Tutorial 7

The following schema is used for all exercises below (related to the Hospital Management Database):

Table Name	Key Column(s)	Other Columns	Relationships *
`Patients`	`PatientID` (PK)	`FirstName` , `LastName` , `Address` , `Gender` , `DateOfBirth`	
`Doctors`	`DoctorID` (PK)	`DocFirstName` , `DocLastName` , `Specialty` , `HireDate` , `Email`	
`Appointments`	`ApptID` (PK)	`PatientID` (FK), `DoctorID` (FK), `ApptDate` , `Status`	`PatientID` → `Patients` , `DoctorID` → `Doctors`
`Rooms`	`RoomID` (PK)	`RoomType` , `Floor` , `Capacity`	
`Admissions`	`AdmissionID` (PK)	`PatientID` (FK), `RoomID` (FK), `AdmissionDate` , `DischargeDate`	`PatientID` → `Patients` , `RoomID` → `Rooms`

* `attribute_name` → `Table_name` , (ex: `PatientID` → `Patients`), means that column `attribute_name` (ex: `PatientID`) is the primary key of the relation `Table_name` (ex: `Patients`)

The database name is: HMDB

Exercise 1:

Write the SQL queries that:

1. List the **first name** and **last name** of all **patients** who live on 'Main Street'
2. Find the `DoctorID` , `DocLastName` , and `Specialty` for all doctors hired **before January 1, 2020**.
3. Display the `ApptID` and `ApptDate` for all appointments that have a `Status` of 'Cancelled'.
4. List the `RoomID` and `Floor` for all rooms with a `Capacity` of exactly 1 or 2.
5. Find the **number of doctors** whose email address ends with `@hospital.org`
6. Display the **last name** and **date of birth** of all patients born **before 1990**.

Exercise 2:

Write the SQL queries that:

1. Count the **number of doctors** in each `Specialty`.
2. Calculate the **total capacity** for rooms on each `Floor`.
3. Find the **average** `Capacity` of all rooms.
4. Show the total **number of appointments** grouped by their `Status` (e.g., 'Scheduled', 'Cancelled', 'Completed').

Exercise 3:

Write the SQL queries that:

1. List the `PatientID` and the **number of admissions** for all patients who have been admitted **more than twice**.
2. Find the `Specialty` where the total number of doctors in that specialty **exceeds 5**.
3. Display the `DoctorID` and the **total count of appointments** for doctors who have **fewer than 10 appointments** on record.
4. Find the `RoomType` where the average capacity is **greater than 3**.

Exercise 4:

Write the SQL queries that:

1. Find the **oldest** `DateOfBirth` and the **youngest** `DateOfBirth` among all patients.
2. Calculate the **age in days** for each admission by finding the difference between the `AdmissionDate` and the current date. Consider only the patients that have been discharged.
3. List the `FirstName` and `LastName` of patients in a single column labeled `FullName`, separated by a comma and a space.
4. Display the `PatientID`, `AdmissionDate`, and `DischargeDate` for all admissions that have been ongoing (no `DischargeDate`) for **more than 30 days**.

Exercise 5:

Write the SQL queries that:

1. List the `FirstName` and `LastName` of all **patients** who have an appointment with a doctor specializing in **'Cardiology'**.
2. Find the `RoomType` and the **number of admissions** associated with each room type. Include room that have **never been used** for admission.
3. List the `DocFirstName` and `DocLastName` of doctors who **do not have any appointments** scheduled.
4. Display the `FirstName` and `LastName` of patients who were admitted to a room on the **'1st Floor'**.

5. Find the `PatientID` and the **total number of different rooms** they have been admitted to, but **only for patients admitted to more than 3 unique rooms**.