CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

CASE STUDY REPORT
ON
HOSPITAL MANAGEMENT SYSTEM

Program Name: BCA

Subject Name/Code: Database Management System (23CAT-251)

Submitted by:

Name: Saket Singh

UID: 23BCA10351

Section: 4(B)

Submitted to:

Name: Arvinder Singh

Designation: Assist. Prof.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Table of Contents

https://github.com/saket73/hospital.git

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

ABSTRACT

- **Introduction:**

  This project simulates a Hospital Management System using SQL. It is designed to demonstrate the basic functionalities of managing hospital data such as patients, doctors, appointments, medications, and prescriptions.

- **Technique:**

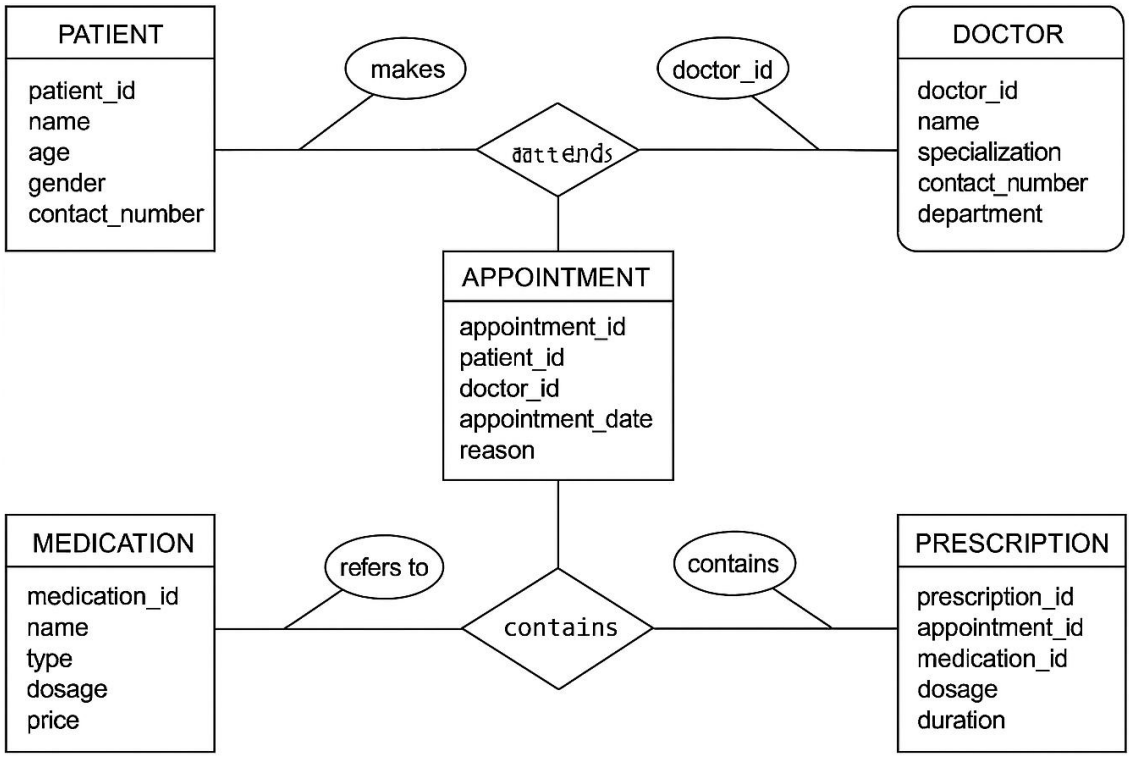
  The project uses MySQL as the relational database system. SQL commands are used for table creation, data insertion, constraints application, and query operations.

- **System Configuration:**
  - **Database**: MySQL 8.0
  - **Interface**: MySQL Workbench
  - **Operating System**: Windows 10 or higher
  - **RAM**: Minimum 4 GB
  - **Storage**: Minimum 1 GB free space for DBMS and project files
  - **Diagram Tool**: Draw.io (for ER diagrams)

- **INPUT:**

  The input consists of predefined data entries for five tables: Patients, Doctors, Appointments, Medications, and Prescriptions. Each table contains 10 sample records.

