Banking Database Security

DBST 668 Database Security

University of Maryland University College

Dr. Micheal Martin, Ph.D.



Table of Contents

[Statement of Work 3](#_Toc341029934)

[Database Name 3](#_Toc341029935)

[Database Objective 3](#_Toc341029936)

[The Problem 3](#_Toc341029937)

[The Security Plan 4](#_Toc341029938)

[Project Timeline 5](#_Toc341029939)

[Technical Details 7](#_Toc341029940)

[Logical Model Diagram 8](#_Toc341029941)

[Entity Relationship Diagram 9](#_Toc341029942)

[Data Definition Language 10](#_Toc341029943)

[Physical Design 10](#_Toc341029944)

[Data Definition Language (DDL) 14](#_Toc341029945)

[Data Manipulation Language 19](#_Toc341029946)

[Creating User Accounts 23](#_Toc341029947)

[Creating Roles 25](#_Toc341029948)

[Security Policies and Granting System Privileges 26](#_Toc341029949)

[Using Views to Manage Restrictions 27](#_Toc341029950)

[Oracle Label Security 54](#_Toc341029951)

[Conclusion 60](#_Toc341029952)

[References 62](#_Toc341029953)

# Statement of Work

### Database Name

Bank Database

### Database Objective

The goal of this project is to apply effective security techniques to a banking database system. The database was designed to store banking details including customer information, banking transactions, branch locations, bank statements and account types. In addition, the database keeps a track of all transaction history for each customer. The system contains entities which reflect each key detailed mentioned. The purpose of the system is to improve customer experience by providing online banking that is convenient, reliable, available, robust and efficient, but most importantly, one that provides a high level of security that will ensure customer privacy and preserve the integrity of the business. Other expected benefits include ease of storing, retrieving and tracking data from various sources within the banking chain. The database will contain customers’ details such as: customer name, address, and phone number. For each customer transaction, the system will capture the transaction name, transaction type, transaction amount, transaction begin and end date as well as other transaction details. Each branch location is also recorded. Details such as branch name, branch address and phone number can quickly identify which branch a customer is associated with. The various account type details such as account type name, and account type code are also recorded in their own tables.

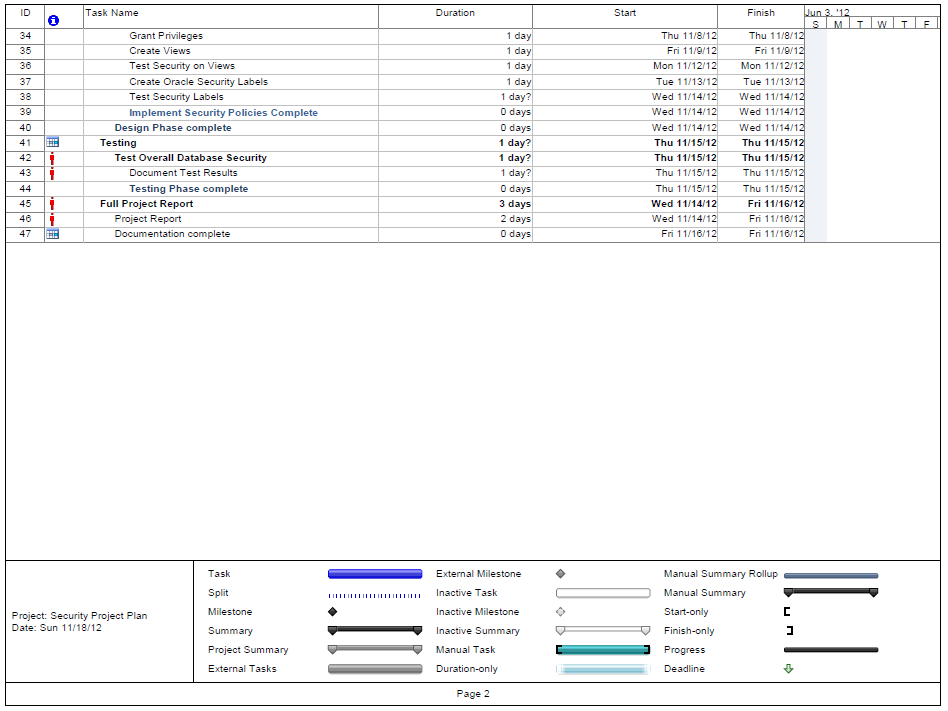
### The Problem

General Banking is a private institution that has branches across the country. The company has several hundred employees and thousands of customers nationwide. The steady advancements in technology and widespread use of the internet increase the demands for online banking. General Banking understands the urgency and importance to implement a system that will be able to accommodate employees and customers full banking needs. The center of this implementation is security. Though not a full online system, the current database houses customer records including social security numbers and other personal information gathered from all the different branches. In recent years, there have been several security breaches which compromised thousands of SSN. Other issues include data redundancy, lack of control in terms of data view, and the inability to mange and close security holes. In an effort to combat these challenges, the company organized a team consisting of senior management, technical staff and some end users. They also sent surveys to each customer to get a better understanding of what was most important in terms of their online banking needs.

# The Security Plan

Database security is of utmost importance to the design of this system. The team performed in-depth analysis of current and potential risks associated with implementation. All components of the system that must be secured were also identified. The database will achieve high level of security by creating roles and assigning privileges to each set of users. Views and Oracle Label Security will be used to hide complexity of tables from users but most importantly they will be designed to mask data from underlying tables so that users only access data that is relevant to them. Instead of storing customers’ social security numbers on the database, sequences will be used to generate a unique identifier for each record. To guarantee strict adherence to the security policies, each security measure is tested and documented.

# Project Timeline



# Technical Details

The database will be developed and managed using Oracle software stored on the dbcourse4a server accessed through Oracle SQL Developer and PortaPutty through UMUC. The Entity Relationship Diagram (ERD) will be created using Oracle SQL Developer Data Modeler. The scripts will include the data definition language (DDL), data manipulation language (DML), Oracle Label Security and views. In addition, security based commands such as GRANTS and REVOKE statements will be used to control the security at different levels of the system. All scripts will be created using SQL and executed using Oracle SQL Developer or SQLPlus through PortaPutty. Any Stored Procedures created will also be created using PL/SQL and run using Oracle Developer.

## E:\School\DBST668\Final Project\Bank_ERD.pngLogical Model Diagram

## E:\School\DBST668\Final Project\Bank_ERD.pngEntity Relationship Diagram

## Data Definition Language

### Physical Design

**Customer**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CUSTOMER\_ID | FNAME | LNAME | MIDDLE\_INITIAL | STREET\_ADDRESS | CITY | STATE | ZIP5 | PHONE | EMAIL\_ADDRESS |

|  |  |  |
| --- | --- | --- |
| CUSTOMER\_ID | NUMBER(10) | Not Null PK |
| FNAME | VARCHAR2(30) |  |
| LNAME | VARCHAR2(30) |  |
| MIDDLE\_INITIAL | VARCHAR2(1) |  |
| STREET\_ADDRESS | VARCHAR(35) |  |
| CITY | VARCHAR(20) |  |
| STATE | VARCHAR(2) |  |
| ZIP5 | VARCHAR(5) |  |
| PHONE | VARCHAR(12) |  |
| EMAIL\_ADDRESS | VARCHAR(50) |  |

**EMPLOYEE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPLOYEE\_ID | TITLE | HIRE\_DATE | SALARY | BRANCH\_ID | CUSTOMER\_ID |

|  |  |  |
| --- | --- | --- |
| EMPLOYEE\_ID | NUMBER(10) | Not Null PK |
| TITLE | VARCHAR2(40) |  |
| HIRE\_DATE | DATE |  |
| SALARY | NUMBER(5,2) |  |
| BRANCH\_ID | NUMBER(10) | FK |
| CUSTOMER\_ID | NUMBER(10) | FK |

**Transaction**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRANS\_ID | TRANS\_NAME | TRANS\_TYPE | TRANS\_CODE | TRANS\_AMOUNT | TRANS\_BEGIN\_DATE | TRANS\_END\_DATE | CUSTOMER\_ID | STATEMENT\_ID | ACCOUNT\_ID | BRANCH\_ID |

|  |  |  |
| --- | --- | --- |
| TRANSACTION\_ID | NUMBER(10) | Not Null PK |
| TRANSACTION\_NAME | VARCHAR2(30) |  |
| TRANSACTION\_TYPE | VARCHAR2(30) |  |
| TRANSACTION\_CODE | VARCHAR2(10) |  |
| TRANSACTION\_AMOUNT | VARCHAR2(35) |  |
| TRANSACTION\_BEGIN\_DATE | DATE |  |
| TRANSACTION\_END\_DATE | DATE |  |
| CUSTOMER\_ID | NUMBER(10) | FK |
| STATEMENT\_ID | NUMBER(10) | FK |
| ACCOUNT\_ID | NUMBER(10) | FK |
| BRANCH\_ID | NUMBER(10) | FK |

**Account\_Type**

|  |  |  |  |
| --- | --- | --- | --- |
| ACCOUNT\_ID | ACCOUNT\_TYPE | ACCOUNT\_CODE | CUSTOMER\_ID |

|  |  |  |
| --- | --- | --- |
| ACCOUNT\_ID | NUMBER(10) | Not Null PK |
| ACCOUNT\_TYPE\_NAME | VARCHAR2(30) |  |
| ACCOUNT\_TYPE\_CODE | VARCHAR2(30) |  |
| CUSTOMER\_ID | NUMBER(10) | FK |

**Bank\_Statement**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATEMENT\_ID | STATEMENT\_NUMBER | STATEMENT\_NAME | STATEMENT\_DATE | CUSTOMER\_ID |

|  |  |  |
| --- | --- | --- |
| STATEMENT\_ID | NUMBER(10) | Not Null PK |
| STATEMENT\_NUMBER | VARCHAR2(10) |  |
| STATEMENT\_NAME | VARCHAR2(60) |  |
| STATEMENT\_DATE | DATE |  |
| CUSTOMER\_ID | NUMBER(10) | FK |

**Board\_Director**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BMEMBER\_ID | BDFNAME | BDLNAME | BD\_PHONE | DIRECTOR\_LEVEL | LOCATION | SECLEV |

|  |  |  |
| --- | --- | --- |
| BMEMBER\_ID | NUMBER(10) | Not Null PK |
| BDFNAME | VARCHAR2(30) |  |
| BDLNAME | VARCHAR2(30) |  |
| BD\_PHONE | VARCHAR2(12) |  |
| LOCATION | VARCHAR2(50) |  |
| SECLEV | NUMBER(10) | OLS |

**Branch**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BRANCH\_ID | BRANCH\_NAME | BRANCH\_STREET | BRANCH\_CITY | BRANCH\_STATE | BRANCH\_ZIP5 | PHONE\_TOLLFREE | PHONE\_LOCAL |

|  |  |  |
| --- | --- | --- |
| BRANCH\_ID | NUMBER(10) | Not Null PK |
| BRANCH\_NAME | VARCHAR2(30) |  |
| BRANCH\_STREET | VARCHAR2(30) |  |
| BRANCH\_CITY | VARCHAR2(20) |  |
| BRANCH\_STATE | VARCHAR2(2) |  |
| BRANCH\_ZIP5 | VARCHAR2(5) |  |
| PHONE\_TOLLFREE | VACHAR2(12) |  |
| PHONE\_LOCAL | VARCHAR2(12) |  |
| BMEMBER\_ID | NUMBER(10) | FK |

