

**CSE 581**  
**Project 2**  
**University Medical Center Database**

Jinbo Wang

216001833

[jwang302@syr.edu](mailto:jwang302@syr.edu)

## Contents

<b>Abstract .....</b>	<b>3</b>
<b>Database Design.....</b>	<b>3</b>
<b>Database Implementation .....</b>	<b>5</b>
<b>Data Testing .....</b>	<b>23</b>
<b>Conclusion .....</b>	<b>35</b>

## Abstract

This project designs, implements, and tests a University Medical Center (UMC) database. This project presents the relationship between patient, employee, and facility of a hospital database. In this project, it has 4 parts, Data Design, Data Implementation, Database Testing, and Conclusion. Data Design part includes the E/R diagram. Data Implementation part has table creation and data insertion. In the Testing part, several views, stored procedures, user defined functions, triggers, transaction, and scripts are shown to verify the database design. Last but not least, the conclusion part contains project analysis and remarks.

## I. Database Design

### 1. Introduction:

This University Medical Center (CMU) is hospital system database which mainly include patient, employee(physician, nurse, other staff), and facility. Every table is designed according to daily life. In this database, it totally use 21 tables to describe the system of hospital.

### 2. Table description

There are some basic tables such as patient, visitors, employee, facility, lab, pharmacy, and imaging. These tables include basic information.

In order to give the relationship to tables, we need to set some keys. (shown in constraints)

I find that most relationship is one to many. And for the one-to-one relationship, it basically used to explore the information which may not be suitable in one table. And for the many-to-many relationship, I add one linking table, which can make the table form many-to-many to two one-to-many relationships such as patient to visitors.

I listed the relationship between tables:

The relation between Facility and NonMedicalStaffSchedule is one-to-many.

The relation between Facility and SurgeryRoomSchedule is one-to-many.

The relation between Facility and InpatientRoomSchedule is one-to-many.

The relation between Facility and EquipmentReservations is one-to-many.

The relation between Facility and Facilitydetail is one-to-many.

The relation between imaging and Facilitydetail is one-to-many.

The relation between Lab and Facilitydetail is one-to-many.

The relation between Pharmacy and Facilitydetail is one-to-many.

The relation between Patient and Costs is one-to-many.

The relation between Patient and InsuranceCoverage is one-to-many.

The relation between Patient and PatientMedicalRecords is one-to-many.

The relation between Patient and VisitingInfo is one-to-many.

The relation between Patient and Billing is one-to-many.

The relation between Visitors and VisitingInfo is one-to-many.

The relation between Patient and Costs is one-to-many.

The relation between Patient and PatientAppointments is one-to-many.

The relation between Employee and PatientAppointments is one-to-many.

The relation between Employee and NonMedicalStaffSchedule is one-to-many.

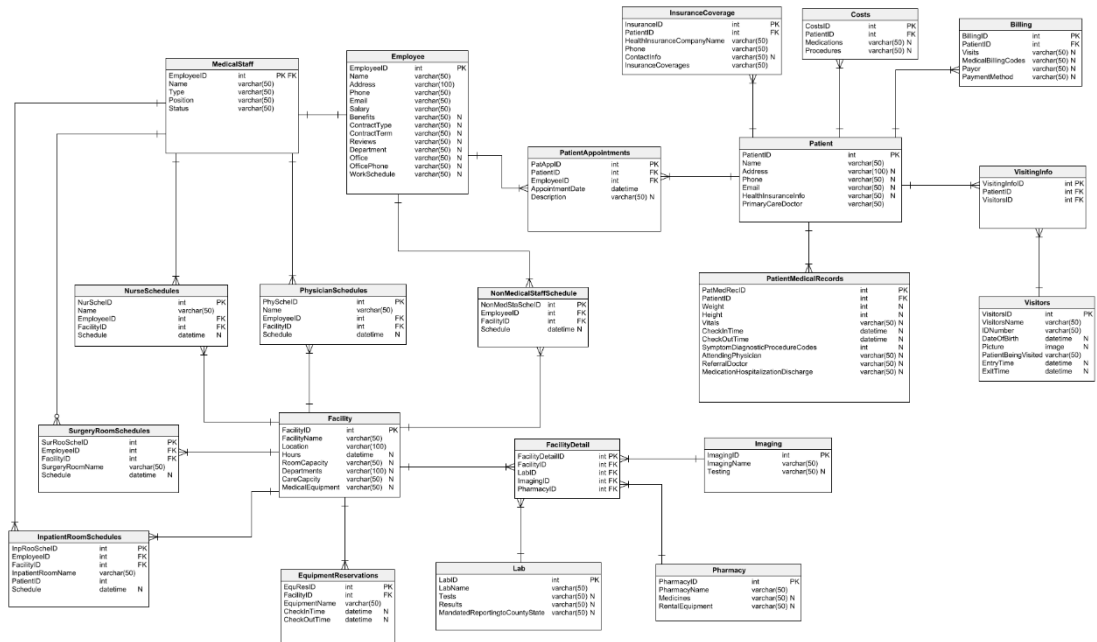
The relation between Employee and MedicalStaff is one-to-one.

The relation between MedicalStaff and PhysicianSchedules is one-to-many.

The relation between MedicalStaff and NurseSchedules is one-to-many.

### 3. Constraints

Table	Primary Key	Foreign Key
Patient	PatientID	
InsuranceCoverage	InsuranceID	PatientID
Costs	CostsID	PatientID
Billing	BillingID	PatientID
VisitingInfo	VisitingInfoID	PatientID, VisitorsID
Visitors	VisitorsID	
PatientMedicalRecords	PatMedRecID	PatientID
PatientAppointments	PatAppID	PatientID, EmployeeID
Employee	EmployeeID	
MedicalStaff	EmployeeID	EmployeeID
NonMedicalStaffSchedule	NonMedStaScheID	EmployeeID, FacilityID
PhysicianSchedules	PhyScheID	EmployeeID, FacilityID
NurseSchedules	NurScheID	EmployeeID, FacilityID
SurgeryRoomSchedules	SurRooScheID	EmployeeID, FacilityID
InpatientRoomSchedules	InpRooScheID	EmployeeID, FacilityID
Facility	FacilityID	
EquipmentReservations	EquResID	FacilityID
FacilityDetail	FacilityDetailID	FacilityID
Lab	LabID	
Pharmacy	PharmacyID	
Imaging	ImagingID	



Vertabelo

## II. Database Implementation

Create tables:

CODE:

--PROJECT 2 JINBO WANG

USE UniversityMedicalCenter

GO

--CREATE TABLES

```
CREATE TABLE Billing (
    BillingID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    Visits varchar(50) NULL,
    MedicalBillingCodes varchar(50) NULL,
    Payor varchar(50) NULL,
    PaymentMethod varchar(50) NULL,
);
```

```
CREATE TABLE Costs (
    CostsID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    Medications varchar(50) NULL,
    Procedures varchar(50) NULL,
);
```

```
CREATE TABLE Employee (
    EmployeeID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
```

```

    Address varchar(100) NOT NULL,
    Phone varchar(50) NOT NULL,
    Email varchar(50) NOT NULL,
    Salary varchar(50) NOT NULL,
    Benefits varchar(50) NULL,
    ContractType varchar(50) NULL,
    ContractTerm varchar(50) NULL,
    Reviews varchar(50) NULL,
    Department varchar(50) NULL,
    Office varchar(50) NULL,
    OfficePhone varchar(50) NULL,
    WorkSchedule varchar(50) NULL,
);

CREATE TABLE EquipmentReservations (
    EquResID int NOT NULL PRIMARY KEY IDENTITY,
    FacilityID int NOT NULL,
    EquipmentName varchar(50) NOT NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
);

CREATE TABLE Facility (
    FacilityID int NOT NULL PRIMARY KEY IDENTITY,
    FacilityName varchar(50) NOT NULL,
    Location varchar(100) NOT NULL,
    Hours datetime NULL,
    RoomCapacity varchar(50) NULL,
    Departments varchar(100) NULL,
    CareCapacity varchar(50) NULL,
    MedicalEquipment varchar(50) NULL,
);

CREATE TABLE FacilityDetail (
    FacilityDetailID int NOT NULL PRIMARY KEY IDENTITY,
    FacilityID int NOT NULL,
    LabID int NOT NULL,
    ImagingID int NOT NULL,
    PharmacyID int NOT NULL,
);

CREATE TABLE Imaging (
    ImagingID int NOT NULL PRIMARY KEY IDENTITY,
    ImagingName varchar(50) NOT NULL,

```

```

        Testing varchar(50) NULL,
    );

CREATE TABLE InpatientRoomSchedules (
    InpRooScheID int NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID int NOT NULL,
    FacilityID int NOT NULL,
    InpatientRoomName varchar(50) NOT NULL,
    PatientID int NOT NULL,
    Schedule datetime NULL,
);

CREATE TABLE InsuranceCoverage (
    InsuranceID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    HealthInsuranceCompanyName varchar(50) NOT NULL,
    Phone varchar(50) NOT NULL,
    ContactInfo varchar(50) NULL,
    InsuranceCoverages varchar(50) NOT NULL,
);

CREATE TABLE Lab (
    LabID int NOT NULL PRIMARY KEY IDENTITY,
    LabName varchar(50) NOT NULL,
    Tests varchar(50) NULL,
    Results varchar(50) NULL,
    MandatedReportingtoCountyState varchar(50) NULL,
);

CREATE TABLE MedicalStaff (
    EmployeeID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Type varchar(50) NOT NULL,
    Position varchar(50) NOT NULL,
    Status varchar(50) NOT NULL,
);

CREATE TABLE NonMedicalStaffSchedule (
    NonMedStaScheID int NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID int NOT NULL,
    FacilityID int NOT NULL,
    Schedule datetime NULL,
);

```

```
CREATE TABLE NurseSchedules (
    NurScheID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    EmployeeID int NOT NULL,
    FacilityID int NOT NULL,
    Schedule datetime NULL,
);
```

```
CREATE TABLE Patient (
    PatientID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Address varchar(100) NULL,
    Phone varchar(50) NULL,
    Email varchar(50) NULL,
    HealthInsuranceInfo varchar(50) NULL,
    PrimaryCareDoctor varchar(50) NOT NULL,
);
```

```
CREATE TABLE PatientAppointments (
    PatAppID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    EmployeeID int NOT NULL,
    AppointmentDate datetime NOT NULL,
    Description varchar(50) NULL,
);
```

```
CREATE TABLE PatientMedicalRecords (
    PatMedRecID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    Weight int NULL,
    Height int NULL,
    Vitals varchar(50) NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
    SymptomDiagnosticProcedureCodes int NULL,
    AttendingPhysician varchar(50) NULL,
    ReferralDoctor varchar(50) NULL,
    MedicationHospitalizationDischarge varchar(50) NULL,
);
```

```
CREATE TABLE Pharmacy (
    PharmacyID int NOT NULL PRIMARY KEY IDENTITY,
    PharmacyName varchar(50) NOT NULL,
    Medicines varchar(50) NULL,
```



```

        RentalEquipment varchar(50) NULL,
    );

CREATE TABLE PhysicianSchedules (
    PhyScheID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    EmployeeID int NOT NULL,
    FacilityID int NOT NULL,
    Schedule datetime NULL,
);

CREATE TABLE SurgeryRoomSchedules (
    SurRooScheID int NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID int NOT NULL,
    FacilityID int NOT NULL,
    SurgeryRoomName varchar(50) NOT NULL,
    Schedule datetime NULL,
);

CREATE TABLE VisitingInfo (
    VisitingInfoID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    VisitorsID int NOT NULL,
);

CREATE TABLE Visitors (
    VisitorsID int NOT NULL PRIMARY KEY IDENTITY,
    VisitorsName varchar(50) NOT NULL,
    IDNumber varchar(50) NOT NULL,
    DateOfBirth datetime NULL,
    Picture image NULL,
    PatientBeingVisited varchar(50) NOT NULL,
    EntryTime datetime NULL,
    ExitTime datetime NULL,
);

--FOREIGN KEYS
ALTER TABLE Billing
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE Costs
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE EquipmentReservations

