**Project Name: Hospital Management Database System**

**1. Project Overview Document**

**Project Title**: Hospital Management Database System  
**Objective**: Design and implement a database system to manage patient records, doctor details, visits/appointments, billing, and reporting using MySQL and DBeaver.

**Key Features:**

* Track patients and their admission/discharge details
* Manage doctor specialties and contacts
* Log patient visits and diagnoses
* Generate and manage billing
* Automate discharge status updates via triggers
* Produce visit and billing reports

**Tools Used**:

* Database: MySQL 8+
* Client: DBeaver
* Language: SQL

**2. ER Diagram (Text Format)**

Patients (patient\_id PK)

├─ name

├─ age

├─ gender

├─ contact

├─ admission\_date

├─ discharge\_date

└─ status

Doctors (doctor\_id PK)

├─ name

├─ specialization

└─ contact

Visits (visit\_id PK)

├─ patient\_id FK → Patients

├─ doctor\_id FK → Doctors

├─ visit\_date

├─ reason

└─ diagnosis

Bills (bill\_id PK)

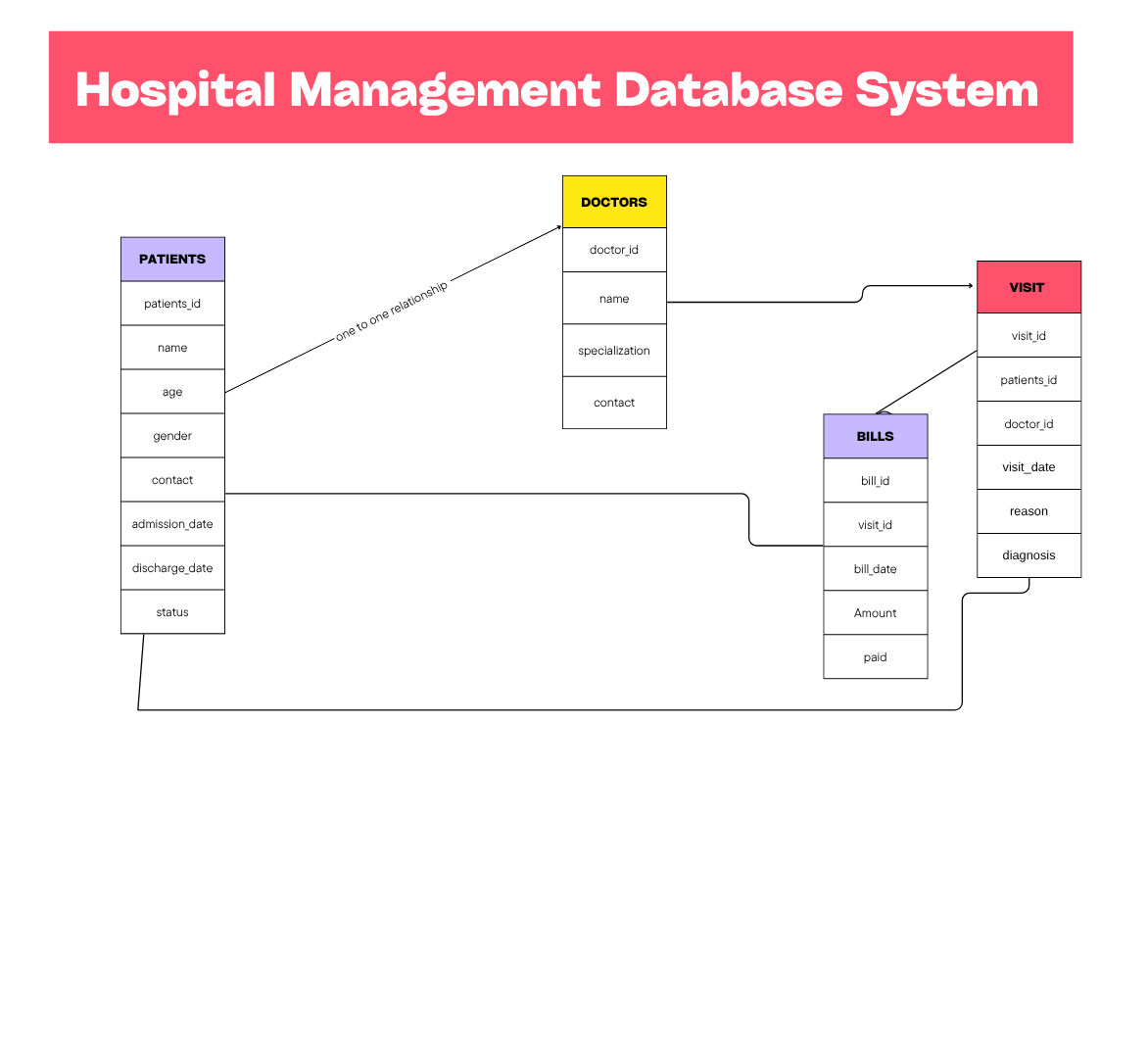
├─ visit\_id FK → Visits

├─ bill\_date

├─ amount

└─ paid

**E-R Diagram**

****

**3. SQL Schema File (hospital\_schema.sql)**

CREATE DATABASE HospitalDB;

