



# **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: Pahulpreet Kaur  
UID: 23BCA10301  
Section: 4-‘B’**

**Submitted To:**

**Name: Mr.Arinder Singh  
Designation: Assistant Professor**



## INTRODUCTION

The **Hospital Management System** is a database project developed using MySQL, aimed at managing patient data, doctors, departments, appointments, medical records, billing, and pharmacy stock. The main objective is to digitise hospital operations, reduce manual workload, and improve efficiency and data accessibility.

The objective is to build a **relational database** that ensures data integrity, easy access, and efficient handling of daily hospital operations. Using **Entity-Relationship modelling** and **SQL queries**, the system allows for smooth patient tracking, appointment scheduling, billing management, and inventory control.

This system allows:

- Easy addition and retrieval of patient and staff data
- Efficient scheduling of appointments
- Maintenance of patient medical records
- Bill generation and payment tracking
- Medicine stock management in pharmacy

## TECHNIQUES

The primary technology used in this project is MySQL, an open-source relational database management system. The following techniques have been implemented:

- **Entity-Relationship Modeling** for data structure visualisation.
- **Normalisation** to organise data efficiently and remove redundancy.
- **SQL Queries** for data manipulation and retrieval.
- **Use of Constraints** like PRIMARY KEY, FOREIGN KEY to enforce relationships.
- **Join operations** to combine data from multiple tables.
- **Aggregate Functions** to summarize and analyze data.
- **Filtering and Sorting** to extract meaningful insights from the dataset.
- **Stored Procedures and Views** (optional enhancements) for automation.

The goal is to simulate a real-time hospital database with multiple users accessing the system concurrently. Though our current system is simplified, it lays the foundation for large-scale enterprise software.

# **SYSTEM CONFIGURATION**

## **Software Requirements:**

- MySQL Server / Workbench
- ER Diagram Design Tool (e.g., dbdiagram.io, Lucid chart)
- Text Editor or IDE (e.g., VS Code, Sublime Text)

## **Hardware Requirements:**

- RAM: Minimum 4 GB (8 GB recommended for better performance)
- Processor: Intel i3 or above
- Storage: Minimum 500 MB for database and MySQL setup

## **Operating System:** Windows/Linux/macOS

The system has been developed and tested on Windows 10 with MySQL Workbench 8.0.34. The choice of open-source tools makes the project accessible and deployable on a wide range of platforms.

