✅ Step-by-Step Guide

🎯 Objective

Build a complete **relational database** using **MySQL** for a real-world scenario of your choice.

💡 1. Choose a Use Case

Pick one real-world system. Here are some ideas:

- **Library Management System**
- **Student Records System**
- **Clinic Appointment Booking**
- **Inventory Management**
- **E-Commerce Product Catalog**

➡️ *Let's go with: **Clinic Appointment Booking System**.*

🗂️ 2. Plan Your Database

Design the structure with the following entities:

- **Patients**
- **Doctors**
- **Appointments**
- **Departments**
- **Users (Admins, Receptionists)**

⚙️ 3. Relationships

- One doctor belongs to one department (1:M)

- One appointment is between one patient and one doctor (M:M via appointments table)
- Each patient can have many appointments (1:M)

🧱 4. SQL File Structure (`clinic_system.sql`)

```
-- Create Department Table

CREATE TABLE Department (
    dept_id INT AUTO_INCREMENT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL UNIQUE
);

-- Create Doctor Table
CREATE TABLE Doctor (
    doctor_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);

-- Create Patient Table
CREATE TABLE Patient (
    patient_id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(100) NOT NULL,
    date_of_birth DATE,
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20)
);

-- Create Appointment Table
CREATE TABLE Appointment (
    appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    doctor_id INT,
    patient_id INT,
    appointment_date DATETIME NOT NULL,
    notes TEXT,
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
```

💠 ERD Overview: Clinic Appointment Booking System

🧱 Entities and Attributes

1. Department

- `dept_id` (PK)
- `dept_name` (UNIQUE, NOT NULL)

2. Doctor

- `doctor_id` (PK)
- `name` (NOT NULL)
- `dept_id` (FK → Department)

3. Patient

- `patient_id` (PK)
- `full_name` (NOT NULL)
- `date_of_birth`
- `email` (UNIQUE)
- `phone`

4. Appointment

- `appointment_id` (PK)
- `doctor_id` (FK → Doctor)
- `patient_id` (FK → Patient)
- `appointment_date` (NOT NULL)

- `notes`

🔗 Relationships

- A **Department** can have **many Doctors** (1:M)
- A **Doctor** can have **many Appointments** (1:M)
- A **Patient** can have **many Appointments** (1:M)
- Each **Appointment** links **one Doctor** and **one Patient**

🏥 Clinic Appointment Booking System – ERD

📘 Entities and Attributes

1. **Department**
    - `dept_id` (Primary Key)
    - `dept_name` (Unique, Not Null)
2. **Doctor**
    - `doctor_id` (Primary Key)
    - `name` (Not Null)
    - `dept_id` (Foreign Key → Department)
3. **Patient**
    - `patient_id` (Primary Key)
    - `full_name` (Not Null)
    - `date_of_birth`
    - `email` (Unique)

- `phone`

4. **Appointment**
    - `appointment_id` (Primary Key)
    - `doctor_id` (Foreign Key → Doctor)
    - `patient_id` (Foreign Key → Patient)
    - `appointment_date` (Not Null)
    - `notes`

🔗 Relationships

- A **Department** can have many **Doctors** (1:M)
- A **Doctor** can have many **Appointments** (1:M)
- A **Patient** can have many **Appointments** (1:M)
- Each **Appointment** is associated with one **Doctor** and one **Patient**

Complete SQL

```
-- clinic_system.sql
-- SQL script to create a Clinic Appointment Booking System database

-- Drop tables in correct dependency order
DROP TABLE IF EXISTS Appointment;
DROP TABLE IF EXISTS Doctor;
DROP TABLE IF EXISTS Patient;
DROP TABLE IF EXISTS Department;

-- Create Department table
CREATE TABLE Department (
    dept_id INT AUTO_INCREMENT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL UNIQUE
);

-- Create Doctor table
CREATE TABLE Doctor (
    doctor_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
```

```
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);

-- Create Patient table
CREATE TABLE Patient (
    patient_id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(100) NOT NULL,
    date_of_birth DATE,
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20)
);

-- Create Appointment table
CREATE TABLE Appointment (
    appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    doctor_id INT NOT NULL,
    patient_id INT NOT NULL,
    appointment_date DATETIME NOT NULL,
    notes TEXT,
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
```

DBML format

```
Table Department {
  dept_id int [pk, increment]
  dept_name varchar(100) [not null, unique]
}

Table Doctor {
  doctor_id int [pk, increment]
  name varchar(100) [not null]
  dept_id int [ref: > Department.dept_id]
}

Table Patient {
  patient_id int [pk, increment]
  full_name varchar(100) [not null]
  date_of_birth date
  email varchar(100) [unique]
  phone varchar(20)
}

Table Appointment {
  appointment_id int [pk, increment]
```

```
  doctor_id int [not null, ref: > Doctor.doctor_id]
  patient_id int [not null, ref: > Patient.patient_id]
  appointment_date datetime [not null]
  notes text
}
```

ERD diagram