



SYRACUSE UNIVERSITY ENGINEERING & COMPUTER SCIENCE

Intro To Database Management System CSE581 - Spring 2023

Project 2 : Patient Access and Registration

Nirmit Patel
npatel13@syr.edu
992323441

Table Of Content

A. Abstract.....	3
B. Design Phase	
a. Introduction.....	4
b. Database Design.....	4
c. E-R Diagram.....	6
d. Schema of the Tables.....	7
C. Implementation	
a. Queries To Create Table.....	14
b. Queries To Insert Data.....	19
D. Testing	
a. Views.....	31
b. Stored Procedure.....	39
c. User Defined Functions.....	47
d. Triggers.....	55
e. Transection.....	62
f. Script.....	70
E. Conclusion.....	74
F. Appendix.....	75

A. Abstract

The following project is centered around the creation of a database from scratch for a healthcare system, specifically for patient access and registration. The project involved designing 20 tables and making design decisions regarding the normalization of the database. The insertion of data into these tables was also an integral part of the initial development. As the healthcare system deals with confidential data, it was imperative to create a user-friendly interface and ensure that the data was stored in a judicious manner. Security risks and the flexibility of the database design were also taken into account as the database size increased.

The developed database also includes views, functions, triggers, stored procedures, transactions, and user scripts to address various business scenarios. These scenarios were tested, and the design was proven to be robust and working efficiently. The flexibility of the database design was prioritized so that scalability and security concerns can be addressed in the future. The normalization structure was followed strictly to prevent data redundancy. Overall, the developed database will aid in managing patient access and registration efficiently while maintaining data confidentiality and security.

B. Design Phase

a. Introduction:

The design phase of a database is of utmost importance as it directly affects the functionality and potential for future feature enhancements. The system's design plays a crucial role in determining the success of the database.

The name of my Database is : PAR

Following a comprehensive analysis and the implementation of multiple normalization stages, I have determined 20 tables that will adequately address the problem statement and account for various scenarios. This stage involves creating an Entity-Relationship Diagram (ERD) to illustrate the interrelationships between tables, followed by outlining the attributes and constraints of each table.

b. Database Design

During the design process I identify two main relationships between tables

1. One - to - one relationship
2. One - to - many relationship

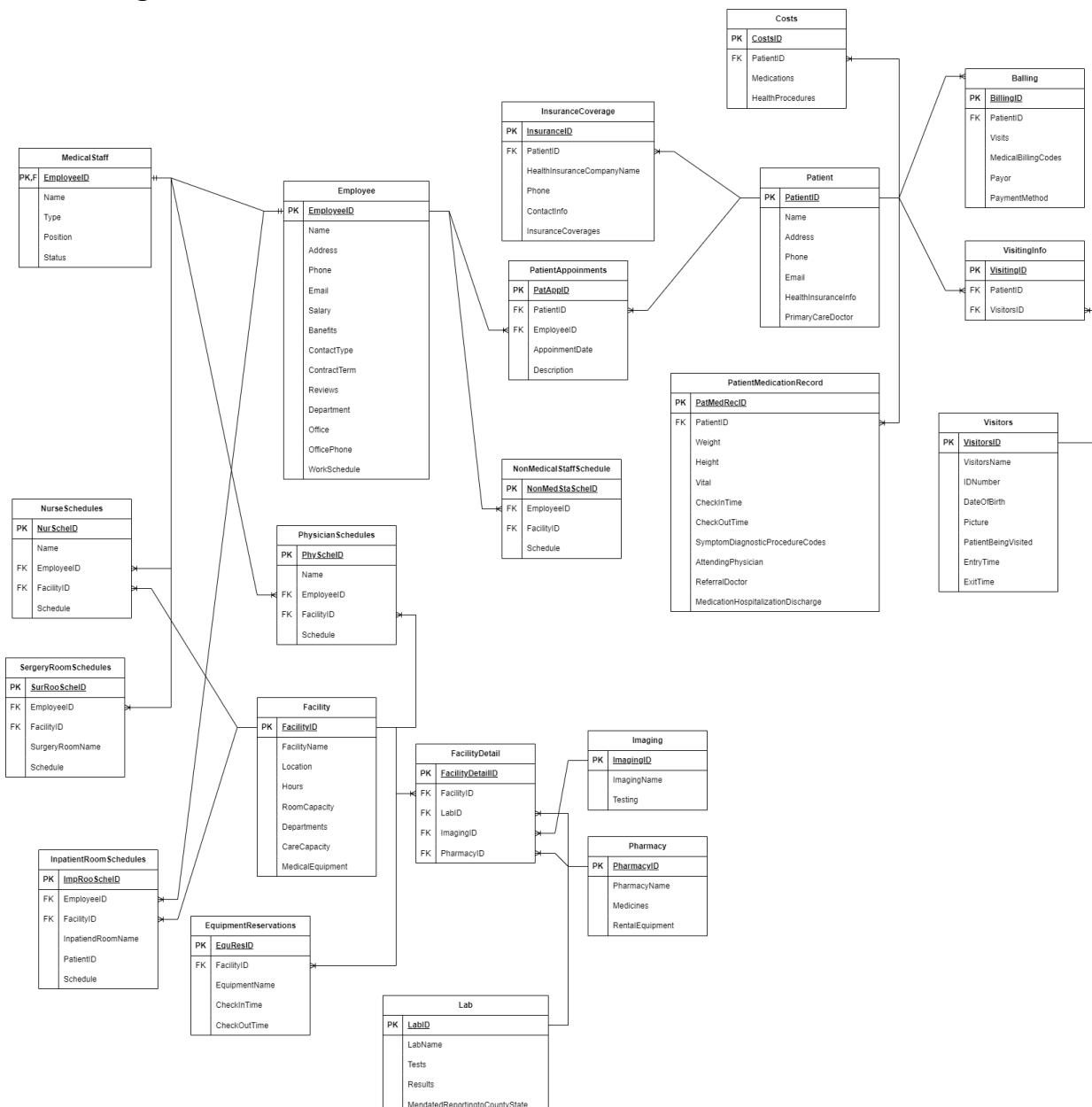
I have incorporated the identified use cases into my design and implementation by utilizing different table join methods. Although there are multiple ways to join tables, I have specifically designed them according to the requirements mentioned in the problem statement.

According to the diagram below, I have defined the following relationships among the tables :

1. Facility :
 - a. Has one-to-many relationship with NonMedicalStaffSchedule.
 - b. Has one-to-many relationship with SurgeryRoomSchedule.
 - c. Has one-to-many relationship with InpatientRoomSchedule.
 - d. Has one-to-many relationship with EquipmentReservations.
 - e. Has one-to-many relationship with Facilitydetail.
2. Imaging:
 - a. Has one-to-many relationship with Facilitydetail.
3. Lab:
 - a. Has one-to-many relationship with Facilitydetail.

4. Pharmacy:
 - a. Has one-to-many relationship with Facilitydetail.
5. Patient:
 - a. Has one-to-many relationship with Costs.
 - b. Has one-to-many relationship with InsuranceCoverage.
 - c. Has one-to-one relationship with PatientMedicalRecords.
 - d. Has one-to-many relationship with VisitingInfo.
 - e. Has one-to-many relationship with Billing.
 - f. Has one-to-many relationship with PatientAppointments.
6. Visitors:
 - a. Has one-to-many relationship with VisitingInfo.
7. Employee:
 - a. Has one-to-many relationship with PatientAppointments.
 - b. Has one-to-many relationship with NonMedicalStaffSchedule.
 - c. Has one-to-many relationship with MedicalStaff.
8. MedicalStaff:
 - a. Has one-to-many relationship with PhysicianSchedules.
 - b. Has one-to-many relationship with NurseSchedules.

c. E-R Diagram



d. Schema Of The Tables

Tables	Column Name	Data Type	Keys
Patient	PatientID	INT	PK
	Name	VARCHAR(50)	
	Address	VARCHAR(100)	
	Phone	VARCHAR(50)	
	Email	VARCHAR(50)	
	HealthInsuranceInfo	VARCHAR(50)	
	PrimaryCareDoctor	VARCHAR(50)	
Billing	BillingID	INT	PK
	PatientID	INT	FK
	Visits	VARCHAR(50)	
	MedicalBillingCodes	VARCHAR(50)	
	Payor	VARCHAR(50)	
	PaymentMethod	VARCHAR(50)	
Costs	CostsID	INT	PK
	PatientID	INT	FK
	Medications	VARCHAR(50)	
	HealthProcedures	VARCHAR(50)	
Employee	EmployeeID	INT	PK
	Name	VARCHAR(50)	

	Address	VARCHAR(50)	
	Phone	VARCHAR(50)	
	Email	VARCHAR(50)	
	Salary	VARCHAR(50)	
	Benefits	VARCHAR(50)	
	ContractType	VARCHAR(50)	
	ContractTerm	VARCHAR(50)	
	Reviews	VARCHAR(50)	
	Department	VARCHAR(50)	
	Office	VARCHAR(50)	
	OfficePhone	VARCHAR(50)	
	WorkSchedule	VARCHAR(50)	
Facility	FacilityID	INT	PK
	FacilityName	VARCHAR(50)	
	Location	VARCHAR(50)	
	Hours	DATETIME	
	RoomCapacity	VARCHAR(50)	
	Departments	VARCHAR(50)	
	CareCapacity	VARCHAR(50)	
	MedicalEquipment	VARCHAR(50)	
EquipmentReservations	EquResID	INT	PK
	FacilityID	INT	FK
	EquipmentName	VARCHAR(50)	

	CheckInTime	DATETIME	
	CheckOutTime	DATETIME	
Imaging	ImagingID	INT	PK
	ImagingName	VARCHAR(50)	
	Testing	VARCHAR(50)	
Lab	LabID	INT	PK
	LabName	VARCHAR(50)	
	Tests	VARCHAR(50)	
	Results	VARCHAR(50)	
	MandatedReporting toCountyState	VARCHAR(50)	
Pharmacy	PharmacyID	INT	PK
	PharmacyName	VARCHAR(50)	
	Medicines	VARCHAR(50)	
	RentalEquipment	VARCHAR(50)	
FacilityDetail	FacilityDetailID	INT	PK
	FacilityID	INT	FK
	LabID	INT	FK
	PharmacyID	INT	FK
	ImagingID	INT	FK
MedicalStaff	EmployeeID	INT	PK, FK

	Name	VARCHAR(50)	
	Type	VARCHAR(50)	
	Position	VARCHAR(50)	
	Status	VARCHAR(50)	
InpatientRoomSchedules	InpRooScheld	INT	PK
	EmployeeID	INT	FK
	FacilityID	INT	FK
	InpatientRoomName	VARCHAR(50)	
	PatientID	INT	
	Schedule	DATETIME	
InsuranceCoverage	InsuranceID	INT	PK
	PatientID	INT	FK
	HealthInsuranceCompanyName	VARCHAR(50)	
	Phone	VARCHAR(50)	
	ContactInfo	VARCHAR(50)	
	InsuranceCoverageS	VARCHAR(50)	
NonMedicalStaffSchedule	NonMedStaScheld	INT	PK
	EmployeeID	INT	FK
	Schedule	DATETIME	
	FacilityID	INT	FK

NurseSchedules	NurSchelID	INT	PK
	Name	VARCHAR(50)	
	EmployeeID	INT	FK
	FacilityID	INT	FK
	Schedule	DATETIME	
PatientAppointments	PatAppID	INT	PK
	PatientID	INT	FK
	EmployeeID	INT	FK
	AppointmentDate	DATETIME	
	Description	VARCHAR(50)	
PatientMedicalRecords	PatMedRecID	INT	PK
	PatientID	INT	FK
	Weight	INT	
	Height	INT	
	Vitals	VARCHAR(50)	
	CheckInTime	DATETIME	
	CheckOutTime	DATETIME	
	SymptomDiagnosticProcedureCodes	INT	
	AttendingPhysician	VARCHAR(50)	
	ReferralDoctor	VARCHAR(50)	
	MedicationHospitali	VARCHAR(50)	

	zationDischarge		
PhysicianSchedule s	PhyScheID	INT	PK
	Name	VARCHAR	
	EmployeeID	INT	FK
	FacilityID	INT	FK
	Schedule	DATETIME	
SurgeryRoomSche dules	SurRooScheID	INT	PK
	EmployeeID	INT	FK
	FacilityID	INT	FK
	SurgeryRoomName	VARCHAR(50)	
	Schedule	DATETIME	
Visitors	VisitorsID	INT	PK
	VisitorsName	VARCHAR(50)	
	IDNumber	VARCHAR(50)	
	DateOfBirth	DATETIME	
	Picture	IMAGE	
	PatientBeingVisited	VARCHAR(50)	
	EntryTime	DATETIME	
	ExitTime	DATETIME	
VisitingInfo	VisitingInfoID	INT	PK

	PatientID	INT	FK
	VisitorsID	INT	FK

C. Implementation

a. Queries To Create Table:

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

```
CREATE TABLE 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,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);
```

```
CREATE TABLE Costs (
    CostsID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    Medications VARCHAR(50) NULL,
    HealthProcedures VARCHAR(50) NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);
```

```
CREATE TABLE Employee (
    EmployeeID INT NOT NULL PRIMARY KEY IDENTITY,
    Name VARCHAR(50) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    Phone VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    Salary VARCHAR(50) NOT NULL,
    Benefits VARCHAR(50) NULL,
```

```

    ContractType VARCHAR(50) NULL,
    ContractTerm VARCHAR(50) NULL,
    Reviews VARCHAR(50) NULL,
    Department VARCHAR(50) NULL,
    Office VARCHAR(50) NULL,
    OfficePhone VARCHAR(50) NULL,
    WorkSchedule VARCHAR(50) NULL
);

```

```

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

```

```

CREATE TABLE EquipmentReservations (
    EquResID INT NOT NULL PRIMARY KEY IDENTITY,
    FacilityID INT NOT NULL,
    EquipmentName VARCHAR(50) NOT NULL,
    CheckInTime DATETIME NULL,
    CheckOutTime DATETIME NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID)
);

```

```

CREATE TABLE Imaging (
    ImagingID INT NOT NULL PRIMARY KEY IDENTITY,
    ImagingName VARCHAR(50) NOT NULL,
    Testing VARCHAR(50) NULL
);

```

```

CREATE TABLE Lab (
    LabID INT NOT NULL PRIMARY KEY IDENTITY,
    LabName VARCHAR(50) NOT NULL,
    Tests VARCHAR(50) NULL,
    Results VARCHAR(50) NULL,

```

```
MandatedReportingtoCountyState VARCHAR(50) NULL,  
);
```

```
CREATE TABLE Pharmacy (  
    PharmacyID INT NOT NULL PRIMARY KEY IDENTITY,  
    PharmacyName VARCHAR(50) NOT NULL,  
    Medicines VARCHAR(50) NULL,  
    RentalEquipment VARCHAR(50) NULL,  
);
```

```
CREATE TABLE FacilityDetail (  
    FacilityDetailID INT NOT NULL PRIMARY KEY IDENTITY,  
    FacilityID INT NOT NULL,  
    LabID INT NOT NULL,  
    ImagingID INT NOT NULL,  
    PharmacyID INT NOT NULL,  
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),  
    FOREIGN KEY (ImagingID) REFERENCES Imaging (ImagingID),  
    FOREIGN KEY (LabID) REFERENCES Lab (LabID),  
    FOREIGN KEY (PharmacyID) REFERENCES Pharmacy (PharmacyID)  
);
```

```
CREATE TABLE MedicalStaff (  
    EmployeeID INT NOT NULL PRIMARY KEY IDENTITY,  
    Name VARCHAR(50) NOT NULL,  
    Type VARCHAR(50) NOT NULL,  
    Position VARCHAR(50) NOT NULL,  
    Status VARCHAR(50) NOT NULL,  
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)  
);
```

```
CREATE TABLE InpatientRoomSchedules (  
    InpRooSchelID INT NOT NULL PRIMARY KEY IDENTITY,  
    EmployeeID INT NOT NULL,  
    FacilityID INT NOT NULL,  
    InpatientRoomName VARCHAR(50) NOT NULL,  
    PatientID INT NOT NULL,  
    Schedule DATETIME NULL,  
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),  
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
```

);

```
CREATE TABLE InsuranceCoverage (
    InsuranceID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    HealthInsuranceCompanyName VARCHAR(50) NOT NULL,
    Phone VARCHAR(50) NOT NULL,
    ContactInfo VARCHAR(50) NULL,
    InsuranceCovCoverages VARCHAR(50) NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);
```

```
CREATE TABLE NonMedicalStaffSchedule (
    NonMedStaSchelID INT NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    Schedule DATETIME NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID)
);
```

```
CREATE TABLE NurseSchedules (
    NurSchelID INT NOT NULL PRIMARY KEY IDENTITY,
    Name VARCHAR(50) NOT NULL,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    Schedule DATETIME NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);
```

```
CREATE TABLE PatientAppointments (
    PatAppID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    EmployeeID INT NOT NULL,
    AppointmentDate DATETIME NOT NULL,
    Description VARCHAR(50) NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);
```

```

CREATE TABLE PatientMedicalRecords (
    PatMedRecID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    Weight INT NULL,
    Height INT NULL,
    Vitals VARCHAR(50) NULL,
    CheckInTime DATETIME NULL,
    CheckOutTime DATETIME NULL,
    SymptomDiagnosticProcedureCodes INT NULL,
    AttendingPhysician VARCHAR(50) NULL,
    ReferralDoctor VARCHAR(50) NULL,
    MedicationHospitalizationDischarge VARCHAR(50) NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

```

```

CREATE TABLE PhysicianSchedules (
    PhySchelID INT NOT NULL PRIMARY KEY IDENTITY,
    Name VARCHAR(50) NOT NULL,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    Schedule DATETIME NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);

```

```

CREATE TABLE SurgeryRoomSchedules (
    SurRooSchelID INT NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    SurgeryRoomName VARCHAR(50) NOT NULL,
    Schedule DATETIME NULL
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);

```

```

CREATE TABLE Visitors (
    VisitorsID INT NOT NULL PRIMARY KEY IDENTITY,
    VisitorsName VARCHAR(50) NOT NULL,
    IDNumber VARCHAR(50) NOT NULL,
    DateOfBirth DATETIME NULL,

```

```

Picture IMAGE NULL,
PatientBeingVisited VARCHAR(50) NOT NULL,
EntryTime DATETIME NULL,
ExitTime DATETIME NULL
);

```

```

CREATE TABLE VisitingInfo (
    VisitingInfoID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    VisitorsID INT NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID),
    FOREIGN KEY (VisitorsID) REFERENCES Visitors (VisitorsID)
);

```

b. Queries To Insert Data

```

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

```

```

SET IDENTITY_INSERT Billing ON
INSERT Billing(BillingID, PatientID, Visits, MedicalBillingCodes, Payor,
PaymentMethod)VALUES
(1, 1, NULL, 'O58155480', NULL, 'Card'),
(2, 2, NULL, 'A58155480', NULL, 'Card'),
(3, 3, NULL, 'E24659070', NULL, 'Card'),
(4, 4, 'Good visits', 'L27021613', NULL, 'Cash'),
(5, 5, 'Good visits', 'P20513244', NULL, 'Card'),
(6, 6, 'Good visits', 'A22638768', NULL, 'Card'),
(7, 7, 'Good visits', 'E22948025', NULL, 'Cash'),
(8, 8, 'Good visits', 'L47581032', NULL, 'Cash'),
(9, 9, NULL, 'P95159496', NULL, 'Card'),
(10, 10, NULL, 'F89141139', NULL, 'Cash')
SET IDENTITY_INSERT Billing OFF

```

```

SET IDENTITY_INSERT Costs ON
INSERT Costs(CostsID, PatientID, Medications, HealthProcedures)VALUES
(1,5, 'Acetaminophen', 'Auscultation'),
(2,6, 'Cyclobenzaprine', 'Medical inspection '),
(3,8, 'Januvia', 'Palpation'),
(4,9, 'Omeprazole', 'Percussion (medicine)'),
(5,10, 'Adderall', 'Vital signs measurement'),
(6,1, 'Cymbalta', 'Biopsy test'),
(7,3, 'Jardiance', 'Blood test'),
(8,2, 'Onpattro', 'Stool test'),
(9,4, 'Amitriptyline', 'Urinalysis'),
(10,7, 'Doxycycline', 'Cardiac stress test')
SET IDENTITY_INSERT Costs OFF

```

```

SET IDENTITY_INSERT Lab ON
INSERT Lab(LabID, LabName, Tests, Results,
MandatedReportingtoCountyState)VALUES
(1, 'Altman Lab', 'Genetic Testing', 'Positive', NULL),
(2, 'Dean Lab', 'Kidney Tests', 'Positive', 'Reported'),
(3, 'Coppage Lab', 'Hepatitis Testing', 'Negative', 'Reported'),
(4, 'McCamant Lab', 'Thyroid Tests', 'Positive', 'Reported'),
(5, 'Immunization Lab', 'Laboratory Tests', 'Positive', 'Reported'),
(6, 'Elder Lab', 'Prenatal Testing', 'Negative', 'Reported'),
(7, 'DeAngelis Lab', 'Urinalysis', 'Positive', 'Reported'),

```

```

(8, 'Vertino Lab', 'Albumin Blood Test', 'Negative', NULL),
(9, 'Waugh Lab', 'Acid-Fast Bacillus (AFB) Tests', 'Positive', 'Reported'),
(10, 'White, Patricia Lab', 'Blood Count Tests', 'Negative', 'Reported')
SET IDENTITY_INSERT Lab OFF

SET IDENTITY_INSERT Facility ON
INSERT Facility(FacilityID, FacilityName, Location, Hours, RoomCapacity, Departments,
CareCapacity, MedicalEquipment) VALUES
(1, 'Cliffside Renal Dialysis', 'Flushing', '2005-12-22', '3 rooms 6 beds', 'pediatrics', 6, 3),
(2, 'Castle Hill Medical Center of New York Inc.', 'Bronx', '2007-01-01', '3 rooms 3 beds',
'surgery', 3, 3),
(3, 'Primary Care Associates', 'Endicott', '1998-02-01', '4 rooms 8 beds', 'Hospital
Extension Clinic', 10, 8),
(4, 'Lutheran Center at Poughkeepsie, Inc', 'Poughkeepsie', '1998-08-24', '3 rooms 6
beds', 'pediatrics', 6, 3),
(5, 'Meadowbrook Endoscopy Center', 'Westbury', '2008-10-15', '4 rooms 4 beds',
'surgery', 4, 4),
(6, 'New York Endoscopy Center', 'White Plains', '2009-10-05', '10 rooms 20 beds',
'Diagnostic and Treatment Center', 20, 20),
(7, 'UHCC - Madison Irving', 'Syracuse', '2010-06-14', '5 rooms 8 beds', 'Certified Home
Health Agency', 20, 15),
(8, 'Warner Place', 'Jamestown', '2010-07-28', '5 rooms 8 beds', 'Adult Day Health Care
Program - Offsite', 10, 8),
(9, 'Griffiss Surgery Center', 'Rome', '2012-01-03', '5 rooms 5 beds', 'surgery', 5, 5),
(10, 'H.K. Freedman Renal Center', 'Plattsburgh', '2013-12-05', '4 rooms 8 beds',
'Diagnostic and Treatment Center', 10, 8)
SET IDENTITY_INSERT Facility OFF

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

```

(10, 6, 'Surgical Lights', 2021-02-07, 2021-02-10)

SET IDENTITY_INSERT EquipmentReservations OFF

SET IDENTITY_INSERT Imaging ON

INSERT Imaging(ImagingID, ImagingName, Testing)VALUES

(1, 'Direct', 'MRI'),

(2, 'Indirect', 'CAT'),

(3, 'Direct', 'MRI'),

(4, 'Direct', 'MRI'),

(5, 'Direct', 'MRI'),

(6, 'Direct', 'CAT'),

(7, 'Indirect', 'MRI'),

(8, 'Indirect', 'MRI'),

(9, 'Indirect', 'MRI'),

(10, 'Indirect', 'CAT')

SET IDENTITY_INSERT Imaging OFF

SET IDENTITY_INSERT Pharmacy ON

INSERT Pharmacy(PharmacyID, PharmacyName, Medicines,

RentalEquipment)VALUES

(1, 'Boone Drug', 'Allopurinol, Alendronate, Alemtuzumab', 'Defibrillators'),

(2, 'Bartell Drugs', 'Ifosfamide, Imatinib, Imipenem', 'Anesthesia Machines'),

(3, 'CVS Pharmacy', 'Dacarbazine, Dactinomycin, Dapsone', 'Patient Monitors'),

(4, 'Discount Drug Mart', 'L-glutamine, abetalol, Lamivudine', 'EKG/ECG
Machines" Heparin Lock Flush '>,

(5, 'Hartig Drug', 'Heparin Lock Flush', 'Sterilizers'),

(6, 'London Drugs', 'Nelarabine, Nelfinavir, Neuromuscular', 'Surgical Tables'),

(7, 'Family Pharmacy', 'Omeprazole, Ondansetron, Oxycodone', 'Blanket and Fluid
Warmers'),

(8, 'Pharmasave', 'Valproic acid, Vancomycin, Vinblastine', 'Electrosurgical Units'),

(9, 'Good Neighbor', 'Efavirenz, Efavirenz / emtricitabine', 'Surgical Tables'),

(10, 'Lawtons', 'Baclofen, Bleomycin, Bortezomib', 'EKG/ECG Machines')

