**Proposed System**

The main aim of this project is to give information about the Functioning of Databases in Hospital System. A hospital database will allow creating and maintaining database including information of all the patients, all the hospital employee’s including the doctors and the receptionist and the money transactions or the payment happening between the hospital and the patients which will be helpful in keeping a track of all sorts of transactions.

Patients entering the hospital will need to register first by filling a form at the reception. Each form has a unique number whose domain will be like all-natural numbers which will be assigned to that patient in the form of pat\_id with the integer datatype and will be getting stored into the patient information database. This database will also include the purpose of visit and along with the name of the doctor they want to see. There will be database containing an information about the doctors within the hospital. Each doctor will be assigned a unique identification number whose domain will be like all-natural numbers. There will be another database containing information about all the employees working in the hospital. Each of these employees will be provided a unique identification number whose domain will be like all-natural numbers.

The user groups involved with respect to Hospital database are,

1. Receptionist
2. Doctors
3. Hospital Staff (Accountant)

**Requirements:**

Hospital database is a vast application and in real time, it involves many users and in this project, we will focus only on three users and their uses in the system.

**Receptionist:** The receptionist will be responsible for managing the appointments with the doctor. Also, the receptionist needs to keep track about the information of all the patients coming in and going out. Each patient who fills up the form will be assigned a particular unique id available on the form. Also, the receptionist can access the database to help the visitors by retrieving all the information about the patients and help them out in the best way. They can access the doctor information like the name of the doctor and their availability etc.

**Doctors:** The Doctor user group can view the database and review the activities going around with the patients. They need to see the lab reports of the patients in order to provide treatment accordingly. From the lab reports, the doctors can decide whether that particular patient need to get admitted or not. For that each patient and doctor should be uniquely defined and should follow the appropriate constraints. According to the type of the treatment given, the patient will be charged and given a room by the member of the staff working under that doctor.

**Accountant:** Accountants will be responsible for handling billing of patients going out or getting discharged. They will be accessing the database in order to check whether or what amount of

advances has been given by the patients and how much more they need to pay. They are also responsible for updating the billing info with the time. The patient will be charged accordingly depending upon the room or the treatment that has been given.

**Overall Objective:**

* Registration: Patients register by filling visiting form.
* Patients Details: Patient Bio data.
* Appointment: Reception set the appointment with the available doctor for the patients who wants to see the doctor for consultation. Other than those, the patients don’t need to take the appointment specially for emergency cases.
* Doctors: Doctors Information + appointment
* Consultation: Doctor checks patient
* Lab report: Patient is either admitted or discharged as per the lab result.
* Staff: Hospital employee information
* Bill: Patient is finally billed out and get discharge.

**Domain Requirements:**

The domain requirements are as following:

1. All the patients, doctors as well as the hospital employees are assigned a unique identification number.
2. All the unique number assigned are not nullable, i.e. they can’t be assigned a null value.
3. The minimum salary of any doctor should be $50000

**Project Normalization**

1. **UNF:**

For our database to be in UNF, let us bring all the attributes from our relations into one relation, say Hospital. Hence our new relation in UNF looks like this.

**Hospital**(pat\_id, pat\_name, pat\_fname, date\_of\_bath, pat\_gender, pat\_street\_address, pat\_city, pat\_state, pat\_country, pat\_postal\_code, pat\_contact, pat\_weight, other\_det, pat\_reg\_date, doc\_id, doc\_name, doc\_fname, doc\_gender, doc\_street\_address, doc\_city, doc\_state, doc\_country, doc\_postal\_code, doc\_contact, designation, salary, lab\_id, lab\_rep\_weight, lab\_rep\_date, lab\_charge, diag\_details, remark, other, staff\_id, staff\_Name, staff\_fname, dept\_name, staff\_gender, staff\_street\_address, staff\_city, staff\_state, staff\_country, staff\_postal\_code, staff\_contact, room\_num, adm\_disch\_date, adm\_disc\_flag, out\_pat\_date, room\_type, tot\_beds, num\_beds\_occupd, status, floor\_num, bill\_id, doc\_charge, med\_charge, room\_charge, operation\_charge, num\_of\_days, staff\_charge, apointm\_datetime, app\_id, app\_date, app\_datetime)

1. **1NF:**

For our database to be in 1NF, it is necessary for us to follow the following rules:

* 1. There must not be any multivalued attribute.
  2. There must be no repetition.
  3. Primary key must be defined.

