-- Patients Table

CREATE TABLE Patients (

patient\_id INT PRIMARY KEY,

patient\_name VARCHAR(100) NOT NULL,

gender VARCHAR(10),

date\_of\_birth DATE,

contact\_number1 VARCHAR(20),

contact\_number2 VARCHAR(20),

address\_line1 VARCHAR(255),

 address\_line2 VARCHAR(255));

INSERT INTO Patients (patient\_id, patient\_name, gender, date\_of\_birth, contact\_number1, contact\_number2, address\_line1, address\_line2)

VALUES

(1, 'Amit Patel', 'Male', '1985-03-15', '+91-9876543210', '+91-8765432109', '123, Shyamal Avenue', 'Satellite, Ahmedabad'),

(2, 'Riya Shah', 'Female', '1992-07-21', '+91-7890123456', '+91-8901234567', '456, Mansi Circle', 'Vastrapur, Ahmedabad'),

(3, 'Suresh Desai', 'Male', '1980-05-08', '+91-8765432101', '+91-7654321098', '789, Ashram Road', 'Ellis Bridge, Ahmedabad'),

(4, 'Jaya Patel', 'Female', '1995-11-30', '+91-7654321090', '+91-6543210987', '234, CG Road', 'Navrangpura, Ahmedabad'),

(5, 'Vivek Shah', 'Male', '1988-09-17', '+91-9876543211', '+91-8765432100', '567, Drive-In Road', 'Thaltej, Ahmedabad'),

(6, 'Neha Patel', 'Female', '1987-12-03', '+91-8765432102', '+91-7654321091', '890, Science City Road', 'Sola, Ahmedabad'),

(7, 'Rajesh Mehta', 'Male', '1983-04-28', '+91-9876543212', '+91-8765432103', '111, Riverfront Road', 'Sabarmati, Ahmedabad'),

(8, 'Aishwarya Shah', 'Female', '1990-02-14', '+91-9876543213', '+91-8765432104', '222, Prahlad Nagar', 'Satellite, Ahmedabad'),

(9, 'Dinesh Patel', 'Male', '1979-06-19', '+91-9876543214', '+91-8765432105', '333, Gurukul Road', 'Memnagar, Ahmedabad'),

(10, 'Pooja Shah', 'Female', '1984-10-25', '+91-9876543215', '+91-8765432106', '444, Ambawadi', 'Paldi, Ahmedabad'),

(11, 'Ankit Patel', 'Male', '1982-08-12', '+91-9876543216', '+91-8765432107', '555, Drive-In Road', 'Thaltej, Ahmedabad'),

(12, 'Rashmi Shah', 'Female', '1993-09-08', '+91-9876543217', '+91-8765432108', '666, Science City Road', 'Sola, Ahmedabad'),

(13, 'Kunal Mehta', 'Male', '1981-07-05', '+91-9876543218', '+91-8765432109', '777, CG Road', 'Navrangpura, Ahmedabad'),

(14, 'Smita Patel', 'Female', '1986-06-02', '+91-9876543219', '+91-8765432110', '888, Prahlad Nagar', 'Satellite, Ahmedabad'),

(15, 'Rajat Shah', 'Male', '1989-11-11', '+91-9876543220', '+91-8765432111', '999, Gurukul Road', 'Memnagar, Ahmedabad'),

(16, 'Khushi Shah', 'Female', '1991-12-18', '+91-9876543221', '+91-8765432112', '101, Ambawadi', 'Paldi, Ahmedabad'),

(17, 'Deepak Patel', 'Male', '1987-03-23', '+91-9876543222', '+91-8765432113', '202, Riverfront Road', 'Sabarmati, Ahmedabad'),

(18, 'Ritika Shah', 'Female', '1984-04-19', '+91-9876543223', '+91-8765432114', '303, Mansi Circle', 'Vastrapur, Ahmedabad'),

(19, 'Harish Patel', 'Male', '1980-01-09', '+91-9876543224', '+91-8765432115', '404, Shyamal Avenue', 'Satellite, Ahmedabad'),

