Scenario No. 1: Imagine a **Hospital Management System** which stores Patients information, Doctors information, Appointments, Departments.

Patients: Stores personal details of each patient (PatientID, PatientName, DOB, Address, PhoneNumber)

Doctors: Stores details of doctors (DoctorID, DoctorName, Specialty, DepartmentID)

Appointments: Tracks patient appointments with doctors (AppointmentID, PatientID, DoctorID, AppointmentDate, Created_At)

Departments: Lists various departments in the hospital (DepartmentID, DepartmentName, Location)

Solve the following statements with the help of SQL Queries

1.Find doctors who specialize in 'Cardiology', 'Neurology', or 'Orthopedics'
2.List patients ordered by date of birth (youngest to oldest)
3.Count how many appointments were made today
4.Get doctor names with their respective department names.
5.Procedure to add a new patient:
6.Set default department for a new doctor (if none provided)

Ans :

CREATE TABLE Departments (DepartmentID INT PRIMARY KEY AUTO_INCREMENT, DepartmentName VARCHAR(100), Location VARCHAR(100));

CREATE TABLE Doctors (DoctorID INT PRIMARY KEY AUTO_INCREMENT, DoctorName VARCHAR(100), Specialty VARCHAR(100), DepartmentID INT , FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));

CREATE TABLE Patients (PatientID INT PRIMARY KEY AUTO_INCREMENT, PatientName VARCHAR(100), DOB DATE, Address VARCHAR(200), PhoneNumber VARCHAR(15));

CREATE TABLE Appointments (AppointmentID INT PRIMARY KEY AUTO_INCREMENT, PatientID INT, DoctorID INT, AppointmentDate DATE, CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (PatientID) REFERENCES Patients(PatientID), FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID));

INSERT INTO Departments (DepartmentName, Location) VALUES ('Cardiology', 'Block A'),
('Neurology', 'Block B'), ('Orthopedics', 'Block C'), ('General Medicine', 'Block D');

INSERT INTO Doctors (DoctorName, Specialty, DepartmentID) VALUES ('Dr. Smith', 'Cardiology', 1), ('Dr. Rose', 'Neurology', 2), ('Dr. Allen', 'Orthopedics', 3), ('Dr. Jake', 'General Medicine', 4), ('Dr. Sam', 'Cardiology', 1);

INSERT INTO Patients (PatientName, DOB, Address, PhoneNumber) VALUES ('Alice', '2000-05-14', '123 Street, NY', '1111111111'), ('Bob', '1990-09-10', '456 Lane, LA', '2222222222'), ('Charlie', '1985-12-25', '789 Avenue, TX', '3333333333');

INSERT INTO Appointments (PatientID, DoctorID, AppointmentDate) VALUES (1, 1, CURDATE()), (2, 2, CURDATE()), (3, 3, '2024-12-01');

Query No.1:
SELECT * FROM Doctors WHERE Specialty IN ('Cardiology', 'Neurology', 'Orthopedics');

Query No.2:
SELECT * FROM Patients ORDER BY DOB DESC;

Query No.3:
SELECT COUNT(*) AS TodayAppointments FROM Appointments WHERE DATE(AppointmentDate) = CURDATE();

Query No.4:
SELECT D.DoctorName, Dept.DepartmentName FROM Doctors D JOIN Departments Dept ON D.DepartmentID = Dept.DepartmentID;
Query No. 5:
```
DELIMITER //
CREATE PROCEDURE AddNewPatient(IN p_name VARCHAR(100), IN p_dob DATE,  IN p_address VARCHAR(255), IN p_phone VARCHAR(15))
BEGIN
  INSERT INTO Patients (PatientName, DOB, Address, PhoneNumber) VALUES (p_name, p_dob, p_address, p_phone);
END;
//
DELIMITER ;
```

CALL AddNewPatient('Daisy', '1995-08-20', '12 Rose St, SF', '4444444444');

select * from Patients;

Query No.5:
```
DELIMITER //
CREATE TRIGGER trg_default_doctor_department
BEFORE INSERT ON Doctors
FOR EACH ROW
```

```
BEGIN
  IF NEW.DepartmentID IS NULL THEN
    SET NEW.DepartmentID = 1;
  END IF;
END;
//
DELIMITER ;

INSERT INTO Doctors (DoctorName, Specialty) VALUES ('Dr. John', Dermatology');

select * from Doctors;
```

Scenario 2: Imagine a **Hospital Management System** which stores Patients information, Doctors information, Appointments, Departments.

