PROJECT REPORT

A report submitted to the

Department of Electrical and Information Engineering

Faculty of Engineering

University of Ruhuna

Sri Lanka

On 11th of September 2023

In completing a Report for the module

EE4350 Database Systems

By:

Gunathilake A. P. B. - EG/2020/3947

Hanan Ahamed M. R. - EG/2020/3949

Hariharasakthy N. - EG/2020/3956

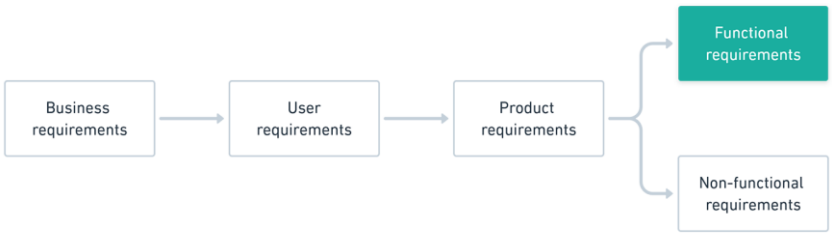
Table of Contents

Chapter 1 : Requirement Analysis

Here are the Functional Requirements and the Data Requirements of the Hospital DBMS.

1.1 Functional Requirements

The primary purpose of a Database Management System (DBMS) is to securely store data. For effective health management, a DBMS is very useful in current times. To keep track of the historical records of a hospital, for effective and efficient data handling, a DBMS is a very helpful tool for the hospital staff. In that case, this system will help in improving the quality of the service provided. Functional requirements are product features or the actions that developers must implement to be capable for the users to achieve their goals. These are the 'must do requirements' of a system.



Source: https://www.nuclino.com/img/articles/functional-requirements.png

In this project, we have identified 12 entities. They are "PATIENT", "DOCTOR", "NURSE", 'STAFF", "DEPARTMENT", "WARD", "APPOINTMENT", "TEST_REPORT", "MEDICINE", 'PRESCRIPTION", "BILL" "RECORD". All the entities are interconnected from many relations. The Entity-Relationship Diagram will be illustrated in the project report.

01) Registration Process: Add new patients, doctors, nurses, … records.
02) Modifying Entries: Modify patient, doctor, nurses, …. Records.
03) Deleting Entries: Removing records.
04) Modify data content.
05) Data Retrieval.

1.2 Data Requirements

Attributes of Entity "PATIENT":

- Patient_id (Should not be NULL)
- Gender
- Age
- Patient_name (Should not be NULL)
- Address
- Doctor_id (Should not be NULL)
- Test_id (Should not be NULL)
- Phone_number (Should not be NULL)
- Medicine_id (Should not be NULL)
- Prescription_id (Should not be NULL)

Attributes of Entity “DOCTOR”:

- Doctor_id (Should not be NULL)
- Specialization
- Gender
- Name_Birth (Should not be NULL)
- Address
- Phone_Number (Should not be NULL)

Attributes of Entity “NURSE”:

- Nurse_id (Should not be NULL)
- Gender
- Age
- Date_of_birth (Should not be NULL)
- Name (Should not be NULL)
- Patient_id (Should not be NULL)
- Phone_number (Should not be NULL)

Attributes of Entity “NON_MEDICAL_STAFF”:

- Staff_id (Should not be NULL)
- Gender
- Job_title (Should not be NULL)
- Name_birth (Should not be NULL)
- Address
- Phone-number

Attributes of Entity “DEPARTMENT”:

- Department_id (Should not be NULL)
- Head_id (Should not be NULL)
- Department_name
- Doctor_id (Should not be NULL)
- Staff_id (Should not be NULL)

Attributes of Entity “WARD”:

- Ward_no (Should not be NULL)
- Capacity
- Availability
- Nurse_id (Should not be NULL)
- Staff_id (Should not abe NULL)
- Patient_id (Should not be NULL)
- Departmnt_id (Should not be NULL)

Attributes of Entity “APPOINTMENT”:

- Reason (Should not be NULL)
- Current Status
- Appointment_date_time (Should not be NULL)

Attributes of Entity “TEST_REPORT”:

- Test_id (Should not be NULL)
- Test_name
- Result
- Tested_date_time

Attributes of Entity “MEDICINE”:

- Quantity (Should not be NULL)
- Description
- Medicine_id (Should not be NULL)
- Medicine_name
- Price (Should not be NULL)

Attributes of Entity “PRESCRIPTION”:

- Prescribed_date
- No_of_days
- Dosage
- Medicine_id (Should not be NULL)
- Doctor_id (Should not be NULL)
- Prescription_id (Should not be NULL)

Attributes of Entity “BILL”:

- Bill_number (Should not be NULL)
- Bill_date
- Medicine_charge (Should not be NULL)
- Ward_charge
- Test_charge
- Patient_id (Should not be NULL)

Attributes of Entity “RECORD”:

- Record_id (Should not be NULL)
- Date_admitted
- Date_discharged
- Doctor_id (Should not be NULL)
- Medicine_id (Should not be NULL)
- Test_id
- Patient_id (Should not be NULL)

Chapter 2 : Conceptual Design

2.1 ER Diagram

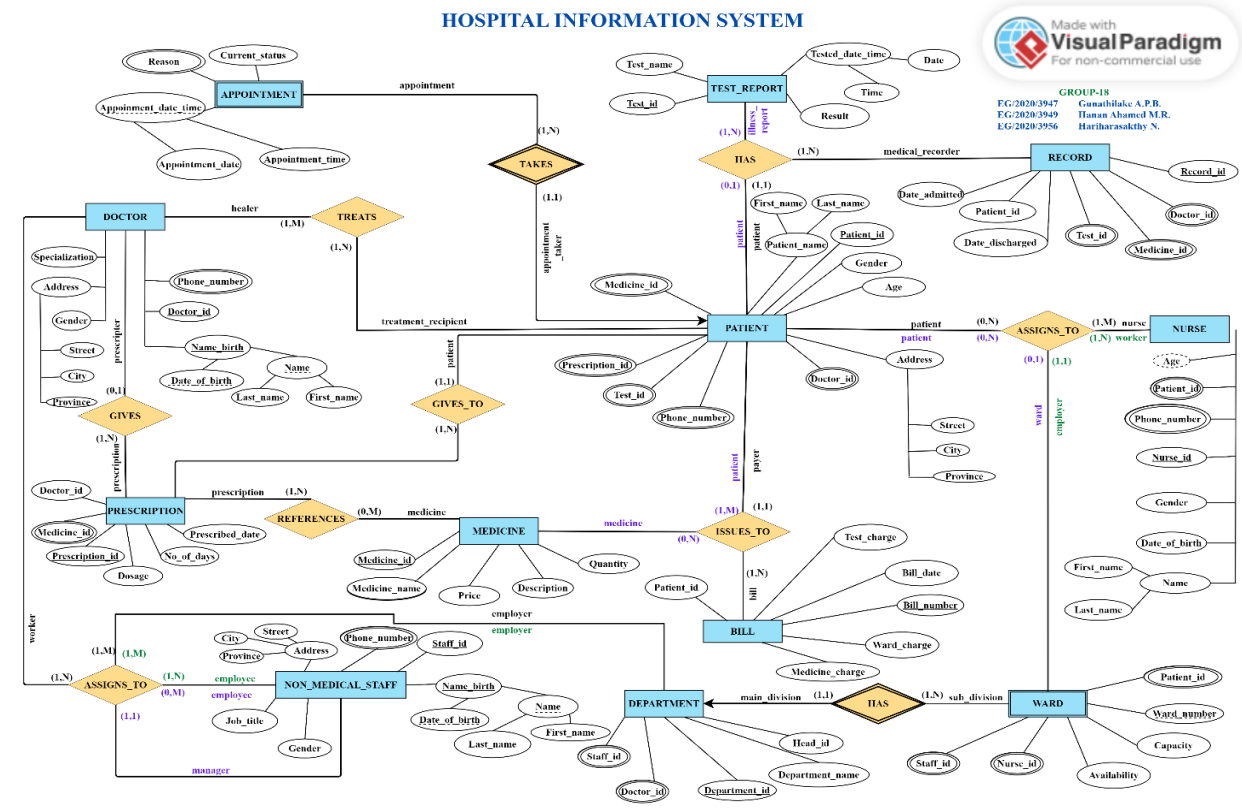


Figure 1:ER Diagram Design

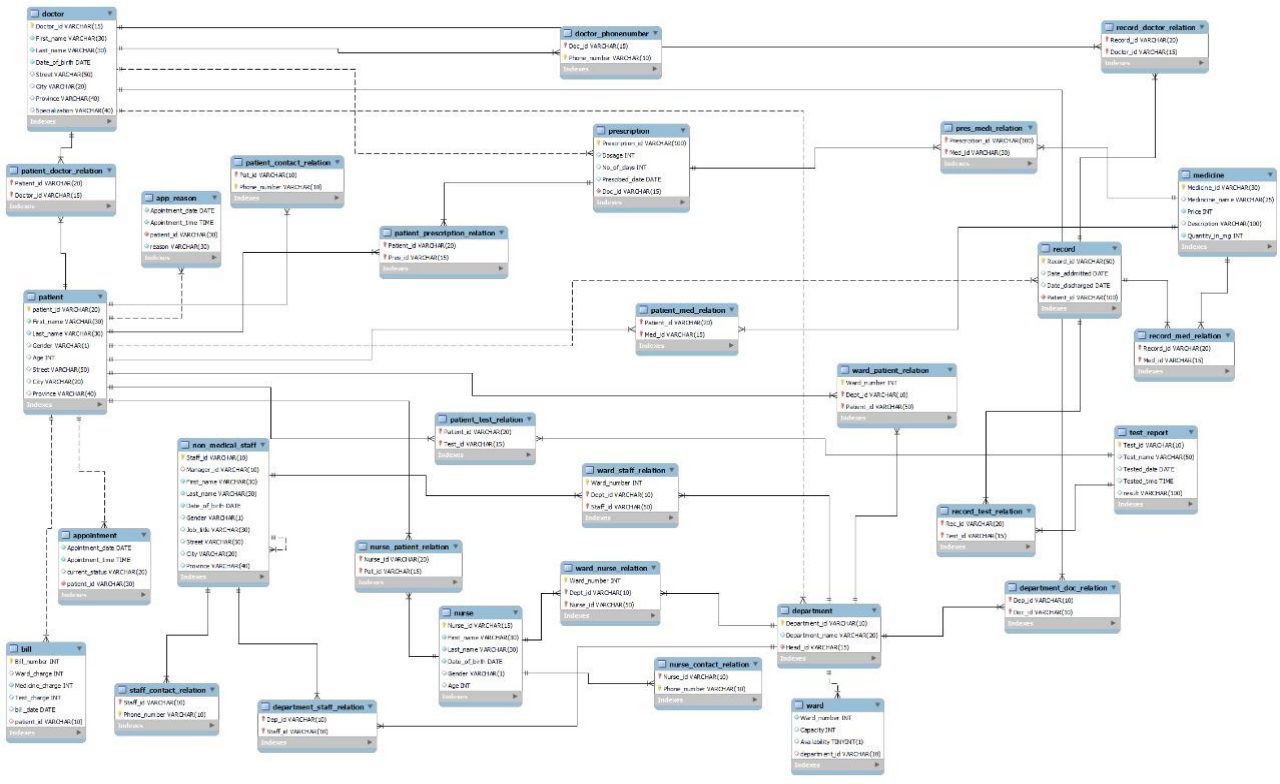
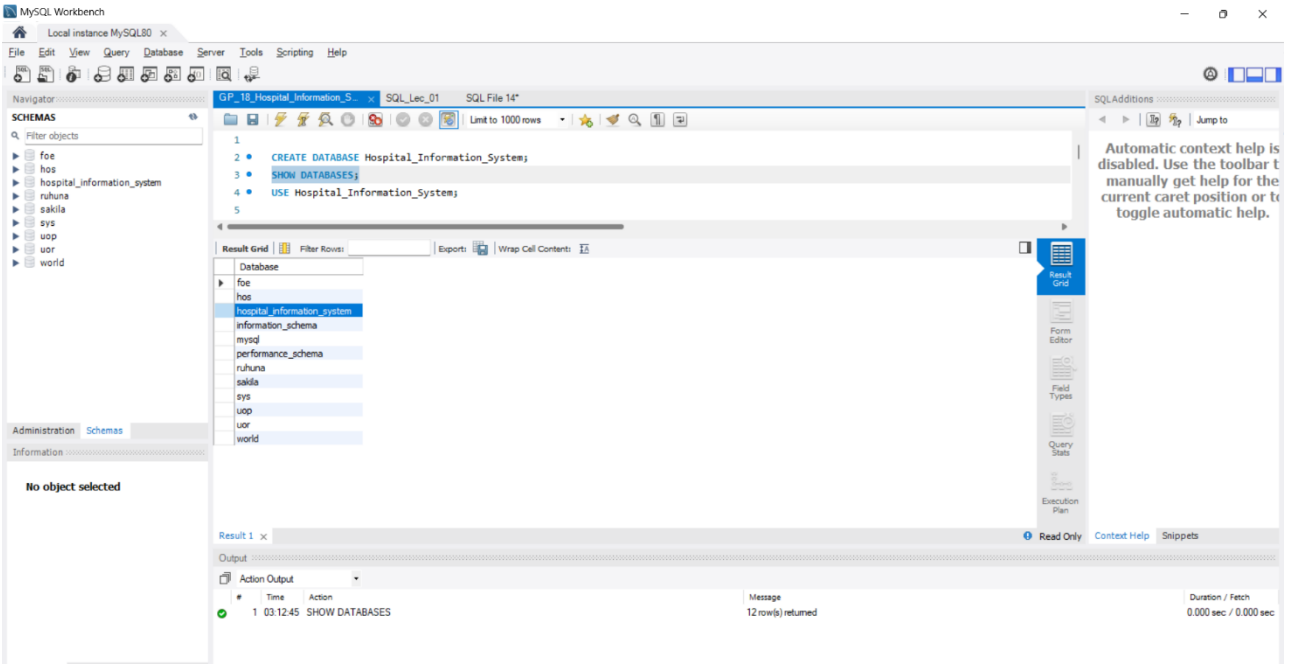