**Account\_History**

|  |  |  |  |
| --- | --- | --- | --- |
| HISTORY\_ID | BALANCE\_DATE | BALANCE\_AMOUNT | TRANSACTION\_ID |

|  |  |  |
| --- | --- | --- |
| HISTORY\_ID | NUMBER(10) | Not Null PK |
| BALANCE\_DATE | DATE |  |
| BALANCE\_AMOUNT | NUMBER(10,2) |  |
| TRANSACTION\_ID | NUMBER(10) | FK |

# 

## Data Definition Language (DDL)

SPOOL /class/db668a/02/finalproject\_extract.txt;

set serveroutput on;   
set linesize 155;  
set pagesize 35;   
  
--DROP STATEMENTS--

drop table customer cascade constraints purge;

drop table account\_type cascade constraints purge;

drop table transaction cascade constraints purge;

drop table bank\_statement cascade constraints purge;

drop table account\_history cascade constraints purge;

drop table board\_director cascade constraints purge;

drop table branch cascade constraints purge;

drop table employee cascade constraints purge;

drop sequence SEQ\_CUSTOMERID;

drop sequence SEQ\_ACCOUNTID;

drop sequence SEQ\_TRANSACTIONID;

drop sequence SEQ\_STATEMENTID;

drop sequence SEQ\_HISTORYID;

drop sequence SEQ\_BOARDID;

drop sequence SEQ\_BRANCHID;

drop sequence SEQ\_EMPLOYEEID;

--CREATION OF CUSTOMER TABLE AND SPOOL RESULTS--

create table customer

(customer\_id number(10),

fname varchar2(30),

lname varchar2(30),

middle\_initial varchar2(1),

street\_address varchar2(35),

city varchar2(20),

state varchar2(2),

zip5 varchar2(5),

phone varchar2(12),

email\_address varchar2(50),

user\_name varchar2(30),

constraint customerid\_pk primary key (customer\_id));

*SQL> desc customer;*

*Name Null? Type*

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

*CUSTOMER\_ID NOT NULL NUMBER(10)*

*FNAME VARCHAR2(30)*

*LNAME VARCHAR2(30)*

*MIDDLE\_INITIAL VARCHAR2(1)*

*STREET\_ADDRESS VARCHAR2(35)*

*CITY VARCHAR2(20)*

*STATE VARCHAR2(2)*

*ZIP5 VARCHAR2(5)*

*PHONE VARCHAR2(12)*

*EMAIL\_ADDRESS VARCHAR2(50)*

*USER\_NAME VARCHAR2(30)*

--CREATION OF ACCOUNT\_TYPE TABLE AND SPOOL RESULTS--

create table account\_type

(account\_id number(10),

account\_type\_name varchar2(30),

account\_type\_code varchar2(30),

customer\_id number(10),

constraint account\_type\_pk primary key (account\_id),

constraint cutomerid\_fk foreign key (customer\_id) references customer(customer\_id));

*SQL> desc account\_type*

*Name Null? Type*

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

*ACCOUNT\_ID NOT NULL NUMBER(10)*

*ACCOUNT\_TYPE\_NAME VARCHAR2(30)*

*ACCOUNT\_TYPE\_CODE VARCHAR2(30)*

*CUSTOMER\_ID NUMBER(10)*

--CREATION OF BANK\_STATEMENT TABLE AND SPOOL RESULTS--

create table bank\_statement

(statement\_id number(10),

statement\_number VARCHAR2(10),

statement\_name varchar2(60),

statement\_date date,

customer\_id number(10),

constraint bank\_statement\_pk primary key (statement\_id),

constraint customerid\_statement\_fk foreign key (customer\_id) references customer(customer\_id));

*SQL> desc bank\_statement*

*Name Null? Type*

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

*STATEMENT\_ID NOT NULL NUMBER(10)*

*STATEMENT\_NUMBER VARCHAR2(10)*

*STATEMENT\_NAME VARCHAR2(60)*

*STATEMENT\_DATE DATE*

*CUSTOMER\_ID NUMBER(10)*

--CREATION OF BOARD\_DIRECTOR TABLE AND SPOOL RESULTS--

create table board\_director

(bmember\_id number(10),

bdfname varchar2(30),

bdlname varchar2(35),

bd\_phone varchar2(12),

director\_level varchar(12),

location varchar2(50),

constraint board\_pk primary key (bmember\_id));

*SQL> desc board\_director;*

*Name Null? Type*

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

*BMEMBER\_ID NOT NULL NUMBER(10)*

*BDFNAME VARCHAR2(30)*

*BDLNAME VARCHAR2(35)*

*BD\_PHONE VARCHAR2(12)*

*DIRECTOR\_LEVEL VARCHAR2(12)*

*LOCATION VARCHAR2(50)*

--CREATION OF BRANCH TABLE AND SPOOL RESULTS--

create table branch

(branch\_id number(10),

branch\_name varchar2(30),

branch\_street varchar2(30),

branch\_city varchar2(20),

branch\_state varchar2(2),

branch\_zip5 varchar2(5),

phone\_tollfree varchar2(12),

phone\_local varchar2(12),

bmember\_id number(10),

constraint branch\_pk primary key (branch\_id),

constraint bdmember\_id\_tran\_fk foreign key (bmember\_id) references board\_director(bmember\_id));

*SQL> desc branch;*

*Name Null? Type*

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

*BRANCH\_ID NOT NULL NUMBER(10)*

*BRANCH\_NAME VARCHAR2(30)*

*BRANCH\_STREET VARCHAR2(30)*

*BRANCH\_CITY VARCHAR2(20)*

*BRANCH\_STATE VARCHAR2(2)*

*BRANCH\_ZIP5 VARCHAR2(5)*

*PHONE\_TOLLFREE VARCHAR2(12)*

*PHONE\_LOCAL VARCHAR2(12)*

*BMEMBER\_ID NUMBER(10)*

--CREATION OF TRANSACTION TABLE AND SPOOL RESULTS--

create table transaction

(transaction\_id number(10),

transaction\_name varchar2(30),

transaction\_type varchar2(30),

transaction\_code varchar2(10),

transaction\_amount varchar2(35),

transaction\_begin\_date date,

transaction\_end\_date date,

customer\_id number(10),

statement\_id number(10),

account\_id number(10),

branch\_id number(10),

constraint transaction\_pk primary key (transaction\_id),

constraint customerid\_tran\_fk foreign key (customer\_id) references customer(customer\_id),

constraint statementid\_fk foreign key (statement\_id) references bank\_statement(statement\_id),

constraint accountid\_fk foreign key (account\_id) references account\_type(account\_id),

constraint branchid\_fk foreign key (branch\_id) references branch(branch\_id));

*SQL> desc transaction*

*Name Null? Type*

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

*TRANSACTION\_ID NOT NULL NUMBER(10)*

*TRANSACTION\_NAME VARCHAR2(30)*

*TRANSACTION\_TYPE VARCHAR2(30)*

*TRANSACTION\_CODE VARCHAR2(10)*

*TRANSACTION\_AMOUNT VARCHAR2(35)*

*TRANSACTION\_BEGIN\_DATE DATE*

*TRANSACTION\_END\_DATE DATE*

*CUSTOMER\_ID NUMBER(10)*

*STATEMENT\_ID NUMBER(10)*

*ACCOUNT\_ID NUMBER(10)*

*BRANCH\_ID NUMBER(10)*

--CREATION OF EMPLOYEE TABLE AND SPOOL RESULTS--

create table employee

(employee\_id number(10),

title varchar2(40),

hire\_date date,

salary number(10,2),

branch\_id number(10),

customer\_id number(10),

department varchar2(40),

constraint employeeid\_pk primary key (employee\_id),

constraint branchid\_emp\_fk foreign key (branch\_id) references branch(branch\_id),

constraint customerid\_emp\_fk foreign key (customer\_id) references customer(customer\_id));

*SQL> desc employee;*

*Name Null? Type*

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

*EMPLOYEE\_ID NOT NULL NUMBER(10)*

*TITLE VARCHAR2(40)*

*HIRE\_DATE DATE*

*SALARY NUMBER(10,2)*

*BRANCH\_ID NUMBER(10)*

*CUSTOMER\_ID NUMBER(10)*

*DEPARTMENT VARCHAR2(40)*

--CREATION OF ACCOUNT\_HSTORY TABLE AND SPOOL RESULTS--

create table account\_history

(history\_id number(10,2),

balance\_date date,

balance\_amount number(10,2),

transaction\_id number (10),

constraint history\_pk primary key (history\_id),

constraint transactionid\_history\_fk foreign key (transaction\_id) references transaction(transaction\_id));

*SQL> desc account\_history*

*Name Null? Type*

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

*HISTORY\_ID NOT NULL NUMBER(10,2)*

*BALANCE\_DATE DATE*

*BALANCE\_AMOUNT NUMBER(10,2)*

*TRANSACTION\_ID NUMBER(10)*

--CREATION OF SEQUENCES--

CREATE SEQUENCE seq\_customerid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_accountid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_transactionid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_statementid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_historyid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_boardid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_branchid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

CREATE SEQUENCE seq\_employeeid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

## Data Manipulation Language

--CUSTOMER TABLE INSERT STATEMENTS

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'WILMER','BRINKLEY', 'A','555 WISCONSIN AVE','BULL ROCK','MD','20888','555-333-4481','WBRINKLEY@AOL.COM','WBRINKLEY02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'WOODY','CHUCK', 'B','333 ALABAMA AVE','TOAD MEAD','VA','20881','555-222-4482','WCHUCK@AOL.COM','WCHUCK02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'BRADY','BROWN', 'C','444 CAROLINA AVE','GINGER HILL','TX','20882','555-111-4483','BBROWN@ABC.COM','BBROWN02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'CAREY','CHRISTEN', 'D','222 MARYLAND AVE','BED ROCK','MD','20883','555-444-4484','CCHRISTEN@GMAIL.COM','CCHRISTEN02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'JOHN','SMITH', 'E','111 NEW YORK AVE','ROCK CREEK','VA','20884','555-555-4485','JSMITH@ABC.COM','JSMITH02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'MICHAEL','MARLON', 'F','667 DAKOTA AVE','FALLS CHURCH','ND','20885','555-665-4486','MMARLON@ABC.COM','MMARLON02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'EDGAR','WILMOTH', 'G','888 ARKANSAS AVE','GERMAN RUN','DC','20886','555-777-4487','EWILMOTH@AOL.COM','EWILMOTH02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'AXEL','FOLEY', 'H','777 MISSISSIPPI AVE','LILLY BECK','DE','20887','555-888-4488','AFOLEY@HOTMAIL.COM','AFOLEY02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'CARLOS','JEFFERSON', 'A','100 MICHIGAN AVE','FROST HEAD','MD','20801','555-999-4489','CJEFFERSON@YAHOO.COM','CJEFFERSON02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'DANIEL','MOSS', 'B','102 ILLIONOIS AVE','LITTLE MAC','CA','20803','555-000-4410','DMOSS@EXCITE.COM','DMOSS02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'MICHY','MARLEY', 'F','667 FALMOTH AVE','FALLS CHURCH','VA','20889','333-665-4486','MMARLEY@YAHOO.COM','MARLEY02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'CARLOS','SANTANA', 'M','999 ANSAS AVE','GERRY RUN','DC','20876','544-776-4487','CSANTANA@AOL.COM','CSANTANA02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'JASMINE','FOLEY', 'T','777 WYOMING AVE','LED BECK','CA','30885','455-878-4400','NICEANDFUNNY@AOL.COM','JFOLEY02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'CARLOS','SWEETEN', 'P','200 GANTE AVE','WINK HEAD','MD','27701','555-888-4489','SWEETEN-CARLOS@EXCITE.COM','CSWEETEN02');

insert into customer values (SEQ\_CUSTOMERID.NEXTVAL,'WILLIAM','WILSON', 'R','102 KENTUCKY AVE','BIG MAC','IA','27803','555-330-9010','WWILSON@AOL.COM','WWILSON02');

