Design and Implementation of University Medical Center Database

CSE 581 - Intro to Database Management Systems

Submitted By Sai Kumar Palevela

Abstract - Healthcare databases are most frequently developed for the purpose of assessing the quality of healthcare, often for a specific disease or within a specific healthcare delivery system.

Accurate and comprehensive healthcare data are vitally important for a variety of purposes. These data may be used for local assessments or evaluations within a healthcare system, such as for specific outpatient conditions or inpatient hospital events. The data may also be used regionally or nationally for assessing performance within or across healthcare systems. Also, while comparisons become enormously difficult, administrative data may be used for comparing across national boundaries, to assess international differences in healthcare and disease.

In this project, we are going to design, implement a database keeping University medical Department in mind. But the principles and standards that we followed can be used to design another database of similar use case.

The implementation of the project includes creating an efficient database which can be used as backend storage for a graphical interface like web application by creating relevant tables. For making this project the efficient one I have used all the available database concepts like View, Stored Procedures, Triggers, Server Roles, Security, Encryption and User Defined Functions. Database design of my project is taken care by using Entity Relationship diagrams and by normalizing it to improve efficiency and reducing redundancy. Even though the main aim for designing is to track the medical facilities provided by university. The main goal of this project is to create a real-life environment which depicts the above model addressing the above business problems.

SAI KUMAR PALEVELA

SUID – 666053899

Contents

1) Database Name:

University Medical Center Database

This total Project is mainly divided in three parts.

- Design
- Implementation
- Testing

2) Design phase:

a) Tables Used

1	**Patients**	
	Patient ID	PK
	Patient First Name	
	Patient Last Name	
	Age	
	Phone Number	
	State	
	City	
	Medical Staff ID	FK
2	**MedicalStaff**	
	MedicalStaff ID	PK
	Medical Staff First name	
	Medical Staff Last Name	
	Address	
	Phone Number	
	Speciality ID	FK
	Hospital ID	FK
	Staff Type	FK
3	**StaffType**	
	StaffType	PK
	Name	
4	**Speciality**	
	SpecialityID	PK
	Designation	
5	**Hospitals**	
	Hospital ID	PK

	Hospital Name	
	Address	
6	**Room Type**	
	Room TypeID	PK
	Name	
7	**Department**	
	Department ID	PK
	Name	
	Description	
8	**Rooms**	
	RoomID	PK
	Room Capacity	
	DepartmentID	FK
	RoomType ID	FK
	Hospital ID	FK
	Price	
9	**MedicalEquipments**	
	MedicalEquipmentID	PK
	Equipment Name	
	Description	
10	**RoomEquipment**	
	RoomEquipment ID	PK
	Equipment ID	FK
	Room ID	FK
11	**ScheduleType**	
	ScheduleTypeID	PK
	Name	
	Description	
12	**MedicalStaffSchedule**	
	Schedule ID	
	Medical Staff ID	
	Schedule Type ID	
	Start Date Time	
	End Date Time	
	Description	
13	**Appointments**	
	Appointment ID	PK
	Patient ID	FK

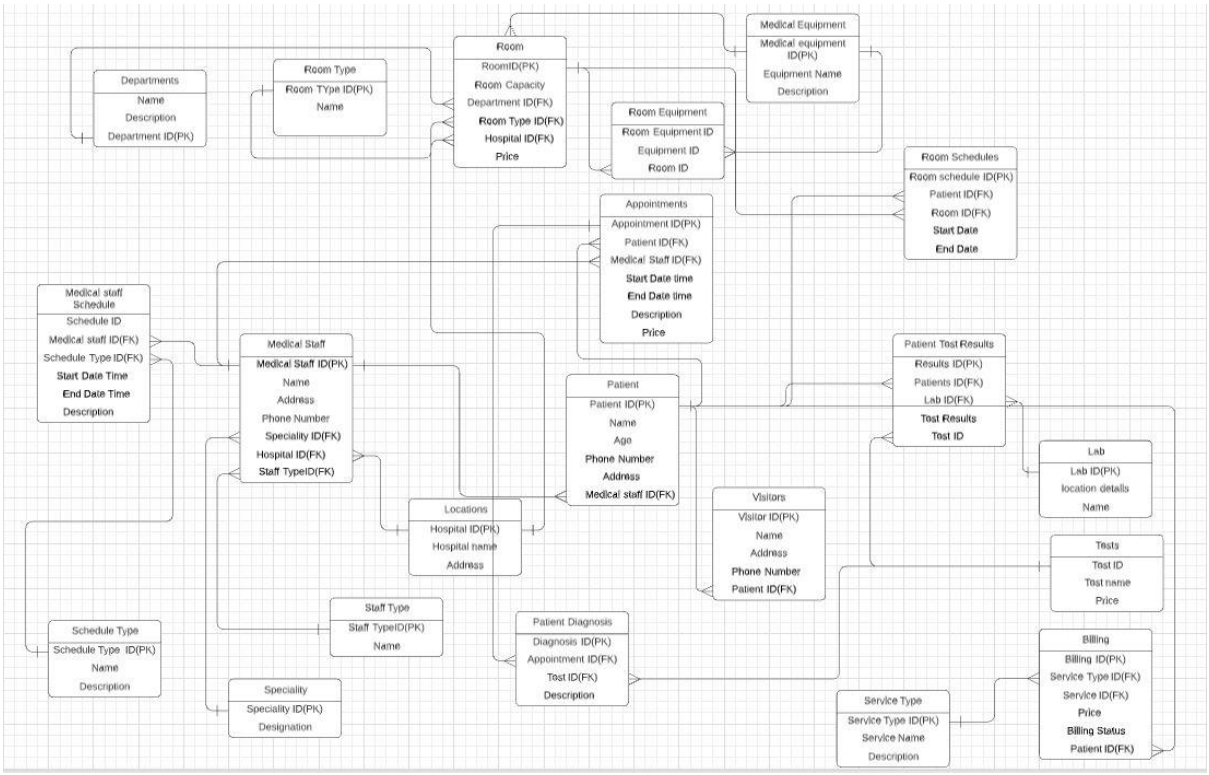
	Medical Staff ID	Fk
	Start Date time	
	End Date time	
	Description	
	Price	
14	**RoomSchedules**	
	Room Schedule ID	PK
	Patient ID	FK
	Room ID	FK
	Start Date	
	End Date	
15	**Labs**	
	Lab ID	PK
	Name	
	Location details	
16	**Tests**	
	Test ID	PK
	Test Name	
	Price	
17	**PatientsTestResults**	
	Results ID	PK
	Patients	
	Lab ID	FK
	Test Results	
	Test ID	FK
18	**Visitors**	
	Visitor ID	PK
	Visitor First Name	
	Visitor Last Name	
	State	
	City	
	Phone Number	
	Patient ID	FK
19	**PatientDiagnosis**	
	Diagnosis ID	PK
	Test ID	FK
	Description	
20	**Billings**	
	Billing ID	PK

	Service Type ID	FK
	Service ID	FK
	Price	
	Billing Status	
	Patient ID	FK
21	**ServiceTypes**	
	ServiceType ID	PK
	Service Name	
	Description	

b) E/R Diagram:

The Entity Relationship(E/R) diagram mainly focusses on how entities in the database are related to each other and how to link them.

Here is the E/R diagram given below that is used for my project.



c) Potential Integrity and Security issues

The potential security and integrity can be reduced by transforming the above database into the above-mentioned format of creating more tables i.e. Normalizing it.
For referential integrity to hold in a relational database, any column in a base table that is declared a foreign key can only contain either null values or values from a parent table's primary key or a candidate key. In other words when a foreign key value is used it must reference a valid, existing primary key in the parent table. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

3) Implementation Phase

a) constraints

Typically, there are two types of constraints. They are:

1. **Normal Constraints**: - Normal constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted. Various types of normal constraints are: NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECH, DEFAULT, UNIQUE, INDEX.
2. **Business Constraints**: They include the relationships between the tables such as One-to-One, Many-to-One, Many-to-One.

In this project I have used One-to-One relations, Many-to-One relations, Many-to-Many relations.
Primary keys (PK), data types, nullable and relationships that are used in my project are listed.

1) Patients Table:

PatientID int IDENTITY PRIMARY KEY NOT NULL,
PatientFirstName varchar (50) NOT NULL,
PatientLastName varchar (50) NOT NULL,
Age int NOT NULL,
City varchar (50) NOT NULL,
State char (2) NOT NULL,
FOREIGN KEY(MedicalStaffID) REFERENCES MedicalStaff (MedicalStaffID)

2) MedicalStaff Table:

MedicalStaffID int IDENTITY PRIMARY KEY NOT NULL,
MedicalStaffFirstName varchar (50) NOT NULL,
MedicalStaffLastName varchar (50) NOT NULL,
Address varchar (50) NOT NULL,
PhoneNumber varchar (50) NULL
FOREIGN KEY(SpecialityID) REFERENCES Speciality(SpecialityID)
FOREIGN KEY(HospitalID) REFERENCES Hospitals(HospitalID)
FOREIGN KEY(StaffTypeID) REFERENCES StaffType(StaffTypeID)

3) StaffType Table:

StaffTypeID int NOT NULL IDENTITY PRIMARY KEY,
Name varchar (50) NOT NULL

4) Specialities Table:

SpecialityID int NOT NULL IDENTITY PRIMARY KEY,
Designation varchar (20) NOT NULL

5) Hospitals Table:

HospitalID int NOT NULL IDENTITY PRIMARY KEY,
Name varchar (50) NOT NULL,
Description varchar (50),

6) Room Types Table

RoomTypeID int NOT NULL IDENTITY PRIMARY KEY,
Name varchar(50) NOT NULL,

7) **Departments Table**

DepartmentID INT IDENTITY PRIMARY KEY NOT NULL,
Name varchar(50) NOT NULL,
Description varchar(50)

8) Rooms Table

RoomID INT IDENTITY PRIMARY KEY NOT NULL,
RoomCapacity varchar (20) NOT NULL,
FOREIGN KEY(DepartmentID) REFERENCES Departments(DepartmentID)
FOREIGN KEY(RoomTypeID) REFERENCES RoomType(RoomTypeID)
FOREIGN KEY(HospitalID) REFERENCES Hospitals(HospitalID)

9) MedicalEquipments Table

MedicalEquipmentID INT IDENTITY PRIMARY KEY NOT NULL,
EquipmentName varchar(50) NOT NULL,
Descriptions varchar(50)

10) RoomEquipments Table

RoomEquipmentID INT NOT NUll IDENTITY PRIMARY KEY,
FOREIGN KEY(EquipmentID) REFERENCES MedicalEquipments(MedicalEquipmentID)
FOREIGN KEY(RoomID) REFERENCES Rooms(RoomID)

11) ScheduleTypes Table

ScheduleTypeID INT NOT NULL IDENTITY PRIMARY KEY,

Name varchar(50) NOT NULL,
Description varchar(50)

12) MedicalStaffSchedules Table

ScheduleID INT NOT NULL IDENTITY PRIMARY KEY,
StartDateTime datetime,
EndDateTime datetime,
Descriptions varchar(50),
FOREIGN KEY(MedicalStaffID) REFERENCES MedicalStaff(MedicalStaffID)
FOREIGN KEY(ScheduleTypeID) REFERENCES ScheduleTypes(ScheduleTypeID)

13) Appointments Table

AppointmentID INT NOT NULL IDENTITY PRIMARY KEY,
StartDateTime datetime,
EndDateTime datetime,
Description varchar(50),
Price int,
FOREIGN KEY(PatientID) REFERENCES Patients(PatientID)
FOREIGN KEY(MedicalStaffID) REFERENCES MedicalStaff(MedicalStaffID)

14) RoomSchedules Table

RoomScheduleID INT NOT NULL IDENTITY PRIMARY KEY,
StartDateTime datetime,
EndDateTime datetime,
FOREIGN KEY(PatientID) REFERENCES Patients(PatientID)
FOREIGN KEY(RoomID) REFERENCES Rooms(RoomID)

15) Labs Table

LabID INT NOT NULL IDENTITY PRIMARY KEY,
LabName varchar(50) NOT NULL,
LocationDetails varchar(50),

16) Tests Table

TestID INT NOT NULL IDENTITY PRIMARY KEY,
TestName varchar(50) NOTNULL,
Price int

17) PatientsTestResults Table

TestResultID INT NOT NULL IDENTITY PRIMARY KEY,
TestResults varchar(50) NOTNULL,

FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
FOREIGN KEY (TestID) REFERENCES Tests(TestID),
FOREIGN KEY (LabID) REFERENCES Labs(LabID)

18) Visitors Table

VisitorID int IDENTITY PRIMARY KEY NOT NULL,
VisitorFirstName varchar (50) NOT NULL,
VisitorLastName varchar (50) NOT NULL,
City varchar (50) NOT NULL,
State char (2) NOT NULL,
PhoneNumber varchar(20),
FOREIGN KEY(PatientID) REFERENCES Patients (PatientID)

19) PatientDiagosis Table

DiagnosisID int IDENTITY PRIMARY KEY NOT NULL,
Description varchar (50) NOT NULL,
FOREIGN KEY(TestID) REFERENCES Tests (TestID)

20) Billings Table

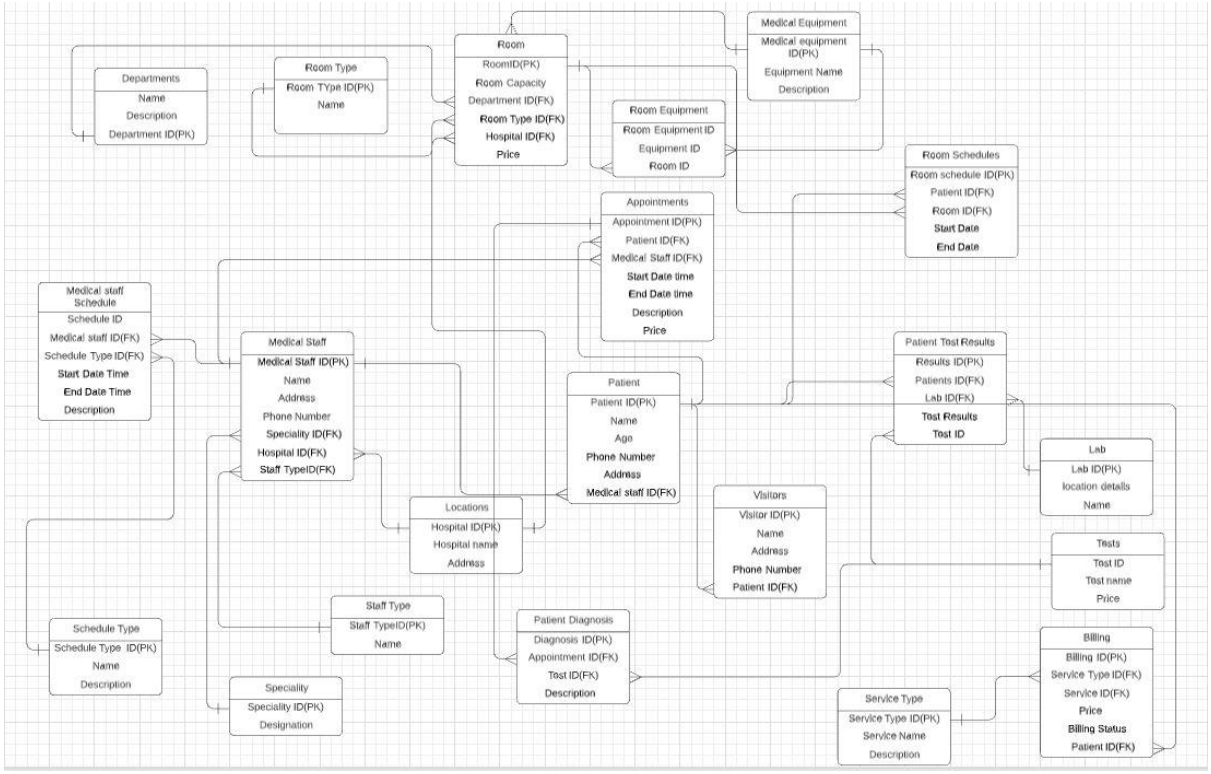
Billing ID int IDENTITY PRIMARY KEY NOT NULL,
Price Int,
BillingStatus int NOTNULL,
Service ID int,
FOREIGN KEY(PatientID) REFERENCES Patients (PatientID),
FOREIGN KEY(ServiceTypeID) REFERENCES ServiceTypes (ServiceTypeID)

21) ServiceTypes Table

ServiceTypeID int IDENTITY PRIMARY KEY NOT NULL,
ServiceName varchar(50),
Description Varchar(50)

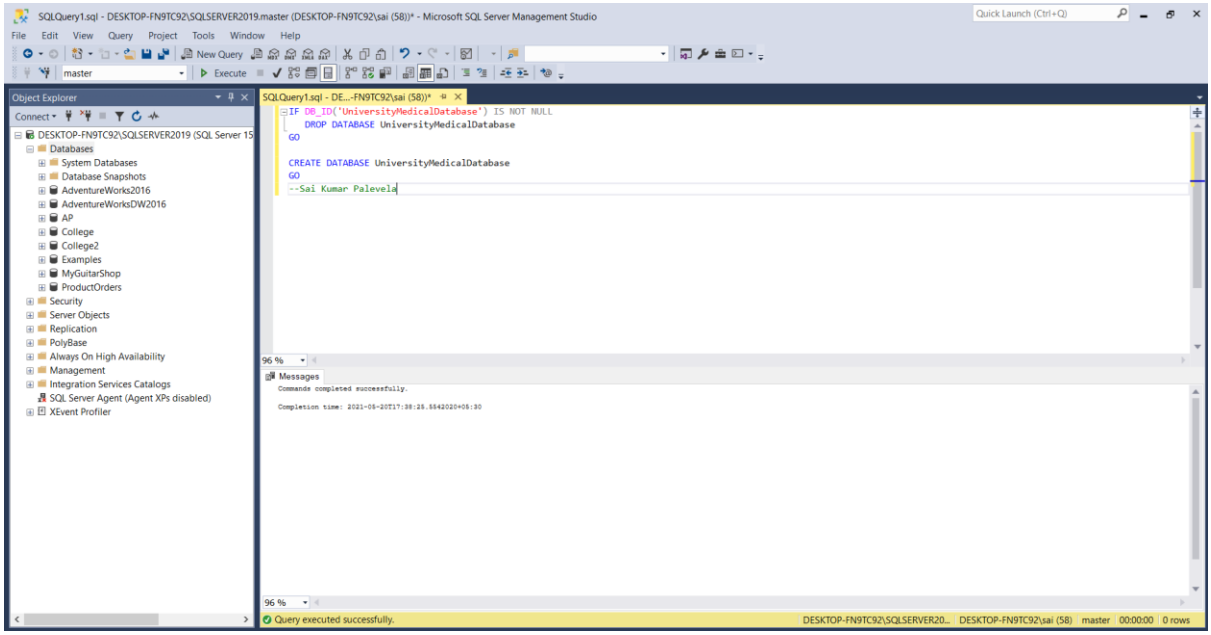
b) Database Normalization:

For a database design to be in 3rd Normal form, it must be in First Normal Form i.e. it must not contain any non-repeating columns and should contain scalar values in all rows, and it must also be in 2nd Normal form (i.e. no partial dependencies) and all the columns must only depend on the primary key (i.e. no transitive dependencies). So here is my database design in 3rd Normal form given below:

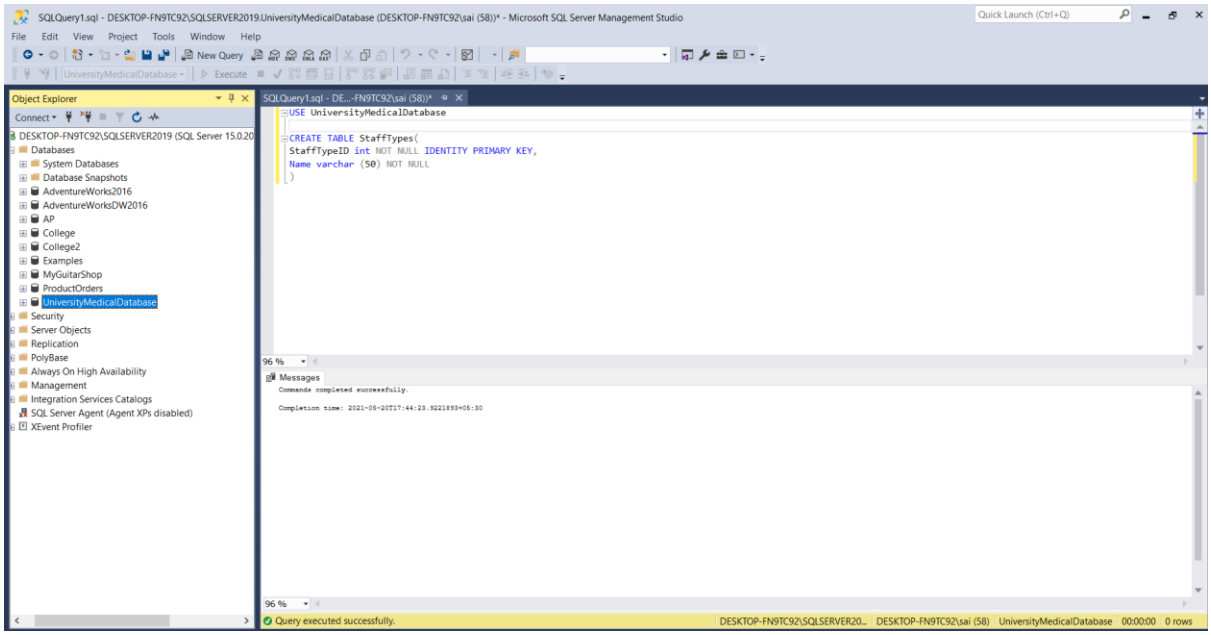


c) Tables Creation

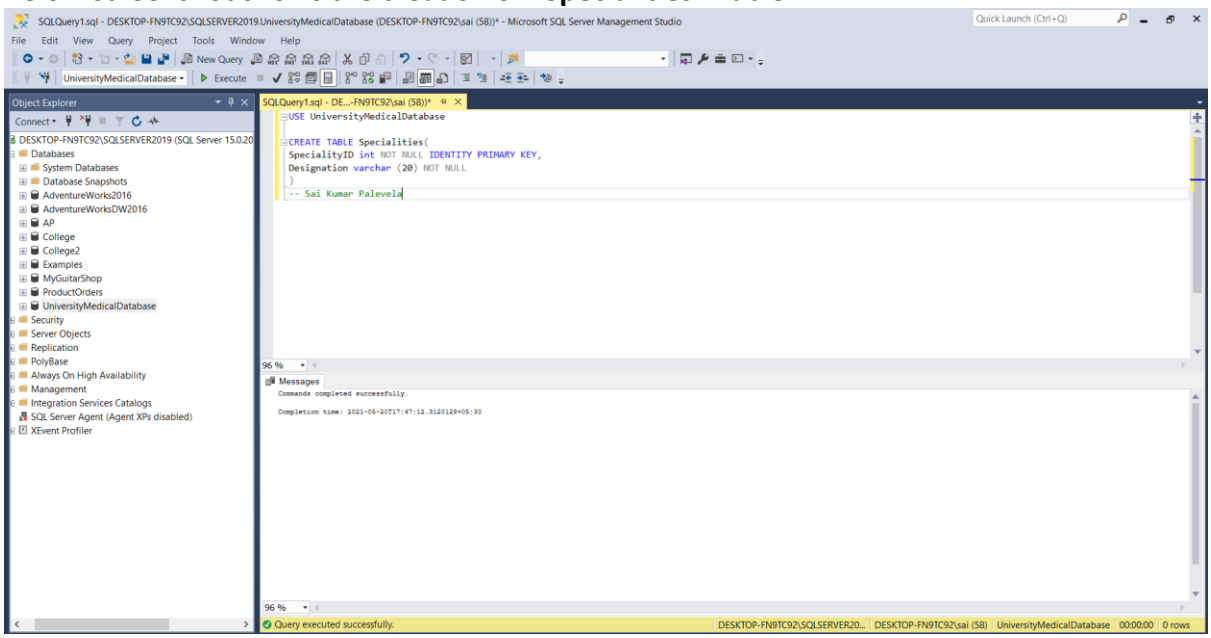
Below screenshots shows the creation of all the tables present in the database



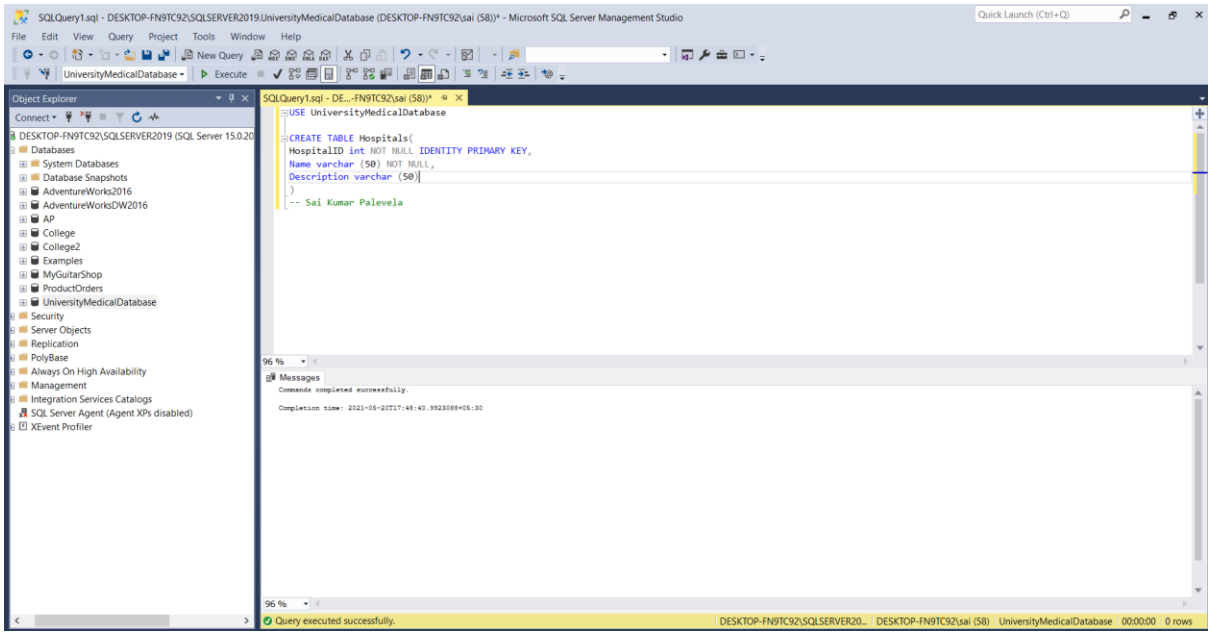
Below screenshot shows the creation of **"StaffTypes"** Table.



