DB2 for z/OS Security:
New Ways to Protect Your Assets

Gayathiri Chandran
IBM Silicon Valley Laboratory
gchandran@us.ibm.com

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

IBM, the IBM logo, ibm.com, z/OS, RACF and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at

- "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml
- Other company, product, or service names may be trademarks or service marks of others.

Agenda

- Trusted context and roles
- Granular Administrative Authorities
- Row and column level access controls
- External Security Enhancements
- Audit - Audit policies
- Audit - Temporal tables
- Summary

COMPLIANCE IS A
CONTINUOUS PROCESS

DB2 9: Trusted context and Role

- Better access control from application servers.
- Allows connections to be established as today. Application attributes are verified before associating it with a trusted context such as the application id and where the request originated
- Supports identity propagation allowing authenticated non z/OS distributed IDs to flow to DB2 to be included in audit logs
- Allows a unique set of privileges by use of a Role to be associated with an application, preventing the misuse of privileges when not accessing through the application
- Provides flexibility by removing object dependency from users
- Addresses administrator challenges

Trusted Context

- Trusted context establishes trust between DB2 and an external entity such as
  - RRSAF (Resource Recovery Services Attachment Facility)
  - DSN Command Processor
  - Application Server
- Once established, a trusted connection provides the ability to
  - Efficiently switch user with optional authentication
  - Acquire special set of privileges using a Role
  - Acquire special RACF Security Label authority
- Manage trusted context using SQL CREATE / ALTER / DROP TRUSTED CONTEXT

Database Role

- Database entity with one or more privileges
- Established only through a trusted connection
- User assigned only one role in a trusted connection
- Can optionally be the OWNER of DB2 objects
- Manage role using SQL CREATE / DROP ROLE

```
CREATE ROLE ADMINROLE;

DB2 native authorization – ROLE keyword for GRANTEE:
GRANT SYSADM TO ROLE ADMINROLE;

RACF exit authorization – CRITERIA(SQLROLE) keyword:
PERMIT DSNADM SUBSYS.SYSADM ID(ADMINA)
       WHEN(CRITERIA(SQLROLE(ADMINROLE)))
```

Trusted context - Local

- Trusted context can be local or remote
- Local trusted context is based upon
  - System Authid
    - User ID associated with the connection
  - JOBNAME
    - Job or started task name associated with the connection

```
Example: Assign a role DBAROLE to any job named ADMINJOB that
connects using auth ID SALLY

CREATE ROLE DBAROLE;

CREATE TRUSTED CONTEXT DBACONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SALLY
   ATTRIBUTES JOBNAME('ADMINJOB')
   DEFAULT ROLE DBAROLE
   ENABLE;
```

Trusted Context - Remote

- Remote trusted context is based upon
  - System Authid
    - User ID associated with the connection
  - ADDRESS
    - Client's IP address, domain name or SERVAUTH security zone name of the connection
  - ENCRYPTION
    - Connection encryption level (NONE | LOW | HIGH)

```
Example: Assign a role TELLER to a connection established from
IP address 9.10.10.120 and the auth ID SRVRID01.

CREATE ROLE TELLER;

CREATE TRUSTED CONTEXT TELLERCONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SRVRID01
   ATTRIBUTES ADDRESS('9.10.10.120')
   DEFAULT ROLE TELLER
   ENABLE;
```

Trusted Context Auth ID Switching

- Allows trusted connection to be used by different users
- Optional authentication requirement
- Specific ROLE and RACF Security Label can be assigned to the user

```
Example: Assign a role TELLER to a connection established from
IP address 9.10.10.120 and the auth ID SRVRID01. Allow MARY and
JOHN to use the connection.

CREATE TRUSTED CONTEXT TELLERCONTEXT
   BASED UPON CONNECTION USING SYSTEM AUTHID SRVRID01
   ATTRIBUTES ADDRESS('9.10.10.120')
   DEFAULT ROLE TELLER
   WITH USE FOR MARY, JOHN
   ENABLE;
```