--BOARD\_DIRECTOR TABLE INSERT STATEMENTS

insert into BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'CRAIG','DOUGLAS','555-550-4440','SENIOR','MID ATLANTIC');

insert into BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MARGE','SIMMS','555-500-4550','SOPHOMORE','SOUTH EAST');

insert into BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MOODY','FYRE','555-550-4440','JUNIOR','PACIFIC');

insert into BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MEL','DOULS','555-990-4441','APPRENTICE','CENTRAL');

--BRANCH TABLE INSERT STATEMENTS

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC SILVER ROCK','123 CANADA STREET','GOLD ROD','DC','22551','800-559-4449','555-12-5000',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC GOLD ROCK','123 GINGER HILL STREET','SILVER ROD','MD','22552','800-558-4448','555-12-4000',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC BRONZ ROCK','123 BUD ROCK STREET','FRAME ROD','NC','22553','800-557-4447','555-12-3000',3);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC PLATNUM ROCK','123 DEDICATION ROAD','MEADOW ROD','FL','22554','800-556-4446','555-12-2000',3);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC ALUMINUM ROCK','123 WYMIL STREET','PEAR TREE','CA','22555','800-555-4445','555-12-1000',4);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC METAL ROCK','123 TUNA MELT STREET','MILTON','DC','22556','800-554-4444','555-12-9000',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC STONE ROCK','123 SEPTEMBER AVE','SILVER SPRING','ND','22557','800-553-4442','555-12-8000',5);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC ZINC ROCK','123 DUNKIN DRIVE','GERMANTOWN','MD','22558','800-552-4443','555-12-7000',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC BRICK ROCK','123 PUMPERNICKEL BOULEVARD','BISMARCK','VA','22559','800-551-4441','555-12-6000',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC DIAMOND ROCK','123 CALMING TERRACE','GOLD ROD','DE','22560','800-550-4440','555-12-3999',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC WATER ROCK','123 WOODLAND DRIVE','GOLD ROD','PA','22261','800-555-4439','555-12-3998',2);

insert into BRANCH Values (SEQ\_BRANCHID.NEXTVAL, 'ABC HEADQUARTERS','223 WOOD BECK WAY','GOLDCLIFF','MD','24260','800-555-4000','555-12-3000',2);

--EMPLOYEE TABLE INSERT STATEMENTS

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'MARKETING MANAGER','01-JAN-99','60000.00',5,3, 'MARKETING');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'TELLER','01-JAN-99','37000.00',5,5,'CUSTOMER SERVICE');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'TELLER SUPERVISOR','01-JAN-99','95000.00',4,6,'CUSTOMER SERVICE');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'CHIEF EXECUTIVE OFFICER','01-JAN-99','95000.00',12,9,'HEADQUARTERS');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'UNDERWRITER','01-JAN-99','75000.00',2,2,'HEADQUARTERS');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'TELLER','01-JAN-99','27000.00',10,10,'CUSTOMER SERVICE');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'DATABASE ADMIN','01-JAN-99','85000.00',2,8,'INFORMATION TECHNOLOGY');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'TELLER','01-JAN-99','35000.00',9,11,'CUSTOMER SERVICE');

insert into employee values (SEQ\_EMPLOYEEID.NEXTVAL,'HR MANAGER','01-JAN-99','85000.00',12,15,'HUMAN RESOURCES');

--ACCOUNT\_TYPE TABLE INSERT STATEMENTS

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR CHECKING','RC0002',11);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR SAVINGS','RS0001',11);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'CREDIT\_CARD','CC0001',14);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'ROTH\_IRA','RI0001',2);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'CERTIFICATE OF DEPOSIT','CD0001',3);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'MONEY MARKET','MM0001',4);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'MORTGAGE LOAN','ML0002',5);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'PERSONAL LOAN','PL0003',6);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'AUTO LOAN','AL0001',7);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR CHECKING','RC0002',8);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR SAVINGS','RS0001',9);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'CREDIT\_CARD','CC0001',10);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'ROTH\_IRA','RI0001',3);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR CHECKING','CD0001',2);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'REGULAR SAVINGS','MM0001',10);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'CREDIT CARD','ML0002',7);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'PERSONAL LOAN','PL0003',3);

insert into account\_type values (SEQ\_ACCOUNTID.NEXTVAL,'AUTO LOAN','AL0001',13);

--BANK STATEMENT TABLE INSERT STATEMENTS

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0001','REGULAR CHECKING ACCOUNT STATEMENT','30-JAN-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0002','REGULAR SAVINGS ACCOUNT STATEMENT','30-JAN-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0003','CREDIT CARD ACCOUNT STATEMENT','30-JAN-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0004','ROTH IRA ACCOUNT STATEMENT','30-JAN-2012',3);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0005','CERTIFICATE OF DEPOSIT ACCOUNT STATEMENT','30-JAN-2012',4);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0006','MONEY MARKET ACCOUNT STATEMENT','30-JAN-2012',5);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0007','MORTGAGE LOAN ACCOUNT STATEMENT','30-JAN-2012',6);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0008','PERSONAL LOAN ACCOUNT STATEMENT','30-JAN-2012',7);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0009','AUTO LOAN ACCOUNT STATEMENT','30-JAN-2012',8);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0010','REGULAR CHECKING STATEMENT','30-JAN-2012',9);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0011','REGULAR SAVINGS STATEMENT','30-JAN-2012',10);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0012','CREDIT CARD ACCOUNT STATEMENT','30-JAN-2012',11);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0013','ROTH IRA ACCOUNT STATEMENT','30-JAN-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0014','REGULAR CHECKING ACCOUNT STATEMENT','30-JAN-2012',3);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0015','REGULAR SAVINGS ACCOUNT STATEMENT','30-JAN-2012',11);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0016','CREDIT CARD ACCOUNT STATEMENT','30-JAN-2012',8);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0017','PERSONAL LOAN ACCOUNT STATEMENT','30-JAN-2012',4);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0018','AUTO LOAN ACCOUNT STATEMENT','30-JAN-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0001','REGULAR CHECKING ACCOUNT STATEMENT','28-FEB-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0002','REGULAR SAVINGS ACCOUNT STATEMENT','28-FEB-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0003','CREDIT CARD ACCOUNT STATEMENT','28-FEB-2012',2);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0004','ROTH IRA ACCOUNT STATEMENT','28-FEB-2012',3);

insert into BANK\_STATEMENT Values (SEQ\_STATEMENTID.NEXTVAL, 'S0005','CERTIFICATE OF DEPOSIT ACCOUNT STATEMENT','28-FEB-2012',4);

--TRANSACTION TABLE INSERT STATEMENTS

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS WITHDRAWAL','WITHDRAWAL','W00001','300.00','10-JAN-12','10-JAN-12',3,2,3,3);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS DEPOSIT','DEPOSIT','D00001','1300.00','10-JAN-12', '10-JAN-12',5,5,3,2);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS TRANSFER','FUND TRANSFER','T00001','250.00','11-JAN-12', '11-JAN-12',2,3,4,3);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CREDIT CARD PAYMENT','CARD PAYMENT','P00001','50.00','10-FEB-12', '10-FEB-12',3,5,8,4);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS DEPOSIT','DEPOSIT','D00001','349.00','12-JAN-12','12-JAN-12',7,2,5,5);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS INQUIRY','INQUIRY','W00001','344.00','30-JAN-12','30-JAN-12',4,4,2,6);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CHECKING WITHDRAWAL','WITHDRAWAL','W00001','30.00' ,'30-JAN-12', '10-JAN-12',6,3,4,7);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CHECKING INQUIRY','INQUIRY','I00001','300.00', '15-JAN-12', '15-JAN-12',3,7,2,10);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CASH ADVANCE','ADVANCE','A00001','200.00','10-FEB-12','10-FEB-12',2,3,8,11);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'LINE OF CREDIT TRANSFER','TRANSFER','T00001','400.00','10-FEB-12','10-FEB-12',9,5,2,3);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CHECKING DEPOSIT','DEPOSIT','D00001','500.00','11-SEP-12', '11-SEP-12',10,5,2,2);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CHECKING TRANSFER','TRANSFER','T00001','100.00','30-SEP-12', '30-SEP-12',11,6,11,3);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS DEPOSIT','DEPOSIT','D00001','900.00','14-FEB-12', '14-FEB-12',3,3,2,3);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'LOAN PAYMENT','PAYMENT','P00001','100.00','15-JAN-12','15-JAN-12',5,3,4,5);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'OVERDRAFT PROTECTON PAYMENT','PAYMENT','P00001','25.00','11-SEP-12','11-SEP-12',6,5,4,2);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS WITHDRAWAL','WITHDRAWAL','W00001','150.00', '11-FEB-12', '11-FEB-12',6,5,8,5);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS WITHDRAWAL','WITHDRAWAL','W00001','120.00' ,'12-JAN-12', '12-JAN-12',6,2,5,8);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'SAVINGS DEPOSIT','DEPOSIT','D00001','30.25','12-APR-12', '12-APR-12',8,9,5,2);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CHECKING WITHDRAWAL','WITHDRAWAL','W00001','75.00' ,'10-FEB-12', '10-FEB-12',6,2,5,4);

insert into transaction values (SEQ\_TRANSACTIONID.NEXTVAL,'CASH ADVANCE','ADVANCE','A00001','45.75', '10-JAN-12', '10-JAN-12',7,2,13,3);

--ACCOUNT HISTORY TABLE INSERT STATEMENTS

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','4750.00',2);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','3750.00',3);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','1050.00',4);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'13-FEB-12','1150.00',5);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'14-JAN-12','1350.00',6);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'15-JAN-12','1750.00',7);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'30-JAN-12','550.00',8);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-SEP-12','750.00',9);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-FEB-12','4500.00',10);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','4000.00',11);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','7500.00',12);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','40.00',13);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','300.00',14);

insert into account\_history values (SEQ\_HISTORYID.NEXTVAL,'10-JAN-12','5550.00',15);

## Creating User Accounts

--STATEMENT STATEMENTS FOR ALL USERS

*drop user wbrinkley02 cascade;*

*drop user wchuck02 cascade;*

*drop user bbrown02 cascade;*

*drop user cchristen02 cascade;*

*drop user jsmith02 cascade;*

*drop user mmarlon02 cascade;*

*drop user ewilmoth02 cascade;*

*drop user afoley02 cascade;*

*drop user cjefferson02 cascade;*

*drop user dmoss02 cascade;*

*drop user marley02 cascade;*

*drop user csantana02 cascade;*

*drop user jfoley02 cascade;*

*drop user csweeten02 cascade;*

*drop user wwilson02 cascade;*

--Drop Board of Directors Users if they already exist--

*drop user craig cascade;*

*drop user marge cascade;*

*drop user moody cascade;*

*drop user mel cascade;*

--CREATE ALL USER ACCOUNTS

**Description**: user accounts are created using strong passwords. Employees are also account holders; therefore, employee accounts are included in the scripts provided.

