IST-659

Project Implementation Report

Hospitech – Hospital Management System

* Rahul Wable & Vasundhara Patil

**CONTENTS**

1. Project Summary
2. Entity and Attribute Table
3. Entity Relationship Diagram
4. Database System Architecture
5. SQL Scripts for creating tables and inserting data
6. Major Data Questions
7. Interfaces

**PROJECT SUMMARY**

Our Project HOSPITECH is an online database system for a civil Hospital in Mumbai, India.

The digitization of healthcare is driving the need for systems to manage and organize large amounts of personal health data in order to drive more comprehensive care for patients across the healthcare continuum. Currently, the civil hospital does not have an online database system to record the details of the patients and provide them with efficient treatment by effectively assigning the doctors. Record of each patient is stored in the form of paper-based system. This system involves taking down patient data on pieces of papers, which are then put into the files and filed in cabinets. This manual work is tedious. Another problem is the missing of pieces of information. Sometimes there are pieces of information the doctor could not find in the paper file. Physicians and their office staffs have said that, they spent more time looking for patient information, than the time they have for the patient.

Our proposed system is a way to:

1) Fully automate the operation of the hospital and managerial system in the hospital.

2) Improve and standardize practice planning of efficient and communication skill in the hospital

3) Improve the accurate record keeping in the hospital

4) Provide timely access to patients and personnel records.

This system will reduce the human errors and help reduce on the time that would be spend on looking for patient’s records. Health professionals and the caregivers would not only be able to find the records pertaining to health of the patient, but it will also be able to give the best possible medical treatment. Installing such management software will help the civil hospital save on paper, ink and other office supplies. Thus, the hospital will function smoothly and save both time and resources by looking for an electronic record management system.

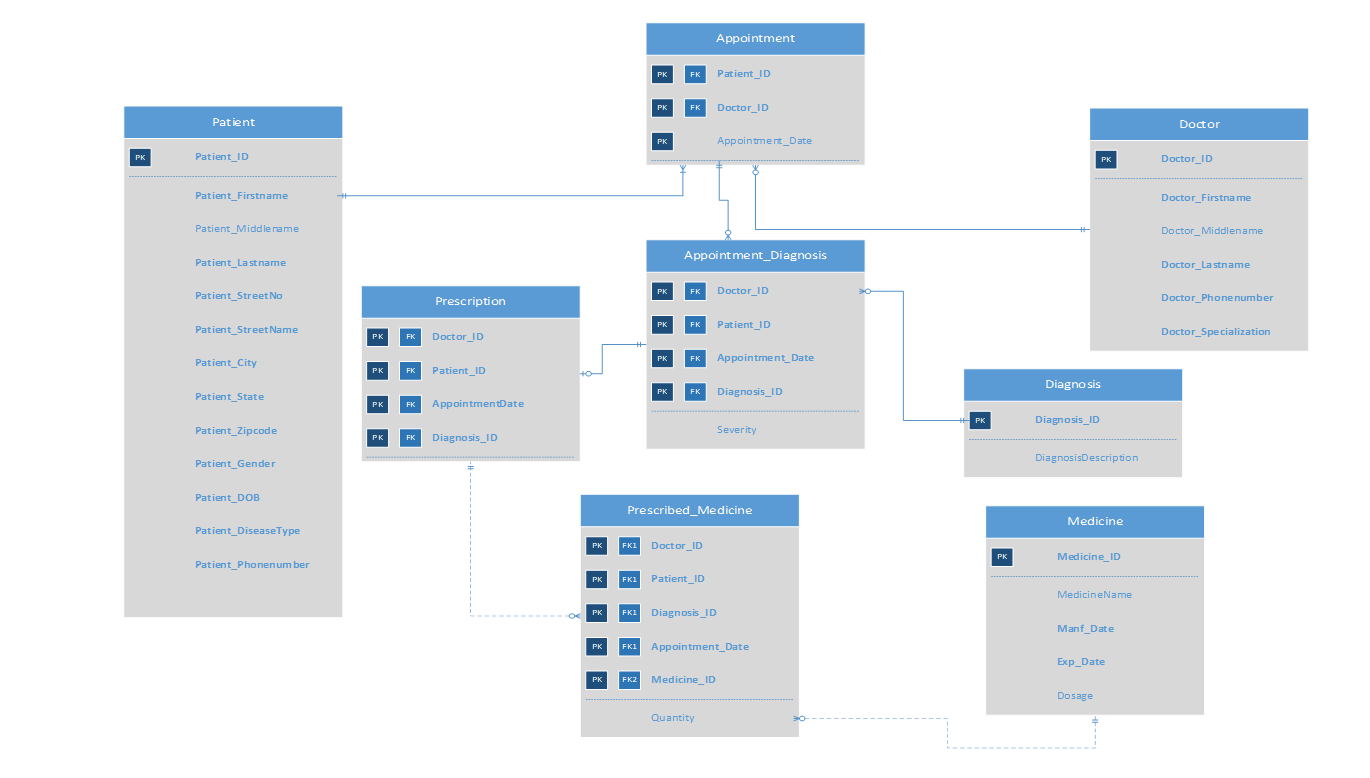
The scope of the system is to develop a database system that automates all the activities of patients in the hospital starting from booking an appointment with a doctor to prescribing medicines. A patient will book an appointment with a doctor. The system will store the patient details like the name, address, DOB, type of disease etc. that will be addressed by a doctor. A patient can book many appointments and many appointments can be associated to one doctor.

Each appointment can consist of zero or more diagnosis. The doctor treats patient depending on the diagnosis. The doctor prescribes medicines to a patient based on the diagnosis. Each prescription is uniquely identified by an appointment date. Each prescription can have one or more medicine.

**ENTITIES & ATTRIBUTES**

|  |  |
| --- | --- |
| Entities and Attributes | Description |
| **PATIENT** |  |
| Patient\_ID | Primary key |
| Patient\_Firstname | This field specifies the first name of the patient |
| Patient\_Middlename | This field specifies the middle name of the patient |
| Patient\_Lastname | This field specifies the last name of the patient |
| Patient\_StreetNo | This field specifies the Street No of the patient |
| Patient\_StreetName | This field specifies the street name of the patient |
| Patient\_City | This field specifies the city the patient stays in |
| Patient\_State | This field specifies the state the patient stays in |
| Pateint\_Zipcode | This field specifies the zip code |
| Patient\_Gender | This field specifies the gender of the patient |
| Patient\_DOB | This field specifies the Date of birth of the patient |
| Patient\_DiseaseType | This field specifies what kind of disease the  patient is suffering from |
| Patient\_Phonenumber | This field specifies the phone number of the patient |
|  |  |
| **DOCTOR** |  |
| Doctor\_ID | This is a unique identifier of every doctor - Primary key |
| Doctor\_Firstname | This field specifies the first name of the doctor |
| Doctor\_Middlename | This field specifies the middle name of the doctor |
| Doctor\_Lastname | This field specifies the last name of the doctor |
| Doctor\_Phonenumber | This field specifies the phone number of the doctor |
| Doctor\_Specialization | This fields specifies the specialty of the doctor.  Whether he is an orthopedic, pediatric, neurologist,  general surgeon, emergency physician etc. |
| **APPOINTMENT** |  |
| Patient\_ID | Foreign key + Primary Key |
| Doctor\_ID | Foreign key + Primary Key |
| Appointmnet\_Date | Primary key. This is the unique identifier of every appointment.  This field specifies the date and time of the appointment. |
|  |  |
|  |  |
| **PRESCRIPTION** |  |
| Appointmnet\_Date | Primary key+ Foreign key |
| Patient\_ID | Primary key+ Foreign key |
| Doctor\_ID | Primary key+ Foreign key |
| Diagnosis\_ID | Primary key+ Foreign key |
|  |  |
| **DIAGNOSIS** |  |
| Diagnosis\_ID | Primary Key |
| DiagnosisDescription | This field specifies the description of the diagnosis. |
|  |  |
| **MEDICINE** |  |
| Medicine\_ID | Primary Key |
| MedicineName | This field specifies the name of the medicine. |
| Manf\_Date | This field specifies the manufacturing date of the medicine |
| Exp\_Date | This field specifies the expiry date of the medicine |
| Dosage | This field specifies the dosage of the medicine. |
|  |  |
| **APPOINTMENT DIAGNOSIS** |  |
| Docot\_ID | Primary key + Foreign key |
| Patient\_ID | Primary key + Foreign key |
| Appointment\_Date | Primary key + Foreign key |
| Diagnosis\_ID | Primary key + Foreign key |
| Severity | This field specifies the severity of the patient. |
|  |  |
| **PRESCRIBED\_MEDICINE** |  |
| Doctor\_ID | Primary Key + Foreign key 1 |
| Patient\_ID | Primary Key + Foreign key 1 |
| Appointment\_Date | Primary Key + Foreign key 1 |
| Medicine\_ID | Primary Key + Foreign key 2 |
|  |  |
| Diagnosis\_ID | Primary key + Foreign key 1 |
| Quantity | This field specifies the quantity of the medicine to be taken by the patient |

**ENTITY RELATIONSHIP DIAGRAM**



**DATABASE SYSTEM ARCHITECTURE**

We used the following tools to create and implement this project.

1. **MS Visio**: We created entity relationship diagram using MS Visio. Here we created entities and defined their attributes along with the keys (primary & foreign). We established the relationships and

dependencies between entities.

1. **SQL Server**: We used SQL Server as the database that stored all the tables and their data. We created tables in the database using SQL queries. Also, we used SQL server to create and deploy trigger required for the project.
2. **MS Access**: We used MS Access to create the interface for the system. Using Access, we linked our tables that were created in SQL Server. Once, the tables were linked, we created forms that could take user input or display the necessary information to the users. Based on the data, we used MS Access to generate reports for the users of the system.

**SQL SCRIPTS FOR CREATING TABLES AND INSERTING DATA**

**CREATING TABLES:**

We created tables in SQL Server using the following SQL queries:

1. **Patient Table**

CREATE TABLE Patient (

PatientID VARCHAR(10) NOT NULL,

Patient\_Firstname VARCHAR(30) NOT NULL,

Patient\_Middlename VARCHAR(30) NOT NULL,

Patient\_Lastname VARCHAR(30) NOT NULL,

Patient\_StreetNo INTEGER NOT NULL,

Patient\_StreetName VARCHAR(30) NOT NULL,

Patient\_City VARCHAR(20) NOT NULL,

Patient\_State VARCHAR(20) NOT NULL,

Patient\_Zipcode CHAR(10) NOT NULL,

Patient\_Gender VARCHAR(10) NOT NULL,

Patient\_DOB DATETIME NOT NULL,

Patient\_DiseaseType VARCHAR(30) NOT NULL,

Patient\_Phonenumber VARCHAR(10),

CONSTRAINT PatientID\_PK PRIMARY KEY (PatientID)

);

1. **Doctor Table**

CREATE TABLE Doctor (

Doctor\_ID VARCHAR(10) NOT NULL,

Doctor\_Firstname VARCHAR(30) NOT NULL,

Doctor\_Middlename VARCHAR(30) NOT NULL,

Doctor\_Lastname VARCHAR(30) NOT NULL,

Doctor\_Phonenumber VARCHAR(10),

Doctor\_Specialization VARCHAR(30) NOT NULL

CONSTRAINT Doctor\_ID\_PK PRIMARY KEY (Doctor\_ID)

);

1. **Appointment Table**

CREATE TABLE Appointment (

PatientID VARCHAR(10) NOT NULL,

Doctor\_ID VARCHAR(10) NOT NULL,

Appointment\_Date DATETIME NOT NULL,

CONSTRAINT Appointment\_PK PRIMARY KEY (PatientID,Doctor\_ID,Appointment\_Date),

CONSTRAINT PatientID\_FK1 FOREIGN KEY (PatientID) REFERENCES Patient,

CONSTRAINT Doctor\_ID\_FK2 FOREIGN KEY (Doctor\_ID) REFERENCES Doctor

);

1. **Diagnosis Table**

CREATE TABLE Diagnosis (

DiagnosisID VARCHAR(10) NOT NULL,

DiagnosisDescription VARCHAR(30) NOT NULL

CONSTRAINT Diagnosis\_PK PRIMARY KEY (DiagnosisID)

);

1. **Medicine Table**

CREATE TABLE Medicine (

Medicine\_ID VARCHAR(10) NOT NULL,

MedicineName VARCHAR(30) NOT NULL,

Manf\_Date DATETIME,

Exp\_Date DATETIME,

Dosage FLOAT NOT NULL

CONSTRAINT Medicine\_ID\_PK PRIMARY KEY (Medicine\_ID)

