

BL

BO LI

INSTRUCTOR: DR. EHAT ERCANLI

## CONTACT

---

NetID: bli158  
SUID: 218717316  
Email: bli158@syr.edu

## STATES MD URGENT CARE CENTER DATABASE

---

CSE 581 PROJECT 2

### ABSTRACT

The “States MD Urgent Care Center Database” project is to manage data and resources for the Urgent Care Center such as employees, patients, centers, billing, etc. The database design will provide an efficient data system for reporting and retrieving data.

The report will describe details about how to design, implement and test the database for the Urgent Care Center.

## Table of Contents

Introduction .....	3
Design.....	3
Necessary Information .....	3
Design Considerations .....	5
E/R Diagram .....	8
Third Normal Form .....	8
Implementation .....	10
Create Database .....	10
Create Tables .....	10
Insert data into tables .....	25
Records .....	37
Testing.....	39
Views .....	39
Stored Procedures .....	42
User Defined Functions .....	44
Triggers .....	46
Transactions .....	48
Scripts .....	50
Conclusion .....	52
Analysis .....	53
Problem Encountered .....	53
Remarks .....	53

## INTRODUCTION

Due to COVID19, more and more patients go to the urgent care center for testing. Besides, other types of patients are getting more to present in the urgent care center as well. Due to lack of data management such as centers' information, patients' information, etc., it's necessary to have a database to store all such data so that the urgent care center could easily view and manage them whenever they want. Also, urgent care center could monitor all patients' situation and give them early treatment based on patient's health history.

The project "States MD Urgent Care Center Database" is to manage data and resources for the Urgent Care Center such as employees, patients, centers, billing, etc.

25 tables are created to build the structure of the database. There are 5 important parts: Centers, Employees, Patients, PatientHealthHistory, and Insurance. The detailed information about the structure and relationships will be discussed in the Design session. In the database, I will implement it with random data, and test it with operations such as stored procedures, functions, triggers, transactions and scripts.

## DESIGN

### A. Necessary Information

#### 1. Determine tables with primary keys, foreign keys and nullabilities.

Table	PK	FK
Centers	CenterID	
Facilities	RoomID, CenterID	CenterID
MedicalEquipments	EquipmentID, CenterID	CenterID
Departments	DeptID	CenterID
Offices	OfficeID	DeptID
Employees	EmployeeID	CenterID, DeptID, OfficeID
Physicians	PhysicianID	EmployeeID
Attending	PatientID, PhysicianID	PatientID, PhysicianID
PatientHealthHistory	PatientID	TransferInfo
Vitals	PatientID	PatientID
Discharge	PatientID	PatientID
EmergencyDept	EmergDeptID	
Costs	PatientID, MedicationID	PatientID, MedicationID
Medications	MedicationID	
VisitHospital	PatientID, HospitalID	PatientID, HospitalID
Hospitals	HospitalID	
Refer	PatientID, DoctorID	PatientID, DoctorID
Doctors	DoctorID	HospitalID
PrimaryCareDoc	PatientID	PatientID, DoctorID
Patients	PatientID	

Billing	BillID	BillID
Testing	TestingID	TestingID
HealthInsurance	HealthInsuranceID	PatientID
Coverages	HealthInsuranceID	HealthInsuranceID
InsuranceCompany	HealthInsuranceID	HealthInsuranceID

## 2. Data Types:

Tables	int	money	smalldatetime
Centers	CenterID, ProcedureCapacity		
Facilities	RoomID, CenterID		
MedcalEquipments	EquipmentID, CenterID		
Departments	DeptID, CenterID		
Offices	OfficeID, DeptID, OfficeNum		
Employees	EmployeeID, CenterID, DeptID, OfficeID	Salary	
Physicians	PhysicianID, EmployeeID		
Attending	PatientID, PhysicianID		Date
PatientHealthHistory	PatientID, TransferInfo		
Vitals	PatientID		
Discharge	PatientID		ExpectTime
EmergencyDept	EmergDeptID		
Costs	PatientID, MedicationID		
Medications	MedicationID	Price	
VisitHospital	PatientID, HospitalID		Date
Hospitals	HospitalID		
Refer	PatientID, DoctorID		Date
Doctors	DoctorID, HospitalID		
PrimaryCareDoc	PatientID, DoctorID		
Patients	PatientID		
Billing	BillID	TotalCharge	
Testing	TestingID		
HealthInsurance	HealthInsuranceID, PatientID	Cost	
Coverages	HealthInsuranceID	CoveragePrice	
InsuranceCompany	HealthInsuranceID		

Note\*: All other attributes are VARCHAR

### 3. Nullabilities:

Table	Nullability Columns
Employees	apt, PastJob, Benefits
Hospitals	Phone, Website
Doctors	Phone, Fax, Email, Background
EmergencyDept	Location
Patients	apt, Email
PatientHealthHistory	TransferInfo
Vitals	BodyTemperature, PulseRate, RespirationRate, BloodPressure
PrimaryCareDoc	DoctorID

## B. Design Considerations

As shown below, the data we need to store in the system includes detailed information of centers, employees, physicians, patients, patient health history, health insurance, doctors, medications, and hospitals.

In the database, there are several main parts such as Centers, Employees, Patients, PatientHealthHistory, Testing, Billing, and Insurance details.

1. **Centers**: we need to know the centers' basic information, address, and contact. 10 columns are set in the Centers table: CenterID, CenterName, BusinessHour, ProcedureCapacity, Location, Address, City, State, ZipCode, Phone. I set CenterID as a primary key so that it can be an identity for other tables such as Departments, Facilities, and so on. There are 4 that tables have relationships with it: Departments, Facilities, MedicalEquipments, and Employees.

Table	Relationship
Centers - Departments	One-to-Many
Centers - Facilities	One-to-Many
Centers - MedicalEquipments	One-to-Many
Centers - Employees	One-to-Many

2. **Employees:** we need to know employee's information, job title, salary, the office, department, and center he/she works at, and work schedule.
- 22 columns are set in the Employees table. I set EmployeeID as a primary key to identify each employee as well as physicians. I also set CenterID, DeptID, and OfficeID as Foreign keys to reference where employees work at.
- There are 4 tables that have relationships with it: Offices, Departments, Centers, and Physicians.
- Note that Physicians – Employees is a one-to-one relationship because we could find physicians information easily from the Employees table but will not interrupt the current physician's type.

Table	Relationship
Offices - Employees	One-to-Many
Departments - Employees	One-to-Many
Centers - Employees	One-to-Many
Physicians - Employees	One-to-One

3. **PatientHealthHistory:** we need to know patients' information, their health history records such as vitals, discharge information, the medications they used, and the hospital they visited.
- 9 columns are set in the PatientHealthHistory table. I set PatientID as a primary key referred to PatientID of Patients table since I consider one patient only has one or zero health history record so that urgent care center could easily make a decision how to treat the patient.
- In this table, to reduce duplication, I create tables such as Vitals, Discharge to store corresponding data.
- There are 8 tables that have relationships with it: Patients, EmergencyDept, Physicians, Vitals, Discharge, Medications, Hospitals, Doctors. However, there are some linking tables link to it such as Attending, Refer, VisitHospital, Costs that are Many-to-Many relationships.

Table	Relationship
PatientHealthHistory - Patients	One-to-One
PatientHealthHistory - EmergencyDept	One-to-Many
PatientHealthHistory - Physicians	Many-to-Many
PatientHealthHistory - Vitals	One-to-One
PatientHealthHistory - Discharge	One-to-One

PatientHealthHistory - Medications	Many-to-Many
PatientHealthHistory - Hospitals	Many-to-Many
PatientHealthHistory - Doctors	Many-to-Many

4. **HealthInsurance:** we need to know insurance information as if a patient has what kind of insurance and the insurance coverages.

8 columns are set in the HealthInsurance table. I set HealthInsuranceID as a primary key to identify other tables such as InsuranceCompany and Coverages. InsuranceCompany and Coverages have separated tables because of data duplication.

There are 3 tables that have relationships with it: Patients, Coverages, and InsuranceCompany. All of them are a one-to-one relationship with it because we have a unique HealthInsuranceID, and based on the HealthInsuranceID, we could find all data easily.

Table	Relationship
HealthInsurance - Patients	One-to-One
HealthInsurance - Coverages	One-to-One
HealthInsurance - InsuranceCompany	One-to-One

5. **Patients:** we discussed a little bit about patients in the table of PatientHealthHistory.

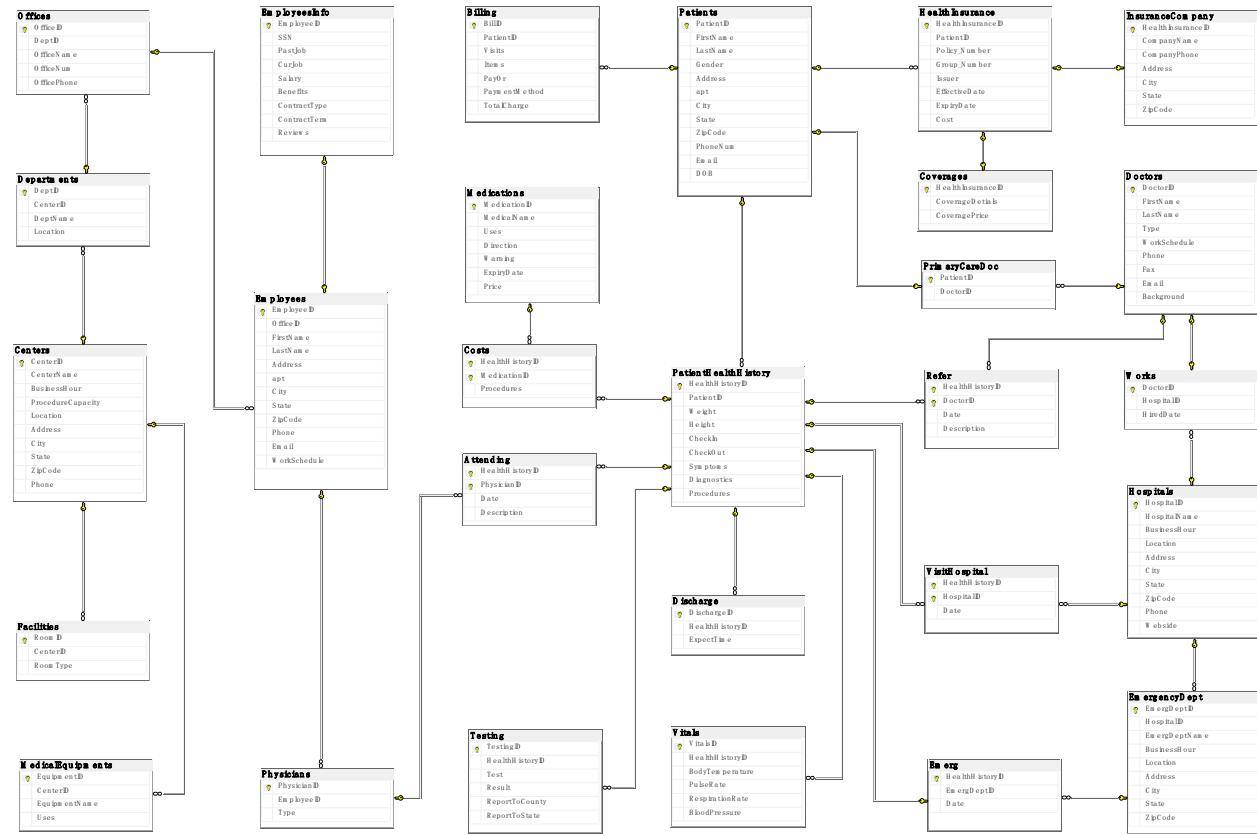
They are identical by PatientID which is a One-to-One relationship. In the table Patients, we could know the patients' information such as names, genders, where they live, contact information, date of birth, and many more from other tables related to it.

12 columns are set in Patients. I set PatientID as a primary key so that other tables can be identified with it.

There are 5 tables that have relationships with it: Billing, Testing, PatientHealthHistory, Doctors, and HealthInsurance. However, table PrimaryCareDoc is a link table between Patients and Doctors.

Table	Relationship
Patients - Billing	One-to-One
Patients - Testing	One-to-One
Patients - PatientHealthHistory	One-to-One
Patients - HealthInsurance	One-to-Many
PrimaryCareDoc - Patients	One-to-Many

## C. E/R Diagram



## D. Third Normal Form

1. Patients(PatientID, FirstName, LastName, Gender, Address, apt, City, State, ZipCode, PhoneNum, Email, DOB)
2. Billing(BillID, Visits, Items, PayOr, PaymentMethod, TotalCharge)
3. Testing(TestingID, Test, Result, ReportToCounty, ReportToState)
4. HealthInsurance(HealthInsuranceID, PatientID, Policy\_Number, Group\_Number, Issuer, EffectiveDate, ExpiryDate, Cost)
5. Coverages(HealthInsuranceID, CoverageDetails, CoveragePrice)
6. InsuranceCompany(HealthInsuranceID, CompanyName, CompanyPhone, Address, City, State, ZipCode)
7. EmergencyDept(EmergDeptID, EmergDeptName, BusinessHour, Location, Address, City, State, ZipCode)
8. PatientHealthHistory(PatientID, Weight, Height, CheckIn, CheckOut, Symptoms, Diagnostics, Procedures, TransferInfo)
9. Discharge(PatientID, ExpectTime)
10. Vitals(PatientID, BodyTemperature, PulseRate, RespirationRate, BloodPressure)
11. Medications(MedicationID, MedicationName, Uses, Direction, Warning, ExpiryDate, Price)
12. Costs(PatientID, MedicationID, Procedures)

13. Hospitals(HospitalID, HospitalName, BusinessHour, Location, Address, City, State, ZipCode, Phone, Webside)
14. VisitHospital(PatientID, HospitalID, Date)
15. Doctors(DoctorID, HospitalID, FirstName, LastName, Type, OperationHour, Phone, Fax, Email, Background)
16. PrimaryCareDoc(PatientID, DoctorID)
17. Refer(PatientID, DoctorID, Date, Description)
18. Centers(CenterID, CenterName, BusinessHour, ProcedureCapacity, Location, Address, City, State, ZipCode, Phone)
19. Facilities(RoomID, CenterID, RoomType)
20. MedicalEquipments(EquipmentID, CenterID, EquipmentName, Uses)
21. Departments(DeptID, CenterID, DeptName, Location)
22. Offices(OfficeID, DeptID, OfficeName, OfficeNum, OfficePhone)
23. Employees(EmployeeID, FirstName, LastName, SSN, Address, apt, City, State, ZipCode, Phone, Email, PastJob, CurJob, Salary, Benefits, ContractType, ContractTerm, Reviews, CenterID, DeptID, OfficeID, WorkSchedule)
24. Physicians(PhysicianID, EmployeeID, Type)
25. Attending(PatientID, PhysicianID, Date, Description)

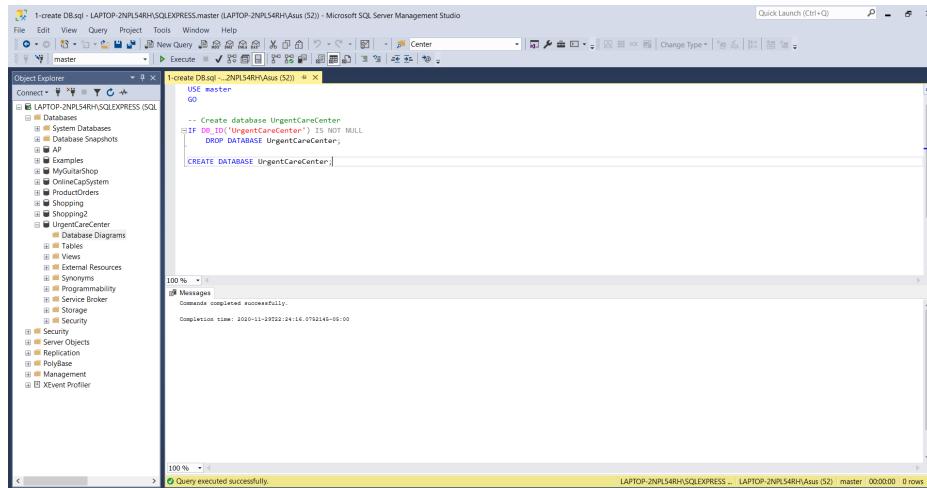
# IMPLEMENTATION

## A. Create Database

```
USE master
GO

-- Create database UrgentCareCenter
IF DB_ID('UrgentCareCenter') IS NOT NULL
DROP DATABASE UrgentCareCenter;

CREATE DATABASE UrgentCareCenter;
```



## B. Create Tables

As shown below, tables are created

```
USE UrgentCareCenter

-- Center Table
CREATE TABLE Centers (
    CenterID           int                      NOT NULL IDENTITY(1,1),
    CenterName         varchar(100)      NOT NULL,
    BusinessHour       varchar(100)      NOT NULL,
    ProcedureCapacity int                      NOT NULL,
    Location           varchar(100)      NOT NULL,
    Address            varchar(50)        NOT NULL,
    City               varchar(50)        NOT NULL,
    State              varchar(50)        NOT NULL,
    ZipCode            varchar(50)        NOT NULL,
    Phone              varchar(50)        NOT NULL,
    CONSTRAINT PK_Center PRIMARY KEY CLUSTERED(CenterID ASC)
);

-- Facility Table
CREATE TABLE Facilities (
    RoomID             int                      NOT NULL,
    CenterID           int                      NOT NULL,
    RoomType           varchar(50)      NOT NULL,
    CONSTRAINT PK_Facility PRIMARY KEY ((CenterID, RoomID)),
);
```

```

        CONSTRAINT FK_Facility_Center    FOREIGN KEY(CenterID)    REFERENCES Centers(CenterID)
);

-- Medical Equipments Table
CREATE TABLE MedicalEquipments (
    EquipmentID          int                                NOT NULL,
    CenterID              int                                NOT NULL,
    EquipmentName         varchar(100)      NOT NULL,
    Uses                  varchar(200)      NOT NULL,
    CONSTRAINT PK_Equipment      PRIMARY KEY      (CenterID, EquipmentID),
    CONSTRAINT FK_Equipment_Center FOREIGN KEY(CenterID) REFERENCES
    Centers(CenterID)
);

-- Departments Table
CREATE TABLE Departments (
    DeptID                int                                NOT NULL,
    CenterIDint           int                                NOT NULL,
    DeptNamevarchar(50)    NOT NULL,
    Locationvarchar(50)    NOT NULL,
    CONSTRAINT PK_Department      PRIMARY KEY
    ASC),                   CLUSTERED(DeptID
    CONSTRAINT FK_Dept_Center      FOREIGN KEY (CenterID) REFERENCES Centers(CenterID)
);

-- Offices Table
CREATE TABLE Offices (
    OfficeIDint           int                                NOT NULL,
    DeptID                int                                NOT NULL,
    OfficeNamevarchar(50)   NOT NULL,
    OfficeNumint           int                                NOT NULL,
    OfficePhonevarchar(50)  NOT NULL,
    CONSTRAINT PK_Office      PRIMARY KEY
    CLUSTERED(OfficeID),
    CONSTRAINT FK_Office_Dept      FOREIGN KEY (DeptID) REFERENCES Departments(DeptID)
);

-- Employees Table
CREATE TABLE Employees (
    EmployeeID             int                                NOT NULL IDENTITY(1,1),
    FirstName               varchar(50)      NOT NULL,
    LastName                varchar(50)      NOT NULL,
    SSN                     varchar(20)      NOT NULL,
    Address                 varchar(100)     NOT NULL,
    apt                     varchar(50)      NULL,
    City                    varchar(50)      NOT NULL,
    State                   varchar(50)      NOT NULL,
    ZipCode                 varchar(50)      NOT NULL,
    Phone                   varchar(50)      NOT NULL,
    Email                   varchar(50)      NOT NULL,
    PastJob                 varchar(50)      NULL,
    CurJob                  varchar(50)      NOT NULL,
    Salary                  money            NOT NULL,
    Benefits                varchar(50)      NULL,
    ContractType            varchar(50)      NOT NULL,
    ContractTerm            varchar(200)     NOT NULL,
    Reviews                 varchar(200)     NOT NULL,
    CenterID                int                                NOT NULL,
    DeptID                 int                                NOT NULL,
    OfficeID                int                                NOT NULL,
    WorkSchedule            varchar(100)     NOT NULL,
    CONSTRAINT Employee_SSN      UNIQUE (SSN),