(20, 'Mona Shah', 'Female', '1988-02-27', '+91-9876543225', '+91-8765432116', '505, Drive-In Road', 'Thaltej, Ahmedabad');

-- Doctors Table

CREATE TABLE Doctors (

doctor\_id INT PRIMARY KEY,

doctor\_name VARCHAR(100) NOT NULL,

specialization VARCHAR(100),

contact\_number VARCHAR(20),

email VARCHAR(100),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);



INSERT INTO Doctors (doctor\_id, doctor\_name, specialization, contact\_number, email, department\_id)

VALUES

(1, 'Dr. Anil Kumar', 'Cardiologist', '+91-9876543210', 'anil.kumar@example.com', 1),

(2, 'Dr. Neha Sharma', 'Dermatologist', '+91-9876543211', 'neha.sharma@example.com', 2),

(3, 'Dr. Rajesh Singh', 'Orthopedic Surgeon', '+91-9876543212', 'rajesh.singh@example.com', 1),

(4, 'Dr. Pooja Verma', 'Gynecologist', '+91-9876543213', 'pooja.verma@example.com', 3),

(5, 'Dr. Vikram Gupta', 'Neurologist', '+91-9876543214', 'vikram.gupta@example.com', 2),

(6, 'Dr. Smita Patel', 'Oncologist', '+91-9876543215', 'smita.patel@example.com', 3),

(7, 'Dr. Rohit Malhotra', 'Pediatrician', '+91-9876543216', 'rohit.malhotra@example.com', 2),

(8, 'Dr. Nisha Singh', 'Psychiatrist', '+91-9876543217', 'nisha.singh@example.com', 1),

(9, 'Dr. Sunil Kapoor', 'Urologist', '+91-9876543218', 'sunil.kapoor@example.com', 3),

(10, 'Dr. Anjali Sharma', 'Ophthalmologist', '+91-9876543219', 'anjali.sharma@example.com', 1),

(11, 'Dr. Arjun Khanna', 'Dentist', '+91-9876543220', 'arjun.khanna@example.com', 2),

(12, 'Dr. Ritu Gupta', 'ENT Specialist', '+91-9876543221', 'ritu.gupta@example.com', 3),

(13, 'Dr. Manoj Saxena', 'Gastroenterologist', '+91-9876543222', 'manoj.saxena@example.com', 1),

(14, 'Dr. Meera Verma', 'Dietitian', '+91-9876543223', 'meera.verma@example.com', 2),

(15, 'Dr. Sameer Kapoor', 'Dermatologist', '+91-9876543224', 'sameer.kapoor@example.com', 3),

(16, 'Dr. Kavita Sharma', 'Psychologist', '+91-9876543225', 'kavita.sharma@example.com', 1),

(17, 'Dr. Rajat Mehta', 'Cardiologist', '+91-9876543226', 'rajat.mehta@example.com', 2),

(18, 'Dr. Nandini Patel', 'Endocrinologist', '+91-9876543227', 'nandini.patel@example.com', 3),

(19, 'Dr. Siddharth Malhotra', 'Oncologist', '+91-9876543228', 'siddharth.malhotra@example.com', 1),

(20, 'Dr. Alisha Singh', 'Neurologist', '+91-9876543229', 'alisha.singh@example.com', 2);

-- Appointments Table

CREATE TABLE Appointments (

appointment\_id SERIAL PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

appointment\_date DATE,

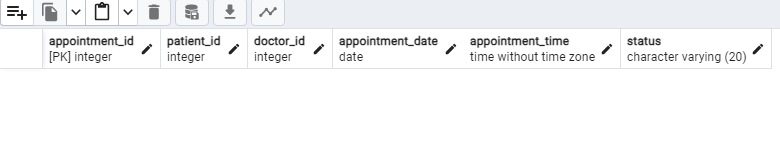
appointment\_time TIME,

status VARCHAR(20),

FOREIGN KEY (patient\_id) REFERENCES Patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctors(doctor\_id)

);



