Sudhanshu Kulkarni and Vedika Shenoy

HEALTHCARE PORTAL

Project implementation report

Table of Contents

[PROJECT SUMMARY 2](#_Toc7449071)

[BUSINESS RULES 3](#_Toc7449072)

[PHYSICAL MODEL 4](#_Toc7449073)

[ENTITY RELATIONSHIP DIAGRAM USING MS VISIO 4](#_Toc7449074)

[ENTITY RELATIONSHIP DIAGRAM IN MS SQL SERVER MANAGEMENT STUDIO 5](#_Toc7449075)

[DATA DICTIONARY 6](#_Toc7449076)

[DATABASE SYSTEM INFRASTRUCTURE 11](#_Toc7449077)

[SQL SCRIPT FOR CREATING AND INSERTING TABLE DATA 12](#_Toc7449078)

[CREATING TABLES IN SQL 12](#_Toc7449079)

[INSERTING TABLES IN SQL 20](#_Toc7449080)

[TRIGGERS 32](#_Toc7449081)

[MAJOR DATA QUESTIONS ANSWERED USING SQL 34](#_Toc7449082)

[INTERFACE IMPLEMENTATION USING FORMS 38](#_Toc7449083)

[LOGIN FORM 38](#_Toc7449084)

[PATIENT FORM 38](#_Toc7449085)

[DOCTOR FORM 39](#_Toc7449086)

[PATIENT SUMMARY FORM 39](#_Toc7449087)

[RECEPTIONIST FORM 40](#_Toc7449088)

[REPORT GENERATION FORM 40](#_Toc7449089)

[INTERFACE IMPLEMENTATION USING REPORTS 41](#_Toc7449090)

[MOST FREQUENT MEDICAL EXAM REPORT 41](#_Toc7449091)

[APPOINTMENTS PER DAY REPORT 41](#_Toc7449092)

[BILLS GENERATED PER PATIENT REPORT 42](#_Toc7449093)

[TOTAL INCOME REPORT 42](#_Toc7449094)

# PROJECT SUMMARY

The project involves the development of a complex database for the Healthcare portal. There are multiple individuals involved in this system namely patients, receptionists, doctors and insurance company. This project is for any organization such as a hospital or local medical clinic who are primarily focused on treating patients with various illnesses. This system will record the patient’s data, book appointments, generate medical exam reports, manage doctor’s schedule, handle insurance company coverage and generate bill for the patient upon request.

When a patient wants to visit a healthcare organization, he must first place an appointment via the portal. After the patient visits the doctor, he is advised to undergo various medical examinations in order to carry out proper diagnosis of the patient’s illness. Then the doctor prescribes further medication or surgery based on the severity of the illness. After this phase, the bill is generated on the portal by the receptionist. The bills that will be generated for the patient will include the medical exam fees and any additional cost incurred. This payment will be covered by the insurance company if the patient has an insurance.

There is a tremendous amount of data generated in this process. The patient needs to be assigned the doctor of the correct specialization for proper treatment. The doctor’s need to approve the appointments such that there is no multiple booking at the same time slot. In the end, the bill generation also involves collecting data from various sources and merging it before presenting it to the patient for payment. As all of these processes are automated, a lot of issues for managing the system are solved and scope for human errors is reduced.

The healthcare portal ensures efficient management of all this data. Each user can access the data that is only relevant to him. The data is organized in a structured manner to ensure automation, which in turn will reduce the manual efforts put in by the people.

This report further describes the entities and the associated attributes, the entity relationship model, business rules and physical implementation which will clarify any doubts which may arise in the minds of the readers.

# BUSINESS RULES

The following are the business rules of the system:

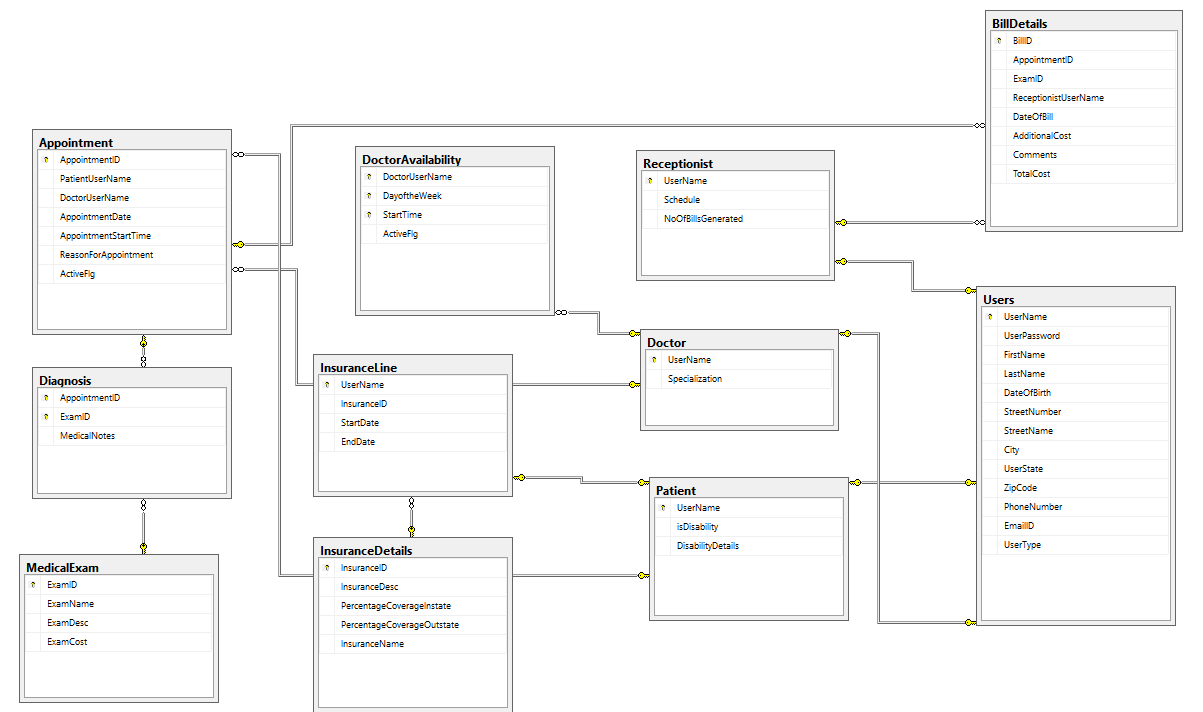
1. A patient must book an appointment before entering the healthcare center and it can only be done using the healthcare portal.
2. Doctors cannot cancel an appointment once they are assigned to it.
3. Doctors can access all the patient data whenever they need it.
4. A patient can undergo one medical exam only once in a day.
5. Patient must show up 15 minutes before the appointment.
6. A doctor can attend multiple appointments, but one appointment must be attended by only one doctor.
7. A patient can book only one doctor for one appointment.
8. A user of the healthcare portal must either be a patient, doctor or a receptionist.
9. One appointment can have only one bill.
10. One appointment can consist of more than one diagnosis, but one diagnosis is pertained to one appointment.
11. Only one type of insurance is provided to patients who opt for it.
12. One Diagnosis (entity name) can be determined by one medical exam, but one type of medical exam can make one or more Diagnosis.
13. One bill can consist of one or more Diagnosis (entity name), but one Diagnosis will be written in one bill.
14. One bill can consist of one or more medical exams and one type of medical exam can be included in many bills.
15. Insurance is provided by only one company to all the patients.

# PHYSICAL MODEL

## ENTITY RELATIONSHIP DIAGRAM USING MS VISIO



## ENTITY RELATIONSHIP DIAGRAM IN MS SQL SERVER MANAGEMENT STUDIO



# DATA DICTIONARY

Entities and their attributes in more detail:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Objects** | **Description** | **Data Type** | **Max length** | **PK/ FK** | **Required** | **Nullable** |
| **Users** | This table stores the details of the users who are registered in the system. |  |  |  |  |  |
| UserName | This is the primary key. This is the login username of the user. | Varchar | 20 | PK | Required | Not Null |
| UserPassword | This is the login password of the user. | Varchar | 20 |  | Required | Not Null |
| FirstName | First name of the user. | Varchar | 20 |  | Required | Not Null |
| LastName | Last name of the user. | Varchar | 20 |  | Required | Not Null |
| DateOfBirth | Date of birth of the user. | Date |  |  | Required | Not Null |
| StreetNumber | Stores the street number of the address section of the user. | Varchar | 5 |  | Required | Not Null |
| StreetName | Stores the street name of the address section of the user. | Varchar | 30 |  | Required | Not Null |
| City | Stores the city of the address section of the user. | Varchar | 15 |  | Required | Not Null |
| UserState | Stores the state of the address section of the user. | Varchar | 2 |  | Required | Not Null |
| ZipCode | Stores the zip code of the address section of the user. | Varchar | 5 |  | Required | Not Null |
| PhoneNumber | Stores the phone number of the user. | Varchar | 10 |  | Required | Not Null |
| EmailID | Stores the email ID of the user. | Varchar | 30 |  | Required | Not Null |
| UserType | Identifies the type of user logged in the portal. | Varchar | 1 |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **Doctor** | This table stores all the information of the doctors who are registered in the system. |  |  |  |  |  |
| Username | This is the primary key. This is the login username of the doctor. | Varchar | 20 | PK, FK | Required | Not Null |
| Specialization | Field of specialization of the doctor. | Varchar | 30 |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **Patient** | This table stores all the information of the patients. |  |  |  |  |  |
| Username | This is the primary key. This is the login username of the patient. | Varchar | 20 | PK, FK | Required | Not Null |
| isDisability | This field stores if the patient has disability or not.  1 – Yes  0 – No | Varchar | 1 |  | Required | Not Null |
| DisabilityDetails | Details of the disability, in case isDisability = 1. | Varchar | 50 |  | Not required | Nullable |
|  |  |  |  |  |  |  |
| **Receptionist** | This table stores all the information of the receptionist. |  |  |  |  |  |
| Username | This is the primary key. This is the login username of the receptionist. | Varchar | 20 | PK, FK | Required | Not Null |
| Schedule | This field stores the schedule of the receptionist. | Varchar | 30 |  | Required | Not Null |
| NoOfBillsGenerated | This field calculates the number of bills each receptionist generates. | Int |  |  | Not required | Nullable |
|  |  |  |  |  |  |  |
| **MedicalExam** | This is the master table which stores all the exams which the hospital offers. |  |  |  |  |  |
| ExamID | This is the primary key. | Varchar | 20 | PK | Required | Not Null |
| ExamName | This field stores the name of the exam. | Varchar | 30 |  | Required | Not Null |
| ExamDesc | This field stores the description of the exam. | Varchar | 300 |  | Required | Not Null |
| ExamCost | Stores the total cost of the exam without insurance. | Int |  |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **InsuranceDetails** | This is the master table which stores the coverage of insurance for exams offered. |  |  |  |  |  |
| InsuranceID | This is the primary key. | Varchar | 10 | PK | Required | Not Null |
| InsuranceName | Name of the insurance. | Varchar | 30 |  | Required | Not Null |
| InsuranceDesc | Description of the insurance. | Varchar | 330 |  | Required | Not Null |
| PercentageCoverageInstate | This field stores the percentage of instate coverage. | Int |  |  | Required | Not Null |
| PercentageCoverageOutstate | This field stores the percentage of outstate coverage. | Int |  |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **InsuranceLine** | This table stores the insurance which each patient has. |  |  |  |  |  |
| UserName | Foreign key from Patient (UserName). | Varchar | 20 | PK, FK | Required | Not Null |
| InsuranceID | Foreign key from InsuranceDetails (InsuranceID). | Varchar | 10 |  | Required | Not Null |
| StartDate | Start date of insurance. | Date |  |  | Required | Not Null |
| EndDate | End date of insurance. | Date |  |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **DoctorAvailability** | This table stores the availability of the doctors. |  |  |  |  |  |
| DoctorUserName | Foreign key from DoctorDetails (Username). | Varchar | 20 | PK, FK | Required | Not Null |
| DayOfTheWeek | Day of week the doctor is available. | Varchar | 10 |  | Required | Not Null |
| StartTime | Start time of the appointment. | Varchar | 10 |  | Required | Not Null |
| ActiveFlg | If the appointment is available or not.  1 – Available  0 – Not available | Varchar | 1 |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **Appointment** | This table stores the details of the appointment which the patient has booked. |  |  |  |  |  |
| AppointmentID | This is the primary key. | Int |  | PK | Required | Not Null |
| PatientUserName | Foreign key from Patient (UserName). | Varchar | 20 | FK | Required | Not Null |
| DoctorUserName | Foreign key from Doctor (UserName). | Varchar | 20 | FK | Required | Not Null |
| AppointmentDate | Date of appointment. | Varchar | 10 |  | Required | Not Null |
| AppointmentStartTime | Time of appointment. Foreign key from DoctorAvailability (StartTime) | Varchar | 10 |  | Required | Not Null |
| ReasonForAppointment | Why was the appointment scheduled. | Varchar | 150 |  | Required | Not Null |
| ActiveFlg | 1. Active appointment 2. Expired appointment | Varchar | 1 |  | Required | Not Null |
|  |  |  |  |  |  |  |
| **Diagnosis** | This table stores the details of exams conducted on the patient. |  |  |  |  |  |
| AppointmentID | Foreign key from Appointment (AppointmentID). | Int |  | PK, FK | Required | Not Null |
| ExamID | Foreign key from MedicalExam (ExamID). | Varchar | 20 | PK, FK | Required | Not Null |
| MedicalNotes | Special notes or readings which the doctor has to take. | Varchar | Max |  | Not required | Nullable |
|  |  |  |  |  |  |  |
| **BillDetails** | This table stores the details of the bills. |  |  |  |  |  |
| BillID | This is the primary key. | Int |  | PK | Required | Not Null |
| AppointmentID | Foreign key from Appointment (AppointmentID). | Int |  | FK | Required | Not Null |
| ExamID | Foreign key from Diagnosis (ExamID). | Varchar |  |  | Required | Not Null |
| ReceptionistUserName | Foreign key from Receptionist (UserName). | Varchar |  | FK | Required | Not Null |
| DateOfBill | Date of bill generation. | Varchar |  |  | Required | Not Null |
| AdditionalCost | Additional costs if any, else 0. | Int |  |  | Required | Not Null |
| Comments | Comments regarding the bill. | Varchar |  |  | Not Required | Nullable |
| TotalCost | Total costs of all exams, which the patient has to pay. | Int |  |  | Required | Not Null |