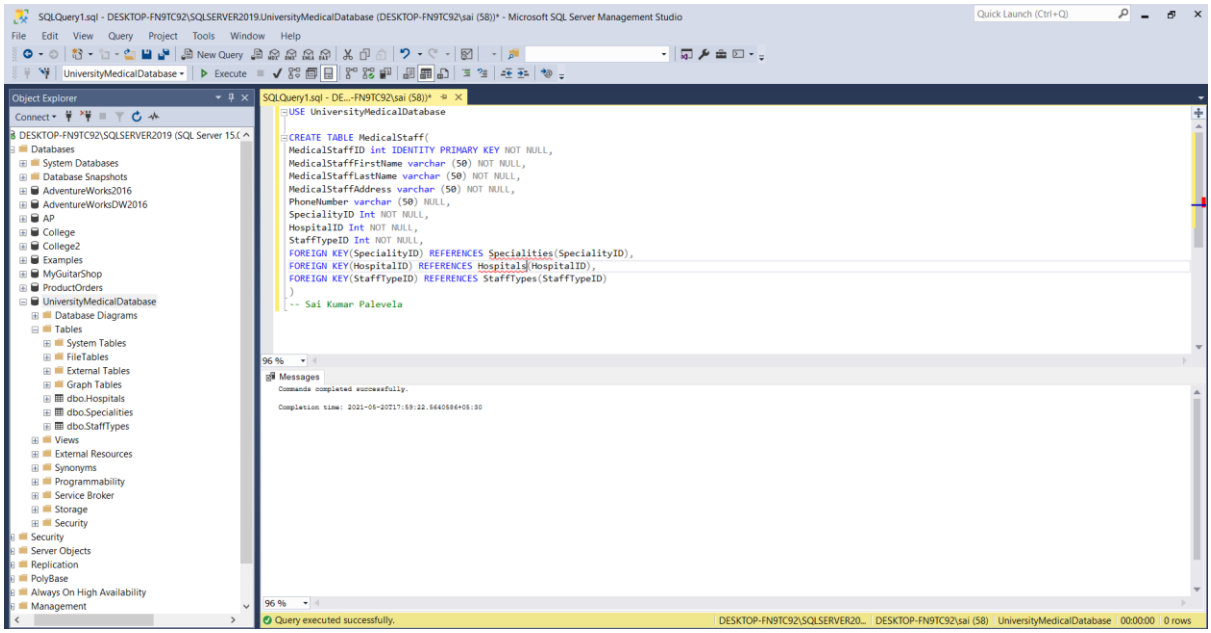
Below screenshot shows the creation of **"Specialities"** Table.



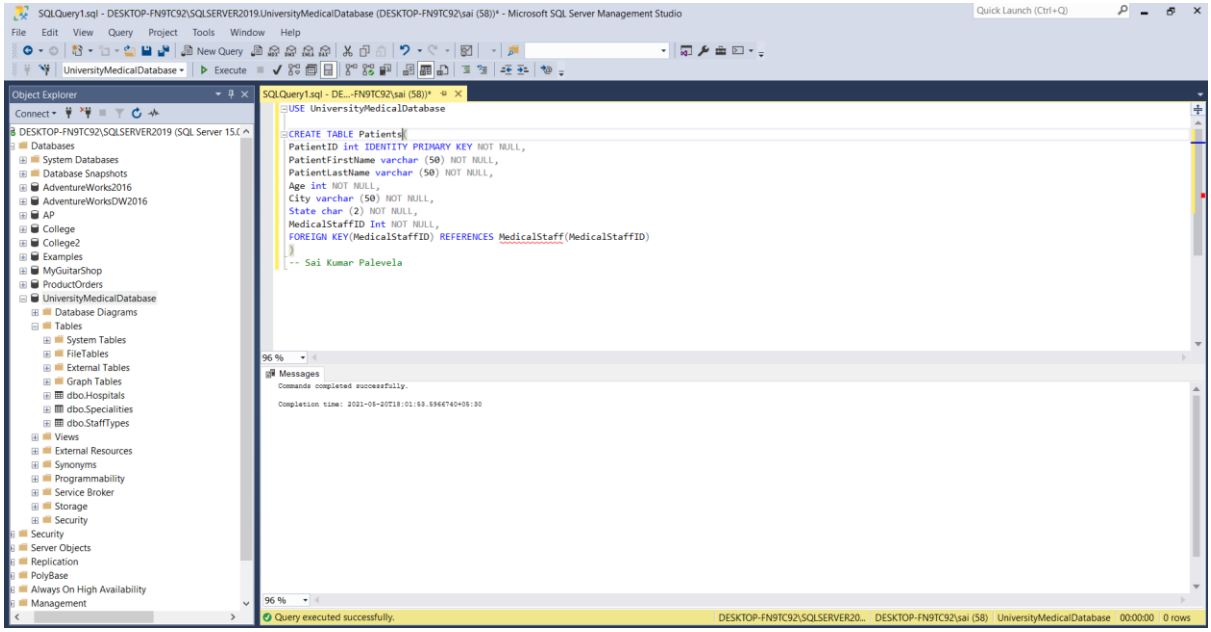
Below screenshot shows the creation of **"Hospitals"** Table.



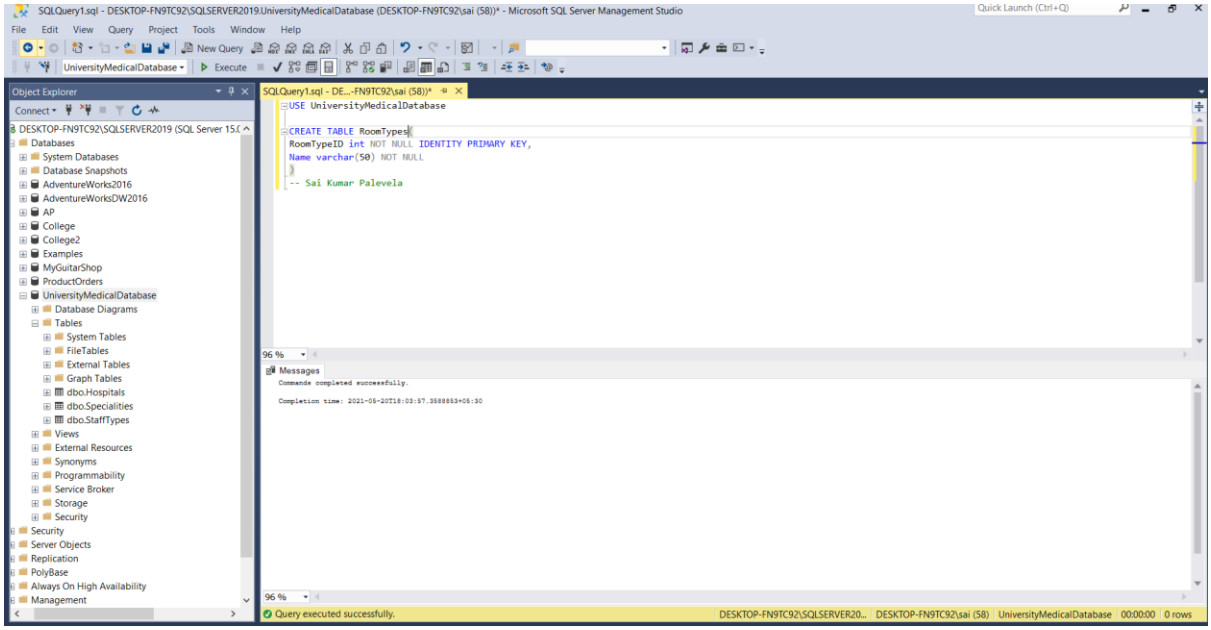
Below screenshot shows the creation of **“MedicalStaff”** Table.



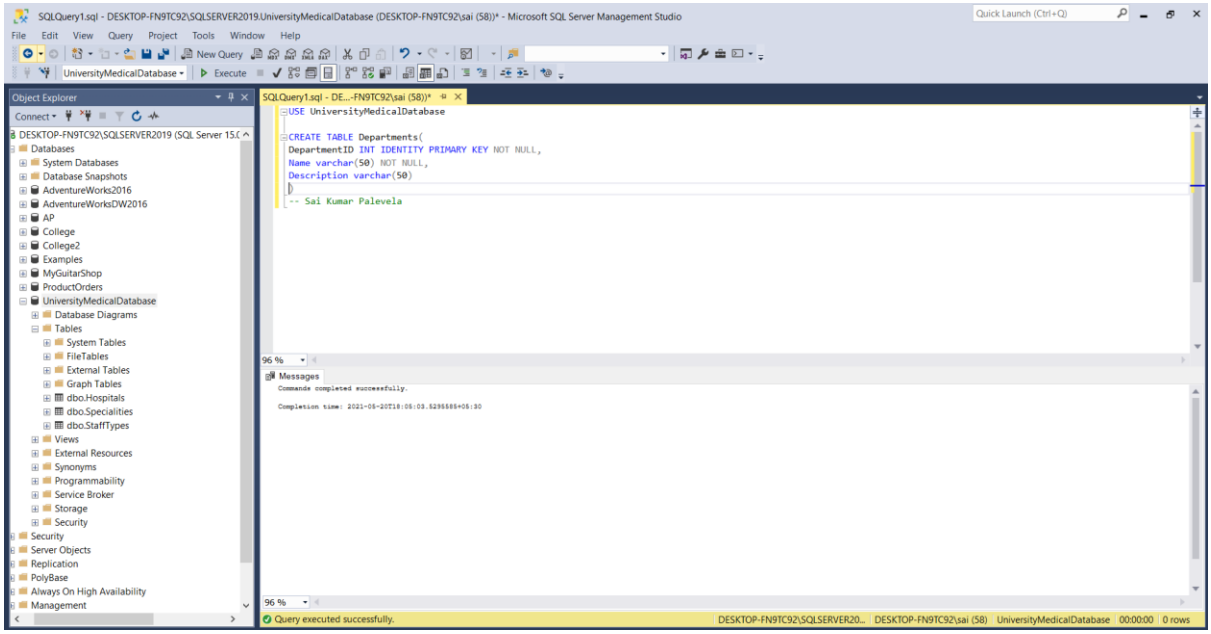
Below screenshot shows the creation of **"Patients"** Table.



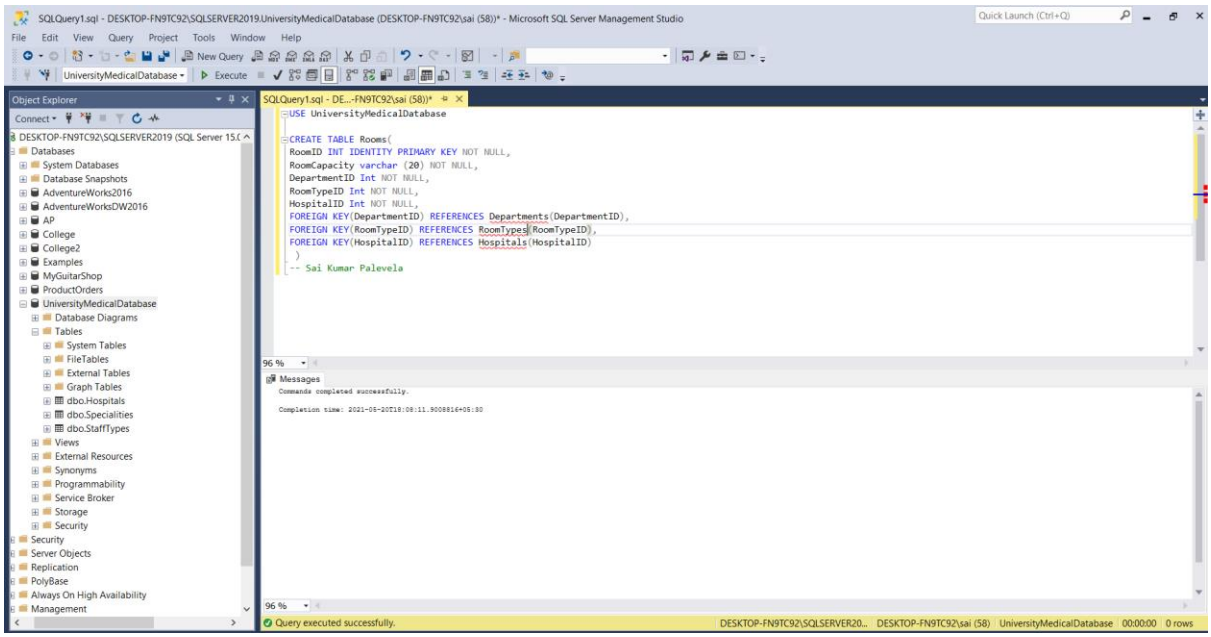
Below screenshot shows the creation of **"RoomTypes"** Table.



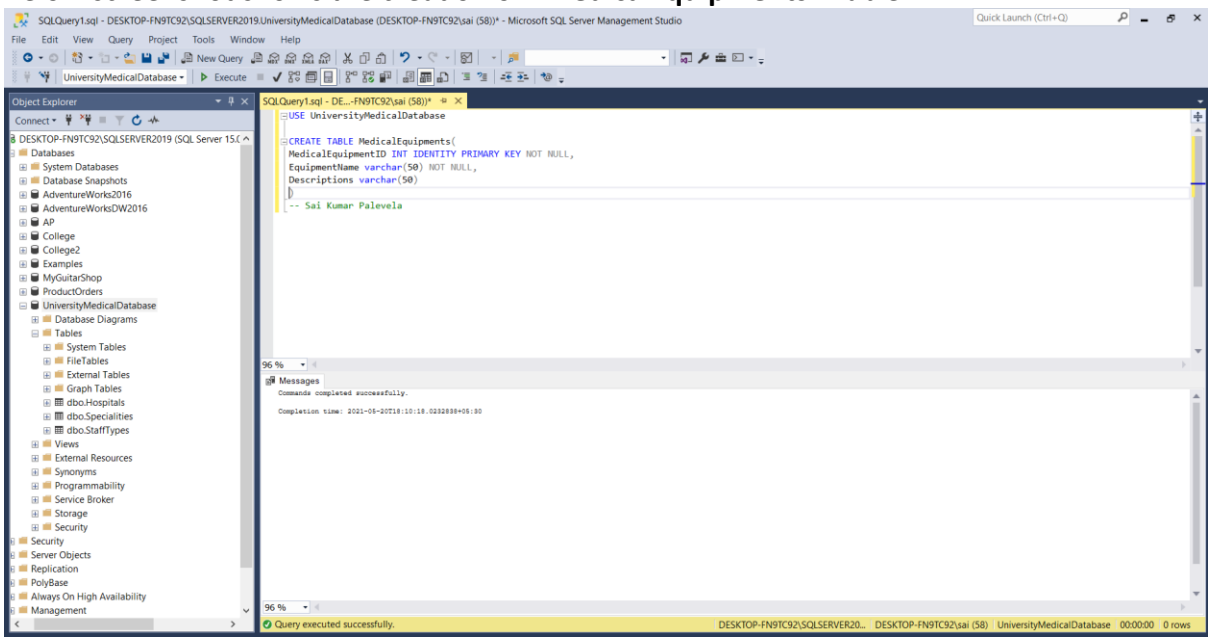
Below screenshot shows the creation of **“Departments”** Table.



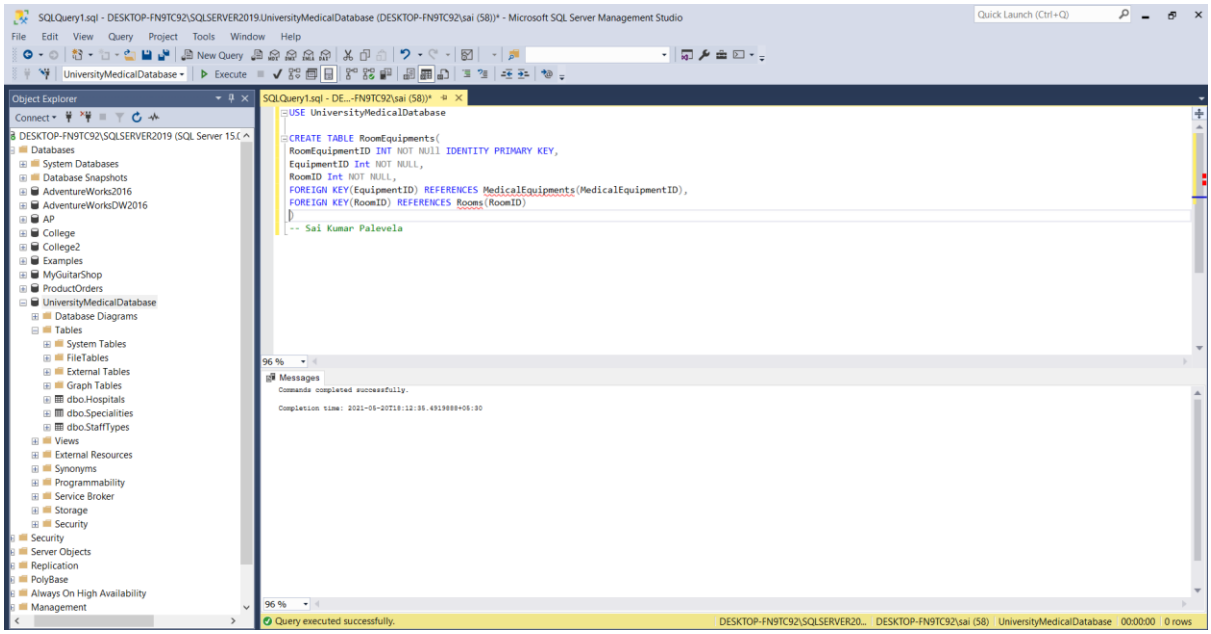
Below screenshot shows the creation of **“Rooms”** Table.



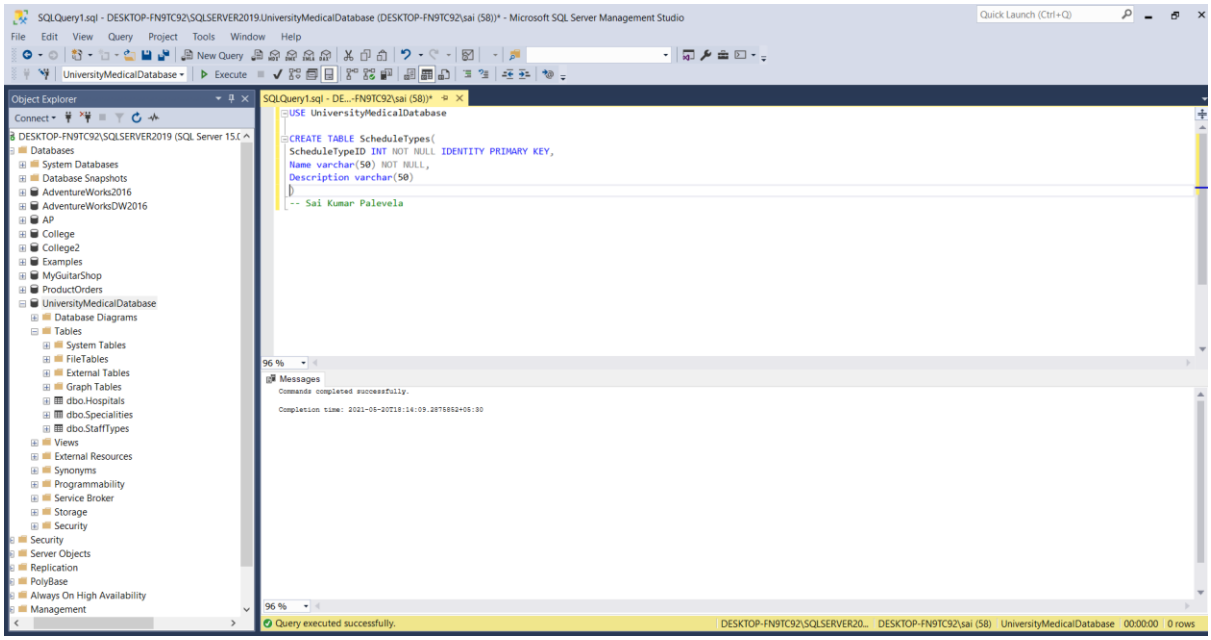
Below screenshot shows the creation of **“MedicalEquipments”** Table.



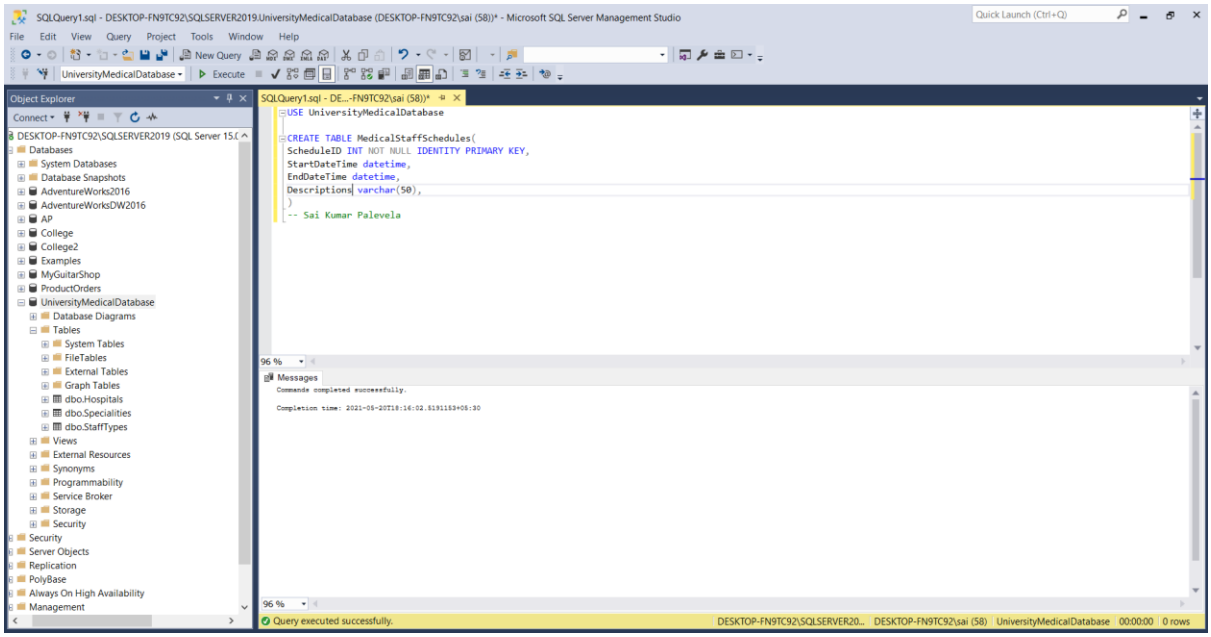
Below screenshot shows the creation of **“RoomEquipments”** Table.