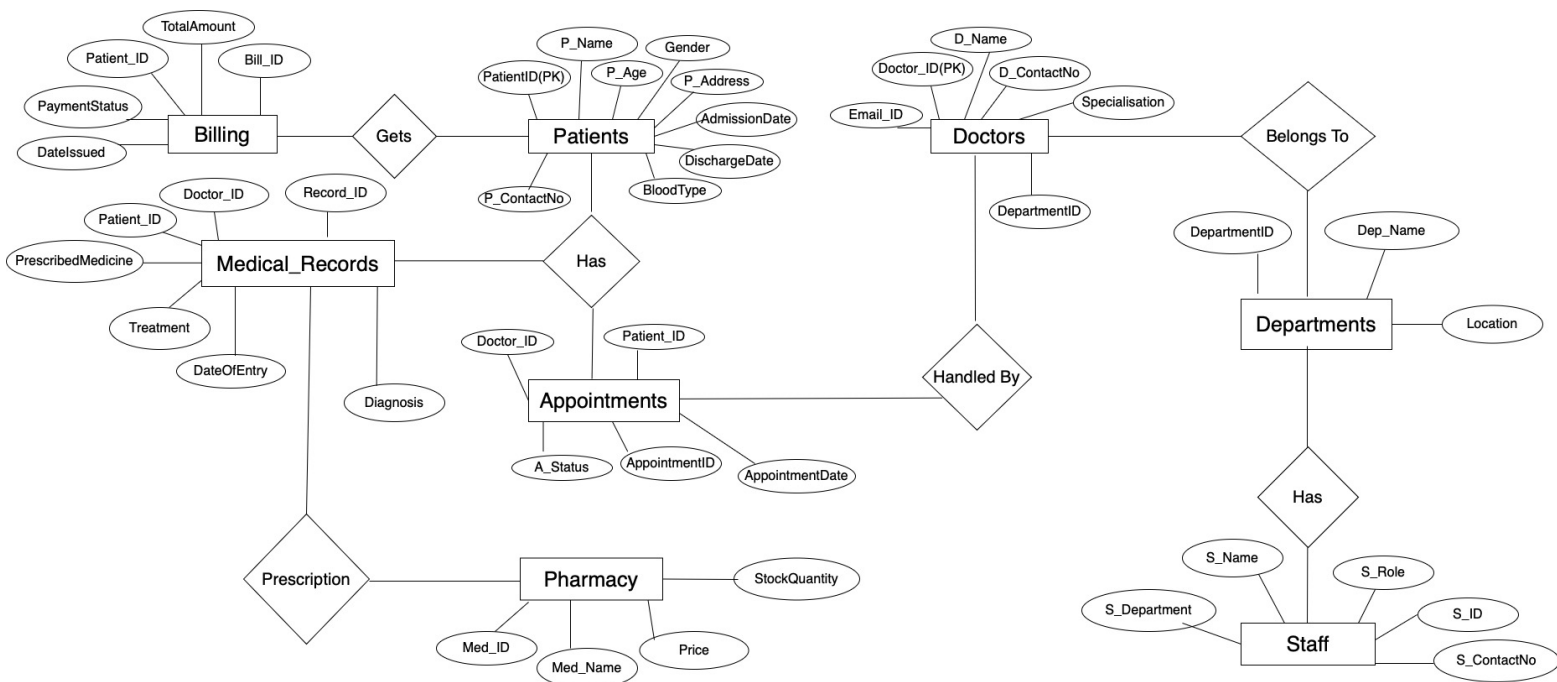
## INPUT

The system accepts a variety of inputs to populate the database and simulate hospital functionality. These inputs include:

- **Patient Information:** Name, age, gender, address, phone number, admission and discharge details.
- **Doctor Information:** Name, specialisation, department, and contact number.
- **Staff Information:** Name, role, department, and phone number.
- **Department Details:** Department name and its location within the hospital.
- **Appointment Data:** Date, time, patient and doctor involved, and reason.
- **Medical Records:** Diagnosis information, prescribed medications, and date.
- **Billing Information:** Total amount billed, payment status, and date of billing.
- **Pharmacy Inventory:** Medicine name, quantity, expiry date, and current stock.

Each input plays a vital role in maintaining an up-to-date and accurate database.

## ENTITY-RELATIONSHIP DIAGRAM



The Entity-Relationship (ER) diagram outlines the structure and relationships among different entities of the hospital. It forms the blueprint for the actual database schema.

Each entity has clearly defined attributes and is connected using appropriate relationships like one-to-many and many-to-one, ensuring normalization and avoiding data redundancy.

## RELATIONSHIP BETWEEN TABLES

These relationships ensure that the relational database mirrors real-world interactions within a hospital.

No.	Relationship Type	Parent Table	Child Table	Foreign key	Description
1	One-to-many	Departments	Doctors	DepartmentID	A department can have many doctors
2	One-to-many	Departments	Staff	DepartmentID	A department can have many staff
3	One-to-many	Doctors	Appointment	DoctorID	A doctor can have many appointments
4	One-to-many	Patients	Appointment	PatientID	A patient can have multiple appointments
5	One-to-many	Patients	MedicalRecord	PatientID	A doctor can have many medical records
6	One-to-one	Patients	Billing	PatientID (unique)	Each patient has one bill

## TABULAR FORMAT (SCHEMA)

Table Name	Primary Key	Foreign Key	Description
Patients	PatientID	—	Stores patient info
Doctors	DoctorID	DepartmentID	Stores doctor details
Staff	StaffID	DepartmentID	Hospital staff info
Appointments	AppointmentID	—	Department info
MedicalRecords	RecordID	PatientID,DoctorID	Mnage appointment
Billing	BillID	PatientID	Records Diagosis
Pharmacy	MedicineID	PatientID	Billing Details
Departments	DepartmentID	—	Med Stock Data