# DATABASE SYSTEM INFRASTRUCTURE

We have used the following tools to develop the database system infrastructure:

* Database system: SQL Server Management Studio
* Interface Design tool: MS Access

# SQL SCRIPT FOR CREATING AND INSERTING TABLE DATA

## CREATING TABLES IN SQL

**Users Table:**

CREATE TABLE Users (

UserName VARCHAR(20) NOT NULL,

UserPassword VARCHAR(20) NOT NULL,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(20) NOT NULL,

DateOfBirth DATE NOT NULL,

StreetNumber VARCHAR(5) NOT NULL,

StreetName VARCHAR(30) NOT NULL,

City VARCHAR(15) NOT NULL,

UserState VARCHAR(2) NOT NULL,

ZipCode VARCHAR(5) NOT NULL,

PhoneNumber VARCHAR (10) NOT NULL,

EmailID VARCHAR(30) NOT NULL,

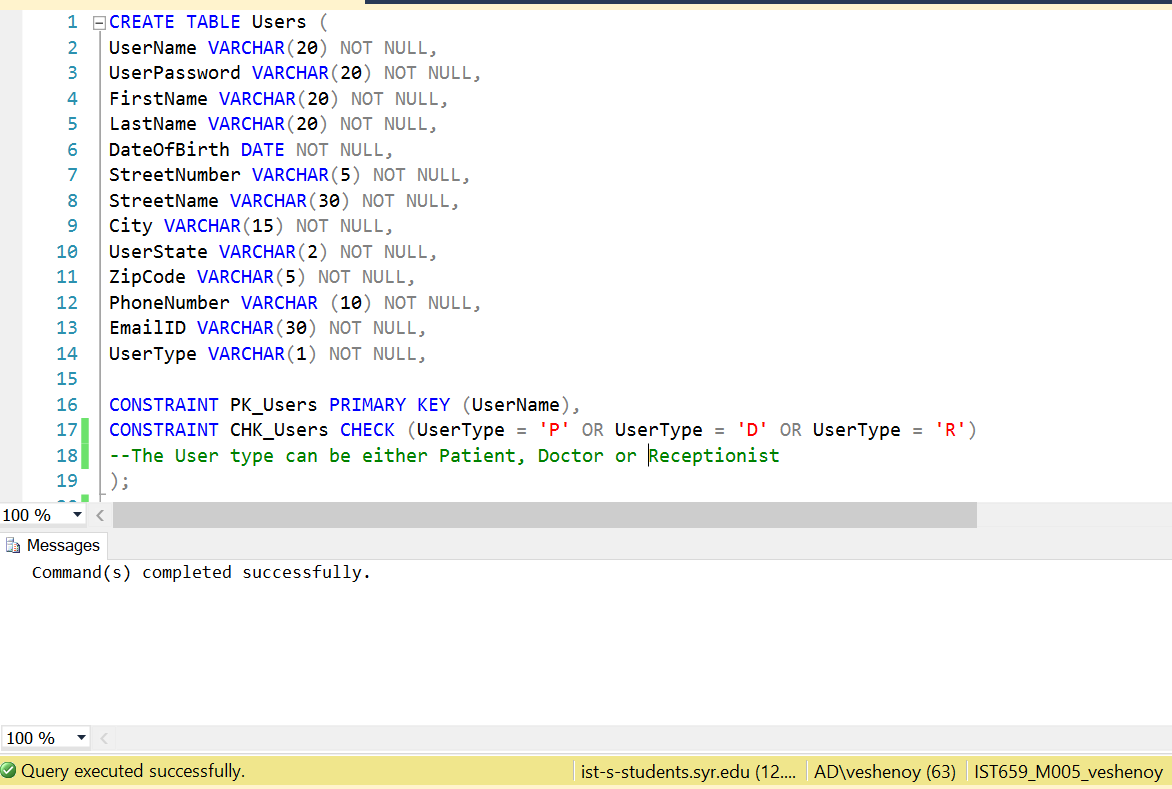
UserType VARCHAR(1) NOT NULL,

CONSTRAINT PK\_Users PRIMARY KEY (UserName),

CONSTRAINT CHK\_Users CHECK (UserType = 'P' OR UserType = 'D' OR UserType = 'R')

--The User type can be either Patient, Doctor or Receptionist

);



**Doctor Table:**

CREATE TABLE Doctor (

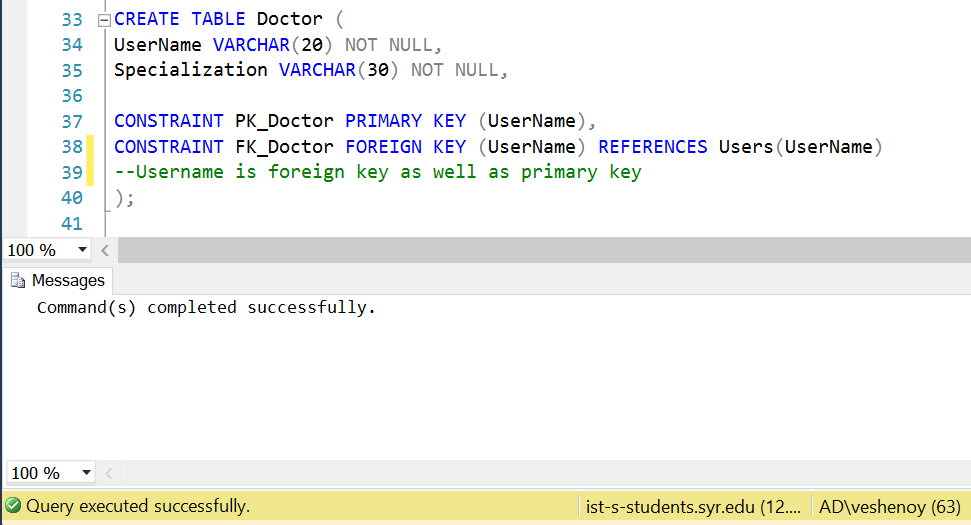
UserName VARCHAR(20) NOT NULL,

Specialization VARCHAR(30) NOT NULL,

CONSTRAINT PK\_Doctor PRIMARY KEY (UserName),

CONSTRAINT FK\_Doctor FOREIGN KEY (UserName) REFERENCES Users(UserName)

);



**Patient Table:**

CREATE TABLE Patient (

UserName VARCHAR(20) NOT NULL,

isDisability VARCHAR(1) NOT NULL,

DisabilityDetails VARCHAR(50),

CONSTRAINT PK\_Patient PRIMARY KEY (UserName),

CONSTRAINT FK\_Patient FOREIGN KEY (UserName) REFERENCES Users(UserName),

CONSTRAINT CHK\_Patient\_isDisability CHECK (isDisability = '1' OR isDisability = '0')

--The CHECK constraint is enforced as the patient may or may not have a disability

);



**Receptionist Table:**

CREATE TABLE Receptionist (

UserName VARCHAR(20) NOT NULL,

Schedule VARCHAR(30) NOT NULL,

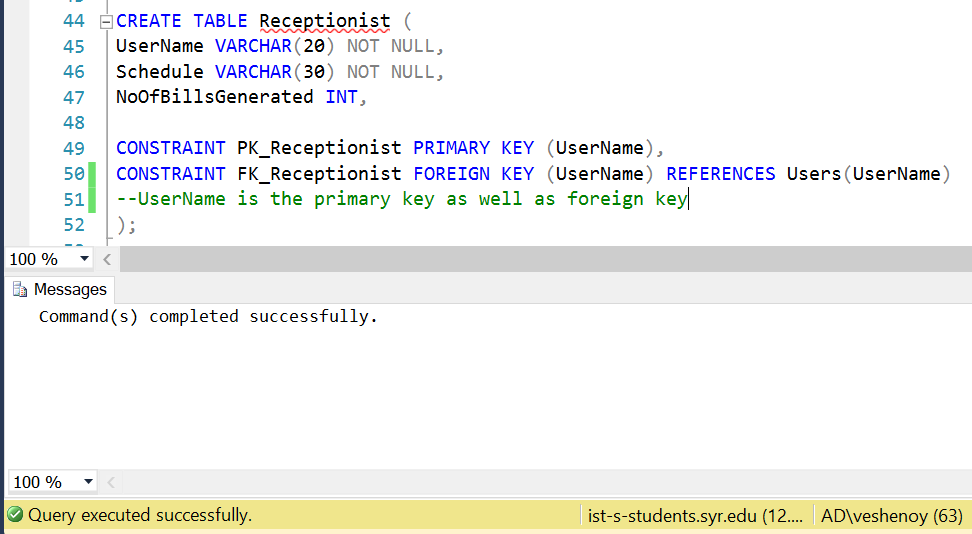
NoOfBillsGenerated INT,

CONSTRAINT PK\_Receptionist PRIMARY KEY (UserName),

CONSTRAINT FK\_Receptionist FOREIGN KEY (UserName) REFERENCES Users(UserName)

--UserName is the primary key as well as foreign key

);



**MedicalExam Table:**

CREATE TABLE MedicalExam (

ExamID VARCHAR(20) NOT NULL,

ExamName VARCHAR(30) NOT NULL,

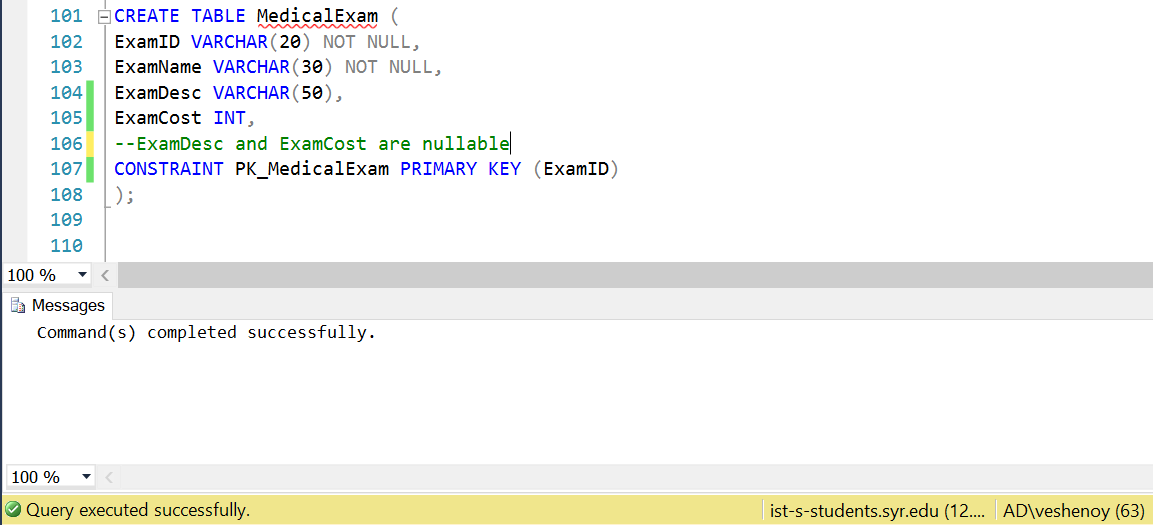
ExamDesc VARCHAR(50),

ExamCost INT,

--ExamDesc and ExamCost are nullable

CONSTRAINT PK\_MedicalExam PRIMARY KEY (ExamID)

);



**InsuranceDetails Table:**

CREATE TABLE InsuranceDetails (

InsuranceID VARCHAR(10) NOT NULL,

InsuranceName VARCHAR(30),

InsuranceDesc VARCHAR(330),

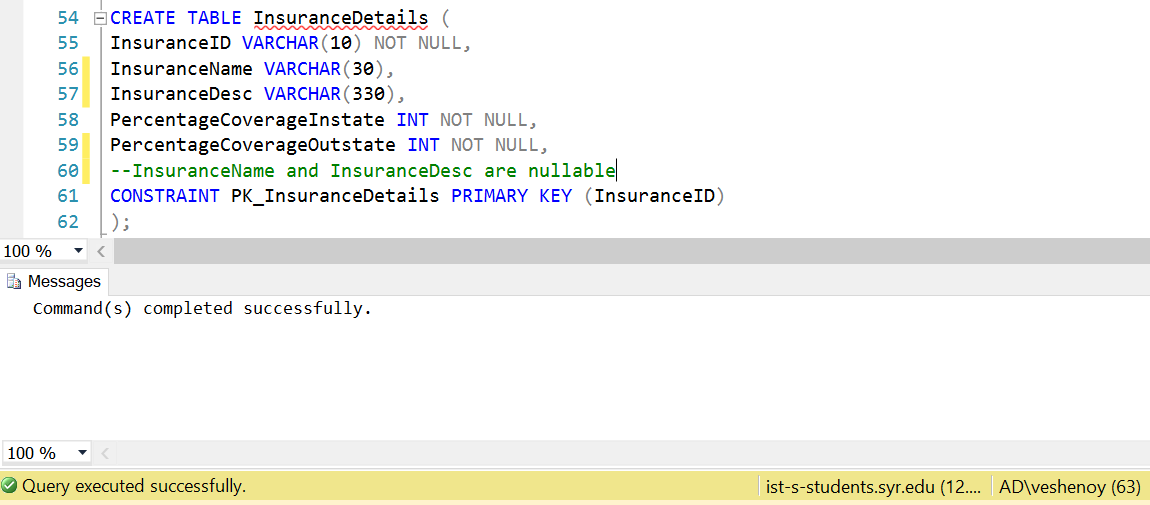
PercentageCoverageInstate INT NOT NULL,

PercentageCoverageOutstate INT NOT NULL,

--InsuranceName and InsuranceDesc are nullable

CONSTRAINT PK\_InsuranceDetails PRIMARY KEY (InsuranceID)

);



**InsuranceLine Table:**

CREATE TABLE InsuranceLine (

UserName VARCHAR(20) NOT NULL,

InsuranceID VARCHAR(10) NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

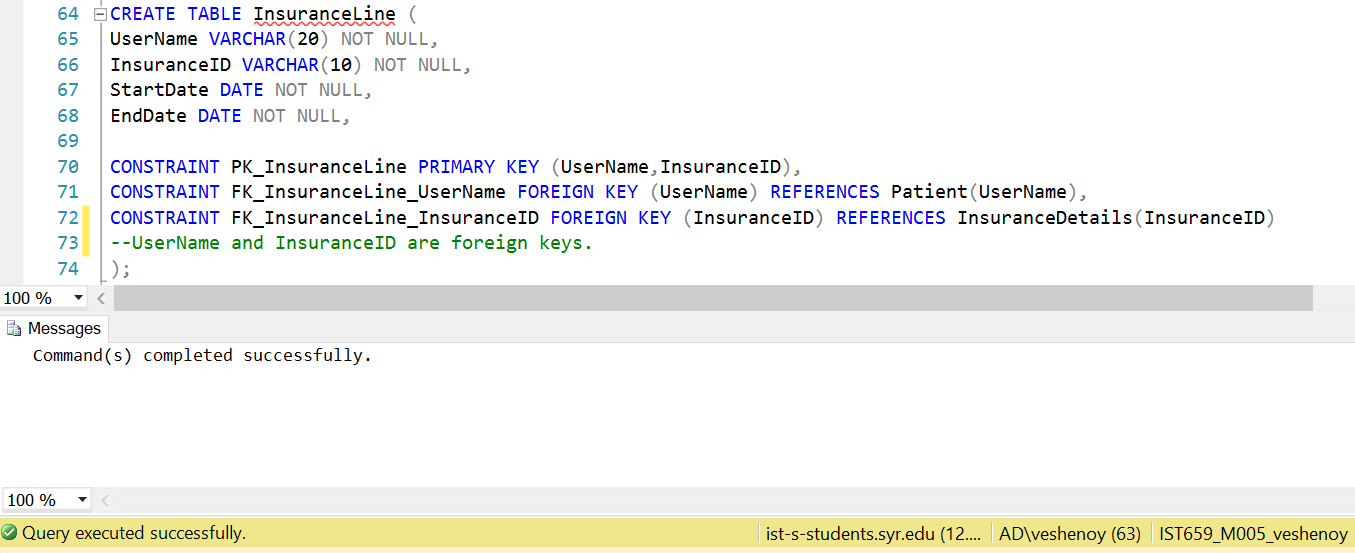
CONSTRAINT PK\_InsuranceLine PRIMARY KEY (UserName,InsuranceID),

CONSTRAINT FK\_InsuranceLine\_UserName FOREIGN KEY (UserName) REFERENCES Patient(UserName),

CONSTRAINT FK\_InsuranceLine\_InsuranceID FOREIGN KEY (InsuranceID) REFERENCES InsuranceDetails(InsuranceID)

--UserName and InsuranceID are foreign keys.

