Tutorial 7 Solutions

Exercise 1:

1. **List the first name and last name of all patients who live on 'Main Street' (use a wildcard search).**

   ```
   use HMDB;
   SELECT FirstName, LastName
   FROM Patients
   WHERE Address LIKE '%Main Street%';
   ```

2. **Find the `DoctorID`, `DocLastName`, and `Specialty` for all doctors hired before January 1, 2020.**

   ```
   SELECT DoctorID, DocLastName, Specialty
   FROM Doctors
   WHERE HireDate < '2020-01-01';
   ```

3. **Display the `ApptID` and `ApptDate` for all appointments that have a `Status` of 'Cancelled'.**

   ```
   SELECT ApptID, ApptDate
   FROM Appointments
   WHERE Status = 'Cancelled';
   ```

4. **List the `RoomID` and `Floor` for all rooms with a `Capacity` of exactly 1 or 2.**

   ```
   SELECT RoomID, Floor
   FROM Rooms
   WHERE Capacity IN (1, 2);
   ```

5. **Find the number of doctors whose email address ends with `@hospital.org`.**

   ```
   SELECT COUNT(DoctorID) AS NumberOfHospitalDoctors
   FROM Doctors
   WHERE Email LIKE '%@hospital.org';
   ```

6. **Display the last name and date of birth of all patients born before 1990.**

   ```
   SELECT LastName, DateOfBirth
   FROM Patients
   WHERE DateOfBirth < '1990-01-01';
   ```

Exercise 2:

1. **Count the number of doctors in each `Specialty`.**

```sql
SELECT Specialty, COUNT(DoctorID) AS NumberOfDoctors
FROM Doctors
GROUP BY Specialty;
```

2. **Calculate the total capacity for rooms on each `Floor`.**

```sql
SELECT Floor, SUM(Capacity) AS TotalFloorCapacity
FROM Rooms
GROUP BY Floor;
```

3. **Find the average `Capacity` of all rooms.**

```sql
SELECT AVG(Capacity) AS AverageRoomCapacity
FROM Rooms;
```

4. **Show the total number of appointments grouped by their `Status`.**

```sql
SELECT Status, COUNT(ApptID) AS TotalAppointments
FROM Appointments
GROUP BY Status;
```

Exercise 3:

1. **List the `PatientID` and the number of admissions for all patients who have been admitted more than twice.**

```sql
SELECT PatientID, COUNT(AdmissionID) AS NumberOfAdmissions
FROM Admissions
GROUP BY PatientID
HAVING COUNT(AdmissionID) > 2;
```

2. **Find the `Specialty` where the total number of doctors in that specialty exceeds 5.**

```sql
SELECT Specialty, COUNT(DoctorID) AS DoctorCount
FROM Doctors
GROUP BY Specialty
HAVING COUNT(DoctorID) > 5;
```

3. **Display the `DoctorID` and the total count of appointments for doctors who have fewer than 10 appointments on record.**

```sql
SELECT DoctorID, COUNT(ApptID) AS TotalAppointments
FROM Appointments
GROUP BY DoctorID
HAVING COUNT(ApptID) < 10;
```

4. **Find the `RoomType` where the average capacity is greater than 3.**

```sql
SELECT RoomType, AVG(Capacity) AS AverageCapacity
FROM Rooms
```

```
GROUP BY RoomType
HAVING AVG(Capacity) > 3;
```

Exercise 4:

1. **Find the oldest `DateOfBirth` and the youngest `DateOfBirth` among all patients.**

```
 SELECT MIN(DateOfBirth) AS OldestPatientDOB, MAX(DateOfBirth) AS
YoungestPatientDOB
 FROM Patients;
```

2. **Calculate the age in days for each admission by finding the difference between the `AdmissionDate`. Consider only the patients that have been discharged**.

```
SELECT AdmissionID, AdmissionDate, DATEDIFF(DAY, AdmissionDate,
GETDATE()) AS DaysSinceAdmission
FROM Admissions
WHERE DischargeDate IS NULL;
```

3. **List the `FirstName` and `LastName` of patients in a single column labeled `FullName`, separated by a comma and a space.**

```
SELECT CONCAT(LastName, ', ', FirstName) AS FullName
FROM Patients;
```

4. **Display the `PatientID`, `AdmissionDate` for all admissions that have been ongoing (no `DischargeDate`) for more than 30 days.**

```
SELECT PatientID, AdmissionDate
FROM Admissions
WHERE DischargeDate IS NULL
  AND DATEDIFF(DAY, AdmissionDate, GETDATE()) > 30;
```

Exercise 5:

1. **List the `FirstName` and `LastName` of all patients who have an appointment with a doctor specializing in 'Cardiology'.**

```
SELECT DISTINCT Patients.FirstName, Patients.LastName
FROM Patients
INNER JOIN Appointments  ON Patients.PatientID = Appointments.PatientID
INNER JOIN Doctors  ON Appointments.DoctorID = Doctors.DoctorID
WHERE Doctors.Specialty = 'Cardiology';
```

2. **Find the `RoomType` and the number of admissions associated with each room type. Include room that have never been used for admission.**

```
 SELECT Rooms.RoomType, COUNT(Admissions.AdmissionID) AS
NumberOfAdmissions
FROM Rooms
```

```
LEFT OUTER JOIN Admissions  ON Rooms.RoomID = Admissions.RoomID
GROUP BY Rooms.RoomType;
```

3. **List the `DocFirstName` and `DocLastName` of doctors who do not have any appointments scheduled.**

```
SELECT Doctors.DocFirstName, Doctors.DocLastName
FROM Doctors
LEFT OUTER JOIN Appointments ON Doctors.DoctorID = Appointments.DoctorID
WHERE Appointments.ApptID IS NULL;
```

4. **Display the `FirstName` and `LastName` of patients who were admitted to a room on the '1st Floor'.**

```
SELECT DISTINCT Patients.FirstName, Patients.LastName
FROM Patients
INNER JOIN Admissions ON Patients.PatientID = Admissions.PatientID
INNER JOIN Rooms  ON Admissions.RoomID = Rooms.RoomID
WHERE Rooms.Floor = 1;
```

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.**

```
SELECT PatientID, COUNT(DISTINCT RoomID) AS UniqueRoomsCount
FROM Admissions
GROUP BY PatientID
HAVING COUNT(DISTINCT RoomID) > 3;
```