

CSE 581 - Intro to Database Management
System
Project 2 Report

Shivani Neharkar
SUID: 320953757

Contents

Abstract.....	2
Database Design.....	2
Database Implementation.....	3
Data Testing.....	45
Conclusion.....	71

Abstract:

The Center For Medical(COM) database is created, put into use, and tested as part of this project. In this project, a hospital database's patient, employee, and facility relationships are shown.

Data Design, Data Implementation, Database Testing, and Conclusion are the four sections of this project. The E/R diagram is included in the Data Design portion. Table construction and data insertion are included in the data implementation part. A number of views, stored procedures, user created functions, triggers, transactions, and scripts are demonstrated in the Testing section to validate the database design. The end section often includes project analysis and remarks.

I. Designing Databases

1. Overview: The University Medical Center (CMU) is a hospital system database that primarily includes patient, employee (physician, nurse, and other staff), and facility data. Every table is created with daily life in mind. This database uses a total of 21 tables to describe the hospital system.
2. Basic table descriptions include those for patients, visitors, employees, facilities, labs, pharmacies, and imaging. The information in these tables is fundamental. We must set some keys in order to provide the relationship to the tables. I discover that the majority of relationships are one to many (constrained) in nature. Additionally, it was primarily used in one-to-one relationships to explore information that might not fit in a single table. I add one linking table for the many-to-many relationship.

Table —Primary Key--- Foreign Key

PatientInfo--- PatientID

CoverageOfInsurance---InsuranceID -----PatientID

Costs---- CostsID----- PatientID

Billing----- BillingID -----PatientID

VisitingInfo----- VisitingInfoID ----PatientID, VisitorsID

Visitors----- VisitorsID

PatientRecords---- PatRecID----- PatientID

Appointments----- AppID -----PatientID, WorkersID

Workers-----WorkersID

Staff -----WorkersID -----WorkersID

ScheduleForNonStaff----- NonStaffID--- WorkersID, FacilityID

DoctorSchedules----- DoctorSchelD----- WorkersID, ProvisionID

SchedulesForNurses---- ScheduleNurseID -----WorkersID, ProvisionID

SurgeryRoomSchedules---- SurRooSchelD -----WorkersID, ProvisionID

AdmitPatient -----AdmitPatientID -----WorkersID, ProvisionID

Provision-----ProvisionID

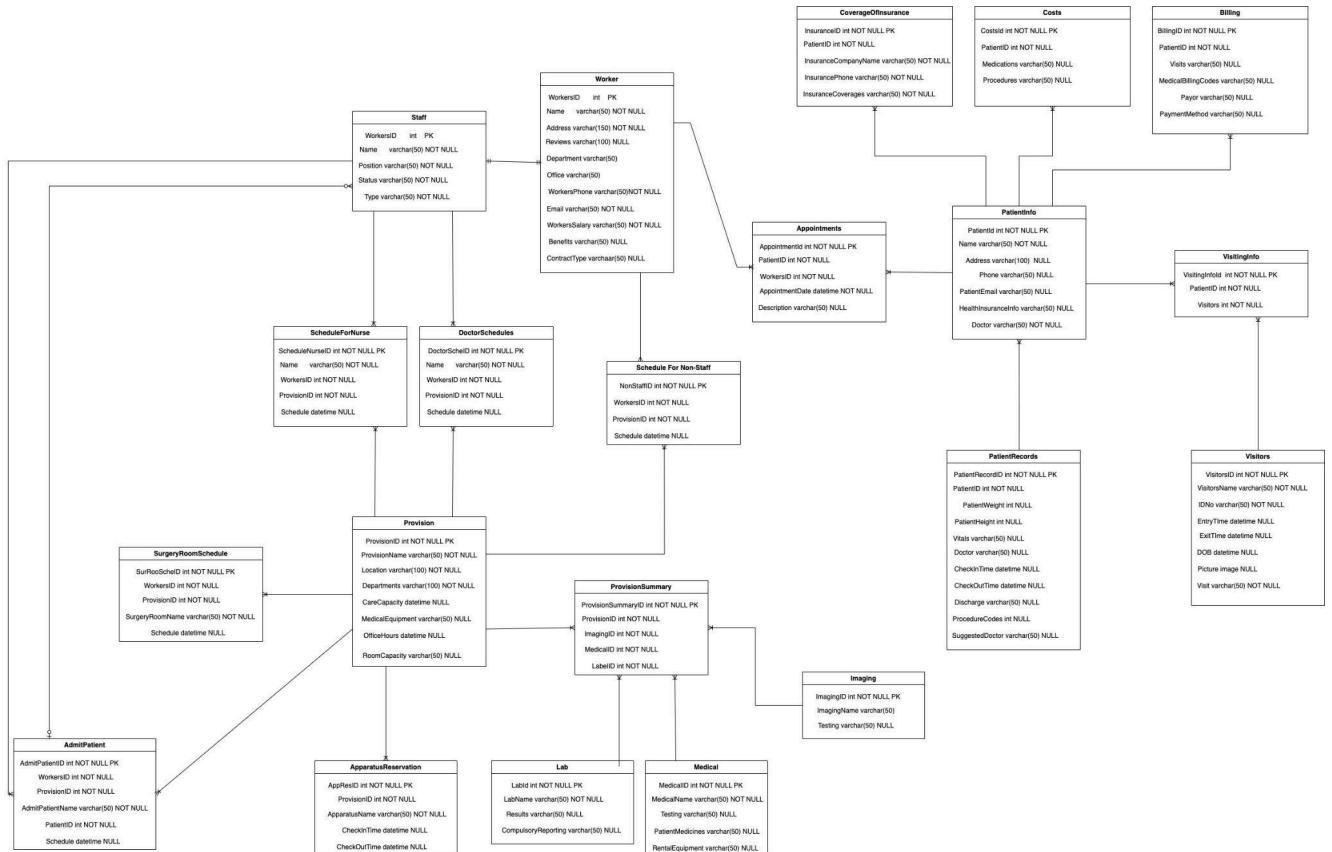
ApparatusReservations -----AppResID----- Provision

ProvisionSummary---ProvisionSummaryID----- ProvisionID

Lab----- LabID

Medical—MedicalID

Imaging----- ImagingID



ER diagram

II. Database Implementation

Workers:

```
CREATE TABLE Workers(
    WorkersID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Address varchar(100) NOT NULL,
    Reviews varchar(50) NULL,
    Department varchar(50) NULL,
    Office varchar(50) NULL,
    WorkersPhone varchar(50) NOT NULL,
    Email varchar(50) NOT NULL,
    WorkersSalary varchar(50) NOT NULL,
    Benefits varchar(50) NULL,
    ContractType varchar(50) NULL,
);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DOLLAR (SQL Server 16.0.1000.6 - Dollar\nehar)'. The 'CenterForMedical' database is selected. The 'Tables' node under 'CenterForMedical' is expanded, showing the 'Workers' table. The 'SQL' tab of the main window contains the following SQL code:

```

CREATE TABLE Workers (
    WorkersID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Address varchar(100) NOT NULL,
    Reviews varchar(50) NULL,
    Department varchar(50) NULL,
    Office varchar(50) NULL,
    WorkersPhone varchar(50) NULL,
    Email varchar(50) NOT NULL,
    WorkersSalary varchar(50) NOT NULL,
    Benefits varchar(50) NULL,
    ContractType varchar(50) NULL,
);

```

The status bar at the bottom indicates 'Query executed successfully.' and 'Completion time: 2023-04-30T15:53:04.4953947-04:00'.

ApparatusReservation:

```

CREATE TABLE ApparatusReservation (
    AppResID int NOT NULL PRIMARY KEY IDENTITY,
    ProvisionID int NOT NULL,
    ApparatusName varchar(50) NOT NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DOLLAR (SQL Server 16.0.1000.6 - Dollar\nehar)'. The 'CenterForMedical' database is selected. The 'Tables' node under 'CenterForMedical' is expanded, showing the 'ApparatusReservation' table. The 'SQL' tab of the main window contains the following SQL code:

```

CREATE TABLE ApparatusReservation (
    AppResID int NOT NULL PRIMARY KEY IDENTITY,
    ProvisionID int NOT NULL,
    ApparatusName varchar(50) NOT NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
);

```

The status bar at the bottom indicates 'Query executed successfully.' and 'Completion time: 2023-04-30T16:14:08.3587068-04:00'.

Provision:

```

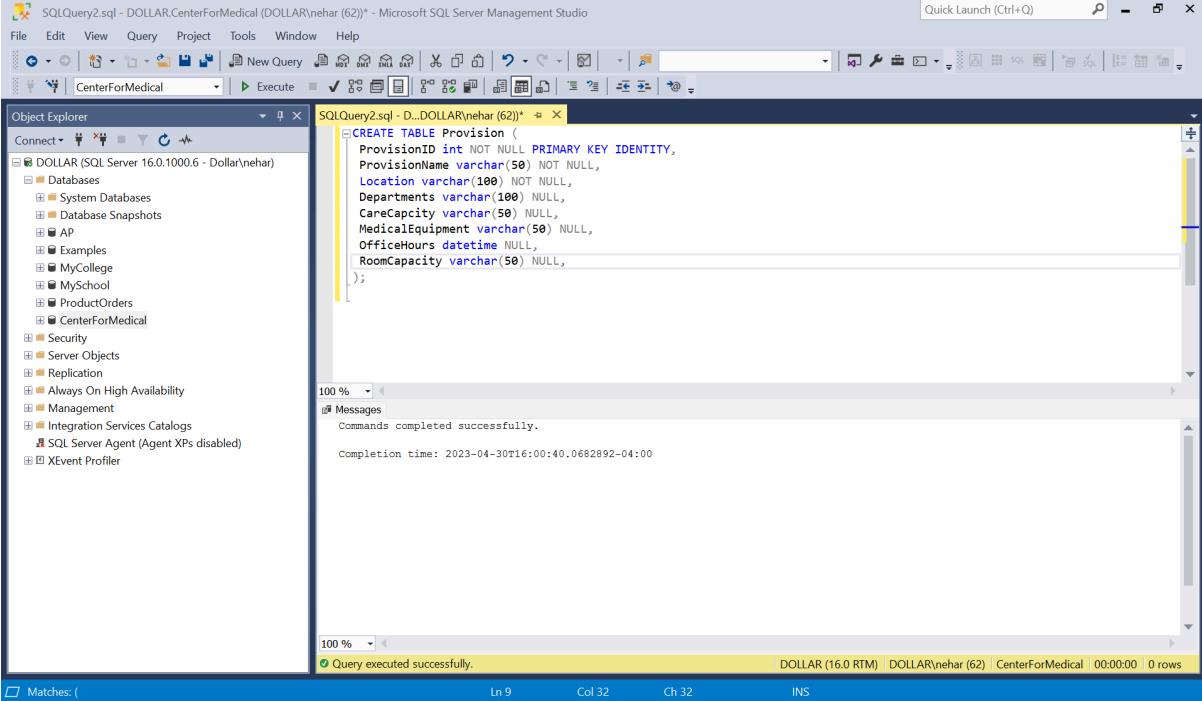
CREATE TABLE Provision(

```

```

ProvisionID int NOT NULL PRIMARY KEY IDENTITY,
ProvisionName varchar(50) NOT NULL,
Location varchar(100) NOT NULL,
Departments varchar(100) NULL,
CareCapacity varchar(50) NULL,
MedicalEquipment varchar(50) NULL,
OfficeHours datetime NULL,
RoomCapacity varchar(50) NULL,
);

```



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'CenterForMedical' selected. The right pane contains a query window titled 'SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62))'. The query is:

```

CREATE TABLE Provision (
    ProvisionID int NOT NULL PRIMARY KEY IDENTITY,
    ProvisionName varchar(50) NOT NULL,
    Location varchar(100) NOT NULL,
    Departments varchar(100) NULL,
    CareCapacity varchar(50) NULL,
    MedicalEquipment varchar(50) NULL,
    OfficeHours datetime NULL,
    RoomCapacity varchar(50) NULL,
);

```

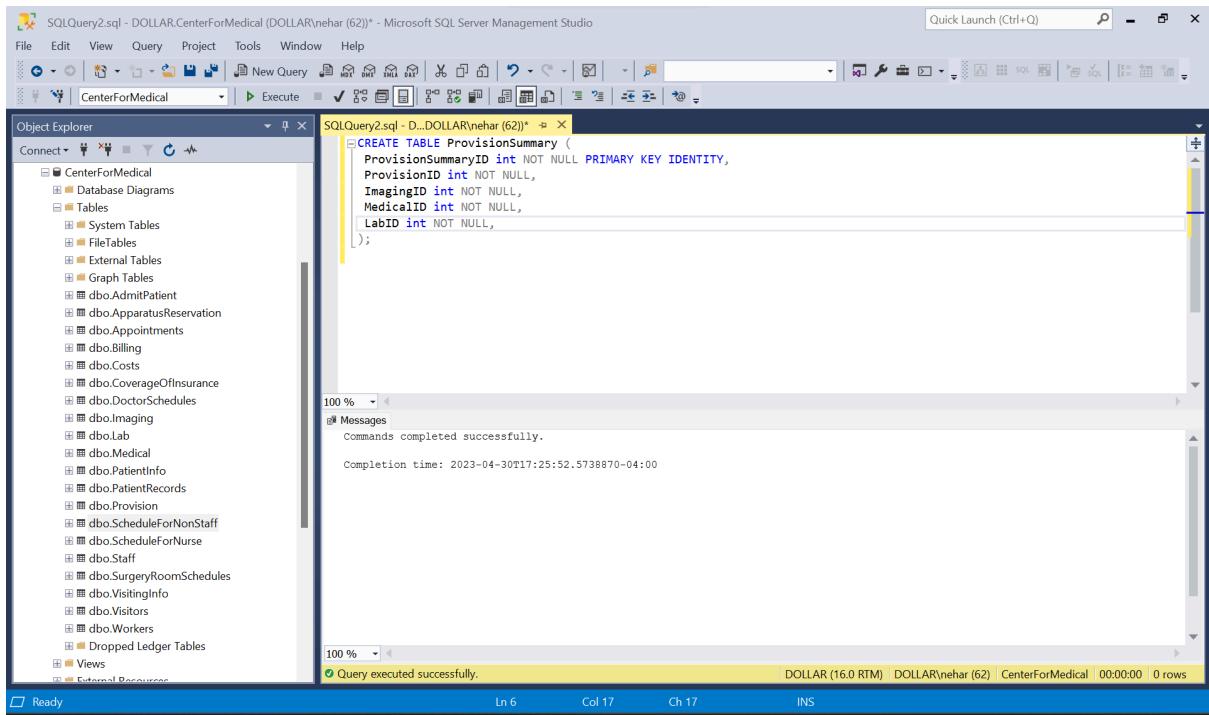
The status bar at the bottom indicates 'Query executed successfully.' and 'Completion time: 2023-04-30T16:00:40.0682892-04:00'.

ProvisionSummary:

```

CREATE TABLE ProvisionSummary (
    ProvisionSummaryID int NOT NULL PRIMARY KEY IDENTITY,
    ProvisionID int NOT NULL,
    ImagingID int NOT NULL,
    MedicalID int NOT NULL,
    LabID int NOT NULL,
);

```



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```
CREATE TABLE ProvisionSummary (
    ProvisionSummaryID int NOT NULL PRIMARY KEY IDENTITY,
    ProvisionID int NOT NULL,
    ImagingID int NOT NULL,
    MedicalID int NOT NULL,
    LabID int NOT NULL,
);
```

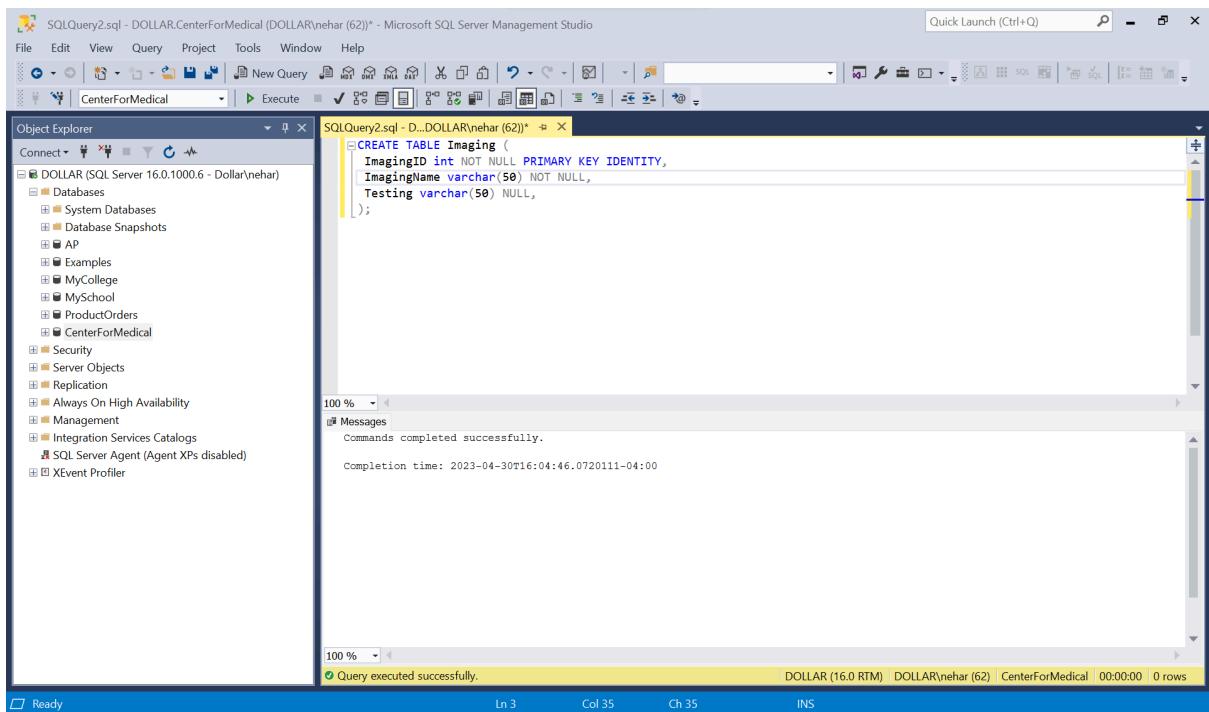
Messages
Commands completed successfully.
Completion time: 2023-04-30T17:25:52.5738870-04:00

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows

Imaging:

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



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

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

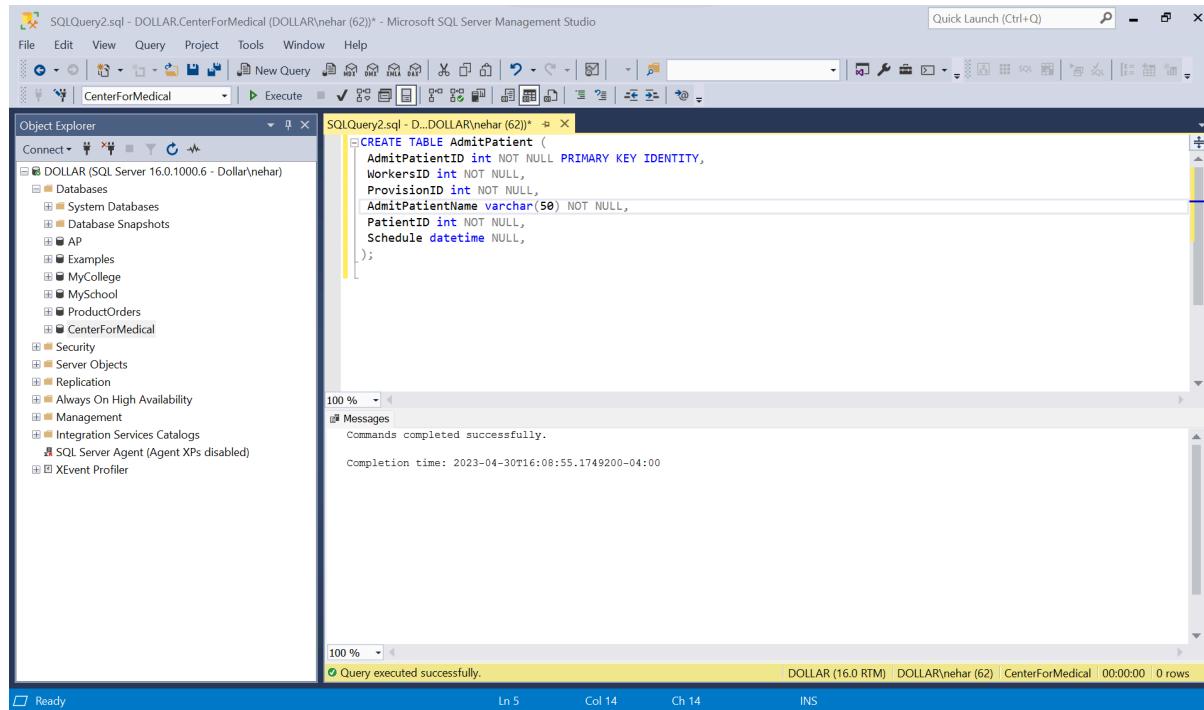
Messages
Commands completed successfully.
Completion time: 2023-04-30T16:04:46.0720111-04:00

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows

AdmitPatient:

```
CREATE TABLE AdmitPatient (
    AdmitPatientID int NOT NULL PRIMARY KEY IDENTITY,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    AdmitPatientName varchar(50) NOT NULL,
    PatientID int NOT NULL,
    Schedule datetime NULL,
);
```



Costs:

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

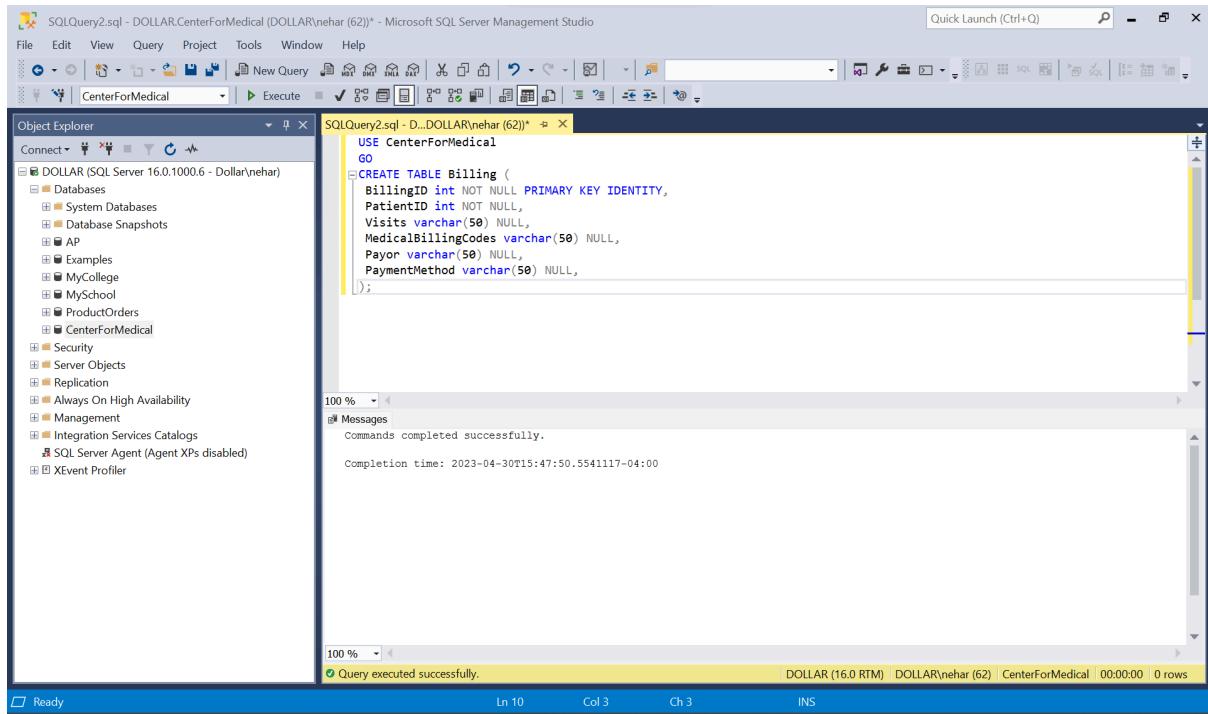
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists databases including 'DOLLAR' (selected), 'MyCollege', 'MySchool', 'ProductOrders', and 'CenterForMedical'. The 'CenterForMedical' database is expanded to show tables like 'Patients', 'Medications', and 'Procedures'. The central pane displays a query window titled 'SQLQuery2.sql - DOLLAR\nehar (62)*'. The query is:

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

The status bar at the bottom indicates 'Query executed successfully.' and 'Completion time: 2023-04-30T15:48:34.2609232-04:00'.

Billing:

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

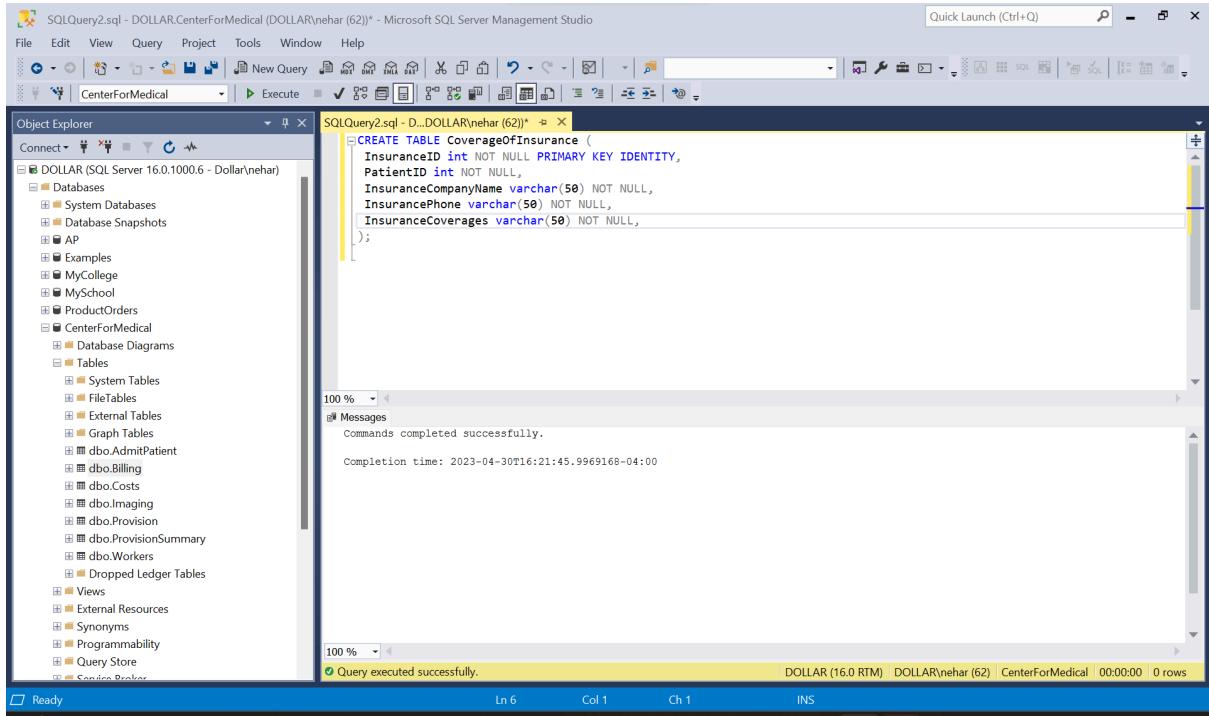


CoverageOfInsurance:

