires de `\sqlib\doc\html` à partir de votre système local vers un serveur Web. Chaque manuel dispose d'un sous-répertoire contenant tous les fichiers HTML et GIF qui le constituent. Assurez-vous que la structure de répertoire reste identique.
2. Configurez le serveur Web de sorte qu'il recherche les fichiers à leur nouvel emplacement. Pour plus de détails, reportez-vous au manuel *Installation et configuration - Informations complémentaires*.
3. La version Java de l'utilitaire Information Center (Infocentre) vous permet d'indiquer une adresse URL de base pour tous les fichiers HTML. Vous devez utiliser cette adresse URL pour obtenir la liste des manuels.
4. Une fois la visualisation des fichiers de manuels activée, il est conseillé de marquer par des signets les rubriques couramment consultées telles que :
 - La liste des manuels
 - La table de matières des manuels couramment affichés
 - Les rubriques fréquemment citées, telles que la rubrique ALTER TABLE.
 - Le formulaire de recherche

Pour plus d'informations sur la prise en charge des fichiers de documentation électronique DB2 Universal Database à partir d'une machine centrale, reportez-vous à l'Annexe relative à NetQuestion dans le manuel *Installation et configuration - Informations complémentaires*.

Recherche d'informations en ligne

Pour rechercher des informations dans les fichiers HTML, procédez selon l'une des méthodes suivantes :

- Cliquez sur **Recherche** dans la partie supérieure du cadre des manuels HTML. Utilisez le formulaire de recherche pour effectuer une recherche sur une rubrique particulière. La fonction de recherche n'est pas disponible dans les environnements Linux, PTX et Silicon Graphics IRIX.
- Cliquez sur **Index** dans la partie supérieure du cadre des manuels HTML. Utilisez l'index pour rechercher une rubrique spécifique dans un manuel.
- Affichez la table des matières ou l'index du manuel HTML et utilisez la fonction de recherche du navigateur Web pour rechercher une rubrique spécifique d'un manuel.
- Utilisez la fonction de signets de l'explorateur Web pour revenir rapidement à une rubrique spécifique.

- Utilisez la fonction de recherche du Centre d'aide et d'information pour effectuer une recherche sur des rubriques spécifiques. Pour plus de détails, reportez-vous à la section «Recherche d'informations à l'aide du Centre d'aide et d'information» à la page 117.

Annexe C. Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevets couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing
IBM Europe Middle-East Africa
Tour Descartes
La Défense 5
2, avenue Gambetta
92066 Paris-La Défense Cedex
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE «EN L'ETAT». IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Ces informations peuvent être soumises à des conditions particulières prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux termes du Contrat sur les produits et services IBM, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Ce document peut contenir des exemples de données et des rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel peut contenir des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquelles ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _indiquez l'année ou les années_. All rights reserved.

Marques

Les termes qui suivent, accompagnés d'un astérisque (*) dans le document, sont des marques d'International Business Machines Corporation dans certains pays.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

Les termes qui suivent sont des marques d'autres sociétés :

Microsoft, Windows et Windows NT sont des marques de Microsoft Corporation dans certains pays.

Java, ou toutes les marques et logos incluant Java, et Solaris sont des marques de Sun Microsystems, Inc.

Tivoli et NetView sont des marques de Tivoli Systems Inc. dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos accompagnés de deux astérisques (**) qui pourraient apparaître dans ce document.

Index

A

ADD CONSTRAINT, instruction 57
Administration Guide vii
affichage
 informations en ligne 116
aide en ligne 114
ALL, utilisé dans une
 sous-requête 52
ALTER TABLE, instruction 57
ANY, mot clé 52
Application Development Guide vii
AS, clause 26
Assistant - Ajout d'une base de
 données 118, 119
Assistant - Configuration des
 performances 119
Assistant - Création d'un espace
 table 119
Assistant - Création d'une base de
 données 118
Assistant - Création d'une table 119
Assistant - Index 119
Assistant - Restauration de la base
 de données 119
Assistant - Sauvegarde de la base de
 données 118
Assistant Configuration de mise à
 jour multisite 118
Assistants
 ajout d'une base de
 données 118, 119
 configuration de mise à jour
 multisite 118
 configuration des
 performances 119
 création d'un espace table 119
 création d'une base de
 données 118
 création d'une table 119
 exécution de tâches 118
 index 119
 restauration de la base de
 données 119
 sauvegarde de la base de
 données 118
association, dans des requêtes 47