```

```

CONSTRAINT      PK_Employee          PRIMARY KEY
CLUSTERED(EmployeeID ASC),
CONSTRAINT      FK_Employee_Center   FOREIGN KEY (CenterID) REFERENCES
Centers(CenterID),
CONSTRAINT      FK_Employee_Dept     FOREIGN KEY (DeptID) REFERENCES Departments(DeptID),
CONSTRAINT      FK_Employee_Office   FOREIGN KEY (OfficeID) REFERENCES
Offices(OfficeID),
);

-- Physicians Table
CREATE TABLE Physicians (           -- Physician is an employee
    PhysicianID      int              NOT NULL IDENTITY(1,1),
    EmployeeID       int              NOT NULL UNIQUE,
    Type             varchar(50)      NOT NULL,
CONSTRAINT      PK_Physician        PRIMARY KEY
CLUSTERED(PhysicianID ASC),
CONSTRAINT      FK_Physician_Employee FOREIGN KEY(EmployeeID) REFERENCES
Employees(EmployeeID)
);

-- Hospitals Table
CREATE TABLE Hospitals (
    HospitalID        int              NOT NULL IDENTITY(1,1),
    HospitalName      varchar(50)      NOT NULL,
    BusinessHour      varchar(100)     NOT NULL,
    Location          varchar(200)     NOT NULL,
    Address           varchar(50)      NOT NULL,
    City              varchar(50)      NOT NULL,
    State             varchar(50)      NOT NULL,
    ZipCode           varchar(50)      NOT NULL,
    Phone             varchar(50)      NULL,
    Website           varchar(100)     NULL,
CONSTRAINT      PK_Hospital        PRIMARY KEY
CLUSTERED(HospitalID
ASC),
);

-- Doctors Table
CREATE TABLE Doctors (
    DoctorID          int              NOT NULL IDENTITY(1,1),
    HospitalID        int              NOT NULL,
    FirstName          varchar(50)      NOT NULL,
    LastName           varchar(50)     NOT NULL,
    Type               varchar(50)      NOT NULL,
    OperationHour     varchar(100)     NOT NULL,
    Phone              varchar(50)      NULL,
    Fax                varchar(50)      NULL,
    Email              varchar(50)      NULL,
    Background         varchar(200)     NULL,
CONSTRAINT      PK_Doctor          PRIMARY KEY
CLUSTERED(DoctorID ASC),
CONSTRAINT      FK_Doctor_Hospital FOREIGN KEY(HospitalID) REFERENCES
Hospitals(HospitalID)
);

-- EmergencyDept Table
CREATE TABLE EmergencyDept (
    EmergDeptID        int              NOT NULL IDENTITY(1,1),
    EmergDeptName      varchar(50)      NOT NULL,
    BusinessHour       varchar(100)     NOT NULL,
    Location           varchar(50)      NULL,
    Address            varchar(50)      NOT NULL,
    City               varchar(50)      NOT NULL,
    State              varchar(50)      NOT NULL,
    ZipCode            varchar(50)      NOT NULL,
CONSTRAINT      PK_EmergencyDept PRIMARY KEY
CLUSTERED(EmergDeptID ASC),
);

```

```

);

-- Medication Table
CREATE TABLE Medications (
    MedicationID      INT                                NOT NULL IDENTITY(1,1),          -- FK
    Patient(PatientID)
    MedicalName       varchar(50)                         NOT NULL,
    Uses              varchar(200)                        NOT NULL,
    Direction         varchar(100)                        NOT NULL,
    Warning           varchar(200)                        NOT NULL,
    ExpiryDate        smalldatetime                     NOT NULL,
    Price              money                            NOT NULL,
    CONSTRAINT        PK_Medication                  PRIMARY KEY
    CLUSTERED(MedicationID ASC),
);

-- Patient Table
CREATE TABLE Patients (
    PatientID         int                               NOT NULL IDENTITY(1,1),
    FirstName          varchar(50)                        NOT NULL,
    LastName           varchar(50)                        NOT NULL,
    Gender             varchar(50)                        NOT NULL,
    Address            varchar(100)                       NOT NULL,
    apt                varchar(50)                         NULL,
    City               varchar(50)                        NOT NULL,
    State              varchar(50)                        NOT NULL,
    ZipCode            varchar(50)                        NOT NULL,
    PhoneNum           varchar(50)                        NOT NULL,
    Email              varchar(50)                        NULL,
    DOB                smalldatetime                     NOT NULL,
    CONSTRAINT        PK_Patient                  PRIMARY KEY
    CLUSTERED(PatientID ASC),
);

-- PatientHealthHistory Table
CREATE TABLE PatientHealthHistory (
    PatientID          int                                NOT NULL,          -- FK
    Patient(PatientID)
    Weight              varchar(20)                         NOT NULL,
    Height              varchar(20)                         NOT NULL,
    CheckIn             smalldatetime                    NOT NULL,
    CheckOut            smalldatetime                    NOT NULL,
    Symptoms            varchar(200)                        NOT NULL,
    Diagnostics         varchar(200)                        NOT NULL,
    Procedures          varchar(200)                        NOT NULL,
    TransferInfo        int                               NULL,
    CONSTRAINT        PK_PatientHealthHistory      PRIMARY KEY
    CLUSTERED(PatientID ASC),
    CONSTRAINT        FK_History_Patient        FOREIGN KEY      (PatientID)
    REFERENCES Patients(PatientID),
    CONSTRAINT        FK_History_EmergDept      FOREIGN KEY      (TransferInfo)
    REFERENCES EmergencyDept (EmergDeptID)
);

-- Vitals Table
CREATE TABLE Vitals (
    PatientID          int                                NOT NULL,
    BodyTemperature     varchar(50)                         NULL,
    PulseRate           varchar(50)                         NULL,
    RespirationRate    varchar(50)                         NULL,
    BloodPressure       varchar(50)                         NULL,
    CONSTRAINT        PK_Vital                  PRIMARY KEY
    CLUSTERED(PatientID),
    CONSTRAINT        FK_Vital_Patient        FOREIGN KEY      (PatientID)      REFERENCES
    PatientHealthHistory(PatientID)
);

```

```

);

-- Discharge table
CREATE TABLE Discharge (
    PatientID          int                                NOT NULL,
    ExpectTime         smalldatetime                    NOT NULL,
    CONSTRAINT PK_Discharge      PRIMARY KEY CLUSTERED(PatientID ASC),
    CONSTRAINT FK_History_Discharge FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID)
);

-- Testing
CREATE TABLE Testing (
    TestingID          int                                NOT NULL,
    Test                varchar(200)                      NOT NULL,
    Result              varchar(200)                      NOT NULL,
    ReportToCounty     varchar(50)                       NOT NULL,
    ReportToState      varchar(50)                       NOT NULL,
    CONSTRAINT PK_Testing      PRIMARY KEY CLUSTERED(TestingID ASC),
    CONSTRAINT FK_Testing_Patient FOREIGN KEY(TestingID) REFERENCES Patients(PatientID)
);

-- Billing
CREATE TABLE Billing (
    BillID              int                                NOT NULL,
    Visits              varchar(50)                      NOT NULL,
    Items               varchar(50)                      NOT NULL,
    PayOr               varchar(50)                      NOT NULL,
    PaymentMethod       varchar(50)                      NOT NULL,
    TotalCharge        money                            NOT NULL,
    CONSTRAINT PK_Bill      PRIMARY KEY CLUSTERED(BillID ASC),
    CONSTRAINT FK_Bill_Patient FOREIGN KEY(BillID) REFERENCES Patients(PatientID)
);

-- Health Insurance Table
CREATE TABLE HealthInsurance (
    HealthInsuranceID   int                                NOT NULL IDENTITY(1,1),
    -- PK
    PatientID          int                                NOT NULL,
    -- FK Patients(PatientID)
    Policy_Number       varchar(50)                      NOT NULL,
    Group_Number        varchar(50)                      NOT NULL,
    Issuer              varchar(50)                      NOT NULL,
    EffectiveDate       smalldatetime                   NOT NULL,
    ExpiryDate          smalldatetime                   NOT NULL,
    Cost                money                            NOT NULL,
    CONSTRAINT Policy_Info      UNIQUE (Policy_Number),
    CONSTRAINT PK_HealthInsurance PRIMARY KEY CLUSTERED(HealthInsuranceID ASC),
    CONSTRAINT FK_Patient_Insurance FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);

-- Coverages Table
CREATE TABLE Coverages (
    HealthInsuranceID   int                                NOT NULL,
    -- FK
    HealthInsurance(HealthInsuranceID)                  NOT NULL,
    CoverageDetails     varchar(500)                     NOT NULL,

```

```

CoveragePrice      money          NOT NULL,
CONSTRAINT      PK_Coverage          PRIMARY KEY
                  CLUSTERED(HealthInsuranceID),
CONSTRAINT      FK_Coverage_Insurance FOREIGN KEY      (HealthInsuranceID)
REFERENCES HealthInsurance(HealthInsuranceID)
);

-- InsuranceCompany Table
CREATE TABLE InsuranceCompany (
    HealthInsuranceID      int          NOT NULL,
    -- FK HealthInsurance(HealthInsuranceID)
    CompanyName            varchar(50)   NOT NULL,
    CompanyPhone           varchar(50)   NOT NULL,
    Address                varchar(50)   NOT NULL,
    City                   varchar(50)   NOT NULL,
    State                  varchar(50)   NOT NULL,
    ZipCode                varchar(50)   NOT NULL,

    CONSTRAINT      PK_InsuranceCompany          PRIMARY KEY
                  CLUSTERED(HealthInsuranceID),
    CONSTRAINT      FK_Company_Insurance      FOREIGN KEY      (HealthInsuranceID)
REFERENCES HealthInsurance(HealthInsuranceID)
);

-- Attends Table
CREATE TABLE Attending (          -- Attending Physicians for Patients
    PatientID        int          NOT NULL,
    PhysicianID       int          NOT NULL,
    Date              smalldatetime NOT NULL,
    --Location        varchar(50)   NOT NULL,
    Description        varchar(200)  NOT NULL,

    CONSTRAINT      PK_Attending          PRIMARY KEY      (PatientID,
PhysicianID),
    CONSTRAINT      FK_Attending_Patient      FOREIGN KEY      (PatientID)
REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT      FK_Attending_Physician      FOREIGN KEY      (PhysicianID)
REFERENCES Physicians(PhysicianID),
);

-- Refer Table
CREATE TABLE Refer (
    PatientID        int          NOT NULL,
    DoctorID         int          NOT NULL,
    Date              smalldatetime NOT NULL,
    Description        varchar(200)  NOT NULL,

    CONSTRAINT      PK_Referr          PRIMARY KEY      (PatientID,
DoctorID),
    CONSTRAINT      FK_Referr_Patient      FOREIGN KEY      (PatientID)
REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT      FK_Referr_Doctor      FOREIGN KEY      (DoctorID)
REFERENCES Doctors(DoctorID),
);

-- VisitHospital
CREATE TABLE VisitHospital (
    PatientID        int          NOT NULL,
    HospitalID       int          NOT NULL,
    Date              smalldatetime NOT NULL,

    CONSTRAINT      PK_VisitHospital          PRIMARY KEY      (PatientID,
HospitalID),
    CONSTRAINT      FK_VisitHospital_Patient      FOREIGN KEY      (PatientID)
REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT      FK_VisitHospital_Hospital      FOREIGN KEY      (HospitalID)
REFERENCES Hospitals(HospitalID),
);

```

```

);
-- PrimaryCareDoc Table
CREATE TABLE PrimaryCareDoc (
    PatientID      int          NOT NULL,
    DoctorID       int          NULL,
    CONSTRAINT     PK_PrimaryCareDoc           PRIMARY KEY
                  CLUSTERED(PatientID ASC),
    CONSTRAINT     FK_PrimaryCareDoc_Patient   FOREIGN KEY
                  REFERENCES Patients(PatientID),
    CONSTRAINT     FK_PrimaryCareDoc_Doctor    FOREIGN KEY
                  REFERENCES Doctors(DoctorID)
);

-- Costs Table
CREATE TABLE Costs (
    PatientID      int          NOT NULL,
    MedicationID   int          NOT NULL,
    Procedures     varchar(200)  NOT NULL,
    CONSTRAINT     PK_Cost            PRIMARY KEY
                  (PatientID,
                   MedicationID),
    CONSTRAINT     FK_Cost_Patient    FOREIGN KEY(PatientID)      REFERENCES
PatientHealthHistory(PatientID),
    CONSTRAINT     FK_Cost_Medication  FOREIGN KEY(MedicationID) REFERENCES
Medications(MedicationID),
);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'UrgentCareCenter' selected. The right pane shows the 'Center' table definition:

```

-- Center Table
CREATE TABLE Centers (
    CenterID      int          NOT NULL,          IDENTITY(1,1),
    CenterName    varchar(100)  NOT NULL,
    BuildingName  varchar(100)  NOT NULL,
    ProcedureCapacity int          NOT NULL,
    Location      varchar(100)  NOT NULL,
    Address       varchar(100)  NOT NULL,
    City          varchar(50)   NOT NULL,
    State         varchar(50)   NOT NULL,
    ZipCode       varchar(50)   NOT NULL,
    Phone         varchar(50)   NOT NULL
);
CONSTRAINT PK_Center PRIMARY KEY CLUSTERED(CenterID ASC)

```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2020-11-29T11:30:13.3173970-05:00".

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (S2)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help Quick Launch (Ctrl+Q) Center

Object Explorer

Connect LAPTOP-2NPL54RH\SQLEXPRESS (S2)

UrgentCareCenter

New Query Execute

```
-- Facility Table
CREATE TABLE [dbo].[facilities] (
    RoomID int NOT NULL,
    CenterID int NOT NULL,
    RoomType varchar(50) NOT NULL,
    CONSTRAINT PK_Facility PRIMARY KEY (CenterID, RoomID),
    CONSTRAINT FK_Facility_Center FOREIGN KEY(CenterID) REFERENCES Centers(CenterID)
);
```

100 % 0 rows Messages Commands completed successfully.

Completion time: 2020-11-29T22:33:13.3178970-05:00

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (S2)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help Quick Launch (Ctrl+Q) Center

Object Explorer

Connect LAPTOP-2NPL54RH\SQLEXPRESS (S2)

UrgentCareCenter

New Query Execute

```
-- Medical Equipment Table
CREATE TABLE [dbo].[MedicalEquipment] (
    EquipmentID int NOT NULL,
    CenterID int NOT NULL,
    EquipmentName varchar(100) NOT NULL,
    Uses varchar(200) NOT NULL,
    CONSTRAINT PK_Equipment PRIMARY KEY (CenterID, EquipmentID),
    CONSTRAINT FK_Equipment_Center FOREIGN KEY(CenterID) REFERENCES Centers(CenterID)
);
```

100 % 0 rows Messages Commands completed successfully.

Completion time: 2020-11-29T22:33:13.3178970-05:00

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- Departments Table
CREATE TABLE [dbo].[Departments] (
    [DeptID] int NOT NULL,
    [CenterID] int NOT NULL,
    [DeptName] varchar(50) NOT NULL,
    [Location] varchar(50) NOT NULL,
    CONSTRAINT PK_Department PRIMARY KEY CLUSTERED([DeptID] ASC),
    CONSTRAINT FK_Dept_Center FOREIGN KEY ([CenterID]) REFERENCES Centers([CenterID])
);
100% ▶
# Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:13.3179970+00:00
```

Query executed successfully.

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- Offices Table
CREATE TABLE [dbo].[Offices] (
    [OfficeID] int NOT NULL,
    [DeptID] int NOT NULL,
    [OfficeName] varchar(50) NOT NULL,
    [OfficeEmail] varchar(50) NOT NULL,
    [OfficePhone] varchar(50) NOT NULL,
    CONSTRAINT PK_Office PRIMARY KEY CLUSTERED([OfficeID]),
    CONSTRAINT FK_Office_Dept FOREIGN KEY ([DeptID]) REFERENCES Departments([DeptID])
);
100% ▶
# Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:13.3179970+00:00
```

Query executed successfully.

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- Employees Table
CREATE TABLE [dbo].[Employees] (
    [EmployeeID] int NOT NULL IDENTITY(1,1),
    [FirstName] varchar(50) NOT NULL,
    [LastName] varchar(50) NOT NULL,
    [SSN] varchar(20) NOT NULL,
    [Address] varchar(50) NOT NULL,
    [City] varchar(50) NOT NULL,
    [ZipCode] varchar(50) NOT NULL,
    [Phone] varchar(50) NOT NULL,
    [Email] varchar(50) NOT NULL,
    [PastJob] varchar(50) NOT NULL,
    [CurJob] varchar(50) NOT NULL,
    [Salary] money NOT NULL,
    [Benefits] varchar(50) NOT NULL,
    [ContractType] varchar(50) NOT NULL,
    [Comments] varchar(200) NOT NULL,
    [Reviews] varchar(200) NOT NULL,
    [CenterID] int NOT NULL,
    [DeptID] int NOT NULL,
    [OfficeID] int NOT NULL,
    [WorkSchedule] varchar(100),
    CONSTRAINT Employee_SSN UNIQUE (SSN),
    CONSTRAINT PK_Employee PRIMARY KEY CLUSTERED([EmployeeID] ASC),
    CONSTRAINT FK_Employee_Center FOREIGN KEY ([CenterID]) REFERENCES Centers([CenterID]),
    CONSTRAINT FK_Employee_Dept FOREIGN KEY ([DeptID]) REFERENCES Departments([DeptID]),
    CONSTRAINT FK_Employee_Office FOREIGN KEY ([OfficeID]) REFERENCES Offices([OfficeID]),
);
100% ▶
# Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:13.3179970+00:00
```

Query executed successfully.

The image displays three separate instances of Microsoft SQL Server Management Studio (SSMS) running simultaneously, each showing the execution of a CREATE TABLE script.

- Top Window:** Shows the creation of the `Physician` table. The table has one column, `EmployeeID`, defined as an integer (int) with a primary key constraint (`PRIMARY KEY`) and a clustered index (`CLUSTERED`). It also includes a foreign key constraint (`CONSTRAINT FK_Physician_Employee`) referencing the `EmployeeID` column in the `Employees` table.
- Middle Window:** Shows the creation of the `Hospital` table. The table has ten columns: `HospitalID` (int, primary key, clustered), `HospitalName` (varchar(50)), `BusinessNumber` (varchar(100)), `Address` (varchar(50)), `City` (varchar(50)), `State` (varchar(50)), `ZipCode` (varchar(50)), `Phone` (varchar(50)), `Website` (varchar(100)), and `PhoneExt` (varchar(50)).
- Bottom Window:** Shows the creation of the `Doctor` table. The table has twelve columns: `DoctorID` (int, primary key, clustered), `HospitalID` (int), `FirstName` (varchar(50)), `LastName` (varchar(50)), `Type` (varchar(50)), `OperationalHours` (varchar(50)), `Phone` (varchar(50)), `Fax` (varchar(50)), `Email` (varchar(50)), `Background` (varchar(200)), and `PhoneExt` (varchar(50)). It includes a foreign key constraint (`CONSTRAINT FK_Doctor_Hospital`) referencing the `HospitalID` column in the `Hospital` table.

All three windows show a successful execution message and a completion time of 2020-11-29T22:33:13.3176970+08:00.

The image displays three separate windows of Microsoft SQL Server Management Studio (SSMS) running on a Windows operating system. Each window shows a different step in the creation of a database named 'UrgentCareCenter'.  
  