);



**DoctorAvailability Table:**

CREATE TABLE DoctorAvailability (

DoctorUserName VARCHAR(20) NOT NULL,

DayoftheWeek VARCHAR(10) NOT NULL,

StartTime TIME NOT NULL,

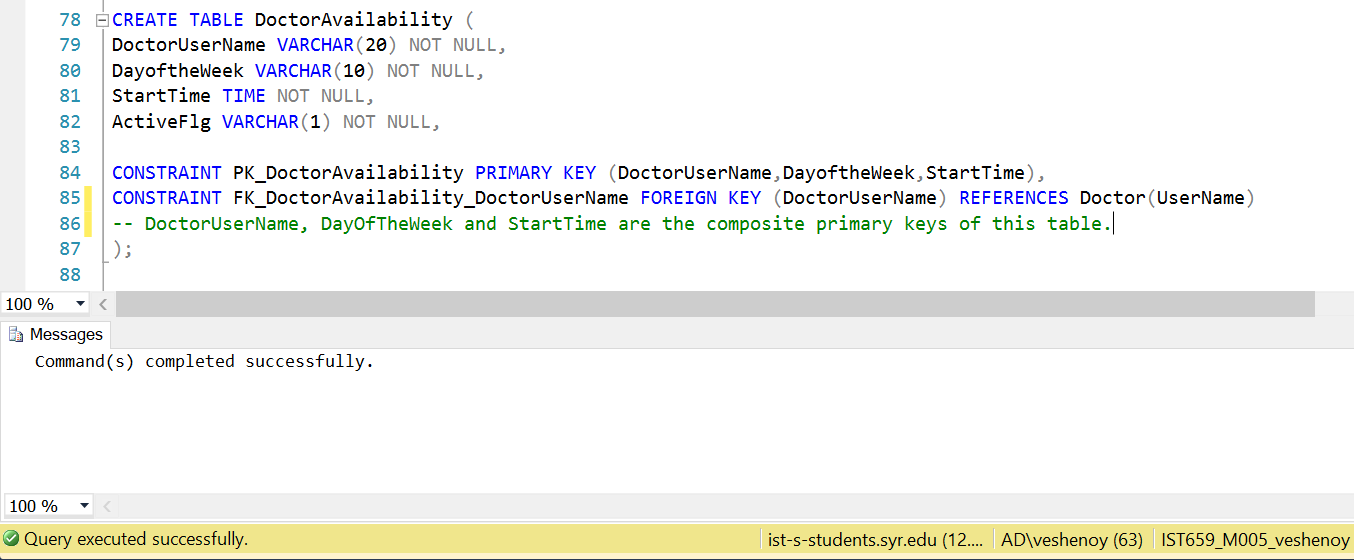
ActiveFlg VARCHAR(1) NOT NULL,

CONSTRAINT PK\_DoctorAvailability PRIMARY KEY (DoctorUserName,DayoftheWeek,StartTime),

CONSTRAINT FK\_DoctorAvailability\_DoctorUserName FOREIGN KEY (DoctorUserName) REFERENCES Doctor(UserName)

-- DoctorUserName, DayOfTheWeek and StartTime are the composite primary keys of this table.

);



**Appointment Table:**

CREATE TABLE Appointment (

AppointmentID int NOT NULL,

PatientUserName varchar(20) NOT NULL,

DoctorUserName varchar(20) NOT NULL,

AppointmentDate varchar(10) NOT NULL,

AppointmentStartTime varchar (10) NOT NULL,

ReasonForAppointment varchar(150)NOT NULL,

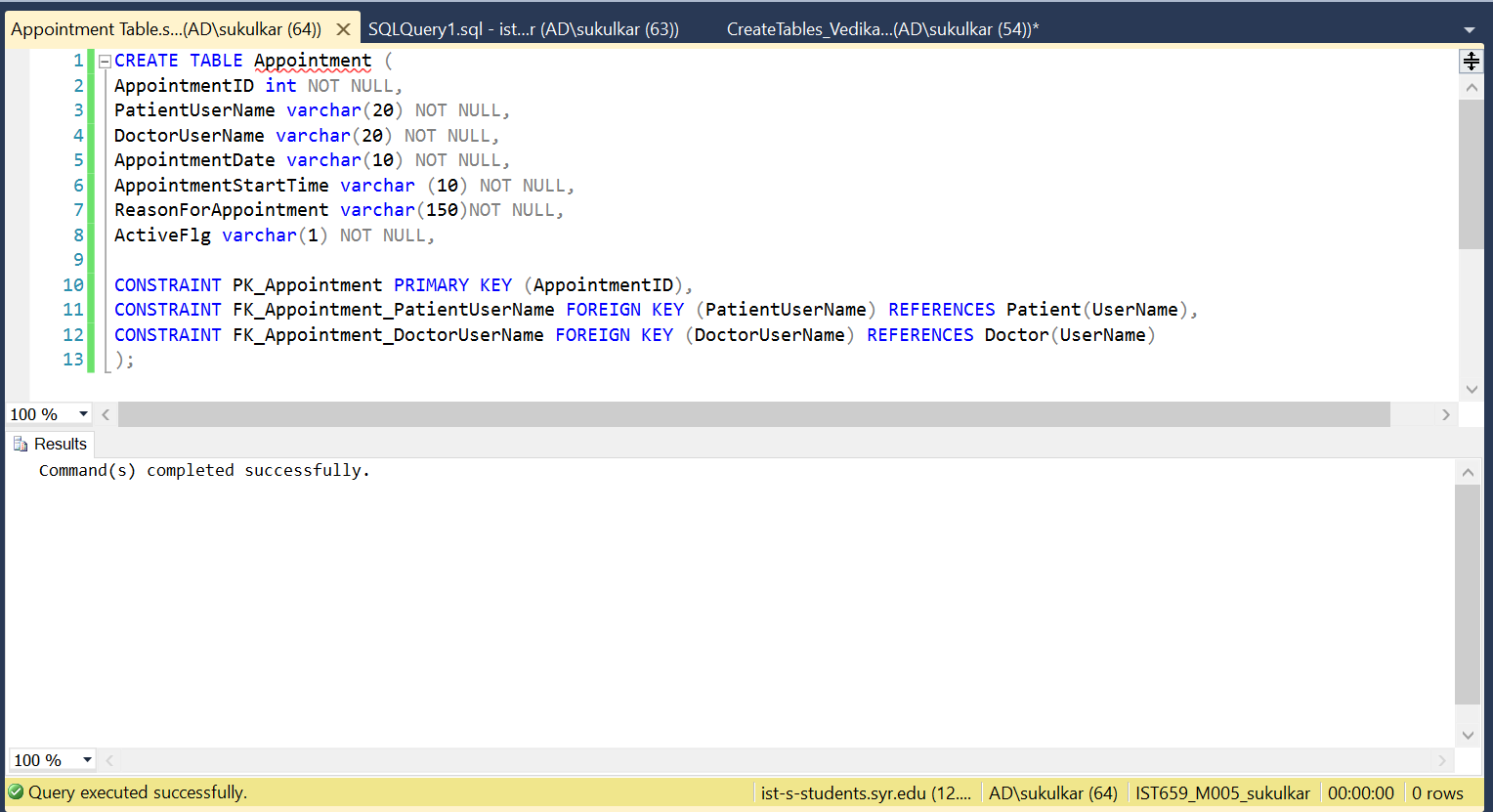
ActiveFlg varchar(1) NOT NULL,

CONSTRAINT PK\_Appointment PRIMARY KEY (AppointmentID),

CONSTRAINT FK\_Appointment\_PatientUserName FOREIGN KEY (PatientUserName) REFERENCES Patient(UserName),

CONSTRAINT FK\_Appointment\_DoctorUserName FOREIGN KEY (DoctorUserName) REFERENCES Doctor(UserName)

);



**Diagnosis Table:**

CREATE TABLE Diagnosis (

AppointmentID INT NOT NULL,

ExamID VARCHAR(20) NOT NULL,

MedicalNotes VARCHAR(150) NOT NULL,

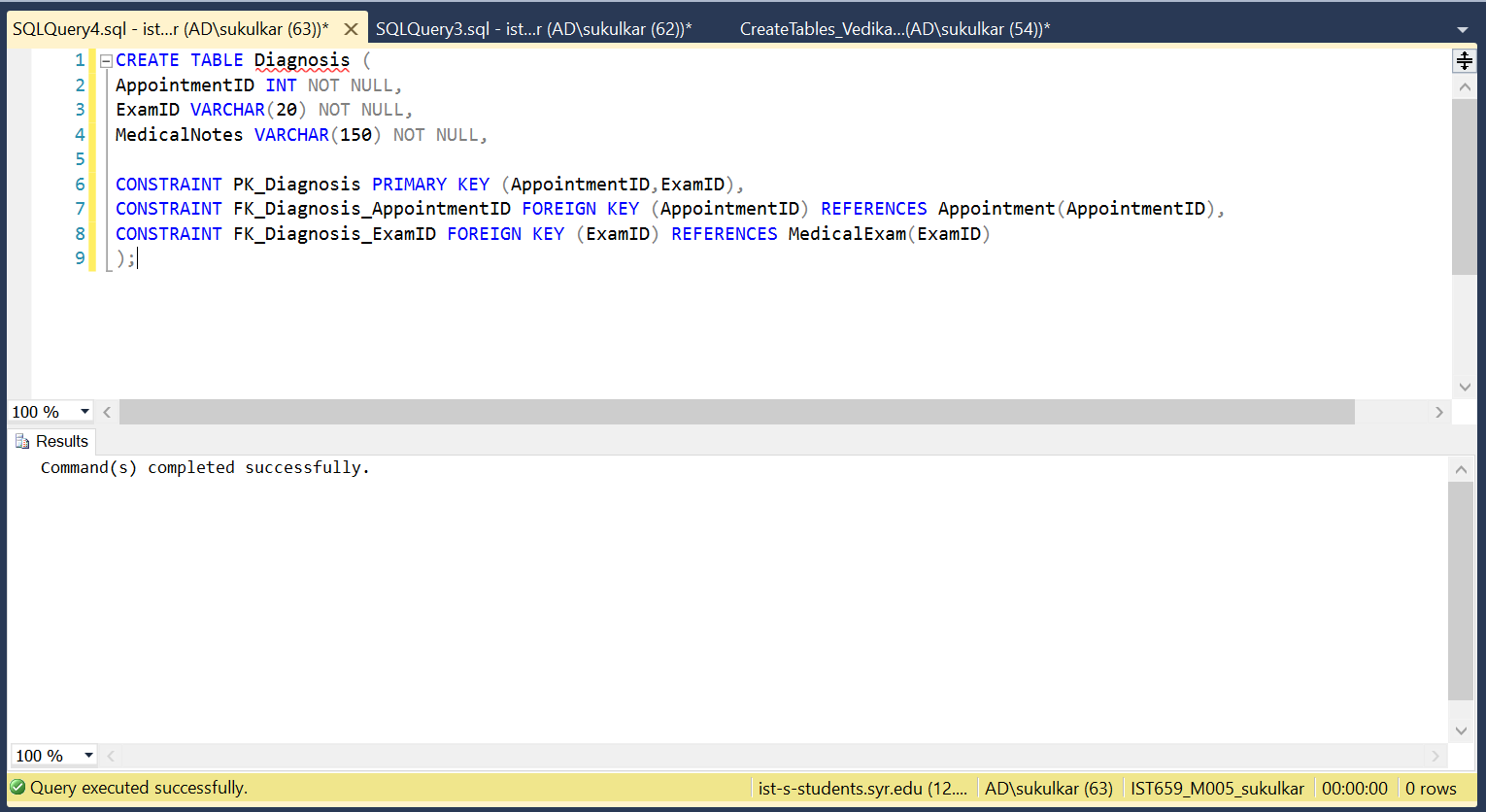
CONSTRAINT PK\_Diagnosis PRIMARY KEY (AppointmentID,ExamID),

CONSTRAINT FK\_Diagnosis\_AppointmentID FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID),

CONSTRAINT FK\_Diagnosis\_ExamID FOREIGN KEY (ExamID) REFERENCES MedicalExam(ExamID)

--ExamID and AppointmentID are the foreign keys.