Patients: Stores personal details of each patient (PatientID, PatientName, DOB, Address, PhoneNumber)

Doctors: Stores details of doctors (DoctorID, DoctorName, Specialty, DepartmentID)

Appointments: Tracks patient appointments with doctors (AppointmentID, PatientID, DoctorID, AppointmentDate, Created_At)

Departments: Lists various departments in the hospital (DepartmentID, DepartmentName, Location)

Solve the following statements with the help of SQL Queries

1.List all doctors ordered by their specialties alphabetically.
2.How many doctors work in each department?
3. Find the total number of appointments per doctor.
4.Show patient names with their appointment dates.
5. Create a procedure to get doctor information by ID.
6.Create a trigger to set appointment date when a new record is added

Ans:

CREATE TABLE Departments (DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100), Location VARCHAR(100));

CREATE TABLE Doctors (DoctorID INT PRIMARY KEY, DoctorName VARCHAR(100), Specialty VARCHAR(100), DepartmentID INT, FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));

CREATE TABLE Patients (PatientID INT PRIMARY KEY, PatientName VARCHAR(100), DOB DATE, Address VARCHAR(255), PhoneNumber VARCHAR(15));

CREATE TABLE Appointments (AppointmentID INT PRIMARY KEY, PatientID INT, DoctorID INT, AppointmentDate DATE, FOREIGN KEY (PatientID) REFERENCES Patients(PatientID), FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID));

INSERT INTO Departments VALUES (1, 'Cardiology', 'Block A'), (2, 'Neurology', 'Block B'), (3, 'Orthopedics', 'Block C');

INSERT INTO Doctors VALUES (101, 'Dr. Smith', 'Cardiologist', 1), (102, 'Dr. John', 'Neurologist', 2), (103, 'Dr. Lisa', 'Orthopedic Surgeon', 3), (104, 'Dr. James', 'Cardiologist',1);

INSERT INTO Patients VALUES (201, 'Alice Brown', '1980-05-20', '123 Elm St', '555-1234'), (202, 'Bob Smith', '1975-09-15', '456 Oak St', '555-5678');

INSERT INTO Appointments VALUES (301, 201, 101, '2025-04-01'), (302, 202, 102, '2025-04-02'), (303, 201, 104, '2025-04-03');

Query 1:
SELECT DoctorName, Specialty, DepartmentID FROM Doctors ORDER BY Specialty ASC;

Query 2:
SELECT DepartmentID, COUNT(*) AS DoctorCount FROM Doctors GROUP BY DepartmentID;

Query 3:
SELECT DoctorID, COUNT(*) AS AppointmentCount FROM Appointments GROUP BY DoctorID;

Query 4:
SELECT p.PatientName, a.AppointmentDate FROM Patients p JOIN Appointments a ON p.PatientID = a.PatientID;

Query 5:
```
DELIMITER //
CREATE PROCEDURE GetDoctorInfo(IN doctor_id INT)
BEGIN
   SELECT DoctorName, Specialty
   FROM Doctors
   WHERE DoctorID = doctor_id;
END //
DELIMITER ;

CALL GetDoctorInfo(101);
```

Query 6:
```
DELIMITER //
CREATE TRIGGER SetAppointmentDate
BEFORE INSERT ON Appointments
FOR EACH ROW
BEGIN
   IF NEW.AppointmentDate IS NULL THEN
      SET NEW.AppointmentDate = CURRENT_DATE;
   END IF;
END //
```

```
DELIMITER ;

INSERT INTO Appointments (AppointmentID, PatientID, DoctorID) VALUES (304, 202,
103);

select * from Appointments;
```

Scenario 3: Imagine a **Hospital Management System** which stores Patients information, Doctors information, Appointments, Departments.