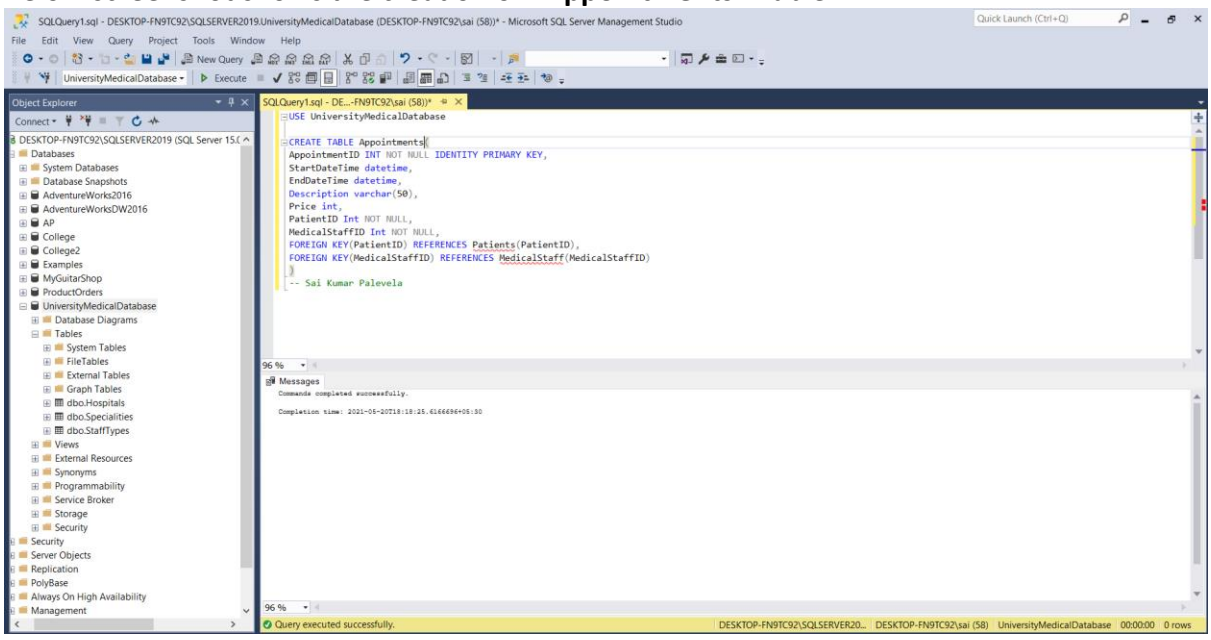
Below screenshot shows the creation of **“ScheduleTypes”** Table.



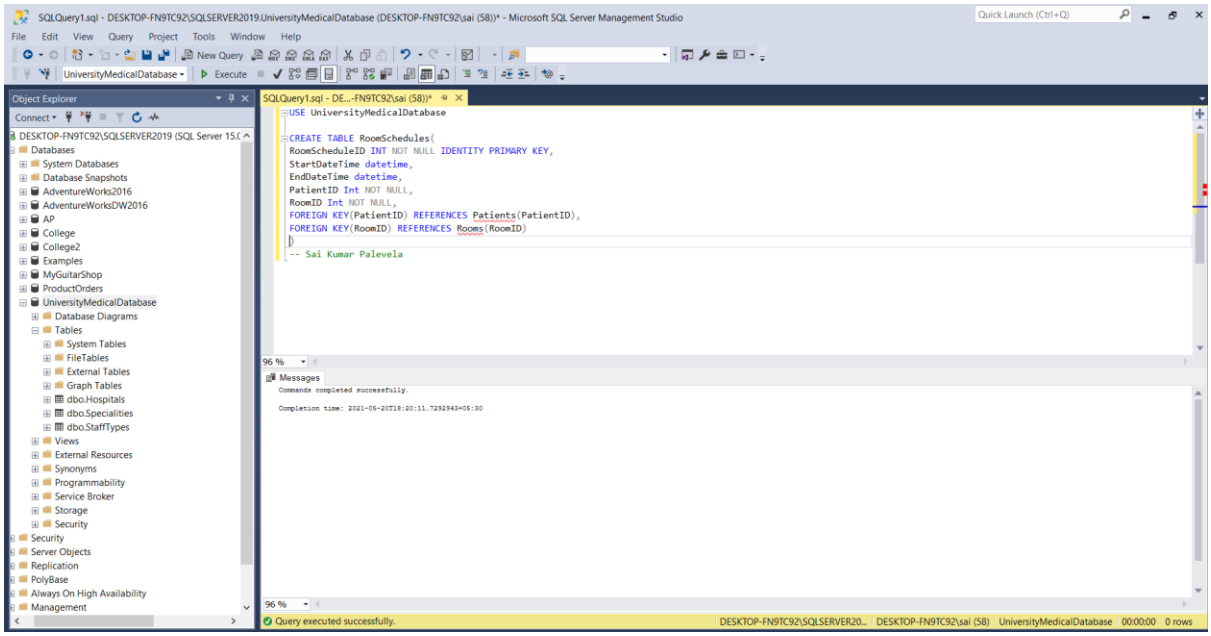
Below screenshot shows the creation of **“MedicalStaffSchedules”** Table.



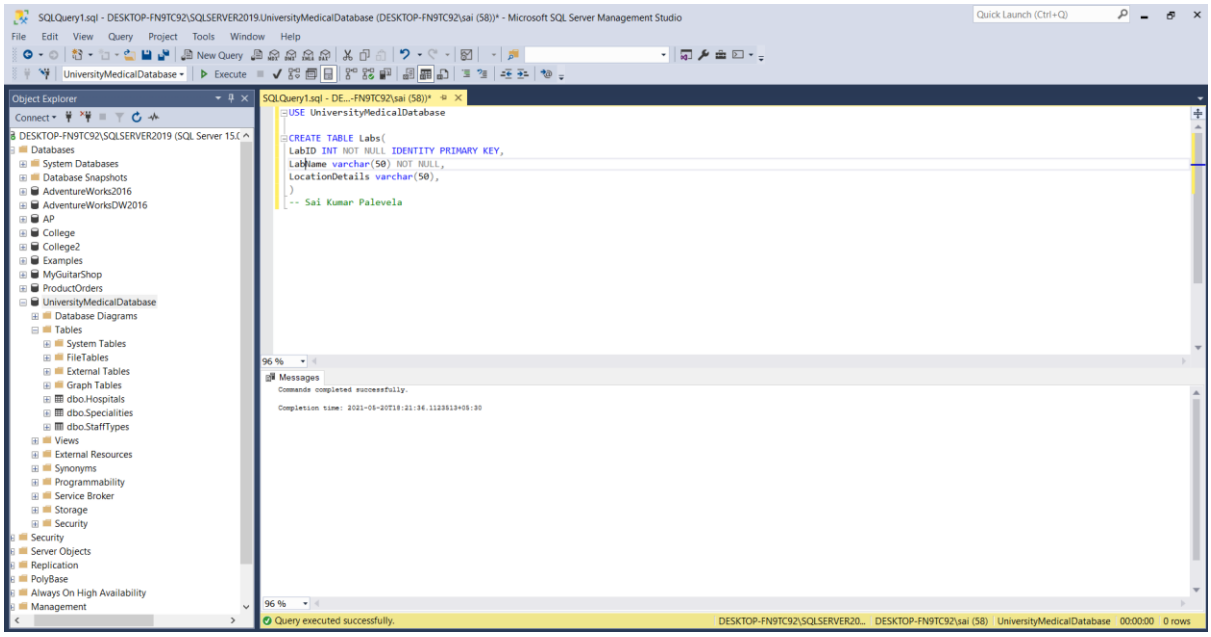
Below screenshot shows the creation of **“Appointments”** Table.



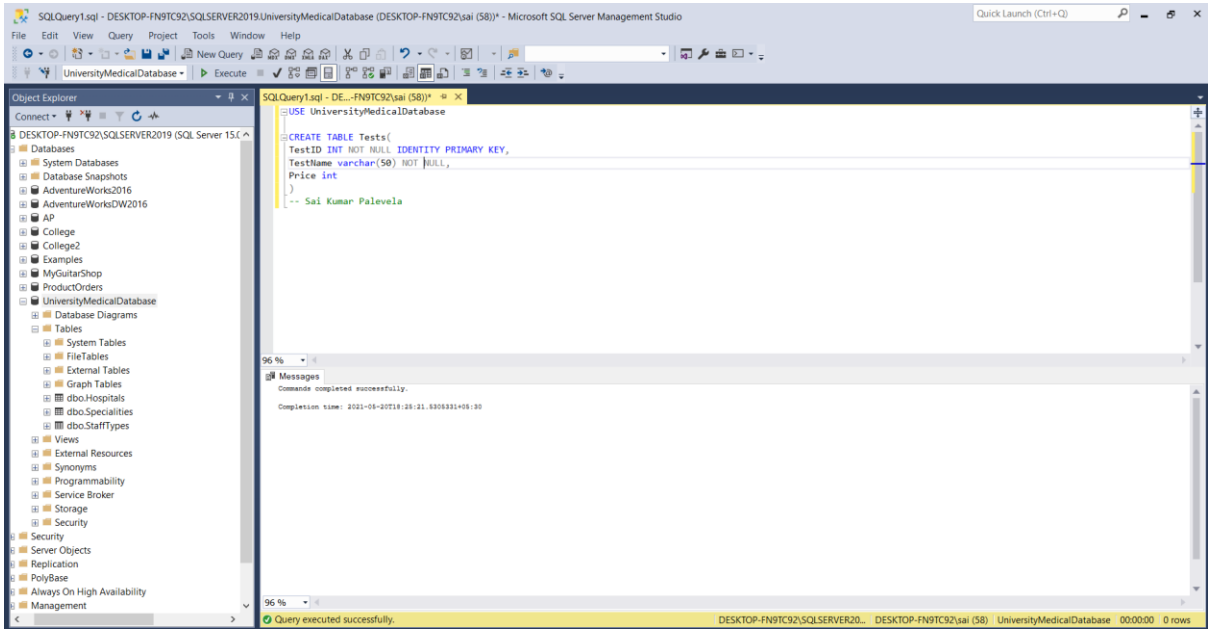
Below screenshot shows the creation of **“RoomSchedules”** Table.



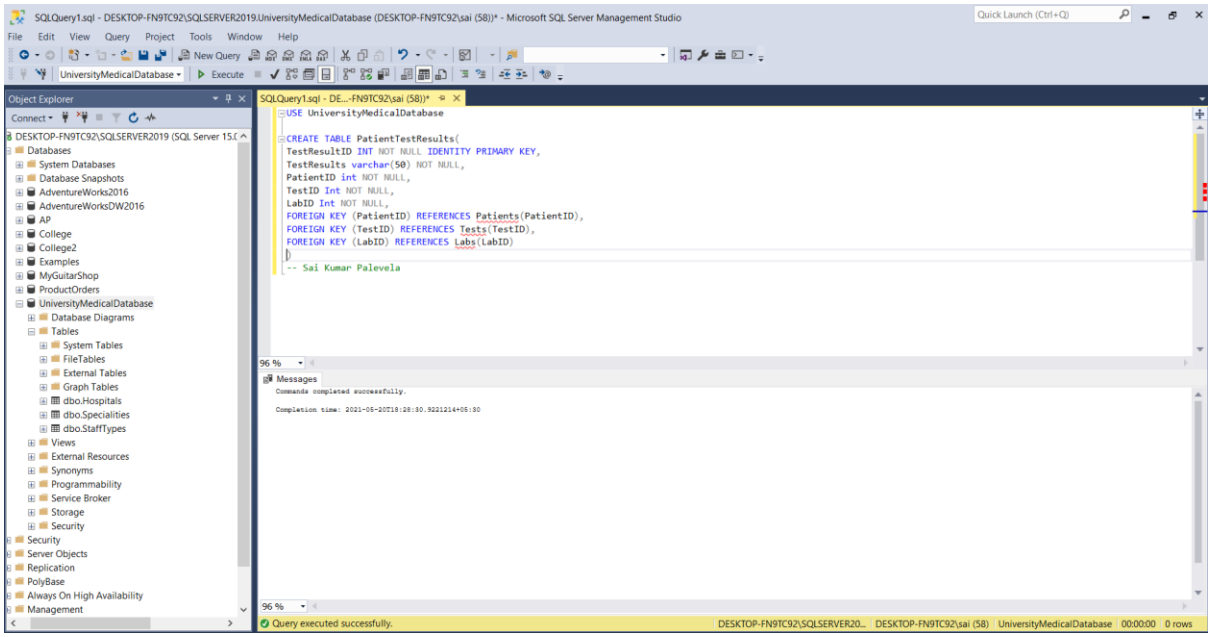
Below screenshot shows the creation of **“Labs”** Table.



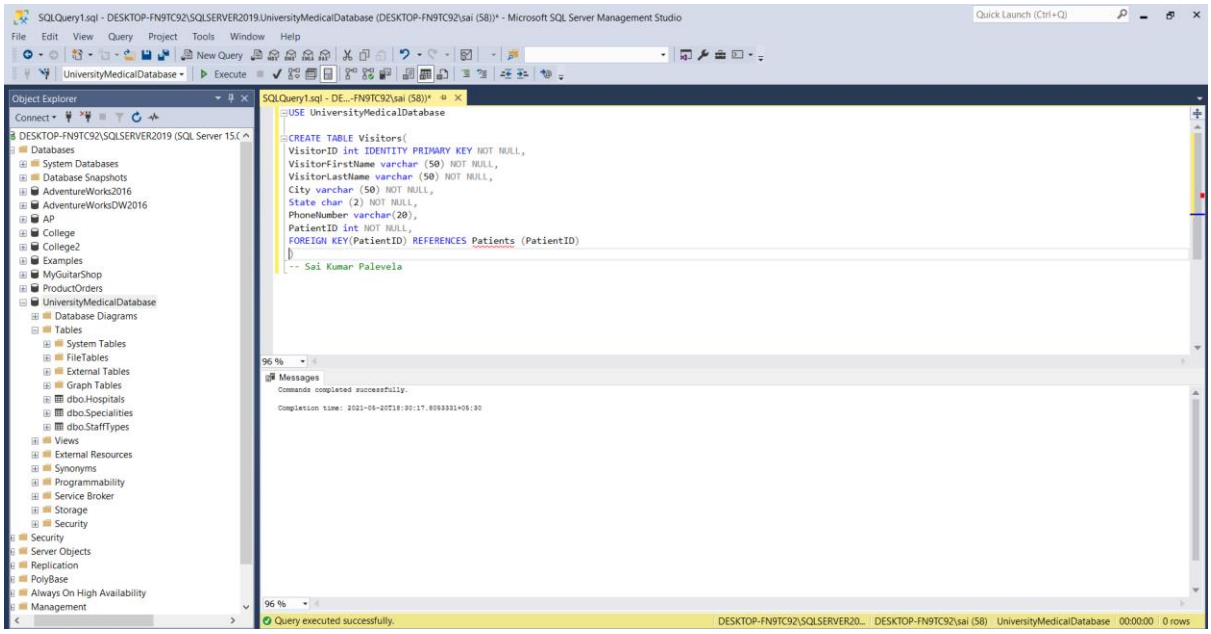
Below screenshot shows the creation of **“Tests”** Table.



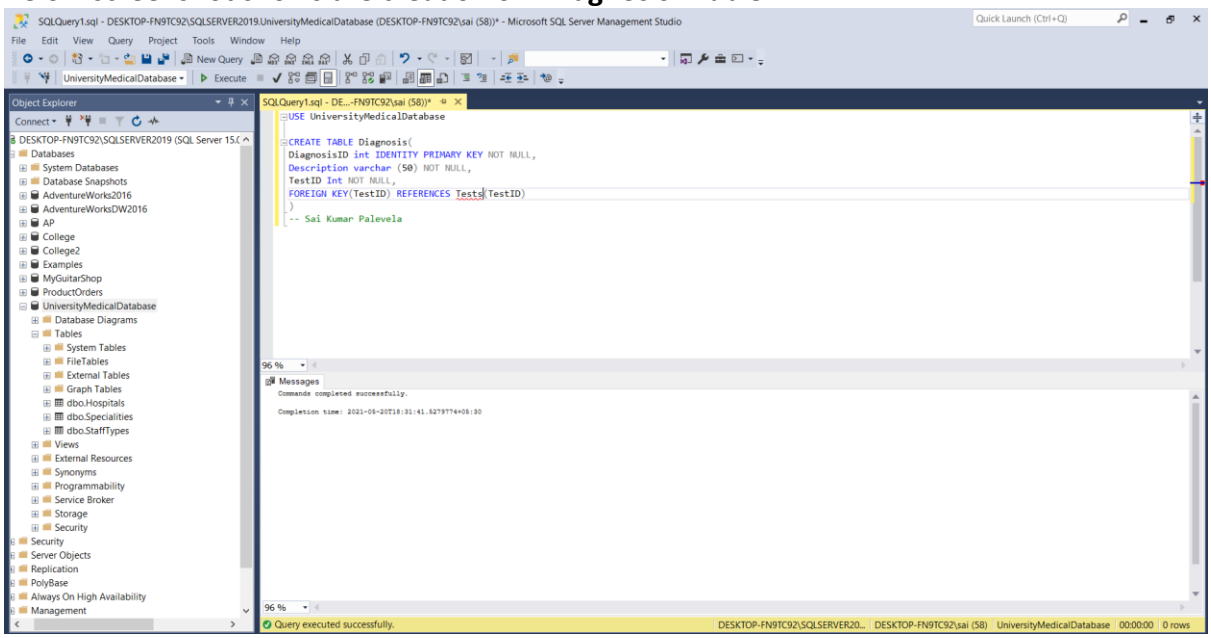
Below screenshot shows the creation of **“PatientTestResults”** Table.



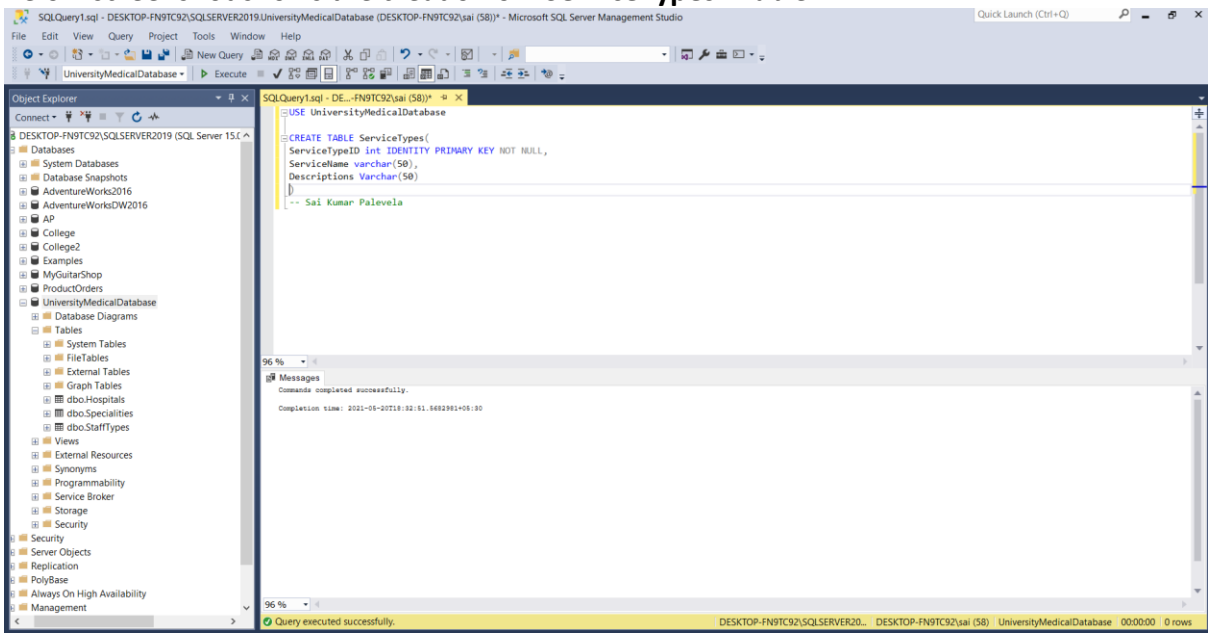
Below screenshot shows the creation of **“Visitors”** Table.



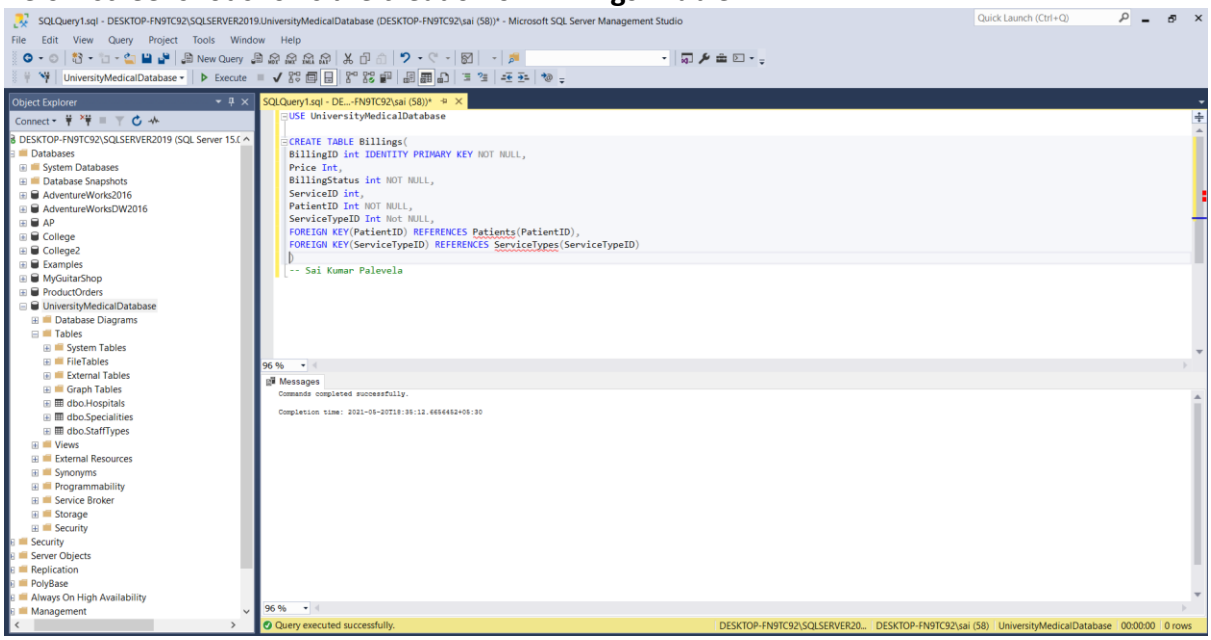
Below screenshot shows the creation of **“Diagnosis”** Table.



Below screenshot shows the creation of **“ServiceTypes”** Table.



Below screenshot shows the creation of **“Billings”** Table.