SET IDENTITY_INSERT Pharmacy OFF

SET IDENTITY_INSERT FacilityDetail ON

INSERT FacilityDetail(FacilityDetailID, FacilityID, LabID, ImagingID,

PharmacyID)VALUES

(1, 1, 1, 4, 7),

(2, 2, 3, 5, 8),

(3, 3, 5, 1, 3),

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

```
SET IDENTITY_INSERT FacilityDetail OFF
```

```
SET IDENTITY_INSERT Employee ON
```

```
INSERT Employee(EmployeeID, Name, Address, Phone, Email, Salary, Benefits,
ContractType, ContractTerm, Reviews, Department, Office, OfficePhone,
WorkSchedule) VALUES
```

```
(1,'Finnian Moyer','15 Alton Street Brooklyn, NY 11207','6558307415',
,'FinnianMoyer@gmail.com','50534','bonus','fixed-price',NULL,NULL,'medical',NULL,'86
84091023','9-17'),
(2,'Kiyan Worthington','501 South Peachtree Lane Bronx, NY 10463',
,'8845227792','KiyanWorthington@gmail.com','37453',NULL,'fixed-price',NULL,NULL,'ad
ministration',NULL,'8376350088','9-17'),
(3,'Minnie Glenn','785 Carriage Dr.West Babylon, NY 11704','0170735302'
,'MinnieGlenn@gmail.com','63765',NULL,'fixed-price',NULL,NULL,'logistics',NULL,'0463
209805','9-17'),
(4,'Inaaya Dyer','81 Old Dunbar Lane Jackson Heights, NY 11372',
,'6310627418','InaayaDyer@gmail.com','78345','bonus','fixed-price',NULL,NULL,'adminis
tration',NULL,'7200016468','8-16'),
(5,'Tariq Myers','2 North Berkshire Avenue Yonkers, NY 10701',
,'5295374578','TariqMyers@gmail.com','68755','bonus','cost-reimbursement',NULL,NUL
L,'administration',NULL,'9611106482','8-16'),
(6,'Vihaan Hulme','24518 Melyssa Meadow Apt. 322 West Adalberto, IN 91173',
,'6069878004','pfeffer.cydney@yahoo.com','54464','bonus','fixed-price',NULL,NULL,'adm
inistration',NULL,'1628118873','9-17'),
(7,'Barry Carpenter','4549 Wilton Valley Sauerberg, PA 23957','6217172459'
,'schuppe.braeden@yahoo.com','45645','bonus','fixed-price',NULL,NULL,'administration'
,NULL,'0772970610','9-17'),
(8,'Jaeden Noel','15722 Wilkinson Isle East Dellastad, WA 49728',
,'5041957090','rae95@hotmail.com','55645','bonus','fixed-price',NULL,NULL,'administrati
on',NULL,'0422424858','9-17'),
(9,'Darrell Naylor','5334 VonRueden Lakes Suite 683 Goodwinville, NE 81010',
,'7909854128','alakin@gmail.com','45645','bonus','fixed-price',NULL,NULL,'administratio
n',NULL,NULL,'9-17'),
```

(10,'Niyah Kemp','110 Skyla Estate Port Mittie, SC 49513',
'6740838458','crooks.alaina@yahoo.com','45645','bonus','fixed-price',NULL,'logistics','a
dministration',NULL,'0814764446','9-17'),
(11,'Rares Hopper','693 Estelle Cliffs Suite 704 Michelemouth, SC 53933',
'4045537794','hannah97@spinka.net','67657','bonus','fixed-price',NULL,NULL,'medical',
NULL,'1805733194','9-17'),
(12,'Hughie Whittington','401 Odessa Turnpike Port Budtown, AR 98911-3045',
'4671739881','zmaggio@wuckert.com','65454','bonus','fixed-price',NULL,NULL,'medical'
,NULL,'7915637384','9-17'),
(13,'Teddie Curran','671 Gottlieb Row Apt. 177 Schadenberg, AL
12452-0485','7967015831'
, 'eli21@schmeler.net','87685','bonus','fixed-price',NULL,NULL,'medical',NULL,'3969268
819','9-17'),
(14,'Jaheim Friedman','892 Matilda Plains Apt. 923 New Arno, MD 53607',
'9131847291','xledner@huel.com','45656','bonus','fixed-price',NULL,NULL,'medical',NU
LL,'2831681366','8-16'),
(15,'Declan Shah','372 Cristopher Roads Lake Santa, AZ 76817',
'8292203050','alberta.casper@lindgren.com','74536','bonus','fixed-price',NULL,NULL,'m
edical',NULL,'4081907698','8-16'),
(16,'Carly Collier','331 Leopold Prairie Apt. 792 West Tracemouth, MA 52000-1820',
'5063202635','delores.christiansen@gmail.com','46345',NULL,'fixed-price',NULL,NULL,'
medical',NULL,'0039484081','8-16'),
(17,'Huxley Bailey','37033 Izaiah Ridges Gianniport, DE 42915','8964400286'
, 'tillman.eloise@morissette.biz','76574',NULL,'fixed-price',NULL,NULL,'medical',NULL,'4
828147894','8-16'),
(18,'Harlow Emerson','823 Alberta Fork Suite 359 Lake Dorothea, MA 48558',
'5192833596','gunnar55@mraz.com','78567',NULL,'fixed-price',NULL,NULL,'logistics',N
ULL,'8050950930','8-16'),
(19,'Teddy Grey','255 Carter Junction Suite 221 Devynbury, SC 86155-1203',
'9828774679','hills.andreanne@yahoo.com','55724',NULL,'fixed-price',NULL,NULL,'med
ical',NULL,NULL,'8-16'),
(20,'Anne Hubbard','4733 Treutel Vista Apt. 708 West Matteoberg, LA 88682',
'6027894995','schmitt.helena@langosh.com','77644',NULL,'fixed-price',NULL,NULL,'log
istics',NULL,NULL,'8-16'),
(21,'Bethanie Delarosa','3670 Brekke Fork Suite 375 Fadelburgh, ND 39924-1567',
'6053408977','king.kellen@nolan.com','73454',NULL,'fixed-price',NULL,NULL,'medical',
NULL,'9257686337','8-16'),
(22,'Imani Currie','8317 Idella Underpass Lake Mack, MO 83033','4791179552'
, 'leuschke.anabel@gmail.com','76544','bonus','fixed-price',NULL,NULL,'logistics',NULL,
NULL,'8-16'),

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

SET IDENTITY_INSERT Employee OFF

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

```
(11,'Rares Hopper','Nurse','visiting','leave'),
(12,'Hughie Whittington','Nurse','PCP','active'),
(13,'Teddie Curran','Nurse','PCP','active'),
(14,'Jaheim Friedman','Nurse','PCP','retired'),
(15,'Declan Shah','Nurse','PCP','active'),
(16,'Carly Collier','Nurse','PCP','retired'),
(17,'Huxley Bailey','Nurse','PCP','active'),
(18,'Harlow Emerson','Nurse','attending','active'),
(19,'Teddy Grey','Nurse','attending','leave'),
(20,'Anne Hubbard','Nurse','visiting','active')
SET IDENTITY_INSERT MedicalStaff OFF
```

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

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

```
SET IDENTITY_INSERT InsuranceCoverage OFF
```

```
SET IDENTITY_INSERT NonMedicalStaffSchedule ON  
INSERT NonMedicalStaffSchedule(NonMedStaSchelD, EmployeeID, FacilityID,  
Schedule)VALUES
```

```
(1, 25, 5, '2021-05-01'),  
(2, 24, 4, '2021-05-02'),  
(3, 23, 3, '2021-05-03'),  
(4, 22, 2, '2021-05-04'),  
(5, 21, 1, '2021-05-05'),  
(6, 26, 6, '2021-05-06'),  
(7, 27, 7, '2021-05-07'),  
(8, 28, 8, '2021-05-08'),  
(9, 29, 9, '2021-05-09'),  
(10, 30, 10, '2021-05-10')
```

```
SET IDENTITY_INSERT NonMedicalStaffSchedule OFF
```

```
SET IDENTITY_INSERT NurseSchedules ON
```

```
INSERT NurseSchedules(NurSchelD,Name, EmployeeID, FacilityID, Schedule)  
VALUES
```

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

```
SET IDENTITY_INSERT NurseSchedules OFF
```

```
SET IDENTITY_INSERT PatientAppointments ON
```

```
INSERT PatientAppointments(PatAppID, PatientID, EmployeeID, AppointmentDate,  
Description) VALUES
```

```
(1,1,1,'2020-09-13',NULL),  
(2,2,5,'2020-11-18',NULL),  
(3,3,5,'2020-12-02',NULL),  
(4,4,3,'2021-03-22',NULL),  
(5,5,6,'2019-02-12',NULL),
```

```
(6,6,23,'2015-07-12',NULL),
(7,7,1,'2021-01-02',NULL),
(8,8,3,'2020-02-19',NULL),
(9,9,6,'2006-05-17',NULL),
(10,10,7,'2005-09-13',NULL)
```

```
SET IDENTITY_INSERT PatientAppointments OFF
```

```
SET IDENTITY_INSERT PatientMedicalRecords ON
INSERT PatientMedicalRecords(PatMedRecID, PatientID, Weight, Height, Vitals,
CheckInTime, CheckOutTime, SymptomDiagnosticProcedureCodes,
AttendingPhysician, ReferralDoctor, MedicationHospitalizationDischarge)VALUES
(1,1,71,171, NULL,'2021-05-02 01:20:03','2021-05-02 13:31:44',1,'Ismael Webb','Ismael
Webb', NULL),
(2,2,72,172, NULL, '2021-05-03 00:12:04', '2021-05-03 06:22:41',2,'Virgil
Jefferson','Virgil Jefferson', NULL),
(3,3,73,173, NULL, '2021-05-03 08:23:50', '2021-05-03 12:48:20',3,'Chad Bryant','Chad
Bryant', NULL),
(4,4,74,174, NULL,'2021-05-03 23:47:54', '2021-05-04 00:00:01',4,'Opal Benson','Opal
Benson', NULL),
(5,5,75,175, NULL, '2021-05-04 07:36:08','2021-05-06 22:28:24',5,'Jodi Bailey','Jodi
Bailey', NULL),
(6,6,76,176, NULL, '2021-05-08 01:49:53','2021-05-08 08:53:42',6,'Pamela
Chambers','Pamela Chambers', NULL),
(7,7,77,177, NULL, '2021-05-10 13:42:06','2021-05-13 15:33:00',7,'Lora Beck','Lora
Beck', NULL),
(8,8,78,178, NULL, '2021-05-14 03:08:23','2021-05-16 05:04:33',8,'Nichole
Klein','Nichole Klein', NULL),
(9,9,79,179, NULL, '2021-05-17 02:34:05','2021-05-17 11:42:52',9,'Donna
Hardy','Donna Hardy', NULL),
(10,10,80,180, NULL, '2021-05-18 01:35:16','2021-05-18 09:31:11',10,'Sadie
Martinez','Sadie Martinez', NULL)
SET IDENTITY_INSERT PatientMedicalRecords OFF
```

```
SET IDENTITY_INSERT PhysicianSchedules ON
INSERT PhysicianSchedules(PhySchID,Name, EmployeeID, FacilityID, Schedule)
VALUES
(1,'Finnian Moyer',1,1,'2021-05-21'),
(2,'Kiyan Worthington',2,2,'2021-04-21'),
(3,'Minnie Glenn',3,3,'2021-05-19'),
(4,'Inaaya Dyer',4,7,'2021-05-12'),
```

```
(5,'Tariq Myers',5,4,'2021-05-10'),
(6,'Vihaan Hulme',6,10,'2021-05-19'),
(7,'Barry Carpenter',7,5,'2021-05-21'),
(8,'Jaeden Noel',8,9,'2021-05-07'),
(9,'Darrell Naylor',9,6,'2021-05-08'),
(10,'Niyah Kemp',10,8,'2021-05-19')
```

```
SET IDENTITY_INSERT PhysicianSchedules OFF
```

```
SET IDENTITY_INSERT SurgeryRoomSchedules ON
INSERT SurgeryRoomSchedules(SurRooSchID, EmployeeID, FacilityID,
SurgeryRoomName, Schedule)VALUES
```

```
(1, 6, 1, 'firstclass1', '2021-05-01'),
(2, 1, 2, 'firstclass2', '2021-05-02'),
(3, 3, 3, 'secondclass1', '2021-05-01'),
(4, 5, 4, 'firstclass3', '2021-05-03'),
(5, 4, 5, 'secondclass2', '2021-05-01'),
(6, 9, 6, 'secondclass3', '2021-05-02'),
(7, 8, 7, 'firstclass4', '2021-05-07'),
(8, 7, 8, 'firstclass5', '2021-05-10'),
(9, 2, 9, 'firstclass6', '2021-05-15'),
(10, 10, 10, 'firstclass7', '2021-05-05')
```

```
SET IDENTITY_INSERT SurgeryRoomSchedules OFF
```

```
SET IDENTITY_INSERT Visitors ON
```

```
INSERT Visitors(VisitorsID, VisitorsName, IDNumber, DateOfBirth, Picture,
PatientBeingVisited, EntryTime, ExitTime) Values
```

```
(1,'Mario Baker','2374739417','1989-09-10', NULL, 'Visted', '2021-05-02
01:20:03','2021-05-02 13:31:44'),
(2,'Jennie Pierce','5567549402','1976-08-25', NULL, 'Visted', '2021-05-03 00:12:04',
'2021-05-03 06:22:41'),
(3,'Adam Rodriquez','9971788390','1979-10-26', NULL, 'Visted', '2021-05-03 08:23:50',
'2021-05-03 12:48:20'),
(4,'Angel Ruiz','6065530153','1999-06-22', NULL, 'Visted', '2021-05-03 23:47:54',
'2021-05-04 00:00:01'),
(5,'Lyle Summers','3429094491','1997-01-09', NULL, 'Visted', '2021-05-04
07:36:08','2021-05-06 22:28:24'),
(6,'Rose Jennings','6071126183','1963-08-03', NULL, 'Visted', '2021-05-08
01:49:53','2021-05-08 08:53:42'),
(7,'Valerie Gibson','0535989612','1980-09-25', NULL, 'Visted', '2021-05-10
13:42:06','2021-05-13 15:33:00'),
```

```
(8,'Neil Grant','1996830502','1985-12-25', NULL, 'Visted', '2021-05-14  
03:08:23','2021-05-16 05:04:33'),  
(9,'Andrew Rodriguez','3460661408','1993-03-21', NULL, 'Visted', '2021-05-17  
02:34:05','2021-05-17 11:42:52'),  
(10,'Kyle Marsh','1610857171','1969-05-10', NULL, 'Visted', '2021-05-18  
01:35:16','2021-05-18 09:31:11')  
SET IDENTITY_INSERT Visitors OFF
```

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

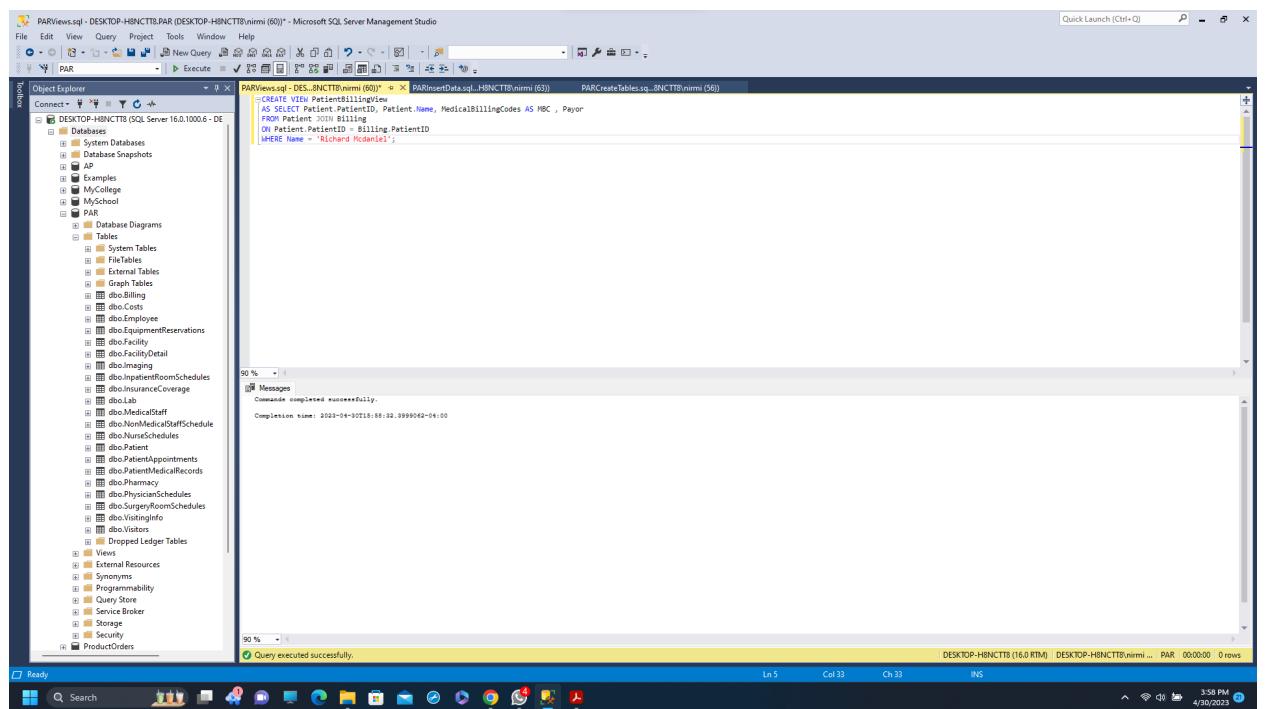
D. Testing

This section involves the creation of multiple views, stored procedures, user-defined functions, triggers, transactions, and scripts to ensure the effectiveness and correctness of the database design.

a. Views:

1. This view involves selecting a patient and retrieving their billing information under specific conditions. The output will include the patient's ID, name, medical billing codes, and payor.

```
CREATE VIEW PatientBillingView
AS SELECT Patient.PatientID, Patient.Name, MedicalBillingCodes AS MBC ,
Payor
FROM Patient JOIN Billing
ON Patient.PatientID = Billing.PatientID
WHERE Name = 'Richard McDaniel';
```



Testing the View :

SELECT * FROM PatientBillingView

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PAR'. The 'Tables' node under 'PAR' contains several tables including 'dbo.Employee', 'dbo.EquipmentReservations', 'dbo.Facility', 'dbo.InpatientRoomSchedules', 'dbo.insuranceCoverage', 'dbo.Lab', 'dbo.MedicalStaff', 'dbo.NonMedicalStaffSchedule', 'dbo.Officeschedule', 'dbo.Patient', 'dbo.PatientAppointments', 'dbo.PatientMedicalRecords', 'dbo.Pharmacy', 'dbo.PhysicianSchedules', 'dbo.PolicyRoomSchedules', 'dbo.VisitingInfo', and 'dbo.Visitors'. The 'Views' node is also visible. The central pane shows a query window with the following SQL statement:

```
SELECT * FROM PatientBillingView
```

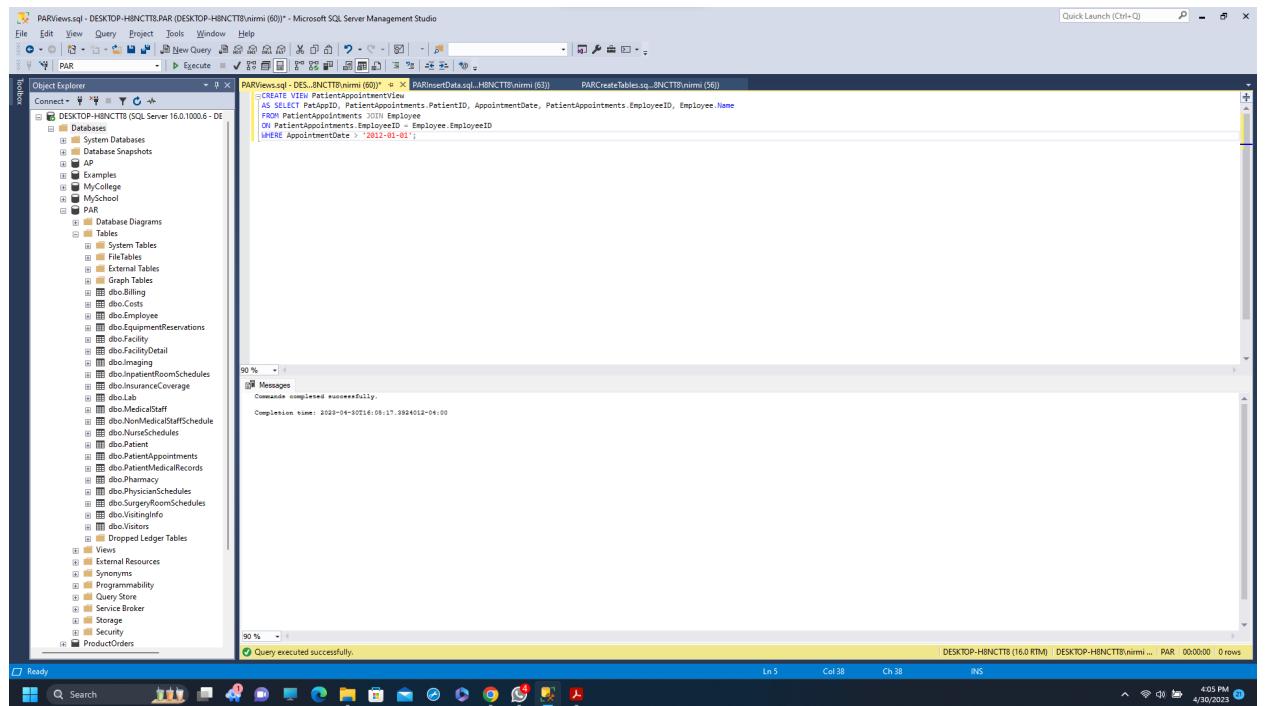
The results pane displays the output of the query:

PatientID	Name	MBC	Payor
1	Richard Modarrel	A58155489	NULL

Below the results, a message states "Query executed successfully." The status bar at the bottom right shows the session details: DESKTOP-H8NCTTB (16.0 RTM) | DESKTOP-H8NCTTB\mirmi | PAR | 00:00:00 | 1 rows.

2. This view utilizes a SELECT statement to retrieve information about patient appointments that occur after January 1st, 2012. The returned data includes the patient's appointment ID, patient ID, appointment date, employee ID, and the name of the employee.

```
CREATE VIEW PatientAppointmentView
AS SELECT PatAppID, PatientAppointments.PatientID, AppointmentDate,
PatientAppointments.EmployeeID, Employee.Name
FROM PatientAppointments JOIN Employee
ON PatientAppointments.EmployeeID = Employee.EmployeeID
WHERE AppointmentDate > '2012-01-01';
```



Testing the View :

```
SELECT * FROM PatientAppointmentView ORDER BY PatientID;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'PAR' containing various schemas like 'AP', 'MyCollege', 'MySchool', and 'dbo'. The 'dbo' schema contains many tables such as 'AP', 'Employee', 'EquipmentReservations', 'FacilityDetail', 'Inpatient', 'InsuranceCoverage', 'Lab', 'MedicalStaff', 'NonMedicalStaffSchedule', 'NonMedicStaffSchedules', 'Patient', 'PatientAppointments', 'PatientMedicalRecords', 'Pharmacy', 'PhysicianSchedules', 'PhysicianRoomSchedules', 'VisitingInfo', and 'Visitors'. The 'PARViews.sql' file in the 'PAR' folder is open in the main window, displaying the query:

```
SELECT * FROM PatientAppointmentView ORDER BY PatientID;
```

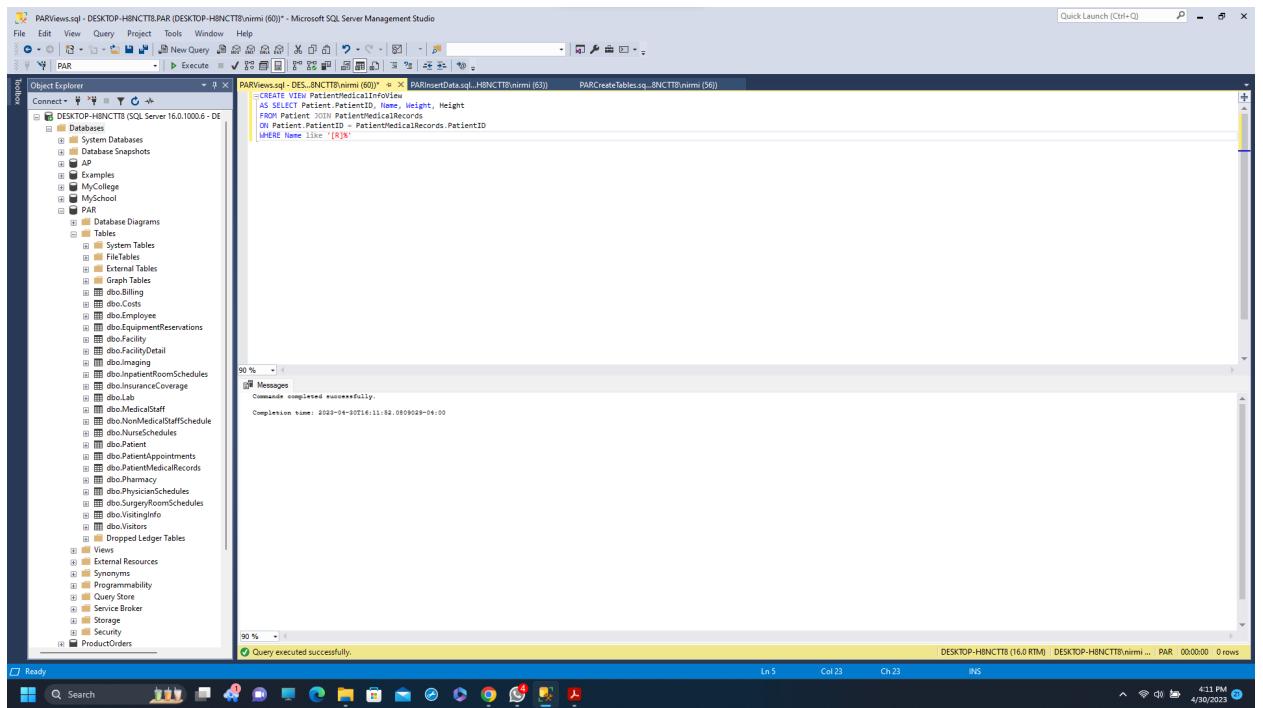
The results pane shows the output of the query:

PatAppID	PatientID	AppointmentDate	EmployeeID	Name
1	1	2020-09-13 00:00:00.000	1	Varian Moyer
2	2	2020-11-10 00:00:00.000	5	Tanq Myers
3	3	2020-12-02 00:00:00.000	5	Tanq Myers
4	4	2021-01-03 00:00:00.000	3	Mervin Glenn
5	5	2019-02-01 00:00:00.000	6	Wanda White
6	6	2015-07-11 00:00:00.000	23	Nathia Olsen
7	7	2021-01-02 00:00:00.000	1	Varian Moyer
8	8	2020-02-19 00:00:00.000	3	Mervin Glenn

A message at the bottom of the results pane says "Query executed successfully." The status bar at the bottom right shows "DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\Nirmi ... PAR 00:00:00 8 rows".

3. This view will display medical information of patients including their ID, name, weight, and height. However, the view is filtered to only include patients whose names contain the letter 'R'.