);

1. **Appointment\_Diagnosis Table**

CREATE TABLE Appointment\_Diagnosis (

Doctor\_ID VARCHAR(10) NOT NULL,

PatientID VARCHAR(10) NOT NULL,

Appointment\_Date DATETIME NOT NULL,

DiagnosisID VARCHAR(10) NOT NULL,

Severity VARCHAR(20),

CONSTRAINT AppointmentD\_PK PRIMARY KEY (PatientID,Doctor\_ID,Appointment\_Date,DiagnosisID),

CONSTRAINT Appointment\_FK1 FOREIGN KEY (PatientID,Doctor\_ID,Appointment\_Date) REFERENCES Appointment(PatientID,Doctor\_ID,Appointment\_Date),

CONSTRAINT Diagnosis\_FK4 FOREIGN KEY (DiagnosisID) REFERENCES Diagnosis

);

1. **Prescription Table**

CREATE TABLE Prescription (

Doctor\_ID VARCHAR(10) NOT NULL,

PatientID VARCHAR(10) NOT NULL,

Appointment\_Date DATETIME NOT NULL,

DiagnosisID VARCHAR(10) NOT NULL,

CONSTRAINT Appointment\_PK2 PRIMARY KEY (PatientID,Doctor\_ID,Appointment\_Date,DiagnosisID),

CONSTRAINT Patient\_\_ID\_FK1 FOREIGN KEY (PatientID) REFERENCES Patient,

CONSTRAINT Doctor\_\_ID\_FK2 FOREIGN KEY (Doctor\_ID) REFERENCES Doctor,

CONSTRAINT Appointment\_FK3 FOREIGN KEY (PatientID,Doctor\_ID,Appointment\_Date) REFERENCES Appointment(PatientID,Doctor\_ID,Appointment\_Date),

CONSTRAINT DiagnosisID\_\_FK4 FOREIGN KEY (DiagnosisID) REFERENCES Diagnosis

);

1. **Prescribed\_Medicine Table**

CREATE TABLE Prescribed\_Medicine (

Doctor\_ID VARCHAR(10) NOT NULL,

PatientID VARCHAR(10) NOT NULL,

Appointment\_Date DATETIME NOT NULL,

DiagnosisID VARCHAR(10) NOT NULL,

Medicine\_ID VARCHAR(10) NOT NULL,

Quantity INTEGER,

CONSTRAINT PrecribedMed\_PK PRIMARY KEY (PatientID,Doctor\_ID,Appointment\_Date,DiagnosisID,Medicine\_ID),

CONSTRAINT PatientID\_FK5 FOREIGN KEY (PatientID) REFERENCES Patient,

CONSTRAINT Doctor\_ID\_FK3 FOREIGN KEY (Doctor\_ID) REFERENCES Doctor,

CONSTRAINT Appointment\_FK4 FOREIGN KEY (PatientID,Doctor\_ID,Appointment\_Date) REFERENCES Appointment(PatientID,Doctor\_ID,Appointment\_Date),

CONSTRAINT DiagnosisID\_FK4 FOREIGN KEY (DiagnosisID) REFERENCES Diagnosis,

CONSTRAINT MedicineID\_FK5 FOREIGN KEY (Medicine\_ID) REFERENCES Medicine

);

**INSERTING DATA:**

We used the following SQL Queries to insert data in the tables created using the above queries.

1. **Patient Table**

INSERT INTO Patient VALUES('P01','Rahul','Dilip','Wable','710','S Beech St','Syracuse','New York','13210','Male','08/31/1976','Diabetes','3156409706');

INSERT INTO Patient VALUES('P02','Sue','Ben','Stokes','740','N Beech St','Raleigh','North Carolina','13410','Female','02/11/1991','Heart Disease','6097654444');

INSERT INTO Patient VALUES('P03','Joe','Chris','Root','400','Clinton St','Miami','Florida','14330','Male','07/30/1986','Cancer','6023407605');

INSERT INTO Patient VALUES('P04','Chris','Mark','Lynn','434','Salina Ave','Dallas','Texas','16540','Male','04/15/1977','Diabetes','3153456706');

1. **Doctor Table**

INSERT INTO Doctor Values('D01','James','Nick','Faulkner','6098765907','Endocrinologist');

INSERT INTO Doctor Values('D02','Jim','Benjamin','Brooks','8076545676','Dermatologist');

INSERT INTO Doctor Values('D03','Dany','John','Rashford','3456548878','Cardiologist');

INSERT INTO Doctor Values('D04','Stephen','Harry','Wings','8075677897','Oncologist');

1. **Appointment Table**

INSERT INTO Appointment Values('P01','D01','08/20/2012')

INSERT INTO Appointment Values('P02','D03','08/21/2013')

INSERT INTO Appointment Values('P03','D04','04/12/2018')

INSERT INTO Appointment Values('P04','D01','10/13/2015')

1. **Diagnosis Table**

INSERT INTO Diagnosis Values('DG1','Undergoing Diabetes Treatment');

INSERT INTO Diagnosis Values('DG2','Undergoing Cancer Treatment');

INSERT INTO Diagnosis Values('DG3','Undergoing Cardio Treatment');

INSERT INTO Diagnosis Values('DG4','Undergoing Skin Treatment');

1. **Medicine Table**

INSERT INTO Medicine Values('M01','Metformin','08/22/2011','08/22/2015','250')

INSERT INTO Medicine Values('M02','Femara','07/20/2012','07/20/2015','125')

INSERT INTO Medicine Values('M03','Accupril','08/02/2011','08/02/2013','350')

INSERT INTO Medicine Values('M04','Clemastine','09/22/2015','10/22/2017','400')

1. **Appointment Diagnosis**

INSERT INTO Appointment\_Diagnosis Values('D01','P01','08/20/2012','DG1','medium');

INSERT INTO Appointment\_Diagnosis Values('D03','P02','08/21/2013','DG3','high');

INSERT INTO Appointment\_Diagnosis Values('D04','P03','04/12/2018','DG2','low');

INSERT INTO Appointment\_Diagnosis Values('D01','P04','10/13/2015','DG1','low');

1. **Prescription Table**

INSERT INTO Prescription Values('D01','P01','08/20/2012','DG1');

INSERT INTO Prescription Values('D03','P02','08/21/2013','DG3');

INSERT INTO Prescription Values('D04','P03','04/12/2018','DG2');

INSERT INTO Prescription Values('D01','P04','10/13/2015','DG1');

1. **Prescribed\_Medicine Table**

INSERT INTO Prescribed\_Medicine Values('D01','P01','08/20/2012','DG1','M01','2');

INSERT INTO Prescribed\_Medicine Values('D03','P02','08/21/2013','DG3','M03','3');

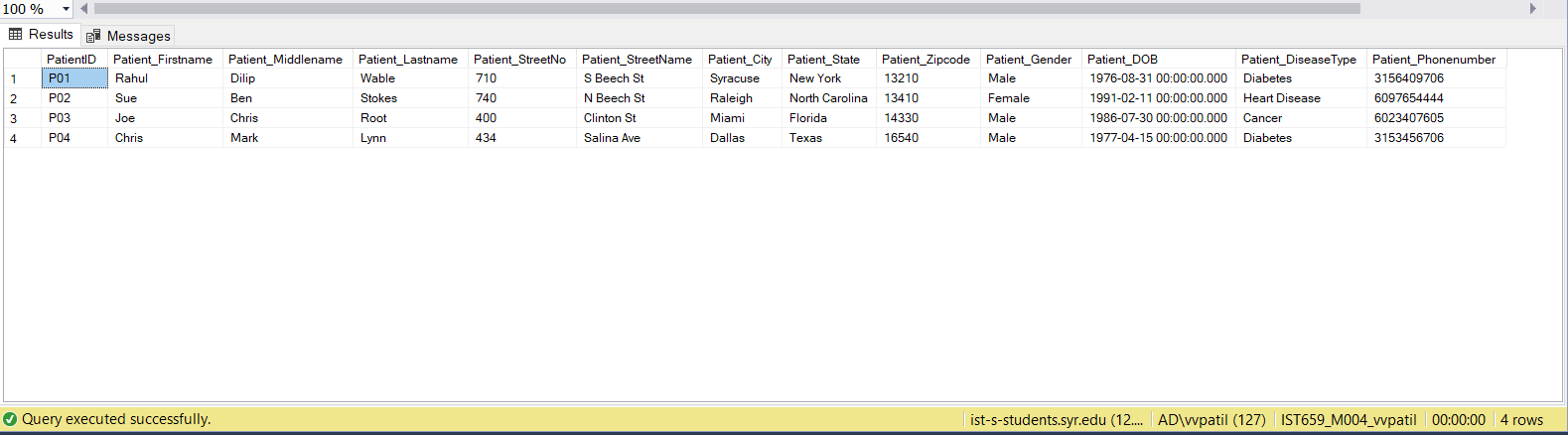
INSERT INTO Prescribed\_Medicine Values('D04','P03','04/12/2018','DG2','M02','1');

INSERT INTO Prescribed\_Medicine Values('D01','P04','10/13/2015','DG1','M01','1');

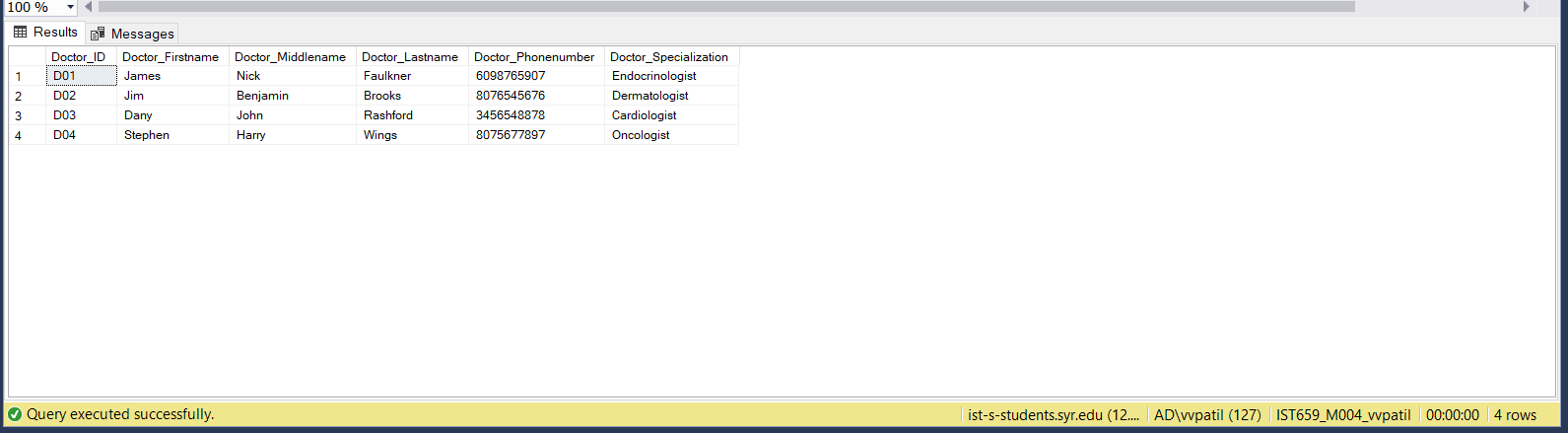
**OUTPUT FOR CREATE AND INSERT SQL SCRIPTS:**

The below tables contain data which was inserted using SQL scripts, Access forms and trigger.

