

**Internship Task** - **RDBMS and SQL Task #6**

**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 Doctors Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| doctor\_id | INT | Primary key |
| first\_name | VARCHAR(100) | Doctor's first name |
| last\_name | VARCHAR(100) | Doctor's last name |
| email | VARCHAR(100) | Doctor's email address |
| phone | VARCHAR(20) | Doctor's phone number |
| department\_id | INT | Foreign key (references Departments) |
| specialty\_id | INT | Foreign key (references Specialties) |
| joining\_date | DATE | Date the doctor joined the hospital |

**Patients Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| patient\_id | INT | Primary key |
| first\_name | VARCHAR(100) | Patient's first name |
| last\_name | VARCHAR(100) | Patient's last name |
| email | VARCHAR(100) | Patient's email address |
| phone | VARCHAR(20) | Patient's phone number |
| date\_of\_birth | DATE | Patient's date of birth |
| gender | VARCHAR(10) | Patient's gender |
| address | TEXT | Patient's address |

**Departments Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| department\_id | INT | Primary key |
| department\_name | VARCHAR(100) | Department name |

**Specialties Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| specialty\_id | INT | Primary key |
| specialty\_name | VARCHAR(100) | Specialty name (e.g., Cardiology, Dermatology) |

**Appointments Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| appointment\_id | INT | Primary key |
| doctor\_id | INT | Foreign key (references Doctors) |
| patient\_id | INT | Foreign key (references Patients) |
| appointment\_date | DATETIME | Date and time of appointment |
| reason | TEXT | Reason for the appointment |
| status | VARCHAR(20) | Status (e.g., 'Scheduled', 'Completed', 'Cancelled') |

**Payments Table:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| payment\_id | INT | Primary key |
| appointment\_id | INT | Foreign key (references Appointments) |
| payment\_date | DATE | Date the payment was made |
| payment\_amount | DECIMAL(10,2) | Amount paid |
| payment\_method | VARCHAR(20) | Payment method (e.g., 'Cash', 'Credit Card', 'Insurance') |

**Case Study: Doctor-Patient Appointment System Background:**

You are a database analyst for a healthcare system managing doctor-patient appointments. You are tasked with generating reports based on appointment data, doctor details, patient information, and payments.

**SQL Queries for the Case Study**

1. **Find the Total Number of Appointments for Each Doctor**

**Ans :**

SELECT doctor.first\_name, doctor.last\_name, COUNT(appointment.appointment\_id) AS TotalAppointments

FROM Doctors doctor

LEFT JOIN Appointments appointment ON doctor.doctor\_id = appointment.doctor\_id

GROUP BY doctor.doctor\_id, doctor.first\_name, doctor.last\_name;

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

**Ans :**

SELECT patient.first\_name, patient.last\_name

FROM Patients patient

JOIN Appointments appointment ON patient.patient\_id = appointment.patient\_id

JOIN Doctors doctor ON appointment.doctor\_id = doctor.doctor\_id

WHERE doctor.first\_name = 'John' AND doctor.last\_name = 'Smith';

1. **Find the Number of Appointments Scheduled in a Specific Department**

**Ans :**

SELECT department.department\_name, COUNT(appointment.appointment\_id) AS TotalAppointments

FROM Appointments appointment

JOIN Doctors doctor ON appointment.doctor\_id = doctor.doctor\_id

JOIN Departments department ON doctor.department\_id = department.department\_id

GROUP BY department.department\_name;

1. **Find the Most Popular Specialty Based on Number of Appointments**

**Ans :**

SELECT specialty.specialty\_name, COUNT(appointment.appointment\_id) AS TotalAppointments

FROM Appointments appointment

JOIN Doctors doctor ON appointment.doctor\_id = doctor.doctor\_id

JOIN Specialties specialty ON doctor.specialty\_id = specialty.specialty\_id

GROUP BY specialty.specialty\_name

ORDER BY TotalAppointments DESC

LIMIT 1;

1. **Get the Total Payment Amount for All Completed Appointments**

**Ans :**

SELECT SUM(payment.payment\_amount) AS TotalPaymentAmount

FROM Payments payment

JOIN Appointments appointment ON payment.appointment\_id = appointment.appointment\_id

WHERE appointment.status = 'Completed';

1. **Find the Number of Patients Seen by Each Doctor**

**Ans:**

SELECT doctor.first\_name, doctor.last\_name, COUNT(DISTINCT appointment.patient\_id) AS NumberOfPatientsSeen

FROM Doctors doctor

LEFT JOIN Appointments appointment ON doctor.doctor\_id = appointment.doctor\_id

GROUP BY doctor.doctor\_id, doctor.first\_name, doctor.last\_name;

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

**Ans :**

SELECT patient.first\_name, patient.last\_name

FROM Patients patient

JOIN Appointments appointment ON patient.patient\_id = appointment.patient\_id

WHERE appointment.status = 'Cancelled';

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

**Ans :**

SELECT appointment.status, COUNT(appointment.appointment\_id) AS TotalAppointments

FROM Appointments appointment

GROUP BY appointment.status;

1. **Get the Average Payment Amount for Completed Appointments**

**Ans :**

SELECT AVG(payment.payment\_amount) AS AveragePaymentAmount

FROM Payments payment

JOIN Appointments appointment ON payment.appointment\_id = appointment.appointment\_id

WHERE appointment.status = 'Completed';

1. **Find the Doctor with the Highest Number of Appointments**

**Ans :**

SELECT doctor.first\_name, doctor.last\_name, COUNT(appointment.appointment\_id) AS TotalAppointments

FROM Doctors doctor

LEFT JOIN Appointments appointment ON doctor.doctor\_id = appointment.doctor\_id

GROUP BY doctor.doctor\_id, doctor.first\_name, doctor.last\_name

ORDER BY TotalAppointments DESC

LIMIT 1;

**Task Summary:**

This SQL case study involves the **Doctor-Patient Appointment System**, focusing on querying and analyzing data related to **doctors**, **patients**, **appointments**, **specialties**, **departments**, and **payments**.

The queries provide insights into:

* 1. **Doctor workload and appointments**: Find how many appointments each doctor has or the number of patients seen.
  2. **Department and specialty analysis**: Understand the appointment distribution by department or specialty.
  3. **Patient behavior**: Track missed appointments, total payments, and payments per appointment status.
  4. **Financial analysis**: Calculate total payments for completed appointments and average payment amounts.

**Key SQL concepts used:**

* + JOIN (to combine data from multiple tables)
  + GROUP BY (to group data based on specific columns like doctor, department, or status)
  + COUNT() and SUM() (to calculate totals)
  + ORDER BY (to rank or sort data based on specific metrics)
  + AVG() (to calculate average payment amounts)