**Hospital Management System Database**

**Report 1**

**Project by: Mohammad Hassan**

**Contents**

[**Introduction** 3](#_Toc198762416)

[**Problem Addressed** 3](#_Toc198762417)

[**Scope** 3](#_Toc198762418)

[**Project at College/University Level** 4](#_Toc198762419)

[**Entities and Attributes** 4](#_Toc198762420)

[**Normalization** 6](#_Toc198762421)

[**First Normal Form (1NF):** 6](#_Toc198762422)

[**Second Normal Form (2NF):** 6](#_Toc198762423)

[**Third Normal Form (3NF):** 7](#_Toc198762424)

[**Relations** 7](#_Toc198762425)

[**Entity-Relation Diagrams (ERD)** 8](#_Toc198762426)

[**Crow Foot Diagram:** 8](#_Toc198762427)

[**Chen Model Diagram** 9](#_Toc198762428)

[**Conclusion** 9](#_Toc198762429)

# **Introduction**

In these modern days, to handle enormous volumes of data concerning patients, employees, medical procedures, billing, and logistics, the healthcare industry need reliable systems. These tasks are automated and streamlined by a hospital management system (HMS) to increase productivity, lower errors, and improve patient care. Using a relational database technique, our effort seeks to model such a system that replicates actual hospital processes. It incorporates core functionalities such as appointment scheduling, billing, admissions, medical records, diagnosis, and staff management.

# **Problem Addressed**

Hospital management done manually (file based system) is very ineffective and prone to mistakes. Without a centralized system, it is challenging to retain and retrieve crucial information concerning staff scheduling, patient history, appointments, medical records, billing, and room availability. The main driving force for the creation of this hospital management system was to solve these issues by building a database that:

* Captures and organizes essential data accurately.
* Provides relationships between entities such as patients, doctors, treatments, and diagnoses.
* Reflects realistic operations found in actual hospitals.
* Supports data integrity and normalization for efficient queries and reduced redundancy.

This project aims to simulate a simplified but practical hospital system architecture that could be expanded into a full software solution in the future.

# **Scope**

This project focuses on organizing and storing hospital data using a database. It includes the following:

* Managing patient details, appointments, and admissions
* Storing diagnoses, prescriptions, treatments, and lab tests
* Keeping records of doctors, nurses, and receptionists
* Organizing departments and assigning staff
* Handling bills, payments, and room charges
* Saving patient medical history and doctor notes

# **Project at College/University Level**

While we are still studying the basics of database systems, this project was created as part of our university-level coursework. We have done our best to make sure that the project demonstrates a firm grasp of important database principles, such as normalization, relationships, and real-world data modeling, even if we are still in the learning phase. Numerous crucial facets of hospital administration are covered by the system, which could eventually be developed into a comprehensive and useful software program for practical usage.

# **Entities and Attributes**

We have come up with a total of 15 entities – meaning 15 different tables – with their appropriate attributes. Below is the summary of the entities and the attributes that have been used in our database, with the reasons of their inclusion:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Patient**

**Attributes**: ***Patient\_ID***, PatientName, Age, Gender, Address, Contact\_Number, DOB, Blood\_Type

**Purpose**: To store essential personal and medical information of patients for identification, communication, and treatment tracking.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Receptionist**

**Attributes**: ***Receptionist\_ID***, R\_Name, Number, Email

**Purpose**: To record details of reception staff responsible for scheduling appointments and managing front-desk operations.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appointment**

**Attributes**: ***Appointment\_ID***, Patient\_ID, Doctor\_ID, Receptionist\_ID, Date, Time, Status

**Purpose**: To manage and track patient appointments with doctors, coordinated by reception staff.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Doctor**

**Attributes**: ***Doctor\_ID***, Doctor\_Name, Specialization, Contact, Email, Lisence\_Number

**Purpose**: To maintain professional and contact details of doctors along with their specialization.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Diagnosis**

**Attributes**: ***Diagnosis\_ID***, Appointment\_ID, Patient\_ID, Doctor\_ID, Description, Date

**Purpose**: To log medical conditions diagnosed by doctors during patient appointments.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lab**

**Attributes**: ***Lab\_ID***, Diagnosis\_ID, Test\_Name, Description, Lab\_Cost

**Purpose**: To store laboratory tests associated with diagnoses, including their names and costs.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prescription**

**Attributes**: ***Prescription\_ID***, Diagnosis\_ID, Medicine\_Name, Dosage, Duration

**Purpose**: To maintain records of medicines prescribed to patients, including dosage and treatment period.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Treatment**