**Security Policy**: Passwords should be at least 8 characters long and should include at least one uppercase letter, at least one number, at least lowercase letter and at least one number.

*create user wbrinkley02 identified by Wink02\_778;*

*create user wchuck02 identified by Chunky03\_888;*

*create user bbrown02 identified by Bronz02\_789;*

*create user cchristen02 identified by Craven87\_878;*

*create user jsmith02 identified by Smelt22\_800;*

*create user mmarlon02 identified by Marti15\_123;*

*create user ewilmoth02 identified by SWeltz16\_423;*

*create user afoley02 identified by Foxy34\_241;*

*create user cjefferson02 identified by Cjxxxx\_123;*

*create user dmoss02 identified by Dmooo12\_456;*

*create user marley02 identified by Marmar2\_123;*

*create user csantana02 identified by Santacruz1\_999;*

*create user jfoley02 identified by Jazzy12\_333;*

*create user csweeten02 identified by CSweething02\_999;*

*create user wwilson02 identified by Wweewee9\_888;*

*--CREATING THE BOARD OF DIRECTOR USERS--*

*create user craig identified by Craten\_999;*

*create user marge identified by Marvy\_333;*

*create user moody identified by Moofy\_999;*

*create user mel identified by Mevie\_888;*

## Creating Roles

--CREATE CUSTOMER AND EMPLOYEE ROLES

*drop role customer\_role;*

*drop role employee\_role;*

*drop role senior\_management\_role;*

*drop role supervisor\_role;*

*drop role all\_customer\_role;*

*drop role hr\_role;*

*drop role dba\_role;*

*create role customer\_role;*

*create role employee\_role;*

*create role senior\_management\_role;*

*create role supervisor\_role;*

*create role hr\_role;*

*create role all\_customer\_role;*

*create role dba\_role;*

*--CREATE BOARD OF DIRECTORS ROLE--*

*drop role director\_role;*

*create role director\_role;*

*Grant director\_role to craig, marge, moody, mel;*

*Grant CREATE SESSION to director\_role;*

## Security Policies and Granting System Privileges

**Security Policy:** All users should be able to connect to the database.

Description**:** senior\_management and supervisors are also members of the employee group; therefore, they will be granted Create Session when that privilege is granted to the employee role.

*Grant Create Session to customer\_role;*

*Grant Create Session to employee\_role;*

*Grant connect, resource to senior\_management\_role with admin option;*

*Grant connect, resource to dba\_role with admin option;*

**Security Policy:** The customer\_role should be granted to all individuals who are not employees and the employee role should be granted to all individuals who are employees. All customers including employees should be granted the all\_customer\_role.

*Grant customer\_role to dmoss02, marley02, csantana02, jfoley02, csweeten02, wwilson02;*

*Grant employee\_role to wbrinkley02, wchuck02, bbrown02, cchristen02, jsmith02, mmarlon02, ewilmoth02, afoley02, cjefferson02;*

Description: The individuals who are a member of the senior\_management group should be granted the senior\_management\_role and those who work in human resources should be granted the hr\_role. In addition, the dba\_role should be granted to the database administrator.

*Grant senior\_management\_role to cchristen02;*

*Grant hr\_role to csweeten02;*

*Grant dba\_role to ewilmoth02;*

**Security Policy**: The database administrator should be granted select, insert, update on all tables.

*Grant INSERT, UPDATE, DELETE on customer to dba\_role;*

*Grant INSERT, UPDATE, DELETE on employee to dba\_role;*

*Grant INSERT, UPDATE, DELETE on account\_type to dba\_role;*

*Grant INSERT, UPDATE, DELETE on transaction to dba\_role;*

*Grant INSERT, UPDATE, DELETE on account\_history to dba\_role;*

*Grant INSERT, UPDATE, DELETE on bank\_statement to dba\_role;*

*Grant INSERT, UPDATE, DELETE on board\_director to dba\_role;*

*Grant INSERT, UPDATE, DELETE on branch to dba\_role;*

**Security Policy**: Employees should be able to view all customer records.

*Grant SELECT on customer to customer\_role;*

### Using Views to Manage Restrictions

**Security Policy:** Each customer should be able to view their own information in the customer table and each employee should be able to view their own record in the employee table.

Description: A view is created that allows CUSTOMER who are not employees to see only their own records in the customer table. The view is called “Own\_CUSTOMER\_Record”. Another view is created that allows EMPLOYEE to see only their own records. That view is called “Own\_EMPLOYEE\_Record”.

*connect db668a02/securityclass27;*

*CREATE OR REPLACE VIEW Own\_Customer\_Record*

*As*

*Select \* FROM CUSTOMER*

*WHERE( user\_name =*

*SYS\_CONTEXT ('USERENV', 'SESSION\_USER') OR*

*user = 'DB668A02') and customer\_id not in (select c.customer\_id from customer c, employee e*

*where c.customer\_id = e.employee\_id)*

*with check option;*

*CREATE OR REPLACE VIEW Own\_Employee\_Record*

*As*

*Select employee\_id, title, hire\_date, salary,*

*branch\_id, c.customer\_id, fname, lname, middle\_initial,*

*street\_address, city, state, zip5, phone, email\_address,*

*user\_name FROM EMPLOYEE E, CUSTOMER C*

*WHERE C.CUSTOMER\_ID = E.EMPLOYEE\_ID*

*AND (c.user\_name =*

*SYS\_CONTEXT ('USERENV', 'SESSION\_USER') OR*

*user = 'DB668A02')*

*with check option;*

**Description:** The customer\_role and employee\_role are granted SELECT privilege on the newly created Own\_Customer\_Record and Own\_Employee\_Record views respectively.

*connect db668a02/securityclass27;*

*Grant select on Own\_Customer\_Record to customer\_role;*

*Grant select on Own\_Employee\_Record to employee\_role;*

*--TESTING THE OWN\_CUSTOMER\_RECORD VIEW--*

*connect dmoss02/Dmooo12\_456;*

*show user;*

*select \* from db668a02.own\_customer\_record;*

*--SPOOL RESULTS--*

*SQL> connect dmoss02/Dmooo12\_456;*

*Connected.*

*SQL> show user;*

*USER is "DMOSS02"*

*SQL> select \* from db668a02.own\_customer\_record;*

*CUSTOMER\_ID FNAME LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

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

*EMAIL\_ADDRESS USER\_NAME*

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

*11 DANIEL MOSS B 102 ILLIONOIS AVE LITTLE MAC CA 20803 555-000-4410*

*DMOSS@EXCITE.COM DMOSS02*

*--USER MARLEY02 IS UNABLE TO VIEW ANOTHER USER'S RECORD--*

*connect marley02/Marmar2\_123;*

*select \* from db668a02.own\_customer\_record*

*where user\_name = '* *DMOSS02';*

*--SPOOL RESULTS--*

*SQL> connect marley02/Marmar2\_123;*

*Connected.*

*SQL> select \* from db668a02.own\_customer\_record*

*2 where user\_name = ' DMOSS02';*

*no rows selected*

*--TESTING THE OWN\_EMPLOYEE\_RECORD VIEW--*

*connect wchuck02/Chunky03\_888;*

*show user;*

*select \* from db668a02.own\_employee\_record;*

*--SPOOL RESULTS--*

*SQL> connect wchuck02/Chunky03\_888;*

*Connected.*

*SQL> show user;*

*USER is "WCHUCK02"*

*SQL> select \* from db668a02.own\_employee\_record;*

*EMPLOYEE\_ID TITLE HIRE\_DATE SALARY BRANCH\_ID CUSTOMER\_ID FNAME*

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

*LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

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

*EMAIL\_ADDRESS USER\_NAME*

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

*3 TELLER 01-JAN-99 37000 5 3 WOODY*

*CHUCK B 333 ALABAMA AVE TOAD MEAD VA 20881 555-222-4482*

*WCHUCK@AOL.COM WCHUCK02*

*--USER BBROWN02 IS UNABLE TO ACCESS ANOTHER EMPLOYEE'S RECORD--*

*connect bbrown02/ Bronz02\_789;*

*select \* from db668a02.own\_employee\_record*

*where user\_name = '* *WBRINKELY02';*

*--SPOOL RESULTS--*

*SQL> connect bbrown02/ Bronz02\_789;*

*Connected.*

*SQL> select \* from db668a02.own\_employee\_record*

*2 where user\_name = ' WBRINKELY02';*

*no rows selected*

**Security Policy**: Each customer and employee should be able to update their own name, address, phone number and email address only. Neither customer nor employee can change the username, used for online access, once it is created. Unique identifiers cannot be changed by users. Employee cannot change their own salary, hire date and title.

*connect db668a02/securityclass27;*

*show user;*

*Grant update (fname, lname, middle\_initial, street\_address, city, state, zip5, phone, email\_address) on Own\_Customer\_Record to customer\_role;*

*Grant update (fname, lname, middle\_initial, street\_address, city, state, zip5, phone, email\_address) on Own\_Employee\_Record to employee\_role;*

*--TESTING CUSTOMER UPDATE--*

*connect dmoss02/Dmooo12\_456;*

*show user;*

*update db668a02.own\_customer\_record*

*set street\_address = '155 CARNEKIE BOULEVARD'*

*where customer\_id = 11;*

*select \* from db668a02.own\_customer\_record;*

*--SPOOL RESULTS--*

*SQL> connect dmoss02/Dmooo12\_456;*

*Connected.*

*SQL> show user;*

*USER is "DMOSS02"*

*SQL> update db668a02.own\_customer\_record*

*2 set street\_address = '155 CARNEKIE BOULEVARD'*

*3 where customer\_id = 11;*

*1 row updated.*

*SQL> select \* from db668a02.own\_customer\_record;*

*CUSTOMER\_ID FNAME LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

*----------- ------------------------------ ------------------------------ - ----------------------------------- EMAIL\_ADDRESS USER\_NAME*

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

*11 DANIEL MOSS B 155 CARNEKIE BOULEVARD LITTLE MAC CA 20803 555-000-4410*

*DMOSS@EXCITE.COM DMOSS02*

*--USER DMOSS02 IS UNABLE TO UDPATE THE STREET\_ADDRESS FOR ANOTHER USER WITH CUSTOMER\_ID = 13--*

*update db668a02.own\_customer\_record*

*set street\_address = '155 FRIDAY JOURNEY'*

*where customer\_id = 13;*

*--SPOOL RESULTS--*

*SQL> update db668a02.own\_customer\_record*

*2 set street\_address = '155 FRIDAY JOURNEY'*

*3 where customer\_id = 13;*

*0 rows updated.*

*--CUSTOMER DMOSS02 IS UNABLE O UPDATE THE USER\_NAME FIELD FOR HIS RECORD--*

*update db668a02.own\_customer\_record*

*set user\_name = 'STEDWICK03'*

*where customer\_id = 11;*

*--SPOOL RESULTS--*

*SQL> update db668a02.own\_customer\_record*

*2 set user\_name = 'STEDWICK03'*

*3 where customer\_id = 11;*

*update db668a02.own\_customer\_record*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