```

```

ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE FacilityDetail
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE FacilityDetail
ADD FOREIGN KEY (ImagingID) REFERENCES Imaging (ImagingID);

ALTER TABLE FacilityDetail
ADD FOREIGN KEY (LabID) REFERENCES Lab (LabID);

ALTER TABLE FacilityDetail
ADD FOREIGN KEY (PharmacyID) REFERENCES Pharmacy (PharmacyID);

ALTER TABLE InpatientRoomSchedules
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE InpatientRoomSchedules
ADD FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID);

ALTER TABLE InsuranceCoverage
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE MedicalStaff
ADD FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID);

ALTER TABLE NurseSchedules
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE NurseSchedules
ADD FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID);

ALTER TABLE PatientAppointments
ADD FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID);

ALTER TABLE PatientAppointments
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE PatientMedicalRecords
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE PhysicianSchedules
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

```

```

ALTER TABLE PhysicianSchedules
ADD FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID);

ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID);

ALTER TABLE NonMedicalStaffSchedule
ADD FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID);

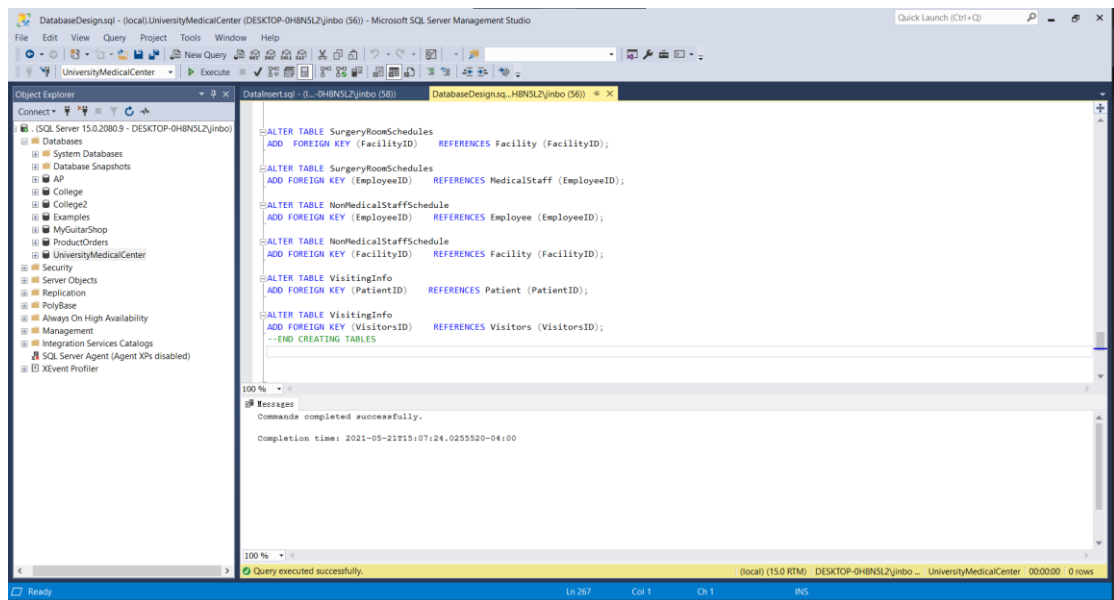
ALTER TABLE NonMedicalStaffSchedule
ADD FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID);

ALTER TABLE VisitingInfo
ADD FOREIGN KEY (PatientID) REFERENCES Patient (PatientID);

ALTER TABLE VisitingInfo
ADD FOREIGN KEY (VisitorsID) REFERENCES Visitors (VisitorsID);

--END CREATING TABLES

```



Data insert:

Code:

```
--PROJECT 2 JINBO WANG
```

```
USE UniversityMedicalCenter
```

```
GO
```

```

--INSERT DATA TO TABLES
--In Frequency of constraints.
SET IDENTITY_INSERT Patient ON
INSERT Patient(PatientID, Name, Address, Phone, Email, HealthInsuranceInfo, PrimaryCareDoctor)
VALUES
(1, 'Paulette Walters', '965 Haverhill Dr, Hamilton, OH 45013', '2095814801',
'yamla@yahoo.com', NULL, 'Dean Hill'),
(2, 'Richard Mcdaniel', '104 S 5th St, Le Claire, IA 52753', '5844635914', 'matsn@gmail.com',
NULL, 'Jake Ellis'),
(3, 'Bert Lee', '11205 Oregon Ave N, Champlin, MN 55316', '3979599966',
'frostman@comcast.net', NULL, 'Stanley Cortez'),
(4, 'Lloyd Austin', '40851 140th St, Groton, SD 57445', '2838493114', 'keutzer@live.com',
NULL, 'Kim Carpenter'),
(5, 'Robin Valdez', '1203 Northwood Dr, Champaign, IL 61821', '8985881698', 'bjoern@msn.com',
NULL, 'Mamie Torres'),
(6, 'Greg Casey', '11118 Zimmerman Rd, Marysville, IN 47141', '1758896071',
'isorashi@sbcglobal.net', NULL, 'Alejandro Pratt'),
(7, 'Ervin Harrison', '202 N 2nd St, Wolcottville, IN 46795', '1177952012',
'pdbaby@optonline.net', NULL, 'Danielle Gilbert'),
(8, 'Eula Carson', '1763 3rd St, Cuyahoga Falls, OH 44221', '6303896245', 'cosimo@att.net',
NULL, 'Tracy Morris'),
(9, 'Christie Moody', '5665 Reindeer Pl, Paso Robles, CA 93446', '8935944099',
'hling@hotmail.com', NULL, 'Raul Stewart'),
(10, 'Felicia Osborne', '28500 County 80 Rd, Orlando, OK 73073', '3755204247',
'adillon@gmail.com', NULL, 'Claude Perkins')
SET IDENTITY_INSERT Patient OFF

SET IDENTITY_INSERT Billing ON
INSERT Billing(BillingID, PatientID, Visits, MedicalBillingCodes, Payor, PaymentMethod)VALUES
(1, 1, NULL, '058155480', NULL, 'Card'),
(2, 2, NULL, 'A58155480', NULL, 'Card'),
(3, 3, NULL, 'E24659070', NULL, 'Card'),
(4, 4, 'Good visits', 'L27021613', NULL, 'Cash'),
(5, 5, 'Good visits', 'P20513244', NULL, 'Card'),
(6, 6, 'Good visits', 'A22638768', NULL, 'Card'),
(7, 7, 'Good visits', 'E22948025', NULL, 'Cash'),
(8, 8, 'Good visits', 'L47581032', NULL, 'Cash'),
(9, 9, NULL, 'P95159496', NULL, 'Card'),
(10, 10, NULL, 'F89141139', NULL, 'Cash')
SET IDENTITY_INSERT Billing OFF

SET IDENTITY_INSERT Costs ON
INSERT Costs(CostsID, PatientID, Medications, Procedures)VALUES
(1,5, 'Acetaminophen', 'Auscultation'),

```

```
(2,6, 'Cyclobenzaprine', 'Medical inspection '),
(3,8, 'Januvia', 'Palpation'),
(4,9, 'Omeprazole', 'Percussion (medicine)'),
(5,10, 'Adderall', 'Vital signs measurement'),
(6,1, 'Cymbalta', 'Biopsy test'),
(7,3, 'Jardiance', 'Blood test'),
(8,2, 'Onpattro', 'Stool test'),
(9,4, 'Amitriptyline', 'Urinalysis'),
(10,7, 'Doxycycline', 'Cardiac stress test')
```

SET IDENTITY\_INSERT Costs OFF

SET IDENTITY\_INSERT Lab ON

INSERT Lab(LabID, LabName, Tests, Results, MandatedReportingtoCountyState)VALUES

```
(1, 'Altman Lab', 'Genetic Testing', 'Positive', NULL),
(2, 'Dean Lab', 'Kidney Tests', 'Positive', 'Reported'),
(3, 'Coppage Lab', 'Hepatitis Testing', 'Negative', 'Reported'),
(4, 'McCamant Lab', 'Thyroid Tests', 'Positive', 'Reported'),
(5, 'Immunization Lab', 'Laboratory Tests', 'Positive', 'Reported'),
(6, 'Elder Lab', 'Prenatal Testing', 'Negative', 'Reported'),
(7, 'DeAngelis Lab', 'Urinalysis', 'Positive', 'Reported'),
(8, 'Vertino Lab', 'Albumin Blood Test', 'Negative', NULL),
(9, 'Waugh Lab', 'Acid-Fast Bacillus (AFB) Tests', 'Positive', 'Reported'),
(10, 'White, Patricia Lab', 'Blood Count Tests', 'Negative', 'Reported')
```

SET IDENTITY\_INSERT Lab OFF

SET IDENTITY\_INSERT Facility ON

INSERT Facility(FacilityID, FacilityName, Location, Hours, RoomCapacity, Departments, CareCapacity, MedicalEquipment) VALUES

```
(1, 'Cliffside Renal Dialysis', 'Flushing', '2005-12-22', '3 rooms 6 beds', 'pediatrics', 6, 3),
(2, 'Castle Hill Medical Center of New York Inc.', 'Bronx', '2007-01-01', '3 rooms 3 beds', 'surgery', 3, 3),
(3, 'Primary Care Associates', 'Endicott', '1998-02-01', '4 rooms 8 beds', 'Hospital Extension Clinic', 10, 8),
(4, 'Lutheran Center at Poughkeepsie, Inc', 'Poughkeepsie', '1998-08-24', '3 rooms 6 beds', 'pediatrics', 6, 3),
(5, 'Meadowbrook Endoscopy Center', 'Westbury', '2008-10-15', '4 rooms 4 beds', 'surgery', 4, 4),
(6, 'New York Endoscopy Center', 'White Plains', '2009-10-05', '10 rooms 20 beds', 'Diagnostic and Treatment Center', 20, 20),
(7, 'UHCC - Madison Irving', 'Syracuse', '2010-06-14', '5 rooms 8 beds', 'Certified Home Health Agency', 20, 15),
(8, 'Warner Place', 'Jamestown', '2010-07-28', '5 rooms 8 beds', 'Adult Day Health Care
```

```

Program - Offsite', 10, 8),
(9, 'Griffiss Surgery Center', 'Rome', '2012-01-03', '5 rooms 5 beds', 'surgery', 5, 5),
(10, 'H.K. Freedman Renal Center', 'Plattsburgh', '2013-12-05', '4 rooms 8 beds', 'Diagnostic
and Treatment Center', 10, 8)
SET IDENTITY_INSERT Facility OFF

