CSE 581 Intro to DBMS

PROJECT2

Rahul Ravikanti

981682170

Table Of Contents

Abstract

The development of a database for the Patient Access and Registration department at the University Medical Center is a significant project aimed at improving patient care and promoting coordination among healthcare providers. The database will be responsible for capturing patient data from admission to discharge, with a focus on optimizing hospital operations, reducing errors, and promoting resource utilization. Since the University Medical Center is a complex organization, the database will need to interface with several administrative departments, including Human Resources, Finance, Information Technology, Marketing, Health Information Management, Facilities Management, Compliance and Legal, and Quality and Risk Management.

The primary goal of this project is to streamline patient registration, including pre-registration, check-in, personal and insurance information, consent forms, payment collection, and medical history. Once registered, patients will undergo several processes, such as admission, diagnosis, treatment, monitoring, tests and procedures, consultations, and discharge planning before leaving the hospital. By organizing patient data in this manner, the database will ensure smooth patient transitions from hospital to home, improve patient care, and optimize resource utilization. Moreover, by providing accurate and up-to-date patient information to healthcare providers, the database will help reduce errors and facilitate coordination among different departments.

Overall, the development of a database for the Patient Access and Registration department at the University Medical Center is a vital initiative towards enhancing patient care and improving hospital operations. By promoting coordination among healthcare providers, optimizing resource utilization, and reducing errors, the database will help the University Medical Center provide better patient care and outcomes.

Tables Creation

The code for creation of tables is as follows:

```
CREATE TABLE Insurances(
   InsuranceID INT PRIMARY KEY IDENTITY(1,1),
   InsuranceName NVARCHAR(100) NOT NULL,
   InsuranceNumber NVARCHAR(100) NOT NULL,
   InsurancePlanType NVARCHAR(50) NOT NULL,
   InsuranceEffectiveDate DATE NOT NULL,
   InsuranceExpirationDate DATE NOT NULL,
);

CREATE TABLE LegalLicenses (
   LicenseID INT PRIMARY KEY IDENTITY(1,1),
   LicenseName NVARCHAR(100) NOT NULL,
   LicenseDescription NVARCHAR(500) NULL
);

CREATE TABLE HealthDepartments (
   DepartmentID INT PRIMARY KEY IDENTITY(1,1),
   DepartmentName NVARCHAR(100) NOT NULL,
   DepartmentDescription NVARCHAR(500) NULL
);

CREATE TABLE Patients (
   PatientID INT PRIMARY KEY IDENTITY(1,1),
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50) NOT NULL,
   DateOfBirth DATE NOT NULL,
   Gender CHAR(1) NOT NULL,
   PhoneNumber VARCHAR(20) NOT NULL,
   Address VARCHAR(100) NOT NULL,
   City VARCHAR(50) NOT NULL,
   State VARCHAR(50) NOT NULL,
   ZipCode VARCHAR(10) NOT NULL,
   InsuranceID INT NOT NULL,
   FOREIGN KEY (InsuranceID) REFERENCES Insurances (InsuranceID),
);
```

```
CREATE TABLE Physicians (
   PhysicianID INT PRIMARY KEY IDENTITY(1,1),
   FirstName NVARCHAR(50) NOT NULL,
   LastName NVARCHAR(50) NOT NULL,
   DateOfBirth DATE NOT NULL,
   Gender NVARCHAR(10) NOT NULL,
   PhoneNumber NVARCHAR(20) NOT NULL,
   LicenseID INT NOT NULL,
   DepartmentID INT NOT NULL,
   CONSTRAINT FK_Physicians_License FOREIGN KEY (LicenseID) REFERENCES
LegalLicenses(LicenseID),
   CONSTRAINT FK_Physicians_Department FOREIGN KEY (DepartmentID) REFERENCES
HealthDepartments(DepartmentID)
);

CREATE TABLE Nurses (
  NurseID INT PRIMARY KEY IDENTITY(1,1),
  FirstName VARCHAR(255) NOT NULL,
  LastName VARCHAR(255) NOT NULL,
  Gender CHAR(1) NOT NULL,
  DateOfBirth DATE NOT NULL,
  PhoneNumber VARCHAR(20) NOT NULL,
  Address VARCHAR(255) NOT NULL,
  City VARCHAR(255) NOT NULL,
  State VARCHAR(255) NOT NULL,
  ZipCode VARCHAR(10) NOT NULL,
  DutiesID INT NOT NULL,
  FOREIGN KEY (DutiesID) REFERENCES MedicalDuties(DutiesID)
);

CREATE TABLE Notes (
   NoteID INT PRIMARY KEY IDENTITY(1,1),
   PatientID INT NOT NULL,
   PhysicianID INT NOT NULL,
   NurseID INT NOT NULL,
   NoteDate DATETIME NOT NULL,
   NoteText NVARCHAR(MAX) NOT NULL,
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
   FOREIGN KEY (NurseID) REFERENCES Nurses(NurseID)
);
```

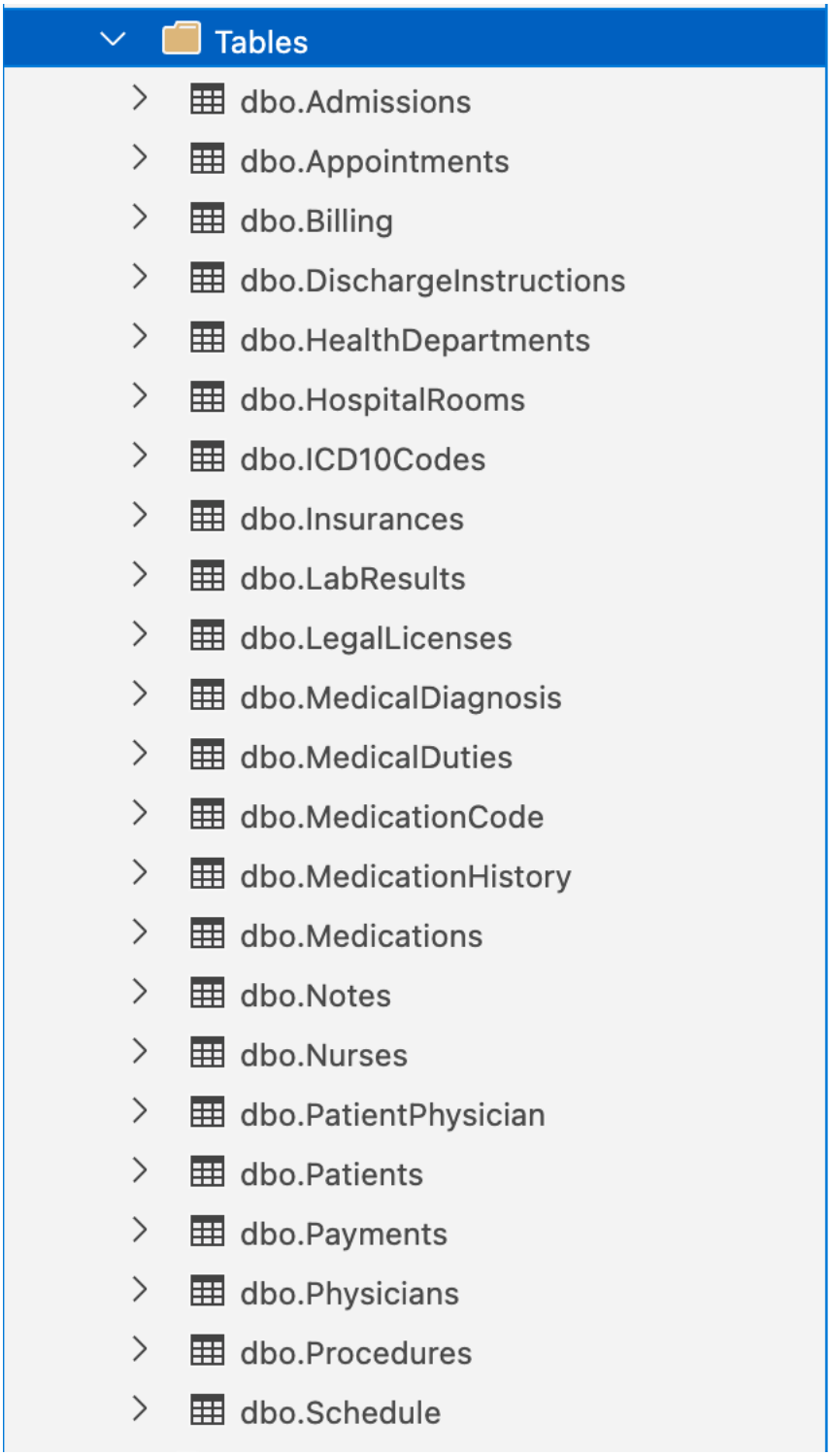
```
CREATE TABLE Appointments (
   AppointmentID INT PRIMARY KEY IDENTITY(1,1),
   PatientPhysicianID INT NOT NULL,
   AppointmentDateTime DATETIME NOT NULL,
   Reason VARCHAR(255) NOT NULL,
   CONSTRAINT FK_Appointments_PatientPhysicianID FOREIGN KEY (PatientPhysicianID)
REFERENCES PatientPhysician (PatientPhysicianID),
   CONSTRAINT UQ_Appointments_AppointmentDateTime UNIQUE (AppointmentDateTime)
);

CREATE TABLE Admissions (
   AdmissionID INT PRIMARY KEY IDENTITY(1,1),
   PatientID INT NOT NULL,
   PhysicianID INT NOT NULL,
   RoomID INT NOT NULL,
   DiagnosisID INT NOT NULL,
   AdmissionDate DATETIME NOT NULL,
   DischargeDate DATETIME NULL DEFAULT NULL,
   FOREIGN KEY (DiagnosisID) REFERENCES MedicalDiagnosis(DiagnosisID),
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (RoomID) REFERENCES HospitalRooms(RoomID),
   FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
   CHECK (AdmissionDate <= DischargeDate)
);


CREATE TABLE Procedures (
  ProcedureID INT PRIMARY KEY IDENTITY(1,1),
  ICD10CodeID INT NOT NULL,
  PatientPhysicianID INT NOT NULL,
  ProcedureDate DATE NOT NULL,
  FOREIGN KEY (ICD10CodeID) REFERENCES ICD10Codes (ICD10CodeID),
  FOREIGN KEY (PatientPhysicianID) REFERENCES PatientPhysician (PatientPhysicianID)
);

CREATE TABLE Billing (
  BillingID INT PRIMARY KEY IDENTITY(1,1),
  PatientID INT NOT NULL,
  PaymentID INT NOT NULL,
  BillingDate DATE NOT NULL,
  Amount DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (PatientID) REFERENCES Patients (PatientID),
```

```
  FOREIGN KEY (PaymentID) REFERENCES Payments (PaymentID),
);

CREATE TABLE LabResults(
   LabResultsID INT PRIMARY KEY IDENTITY(1,1),
   TestName VARCHAR(100),
   TestDate DATE,
   TestResults VARCHAR(100),
   PatientID INT,
   PhysicianID INT,
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
   FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID)
);

CREATE TABLE Medications (
   MedicationID INT PRIMARY KEY IDENTITY(1,1),
   MedicationName VARCHAR(100),
   Dosage VARCHAR(50),
   Frequency VARCHAR(50),
   PrescribingPhysicianID INT,
   PatientID INT,
   FOREIGN KEY (PrescribingPhysicianID) REFERENCES Physicians(PhysicianID),
   FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);

CREATE TABLE ICD10Codes (
  ICD10CodeID INT PRIMARY KEY IDENTITY(1,1),
  Code VARCHAR(10) NOT NULL,
  Description VARCHAR(255) NOT NULL
);

CREATE TABLE Payments (
   PaymentID INT PRIMARY KEY IDENTITY(1,1),
   PaymentDate DATE NOT NULL,
   Amount DECIMAL(10,2) NOT NULL
);
```

```
CREATE TABLE PatientPhysician (
   PatientPhysicianID INT PRIMARY KEY IDENTITY(1,1),
   PatientID INT NOT NULL,
   PhysicianID INT NOT NULL,
   CONSTRAINT FK_PatientPhysician_PatientID FOREIGN KEY (PatientID) REFERENCES Patients
(PatientID),
   CONSTRAINT FK_PatientPhysician_PhysicianID FOREIGN KEY (PhysicianID) REFERENCES
Physicians (PhysicianID)
);

CREATE TABLE MedicalDuties (
   DutiesID INT PRIMARY KEY IDENTITY(1,1),
   DutiesName NVARCHAR(100) NOT NULL,
   DutiesDescription NVARCHAR(500) NULL
);



CREATE TABLE HospitalRooms (
   RoomID INT PRIMARY KEY IDENTITY(1,1),
   RoomNumber NVARCHAR(50) NOT NULL,
   RoomType NVARCHAR(50) NOT NULL,
   RoomDescription NVARCHAR(500) NULL
);

CREATE TABLE MedicalDiagnosis (
  DiagnosisID INT PRIMARY KEY IDENTITY(1,1),
  PatientID INT NOT NULL,
  PhysicianID INT NOT NULL,
  ICD10CodeID INT NOT NULL,
  DiagnosisDate DATE NOT NULL,
  FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
  FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
  FOREIGN KEY (ICD10CodeID) REFERENCES ICD10Codes (ICD10CodeID),
);
```

```
CREATE TABLE MedicationHistory(
MedicationHistoryID INT PRIMARY KEY IDENTITY(1,1),
PatientID INT NOT NULL,
MedicationID INT NOT NULL,
FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
FOREIGN KEY (MedicationID) REFERENCES Medications(MedicationID)
);

CREATE TABLE MedicationCode (
   MedicationCodeID INT PRIMARY KEY IDENTITY(1,1),
   MedicationID INT,
   CodeValue VARCHAR(50),
   CodeDescription VARCHAR(255),
   FOREIGN KEY (MedicationID) REFERENCES Medications(MedicationID)
);

CREATE TABLE Schedule (
   ScheduleID INT PRIMARY KEY IDENTITY(1,1),
   PhysicianID INT NOT NULL,
   AppointmentID INT NOT NULL,
   ScheduleDate DATE NOT NULL,
   ScheduleTime TIME NOT NULL,
   FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID),
   FOREIGN KEY (AppointmentID) REFERENCES Appointments(AppointmentID),
   UNIQUE (PhysicianID, ScheduleDate, ScheduleTime),
   CHECK (ScheduleDate >= GETDATE()),
   CHECK (ScheduleTime >= '00:00:00' AND ScheduleTime <= '23:59:59')
);

CREATE TABLE DischargeInstructions(
DischargeInstructionID INT PRIMARY KEY IDENTITY(1,1),
PatientID INT NOT NULL,
PhysicianID INT NOT NULL,
DischargeInstructions VARCHAR(250),
DischargeDate DATE NOT NULL,
FOREIGN KEY (PatientID) REFERENCES Patients (PatientID),
FOREIGN KEY (PhysicianID) REFERENCES Physicians(PhysicianID)
);
```

The tables created in UniversityMedicalCenter



The insert statements for inserting data into above tables

```
INSERT INTO LegalLicenses (LicenseName, LicenseDescription) VALUES
('Medical License', 'This license allows a person to practice medicine legally.'),
('Nursing License', 'This license allows a person to practice nursing legally.'),
('Pharmacy License', 'This license allows a person to operate a pharmacy legally.'),
('Dental License', 'This license allows a person to practice dentistry legally.'),
('Physical Therapy License', 'This license allows a person to practice physical therapy legally.'),
('Occupational Therapy License', 'This license allows a person to practice occupational therapy
legally.'),
('Chiropractic License', 'This license allows a person to practice chiropractic care legally.'),
('Psychologist License', 'This license allows a person to practice psychology legally.'),
('Veterinary License', 'This license allows a person to practice veterinary medicine legally.'),
('EMT License', 'This license allows a person to practice emergency medical services legally.');
```

```
INSERT INTO Insurances (InsuranceName, InsuranceNumber, InsurancePlanType,
InsuranceEffectiveDate, InsuranceExpirationDate) VALUES
('Blue Cross Blue Shield', '12345', 'PPO', '2022-01-01', '2022-12-31'),
('Aetna', '98765', 'HMO', '2022-02-01', '2022-11-30'),
('Cigna', '45678', 'POS', '2022-03-01', '2022-12-31'),
('Humana', '87654', 'EPO', '2022-04-01', '2022-10-31'),
('United Healthcare', '23456', 'HSA', '2022-05-01', '2022-12-31'),
('Kaiser Permanente', '56789', 'HMO', '2022-06-01', '2022-11-30'),
('Molina Healthcare', '54321', 'PPO', '2022-07-01', '2022-12-31'),
('Oscar Health', '67890', 'POS', '2022-08-01', '2022-10-31'),
('Medicare', '78901', 'Original Medicare', '2022-09-01', '2022-12-31'),
('Medicaid', '90123', 'Managed Medicaid', '2022-10-01', '2022-12-31');
```

```
INSERT INTO HealthDepartments (DepartmentName, DepartmentDescription) VALUES
('Pediatrics', 'Department of medicine focused on the care of infants, children, and adolescents.'),
('Ophthalmology', 'Department of medicine focused on the diagnosis and treatment of eye disorders.'),
('Orthopedics', 'Department of medicine focused on the diagnosis and treatment of musculoskeletal disorders.'),
('Neurology', 'Department of medicine focused on the diagnosis and treatment of disorders of the nervous system.'),
('Cardiology', 'Department of medicine focused on the diagnosis and treatment of heart and cardiovascular disorders.'),
('Gastroenterology', 'Department of medicine focused on the diagnosis and treatment of digestive system disorders.'),
('Obstetrics and Gynecology', 'Department of medicine focused on the care of women during pregnancy, childbirth, and the postpartum period.'),
('Dermatology', 'Department of medicine focused on the diagnosis and treatment of skin disorders.'),
('Otolaryngology', 'Department of medicine focused on the diagnosis and treatment of ear, nose, and throat disorders.'),
('Psychiatry', 'Department of medicine focused on the diagnosis and treatment of mental health disorders.');
```

```
INSERT INTO Patients (FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Address, City, State, ZipCode, InsuranceID) VALUES
('John', 'Doe', '1985-06-12', 'M', '555-555-1212', '123 Main St', 'Anytown', 'CA', '12345', 1),
('Jane', 'Smith', '1990-09-22', 'F', '555-555-1313', '456 Maple Ave', 'Othertown', 'NY', '54321', 2),
('David', 'Lee', '1972-03-04', 'M', '555-555-1414', '789 Oak Blvd', 'Somewhere', 'TX', '67890', 3),
('Lisa', 'Nguyen', '1988-12-31', 'F', '555-555-1515', '321 Elm St', 'Anytown', 'CA', '12345', 1),
('Michael', 'Garcia', '1975-07-20', 'M', '555-555-1616', '654 Cedar Ave', 'Othertown', 'NY', '54321', 2),
('Emily', 'Kim', '1995-02-15', 'F', '555-555-1717', '987 Pine St', 'Somewhere', 'TX', '67890', 3),
('William', 'Johnson', '1980-11-02', 'M', '555-555-1818', '555 Fifth St', 'Anytown', 'CA', '12345', 1),
('Sarah', 'Martinez', '1989-04-27', 'F', '555-555-1919', '777 Sixth Ave', 'Othertown', 'NY', '54321', 2),
('Daniel', 'Rodriguez', '1978-08-14', 'M', '555-555-2020', '111 Seventh St', 'Somewhere', 'TX', '67890', 3),
('Megan', 'Chen', '1993-01-07', 'F', '555-555-2121', '222 Eighth Ave', 'Anytown', 'CA', '12345', 1);
```

```
INSERT INTO Physicians (FirstName, LastName, DateOfBirth, Gender, PhoneNumber, LicenseID,
DepartmentID) VALUES
('Sophia', 'Lee', '1986-09-12', 'F', '555-555-1212', 11, 7),
('Ethan', 'Nguyen', '1978-11-23', 'M', '555-555-1313', 12, 8),
('Ava', 'Garcia', '1991-02-01', 'F', '555-555-1414', 13, 9),
('Jacob', 'Kim', '1983-05-31', 'M', '555-555-1515', 14, 10),
('Isabella', 'Johnson', '1988-08-10', 'F', '555-555-1616', 15, 11),
('Noah', 'Martinez', '1985-03-17', 'M', '555-555-1717', 16, 12),
('Olivia', 'Rodriguez', '1993-07-28', 'F', '555-555-1818', 17, 13),
('Liam', 'Chen', '1980-12-02', 'M', '555-555-1919', 18, 14),
('Emma', 'Gonzalez', '1982-01-15', 'F', '555-555-2020', 19, 15),
('Mason', 'Perez', '1974-09-30', 'M', '555-555-2121', 20, 16);


INSERT INTO PatientPhysician (PatientID, PhysicianID)
VALUES (1, 2),
(2, 3),
(3, 4),
(4, 5),
(5, 6),
(6, 7),
(7, 8),
(8, 9),
(9, 10),
(10, 1);

INSERT INTO Medications (MedicationName, Dosage, Frequency, PrescribingPhysicianID,
PatientID)
VALUES ('Lisinopril', '20mg', 'Daily', 3, 1),
    ('Metformin', '1000mg', 'Twice daily', 5, 2),
    ('Levothyroxine', '75mcg', 'Daily', 2, 3),
    ('Atorvastatin', '40mg', 'Nightly', 1, 4),
    ('Albuterol', '90mcg', 'As needed', 4, 5),
    ('Ibuprofen', '800mg', 'Every 8 hours', 2, 6),
    ('Hydrocodone', '10mg', 'As needed', 6, 7),
    ('Trazodone', '50mg', 'Nightly', 1, 8),
    ('Omeprazole', '20mg', 'Daily', 4, 9),
    ('Aspirin', '81mg', 'Daily', 5, 10);
```

```
INSERT INTO MedicationHistory (PatientID, MedicationID)
VALUES (1, 1),
(2, 3),
(3, 5),
(4, 7),
(5, 9),
(6, 2),
(7, 4),
(8, 6),
(9, 8),
(10, 10);

INSERT INTO MedicalDuties (DutiesName, DutiesDescription)
VALUES
('Diagnosing and treating illnesses', 'Physicians diagnose and treat illnesses and injuries'),
('Prescribing and administering medication', 'Physicians prescribe and administer medications
to patients'),
('Performing medical procedures', 'Physicians perform medical procedures such as surgeries'),
('Ordering and interpreting diagnostic tests', 'Physicians order and interpret diagnostic tests
such as blood tests, X-rays, and CT scans'),
('Providing preventative care and health education', 'Physicians provide preventative care and
health education to patients'),
('Managing patient care and medical records', 'Physicians manage patient care and medical
records'),
('Collaborating with other healthcare professionals', 'Physicians collaborate with other
healthcare professionals such as nurses and therapists'),
('Managing and supervising medical staff', 'Physicians manage and supervise medical staff such
as nurses and medical assistants'),
('Researching and developing new treatments and technologies', 'Physicians participate in
research and development of new treatments and technologies'),
('Advocating for patient rights and healthcare policy', 'Physicians advocate for patient rights and
healthcare policy');
```

```
INSERT INTO Nurses (FirstName, LastName, Gender, DateOfBirth, PhoneNumber, Address, City, State, ZipCode, DutiesID)
VALUES
('Emily', 'Smith', 'F', '1990-05-12', '555-1234', '123 Main St', 'New York', 'NY', '10001', 1),
('Megan', 'Johnson', 'F', '1993-09-15', '555-2345', '456 Maple Ave', 'Los Angeles', 'CA', '90001', 2),
('Liam', 'Brown', 'M', '1991-02-20', '555-3456', '789 Oak St', 'Chicago', 'IL', '60007', 1),
('Avery', 'Davis', 'F', '1989-11-25', '555-4567', '321 Pine Rd', 'Miami', 'FL', '33010', 3),
('Noah', 'Garcia', 'M', '1992-07-04', '555-5678', '654 Cedar Blvd', 'Dallas', 'TX', '75001', 2),
('Abigail', 'Martinez', 'F', '1988-12-30', '555-6789', '987 Elm St', 'San Francisco', 'CA', '94101', 3),
('Ethan', 'Wilson', 'M', '1994-03-10', '555-7890', '246 Birch Dr', 'Boston', 'MA', '02101', 1),
('Olivia', 'Anderson', 'F', '1990-08-22', '555-8901', '753 Oak Ave', 'Seattle', 'WA', '98101', 2),
('Mason', 'Taylor', 'M', '1993-01-17', '555-9012', '159 Maple Ln', 'Houston', 'TX', '77001', 3),
('Sophia', 'Thomas', 'F', '1989-06-03', '555-0123', '852 Elm Blvd', 'Atlanta', 'GA', '30301', 1);

INSERT INTO Notes (PatientID, PhysicianID, NurseID, NoteDate, NoteText)
VALUES
   (1, 6, 1, '2023-04-29 13:00:00', 'Patient presented with mild fever and cough. Prescribed antibiotics.'),
   (2, 7, 2, '2023-04-29 14:30:00', 'Patient complaining of pain in left knee. Advised to rest and prescribed pain medication.'),
   (3, 8, 3, '2023-04-29 16:00:00', 'Patient recovering well from surgery. Incision site looks clean and dry.'),
   (4, 9, 4, '2023-04-30 09:00:00', 'Patient reports feeling dizzy and lightheaded. Vital signs within normal range.'),
   (5, 10, 5, '2023-04-30 10:30:00', 'Patient presented with rash on arms and legs. Prescribed antihistamines.'),
   (6, 6, 6, '2023-04-30 12:00:00', 'Patient feeling better after taking antibiotics. Advised to complete full course of medication.'),
   (7, 7, 7, '2023-04-30 13:30:00', 'Patient reports feeling more pain in left knee. Prescribed stronger pain medication.'),
   (8, 8, 8, '2023-04-30 15:00:00', 'Patient discharged from hospital. Instructed to follow up with physician in two weeks.'),
   (9, 9, 9, '2023-04-30 16:30:00', 'Patient feeling better after resting for an hour. Discharged from emergency department.'),
   (10, 10, 10, '2023-05-01 09:00:00', 'Patient reports no improvement in rash. Referred to dermatologist for further evaluation.');
```

```
INSERT INTO Appointments (PatientPhysicianID, AppointmentDateTime, Reason)
VALUES
(1, '2023-05-10 10:00:00', 'Routine check-up'),
(2, '2023-05-11 14:30:00', 'Follow-up after surgery'),
(3, '2023-05-12 09:00:00', 'Annual physical exam'),
(4, '2023-05-13 11:00:00', 'Consultation for back pain'),
(5, '2023-05-14 13:00:00', 'Vaccination appointment'),
(6, '2023-05-15 15:30:00', 'Consultation for allergies'),
(7, '2023-05-16 16:00:00', 'Follow-up for diabetes management'),
(8, '2023-05-17 10:30:00', 'Routine check-up'),
(9, '2023-05-18 11:30:00', 'Follow-up after hospitalization'),
(10, '2023-05-19 14:00:00', 'Consultation for weight management');

INSERT INTO ICD10Codes (Code, Description)
VALUES
   ('A00', 'Cholera'),
   ('A01', 'Typhoid and paratyphoid fevers'),
   ('A02', 'Other salmonella infections'),
   ('A03', 'Shigellosis'),
   ('A04', 'Other bacterial intestinal infections'),
   ('A05', 'Other bacterial foodborne intoxications, not elsewhere classified'),
   ('A06', 'Amoebiasis'),
   ('A07', 'Other protozoal intestinal diseases'),
   ('A08', 'Viral and other specified intestinal infections'),
   ('A09', 'Diarrhoea and gastro-enteritis of presumed infectious origin');

INSERT INTO MedicalDiagnosis (PatientID, PhysicianID, ICD10CodeID, DiagnosisDate)
VALUES
(1, 6, 1, '2023-04-29'),
(2, 7, 2, '2023-04-29'),
(3, 8, 3, '2023-04-29'),
(4, 9, 4, '2023-04-30'),
(5, 10, 5, '2023-04-30'),
(6, 6, 6, '2023-04-30'),
(7, 7, 7, '2023-04-30'),
(8, 8, 8, '2023-04-30'),
(9, 9, 9, '2023-04-30'),
(10, 10, 10, '2023-05-01');
```

```
INSERT INTO HospitalRooms (RoomNumber, RoomType, RoomDescription)
VALUES
('101', 'Standard', 'A standard room with a single bed'),
('102', 'Standard', 'A standard room with a single bed'),
('103', 'Standard', 'A standard room with a single bed'),
('201', 'Standard', 'A standard room with a single bed'),
('202', 'Standard', 'A standard room with a single bed'),
('203', 'Standard', 'A standard room with a single bed'),
('301', 'Deluxe', 'A deluxe room with two beds and a view of the garden'),
('302', 'Deluxe', 'A deluxe room with two beds and a view of the garden'),
('303', 'Deluxe', 'A deluxe room with two beds and a view of the garden'),
('401', 'Suite', 'A luxurious suite with a king-sized bed and a separate living area');
```