1. \*\*Top Window:\*\* Shows the creation of the 'EmergencyDept' table. The code is as follows:

```
-- Emergency Department Table  
CREATE TABLE EmergencyDept (  
    EmergencyDeptID int NOT NULL IDENTITY(1,1),  
    EmergencyDeptName varchar(50) NOT NULL,  
    BusinessPhone varchar(100) NOT NULL,  
    Location varchar(50) NULL,  
    Address varchar(100) NOT NULL,  
    City varchar(50) NOT NULL,  
    State varchar(50) NOT NULL,  
    ZipCode varchar(50) NOT NULL,  
);  
CONSTRAINT PK_EmergencyDept PRIMARY KEY CLUSTERED(EmergencyDeptID ASC),
```

  
2. \*\*Middle Window:\*\* Shows the creation of the 'Medication' table. The code is as follows:

```
-- Medication Table  
CREATE TABLE Medications (  
    MedicationID int NOT NULL IDENTITY(1,1), -- FK Patient(PatientID)  
    MedicationName varchar(50) NOT NULL,  
    Uses varchar(100) NOT NULL,  
    Duration varchar(100) NOT NULL,  
    Warning varchar(200) NOT NULL,  
    ExpiryDate smalldatetime NOT NULL,  
    Price money NOT NULL,  
);  
CONSTRAINT PK_Medication PRIMARY KEY CLUSTERED(MedicationID ASC),
```

  
3. \*\*Bottom Window:\*\* Shows the creation of the 'Patient' table. The code is as follows:

```
-- Patient Table  
CREATE TABLE Patients (  
    PatientID int NOT NULL IDENTITY(1,1),  
    FirstName varchar(50) NOT NULL,  
    LastName varchar(50) NOT NULL,  
    Gender varchar(50) NOT NULL,  
    Address varchar(100) NOT NULL,  
    apt varchar(50) NOT NULL,  
    City varchar(50) NOT NULL,  
    State varchar(50) NOT NULL,  
    ZipCode varchar(50) NOT NULL,  
    PhoneNumber varchar(50) NOT NULL,  
    Email varchar(50) NULL,  
    DOB smalldatetime NOT NULL,  
);  
CONSTRAINT PK_Patient PRIMARY KEY CLUSTERED(PatientID ASC),
```

  
Each window also includes a 'Messages' pane at the bottom showing the command completed successfully and the completion time.

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (S2)) - Microsoft SQL Server Management Studio

```
-- PatientHealthHistory Table
CREATE TABLE [dbo].[PatientHealthHistory] (
    PatientID int NOT NULL, -- FK Patient(PatientID)
    Height varchar(20) NOT NULL,
    Weight varchar(20) NOT NULL,
    CheckIn smalldatetime NOT NULL,
    CheckOut smalldatetime NOT NULL,
    Symptoms varchar(200) NOT NULL,
    Diagnoses varchar(200) NOT NULL,
    Procedures varchar(200) NOT NULL,
    TransferInfo int NULL,
)
CONSTRAINT PK_PatientHealthHistory PRIMARY KEY CLUSTERED(PatientID ASC),
CONSTRAINT FK_History_Patient FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
CONSTRAINT FK_History_EmergDept FOREIGN KEY (TransferInfo) REFERENCES Emergencies(EmergDeptID)
```

100 % 100 %

Messages

Commands completed successfully.

Completion time: 2020-11-19T22:33:13.3174970+01:00

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (S2)) - Microsoft SQL Server Management Studio

```
-- Vital Table
CREATE TABLE [dbo].[Vitals] (
    PatientID int NOT NULL,
    BodyTemperature varchar(50) NULL,
    PulseRate varchar(50) NULL,
    RespiratoryRate varchar(50) NULL,
    BloodPressure varchar(50) NULL,
)
CONSTRAINT PK_Vital PRIMARY KEY CLUSTERED(PatientID),
CONSTRAINT FK_Vital_Patient FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID)
```

100 % 100 %

Messages

Commands completed successfully.

Completion time: 2020-11-19T22:33:13.3214970+01:00

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (S2)) - Microsoft SQL Server Management Studio

```
-- Discharge table
CREATE TABLE [dbo].[Discharge] (
    PatientID int NOT NULL,
    ExpectTime smalldatetime NOT NULL,
)
CONSTRAINT PK_Discharge PRIMARY KEY CLUSTERED(PatientID ASC),
CONSTRAINT FK_History_Discharge FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID)
```

100 % 100 %

Messages

Commands completed successfully.

Completion time: 2020-11-19T22:33:13.3174970+01:00

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Connect UrgentCareCenter

Object Explorer

-- Testing  
`CREATE TABLE Testing (TestingID int NOT NULL, Test varchar(200) NOT NULL, Result varchar(200) NOT NULL, ReportToCounty varchar(50) NOT NULL, ReportToDate varchar(50) NOT NULL, CONSTRAINT PK_Testing PRIMARY KEY CLUSTERED(TestingID ASC), CONSTRAINT FK_Testing_Patient FOREIGN KEY(TestingID) REFERENCES Patients(PatientID)`

100 %

Messages

Command completed successfully.

Completion time: 2020-11-19T22:30:13.3174970+08:00

Query executed successfully.

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Connect UrgentCareCenter

Object Explorer

-- Billing  
`CREATE TABLE Billing (BillID int NOT NULL, Visits varchar(50) NOT NULL, Items varchar(50) NOT NULL, Payee varchar(50) NOT NULL, PaymentMethod varchar(50) NOT NULL, TotalCharge money NOT NULL, CONSTRAINT PK_Bill PRIMARY KEY CLUSTERED(BillID ASC), CONSTRAINT FK_Bill_Patient FOREIGN KEY(BillID) REFERENCES Patients(PatientID)`

100 %

Messages

Command completed successfully.

Completion time: 2020-11-19T22:30:13.3174970+08:00

Query executed successfully.

UrgentCenter.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Connect UrgentCareCenter

Object Explorer

-- Health Insurance Table  
`CREATE TABLE HealthInsurance (HealthInsuranceID int NOT NULL IDENTITY(1,1), PatientID int NOT NULL, Policy_Number varchar(50) NOT NULL, Group_Number varchar(50) NOT NULL, Issue varchar(50) NOT NULL, EffectiveDate smalldatetime NOT NULL, ExpirationDate smalldatetime NOT NULL, Cost money NOT NULL, CONSTRAINT Policy_Info UNIQUE (Policy_Number), CONSTRAINT PK_HealthInsurance PRIMARY KEY CLUSTERED(HealthInsuranceID ASC), CONSTRAINT FK_Patient_Insurance FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)`

100 %

Messages

Command completed successfully.

Completion time: 2020-11-19T22:30:13.3174970+08:00

Query executed successfully.

UrgentCenter.sql - LAPTOP-ZNPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-ZNPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- Coverage Table
CREATE TABLE [dbo].[Coverage] (
    HealthInsuranceID int NOT NULL, -- FK HealthInsurance(HealthInsuranceID)
    CoverageDetails varchar(500) NOT NULL,
    CoveragePrice money NOT NULL
);
GO
ALTER TABLE [dbo].[Coverage] ADD CONSTRAINT [PK_Coverage] PRIMARY KEY (HealthInsuranceID);
ALTER TABLE [dbo].[Coverage] ADD CONSTRAINT [FK_Coverage_Insurance] FOREIGN KEY (HealthInsuranceID) REFERENCES HealthInsurance(HealthInsuranceID);
GO
100 %
Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:18.3178979+04:00
```

UrgentCenter.sql - LAPTOP-ZNPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-ZNPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- InsuranceCompany Table
CREATE TABLE [dbo].[InsuranceCompany] (
    HealthInsuranceID int NOT NULL, -- FK HealthInsurance(HealthInsuranceID)
    CompanyName varchar(50) NOT NULL,
    CompanyPhone varchar(50) NOT NULL,
    Address varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    ZipCode varchar(50) NOT NULL
);
GO
ALTER TABLE [dbo].[InsuranceCompany] ADD CONSTRAINT [PK_InsuranceCompany] PRIMARY KEY (HealthInsuranceID);
ALTER TABLE [dbo].[InsuranceCompany] ADD CONSTRAINT [FK_Company_Insurance] FOREIGN KEY (HealthInsuranceID) REFERENCES HealthInsurance(HealthInsuranceID);
GO
100 %
Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:18.3178979+04:00
```

UrgentCenter.sql - LAPTOP-ZNPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-ZNPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

```
-- AttendingPhysiciansTable
CREATE TABLE [dbo].[AttendingPhysicians] (
    PatientID int NOT NULL, -- Attending Physicians for Patients
    PhysicianID int NOT NULL,
    DateAttending smallint NOT NULL,
    Location varchar(50) NOT NULL,
    Description varchar(200) NOT NULL
);
GO
ALTER TABLE [dbo].[AttendingPhysicians] ADD CONSTRAINT [PK_Attending] PRIMARY KEY (PatientID, PhysicianID);
ALTER TABLE [dbo].[AttendingPhysicians] ADD CONSTRAINT [FK_Attending_Patient] FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID);
ALTER TABLE [dbo].[AttendingPhysicians] ADD CONSTRAINT [FK_Attending_Physician] FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID);
GO
100 %
Messages
Command completed successfully.
Completion time: 2020-11-29T22:39:18.3178979+04:00
```

```

UrgentCenter.mdf - [N:5489\Ause (S2)] > X
CREATE TABLE Referrals (
    PatientID int NOT NULL,
    DoctorID int NOT NULL,
    Date smalldatetime NOT NULL,
    Description varchar(200) NOT NULL,
    CONSTRAINT PK_Referrals PRIMARY KEY (PatientID, DoctorID),
    CONSTRAINT FK_Referrals_Patient FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT FK_Referrals_Doctor FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
100 % > 
0 Messages
Command completed successfully.
Completion time: 2023-11-29T22:39:13.317970+00:00

UrgentCenter.mdf - [N:5489\Ause (S2)] > X
CREATE TABLE VisitHistories (
    PatientID int NOT NULL,
    HospitalID int NOT NULL,
    Date smalldatetime NOT NULL,
    CONSTRAINT PK_VisitHistories PRIMARY KEY (PatientID, HospitalID),
    CONSTRAINT FK_VisitHistories_Patient FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT FK_VisitHistories_Hospital FOREIGN KEY (HospitalID) REFERENCES Hospitals(HospitalID)
);
100 % > 
0 Messages
Command completed successfully.
Completion time: 2023-11-29T22:39:13.317970+00:00

UrgentCenter.mdf - [N:5489\Ause (S2)] > X
CREATE TABLE PrimaryCareDoc (
    PatientID int NOT NULL,
    DoctorID int NOT NULL,
    CONSTRAINT PK_PrimaryCareDoc PRIMARY KEY (PatientID, DoctorID),
    CONSTRAINT FK_PrimaryCareDoc_Patient FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    CONSTRAINT FK_PrimaryCareDoc_Doctor FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
100 % > 
0 Messages
Command completed successfully.
Completion time: 2023-11-29T22:39:13.317970+00:00

UrgentCenter.mdf - [N:5489\Ause (S2)] > X
CREATE TABLE Costs (
    PatientID int NOT NULL,
    MedicationID int NOT NULL,
    Procedures varchar(200) NOT NULL,
    CONSTRAINT PK_Cost PRIMARY KEY (PatientID, MedicationID),
    CONSTRAINT FK_Cost_Patient FOREIGN KEY (PatientID) REFERENCES PatientHealthHistory(PatientID),
    CONSTRAINT FK_Cost_Medication FOREIGN KEY (MedicationID) REFERENCES Medications(MedicationID)
);
100 % > 
0 Messages
Command completed successfully.
Completion time: 2023-11-29T22:39:13.317970+00:00

```

## C. Insert Data into Tables

```

USE UrgentCareCenter

SET IDENTITY_INSERT Patients ON
INSERT Patients(PatientID, FirstName, LastName, Gender, Address, apt, City, State, ZipCode, PhoneNum, Email, DOB)
VALUES
(1, 'Mike', 'James', 'Male', '111 3rd Ave', 'apt 5', 'Syracuse', 'NY', '13201', '3153332553',
'MikeJames@abc.com', '1985-10-12'),
(2, 'Robert', 'Williams', 'Male', '809 Comstock Ave', 'apt 10', 'Syracuse', 'NY', '13210',
'3152854910', 'RobertWilliams@abc.com', '1992-05-21'),
(3, 'Ann', 'Whitaker', 'Female', '1045 Irving St', null, 'San Francisco', 'CA', '94130',
'4159572058', 'AnnWhitaker@abc.com', '1996-01-20'),
(4, 'Lisa', 'Hurlet', 'Female', '4018 Brighton Ave', 'apt 2', 'Syracuse', 'NY', '13210', '3159930205',
'LisaHurlet@abc.com', '1978-08-25'),
(5, 'John', 'Webber', 'Male', '567 Euclid Ave', null, 'Syracuse', 'NY', '13210', '3150295867',
'JohnWebber@abc.com', '1964-10-28')
SET IDENTITY_INSERT Patients OFF
GO

INSERT Billing(BillID, Visits, Items, PayOr, PaymentMethod, TotalCharge)
VALUES
(1, '2020-04-13 to 2020-04-18', 'X-Ray, Pharmacy', 'Self', 'Credit Card', 12000),
(2, '2020-05-01 to 2020-05-01', 'Nucleic Acid Amplification Testing', 'Insurance', 'Credit Card',
350),
(3, '2020-06-02 to 2020-07-02', 'NCAT, Laboratory, Intensive Care Unit', 'Insurance', 'Debit Card',
2000),
(4, '2020-07-13 to 2020-07-14', 'Emergency Room', 'Insurance', 'Check', 5000),
(5, '2020-09-08 to 2020-09-13', 'EKG/ECG', 'Self', 'Check', 2800);
GO

INSERT Testing(TestingID, Test, Result, ReportToCounty, ReportToState)
VALUES
(1, 'Heart Attack', 'None', 'Onondaga', 'NY'),
(2, 'Covid19', 'Negative', 'Onondaga', 'NY'),
(3, 'Covid19', 'Positive', 'San Francisco', 'CA'),
(4, 'Body Injury', 'Serious', 'Onondaga', 'NY'),
(5, 'Hypertension', 'Serious', 'Onondaga', 'NY');
GO

----- Insurance -----
SET IDENTITY_INSERT HealthInsurance ON
INSERT HealthInsurance(HealthInsuranceID, PatientID, Policy_Number, Group_Number, Issuer,
EffectiveDate, ExpiryDate, Cost)
VALUES
(1, 2, 'AEFG892', 'E59S257213', 'E59S257213', '2019-10-23', '2020-10-22', 1096.25),
(2, 3, '9250DGW', 'B32HS02KCW', 'B32HS02KCW', '2020-02-03', '2021-02-02', 2498.96),
(3, 4, 'ANF2R52', '038SWH7W21', '038SWH7W21', '2020-01-01', '2021-12-31', 3696.27);
SET IDENTITY_INSERT HealthInsurance OFF
GO

INSERT Coverages(HealthInsuranceID, CoverageDetails, CoveragePrice)
VALUES
(1, 'Insurance covers the test of COVID19', 350),
(2, 'Insurance covers the test of COVID-19', 350),
(3, 'Insurance covers the protection of Body Injury', 2000);
GO
INSERT InsuranceCompany(HealthInsuranceID, CompanyName, CompanyPhone, Address, City, State, ZipCode)
VALUES
(1, 'Aetna', '8553351407', '441 S Salina St', 'Syracuse', 'NY', '13202'),
(2, 'Blue Cross', '4157353807', '899 Howard St', 'San Francisco', 'CA', '94103'),
(3, 'Humana', '3157015843', '338 S Salina ST', 'Syracuse', 'NY', '13202');
GO

SET IDENTITY_INSERT EmergencyDept ON
INSERT EmergencyDept(EmergDeptID, EmergDeptName, BusinessHour, Location, Address, City, State,
ZipCode)

```

```

VALUES
(1, 'Upstate Medical Center Emergency Department', 'Open 24 hours', 'Upstate University Hospital',
'750 E Adam St', 'Syracuse', 'NY', '13210'),
(2, 'Anburn Community Hospital: Emergency Room', 'Open 24 hours', 'Auburn Community Hospital', '17
Lansing St', 'Auburn', 'NY', '13021'),
(3, 'UCSF Emergency Department', 'Open 24 hours', 'UCSF Medical Center', '505 Parnassus Ave', 'San
Francisco', 'CA', '94143');
SET IDENTITY_INSERT EmergencyDept OFF
GO

INSERT PatientHealthHistory(PatientID, Weight, Height, CheckIn, CheckOut, Symptoms, Diagnostics,
Procedures, TransferInfo)
VALUES
(1, '160lb', '170cm', '2020-04-13 08:00:00', '2020-04-13 10:00:00', 'Heart Burn and Hurt. Feeling
Anxious', 'ECG, Echocardiogram, CT', 'Medical Treatment', 1),
(2, '176lb', '175cm', '2020-04-20 09:30:00', '2020-04-20 11:30:00', 'Coughing, Dizzy', 'Nucleic Acid
Amplification Testing', 'Suggestion: Drink more water', null),
(3, '130lb', '164cm', '2020-05-28 09:30:00', '2020-05-28 11:30:00', 'Coughing, Fewer, Difficulty
Breathing', 'Nucleic Acid Amplification Testing', 'ICU', 3),
(4, '128lb', '160cm', '2020-07-01 08:30:00', '2020-07-01 10:30:00', 'Need Surgery, Car Accident
attacking Body', 'CT, X-Ray', 'Surgery', 2),
(5, '220lb', '185cm', '2020-09-01 08:30:00', '2020-09-01 10:30:00', 'Tinnitus, Dizzy and Vomiting',
'First Check: ECG, GFR, Blood Lipids, Urinalysis', 'Medical Treatment', null);
GO

INSERT Discharge(PatientID, ExpectTime)
VALUES
(1, '2020-04-18'),
(2, '2020-05-01'),
(3, '2020-07-02'),
(4, '2020-07-14'),
(5, '2020-09-13');
GO

INSERT Vitals(PatientID, BodyTemperature, PulseRate, RespirationRate, BloodPressure)
VALUES
(1, '97.7F', '130bpm', '20bpm', '130mmHg'),
(2, '98.06', '90bpm', '14bpm', '100mmHg'),
(3, '101.12', '105bpm', '10bpm', '120mmHg'),
(4, '96.8', '140bpm', '22bpm', '140mmHg'),
(5, '97.5', '125bpm', '18bpm', '150mmHg');
GO

SET IDENTITY_INSERT Medications ON
INSERT Medications(MedicationID, MedicationName, Uses, Direction, Warning, ExpiryDate, Price)
VALUES
(1, 'Aspirin', 'It can reduce the risk of heart attack', 'Take 1 or 2 tablets every 4 hours',
'It irritates your stomach lining and can trigger gastrointestinal upset, ulcers and bleeding', '2021-
12-20', 20.05),
(2, 'Amoxil', 'Treat bacterial infections', 'Take it every 12 hours with or without food',
'Rash, itching/swelling (especially of the face/tongue/throat), severe dizziness, trouble breathing',
'2022-03-12', 10.71),
(3, 'Thiazide diuretics', 'Clear fluid from the body', 'once-a-day dose',
'Thiazide diuretics can lower potassium and magnesium blood levels since they are both eliminated in
urine.', '2021-05-09', 30.25),
(4, 'Opioids', 'Relieve pain', 'Please conduct a doctor',
'Serious risks of misuse and abuse, which can lead to addiction, overdose and death', '2021-02-02',
6),
(5, 'Tylenol Cough', 'Cough Relief', 'Two tablets every 12 hours',
'Do not misuse it', '2021-04-12', 20.08),
(6, 'Ambien', 'Help sleep well and fast', 'One tablets a day',
'Misuse leads to tragic injuries or even deaths', '2020-12-20', 10.29),
(7, 'Ibuprofen', 'Reduce Fever', 'Conduct a doctor',
'Do not misuse', '2021-09-13', 13.98);
SET IDENTITY_INSERT Medications OFF
GO

INSERT Costs(PatientID, MedicationID, Procedures)
VALUES