Figure 2:ER Diagram design from MySQL Workbench

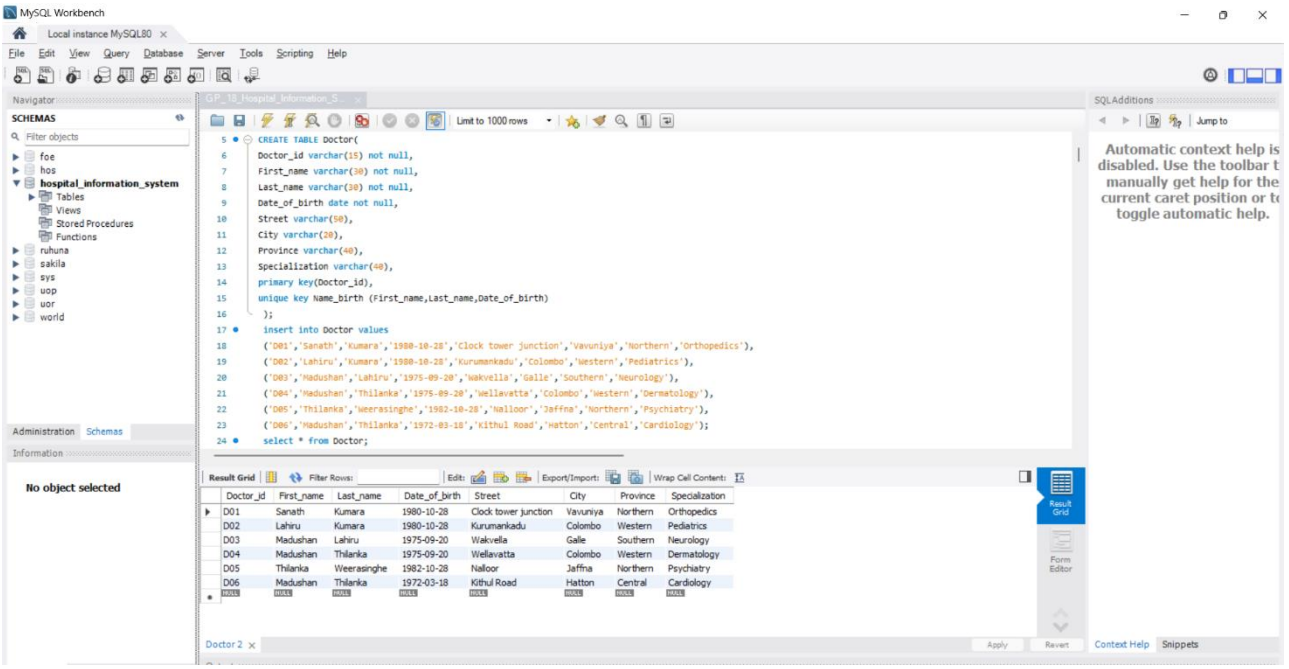
Chapter 3 : Implementation

3.1 Schema Creation

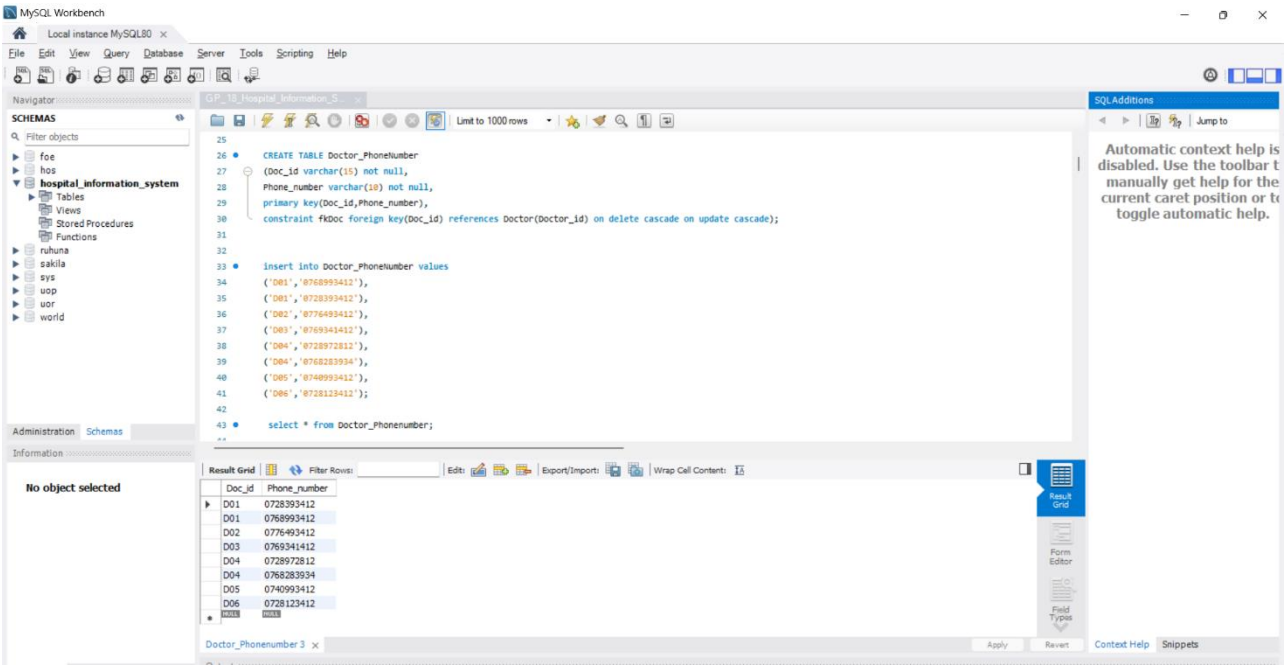


3.2 Creation of Tables & Insertion of data

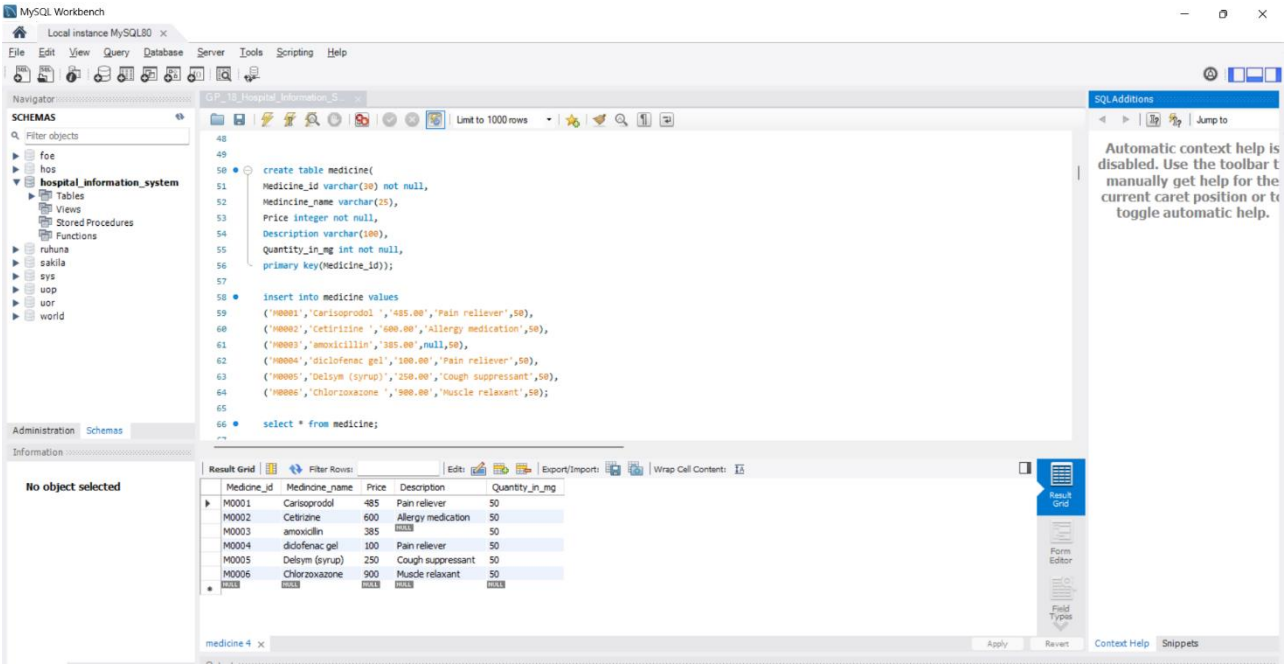
3.2.1 Doctor Table



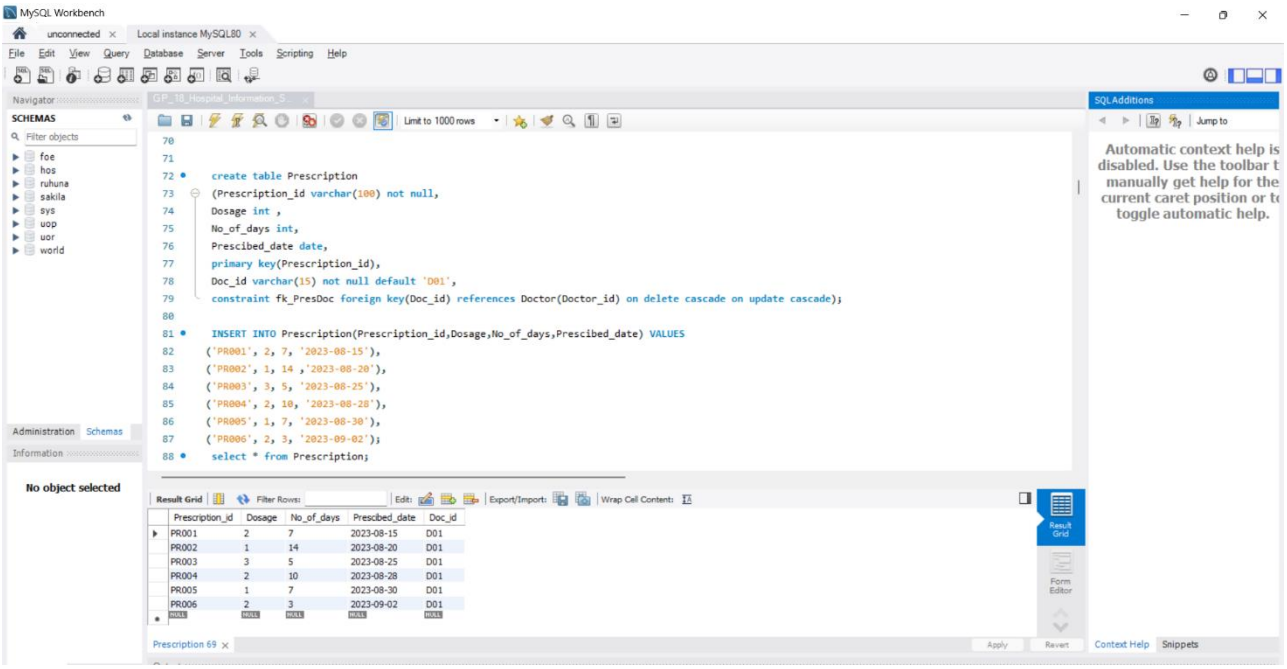
3.2.2 Doctor_PhoneNumber Table



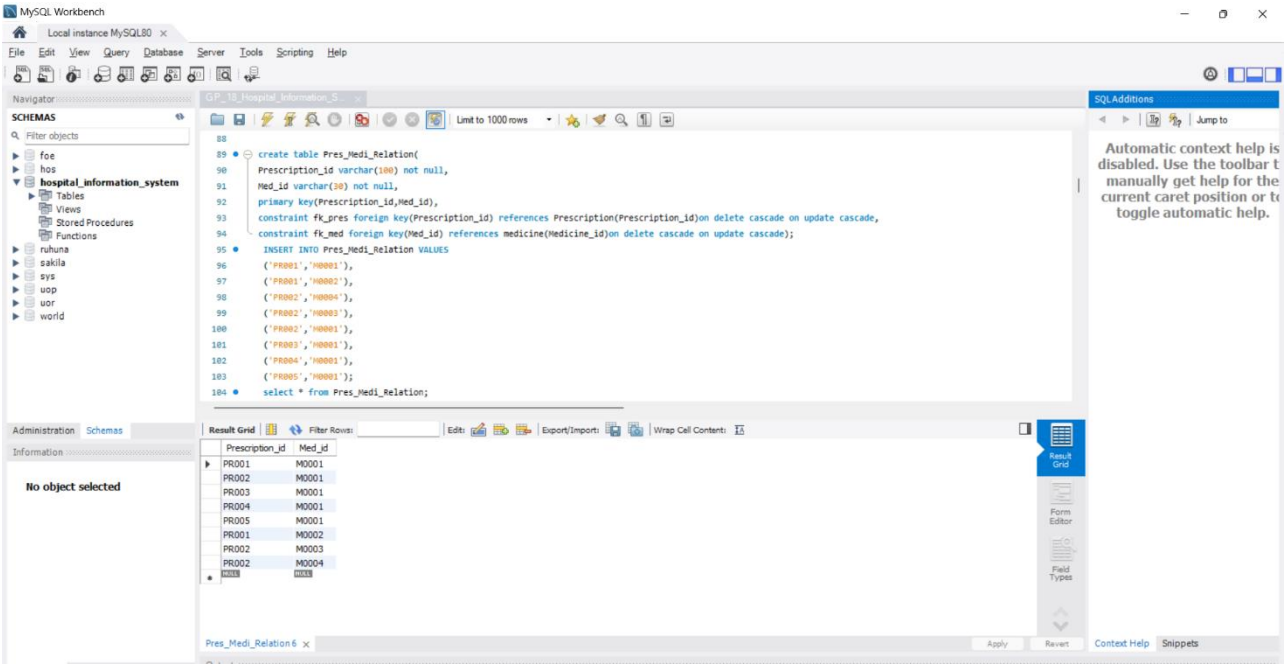
3.2.3 Medicine Table



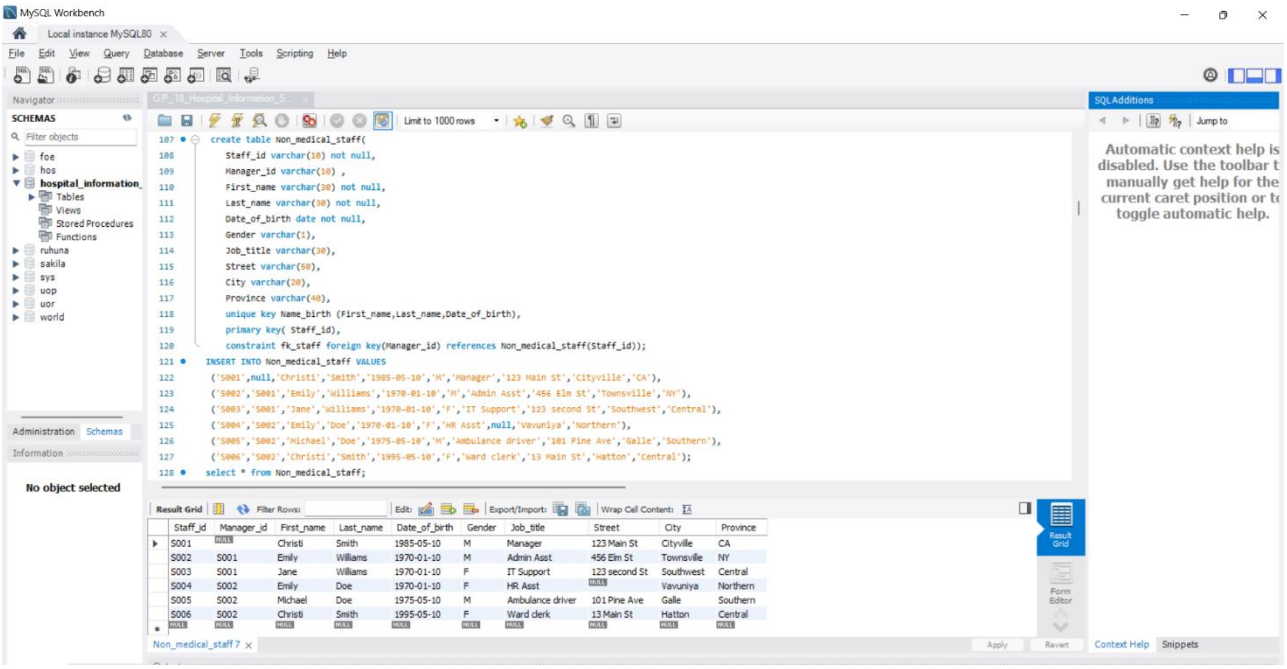
3.2.4 Prescription Table



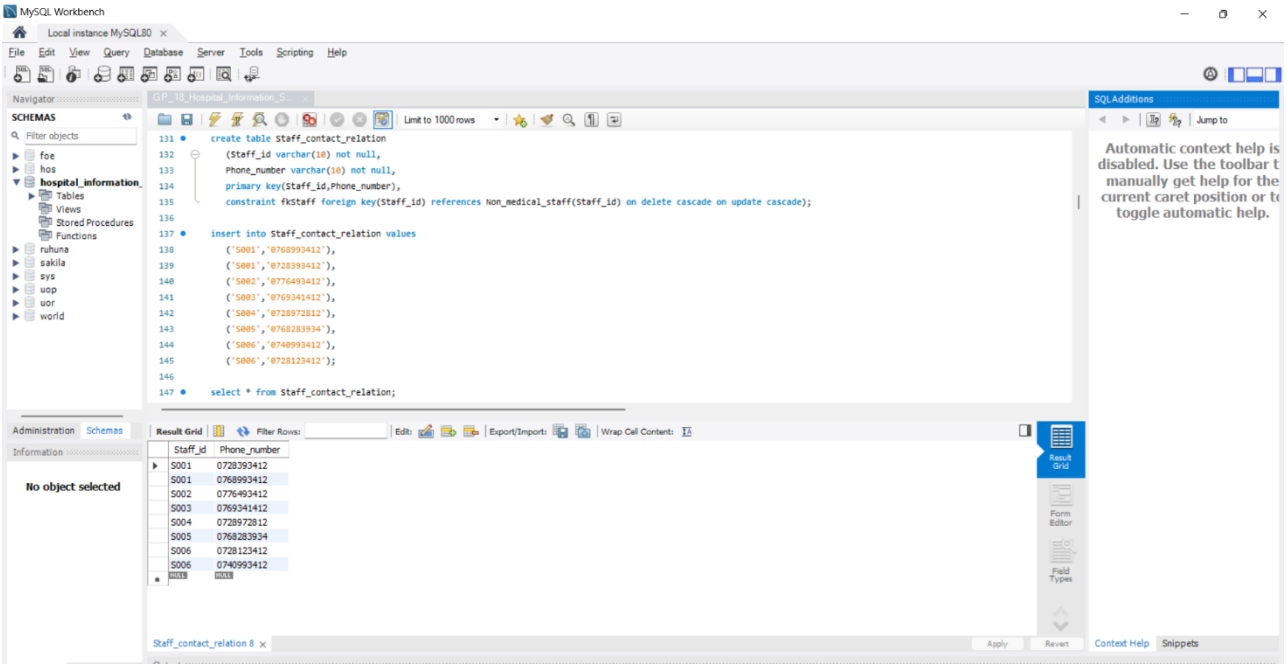
3.2.5 Pres_Medi_Relation Table



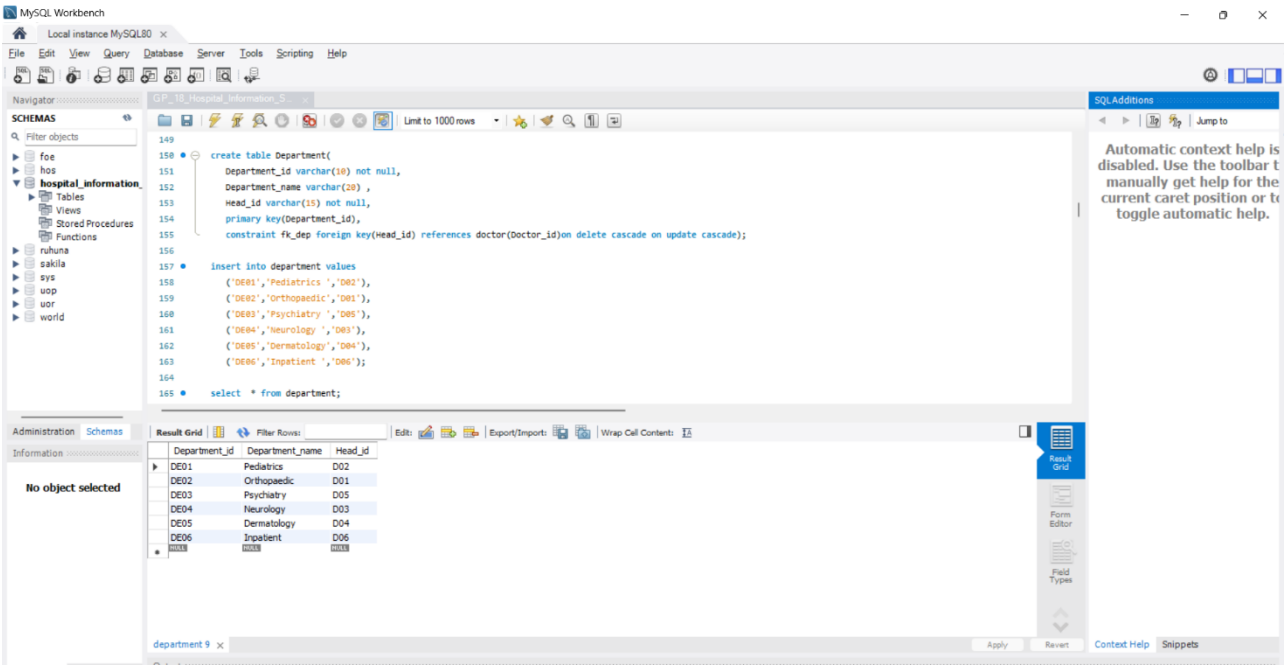
3.2.6 Non_medical_staff Table



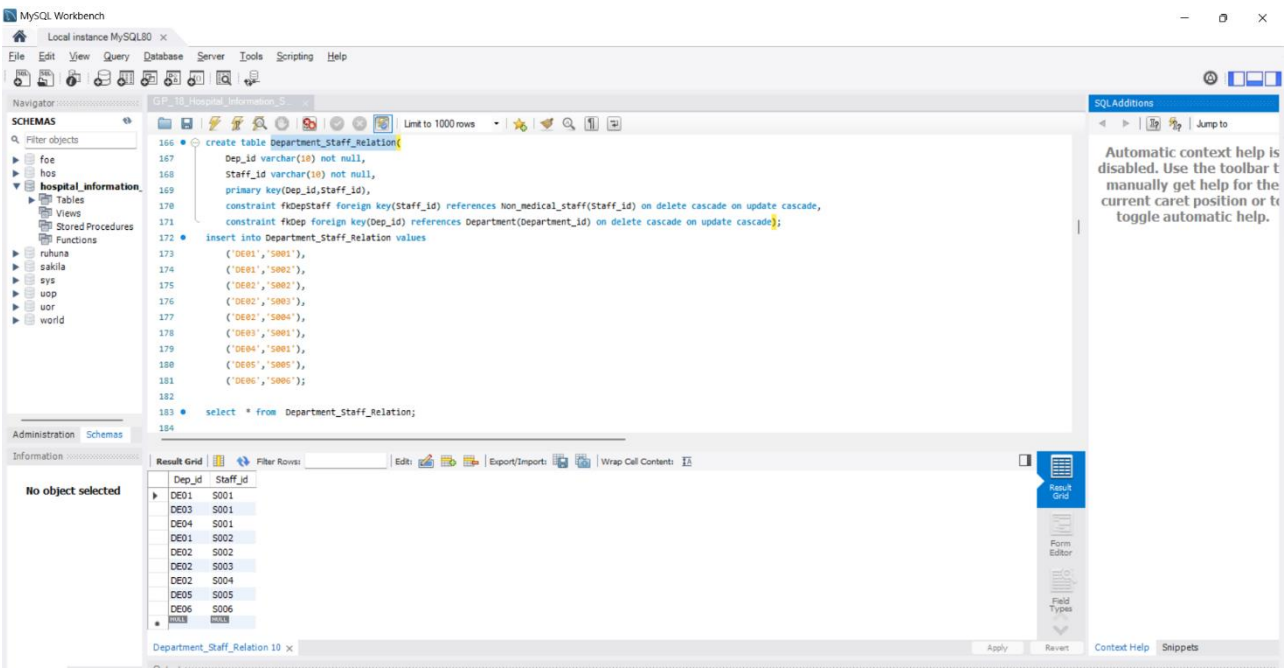
3.2.7 Staff_contact_relation Table



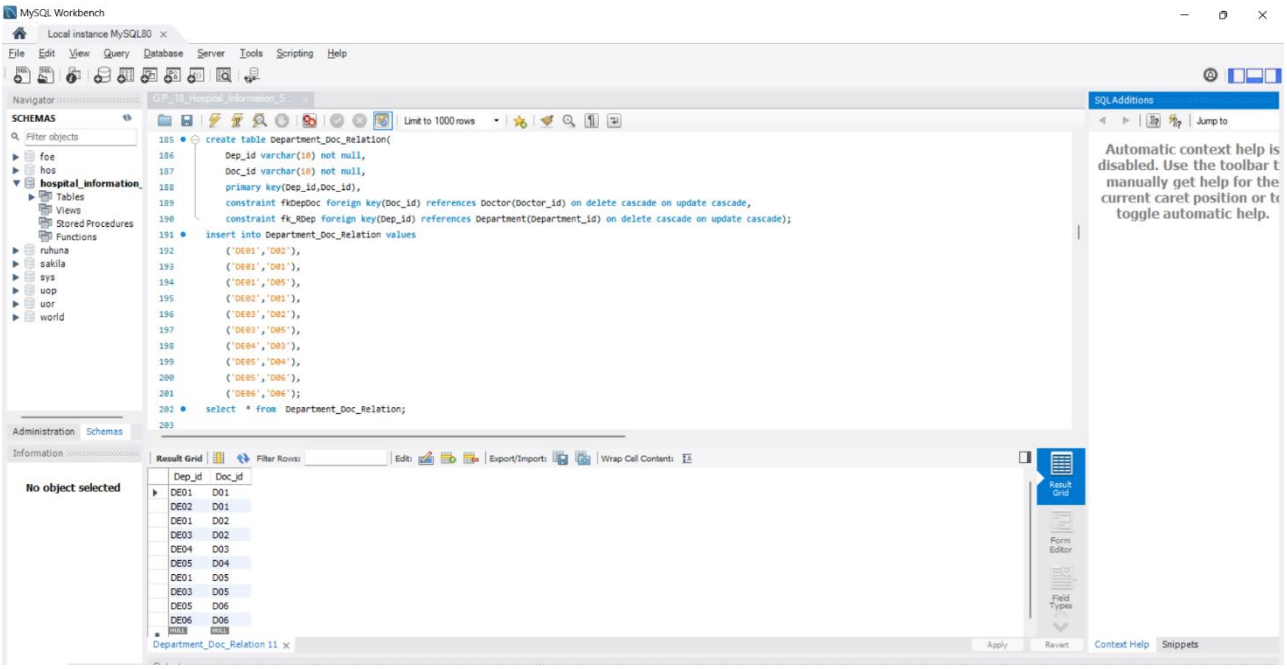
3.2.8 Department Table



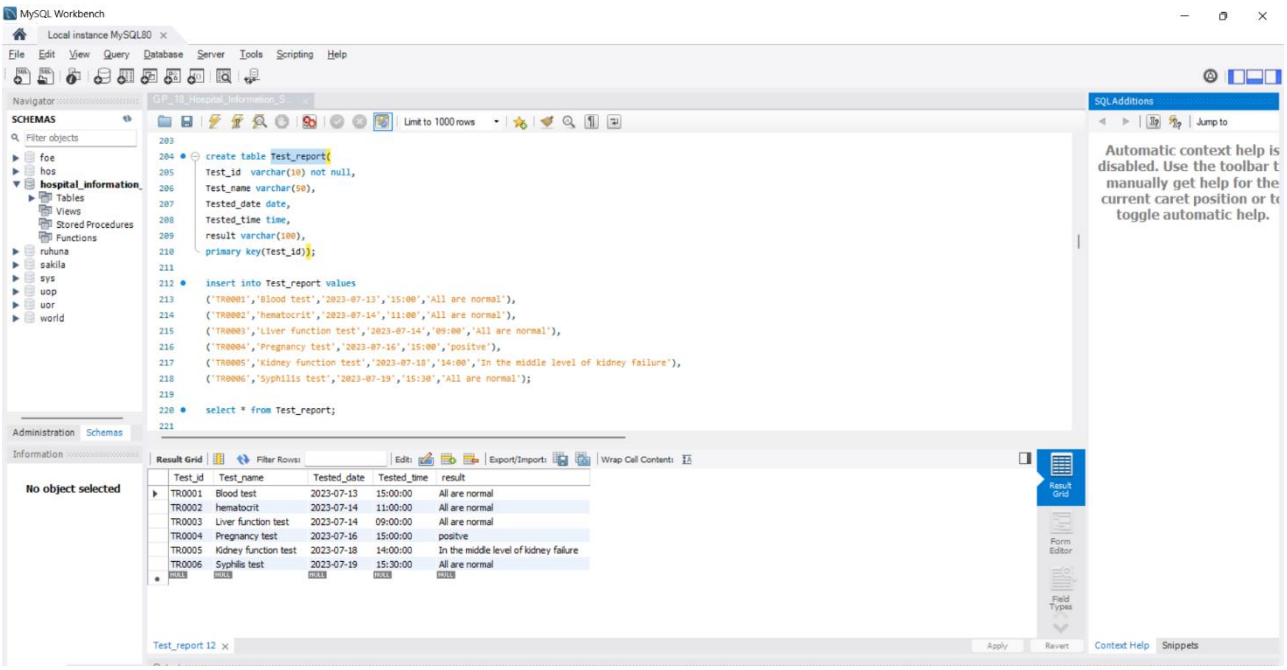
3.2.9 Department_Staff_Relation Table



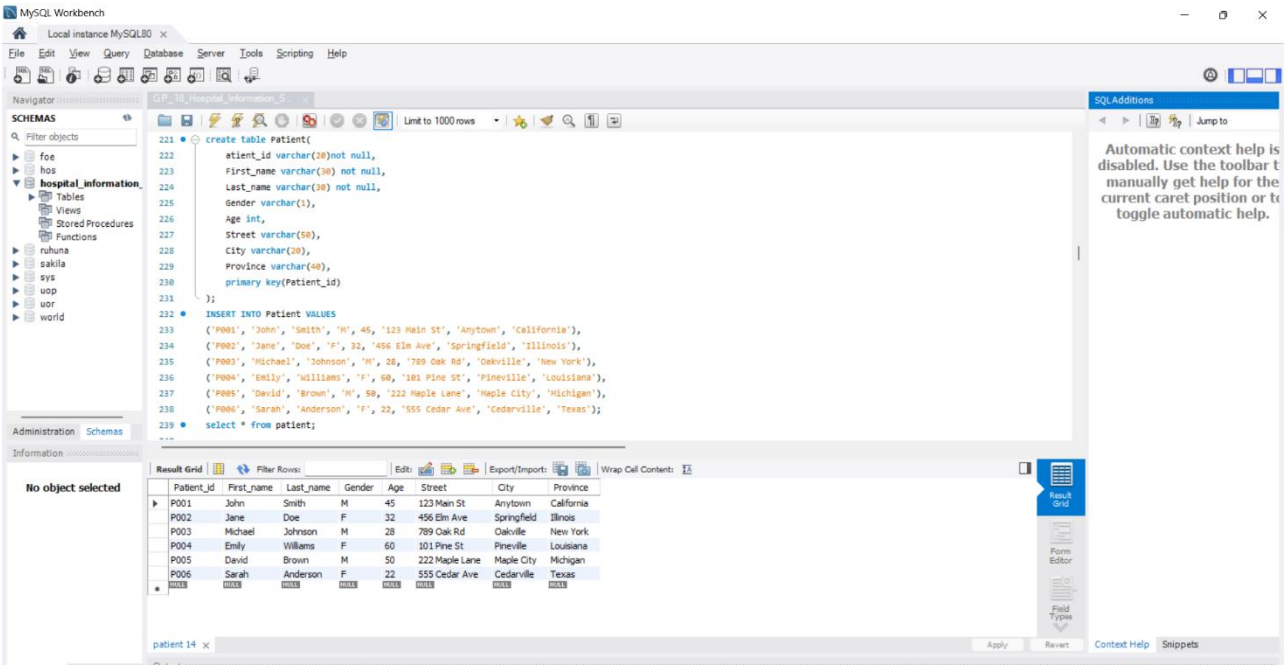
3.2.10 Department_Doc_Relation Table



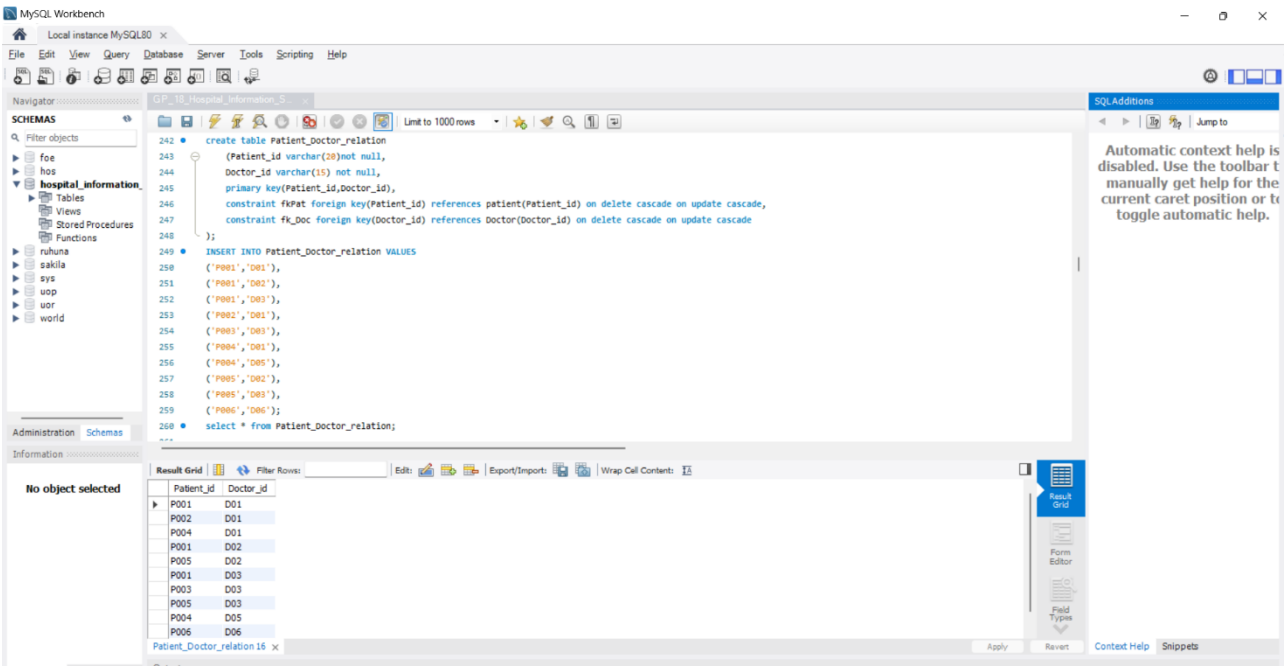
3.2.11 Test_report Table



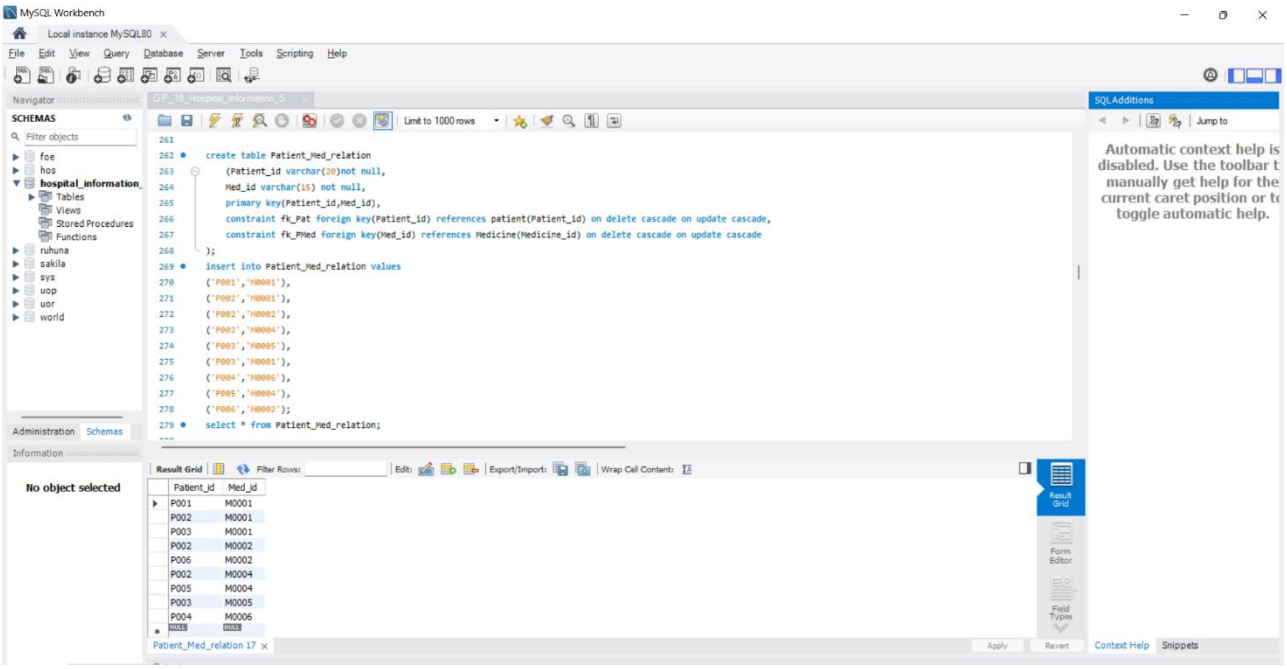
3.2.12 Patient Table



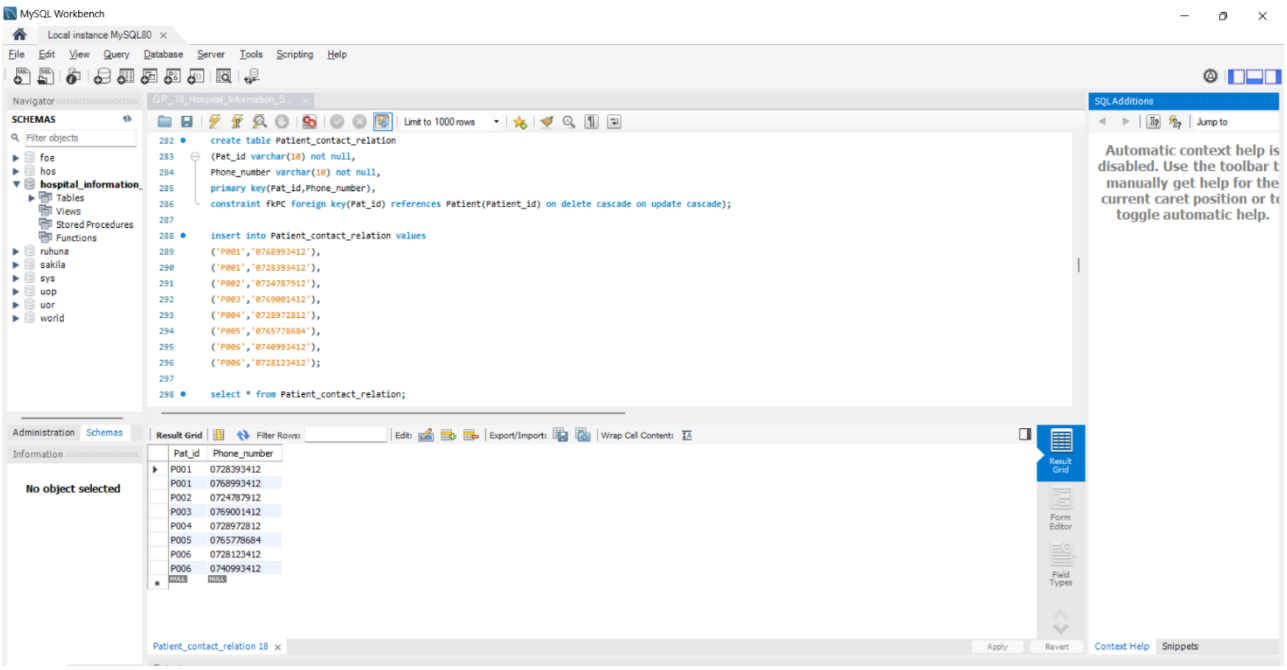
3.2.13 Patient_Doctor_relation Table



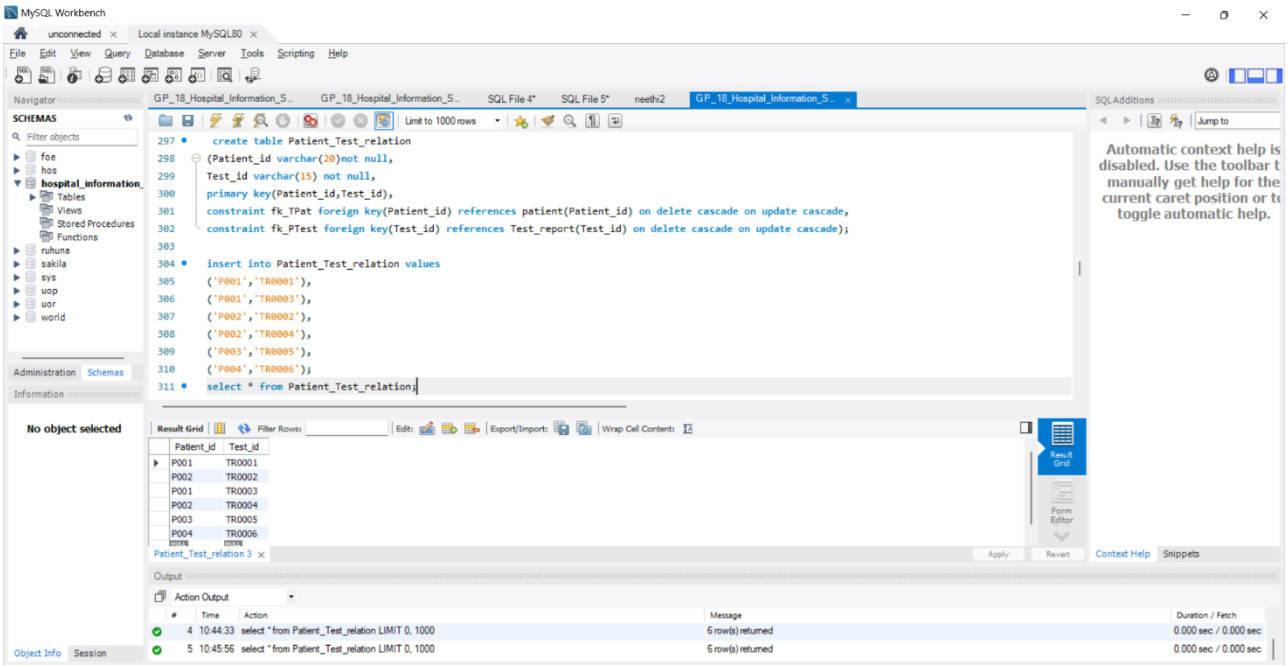
3.2.14 Patient_Med_relation Table



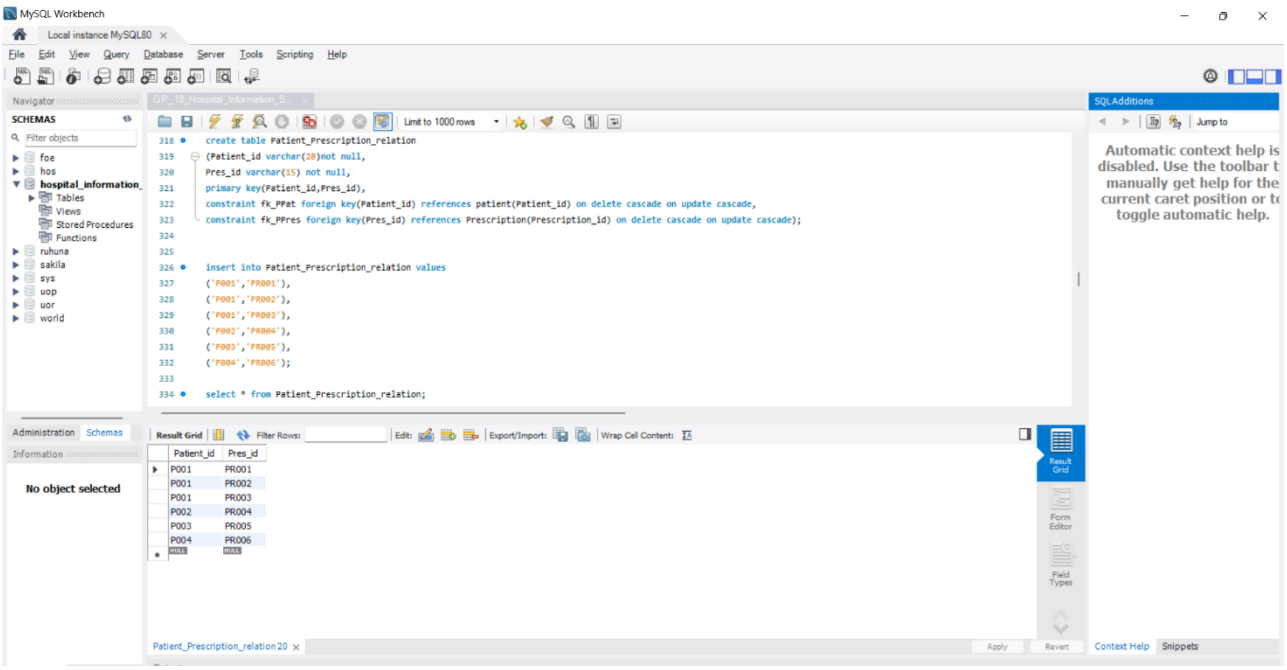
3.2.15 Patient_contact_relation Table



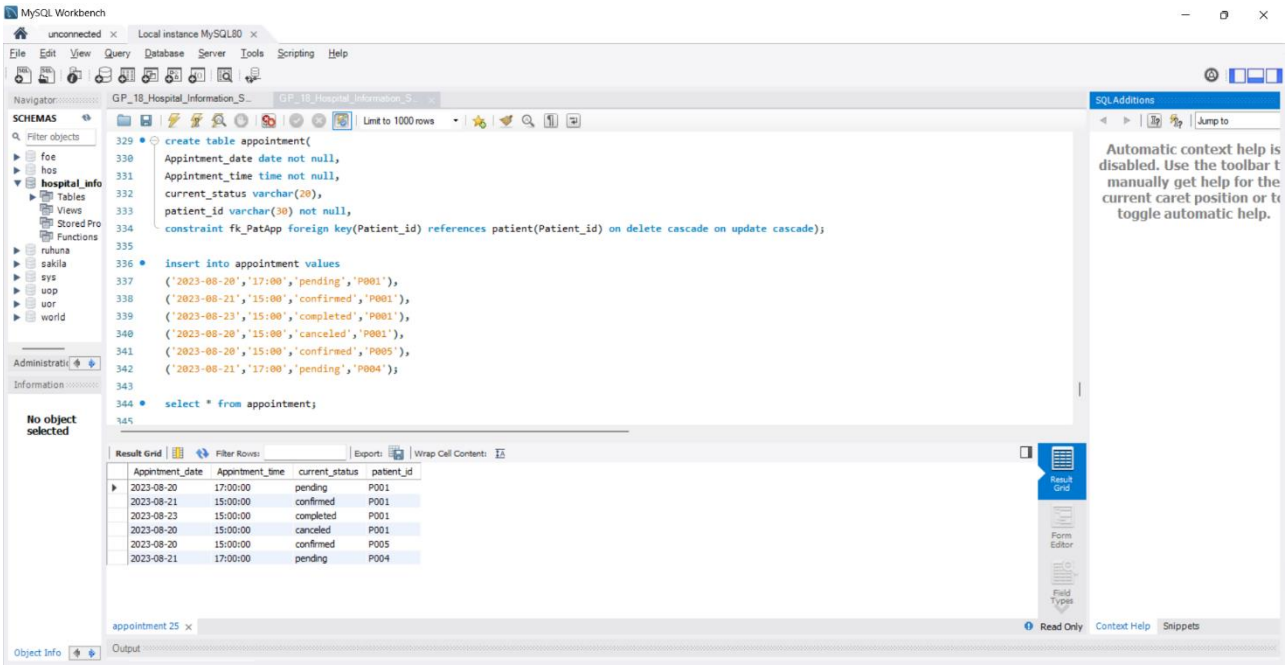
3.2.16 Patient_Test_relation Table



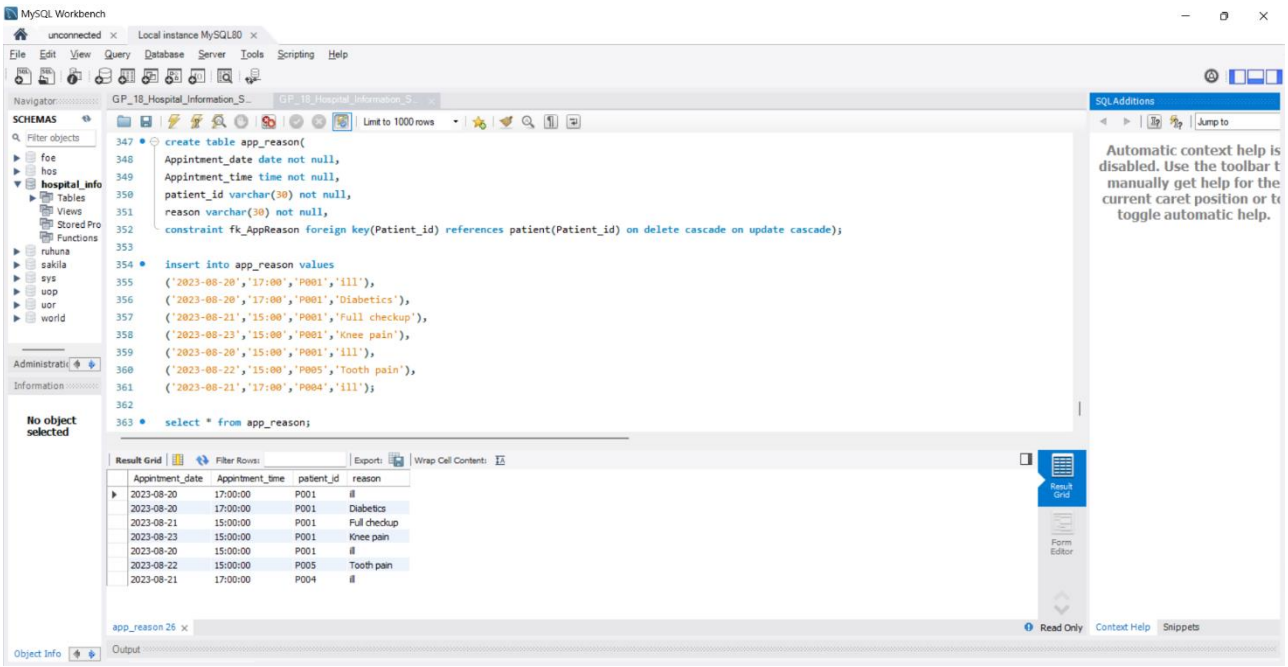
3.2.17 Patient_Prescription_relation Table



