

|  |
| --- |
|  |
| **Emergent Investment Bank** |
|  |
| **Transaction Security System (TSS)  Bright Star Consulting**  **Information Technology Program, University of Maryland University College**  **ITEC DBST 668 Section 9040**  **Faculty Member: Dr. Michael Martin**  **Faculty Member Assistant: Haroon Y Serang** |
|  |
| **Authors** |

**Table of Contents**

[INTRODUCTION: EXECUTIVE SUMMARY 1](#_Toc341539835)

[PROJECT TIMELINE 2](#_Toc341539836)

[CONCEPTUAL MODEL 3](#_Toc341539837)

[FUNCTIONAL ROLES 4](#_Toc341539838)

[ER DIAGRAM 4](#_Toc341539839)

[LOGICAL MODEL 5](#_Toc341539840)

[ENTITY RELATIONSHIP DIAGRAM 7](#_Toc341539841)

[PROJECT ASSUMPTIONS 8](#_Toc341539842)

[TSS PHYSICAL MODEL – DDL SCRIPT: TSS.SQL 9](#_Toc341539843)

[DATABASE DLL OUTPUT LOG 21](#_Toc341539844)

[OLS\_TSS02 SECURITY – SECURITY DDL - SCRIPT: SEC.SQL 40](#_Toc341539845)

[OLS\_TSS02 - SECURED DATA DML - SCRIPT: TSSDATA.SQL 46](#_Toc341539846)

[DATABASE DATA ENTRY OUPUT LOG 48](#_Toc341539847)

[DATABASE DATA REVIEW LOG 52](#_Toc341539848)

[OLS\_TSS02 - AGENTS SECURITY TEST - SCRIPT TSEC\_AG.SQL 54](#_Toc341539849)

[AGENT SECURITY TEST OUTPUT LOG 56](#_Toc341539850)

[OLS\_TSS02 – ACM SECURITY TEST - SCRIPT TSEC\_ACM.SQL 60](#_Toc341539851)

[ACCOUNT MANAGER SECURITY TEST OUTPUT LOG 63](#_Toc341539852)

[OLS\_TSS02 – BM SECURITY TEST - SCRIPT TSEC\_BM.SQL 68](#_Toc341539853)

[BANK MANAGER SECURITY TEST OUTPUT LOG 71](#_Toc341539854)

INTRODUCTION: EXECUTIVE SUMMARY

Emergent Investment Bank is a growing financial institution in the state of Maryland. The bank is recognized as a leader in providing investment services to its customers. The growing success of ECB has revealed significant issues related to data security. The Chief Security Officer (CSO) is concerned that the current security implementation, based on Fine-Grained Access Controls (FGAC) with views, is increasingly complex to manage. The security management overhead for agents, account managers and bank managers has put significant pressure on the bank security budget. While views are a popular method of implementing security, as the bank grows, the need to mix security policies with different DML statements increases complexity (Knox, 2004). The need to create different views for agents, account managers, and bank managers to manage more complex roles and policies is a problem in need of a solution.

Bright Star Consulting group has identified a solution to address the CSO’s concern about the issues discovered in our completed business analysis. The bank will need to implement new database security architecture to secure critical data as the bank grows. The Transaction Security System (TSS) is a database security implementation that will be deployed based on Oracle’s Oracle Label Security (OLS) technology. The roles and policies will be implemented based on table and row level access. TSS is well suited for data labeling, roles, user authorizations, and database objects read and write privilege management. EIB’s key users are agents, account managers, and bank managers. Each will have defined access levels and privileges on database objects.

**Diagram Tool:** Oracle SQL Developer Data Modeler version 3.0.0.665. Platform Dell/Windows XP

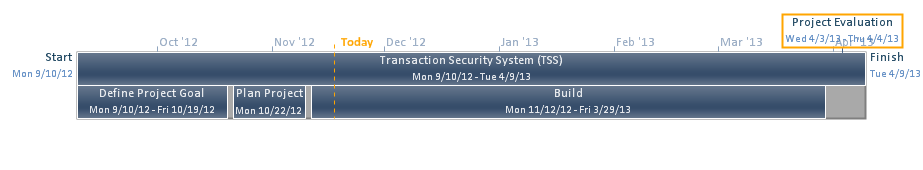
**Database:** Oracle 11g using Nova.

**Hardware and Software:**

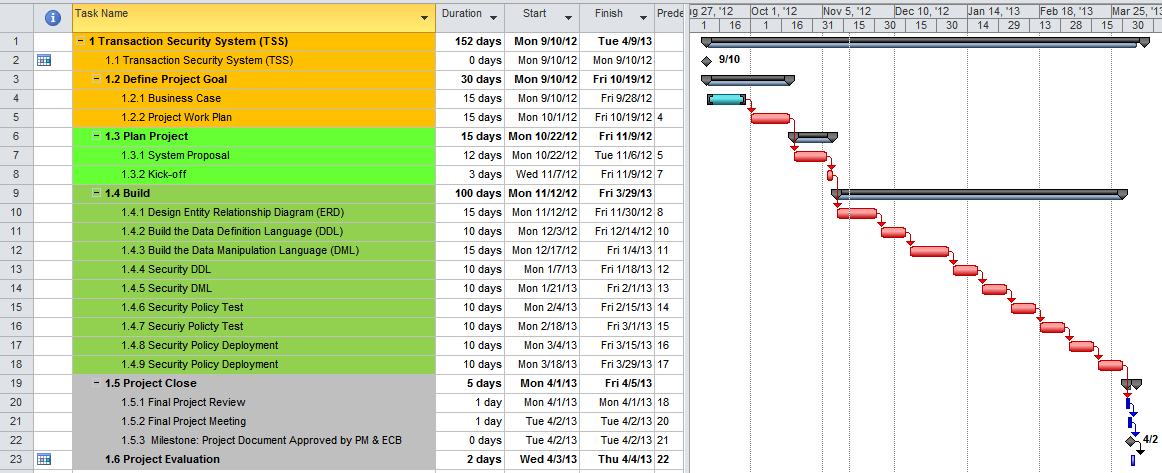
UMUC server system Nova, Operating system is UNIX. Sun Solaris Connect to Oracle via PortaPutty.

Oracle SQL Developer version 3.0.04.

PROJECT TIMELINE



**GANT CHART**



CONCEPTUAL MODEL

### FUNCTIONAL ROLES

**Agents**

Users who provide personal customer service to low-net-worth account holders. Agents use a banking application to check customer balances, record deposits, withdraw funds, and update customer information for a specific type of customer.

Requirements: SELECT, UPDATE, INSERT

Objects: TBL\_CUSTOMER, TBL\_ACCOUNT, TBL\_TRANSACTION,TBL\_ACCT\_TYPE,

TBL\_TRANS\_TYPE

**Account Managers**

Users who provide personal customer service to high-net-worth account holders. Agents use a banking application to check customer balances, record deposits, withdraw funds, and update customer information for a specific type of customer.

Additionally, they are able to add new customers and accounts for low and high-net-worth clients.

AMs have the privilege to manage low-net-worth accounts and deactivate accounts. The restriction is a business rule to reduce security audit complexity.

**Requirements**: SELECT, UPDATE, INSERT

**Objects**: TBL\_CUSTOMER, TBL\_ACCOUNT, TBL\_TRANSACTION,TBL\_ACCT\_TYPE,

TBL\_TRANS\_TYPE, TBL\_ACCESS

**Bank Managers**

Users who provide support to Agents and AMs with account privileges as super users of the transaction security system. They are able to assign roles and privileges to Agents and AMs in addition to be being able to perform all the tasks Agents or Account Managers are able to perform. Bank managers are the only users allowed to delete records in the system.

**Requirements**: SELECT, UPDATE, INSERT, DELETE

**Objects**: TBL\_CUSTOMER, TBL\_ACCOUNT, TBL\_TRANSACTION, TBL\_ACCT\_TYPE,

TBL\_TRANS\_TYPE, TBL\_ACCESS

### ER DIAGRAM

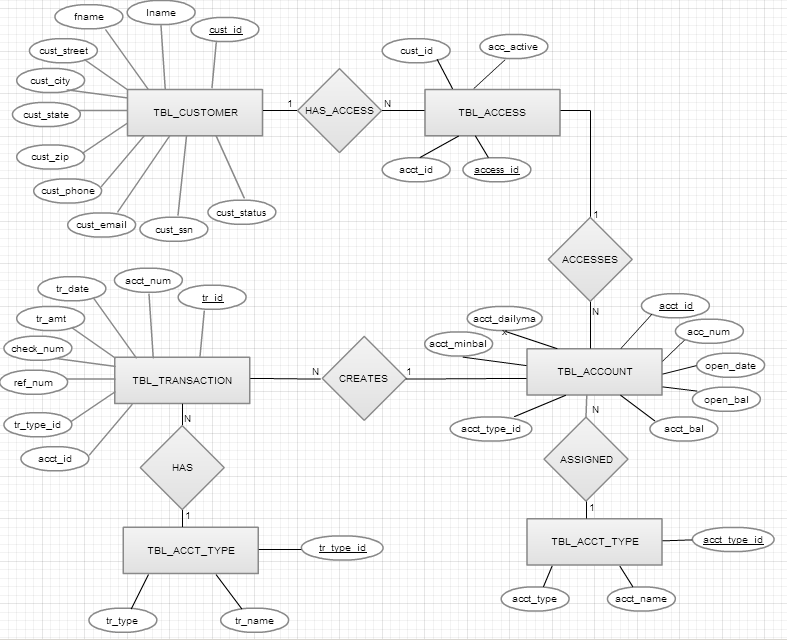


Figure 1: tss er diagram

LOGICAL MODEL

**RULES TO MAP ER DIAGRAMS TO RELATIONS**

The applied rules are based on the discussion on the fundamentals of database systems by (Elmasri & Navathe, 2011).  **Referential Integrity** – rule that states that any foreign key value (on the relation of the many side) MUST match a primary key value in the relation of the one side.

**One-to-Many -** Primary key on the one side becomes a foreign key on the many side

**Many-to-Many -** Create a new relation with the primary keys of the two entities as its primary key

**Mapping Regular Entities to Relations**

1. **Simple attributes** – E-R attributes map directly onto the relation
2. **Composite attributes** - Use only their simple, component attributes
3. **Multivalued Attribute -** Becomes a separate relation with a foreign key taken from the superior entity

TBL\_CUSTOMER

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cust\_id | lname | fname | cust\_steet | cust\_city | cust\_state | cust\_zip | cust\_phone | cust\_email | cust\_ssn | cust\_type | cust\_status |

TBL\_ACCESS

|  |  |  |  |
| --- | --- | --- | --- |
| access\_id | cust\_id | acct\_num | acc\_active |

TBL\_ACCOUNT

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| acct\_id | acct\_num | open\_date | open\_bal | acct\_bal | acct\_minbal | close\_date | acct\_active | acct\_type\_id |

TBL\_ACCT\_TYPE

|  |  |
| --- | --- |
| acct\_type\_id | acct\_name |

TBL\_TRANSACTION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tr\_id | acct\_num | tr\_date | tr\_amt | check\_num | ref\_num | tr\_type\_id |

TBL\_TRANS\_TYPE

|  |  |  |
| --- | --- | --- |
| tr\_type\_id | tr\_type | tr\_name |

**Mapping Associative Entities**

1. **Identifier Not Assigned** 
   1. Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
2. **Identifier Assigned** 
   1. It is natural and familiar to end-users
   2. Default identifier may not be unique

TBL\_CUSTOMER

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cust\_id | lname | fname | cust\_steet | cust\_city | cust\_state | cust\_zip | cust\_phone | cust\_email | cust\_ssn | cust\_status |

TBL\_ACCESS

|  |  |  |  |
| --- | --- | --- | --- |
| access\_id | cust\_id | acct\_id | acc\_active |

TBL\_ACCOUNT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| acct\_id | acc\_num | open\_date | open\_bal | acct\_bal | acct\_dailymax | acct\_minbal | acct\_type\_id |

### ENTITY RELATIONSHIP DIAGRAM

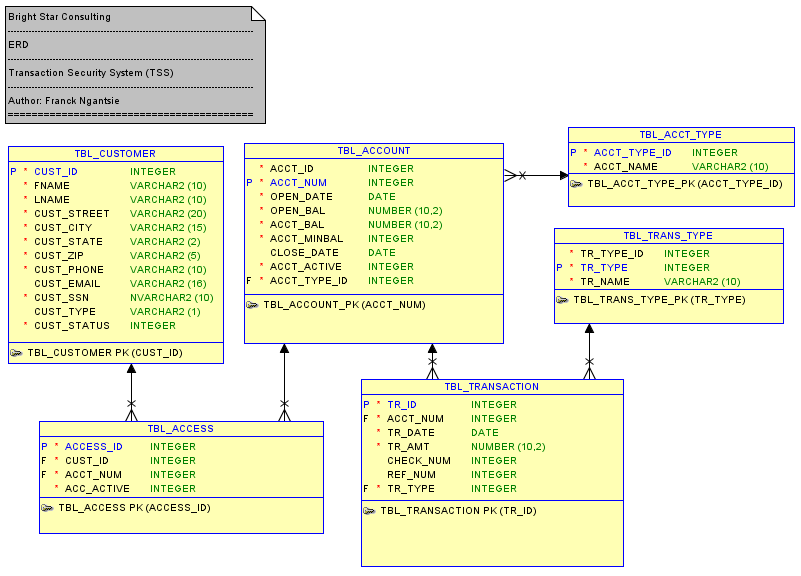


Figure 2: Entity Relationship Details

### PROJECT ASSUMPTIONS

**Assumption#1**: the account manager determines which level of security should be assigned to a client based on predefined internal rules.

**Assumption#2**: this project strictly assumes that all clients walk into a bank branch for all activities

**Assumption#3**: executives want security to be role and classification based. Security should be applied throughout the organization with levels 1, 2, and 3 assigned to all data with level 1 being the least privileged and level 3 the most privileged.

