Business Intelligence Project

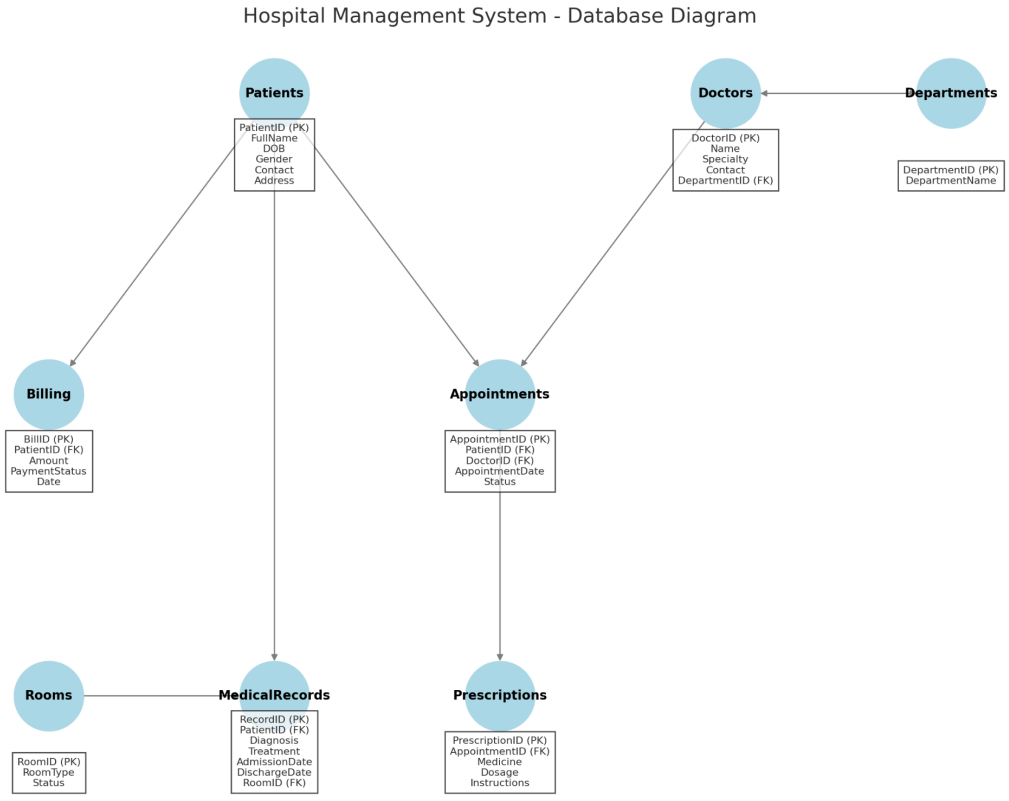
Instruction:

1. Need to complete these projects on your own.
2.Try to understand the project by studying it multiple times.
3.Act like you're a Sr Business Intelligent and your manager tasked you with this project.
4.Dont ask any foolish question, only ask relevant questions and be prepared to face further counter questions related to the project.

1. SQL Server Development

Task 01: Create tables

Create all of these tables in a database



Must ensure:

1. Primary keys of all the tables should be assigned automatically.
2. "Appointments" table "Status" column only support these three data
   i.Scheduled
   ii.Completed
   iii.Cancelled

   If it tries to insert different data rather than these three then it should give an error.

3. "Billing" table "PaymentStatus" column should only support these three data
   i.Paid
   ii.Unpaid
   iii.Pending

   If it tries to insert different data rather than these three then it should give an error.

4. "Rooms" table "Status" column should only support these three data
   i.Available
   ii.Occupied
   iii.Under Maintenance

   If it tries to insert different data rather than these three then it should give an error.

Task 02: Make Stored Procedure

1. **Create a stored procedure to schedule a new appointment.**

   You are required to create a stored procedure named **ScheduleAppointment** that schedules a new appointment for a patient with a doctor.

   - **Requirements:**

     - The procedure should accept the following parameters:
       - @PatientID – ID of the patient booking the appointment.
       - @DoctorID – ID of the doctor assigned.
       - @AppointmentDate– Date and time of the appointment.

    - @Status– Appointment status (Scheduled, Completed, or Cancelled).
- It should **insert a new record** into the Appointments table.
- The procedure should print a message **"Appointment Scheduled Successfully"** after insertion.

2. **Create a stored procedure retrieve patient history**

   Create a stored procedure named **GetPatientHistory** that retrieves the complete **medical history** of a patient.

   🔹 **Requirements:**

   - The procedure should accept **@PatientID** as input.
   - It should retrieve:
     - FullName (Patient's Name)
     - AppointmentDate (Date of appointment)
     - Doctor Name (Who treated the patient)
     - Diagnosis (If available)
     - Treatment (If available)
   - The procedure should **use JOINs** between the Patients, Appointments, Doctors, and MedicalRecords tables to return results.
   - Results should be **ordered by Appointment Date (Latest first).**

Task 03: Make Views

1. **Doctor-wise Appointments**

   Create a **SQL View** named DoctorAppointments that displays the total number of appointments each doctor has.

   - The view should include the doctor's **name**, **specialty**, and the **total number of appointments** assigned to them.
   - Ensure all doctors are included in the result, even if they have **no appointments**.
   - Use the COUNT function to calculate the number of appointments for each doctor.
   - Group the results by **doctor's name** and **specialty**.

2.Patients Admitted per Department

You are required to create a **view** named **PatientsPerDepartment** that calculates the **total number of patients admitted per department**.

Requirements:

- The view should retrieve:
  - **Department Name** from the Departments table.
  - **Total Patients** admitted to that department.
- A patient is considered **admitted** if they have an **appointment** and a **medical record** in the system.
- You must use **JOINs** between:
  - Departments
  - Doctors
  - Appointments
  - MedicalRecords
- Use **COUNT(DISTINCT PatientID)** to ensure unique patient counts.
- **GROUP BY Department Name** to get results department-wise.