*--TESTING EMPLOYEE UPDATE--*

*connect afoley02/Foxy34\_241;*

*show user;*

*update db668a02.own\_employee\_record*

*set phone = '444-333-2222'*

*where customer\_id = 9;*

*select \* from db668a02.own\_employee\_record;*

*SQL> connect afoley02/Foxy34\_241;*

*Connected.*

*SQL> show user;*

*USER is "AFOLEY02"*

*SQL> update db668a02.own\_employee\_record*

*2 set phone = '444-333-2222'*

*3 where customer\_id = 9;*

*1 row updated.*

*--SPOOL RESULTS--*

*SQL> select \* from db668a02.own\_employee\_record;*

*EMPLOYEE\_ID TITLE HIRE\_DATE SALARY BRANCH\_ID CUSTOMER\_ID FNAME*

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

*LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

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

*EMAIL\_ADDRESS USER\_NAME*

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

*9 TELLER 01-JAN-99 35000 9 9 AXEL*

*FOLEY H 777 MISSISSIPPI AVE LILLY BECK DE 20887 444-333-2222*

*AFOLEY@HOTMAIL.COM AFOLEY02*

*--USER AXEL02 IS UNABLE TO UDPATE THE PHONE NUMBER OF ANOTHER EMPLOYEE--*

*update db668a02.own\_employee\_record*

*set phone = '777-303-2299'*

*where customer\_id = 12;*

*--SPOOL RESULTS--*

*SQL> update db668a02.own\_employee\_record*

*2 set phone = '777-303-2299'*

*3 where customer\_id = 12;*

*0 rows updated.*

*--USER AXEL02 IS UNABLE TO UPDATE THE SALARY INFORMATION IN HIS OWN RECORD--*

*update db668a02.own\_employee\_record*

*set salary = 95000*

*where customer\_id = 9;*

*--SPOOL RESULTS--*

*SQL> update db668a02.own\_employee\_record*

*2 set salary = 95000*

*3 where customer\_id = 9;*

*update db668a02.own\_employee\_record*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

**Security Policy:** Human Resources should be able to view all employee personal details including employee\_id, name, address, phone number, email address, title and salary.

*connect db668a02/securityclass27;*

*CREATE OR REPLACE VIEW HR\_Employee\_Record*

*As*

*Select employee\_id, title, hire\_date, salary,*

*branch\_id, c.customer\_id, fname, lname, middle\_initial,*

*street\_address, city, state, zip5, phone, email\_address,*

*user\_name FROM EMPLOYEE E, CUSTOMER C*

*WHERE C.CUSTOMER\_ID = E.CUSTOMER\_ID;*

Description: The hr\_role is granted SELECT privilege on the newly created HR\_Employee\_Record view.

*show user;*

*Grant select on HR\_Employee\_Record to hr\_role;*

*Grant hr\_role to csweeten02;*

*--USER CSWEETEN02 WHO IS A MEMBER OF THE HR DEPT CAN VIEW HR\_EMPLOYEE\_RECORD--*

*connect csweeten02/CSweething02\_999;*

*show user;*

*select \* from db668a02.HR\_Employee\_Record;*

*--SPOOL RESULTS--*

*SQL> connect csweeten02/CSweething02\_999;*

*Connected.*

*SQL> show user;*

*USER is "CSWEETEN02"*

*SQL> select \* from db668a02.HR\_Employee\_Record;*

*EMPLOYEE\_ID TITLE HIRE\_DATE SALARY BRANCH\_ID CUSTOMER\_ID FNAME*

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

*LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

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

*EMAIL\_ADDRESS USER\_NAME*

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

*6 UNDERWRITER 01-JAN-99 75000 2 2 WILMER*

*BRINKLEY A 555 WISCONSIN AVE BULL ROCK MD 20888 555-333-4481*

*WBRINKLEY@AOL.COM WBRINKLEY02*

*2 MARKETING MANAGER 01-JAN-99 60000 5 3 WOODY*

*CHUCK B 333 ALABAMA AVE TOAD MEAD VA 20881 555-222-4482*

*WCHUCK@AOL.COM WCHUCK02*

*3 TELLER 01-JAN-99 37000 5 5 CAREY*

*CHRISTEN D 222 MARYLAND AVE BED ROCK MD 20883 555-444-4484*

*CCHRISTEN@GMAIL.COM CCHRISTEN02*

*4 TELLER SUPERVISOR 01-JAN-99 95000 4 6 JOHN*

*SMITH E 111 NEW YORK AVE ROCK CREEK VA 20884 555-555-4485*

*JSMITH@ABC.COM JSMITH02*

*9 rows selected.*

--USER JOHN SMITH IS UNABLE TO VIEW THE HR\_EMPLOYEE\_RECORD view--

*connect jsmith02/Smelt22\_800;*

*show user;*

*select \* from db668a02.HR\_Employee\_Record;*

*--SPOOL RESULTS--*

*SQL> connect jsmith02/Smelt22\_800;*

*Connected.*

*SQL> show user;*

*USER is "JSMITH02"*

*SQL> select \* from db668a02.HR\_Employee\_Record;*

*select \* from db668a02.HR\_Employee\_Record*

*\**

*ERROR at line 1:*

*ORA-00942: table or view does not exist*

**Security Policy:** The Human Resources department should be able to update the name, address, email address, title, salary, and phone number of all employee records. HR cannot update the employees’ user name used for online access.

*connect db668a02/securityclass27;*

*show user;*

*Grant update (fname, lname, middle\_initial, street\_address, city, state, zip5, phone, email\_address, title, hire\_date, salary) on db668a02.HR\_Employee\_Record to hr\_role;*

*connect csweeten02/CSweething02\_999;*

*update db668a02.HR\_Employee\_Record*

*set salary = 40000*

*where employee\_id = 3;*

*select \* from**db668a02.HR\_Employee\_Record;*

*SQL> connect csweeten02/CSweething02\_999;*

*Connected.*

*SQL> update db668a02.HR\_Employee\_Record*

*2 set salary = 40000*

*3 where employee\_id = 3;*

*1 row updated.*

*SQL> select \* from db668a02.HR\_Employee\_Record;*

*EMPLOYEE\_ID TITLE HIRE\_DATE SALARY BRANCH\_ID CUSTOMER\_ID FNAME*

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

*LNAME M STREET\_ADDRESS CITY ST ZIP5 PHONE*

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

*EMAIL\_ADDRESS USER\_NAME*

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

*6 UNDERWRITER 01-JAN-99 75000 2 2 WILMER*

*BRINKLEY A 555 WISCONSIN AVE BULL ROCK MD 20888 555-333-4481*

*WBRINKLEY@AOL.COM WBRINKLEY02*

*2 MARKETING MANAGER 01-JAN-99 60000 5 3 WOODY*

*CHUCK B 333 ALABAMA AVE TOAD MEAD VA 20881 555-222-4482*

*WCHUCK@AOL.COM WCHUCK02*

*3 TELLER 01-JAN-99 40000 5 5 CAREY*

*CHRISTEN D 222 MARYLAND AVE BED ROCK MD 20883 555-444-4484*

*CCHRISTEN@GMAIL.COM CCHRISTEN02*

*9 rows selected.*

**--**THE HR MANAGER CSWEETEN02 IS UNABLE TO UPDATE THE USERNAME OF THE EMPLOYEES--

*update db668a02.HR\_Employee\_Record*

*set user\_name = 'DMOSSIE22'*

*where employee\_id = 9;*

***--SPOOL RESULTS****--*

*SQL> update db668a02.HR\_Employee\_Record*

*2 set user\_name = 'DMOSSIE22'*

*3 where employee\_id = 9;*

*update db668a02.HR\_Employee\_Record*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

**Security Policy**: Supervisors can view employees’ id, name, address, email address, phone number, hire date, branch\_id, user name and title. The supervisor cannot perform any type of update to the table.

*connect db668a02/securityclass27;*

*CREATE OR REPLACE VIEW Supervisor\_Employee\_Record*

*As*

*Select employee\_id, title, hire\_date,*

*branch\_id, c.customer\_id, fname, lname, middle\_initial,*

*street\_address, city, state, zip5, phone, email\_address,*

*user\_name FROM EMPLOYEE E, CUSTOMER C*

*WHERE C.CUSTOMER\_ID = E.CUSTOMER\_ID;*

Description: The supervisor\_role is granted SELECT privilege on the newly created Supervisor\_Employee\_Record view.

*show user;*

*Grant SELECT on Supervisor\_Employee\_Record to supervisor\_role;*

*Grant supervisor\_role to JSMITH02;*

*--THE SUPERVISORS CAN VIEW THE SUPERVISOR\_EMPLOYEE\_RECORD--*

*connect jsmith02/Smelt22\_800;*

*show user;*

*select \* from db668a02.Supervisor\_Employee\_Record;*

*--SPOOL RESULTS--*

*SQL> connect jsmith02/Smelt22\_800;*

*Connected.*

*SQL> show user;*

*USER is "JSMITH02"*

*SQL> select \* from db668a02.Supervisor\_Employee\_Record;*

*EMPLOYEE\_ID TITLE HIRE\_DATE BRANCH\_ID CUSTOMER\_ID FNAME LNAME M*

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

*STREET\_ADDRESS CITY ST ZIP5 PHONE EMAIL\_ADDRESS*

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

*USER\_NAME*

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

*6 UNDERWRITER 01-JAN-99 2 2 WILMER BRINKLEY A*

*555 WISCONSIN AVE BULL ROCK MD 20888 555-333-4481 WBRINKLEY@AOL.COM*

*WBRINKLEY02*

*2 MARKETING MANAGER 01-JAN-99 5 3 WOODY CHUCK B*

*333 ALABAMA AVE TOAD MEAD VA 20881 555-222-4482 WCHUCK@AOL.COM*

*WCHUCK02*

*3 TELLER 01-JAN-99 5 5 CAREY CHRISTEN D*

*222 MARYLAND AVE BED ROCK MD 20883 555-444-4484 CCHRISTEN@GMAIL.COM*

*CCHRISTEN02*

*9 rows selected.*

*--THE SUPERVISORS ARE UNABLE TO UPDATE THE SUPERVISOR\_EMPLOYEE\_RECORD VIEW--*

*update db668a02. Supervisor\_Employee\_Record*

*set lname = 'WESLEY'*

*where employee\_id = 3;*

*-*

*-SPOOL RESULTS--*

*SQL> update db668a02. Supervisor\_Employee\_Record*

*2 set lname = 'WESLEY'*

*3 where employee\_id = 3;*

*update db668a02. Supervisor\_Employee\_Record*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

**Security Policy**: Each customer should be able to view transaction detail associated with his or her own account. The customer should also be able to see the branch name from which the transaction took place and the account type name for the transaction.