TSS PHYSICAL MODEL – DDL SCRIPT: TSS.SQL

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--BRIGHT STAR CONSULTING - DATABASE SECURITY PROJECT - DBST 668 FALL SEMESTER 2012--

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CONNECT DB668A02/UMUC2012;

-------------------------------------------------------

--DATA DEFINITION LANGUAGE – DDL--

-------------------------------------------------------

------------------------------

--DROP SEQUENCES

------------------------------

DROP SEQUENCE CUST\_ID\_SEQ;

DROP SEQUENCE ACCT\_ID\_SEQ;

DROP SEQUENCE ACCESS\_ID\_SEQ;

DROP SEQUENCE TRANS\_ID\_SEQ;

DROP SEQUENCE TRANS\_TYPE\_ID\_SEQ;

DROP SEQUENCE ACCT\_TYPE\_ID\_SEQ;

DROP SEQUENCE ACCT\_NUM\_SEQ;

------------------------

--DROP INDEXES

------------------------

--TBL\_CUSTOMER

DROP INDEX INX\_CUSTOMER\_ON\_FNAME;

DROP INDEX INX\_CUSTOMER\_ON\_LNAME;

DROP INDEX INX\_CUSTOMER\_ON\_CUST\_SSN;

--TBL\_TRANSACTION

DROP INDEX INX\_TRANSACTION\_DATE\_DESC;

DROP INDEX INX\_TRANSACTION\_ON\_ACCT\_NUM;

DROP INDEX INX\_TRANSACTION\_ON\_CHECK\_NUM;

----------------------

--DROP TABLES

----------------------

DROP TABLE TBL\_TRANSACTION;

DROP TABLE TBL\_ACCESS;

DROP TABLE TBL\_ACCOUNT;

DROP TABLE TBL\_ACCT\_TYPE;

DROP TABLE TBL\_CUSTOMER;

DROP TABLE TBL\_TRANS\_TYPE;

--------------------------

--CREATE TABLES

---------------------------

CREATE TABLE TBL\_CUSTOMER

(

CUST\_ID INTEGER PRIMARY KEY,

FNAME VARCHAR2(10) NOT NULL ,

LNAME VARCHAR2(10) NOT NULL ,

CUST\_STREET VARCHAR2(20) NOT NULL ,

CUST\_CITY VARCHAR2(15) NOT NULL ,

CUST\_STATE VARCHAR2(2) NOT NULL ,

CUST\_ZIP VARCHAR2(5) NOT NULL ,

CUST\_PHONE VARCHAR2(12) NOT NULL ,

CUST\_EMAIL VARCHAR2(16),

CUST\_SSN VARCHAR2(10) NOT NULL ,

CUST\_TYPE VARCHAR2(1) NOT NULL ,

CUST\_STATUS INTEGER NOT NULL);

--CUST\_TYPE FOR NORMALIZATION COULD BE IN A SEPARATE TABLE.WE ASSUMNE THE CLIENT DECLARES

--WETHER THEY ARE PERSONAL CLIENT VALUE = P, CORPORATE CLIENT VALUE =C AND VIP = V.

-- CUST\_STATUS VALUE OF 1 = ACTIVE AND 0 = INACTIVE

CREATE TABLE TBL\_ACCT\_TYPE

(

ACCT\_TYPE\_ID INTEGER PRIMARY KEY ,

ACCT\_NAME VARCHAR2(10) NOT NULL);

CREATE TABLE TBL\_TRANS\_TYPE

(

TR\_TYPE\_ID INTEGER NOT NULL ,

TR\_TYPE INTEGER PRIMARY KEY ,

TR\_NAME VARCHAR2(10) NOT NULL

)

;

COMMENT ON TABLE TBL\_ACCT\_TYPE IS '

1 - CHECKING

2 - SAVING

3 - CORPORATE

4 - INVESTMENT'

;

COMMENT ON TABLE TBL\_TRANS\_TYPE IS '-- TRANSACTION TYPES (TR\_TYPE) DEFINED AS FOLLOWS:

-- 1 DEPOSIT

-- 2 WITHDRAWAL

-- 3 TRANSFER

-- 4 INTEREST PAYMENT

-- 5 OPEN ACCOUNT

-- 6 CLOSE ACCOUNT'

;

CREATE TABLE TBL\_ACCOUNT

(

ACCT\_ID INTEGER NOT NULL,

ACCT\_NUM INTEGER PRIMARY KEY,

OPEN\_DATE DATE NOT NULL,

OPEN\_BAL NUMBER(10,2) NOT NULL,

ACCT\_BAL NUMBER(10,2) NOT NULL,

ACCT\_MINBAL NUMBER(10,2) NOT NULL,

CLOSE\_DATE DATE,

ACCT\_ACTIVE INTEGER NOT NULL, /\* NEW ATTRIBUTE - A 1 INDICATES THAT THE ACCOUNT IS OPEN AND A 0 SIGNIFIES THAT THE ACCOUNT HAS BEEN CLOSED\*/

ACCT\_TYPE\_ID INTEGER NOT NULL,

CONSTRAINT FK\_TBL\_ACCOUNT\_ACCT\_TYPE\_ID FOREIGN KEY (ACCT\_TYPE\_ID) REFERENCES TBL\_ACCT\_TYPE(ACCT\_TYPE\_ID));

CREATE TABLE TBL\_ACCESS

(

ACCESS\_ID INTEGER PRIMARY KEY,

CUST\_ID INTEGER NOT NULL,

ACCT\_NUM INTEGER NOT NULL,

ACC\_ACTIVE INTEGER NOT NULL, /\* THIS ATTRIBUTE SIGNIFIES THAT THE ACCESS PAIR IS ACTIVE - A 1 SIGNIFIES ACTIVE AND A 0 INDICATES INACTIVE \*/

CONSTRAINT FK\_TBL\_ACCESS\_ACCT\_NUM FOREIGN KEY (ACCT\_NUM) REFERENCES TBL\_ACCOUNT(ACCT\_NUM));

CREATE TABLE TBL\_TRANSACTION

(

TR\_ID INTEGER PRIMARY KEY,

ACCT\_NUM INTEGER NOT NULL,

TR\_DATE DATE NOT NULL,

TR\_AMT NUMBER(10,2) NOT NULL,

CHECK\_NUM INTEGER,

REF\_NUM INTEGER,

TR\_TYPE INTEGER NOT NULL,

CONSTRAINT FK\_TBL\_TRANSACTION\_ACCT\_NUM FOREIGN KEY (ACCT\_NUM) REFERENCES TBL\_ACCOUNT(ACCT\_NUM),

CONSTRAINT FK\_TBL\_TRANSACTION\_TR\_TYPE FOREIGN KEY (TR\_TYPE) REFERENCES TBL\_TRANS\_TYPE(TR\_TYPE));

--------------------------------

---CREATE VIEWS

-------------------------------

-----------------------------------

--CREATE AGENT VIEWS

-----------------------------------

--DISPLAYS CUSTOMER CONTACT INFORMATION

CREATE OR REPLACE VIEW AG\_CUST\_CONTACT\_VIEW

AS

SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

FROM DB668A02.TBL\_CUSTOMER

WHERE CUST\_TYPE = 'P'

AND CUST\_STATUS = 1;

--TEST VIEW

SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

--DISPLAYS CUSTOMER ACCOUNT INFORMATION

CREATE OR REPLACE VIEW AG\_CUST\_ACCT\_VIEW

AS

SELECT C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NAME

FROM DB668A02.TBL\_CUSTOMER C

INNER JOIN DB668A02.TBL\_ACCESS AC ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM= AC.ACCT\_NUM

INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID = A.ACCT\_TYPE\_ID

WHERE C.CUST\_TYPE = 'P'

AND C.CUST\_STATUS = 1;

--TEST VIEW

SELECT \* FROM DB668A02.AG\_CUST\_ACCT\_VIEW;

--DISPLAYS TRANSACTION ACTIVITY INFORMATION

CREATE OR REPLACE VIEW AG\_CUST\_TRANS\_VIEW

AS

SELECT C.CUST\_ID,C.FNAME,C.LNAME,T.TR\_DATE,T.TR\_AMT,TT.TR\_NAME

FROM DB668A02.TBL\_TRANSACTION T

INNER JOIN DB668A02.TBL\_ACCESS AC ON AC.ACCT\_NUM=T.ACCT\_NUM

INNER JOIN DB668A02.TBL\_CUSTOMER C ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE = TT.TR\_TYPE

WHERE AC.ACC\_ACTIVE = '1'

AND CUST\_STATUS = 1;

--TEST VIEW

SELECT \* FROM DB668A02.AG\_CUST\_TRANS\_VIEW;

---------------------------------------------------------

--CREATE ACCOUNT MANAGER VIEWS

---------------------------------------------------------

--DISPLAYS CUSTOMER CONTACT INFORMATION

CREATE OR REPLACE VIEW ACM\_CUST\_CONTACT\_VIEW

AS

SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

FROM DB668A02.TBL\_CUSTOMER

WHERE CUST\_TYPE IN ('P','C')

AND CUST\_STATUS = 1;

--TEST VIEW

SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

--DISPLAYS CUSTOMER ACCOUNT INFORMATION

CREATE OR REPLACE VIEW ACM\_CUST\_ACCT\_VIEW

AS

SELECT C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NAME

FROM DB668A02.TBL\_CUSTOMER C

INNER JOIN DB668A02.TBL\_ACCESS AC ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM= AC.ACCT\_NUM

INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID = A.ACCT\_TYPE\_ID

WHERE C.CUST\_TYPE IN ('P','C')

AND C.CUST\_STATUS = 1;

--TEST VIEW

SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

--DISPLAYS TRANSACTION ACTIVITY INFORMATION

CREATE OR REPLACE VIEW ACM\_CUST\_TRANS\_VIEW

AS

SELECT C.CUST\_ID,AC.ACCT\_NUM,C.FNAME,C.LNAME,T.TR\_DATE,T.TR\_AMT,TT.TR\_NAME

FROM DB668A02.TBL\_TRANSACTION T

INNER JOIN DB668A02.TBL\_ACCESS AC ON AC.ACCT\_NUM=T.ACCT\_NUM

INNER JOIN DB668A02.TBL\_CUSTOMER C ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE = TT.TR\_TYPE

WHERE AC.ACC\_ACTIVE = '1'

AND CUST\_STATUS = 1

AND C.CUST\_TYPE IN ('P','C');

--TEST VIEW

SELECT \* FROM DB668A02.ACM\_CUST\_TRANS\_VIEW;

------------------------------------------------------

--CREATE BANKER MANAGER VIEWS

-------------------------------------------------------

--DISPLAYS CUSTOMER CONTACT INFORMATION

CREATE OR REPLACE VIEW BM\_CUST\_CONTACT\_VIEW

AS

SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

FROM DB668A02.TBL\_CUSTOMER;

--TEST VIEW

SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

--DISPLAYS CUSTOMER ACCOUNT INFORMATION

CREATE OR REPLACE VIEW BM\_CUST\_ACCT\_VIEW

AS

SELECT C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NAME

FROM DB668A02.TBL\_CUSTOMER C

INNER JOIN DB668A02.TBL\_ACCESS AC ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM= AC.ACCT\_NUM

INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID = A.ACCT\_TYPE\_ID;

--TEST VIEW

SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

--DISPLAYS TRANSACTION ACTIVITY INFORMATION

CREATE OR REPLACE VIEW BM\_CUST\_TRANS\_VIEW

AS

SELECT C.CUST\_ID,C.FNAME,C.LNAME,AC.ACCT\_NUM,T.TR\_DATE,T.TR\_AMT,TT.TR\_NAME

FROM DB668A02.TBL\_TRANSACTION T

INNER JOIN DB668A02.TBL\_ACCESS AC ON AC.ACCT\_NUM=T.ACCT\_NUM

INNER JOIN DB668A02.TBL\_CUSTOMER C ON C.CUST\_ID=AC.CUST\_ID

INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE = TT.TR\_TYPE;

--TEST VIEW

SELECT \* FROM DB668A02.BM\_CUST\_TRANS\_VIEW;

--DISPLAYS ACCOUNT TYPE INFORMATION

CREATE OR REPLACE VIEW BM\_ACCT\_TYPE\_VIEW

AS

SELECT \*

FROM DB668A02.TBL\_ACCT\_TYPE;

--TEST VIEW

SELECT \* FROM DB668A02.BM\_ACCT\_TYPE\_VIEW;

--DISPLAYS TRANSACTION TYPE INFORMATION

CREATE OR REPLACE VIEW BM\_TRANS\_TYPE\_VIEW

AS

SELECT \*

FROM DB668A02.TBL\_TRANS\_TYPE;

--TEST VIEW

SELECT \* FROM DB668A02.BM\_TRANS\_TYPE\_VIEW;

--DISPLAYS INVESTMENT ACCOUNTS

CREATE OR REPLACE VIEW BM\_INVEST\_ACCT\_VIEW

AS

SELECT AT.ACCT\_NAME,A.OPEN\_BAL,A.ACCT\_BAL

FROM DB668A02.TBL\_ACCOUNT A

INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID = A.ACCT\_TYPE\_ID

WHERE AT.ACCT\_NAME='INVESTMENT';

--TEST VIEW

SELECT \* FROM DB668A02.BM\_INVEST\_ACCT\_VIEW;

----------------------------

--CREATE INDEXES

----------------------------

--TBL\_CUSTOMER

CREATE INDEX INX\_CUSTOMER\_ON\_FNAME ON TBL\_CUSTOMER(FNAME);

CREATE INDEX INX\_CUSTOMER\_ON\_LNAME ON TBL\_CUSTOMER(LNAME);

CREATE INDEX INX\_CUSTOMER\_ON\_CUST\_SSN ON TBL\_CUSTOMER(CUST\_SSN);

--TBL\_TRANSACTION

CREATE INDEX INX\_TRANSACTION\_DATE\_DESC ON TBL\_TRANSACTION(TR\_DATE DESC);

CREATE INDEX INX\_TRANSACTION\_ON\_ACCT\_NUM ON TBL\_TRANSACTION(ACCT\_NUM);

CREATE INDEX INX\_TRANSACTION\_ON\_CHECK\_NUM ON TBL\_TRANSACTION(CHECK\_NUM);

---------------------------------

--CREATE SEQUENCES

---------------------------------

CREATE SEQUENCE CUST\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

ORDER;

CREATE SEQUENCE ACCT\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

CREATE SEQUENCE ACCESS\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

CREATE SEQUENCE TRANS\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

CREATE SEQUENCE TRANS\_TYPE\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

CREATE SEQUENCE ACCT\_TYPE\_ID\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

CREATE SEQUENCE ACCT\_NUM\_SEQ MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE

ORDER;

------------------------------------------

--CREATE INSERT TRIGGERS

-------------------------------------------

--TBL\_CUSTOMER TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER CUST\_ID\_TRIG