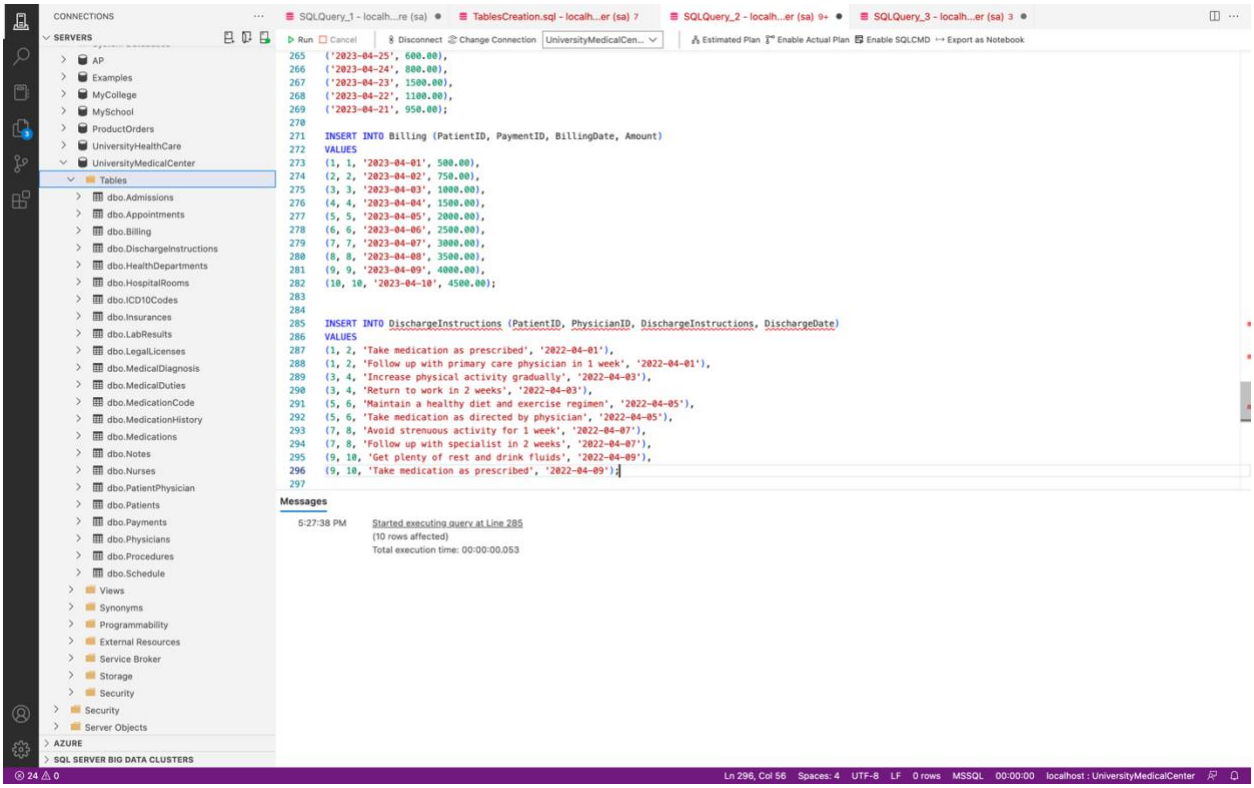
```
INSERT INTO Admissions (PatientID, PhysicianID, RoomID, DiagnosisID, AdmissionDate,
DischargeDate)
VALUES
(1, 3, 2, 1, '2022-04-01 12:00:00', '2022-04-05 10:00:00'),
(2, 2, 1, 3, '2022-05-10 08:00:00', '2022-05-15 14:00:00'),
(3, 1, 3, 2, '2022-06-20 16:00:00', '2022-06-22 18:00:00'),
(4, 4, 4, 4, '2022-07-03 10:00:00', '2022-07-06 12:00:00'),
(5, 2, 5, 5, '2022-08-01 07:00:00', '2022-08-02 16:00:00'),
(6, 3, 6, 6, '2022-09-05 14:00:00', NULL),
(7, 1, 7, 7, '2022-10-11 11:00:00', NULL),
(8, 4, 8, 8, '2022-11-20 08:00:00', NULL),
(9, 2, 9, 9, '2022-12-01 15:00:00', NULL),
(10, 3, 10, 10, '2023-01-02 10:00:00', NULL);
```

```
INSERT INTO Procedures (ICD10CodeID, PatientPhysicianID, ProcedureDate)
VALUES
   (1, 1, '2022-03-15'),
   (2, 2, '2022-04-20'),
   (3, 3, '2022-05-10'),
   (4, 4, '2022-06-01'),
   (5, 5, '2022-07-04'),
   (6, 6, '2022-08-15'),
   (7, 7, '2022-09-21'),
   (8, 8, '2022-10-10'),
   (9, 9, '2022-11-02'),
   (10, 10, '2022-12-25');
```

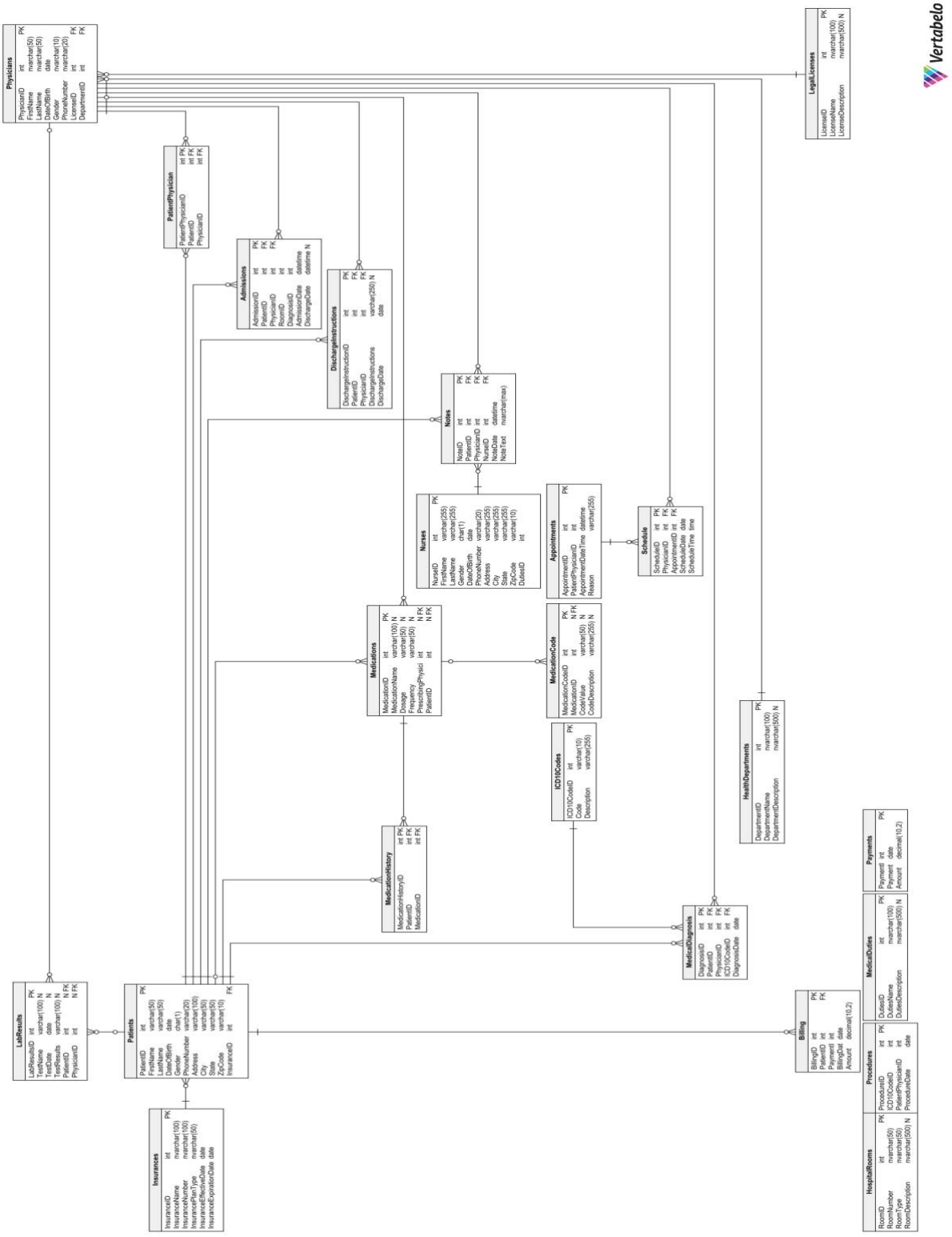
```
INSERT INTO MedicationCode (MedicationID, CodeValue, CodeDescription)
VALUES
   (1, '123', 'Medication A code 123'),
   (2, '456', 'Medication B code 456'),
   (3, '789', 'Medication C code 789'),
   (4, '321', 'Medication D code 321'),
   (5, '654', 'Medication E code 654'),
   (6, '987', 'Medication F code 987'),
   (7, '741', 'Medication G code 741'),
   (8, '852', 'Medication H code 852'),
   (9, '963', 'Medication I code 963'),
   (10, '159', 'Medication J code 159');

INSERT INTO LabResults (TestName, TestDate, TestResults, PatientID, PhysicianID)
VALUES
   ('Blood sugar test', '2023-04-29', '120 mg/dL', 1, 5),
   ('Urinalysis', '2023-04-28', 'Normal', 2, 6),
   ('Complete blood count', '2023-04-27', 'Normal', 3, 7),
   ('Electrocardiogram', '2023-04-26', 'Sinus rhythm', 4, 8),
   ('Stool culture', '2023-04-25', 'Positive for Salmonella', 5, 9),
   ('Thyroid function test', '2023-04-24', 'Normal', 6, 10),
   ('Liver function test', '2023-04-23', 'Elevated ALT levels', 7, 1),
   ('Kidney function test', '2023-04-22', 'Normal', 8, 2),
   ('Lipid panel', '2023-04-21', 'Elevated LDL cholesterol', 9, 3),
   ('Pregnancy test', '2023-04-20', 'Positive', 10, 4);

INSERT INTO Schedule (PhysicianID, AppointmentID, ScheduleDate, ScheduleTime)
VALUES
(1, 1, '2023-05-01', '09:00:00'),
(1, 2, '2023-05-02', '10:00:00'),
(1, 3, '2023-05-03', '11:00:00'),
(2, 4, '2023-05-04', '14:00:00'),
(2, 5, '2023-05-05', '15:00:00'),
(2, 6, '2023-05-06', '16:00:00'),
(3, 7, '2023-05-07', '09:00:00'),
(3, 8, '2023-05-08', '10:00:00'),
(3, 9, '2023-05-09', '11:00:00'),
(4, 10, '2023-05-10', '14:00:00');
```

```
INSERT INTO Payments (PaymentDate, Amount)
VALUES
('2023-04-30', 500.00),
('2023-04-29', 1000.00),
('2023-04-28', 750.00),
('2023-04-27', 1250.00),
('2023-04-26', 900.00),
('2023-04-25', 600.00),
('2023-04-24', 800.00),
('2023-04-23', 1500.00),
('2023-04-22', 1100.00),
('2023-04-21', 950.00);

INSERT INTO Billing (PatientID, PaymentID, BillingDate, Amount)
VALUES
(1, 1, '2023-04-01', 500.00),
(2, 2, '2023-04-02', 750.00),
(3, 3, '2023-04-03', 1000.00),
(4, 4, '2023-04-04', 1500.00),
(5, 5, '2023-04-05', 2000.00),
(6, 6, '2023-04-06', 2500.00),
(7, 7, '2023-04-07', 3000.00),
(8, 8, '2023-04-08', 3500.00),
(9, 9, '2023-04-09', 4000.00),
(10, 10, '2023-04-10', 4500.00);


INSERT INTO DischargeInstructions (PatientID, PhysicianID, DischargeInstructions,
DischargeDate)
VALUES
(1, 2, 'Take medication as prescribed', '2022-04-01'),
(1, 2, 'Follow up with primary care physician in 1 week', '2022-04-01'),
(3, 4, 'Increase physical activity gradually', '2022-04-03'),
(3, 4, 'Return to work in 2 weeks', '2022-04-03'),
(5, 6, 'Maintain a healthy diet and exercise regimen', '2022-04-05'),
(5, 6, 'Take medication as directed by physician', '2022-04-05'),
(7, 8, 'Avoid strenuous activity for 1 week', '2022-04-07'),
(7, 8, 'Follow up with specialist in 2 weeks', '2022-04-07'),
(9, 10, 'Get plenty of rest and drink fluids', '2022-04-09'),
(9, 10, 'Take medication as prescribed', '2022-04-09');
```

Successful execution of insert statements



The ER Diagram for the database UniversityMedicalCenter is as follows (Rotated).

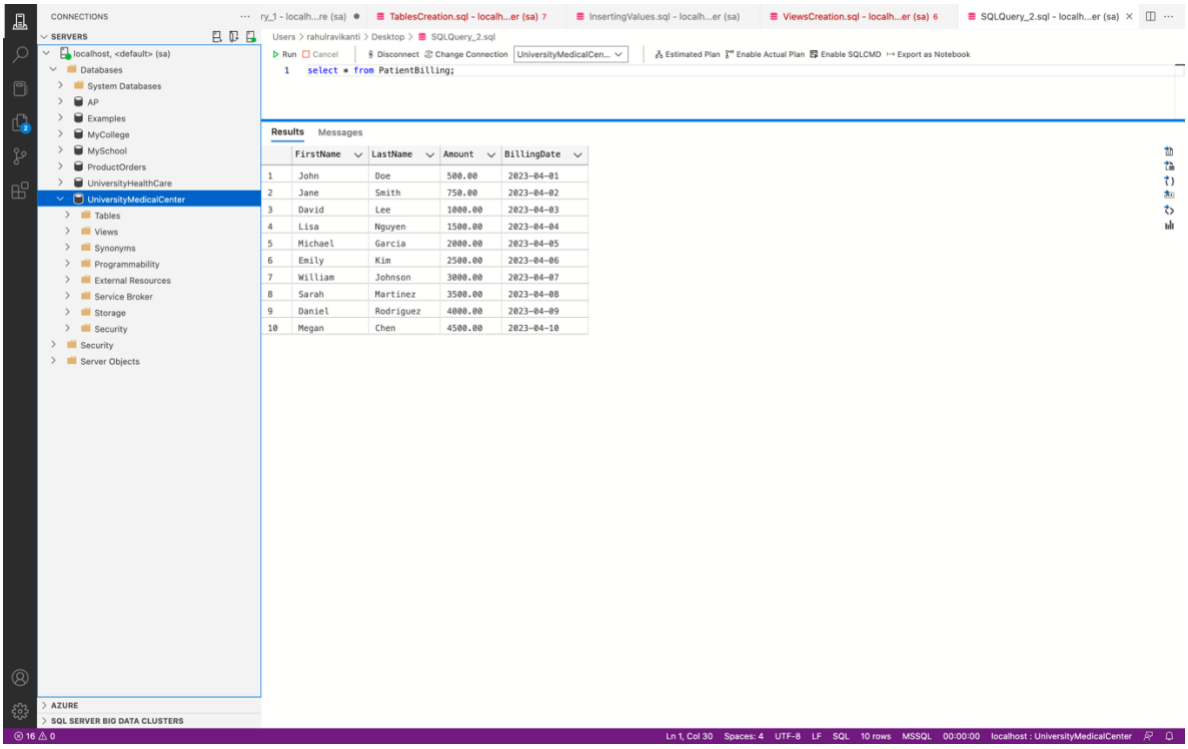


Code for creating views

View that shows all patients and their corresponding medications:

```
CREATE VIEW PatientMedications AS
SELECT p.FirstName, p.LastName, m.MedicationName, m.Dosage, m.Frequency
FROM Patients p
INNER JOIN Medications m ON p.PatientID = m.PatientID;
```

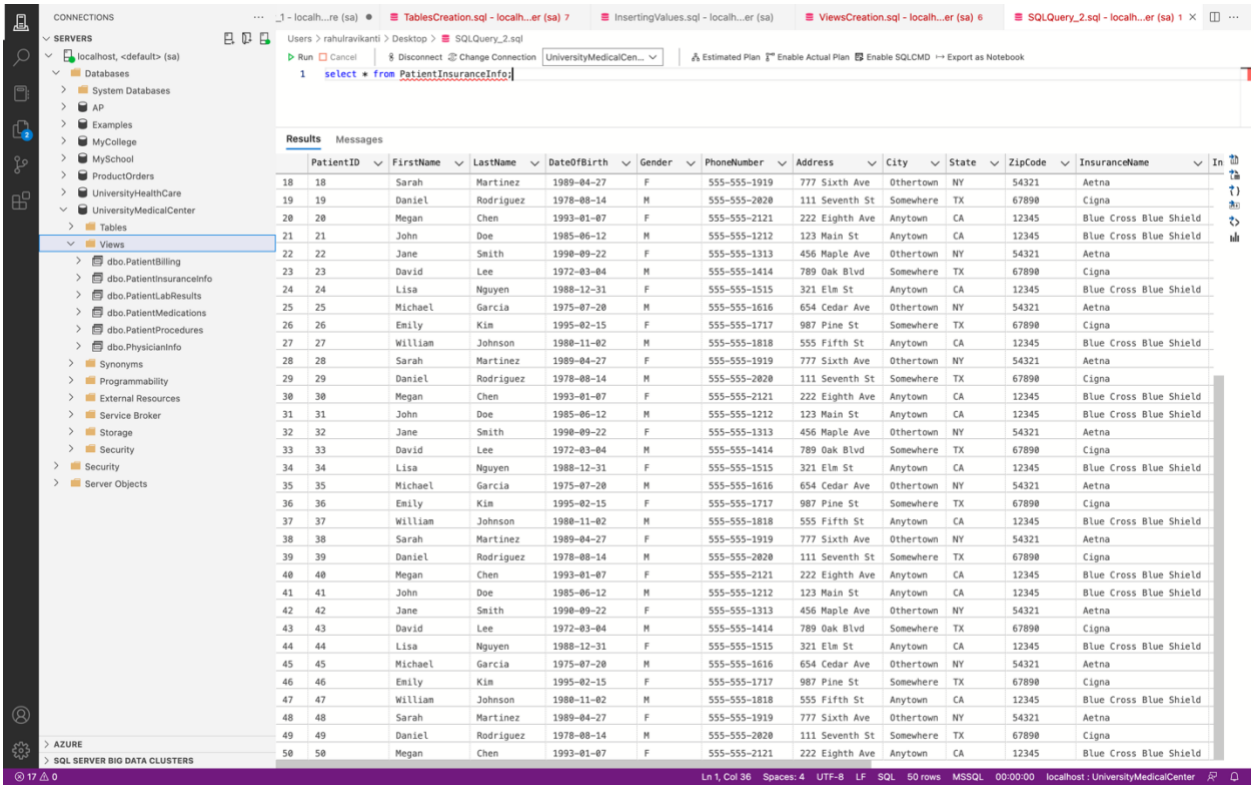
```
select * from PatientBilling;
```



Retrieved 10 rows.

View that shows all patients and their corresponding billing information:

```
CREATE VIEW PatientBilling AS
SELECT p.FirstName, p.LastName, b.Amount, b.BillingDate
FROM Patients p
INNER JOIN Billing b ON p.PatientID = b.PatientID;

select * from PatientInsuranceInfo;
```

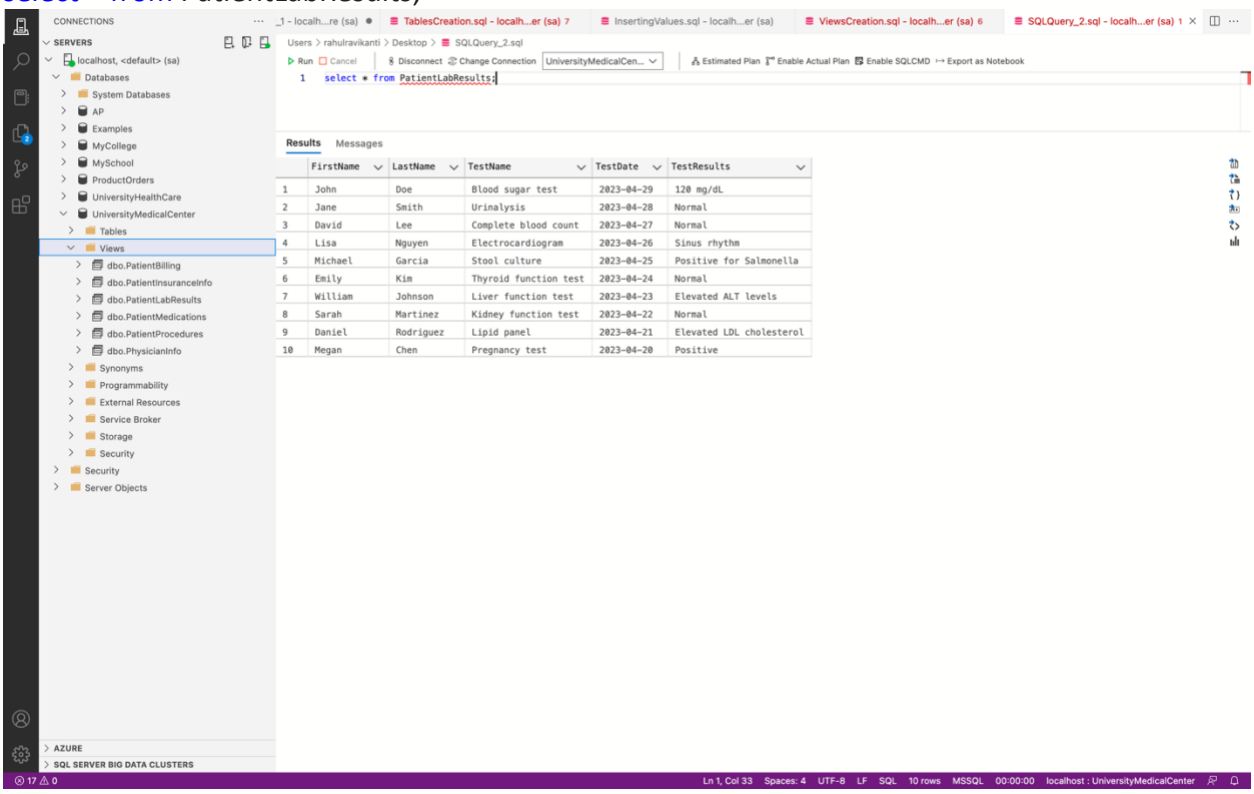


Retrieved 50 rows.

View that shows all patients and their corresponding lab results:

```
CREATE VIEW PatientLabResults AS
SELECT p.FirstName, p.LastName, l.TestName, l.TestDate, l.TestResults
FROM Patients p
INNER JOIN LabResults l ON p.PatientID = l.PatientID;
```

```
select * from PatientLabResults;
```

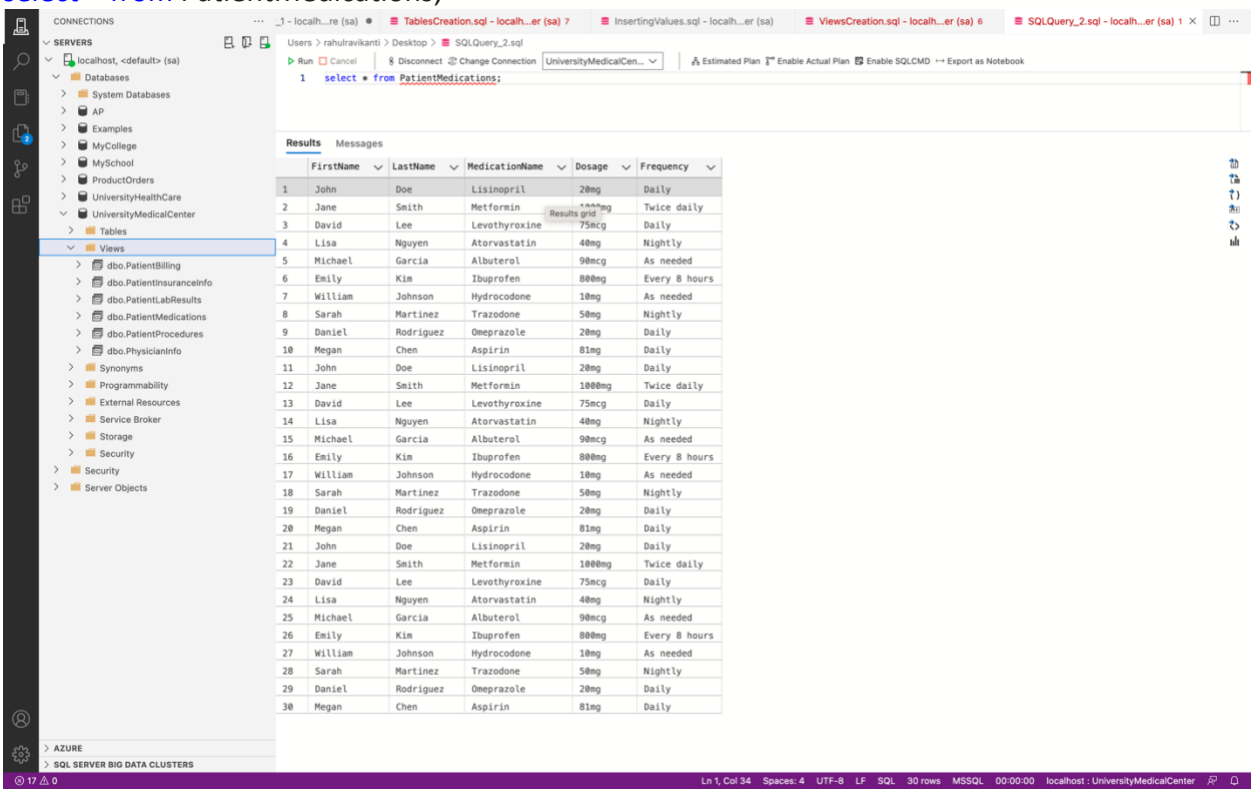


Retrieved 10 rows.