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

VALUES

(1, 1, '2023-10-10', '10:00:00', 'Scheduled'),

(2, 3, '2023-10-11', '14:30:00', 'Scheduled'),

(3, 5, '2023-10-12', '11:15:00', 'Scheduled'),

(4, 2, '2023-10-13', '09:45:00', 'Completed'),

(5, 7, '2023-10-14', '16:20:00', 'Cancelled'),

(6, 4, '2023-10-15', '13:00:00', 'Scheduled'),

(7, 9, '2023-10-16', '10:45:00', 'Scheduled'),

(8, 6, '2023-10-17', '15:30:00', 'Scheduled'),

(9, 8, '2023-10-18', '14:00:00', 'Completed'),

(10, 10, '2023-10-19', '12:30:00', 'Cancelled'),

(11, 12, '2023-10-20', '17:15:00', 'Scheduled'),

(12, 14, '2023-10-21', '11:45:00', 'Scheduled'),

(13, 11, '2023-10-22', '14:10:00', 'Scheduled'),

(14, 13, '2023-10-23', '10:25:00', 'Completed'),

(15, 15, '2023-10-24', '16:40:00', 'Cancelled'),

(16, 17, '2023-10-25', '13:55:00', 'Scheduled'),

(17, 16, '2023-10-26', '15:20:00', 'Scheduled'),

(18, 18, '2023-10-27', '09:30:00', 'Completed'),

(19, 20, '2023-10-28', '18:05:00', 'Scheduled'),

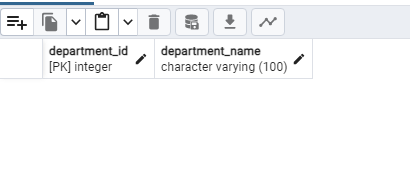
(20, 19, '2023-10-29', '12:15:00', 'Scheduled');

-- Departments Table

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100) NOT NULL

);

INSERT INTO Departments (department\_id, department\_name) VALUES

(1, 'Cardiology'),

(2, 'Orthopedics'),

(3, 'Neurology'),

(4, 'Oncology'),

(5, 'Pediatrics'),

(6, 'Gynecology'),

(7, 'Dermatology'),

(8, 'Ophthalmology'),

(9, 'ENT'),

(10, 'Urology'),

(11, 'Psychiatry'),

(12, 'Radiology'),

(13, 'Internal Medicine'),

(14, 'Emergency Medicine'),

(15, 'Anesthesiology'),

(16, 'Pathology'),

(17, 'Physical Therapy'),

(18, 'Occupational Therapy'),

(19, 'Nephrology'),

(20, 'Endocrinology');

-- Medical Records Table

CREATE TABLE MedicalRecords (

record\_id INT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

admission\_date DATE,

discharge\_date DATE,

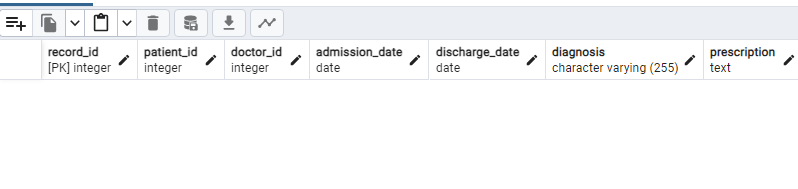
diagnosis VARCHAR(255),

prescription TEXT,

FOREIGN KEY (patient\_id) REFERENCES Patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctors(doctor\_id)

);



INSERT INTO MedicalRecords (record\_id, patient\_id, doctor\_id, admission\_date, discharge\_date, diagnosis, prescription)

VALUES

(1, 1, 1, '2023-01-05', '2023-01-10', 'Common Cold', 'Rest and fluids.'),

(2, 2, 3, '2023-02-12', '2023-02-20', 'Influenza', 'Prescribed antiviral medication.'),

(3, 4, 2, '2023-03-18', '2023-03-25', 'Allergic Reaction', 'Avoid allergen. Prescribed allergy medication.'),

(4, 5, 5, '2023-04-02', '2023-04-09', 'Sprained Ankle', 'Rest, ice, compression, elevation (RICE).'),

(5, 3, 4, '2023-05-15', '2023-05-22', 'Gastric Problems', 'Dietary changes and antacids.'),

