```sql
CREATE DATABASE HospitalManagementSystem;
USE HospitalManagementSystem;

CREATE TABLE Departments (
    DepartmentID INT AUTO_INCREMENT PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL,
    Location VARCHAR(100),
    HeadOfDepartment INT
);

CREATE TABLE Patients (
    PatientID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE,
    Gender CHAR(1),
    Address VARCHAR(200),
    Phone VARCHAR(20),
    Email VARCHAR(100),
    BloodType VARCHAR(5),
    RegistrationDate DATE
);

CREATE TABLE Doctors (
    DoctorID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Specialty VARCHAR(100),
    Phone VARCHAR(20),
    Email VARCHAR(100),
    DepartmentID INT,
    HireDate DATE,
    Salary DECIMAL(10,2),
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);

CREATE TABLE Appointments (
    AppointmentID INT AUTO_INCREMENT PRIMARY KEY,
    DoctorID INT NOT NULL,
    PatientID INT NOT NULL,
    AppointmentDate DATETIME NOT NULL,
    Purpose VARCHAR(200),
    Status ENUM('Scheduled', 'Completed', 'Cancelled', 'No-Show')
DEFAULT 'Scheduled',
    Notes TEXT,
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);


CREATE TABLE MedicalRecords (
    RecordID INT AUTO_INCREMENT PRIMARY KEY,
    PatientID INT,
    DoctorID INT,
    DateCreated DATETIME DEFAULT NOW(),
    Diagnosis VARCHAR(200),
    Treatment TEXT,
    Prescription TEXT,
    Notes TEXT,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);

CREATE TABLE Rooms (
    RoomID INT AUTO_INCREMENT PRIMARY KEY,
    RoomNumber VARCHAR(20) NOT NULL,
    RoomType ENUM('General', 'Private', 'ICU', 'Operation Theater',
'Emergency') NOT NULL,
    DepartmentID INT,
    Status ENUM('Available', 'Occupied', 'Maintenance') DEFAULT
'Available',
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);

CREATE TABLE Admissions (
    AdmissionID INT AUTO_INCREMENT PRIMARY KEY,
    PatientID INT,
    RoomID INT,
    AdmissionDate DATETIME NOT NULL,
    DischargeDate DATETIME,
    Reason VARCHAR(200),
    Status ENUM('Admitted', 'Discharged', 'Transferred') DEFAULT
'Admitted',
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)
);

CREATE TABLE Billing (
    BillID INT AUTO_INCREMENT PRIMARY KEY,
    PatientID INT,
    AdmissionID INT,
    TotalAmount DECIMAL(10,2) NOT NULL,
    PaidAmount DECIMAL(10,2) DEFAULT 0,
    BillDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    DueDate DATE,
    Status ENUM('Pending', 'Paid', 'Partial', 'Overdue') DEFAULT
'Pending',
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (AdmissionID) REFERENCES Admissions(AdmissionID)
);

CREATE TABLE Staff (
    StaffID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Role VARCHAR(50) NOT NULL,
    DepartmentID INT,
    Phone VARCHAR(20),
    Email VARCHAR(100),
    HireDate DATE,
    Salary DECIMAL(10,2),
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);

ALTER TABLE Departments
```

```
ADD CONSTRAINT FK_Departments_Doctors
FOREIGN KEY (HeadOfDepartment) REFERENCES Doctors(DoctorID)
ON DELETE SET NULL;

INSERT INTO Departments (DepartmentName, Location) VALUES
('Cardiology', 'Block A'),