B

base de données relationnelle,
 définition 1
base de données relationnelle à
 noeuds multiples, définition 1
base de données relationnelle
 partitionnée, définition 1
base de données SAMPLE 77
 création 78
 effacement 79
BETWEEN, prédicat 51
Bibliographie vii
Bibliothèque DB2
 affichage en ligne 116
 aide en ligne 114
 assistants 118
 Centre d'aide et
 d'information 117
 commande de manuels
 imprimés 112
 configuration d'un serveur de
 documents 120
 identificateur de langue pour les
 manuels 110
 impression des manuels au
 format PDF 112
 informations de dernière
 minute 111
 manuels 99
 recherche en ligne 120
 structure 99
BIGINT, type de données 5
BLOB, chaîne 73
BLOB, type de données 73

C

catalogues système 75
Centre d'aide et d'information 117
chaîne
 LOB (objets de grande taille) 73
chaîne de caractères
 en tant que type de données 5
 longueur fixe 5
 longueur variable 5
chaîne graphique
 longueur fixe 5
 longueur variable 5
CHAR, type de données 5
classement des lignes 23

clé

 associée 56
 composée 56
 définition 56
 primaire 56
 unique 56
clé associée 56
clé composée 56
clé parente, définition 56
clé primaire 56
clé unique 56
 contrainte d'unicité 56
CLOB, chaîne 73
CLOB, type de données 73
colonne
 ASC, tri par ordre croissant 23
 définition 3
 DESC, tri par ordre
 décroissant 23
colonne, fonction 29
 AVG 29
 COUNT 29
 MAX 29
 MIN 29
colonne de groupage, définition 31
condition de jointure 62
condition de recherche 20
Configuration d'un serveur de
 documents 120
CONNECT, instruction 18
 explicite 18
 implicite 18
contrainte d'unicité 56
contraintes
 contraintes d'unicité 9
 contraintes référentielles 9
contraintes d'intégrité référentielle
 clé associée 56
 clé parente 56
 définition 55
 description 56
contraintes d'unicité
 définition 55
contraintes de vérification de table
 définition 55
 description 57
 vérification différée de
 contrainte 57

- conversion de données
 - conditions de jointure 64
 - opérateurs ensemblistes 49
- corrélation
 - description 40
 - nom 43
 - sous-requête 41
 - sous-requêtes utilisant des jointures 44
- CREATE DISTINCT TYPE 71
- CREATE FUNCTION 72
- CREATE TABLE, instruction 9
 - NOT NULL/NOT NULL WITH DEFAULT, valeur de colonne 9
- CREATE TRIGGER 58
- CREATE VIEW, instruction 13
 - WITH CHECK OPTION 13
- création de la base de données
 - SAMPLE 78
- CUBE 67
 - lignes à tabulations croisées 67
 - lignes de sous-totaux 67
- CURRENT DATE, registre spécial 74
- CURRENT FUNCTION PATH, registre spécial 74
- CURRENT SERVER, registre spécial 75
- CURRENT TIME, registre spécial 75
- CURRENT TIMESTAMP, registre spécial 75
- CURRENT TIMEZONE, registre spécial 75

D

- DATE, type de données 5
- DATETIME, type de données 5
- DBLOB, chaîne 73
- DBLOB, type de données 73
- décimal, description 5
- DECIMAL, type de données 5
- déclencheurs
 - CREATE TRIGGER 58
 - déclencheur AFTER 58
 - déclencheur BEFORE 58
 - définition 55
 - description 58
 - variables de transition 61
- DELETE, instruction 13
- DISTINCT, mot clé 24, 30
- distinct, type de données 71
- DOUBLE, type de données 5

E

- effacement de la base de données
 - SAMPLE 79
- élimination des lignes en double 24
- emplacement des objets LOB, définition 74
- entier binaire, description 5
- EXCEPT, opérateur 48
 - classement des résultats 49
 - limites d'utilisation de l'opérateur 49
 - types de données 49
- EXCEPT ALL 48
- EXISTS, prédicat 51
- expression CASE
 - description 36
 - fonction SIGN 36
- expression de table commune
 - description 39
- expressions 25
- expressions, noms 26
- expressions de table
 - description 38
- expressions de table imbriquées, description 38
- extraction de données 19

F

- FLOAT, type de données 5
- fonction
 - colonne 29
 - définie par l'utilisateur 29
 - description 28
 - intégrée 29
 - OLAP (traitement analytique en ligne) 68
 - scalaire 29
 - table 31
- fonction de table
 - SQLCACHE_SNAPSHOT 31
- fonction de table externe 72
- fonction de table externe OLE DB 72
- fonction dérivée 72
- fonction scalaire externe 72
- fonctions de colonne 29
- fonctions scalaires
 - ABS 30
 - HEX 30
 - LENGTH 30
 - SIGN 30
 - YEAR 30
- fonctions UDF 72
 - définition 72
 - fonction de table externe 72