```
CREATE VIEW PatientMedicalInfoView
AS SELECT Patient.PatientID, Name, Weight, Height
FROM Patient JOIN PatientMedicalRecords
ON Patient.PatientID = PatientMedicalRecords.PatientID
WHERE Name like '[R]%'
```



Testing the View:
SELECT * FROM PatientMedicalInfoView;

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including the PAR database and its tables such as AP, Examples, MyCollege, MySchool, and various dbo tables like Facility, Employee, and MedicalStaff. The central pane shows a query window with the following SQL code:

```
SELECT * FROM PatientMedicalInfoView;
```

The results pane displays the output of the query:

PatientID	Name	Weight	Height
2	Richard Modrieh	72	172
5	Robin Valdez	75	175

A status bar at the bottom indicates "Query executed successfully." and "DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\Nirmit 00:00:00 2 rows".

4. This view utilizes a select statement to retrieve Physician Schedule information where PhySchelD equals 4. The returned columns include PhySchelD, EmployeeID, PhysicianInform, and status.

```
CREATE VIEW PhysicianScheduleView
AS SELECT PhySchelD, MedicalStaff.EmployeeID,
CONCAT(MedicalStaff.Name, ',', Position, ',', Schedule) AS PhysicianInform,
Status
FROM MedicalStaff JOIN PhysicianSchedules
ON MedicalStaff.EmployeeID = PhysicianSchedules.EmployeeID;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'PAR' selected. The right pane contains a query window with the following SQL code:

```
=CREATE VIEW PhysicianScheduleView
AS SELECT PhySchelD, MedicalStaff.EmployeeID,
CONCAT(MedicalStaff.Name, ',', Position, ',', Schedule) AS PhysicianInform,
Status
FROM MedicalStaff JOIN PhysicianSchedules
ON MedicalStaff.EmployeeID = PhysicianSchedules.EmployeeID;
```

Below the code, the 'Messages' section shows the message: "Command completed successfully." and the completion time: "Completion time: 2023-04-30T16:16:16.5406970+04:00". At the bottom of the screen, the Windows taskbar is visible with various icons and the system clock showing "4:36 PM 4/30/2023".

Testing the View:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including the PAR database which contains various tables and views. The central pane shows a query window with the following SQL code:

```
SELECT * FROM PhysicianScheduleView WHERE PhySchID = 4;
```

The results pane displays the output of the query:

PhySchID	EmployeeID	PhysicianInfo	Status	
1	4	Inaya Dyer PCP	May 12 2021 12:00AM	active

A status bar at the bottom indicates "Query executed successfully." and shows the system time as 4:17 PM on 4/30/2023.

b. Stored procedures:

1. This stored procedure is used to retrieve information about a facility. It begins by checking if a previous version of the procedure exists and drops it if so. It then creates a new procedure.

GO

```
IF OBJECT_ID('spFacilityInfo') IS NOT NULL
DROP PROC spFacilityInfo;
```

GO

```
CREATE PROC spFacilityInfo
@FacilityName VARCHAR(50)
AS
SELECT Location, Hours, RoomCapacity, Departments
FROM Facility
WHERE Facility.FacilityName = @FacilityName
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'PAR' with various objects like tables, stored procedures, and triggers. The central pane contains the T-SQL code for creating the stored procedure. The status bar at the bottom indicates the command was completed successfully.

```
IF OBJECT_ID('spFacilityInfo') IS NOT NULL
DROP PROC spFacilityInfo;
GO
CREATE PROC spFacilityInfo
@FacilityName VARCHAR(50)
AS
SELECT Location, Hours, RoomCapacity, Departments
FROM Facility
WHERE Facility.FacilityName = @FacilityName
```

Testing the Stored Procedure:

`EXEC spFacilityInfo 'Lutheran Center at Poughkeepsie, Inc';`

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PAR'. The 'PAR' database contains several schemas like 'AP', 'dbo', 'MyCollege', 'MySchool', and 'PAR'. Under 'PAR', there are tables such as 'Facility', 'FacilityDetail', 'FacilityHours', 'FacilityRooms', 'InsuranceCoverage', 'Lab', 'MedicalStaff', 'NursingSchedule', 'Patient', 'PatientAppointments', 'PatientMedicalRecords', 'Pharmacy', 'PhysicalSchedules', 'PhysicalRoomSchedules', 'VisitingInfo', and 'Visitors'. The 'PARCreateTables.sql' file is open in the center pane, showing the command `EXEC spFacilityInfo 'Lutheran Center at Poughkeepsie, Inc';`. The 'PARViews.sql' file is also listed in the object explorer. The 'Results' tab in the center pane shows the output of the query, which is a single row of data:

Location	Hours	RoomCapacity	Departments
Poughkeepsie	1998-08-24 00:00:00.000	3 rooms	6 beds
			pediatrics

A status bar at the bottom indicates 'Query executed successfully.' and shows the system time as 4:24 PM on 4/30/2023.

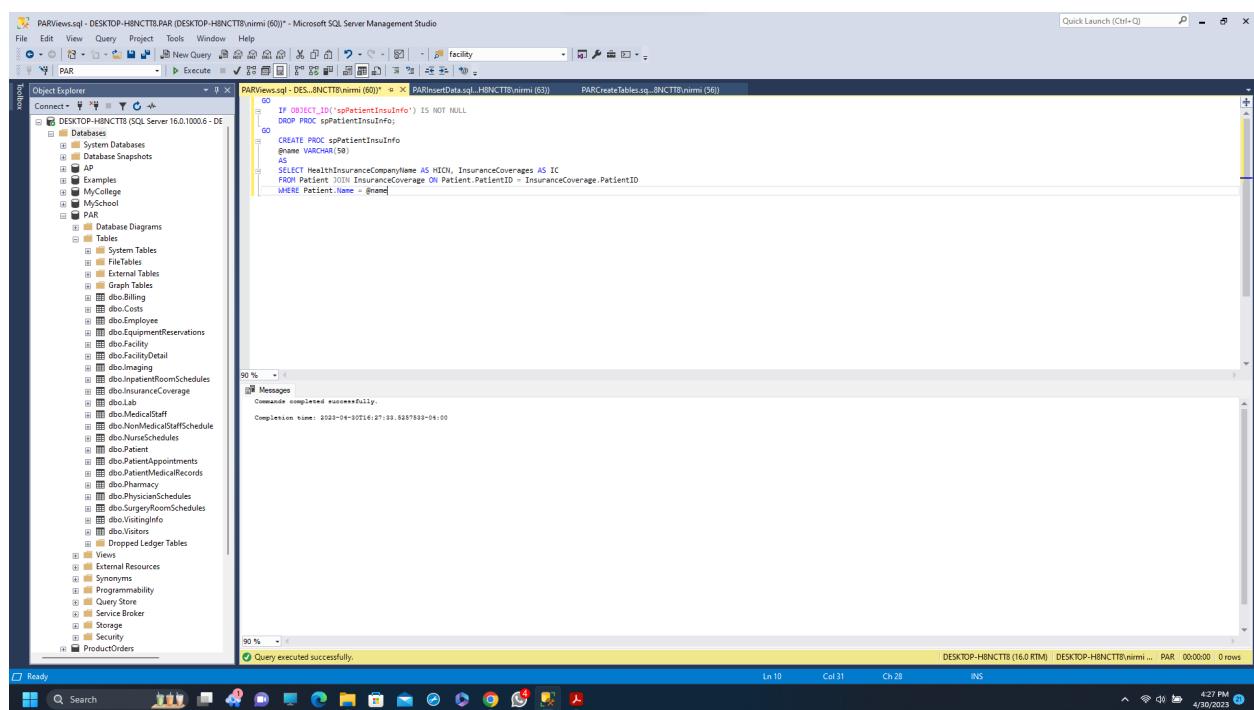
2. This stored procedure is designed to retrieve information about a patient's insurance coverage. It queries the database to fetch the name of the health insurance company and the details of the insurance coverage.

GO

```
IF OBJECT_ID('spPatientInsulInfo') IS NOT NULL  
DROP PROC spPatientInsulInfo;
```

GO

```
CREATE PROC spPatientInsulInfo  
@name VARCHAR(50)  
AS  
SELECT HealthInsuranceCompanyName AS HICN, InsuranceCov...  
AS IC  
FROM Patient JOIN InsuranceCoverage ON Patient.PatientID =  
InsuranceCoverage.PatientID  
WHERE Patient.Name = @name
```



Testing the Stored Procedure:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PAR' (DESKTOP-H8NCTTB.PAR). The 'PARViews.sql' file is open in the main pane, showing the T-SQL code for creating a view. Below it, the 'PARInsertData.sql' file is open, containing the stored procedure 'spPatientInfo'. A query window titled 'PARCreateTables.sql' is also visible. The status bar at the bottom indicates 'Query executed successfully.' and shows the system time as 4:28 PM on 4/30/2023.

```
EXEC spPatientInfo 'Paulette Walters';
```

3. This stored procedure is designed to retrieve information about a physician's email and phone number. It queries the database to fetch the email of the physician and phone number.

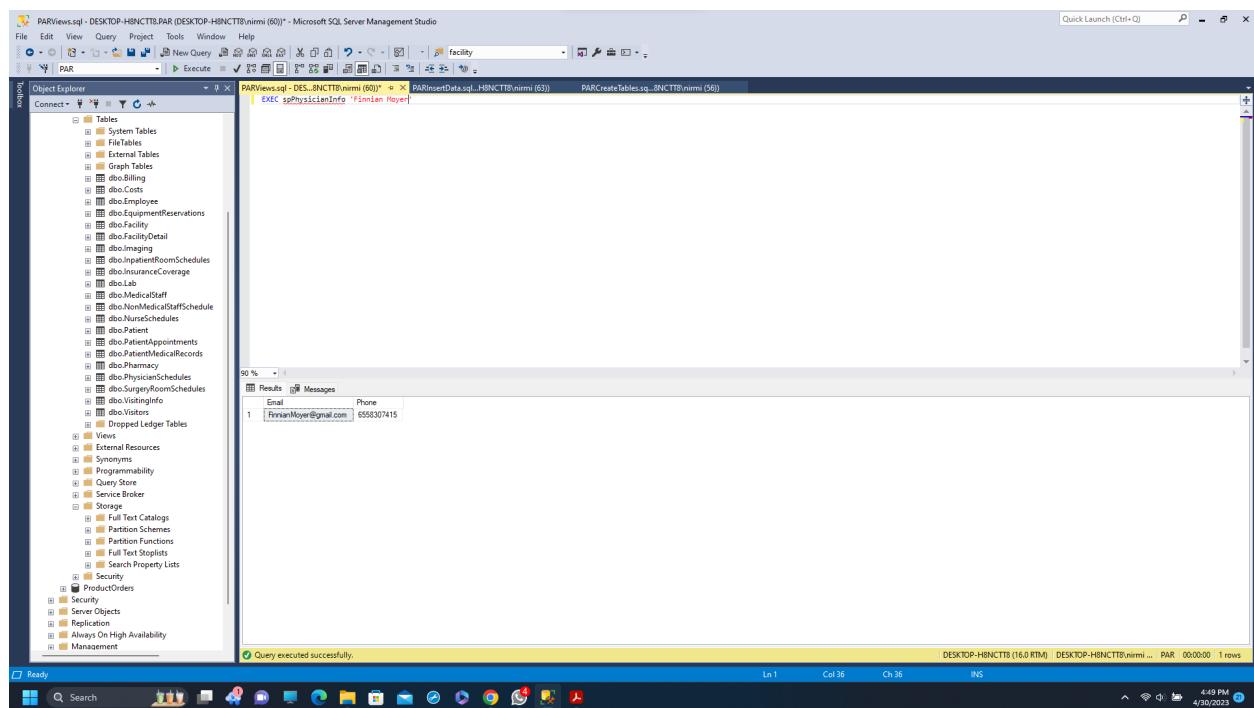
```
CREATE PROC spPhysicianInfo
@name VARCHAR(50)
AS
SELECT Email, Phone
FROM Employee JOIN PhysicianSchedules ON Employee.EmployeeID =
PhysicianSchedules.EmployeeID
WHERE Employee.Name = @name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. The central pane displays the script for the stored procedure `spPhysicianInfo`. The bottom pane shows the results of executing the query, which returns one row with Email and Phone columns. The status bar at the bottom indicates the query was executed successfully.

Email	Phone
FinnianMoyer@gmail.com	6558307415

Query executed successfully.

Testing Stored Procedure: EXEC spPhysicianInfo 'Finnian Moyer'



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to 'DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmit) (60)'.

In the Object Explorer on the left, the database 'PAR' is selected, showing various tables like 'System Tables', 'External Tables', 'Graph Tables', and numerous 'dbo.' prefixed tables such as 'Billing', 'Costs', 'Employee', etc.

The central pane displays a query window with the following command:

```
EXEC spPhysicianInfo 'Finnian Moyer'
```

The results pane shows the output of the query:

Email	Phone
FinnianMoyer@gmail.com	6558307415

A status bar at the bottom right shows the session details: DESKTOP-H8NCTTB (16.0 RTM) | DESKTOP-H8NCTTB\Nirmit | PAR | 00:00:00 | 1 rows.

4. This stored procedure is designed to retrieve information about non medical staff member's email and phone number. It queries the database to fetch the email of the medical staff member's and phone number.

```
CREATE PROC spNonMedicalInfo
@name VARCHAR(50)
AS
SELECT Email, Phone
FROM Employee JOIN NonMedicalStaffSchedule ON Employee.EmployeeID =
NonMedicalStaffSchedule.EmployeeID
WHERE Employee.Name = @name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. In the center, the script pane displays the T-SQL code for creating the stored procedure:

```
--CREATE PROC spNonMedicalInfo
@name VARCHAR(50)
AS
--SELECT Email, phone
--FROM Employee JOIN NonMedicalStaffSchedule ON Employee.EmployeeID =
--NonMedicalStaffSchedule.EmployeeID
--WHERE Employee.Name = @name
```

Below the script pane, the Messages window shows the command completed successfully with a completion time of 2023-04-30T17:08:29.4490469+04:00. At the bottom, a status bar indicates the query was executed successfully.

Testing the stored procedure:
EXEC spNonMedicalInfo 'Fleur Sampson'

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "PARViews.sql - DES...H8NCTTB(nirmi (60)) - Microsoft SQL Server Management Studio". The Object Explorer on the left shows a database structure with various tables and objects. In the center, a query window displays the command "EXEC spNonMedicalInfo 'Fleur Sampson'". The results pane below shows a single row of data:

Email	Phone
xidkinison@yahoo.com	2368578461

A status bar at the bottom indicates "Query executed successfully." and "DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB(nirmi ... PAR 00:00:00 | 1 rows". The system tray at the bottom right shows the date and time as "5:11 PM 4/30/2023".

c. User defined functions

1. This user-defined function takes an input parameter of InsuranceID and uses it to find the corresponding PatientID and Insurance Coverage information.

```
CREATE FUNCTION InsurCov
(@InsuranceID INT)
RETURNS INT
BEGIN
    RETURN (SELECT InsuranceID FROM InsuranceCoverage
    WHERE InsuranceID=@InsuranceID)
END;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including tables like 'PAR', 'Facility', 'Patient', etc. The central pane displays the T-SQL code for creating the 'InsurCov' function:

```
CREATE FUNCTION InsurCov
(@InsuranceID INT)
RETURNS INT
BEGIN
    RETURN (SELECT InsuranceID FROM InsuranceCoverage
    WHERE InsuranceID=@InsuranceID)
END;
```

The status bar at the bottom indicates "Query executed successfully." and "0 rows". The bottom right corner shows the system tray with the date and time: "4/30/2023 4:32 PM".

Testing the User Defined Functions:
SELECT PatientID, InsuranceCovages FROM InsuranceCoverage WHERE
InsuranceID = dbo.InsuranceCov(2);

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PAR'. The 'Tables' node under 'PAR' contains many tables, including 'InsuranceCoverage'. A query window titled 'PARViews.sql - DES...8NCTTB.nimi (60)' is open, containing the following SQL code:

```
SELECT PatientID, InsuranceCovages FROM InsuranceCoverage WHERE InsuranceID = dbo.InsuranceCov(2);
```

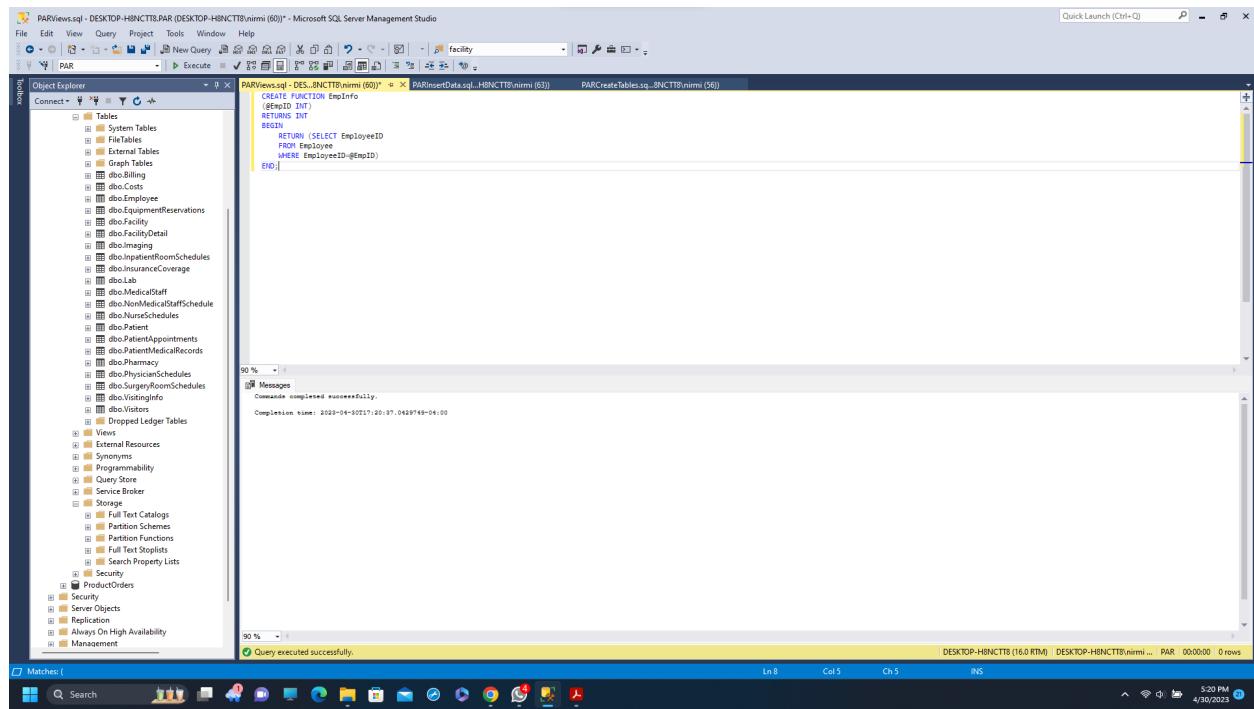
The results pane shows the output of the query:

PatientID	InsuranceCovages
1	Full Coverage
2	Full Coverage

A status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

2. This function is designed to retrieve the information of an employee from the database by inputting the employeeID as a parameter. It will return the employee's name, job title, and department name.

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



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. The central pane displays the T-SQL code for creating the 'EmplInfo' function. The status bar at the bottom indicates the command was completed successfully at 04:50:17 on 04/30/2023.

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

Testing the User Defined Function:
SELECT EmployeeID, Salary, Benefits, ContractType FROM Employee WHERE EmployeeID=dbo.EmpInfo(6);

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'dbo' schema, including tables like 'Employee', 'Facility', and 'Patient', along with other objects like 'Views', 'Functions', and 'Procedures'. The 'PARViews.sql' file is open in the center pane, containing the SQL query: 'SELECT EmployeeID, Salary, Benefits, ContractType FROM Employee WHERE EmployeeID=dbo.EmpInfo(6);'. The results pane below shows a single row of data: EmployeeID 6, Salary 54464, Benefits bonus, and ContractType fixed price. A status bar at the bottom indicates the query was executed successfully.

EmployeeID	Salary	Benefits	ContractType
6	54464	bonus	fixed price

3. This function is designed to retrieve the information of a Patient from the database by inputting the PatientID as a parameter. It will return the employee's name, Phone, and Email.

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

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, there is a tree view of database objects including Tables, Views, External Resources, Synonyms, Programmability, Query Store, Service Broker, Storage, Full Text Catalogs, Partition Schemes, Partition Functions, Full Text Stoplists, Search Property Lists, Security, ProductOrders, Server Objects, Replication, Always On High Availability, and Management. The central pane displays the T-SQL code for creating the function:

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

Below the code, the Messages pane shows the output: "Commands completed successfully." and "Completion time: 2023-04-30T17:53:02.4159791-04:00". At the bottom, the status bar indicates "Query executed successfully.", "Ln 8 Col 5 Ch 5 INS", and the system status "DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB.nirmi ... PAR 00:00:00 0 rows". The taskbar at the bottom shows various application icons.

Testing User Defined Function:
SELECT PatientID, Name, Email FROM Patient WHERE
PatientID=dbo.PatientInfo(5);

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'PAR' database, including tables like 'Patient', 'PatientInfo', and 'PatientReservations'. The 'PARViews.sql' file is open in the center pane, containing the query:

```
SELECT PatientID, Name, Email FROM Patient WHERE PatientID=dbo.PatientInfo(5);
```

The results pane below shows a single row of data:

PatientID	Name	Email
5	Robin Valdez	bpoen@mnn.com

A status bar at the bottom indicates "Query executed successfully." and shows the session details: DESKTOP-H8NCTTB (16.0 RTM) | DESKTOP-H8NCTTB.nimf | PAR | 00:00:00 | 1 rows.

4. This function is designed to retrieve Visitor information from the database.

```
CREATE FUNCTION VisInfo
(@VisID INT)
RETURNS INT
BEGIN
    RETURN (SELECT VisitorsID
    FROM Visitors
    WHERE VisitorsID=@VisID)
END;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, there is a tree view of database objects under the 'PAR' database, including tables like 'System Tables', 'Facility', 'Visitors', and 'VisitingInfo'. The central pane displays the T-SQL code for creating the 'VisInfo' function:

```
CREATE FUNCTION VisInfo
(@VisID INT)
RETURNS INT
BEGIN
    RETURN (SELECT VisitorsID
    FROM Visitors
    WHERE VisitorsID=@VisID)
END;
```

Below the code, the 'Messages' pane shows the command completed successfully with the timestamp 'Completion time: 2023-04-30T17:58:19.032144+00:00'. At the bottom of the screen, the Windows taskbar is visible with various icons and the system clock showing '5:58 PM 4/30/2023'.

Testing User Defined Function:

```
SELECT VisitorsID, VisitorsName, IDNumber FROM Visitors WHERE  
VisitorsID=dbo.VisInfo(1);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'PAR' database, including tables like 'System Tables', 'FileTables', 'External Tables', 'Extended Tables', 'User Tables', 'dbo.Billing', 'dbo.Costs', 'dbo.Employee', 'dbo.EmployeeReservations', 'dbo.Facility', 'dbo.FacilityDetail', 'dbo.Imaging', 'dbo.InpatientRoomSchedules', 'dbo.insuranceCoverage', 'dbo.MedicalStaff', 'dbo.NonMedicalStaffSchedule', 'dbo.NurseSchedules', 'dbo.Patient', 'dbo.PatientAppointments', 'dbo.PatientCallRecords', 'dbo.Pharmacy', 'dbo.PhysicianSchedules', 'dbo.SurgeryRoomSchedules', 'dbo.VisitingInfo', 'dbo.Visitors', and 'Dropped Ledger Tables'. The 'PARViews.sql' file is open in the main window, containing the query: `SELECT VisitorsID, VisitorsName, IDNumber FROM Visitors WHERE VisitorsID=dbo.VisInfo(1);`. The results pane shows a single row:

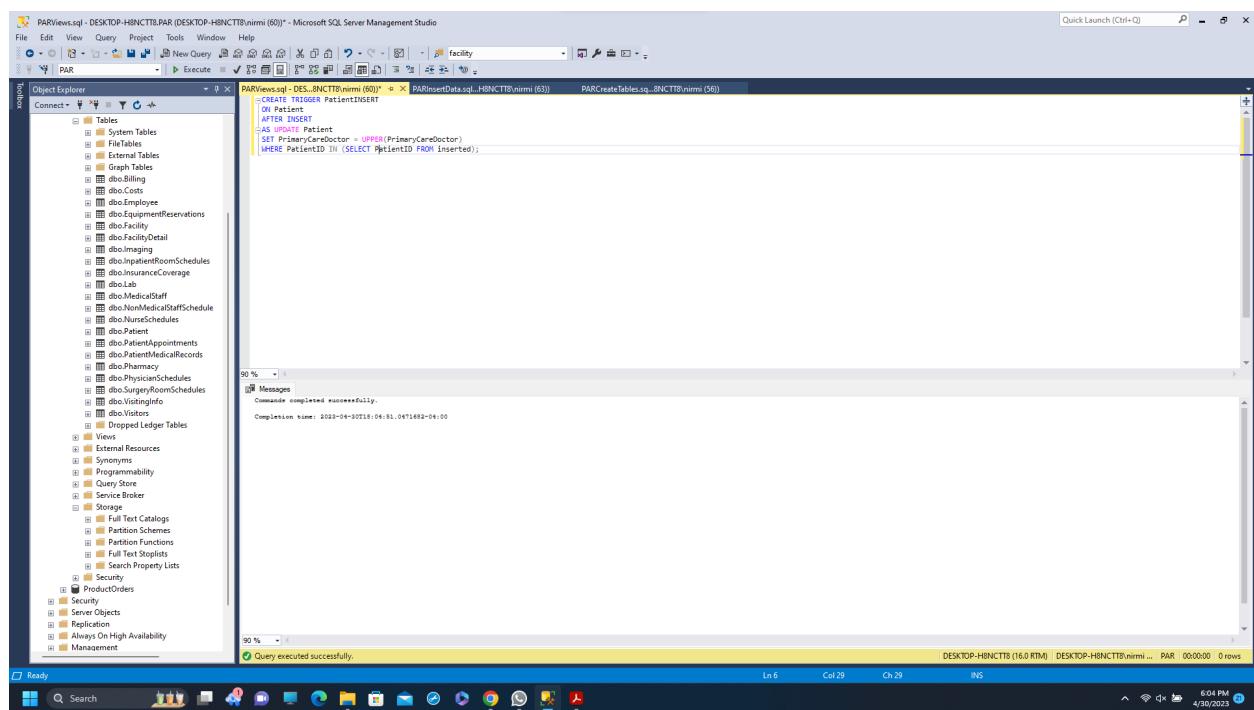
VisitorsID	VisitorsName	IDNumber
1	Mario Baker	237473941

. A status bar at the bottom indicates "Query executed successfully." and "1 rows".

d. Triggers

1. This CREATE TRIGGER statement creates a trigger that activates after a record is inserted into the Patient table. The trigger updates the name of the primary care doctor to be in uppercase mixed case.