**Attributes**: ***Treatment\_ID***, Diagnosis\_ID, Treatment\_Type, Description, Treatment\_Cost

**Purpose**: To record medical or surgical treatments administered to patients with associated costs.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Admission**

**Attributes**: ***Admission\_ID***, Patient\_ID, Diagnosis\_ID, Room\_ID, Admission\_Date, Discharge\_Date, Room\_Cost

**Purpose**: To document patient hospital stays, assigned rooms, and related charges.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Room**

**Attributes**: ***Room\_ID***, Room\_Number, Type, Status

**Purpose**: To track hospital room numbers, types (e.g., ICU, General), and occupancy status.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nurse**

**Attributes**: ***Nurse\_ID***, Nurse\_Name, Contact\_Number, Shift\_Timings, Room\_ID

**Purpose**: To store details of nurses and their room assignments and shift schedules.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Medical History**

**Attributes**: ***History\_ID***, Patient\_ID, Doctor\_ID, Description, Date\_Recorded

***Purpose:*** To log a patient’s previous health conditions and treatments as recorded ***by*** doctors.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Bill**

**Attributes**: ***Bill\_ID***, Patient\_ID, Appointment\_ID, Receptionist\_ID, Amount, Bill\_Date, Bill\_Status

**Purpose**: To record billing information related to appointments, including payment tracking and responsible receptionist.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Employee**

**Attributes**: ***Employee\_ID***, Name, Role, Contact\_Number, Department\_ID, Joining\_Date

**Purpose**: To manage records of all hospital employees including doctors, nurses, and administrative staff.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Department**

**Attributes**: ***Department\_ID***, Department\_Name

**Purpose**: To organize hospital staff and services into structured departments.

# **Normalization**

**In database design, Normalization is a procedure that arranges tables and their relationships in a systematic manner to reduce data redundancy and guarantee data integrity. Using foreign keys to identify relationships between the larger, more complex tables and breaking them up into smaller, simpler ones is the aim.**

## **First Normal Form (1NF):**

To ensure that all tables are in **First Normal Form (1NF)**, we must eliminate any multivalued attributes. All tables in the system meet the criteria for 1NF, except for **Prescription** and **Lab**.

* The **Prescription** table contains the attribute *MedicineName*, which can hold multiple values since a single prescription may include several medicines. To resolve this, we have created a separate table called **PrescriptionMedicine** to store each medicine associated with a prescription individually.
* Similarly, the **Lab** table includes the attribute *Test\_Name*, which is also multivalued because a patient can undergo multiple tests. To normalize this, we split the Lab table into two: **Lab** and **Test**, where the Lab table holds general lab request information, and the Test table stores individual test details linked to the Lab.

**Prescription**

**Attributes**: PrescriptionID, DiagnosisID

**PrescriptionMedicine**

**Attributes**: PrescriptionMedicineID, PrescriptionID, MedicineName, Dosage, Instructions, Duration

**Lab**

**Attributes**: Lab\_ID, DiagnosisID

**Test**

**Attributes**: Test\_ID, Lab\_ID, Test\_Name, Description, Test\_Cost

Now after splitting these into two tables, all the tables are now in 1NF. Additionally, now there are a total of 17 entities – means 17 tables.

## **Second Normal Form (2NF):**

To ensure the tables are in 2NF all non-key attributes must be fully dependent on the entire primary key. All the tables in our database are already in 2NF as there is no partial dependency in any table.

## **Third Normal Form (3NF):**

To ensure the tables are in 3NF there should be no Transitive dependency, which means no non-key attributes must depend on other non-key attributes. All the tables in our database are already in 3NF as there is no transitive dependency in any table.

