

**DEPARTMENT OF COMPUTER SCIENCE**

  FALL 2022-23

       DATABASE FINAL PROJECT REPORT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| COURSE | INTRODUCTION TO DATABASE | SECTION | G | GROUP | 04 |

**GROUP MEMBERS:**

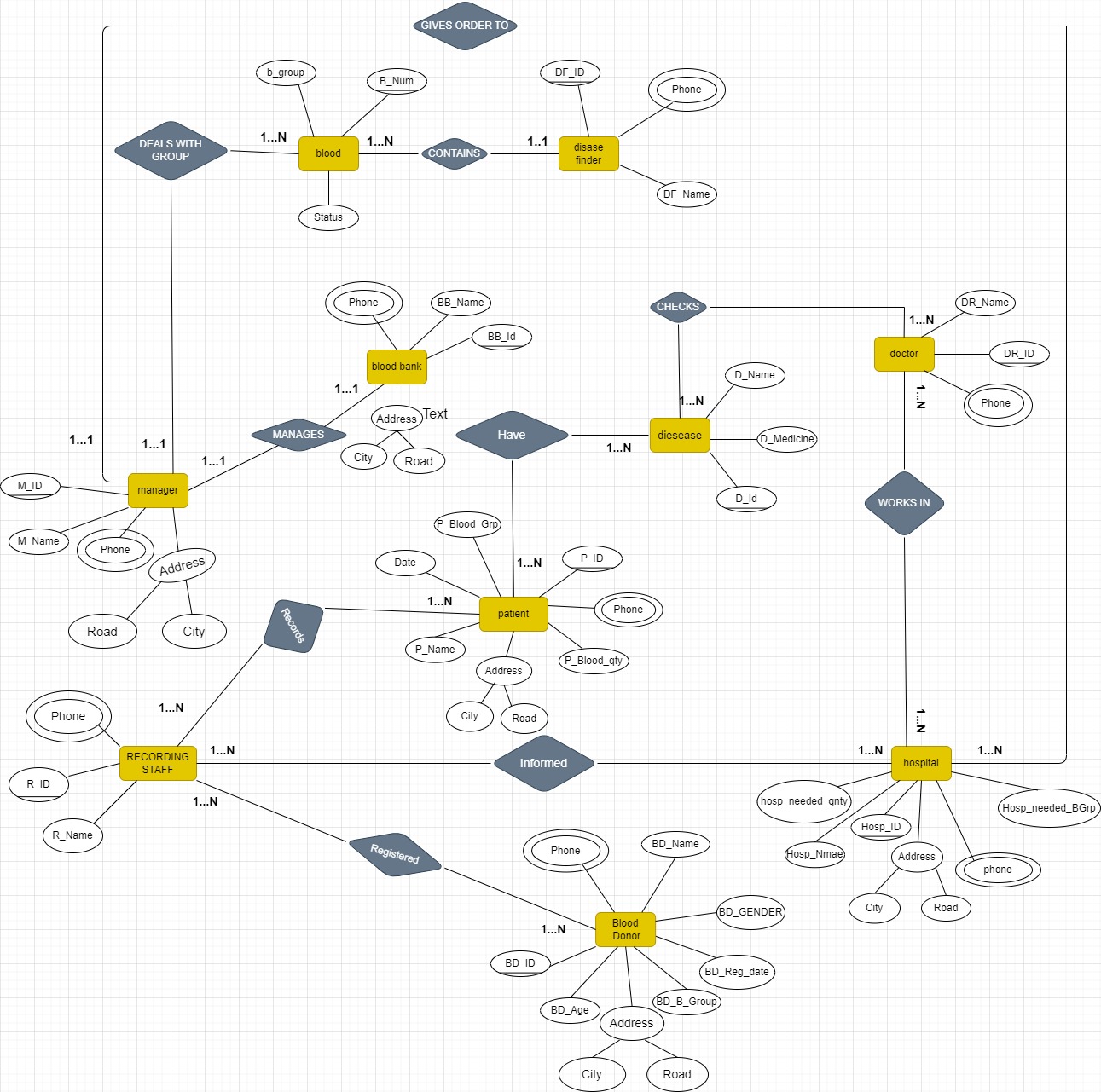
|  |  |
| --- | --- |
| NAME | STUDENT ID |
| MD TAFHIMUL HAQUE SADI | 22-47071-1 |
| MD RAKIBUL ISLAM | 22-47102-1 |
| ABDULLAH AL HASIB | 22-47055-1 |
| SOURAV MISTRY | 22-46077-1 |