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



Testing the Trigger:

INSERT Patient

```
VALUES('Kamal Patel', '150 Westcot street, NY 13244', '992323441',
'kpatel@gmail.com', NULL, 'Jack Ryan');
```

SELECT * FROM Patient

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. The central pane contains a query window with the following content:

```
--INSERT Patient
VALUES('Kamal Patel', '150 Westcot street, NY 13244', '992323441',
'kpatel@gmail.com', NULL, 'Jack Ryan');

SELECT * FROM Patient;
```

The results pane displays the following table:

PatentID	Name	Address	Phone	Email	HealthInsuranceInfo	PrimaryCareDoctor
1	Paulette Walker	965 Havemill Dr. Hamilton, OH 45013	2095514801	yariya@yahoo.com	NULL	Dean Hill
2	Richard Moderate	104 S 5th St. Le Claire, IA 52753	5844635914	metan@gmail.com	NULL	Jake Ellis
3	Bert Lee	11205 Oregon Ave N. Champlin, MN 55316	379599966	frostman@comcast.net	NULL	Stanley Cotez
4	Lloyd Austin	40951 140th St. Grand Island, SD 57446	2838493114	keuter@vive.com	NULL	Ken Carpenter
5	Roger Valdez	1200 W 10th Street, Oklahoma City, OK 73101	8005551212	bjones@bjonesglobal.net	NULL	Marcia Sims
6	Greg Carter	11118 Zimmerman Rd. Mayfield, IN 47141	1750995771	gcarter@bjcglobal.net	NULL	Alexander Pratt
7	Ervin Hansen	202 N 3rd St. Waukesha, WI 45795	1177952012	phabdy@osfnet.net	NULL	Danielle Gilbert
8	Eula Carson	1763 3rd St. Cuyahoga Falls, OH 44221	6303896245	cosmo@att.net	NULL	Tracy Morris
9	Christie Moody	5665 Reindeer Pt. Paso Robles, CA 93446	8935940099	hling@hotmail.com	NULL	Paul Stewart
10	Felicia Osborne	28500 County Rd. Orlando, OK 73073	3755204247	adillon@gmail.com	NULL	Claude Perkins
11	Kamal Patel	150 Westcot street, NY 13244	992323441	kpatel@gmail.com	NULL	JACK RYAN

Below the results, a message indicates "Query executed successfully." The status bar at the bottom right shows "DESKTOP-HBNCTTB (16.0 RTM) | DESKTOP-HBNCTTB(nirmi... | PAR | 00:00:00 | 11 rows".

2. This Trigger statement will not let user alter the database

```
CREATE TRIGGER DATA  
ON DATABASE  
FOR ALTER_TABLE, DROP_TABLE  
AS PRINT'Forbid To Change Database!  
ROLLBACK
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "PARViews.sql - DESKTOP-H8NCTTB\unirmi (60) - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, displaying a tree view of database objects including Tables, Views, External Resources, and Security. The right pane contains a query window with the following content:

```
--:CREATE TRIGGER DATA  
FOR DATABASE  
ON DROPTABLE  
AS  
ALTER 'ParView' TO Change Database!  
ROLLBACK
```

Below the query window, the status bar indicates "Query executed successfully." and "Completion time: 2023-04-30T18:09:49.7260018-04:00". The bottom right corner of the status bar shows the system name "DESKTOP-H8NCTTB (16.0 RTM)", the database name "PAR", the transaction count "000000", and the row count "0 rows".

3. This CREATE TRIGGER statement creates a trigger that activates after a record is inserted into the Employee table. The trigger updates the name of the Employee to be in lower mixed case.

```
CREATE TRIGGER EmployeeInsert
ON Employee
AFTER INSERT
AS UPDATE Employee
SET Name = LOWER(Name)
WHERE EmployeeID IN (SELECT EmployeeID FROM inserted);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'Tables' node, there is a new trigger named 'EmployeeInsert'. The script pane displays the T-SQL code for creating this trigger:

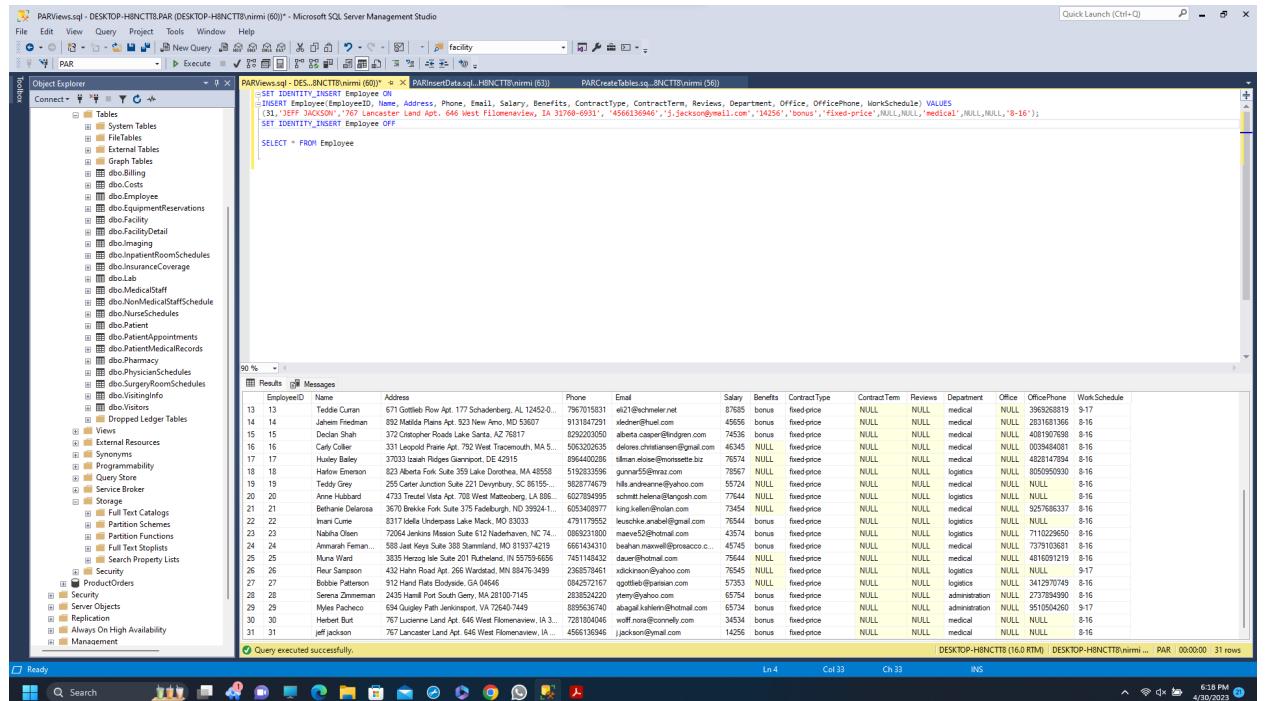
```
CREATE TRIGGER EmployeeInsert
ON Employee
AFTER INSERT
AS UPDATE Employee
SET Name = LOWER(Name)
WHERE EmployeeID IN (SELECT EmployeeID FROM inserted);
```

The status bar at the bottom indicates 'Query executed successfully.' and shows the execution time as 00:00:00 and 0 rows affected.

Testing the Trigger:

```
SET IDENTITY_INSERT Employee ON
INSERT Employee(EmployeeID, Name, Address, Phone, Email, Salary, Benefits,
ContractType, ContractTerm, Reviews, Department, Office, OfficePhone,
WorkSchedule) VALUES
(31,'JEFF JACKSON','767 Lancaster Land Apt. 646 West Filomenaview, IA
31760-6931',
'4566136946','j.jackson@ymail.com','14256','bonus','fixed-price',NULL,NULL,'me
dical',NULL,NULL,'8-16');
SET IDENTITY_INSERT Employee OFF
```

```
SELECT * FROM Employee
```



4. This CREATE TRIGGER statement creates a trigger that activates after a record is updated in the Visitors table. The trigger updates the name of the visitor to be in upper mixed case.

```
CREATE TRIGGER VisitorUpdate
ON Visitors
AFTER UPDATE
AS UPDATE Visitors
SET VisitorsName = UPPER(VisitorsName)
WHERE VisitorsID IN (SELECT VisitorsID FROM inserted);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, there is a tree view of database objects under the 'Tables' node. In the center pane, a query window displays the following T-SQL code:

```
=CREATE TRIGGER VisitorUpdate
ON [Visitors]
AFTER UPDATE
AS UPDATE Visitors
SET VisitorsName = UPPER(VisitorsName)
WHERE VisitorsID IN (SELECT VisitorsID FROM inserted);
```

Below the code, the 'Messages' pane shows the output: "Command completed successfully." At the bottom of the screen, the taskbar shows the system date and time as 4/30/2023 6:22 PM.

Testing the Trigger:

UPDATE Visitors

```
SET VisitorsName = 'james bond', IDNumber = '1456854692'
WHERE VisitorsID = 10;
```

SELECT * FROM Visitors

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, there is a tree view of database objects under the 'PAR' database, including tables like 'Facility', 'Visitors', and 'PatientAppointments'. The 'Visitors' table is selected. In the center, the 'PARViews.sql' file is open in the Query Editor, containing the following SQL code:

```
SET VisitorsName = 'james bond', IDNumber = '1456854692'
WHERE VisitorsID = 10;
```

Below the code, the results pane displays the output of the query. A table titled 'Results' shows 10 rows of data from the 'Visitors' table. The columns are: VisitorsID, VisitorName, IDNumber, DateOfBirth, Picture, PatientBeingVisited, EntryTime, and ExitTime. The data includes entries for visitors like Mario Baker, Jenne Pierce, Adam Rodriguez, Angel Ruiz, Lyle Santiago, Rose Jennings, Vidya Gibson, Ned Grant, and Andrew Rodriguez, with the last row being James Bond.

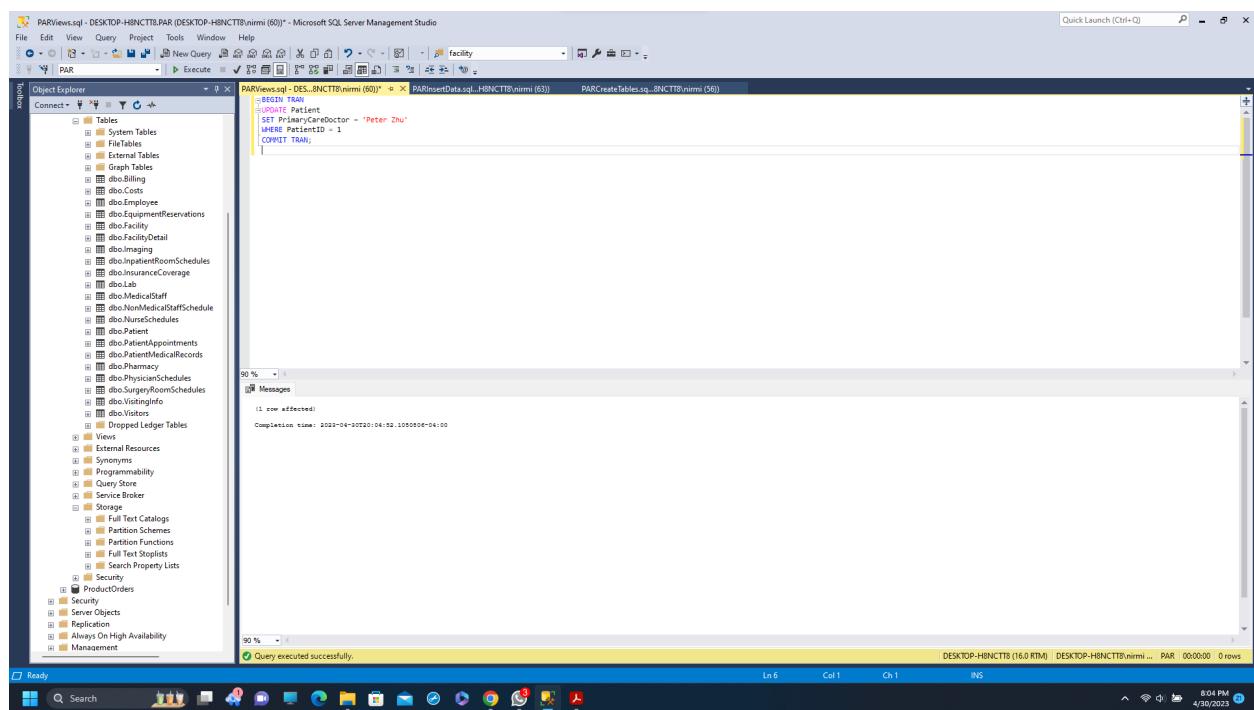
VisitorsID	VisitorName	IDNumber	DateOfBirth	Picture	PatientBeingVisited	EntryTime	ExitTime
1	Mario Baker	2374739417	1989-09-10 00:00:00.000	NULL	Visited	2021-05-02 01:20:03.000	2021-05-02 13:31:44.000
2	Jenne Pierce	5567548402	1976-08-25 00:00:00.000	NULL	Visited	2021-05-01 00:12:04.000	2021-05-01 06:22:41.000
3	Adam Rodriguez	9577178390	1970-10-26 00:00:00.000	NULL	Visited	2021-05-01 08:23:50.000	2021-05-01 12:48:20.000
4	Angel Ruiz	6000000000	1995-07-15 00:00:00.000	NULL	Visited	2021-05-01 08:23:50.000	2021-05-01 12:48:20.000
5	Lyle Santiago	3429284591	1965-07-09 00:00:00.000	NULL	Visited	2021-05-01 07:36:58.000	2021-05-01 09:29:24.000
6	Rose Jennings	6071121613	1963-08-03 00:00:00.000	NULL	Visited	2021-05-01 04:49:53.000	2021-05-01 06:53:42.000
7	Vidya Gibson	0535698612	1980-09-25 00:00:00.000	NULL	Visited	2021-05-10 13:42:05.000	2021-05-10 15:33:00.000
8	Ned Grant	1995330502	1985-12-25 00:00:00.000	NULL	Visited	2021-05-14 03:08:23.000	2021-05-16 05:04:33.000
9	Andrew Rodriguez	3460661408	1993-03-21 00:00:00.000	NULL	Visited	2021-05-17 02:34:05.000	2021-05-17 11:42:52.000
10	JAMES BOND	1456854692	1969-05-10 00:00:00.000	NULL	Visited	2021-05-18 01:35:16.000	2021-05-18 09:31:11.000

At the bottom of the screen, the status bar shows 'Query executed successfully.' and other system information. The taskbar at the bottom of the window shows various icons for the operating system.

e. Transection:

- At the hospital, a patient's primary care doctor needs to be changed due to the retirement of their former doctor. To update the primary care doctor for the patient with ID 1, a transaction is executed. A select statement is then used to confirm the update.

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



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'PAR' is selected, displaying various tables like 'Patient', 'Facility', 'Lab', etc. In the center pane, a query window titled 'PARViews.sql - DES...8NCTTB(nimi) (60)' contains the following T-SQL code:

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

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

- Completion time: 2023-04-30T20:04:52.1050000+04:00
- 1 row affected
- Query executed successfully.

The status bar at the bottom indicates: DESKTOP-H8NCTTB (16.0 RTM) | DESKTOP-H8NCTTB(nimi) ... PAR 00:00:00 0 rows. The system tray shows icons for battery, signal, and network.

Testing the Transection:

SELECT PatientID, PrimaryCareDoctor FROM Patient WHERE PatientID = 1;

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'Tables' category, including System Tables, External Tables, Graph Tables, and numerous tables starting with 'dbo.' such as Billing, Costs, Employee, EquipmentReservations, Facility, Imaging, InpatientRoomschedules, InsuranceCoverage, MedicalStaff, NonMedicalStaffSchedule, NurseSchedules, Patient, PatientAppointments, PatientMedicalRecords, PhysicianSchedules, SurgeryRoomschedules, VisitingInfo, and Visitors. It also shows Dropped Ledger Tables, Views, External Resources, Synonyms, Programmability, Query Store, Service Broker, Security, ProductOrders, and Server Objects.

The central pane displays the results of the following SQL query:

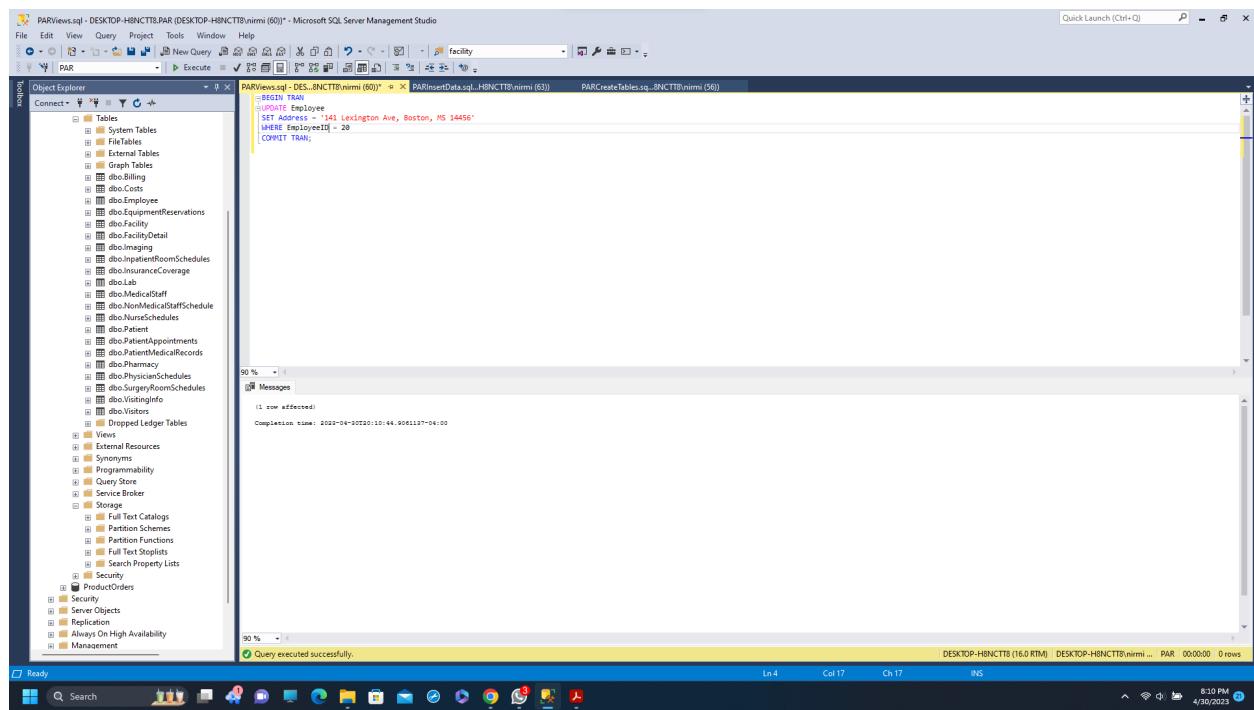
```
SELECT PatientID, PrimaryCareDoctor
FROM Patient
WHERE PatientID = 1;
```

The results grid shows one row with PatientID 1 and PrimaryCareDoctor Peter Zhu.

At the bottom of the screen, the Windows taskbar is visible with icons for Start, Search, Task View, File Explorer, Edge, Mail, Photos, Google Chrome, File Explorer, Task View, and File Explorer. The system tray shows the date and time as 8:05 PM 4/30/2023.

2. At the hospital, an employee's address needs to be changed due to them moving. To update the Address for the employee with ID 20, a transaction is executed. A select statement is then used to confirm the update.

```
BEGIN TRAN
UPDATE Employee
SET Address = '141 Lexington Ave, Boston, MS 14456'
WHERE EmployeeID = 20
COMMIT TRAN;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'PAR'. The central pane shows the results of the following T-SQL script:

```
BEGIN TRAN
UPDATE Employee
SET Address = '141 Lexington Ave, Boston, MS 14456'
WHERE EmployeeID = 20
COMMIT TRAN;
```

The status bar at the bottom right indicates the command was completed successfully with 1 row affected, at a completion time of 2023-04-30T20:10:44.9081117+04:00.

Testing the Transaction:

`SELECT * FROM Employee WHERE EmployeeID = 20`

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. The central pane displays the results of a query:

```
SELECT * FROM Employee WHERE EmployeeID = 20
```

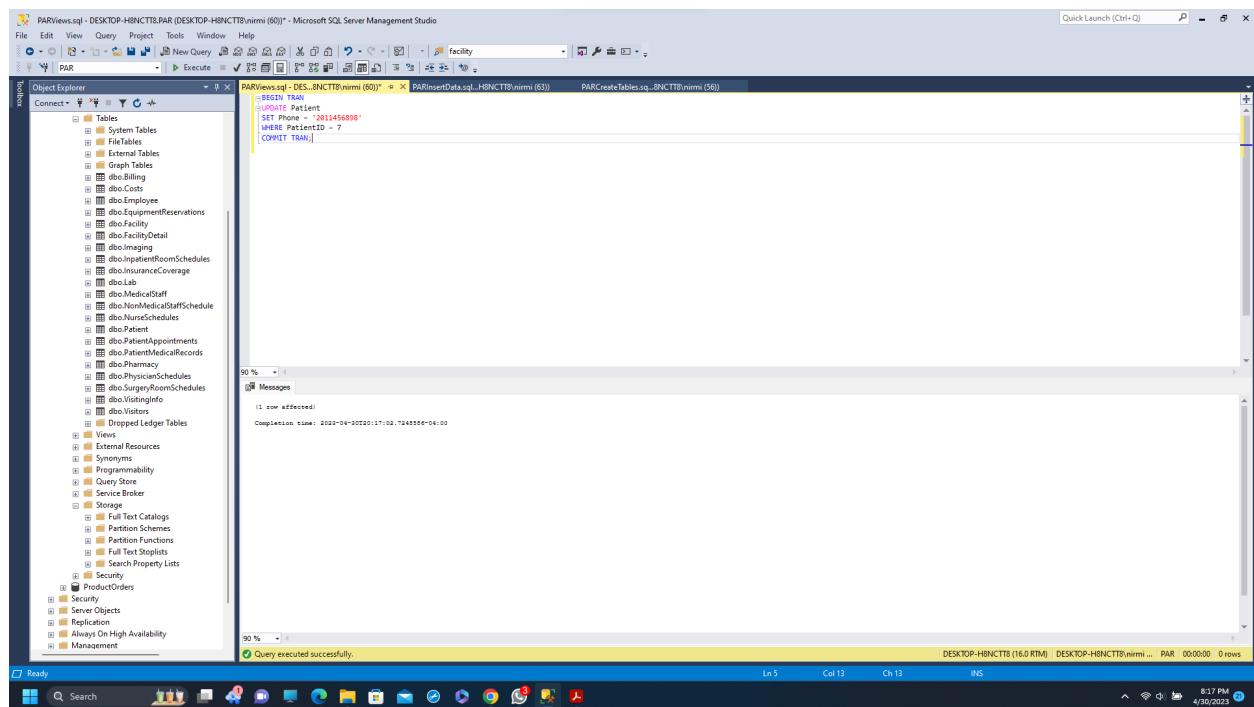
The results grid shows one row of data:

EmployeeID	Name	Address	Phone	Email	Salary	Benefits	ContractType	ContractTerm	Reviews	Department	Office	OfficePhone	Work.Schedule
20	Anne Hubbard	141 Lexington Ave, Boston, MS 14456	6027694955	schmitt.helena@langosta.com	77644	NULL	fixed-price	NULL	NULL	logistics	NULL	8-16	

At the bottom, a message bar indicates "Query executed successfully." and shows statistics: Ln 1, Col 45, Ch 45, INS.

3. At the hospital, a patient's phone number needs to be changed due to them changing it. To update the phone number for the patient with ID 7, a transaction is executed. A select statement is then used to confirm the update.

```
BEGIN TRAN
UPDATE Patient
SET Phone = '2011456898'
WHERE PatientID = 7
COMMIT TRAN;
```



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, there is a tree view of database objects under the 'Tables' category. In the center pane, a query window displays the following T-SQL code:

```
BEGIN TRAN
UPDATE Patient
SET Phone = '2011456898'
WHERE PatientID = 7
COMMIT TRAN;
```

Below the code, the results pane shows a message: '(1 row affected)'. At the bottom of the results pane, it says 'Completion time: 2023-04-30T20:17:02.7245556-04:00'. The status bar at the bottom right indicates 'Query executed successfully.' The system tray at the bottom shows various icons, and the taskbar shows the date and time as 4/30/2023 8:17 PM.