(6, 6, 6, '2023-06-08', '2023-06-15', 'Migraine', 'Prescribed pain relievers. Avoid triggers.'),

(7, 7, 7, '2023-07-03', '2023-07-10', 'Hypertension', 'Prescribed blood pressure medication. Dietary changes.'),

(8, 8, 8, '2023-08-20', '2023-08-28', 'Diabetes Management', 'Insulin therapy and dietary control.'),

(9, 9, 9, '2023-09-12', '2023-09-20', 'Chronic Back Pain', 'Physical therapy and pain relievers.'),

(10, 10, 10, '2023-10-01', '2023-10-07', 'Sinusitis', 'Nasal irrigation and prescribed antibiotics.'),

(11, 11, 11, '2023-11-18', '2023-11-25', 'Urinary Tract Infection', 'Prescribed antibiotics and increased fluid intake.'),

(12, 12, 12, '2023-12-05', '2023-12-12', 'Anxiety Disorder', 'Therapy sessions and prescribed anti-anxiety medication.'),

(13, 13, 13, '2024-01-08', '2024-01-15', 'Dermatitis', 'Topical creams and avoiding irritants.'),

(14, 14, 14, '2024-02-22', '2024-03-01', 'Acute Bronchitis', 'Rest, fluids, and prescribed cough syrup.'),

(15, 15, 15, '2024-03-15', '2024-03-22', 'Gastroenteritis', 'Oral rehydration therapy and bland diet.'),

(16, 16, 16, '2024-04-02', '2024-04-09', 'Seasonal Allergies', 'Antihistamines and avoiding allergens.'),

(17, 17, 17, '2024-05-18', '2024-05-25', 'Conjunctivitis', 'Prescribed eye drops and keeping eyes clean.'),

(18, 18, 18, '2024-06-08', '2024-06-15', 'Tennis Elbow', 'Rest, physical therapy, and pain relievers.'),

(19, 19, 19, '2024-07-03', '2024-07-10', 'Stress-Related Headache', 'Relaxation techniques and pain relievers.'),

(20, 20, 20, '2024-08-20', '2024-08-28', 'Insomnia', 'Sleep hygiene practices and relaxation exercises.');

-- Nurses Table

CREATE TABLE Nurses (

nurse\_id INT PRIMARY KEY,

nurse\_name VARCHAR(100) NOT NULL,

contact\_number VARCHAR(20),

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);



INSERT INTO Nurses (nurse\_id, nurse\_name, contact\_number, department\_id)

VALUES

(1, 'Nurse 1', '+91-9876543210', 1),

(2, 'Nurse 2', '+91-9876543211', 2),

(3, 'Nurse 3', '+91-9876543212', 3),

(4, 'Nurse 4', '+91-9876543213', 4),

(5, 'Nurse 5', '+91-9876543214', 5),

(6, 'Nurse 6', '+91-9876543215', 6),

(7, 'Nurse 7', '+91-9876543216', 7),

(8, 'Nurse 8', '+91-9876543217', 8),

(9, 'Nurse 9', '+91-9876543218', 9),

(10, 'Nurse 10', '+91-9876543219', 10),

(11, 'Nurse 11', '+91-9876543220', 1),

(12, 'Nurse 12', '+91-9876543221', 2),

(13, 'Nurse 13', '+91-9876543222', 3),

(14, 'Nurse 14', '+91-9876543223', 4),

(15, 'Nurse 15', '+91-9876543224', 5),

(16, 'Nurse 16', '+91-9876543225', 6),

(17, 'Nurse 17', '+91-9876543226', 7),

(18, 'Nurse 18', '+91-9876543227', 8),

(19, 'Nurse 19', '+91-9876543228', 9),

(20, 'Nurse 20', '+91-9876543229', 10);

-- Beds Table

CREATE TABLE Beds (

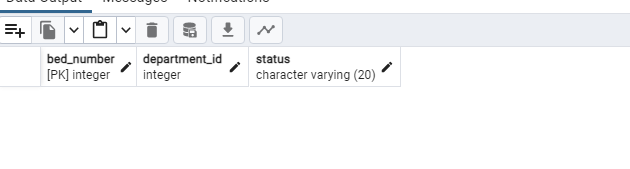
bed\_number INT PRIMARY KEY,

department\_id INT,

status VARCHAR(20),

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);