```

CREATE TABLE CoverageOfInsurance (
    InsuranceID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    InsuranceCompanyName varchar(50) NOT NULL,
    InsurancePhone varchar(50) NOT NULL,
    InsuranceCovgeries varchar(50) NOT NULL,
);

```



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```

CREATE TABLE CoverageOfInsurance (
    InsuranceID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    InsuranceCompanyName varchar(50) NOT NULL,
    InsurancePhone varchar(50) NOT NULL,
    InsuranceCovernages varchar(50) NOT NULL,
);

```

100 %

Messages

Commands completed successfully.

Completion time: 2023-04-30T16:21:45.9969168-04:00

100 %

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows

Lab:

CREATE TABLE Lab (

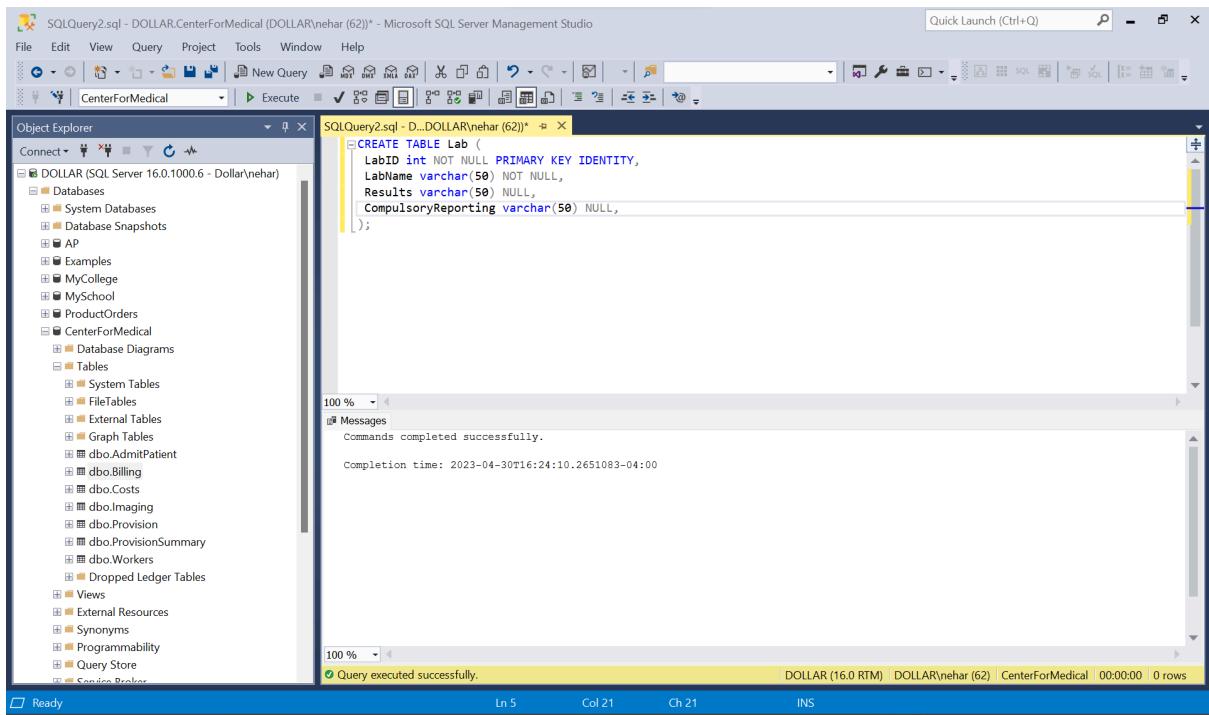
LabID int NOT NULL PRIMARY KEY IDENTITY,

LabName varchar(50) NOT NULL,

Results varchar(50) NULL,

CompulsoryReporting varchar(50) NULL,

);



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```

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

```

100 %

Messages

Commands completed successfully.

Completion time: 2023-04-30T16:24:10.2651083-04:00

100 %

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows

Staff:

```
CREATE TABLE Staff (
    WorkersID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Position varchar(50) NOT NULL,
    Status varchar(50) NOT NULL,
    Type varchar(50) NOT NULL,
);

);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including databases like DOLLAR, tables such as CenterForMedical, and various system objects. The central pane contains the T-SQL code for creating the Staff table. Below the code, the Messages pane shows the successful execution of the command with a completion time of 2023-04-30T16:26:05.9042819-04:00. The status bar at the bottom indicates the query was executed successfully.

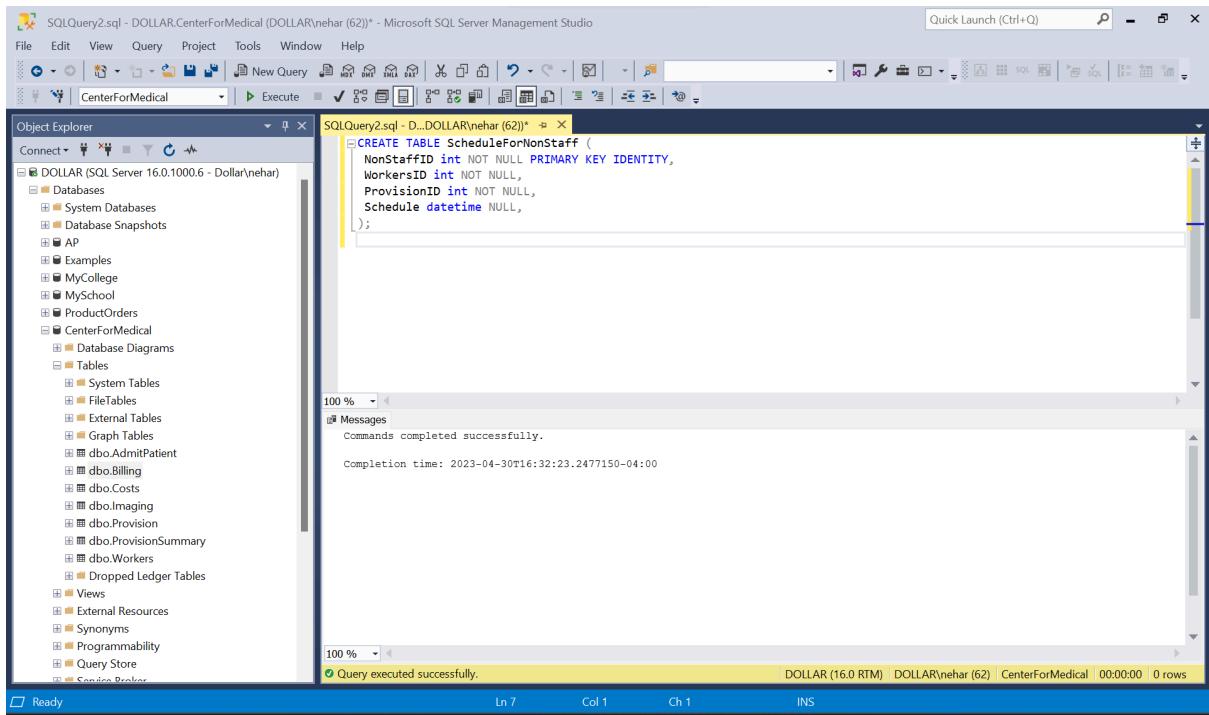
```
SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
File Connect Object Explorer CenterForMedical New Query Execute
CREATE TABLE Staff (
    WorkersID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Position varchar(50) NOT NULL,
    Status varchar(50) NOT NULL,
    Type varchar(50) NOT NULL,
);
;

100 % Messages Commands completed successfully.
Completion time: 2023-04-30T16:26:05.9042819-04:00
100 % Query executed successfully.
Ln 5 Col 30 Ch 30 INS
DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows
Ready
```

ScheduleForNonStaff:

```
CREATE TABLE ScheduleForNonStaff (
    NonStaffID int NOT NULL PRIMARY KEY IDENTITY,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    Schedule datetime NULL,
);

);
```



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```
CREATE TABLE ScheduleForNonStaff (
    NonStaffID int NOT NULL PRIMARY KEY IDENTITY,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    schedule datetime NULL,
);
```

Object Explorer

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

CenterForMedical

DOLLAR (SOL Server 16.0.1000.6 - Dollar\nehar)

Databases System Databases Database Snapshots AP Examples MyCollege MySchool ProductOrders CenterForMedical Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.AdmitPatient dbo.Billing dbo.Costs dbo.Imaging dbo.Provision dbo.ProvisionSummary dbo.Workers Dropped Ledger Tables Views External Resources Synonyms Programmability Query Store Service Broker

Messages Commands completed successfully.

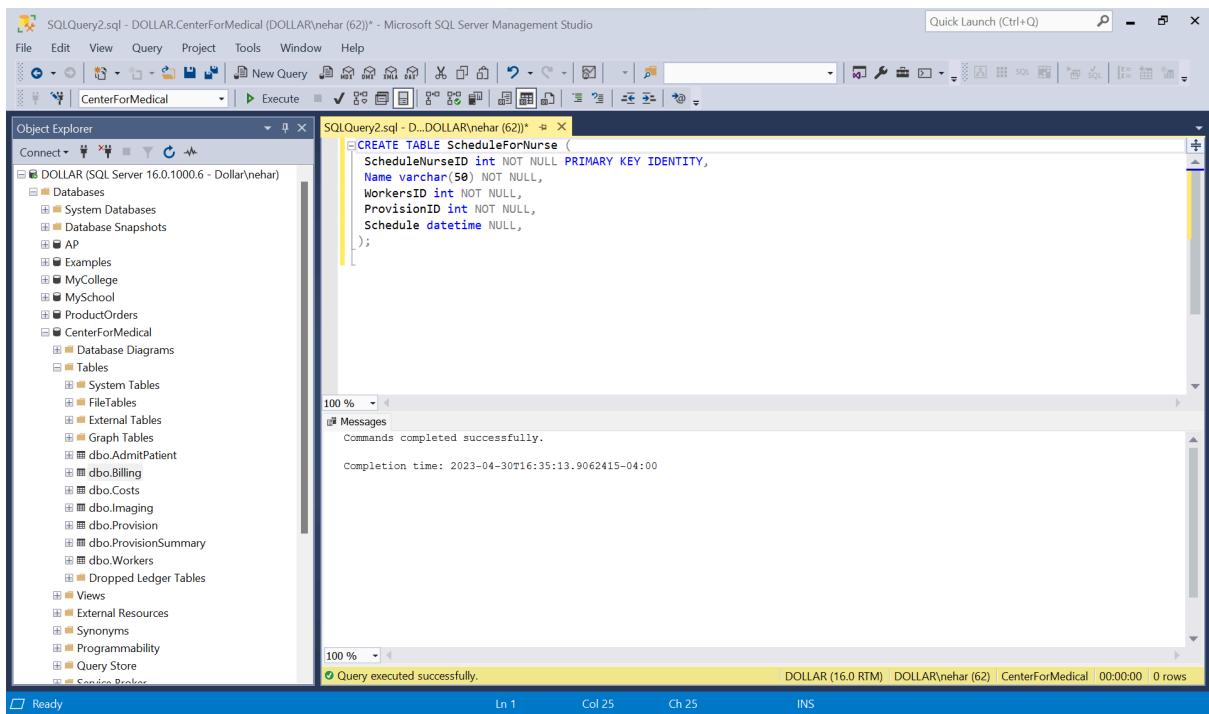
Completion time: 2023-04-30T16:32:23.2477150-04:00

Query executed successfully.

DOLLAR (16.0 RTM) DOLLAR\nehar (62) CenterForMedical 00:00:00 0 rows

ScheduleForNurse:

```
CREATE TABLE ScheduleForNurse (
    ScheduleNurseID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    Schedule datetime NULL,
);
```



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```
CREATE TABLE ScheduleForNurse (
    ScheduleNurseID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    Schedule datetime NULL,
);
```

Object Explorer

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

CenterForMedical

DOLLAR (SOL Server 16.0.1000.6 - Dollar\nehar)

Databases System Databases Database Snapshots AP Examples MyCollege MySchool ProductOrders CenterForMedical Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.AdmitPatient dbo.Billing dbo.Costs dbo.Imaging dbo.Provision dbo.ProvisionSummary dbo.Workers Dropped Ledger Tables Views External Resources Synonyms Programmability Query Store Service Broker

Messages Commands completed successfully.

Completion time: 2023-04-30T16:35:13.9062415-04:00

Query executed successfully.

DOLLAR (16.0 RTM) DOLLAR\nehar (62) CenterForMedical 00:00:00 0 rows

PatientInfo:

```
CREATE TABLE PatientInfo (
    PatientID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Address varchar(100) NULL,
    Phone varchar(50) NULL,
    PatientEmail varchar(50) NULL,
    HealthInsuranceInfo varchar(50) NULL,
    Doctor varchar(50) NOT NULL,
);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including the CenterForMedical database and its tables. In the center, the SQL Query window contains the SQL code for creating the PatientInfo table. The code is as follows:

```
CREATE TABLE PatientInfo (
    PatientID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    Address varchar(100) NULL,
    Phone varchar(50) NULL,
    PatientEmail varchar(50) NULL,
    HealthInsuranceInfo varchar(50) NULL,
    Doctor varchar(50) NOT NULL,
);
```

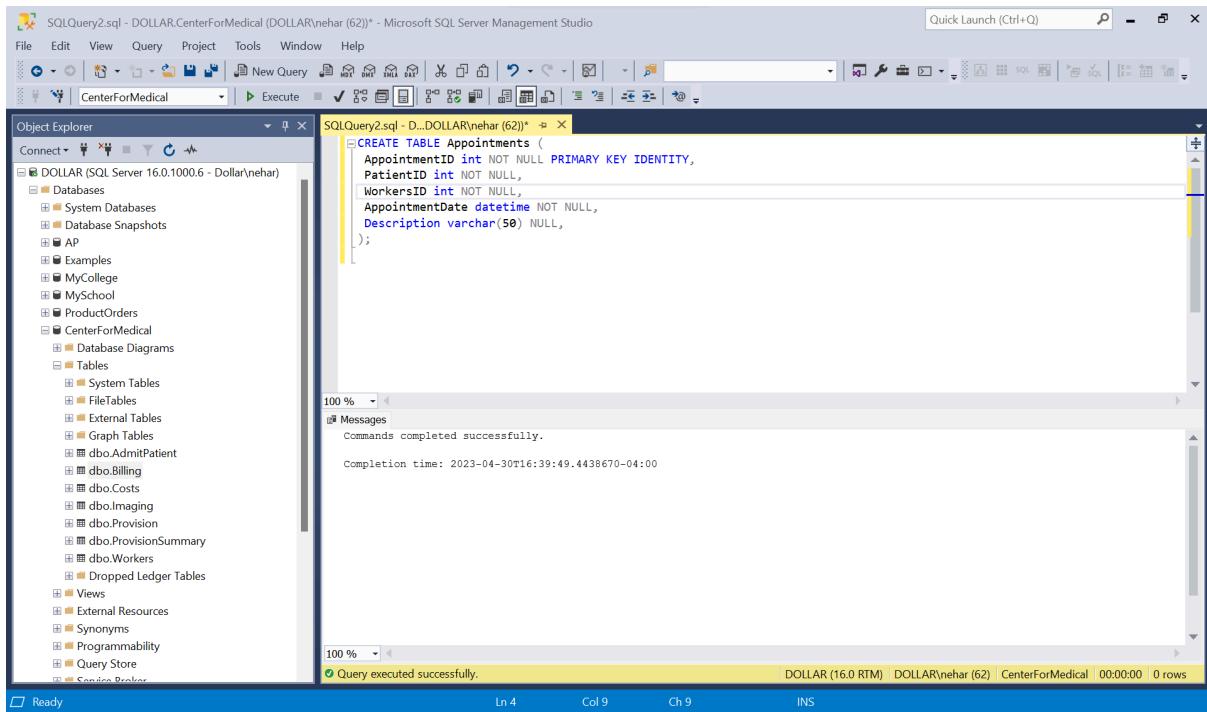
Below the code, the Messages pane shows the execution results:

- Commands completed successfully.
- Completion time: 2023-04-30T16:37:05.0578062-04:00
- Query executed successfully.

The status bar at the bottom indicates the session details: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows.

Appointments:

```
CREATE TABLE Appointments (
    AppointmentID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    WorkersID int NOT NULL,
    AppointmentDate datetime NOT NULL,
    Description varchar(50) NULL,
);
```



PatientRecords:

```

CREATE TABLE PatientRecords (
    PatientRecordID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    PatientWeight int NULL,
    PatientHeight int NULL,
    Vitals varchar(50) NULL,
    Doctor varchar(50) NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
    Discharge varchar(50) NULL,
    ProcedureCodes int NULL,
    SuggestedDoctor varchar(50) NULL,
);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the CenterForMedical database which contains several tables like AdminPatient, Billing, Costs, Imaging, Provision, and Workers. The main query window displays the following SQL code:

```

CREATE TABLE PatientRecords (
    PatientRecordID int NOT NULL PRIMARY KEY IDENTITY,
    PatientID int NOT NULL,
    PatientWeight int NULL,
    PatientHeight int NULL,
    Vitals varchar(50) NULL,
    Doctor varchar(50) NULL,
    CheckInTime datetime NULL,
    CheckOutTime datetime NULL,
    Discharge varchar(50) NULL,
    ProcedureCodes int NULL,
    SuggestedDoctor varchar(50) NULL,
);

```

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

Medical:

```

CREATE TABLE Medical (
    MedicalID int NOT NULL PRIMARY KEY IDENTITY,
    MedicalName varchar(50) NOT NULL,
    PatientMedicines varchar(50) NULL,
    RentalEquipment varchar(50) NULL,
);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the CenterForMedical database which contains several tables like AdminPatient, Billing, Costs, Imaging, Provision, and Workers. The main query window displays the following SQL code:

```

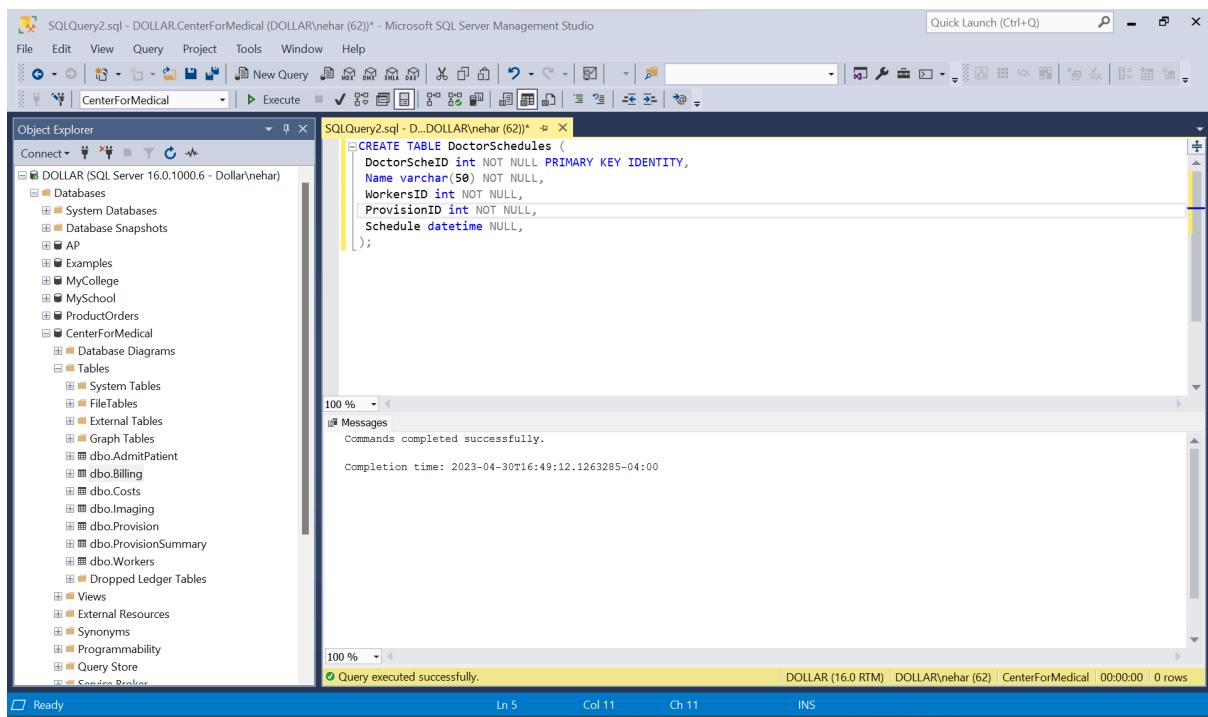
CREATE TABLE Medical (
    MedicalID int NOT NULL PRIMARY KEY IDENTITY,
    MedicalName varchar(50) NOT NULL,
    PatientMedicines varchar(50) NULL,
    RentalEquipment varchar(50) NULL,
);

```

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

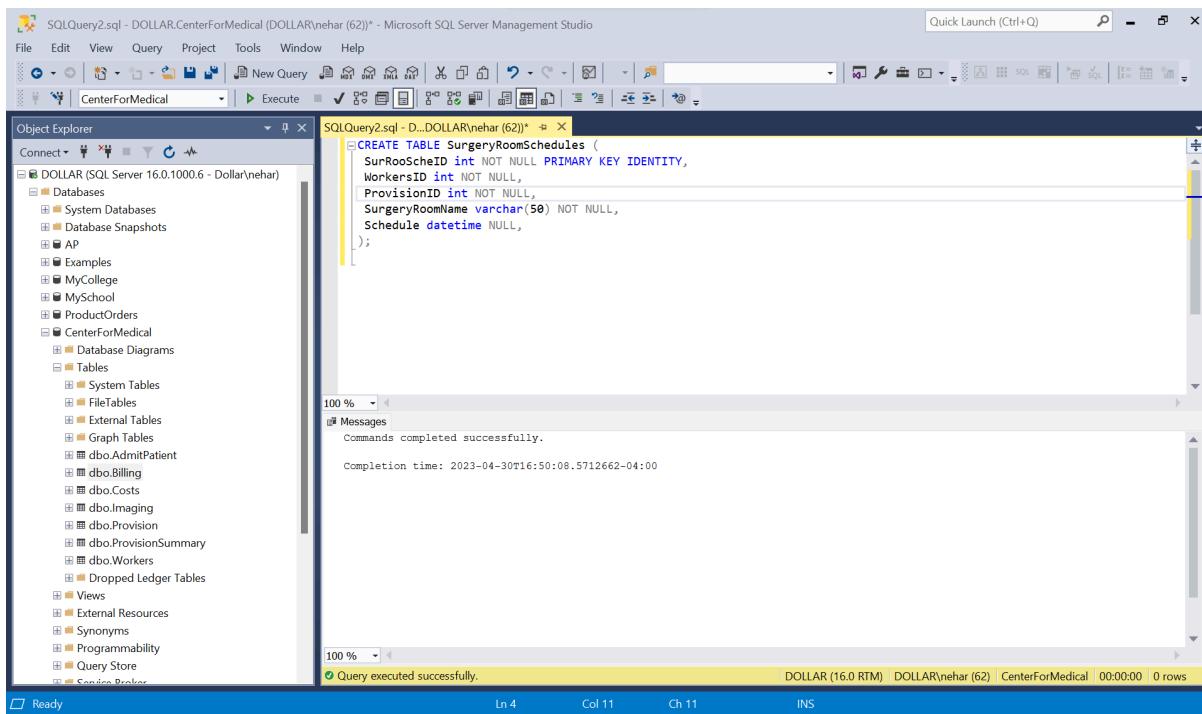
DoctorSchedules:

```
CREATE TABLE DoctorSchedules (
    DoctorScheID int NOT NULL PRIMARY KEY IDENTITY,
    Name varchar(50) NOT NULL,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    Schedule datetime NULL,
);
```



SurgeryRoomSchedules:

```
CREATE TABLE SurgeryRoomSchedules (
    SurRooScheID int NOT NULL PRIMARY KEY IDENTITY,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    SurgeryRoomName varchar(50) NOT NULL,
    Schedule datetime NULL,
);
```



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

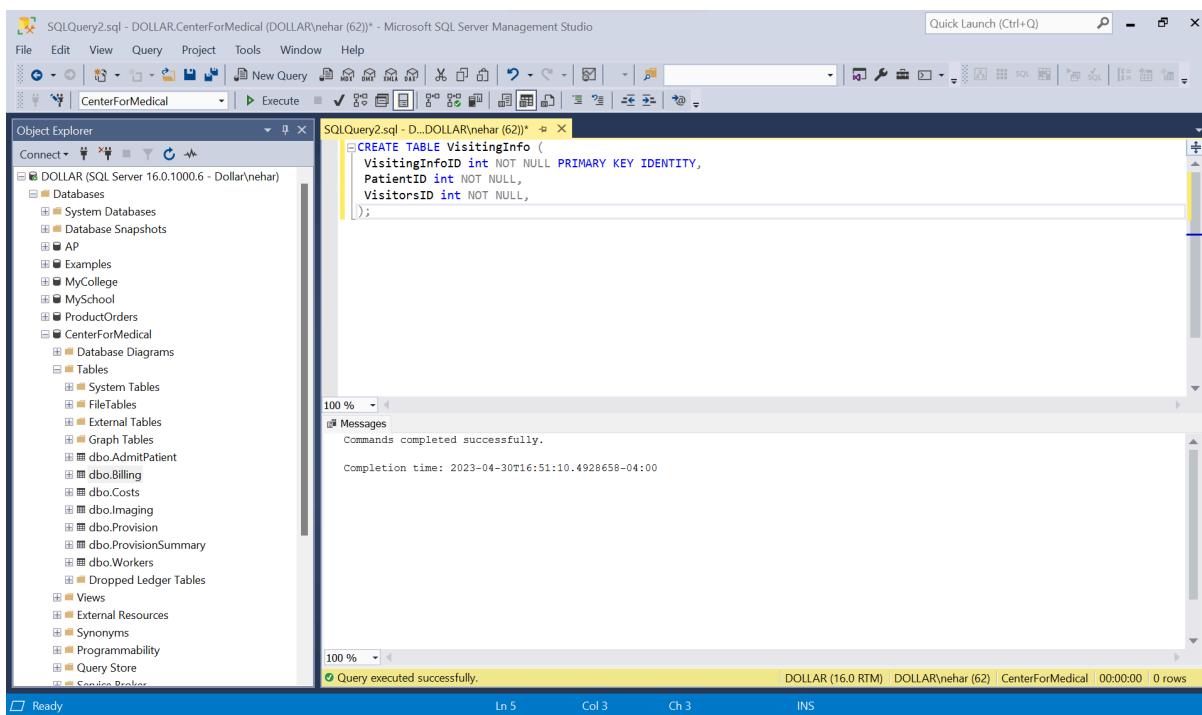
```
CREATE TABLE SurgeryRoomSchedules (
    SurRo ScheID int NOT NULL PRIMARY KEY IDENTITY,
    WorkersID int NOT NULL,
    ProvisionID int NOT NULL,
    SurgeryRoomName varchar(50) NOT NULL,
    Schedule datetime NULL,
);
```

Messages
Commands completed successfully.
Completion time: 2023-04-30T16:50:08.5712662-04:00

Query executed successfully.

VisitingInfo:

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



SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

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

Messages
Commands completed successfully.
Completion time: 2023-04-30T16:51:10.4928658-04:00

Query executed successfully.

Visitors:

```
CREATE TABLE Visitors (
```

```

VisitorsID int NOT NULL PRIMARY KEY IDENTITY,
VisitorsName varchar(50) NOT NULL,
IDNo varchar(50) NOT NULL,
EntryTime datetime NULL,
ExitTime datetime NULL,
DOB datetime NULL,
Picture image NULL,
Visits varchar(50) NOT NULL,
);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'CenterForMedical'. The 'Tables' node under this database is expanded, showing various table names like 'AdmitPatient', 'Billing', 'Costs', etc. The central pane displays the SQL code for creating the 'Visitors' table:

```

CREATE TABLE Visitors (
    VisitorsID int NOT NULL PRIMARY KEY IDENTITY,
    VisitorsName varchar(50) NOT NULL,
    IDNo varchar(50) NOT NULL,
    EntryTime datetime NULL,
    ExitTime datetime NULL,
    DOB datetime NULL,
    Picture image NULL,
    Visits varchar(50) NOT NULL,
);

```

Below the code, the message 'Commands completed successfully.' is displayed, along with the completion time: 'Completion time: 2023-04-30T16:53:03.7022758-04:00'. At the bottom of the central pane, it says 'Query executed successfully.' The status bar at the bottom indicates 'Ln 4 Col 6 Ch 6 INS'.

ALTER TABLE Billing

```

ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE Costs
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE ApparatusReservation
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE ProvisionSummary
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE ProvisionSummary
ADD FOREIGN KEY (ImagingID) REFERENCES Imaging (ImagingID);
ALTER TABLE ProvisionSummary
ADD FOREIGN KEY (LabID) REFERENCES Lab (LabID);
ALTER TABLE ProvisionSummary
ADD FOREIGN KEY (MedicalID) REFERENCES Medical (MedicalID);
ALTER TABLE AdmitPatient
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);

```

```
ALTER TABLE AdmitPatient
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE CoverageOfInsurance
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE Staff
ADD FOREIGN KEY (WorkersID) REFERENCES Workers (WorkersID);
ALTER TABLE ScheduleForNurse
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE ScheduleForNurse
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE Appointments
ADD FOREIGN KEY (WorkersID) REFERENCES Workers (WorkersID);
ALTER TABLE Appointments
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE PatientRecords
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE DoctorSchedules
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE DoctorSchedules
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE ScheduleForNonStaff
ADD FOREIGN KEY (WorkersID) REFERENCES Workers (WorkersID);
ALTER TABLE ScheduleForNonStaff
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE VisitingInfo
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE VisitingInfo
ADD FOREIGN KEY (VisitorsID) REFERENCES Visitors (VisitorsID);
```

```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Connect Connect to CenterForMedical Execute
Object Explorer
SQLQuery2.sql - D...DOLLAR\nehar (62)*
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE DoctorSchedules
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE SurgeryRoomSchedules
ADD FOREIGN KEY (WorkersID) REFERENCES Staff (WorkersID);
ALTER TABLE ScheduleForNonStaff
ADD FOREIGN KEY (WorkersID) REFERENCES Workers (WorkersID);
ALTER TABLE ScheduleForNonStaff
ADD FOREIGN KEY (ProvisionID) REFERENCES Provision (ProvisionID);
ALTER TABLE VisitingInfo
ADD FOREIGN KEY (PatientID) REFERENCES PatientInfo (PatientID);
ALTER TABLE VisitingInfo
ADD FOREIGN KEY (VisitorsID) REFERENCES Visitors (VisitorsID);

100 %
Messages
Commands completed successfully.

Completion time: 2023-04-30T17:28:24.4275759-04:00

100 %
Query executed successfully.
Ln 48 Col 58 Ch 58 INS
DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows

```

Insertion OF DATA

Table : PatientInfo

```
INSERT INTO PatientInfo (PatientID, Name, Address, Phone, PatientEmail, HealthInsuranceInfo, Doctor)
```

VALUES

```
(11, 'Stephanie Smith', '321 Oak Lane', '5555551212', 'stephanie.smith@gmail.com', 'Aetna', 'John Kim'),
(12, 'Ryan Brown', '456 Elm Street', '5555553434', 'ryan.brown@gmail.com', 'Blue Cross', 'Emily Nguyen'),
(13, 'Michelle Rodriguez', '789 Maple Road', '5555555656', 'michelle.rodriguez@gmail.com', 'Cigna', 'John Kim'),
(14, 'William Chen', '234 Cedar Street', '5555557878', 'william.chen@gmail.com', 'United Healthcare', 'Emily Nguyen'),
(15, 'Amanda Wilson', '567 Pine Avenue', '5555559090', 'amanda.wilson@gmail.com', 'Aetna', 'John Kim'),
(16, 'Steven Kim', '890 Walnut Street', '5555552323', 'steven.kim@gmail.com', 'Blue Cross', 'Emily Nguyen'),
(17, 'Rachel Davis', '901 Spruce Street', '5555554545', 'rachel.davis@gmail.com', 'Cigna', 'John Kim'),
(18, 'Ethan Nguyen', '345 Cedar Road', '5555556767', 'ethan.nguyen@gmail.com', 'United Healthcare', 'Emily Nguyen'),
(19, 'Olivia Patel', '678 Birch Lane', '5555558989', 'olivia.patel@gmail.com', 'Aetna', 'John Kim'),
(20, 'Brandon Lee', '234 Oak Road', '5555550101', 'brandon.lee@gmail.com', 'Blue Cross', 'Emily Nguyen');
```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer ▾ ×

CenterForMedical

Connect ▾

SQL Database Tables FileTable External Graph T dbo.Ad dbo.Ap dbo.Ap dbo.Bill dbo.Co dbo.Co dbo.Do dbo.Imc dbo.Lab dbo.Me dbo.Pat dbo.Pat dbo.Pro dbo.Sch dbo.Sch dbo.Sta dbo.Sur dbo.Visi dbo.Visi dbo.Wc Droppe Diagrams

CenterForMe

New Query Find Replace All Open Recent SQL SQL Scripts

Execute

SET IDENTITY_INSERT PatientInfo ON
INSERT INTO PatientInfo (PatientID, Name, Address, Phone, PatientEmail, HealthInsuranceInfo, Doctor)
VALUES
(1, 'Stephanie Smith', '321 Oak Lane', '5555551212', 'stephanie.smith@gmail.com', 'Aetna', 'John Kim'),
(2, 'Ryan Brown', '456 Elm Street', '5555553434', 'ryan.brown@gmail.com', 'Blue Cross', 'Emily Nguyen'),
(3, 'Michelle Rodriguez', '789 Maple Road', '5555556565', 'michelle.rodriguez@gmail.com', 'Cigna', 'John Kim'),
(4, 'William Chen', '234 Cedar Street', '5555557878', 'william.chen@gmail.com', 'United Healthcare', 'Emily Nguyen'),
(5, 'Amanda Wilson', '567 Pine Avenue', '5555559090', 'amanda.wilson@gmail.com', 'Aetna', 'John Kim'),
(6, 'Steven Kim', '890 Walnut Street', '5555552232', 'steven.kim@gmail.com', 'Blue Cross', 'Emily Nguyen'),
(7, 'Rachel Davis', '901 Spruce Street', '5555554545', 'rachel.davis@gmail.com', 'Cigna', 'John Kim'),
(8, 'Ethan Nguyen', '345 Cedar Road', '5555556767', 'ethan.nguyen@gmail.com', 'United Healthcare', 'Emily Nguyen'),
(9, 'Olivia Patel', '678 Birch Lane', '5555558989', 'olivia.patel@gmail.com', 'Aetna', 'John Kim'),
(10, 'Brandon Lee', '234 Oak Road', '5555550101', 'brandon.lee@gmail.com', 'Blue Cross', 'Emily Nguyen')
SET IDENTITY_INSERT PatientInfo OFF

100 %

Messages

(10 rows affected)

Completion time: 2023-04-30T17:56:35.4707071-04:00

100 %

Query executed successfully.

DOLLAR (16.0 RTM) DOLLAR\nehar (62) CenterForMedical 00:00:00 0 rows

Table:Billing

```
INSERT INTO Billing (BillingID, PatientID, Visits, MedicalBillingCodes, Payor, PaymentMethod)
```

VALUES

(1, 1, 'Regular visits', 'L78945612', 'Blue Cross', 'Check')

(2, 2, 'Regular visits', 'P36985214', 'Cigna', 'Card'),

(3, 3, 'Regular visits', 'A25836914', 'Aetna', 'Check'),

(4, 4, 'Good visits', 'L27021613', 'United Healthcare',

(5, 5, 'Good visits', 'P20513244', 'Blue Cross', 'Card'),

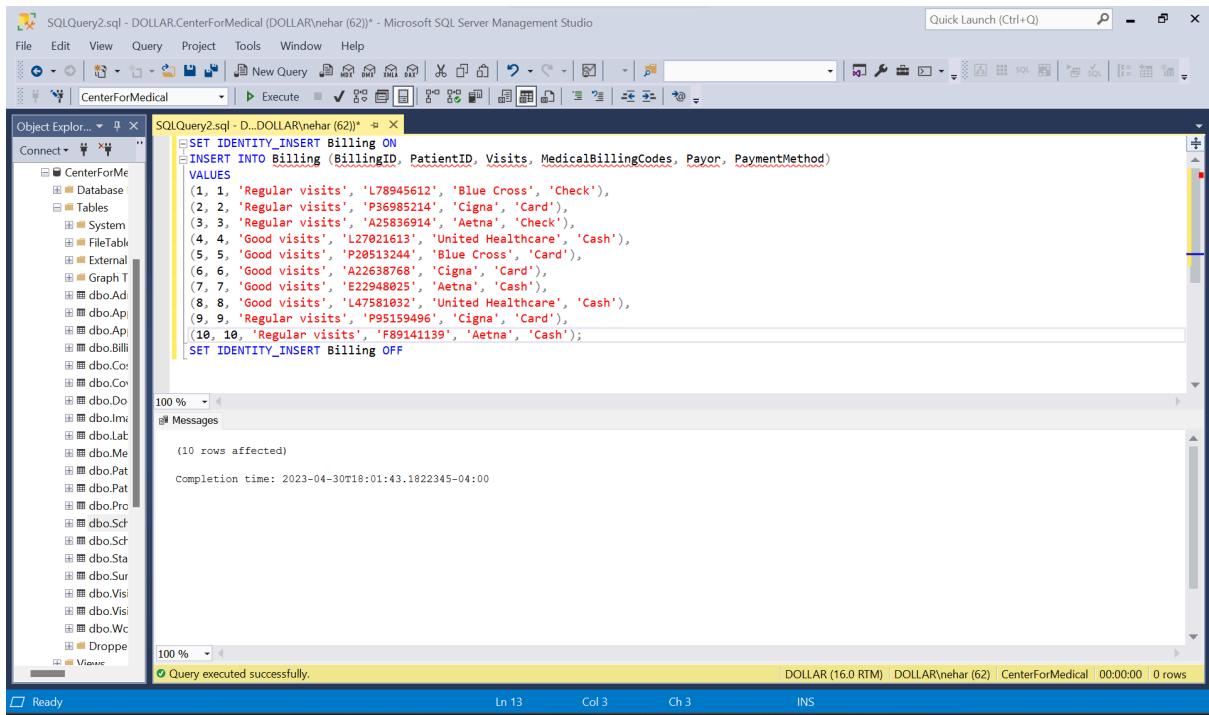
(6, 6, 'Good visits', 'A22638768', 'Cigna', 'Card'),

(7, 7, 'Good visits', 'E22948025', 'Aetna', 'Cash'),

(8, 8, 'Good visits', 'L47581032', 'United Healthca

(9, 9, 'Regular visits', 'P95159496', 'Cigna', 'Card

(10, 10, 'Regular visits', 'F89141139', 'Aetna', 'Cash');



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```

SET IDENTITY_INSERT Billing ON
INSERT INTO Billing (BillingID, PatientID, Visits, MedicalBillingCodes, Payer, PaymentMethod)
VALUES
(1, 1, 'Regular visits', 'L78945612', 'Blue Cross', 'Check'),
(2, 2, 'Regular visits', 'P36985214', 'Cigna', 'Card'),
(3, 3, 'Regular visits', 'A25836914', 'Aetna', 'Check'),
(4, 4, 'Good visits', 'L27021613', 'United Healthcare', 'Cash'),
(5, 5, 'Good visits', 'P28513244', 'Blue Cross', 'Card'),
(6, 6, 'Good visits', 'A22638768', 'Cigna', 'Card'),
(7, 7, 'Good visits', 'E22948025', 'Aetna', 'Cash'),
(8, 8, 'Good visits', 'L47581032', 'United Healthcare', 'Cash'),
(9, 9, 'Regular visits', 'P95159496', 'Cigna', 'Card'),
(10, 10, 'Regular visits', 'F89141139', 'Aetna', 'Cash');

SET IDENTITY_INSERT Billing OFF
  
```

The message pane below the query window shows "(10 rows affected)". The status bar at the bottom indicates "Query executed successfully." and provides details about the session: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows.

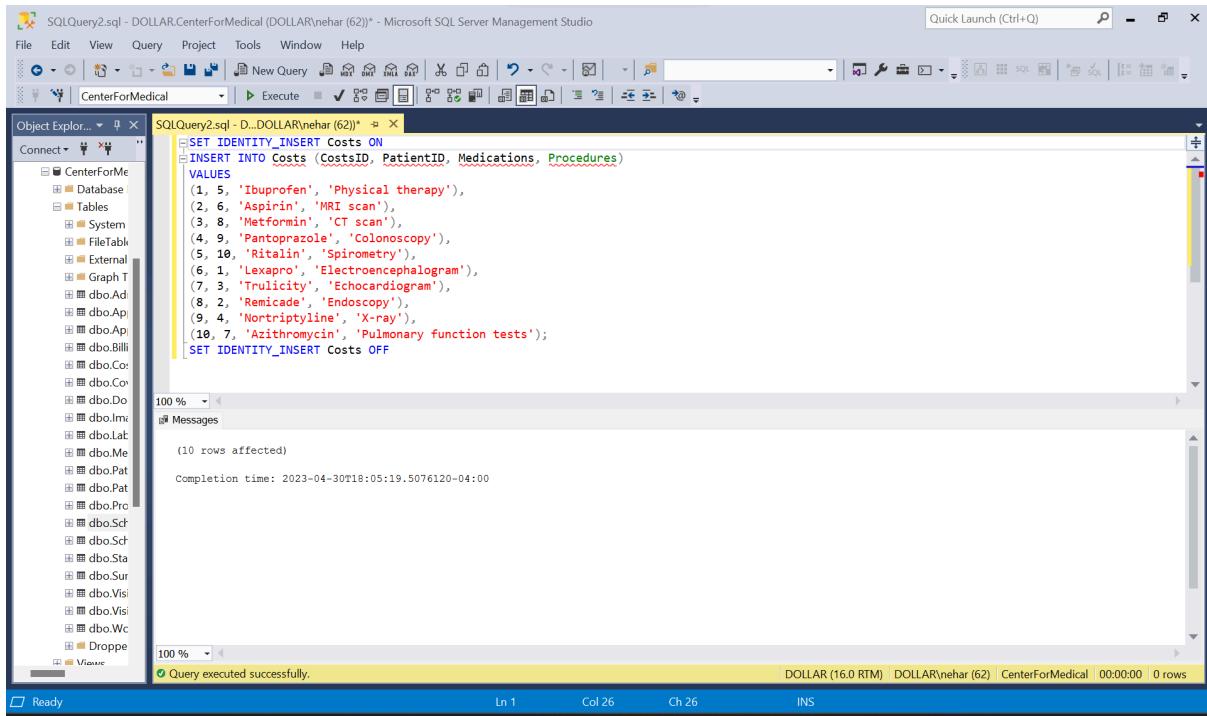
Table: Costs

INSERT INTO Costs (CostsID, PatientID, Medications, Procedures)

VALUES

```

(1, 5, 'Ibuprofen', 'Physical therapy'),
(2, 6, 'Aspirin', 'MRI scan'),
(3, 8, 'Metformin', 'CT scan'),
(4, 9, 'Pantoprazole', 'Colonoscopy'),
(5, 10, 'Ritalin', 'Spirometry'),
(6, 1, 'Lexapro', 'Electroencephalogram'),
(7, 3, 'Trulicity', 'Echocardiogram'),
(8, 2, 'Remicade', 'Endoscopy'),
(9, 4, 'Nortriptyline', 'X-ray'),
(10, 7, 'Azithromycin', 'Pulmonary function tests');
  
```



```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q) 
CenterForMedical Execute 
SET IDENTITY_INSERT Costs ON
INSERT INTO Costs (CostsID, PatientID, Medications, Procedures)
VALUES
(1, 5, 'Ibuprofen', 'Physical therapy'),
(2, 6, 'Aspirin', 'MRI scan'),
(3, 8, 'Metformin', 'CT scan'),
(4, 9, 'Pantoprazole', 'Colonoscopy'),
(5, 10, 'Ritalin', 'Spirometry'),
(6, 1, 'Lexapro', 'Electroencephalogram'),
(7, 3, 'Trulicity', 'Echocardiogram'),
(8, 2, 'Remicade', 'Endoscopy'),
(9, 4, 'Nortriptyline', 'X-ray'),
(10, 7, 'Azithromycin', 'Pulmonary function tests');
SET IDENTITY_INSERT Costs OFF

100 % 
Messages
(10 rows affected)
Completion time: 2023-04-30T18:05:19.5076120-04:00

100 % 
Query executed successfully. DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 0 rows
Ln 1 Col 26 Ch 26 INS

