0. Database Creation

In [1]:
```
CREATE DATABASE Hospital_System;
```

Commands completed successfully.
Total execution time: 00:00:00.207

In [2]:
```
USE Hospital_System;
```

Commands completed successfully.
Total execution time: 00:00:00.002

1. Inserting Basic Entities

NOTE: Name (FirstName in composite name attributes), Salary, Rule (in case of Employee entity) are considered elementary business requirements for every entity, hence NOT NULL constraint.

In [3]:
```
CREATE TABLE Department (
    DepID INT PRIMARY KEY,
    DepName VARCHAR(100) NOT NULL,
    ManagerID INT NOT NULL
    -- FK in 1:1 T:P relationship with Doctor
);

CREATE TABLE Doctor (
    DocID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    SecondName VARCHAR(50),
    ThirdName VARCHAR(50),
    Salary DECIMAL(10, 2) NOT NULL,
    SupervisorID INT
    -- FK in M:1 T:P unary (recursive) relationship
);

CREATE TABLE Clinic (
    ClinicID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Zcode VARCHAR(5),
    Street VARCHAR(100),
    City VARCHAR(25),
    DocID INT NOT NULL
    -- FK in 1:1 T:P relationship with Doctor
);

CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    SecondName VARCHAR(50),
    ThirdName VARCHAR(50),
    Salary DECIMAL(10, 2) NOT NULL,
    Emp_Rule VARCHAR(50) NOT NULL
);

CREATE TABLE Patient (
    PatientID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    SecondName VARCHAR(50),
    ThirdName VARCHAR(50),
    Gender CHAR(1) CHECK ( Gender IN ('M', 'F') ),
    Phone VARCHAR(20),
    City VARCHAR(25)
);

CREATE TABLE Room (
    RoomID INT PRIMARY KEY,
    Capacity INT CHECK (Capacity BETWEEN 1 AND 4),
    Type VARCHAR(50),
    Availability VARCHAR(20) Check (Availability in ('Available', 'Not Available')) DEFAULT 'Available',
    DepID INT NOT NULL -- FK in M:1 T:T relationship with Department

);

CREATE TABLE Nurse (
    NurseID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    SecondName VARCHAR(50),
    ThirdName VARCHAR(50),
    Salary DECIMAL(10, 2) NOT NULL
```

```
);

CREATE TABLE Appointment (
    AppoID INT PRIMARY KEY,
    Status VARCHAR(50),
    ClinicID INT NOT NULL
    -- FK in M:1 T:P relationship with Clinic
);
```

Commands completed successfully.

Total execution time: 00:00:00.021

2. Adding FK Constraints (to 2-table-mapped relationships)

In [4]:
```
-- Doctor_Doctor: M:1 T:P
ALTER TABLE Doctor
ADD FOREIGN KEY (SupervisorID) REFERENCES Doctor(DocID);

-- Department_Doctor (managed by): 1:1 T:P
ALTER TABLE Department
ADD FOREIGN KEY (ManagerID) REFERENCES Doctor(DocID);

-- Clinic_Doctor: 1:1 T:P
ALTER TABLE Clinic
ADD FOREIGN KEY (DocID) REFERENCES Doctor(DocID);

-- Room_Department: M:1 T:T
ALTER TABLE Room
ADD FOREIGN KEY (DepID) REFERENCES Department(DepID);

-- Appointment_Clinic: M:1 T:P
ALTER TABLE Appointment
ADD FOREIGN KEY (ClinicID) REFERENCES Clinic(ClinicID);
```

Commands completed successfully.

Total execution time: 00:00:00.010

3. Adding 3-Table-Mapped Relationships (along with respective FK constraints)

In [5]:
```
-- Employee_Department: M:1 0:1
CREATE TABLE Emp_Dep (
    EmpID INT,
    DepID INT,
    StartingDate DATE,
    PRIMARY KEY (EmpID, DepID)
);
ALTER TABLE Emp_Dep
ADD FOREIGN KEY (EmpID) REFERENCES Employee(EmpID),
    FOREIGN KEY (DepID) REFERENCES Department(DepID);

-- Doctor_Department (assigned to): M:1 0:1
CREATE TABLE Doc_Dep (
    DocID INT,
    DepID INT,
    StartingDate DATE,
    PRIMARY KEY (DocID, DepID)
);
ALTER TABLE Doc_Dep
ADD FOREIGN KEY (DocID) REFERENCES Doctor(DocID),
    FOREIGN KEY (DepID) REFERENCES Department(DepID);

-- Patient_Room: M:1 0:0
CREATE TABLE Patient_Room (
    PatientID INT,
    RoomID INT,
    StartingDate DATE,
    stayingTime INT,
    PRIMARY KEY (PatientID, RoomID)
);
ALTER TABLE Patient_Room
ADD FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (RoomID) REFERENCES Room(RoomID);

-- Room_Nurse: M:M 1:1
CREATE TABLE Room_Nurse (
    RoomID INT,
        NurseID INT,
        PRIMARY KEY (RoomID, NurseID)
);
ALTER TABLE Room_Nurse
```

```
ADD FOREIGN KEY (RoomID) REFERENCES Room(RoomID),
    FOREIGN KEY (NurseID) REFERENCES Nurse(NurseID);
```

Commands completed successfully.

Total execution time: 00:00:00.016

4. Add Ternary Relationship

In [6]:
```
CREATE TABLE Appointment_Patient_Doc (
    AppoID INT,
    PatientID INT,
    DocID INT,
    Date DATE,
    Cost DECIMAL(10, 2),
    PRIMARY KEY (AppoID, PatientID, DocID)
);
ALTER TABLE Appointment_Patient_Doc
ADD FOREIGN KEY (AppoID) REFERENCES Appointment(AppoID),
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (DocID) REFERENCES Doctor(DocID);
```

Commands completed successfully.

Total execution time: 00:00:00.004