Trusted Context Auth ID Switch options

- Switch user options:
  - Authorization name
  - EXTERNAL SECURITY PROFILE Profile-name
    - DB2 primary authorization id or one of their groups has to be permitted to use the specified profile.
  - PUBLIC
- Distributed Identity
  - Exploits RACF distributed identity mapping capability
  - RACF RACMAP command is used to associate a distributed ID to a DB2 RACF user ID.

```
Example: Map distributed user ID, APPUSR01 to RACF ID, RACFID1
using Registry name, USERREGISTRY01.

SETROPTS CLASSACT(IDIDMAP) RACLIST(IDIDMAP)
RACMAP ID(RACFID1) MAP -
  USERDIDFILTER(NAME('APPUSR01')) -
  REGISTRY(NAME('USERREGISTRY01'))
SETR RACLIST(IDIDMAP) REFRESH
```

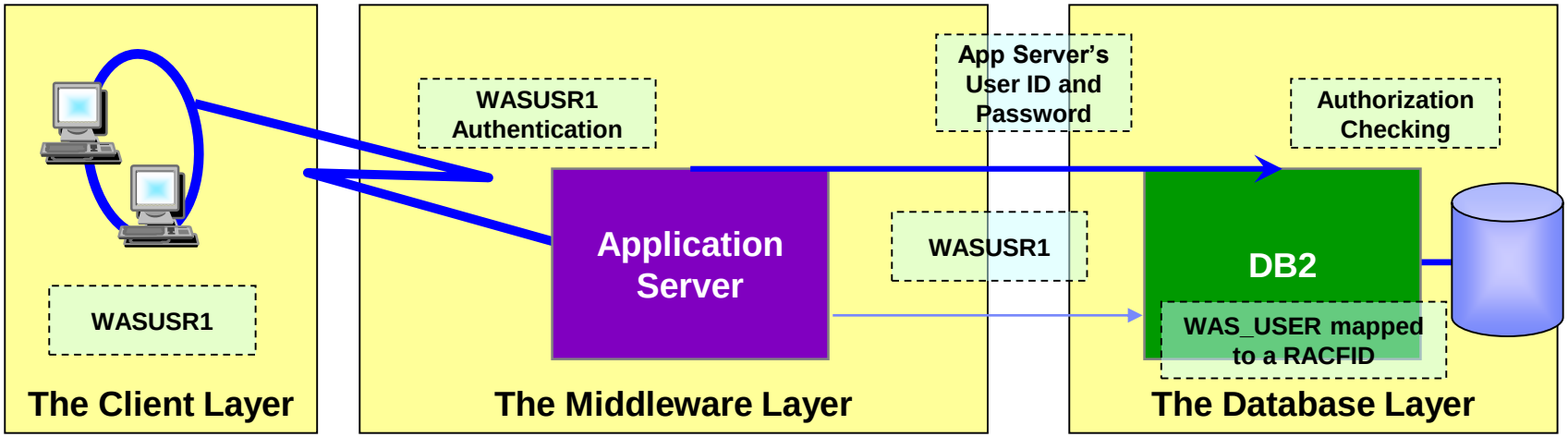
Trusted Context RACF support

- When DB2's native authorization mechanisms are used, RACF is completely uninvolved in the access control decision
- When RACF is used to control access to DB2 objects access is permitted using the CRITERIA keyword on the PERMIT command:

```
PERMIT DSNADM DSND.SYSADM ID(MARKN) WHEN(CRITERIA(SQLROLE(SysProg))
PERMIT DSNADM DSND.SYSADM ID(*)     WHEN(CRITERIA(SQLROLE(SysProg)))
```

  - Warning: The SQLROLE value is a mixed case value!
    - "SysProg" ^= "SYSPROG" ^= "sysprog" !