# **Relations**

**Patient → Appointment: 1:M** One Patient can have many Appointments

**Doctor → Appointment: 1:M** One Doctor can attend many Appointments

**Receptionist → Appointment: 1:M** One Receptionist can create many Appointments

**Appointment → Diagnosis: 1:M** One Appointment can lead to many Diagnosis

**Patient → Diagnosis: 1:M** One Patient can have many Diagnosis

**Doctor → Diagnosis: 1:M** One Doctor can perform many Dignosis

**Diagnosis → Lab: 1:1** One Diagnosis can have one Lab

**Lab → Test: 1:M** One Lab can have many Tests

**Diagnosis → Prescription: 1:1** One Diagnosis will cause One Prescription

**Prescription → PrescriptionMedicine: 1:M** One Precription will have many Medicines

**Diagnosis → Treatment: 1:M** One Diagnosis can have many Treatments

**Patient → Admission: 1:M** One Patient can have many Admissions

**Diagnosis → Admission: 1:1** One Diagnosis can have one Admission

**Room → Admission: M:M** Many Rooms can have many Admissions

**Room → Nurse: 1:M** One Room will have many Nurses

**Patient → Medical\_History: 1:M** One Patient can have many Records

**Doctor → Medical\_History: M:M** Many Doctors can person many Record updating

**Bill → Patient: M:1** Many Bills can belong to one Patient

**Appointment → Bill: 1:M** One Appointment can lead to many Bills

**Bill → Receptionist: M:1** Many Bills can be generated by one Receptionist

**Employee → Department: M:1** Many Employee will belong to 1 Department

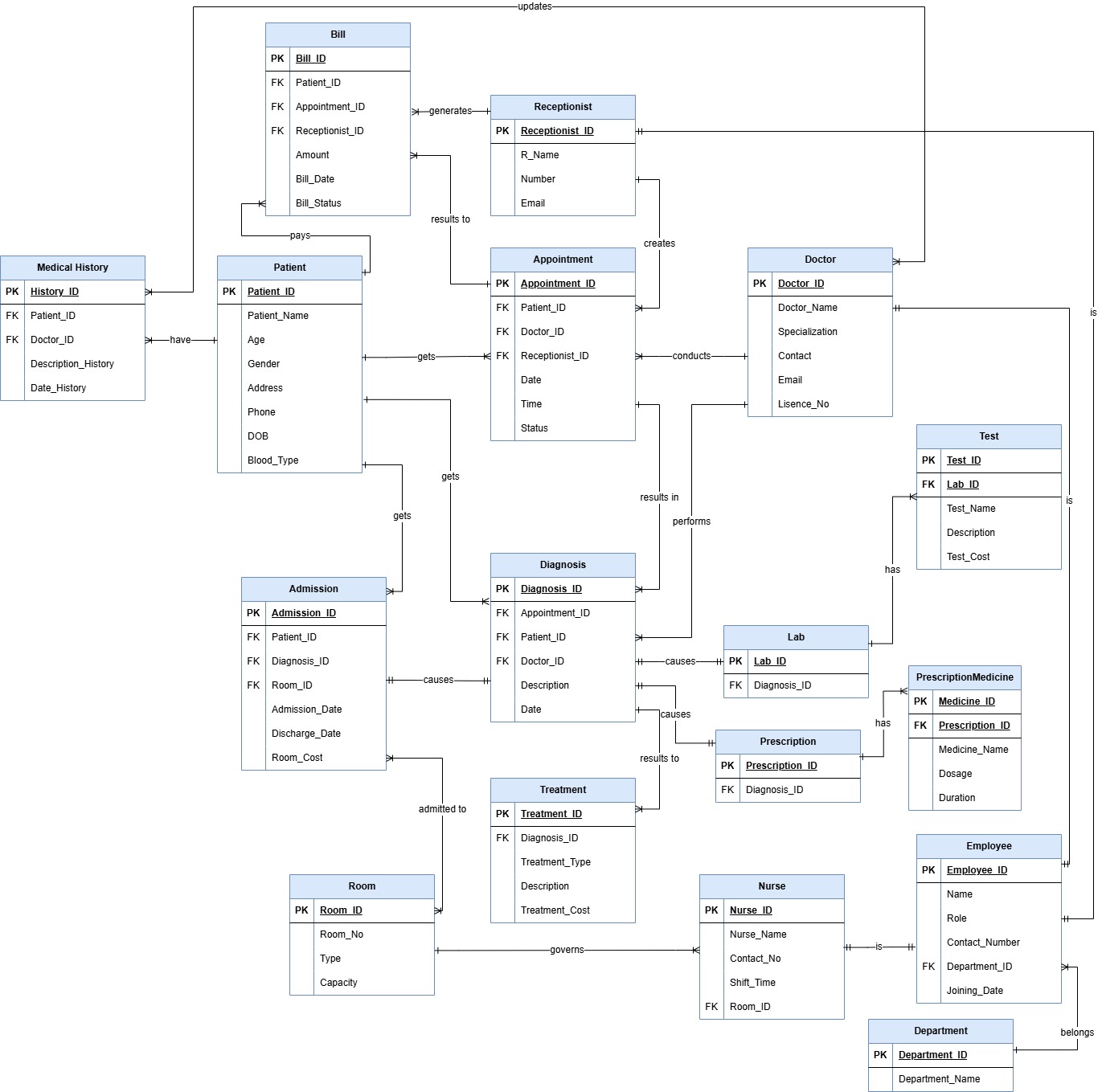
**Doctor → Employee: 1:1** Each Doctor will be an Employee