Testing the Transaction:

SELECT * FROM Patient WHERE PatientID = 7

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'Tables' category. The central pane displays the results of a query: 'SELECT * FROM Patient WHERE PatientID = 7'. The result set contains one row of data:

PatientID	Name	Address	Phone	Email	HealthInsuranceInfo	PrimaryCareDoctor
7	Evin Harrison	202 N 2nd St, Wolcottville, IN 46795	2011456898	pdbaby@optonline.net	NULL	Danielle Gibert

A status bar at the bottom indicates 'Query executed successfully.' and shows the session details: DESKTOP-H8NCTTB (16.0 RTM) | DESKTOP-H8NCTTB\Nirmit | PAR | 00:00:00 | 1 rows.

4. At the hospital, a patient's Email needs to be changed due to them changing it. To update the email for the patient with ID 9, a transaction is executed. A select statement is then used to confirm the update.

```
BEGIN TRAN
UPDATE Patient
SET Email = 'abs.123@xyz.com'
WHERE PatientID = 9
COMMIT TRAN;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like Tables, Views, and Procedures. The central pane contains a query window with the following SQL code:

```
BEGIN TRAN
UPDATE Patient
SET Email = 'abs.123@xyz.com'
WHERE PatientID = 9
COMMIT TRAN;
```

Below the code, the results pane shows a single row affected:

1 row affected

Completion time: 2023-04-30T20:22:51.1250000+04:00

At the bottom right, a message indicates the query was executed successfully:

Query executed successfully.

Testing the transection :

`SELECT * FROM Patient WHERE PatientID = 9;`

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects under the 'Tables' category. The 'PARViews.sql' file is open in the center pane, containing the query: `SELECT * FROM Patient WHERE PatientID = 9;`. The results pane below shows a single row of data:

PatientID	Name	Address	Phone	Email	HealthInsuranceInfo	PrimaryCareDoctor
9	Christie Moody	5665 Reindeer Pt, Paso Robles, CA 93446	893594099	abc.123@xyz.com	NULL	Raul Stewart

A status bar at the bottom indicates "Query executed successfully." and "1 rows".

f. Script

- This script is responsible for creating a new user-defined database role called "PatientEntry" in the database. After that, it grants the UPDATE permission for the Patient table to the newly created role. Furthermore, the script creates a login ID with the name "JOY1999" and the password "123456", and sets the default database for this login to be the PAR database. Finally, it creates a user called "Nirmit" for the login and assigns this user to the PatientEntry role.

```

CREATE ROLE PatientEntry;
GRANT UPDATE
ON Patient
TO PatientEntry;
CREATE LOGIN JOY1999
WITH PASSWORD ='123456',
DEFAULT_DATABASE = PAR;
CREATE USER Nirmit FOR LOGIN JOY1999;
ALTER ROLE PatientEntry ADD MEMBER Nirmit;

```

```

PARViews.sql - DES...NCTB(nirmi (60)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
File Edit View Query Project Tools Window Help
PARViews.sql - DES...NCTB(nirmi (60)) => PARInsertData.sql - H8NCTB(nirmi (63)) PARCreateTables.sql - H8NCTB(nirmi (56))
CREATE ROLE PatientEntry;
GRANT UPDATE
ON Patient
TO PatientEntry;
CREATE LOGIN JOY1999
WITH PASSWORD ='123456',
DEFAULT_DATABASE = PAR;
CREATE USER Nirmit FOR LOGIN JOY1999;
ALTER ROLE PatientEntry ADD MEMBER Nirmit;

90 % ↓
Messages
Commands completed successfully.
Completion time: 2023-04-30T20:36:49.814347+04:00
90 % ↓
Query executed successfully.

Ln 9 Col 49 Ch 43 INS
DESKTOP-H8NCTB (16.0 RTM) | DESKTOP-H8NCTB(nirmi ... | PAR | 0 rows
Ready
8:36 PM
4/30/2023

```

2. This script is responsible for creating a new user-defined database role called "EmployeeUpdate" in the database. After that, it grants the UPDATE permission for the Employee table to the newly created role. Furthermore, the script creates a login ID with the name "anu218" and the password "abcd", and sets the default database for this login to be the PAR database. Finally, it creates a user called "Anu" for the login and assigns this user to the PatientEntry role.

```

CREATE ROLE EmployeeUpdate;
GRANT UPDATE
ON Employee
TO EmployeeUpdate;
CREATE LOGIN Anu218
WITH PASSWORD ='abcd',
DEFAULT_DATABASE = PAR;
CREATE USER Anu FOR LOGIN Anu218;
ALTER ROLE EmployeeUpdate ADD MEMBER Anu;

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure, including databases, tables, and stored procedures. The Results pane on the right shows the output of a SQL script. The script creates a role 'EmployeeUpdate', grants update permissions on the 'Employee' table to this role, creates a login 'Anu218' with password 'abcd', sets the default database to 'PAR', creates a user 'Anu' for this login, and adds 'Anu' to the 'EmployeeUpdate' role. The results pane indicates the command completed successfully at 00:00:00 on 04/30/2023.

```

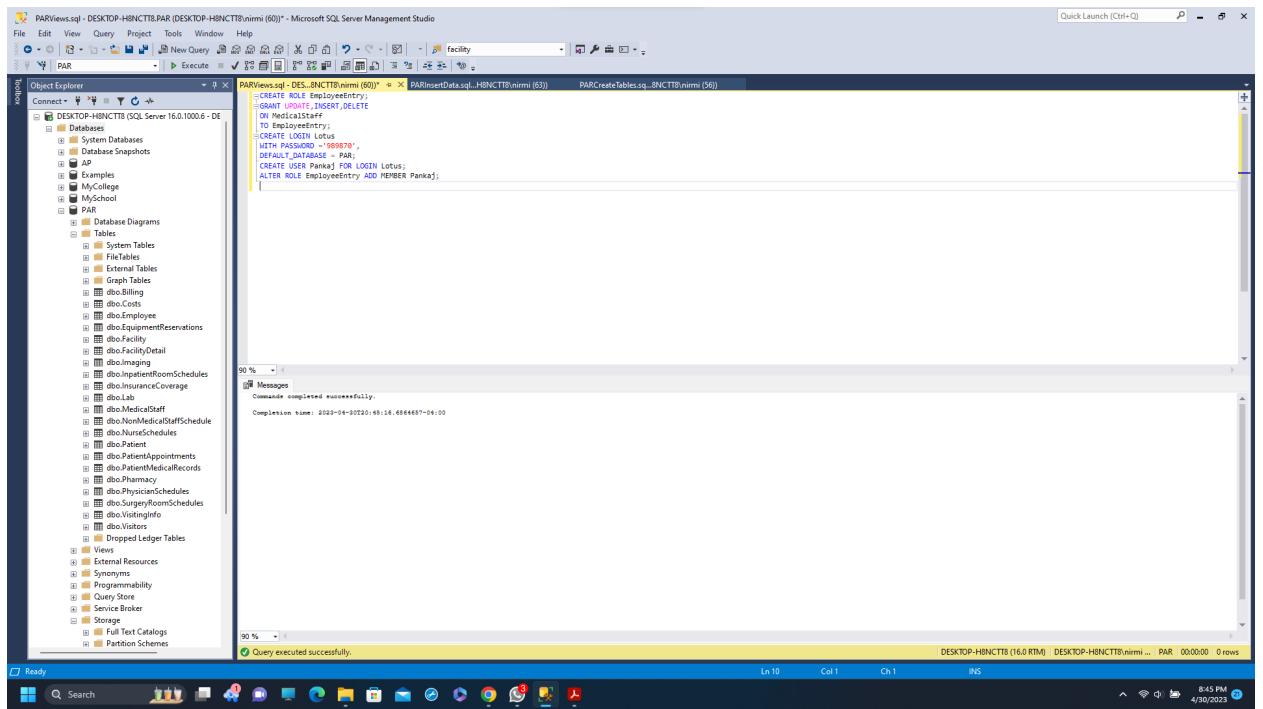
CREATE ROLE EmployeeUpdate;
GRANT UPDATE
ON Employee
TO EmployeeUpdate;
CREATE LOGIN Anu218
WITH PASSWORD ='abcd',
DEFAULT_DATABASE = PAR;
CREATE USER Anu FOR LOGIN Anu218;
ALTER ROLE EmployeeUpdate ADD MEMBER Anu;

(0 rows affected)
Command completed successfully.
Completion time: 2023-04-30T20:41:59.7651288+04:00

```

3. This script is similar to the above script. But with different security levels, passwords, and roles.

```
CREATE ROLE EmployeeEntry;
GRANT UPDATE,INSERT,DELETE
ON MedicalStaff
TO EmployeeEntry;
CREATE LOGIN Lotus
WITH PASSWORD ='989870',
DEFAULT_DATABASE = PAR;
CREATE USER Pankaj FOR LOGIN Lotus;
ALTER ROLE EmployeeEntry ADD MEMBER Pankaj;
```



4. This script is similar to the above script. But with different security levels, passwords, and roles.

```

CREATE ROLE EmployeeEntryA;
GRANT INSERT,DELETE
ON MedicalStaff
TO EmployeeEntryA;
CREATE LOGIN Moon3
WITH PASSWORD ='743301',
DEFAULT_DATABASE = PAR;
CREATE USER ChadrikaPP FOR LOGIN Moon3;
ALTER ROLE EmployeeEntryA ADD MEMBER ChadrikaPP;

```

```

PARViews.sql - DES...NCTB(nirmi (60)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
PAR [PAR] Execute New Query
PARViews.sql - DES...NCTB(nirmi (60)) PARInsertData.sql - H8NCTB(nirmi (63)) PARCreateTables.sql - H8NCTB(nirmi (56))
PARViews.sql - DES...NCTB(nirmi (60)) PARInsertData.sql - H8NCTB(nirmi (63)) PARCreateTables.sql - H8NCTB(nirmi (56))

Object Explorer
PAR [PAR]
  DESKTOP-H8NCTB (SQL Server 16.0.1000.6 - DE
    Databases
      System Databases
      Database Snapshots
      Examples
      MyCollege
      MySchool
    PAR
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        Partitioned Tables
        Index Cost
        Employee
        EquipmentReservations
        Facility
        FacilityDetail
        FacilityGroup
        InsuranceCoverage
        Lab
        MedicalStaff
        NonMedicalStaffSchedule
        NonMedicalStaffSchedules
        Patient
        PatientAppointments
        PatientMedicalRecords
        Pharmacy
        PhysicianSchedules
        SurgeryRoomSchedules
        Vitals
        Dropped Ledger Tables
      Views
      External Resources
      Synonyms
      Asymmetric
      Query Store
      Service Broker
      Storage
      Text Catalogs
      Partition Schemes
  90 % 90 %
  Ready Search
  Messages
  Command completed successfully.
  Completion time: 2023-04-30T20:52:05.7698841+04:00
  90 % 90 %
  Query executed successfully.
  Ln 9 Col 48 Ch 48 INS
  DESKTOP-H8NCTB (16.0 RTM) | DESKTOP-H8NCTB(nirmi ... | PAR 00:00:00 0 rows
  8:52 PM 4/30/2023

```

E. Conclusion

This project has been highly significant as it has enabled me to develop a comprehensive database from scratch. The process of designing, implementing, testing, and debugging the database has been a valuable learning experience, and I have encountered several challenges that were not present in previous projects. I now have a better understanding of SQL statements and can use them more effectively.

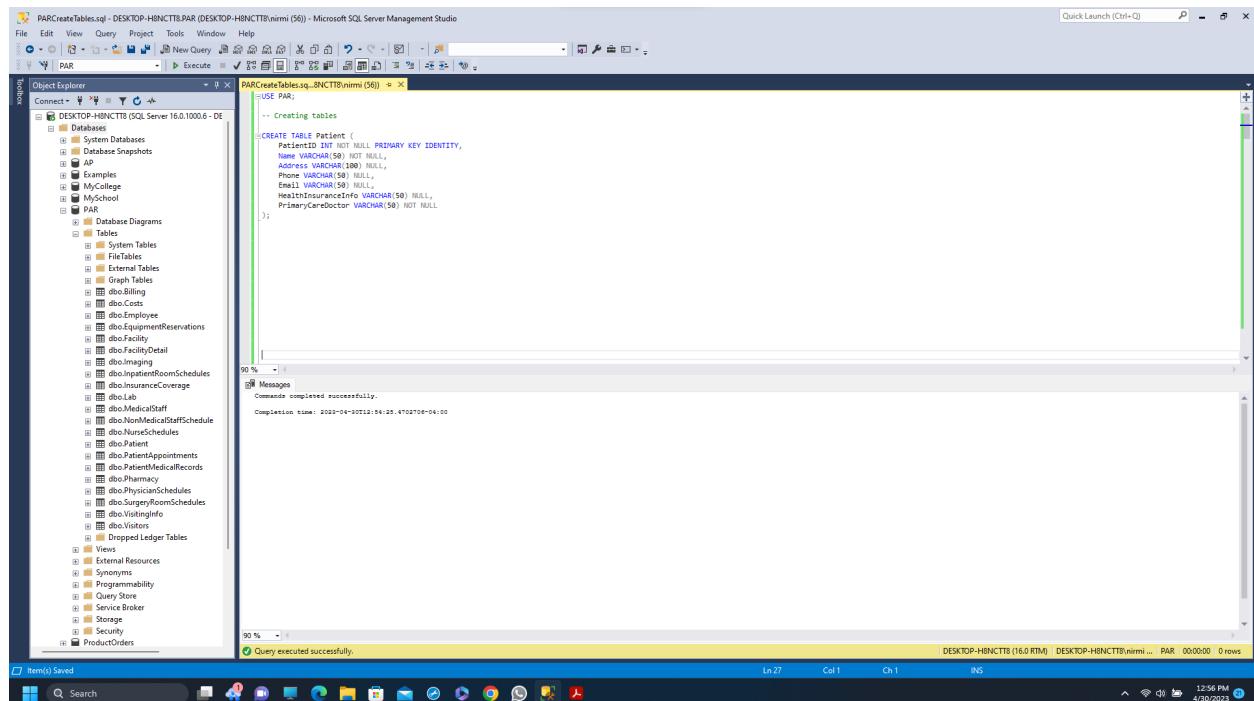
The Microsoft SQL Server Management software has been instrumental in creating a realistic database. It has user-friendly features for retrieving and updating data, and offers reliable backup and security options.

This project has sparked my interest in understanding the relationships between objects. In our daily lives, we often overlook basic relationships, but through this project, I have realized how important it is to efficiently filter and manage them. It has shown me that things are not always as straightforward as we might assume.

F. Appendix

This appendix includes all the screenshots for table creation and data entries:

1. Table Creation:



```

USE PAR;
-- Creating tables
CREATE TABLE Patient
(
    PatientID INT NOT NULL PRIMARY KEY IDENTITY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Address VARCHAR(100) NULL,
    Phone VARCHAR(50) NULL,
    Email NVARCHAR(50) NULL,
    HealthInsuranceInfo VARCHAR(50) NULL,
    PrimaryCareDoctor VARCHAR(50) NOT NULL
);

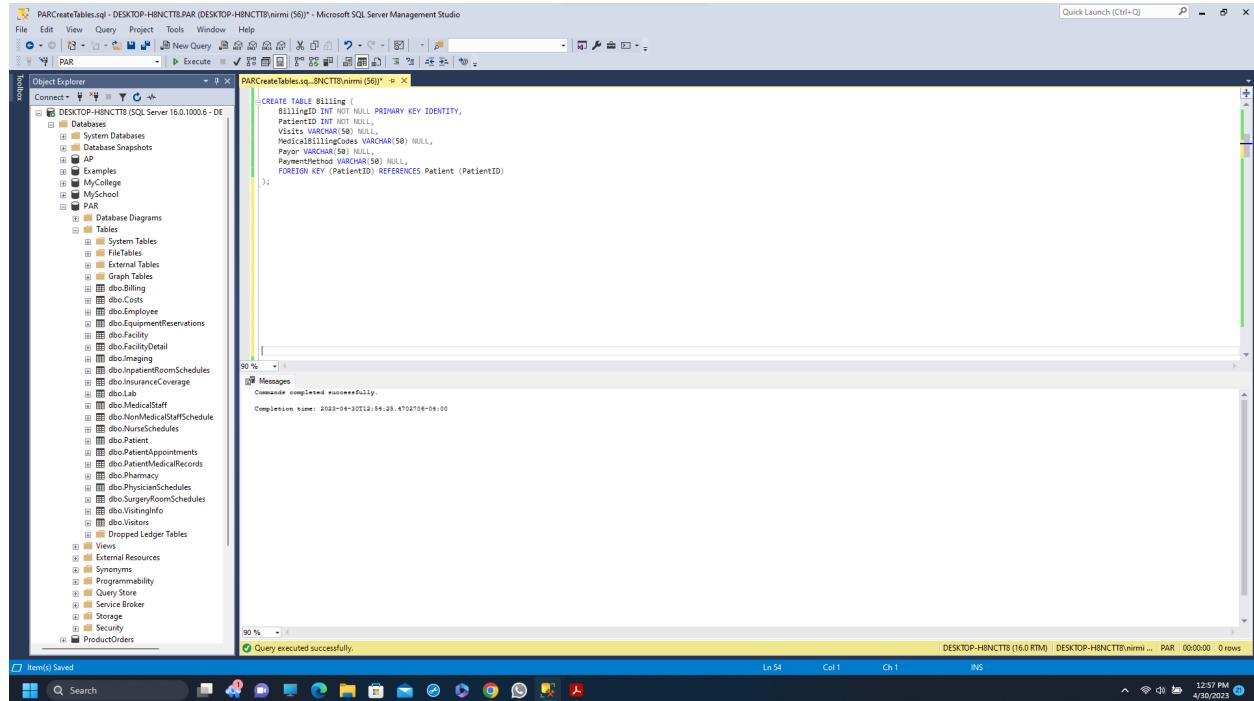
```

Messages

Commands completed successfully.

Completion time: 2023-04-03T12:54:28.4702756+04:00

Query executed successfully.



```

CREATE TABLE BILLING
(
    BillID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    Visits VARCHAR(50) NULL,
    MedicalBillingCode VARCHAR(50) NULL,
    PaymentType VARCHAR(50) NULL,
    PaymentMethod VARCHAR(50) NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

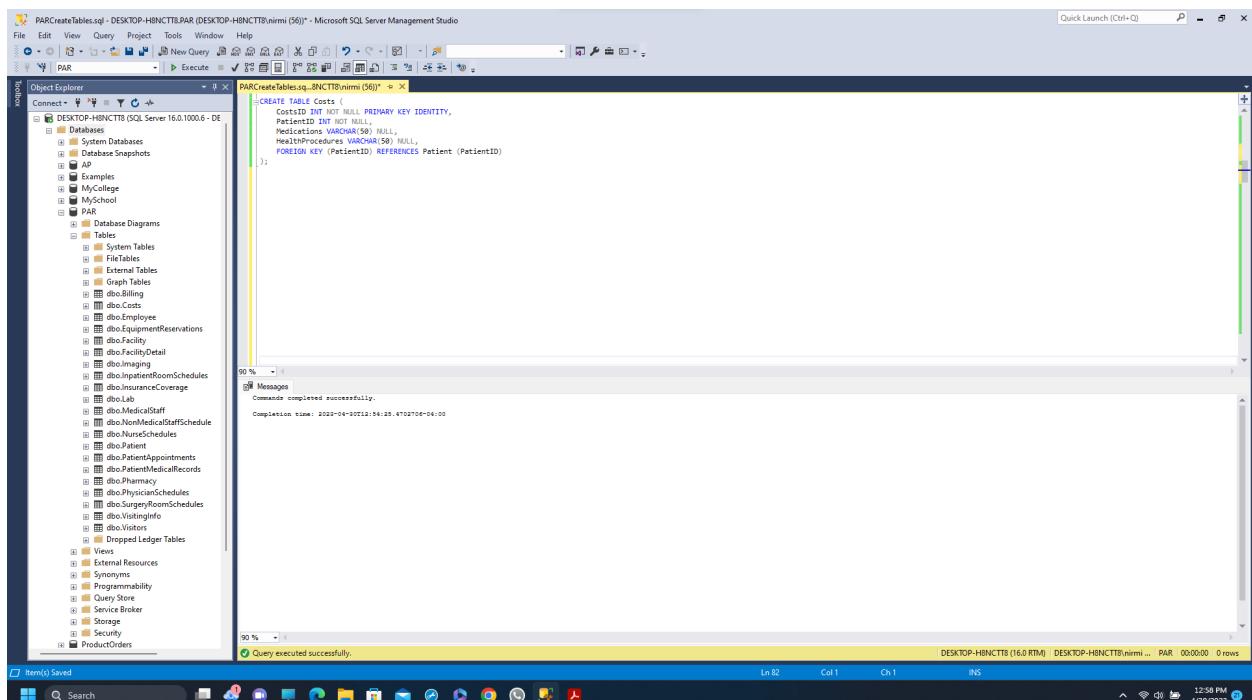
```

Messages

Commands completed successfully.

Completion time: 2023-04-03T12:54:28.4702756+04:00

Query executed successfully.



```

CREATE TABLE Costs (
    CostID INT NOT NULL PRIMARY KEY IDENTITY,
    PatientID INT NOT NULL,
    Medications VARCHAR(50) NULL,
    HealthProcedure VARCHAR(50) NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

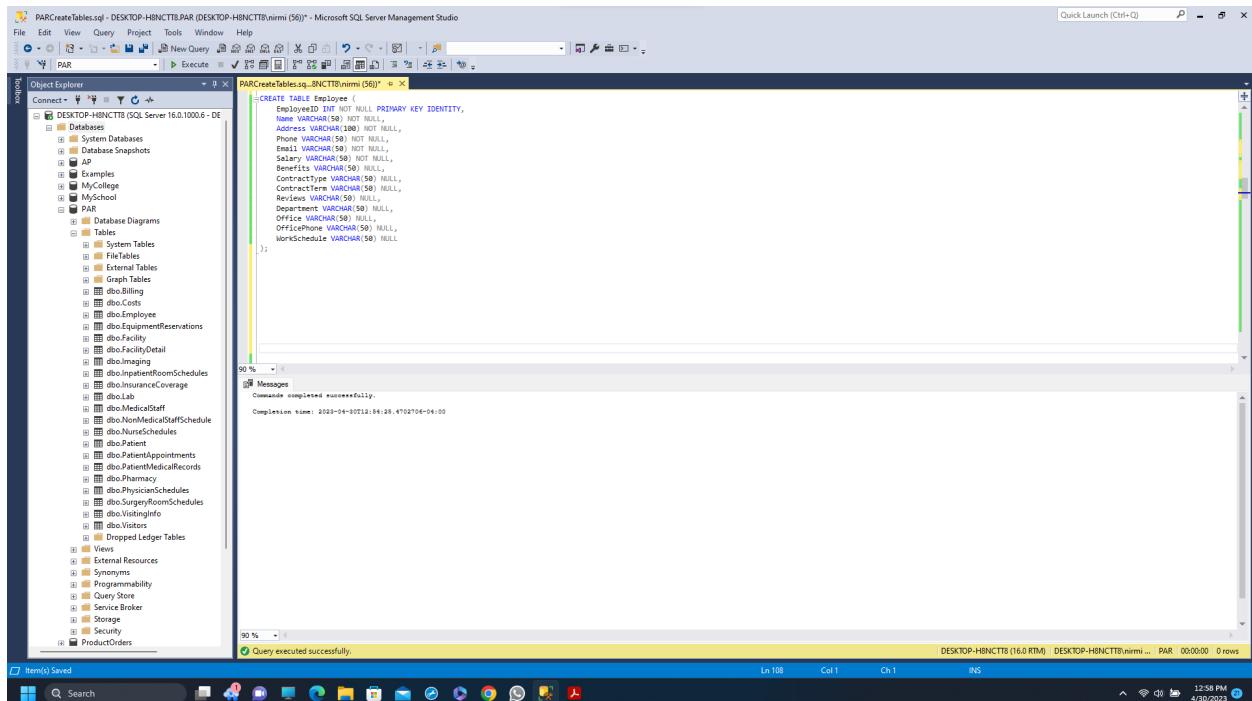
```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.



```

CREATE TABLE Employee (
    EmployeeID INT NOT NULL PRIMARY KEY IDENTITY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    Phone VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    Salary VARCHAR(50) NOT NULL,
    Benefits VARCHAR(50) NULL,
    ContractType VARCHAR(50) NULL,
    Department VARCHAR(50) NULL,
    ManagerID INT NULL,
    Reviews VARCHAR(50) NULL,
    Department VARCHAR(50) NULL,
    SupervisorID INT NULL,
    OfficePhone VARCHAR(50) NULL,
    WorkSchedule VARCHAR(50) NULL
);

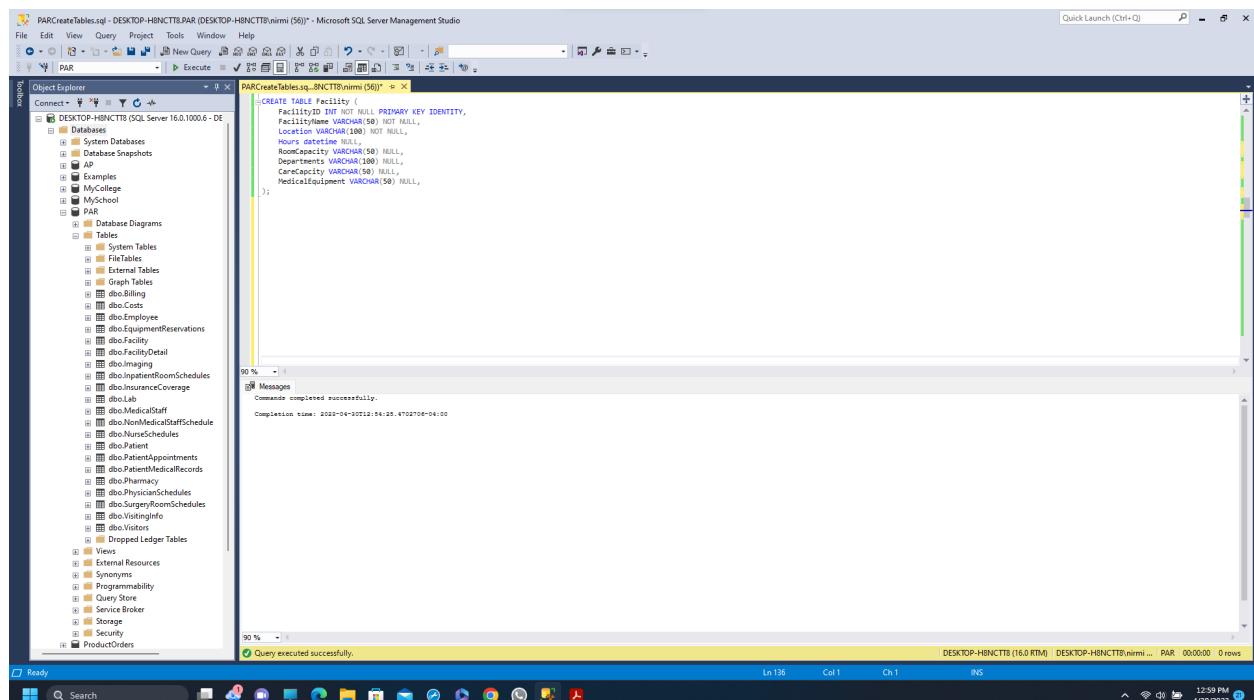
```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.



```

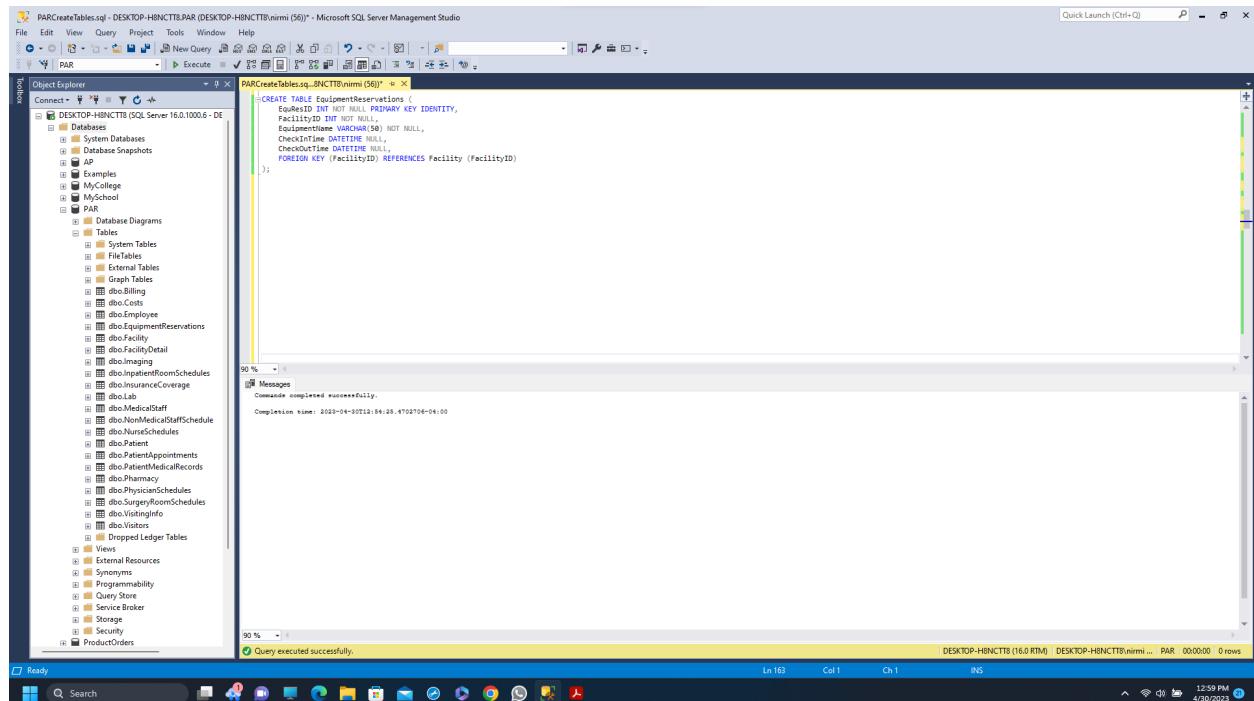
CREATE TABLE Facility (
    FacilityID INT NOT NULL PRIMARY KEY IDENTITY,
    Name VARCHAR(50) NOT NULL,
    Location VARCHAR(100) NOT NULL,
    HourlyRate TIME NULL,
    DailyRate MONEY NULL,
    Department VARCHAR(100) NULL,
    CareCapacity VARCHAR(50) NULL,
    MedicalEquipment VARCHAR(50) NULL,
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00