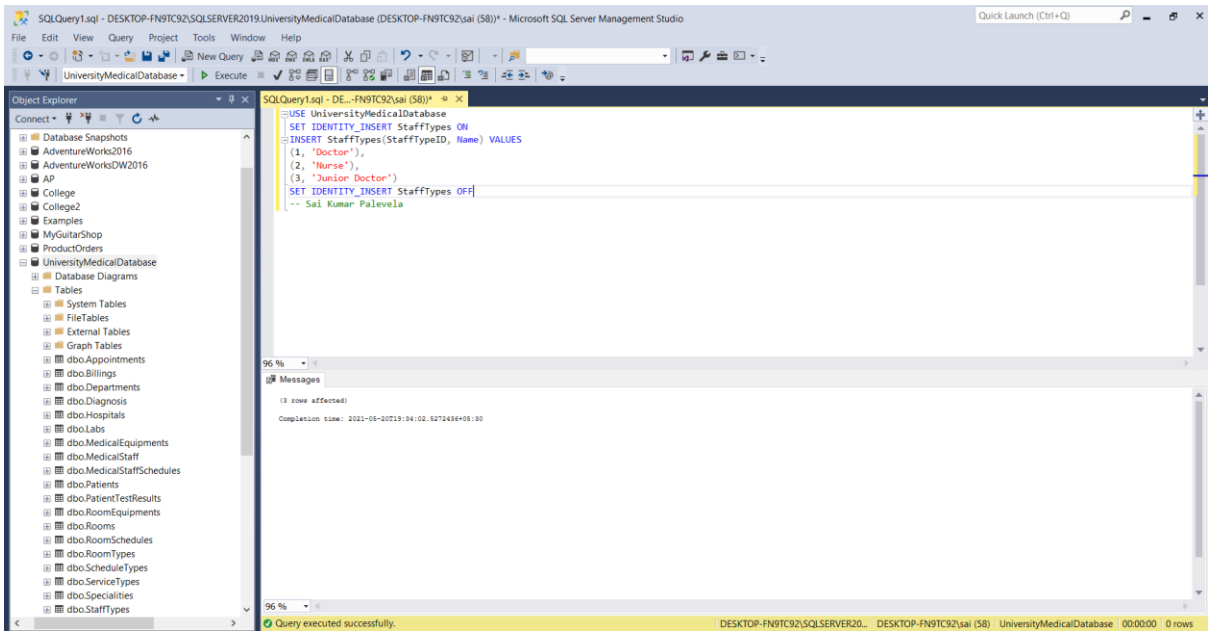


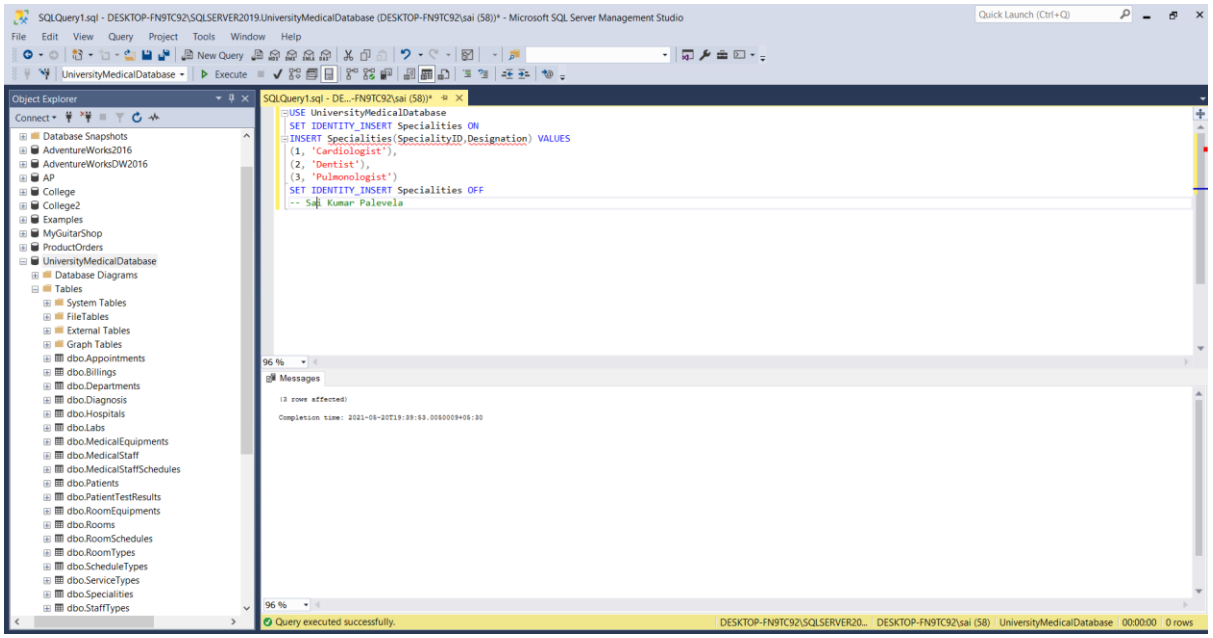
4) Testing Phase:

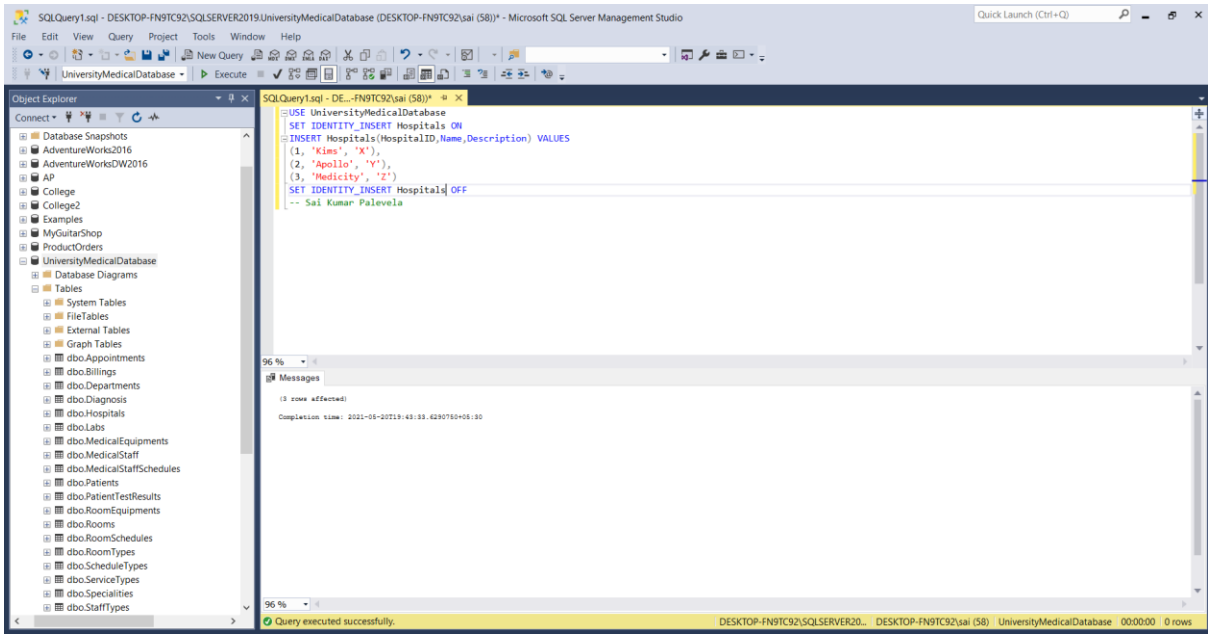
a) Populating the database with Test information:

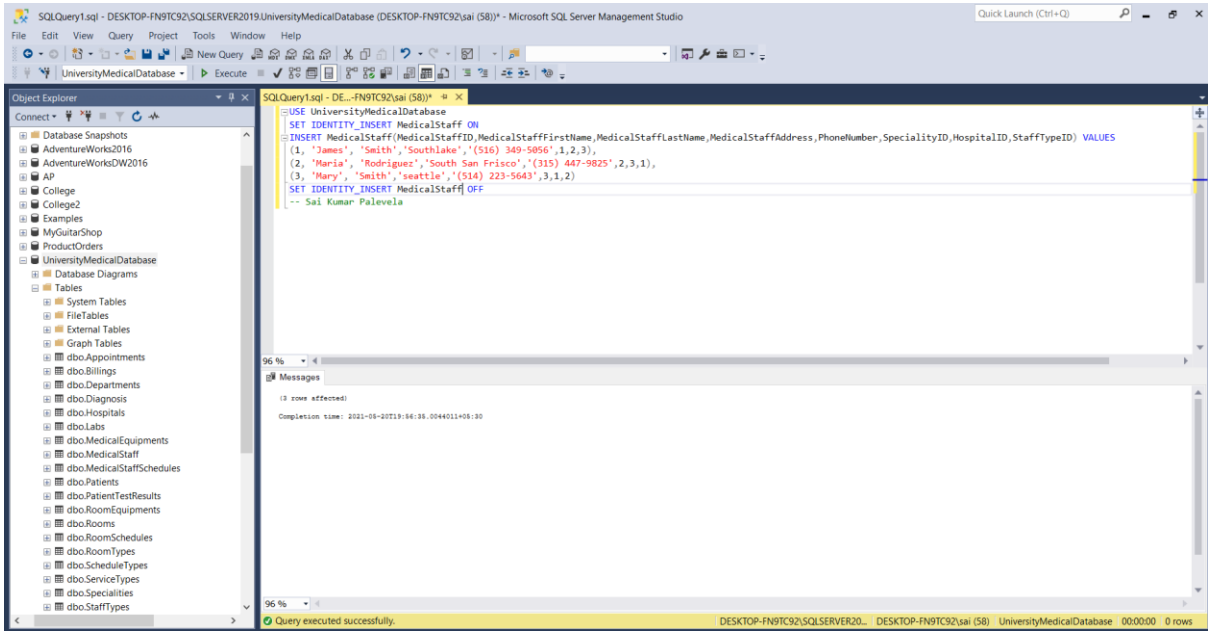
Here I have created all the tables with the constraints and relations mentioned above in the implementation phase.

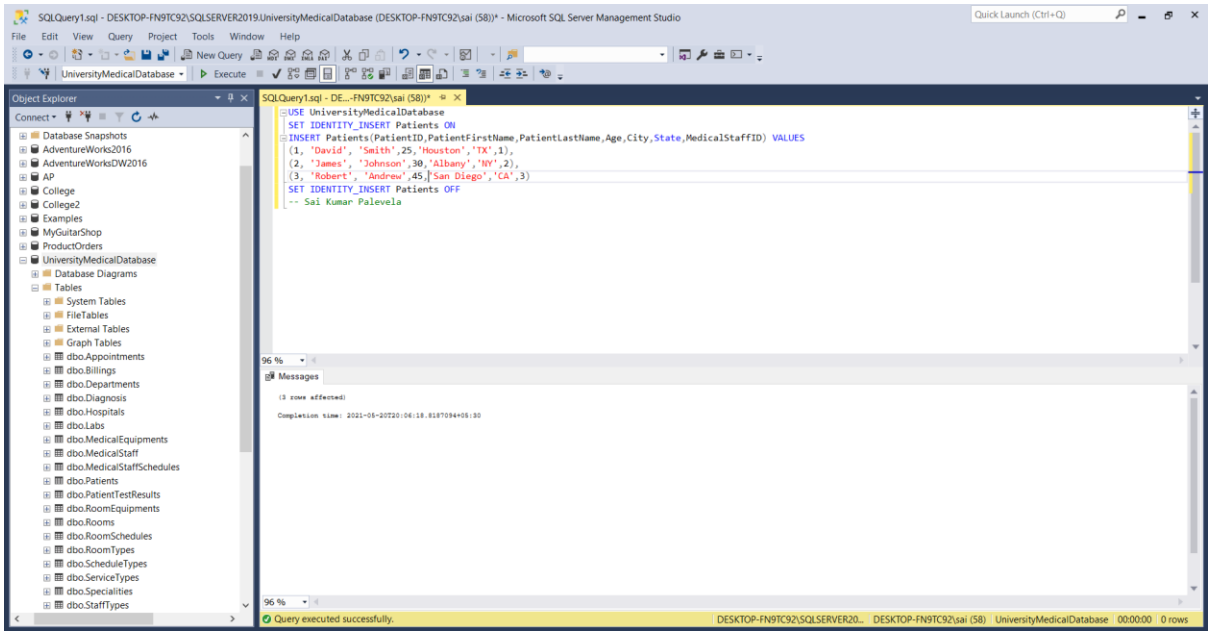
Here is a screenshot of showing all the tables that I have created in my database.

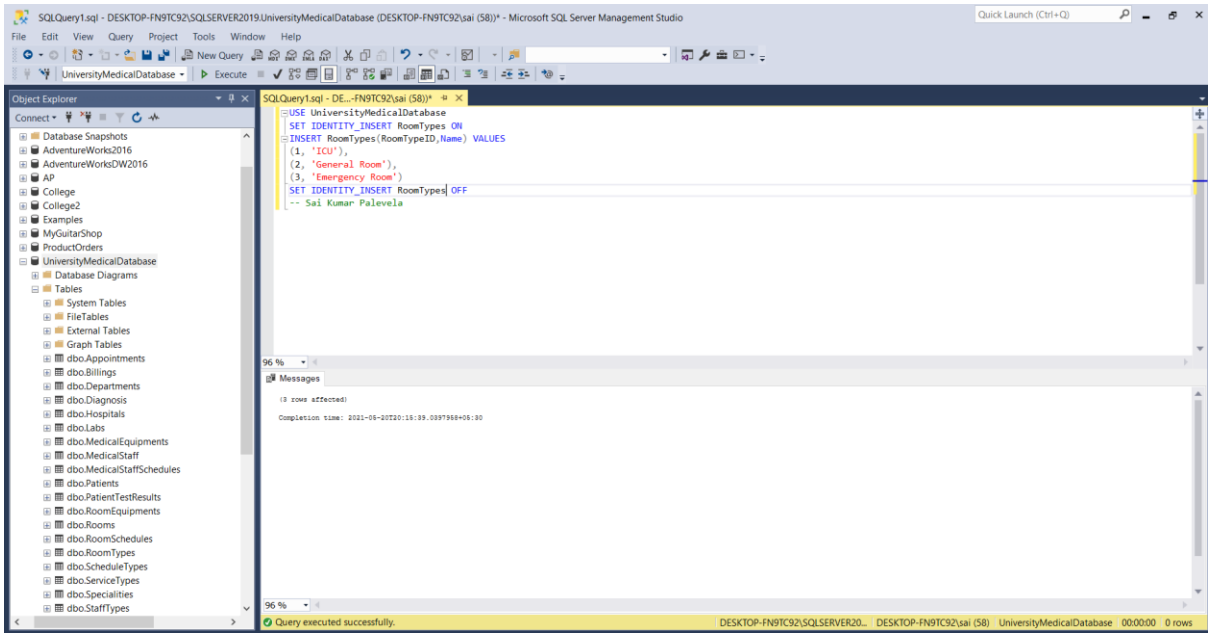

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT StaffTypes ON
INSERT StaffTypes(StaffTypeID, Name) VALUES
(1, 'Doctor'),
(2, 'Nurse'),
(3, 'Junior Doctor')
SET IDENTITY_INSERT StaffTypes OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T19:34:02.5272436+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

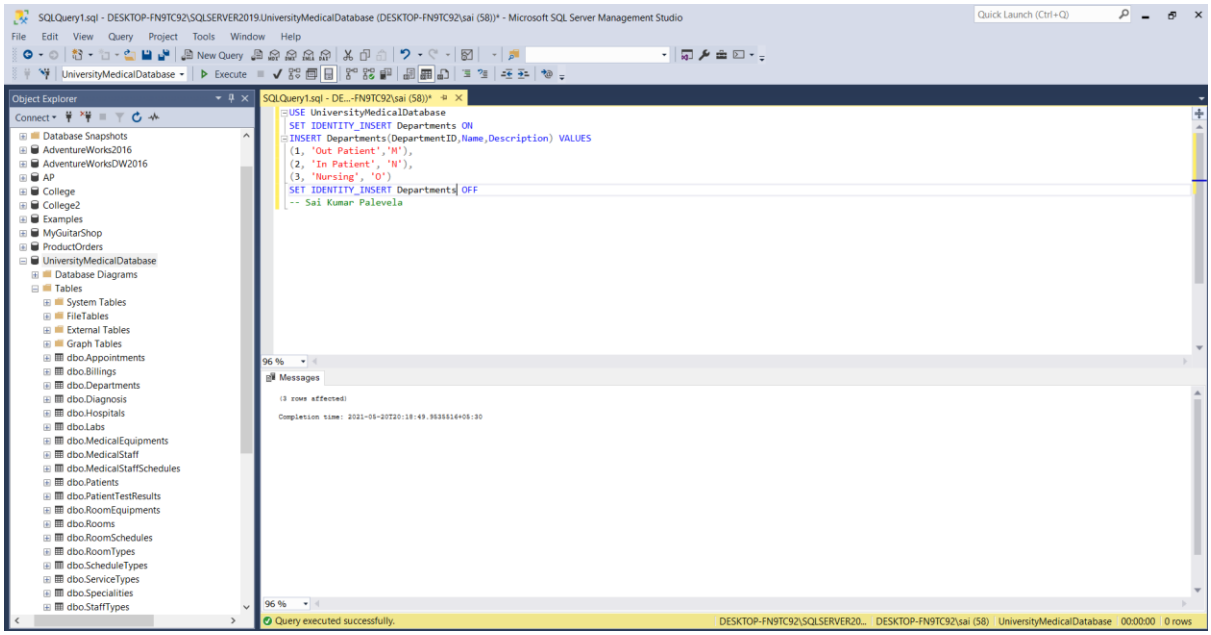

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Specialities ON
INSERT Specialities(SpecialityID,Designation) VALUES
(1, 'Cardiologist'),
(2, 'Dentist'),
(3, 'Pulmonologist')
SET IDENTITY_INSERT Specialities OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T19:39:53.0050009+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

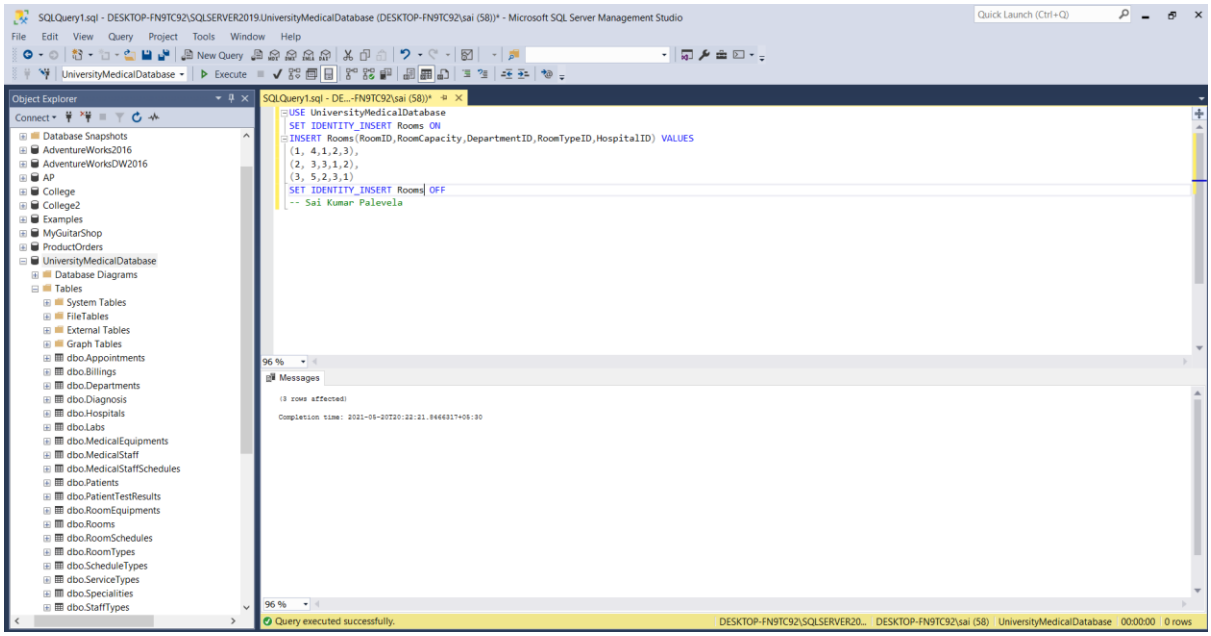

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Hospitals ON
INSERT Hospitals(HospitalID,Name,Description) VALUES
(1, 'Kims', 'X'),
(2, 'Apollo', 'Y'),
(3, 'Medicity', 'Z')
SET IDENTITY_INSERT Hospitals OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T19:43:33.6290750+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

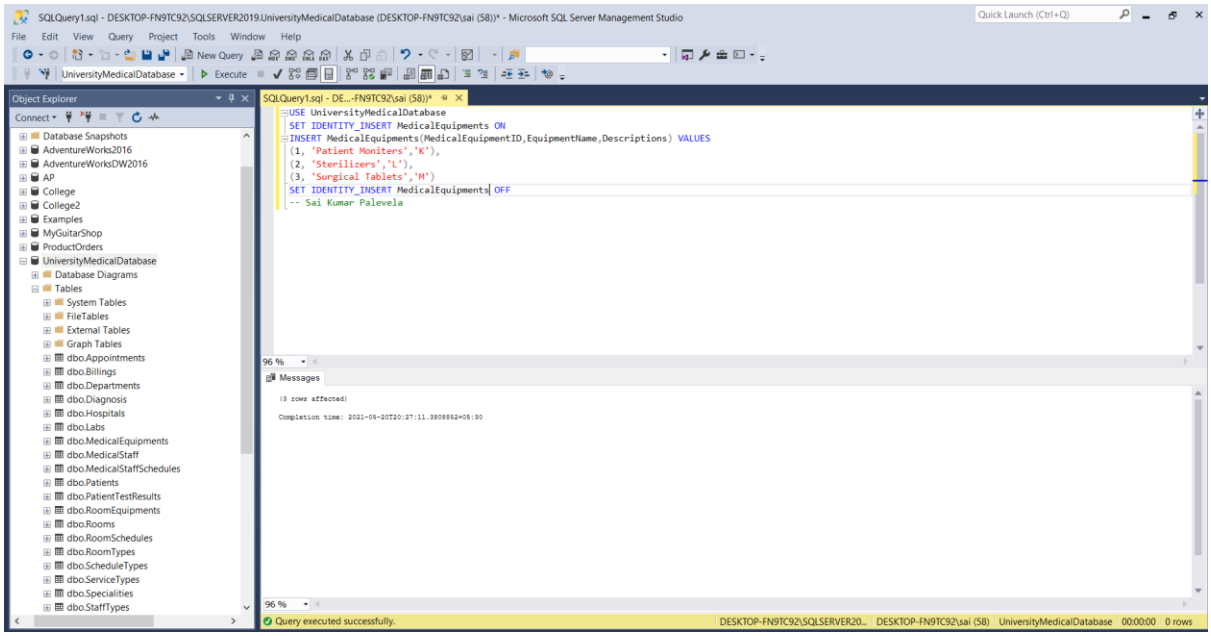

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT MedicalStaff ON
INSERT MedicalStaff(MedicalStaffID,MedicalStaffFirstName,MedicalStaffLastName,MedicalStaffAddress,PhoneNumber,SpecialityID,HospitalID,StaffTypeID) VALUES
(1, 'James', 'Smith','Southlake','(516) 349-5056',1,2,3),
(2, 'Maria', 'Rodriguez','South San Frisco','(315) 447-9825',2,3,1),
(3, 'Mary', 'Smith','seattle','(514) 223-5643',3,1,2)
SET IDENTITY_INSERT MedicalStaff OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T19:56:36.0044011+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

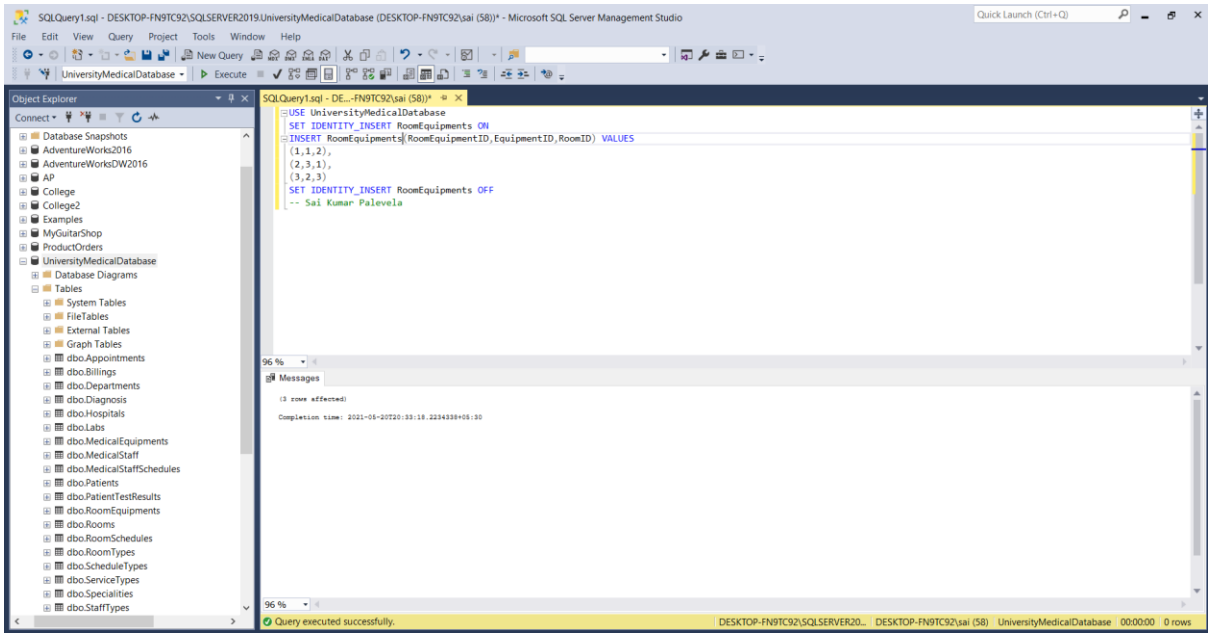

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Patients ON
INSERT Patients(PatientID,PatientFirstName,PatientLastName,Age,City,State,MedicalStaffID) VALUES
(1, 'David', 'Smith',25,'Houston','TX',1),
(2, 'James', 'Johnson',30,'Albany','NY',2),
(3, 'Robert', 'Andrew',45,'San Diego','CA',3)
SET IDENTITY_INSERT Patients OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:06:18.8187094+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