);



**BillDetails Table:**

CREATE TABLE BillDetails (

BillID INT NOT NULL,

AppointmentID INT NOT NULL,

ExamID VARCHAR(20) NOT NULL,

ReceptionistUserName VARCHAR(20) NOT NULL,

DateOfBill DATE NOT NULL,

AdditionalCost INT NOT NULL,

Comments VARCHAR(50),

TotalCost INT NOT NULL,

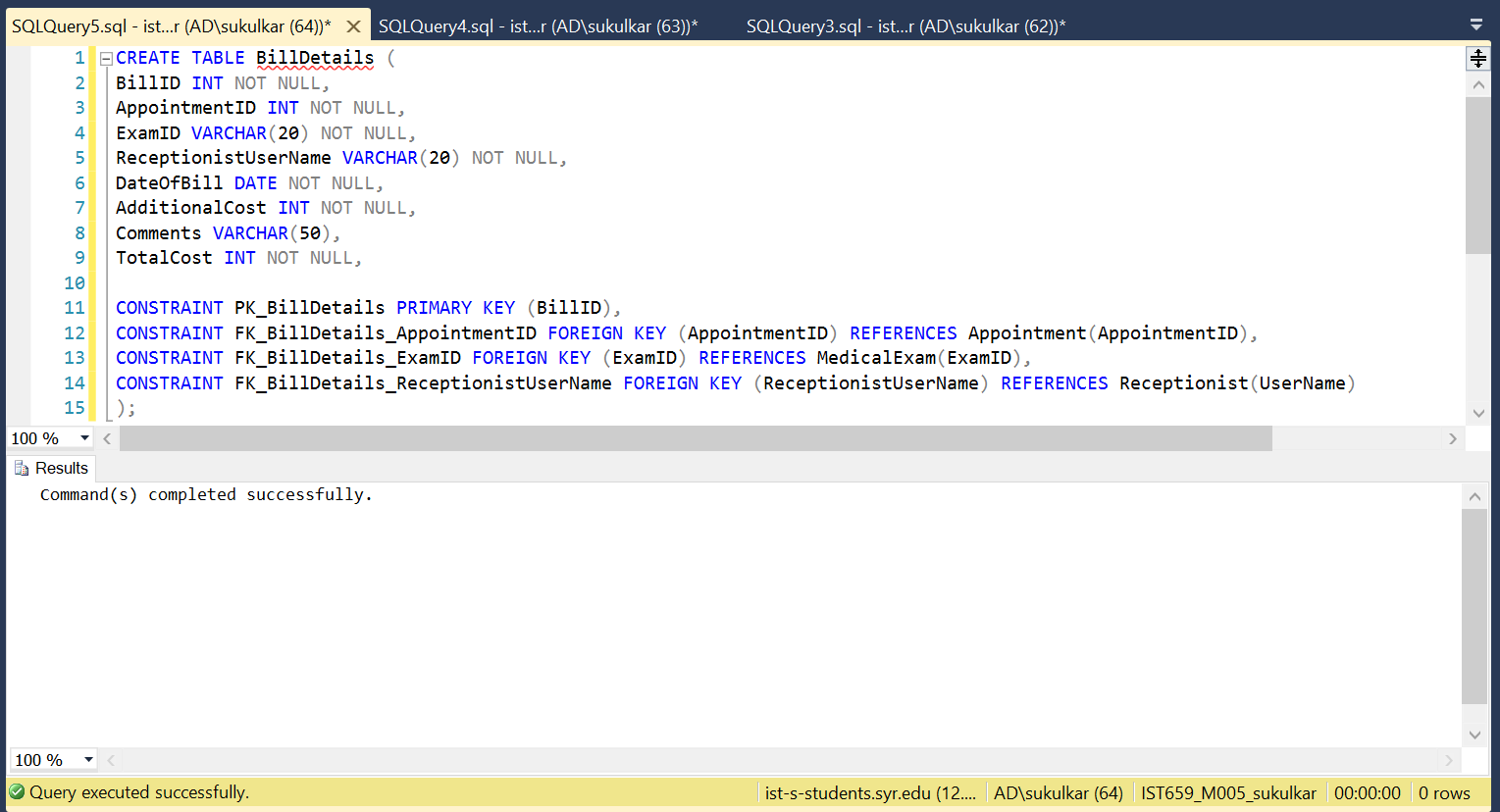
CONSTRAINT PK\_BillDetails PRIMARY KEY (BillID),

CONSTRAINT FK\_BillDetails\_AppointmentID FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID),

CONSTRAINT FK\_BillDetails\_ExamID FOREIGN KEY (ExamID) REFERENCES MedicalExam(ExamID),

CONSTRAINT FK\_BillDetails\_ReceptionistUserName FOREIGN KEY (ReceptionistUserName) REFERENCES Receptionist(UserName)

);



## INSERTING TABLES IN SQL

**Users Table:**

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('sudkul','apple@123','Sudhanshu','Kulkarni','1987-10-21','143','Avondale Place','Syracuse','NY','13210','3152647895','sudkul@gmail.com','D');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('vedshe','mango@123','Vedika','Shenoy','1991-06-06','420','Westcott Street','Syracuse','NY','13210','3125978630','vedshe@gmail.com','D');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('amitdj','pine@123','Amit','Jadhav','1978-03-28','141','Avondale Place','Syracuse','NY','13210','3126547890','amitdj@gmail.com','D');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('manded','orange@123','Manan', 'Dedhia','1965-06-30','422','Westcott Street','Syracuse','NY','13210','3251469873','manded@gmail.com','D');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('advkam','pear@123','Advaith','Kamath','1980-07-09','421','Westcott Street','Syracuse','NY','13210','3698563214','advkam@gmail.com','D');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('dhrbht','potato@123','Dhruv','Bhatti','1989-03-23','141','Avondale Place','Syracuse','NY','13210','3985463217','dhrbht@gmail.com','P');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('anayak','cabbage@123','Anmol','Nayak','1972-03-20','133','Avondale Place','Syracuse','NY','13210','3986542217','anayak@gmail.com','P');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('mstewart','brocolli@123','Martha','Stewart','1989-04-13','101','Westcott Street','Syracuse','NY','13210','3467542217','mstewart@gmail.com','P');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('jstevens','pepper@123','Jacob','Stevens','1990-06-25','189','James Street','Syracuse','NY','13210','3467548643','jstevens@gmail.com','P');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('amendosa','spinach@123','Alwin','Mendosa','1970-02-15','155','Euclid Ave','Syracuse','NY','13210','3467689743','amendosa@gmail.com','P');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('kruvor','slave@123','Kruti','Vora','1959-06-18','562','Westcott Street','Syracuse','NY','13210','9875641236','kruvor@gmail.com','R');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('eclarke','peanut@123','Emilia','Clarke','1999-01-31','500','Euclid Ave','Syracuse','NY','13210','9876432236','eclarke@gmail.com','R');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('rpatt','cashew@123','Robert','Pattinson','1963-03-30','800','Comstock Ave','Syracuse','NY','13210','9876332236','rpatt@gmail.com','R');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

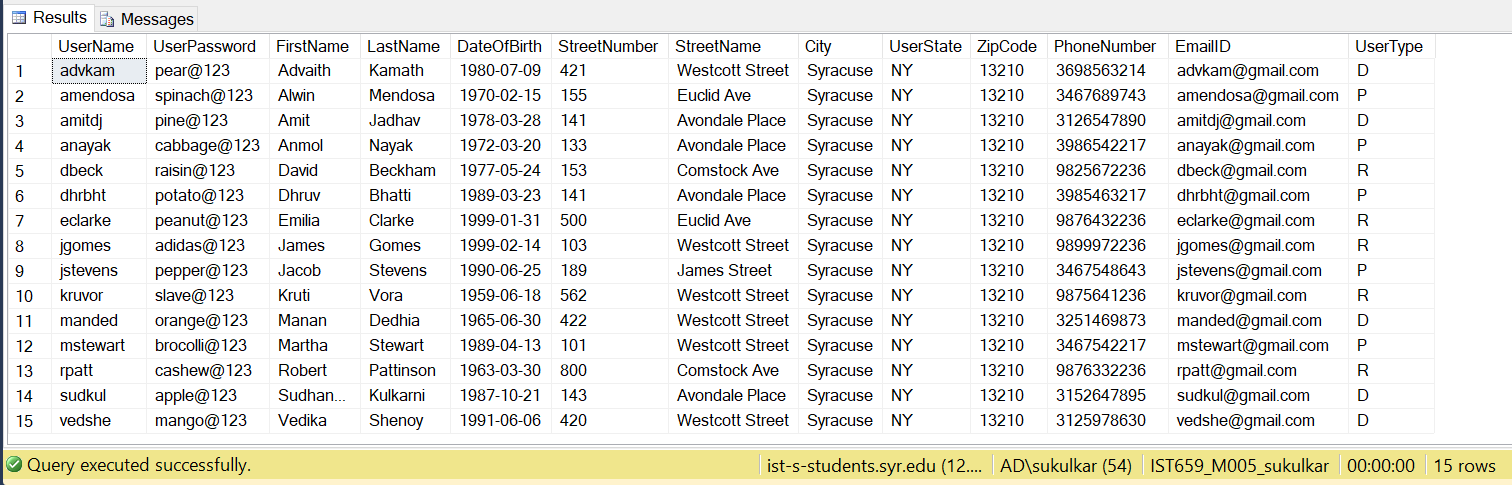
('dbeck','raisin@123','David','Beckham','1977-05-24','153','Comstock Ave','Syracuse','NY','13210','9825672236','dbeck@gmail.com','R');

INSERT INTO [dbo].[Users]

(UserName,UserPassword,FirstName,LastName,DateOfBirth,StreetNumber,StreetName,City,UserState,ZipCode,PhoneNumber,EmailID,UserType)

VALUES

('jgomes','adidas@123','James','Gomes','1999-02-14','103','Westcott Street','Syracuse','NY','13210','9899972236','jgomes@gmail.com','R');



**Doctor Table:**

INSERT INTO [dbo].[Doctor] (UserName,Specialization)

VALUES ('sudkul','Cardiologist');

INSERT INTO [dbo].[Doctor] (UserName,Specialization)

VALUES ('vedshe','Psychiatrist');

INSERT INTO [dbo].[Doctor] (UserName,Specialization)

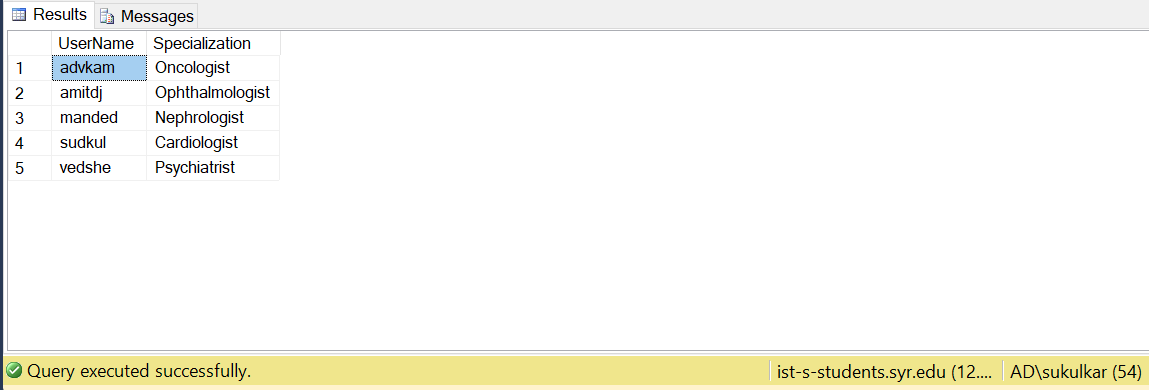
VALUES ('amitdj','Opthamologist');

INSERT INTO [dbo].[Doctor] (UserName,Specialization)

VALUES ('manded','Nephrologist');

INSERT INTO [dbo].[Doctor] (UserName,Specialization)

VALUES ('advkam','Oncologist');



**Patient Table:**

INSERT INTO [dbo].[Patient] (UserName, isDisability)

VALUES ('dhrbht','0');

INSERT INTO [dbo].[Patient] (UserName, isDisability)

VALUES ('anayak','0');

INSERT INTO [dbo].[Patient] (UserName, isDisability, DisabilityDetails)

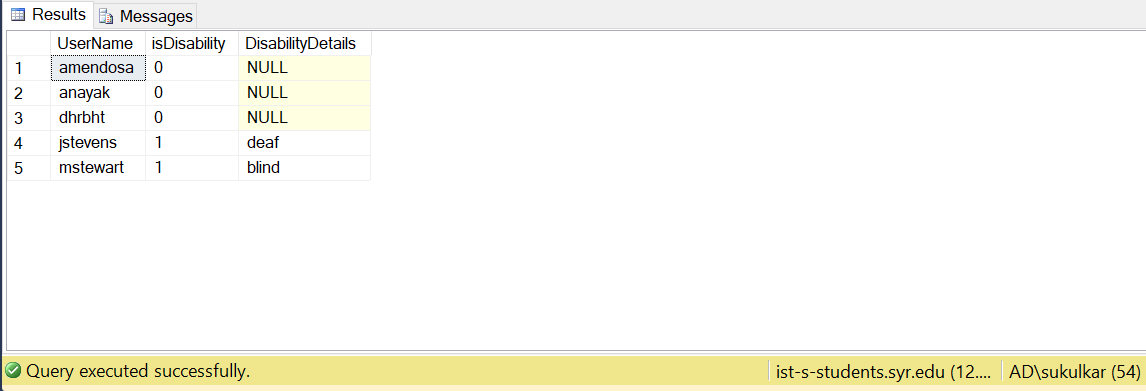
VALUES ('mstewart','1','blind');

INSERT INTO [dbo].[Patient] (UserName, isDisability, DisabilityDetails)

VALUES ('jstevens','1','deaf');

INSERT INTO [dbo].[Patient] (UserName, isDisability)

VALUES ('amendosa','0');



**Receptionist Table:**

INSERT INTO [dbo].[Receptionist] (UserName,Schedule,NoOfBillsGenerated)

VALUES ('kruvor','8am-4pm','');

INSERT INTO [dbo].[Receptionist] (UserName,Schedule,NoOfBillsGenerated)

VALUES ('eclarke','8am-4pm','');

INSERT INTO [dbo].[Receptionist] (UserName,Schedule,NoOfBillsGenerated)