The database follows the first two rules, but we need to assign a primary key. The set of primary keys that define the database are

* + - **pat\_id**
    - **doc\_id**
    - **lab\_id**
    - **date\_of\_adm**
    - **staff\_id**
    - **room\_no**
    - **out\_pat\_date**
    - **bill\_id**
    - **app\_id**

Also, together these are unique.

1. **2NF:**

For our database to be in 2NF, we need to make sure that all non-key fields must depend on all the primary keys. For our table to be in 2NF we need to make the following changes:

1. Patient Information depends upon the patient ID that will be getting assigned to them while registration. Therefore, pat\_name, pat\_fname, date\_of\_birth, pat\_gender, pat\_street\_address, pat\_city, pat\_state, pat\_country, pat\_postal\_code, pat\_contact, pat\_weight, other\_det, pat\_reg\_date depends upon the pat\_id.
2. Doctor Information depends upon the Doctor ID and not on other primary keys. Therefore, doc\_name, doc\_fname, doc\_gender, doc\_street\_address, doc\_city, doc\_state, doc\_country, doc\_postal\_code, doc\_contact, designation, salary, doc\_charge depends upon doc\_id.
3. Then lab\_rep\_weight, lab\_rep\_date, diag\_details, remark, lab\_charge, other depends upon lab\_id.
4. Then staff\_name, staff\_fname, dept\_name, staff\_gender, staff\_street\_address, staff\_city, staff\_state, staff\_country, staff\_postal\_code, staff\_contact, staff\_charge depends upon staff\_id.
5. Then adm\_disch\_date, adm\_disc\_flag, no\_of\_days depends upon pat\_id, room\_no.
6. Then room\_type, tot\_beds, num\_beds\_occupd, status, floor\_num, room\_charge depends upon room\_num.
7. doc\_charge, staff\_charge, lab\_charge, med\_charge, operation\_charge, room\_charge, bill\_datetime depends upon bill\_id
8. Then app\_date, app\_datetime depends upon app\_id, doc\_id and pat\_id.
9. lab\_date depends upon the lab\_id, pat\_id, doc\_id as this is the junction table of the Patient, Doctor and Lab information relation.

**Hence the new relation schema is:**

**Patientinfo** (pat\_id, pat\_name, pat\_fname, date\_of\_birth, pat\_gender, pat\_street\_address, pat\_city, pat\_state, pat\_country, pat\_postal\_code, pat\_contact, pat\_weight, other\_det, pat\_reg\_date)

**Doctorinfo** (doc\_id, doc\_name, doc\_fname, doc\_gender, doc\_street\_address, doc\_city, doc\_state, doc\_country, doc\_postal\_code, doc\_contact, designation, salary, doc\_charge)

**Doc\_Staff** (doc\_id, staff\_id) [Doc\_Staff relation is the relation which indicates the staff which has been appointed as a helper to the doctor. So, each doctor can have just one helper for example, nurse or ward boy and each doctor will be having a helper]

**Labinfo** (lab\_id, lab\_rep\_weight, lab\_rep\_date, diag\_details, remark, other, lab\_charge)

**Assist** (lab\_id, pat\_id, doc\_id, lab\_rep\_date)

**Staffinfo** (staff\_id, s\_Name, staff\_fname, dept\_name, staf\_gender, staff\_gender, staff\_street\_address, staff\_city, staff\_state, staff\_country, staff\_postal\_code, staff\_contact, staff\_charge)

**Inpatientinfo** (room\_num, pat\_id, adm\_disch\_date, adm\_disc\_flag, num\_of\_days)

**Outpatientinfo** (pat\_id, out\_pat\_date)

**Roominfo** (room\_num, room\_type, tot\_beds, num\_beds\_occupd, status, floor\_num, room\_charge)

**Medicalcharge** (bill\_id, doc\_charge, lab\_charge, staff\_charge, med\_charge, operation\_charge, room\_charge, bill\_datetime)

**Bill\_Pat (**bill\_id, pat\_id)

**Schedinfo** (app\_id, pat\_id, doc\_id, app\_date, app\_datetime)

**4) 3NF:**

For our database to be in 3NF, we need to make sure that, no non-key value depends upon one another.

