Team 6

Veetrag, Sam, Amitesh

Assignment – Design

```sql
-- Create Database

CREATE DATABASE IF NOT EXISTS HospitalManagementSystem;

USE HospitalManagementSystem;


-- Table: Role

CREATE TABLE Role (

    RoleID INT PRIMARY KEY AUTO_INCREMENT,

    RoleName VARCHAR(50) NOT NULL UNIQUE,

    Description TEXT

);


-- Table: User

CREATE TABLE User (

    UserID INT PRIMARY KEY AUTO_INCREMENT,

    FirstName VARCHAR(50) NOT NULL,

    LastName VARCHAR(50) NOT NULL,

    Username VARCHAR(50) UNIQUE NOT NULL,

    Password VARCHAR(255) NOT NULL,

    Email VARCHAR(100) UNIQUE NOT NULL,

    RoleID INT,

    AccountStatus ENUM('Active', 'Inactive', 'Suspended') DEFAULT 'Active',

    RegistrationDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    LastLogin TIMESTAMP NULL,

    FOREIGN KEY (RoleID) REFERENCES Role(RoleID) ON DELETE CASCADE

);
```

```sql
-- Table: LoginAttempt
CREATE TABLE LoginAttempt (
    LoginID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    LoginTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    SuccessStatus BOOLEAN,
    IPAddress VARCHAR(45),
    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE
);

-- Table: PasswordReset
CREATE TABLE PasswordReset (
    ResetToken VARCHAR(255) PRIMARY KEY,
    UserID INT,
    ExpirationDate TIMESTAMP NOT NULL,
    ResetStatus ENUM('Pending', 'Used', 'Expired') DEFAULT 'Pending',
    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE
);

-- Table: Patient
CREATE TABLE Patient (
    PatientID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT NOT NULL,
    DateOfBirth DATE NOT NULL,
    Gender ENUM('Male', 'Female', 'Other'),
    Address TEXT,
    ContactNumber VARCHAR(15),
    EmergencyContact VARCHAR(100),
    BloodType VARCHAR(5),
    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE
```

```sql
);


-- Table: MedicalRecord
CREATE TABLE MedicalRecord (

    RecordID INT PRIMARY KEY AUTO_INCREMENT,

    PatientID INT NOT NULL,

    DateCreated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    LastUpdated TIMESTAMP NULL,

    MedicalHistory TEXT,

    Allergies TEXT,

    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE
);


-- Table: Department
CREATE TABLE Department (

    DepartmentID INT PRIMARY KEY AUTO_INCREMENT,

    DepartmentName VARCHAR(100) NOT NULL UNIQUE,

    Location VARCHAR(100),

    ManagerID INT NULL,

    FOREIGN KEY (ManagerID) REFERENCES User(UserID) ON DELETE SET NULL
);


-- Table: Provider
CREATE TABLE Provider (

    ProviderID INT PRIMARY KEY AUTO_INCREMENT,

    UserID INT NOT NULL,

    Specialization VARCHAR(100),

    LicenseNumber VARCHAR(50) UNIQUE,

    DepartmentID INT,

    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE,
```

```sql
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID) ON DELETE SET NULL
);


-- Table: DutyRoster
CREATE TABLE DutyRoster (
    RosterID INT PRIMARY KEY AUTO_INCREMENT,
    ProviderID INT NOT NULL,
    ShiftDate DATE NOT NULL,
    ShiftStart TIME NOT NULL,
    ShiftEnd TIME NOT NULL,
    Status ENUM('Scheduled', 'On-Duty', 'Completed', 'Cancelled') DEFAULT 'Scheduled',
    FOREIGN KEY (ProviderID) REFERENCES Provider(ProviderID) ON DELETE CASCADE
);
-- Table: Room
CREATE TABLE Room (
    RoomID INT PRIMARY KEY AUTO_INCREMENT,
    RoomNumber VARCHAR(10) NOT NULL UNIQUE,
    RoomType VARCHAR(50),
    Floor INT,
    Status ENUM('Available', 'Occupied', 'Maintenance') DEFAULT 'Available'
);


-- Table: AppointmentType
CREATE TABLE AppointmentType (
    TypeID INT PRIMARY KEY AUTO_INCREMENT,
    TypeName VARCHAR(50) NOT NULL UNIQUE,