Trusted connections provide more effective controls and accurate audit trail for remote access



- The application server's user ID and password are used to establish the trusted connection
- The user is switched in the trusted connection and client user ID is propagated to the server and checked for database access
- RACF distributed identities allows to map client user ID to RACF user ID
  - End user identity not used for privilege checking, but contained in both RACF and DB2 audit logs
- Certificate authentication eliminates the exposure to RACF IDs and passwords on distributed platforms

New improved security features provide more effective controls and accurate audit trail for remote access

- Support client certificate authentication in z/OS V1R10
  - AT-TLS secure handshake accomplishes identification and authentication for client certificates
  - DB2 client driver presents its certificate as identification and its *proof-of-possession* as authentication
  - DB2 server can retrieve the user ID associated with the client certificate in SAF for the AT-TLS policy rule configuration:
    HandshakeRole = ServerWithClientAuth, ClientAuthType = SAFCheck
  - RACF certificate name filtering (RACDCERT MAP command) can map many certificates with one RACF userid
- Support password phrases in z/OS V1R10
  - A RACF password phrase is a character string made up of mixed-case letters, numbers, special characters, and is between 9 to 100 characters long
  - Can be used instead of a traditional 8-character password

DB2 10: New Granular System Authorities

Prior to DB2 10

- SYSADM
- DBADM
- DBCTRL
- DBMAINT
- SYSCTRL
- PACKADM
- SYSOPR

Granular Authorities in DB2 10

- System DBADM
- ACCESSCTRL
- DATAACCESS
- SECADM
- SQLADM
- EXPLAIN

System DBADM: Allows management of objects

DATAACCESS: Access to data in all user tables

ACCESSCTRL: Controls access to data

SECADM: Performs security related tasks

SQLADM: Allows monitoring and tuning without access to data

DB2 10: Install Security Parameters

- **Prevents SYSADM and SYSCTRL from granting or revoking privileges**
  - Install parameter, **SEPARATE_SECURITY**
  - Install **SECADM** authority manages subsystem security
  - SYSADM and SYSCTRL can no longer implicitly grant or revoke privileges
  - Install SYSADM authority not impacted
- **Control cascading effect of revokes**
  - Install parameter, **REVOKE_DEP_PRIVILEGES**
  - **INCLUDING / NOT INCLUDING DEPENDENT PRIVILEGES** clause can be specified on SQL REVOKE statements

RACF Support for the New Administrative Authorities

- RACF Access Control Module has been modified to honor the setting of SEPARATE_SECURITY
- New DB2 System Privilege Checks

DB2 Authority	Resource	Class
SECADM	<subsystem>.SECADM	DSNADM
System DBADM	<subsystem>.SYSDBADM	DSNADM
DATAACCESS	<subsystem>.DATAACCESS	DSNADM
ACCESSCTRL	<subsystem>.ACCESSCTRL	DSNADM
SQLADM	<subsystem>.SQLADM	MDSNSM
EXPLAIN	<subsystem>.EXPLAIN	MDSNSM

DB2 10: Row and Column Access Controls

New table controls to protect against unplanned SQL access

- Define additional data controls at the row and column level
  - Security policies are defined using SQL
  - Separate security logic from application logic
- Security policies based on real time session attributes
  - Protects against SQL injection attacks
  - Determines how column values are returned
  - Determines which rows are returned
- No need to remember various view or application names
  - No need to manage many views; no view updates or audit issues
- All access via SQL including privileged users, adhoc query tools, report generation tools is protected
- Policies can be added, modified, or removed to meet current company rules without change to applications

Table controls to protect SQL access to individual row level

- Establish a row policy for a table
  - Filter rows out of answer set
  - Policy can use session information, e.g. the SQL ID is in what group or user is using what role, to control which row is returned in result set
  - Applicable to SELECT, INSERT, UPDATE, DELETE, & MERGE
  - Defined as a row permission:

CREATE PERMISSION policy-name ON table-name
FOR ROWS WHERE search-condition
ENFORCED FOR ALL ACCESS ENABLE;

Table controls to protect SQL access to individual column level

- Establish a column policy for a table
  - Mask column values in answer set
  - Policy can use session information, e.g. the SQL ID is in what group or user is using what role, to control what masked value is returned in result set
  - Applicable to the output of outermost subselect
  - Defined as column masks :

CREATE MASK mask-name ON table-name
FOR COLUMN column-name RETURN CASE-expression
ENABLE;