## TABLE CREATION

### 1. Patient Table:

```
• create table Patients(  
  Patient_ID int primary key,  
  P_Name char(30),  
  P_Age int,  
  Gender char(20),  
  P_ContactNo int,  
  P_Address varchar(50),  
  BloodType varchar(10),  
  AdmissionDate date,  
  DischargeDate date  
);
```



- **insert into** Patients(Patient\_ID,P\_Name,P\_Age,Gender,P\_ContactNo,P\_Address,BloodType,AdmissionDate,DischargeDate)  
**values**  
(101,'Rahul Sharma',45,'Male',9945783,'Delhi,India','B+', '2024-02-10', '2024-02-15'),  
(102,'Priya Verma',30,'Female',896005,'Mumbai,India','O+', '2024-03-05', NULL),  
(103,'Amit Singh',55,'Male',781432,'Kolkata,India','A-', '2024-01-20', '2024-02-01'),  
(104,'Sneha Kapoor',28,'Female',620043,'Bangalore,India','AB+', '2024-02-22', '2024-02-28'),  
(105,'Rohit Mehta',33,'Male',907611,'Chennai,India','O-', '2024-03-10', NULL),  
(106,'Neha Joshi',40,'Female',994310,'Pune,India','B-', '2024-01-15', '2024-01-30'),  
(107,'Varun Malhotra',29,'Male',879901,'Hyderabad,India','A+', '2024-02-05', '2024-02-12'),  
(108,'Ananya Sen',35,'Female',624821,'Jaipur,India','AB-', '2024-03-02', NULL);

## 2. Doctor Table:

- **create table** Doctors(  
    Doctor\_ID **int primary key**,  
    D\_Name **char**(30),  
    Specialization **char**(20),  
    D\_ContactNo **int**,  
    Email\_ID **varchar**(20),  
    DepartmentID **int**  
);
- **insert into** Doctors(Doctor\_ID,D\_Name,Specialization,D\_ContactNo,Email\_ID,DepartmentID)  
**values**  
(201,'Dr.Arjun Rao','Cardiology',985621,'Arjun01@hospital.com',1),  
(202,'Dr.Meera Das','Neurology',870023,'Meera02@hospital.com',2),  
(203,'Dr.Karan Jain','Orthopedics',992567,'Karan03@hospital.com',3),  
(204,'Dr.Aditi Roy','Dermatology',790112,'Aditi04@hospital.com',4),  
(205,'Dr.Ravi Gupta','Pediatrics',975344,'Ravi05@hospital.com',5),  
(206,'Dr.Sonia Kapoor','Gynecology',885132,'Sonia06@hospital.com',6),  
(207,'Dr.Aman Khanna','ENT',799203,'Aman07@hospital.com',7),  
(208,'Dr.Pooja Mehta','General Medicine',992116,'Pooja08@hospital.com',8);

### 3. Department Table

- **create table** Department(  
    DepartmentID **int primary key**,  
    Dep\_Name **char(30)**,  
    Location **varchar(20)**  
);
- **insert into** Department(DepartmentID, Dep\_Name, Location)  
**values**  
    (1, 'Cardiology', 'Block A-1st Floor'),  
    (2, 'Neurology', 'Block B-2nd Floor'),  
    (3, 'Orthopedics', 'Block C-3rd Floor'),  
    (4, 'Dermatology', 'Block D-4th Floor'),  
    (5, 'Pediatrics', 'Block E-5th Floor'),  
    (6, 'Gynecology', 'Block F-6th Floor'),  
    (7, 'ENT', 'Block G-7th Floor'),  
    (8, 'General Medicine', 'Block H-Ground Floor');

### 4. Appointment Table

- **create table** Appointments(  
    AppointmentID **int primary key**,  
    Patient\_ID **int**,  
    Doctor\_ID **int**,  
    AppointmentDate **date**,  
    A\_status **char(20)**  
);