Hence, these are the following changes that we must make:

1. As salary and doc\_charge is based upon the designation of the doctor, we need to separate it out and need to create another relation.
2. Also as room charge is based upon the type of the room, we need to separate it out and need to create another relation.
3. Also as staff charge is based upon the person working in a department, so we need to separate it out and need to create another relation.
4. Unfortunately, it does not have the ability to add timestamp twice so we had to remove one of them So we decided to replace date\_of\_disc to adm\_disch\_flag char(1) which will indicate whether the timestamp value represents the admission date ('A') or the discharge date ('D').

Hence the relation schema looks like this:

**Patientinfo** (pat\_id, pat\_name, pat\_fname, date\_of\_birth, pat\_gender, pat\_street\_address, pat\_city, pat\_state, pat\_country, pat\_postal\_code, pat\_contact, pat\_weight, other\_det, pat\_reg\_date)

**Doctorinfo** (doc\_id, doc\_name, doc\_fname, doc\_gender, doc\_street\_address, doc\_city, doc\_state, doc\_country, doc\_postal\_code, doc\_contact, designation)

**Dr\_design** (designation, salary, doc\_charge)

**Doc\_Staff** (doc\_id, staff\_id)

**Labinfo** (lab\_id, lab\_rep\_weight, lab\_rep\_date, diag\_details, remark, other, lab\_charge)

**Assist** (lab\_id, pat\_id, doc\_id, lab\_rep\_date)

**Staffinfo** (staff\_id, staff\_Name, staff\_fname, dept\_name, staff\_gender,

staff\_street\_address, staff\_city, staff\_state, staff\_country, staff\_postal\_code, staff\_contact)

**Staff\_dept\_charge (**dept\_name, staff\_charge)

**Inpatientinfo** (room\_num, pat\_id, adm\_disch\_date, adm\_disc\_flag, num\_of\_days)

**Outpatientinfo** (pat\_id, out\_pat\_date)

**Roominfo** (room\_num, room\_type, tot\_beds, num\_beds\_occupd, status, floor\_num)

**Medicalcharge** (bill\_id, doc\_charge, lab\_charge, staff\_charge, med\_charge, operation\_charge, room\_charge, apointm\_datetime)

**Inpatientinfo** (room\_num, pat\_id, adm\_disch\_date, adm\_disc\_flag, num\_of\_days)

**Outpatientinfo** (pat\_id, out\_pat\_date)

**Roominfo** (room\_num, room\_type, tot\_beds, num\_beds\_occupd, status, floor\_num)

**Medicalcharge** (bill\_id, doc\_charge, lab\_charge, staff\_charge, med\_charge, operation\_charge, room\_charge, bill\_datetime)

**Roomcharge** (room\_type, room\_charge)

**Bill\_Pat (**bill\_id, pat\_id)

**Schedinfo** (app\_id, pat\_id, doc\_id, app\_date, app\_datetime)

Now, we can see all the above-mentioned relations are in 3NF. **Note:**

1. After normalization, I found out that Billinfo table needs to be calculated by joining other tables. So, I divided all the charges like medical charges, operation charges, doctor charges etc separately which I thought it would be correct way of doing as per the rules of normalization.
2. Also, primary keys for some of the relations are now different after applying normalization as compare to earlier.