Define table policies based on who or how the table is being accessed

- SESSION_USER - Primary authorization ID of the process
- CURRENT SQLID - SQL authorization ID of the process
- VERIFY_GROUP_FOR_USER function
  - Get the authorization IDs for the value in SESSION_USER
  - Returns 1 if any of those authorization IDs is in the argument list

```
WHERE
   VERIFY_GROUP_FOR_USER (SESSION_USER, 'MGR', 'PAYROLL') = 1
```

- VERIFY_ROLE_FOR_USER function
  - Get the role for the value in SESSION_USER
  - Return 1 if the role is in the argument list

```
WHERE
   VERIFY_ROLE_FOR_USER (SESSION_USER, 'MGR', 'PAYROLL') = 1
```

Managing row and column access controls

- When activated row and column access controls:
  - All row permissions are connected with ‘OR’ to filter out rows
  - All column masks are applied to mask output
  - All access to the table is prevented if no user-defined row permissions
- When deactivated row and column access controls:
  - Opens all access to the table

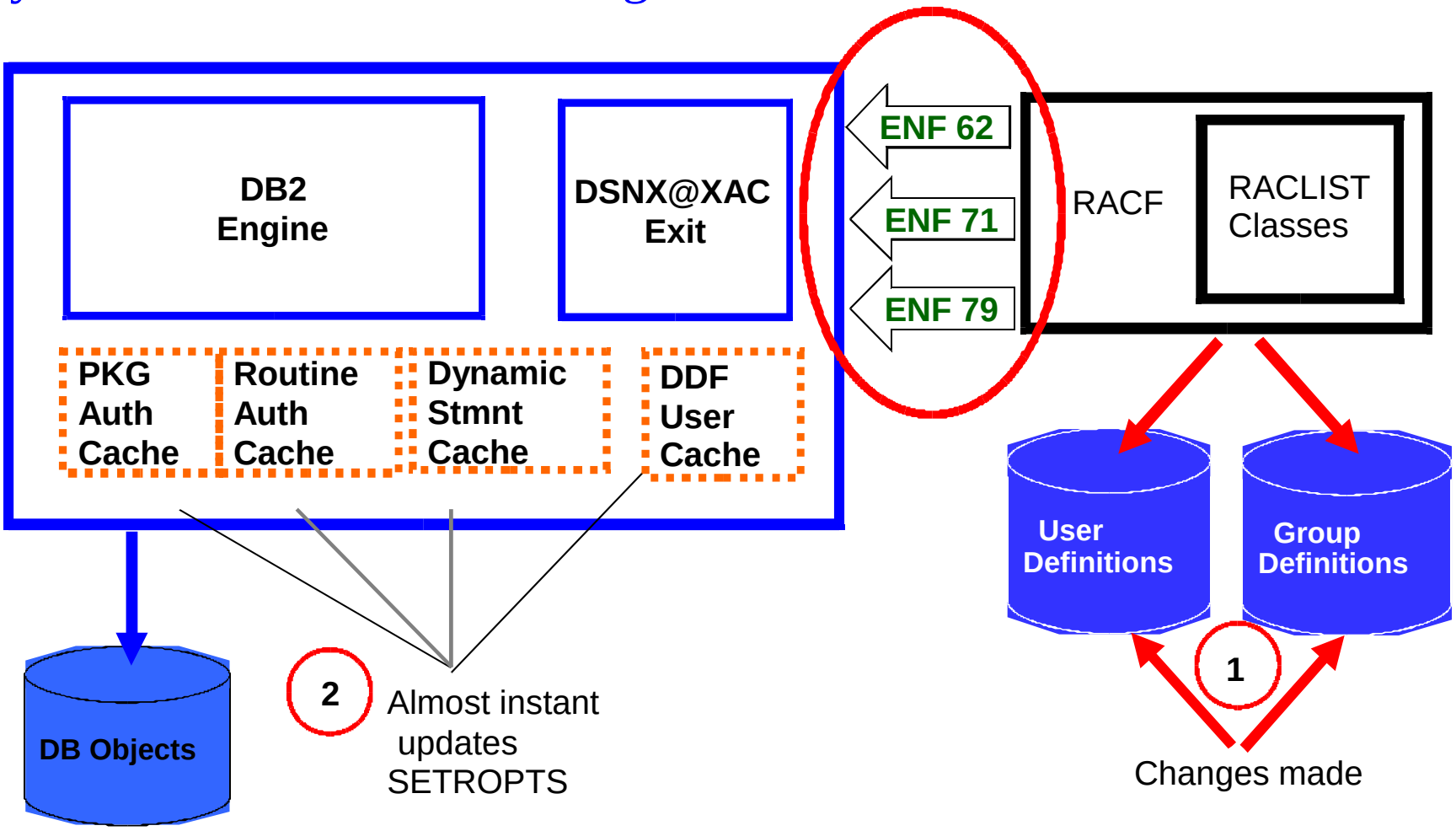
```
ALTER  TABLE   table-name
   ACTIVATE ROW       ACCESS  CONTROL
   ACTIVATE COLUMN  ACCESS CONTROL;

ALTER  TABLE   table-name
   DEACTIVATE ROW      ACCESS  CONTROL
   DEACTIVATE COLUMN ACCESS CONTROL;
```