View that shows all patients and their corresponding procedures:

```
CREATE VIEW PatientProcedures AS
SELECT p.FirstName, p.LastName, pr.ICD10CodeID, pr.ProcedureDate
FROM Patients p
INNER JOIN PatientPhysician pp ON p.PatientID = pp.PatientID
INNER JOIN Procedures pr ON pp.PatientPhysicianID = pr.PatientPhysicianID;
```

```
select * from PatientMedications;
```



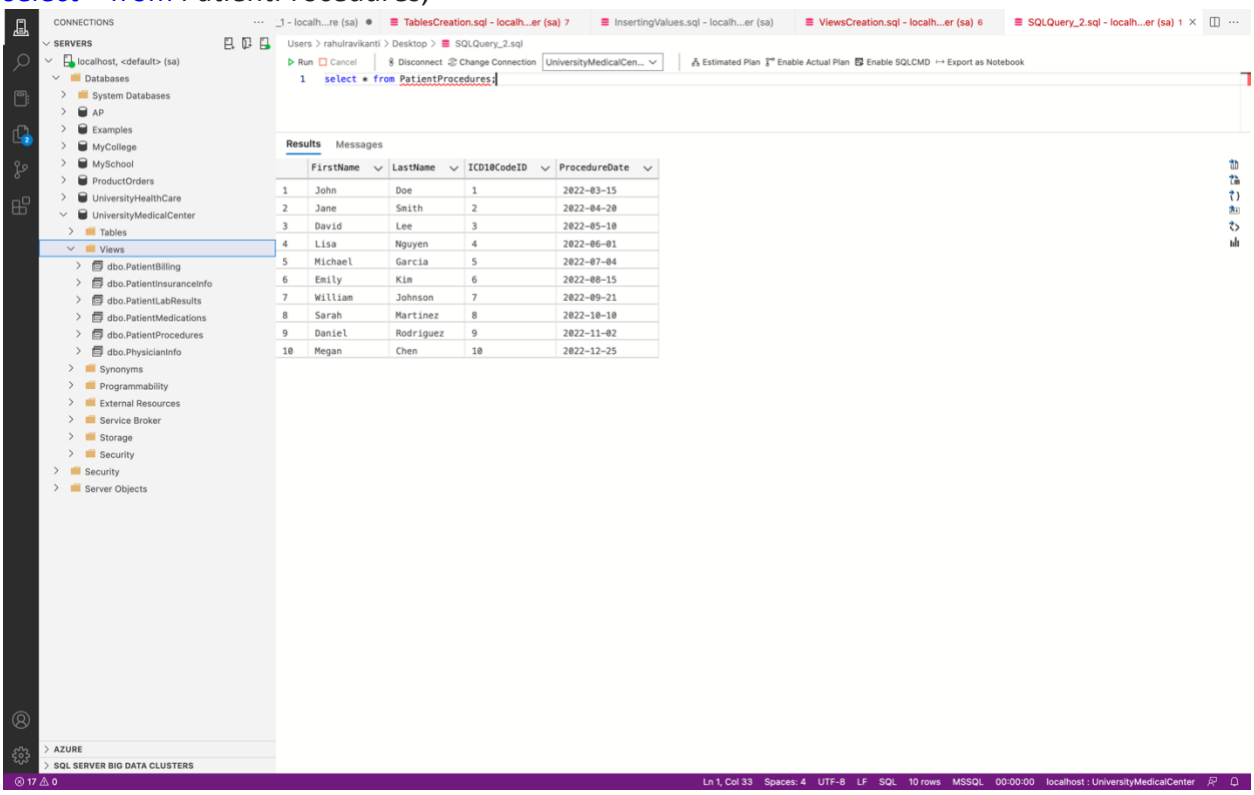
	FirstName	LastName	MedicationName	Dosage	Frequency
1	John	Doe	Lisinopril	20mg	Daily
2	Jane	Smith	Metformin	1000mg	Twice daily
3	David	Lee	Levothyroxine	75mcg	Daily
4	Lisa	Nguyen	Atorvastatin	40mg	Nightly
5	Michael	Garcia	Albuterol	90mcg	As needed
6	Emily	Kim	Ibuprofen	800mg	Every 8 hours
7	William	Johnson	Hydrocodone	10mg	As needed
8	Sarah	Martinez	Trazodone	50mg	Nightly
9	Daniel	Rodriguez	Omeprazole	20mg	Daily
10	Megan	Chen	Aspirin	81mg	Daily
11	John	Doe	Lisinopril	20mg	Daily
12	Jane	Smith	Metformin	1000mg	Twice daily
13	David	Lee	Levothyroxine	75mcg	Daily
14	Lisa	Nguyen	Atorvastatin	40mg	Nightly
15	Michael	Garcia	Albuterol	90mcg	As needed
16	Emily	Kim	Ibuprofen	800mg	Every 8 hours
17	William	Johnson	Hydrocodone	10mg	As needed
18	Sarah	Martinez	Trazodone	50mg	Nightly
19	Daniel	Rodriguez	Omeprazole	20mg	Daily
20	Megan	Chen	Aspirin	81mg	Daily
21	John	Doe	Lisinopril	20mg	Daily
22	Jane	Smith	Metformin	1000mg	Twice daily
23	David	Lee	Levothyroxine	75mcg	Daily
24	Lisa	Nguyen	Atorvastatin	40mg	Nightly
25	Michael	Garcia	Albuterol	90mcg	As needed
26	Emily	Kim	Ibuprofen	800mg	Every 8 hours
27	William	Johnson	Hydrocodone	10mg	As needed
28	Sarah	Martinez	Trazodone	50mg	Nightly
29	Daniel	Rodriguez	Omeprazole	20mg	Daily
30	Megan	Chen	Aspirin	81mg	Daily

Retrieved 30 rows.

View that displays patient information along with their insurance information:

```
CREATE VIEW PatientInsuranceInfo AS
SELECT p.PatientID, p.FirstName, p.LastName, p.DateOfBirth, p.Gender, p.PhoneNumber,
p.Address, p.City, p.State, p.ZipCode, i.InsuranceName, i.InsuranceNumber,
i.InsurancePlanType, i.InsuranceEffectiveDate, i.InsuranceExpirationDate
FROM Patients p
JOIN Insurances i ON p.InsuranceID = i.InsuranceID;
```

```
select * from PatientProcedures;
```

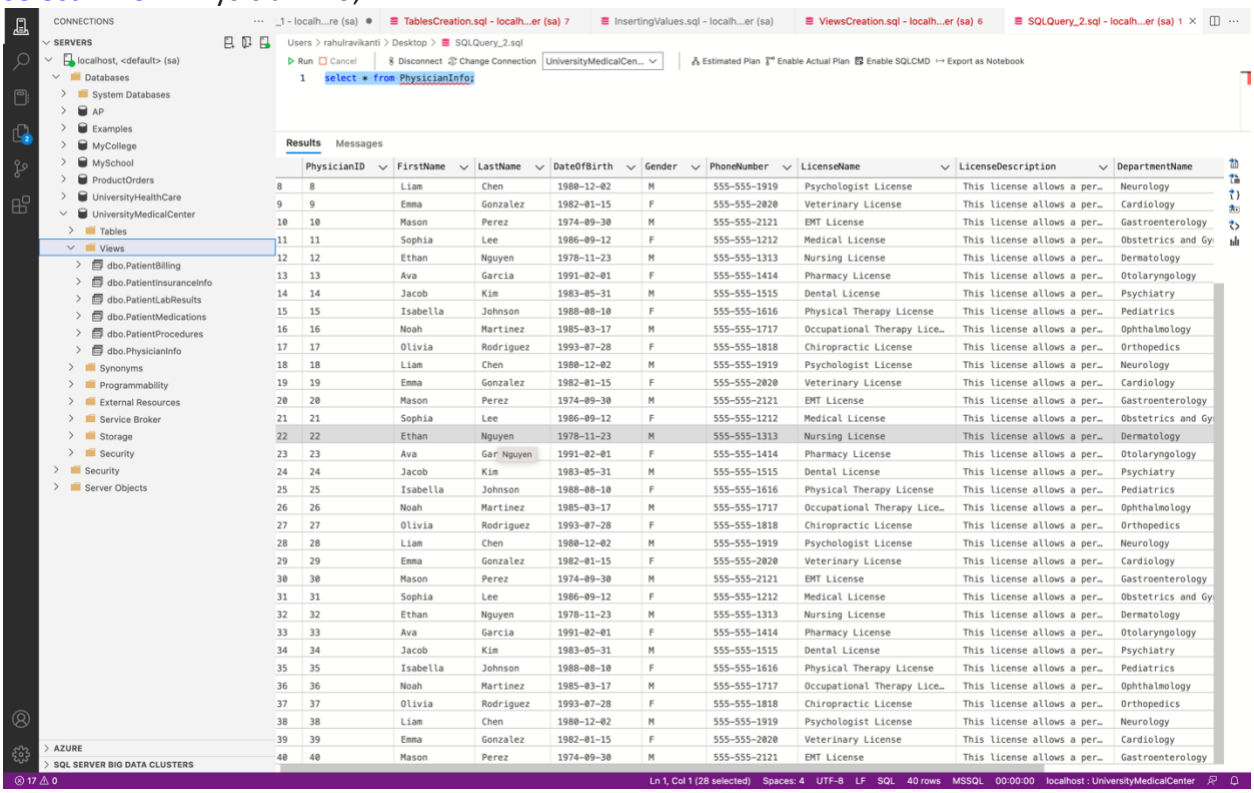


Retrieved 10 rows.

View that displays physician information along with their department and license information:

```
CREATE VIEW PhysicianInfo AS
SELECT phy.PhysicianID, phy.FirstName, phy.LastName, phy.DateOfBirth, phy.Gender,
phy.PhoneNumber, l.LicenseName, l.LicenseDescription, d.DepartmentName,
d.DepartmentDescription
FROM Physicians phy
JOIN LegalLicenses l ON phy.LicenseID = l.LicenseID
JOIN HealthDepartments d ON phy.DepartmentID = d.DepartmentID;
```

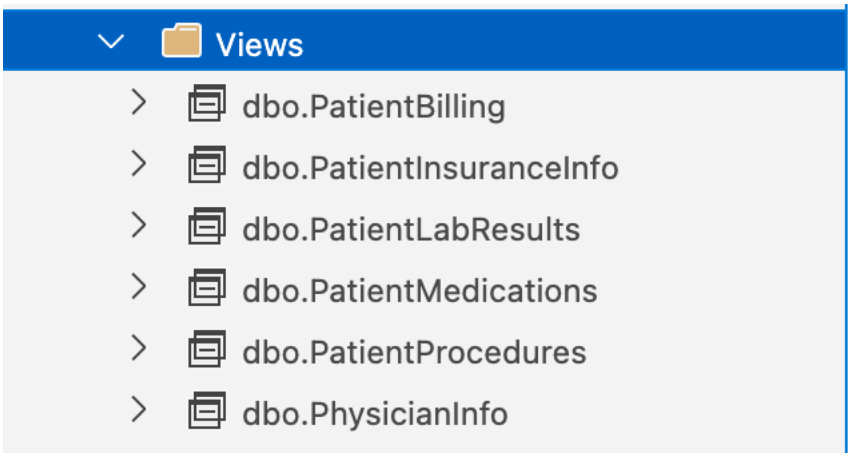
```
select * from PhysicianInfo;
```



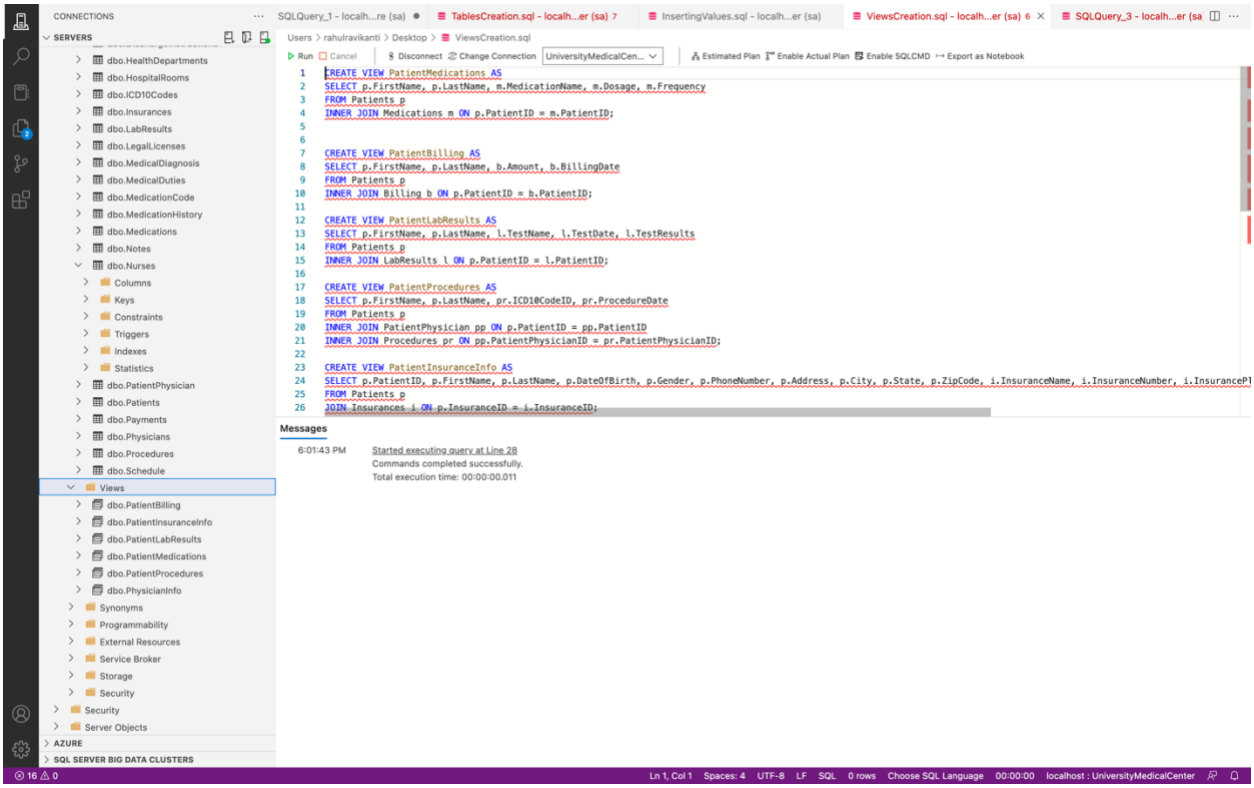
	PhysicianID	FirstName	LastName	DateOfBirth	Gender	PhoneNumber	LicenseName	LicenseDescription	DepartmentName
8	8	Liam	Chen	1980-12-02	M	555-555-1919	Psychologist License	This license allows a per…	Neurology
9	9	Emma	Gonzalez	1982-01-15	F	555-555-2020	Veterinary License	This license allows a per…	Cardiology
10	10	Mason	Perez	1974-09-30	M	555-555-2121	EMT License	This license allows a per…	Gastroenterology
11	11	Sophia	Lee	1986-09-12	F	555-555-1212	Medical License	This license allows a per…	Obstetrics and Gy
12	12	Ethan	Nguyen	1978-11-23	M	555-555-1313	Nursing License	This license allows a per…	Dermatology
13	13	Ava	Garcia	1991-02-01	F	555-555-1414	Pharmacy License	This license allows a per…	Otolaryngology
14	14	Jacob	Kim	1983-05-31	M	555-555-1515	Dental License	This license allows a per…	Psychiatry
15	15	Isabella	Johnson	1988-08-10	F	555-555-1616	Physical Therapy License	This license allows a per…	Pediatrics
16	16	Noah	Martinez	1985-03-17	M	555-555-1717	Occupational Therapy Lice…	This license allows a per…	Ophthalmology
17	17	Olivia	Rodriguez	1993-07-28	F	555-555-1818	Chiropractic License	This license allows a per…	Orthopedics
18	18	Liam	Chen	1980-12-02	M	555-555-1919	Psychologist License	This license allows a per…	Neurology
19	19	Emma	Gonzalez	1982-01-15	F	555-555-2020	Veterinary License	This license allows a per…	Cardiology
20	20	Mason	Perez	1974-09-30	M	555-555-2121	EMT License	This license allows a per…	Gastroenterology
21	21	Sophia	Lee	1986-09-12	F	555-555-1212	Medical License	This license allows a per…	Obstetrics and Gy
22	22	Ethan	Nguyen	1978-11-23	M	555-555-1313	Nursing License	This license allows a per…	Dermatology
23	23	Ava	Gar Nguyen	1991-02-01	F	555-555-1414	Pharmacy License	This license allows a per…	Otolaryngology
24	24	Jacob	Kim	1983-05-31	M	555-555-1515	Dental License	This license allows a per…	Psychiatry
25	25	Isabella	Johnson	1988-08-10	F	555-555-1616	Physical Therapy License	This license allows a per…	Pediatrics
26	26	Noah	Martinez	1985-03-17	M	555-555-1717	Occupational Therapy Lice…	This license allows a per…	Ophthalmology
27	27	Olivia	Rodriguez	1993-07-28	F	555-555-1818	Chiropractic License	This license allows a per…	Orthopedics
28	28	Liam	Chen	1980-12-02	M	555-555-1919	Psychologist License	This license allows a per…	Neurology
29	29	Emma	Gonzalez	1982-01-15	F	555-555-2020	Veterinary License	This license allows a per…	Cardiology
30	30	Mason	Perez	1974-09-30	M	555-555-2121	EMT License	This license allows a per…	Gastroenterology
31	31	Sophia	Lee	1986-09-12	F	555-555-1212	Medical License	This license allows a per…	Obstetrics and Gy
32	32	Ethan	Nguyen	1978-11-23	M	555-555-1313	Nursing License	This license allows a per…	Dermatology
33	33	Ava	Garcia	1991-02-01	F	555-555-1414	Pharmacy License	This license allows a per…	Otolaryngology
34	34	Jacob	Kim	1983-05-31	M	555-555-1515	Dental License	This license allows a per…	Psychiatry
35	35	Isabella	Johnson	1988-08-10	F	555-555-1616	Physical Therapy License	This license allows a per…	Pediatrics
36	36	Noah	Martinez	1985-03-17	M	555-555-1717	Occupational Therapy Lice…	This license allows a per…	Ophthalmology
37	37	Olivia	Rodriguez	1993-07-28	F	555-555-1818	Chiropractic License	This license allows a per…	Orthopedics
38	38	Liam	Chen	1980-12-02	M	555-555-1919	Psychologist License	This license allows a per…	Neurology
39	39	Emma	Gonzalez	1982-01-15	F	555-555-2020	Veterinary License	This license allows a per…	Cardiology
40	40	Mason	Perez	1974-09-30	M	555-555-2121	EMT License	This license allows a per…	Gastroenterology

Retrieved 40 rows.

Views created after successful execution of above code.



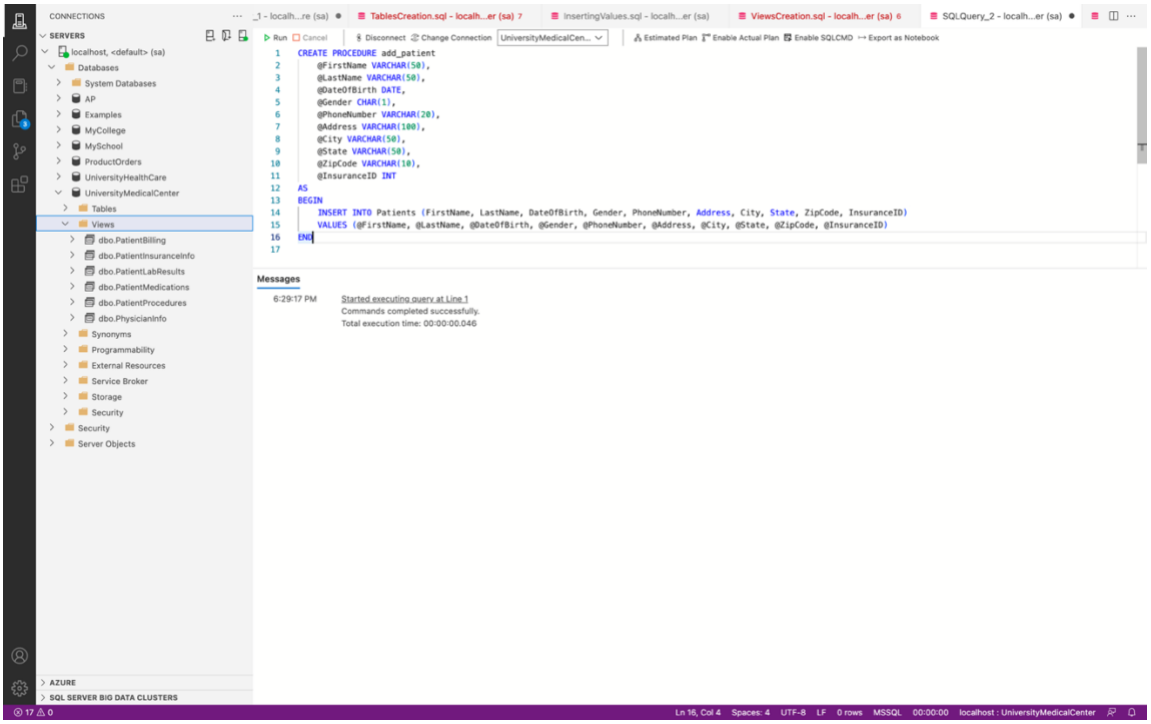
Execution of Views



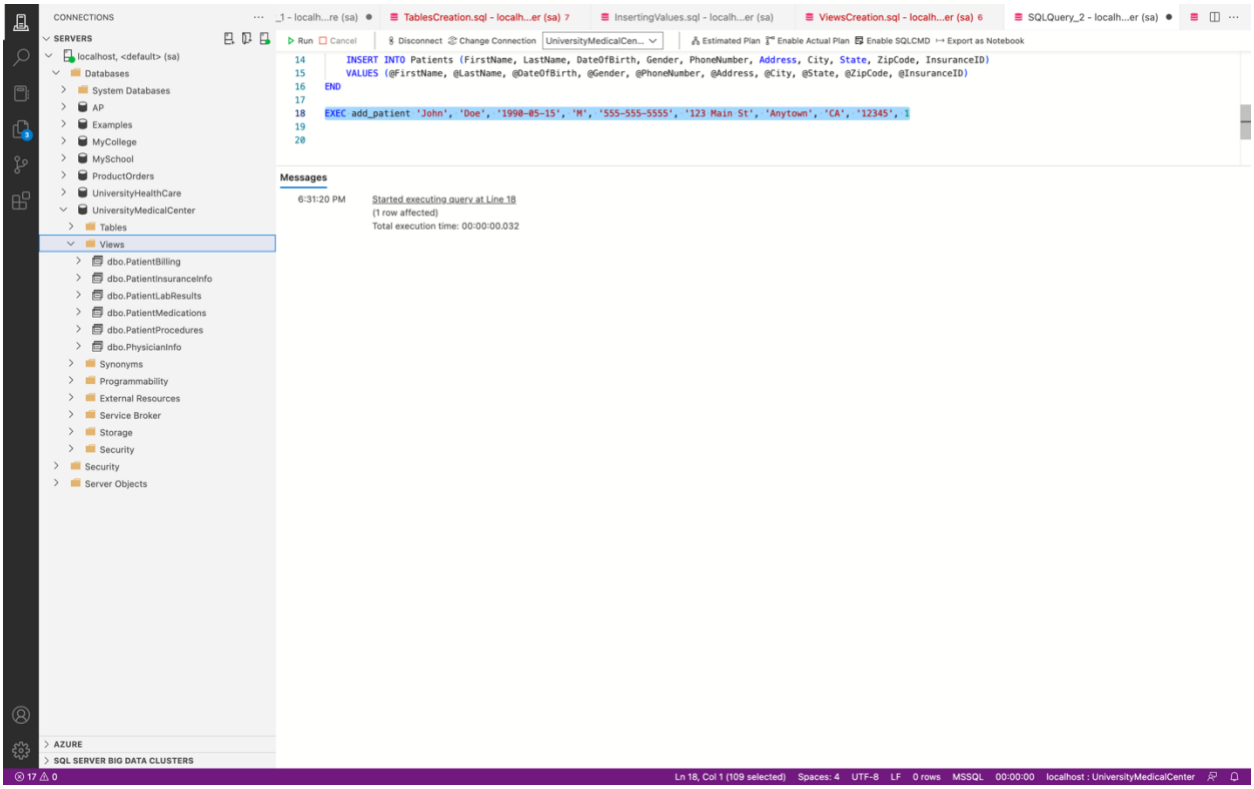
Stored Procedures:

1. Stored Procedure for adding a new patient

```
CREATE PROCEDURE add_patient
  @FirstName VARCHAR(50),
  @LastName VARCHAR(50),
  @DateOfBirth DATE,
  @Gender CHAR(1),
  @PhoneNumber VARCHAR(20),
  @Address VARCHAR(100),
  @City VARCHAR(50),
  @State VARCHAR(50),
  @ZipCode VARCHAR(10),
  @InsuranceID INT
AS
BEGIN
  INSERT INTO Patients (FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Address,
City, State, ZipCode, InsuranceID)
  VALUES (@FirstName, @LastName, @DateOfBirth, @Gender, @PhoneNumber, @Address,
@City, @State, @ZipCode, @InsuranceID)
END
```



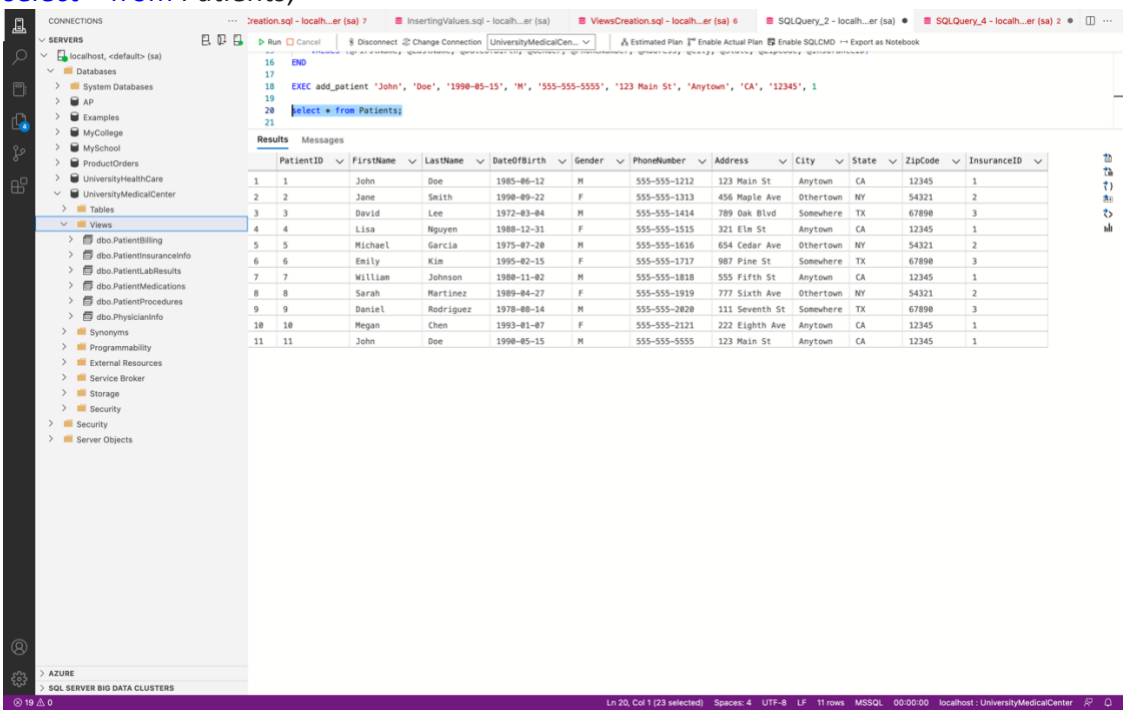
EXEC add_patient 'John', 'Doe', '1990-05-15', 'M', '555-555-5555', '123 Main St', 'Anytown', 'CA', '12345', 1



