WELCOME TO OUR DB PROJECT

DATABASE DESIGN DOCUMENTATION

FOR

Hospital database

DEC.2024

Table of Contents :-

1.Introduction

Purpose
The purpose of this document is to provide detailed documentation of the Hospital Management System database design. This includes the tables, relationships, primary and foreign keys, and constraints that ensure data integrity.

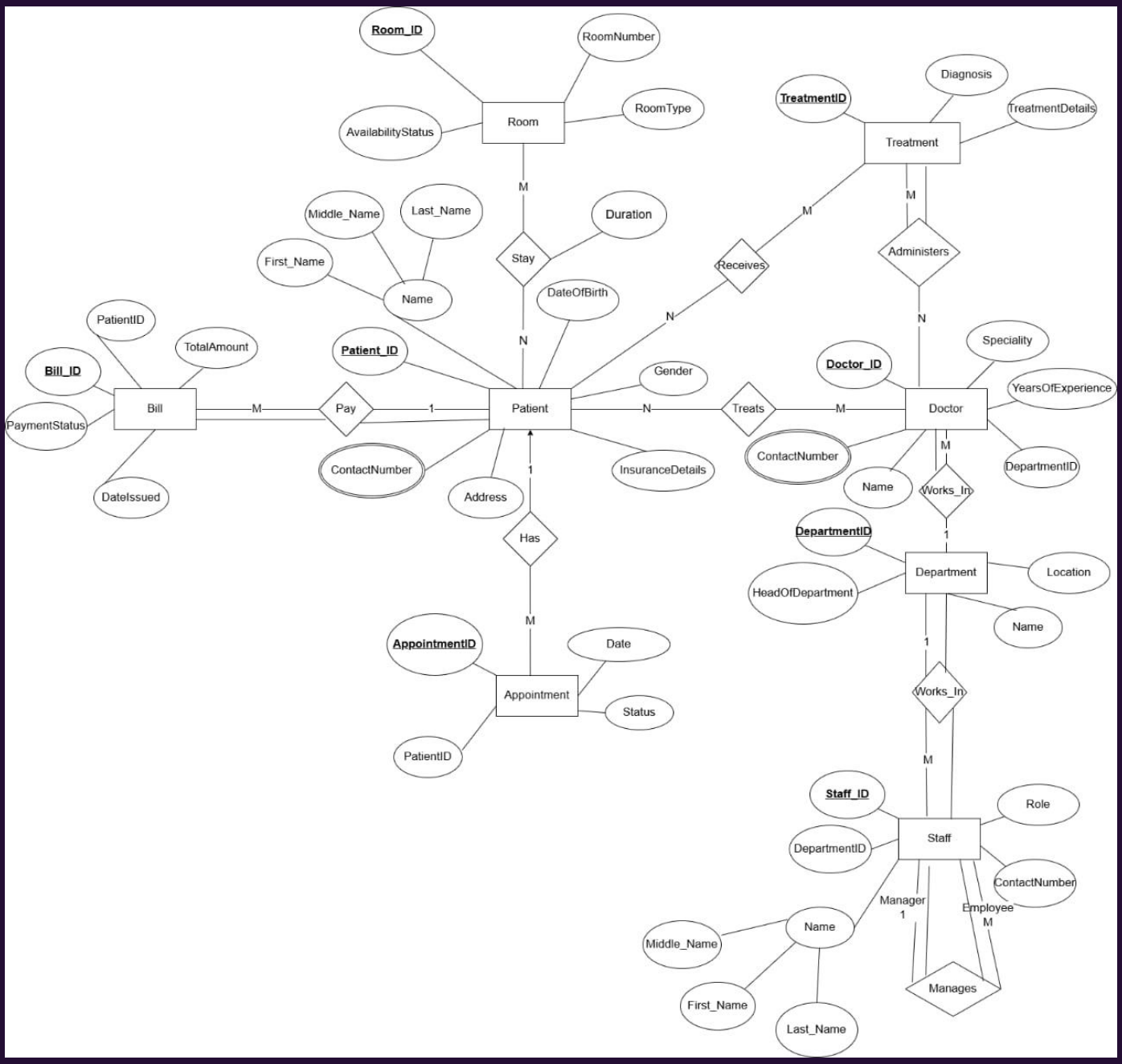
Scope
This database design supports a hospital management system that manages patients, doctors, staff, rooms, treatments, bills, and appointments.