BEFORE INSERT ON TBL\_CUSTOMER

FOR EACH ROW

BEGIN

IF :NEW.CUST\_ID IS NULL THEN

SELECT CUST\_ID\_SEQ.NEXTVAL

INTO :NEW.CUST\_ID

FROM DUAL;

END IF;

END;

/

--TBL\_ACCT\_TYPE TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER ACCT\_TYPE\_ID\_TRIG

BEFORE INSERT ON TBL\_ACCT\_TYPE

FOR EACH ROW

BEGIN

IF :NEW.ACCT\_TYPE\_ID IS NULL THEN

SELECT ACCT\_TYPE\_ID\_SEQ.NEXTVAL

INTO :NEW.ACCT\_TYPE\_ID

FROM DUAL;

END IF;

END;

/

--TBL\_TRANS\_TYPE TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER TRANS\_TYPE\_ID\_TRIG

BEFORE INSERT ON TBL\_TRANS\_TYPE

FOR EACH ROW

BEGIN

IF :NEW.TR\_TYPE\_ID IS NULL THEN

SELECT TRANS\_TYPE\_ID\_SEQ.NEXTVAL

INTO :NEW.TR\_TYPE\_ID

FROM DUAL;

END IF;

END;

/

--TBL\_ACCOUNT TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER ACCT\_ID\_TRIG

BEFORE INSERT ON TBL\_ACCOUNT

FOR EACH ROW

BEGIN

IF :NEW.ACCT\_ID IS NULL THEN

SELECT ACCT\_ID\_SEQ.NEXTVAL

INTO :NEW.ACCT\_ID

FROM DUAL;

END IF;

END;

/

--TBL\_ACCESS TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER ACCESS\_ID\_TRIG

BEFORE INSERT ON TBL\_ACCESS

FOR EACH ROW

BEGIN

IF :NEW.ACCESS\_ID IS NULL THEN

SELECT ACCESS\_ID\_SEQ.NEXTVAL

INTO :NEW.ACCESS\_ID

FROM DUAL;

END IF;

END;

/

--TBL\_TRANSACTION TRIGGER AND TEST INSERT

CREATE OR REPLACE TRIGGER TR\_ID\_TRIG

BEFORE INSERT ON TBL\_TRANSACTION

FOR EACH ROW

BEGIN

IF :NEW.TR\_ID IS NULL THEN

SELECT TRANS\_ID\_SEQ.NEXTVAL

INTO :NEW.TR\_ID

FROM DUAL;

END IF;

END;

/

CREATE OR REPLACE TRIGGER UPDATE\_ACCT\_BAL

AFTER INSERT ON TBL\_TRANSACTION

DECLARE

ACCTNUM NUMBER(38, 0);

TRAMT NUMBER(38,0);

TRTYPE NUMBER(38,0);

BEGIN

SELECT TR\_TYPE INTO TRTYPE

FROM TBL\_TRANSACTION

WHERE TR\_ID = (

SELECT MAX(TR\_ID)

FROM TBL\_TRANSACTION);

SELECT T.TR\_AMT INTO TRAMT

FROM TBL\_TRANSACTION T JOIN TBL\_ACCESS ACE

ON T.ACCT\_NUM = ACE.ACCT\_NUM

WHERE T.TR\_ID = (

SELECT MAX(TR\_ID)

FROM TBL\_TRANSACTION);

SELECT ACE.ACCT\_NUM INTO ACCTNUM

FROM TBL\_TRANSACTION T JOIN TBL\_ACCESS ACE

ON T.ACCT\_NUM = ACE.ACCT\_NUM

WHERE T.TR\_ID = (

SELECT MAX(TR\_ID)

FROM TBL\_TRANSACTION);

CASE TRTYPE

WHEN 1 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = ACCT\_BAL + TRAMT

WHERE ACCT\_NUM = ACCTNUM;

WHEN 2 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = ACCT\_BAL - TRAMT

WHERE ACCT\_NUM = ACCTNUM;

WHEN 3 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = ACCT\_BAL - TRAMT

WHERE ACCT\_NUM = ACCTNUM;

WHEN 4 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = ACCT\_BAL + TRAMT

WHERE ACCT\_NUM = ACCTNUM;

WHEN 5 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = ACCT\_BAL + TRAMT

WHERE ACCT\_NUM = ACCTNUM;

WHEN 6 THEN

UPDATE TBL\_ACCOUNT

SET ACCT\_BAL = 0

WHERE ACCT\_NUM = ACCTNUM;

END CASE;

END;

/

SHOW ERRORS

--------------------------------------

--TABLE CATALOG

-------------------------------------

DESCRIBE TBL\_CUSTOMER;

DESCRIBE TBL\_ACCT\_TYPE;

DESCRIBE TBL\_TRANS\_TYPE;

DESCRIBE TBL\_ACCOUNT;

DESCRIBE TBL\_ACCESS;

DESCRIBE TBL\_TRANSACTION;

SQL> --------------------------------------

SQL> --TABLE CATALOG

SQL> -------------------------------------

SQL>

SQL> DESCRIBE TBL\_CUSTOMER;

Name Null? Type

----------------------------------------- -------- ----------------------------

CUST\_ID NOT NULL NUMBER(38)

FNAME NOT NULL VARCHAR2(10)

LNAME NOT NULL VARCHAR2(10)

CUST\_STREET NOT NULL VARCHAR2(20)

CUST\_CITY NOT NULL VARCHAR2(15)

CUST\_STATE NOT NULL VARCHAR2(2)

CUST\_ZIP NOT NULL VARCHAR2(5)

CUST\_PHONE NOT NULL VARCHAR2(12)

CUST\_EMAIL VARCHAR2(16)

CUST\_SSN NOT NULL VARCHAR2(10)

CUST\_TYPE NOT NULL VARCHAR2(1)

CUST\_STATUS NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_ACCT\_TYPE;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCT\_TYPE\_ID NOT NULL NUMBER(38)

ACCT\_NAME NOT NULL VARCHAR2(10)

SQL> DESCRIBE TBL\_TRANS\_TYPE;

Name Null? Type

----------------------------------------- -------- ----------------------------

TR\_TYPE\_ID NOT NULL NUMBER(38)

TR\_TYPE NOT NULL NUMBER(38)

TR\_NAME NOT NULL VARCHAR2(10)

SQL> DESCRIBE TBL\_ACCOUNT;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCT\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

OPEN\_DATE NOT NULL DATE

OPEN\_BAL NOT NULL NUMBER(10,2)

ACCT\_BAL NOT NULL NUMBER(10,2)

ACCT\_MINBAL NOT NULL NUMBER(10,2)

CLOSE\_DATE DATE

ACCT\_ACTIVE NOT NULL NUMBER(38)

ACCT\_TYPE\_ID NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_ACCESS;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCESS\_ID NOT NULL NUMBER(38)

CUST\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

ACC\_ACTIVE NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_TRANSACTION;

Name Null? Type

----------------------------------------- -------- ----------------------------

TR\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

TR\_DATE NOT NULL DATE

TR\_AMT NOT NULL NUMBER(10,2)

CHECK\_NUM NUMBER(38)

REF\_NUM NUMBER(38)

TR\_TYPE NOT NULL NUMBER(38)

### DATABASE DLL OUTPUT LOG

SQL>

SQL>/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SQL>--BRIGHT STAR CONSULTING - DATABASE SECURITY PROJECT -

DBST 668 FALL SEMESTER 2012

SQL>/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SQL>

SQL> @/class/db668a/02/tss.sql

SQL> CONNECT DB668A02/umuc2012;

Connected.

SQL>

SQL> --------------------------------------------

SQL> --DATA DEFINITION LANGUAGE - DDL

SQL> --------------------------------------------

SQL>

SQL>

SQL> --------------------

SQL> --DROP SEQUENCES

SQL> --------------------

SQL>

SQL>

SQL> DROP SEQUENCE CUST\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE ACCT\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE ACCESS\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE TRANS\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE TRANS\_TYPE\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE ACCT\_TYPE\_ID\_SEQ;

Sequence dropped.

SQL> DROP SEQUENCE ACCT\_NUM\_SEQ;

Sequence dropped.

SQL>

SQL> --------------------

SQL> --DROP INDEXES

SQL> --------------------

SQL>

SQL> --TBL\_CUSTOMER

SQL> DROP INDEX INX\_CUSTOMER\_ON\_FNAME;

Index dropped.

SQL> DROP INDEX INX\_CUSTOMER\_ON\_LNAME;

Index dropped.

SQL> DROP INDEX INX\_CUSTOMER\_ON\_CUST\_SSN;

Index dropped.

SQL>

SQL> --TBL\_TRANSACTION

SQL> DROP INDEX INX\_TRANSACTION\_DATE\_DESC;

Index dropped.

SQL> DROP INDEX INX\_TRANSACTION\_ON\_ACCT\_NUM;

Index dropped.

SQL> DROP INDEX INX\_TRANSACTION\_ON\_CHECK\_NUM;

Index dropped.

SQL>

SQL> ----------------

SQL> --DROP TABLES

SQL> ----------------

SQL>

SQL> DROP TABLE TBL\_TRANSACTION;

Table dropped.

SQL> DROP TABLE TBL\_ACCESS;

Table dropped.

SQL> DROP TABLE TBL\_ACCOUNT;

Table dropped.

SQL> DROP TABLE TBL\_ACCT\_TYPE;

Table dropped.

SQL> DROP TABLE TBL\_CUSTOMER;

Table dropped.

SQL> DROP TABLE TBL\_TRANS\_TYPE;

Table dropped.

SQL>

SQL>

SQL>

SQL> ------------------

SQL> --CREATE OBJECTS

SQL> ------------------

SQL>

SQL>

SQL> -----------------

SQL> --CREATE TABLES

SQL> -----------------

SQL>

SQL>

SQL>

SQL> CREATE TABLE TBL\_CUSTOMER

2 (

3 CUST\_ID INTEGER PRIMARY KEY,

4 FNAME VARCHAR2(10) NOT NULL ,

5 LNAME VARCHAR2(10) NOT NULL ,

6 CUST\_STREET VARCHAR2(20) NOT NULL ,

7 CUST\_CITY VARCHAR2(15) NOT NULL ,

8 CUST\_STATE VARCHAR2(2) NOT NULL ,

9 CUST\_ZIP VARCHAR2(5) NOT NULL ,

10 CUST\_PHONE VARCHAR2(12) NOT NULL ,

11 CUST\_EMAIL VARCHAR2(16),

12 CUST\_SSN VARCHAR2(10) NOT NULL ,

13 CUST\_TYPE VARCHAR2(1) NOT NULL ,

14 CUST\_STATUS INTEGER NOT NULL);

Table created.

SQL>

SQL> --CUST\_TYPE for normalization could be in a separate

table.We assumne the client declares

SQL> --wether they are personal client value = p, corporate client

value =c and VIP = v.

SQL> -- CUST\_STATUS value of 1 = active and 0 = inactive

SQL>

SQL> CREATE TABLE TBL\_ACCT\_TYPE

2 (

3 ACCT\_TYPE\_ID INTEGER PRIMARY KEY ,

4 ACCT\_NAME VARCHAR2(10) NOT NULL);

Table created.

SQL>

SQL> CREATE TABLE TBL\_TRANS\_TYPE

2 (

3 TR\_TYPE\_ID INTEGER NOT NULL ,

4 TR\_TYPE INTEGER PRIMARY KEY ,

5 TR\_NAME VARCHAR2(10) NOT NULL

6 )

7 ;

Table created.

SQL>

SQL> SET ECHO OFF;

SQL>

SQL>

SQL> COMMENT ON TABLE TBL\_ACCT\_TYPE IS '

2 1 - Checking

3 2 - Saving

4 3 - Corporate

5 4 - Investment'

6 ;

Comment created.

SQL> COMMENT ON TABLE TBL\_TRANS\_TYPE IS '-- Transaction

types (TR\_TYPE) defined as follows:

2 -- 1 deposit

3 -- 2 withdrawal

4 -- 3 transfer

5 -- 4 interest payment

6 -- 5 Open Account

7 -- 6 Close Account'

8 ;

Comment created.

SQL>

SQL>

SQL>

SQL> CREATE TABLE TBL\_ACCOUNT

2 (

3 ACCT\_ID INTEGER NOT NULL,

4 ACCT\_NUM INTEGER PRIMARY KEY,

5 OPEN\_DATE DATE NOT NULL,

6 OPEN\_BAL NUMBER(10,2) NOT NULL,

7 ACCT\_BAL NUMBER(10,2) NOT NULL,

8 ACCT\_MINBAL NUMBER(10,2) NOT NULL,

9 CLOSE\_DATE DATE,

10 ACCT\_ACTIVE INTEGER NOT NULL, /\* new attribute - a 1

indicates that the account is open and a 0 signifies that the

account has been closed\*/

11 ACCT\_TYPE\_ID INTEGER NOT NULL,

12 CONSTRAINT FK\_TBL\_ACCOUNT\_ACCT\_TYPE\_ID FOREIGN KEY

(ACCT\_TYPE\_ID) REFERENCES TBL\_ACCT\_TYPE(ACCT\_TYPE\_ID));

Table created.

SQL>

SQL>

SQL>

SQL> CREATE TABLE TBL\_ACCESS

2 (

3 ACCESS\_ID INTEGER PRIMARY KEY,

4 CUST\_ID INTEGER NOT NULL,

5 ACCT\_NUM INTEGER NOT NULL,

6 ACC\_ACTIVE INTEGER NOT NULL, /\* this attribute signifies

that the access pair is active - a 1 signifies active and a 0

indicates inactive \*/

7 CONSTRAINT FK\_TBL\_ACCESS\_ACCT\_NUM FOREIGN KEY

(ACCT\_NUM) REFERENCES TBL\_ACCOUNT(ACCT\_NUM));

Table created.

SQL>

SQL> CREATE TABLE TBL\_TRANSACTION

2 (

3 TR\_ID INTEGER PRIMARY KEY,

4 ACCT\_NUM INTEGER NOT NULL,

5 TR\_DATE DATE NOT NULL,

6 TR\_AMT NUMBER(10,2) NOT NULL,

7 CHECK\_NUM INTEGER,

8 REF\_NUM INTEGER,

9 TR\_TYPE INTEGER NOT NULL,

10 CONSTRAINT FK\_TBL\_TRANSACTION\_ACCT\_NUM FOREIGN

KEY (ACCT\_NUM) REFERENCES TBL\_ACCOUNT(ACCT\_NUM),

11 CONSTRAINT FK\_TBL\_TRANSACTION\_TR\_TYPE FOREIGN KEY

(TR\_TYPE) REFERENCES TBL\_TRANS\_TYPE(TR\_TYPE));

Table created.