```

```

SET IDENTITY_INSERT EquipmentReservations ON
INSERT EquipmentReservations(EquResID, FacilityID, EquipmentName, CheckInTime,
CheckOutTime)VALUES
(1, 1, 'EKG and ECGs', 2021-02-15, 2021-02-17),
(2, 3, 'Autoclaves', 2021-03-01, 2021-03-15),
(3, 5, 'Patient Monitors', 2021-03-01, NULL),
(4, 4, 'C-Arms', 2021-01-15, 2021-01-30),
(5, 9, 'Sterilizers', 2021-05-03, 2021-05-20),
(6, 8, 'Surgical Tables', 2021-03-10, NULL),
(7, 10, 'Blanket and Fluid Warmers', 2021-03-10, 2021-03-12),
(8, 2, 'Anesthesia Machines', 2021-02-01, 2021-02-10),
(9, 7, 'Defibrillators', 2021-04-03, NULL),
(10, 6, 'Surgical Lights', 2021-02-07, 2021-02-10)

```

```

SET IDENTITY_INSERT EquipmentReservations OFF

```

```

SET IDENTITY_INSERT Imaging ON
INSERT Imaging(ImagingID, ImagingName, Testing)VALUES
(1, 'Direct', 'MRI'),
(2, 'Indirect', 'CAT'),
(3, 'Direct', 'MRI'),
(4, 'Direct', 'MRI'),
(5, 'Direct', 'MRI'),
(6, 'Direct', 'CAT'),
(7, 'Indirect', 'MRI'),
(8, 'Indirect', 'MRI'),
(9, 'Indirect', 'MRI'),
(10, 'Indirect', 'CAT')

```

```

SET IDENTITY_INSERT Imaging OFF

```

```

SET IDENTITY_INSERT Pharmacy ON
INSERT Pharmacy(PharmacyID, PharmacyName, Medicines, RentalEquipment)VALUES
(1, 'Boone Drug', 'Allopurinol, Alendronate, Alemtuzumab', 'Defibrillators'),
(2, 'Bartell Drugs', 'Ifosfamide, Imatinib, Imipenem', 'Anesthesia Machines'),
(3, 'CVS Pharmacy', 'Dacarbazine, Dactinomycin, Dapsone', 'Patient Monitors'),
(4, 'Discount Drug Mart', 'L-glutamine, abetalol, Lamivudine', 'EKG/ECG Machines','Heparin Lock
Flush '),
(5, 'Hartig Drug', 'Heparin Lock Flush', 'Sterilizers'),
(6, 'London Drugs', 'Nelarabine, Nelfinavir, Neuromuscular', 'Surgical Tables'),

```

```
(7, 'Family Pharmacy', 'Omeprazole, Ondansetron, Oxycodone', 'Blanket and Fluid Warmers'),
(8, 'Pharmasave', 'Valproic acid, Vancomycin, Vinblastine', 'Electrosurgical Units'),
(9, 'Good Neighbor', 'Efavirenz, Efavirenz / emtricitabine', 'Surgical Tables'),
(10, 'Lawtons', 'Baclofen, Bleomycin, Bortezomib', 'EKG/ECG Machines')
SET IDENTITY_INSERT Pharmacy OFF
```

```
SET IDENTITY_INSERT FacilityDetail ON
```

```
INSERT FacilityDetail(FacilityDetailID, FacilityID, LabID, ImagingID, PharmacyID)VALUES
(1, 1, 1, 4, 7),
(2, 2, 3, 5, 8),
(3, 3, 5, 1, 3),
(4, 4, 10, 8, 1),
(5, 5, 4, 9, 2),
(6, 6, 7, 6, 5),
(7, 7, 6, 3, 4),
(8, 8, 2, 2, 6),
(9, 9, 9, 7, 9),
(10, 10, 8, 10, 10)
```

```
SET IDENTITY_INSERT FacilityDetail OFF
```

```
SET IDENTITY_INSERT Employee ON
```

```
INSERT Employee(EmployeeID, Name, Address, Phone, Email, Salary, Benefits, ContractType,
ContractTerm, Reviews, Department, Office, OfficePhone, WorkSchedule) VALUES
(1, 'Finnian Moyer', '15 Alton Street Brooklyn, NY
11207', '6558307415', 'FinnianMoyer@gmail.com', '50534', 'bonus', 'fixed-
price', NULL, NULL, 'medical', NULL, '8684091023', '9-17'),
(2, 'Kiyan Worthington', '501 South Peachtree Lane Bronx, NY 10463',
'8845227792', 'KiyanWorthington@gmail.com', '37453', NULL, 'fixed-
price', NULL, NULL, 'administration', NULL, '8376350088', '9-17'),
(3, 'Minnie Glenn', '785 Carriage Dr. West Babylon, NY
11704', '0170735302', 'MinnieGlenn@gmail.com', '63765', NULL, 'fixed-
price', NULL, NULL, 'logistics', NULL, '0463209805', '9-17'),
(4, 'Inaaya Dyer', '81 Old Dunbar Lane Jackson Heights, NY 11372',
'6310627418', 'InaayaDyer@gmail.com', '78345', 'bonus', 'fixed-
price', NULL, NULL, 'administration', NULL, '7200016468', '8-16'),
(5, 'Tariq Myers', '2 North Berkshire Avenue Yonkers, NY 10701',
'5295374578', 'TariqMyers@gmail.com', '68755', 'bonus', 'cost-
reimbursement', NULL, NULL, 'administration', NULL, '9611106482', '8-16'),
(6, 'Vihaan Hulme', '24518 Melyssa Meadow Apt. 322 West Adalberto, IN 91173',
'6069878004', 'pfeffer.cydne@yahoo.com', '54464', 'bonus', 'fixed-
price', NULL, NULL, 'administration', NULL, '1628118873', '9-17'),
(7, 'Barry Carpenter', '4549 Wilton Valley Sauerberg, PA
23957', '6217172459', 'schuppe.braeden@yahoo.com', '45645', 'bonus', 'fixed-
price', NULL, NULL, 'administration', NULL, '0772970610', '9-17'),
```

(8, 'Jaeden Noel', '15722 Wilkinson Isle East Dellastad, WA 49728',  
 '5041957090', 'rae95@hotmail.com', '55645', 'bonus', 'fixed-  
 price', NULL, NULL, 'administration', NULL, '0422424858', '9-17'),

(9, 'Darrell Naylor', '5334 VonRueden Lakes Suite 683 Goodwinville, NE 81010',  
 '7909854128', 'alakin@gmail.com', '45645', 'bonus', 'fixed-  
 price', NULL, NULL, 'administration', NULL, NULL, '9-17'),

(10, 'Niyah Kemp', '110 Skyla Estate Port Mittie, SC 49513',  
 '6740838458', 'crooks.alaina@yahoo.com', '45645', 'bonus', 'fixed-  
 price', NULL, 'logistics', 'administration', NULL, '0814764446', '9-17'),

(11, 'Rares Hopper', '693 Estelle Cliffs Suite 704 Michelemouth, SC 53933',  
 '4045537794', 'hannah97@spinka.net', '67657', 'bonus', 'fixed-  
 price', NULL, NULL, 'medical', NULL, '1805733194', '9-17'),

(12, 'Hughie Whittington', '401 Odessa Turnpike Port Budtown, AR 98911-3045',  
 '4671739881', 'zmaggio@wuckert.com', '65454', 'bonus', 'fixed-  
 price', NULL, NULL, 'medical', NULL, '7915637384', '9-17'),

(13, 'Teddie Curran', '671 Gottlieb Row Apt. 177 Schadenberg, AL 12452-  
 0485', '7967015831', 'eli21@schmeler.net', '87685', 'bonus', 'fixed-  
 price', NULL, NULL, 'medical', NULL, '3969268819', '9-17'),

(14, 'Jaheim Friedman', '892 Matilda Plains Apt. 923 New Arno, MD 53607',  
 '9131847291', 'xledner@huel.com', '45656', 'bonus', 'fixed-  
 price', NULL, NULL, 'medical', NULL, '2831681366', '8-16'),

(15, 'Declan Shah', '372 Cristopher Roads Lake Santa, AZ 76817',  
 '8292203050', 'alberta.casper@lindgren.com', '74536', 'bonus', 'fixed-  
 price', NULL, NULL, 'medical', NULL, '4081907698', '8-16'),

(16, 'Carly Collier', '331 Leopold Prairie Apt. 792 West Tracemouth, MA 52000-1820',  
 '5063202635', 'delores.christiansen@gmail.com', '46345', NULL, 'fixed-  
 price', NULL, NULL, 'medical', NULL, '0039484081', '8-16'),

(17, 'Huxley Bailey', '37033 Izaiah Ridges Gianniport, DE  
 42915', '89644400286', 'tillman.eloise@morissette.biz', '76574', NULL, 'fixed-  
 price', NULL, NULL, 'medical', NULL, '4828147894', '8-16'),

(18, 'Harlow Emerson', '823 Alberta Fork Suite 359 Lake Dorothea, MA 48558',  
 '5192833596', 'gunnar55@mraz.com', '78567', NULL, 'fixed-  
 price', NULL, NULL, 'logistics', NULL, '8050950930', '8-16'),

(19, 'Teddy Grey', '255 Carter Junction Suite 221 Devynbury, SC 86155-1203',  
 '9828774679', 'hills.andreanne@yahoo.com', '55724', NULL, 'fixed-  
 price', NULL, NULL, 'medical', NULL, NULL, '8-16'),

(20, 'Anne Hubbard', '4733 Treutel Vista Apt. 708 West Matteoberg, LA 88682',  
 '6027894995', 'schmitt.helena@langosh.com', '77644', NULL, 'fixed-  
 price', NULL, NULL, 'logistics', NULL, NULL, '8-16'),

(21, 'Bethanie Delarosa', '3670 Brekke Fork Suite 375 Fadelburgh, ND 39924-1567',  
 '6053408977', 'king.kellen@nolan.com', '73454', NULL, 'fixed-  
 price', NULL, NULL, 'medical', NULL, '9257686337', '8-16'),