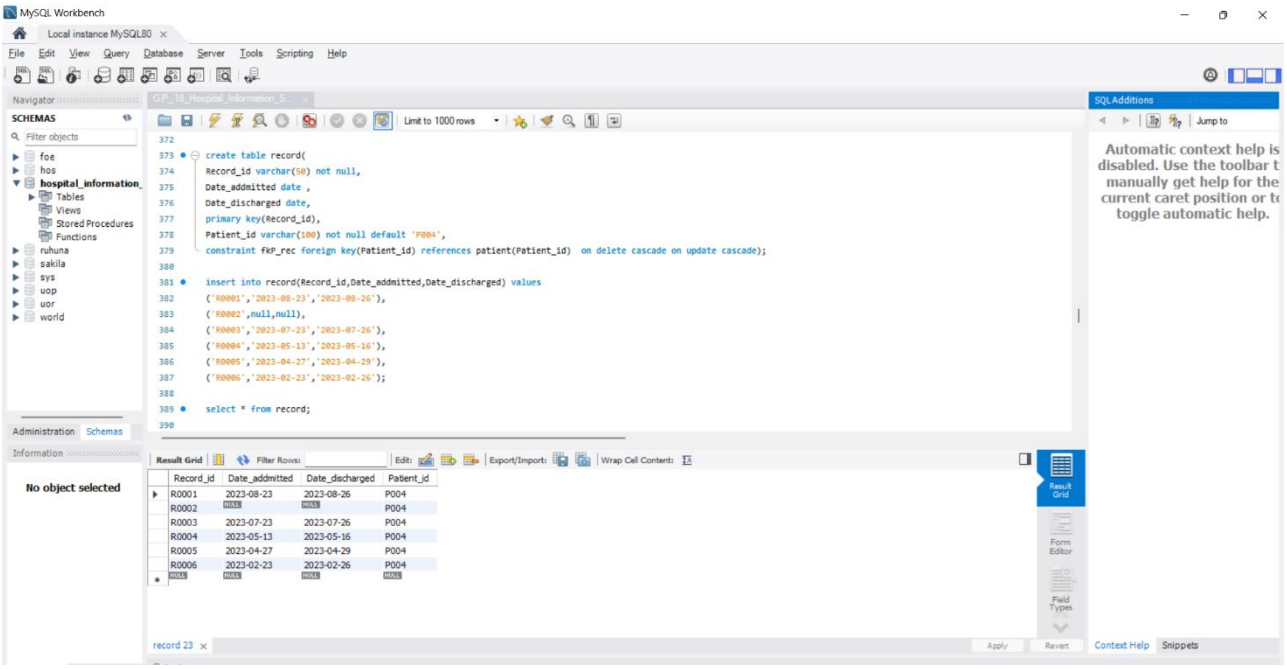
3.2.18 Appointment Table



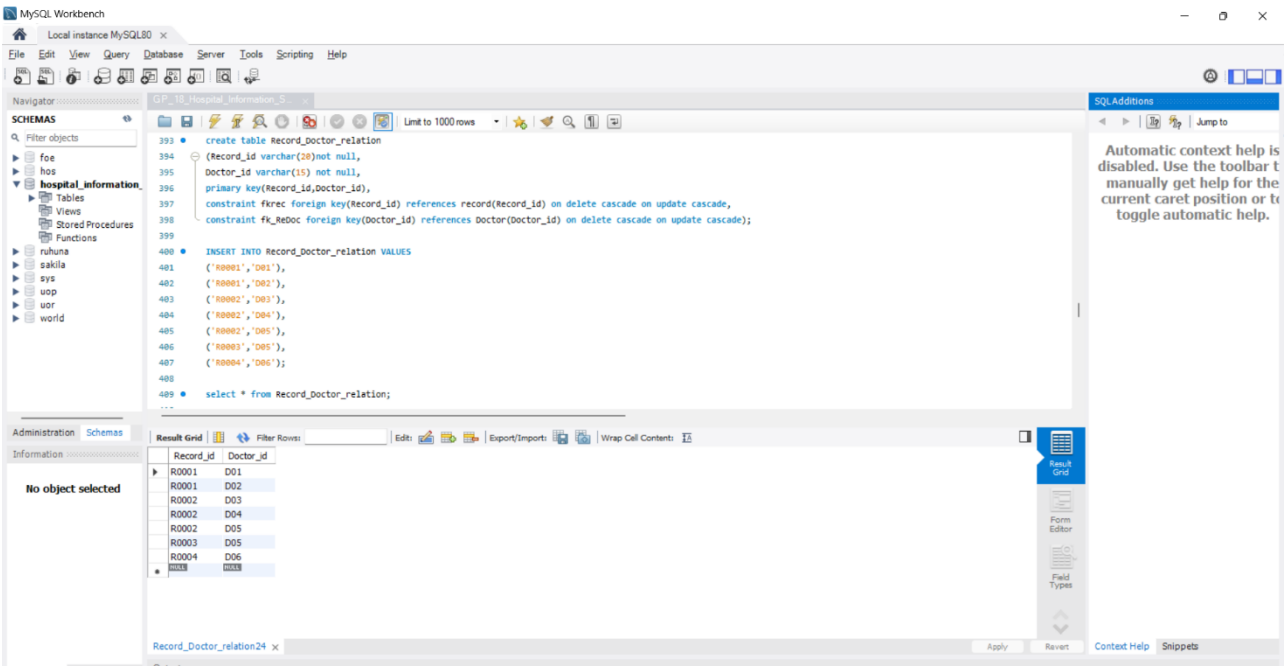
3.2.19 App_reason Table



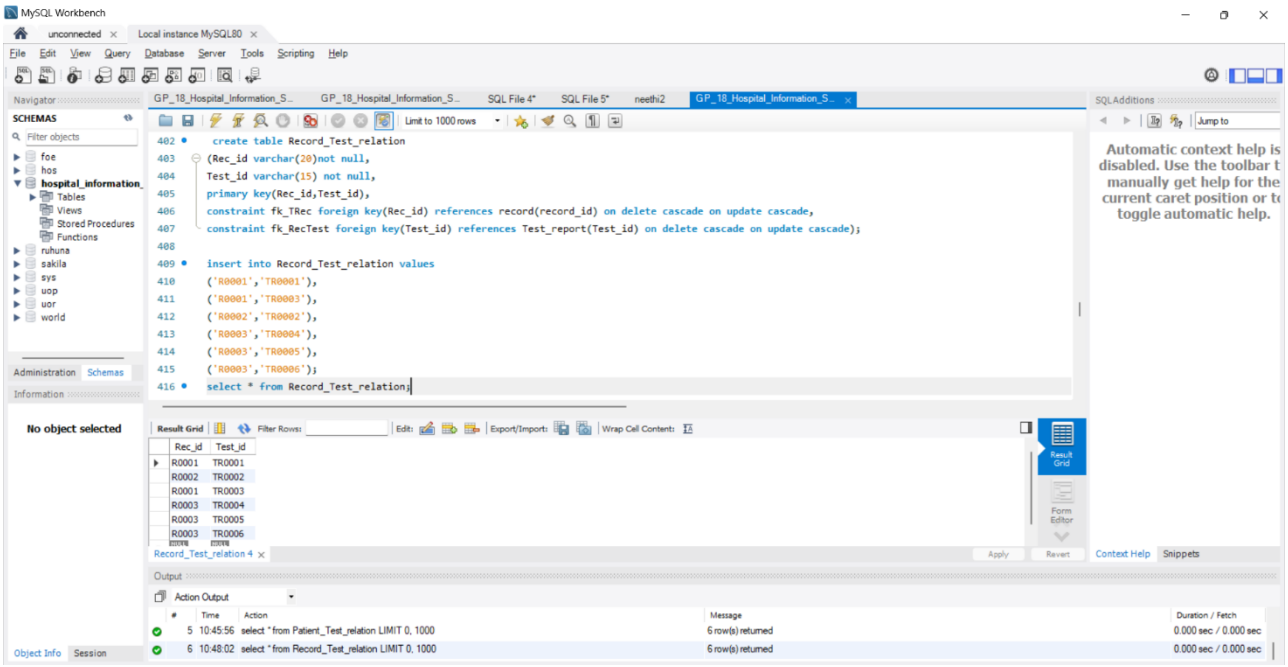
3.2.20 Record Table



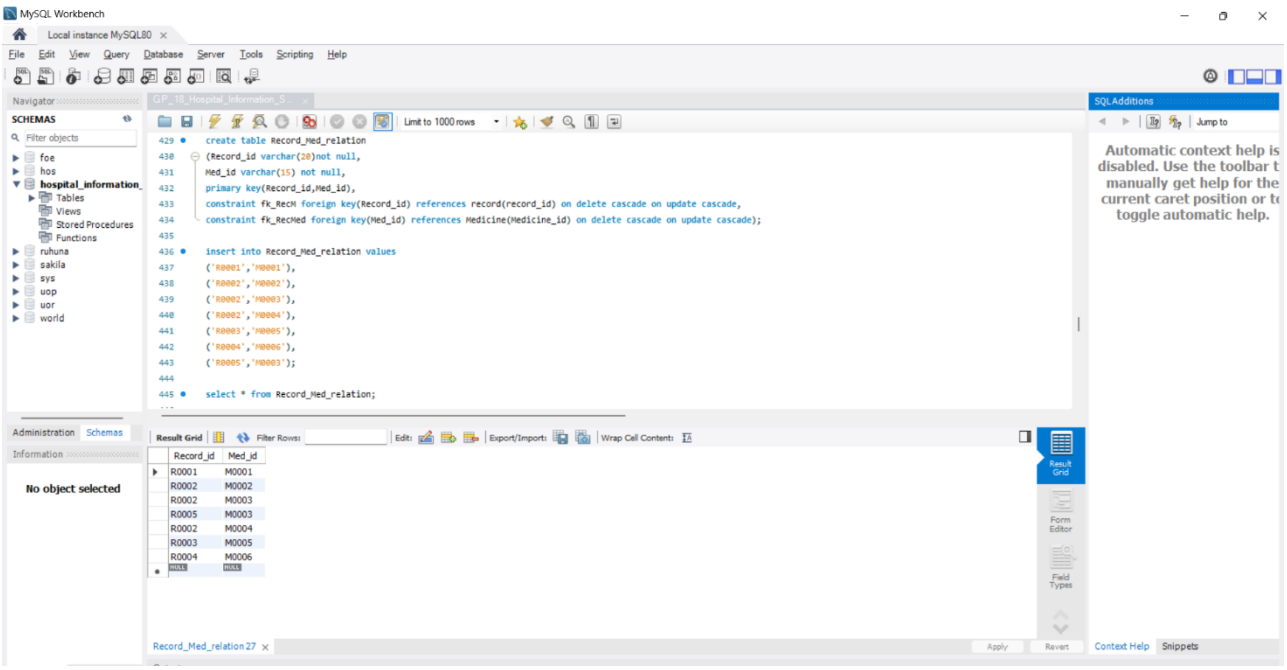
3.2.21 Record_Doctor_relation Table



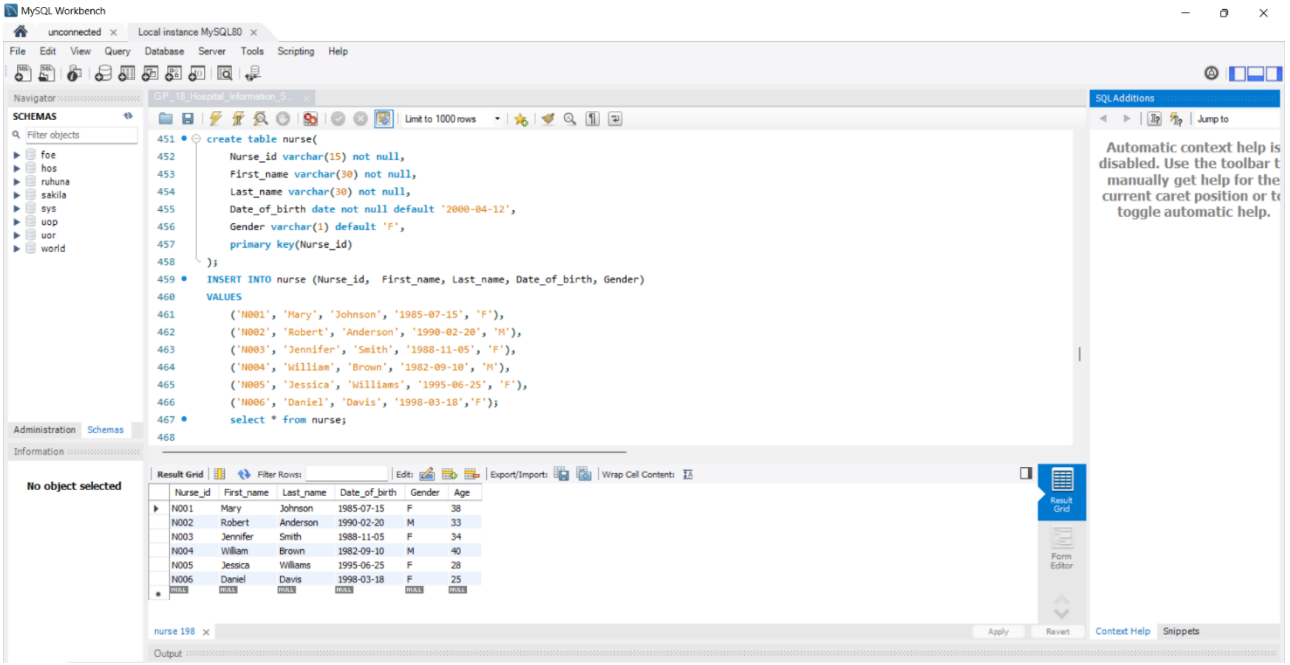
3.2.22 Record_Test_relation Table



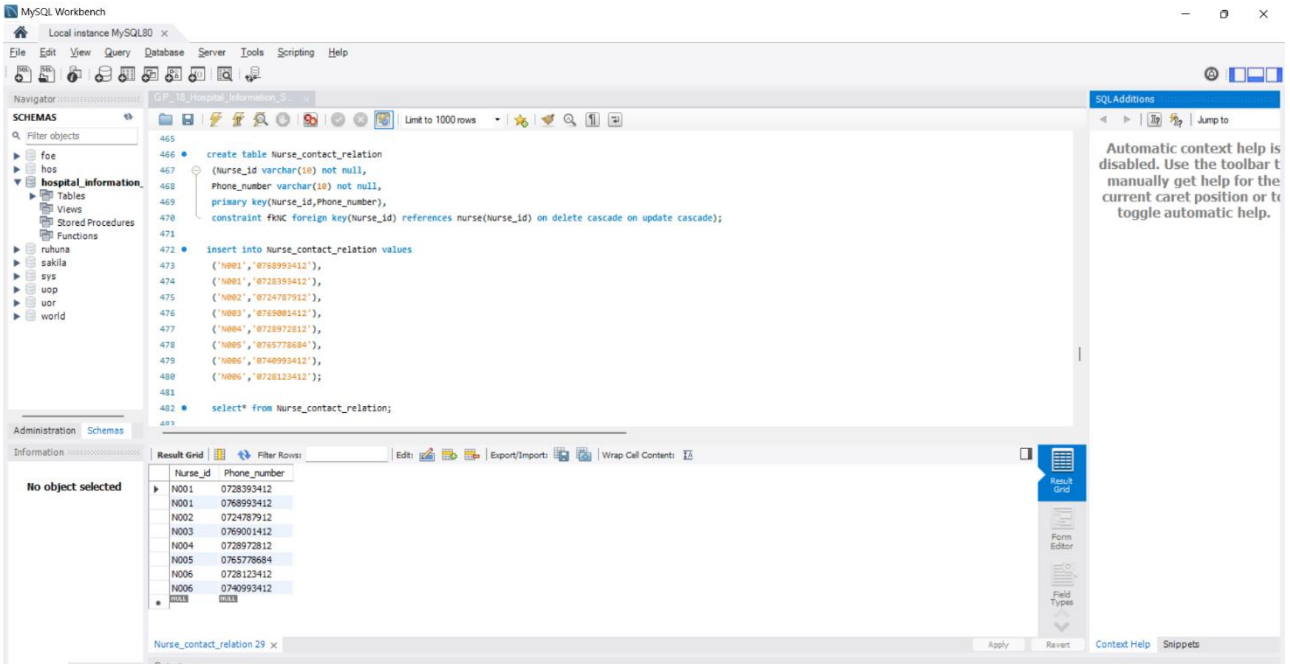
3.2.23 Record_Med_relation Table



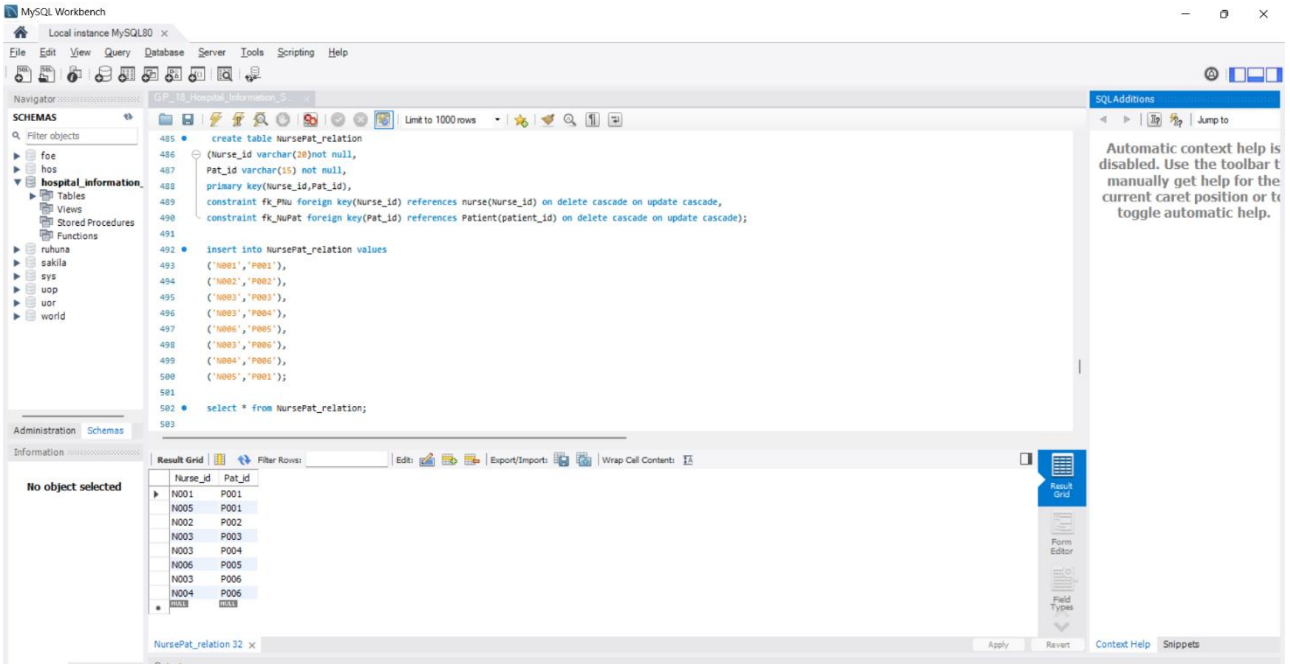
3.2.24 Nurse Table



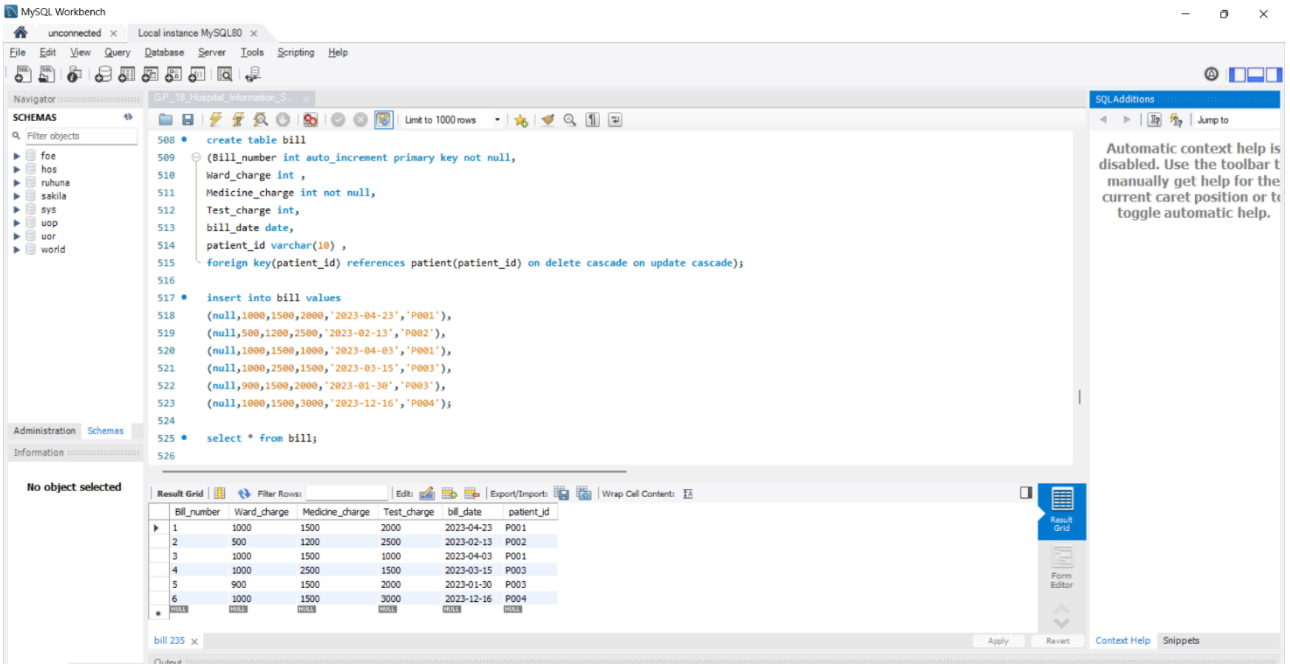
3.2.25 Nurse_contact_relation Table



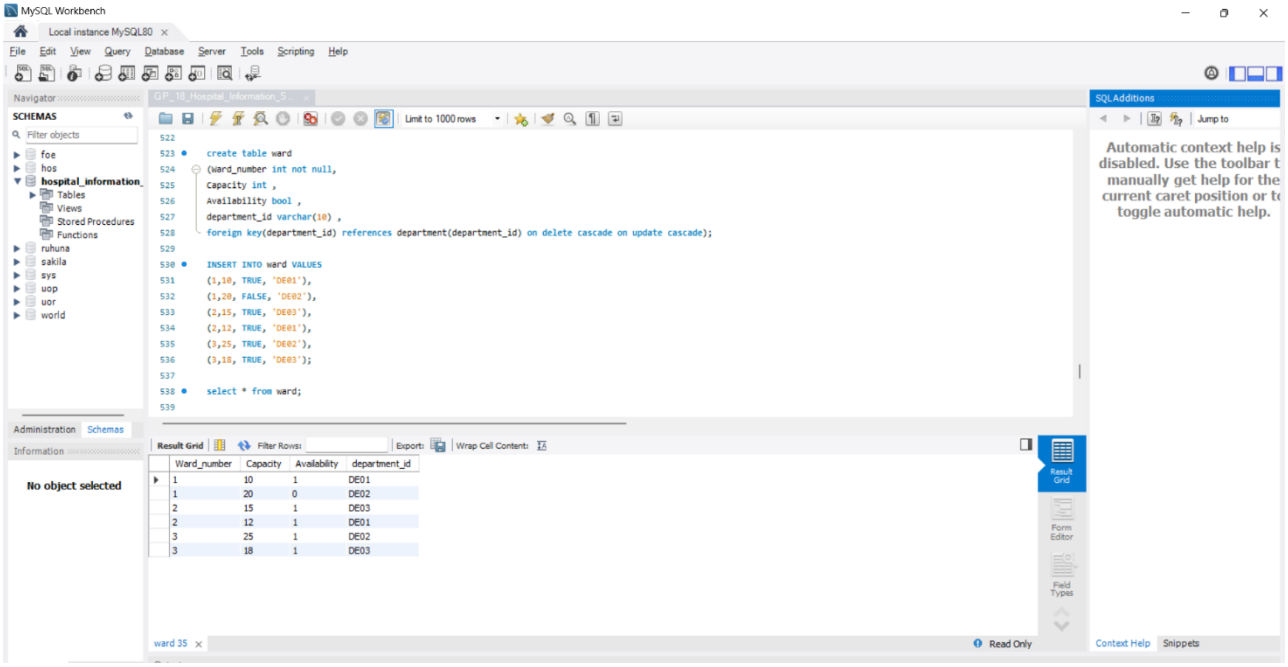
3.2.26 NursePat_relation Table



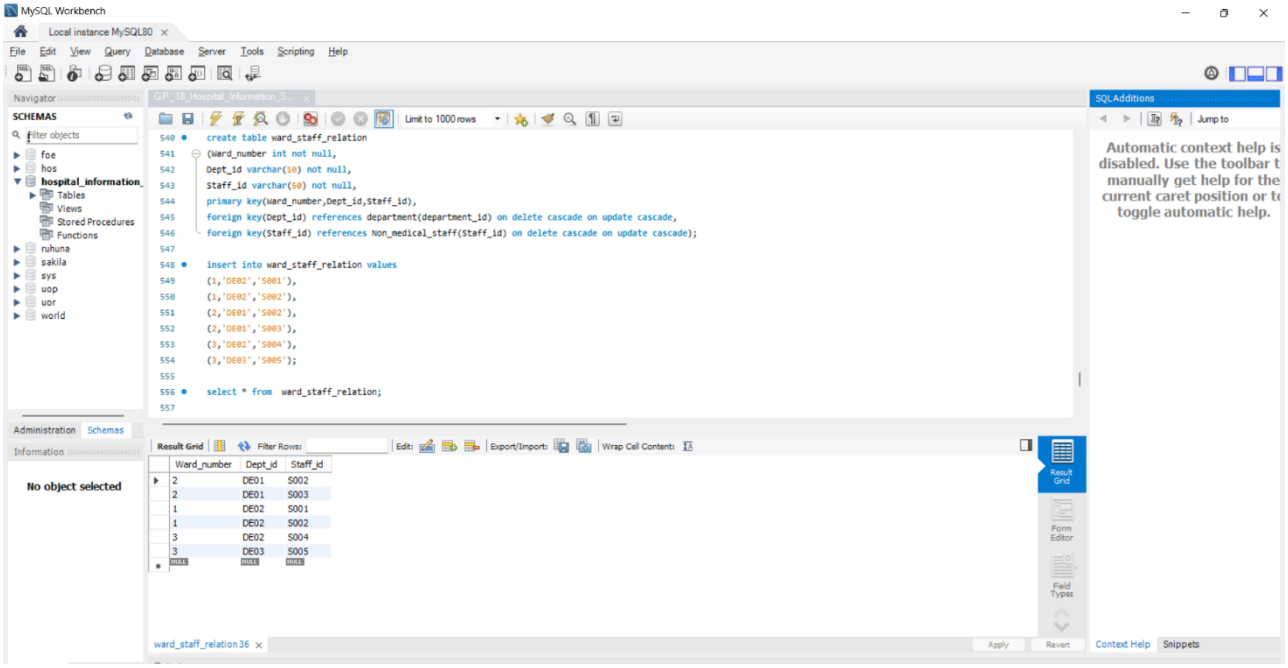
3.2.27 Bill Table



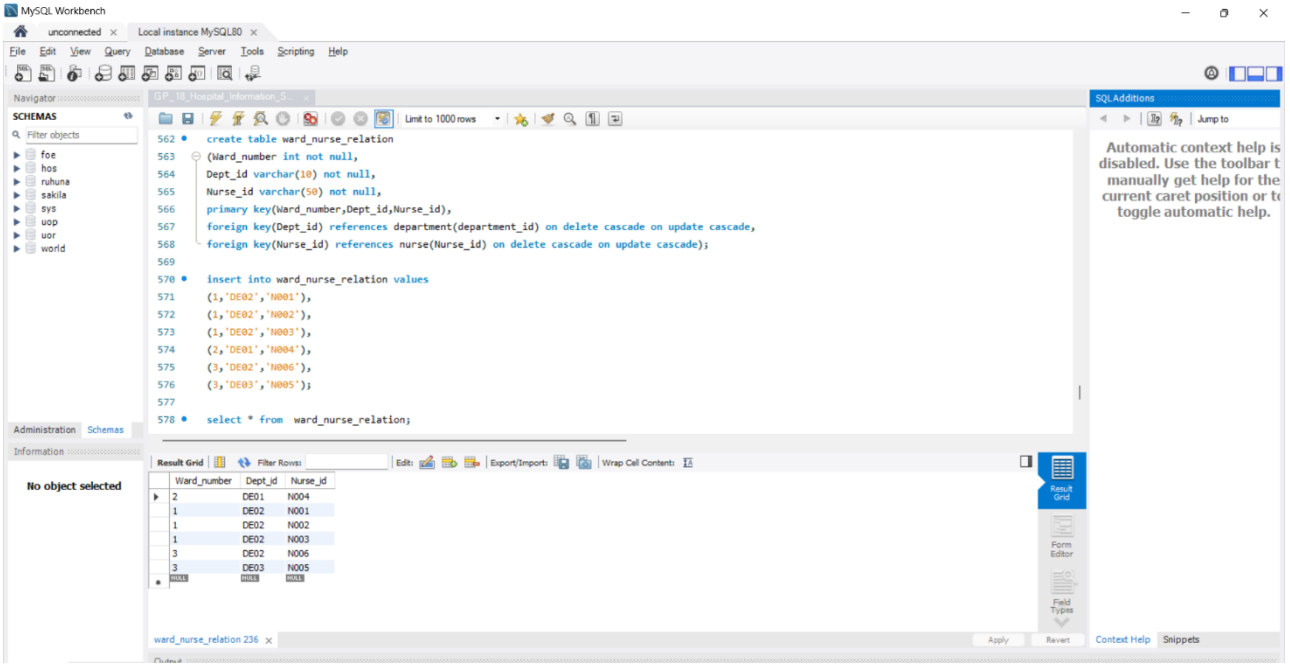
3.2.28 Ward Table



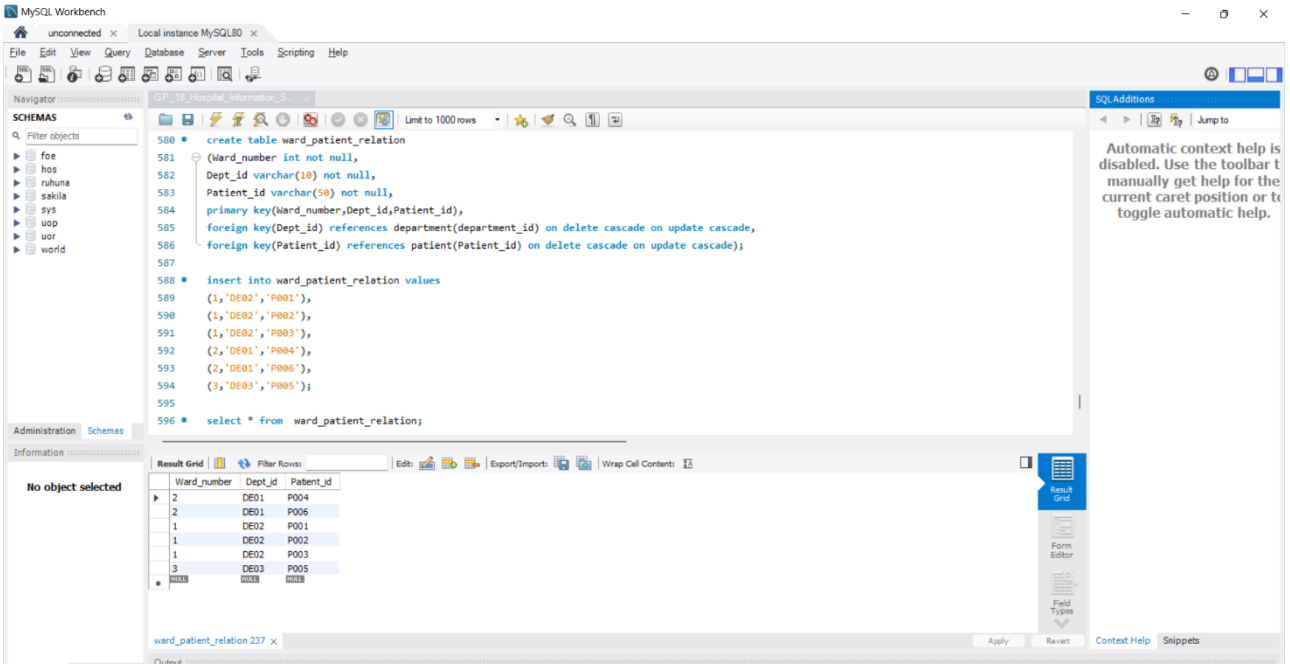
3.2.29 Ward_staff_relation Table



3.2.30 Ward_nurse_relation Table

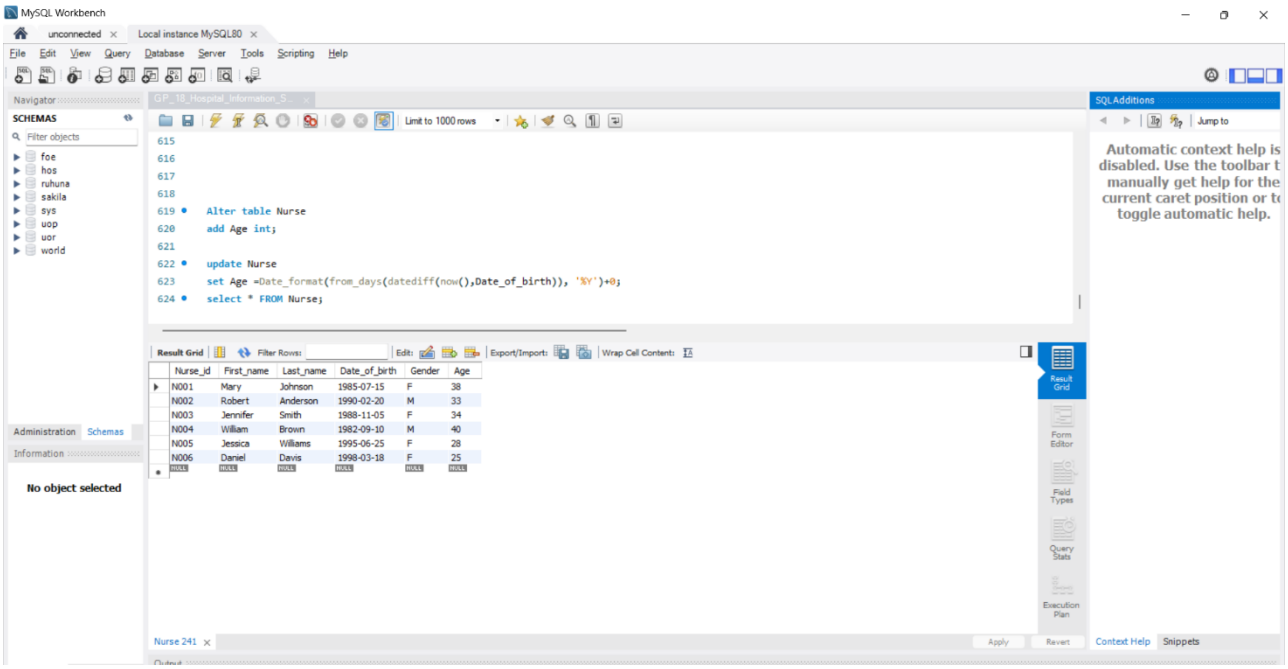


3.2.31 Ward_patient_relation Table



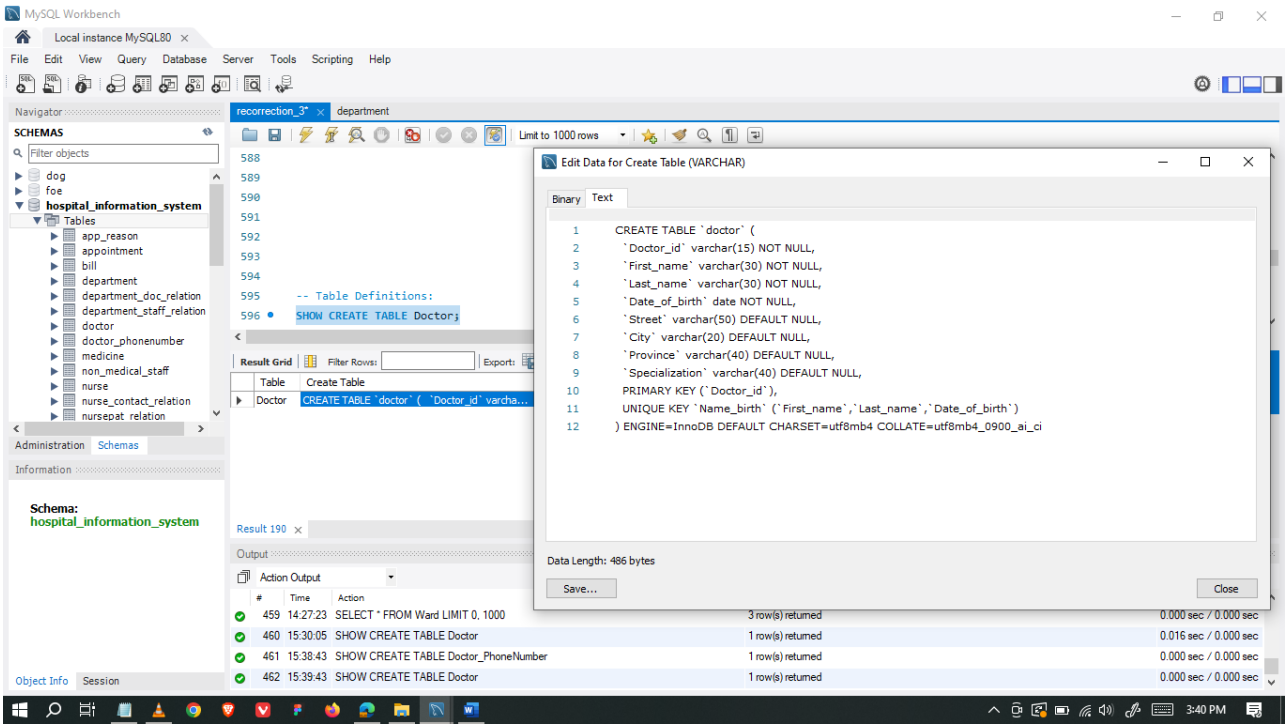
3.3 Alteration & deriving age from date of birth