**BLOOD DONATION MANAGEMENT SYSTEM**

CASE STUDY:

Blood is collected, stored, and given to people who need it by a blood bank. Donors are the individuals who give blood. Following that, the banks separate the blood they receive into the various blood groups. They also check to see if the blood has been contaminated. The primary goal of the blood bank is to supply the hospitals and healthcare systems with blood that will save the patient's life. Without sufficient and pure blood, hospital can keep the healthcare system running. Each blood bank's top priority is to keep an eye on the blood's quality as well as the "donors," or those who provide blood. The blood bank manager’s no responsibility is to connect between donor and hospital. Every patient will have different attributes. Doctor will check the donor’s blood for possible diseases and he/she works in a hospital. The recording staff’s duty is to keep all the records of the donor, patient and provide the information to the hospital. The disease finder will test blood sample to find any kind of disease. The existence of disease entity is totally depends on patient entity. Every patient must have a disease. The manager manages many blood samples, but he/she deals with only one blood bank at a time. A recording staff or a doctor can work for one or many hospitals at a time. A disease finder tests one or many blood samples. Doctor checks a patient for one or many kinds of disease. Recording staff keeps and passes all the records of blood donor and patient to the hospital. To identify the blood donors and the patients, recording staff and hospital stores their ID along with their name. Every doctor, manager, recording staff, hospital, blood bank has an unique identification number to specifically identify them.

ER DIAGRAM:

****

INFORMATION OF ENTITIES:

In total we have eight entities and information of each entity is mentioned below:-

1. **Blood\_Donor**: (Attributes – bd\_ID, bd\_name, bd\_sex, bd\_age, bd\_Bgroup, bd\_reg\_date, phone, city, road)

The donor is the person who donates blood, on donation a donor id (bd\_ID) is generated and used as primary key to identify the donor information. Other than that name, age , sex , blood group, phone number and registration dates will be stored in database under Blood\_Donor entity.

2. **Manager:** (Attributes – m\_ID, m\_Name, phone, city, road)

The blood bank manager is the person who takes care of the available blood samples in the blood bank, he is also responsible for handling blood requests from recipients and hospitals. Blood manager has a unique indentification number (m\_ID) used as primary key along with name and phone number of blood bank manager will be stored in data base under BB\_Manager entity.

3. **Recording\_Staff :** (Attributes – r\_ID, r\_Name, phone)

The recording staff is a person who registers the blood donor and recipients and the Recording\_Staff enitity has r\_ID which is primary key along with recorder’s name and recorder’s phone number will also be stored in the data base under Recording \_Staff entity.

4. **DiseaseFinder:** (Attributes - df\_ID, df\_name, phone)

In data base , under DiseaseFinder entity we will store the information of the doctor who checks the blood for any kind of contaminations. To store that information we have unique identification number (df\_ID) as primary key. Along with name and phone number of the doctor will also be stored under same entitity.

5. **Hospital**: (Attributes – hosp\_ID, hosp\_name, hosp\_needed\_Bgrp, hosp\_needed\_Bqnty, phone, cIty, road) In the data base, under Hospital entity we will store the information of hospitals. In this hosp\_ID is the primary key. We will store hospital name and the blood quantity reqiured at the hospital.

6.**Patient:**(Attributes–P\_BLOOD\_GROUP,P\_ID,P\_NAME,P\_BLOOD\_QTY,ADDRESS,DATE,PHONE)

In the data base, Patient entity will have all the information of the patient who will receive the blood. Here P\_ID is the unique identification number as primary key.

7. **Blood**: (Attributes- B\_GROUP,B\_NUM,STATUS)

In the database, Blood entity will have all the information like blood group and current status of the blood. We have B\_NUM as a primary key in this entity.

8. **Doctor**: (Attributes- DR\_NAME,DR\_ID,PHONE)

In the database, Doctor entity will have details information of the doctor who will check patients to find the disease. Here DR\_ID is the unique identification number as primary key

9. **Blood Bank**: (Attributes- BB\_NAME,BB\_ID,PHONE,ADDRESS)

In the database, Blood bank entity will have details information of the Blood bank where collected blood’s are stored. We have BB\_ID as a primary key in this entity.

10. **Disease**: (Attributes- D\_ID,D\_NAME,D\_MEDICINE)

In the data base, Disease entity will have all the information of the disease which can be found by checking the patient. Here D\_ID is the unique identification number as primary key.