- fonctions UDF 72 (*suite*)
 - fonction de table externe OLE DB 72
 - fonction dérivée 72
 - fonction scalaire externe 72
- FROM, clause 19
- FULLSELECT 35
 - ALL, mot clé 52
 - ANY, mot clé 52
 - avec instruction INSERT 10
 - sous-requête 10, 52
- FULLSELECT, définition 10
- FULLSELECT scalaires
 - description 35
- fusion des résultats de plusieurs requêtes 47

G

- gestionnaire de bases de données 1
- GROUP BY 25
- GROUP BY, clause
 - avec clause HAVING 32
 - colonne de groupe 31

H

- HAVING 25
- HAVING, clause
 - description 32
- HTML
 - programmes exemples 110

I

- ID autorisation 4
- identificateur de langue
 - manuels 110
- imbrication de sous-requêtes
 - corrélées 45
- impression des manuels au format PDF 112
- IN, prédicat 50
- informations de dernière minute 111
- informations en ligne
 - affichage 116
 - recherche 120
- INSERT, instruction 10
 - NOT NULL/NOT NULL WITH DEFAULT, valeur de colonne 10
- installation
 - navigateur Netscape 116
- INTEGER, type de données 5
- interpréteur de commandes 1
- INTERSECT, opérateur 49
 - classement des résultats 49

INTERSECT, opérateur 49 (*suite*)
limites d'utilisation de
l'opérateur 49
types de données 49
INTERSECT ALL 49

J

jointure
conditions de jointure 62
conversion de données 64
définition 26
produit croisé 62
sans conditions de jointure 62
sous-requêtes corrélées 44
jointure externe
description 62
jointure externe complète 62
jointure externe droite 62
jointure externe gauche 62
jointure externe complète 62
jointure externe droite 62
jointure externe gauche 62
jointure interne 62

L

langage de procédure SQL vii
ligne
définition 3
sélection 20
lignes de sous-totaux 67
lignes de tabulation croisée 67
LIKE, prédicat 51
liste de sélection 19
LOB
chaîne, définition 74
releveur de coordonnées,
définition 74

M

manuels 99, 112
messages d'erreur
identificateur de message 18
SQLSCODE 18
SQLSTATE 18
Mise en route vii
modèle de table CL_SCHED 79
modèle de table DEPARTMENT 79
modèle de table EMP_ACT 83
modèle de table EMP_PHOTO 85
modèle de table EMP_RESUME 85
modèle de table EMPLOYEE 80
modèle de table IN_TRAY 86
modèle de table ORG 86
modèle de table PROJECT 87
modèle de table SALES 88
modèle de table STAFF 89

modèle de table STAFFG 90
modèles de tables 77, 99

N

navigateur Netscape
installation 116
nom de corrélation
référence qualifiée à un nom de
colonne 40
règles 40
nombres, description 5
NOT BETWEEN, prédicat 51
NOT EXISTS, prédicat 51
NOT IN, prédicat 50
NOT LIKE, prédicat 51
NULL, description de la valeur 5
NULL, valeur 47
suppression d'une valeur de
colonne 12

O

OLAP, fonctions 68
classement des lignes 68
groupe d'agrégation 68
partitionnement des lignes 68
opérateur de comparaison utilisé
dans une sous-requête 52
opérateurs arithmétiques 25
ORDER BY, clause 23
opérateurs ensemblistes 49
ordre des opérations 25, 29

P

PDF 112
précision, en tant qu'attribut
numérique 5
prédicat
IS NOT NULL 20
IS NULL 20
prédicat de niveau supérieur 52
produit croisé 62
programmes exemples
HTML 110
multiplateformes 110

Q

qualification des objets 4, 17

R

REAL, type de données 5
recherche
informations en ligne 117, 120
référence corrélée, description 41
registre spécial 74
CURRENT DATE 74
CURRENT DEGREE 74

registre spécial 74 (*suite*)

CURRENT FUNCTION
PATH 74
CURRENT PATH 74
CURRENT SERVER 74
CURRENT TIME 75
CURRENT TIMESTAMP 75
CURRENT TIMEZONE 75
USER 75
relations entre les tables et les
vues 13
releveur de coordonnées 74
Remarques sur le produit 111
requête externe, corrélation 43
requêtes, association 49
requêtes récursives, description 67
restrictions
opérateurs ensemblistes 49
ROLLUP 67
lignes de sous-totaux 67

S

scalaire, fonction 29
DECIMAL 38
schéma
définition 4
schémas réservés 4
SELECT, instruction 19
SET, clause
avec instruction UPDATE 12
SET CONSTRAINTS, instruction 57
signe, en tant qu'attribut
numérique 5
SMALLINT, type de données 5
SmartGuides
assistants 118
SOME, mot clé 52
sous-requête
définition 28
sous-requête corrélée
description 41
quand utiliser 43
SQL (Langage d'interrogation
structuré), définition 1
SQL interactif, définition 1
SQL Reference vii
structure de données
colonne 3
ligne 3
valeur 3