```

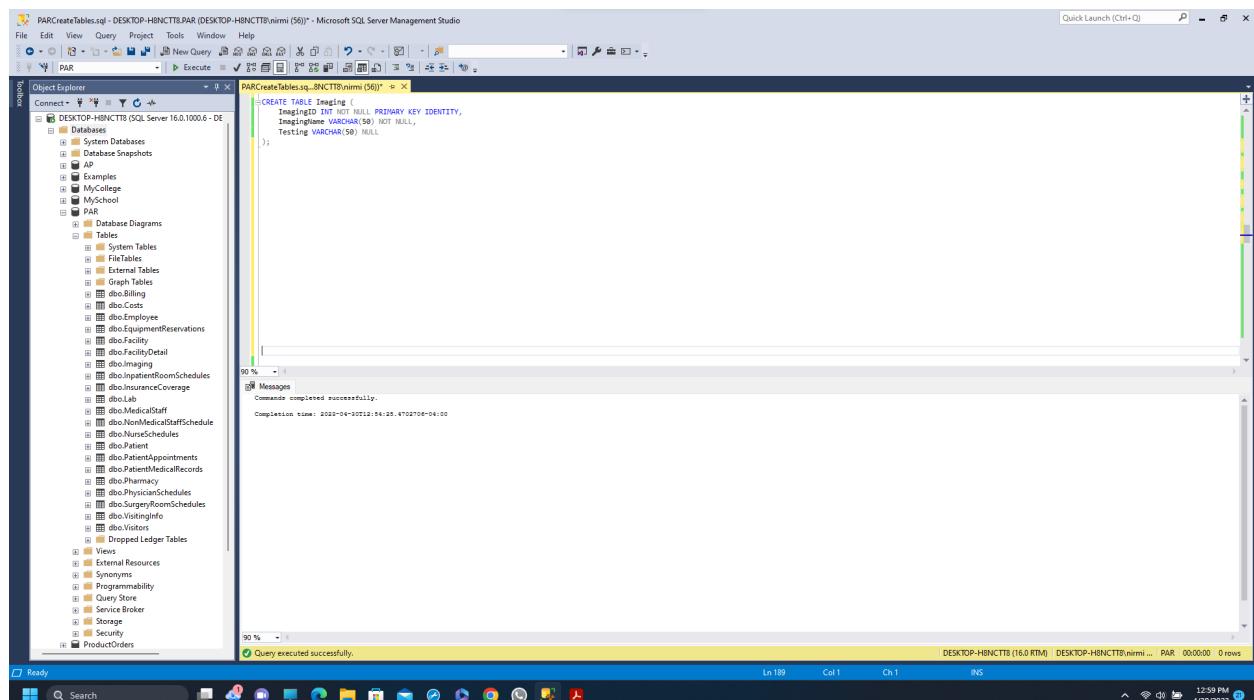
CREATE TABLE EquipmentReservations (
    ReservationID INT NOT NULL PRIMARY KEY IDENTITY,
    FacilityID INT NOT NULL,
    EquipmentID VARCHAR(50) NOT NULL,
    CheckInTime DATETIME NOT NULL,
    CheckOutTime DATETIME NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00



```

CREATE TABLE Invoicing
(
    InvoicingID INT PRIMARY KEY IDENTITY,
    InvoicingDate VARCHAR(50) NOT NULL,
    Testing VARCHAR(50) NULL
);

```

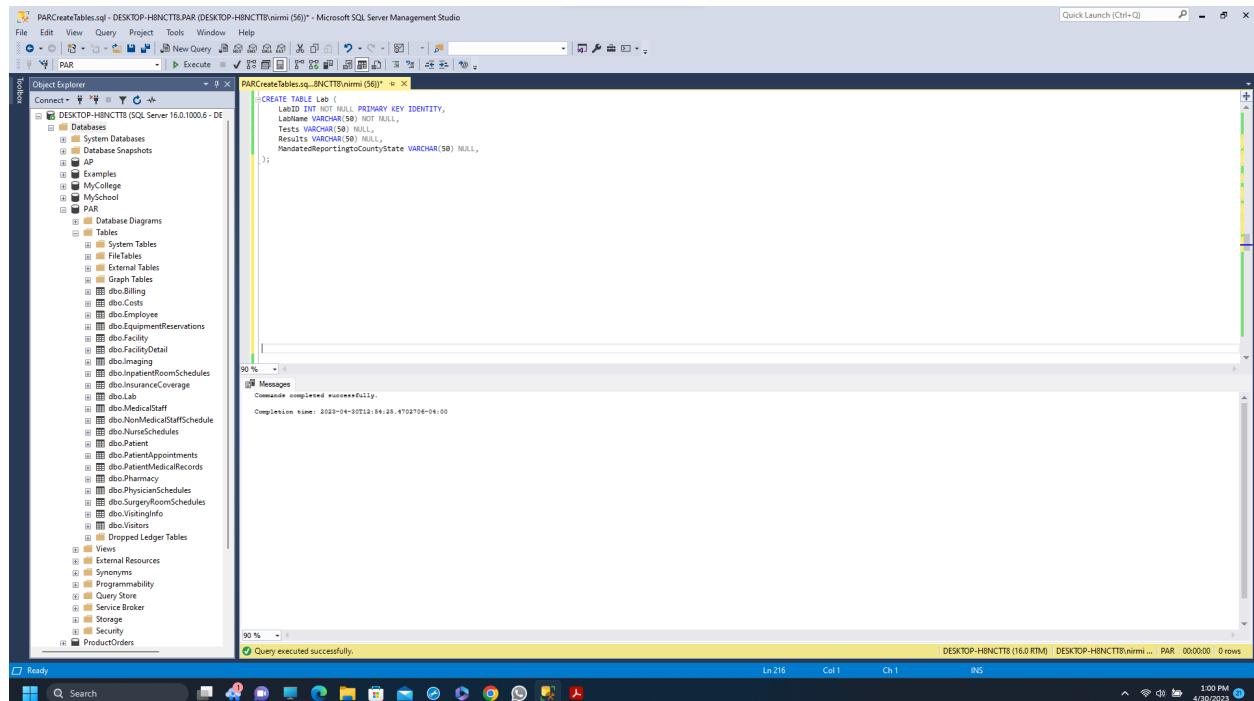
Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

LN 199 Col 1 Ch 1 INS



```

CREATE TABLE Lab
(
    LabID INT PRIMARY KEY IDENTITY,
    LabName VARCHAR(50) NULL,
    Tests VARCHAR(50) NULL,
    Results VARCHAR(50) NULL,
    MandatedReportingToCountyState VARCHAR(50) NULL,
    );

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

LN 216 Col 1 Ch 1 INS

The image shows two side-by-side instances of Microsoft SQL Server Management Studio (SSMS) running on a Windows operating system. Both instances are connected to the same database, 'PAR', on a local server ('DESKTOP-H8NCTTB').

Top Window (Left):

```

CREATE TABLE Pharmacy (
    PharmacyID INT NOT NULL PRIMARY KEY IDENTITY,
    Name VARCHAR(50) NOT NULL,
    Medicine VARCHAR(50) NULL,
    RentalEquipment VARCHAR(50) NULL,
);

```

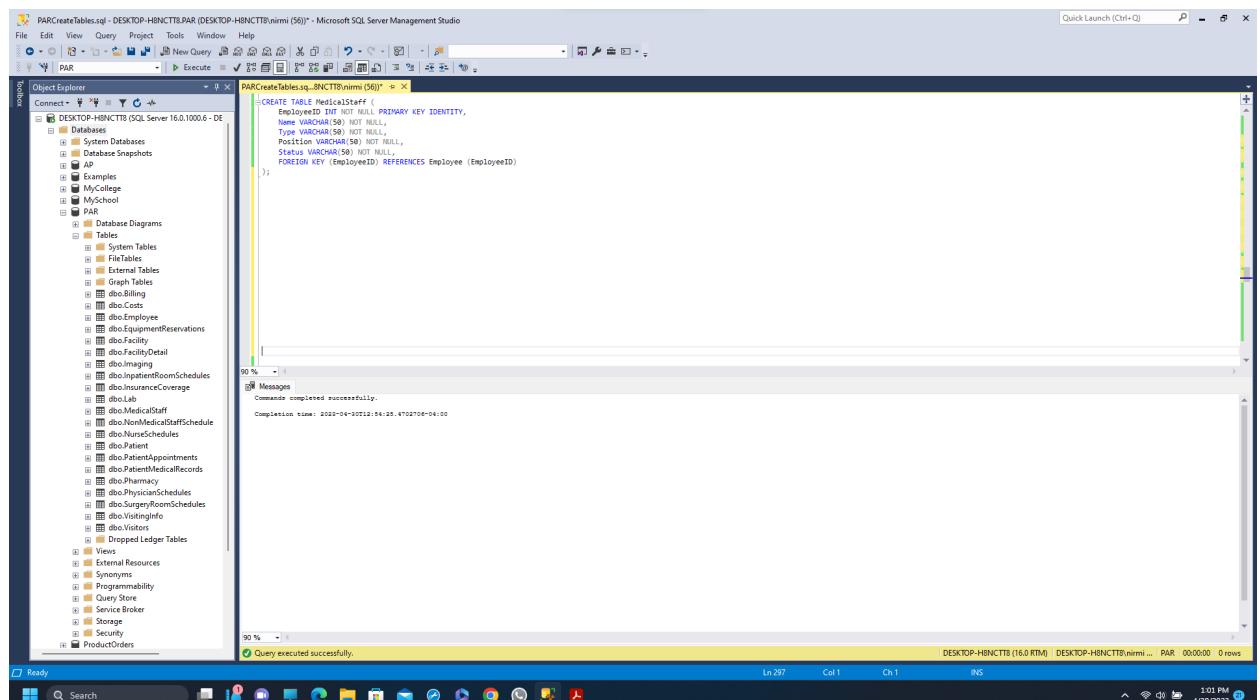
Bottom Window (Right):

```

CREATE TABLE FacilityDetail (
    FacilityDetailID INT NOT NULL PRIMARY KEY IDENTITY,
    FacilityID NOT NULL,
    LabID INT NOT NULL,
    ImagingID INT NOT NULL,
    ImagingDetailID INT NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (ImagingID) REFERENCES Imaging (ImagingID),
    FOREIGN KEY (LabID) REFERENCES Lab (LabID),
    FOREIGN KEY (PharmacyID) REFERENCES Pharmacy (PharmacyID)
);

```

In both windows, the status bar at the bottom indicates the command was completed successfully with a completion time of 2023-04-30T12:54:28.4702756-04:00. The top window shows 0 rows affected, while the bottom window shows 0 rows affected.



```

CREATE TABLE MedicStaff (
    EmployeeID INT NOT NULL PRIMARY KEY IDENTITY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Position VARCHAR(50) NOT NULL,
    Status VARCHAR(50) NOT NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)
);

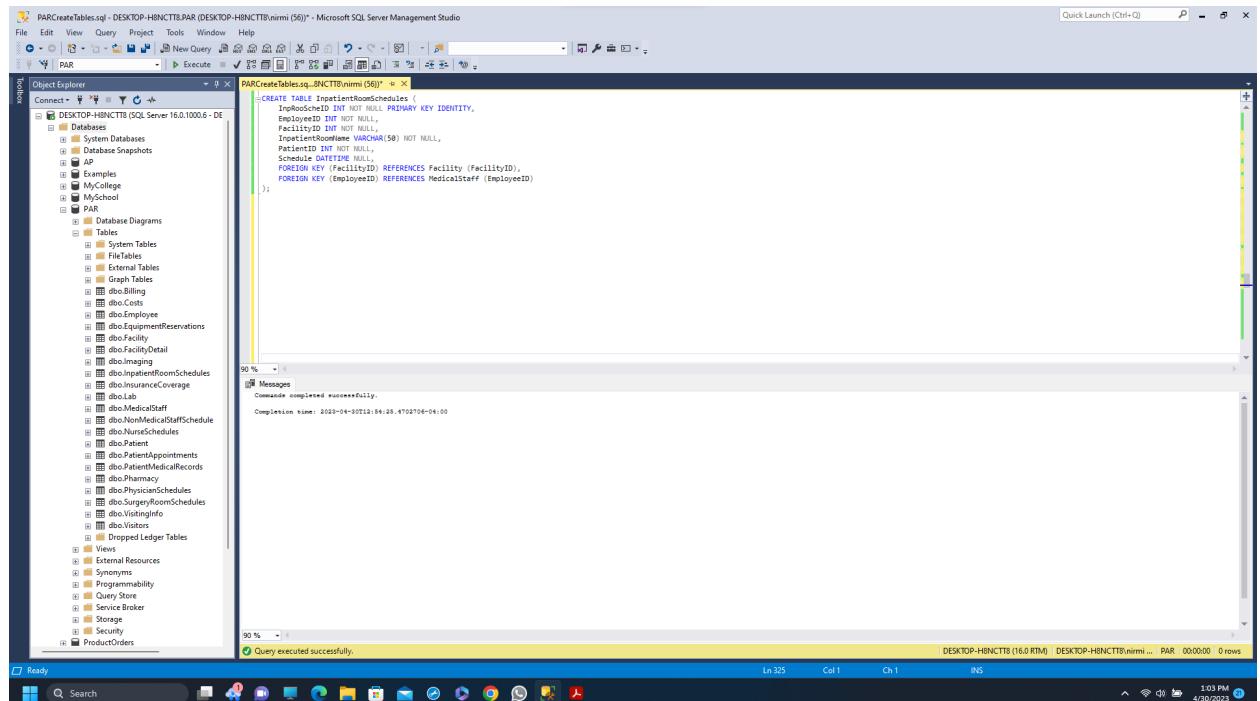
```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.



```

CREATE TABLE InpatientRoomSchedules (
    RoomScheduleID INT NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    InpatientRoomName VARCHAR(50) NOT NULL,
    RoomCapacity INT NOT NULL,
    Schedule DATETIME NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

```

CREATE TABLE InsuranceCoverage (
    InsuredID INT NOT NULL PRIMARY KEY IDENTITY,
    HealthInsuranceCompany VARCHAR(50) NOT NULL,
    Phone VARCHAR(50) NOT NULL,
    InsuranceCov VARCHAR(50) NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

```

CREATE TABLE NonMedicalStaffSchedule (
    EmployeeID INT NOT NULL PRIMARY KEY IDENTITY,
    EmployeeName NVARCHAR(50) NOT NULL,
    FacilityID INT NOT NULL,
    Schedule DATETIME NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

Screenshot 1: Microsoft SQL Server Management Studio (PARCreateTables.sql)

```

CREATE TABLE Nurseschedules (
    NurseScheduleID INT PRIMARY KEY IDENTITY,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    NonMedicalStaffScheduleID INT NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

LN 406 Col 1 Ch 1 INS

Ready

DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\NIRMI ... PAR 00:00:00 0 rows

104 PM 4/30/2023

Screenshot 2: Microsoft SQL Server Management Studio (PARCreateTables.sql)

```

CREATE TABLE PatientAppointments (
    PatientAppointmentID INT PRIMARY KEY IDENTITY,
    EmployeeID INT NOT NULL,
    PatientID INT NOT NULL,
    AppointmentDate DATETIME NOT NULL,
    NonMedicalStaffScheduleID INT NOT NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID),
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

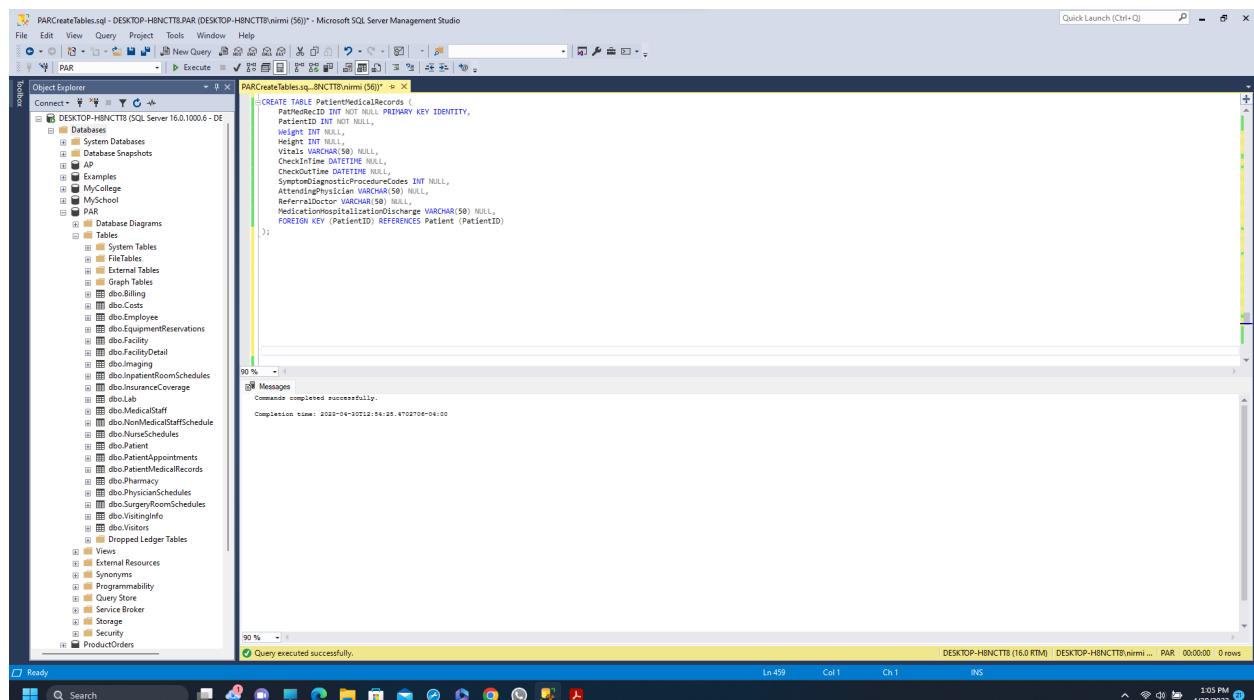
Query executed successfully.

LN 432 Col 1 Ch 1 INS

Ready

DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\NIRMI ... PAR 00:00:00 0 rows

104 PM 4/30/2023



```

CREATE TABLE PatientMedicalRecords (
    PatientID INT NOT NULL PRIMARY KEY IDENTITY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Weight INT NULL,
    Height INT NULL,
    FacilityID INT NULL,
    CheckInTime DATETIME NULL,
    CheckOutTime DATETIME NULL,
    ReferringPhysician NVARCHAR(50) NULL,
    ReferringDoctor NVARCHAR(50) NULL,
    MedicationInstruction NVARCHAR(50) NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient (PatientID)
);

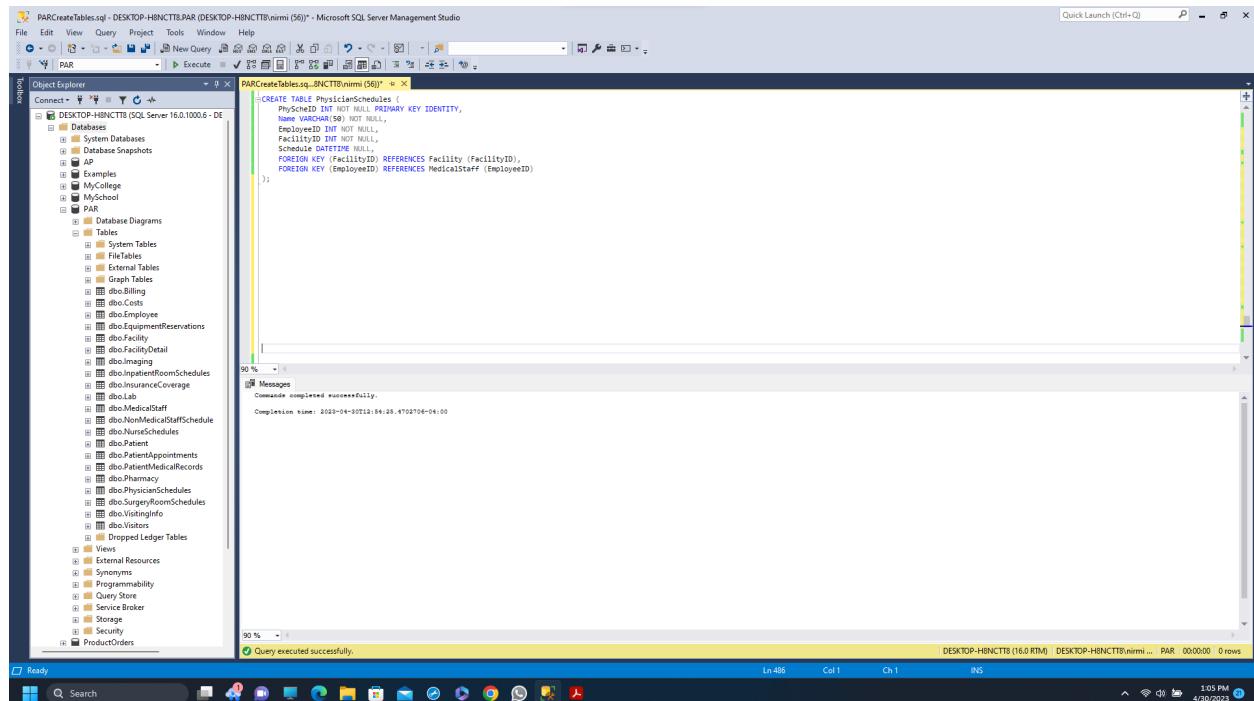
```

Messages

Commands completed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.



```

CREATE TABLE PhysicianSchedules (
    PhysicianID INT NOT NULL PRIMARY KEY IDENTITY,
    EmployeeID NVARCHAR(50) NOT NULL,
    EmployeeID INT NOT NULL,
    FacilityID INT NOT NULL,
    EmployeeID INT NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);

```

Messages

Commands completed successfully.