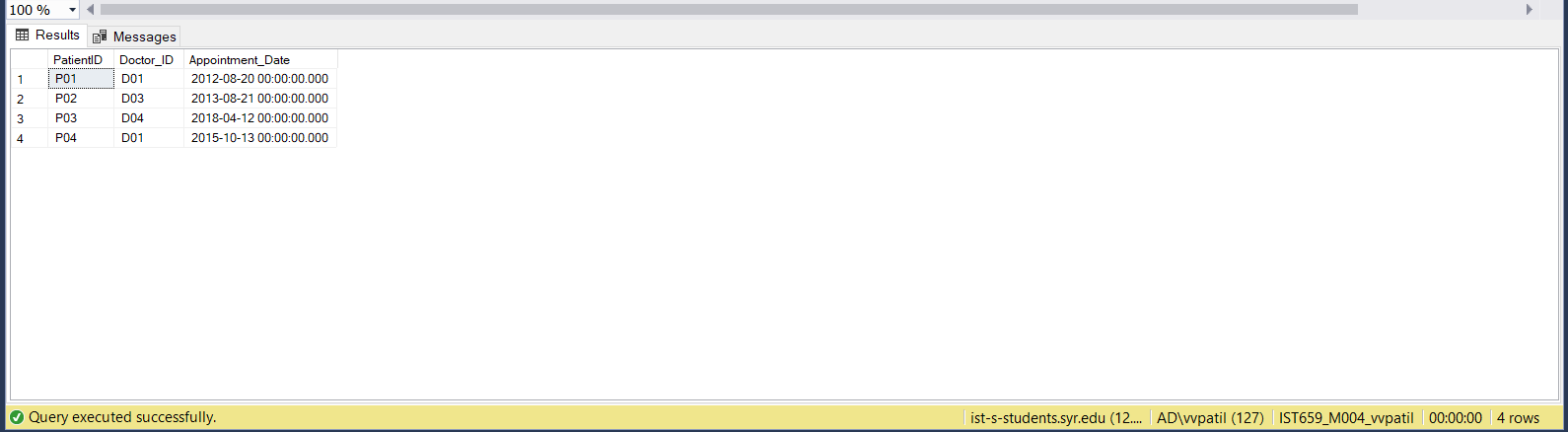
1. **Patient Table**



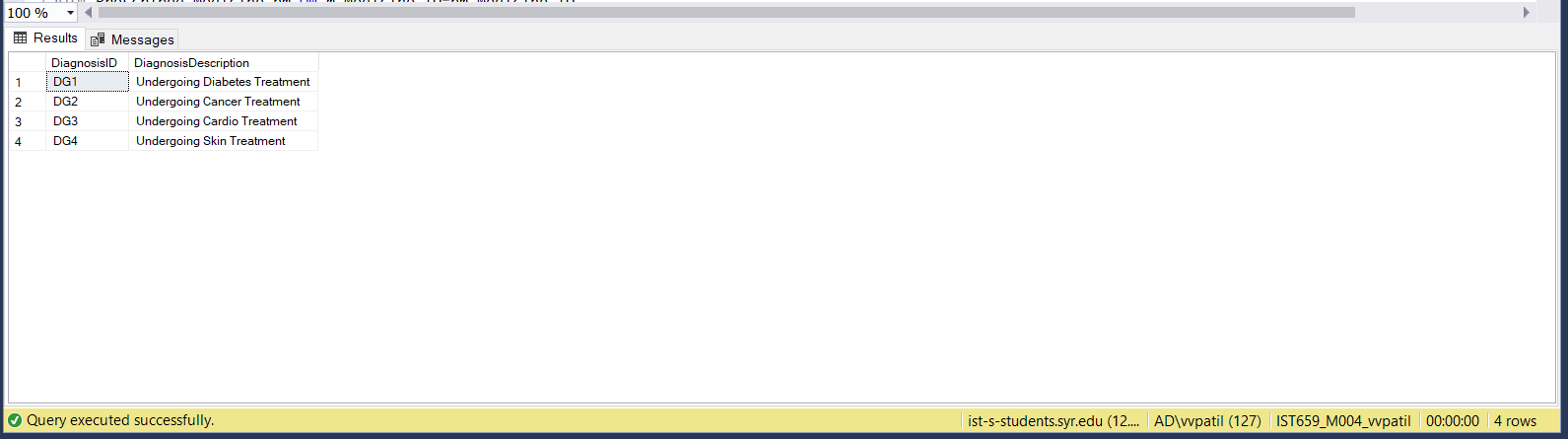
1. **Doctor Table**



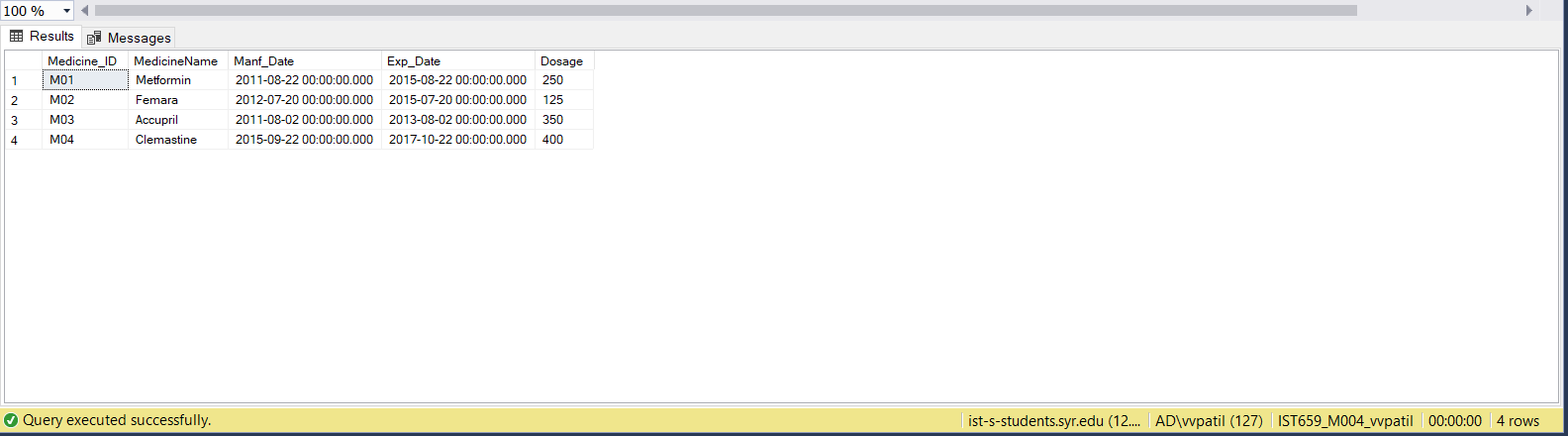
1. **Appointment Table**



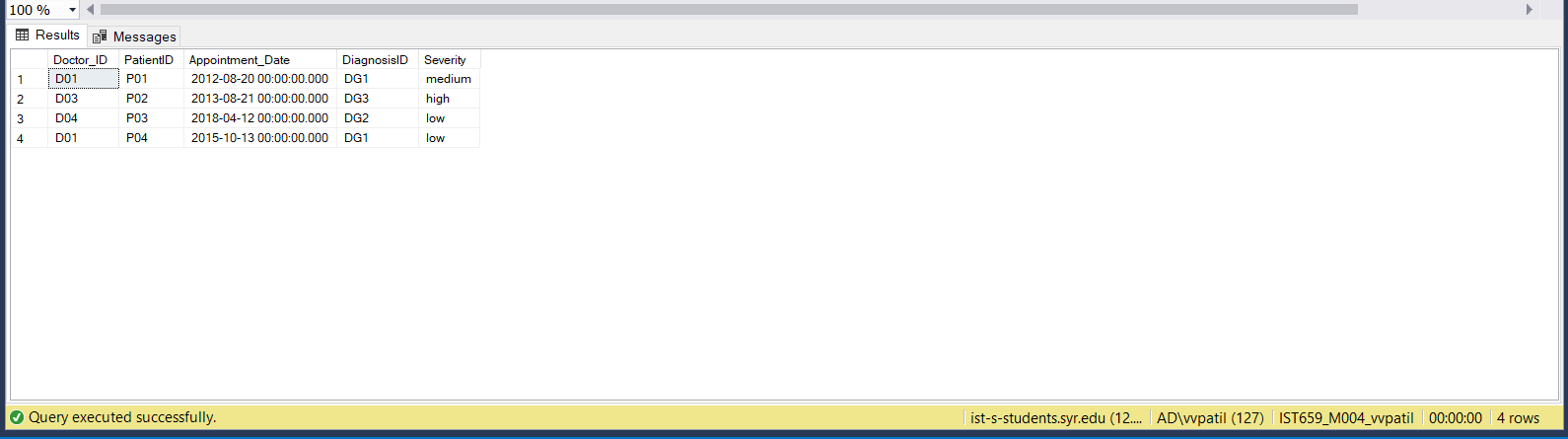
1. **Diagnosis Table**



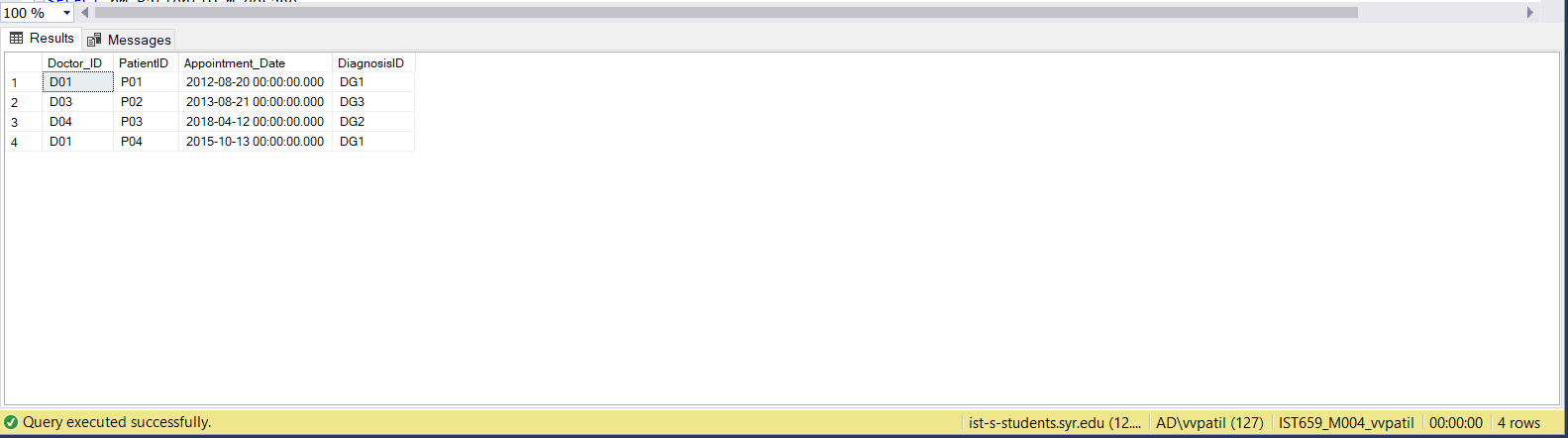
1. **Medicine Table**



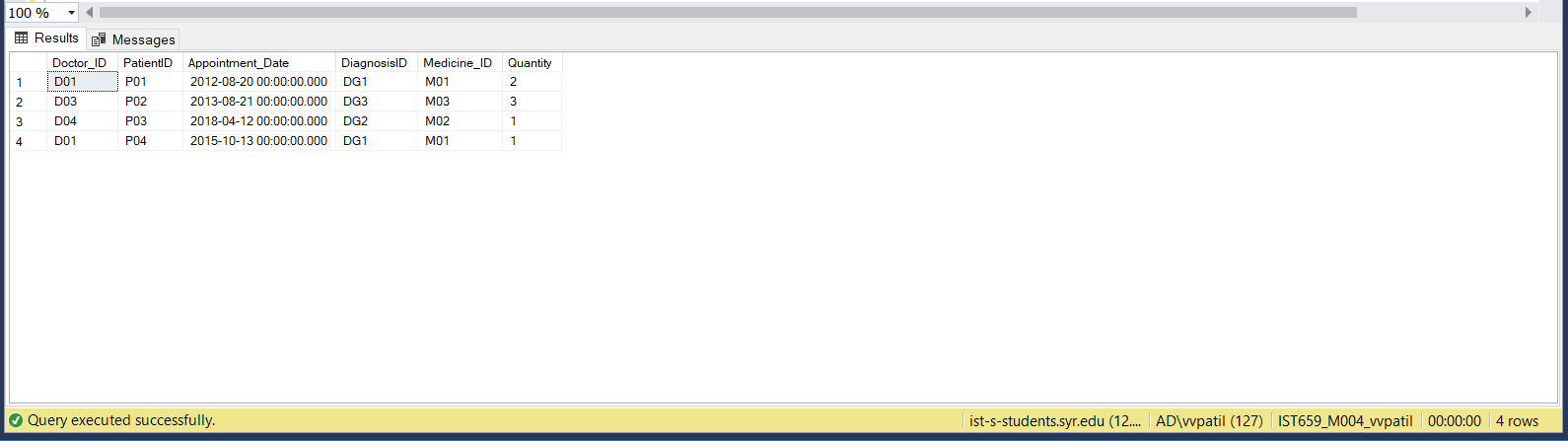
1. **Appointment Diagnosis**



1. **Prescription Table**



1. **Prescribed\_Medicine Table**



**MAJOR DATA QUESTIONS**

**We have answered the major data questions using SQL queries in MS Access. We have created forms and reports to display the same to the user for easy understanding.**