- **insert into** Appointments(AppointmentID,Patient\_ID,Doctor\_ID,AppointmentDate,A\_Status)  
**values**  
(301,101,201,'2024-02-11','Completed'),  
(302,102,202,'2024-03-06','Scheduled'),  
(303,103,203,'2024-01-21','Completed'),  
(304,104,204,'2024-02-23','Completed'),  
(305,105,205,'2024-03-11','Scheduled'),  
(306,106,206,'2024-01-16','Completed'),  
(307,107,207,'2024-02-06','Completed'),  
(308,108,208,'2024-03-03','Scheduled');

## 5. Medical Record Table

- **create table** Medical\_Records(  
Record\_ID **int primary key**,  
Patient\_ID **int**,  
Diagnosis **char(20)**,  
Treatment **char(20)**,  
PrescribedMedicine **char(20)**,  
Doctor\_ID **int**,  
DateOfEntry **date**  
);
- **insert into** Medical\_Records(Record\_ID,Patient\_ID,Diagnosis,Treatment,PrescribedMedicine,Doctor\_ID ,DateOfEntry)  
**values**  
(401,101,'Hypertension','Medication','Amlodipine',201,'2024-02-11'),  
(402,102,'Migraine','Therapy','Sumatriptan',202,'2024-03-06'),  
(403,103,'Fracture','Surgery','Painkillers',203,'2024-01-21'),  
(404,104,'Skin Allergy','Ointment','Hydrocortisone',204,'2024-02-23'),  
(405,105,'Fever','Rest & Fluids','Paracetamol',205,'2024-03-11'),  
(406,106,'Pregnancy Check','Monitoring','Vitamins',206,'2024-01-16'),  
(407,107,'Ear Infection','Antibiotics','Amoxicillin',207,'2024-02-06'),  
(408,108,'Diabetes','MDiet Control','Insulin',208,'2024-03-03');

## 6. Billing Table

- `create table Billing(  
    Bill_ID int primary key,  
    Patient_ID int,  
    TotalAmount int,  
    PaymentStatus char(20),  
    DateIssued date  
);`
- `insert into Billing(Bill_ID,Patient_ID,TotalAmount,PaymentStatus,DateIssued)  
values  
(501,101,5000,'Paid','2024-02-15'),  
(502,102,3000,'Pending','2024-03-06'),  
(503,103,15000,'Paid','2024-01-21'),  
(504,104,7000,'Paid','2024-02-23'),  
(505,105,2500,'Pending','2024-03-11'),  
(506,106,4000,'Paid','2024-01-16'),  
(507,107,6000,'Paid','2024-02-06'),  
(508,108,9000,'Pending','2024-03-03');`

## 7. Pharmacy Table

- `create table Pharmacy(  
    Med_ID int primary key,  
    Med_Name char(20),  
    StockQuantity int,  
    Price int  
);`

- **insert into** Pharmacy(Med\_ID,Med\_Name,StockQuantity,Price)  
**values**  
(601,'Paracetamol',100,10),  
(602,'Amoxicillin',50,20),  
(603,'Amlodipine',80,15),  
(604,'Insulin',30,50),  
(605,'Sumatriptan',20,40),  
(606,'Hydrocortisone',25,25),  
(607,'Painkillers',60,12),  
(608,'Vitamins',90,30);

## 8. Staff Table

- **create table** Staff(  
    S\_ID **int primary key**,  
    S\_Name **char(40)**,  
    S\_Role **char(30)**,  
    S\_ContactNo **int**,  
    S\_Department **int**  
);
- **insert into** Staff(S\_ID,S\_Name ,S\_Role,S\_ContactNo,S\_Department)  
**values**  
(701,'Sunita Sharma','Nurse',991462,1),  
(702,'Ramesh Gupta','Receptionist',98743,8),  
(703,'Anjali Verma','Lab Technician',80032,2),  
(704,'Suresh Mehta','Pharmacist',781400,8),  
(705,'SPoonam Kaur','Ward Boy',894211,3),  
(706,'Kavita Joshi','Nurse',774112,5),  
(707,'Mohit Bansal','Cleaner',993899,6),  
(708,'SDeepa Rani','Accountant',98148,8);