- **ER DIAGRAM:**



- **ER DIAGRAM DESCRIPTION**

The ER Diagram consists of the following entities: The ER diagram represents the relationship between different entities: Patients, Doctors, Appointments, Medications, and Prescriptions. Each appointment connects a patient and a doctor, while prescriptions link appointments to medications.

Relationships:

This section defines how different tables relate to each other using primary and foreign keys. These relationships ensure data consistency across the database.

- **TABLE RELATIONSHIPS:**
- **One-to-Many** between **Doctors** and **Appointments**
- **One-to-Many** between **Patients** and **Appointments**
- **One-to-One** between **Appointments** and **Prescriptions**
- **Many-to-One** from **Prescriptions** to **Medications**

- **TABLE CREATION**

PATIENT TABLE

```
4  CREATE TABLE Patients (
5      patient_id INT PRIMARY KEY,
6      name VARCHAR(50),
7      age INT,
8      gender VARCHAR(10),
9      contact_number VARCHAR(15)
10 );
```

DOCTOR TABLE

```
24 CREATE TABLE Doctors (
25     doctor_id INT PRIMARY KEY,
26     name VARCHAR(50),
27     specialization VARCHAR(30),
28     contact_number VARCHAR(15),
29     department VARCHAR(30)
30 );
```

APPOINTMENT TABLE

```sql
CREATE TABLE Appointments (
    appointment_id INT PRIMARY KEY,
    patient_id INT,
    doctor_id INT,
    appointment_date DATE,
    reason VARCHAR(100),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

MEDICATIONS TABLE

```sql
CREATE TABLE Medications (
    medication_id INT PRIMARY KEY,
    name VARCHAR(50),
    type VARCHAR(30),
    dosage VARCHAR(20),
    price DECIMAL(10,2)
);
```

PRESCRIPTIONS TABLE

```sql
CREATE TABLE Prescriptions (
    prescription_id INT PRIMARY KEY,
    appointment_id INT,
    medication_id INT,
    dosage VARCHAR(20),
    duration VARCHAR(20),
    FOREIGN KEY (appointment_id) REFERENCES Appointments(appointment_id),
    FOREIGN KEY (medication_id) REFERENCES Medications(medication_id)
);
```

- ## TABLE REALTION:

In the Hospital Management System, the relationships between tables are crucial for maintaining data consistency and correct entity mappings. Below are the primary relationships:

1. **Patient to Appointment:**
   - One-to-Many (A patient can have multiple appointments, but each appointment is related to only one patient).
2. **Doctor to Appointment:**
   - One-to-Many (A doctor can have multiple appointments, but each appointment is related to only one doctor).
3. **Appointment to Prescription:**
   - One-to-One (Each appointment can have only one prescription, and each prescription is associated with only one appointment).
4. **Prescription to Medication:**
   - Many-to-Many (A prescription can have multiple medications, and a medication can be prescribed in multiple prescriptions).
5. **Patient to Prescription:**
   - One-to-Many (A patient can have multiple prescriptions, but each prescription belongs to one patient).

- ## TABULAR FORMAT:

Table Name	Description	Relationship
Patients	Stores personal details of the patients	-
Doctors	Stores details about the doctors (ID, specialization)	-
Appointments	Stores details about appointments (date, reason)	One-to-Many (Patient → Appointment)
Medications	Stores medication details (name, type, dosage, price)	-

Table Name	Description	Relationship
Prescriptions	Stores details of prescriptions given to patients	One-to-Many (Appointment → Prescription)
Patient_Medication	Mapping table for prescriptions and medications	Many-to-Many (Prescription ↔ Medication)

- **SQL IMPLEMENTATION Code:**

```
CREATE DATABASE Hospital_Management;
USE Hospital_Management;

CREATE TABLE Patients (
    patient_id INT PRIMARY KEY,
    name VARCHAR (50),
    age INT,
    gender VARCHAR (10),
    contact_number VARCHAR (15)
);

INSERT INTO Patients VALUES
(1, 'Amit Sharma', 30, 'Male', '9876543210'),
(2, 'Sonal Gupta', 25, 'Female', '9876543211'),
(3, 'Rahul Mehta', 40, 'Male', '9876543212'),
(4, 'Priya Desai', 35, 'Female', '9876543213'),
```