*connect db668a02/securityclass27;*

*show user;*

*CREATE OR REPLACE VIEW Own\_Customer\_Transaction*

*As*

*Select c.customer\_id, b.branch\_name, c.fname, c.lname, c.middle\_initial,*

*c.street\_address, c.city, c.state, c.zip5, c.phone, c.email\_address,*

*c.user\_name, t.transaction\_name, t.transaction\_amount, t.transaction\_begin\_date, t.transaction\_end\_date, a.account\_type\_name*

*FROM CUSTOMER c, TRANSACTION t, BRANCH b, ACCOUNT\_TYPE a*

*WHERE c.customer\_id = t.customer\_id AND*

*t.branch\_id = b.branch\_id AND*

*t.account\_id = a.account\_id AND*

*(c.user\_name =*

*SYS\_CONTEXT ('USERENV', 'SESSION\_USER') OR*

*user = 'DB668A02')*

*with check option;*

Description: The all\_customer\_role is granted SELECT privilege on the newly created Own\_Customer\_Transactionview and the view is granted to all users.

*Grant SELECT on Own\_Customer\_Transaction to All\_Customer\_Role;*

*Grant All\_customer\_role to dmoss02, marley02, csantana02, jfoley02, csweeten02, wwilson02, wbrinkley02, wchuck02, bbrown02, cchristen02, jsmith02, mmarlon02, ewilmoth02, afoley02, cjefferson02;*

*--TEST SHOWING USER CCHRISTEN02 IS ABLE TO SEE ALL HER TRANSACTIONS--*

*connect cchristen02/Craven87\_878;*

*show user;*

*select \* from db668a02.Own\_Customer\_Transaction;*

**--*SPOOL RESULTS--***

*SQL> connect cchristen02/Craven87\_878;*

*Connected.*

*SQL> show user;*

*USER is "CCHRISTEN02"*

*SQL> select \* from db668a02.Own\_Customer\_Transaction;*

*CUSTOMER\_ID BRANCH\_NAME FNAME LNAME M STREET\_ADDRESS*

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

*CITY ST ZIP5 PHONE EMAIL\_ADDRESS USER\_NAME TRANSACTION\_NAME*

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

*TRANSACTION\_AMOUNT TRANSACTI TRANSACTI ACCOUNT\_TYPE\_NAME*

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

*5 ABC SILVER ROCK CAREY CHRISTEN D 222 MARYLAND AVE*

*BED ROCK MD 20883 555-444-4484 CCHRISTEN@GMAIL.COM CCHRISTEN02 SAVINGS DEPOSIT*

*1300.00 10-JAN-12 10-JAN-12 REGULAR SAVINGS*

*5 ABC PLATNUM ROCK CAREY CHRISTEN D 222 MARYLAND AVE*

*BED ROCK MD 20883 555-444-4484 CCHRISTEN@GMAIL.COM CCHRISTEN02 LOAN PAYMENT*

*100.00 15-JAN-12 15-JAN-12 CREDIT\_CARD*

*--CCHRISTEN02 IS UNABLE TO SEE THE TRANSACTIONS FOR OTHER CUSTOMERS--*

*select \* from db668a02.Own\_Customer\_Transaction*

*where customer\_id = 8;*

*--SPOOL RESULTS--*

*SQL> select \* from db668a02.Own\_Customer\_Transaction*

*2 where customer\_id = 8;*

*no rows selected*

**Security Policy**: Each employee should be able to view transaction detail associated with customer accounts that do not belong to other employees. From this view, the employee should be able to see his or her own set of transaction as well but not the transaction for other employees.

*connect db668a02/securityclass27;*

*CREATE OR REPLACE VIEW Customer\_Transaction*

*As*

*Select c.customer\_id, b.branch\_name, c.fname, c.lname, c.middle\_initial,*

*c.street\_address, c.city, c.state, c.zip5, c.phone, c.email\_address,*

*c.user\_name, t.transaction\_name, t.transaction\_amount, t.transaction\_begin\_date, t.transaction\_end\_date, a.account\_type\_name*

*FROM CUSTOMER c, TRANSACTION t, BRANCH b, ACCOUNT\_TYPE a*

*WHERE c.customer\_id = t.transaction\_id AND*

*t.branch\_id = b.branch\_id AND*

*t.account\_id = a.account\_id AND*

*c.customer\_id not in (select employee\_id from employee)*

*union*

*Select c.customer\_id, b.branch\_name, c.fname, c.lname, c.middle\_initial,*

*c.street\_address, c.city, c.state, c.zip5, c.phone, c.email\_address,*

*c.user\_name, t.transaction\_name, t.transaction\_amount, t.transaction\_begin\_date, t.transaction\_end\_date, a.account\_type\_name*

*FROM CUSTOMER c, TRANSACTION t, BRANCH b, ACCOUNT\_TYPE a*

*WHERE c.customer\_id = t.transaction\_id AND*

*t.branch\_id = b.branch\_id AND*

*t.account\_id = a.account\_id AND*

*(c.user\_name =*

*SYS\_CONTEXT ('USERENV', 'SESSION\_USER') OR*

*user = 'DB668A02')*

*with check option;*

Description: The employee\_role is granted SELECT privilege on the newly created Customer\_Transaction view.

*Grant SELECT on Customer\_Transaction to employee\_role;*

*--TEST: THE EMPLOYEE WBRINKLEY02 IS ABLE TO SEE ALL HIS TRANSACTIONS AS WELL AS TRANSACTIONS FOR NON-EMPLOYEES--*

*Connect WBRINKLEY02/Wink02\_778;*

*Select \* from db668a02.Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> Connect WBRINKLEY02/Wink02\_778;*

*Connected.*

*SQL> Select \* from db668a02.Customer\_Transaction;*

*CUSTOMER\_ID BRANCH\_NAME FNAME LNAME M STREET\_ADDRESS*

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

*CITY ST ZIP5 PHONE EMAIL\_ADDRESS USER\_NAME TRANSACTION\_NAME*

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

*TRANSACTION\_AMOUNT TRANSACTI TRANSACTI ACCOUNT\_TYPE\_NAME*

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

*2 ABC GOLD ROCK WILMER BRINKLEY A 555 WISCONSIN AVE*

*BULL ROCK MD 20888 555-333-4481 WBRINKLEY@AOL.COM WBRINKLEY02 SAVINGS WITHDRAWAL*

*300.00 10-JAN-12 10-JAN-12 REGULAR SAVINGS*

*11 ABC GOLD ROCK DANIEL MOSS B 155 CARNEKIE BOULEVARD*

*LITTLE MAC CA 20803 555-000-4410 DMOSS@EXCITE.COM DMOSS02 LINE OF CREDIT TRANSFER*

*400.00 10-FEB-12 10-FEB-12 REGULAR CHECKING*

*12 ABC SILVER ROCK MICHY MARLEY F 667 FALMOTH AVE*

*FALLS CHURCH VA 20889 333-665-4486 MMARLEY@YAHOO.COM MARLEY02 CHECKING DEPOSIT*

*500.00 11-SEP-12 11-SEP-12 REGULAR CHECKING*

*13 ABC GOLD ROCK CARLOS SANTANA M 999 ANSAS AVE*

*GERRY RUN DC 20876 544-776-4487 CSANTANA@AOL.COM CSANTANA02 CHECKING TRANSFER*

*100.00 30-SEP-12 30-SEP-12 REGULAR CHECKING*

7 rows selected.

--WBRINKLEY02 IS UNABLE TO SEE THE TRANSACTIONS FOR FELLOW EMPLOYEES--

*Select \* from db668a02. Customer\_Transaction*

*where customer\_id = 8;*

*--SPOOL RESULTS*

*SQL> Select \* from db668a02. Customer\_Transaction*

*2 where customer\_id = 8;*

*no rows selected*

**Security Policy**: DBA and Senior Manager should be able to view and update all customer transactions. E.g. if a transaction was made in error, only these individuals can view or reverse the transaction.

*connect db668a02/securityclass27;*

*CREATE OR REPLACE VIEW All\_Customer\_Transaction*

*As*

*Select c.customer\_id, b.branch\_name, c.fname, c.lname, c.middle\_initial,*

*c.street\_address, c.city, c.state, c.zip5, c.phone, c.email\_address,*

*c.user\_name, t.transaction\_name, t.transaction\_amount, t.transaction\_begin\_date, t.transaction\_end\_date, a.account\_type\_name*

*FROM CUSTOMER c, TRANSACTION t, BRANCH b, ACCOUNT\_TYPE a*

*WHERE c.customer\_id = t.transaction\_id AND*

*t.branch\_id = b.branch\_id AND*

*t.account\_id = a.account\_id;*

Description: The *All\_Customer\_Transaction* is granted UPDATE and SELECT privilege on the newly created Customer\_Transaction view.

*Grant SELECT, UPDATE on All\_Customer\_Transaction to senior\_management\_role;*

*Grant SELECT, UPDATE on All\_Customer\_Transaction to dba\_role;*

*--SENIOR MANAGEMENT WHO IS CCHRISTEN02 IS ABLE TO VIEW AND UPATE THE ALL\_CUSTOMER\_TRANSACTION VIEW--*

*connect cchristen02/Craven87\_878;*

*show user;*

*select \* from db668a02.All\_Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> connect cchristen02/Craven87\_878;*

*Connected.*

*SQL> show user;*

*USER is "CCHRISTEN02"*

*SQL> select \* from db668a02.All\_Customer\_Transaction;*

*CUSTOMER\_ID BRANCH\_NAME FNAME LNAME M STREET\_ADDRESS*

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

*CITY ST ZIP5 PHONE EMAIL\_ADDRESS USER\_NAME TRANSACTION\_NAME*

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

*TRANSACTION\_AMOUNT TRANSACTI TRANSACTI ACCOUNT\_TYPE\_NAME*

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

*9 ABC BRICK ROCK AXEL FOLEY H 777 MISSISSIPPI AVE*

*LILLY BECK DE 20887 444-333-2222 AFOLEY@HOTMAIL.COM AFOLEY02 CHECKING INQUIRY*

*300.00 15-JAN-12 15-JAN-12 REGULAR CHECKING*

*7 ABC ALUMINUM ROCK MICHAEL MARLON F 667 DAKOTA AVE*

*FALLS CHURCH ND 20885 555-665-4486 MMARLON@ABC.COM MMARLON02 SAVINGS INQUIRY*

*344.00 30-JAN-12 30-JAN-12 REGULAR CHECKING*

*11 ABC GOLD ROCK DANIEL MOSS B 155 CARNEKIE BOULEVARD*

*LITTLE MAC CA 20803 555-000-4410 DMOSS@EXCITE.COM DMOSS02 LINE OF CREDIT TRANSFER*

*400.00 10-FEB-12 10-FEB-12 REGULAR CHECKING*

*15 rows selected.*

*update db668a02.All\_Customer\_Transaction*

*set transaction\_amount ='109.00'*

*where customer\_id = 15;*

*select \* from db668a02.All\_Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> update db668a02.All\_Customer\_Transaction*