Audience
This document is intended for developers, database administrators, and system analysts involved in the development and maintenance of the Hospital Management System.

Team Members
- **Mohamed Samir**
- **Mohaymen Hany**
- **Yousef Sherif**
- **Mostafa Ashraf**
- **Mohamed Salama**

2.ERD Overview

The following is the Entity-Relationship Diagram (ERD) representing hospital Database system



3.Data Requirements

Key Components:-

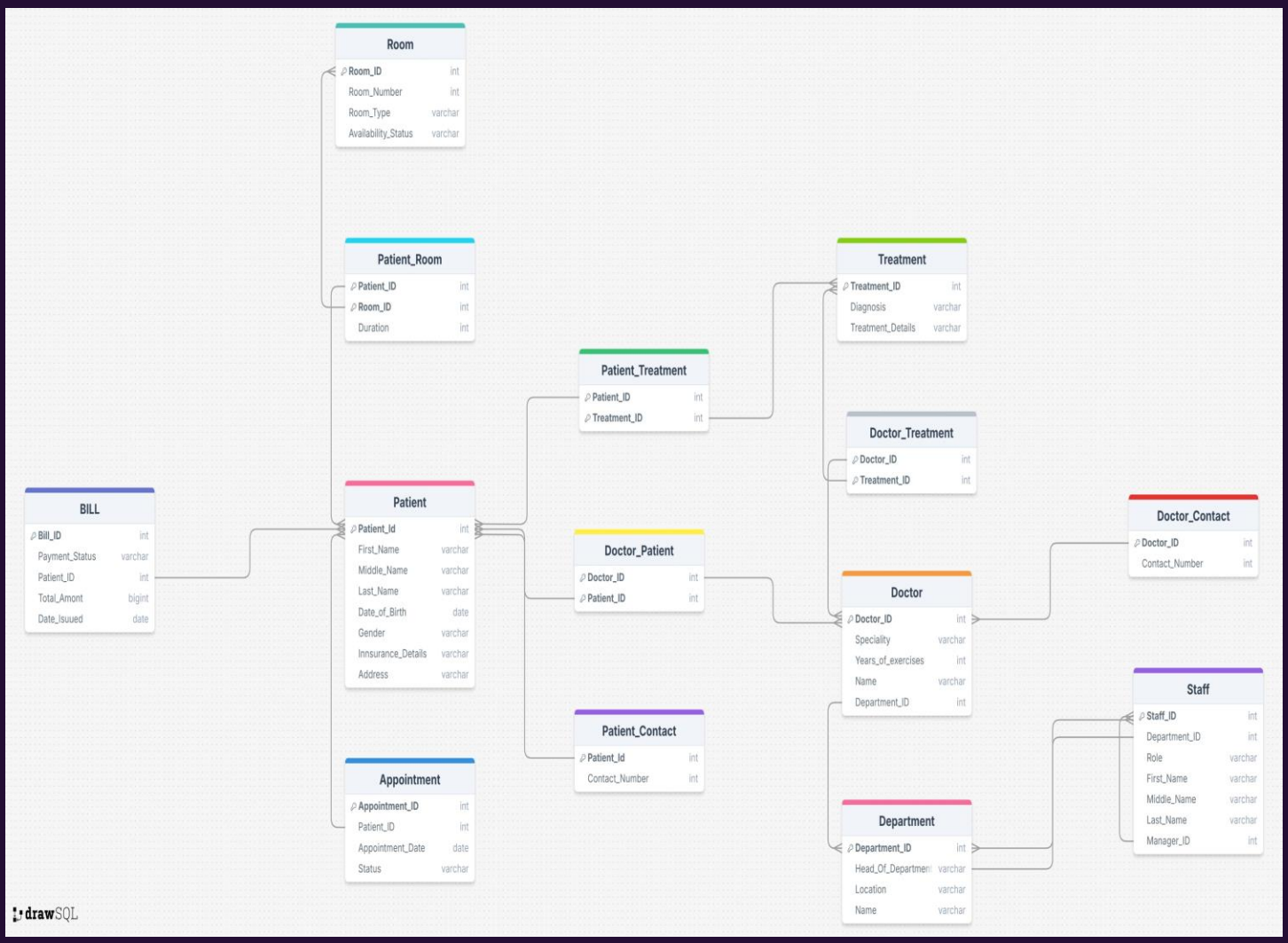
- **Entities**: Patient, Doctor, Staff, Department, Room, Treatment, Bill, Appointment.
- **Relationships**:
  - A patient pay multiple bills, has appointments, stays in room, and get treated by doctor.
  - A doctor treats patients, administrates treatments and works in a department.
  - Staff members work in departments and may manage other employees.
  - Rooms are assigned to patients for their stay.
  - Treatments are administered to patients by doctors.
  - Bills get paid by patients
  - Appointments are assigned to patients for their time.