Diagram

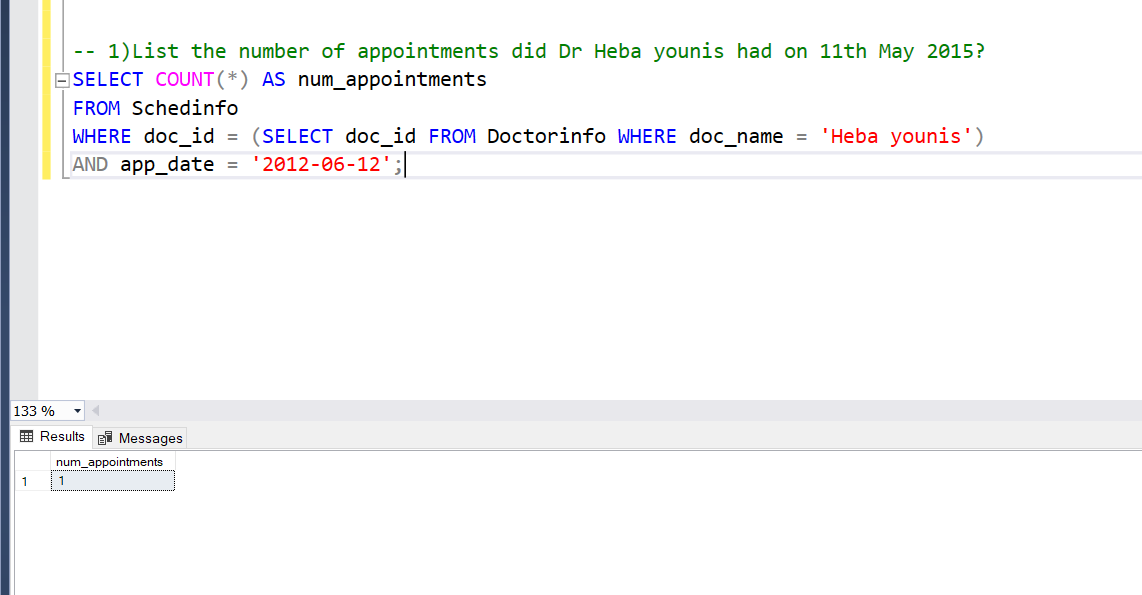
Description automatically generatedThe final Look of Database Schema