('Neurology', 'Block B'),
('Pediatrics', 'Block C'),
('Orthopedics', 'Block D'),
('Emergency', 'Block E'),
('Oncology', 'Block F'),
('Radiology', 'Block G'),
('Urology', 'Block H'),
('Dermatology', 'Block I'),
('ENT', 'Block J');

INSERT INTO Patients (FirstName, LastName, DateOfBirth, Gender,
Address, Phone, Email, BloodType, RegistrationDate) VALUES
('John', 'Doe', '1985-06-15', 'M', '123 Main St', '9000000001',
'john@example.com', 'A+', '2022-01-10'),
('Jane', 'Smith', '1990-08-22', 'F', '456 Oak Ave', '9000000002',
'jane@example.com', 'B+', '2022-02-11'),
('Alice', 'Johnson', '1975-03-30', 'F', '789 Pine Rd', '9000000003',
'alice@example.com', 'O-', '2022-03-15'),
('Bob', 'Brown', '2000-12-01', 'M', '321 Maple Ln', '9000000004',
'bob@example.com', 'AB+', '2022-04-20'),
('Tom', 'White', '1982-07-07', 'M', '654 Elm St', '9000000005',
'tom@example.com', 'A-', '2022-05-25'),
('Sara', 'Lee', '1995-11-11', 'F', '987 Cedar Blvd', '9000000006',
'sara@example.com', 'B-', '2022-06-30'),
('Ryan', 'Davis', '1988-01-05', 'M', '123 Hilltop Dr', '9000000007',
'ryan@example.com', 'O+', '2022-07-01'),
('Emily', 'Clark', '1999-10-10', 'F', '456 Sunset Rd', '9000000008',
'emily@example.com', 'AB-', '2022-08-15'),
('Mark', 'Lewis', '1970-04-25', 'M', '789 Sunrise St', '9000000009',
'mark@example.com', 'A+', '2022-09-19'),
('Olivia', 'Walker', '2001-09-09', 'F', '111 Lake View', '9000000010',
'olivia@example.com', 'O+', '2022-10-22');

INSERT INTO Doctors (FirstName, LastName, Specialty, Phone, Email,
DepartmentID, HireDate, Salary) VALUES
('Dr. John', 'Smith', 'Cardiology', '8000000001',
'drjohn@hospital.com', 1, '2015-06-01', 120000.00),
('Dr. Alice', 'Brown', 'Neurology', '8000000002',
'dralice@hospital.com', 2, '2016-07-12', 125000.00),
('Dr. Bob', 'Taylor', 'Pediatrics', '8000000003', 'drbob@hospital.com',
3, '2017-08-20', 110000.00),
('Dr. Sarah', 'Lee', 'Orthopedics', '8000000004',
'drsarah@hospital.com', 4, '2018-09-15', 130000.00),
('Dr. Mike', 'Davis', 'Emergency', '8000000005', 'drmike@hospital.com',
5, '2019-10-05', 115000.00),
('Dr. Karen', 'Clark', 'Oncology', '8000000006',
'drkaren@hospital.com', 6, '2020-11-11', 140000.00),
('Dr. Kevin', 'Moore', 'Radiology', '8000000007',
'drkevin@hospital.com', 7, '2021-01-01', 135000.00),
('Dr. Rachel', 'Martin', 'Urology', '8000000008',
'drrachel@hospital.com', 8, '2021-02-14', 118000.00),
('Dr. Paul', 'Allen', 'Dermatology', '8000000009',
'drpaul@hospital.com', 9, '2021-03-30', 122000.00),
('Dr. Amy', 'Scott', 'ENT', '8000000010', 'dramyscott@hospital.com',
10, '2021-04-20', 121000.00);

INSERT INTO Appointments (DoctorID, PatientID, AppointmentDate,
Purpose, Status, Notes) VALUES
(1, 1, '2025-08-01 10:00:00', 'General Checkup', 'Scheduled', 'First
visit - basic vitals.'),
(2, 2, '2025-08-02 11:30:00', 'Dental Cleaning', 'Completed', 'No
cavities, advised flossing.'),
(3, 3, '2025-08-03 14:00:00', 'Eye Test', 'Scheduled', 'Patient
complains of blurry vision.'),
(1, 4, '2025-08-04 09:15:00', 'Follow-up', 'Cancelled', 'Rescheduled
due to doctor absence.'),