```

```

(1, 1, 'The patient needs Aspirin to reduce heart attack'),
(1, 6, 'Help patient sleep well at night to reduce heart attack'),
(2, 5, 'Relief cough, suggestion: drink more water'),
(3, 2, 'Treatment for bacterial'),
(3, 5, 'Relief cough'),
(3, 7, 'Reduce fever'),
(4, 4, 'Reduce pain'),
(5, 3, 'Treatment for Hypertension'),
(5, 6, 'Help sleep well');
GO

SET IDENTITY_INSERT Hospitals ON
INSERT Hospitals(HospitalID, HospitalName, BusinessHour, Location, Address, City, State, ZipCode,
Phone, Webside)
VALUES
(1, 'Upstate Community Hospital', 'Open 24 hours', 'University Hospital', '4900 Broad Rd', 'Syracuse',
'NY', '13215', '3154925011', 'upstate.edu'),
(2, 'Community General Hospital', 'Open 24 hours', 'Preventive Medicine Associates', '5415 W Genesee
St #101', 'Camillus', 'NY', '13031', '3154681997', 'upstate.edu'),
(3, 'Upstate University Hospital', 'Open 24 hours', 'University Hospital', '750 E Adams St',
'Syracuse', 'NY', '13210', '3154645540', 'upstate.edu'),
(4, 'Oneida Health Hospital', 'Open 24 hours', 'Oneida', '321 Fenesee St', 'Oneida', 'NY', '13421',
'3153636000', 'oneidahealth.org'),
(5, 'Laguna Honda Hospital', 'Open 24 hours', 'Daly City', '375 Laguna Honda Blvd', 'San Francisco',
'CA', '94116', '4157592300', 'lagunahonda.org');
SET IDENTITY_INSERT Hospitals OFF
GO

INSERT VisitHospital(PatientID, HospitalID, Date)
VALUES
(1, 1, '2020-04-13 08:00:00'),
(2, 4, '2020-05-01 09:30:00'),
(3, 5, '2020-06-02 09:30:00'),
(4, 3, '2020-07-13 10:30:00'),
(5, 2, '2020-09-08 10:30:00');
GO

SET IDENTITY_INSERT Doctors ON
INSERT Doctors(DoctorID, HospitalID, FirstName, LastName, Type, OperationHour, Phone, Fax, Email,
Background)
VALUES
(1, 1, 'Angie', 'McKue', 'Cardiovascular Medicine', '8:00 - 17:00', '3154830295',
'AngieMcKue@fax.com', 'AngieMcKue@abc.com', 'McKue is professional on treatment of Heart Disease'),
(2, 2, 'Lucina', 'Lentsch', 'Internal Medicine', '8:00 - 17:00', null, 'LucinaLentsch@fax.com',
'LucinaLentsch@abc.com', 'Lentsch is good at Internal Medicine'),
(3, 3, 'Katlin', 'Babidge', 'Orthopedics', '8:00 - 17:00', '3152058401', null,
'KatlinBabidge@abc.com', 'Babidge is good at Orthopedics'),
(4, 4, 'Carl', 'Deverille', 'Respiratory', '8:00 - 17:00', '3152940642', 'CarlDeverille@fax.com',
'CarlDeverille@abc.com', 'Deverille is good at Respiratory'),
(5, 5, 'Jodi', 'Clues', 'Respiratory', '8:00 - 17:00', '4152048606', 'JodiClues@fax.com',
'JodiClues@abc.com', 'Clues is good at Respiratory'),
(6, 3, 'Jens', 'Hardy', 'Brain Surgery', '8:00 - 17:00', '3152851406', null, null, null),
(7, 4, 'Lindy', 'Pinn', 'Neuroogy', '8:00 - 17:00', null, null, null, null);
SET IDENTITY_INSERT Doctors OFF
GO

INSERT PrimaryCareDoc(PatientID, DoctorID)
VALUES
(1, 1),
(2, 4),
(3, 5),
(4, 3),
(5, 2);
GO

INSERT Refer(PatientID, DoctorID, Date, Description)
VALUES
(1, 1, '2020-04-13 08:00:00', 'To See Cardiovascular Medicine Doctor'),
(2, 4, '2020-05-01 09:30:00', 'To See Doctor'),

```

```

(3, 5, '2020-06-02 09:30:00', 'To See Doctor'),
(4, 3, '2020-07-13 10:30:00', 'To See Doctor'),
(5, 2, '2020-09-08 10:30:00', 'To See Doctor'));
GO

SET IDENTITY_INSERT Centers ON
INSERT Centers(CenterID, CenterName, BusinessHour, ProcedureCapacity, Location, Address, City, State, ZipCode, Phone)
VALUES
(1, 'States MD Urgent Care Center', 'Open 24 hours', '500', 'Genesee', '3504 W Genesee St', 'Syracuse', 'NY', '13219', '3154010754'),
(2, 'States MD Urgent Care Center', 'Open 24 hours', '300', 'Oneida', '109 Genesee St', 'Oneida', 'NY', '13421', '3152315530'),
(3, 'States MD Urgent Care Center', 'Open 24 hours', '1000', 'Parnassus Campus', '400 Parnassus Ave', 'San Francisco', 'CA', '94122', '4153532602');
SET IDENTITY_INSERT Centers OFF
GO

INSERT Facilities(RoomID, CenterID, RoomType)
VALUES
(1, 1, 'Waiting Room'),
(2, 1, 'Detection Room'),
(3, 1, 'Emergency Room'),
(1, 2, 'Waiting Room'),
(2, 2, 'Detection Room'),
(3, 2, 'Emergency Room'),
(1, 3, 'Waiting Room'),
(2, 3, 'Detection Room'),
(3, 3, 'Emergency Room');
GO

INSERT MedicalEquipments(EquipmentID, CenterID, EquipmentName, Uses)
VALUES
(1, 1, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 1, 'Philips Heartstart Onsite Home Aed Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 1, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking'),
(1, 2, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 2, 'Philips Heartstart Onsite Home Aed Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 2, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking'),
(1, 3, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 3, 'Philips Heartstart Onsite Home Aed Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 3, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking');
GO

INSERT Departments(DeptID, CenterID, DeptName, Location)
VALUES
(1, 1, 'Emergency Department', 'Emergency Area'),
(2, 1, 'Ambulance Department', 'Ambulance Area'),
(3, 1, 'Physician Department', 'Physician Area'),
(4, 2, 'Emergency Department', 'Emergency Area'),
(5, 2, 'Ambulance Department', 'Ambulance Area'),
(6, 2, 'Physician Department', 'Physician Area'),
(7, 3, 'Emergency Department', 'Emergency Area'),
(8, 3, 'Ambulance Department', 'Ambulance Area'),
(9, 3, 'Physician Department', 'Physician Area');
GO

INSERT Offices(OfficeID, DeptID, OfficeName, OfficeNum, OfficePhone)
VALUES
(1, 1, 'Emergency Office #1', '101', '3152950601'),
(2, 1, 'Emergency Office #2', '104', '3152950602'),
(3, 2, 'Ambulance Office #1', '107', '3152950611'),
(4, 2, 'Ambulance Office #2', '100', '3152950612'),
(5, 2, 'Ambulance Office #3', '103', '3152950613'),
(6, 3, 'Physician Office', '110', '3152950621'),
(7, 4, 'Emergency Office #1', '101', '3152048601'),

```

```

(8, 4, 'Emergency Office #2', '104', '3152048602'),
(9, 5, 'Ambulance Office #1', '102', '3152048611'),
(10, 5, 'Ambulance Office #2', '106', '3152048612'),
(11, 6, 'Physician Office', '100', '3152048621'),
(12, 7, 'Emergency Office #1', '102', '4152305901'),
(13, 7, 'Emergency Office #2', '104', '4152305902'),
(14, 7, 'Emergency Office #3', '106', '4152305903'),
(15, 8, 'Ambulance Office #1', '100', '4152305911'),
(16, 8, 'Ambulance Office #2', '108', '4152305912'),
(17, 9, 'Physician Office', '123', '4152305921')
GO

SET IDENTITY_INSERT Employees ON
INSERT Employees(EmployeeID, FirstName, LastName, SSN, Address, apt, City, State, ZipCode, Phone,
Email, PastJob, CurJob, Salary, Benefits, ContractType, ContractTerm,
Reviews, CenterID, DeptID, OfficeID, WorkSchedule)
VALUES
(1, 'Sophia', 'Jackson', '123-45-6789', '111 1st st', 'apt 5', 'Syracuse', 'NY', '13210',
'3155931023', 'SophiaJackson@abc.com', null, 'Nurse', 54000, 'Insurance provided',
'Lump Sum Contract', '1 year', 'Excellent', 1, 1, 1, '9am - 5pm'),
(2, 'Emma', 'Aiden', '204-12-5292', '222 2st st', null, 'Syracuse', 'NY', '13210', '3150203928',
'EmmaAiden@abc.com', 'Nurse', 79000, 'Stocks & Insurance',
'Cost Plus Contract', '5 years', 'Excellent', 1, 1, 1, '9am - 5pm'),
(3, 'Olivia', 'Lucas', '103-20-1059', '333 23st st', 'apt 1', 'Syracuse', 'NY', '13210', '3152039571',
'OliviaLucas@abc.com', null, 'Nurse', 53000, 'Insurance',
'Lump Sum Contract', '2 years', 'Excellent', 1, 1, 2, '9am - 5pm'),
(4, 'Ava', 'Liam', '103-20-1048', '444 23st st', null, 'Syracuse', 'NY', '13210', '3151038261',
'Avaliam@abc.com', 'Nurse', 'Nurse', 83000, 'Stocks & Insurance',
'Cost Plus Contract', '10 years', 'Excellent', 1, 1, 2, '9am - 5pm'),
(5, 'Rajesh', 'Menon', '283-10-2057', '555 1st st', null, 'Syracuse', 'NY', '13210', '3151920386',
'RajeshMenon@abc.com', 'Driver', 'Driver', 53000, 'Health Insurance, extra PTO',
'Unit Price Contract', '2 years', 'Good', 1, 2, 3, '11am - 11pm'),
(6, 'Kriti', 'Sharma', '401-12-1957', '666 4th st', 'apt 6', 'Syracuse', 'NY', '13210', '3152059261',
'KritiSharma@abc.com', null, 'Aid man', 76000, 'Health Insurance, extra PTO, stock',
'Cost Plus Contract', '3 years', 'Good', 1, 2, 3, '11am - 11pm'),
(7, 'Pallavi', 'Roy', '201-20-0271', '777 23rd st', null, 'Syracuse', 'NY', '13210', '3152019562',
'PallaviRoy@abc.com', 'Aid man', 'Aid man', 87000, 'Health Insurance, extra PTO, stock',
'Cost Plus Contract', '5 years', 'Excellent', 1, 2, 3, '11am - 11pm'),
(8, 'Aman', 'Verma', '185-20-1462', '888 3rd st', null, 'Syracuse', 'NY', '13210', '3151018262',
'AmanVerma@abc.com', null, 'Driver', 58000, 'Health Insurance, extra PTO',
'Cost Plus Contract', '2 years', 'Excellent', 1, 2, 4, '5am - 5pm'),
(9, 'John', 'Turner', '291-10-2947', '999 3rd st', 'apt 3', 'Syracuse', 'NY', '13210', '3150001928',
'JohnTurner@abc.com', null, 'Aid man', 78000, 'Health Insurance, stock',
'Incentive Contract', '2 years', 'Excellent', 1, 2, 4, '5am - 5pm'),
(10, 'Anders', 'Maria', '295-18-2058', '345 Winchell Pl', null, 'Syracuse', 'NY', '13210',
'7655557878', 'AndersMaria@abc.com', null, 'Aid man', 68000, 'Health Insurance',
'Cost Plus Contract', '2 years', 'Excellent', 1, 2, 4, '5am - 5pm'),
(11, 'Trujillo', 'Ana', '204-19-0597', '1298 E Smathers St', null, 'Syracuse', 'NY', '13210',
'5015557733', 'TrujilloAna@abc.com', 'Driver', 'Driver', 68500, 'Health Insurance',
'Cost Plus Contract', '3 years', 'Good', 1, 2, 5, '8am - 8pm'),
(12, 'Moreno', 'Antonio', '502-19-0003', '6925 N Parkland Ave', null, 'Syracuse', 'NY', '13210',
'2535558332', 'MorenoAntonio@abc.com', 'Aid man', 'Aid man', 78000, 'Health Insurance, extra PTO',
'Incentive Contract', '5 years', 'Excellent', 1, 2, 5, '8am - 8pm'),
(13, 'Hardy', 'Thomas', '205-19-2756', '83 d''Urberville Ln', null, 'Syracuse', 'NY', '13210',
'4785551139', 'HardyThomas@abc.com', 'Aid man', 'Aid man', 78000, 'Health Insurance, extra PTO',
'Incentive Contract', '5 years', 'Excellent', 1, 2, 5, '8am - 8pm'),
(14, 'Berglund', 'Christina', '693-01-2846', '22717 E 73rd Ave', null, 'Syracuse', 'NY', '13210',
'3195551139', 'BerglundChristina@abc.com', null, 'Physician', 90000, 'Health Insurance, extra PTO',
'Cost Plus Contract', '5 years', 'Excellent', 1, 3, 6, '8am - 5pm'),
(15, 'Moos', 'Hanna', '295-10-1297', '1778 N Bovine Ave', null, 'Syracuse', 'NY', '13210',
'3095558755', 'MoosHanna@abc.com', 'Physician', 'Physician', 130000, 'Health Insurance, extra PTO',
'Cost Plus Contract', '10 years', 'Excellent', 1, 3, 6, '11am - 7pm'),
(16, 'Citeaux', 'Fred', '201-49-1908', '1234 Main St', null, 'Syracuse', 'NY', '13210', '3095551914',
'CiteauxFred@abc.com', 'Physician', 'Physician', 135000, 'Health Insurance, extra PTO',
'Cost Plus Contract', '8 years', 'Excellent', 1, 3, 6, '3pm - 11pm'),
(17, 'Summer', 'Martin', '293-10-0298', '1877 Ete Ct', 'apt 11', 'Syracuse', 'NY', '13210',
'3375559441', 'SummerMartin@abc.com', null, 'Physician', 125000, 'Health Insurance, extra PTO',
'Incentive Contract', '3 years', 'Excellent', 1, 3, 6, '8am - 5pm'),

```

(18, 'Lebihan', 'Laurence', '201-75-0991', '717 E Michigan Ave', null, 'Syracuse', 'NY', '13210', '3125559441', 'LebihanLaurence@abc.com', null, 'Nurse', 65000, 'Health Insurance, extra PTO', 'Lump Sum Contract', '1 years', 'Good', 2, 4, 7, '8am - 5pm'),  
 (19, 'Lincoln', 'Elizabeth', '124-20-1122', '4562 Rt 78 E', null, 'Syracuse', 'NY', '13210', '3605552680', 'LincolnElizabeth@abc.com', 'Nurse', 'Nurse', 75000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 2, 4, 7, '8am - 5pm'),  
 (20, 'Snyder', 'Howard', '105-27-1902', '2732 Baker Blvd.', null, 'Syracuse', 'NY', '13210', '5035557555', 'SnyderHoward@abc.com', 'Nurse', 'Nurse', 77000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 2, 4, 8, '11am - 7pm'),  
 (21, 'Latimer', 'Yoshi', '182-00-1038', 'City Center Plaza 516 Main St.', null, 'Syracuse', 'NY', '13210', '5035556874', 'LatimerYoshi@abc.com', 'Nurse', 'Nurse', 77000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 2, 4, 8, '11am - 7pm'),  
 (22, 'Steel', 'John', '193-00-2001', '12 Orchestra Terrace', null, 'Syracuse', 'NY', '13210', '5095557969', 'SteelJohn@abc.com', 'Driver', 'Driver', 54000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '2 years', 'Excellent', 2, 5, 9, '11am - 11pm'),  
 (23, 'Yorres', 'Jaime', '183-19-0023', '87 Polk St. Suite 5', null, 'Syracuse', 'NY', '13210', '4155555938', 'YorresJaime@abc.com', null, 'Aid man', 65000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '2 years', 'Excellent', 2, 5, 9, '11am - 11pm'),  
 (24, 'Wilson', 'Fran', '104-20-1977', '89 Chiaroscuro Rd.', null, 'Syracuse', 'NY', '13210', '5035559573', 'WilsonFran@abc.com', 'Aid man', 'Aid man', 68000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 2, 5, 9, '11am - 11pm'),  
 (25, 'Phillips', 'Rene', '010-20-4184', '2743 Bering St.', null, 'Syracuse', 'NY', '13210', '9075557584', 'PhillipsRene@abc.com', 'Driver', 'Driver', 52000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '2 years', 'Excellent', 2, 5, 10, '5am - 5pm'),  
 (26, 'Wilson', 'Paula', '103-77-0287', '2817 Milton Dr.', null, 'Syracuse', 'NY', '13210', '5055555939', 'WilsonPaula@abc.com', 'Aid man', 'Aid man', 67900, 'Health Insurance, extra PTO', 'Cost Plus Contract', '2 years', 'Excellent', 2, 5, 10, '5am - 5pm'),  
 (27, 'Pavarotti', 'Jose', '942-01-2856', '187 Suffolk Ln.', null, 'Syracuse', 'NY', '13210', '2085558097', 'PavarottiJose@abc.com', 'Aid man', 'Aid man', 68000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '2 years', 'Excellent', 2, 5, 10, '5am - 5pm'),  
 (28, 'Braunschweiger', 'Art', '293-10-1912', 'P.O. Box 555', null, 'Syracuse', 'NY', '13210', '3075554680', 'BraunschweigerArt@abc.com', 'Physician', 'Physician', 98000, 'Health Insurance, extra PTO',  
   'Cost Plus Contract', '5 years', 'Excellent', 2, 6, 11, '8am - 5pm'),  
 (29, 'Nixon', 'Liz', '859-10-2857', '89 Jefferson Way Suite 2', null, 'Syracuse', 'NY', '13210', '4015553612', 'NixonLiz@abc.com', 'Physician', 'Physician', 130200, 'Health Insurance, extra PTO', 'Cost Plus Contract', '5 years', 'Excellent', 2, 6, 11, '8am - 5pm'),  
 (30, 'Wong', 'Liu', '492-10-6749', '55 Grizzly Peak Rd.', 'apt 1', 'San Francisco', 'CA', '94101', '4065555834', 'WongLiu@abc.com', 'Nurse', 'Nurse', 69000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 3, 7, 12, '8am - 11pm'),  
 (31, 'Nagy', 'Helvetius', '591-19-0059', '722 DaVinci Blvd.', null, 'San Francisco', 'CA', '94101', '3515551219', 'NagyHelvetius@abc.com', 'Nurse', 'Nurse', 69000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '5 years', 'Excellent', 3, 7, 12, '8am - 11pm'),  
 (32, 'Jablonski', 'Karl', '195-20-3957', '305 - 14th Ave. S.', 'Suite 3B', 'San Francisco', 'CA', '94101', '2065554112', 'JablonskiKarl@abc.com', null, 'Nurse', 60000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '4 years', 'Good', 3, 7, 13, '8am - 8pm'),  
 (33, 'Chelan', 'Donna', '195-10-5867', '2299 E Baylor Dr.', null, 'San Francisco', 'CA', '94101', '4695558828', 'ChelanDonna@abc.com', 'Nurse', 'Nurse', 68000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 3, 7, 13, '8am - 8pm'),  
 (34, 'Moos', 'Li', '295-10-0101', '1778 N Bovine Ave', null, 'San Francisco', 'CA', '94101', '4150248521', 'MoosLi@abc.com', 'Nurse', 'Nurse', 78000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '10 years', 'Excellent', 3, 7, 14, '11am - 7pm'),  
 (35, 'Citeaux', 'Fran', '201-49-0202', '1234 Main St', null, 'San Francisco', 'CA', '94101', '4152048271', 'CiteauxFran@abc.com', 'Driver', 'Driver', 65000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 3, 8, 15, '8am - 5pm'),  
 (36, 'Summer', 'Jose', '293-10-0303', '1877 Ete Ct', 'apt 11', 'San Francisco', 'CA', '94101', '4152817321', 'SummerJose@abc.com', null, 'Aid man', 78000, 'Health Insurance, extra PTO', 'Incentive Contract', '3 years', 'Excellent', 3, 8, 15, '8am - 5pm'),  
 (37, 'Lebihan', 'Martin', '201-75-0404', '717 E Michigan Ave', null, 'San Francisco', 'CA', '94101', '4152010000', 'LebihanMartin@abc.com', null, 'Driver', 65000, 'Health Insurance, extra PTO', 'Lump Sum Contract', '1 years', 'Good', 3, 8, 16, '11am - 11pm'),  
 (38, 'Lincoln', 'Menon', '124-20-0505', '4562 Rt 78 E', null, 'San Francisco', 'CA', '94101', '4152221111', 'LincolnMenon@abc.com', 'Aid man', 'Aid man', 75000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 3, 8, 16, '11am - 11pm'),  
 (39, 'Snyder', 'Aiden', '105-27-0606', '2732 Baker Blvd.', null, 'San Francisco', 'CA', '94101', '4152235501', 'SnyderAiden@abc.com', 'Physician', 'Physician', 100000, 'Health Insurance, extra PTO', 'Cost Plus Contract', '3 years', 'Excellent', 3, 9, 17, '8am - 7pm'),

