SQL case-based assignment for a Doctor-Patient Appointment System. This scenario involves managing data related to doctors, patients, appointments, specialties, and hospital departments. I'll provide the database schema followed by 10 SQL queries based on the system's requirements.

Database Schema

```
CREATE DATABASE HOSPITAL;

USE HOSPITAL;
```

DEPARTMENTS TABLE

```
CREATE TABLE DEPARTMENTS (
DEPARTMENT_ID INT PRIMARY KEY,
DEPARTMENT_NAME VARCHAR(100) );
```

```
INSERT INTO DEPARTMENTS (DEPARTMENT_ID, DEPARTMENT_NAME) VALUES

(1, 'Cardiology'),

(2, 'Neurology'),

(3, 'Orthopedics'),

(4, 'Pediatrics'),

(5, 'General Surgery'),

(6, 'Dermatology'),

(7, 'Dentistry'),

(8, 'Endocrinology'),

(9, 'Gynecology'),

(10, 'Ophthalmology');
```

SPECIALTIES TABLE

```
CREATE TABLE SPECIALTIES (
SPECIALTY_ID INT PRIMARY KEY,
SPECIALTY_NAME VARCHAR(100) );

INSERT INTO SPECIALTIES (SPECIALTY_ID, SPECIALTY_NAME) VALUES
(1, 'Cardiologist'),
(2, 'Neurologist'),
(3, 'Orthopedic Surgeon'),
(4, 'Pediatrician'),
(5, 'General Surgeon'),
(6, 'Dermatologist'),
(7, 'Dentist'),
(8, 'Endocrinologist'),
(9, 'Gynecologist'),
(10, 'Ophthalmologist');
```

PATIENTS TABLE

```
CREATE TABLE PATIENTS (
PATIENT_ID INT PRIMARY KEY,
FIRST_NAME VARCHAR(100),
LAST_NAME VARCHAR(100),
EMAIL VARCHAR(100),
PHONE VARCHAR(20),
DATE_OF_BIRTH DATE,
GENDER VARCHAR(10),
ADDRESS TEXT );

INSERT INTO PATIENTS (PATIENT_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE, DATE_OF_BIRTH,
GENDER, ADDRESS)
VALUES
   (1, 'John', 'Doe', 'john.doe@email.com', '1234567890', '1985-07-12', 'Male', '123 Main St, New York,
NY'),
   (2, 'Jane', 'Smith', 'jane.smith@email.com', '9876543210', '1990-03-22', 'Female', '456 Elm St, Los
Angeles, CA'),
   (3, 'Amit', 'Kumar', 'amit.kumar@email.com', '9087654321', '1978-11-02', 'Male', '789 Oak St, Delhi,
India'),
```

(4, 'Sara', 'Lee', 'sara.lee@email.com', '9871234560', '2001-01-15', 'Female', '321 Pine St, Toronto, Canada'),
(5, 'Ravi', 'Patel', 'ravi.patel@email.com', '9012345678', '1995-06-30', 'Male', '654 Maple St, Mumbai, India');

DOCTORS TABLE

```
CREATE TABLE DOCTORS (
DOCTOR_ID INT PRIMARY KEY,
FIRST_NAME VARCHAR(100),
LAST_NAME VARCHAR(100),
EMAIL VARCHAR(100),
PHONE VARCHAR(20),
DEPARTMENT_ID INT,
SPECIALTY_ID INT,
JOINING_DATE DATE,
FOREIGN KEY(DEPARTMENT_ID) REFERENCES DEPARTMENTS(DEPARTMENT_ID),
FOREIGN KEY(SPECIALTY_ID) REFERENCES SPECIALTIES(SPECIALTY_ID) );

INSERT INTO DOCTORS (DOCTOR_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE, DEPARTMENT_ID,
SPECIALTY_ID, JOINING_DATE)
VALUES
(1, 'Alice', 'Brown', 'alice.brown@email.com', '9876543210', 1, 1, '2020-01-15'),
(2, 'Bob', 'Johnson', 'bob.johnson@email.com', '8765432109', 2, 2, '2019-03-12'),
(3, 'Catherine', 'Davis', 'catherine.davis@email.com', '7654321098', 3, 3, '2021-07-01'),
(4, 'David', 'Wilson', 'david.wilson@email.com', '6543210987', 4, 4, '2018-09-25'),
(5, 'Eva', 'Martin', 'eva.martin@email.com', '5432109876', 5, 5, '2022-06-10');
```