Normalization**:**

**1. Registered (R\_ID, R-Name ,Phone, BD-Id, BD-Age, BD-B-Group, BD-Reg-Date, BD-Sex, BD-Name, City, Road, Phone)**

1NF: Phone is multivalued ,

Registered (R\_ID, R-Name ,Phone, BD-Id, BD-Age, BD-B-Group, BD-Reg-Date, BD-Sex, BD- Name, City, Road, Phone)

2NF:

1. R-id, R-Name, BD-ID, PH-id
2. Phone , PH-Id
3. BD-Id, BD-Age ,BD-G-Group ,BD-Reg-Date, BD-Sex, BD-Name, City ,Road, city, road, PH-ID
4. PHONE, PH-ID

 3 NF:

1. R-id, R-Name, BD-Id, PH-Id
2. Phone, PH-Id
3. BD-Id, BD-Name, BD-Age ,BD-B-Group ,BD-Reg-Date, BD-GENDER, PH-Id, A-Id
4. City, Road, A-Id

**2. HAVE (D-Id, D-NAME, D-MEDICINE, P-ID, P-NAME, P-BLOOD-GRP, P-BLOOD-QNTY, DATE, PHONE, CITY)**

1 NF:

PHONE is multivalued

HAVE (D-Id, D-NAME, D-MEDICINE, P-ID, P-NAME, P-BLOOD-GRP, P-BLOOD-QNTY, DATE, CITY)

2 NF:

* + 1. D-Id, D-NAME, D-MEDICINE, P-ID
    2. P-ID, P-NAME, P-BLOOD-GRP, P-BLOOD-QNTY, DATE, CITY, ROAD
    3. PHONE, PH-ID

3 NF:

* + 1. D-Id, D-NAME, D-MEDICINE, P-ID
    2. P-ID, P-NAME, DATE, PH-ID, A-ID
    3. PHONE, PH-ID
    4. A-ID, CITY, ROAD
    5. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID

**3. CHECKS (D-ID, D-NAME, D-MEDICINE, DR-ID, DR-NAME, PHONE)**

1 NF: PHONE is multivalued

CHECKS (D-ID, D-NAME, D-MEDICINE, DR-ID, DR-NAME)

2 NF:

* 1. D-ID, D-NAME, D-MEDICINE
  2. PH-ID, PHONE
  3. DR-ID, DR-NAME, D-ID, PH-ID

3 NF:

* 1. D-ID, D-NAME, D-MEDICINE
  2. PH-ID, PHONE
  3. DR-ID, DR-NAME, D-ID, PH-ID

**4. DEALS WITH GROUP (M-ID, M\_NAME, PHONE, ROAD, CITY, B\_GROUP, B-NUM, STATUS)**

1 NF: PHONE IS A MULTIVALUED

DEALS WITH GROUP(M-ID, M\_NAME, ROAD, CITY, B\_GROUP, B-NUM, STATUS)

2 NF:

1. M-ID, M\_NAME, ROAD, CITY, PH-ID
2. PH-ID, PHONE
3. B-NUM, B\_GROUP, STATUS, M-ID

3 NF:

* 1. M-ID, M\_NAME, PH-ID, A-ID
  2. PH-ID, PHONE
  3. A-ID, CITY, ROAD
  4. B-NUM, B\_GROUP, STATUS, M-ID

**5. RECORDS (R-ID, R-NAME, PHONE, P-ID,P-NAME,P-BLOOD-GRP,P-BLOOD-QTY,DATE,PHONE,CITY,ROAD)**

1 NF: PHONE IS MULTI-VALUED.

RECOEDS (R-ID, R-NAME, P-ID, P-NAME,P-BLOOD-GRP,P-BLOOD-QTY,DATE ,CITY,ROAD)

2NF:

1. R-ID, R-NAME, P-ID,
2. PH-ID, PHONE
3. P-ID, P-NAME, P-BOOD-GRP, P-BLOOD-QNTY, DATE, CITY, ROAD

3NF:

1. R-ID, R-NAME, P-ID, PH-ID
2. PH-ID, PHONE
3. P-ID, P-NAME, PH-ID, DATE, A-ID
4. A-ID, CITY, ROAD
5. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID

**6. MANAGES (M-ID, M-NAME, PHONE, CITY, ROAD, BB-NAME, BB-ID, PHONE)**

1NF:

PHONE IS MULTI-VALUED

MANAGES (M-ID, M-NAME, CITY, ROAD,BB-NAME,BB-ID)

2NF:

1. M-ID, M-NAME,CITY,ROAD, PH-ID
2. PHONE, PH-ID
3. BB-ID, BB-NAME, M-ID
4. PHONE, PH-ID

3NF:

1. M-ID,M-NAME, PH-ID, A-ID
2. CITY,ROAD,A-ID
3. PHONE,PH-ID
4. BB-ID, BB-NAME, PH-ID, A-ID, M-ID

**7.INFORMED(R-ID,R-NAME,PHONE,HOSP-ID,HOSP-NAME,PHONE,HASP-NEEDED-BGRP,CITY,ROAD, HOSP-NEEDED- QNTY)**

1NF: PHONE IS MULTI-VALUED

INFORMED (R-ID,R-NAME ,HOSP-ID,HOSP-NAME ,HASP-NEEDED-BGRP,CITY,ROAD, HOSP-NEEDED- QNTY)

2NF:

1. R-ID,R-NAME,HOSP-ID,PH-ID
2. PH-ID, PHONE
3. HOSP-ID, HOSP-NAME,HOSP-NEEDED-BGRP,HOSP-NEEDED-QNTY,CITY,ROAD,PH-ID

3NF:

1. R-ID,R-NAME,HOSP-ID,PH-ID
2. PH-ID, PHONE
3. HOSP-ID, HOSP-NAME,A-ID, PH-ID
4. A-ID,CITY,ROAD
5. HOSP-NEEDED-B-GRP,HOSP-NEEDED-QNTY,HOSP-ID

**8. WORKS IN (DR-ID,DR-NAME,PHONE,HOSP-ID,HODP-NAME,HOSP-NEEDED-QNTY, HOSP-NEEDED-BGRP,PHONE,CITY,ROAD)**

1NF: PHONE IS MULTI-VALUED

WORKS IN (DR-ID,DR-NAME ,HOSP-ID,HODP-NAME,HOSP-NEEDED-QNTY, HOSP-NEEDED-BGRP, CITY,ROAD)

2NF:

1. DR-ID,DR-NAME,PH-ID,HOSP-ID
2. PH-ID,PHONE
3. HOSP-ID,HOSP-NAME,HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP,CITY,ROAD,PH-ID

3NF:

1. DR-ID,DR-NAME,PH-ID,HOSP-ID
2. PH-ID,PHONE
3. HOSP-ID,HOSP-NAME,PH-ID,A-ID
4. A-ID,CITY,ROAD
5. HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP,HOSP-ID

**9. GIVES ORDER TO ( M-ID,M-NAME,PHONE,ROAD,CITY, HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP, HOSP-ID, HOSP-NAME,CITY,ROAD,PHONE)**

1NF: PHONE IS MULTI-VALUED

GIVES ORDER TO ( M-ID,M-NAME ,ROAD,CITY, HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP, HOSP-ID, HOSP-NAME,CITY,ROAD)

2NF:

1. M-ID,M-NAME,ROAD,CITY,PH-ID
2. PH-ID,PHONE
3. HOSP-ID,HOSP-NAME,HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP,CITY,ROAD,PH-ID,M-ID

3NF:

1. M-ID,M-NAME,PH-ID,A-ID
2. PH-ID,PHONE
3. A-ID,CITY,ROAD
4. HOSP-ID, HOSP-NAME, PH-ID, A-ID, M-ID
5. HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP, HOSP-ID

**10. CONTAINS ( B-GROUP,B-NUM,STATUS,DF-ID,PHONE,DF-NAME)**

1NF: PHONE IS MULTI-VALUED

CONTAINS ( B-GROUP,B-NUM,STATUS,DF-ID ,DF-NAME)