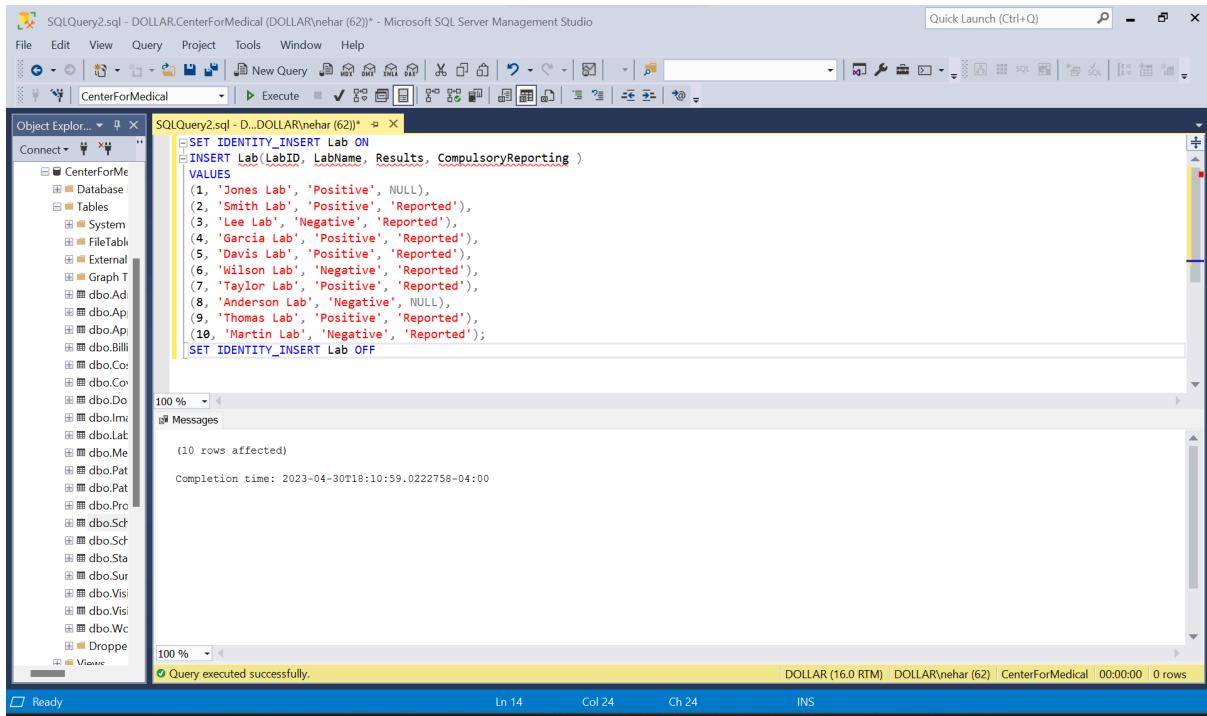
```

Table:Lab

```

SET IDENTITY_INSERT Lab ON
INSERT Lab(LabID, LabName, Results, CompulsoryReporting )
VALUES
(1, 'Jones Lab', 'Positive', NULL),
(2, 'Smith Lab', 'Positive', 'Reported'),
(3, 'Lee Lab', 'Negative', 'Reported'),
(4, 'Garcia Lab', 'Positive', 'Reported'),
(5, 'Davis Lab', 'Positive', 'Reported'),
(6, 'Wilson Lab', 'Negative', 'Reported'),
(7, 'Taylor Lab', 'Positive', 'Reported'),
(8, 'Anderson Lab', 'Negative', NULL),
(9, 'Thomas Lab', 'Positive', 'Reported'),
(10, 'Martin Lab', 'Negative', 'Reported');
SET IDENTITY_INSERT Lab OFF

```



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the 'CenterForMed' database selected. The right pane contains a query window titled 'SQLQuery2.sql - D...DOLLAR\nehar (62)*'. The query is:

```

SET IDENTITY_INSERT Lab ON
INSERT Lab(LabID, LabName, Results, CompulsoryReporting)
VALUES
(1, 'Jones Lab', 'Positive', NULL),
(2, 'Smith Lab', 'Positive', 'Reported'),
(3, 'Lee Lab', 'Negative', 'Reported'),
(4, 'Garcia Lab', 'Positive', 'Reported'),
(5, 'Davis Lab', 'Positive', 'Reported'),
(6, 'Wilson Lab', 'Negative', 'Reported'),
(7, 'Taylor Lab', 'Positive', 'Reported'),
(8, 'Anderson Lab', 'Negative', NULL),
(9, 'Thomas Lab', 'Positive', 'Reported'),
(10, 'Martin Lab', 'Negative', 'Reported');
SET IDENTITY_INSERT Lab OFF
  
```

The status bar at the bottom indicates 'Query executed successfully.' and shows the completion time as 2023-04-30T18:10:59.0222758-04:00.

TABLE: Provision

INSERT Provision(ProvisionID, ProvisionName, Location, Departments, CareCapacity, MedicalEquipment, OfficeHours, RoomCapacity)

VALUES

```

(1, 'General Ward', 'Building A', 'Internal Medicine', 20, 'ECG Machine', '2023-05-01 08:00:00', 30),
(2, 'Pediatric Ward', 'Building B', 'Pediatrics', 15, 'Oxygen Tank', '2023-05-01 07:00:00', 20),
(3, 'Cardiac Care Unit', 'Building C', 'Cardiology', 10, 'Defibrillator', '2023-05-01 09:00:00', 12),
(4, 'Surgical Ward', 'Building A', 'Surgery', 18, 'Surgical Tools', '2023-05-01 10:00:00', 25),
(5, 'Maternity Ward', 'Building B', 'Obstetrics', 12, 'Ultrasound Machine', '2023-05-01 11:00:00', 18),
(6, 'Intensive Care Unit', 'Building C', 'Critical Care', 8, 'Ventilator', '2023-05-01 12:00:00', 10),
(7, 'Psychiatric Ward', 'Building A', 'Psychiatry', 6, 'Medication Dispenser', '2023-05-01 13:00:00', 8),
(8, 'Rehabilitation Unit', 'Building B', 'Physical Therapy', 15, 'Treadmill', '2023-05-01 14:00:00', 20),
(9, 'Emergency Department', 'Building C', 'Emergency Medicine', 25, 'CT Scanner', '2023-05-01 15:00:00', 40),
(10, 'Oncology Ward', 'Building A', 'Oncology', 8, 'Chemotherapy Pump', '2023-05-01 16:00:00', 12);
  
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62))* - Microsoft SQL Server Management Studio". The main area displays the following T-SQL code:

```

SET IDENTITY_INSERT Provision ON
INSERT Provision(ProvisionID, ProvisionName, Location, Departments, CareCapacity, MedicalEquipment, OfficeHours, RoomCapacity)
VALUES
(1, 'General Ward', 'Building A', 'Internal Medicine', 20, 'ECG Machine', '2023-05-01 08:00:00', 30),
(2, 'Pediatric Ward', 'Building B', 'Pediatrics', 15, 'Oxygen Tank', '2023-05-01 07:00:00', 20),
(3, 'Cardiac Care Unit', 'Building C', 'Cardiology', 10, 'Defibrillator', '2023-05-01 09:00:00', 12),
(4, 'Surgical Ward', 'Building A', 'Surgery', 18, 'Surgical Tools', '2023-05-01 10:00:00', 25),
(5, 'Maternity Ward', 'Building B', 'Obstetrics', 12, 'Ultrasound Machine', '2023-05-01 11:00:00', 18),
(6, 'Intensive Care Unit', 'Building C', 'Critical Care', 8, 'Ventilator', '2023-05-01 12:00:00', 10),
(7, 'Psychiatric Ward', 'Building A', 'Psychiatry', 6, 'Medication Dispenser', '2023-05-01 13:00:00', 8),
(8, 'Rehabilitation Unit', 'Building B', 'Physical Therapy', 15, 'Treadmill', '2023-05-01 14:00:00', 20),
(9, 'Emergency Department', 'Building C', 'Emergency Medicine', 25, 'CT Scanner', '2023-05-01 15:00:00', 40),
(10, 'Oncology Ward', 'Building A', 'Oncology', 8, 'Chemotherapy Pump', '2023-05-01 16:00:00', 12)
SET IDENTITY_INSERT Provision OFF
  
```

The results pane shows "(10 rows affected)" and the completion time "Completion time: 2023-04-30T18:28:00.4097744-04:00". The status bar at the bottom indicates "DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows".

TABLE:ApparatusReservation

```

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

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

```

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

(10 rows affected)

Completion time: 2023-04-30T18:38:02.5699988-04:00

Query executed successfully.

Table :Imaging

```

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

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio". The main area displays the following SQL script:

```

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

```

Below the script, the status bar shows "100 %", "Messages", "(10 rows affected)", and "Completion time: 2023-04-30T18:40:42.9630381-04:00". At the bottom, a yellow bar indicates "Query executed successfully." The footer shows "DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows".

TABLE MEDICAL:

```

SET IDENTITY_INSERT Medical ON
INSERT Medical (MedicalID, MedicalName, PatientMedicines, RentalEquipment)VALUES
(1, 'Walgreens', 'Amoxicillin, Aspirin, Atorvastatin', 'Defibrillators'),
(2, 'Rite Aid', 'Ibuprofen, Insulin, Isosorbide', 'Anesthesia Machines'),
(3, 'CVS Pharmacy', 'Naproxen, Nitrofurantoin, Nortriptyline', 'Patient Monitors'),
(4, 'Walmart Pharmacy', 'Losartan, Lovastatin, Lurasidone', 'EKG/ECG Machines'),
(5, 'CVS Pharmacy', 'Metformin, Methadone, Methocarbamol', 'Sterilizers'),
(6, 'Walgreens', 'Methotrexate, Methylprednisolone, Midazolam', 'Surgical Tables'),
(7, 'Rite Aid', 'Omeprazole, Ondansetron, Oxycodone', 'Blanket and Fluid Warmers'),
(8, 'Walmart Pharmacy', 'Propranolol, Pyridostigmine, Quetiapine', 'Electrosurgical Units'),
(9, 'Walgreens', 'Rabeprazole, Ranitidine, Rifaximin', 'Surgical Tables'),
(10, 'CVS Pharmacy', 'Simvastatin, Spironolactone, Sucralfate', 'EKG/ECG Machines')

```

```
SET IDENTITY_INSERT Medical OFF
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio". The main area contains the following SQL code:

```

SET IDENTITY_INSERT Medical ON
INSERT Medical (MedicalID, MedicalName, PatientMedicines, RentalEquipment) VALUES
(1, 'Walgreens', 'Amoxicillin, Aspirin, Atorvastatin', 'Defibrillators'),
(2, 'Rite Aid', 'Ibuprofen, Insulin, Isosorbide', 'Anesthesia Machines'),
(3, 'CVS Pharmacy', 'Naproxen, Nitrofurantoin, Nortriptyline', 'Patient Monitors'),
(4, 'Walmart Pharmacy', 'Losartan, Lovastatin, Lurasidone', 'EKG/ECG Machines'),
(5, 'CVS Pharmacy', 'Metformin, Methadone, Methocarbamol', 'Sterilizers'),
(6, 'Walgreens', 'Methotrexate, Methylprednisolone, Midazolam', 'Surgical Tables'),
(7, 'Rite Aid', 'Omeprazole, Ondansetron, Oxycodone', 'Blanket and Fluid Warmers'),
(8, 'Walmart Pharmacy', 'Propranolol, Pyridostigmine, Quetiapine', 'Electrosurgical Units'),
(9, 'Walgreens', 'Rabeprazole, Ranitidine, Rifaximin', 'Surgical Tables'),
(10, 'CVS Pharmacy', 'Simvastatin, Spironolactone, Sucralfate', 'EKG/ECG Machines')

SET IDENTITY_INSERT Medical OFF

```

The status bar at the bottom indicates "Query executed successfully." and shows statistics: Ln 14, Col 28, Ch 28, INS.

Table: Provisional Summary:

```

SET IDENTITY_INSERT ProvisionSummary ON
INSERT INTO ProvisionSummary(ProvisionSummaryID, ProvisionID, ImagingID, MedicalID,
LabID)
VALUES
(1, 1, 4, 1, 6),
(2, 2, 5, 3, 9),
(3, 3, 2, 5, 7),
(4, 4, 8, 10, 2),
(5, 5, 9, 4, 1),
(6, 6, 6, 7, 5),
(7, 7, 3, 6, 4),
(8, 8, 2, 2, 6),
(9, 9, 7, 9, 3),
(10, 10, 10, 8, 10);
SET IDENTITY_INSERT ProvisionSummary OFF

```

```

SET IDENTITY_INSERT ProvisionSummary ON
INSERT INTO ProvisionSummary(ProvisionSummaryID, ProvisionID, ImagingID, MedicalID, LabID)
VALUES
(1, 1, 4, 1, 6),
(2, 2, 5, 3, 9),
(3, 3, 2, 5, 7),
(4, 4, 8, 10, 2),
(5, 5, 9, 4, 1),
(6, 6, 6, 7, 5),
(7, 7, 3, 6, 4),
(8, 8, 2, 2, 6),
(9, 9, 7, 9, 3),
(10, 10, 10, 8, 10);
SET IDENTITY_INSERT ProvisionSummary OFF
  
```

(10 rows affected)

Completion time: 2023-04-30T18:52:58.5275718-04:00

Query executed successfully.

TABLE Workers:

```

SET IDENTITY_INSERT Workers ON
INSERT Workers(WorkersID, Name, Address, Reviews, Department, Office, WorkersPhone,
Email, WorkersSalary, Benefits, ContractType)
VALUES
(1, 'John Doe', '123 Main St, Anytown USA', '4.5/5', 'Sales', 'Office A', '555-1234',
'johndoe@example.com', '$50,000', 'Health Insurance', 'Full-Time'),
(2, 'Jane Smith', '456 Oak St, Anytown USA', '4/5', 'Marketing', 'Office B', '555-5678',
'janessmith@example.com', '$60,000', NULL, 'Full-Time'),
(3, 'Bob Johnson', '789 Maple St, Anytown USA', '3.5/5', 'IT', 'Office C', '555-9012',
'bobjohnson@example.com', '$80,000', '401k', 'Full-Time'),
(4, 'Sara Lee', '246 Elm St, Anytown USA', '4.5/5', 'Customer Service', 'Office D', '555-3456',
'saralee@example.com', '$45,000', 'Vacation Time', 'Part-Time'),
(5, 'Tom Smith', '135 Pine St, Anytown USA', '4/5', 'Finance', 'Office A', '555-7890',
'tomsmith@example.com', '$75,000', NULL, 'Full-Time'),
(6, 'Mary Johnson', '579 Cedar St, Anytown USA', '3/5', 'Human Resources', 'Office C',
'555-2345', 'maryjohnson@example.com', '$70,000', 'Health Insurance', 'Full-Time'),
(7, 'Bill Davis', '802 Walnut St, Anytown USA', '4/5', 'Sales', 'Office B', '555-6789',
'billdavis@example.com', '$55,000', '401k', 'Full-Time'),
(8, 'Emily Brown', '931 Cherry St, Anytown USA', '5/5', 'Marketing', 'Office D', '555-0123',
'emilybrown@example.com', '$65,000', 'Vacation Time', 'Full-Time'),
(9, 'Chris Green', '246 Oak St, Anytown USA', '3.5/5', 'IT', 'Office A', '555-4567',
'chrisgreen@example.com', '$90,000', NULL, 'Full-Time'),
(10, 'Lisa Johnson', '468 Pine St, Anytown USA', '4/5', 'Finance', 'Office C', '555-8901',
'lisajohnson@example.com', '$80,000', 'Health Insurance', 'Full-Time'),
(11, 'Mike Smith', '789 Cedar St, Anytown USA', '4.5/5', 'Customer Service', 'Office D',
'555-2345', 'mikesmith@example.com', '$50,000', NULL, 'Full-Time'),
(12, 'Karen Lee', '468 Oak St, Anytown USA', '4/5', 'Sales', 'Office B', '555-6789',
'karenlee@example.com', '$60,000', '401k', 'Full-Time'),
  
```

(13, 'David Johnson', '931 Maple St, Anytown USA', '3.5/5', 'Marketing', 'Office C', '555-0123', 'davidjohnson@example.com', '\$80,000', 'Vacation Time', 'Full-Time'),
(14, 'Amy Davis', '802 Pine St, Anytown USA', '4.5/5', 'IT', 'Office A', '555-4567', 'amydavis@example.com', '\$45,000', NULL, 'Part-Time'),
(15, 'Mark Green', '579 Elm St, Anytown USA', '4/5', 'Finance', 'Office D', '555-8901', 'markgreen@example.com', '\$75,000', 'Health Insurance', 'Full-Time'),
(16, 'Rachel Johnson', '123 Main St, Anytown USA', '3/5', 'Human Resources', 'Office C', '555-1234', 'racheljohnson@example.com', '\$70,000', '401k', 'Full-Time'),
(17, 'Brian Brown', '246 Oak St, Anytown USA', '4/5', 'Sales', 'Office B', '555-5678', 'brianbrown@example.com', '\$55,000', 'Vacation Time', 'Full-Time'),
(18, 'Sarah Lee', '468 Cedar St, Anytown USA', '5/5', 'Marketing', 'Office A', '555-9012', 'sarahlee@example.com', '\$65,000', NULL, 'Full-Time'),
(19, 'Jason Smith', '802 Pine St, Anytown USA', '3.5/5', 'IT', 'Office C', '555-3456', 'jasonsmith@example.com', '\$90,000', 'Health Insurance', 'Full-Time'),
(20, 'Katie Johnson', '135 Maple St, Anytown USA', '4/5', 'Finance', 'Office D', '555-7890', 'katiejohnson@example.com', '\$80,000', '401k', 'Full-Time'),
(21, 'Adam Rodriguez', '246 Cherry St, Anytown USA', '4/5', 'Sales', 'Office B', '555-2345', 'adamrodriguez@example.com', '\$57,000', 'Vacation Time', 'Full-Time'),
(22, 'Sarah Lee', '468 Oak St, Anytown USA', '3/5', 'Customer Service', 'Office D', '555-6789', 'sarahlee@example.com', '\$40,000', NULL, 'Part-Time'),
(23, 'David Miller', '579 Elm St, Anytown USA', '4/5', 'IT', 'Office A', '555-0123', 'davidmiller@example.com', '\$85,000', '401k', 'Full-Time'),
(24, 'Ella Scott', '802 Cedar St, Anytown USA', '4.5/5', 'Marketing', 'Office B', '555-4567', 'ellascott@example.com', '\$70,000', 'Health Insurance', 'Full-Time'),
(25, 'Andrew Lee', '123 Pine St, Anytown USA', '3.5/5', 'Finance', 'Office C', '555-8901', 'andrewlee@example.com', '\$80,000', '401k', 'Full-Time'),
(26, 'Hannah Davis', '579 Maple St, Anytown USA', '4.5/5', 'Customer Service', 'Office D', '555-2345', 'hannahdavis@example.com', '\$45,000', NULL, 'Part-Time'),
(27, 'Erica Green', '931 Elm St, Anytown USA', '3/5', 'IT', 'Office A', '555-6789', 'ericagreen@example.com', '\$75,000', 'Vacation Time', 'Full-Time'),
(28, 'Maxwell Smith', '468 Walnut St, Anytown USA', '4/5', 'Sales', 'Office B', '555-0123', 'maxwellsmith@example.com', '\$58,000', NULL, 'Full-Time'),
(29, 'Olivia Brown', '802 Oak St, Anytown USA', '5/5', 'Marketing', 'Office C', '555-4567', 'oliviabrown@example.com', '\$65,000', 'Health Insurance', 'Full-Time'),
(30, 'Lucas Johnson', '579 Pine St, Anytown USA', '4.5/5', 'Finance', 'Office A', '555-8901', 'lucasjohnson@example.com', '\$85,000', '401k', 'Full-Time');

SET IDENTITY_INSERT Workers OFF

```

SET IDENTITY_INSERT Workers ON
INSERT Workers(WorkersID, Name, Address, Reviews, Department, Office, WorkersPhone, Email, WorkersSalary, Benefits, ContractType)
VALUES
(1, 'John Doe', '123 Main St, Anytown USA', '4.5/5', 'Sales', 'Office A', '555-1234', 'johndoe@example.com', '$50,000', 'Health Insurance', 'Full-Time'),
(2, 'Jane Smith', '456 Oak St, Anytown USA', '4/5', 'Marketing', 'Office B', '555-5678', 'janesmith@example.com', '$60,000', NULL, 'Full-Time'),
(3, 'Bob Johnson', '789 Maple St, Anytown USA', '3.5/5', 'IT', 'Office C', '555-9012', 'bobjohnson@example.com', '$80,000', '401k', 'Full-Time'),
(4, 'Sara Lee', '246 Elm St, Anytown USA', '4.5/5', 'Customer Service', 'Office D', '555-3456', 'saralee@example.com', '$45,000', 'Vacation Time', 'Part-Time'),
(5, 'Tom Smith', '135 Pine St, Anytown USA', '4/5', 'Finance', 'Office A', '555-7890', 'tomsmith@example.com', '$75,000', NULL, 'Full-Time'),
(6, 'Mary Johnson', '579 Cedar St, Anytown USA', '3/5', 'Human Resources', 'Office C', '555-2345', 'maryjohnson@example.com', '$70,000', 'Health Insurance', 'Full-Time'),
(7, 'Bill Davis', '802 Walnut St, Anytown USA', '4/5', 'Sales', 'Office B', '555-6789', 'billdavis@example.com', '$55,000', '401k', 'Full-Time'),
(8, 'Emily Brown', '931 Cherry St, Anytown USA', '5/5', 'Marketing', 'Office D', '555-0123', 'emilybrown@example.com', '$65,000', 'Vacation Time', 'Part-Time'),
(9, 'Chris Green', '246 Oak St, Anytown USA', '3.5/5', 'IT', 'Office A', '555-4567', 'chrисgreen@example.com', '$90,000', NULL, 'Full-Time'),
(10, 'Lisa Johnson', '468 Pine St, Anytown USA', '4/5', 'Finance', 'Office C', '555-8901', 'lisajohnson@example.com', '$80,000', 'Health Insurance', 'Full-Time')
SET IDENTITY_INSERT Workers OFF
  
```

(10 rows affected)

Completion time: 2023-04-30T19:01:19.4372156-04:00

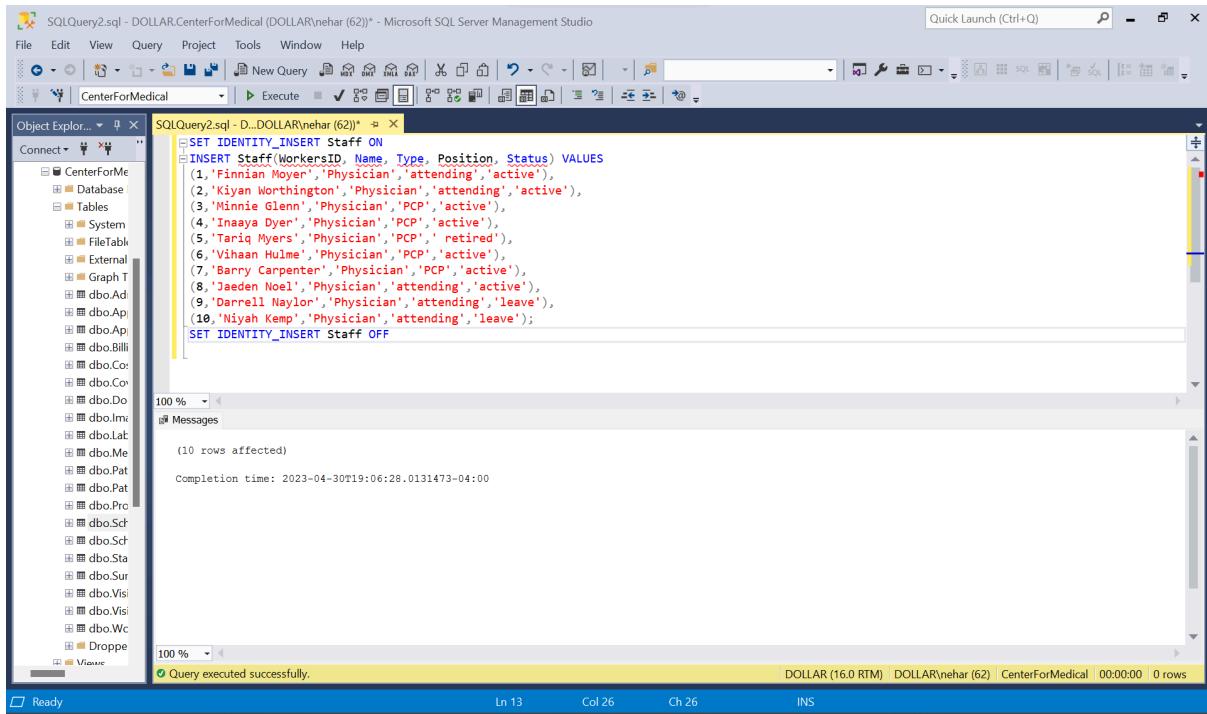
Query executed successfully.