Patients: Stores personal details of each patient (PatientID, PatientName, DOB, Address, PhoneNumber)

Doctors: Stores details of doctors (DoctorID, DoctorName, Specialty, DepartmentID)

Appointments: Tracks patient appointments with doctors (AppointmentID, PatientID, DoctorID, AppointmentDate, Created_At)

Departments: Lists various departments in the hospital (DepartmentID, DepartmentName, Location)

Solve the following statements with the help of SQL Queries

1.Which patients are considered senior citizens (over 65 years old) and might need special care?
2.What is the average age of all patients?
3. List all doctors with their department names.
4.Create a procedure to find appointments for a specific doctor.
5.Find the youngest patient in the hospital.
6.Write a stored procedure using a cursor that goes through all patients and displays the names and phone numbers of those who are above 60 years old (senior citizens).

Ans:

```
CREATE TABLE Patients (PatientID INT PRIMARY KEY, PatientName VARCHAR(100), Age INT, Address VARCHAR(255), PhoneNumber VARCHAR(15));

CREATE TABLE Departments (DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100), Location VARCHAR(100));

CREATE TABLE Doctors (DoctorID INT PRIMARY KEY, DoctorName VARCHAR(100), Specialty VARCHAR(100), DepartmentID INT, FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));

CREATE TABLE Appointments (AppointmentID INT PRIMARY KEY, PatientID INT, DoctorID INT, AppointmentDate DATE, FOREIGN KEY (PatientID) REFERENCES Patients(PatientID), FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID));

INSERT INTO Patients VALUES (1, 'John Doe', 70, '123 Elm Street', '1234567890'),
(2, 'Jane Smith', 30, '456 Oak Street', '0987654321'), (3, 'Mike Johnson', 67, '789 Pine Street', '1122334455'), (4, 'Alice Brown', 25, '321 Maple Street', '6677889900');
```

INSERT INTO Departments VALUES (1, 'Cardiology', 'Block A'), (2, 'Neurology', 'Block B'), (3, 'Pediatrics', 'Block C');

INSERT INTO Doctors VALUES (1, 'Dr. Adams', 'Cardiologist', 1), (2, 'Dr. Baker', 'Neurologist', 2), (3, 'Dr. Clark', 'Pediatrician', 3);

INSERT INTO Appointments VALUES (1, 1, 1, '2025-04-01'), (2, 2, 2, '2025-04-03'), (3, 3, 1, '2025-04-05'), (4, 4, 3, '2025-04-07');

Query 1:
SELECT PatientID, PatientName, Age, PhoneNumber FROM Patients WHERE Age > 65;

Query 2:
SELECT AVG(Age) AS AveragePatientAge FROM Patients;

Query 3:
SELECT d.DoctorName, dept.DepartmentName FROM Doctors d JOIN Departments dept ON d.DepartmentID = dept.DepartmentID;

Query 4:
```
DELIMITER //
CREATE PROCEDURE GetDoctorAppointments(IN doctor_id_param INT)
BEGIN
   SELECT AppointmentID, PatientID, AppointmentDate FROM Appointments WHERE DoctorID = doctor_id_param;
END //
DELIMITER ;

CALL GetDoctorAppointments(1);

SELECT PatientName, Age FROM Patients ORDER BY Age ASC Limit 1;
```

Query 5:
```
DELIMITER //

CREATE PROCEDURE ShowSeniorPatients()
BEGIN
   DECLARE done INT DEFAULT 0;
   DECLARE pName VARCHAR(100);
   DECLARE pAge INT;
   DECLARE pPhone VARCHAR(15);

   DECLARE patient_cursor CURSOR FOR
      SELECT PatientName, Age, PhoneNumber FROM Patients;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN patient_cursor;
    read_loop: LOOP
    FETCH patient_cursor INTO pName, pAge, pPhone;
        IF done THEN
            LEAVE read_loop;
        END IF;

        IF pAge > 60 THEN
            SELECT CONCAT('Name: ', pName, ' | Phone: ', pPhone) AS SeniorCitizen;
        END IF;
    END LOOP;

    CLOSE patient_cursor;
END //

DELIMITER ;

call ShowSeniorPatients();
```

Scenario 4: Imagine a **Hospital Management System** which stores Patients information, Doctors information, Appointments, Departments.

Patients: Stores personal details of each patient (PatientID, PatientName, DOB, Address, PhoneNumber)

Doctors: Stores details of doctors (DoctorID, DoctorName, Specialty, DepartmentID)

Appointments: Tracks patient appointments with doctors (AppointmentID, PatientID, DoctorID, AppointmentDate, Created_At)

Departments: Lists various departments in the hospital (DepartmentID, DepartmentName, Location)

Solve the following statements with the help of SQL Queries

1.List all doctors who work in the Cardiology department.
2.List patients between 30 and 50 years old.
3.Who are the 3 oldest patients in the hospital?
4. Create a procedure that returns patient counts by age groups
5.Create a view that shows patient ID, name, age, and phone number for all patients.
6.Create a function that counts how many appointments a patient has.