```
(5, 'Anil Kumar', 50, 'Male', '9876543214'),
(6, 'Meena Rathi', 45, 'Female', '9876543215'),
(7, 'Rohit Jain', 28, 'Male', '9876543216'),
(8, 'Kavita Yadav', 32, 'Female', '9876543217'),
(9, 'Vikram Singh', 38, 'Male', '9876543218'),
(10, 'Pooja Sinha', 29, 'Female', '9876543219');

CREATE TABLE Doctors (
    doctor_id INT PRIMARY KEY,
    name VARCHAR (50),
    specialization VARCHAR (30),
    contact_number VARCHAR (15),
    department VARCHAR (30)
);

INSERT INTO Doctors VALUES
(1, 'Dr. Sharma', 'Cardiology', '9990001111', 'Heart'),
(2, 'Dr. Joshi', 'Neurology', '9990002222', 'Brain'),
(3, 'Dr. Kapoor', 'Orthopedics', '9990003333', 'Bone'),
(4, 'Dr. Verma', 'Pediatrics', '9990004444', 'Children'),
(5, 'Dr. Rao', 'Dermatology', '9990005555', 'Skin'),
(6, 'Dr. Sen', 'ENT', '9990006666', 'Ear Nose Throat'),
(7, 'Dr. Mishra', 'Psychiatry', '9990007777', 'Mental Health'),
(8, 'Dr. Naik', 'Gastroenterology', '9990008888', 'Stomach'),
(9, 'Dr. Pathak', 'Ophthalmology', '9990009999', 'Eye'),
(10, 'Dr. Das', 'Pulmonology', '9990001212', 'Lungs');
```

```
CREATE TABLE Appointments (
    appointment_id INT PRIMARY KEY,
    patient_id INT,
    doctor_id INT,
    appointment_date DATE,
    reason VARCHAR (100),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);

INSERT INTO Appointments VALUES
(1, 1, 1, '2025-04-01', 'Chest Pain'),
(2, 2, 2, '2025-04-02', 'Headache'),
(3, 3, 3, '2025-04-03', 'Back Pain'),
(4, 4, 4, '2025-04-04', 'Fever'),
(5, 5, 5, '2025-04-05', 'Skin Rash'),
(6, 6, 6, '2025-04-06', 'Ear Pain'),
(7, 7, 7, '2025-04-07', 'Anxiety'),
(8, 8, 8, '2025-04-08', 'Stomach Pain'),
(9, 9, 9, '2025-04-09', 'Blurred Vision'),
(10, 10, 10, '2025-04-10', 'Cough');

CREATE TABLE Medications (
    medication_id INT PRIMARY KEY,
    name VARCHAR (50),
    type VARCHAR (30),
    dosage VARCHAR (20),
    price DECIMAL (10,2)
```

```
);

INSERT INTO Medications VALUES
(1, 'Paracetamol', 'Tablet', '500mg', 10.00),
(2, 'Crocin', 'Tablet', '650mg', 15.00),
(3, 'Amoxicillin', 'Capsule', '250mg', 20.00),
(4, 'Azithromycin', 'Tablet', '500mg', 25.00),
(5, 'Cetirizine', 'Tablet', '10mg', 5.00),
(6, 'Dolo', 'Tablet', '650mg', 12.00),
(7, 'Ibuprofen', 'Tablet', '400mg', 8.00),
(8, 'Pantoprazole', 'Tablet', '40mg', 18.00),
(9, 'Ranitidine', 'Tablet', '150mg', 9.00),
(10, 'Metformin', 'Tablet', '500mg', 22.00);

CREATE TABLE Prescriptions (
    prescription_id INT PRIMARY KEY,
    appointment_id INT,
    medication_id INT,
    dosage VARCHAR (20),
    duration VARCHAR (20),
    FOREIGN KEY (appointment_id) REFERENCES
Appointments(appointment_id),
    FOREIGN KEY (medication_id) REFERENCES
Medications(medication_id)
);
```