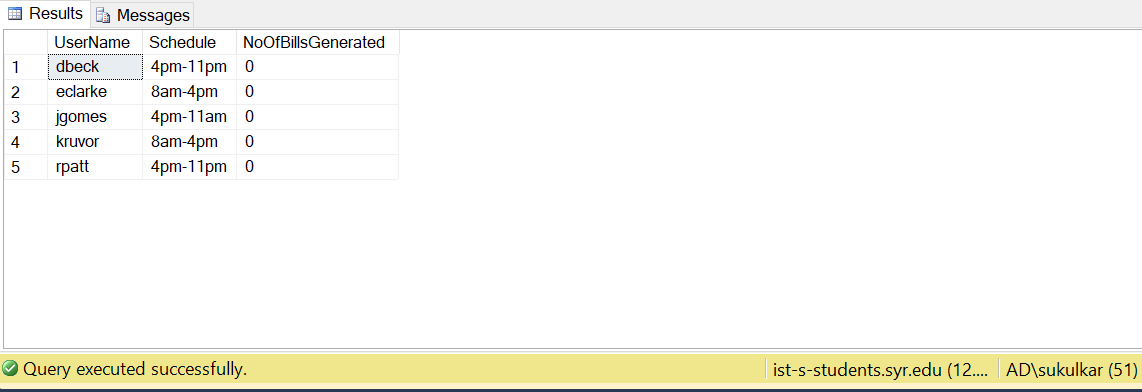
VALUES ('rpatt','4pm-11pm','');

INSERT INTO [dbo].[Receptionist] (UserName,Schedule,NoOfBillsGenerated)

VALUES ('dbeck','4pm-11pm','');

INSERT INTO [dbo].[Receptionist] (UserName,Schedule,NoOfBillsGenerated)

VALUES ('jgomes','4pm-11am','');



**MedicalExam Table:**

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('101','Blood Exam','A blood test is a laboratory analysis performed on a blood sample that is usually extracted from a vein in the arm using a hypodermic needle, or via fingerprick.','50')

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('102','X-Ray','X-ray is a quick and simple imaging test that can spot problems in your bones, teeth, chest and more.','80')

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('103','CT Scan','A computerized tomography (CT) scan combines a series of X-ray images taken from different angles around your body and uses computer processing to create cross-sectional images (slices) of the bones, blood vessels and soft tissues inside your body.',200)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('104','MRI Scan','Magnetic resonance imaging is a medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body in both health and disease.',350)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('105','Urine Exam','The tests detect and/or measure several substances in the urine, such as byproducts of normal and abnormal metabolism, cells, cellular fragments, and bacteria',100)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('106','Eye Exam','An eye examination is a series of tests performed by an ophthalmologist, optometrist, or orthoptist, optician, assessing vision and ability to focus on and discern objects, as well as other tests and examinations pertaining to the eyes.',100)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('107', 'Contact Lens Exam','Your eye doctor will perform special tests during a contact lens exam to evaluate your vision with contacts.',150)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

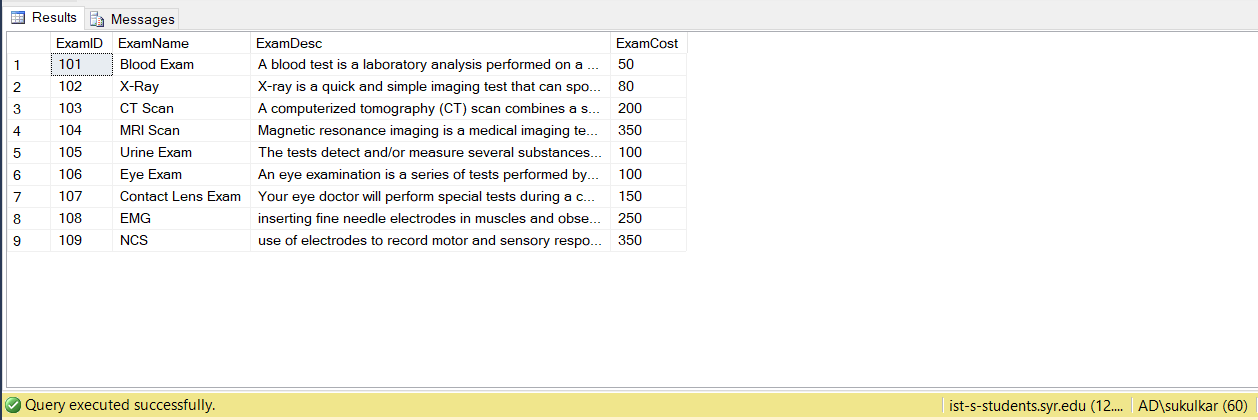
VALUES

('108','EMG','inserting fine needle electrodes in muscles and observing the recorded motor unit potentials when the muscles are activated to help distinguish whether weakness is due to muscle or nerve dysfunction',250)

INSERT INTO [dbo].[MedicalExam] (ExamID,ExamName,ExamDesc,ExamCost)

VALUES

('109','NCS','use of electrodes to record motor and sensory responses that are propagated by electrical stimuli. This test can help distinguish location of a nervous system lesion',350)



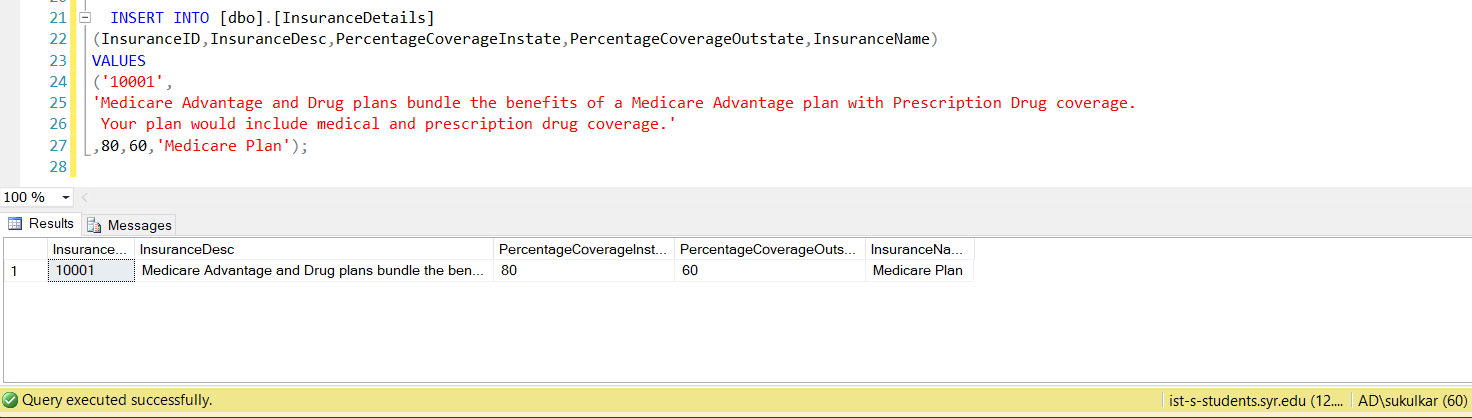
**InsuranceDetails Table:**

INSERT INTO [dbo].[InsuranceDetails] (InsuranceID,InsuranceDesc,PercentageCoverageInstate,PercentageCoverageOutstate,InsuranceName)

VALUES

('10001',

'Medicare Advantage and Drug plans bundle the benefits of a Medicare Advantage plan with Prescription Drug coverage. Your plan would include medical and prescription drug coverage.',80,60,'Medicare Plan');



**InsuranceLine Table:**

INSERT INTO [dbo].[InsuranceLine] (UserName,InsuranceID,StartDate,EndDate)

VALUES ('amendosa','10001',GETDATE(),(SELECT DATEADD(MONTH, 8, GETDATE())))

INSERT INTO [dbo].[InsuranceLine] (UserName,InsuranceID,StartDate,EndDate)

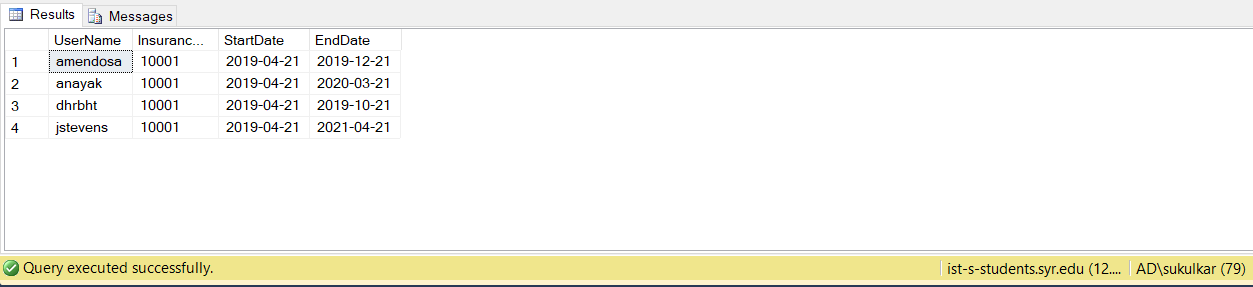
VALUES ('anayak','10001',GETDATE(),(SELECT DATEADD(month, 11, GETDATE())))

INSERT INTO [dbo].[InsuranceLine] (UserName,InsuranceID,StartDate,EndDate)

VALUES ('dhrbht','10001',GETDATE(),(SELECT DATEADD(MONTH, 6, GETDATE())))

INSERT INTO [dbo].[InsuranceLine] (UserName,InsuranceID,StartDate,EndDate)

VALUES ('jstevens','10001',GETDATE(),(SELECT DATEADD(YEAR, 2, GETDATE())))



**DoctorAvailability Table:**

In order to add doctor availability, we have added While loop for each doctor and each day of the week.

DECLARE @i int = 12

WHILE @i < 17

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('advkam','Thursday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 11

WHILE @i < 15

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('advkam','Monday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 9

WHILE @i < 15

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('advkam','Wednesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('amitdj','Tuesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('amitdj','Wednesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('amitdj','Thursday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('amitdj','Friday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('manded','Friday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 11

WHILE @i < 17

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('manded','Monday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 10

WHILE @i < 16

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('manded','Wedmesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('manded','Tuesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 9

WHILE @i < 15

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('sudkul','Monday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 9

WHILE @i < 15

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('sudkul','Tuesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 12

WHILE @i < 18

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('sudkul','Wednesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 12

WHILE @i < 18

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('sudkul','Thursday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 21

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('sudkul','Friday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 9

WHILE @i < 14

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('vedshe','Monday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 9

WHILE @i < 14

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('vedshe','Tuesday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('vedshe','Thursday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END

DECLARE @i int = 14

WHILE @i < 19

BEGIN

INSERT INTO [dbo].[DoctorAvailability]

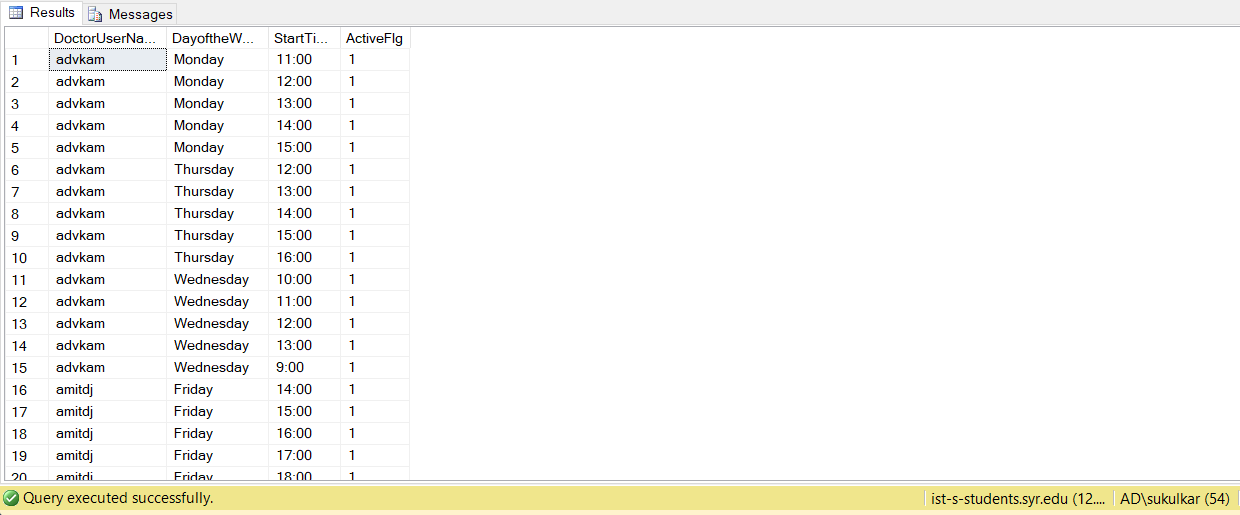
(DoctorUserName,DayoftheWeek,StartTime,ActiveFlg)

VALUES

('vedshe','Friday',CONVERT(varchar,@i)+':00',1)

SET @i = @i + 1

END



**Appointment Table:**

INSERT INTO [dbo].[Appointment]

(AppointmentID,PatientUserName,DoctorUserName,AppointmentDate,AppointmentStartTime,ReasonForAppointment,ActiveFlg)

VALUES ('1','dhrbht','sudkul','Monday','12:00','Fever','1');

INSERT INTO [dbo].[Appointment]

(AppointmentID,PatientUserName,DoctorUserName,AppointmentDate,AppointmentStartTime,ReasonForAppointment,ActiveFlg)

VALUES ('2','anayak','sudkul','Monday','11:00','Heart Attack','1');

INSERT INTO [dbo].[Appointment]

(AppointmentID,PatientUserName,DoctorUserName,AppointmentDate,AppointmentStartTime,ReasonForAppointment,ActiveFlg)

VALUES ('3','amendosa','vedshe','Monday','9:00','Trauma','1');

INSERT INTO [dbo].[Appointment]

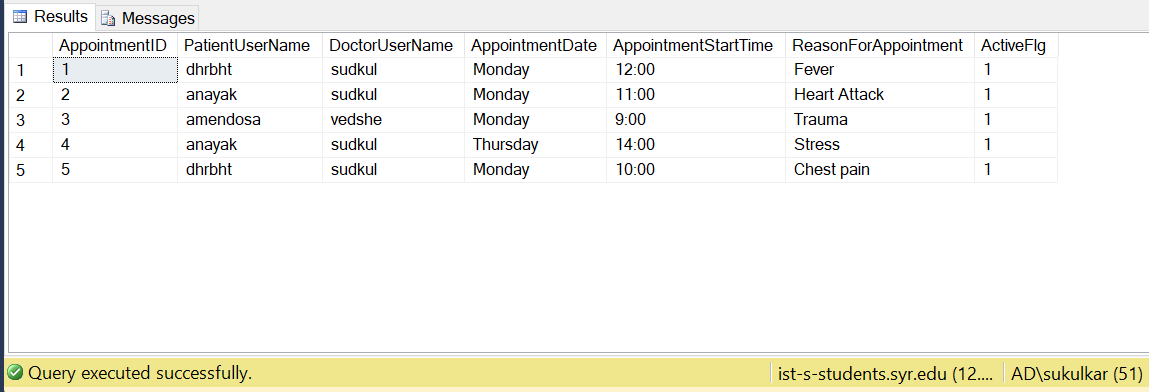
(AppointmentID,PatientUserName,DoctorUserName,AppointmentDate,AppointmentStartTime,ReasonForAppointment,ActiveFlg)