USE HospitalDB;

CREATE TABLE Patients (

patient\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

age INT,

gender ENUM('Male', 'Female', 'Other'),

contact VARCHAR(15),

admission\_date DATE,

discharge\_date DATE,

status VARCHAR(50)

);

CREATE TABLE Doctors (

doctor\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

specialization VARCHAR(100),

contact VARCHAR(15)

);

CREATE TABLE Visits (

visit\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

visit\_date DATE,

reason VARCHAR(255),

diagnosis TEXT,

FOREIGN KEY (patient\_id) REFERENCES Patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctors(doctor\_id)

);

CREATE TABLE Bills (

bill\_id INT AUTO\_INCREMENT PRIMARY KEY,

visit\_id INT,

bill\_date DATE,

amount DECIMAL(10,2),

paid BOOLEAN DEFAULT FALSE,

FOREIGN KEY (visit\_id) REFERENCES Visits(visit\_id)

);

**4. Sample Data (sample\_data.sql)**

-- Insert Doctors

INSERT INTO Doctors (name, specialization, contact) VALUES

('Dr. A Sharma', 'Cardiology', '9876543210'),

('Dr. B Mehta', 'Neurology', '9876543211');

-- Insert Patients

INSERT INTO Patients (name, age, gender, contact, admission\_date, status) VALUES

('John Doe', 45, 'Male', '9811122233', '2025-07-20', 'Admitted'),

('Anita Kumari', 34, 'Female', '9822233445', '2025-07-22', 'Admitted');

-- Insert Visits

INSERT INTO Visits (patient\_id, doctor\_id, visit\_date, reason, diagnosis) VALUES

(1, 1, '2025-07-21', 'Chest pain', 'Mild heart condition'),

(2, 2, '2025-07-23', 'Headache', 'Migraine');

-- Insert Bills

INSERT INTO Bills (visit\_id, bill\_date, amount, paid) VALUES

(1, '2025-07-21', 3000.00, TRUE),

(2, '2025-07-23', 2500.00, FALSE);

**5. Stored Procedure File (procedures.sql)**

DELIMITER //

CREATE PROCEDURE CalculateBill(IN visitId INT, IN baseAmount DECIMAL(10,2))

BEGIN

DECLARE totalAmount DECIMAL(10,2);

SET totalAmount = baseAmount + (baseAmount \* 0.18); -- 18% tax

INSERT INTO Bills (visit\_id, bill\_date, amount, paid)

VALUES (visitId, CURDATE(), totalAmount, FALSE);

END //

DELIMITER ;

**6. Triggers (triggers.sql)**

DELIMITER //

CREATE TRIGGER update\_status\_after\_discharge

BEFORE UPDATE ON Patients

FOR EACH ROW

BEGIN

IF NEW.discharge\_date IS NOT NULL THEN

SET NEW.status = 'Discharged';

END IF;

END //

DELIMITER ;

**7. Queries and Reports (queries.sql)**

**A. List Appointments:**

SELECT v.visit\_id, p.name AS patient, d.name AS doctor, v.visit\_date, v.reason

FROM Visits v

JOIN Patients p ON v.patient\_id = p.patient\_id

JOIN Doctors d ON v.doctor\_id = d.doctor\_id;

**B. Unpaid Bills:**

SELECT b.bill\_id, p.name AS patient, b.amount, b.bill\_date

FROM Bills b

JOIN Visits v ON b.visit\_id = v.visit\_id

JOIN Patients p ON v.patient\_id = p.patient\_id

WHERE b.paid = FALSE;

**C. Visit & Billing Report:**

SELECT

p.name AS patient\_name,

d.name AS doctor\_name,

v.visit\_date,

v.reason,

v.diagnosis,

b.amount,

b.paid

FROM Visits v

JOIN Patients p ON v.patient\_id = p.patient\_id

JOIN Doctors d ON v.doctor\_id = d.doctor\_id

LEFT JOIN Bills b ON v.visit\_id = b.visit\_id;

**8. Usage Instructions (README)**

# Hospital Management Database System

## Overview

A mini MySQL project to manage hospital operations: patients, doctors, visits, and billing.

## Tools Required

- MySQL Server

- DBeaver (or any SQL client)