## SQL QUERIES (13 Queries)

- **select\*from Patients;**

Patient_ID	P_Name	P_Age	Gender	P_ContactNo	P_Address	BloodType	AdmissionDate	DischargeDate
101	Rahul Sharma	45	Male	9945783	Delhi,India	B+	2024-02-10	2024-02-15
102	Priya Verma	30	Female	896005	Mumbai,India	O+	2024-03-05	NULL
103	Amit Singh	55	Male	781432	Kolkata,India	A-	2024-01-20	2024-02-01
104	Sneha Kapoor	28	Female	620043	Bangalore,India	AB+	2024-02-22	2024-02-28
105	Rohit Mehta	33	Male	907611	Chennai,India	O-	2024-03-10	NULL
106	Neha Joshi	40	Female	994310	Pune,India	B-	2024-01-15	2024-01-30
107	Varun Malhotra	29	Male	879901	Hyderabad,India	A+	2024-02-05	2024-02-12
108	Ananya Sen	35	Female	624821	Jaipur,India	AB-	2024-03-02	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **select** doctors.D\_Name,Department.Dep\_Name  
**from** Doctors  
**inner join** Department **on** Doctors.DepartmentID=Department.DepartmentID;

D_Name	Dep_Name
Dr.Arjun Rao	Cardiology
Dr.Meera Das	Neurology
Dr.Karan Jain	Orthopedics
Dr.Aditi Roy	Dermatology
Dr.Ravi Gupta	Pediatrics
Dr.Sonia Kapoor	Gynecology
Dr.Aman Khanna	ENT
Dr.Pooja Mehta	General Medicine



- `select sum(TotalAmount) as TotalPaidAmount  
from Billing  
where PaymentStatus='Paid';`

TotalPaidAmount
40000

- `select Staff.S_Name,Staff.S_Role,Department.Dep_Name  
from Staff  
join Department on Staff.S_Department=Department.DepartmentID  
where Department.Dep_Name='General Medicine';`

S_Name	S_Role	Dep_Name
Ramesh Gupta	Receptionist	General Medicine
Suresh Mehta	Pharmacist	General Medicine
SDeepa Rani	Accountant	General Medicine

- **select**\***from** Pharmacy  
**where** StockQuantity<30;

Med_ID	Med_Name	StockQuanti...	Price
605	Sumatriptan	20	40
606	Hydrocortisone	25	25
NULL	NULL	NULL	NULL

- **update** Billing  
**Set** PaymentStatus='Paid'  
**where** Bill\_ID=502;

Bill_ID	Patient_ID	TotalAmount	PaymentStatus	DateIssued
501	101	5000	Paid	2024-02-15
502	102	3000	Paid	2024-03-06
503	103	15000	Paid	2024-01-21
504	104	7000	Paid	2024-02-23
505	105	2500	Pending	2024-03-11
506	106	4000	Paid	2024-01-16
507	107	6000	Paid	2024-02-06
508	108	9000	Pending	2024-03-03
NULL	NULL	NULL	NULL	NULL



- `select*from` Appointments  
`where` AppointmentDate>'2024-03-01';

Appointment...	Patient_ID	Doctor_ID	AppointmentD...	A_status
302	102	202	2024-03-06	Scheduled
305	105	205	2024-03-11	Scheduled
308	108	208	2024-03-03	Scheduled
NULL	NULL	NULL	NULL	NULL

- `select` Patients.P\_Name,Medical\_Records.Diagnosis,Medical\_Records.PrescribedMedicine  
`from` Medical\_Records  
`inner join` Patients `on` Patients.Patient\_ID=Medical\_Records.Patient\_ID;