DB2 11: External Security (DSNX@XAC) consistency with DB2 Security – Owner Authorization

- Support OWNER privileges for authorization
  - Allows owner to be checked for authorization on BIND and REBIND commands
  - Supports dynamic SQL authorization using DYNAMICRULES behavior
    - Package owner
    - ID that executes the package
    - ID that defined the routine
    - ID that invokes the routine
  - Allows automatic rebind (AUTOBIND)
  - Owner can be a RACF ID, GROUP or ROLE. DB2 provides owner ACEE to RACF
  - Similar behavior between DB2 native and RACF exit authorization
  - New installation parameter, AUTHEXIT_CHECK is used to govern owner authorization

DB2 11: External Security (DSNX@XAC) Enhancements
Sync RACF Permission Changes to DB2 Cache



RACF ENF Signals Heard by DB2

- RACF Event Notifications (ENF)
  - Notifications generated by RACF when a profile is changed
- DB2 11 listens for:
  - ENF 62: RACF options refreshed
    - SETROPTS RACLIST REFRESH
  - ENF 71: User permissions changed
    - ALTUSER REVOKE, CONNECT REVOKE, DELUSER, DELGROUP, REMOVE
  - ENF 79: User permissions to access resource changed
    - PERMIT..DELETE, ACCESS(NONE), RESET, WHEN(CRITERIA(SQLROLE...))
    - RALTER.. UACC(NONE), DELMEM; RDELETE
    - On receipt of ENF 79, DB2 stores the changes and refreshes cache entries only when ENF 62 is heard
    - Requirement: RACF class descriptor table must have SIGNAL = YES
      - Enabled for IBM supplied RACF resource classes for DB2
- New installation parameter, AUTHEXIT_CACHEREFRESH is used to govern cache refresh
- Cache refresh considerations link:

  http://www-01.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc.seca/src/tpc/db2z_engsignalprocessing.dita

DB2 V11 – RACF exit enhancements

- Support new Global Variables privileges

DB2 Privilege	Resource	Class
READ	<subsystem>.schema-name.variable-name	MDSNGV
WRITE	<subsystem>. schema-name.variable-name	MDSNGV

- Provide RACLISTed classes at DB2 start in the new XAPLCLST field
- Remove RC8 and reason code 17 on AUTOBIND for UDFs

Auditing in DB2

- Who is privileged to access what data?
  - Most of the catalog tables describe the DB2 objects, such as tables, views, table spaces, packages, and plans
  - If using DB2 native authorization, several other tables (every table with the character string "AUTH" in its name) hold records of every granted privilege or authority.
- Who accessed what data?
  - You can find answers by using the audit trace, another important audit trail for DB2