3.3.1 Nurse Table

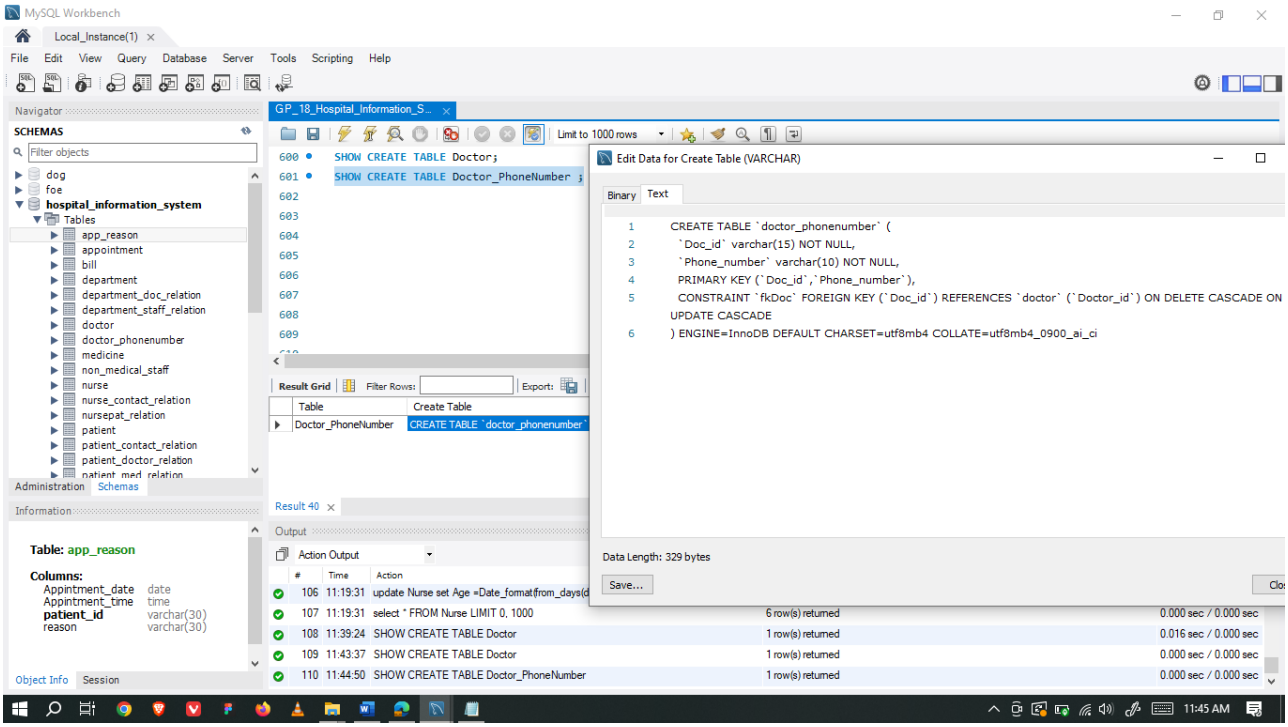


3.4 Table Definitions

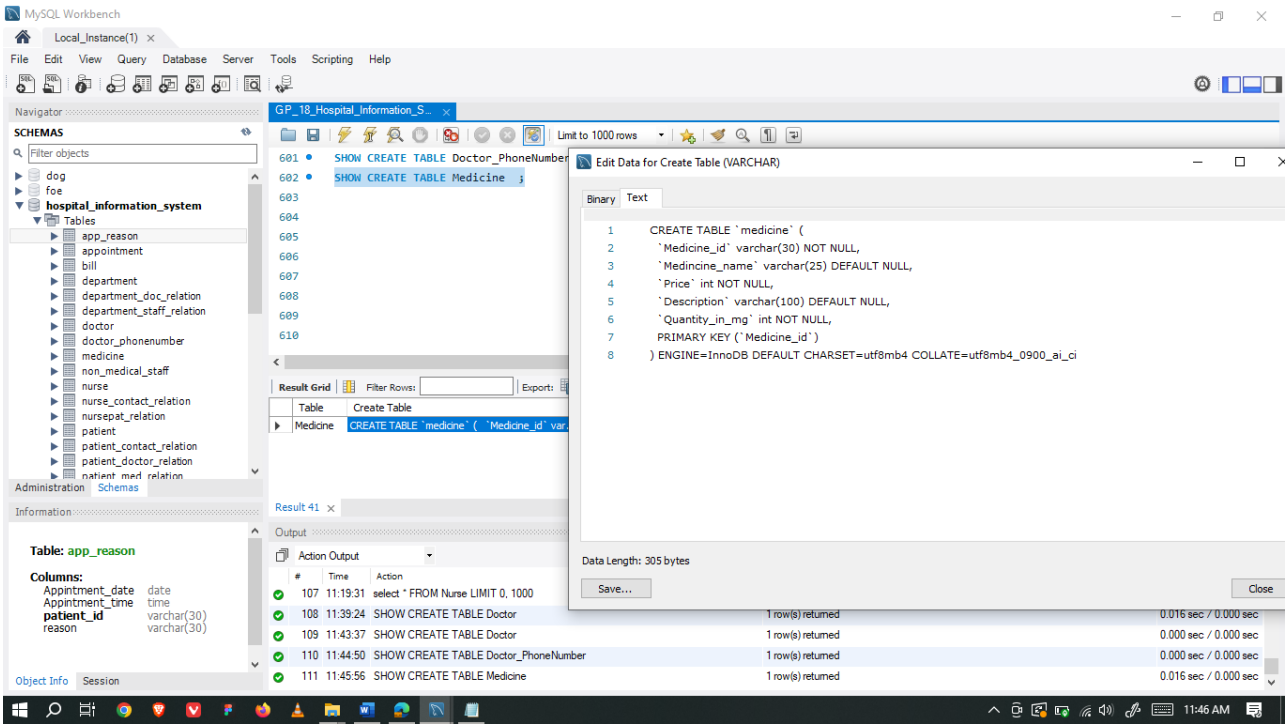
3.4.1 Doctor Table



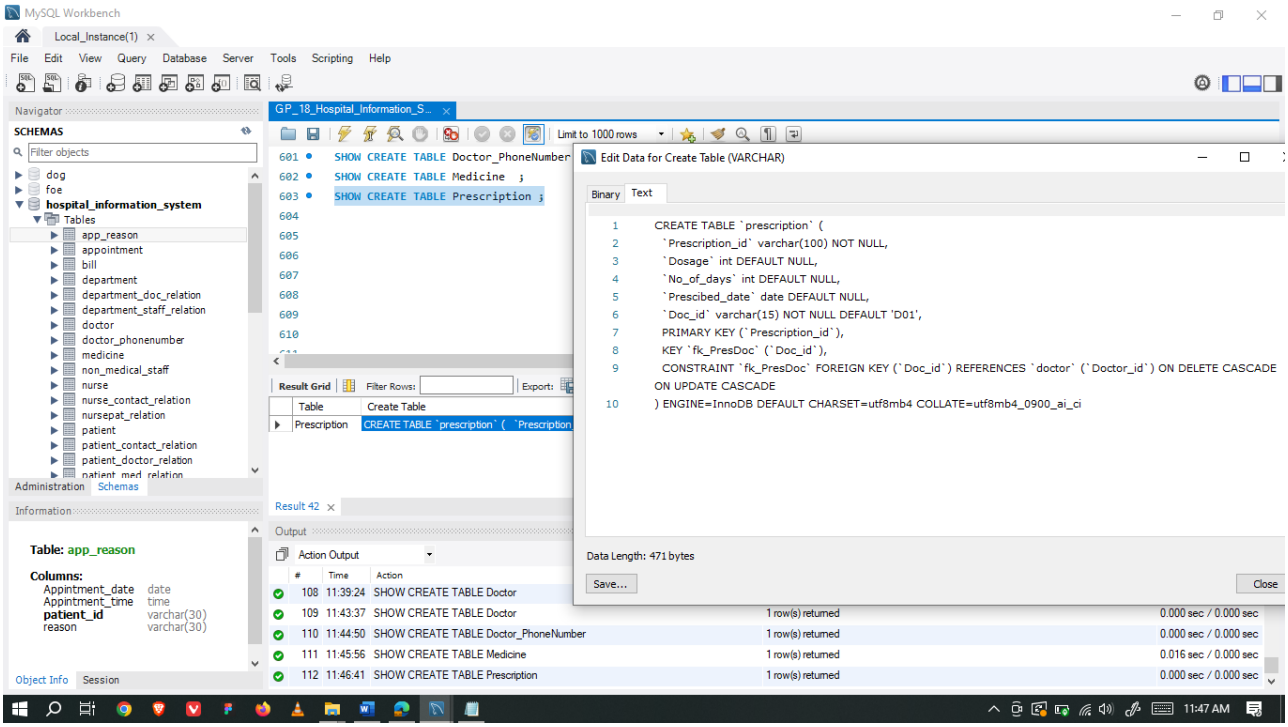
3.4.2 Doctor_PhoneNumber Table



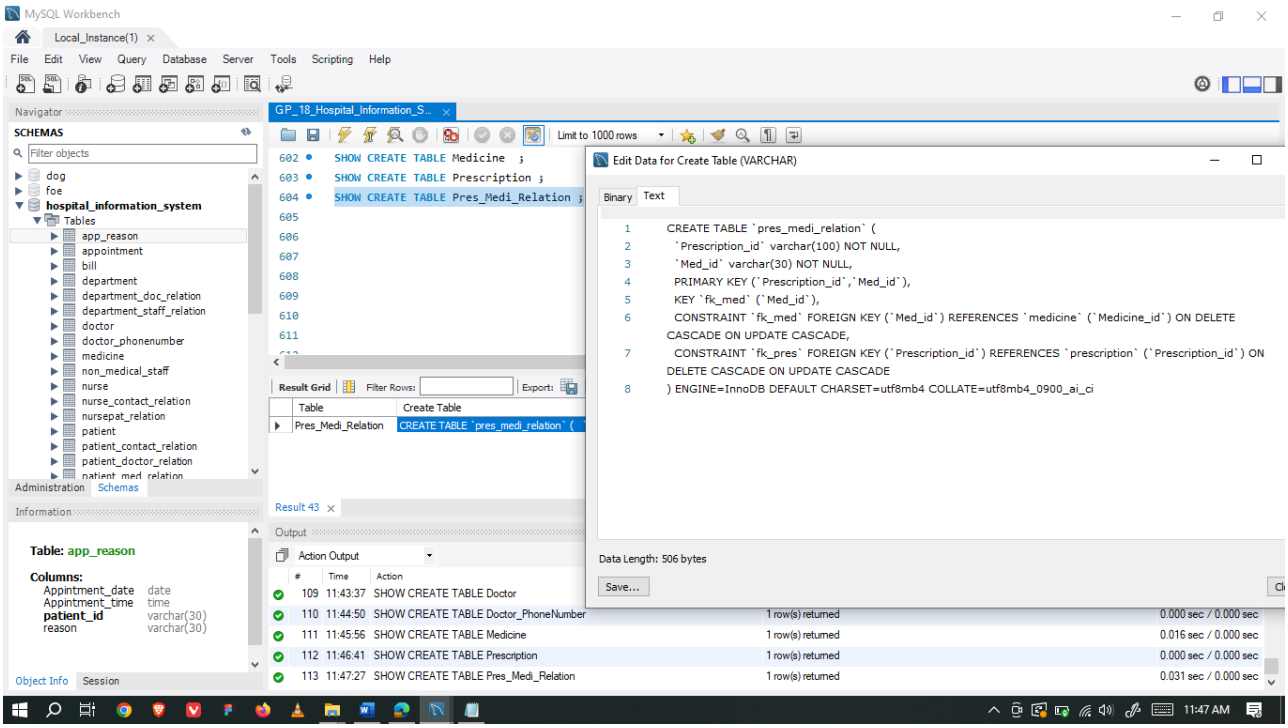
3.4.3 Medicine Table



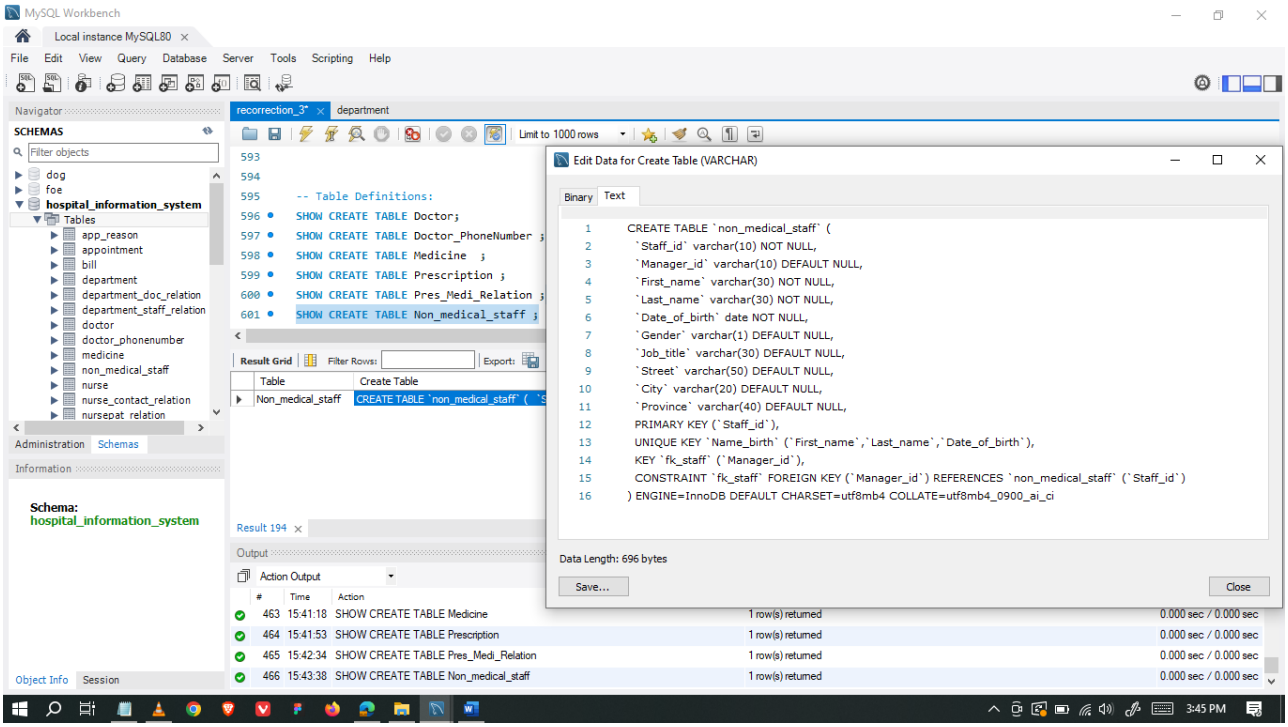
3.4.4 Prescription Table



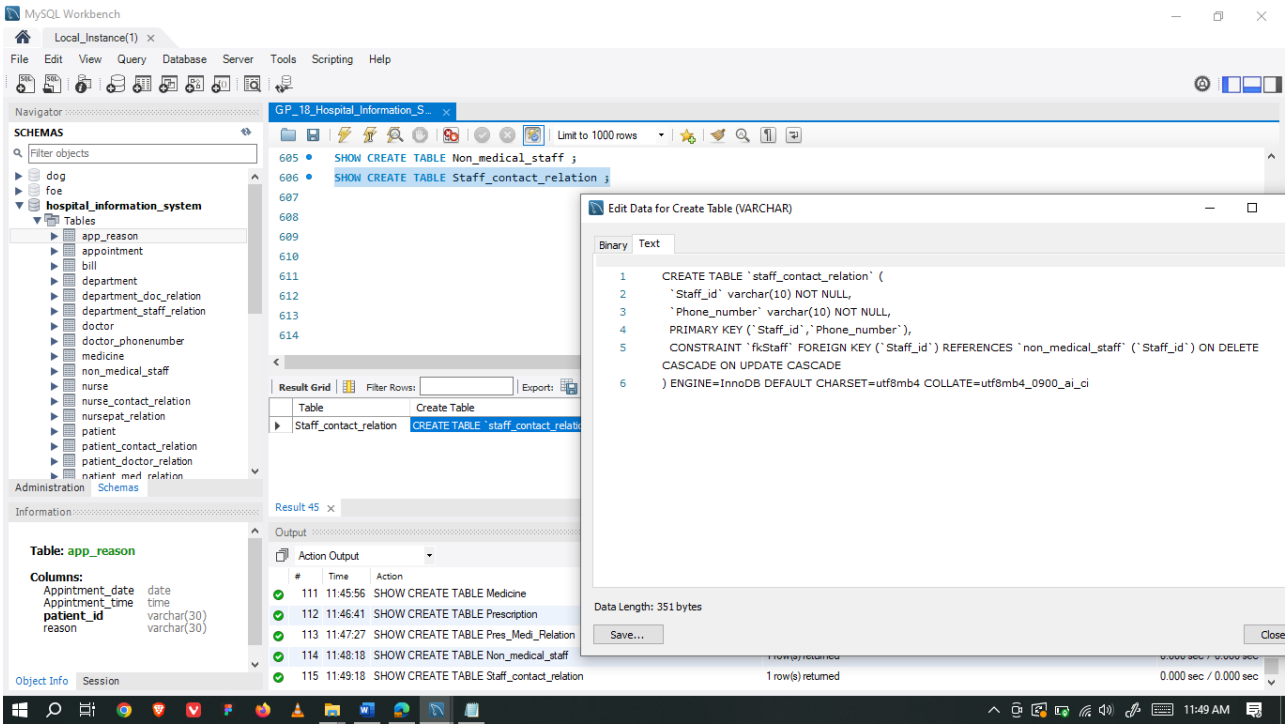
3.4.5 Pres_Medi_Relation Table



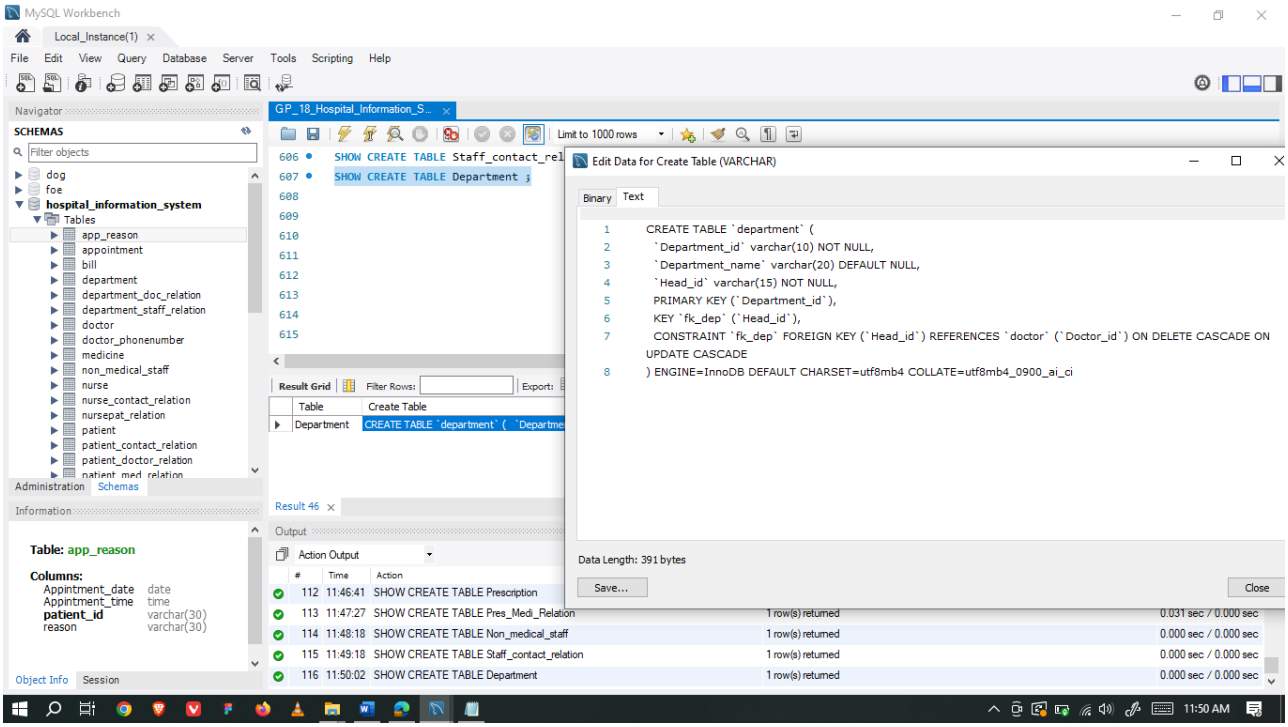
3.4.6 Non_medical_staff Table



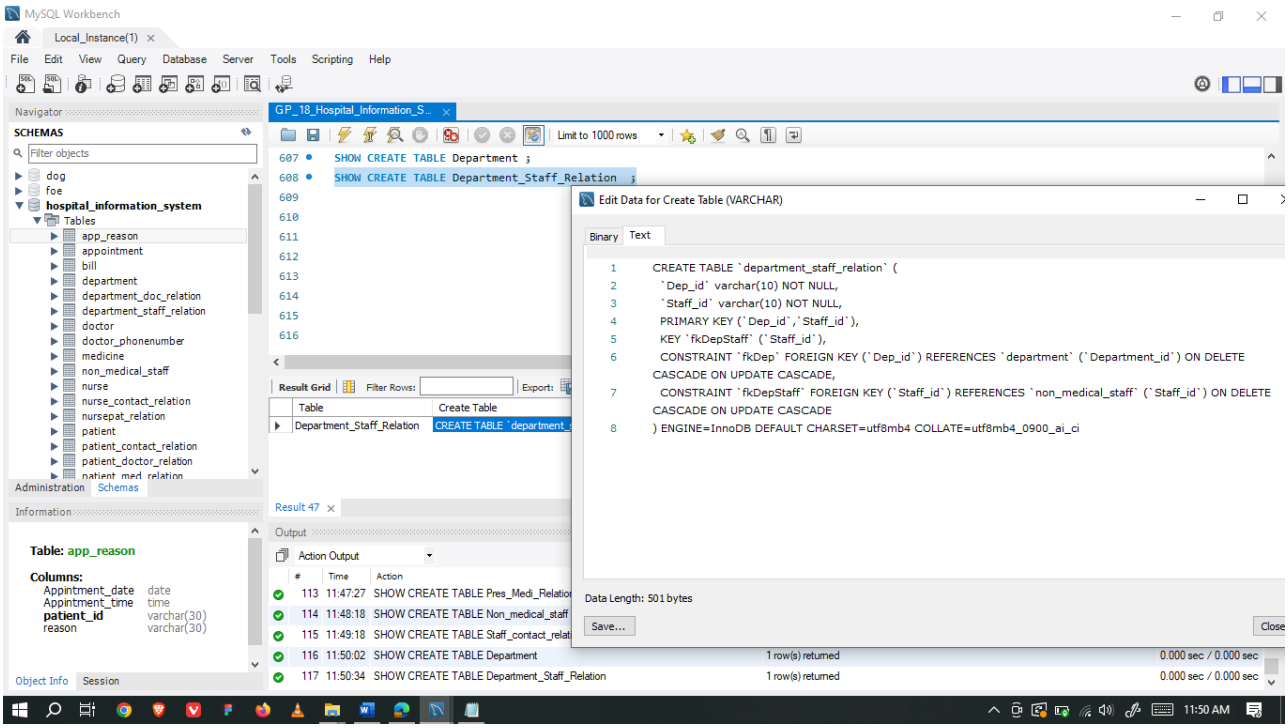
3.4.7 Staff_contact_relation Table



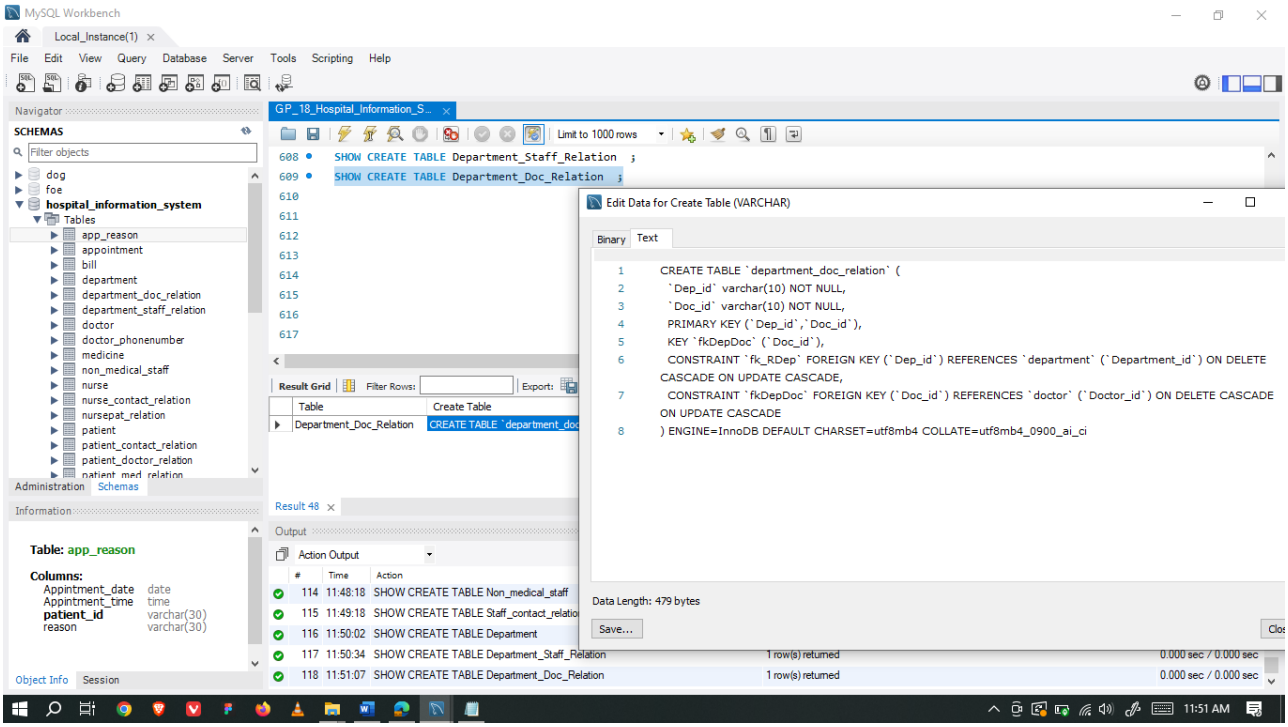
3.4.8 Department Table



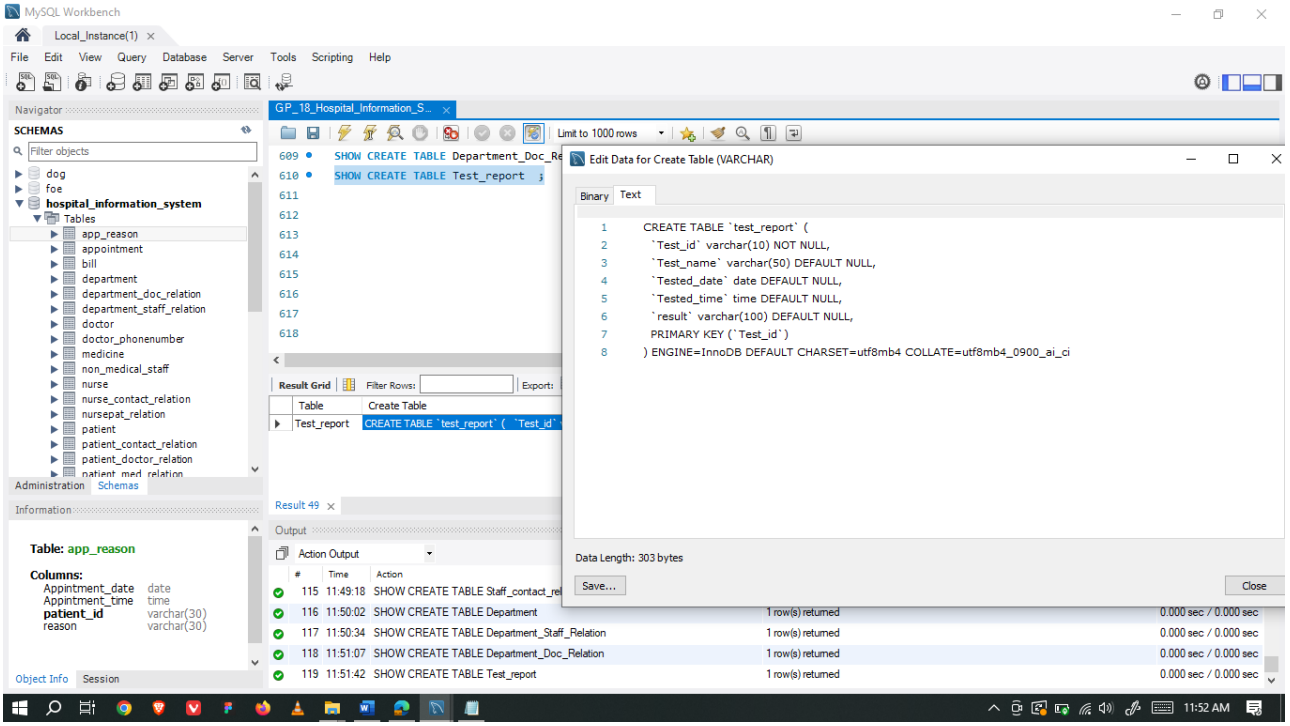
3.4.9 Department_Staff_Relation Table



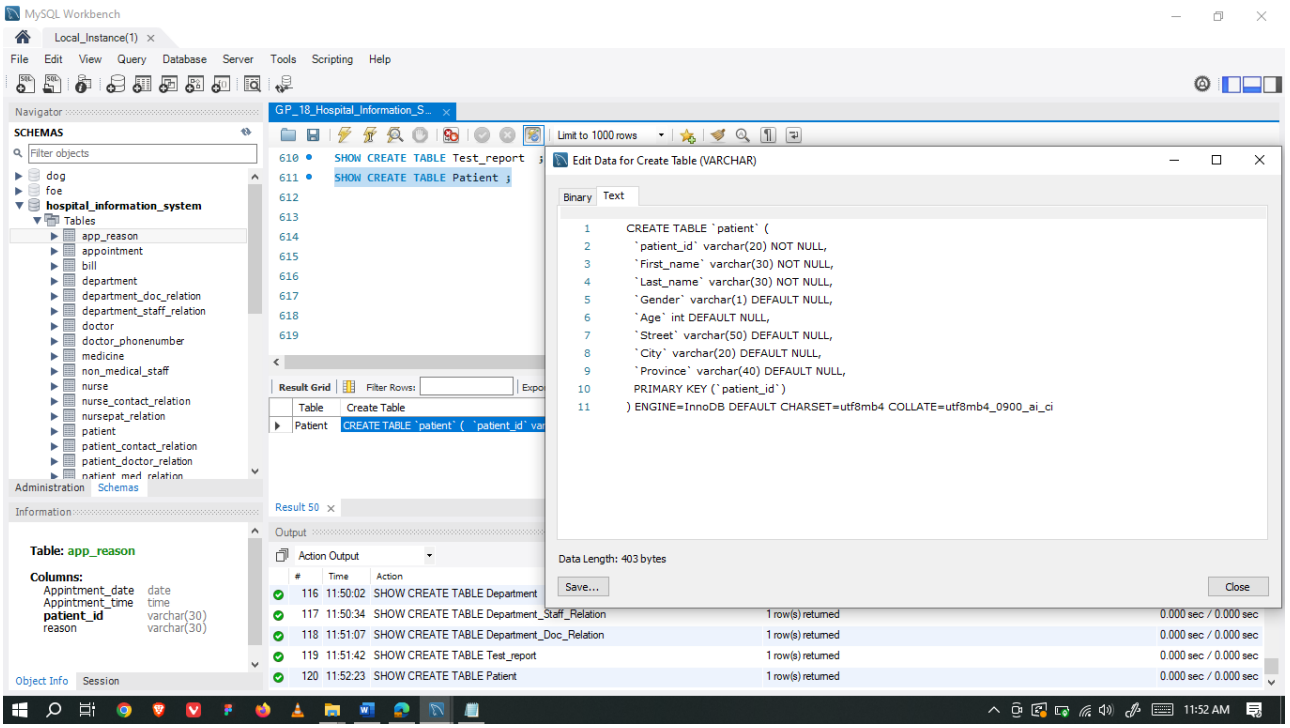
3.4.10 Department_Doc_Relation Table



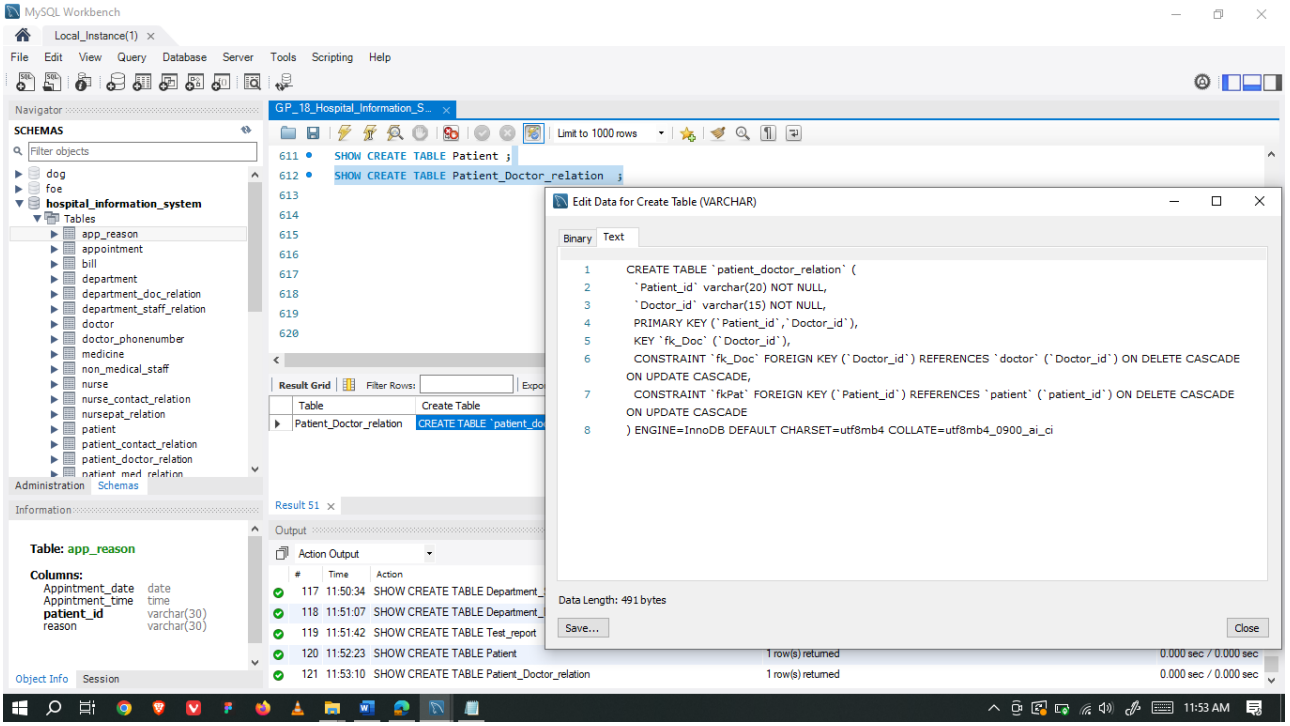
3.4.11 Test_report Table



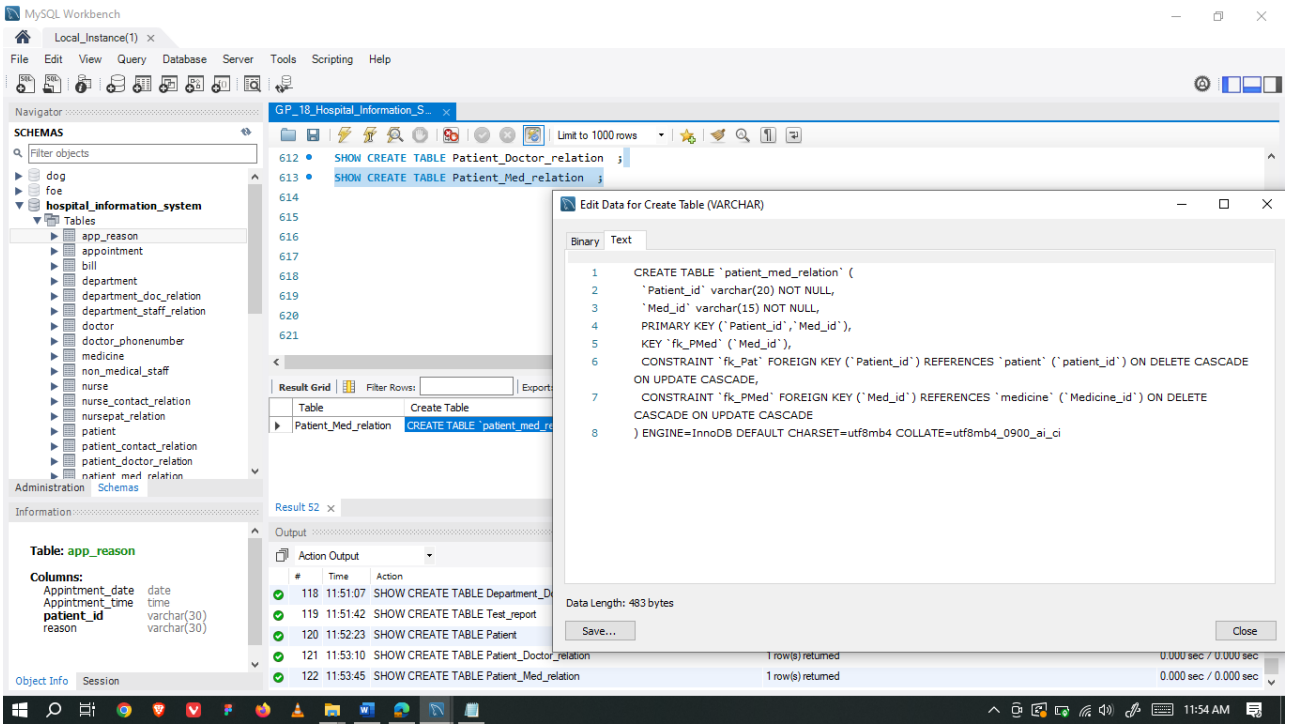
3.4.12 Patient Table



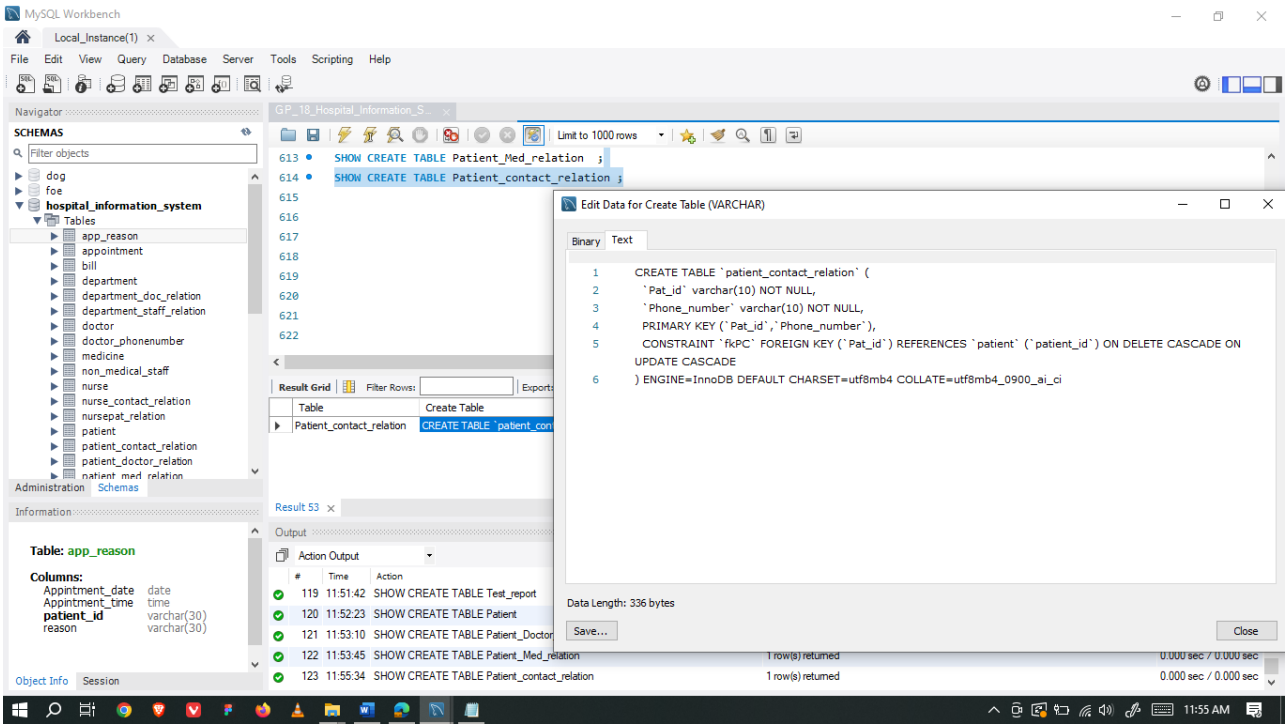
3.4.13 Patient_Doctor_relation Table



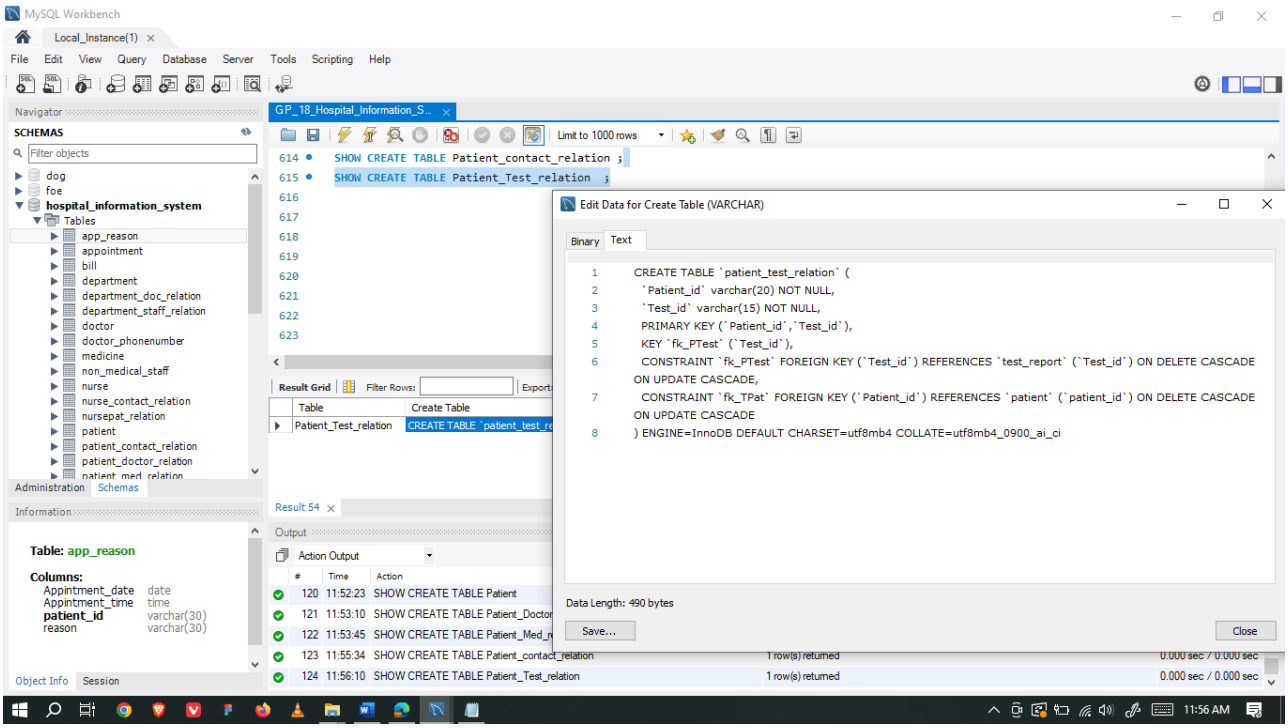
3.4.14 Patient_Med_relation Table



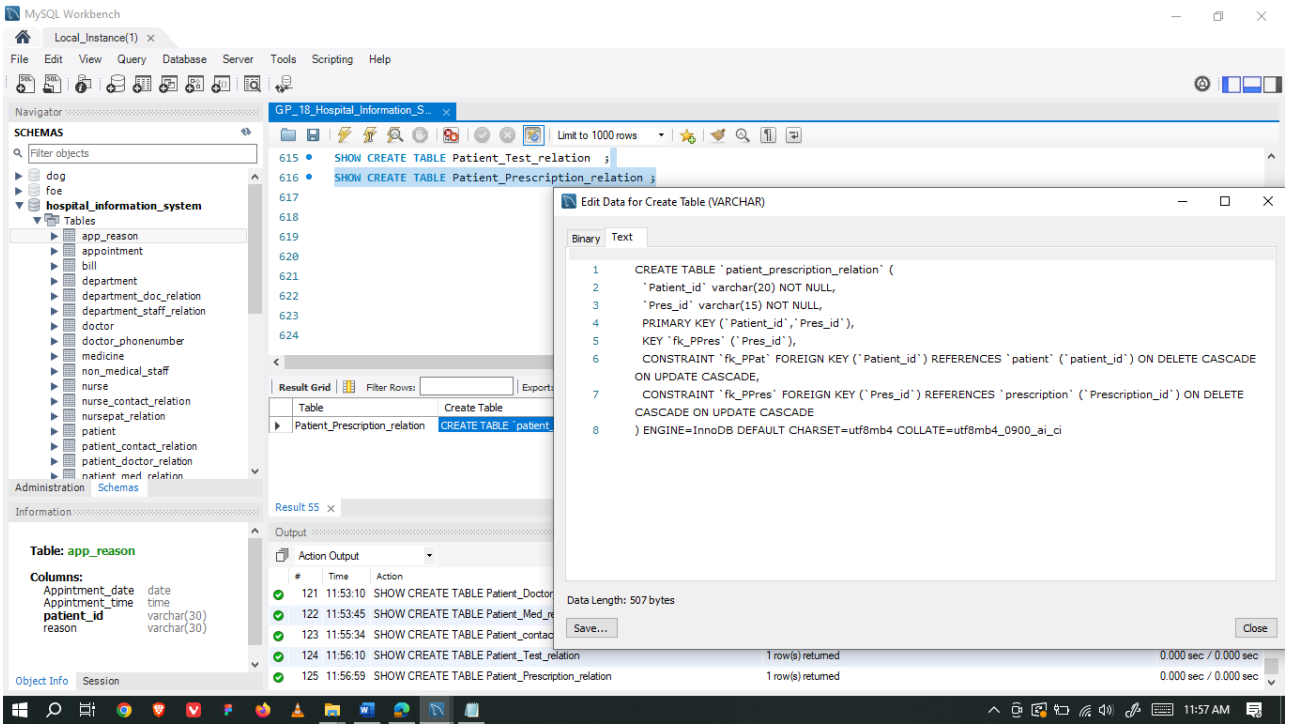
3.4.15 Patient_contact_relation Table



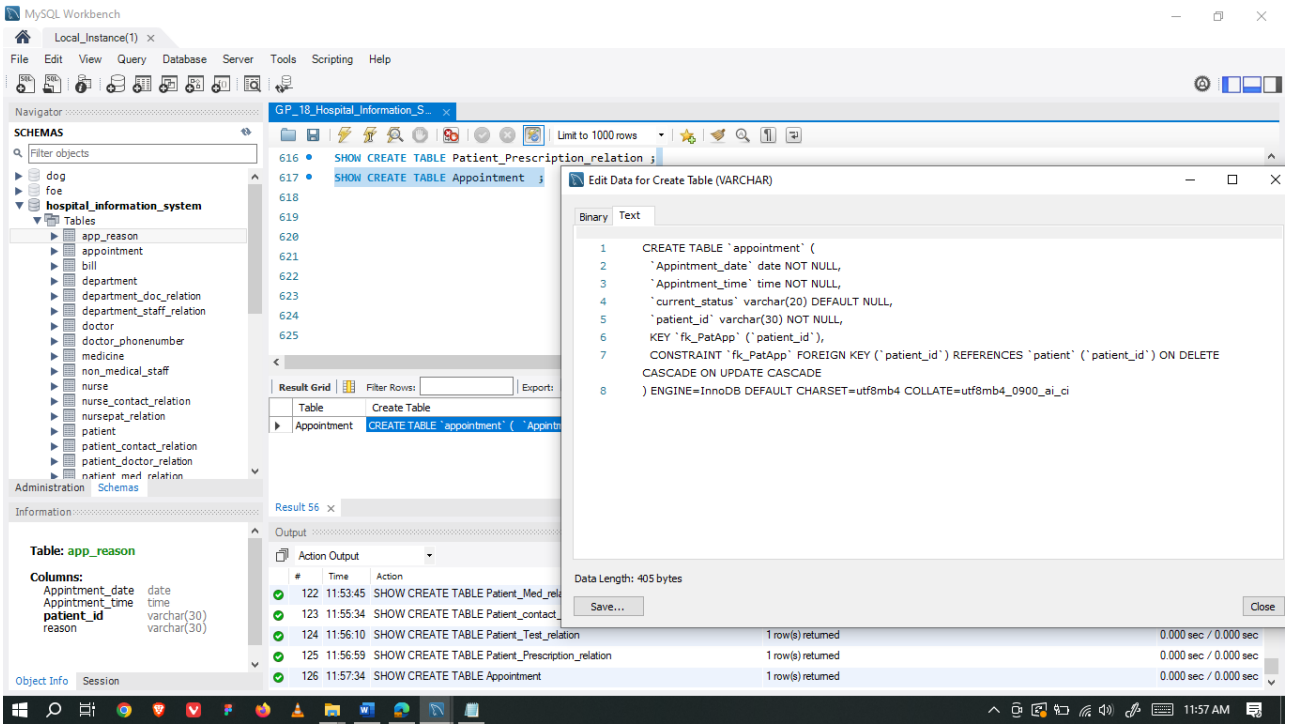
3.4.16 Patient_Test_relation Table



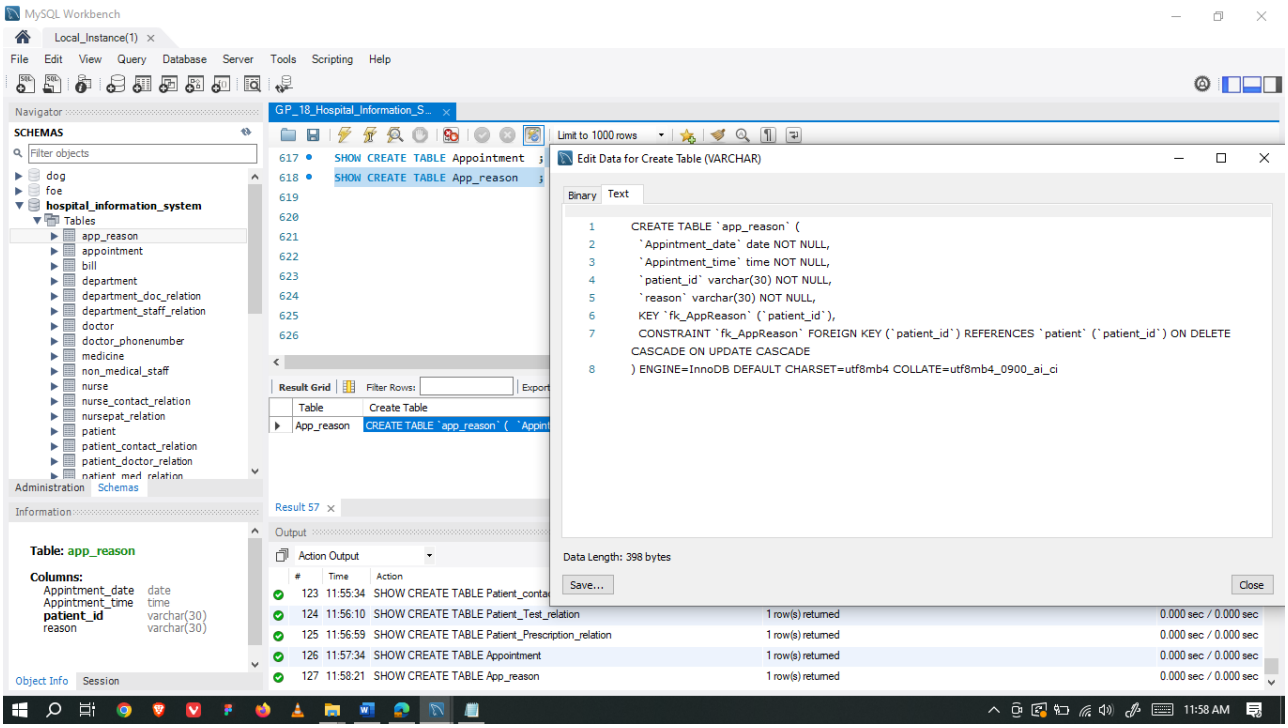
3.4.17 Patient_Prescription_relation Table