INSERT INTO Beds (bed\_number, department\_id, status)

VALUES

(1, 1, 'Occupied'),

(2, 1, 'Available'),

(3, 2, 'Occupied'),

(4, 2, 'Available'),

(5, 3, 'Occupied'),

(6, 3, 'Available'),

(7, 4, 'Occupied'),

(8, 4, 'Available'),

(9, 5, 'Occupied'),

(10, 5, 'Available'),

(11, 6, 'Occupied'),

(12, 6, 'Available'),

(13, 7, 'Occupied'),

(14, 7, 'Available'),

(15, 8, 'Occupied'),

(16, 8, 'Available'),

(17, 9, 'Occupied'),

(18, 9, 'Available'),

(19, 10, 'Occupied'),

(20, 10, 'Available');

------------------------Query------------------------

--1. Find patients who have had multiple appointments in the same day

SELECT p.patient\_name, a.appointment\_date, COUNT(a.appointment\_id) AS num\_appointments

FROM Patients p

INNER JOIN Appointments a ON p.patient\_id = a.patient\_id

GROUP BY p.patient\_name, a.appointment\_date

HAVING COUNT(a.appointment\_id) > 1;

--2. List doctors and their number of appointments, ordered by the number of appointments

SELECT d.doctor\_name, COUNT(a.appointment\_id) AS num\_appointments

FROM Doctors d

LEFT JOIN Appointments a ON d.doctor\_id = a.doctor\_id

GROUP BY d.doctor\_name

ORDER BY num\_appointments DESC;

--3. Retrieve all departments along with the number of doctors in each department

SELECT d.department\_name, COUNT(doc.doctor\_id) AS num\_doctors

FROM Departments d

LEFT JOIN Doctors doc ON d.department\_id = doc.department\_id

GROUP BY d.department\_name;

--4. List patients with medical records diagnosed with 'Fever'

SELECT p.patient\_name, m.diagnosis

FROM Patients p

INNER JOIN MedicalRecords m ON p.patient\_id = m.patient\_id

WHERE m.diagnosis = 'Fever';

--5. retrieves all appointments for a specific patient ID

CREATE OR REPLACE FUNCTION GetPatientAppointments(patientId INT) RETURNS TABLE (

appointment\_id INT,

appointment\_date DATE,

appointment\_time TIME,

doctor\_name VARCHAR(100),

status VARCHAR(20)

) AS $$

BEGIN

RETURN QUERY

SELECT a.appointment\_id, a.appointment\_date, a.appointment\_time, d.doctor\_name, a.status

FROM Appointments a

INNER JOIN Doctors d ON a.doctor\_id = d.doctor\_id

WHERE a.patient\_id = patientId;

END;

$$ LANGUAGE plpgsql;

-- Call the stored procedure for patient with ID 1

SELECT \* FROM GetPatientAppointments(1);

--6. Crate a function that retrieves the age of the pateint by the given date of birth

CREATE OR REPLACE FUNCTION CalculatePatientAge() RETURNS TABLE (

patient\_id INT,

patient\_name VARCHAR(100),

age INT

) AS $$

DECLARE

patient\_record Patients%ROWTYPE;

BEGIN

-- Cursor to fetch patient records

FOR patient\_record IN (SELECT \* FROM Patients)

LOOP

-- Calculate age using date\_of\_birth and current date

age := EXTRACT(YEAR FROM age(current\_date, patient\_record.date\_of\_birth));

-- Return patient\_id, patient\_name, and age

RETURN QUERY SELECT patient\_record.patient\_id, patient\_record.patient\_name, age;

END LOOP;

END;

$$ LANGUAGE plpgsql;

SELECT \* FROM CalculatePatientAge();

--7. Write a PL/pgSQL function named CalculateAveragePatientAge that calculates the

--average age of patients based on their date\_of\_birth in the Patients table.