4.Normalized Schema Design

After applying mapping & normalization concepts, we made new tables which their attributes come from two entities.

-The new tables are:

- Patient_Room
- Patient_Treatment
- Patient_Contact
- Doctor_Patient
- Doctor_Treatment
- Doctor_Contact



5.Table Attributes Definitions

a)Ordinary Tables:

1. PATIENT

Attributes	Data Type	Constraints	Description
Patient_ID	INT	PRIMARY KEY	Unique identifier for the patient
First_Name	VARCHAR(100)	NOT NULL	First name of the patient
Middle_Name	VARCHAR(100)	NOT NULL	Middle name of the patient
Last_Name	VARCHAR(100)	NOT NULL	Last name of the patient
DateOfBirth	DATE	NULL	Date of birth
Gender	VARCHAR(10)	CHECK (Gender IN ('M','F','Other'))	Gender of the patient
Address	VARCHAR(100)	NULL	Residential address
InsuranceDetails	VARCHAR(150)	NULL	Insurance information

2. TREATMENT

Attributes	Data Type	Constraints	Description
TreatmentID	INT	PRIMARY KEY	Unique identifier for treatment
Diagnosis	VARCHAR(200)	NULL	Diagnosis description
TreatmentDetails	TEXT	NULL	Detailed treatment notes

3. DOCTOR

Attributes	Data Type	Constraints	Description
Doctor_ID	INT	PRIMARY KEY	Unique identifier for the doctor
Name	VARCHAR(100)	NOT NULL	Full name of the doctor
Specialty	VARCHAR(30)	NOT NULL	Doctor's specialty
YearsOfExperience	INT	CHECK (YearsOf Experience >= 0)	Years of experience
ContactNumber	VARCHAR(15)	NULL	Phone number
DepartmentID	INT	FOREIGN KEY REFERENCES Department	Links to the Department table

4. DEPARTMENT

Attributes	Data Type	Constraints	Description
DepartmentID	INT	PRIMARY KEY	Unique identifier for the department
Name	VARCHAR(100)	NOT NULL	Department name
Location	VARCHAR(50)	NULL	Physical location of the department
HeadOfDepartment	INT	FOREIGN KEY REFERENCES Staff	References Staff table (manager)

5. Staff

Attributes	Data Type	Constraints	Description
Staff_ID	INT	PRIMARY KEY	Unique identifier for staff
First_Name	VARCHAR(100)	NOT NULL	Staff first name
Middle_Name	VARCHAR(100)	NOT NULL	Staff middle name
Last_Name	VARCHAR(100)	NOT NULL	Staff last name
Role	VARCHAR(50)	NOT NULL	Job role
DepartmentID	INT	FOREIGN KEY REFERENCES Department	Works in which department
ManagerID	INT	FOREIGN KEY REFERENCES Staff (Self-Ref)	Reports to which manager