VALUES ('4','anayak','sudkul','Thursday','14:00','Stress','1');

INSERT INTO [dbo].[Appointment]

(AppointmentID,PatientUserName,DoctorUserName,AppointmentDate,AppointmentStartTime,ReasonForAppointment,ActiveFlg)

VALUES ('5','dhrbht','sudkul','Monday','10:00','Chest pain','1');



**Diagnosis Table:**

INSERT INTO [dbo].[Diagnosis] (AppointmentID, ExamID, MedicalNotes)

VALUES ('1','101','Normal');

INSERT INTO [dbo].[Diagnosis] (AppointmentID, ExamID, MedicalNotes)

VALUES ('2','102','Normal');

INSERT INTO [dbo].[Diagnosis] (AppointmentID, ExamID, MedicalNotes)

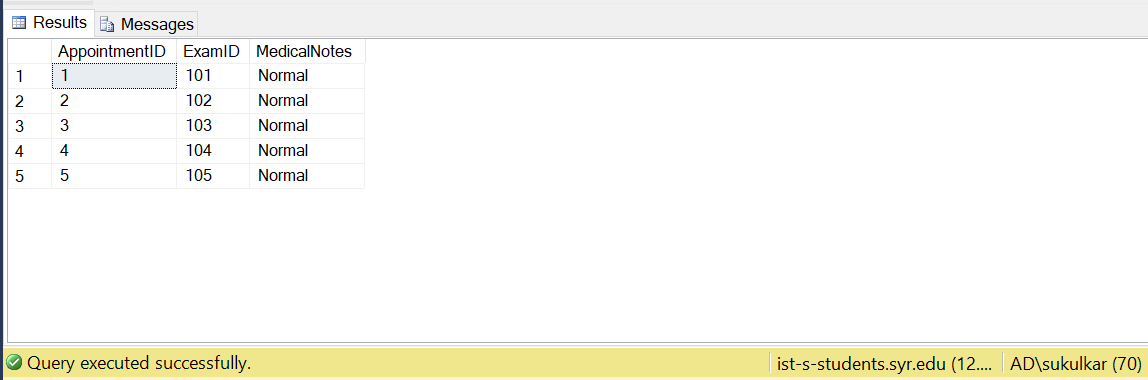
VALUES ('3','103','Normal');

INSERT INTO [dbo].[Diagnosis] (AppointmentID, ExamID, MedicalNotes)

VALUES ('4','104','Normal');

INSERT INTO [dbo].[Diagnosis] (AppointmentID, ExamID, MedicalNotes)

VALUES ('5','105','Normal');



**BillDetails Table:**

INSERT INTO BillDetails

(BillID, AppointmentID,ExamID,ReceptionistUserName, DateOfBill, AdditionalCost, Comments ,TotalCost)

VALUES ('1','1','101','kruvor','Apr 24 2019','20','None','50');

INSERT INTO BillDetails

(BillID, AppointmentID,ExamID,ReceptionistUserName, DateOfBill, AdditionalCost, Comments ,TotalCost)

VALUES ('2','2','102','kruvor','Apr 24 2019','10','None','100');

INSERT INTO BillDetails

(BillID, AppointmentID,ExamID,ReceptionistUserName, DateOfBill, AdditionalCost, Comments ,TotalCost)

VALUES ('3','3','103','kruvor','Apr 24 2019','20','None','50');

INSERT INTO BillDetails

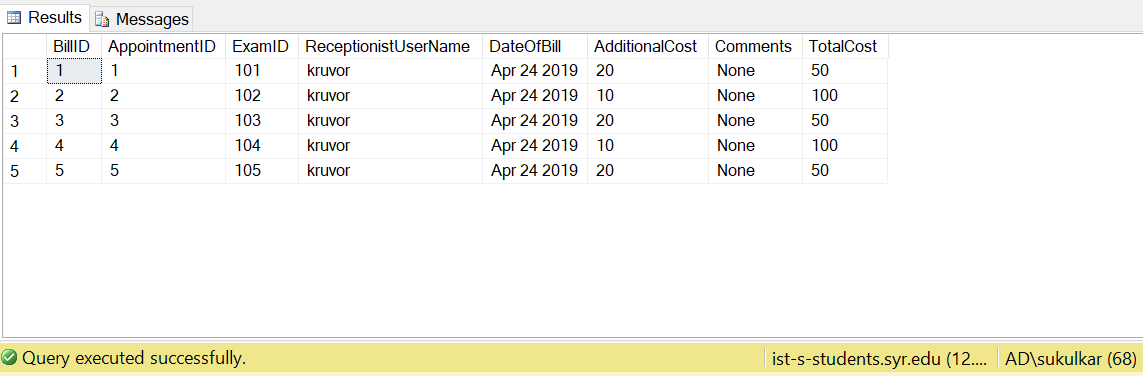
(BillID, AppointmentID,ExamID,ReceptionistUserName, DateOfBill, AdditionalCost, Comments ,TotalCost)

VALUES ('4','4','104','kruvor','Apr 24 2019','10','None','100');

INSERT INTO BillDetails

(BillID, AppointmentID,ExamID,ReceptionistUserName, DateOfBill, AdditionalCost, Comments ,TotalCost)

VALUES ('5','5','105','kruvor','Apr 24 2019','20','None','50');



## TRIGGERS

1. **After Insert Trigger for Patient appointment upon Bill Generation:**

CREATE TRIGGER [dbo].[AppointmentFlag] ON [dbo].[BillDetails]

AFTER INSERT

AS

BEGIN

UPDATE Appointment

SET ActiveFlg = '0'

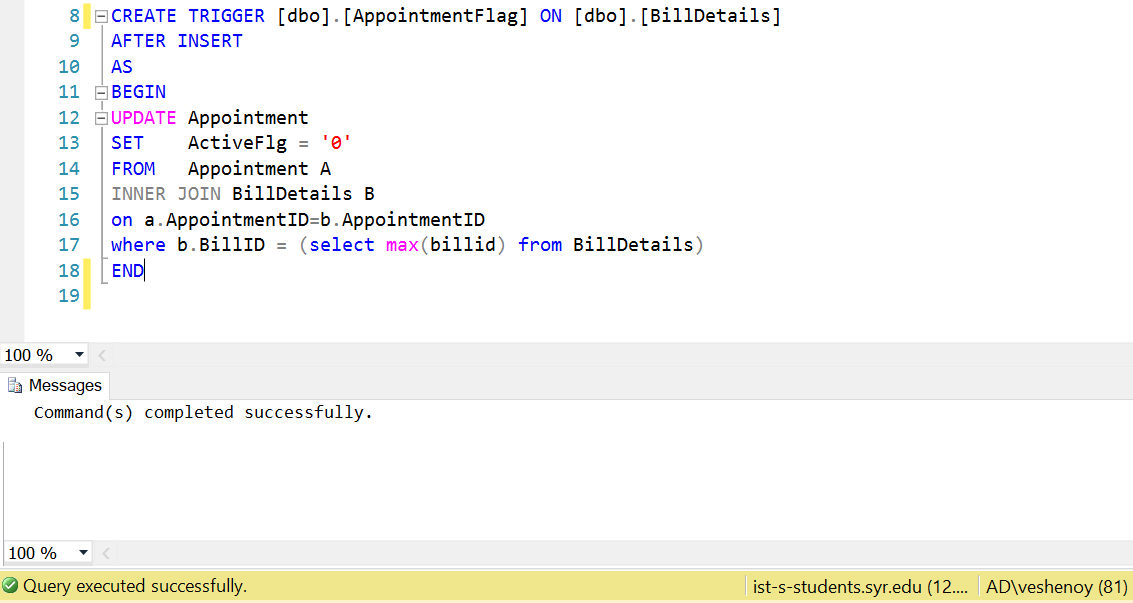
FROM Appointment A

INNER JOIN BillDetails B

on a.AppointmentID=b.AppointmentID

where b.BillID = (select max(billid) from BillDetails)

END



1. **After Insert Trigger for Doctor Availability upon Bill Generation:**

CREATE TRIGGER [dbo].[DoctorAppointmentFlag] ON [dbo].[BillDetails]

AFTER INSERT

AS

BEGIN

UPDATE DoctorAvailability

SET ActiveFlg = '1'

FROM DoctorAvailability D

INNER JOIN Appointment A

on A.DoctorUserName= D.DoctorUserName

INNER JOIN BillDetails B

on B.AppointmentID = A.AppointmentID

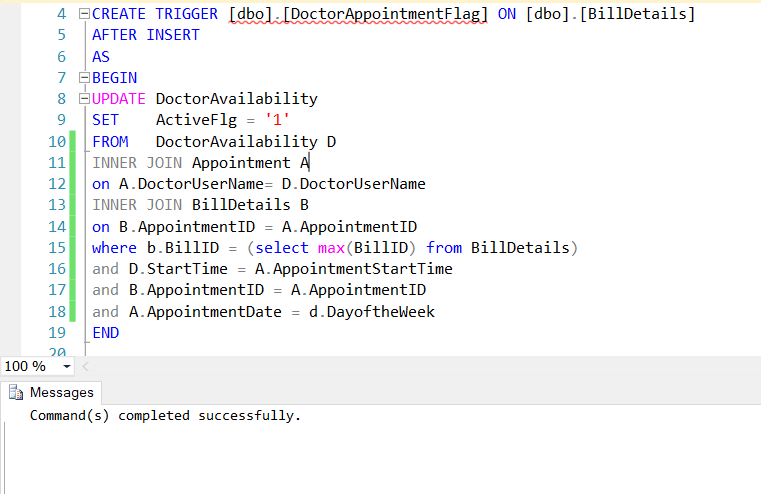
where b.BillID = (select max(BillID) from BillDetails)

and D.StartTime = A.AppointmentStartTime

and B.AppointmentID = A.AppointmentID

and A.AppointmentDate = d.DayoftheWeek

END



1. **After Insert Trigger for Doctor Availability upon Appointment:**

CREATE TRIGGER [dbo].[InitialOnAppointmentFlag] ON [dbo].[Appointment]

AFTER INSERT

AS

BEGIN

UPDATE DoctorAvailability

SET ActiveFlg = '0'

FROM DoctorAvailability D

INNER JOIN Appointment A

on A.DoctorUserName= D.DoctorUserName

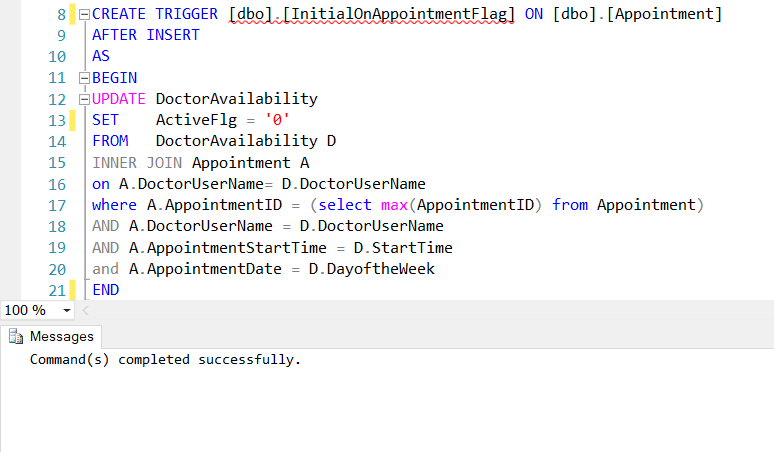
where A.AppointmentID = (select max(AppointmentID) from Appointment)

AND A.DoctorUserName = D.DoctorUserName

AND A.AppointmentStartTime = D.StartTime

and A.AppointmentDate = D.DayoftheWeek

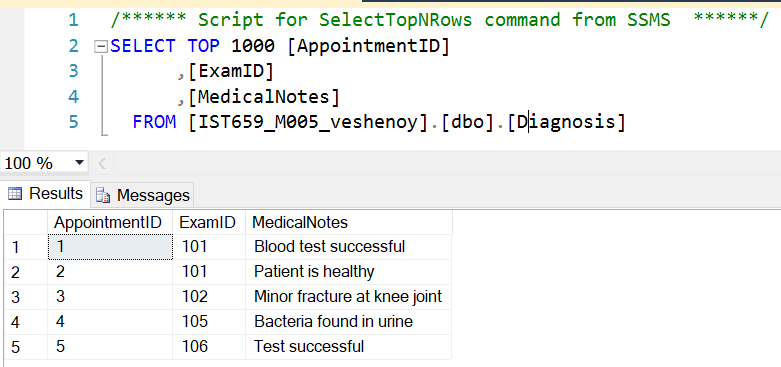
END



# MAJOR DATA QUESTIONS ANSWERED USING SQL

1. How many medical examinations are being conducted at the healthcare organization?

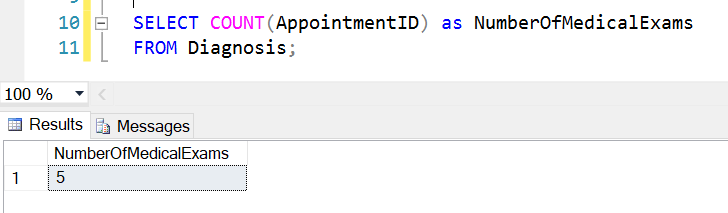
DATBASE:



QUERY:

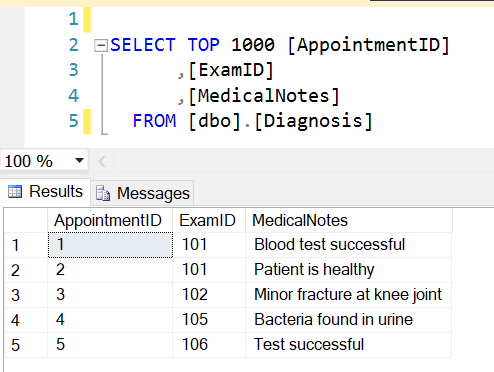
SELECT COUNT(AppointmentID) as NumberOfMedicalExams