Inserted a patient record using the stored procedure

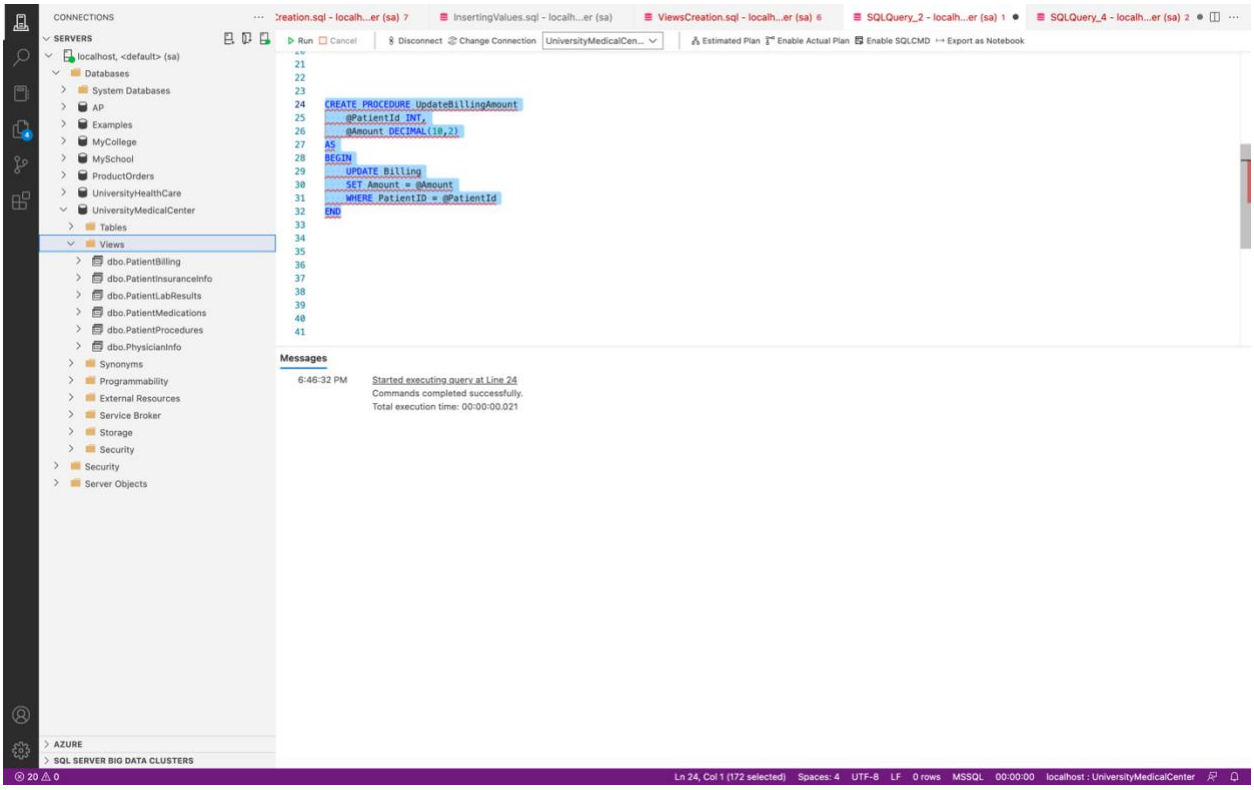
Added 1 row (new patient using the stored procedure)

```
select * from Patients;
```

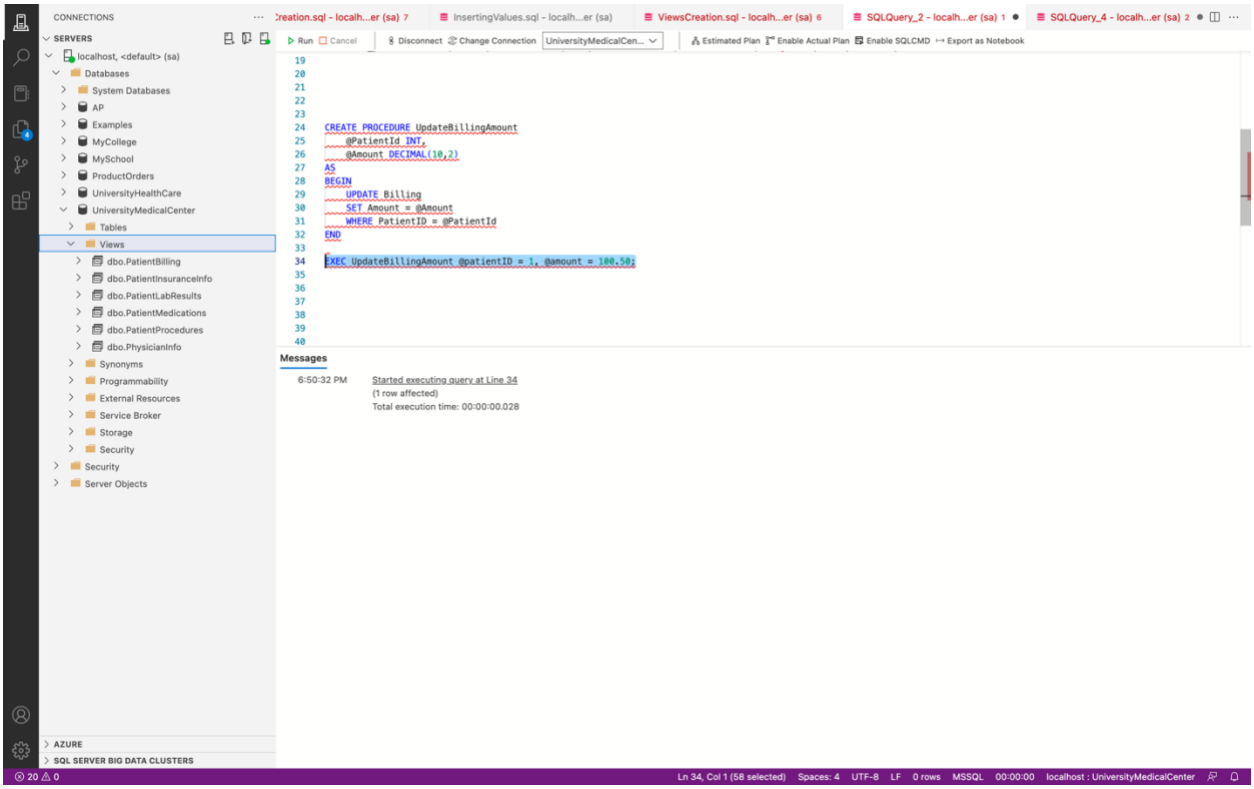


2. Stored Procedure for updating billing amount of a patient

```
CREATE PROCEDURE UpdateBillingAmount
   @PatientId INT,
   @Amount DECIMAL(10,2)
AS
BEGIN
   UPDATE Billing
   SET Amount = @Amount
   WHERE PatientID = @PatientId
END
```



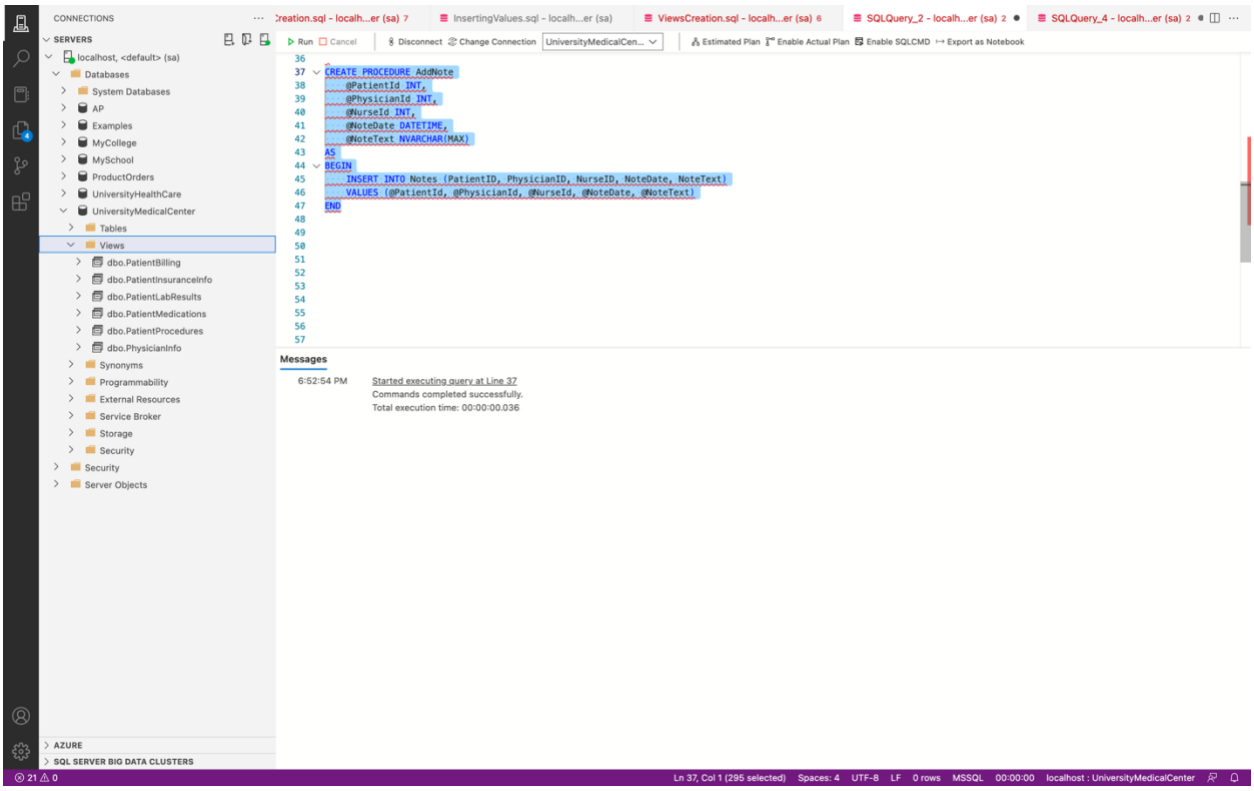

```
EXEC UpdateBillingAmount @patientID = 1, @amount = 100.50;
```



1 Row affected (updated billing amount of patient with PatientID 1).

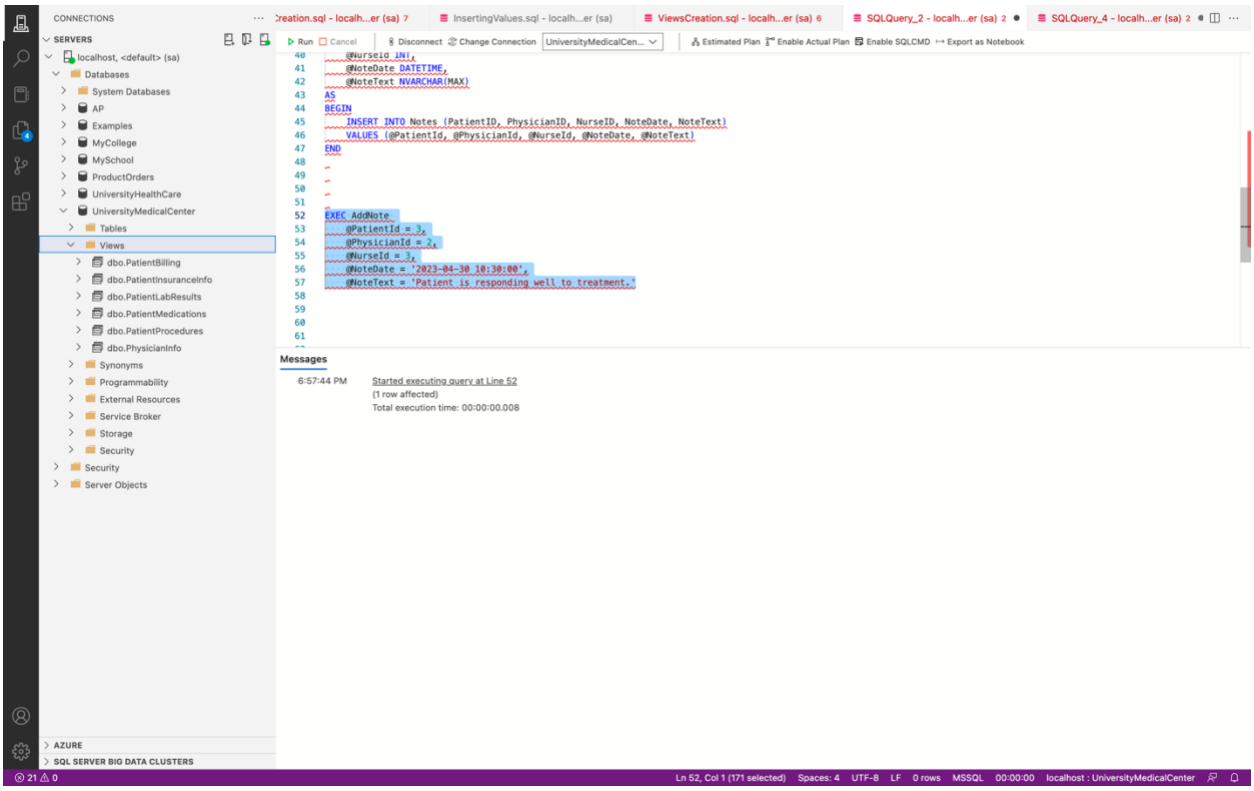
3. Stored procedure for adding a note

```
CREATE PROCEDURE AddNote
   @PatientId INT,
   @PhysicianId INT,
   @NurseId INT,
   @NoteDate DATETIME,
   @NoteText NVARCHAR(MAX)
AS
BEGIN
   INSERT INTO Notes (PatientID, PhysicianID, NurseID, NoteDate, NoteText)
   VALUES (@PatientId, @PhysicianId, @NurseId, @NoteDate, @NoteText)
END
```




```
EXEC AddNote
   @PatientId = 3,
```

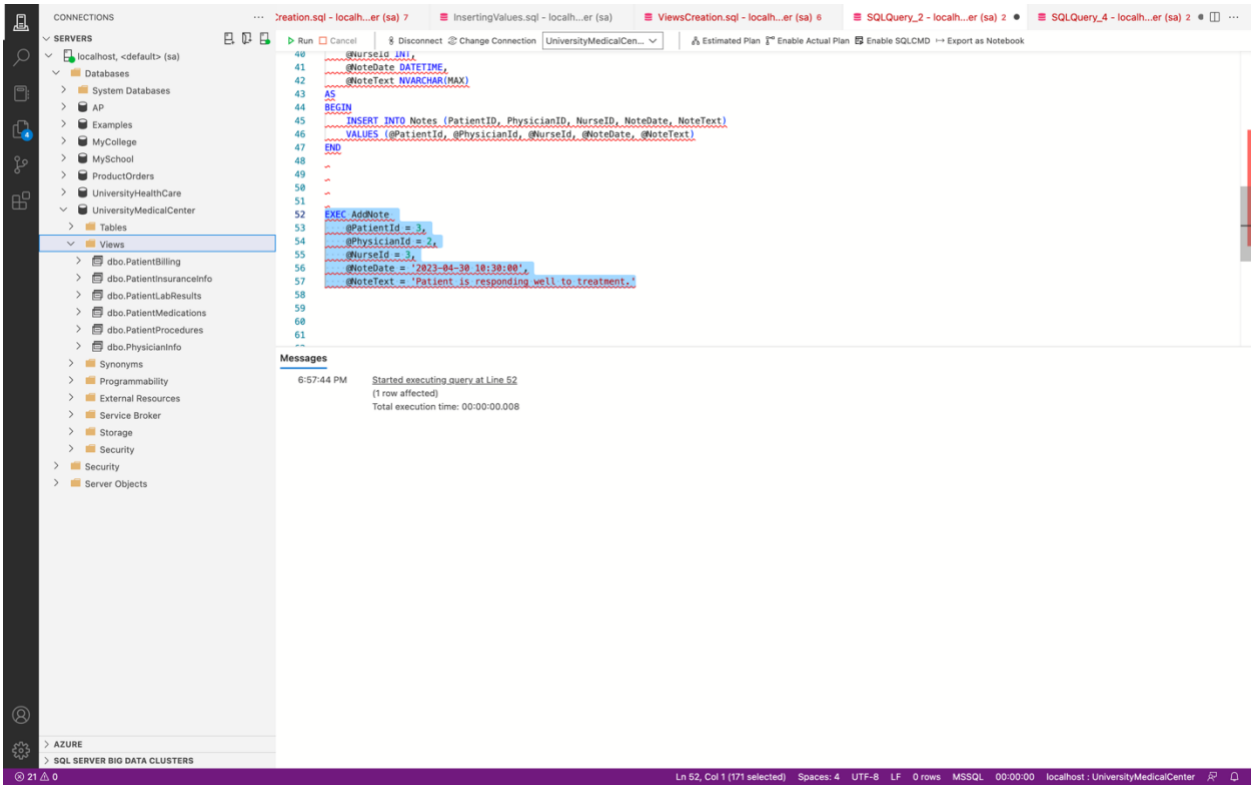
```
@PhysicianId = 2,
@NurseId = 3,
@NoteDate = '2023-04-30 10:30:00',
@NoteText = 'Patient is responding well to treatment.'
```



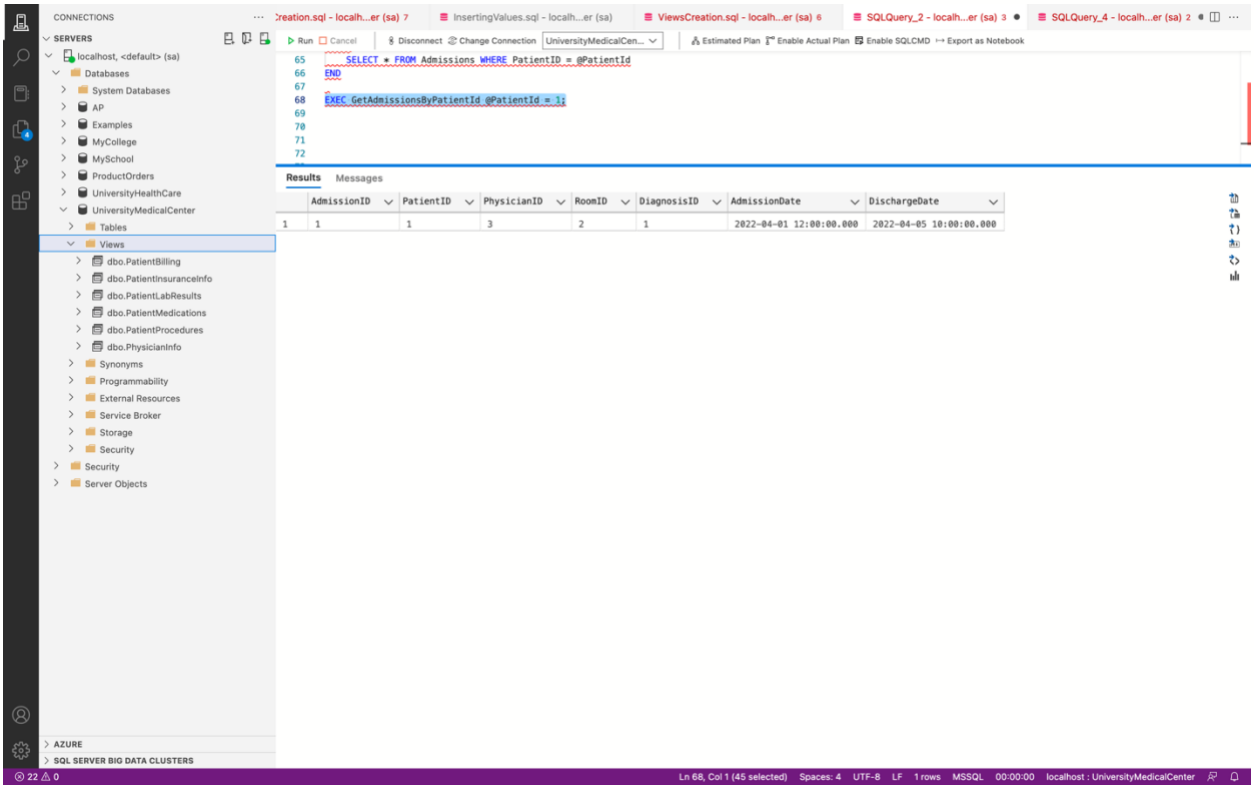
1 Row Affected (Added a note).

4. Stored procedure for getting all admissions for a specific patient.

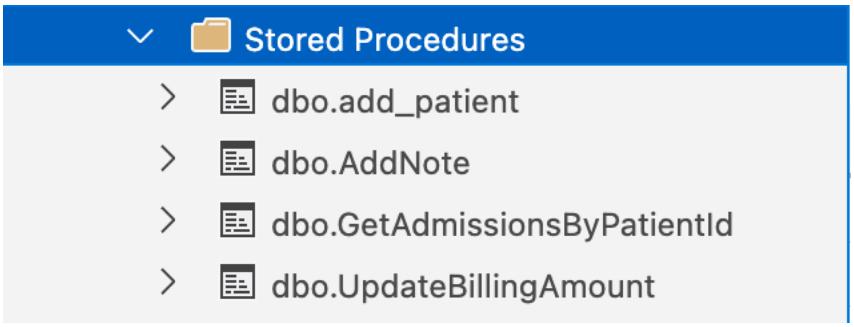
```
CREATE PROCEDURE GetAdmissionsByPatientId
   @PatientId INT
AS
BEGIN
   SELECT * FROM Admissions WHERE PatientID = @PatientId
END
```



```
EXEC GetAdmissionsByPatientId @PatientId = 1;
```



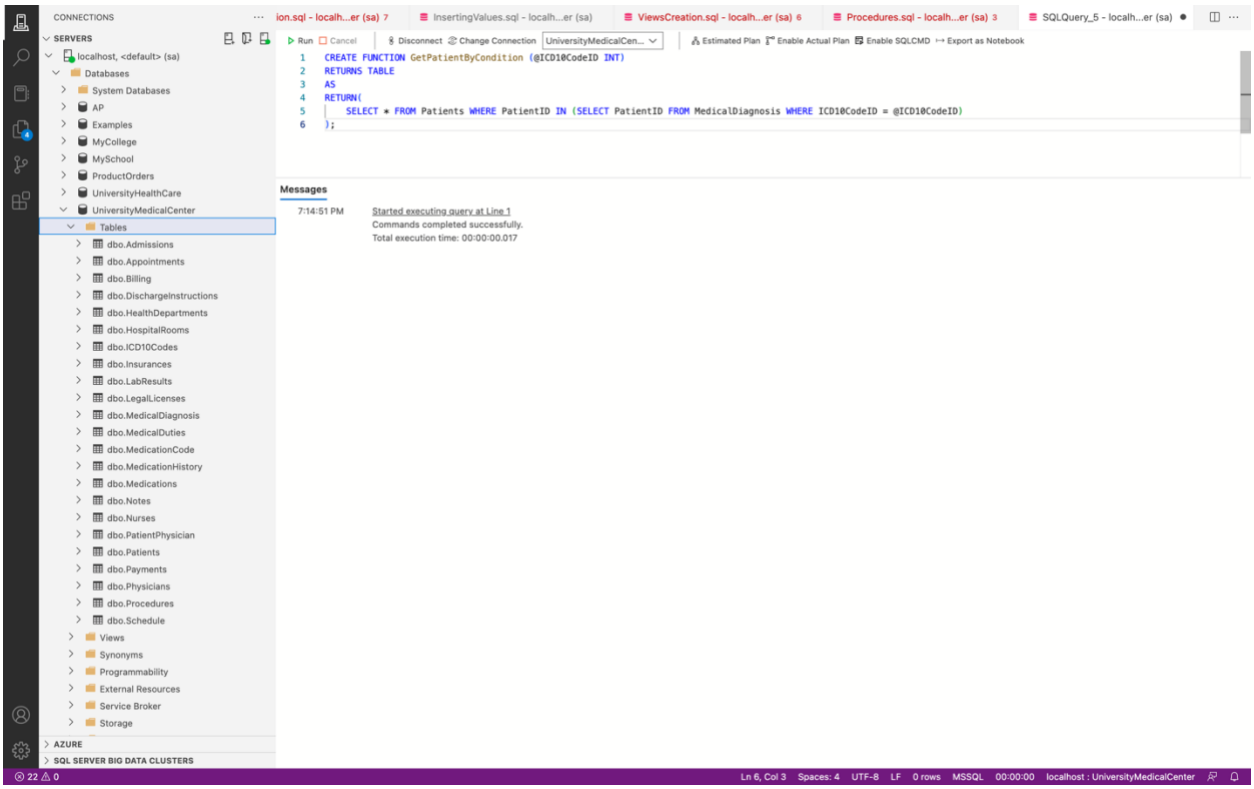
Created Stored Procedures



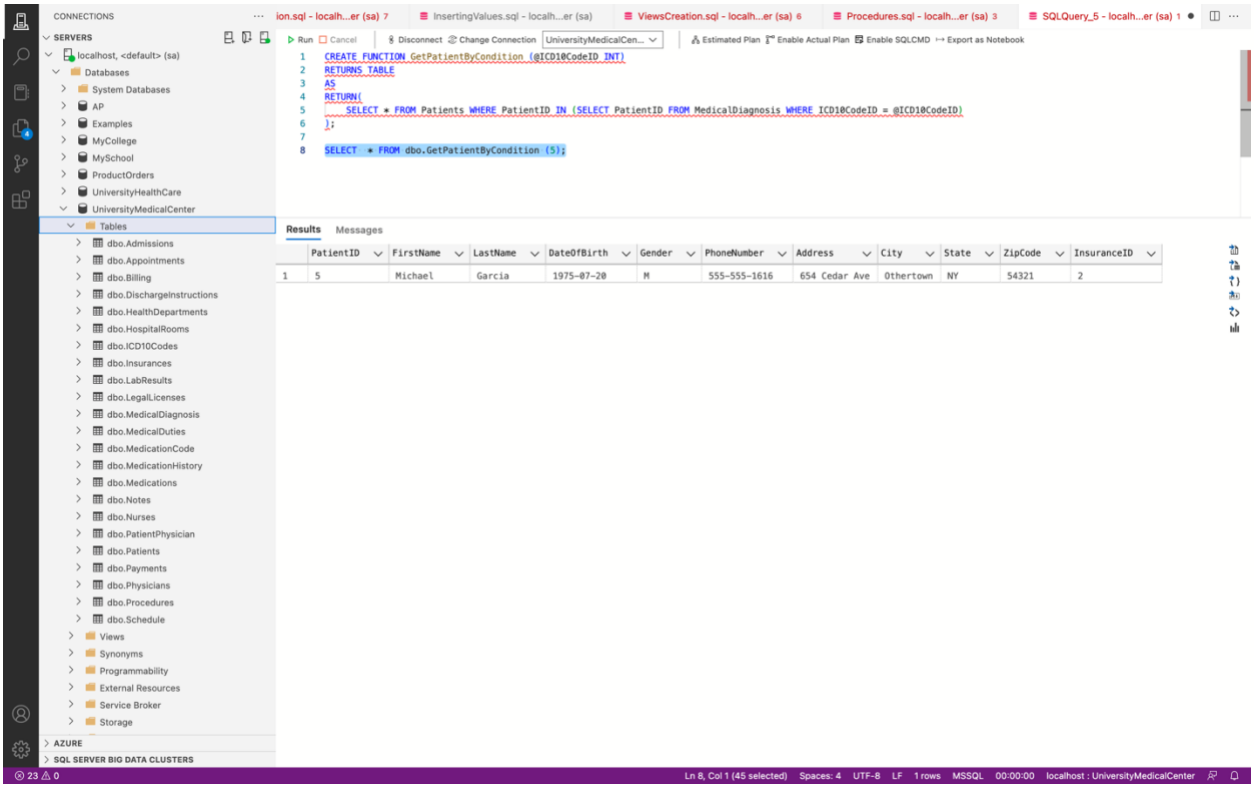
Functions

1. Function for getting patient details by condition.

```
CREATE FUNCTION GetPatientByCondition (@ICD10CodeID INT)
RETURNS TABLE
AS
RETURN(
   SELECT * FROM Patients WHERE PatientID IN (SELECT PatientID FROM MedicalDiagnosis
WHERE ICD10CodeID = @ICD10CodeID)
);
```