(22, 'Imani Currie', '8317 Idella Underpass Lake Mack, MO  
 83033', '4791179552', 'leuschke.anabel@gmail.com', '76544', 'bonus', 'fixed-



```

price',NULL,NULL,'logistics',NULL,NULL,'8-16'),
(23,'Nabiha Olsen','72064 Jenkins Mission Suite 612 Naderhaven, NC 74759-2212',
'0869231800','maeve52@hotmail.com','43574','bonus','fixed-
price',NULL,NULL,'logistics',NULL,'7110229650','8-16'),
(24,'Ammarah Fernandez','588 Jast Keys Suite 388 Stammland, MO 81937-
4219','6661434310','beahan.maxwell@prosacco.com','45745','bonus','fixed-
price',NULL,NULL,'medical',NULL,'7379103681','8-16'),
(25,'Muna Ward','3835 Herzog Isle Suite 201 Rutheland, IN 55759-6656',
'7451148432','dauer@hotmail.com','75644',NULL,'fixed-
price',NULL,NULL,'medical',NULL,'4816091219','8-16'),
(26,'Fleur Sampson','432 Hahn Road Apt. 266 Wardstad, MN 88476-
3499','2368578461','xdickinson@yahoo.com','76545',NULL,'fixed-
price',NULL,NULL,'logistics',NULL,NULL,'9-17'),
(27,'Bobbie Patterson','912 Hand Flats Elodyside, GA 04646',
'0842572167','qgottlieb@parisian.com','57353',NULL,'fixed-
price',NULL,NULL,'logistics',NULL,'3412970749','8-16'),
(28,'Serena Zimmerman','2435 Hamill Port South Gerry, MA 28100-7145',
'2838524220','yterry@yahoo.com','65754','bonus','fixed-
price',NULL,NULL,'administration',NULL,'2737894990','8-16'),
(29,'Myles Pacheco','694 Quigley Path Jenkinsport, VA 72640-7449',
'8895636740','abagail.kshlerin@hotmail.com','65734','bonus','fixed-
price',NULL,NULL,'administration',NULL,'9510504260','9-17'),
(30,'Herbert Burt','767 Lucienne Land Apt. 646 West Filomenaview, IA 31760-6929',
'7281804046','wolff.nora@connelly.com','34534','bonus','fixed-
price',NULL,NULL,'medical',NULL,NULL,'8-16')

```

SET IDENTITY\_INSERT Employee OFF

SET IDENTITY\_INSERT MedicalStaff ON

INSERT MedicalStaff(EmployeeID, Name, Type, Position, Status) VALUES

```

(1,'Finnian Moyer','Physician','attending','active'),
(2,'Kiyan Worthington','Physician','attending','active'),
(3,'Minnie Glenn','Physician','PCP','active'),
(4,'Inaaya Dyer','Physician','PCP','active'),
(5,'Tariq Myers','Physician','PCP','retired'),
(6,'Vihaan Hulme','Physician','PCP','active'),
(7,'Barry Carpenter','Physician','PCP','active'),
(8,'Jaeden Noel','Physician','attending','active'),
(9,'Darrell Naylor','Physician','attending','leave'),
(10,'Niyah Kemp','Physician','attending','leave'),
(11,'Rares Hopper','Nurse','visiting','leave'),
(12,'Hughie Whittington','Nurse','PCP','active'),
(13,'Teddie Curran','Nurse','PCP','active'),
(14,'Jaheim Friedman','Nurse','PCP','retired'),
(15,'Declan Shah','Nurse','PCP','active'),

```

```

(16, 'Carly Collier', 'Nurse', 'PCP', 'retired'),
(17, 'Huxley Bailey', 'Nurse', 'PCP', 'active'),
(18, 'Harlow Emerson', 'Nurse', 'attending', 'active'),
(19, 'Teddy Grey', 'Nurse', 'attending', 'leave'),
(20, 'Anne Hubbard', 'Nurse', 'visiting', 'active')
SET IDENTITY_INSERT MedicalStaff OFF

```

```

SET IDENTITY_INSERT InpatientRoomSchedules ON

```

```

INSERT InpatientRoomSchedules(InpRooScheID, FacilityID, EmployeeID, InpatientRoomName,
PatientID, Schedule)VALUES
(1, 1, 11, '1001', 1, '2021-05-01'),
(2, 2, 12, '1002', 2, '2021-05-02'),
(3, 3, 13, '1003', 3, '2021-05-03'),
(4, 4, 14, '1004', 4, '2021-05-04'),
(5, 5, 15, '1005', 5, '2021-05-05'),
(6, 6, 16, '1006', 6, '2021-05-06'),
(7, 7, 17, '1007', 7, '2021-05-07'),
(8, 8, 18, '1008', 8, '2021-05-08'),
(9, 9, 19, '1009', 9, '2021-05-09'),
(10, 10, 10, '1100', 10, '2021-05-10')
SET IDENTITY_INSERT InpatientRoomSchedules OFF

```

```

SET IDENTITY_INSERT InsuranceCoverage ON

```

```

INSERT InsuranceCoverage(InsuranceID, PatientID, HealthInsuranceCompanyName, Phone,
ContactInfo, InsuranceCoverages) VALUES
(1, 1, 'UnitedHealthcare Group', '6442725896', NULL, 'Full Coverage'),
(2, 2, 'Anthem', '3527353380', NULL, 'Full Coverage'),
(3, 3, 'Cigna', '8840713595', NULL, 'Half Coverage'),
(4, 4, 'UnitedHealthcare Group', '6442725896', NULL, 'Full Coverage'),
(5, 5, 'Anthem', '3527353380', NULL, 'Full Coverage'),
(6, 6, 'Anthem', '3527353380', NULL, 'Half Coverage'),
(7, 7, 'UnitedHealthcare Group', '6442725896', NULL, 'Half Coverage'),
(8, 8, 'Humana', '4831732029', NULL, 'Full Coverage'),
(9, 9, 'Aetna', '6171301124', NULL, 'Half Coverage'),
(10, 10, 'Cigna', '8840713595', NULL, 'Full Coverage')
SET IDENTITY_INSERT InsuranceCoverage OFF

```

```

SET IDENTITY_INSERT NonMedicalStaffSchedule ON

```

```

INSERT NonMedicalStaffSchedule(NonMedStaScheID, EmployeeID, FacilityID, Schedule)VALUES
(1, 25, 5, '2021-05-01'),
(2, 24, 4, '2021-05-02'),
(3, 23, 3, '2021-05-03'),
(4, 22, 2, '2021-05-04'),
(5, 21, 1, '2021-05-05'),

```

```

(6, 26, 6, '2021-05-06'),
(7, 27, 7, '2021-05-07'),
(8, 28, 8, '2021-05-08'),
(9, 29, 9, '2021-05-09'),
(10, 30, 10, '2021-05-10')
SET IDENTITY_INSERT NonMedicalStaffSchedule OFF

SET IDENTITY_INSERT NurseSchedules ON
INSERT NurseSchedules(NurScheID, Name, EmployeeID, FacilityID, Schedule) VALUES
(1, 'Rares Hopper', 11, 1, '2019-07-12'),
(2, 'Hughie Whittington', 12, 2, '2020-03-23'),
(3, 'Teddie Curran', 13, 3, '2020-07-01'),
(4, 'Jaheim Friedman', 14, 4, '2020-11-01'),
(5, 'Declan Shah', 15, 5, '2019-09-15'),
(6, 'Carly Collier', 16, 6, '2019-12-17'),
(7, 'Huxley Bailey', 17, 7, '2020-06-21'),
(8, 'Harlow Emerson', 18, 8, '2020-07-27'),
(9, 'Teddy Grey', 19, 9, '2019-05-14'),
(10, 'Anne Hubbard', 20, 10, '2020-07-12')
SET IDENTITY_INSERT NurseSchedules OFF

SET IDENTITY_INSERT PatientAppointments ON
INSERT PatientAppointments(PatAppID, PatientID, EmployeeID, AppointmentDate, Description) VALUES
(1, 1, 1, '2020-09-13', NULL),
(2, 2, 5, '2020-11-18', NULL),
(3, 3, 5, '2020-12-02', NULL),
(4, 4, 3, '2021-03-22', NULL),
(5, 5, 6, '2019-02-12', NULL),
(6, 6, 23, '2015-07-12', NULL),
(7, 7, 1, '2021-01-02', NULL),
(8, 8, 3, '2020-02-19', NULL),
(9, 9, 6, '2006-05-17', NULL),
(10, 10, 7, '2005-09-13', NULL)
SET IDENTITY_INSERT PatientAppointments OFF

SET IDENTITY_INSERT PatientMedicalRecords ON
INSERT PatientMedicalRecords(PatMedRecID, PatientID, Weight, Height, Vitals, CheckInTime,
CheckOutTime, SymptomDiagnosticProcedureCodes, AttendingPhysician, ReferralDoctor,
MedicationHospitalizationDischarge) VALUES
(1, 1, 71, 171, NULL, '2021-05-02 01:20:03', '2021-05-02 13:31:44', 1, 'Ismael Webb', 'Ismael Webb',
NULL),
(2, 2, 72, 172, NULL, '2021-05-03 00:12:04', '2021-05-03 06:22:41', 2, 'Virgil Jefferson', 'Virgil
Jefferson', NULL),

```

```

(3,3,73,173, NULL, '2021-05-03 08:23:50', '2021-05-03 12:48:20',3,'Chad Bryant','Chad Bryant',
NULL),
(4,4,74,174, NULL, '2021-05-03 23:47:54', '2021-05-04 00:00:01',4,'Opal Benson','Opal Benson',
NULL),
(5,5,75,175, NULL, '2021-05-04 07:36:08', '2021-05-06 22:28:24',5,'Jodi Bailey','Jodi Bailey',
NULL),
(6,6,76,176, NULL, '2021-05-08 01:49:53', '2021-05-08 08:53:42',6,'Pamela Chambers','Pamela
Chambers', NULL),
(7,7,77,177, NULL, '2021-05-10 13:42:06', '2021-05-13 15:33:00',7,'Lora Beck','Lora Beck',
NULL),
(8,8,78,178, NULL, '2021-05-14 03:08:23', '2021-05-16 05:04:33',8,'Nichole Klein','Nichole
Klein', NULL),
(9,9,79,179, NULL, '2021-05-17 02:34:05', '2021-05-17 11:42:52',9,'Donna Hardy','Donna Hardy',
NULL),
(10,10,80,180, NULL, '2021-05-18 01:35:16', '2021-05-18 09:31:11',10,'Sadie Martinez','Sadie
Martinez', NULL)
SET IDENTITY_INSERT PatientMedicalRecords OFF

```

```

SET IDENTITY_INSERT PhysicianSchedules ON
INSERT PhysicianSchedules (PhyScheID,Name, EmployeeID, FacilityID, Schedule) VALUES
(1,'Finnian Moyer',1,1,'2021-05-21'),
(2,'Kiyan Worthington',2,2,'2021-04-21'),
(3,'Minnie Glenn',3,3,'2021-05-19'),
(4,'Inaaya Dyer',4,7,'2021-05-12'),
(5,'Tariq Myers',5,4,' 2021-05-10'),
(6,'Vihaan Hulme',6,10,'2021-05-19'),
(7,'Barry Carpenter',7,5,'2021-05-21'),
(8,'Jaeden Noel',8,9,'2021-05-07'),
(9,'Darrell Naylor',9,6,'2021-05-08'),
(10,'Niyah Kemp',10,8,'2021-05-19')
SET IDENTITY_INSERT PhysicianSchedules OFF