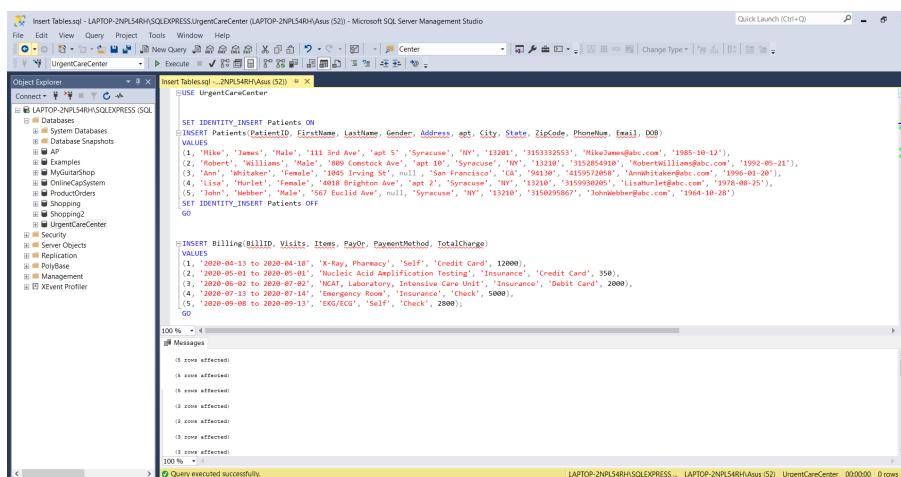
```

(40, 'Latimer', 'Hanna', '182-00-0707', 'City Center Plaza 516 Main St.', null, 'San Francisco', 'CA', '94101', '4152210405', 'LatimerHanna@abc.com', 'Physician', 'Physician', 103000, 'Health Insurance, extra PTO',
'Cost Plus Contract', '3 years', 'Excellent', 3, 9, 17, '8am - 7pm');
SET IDENTITY_INSERT Employees OFF
GO

SET IDENTITY_INSERT Physicians ON
INSERT Physicians(PhysicianID, EmployeeID, Type)
VALUES
(1, 14, 'Family Physician'),
(2, 15, 'Internists'),
(3, 16, 'Surgeons'),
(4, 17, 'Anesthesiologist'),
(5, 28, 'Family Physician'),
(6, 29, 'Internists'),
(7, 39, 'Family Physician'),
(8, 40, 'Internists');
SET IDENTITY_INSERT Physicians OFF
GO

INSERT Attending(PatientID, PhysicianID, Date, Description)
VALUES
(1, 1, '2020-04-05 08:00:00', 'Mike feels bad on heart, Berglund helps him'),
(1, 2, '2020-04-05 08:00:00', 'Mike feels bad on heart, Moos helps him'),
(2, 4, '2020-04-29 09:30:00', 'Robert cough, Summer helps him'),
(3, 7, '2020-06-01 09:30:00', 'Ann feels getting covid19, Snyder helps her'),
(3, 8, '2020-06-01 09:30:00', 'Ann feels getting covid19, Latimer helps her'),
(4, 5, '2020-07-13 10:00:00', 'Lisa get car accidents to hurt body, Braunschweiger helps her'),
(4, 6, '2020-07-13 10:00:00', 'Lisa get car accidents to hurt body, Nixon helps her'),
(5, 3, '2020-09-07 15:30:00', 'John feels dizzy (hypertension), Citeaux helps him'),
(5, 4, '2020-09-07 15:30:00', 'John feels dizzy (hypertension), Summer helps him');
GO

```



Insert Table.sql - LAPTOP-2NPL54RH\SQLEXPRESS\UrgentCareCenter (LAPTOP-2NPL54RH\Asus (S2)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect to Server

LAPTOP-2NPL54RH\SQLEXPRESS (SQL Server) -> LAPTOP-2NPL54RH\Asus (S2)

- Connect
- Connect to Database
- Databases
- System Databases
- Database Snapshots
- File
- Filegroup
- Examples
- My�다Shop
- OnlineCareSystem
- Products
- Shopping
- Shipping
- UrgentCareCenter
- Security
- Server Objects
- Replication
- Polybase
- Management
- XEvent Profiler

Insert Table.sql - Result

```
----- Testing -----
SET IDENTITY_INSERT TestingID, Test, Result, ReportToCounty, ReportToState
VALUES
(1, 'Positive', 'Affect', 'Oncodage', 'NY'),
(2, 'Covid19', 'Negative', 'Ondodage', 'NY'),
(3, 'Covid19', 'Positive', 'San Francisco', 'CA'),
(4, 'Body Injury', 'Serious', 'Ondodage', 'NY'),
(5, 'Hypertension', 'Serious', 'Ondodage', 'NY');

----- Insurance -----
SET IDENTITY_INSERT HealthInsurance ON
INSERT HealthInsurance([HealthInsuranceID], PatientID, Policy_Number, Group_Number, Issuer, EffectiveDate, ExpiryDate, Cost)
VALUES
(1, 'AEFG092', 'E595257213', 'E595257213', '2019-08-23', '2020-10-22', 1000.25),
(2, 3, '9250000', '032H502KCM', '832H502KCM', '2020-02-03', '2021-02-02', 2498.96),
(3, 4, 'JAN20252', '0385AH7W21', '0385AH7W21', '2020-01-01', '2021-12-31', 3696.27);
SET IDENTITY_INSERT HealthInsurance OFF
GO

----- Messages -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(6, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- Coverage -----
SET IDENTITY_INSERT Coverage ON
INSERT Coverage([HealthInsuranceID], CoverageDetails, CoveragePrice)
VALUES
(1, 'Insurance covers the test of COVID19', 350),
(2, 'Insurance covers the test of COVID-19', 350),
(3, 'Insurance covers the protection of Body Injury', 2000);
GO

----- InsuranceCompany -----
SET IDENTITY_INSERT HealthInsuranceID ON
INSERT HealthInsuranceID([HealthInsuranceID], CompanyName, CompanyPhone, Address, City, State, ZipCode)
VALUES
(1, 'Upscate Medical Center', '8853351407', '441 S Salina St', 'Syracuse', 'NY', '13202'),
(2, 'Auburn Community Hospital', '451572536007', '899 Howard St', 'San Francisco', 'CA', '94103'),
(3, 'Humania', '3157015843', '338 S Salina ST', 'Syracuse', 'NY', '13202');
GO

----- EmergencyDept -----
SET IDENTITY_INSERT EmergencyDept ON
INSERT EmergencyDept([EmergencyDeptID], EmergencyDeptName, BusinessHour, Location, Address, City, State, ZipCode)
VALUES
(1, 'Upstate Medical Center Emergency Department', 'Open 24 hours', 'Upstate University Hospital', '750 E Adam St', 'Syracuse', 'NY', '13210'),
(2, 'Auburn Community Hospital Emergency Room', 'Open 24 hours', 'Auburn Community Hospital', '17 Lansing St', 'Auburn', 'NY', '13021'),
(3, 'UCSF Medical Center', 'Open 24 hours', 'UCSF Medical Center', '595 Parnassus Ave', 'San Francisco', 'CA', '94143');
SET IDENTITY_INSERT EmergencyDept OFF
GO

----- Message -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- PatientHistory -----
SET IDENTITY_INSERT PatientHistory ON
INSERT PatientHistory([PatientID], weight, Height, CheckIn, Checkout, Symptoms, Diagnostics, Procedures, TransferInfo)
VALUES
(1, '160lb', '170cm', '2020-04-13 00:00:00', '2020-04-13 10:00:00', 'Heart Burn and Hurt, Feeling Anxious', 'ECG, Echoangiogram, CT', 'Medical Treatment', 1),
(2, '160lb', '175cm', '2020-04-20 09:00:00', '2020-04-20 11:30:00', 'Coughing, Dizzy', 'Nucleic Acid Amplification Testing', 'Suggestion: Drink more water', null),
(3, '160lb', '164cm', '2020-05-26 08:30:00', '2020-05-28 11:30:00', 'Coughing, Fever, Difficultly Breathing', 'Nucleic Acid Amplification Testing', 'ICU', 3),
(4, '160lb', '160cm', '2020-07-01 08:30:00', '2020-07-01 10:30:00', 'Need Surgery, Car Accident attacking Body', 'CT, X-Ray', 'Surgery', 2),
(5, '220lb', '185cm', '2020-09-01 08:30:00', '2020-09-01 08:30:00', 'Dizziness, Blurry and Vomiting', 'First Check: ECG, GFR, Blood Lipids, Urinalysis', 'Medical Treatment', 5);

----- Discharge -----
SET IDENTITY_INSERT Discharge ON
INSERT Discharge([PatientID], ExpectTime)
VALUES
(1, '2020-04-18'),
(2, '2020-05-01'),
(3, '2020-06-01'),
(4, '2020-07-14'),
(5, '2020-09-13');

----- VitalSigns -----
SET IDENTITY_INSERT PatientID ON
INSERT PatientID([PatientID], BodyTemperature, PulseRate, RespirationRate, BloodPressure)
VALUES
(1, '97.7F', '130bpm', '170bpm', '170mmHg'),
(2, '98.06', '140bpm', '180bpm', '180mmHg'),
(3, '100.11', '160bpm', '160bpm', '120mmHg'),
(4, '99.98', '140bpm', '220bpm', '140mmHg'),
(5, '97.5', '120bpm', '180bpm', '150mmHg');

----- Message -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- UrgentCareCenter -----
SET IDENTITY_INSERT UrgentCareCenter ON
INSERT UrgentCareCenter([UrgentCareCenterID], UrgentCareCenterName, Address, City, State, ZipCode)
VALUES
(1, 'UrgentCareCenter', '123 Main Street', 'New York', 'NY', '10001');

```

Query executed successfully.

Insert Table.sql - LAPTOP-2NPL54RH\SQLEXPRESS\UrgentCareCenter (LAPTOP-2NPL54RH\Asus (S2)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect to Server

LAPTOP-2NPL54RH\SQLEXPRESS (SQL Server) -> LAPTOP-2NPL54RH\Asus (S2)

- Connect
- Connect to Database
- Databases
- System Databases
- Database Snapshots
- File
- Filegroup
- Examples
- My�다Shop
- OnlineCareSystem
- Products
- Shopping
- Shipping
- UrgentCareCenter
- Security
- Server Objects
- Replication
- Polybase
- Management
- XEvent Profiler

Insert Table.sql - Result

```
----- Testing -----
SET IDENTITY_INSERT TestingID, Test, Result, ReportToCounty, ReportToState
VALUES
(1, 'Positive', 'Affect', 'Oncodage', 'NY'),
(2, 'Covid19', 'Negative', 'Ondodage', 'NY'),
(3, 'Covid19', 'Positive', 'San Francisco', 'CA'),
(4, 'Body Injury', 'Serious', 'Ondodage', 'NY'),
(5, 'Hypertension', 'Serious', 'Ondodage', 'NY');

----- Insurance -----
SET IDENTITY_INSERT HealthInsurance ON
INSERT HealthInsurance([HealthInsuranceID], PatientID, Policy_Number, Group_Number, Issuer, EffectiveDate, ExpiryDate, Cost)
VALUES
(1, 'AEFG092', 'E595257213', 'E595257213', '2019-08-23', '2020-10-22', 1000.25),
(2, 3, '9250000', '032H502KCM', '832H502KCM', '2020-02-03', '2021-02-02', 2498.96),
(3, 4, 'JAN20252', '0385AH7W21', '0385AH7W21', '2020-01-01', '2021-12-31', 3696.27);
SET IDENTITY_INSERT HealthInsurance OFF
GO

----- Messages -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- Coverage -----
SET IDENTITY_INSERT Coverage ON
INSERT Coverage([HealthInsuranceID], CoverageDetails, CoveragePrice)
VALUES
(1, 'Insurance covers the test of COVID19', 350),
(2, 'Insurance covers the test of COVID-19', 350),
(3, 'Insurance covers the protection of Body Injury', 2000);
GO

----- InsuranceCompany -----
SET IDENTITY_INSERT HealthInsuranceID ON
INSERT HealthInsuranceID([HealthInsuranceID], CompanyName, CompanyPhone, Address, City, State, ZipCode)
VALUES
(1, 'Upscate Medical Center', '8853351407', '441 S Salina St', 'Syracuse', 'NY', '13202'),
(2, 'Auburn Community Hospital', '451572536007', '899 Howard St', 'San Francisco', 'CA', '94103'),
(3, 'Humania', '3157015843', '338 S Salina ST', 'Syracuse', 'NY', '13202');
GO

----- EmergencyDept -----
SET IDENTITY_INSERT EmergencyDept ON
INSERT EmergencyDept([EmergencyDeptID], EmergencyDeptName, BusinessHour, Location, Address, City, State, ZipCode)
VALUES
(1, 'Upstate Medical Center Emergency Department', 'Open 24 hours', 'Upstate University Hospital', '750 E Adam St', 'Syracuse', 'NY', '13210'),
(2, 'Auburn Community Hospital Emergency Room', 'Open 24 hours', 'Auburn Community Hospital', '17 Lansing St', 'Auburn', 'NY', '13021'),
(3, 'UCSF Medical Center', 'Open 24 hours', 'UCSF Medical Center', '595 Parnassus Ave', 'San Francisco', 'CA', '94143');
SET IDENTITY_INSERT EmergencyDept OFF
GO

----- Message -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- PatientHistory -----
SET IDENTITY_INSERT PatientHistory ON
INSERT PatientHistory([PatientID], weight, Height, CheckIn, Checkout, Symptoms, Diagnostics, Procedures, TransferInfo)
VALUES
(1, '160lb', '170cm', '2020-04-13 00:00:00', '2020-04-13 10:00:00', 'Heart Burn and Hurt, Feeling Anxious', 'ECG, Echoangiogram, CT', 'Medical Treatment', 1),
(2, '160lb', '175cm', '2020-04-20 09:00:00', '2020-04-20 11:30:00', 'Coughing, Dizzy', 'Nucleic Acid Amplification Testing', 'Suggestion: Drink more water', null),
(3, '160lb', '164cm', '2020-05-26 08:30:00', '2020-05-28 11:30:00', 'Coughing, Fever, Difficultly Breathing', 'Nucleic Acid Amplification Testing', 'ICU', 3),
(4, '160lb', '160cm', '2020-07-01 08:30:00', '2020-07-01 10:30:00', 'Need Surgery, Car Accident attacking Body', 'CT, X-Ray', 'Surgery', 2),
(5, '220lb', '185cm', '2020-09-01 08:30:00', '2020-09-01 08:30:00', 'Dizziness, Blurry and Vomiting', 'First Check: ECG, GFR, Blood Lipids, Urinalysis', 'Medical Treatment', 5);

----- Discharge -----
SET IDENTITY_INSERT Discharge ON
INSERT Discharge([PatientID], ExpectTime)
VALUES
(1, '2020-04-18'),
(2, '2020-05-01'),
(3, '2020-06-01'),
(4, '2020-07-14'),
(5, '2020-09-13');

----- VitalSigns -----
SET IDENTITY_INSERT PatientID ON
INSERT PatientID([PatientID], BodyTemperature, PulseRate, RespirationRate, BloodPressure)
VALUES
(1, '97.7F', '130bpm', '170bpm', '170mmHg'),
(2, '98.06', '140bpm', '180bpm', '180mmHg'),
(3, '100.11', '160bpm', '160bpm', '120mmHg'),
(4, '99.98', '140bpm', '220bpm', '140mmHg'),
(5, '97.5', '120bpm', '180bpm', '150mmHg');

----- Message -----
SET IDENTITY_INSERT Message ON
INSERT Message([MessageID], PatientID, MessageText, MessageType, MessageDate)
VALUES
(1, 1, 'I have tested positive for COVID-19.', 'Text', '2020-03-23'),
(2, 1, 'I have tested negative for COVID-19.', 'Text', '2020-03-23'),
(3, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(4, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23'),
(5, 1, 'I have been diagnosed with COVID-19.', 'Text', '2020-03-23');

----- UrgentCareCenter -----
SET IDENTITY_INSERT UrgentCareCenter ON
INSERT UrgentCareCenter([UrgentCareCenterID], UrgentCareCenterName, Address, City, State, ZipCode)
VALUES
(1, 'UrgentCareCenter', '123 Main Street', 'New York', 'NY', '10001');

```

Query executed successfully.



Three screenshots of Microsoft SQL Server Management Studio (SSMS) showing the execution of three separate T-SQL scripts to insert data into tables in the 'UrgentCareCenter' database.