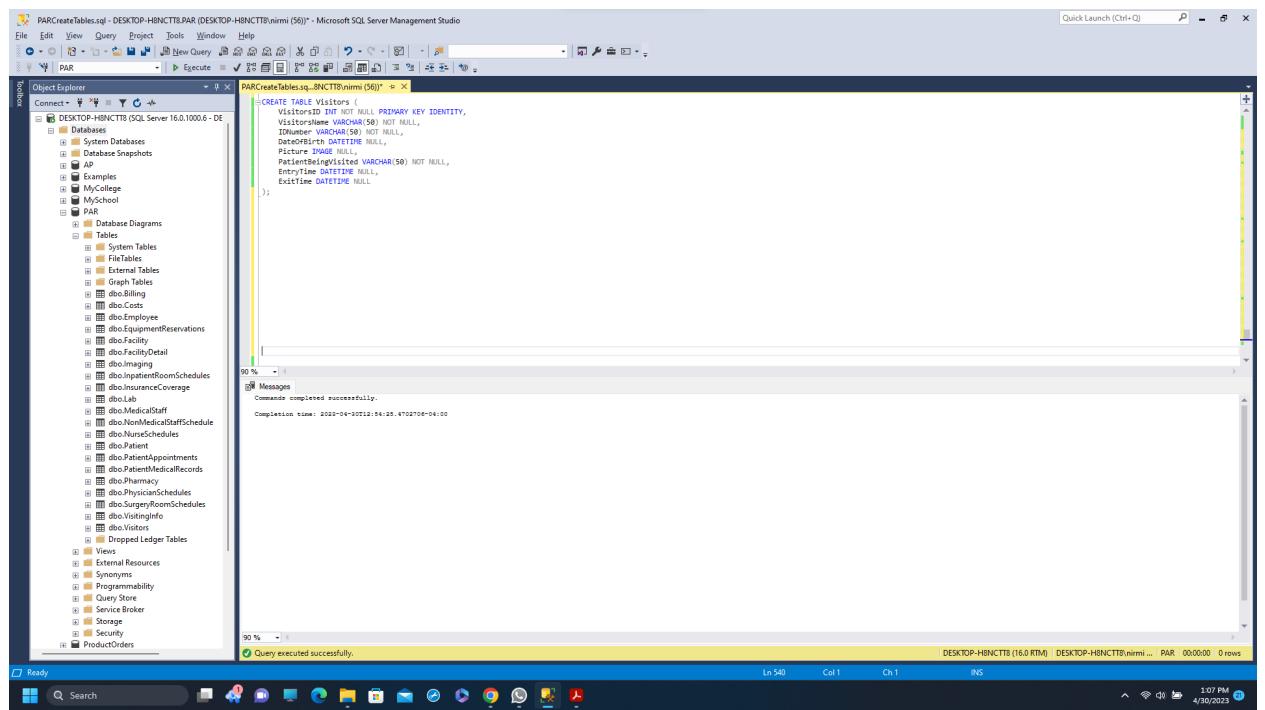
Completion time: 2023-04-30T12:54:28.4702756-04:00

Query executed successfully.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "PARCreateTables.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (56)) - Microsoft SQL Server Management Studio". The Object Explorer sidebar shows the database structure, including the "PAR" database which contains various tables like "SurgeryRoomSchedules", "Facility", "Employee", etc. The main query editor window displays a CREATE TABLE statement for "SurgeryRoomSchedules". The command is:

```
CREATE TABLE SurgeryRoomSchedules (
    SurRoomScheduleID INT PRIMARY KEY IDENTITY,
    FacilityID INT NOT NULL,
    SurgeryRoomName VARCHAR(50) NOT NULL,
    RoomCapacity INT NOT NULL,
    FOREIGN KEY (FacilityID) REFERENCES Facility (FacilityID),
    FOREIGN KEY (EmployeeID) REFERENCES MedicalStaff (EmployeeID)
);
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2023-04-30T12:54:25.4702756-04:00". The bottom right corner shows the system tray with the date and time: "4/30/2023 10:07 PM".



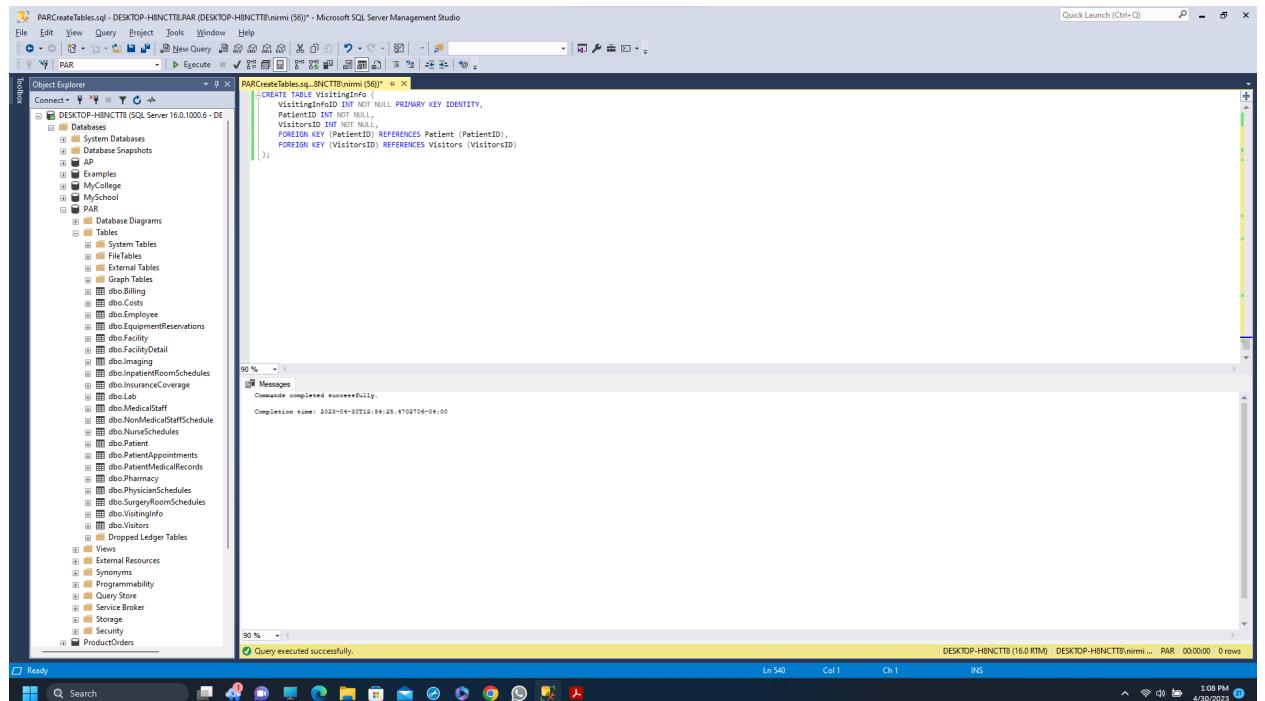
```

CREATE TABLE [dbo].[Visitors]
(
    [VisitorID] INT NOT NULL PRIMARY KEY IDENTITY,
    [VisitorName] VARCHAR(50) NOT NULL,
    [IDNumber] VARCHAR(50) NOT NULL,
    [DateOfBirth] DATETIME NULL,
    [Gender] CHAR(1) NULL,
    [PatientBeingVisited] VARCHAR(50) NOT NULL,
    [EntryTime] DATETIME NULL,
    [ExitTime] DATETIME NULL
);

```

Query executed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00



```

CREATE TABLE [dbo].[Visitors]
(
    [VisitorID] INT NOT NULL PRIMARY KEY IDENTITY,
    [PatientID] INT NOT NULL,
    [VisitorsID] INT NOT NULL,
    FOREIGN KEY ([PatientID]) REFERENCES Patient ([PatientID]),
    FOREIGN KEY ([VisitorsID]) REFERENCES Visitors ([VisitorsID])
);

```

Query executed successfully.

Completion time: 2023-04-30T12:54:28.4702756-04:00

2. Data Entries:

Two screenshots of Microsoft SQL Server Management Studio (SSMS) showing the execution of two SQL scripts: PARInsertData.sql and PARCreateTables.sql.

Screenshot 1 (Top): PARInsertData.sql Execution

```

SET IDENTITY_INSERT Patient OFF
-- Insert patient data
(1, 'Paulette Walters', '965 Havenhill Dr., Hamilton, OH 45013', '2895814801', 'jma@yahoo.com', NULL, 'Dean Hill'),
(2, 'Richard Mcneil', '184 S 5th St., Le Claire, IA 52753', '584635914', 'met@gmail.com', NULL, 'Jake Ellis'),
(3, 'Linda Jackson', '1234 Main Street, Cedar Rapids, IA 52404', '3195551234', 'linda.jackson@workplace.com', NULL, 'Sarah Parker'),
(4, 'Lloyd Austin', '48851 140th st, Groton, SD 57445', '2838493114', 'keuter@live.com', NULL, 'Kim Carpenter'),
(5, 'Robin Valdez', '1208 Northwood Dr., Champaign, IL 61821', '8985801608', 'tboern@msn.com', NULL, 'Michele Torres'),
(6, 'Greg Casey', '1111 Zionsville Rd., Maryland, MD 21201', '1775598971', 'igoreshnikovglobal.net', NULL, 'Alejandro Pratt'),
(7, 'John Williams', '200 2nd St., Suite 1111, Atlanta, GA 30303', '4045555555', 'john.williams@workplace.com', NULL, 'Dwight Gilbert'),
(8, 'Tulia Carson', '1763 3rd St., Coayhoga Falls, OH 44221', '6393896245', 'costinnett.net', NULL, 'Tracy Morris'),
(9, 'Christie Moody', '5665 Relander Pl., Pasco Roiles, CA 93440', '8935944099', 'ching@hotmail.com', NULL, 'Rene Stewart'),
(10, 'Lorraine Johnson', '1234 Main Street, Cedar Rapids, IA 52404', '3195551234', 'lorrainejohnson@workplace.com', NULL, 'Claude Perkins')
SET IDENTITY_INSERT Patient ON
  
```

Query executed successfully.

Screenshot 2 (Bottom): PARCreateTables.sql Execution

```

SET IDENTITY_INSERT Billing OFF
-- Create Billing table
CREATE TABLE Billing(BillingID INT IDENTITY(1,1) PRIMARY KEY, PatientID INT, Visits INT, MedicalBillingCodes NVARCHAR(50), Payor NVARCHAR(50), PaymentMethod NVARCHAR(50))
-- Insert sample data into Billing
(1, 1, NULL, '0881554802', NULL, 'Card'),
(1, 2, NULL, '0881554802', NULL, 'Card'),
(3, 3, NULL, 'E24659870', NULL, 'Card'),
(4, 4, 'Good visits', 'L27021631',NULL, 'Cash'),
(5, 5, 'Good visits', 'A22948025',NULL, 'Card'),
(6, 6, 'Good visits', 'A22838763',NULL, 'Card'),
(7, 7, 'Good visits', 'E22948025',NULL, 'Cash'),
(8, 8, 'Good visits', 'L47581032',NULL, 'Cash'),
(9, 9, 'Good visits', 'A22838763',NULL, 'Card'),
(10, 10, NULL, 'F89341139', NULL, 'Card')
SET IDENTITY_INSERT Billing ON
  
```

Query executed successfully.

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```

SET IDENTITY_INSERT Medications ON
INSERT Costs INTO MedicationCosts (MedicationID, ProcedureID, Medications, HealthProcedures)VALUES
(1,5, 'Acetaminophen', 'Aspirin'),
(2,6, 'Cyclohexespine', 'Medical inspection'),
(3,7, 'Ibuprofen', 'Urinalysis'),
(4,8, 'Omeprazole', 'Percussion (medicine)'),
(5,9, 'Adrenal', 'Vital signs measurement'),
(6,10, 'Diphenhydramine', 'Blood test'),
(7,3, 'Sandlace', 'Blood test'),
(8,2, 'Opertro', 'Stool test'),
(9,4, 'Nitriphyline', 'Urinalysis'),
(10,1, 'Benzoylamine', 'Cardiac stress test')
SET IDENTITY_INSERT Medications OFF

```

PARCreateTables.sql - DESKTOP-H8NCTTB\Nirmi (63) - Microsoft SQL Server Management Studio

```

CREATE TABLE [dbo].[MedicationCosts] (
    [MedicationID] INT NOT NULL,
    [ProcedureID] INT NOT NULL,
    [Medications] NVARCHAR(50) NOT NULL,
    [HealthProcedures] NVARCHAR(50) NOT NULL,
    CONSTRAINT [PK_MedicationCosts] PRIMARY KEY CLUSTERED ([MedicationID] ASC, [ProcedureID] ASC)
) ON [PRIMARY]

```

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```

SET IDENTITY_INSERT Lab ON
INSERT Lab INTO Lab (LabID, LabName, Tests, Results, MandatesReportingCountyState)VALUES
(1, 'Blood Lab', 'Blood Tests', 'Positive', 'Reported'),
(2, 'Dean Lab', 'Kidney Tests', 'Positive', 'Reported'),
(3, 'Cottage Lab', 'Hepatitis Testing', 'Negative', 'Reported'),
(4, 'Eckart Lab', 'Urine Tests', 'Positive', 'Reported'),
(5, 'Immunization Lab', 'Laboratory Tests', 'Positive', 'Reported'),
(6, 'Elmer Lab', 'Prenatal Testing', 'Negative', 'Reported'),
(7, 'Veronica Lab', 'Blood Tests', 'Positive', 'Reported'),
(8, 'Veronica Lab', 'Albain Blood Test', 'Positive', 'NULL'),
(9, 'Neigh Lab', 'Acid-Fast Sputillus (AFB) Tests', 'Positive', 'Reported'),
(10, 'White, Patricia Lab', 'Blood Count Tests', 'Negative', 'Reported')
SET IDENTITY_INSERT Lab OFF

```

PARCreateTables.sql - DESKTOP-H8NCTTB\Nirmi (63) - Microsoft SQL Server Management Studio

```

CREATE TABLE [dbo].[Lab] (
    [LabID] INT NOT NULL,
    [LabName] NVARCHAR(50) NOT NULL,
    [Tests] NVARCHAR(50) NOT NULL,
    [Results] NVARCHAR(50) NOT NULL,
    [MandatesReportingCountyState] NVARCHAR(50) NOT NULL,
    CONSTRAINT [PK_Lab] PRIMARY KEY CLUSTERED ([LabID] ASC)
) ON [PRIMARY]

```

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```
SET IDENTITY_INSERT Facility ON
INSERT Facility(FacilityID, FacilityName, Location, Hours, RoomCapacity, Departments, CareCapacity, MedicalEquipment) VALUES
(1, 'Central Hospital', 'Flushing', '2008-12-23', '3 rooms 6 beds', 'Pediatrics', 6, 3),
(2, 'Castle Hill Medical Center of New York Inc.', 'Bronx', '2007-01-01', '3 rooms 3 beds', 'Surgery', 3, 3),
(3, 'Primary Care Associates', 'Endicott', '1998-02-01', '4 rooms 8 beds', 'Hospital Extension Clinic', 10, 8),
(4, 'North Shore University Hospital', 'Manhattan', '2009-03-20', '5 rooms 10 beds', 'Emergency Services', 6, 5),
(5, 'Headbrook Endoscopy Center', 'Westbury', '2008-10-15', '4 rooms 4 beds', 'Surgery', 4, 4),
(6, 'New York Endoscopy Center', 'White Plains', '2009-10-05', '10 rooms 20 beds', 'Diagnostic and Treatment Center', 20, 20),
(7, 'PNCI - Primary Care Institute', 'Bronx', '2008-04-01', '10 rooms 10 beds', 'Primary Care Institute', 20, 15),
(8, 'Health Place', 'Bronx', '2008-07-23', '5 rooms 8 beds', 'Adult Day Health Care Program', 10, 8),
(9, 'Griffiss Surgery Center', 'Rome', '2012-01-03', '5 rooms 5 beds', 'Surgery', 5, 5),
(10, 'H.K. Freedman Renal Center', 'Pittsburgh', '2013-12-05', '4 rooms 8 beds', 'Diagnostic and Treatment Center', 10, 8)
SET IDENTITY_INSERT Facility OFF
```

PARCreateTable.sql - DESKTOP-H8NCTTB\Nirmi (56) - Microsoft SQL Server Management Studio

```
CREATE TABLE EquipmentReservations
(
    ReservationID INT IDENTITY(1,1) PRIMARY KEY,
    EquipmentID INT,
    FacilityID INT,
    CheckInTime DATETIME,
    CheckOutTime DATETIME
)
```

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```
SET IDENTITY_INSERT EquipmentReservations ON
INSERT EquipmentReservations(EquipmentID, FacilityID, CheckInTime, CheckOutTime)VALUES
(1, 1, '2021-03-01', '2021-03-17'),
(2, 3, 'Autoclaves', 2021-03-01, 2021-03-15),
(3, 5, 'Patient Monitors', 2021-03-01, NULL),
(4, 7, 'Surgical Tools', 2021-03-01, 2021-03-05),
(5, 9, 'Sterilizers', 2021-03-01, 2021-03-20),
(6, 8, 'Surgical Tables', 2021-03-10, NULL),
(7, 10, 'Electro and Fluid Infusion', 2021-03-10, 2021-03-12),
(8, 6, 'Dishwasher Machine', 2021-02-01, 2021-02-10),
(9, 7, 'Defibrillators', 2021-04-03, NULL),
(10, 6, 'Surgical Lights', 2021-02-07, 2021-02-10)
SET IDENTITY_INSERT EquipmentReservations OFF
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to 'DESKTOP-H8NCTTB' (SQL Server 16.0.1000.6 - DE). The Object Explorer sidebar shows the database structure, including tables like 'dbo.Billing', 'dbo.Costs', 'dbo.Employee', etc. The main query window displays the execution of a stored procedure named 'PARInsertData'. The command inserted 10 rows into the 'Image' table, with values ranging from '(1, 'Direct', 'MRI')' to '(10, 'Indirect', 'MRI10')'. The execution completed successfully, as indicated by the message 'Query executed successfully.' at the bottom.

```
SET IDENTITY_INSERT Image ON
INSERT Imaging.Imaging IN ImageName, Testing VALUES
(1, 'Direct', 'MRI'),
(2, 'Indirect', 'CAT'),
(3, 'Indirect', 'MRI'),
(4, 'Direct', 'MRI'),
(5, 'Direct', 'MRI'),
(6, 'Indirect', 'MRI'),
(7, 'Indirect', 'MRI7'),
(8, 'Indirect', 'MRI8'),
(9, 'Indirect', 'MRI9'),
(10, 'Indirect', 'MRI10')
SET IDENTITY_INSERT Image OFF
```

PARInsertData.sql - DESKTOP-HBNCTTB.PAR (DESKTOP-HBNCTTB\mrimi (63)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query New Query - Execute

PAR

Object Explorer

PARInsertData.sql - HBNCTTB.mrimi (63)* | PARCreateTables.sql - HBNCTTB.mrimi (56)

SET IDENTITY_INSERT Pharmacy ON
1. 'Bone Drug', 'Allopurinol', 'Aldendronate', 'Alentuzumab', 'Defibrillators',
(1, 'Bone Drug', 'Allopurinol', 'Aldendronate', 'Alentuzumab', 'Defibrillators'),
(2, 'Bentell Drug', 'Iofosfamide', 'Imatinib', 'Injepmer', 'Anesthesia Machines'),
(3, 'Blood Pharmacy', 'Deseretine', 'Dactinomycin', 'Daptomycin', 'Patient Monitors'),
(4, 'Blood Product Manager', 'Glatuzumab', 'Mestinon', 'Lamivudine', 'EDGED Machines', 'Heparin Lock Flush'),
(5, 'Hemipig Drug', 'Heparin Lock Flush', 'Sterilizers'),
(6, 'London Drugs', 'Metaramine', 'Nelkinavir', 'Neuromuscular', 'Surgical Tablets'),
(7, 'London Drugs', 'Metformin', 'Nimodipine', 'Oxymetazoline', 'Electrosurgical Fluid Warmers'),
(8, 'Pharmaseas', 'Valproic acid', 'Vancomycin', 'Vimostazine', 'Electrosurgical Units'),
(9, 'Good Neighbor', 'Efavirenz', 'Efavirenz / emtricitabine', 'Surgical Tables'),
(10, 'Good Neighbor', 'Efavirenz', 'Efavirenz / tenofovir', 'Bortezomib', 'EDG/EDG Machines')
SET IDENTITY_INSERT Pharmacy OFF

Messages

(10 rows affected)
(10 rows affected)

90 %

90 %

Query executed successfully.

DESKTOP-HBNCTTB (16.0 RTM) | DESKTOP-HBNCTTB.mrimi ... | PAR | 00:00:00 | 0 rows

Ready

File Edit View Insert Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Lin 217 Col 1 Ch 1 INS

4/30/2024 1:19 PM

Nirmit Patel (SUID : 992323441)

Project 2

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to 'DESKTOP-HBNCTTB' (SQL Server 16.0.1000.6 - DE). The left sidebar displays the 'Object Explorer' with the 'PAR' database selected. The main pane shows two queries: 'PARInsertData.sql' and 'PARCreateTables.sql'. The 'PARCreateTables.sql' query is currently executing, as indicated by the progress bar at the bottom of the results grid. The results show 10 rows inserted into the 'FacilityDetail' table, each containing values from 1 to 10. A message at the bottom of the results grid states 'Query executed successfully.'

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

```
(10 rows affected)
```

90 %

Messages

Query executed successfully.

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB.nims (63)) - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

Quick Launch (Ctrl+Q)

PAR Object Explorer Connect Databases Snapshots AP Examples MyCollege MySchool PAR Database Diagrams Tables System Tables External Tables Graphs Reporting db-Corts db-Employees db-EquipmentReservations db-Facility db-FacilityDetail db-Reporting db-InsurerItemSchedules db-InsuranceCoverage db-Lab db-MedicalStaff db-MedicalStaffSchedule db-NursingSchedules db-Patient db-PatientAppointments db-PatientMedicalRecords db-Pharmacy db-PhysicianSchedules db-RoomSchedules db-VisitingInfo db-Visitor Dropped Ledger Tables Views External Resources Synonyms Programmability Query Store Service Broker Storage Security ProductOrders

PARInsertData.sql - DESKTOP-H8NCTTB.nims (63) | PARCreateTables.sql - H8NCTTB.nims (50)

```
SET IDENTITY_INSERT Employee ON
INSERT Employee([EmployeeID], Name, Address, Phone, Email, Salary, [Title], ContractType, ContractTerm, Reviews, Department, Office, OfficePhone, WorkSchedule) VALUES
--Employee 1
(1,'John Smith', '123 Main Street, Anytown, USA', '(555) 123-4567', 'jsmith@email.com', 50000, 'Software Developer', 'Full-Time', 'Excellent', 'IT', '123 Main St.', '(555) 123-4567', '9-5 PM')
--Employee 2
(2,'Kiyani Worthington', '581 South Peachtree Lane, Bronx, NY 10463', '8885227792', 'kiyanaworthington@gmail.com', 73453, 'NULL', 'fixed-price', 'NULL,NULL', 'administration', 'NULL', '8376350888', '9-17')
--Employee 3
(3,'Minnie Gileen', '785 Carnegie Drive, West Babylon, NY 11764', '6097353807', 'minniegileen@gmail.com', 637651, 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '0463209885', '9-17')
--Employee 4
(4,'Diana M. Williams', '1234 Elmwood Ave., Anytown, USA', '(555) 123-4568', 'dianamw@gmail.com', 598551, 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '9-16')
--Employee 5
(5,'Terri Myers', '2 North Berkshire Avenue, Woonsocket, RI 02895', '4567890123', 'terri.myers@gmail.com', 68755, 'NULL', 'cost-reimbursement', 'NULL,NULL', 'administration', 'NULL', '9611186462', '8-16')
--Employee 6
(6,'Vilma Huhan', '24518 Melysia Meadow Apt. 332 Meer Adelanto, CA 91173', '6098788904', 'refref_ciney@yahoo.com', '54464', 'bonus', 'fixed-price', 'NULL,NULL', 'administration', 'NULL', '1628118873', '9-17')
--Employee 7
(7,'Barry Carpenter', '456 Wilson Street, Saugerties, NY 12477', '5555555555', 'barry.carpenter@gmail.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '7737970800', '9-17')
--Employee 8
(8,'Lorraine S. Jones', '15780 10th Street, Denver, CO 80210', '5555555555', 'lorraine.jones@gmail.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '9-17')
--Employee 9
(9,'Darell Maynor', '3338 VonRuden Lakes Suite 683 Goodinville, NE 81810', '7998551210', 'alexking@gmail.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'administration', 'NULL', '0422473888', '9-17')
--Employee 10
(10,'Niyah Kem', '110 Skyla Estate Point Hittie, SC 49513', '6748384587', 'crooks.alicia@yahoo.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '0814764444', '9-17')
--Employee 11
(11,'Troy Jackson', '1234 Elmwood Ave., Anytown, USA', '(555) 123-4569', 'troy.jackson@gmail.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '9-17')
--Employee 12
(12,'Maggie Whittington', '401 Odessa Turnpike Port Budown, AR 89911-3645', '4671739881', 'maggiowhukert.com', '65454', 'bonus', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '7915337384', '9-17')
--Employee 13
(13,'Tessie Currie', '671 Gottilie Row Apt. 177 Schiedenberga, AL 12451-0085', '8786715831', 'telli2@chmeler.net', '87855', 'bonus', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '3989926819', '9-17')
--Employee 14
(14,'Lorraine S. Jones', '15780 10th Street, Denver, CO 80210', '5555555555', 'lorraine.jones@gmail.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '9-17')
--Employee 15
(15,'Deanne Snodgrass', '172 Crisscross Road, Lake Cedar, MI 49637-1000', '8252208900', 'alberta.casser@lindeng.com', '45645', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '9-17')
--Employee 16
(16,'Carly Collier', '331 Leopold Prairie Apt. 792 West Traemouth, MA 02808-1828', '5080203835', 'delores.christiansen@gmail.com', '45645', 'NULL', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '0089440881', '8-16')
--Employee 17
(17,'Lorraine S. Jones', '15780 10th Street, Denver, CO 80210', '5555555555', 'lorraine.jones@gmail.com', '45645', 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '8-16')
--Employee 18
(18,'Teddy Grey', '255 Carter Junction Suite 221 Deveryn, SC 86155-1280', '9887274679', 'hilts.andreanne@yahoo.com', '53724', 'NULL', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '8-16')
--Employee 19
(19,'Teddy Grey', '255 Carter Junction Suite 221 Deveryn, SC 86155-1280', '9887274679', 'hilts.andreanne@yahoo.com', '53724', 'NULL', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '8-16')
--Employee 20
(20,'Annet Hobbs', '4733 Trestel Vista Apt. 708 West Mettisberg, LA 88682-1000', '6827846995', 'scwmit.helene@anglo.com', '77644', 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '8-16')
--Employee 21
(21,'Reuben Burt', '767 Lucienne Land Apt. 646 West Filomenaville, IA 31760-6939', '7281904046', 'wolff.nora@connelly.com', '34334', 'bonus', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '8-16')
--Employee 22
(22,'Imani Currie', '3832 Idella Underseed Lake Neck, NC 83883', '4791170552', 'lesusche.estelle@gmail.com', '76544', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '8-16')
--Employee 23
(23,'Amara Jenkins', '27064 Jenkins Mission Suite 612 Naderhaven, NC 74759-2212', '8606231800', 'mee55@mtl1.com', '45574', 'bonus', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '7108295608', '8-16')
--Employee 24
(24,'Amaran Fenwick', '1588 Jade Keys Suite 388 Stamford, CT 06917-4200', '6094434911', 'teaman.fenwick@soas.com', '45574', 'bonus', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '7379106861', '8-16')
--Employee 25
(25,'Lorraine S. Jones', '15780 10th Street, Denver, CO 80210', '5555555555', 'lorraine.jones@gmail.com', '45645', 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '8-16')
--Employee 26
(26,'Fleur Patterson', '432 Hahn Road Apt. 266 Herstedt, MN 88476-3499', '3268578461', 'xdixitson@yahoo.com', '76545', 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '9876543210', '8-16')
--Employee 27
(27,'Bobbie Patterson', '912 Hand Flats Elodiotype, GA 68464', '0852373171', 'aggettib@orizon.com', '57353', 'NULL', 'fixed-price', 'NULL,NULL', 'logistics', 'NULL', '3412970749', '8-16')
--Employee 28
(28,'Myles Pacheco', '1994 Quigley Path Jenkinsport, VA 72548-7449', '8895636740', 'abagxl.kishner@hotmail.com', '65734', 'bonus', 'fixed-price', 'NULL,NULL', 'administration', 'NULL', '951996269', '8-16')
--Employee 29
(29,'Myles Pacheco', '1994 Quigley Path Jenkinsport, VA 72548-7449', '8895636740', 'abagxl.kishner@hotmail.com', '65734', 'bonus', 'fixed-price', 'NULL,NULL', 'administration', 'NULL', '951996269', '9-17')
--Employee 30
(30,'Herbert Burt', '767 Lucienne Land Apt. 646 West Filomenaville, IA 31760-6939', '7281904046', 'wolff.nora@connelly.com', '34334', 'bonus', 'fixed-price', 'NULL,NULL', 'medical', 'NULL', '8-16')

SET IDENTITY_INSERT Employee OFF
```

The image shows two separate instances of Microsoft SQL Server Management Studio (SSMS) running on a Windows operating system. Both instances are connected to the same database, DESKTOP-H8NCTTB (SQL Server 16.0.1000.6 - DE).