1. **What is the dosage of medicine given to a patient?**

* **SQL Query**

SELECT p.Patient\_Firstname,p.Patient\_Lastname, pm.PatientID,m.dosage

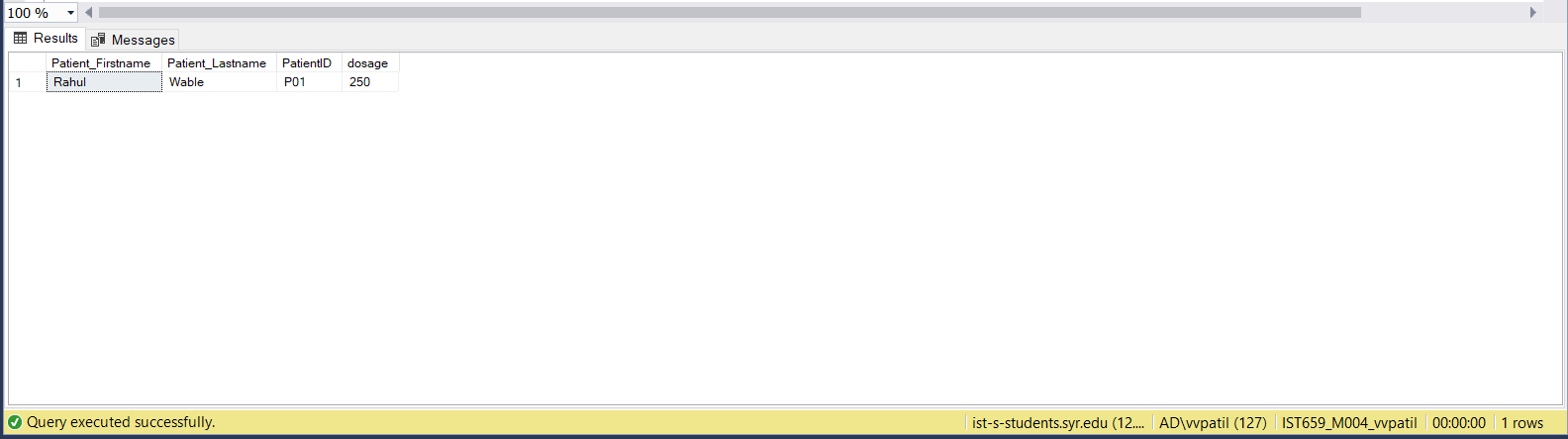
FROM Medicine m

JOIN Prescribed\_Medicine pm ON m.Medicine\_ID=pm.Medicine\_ID

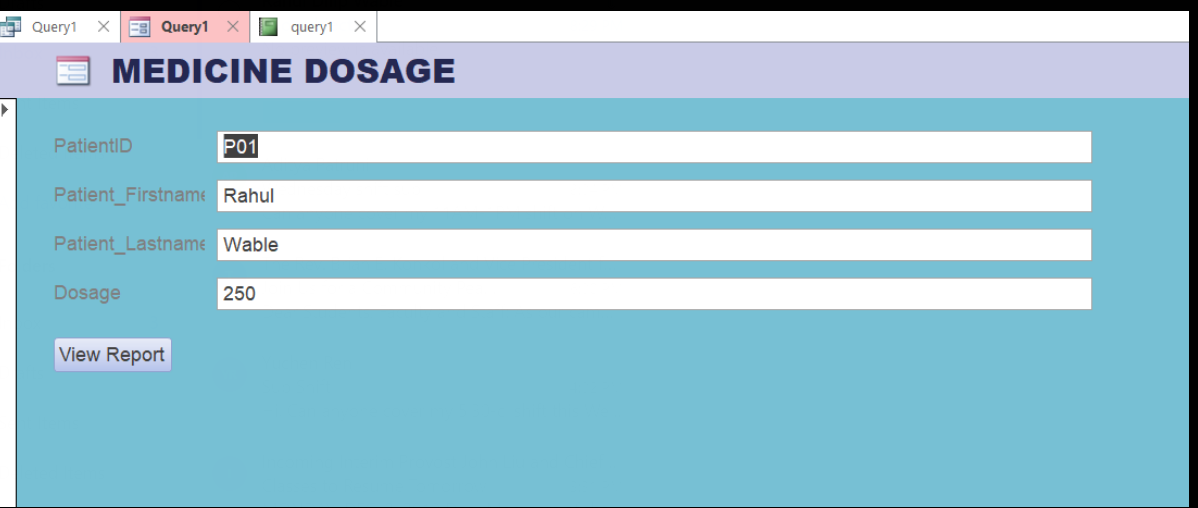
JOIN Patient p ON pm.PatientID=p.PatientID

WHERE pm.PatientID='P01'

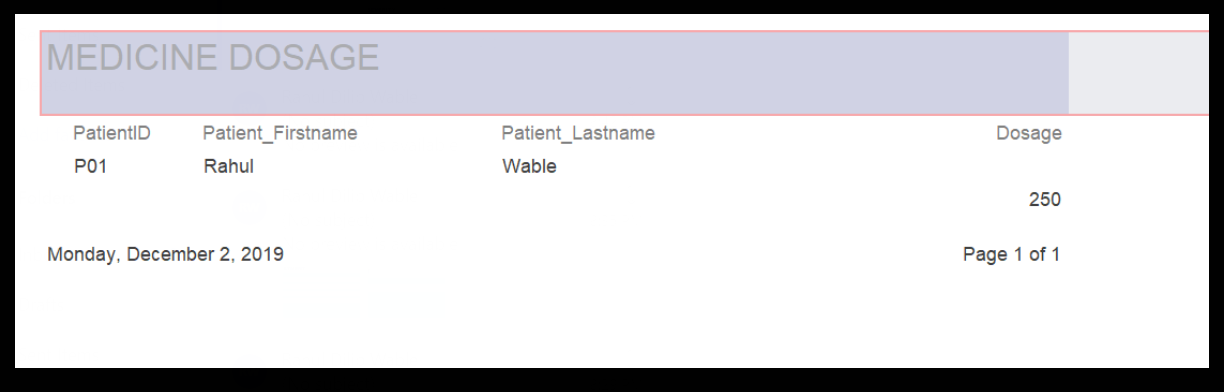
* **Output (Datasheet)**



* **Output (Form)**



* **Output (Report)**



1. **How severe is a patient’s condition?**

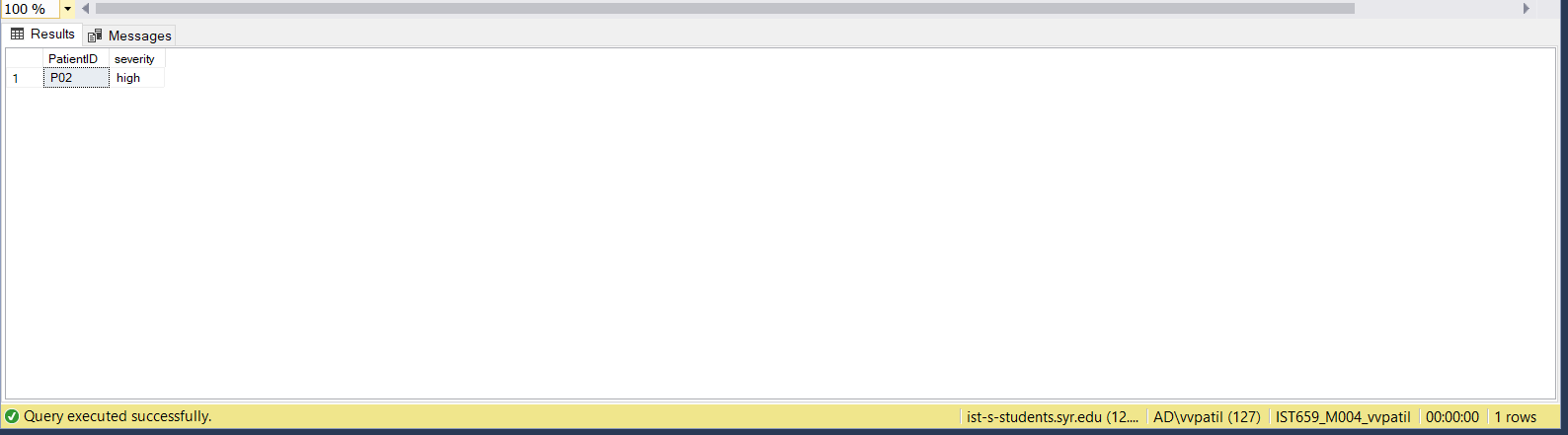
* **SQL Query**

SELECT PatientID,severity

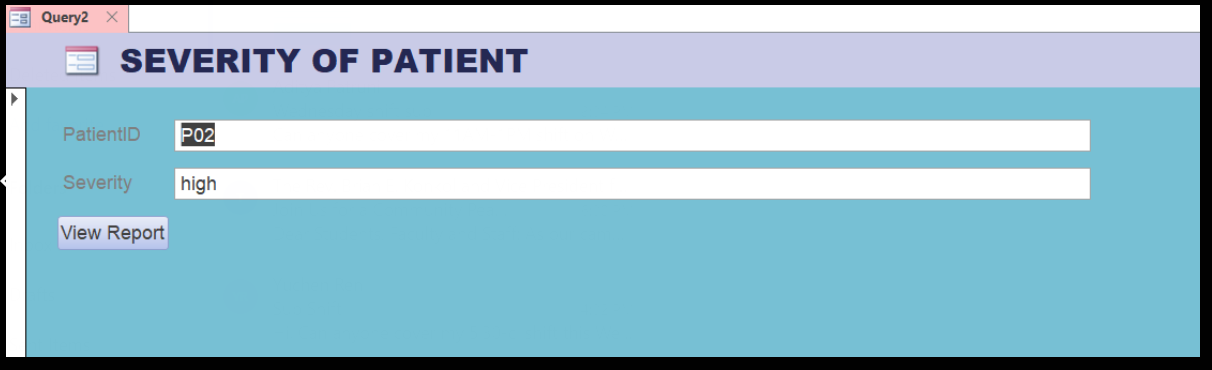
FROM Appointment\_Diagnosis

WHERE PatientID='P02'

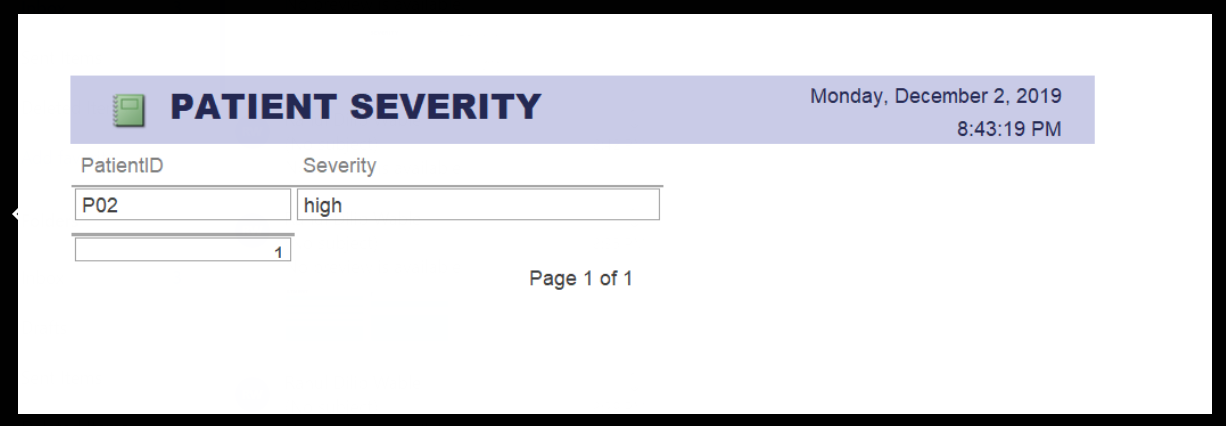
* **Output (Datasheet)**



* **Output (Form)**



* **Output (Report)**



1. **What is the type of treatment given to a patient?**

* **SQL Query**

SELECT p.PatientID,p.Patient\_Firstname,p.Patient\_Lastname,d.DiagnosisDescription