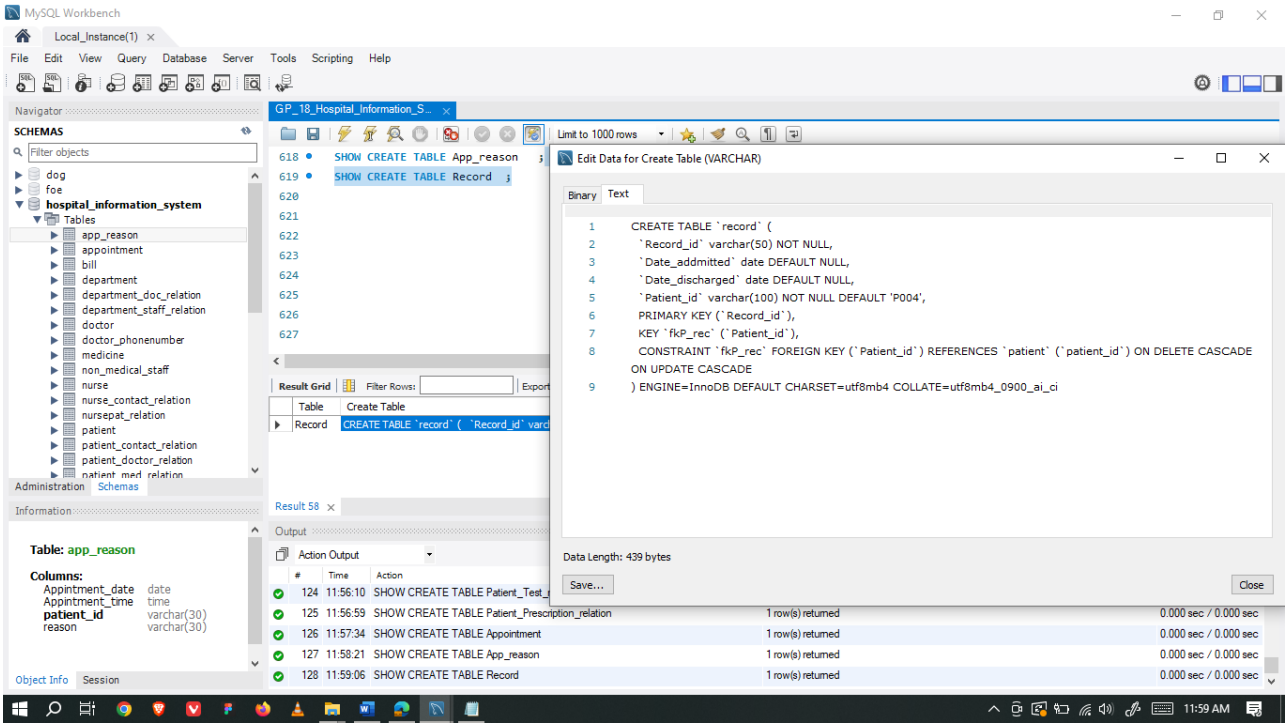
3.4.18 Appointment Table



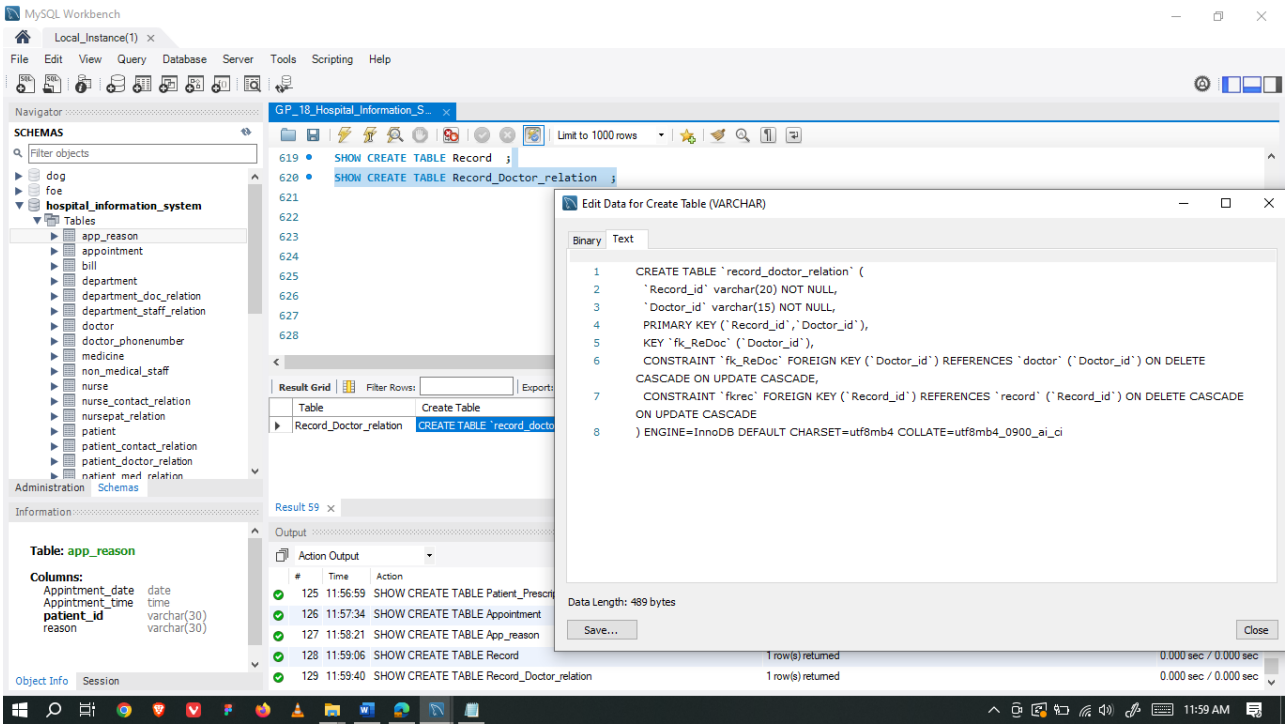
3.4.19 App_reason Table



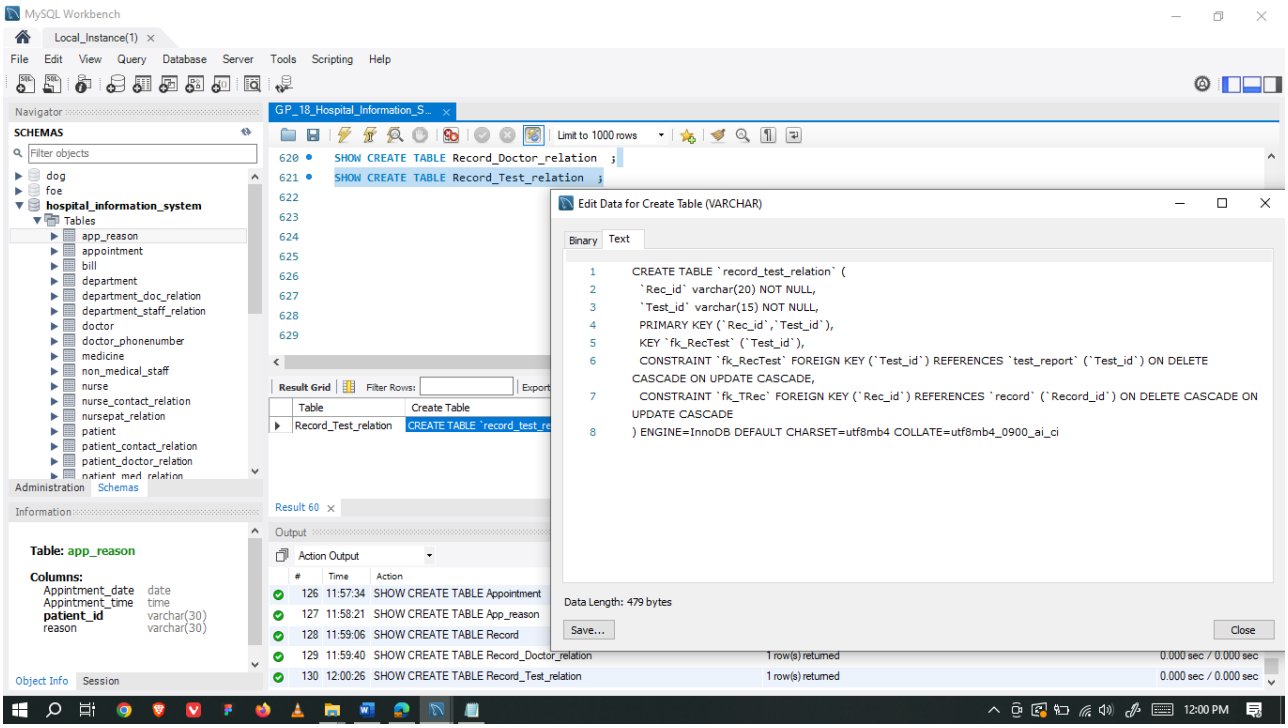
3.4.20 Record Table



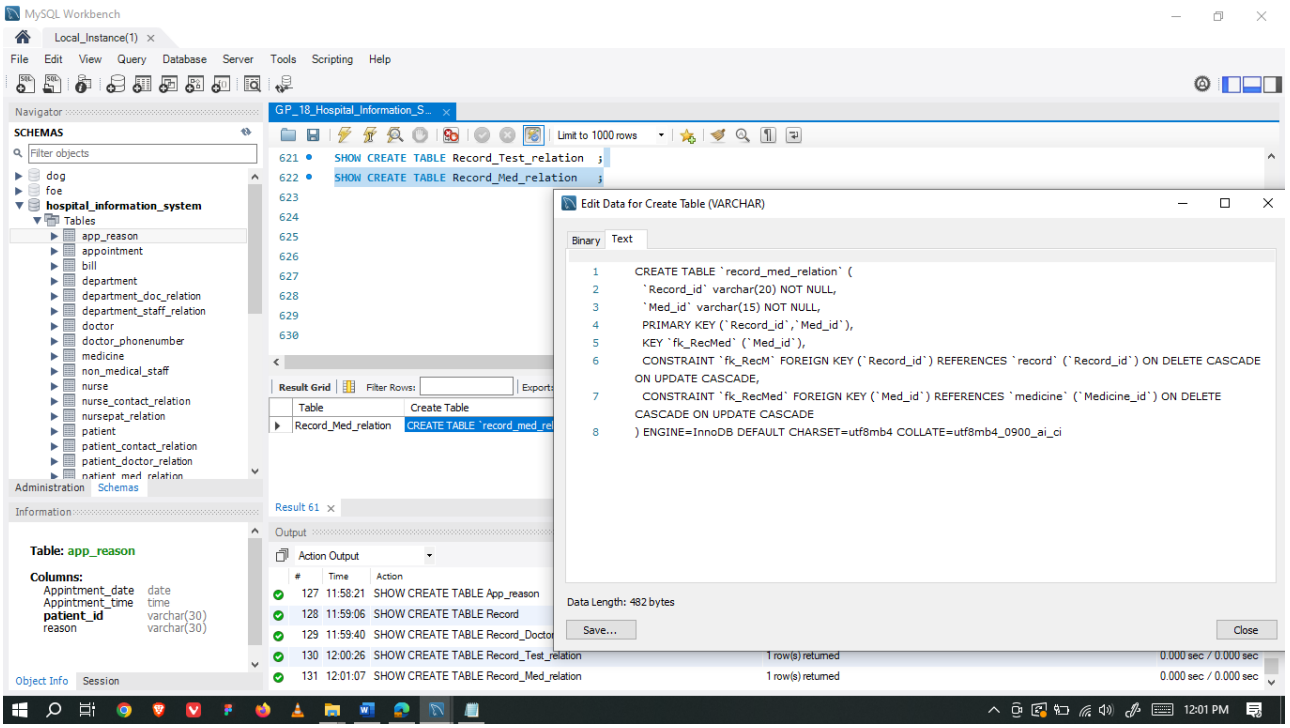
3.4.21 Record_Doctor_relation Table



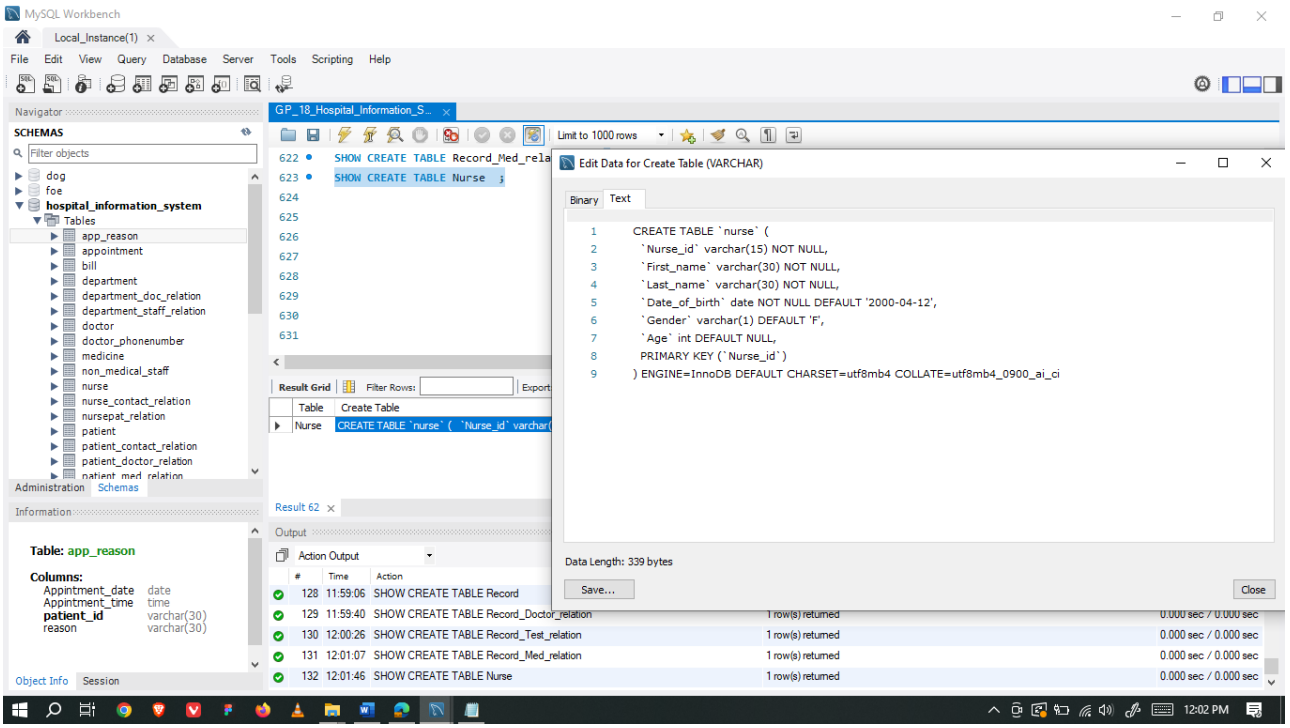
3.4.22 Record_Test_relation Table



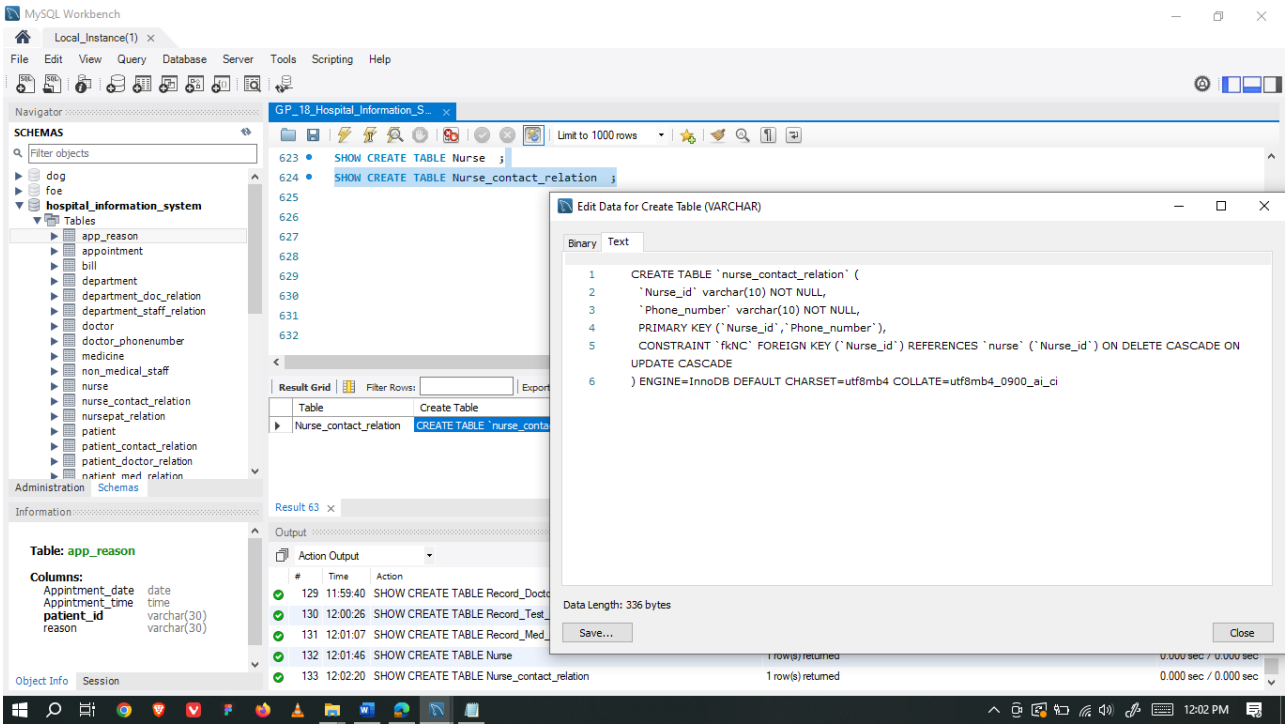
3.4.23 Record_Med_relation Table



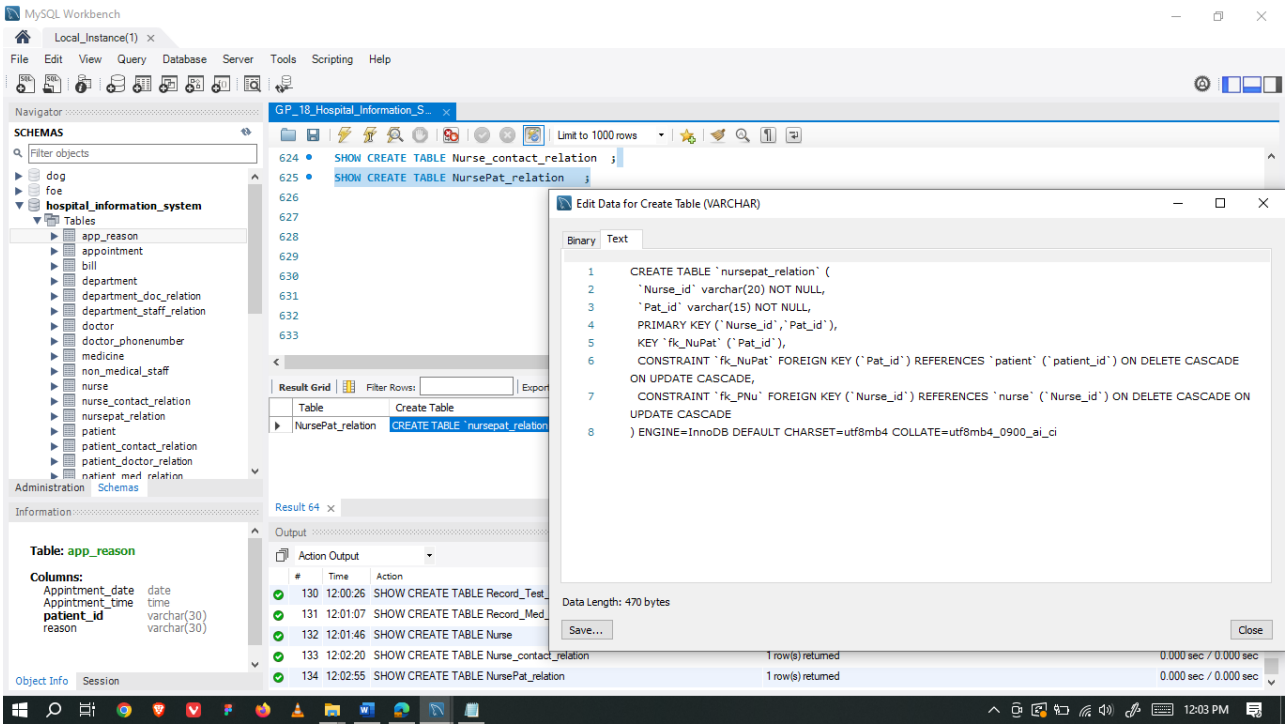
3.4.24 Nurse Table



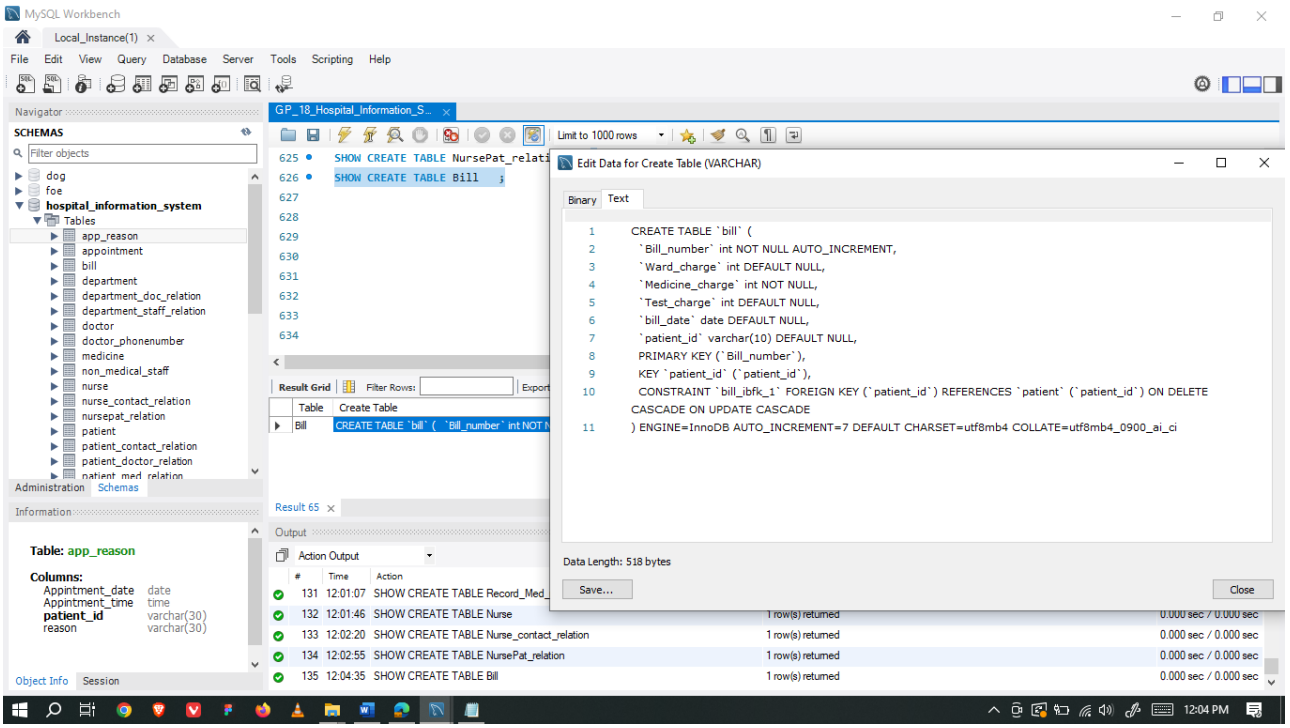
3.4.25 Nurse_contact_relation Table



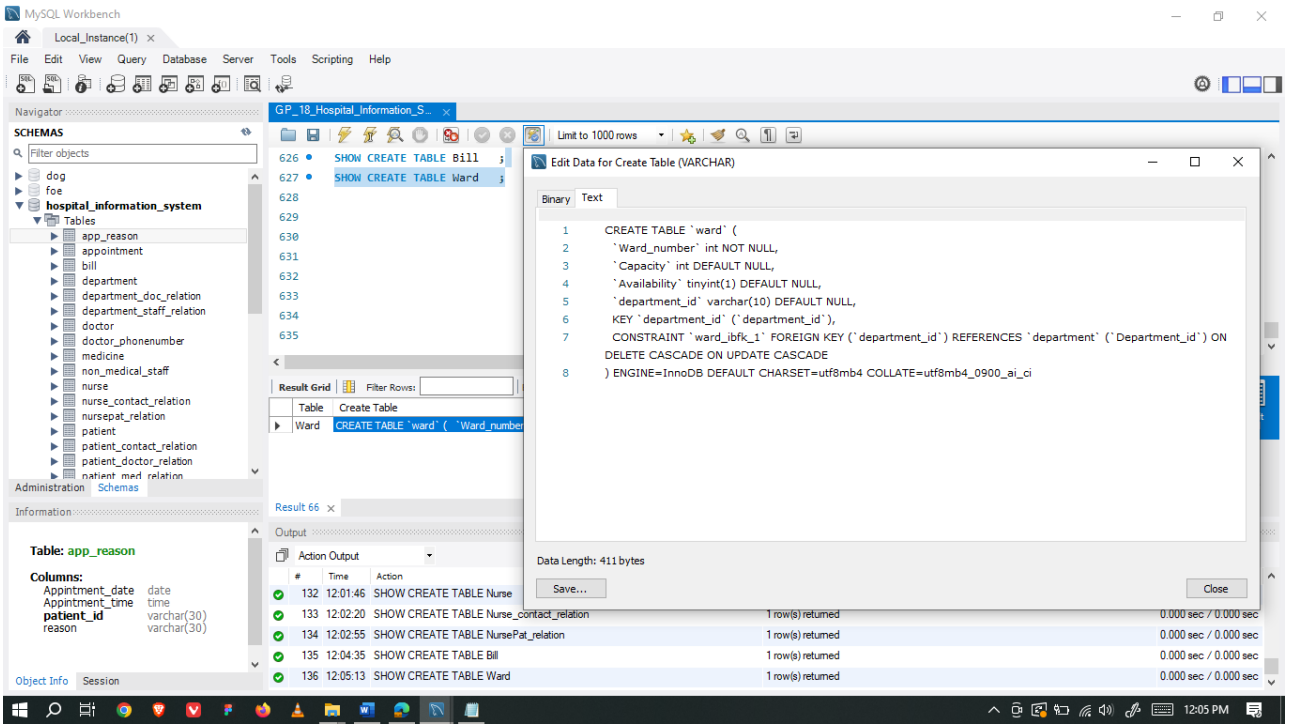
3.4.26 NursePat_relation Table



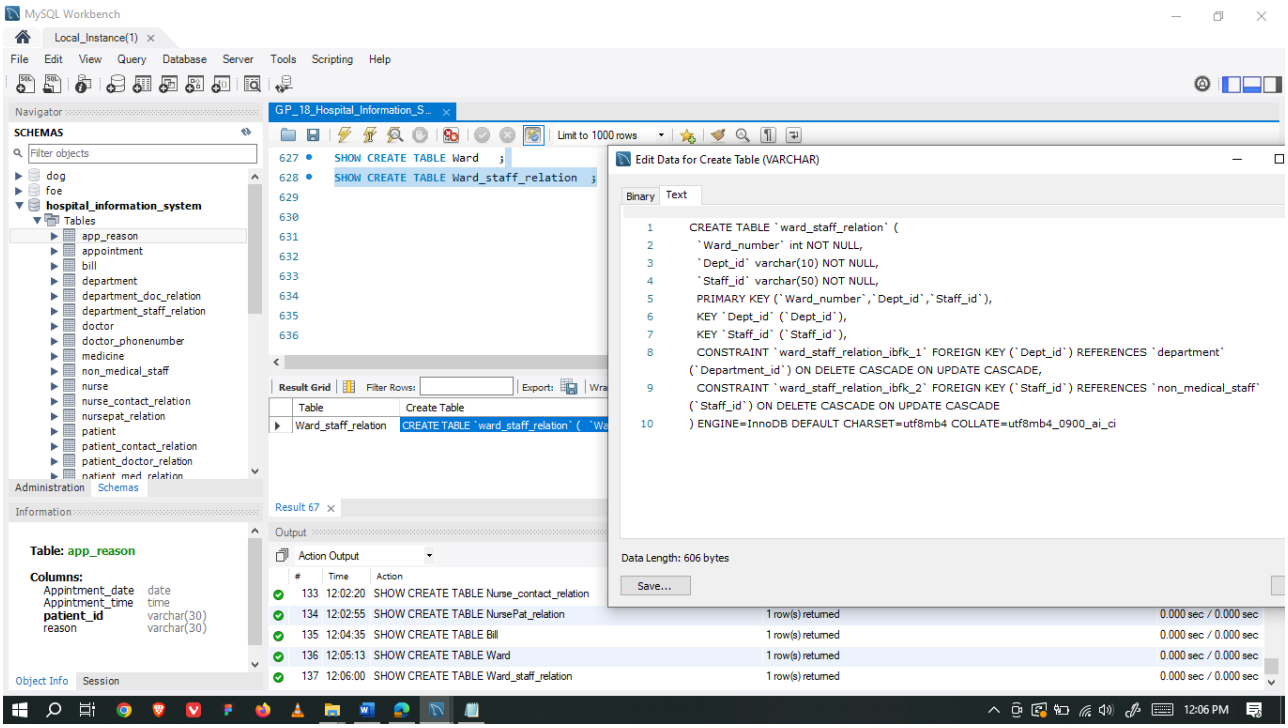
3.4.27 Bill Table



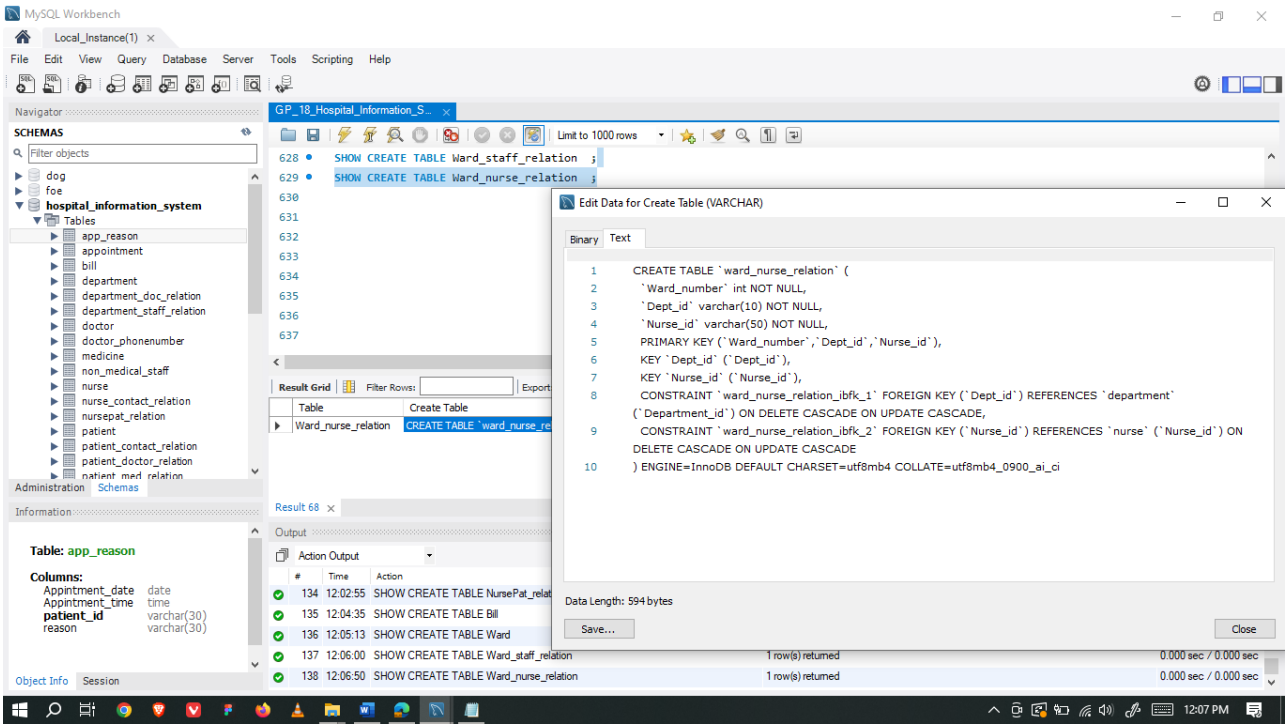
3.4.28 Ward Table



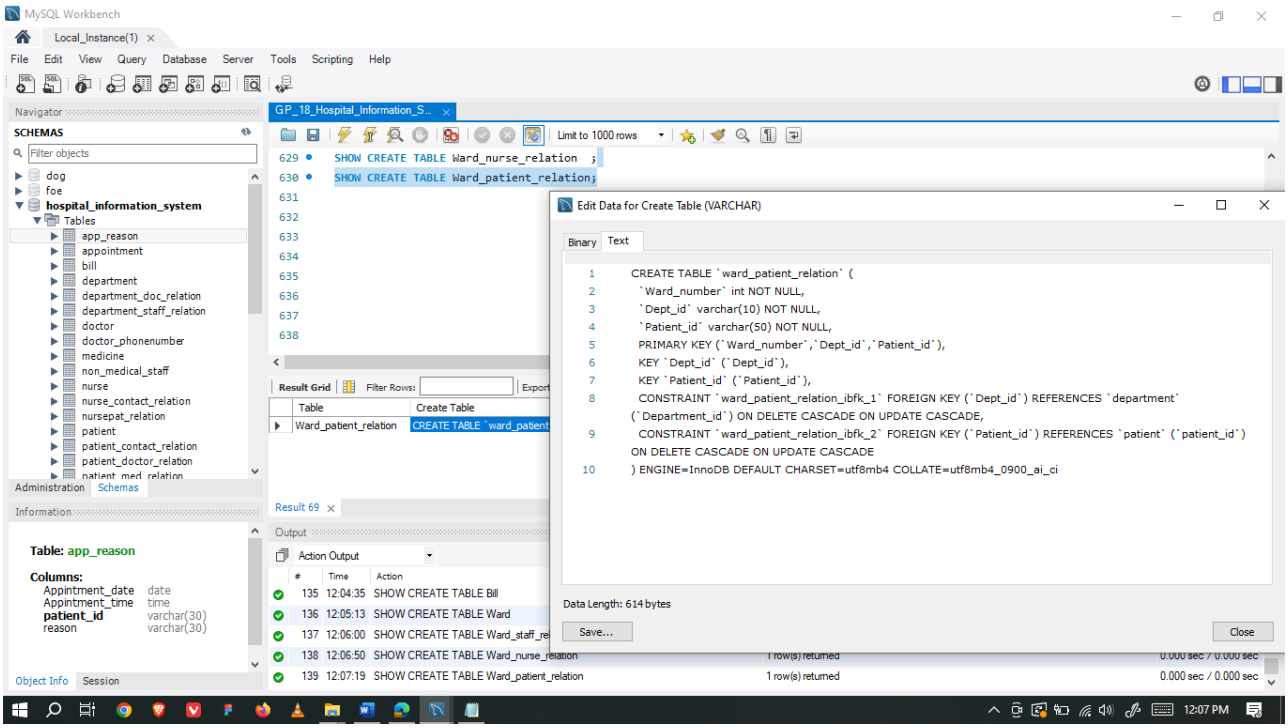
3.4.29 Ward_staff_relation Table



3.4.30 Ward_nurse_relation Table

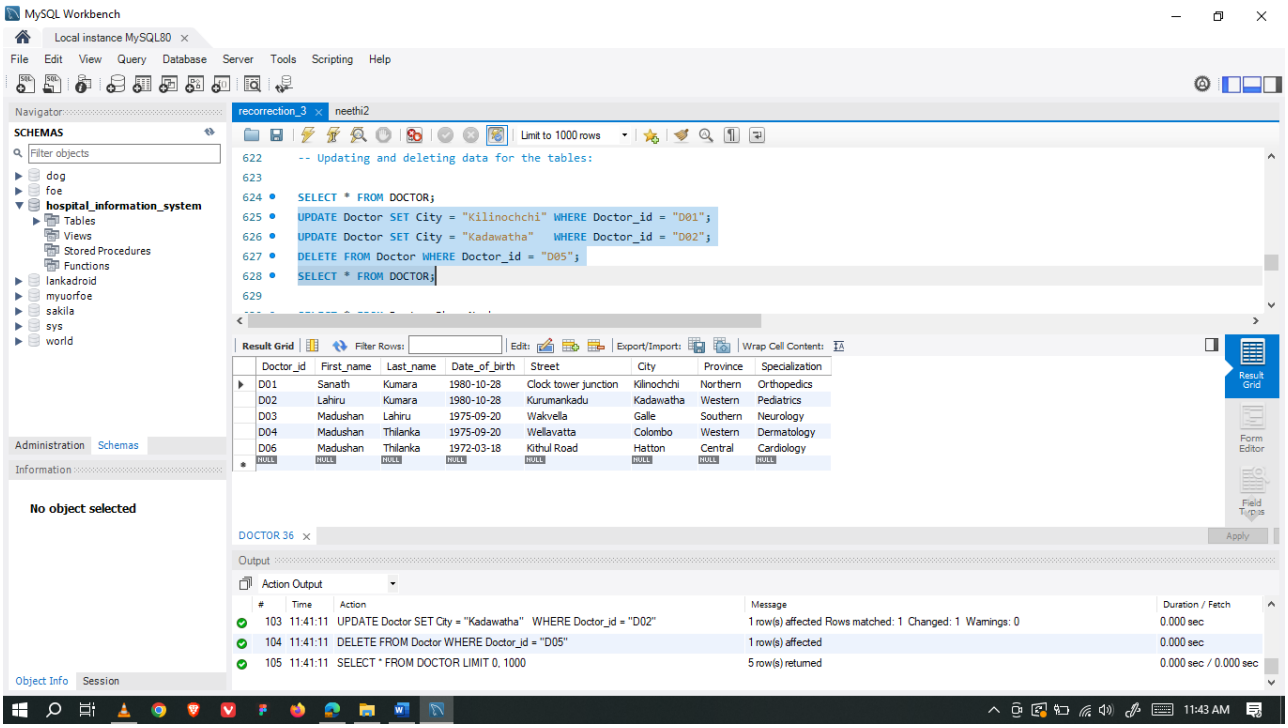


3.4.31 Ward_patient_relation Table

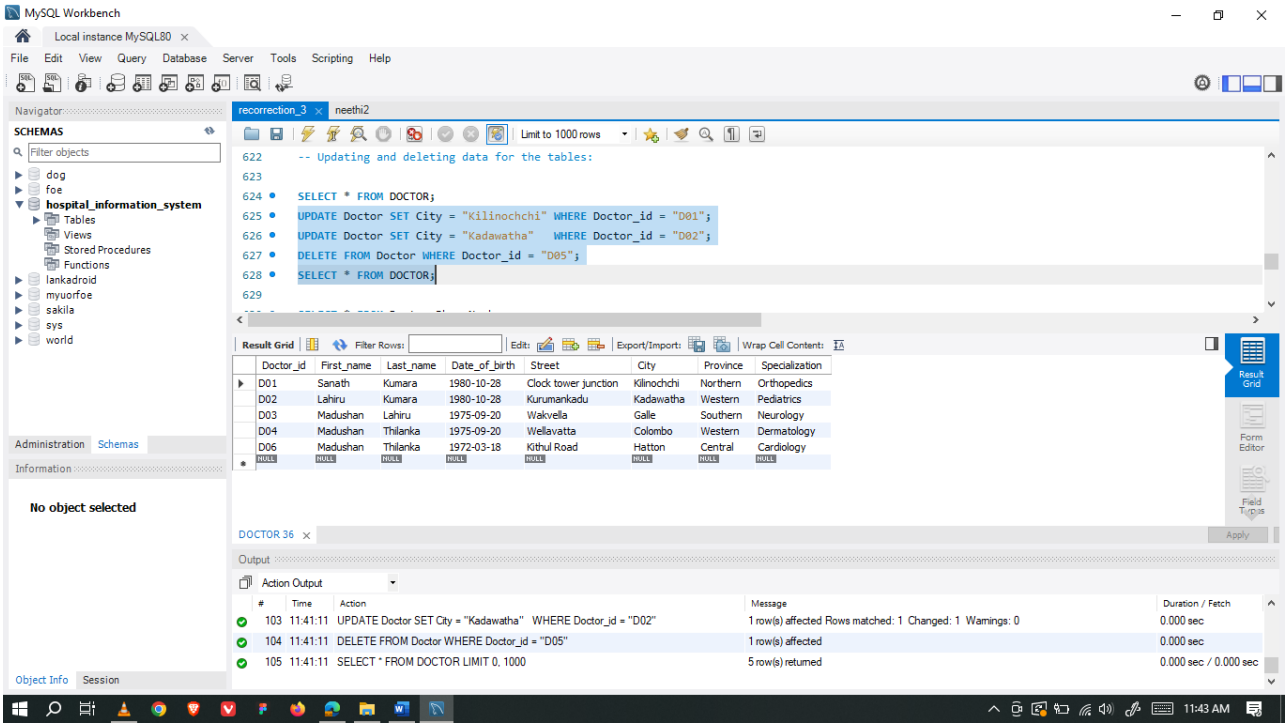


3.5 Updating & Deleting data of tables

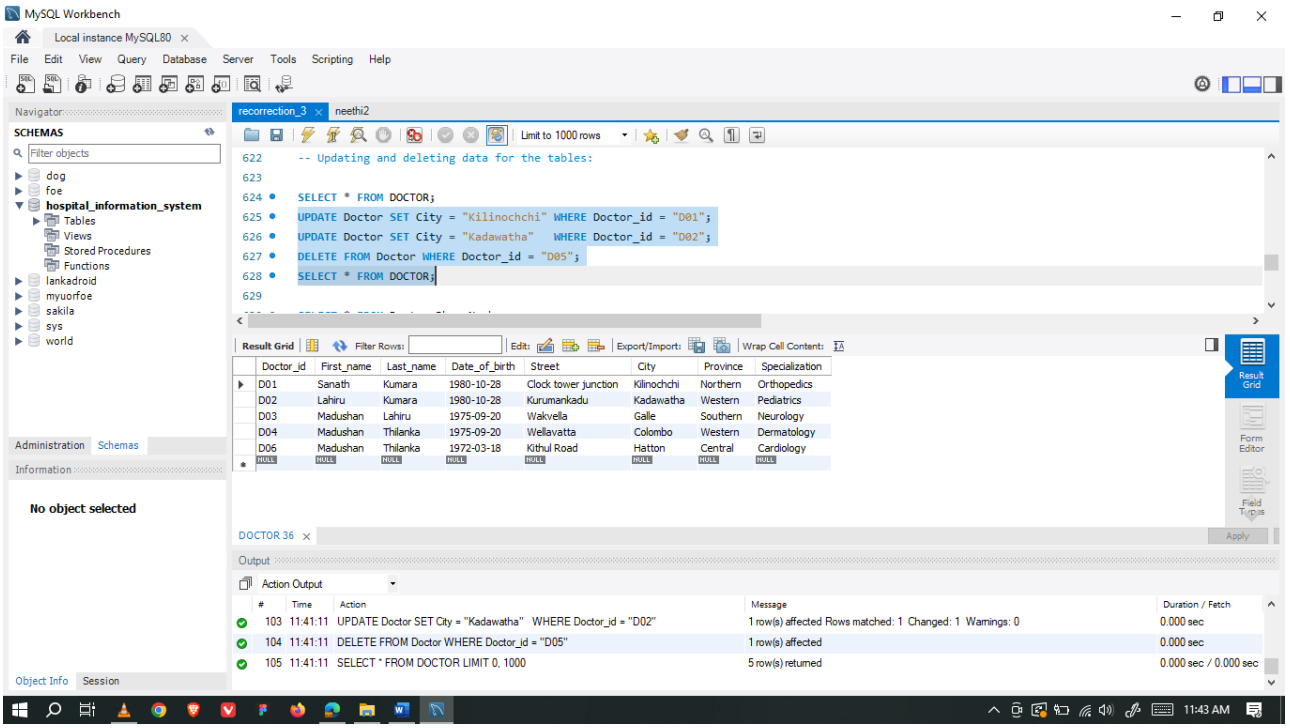
3.5.1 Doctor Table



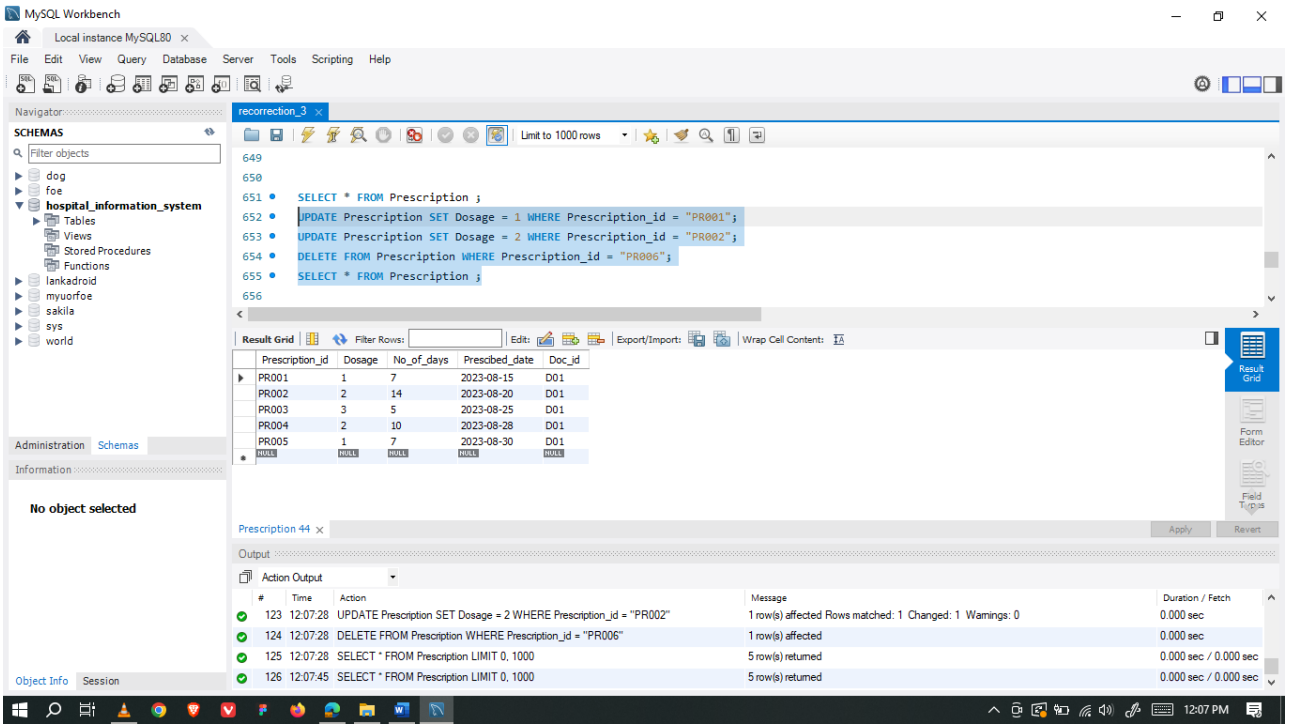
3.5.2 Doctor_PhoneNumber



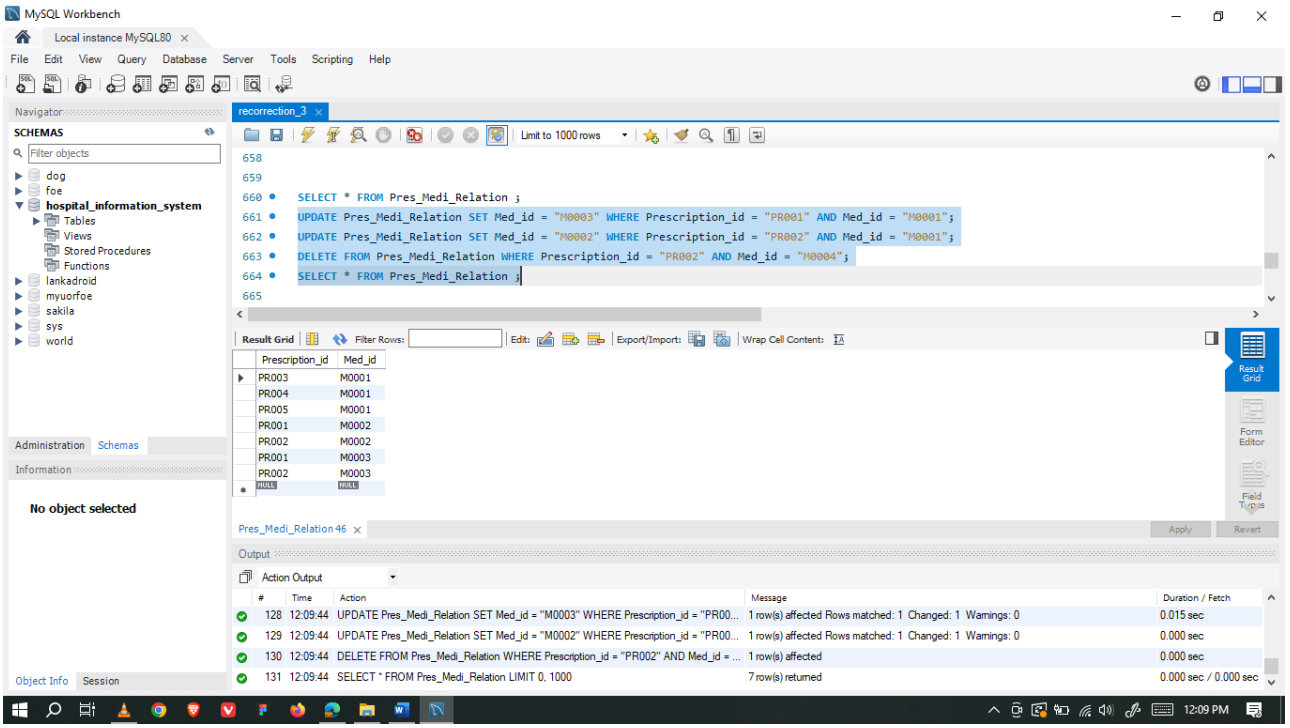
3.5.3 Medicine



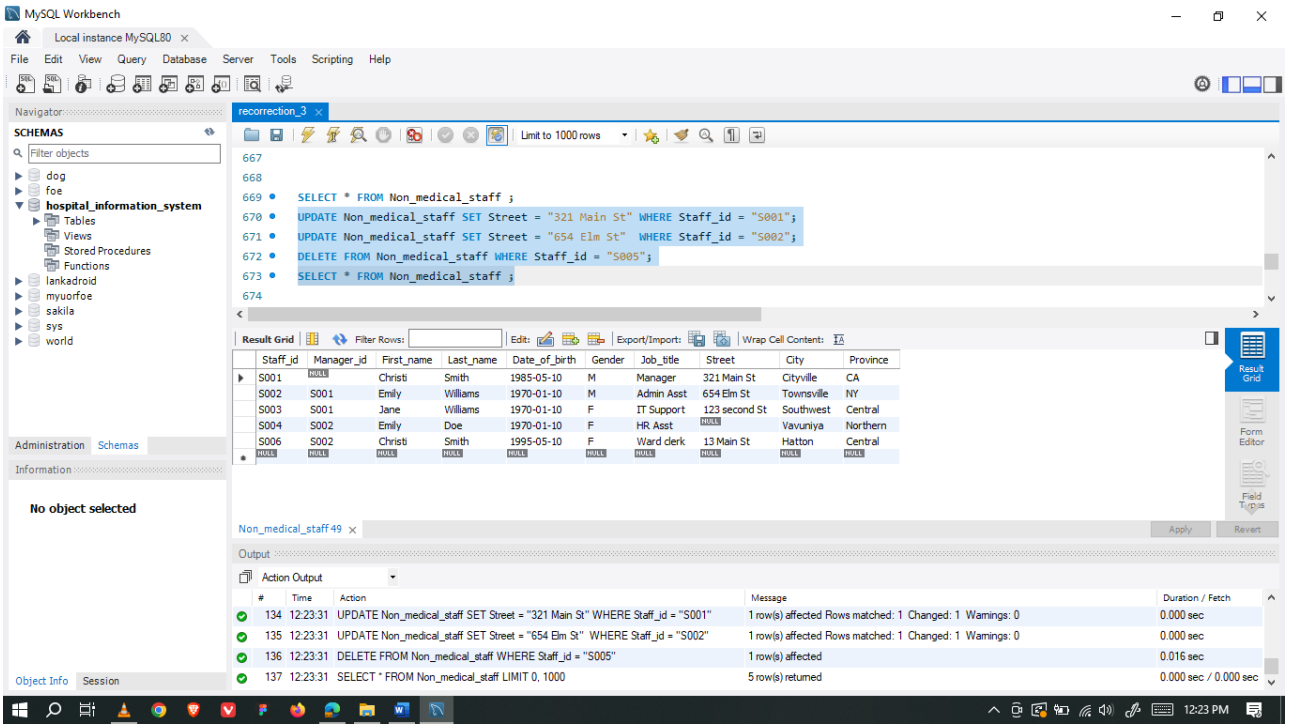
3.5.4 Prescription



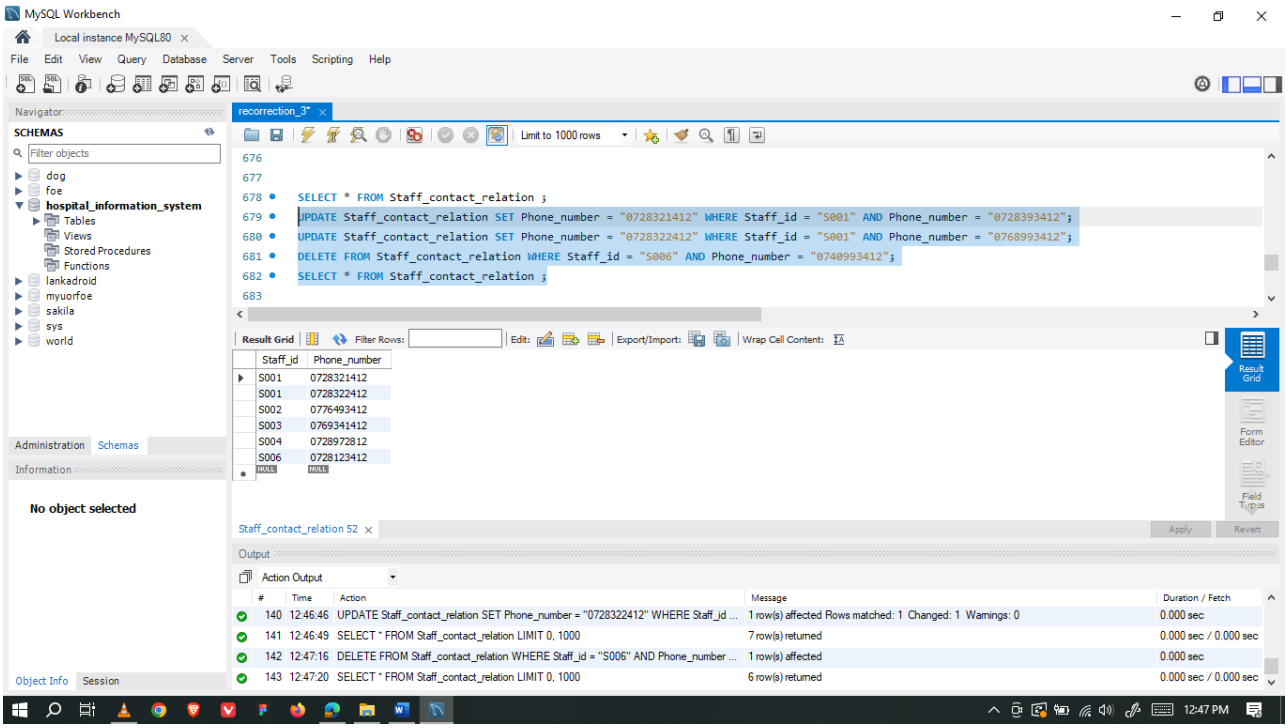
3.5.5 Pres_Medi_Relation



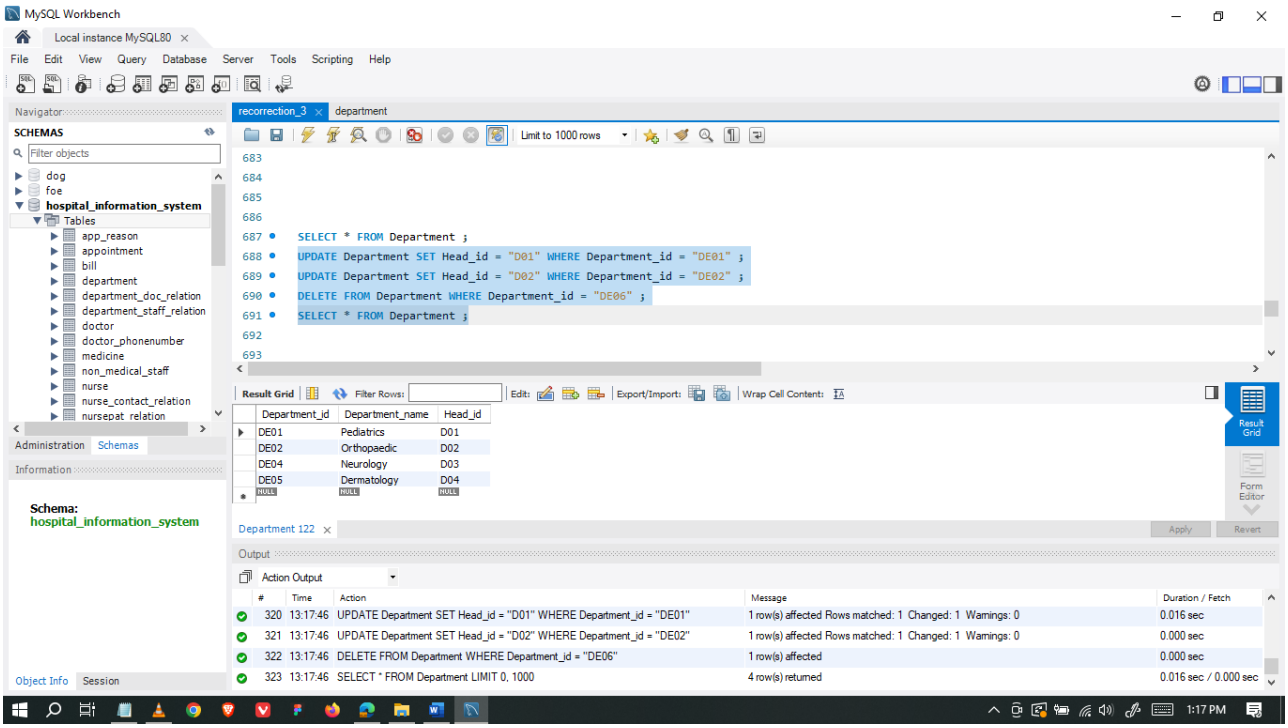
3.5.6 Non_medical_staff



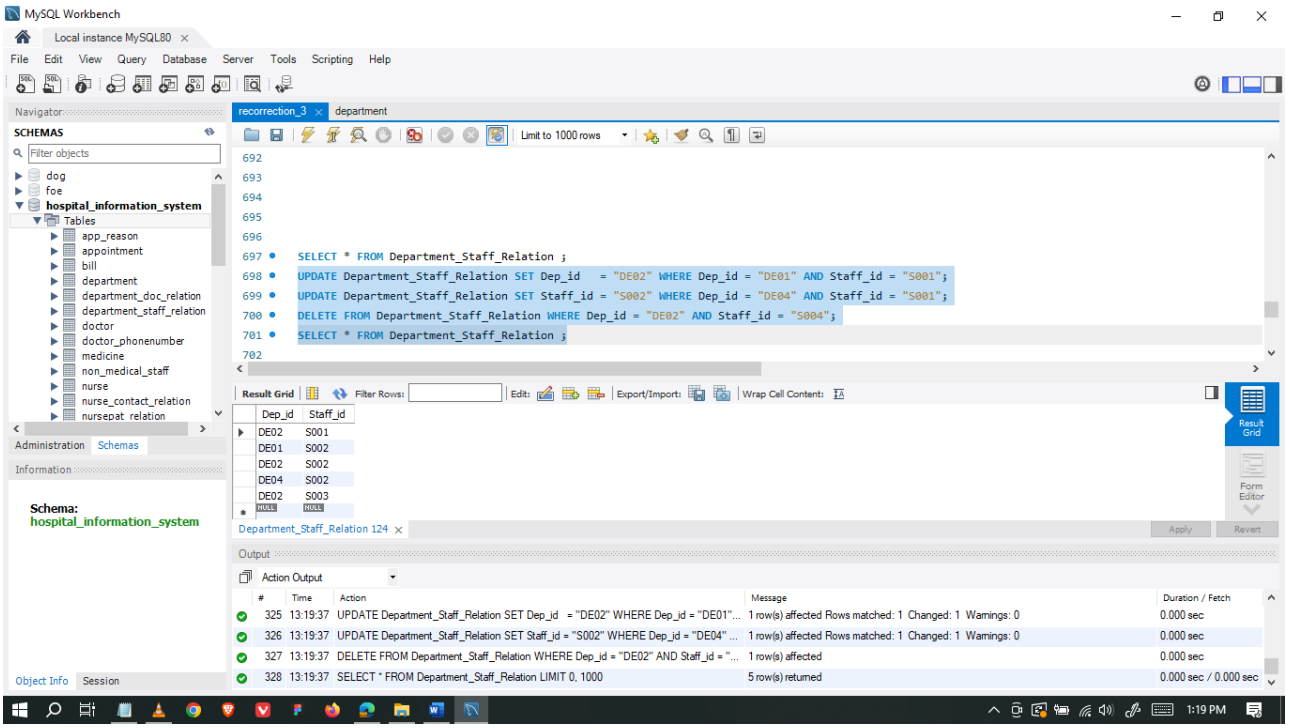
3.5.7 Staff_contact_relation



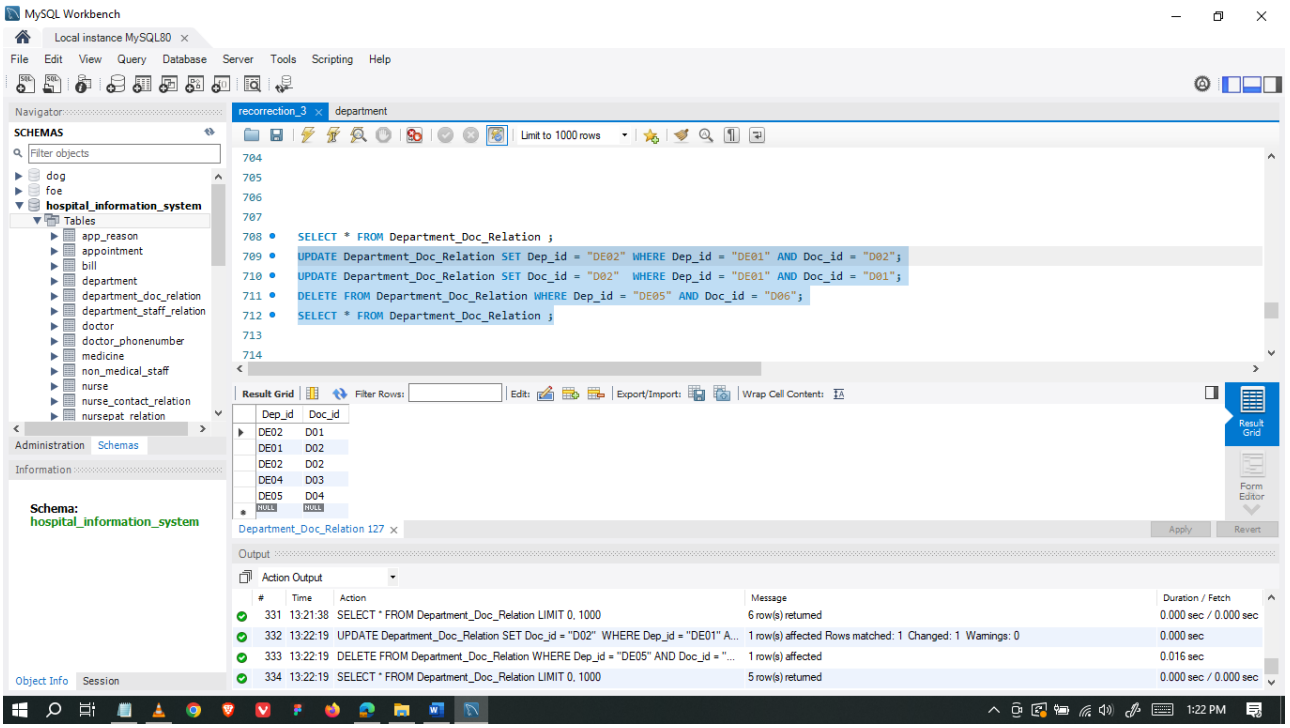
3.5.8 Department



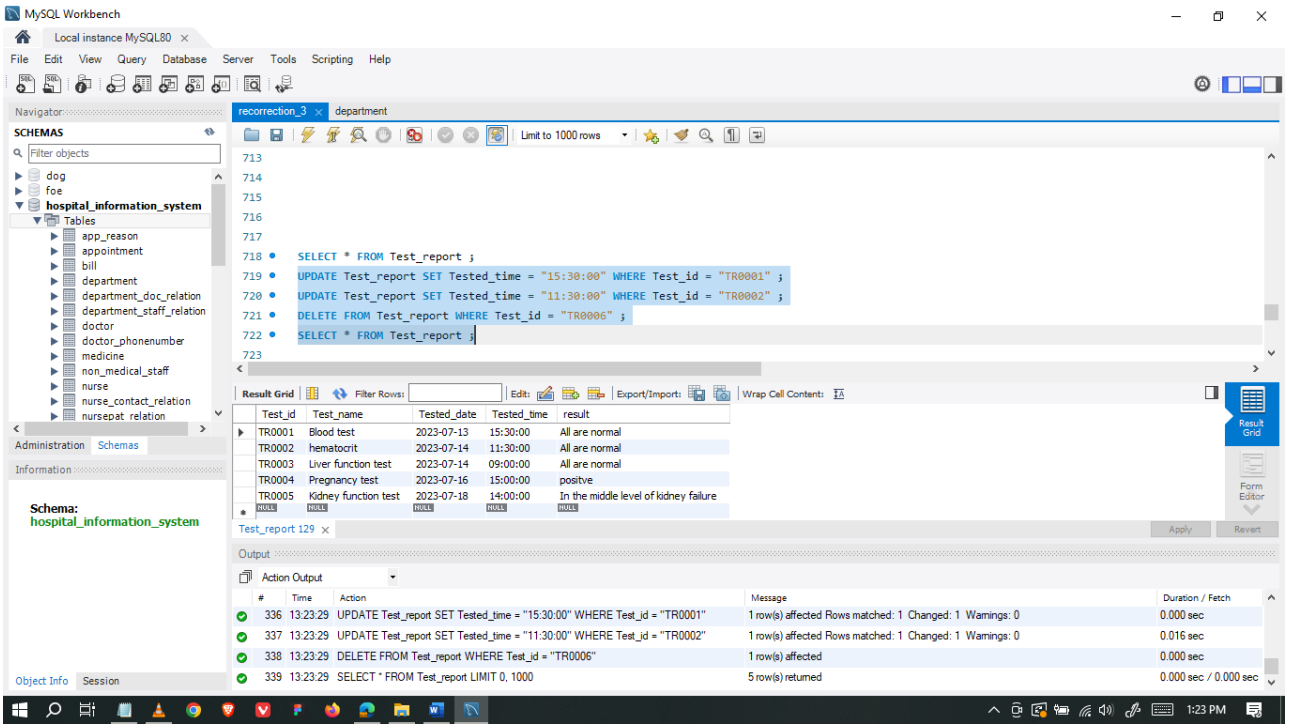
3.5.9 Department_Staff_Relation