    Duration INT, -- Duration in minutes
    Description TEXT
);
```

```sql
-- Table: Appointment
CREATE TABLE Appointment (
    AppointmentID INT PRIMARY KEY AUTO_INCREMENT,
    PatientID INT NOT NULL,
    ProviderID INT NOT NULL,
    RoomID INT,
    TypeID INT,
    AppointmentDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    Status ENUM('Scheduled', 'Checked-In', 'Completed', 'Cancelled') DEFAULT 'Scheduled',
    Notes TEXT,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE,
    FOREIGN KEY (ProviderID) REFERENCES Provider(ProviderID) ON DELETE CASCADE,
    FOREIGN KEY (RoomID) REFERENCES Room(RoomID) ON DELETE SET NULL,
    FOREIGN KEY (TypeID) REFERENCES AppointmentType(TypeID) ON DELETE SET NULL
);
-- Table: Medication
CREATE TABLE Medication (
    MedicationID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL UNIQUE,
    GenericName VARCHAR(100),
    Category VARCHAR(50),
    Manufacturer VARCHAR(100),
    UnitPrice DECIMAL(10, 2) NOT NULL
);


-- Table: Inventory
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY AUTO_INCREMENT,
    MedicationID INT NOT NULL,
```

```sql
    BatchNumber VARCHAR(50),

    StockLevel INT NOT NULL,

    ExpiryDate DATE NOT NULL,

    Location VARCHAR(50),

    FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID) ON DELETE CASCADE
);


-- Table: Prescription
CREATE TABLE Prescription (

    PrescriptionID INT PRIMARY KEY AUTO_INCREMENT,

    PatientID INT NOT NULL,

    ProviderID INT NOT NULL,

    PrescriptionDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    Status ENUM('Active', 'Completed', 'Cancelled') DEFAULT 'Active',

    Notes TEXT,

    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE,

    FOREIGN KEY (ProviderID) REFERENCES Provider(ProviderID) ON DELETE CASCADE
);


-- Table: PrescriptionDetail
CREATE TABLE PrescriptionDetail (

    DetailID INT PRIMARY KEY AUTO_INCREMENT,

    PrescriptionID INT NOT NULL,

    MedicationID INT NOT NULL,

    Dosage VARCHAR(50),

    Frequency VARCHAR(50),

    Duration INT, -- Duration in days

    Quantity INT,

    Instructions TEXT,

    FOREIGN KEY (PrescriptionID) REFERENCES Prescription(PrescriptionID) ON DELETE CASCADE,
```

```sql
    FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID) ON DELETE CASCADE
);
-- Table: LabType
CREATE TABLE LabType (
    LabTypeID INT PRIMARY KEY AUTO_INCREMENT,
    TypeName VARCHAR(100) NOT NULL UNIQUE,
    Description TEXT,
    ProcessingTime INT -- Processing time in hours
);


-- Table: LabTest
CREATE TABLE LabTest (
    LabTestID INT PRIMARY KEY AUTO_INCREMENT,
    PatientID INT NOT NULL,
    ProviderID INT NOT NULL,
    LabTypeID INT NOT NULL,
    OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Status ENUM('Ordered', 'Sample-Collected', 'Processing', 'Completed', 'Cancelled') DEFAULT 'Ordered',
    Priority ENUM('Routine', 'Urgent', 'Emergency') DEFAULT 'Routine',
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE,
    FOREIGN KEY (ProviderID) REFERENCES Provider(ProviderID) ON DELETE CASCADE,
    FOREIGN KEY (LabTypeID) REFERENCES LabType(LabTypeID) ON DELETE CASCADE
);


-- Table: LabResult
CREATE TABLE LabResult (
    ResultID INT PRIMARY KEY AUTO_INCREMENT,
    LabTestID INT NOT NULL,
    ResultDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ResultValue TEXT,
```

```sql
    ReferenceRange VARCHAR(100),

    Interpretation TEXT,

    TechnicianID INT,

    FOREIGN KEY (LabTestID) REFERENCES LabTest(LabTestID) ON DELETE CASCADE,

    FOREIGN KEY (TechnicianID) REFERENCES Provider(ProviderID) ON DELETE CASCADE

);
-- Table: Insurance