```
INSERT INTO Prescriptions VALUES
(1, 1, 1, '500mg', '5 days'),
(2, 2, 2, '650mg', '3 days'),
(3, 3, 3, '250mg', '7 days'),
(4, 4, 4, '500mg', '3 days'),
(5, 5, 5, '10mg', '5 days'),
(6, 6, 6, '650mg', '2 days'),
(7, 7, 7, '400mg', '4 days'),
(8, 8, 8, '40mg', '10 days'),
(9, 9, 9, '150mg', '5 days'),
(10, 10, 10, '500mg', '15 days');

SELECT * FROM Patients;

SELECT name, specialization FROM Doctors;

SELECT * FROM Appointments WHERE patient_id = 3;

SELECT * FROM Medications WHERE price > 10;

SELECT * FROM Appointments WHERE appointment_date = '2025-04-05';

SELECT P.name AS Patient, D.name AS Doctor, A.appointment_date
FROM Appointments A
JOIN Patients P ON A.patient_id = P.patient_id
JOIN Doctors D ON A.doctor_id = D.doctor_id;
```

```
SELECT A.appointment_id, P.name, M.name AS Medication
FROM Prescriptions PR
JOIN Appointments A ON PR.appointment_id = A.appointment_id
JOIN Medications M ON PR.medication_id = M.medication_id
JOIN Patients P ON A.patient_id = P.patient_id;

SELECT D.name, COUNT (*) AS Total_Appointments
FROM Appointments A
JOIN Doctors D ON A.doctor_id = D.doctor_id
GROUP BY D.name;

SELECT COUNT (*) AS Total_Patients FROM Patients;

SELECT AVG (price) AS Avg_Price FROM Medications;

SELECT * FROM Medications WHERE price BETWEEN 10 AND 20;

SELECT appointment_date, COUNT (*) FROM Appointments GROUP BY
appointment_date;

SELECT * FROM Patients WHERE age > 30;

SELECT * FROM Doctors WHERE department = 'Heart';

SELECT * FROM Medications WHERE name LIKE '%in%';

SELECT name FROM Patients
WHERE patient_id IN (SELECT patient_id FROM Appointments);
```

```
SELECT name FROM Doctors
WHERE doctor_id NOT IN (SELECT doctor_id FROM Appointments);

SELECT * FROM Medications ORDER BY price DESC LIMIT 5;

SELECT * FROM Appointments ORDER BY appointment_date DESC
LIMIT 3;

SELECT PR.prescription_id, PA.name AS Patient, D.name AS Doctor,
M.name AS Medication, PR.dosage, PR.duration
FROM Prescriptions PR
JOIN Appointments A ON PR.appointment_id = A.appointment_id
JOIN Patients PA ON A.patient_id = PA.patient_id
JOIN Doctors D ON A.doctor_id = D.doctor_id
JOIN Medications M ON PR.medication_id = M.medication_id;
```

- **SQL QUERIES WITH OUTPUT:**

```
108 •    SELECT * FROM Patients;
```

Result Grid | Filter Rows: | Edit:

patient_id	name	age	gender	contact_number
1	Amit Sharma	30	Male	9876543210
2	Sonal Gupta	25	Female	9876543211
3	Rahul Mehta	40	Male	9876543212
4	Priya Desai	35	Female	9876543213
5	Anil Kumar	50	Male	9876543214
6	Meena Rathi	45	Female	9876543215
7	Rohit Jain	28	Male	9876543216

```
110 •   SELECT name, specialization FROM Doctors;
111
```

Result Grid | Filter Rows: | Export:

name	specialization
Dr. Sharma	Cardiology
Dr. Joshi	Neurology
Dr. Kapoor	Orthopedics
Dr. Verma	Pediatrics
Dr. Rao	Dermatology
Dr. Sen	ENT
Dr. Mishra	Psychiatry

```
112 •   SELECT * FROM Appointments WHERE patient_id = 3;
113
```

Result Grid | Filter Rows: | Edit:

appointment_id	patient_id	doctor_id	appointment_date	reason
3	3	3	2025-04-03	Back Pain
NULL	NULL	NULL	NULL	NULL

```
114 •   SELECT * FROM Medications WHERE price > 10;
115
```

Result Grid | Filter Rows: | Edit:

medication_id	name	type	dosage	price
2	Crocin	Tablet	650mg	15.00
3	Amoxicillin	Capsule	250mg	20.00
4	Azithromycin	Tablet	500mg	25.00
6	Dolo	Tablet	650mg	12.00
8	Pantoprazole	Tablet	40mg	18.00
10	Metformin	Tablet	500mg	22.00
NULL	NULL	NULL	NULL	NULL

