**Sample Project Chapter 8 - Implementing Security Features for The Art Gallery Database**

We will work with the normalized purely relational database created at the end of chapter 6. The DDL statements to create the ddl are shown below and are stored in a text file called *Chapter8SQLCode* in this directory. You should use this file.

* Step 8.1 - Create a value-independent view that hides some private information.

The view will be of the Artist table, but without the social security number, sales, or interview information.

CREATE VIEW ArtistView1 AS

SELECT artistId, firstName, lastName, areaCode, telephoneNumber, street, zip, usualMedium, usualStyle, usualType

FROM Artist;

* Step 8.2 - Create a value-dependent view that shows the title, artist, medium, size, style of type of all artworks whose prices are over 10,000.

CREATE VIEW ExpensiveArt AS

SELECT WorkTitle, firstName, lastName, workMedium, workSize, workStyle, workType

FROM Artwork w, Artist a

WHERE w.artistid = a.artistid;

* Step 8.3 - Create a user and authorize that person to read the ExpensiveArt view. Begin an authorization graph. *Note: Depending on your rights, you may or may not be able to create and authorize users.*

CREATE USER U1 IDENTIFIED BY SESAME;

GRANT SELECT ON ExpensiveArt TO UI;

* Step 8.4 - Create four other users and Authorize then to access and/or modify various parts of the database, and update the authorization graph.

CREATE USER U2 IDENTIFIED BY SECRET2;

CREATE USER U3 IDENTIFIED BY SECRET3;

CREATE USER U4 IDENTIFIED BY SECRET4;

CREATE USER U5 IDENTIFIED BY SECRET5;

GRANT SELECT ON Collector TO U2;

GRANT ALL PRIVILEGES ON Collector TO U3;

GRANT SELECT, INSERT ON Sale TO U4 WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON Artwork TO U5;

The authorization graph is shown in Figure S.8.1

Figure S.8.1 - Authorization Graph for The Art Gallery

* Step 8.5 - Create a role, give it some privileges, and grant the role to a user.

CREATE ROLE MYUSERS;

GRANT ALL PRIVILEGES ON ARTWORK TO MYUSERS;

GRANT MYUSERS TO U1;

* Step 8.6 - Write a trigger for an audit trail for updates to a sensitive item that users can update.

This trigger will monitor changes to the asking price of an artwork.

--create the table for the audit trail

create table ArtworkPriceAudit(

dateofChange DATE,

changeuser varchar2(20),

IDofArtwork number(6),

oldPrice number(8,2),

newPrice number(8,2));

-- create the trigger

CREATE OR REPLACE TRIGGER ArtworkPriceAuditTrail

BEFORE UPDATE OF askingPrice ON Artwork

FOR EACH ROW

BEGIN

INSERT INTO ArtworkPriceAudit

VALUES(SYSDATE, USER, :OLD.artworkId,:OLD.askingPrice,

:NEW.askingPrice);

END;

/

-- test the trigger

commit;

update artwork set askingPrice = 100;

select askingPrice from artwork;

select \* from artworkPriceAudit;

rollback;

select askingPrice from artwork;