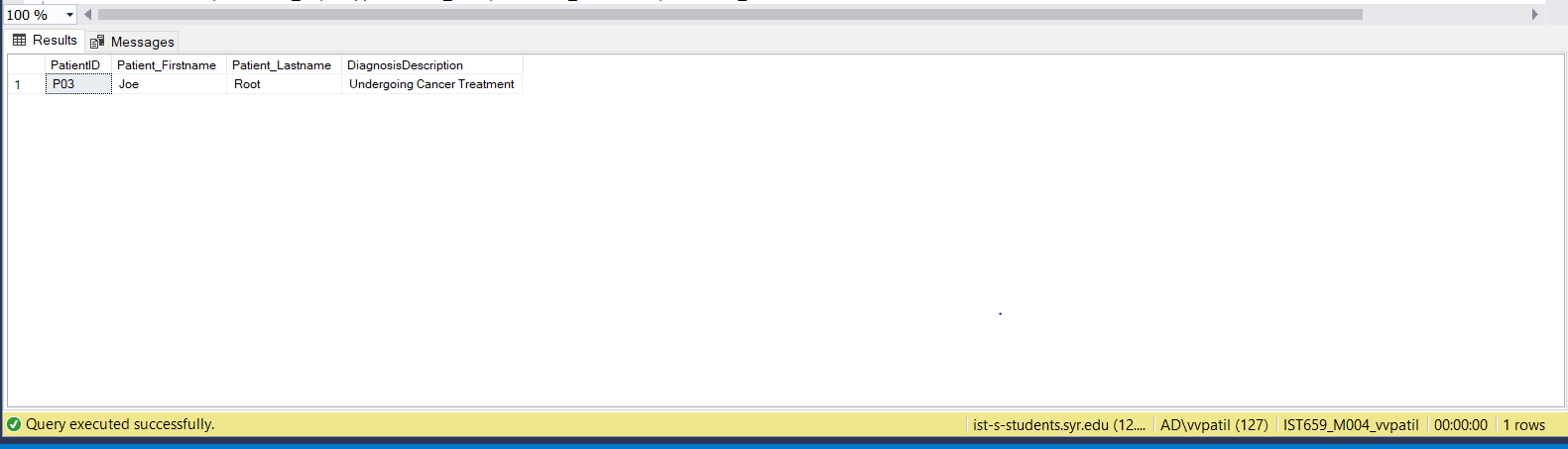
FROM Diagnosis d

JOIN Appointment\_Diagnosis ad On d.DiagnosisID=ad.DiagnosisID

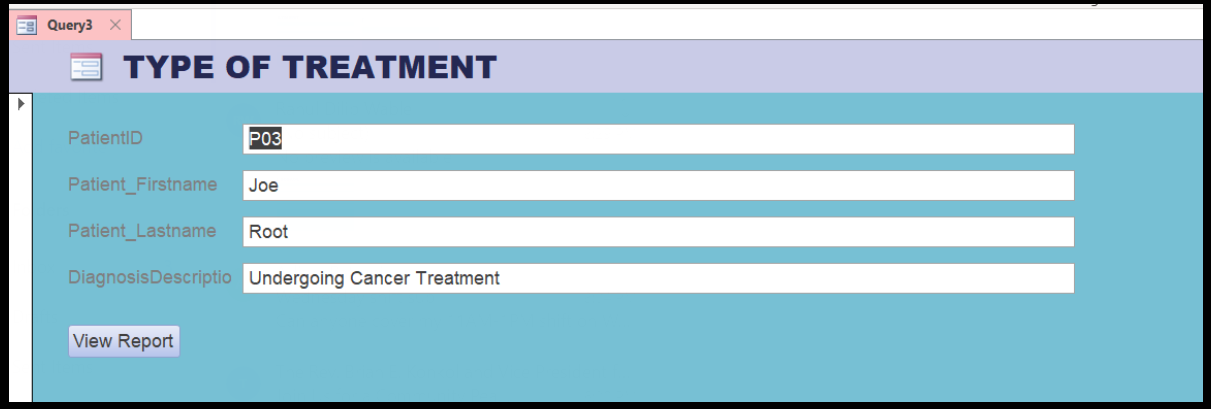
JOIN Patient p ON ad.PatientID=p.PatientID

WHERE p.PatientID='P03'

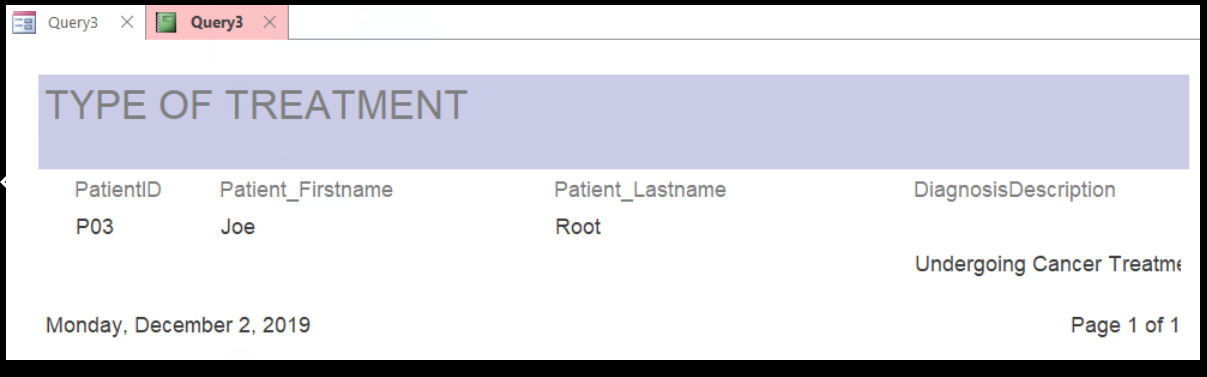
* **Output (Datasheet)**



* **Output (Form)**



* **Output (Report)**



1. **How many patients visited the same Doctor?**

* **SQL Query**

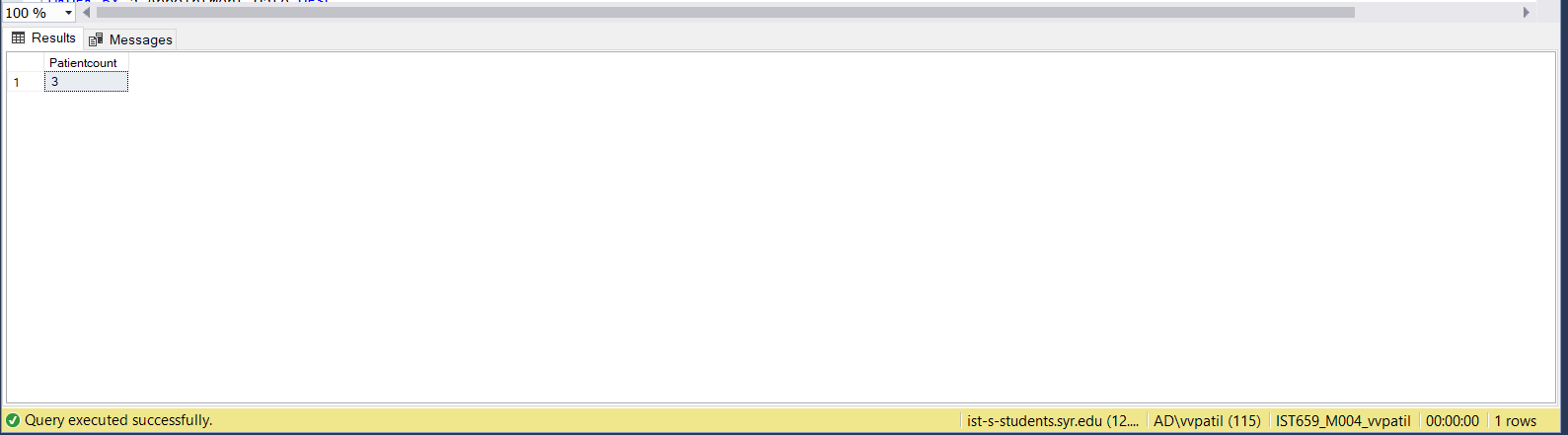
SELECT Count(PatientID) AS Patientcount

FROM Appointment a

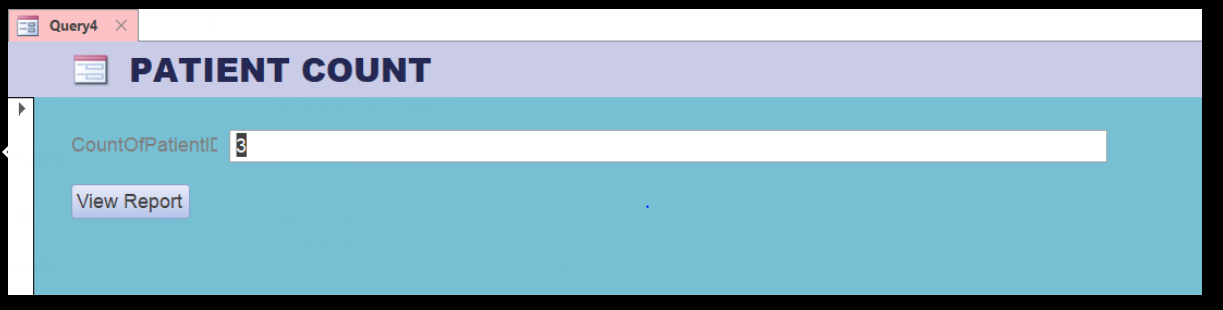
JOIN Doctor d ON d.Doctor\_ID= a.Doctor\_ID

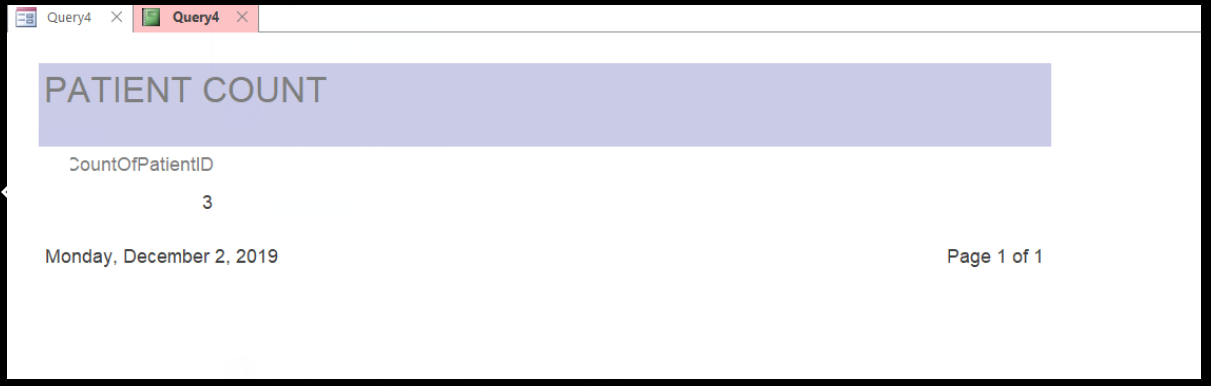
WHERE d.Doctor\_ID='D01'

* **Output** **(Datasheet)**



* **Output (Form)**
* **Output (Report)**





1. **What is the Patient ID with respect to descending Appointment date?**

* **SQL Query**

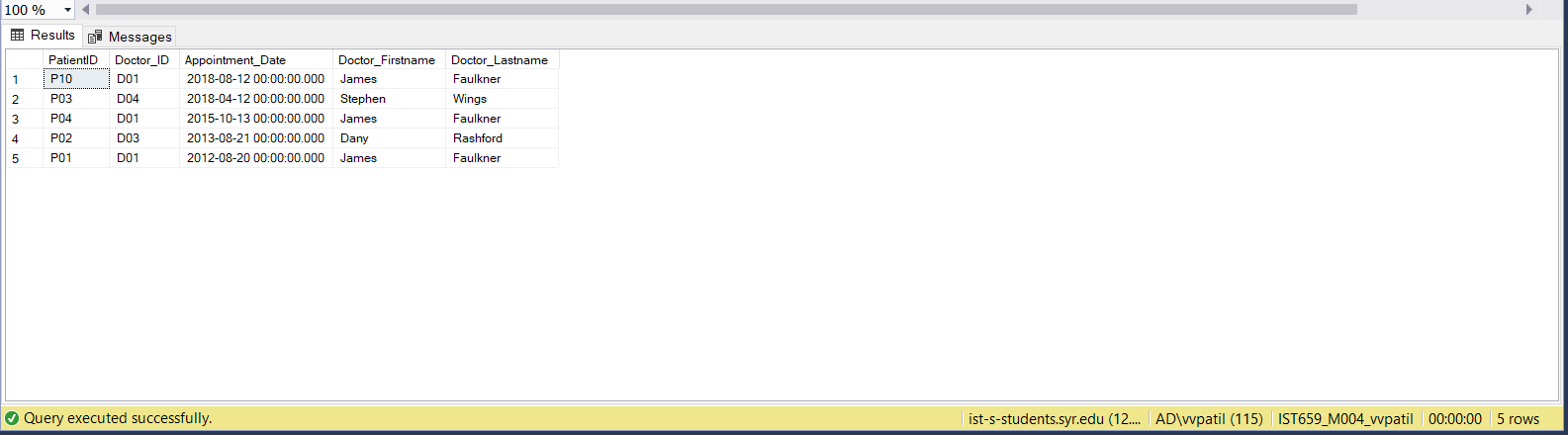
SELECT a.PatientID,d.Doctor\_ID,a.Appointment\_Date,d.Doctor\_Firstname,d.Doctor\_Lastname