SQL>

SQL>

SQL> --------------------------------

SQL> ---CREATE VIEWS

SQL> -------------------------------

SQL>

SQL> ---------------------

SQL> --CREATE AGENT VIEWS

SQL> ---------------------

SQL>

SQL> --DISPLAYS CUSTOMER CONTACT INFORMATION

SQL> CREATE OR REPLACE VIEW AG\_CUST\_CONTACT\_VIEW

2 AS

3 SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

4 FROM DB668A02.TBL\_CUSTOMER

5 WHERE CUST\_TYPE = 'P'

6 AND CUST\_STATUS = 1;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS CUSTOMER ACCOUNT INFORMATION

SQL> CREATE OR REPLACE VIEW AG\_CUST\_ACCT\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NA

ME

4 FROM DB668A02.TBL\_CUSTOMER C

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

C.CUST\_ID=AC.CUST\_ID

6 INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM=

AC.ACCT\_NUM

7 INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID =

A.ACCT\_TYPE\_ID

8 WHERE C.CUST\_TYPE = 'P'

9 AND C.CUST\_STATUS = 1;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.AG\_CUST\_ACCT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS TRANSACTION ACTIVITY INFORMATION

SQL> CREATE OR REPLACE VIEW AG\_CUST\_TRANS\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,T.TR\_DATE,T.TR\_AMT,TT.TR\_NAME

4 FROM DB668A02.TBL\_TRANSACTION T

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

AC.ACCT\_NUM=T.ACCT\_NUM

6 INNER JOIN DB668A02.TBL\_CUSTOMER C ON

C.CUST\_ID=AC.CUST\_ID

7 INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE =

TT.TR\_TYPE

8 WHERE AC.ACC\_ACTIVE = '1'

9 AND CUST\_STATUS = 1;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.AG\_CUST\_TRANS\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL>

SQL> -------------------------------

SQL> --CREATE ACCOUNT MANAGER VIEWS

SQL> -------------------------------

SQL>

SQL>

SQL> --DISPLAYS CUSTOMER CONTACT INFORMATION

SQL> CREATE OR REPLACE VIEW ACM\_CUST\_CONTACT\_VIEW

2 AS

3 SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

4 FROM DB668A02.TBL\_CUSTOMER

5 WHERE CUST\_TYPE IN ('P','C')

6 AND CUST\_STATUS = 1;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS CUSTOMER ACCOUNT INFORMATION

SQL> CREATE OR REPLACE VIEW ACM\_CUST\_ACCT\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NA

ME

4 FROM DB668A02.TBL\_CUSTOMER C

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

C.CUST\_ID=AC.CUST\_ID

6 INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM=

AC.ACCT\_NUM

7 INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID =

A.ACCT\_TYPE\_ID

8 WHERE C.CUST\_TYPE IN ('P','C')

9 AND C.CUST\_STATUS = 1;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS TRANSACTION ACTIVITY INFORMATION

SQL> CREATE OR REPLACE VIEW ACM\_CUST\_TRANS\_VIEW

2 AS

3 SELECT

C.CUST\_ID,AC.ACCT\_NUM,C.FNAME,C.LNAME,T.TR\_DATE,T.TR\_AMT,TT.T

R\_NAME

4 FROM DB668A02.TBL\_TRANSACTION T

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

AC.ACCT\_NUM=T.ACCT\_NUM

6 INNER JOIN DB668A02.TBL\_CUSTOMER C ON

C.CUST\_ID=AC.CUST\_ID

7 INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE =

TT.TR\_TYPE

8 WHERE AC.ACC\_ACTIVE = '1'

9 AND CUST\_STATUS = 1

10 AND C.CUST\_TYPE IN ('P','C');

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_TRANS\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL> -------------------------------

SQL> --CREATE BANKER MANAGER VIEWS

SQL> -------------------------------

SQL>

SQL> --DISPLAYS CUSTOMER CONTACT INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_CUST\_CONTACT\_VIEW

2 AS

3 SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

4 FROM DB668A02.TBL\_CUSTOMER;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS CUSTOMER ACCOUNT INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_CUST\_ACCT\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NA

ME

4 FROM DB668A02.TBL\_CUSTOMER C

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

C.CUST\_ID=AC.CUST\_ID

6 INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM=

AC.ACCT\_NUM

7 INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID =

A.ACCT\_TYPE\_ID;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

no rows selected

SQL>

SQL> --DISPLAYS CUSTOMER CONTACT INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_CUST\_CONTACT\_VIEW

2 AS

3 SELECT CUST\_ID,FNAME,LNAME,CUST\_PHONE,CUST\_EMAIL

4 FROM DB668A02.TBL\_CUSTOMER;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS CUSTOMER ACCOUNT INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_CUST\_ACCT\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,A.ACCT\_NUM,A.ACCT\_BAL,AT.ACCT\_NA

ME

4 FROM DB668A02.TBL\_CUSTOMER C

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

C.CUST\_ID=AC.CUST\_ID

6 INNER JOIN DB668A02.TBL\_ACCOUNT A ON A.ACCT\_NUM=

AC.ACCT\_NUM

7 INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID =

A.ACCT\_TYPE\_ID;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL> --DISPLAYS TRANSACTION ACTIVITY INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_CUST\_TRANS\_VIEW

2 AS

3 SELECT

C.CUST\_ID,C.FNAME,C.LNAME,AC.ACCT\_NUM,T.TR\_DATE,T.TR\_AMT,TT.T

R\_NAME

4 FROM DB668A02.TBL\_TRANSACTION T

5 INNER JOIN DB668A02.TBL\_ACCESS AC ON

AC.ACCT\_NUM=T.ACCT\_NUM

6 INNER JOIN DB668A02.TBL\_CUSTOMER C ON

C.CUST\_ID=AC.CUST\_ID

7 INNER JOIN DB668A02.TBL\_TRANS\_TYPE TT ON T.TR\_TYPE =

TT.TR\_TYPE;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_CUST\_TRANS\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL> --DISPLAYS ACCOUNT TYPE INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_ACCT\_TYPE\_VIEW

2 AS

3 SELECT \*

4 FROM DB668A02.TBL\_ACCT\_TYPE;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_ACCT\_TYPE\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL> --DISPLAYS TRANSACTION TYPE INFORMATION

SQL> CREATE OR REPLACE VIEW BM\_TRANS\_TYPE\_VIEW

2 AS

3 SELECT \*

4 FROM DB668A02.TBL\_TRANS\_TYPE;

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_TRANS\_TYPE\_VIEW;

no rows selected

SQL>

SQL>

SQL> --DISPLAYS INVESTMENT ACCOUNTS

SQL> CREATE OR REPLACE VIEW BM\_INVEST\_ACCT\_VIEW

2 AS

3 SELECT AT.ACCT\_NAME,A.OPEN\_BAL,A.ACCT\_BAL

4 FROM DB668A02.TBL\_ACCOUNT A

5 INNER JOIN DB668A02.TBL\_ACCT\_TYPE AT ON AT.ACCT\_TYPE\_ID =

A.ACCT\_TYPE\_ID

6 WHERE AT.ACCT\_NAME='Investment';

View created.

SQL>

SQL> --TEST VIEW

SQL> SELECT \* FROM DB668A02.BM\_INVEST\_ACCT\_VIEW;

no rows selected

SQL>

SQL>

SQL>

SQL> ----------------

SQL> --CREATE INDEXES

SQL> ----------------

SQL> --TBL\_CUSTOMER

SQL> CREATE INDEX INX\_CUSTOMER\_ON\_FNAME ON

TBL\_CUSTOMER(FNAME);

Index created.

SQL>

SQL> CREATE INDEX INX\_CUSTOMER\_ON\_LNAME ON

TBL\_CUSTOMER(LNAME);

Index created.

SQL>

SQL> CREATE INDEX INX\_CUSTOMER\_ON\_CUST\_SSN ON

TBL\_CUSTOMER(CUST\_SSN);

Index created.

SQL>

SQL>

SQL> --TBL\_TRANSACTION

SQL>

SQL> CREATE INDEX INX\_TRANSACTION\_DATE\_DESC ON

TBL\_TRANSACTION(TR\_DATE DESC);

Index created.

SQL>

SQL> CREATE INDEX INX\_TRANSACTION\_ON\_ACCT\_NUM ON

TBL\_TRANSACTION(ACCT\_NUM);

Index created.

SQL>

SQL> CREATE INDEX INX\_TRANSACTION\_ON\_CHECK\_NUM ON

TBL\_TRANSACTION(CHECK\_NUM);

Index created.

SQL>

SQL>

SQL>

SQL> --------------------

SQL> --CREATE SEQUENCES

SQL> --------------------

SQL>

SQL>

SQL> CREATE SEQUENCE CUST\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 ORDER;

Sequence created.

SQL>

SQL> CREATE SEQUENCE ACCT\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL> CREATE SEQUENCE ACCESS\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL> CREATE SEQUENCE TRANS\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL> CREATE SEQUENCE TRANS\_TYPE\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL> CREATE SEQUENCE ACCT\_TYPE\_ID\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL>

SQL> CREATE SEQUENCE ACCT\_NUM\_SEQ MINVALUE 1

2 START WITH 1

3 INCREMENT BY 1

4 NOCACHE

5 NOCYCLE

6 ORDER;

Sequence created.

SQL>

SQL> SET ECHO OFF;

SQL>

SQL>

SQL>

SQL> -------------------------

SQL> --CREATE INSERT TRIGGERS

SQL> -------------------------

SQL>

SQL> --TBL\_CUSTOMER TRIGGER AND TEST INSERT

SQL>

SQL> CREATE OR REPLACE TRIGGER CUST\_ID\_TRIG

2 BEFORE INSERT ON TBL\_CUSTOMER

3 FOR EACH ROW

4 BEGIN

5 IF :new.CUST\_ID is NULL THEN

6 SELECT CUST\_ID\_SEQ.NEXTVAL

7 INTO :new.CUST\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL>

SQL> --TBL\_ACCT\_TYPE TRIGGER AND TEST INSERT

SQL> CREATE OR REPLACE TRIGGER ACCT\_TYPE\_ID\_TRIG

2 BEFORE INSERT ON TBL\_ACCT\_TYPE

3 FOR EACH ROW

4 BEGIN

5 IF :new.ACCT\_TYPE\_ID is NULL THEN

6 SELECT ACCT\_TYPE\_ID\_SEQ.NEXTVAL

7 INTO :new.ACCT\_TYPE\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL> --TBL\_TRANS\_TYPE TRIGGER AND TEST INSERT

SQL> CREATE OR REPLACE TRIGGER TRANS\_TYPE\_ID\_TRIG

2 BEFORE INSERT ON TBL\_TRANS\_TYPE

3 FOR EACH ROW

4 BEGIN

5 IF :new.TR\_TYPE\_ID is NULL THEN

6 SELECT TRANS\_TYPE\_ID\_SEQ.NEXTVAL

7 INTO :new.TR\_TYPE\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL> --TBL\_ACCOUNT TRIGGER AND TEST INSERT

SQL> CREATE OR REPLACE TRIGGER ACCT\_ID\_TRIG

2 BEFORE INSERT ON TBL\_ACCOUNT

3 FOR EACH ROW

4 BEGIN

5 IF :new.ACCT\_ID is NULL THEN

6 SELECT ACCT\_ID\_SEQ.NEXTVAL

7 INTO :new.ACCT\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL> --TBL\_ACCESS TRIGGER AND TEST INSERT

SQL> CREATE OR REPLACE TRIGGER ACCESS\_ID\_TRIG

2 BEFORE INSERT ON TBL\_ACCESS

3 FOR EACH ROW

4 BEGIN

5 IF :new.ACCESS\_ID is NULL THEN

6 SELECT ACCESS\_ID\_SEQ.NEXTVAL

7 INTO :new.ACCESS\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL> --TBL\_TRANSACTION TRIGGER AND TEST INSERT

SQL> CREATE OR REPLACE TRIGGER TR\_ID\_TRIG

2 BEFORE INSERT ON TBL\_TRANSACTION

3 FOR EACH ROW

4 BEGIN

5 IF :new.TR\_ID is NULL THEN

6 SELECT TRANS\_ID\_SEQ.NEXTVAL

7 INTO :new.TR\_ID

8 FROM DUAL;

9 END IF;

10 END;

11 /

Trigger created.

SQL>

SQL>

SQL>

SQL> CREATE OR REPLACE TRIGGER Update\_Acct\_Bal

2 AFTER INSERT ON TBL\_TRANSACTION

3

4 DECLARE

5 AcctNum Number(38, 0);

6 TrAmt Number(38,0);

7 TrType Number(38,0);

8

9 BEGIN

10

11 SELECT TR\_TYPE INTO TrType

12 FROM TBL\_TRANSACTION

13 WHERE TR\_ID = (

14 SELECT MAX(TR\_ID)

15 FROM TBL\_TRANSACTION);

16

17 SELECT t.tr\_amt INTO TrAmt

18 FROM TBL\_TRANSACTION t JOIN TBL\_ACCESS ace

19 ON t.ACCT\_NUM = ace.ACCT\_NUM

20 WHERE t.TR\_ID = (

21 SELECT MAX(TR\_ID)

22 FROM TBL\_TRANSACTION);

23

24 SELECT ace.ACCT\_NUM INTO AcctNum

25 FROM TBL\_TRANSACTION t JOIN TBL\_ACCESS ace

26 ON t.ACCT\_NUM = ace.ACCT\_NUM

27 WHERE t.TR\_ID = (

28 SELECT MAX(TR\_ID)

29 FROM TBL\_TRANSACTION);

30

31 CASE TrType

32 WHEN 1 THEN

33 UPDATE TBL\_ACCOUNT

34 SET ACCT\_BAL = ACCT\_BAL + TrAmt

35 WHERE ACCT\_NUM = AcctNum;

36 WHEN 2 THEN

37 UPDATE TBL\_ACCOUNT

38 SET ACCT\_BAL = ACCT\_BAL - TrAmt

39 WHERE ACCT\_NUM = AcctNum;

40 WHEN 3 THEN

41 UPDATE TBL\_ACCOUNT

42 SET ACCT\_BAL = ACCT\_BAL - TrAmt

43 WHERE ACCT\_NUM = AcctNum;

44 WHEN 4 THEN

45 UPDATE TBL\_ACCOUNT

46 SET ACCT\_BAL = ACCT\_BAL + TrAmt

47 WHERE ACCT\_NUM = AcctNum;

48 WHEN 5 THEN

49 UPDATE TBL\_ACCOUNT

50 SET ACCT\_BAL = ACCT\_BAL + TrAmt

51 WHERE ACCT\_NUM = AcctNum;

52 WHEN 6 THEN

53 UPDATE TBL\_ACCOUNT

54 SET ACCT\_BAL = 0

55 WHERE ACCT\_NUM = AcctNum;

56 END case;

57

58

59 END;

60 /

Trigger created.