Audit Trace Records

- DB2 uses SMF and/or GTF and/or monitor program for trace data
- Trace types
  - -Accounting -Audit -Monitor -Performance -Statistics
- Audit - Selective tracing with 11 classes of information
  - Access denials
  - Authorization changes
  - Changes to the structure of data (such as dropping a table)
  - Changes to data values (such as updating or inserting records)
  - Reading of data values (such as select)
  - Changes in authorization IDs
  - Utilities changes
  - Trusted context information
  - Audit Administrative Authorities

```
-START TRACE (AUDIT) CLASS (4,6) DEST (GTF) LOCATION (*)
```

DB2 10: Audit Policies

- New Audit policy allows you to comply without the need of external collectors. Managed in the DB2 catalog.

- Auditor can define an audit policy to audit any access to specific tables for specific programs during day
  - Audit policy does not require AUDIT clause to be specified using DDL
  - Audit policy generate records for all SQL read and update access
  - Audit policy includes additional records identifying the specific SQL statements
  - Audit policy provides wildcarding of based on table names

- Auditor can define an audit policy to identify any unusual use of a privileged authority
  - Records each use of an administrative authority
  - Audit records written only when authority is used for access

How to exploit Audit policies

- Security administrator using the new SECADM authority maintains DB2 audit policies in a new catalog table
  - SYSIBM.SYSAUDITPOLICIES
- Audit policies enabled using –STA TRACE command
- Audit policies disabled using –STO TRACE command
- Up to 8 audit policies can be specified to auto start or auto start as secure during DB2 start up
- Only user with SECADM authority can stop a secure audit policy trace
- Supports 8 categories
  - CHECKING, VALIDATE, OBJMAINT, EXECUTE
  - CONTEXT, SECMAINT, SYSADMIN, DBADMIN

Example: Dynamic auditing of tables

- Audit all the tables that start with ‘PAY’ in EMPLOYEE schema
  - Does not require AUDIT clause to be specified during table definition

```
INSERT INTO SYSIBM.SYSAUDITPOLICIES (AUDITPOLICYNAME, OBJECTSCHEMA,
OBJECTNAME, OBJECTTYPE, EXECUTE)
  VALUES ('TABADT1',‘EMPLOYEE','’’PAY%’’','T','A');

-STA TRACE (AUDIT) DEST (GTF) AUDTPLCY(TABADT1);
```

Example – Audit privileged authority

- Audit successful execution of all actions using installation SYSADM authority and system DBADM authority

```
INSERT INTO SYSIBM.SYSAUDITPOLICIES
  (AUDITPOLICYNAME, SYSADMIN, DBADMIN)
 VALUES (‘AUDITADMIN',‘I',‘B');

-STA TRACE (AUDIT) DEST (GTF) AUDTPLCY(AUDITADMIN);
```

DB2 10: Temporal table

DB2 can now manage different versions of your data

- Temporal table allows DB2 to automatically maintain different versions of your data
- Two types of time sequences of table rows are supported through the introduction of database defined time periods
  - SYSTEM_TIME is used to support data “versioning” which archives old rows into a history table
  - BUSINESS_TIME is a period that represents when a row is valid to the user or application
  - BITEMPORAL table combines SYSTEM_TIME period and BUSINESS_TIME period

Defining system period on an existing table

- System versioning is implemented by altering an existing or creating a table with two timestamps, a history table, and defining the versioning relationship between tables

```
CREATE TABLE POLICY_INFO
 (POLICY_ID CHAR(10) NOT NULL,
  COVERAGE INT NOT NULL,
  SYS_START TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN,
  SYS_END TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END,
  CREATE_ID TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID,
  PERIOD SYSTEM_TIME(SYS_START,SYS_END));

CREATE TABLE HIST_POLICY_INFO
 (POLICY_ID CHAR(10) NOT NULL,
  COVERAGE INT NOT NULL,
  SYS_START TIMESTAMP(12) NOT NULL,
  SYS_END TIMESTAMP(12) NOT NULL,
  CREATE_ID TIMESTAMP(12));
```