6. Appointment

Attributes	Data Type	Constraints	Description
AppointmentID	INT	PRIMARY KEY	Unique identifier for the appointment
PatientID	INT	FOREIGN KEY REFERENCES Patient	Links to Patient table
AppointmentDate	DATE	NOT NULL	Date of the appointment
Status	VARCHAR (50)	CHECK (Status IN ('Scheduled', 'Completed', 'Cancelled'))	Appointment status

7. ROOM

Attributes	Data Type	Constraints	Description
Room_ID	INT	PRIMARY KEY	Unique identifier for the room
RoomNumber	VARCHAR(10)	NOT NULL	Room number
RoomType	VARCHAR(50)	NULL	Type of the room
AvailabilityStatus	VARCHAR(20)	CHECK (AvailabilityStatus IN ('Available','Occupied'))	Room status

8. BILL

Attributes	Data Type	Constraints	Description
Bill_ID	INT	PRIMARY KEY	Unique identifier for the bill
PatientID	INT	FOREIGN KEY REFERENCES Patient	Links to Patient table
TotalAmount	DECIMAL(10,2)	NOT NULL	Total bill amount
PaymentStatus	VARCHAR(20)	CHECK (PaymentStatus IN ('Paid','Unpaid'))	Payment status
DateIssued	DATE	NOT NULL	Bill issue date

b)Relationship Tables:

9.PATIENT_CONTACT

Attributes	Data Type	Constraints	Description
Patient_ID	int	FOREIGN KEY references Patient(Patient_ID)	Unique identifier for the patient
Contact_Number	int	NOT NULL	Stores the patient's contact number

10. PATIENT_ROOM

Attributes	Data Type	Constraints	Description
Patient_ID	int	FOREIGN KEY references Patient(Patient_ID)_ID)	Unique identifier for the patient
Room_ID	int	FOREIGN KEY references Room(Room_ID)	Unique identifier for the Room
Duration	int	NOT NULL	Stores the Duration of staying the Patient at the Room

11. DOCTOR_PATIENT

Attributes	Data Type	Constraints	Description
Doctor_ID	int	FOREIGN KEY references Doctor(Doctor_ID)	Unique identifier for the doctor
Patient_ID	int	FOREIGN KEY references Patient(Patient_ID)	Unique identifier for the patient

12. DOCTOR_CONTACT

Attributes	Data Type	Constraints	Description
Doctor_ID	int	FOREIGN KEY references Doctor(Doctor_ID)	Unique identifier for the doctor
Contact_Number	int	NOT NULL)	Stores the doctor's contact number

13. PATIENT_TREATMENT

Attributes	Data Type	Constraints	Description
Patient_ID	int	FOREIGN KEY references Patient(Patient_ID)	Unique identifier for the patient
Treatment_ID	int	FOREIGN KEY references Treatment(Treatment_ID)	Unique identifier for treatment

14. DOCTOR_TREATMENT

Attributes	Data Type	Constraints	Description
Doctor_ID	int	FOREIGN KEY references Doctor(Doctor_ID)	Unique identifier for the doctor
Treatment_ID	int	FOREIGN KEY references Treatment(Treatment_ID)	Unique identifier for treatment

6.Detailed Relationships

Relationship	Description
Patient – Bill	One-to-Many: A patient may pay multiple Bills
Patient – Appointment	One-to-Many: A patient can have multiple Appointments
Patient - Doctor	Many-to-Many: Many patients can be treated by many Doctors
Patient - Treatment	Many-to-Many: Many patients can receive many Treatments
Doctor – Treatment	Many -to-Many: Many Doctors can administrate multiple Treatments
Staff – Department	Many-to-One: Many Staff work in a Department
Doctor – Department	Many-to-One: Many Doctors work in one Department
Staff (Self-Referencing)	One-to-Many: A Staff manager manages many other Staff
Room – Patient	Many -to-Many: Many Patients can stay in many Rooms

---

THANKS FOR READING