*2 set transaction\_amount ='109.00'*

*3 where customer\_id = 15*

*4 ;*

*1 row updated.*

*SQL> select \* from db668a02.All\_Customer\_Transaction;*

*CUSTOMER\_ID BRANCH\_NAME FNAME LNAME M STREET\_ADDRESS*

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

*CITY ST ZIP5 PHONE EMAIL\_ADDRESS USER\_NAME TRANSACTION\_NAME*

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

*TRANSACTION\_AMOUNT TRANSACTI TRANSACTI ACCOUNT\_TYPE\_NAME*

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

*15 ABC PLATNUM ROCK CARLOS SWEETEN P 200 GANTE AVE*

*WINK HEAD MD 27701 555-888-4489 SWEETEN-CARLOS@EXCITE.COM CSWEETEN02 LOAN PAYMENT*

*109.00 15-JAN-12 15-JAN-12 CREDIT\_CARD*

*--THE DBA WHO IS EWILMOTH02 IS ABLE TO VIEW AND UPATE THE ALL\_CUSTOMER\_TRANSACTION VIEW--*

*connect ewilmoth02/SWeltz16\_423;*

*show user;*

*select \* from db668a02.All\_Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> connect ewilmoth02/SWeltz16\_423;*

*Connected.*

*SQL> show user;*

*USER is "EWILMOTH02"*

*SQL> select \* from db668a02.All\_Customer\_Transaction;*

*CUSTOMER\_ID BRANCH\_NAME FNAME LNAME M STREET\_ADDRESS*

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

*CITY ST ZIP5 PHONE EMAIL\_ADDRESS USER\_NAME TRANSACTION\_NAME*

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

*TRANSACTION\_AMOUNT TRANSACTI TRANSACTI ACCOUNT\_TYPE\_NAME*

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

*9 ABC BRICK ROCK AXEL FOLEY H 777 MISSISSIPPI AVE*

*LILLY BECK DE 20887 444-333-2222 AFOLEY@HOTMAIL.COM AFOLEY02 CHECKING INQUIRY*

*300.00 15-JAN-12 15-JAN-12 REGULAR CHECKING*

*7 ABC ALUMINUM ROCK MICHAEL MARLON F 667 DAKOTA AVE*

*FALLS CHURCH ND 20885 555-665-4486 MMARLON@ABC.COM MMARLON02 SAVINGS INQUIRY*

*344.00 30-JAN-12 30-JAN-12 REGULAR CHECKING*

*11 ABC GOLD ROCK DANIEL MOSS B 155 CARNEKIE BOULEVARD*

*LITTLE MAC CA 20803 555-000-4410 DMOSS@EXCITE.COM DMOSS02 LINE OF CREDIT TRANSFER*

*400.00 10-FEB-12 10-FEB-12 REGULAR CHECKING*

*15 rows selected.*

*update db668a02.All\_Customer\_Transaction*

*set transaction\_amount ='220.00'*

*where customer\_id = 10;*

*select \* from db668a02.All\_Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> update db668a02.All\_Customer\_Transaction*

*2 set transaction\_amount ='220.00'*

*3 where customer\_id = 10;*

*1 row updated.*

*--OTHER USERS WHO ARE NOT MEMBERS OF THESE GROUPS ARE UNABLE TO ACCESS OR UPDATE THE ALL\_CUSTOMER\_TRANSACTION VIEW.--*

*connect marley02/Marmar2\_123;*

*show user;*

*select \* from db668a02.All\_Customer\_Transaction;*

*--SPOOL RESULTS--*

*SQL> connect marley02/Marmar2\_123;*

*Connected.*

*SQL> show user;*

*USER is "MARLEY02"*

*SQL> select \* from db668a02.All\_Customer\_Transaction;*

*select \* from db668a02.All\_Customer\_Transaction*

*\**

*ERROR at line 1:*

*ORA-00942: table or view does not exist*

*update db668a02.All\_Customer\_Transaction*

*set transaction\_amount ='420.00'*

*where customer\_id = 11;*

*--SPOOL RSULTS--*

*SQL> update db668a02.All\_Customer\_Transaction*

*2 set transaction\_amount ='420.00'*

*3 where customer\_id = 11;*

*update db668a02.All\_Customer\_Transaction*

*\**

*ERROR at line 1:*

*ORA-00942: table or view does not exist*

**Security Policy:** All users can view basic branch details e.g. branch name and location. Displaying the branch\_id is not necessary. Only the DBA can update the records in this table.

*connect db668a02/securityclass27;*

*show user;*

*CREATE OR REPLACE VIEW View\_Branch*

*As*

*Select branch\_name Branch, branch\_street Street, branch\_city City,*

*branch\_state State, branch\_zip5 Zip, phone\_tollfree "Toll Free", phone\_local Phone from branch;*

Description: The *all\_customer\_role* is granted SELECT privilege on the newly created *View\_Branch* view. In addition to the SELECT privilege, the dba\_role is granted UPDATE on the View\_Branch view.

*grant SELECT ON View\_Branch to All\_Customer\_Role;*

*grant UPDATE on View\_Branch to dba\_role;*

*--USERS CAN VIEW ALL THE RECORDS IN THE VIEW\_BRANCH VIEW BUT ARE UNABLE, EXCEPT FOR THE DBA, TO UPDATE THE TABLE--*

*connect marley02/Marmar2\_123;*

*show user;*

*select \* from db668a02.View\_Branch;*

*---SPOOL RESULTS--*

*SQL> connect marley02/Marmar2\_123;*

*Connected.*

*SQL> show user;*

*USER is "MARLEY02"*

*SQL> select \* from db668a02.View\_Branch;*

*BRANCH STREET CITY ST ZIP Toll Free PHONE*

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

*ABC SILVER ROCK 123 CANADA STREET GOLD ROD DC 22551 800-559-4449 555-12-5000*

*ABC GOLD ROCK 123 GINGER HILL STREET SILVER ROD MD 22552 800-558-4448 555-12-4000*

*ABC BRONZ ROCK 123 BUD ROCK STREET FRAME ROD NC 22553 800-557-4447 555-12-3000*

*ABC PLATNUM ROCK 123 DEDICATION ROAD MEADOW ROD FL 22554 800-556-4446 555-12-2000*

*ABC ALUMINUM ROCK 123 WYMIL STREET PEAR TREE CA 22555 800-555-4445 555-12-1000*

*ABC METAL ROCK 123 TUNA MELT STREET MILTON DC 22556 800-554-4444 555-12-9000*

*ABC STONE ROCK 123 SEPTEMBER AVE SILVER SPRING ND 22557 800-553-4442 555-12-8000*

*ABC ZINC ROCK 123 DUNKIN DRIVE GERMANTOWN MD 22558 800-552-4443 555-12-7000*

*ABC BRICK ROCK 123 PUMPERNICKEL BOULEVARD BISMARCK VA 22559 800-551-4441 555-12-6000*

*ABC DIAMOND ROCK 123 CALMING TERRACE GOLD ROD DE 22560 800-550-4440 555-12-3999*

*ABC WATER ROCK 123 WOODLAND DRIVE GOLD ROD PA 22261 800-555-4439 555-12-3998*

*ABC HEADQUARTERS 223 WOOD BECK WAY GOLDCLIFF MD 24260 800-555-4000 555-12-3000*

*12 rows selected.*

*update db668a02. View\_Branch*

*set branch ='NATION DELIBERATE'*

*where branch like '%SILVER%';*

*--SPOOL RESULTS*

*SQL> update db668a02. View\_Branch*

*2 set branch ='NATION DELIBERATE'*

*3 where branch like '%SILVER%';*

*update db668a02. View\_Branch*

*\**

*ERROR at line 1:*

*ORA-01031: insufficient privileges*

*--THE DBA IS ABLE TO SELECT AND UPDATE THE VIEW\_BRANCH VIEW--*

*connect ewilmoth02/SWeltz16\_423;*

*show user;*

*select \* from db668a02.View\_Branch;*

*SQL> connect ewilmoth02/SWeltz16\_423;*

*Connected.*

*SQL> show user;*

*USER is "EWILMOTH02"*

*SQL> select \* from db668a02.View\_Branch;*

*BRANCH STREET CITY ST ZIP Toll Free PHONE*

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

*ABC SILVER ROCK 123 CANADA STREET GOLD ROD DC 22551 800-559-4449 555-12-5000*

*ABC GOLD ROCK 123 GINGER HILL STREET SILVER ROD MD 22552 800-558-4448 555-12-4000*

*ABC BRONZ ROCK 123 BUD ROCK STREET FRAME ROD NC 22553 800-557-4447 555-12-3000*

*ABC PLATNUM ROCK 123 DEDICATION ROAD MEADOW ROD FL 22554 800-556-4446 555-12-2000*

*ABC ALUMINUM ROCK 123 WYMIL STREET PEAR TREE CA 22555 800-555-4445 555-12-1000*

*ABC METAL ROCK 123 TUNA MELT STREET MILTON DC 22556 800-554-4444 555-12-9000*

*ABC STONE ROCK 123 SEPTEMBER AVE SILVER SPRING ND 22557 800-553-4442 555-12-8000*

*ABC ZINC ROCK 123 DUNKIN DRIVE GERMANTOWN MD 22558 800-552-4443 555-12-7000*

*ABC BRICK ROCK 123 PUMPERNICKEL BOULEVARD BISMARCK VA 22559 800-551-4441 555-12-6000*

*ABC DIAMOND ROCK 123 CALMING TERRACE GOLD ROD DE 22560 800-550-4440 555-12-3999*

*ABC WATER ROCK 123 WOODLAND DRIVE GOLD ROD PA 22261 800-555-4439 555-12-3998*

*ABC HEADQUARTERS 223 WOOD BECK WAY GOLDCLIFF MD 24260 800-555-4000 555-12-3000*

*12 rows selected.*

*update db668a02. View\_Branch*

*set branch ='NATION DELIBERATE'*

*where branch like '%SILVER%';*

*select \* from db668a02. View\_Branch;*

*--SPOOL RESULTS--*

*SQL> update db668a02. View\_Branch*

*2 set branch ='NATION DELIBERATE'*

*3 where branch like '%SILVER%';*

*1 row updated.*

**Security Policy:** All customers including employees should be able to access only his or her own bank statement.

*connect db668a02/securityclass27;*

*show user;*

*CREATE OR REPLACE VIEW Own\_Bank\_Statement*

*As*

*Select c.customer\_id, b.statement\_id, c.fname, c.lname, c.middle\_initial,*

*c.street\_address, c.city, c.state, c.zip5, c.phone, c.email\_address,*

*b.statement\_number, b.statement\_date, t.transaction\_name, t.transaction\_type, t.transaction\_code, t.transaction\_amount, t.transaction\_begin\_date, t.transaction\_end\_date, h.balance\_amount*

*FROM CUSTOMER c, BANK\_STATEMENT b, TRANSACTION t, ACCOUNT\_HISTORY h*

*WHERE c.customer\_id =b.customer\_id AND*

*b.statement\_id = t.statement\_id AND*

*t.transaction\_id = h.transaction\_id AND*

*(c.user\_name =*

*SYS\_CONTEXT ('USERENV', 'SESSION\_USER') OR*

*user = 'DB668A02')*

*with check option;*

Description: The all\_customer\_role is granted SELECT privilege on the newly created Own\_Bank\_Statementview and the view is granted to all users.