(4, 5, '2025-08-05 16:45:00', 'Cardiology Consultation', 'Scheduled',
'Check ECG and BP history.'),
(2, 6, '2025-08-06 13:20:00', 'Toothache', 'Completed', 'Filled cavity
in upper left molar.'),
(3, 7, '2025-08-07 10:00:00', 'Eye Irritation', 'Completed',
'Prescribed lubricating drops.'),
(5, 8, '2025-08-08 12:30:00', 'Dermatology Visit', 'No-Show', 'Patient
did not attend.'),
(1, 9, '2025-08-09 11:00:00', 'Flu Symptoms', 'Scheduled', 'Check for
fever, cough, body aches.'),
(4, 10, '2025-08-10 15:30:00', 'Heartburn', 'Completed', 'Advised
dietary changes and meds.');


INSERT INTO MedicalRecords (PatientID, DoctorID, DateCreated,
Diagnosis, Treatment, Prescription, Notes) VALUES
(1, 1, NOW(), 'Hypertension', 'Lifestyle changes and medication',
'Amlodipine 5mg', ''),
(2, 2, NOW(), 'Migraine', 'Pain management', 'Sumatriptan 50mg', 'Avoid
triggers'),
(3, 3, NOW(), 'Fever', 'Paracetamol and rest', 'Paracetamol 500mg',
''),
(4, 4, NOW(), 'Fracture', 'Casting', 'Painkillers', '6-week recovery'),
(5, 5, NOW(), 'Asthma', 'Inhaler therapy', 'Salbutamol inhaler', ''),
(6, 6, NOW(), 'Cancer', 'Chemotherapy', 'Custom Protocol', 'Cycle 1
complete'),
(7, 7, NOW(), 'Kidney stone', 'Surgery', 'Painkillers', ''),
(8, 8, NOW(), 'Skin allergy', 'Topical creams', 'Hydrocortisone cream',
''),
(9, 9, NOW(), 'Acne', 'Creams and oral meds', 'Isotretinoin', ''),
(10,10,NOW(), 'Hearing loss', 'Ear irrigation', 'Nasal spray', '');

INSERT INTO Rooms (RoomNumber, RoomType, DepartmentID, Status) VALUES
('101', 'General', 1, 'Available'),
('102', 'Private', 2, 'Occupied'),
('201', 'ICU', 3, 'Available'),
('202', 'Operation Theater', 4, 'Maintenance'),
('301', 'Emergency', 5, 'Occupied'),
('302', 'General', 6, 'Available'),
('401', 'Private', 7, 'Occupied'),
('402', 'ICU', 8, 'Available'),
('501', 'Emergency', 9, 'Available'),
('502', 'General', 10, 'Maintenance');
```

```
INSERT INTO Admissions (AdmissionID, PatientID, RoomID, AdmissionDate,
DischargeDate, Reason, Status) VALUES
(1, 1, 1, '2025-07-20 10:00:00', NULL, 'High BP', 'Admitted'),
(2, 2, 2, '2023-07-18 08:30:00', '2025-07-21 14:00:00', 'Seizure',
'Discharged'),
(3, 3, 3, '2025-07-25 09:00:00', NULL, 'High fever', 'Admitted'),
(4, 4, 4, '2025-07-27 13:00:00', NULL, 'Bone fracture', 'Admitted'),
(5, 5, 5, '2025-07-28 16:45:00', NULL, 'Severe asthma', 'Admitted'),
(6, 6, 6, '2022-07-29 11:00:00', NULL, 'Chemo session', 'Admitted'),
(7, 7, 7, '2022-07-30 08:00:00', NULL, 'Stone surgery', 'Admitted'),
(8, 8, 8, '2025-07-31 12:00:00', NULL, 'Skin check', 'Admitted'),
(9, 9, 9, '2025-08-01 14:30:00', NULL, 'Acne consultation',
'Admitted'),
(10, 10, 10, '2023-08-01 09:15:00', NULL, 'Hearing loss', 'Admitted');

INSERT INTO Billing (PatientID, AdmissionID, TotalAmount, PaidAmount,
DueDate, Status, PaymentMethod) VALUES
(1, 1, 5000.00, 2500.00, '2025-08-05', 'Partial', 'Cash'),
(2, 2, 8000.00, 8000.00, '2025-07-25', 'Paid', 'Credit Card'),
(3, 3, 3000.00, 0.00, '2025-08-10', 'Pending', 'Insurance'),
(4, 4, 7000.00, 1000.00, '2025-08-08', 'Partial', 'Cash'),
(5, 5, 6500.00, 6500.00, '2025-08-10', 'Paid', 'UPI'),