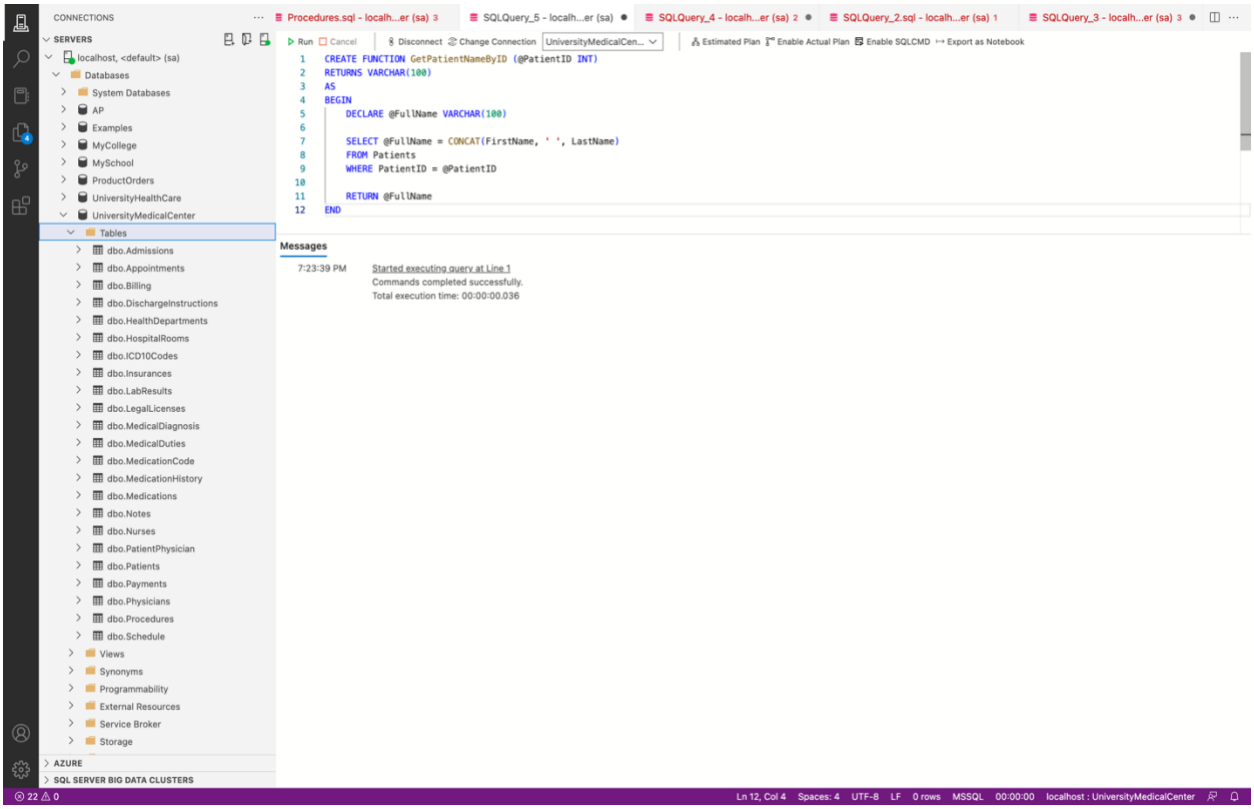
```
SELECT  * FROM dbo.GetPatientByCondition (5);
```



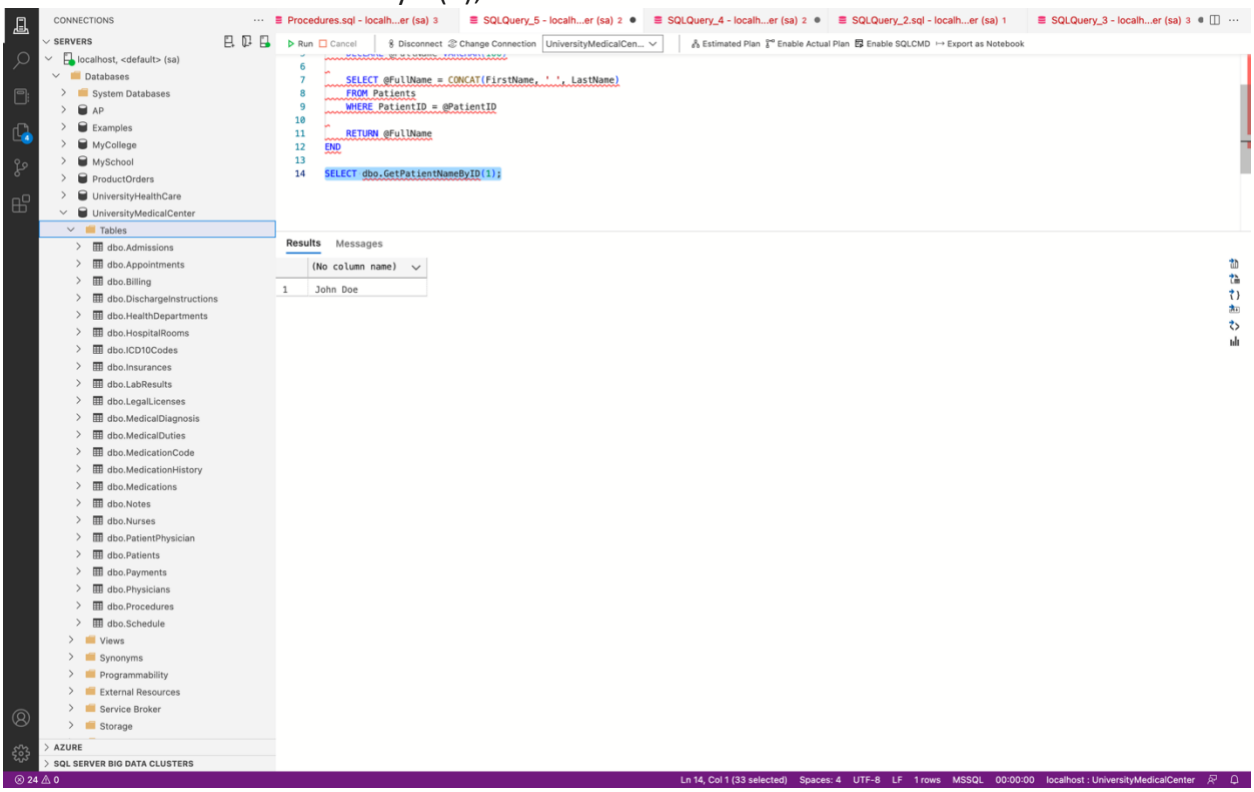
Retrieved 1 row.

2. Get patient full name by ID.

```
CREATE FUNCTION GetPatientNameByID (@PatientID INT)
RETURNS VARCHAR(100)
AS
BEGIN
   DECLARE @FullName VARCHAR(100)

   SELECT @FullName = CONCAT(FirstName, ' ', LastName)
   FROM Patients
   WHERE PatientID = @PatientID

   RETURN @FullName
END
```



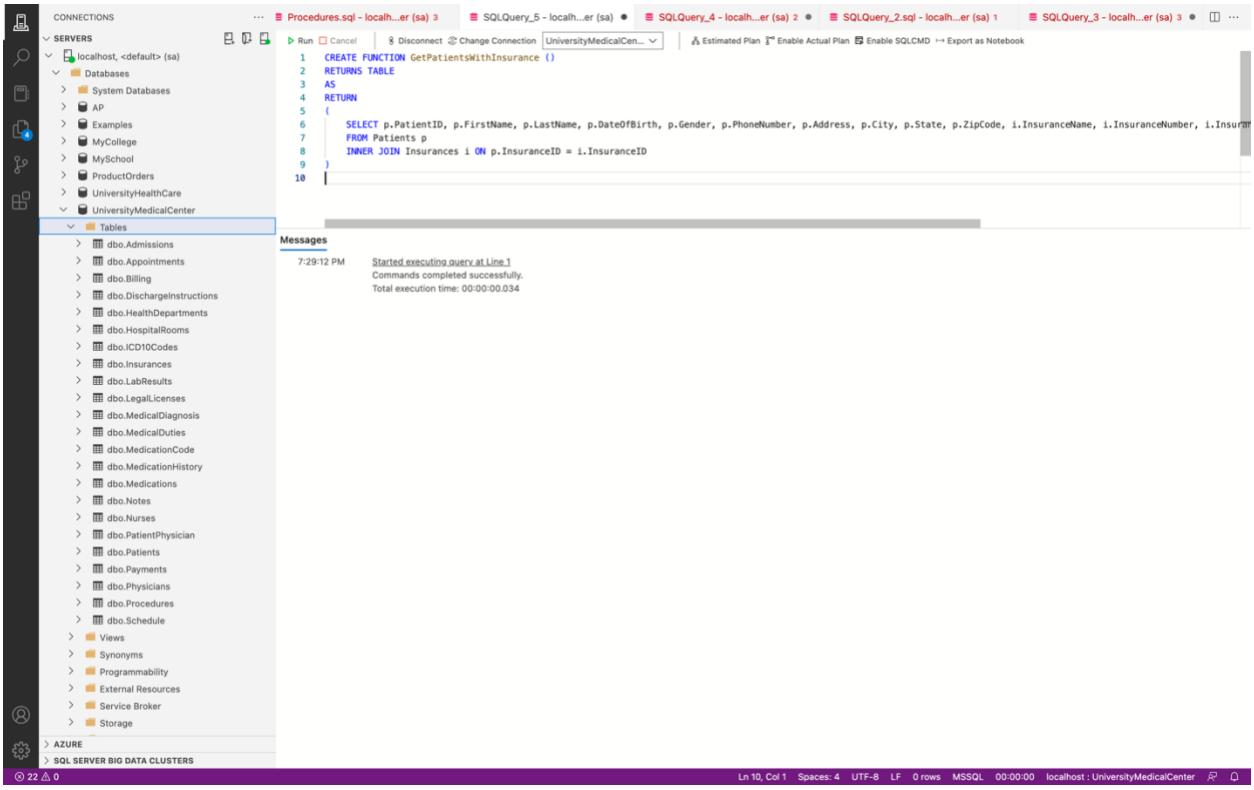
```
SELECT dbo.GetPatientNameByID(1);
```



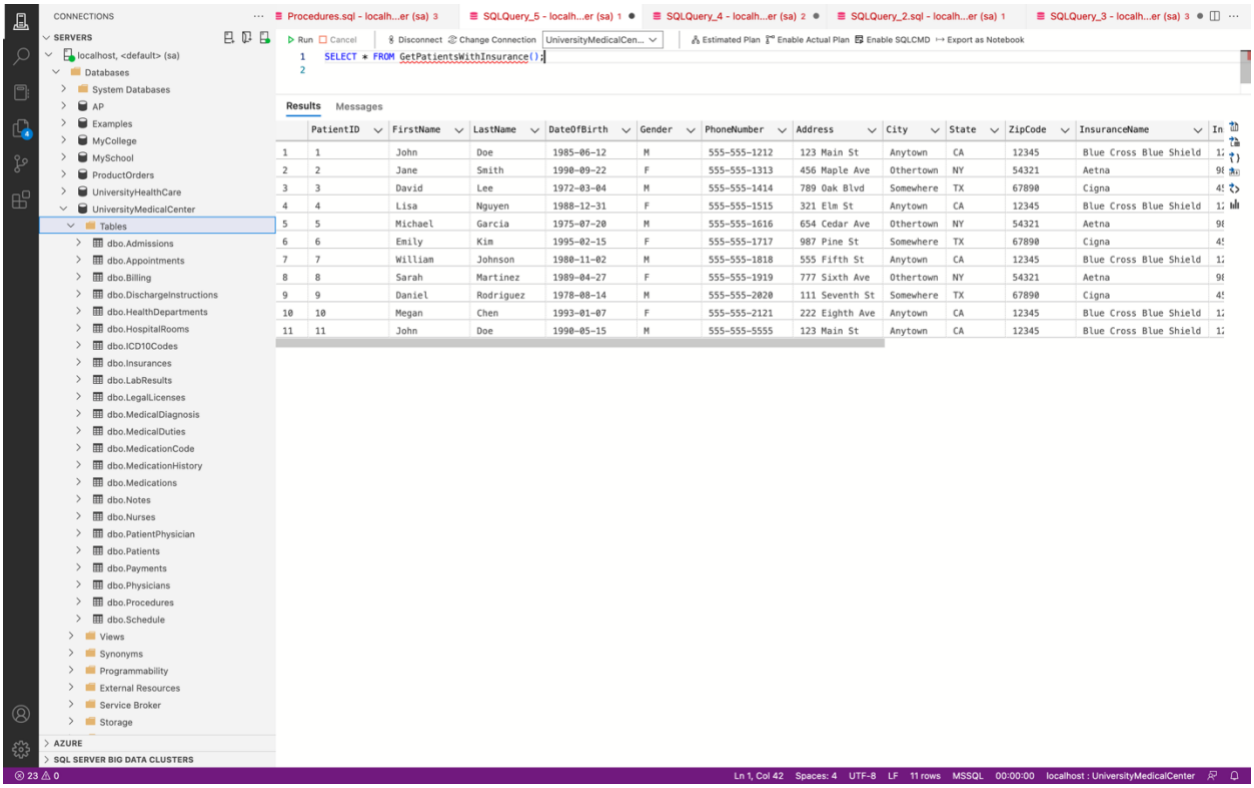
Retrieved 1 Row.

3. Function to get a list of all patients and their insurance information:

```
CREATE FUNCTION GetPatientsWithInsurance ()
RETURNS TABLE
AS
RETURN
(
   SELECT p.PatientID, p.FirstName, p.LastName, p.DateOfBirth, p.Gender, p.PhoneNumber,
p.Address, p.City, p.State, p.ZipCode, i.InsuranceName, i.InsuranceNumber,
i.InsurancePlanType, i.InsuranceEffectiveDate, i.InsuranceExpirationDate
   FROM Patients p
   INNER JOIN Insurances i ON p.InsuranceID = i.InsuranceID
)
```

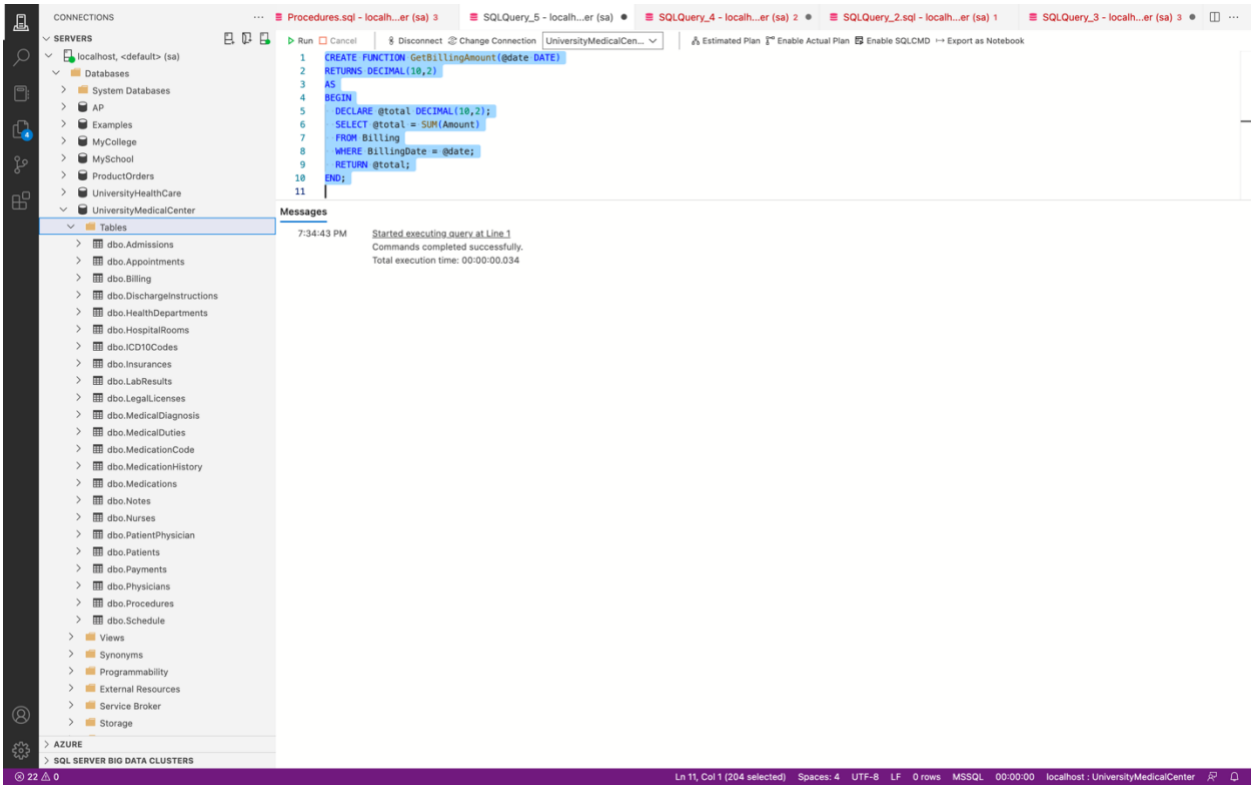


```
SELECT * FROM GetPatientsWithInsurance();
```

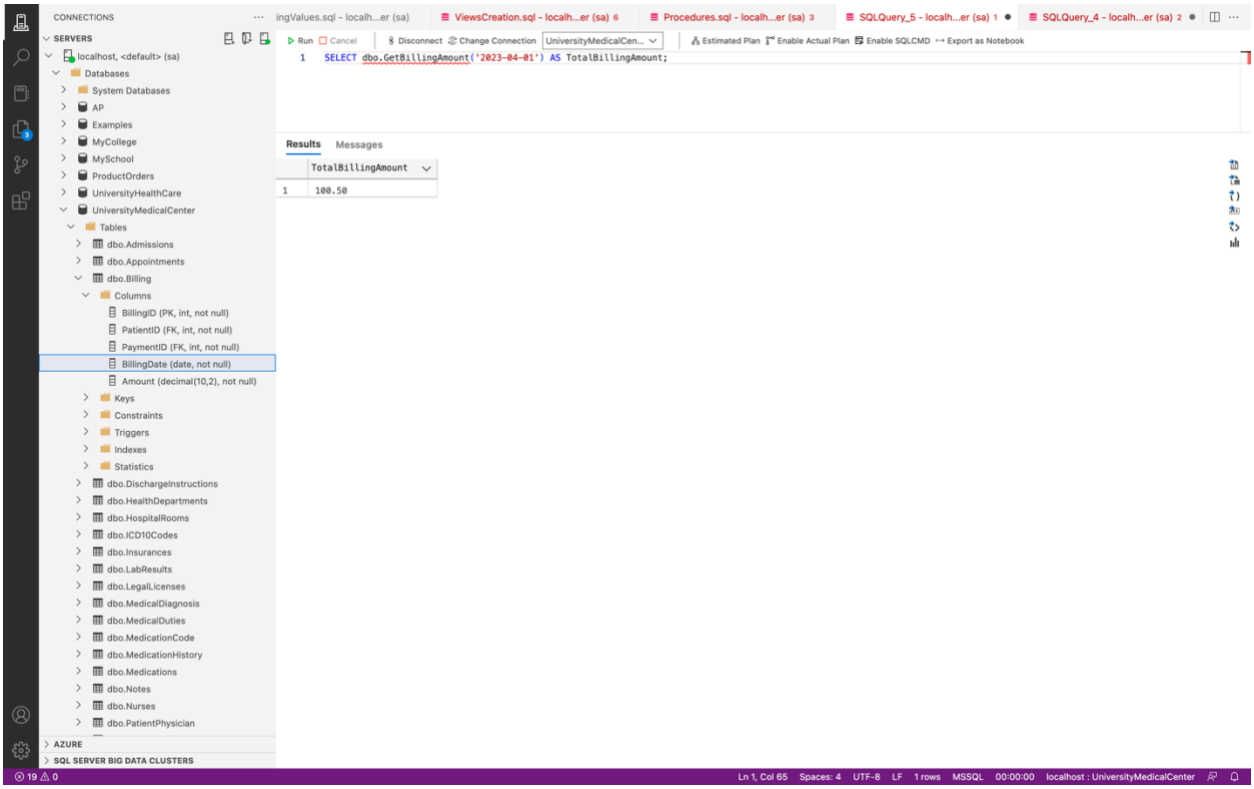


4. Function to get billing amount.

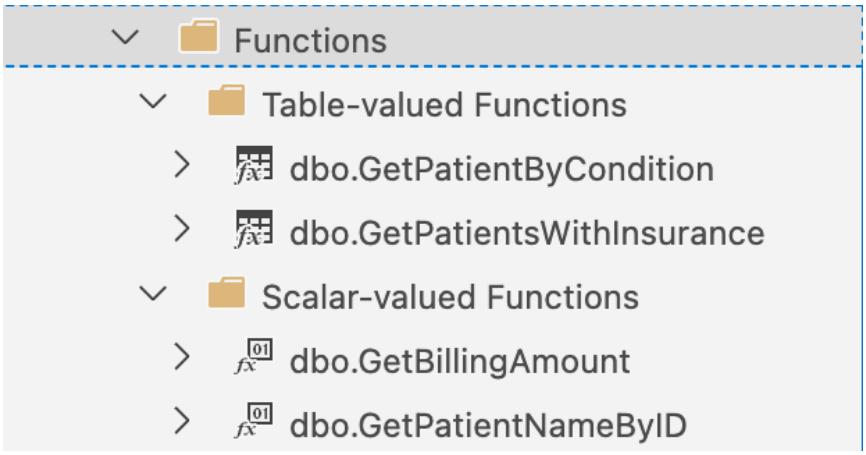
```
CREATE FUNCTION GetBillingAmount(@date DATE)
RETURNS DECIMAL(10,2)
AS
BEGIN
  DECLARE @total DECIMAL(10,2);
  SELECT @total = SUM(Amount)
  FROM Billing
  WHERE BillingDate = @date;
  RETURN @total;
END;
```



```
SELECT dbo.GetBillingAmount('2023-04-01') AS TotalBillingAmount;
```



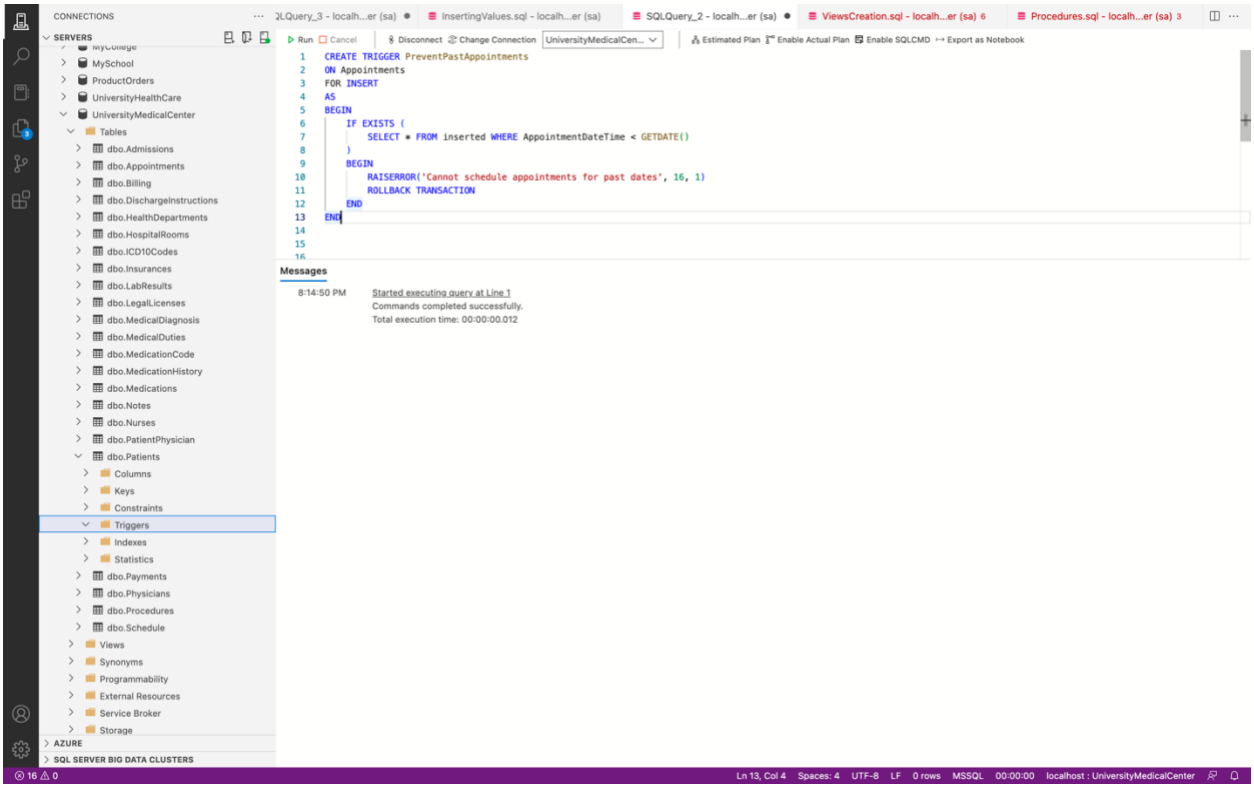
The created Functions



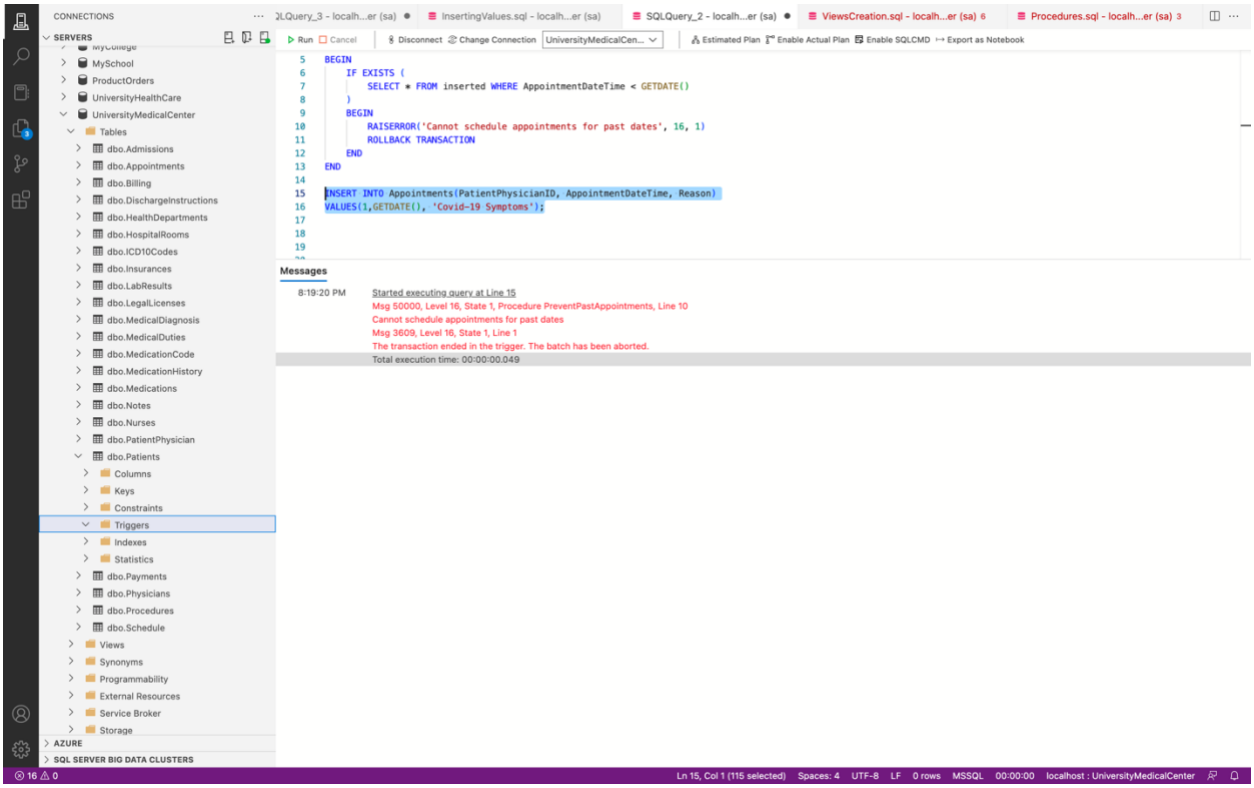
Triggers

1. Triggers for preventing past appointments.

```
CREATE TRIGGER PreventPastAppointments
ON Appointments
FOR INSERT
AS
BEGIN
   IF EXISTS (
      SELECT * FROM inserted WHERE AppointmentDateTime < GETDATE()
   )
   BEGIN
      RAISERROR('Cannot schedule appointments for past dates', 16, 1)
      ROLLBACK TRANSACTION
   END
END
```