FROM Appointment a

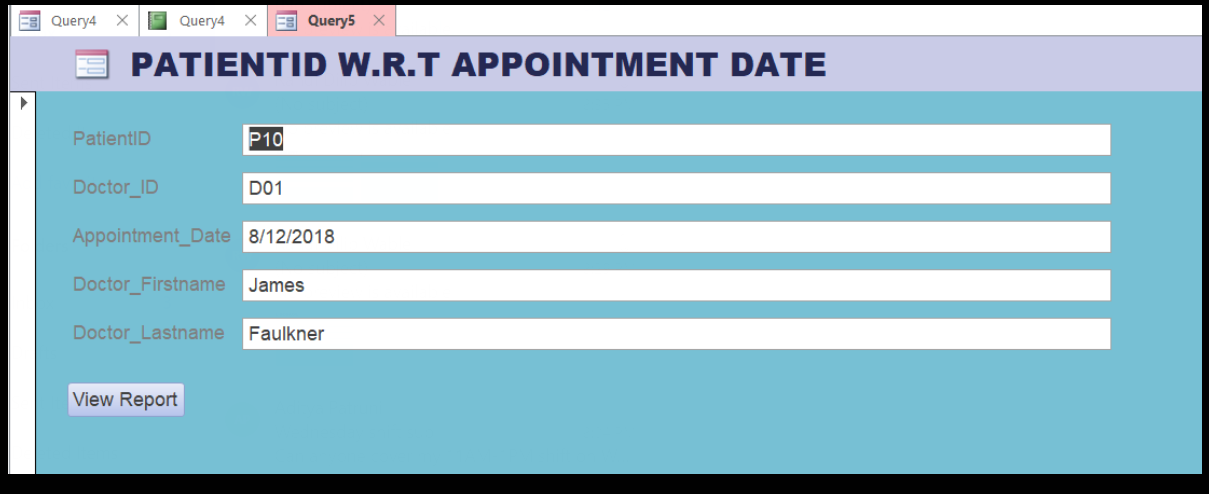
JOIN Doctor d on a.Doctor\_ID=d.Doctor\_ID

ORDER BY a.Appointment\_Date DESC

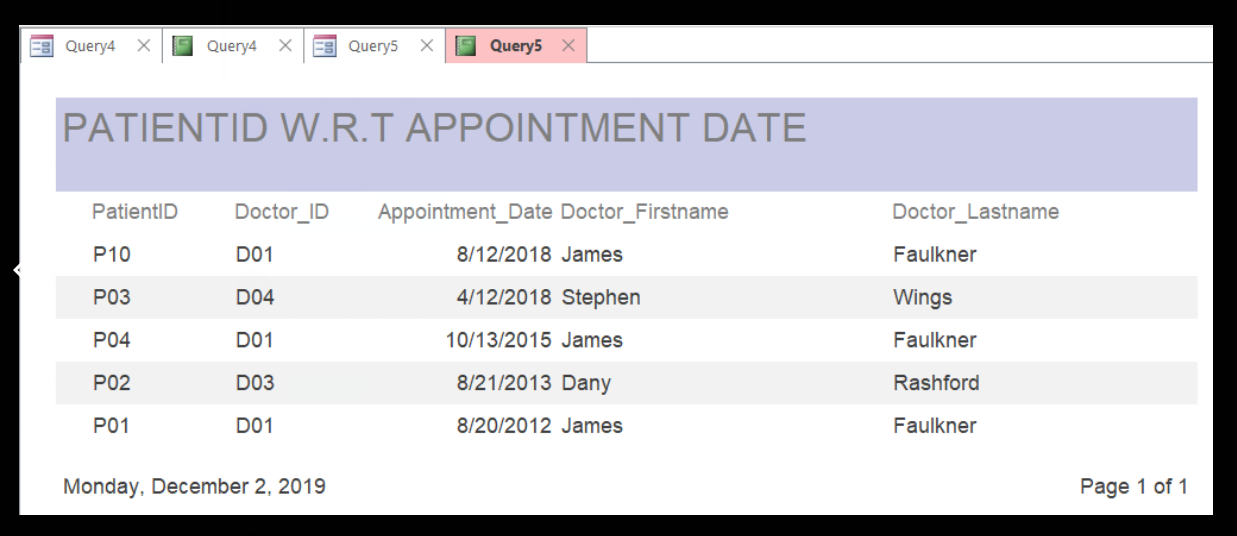
* **Output (Datasheet)**



* **Output (Form)**



* **Output (Report)**



1. **Which patients visited the same Doctor ?**

* **SQL Query**

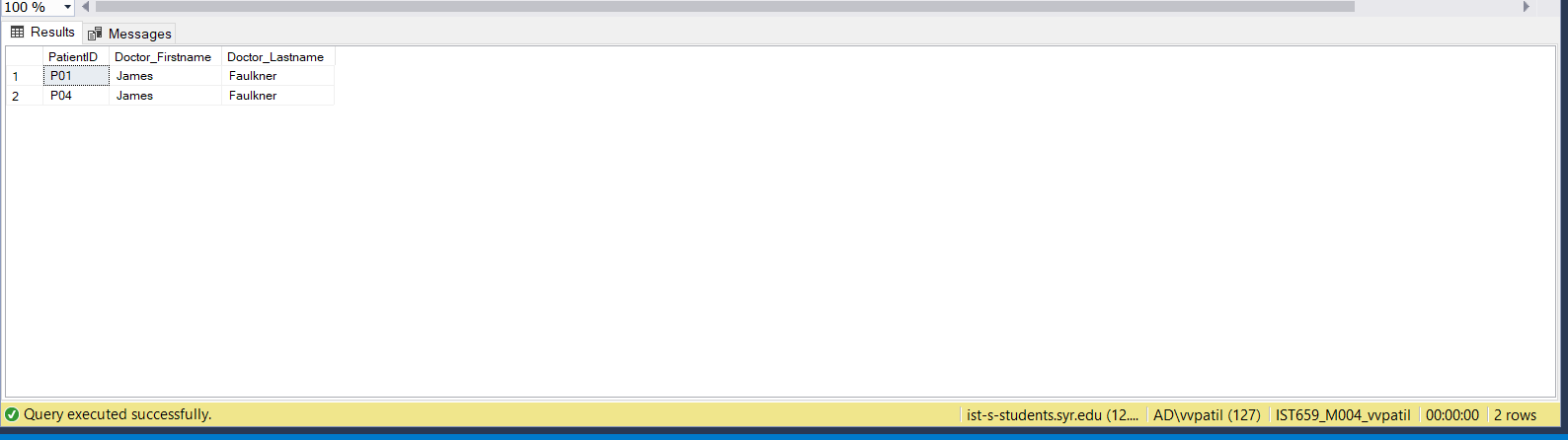
SELECT a.PatientID,d.Doctor\_Firstname,d.Doctor\_Lastname

FROM Doctor d

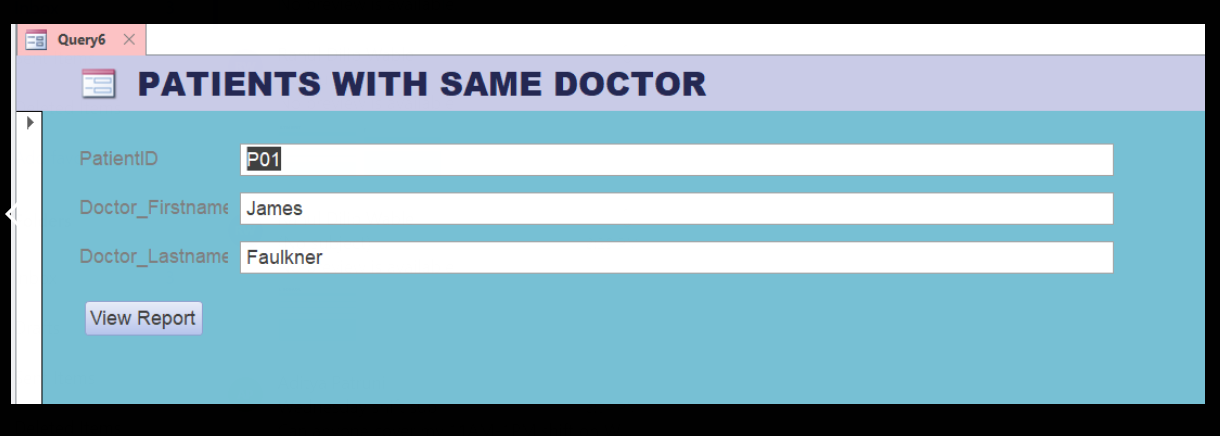
JOIN Appointment a ON a.Doctor\_ID=d.Doctor\_ID

WHERE a.Doctor\_ID='D01'

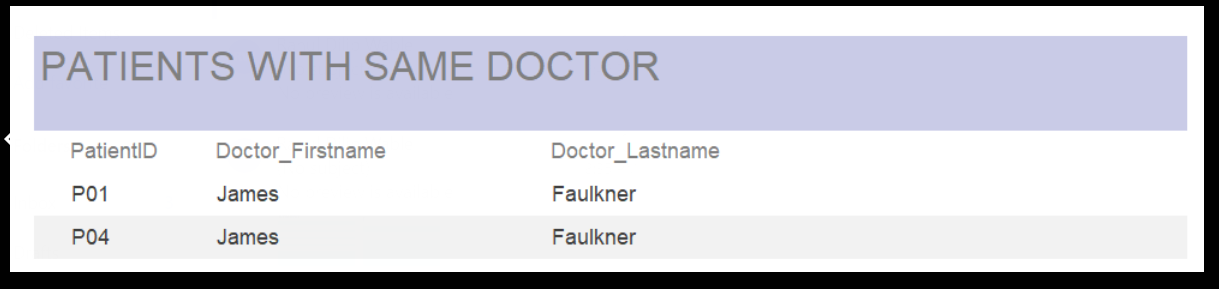
* **Output (Datasheet)**



* **Output (Form)**



* **Output (Report)**



**VIEWS IN SQL & MS Access**

**SQL Query:**

CREATE VIEW [Patient\_History]