P_Name	Diagnosis	PrescribedMedici...
Rahul Sharma	Hypertension	Amlodipine
Priya Verma	Migraine	Sumatriptan
Amit Singh	Fracture	Painkillers
Sneha Kapoor	Skin Allergy	Hydrocortisone
Rohit Mehta	Fever	Paracetamol
Neha Joshi	Pregnancy Check	Vitamins
Varun Malhotra	Ear Infection	Amoxicillin
Ananya Sen	Diabetes	Insulin

- **SELECT** P\_Age, COUNT(\*) **AS** PatientCount  
**FROM** Patients  
**GROUP BY** P\_Age  
**HAVING** COUNT(\*) <2;

P_Age	PatientCount
45	1
30	1
55	1
28	1
33	1
40	1
29	1
35	1

- **SELECT** P\_Name, AdmissionDate,  
COUNT(\*) **OVER** () **AS** FebAdmissions  
**FROM** Patients  
**WHERE** MONTH(AdmissionDate) = 2 **AND** YEAR(AdmissionDate) = 2024;

P_Name	AdmissionDate	FebAdmissions
Rahul Sharma	2024-02-10	3
Sneha Kapoor	2024-02-22	3
Varun Malhotra	2024-02-05	3

- Select** Doctors.D\_Name,Doctors.Specialization,Department.Location  
**from** Doctors  
**join** Department **on** Department.DepartmentID=Doctors.DepartmentID  
**where** Department.Location **Like** '%1st Floor%' **OR** Department.Location **Like** '%2nd Floor%';

	D_Name	Specialization	Location
	Dr.Arjun Rao	Cardiology	Block A-1st Floor
	Dr.Meera Das	Neurology	Block B-2nd Floor

- select** Patients.P\_Name,Billing.TotalAmount  
**from** Billing  
**join** Patients **on** Billing.Patient\_ID=Patients.Patient\_ID  
**Order BY** Billing.TotalAmount;

	P_Name	TotalAmount
	Rohit Mehta	2500
	Priya Verma	3000
	Neha Joshi	4000
	Rahul Sharma	5000
	Varun Malhotra	6000
	Sneha Kapoor	7000
	Ananya Sen	9000
	Amit Singh	15000

- ```
select Patients.P_Name,Billing.TotalAmount
from Billing
join Patients on Billing.Patient_ID=Patients.Patient_ID
Order BY Billing.TotalAmount desc
Limit 1;
```

| P_Name     | TotalAmount |
|------------|-------------|
| Amit Singh | 15000       |



## **SUMMARY**

The Hospital Management System (HMS) designed in this project serves as a comprehensive solution for managing hospital operations through a relational MySQL database. It incorporates crucial modules such as patient registration, doctor scheduling, staff management, appointment booking, medical record keeping, billing, and pharmacy stock handling.

Each entity is normalised and linked using appropriate relationships to ensure data consistency, integrity, and reduced redundancy. The implementation of SQL queries demonstrates the ability to retrieve complex information using techniques like joins, aggregate functions, filtering, and grouping. This system lays the foundation for building a larger, more interactive platform to support hospital administration.

Through this project, the importance of a well-structured back-end system is highlighted, which can support healthcare professionals by reducing manual workload and enhancing data accessibility.



## **CONCLUSION**

### **Observations:**

- The database successfully simulates the core functionalities of a hospital.
- Relationships between entities such as patients, doctors, and departments are clearly established.
- SQL queries enhance decision-making and streamline data management.
- Data retrieval is efficient and scalable for larger datasets.

### **Limitations:**

- The system lacks a user-friendly interface for non-technical users.
- The pharmacy table is not directly connected to the medical prescriptions.
- Real-time features such as alerts, notifications, or appointment reminders are not implemented.
- The system currently assumes static data without user authentication or access control.

## **Future Scope:**

- Integration with a web-based or mobile front-end for ease of access.
- Implementing user roles for doctors, patients, and administrators with login-based access.
- Linking pharmacy inventory to prescriptions for automatic stock management.
- Adding modules for emergency services, diagnostics, and insurance claims.
- Real-time reporting, dashboards, and analytics for administrative decision-making.