```

```

SET IDENTITY_INSERT SurgeryRoomSchedules ON
INSERT SurgeryRoomSchedules (SurRooScheID, EmployeeID, FacilityID, SurgeryRoomName,
Schedule)VALUES
(1, 6, 1, 'firstclass1', '2021-05-01'),
(2, 1, 2, 'firstclass2', '2021-05-02'),
(3, 3, 3, 'secondclass1', '2021-05-01'),
(4, 5, 4, 'firstclass3', '2021-05-03'),
(5, 4, 5, 'secondclass2', '2021-05-01'),
(6, 9, 6, 'secondclass3', '2021-05-02'),
(7, 8, 7, 'firstclass4', '2021-05-07'),
(8, 7, 8, 'firstclass5', '2021-05-10'),

```

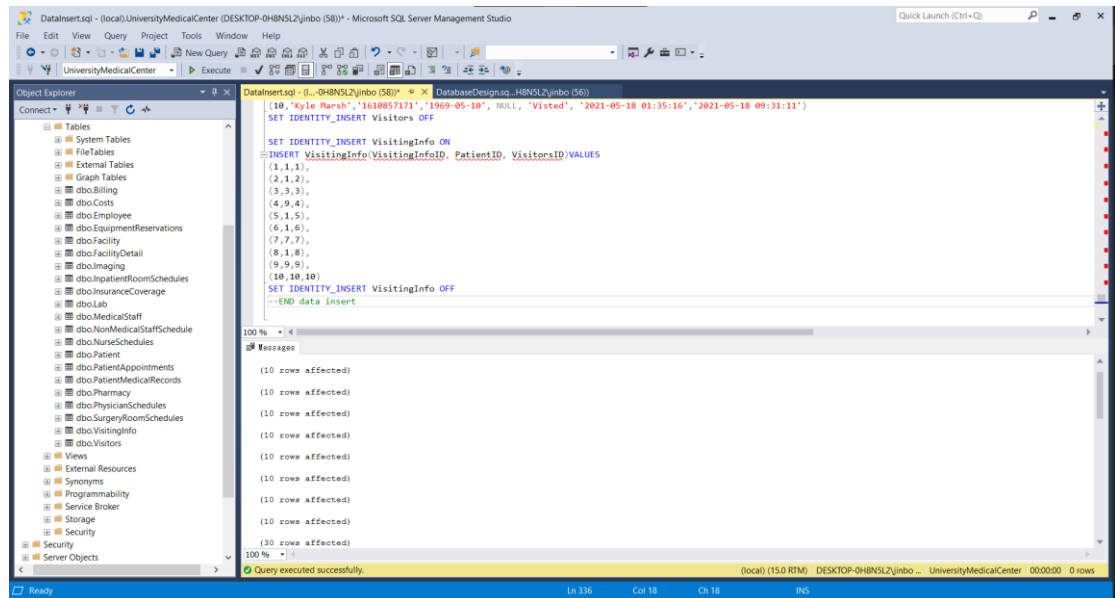
```

(9, 2, 9, 'firstclass6', '2021-05-15'),
(10, 10, 10, 'firstclass7', '2021-05-05')
SET IDENTITY_INSERT SurgeryRoomSchedules OFF

SET IDENTITY_INSERT Visitors ON
INSERT Visitors(VisitorsID, VisitorsName, IDNumber, DateOfBirth, Picture, PatientBeingVisited,
EntryTime, ExitTime) Values
(1,'Mario Baker','2374739417','1989-09-10', NULL, 'Visted', '2021-05-02 01:20:03','2021-05-02
13:31:44'),
(2,'Jennie Pierce','5567549402','1976-08-25', NULL, 'Visted', '2021-05-03 00:12:04', '2021-05-
03 06:22:41'),
(3,'Adam Rodriquez','9971788390','1979-10-26', NULL, 'Visted', '2021-05-03 08:23:50', '2021-
05-03 12:48:20'),
(4,'Angel Ruiz','6065530153','1999-06-22', NULL, 'Visted', '2021-05-03 23:47:54', '2021-05-04
00:00:01'),
(5,'Lyle Summers','3429094491','1997-01-09', NULL, 'Visted', '2021-05-04 07:36:08','2021-05-06
22:28:24'),
(6,'Rose Jennings','6071126183','1963-08-03', NULL, 'Visted', '2021-05-08 01:49:53','2021-05-
08 08:53:42'),
(7,'Valerie Gibson','0535989612','1980-09-25', NULL, 'Visted', '2021-05-10 13:42:06','2021-05-
13 15:33:00'),
(8,'Neil Grant','1996830502','1985-12-25', NULL, 'Visted', '2021-05-14 03:08:23','2021-05-16
05:04:33'),
(9,'Andrew Rodriguez','3460661408','1993-03-21', NULL, 'Visted', '2021-05-17 02:34:05','2021-
05-17 11:42:52'),
(10,'Kyle Marsh','1610857171','1969-05-10', NULL, 'Visted', '2021-05-18 01:35:16','2021-05-18
09:31:11')
SET IDENTITY_INSERT Visitors OFF

SET IDENTITY_INSERT VisitingInfo ON
INSERT VisitingInfo(VisitingInfoID, PatientID, VisitorsID)VALUES
(1,1,1),
(2,1,2),
(3,3,3),
(4,9,4),
(5,1,5),
(6,1,6),
(7,7,7),
(8,1,8),
(9,9,9),
(10,10,10)
SET IDENTITY_INSERT VisitingInfo OFF
--END data insert

```



I use some basic statements to extract some tables to show my database creation.

Code:

```

USE UniversityMedicalCenter
GO

```

```

SELECT *
FROM Employee

```

```

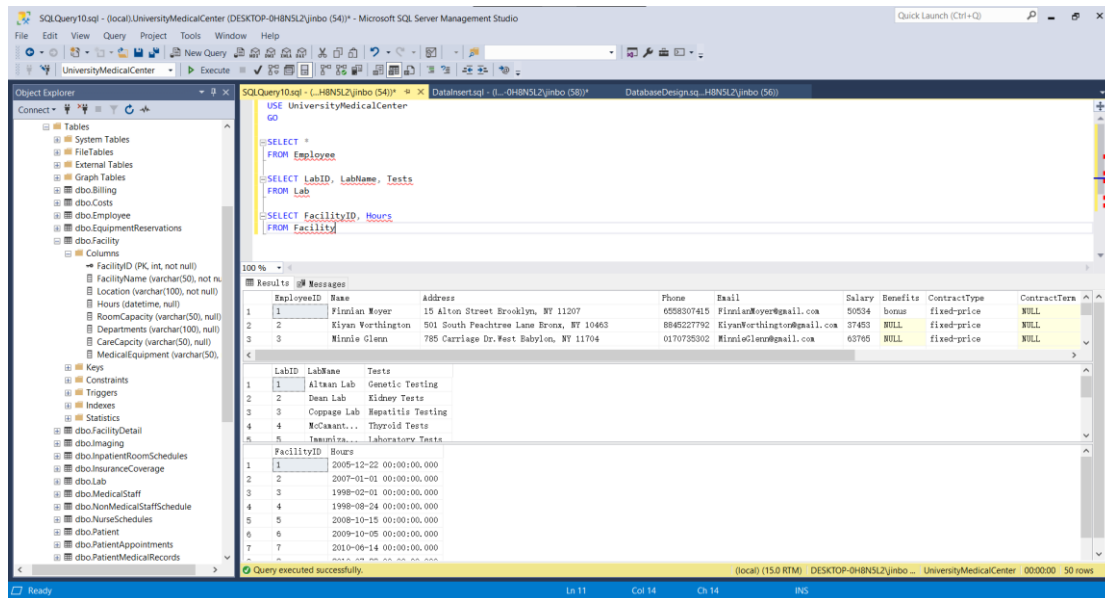
SELECT LabID, LabName, Tests
FROM Lab

```

```

SELECT FacilityID, Hours
FROM Facility

```



### III. Database Testing

In this part, several views, stored procedures, user defined functions, triggers, transactions and scripts to verify the database design.

#### 1. Views

- 1) In this view, I will select patient and find patient's billing in certain condition. It will return the patientID, PatientName, MedicalBillingCodes, Payor.

--VIEW 1

USE UniversityMedicalCenter

GO

CREATE VIEW PatientBilling\_V

AS

SELECT dbo.Patient.PatientID, dbo.Patient.Name, MedicalBillingCodes AS

MBC , Payor

FROM Patient JOIN Billing

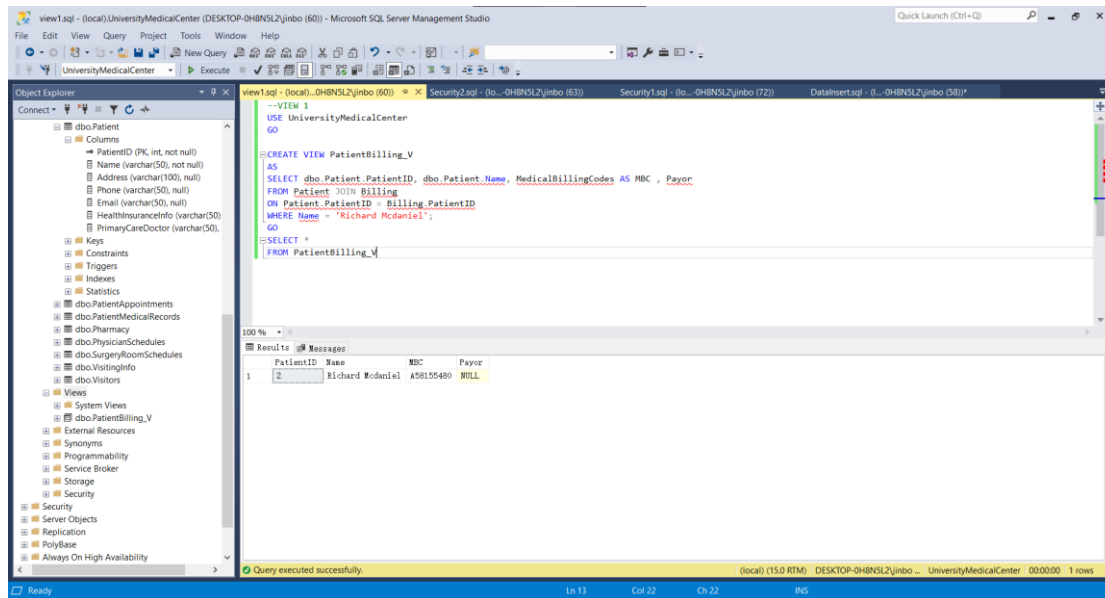
ON Patient.PatientID = Billing.PatientID

WHERE Name = 'Richard Mcdaniel';

GO

SELECT \*

FROM PatientBilling\_V



- 2) In this view, it use select statement and returns which the date after 2009-01-01. It will return Patient Appointments ID, PatientID, AppointmentDate, EmployeeID, and the Employee name.