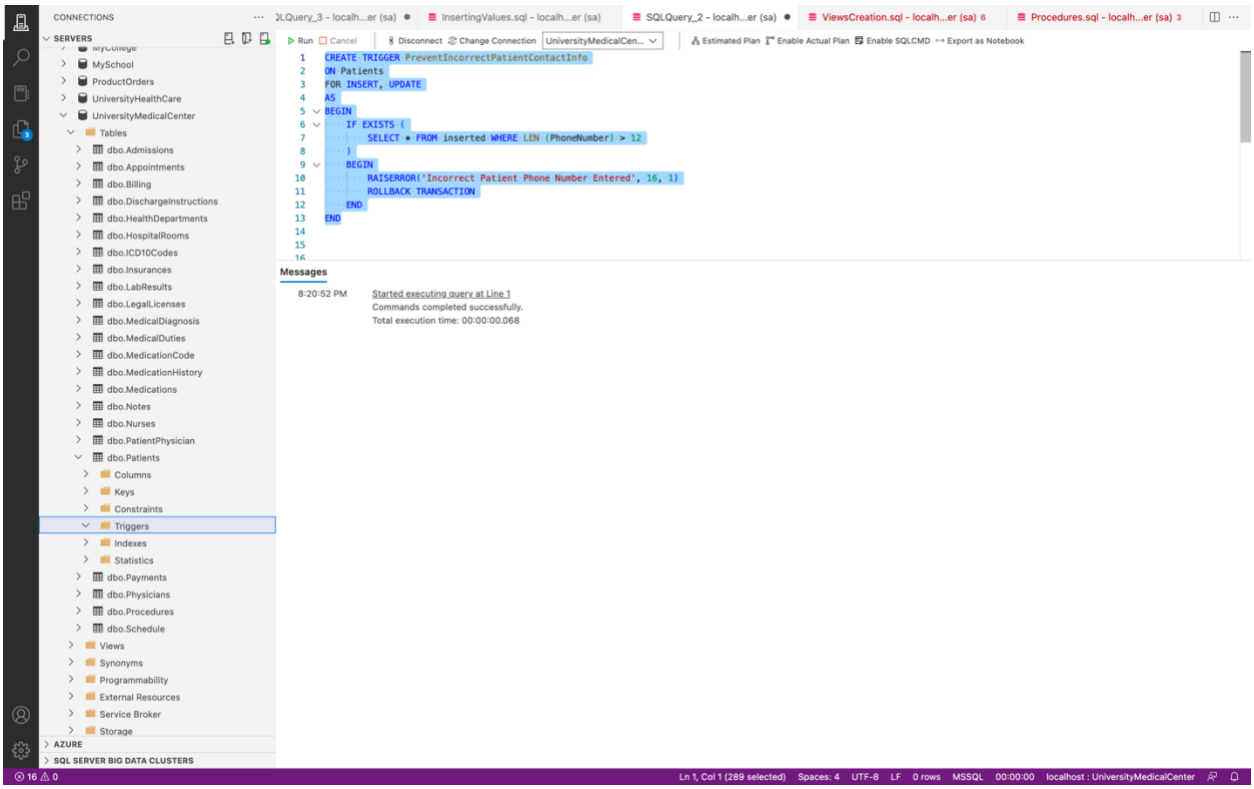
```
INSERT INTO Appointments(PatientPhysicianID, AppointmentDateTime, Reason)
VALUES(1,GETDATE(), 'Covid-19 Symptoms');
```



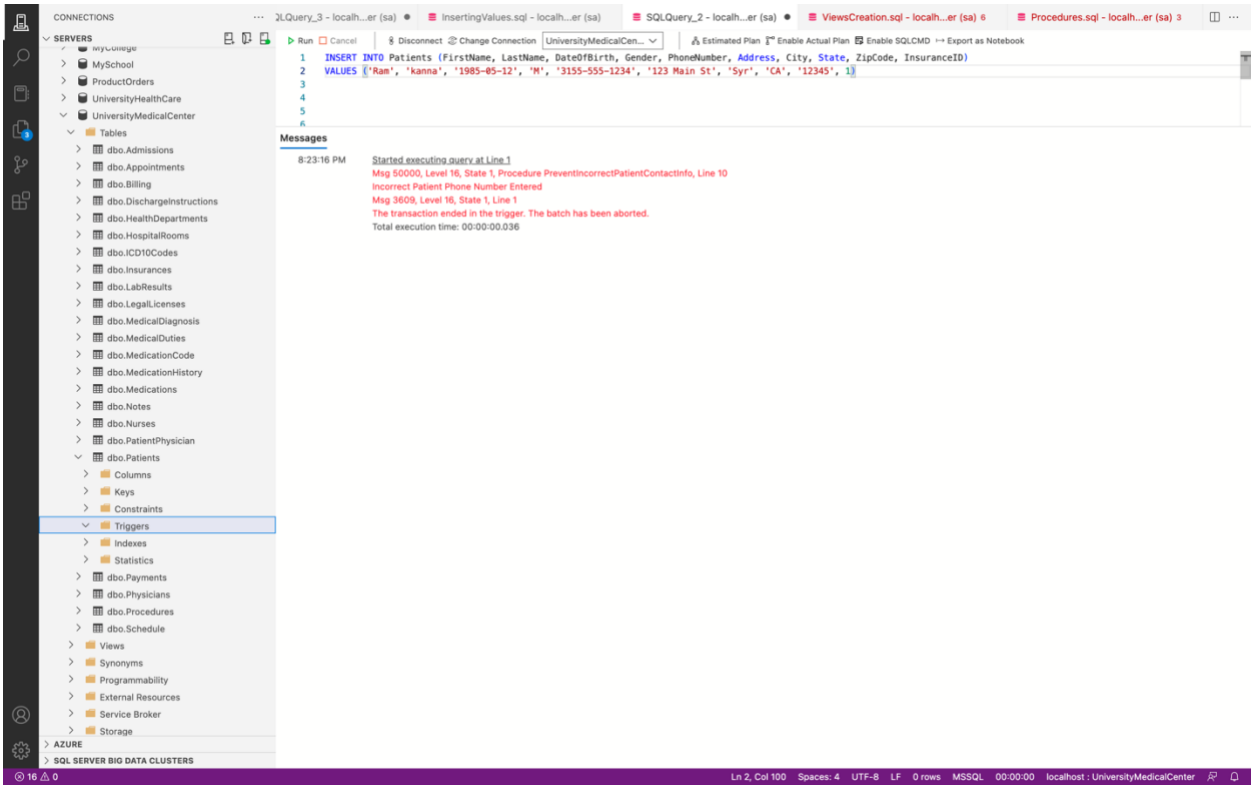
We got the error as expected.

2. Trigger for entering incorrect phone number

```
CREATE TRIGGER PreventIncorrectPatientContactInfo
ON Patients
FOR INSERT, UPDATE
AS
BEGIN
   IF EXISTS (
      SELECT * FROM inserted WHERE LEN (PhoneNumber) > 12
   )
   BEGIN
      RAISERROR('Incorrect Patient Phone Number Entered', 16, 1)
      ROLLBACK TRANSACTION
   END
END
```



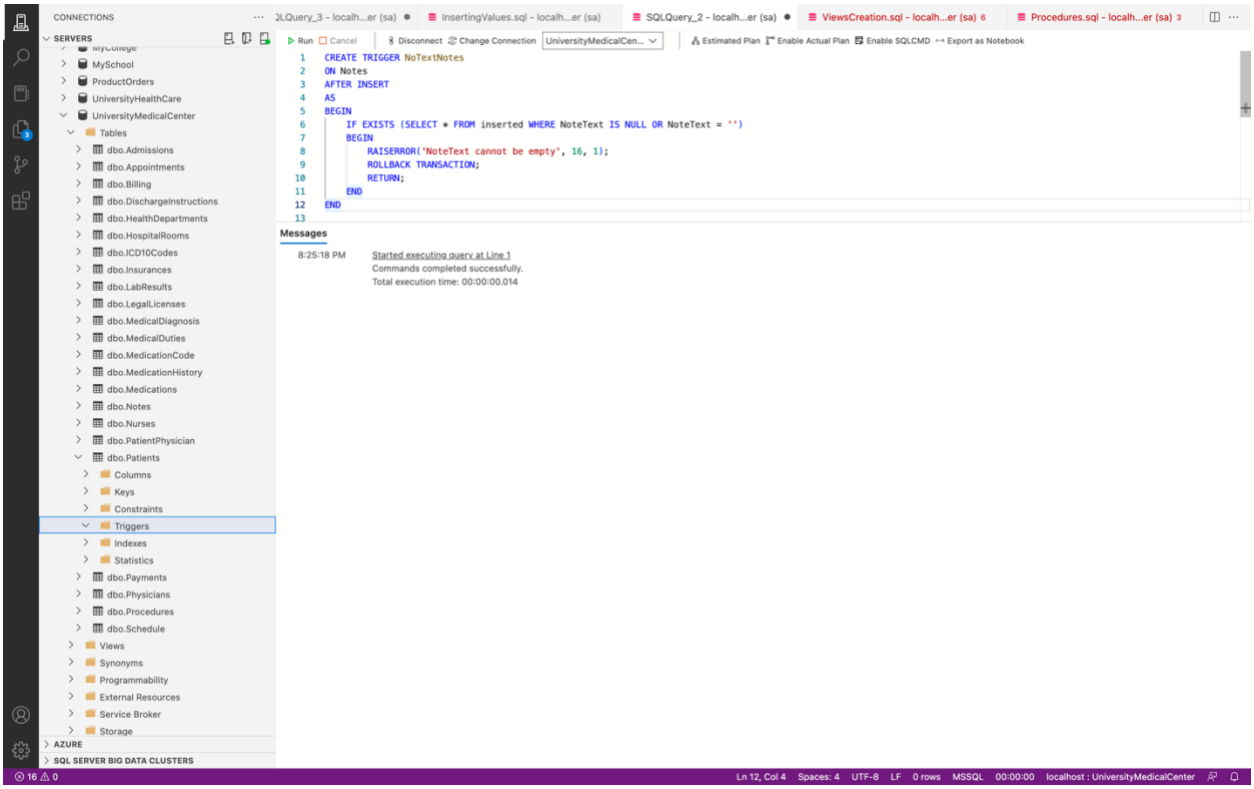
```
INSERT INTO Patients (FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Address, City, State, ZipCode, InsuranceID)
VALUES ('Ram', 'kanna', '1985-05-12', 'M', '3155-555-1234', '123 Main St', 'Syr', 'CA', '12345', 1)
```



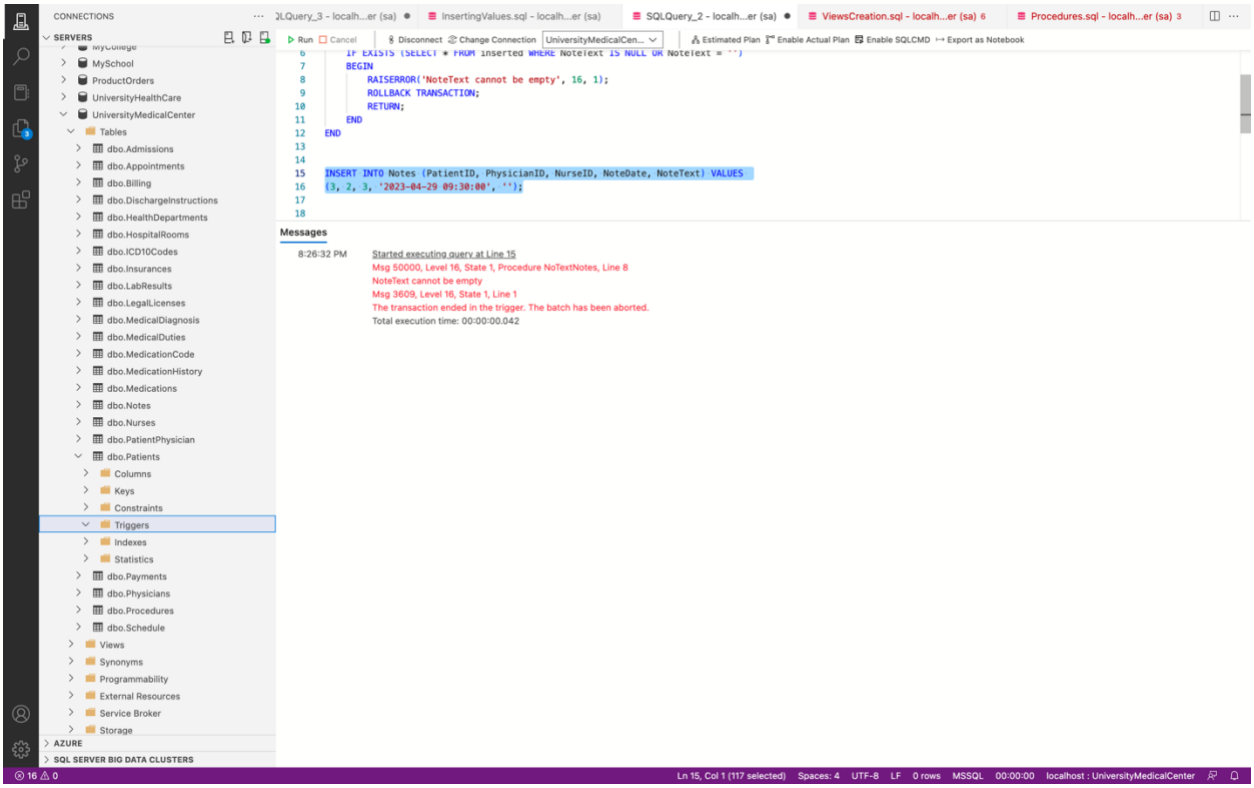
Error occurred as expected.

3. Trigger for getting notes with no text

```
CREATE TRIGGER NoTextNotes
ON Notes
AFTER INSERT
AS
BEGIN
   IF EXISTS (SELECT * FROM inserted WHERE NoteText IS NULL OR NoteText = '')
   BEGIN
      RAISERROR('NoteText cannot be empty', 16, 1);
      ROLLBACK TRANSACTION;
      RETURN;
   END
END
```



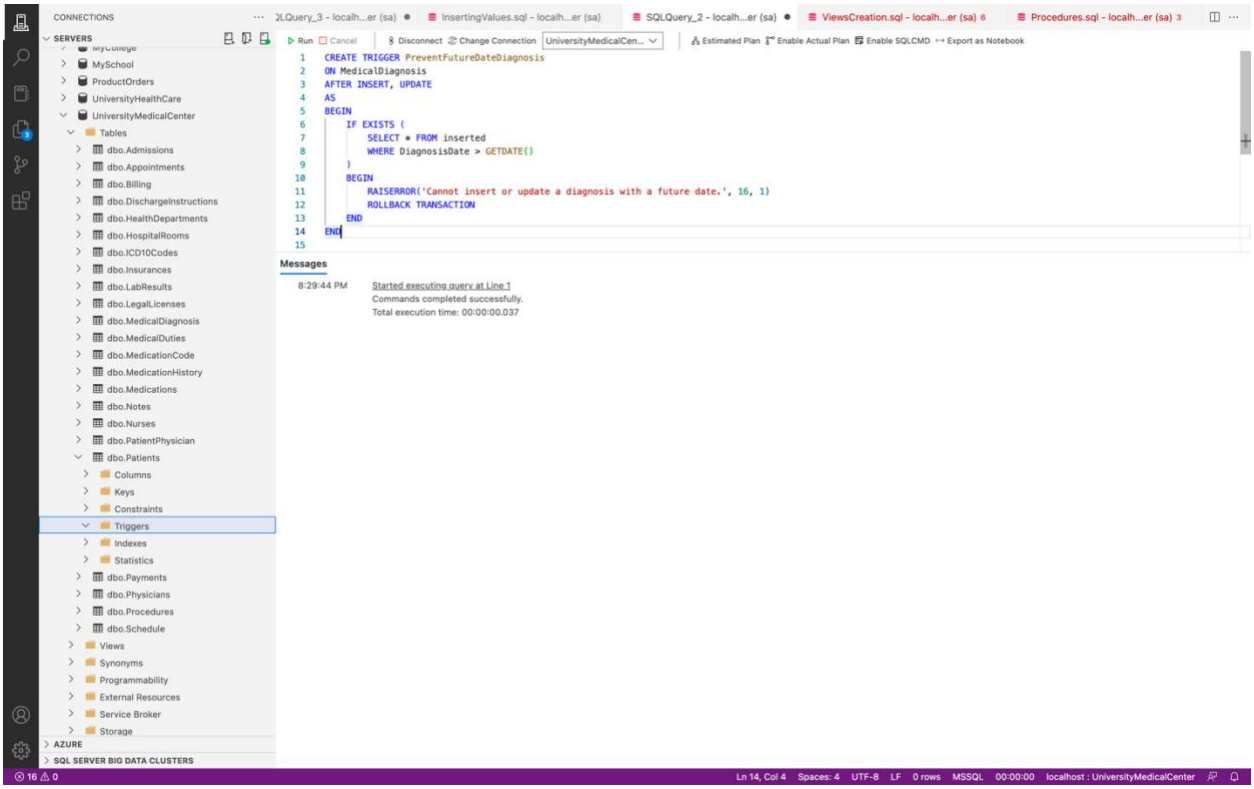
```
INSERT INTO Notes (PatientID, PhysicianID, NurseID, NoteDate, NoteText) VALUES
(3, 2, 3, '2023-04-29 09:30:00', '');
```



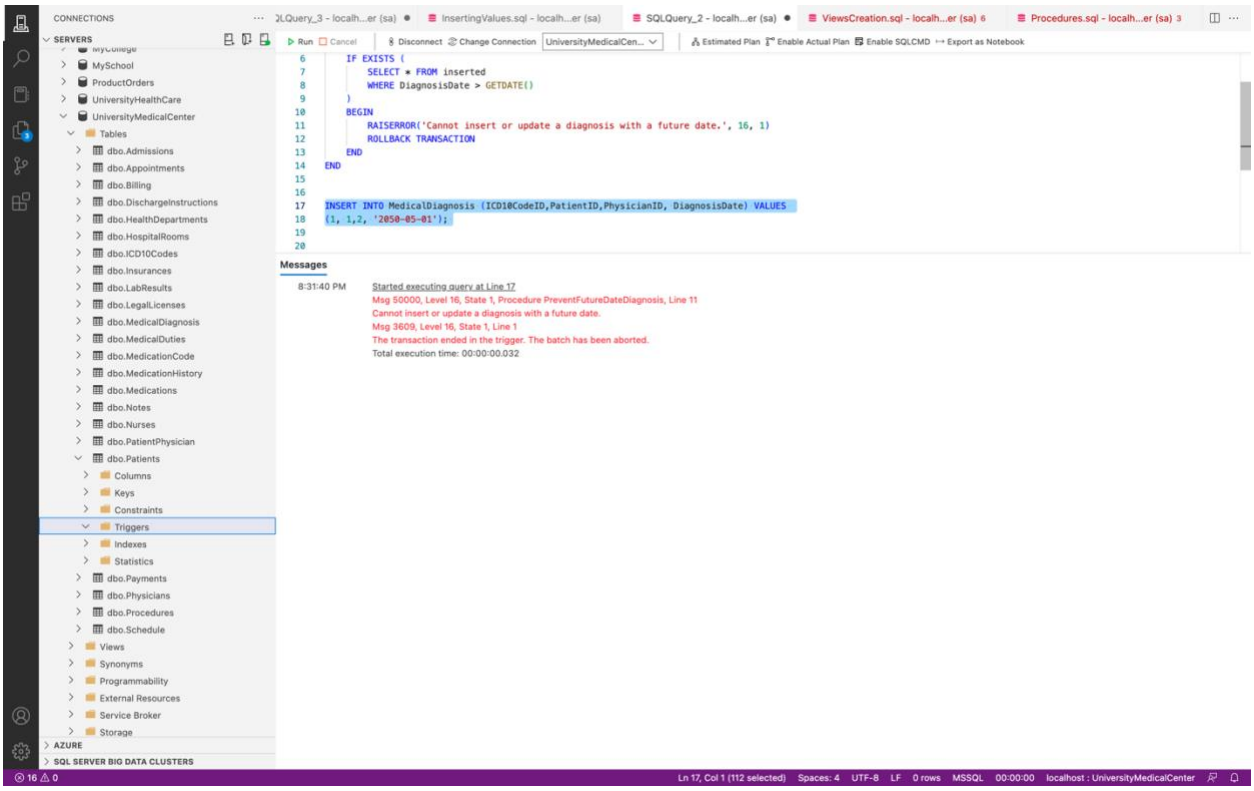
Error occurred As expected.

4. Trigger to prevent a diagnosis with future date

```
CREATE TRIGGER PreventFutureDateDiagnosis
ON MedicalDiagnosis
AFTER INSERT, UPDATE
AS
BEGIN
  IF EXISTS (
    SELECT * FROM inserted
    WHERE DiagnosisDate > GETDATE()
  )
  BEGIN
    RAISERROR('Cannot insert or update a diagnosis with a future date.', 16, 1)
    ROLLBACK TRANSACTION
  END
END
```



```
INSERT INTO MedicalDiagnosis (ICD10CodeID,PatientID,PhysicianID, DiagnosisDate) VALUES
(1, 1,2, '2050-05-01');
```

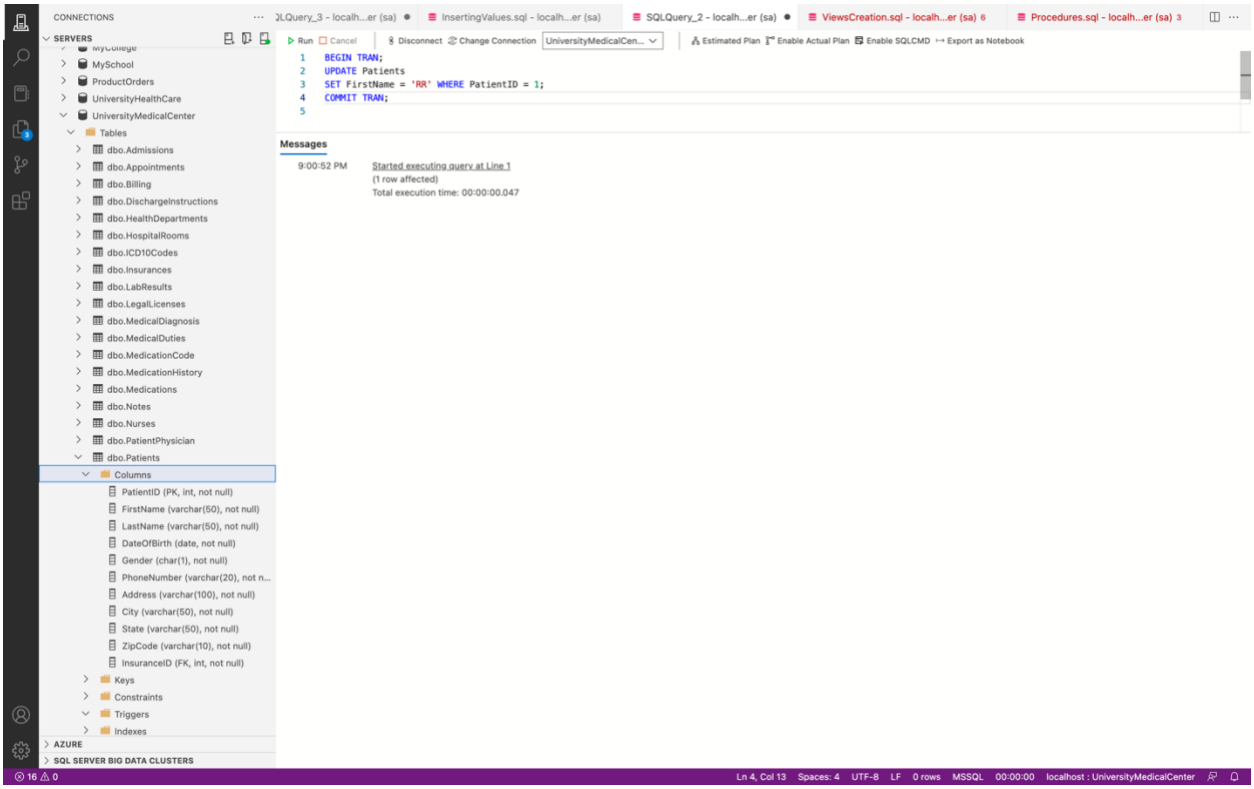


Error occurred as expected.