Ans:

```
CREATE TABLE Departments (DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100), Location VARCHAR(100));

CREATE TABLE Doctors (DoctorID INT PRIMARY KEY, DoctorName VARCHAR(100), Specialty VARCHAR(100), DepartmentID INT, FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID));

CREATE TABLE Patients (PatientID INT PRIMARY KEY, PatientName VARCHAR(100), Age INT, Address VARCHAR(200), PhoneNumber VARCHAR(15));

CREATE TABLE Appointments (AppointmentID INT PRIMARY KEY, PatientID INT, DoctorID INT, AppointmentDate DATE, FOREIGN KEY (PatientID) REFERENCES Patients(PatientID), FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID));

INSERT INTO Departments VALUES (1, 'Cardiology', 'Block A');
INSERT INTO Departments VALUES (2, 'Neurology', 'Block B');
INSERT INTO Departments VALUES (3, 'Orthopedics', 'Block C');

INSERT INTO Doctors VALUES (101, 'Dr. Smith', 'Cardiologist', 1);
INSERT INTO Doctors VALUES (102, 'Dr. Adams', 'Neurologist', 2);
```

```
INSERT INTO Doctors VALUES (103, 'Dr. Ray', 'Orthopedic Surgeon', 3);

INSERT INTO Patients VALUES (201, 'John Doe', 45, '123 Elm St', '1234567890');
INSERT INTO Patients VALUES (202, 'Jane Roe', 32, '456 Oak St', '9876543210');
INSERT INTO Patients VALUES (203, 'Mary Lane', 60, '789 Pine St', '4561237890');
INSERT INTO Patients VALUES (204, 'Peter Hall', 50, '321 Birch St', '3216549870');
INSERT INTO Patients VALUES (205, 'Lucy Gray', 29, '654 Maple St', '7418529630');

INSERT INTO Appointments VALUES (301, 201, 101, '2025-04-01');
INSERT INTO Appointments VALUES (302, 202, 101, '2025-04-02');
INSERT INTO Appointments VALUES (303, 203, 103, '2025-04-03');
INSERT INTO Appointments VALUES (304, 204, 102, '2025-04-04');
INSERT INTO Appointments VALUES (305, 201, 101, '2025-04-05');
```

Query 1:

```
SELECT d.DoctorName, d.Specialty FROM Doctors d JOIN Departments dept ON
d.DepartmentID = dept.DepartmentID WHERE dept.DepartmentName = 'Cardiology';
```

Query 2:

```
SELECT PatientName, Age FROM Patients WHERE Age BETWEEN 30 AND 50;
```

Query 3:

```
SELECT PatientName, Age FROM Patients ORDER BY Age DESC LIMIT 3;
```

Query 4:

```
DELIMITER //
CREATE PROCEDURE GetPatientAgeGroups()
BEGIN
  SELECT
    CASE
      WHEN Age < 18 THEN 'Under 18'
      WHEN Age BETWEEN 18 AND 30 THEN '18-30'
      WHEN Age BETWEEN 31 AND 50 THEN '31-50'
      WHEN Age BETWEEN 51 AND 70 THEN '51-70'
      ELSE 'Over 70'
    END AS AgeGroup,
    COUNT(*) AS PatientCount
  FROM Patients
  GROUP BY AgeGroup
  ORDER BY MIN(Age);
END//
DELIMITER ;

CALL GetPatientAgeGroups();
```

Query 5:

```
CREATE VIEW PatientDirectory AS SELECT PatientID, PatientName, Age, PhoneNumber
FROM Patients;

select * from PatientDirectory;
```

Query 6:

```
DELIMITER //
CREATE FUNCTION CountAppointments(p_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
   DECLARE appt_count INT;

   SELECT COUNT(*) INTO appt_count
   FROM Appointments
   WHERE PatientID = p_id;

   RETURN appt_count;
END //
DELIMITER ;


SELECT PatientName, CountAppointments(PatientID) AS AppointmentCount FROM
Patients;
```