STUDENT PROJECTS: Implementing Security Features for Student Projects

Read the sample project steps (The Art Gallery) for this chapter and apply the same techniques to the student project that you are developing. Use the normalized purely relational database you created at the end of chapter 6.

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

```
CREATE OR REPLACE VIEW Donor_Public_View AS
SELECT donorId, name, affiliation, circleType
FROM Donor1;

CREATE OR REPLACE VIEW Payment_Public_View AS
SELECT paymentId, pledgeId, amountPaid, paymentType, paymentTiming
FROM Payment1;
```

- Step 8.2 Create a value-dependent view that screens some data.

```
CREATE OR REPLACE VIEW Payment_Amount AS
SELECT paymentId, pledgeId, amountPaid, paymentType
FROM Payment1
WHERE amountPaid >= 150;
```

- Step 8.3 Set up an audit trail for updates to a sensitive item that users can update and test it by updating the item.

```
CREATE TABLE Audit_Payment(
dateAndTimeOfUpdate DATE,
userId VARCHAR2(20),
oldPaymentId NUMBER,
 oldAmount NUMBER(10,2),
 newAmount NUMBER(10,2),
CONSTRAINT paymentAudit_pk PRIMARY KEY(dateAndTimeOfUpdate, userId)
);

CREATE OR REPLACE TRIGGER PaymentAuditTrail
BEFORE UPDATE OF amountPaid ON Payment1
FOR EACH ROW
BEGIN
INSERT INTO Audit_Payment
VALUES (SYSDATE, USER, :OLD.paymentId, :OLD.amountPaid, :NEW.amountPaid);
END;
/

UPDATE Payment1
SET amountPaid = 999
WHERE paymentId = 1;

SELECT * FROM Audit_Payment;
ROLLBACK;
```