Transactions

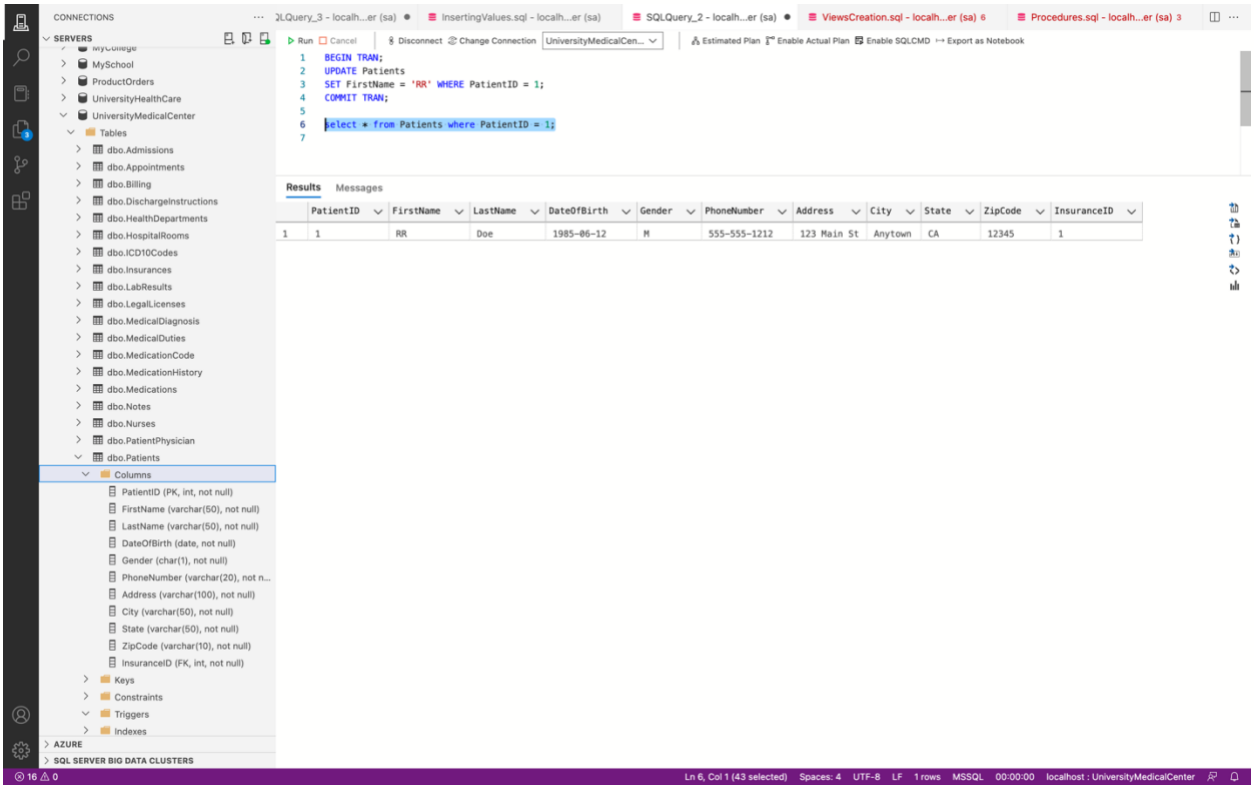
1.

```
BEGIN TRAN;
UPDATE Patients
SET FirstName = 'RR' WHERE PatientID = 1;
COMMIT TRAN;
```



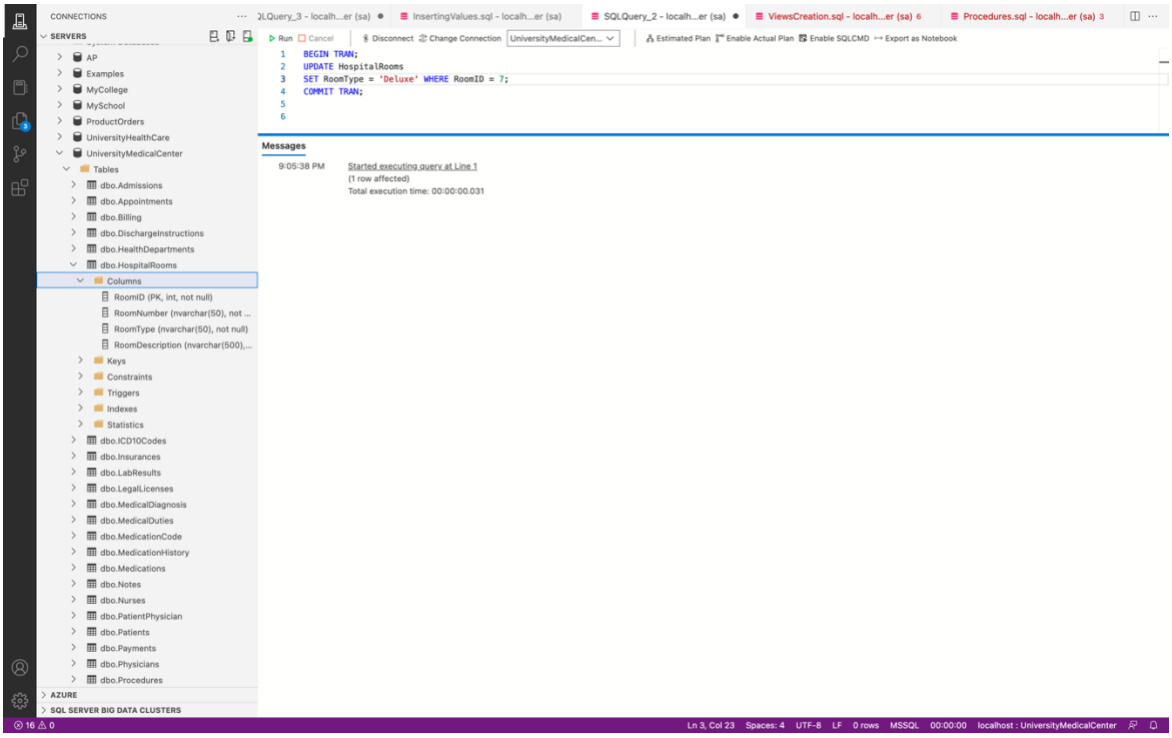
Updated the FirstName of PatientId 1

```
select * from Patients where PatientID = 1;
```



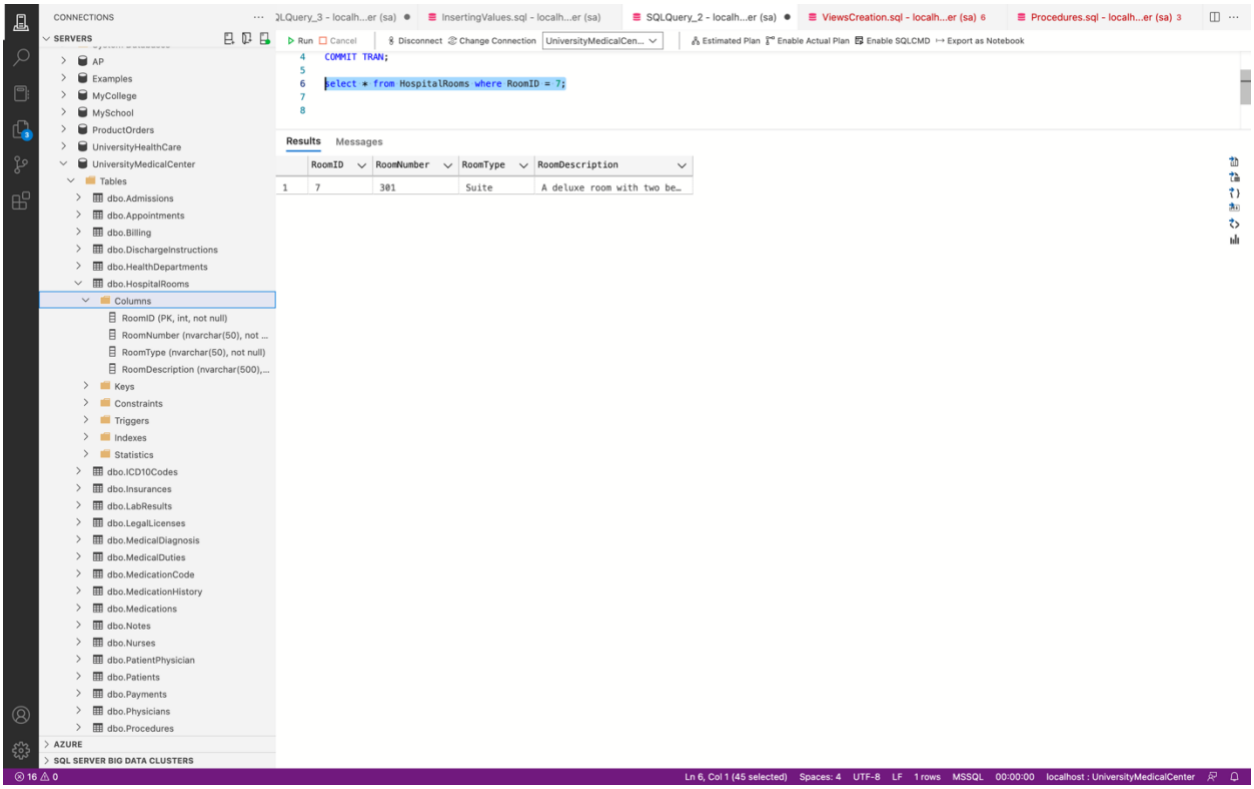
2.

```
BEGIN TRAN;
UPDATE HospitalRooms
SET RoomType = 'Suite' WHERE RoomID = 7;
COMMIT TRAN;
```



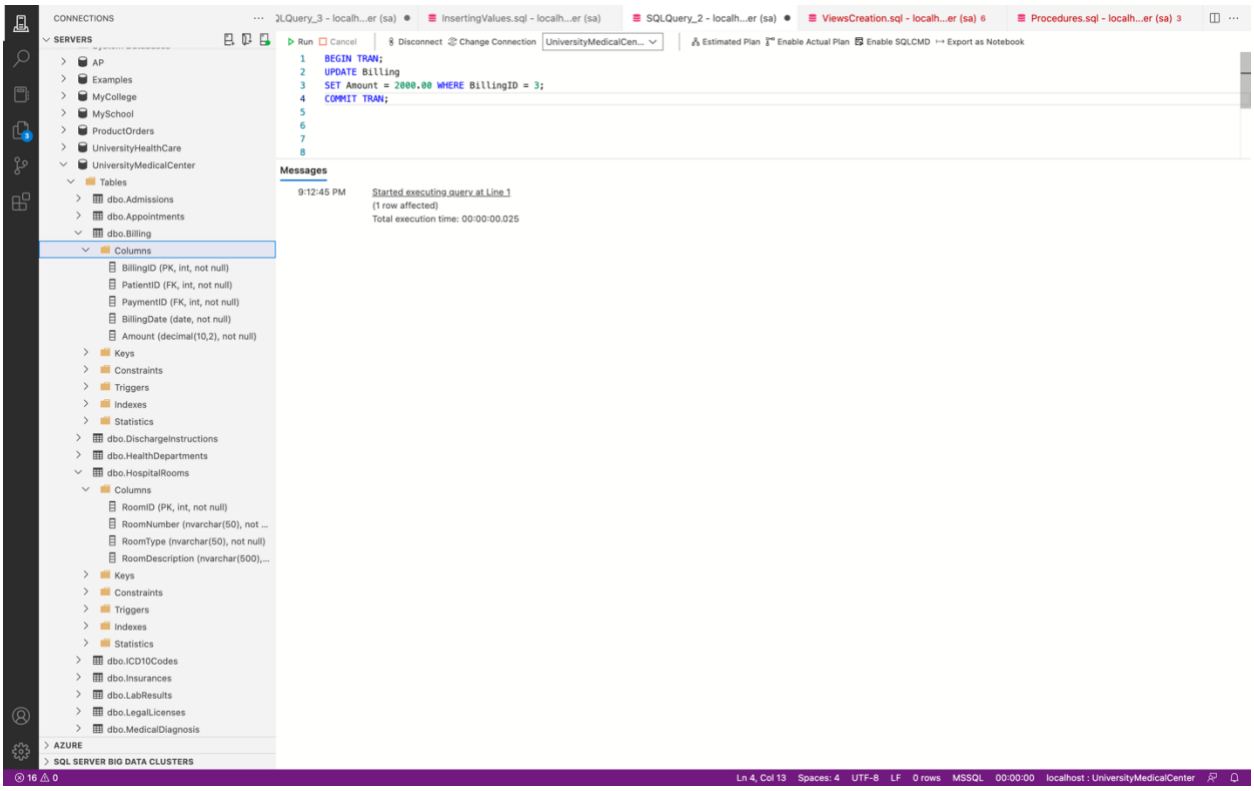
Updated the room type to Suite for roomID 7

```
select * from HospitalRooms where RoomID = 7;
```

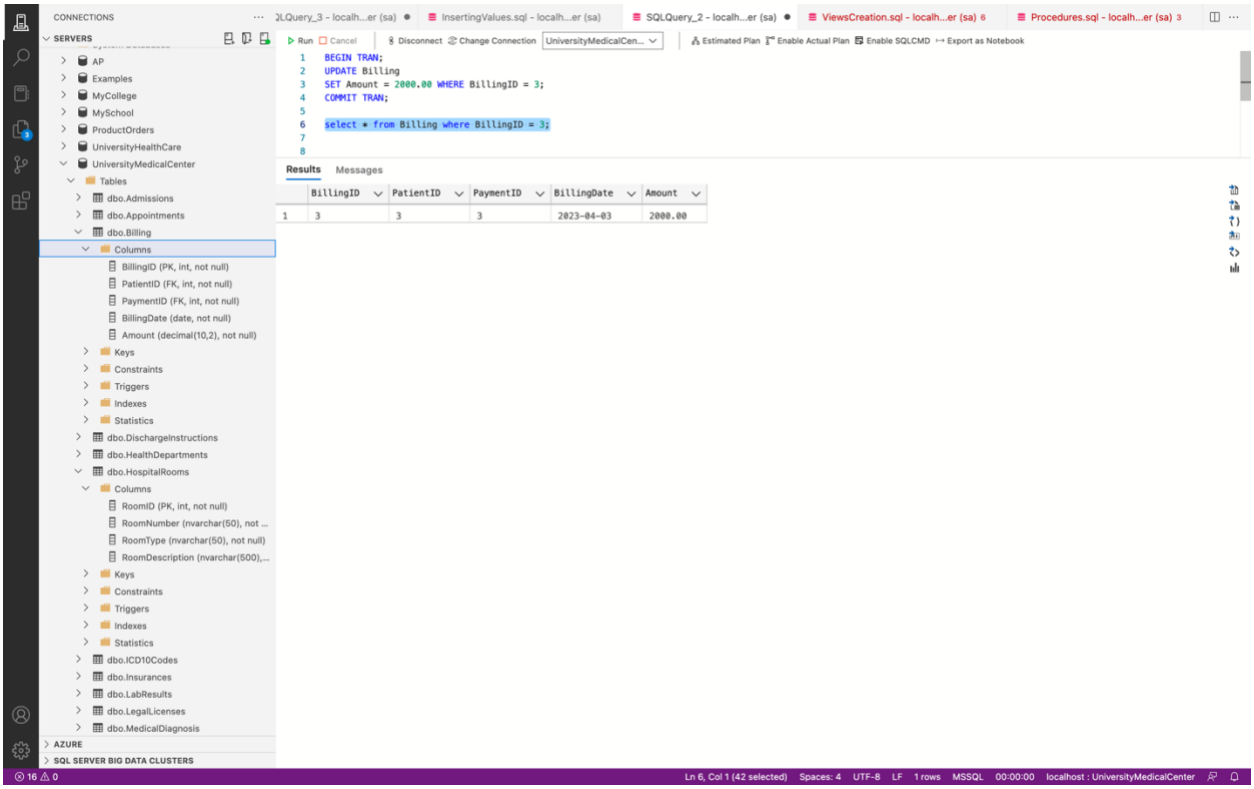


3.

```
BEGIN TRAN;
UPDATE Billing
SET Amount = 2000.00 WHERE BillingID = 3;
COMMIT TRAN;
```

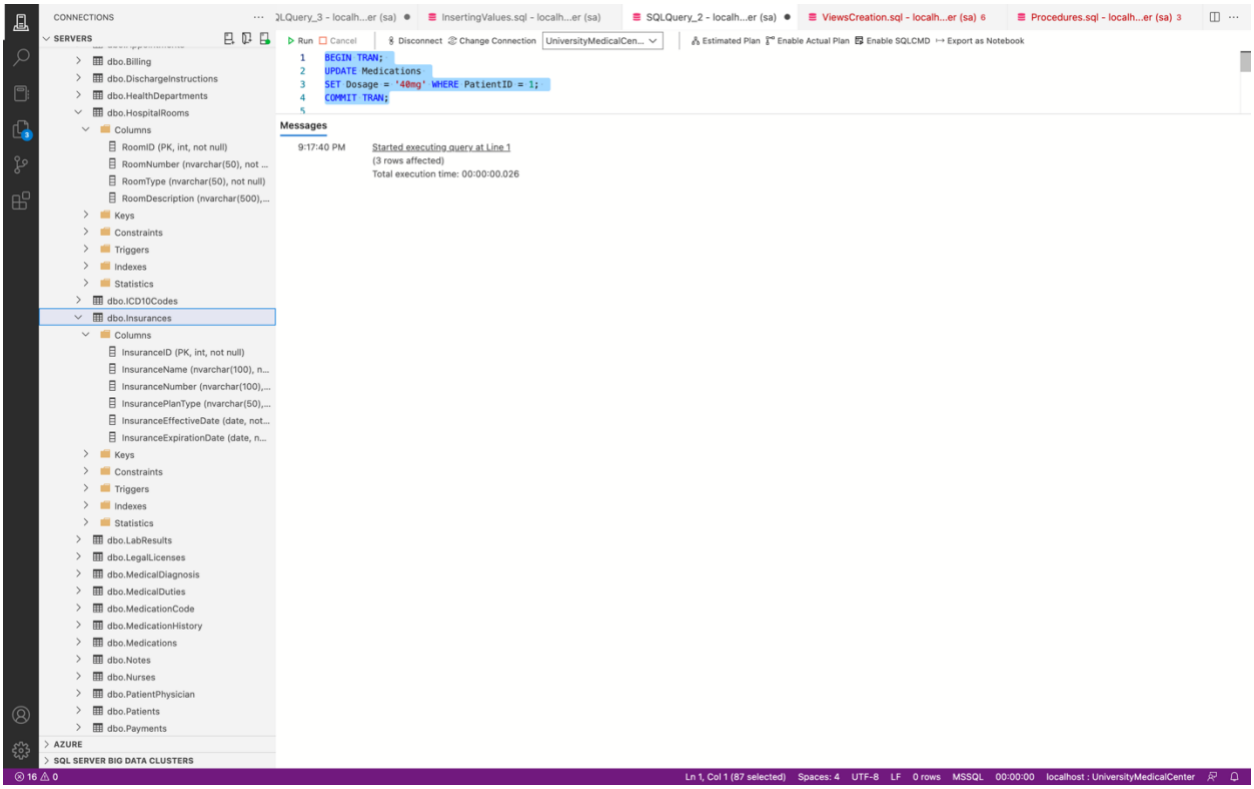


```
select * from Billing where BillingID = 3;
```



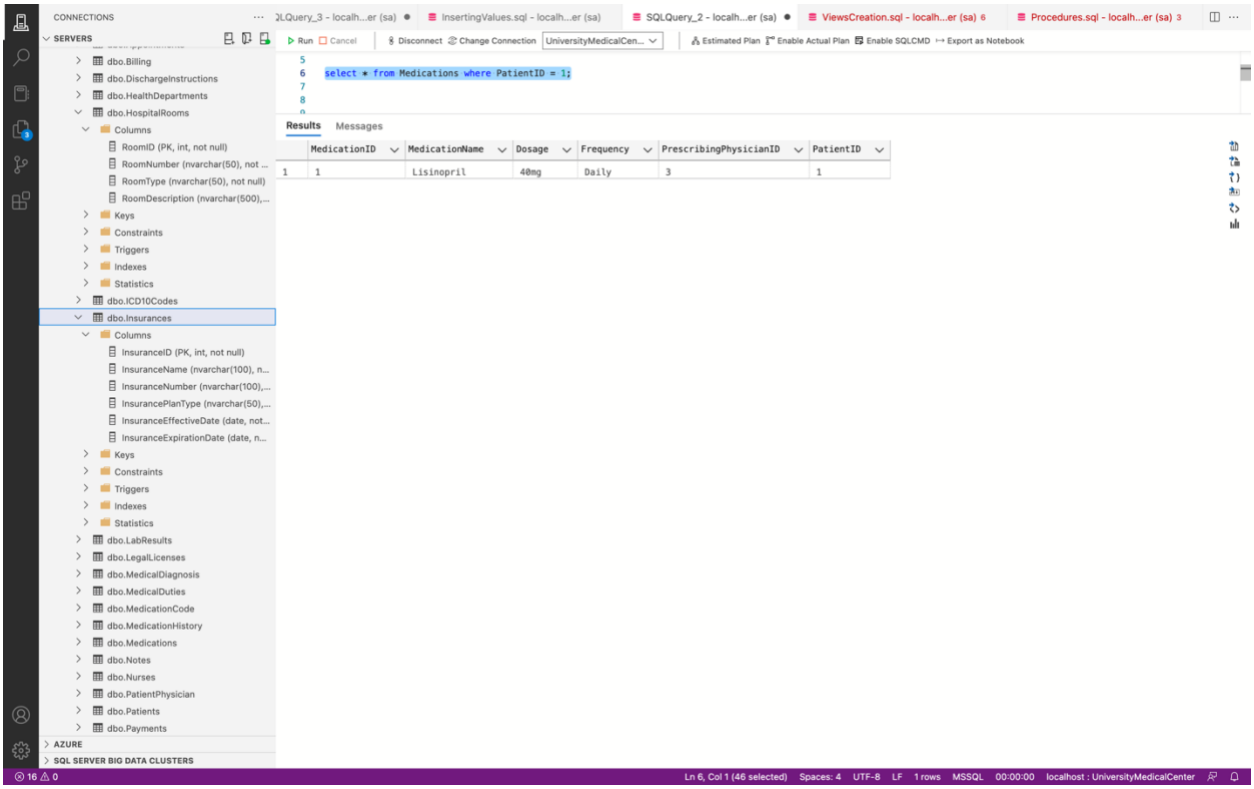
4.

```
BEGIN TRAN;
UPDATE Medications
SET Dosage = '40mg' WHERE PatientID = 1;
COMMIT TRAN;
```



Updated dosage of patientID to 40mg.

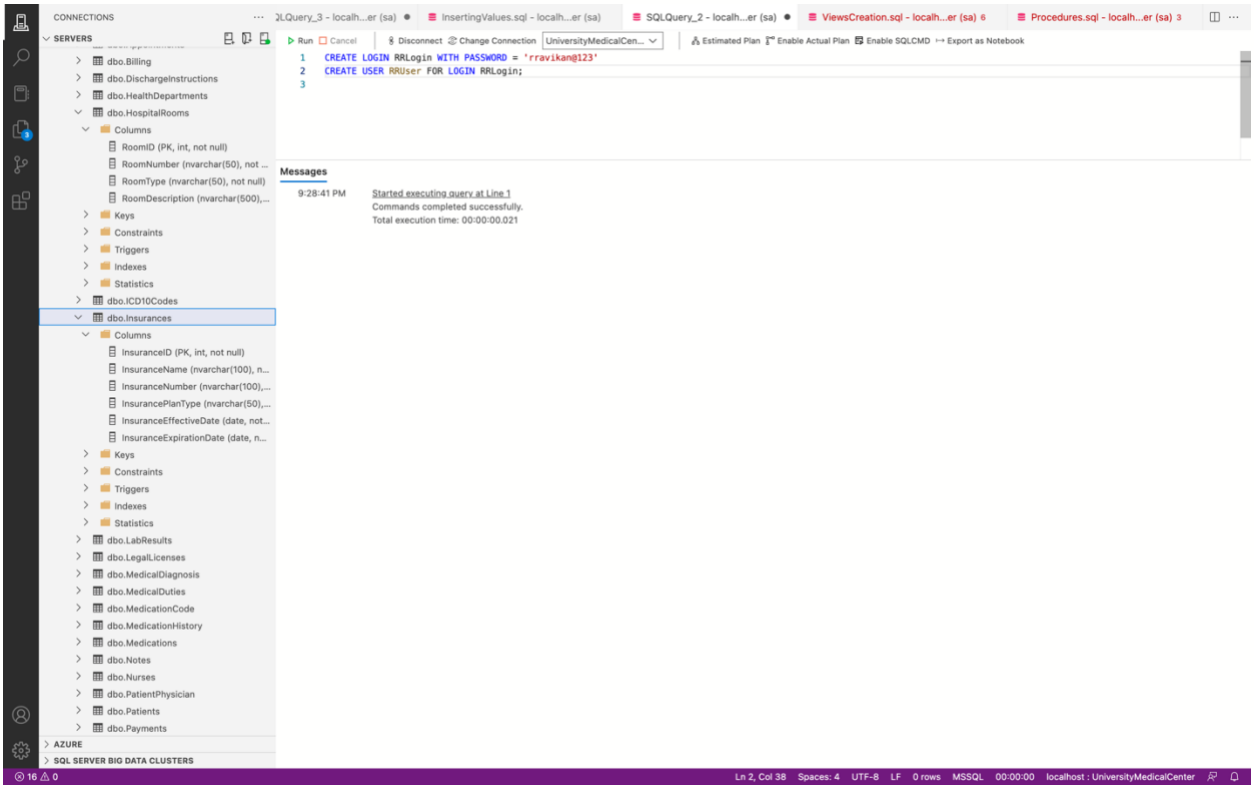
```
select * from Medications where PatientID = 1;
```