(6, 6, 9000.00, 0.00, '2025-08-12', 'Pending', 'Insurance'),
(7, 7, 8500.00, 4000.00, '2025-08-15', 'Partial', 'Debit Card'),
(8, 8, 2000.00, 2000.00, '2025-08-18', 'Paid', 'Cash'),
(9, 9, 1500.00, 0.00, '2025-08-20', 'Overdue', 'Cash'),
(10, 10, 4000.00, 1000.00, '2025-08-22', 'Partial', 'Card');

INSERT INTO Staff (FirstName, LastName, Role, DepartmentID, Phone,
Email, HireDate, Salary) VALUES
('Alice', 'Morgan', 'Nurse', 1, '7000000001', 'alice.m@hospital.com',
'2020-01-10', 35000.00),
('Tom', 'Jones', 'Lab Technician', 2, '7000000002',
'tom.j@hospital.com', '2019-03-12', 40000.00),
('Nina', 'Patel', 'Receptionist', 3, '7000000003',
'nina.p@hospital.com', '2021-06-01', 30000.00),
('Sam', 'Wilson', 'Pharmacist', 4, '7000000004', 'sam.w@hospital.com',
'2018-05-05', 45000.00),
('Karen', 'Singh', 'Accountant', 5, '7000000005',
'karen.s@hospital.com', '2017-09-20', 42000.00),
('Ravi', 'Kumar', 'Technician', 6, '7000000006', 'ravi.k@hospital.com',
'2019-11-30', 39000.00),
('Linda', 'Sharma', 'Cleaner', 7, '7000000007', 'linda.s@hospital.com',
'2022-02-22', 25000.00),
('George', 'Fernandez', 'Security', 8, '7000000008',
'george.f@hospital.com', '2021-08-18', 28000.00),
('Priya', 'Verma', 'Nurse', 9, '7000000009', 'priya.v@hospital.com',
'2020-12-12', 36000.00),
('David', 'Lee', 'Manager', 10, '7000000010', 'david.l@hospital.com',
'2016-04-10', 55000.00);

INSERT INTO Staff (FirstName, LastName, Role, DepartmentID, Phone,
Email, HireDate, Salary) VALUES
('Ravi', 'Sharma', 'Nurse', 1, '9876500001',
'ravi.sharma@hospital.com', '2024-03-01', 40000),
('Deepa', 'Menon', 'Lab Tech', 1, '9876500002',
'deepa.menon@hospital.com', '2024-03-05', 47000),
('Akash', 'Verma', 'Receptionist', 1, '9876500003',
'akash.verma@hospital.com', '2024-03-10', 52000);


select * from admissions;
select * from appointments;
select * from billing;
select * from departments;
select * from doctors;
select * from medicalrecords;
select * from patients;
select * from rooms;
select * from staff;

-- Patient Information with Last Appointment--
SELECT p.PatientID,CONCAT(p.FirstName, ' ', p.LastName) AS
PatientName,p.Phone,p.Email,
    MAX(a.AppointmentDate) AS LastAppointmentDate,CONCAT(d.FirstName, '
', d.LastName) AS LastDoctorSeen
FROM Patients p
LEFT JOIN Appointments a ON p.PatientID = a.PatientID
LEFT JOIN Doctors d ON a.DoctorID = d.DoctorID
GROUP BY p.PatientID, p.FirstName, p.LastName, p.Phone, p.Email,
d.FirstName, d.LastName
ORDER BY p.LastName, p.FirstName;

-- Doctor Schedule for a Specific Day --
SELECT d.DoctorID,CONCAT(d.FirstName, ' ', d.LastName) AS
DoctorName,dep.DepartmentName,
    a.  AppointmentDate,CONCAT(p.FirstName, ' ', p.LastName) AS
        PatientName,a.Purpose,a.Status
FROM Doctors d
JOIN Departments dep ON d.DepartmentID = dep.DepartmentID
LEFT JOIN Appointments a ON d.DoctorID = a.DoctorID
LEFT JOIN Patients p ON a.PatientID = p.PatientID
WHERE DATE(a.AppointmentDate) = '2025-08-03'
ORDER BY d.LastName, d.FirstName, a.AppointmentDate;

--  Current Hospital Occupancy --
SELECT r.RoomNumber,r.RoomType,dep.DepartmentName,r.Status,