CREATE OR REPLACE FUNCTION CalculateAveragePatientAge() RETURNS NUMERIC AS $$

DECLARE

total\_age NUMERIC := 0;

total\_patients INT := 0;

average\_age NUMERIC;

patient\_dob DATE;

BEGIN

-- Cursor to fetch date\_of\_birth values

FOR patient\_dob IN (SELECT date\_of\_birth FROM Patients)

LOOP

total\_age := total\_age + EXTRACT(YEAR FROM age(current\_date, patient\_dob));

total\_patients := total\_patients + 1;

END LOOP;

-- Calculate average age

IF total\_patients > 0 THEN

average\_age := total\_age / total\_patients;

ELSE

average\_age := 0;

END IF;

RETURN average\_age;

END;

$$ LANGUAGE plpgsql;

SELECT \* FROM CalculateAveragePatientAge();

--8. Create a cursor to fetch patient information for specific doctors' appointments

DO $$

DECLARE

doctor\_id\_to\_query INT := 1; -- Specify the doctor's ID for whom you want to fetch patients' information

patient\_record Patients%ROWTYPE;

appointment\_record Appointments%ROWTYPE;

-- Declare a cursor to fetch patients based on doctor\_id

patient\_cursor cursor for

SELECT \* FROM Patients WHERE patient\_id IN (SELECT patient\_id FROM Appointments WHERE doctor\_id = doctor\_id\_to\_query);

BEGIN

-- Open and fetch data from the patient\_cursor

OPEN patient\_cursor;

-- Loop through the cursor and fetch patient details

LOOP

FETCH patient\_cursor INTO patient\_record;

EXIT WHEN NOT FOUND;

-- Fetch appointment details for the patient

SELECT \* INTO appointment\_record FROM Appointments WHERE patient\_id = patient\_record.patient\_id AND doctor\_id = doctor\_id\_to\_query;

-- Display patient and appointment information

RAISE NOTICE 'Patient ID: %, Patient Name: %, Appointment ID: %, Appointment Date: %',

patient\_record.patient\_id, patient\_record.patient\_name,

appointment\_record.appointment\_id, appointment\_record.appointment\_date;

END LOOP;

-- Close the cursor

CLOSE patient\_cursor;

END $$;

--9. to create a cursor query that fetches the patient name,

--doctor name, and appointment date for each medical record

DO $$

DECLARE

patient\_name VARCHAR(100);

doctor\_name VARCHAR(100);

appointment\_date DATE;

medical\_record\_cursor CURSOR FOR

SELECT p.patient\_name, d.doctor\_name, a.appointment\_date

FROM Appointments a

JOIN Patients p ON a.patient\_id = p.patient\_id

JOIN Doctors d ON a.doctor\_id = d.doctor\_id;

BEGIN

-- Open the medical\_record\_cursor

OPEN medical\_record\_cursor;

-- Loop through the cursor and fetch data for each medical record

LOOP

-- Fetch data from the cursor

FETCH medical\_record\_cursor INTO patient\_name, doctor\_name, appointment\_date;

EXIT WHEN NOT FOUND;

-- Display patient name, doctor name, and appointment date

RAISE NOTICE 'Patient Name: %, Doctor Name: %, Appointment Date: %', patient\_name, doctor\_name, appointment\_date;

END LOOP;

-- Close the cursor

CLOSE medical\_record\_cursor;

END $$;

--10. retrieves the number of appointments made by each patient.

CREATE OR REPLACE PROCEDURE GetAppointmentCounts()

LANGUAGE plpgsql

AS $$

DECLARE

patient\_record RECORD;

patient\_id INT;

patient\_name VARCHAR(100);

appointment\_count INT;

BEGIN

-- Fetch patient IDs and their corresponding appointment counts

FOR patient\_record IN (SELECT p.patient\_id, p.patient\_name, COUNT(a.appointment\_id) AS num\_appointments

FROM Patients p

LEFT JOIN Appointments a ON p.patient\_id = a.patient\_id

GROUP BY p.patient\_id, p.patient\_name)