--VIEW 2

USE UniversityMedicalCenter

GO

CREATE VIEW PatientAppointment\_V

AS

SELECT PatAppID, dbo.PatientAppointments.PatientID, AppointmentDate,  
dbo.PatientAppointments.EmployeeID, dbo.Employee.Name

FROM PatientAppointments JOIN Employee

ON PatientAppointments.EmployeeID = Employee.EmployeeID

WHERE AppointmentDate > '2009-01-01';

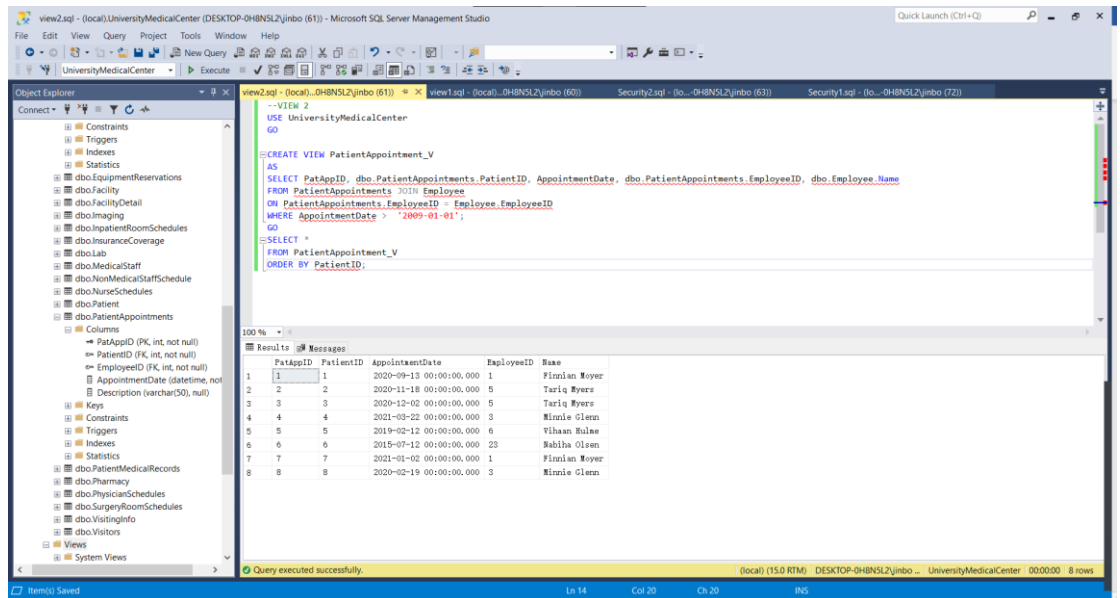
GO

SELECT \*

FROM PatientAppointment\_V

ORDER BY PatientID;





- 3) In this view, it returns PatientMedicalInfo, such as PatientID, Name, Weight, and Height. The search condition is Patient Name with letter 'E'.

```
--VIEW 3
```

```
USE UniversityMedicalCenter
```

```
GO
```

```
CREATE VIEW PatientMedicalInfo_V
```

```
AS
```

```
SELECT dbo.Patient.PatientID, Name, Weight, Height
```

```
FROM Patient JOIN PatientMedicalRecords
```

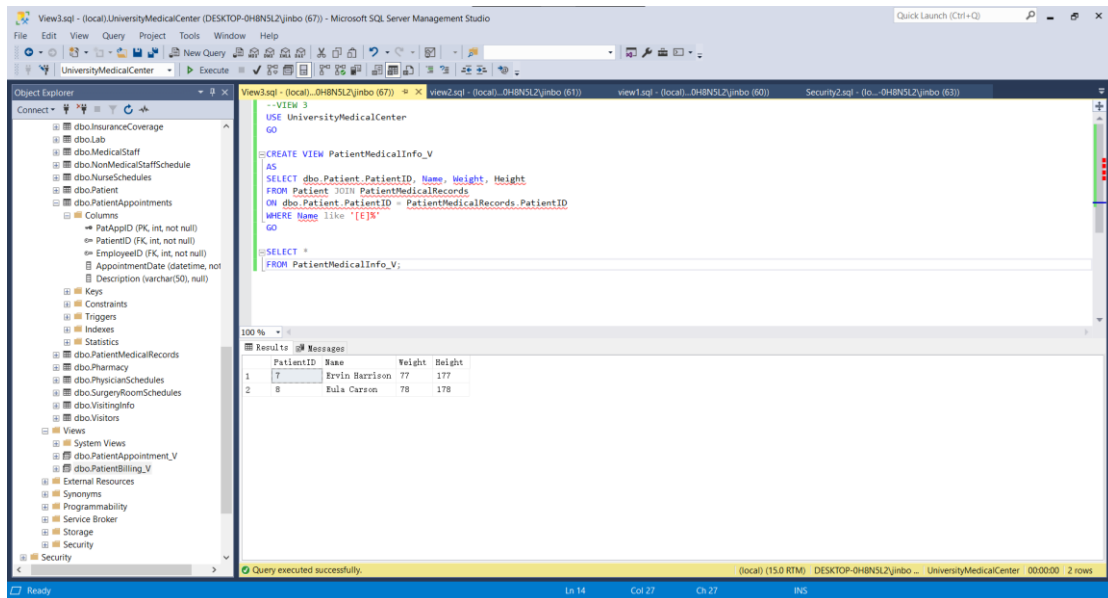
```
ON dbo.Patient.PatientID = PatientMedicalRecords.PatientID
```

```
WHERE Name like '[E]%'
```

```
GO
```

```
SELECT *
```

```
FROM PatientMedicalInfo_V;
```



- 4) In this view, the select statement returns PhyScheID equals 7. It returns Physician Schedule information, such as PhyScheID, EmployeeID, PhysicianInform, and status.

--VIEW 4

USE UniversityMedicalCenter

GO

CREATE VIEW PhysicianSchedule\_V

AS

SELECT PhyScheID, MedicalStaff.EmployeeID, CONCAT(MedicalStaff.Name, ',',  
Position, ',', Schedule) AS PhysicianInform, Status

FROM MedicalStaff JOIN PhysicianSchedules

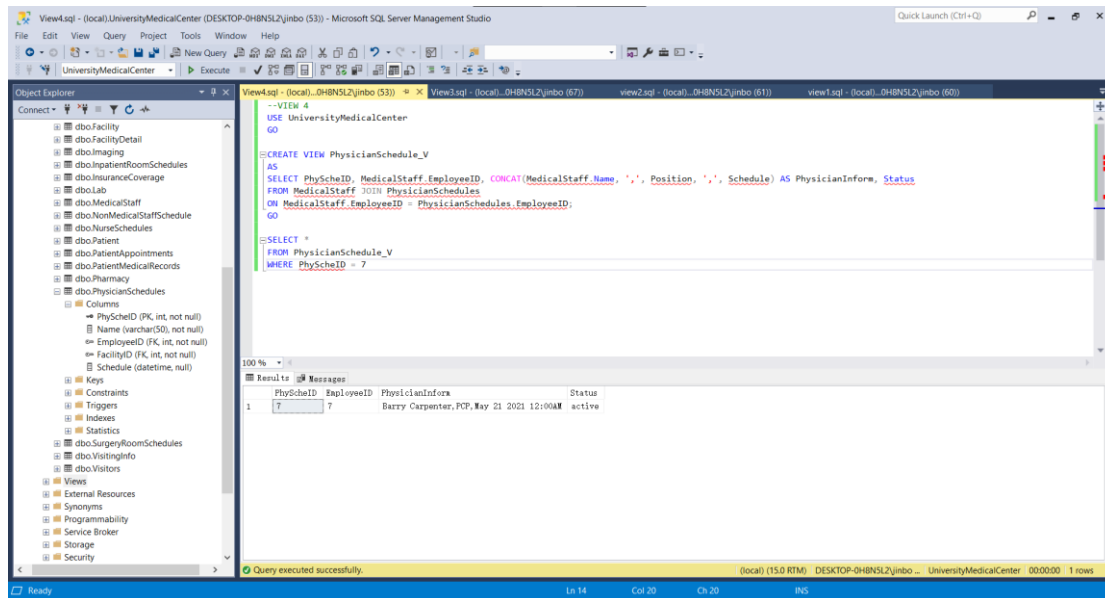
ON MedicalStaff.EmployeeID = PhysicianSchedules.EmployeeID;

GO

SELECT \*

FROM PhysicianSchedule\_V

WHERE PhyScheID = 7



## 2. Stored procedures

- 1) This stored procedure used to find a facility information. First it drops procedure if previous procedure exists. And then create a procedure.

--Stored Procedure 1

```
USE UniversityMedicalCenter
```

```
GO
```

```
IF OBJECT_ID('spFacilityInfo') IS NOT NULL
```

```
DROP PROC spFacilityInfo;
```

```
GO
```

```
CREATE PROC spFacilityInfo
```

```
@FacilityName VARCHAR(50)
```

```
AS
```

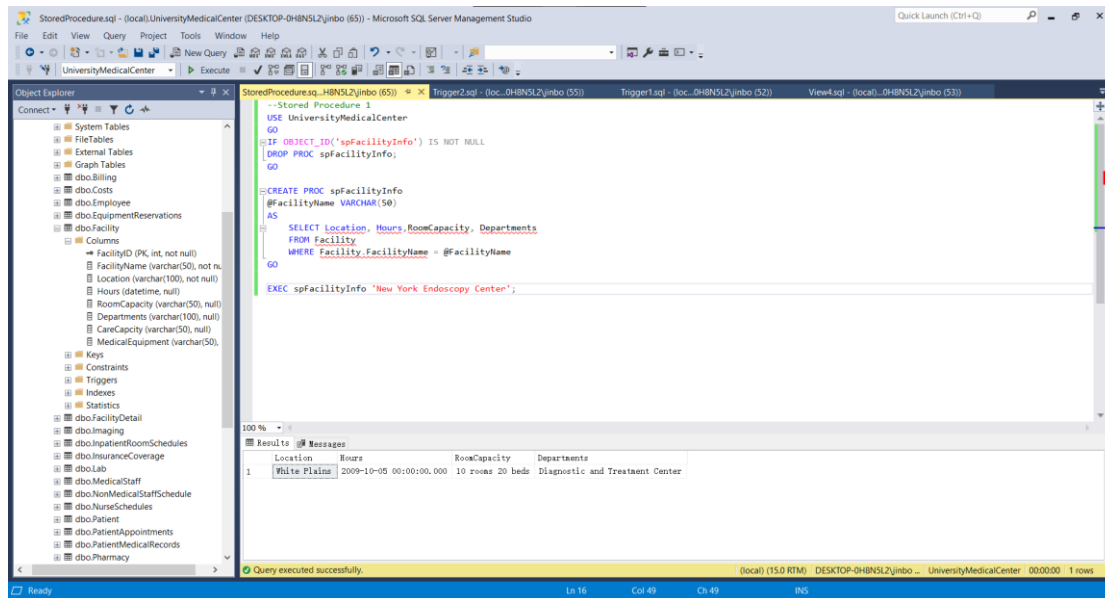
```
SELECT Location, Hours, RoomCapacity, Departments
```

```
FROM Facility
```

```
WHERE Facility.FacilityName = @FacilityName
```

```
GO
```

```
EXEC spFacilityInfo 'New York Endoscopy Center';
```



- 2) This stored procedure used to find patient insurance information. It will return health insurance company name and insurance coverages.