**The final Look of the ERD**

**User 1-For Receptionist:**

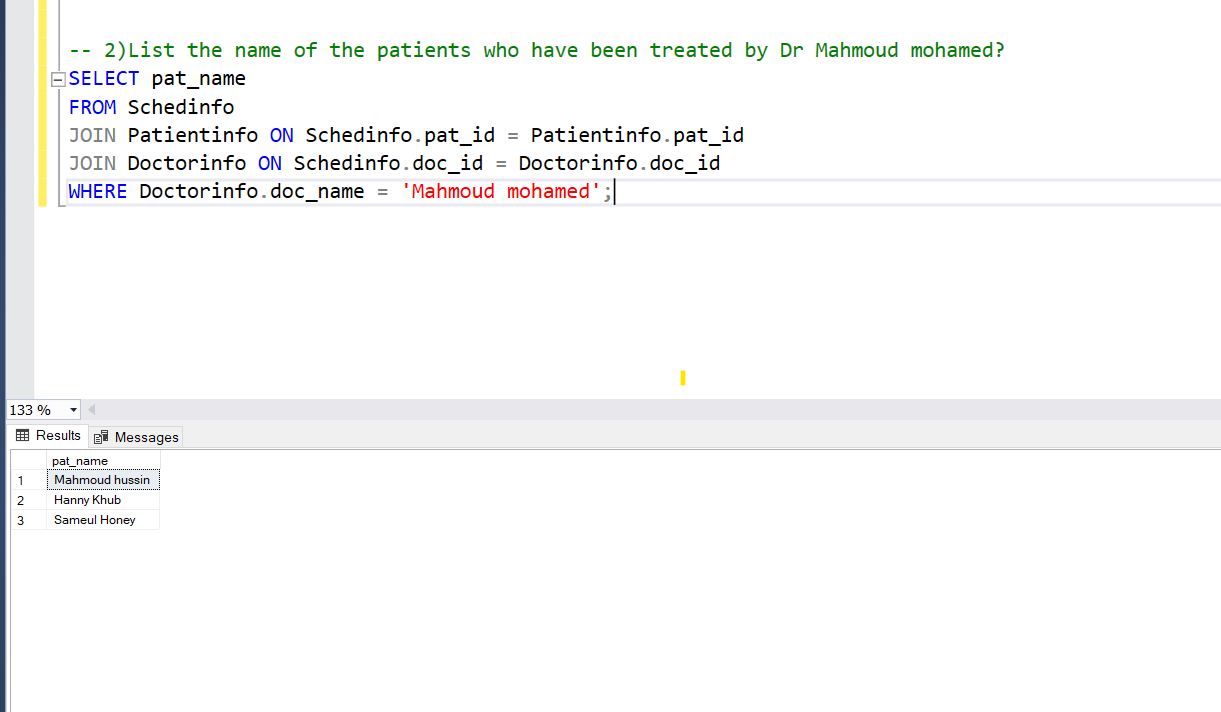
**Query 1**

1. **List the number of appointments did Dr Heba younis** **had on 11th May 2015?**



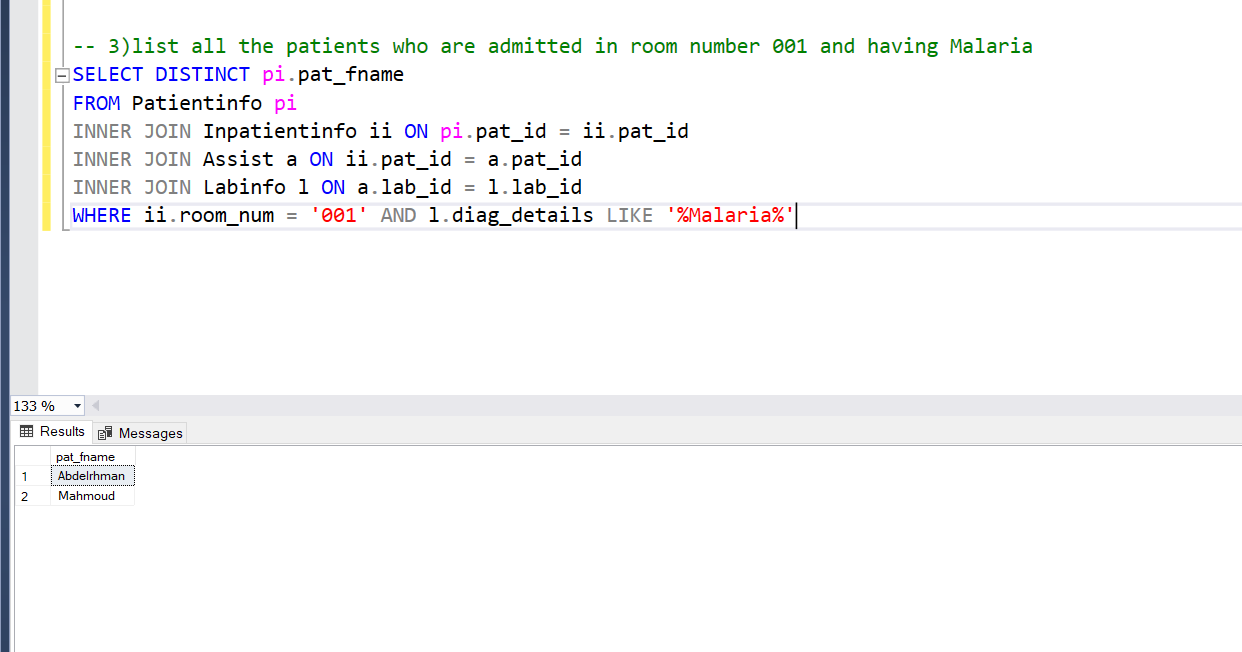
**Query 2**

1. **List the name of the patients who have been treated by Dr Mahmoud mohamed?**