LOOP

-- Assign values to variables

patient\_id := patient\_record.patient\_id;

patient\_name := patient\_record.patient\_name;

appointment\_count := patient\_record.num\_appointments;

-- Display patient name and appointment count

RAISE NOTICE 'Patient ID: %, Patient Name: %, Number of Appointments: %', patient\_id, patient\_name, appointment\_count;

END LOOP;

END;

$$;

CALL GetAppointmentCounts();

--11. Retrive the total medical cost for the given patient id

CREATE OR REPLACE PROCEDURE CalculateTotalMedicalCost(patient\_id\_input INT)

LANGUAGE plpgsql

AS $$

DECLARE

total\_cost NUMERIC := 0;

BEGIN

-- Calculate total medical cost for the given patient

SELECT SUM(total\_cost) INTO total\_cost

FROM MedicalRecords

WHERE patient\_id = patient\_id\_input;

-- Display total medical cost

RAISE NOTICE 'Total Medical Cost for Patient %: %', patient\_id\_input, total\_cost;

END;

$$;

-- Example usage: Call the procedure with a specific patient ID

CALL CalculateTotalMedicalCost(1); -- Replace '1' with the desired patient ID

--12. Retrieve the total number of occupied beds in each department

SELECT d.department\_name, COUNT(b.bed\_number) AS occupied\_beds

FROM Departments d

LEFT JOIN Beds b ON d.department\_id = b.department\_id AND b.status = 'Occupied'

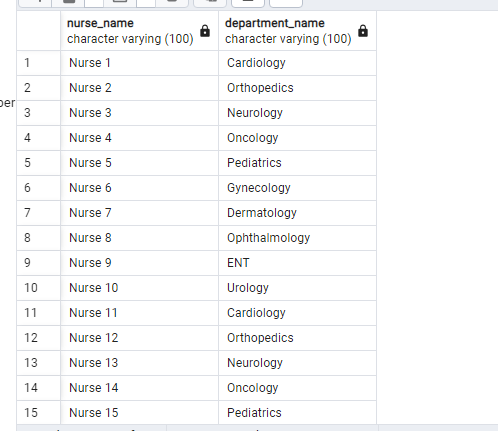
GROUP BY d.department\_name;

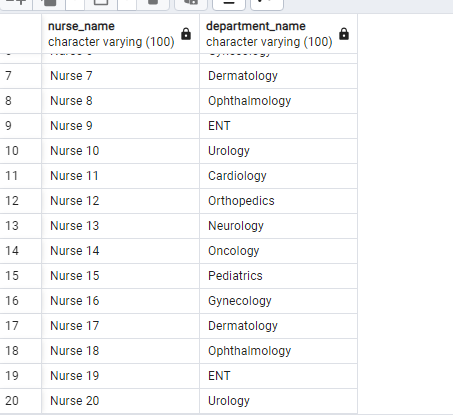
--13. Retrieve nurses along with their assigned departments

SELECT n.nurse\_name, d.department\_name

FROM Nurses n

JOIN Departments d ON n.department\_id = d.department\_id;





--14. Retrieve the department with the highest number of occupied beds

SELECT d.department\_name, COUNT(b.bed\_number) AS occupied\_beds

FROM Departments d

LEFT JOIN Beds b ON d.department\_id = b.department\_id AND b.status = 'Occupied'

GROUP BY d.department\_name

ORDER BY COUNT(b.bed\_number) DESC

LIMIT 1;



--15. Retrieve patients who have had more than one appointment with the same doctor.

SELECT p.patient\_name, d.doctor\_name, COUNT(a.appointment\_id) AS num\_appointments

FROM Patients p

JOIN Appointments a ON p.patient\_id = a.patient\_id

JOIN Doctors d ON a.doctor\_id = d.doctor\_id

GROUP BY p.patient\_name, d.doctor\_name

HAVING COUNT(a.appointment\_id) > 1;

--16. Retrieve doctors and the number of different patients they have treated

SELECT d.doctor\_name, COUNT(DISTINCT a.patient\_id) AS num\_patients

FROM Doctors d

JOIN Appointments a ON d.doctor\_id = a.doctor\_id

GROUP BY d.doctor\_name;