FROM Diagnosis;



1. Which type of medical examination is being carried out most frequently?

DATABASE:



QUERY:

SELECT M.ExamID, M.ExamName

FROM MedicalExam M

INNER JOIN

Diagnosis D

ON M.ExamID = D.ExamID

GROUP BY M.ExamID, M.ExamName

HAVING COUNT(D.ExamID)=

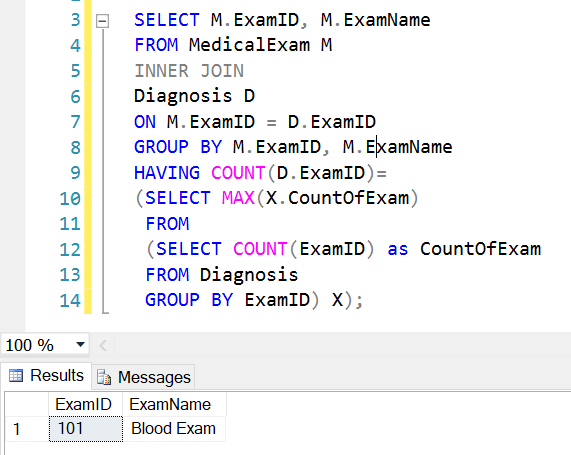
(SELECT MAX(X.CountOfExam)

FROM

(SELECT COUNT(ExamID) as CountOfExam

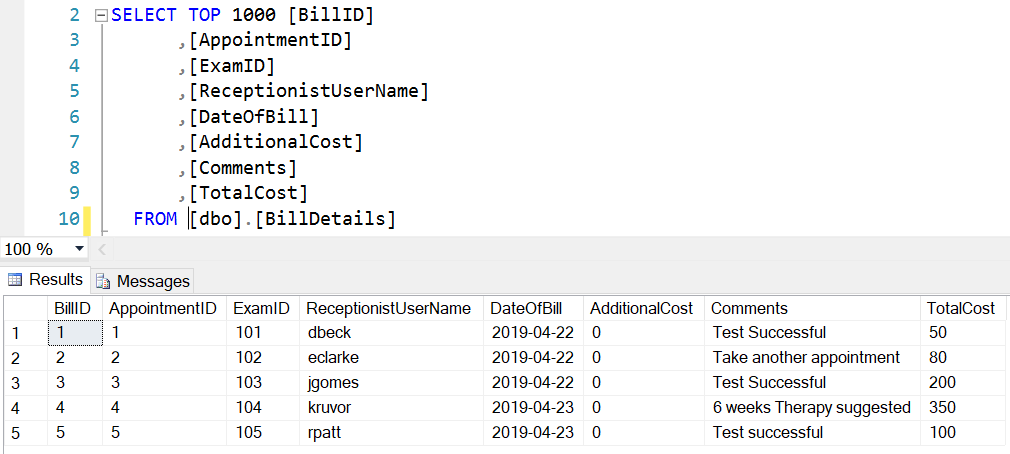
FROM Diagnosis

GROUP BY ExamID) X);



1. How much is the income of the healthcare organization?

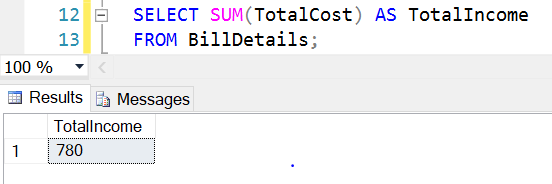
DATABASE:



QUERY:

SELECT SUM(TotalCost) as TotalIncome

FROM BillDetails;



1. What is the maximum number of patients that can be admitted in the healthcare organization?

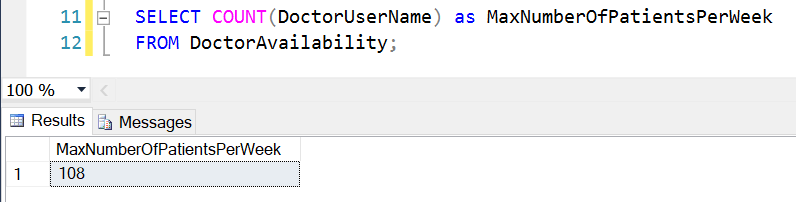
DATABASE:



QUERY:

SELECT COUNT(DoctorUserName) as MaxNumberOfPatientsPerWeek

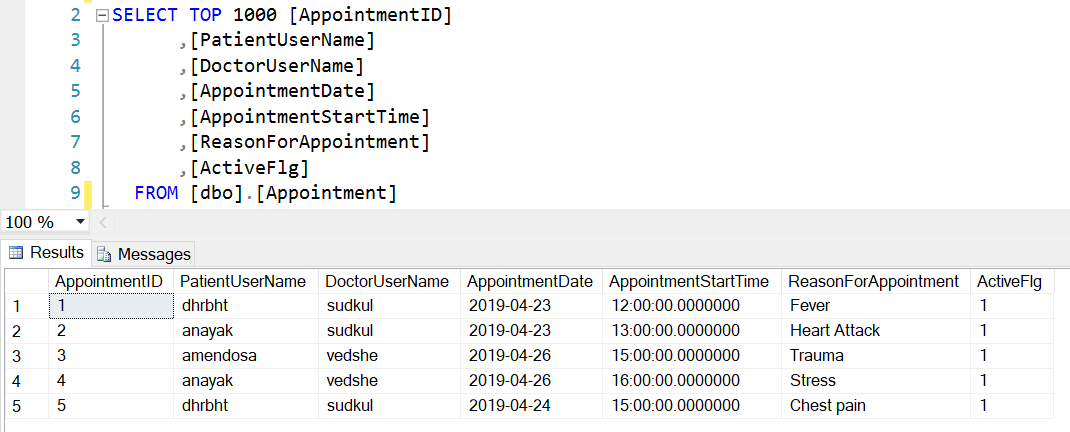
FROM DoctorAvailability;



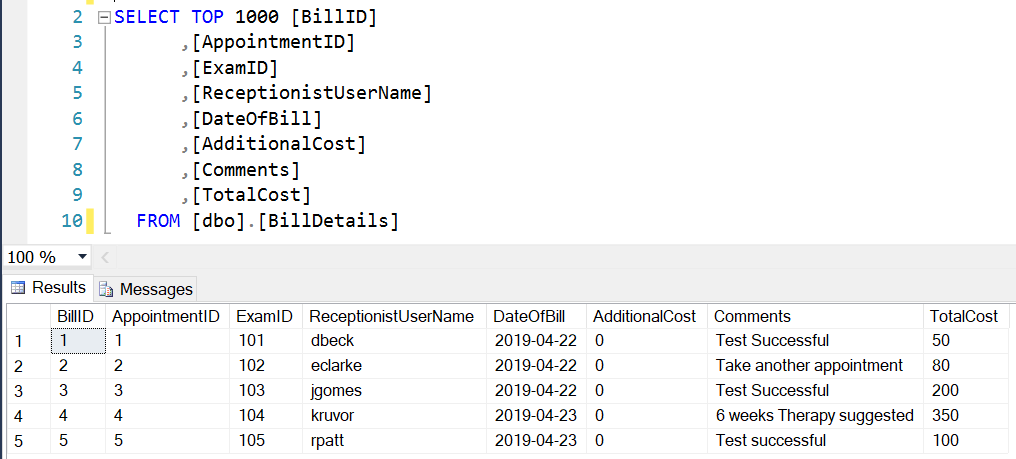
1. How many bills are being generated by a patient?

DATABASE:

1. Appointment Table



Bill Details Table:



QUERY:

SELECT A.PatientUserName, COUNT(BillID) AS NoOfBills

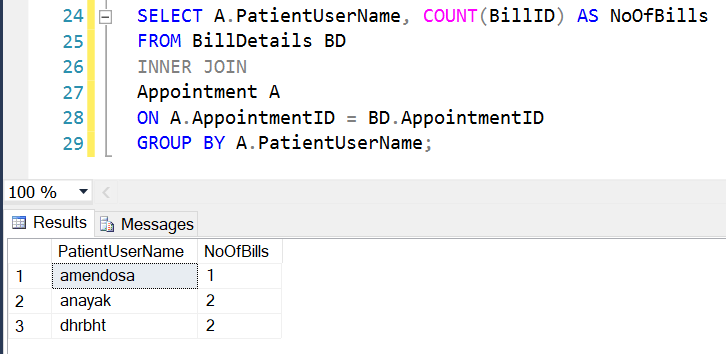
FROM BillDetails BD

INNER JOIN

Appointment A

ON A.AppointmentID = BD.AppointmentID

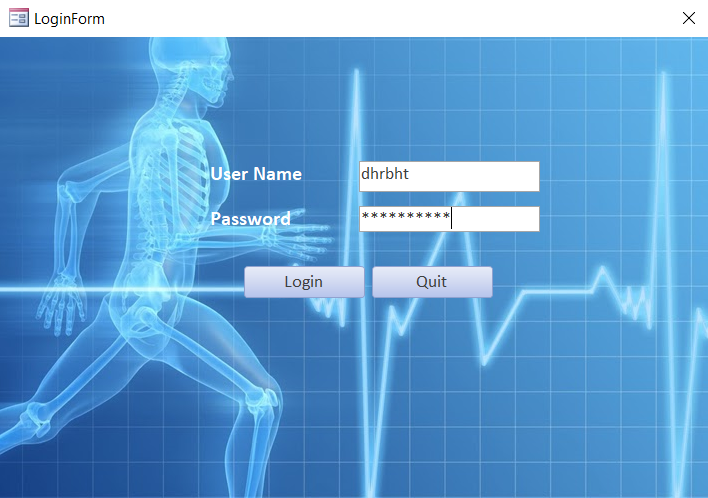
GROUP BY A.PatientUserName;



# INTERFACE IMPLEMENTATION USING FORMS

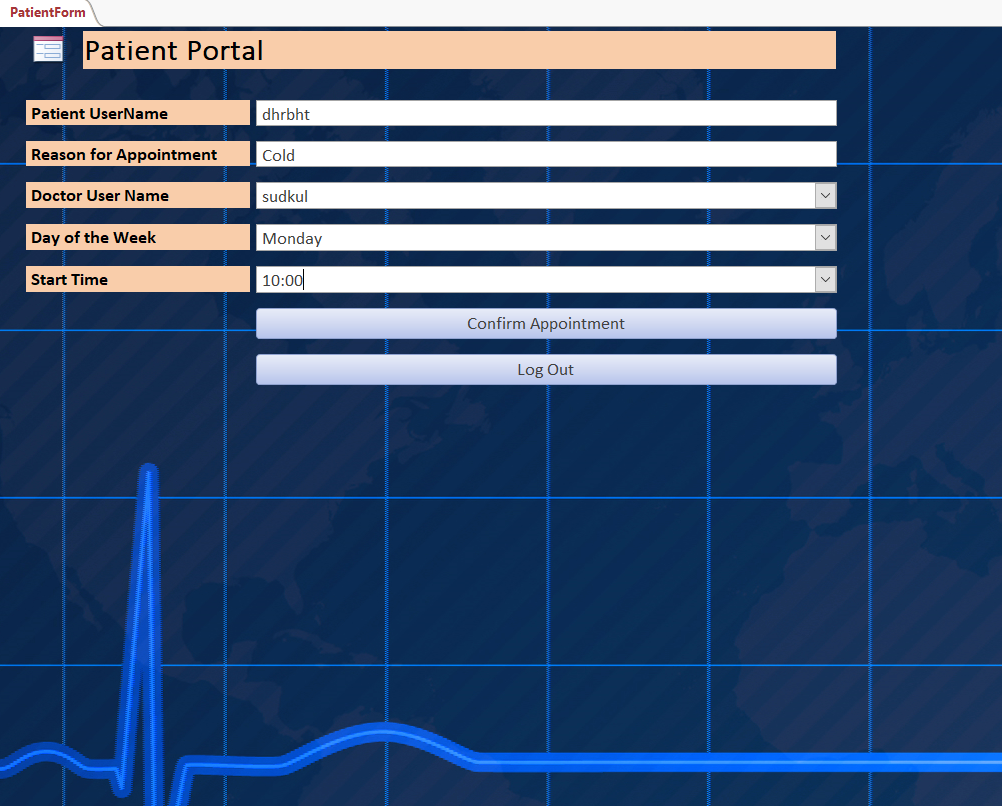
## LOGIN FORM

The login form enables the patient, doctor and the receptionist to log into the portal.



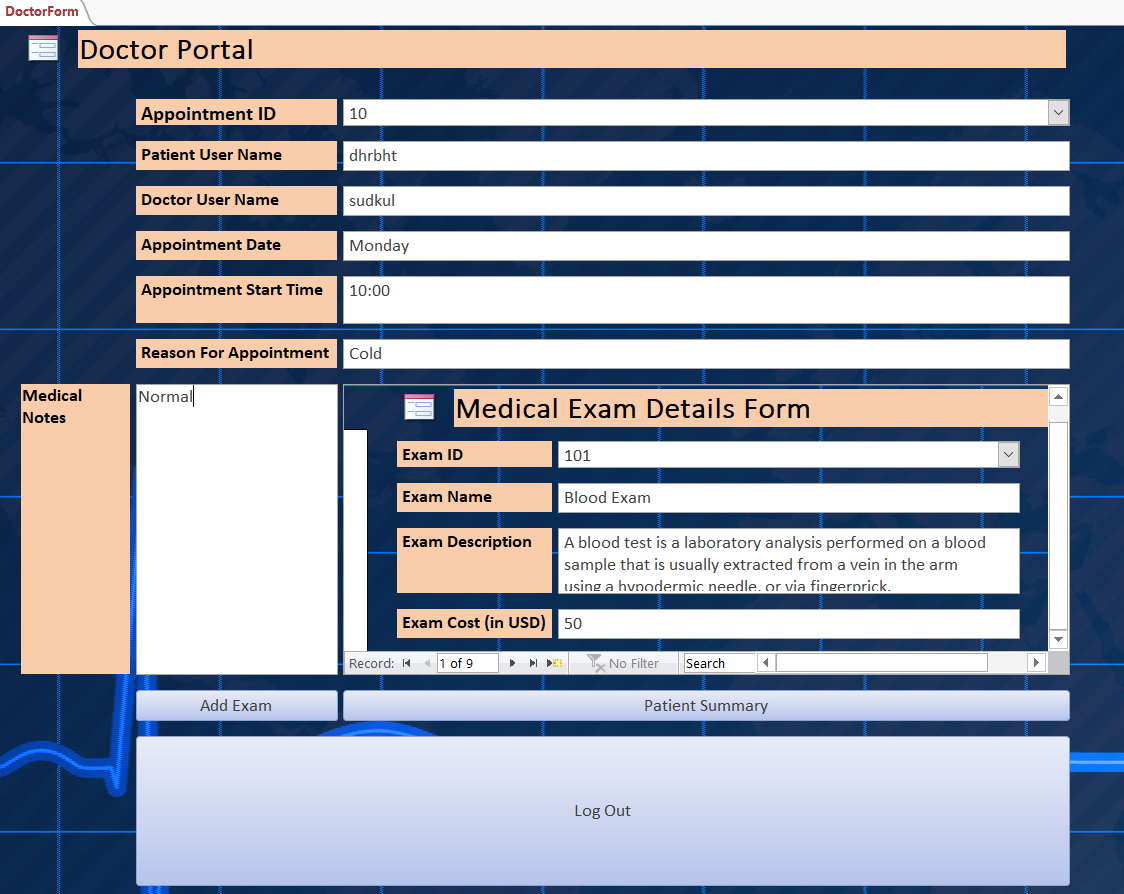
## PATIENT FORM

The patient can book an appointment using this form. After confirming the appointment day and time, he can log out of the system.