Table Staff

```

SET IDENTITY_INSERT Staff ON
INSERT Staff(WorkersID, Name, Type, Position, Status) VALUES
(1,'Finnian Moyer','Physician','attending','active'),
(2,'Kiyan Worthington','Physician','attending','active'),
(3,'Minnie Glenn','Physician','PCP','active'),
(4,'Inaaya Dyer','Physician','PCP','active'),
(5,'Tariq Myers','Physician','PCP','retired'),
(6,'Vihaan Hulme','Physician','PCP','active'),
(7,'Barry Carpenter','Physician','PCP','active'),
(8,'Jaeden Noel','Physician','attending','active'),
(9,'Darrell Naylor','Physician','attending','leave'),
(10,'Niyah Kemp','Physician','attending','leave'),
(21,'Finnian Moyer','Nurse','attending','active'),
(22,'Kiyan Worthington','Nurse','attending','active'),
(23,'Minnie Glenn','Nurse','PCP','active'),
(24,'Inaaya Dyer','Nurse','PCP','active'),
(25,'Tariq Myers','Nurse','PCP','retired'),
(26,'Vihaan Hulme','Nurse','PCP','active'),
(27,'Barry Carpenter','Nurse','PCP','active'),
(28,'Jaeden Noel','Nurse','attending','active'),
(29,'Darrell Naylor','Nurse','attending','leave'),
(30,'Niyah Kemp','Nurse','attending','leave');
  
```

```
SET IDENTITY_INSERT Staff OFF
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio". The main area contains a query window with the following SQL code:

```

SET IDENTITY_INSERT Staff ON
INSERT Staff(WorkersID, Name, Type, Position, Status) VALUES
(1,'Finnian Moyer','Physician','attending','active'),
(2,'Kiyan Worthington','Physician','attending','active'),
(3,'Minnie Glenn','Physician','PCP','active'),
(4,'Inaaya Dyer','Physician','PCP','active'),
(5,'Tariq Myers','Physician','PCP','retired'),
(6,'Vihaan Huime','Physician','PCP','active'),
(7,'Barry Carpenter','Physician','PCP','active'),
(8,'Jaeden Noel','Physician','attending','active'),
(9,'Darrell Naylor','Physician','attending','leave'),
(10,'Niyah Kemp','Physician','attending','leave');
SET IDENTITY_INSERT Staff OFF

```

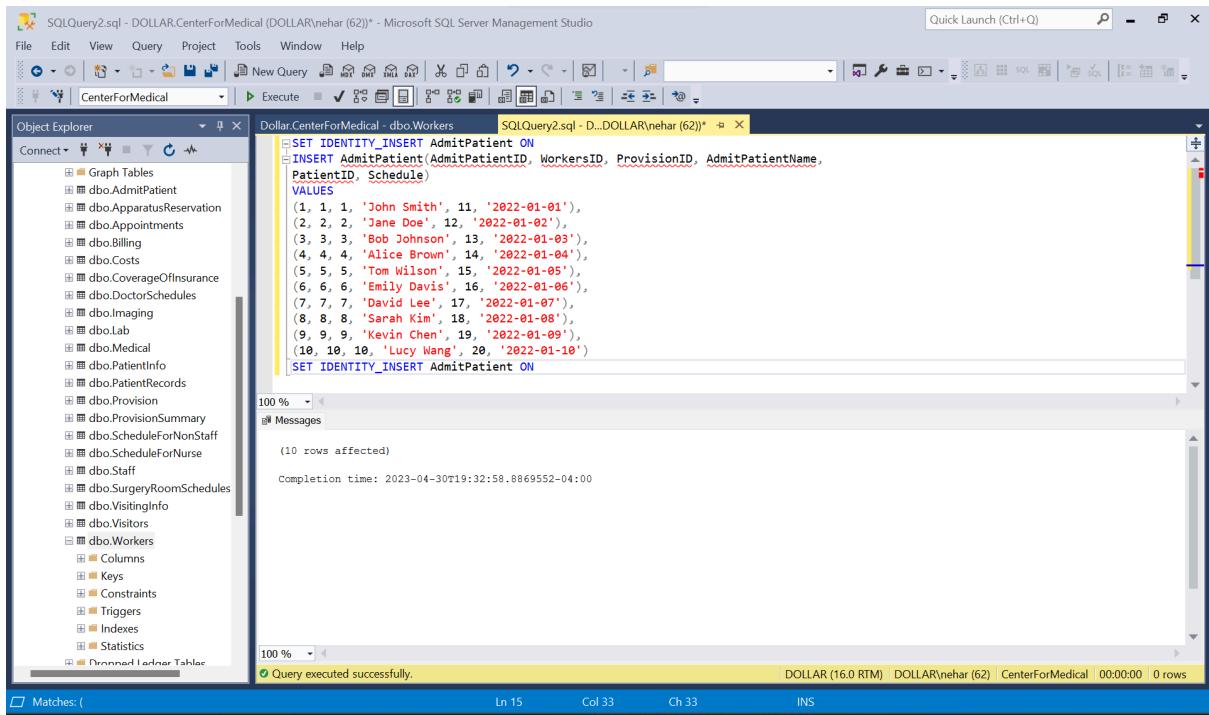
The results pane shows the message "(10 rows affected)" and the completion time "Completion time: 2023-04-30T19:06:28.0131473-04:00". The status bar at the bottom indicates "DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows".

TABLE AdmitPatient:

```

SET IDENTITY_INSERT AdmitPatient ON
INSERT AdmitPatient(AdmitPatientID, WorkersID, ProvisionID, AdmitPatientName,
PatientID, Schedule)
VALUES
(1, 1, 1, 'John Smith', 11, '2022-01-01'),
(2, 2, 2, 'Jane Doe', 12, '2022-01-02'),
(3, 3, 3, 'Bob Johnson', 13, '2022-01-03'),
(4, 4, 4, 'Alice Brown', 14, '2022-01-04'),
(5, 5, 5, 'Tom Wilson', 15, '2022-01-05'),
(6, 6, 6, 'Emily Davis', 16, '2022-01-06'),
(7, 7, 7, 'David Lee', 17, '2022-01-07'),
(8, 8, 8, 'Sarah Kim', 18, '2022-01-08'),
(9, 9, 9, 'Kevin Chen', 19, '2022-01-09'),
(10, 10, 10, 'Lucy Wang', 20, '2022-01-10')
SET IDENTITY_INSERT AdmitPatient ON

```



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists database objects for the 'CenterForMedical' database, including tables like Graph Tables, dbo.AdmitPatient, dbo.ApparatusReservation, etc. The main window displays a query editor with the following T-SQL code:

```

SET IDENTITY_INSERT AdmitPatient ON
INSERT AdmitPatient(AdmitPatientID, WorkersID, ProvisionID, AdmitPatientName,
PatientID, Schedule)
VALUES
(1, 1, 1, 'John Smith', 11, '2022-01-01'),
(2, 2, 2, 'Jane Doe', 12, '2022-01-02'),
(3, 3, 3, 'Bob Johnson', 13, '2022-01-03'),
(4, 4, 4, 'Alice Brown', 14, '2022-01-04'),
(5, 5, 5, 'Tom Wilson', 15, '2022-01-05'),
(6, 6, 6, 'Emily Davis', 16, '2022-01-06'),
(7, 7, 7, 'David Lee', 17, '2022-01-07'),
(8, 8, 8, 'Sarah Kim', 18, '2022-01-08'),
(9, 9, 9, 'Kevin Chen', 19, '2022-01-09'),
(10, 10, 10, 'Lucy Wang', 20, '2022-01-10')
SET IDENTITY_INSERT AdmitPatient ON

```

The status bar at the bottom indicates 'Query executed successfully.' and shows statistics: Ln 15, Col 33, Ch 33, INS.

TABLE: CoverageOfInsurance

```

SET IDENTITY_INSERT CoverageOfInsurance ON
INSERT CoverageOfInsurance(InsuranceID, PatientID, InsuranceCompanyName,
InsurancePhone, InsuranceCov
VALUES
(1, 1, 'Blue Cross Blue Shield', '555-555-1212', 'Full Coverage'),
(2, 2, 'Aetna', '555-555-1313', 'Half Coverage'),
(3, 3, 'UnitedHealthcare', '555-555-1414', 'Full Coverage'),
(4, 4, 'Humana', '555-555-1515', 'Half Coverage'),
(5, 5, 'Cigna', '555-555-1616', 'Full Coverage'),
(6, 6, 'Anthem', '555-555-1717', 'Full Coverage'),
(7, 7, 'Aetna', '555-555-1818', 'Half Coverage'),
(8, 8, 'UnitedHealthcare', '555-555-1919', 'Full Coverage'),
(9, 9, 'Humana', '555-555-2020', 'Half Coverage'),
(10, 10, 'Blue Cross Blue Shield', '555-555-2121', 'Full Coverage')

```

SET IDENTITY_INSERT CoverageOfInsurance OFF

```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
CenterForMedical Execute
Object Explorer
Dollar.CenterForMedical - dbo.Workers SQLQuery2.sql - D...DOLLAR\nehar (62)*
SET IDENTITY_INSERT CoverageOfInsurance ON
INSERT CoverageOfInsurance(InsuranceID, PatientID, InsuranceCompanyName, InsurancePhone, InsuranceCoverages)
VALUES
(1, 1, 'Blue Cross Blue Shield', '555-555-1212', 'Full Coverage'),
(2, 2, 'Aetna', '555-555-1313', 'Half Coverage'),
(3, 3, 'UnitedHealthcare', '555-555-1414', 'Full Coverage'),
(4, 4, 'Humana', '555-555-1515', 'Half Coverage'),
(5, 5, 'Cigna', '555-555-1616', 'Full Coverage'),
(6, 6, 'Anthem', '555-555-1717', 'Full Coverage'),
(7, 7, 'Aetna', '555-555-1818', 'Half Coverage'),
(8, 8, 'UnitedHealthcare', '555-555-1919', 'Full Coverage'),
(9, 9, 'Humana', '555-555-2020', 'Half Coverage'),
(10, 10, 'Blue Cross Blue Shield', '555-555-2121', 'Full Coverage')
SET IDENTITY_INSERT CoverageOfInsurance OFF

100 %
Messages
(10 rows affected)

Completion time: 2023-04-30 19:42:44.5362437-04:00

100 %
Query executed successfully.
DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows
Ln 14 Col 44 Ch 44 INS
Ready

```

ScheduleForNonStaff:

```

SET IDENTITY_INSERT ScheduleForNonStaff ON
INSERT ScheduleForNonStaff(NonStaffID, WorkersID, ProvisionID, Schedule)VAIUES
(1, 11, 5, '2021-05-01'),
(2, 12, 4, '2021-05-02'),
(3, 13, 3, '2021-05-03'),
(4, 14, 2, '2021-05-04'),
(5, 15, 1, '2021-05-05'),
(6, 16, 6, '2021-05-06'),
(7, 17, 7, '2021-05-07'),
(8, 18, 8, '2021-05-08'),
(9, 19, 9, '2021-05-09'),
(10, 20, 10, '2021-05-10')
SET IDENTITY_INSERT ScheduleForNonStaff OFF

```

```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
File New Query Object Explorer CenterForMedical Execute SQL Server Object Browser Task List
SQLQuery2.sql - D...DOLLAR\nehar (62)*
SET IDENTITY_INSERT ScheduleForNonStaff ON
INSERT ScheduleForNonStaff(NonStaffID, WorkersID, ProvisionID, Schedule)VALUES
(1, 11, 5, '2021-05-01'),
(2, 12, 4, '2021-05-02'),
(3, 13, 3, '2021-05-03'),
(4, 14, 2, '2021-05-04'),
(5, 15, 1, '2021-05-05'),
(6, 16, 6, '2021-05-06'),
(7, 17, 7, '2021-05-07'),
(8, 18, 8, '2021-05-08'),
(9, 19, 9, '2021-05-09'),
(10, 20, 10, '2021-05-10')
SET IDENTITY_INSERT ScheduleForNonStaff OFF

```

Messages
(10 rows affected)
Completion time: 2023-04-30 19:55:18.7349817-04:00

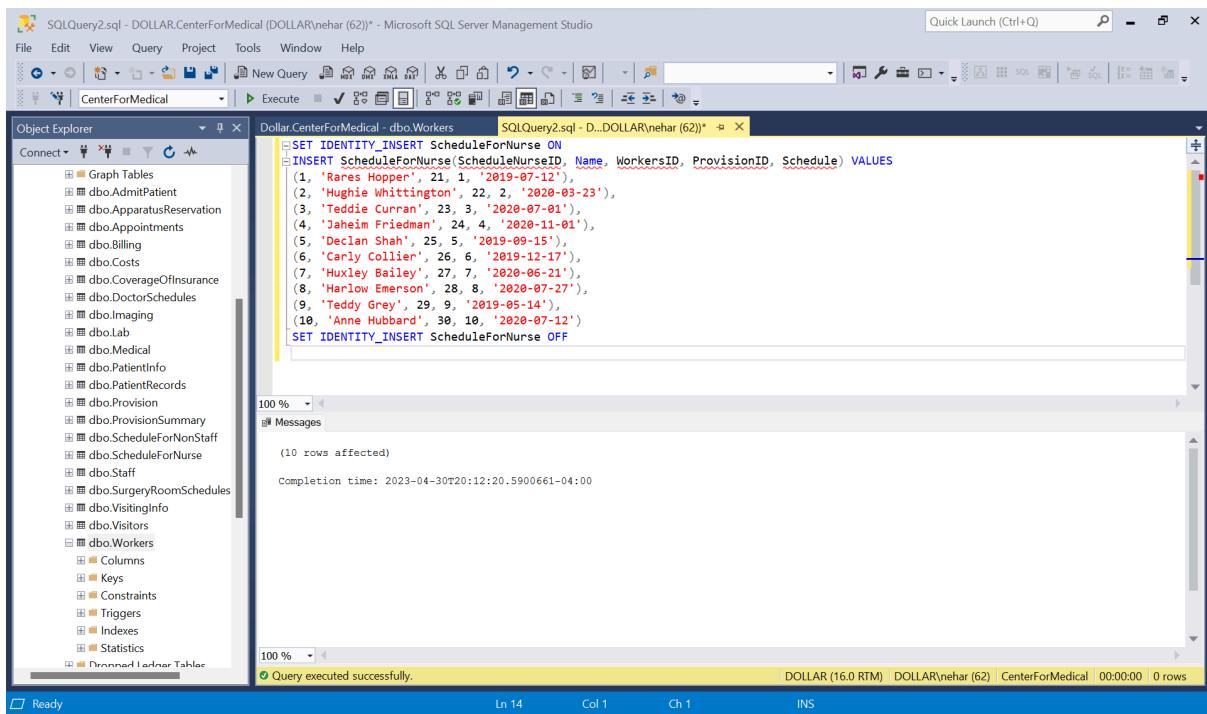
Query executed successfully.

Table :ScheduleForNurse

```

SET IDENTITY_INSERT ScheduleForNurse ON
INSERT ScheduleForNurse(ScheduleNurseID, Name, WorkersID, ProvisionID, Schedule)
VALUES
(1, 'Rares Hopper', 21, 1, '2019-07-12'),
(2, 'Hughie Whittington', 22, 2, '2020-03-23'),
(3, 'Teddie Curran', 23, 3, '2020-07-01'),
(4, 'Jaheim Friedman', 24, 4, '2020-11-01'),
(5, 'Declan Shah', 25, 5, '2019-09-15'),
(6, 'Carly Collier', 26, 6, '2019-12-17'),
(7, 'Huxley Bailey', 27, 7, '2020-06-21'),
(8, 'Harlow Emerson', 28, 8, '2020-07-27'),
(9, 'Teddy Grey', 29, 9, '2019-05-14'),
(10, 'Anne Hubbard', 30, 10, '2020-07-12')
SET IDENTITY_INSERT ScheduleForNurse OFF

```



```

SET IDENTITY_INSERT ScheduleForNurse ON
INSERT ScheduleForNurse(ScheduleNurseID, Name, WorkersID, ProvisionID, Schedule) VALUES
(1, 'Rares Hopper', 21, 1, '2019-07-12'),
(2, 'Hughie Whittington', 22, 2, '2020-03-23'),
(3, 'Teddie Curran', 23, 3, '2020-07-01'),
(4, 'Jaheim Friedman', 24, 4, '2020-11-01'),
(5, 'Declan Shah', 25, 5, '2019-09-15'),
(6, 'Carly Collier', 26, 6, '2019-12-17'),
(7, 'Huxley Bailey', 27, 7, '2020-06-21'),
(8, 'Harlow Emerson', 28, 8, '2020-07-27'),
(9, 'Teddy Grey', 29, 9, '2019-05-14'),
(10, 'Anne Hubbard', 30, 10, '2020-07-12')
SET IDENTITY_INSERT ScheduleForNurse OFF
  
```

Messages
(10 rows affected)
Completion time: 2023-04-30T20:12:20.5900661-04:00

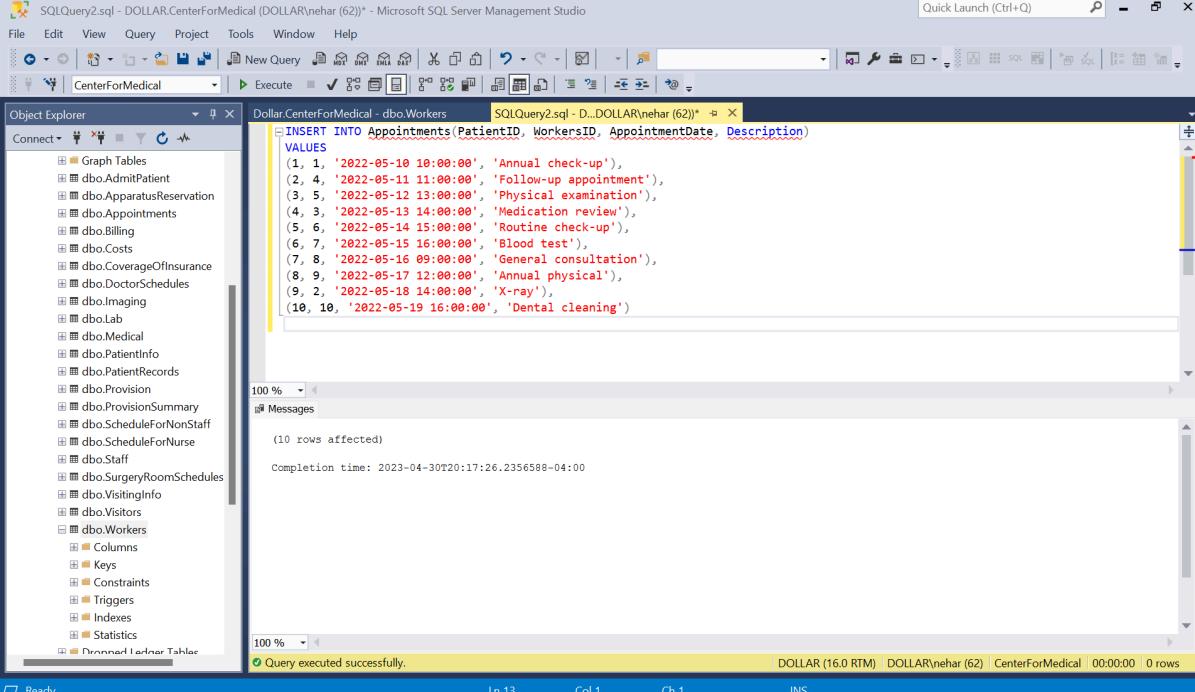
Query executed successfully.

Table Appointments:

INSERT INTO Appointments(PatientID, WorkersID, AppointmentDate, Description)
VALUES

```

(1, 1, '2022-05-10 10:00:00', 'Annual check-up'),
(2, 4, '2022-05-11 11:00:00', 'Follow-up appointment'),
(3, 5, '2022-05-12 13:00:00', 'Physical examination'),
(4, 3, '2022-05-13 14:00:00', 'Medication review'),
(5, 6, '2022-05-14 15:00:00', 'Routine check-up'),
(6, 7, '2022-05-15 16:00:00', 'Blood test'),
(7, 8, '2022-05-16 09:00:00', 'General consultation'),
(8, 9, '2022-05-17 12:00:00', 'Annual physical'),
(9, 2, '2022-05-18 14:00:00', 'X-ray'),
(10, 10, '2022-05-19 16:00:00', 'Dental cleaning')
  
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Graph Tables, dbo.AdmitPatient, dbo.ApparatusReservation, etc. The main query window contains the following SQL code:

```
INSERT INTO Appointments(PatientID, WorkersID, AppointmentDate, Description)
VALUES
(1, 1, '2022-05-10 10:00:00', 'Annual check-up'),
(2, 4, '2022-05-11 11:00:00', 'Follow-up appointment'),
(3, 5, '2022-05-12 13:00:00', 'Physical examination'),
(4, 3, '2022-05-13 14:00:00', 'Medication review'),
(5, 6, '2022-05-14 15:00:00', 'Routine check-up'),
(6, 7, '2022-05-15 16:00:00', 'Blood test'),
(7, 8, '2022-05-16 09:00:00', 'General consultation'),
(8, 9, '2022-05-17 12:00:00', 'Annual physical'),
(9, 2, '2022-05-18 14:00:00', 'X-ray'),
(10, 10, '2022-05-19 16:00:00', 'Dental cleaning')
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2023-04-30 20:17:26.2356588-04:00".

TABLES PatientsRecord

```
SET IDENTITY_INSERT PatientRecords ON
INSERT PatientRecords(PatientRecordID, PatientID, PatientWeight, PatientHeight, Vitals,
Doctor, CheckInTime, CheckOutTime, Discharge, ProcedureCodes,
SuggestedDoctor)VALUES
(1, 1, 80, 180, '120/80', 'Dr. Johnson', '2022-02-01 08:00:00', '2022-02-01 13:00:00', 'Yes',
1001, 'Dr. Lee'),
(2, 2, 65, 165, '130/85', 'Dr. Kim', '2022-02-05 09:30:00', '2022-02-05 14:00:00', 'Yes', 1002,
'Dr. Park'),
(3, 3, 90, 175, '140/90', 'Dr. Brown', '2022-02-07 12:00:00', '2022-02-07 16:30:00', 'No',
NULL, 'Dr. Smith'),
(4, 4, 75, 170, '120/80', 'Dr. Davis', '2022-02-10 10:00:00', '2022-02-10 12:00:00', 'Yes', 1003,
'Dr. Johnson'),
(5, 5, 85, 180, '135/90', 'Dr. Lee', '2022-02-12 08:30:00', '2022-02-12 14:00:00', 'Yes', 1004,
'Dr. Kim'),
(6, 6, 70, 160, '130/85', 'Dr. Park', '2022-02-15 11:00:00', '2022-02-15 15:30:00', 'No', NULL,
'Dr. Brown'),
(7, 7, 100, 190, '140/90', 'Dr. Smith', '2022-02-20 09:00:00', '2022-02-20 12:30:00', 'Yes',
1005, 'Dr. Davis'),
(8, 8, 55, 150, '120/80', 'Dr. Johnson', '2022-02-25 13:00:00', '2022-02-25 16:00:00', 'Yes',
1006, 'Dr. Lee'),
(9, 9, 75, 170, '130/85', 'Dr. Kim', '2022-02-28 10:30:00', '2022-02-28 13:00:00', 'No', NULL,
'Dr. Park'),
(10, 10, 95, 180, '140/90', 'Dr. Brown', '2022-03-02 08:00:00', '2022-03-02 14:30:00', 'Yes',
1007, 'Dr. Smith');
```

SET IDENTITY_INSERT PatientRecords OFF