**Screenshot 1: Inserting data into the PrimaryCareDoc table.**

```

-- Insert into PrimaryCareDoc(PatientID, DoctorID)
INSERT INTO PrimaryCareDoc(PatientID, DoctorID)
VALUES
(1, 1),
(2, 4),
(3, 1),
(4, 3),
(5, 2);
GO

```

**Screenshot 2: Inserting data into the Referrals table.**

```

-- Insert into Referrals(PatientID, DoctorID, Date, Description)
INSERT INTO Referrals(PatientID, DoctorID, Date, Description)
VALUES
(1, 1, '2020-04-13 08:00:00', 'To See Cardiovascular Medicine Doctor'),
(2, 4, '2020-05-01 09:30:00', 'To See Doctor'),
(3, 1, '2020-05-15 10:00:00', 'To See Doctor'),
(4, 3, '2020-07-13 10:30:00', 'To See Doctor'),
(5, 2, '2020-09-08 10:30:00', 'To See Doctor');
GO

```

**Screenshot 3: Inserting data into the Centers table.**

```

-- Insert into Centers(CenterID, CenterName, BusinessHour, ProcedureCapacity, location, Address, City, State, ZipCode, Phone)
INSERT INTO Centers(CenterID, CenterName, BusinessHour, ProcedureCapacity, location, Address, City, State, ZipCode, Phone)
VALUES
(1, 'Statewide MD Urgent Care Center', 'Open 24 hours', '1000', 'W Genesee St', 'Syracuse', 'NY', '13219', '(315)810754'),
(2, 'Statewide MD Urgent Care Center', 'Open 24 hours', '1000', '109 Genesee St', 'Oneida', 'NY', '13421', '(315)2315534'),
(3, 'Statewide MD Urgent Care Center', 'Open 24 hours', '1000', 'Parnassus Campus', '1480 Parnassus Ave', 'San Francisco', 'CA', '94122', '(415)3532602');
SET IDENTITY_INSERT Centers OFF
GO

```

**Screenshot 4: Inserting data into the Facilities table.**

```

-- Insert into Facilities(RoomID, CenterID, RoomType)
INSERT INTO Facilities(RoomID, CenterID, RoomType)
VALUES
(1, 1, 'Waiting Room'),
(2, 1, 'Treatment Room'),
(3, 1, 'Emergency Room'),
(4, 1, 'Treatment Room'),
(5, 2, 'Waiting Room'),
(6, 2, 'Treatment Room'),
(7, 2, 'Emergency Room'),
(8, 2, 'Treatment Room'),
(9, 3, 'Waiting Room'),
(10, 3, 'Treatment Room'),
(11, 3, 'Emergency Room');
GO

```

**Screenshot 5: Inserting data into the MedicalEquipment table.**

```

-- Insert into MedicalEquipment(EquipmentID, CenterID, EquipmentName, Uses)
INSERT INTO MedicalEquipment(EquipmentID, CenterID, EquipmentName, Uses)
VALUES
(1, 1, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 1, 'Philips Heartstart Onsite Home Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 1, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking'),
(1, 2, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 2, 'Philips Heartstart Onsite Home Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 2, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking'),
(1, 3, '3M Littmann Classic III Stethoscope', 'Monitor and assesses a wide range of patients'),
(2, 3, 'Philips Heartstart Onsite Home Package', 'Treat people experiencing sudden cardiac arrest'),
(3, 3, 'Air Hawk Worlds Lightest Power Wheelchair', 'Help avoiding unnecessary walking');
GO

```

**Screenshot 6: Inserting data into the Departments table.**

```

-- Insert into Departments(DepartmentID, CenterID, Department, location)
INSERT INTO Departments(DepartmentID, CenterID, Department, location)
VALUES
(1, 1, 'Emergency Department', 'Emergency Area'),
(2, 1, 'Ambulance Department', 'Ambulance Area'),
(3, 1, 'Physician Department', 'Physician Area'),
(4, 1, 'Treatment Department', 'Treatment Area'),
(5, 2, 'Ambulance Department', 'Ambulance Area'),
(6, 2, 'Physician Department', 'Physician Area'),
(7, 2, 'Treatment Department', 'Treatment Area'),
(8, 3, 'Ambulance Department', 'Ambulance Area'),
(9, 3, 'Physician Department', 'Physician Area');
GO

```

Three screenshots of Microsoft SQL Server Management Studio (SSMS) showing the execution of three separate SQL scripts: `InsertTables1.sql`, `InsertTables2.sql`, and `InsertTables3.sql`. Each script inserts data into tables `Offices`, `Employees`, and `Reviews`.

**Script 1: InsertTables1.sql**

```

INSERT INTO Offices(OfficeID, OfficeName, OfficeNum, OfficePhone)
VALUES
    ('1', 'Emergency Office #1', '1801', '3152950617'),
    ('2', 'Emergency Office #2', '1802', '3152950627'),
    ('3', 'Emergency Office #3', '1803', '3152950637'),
    ('4', 'Ambulance Office #2', '1804', '3152950612'),
    ('5', 'Ambulance Office #3', '1805', '3152950613'),
    ('6', 'Physician Office #1', '1806', '3152950601'),
    ('7', 'Physician Office #2', '1807', '3152950601'),
    ('8', 'Emergency Office #2', '1808', '3152950602'),
    ('9', 'Ambulance Office #1', '1809', '3152950611'),
    ('10', 'Ambulance Office #2', '1810', '3152950612'),
    ('11', 'Physician Office', '1808', '3152950621'),
    ('12', 'Emergency Office #1', '1809', '4152505961'),
    ('13', 'Emergency Office #2', '1810', '4152505962'),
    ('14', 'Emergency Office #3', '1806', '4152505963'),
    ('15', 'Ambulance Office #1', '1807', '4152505951'),
    ('16', 'Ambulance Office #2', '1808', '4152505952'),
    ('17', 'Physician Office', '1234', '4152505951')
GO

```

**Script 2: InsertTables2.sql**

```

SET IDENTITY_INSERT Employees ON
INSERT INTO Employees(EmpID, FirstName, LastName, SSN, Address, apt, City, State, ZipCode, Phone, Email, PostJob, CurJob, Salary, Benefits, ContractType, ContractTerm, Reviews, CenterID, DeptID, OfficedID, WorkSchedule)
VALUES
    ('1', 'Sophia', 'Jackson', '123-45-6789', '111 1st st', 'apt 5', 'Syracuse', 'NY', '13210', '3155931023', 'SophiaJackson@abc.com', null, 'Nurse', 54000, 'Insurance provide', 'Leap Sum Contract', '3 years', 'Excellent', 1, 1, 'Nan - 5pm'),
    ('2', 'Ema', 'Alden', '200-12-3452', '1222 2nd st', null, 'Syracuse', 'NY', '13210', '3150203928', 'EmaAlden@abc.com', 'Nurse', 'Nurse', 79000, 'Stocks & Insurance', 'Cost Plus Contract', '2 years', 'Good', 1, 2, 'Nan - 5pm'),
    ('3', 'Olivia', 'Lorenz', '345-20-1909', '1333 2nd st', 'apt 1', 'Syracuse', 'NY', '13210', '3152009579', 'OliviaLorenz@abc.com', null, 'Nurse', 53000, 'Insurance', 'Cost Sum Contract', '2 years', 'Excellent', 1, 1, 'Nan - 5pm'),
    ('4', 'Liam', 'Shane', '123-12-1997', '100 1st st', 'apt 1', 'Syracuse', 'NY', '13210', '3151082821', 'LiamShane@abc.com', 'Nurse', 'Nurse', 83000, 'Stocks & Insurance', 'Cost Plus Contract', '10 years', 'Excellent', 1, 1, 'Nan - 5pm'),
    ('5', 'Rajesh', 'Menon', '283-10-2897', '555 2nd st', null, 'Syracuse', 'NY', '13210', '3151920386', 'RajeshMenon@abc.com', 'Driver', 'Driver', 53000, 'Health Insurance, extra PFI', 'Cost Plus Contract', '2 years', 'Good', 1, 2, 'Nan - 5pm'),
    ('6', 'Krisli', 'Sharma', '200-12-1997', '100 1st st', 'apt 1', 'Syracuse', 'NY', '13210', '3150895261', 'KrisliSharma@abc.com', null, 'Aid man', 76000, 'Health Insurance, extra PFI', 'Cost Plus Contract', '3 years', 'Good', 1, 2, 'Nan - 5pm'),
    ('7', 'Palavi', 'Roy', '200-10-2821', '777 2nd st', null, 'Syracuse', 'NY', '13210', '3150305062', 'PalaviRoy@abc.com', 'Aid man', 'Aid man', 87000, 'Health Insurance, extra PFI', 'Cost Plus Contract', '2 years', 'Excellent', 1, 1, 'Nan - 5pm'),
    ('8', 'Aman', 'Vera', '188-10-1682', '888 3rd st', null, 'Syracuse', 'NY', '13210', '3150818202', 'AmanVera@abc.com', null, 'Driver', 58000, 'Health Insurance, extra PFI', 'Cost Plus Contract', '3 years', 'Excellent', 1, 2, 4, 'Nan - 5pm'),
    ('9', 'John', 'Turner', '200-12-1997', '999 3rd st', null, 'Syracuse', 'NY', '13210', '3150999128', 'JohnTurner@abc.com', null, 'Aid man', 78000, 'Health Insurance, extra PFI', 'Incentive Contract', '2 years', 'Excellent', 1, 2, 4, 'Nan - 5pm'),
    ('10', 'Andrea', 'Maria', '205-10-2658', '345 Windell Pt', null, 'Syracuse', 'NY', '13210', '7655557878', 'AndreaMaria@abc.com', null, 'Aid man', 68000, 'Health Insurance, extra PFI', 'Cost Plus Contract', '3 years', 'Excellent', 1, 2, 4, 'Nan - 5pm')
GO

```

**Script 3: InsertTables3.sql**

```

SET IDENTITY_INSERT Reviews ON
INSERT INTO Reviews(CenterID, EmployeeID, Rating, ReviewText, DateEntered)
VALUES
    ('1', '1', '5', 'Great service!', '2024-09-09'),
    ('1', '2', '4', 'Good experience.', '2024-09-09'),
    ('1', '3', '5', 'Excellent care!', '2024-09-09'),
    ('1', '4', '4', 'Good service.', '2024-09-09'),
    ('1', '5', '5', 'Outstanding!', '2024-09-09'),
    ('2', '1', '5', 'Exceptional!', '2024-09-09'),
    ('2', '2', '4', 'Good service.', '2024-09-09'),
    ('2', '3', '5', 'Excellent care!', '2024-09-09'),
    ('2', '4', '4', 'Good service.', '2024-09-09'),
    ('2', '5', '5', 'Outstanding!', '2024-09-09'),
    ('3', '1', '5', 'Great service!', '2024-09-09'),
    ('3', '2', '4', 'Good experience.', '2024-09-09'),
    ('3', '3', '5', 'Excellent care!', '2024-09-09'),
    ('3', '4', '4', 'Good service.', '2024-09-09'),
    ('3', '5', '5', 'Outstanding!', '2024-09-09'),
    ('4', '1', '5', 'Exceptional!', '2024-09-09'),
    ('4', '2', '4', 'Good service.', '2024-09-09'),
    ('4', '3', '5', 'Excellent care!', '2024-09-09'),
    ('4', '4', '4', 'Good service.', '2024-09-09'),
    ('4', '5', '5', 'Outstanding!', '2024-09-09'),
    ('5', '1', '5', 'Great service!', '2024-09-09'),
    ('5', '2', '4', 'Good experience.', '2024-09-09'),
    ('5', '3', '5', 'Excellent care!', '2024-09-09'),
    ('5', '4', '4', 'Good service.', '2024-09-09'),
    ('5', '5', '5', 'Outstanding!', '2024-09-09'),
    ('6', '1', '5', 'Exceptional!', '2024-09-09'),
    ('6', '2', '4', 'Good service.', '2024-09-09'),
    ('6', '3', '5', 'Excellent care!', '2024-09-09'),
    ('6', '4', '4', 'Good service.', '2024-09-09'),
    ('6', '5', '5', 'Outstanding!', '2024-09-09'),
    ('7', '1', '5', 'Great service!', '2024-09-09'),
    ('7', '2', '4', 'Good experience.', '2024-09-09'),
    ('7', '3', '5', 'Excellent care!', '2024-09-09'),
    ('7', '4', '4', 'Good service.', '2024-09-09'),
    ('7', '5', '5', 'Outstanding!', '2024-09-09'),
    ('8', '1', '5', 'Exceptional!', '2024-09-09'),
    ('8', '2', '4', 'Good service.', '2024-09-09'),
    ('8', '3', '5', 'Excellent care!', '2024-09-09'),
    ('8', '4', '4', 'Good service.', '2024-09-09'),
    ('8', '5', '5', 'Outstanding!', '2024-09-09'),
    ('9', '1', '5', 'Great service!', '2024-09-09'),
    ('9', '2', '4', 'Good experience.', '2024-09-09'),
    ('9', '3', '5', 'Excellent care!', '2024-09-09'),
    ('9', '4', '4', 'Good service.', '2024-09-09'),
    ('9', '5', '5', 'Outstanding!', '2024-09-09'),
    ('10', '1', '5', 'Exceptional!', '2024-09-09'),
    ('10', '2', '4', 'Good service.', '2024-09-09'),
    ('10', '3', '5', 'Excellent care!', '2024-09-09'),
    ('10', '4', '4', 'Good service.', '2024-09-09'),
    ('10', '5', '5', 'Outstanding!', '2024-09-09')
GO

```



## D. Records

Three screenshots of Microsoft SQL Server Management Studio (SSMS) showing the results of a query named "Select\_Records.sql". The query retrieves data from various tables in the "UrgentCareCenter" database.

**Table 1: Patient Data**

```

USE UrgentCareCenter
SELECT * FROM Patients;
SELECT * FROM Billing;
SELECT * FROM Testing;
SELECT * FROM HealthInsurance;
SELECT * FROM InsuranceCompany;
SELECT * FROM PatientHealthHistory;
SELECT * FROM EmergencyDept;
SELECT * FROM VitalSigns;
SELECT * FROM Vital;
SELECT * FROM Hospital;
SELECT * FROM PrimaryCareDoc;
SELECT * FROM Referrals;
SELECT * FROM VisitHospital;
SELECT * FROM VisitAmbulatory;
SELECT * FROM Costs;
SELECT * FROM Centers;
SELECT * FROM Facilities;
SELECT * FROM MedicalEquipment;
SELECT * FROM Departments;
SELECT * FROM Offices;
SELECT * FROM Physicians;
SELECT * FROM Attending

```

**Table 2: Test Results**

```

BIRD_Vitals
Item    PayOr PaymentMethod TotalCharge
X-Ray, Pharmacy Self Credit Card 1200.00
Blood Type Test Impression Testing Self Credit Card 100.00
NCAT Laboratory, Intensive Care Unit Insurance Debit Card 2000.00
Emergency Room Insurance Check 5000.00
Bell Check 2000.00

```

**Table 3: Health Insurance Details**

```

HealthInsuranceID PatientID Policy_Number Group_Number Insurer EffectiveDate ExpiryDate Cost
1 1 2020-04-11 2020-04-18 050827213 2020-03-00 2020-03-00 200.26
2 2 90500KOW 8024802KOW B3DHS50KOW 2020-03-00 2020-03-00 200.00
3 3 4 AF#F02 03B8WVH1W21 03B8WVH1W21 2020-01-01 2020-12-31 000.00 3896.27

```

**Table 4: Coverage Details**

```

CoverageID PatientID CoveragePlan
1 1 Insurance coverage for the test of COVID-19 200.00
2 2 Insurance coverage for the test of COVID-19 300.00
3 3 Insurance coverage for the test of COVID-19 200.00

```

**Table 5: Physician Data**

```

PhysicianID CompanyName CompanyPhone Address City State ZipCode
1 1 415755877 899 Howard St San Francisco CA 94103
2 2 Blue Cross 415755877 899 Howard St San Francisco CA 94103
3 3 Humans 917011843 338 S Service Brooklyn NY 13202

```

**Table 6: Emergency Department Data**

```

EmergencyID EmergencyName EmergencyType Location Address City State ZipCode
1 1 Upstate Medical Center Emergency Department Open 24 hours Upstate University Hospital Admin Bldg Syracuse NY 13210
2 2 Asterias Hospital Emergency Room Open 24 hours Asterias Hospital TTJ Annex Bldg Admin Bldg Syracuse NY 13210
3 3 UCSF Emergency Department Open 24 hours UCSF Medical Center 505 Parnassus Ave San Fr. CA 94143

```

**Table 7: Physician Work Times**

```

PhysicianID DoctorID Date Description
1 1 2020-04-13 08:00:00 To See Cardiovascular Medicine Doctor
2 2 2020-05-01 09:30:00 To See Doctor

```

Select\_Records.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help Quick Launch (Ctrl+Q) Center

Object Explorer Results Messages

1 PatientID DoctorID Date Description

1 1 1 2020-04-13 08:00:00 To See Cardiologist Doctor

2 2 4 2020-05-01 09:30:00 To See Doctor

3 3 4 2020-07-11 10:30:00 To See Doctor

4 4 3 2020-08-01 10:30:00 To See Doctor

5 5 2 2020-08-01 10:30:00 To See Doctor

1 PatientID HospitalID Date

1 1 1 2020-04-11 08:00:00

2 2 1 2020-04-01 09:00:00

3 3 5 2020-02-02 09:00:00

4 4 3 2020-01-13 10:00:00

5 5 2 2020-01-13 10:00:00

1 MedicationID MedicineName User Directions Warning ExpiryDate Price

1 1 Aspirin Can reduce the risk of heart attack Take 1 or 2 tablets every 4 hours Irritates your stomach lining and can trigger 2021-12-20 00:00:00 20.05

2 2 Tums Treat bacterial infections Take it every 12 hours w/ or without food Heartburn (especially if the taste). 2022-03-20 00:00:00 10.71

3 3 Tylenol Relieve pain from the body Reduce fever

4 4 Omeprazole Please conduct a doctor Serious side effects can lower potassium and m. 2021-02-02 00:00:00 6.00

5 5 Cough Relief Cough relief Two tablets every 12 hours Do not misuse.

6 6 Ambien Help sleep well and fast One tablet a day Misuse leads to tragic injuries or even deaths 2021-12-20 00:00:00 20.08

7 7 Diazepam Relieve anxiety Conduct a doctor Do not misuse 2021-06-13 00:00:00 13.98

1 MedicationID ProcedureID

2 1 6 Helps people sleep well at night to reduce heart.

3 2 5 Relieve cough, suggestion drink more water

3 2 7 Take a bowel movement

5 3 5 Relieve cough

6 3 7 Reduce fever

4 4 4 Reduce anxiety

5 5 3 Treatment for hypertension

9 5 6 Helps sleep well

1 CenterName BusinessHours ProcedureCapacity Location Address City State ZipCode Phone

1 States MD Urgent Care Center Open 24 hours 300 30th W Genesee St Owosso NY 13219 3194113764

2 States MD Urgent Care Center Open 24 hours 300 109 Genesee St Owosso NY 13421 3152315530

3 States MD Urgent Care Center Open 24 hours 1000 400 Parnassus Ave San Fr. CA 94122 4153532602

1 RoomID CenterID RoomType

1 1 Waiting Room

Query executed successfully. LAPTOP-2NPL54RH\SQLEXPRESS... LAPTOP-2NPL54RH\Asus (52) UrgentCareCenter\_000001 189 rows

Select\_Records.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help Quick Launch (Ctrl+Q) Center

Object Explorer Results Messages

1 RoomID CenterID RoomType

2 2 Detection Room

3 3 1 Emergency Rm.

3 3 2 Waiting Room

5 2 2 Detection Room

6 3 2 Emergency Rm.

7 1 1 Waiting Room

8 2 3 Detection Room

9 3 3 Emergency Rm.