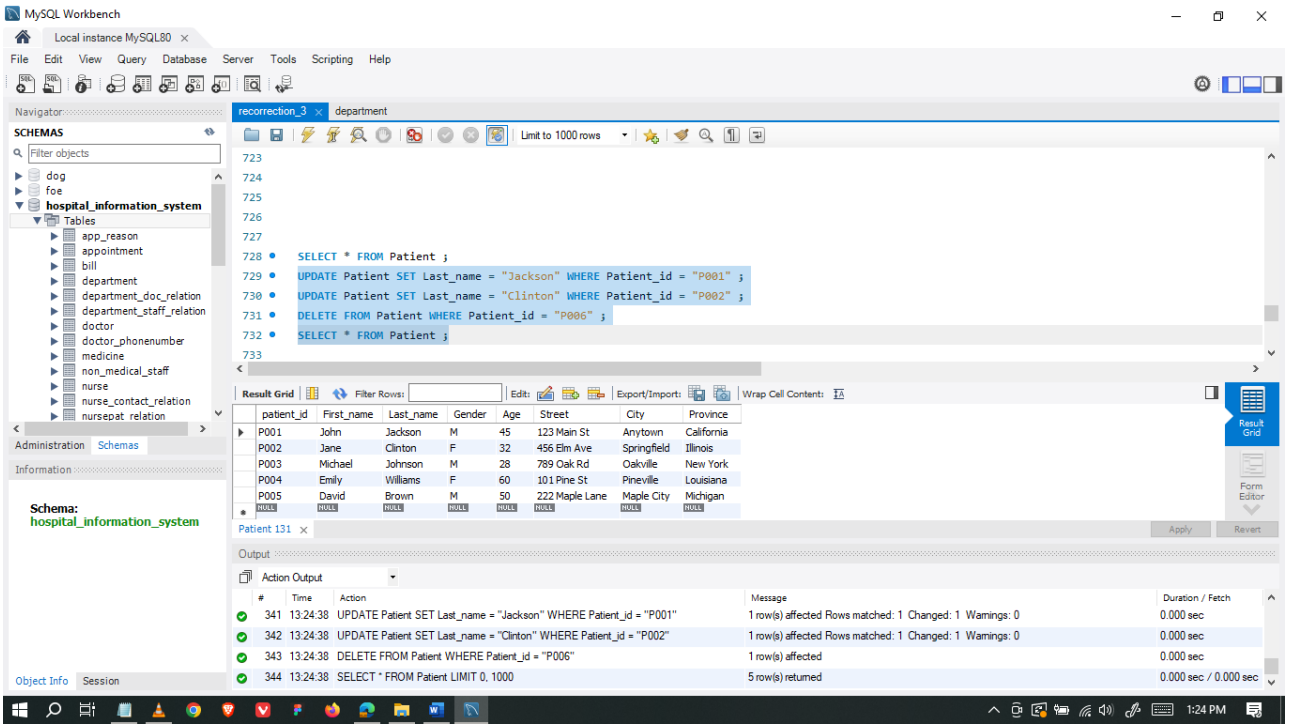
3.5.10 Department_Doc_Relation



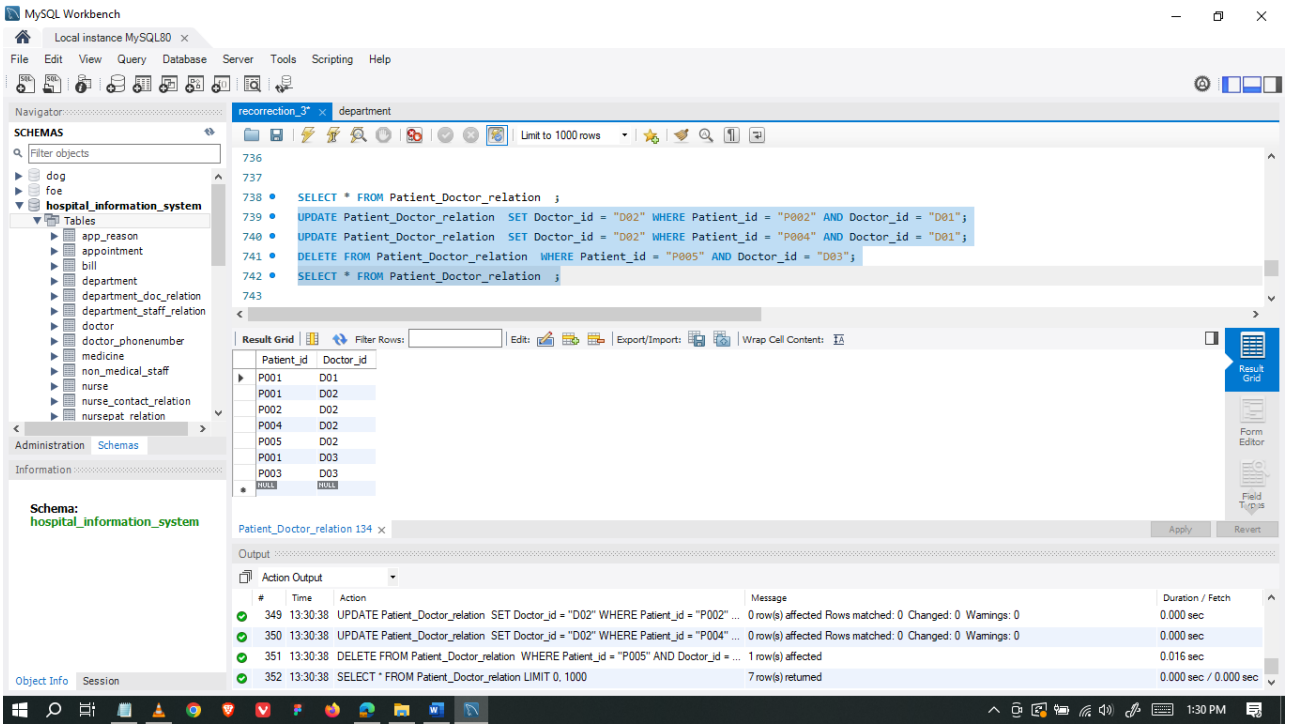
3.5.11 Test_report



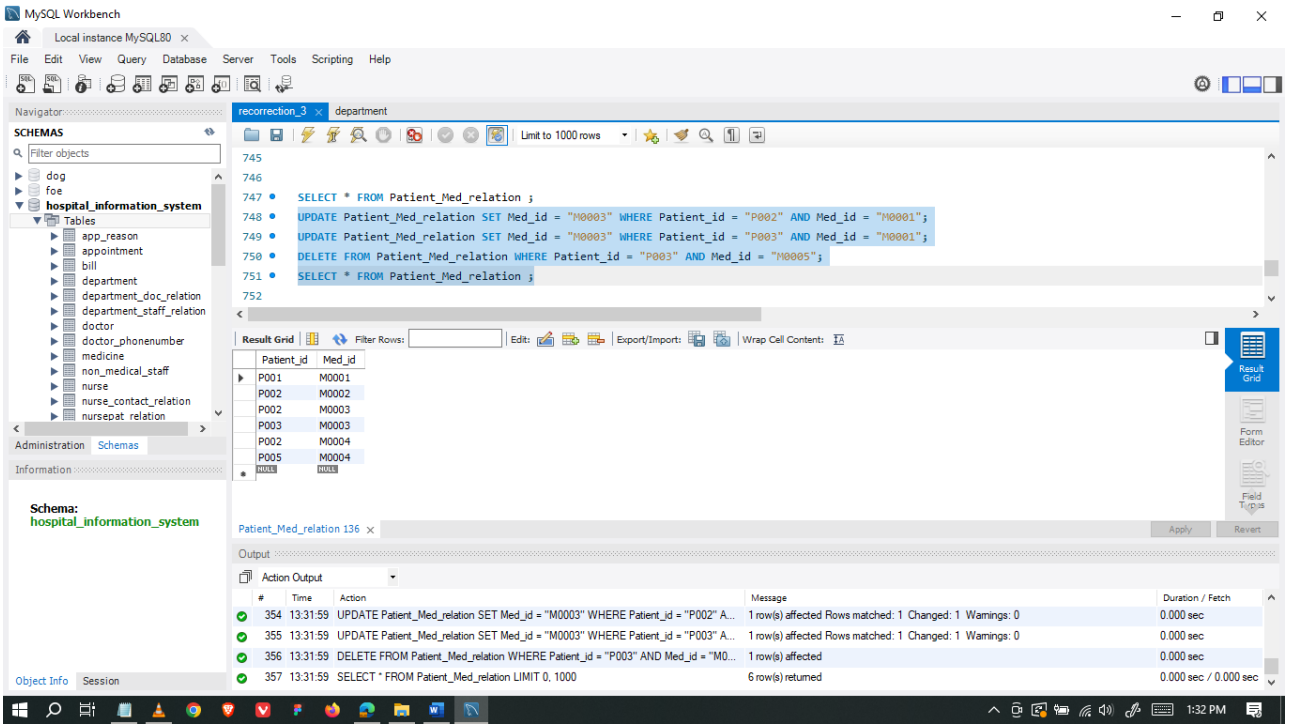
3.5.12 Patient



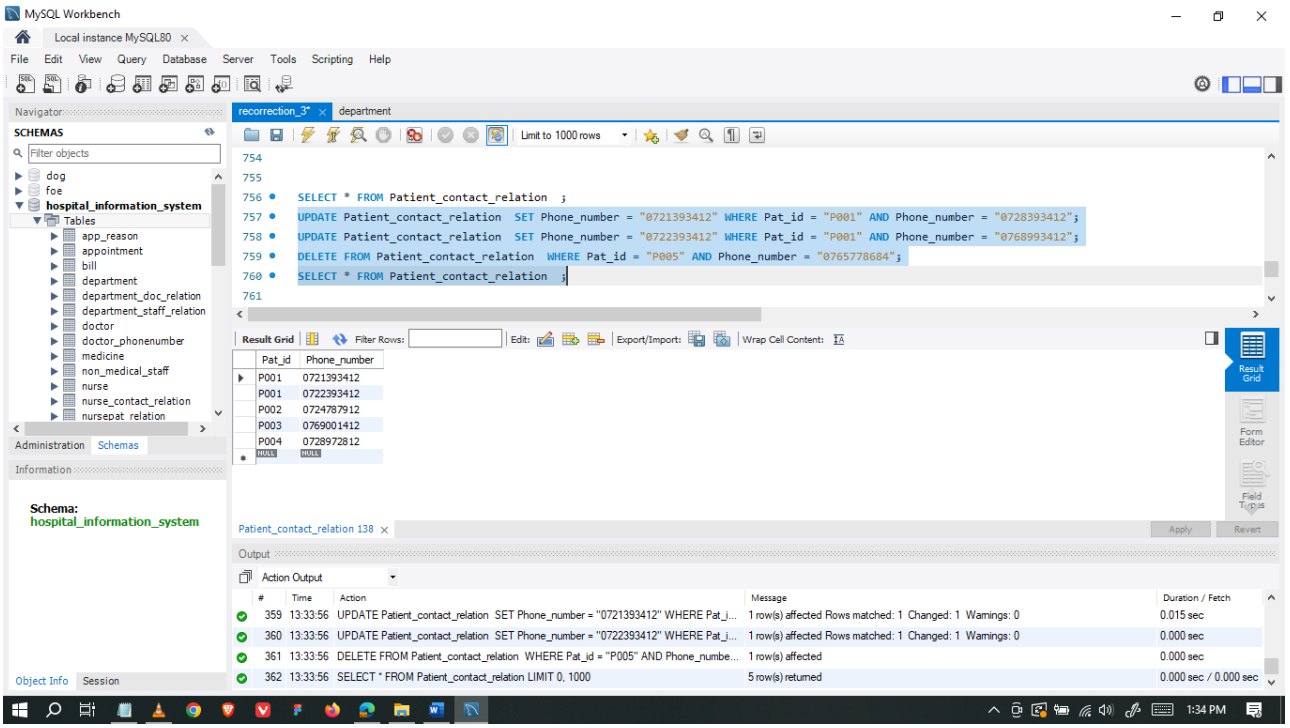
3.5.13 Patient_Doctor_relation



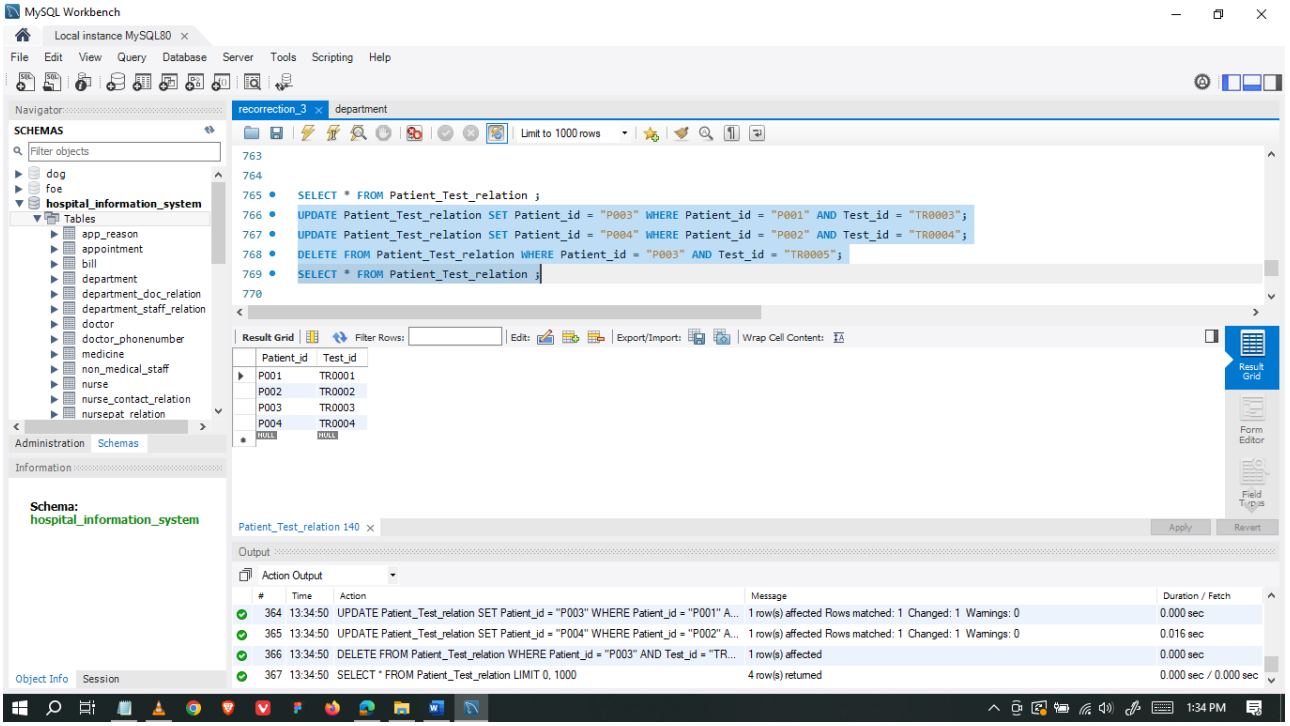
3.5.14 Patient_Med_relation



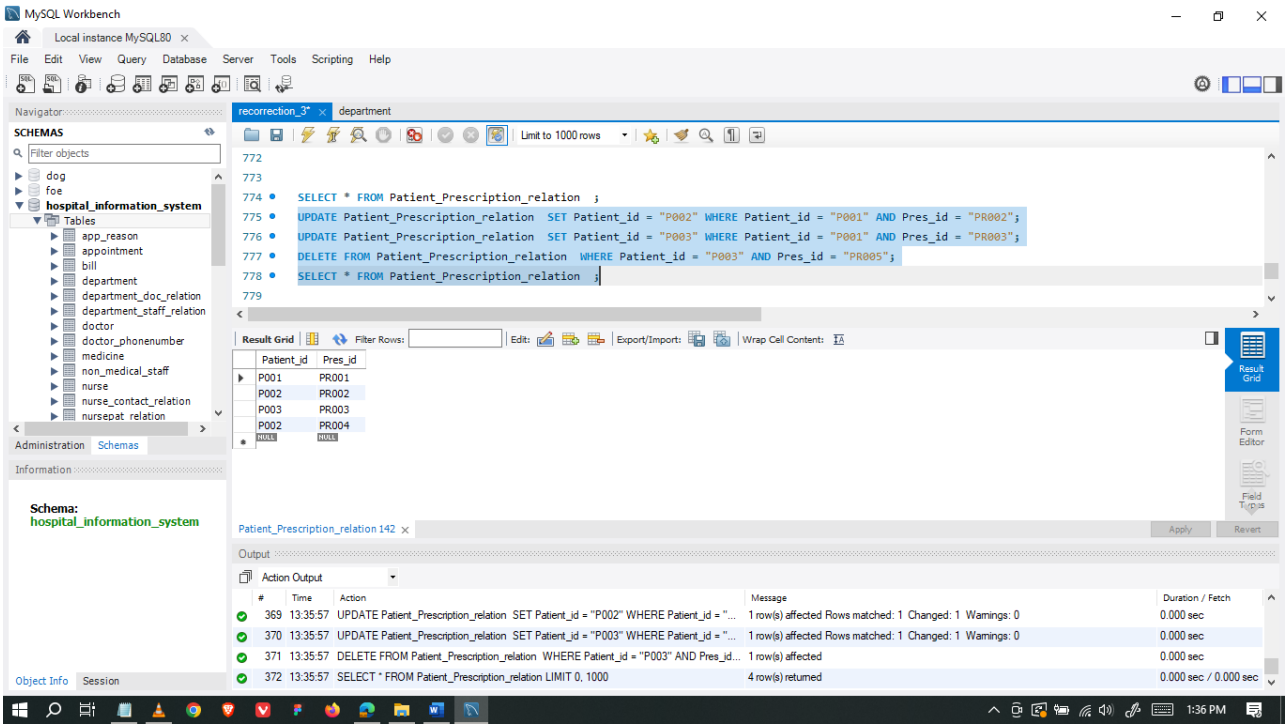
3.5.15 Patient_contact_relation



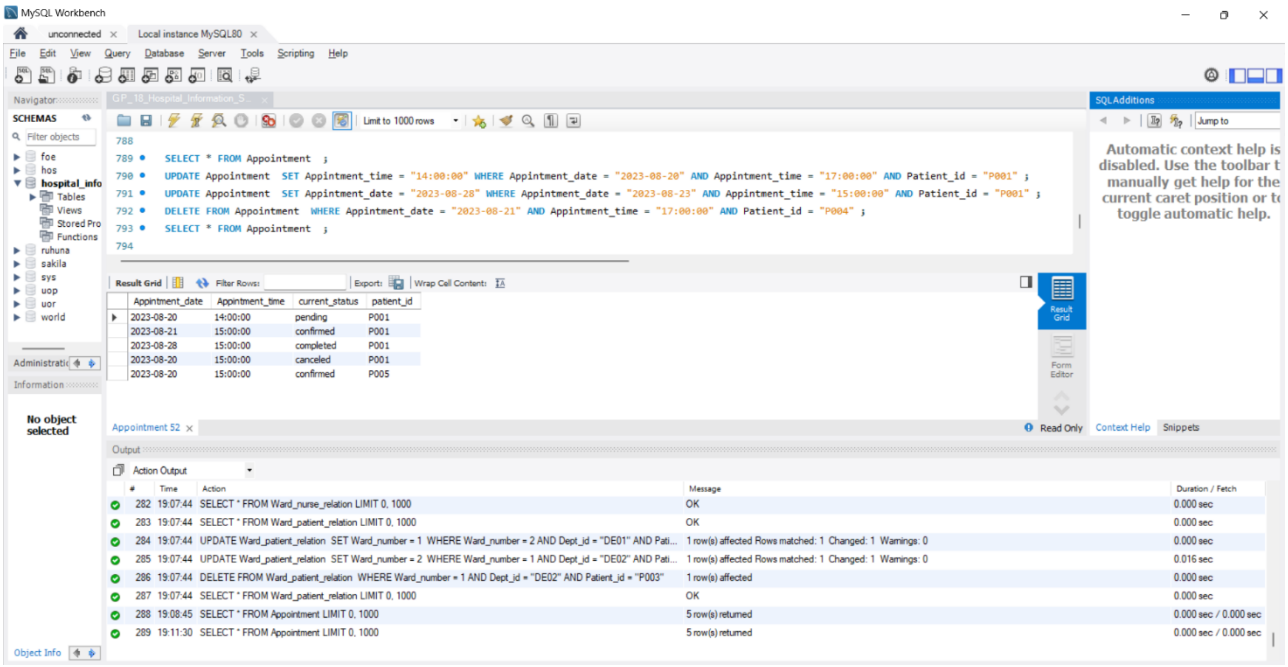
3.5.16 Patient_Test_relation



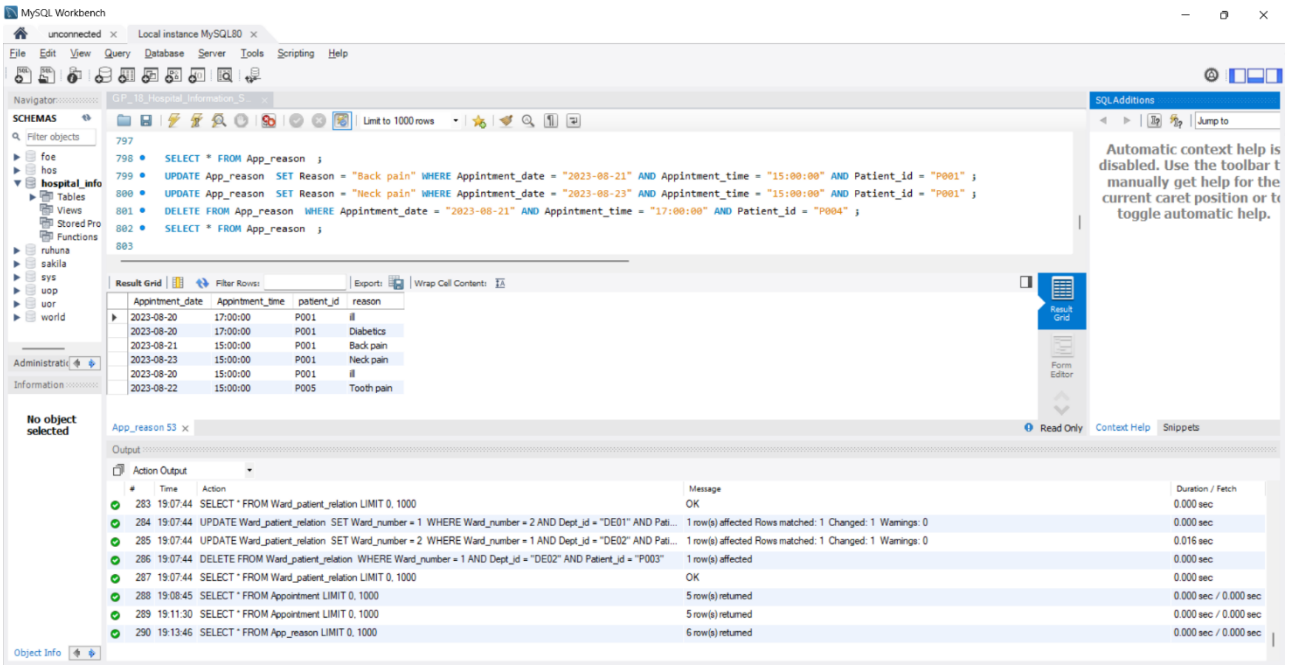
3.5.17 Patient_Prescription_relation



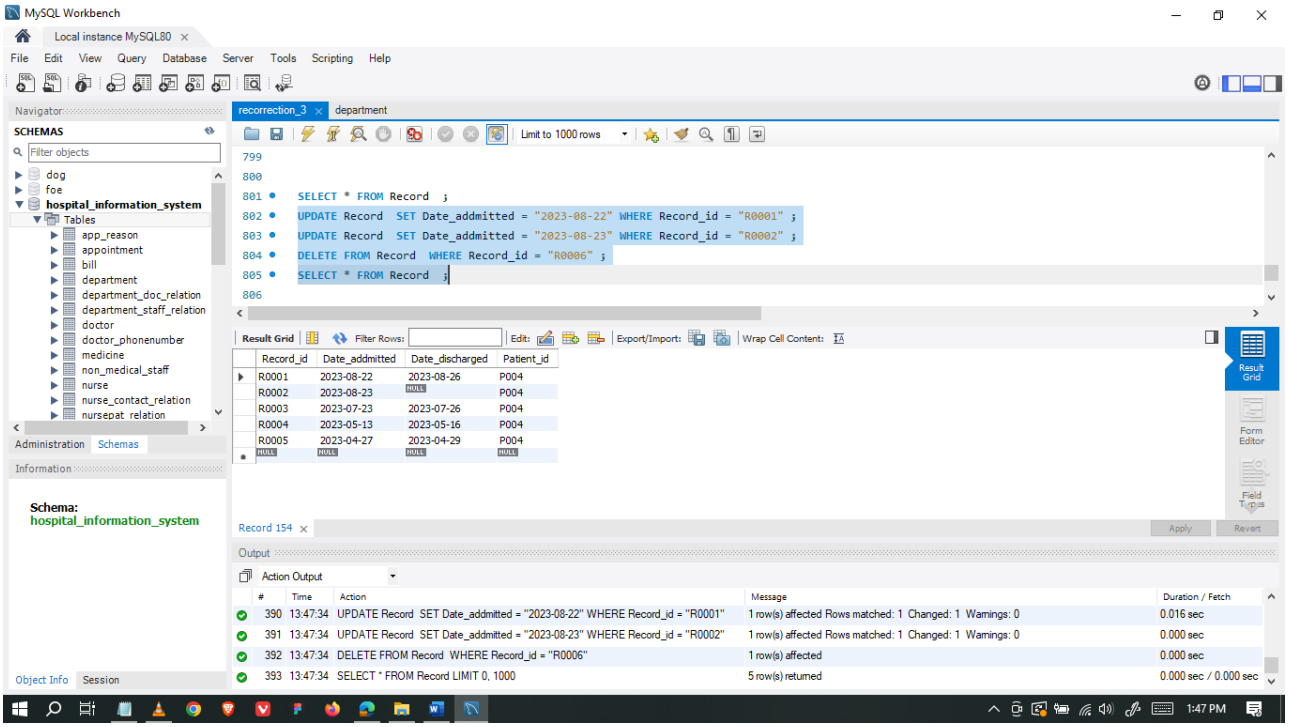
3.5.18 Appointment



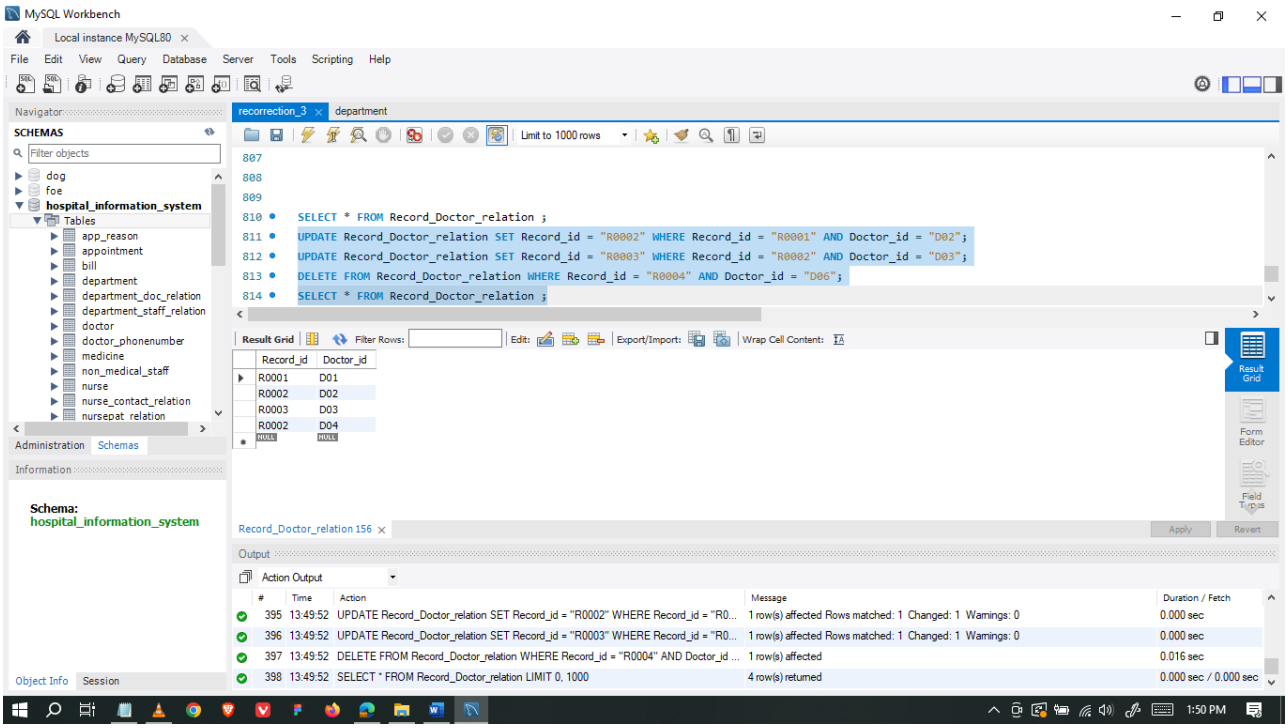
3.5.19 App_reason



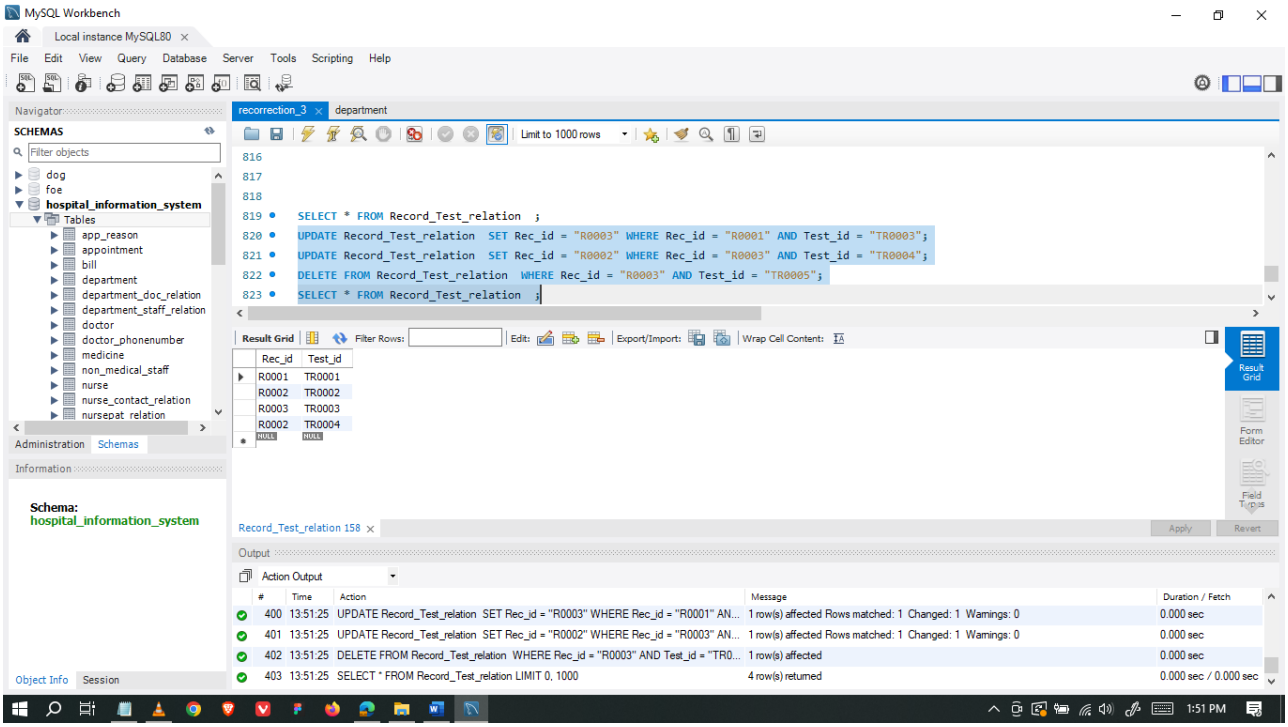
3.5.20 Record



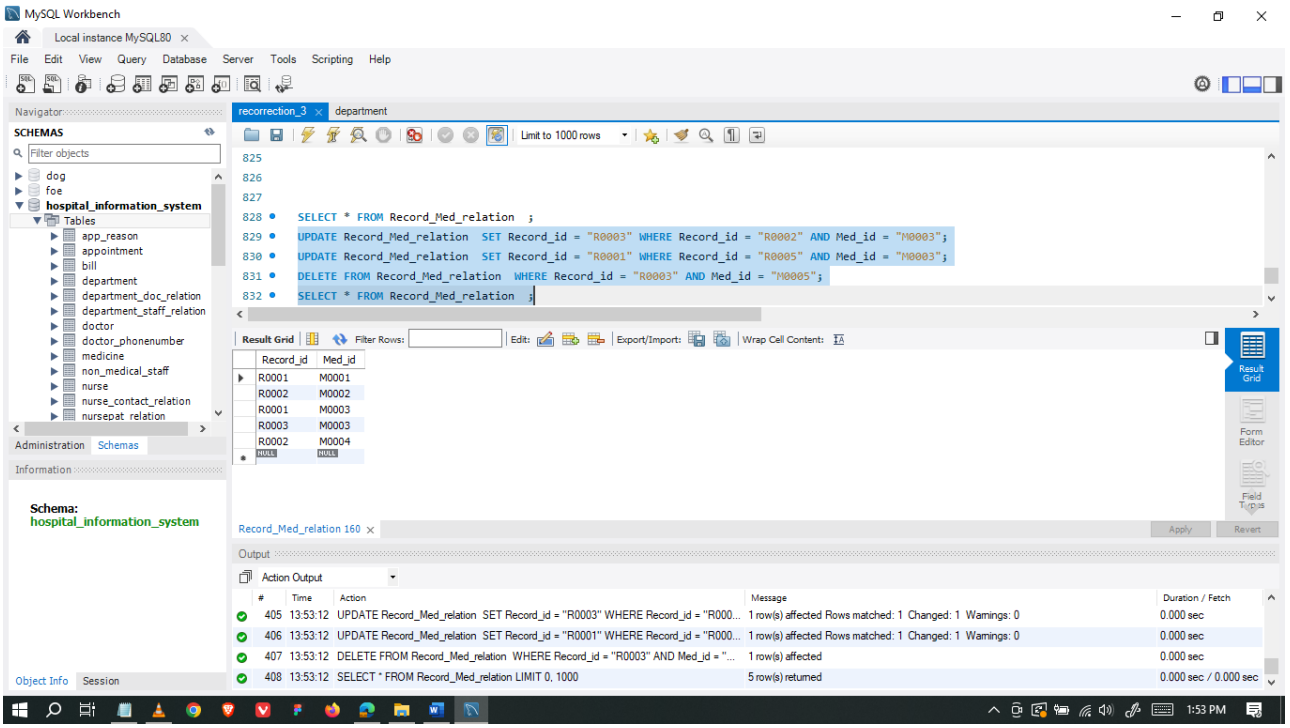
3.5.21 Record_Doctor_relation



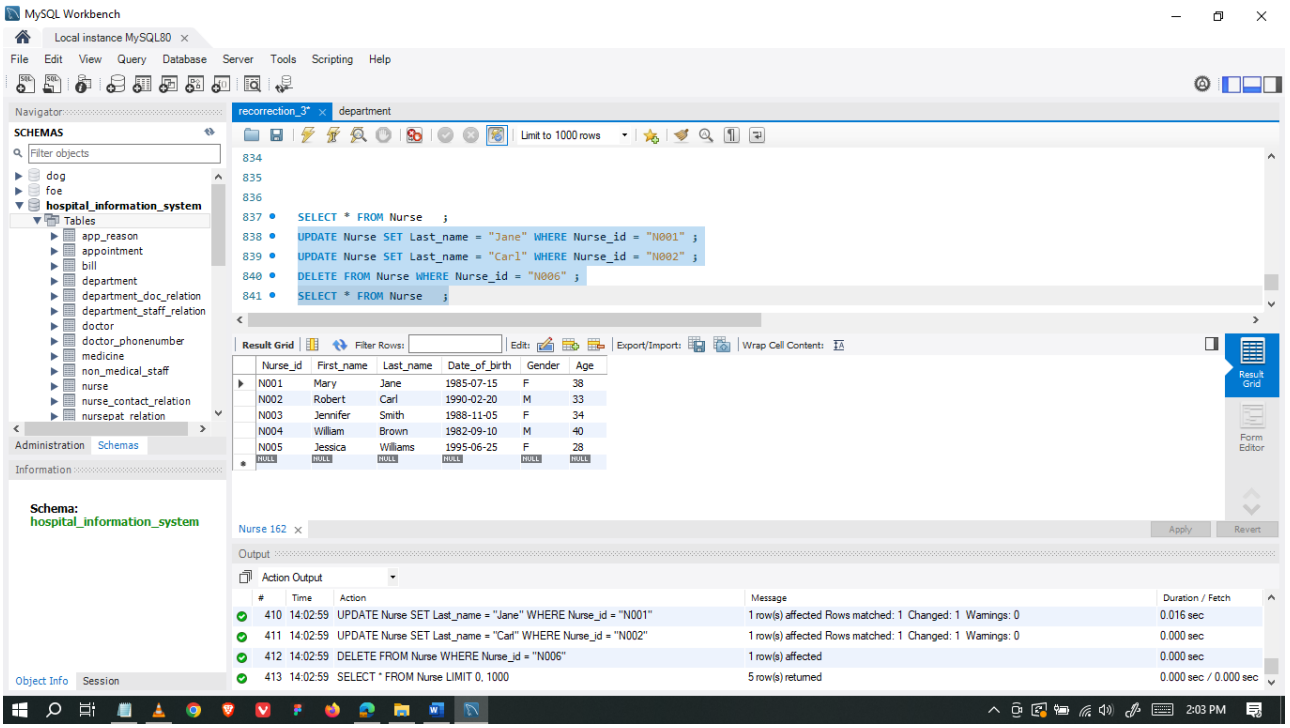
3.5.22 Record_Test_relation



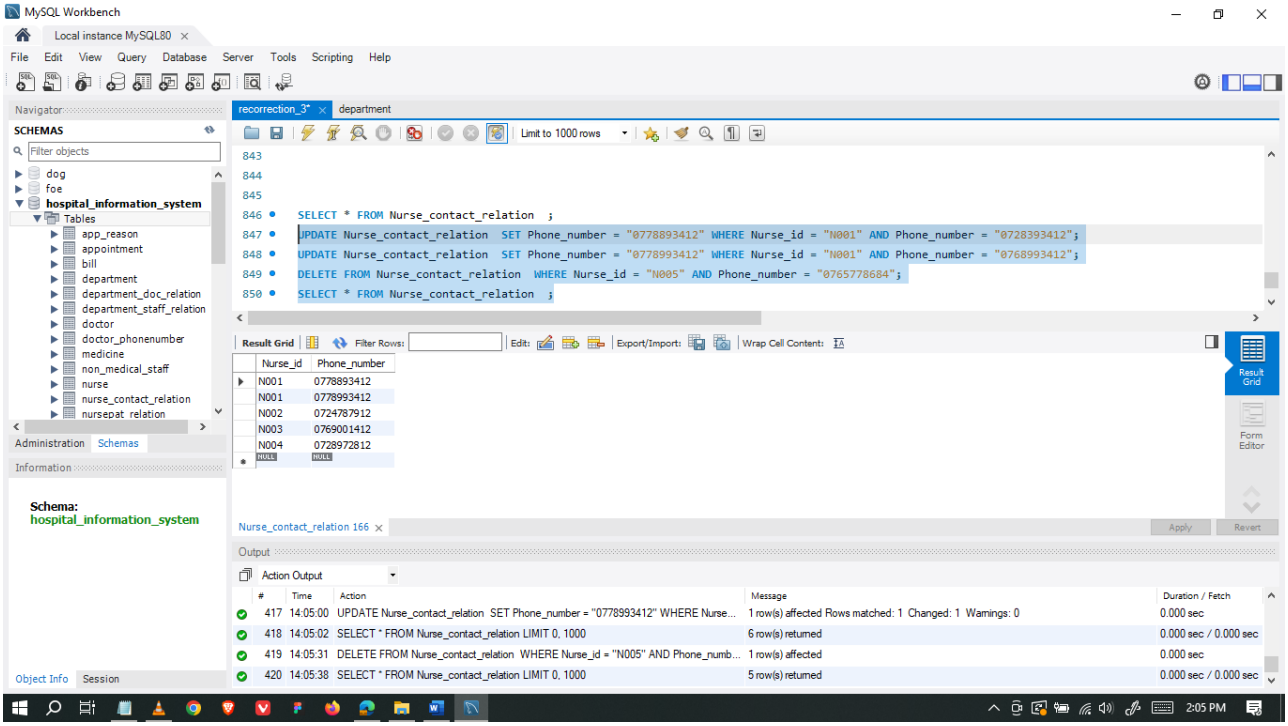
3.5.23 Record_Med_relation



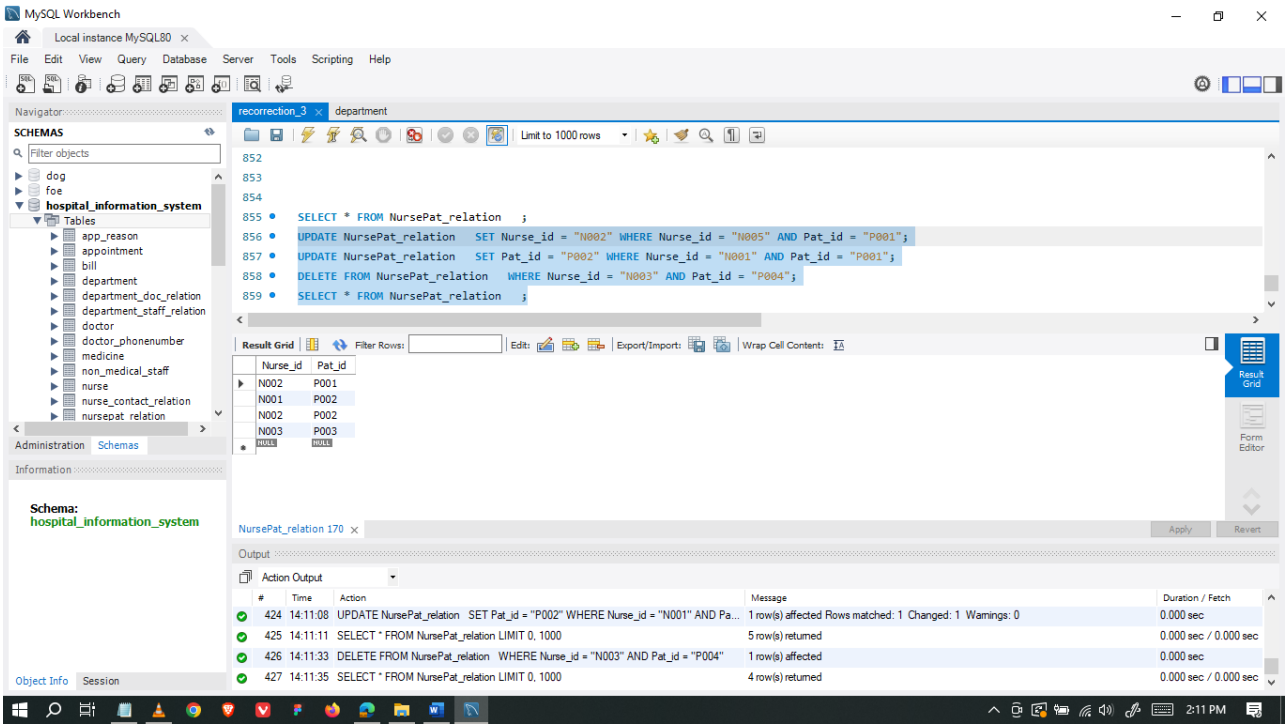
3.5.24 Nurse



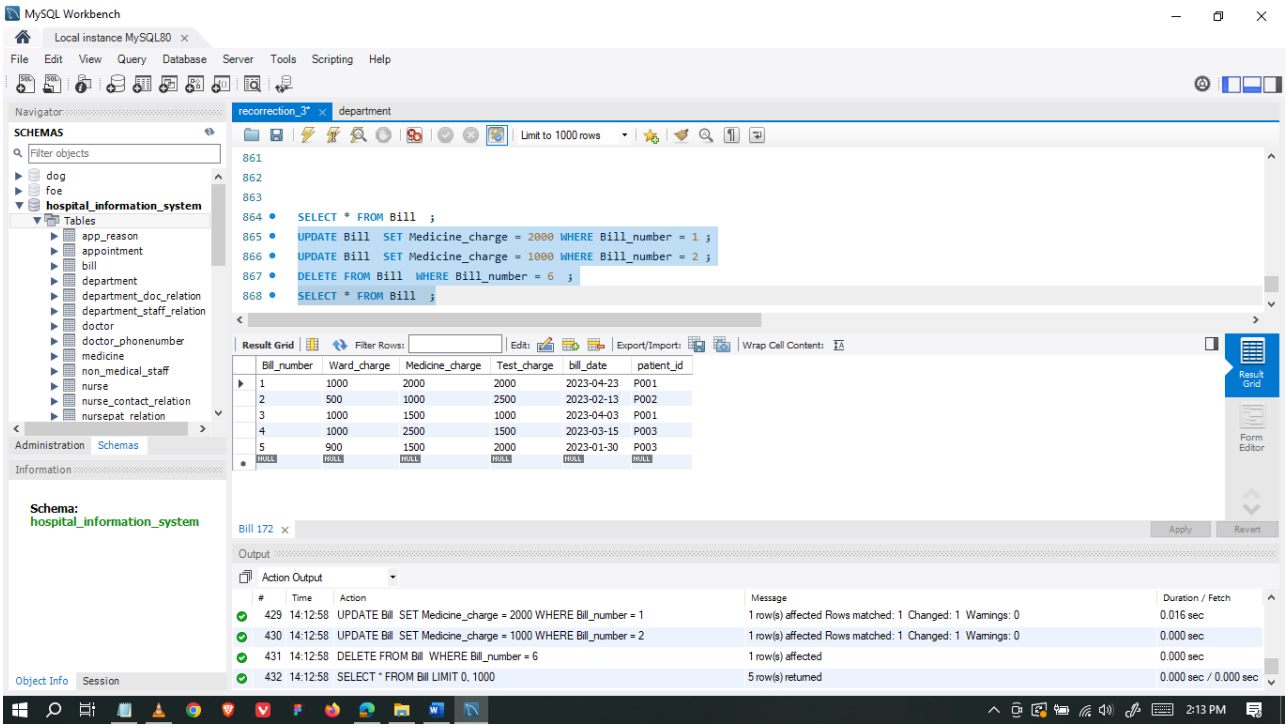
3.5.25 Nurse_contact_relation



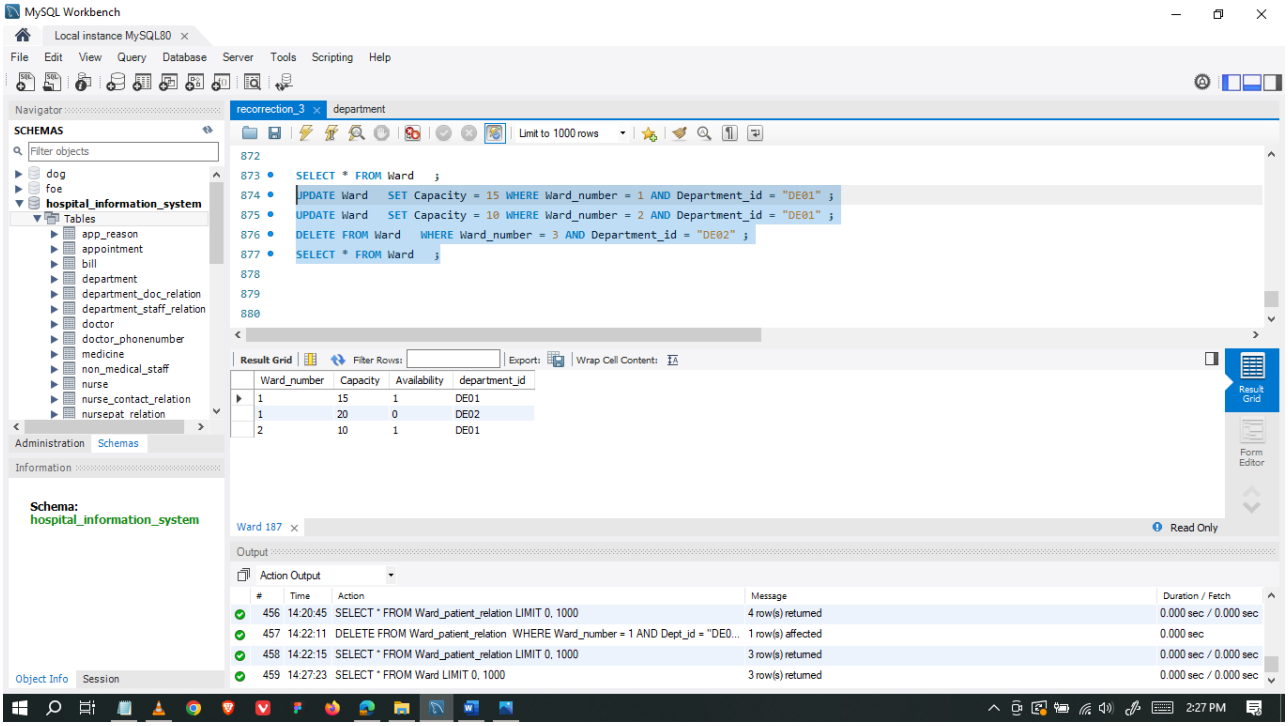
3.5.26 NursePat_relation



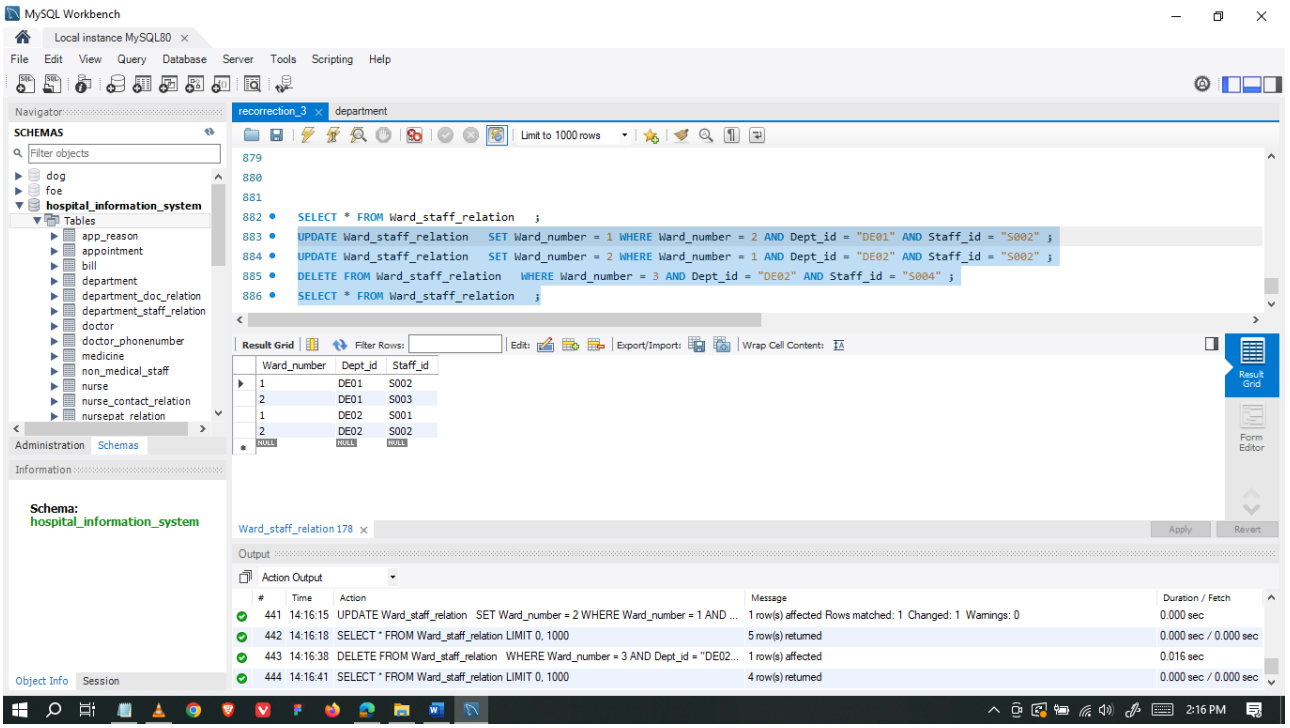
3.5.27 Bill



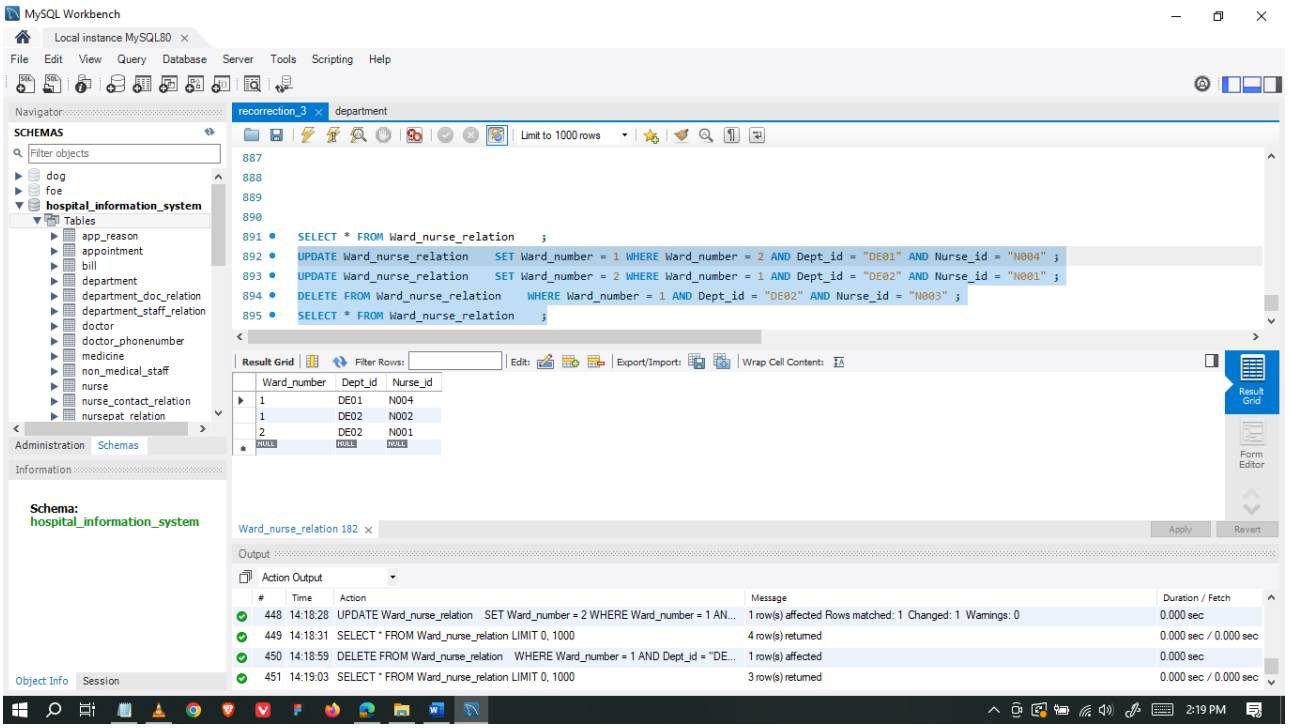
3.5.28 Ward



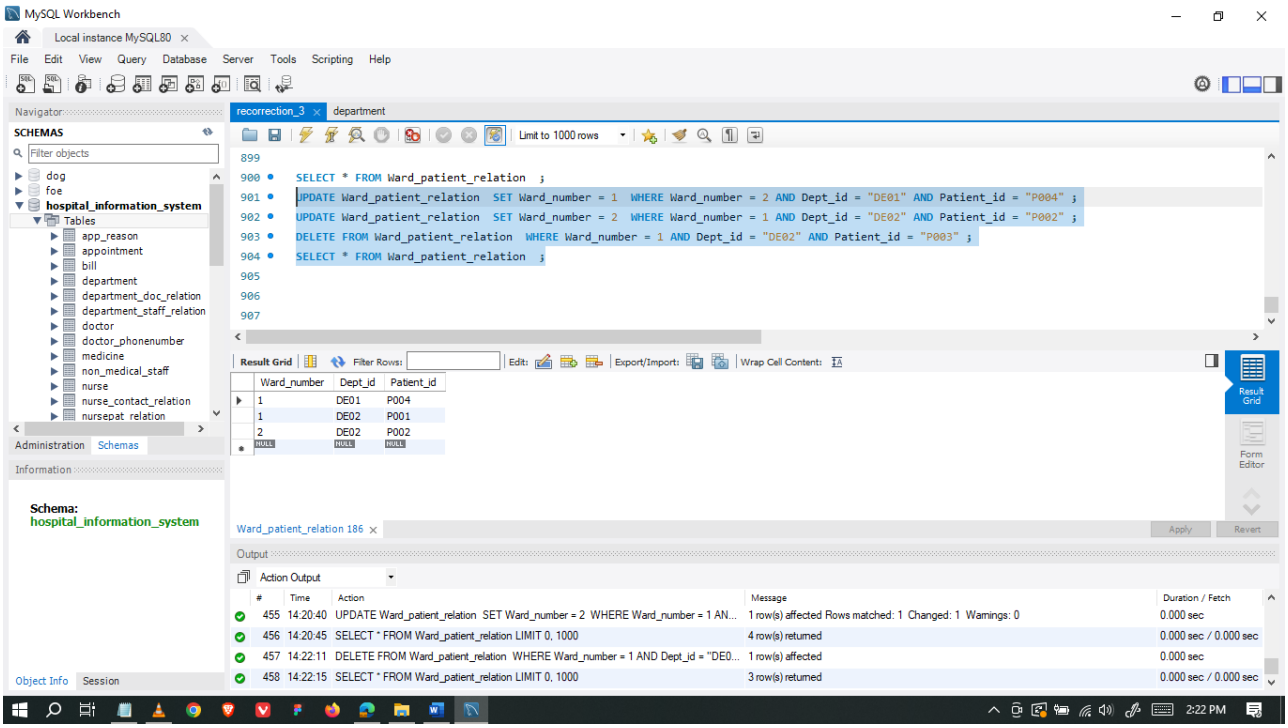
3.5.29 Ward_staff_relation



3.5.30 Ward_nurse_relation



3.5.31 Ward_patient_relation

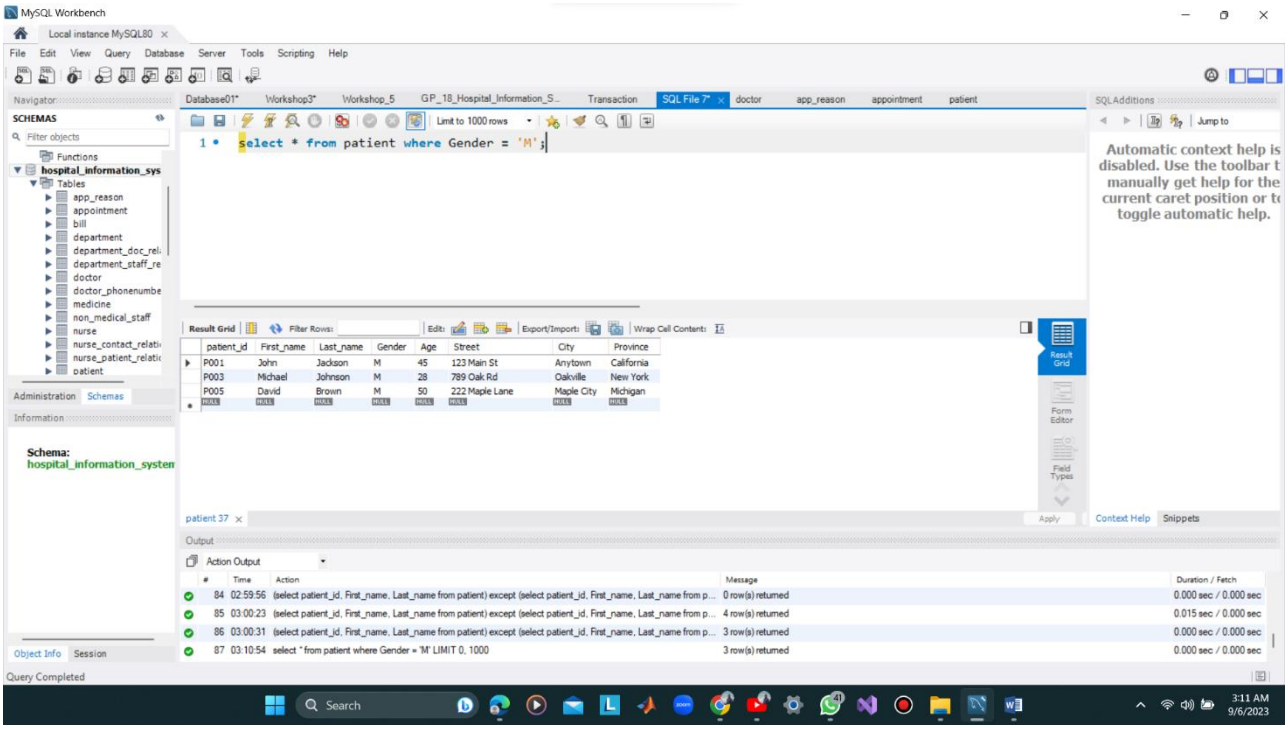


Chapter 4 : Transactions

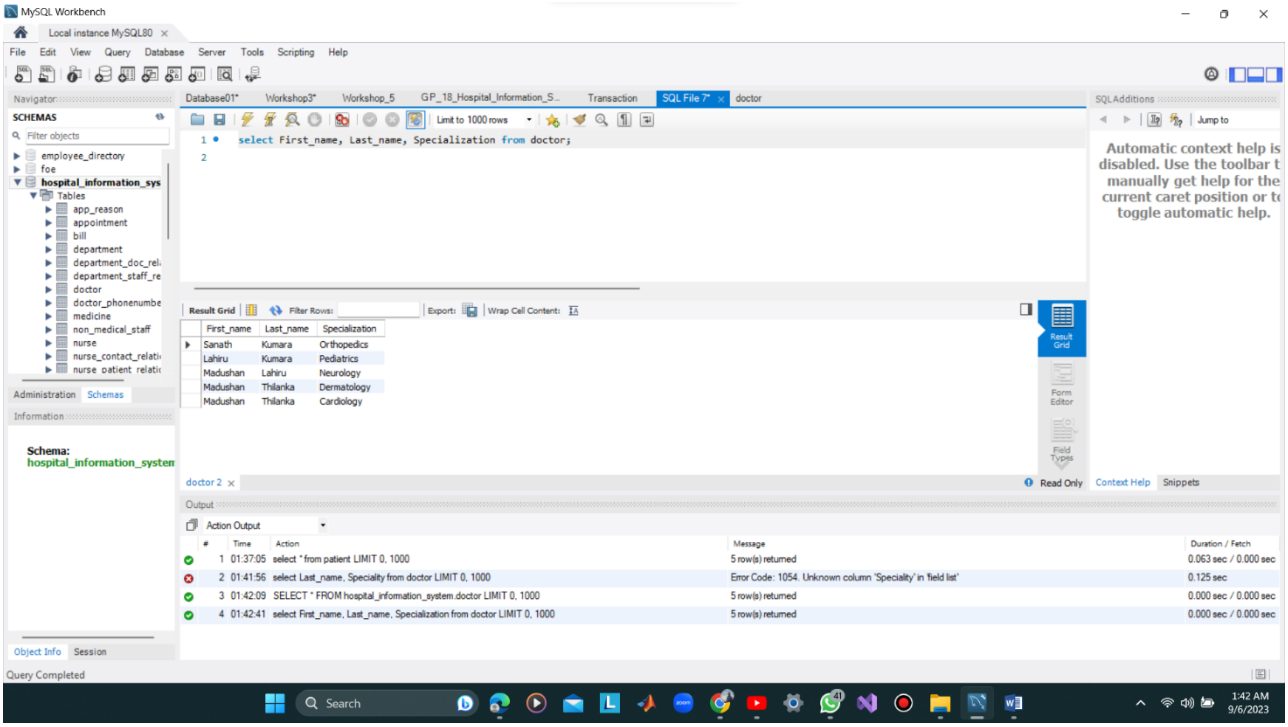
Below shown screenshots consist seven simple queries and thirteen complex queries to retrieve data from the database.

