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

CREATE VIEW Public\_Ward AS

SELECT

    Ward\_Number,

    Ward\_Name,

    Ward\_Location

FROM

    Ward;

SELECT \* FROM Public\_Ward;

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

CREATE VIEW Public\_Patient AS

SELECT

Patient\_Number,

Patient\_Name,

Address

FROM

Patient;

SELECT \* FROM Public\_Patient;

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

CREATE VIEW Public\_Medication AS

SELECT

    Medication\_Name,

    Patient\_Number,

    Drug\_Number

FROM

    Medication;

SELECT \* FROM Public\_Medication;

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

CREATE VIEW Public\_Staff AS

SELECT

Staff\_Number,

Staff\_Name,

Qualification

FROM

Staff;

SELECT \* FROM Public\_Staff;

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

CREATE VIEW Public\_Supply AS

SELECT

    Supply\_Number,

    Supply\_Name,

    Quantity\_In\_Stock

FROM

    Supply;

SELECT \* FROM Public\_Supply;

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

CREATE VIEW Public\_Requisition AS

SELECT

Requisition\_Number,

Quantity\_Required,

Date\_Of\_Order,

Delivery\_Date

FROM

Requisition;

SELECT \* FROM Public\_Requisition;

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

CREATE VIEW Public\_Supplier AS

SELECT

Supplier\_Number,

Supplier\_Name,

Address

FROM

Supplier;

SELECT \* FROM Public\_Supplier;

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

CREATE VIEW Patient\_Age AS

SELECT Patient\_Number, Patient\_Name, Age, Address, Contact\_Number, Ward\_Number

FROM Patient

WHERE Age > 60;

SELECT \* FROM Patient\_Age;

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

CREATE VIEW Staff\_Past\_Experience AS

SELECT Staff\_Number, Staff\_Name, Qualification, Past\_Experience, Ward\_Number

FROM Staff

WHERE Staff\_Name = "Isabella Turner" AND Ward\_Number = 12;

SELECT \* FROM Staff\_Past\_Experience;

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

CREATE VIEW Supplier\_FullName AS

SELECT Supplier\_Number, Supplier\_Name, Address, Email, Telephone\_Number, Fax\_Number

FROM Supplier

WHERE Supplier\_Name LIKE 'a%';

SELECT \* FROM Supplier\_FullName;

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 IF NOT EXISTS AuditTrail (

AuditTrail\_ID INT AUTO\_INCREMENT PRIMARY KEY,

TableName VARCHAR(100) NOT NULL,

Action VARCHAR(10) NOT NULL,

Old\_Values JSON,

New\_Values JSON,

Update\_DateTime TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB;

DELIMITER //

CREATE TRIGGER trg\_Patient\_AuditTrail\_Update

AFTER UPDATE ON Patient FOR EACH ROW

BEGIN

INSERT INTO AuditTrail (TableName, Action, Old\_Values, New\_Values)

VALUES ('Patient', 'UPDATE', JSON\_OBJECT(

'Patient\_Number', OLD.Patient\_Number,

'Patient\_Name', OLD.Patient\_Name,

'Age', OLD.Age,

'Address', OLD.Address,

'Contact\_Number', OLD.Contact\_Number,

'Ward\_Number', OLD.Ward\_Number

), JSON\_OBJECT(

'Patient\_Number', NEW.Patient\_Number,

'Patient\_Name', NEW.Patient\_Name,

'Age', NEW.Age,

'Address', NEW.Address,

'Contact\_Number', NEW.Contact\_Number,

'Ward\_Number', NEW.Ward\_Number

));

END;

//

DELIMITER;

UPDATE Patient SET Age = 36 WHERE Patient\_Number = 1;

SELECT \* FROM AuditTrail;