SQL> SHOW ERRORS

No errors.

SQL>

SQL>

SQL> --------------------------------------

SQL> --TABLE CATALOG

SQL> -------------------------------------

SQL>

SQL> DESCRIBE TBL\_CUSTOMER;

Name Null? Type

----------------------------------------- -------- ----------------------------

CUST\_ID NOT NULL NUMBER(38)

FNAME NOT NULL VARCHAR2(10)

LNAME NOT NULL VARCHAR2(10)

CUST\_STREET NOT NULL VARCHAR2(20)

CUST\_CITY NOT NULL VARCHAR2(15)

CUST\_STATE NOT NULL VARCHAR2(2)

CUST\_ZIP NOT NULL VARCHAR2(5)

CUST\_PHONE NOT NULL VARCHAR2(12)

CUST\_EMAIL VARCHAR2(16)

CUST\_SSN NOT NULL VARCHAR2(10)

CUST\_TYPE NOT NULL VARCHAR2(1)

CUST\_STATUS NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_ACCT\_TYPE;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCT\_TYPE\_ID NOT NULL NUMBER(38)

ACCT\_NAME NOT NULL VARCHAR2(10)

SQL> DESCRIBE TBL\_TRANS\_TYPE;

Name Null? Type

----------------------------------------- -------- ----------------------------

TR\_TYPE\_ID NOT NULL NUMBER(38)

TR\_TYPE NOT NULL NUMBER(38)

TR\_NAME NOT NULL VARCHAR2(10)

SQL> DESCRIBE TBL\_ACCOUNT;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCT\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

OPEN\_DATE NOT NULL DATE

OPEN\_BAL NOT NULL NUMBER(10,2)

ACCT\_BAL NOT NULL NUMBER(10,2)

ACCT\_MINBAL NOT NULL NUMBER(10,2)

CLOSE\_DATE DATE

ACCT\_ACTIVE NOT NULL NUMBER(38)

ACCT\_TYPE\_ID NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_ACCESS;

Name Null? Type

----------------------------------------- -------- ----------------------------

ACCESS\_ID NOT NULL NUMBER(38)

CUST\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

ACC\_ACTIVE NOT NULL NUMBER(38)

SQL> DESCRIBE TBL\_TRANSACTION;

Name Null? Type

----------------------------------------- -------- ----------------------------

TR\_ID NOT NULL NUMBER(38)

ACCT\_NUM NOT NULL NUMBER(38)

TR\_DATE NOT NULL DATE

TR\_AMT NOT NULL NUMBER(10,2)

CHECK\_NUM NUMBER(38)

REF\_NUM NUMBER(38)

TR\_TYPE NOT NULL NUMBER(38)

SQL>

SQL>

OLS\_TSS02 SECURITY – SECURITY DDL - SCRIPT: SEC.SQL

SET ECHO ON;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--BRIGHT STAR CONSULTING - DATABASE SECURITY PROJECT - DBST 668 FALL SEMESTER 2012

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CONNECT DB668A02/umuc2012 as sysdba;

SET ECHO OFF;

----------------------------------------------------------------

--CREATE USERS WITH SESSION PRIVILEGES

----------------------------------------------------------------

--ADAMS02 EXAMPLE OF AGENT (AG)

DROP USER ADAMS02 CASCADE;

CREATE USER ADAMS02 IDENTIFIED BY nZh5Dmgt;

GRANT CREATE SESSION TO ADAMS02;

--BAKER02 EXAMPLE OF ACCOUNT MANAGER (ACM)

DROP USER BAKER02 CASCADE;

CREATE USER BAKER02 IDENTIFIED BY hcX0miWm;

GRANT CREATE SESSION TO BAKER02;

--CHUCK02 EXAMPLE OF BANK MANAGER (BM)

DROP USER CHUCK02 CASCADE;

CREATE USER CHUCK02 IDENTIFIED BY S1QdfhU7;

GRANT CREATE SESSION TO CHUCK02;

--------------------------------------------------------------------------------------------------------------

--CREATE USER ROLES, ASSIGN ROLE PRIVILEGES, ASSIGN ROLES TO USERS

--------------------------------------------------------------------------------------------------------------

--DROP ROLES FOR TESTING

DROP ROLE AG\_ROLE;

DROP ROLE ACM\_ROLE;

DROP ROLE BM\_ROLE;

--ROLES

CREATE ROLE AG\_ROLE;

CREATE ROLE ACM\_ROLE;

CREATE ROLE BM\_ROLE;

GRANT CREATE SESSION TO AG\_ROLE,ACM\_ROLE,BM\_ROLE;

--AGENT\_ROLE PRIVILEGES

GRANT SELECT,UPDATE ON DB668A02.TBL\_CUSTOMER TO AG\_ROLE;

GRANT SELECT,UPDATE ON DB668A02.TBL\_TRANSACTION TO AG\_ROLE;

GRANT SELECT ON DB668A02.TBL\_ACCOUNT TO AG\_ROLE;

GRANT SELECT ON DB668A02.TBL\_ACCESS TO AG\_ROLE;

--ACM\_ROLE PRIVILEGES

GRANT SELECT,UPDATE,INSERT ON DB668A02.TBL\_CUSTOMER TO ACM\_ROLE;

GRANT SELECT,UPDATE,INSERT ON DB668A02.TBL\_ACCOUNT TO ACM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_TRANSACTION TO ACM\_ROLE;

GRANT SELECT,UPDATE,INSERT ON DB668A02.TBL\_ACCESS TO ACM\_ROLE;

GRANT SELECT ON DB668A02.TBL\_ACCT\_TYPE TO ACM\_ROLE;

GRANT SELECT ON DB668A02.TBL\_TRANS\_TYPE TO ACM\_ROLE;

--BM\_ROLE PRIVILEGES

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_CUSTOMER TO BM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_ACCOUNT TO BM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_TRANSACTION TO BM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_ACCESS TO BM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_ACCT\_TYPE TO BM\_ROLE;

GRANT SELECT,UPDATE,INSERT,DELETE ON DB668A02.TBL\_TRANS\_TYPE TO BM\_ROLE;

--------------------------------------------------------------

--GRANT SEQUENCE PRIVILEGES TO ROLES

--------------------------------------------------------------

--AG\_ROLE

GRANT SELECT ON DB668A02.CUST\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.ACCESS\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_TYPE\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_TYPE\_ID\_SEQ TO AG\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_NUM\_SEQ TO AG\_ROLE;

--ACM\_ROLE

GRANT SELECT ON DB668A02.CUST\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACCESS\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_TYPE\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_TYPE\_ID\_SEQ TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_NUM\_SEQ TO ACM\_ROLE;

--BM\_ROLE

GRANT SELECT ON DB668A02.CUST\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.ACCESS\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.TRANS\_TYPE\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_TYPE\_ID\_SEQ TO BM\_ROLE;

GRANT SELECT ON DB668A02.ACCT\_NUM\_SEQ TO BM\_ROLE;

-----------------------------------------

--GRANT ROLES TO USERS

-----------------------------------------

GRANT AG\_ROLE TO ADAMS02;

GRANT ACM\_ROLE TO BAKER02;

GRANT BM\_ROLE TO CHUCK02;

--------------------------------------------------------------

--GRANT PRIVILEGES ON VIEWS TO ROLES

-------------------------------------------------------------

--AGENT\_ROLE VIEW PRIVILEGES

GRANT SELECT ON DB668A02.AG\_CUST\_CONTACT\_VIEW TO AG\_ROLE;

GRANT SELECT ON DB668A02.AG\_CUST\_ACCT\_VIEW TO AG\_ROLE;

GRANT SELECT ON DB668A02.AG\_CUST\_TRANS\_VIEW TO AG\_ROLE;

--ACM\_ROLE VIEW PRIVILEGES

GRANT SELECT ON DB668A02.ACM\_CUST\_CONTACT\_VIEW TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACM\_CUST\_ACCT\_VIEW TO ACM\_ROLE;

GRANT SELECT ON DB668A02.ACM\_CUST\_TRANS\_VIEW TO ACM\_ROLE;

--BM\_ROLE VIEW PRIVILEGES

GRANT SELECT ON DB668A02.BM\_CUST\_CONTACT\_VIEW TO BM\_ROLE;

GRANT SELECT ON DB668A02.BM\_CUST\_ACCT\_VIEW TO BM\_ROLE;

GRANT SELECT ON DB668A02.BM\_CUST\_TRANS\_VIEW TO BM\_ROLE;

GRANT SELECT ON DB668A02.BM\_ACCT\_TYPE\_VIEW TO BM\_ROLE;

GRANT SELECT ON DB668A02.BM\_TRANS\_TYPE\_VIEW TO BM\_ROLE;

GRANT SELECT ON DB668A02.BM\_INVEST\_ACCT\_VIEW TO BM\_ROLE;

----------------------------------------------------------

--DROP SECURITY BEFORE RE-CREATION

----------------------------------------------------------

BEGIN

SA\_SYSDBA.DROP\_POLICY (

POLICY\_NAME => 'TSSOLS02',

DROP\_COLUMN => TRUE);

END;

/

BEGIN

SA\_SYSDBA.CREATE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

COLUMN\_NAME => 'OLS\_TSS02');

END;

/

-------------------------------------------------------------------------------------------------

-- CREATE SECURITY OFFICER ACCOUNT AND GRANT PRIVILEGES

-------------------------------------------------------------------------------------------------

--GRANT SECURITY ACCESS PRIVILEGES TO DB668A02

--GRANT CREATE SESSION TO DB668A02 ;

GRANT EXECUTE ON SA\_SYSDBA TO DB668A02 ;

GRANT EXECUTE ON SA\_COMPONENTS TO DB668A02 ;

GRANT EXECUTE ON SA\_LABEL\_ADMIN TO DB668A02 ;

GRANT EXECUTE ON SA\_USER\_ADMIN TO DB668A02 ;

GRANT EXECUTE ON SA\_POLICY\_ADMIN TO DB668A02 ;

GRANT EXECUTE ON SA\_AUDIT\_ADMIN TO DB668A02 ;

GRANT EXECUTE ON CHAR\_TO\_LABEL TO DB668A02 ;

GRANT LBAC\_DBA TO DB668A02 ;

GRANT EXECUTE ON TO\_LBAC\_DATA\_LABEL TO DB668A02 WITH GRANT OPTION ;

GRANT TSSOLS02\_DBA TO DB668A02 ;

----------------------------------------------------------------------------------------------------------------------------------

--CREATE THE LEVELS BY EXECUTING THE SA\_COMPONENTS.CREATE LEVEL PROCEDURE

----------------------------------------------------------------------------------------------------------------------------------

BEGIN

SA\_COMPONENTS.CREATE\_LEVEL

(POLICY\_NAME => 'TSSOLS02',

LONG\_NAME => 'AGENT',

SHORT\_NAME => 'AG',

LEVEL\_NUM => 1002);

SA\_COMPONENTS.CREATE\_LEVEL

(POLICY\_NAME => 'TSSOLS02',

LONG\_NAME => 'ACCOUNT MANAGER',

SHORT\_NAME => 'ACM',

LEVEL\_NUM => 2002);

SA\_COMPONENTS.CREATE\_LEVEL (

POLICY\_NAME => 'TSSOLS02',

LONG\_NAME => 'BANK MANAGER',

SHORT\_NAME => 'BM',

LEVEL\_NUM => 3002);

END;

/

--------------------------------------------------------------------------------------------------------------------------------

--CREATE LABELS ON LEVELS WITH THE SA\_LABEL\_ADMIN.CREATE\_LABEL PROCEDURE

--------------------------------------------------------------------------------------------------------------------------------

BEGIN

SA\_LABEL\_ADMIN.CREATE\_LABEL (

POLICY\_NAME => 'TSSOLS02',

LABEL\_TAG => 1,

LABEL\_VALUE => 'AG');

SA\_LABEL\_ADMIN.CREATE\_LABEL (

POLICY\_NAME => 'TSSOLS02',

LABEL\_TAG => 2,

LABEL\_VALUE => 'ACM');

SA\_LABEL\_ADMIN.CREATE\_LABEL (

POLICY\_NAME => 'TSSOLS02',

LABEL\_TAG => 3,

LABEL\_VALUE => 'BM');

END;

/

--------------------------------------------------------

--APPLY POLICY WITH READ\_CONTROL

--------------------------------------------------------

BEGIN

--REMOVE POLICY TBL\_CUSTOMER

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_CUSTOMER');

--APPLY POLICY TBL\_CUSTOMER

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_CUSTOMER',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

--REMOVE POLICY TBL\_TRANSACTION

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_TRANSACTION');

--APPLY POLICY TBL\_TRANSACTION

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_TRANSACTION',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

--REMOVE POLICY TBL\_ACCOUNT

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCOUNT');

--APPLY POLICY TBL\_ACCOUNT

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCOUNT',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

--REMOVE POLICY TBL\_ACCT\_TYPE

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCT\_TYPE');

--APPLY POLICY TBL\_ACCT\_TYPE

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCT\_TYPE',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

--REMOVE POLICY TBL\_TRANS\_TYPE

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_TRANS\_TYPE');

--APPLY POLICY TBL\_TRANS\_TYPE

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_TRANS\_TYPE',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

--REMOVE POLICY TBL\_ACCESS

SA\_POLICY\_ADMIN.REMOVE\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCESS');

--APPLY POLICY TBL\_ACCESS

SA\_POLICY\_ADMIN.APPLY\_TABLE\_POLICY (

POLICY\_NAME => 'TSSOLS02',

SCHEMA\_NAME => 'DB668A02',

TABLE\_NAME => 'TBL\_ACCESS',

TABLE\_OPTIONS => 'READ\_CONTROL,WRITE\_CONTROL,LABEL\_UPDATE');

--TABLE\_OPTIONS => 'NO\_CONTROL');

END;