4.1 Simple Queries

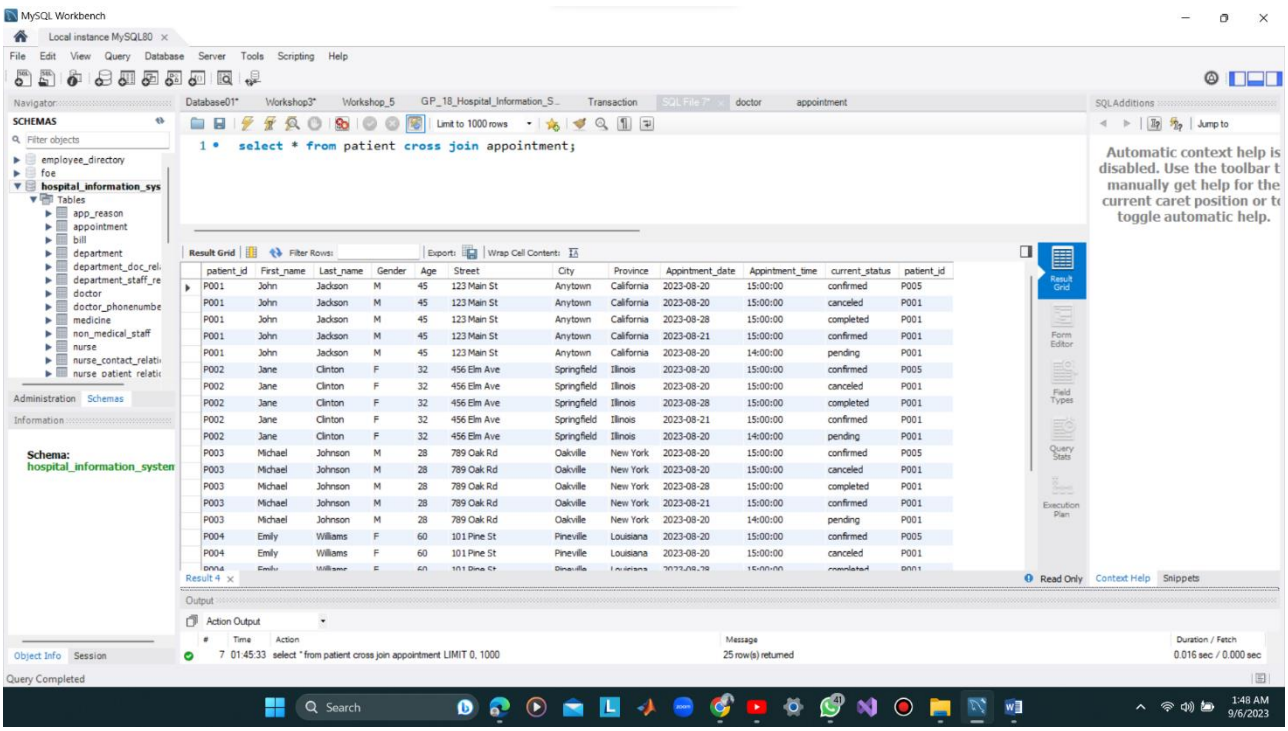
- Select Operation



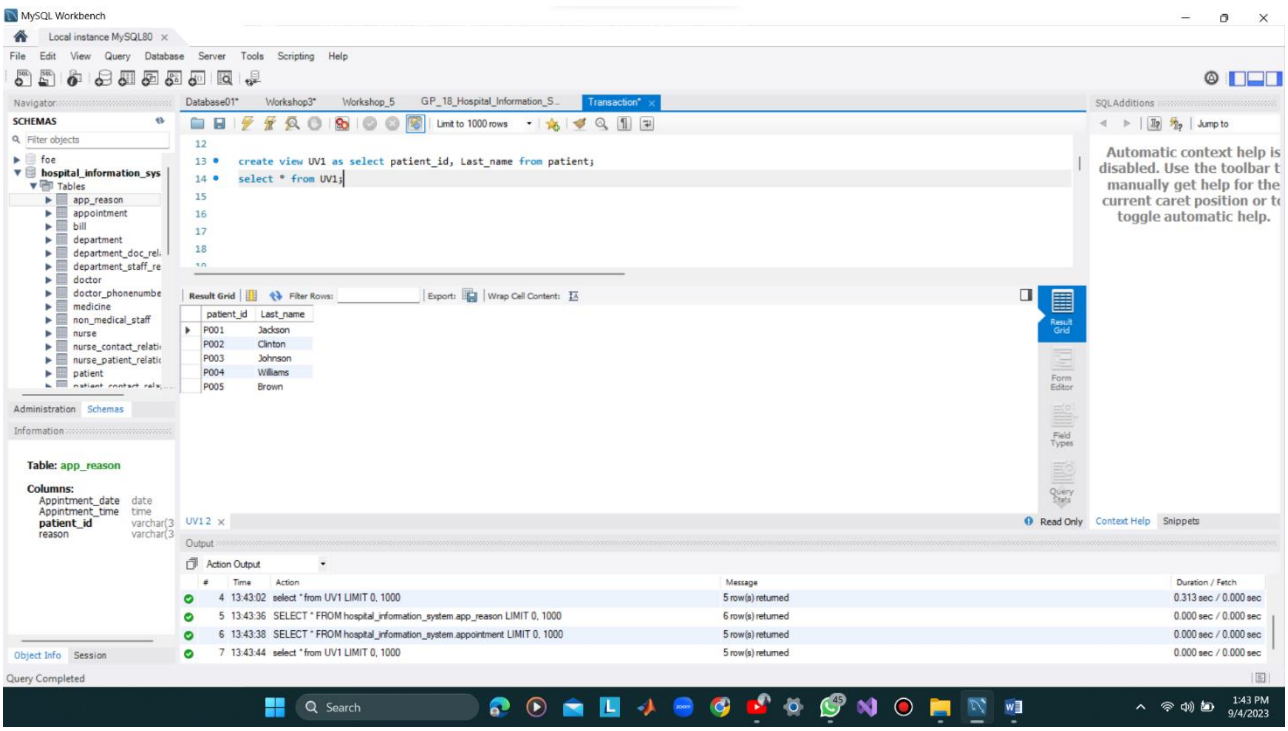
- Project Operation



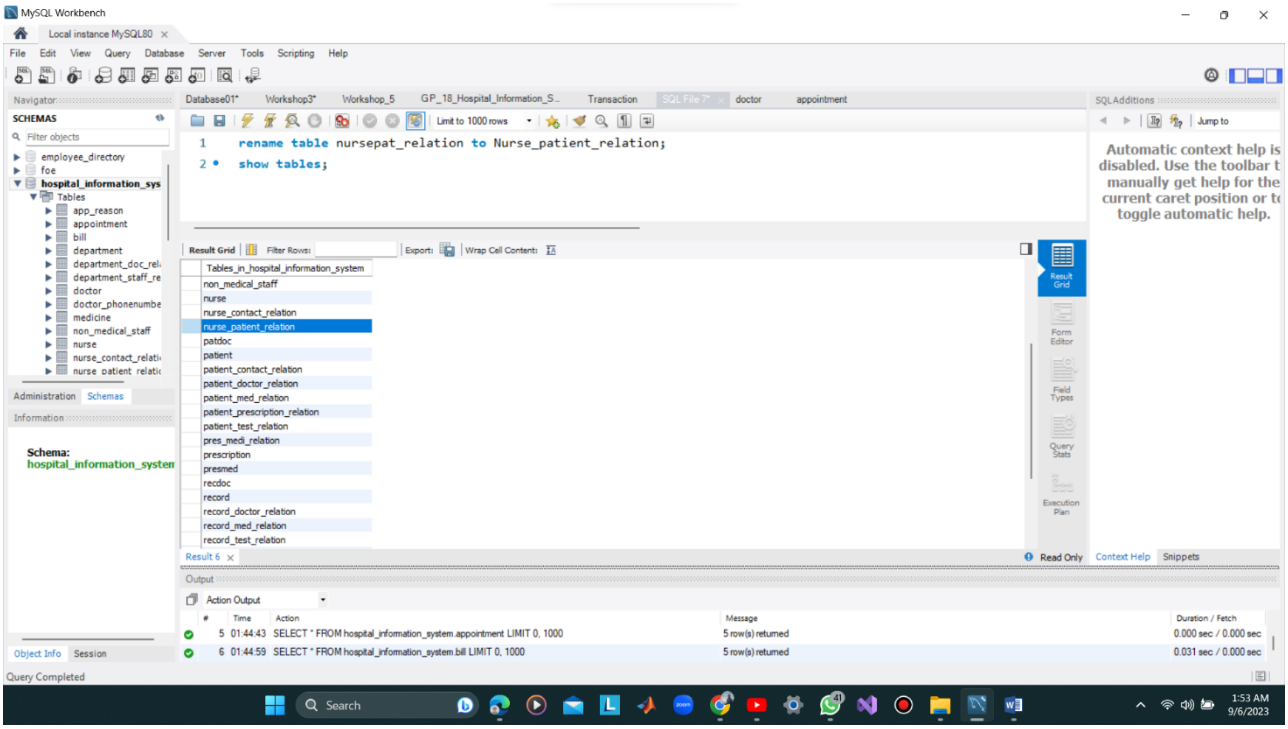
- Cartesian Product Operation



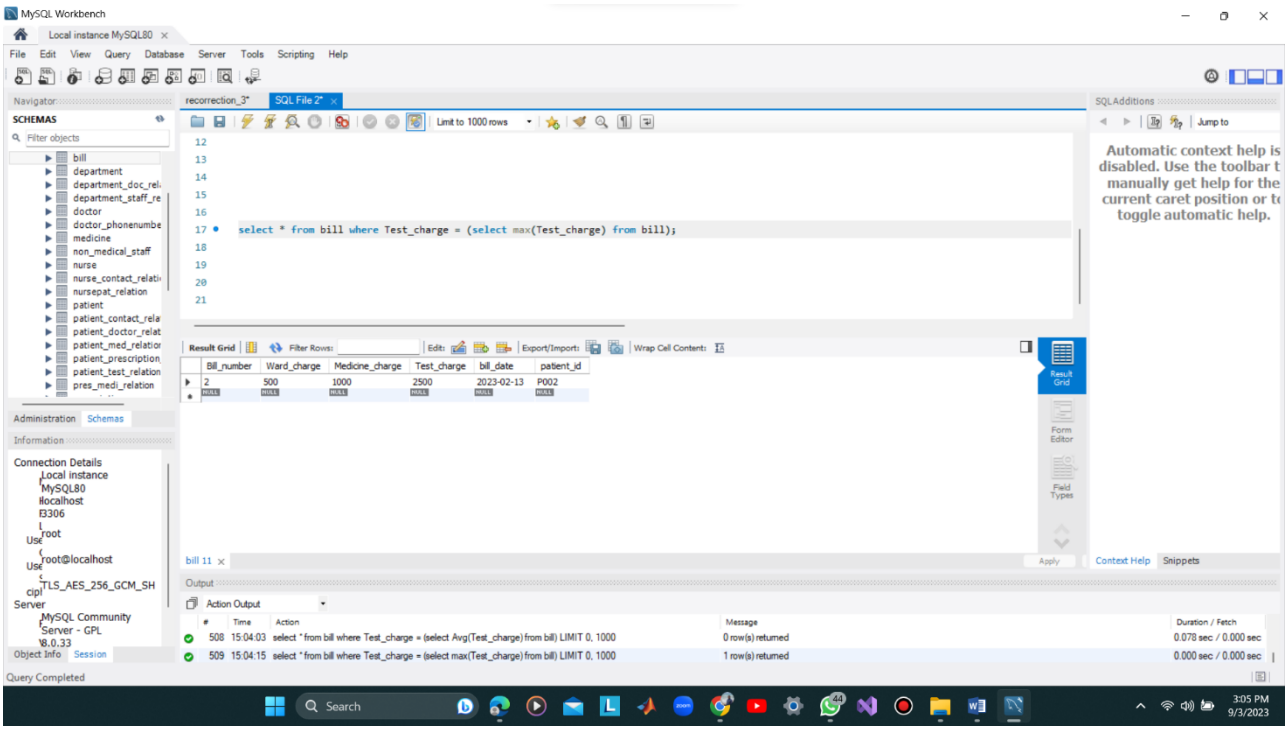
- Create a user view



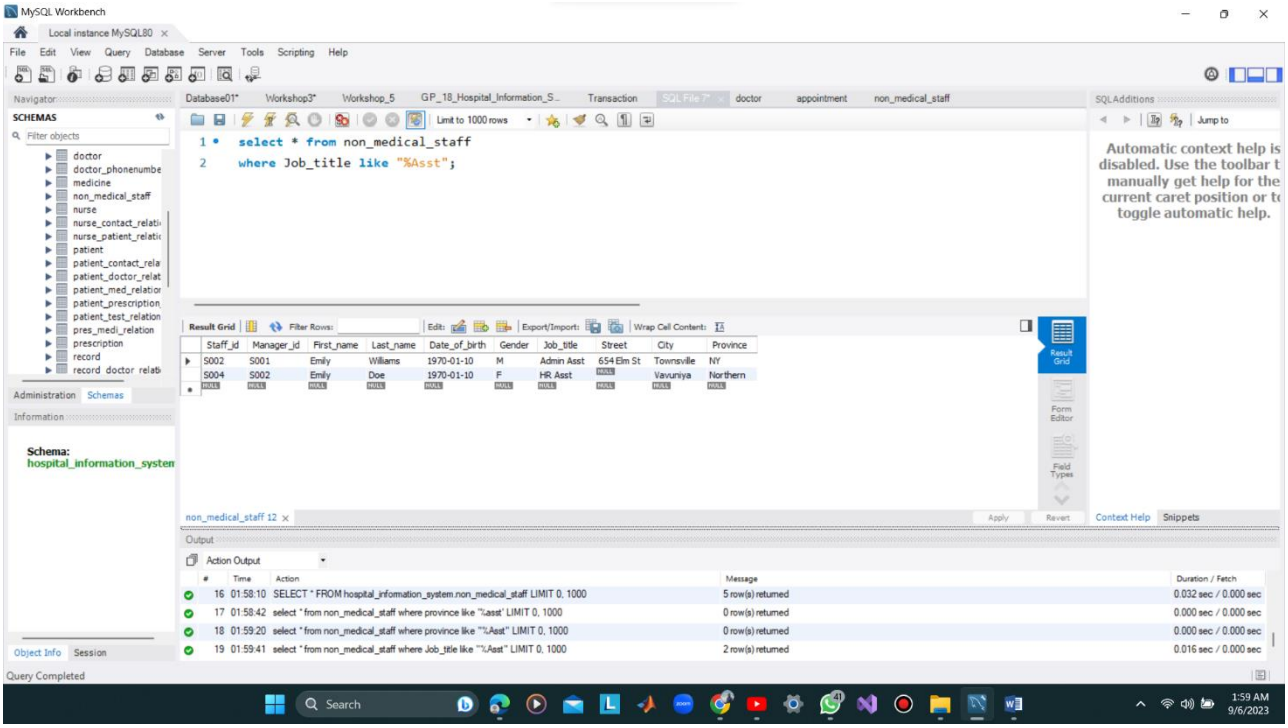
➢ Renaming Operation



➢ Use of an aggregation function



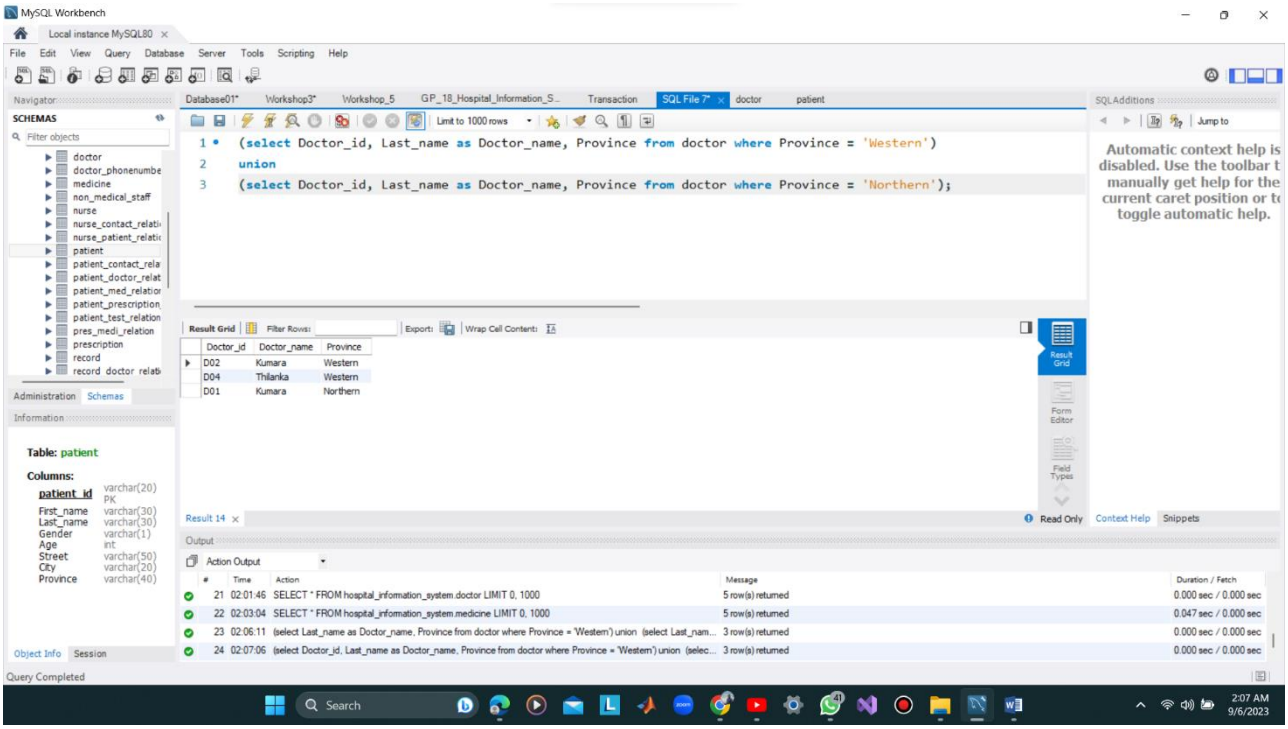
- Use of LIKE keyword



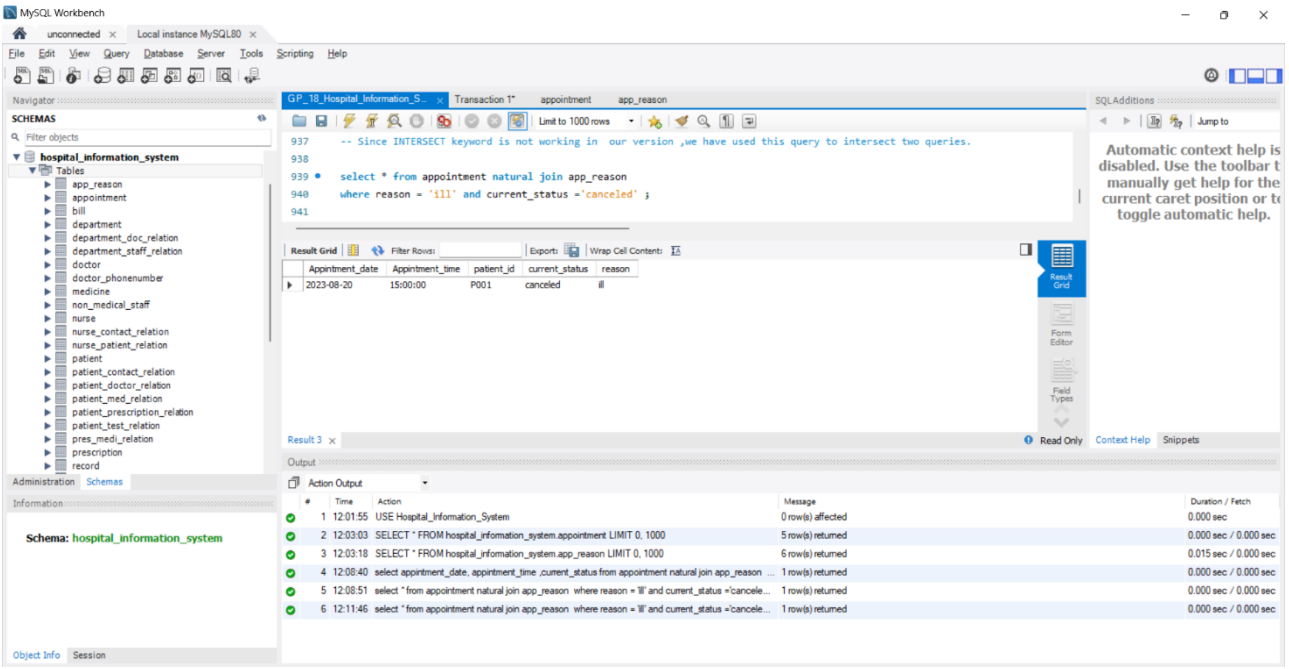
4.2 Complex Queries

4.2.1 Basic Set Operations without User views

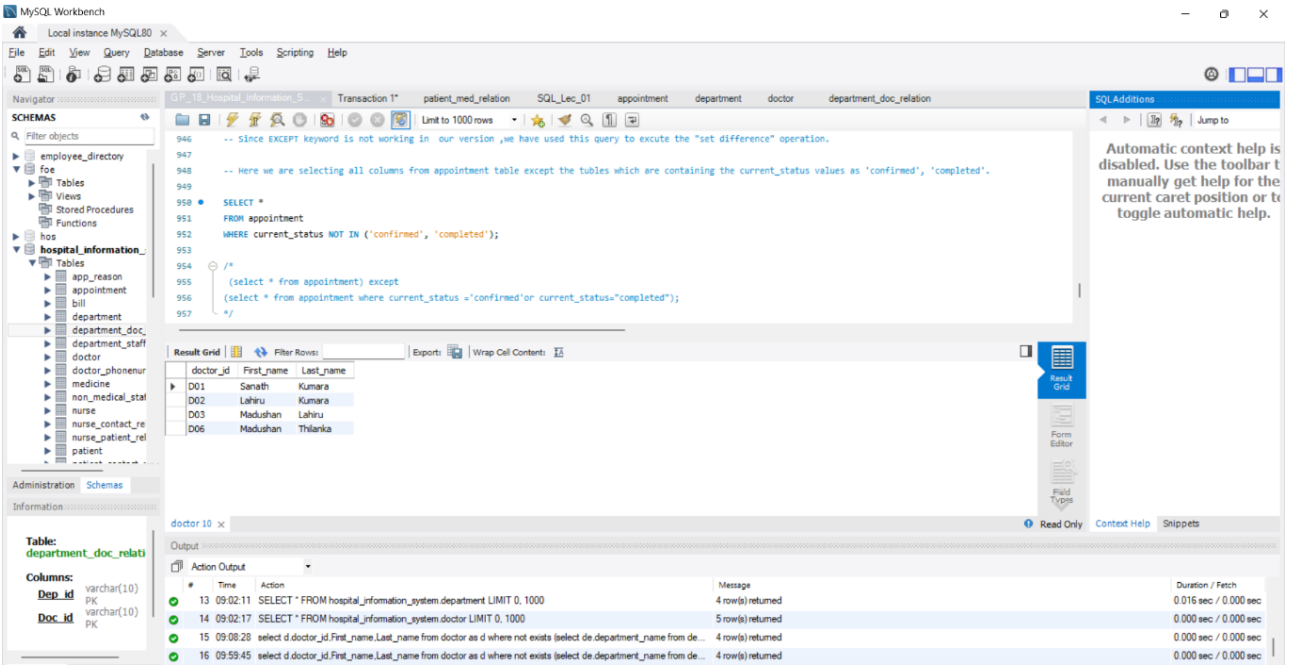
- Union Operation



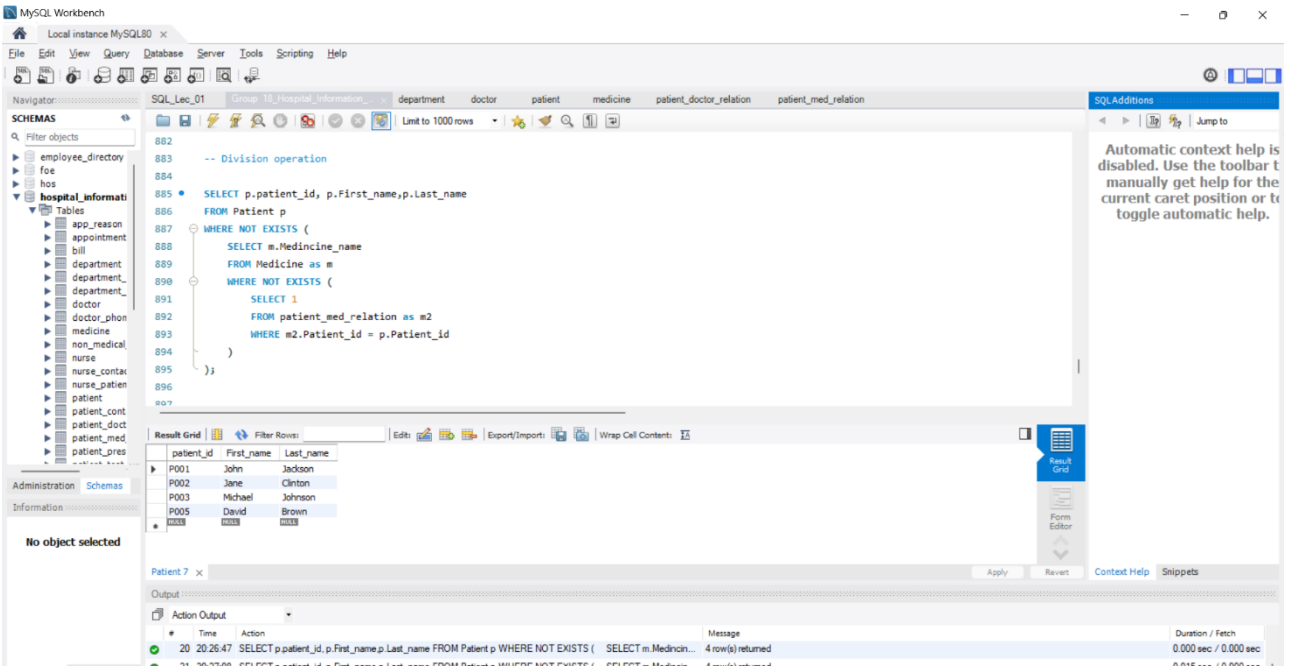
- Intersection Operation



- Set Difference Operation

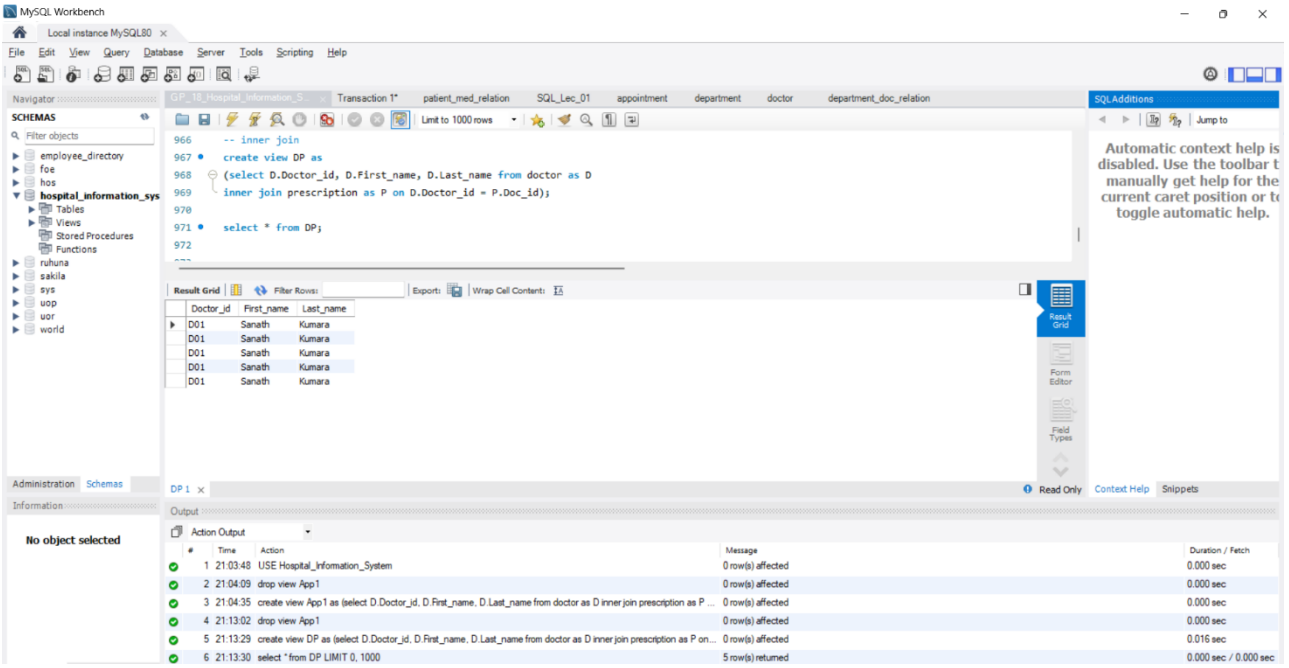


- Division Operation

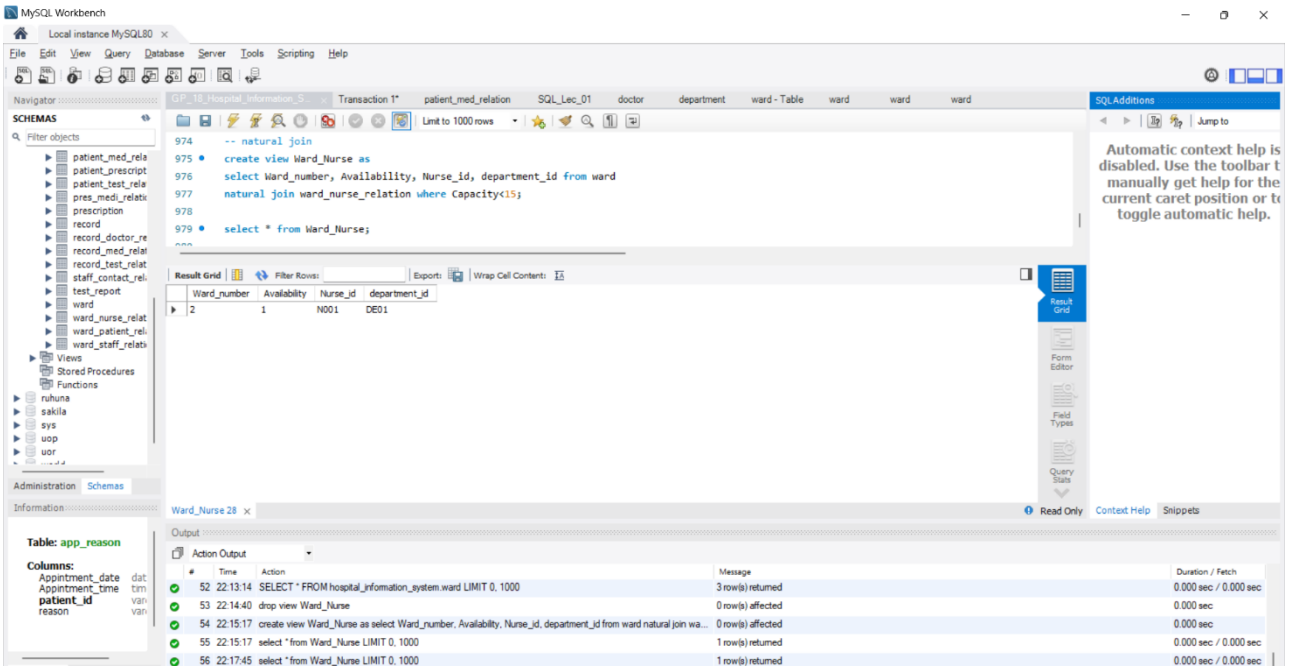


4.2.2 Relational algebraic operations by using user views

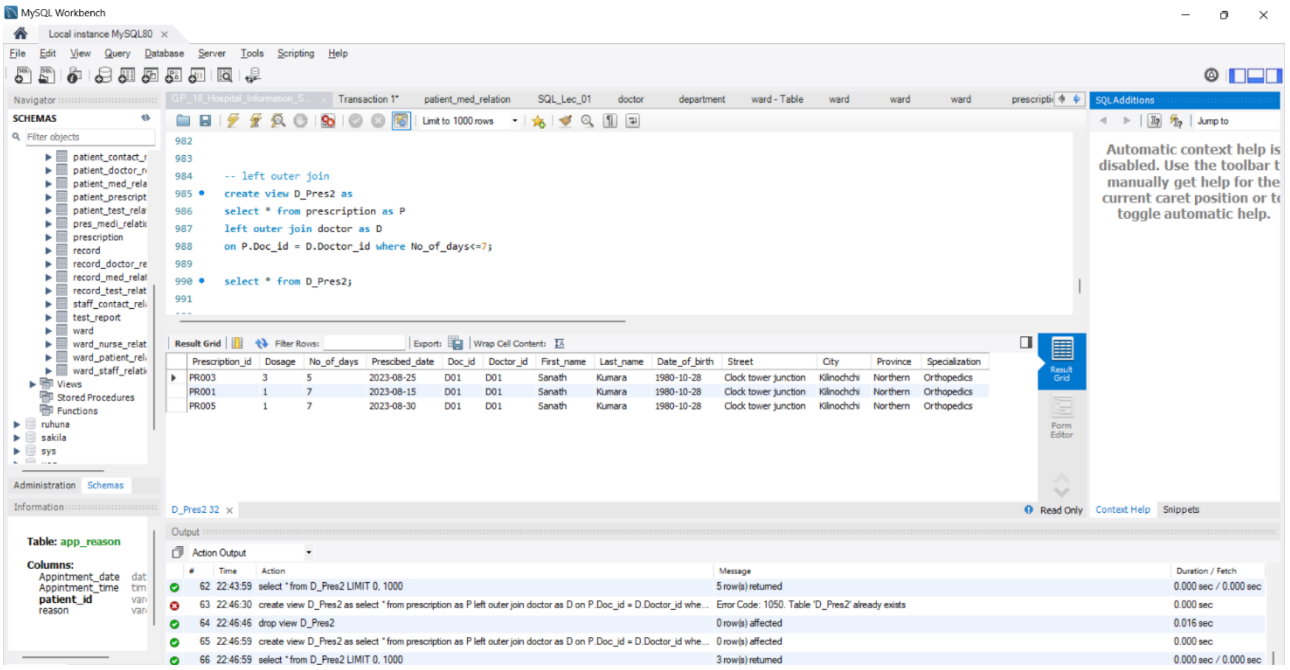
- Inner Join



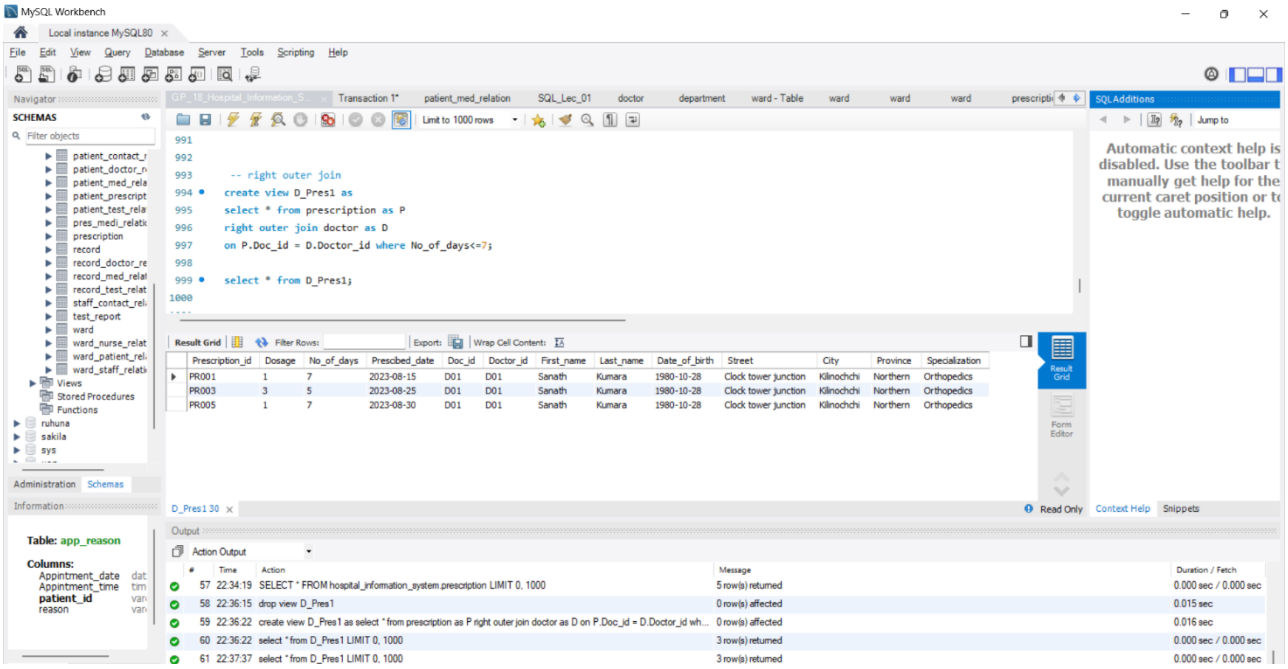
- Natural Join



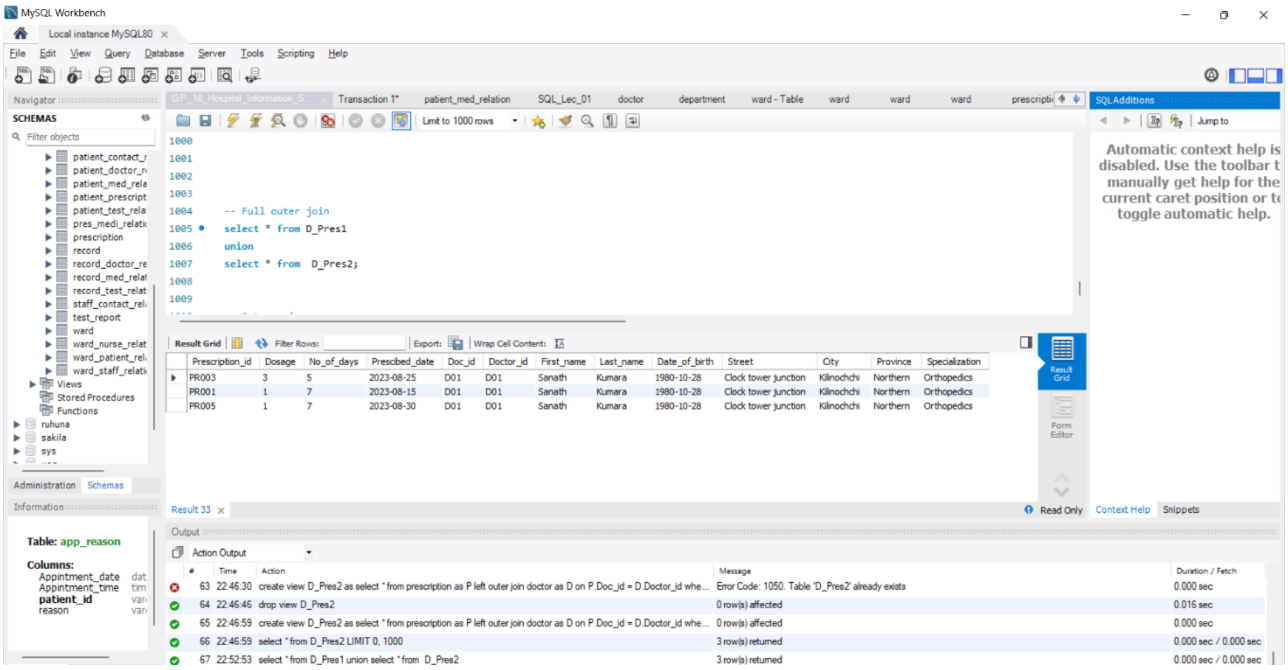
- Left Outer Join



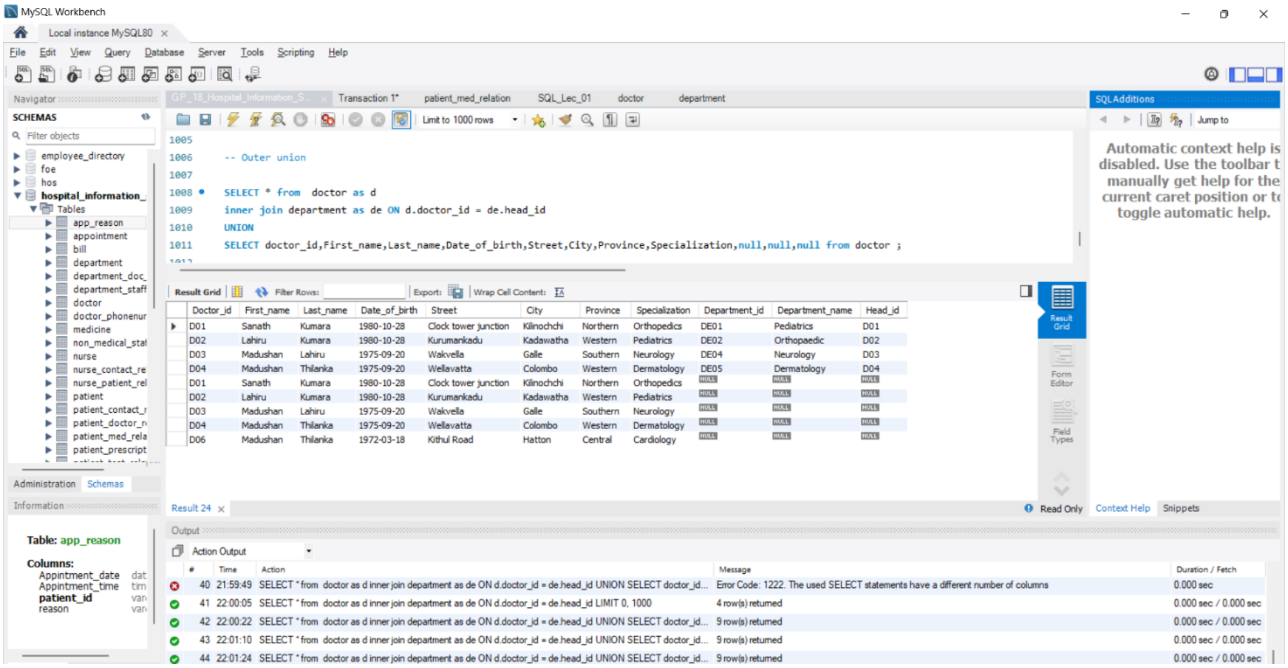
- Right Outer Join



- Full Outer Join

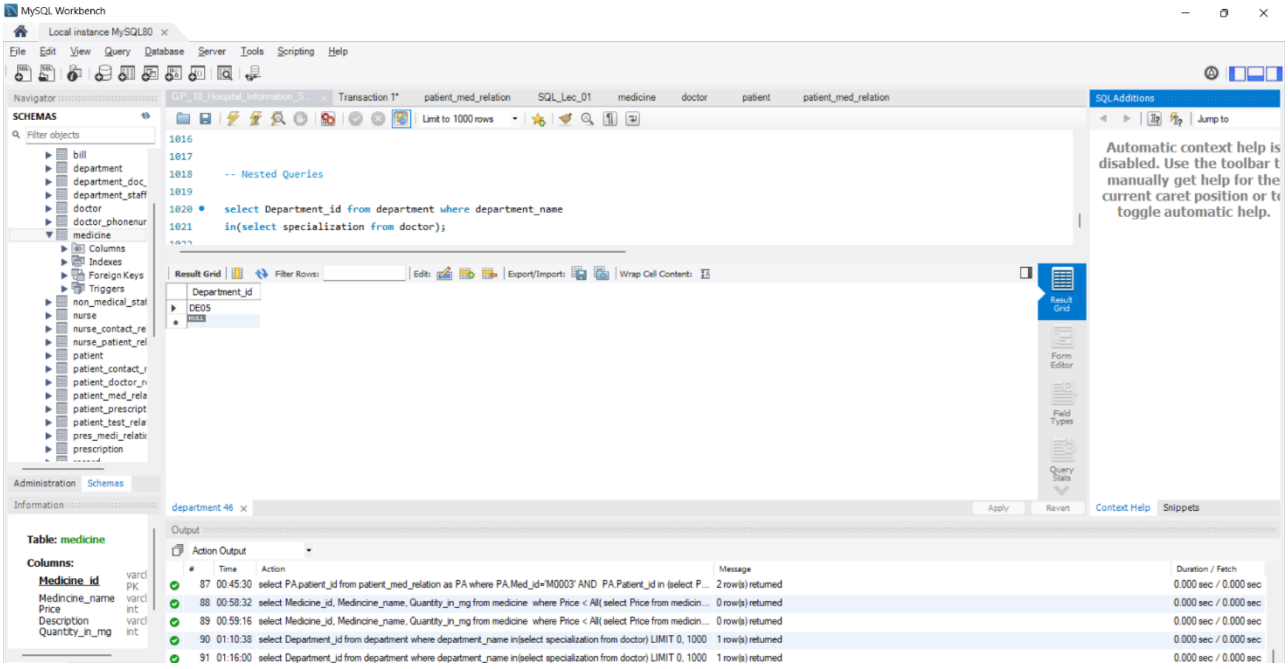


➢ Outer Union

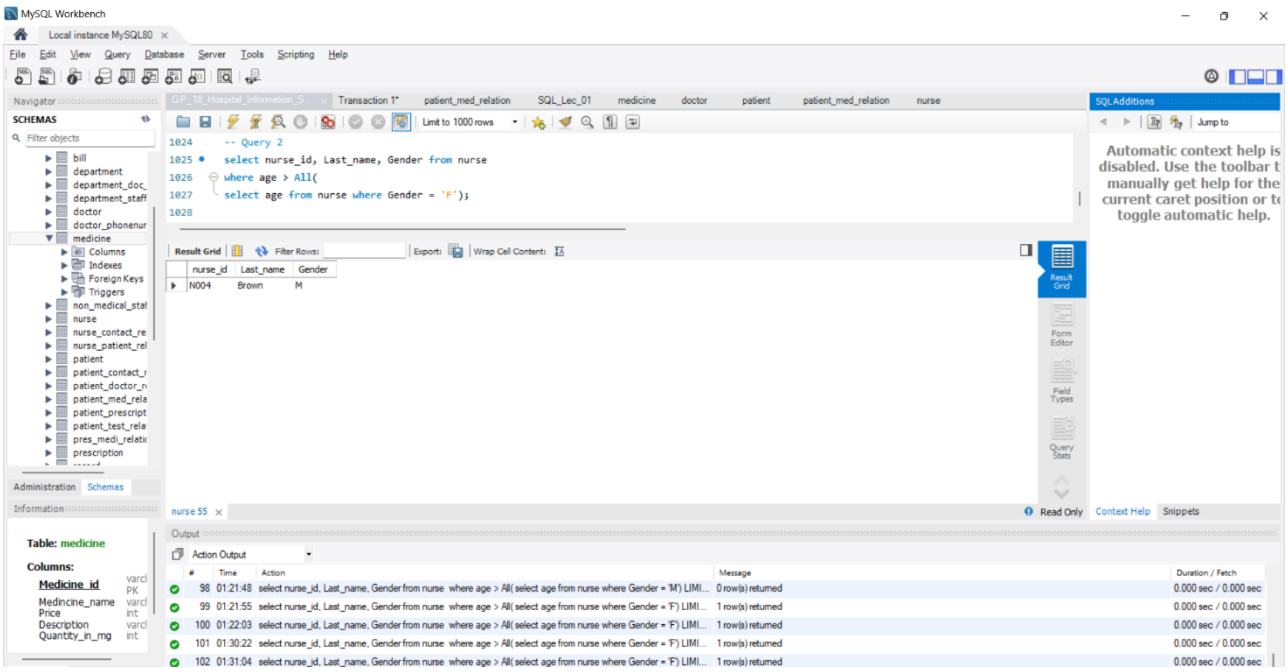


4.2.3 Nested Queries

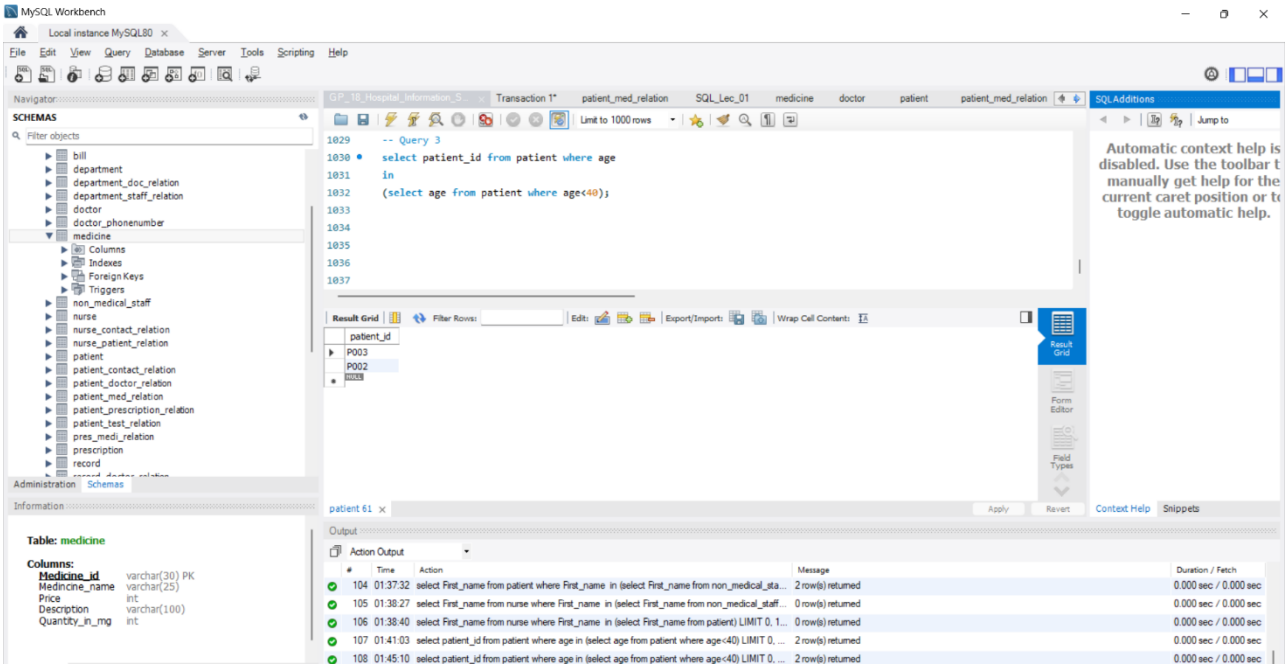
❖ Nested Query 1



❖ Nested Query 2



❖ Nested Query 3



Chapter 5 : Tuning

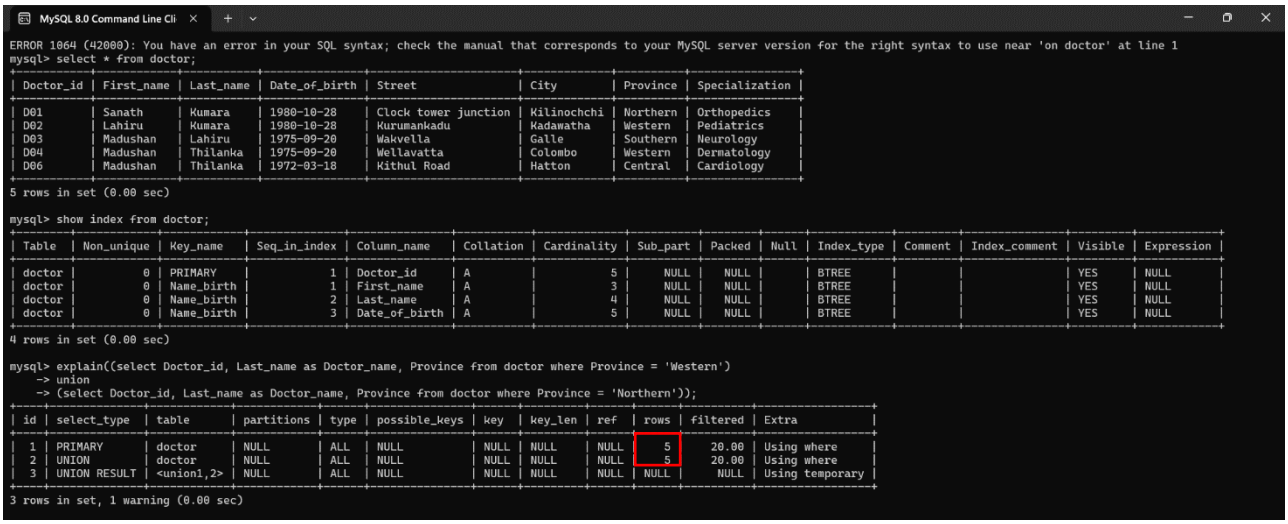
The indexing technique is used to selected complex queries written in Chapter 4. In order to provide a clear understanding, the following steps are taken in query tuning for each complex query.

- ✓ Drop existing indices that are externally created in used tables
- ✓ Show indexes of table before creating suitable index (using SHOW INDEX command) to know about the existing indices.
- ✓ Show number of accessed rows before creating suitable index (using EXPLAIN command) when we are retrieving some particular data.
- ✓ Create suitable index in order to tune the query .
- ✓ Show indexes of table after creating suitable index (using SHOW INDEX command) to ensure that the index was created.
- ✓ Show the number of accessed rows after creating suitable index (using EXPLAIN command) when we are retrieving some particular data.

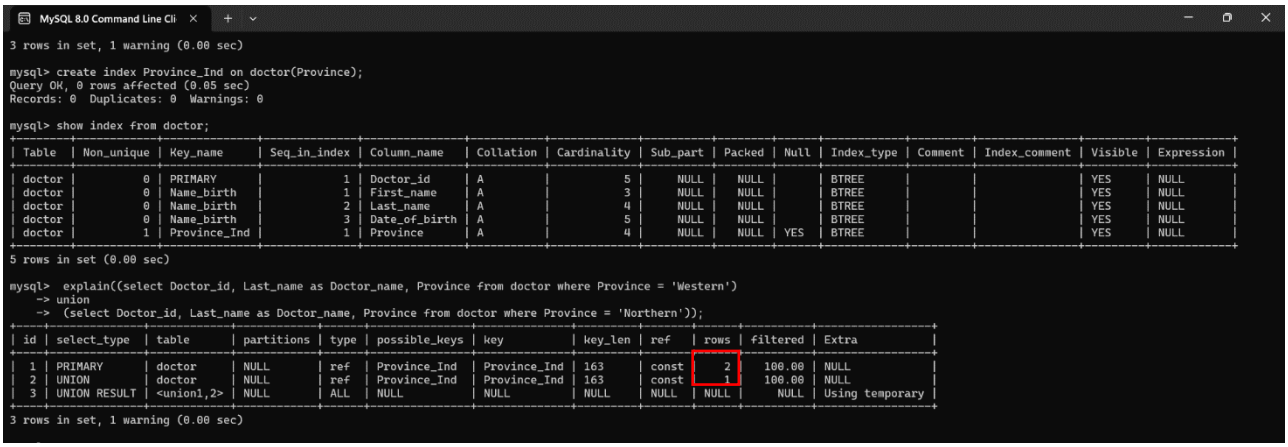
Tunning of a query is demonstrated in this chapter by comparing the number of accessed rows in the explain table before and after creating a suitable index. If the number of accessed rows decreases after the index is created, the query has been properly tuned. All of the screenshots included below meet that requirement.

❖ Tuning of Union Operation

- Before Tuning

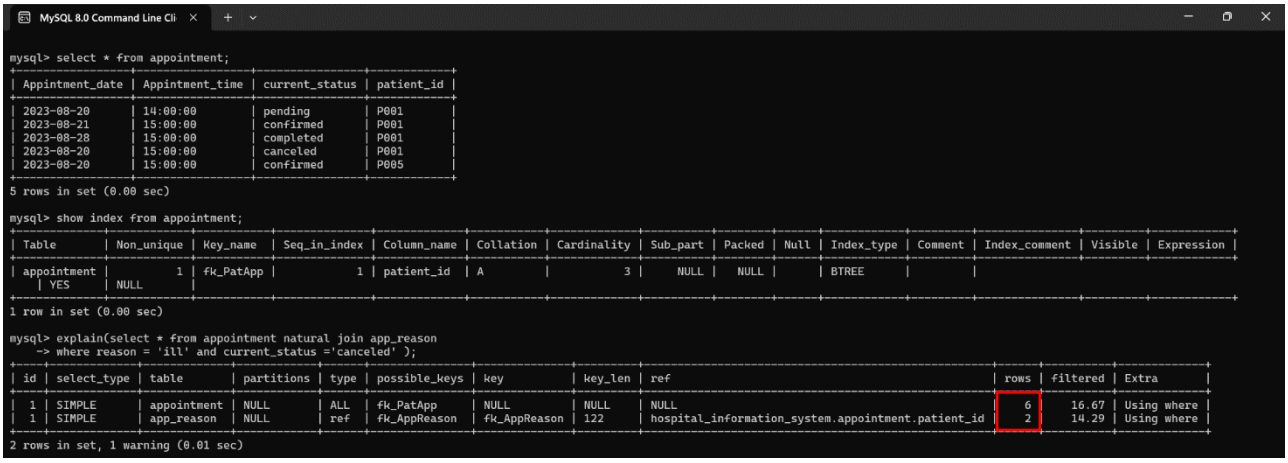


- After Tuning

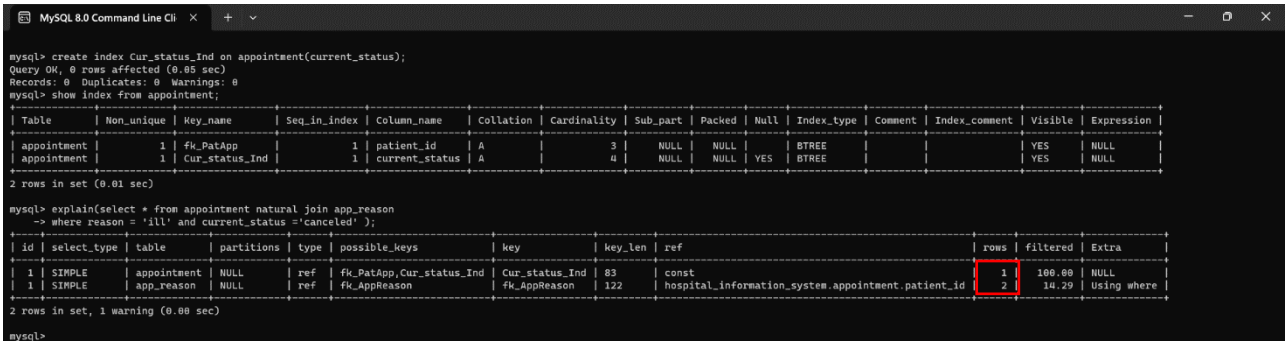


❖ Tuning of Intersection operation

- Before Tuning

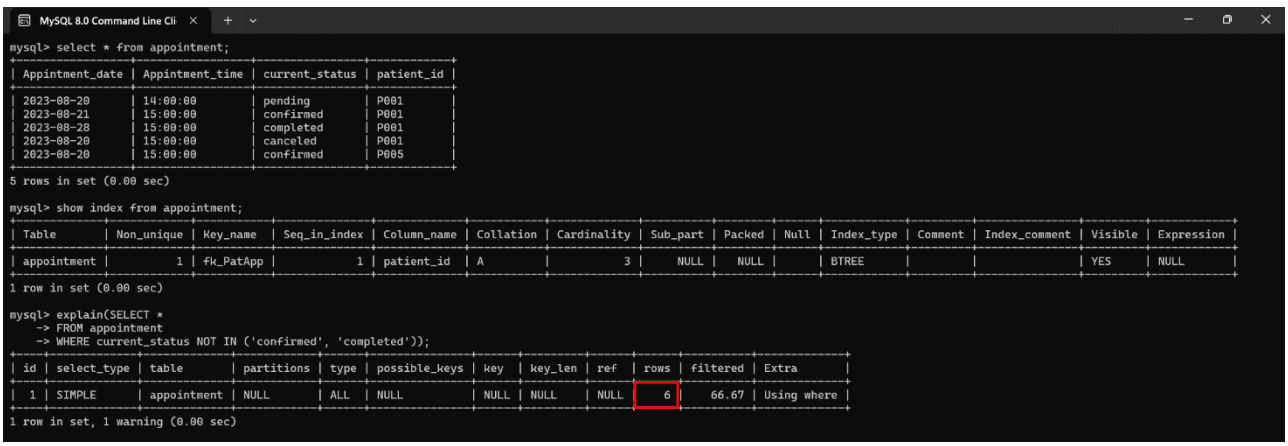


- After Tuning

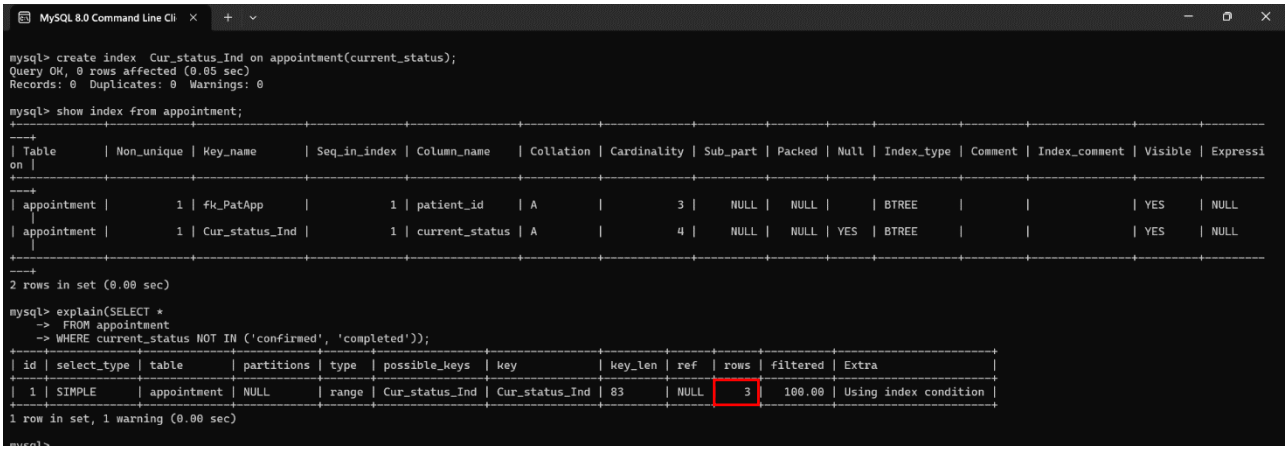


❖ Tuning of Set Difference Operation

- Before Tuning

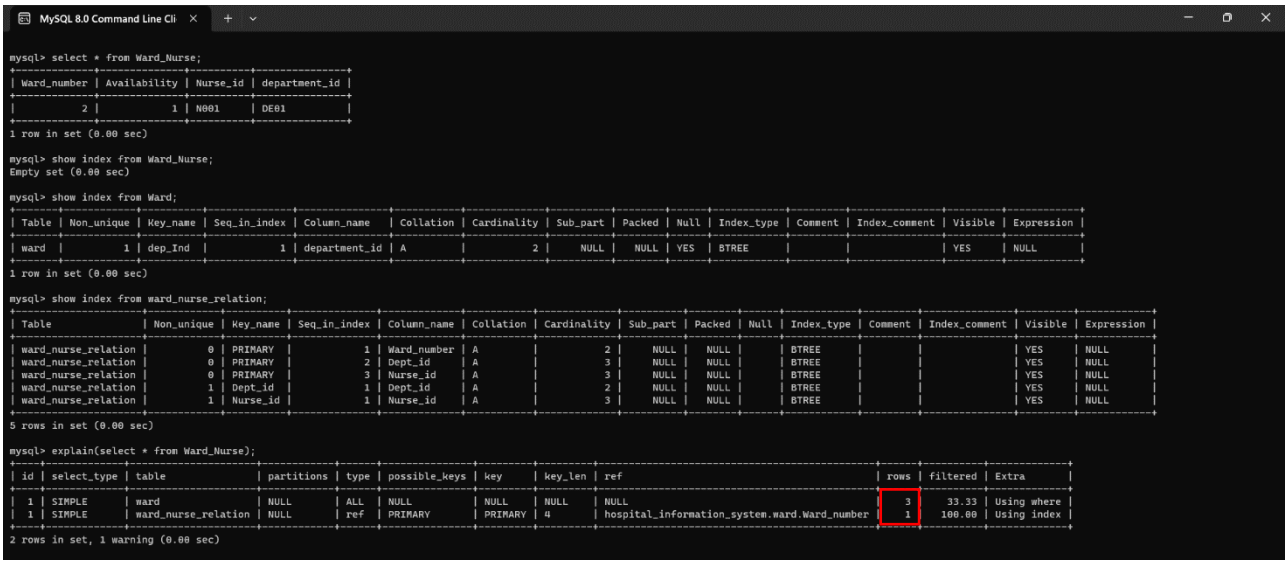


- After Tuning

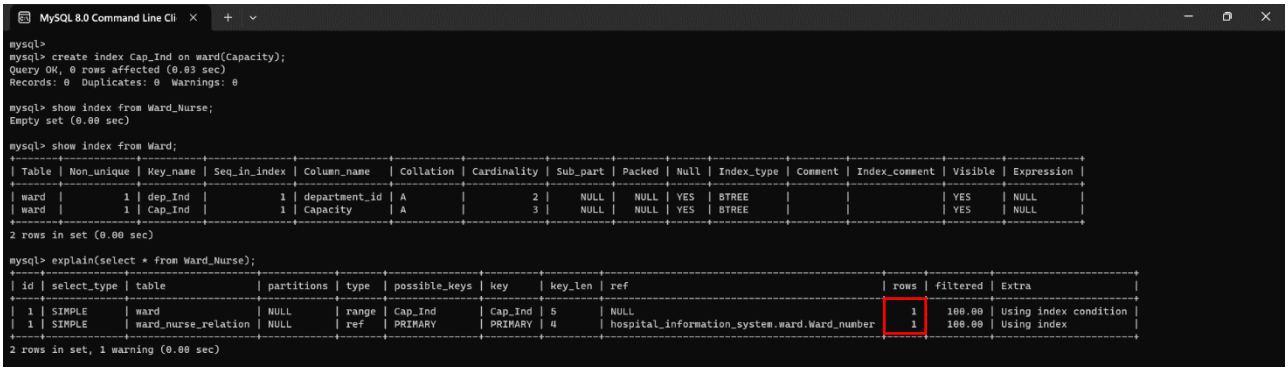


❖ Tuning of Natural join operation

- Before Tuning

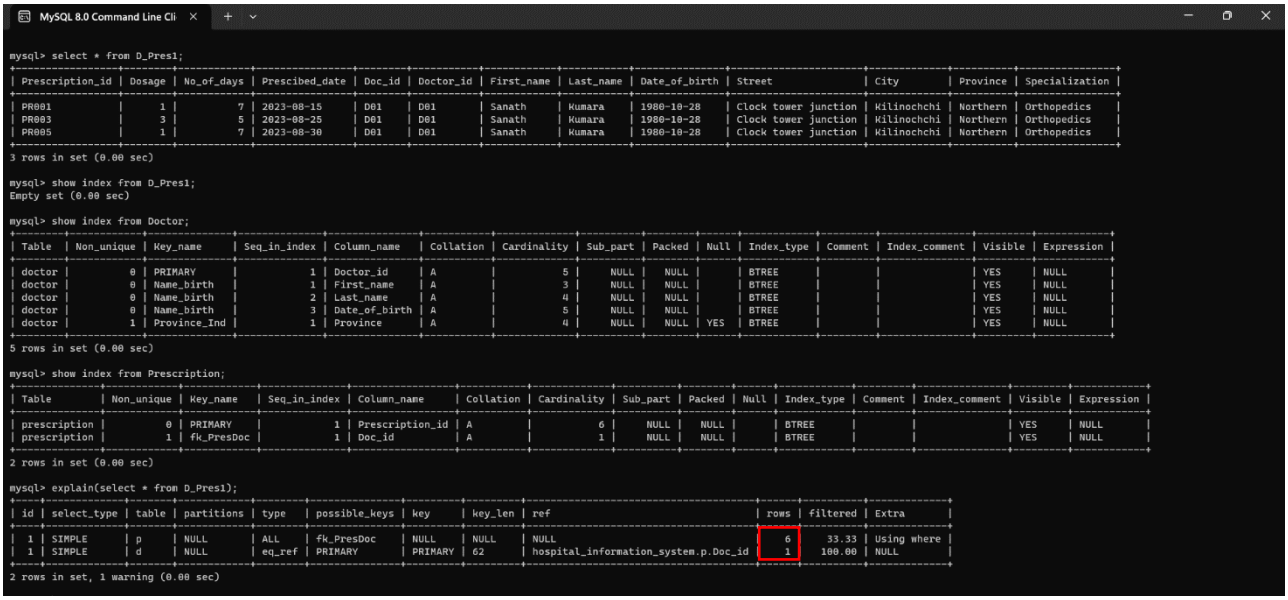


- After Tuning



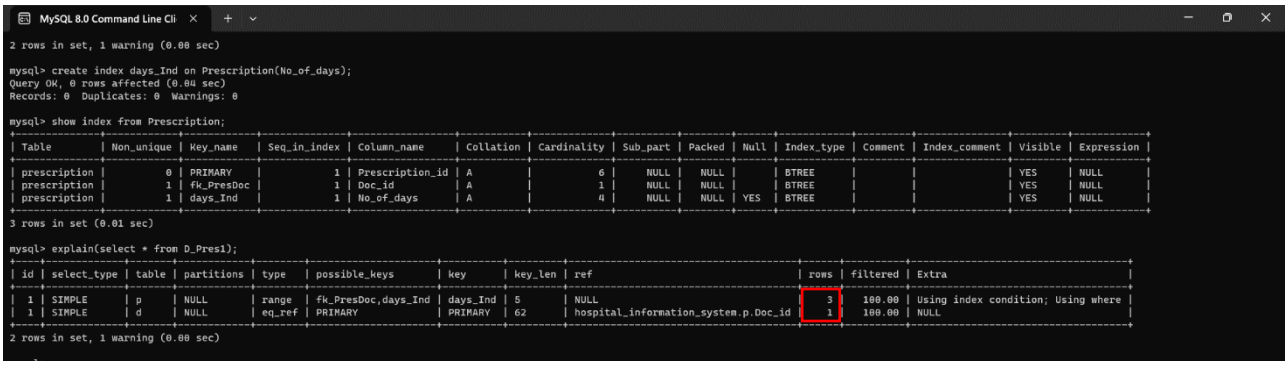
❖ Tuning of Right outer join operation

- Before Tuning



```
mysql> select * from D_Pres1;
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
| Prescription_id | Dosage | No_of_days  | Prescibed_date | Doc_id  | Doctor_id | First_name | Last_name  | Date_of_birth | Street              | City     | Province       | Specialization |
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
| PR001           |      1 |           7 | 2023-08-15     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
| PR003           |      3 |           5 | 2023-08-25     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
| PR005           |      1 |           7 | 2023-08-30     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
3 rows in set (0.00 sec)

mysql> show index from D_Pres1;
Empty set (0.00 sec)

mysql> show index from Doctor;
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name      | Seq_in_index | Column_name  | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| doctor |          0 | PRIMARY       |            1 | Doctor_id    | A         |           5 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            1 | First_name   | A         |           3 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            2 | Last_name    | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            3 | Date_of_birth | A         |           5 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          1 | Province_Ind  |            1 | Province     | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
5 rows in set (0.00 sec)

mysql> show index from Prescription;
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table        | Non_unique | Key_name     | Seq_in_index | Column_name     | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| prescription |          0 | PRIMARY      |            1 | Prescription_id | A         |           6 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| prescription |          1 | fk_PresDoc   |            1 | Doc_id          | A         |           1 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.00 sec)

mysql> explain(select * from D_Pres1);
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
| id | select_type | table | partitions | type   | possible_keys | key     | key_len | ref                                     | rows | filtered | Extra       |
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
|  1 | SIMPLE      | p     | NULL       | ALL    | fk_PresDoc    | NULL    | NULL    | NULL                                    |    6 |    33.33 | Using where |
|  1 | SIMPLE      | d     | NULL       | eq_ref | PRIMARY       | PRIMARY | 62      | hospital_information_system.p.Doc_id |    1 |   100.00 | NULL        |
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
2 rows in set, 1 warning (0.00 sec)
```

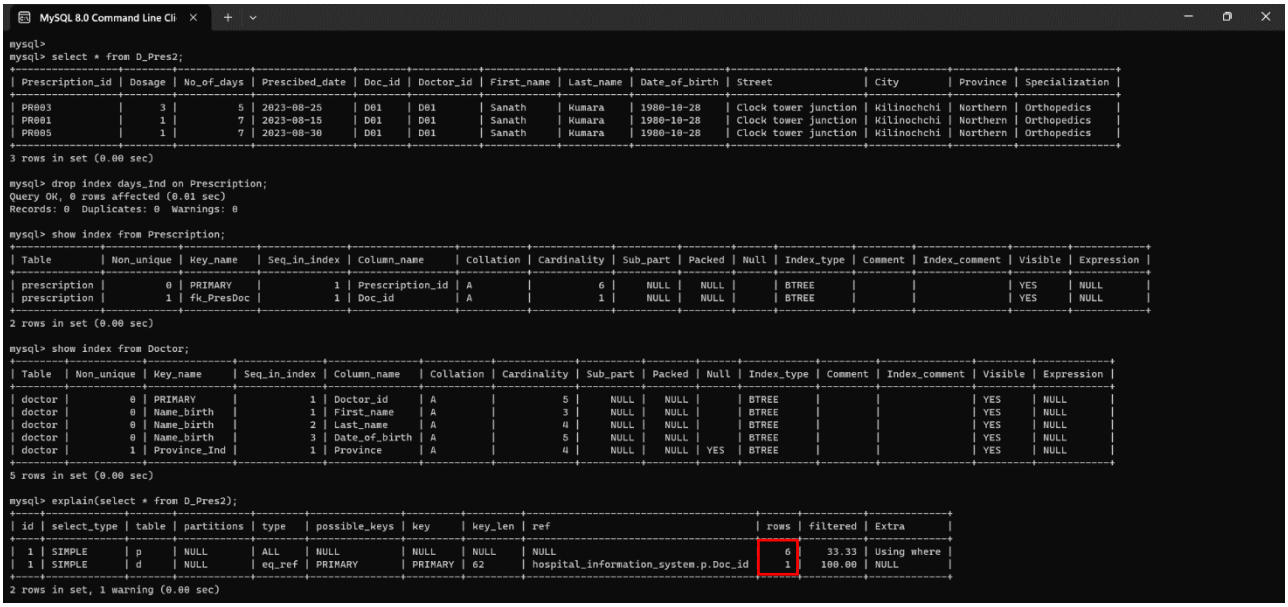
- After Tuning



