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

-- Sprint 1: ERD & Schema Design for Healthcare Diagnostics

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

-- 1. Branch Table

CREATE TABLE Branch (

branch\_id NUMBER PRIMARY KEY,

branch\_name VARCHAR2(100) NOT NULL,

location VARCHAR2(150),

contact\_no VARCHAR2(15)

);

-- 2. Department Table

CREATE TABLE Department (

dept\_id NUMBER PRIMARY KEY,

dept\_name VARCHAR2(100) NOT NULL,

branch\_id NUMBER REFERENCES Branch(branch\_id)

);

-- 3. Doctor Table

CREATE TABLE Doctor (

doctor\_id NUMBER PRIMARY KEY,

doctor\_name VARCHAR2(100) NOT NULL,

specialization VARCHAR2(100),

dept\_id NUMBER REFERENCES Department(dept\_id)

);

-- 4. Patient Table

CREATE TABLE Patient (

patient\_id NUMBER PRIMARY KEY,

patient\_name VARCHAR2(100) NOT NULL,

dob DATE,

gender VARCHAR2(10),

contact\_no VARCHAR2(15),

email VARCHAR2(100)

);

-- 5. Visit Table

CREATE TABLE Visit (

visit\_id NUMBER PRIMARY KEY,

patient\_id NUMBER REFERENCES Patient(patient\_id),

doctor\_id NUMBER REFERENCES Doctor(doctor\_id),

visit\_date DATE DEFAULT SYSDATE,

reason VARCHAR2(255)

);

-- 6. Test Table

CREATE TABLE Test (

test\_id NUMBER PRIMARY KEY,

test\_name VARCHAR2(100) NOT NULL,

price NUMBER(8,2)

);

-- 7. Test Package Table

CREATE TABLE Test\_Package (

package\_id NUMBER PRIMARY KEY,

package\_name VARCHAR2(100),

total\_price NUMBER(8,2)

);

-- 8. Package\_Test Junction Table

CREATE TABLE Package\_Test (

package\_id NUMBER REFERENCES Test\_Package(package\_id),

test\_id NUMBER REFERENCES Test(test\_id),

PRIMARY KEY (package\_id, test\_id)

);

-- 9. Test\_Report Table

CREATE TABLE Test\_Report (

report\_id NUMBER PRIMARY KEY,

visit\_id NUMBER REFERENCES Visit(visit\_id),

test\_id NUMBER REFERENCES Test(test\_id),

result VARCHAR2(100),

report\_date DATE DEFAULT SYSDATE

);

-- 10. Technician Table

CREATE TABLE Technician (

technician\_id NUMBER PRIMARY KEY,

technician\_name VARCHAR2(100),

branch\_id NUMBER REFERENCES Branch(branch\_id)

);

-- 11. Inventory Table

CREATE TABLE Inventory (

item\_id NUMBER PRIMARY KEY,

item\_name VARCHAR2(100),

quantity NUMBER,

last\_updated DATE DEFAULT SYSDATE,

branch\_id NUMBER REFERENCES Branch(branch\_id)

);

-- 12. Bill Table

CREATE TABLE Bill (

bill\_id NUMBER PRIMARY KEY,

visit\_id NUMBER REFERENCES Visit(visit\_id),

total\_amount NUMBER(10,2),

discount NUMBER(5,2),

net\_amount NUMBER(10,2),

billing\_date DATE DEFAULT SYSDATE

);

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

-- Sprint 2: Sample PL/SQL Packages and Procedures

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

-- 1. Procedure to Register New Patient

CREATE OR REPLACE PROCEDURE add\_patient(

p\_id NUMBER,

p\_name VARCHAR2,

p\_dob DATE,

p\_gender VARCHAR2,

p\_contact VARCHAR2,

p\_email VARCHAR2

) AS

BEGIN

INSERT INTO Patient(patient\_id, patient\_name, dob, gender, contact\_no, email)

VALUES(p\_id, p\_name, p\_dob, p\_gender, p\_contact, p\_email);

END;

/

-- 2. Function to calculate net bill

CREATE OR REPLACE FUNCTION calculate\_net\_amount(

total NUMBER,

discount NUMBER

) RETURN NUMBER IS

BEGIN

RETURN total - (total \* discount / 100);

END;

/

-- 3. Procedure to Generate Bill

CREATE OR REPLACE PROCEDURE generate\_bill(

p\_bill\_id NUMBER,

p\_visit\_id NUMBER,

p\_total NUMBER,

p\_discount NUMBER

) AS

v\_net NUMBER;

BEGIN

v\_net := calculate\_net\_amount(p\_total, p\_discount);

INSERT INTO Bill(bill\_id, visit\_id, total\_amount, discount, net\_amount)

VALUES(p\_bill\_id, p\_visit\_id, p\_total, p\_discount, v\_net);

END;

/

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

-- Sprint 3: Reports & Analytics

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

-- 1. List of patients with visit count

SELECT p.patient\_name, COUNT(v.visit\_id) AS visit\_count

FROM Patient p

JOIN Visit v ON p.patient\_id = v.patient\_id

GROUP BY p.patient\_name;

-- 2. Total Revenue by Branch

SELECT b.branch\_name, SUM(bill.net\_amount) AS total\_revenue

FROM Branch b

JOIN Department d ON b.branch\_id = d.branch\_id

JOIN Doctor doc ON d.dept\_id = doc.dept\_id

JOIN Visit v ON doc.doctor\_id = v.doctor\_id

JOIN Bill bill ON v.visit\_id = bill.visit\_id

GROUP BY b.branch\_name;

-- 3. Procedure: Inventory Alert

CREATE OR REPLACE PROCEDURE inventory\_alert IS

BEGIN

FOR item IN (SELECT item\_name, quantity FROM Inventory WHERE quantity < 10)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Low Stock: ' || item.item\_name || ' - Qty: ' || item.quantity);

END LOOP;

END;

/