**Query 3**

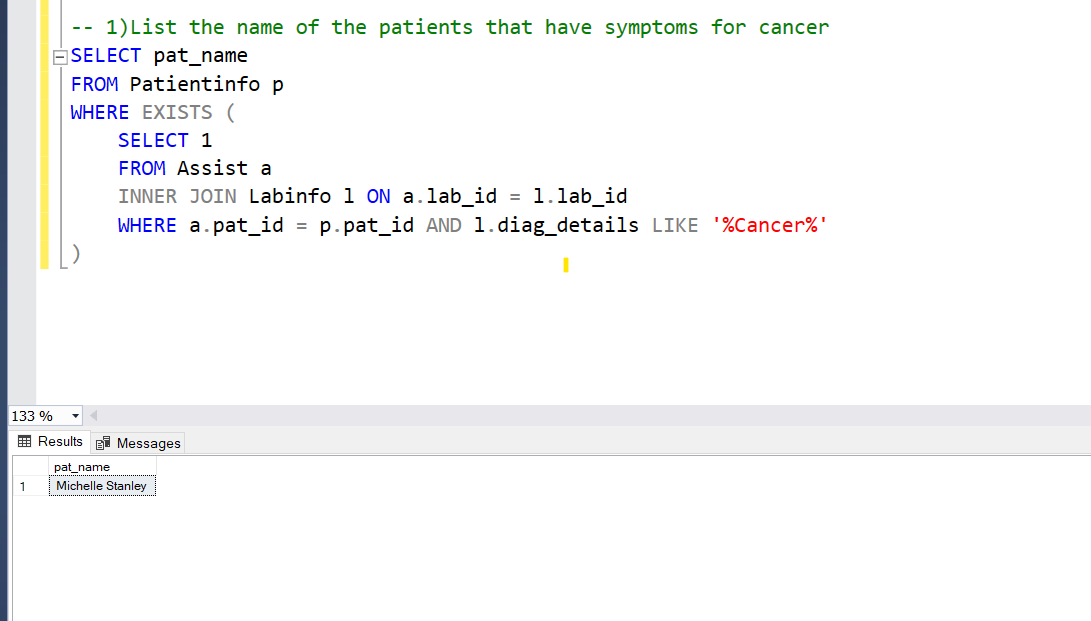
1. **List all the patients who are admitted in room no 001 and having Malaria?**



