Database Management System

Semester-III (Batch-2024)

Hospital Appointment Booking System

G11-DB-PID-4

Supervised By:
Mr. Sachin Garg

Submitted By:
Aastha Sharma (2410990812)
Abhay Dhasmana (2410990813)
Anjali Sharma (2410990840)
Anshu Sharma (2410990849)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

-

Hospital Appointment Booking System

1. Introduction

The Hospital Appointment Booking System is designed to automate and manage patient appointments in a healthcare facility. It replaces manual scheduling with a digital platform where patients can book, reschedule, or cancel appointments, while doctors and administrators can efficiently manage schedules.

In today's fast-paced healthcare environment, managing patient appointments manually leads to inefficiencies such as long waiting times, scheduling conflicts, and administrative burdens. A Hospital Appointment Booking System addresses these challenges by providing a digital, automated, and centralized solution for appointment management.

This Database Management System project demonstrates how structured data storage, retrieval, and manipulation can optimize healthcare workflows, benefiting patients, doctors, and hospital administrators.

2. Problem Overview

Hospital Appointment Management in many healthcare facilities is still handled through paper-based registers or basic spreadsheets, which leads to issues such as data redundancy, inconsistency, difficulty in retrieving information, and lack of security. Without a proper database management system, tracking doctor schedules, managing patient records, and avoiding appointment conflicts becomes inefficient and error-prone.

By implementing a centralized relational database, the system will improve accuracy, security, and accessibility of hospital appointment data while reducing redundancy and operational delays.

In the absence of a well-designed management system, the following challenges arise:

- Manual processing delays in retrieving and updating information.
- Data redundancy and inconsistency due to poorly structured records.
- Security risks due to unprotected or poorly managed records.
- Limited capabilities for reporting and analytics.

A well-structured Hospital Appointment Booking System eliminates these problems by providing a centralized, relational database with properly defined entities, relationships, and constraints.

-

3. Scope of the Project

The scope of this project is to design a relational database that will:

- Store patient details, medical history, and appointment records.
- Maintain doctor profiles, specialties, and availability schedules.
- Enable online booking, rescheduling, and cancellation.
- Organize hospital departments and assign doctors.
- Support patients, doctors, and administrators with role-based access.

4. Objectives

The primary objectives of the Hospital Appointment Booking System are:

a) **Centralized Patient Management** – Maintain a database of patient details (like medical history, appointment history, etc).
b) **Doctor Information Storage** – Store doctor details including specialization, availability, consultation hours.
c) **Appointment Scheduling** – Enable patients to book, update, or cancel appointments.
d) **Conflict Management** – Prevent double-booking for the same doctor and time slot.
e) **Billing & Payments** – Store consultation charges and generate bills for appointments.
f) **Prescription & Reports** – Keep track of doctor's prescriptions and test reports for patients.
g) **Efficient Data Retrieval** – Use queries for patient history, doctor schedules, daily appointments.
h) **Data Security** – Ensure sensitive patient data is stored securely and accessed only by authorized roles.

5. Significance of the Project

The proposed Hospital Appointment Booking System will significantly improve banking operations by:

- Optimizes scheduling to minimize delays and overcrowding
- Provides convenient online booking and reminders.
- Replaces error-prone paper-based systems with digital accuracy
- Generates reports for performance monitoring and decision making.
- Reduces operational costs associated with manual systems

-

- **Main Entities and Attributes:**

1. **Patient**
   a) Patient_ID (Primary Key)
   b) Name
   c) Age
   d) Gender
   e) Contact
   f) Address
   g) Medical_History (optional – can be a separate table if detailed)

2. **Doctor**

   a) Doctor_ID *(Primary Key)*
   b) Name
   c) Specialization
   d) Contact
   e) Fees
   f) Availability (time slots / schedule)

3. **Appointment**

   a) Appointment_ID *(Primary Key)*
   b) Patient_ID *(Foreign Key → Patient)*
   c) Doctor_ID *(Foreign Key → Doctor)*
   d) Date
   e) Time
   f) Status *(Booked / Completed / Cancelled)*

---

4. **Prescription**

   a) Prescription_ID *(Primary Key)*
   b) Appointment_ID *(Foreign Key → Appointment*
   c) Medicine
   d) Dosage
   e) Instructions