```
116 •  SELECT * FROM Appointments WHERE appointment_date = '2025-04-05';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

appointment_id	patient_id	doctor_id	appointment_date	reason
5	5	5	2025-04-05	Skin Rash
NULL	NULL	NULL	NULL	NULL

```
138 •  SELECT * FROM Medications WHERE price BETWEEN 10 AND 20;
```

Result Grid | Filter Rows: | Edit: | Export/Impor

medication_id	name	type	dosage	price
1	Paracetamol	Tablet	500mg	10.00
2	Crocin	Tablet	650mg	15.00
3	Amoxicillin	Capsule	250mg	20.00
6	Dolo	Tablet	650mg	12.00
8	Pantoprazole	Tablet	40mg	18.00
NULL	NULL	NULL	NULL	NULL

```
118 •  SELECT P.name AS Patient, D.name AS Doctor, A.appointment_date
119    FROM Appointments A
120    JOIN Patients P ON A.patient_id = P.patient_id
121    JOIN Doctors D ON A.doctor_id = D.doctor_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Patient	Doctor	appointment_date
Amit Sharma	Dr. Sharma	2025-04-01
Sonal Gupta	Dr. Joshi	2025-04-02
Rahul Mehta	Dr. Kapoor	2025-04-03
Priya Desai	Dr. Verma	2025-04-04
Anil Kumar	Dr. Rao	2025-04-05
Meena Rathi	Dr. Sen	2025-04-06
Rohit Jain	Dr. Mishra	2025-04-07

```
129 •   SELECT D.name, COUNT(*) AS Total_Appointments
130     FROM Appointments A
131     JOIN Doctors D ON A.doctor_id = D.doctor_id
132     GROUP BY D.name;
```

Result Grid | Filter Rows: | Export: | Wrap

name	Total_Appointments
Dr. Sharma	1
Dr. Joshi	1
Dr. Kapoor	1
Dr. Verma	1
Dr. Rao	1
Dr. Sen	1
Dr. Mishra	1

```
123 •   SELECT A.appointment_id, P.name, M.name AS Medication
124     FROM Prescriptions PR
125     JOIN Appointments A ON PR.appointment_id = A.appointment_id
126     JOIN Medications M ON PR.medication_id = M.medication_id
127     JOIN Patients P ON A.patient_id = P.patient_id;
128
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

appointment_id	name	Medication
1	Amit Sharma	Paracetamol
2	Sonal Gupta	Crocin
3	Rahul Mehta	Amoxicillin
4	Priya Desai	Azithromycin
5	Anil Kumar	Cetirizine
6	Meena Rathi	Dolo
7	Rohit Jain	Ibuprofen

```
134 •    SELECT COUNT(*) AS Total_Patients FROM Patients;
135
```

Result Grid | Filter Rows: | Export: | Wrap Cell

Total_Patients
10

```
136 •    SELECT AVG(price) AS Avg_Price FROM Medications;
137
```

Result Grid | Filter Rows: | Export: | Wrap Cell

Avg_Price
14.400000

```
140 •    SELECT appointment_date, COUNT(*) FROM Appointments GROUP BY appointment_date;
141
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

appointment_date	COUNT(*)
2025-04-01	1
2025-04-02	1
2025-04-03	1
2025-04-04	1
2025-04-05	1
2025-04-06	1
2025-04-07	1

```
142 •   SELECT * FROM Patients WHERE age > 30;
143
```

Result Grid | Filter Rows: | Edit:

patient_id	name	age	gender	contact_number
3	Rahul Mehta	40	Male	9876543212
4	Priya Desai	35	Female	9876543213
5	Anil Kumar	50	Male	9876543214
6	Meena Rathi	45	Female	9876543215
8	Kavita Yadav	32	Female	9876543217
9	Vikram Singh	38	Male	9876543218
NULL	NULL	NULL	NULL	NULL