/

-----------------------------------------

--AUTHORIZE ROLE ACCESS

-----------------------------------------

BEGIN

SA\_USER\_ADMIN.SET\_USER\_LABELS (

POLICY\_NAME => 'TSSOLS02',

USER\_NAME => 'ADAMS02',

MAX\_READ\_LABEL => 'AG');

SA\_USER\_ADMIN.SET\_USER\_LABELS (

POLICY\_NAME => 'TSSOLS02',

USER\_NAME => 'BAKER02',

MAX\_READ\_LABEL => 'ACM');

SA\_USER\_ADMIN.SET\_USER\_LABELS (

POLICY\_NAME => 'TSSOLS02',

USER\_NAME => 'CHUCK02',

MAX\_READ\_LABEL => 'BM');

END;

/

OLS\_TSS02 - SECURED DATA DML - SCRIPT: TSSDATA.SQL

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--BRIGHT STAR CONSULTING - DATABASE SECURITY PROJECT - DBST 668 FALL SEMESTER 2012--

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

---------------------------------------------------------------------------------------------

-- INSERT ACCOUNT, TRANSACTION TYPES AND INITIAL DATA WITH SUPERUSER CHUCK02

---------------------------------------------------------------------------------------------

CONNECT CHUCK02/S1QdfhU7;

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Checking',1);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Saving',1);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Corporate',2);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Investment',3);

SELECT \* FROM DB668A02.TBL\_ACCT\_TYPE;

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,1,'Deposit',1);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,2,'Withdrawl',1);

SELECT \* FROM DB668A02.TBL\_TRANS\_TYPE;

--INSERT CUSTOMER DATA IN TBL\_CUSTOMER

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Joe','Biden', 'Anon Ave', 'Atown', 'PA', '60255','200-000-0000', 'jbiden@amail.com', 111111111,'P', 1,1);

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Barack','Obama', 'Penn Ave', 'Washtown', 'DC', '20001','100-000-0000', 'bobama@amail.com', 222222222,'C', 1,2);

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Paul','Ryan', 'Cap Hil Ave', 'Washtown', 'DC', '20001','300-000-0000', 'pryan@amail.com', 333333333,'C', 1,2);

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Mitt', 'Romney', 'Mass Ave', 'Masstown', 'MA', '30602','111-111-1111', 'mromney@bain.com', 444444444,'V', 1,3);

SELECT CUST\_ID,FNAME,LNAME,OLS\_TSS02 FROM DB668A02.TBL\_CUSTOMER;

--INSERT CUSTOMER DATA IN TBL\_CUSTOMER

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('10/06/2006','MM/DD/YYYY'),10000,10000,1000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Checking'),1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/06/2012','MM/DD/YYYY'),1000000,1000000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/06/2008','MM/DD/YYYY'),100000,100000,1000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Corporate'),2);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/06/2012','MM/DD/YYYY'),10000000,10000000,100000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Investment'),3);

--SELECT

SELECT ACCT\_ID,ACCT\_NUM,OPEN\_BAL,ACCT\_BAL,OLS\_TSS02 FROM DB668A02.TBL\_ACCOUNT;

--INSERT CUSTOMER ACCOUNT ACCESS DATA

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Joe' AND LNAME='Biden'),1111,1,1);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Barack' AND LNAME='Obama'),3333,1,2);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Paul' AND LNAME='Ryan'),2222,1,1);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Mitt' AND LNAME='Romney'),4444,1,3);

SELECT \* FROM DB668A02.TBL\_ACCESS;

--INSERT TRANSACTION DATA

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('10/14/2008','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/2010','MM/DD/YYYY'),111,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/01/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/06/2012','MM/DD/YYYY'),22000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/07/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/01/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/06/2012','MM/DD/YYYY'),22000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/07/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('01/14/2012','MM/DD/YYYY'),50000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),3);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/03/2012','MM/DD/YYYY'),1000000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),3);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

SELECT TR\_ID,ACCT\_NUM,TR\_DATE,TR\_AMT,OLS\_TSS02 FROM DB668A02.TBL\_TRANSACTION;

### DATABASE DATA ENTRY OUPUT LOG

SQL>

SQL> @/class/db668a/02/tssdata.sql

SQL> SET ECHO ON

SQL> /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SQL> --BRIGHT STAR CONSULTING - DATABASE SECURITY PROJECT - DBST 668 FALL SEMESTER 2012

SQL> /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SQL>

SQL> ------------------------------------------------------------------------------------------------------------------------------------------------

SQL> --OLS SECURITY DML: INSERT ACCOUNT,TRANSACTION TYPES AND INITIAL DATA WITH SUPERUSER CHUCK02

SQL> ------------------------------------------------------------------------------------------------------------------------------------------------

SQL> CONNECT CHUCK02/c;

Connected.

SQL>

SQL>

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Checking',1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Saving',1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Corporate',2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Investment',3);

1 row created.

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_ACCT\_TYPE;

ACCT\_TYPE\_ID ACCT\_NAME OLS\_TSS02

------------ ---------- ----------

2 Checking 1

3 Saving 1

4 Corporate 2

5 Investment 3

SQL>

SQL>

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,1,'Deposit',1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,2,'Withdrawl',1);

1 row created.

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_TRANS\_TYPE;

TR\_TYPE\_ID TR\_TYPE TR\_NAME OLS\_TSS02

---------- ---------- ---------- ----------

2 1 Deposit 1

3 2 Withdrawl 1

SQL>

SQL>

SQL> --INSERT CUSTOMER DATA IN TBL\_CUSTOMER

SQL>

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Joe','Biden', 'Anon Ave', 'Atown', 'PA', '60255','200-000-0000', 'jbiden@amail.com', 111111111,'P', 1,1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Barack','Obama', 'Penn Ave', 'Washtown', 'DC', '20001','100-000-0000', 'bobama@amail.com', 222222222,'C', 1,2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Paul','Ryan', 'Cap Hil Ave', 'Washtown', 'DC', '20001','300-000-0000', 'pryan@amail.com', 333333333,'C', 1,2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Mitt', 'Romney', 'Mass Ave', 'Masstown', 'MA', '30602','111-111-1111', 'mromney@bain.com', 444444444,'V', 1,3);

1 row created.

SQL>

SQL> SELECT CUST\_ID,FNAME,LNAME,OLS\_TSS02 FROM DB668A02.TBL\_CUSTOMER;

CUST\_ID FNAME LNAME OLS\_TSS02

---------- ---------- ---------- ----------

2 Joe Biden 1

3 Barack Obama 2

4 Paul Ryan 2

5 Mitt Romney 3

SQL>

SQL>

SQL> --INSERT CUSTOMER DATA IN TBL\_CUSTOMER

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('10/06/2006','MM/DD/YYYY'),10000,10000,1000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Checking'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/06/2012','MM/DD/YYYY'),1000000,1000000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/06/2008','MM/DD/YYYY'),100000,100000,1000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Corporate'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/06/2012','MM/DD/YYYY'),10000000,10000000,100000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Investment'),3);

1 row created.

SQL> --SELECT

SQL> SELECT ACCT\_ID,ACCT\_NUM,OPEN\_BAL,ACCT\_BAL,OLS\_TSS02 FROM DB668A02.TBL\_ACCOUNT;

ACCT\_ID ACCT\_NUM OPEN\_BAL ACCT\_BAL OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 1111 10000 10000 1

3 2222 1000000 1000000 1

4 3333 100000 100000 2

5 4444 10000000 10000000 3

SQL>

SQL>

SQL> --INSERT CUSTOMER ACCOUNT ACCESS DATA

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Joe' AND LNAME='Biden'),1111,1,1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Barack' AND LNAME='Obama'),3333,1,2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Paul' AND LNAME='Ryan'),2222,1,1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Mitt' AND LNAME='Romney'),4444,1,3);

1 row created.

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 1 1

5 5 4444 1 3

SQL>

SQL>

SQL>

SQL> --INSERT TRANSACTION DATA

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('10/14/2008','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/2010','MM/DD/YYYY'),111,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

1 row created.

SQL>

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/01/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/06/2012','MM/DD/YYYY'),22000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('10/07/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/01/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/06/2012','MM/DD/YYYY'),22000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('10/07/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('01/14/2012','MM/DD/YYYY'),50000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),3);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/03/2012','MM/DD/YYYY'),1000000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Withdrawl'),3);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

1 row created.

SQL>

SQL>

SQL> SELECT TR\_ID,ACCT\_NUM,TR\_DATE,TR\_AMT,OLS\_TSS02 FROM DB668A02.TBL\_TRANSACTION;

TR\_ID ACCT\_NUM TR\_DATE TR\_AMT OLS\_TSS02

---------- ---------- --------- ---------- ----------

2 1111 14-OCT-08 500 1

3 1111 14-DEC-10 111 1

4 1111 14-DEC-12 10000 1

5 2222 01-OCT-12 10000 2

6 2222 06-OCT-12 22000 2

7 2222 07-OCT-12 10000 2

8 3333 01-OCT-12 10000 2

9 3333 06-OCT-12 22000 2

10 3333 07-OCT-12 10000 2

11 4444 14-JAN-12 50000 3

12 4444 03-OCT-12 1000000 3

13 4444 07-OCT-12 1500000 3

12 rows selected.

SQL>

### DATABASE DATA REVIEW LOG

**(Note that some columns where omitted for better table presentation)**

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_TRANS\_TYPE;

TR\_TYPE\_ID TR\_TYPE TR\_NAME OLS\_TSS02

---------- ---------- ---------- ----------

2 1 Deposit 1

3 2 Withdrawl 1

SQL> SELECT CUST\_ID,FNAME,LNAME,OLS\_TSS02 FROM DB668A02.TBL\_CUSTOMER;

CUST\_ID FNAME LNAME OLS\_TSS02

---------- ---------- ---------- ----------

2 Joe Biden 1

3 Barack Obama 2

4 Paul Ryan 2

5 Mitt Romney 3

SQL> SELECT \* FROM DB668A02.TBL\_TRANS\_TYPE;

TR\_TYPE\_ID TR\_TYPE TR\_NAME OLS\_TSS02

---------- ---------- ---------- ----------

2 1 Deposit 1

3 2 Withdrawl 1

SQL> SELECT ACCT\_ID, ACCT\_NUM, OPEN\_BAL, ACCT\_BAL, OLS\_TSS02 FROM DB668A02.TBL\_ACCOUNT;

ACCT\_ID ACCT\_NUM OPEN\_BAL ACCT\_BAL OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 1111 10000 10000 1

3 2222 1000000 1000000 1

4 3333 100000 100000 2

5 4444 10000000 10000000 3

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 1 1

5 5 4444 1 3

SQL> SELECT TR\_ID,ACCT\_NUM,TR\_DATE,TR\_AMT,OLS\_TSS02 FROM DB668A02.TBL\_TRANSACTION;

TR\_ID ACCT\_NUM TR\_DATE TR\_AMT OLS\_TSS02

---------- ---------- --------- ---------- ----------

2 1111 14-OCT-08 500 1

3 1111 14-DEC-10 111 1

4 1111 14-DEC-12 10000 1

5 2222 01-OCT-12 10000 2

6 2222 06-OCT-12 22000 2

7 2222 07-OCT-12 10000 2

8 3333 01-OCT-12 10000 2

9 3333 06-OCT-12 22000 2

10 3333 07-OCT-12 10000 2

11 4444 14-JAN-12 50000 3

12 4444 03-OCT-12 1000000 3

13 4444 07-OCT-12 1500000 3

12 rows selected.

OLS\_TSS02 - AGENTS SECURITY TEST - SCRIPT TSEC\_AG.SQL

SET ECHO ON;

**-----------------------------------------**

**--AGENT SECURITY POLICY**

**-----------------------------------------**

Secury Policy: Agents are assigned security level 1

They are strictly not permitted to access or write any data at any other level.

1. They are permitted to view customer contact information at level 1

2. They aer permitted to view customer account information at level 1

3. They are permitted to view customer transaction activity at level 1

4. They are permitted to update customer data at level 1 or below

5. They are permitted to add transactions for existing customers at level 1

6. They are not permitted to update customer data at a higher policy level

7. Agents are not permitted to add any new records other than transactions at level 1

8. They are not permitted to delete any data

-----------------------------------------------------------------------------------------------

--TEST WITH SOME DATA IN THE TABLE

----------------------------------------------------------------------------------------------

CONNECT ADAMS02/nZh5Dmgt;

--1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION

SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

--2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION

SELECT \* FROM DB668A02.AG\_CUST\_ACCT\_VIEW;

--3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY

SELECT \* FROM DB668A02.AG\_CUST\_TRANS\_VIEW;

--4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT OLS\_TSS02 LEVEL 1

SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

UPDATE DB668A02.TBL\_CUSTOMER

SET CUST\_PHONE='202-222-3333'

WHERE CUST\_ID=2;

SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

--5. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT OLS\_TSS02 LEVEL 1

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/1999','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

--6. THEY ARE NOT PERMITTED TO UPDATE CUSTOMER DATA AT A HIGHER POLICY LEVEL EVEN IF THEY GUESS THE LAST NAME

UPDATE DB668A02.TBL\_CUSTOMER

SET CUST\_PHONE='202-222-3333'

WHERE LNAME='Romney';

--7. AGENTS ARE NOT PERMITTED TO ADD ANY NEW RECORDS OTHER THAN TRANSACTIONS FOR LEVEL 1 CUSTOMERS

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL, 'Hillary','Clinton', 'State Ave', 'DCtown', 'DC', '20003','202-222-3333', 'hillary@amail.com', 555555555,'P', 1,1);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Personal Checking',1);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,1,'Deposit',1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,5555,TO\_DATE('11/14/2012','MM/DD/YYYY'),1000,1000,100,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Personal Checking'),1);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Hillary' AND LNAME='Clinton'),5555,1,1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('12/14/1999','MM/DD/YYYY'),100000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

--11.THEY ARE NOT PERMITTED TO DELETE ANY DATA

--DATA TO TEST DELETION ATTEMPT

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),1);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

DELETE FROM DB668A02.TBL\_TRANSACTION

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCESS

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCOUNT

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_CUSTOMER

WHERE LNAME='Franck';

DELETE FROM DB668A02.TBL\_ACCT\_TYPE

WHERE ACCT\_NAME ='MMarket';

DELETE FROM DB668A02.TBL\_TRANS\_TYPE

WHERE TR\_NAME ='401K';

SET ECHO OFF;

### AGENT SECURITY TEST OUTPUT LOG

SQL> @/class/db668a/02/tsec\_ag.sql

SQL> --------------------------------------

SQL> --AGENT SECURITY POLICY

SQL> --------------------------------------

SQL> /\*

SQL> OLS POLICY: AGENTS ARE ASSIGNED SECURITY LEVEL 1

SQL> THEY ARE STRICTLY NOT PERMITTED TO ACCESS OR WRITE ANY DATA AT ANY OTHER LEVEL.