As

SELECT

a.Appointment\_Date,

p.PatientID,p.Patient\_Firstname,p.Patient\_Lastname,

d.Doctor\_ID,D.Doctor\_Firstname,d.Doctor\_Lastname,

dg.DiagnosisDescription,

m.MedicineName

FROM Appointment a

JOIN Doctor d ON a.Doctor\_ID=d.Doctor\_ID

JOIN Patient p ON a.PatientID=p.PatientID

JOIN Appointment\_Diagnosis ad ON a.Appointment\_Date=ad.Appointment\_Date

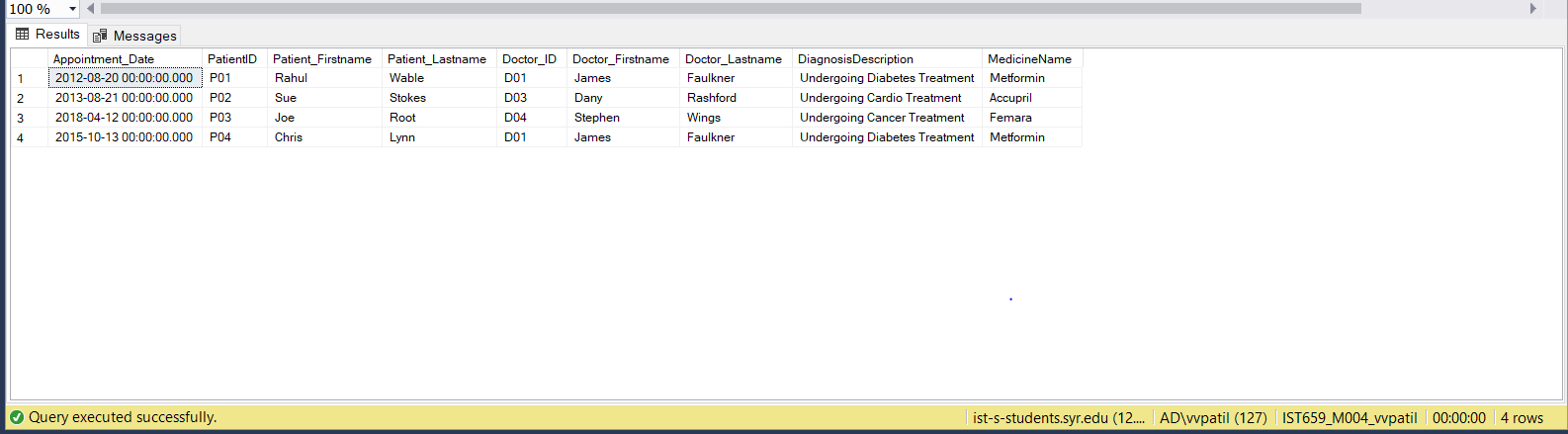
JOIN Diagnosis dg ON ad.DiagnosisID=dg.DiagnosisID

JOIN Prescribed\_Medicine pm ON p.PatientID=pm.PatientID

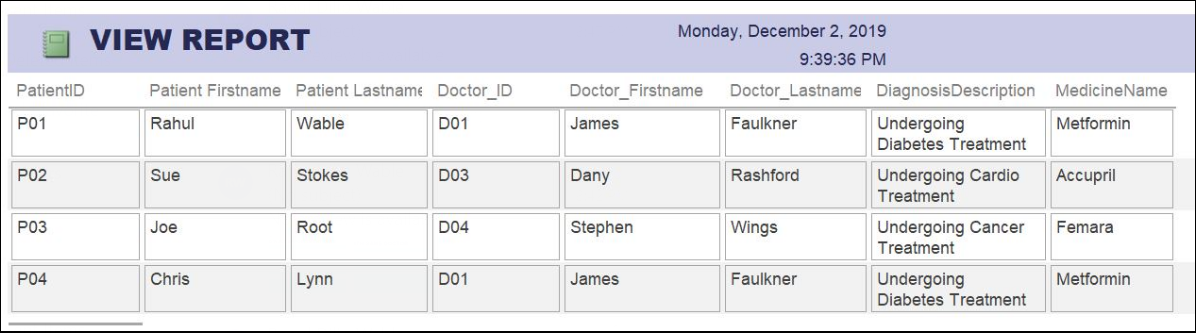
JOIN Medicine m ON pm.Medicine\_ID=m.Medicine\_ID

SELECT \* FROM [Patient\_History]

**Output:**



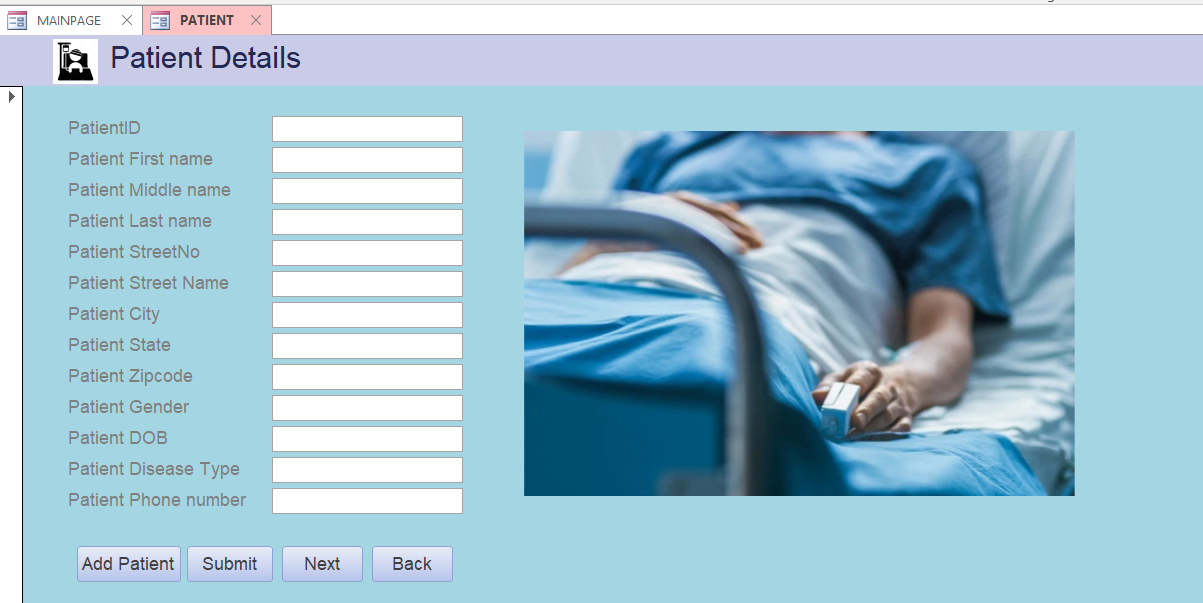
**Output (Report):**

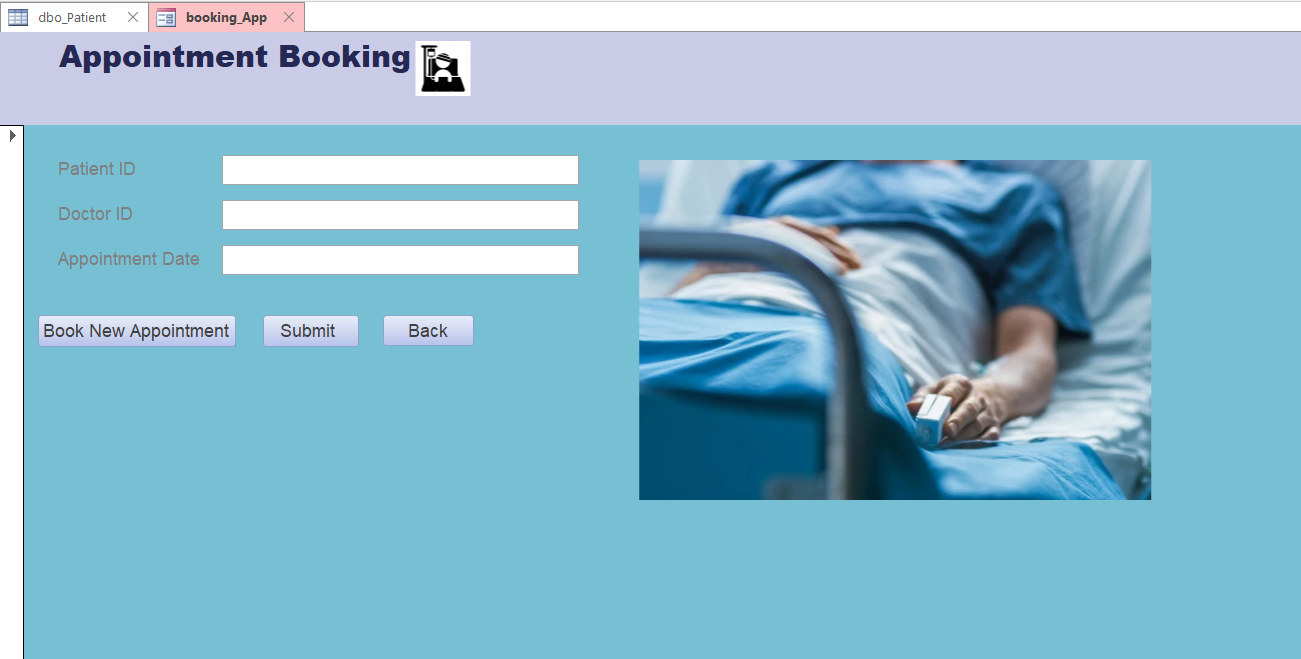


**INTERFACES**

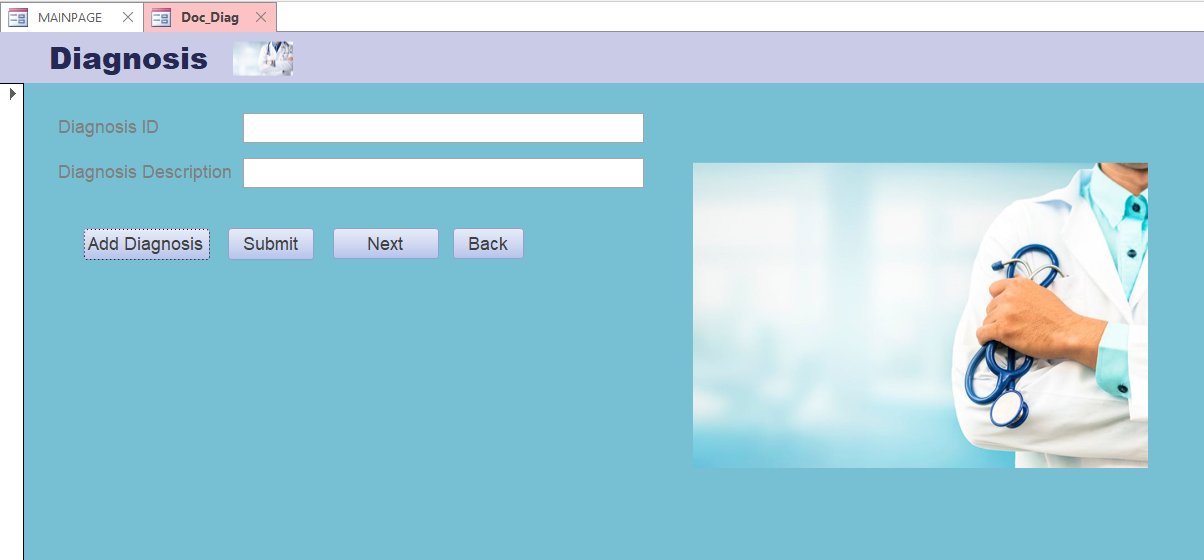
**The users of the system are categorized into three groups:**

1. **Patient**

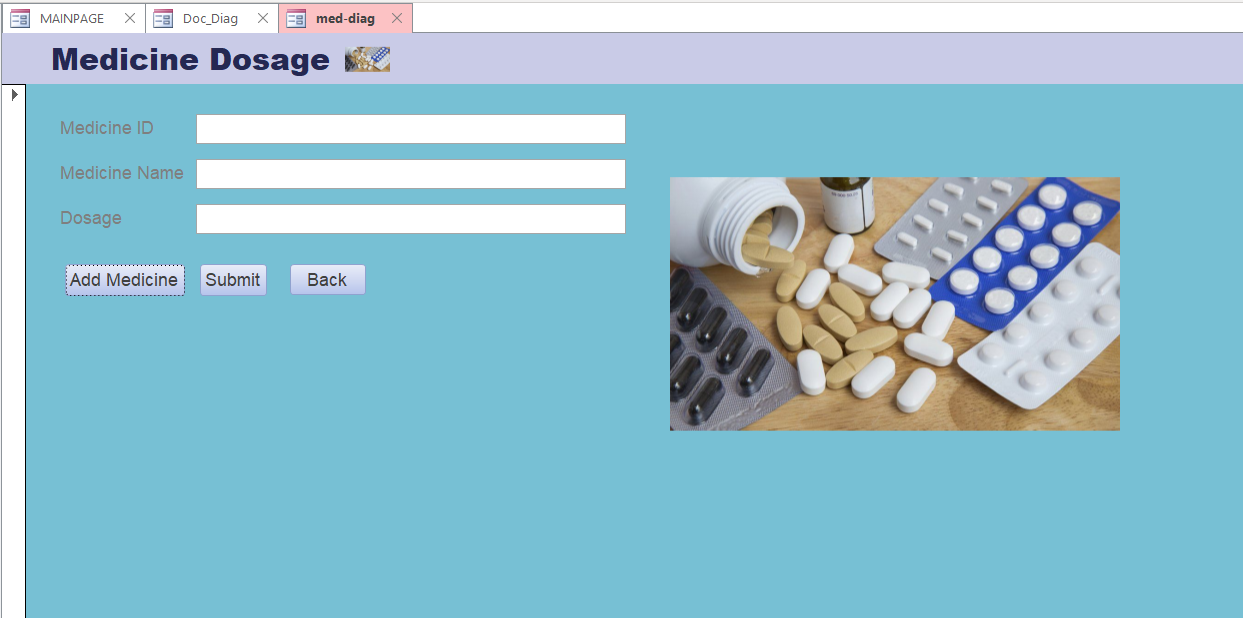
**STEP 1: Patient enters the details like**