5. **Bill / Payment**

   a) Bill_ID *(Primary Key)*
   b) Appointment_ID *(Foreign Key → Appointment)*
   c) Amount
   d) Status *(Paid / Unpaid / Pending)*
   e) PaymentDate (optional)

-

6. **Medical History:**

a) History_ID
b) Patient_ID
c) Description
d) Date_Recorded

7. **Relationships:**

a) **Patient–Appointment**: One-to-Many
b) **Doctor–Appointment**: One-to-Many
c) **Patient-Doctor**: Many-to-Many
d) **Appointment–Bill**: One-to-One
e) **Doctor--Prescription**: One-to-Many (optional)
f) **Patient–Bill**: One-to-Many (via Appointment)
g) **Patient–Prescription:** One-to-Many

•

- **ER Model and Diagram**

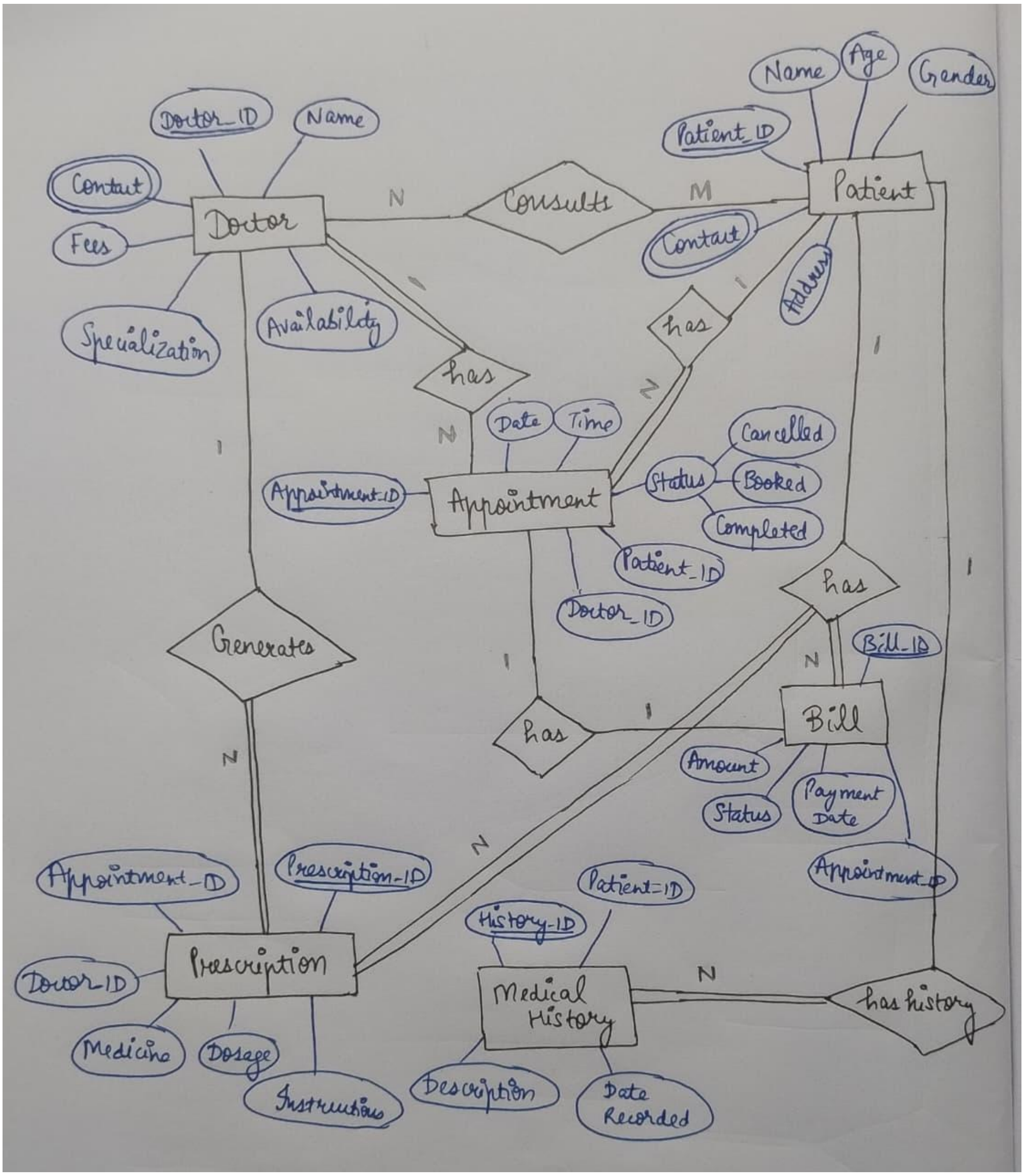

Fig.1: ER Diagram of Hospital Appointment Booking System

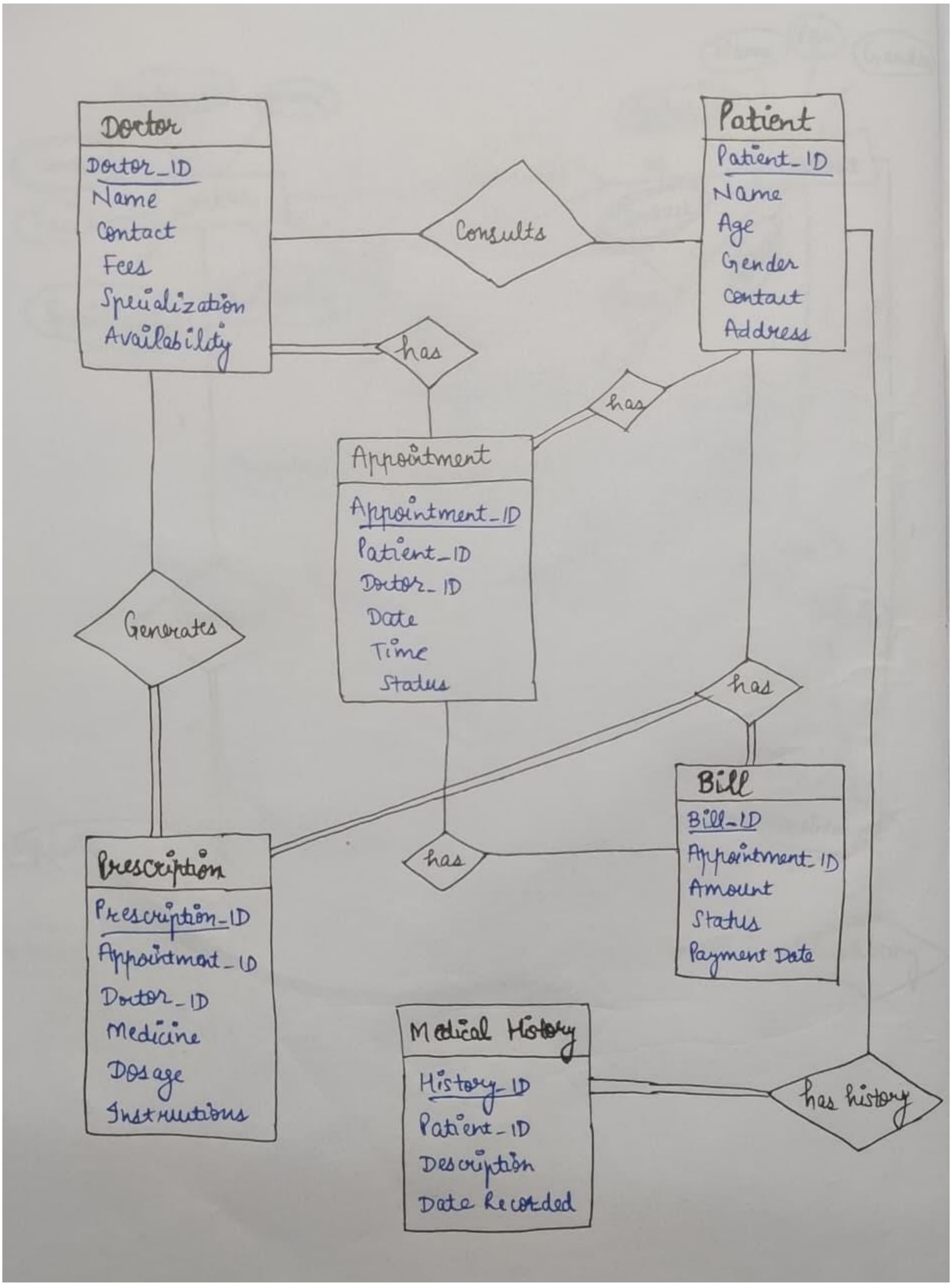

Fig.2: ER Model of Hospital Appointment Booking System

ER to Relational Model (Schema)