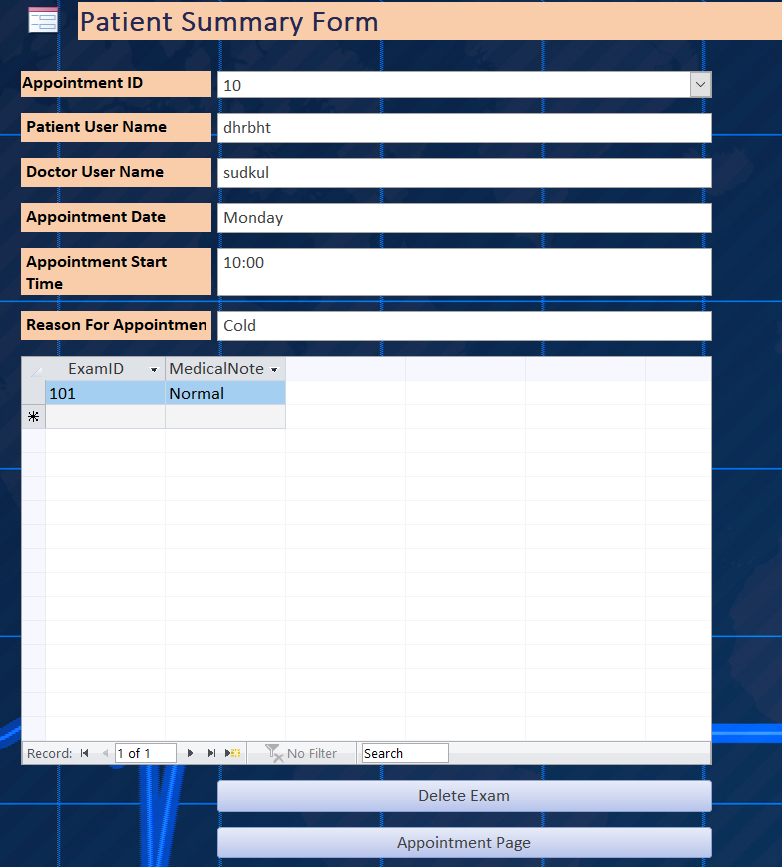
## DOCTOR FORM

This form allows the doctor to view his appointments and add medical exam details after diagnosing the patient. Also, there is a button which navigates the doctor to the patient summary form. Finally, the doctor can log out of the system.



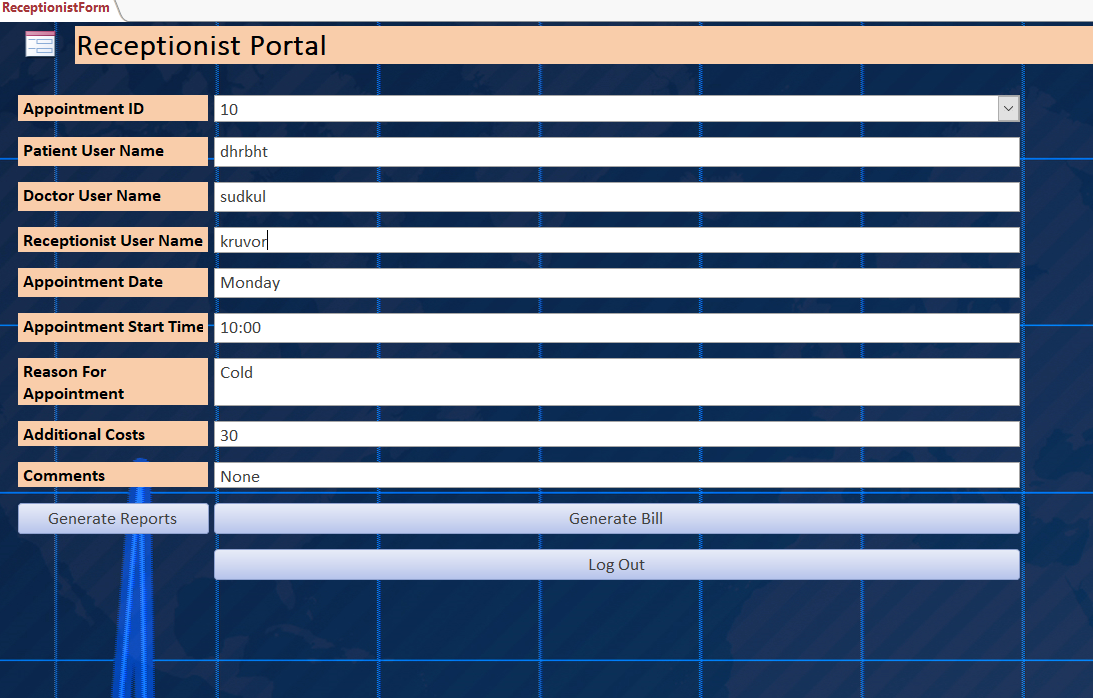
## PATIENT SUMMARY FORM

The doctor can view the details of the patient such as his appointment details and his medical exam details. Also, the doctor can delete the exam details.



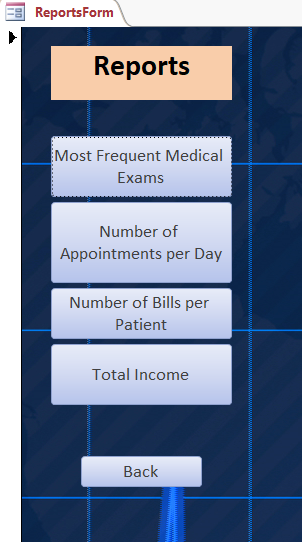
## RECEPTIONIST FORM

This form consists of receptionist portal information, which gives the feature of bill generation. Also, there is a button which can navigate to report generation form. Also, the receptionist can log out of the system.



## REPORT GENERATION FORM

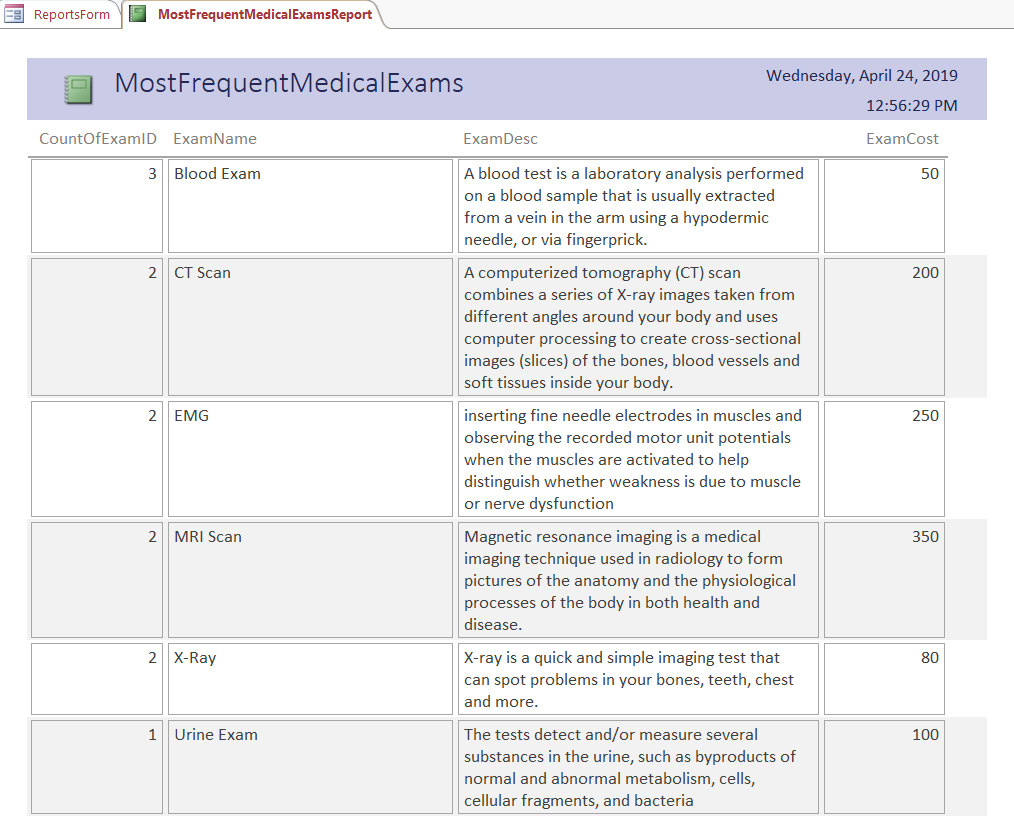
This form consists of buttons which can generate reports according to the data available in the healthcare portal database. The four reports generated are as follows: Most frequent medical exam report, Number of appointments per day report, Number of bills per patient and total income.



# INTERFACE IMPLEMENTATION USING REPORTS

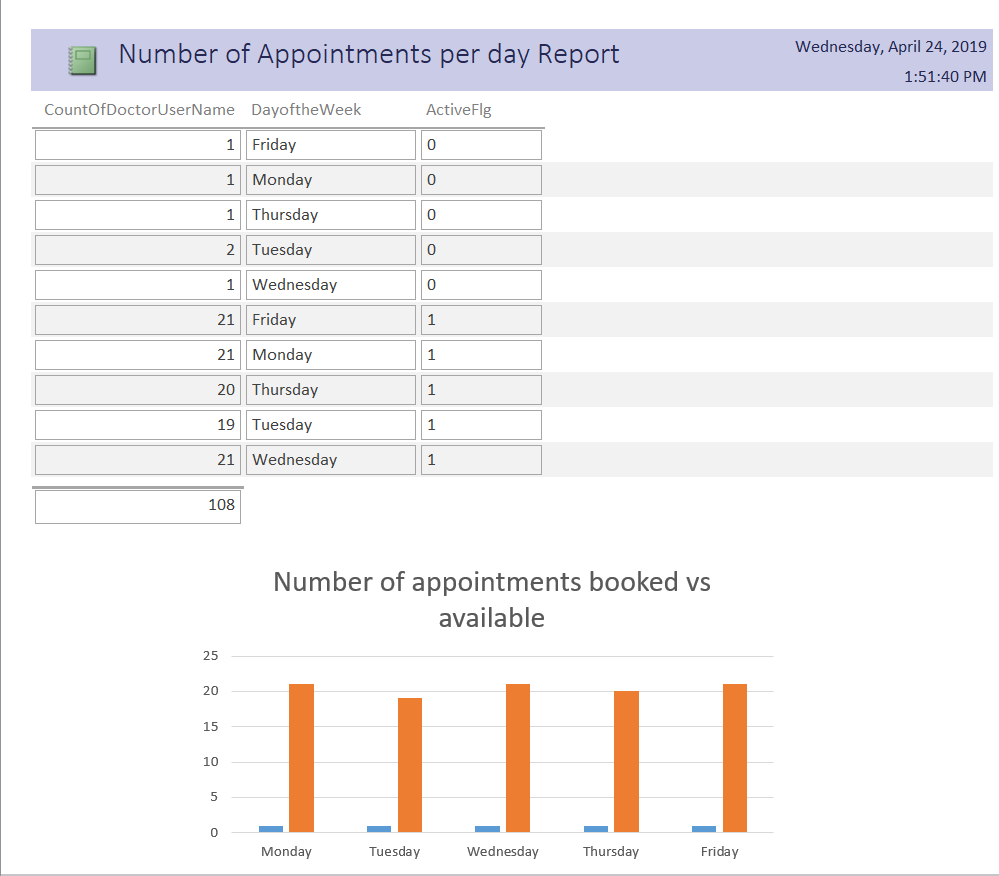
## MOST FREQUENT MEDICAL EXAM REPORT

This report shows the most frequent medical exams conducted at the healthcare center.



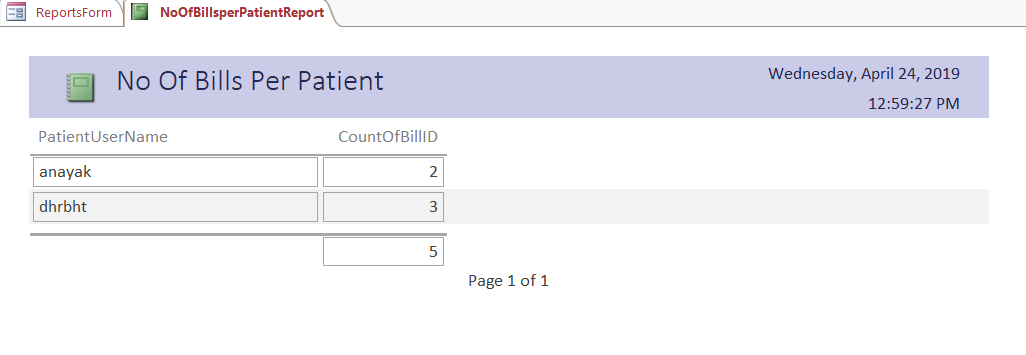
## APPOINTMENTS PER DAY REPORT

This report shows the number of appointments booked as well as available according to day of the week.



## BILLS GENERATED PER PATIENT REPORT

This report displays the number of bills generated by each patient at the healthcare center.



## TOTAL INCOME REPORT

The total income of the healthcare center can be observed with the help of this report.