**Nurse → Employee: 1:1** Each Nurse will be an Employee

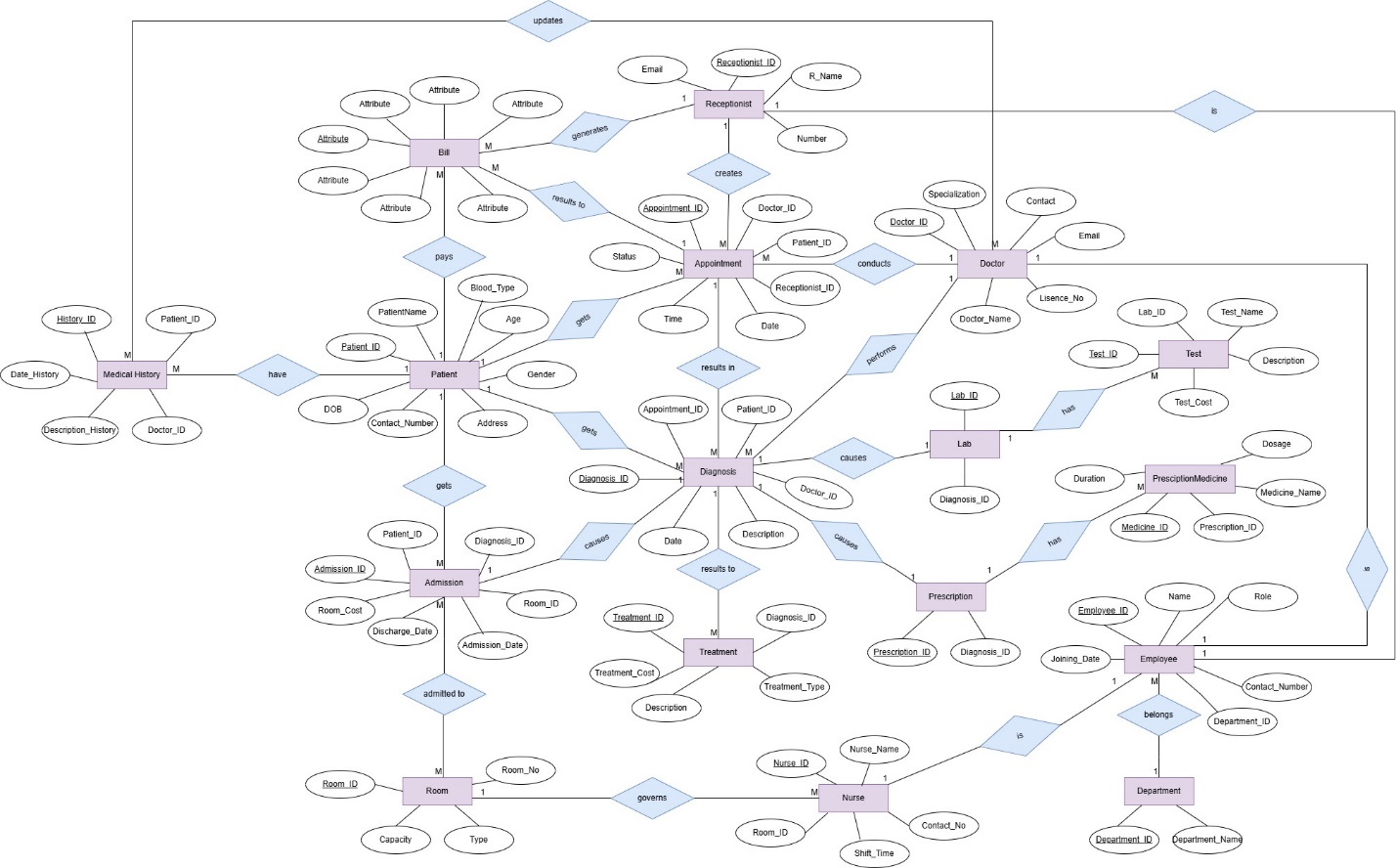
**Receptionist → Employee: 1:1** Each Receptionist will be an Employee

# **Entity-Relation Diagrams (ERD)**

## **Crow Foot Diagram:**



## **Chen Model Diagram**



# **Conclusion**

The Hospital Management System project demonstrates how database systems can effectively simulate real-life hospital environments. By identifying core entities and their relationships, normalizing data structures, and maintaining clear linkages between components, the system lays the groundwork for scalable, efficient, and maintainable hospital data operations. This project not only showcases key database design principles but also provides insight into how technology can address real-world healthcare challenges through well-structured information systems.