```

SET IDENTITY_INSERT PatientRecords ON
INSERT PatientRecords(PatientRecordID, PatientID, PatientWeight, PatientHeight, Vitals, Doctor, CheckInTime, CheckOutTime, DischargeStatus)
VALUES
(1, 1, 80, '120/80', 'Dr. Johnson', '2022-02-01 08:00:00', '2022-02-01 13:00:00', 'Yes', 1001, 'Dr. Lee'),
(2, 2, 65, 165, '130/85', 'Dr. Kim', '2022-02-05 09:30:00', '2022-02-05 14:00:00', 'Yes', 1002, 'Dr. Park'),
(3, 3, 90, 175, '140/90', 'Dr. Brown', '2022-02-07 12:00:00', '2022-02-07 16:30:00', 'No', NULL, 'Dr. Smith'),
(4, 4, 75, 170, '120/80', 'Dr. Davis', '2022-02-10 10:00:00', '2022-02-10 12:00:00', 'Yes', 1003, 'Dr. Johnson'),
(5, 5, 85, 180, '135/90', 'Dr. Lee', '2022-02-12 08:30:00', '2022-02-12 14:00:00', 'Yes', 1004, 'Dr. Kim'),
(6, 6, 70, 160, '130/85', 'Dr. Park', '2022-02-15 11:00:00', '2022-02-15 15:30:00', 'No', NULL, 'Dr. Brown'),
(7, 7, 100, 190, '140/90', 'Dr. Smith', '2022-02-20 09:00:00', '2022-02-20 12:30:00', 'Yes', 1005, 'Dr. Davis'),
(8, 8, 55, 150, '120/80', 'Dr. Johnson', '2022-02-25 13:00:00', '2022-02-25 16:00:00', 'Yes', 1006, 'Dr. Lee'),
(9, 9, 75, 170, '130/85', 'Dr. Kim', '2022-02-28 10:30:00', '2022-02-28 13:00:00', 'No', NULL, 'Dr. Park'),
(10, 10, 95, 180, '140/90', 'Dr. Brown', '2022-03-02 08:00:00', '2022-03-02 14:30:00', 'Yes', 1007, 'Dr. Smith');

SET IDENTITY_INSERT PatientRecords OFF

```

10 rows affected

Completion time: 2023-04-30 20:31:39.3356892-04:00

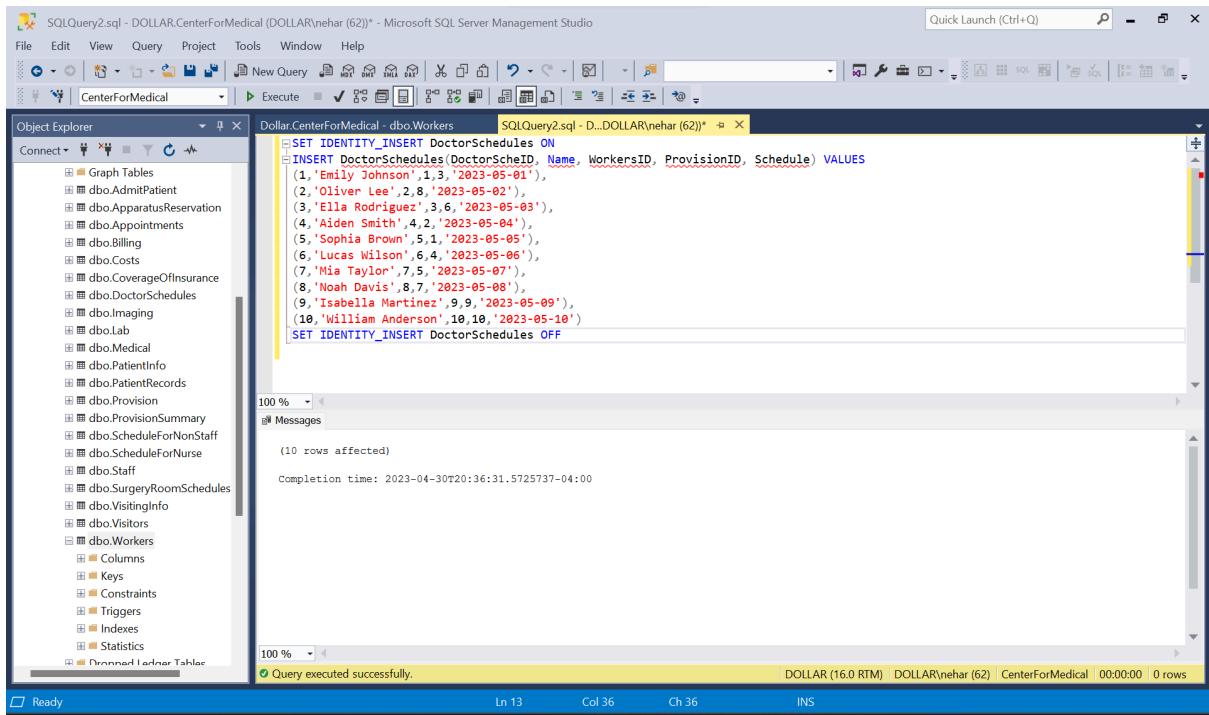
Query executed successfully.

Table Doctor Schedule:

```

SET IDENTITY_INSERT DoctorSchedules ON
INSERT DoctorSchedules(DoctorScheduleID, Name, WorkersID, ProvisionID, Schedule)
VALUES
(1,'Emily Johnson',1,3,'2023-05-01'),
(2,'Oliver Lee',2,8,'2023-05-02'),
(3,'Ella Rodriguez',3,6,'2023-05-03'),
(4,'Aiden Smith',4,2,'2023-05-04'),
(5,'Sophia Brown',5,1,'2023-05-05'),
(6,'Lucas Wilson',6,4,'2023-05-06'),
(7,'Mia Taylor',7,5,'2023-05-07'),
(8,'Noah Davis',8,7,'2023-05-08'),
(9,'Isabella Martinez',9,9,'2023-05-09'),
(10,'William Anderson',10,10,'2023-05-10')
SET IDENTITY_INSERT DoctorSchedules OFF

```



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Graph Tables, dbo.AdmitPatient, dbo.ApparatusReservation, etc. The main window displays a T-SQL script:

```

SET IDENTITY_INSERT DoctorSchedules ON
INSERT DoctorSchedules(DoctorScheduleID, Name, WorkersID, ProvisionID, Schedule) VALUES
(1, 'Emily Johnson', 1, 3, '2023-05-01'),
(2, 'Oliver Lee', 2, 8, '2023-05-02'),
(3, 'Ella Rodriguez', 3, 6, '2023-05-03'),
(4, 'Aiden Smith', 4, 2, '2023-05-04'),
(5, 'Sophia Brown', 5, 1, '2023-05-05'),
(6, 'Lucas Wilson', 6, 4, '2023-05-06'),
(7, 'Mia Taylor', 7, 5, '2023-05-07'),
(8, 'Noah Davis', 8, 7, '2023-05-08'),
(9, 'Isabella Martinez', 9, 9, '2023-05-09'),
(10, 'William Anderson', 10, 10, '2023-05-10')
SET IDENTITY_INSERT DoctorSchedules OFF

```

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

Table: SurgeryRoomschedule:

```

SET IDENTITY_INSERT SurgeryRoomSchedules ON
INSERT SurgeryRoomSchedules(SurgeryRoomScheduleID, WorkersID, ProvisionID,
SurgeryRoomName, Schedule)
VALUES
(1, 6, 1, 'firstclass1', '2021-05-04'),
(2, 1, 2, 'firstclass2', '2021-05-06'),
(3, 3, 3, 'secondclass1', '2021-05-07'),
(4, 5, 4, 'firstclass3', '2021-05-08'),
(5, 4, 5, 'secondclass2', '2021-05-10'),
(6, 9, 6, 'secondclass3', '2021-05-12'),
(7, 8, 7, 'firstclass4', '2021-05-13'),
(8, 7, 8, 'firstclass5', '2021-05-15'),
(9, 2, 9, 'firstclass6', '2021-05-17'),
(10, 10, 10, 'firstclass7', '2021-05-18')
SET IDENTITY_INSERT SurgeryRoomSchedules OFF

```

```

SET IDENTITY_INSERT SurgeryRoomSchedules ON
INSERT SurgeryRoomSchedules(SurRooSchID, WorkersID, ProvisionID, SurgeryRoomName, Schedule)
VALUES
(1, 6, 1, 'firstclass1', '2021-05-04'),
(2, 1, 2, 'firstclass2', '2021-05-06'),
(3, 3, 3, 'secondclass1', '2021-05-07'),
(4, 5, 4, 'firstclass3', '2021-05-08'),
(5, 4, 5, 'secondclass2', '2021-05-10'),
(6, 9, 6, 'secondclass3', '2021-05-12'),
(7, 8, 7, 'firstclass4', '2021-05-13'),
(8, 7, 8, 'firstclass5', '2021-05-15'),
(9, 2, 9, 'firstclass6', '2021-05-17'),
(10, 10, 10, 'firstclass7', '2021-05-18')
SET IDENTITY_INSERT SurgeryRoomSchedules OFF
  
```

(10 rows affected)

Completion time: 2023-04-30T20:40:28.4538295-04:00

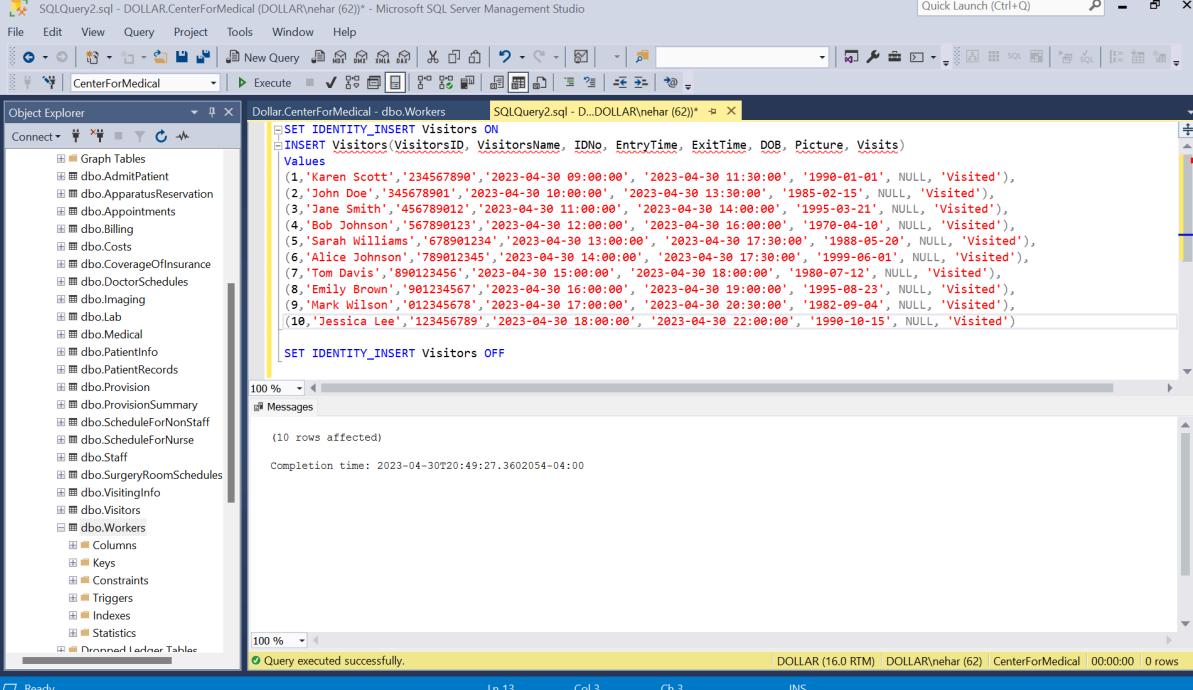
Query executed successfully.

Table : Visitors

```

SET IDENTITY_INSERT Visitors ON
INSERT Visitors(VisitorsID, VisitorsName, IDNo, EntryTime, ExitTime, DOB, Picture, Visits)
Values
(1,'Karen Scott','234567890','2023-04-30 09:00:00', '2023-04-30 11:30:00', '1990-01-01',
NULL, 'Visited'),
(2,'John Doe','345678901','2023-04-30 10:00:00', '2023-04-30 13:30:00', '1985-02-15',
NULL, 'Visited'),
(3,'Jane Smith','456789012','2023-04-30 11:00:00', '2023-04-30 14:00:00', '1995-03-21',
NULL, 'Visited'),
(4,'Bob Johnson','567890123','2023-04-30 12:00:00', '2023-04-30 16:00:00', '1970-04-10',
NULL, 'Visited'),
(5,'Sarah Williams','678901234','2023-04-30 13:00:00', '2023-04-30 17:30:00', '1988-05-20',
NULL, 'Visited'),
(6,'Alice Johnson','789012345','2023-04-30 14:00:00', '2023-04-30 17:30:00', '1999-06-01',
NULL, 'Visited'),
(7,'Tom Davis','890123456','2023-04-30 15:00:00', '2023-04-30 18:00:00', '1980-07-12',
NULL, 'Visited'),
(8,'Emily Brown','901234567','2023-04-30 16:00:00', '2023-04-30 19:00:00', '1995-08-23',
NULL, 'Visited'),
(9,'Mark Wilson','012345678','2023-04-30 17:00:00', '2023-04-30 20:30:00', '1982-09-04',
NULL, 'Visited'),
(10,'Jessica Lee','123456789','2023-04-30 18:00:00', '2023-04-30 22:00:00', '1990-10-15',
NULL, 'Visited')
  
```

SET IDENTITY_INSERT Visitors OFF



```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
CenterForMedical Execute
Object Explorer
Dollar.CenterForMedical - dbo.Visitors
SQLQuery2.sql - D...DOLLAR\nehar (62)*
SET IDENTITY_INSERT Visitors ON
INSERT Visitors(VisitorsID, VisitorsName, IDNo, EntryTime, ExitTime, DOB, Picture, Visits)
Values
(1, 'Karen Scott', '234567890', '2023-04-30 09:00:00', '2023-04-30 11:30:00', '1990-01-01', NULL, 'Visited'),
(2, 'John Doe', '345678901', '2023-04-30 10:00:00', '2023-04-30 13:30:00', '1985-02-15', NULL, 'Visited'),
(3, 'Jane Smith', '456789012', '2023-04-30 11:00:00', '2023-04-30 14:00:00', '1995-03-21', NULL, 'Visited'),
(4, 'Bob Johnson', '567890123', '2023-04-30 12:00:00', '2023-04-30 16:00:00', '1970-04-10', NULL, 'Visited'),
(5, 'Sarah Williams', '678901234', '2023-04-30 13:00:00', '2023-04-30 14:00:00', '1988-05-20', NULL, 'Visited'),
(6, 'Alice Johnson', '789012345', '2023-04-30 14:00:00', '2023-04-30 17:30:00', '1999-06-01', NULL, 'Visited'),
(7, 'Tom Davis', '890123456', '2023-04-30 15:00:00', '2023-04-30 18:00:00', '1980-07-12', NULL, 'Visited'),
(8, 'Emily Brown', '901234567', '2023-04-30 16:00:00', '2023-04-30 19:00:00', '1995-08-23', NULL, 'Visited'),
(9, 'Mark Wilson', '012345678', '2023-04-30 17:00:00', '2023-04-30 20:30:00', '1982-09-04', NULL, 'Visited'),
(10, 'Jessica Lee', '123456789', '2023-04-30 18:00:00', '2023-04-30 22:00:00', '1990-10-15', NULL, 'Visited')
SET IDENTITY_INSERT Visitors OFF

```

10 rows affected

Completion time: 2023-04-30 20:49:27.3602054-04:00

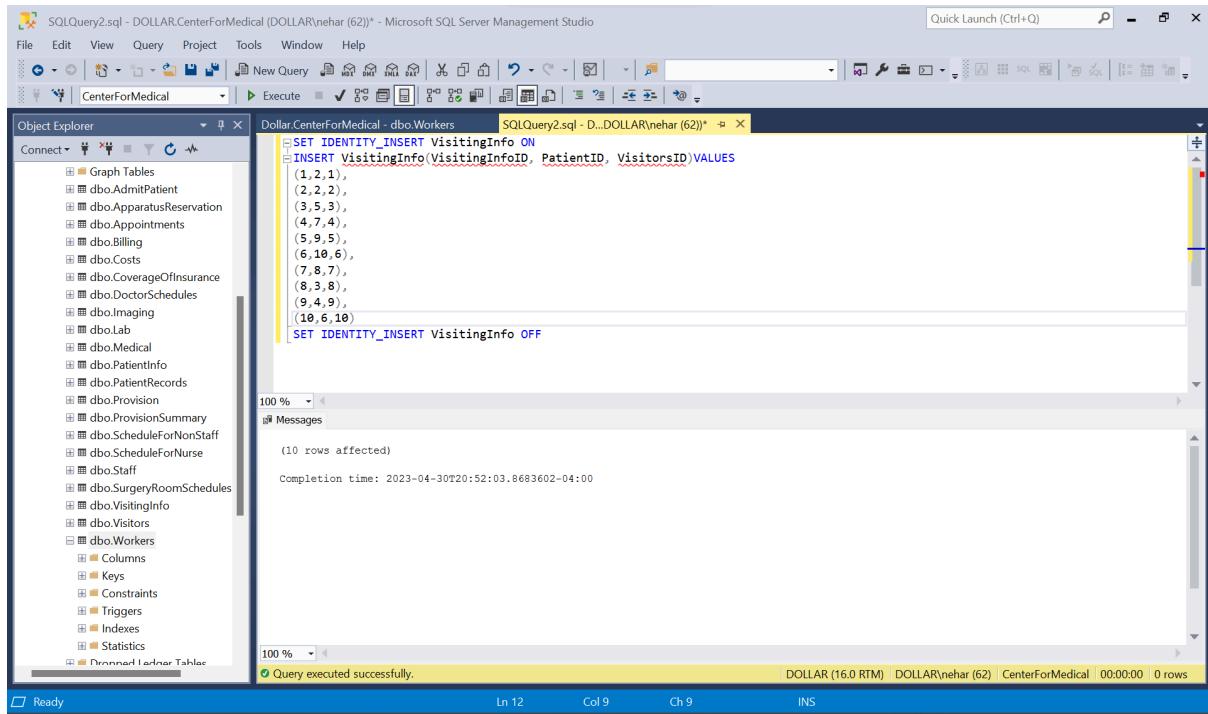
Query executed successfully.

TABLE :VisitingInfo

SET IDENTITY_INSERT VisitingInfo ON
INSERT VisitingInfo(VisitingInfoID, PatientID, VisitorsID)VALUES

(1,2,1),
(2,2,2),
(3,5,3),
(4,7,4),
(5,9,5),
(6,10,6),
(7,8,7),
(8,3,8),
(9,4,9),
(10,6,10)

SET IDENTITY_INSERT VisitingInfo OFF



```

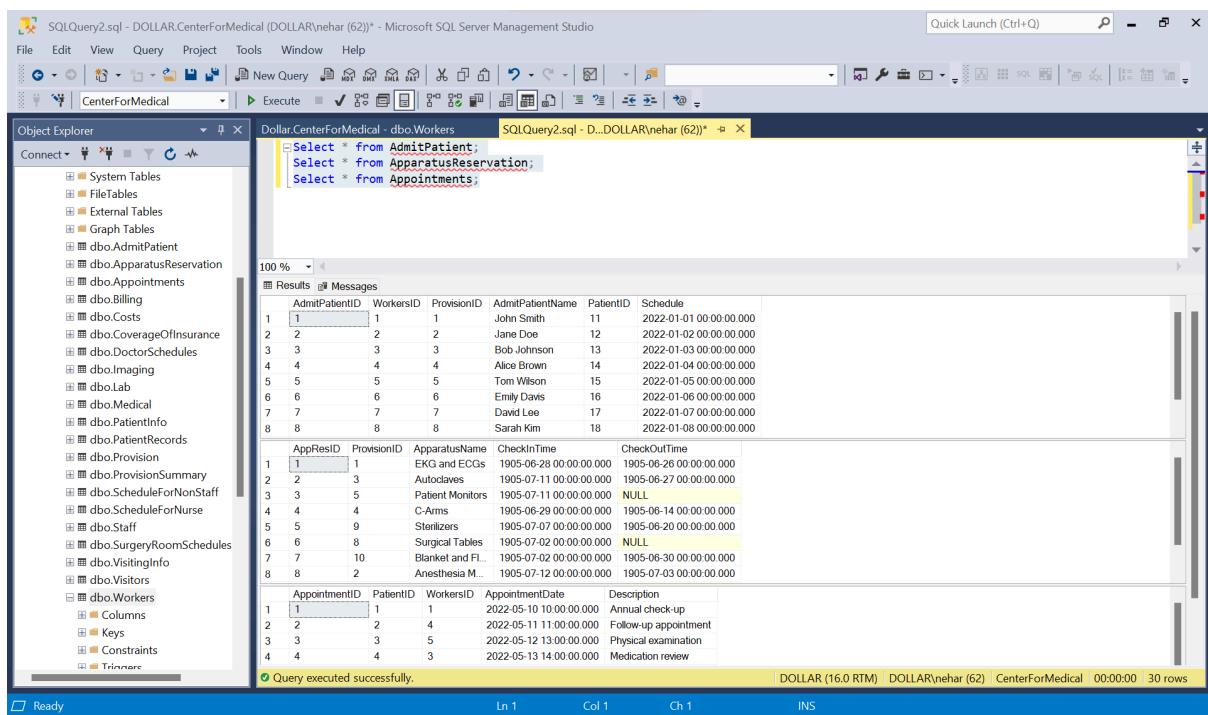
SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
CenterForMedical Execute New Query SQL Server Object Explorer
Object Explorer
Connect > CenterForMedical > CenterForMedical - dbo.Workers SQLQuery2.sql - D...DOLLAR\nehar (62)*
SET IDENTITY_INSERT VisitingInfo ON
INSERT VisitingInfo(VisitingInfoID, PatientID, VisitorsID)VALUES
(1,2,1),
(2,2,2),
(3,5,3),
(4,7,4),
(5,9,5),
(6,10,6),
(7,8,7),
(8,3,8),
(9,4,9),
(10,6,10)
SET IDENTITY_INSERT VisitingInfo OFF

100 % Messages
(10 rows affected)
Completion time: 2023-04-30 20:52:03.8683602-04:00

100 % Results
Query executed successfully.
DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows
Ln 12 Col 9 Ch 9 INS
Ready

```

DISPLAY OF DATABASE



```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
CenterForMedical Execute New Query SQL Server Object Explorer
Object Explorer
Connect > CenterForMedical > CenterForMedical - dbo.Workers SQLQuery2.sql - D...DOLLAR\nehar (62)*
Select * from AdmitPatient;
Select * from ApparatusReservation;
Select * from Appointments;

100 % Messages
Results
AdmitPatientID WorkersID ProvisionID AdmitPatientName PatientID Schedule
1 1 1 John Smith 11 2022-01-01 00:00:00.000
2 2 2 Jane Doe 12 2022-01-02 00:00:00.000
3 3 3 Bob Johnson 13 2022-01-03 00:00:00.000
4 4 4 Alice Brown 14 2022-01-04 00:00:00.000
5 5 5 Tom Wilson 15 2022-01-05 00:00:00.000
6 6 6 Emily Davis 16 2022-01-06 00:00:00.000
7 7 7 David Lee 17 2022-01-07 00:00:00.000
8 8 8 Sarah Kim 18 2022-01-08 00:00:00.000

AppResID ProvisionID ApparatusName CheckInTime CheckOutTime
1 1 EKG and ECGs 1905-06-28 00:00:00.000 1905-06-26 00:00:00.000
2 2 3 Autoclaves 1905-07-11 00:00:00.000 1905-06-27 00:00:00.000
3 3 5 Patient Monitors 1905-07-11 00:00:00.000 NULL
4 4 4 C-Arms 1905-06-29 00:00:00.000 1905-06-14 00:00:00.000
5 5 9 Sterilizers 1905-07-07 00:00:00.000 1905-06-20 00:00:00.000
6 6 8 Surgical Tables 1905-07-02 00:00:00.000 NULL
7 7 10 Blanket and Fl... 1905-07-02 00:00:00.000 1905-06-30 00:00:00.000
8 8 2 Anesthesia M... 1905-07-12 00:00:00.000 1905-07-03 00:00:00.000

AppointmentID PatientID WorkersID AppointmentDate Description
1 1 1 2022-05-10 10:00:00.000 Annual check-up
2 2 2 4 2022-05-11 11:00:00.000 Follow-up appointment
3 3 3 5 2022-05-12 13:00:00.000 Physical examination
4 4 4 3 2022-05-13 14:00:00.000 Medication review

100 % Results
Query executed successfully.
DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 30 rows
Ln 1 Col 1 Ch 1 INS
Ready

```

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

CenterForMedical

Object Explorer

Dollar.CenterForMedical - dbo.Workers