Scripts

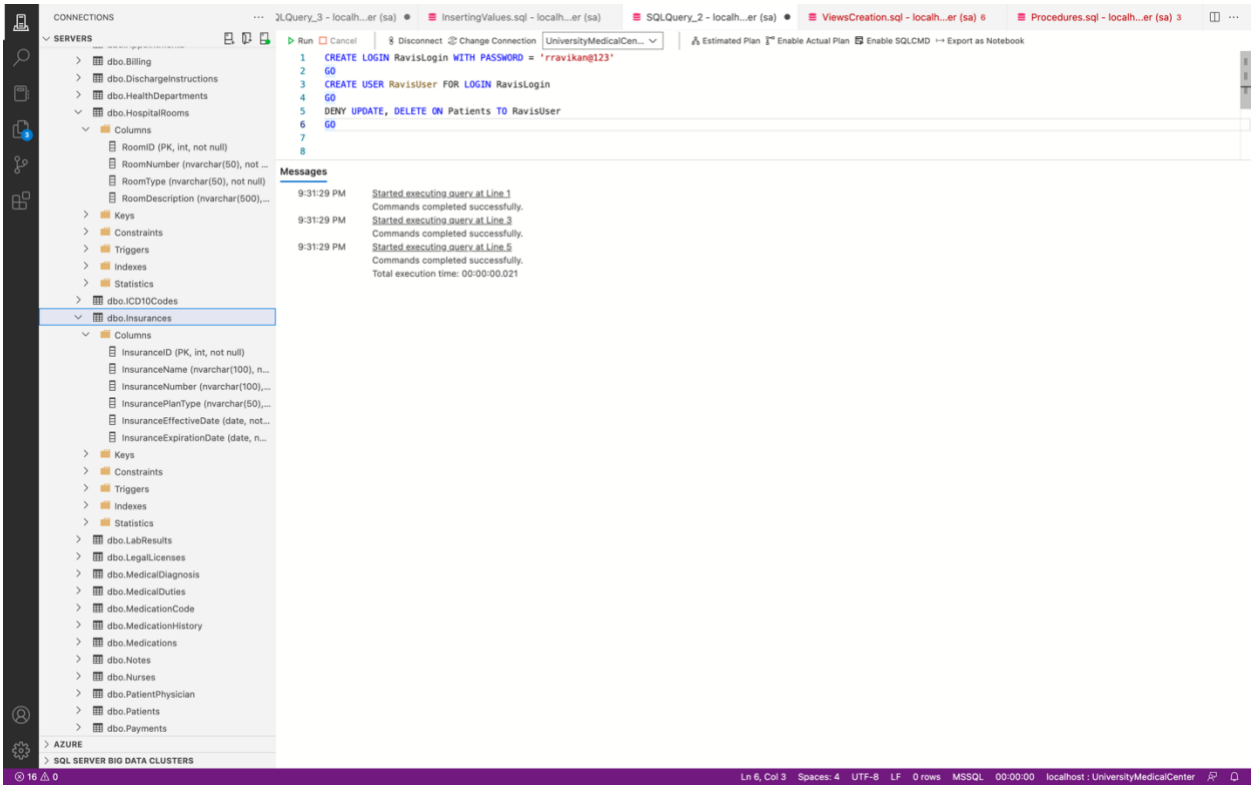
1.

```
CREATE LOGIN RRLogin WITH PASSWORD = 'rravikan@123'
CREATE USER RRUser FOR LOGIN RRLogin;
```



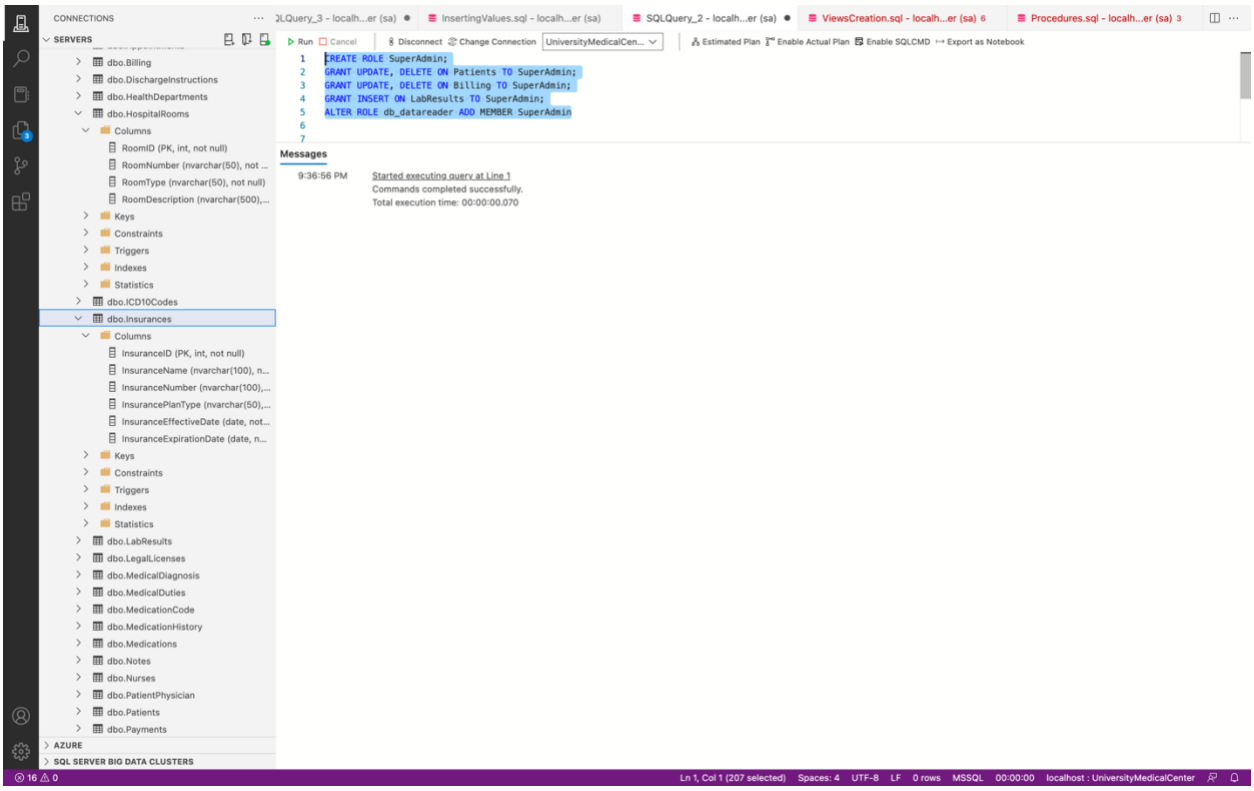
2.

```
CREATE LOGIN RavisLogin WITH PASSWORD = 'rravikan@123'
GO
CREATE USER RavisUser FOR LOGIN RavisLogin
GO
DENY UPDATE, DELETE ON Patients TO RavisUser
GO
```



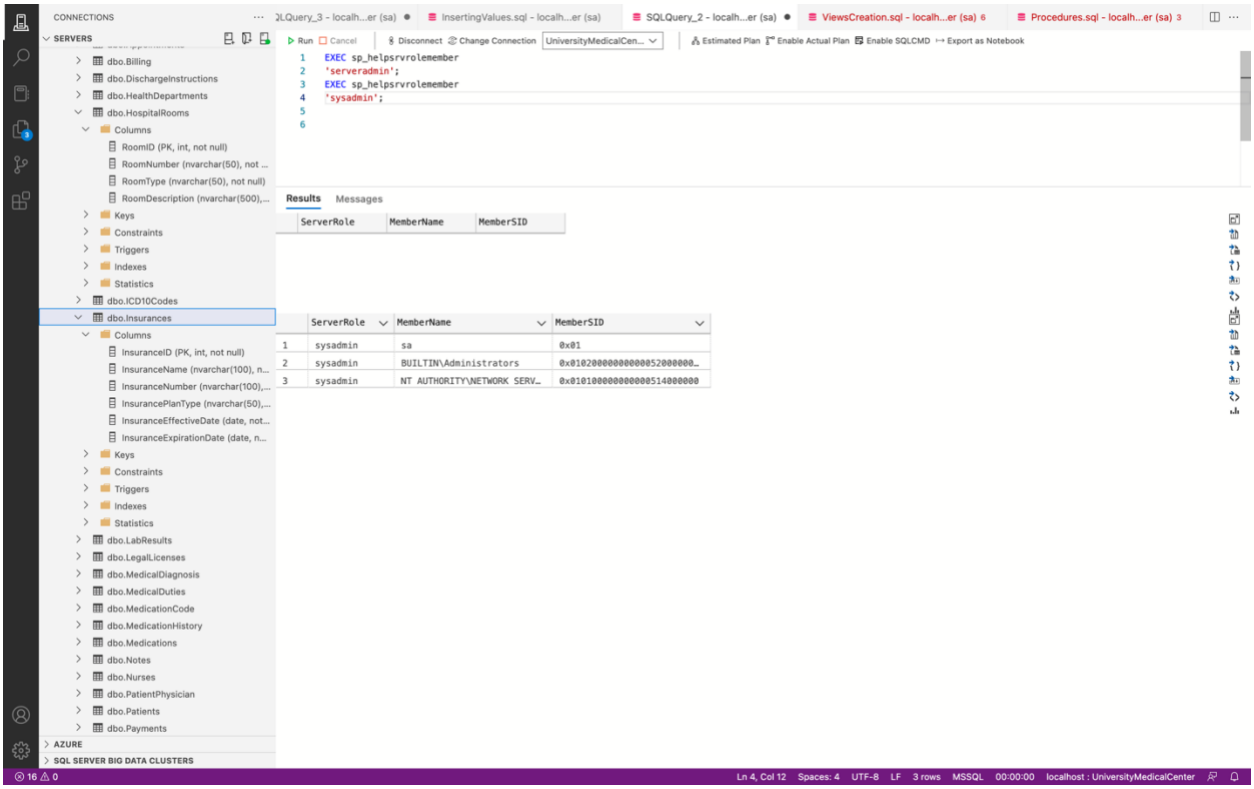
3.

```
CREATE ROLE SuperAdmin;
GRANT UPDATE, DELETE ON Patients TO SuperAdmin;
GRANT UPDATE, DELETE ON Billing TO SuperAdmin;
GRANT INSERT ON LabResults TO SuperAdmin;
ALTER ROLE db_datareader ADD MEMBER SuperAdmin
```



4.

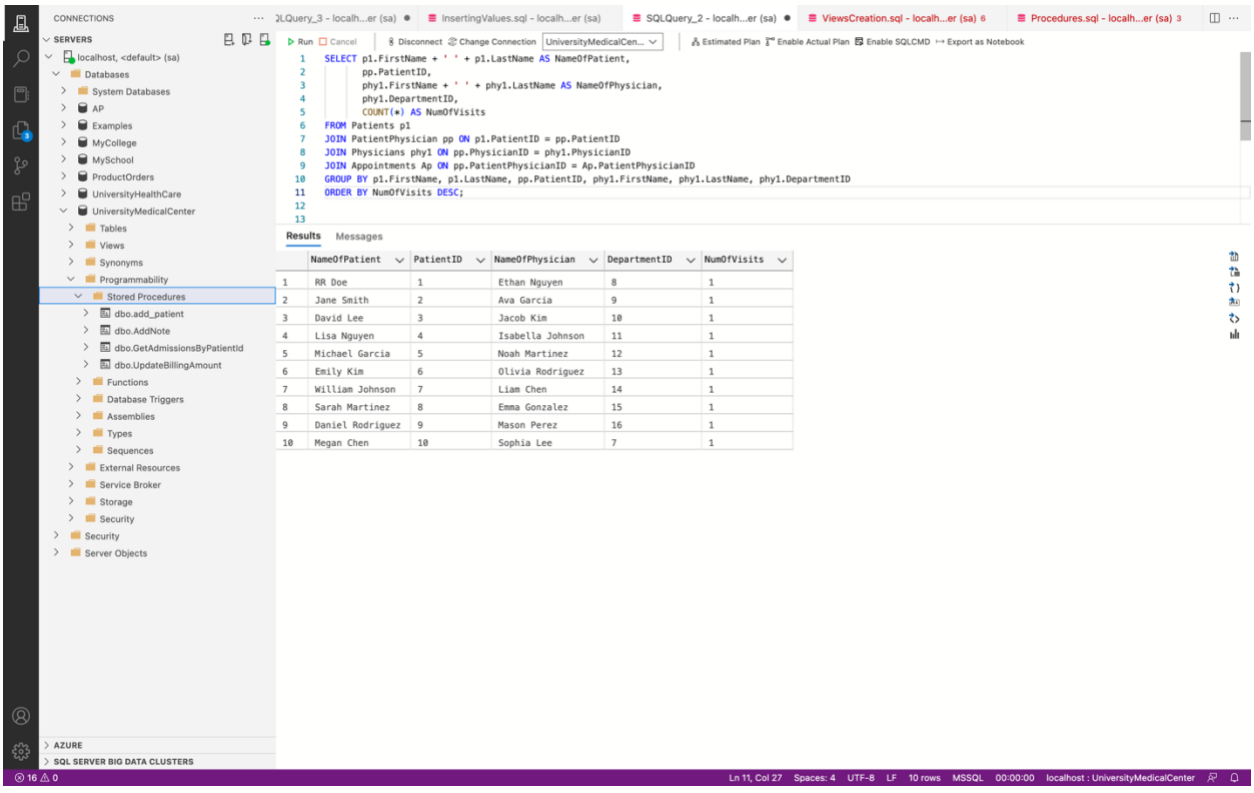
```
EXEC sp_helpsrvrolemember
'serveradmin';
EXEC sp_helpsrvrolemember
'sysadmin';
```



Business Reports

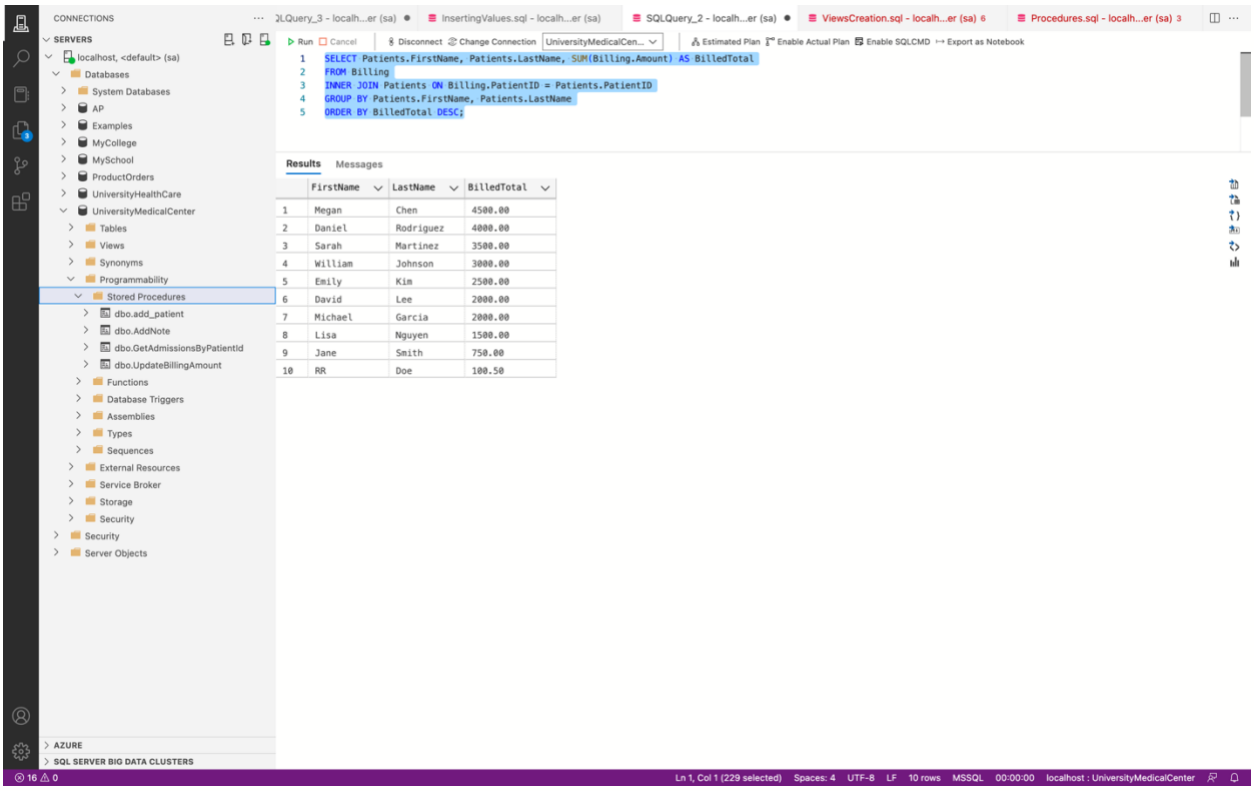
1.

```
SELECT p1.FirstName + ' ' + p1.LastName AS NameOfPatient,
    pp.PatientID,
    phy1.FirstName + ' ' + phy1.LastName AS NameOfPhysician,
    phy1.DepartmentID,
    COUNT(*) AS NumOfVisits
FROM Patients p1
JOIN PatientPhysician pp ON p1.PatientID = pp.PatientID
JOIN Physicians phy1 ON pp.PhysicianID = phy1.PhysicianID
JOIN Appointments Ap ON pp.PatientPhysicianID = Ap.PatientPhysicianID
GROUP BY p1.FirstName, p1.LastName, pp.PatientID, phy1.FirstName, phy1.LastName,
phy1.DepartmentID
ORDER BY NumOfVisits DESC;
```



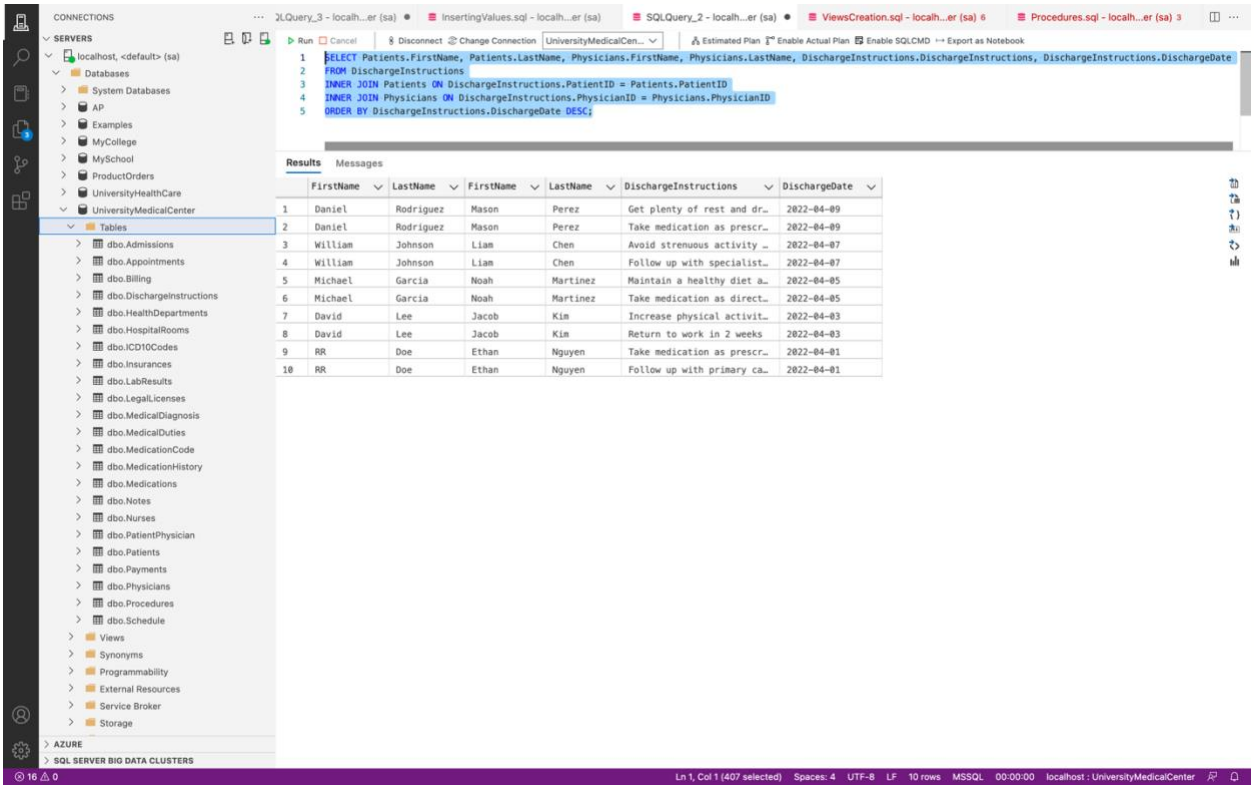
2.

```
SELECT Patients.FirstName, Patients.LastName, SUM(Billing.Amount) AS BilledTotal
FROM Billing
INNER JOIN Patients ON Billing.PatientID = Patients.PatientID
GROUP BY Patients.FirstName, Patients.LastName
ORDER BY BilledTotal DESC;
```



3.

```
SELECT Patients.FirstName, Patients.LastName, Physicians.FirstName, Physicians.LastName,
DischargeInstructions.DischargeInstructions, DischargeInstructions.DischargeDate
FROM DischargeInstructions
INNER JOIN Patients ON DischargeInstructions.PatientID = Patients.PatientID
INNER JOIN Physicians ON DischargeInstructions.PhysicianID = Physicians.PhysicianID
ORDER BY DischargeInstructions.DischargeDate DESC;
```

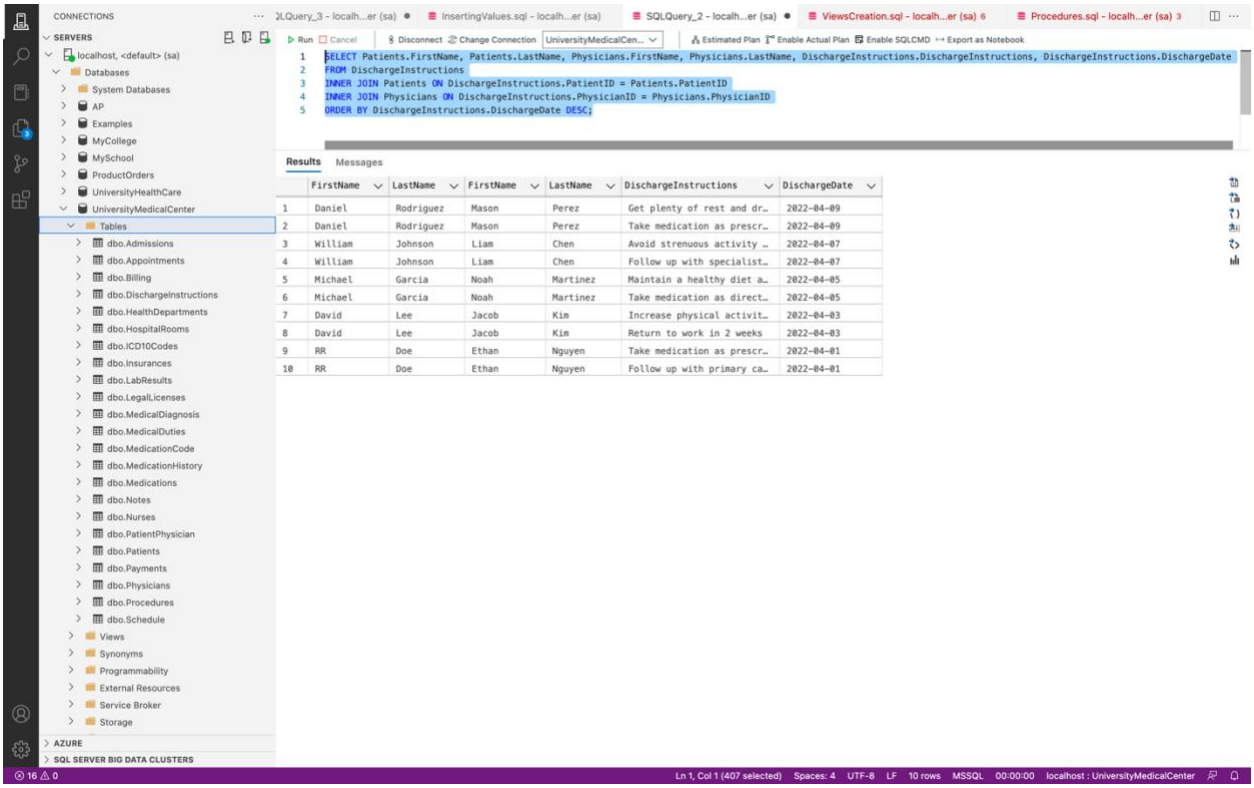


	FirstName	LastName	FirstName	LastName	DischargeInstructions	DischargeDate
1	Daniel	Rodriguez	Mason	Perez	Get plenty of rest and dr…	2022-04-09
2	Daniel	Rodriguez	Mason	Perez	Take medication as prescr…	2022-04-09
3	William	Johnson	Liam	Chen	Avoid strenuous activity …	2022-04-07
4	William	Johnson	Liam	Chen	Follow up with specialist…	2022-04-07
5	Michael	Garcia	Noah	Martinez	Maintain a healthy diet a…	2022-04-05
6	Michael	Garcia	Noah	Martinez	Take medication as direct…	2022-04-05
7	David	Lee	Jacob	Kim	Increase physical activit…	2022-04-03
8	David	Lee	Jacob	Kim	Return to work in 2 weeks	2022-04-03
9	RR	Doe	Ethan	Nguyen	Take medication as prescr…	2022-04-01
10	RR	Doe	Ethan	Nguyen	Follow up with primary ca…	2022-04-01

4.

```
SELECT HospitalRooms.RoomType, COUNT(*) AS CountOfAdmissions, COUNT(DISTINCT
Admissions.RoomID) AS CountOfOccupiedRooms
FROM Admissions
INNER JOIN HospitalRooms ON Admissions.RoomID = HospitalRooms.RoomID
GROUP BY HospitalRooms.RoomType;
```

```
SELECT Admissions.AdmissionID, CONCAT(Patients.FirstName, ' ', Patients.LastName) AS
NameOfPatient, HospitalRooms.RoomNumber, HospitalRooms.RoomType,
Admissions.AdmissionDate, Admissions.DischargeDate
FROM Admissions
INNER JOIN HospitalRooms ON Admissions.RoomID = HospitalRooms.RoomID
INNER JOIN Patients ON Admissions.PatientID = Patients.PatientID
ORDER BY HospitalRooms.RoomType, Admissions.AdmissionDate;
```



Conclusion

To conclude, the database design and implementation project was a comprehensive process that started with understanding the project statement and specifications, followed by determining the information to store, identifying tables, and addressing potential security and integrity issues. The next step involved designing an E/R diagram and normalizing the design into its 3rd Normal Form. Then, during the implementation phase, the schema was created along with tables, necessary relationships, and constraints, and potential security and integrity issues were resolved. This phase also involved creating views, stored procedures, user-defined functions, triggers, transactions, and scripts to create users with various security levels, passwords, and roles. During the testing phase, the database was populated with meaningful test data and the views, stored procedures, functions, triggers, transactions, and scripts were tested with complex scenarios. Finally, at least four business reports were generated.

The project highlighted the significance of careful planning, attention to detail, and a comprehensive testing process in ensuring database integrity and security. The resulting database provided a solid foundation for further development and growth, capable of supporting complex business reports and insights that could assist in decision-making and drive success.

Remarks

- The project has taught me the importance of careful planning when designing a database system for a complex organization like a hospital.
- I learned how to determine the necessary information to store and how to design tables and logic for testing and implementation.
- The project showed me how to address potential integrity and security issues in a database system.
- I gained experience in creating views, stored procedures, functions, triggers, and transactions for a database system.
- The project allowed me to practice creating business reports and extracting meaningful insights from data using complex queries with aggregate and analytic functions, CTEs, and OLAP transactions.
- I learned how to normalize a database design into its 3rd Normal Form and to use the CamelCase naming standard to make the design easy to read.
- The project taught me how to code a database system from scratch, without relying on auto-generated SQL.
- I gained experience in testing a database system's logic and integrity by running complex scenarios and transactions and explaining each scenario's objective in the comments area.
- I learned how to generate business reports to provide valuable insights to decision-makers, including creating visualizations using BI tools like Microsoft Power BI.
- The project allowed me to practice populating a database with test data to ensure its functionality.
- I gained a deeper understanding of the importance of database systems in healthcare organizations and how they can facilitate care coordination among healthcare providers and optimize hospital operations.
- The project showed me how to create a descriptive E/R diagram using a modeling tool like Untitled Diagram, Visual Paradigm, Vertabelo, or dbdiagram.io.
- I learned how to create scripts to create users with various security levels, passwords, and roles in a database system.
- The project taught me the importance of documenting my work and providing descriptive comments in my code to ensure its readability and maintainability.
- I gained experience in analyzing data and presenting my findings in a clear and concise manner in a report.