*Grant SELECT on Own\_Bank\_Statement* *to All\_Customer\_Role;*

*--TESTING ACCESS TO OWN\_BANK\_STATEMENT--*

*connect wbrinkley02/Wink02\_778;*

*show user;*

*select \* from db668a02.Own\_Bank\_Statement;*

*--SPOOL RESULTS--*

*SQL> connect wbrinkley02/Wink02\_778;*

*show user;*

*select \* from db668a02.Own\_Bank\_Statement;*

*Connected.*

*SQL> USER is "WBRINKLEY02"*

*SQL>*

*CUSTOMER\_ID STATEMENT\_ID FNAME LNAME M STREET\_ADDRESS CITY ST ZIP5*

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

*PHONE EMAIL\_ADDRESS STATEMENT\_ STATEMENT TRANSACTION\_NAME TRANSACTION\_TYPE*

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

*TRANSACTIO TRANSACTION\_AMOUNT TRANSACTI TRANSACTI BALANCE\_AMOUNT*

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

*2 2 WILMER BRINKLEY A 555 WISCONSIN AVE BULL ROCK MD 20888*

*555-333-4481 WBRINKLEY@AOL.COM S0001 30-JAN-12 SAVINGS WITHDRAWAL WITHDRAWAL*

*W00001 300.00 10-JAN-12 10-JAN-12 4750*

*2 3 WILMER BRINKLEY A 555 WISCONSIN AVE BULL ROCK MD 20888*

*555-333-4481 WBRINKLEY@AOL.COM S0002 30-JAN-12 SAVINGS TRANSFER FUND TRANSFER*

*T00001 250.00 11-JAN-12 11-JAN-12 1050*

*2 2 WILMER BRINKLEY A 555 WISCONSIN AVE BULL ROCK MD 20888*

*555-333-4481 WBRINKLEY@AOL.COM S0001 30-JAN-12 SAVINGS DEPOSIT DEPOSIT*

*D00001 349.00 12-JAN-12 12-JAN-12 1350*

*8 rows selected.*

*--USER BRINKLEY02 IS UNABLE TO VIEW THE BANK STATMENT OF ANOTHER EMPLOYEE;*

*select \* from db668a02.Own\_Bank\_Statement*

*where customer\_id = ;*

**Security Policy**: The board of directors should only see records based on the security level established for each member.

Description: The security for the board\_director table will be established using Oracle Label Security.

### Oracle Label Security

Description: create view from the employee table so that employees can only view the records based on their security level see the name, title, customer\_id and department of all employees. This view will be used in the Oracle Label Seurity referenced below.

**Security Policy**: Employees should only be able to see the records in the All\_Employee view based on their security access.

*connect db668a02/securityclass27;*

*Grant Select on board\_director to director\_role;*

--USING THE LBACSYS ACCOUNT TO CREATE SECURITY POLICY--

*connect lbacsys/lbac333;*

--DROP POLICY IF IT IS ALREADY CREATED AND CREATE NEW POLICY--

*execute sa\_sysdba.drop\_policy ('Director', true);*

*execute sa\_sysdba.create\_policy ('Director','SecLev');*

--GRANTING THE APPROPRIATE PRIVILEGES TO MANAGE LABEL TO SECMAN02 USER--

*grant Director\_dba to secman02;*

*grant execute on sa\_components to secman02;*

*grant execute on sa\_label\_admin to secman02;*

*grant execute on sa\_user\_admin to secman02;*

*grant execute on char\_to\_label to secman02;*

*grant execute on sa\_policy\_admin to secman02;*

--ESTABLISHING SECURITY LEVELS AS SECMAN02--

*connect secman02/s;*

*show user;*

*execute sa\_components.create\_level ('Director',600,'A','Apprentice');*

*execute sa\_components.create\_level ('Director',700,'S','Sophomore');*

*execute sa\_components.create\_level ('Director',800,'J','Junior');*

*execute sa\_components.create\_level ('Director',900,'M','Master');*

--CREATING THE SECURITY LABELS--

*execute sa\_label\_admin.create\_label ('Director',38002,'M');*

*execute sa\_label\_admin.create\_label ('Director',48002,'J');*

*execute sa\_label\_admin.create\_label ('Director',58002,'S');*

*execute sa\_label\_admin.create\_label ('Director',68002,'A');*

--SETTING USER LEVELS--

*execute sa\_user\_admin.set\_levels ('Director','CRAIG','M','A','M','A');*

*execute sa\_user\_admin.set\_levels ('Director','MARGE','J','A','J','A');*

*execute sa\_user\_admin.set\_levels ('Director','MOODY','S','A','S','A');*

*execute sa\_user\_admin.set\_levels ('Director','MEL','A','A','A','A');*

--APPLYING THE DIRECTOR POLICY TO THE EMPLOYEE TABLE--

*connect db668a02/securityclass27;*

*drop sequence SEQ\_BOARDID;*

*drop table board\_director cascade constraints purge;*

*create table board\_director*

*(bmember\_id number(10),*

*bdfname varchar2(30),*

*bdlname varchar2(35),*

*bd\_phone varchar2(12),*

*director\_level varchar(12),*

*location varchar2(50),*

*constraint board\_pk primary key (bmember\_id));*

*Grant SELECT ON BOARD\_DIRECTOR TO SECMAN02;*

*Grant SELECT ON BOARD\_DIRECTOR TO DIRECTOR\_ROLE;*

*Grant DIRECTOR\_ROLE to craig, marge, moody, mel;*

*connect secman02/s;*

*execute sa\_policy\_admin.apply\_table\_policy ('Director','DB668A02','BOARD\_DIRECTOR','READ\_CONTROL');*

CREATE SEQUENCE seq\_boardid start with 1 increment by 1 minvalue 0 nomaxvalue nocache;

insert into db668a02.BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'CRAIG','DOUGLAS','555-550-4440','SENIOR','MID ATLANTIC', 38002);

insert into db668a02.BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MARGE','SIMMS','555-500-4550','SOPHOMORE','SOUTH EAST', 58002);

insert into db668a02.BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MOODY','FYRE','555-550-4440','JUNIOR','PACIFIC', 48002);

insert into db668a02.BOARD\_DIRECTOR values (SEQ\_BOARDID.NEXTVAL, 'MEL','DOULS','555-990-4441','APPRENTICE','CENTRAL',68002);

----CONNECT AND TEST LABEL AS USER CRAIG--

connect *craig/Craten\_999*;

show user;

select \* from db668a02.board\_director;

*--SPOOL RESULTS--*

*SQL> connect craig/Craten\_999;*

*Connected.*

*SQL>*

*SQL> show user;*

*USER is "CRAIG"*

*SQL>*

*SQL> select \* from db668a02.board\_director;*

*BMEMBER\_ID BDFNAME BDLNAME BD\_PHONE DIRECTOR\_LEV LOCATION*

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

*SECLEV*

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

*6 CRAIG DOUGLAS 555-550-4440 SENIOR MID ATLANTIC*

*38002*

*7 MARGE SIMMS 555-500-4550 SOPHOMORE SOUTH EAST*

*58002*

*8 MOODY FYRE 555-550-4440 JUNIOR PACIFIC*

*48002*

*9 MEL DOULS 555-990-4441 APPRENTICE CENTRAL*

*68002*

--CONNECT AND TEST LABEL AS USER MARGE--

*connect marge/Marvy\_333;*

*show user;*

*select \* from db668a02.board\_director;*

*--SPOOL RESULTS--*

*SQL> connect marge/Marvy\_333;*

*Connected.*

*SQL>*

*SQL> show user;*

*USER is "MARGE"*

*SQL>*

*SQL> select \* from db668a02.board\_director;*

*BMEMBER\_ID BDFNAME BDLNAME BD\_PHONE DIRECTOR\_LEV LOCATION*

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

*SECLEV*

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

*7 MARGE SIMMS 555-500-4550 SOPHOMORE SOUTH EAST*

*58002*

*8 MOODY FYRE 555-550-4440 JUNIOR PACIFIC*

*48002*

*9 MEL DOULS 555-990-4441 APPRENTICE CENTRAL*

*68002*

--CONNECT AND TEST LABEL AS USER MOODY--

connect *moody/Moofy\_999*;

show user;

select \* from db668a02.board\_director;

*--SPOOL RESULTS--*

*SQL> connect moody/Moofy\_999;*

*Connected.*

*SQL>*

*SQL> show user;*

*USER is "MOODY"*

*SQL>*

*SQL> select \* from db668a02.board\_director;*

*BMEMBER\_ID BDFNAME BDLNAME BD\_PHONE DIRECTOR\_LEV LOCATION*

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

*SECLEV*

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

*7 MARGE SIMMS 555-500-4550 SOPHOMORE SOUTH EAST*

*58002*

*9 MEL DOULS 555-990-4441 APPRENTICE CENTRAL*

*68002*

--CONNECT AND TEST LABEL AS USER MEL--

*connect mel/Mevie\_888;*

*show user;*

*select \* from db668a02.board\_director;*

*--SPOOL RESULTS--*

*SQL> connect mel/Mevie\_888;*

*Connected.*

*SQL> show user;*

*USER is "MEL"*

*SQL> select \* from db668a02.board\_director;*

*BMEMBER\_ID BDFNAME BDLNAME BD\_PHONE DIRECTOR\_LEV LOCATION*

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

*SECLEV*

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

*9 MEL DOULS 555-990-4441 APPRENTICE CENTRAL*

*68002*

# Conclusion

Within recent years, there has been significant advancement in technology and an exponential growth in database systems. As businesses grow their database repository the need for advance security increases. Effective database security management is a major component to having a successful, reliable, and secure database environment. In order to achieve this, a detailed database security plan is paramount. The security plan will include a comprehensive outline of the security needs of the business and will include a wide range of policies and procedures. The lessons learned from completing this project are significant. When establishing a security plan, the first step is to acquire a solid understanding of the business requirements and then determine what areas of the system will need to be secure. This process also involves understanding any current security plan and determining any enhancements that can be made to that plan. Identifying all security vulnerabilities in the system is crucial to developing and implementing a successful security plan. This process can only be effective if key stakeholders are actively involved since the initial phase of the project.

At the start of this project, the security needs of the business were researched and identified. General Banking reported using social security numbers as unique identifier in their production database. Other issues include data redundancy, lack of control and major privacy concerns. Though there are a number of techniques that can be used to increase database security, the techniques used here efficiently meet all the business requirements. The use of database views and Oracle Label Security, implemented using Oracle methods and procedures, were heavily utilized. Because this database is a banking system that consists of customer transactions, strict security of the system is critically important. The views were used to hide sensitive data and to restrict the data that is available to users. One example is the salary information stored for each employee. It is crucial that an employee cannot change his or her salary nor can any employee view the salary information of another employee. Each security policy defined in this project was carefully detailed and forcefully implemented based on the security needs of the fictitious business, General Banking.

# References

Elmasri R., & Navathe. S.B. ( 2011). *Fundamentals of Database Systems* (3ed). Boston, MA: Pearson Education

Goodrich, M.T, & Tamassia R. (2011*). Introduction to computer security*. Boston, MA: Pearson Education.

Knox, D. C. (2004). *Effective oracle database 10g security by design*. Emeryville, CA: McGraw-Hill Companies.

Needham, P. (2008). Oracle Label Security Best Practices. Retrieved October 15, 2012 from http://www.oracle.com/technetwork/database/security/twp-security-db-label-best-practice-134426.pdf

Rischert, A. (2009). *Oracle SQL By Example (4ed)*. Boston, MA: Pearson Education

Theriault, M. & Henry, W. (1998). Oracle Security. Retrieved November 1, 2012 from http://oreilly.com/catalog/orasec/chapter/ch07.html