```
Select * from Billing;
Select * from Costs;
Select * from CoverageOfInsurance;
```

Results Messages

BillingID	PatientID	Visits	MedicalBillingCodes	Payer	PaymentMethod
1	1	Regular visits	L78945612	Blue Cross	Check
2	2	Regular visits	P36985214	Cigna	Card
3	3	Regular visits	A258536914	Aetna	Check
4	4	Good visits	L27021613	United Healthcare	Cash
5	5	Good visits	P20513244	Blue Cross	Card
6	6	Good visits	A22638768	Cigna	Card
7	7	Good visits	E22948025	Aetna	Cash
8	8	Good visits	L47581032	United Healthcare	Cash

CostsID	PatientID	Medications	Procedures
1	5	Ibuprofen	Physical therapy
2	6	Aspirin	MRI scan
3	8	Metformin	CT scan
4	9	Pantoprazole	Colonoscopy
5	10	Ritalin	Spirometry
6	1	Lexapro	Electroencephalogram
7	3	Trulicity	Echocardiogram
8	2	Remicade	Endoscopy

InsuranceID	PatientID	InsuranceCompanyName	InsurancePhone	InsuranceCov
1	1	Blue Cross Blue Shield	555-555-1212	Full Coverage
2	2	Aetna	555-555-1313	Half Coverage
3	3	UnitedHealthcare	555-555-1414	Full Coverage
4	4	Humana	555-555-1515	Half Coverage

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 30 rows

Ln 3 Col 34 Ch 34 INS

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

CenterForMedical

Object Explorer

Dollar.CenterForMedical - dbo.Workers

```
Select * from DoctorSchedules;
Select * from Imaging;
Select * from Lab;
```

Results Messages

DoctorScheID	Name	WorkersID	ProvisionID	Schedule
1	Emily Johnson	1	3	2023-05-01 00:00:00.000
2	Oliver Lee	2	8	2023-05-02 00:00:00.000
3	Ella Rodriguez	3	6	2023-05-03 00:00:00.000
4	Aiden Smith	4	2	2023-05-04 00:00:00.000
5	Sophia Brown	5	1	2023-05-05 00:00:00.000
6	Lucas Wilson	6	4	2023-05-06 00:00:00.000
7	Mia Taylor	7	5	2023-05-07 00:00:00.000
8	Noah Davis	8	7	2023-05-08 00:00:00.000

ImagingID	ImagingName	Testing
1	Direct	CT
2	Indirect	PET
3	Direct	CT
4	Indirect	PET
5	Direct	MRI
6	Indirect	CT
7	Direct	MRI
8	Indirect	PET

LabID	LabName	Results	CompulsoryReporting
1	Jones Lab	Positive	NULL
2	Smith Lab	Positive	Reported
3	Lee Lab	Negative	Reported
4	Garcia L...	Positive	Reported

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 30 rows

Ln 3 Col 18 Ch 18 INS

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Dollar.CenterForMedical - dbo.Workers

```
Select * from Medical;
Select * from PatientInfo;
Select * from PatientRecords;
```

Results Messages

MedicalID	MedicanName	PatientMedicines	RentalEquipment
1	Walgreens	Amoxicillin, Aspirin, Atorvastatin	Defibrillators
2	Rite Aid	Ibuprofen, Insulin, Isosorbide	Anesthesia Machines
3	CVS Pharmacy	Naproxen, Nitrofurantoin, Nortriphyline	Patient Monitors
4	Walmart Pharmacy	Losatan, Lovastatin, Lurasidone	EKG/ECG Machines
5	CVS Pharmacy	Methfoma, Methadone, Methocarbamol	Sterilizers
6	Walgreens	Methotrexate, Methylprednisolone, Midazolam	Surgical Tables
7	Rite Aid	Omeprazole, Ondansetron, Oxycodone	Blanket and Fluid Warmers
8	Walmart Pharmacy	Propantheline, Pyridostigmine, Quetiapine	Electrosurgical Units

PatientID	Name	Address	Phone	PatientEmail	HealthInsuranceInfo	Doctor
1	Stephanie Smith	321 Oak Lane	555551212	stephanie.smith@gmail.com	Aetna	John Kim
2	Ryan Brown	456 Elm Street	555553434	ryan.brown@gmail.com	Blue Cross	Emily Nguyen
3	Michelle Rodriguez	789 Maple Road	555555656	michelle.rodriguez@gmail.com	Cigna	John Kim
4	William Chen	234 Cedar Street	5555557678	william.chen@gmail.com	United Healthcare	Emily Nguyen
5	Amanda Wilson	567 Pine Avenue	5555559000	amanda.wilson@gmail.com	Aetna	John Kim
6	Steven Kim	890 Walnut Street	5555552323	steven.kim@gmail.com	Blue Cross	Emily Nguyen
7	Rachel Davis	901 Spruce Street	5555554545	rachel.davis@gmail.com	Cigna	John Kim
8	Ethan Nguyen	345 Cedar Road	5555567676	ethan.nguyen@gmail.com	United Healthcare	Emily Nguyen

PatientRecordID	PatientID	PatientWeight	PatientHeight	Vitals	Doctor	CheckInTime	CheckOutTime	Discharge	ProcedureCodes	SuggestedDoctor
1	1	80	180	120/80	Dr. Johnson	2022-02-01 08:00:00.000	2022-02-01 13:00:00.000	Yes	1001	Dr. Lee
2	2	65	165	130/85	Dr. Kim	2022-02-05 09:30:00.000	2022-02-05 14:00:00.000	Yes	1002	Dr. Park
3	3	90	175	140/90	Dr. Brown	2022-02-07 12:00:00.000	2022-02-07 16:30:00.000	No	NULL	Dr. Smith
4	4	75	170	120/80	Dr. Davis	2022-02-10 10:00:00.000	2022-02-10 12:00:00.000	Yes	1003	Dr. Johnson

Query executed successfully.

Ln 3 Col 29 Ch 29 INS

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Dollar.CenterForMedical - dbo.Workers

```
Select * from Provision;
Select * from ProvisionSummary;
Select * from ScheduleForNonStaff;
```

Results Messages

ProvisionID	ProvisionName	Location	Departments	CareCapacity	MedicalEquipment	OfficeHours	RoomCapacity
1	General Ward	Building A	Internal Medicine	20	ECG Machine	2023-05-01 08:00:00.000	30
2	Pediatric Ward	Building B	Pediatrics	15	Oxygen Tank	2023-05-01 07:00:00.000	20
3	Cardiac Care Unit	Building C	Cardiology	10	Defibrillator	2023-05-01 09:00:00.000	12
4	Surgical Ward	Building A	Surgery	18	Surgical Tools	2023-05-01 10:00:00.000	25
5	Maternity Ward	Building B	Obstetrics	12	Ultrasound Machine	2023-05-01 11:00:00.000	18
6	Intensive Care Unit	Building C	Critical Care	8	Ventilator	2023-05-01 12:00:00.000	10
7	Psychiatric Ward	Building A	Psychiatry	6	Medication Dispenser	2023-05-01 13:00:00.000	8
8	Rehabilitation Unit	Building B	Physical Therapy	15	Treadmill	2023-05-01 14:00:00.000	20

ProvisionSummaryID	ProvisionID	ImagingID	MedicalID	LabID
1	1	4	1	6
2	2	5	3	9
3	3	2	5	7
4	4	8	10	2
5	5	9	4	1
6	6	6	7	5
7	7	3	6	4
8	8	2	2	6

NonStaffID	WorkersID	ProvisionID	Schedule
1	11	5	2021-05-01 00:00:00.000
2	12	4	2021-05-02 00:00:00.000
3	13	3	2021-05-03 00:00:00.000
4	14	2	2021-05-04 00:00:00.000

Query executed successfully.

Ln 3 Col 34 Ch 34 INS

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

CenterForMedical

Object Explorer

Dollar.CenterForMedical - dbo.Workers

```
Select * from ScheduleForNurses;
Select * from Staff;
Select * from SurgeryRoomSchedules;
```

Results Messages

ScheduleNurseID	Name	WorkersID	ProvisionID	Schedule
1	Rares Hopper	21	1	2019-07-12 00:00:00.000
2	Hughie Whittington	22	2	2020-03-23 00:00:00.000
3	Teddie Curran	23	3	2020-07-01 00:00:00.000
4	Jaheim Friedman	24	4	2020-11-01 00:00:00.000
5	Declan Shah	25	5	2019-09-15 00:00:00.000
6	Carly Collier	26	6	2019-12-17 00:00:00.000
7	Huxley Bailey	27	7	2020-06-21 00:00:00.000
8	Harlow Emerson	28	8	2020-07-27 00:00:00.000

WorkersID	Name	Position	Status	Type
1	Finnian Moyer	attending	active	Physician
2	Kyan Worthington	attending	active	Physician
3	Minnie Glenn	PCP	active	Physician
4	Inaya Dyer	PCP	active	Physician
5	Tariq Myers	PCP	retired	Physician
6	Vihnaan Huima	PCP	active	Physician
7	Barry Carpenter	PCP	active	Physician
8	Jaeden Noel	attending	active	Physician

SurRooSchID	WorkersID	ProvisionID	SurgeryRoomName	Schedule
1	6	1	firstclass1	2021-05-04 00:00:00.000
2	1	2	firstclass2	2021-05-06 00:00:00.000
3	3	3	secondclass1	2021-05-07 00:00:00.000
4	5	4	firstclass3	2021-05-08 00:00:00.000

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 40 rows

Ln 3 Col 35 Ch 35 INS

SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

CenterForMedical

Object Explorer

Dollar.CenterForMedical - dbo.VisitingInfo

```
Select * from VisitingInfo;
Select * from Visitors;
Select * from Workers;
```

Results Messages

VisitingInfoID	PatientID	VisitorsID
1	2	1
2	2	2
3	5	3
4	7	4
5	9	5
6	10	6
7	8	7
8	3	8

VisitorsID	VisitorsName	IDNo	EntryTime	ExitTime	DOB	Picture	Visits
1	Karen Scott	234567890	2023-04-30 09:00:00.000	2023-04-30 11:30:00.000	1990-01-01 00:00:00.000	NULL	Visited
2	John Doe	345678901	2023-04-30 10:00:00.000	2023-04-30 13:30:00.000	1985-02-15 00:00:00.000	NULL	Visited
3	Jane Smith	456789012	2023-04-30 11:00:00.000	2023-04-30 14:00:00.000	1995-03-21 00:00:00.000	NULL	Visited
4	Bob Johnson	567890123	2023-04-30 12:00:00.000	2023-04-30 16:00:00.000	1970-04-10 00:00:00.000	NULL	Visited
5	Sarah Williams	678901234	2023-04-30 13:00:00.000	2023-04-30 17:30:00.000	1988-05-20 00:00:00.000	NULL	Visited
6	Alice Johnson	789012345	2023-04-30 14:00:00.000	2023-04-30 17:30:00.000	1999-06-01 00:00:00.000	NULL	Visited
7	Tom Davis	890123456	2023-04-30 15:00:00.000	2023-04-30 18:00:00.000	1980-07-12 00:00:00.000	NULL	Visited
8	Emily Brown	901234567	2023-04-30 16:00:00.000	2023-04-30 19:00:00.000	1995-08-23 00:00:00.000	NULL	Visited

WorkersID	Name	Address	Reviews	Department	Office	WorkersPhone	Email	WorkersSalary	Benefits	ContractType
1	John Doe	123 Main St, Anytown USA	4.5/5	Sales	Office A	555-1234	john.doe@example.com	\$50,000	Health Insurance	Full-Time
2	Jane Smith	456 Oak St, Anytown U...	4/5	Marketing	Office B	555-5678	janesmith@example.com	\$60,000	NULL	Full-Time
3	Bob Johnson	789 Maple St, Anytown U...	3.5/5	IT	Office C	555-9012	bob.johnson@example.com	\$80,000	401k	Full-Time
4	Sara Lee	246 Elm St, Anytown USA	4.5/5	Customer...	Office D	555-3456	sara.lee@example.com	\$45,000	Vacation Time	Part-Time

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 50 rows

Ln 3 Col 22 Ch 22 INS

III. Database Testing

1.VIEWS

VIEW 1

```

CREATE VIEW PatientBilling_V
AS
SELECT dbo.PatientInfo.PatientID, dbo.PatientInfo.Name, MedicalBillingCodes AS
MBC , Payor
FROM PatientInfo JOIN Billing
ON PatientInfo.PatientID = Billing.PatientID
WHERE Name = 'Stephanie Smith';
GO
SELECT *
FROM PatientBilling_V

```

Results

PatientID	Name	MBC	Payor
1	Stephanie Smith	L78945612	Blue Cross

Query executed successfully.

VIEW 2

```

CREATE VIEW PatientAppointment_V
AS
SELECT AppointmentID, dbo.Appointments.PatientID, AppointmentDate,
dbo.Appointments.WorkersID, dbo.Workers.Name
FROM Appointments JOIN Workers
ON Appointments.WorkersID = Workers.WorkersID
WHERE AppointmentDate > '2022-01-01';
GO
SELECT *
FROM PatientAppointment_V
ORDER BY PatientID;

```

Results

AppointmentID	PatientID	AppointmentDate	WorkersID	Name
1	1	2022-05-10 10:00:00.000	1	John Doe
2	2	2022-05-11 11:00:00.000	4	Sara Lee
3	3	2022-05-12 13:00:00.000	5	Tom Smith
4	4	2022-05-13 14:00:00.000	3	Bob Johnson
5	5	2022-05-14 15:00:00.000	6	Mary Johnson
6	6	2022-05-15 16:00:00.000	7	Bill Davis
7	7	2022-05-16 09:00:00.000	8	Emily Brown
8	8	2022-05-17 12:00:00.000	9	Chris Green
9	9	2022-05-18 14:00:00.000	2	Jane Smith
10	10	2022-05-19 16:00:00.000	10	Lisa Johnson

Query executed successfully.

VIEW 3

```

CREATE VIEW PatientMedicalInfo_V
AS
SELECT dbo.PatientInfo.PatientID, Name, PatientWeight, PatientHeight
FROM PatientInfo JOIN PatientRecords
ON dbo.PatientInfo.PatientID = PatientRecords.PatientID
WHERE Name like '[E]%'
GO
SELECT *
FROM PatientMedicalInfo_V;

```

Completion time: 2023-04-30T21:15:11.3823953-04:00

Query completed with errors.

VIEW 4

```

CREATE VIEW PhysicianSchedule_V
AS
SELECT DoctorScheID, Staff.WorkersID, CONCAT(Staff.Name, ', ', Position, ', ', Schedule) AS DoctorInform, Status
FROM Staff JOIN DoctorSchedules
ON Staff.WorkersID = DoctorSchedules.WorkersID;
GO
SELECT *
FROM PhysicianSchedule_V
WHERE DoctorScheID = 7;

```

DoctorScheID	WorkersID	DoctorInform	Status
1	7	Barry Carpenter,PCP,May 7 2023 12:00AM	active

Query executed successfully.

2. Stored procedures

Stored procedures 1

```

IF OBJECT_ID(spProvisionInfo) IS NOT NULL
DROP PROC spProvisionInfo;
GO
CREATE PROC spProvisionInfo

```

```

@ProvisionName VARCHAR(50)
AS
SELECT Location, OfficeHours, RoomCapacity, Departments
FROM Provision
WHERE Provision.ProvisionName = @ProvisionName
GO
EXEC spProvisionInfo 'Maternity Ward';

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62))". The main area displays a T-SQL script:

```

IF OBJECT_ID('spProvisionInfo') IS NOT NULL
DROP PROC spProvisionInfo;
GO
CREATE PROC spProvisionInfo
@ProvisionName VARCHAR(50)
AS
SELECT Location, OfficeHours, RoomCapacity, Departments
FROM Provision
WHERE Provision.ProvisionName = @ProvisionName
GO
EXEC spProvisionInfo 'Maternity Ward';

```

The "Results" tab shows the output of the EXEC statement:

	Location	OfficeHours	RoomCapacity	Departments
1	Building B	2023-05-01 11:00:00.000	18	Obstetrics

At the bottom, a message bar indicates: "Query executed successfully." and "DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 1 rows".

Stored procedures 2

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo;

```

```

GO
CREATE PROC spPatientInsuInfo
@name VARCHAR(50)
AS
SELECT InsuranceCompanyName AS HICN, HealthInsuranceInfo AS IC
FROM PatientInfo JOIN CoverageOfInsurance ON Patient.PatientID =
CoverageOfInsurance .PatientID
WHERE PatientInfo.Name = @name
GO
EXEC spPatientInsuInfo 'Stephanie Smith';

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, etc. The main window displays a query script named 'SQLQuery2.sql' which contains a stored procedure definition. The script is as follows:

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo;
GO
CREATE PROC spPatientInsuInfo
@name VARCHAR(50)
AS
SELECT InsuranceCompanyName AS HICN, HealthInsuranceInfo AS IC
FROM PatientInfo JOIN CoverageOfInsurance ON PatientInfo.PatientID =
CoverageOfInsurance.PatientID
WHERE PatientInfo.Name = @name;
GO
EXEC spPatientInsuInfo 'Stephanie Smith';
  
```

The Results tab shows the output of the EXEC command:

HICN	IC
Blue Cross Blue Shield	Aetna

Below the results, a message indicates "Query executed successfully."

Stored procedures 3

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo; /*Drop procedure if previous procedure exists*/
GO
/*Create a procedure to find a patient's visitors information*/
CREATE PROCEDURE GetPatientsByPhysicianID (@PhysicianID INT)
AS
BEGIN
    SELECT PatientInfo.*
    FROM PatientInfo
    INNER JOIN Staff ON PatientInfo.PatientID = Staff.WorkersID
    WHERE Staff.WorkersID = @PhysicianID;
END

EXEC GetPatientsByPhysicianID @PhysicianID=1;
  
```

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo; /*Drop procedure if previous procedure exists*/
GO
/*Create a procedure to find a patient's visitors information*/
CREATE PROCEDURE GetPatientsByPhysicianID (@PhysicianID INT)
AS
BEGIN
    SELECT PatientInfo.*
    FROM PatientInfo
    INNER JOIN Staff ON PatientInfo.PatientID = Staff.WorkersID
    WHERE Staff.WorkersID = @PhysicianID;
END

EXEC GetPatientsByPhysicianID @PhysicianID=1;

```

100 %

	PatientID	Name	Address	Phone	PatientEmail	HealthInsuranceInfo	Doctor
1	1	Stephanie Smith	321 Oak Lane	5555551212	stephanie.smith@gmail.com	Aetna	John Kim

Query executed successfully.

Stored procedures 4

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo; /*Drop procedure if previous procedure exists*/
GO

```

```

/*Create a procedure to find a patient's visitors information*/
CREATE PROCEDURE GetInfoByPatientID (@PhysicianID INT)
AS
BEGIN
    SELECT PatientRecords.*
    FROM Appointments
    INNER JOIN PatientRecords ON Appointments.PatientID = PatientRecords.PatientID
    WHERE Appointments.PatientID = @PhysicianID;
END

EXEC GetInfoByPatientID @PhysicianID=2;

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'CenterForMedical' database is selected. In the center pane, a script window titled 'SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62))' contains the following T-SQL code:

```

IF OBJECT_ID('spPatientInsuInfo') IS NOT NULL
DROP PROC spPatientInsuInfo; /*Drop procedure if previous procedure exists*/
GO

/*Create a procedure to find a patient's visitors information*/
CREATE PROCEDURE GetInfoByPatientID (@PhysicianID INT)
AS
BEGIN
    SELECT PatientRecords.*
    FROM Appointments
    INNER JOIN PatientRecords ON Appointments.PatientID = PatientRecords.PatientID
    WHERE Appointments.PatientID = @PhysicianID;
END

EXEC GetInfoByPatientID @PhysicianID=2;

```

The 'Results' tab shows the output of the EXEC statement:

PatientRecordID	PatientID	PatientWeight	PatientHeight	Vitals	Doctor	CheckInTime	CheckOutTime	Discharge	ProcedureCodes	SuggestedDoctor
1	2	65	165	130/85	Dr. Kim	2022-02-05 09:30:00.000	2022-02-05 14:00:00.000	Yes	1002	Dr. Park

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

3.USER DEFINED FUNCTIONS:

USER DEFINED FUNCTIONS 1

```

CREATE FUNCTION InsurCov
(@InsuranceID INT)
RETURNS INT
BEGIN
RETURN (SELECT InsuranceID FROM CoverageOfInsurance
WHERE InsuranceID=@InsuranceID)
END;
SELECT PatientID, InsuranceCoverages
FROM CoverageOfInsurance
WHERE InsuranceID=dbo.InsurCov(9);

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'CenterForMedical' is selected. In the center pane, a query window titled 'SQLQuery2.sql - DOLLAR\nehar (62)*' displays the following T-SQL code:

```

CREATE FUNCTION InsurCov
(@InsuranceID INT)
RETURNS INT
BEGIN
RETURN (SELECT InsuranceID FROM CoverageOfInsurance
WHERE InsuranceID=@InsuranceID)
END;
SELECT PatientID, InsuranceCoverages
FROM CoverageOfInsurance
WHERE InsuranceID=dbo.InsurCov(9);

```

The 'Results' tab shows the output of the last query:

PatientID	InsuranceCoverages
1	Half Coverage

At the bottom of the screen, a status bar indicates: 'Query executed successfully.' followed by 'DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 1 rows'.

USER DEFINED FUNCTIONS 2

```

CREATE FUNCTION EmplInfo
(@EmpID INT)
RETURNS INT
BEGIN
RETURN (SELECT WorkersID
FROM Workers
WHERE WorkersID=@EmpID)
END;
SELECT WorkersID, WorkersSalary, Benefits, ContractType
FROM Workers
WHERE WorkersID=dbo.EmplInfo(3);

```

```

CREATE FUNCTION EmpInfo
(
    @EmpID INT
)
RETURNS INT
BEGIN
    RETURN (SELECT WorkersID
    FROM Workers
    WHERE WorkersID=@EmpID)
END;
SELECT WorkersID, WorkersSalary, Benefits, ContractType
FROM Workers
WHERE WorkersID=dbo.EmpInfo(3);

```

	WorkersID	WorkersSalary	Benefits	ContractType
1	3	\$80,000	401k	Full-Time

Query executed successfully.

USER DEFINED FUNCTIONS 3

```

CREATE FUNCTION InsuranceInfo
(@PatientID INT)
RETURNS INT
BEGIN
RETURN (SELECT PatientID
FROM PatientInfo
WHERE PatientID=@PatientID)
END;

```

```

SELECT InsuranceID, PatientID, InsuranceCompanyName, InsuranceCov
FROM CoverageOfInsurance
WHERE PatientID=dbo.InsuranceInfo(4);

```

```

CREATE FUNCTION InsuranceInfo
(
    @PatientID INT
)
RETURNS INT
BEGIN
    RETURN (SELECT PatientID
    FROM PatientInfo
    WHERE PatientID=@PatientID)
END;

SELECT InsuranceID, PatientID, InsuranceCompanyName, InsuranceCov
FROM CoverageOfInsurance
WHERE PatientID=dbo.InsuranceInfo(4);

```

InsuranceID	PatientID	InsuranceCompanyName	InsuranceCov
1	4	Humana	Half Coverage

Query executed successfully.

USER DEFINED FUNCTIONS 4:

```

CREATE FUNCTION BillingInfo
(
    @PatientID INT
)
RETURNS INT
BEGIN
    RETURN (SELECT PatientID
    FROM PatientInfo
    WHERE PatientID=@PatientID)
END;

```

```

SELECT BillingID, PatientID, Payor, PaymentMethod
FROM Billing
WHERE PatientID=dbo.BillingInfo(1);

```

```

CREATE FUNCTION BillingInfo
(
    @PatientID INT
)
RETURNS INT
BEGIN
    RETURN (SELECT PatientID
            FROM PatientInfo
            WHERE PatientID = @PatientID)
END;

SELECT BillingID, PatientID, Payor, PaymentMethod
FROM Billing
WHERE PatientID = dbo.BillingInfo(1);

```

Results

BillingID	PatientID	Payor	PaymentMethod
1	1	Blue Cross	Check

Query executed successfully.

4. TRIGGER

TRIGGER 1:

This trigger will update the Reviews column of the corresponding worker in the Workers table with the value 'Positive' whenever a new record is inserted into the Billing table. The trigger uses an INNER JOIN to match the PatientID column in the Billing table with the WorkersID column in the Workers table.

```

CREATE TRIGGER trg_UpdateReviews
ON Billing
AFTER INSERT
AS
BEGIN
    UPDATE Workers
    SET Reviews = 'Positive'
    FROM Workers
    INNER JOIN inserted ON Workers.WorkersID = inserted.PatientID
END

```

Messages

Commands completed successfully.