    CASE
        WHEN a.PatientID IS NOT NULL THEN CONCAT(p.FirstName, ' ',
p.LastName)
        ELSE 'Available'END AS OccupiedBy,a.AdmissionDate,a.Reason
FROM Rooms r
LEFT JOIN Departments dep ON r.DepartmentID = dep.DepartmentID
LEFT JOIN Admissions a ON r.RoomID = a.RoomID AND a.Status = 'Admitted'
LEFT JOIN Patients p ON a.PatientID = p.PatientID
ORDER BY r.RoomNumber;

-- Department Statistics --
SELECT d.DepartmentID,d.DepartmentName,COUNT(DISTINCT doc.DoctorID) AS
NumberOfDoctors,
    COUNT(DISTINCT s.StaffID) AS NumberOfStaff,COUNT(DISTINCT
a.PatientID) AS PatientsAdmittedLastMonth,
    SUM(b.TotalAmount) AS TotalBillingLastMonth
FROM Departments d
LEFT JOIN Doctors doc ON d.DepartmentID = doc.DepartmentID
LEFT JOIN Staff s ON d.DepartmentID = s.DepartmentID
LEFT JOIN Admissions a ON d.DepartmentID = (SELECT r.DepartmentID FROM
Rooms r WHERE r.RoomID = a.RoomID LIMIT 1)
    AND a.AdmissionDate >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
LEFT JOIN Billing b ON a.AdmissionID = b.AdmissionID
GROUP BY d.DepartmentID, d.DepartmentName
ORDER BY d.DepartmentName;

-- Patient Billing Summary --
SELECT p.PatientID,CONCAT(p.FirstName, ' ', p.LastName) AS PatientName,
    COUNT(b.BillID) AS NumberOfBills,SUM(b.TotalAmount) AS TotalBilled,
    SUM(b.PaidAmount) AS TotalPaid,SUM(b.TotalAmount) -
SUM(b.PaidAmount) AS BalanceDue,
    MAX(b.DueDate) AS LatestDueDate
FROM Patients p
LEFT JOIN Billing b ON p.PatientID = b.PatientID
GROUP BY p.PatientID, p.FirstName, p.LastName
HAVING  TotalBilled > 0
ORDER BY BalanceDue DESC;

-- Doctor Appointment Analysis --
SELECT d.DoctorID,CONCAT(d.FirstName, ' ', d.LastName) AS DoctorName,
dep.DepartmentName,COUNT(a.AppointmentID) AS TotalAppointments,
    SUM(CASE WHEN a.Status = 'Completed' THEN 1 ELSE 0 END) AS
CompletedAppointments,
    SUM(CASE WHEN a.Status = 'Cancelled' THEN 1 ELSE 0 END) AS
CancelledAppointments,
    SUM(CASE WHEN a.Status = 'No-Show' THEN 1 ELSE 0 END) AS
NoShowAppointments,
    COUNT(a.AppointmentID) * 100.0 / (SELECT COUNT(*)
        FROM Appointments
        WHERE DoctorID = d.DoctorID
        AND AppointmentDate >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
    ) AS PercentageOfTotal
FROM Doctors d
JOIN Departments dep ON d.DepartmentID = dep.DepartmentID
LEFT JOIN Appointments a ON d.DoctorID = a.DoctorID AND
a.AppointmentDate >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
GROUP BY d.DoctorID, d.FirstName, d.LastName, dep.DepartmentName
ORDER BY TotalAppointments DESC;

-- Medication Prescription Report --
SELECT p.PatientID,CONCAT(p.FirstName, ' ', p.LastName) AS PatientName,
    CONCAT(d.FirstName, ' ', d.LastName) AS
PrescribingDoctor,mr.DateCreated AS PrescriptionDate,
    mr.Diagnosis,mr.Prescription
FROM MedicalRecords mr
JOIN Patients p ON mr.PatientID = p.PatientID
JOIN Doctors d ON mr.DoctorID = d.DoctorID
WHERE mr.Prescription IS NOT NULL AND mr.DateCreated >=
DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
ORDER BY mr.DateCreated DESC;

--  Room Utilization Report --
SELECT r.RoomType,COUNT(r.RoomID) AS TotalRooms,