--StoredProcedure 2

USE UniversityMedicalCenter

GO

IF OBJECT\_ID('spPatientInsuInfo') IS NOT NULL

DROP PROC spPatientInsuInfo; /\*Drop procedure if previous procedure exists\*/

GO

/\*Create a procedure to find a patient's visitors information\*/

CREATE PROC spPatientInsuInfo

@name VARCHAR(50)

AS

SELECT HealthInsuranceCompanyName AS HICN, InsuranceCoverages AS IC

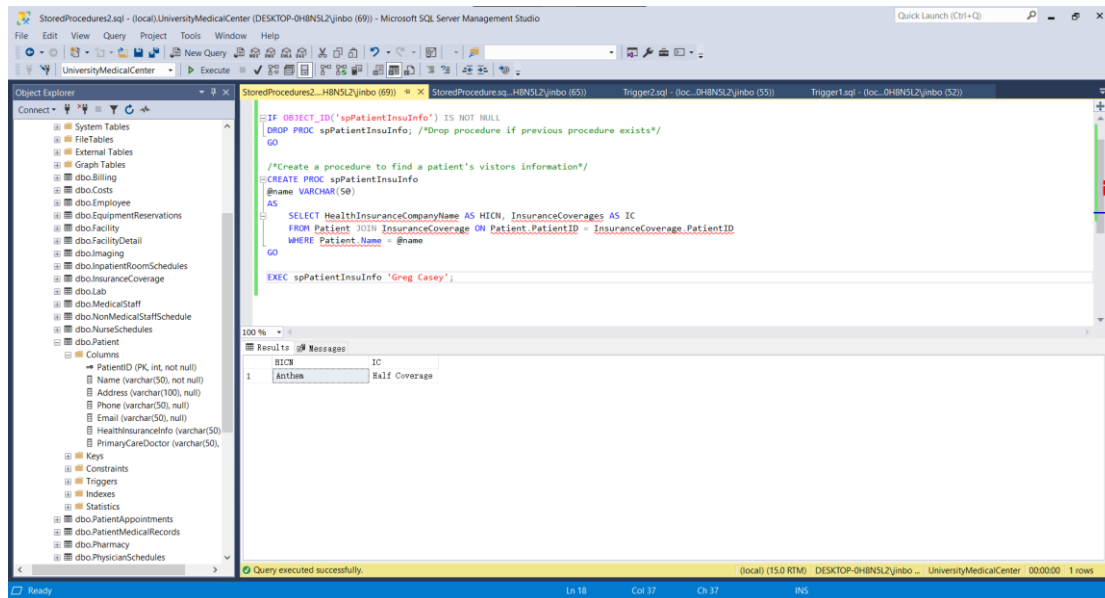
FROM Patient JOIN InsuranceCoverage ON Patient.PatientID =

InsuranceCoverage.PatientID

WHERE Patient.Name = @name

GO

EXEC spPatientInsuInfo 'Greg Casey';



### 3. User defined functions

- 1) In this function, it uses InsuranceID to find PatientID and Insurance Coverage.

--Function 1

USE UniversityMedicalCenter

GO

CREATE FUNCTION InsurCov

(@InsuranceID INT)

RETURNS INT

BEGIN

RETURN (SELECT InsuranceID FROM InsuranceCoverage

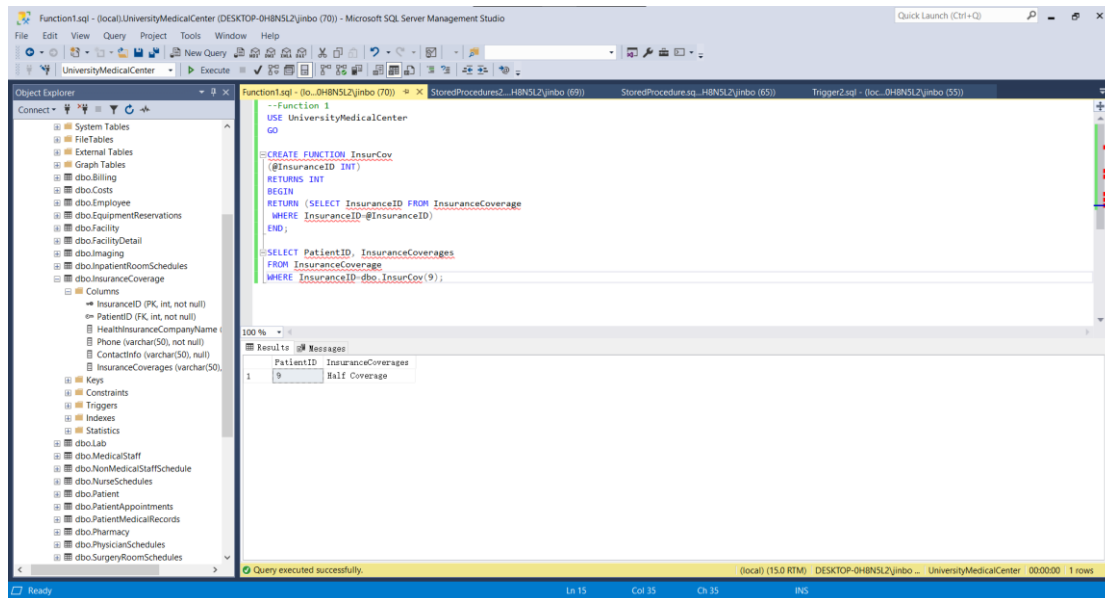
WHERE InsuranceID=@InsuranceID)

END;

SELECT PatientID, InsuranceCoverages

FROM InsuranceCoverage

WHERE InsuranceID=dbo.InsurCov(9);



2) This function is used to find Employee information by input employeeID.

--Function 2

```

USE UniversityMedicalCenter
GO

```

```

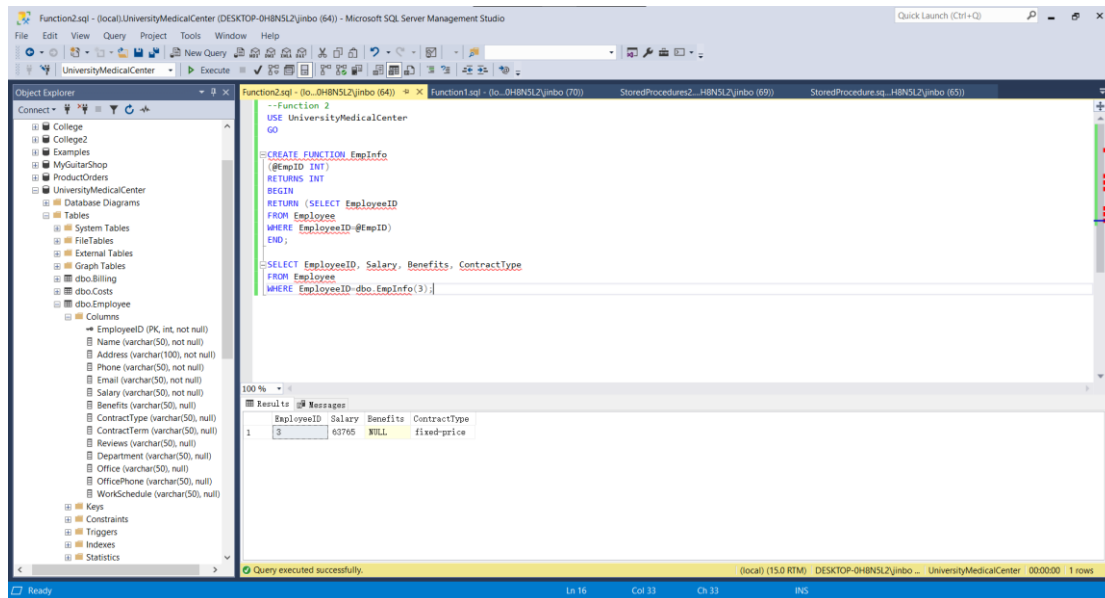
CREATE FUNCTION EmpInfo
(@EmpID INT)
RETURNS INT
BEGIN
RETURN (SELECT EmployeeID
FROM Employee
WHERE EmployeeID=@EmpID)
END;

```

```

SELECT EmployeeID, Salary, Benefits, ContractType
FROM Employee
WHERE EmployeeID=dbo.EmpInfo(3);

```



#### 4. Triggers

- 1) This CREATE TRIGGER statement makes the name of primary care doctor after inserted into Patient Table become Upper case form mixed case. The two select statements help to verify the change.

```

--TRIGGER 1
USE UniversityMedicalCenter
GO

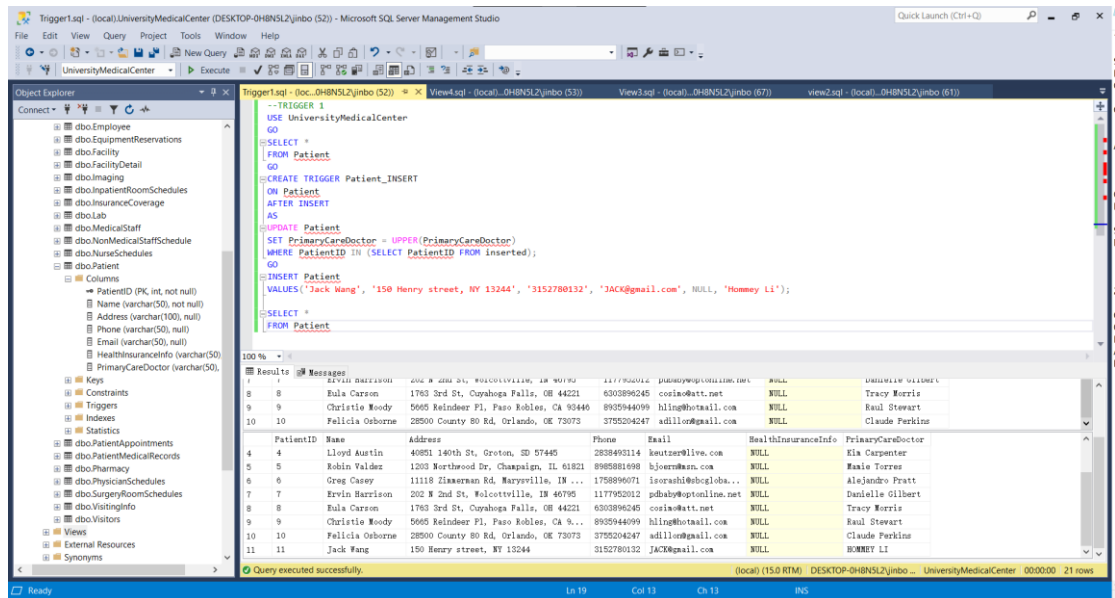
SELECT *
FROM Patient
GO

CREATE TRIGGER Patient_INSERT
ON Patient
AFTER INSERT
AS
UPDATE Patient
SET PrimaryCareDoctor = UPPER(PrimaryCareDoctor)
WHERE PatientID IN (SELECT PatientID FROM inserted);
GO

INSERT Patient
VALUES('Jack Wang', '150 Henry street, NY 13244', '3152780132',
'JACK@gmail.com', NULL, 'Hommey Li');

SELECT *
FROM Patient

```



2) This trigger is used to prevent changing data.