APPOINTMENTS TABLE

```
CREATE TABLE APPOINTMENTS (
APPOINTMENT_ID INT PRIMARY KEY,
DOCTOR_ID INT,
PATIENT_ID INT,
APPOINTMENT_DATE DATETIME,
```

```
REASON TEXT,
STATUS VARCHAR(20),
FOREIGN KEY(DOCTOR_ID) REFERENCES DOCTORS(DOCTOR_ID),
FOREIGN KEY(PATIENT_ID) REFERENCES PATIENTS(PATIENT_ID) );

INSERT INTO APPOINTMENTS (APPOINTMENT_ID, DOCTOR_ID, PATIENT_ID, APPOINTMENT_DATE,
REASON, STATUS)
VALUES
   (1, 1, 1, '2025-05-01 09:00:00', 'Routine check-up', 'Completed'),
   (2, 2, 2, '2025-05-02 10:30:00', 'Headache and dizziness', 'Completed'),
   (3, 3, 3, '2025-05-03 11:00:00', 'Knee pain', 'Scheduled'),
   (4, 4, 4, '2025-05-04 14:00:00', 'Fever and cold', 'Cancelled'),
   (5, 5, 5, '2025-05-05 15:30:00', 'Skin rash', 'Scheduled');
```

PAYMENTS Table

```
CREATE TABLE PAYMENTS (
   PAYMENT_ID INT PRIMARY KEY,
   APPOINTMENT_ID INT,
   PAYMENT_DATE DATE,
   PAYMENT_AMOUNT DECIMAL(10,2),
   PAYMENT_METHOD VARCHAR(20),
   FOREIGN KEY (APPOINTMENT_ID) REFERENCES APPOINTMENTS(APPOINTMENT_ID)
);

-- Insert Multiple Payment Records
INSERT INTO PAYMENTS (PAYMENT_ID, APPOINTMENT_ID, PAYMENT_DATE, PAYMENT_AMOUNT,
PAYMENT_METHOD)
VALUES
   (1, 1, '2025-05-01', 100.00, 'Cash'),
   (2, 2, '2025-05-02', 150.50, 'Credit Card'),
   (3, 3, '2025-05-03', 200.00, 'UPI'),
   (4, 5, '2025-05-05', 120.00, 'Debit Card');
```

SQL Queries

1. Find the Total Number of Appointments for Each Doctor

```
SELECT
   d.DOCTOR_ID,
   d.FIRST_NAME,
   d.LAST_NAME,
   COUNT(a.APPOINTMENT_ID) AS TOTAL_APPOINTMENTS
FROM DOCTORS d
LEFT JOIN APPOINTMENTS a ON d.DOCTOR_ID = a.DOCTOR_ID
GROUP BY d.DOCTOR_ID, d.FIRST_NAME, d.LAST_NAME;
```

2. List All Patients Who Have an Appointment with a Specific Doctor (e.g., Dr. John Smith)

```
SELECT
      p.PATIENT_ID,
   p.FIRST_NAME,
   p.LAST_NAME,
      p.EMAIL,
      p.PHONE
   FROM PATIENTS p
JOIN APPOINTMENTS a ON p.PATIENT_ID = a.PATIENT_ID
JOIN DOCTORS d ON a.DOCTOR_ID = d.DOCTOR_ID
WHERE d.FIRST_NAME = 'John' AND d.LAST_NAME = 'Smith';
```