    SUM(CASE WHEN r.Status = 'Occupied' THEN 1 ELSE 0 END) AS
OccupiedRooms,
    SUM(CASE WHEN r.Status = 'Available' THEN 1 ELSE 0 END) AS
AvailableRooms,
    SUM(CASE WHEN r.Status = 'Maintenance' THEN 1 ELSE 0 END) AS
MaintenanceRooms,
    (SUM(CASE WHEN r.Status = 'Occupied' THEN 1 ELSE 0 END) * 100.0 /
COUNT(r.RoomID)) AS OccupancyRate
FROM Rooms r
GROUP BY r.RoomType
ORDER BY OccupancyRate DESC;

-- Staff Salary Analysis by Department --
SELECT d.DepartmentName,COUNT(s.StaffID) AS NumberOfStaff,
    MIN(s.Salary) AS MinSalary,MAX(s.Salary) AS MaxSalary,
    AVG(s.Salary) AS AvgSalary,SUM(s.Salary) AS TotalSalary
FROM Staff s
LEFT JOIN Departments d ON s.DepartmentID = d.DepartmentID
GROUP BY d.DepartmentName
UNION
SELECT 'Administration' AS DepartmentName,COUNT(s.StaffID) AS
NumberOfStaff,
    MIN(s.Salary) AS MinSalary,MAX(s.Salary) AS MaxSalary,
    AVG(s.Salary) AS AvgSalary,SUM(s.Salary) AS TotalSalary
FROM Staff s
WHERE s.DepartmentID IS NULL
ORDER BY TotalSalary DESC;

-- Patient Admission Trends --
SELECT YEAR(AdmissionDate) AS Year, MONTH(AdmissionDate) AS Month,
    COUNT(a.AdmissionID) AS NumberOfAdmissions,
AVG(DATEDIFF(COALESCE(a.DischargeDate, CURDATE()), a.AdmissionDate)) AS
AvgLengthOfStay,
    SUM(b.TotalAmount) AS TotalBillingAmount
FROM Admissions a
LEFT JOIN Billing b ON a.AdmissionID = b.AdmissionID
GROUP BY YEAR(AdmissionDate), MONTH(AdmissionDate)
ORDER BY Year, Month;

ALTER TABLE Appointments
ADD CONSTRAINT FK_Appointments_Patient
FOREIGN KEY (PatientID) REFERENCES Patients(PatientID);

-- Stored Procedure --
-- Schedule New Appointment --
DELIMITER $$
CREATE PROCEDURE sp_ScheduleAppointment (
    IN p_PatientID INT,
    IN p_DoctorID INT,
    IN p_AppointmentDate DATETIME,
    IN p_Purpose VARCHAR(200)
)
BEGIN
    IF EXISTS (
        SELECT 1 FROM Appointments
        WHERE DoctorID = p_DoctorID
        AND ABS(TIMESTAMPDIFF(MINUTE, AppointmentDate,
p_AppointmentDate)) < 30
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Doctor already has an appointment within 30
minutes of this time.';
    ELSE
        INSERT INTO Appointments (PatientID, DoctorID, AppointmentDate,
Purpose)
        VALUES (p_PatientID, p_DoctorID, p_AppointmentDate, p_Purpose);

        SELECT 'Appointment scheduled successfully.' AS Result;
    END IF;
END $$
DELIMITER ;

CALL sp_ScheduleAppointment(1, 2, '2025-08-02 10:30:00', 'General
Checkup');

-- Admit Patients --
DELIMITER //
CREATE PROCEDURE sp_AdmitPatient (
    IN p_PatientID INT,
    IN p_RoomID INT,
    IN p_Reason VARCHAR(200)
)
BEGIN
    DECLARE v_RoomStatus VARCHAR(20);
    SELECT Status INTO v_RoomStatus FROM Rooms WHERE RoomID = p_RoomID;
    IF v_RoomStatus != 'Available' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Selected room is
not available for admission.';
    ELSE
        UPDATE Rooms SET Status = 'Occupied' WHERE RoomID = p_RoomID;
        INSERT INTO Admissions (PatientID, RoomID, AdmissionDate,
Reason)
        VALUES (p_PatientID, p_RoomID, NOW(), p_Reason);

        SELECT 'Patient admitted successfully.' AS Result;