```
144 •   SELECT * FROM Doctors WHERE department = 'Heart';
145
```

Result Grid | Filter Rows: | Edit:

doctor_id	name	specialization	contact_number	department
1	Dr. Sharma	Cardiology	9990001111	Heart
NULL	NULL	NULL	NULL	NULL

```
154 •   SELECT * FROM Medications ORDER BY price DESC LIMIT 5;
155
```

Result Grid | Filter Rows: | Edit: | Export/Im

medication_id	name	type	dosage	price
4	Azithromycin	Tablet	500mg	25.00
10	Metformin	Tablet	500mg	22.00
3	Amoxicillin	Capsule	250mg	20.00
8	Pantoprazole	Tablet	40mg	18.00
2	Crocin	Tablet	650mg	15.00
NULL	NULL	NULL	NULL	NULL

```
146 •   SELECT * FROM Medications WHERE name LIKE '%in%';
147
```

Result Grid | Filter Rows: | Edit:

medication_id	name	type	dosage	price
2	Crocin	Tablet	650mg	15.00
3	Amoxicillin	Capsule	250mg	20.00
4	Azithromycin	Tablet	500mg	25.00
5	Cetirizine	Tablet	10mg	5.00
9	Ranitidine	Tablet	150mg	9.00
10	Metformin	Tablet	500mg	22.00
NULL	NULL	NULL	NULL	NULL

```
148 •   SELECT name FROM Patients
149     WHERE patient_id IN (SELECT patient_id FROM Appointments);
150
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Amit Sharma
Sonal Gupta
Rahul Mehta
Priya Desai
Anil Kumar
Meena Rathi
Rohit Jain

```
151 •   SELECT name FROM Doctors
152     WHERE doctor_id NOT IN (SELECT doctor_id FROM Appointments);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name

```
158 •  SELECT PR.prescription_id, PA.name AS Patient, D.name AS Doctor,
159    M.name AS Medication, PR.dosage, PR.duration
160    FROM Prescriptions PR
161    JOIN Appointments A ON PR.appointment_id = A.appointment_id
162    JOIN Patients PA ON A.patient_id = PA.patient_id
163    JOIN Doctors D ON A.doctor_id = D.doctor_id
164    JOIN Medications M ON PR.medication_id = M.medication_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

prescription_id	Patient	Doctor	Medication	dosage	duration
1	Amit Sharma	Dr. Sharma	Paracetamol	500mg	5 days
2	Sonal Gupta	Dr. Joshi	Crocin	650mg	3 days
3	Rahul Mehta	Dr. Kapoor	Amoxicillin	250mg	7 days
4	Priya Desai	Dr. Verma	Azithromycin	500mg	3 days
5	Anil Kumar	Dr. Rao	Cetirizine	10mg	5 days
6	Meena Rathi	Dr. Sen	Dolo	650mg	2 days
7	Rohit Jain	Dr. Mishra	Ibuprofen	400mg	4 days

- **SUMMARY:**

Key Highlights:

- Efficient relational structure
- Clear table definitions
- Proper constraints and normalization

- **Modular Table Setup**:

  Each module (Patients, Doctors, etc.) is separated logically and structurally, making the database **scalable** and **maintainable**.

- **Learning Outcomes:**
- Gained practical experience in DBMS concepts
- Understood schema design and entity relationships
- Enhanced SQL querying skills

Project Application:

Applicable in real-world hospital systems to manage and monitor operational data efficiently.

Technologies Used:

- MySQL
- SQL Workbench / Command Line
- Windows/Linux OS

Objectives:

• Learn database schema design
• Practice data insertion and query execution
• Understand relationships among data entities
• Simulate real-life hospital management operations

Relationships:

• One-to-many: One doctor can have many appointments.
• One-to-many: One patient can book multiple appointments.
• One-to-one: One prescription per appointment.
• Many-to-one: Many prescriptions can include same medication.

CONCLUSION:

The Hospital Management System project successfully demonstrates the use of SQL in managing a healthcare database. It showcases efficient use of relational models, data integrity via constraints, and various query operations.

Observations:

- SQL effectively manages relational data.

Limitations:

- No real-time data input
- No UI integration
- Static sample data only

In conclusion,

This project provides a practical demonstration of database design and management. It helps understand real-life applications of SQL and how data is structured, queried, and maintained in healthcare environments.