```
2 rows in set, 1 warning (0.00 sec)

mysql> create index days_Ind on Prescription(No_of_days);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from Prescription;
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table        | Non_unique | Key_name     | Seq_in_index | Column_name     | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| prescription |          0 | PRIMARY      |            1 | Prescription_id | A         |           6 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| prescription |          1 | fk_PresDoc   |            1 | Doc_id          | A         |           1 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| prescription |          1 | days_Ind     |            1 | No_of_days      | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.01 sec)

mysql> explain(select * from D_Pres1);
+----+-------------+-------+------------+--------+-----------------------+----------+---------+-----------------------------------------+------+----------+-----------------------------+
| id | select_type | table | partitions | type   | possible_keys         | key      | key_len | ref                                     | rows | filtered | Extra                       |
+----+-------------+-------+------------+--------+-----------------------+----------+---------+-----------------------------------------+------+----------+-----------------------------+
|  1 | SIMPLE      | p     | NULL       | range  | fk_PresDoc,days_Ind | days_Ind | 5       | NULL                                    |    3 |   100.00 | Using index condition; Using where |
|  1 | SIMPLE      | d     | NULL       | eq_ref | PRIMARY               | PRIMARY  | 62      | hospital_information_system.p.Doc_id |    1 |   100.00 | NULL                        |
+----+-------------+-------+------------+--------+-----------------------+----------+---------+-----------------------------------------+------+----------+-----------------------------+
2 rows in set, 1 warning (0.00 sec)
```

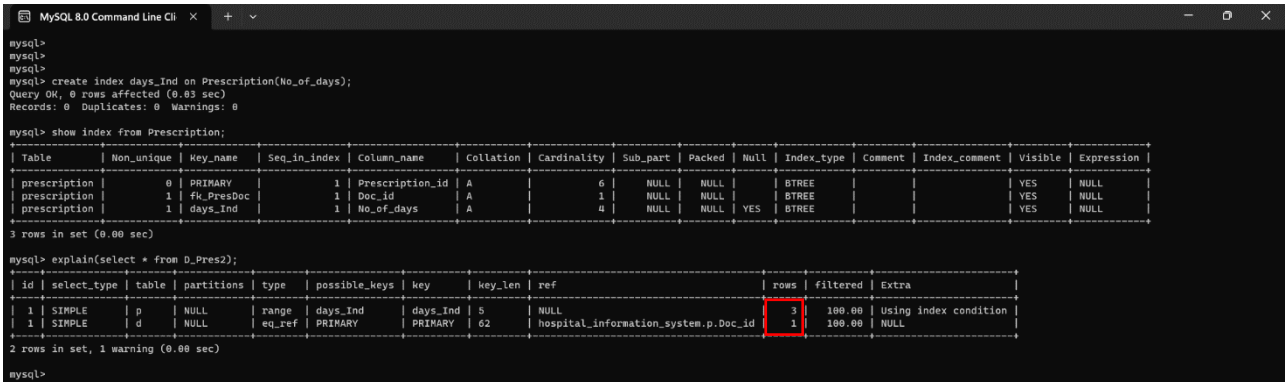
❖ Tuning of Left outer join operation

- Before Tuning



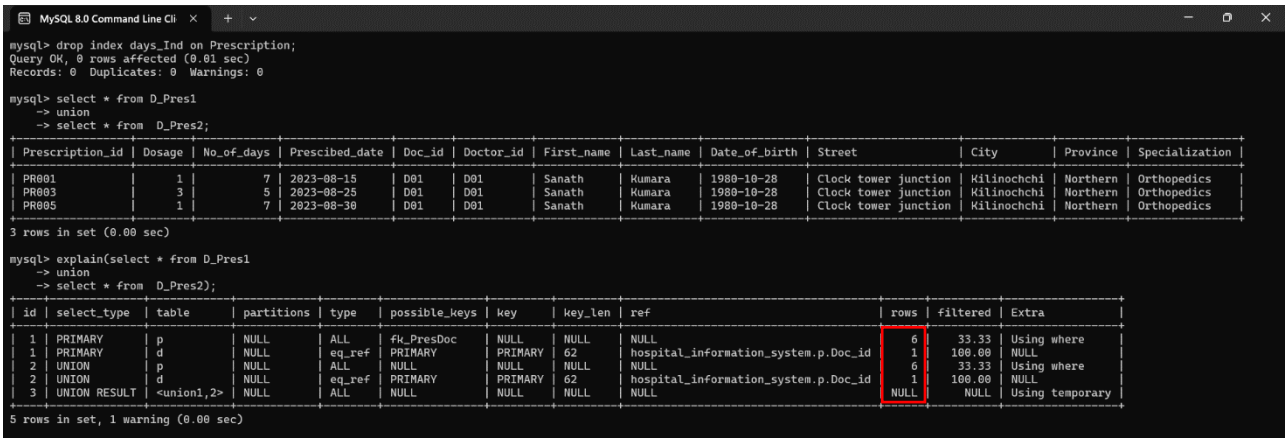
```
mysql>
mysql> select * from D_Pres2;
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
| Prescription_id | Dosage | No_of_days  | Prescibed_date | Doc_id  | Doctor_id | First_name | Last_name  | Date_of_birth | Street              | City     | Province       | Specialization |
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
| PR003           |      3 |           5 | 2023-08-25     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
| PR001           |      1 |           7 | 2023-08-15     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
| PR005           |      1 |           7 | 2023-08-30     | D01     | D01       | Sanath     | Kumara     | 1980-10-28    | Clock tower junction | Kilinochchi | Northern | Orthopedics    |
+-----------------+--------+-------------+----------------+---------+-----------+------------+------------+---------------+---------------------+----------+----------------+
3 rows in set (0.00 sec)

mysql> drop index days_Ind on Prescription;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from Prescription;
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table        | Non_unique | Key_name     | Seq_in_index | Column_name     | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| prescription |          0 | PRIMARY      |            1 | Prescription_id | A         |           6 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| prescription |          1 | fk_PresDoc   |            1 | Doc_id          | A         |           1 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+--------------+------------+--------------+--------------+-----------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.00 sec)

mysql> show index from Doctor;
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name      | Seq_in_index | Column_name  | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| doctor |          0 | PRIMARY       |            1 | Doctor_id    | A         |           5 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            1 | First_name   | A         |           3 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            2 | Last_name    | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          0 | Name_birth    |            3 | Date_of_birth | A         |           5 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| doctor |          1 | Province_Ind  |            1 | Province     | A         |           4 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+--------+------------+---------------+--------------+--------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
5 rows in set (0.00 sec)

mysql> explain(select * from D_Pres2);
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
| id | select_type | table | partitions | type   | possible_keys | key     | key_len | ref                                     | rows | filtered | Extra       |
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
|  1 | SIMPLE      | p     | NULL       | ALL    | NULL          | NULL    | NULL    | NULL                                    |    6 |    33.33 | Using where |
|  1 | SIMPLE      | d     | NULL       | eq_ref | PRIMARY       | PRIMARY | 62      | hospital_information_system.p.Doc_id |    1 |   100.00 | NULL        |
+----+-------------+-------+------------+--------+---------------+---------+---------+-----------------------------------------+------+----------+-------------+
2 rows in set, 1 warning (0.00 sec)
```

- After Tuning

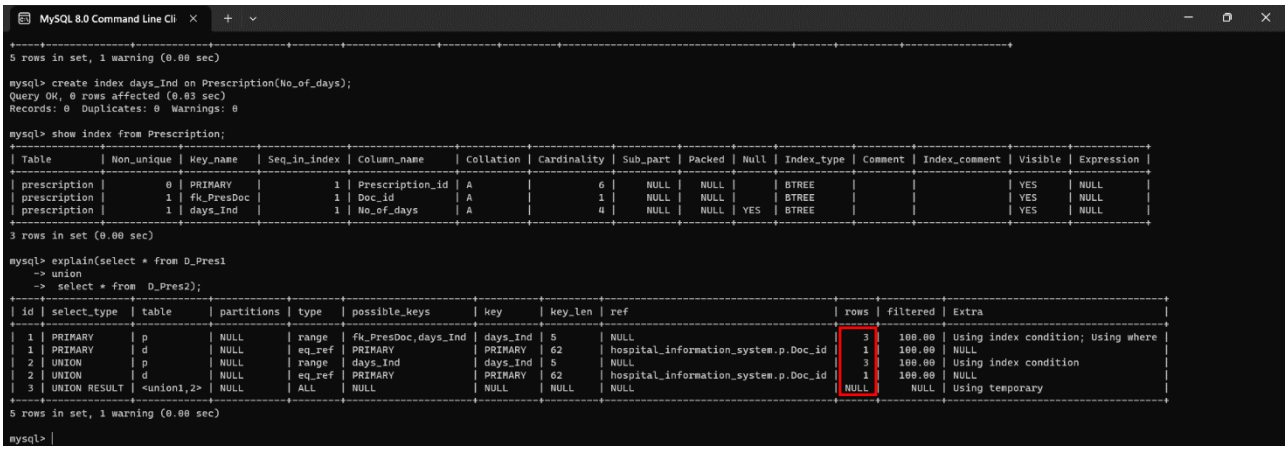


❖ Tuning of Full outer join operation

- Before Tuning

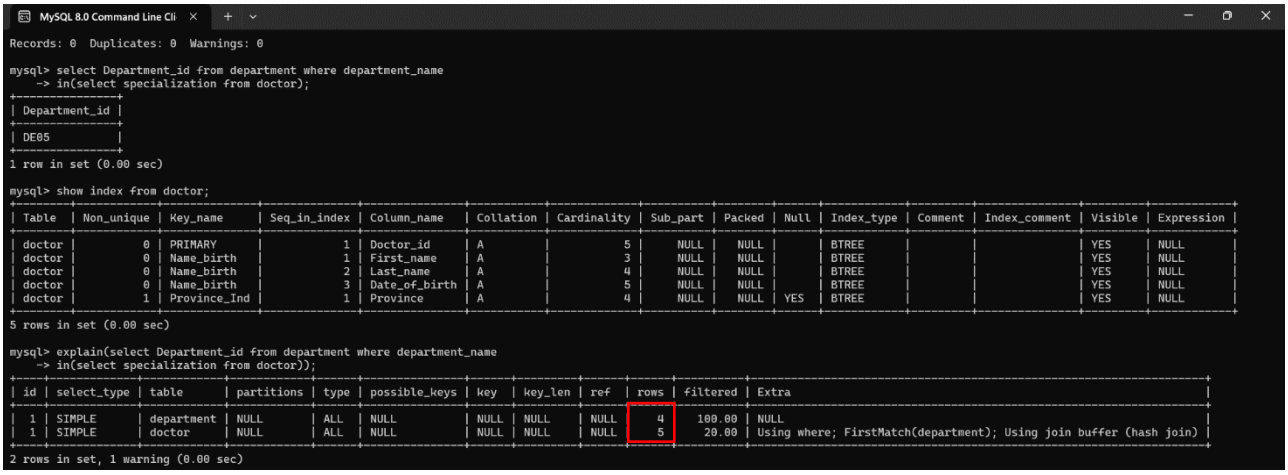


- After Tuning

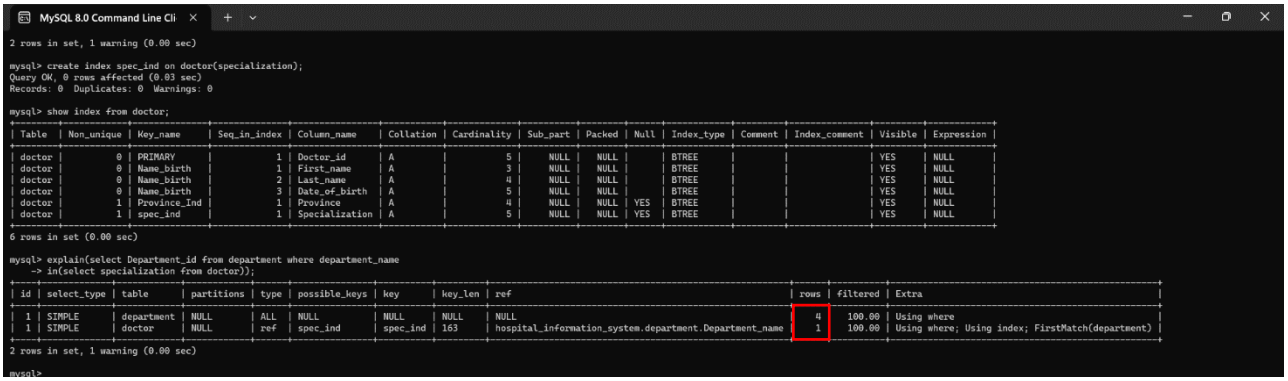


❖ Tuning of Nested Query 1

- Before Tuning

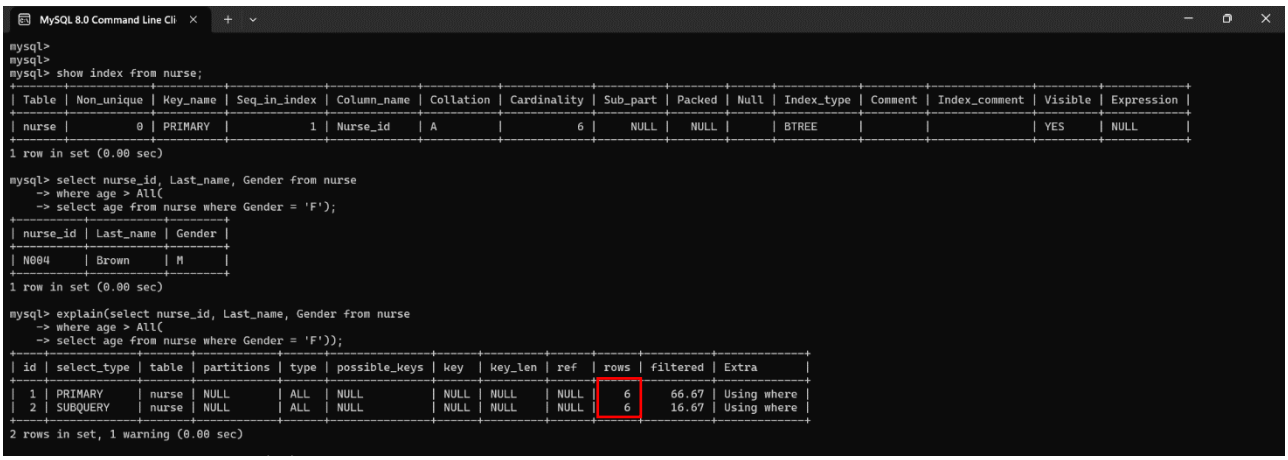


- After Tuning

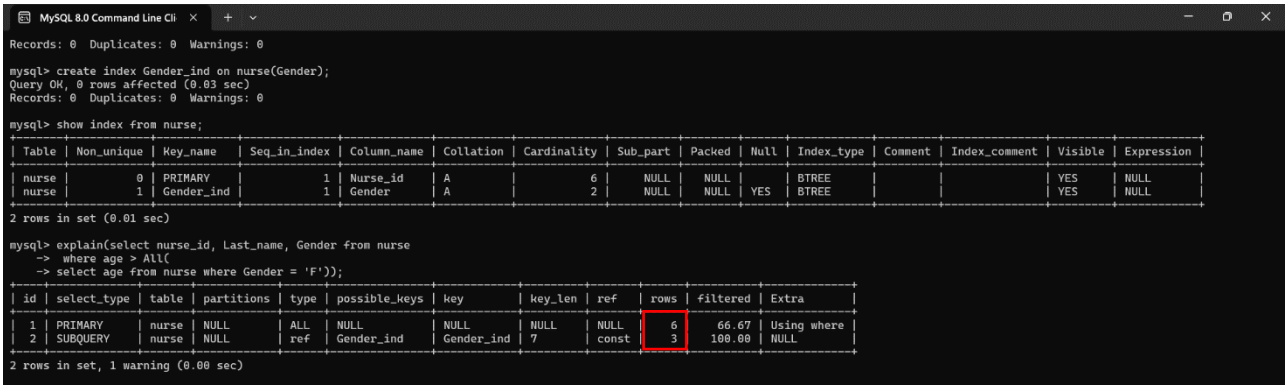


❖ Tuning of Nested Query 2

- Before Tuning

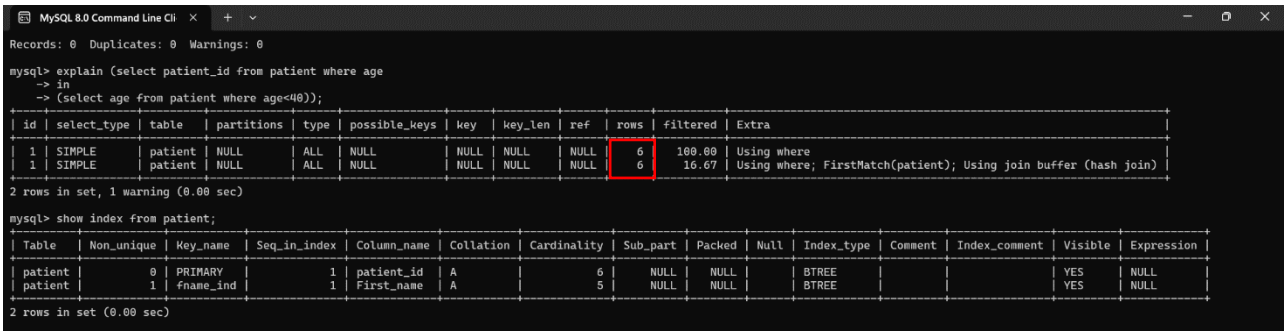


- After Tuning



❖ Tuning of Nested Query 3

- Before Tuning



- After Tuning