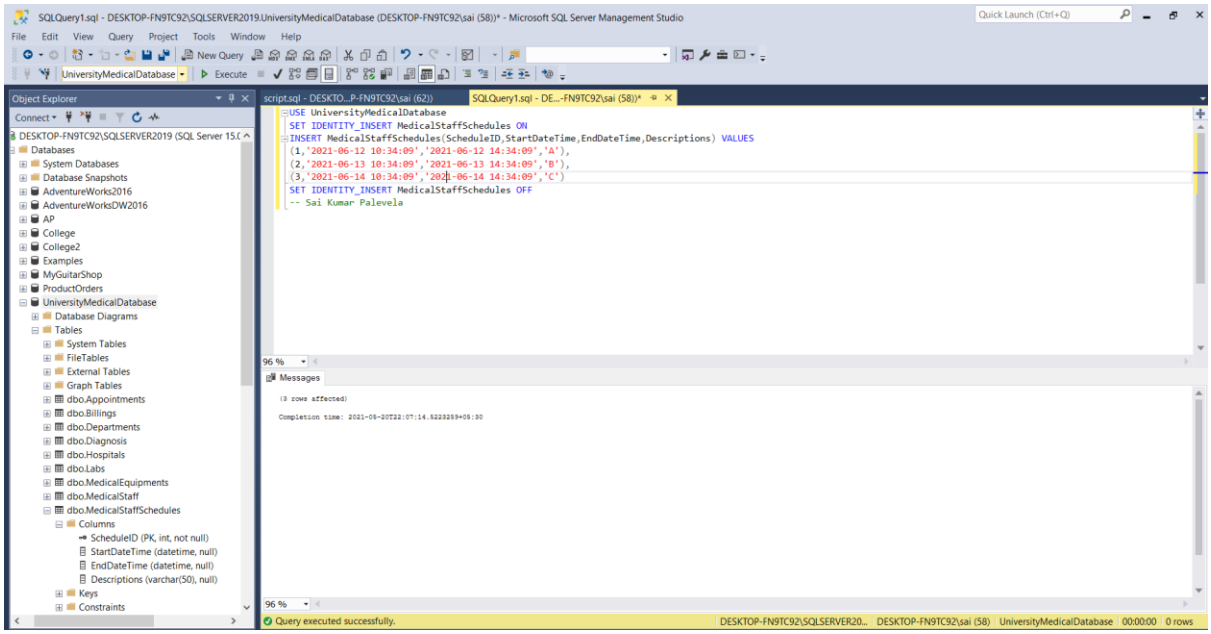

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT RoomTypes ON
INSERT RoomTypes(RoomTypeID,Name) VALUES
(1, 'ICU'),
(2, 'General Room'),
(3, 'Emergency Room')
SET IDENTITY_INSERT RoomTypes OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:15:39.0397968+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

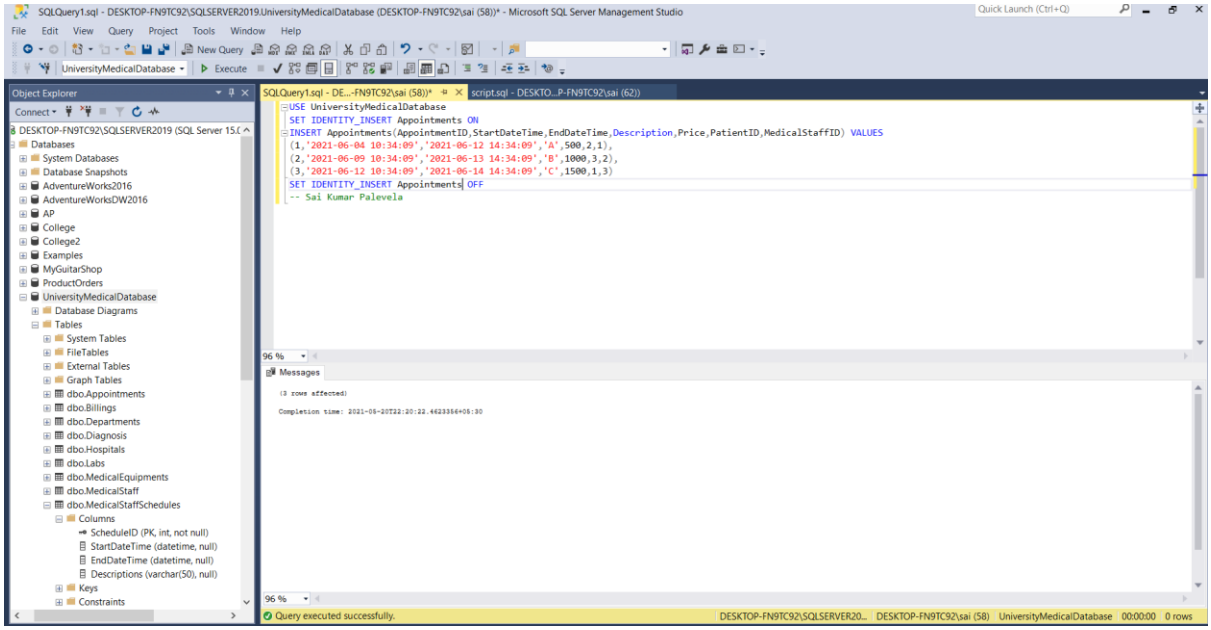

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Departments ON
INSERT Departments(DepartmentID,Name,Description) VALUES
(1, 'Out Patient','M'),
(2, 'In Patient', 'N'),
(3, 'Nursing', 'O')
SET IDENTITY_INSERT Departments OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:18:49.9535514+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

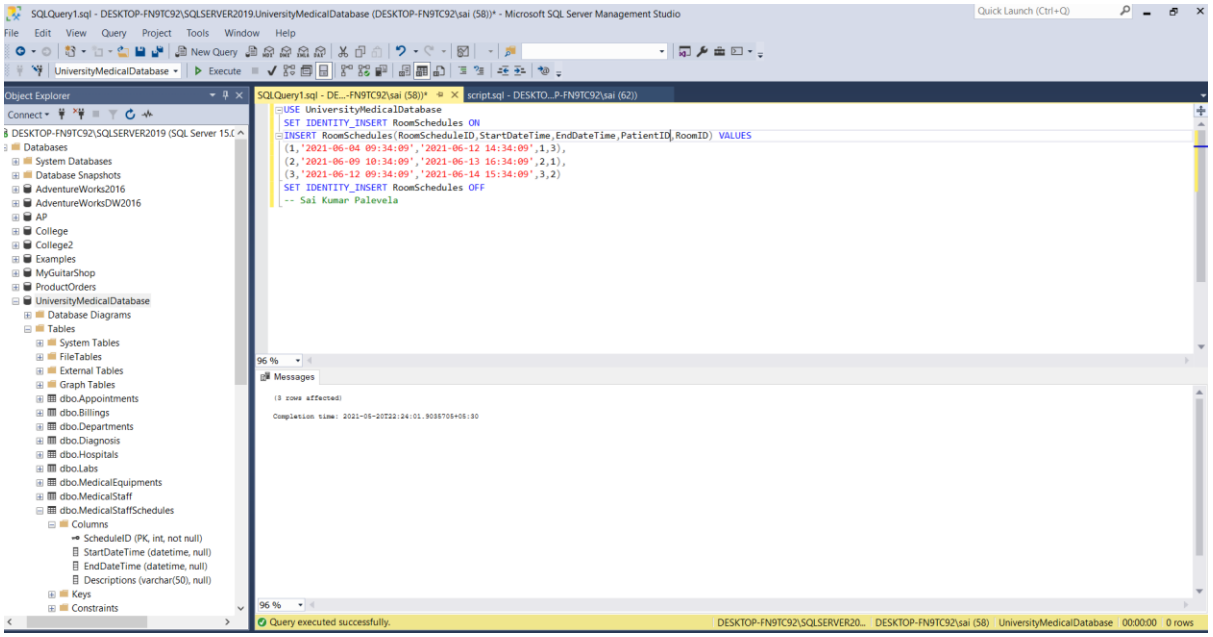

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Rooms ON
INSERT Rooms(RoomID,RoomCapacity,DepartmentID,RoomTypeID,HospitalID) VALUES
(1, 4,1,2,3),
(2, 3,3,1,2),
(3, 5,2,3,1)
SET IDENTITY_INSERT Rooms OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:22:21.8466317+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

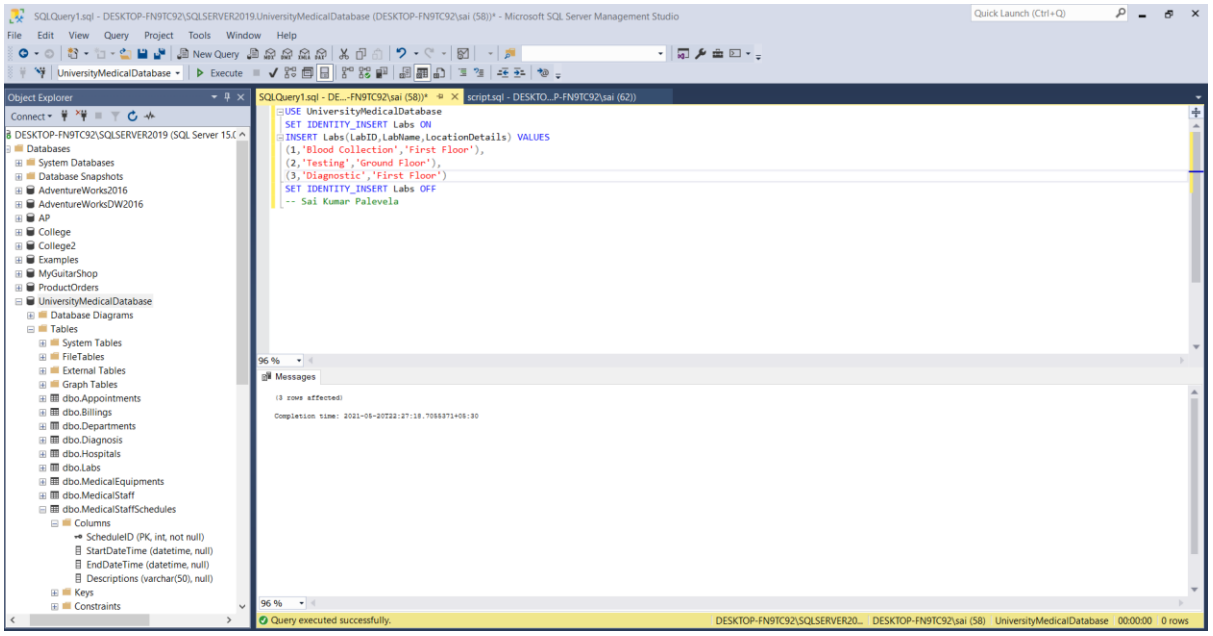

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT MedicalEquipments ON
INSERT MedicalEquipments(MedicalEquipmentID,EquipmentName,Descriptions) VALUES
(1, 'Patient Moniters','K'),
(2, 'Sterilizers','L'),
(3, 'Surgical Tablets','M')
SET IDENTITY_INSERT MedicalEquipments OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:27:11.3808862+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

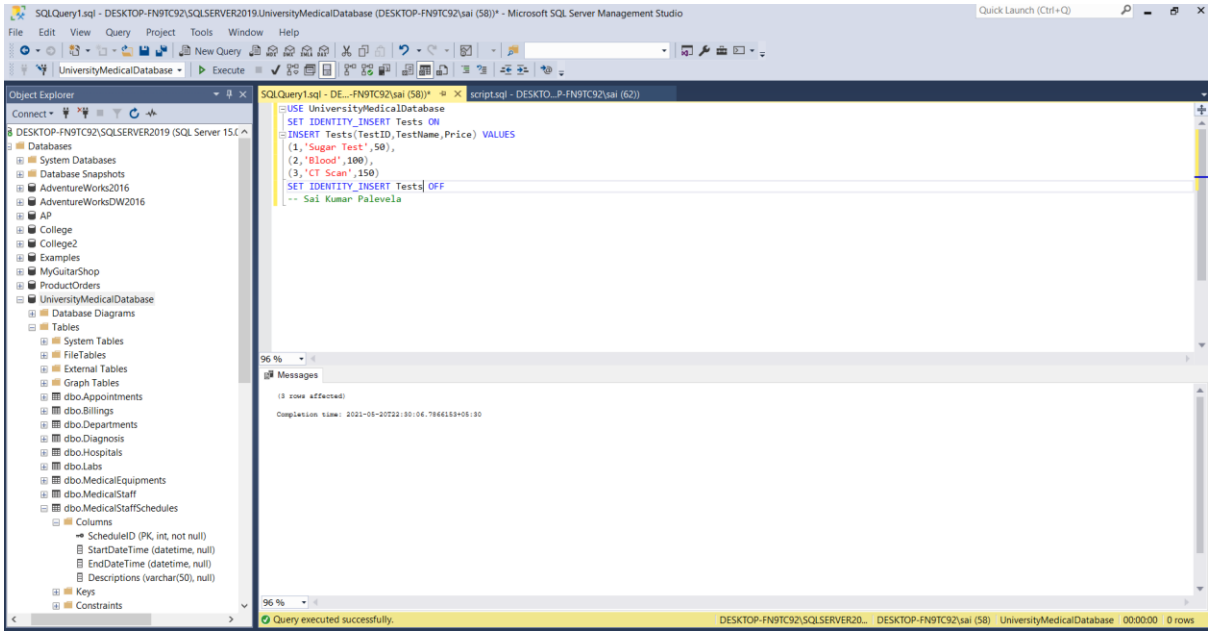

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
dbo.RoomTypes
dbo.ScheduleTypes
dbo.ServiceTypes
dbo.Specialities
dbo.StaffTypes
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT RoomEquipments ON
INSERT RoomEquipments(RoomEquipmentID,EquipmentID,RoomID) VALUES
(1,1,2),
(2,3,1),
(3,2,3)
SET IDENTITY_INSERT RoomEquipments OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T20:33:18.2234338+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

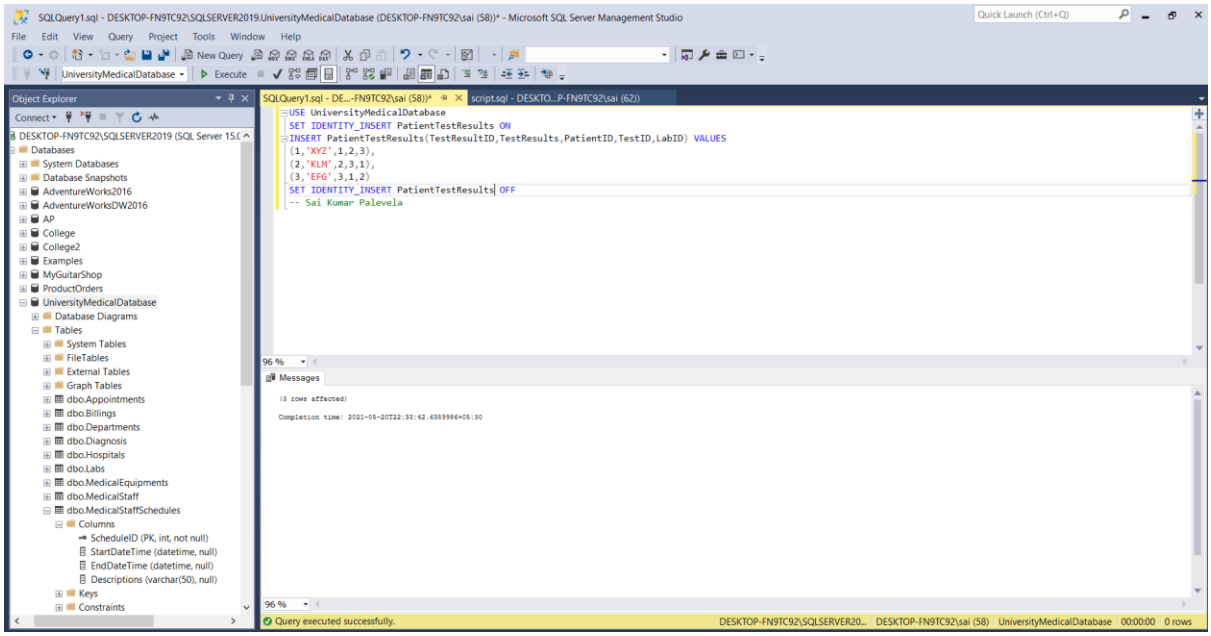

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
script.sql - DESKTO...P-FN9TC92\sai (62))
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
USE UniversityMedicalDatabase
SET IDENTITY_INSERT MedicalStaffSchedules ON
INSERT MedicalStaffSchedules(ScheduleID,StartDateTime,EndDateTime,Descriptions) VALUES
(1,'2021-06-12 10:34:09','2021-06-12 14:34:09','A'),
(2,'2021-06-13 10:34:09','2021-06-13 14:34:09','B'),
(3,'2021-06-14 10:34:09','2021-06-14 14:34:09','C')
SET IDENTITY_INSERT MedicalStaffSchedules OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T22:07:14.5229269+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

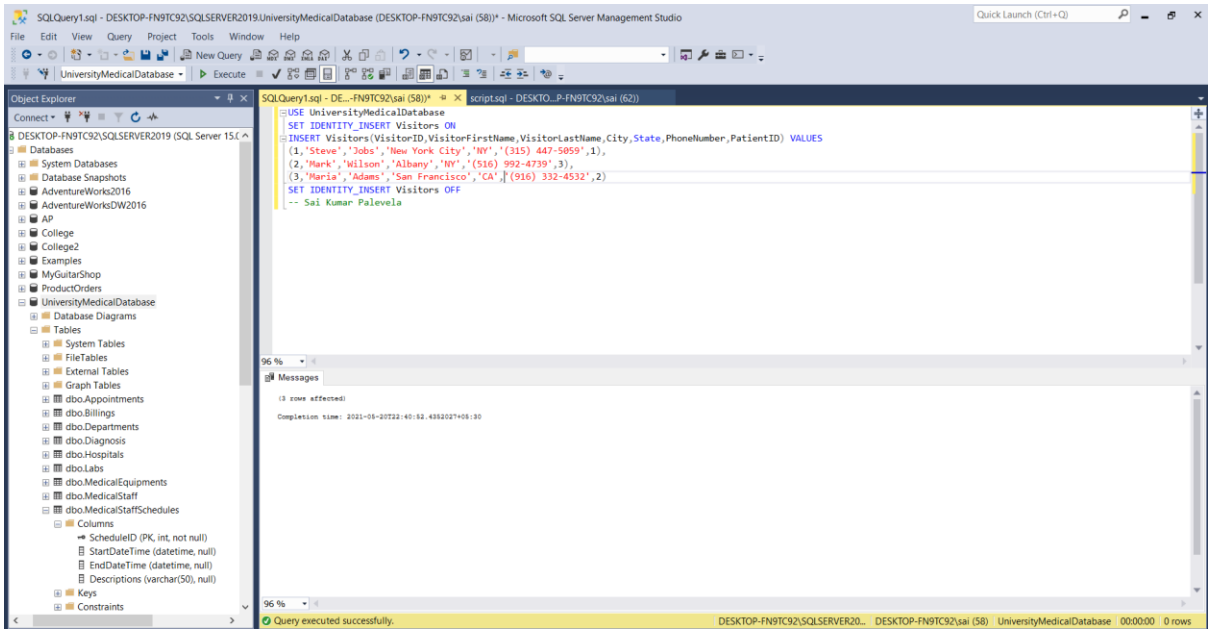

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Appointments ON
INSERT Appointments(AppointmentID,StartDateTime,EndDateTime,Description,Price,PatientID,MedicalStaffID) VALUES
(1,'2021-06-04 10:34:09','2021-06-12 14:34:09','A',500,2,1),
(2,'2021-06-09 10:34:09','2021-06-13 14:34:09','B',1000,3,2),
(3,'2021-06-12 10:34:09','2021-06-14 14:34:09','C',1500,1,3)
SET IDENTITY_INSERT Appointments OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T22:20:22.4623364+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

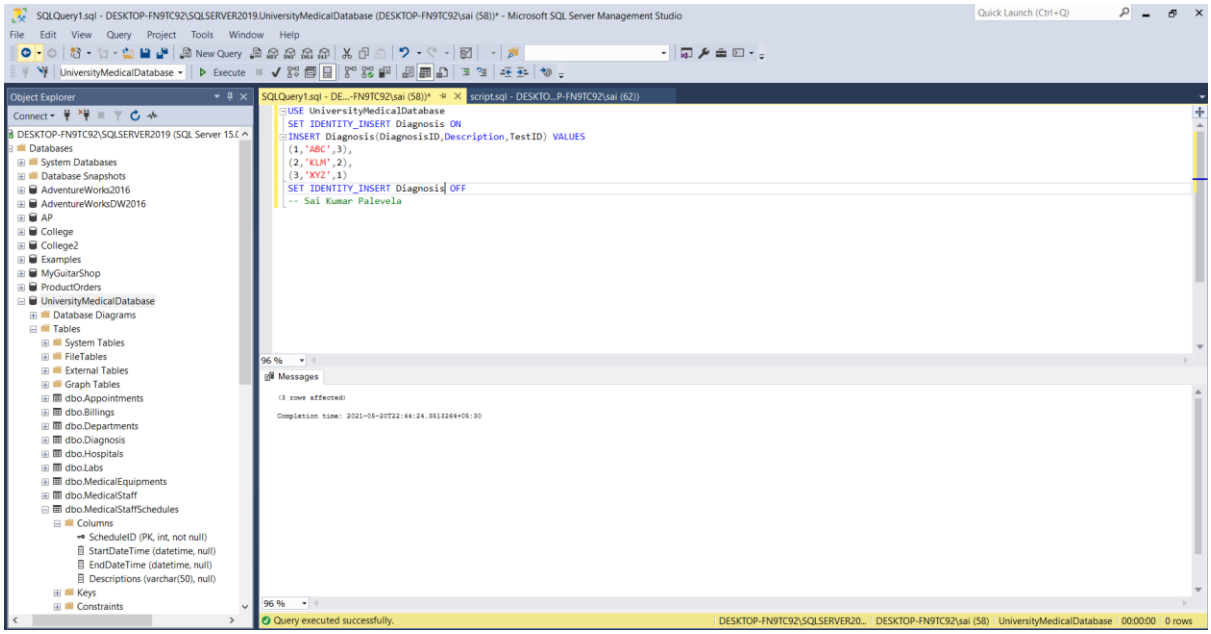

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT RoomSchedules ON
INSERT RoomSchedules(RoomScheduleID,StartDateTime,EndDateTime,PatientID,RoomID) VALUES
(1,'2021-06-04 09:34:09','2021-06-12 14:34:09',1,3),
(2,'2021-06-09 10:34:09','2021-06-13 16:34:09',2,1),
(3,'2021-06-12 09:34:09','2021-06-14 15:34:09',3,2)
SET IDENTITY_INSERT RoomSchedules OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T22:24:01.9035705+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

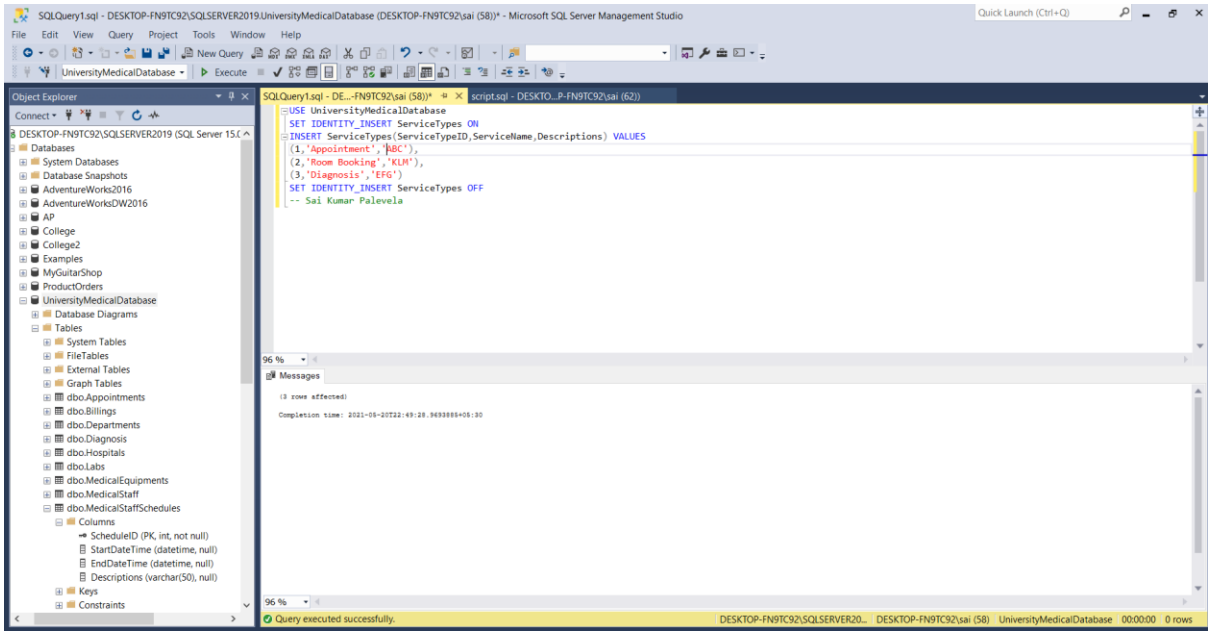

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Labs ON
INSERT Labs(LabID,LabName,LocationDetails) VALUES
(1,'Blood Collection','First Floor'),
(2,'Testing','Ground Floor'),
(3,'Diagnostic','First Floor')
SET IDENTITY_INSERT Labs OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T22:27:18.7065371+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows

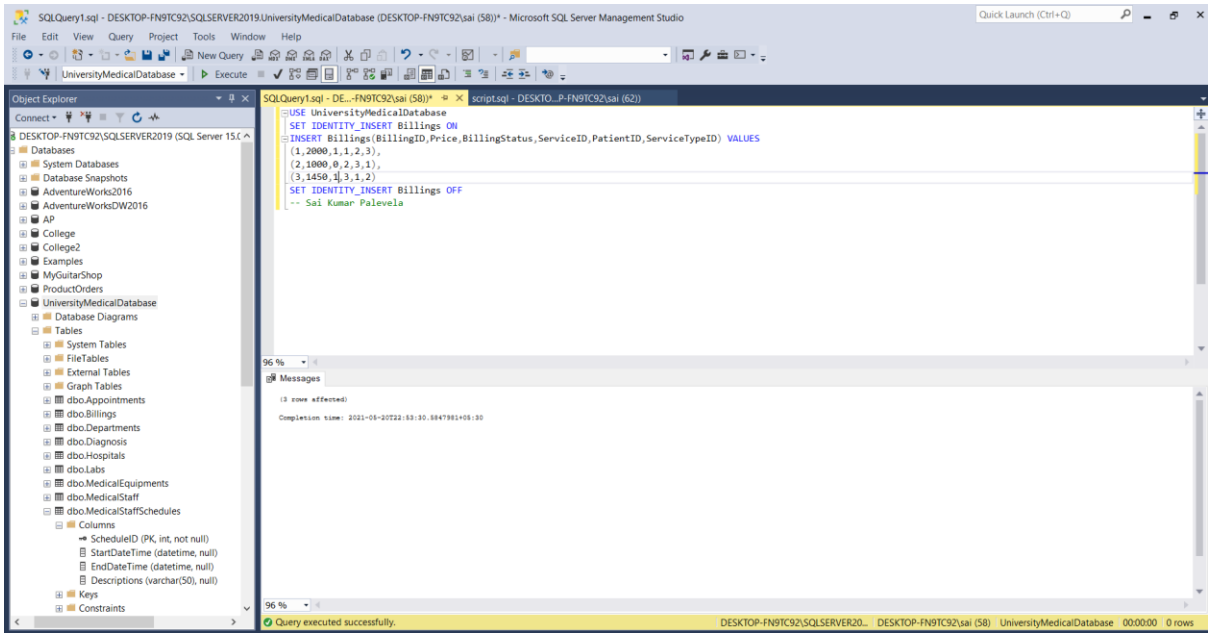

SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Tests ON
INSERT Tests(TestID,TestName,Price) VALUES
(1,'Sugar Test',50),
(2,'Blood',100),
(3,'CT Scan',150)
SET IDENTITY_INSERT Tests OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T22:30:06.7864183+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows


SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
UniversityMedicalDatabase
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT PatientTestResults ON
INSERT PatientTestResults(TestResultID,TestResults,PatientID,TestID,LabID) VALUES
(1,'XYZ',1,2,3),
(2,'KLM',2,3,1),
(3,'EFG',3,1,2)
SET IDENTITY_INSERT PatientTestResults OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-06-20T22:33:42.6859984+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows


SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
UniversityMedicalDatabase
Execute
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Visitors ON
INSERT Visitors(VisitorID,VisitorFirstName,VisitorLastName,City,State,PhoneNumber,PatientID) VALUES
(1,'Steve','Jobs','New York City','NY','(315) 447-5059',1),
(2,'Mark','Wilson','Albany','NY','(516) 992-4739',3),
(3,'Maria','Adams','San Francisco','CA','(916) 332-4532',2)
SET IDENTITY_INSERT Visitors OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T22:40:52.4352027+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows


SQLQuery1.sql - DESKTOP-FN9TC92\SQLSERVER2019.UniversityMedicalDatabase (DESKTOP-FN9TC92\sai (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
UniversityMedicalDatabase
Execute
Object Explorer
Connect
DESKTOP-FN9TC92\SQLSERVER2019 (SQL Server 15.0
Databases
System Databases
Database Snapshots
AdventureWorks2016
AdventureWorksDW2016
AP
College
College2
Examples
MyGuitarShop
ProductOrders
UniversityMedicalDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Appointments
dbo.Billings
dbo.Departments
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
SQLQuery1.sql - DE...-FN9TC92\sai (58))*
script.sql - DESKTO...P-FN9TC92\sai (62))
USE UniversityMedicalDatabase
SET IDENTITY_INSERT Diagnosis ON
INSERT Diagnosis(DiagnosisID,Description,TestID) VALUES
(1,'ABC',3),
(2,'KLM',2),
(3,'XYZ',1)
SET IDENTITY_INSERT Diagnosis OFF
-- Sai Kumar Palevela
96 %
Messages
(3 rows affected)
Completion time: 2021-05-20T22:44:24.3513264+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (58)
UniversityMedicalDatabase
00:00:00
0 rows





b) Views:

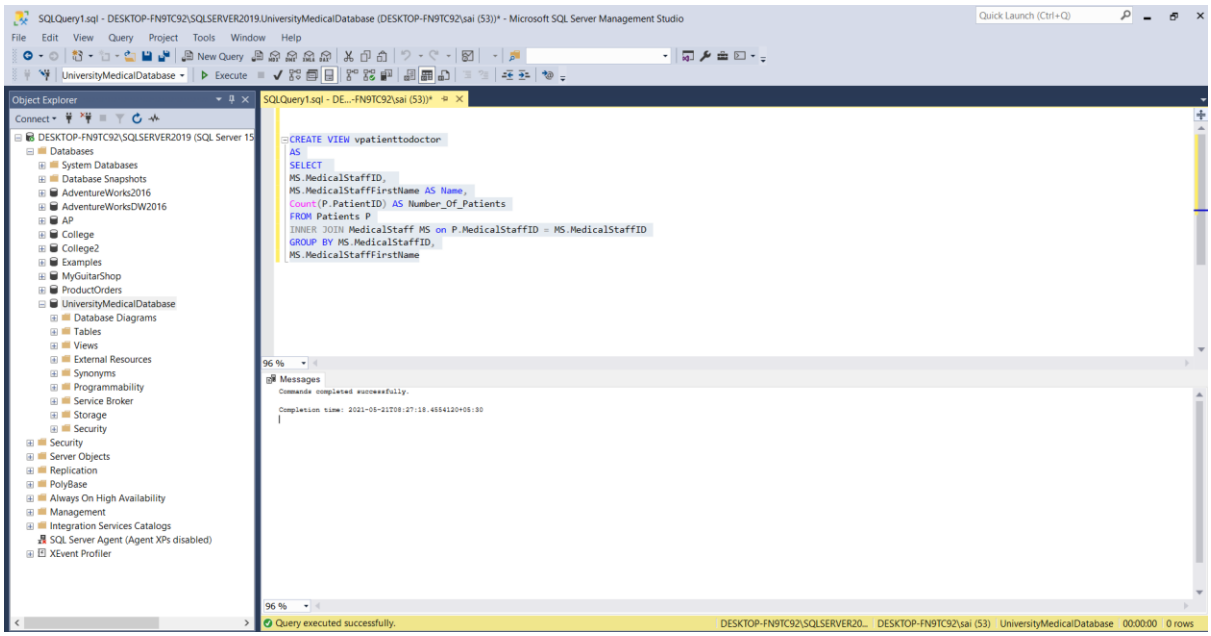
In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table, however the view itself does not store any data in it. The fields in a view are fields from one or more real tables in the database.
Here is the list of views that I have created and used in my project.

View 1:

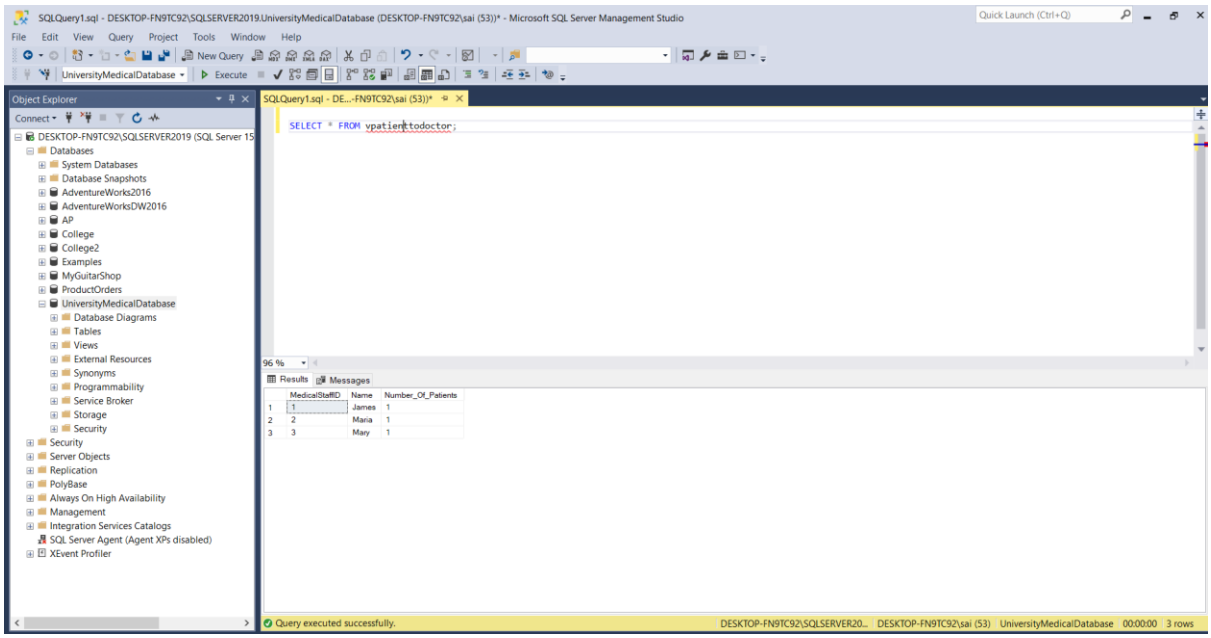
Number of patients for each doctor

```
CREATE VIEW vpatienttodoctor
AS
SELECT
```

```
MS.MedicalStaffID,
MS.MedicalStaffFirstName AS Name,
Count(P.PatientID) AS Number_Of_Patients
FROM Patients P
INNER JOIN MedicalStaff MS on P.MedicalStaffID = MS.MedicalStaffID
GROUP BY MS.MedicalStaffID,
MS.MedicalStaffFirstName
```



Below screenshot shows the result of the above created view

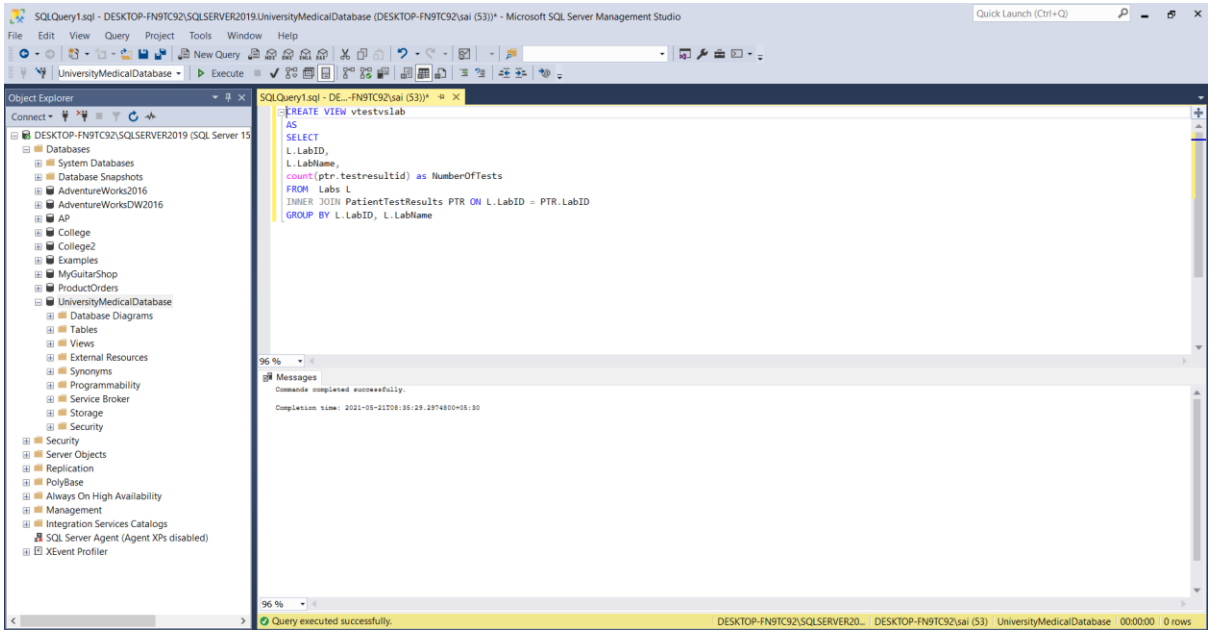


View 2:

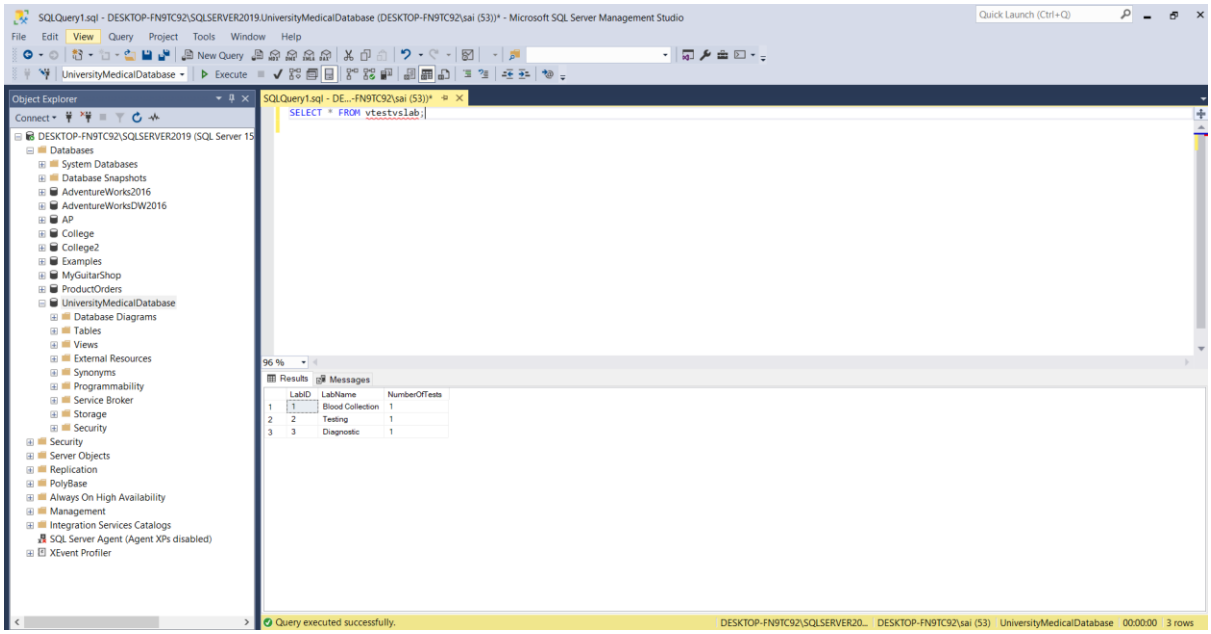
Total tests conducted in each lab.

```
CREATE VIEW vtestvslab
AS
```

```
SELECT
L.LabID,
L.LabName,
count(ptr.testresultid) as NumberOfTests
FROM  Labs L
INNER JOIN PatientTestResults PTR ON L.LabID = PTR.LabID
GROUP BY L.LabID, L.LabName
```



Below screenshot shows the result of the above created view

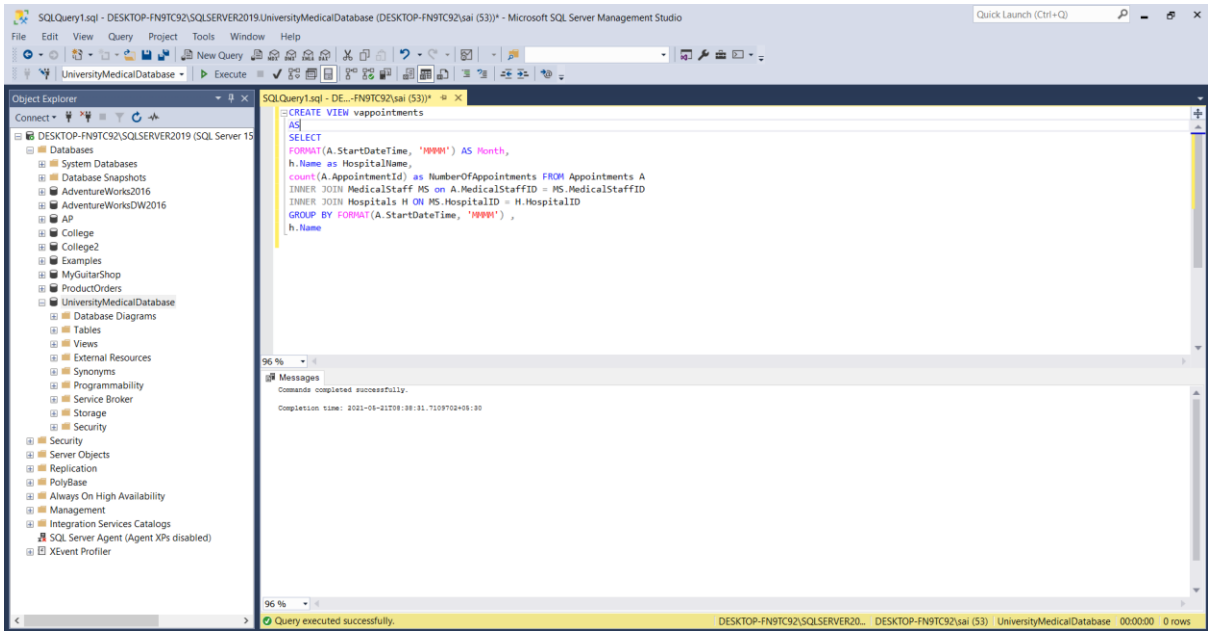


View 3:

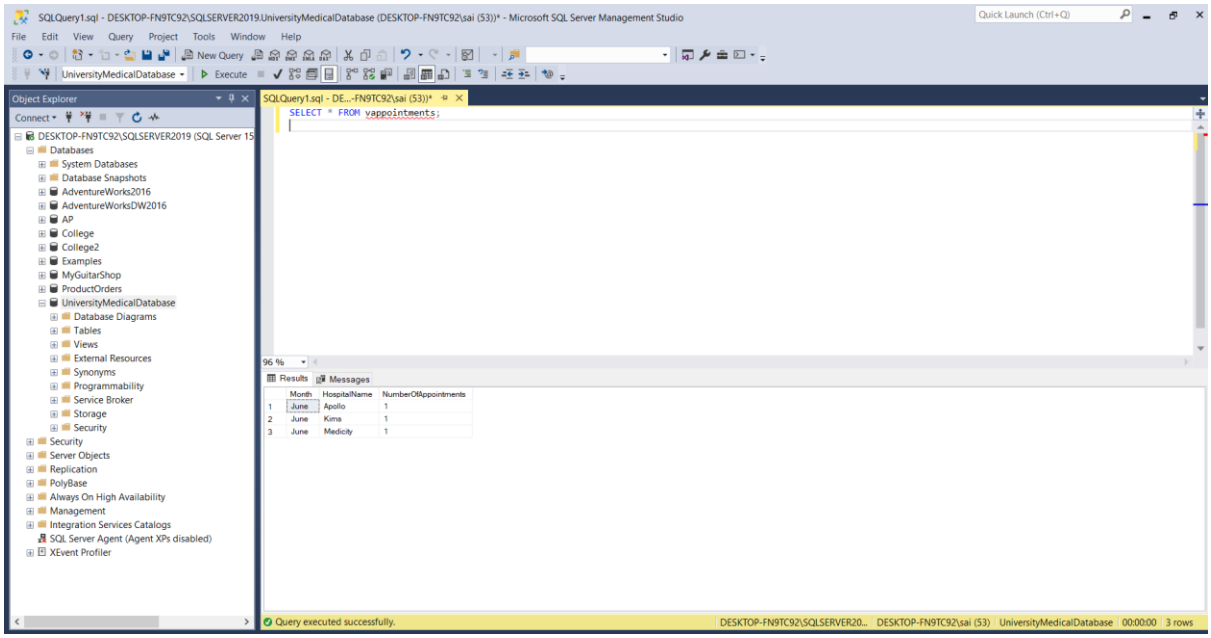
Number of appointments each month at each hospital

```
CREATE VIEW vappointments
AS
```

```
SELECT
FORMAT(A.StartDateTime, 'MMMM') AS Month,
h.Name as HospitalName,
count(A.AppointmentId) as NumberOfAppointments FROM Appointments A
INNER JOIN MedicalStaff MS on A.MedicalStaffID = MS.MedicalStaffID
INNER JOIN Hospitals H ON MS.HospitalID = H.HospitalID
GROUP BY FORMAT(A.StartDateTime, 'MMMM') ,
h.Name
```



Below screenshot shows the result of the above created view

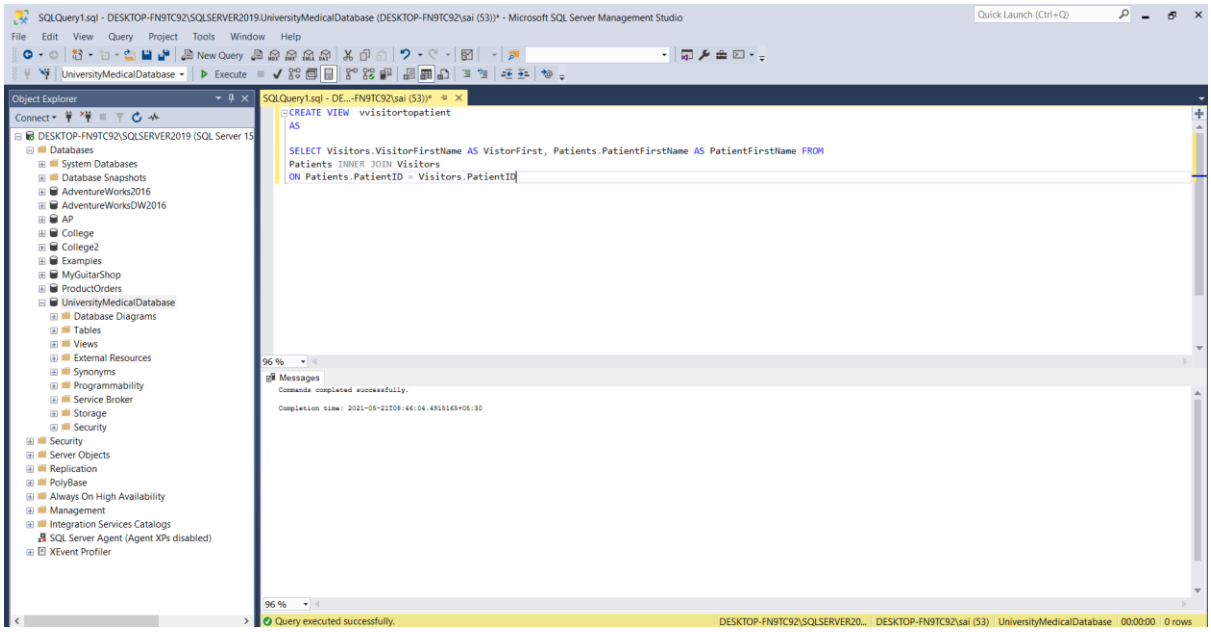


VIEW 4:

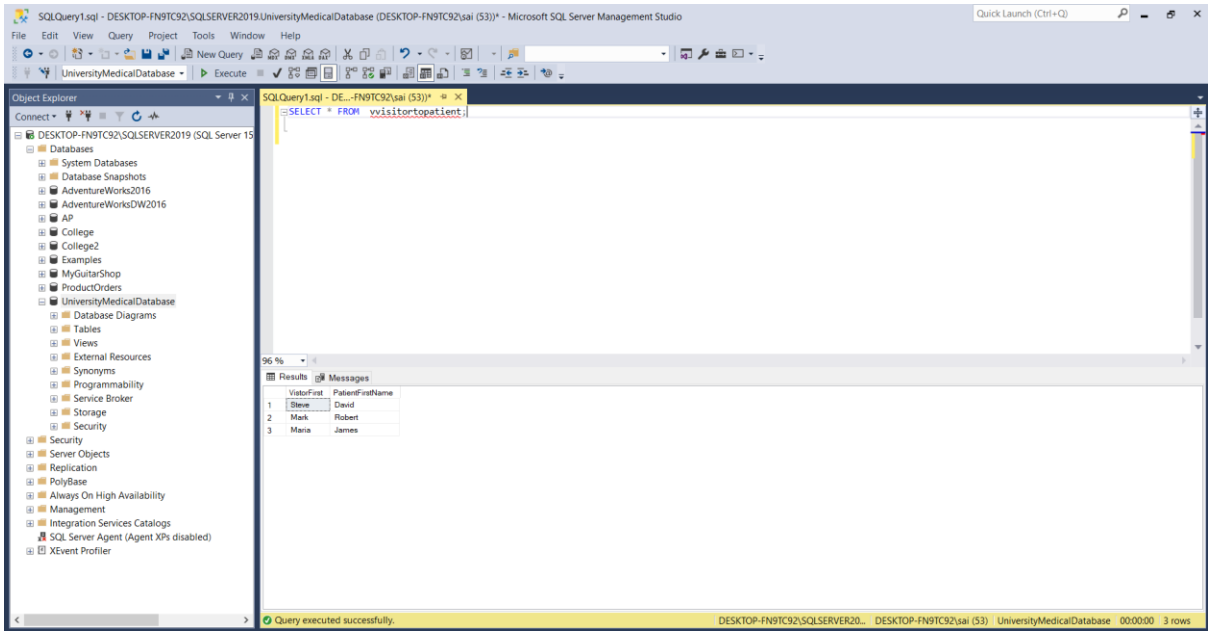
Visitor that has come to attend patient.

```
CREATE VIEW  vvisitortopatient
AS
```

```
SELECT Visitors.VisitorFirstName AS VistorFirst, Patients.PatientFirstName AS
PatientFirstName FROM
Patients INNER JOIN Visitors
ON Patients.PatientID = Visitors.PatientID
```



Below screenshot shows the result of the above created view

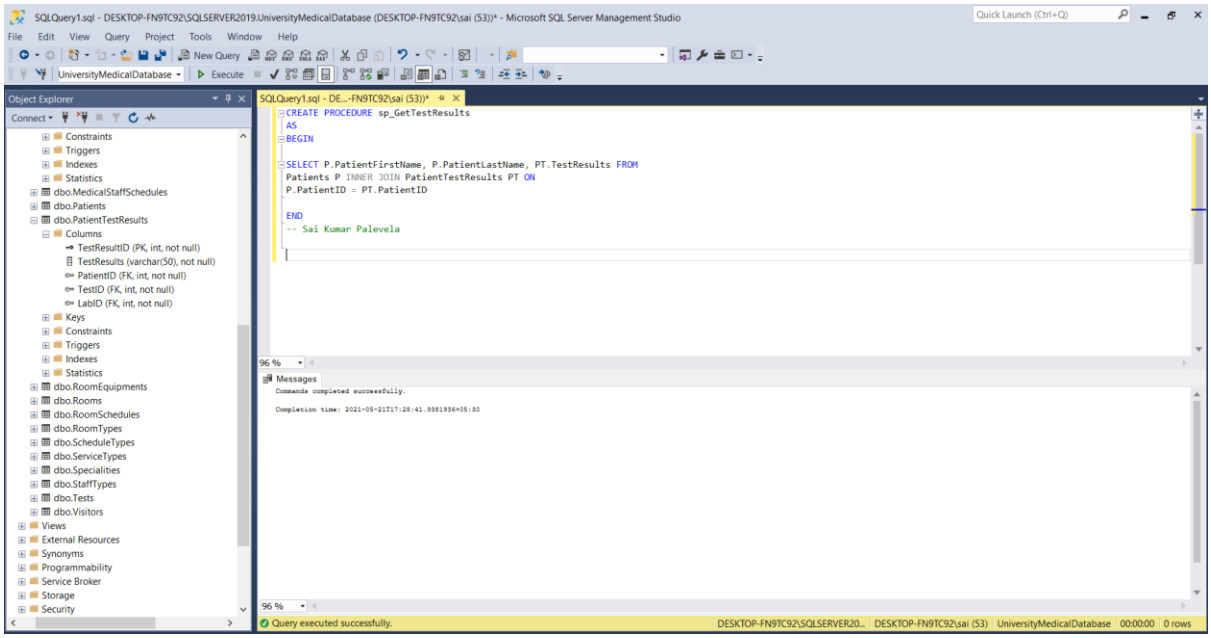


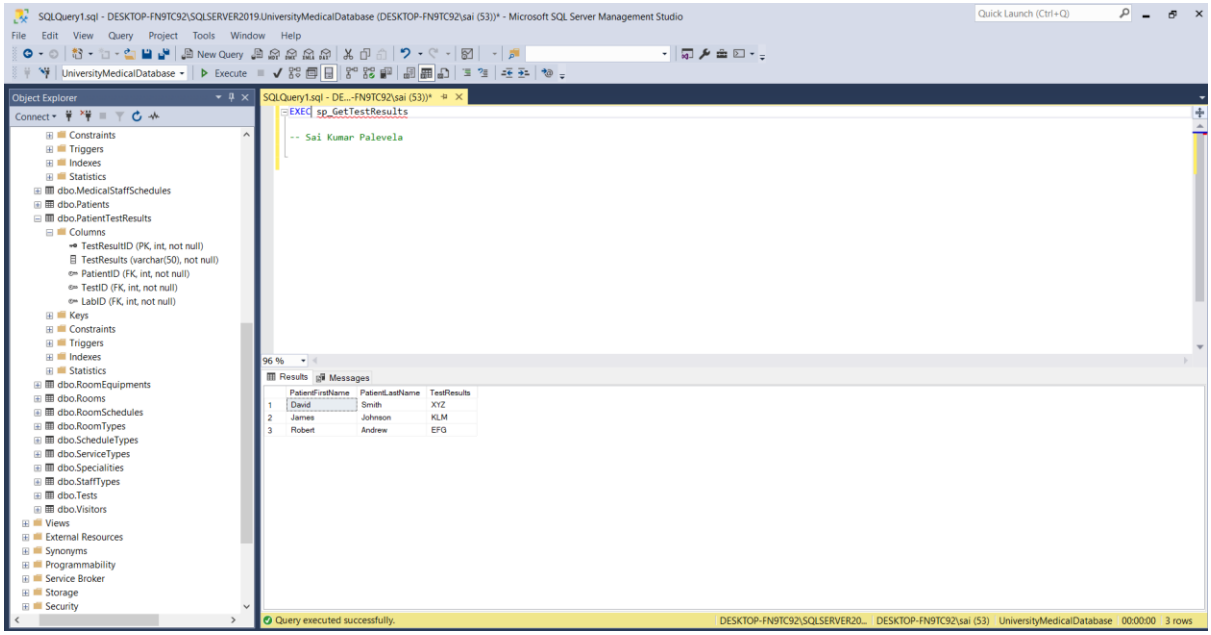
c) Stored Procedures

A stored procedure is a prepared SQL code that you can save, so the code can be reused repeatedly. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.
Here is the list of stored procedures used in the project.

Stored Procedure 1:

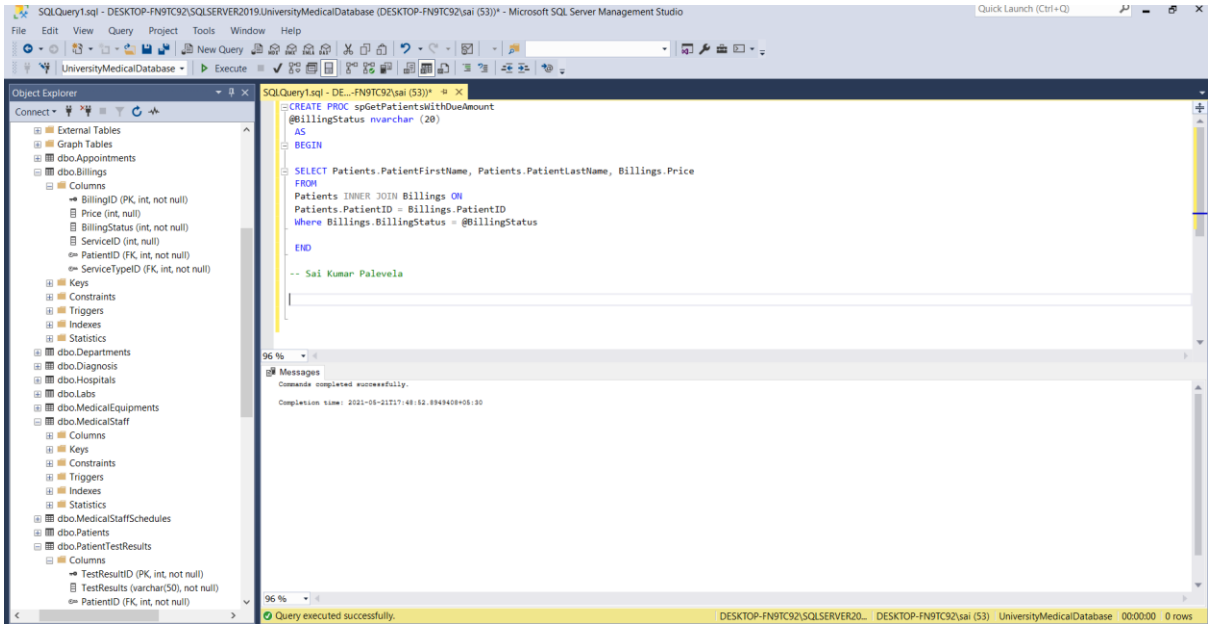
```
CREATE PROCEDURE sp_GetTestResults
AS
BEGIN

SELECT P.PatientFirstName, P.PatientLastName, PT.TestResults FROM
Patients P INNER JOIN PatientTestResults PT ON
P.PatientID = PT.PatientID

END
-- Sai Kumar Palevela
```



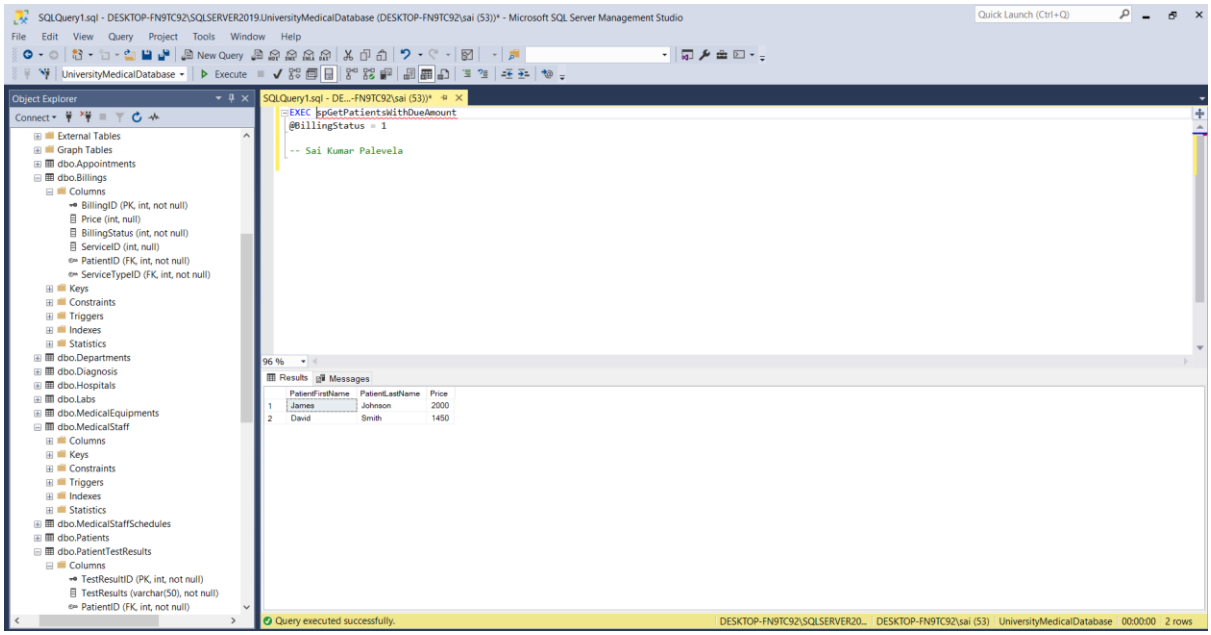


Stored Procedure 2:
Getting Patients who have not paid their fee.

```
CREATE PROC spGetPatientsWithDueAmount
@BillingStatus nvarchar (20)
 AS
 BEGIN

 SELECT Patients.PatientFirstName, Patients.PatientLastName, Billings.Price
 FROM
 Patients INNER JOIN Billings ON
 Patients.PatientID = Billings.PatientID
 Where Billings.BillingStatus = @BillingStatus

 END

-- Sai Kumar Palevela
```



Below is the Screenshot of Execution of above Stored Procedure.



d) Functions:

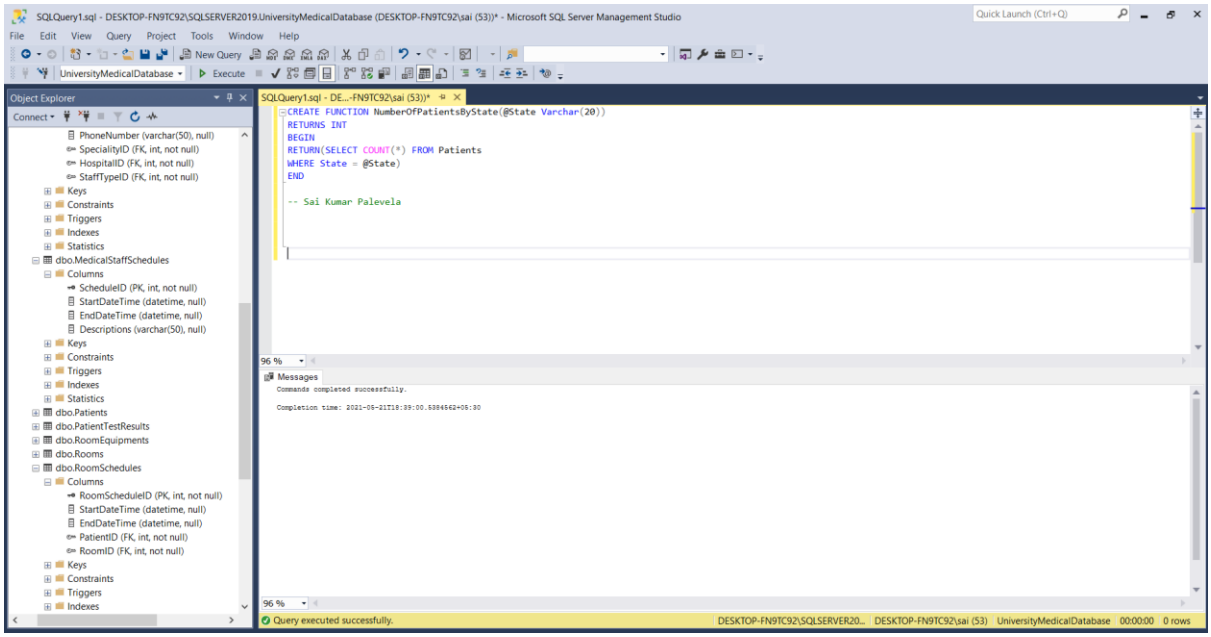
Functions: (Pre-defined and User Defined Functions)

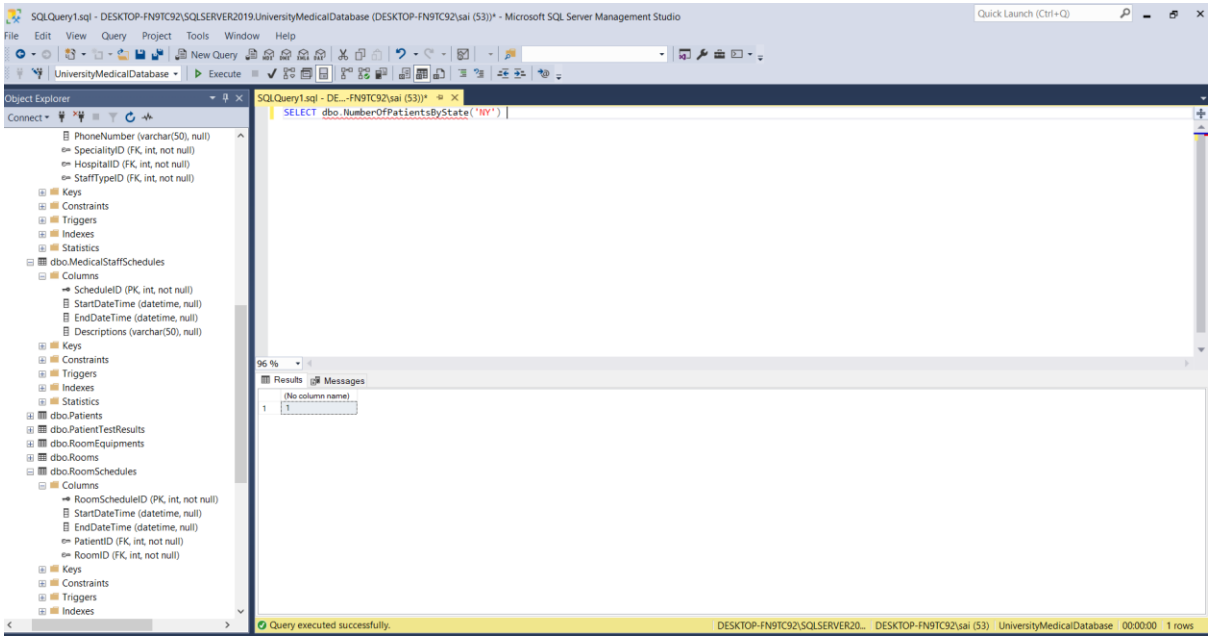
In SQL, there are many Pre-defined Aggregate function like COUNT, MIN, MAX, SUM, AVG etc. These Pre-defined functions are used in SQL SELECT expressions to calculate values and manipulate data. These functions can be used anywhere in the expressions.
Here are the List of Function used in the Project.

First Function:

```
CREATE FUNCTION NumberOfPatientsByState(@State Varchar(20))
RETURNS INT
BEGIN
RETURN(SELECT COUNT(*) FROM Patients
WHERE State = @State)
END
```

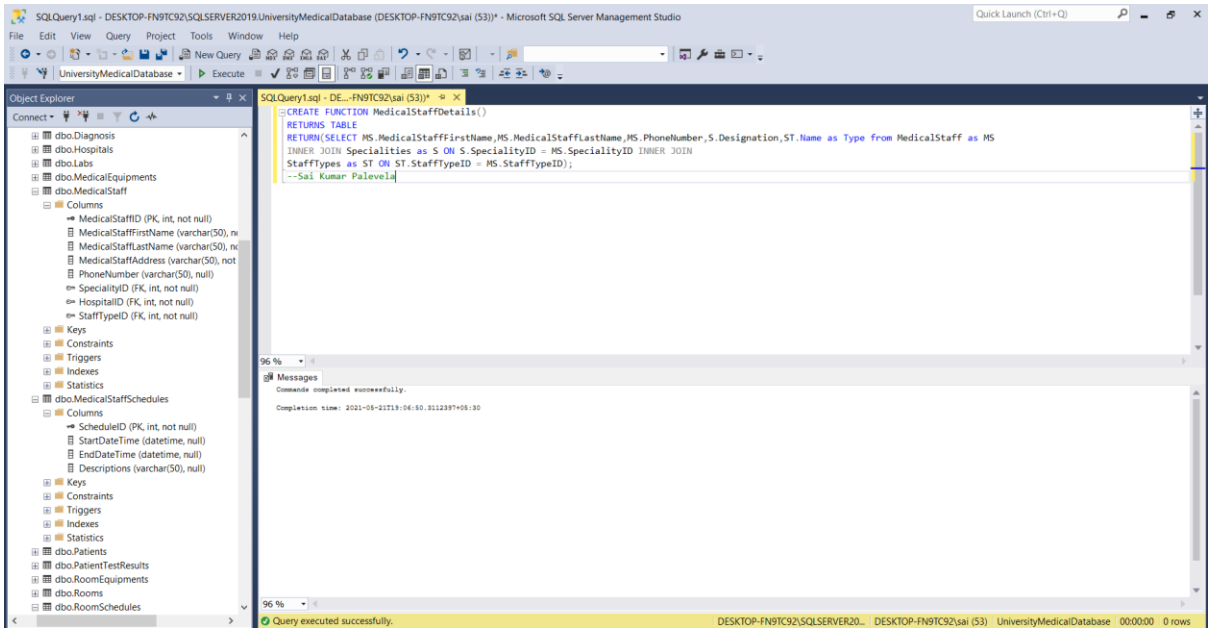
-- Sai Kumar Palevela



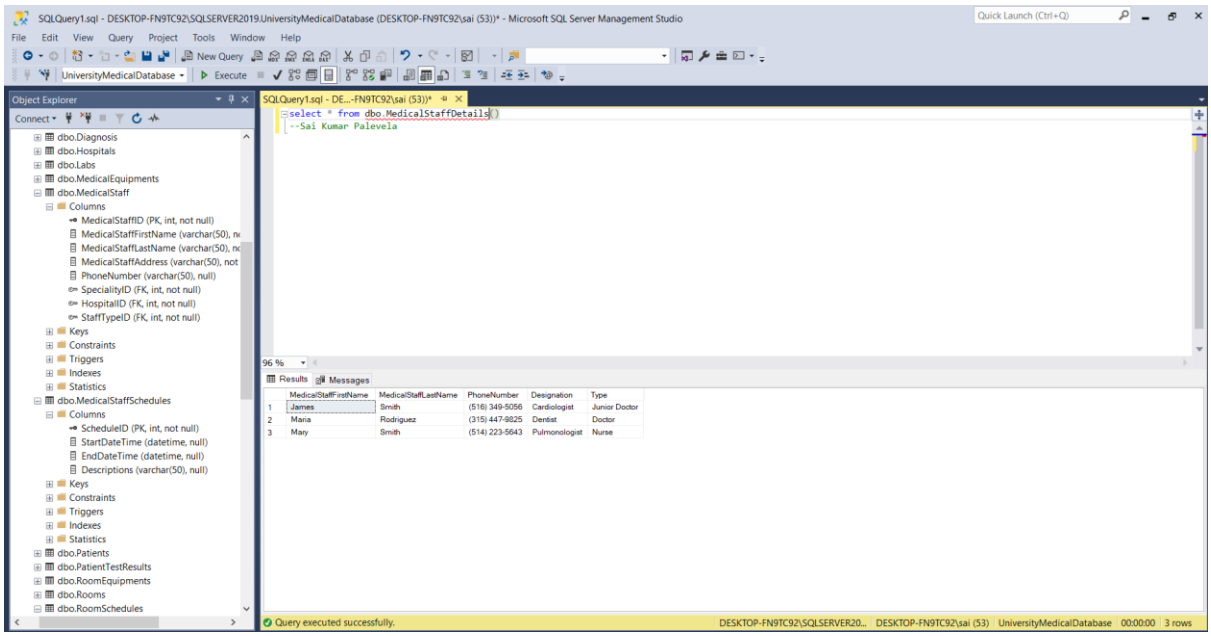


Second Function:

```
CREATE FUNCTION MedicalStaffDetails()
RETURNS TABLE
RETURN(SELECT
MS.MedicalStaffFirstName,MS.MedicalStaffLastName,MS.PhoneNumber,S.Designation,ST.Name as Type from MedicalStaff as MS
INNER JOIN Specialities as S ON S.SpecialityID = MS.SpecialityID INNER JOIN
StaffTypes as ST ON ST.StaffTypeID = MS.StaffTypeID);
```




```
select * from dbo.MedicalStaffDetails()
--Sai Kumar Palevela
```