1 EquipmentID CenterID EquipmentName Uses

2 2 1 Philips Heartstart Onsite Home And Package Help people experiencing sudden cardiac arrest

3 3 1 Air-Heek Works Light Power Wheelchair Help avoiding unnecessary walking

4 1 2 3M Ultron Classic II Defibscope Monitor and assess a wide range of patients

5 2 2 Philips Heartstart Onsite Home And Package Treat people experiencing sudden cardiac arrest

6 3 2 Air-Heek Works Light Power Wheelchair Monitor and assess a wide range of patients

7 1 3 3M Ultron Classic II Defibscope Monitor and assess a wide range of patients

8 2 3 Philips Heartstart Onsite Home And Package Treat people experiencing sudden cardiac arrest

9 3 3 Air-Heek Works Light Power Wheelchair Help avoiding unnecessary walking

1 DeptID CenterID Department

2 2 1 Ambulance Department Ambulance Area

3 3 1 Physician Department Physician Area

4 4 1 Physician Department Physician Area

5 5 2 Ambulance Department Ambulance Area

6 6 2 Physician Department Physician Area

7 7 3 Emergency Department Emergency Area

8 8 3 Ambulance Department Ambulance Area

9 9 3 Physician Department Physician Area

1 OfficeID DeptID Department

1 1 1 Emergency Office #1 101 3152050601

1 1 2 Emergency Office #2 102 3152050602

3 3 2 Ambulance Office #1 107 3152050611

4 4 2 Ambulance Office #2 100 3152050612

5 5 3 Ambulance Office #3 108 3152050623

6 6 2 Physician Office 110 3152050621

7 7 4 Emergency Office #1 101 3152048001

8 8 4 Emergency Office #2 104 3152048002

Query executed successfully. LAPTOP-2NPL54RH\SQLEXPRESS... LAPTOP-2NPL54RH\Asus (52) UrgentCareCenter\_000001 189 rows

Select\_Records.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (52)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help Quick Launch (Ctrl+Q) Center

Object Explorer Results Messages

1 EmployeeID FirstName LastName SSN Address City State ZipCode Phone Email

4 4 Asia Liam 103-20-1048 442 E. NJ. Syracuse NY 13210 3151010261 Asia.liam@abc.com

5 5 Rajesh Manish 283-10-2057 555 1st NJ. Syracuse NY 13210 3151010238 Rajesh.Mani@abc.com

6 6 Kim Barbara 201-20-0001 123 Main St. NJ. Syracuse NY 13210 3151010240 Kim.Barbara@abc.com

7 7 Palau Roy 201-20-0271 777 2nd. NJ. Syracuse NY 13210 3152019562 Palau.Roy@abc.com

8 8 Aman Verma 189-20-1462 888 3rd. NJ. Syracuse NY 13210 3151013202 Aman.Verma@abc.com

9 9 John Smith 201-20-0002 123 Main St. NJ. Syracuse NY 13210 3151010241 John.Smith@abc.com

10 10 Andrea Marie 295-10-2056 345 W. NJ. Syracuse NY 13210 7855507878 Andrea.Marie@abc.com

11 11 Trujillo Ana 204-10-0597 1208 E. NJ. Syracuse NY 13210 5015550773 Trujillo.An@abc.com

12 12 Michael James 201-20-0003 123 Main St. NJ. Syracuse NY 13210 3151010242 Michael.James@abc.com

13 13 7 Emergency Office #2 104 4152050601

14 14 Emergency Office #3 105 3152050603

Query executed successfully. LAPTOP-2NPL54RH\SQLEXPRESS... LAPTOP-2NPL54RH\Asus (52) UrgentCareCenter\_000001 9 rows

## TESTING

### A. Views

1. In this view, I select patients who only have insurance. The view returns some of the patients' information and their insurance coverage details.

```
USE UrgentCareCenter  
GO
```

```
CREATE VIEW PatientInsurance  
AS  
SELECT FirstName + ' ' + LastName AS FullName, Gender, DOB, CoverageDetails,  
CoveragePrice  
FROM Patients JOIN HealthInsurance  
    ON Patients.PatientID = HealthInsurance.PatientID  
JOIN Coverages  
    ON HealthInsurance.HealthInsuranceID = Coverages.HealthInsuranceID;  
GO
```

```
SELECT * FROM PatientInsurance;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'UrgentCareCenter' database selected. The central pane displays the T-SQL code for creating the 'PatientInsurance' view. The code includes a comment explaining the purpose of the view, the SELECT statement, and the JOIN clauses. Below the code, the results of the query are shown in a table. The table has columns: FullName, Gender, DOB, CoverageDetails, and CoveragePrice. The data returned is as follows:

FullName	Gender	DOB	CoverageDetails	CoveragePrice
John Williams	Male	1990-05-20 00:00:00	Insurance covers the test of COVID-19	200.00
Alice White	Female	1995-01-20 00:00:00	Insurance covers the test of COVID-19	250.00
Lisa Hurst	Female	1978-08-25 00:00:00	Insurance covers the protection of Body-Hair	2000.00

At the bottom of the screen, a message indicates "Query executed successfully".

2. In this view, I select which center did patients go based on PatientHealthHistory and their attending physicians. The view returns the patient's info and center info.

```
USE UrgentCareCenter  
GO
```

```
CREATE VIEW WherePatientWent  
AS  
SELECT DISTINCT Patients.LastName + ', ' + Patients.FirstName AS Patient,  
Gender, CenterName,  
    Centers.Address + ', ' + Centers.City + ', ' + Centers.State + ', '  
+ Centers.ZipCode AS 'Center Location'  
FROM Patients JOIN PatientHealthHistory  
    ON Patients.PatientID = PatientHealthHistory.PatientID  
JOIN Attending  
    ON PatientHealthHistory.HealthHistoryID = Attending.HealthHistoryID  
JOIN Physicians
```

```

    ON Physicians.PhysicianID = Attending.PhysicianID
JOIN Employees
    ON Employees.EmployeeID = Physicians.EmployeeID
JOIN Offices
    ON Employees.OfficeID = Offices.OfficeID
JOIN Departments
    ON Offices.DeptID = Departments.DeptID
JOIN Centers
    ON Centers.CenterID = Departments.CenterID;
GO

```

```
SELECT * FROM WherePatientWent;
```

```

CREATE VIEW WherePatientWent
AS
SELECT DISTINCT Patients.LastName + ' ' + Patients.FirstName AS Patient, Gender, CenterName,
    Centers.Address + ' ' + Centers.City + ' ' + Centers.State + ' ' + Centers.ZipCode AS "Center Location"
FROM Patients
JOIN PatientHealthHistory ON Patients.PatientID = PatientHealthHistory.PatientID
JOIN Attending
    ON PatientHealthHistory.HealthHistoryID = Attending.HealthHistoryID
JOIN Physicians
    ON Physicians.PhysicianID = Attending.PhysicianID
JOIN Employee
    ON Employees.EmployeeID = Physicians.EmployeeID
JOIN Offices
    ON Employees.OfficeID = Offices.OfficeID
JOIN Departments
    ON Offices.DeptID = Departments.DeptID
JOIN Centers
    ON Centers.CenterID = Departments.CenterID;
GO

```

Results:

Patient	Gender	CenterName	Center Location
Mark,Lisa	Female	States MD Urgent Care Center	100 Genesee St, Oneida, NY 13420
Smith,John	Male	States MD Urgent Care Center	3504 W Genesee St, Syracuse, NY 13219
Webber,John	Male	States MD Urgent Care Center	3504 W Genesee St, Syracuse, NY 13219
Whitler,Arc	Female	States MD Urgent Care Center	400 Parkview Ave, San Francisco, CA 94122
Williams,Robert	Male	States MD Urgent Care Center	3504 W Genesee St, Syracuse, NY 13219

Query executed successfully.

3. In this view, I would like to find out how much each patient paid for medications as total. It's important to know how much each patient spent on their medications.

```
USE UrgentCareCenter
GO
```

```

CREATE VIEW MedicationsCost
AS
SELECT LastName + ' ' + FirstName AS FullName, SUM(Price) AS TotalCost
FROM Patients JOIN PatientHealthHistory
    ON Patients.PatientID = PatientHealthHistory.PatientID
JOIN Costs
    ON PatientHealthHistory.HealthHistoryID = Costs.HealthHistoryID
JOIN Medications
    ON Costs.MedicationID = Medications.MedicationID
GROUP BY LastName, FirstName;
GO

```

```
SELECT * FROM MedicationsCost;
```

```

CREATE VIEW MedicationsCost
AS
SELECT LastName + ' ' + FirstName AS FullName, SUM([Price]) AS TotalCost
FROM Patients JOIN PatientHealthHistory
ON Patients.PatientID = PatientHealthHistory.PatientID
JOIN Costs
ON PatientHealthHistory.HealthHistoryID = Costs.HealthHistoryID
JOIN Medications
ON Costs.MedicationID = Medications.MedicationID
GROUP BY LastName, FirstName;
GO
SELECT * FROM MedicationsCost;

```

FullName	TotalCost
Miller, Amy	7.75
Webster, John	40.54
Harris, Lisa	6.00
Jones, Mark	18.50
Williams, Robert	20.00

4. In this view, we could find out which center, department, and office each physician works. We use this view to help further patients find physicians easier by their locations. The view returns physician Locations as well as physician information.

```

USE UrgentCareCenter
GO

CREATE VIEW PhysicianLoc
AS
SELECT Employees.LastName + ' ' + Employees.FirstName AS Physician,
       Type, WorkSchedule, CenterName,
       Centers.Address + ', ' + Centers.City + ', ' + Centers.State + ', '
+ Centers.ZipCode AS 'Center Location',
       Departments.Location AS 'Department Location', Offices.OfficeNum AS
'Room Number'
FROM Physicians JOIN Employees
    ON Employees.EmployeeID = Physicians.EmployeeID
JOIN Offices
    ON Offices.OfficeID = Employees.EmployeeID
JOIN Departments
    ON Departments.DeptID = Offices.DeptID
JOIN Centers
    ON Centers.CenterID = Departments.CenterID;
GO

SELECT * FROM PhysicianLoc;

```

```

USE UrgentCareCenter
GO
/*
In this view, we could find which center, department and office each physician works.
We use this view to help further patients find physician easier by their locations.
The view returns physician locations as well as physician information
*/
CREATE VIEW PhysicianLoc
AS
SELECT Employees.LastName + ', ' + Employees.FirstName AS Physician,
       Type, WorkSchedule, CenterName,
       Centers.Address + ', ' + Centers.City + ', ' + Centers.State + ', ' + Centers.ZipCode AS 'Center_location',
       Departments.Location AS 'Department_location', Offices.OfficeName AS 'Office_Number'
  FROM Physicians JOIN Employees
    ON Employees.EmployeeID = Physicians.EmployeeID
   JOIN Offices ON Offices.OfficeID = Employees.EmployeeID
   JOIN Departments ON Departments.DeptID = Offices.DeptID
  JOIN Centers
    ON Centers.CenterID = Departments.CenterID
 GO
SELECT * FROM PhysicianLoc;

```

Results (4 rows)

Physician	Type	WorkSchedule	CenterName	Center Location	DepartmentLocation	Room Number	
John Doe	Physician	8am-7pm	State MO	Urgent Care Center	450 Pennsylvania Ave, San Francisco, CA 94112	Emergency Area	100
Hans Mose	Nurse	11am-7pm	State MO	Urgent Care Center	450 Pennsylvania Ave, San Francisco, CA 94112	Ambulance Area	100
Fred Olsen	Nurse	9am-1pm	State MO	Urgent Care Center	450 Pennsylvania Ave, San Francisco, CA 94112	Emergency Area	100
Mary Brown	Administrative	8am-7pm	State MO	Urgent Care Center	450 Pennsylvania Ave, San Francisco, CA 94112	Pharmacy	123

## B. Stored Procedures

- In this procedure, I would like to see how much each position of an employee from the center earns averagely in a state. The procedure will return a current job title, avg earns, locations.

```

USE UrgentCareCenter
GO

CREATE PROC spAvgEarning
    @State      varchar(50) = NULL,
    @Position    varchar(50) = NULL
AS
IF @State IS NULL OR @Position IS NULL
    THROW 50001, 'Error due to @State and @Position require non-null values!', 1;

SELECT CurJob AS Position, Centers.State AS State, AVG(Salary) AS 'Average Salary'
FROM Employees JOIN EmployeesInfo
    ON Employees.EmployeeID = EmployeesInfo.EmployeeID
JOIN Offices
    ON Offices.OfficeID = Employees.OfficeID
JOIN Departments
    ON Departments.DeptID = Offices.DeptID
JOIN Centers
    ON Centers.CenterID = Departments.CenterID
WHERE Centers.State LIKE @State AND
    CurJob LIKE @Position
GROUP BY CurJob, Centers.State;
GO

BEGIN TRY
    EXEC spAvgEarning 'NY', 'Physician'
END TRY
BEGIN CATCH
    PRINT 'An Error Occurred!';

```

```
PRINT 'Error Number: ' + CONVERT(varchar, ERROR_NUMBER());
PRINT 'Error Message: ' + CONVERT(varchar, ERROR_MESSAGE());
END CATCH

GO
```

Object Explorer

- LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter
- Databases
- Security
- Replication
- Maintenance
- Management
- XEvent Profiler

Toolbox

ProcList [LAPTOP-2NPL54RH\Ass (S)]

```
CREATE PROC spPhysician
    @State varchar(50) = NULL,
    @Position varchar(50) = NULL
AS
    IF (@State IS NULL OR @Position IS NULL
        THROW 50001, 'Error due to @State and @Position require non-null values!', 1);
    SELECT CurJob AS Position, Centers.State AS State, AVG(Salary) AS 'Average Salary'
    FROM Employees
    JOIN EmployeeInfo ON Employees.EmployeeID = EmployeeInfo.EmployeeID
    JOIN Offices ON Employees.OfficeID = Offices.OfficeID
    JOIN Departments ON Departments.DeptID = Offices.DeptID
    JOIN Centers ON Centers.CenterID = Departments.CenterID
    WHERE EmployeeInfo.Position = @Position
    GROUP BY CurJob, Centers.State;
    GO

BEGIN TRY
    EXEC spAvgGaining 'NY', 'Physician'
END TRY
BEGIN CATCH
    PRINT 'An Error Occurred!';
    PRINT 'Error Number: ' + CONVERT(varchar, ERROR_NUMBER());
    PRINT 'Error Message: ' + CONVERT(varchar, ERROR_MESSAGE());
END CATCH
```

Results

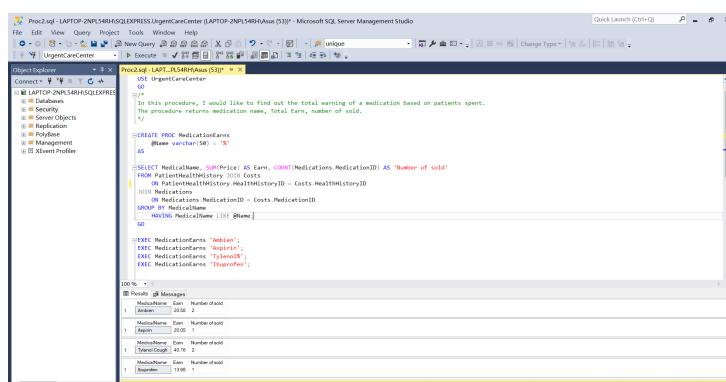
Position	State	Average Salary
(Physician)	NY	11833.033

Query executed successfully.

2. In this procedure, I would like to find out the total earning of a medication based on patients spent. The procedure returns medication name, Total Earn, number of sold.

USE UrgentCareCenter  
GO

```
CREATE PROC MedicationEarns
    @Name varchar(50) = '%'
AS
SELECT MedicalName, SUM(Price) AS Earn, COUNT(Medications.MedicationID) AS
'Number of sold'
FROM PatientHealthHistory JOIN Costs
    ON PatientHealthHistory.HealthHistoryID = Costs.HealthHistoryID
JOIN Medications
    ON Medications.MedicationID = Costs.MedicationID
GROUP BY MedicalName
    HAVING MedicalName LIKE @Name;
GO
EXEC MedicationEarns 'Ambien';
EXEC MedicationEarns 'Aspirin';
EXEC MedicationEarns 'Tylenol%';
EXEC MedicationEarns 'Ibuprofen';
```



## C. User Defined Functions

1. In this function, I would like to see each patient spent after insurance whoever has insurance or not, which means that the (**BalanceDue = total cost in billing - insurance coverage + insurance they paid**). The function returns Patient info, what disease he/she got, items he/she used, and balance due.

```
USE UrgentCareCenter
GO

CREATE FUNCTION fnMaxCost()
    RETURNS TABLE

    RETURN
        (SELECT LastName + ', ' + FirstName AS FullName,
            Test, Items, PayOr, TotalCharge - IsNull(CoveragePrice, 0) +
            IsNull(Cost, 0) AS BalanceDue
        FROM Patients JOIN Billing
            ON Patients.PatientID = Billing.BillID
        JOIN Testing
            ON Patients.PatientID = Testing.TestingID
        LEFT JOIN HealthInsurance
            ON Patients.PatientID = HealthInsurance.PatientID
        LEFT JOIN Coverages
            ON HealthInsurance.HealthInsuranceID = Coverages.HealthInsuranceID);
GO
```

```
SELECT * FROM dbo.fnMaxCost()
ORDER BY BalanceDue DESC
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database 'UrgentCareCenter' with its tables and objects. The main pane displays the T-SQL code for creating the function 'fnMaxCost'. Below the code, a results grid shows the output of the query 'SELECT \* FROM dbo.fnMaxCost() ORDER BY BalanceDue DESC'. The results grid has columns: FullName, Test, Items, PayOr, and BalanceDue. The data is as follows:

	FullName	Test	Items	PayOr	BalanceDue
1	James, Mike	Head, Neck	X-Ray, Pharmacy	Sell	12000.00
2	Kathy, Linda	Body, Head, Abdomen, Room	Medication, Pharmacy	Insurance	4148.95
3	Whitaker, Ann	Covid19	NCAT, Laboratory, Intensive Care Unit	Insurance	4148.95
4	Webster, John	Hypertension	EMG/ECHO	Sell	2800.00
5	Williams, Robert	Covid19	Nucleic Acid Amplification Testing	Insurance	1096.25

2. In this function, I would like to use the hospital visit date range by patients to show which hospital they visited. The function returns patients' info, hospital info, and the time they matched.

```
USE UrgentCareCenter
GO
```

```

CREATE FUNCTION fnVisitedHospital
    (@checkIn      smalldatetime,
     @checkOut     smalldatetime)
    RETURNS TABLE

RETURN
    (SELECT Patients.LastName + ', ' + Patients.FirstName AS FullName,
           HospitalName, Date AS 'Visit Date'
    FROM Patients JOIN PatientHealthHistory
        ON Patients.PatientID = PatientHealthHistory.PatientID
    JOIN VisitHospital
        ON PatientHealthHistory.HealthHistoryID =
    VisitHospital.HealthHistoryID
    JOIN Hospitals
        ON VisitHospital.HospitalID = Hospitals.HospitalID
    WHERE Date BETWEEN @checkIn AND @checkOut);
GO

```

```

SELECT * FROM dbo.fnVisitedHospital('2020-07-01', '2020-12-12')
SELECT * FROM dbo.fnVisitedHospital('2020-05-01', '2020-08-13')

```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'Func2.sql - [LAPTOP-2NPL54RHSQLEXPRESS].UrgentCareCenter (LAPTOP-2NPL54RH Aus (53)) - Microsoft SQL Server Management Studio' displays the T-SQL code for creating the function. Below the code, two result sets are shown in the 'Results' tab:

FulName	HospitalName	VisitDate
Williams, Robert	Owens Health Hospital	2020-07-13 10:00:00
Wade, John	Urgent University Hospital	2020-07-13 10:00:00

FulName	HospitalName	VisitDate
Williams, Robert	Owens Health Hospital	2020-05-01 09:30:00
Wade, John	Urgent University Hospital	2020-05-01 09:30:00
Market, Lisa	Urgent University Hospital	2020-07-13 10:00:00

At the bottom, a message indicates 'Query executed successfully.'

## D. Triggers

1. In this trigger, every patient must have billing information. So, the trigger avoids billing of patients to be deleted. Once billing is deleted without deleting the corresponded patient, it will throw an error.

```

USE UrgentCareCenter
GO

CREATE TRIGGER Billing_DELETE_RI
    ON Billing
    AFTER DELETE
AS
    IF EXISTS (SELECT * FROM Deleted JOIN Patients
                ON Deleted.BillID = Patients.PatientID)
        BEGIN
            ;
            THROW 50002, 'BillingID in use.', 1;
            ROLLBACK TRAN;
        END;
GO

```

```
DELETE FROM Billing
WHERE BillID = 1
```

```
USE UrgentCareCenter
GO

/*
In this trigger, every patient has to have billing information.
So the trigger avoids billing of patients to be deleted.
Once billing is deleted without deleting corresponding patient, it will throw error
*/
CREATE TRIGGER Billing_DELETE_RT
ON Billing
AFTER DELETE
AS
IF EXISTS (SELECT * FROM Deleted JOIN Patients
            ON Deleted.BillID = Patients.PatientID)
BEGIN
    THROW 50002, 'BillingID in use.', 1;
    ROLLBACK TRAN;
END;
GO

DELETE FROM Billing
WHERE BillID = 1
```

Completion time: 2020-11-23T23:36:55.1619122+05:00

Query completed with errors.