CREATE TABLE Insurance (

    InsuranceID INT PRIMARY KEY AUTO_INCREMENT,

    ProviderName VARCHAR(100) NOT NULL UNIQUE,

    ContactInfo TEXT,

    PolicyDetails TEXT

);


-- Table: PatientInsurance
CREATE TABLE PatientInsurance (

    PatientInsuranceID INT PRIMARY KEY AUTO_INCREMENT,

    PatientID INT NOT NULL,

    InsuranceID INT NOT NULL,

    PolicyNumber VARCHAR(50) NOT NULL UNIQUE,

    StartDate DATE NOT NULL,

    EndDate DATE,

    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE,

    FOREIGN KEY (InsuranceID) REFERENCES Insurance(InsuranceID) ON DELETE CASCADE

);


-- Table: Bill
CREATE TABLE Bill (

    BillID INT PRIMARY KEY AUTO_INCREMENT,

    PatientID INT NOT NULL,
```

```sql
    GeneratedDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    DueDate DATE NOT NULL,

    TotalAmount DECIMAL(10, 2) NOT NULL,

    Status ENUM('Pending', 'Paid', 'Overdue', 'Cancelled') DEFAULT 'Pending',

    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE

);


-- Table: PaymentMethod
CREATE TABLE PaymentMethod (

    MethodID INT PRIMARY KEY AUTO_INCREMENT,

    MethodName VARCHAR(50) NOT NULL UNIQUE,

    Description TEXT

);


-- Table: Payment
CREATE TABLE Payment (

    PaymentID INT PRIMARY KEY AUTO_INCREMENT,

    BillID INT NOT NULL,

    Amount DECIMAL(10, 2) NOT NULL,

    PaymentDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    MethodID INT NOT NULL,

    TransactionReference VARCHAR(100) UNIQUE,

    Status ENUM('Pending', 'Completed', 'Failed') DEFAULT 'Pending',

    FOREIGN KEY (BillID) REFERENCES Bill(BillID) ON DELETE CASCADE,

    FOREIGN KEY (MethodID) REFERENCES PaymentMethod(MethodID) ON DELETE CASCADE

);
-- Table: NotificationType
CREATE TABLE NotificationType (

    TypeID INT PRIMARY KEY AUTO_INCREMENT,

    TypeName VARCHAR(50) NOT NULL UNIQUE,
```

```sql
    Description TEXT,

    Template TEXT

);


-- Table: Notification

CREATE TABLE Notification (

    NotificationID INT PRIMARY KEY AUTO_INCREMENT,

    TypeID INT NOT NULL,

    UserID INT NOT NULL,

    Title VARCHAR(100) NOT NULL,

    Message TEXT NOT NULL,

    SentDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    ReadDate TIMESTAMP NULL,

    Status ENUM('Pending', 'Sent', 'Read', 'Failed') DEFAULT 'Pending',

    FOREIGN KEY (TypeID) REFERENCES NotificationType(TypeID) ON DELETE CASCADE,

    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE

);

-- Table: Survey

CREATE TABLE Survey (

    SurveyID INT PRIMARY KEY AUTO_INCREMENT,

    Title VARCHAR(100) NOT NULL UNIQUE,

    Description TEXT,

    StartDate DATE NOT NULL,

    EndDate DATE NOT NULL,

    Status ENUM('Draft', 'Active', 'Closed') DEFAULT 'Draft'

);


-- Table: SurveyResponse

CREATE TABLE SurveyResponse (

    ResponseID INT PRIMARY KEY AUTO_INCREMENT,
```

```sql
    SurveyID INT NOT NULL,

    PatientID INT NOT NULL,

    ResponseDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    Responses TEXT NOT NULL,

    FOREIGN KEY (SurveyID) REFERENCES Survey(SurveyID) ON DELETE CASCADE,

    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE
);


-- Table: Event
CREATE TABLE Event (

    EventID INT PRIMARY KEY AUTO_INCREMENT,

    EventType ENUM('Reminder', 'Follow-up', 'Survey', 'Alert') NOT NULL,

    RelatedID INT, -- Can reference other table IDs like AppointmentID or SurveyID

    UserID INT NOT NULL,

    EventDate TIMESTAMP NOT NULL,

    Description TEXT,

    Status ENUM('Scheduled', 'Triggered', 'Completed', 'Cancelled') DEFAULT 'Scheduled',

    FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE
);
```