--TRIGGER 2

USE UniversityMedicalCenter

GO

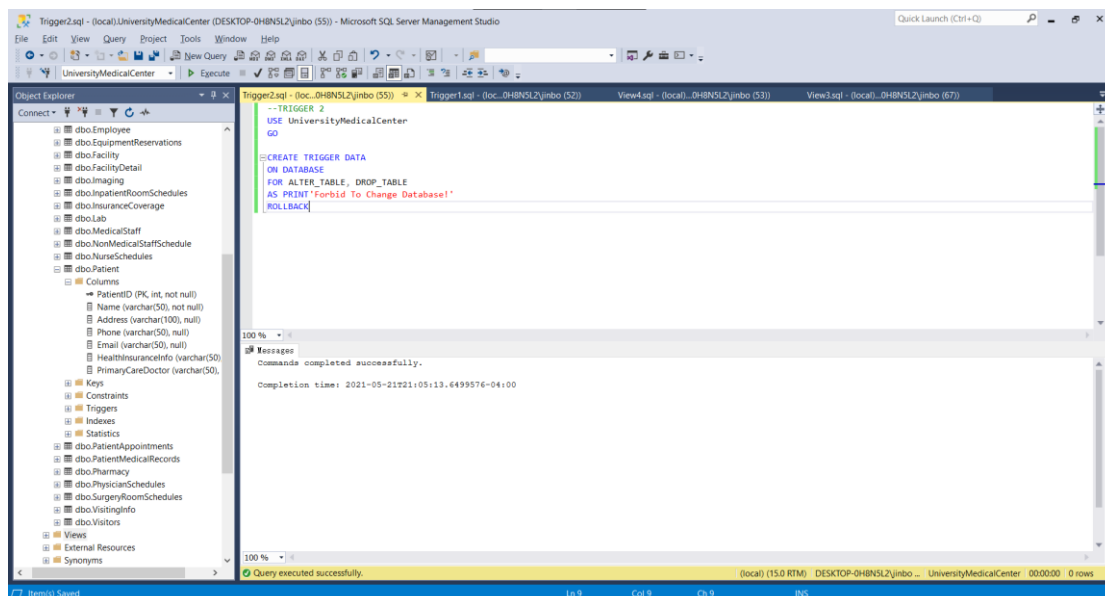
CREATE TRIGGER DATA

ON DATABASE

FOR ALTER\_TABLE, DROP\_TABLE

AS PRINT 'Forbid To Change Database!'

ROLLBACK



## 5. Transactions

1) In the hospital, one patient's primary care doctor should be change, because the former doctor has retired. The transaction update the primary care doctor which patient's ID is 1. The following select statement verify the result.



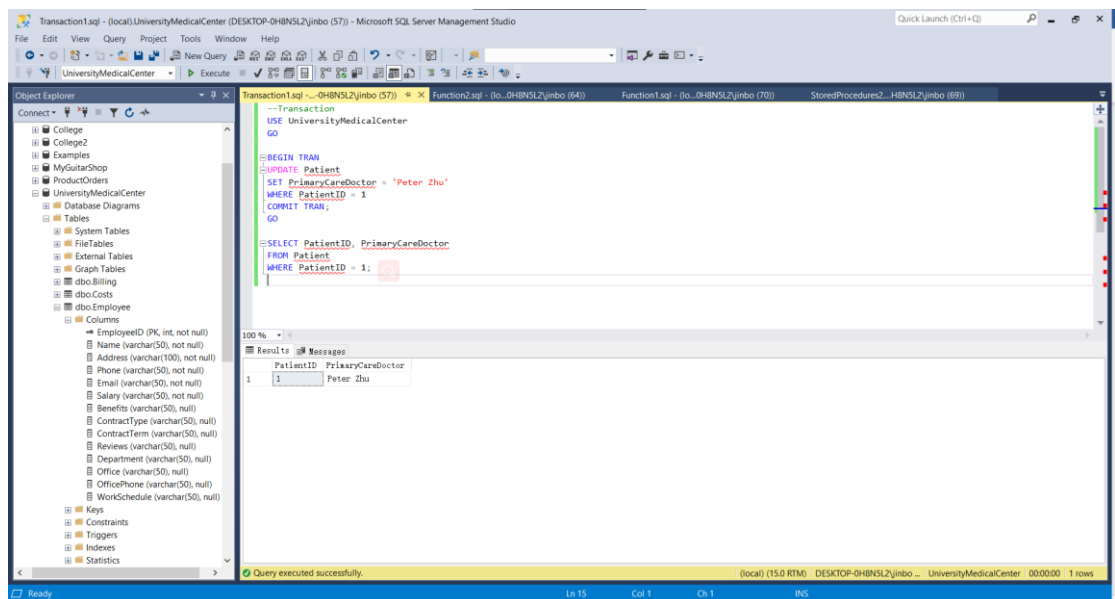
```

--Transaction
USE UniversityMedicalCenter
GO

BEGIN TRAN
UPDATE Patient
SET PrimaryCareDoctor = 'Peter Zhu'
WHERE PatientID = 1
COMMIT TRAN;
GO

SELECT PatientID, PrimaryCareDoctor
FROM Patient
WHERE PatientID = 1;

```



## 6. Security

- 1) This script creates a user-defined database role named PatientEntry in the database and gives UPDATE permission to the new role for the Patient table. Then it creates a login ID named “Happy2021” with the password “123456” and sets the default database for the login to the UniversityMedicalCenter database. The script creates a user named “Jinbo” for the login and assigns the user to the PatientEntry role.

```

CREATE ROLE PatientEntry; -- create user defined database role
GRANT UPDATE -- set update permission
ON Patient --set table
TO PatientEntry;

```

```

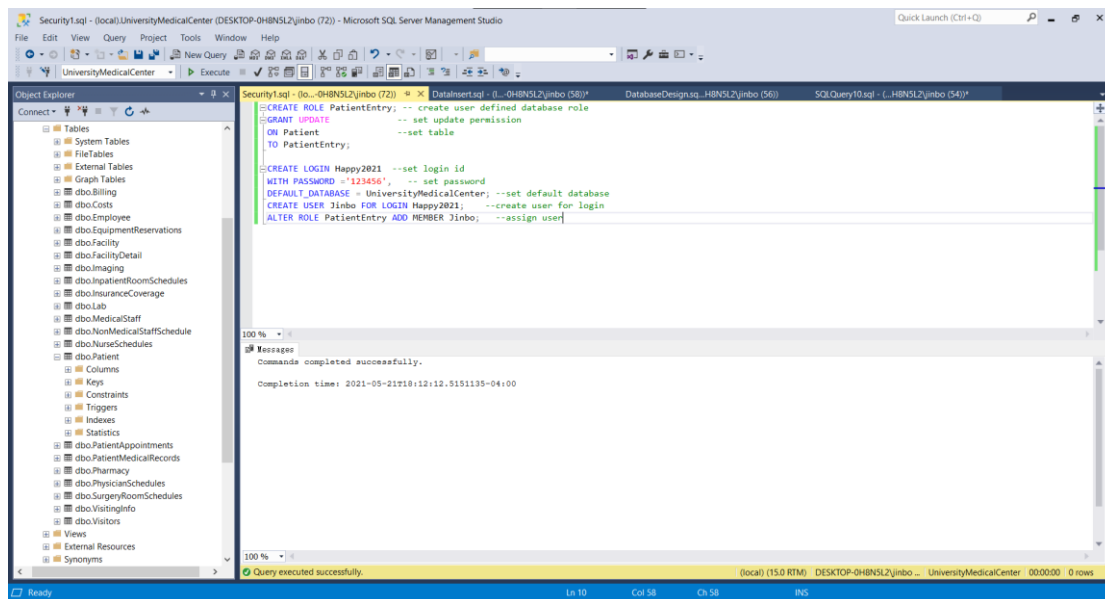
CREATE LOGIN Happy2021 --set login id
WITH PASSWORD = '123456', -- set password

```

```

DEFAULT_DATABASE = UniversityMedicalCenter; --set default database
CREATE USER Jinbo FOR LOGIN Happy2021;    --create user for login
ALTER ROLE PatientEntry ADD MEMBER Jinbo;  --assign user

```



- 2) This script is similar with above script. But with different security levels, passwords, and roles.

```

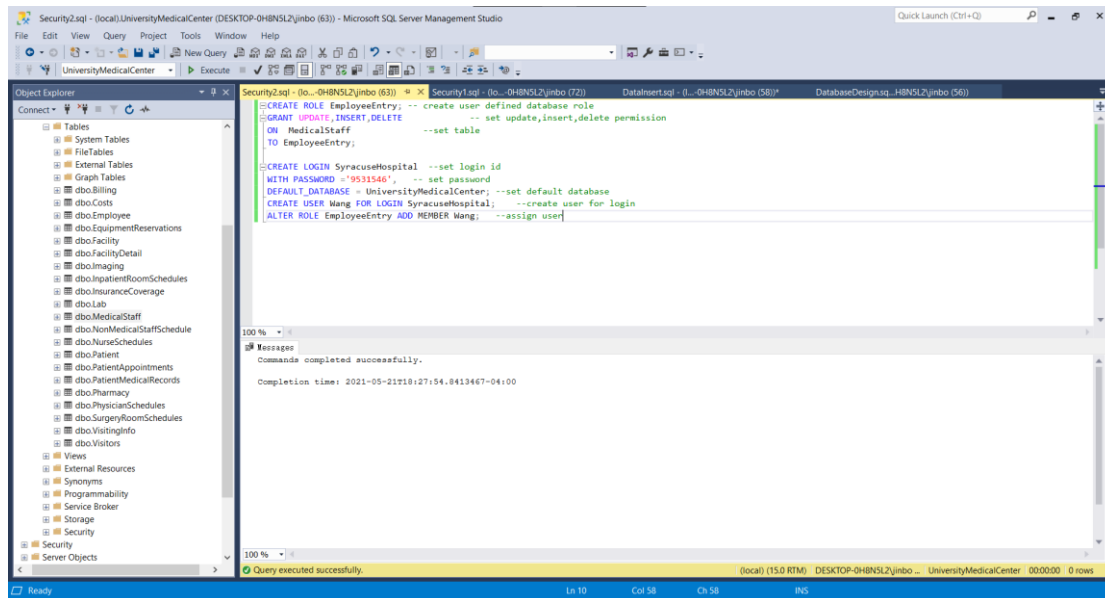
CREATE ROLE EmployeeEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON MedicalStaff            --set table
TO EmployeeEntry;

```

```

CREATE LOGIN SyracuseHospital --set login id
WITH PASSWORD = '9531546',    -- set password
DEFAULT_DATABASE = UniversityMedicalCenter; --set default database
CREATE USER Wang FOR LOGIN SyracuseHospital; --create user for login
ALTER ROLE EmployeeEntry ADD MEMBER Wang; --assign use

```



#### IV. Conclusion

Project analysis:

This project is very meaningful which helps to establish a database in every aspects. Through design, implement, test, and fix, I learned a lot and encountered some difficulties which did not appear in previous labs. Needless to say, I can better use SQL statement and have a better understanding.

The Microsoft SQL server management is useful in building a real world database. It is easy to retrieve and update, reliable backup and security.

Remarks:

Through this project, I find the interest in thinking the relationship between objects. In daily life, we may omit some basic relationships. And if we realized to efficiently filter them, things do not as simple as we imagine.