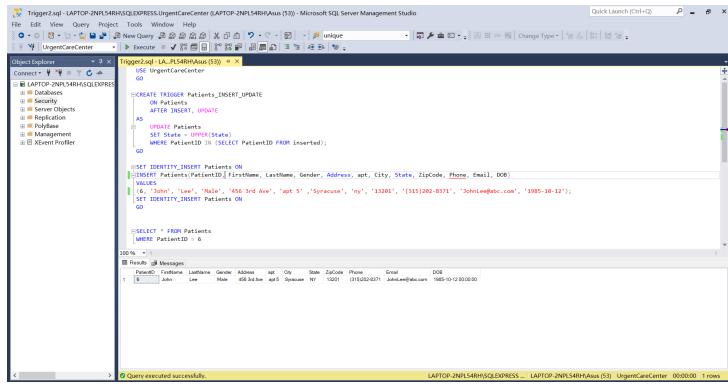
2. In this Trigger, whenever inserting a patient or the existing patient info is updated, the state where they live must be uppercase.

```
USE UrgentCareCenter
GO

CREATE TRIGGER Patients_INSERT_UPDATE
ON Patients
AFTER INSERT, UPDATE
AS
UPDATE Patients
SET State = UPPER(State)
WHERE PatientID IN (SELECT PatientID FROM inserted);
GO

SET IDENTITY_INSERT Patients ON
INSERT Patients(PatientID, FirstName, LastName, Gender, Address, apt, City,
State, ZipCode, Phone, Email, DOB)
VALUES
(6, 'John', 'Lee', 'Male', '456 3rd Ave', 'apt 5', 'Syracuse', 'ny', '13201',
'(315)202-8371', 'JohnLee@abc.com', '1985-10-12');
SET IDENTITY_INSERT Patients ON
GO

SELECT * FROM Patients
WHERE PatientID = 6;
```



## E. Transactions

1. In this transaction, I would like to add a new physician to the Physicians' table. Once inserting a new physician, the employee is also increased by 1. I would select the latest employeeID + 1 as the new employeeID for the new physician. Also, I would select the latest physicianID + 1 as the new PhysicianID for the new physician. Then, I insert the physician's information into the Employees table.

```
USE UrgentCareCenter
GO

IF EXISTS(SELECT * FROM sys.triggers
          WHERE OBJECT_ID = OBJECT_ID('Physician_New'))
DROP TRIGGER [dbo].[Physician_New]
GO

BEGIN TRAN
    DECLARE @EmployeeID int
    SELECT @EmployeeID = MAX(EmployeeID) + 1
    FROM Employees;

    DECLARE @PhysicianID int
    SELECT @PhysicianID = MAX(PhysicianID) + 1
    FROM Physicians;

    SET IDENTITY_INSERT Employees ON
    INSERT INTO Employees(EmployeeID, OfficeID, FirstName, LastName, Address,
apt, City, State, ZipCode, Phone, Email, WorkSchedule)
    VALUES (@EmployeeID, 17, 'John', 'Han', '65 Walnut St', 'apt 6', 'San
Francisco', 'CA', '94101', '(415)202-0395', 'JohnHan@abc.com', '8am - 7pm');
    SET IDENTITY_INSERT Employees OFF

    INSERT INTO EmployeesInfo(EmployeeID, SSN, PastJob, CurJob, Salary,
Benefits, ContractType, ContractTerm, Reviews)
    VALUES (@EmployeeID, '182-28-1100', 'Physician', 'Physician', 103500,
'Health Insurance', 'Cost Plus Contract', '3 years', 'Excellent');

    SET IDENTITY_INSERT Physicians ON
```

```
INSERT Physicians(PhysicianID, EmployeeID, Type)
VALUES (@PhysicianID, @EmployeeID, 'Family Physician');
SET IDENTITY_INSERT Physicians OFF
```

**COMMIT TRAN;**

Transact - LAPTOP-NP54RH\SQLEXPRESS\UrgentCareCenter (LAPTOP-2NPL54RH\Asus (5)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help New Help Patient Change Type

Object Explorer

Connect to... UrgentCareCenter

LAPTOP-2NPL54RH\SQLEXPRESS (SQL Database Engine)

System Databases

Database Snapshots

AP

Replicates

MyGuitarShop

OnlineCapSystem

ProductOrders

Shopping

Software2

UrgentCareCenter

Security

User Objects

Replication

PolyBase

Management

XEvent Profiler

SQLQuery33.sql - [L:\NP54RH\Asus (5)] Transact - LAPTOP-NP54RH\Asus (5) \* x

USE UrgentCareCenter

```
IF EXISTS (SELECT * FROM sys.triggers WHERE name = 'Object_ID_Physician_New')
DROP TRIGGER [dbo].[Physician_New]
GO

BEGIN TRAN
DECLARE @EmployeeID int
SELECT @EmployeeID = MAX([EmployeeID]) + 1
FROM Employees;

DECLARE @PhysicianID int
SELECT @PhysicianID = MAX([PhysicianID]) + 1
FROM Physicians;

SET IDENTITY_INSERT Employees ON
INSERT INTO Employees([EmployeeID, FirstName, LastName, SSN, Address, apt, City, State, ZipCode, Phone, Email, PastJob, CurJob, Salary, Benefits, ContractType, ContractTerm, Review, CenterID, DeptID, OfcExtID, WorkSchedule])
VALUES (@EmployeeID, 'John', 'Han', '182-28-1100', '65 Walnut St.', 'apt 6', 'San Francisco', 'CA', '94108', '(415)2802095', 'JohnHan@uac.com', 'Physician', '1035000, 'Health Insurance ', 'Cost Plus Contract', '3 years', 'Excellent', 3, 9, 17, '8am - 7pm');
SET IDENTITY_INSERT Employees OFF

SET IDENTITY_INSERT Physicians ON
INSERT INTO Physicians([PhysicianID, EmployeeID, Type])
VALUES (@PhysicianID, @EmployeeID, 'Family Physician');
SET IDENTITY_INSERT Physicians OFF
COMMIT TRAN;
```

100 % 4 Messages

(1 row affected)

(1 row affected)

100 % 0 rows

Query executed successfully.

## Before Transaction Execute

SQLQuery21.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Aus (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute Save As... Refresh Unique

Object Explorer

Connect to... UrgentCareCenter

Databases Security Logins Objects Replication PolyBase Management XEvent Profiler

SQLQuery21.sql - LAPTOP-2NPL54RH\Aus (54) - Trans.sql - (LAPTOP-2NPL54RH\Aus (53))

```
USE UrgentCareCenter;
GO

--SELECT * FROM Employees;
--SELECT * FROM EmployeesInfo;
--SELECT * FROM Physicians;
```

Results (0 rows)

Messages

EmployeeID	SSN	PwdHash	OutDate	Sal	Benefit	ContractType	Review
33	33-10-0867	9876543210	60000.00	Health Insurance, extra PTO	Cost Plus Contract	3 years	Excellent
34	29-10-0301	1234567890	78000.00	Health Insurance, extra PTO	Cost Plus Contract	10 years	Excellent
35	29-10-0302	9876543210	78000.00	Health Insurance, extra PTO	Cost Plus Contract	10 years	Excellent
36	36-15-0000	1234567890	78000.00	Summer, Jose	Part Time	11 months	Good
37	37-17-0000	9876543210	78000.00	Winter, John	Part Time	11 months	Good
38	38-16-0000	1234567890	78000.00	Lincoln, Michael	PTO & Benefits	1 year	Good
39	39-17-0000	9876543210	78000.00	Spencer, Alden	Part Time	11 months	Good
40	40-17-0000	1234567890	78000.00	Lorraine, Linda	Part Time	11 months	Good

Physicals

PhysicianID	EmployeeID	Type
1	1	Family Physician
2	2	Internists
3	3	Surgeon
4	4	Orthopedic
5	5	Family Physician
6	6	Internists
7	7	Family Physician
8	8	Internists

Query executed successfully.

## After Transaction Execute

The screenshot shows the Microsoft SQL Server Management Studio interface with a query window titled "SQLQuery21.sql - LAPTOP-2NPL54RH\SQLEXPRESS.UrgentCareCenter (LAPTOP-2NPL54RH\Asus (54))". The query window contains the following T-SQL code:

```

USE UrgentCareCenter;
GO
SELECT * FROM Employees;
SELECT * FROM EmployeesInfo;
SELECT * FROM Physicians;

```

The results pane displays 91 rows of data across three tables. The first two rows of the Physicians table are highlighted with red circles, and the entire results grid is also circled in red.

EmployeeID	SSN	Patiable	ContractTerm	Billing	ContractType	ContractTerm	Reviews				
34	34	14	Moon	U	1775 N Bovine Ave	NULL	San Fr., CA	94101	(415)502-8571	Moon@abc.com	11am - 7pm
35	35	15	Clesea	Fran	1234 Main St	NULL	San Fr., CA	94101	(415)504-8271	CleseaFran@abc.com	8am - 8pm
36	36	16	Summer	Wendy	1345 Elm St	NULL	San Fr., CA	94101	(415)502-8571	SummerWendy@abc.com	8am - 8pm
37	37	16	Marky		7172 Maligen Ave	NULL	San Fr., CA	94101	(415)502-0000	MarkyMarky@abc.com	11am - 11pm
38	38	16	Lincoln	Merson	4567 Rn 70 E	NULL	San Fr., CA	94101	(415)522-1111	LincolnMerson@abc.com	11am - 11pm
39	39	17	John	Leinen	2722 Baker St	NULL	San Fr., CA	94101	(415)502-0000	JohnLeinen@abc.com	8am - 7pm
40	40	17	John	Han	65 Wm St	appt	San Fr., CA	94101	(415)522-0403	JohnHan@abc.com	8am - 7pm
41	41	17	John	Han	65 Wm St	appt	San Fr., CA	94101	(415)522-0390	JohnHan@abc.com	8am - 7pm
2	2	15	Internals								
3	3	16	Surgeon								
4	4	17	Anesthesiologist								
5	5	28	Family Physician								
6	6	29	Internals								
7	7	30	Family Physician								
8	8	40	Internals								
9	9	41	Family Physician								

At the bottom of the results pane, it says "Query executed successfully." and "LAPTOP-2NPL54RH\SQLEXPRESS ... LAPTOP-2NPL54RH\Asus (54) UrgentCareCenter 0000000 91 rows".

## F. Scripts

- I would like to create Roles such as Admin, Physician, RN, Receptionist, and Accounting, and give them partial permissions to allow them to view or manipulate specific tables.

```

USE UrgentCareCenter
GO

-- Have all permissions for the DB
CREATE ROLE AdminEntry
ALTER ROLE db_owner ADD MEMBER AdminEntry;
GO

-- Have permissions on attending between Physicians and Patients
CREATE ROLE PhysicianEntry
GRANT UPDATE, INSERT, DELETE, SELECT ON Attending TO PhysicianEntry;
GO

-- Have permissions to view and manipulate patients' health history information
CREATE ROLE RNEntry
GRANT UPDATE, INSERT, DELETE, SELECT ON PatientHealthHistory TO RNEntry;
GRANT UPDATE, INSERT, DELETE, SELECT ON Vitals TO RNEntry;
GRANT UPDATE, INSERT, DELETE, SELECT ON Discharge TO RNEntry;
GRANT UPDATE, INSERT, DELETE, SELECT ON Testing TO RNEntry;
GO

-- Have permissions to view and manipulate patients' information and their health insurance
CREATE ROLE ReceptionistEntry
GRANT UPDATE, INSERT, DELETE, SELECT ON Patients TO ReceptionistEntry;
GRANT SELECT ON PatientHealthHistory TO ReceptionistEntry;
GRANT SELECT ON HealthInsurance TO ReceptionistEntry;
GRANT SELECT ON InsuranceCompany TO ReceptionistEntry;
GRANT SELECT ON Coverages TO ReceptionistEntry;
GO

-- Have permissions to view patients' billing and their health insurance coverages.
CREATE ROLE AccountingEntry
GRANT SELECT ON Patients TO AccountingEntry;
GRANT SELECT ON Billing TO AccountingEntry;

```

```

GRANT SELECT ON HealthInsurance TO AccountingEntry;
GRANT SELECT ON Coverages TO AccountingEntry;
GO

```

```

USE [UrgentCareCenter]
GO
-- Have all permissions for the DB
CREATE ROLE AdminEntry
ALTER ROLE db_owner ADD MEMBER AdminEntry;
GO

-- Have permission on attending between Physicians and Patients
CREATE ROLE PhysicianEntry
GRANT UPDATE, INSERT, DELETE, SELECT ON Attending TO PhysicianEntry;
GO

-- Have permissions to view and manipulate patients' health history information
CREATE ROLE RNEntry
GRANT UPDATE, INSERT, DELETE, SELECT ON PatientHealthHistory TO RNEntry;
GRANT SELECT, INSERT, DELETE, SELECT ON Vitalsto RNEntry;
GRANT UPDATE, INSERT, DELETE, SELECT ON Discharge TO RNEntry;
GRANT UPDATE, INSERT, DELETE, SELECT ON Testing TO RNEntry;
GO

-- Messages
Commands completed successfully.

Completion time: 2020-11-07T19:28:04.5800242+05:00

```

```

USE [UrgentCareCenter]
GO
-- Have permissions to view and manipulate patients' information and their health insurance
CREATE ROLE ReceptionistEntry
GRANT SELECT, INSERT, UPDATE, DELETE ON Patients TO ReceptionistEntry;
GRANT SELECT ON PatientHealthHistory TO ReceptionistEntry;
GRANT SELECT ON HealthInsurance TO ReceptionistEntry;
GRANT SELECT ON InsuranceCompany TO ReceptionistEntry;
GRANT SELECT ON Coverages TO ReceptionistEntry;
GO

-- Have permissions to view patients' billing and their health insurance coverages.
CREATE ROLE AccountingEntry
GRANT SELECT ON Patients TO AccountingEntry;
GRANT SELECT ON Billing TO AccountingEntry;
GRANT SELECT ON HealthInsurance TO AccountingEntry;
GRANT SELECT ON Coverages TO AccountingEntry;
GO

-- Messages
Commands completed successfully.

Completion time: 2020-11-07T19:28:09.5800242+05:00

```

2. I would like to assign people with login username and password to access the corresponding roles so that they can do the job in the center.

```

USE UrgentCareCenter
GO

-- Assign Robert as administrator to manage all database
CREATE LOGIN Administrator WITH PASSWORD = '123456',
DEFAULT_DATABASE = UrgentCareCenter;
CREATE USER Robert FOR LOGIN Administrator;
ALTER ROLE AdminEntry ADD MEMBER Robert;
GO

-- Assign Berglund as one of the physicians to manage attending between physicians and patients
CREATE LOGIN Physician WITH PASSWORD = '123456',
DEFAULT_DATABASE = UrgentCareCenter;
CREATE USER Berglund FOR LOGIN Physician;
ALTER ROLE PhysicianEntry ADD MEMBER Berglund;
GO

-- Assign Emma as one of the RNs to manage patients' health history information
CREATE LOGIN Nurse WITH PASSWORD = '123456',

```

```

DEFAULT_DATABASE = UrgentCareCenter;
CREATE USER Emma FOR LOGIN Nurse;
ALTER ROLE RNEntry ADD MEMBER Emma;
GO

-- Assign John as receptionist to view and manipulate patients' information and their health insurance
CREATE LOGIN Reception WITH PASSWORD = '123456',
DEFAULT_DATABASE = UrgentCareCenter;
CREATE USER John FOR LOGIN Reception;
ALTER ROLE ReceptionistEntry ADD MEMBER John;
GO

-- Assign Kevin as Accounting to view patients' billing and their health insurance coverages
CREATE LOGIN Accounting WITH PASSWORD = '123456',
DEFAULT_DATABASE = UrgentCareCenter;
CREATE USER Kevin FOR LOGIN Accounting;
ALTER ROLE AccountingEntry ADD MEMBER Kevin;
GO

```

Users.sql - LAPTOP-2NPL54RH\SQLEXPRESS\UrgentCareCenter (LAPTOP-2NPL54RH\Asus (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer    Results pane    Messages

UrgentCareCenter

... Assign Robert as administrator to manage all database objects

CREATE LOGIN Administrator WITH PASSWORD = '123456';  
 DEFAULT\_DATABASE = UrgentCareCenter;  
 CREATE USER Robert FOR LOGIN Administrator;  
 ALTER ROLE AdminEntry ADD MEMBER Robert;  
 GO

... Assign Berglund as one of the physicians to manage attending between physicians and patients

CREATE LOGIN Physician WITH PASSWORD = '123456';  
 DEFAULT\_DATABASE = UrgentCareCenter;  
 CREATE USER Berglund FOR LOGIN Physician;  
 ALTER ROLE PhysicianEntry ADD MEMBER Berglund;  
 GO

... Assign Emma as one of the RNs to manage patients' health history information

CREATE LOGIN Nurse WITH PASSWORD = '123456';  
 DEFAULT\_DATABASE = UrgentCareCenter;  
 CREATE USER Emma FOR LOGIN Nurse;  
 ALTER ROLE RNEntry ADD MEMBER Emma;  
 GO

100 %

Messages

Commands completed successfully.

Completion time: 2020-11-07T19:28:07.4601140+08:00

LAPTOP-2NPL54RH\SQLEXPRESS ... LAPTOP-2NPL54RH\Asus (54) UrgentCareCenter 000000 0 rows

Users.sql - LAPTOP-2NPL54RH\SQLEXPRESS\UrgentCareCenter (LAPTOP-2NPL54RH\Asus (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer    Results pane    Messages

UrgentCareCenter

-- Assign John as receptionist to view and manipulate patients' information and their health insurance

CREATE LOGIN Reception WITH PASSWORD = '123456';  
 DEFAULT\_DATABASE = UrgentCareCenter;  
 CREATE USER John FOR LOGIN Reception;  
 ALTER ROLE ReceptionistEntry ADD MEMBER John;  
 GO

-- Assign Kevin as Accounting to view patients' billing and their health insurance coverages

CREATE LOGIN Accounting WITH PASSWORD = '123456';  
 DEFAULT\_DATABASE = UrgentCareCenter;  
 CREATE USER Kevin FOR LOGIN Accounting;  
 ALTER ROLE AccountingEntry ADD MEMBER Kevin;  
 GO

100 %

Messages

Commands completed successfully.

Completion time: 2020-11-07T19:28:07.4601140+08:00

LAPTOP-2NPL54RH\SQLEXPRESS ... LAPTOP-2NPL54RH\Asus (54) UrgentCareCenter 000000 0 rows

## **CONCLUSION**

### **A. Analysis**

In this project, I cannot guarantee the database has a perfect design, but most of the parts are logical, and it is considered as a valid reason because all implementations such as views, stored procedures, user defined functions, etc., are trivial scenarios and meaningful. For example, InsuranceCompany and Coverags can be apart from HealthInsurance. Physicians can be Employees in the center. Each employee has his/her own department and office, which means Centers have several departments, and each department has several offices. All the rest of the tables' relationships are shown in the E-R diagram.

### **B. Problem Encountered**

In this project, I encountered a lot of problems and issues in the design part. I spent a couple of days making a proposal and pattern how to design it more logical and efficient for valid reasons such as implementing on views, stored procedures, functions, and many more. When I completed the design, I found lots of unreasonable problems based on the E-R diagram generated by the proposal. I noticed that Physicians can be Employees, Health Insurance can be separated into several tables according to the Third Normal Form. In addition, some related tables have repetitive attributes, for example, the Physicians table has FirstName and LastName, and the Employees table also has FirstName and LastName. I also spent the time to remove all repetitive attributes in the database. This kind of problem helps me understand how database design is important because if there's a terrible design on the database, it will be lots of trouble for those who maintain the database and manipulate data. For example, inserting data will be inconsistent among all tables.

### **C. Remarks**

In this project, I learn a lot and have a deeper understanding of database design, implementation, and those testing sessions such as stored procedure, functions, triggers, etc. I'm much clearer and more familiar with the skills to design databases and manipulate data. However, the hardest part is still database design because I have to consider all possibilities with views, business logics and transactions to be incorporated into the design.