- Patient (Patient_ID [PK] , Name, Age, Gender, Contact, Address)
- Doctor (Doctor_ID [PK], Name, Specialization, Contact, Fees, Availability)
- Appointment (Appointment_ID [PK], Patient_ID [FK], DoctorID [FK], Date, Time, Status)
- Prescription (Prescription_ID [PK], Appointment_ID [FK], Doctor_ID [FK], Medicine, Dosage, Instructions)
- Bill (Bill_ID [PK], Appointment_ID[FK], Amount, Date, Status)
- Medical History (History_ID [PK], Patient_ID [FK] , Description, Date_Recorded)

Relational Algebra Queries :

Q1. List all doctor's names and specializations

π Name, Specialization (Doctor)

Q2. Find doctors who have no appointments.

π Doctor_ID (Doctor) – π Doctor_ID (Appointment)

Q3. Find details of all doctors who charge fees greater than 500.

σFees>500(Doctor)

Q4. Get a doctor's table with clearer headings (rename Name → DoctorName, Contact → Phone).

ρDoctorName/Name,Phone/Contact (Doctor)

Q5. Find all IDs of patients and doctors (just IDs in one list).

πPatient_ID (Patient) ∪ πDoctor_ID (Doctor)

Q6. Get prescriptions for appointment 301

π Medicine, Dosage, Instructions (σ Appointment_ID = 301 (Prescription))

Q7. Find patients who booked appointments with all doctors.

πPatient_ID (Appointment) ÷ π Doctor_ID (Doctor)

Q8. List all doctors who are available in the morning

π Name, Specialization (σ Availability = 'Morning' (Doctor))

Q9. Find doctor with highest fees

Doctor ⋈ σ Fees = MAX(Doctor.Fees) (Doctor)

Q10. Count number of doctors in each department

γ Department_ID; COUNT(DoctorID) → DoctorCount (Doctor)

Q11. Find patients who had both Cancelled and Completed appointments

πPatient_ID (σStatus=’Cancelled’ (Appointment))∩πPatient_ID (σStatus=’Completed’ (Appointment))

Q12. Find patients who have not booked any appointment

π Patient_ID, Name (Patient) − π Patient_ID, Name (Patient ⋈ Appointment)

Q13. Find patient–doctor pairs on the same day.

σAppointment.Date=’2025−08−26’ (Patient×Doctor×Appointment)

Q14. Show doctors and the dates when they prescribed medicines.

πDoctor.Name,Appointment.Date (Doctor ⋈Appointment ⋈Prescription)

Q15. Find total revenue collected (sum of all bills)

γ SUM(Amount) → TotalRevenue (Bill)

Q16. Show appointment IDs with their bill amounts.

πAppointment.Appointment_ID,Bill.Amount(Appointment ⋈Bill)

Q17. Find total number of appointments per doctor.

γ Doctor_ID;COUNT(Appointment_ID)(Appointment)

Q18. Find patients treated by ‘Dr. Sharma’

π Patient.Name ((Appointment ⋈ Doctor) ⋈ Patient) σ Doctor.Name = 'Dr. Sharma'

Q19. Find patients who have both medical history records and at least one appointment.

πPatient_ID (MedicalHistory) ∩ πPatient_ID (Appointment)

Q20. Find prescriptions containing 'Aspirin'

π Appointment_ID, Medicine, Dosage
(σ Medicine = 'Aspirin' (Prescription))

Q21.Show the names of patients and their appointment dates.

πPatient.Name,Appointment.Date(Patient ⋈Appointment)

Q22. Find patients who do not have medical history records

π Patient_ID, Name (Patient) − π Patient_ID, Name (Patient ⋈ MedicalHistory)

Q23. Find appointments that are booked but not yet completed.

πAppointment_ID(σStatus=’Booked’(Appointment))−πAppointment_ID(σStatus=’Completed’(Appointment))

Q24. Find prescriptions where the dosage is "2 times a day".

σDosage='2timesaday'(Prescription)

Q25. Find doctors with fees greater than 600

π Name, Specialization, Fees (σ Fees > 600 (Doctor))