SQL>

SQL> 1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 1

SQL> 2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 1

SQL> 3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 1

SQL> 4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 1 OR BELOW

SQL> 5. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 1

SQL> 6. THEY ARE NOT PERMITTED TO UPDATE CUSTOMER DATA AT A HIGHER POLICY LEVEL

SQL> 7. AGENTS ARE NOT PERMITTED TO ADD ANY NEW RECORDS OTHER THAN TRANSACTIONS AT LEVEL 1

SQL> 8. THEY ARE NOT PERMITTED TO DELETE ANY DATA

SQL> \*/

SQL>

SQL> -----------------------------------------------------------------------------------------------

SQL> --TEST WITH SOME DATA IN THE TABLE

SQL> ----------------------------------------------------------------------------------------------

SQL>

SQL> CONNECT ADAMS02/nZh5Dmgt;

Connected.

SQL>

SQL>

SQL> --1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION

SQL> SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

SQL>

SQL> --2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION

SQL> SELECT \* FROM DB668A02.AG\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

SQL>

SQL> --3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY

SQL> SELECT \* FROM DB668A02.AG\_CUST\_TRANS\_VIEW;

CUST\_ID FNAME LNAME TR\_DATE TR\_AMT TR\_NAME

---------- ---------- ---------- --------- ---------- ----------

2 Joe Biden 14-OCT-08 500 Deposit

2 Joe Biden 14-DEC-10 111 Withdrawl

2 Joe Biden 14-DEC-12 10000 Deposit

SQL>

SQL>

SQL> --4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT OLS\_TSS02 LEVEL 1

SQL> SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

SQL>

SQL> UPDATE DB668A02.TBL\_CUSTOMER

2 SET CUST\_PHONE='202-222-3333'

3 WHERE CUST\_ID=2;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.AG\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 202-222-3333 jbiden@amail.com

SQL>

SQL> --5. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT OLS\_TSS02 LEVEL 1

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('12/14/1999','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

1 row created.

SQL>

SQL> --6. THEY ARE NOT PERMITTED TO UPDATE CUSTOMER DATA AT A HIGHER POLICY LEVEL EVEN IF THEY GUESS THE LAST NAME

SQL> UPDATE DB668A02.TBL\_CUSTOMER

2 SET CUST\_PHONE='202-222-3333'

3 WHERE LNAME='Romney';

0 rows updated.

SQL>

SQL> --7. AGENTS ARE NOT PERMITTED TO ADD ANY NEW RECORDS OTHER THAN TRANSACTIONS FOR LEVEL 1 CUSTOMERS

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL, 'Hillary','Clinton', 'State Ave', 'DCtown', 'DC', '20003','202-222-3333', 'hillary@amail.com', 555555555,'P', 1,1);

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL, 'Hillary','Clinton', 'State Ave', 'DCtown', 'DC', '20003','202-222-3333', 'hillary@amail.com', 555555555,'P', 1,1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Personal Checking',1);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Personal Checking',1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,1,'Deposit',1);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,1,'Deposit',1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,5555,TO\_DATE('11/14/2012','MM/DD/YYYY'),1000,1000,100,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Personal Checking'),1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,5555,TO\_DATE('11/14/2012','MM/DD/YYYY'),1000,1000,100,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Personal Checking'),1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Hillary' AND LNAME='Clinton'),5555,1,1);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Hillary' AND LNAME='Clinton'),5555,1,1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('12/14/1999','MM/DD/YYYY'),100000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,3333,TO\_DATE('12/14/1999','MM/DD/YYYY'),100000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3)

\*

ERROR at line 1:

ORA-28117: integrity constraint violated - parent record not found

SQL>

SQL>

SQL>

SQL> --11.THEY ARE NOT PERMITTED TO DELETE ANY DATA

SQL> --DATA TO TEST DELETION ATTEMPT

SQL>

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,1);

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),1);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,2);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,2)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1)

\*

ERROR at line 1:

ORA-28117: integrity constraint violated - parent record not found

SQL>

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANSACTION

2 WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_TRANSACTION

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCESS

2 WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCESS

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCOUNT

2 WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCOUNT

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_CUSTOMER

2 WHERE LNAME='Franck';

DELETE FROM DB668A02.TBL\_CUSTOMER

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCT\_TYPE

2 WHERE ACCT\_NAME ='MMarket';

DELETE FROM DB668A02.TBL\_ACCT\_TYPE

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANS\_TYPE

2 WHERE TR\_NAME ='401K';

DELETE FROM DB668A02.TBL\_TRANS\_TYPE

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> SET ECHO OFF;

SQL>

OLS\_TSS02 – ACM SECURITY TEST - SCRIPT TSEC\_ACM.SQL

SET ECHO ON;

**--------------------------------------------------------------**

**--ACCOUNT MANAGER SECURITY POLICY**

**---------------------------------------------------------------**

Security policy: Account Managers are assigned security level 2.

They are strictly not permitted to access or write any data at a higher level.

They are permitted full access to data at security level 2 or below.

1. They are permitted to view customer contact information at level 2 or below

2. They aer permitted to view customer account information at level 2 or below

3. They are permitted to view customer transaction activity at level 2 or below

4. They are permitted to update customer data at level 2 or below

5. They are permitted to update existing account data at level 2 or below

6. They are permitted to update access data at level 2 or below

7. They are permitted to add transactions for existing customers at level 2 or below

8. They are permitted to add new customers and assign the appropriate security level

9. They are permitted to add new accounts and assign the appropriate security level

10. They are not permitted to manage account types and transaction types in any fashion

11. They are not permitted to delete any data except for transactions in case of corrections

-----------------------------------------------------------------------------------------------

--TEST WITH SOME DATA IN THE TABLE

----------------------------------------------------------------------------------------------

CONNECT BAKER02/hcX0miWm;

--1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 2 OR BELOW

SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

--2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 2 OR BELOW

SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

--3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 2 OR BELOW

SELECT \* FROM DB668A02.ACM\_CUST\_TRANS\_VIEW;

--4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 2 OR BELOW

--UPDATE CUSTOMER PAUL RYAN'S PHONE NUMBER

SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

UPDATE DB668A02.TBL\_CUSTOMER

SET CUST\_PHONE='202-555-5555'

WHERE CUST\_ID=4;

SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

--5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 2 OR BELOW

--CHANGE THE ACCOUNT TYPE FOR RYAN FROM SAVING TO CHECKING

SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

UPDATE DB668A02.TBL\_ACCOUNT

SET ACCT\_TYPE\_ID =2

WHERE ACCT\_NUM=2222;

SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

--6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 2 OR BELOW

--SET ACCOUNT 2222 TO INACTIVE WITH VALUE 0

SELECT \* FROM DB668A02.TBL\_ACCESS;

UPDATE DB668A02.TBL\_ACCESS

SET ACC\_ACTIVE =0

WHERE ACCT\_NUM=2222;

SELECT \* FROM DB668A02.TBL\_ACCESS;

UPDATE DB668A02.TBL\_ACCESS

SET ACC\_ACTIVE =1

WHERE ACCT\_NUM=2222;

--7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 2 OR BELOW

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('11/14/2012','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('11/24/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

--8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Nancy', 'Pelosi', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

--9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS, ACCESS, AND ASSIGN THE APPROPRIATE SECURITY LEVEL

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Nancy' AND LNAME='Pelosi'),8888,1,2);

--10. THEY ARE NOT PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Money Market',2);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,3,'Transfer',1);

--11.THEY ARE NOT PERMITTED TO DELETE ANY DATA EXCEPT TRANSACTION AT THE APPROPRIATE SECURITY LEVEL

--ADD PELOSI TRANSACTION

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('11/24/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

--DELETE PELOSI TRANSACTION

DELETE FROM DB668A02.TBL\_TRANSACTION WHERE ACCT\_NUM=8888;

--DATA TO TEST DELETION ATTEMPT

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,2);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

DELETE FROM DB668A02.TBL\_TRANSACTION

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCESS

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCOUNT

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_CUSTOMER

WHERE LNAME='Franck';

DELETE FROM DB668A02.TBL\_ACCT\_TYPE

WHERE ACCT\_NAME ='MMarket';

DELETE FROM DB668A02.TBL\_TRANS\_TYPE

WHERE TR\_NAME ='401K';

SET ECHO OFF;

### ACCOUNT MANAGER SECURITY TEST OUTPUT LOG

SQL>

SQL> @/class/db668a/02/tsec\_acm.sql

SQL> ------------------------------------------------------------

SQL> --ACCOUNT MANAGER SECURITY POLICY

SQL> ------------------------------------------------------------

SQL> /\*

SQL> OLS POLICY: ACCOUNT MANAGERS ARE ASSIGNED SECURITY LEVEL 2

SQL> THEY ARE STRICTLY NOT PERMITTED TO ACCESS OR WRITE ANY DATA AT A HIGHER LEVEL.

SQL> THEY ARE PERMITTED FULL ACCESS TO DATA AT SUCURITY LEVEL 2 OR BELOW.

SQL>

SQL> 1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 2 OR BELOW

SQL> 2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 2 OR BELOW

SQL> 3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 2 OR BELOW

SQL> 4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 2 OR BELOW

SQL> 5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 2 OR BELOW

SQL> 6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 2 OR BELOW

SQL> 7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 2 OR BELOW

SQL> 8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> 9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> 10. THEY ARE NOT PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

SQL> 11.THEY ARE NOT PERMITTED TO DELETE ANY DATA EXCEPT FOR TRANSACTIONS IN CASE OF CORRECTIONS

SQL> \*/

SQL>

SQL>

SQL> -----------------------------------------------------------------------------------------------

SQL> --TEST WITH SOME DATA IN THE TABLE

SQL> ----------------------------------------------------------------------------------------------

SQL> CONNECT BAKER02/hcX0miWm;

Connected.

SQL>

SQL> --1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 2 OR BELOW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 300-000-0000 pryan@amail.com

SQL>

SQL> --2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 2 OR BELOW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

3 Barack Obama 3333 78000 Corporate

4 Paul Ryan 2222 978000 Saving

SQL>

SQL> --3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 2 OR BELOW

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_TRANS\_VIEW;

CUST\_ID ACCT\_NUM FNAME LNAME TR\_DATE TR\_AMT TR\_NAME

---------- ---------- ---------- ---------- --------- ---------- ----------

4 2222 Paul Ryan 07-OCT-12 10000 Deposit

3 3333 Barack Obama 07-OCT-12 10000 Deposit

2 1111 Joe Biden 14-DEC-12 10000 Deposit

2 1111 Joe Biden 14-OCT-08 500 Deposit

4 2222 Paul Ryan 06-OCT-12 22000 Withdrawl

4 2222 Paul Ryan 01-OCT-12 10000 Withdrawl

3 3333 Barack Obama 06-OCT-12 22000 Withdrawl

3 3333 Barack Obama 01-OCT-12 10000 Withdrawl

2 1111 Joe Biden 14-DEC-10 111 Withdrawl

9 rows selected.

SQL>

SQL> --4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 2 OR BELOW

SQL> --UPDATE CUSTOMER PAUL RYAN'S PHONE NUMBER

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 300-000-0000 pryan@amail.com

SQL>

SQL> UPDATE DB668A02.TBL\_CUSTOMER

2 SET CUST\_PHONE='202-555-5555'

3 WHERE CUST\_ID=4;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 202-555-5555 pryan@amail.com

SQL>

SQL> --5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 2 OR BELOW

SQL> --CHANGE THE ACCOUNT TYPE FOR RYAN FROM SAVING TO CHECKING

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

3 Barack Obama 3333 78000 Corporate

4 Paul Ryan 2222 978000 Saving

SQL>

SQL> UPDATE DB668A02.TBL\_ACCOUNT

2 SET ACCT\_TYPE\_ID =2

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.ACM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

3 Barack Obama 3333 78000 Corporate

4 Paul Ryan 2222 978000 Checking

SQL>

SQL> --6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 2 OR BELOW

SQL> --SET ACCOUNT 2222 TO INACTIVE WITH VALUE 0

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 1 1

SQL>

SQL> UPDATE DB668A02.TBL\_ACCESS

2 SET ACC\_ACTIVE =0

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 0 1

SQL>

SQL> UPDATE DB668A02.TBL\_ACCESS

2 SET ACC\_ACTIVE =1

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> --7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 2 OR BELOW

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('11/14/2012','MM/DD/YYYY'),500,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('11/24/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL>

SQL> --8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Nancy', 'Pelosi', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

1 row created.

SQL>

SQL> --9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS, ACCESS, AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Nancy' AND LNAME='Pelosi'),8888,1,2);

1 row created.

SQL>

SQL> --10. THEY ARE NOT PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Money Market',2);

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'Money Market',2)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,3,'Transfer',1);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,3,'Transfer',1)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> --11.THEY ARE NOT PERMITTED TO DELETE ANY DATA EXCEPT TRANSACTION AT THE APPROPRIATE SECURITY LEVEL

SQL> --ADD PELOSI TRANSACTION

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('11/24/2012','MM/DD/YYYY'),10000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL> --DELETE PELOSI TRANSACTION

SQL> DELETE FROM DB668A02.TBL\_TRANSACTION WHERE ACCT\_NUM=8888;

1 row deleted.

SQL>

SQL> --DATA TO TEST DELETION ATTEMPT

SQL>

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANSACTION

2 WHERE ACCT\_NUM=9999;

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCESS

2 WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCESS

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCOUNT

2 WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCOUNT

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_CUSTOMER

2 WHERE LNAME='Franck';

DELETE FROM DB668A02.TBL\_CUSTOMER

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCT\_TYPE

2 WHERE ACCT\_NAME ='MMarket';

DELETE FROM DB668A02.TBL\_ACCT\_TYPE

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANS\_TYPE

2 WHERE TR\_NAME ='401K';

DELETE FROM DB668A02.TBL\_TRANS\_TYPE

\*

ERROR at line 1:

ORA-01031: insufficient privileges

SQL>

SQL> SET ECHO OFF;

SQL>

OLS\_TSS02 – BM SECURITY TEST - SCRIPT TSEC\_BM.SQL

**--------------------------------------------------------**

**--BANK MANAGER SECURITY POLICY**

**--------------------------------------------------------**

Security Policy: Bank Managers are assigned security level 3

They are permitted all actions select, insert, update, and delete on all tables.

