-- Patients table

CREATE TABLE patients (

patient\_id INT PRIMARY KEY,

name VARCHAR(100),

gender VARCHAR(10),

dob DATE

);

-- Doctors table

CREATE TABLE doctors (

doctor\_id SERIAL PRIMARY KEY,

name VARCHAR(100),

specialty VARCHAR(50)

);

-- Appointments table

CREATE TABLE appointments (

appointment\_id SERIAL PRIMARY KEY,

patient\_id INT REFERENCES patients(patient\_id),

doctor\_id INT REFERENCES doctors(doctor\_id),

appointment\_date DATE,

status VARCHAR(20)

);

-- Billing table

CREATE TABLE billing (

bill\_id SERIAL PRIMARY KEY,

patient\_id INT REFERENCES patients(patient\_id),

amount NUMERIC(10, 2),

bill\_date DATE DEFAULT CURRENT\_DATE

);

-- Audit table (for logging appointment inserts)

CREATE TABLE appointment\_audit (

log\_id SERIAL PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

log\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Sequence for patient\_id auto-assignment

CREATE SEQUENCE seq\_patient\_id START 1;

-- Function to assign patient ID automatically from sequence

CREATE OR REPLACE FUNCTION assign\_patient\_id()

RETURNS TRIGGER AS $$

BEGIN

NEW.patient\_id := nextval('seq\_patient\_id');

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Trigger to auto-fill patient\_id before insert

CREATE TRIGGER trg\_before\_insert\_patient

BEFORE INSERT ON patients

FOR EACH ROW

EXECUTE FUNCTION assign\_patient\_id();

CREATE OR REPLACE FUNCTION get\_total\_bill(p\_id INT)

RETURNS NUMERIC AS $$

DECLARE

total NUMERIC;

BEGIN

SELECT COALESCE(SUM(amount), 0) INTO total

FROM billing

WHERE patient\_id = p\_id;

RETURN total;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE PROCEDURE add\_patient(

p\_name TEXT,

p\_gender TEXT,

p\_dob DATE

)

LANGUAGE plpgsql

AS $$

BEGIN

INSERT INTO patients(name, gender, dob)

VALUES (p\_name, p\_gender, p\_dob);

END;

$$;

-- Function to log appointment inserts

CREATE OR REPLACE FUNCTION log\_appointment()

RETURNS TRIGGER AS $$

BEGIN

INSERT INTO appointment\_audit(patient\_id, doctor\_id)

VALUES (NEW.patient\_id, NEW.doctor\_id);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Trigger for appointment logging

CREATE TRIGGER trg\_log\_appointment

AFTER INSERT ON appointments

FOR EACH ROW

EXECUTE FUNCTION log\_appointment();

-- Insert doctors

INSERT INTO doctors(name, specialty) VALUES

('Dr. Mulu Tesfaye', 'Cardiology'),

('Dr. Eyerusalem Lemma', 'Pediatrics');

-- Insert patients using procedure

CALL add\_patient('Abel Mulu', 'Male', '2000-10-05');

CALL add\_patient('Sara Abate', 'Female', '1985-03-15');

-- Insert appointments

INSERT INTO appointments(patient\_id, doctor\_id, appointment\_date, status)

VALUES (1, 1, CURRENT\_DATE, 'Scheduled'),

(2, 2, CURRENT\_DATE, 'Confirmed');

-- Insert billing records

INSERT INTO billing(patient\_id, amount) VALUES

(1, 350.00),

(1, 150.00),

(2, 500.00);