Completion time: 2023-04-30 22:25:17.8545739-04:00

Query executed successfully.

**CREATE TRIGGER trg_UpdateReviews
ON Billing**

```

AFTER INSERT
AS
BEGIN
    UPDATE Workers
    SET Reviews = 'Positive'
    FROM Workers
    INNER JOIN inserted ON Workers.WorkersID = inserted.PatientID
END

```

TRIGGER 2:

This trigger is designed to automatically update the Department field in the Workers table when a new record is inserted into the Staff table. Specifically, it sets the Department field to "Staff" for the worker whose WorkersID matches the newly inserted WorkersID.

You can modify this trigger to fit your specific needs, such as updating other fields in the Workers table or performing additional checks before updating the Workers table.

```

CREATE TRIGGER tr_Staff_Workers_Insert
ON Staff
FOR INSERT
AS
BEGIN
    DECLARE @WorkersID INT

    SELECT @WorkersID = WorkersID FROM inserted

    UPDATE Workers
    SET Department = 'Staff'
    WHERE WorkersID = @WorkersID
END

```

```

CREATE TRIGGER tr_Staff_Workers_Insert
ON Staff
FOR INSERT
AS
BEGIN
    DECLARE @WorkersID INT
    SELECT @WorkersID = WorkersID FROM inserted
    UPDATE Workers
    SET Department = 'Staff'
    WHERE WorkersID = @WorkersID
END

```

TRIGGER 3:

This trigger will update the Status column of the Staff table to 'Inactive' for all staff members whose contract type is changed to 'Part-Time' in the Workers table.

```

CREATE TRIGGER update_staff_status
ON Workers
AFTER UPDATE
AS
IF UPDATE(ContractType)
BEGIN
    UPDATE Staff
    SET Status = 'Inactive'
    WHERE WorkersID IN (SELECT WorkersID FROM inserted WHERE ContractType =
    'Part-Time')
END;

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'CenterForMedical' is selected. In the center pane, a script window titled 'SQLQuery2.sql' contains the following T-SQL code:

```

CREATE TRIGGER update_staff_status
ON Workers
AFTER UPDATE
AS
IF UPDATE(ContractType)
BEGIN
    UPDATE Staff
    SET Status = 'Inactive'
    WHERE WorkersID IN (SELECT WorkersID FROM inserted WHERE ContractType = 'Part-Time')
END;

```

The 'Messages' pane at the bottom displays the output of the query execution:

- Commands completed successfully.
- Completion time: 2023-04-30T22:27:27.8807022-04:00

The status bar at the bottom right shows: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows.

TRIGGER 4

This trigger will update the Reviews column of the corresponding worker in the Workers table with the value ‘Negative’ whenever a new record is inserted into the Billing table. The trigger uses an INNER JOIN to match the PatientID column in the Billing table with the WorkersID column in the Workers table.

```

CREATE TRIGGER UpdatingReviews
ON Billing
AFTER INSERT
AS
BEGIN
    UPDATE Workers
    SET Reviews = 'Negative'
    FROM Workers
    INNER JOIN inserted ON Workers.WorkersID = inserted.PatientID
END

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, and several dbo.* tables such as AdmitPatient, ApparatusReservation, Appointments, Billing, Costs, CoverageOfInsurance, DoctorSchedules, Imaging, Lab, Medical, PatientInfo, PatientRecords, Provision, ProvisionSummary, ScheduleForNonStaff, ScheduleForNurse, Staff, SurgeryRoomSchedules, VisitingInfo, and Visitors. The SQL Query Editor window on the right contains the following T-SQL code:

```

CREATE TRIGGER UpdatingReviews
ON Billing
AFTER INSERT
AS
BEGIN
    UPDATE Workers
    SET Reviews = 'Negative'
    FROM Workers
    INNER JOIN inserted ON Workers.WorkersID = inserted.PatientID
END

```

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

5. Transactions:

Transactions 1:

```

BEGIN TRAN
UPDATE PatientInfo
SET Doctor = 'Peter Zhu'
WHERE PatientID = 1
COMMIT TRAN;
GO
SELECT PatientID, Doctor
FROM PatientInfo
WHERE PatientID = 1;

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, Graph Tables, and many dbo.* tables. The main pane displays a T-SQL script:

```

BEGIN TRAN
UPDATE PatientInfo
SET Doctor = 'Peter Zhu'
WHERE PatientID = 1
COMMIT TRAN;
GO
SELECT PatientID, Doctor
FROM PatientInfo
WHERE PatientID = 1;

```

The Results tab shows the output of the SELECT statement:

PatientID	Doctor
1	Peter Zhu

Below the results, a message indicates: "Query executed successfully." The status bar at the bottom shows: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 1 rows.

Transactions 2:

```

BEGIN TRAN
UPDATE PatientInfo
SET Doctor = 'Dr. Michael Johnson'
WHERE PatientID = 1
COMMIT TRAN

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, Graph Tables, and many dbo.* tables. The main pane displays a T-SQL script:

```

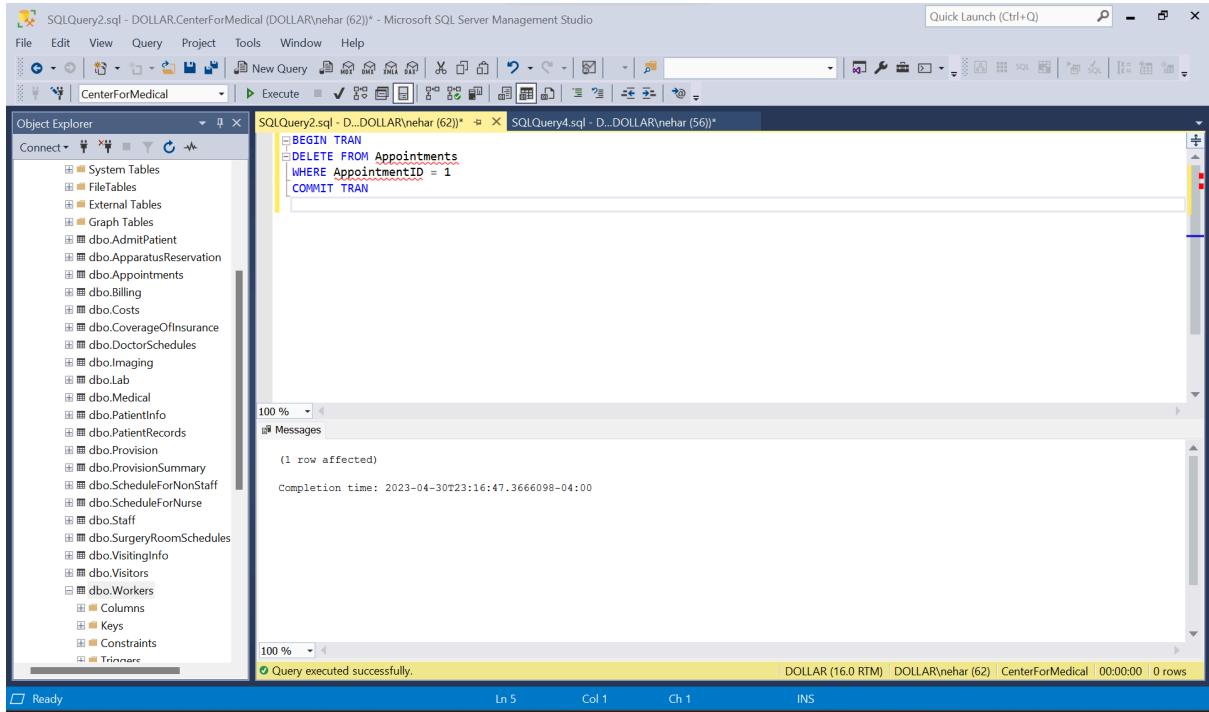
BEGIN TRAN
UPDATE PatientInfo
SET Doctor = 'Dr. Michael Johnson'
WHERE PatientID = 1
COMMIT TRAN;

```

The Results tab shows the output of the UPDATE statement, indicating 1 row affected. The status bar at the bottom shows: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows.

Transactions 3:

```
BEGIN TRAN
DELETE FROM Appointments
WHERE AppointmentID = 1
COMMIT TRAN
```



The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays a tree view of database objects under the 'CenterForMedical' database, including System Tables, External Tables, Graph Tables, and various dbo.* tables like AdmitPatient, ApparatusReservation, Appointments, Billing, Costs, CoverageOfInsurance, DoctorSchedules, Imaging, Lab, Medical, PatientInfo, PatientRecords, Provision, ProvisionSummary, ScheduleForNonStaff, ScheduleForNurse, Staff, SurgeryRoomSchedules, VisitingInfo, Visitors, Workers, and their respective sub-objects like Columns, Keys, Constraints, and Triggers.

The main query window contains the following T-SQL code:

```
BEGIN TRAN
DELETE FROM Appointments
WHERE AppointmentID = 1
COMMIT TRAN
```

The results pane shows the output of the query:

- (1 row affected)
- Completion time: 2023-04-30T23:16:47.3666098-04:00

At the bottom, a status bar indicates "Query executed successfully." and shows the session details: DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 0 rows.

Transactions 4:

```
BEGIN TRAN
ALTER TABLE Appointments
ADD AppointmentTypes varchar(50) NOT NULL DEFAULT 'General'
COMMIT TRAN;
```

```

BEGIN TRAN
ALTER TABLE Appointments
ADD AppointmentTypes varchar(50) NOT NULL DEFAULT 'General'
COMMIT TRAN;
  
```

Messages

Commands completed successfully.

Completion time: 2023-04-30T23:18:39.7067447-04:00

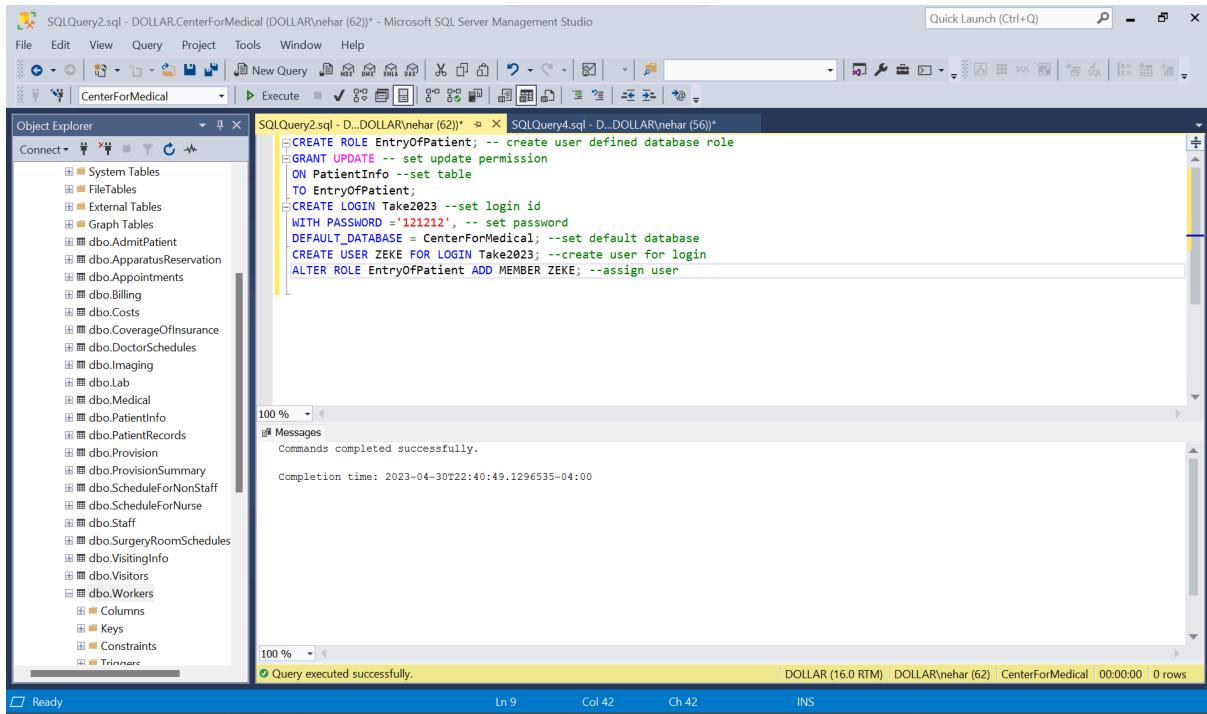
Query executed successfully.

6. Security

Security 1:

```

CREATE ROLE EntryOfPatient; -- create user defined database role
GRANT UPDATE -- set update permission
ON PatientInfo --set table
TO EntryOfPatient;
CREATE LOGIN Take2023 --set login id
WITH PASSWORD = '121212', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER ZEKE FOR LOGIN Take2023; --create user for login
ALTER ROLE EntryOfPatient ADD MEMBER ZEKE; --assign user
  
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery2.sql - DOLLAR.CenterForMedical (DOLLAR\nehar (62)) - Microsoft SQL Server Management Studio". The main area displays a T-SQL script for creating a database:

```

CREATE ROLE EntryOfPatient; -- create user defined database role
GRANT UPDATE -- set update permission
ON PatientInfo --set table
TO EntryOfPatient;
CREATE LOGIN Take2023 --set login id
WITH PASSWORD = '121212', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER ZEKE FOR LOGIN Take2023; --create user for login
ALTER ROLE EntryOfPatient ADD MEMBER ZEKE; --assign user
  
```

The "Messages" pane below the script shows the execution results:

- Commands completed successfully.
- Completion time: 2023-04-30T22:40:49.1296535-04:00

The status bar at the bottom indicates "Query executed successfully." and shows statistics: Ln 9, Col 42, Ch 42, and INS.

Security 2:

```

CREATE ROLE WorkersEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Staff --set table
TO WorkersEntry;
CREATE LOGIN UniMedical --set login id
WITH PASSWORD = '5643287', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER Shiv FOR LOGIN UniMedical; --create user for login
ALTER ROLE WorkersEntry ADD MEMBER Shiv; --assign use
  
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, etc. The main pane displays T-SQL code for creating a database role and a user:

```

CREATE ROLE WorkersEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Staff --set table
TO WorkersEntry;
CREATE LOGIN UniMedical --set login id
WITH PASSWORD = '5643287', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER Shiv FOR LOGIN UniMedical; --create user for login
ALTER ROLE WorkersEntry ADD MEMBER Shiv; --assign use

```

The Messages pane at the bottom shows the command completed successfully with a completion time of 2023-04-30T22:43:07.1926575-04:00.

Security 3:

```

CREATE ROLE VisitorEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Visitors --set table
TO VisitorEntry;
CREATE LOGIN VisitorMedical --set login id
WITH PASSWORD = '87654', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER Lion FOR LOGIN VisitorMedical; --create user for login
ALTER ROLE VisitorEntry ADD MEMBER Lion; --assign use

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various system and user tables within the 'CenterForMedical' database. The main pane displays T-SQL code for creating a database role named 'VisitorEntry'. The code includes granting update, insert, and delete permissions on the 'Visitors' table to the 'VisitorEntry' role. It also creates a login 'VisitorMedical' with password '87654', sets it as the default database, creates a user 'Lion' for the login, and adds 'Lion' to the 'VisitorEntry' role.

```

CREATE ROLE VisitorEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Visitors --set table
TO VisitorEntry;
CREATE LOGIN VisitorMedical --set login id
WITH PASSWORD = '87654', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER Lion FOR LOGIN VisitorMedical; --create user for login
ALTER ROLE VisitorEntry ADD MEMBER Lion; --assign use
  
```

Messages pane: Commands completed successfully.
Completion time: 2023-04-30T22:46:47.8238293-04:00

Query executed successfully.

Security 4:

```

CREATE ROLE AppointmentEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Appointments --set table
TO AppointmentEntry;
CREATE LOGIN AppointmentHistory --set login id
WITH PASSWORD = '456789', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER App FOR LOGIN AppointmentHistory; --create user for login
ALTER ROLE AppointmentEntry ADD MEMBER App; --assign use
  
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various system and user tables under the 'CenterForMedical' database. The main pane displays T-SQL code for creating a database role 'AppointmentEntry' with UPDATE, INSERT, and DELETE permissions on the 'Appointments' table. It also creates a login 'AppointmentHistory' with password '456789', sets it as the default database, and creates a user 'App' for this login. Finally, it adds 'App' to the 'AppointmentEntry' role.

```

CREATE ROLE AppointmentEntry; -- create user defined database role
GRANT UPDATE,INSERT,DELETE -- set update,insert,delete permission
ON Appointments --set table
TO AppointmentEntry;
CREATE LOGIN AppointmentHistory --set login id
WITH PASSWORD = '456789', -- set password
DEFAULT_DATABASE = CenterForMedical; --set default database
CREATE USER App FOR LOGIN AppointmentHistory; --create user for login
ALTER ROLE AppointmentEntry ADD MEMBER App; --assign use
  
```

Messages pane: Commands completed successfully.
Completion time: 2023-04-30T22:48:56.0184093-04:00

Query executed successfully.

7. Business Report:

Business Report 1

```

CREATE PROCEDURE PatientHeight @Height int
AS
BEGIN
    SELECT count(PatientID) as PID
    FROM PatientRecords
    WHERE PatientHeight >= @Height ;
END;
GO
  
```

```

USE CenterForMedical;
GO
EXEC PatientHeight @Height = 30;
GO
  
```

```

CREATE PROCEDURE PatientHeight @Height int
AS
BEGIN
    SELECT count(PatientID) as PID
    FROM PatientRecords
    WHERE PatientHeight >= @Height ;
END;
GO

USE CenterForMedical;
GO

EXEC PatientHeight @Height = 30;
GO

```

Results

PID
10

Query executed successfully.

Business Report 2

```
CREATE PROCEDURE findNumberOfAdmissionsFromAndBefore @DateTime1 DATETIME,
@DateTime2 DATETIME
```

```
AS
```

```
BEGIN
```

```
    SELECT count(AppResID) as TotalAdmissions
        FROM ApparatusReservation
        WHERE CheckInTime >= @DateTime1 AND CheckOutTime <= @DateTime2 ;
```

```
END;
```

```
GO
```

```
USE CenterForMedical;
```

```
GO
```

```
EXEC findNumberOfAdmissionsFromAndBefore @DateTime1 = '2023-05-01 10:00:00',
@DateTime2 = '2023-05-31 11:59:00';
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, Graph Tables, and multiple dbo.* tables. The main pane displays a T-SQL script for a stored procedure named 'findNumberOfAdmissionsFromAndBefore'. The script counts the number of admissions from a specific date range. Below the script, a 'Results' tab shows a single row with 'TotalAdmissions' set to 0. A status bar at the bottom indicates the query was executed successfully.

```

CREATE PROCEDURE findNumberOfAdmissionsFromAndBefore @DateTime1 DATETIME, @DateTime2 DATETIME
AS
BEGIN
    SELECT count(AppResID) as TotalAdmissions
    FROM ApparatusReservation
    WHERE CheckInTime >= @DateTime1 AND CheckOutTime <= @DateTime2 ;
END;
GO

USE CenterForMedical;
GO

EXEC findNumberOfAdmissionsFromAndBefore @DateTime1 = '2023-05-01 10:00:00', @DateTime2 = '2023-05-31 11:59:00';
GO

```

Business Report 3:

```

CREATE PROCEDURE PatientsRecords @DateTime1 DATETIME, @DateTime2
DATETIME
AS
BEGIN
    SELECT count(PatientRecordID) as PatientRecords
        FROM PatientRecords
        WHERE CheckInTime >= @DateTime1 AND CheckOutTime <= @DateTime2 ;
END;

```

GO

```

USE CenterForMedical;
GO

```

```

EXEC PatientsRecords @DateTime1 = '2023-05-01 10:00:00', @DateTime2 = '2023-05-31
11:59:00';
GO

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, Graph Tables, and many dbo.* tables. The main pane displays a T-SQL script for a stored procedure named 'PatientsRecords'. The script uses parameters @DateTime1 and @DateTime2 to count patient records between specific dates. It includes a USE statement for the 'CenterForMedical' database and an EXEC statement with specific date values. The Results tab shows the output: a single row with 'PatientRecords' value 0.

```

CREATE PROCEDURE PatientsRecords @DateTime1 DATETIME, @DateTime2 DATETIME
AS
BEGIN
    SELECT count(PatientRecordID) as PatientRecords
    FROM PatientRecords
    WHERE CheckInTime >= @DateTime1 AND CheckOutTime <= @DateTime2 ;
END;
GO

USE CenterForMedical;
GO

EXEC PatientsRecords @DateTime1 = '2023-05-01 10:00:00', @DateTime2 = '2023-05-31 11:59:00';
GO

```

Query executed successfully.

Business Report 4:

```

CREATE PROCEDURE PatientWeight @Weight int
AS
BEGIN
    SELECT count(PatientID) as PID
        FROM PatientRecords
        WHERE PatientWeight >= @Weight ;
END;
GO

USE CenterForMedical;
GO
EXEC PatientWeight @Weight = 80;
GO

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, FileTables, External Tables, Graph Tables, and numerous dbo.* tables. The main pane displays a T-SQL script for creating a stored procedure named PatientWeight. The script includes a SELECT statement to count PatientID from PatientRecords where PatientWeight is greater than or equal to the input parameter @Weight, followed by a USE CenterForMedical; GO statement and an EXEC PatientWeight @Weight = 80; GO statement. The Results tab shows the output: a single row with PID 1 and value 5. A status bar at the bottom indicates the query was executed successfully.

```
CREATE PROCEDURE PatientWeight @Weight int
AS
BEGIN
    SELECT count(PatientID) as PID
    FROM PatientRecords
    WHERE PatientWeight >= @Weight ;
END;
GO

USE CenterForMedical;
GO

EXEC PatientWeight @Weight = 80;
GO
```

Results

PID
1
5

Message

Query executed successfully.

DOLLAR (16.0 RTM) | DOLLAR\nehar (62) | CenterForMedical | 00:00:00 | 1 rows

Ln 4 Col 36 Ch 36 INS

Conclusion:

This project's extensive database establishment capabilities are what make it significant. I learned a lot during the design, implementation, testing, and debugging phases and faced difficulties I had never seen before in other laboratories. My ability to use SQL statements has undoubtedly increased, as has the depth of my understanding of the topic. Additionally, the Microsoft SQL server management system turned out to be a useful tool for building a database that can be used in real-world scenarios. Data retrieval and updating were made simple by its user-friendly interface, which also offered trustworthy backup and security safeguards.

Remarks:

Making tables: Since making tables is one of SQL's fundamental features, employing it in your project would have given you the chance to put your understanding of the SQL syntax for doing so to the test. You would have gained knowledge of how to design constraints and relationships between tables, as well as how to define columns and data types.

Triggers: When specific events, such as adding or modifying data, take place, triggers are used to automate database processes. You could have learned how to create SQL code that reacts to these events by implementing triggers in your project, such as updating associated tables, notifying recipients, or reporting changes.

Special Procedures: Special procedures are reusable chunks of SQL code that carry out certain duties. Examples include stored procedures and user-defined functions. You could have learned how to create modular, effective SQL code that can be applied to numerous queries or apps by using specific procedures in your project.

Views: Views are virtual tables that display a specific subset of data from one or more database tables in a customizable way. You could have learned how to use SQL to filter, aggregate, and join data such that it is more available to and valuable to end users if you had created views in your project.

Scripts: SQL scripts are collections of SQL statements that can be executed collectively to complete a series of tasks, including filling a database or updating numerous tables simultaneously. You could have developed your ability to write intricate SQL code that coordinates numerous tasks and manages mistakes and exceptions by using scripts in your project.

User created functions: You can define your own unique custom functions in SQL, enabling you to carry out intricate calculations or data manipulations. You could have learned how to create SQL code that contains business logic and can be used to numerous queries by using user-defined functions in your project.

Business reports: Business reports are papers that display database data in a structured and illustrative manner using graphs, tables, and charts. You may have learned how to use SQL to query, analyze, and present data in a way that satisfies the demands of stakeholders and decision-makers by developing business reports for your project.

Multiple SQL statements are combined into a single, atomic operation to form a transaction, which is a unit of work. You might have developed your skills in writing SQL code that assures data integrity, consistency, and concurrency in a multi-user environment had you used transactions in your project.