**STEP 2: Patient books an appointmen**

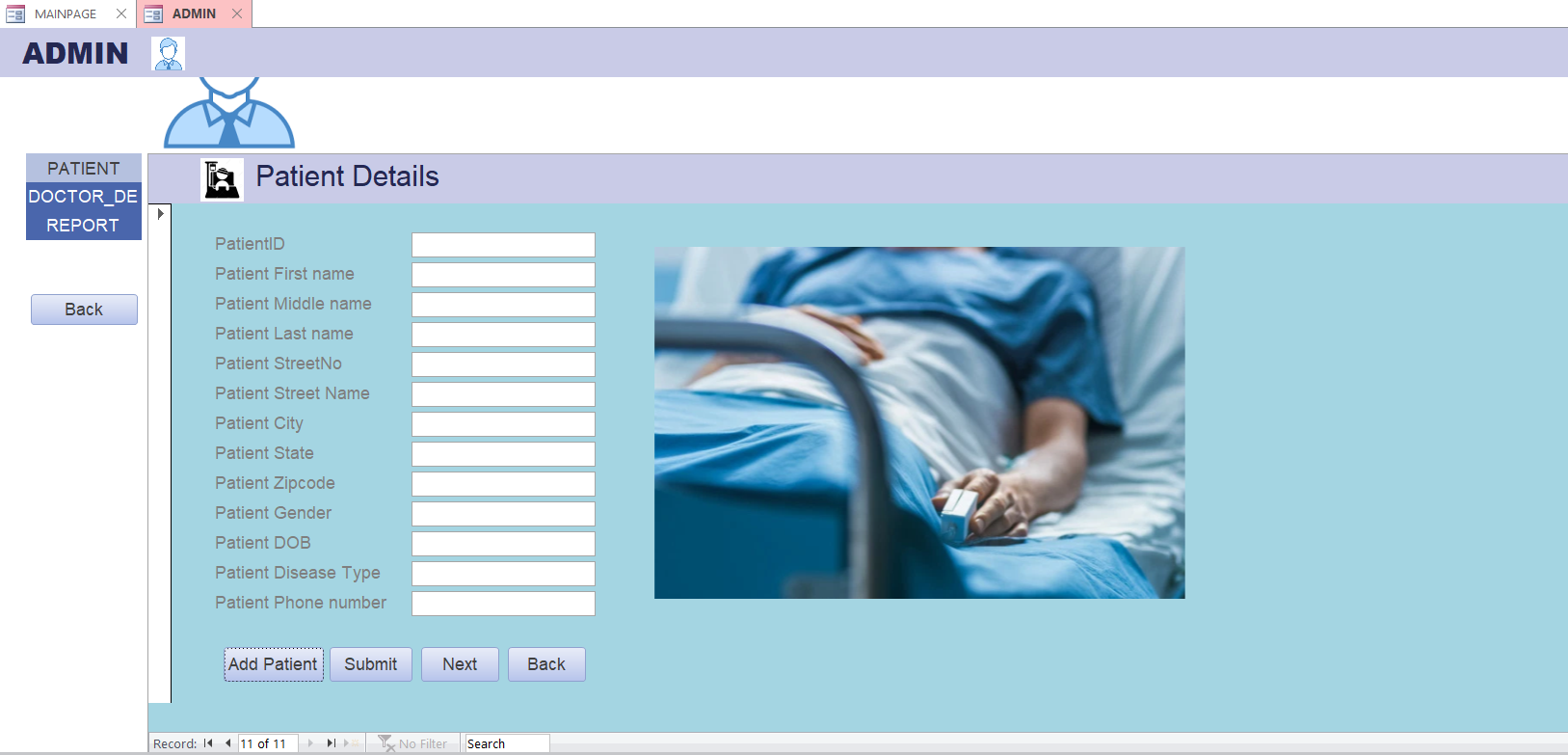
**2.Doctor**

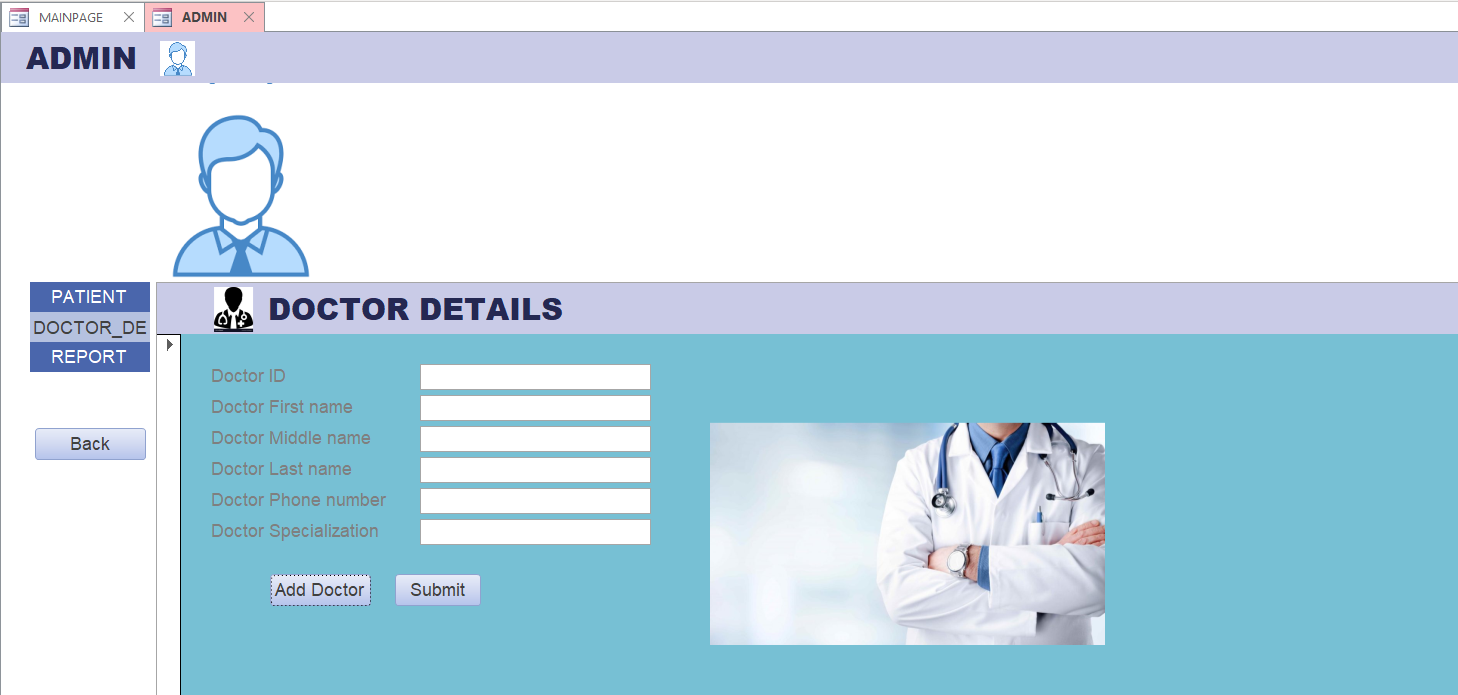
**STEP 1: Doctor enters the patients diagnosis details**

**STEP 2: Doctor will prescribe dosage of medicines**

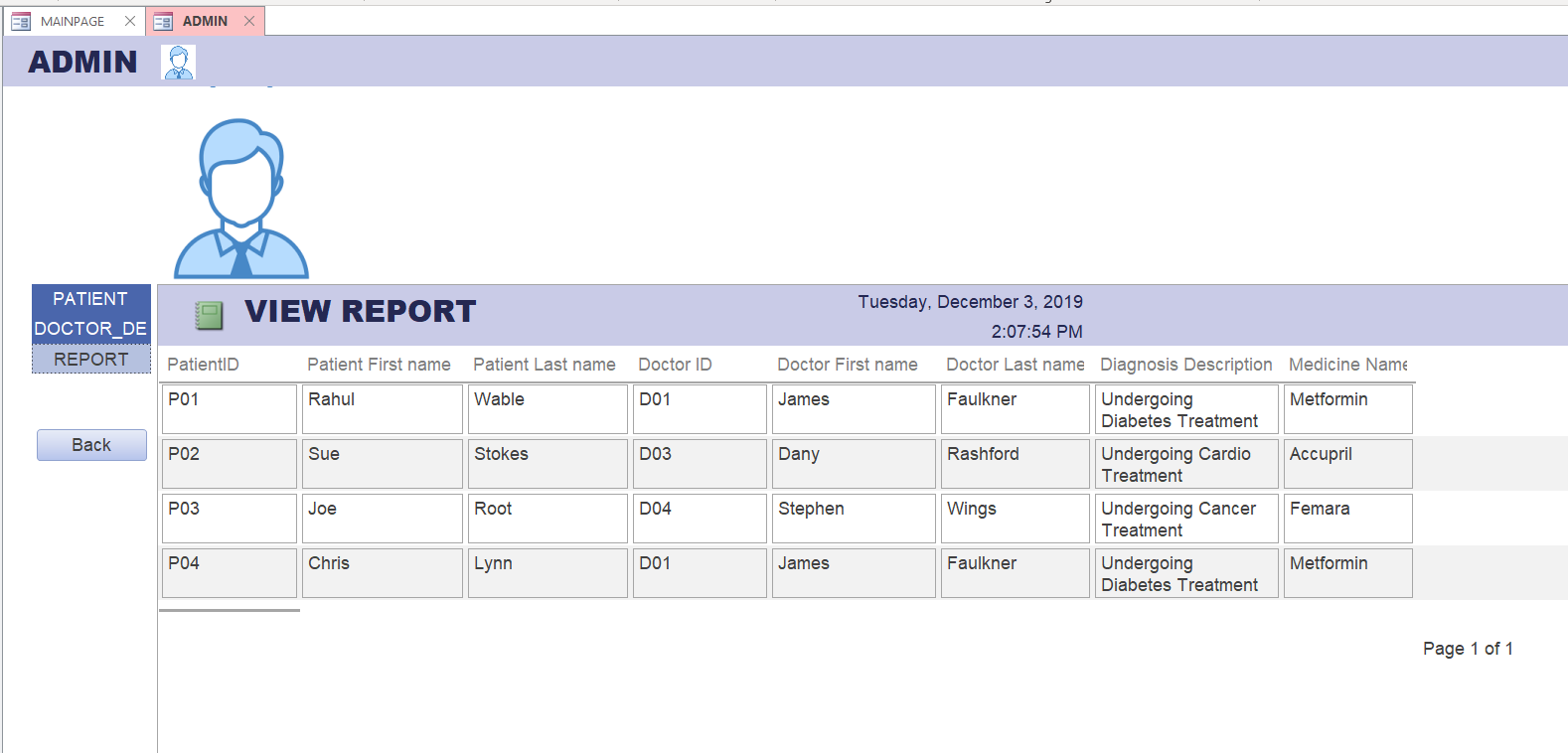


**3.Admin**

**STEP 1: Admin will enter the patient details if in case patient can’t enter the details himself.**

 **STEP 2: Details of Doctor will be entered that will be entered by the Admin.**

**STEP 3:Generate Report**



**TRIGGERS**

**SQL Query:**

CREATE TRIGGER [updateDosage]

ON Medicine

AFTER INSERT,DELETE

AS BEGIN

SET NOCOUNT ON;

INSERT INTO meddosage(

Medicine\_ID ,

MedicineName ,

Manf\_Date ,

Exp\_Date,

Dosage,

updated\_at,

operation

)

SELECT

i.Medicine\_ID ,

MedicineName ,

Manf\_Date ,

Exp\_Date,

Dosage,

GETDATE(),

'INS'

FROM INSERTED AS i

UNION ALL

SELECT

d.Medicine\_ID ,

MedicineName ,

Manf\_Date ,

Exp\_Date,

Dosage,

GETDATE(),

'DEL'

FROM DELETED AS d;

END

INSERT INTO Medicine Values('M05','Soframycin','08/22/2012','08/01/2015','450')

SELECT \* FROM meddosage

**Output:**