3. Find the Number of Appointments Scheduled in a Specific Department

```
SELECT
```

```
    dep.DEPARTMENT_NAME,
    COUNT(a.APPOINTMENT_ID) AS TOTAL_APPOINTMENTS
FROM APPOINTMENTS a
JOIN DOCTORS d ON a.DOCTOR_ID = d.DOCTOR_ID
JOIN DEPARTMENTS dep ON d.DEPARTMENT_ID = dep.DEPARTMENT_ID
WHERE dep.DEPARTMENT_NAME = 'Cardiology'
GROUP BY dep.DEPARTMENT_NAME;
```

4. Find the Most Popular Specialty Based on Number of Appointments

```
SELECT
    s.SPECIALTY_NAME,
    COUNT(a.APPOINTMENT_ID) AS TOTAL_APPOINTMENTS
FROM APPOINTMENTS a
JOIN DOCTORS d ON a.DOCTOR_ID = d.DOCTOR_ID
JOIN SPECIALTIES s ON d.SPECIALTY_ID = s.SPECIALTY_ID
GROUP BY s.SPECIALTY_NAME
ORDER BY TOTAL_APPOINTMENTS DESC
LIMIT 1;
```

5. Get the Total Payment Amount for All Completed Appointments

```
SELECT

   SUM(p.PAYMENT_AMOUNT) AS TOTAL_PAYMENTS

FROM PAYMENTS p

JOIN APPOINTMENTS a ON p.APPOINTMENT_ID = a.APPOINTMENT_ID

WHERE a.STATUS = 'Completed';
```

6. Find the Number of Patients Seen by Each Doctor

```
SELECT
    d.DOCTOR_ID,
    d.FIRST_NAME,
    d.LAST_NAME,
    COUNT(DISTINCT a.PATIENT_ID) AS PATIENTS_SEEN
FROM DOCTORS d
LEFT JOIN APPOINTMENTS a ON d.DOCTOR_ID = a.DOCTOR_ID AND a.STATUS = 'Completed'
GROUP BY d.DOCTOR_ID, d.FIRST_NAME, d.LAST_NAME;
```

7. List All Patients Who Have Missed Their Appointments (Status 'Cancelled')

```
SELECT
    DISTINCT p.PATIENT_ID,
    p.FIRST_NAME,
    p.LAST_NAME,
```

```
    p.EMAIL,
    p.PHONE
FROM PATIENTS p
JOIN APPOINTMENTS a ON p.PATIENT_ID = a.PATIENT_ID
WHERE a.STATUS = 'Cancelled';
```

8. Find the Total Number of Appointments for Each Status (Scheduled, Completed, Cancelled)

```
SELECT

   STATUS,

   COUNT(APPOINTMENT_ID) AS TOTAL_APPOINTMENTS

FROM APPOINTMENTS

GROUP BY STATUS;
```

9.Get the Average Payment Amount for Completed Appointments

```
SELECT
   AVG(p.PAYMENT_AMOUNT) AS AVERAGE_PAYMENT
FROM PAYMENTS p
JOIN APPOINTMENTS a ON p.APPOINTMENT_ID = a.APPOINTMENT_ID
WHERE a.STATUS = 'Completed';
```

10.Find the Doctor with the Highest Number of Appointments

```
SELECT

   d.DOCTOR_ID,

   d.FIRST_NAME,

   d.LAST_NAME,

   COUNT(a.APPOINTMENT_ID) AS TOTAL_APPOINTMENTS

FROM DOCTORS d

JOIN APPOINTMENTS a ON d.DOCTOR_ID = a.DOCTOR_ID

GROUP BY d.DOCTOR_ID, d.FIRST_NAME, d.LAST_NAME

ORDER BY TOTAL_APPOINTMENTS DESC
```

LIMIT 1;