**User 2-For Doctor:**

**Query 1**

1. **List the name of the patients that have symptoms for cancer?**



**Query 2**

1. **List the appointments that had been scheduled so far with doctor Heba?**

**Graphical user interface, text, application

Description automatically generated**

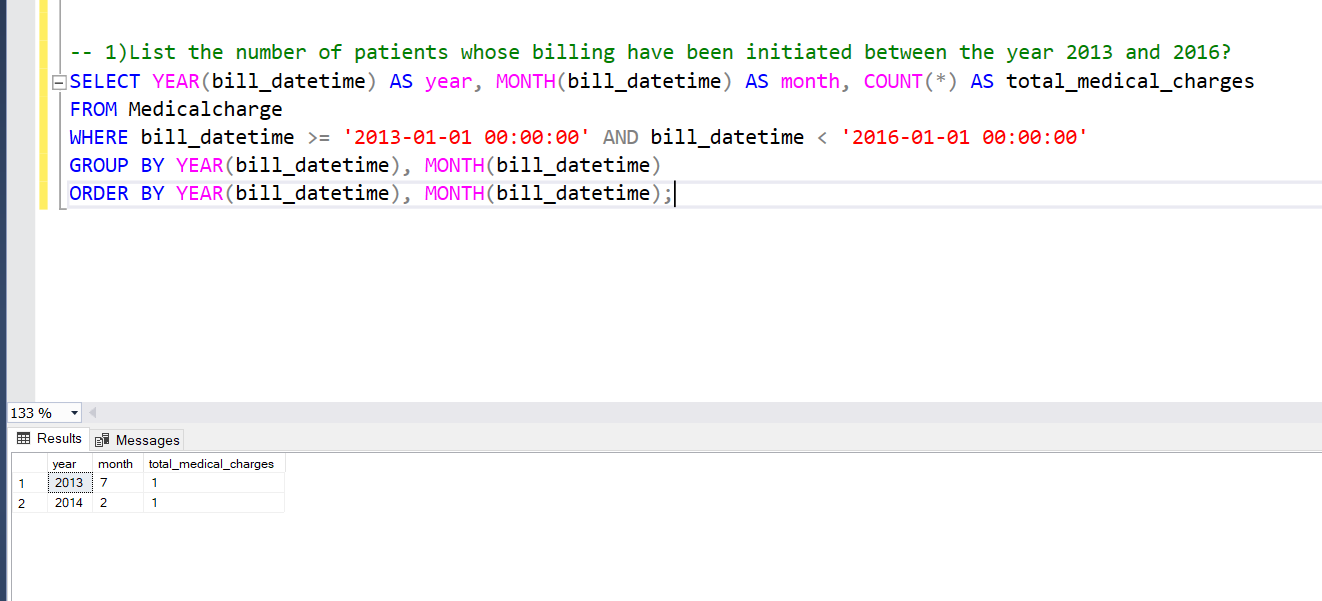
**Query 3**

1. **List the name of the staff that has been working with Dr Abdelrhman?**
2. Graphical user interface, application, Word

   Description automatically generated

**User 3-For Accountant:**

**Query 1**

1. **List the number of patients whose billing have been initiated between the year 2013 and 2016?** 

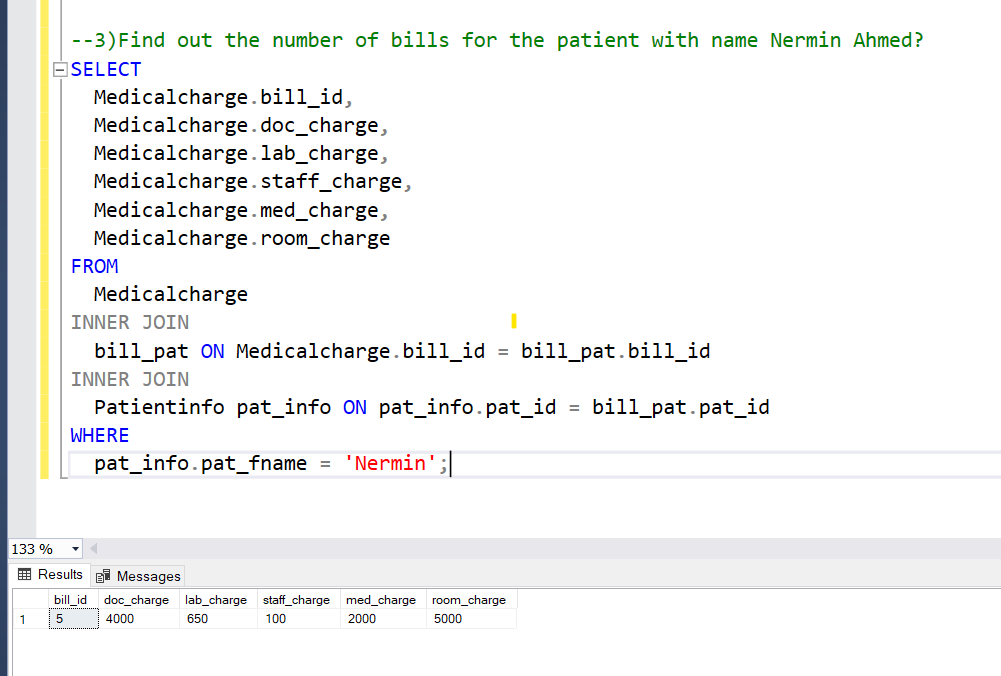
**Query 2**

1. **List the total bill of the patient with name Mahmoud hussin?**

Graphical user interface, text, application, email

Description automatically generated

**Query 3**

1. **Find out the number of bills for the patient with name Nermin Ahmed?**