They assume all the rights of all lower level security policies.

Unlike acm, bank managers are able to to fully manager account types and transactions types.

1. They are permitted to view customer contact information at level 3 or below

2. They aer permitted to view customer account information at level 3 or below

3. They are permitted to view customer transaction activity at level 3 or below

4. They are permitted to update customer data at level 3 or below

5. They are permitted to update existing account data at level 3 or below

6. They are permitted to update access data at level 3 or below

7. They are permitted to add transactions for existing customers at level 3 or below

8. They are permitted to add new customers and assign the appropriate security level

9. They are permitted to add new accounts and assign the appropriate security level

10.they are permitted to manage account types and transaction types in any fashion

11.they are permitted to delete any data

-----------------------------------------------------------------------------------------------

--TEST WITH SOME DATA IN THE TABLE

----------------------------------------------------------------------------------------------

CONNECT CHUCK02/S1QdfhU7;

--1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 3 OR BELOW

--DISPLAYS CUSTOMER CONTACT INFORMATION

SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

--2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 3 OR BELOW

--DISPLAYS CUSTOMER ACCOUNT INFORMATION

SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

--DISPLAYS INVESTMENT ACCOUNTS

SELECT \* FROM DB668A02.BM\_INVEST\_ACCT\_VIEW;

--3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 3 OR BELOW

--DISPLAYS TRANSACTION TYPE INFORMATION

SELECT \* FROM DB668A02.BM\_TRANS\_TYPE\_VIEW;

--4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 3 OR BELOW

--UPDATE CUSTOMER PAUL RYAN'S PHONE NUMBER

SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

UPDATE DB668A02.TBL\_CUSTOMER

SET CUST\_PHONE='999-999-9999'

WHERE CUST\_ID=4;

SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

--5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 2 OR BELOW

--CHANGE THE ACCOUNT TYPE FOR RYAN FROM SAVING TO INVESTMENT

SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

UPDATE DB668A02.TBL\_ACCOUNT

SET ACCT\_TYPE\_ID =3

WHERE ACCT\_NUM=2222;

SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

--6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 2 OR BELOW

--SET ACCOUNT 2222 TO INACTIVE WITH VALUE 0

SELECT \* FROM DB668A02.TBL\_ACCESS;

UPDATE DB668A02.TBL\_ACCESS

SET ACC\_ACTIVE =0

WHERE ACCT\_NUM=2222;

SELECT \* FROM DB668A02.TBL\_ACCESS;

UPDATE DB668A02.TBL\_ACCESS

SET ACC\_ACTIVE =1

WHERE ACCT\_NUM=2222;

--7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 2 OR BELOW

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('05/14/2012','MM/DD/YYYY'),5555,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('05/24/2012','MM/DD/YYYY'),55555,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

--8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Nancy', 'Pelosi', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

--9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS, ACCESS, AND ASSIGN THE APPROPRIATE SECURITY LEVEL

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Nancy' AND LNAME='Pelosi'),8888,1,2);

--10. THEY ARE PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'MMarket',2);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,3,'Transfer',1);

INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,4,'401K',3);

--11.THEY ARE PERMITTED TO DELETE ANY TRANSACTION DATA

--ADD MITT ROMNEY TRANSACTION

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

--DELETE MITT ROMNEY'S LATEST TRANSACTION

DELETE FROM DB668A02.TBL\_TRANSACTION

WHERE ACCT\_NUM=4444

AND TR\_ID= (SELECT MAX(TR\_ID) FROM DB668A02.TBL\_TRANSACTION) ;

--DATA TO TEST DELETION ATTEMPT

INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,3);

INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Investment'),3);

INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,3);

INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

DELETE FROM DB668A02.TBL\_TRANSACTION

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCESS

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_ACCOUNT

WHERE ACCT\_NUM=9999;

DELETE FROM DB668A02.TBL\_CUSTOMER

WHERE LNAME='Franck';

DELETE FROM DB668A02.TBL\_ACCT\_TYPE

WHERE ACCT\_NAME ='MMarket';

DELETE FROM DB668A02.TBL\_TRANS\_TYPE

WHERE TR\_NAME ='401K';

SET ECHO OFF;

### BANK MANAGER SECURITY TEST OUTPUT LOG

SQL> @/class/db668a/02/tsec\_bm.sql

SQL> --------------------------------------

SQL> --BANK MANAGER SECURITY POLICY

SQL> --------------------------------------

SQL> /\*

SQL> OLS POLICY: BANK MANAGERS ARE ASSIGNED SECURITY LEVEL 3

SQL> THEY ARE PERMITTED ALL ACTIONS SELECT, INSERT, UPDATE, AND DELETE ON ALL TABLES.

SQL> THEY ASSUME ALL THE RIGHTS OF ALL LOWER LEVEL SECURITY POLICIES.

SQL> UNLIKE ACM, BANK MANAGERS ARE ABLE TO TO FULLY MANAGER ACCOUNT TYPES AND TRANSACTIONS TYPES.

SQL>

SQL> 1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 3 OR BELOW

SQL> 2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 3 OR BELOW

SQL> 3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 3 OR BELOW

SQL> 4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 3 OR BELOW

SQL> 5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 3 OR BELOW

SQL> 6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 3 OR BELOW

SQL> 7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 3 OR BELOW

SQL> 8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> 9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> 10.THEY ARE PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

SQL> 11.THEY ARE PERMITTED TO DELETE ANY DATA

SQL> \*/

SQL>

SQL> -----------------------------------------------------------------------------------------------

SQL> --TEST WITH SOME DATA IN THE TABLE

SQL> ----------------------------------------------------------------------------------------------

SQL>

SQL> CONNECT CHUCK02/S1QdfhU7;

Connected.

SQL>

SQL>

SQL> --1. THEY ARE PERMITTED TO VIEW CUSTOMER CONTACT INFORMATION AT LEVEL 3 OR BELOW

SQL> --DISPLAYS CUSTOMER CONTACT INFORMATION

SQL> SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 300-000-0000 pryan@amail.com

5 Mitt Romney 111-111-1111 mromney@bain.com

SQL>

SQL> --2. THEY AER PERMITTED TO VIEW CUSTOMER ACCOUNT INFORMATION AT LEVEL 3 OR BELOW

SQL> --DISPLAYS CUSTOMER ACCOUNT INFORMATION

SQL> SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

4 Paul Ryan 2222 978000 Saving

3 Barack Obama 3333 78000 Corporate

5 Mitt Romney 4444 10450000 Investment

SQL> --DISPLAYS INVESTMENT ACCOUNTS

SQL> SELECT \* FROM DB668A02.BM\_INVEST\_ACCT\_VIEW;

ACCT\_NAME OPEN\_BAL ACCT\_BAL

---------- ---------- ----------

Investment 10000000 10450000

SQL>

SQL> --3. THEY ARE PERMITTED TO VIEW CUSTOMER TRANSACTION ACTIVITY AT LEVEL 3 OR BELOW

SQL> --DISPLAYS TRANSACTION TYPE INFORMATION

SQL> SELECT \* FROM DB668A02.BM\_TRANS\_TYPE\_VIEW;

TR\_TYPE\_ID TR\_TYPE TR\_NAME

---------- ---------- ----------

2 1 Deposit

3 2 Withdrawl

SQL>

SQL> --4. THEY ARE PERMITTED TO UPDATE CUSTOMER DATA AT LEVEL 3 OR BELOW

SQL> --UPDATE CUSTOMER PAUL RYAN'S PHONE NUMBER

SQL> SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 300-000-0000 pryan@amail.com

5 Mitt Romney 111-111-1111 mromney@bain.com

SQL>

SQL> UPDATE DB668A02.TBL\_CUSTOMER

2 SET CUST\_PHONE='999-999-9999'

3 WHERE CUST\_ID=4;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.BM\_CUST\_CONTACT\_VIEW;

CUST\_ID FNAME LNAME CUST\_PHONE CUST\_EMAIL

---------- ---------- ---------- ------------ ----------------

2 Joe Biden 200-000-0000 jbiden@amail.com

3 Barack Obama 100-000-0000 bobama@amail.com

4 Paul Ryan 999-999-9999 pryan@amail.com

5 Mitt Romney 111-111-1111 mromney@bain.com

SQL>

SQL> --5. THEY ARE PERMITTED TO UPDATE EXISTING ACCOUNT DATA AT LEVEL 2 OR BELOW

SQL> --CHANGE THE ACCOUNT TYPE FOR RYAN FROM SAVING TO INVESTMENT

SQL> SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

4 Paul Ryan 2222 978000 Saving

3 Barack Obama 3333 78000 Corporate

5 Mitt Romney 4444 10450000 Investment

SQL>

SQL> UPDATE DB668A02.TBL\_ACCOUNT

2 SET ACCT\_TYPE\_ID =3

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.BM\_CUST\_ACCT\_VIEW;

CUST\_ID FNAME LNAME ACCT\_NUM ACCT\_BAL ACCT\_NAME

---------- ---------- ---------- ---------- ---------- ----------

2 Joe Biden 1111 20389 Checking

4 Paul Ryan 2222 978000 Saving

3 Barack Obama 3333 78000 Corporate

5 Mitt Romney 4444 10450000 Investment

SQL>

SQL> --6. THEY ARE PERMITTED TO UPDATE ACCESS DATA AT LEVEL 2 OR BELOW

SQL> --SET ACCOUNT 2222 TO INACTIVE WITH VALUE 0

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 1 1

5 5 4444 1 3

SQL>

SQL> UPDATE DB668A02.TBL\_ACCESS

2 SET ACC\_ACTIVE =0

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> SELECT \* FROM DB668A02.TBL\_ACCESS;

ACCESS\_ID CUST\_ID ACCT\_NUM ACC\_ACTIVE OLS\_TSS02

---------- ---------- ---------- ---------- ----------

2 2 1111 1 1

3 3 3333 1 2

4 4 2222 0 1

5 5 4444 1 3

SQL>

SQL> UPDATE DB668A02.TBL\_ACCESS

2 SET ACC\_ACTIVE =1

3 WHERE ACCT\_NUM=2222;

1 row updated.

SQL>

SQL> --7. THEY ARE PERMITTED TO ADD TRANSACTIONS FOR EXISTING CUSTOMERS AT LEVEL 2 OR BELOW

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,1111,TO\_DATE('05/14/2012','MM/DD/YYYY'),5555,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,2222,TO\_DATE('05/24/2012','MM/DD/YYYY'),55555,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),2);

1 row created.

SQL>

SQL>

SQL> --8. THEY ARE PERMITTED TO ADD NEW CUSTOMERS AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Nancy', 'Pelosi', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,2);

1 row created.

SQL>

SQL> --9. THEY ARE PERMITTED TO ADD NEW ACCOUNTS, ACCESS, AND ASSIGN THE APPROPRIATE SECURITY LEVEL

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,8888,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Saving'),2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Nancy' AND LNAME='Pelosi'),8888,1,2);

1 row created.

SQL>

SQL> --10. THEY ARE PERMITTED TO MANAGE ACCOUNT TYPES AND TRANSACTION TYPES IN ANY FASHION

SQL> INSERT INTO DB668A02.TBL\_ACCT\_TYPE VALUES (DB668A02.ACCT\_TYPE\_ID\_SEQ.NEXTVAL,'MMarket',2);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,3,'Transfer',1);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANS\_TYPE VALUES (DB668A02.TRANS\_TYPE\_ID\_SEQ.NEXTVAL,4,'401K',3);

1 row created.

SQL>

SQL> --11.THEY ARE PERMITTED TO DELETE ANY TRANSACTION DATA

SQL> --ADD MITT ROMNEY TRANSACTION

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,4444,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

1 row created.

SQL>

SQL> --DELETE MITT ROMNEY'S LATEST TRANSACTION

SQL> DELETE FROM DB668A02.TBL\_TRANSACTION

2 WHERE ACCT\_NUM=4444

3 AND TR\_ID= (SELECT MAX(TR\_ID) FROM DB668A02.TBL\_TRANSACTION) ;

1 row deleted.

SQL>

SQL> --DATA TO TEST DELETION ATTEMPT

SQL>

SQL> INSERT INTO DB668A02.TBL\_CUSTOMER VALUES (DB668A02.CUST\_ID\_SEQ.NEXTVAL,'Franck', 'Franck', 'House Ave', 'Htown', 'DC', '20002','777-777-7777', 'nancy@amail.com', 888888888,'V', 1,3);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCOUNT VALUES (DB668A02.ACCT\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('12/14/2000','MM/DD/YYYY'),8888888,100000,10000,'',1,(select acct\_type\_id from DB668A02.TBL\_acct\_type where acct\_name = 'Investment'),3);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_ACCESS VALUES (DB668A02.ACCESS\_ID\_SEQ.NEXTVAL,(select cust\_id from DB668A02.TBL\_customer where FName = 'Franck' AND LNAME='Franck'),9999,1,3);

1 row created.

SQL> INSERT INTO DB668A02.TBL\_TRANSACTION VALUES (DB668A02.TRANS\_ID\_SEQ.NEXTVAL,9999,TO\_DATE('10/07/2012','MM/DD/YYYY'),1500000,'','',(select tr\_type from DB668A02.TBL\_trans\_type where tr\_name = 'Deposit'),3);

1 row created.

SQL>

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANSACTION

2 WHERE ACCT\_NUM=9999;

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCESS

2 WHERE ACCT\_NUM=9999;

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCOUNT

2 WHERE ACCT\_NUM=9999;

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_CUSTOMER

2 WHERE LNAME='Franck';

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_ACCT\_TYPE

2 WHERE ACCT\_NAME ='MMarket';

1 row deleted.

SQL>

SQL> DELETE FROM DB668A02.TBL\_TRANS\_TYPE

2 WHERE TR\_NAME ='401K';

1 row deleted.

SQL>

SQL> SET ECHO OFF;

SQL>

**Works Cited**

Knox, D. C. (2004). *Effective Oracle Database 10g Security by Design.* Emeryville, CA: McGraw-Hill.

Oracle. (n.d.). *Oracle11g Database Online Documentation*. Retrieved Nov 18 2012, from docs.oracle.com: Retrieved from http://docs.oracle.com/cd/E24693\_01/nav/security.htm

Rischert, A. (2010). *Oracle SQL by Example, Fourth Edition.* Upper Saddle River, NJ: Prentice Hall.