    END IF;
END //
DELIMITER ;

CALL sp_AdmitPatient(1, 3, 'Severe headache and dizziness');

-- Generate Patient Bill --
DELIMITER $$
CREATE PROCEDURE sp_GeneratePatientBill (
    IN p_PatientID INT,
    IN p_AdmissionID INT,
    IN p_TotalAmount DECIMAL(10,2),
    IN p_PaymentMethod VARCHAR(50)
)
BEGIN
    DECLARE v_DueDate DATE;
    SET v_DueDate = DATE_ADD(CURDATE(), INTERVAL 30 DAY);
    INSERT INTO Billing (
        PatientID,
        AdmissionID,
        TotalAmount,
        BillDate,
        DueDate,
        PaymentMethod
    )
    VALUES (
        p_PatientID,
        p_AdmissionID,
        p_TotalAmount,
        NOW(),
        v_DueDate,
        p_PaymentMethod
    );
    SELECT 'Bill generated successfully.' AS Result, LAST_INSERT_ID()
AS BillID;
END$$
DELIMITER ;

CALL sp_GeneratePatientBill(1, 3, 7500.00, 'Credit Card');

-- Discharge Patient --
DELIMITER //
CREATE PROCEDURE sp_DischargePatient (
    IN p_AdmissionID INT
)
BEGIN
    DECLARE v_RoomID INT DEFAULT NULL;
    DECLARE v_PatientID INT DEFAULT NULL;
    SELECT RoomID, PatientID INTO v_RoomID, v_PatientID
    FROM Admissions
    WHERE AdmissionID = p_AdmissionID AND Status = 'Admitted';
    IF v_RoomID IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Admission record not found or patient
already discharged.';
    ELSE
        UPDATE Admissions
        SET DischargeDate = NOW(), Status = 'Discharged'
        WHERE AdmissionID = p_AdmissionID;
        UPDATE Rooms
        SET Status = 'Available'
        WHERE RoomID = v_RoomID;
        SELECT 'Patient discharged successfully.' AS Result;
    END IF;
END //
DELIMITER ;

CALL sp_DischargePatient(9);


-- Views --
-- Active Patients View  --
CREATE VIEW vw_ActivePatients AS
SELECT p.PatientID,CONCAT(p.FirstName, ' ', p.LastName) AS PatientName,
    p.Phone,p.Email,a.AdmissionDate,r.RoomNumber,d.DepartmentName,a.Rea
son
FROM Patients p
JOIN Admissions a ON p.PatientID = a.PatientID AND a.Status =
'Admitted'
JOIN Rooms r ON a.RoomID = r.RoomID
LEFT JOIN Departments d ON r.DepartmentID = d.DepartmentID;
```

```
SELECT * FROM vw_ActivePatients;

-- Doctor Schedule View --
CREATE VIEW vw_DoctorSchedule AS
SELECT d.DoctorID,CONCAT(d.FirstName, ' ', d.LastName) AS DoctorName,
    dep.DepartmentName,a.AppointmentDate,
    CONCAT(p.FirstName, ' ', p.LastName) AS PatientName,
    a.  Purpose,a.Status
FROM Doctors d
JOIN Departments dep ON d.DepartmentID = dep.DepartmentID
LEFT JOIN Appointments a ON d.DoctorID = a.DoctorID
LEFT JOIN Patients p ON a.PatientID = p.PatientID
WHERE a.AppointmentDate >= CURDATE();

SELECT * FROM vw_DoctorSchedule;

--  Financial Summary View --
CREATE VIEW vw_FinancialSummary AS
SELECT YEAR(b.BillDate) AS Year, MONTH(b.BillDate) AS Month,
    COUNT(b.BillID) AS NumberOfBills, SUM(b.TotalAmount) AS
TotalBilled,
    SUM(b.PaidAmount) AS TotalPaid,SUM(b.TotalAmount) -
SUM(b.PaidAmount) AS Outstanding,
    (SUM(b.PaidAmount) * 100.0 / NULLIF(SUM(b.TotalAmount), 0)) AS
CollectionRate
FROM Billing b
GROUP BY YEAR(b.BillDate), MONTH(b.BillDate);

SELECT * FROM vw_FinancialSummary;
```