T

table
base de données SAMPLE 77
clé associée 56

- table (*suite*)
 - clé primaire 56
 - clé unique 56
 - combinaison de données (jointure) 26
 - contrainte d'unicité 56
 - définition 3
 - fonctions 31
 - qualification d'un nom de colonne 40
 - table de base 3
 - table résultat 3
- table de base 3
- table résultat 3
- TIME, type de données 5
- TIMESTAMP, type de données 5
- traitement analytique en ligne (OLAP) 68
- transtypage de types de données
 - description 36
- type de données
 - distinct 71
- types de données
 - BIGINT 5
 - CHAR 5
 - DATE 5
 - DATETIME 5
 - DECIMAL 5
 - DOUBLE 5
 - FLOAT 5
 - INTEGER 5
 - REAL 5
 - SMALLINT 5
 - TIME 5
 - TIMESTAMP 5
 - VARCHAR 5

U

- UNION, opérateur 47, 48
 - classement des résultats 48
 - description 47
 - limites d'utilisation de l'opérateur 49
 - types de données 49
- UNION ALL 47
- UPDATE, instruction 12
- USER, registre spécial 75
- utilisation d'une vue pour modifier des tables 15
 - WITH CHECK OPTION 15

V

- valeur
 - définition 3
- valeur dans SQL 5

- valeurs utilisées pour l'horodatage, description 5
- VALUES, clause
 - avec instruction INSERT 10
- VARCHAR, type de données 5
- vérification de son existence 51
- vue
 - avantages 4
 - description 4
 - qualification d'un nom de colonne 40

W

- WHERE, clause 20
 - combinaison de données (jointure) dans une instruction SELECT 26
 - remarques sur le groupage 32
- WITH, clause 39
- WITH CHECK OPTION 15

Comment prendre contact avec IBM

Si votre question est d'ordre technique, étudiez tout d'abord les solutions présentées dans le manuel *Troubleshooting Guide* avant de prendre contact avec le Service clients DB2. Ce manuel indique les informations susceptibles d'aider le Service clients à mieux répondre à vos besoins.

Pour obtenir des informations ou commander des produits DB2 avant de prendre contact avec le Service clients DB2 Universal Database, prenez contact avec votre partenaire commercial IBM.

Aux États-Unis, composez l'un des numéros suivants :

- 1-800-237-5511 pour obtenir le Service clients,
- 1-888-426-4343 pour connaître les options de service disponibles.

Infos produit

Aux États-Unis, composez l'un des numéros ci-après.

- Pour commander des produits ou obtenir des informations générales, composez le 1-800-IBM-CALL (1-800-426-2255) ou 1-800-3IBM-OS2 (1-800-342-6672).
- Pour commander des manuels, composez le 1-800-879-2755.

<http://www.ibm.com/software/data/>

Les pages DB2 World Wide Web fournissent des informations sur DB2, des descriptions de produit, les programmes de formation et d'autres informations.

<http://www.ibm.com/software/data/db2/library/>

DB2 Product and Service Technical Library permet d'accéder à des forums Q&A (questions/réponses), d'obtenir des correctifs et les dernières informations techniques sur DB2.

Remarque : (Il est possible que ces informations ne soient disponibles qu'en anglais.)

<http://www.elink.ibm.com/pbl/pbl/>

Le site Web de commande internationale de manuels fournit les informations correspondantes.

<http://www.ibm.com/education/certify/>

Le programme Professional Certification Program du site Web IBM fournit des informations sur les tests de certification concernant différents produits IBM, dont DB2.

ftp.software.ibm.com

Établissez une connexion anonyme. Des démonstrations, des correctifs, des informations et des outils associés à DB2 ou à des produits connexes sont disponibles dans le répertoire /ps/products/db2.

comp.databases.ibm-db2, bit.listserv.db2-l

Ces newsgroups sont accessibles à tous ceux qui souhaitent partager leurs expériences sur les produits DB2.

Sur CompuServe : GO IBMDB2

Exécutez cette commande pour accéder aux forums IBM DB2. Tous les produits DB2 sont pris en charge sur ces forums.

En dehors des Etats-Unis, pour savoir comment prendre contact avec IBM, consultez l'annexe A du manuel **IBM Software Support Handbook**. Pour accéder à ce document, allez sur le site Web :

<http://www.ibm.com/support/>, puis effectuez une recherche sur le mot clé «handbook».

Remarque : Dans certains pays, les distributeurs agréés peuvent contacter leur centre d'assistance au lieu de prendre contact avec le centre de support IBM.



Référence: CT7YHFR

SC11-1655-00



CT7YHFR