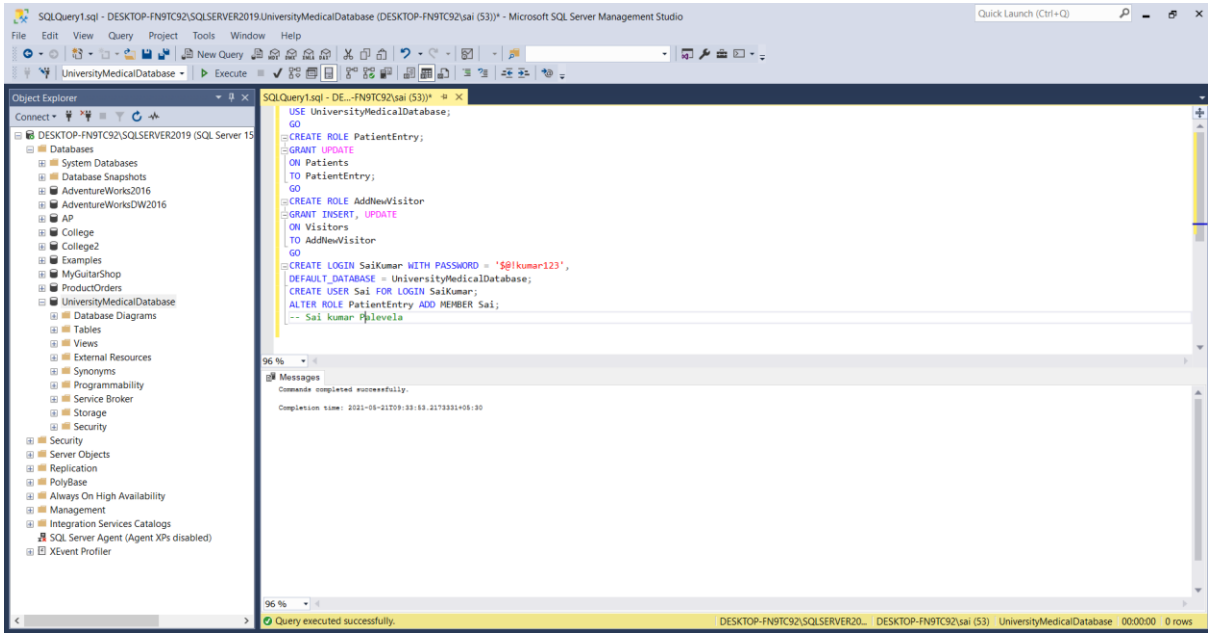


e) Scripts

Script : create users with various security levels, password assignments, roles, encryptions

```
USE UniversityMedicalDatabase;
GO
CREATE ROLE PatientEntry;
GRANT UPDATE
ON Patients
TO PatientEntry;
GO
CREATE ROLE AddNewVisitor
```

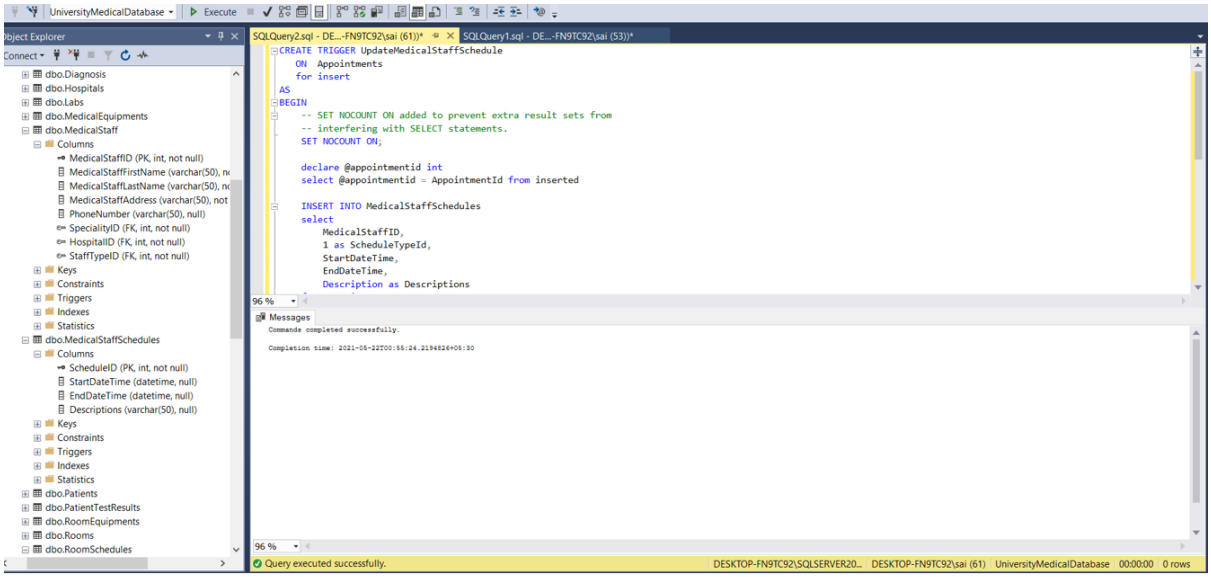
```
GRANT INSERT, UPDATE
ON Visitors
TO AddNewVisitor
GO
CREATE LOGIN SaiKumar WITH PASSWORD = '$@!kumar123',
DEFAULT_DATABASE = UniversityMedicalDatabase;
CREATE USER Sai FOR LOGIN SaiKumar;
ALTER ROLE PatientEntry ADD MEMBER Sai;
```

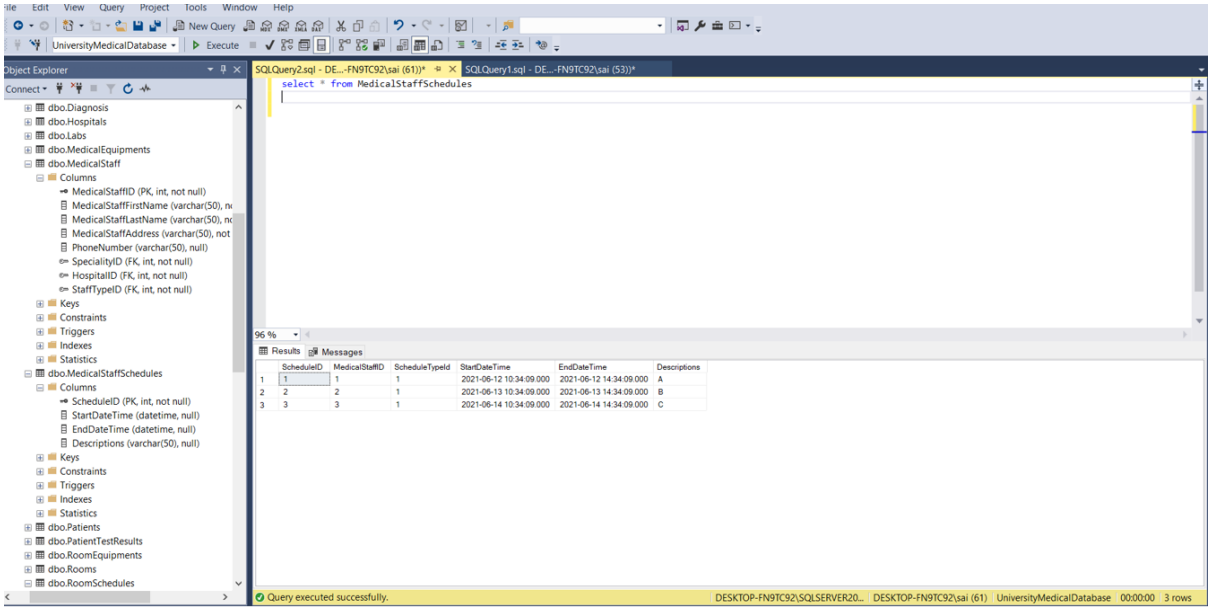


f) Triggers

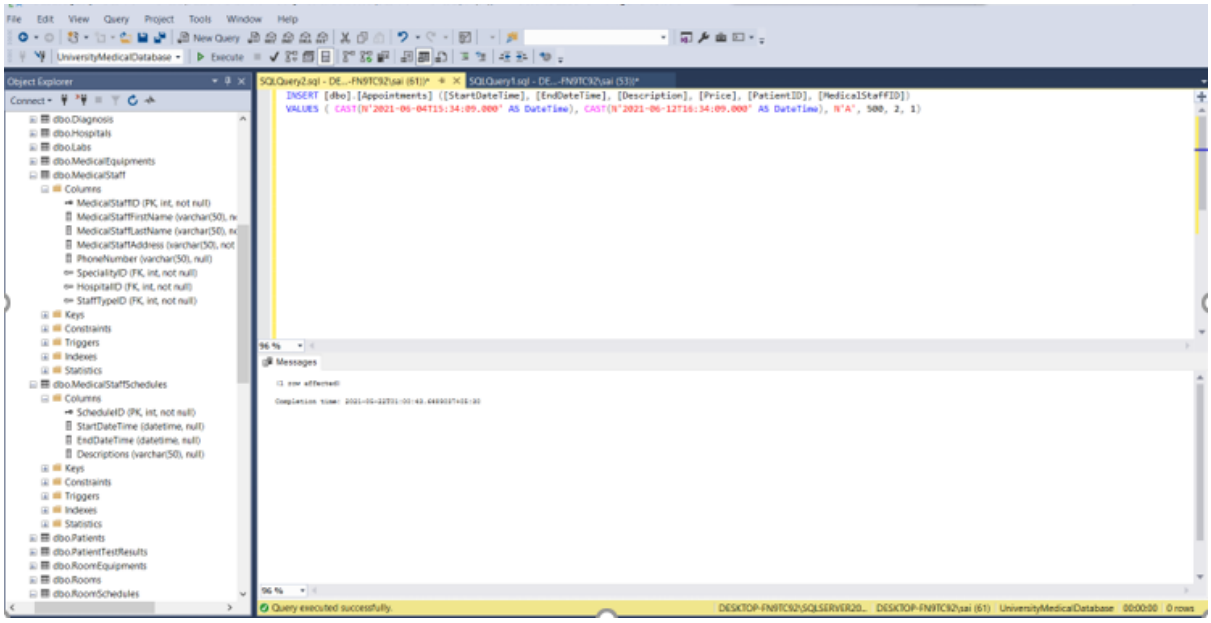
Trigger1:

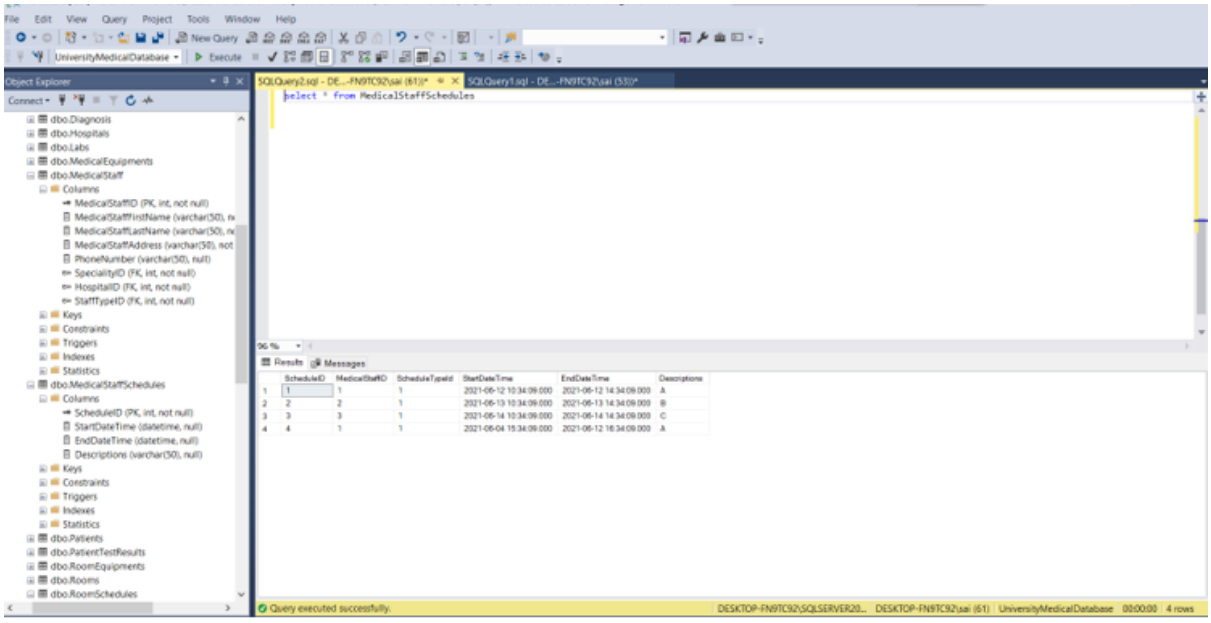
```
CREATE TRIGGER UpdateMedicalStaffSchedule
   ON  Appointments
   for insert
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;

	declare @appointmentid int
	select @appointmentid = AppointmentId from inserted

	INSERT INTO MedicalStaffSchedules
	select
		MedicalStaffID,
		1 as ScheduleTypeId,
		StartDateTime,
		EndDateTime,
		Description as Descriptions
	from Appointments
		WHERE AppointmentID = @appointmentid
--Sai Kumar Palevela
```


UniversityMedicalDatabase
Execute
Object Explorer
Connect
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
Columns
MedicalStaffID (PK, int, not null)
MedicalStaffFirstName (varchar(50), n
MedicalStaffLastName (varchar(50), n
MedicalStaffAddress (varchar(50), not
PhoneNumber (varchar(50), null)
SpecialityID (FK, int, not null)
HospitalID (FK, int, not null)
StaffTypeID (FK, int, not null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
SQLQuery2.sql - DE...-FN9TC92\sai (61))*
SQLQuery1.sql - DE...-FN9TC92\sai (53))*
CREATE TRIGGER UpdateMedicalStaffSchedule
ON Appointments
for insert
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;
declare @appointmentid int
select @appointmentid = AppointmentId from inserted
INSERT INTO MedicalStaffSchedules
select
MedicalStaffID,
1 as ScheduleTypeId,
StartDateTime,
EndDateTime,
Description as Descriptions
96 %
Messages
Commands completed successfully.
Completion time: 2021-06-22T00:55:24.2194826+05:30
96 %
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (61)
UniversityMedicalDatabase
00:00:00
0 rows


File Edit View Query Project Tools Window Help
New Query
UniversityMedicalDatabase
Execute
Object Explorer
Connect
dbo.Diagnosis
dbo.Hospitals
dbo.Labs
dbo.MedicalEquipments
dbo.MedicalStaff
Columns
MedicalStaffID (PK, int, not null)
MedicalStaffFirstName (varchar(50), n
MedicalStaffLastName (varchar(50), n
MedicalStaffAddress (varchar(50), not
PhoneNumber (varchar(50), null)
SpecialityID (FK, int, not null)
HospitalID (FK, int, not null)
StaffTypeID (FK, int, not null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.MedicalStaffSchedules
Columns
ScheduleID (PK, int, not null)
StartDateTime (datetime, null)
EndDateTime (datetime, null)
Descriptions (varchar(50), null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Patients
dbo.PatientTestResults
dbo.RoomEquipments
dbo.Rooms
dbo.RoomSchedules
SQLQuery2.sql - DE...-FN9TC92\sai (61))*
SQLQuery1.sql - DE...-FN9TC92\sai (53))*
select * from MedicalStaffSchedules
96 %
Results
Messages
ScheduleID MedicalStaffID ScheduleTypeId StartDateTime EndDateTime Descriptions
1 1 1 1 2021-06-12 10:34:09.000 2021-06-12 14:34:09.000 A
2 2 2 1 2021-06-13 10:34:09.000 2021-06-13 14:34:09.000 B
3 3 3 1 2021-06-14 10:34:09.000 2021-06-14 14:34:09.000 C
Query executed successfully.
DESKTOP-FN9TC92\SQLSERVER20...
DESKTOP-FN9TC92\sai (61)
UniversityMedicalDatabase
00:00:00
3 rows

Testing of Trigger:

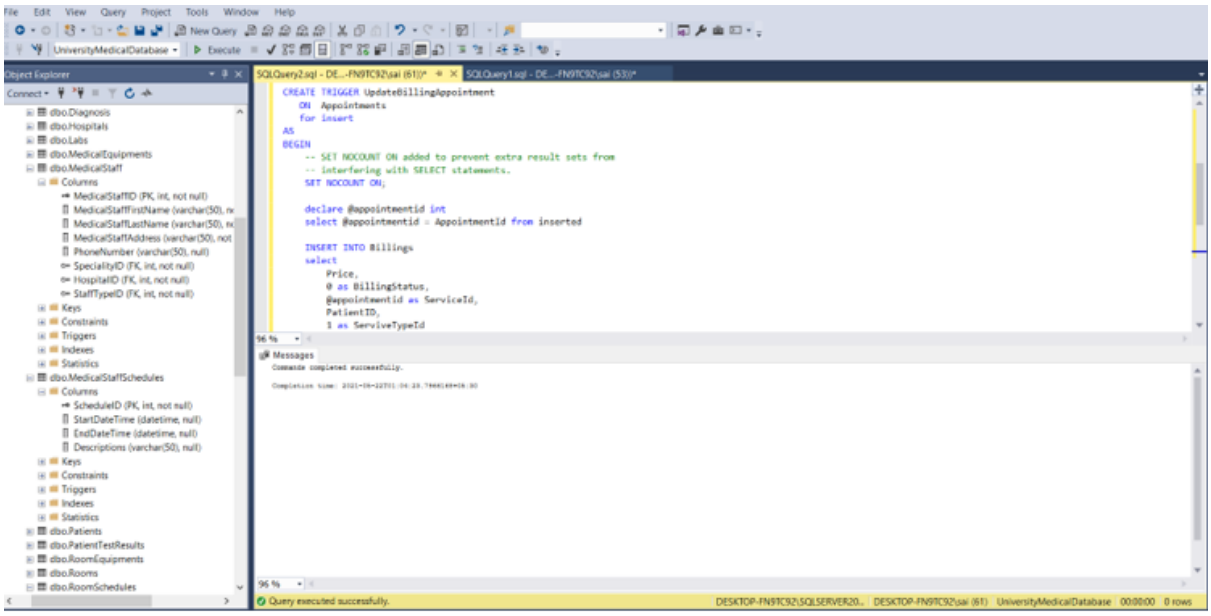


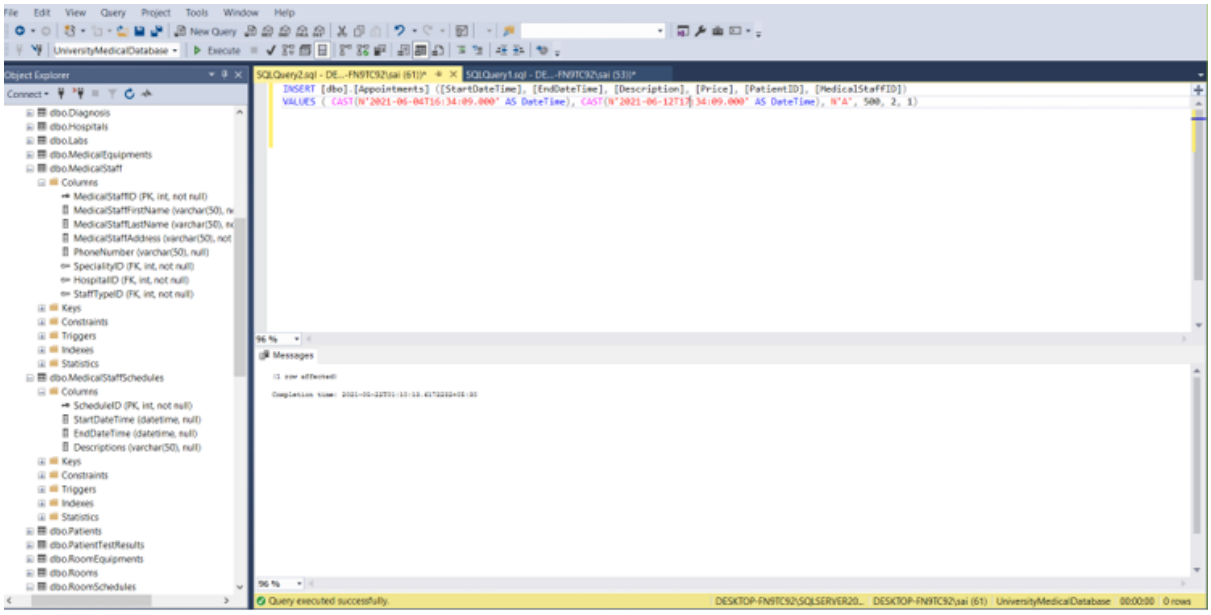


Trigger 2:

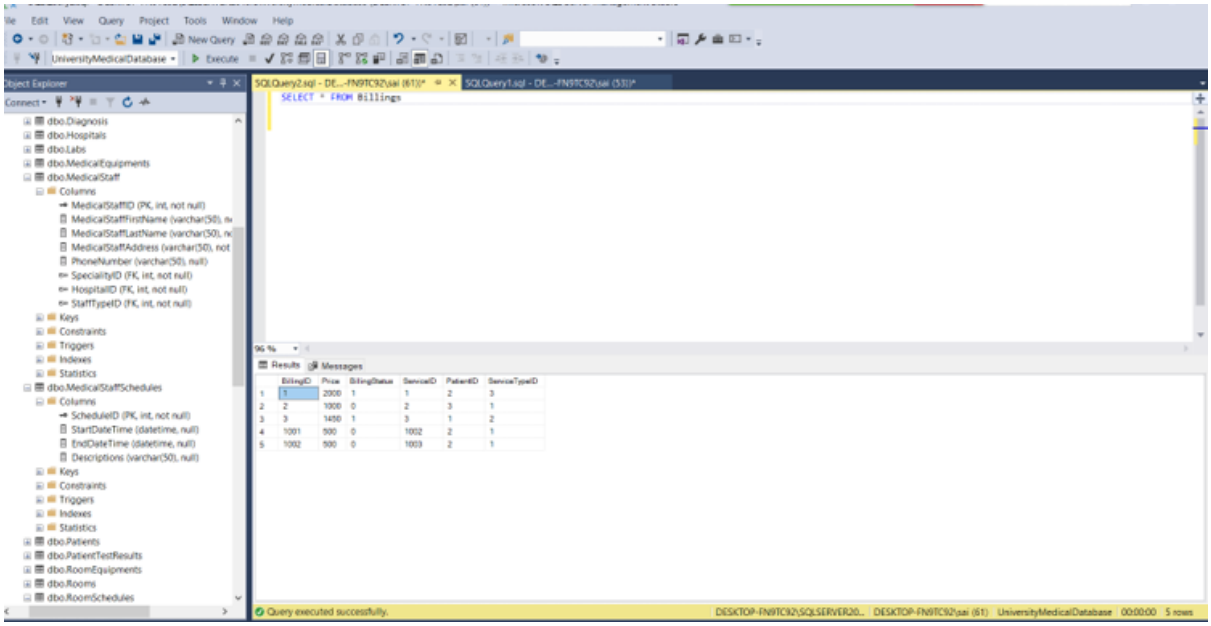
```
CREATE TRIGGER UpdateBillingAppointment
   ON  Appointments
   for insert
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;
```

```
        declare @appointmentid int
        select @appointmentid = AppointmentId from inserted

        INSERT INTO Billings
        select
                Price,
                0 as BillingStatus,
                @appointmentid as ServiceId,
                PatientID,
                1 as ServiveTypeId
        from Appointments
                WHERE AppointmentID = @appointmentid

END
```





Testing Trigger:



5) Conclusion:

The implementation and design of database to keep track of every Patient status throughout the Diagnosis process from the time s/he got admitted to hospital till s/he got discharged from hospital.
This database is created from the scratch and all the data that I have used in my project is considered from the real-life values.
In this project, all the basic database concepts are implemented i.e. Starting with thorough understanding of the user/customer requirements and identify the objects. Then creation of tables, determining its constraints, relationships between the tables used, Entity Relationship diagram, inserting data, usage of standard queries, Normalizing the database, creation of Views, usage of Predefined and User defined functions, Stored Procedures, Triggers and ending with transactions. Finally, ends with the test cases for all the business scenarios. By doing this project I have got great command over various skills on all the database concepts also got experience on working with the real-world project from scratch.