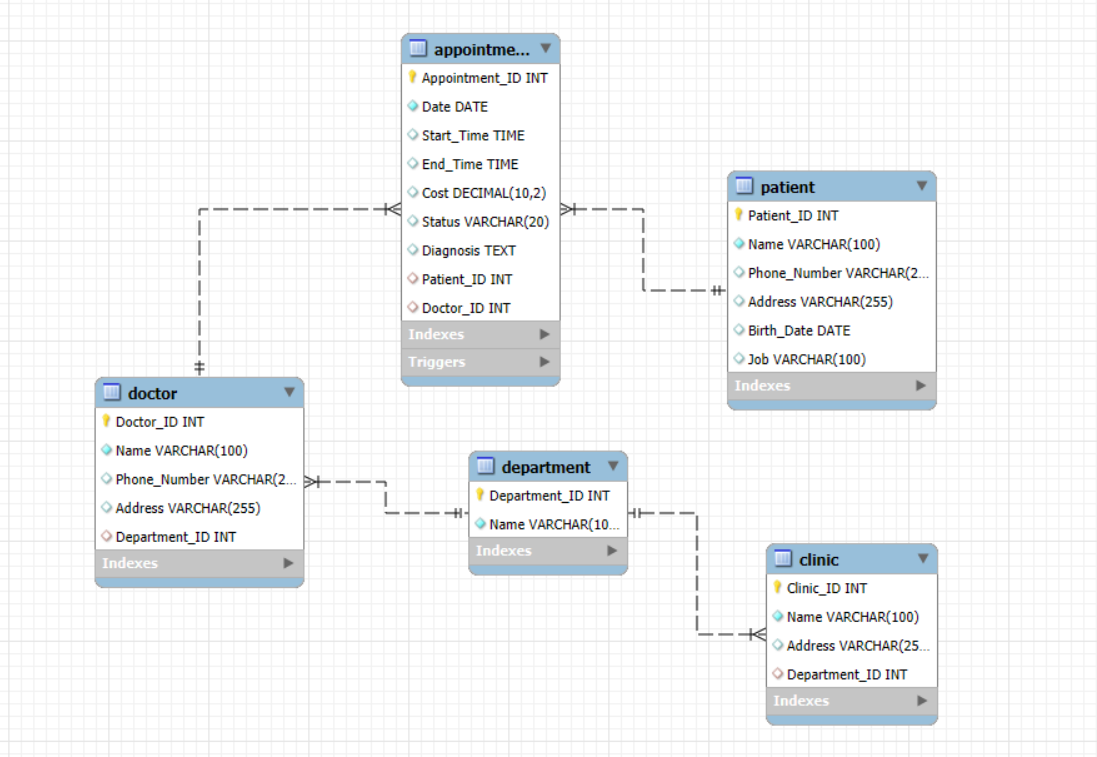
|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| Department | Department\_ID (Primary Key) |
| Name |
| Clinic | Clinic\_ID (Primary Key) |
| Name |
| Address |
| Department\_ID (Foreign Key → Department) |
| Doctor | Doctor\_ID (Primary Key) |
| Name |
| Phone\_Number |
| Address |
| Department\_ID (Foreign Key → Department) |
| Patient | Patient\_ID (Primary Key) |
| Name |
| Phone\_Number |
| Address |
| Birth\_Date |
| Job |
| Appointment | Appointment\_ID (Primary Key) |
| Appointment\_Date |
| Start\_Time |
| End\_Time |
| Cost |
| Status (Scheduled, In Progress, Postponed) |
| Diagnosis |
| Patient\_ID (Foreign Key → Patient) |
| Doctor\_ID (Foreign Key → Doctor) |

Step 1:

**Step 2: ER Diagram**



**Step 3&4: SQL Script**

**1. Creating the Database**

The script begins by creating a new database called "clinic\_management" using the `CREATE DATABASE` command. It then selects this database using `USE clinic\_management` to make sure all tables and data are created in it.

**2. Creating Tables**

Each `CREATE TABLE` statement defines a table for one entity in the system (Department, Clinic, Doctor, Patient, Appointment). Primary keys ensure each record is unique. Foreign keys link related tables together to ensure data consistency.

**3. Inserting Sample Data**

The "INSERT INTO" statements fill each table with sample data. At least 10 rows are included for each table to simulate real scenarios .

**4. Example Queries**

At the end of the script, three example queries are written to show how to retrieve meaningful information:  
- A query to get names of patients diagnosed with 'Fatty Liver' in the past year.  
- A query to list addresses of clinics under the 'Cardiology' department.  
- A query to calculate the total cost paid by a specific patient over the last 3 years.

**Step 5: Design Explanation**

**Overview**

The clinic management system is designed to facilitate the process of managing departments, clinics, doctors, patients, and scheduling appointments. The database allows storing medical history, ensuring that patient data and appointment records are consistent, traceable, and easy to retrieve.

**Entity Design Rationale**

• Department: Stores information about medical departments (e.g., Cardiology, Radiology).  
• Clinic: Represents the physical locations linked to a department.  
• Doctor: Assigned to a department and responsible for appointments.  
• Patient: Contains personal and contact details of patients.  
• Appointment: Connects a patient and a doctor, with timing, cost, and diagnosis.

**Relationships:**

• One-to-Many relationships are used:  
 - One department has many clinics.  
 - One department has many doctors.  
 - One doctor can have many appointments.  
 - One patient can have many appointments.

**Implementation Notes:**  
• Primary keys ensure each record is unique.  
• Foreign keys ensure consistent linkage between tables.  
• At least 10 test records were inserted per table for proper testing.