Query 1 (Top Window):

```

SET IDENTITY_INSERT MedicalStaff ON
INSERT INTO [dbo].[MedicalStaff]([EmployeeID], [Name], [Type], [Position], [Status]) VALUES
(1, 'Vimian Moyer', 'Physician', 'attending', 'active'),
(2, 'Alyson Worthington', 'Physician', 'attending', 'active'),
(3, 'Leland Kinsella', 'Physician', 'attending', 'active'),
(4, 'Inesya Dyer', 'Physician', 'PCP', 'active'),
(5, 'Tariq Ayers', 'Physician', 'PCP', 'retired'),
(6, 'Dwight Gandy', 'Physician', 'PCP', 'active'),
(7, 'Harry Carpenter', 'Physician', 'PCP', 'active'),
(8, 'Jaedee Noel', 'Physician', 'attending', 'active'),
(9, 'Darrell Taylor', 'Physician', 'attending', 'leave'),
(10, 'Cecilia Ladd', 'Physician', 'attending', 'leave'),
(11, 'Randy Hopper', 'Nurse', 'visiting', 'leave'),
(12, 'Augie Whittington', 'Nurse', 'PCP', 'active'),
(13, 'Lynne Gandy', 'Nurse', 'PCP', 'active'),
(14, 'Jamie Friedman', 'Nurse', 'PCP', 'retired'),
(15, 'Deelan Shan', 'Nurse', 'PCP', 'active'),
(16, 'Cynthia Ladd', 'Nurse', 'PCP', 'retired'),
(17, 'Audrey Bailey', 'Nurse', 'PCP', 'active'),
(18, 'Merlow Emerson', 'Nurse', 'attending', 'active'),
(19, 'Teddy Grey', 'Nurse', 'attending', 'leave'),
(20, 'Anna Hobbs', 'Nurse', 'visiting', 'active')
SET IDENTITY_INSERT MedicalStaff OFF

```

Query 2 (Bottom Window):

```

SET IDENTITY_INSERT InpatientRoomSchedules ON
INSERT INTO [dbo].[InpatientRoomSchedules]([InpatientRoomScheduleID], FacilityID, EmployeeID, InPatientRoomName, PatientID, Schedule)VALUES
(1, '11', '1001', '1', '2021-05-01'),
(2, '12', '1002', '2', '2021-05-02'),
(3, '13', '1003', '3', '2021-05-03'),
(4, '14', '1004', '4', '2021-05-04'),
(5, '15', '1005', '5', '2021-05-05'),
(6, '16', '1006', '6', '2021-05-06'),
(7, '17', '1007', '7', '2021-05-07'),
(8, '18', '1008', '8', '2021-05-08'),
(9, '19', '1009', '9', '2021-05-09'),
(10, '20', '1010', '10', '2021-05-10')
SET IDENTITY_INSERT InpatientRoomSchedules OFF

```

Both queries have executed successfully, as indicated by the status bar at the bottom of each SSMS window.

The screenshot shows two separate sessions in Microsoft SQL Server Management Studio (SSMS) against the PAR database.

Session 1 (Top Window):

```

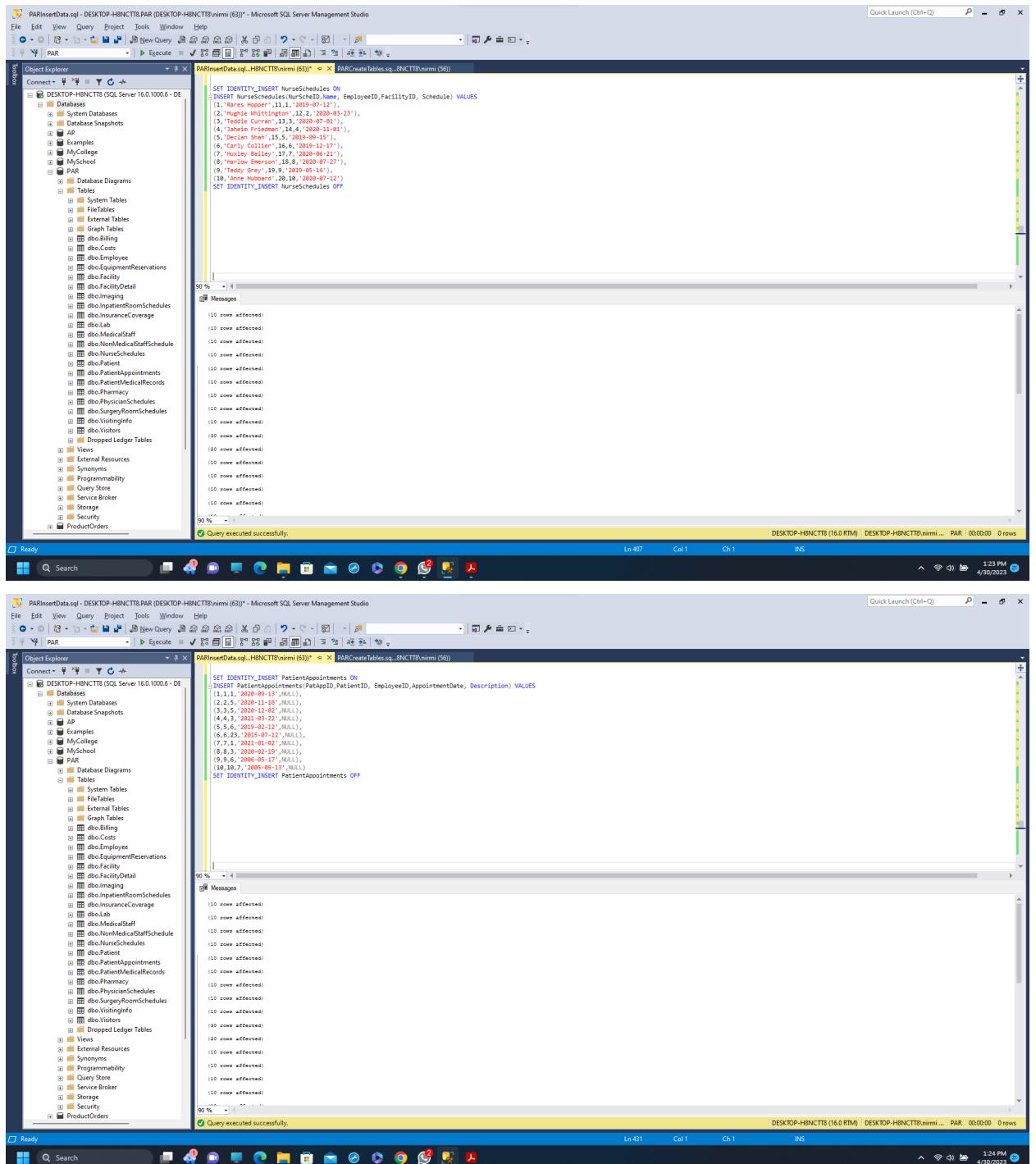
SET IDENTITY_INSERT InsuranceCoverage ON
INSERT InsuranceCoverage(InsuranceID, PatientID, HealthInsuranceCompanyName, Phone, ContactInfo, InsuranceCoverages) VALUES
(1, 1, 'UnitedHealthcare Group', '6442725986', NULL, 'Full Coverage'),
(2, 2, 'Anthem', '3527353380', NULL, 'Full Coverage'),
(3, 3, 'Cigna', '442725986', NULL, 'Full Coverage'),
(4, 4, 'UnitedHealthcare Group', '6442725986', NULL, 'Full Coverage'),
(5, 5, 'Anthem', '3527353380', NULL, 'Full Coverage'),
(6, 6, 'Cigna', '442725986', NULL, 'Full Coverage'),
(7, 7, 'UnitedHealthcare Group', '6442725986', NULL, 'Half Coverage'),
(8, 8, 'Humana', '4831732929', NULL, 'Full Coverage'),
(9, 9, 'Aetna', '6713031247', NULL, 'Half Coverage'),
(10, 10, 'Cigna', '0800713595', NULL, 'Full Coverage')
SET IDENTITY_INSERT InsuranceCoverage OFF
  
```

Session 2 (Bottom Window):

```

SET IDENTITY_INSERT NonMedicalStaffSchedule ON
INSERT NonMedicalStaffSchedule(NonMedStaffScheduleID, EmployeeID, FacilityID, Schedule)VALUES
(1, 25, 5, '2021-05-01'),
(2, 24, 4, '2021-05-02'),
(3, 23, 3, '2021-05-03'),
(4, 22, 2, '2021-05-04'),
(5, 21, 1, '2021-05-05'),
(6, 20, 10, '2021-05-06'),
(7, 27, 7, '2021-05-07'),
(8, 28, 8, '2021-05-08'),
(9, 29, 9, '2021-05-09'),
(10, 30, 10, '2021-05-10')
SET IDENTITY_INSERT NonMedicalStaffSchedule OFF
  
```

Both sessions show a successful execution message at the bottom: "Query executed successfully."



The screenshot shows two Microsoft SQL Server Management Studio windows side-by-side.

Top Window (PARInsertData.sql):

```

SET IDENTITY_INSERT NursesMedicines ON
INSERT NursesMedicines(NurseID,Name,EmployeeID,FacilityID,Schedule) VALUES
(1,'Naree Kopper',11,1,'2019-07-12'),
(2,'Augie Whittington',12,2,'2020-03-23'),
(3,'Liam Gandy',13,3,'2019-09-17'),
(4,'Janice Friedman',14,4,'2020-11-01'),
(5,'Declan Shan',15,5,'2019-09-15'),
(6,'Dawn Ladd',16,6,'2020-07-27'),
(7,'Husley Bailey',17,7,'2020-06-21'),
(8,'Harlow Emerson',18,8,'2020-07-27'),
(9,'Teddy Grey',19,9,'2019-08-14'),
(10,'Annie Robard',20,10,'2020-07-12')
SET IDENTITY_INSERT NursesMedicines OFF
  
```

Bottom Window (PARInsertData.sql):

```

SET IDENTITY_INSERT PatientAppointments ON
INSERT PatientAppointments(PetAppID,PatientID,EmployeeID,AppointmentDate,Description) VALUES
(1,1,1,'2020-09-13',NULL),
(2,2,5,'2020-11-18',NULL),
(3,3,3,'2020-09-25',NULL),
(4,4,3,'2021-03-22',NULL),
(5,5,6,'2019-02-12',NULL),
(6,6,25,'2020-01-13',NULL),
(7,7,25,'2021-01-02',NULL),
(8,8,3,'2020-02-19',NULL),
(9,9,6,'2020-05-17',NULL),
(10,10,1,'2020-06-13',NULL)
SET IDENTITY_INSERT PatientAppointments OFF
  
```

Both windows show the message "Query executed successfully." at the bottom.

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```

SET IDENTITY_INSERT PatientMedicalRecords ON
INSERT INTO PatientMedicalRecords (PatientID, ClientID, weight, height, vitals, CheckInTime, CheckOutTime, SymptomDiagnosticProcedureCodes, AttendingPhysician, ReferralDoctor, MedicationHospitalizationDischarge) VALUES
(1, 1, 131, '2021-05-03 08:12:04', '2021-05-03 13:31:44', '1', 'Insel Hepp', 'Insel Hepp', NULL),
(2, 2, 72, 172, NULL, '2021-05-03 08:12:04', '2021-05-03 08:22:41', '2', 'Virgil Jefferson', 'Virgil Jefferson', NULL),
(3, 3, 73, 173, NULL, '2021-05-03 08:23:05', '2021-05-03 12:48:20', '3', 'Chad Bryant', 'Chad Bryant', NULL),
(4, 4, 74, 174, NULL, '2021-05-03 08:23:05', '2021-05-03 12:48:20', '4', 'Loreen Beck', 'Loreen Beck', NULL),
(5, 5, 75, 175, NULL, '2021-05-04 07:36:08', '2021-05-06 22:28:24', '5', 'Jodi Bailey', 'Jodi Bailey', NULL),
(6, 6, 76, 176, NULL, '2021-05-08 01:49:53', '2021-05-08 08:53:42', '6', 'Pamela Chambers', 'Pamela Chambers', NULL),
(7, 7, 77, 177, NULL, '2021-05-10 08:08:23', '2021-05-10 13:48:33', '7', 'Maurice Klein', 'Maurice Klein', NULL),
(8, 8, 78, 178, NULL, '2021-05-14 03:08:23', '2021-05-15 05:04:33', '8', 'Melanie Klein', 'Melanie Klein', NULL),
(9, 9, 79, 179, NULL, '2021-05-17 02:34:05', '2021-05-17 11:42:32', '9', 'Donna Hardy', 'Donna Hardy', NULL),
(10, 10, 80, 180, NULL, '2021-05-18 01:35:16', '2021-05-18 09:31:11', '10', 'Sadie Martinez', 'Sadie Martinez', NULL)
SET IDENTITY_INSERT PatientMedicalRecords OFF

```

Messages

```

(10 rows affected)

```

Query executed successfully.

LN 456 Col 1 Ch 1 INS DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\Nirmi ... PAR 00:00:00 0 rows

PARInsertData.sql - DESKTOP-H8NCTTB.PAR (DESKTOP-H8NCTTB\Nirmi (63)) - Microsoft SQL Server Management Studio

```

SET IDENTITY_INSERT PhysicianSchedules ON
INSERT INTO PhysicianSchedules (PhysicianName, EmployeeID, FacilityID, Schedule) VALUES
(1, 'Kylie Wootton', '2,2', '2021-05-21'),
(2, 'Minnie Glen', '3,3', '2021-05-19'),
(3, 'Lorraine Hart', '4,4', '2021-05-21'),
(4, 'Tarcia West', '5,4', '2021-05-18'),
(5, 'Vince Hulen', '6,10', '2021-05-19'),
(7, 'Berry Carpenter', '7,5', '2021-05-21'),
(8, 'Dawn Jackson', '8,9', '2021-05-21'),
(9, 'Dorell Naylor', '9,6', '2021-05-08'),
(10, 'Niyah Kemp', '10,8', '2021-05-19')
SET IDENTITY_INSERT PhysicianSchedules OFF

```

Messages

```

(10 rows affected)

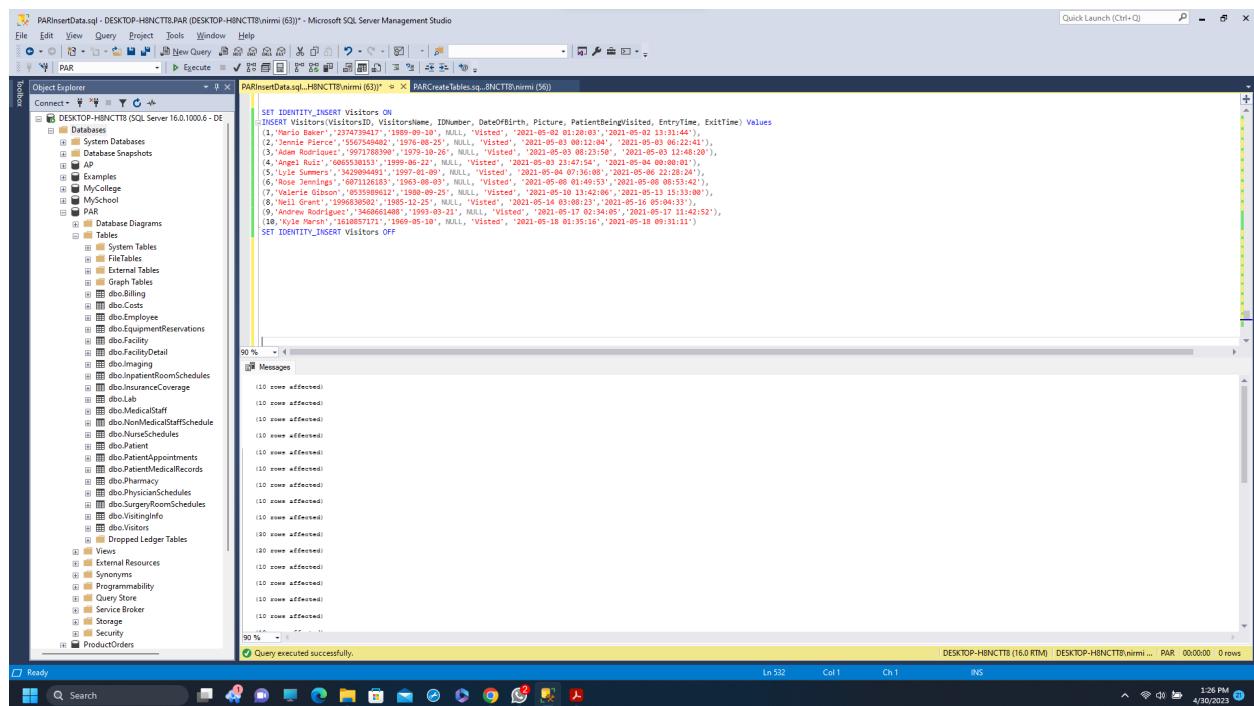
```

Query executed successfully.

LN 482 Col 1 Ch 1 INS DESKTOP-H8NCTTB (16.0 RTM) DESKTOP-H8NCTTB\Nirmi ... PAR 00:00:00 0 rows

```
SET IDENTITY_INSERT SurgeryRoomSchedules ON
INSERT SurgeryRoomSchedules(SurgeryRoomScheduleID, EmployeeID, FacilityID, SurgeryRoomName, Schedule)VALUES
(1, 6, 1, 'firstclass1', '2021-05-01'),
(1, 6, 1, 'secondclass1', '2021-05-01'),
(3, 3, 3, 'firstclass1', '2021-05-01'),
(3, 3, 3, 'secondclass1', '2021-05-03'),
(4, 5, 4, 'firstclass2', '2021-05-01'),
(5, 5, 5, 'secondclass2', '2021-05-01'),
(6, 6, 6, 'firstclass3', '2021-05-02'),
(7, 7, 7, 'firstclass4', '2021-05-07'),
(8, 7, 8, 'firstclass5', '2021-05-10'),
(9, 8, 9, 'firstclass6', '2021-05-12'),
(10, 10, 10, 'firstclass7', '2021-05-05')
SET IDENTITY_INSERT SurgeryRoomSchedules OFF
```

Query executed successfully.

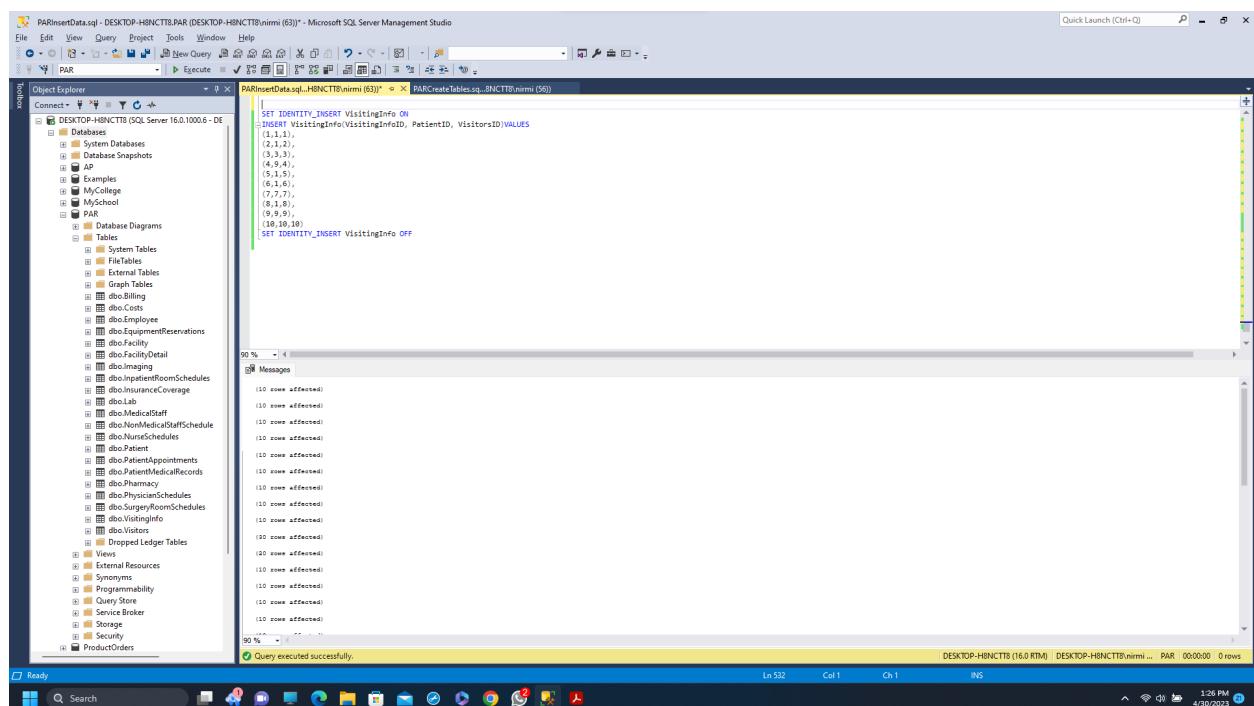


```

SET IDENTITY_INSERT Visitors ON
INSERT Visitors(VisitorsID, VisitorName, IDNumber, DateOfBirth, Picture, PatientBeingVisited, EntryTime, ExitTime) VALUES
(1,'Maria Baker',2347349417,'1989-09-10',NULL,'Visited', '2021-05-02 01:20:03','2021-05-02 13:31:44'),
(2,'Jennifer Pierce',5567548402,'1976-08-25',NULL,'Visited', '2021-05-03 08:12:04','2021-05-03 08:12:41'),
(3,'John Smith',1234567890,'1995-03-15',NULL,'Visited', '2021-05-04 09:00:00','2021-05-04 09:00:00'),
(4,'Angel Ruiz',6065303033,'1999-06-22',NULL,'Visited', '2021-05-03 23:45:54','2021-05-04 00:00:01'),
(5,'Syle Summers',3429844801,'1987-01-09',NULL,'Visited', '2021-05-04 07:36:08','2021-05-06 22:08:24'),
(6,'Rose Jennings',6071261857,'1988-08-03',NULL,'Visited', '2021-05-05 08:00:51','2021-05-08 08:53:42'),
(7,'Tommy Gibson',1234567890,'1998-09-10',NULL,'Visited', '2021-05-06 13:00:00','2021-05-06 13:00:00'),
(8,'Neill Grant',1996389392,'1985-12-25',NULL,'Visited', '2021-05-14 03:00:23','2021-05-16 05:04:33'),
(9,'Andrew Rodriguez',3468614088,'1993-03-21',NULL,'Visited', '2021-05-17 02:34:05','2021-05-17 11:42:52'),
(10,'Kathy Davis',5109871121,'1969-05-10',NULL,'Visited', '2021-05-18 01:35:16','2021-05-18 09:31:11')
SET IDENTITY_INSERT Visitors OFF

```

Query executed successfully.



```

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

```

Query executed successfully.