Defining system period on an existing table

- After the base and history tables are appropriately defined:
  - ALTER TABLE table-name ADD VERSIONING is specified on the base table that is to be versioned
- Auditor can query historical data through SQL
  - DB2 rewrites the user's query to include data from the history table

```
ALTER TABLE POLICY_INFO
 ADD VERSIONING USE HISTORY TABLE HIST_POLICY_INFO;
```

DB2 11: Audit change data

- DB2 V10 system versioning feature provided an auditing solution to track WHEN the data is modified.
- DB2 V11 (APARs PM99683/PI15298/PI15666) provides support to track
  - WHO modified the data
  - WHAT action caused the data modification
  - Supported on SQL CREATE and ALTER TABLE statements using new GENERATED ALWAYS AS clause

Audit change data – WHO modified data

- **Special registers** can be specified using GENERATED ALWAYS AS clause to audit WHO modified data
  - CURRENT CLIENT_ACCTNG VARCHAR(255)
  - CURRENT CLIENT_APPLNAME VARCHAR(255)
  - CURRENT CLIENT_CORR_TOKEN VARCHAR(255)
  - CURRENT CLIENT_USERID VARCHAR(255)
  - CURRENT CLIENT_WRKSTNNAME VARCHAR(255)
  - CURRENT SERVER CHAR(16)
  - CURRENT SQLID VARCHAR(8)
  - SESSION_USER or USER VARCHAR(128)

Example using SESSION_USER special register:

```
CREATE TABLE Bank_Account
(Account_Num INT  NOT NULL,
 Balance INT,
 USER_ID VARCHAR(128) GENERATED ALWAYS AS (SESSION_USER));
```

Audit change data – WHO modified data

- **Session Variables** can be specified using GENERATED ALWAYS AS clause to audit the process that modified data
  - SYSIBM.PACKAGE_NAME VARCHAR(128)
  - SYSIBM.PACKAGE_SCHEMA VARCHAR(128)
  - SYSIBM.PACKAGE_VERSION VARCHAR(122)

Example using Session Variables:

```
CREATE TABLE Bank_Account
(Account_Num INT  NOT NULL,
 Balance INT,
 Package_Name VARCHAR(128) GENERATED ALWAYS AS (SYSIBM.PACKAGE_NAME));
```

Audit change data – WHAT action caused data change

- New DATA CHANGE OPERATION keyword can be specified using GENERATED ALWAYS clause to audit the action that caused data modification
  - DB2 generates one of the following values:
    - I – Insert operation
    - U – Update operation
    - D – Delete operation

Example for DATA CHANGE OPERATION:

```
CREATE TABLE Bank_Account
(Account_Num INT  NOT NULL,
 Balance INT,
 USER_ID VARCHAR(128) GENERATED ALWAYS AS (SESSION_USER),
 OPERATION CHAR(1) GENERATED ALWAYS AS (DATA CHANGE OPERATION));
```

Summary

- Trusted connections provide better user accountability and improved compliance.
- Granular administrative authorities help reduce data exposure for administrators
- Row and column access table controls to safe guard your data
- Access Control Authorization Exit enhancements provide consistent security model and improved RACF integration
- Auditing features using audit policies provide better auditing capabilities
- Temporal data to comply with regulations to maintain historical data

References

- Security Functions of IBM DB2 10 for z/OS (SG24-7959-00)
  - http://www.redbooks.ibm.com
- DB2 10 for z/OS Technical Overview (SG24-7892-00)
  - http://www.redbooks.ibm.com
- DB2 11 for z/OS Managing Security (SC19-4061)
  - http://www-01.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc/src/alltoc/db2z_msecuhome.dita?lang=en
- DB2 11 for z/OS RACF Access Control Module Guide (SC19-4065)
  - http://www-01.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/com.ibm.db2z11.doc.racf/src/racf/db2z_racf.dita?lang=en
- DB2 10 for z/OS: Configuring SSL for Secure Client-Server communications - Red paper
  - http://www.redbooks.ibm.com/redpieces/abstracts/redp4799.html?Open
- DB2 for z/OS Information Center
  - http://www-01.ibm.com/support/knowledgecenter/?lang=en