2 NF:

1. DF-ID,DF-NAME,PH-ID
2. PH-ID,PHONE
3. B-GROUP,B-NUM,STATUS,DF-ID

3 NF:

1. DF-ID,DF-NAME,PH-ID
2. PH-ID,PHONE
3. B-GROUP,B-NUM,STATUS,DF-ID

**Total Table Creation:**

1. R-id, R-Name, BD-Id, PH-Id

1. Phone, PH-Id
2. BD-Id, BD-Name, BD-Age ,BD-B-Group ,BD-Reg-Date, BD-GENDER, PH- Id, A-Id
3. City, Road, A-Id
4. R-ID, R-NAME, P-ID, PH-ID
5. PH-ID, PHONE
6. P-ID, P-NAME, PH-ID, DATE, A-ID
7. A-ID, CITY, ROAD
8. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID
9. R-ID,R-NAME,HOSP-ID,PH-ID
10. PH-ID, PHONE
11. HOSP-ID, HOSP-NAME,A-ID, PH-ID
12. A-ID,CITY,ROAD
13. HOSP-NEEDED-B-GRP,HOSP-NEEDED-QNTY,HOSP-ID
14. D-Id, D-NAME, D-MEDICINE, P-ID
15. P-ID, P-NAME, DATE, PH-ID, A-ID
16. PHONE, PH-ID
17. A-ID, CITY, ROAD
18. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID
19. D-ID, D-NAME, D-MEDICINE
20. PH-ID, PHONE
21. DR-ID, DR-NAME, D-ID, PH-ID
22. DR-ID,DR-NAME,PH-ID,HOSP-ID
23. PH-ID,PHONE
24. HOSP-ID,HOSP-NAME,PH-ID,A-ID
25. A-ID,CITY,ROAD
26. HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP,HOSP-ID
27. M-ID,M-NAME,PH-ID,A-ID
28. PH-ID,PHONE
29. A-ID,CITY,ROAD
30. HOSP-ID, HOSP-NAME, PH-ID, A-ID, M-ID
31. HOSP-NEEDED-QNTY,HOSP-NEEDED-BGRP, HOSP-ID
32. M-ID,M-NAME, PH-ID, A-ID
33. CITY,ROAD,A-ID
34. PHONE,PH-ID
35. BB-ID, BB-NAME, PH-ID, A-ID, M-ID
36. M-ID, M\_NAME, PH-ID, A-ID
37. PH-ID, PHONE
38. A-ID, CITY, ROAD
39. B-NUM, B\_GROUP, STATUS, M-ID
40. DF-ID,DF-NAME,PH-ID
41. PH-ID,PHONE
42. B-GROUP,B-NUM,STATUS,DF-ID

**Final Table:**

1.RECORDING\_STAFF (R-ID, R-NAME, PH-Id, BD-ID, P-ID,  HOSP-ID)

2. PHONE (PHONE, PH-ID)

3.ADDRESS (CITY, ROAD, A-ID)

4. DONER (BD-ID, BD-NAME, BD-AGE ,BD-B-GROUP ,BD-REG-DATE, BD-GENDER, PH-ID, A-ID)

5. PATIENT (P-ID, P-NAME, PH-ID, DATE, A-ID)

6. PATIENT\_BLOOD\_INFO (P-BLOOD-GRP, P-BLOOD-QNTY, P-ID)

7. HOSPITAL (HOSP-ID, HOSP-NAME, A-ID, PH-ID)

8. H\_NEEDED\_BLOOD (HOSP-NEEDED-B-GRP, HOSP-NEEDED-QNTY, HOSP-ID)

9. DISEASE (D-ID, D-NAME, D-MEDICINE, P-ID)

10. DOCTOR (DR-ID, DR-NAME, PH-ID, HOSP-ID, D-ID)

11. MANAGER (M-ID, M-NAME, PH-ID, A-ID)

12. GIVES\_ORDER (HOSP-ID, HOSP-NAME, PH-ID, A-ID, M-ID)

13. BLOOD\_BANK (BB-ID, BB-NAME, PH-ID, A-ID, M-ID)

14. BLOOD\_GROUP (B-NUM, B\_GROUP, STATUS, M-ID, DF-ID)

15. DISEASE\_FINDER (DF-ID, DF-NAME, PH-ID)

TABLE CREATION:

1 . PHONE TABLE

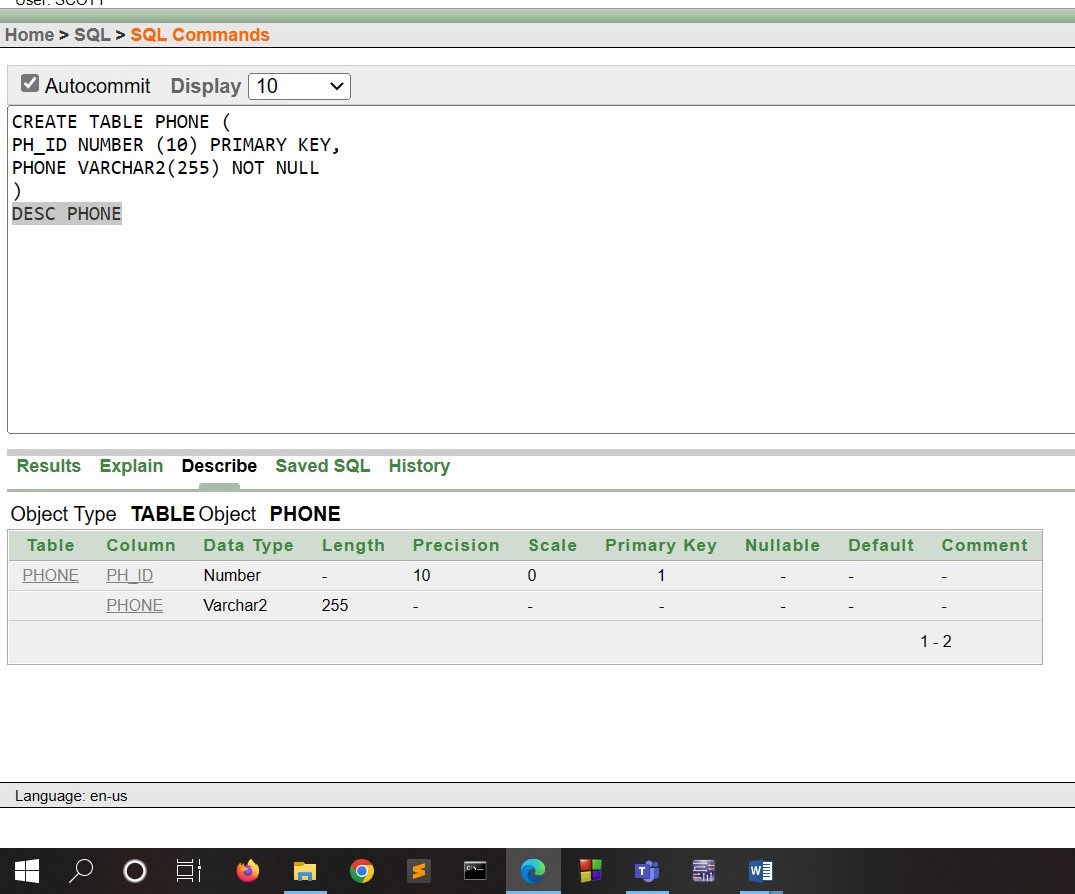
QUERY:

CREATE TABLE PHONE (

PH\_ID NUMBER (10) PRIMARY KEY,

PHONE VARCHAR2(255) NOT NULL

)



2 . ADDRESS TABLE

QUERY:

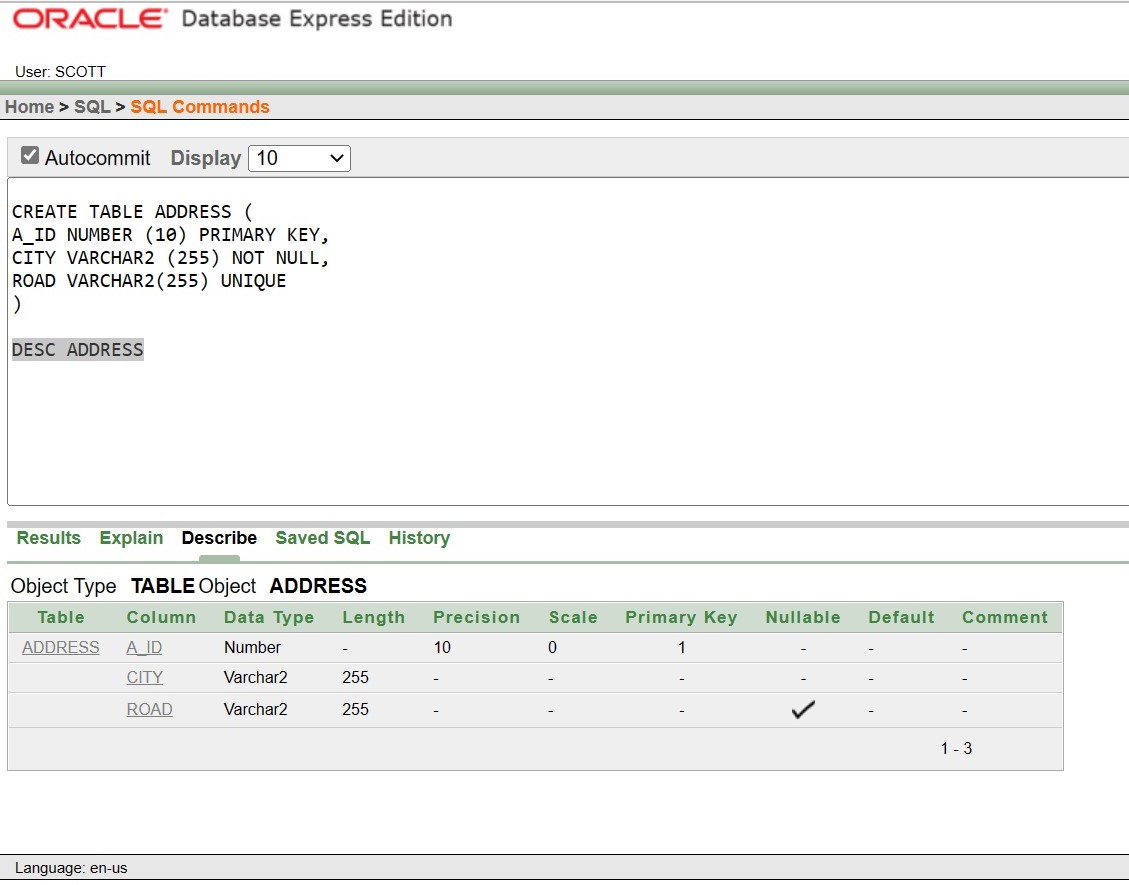
CREATE TABLE ADDRESS (

A\_ID NUMBER (10) PRIMARY KEY,

CITY VARCHAR2 (255) NOT NULL,

ROAD VARCHAR2(255) UNIQUE

)



3 . DONER TABLE

QUERY:

CREATE TABLE DONER(

BD\_ID INT PRIMARY KEY,

BD\_NAME VARCHAR2 (255) NOT NULL,

BD\_AGE NUMBER (10) CHECK (BD\_AGE>18),

BD\_B\_GROUP VARCHAR2 (255) UNIQUE,

BD\_REG\_DATE DATE,

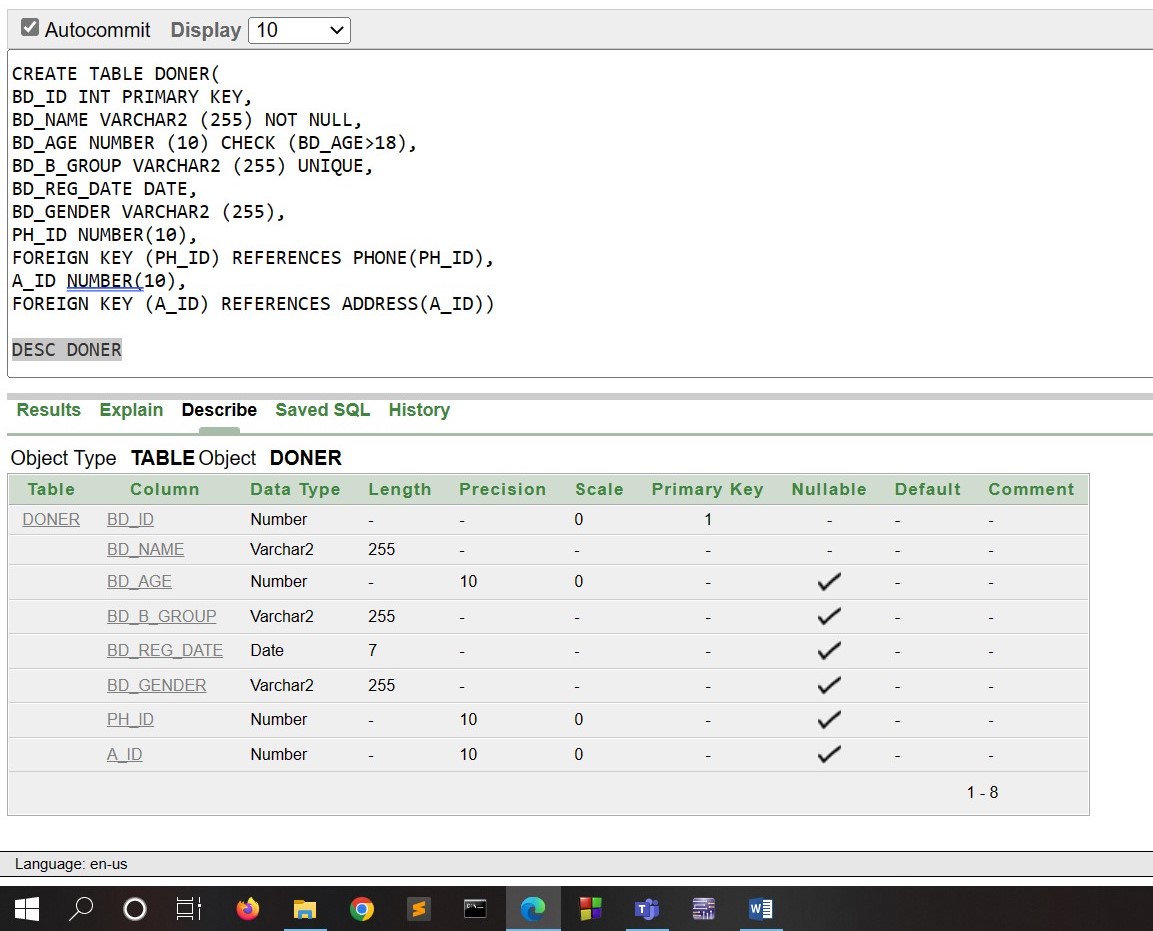
BD\_GENDER VARCHAR2 (255),

PH\_ID NUMBER(10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

A\_ID NUMBER(10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID))



4 . PATIENT TABLE

QUERY:

CREATE TABLE PATIENT (

P\_ID INT PRIMARY KEY,

P\_NAME VARCHAR2 (255) NOT NULL,

PA\_DATE DATE,

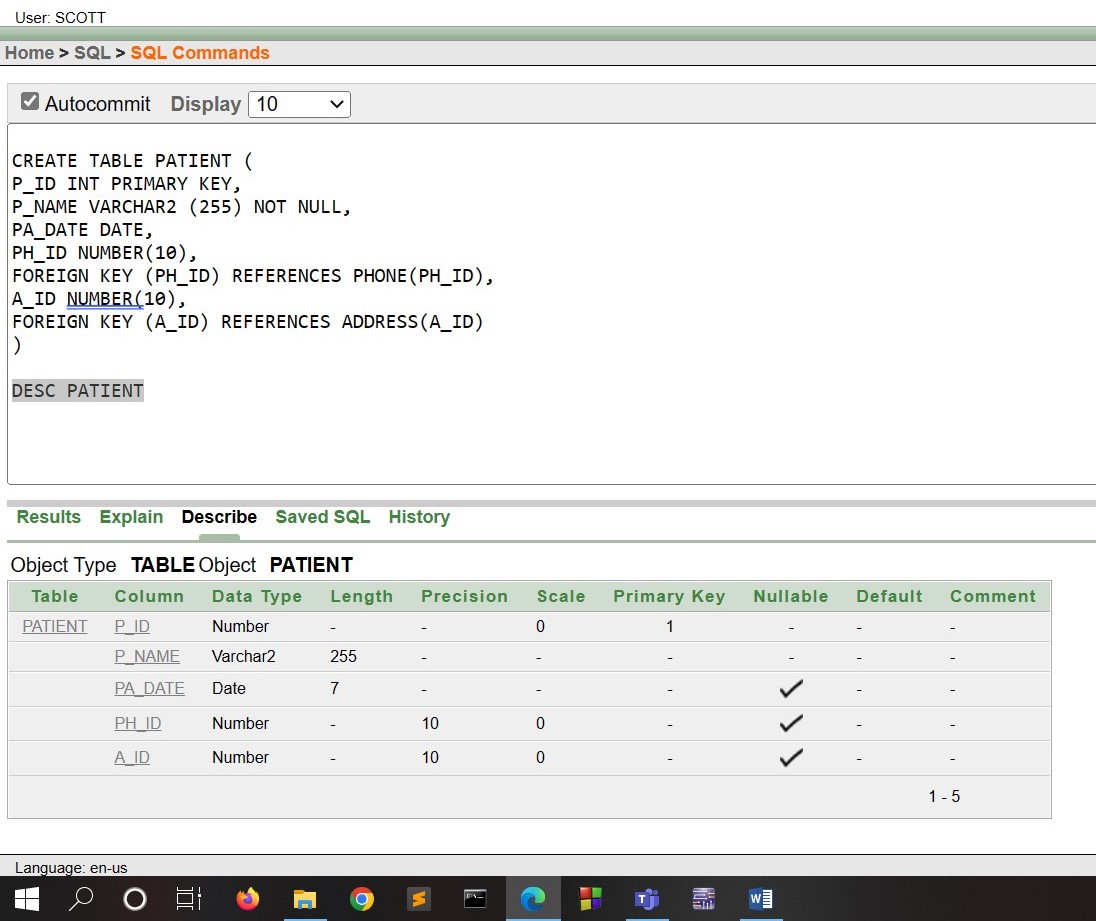
PH\_ID NUMBER(10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

A\_ID NUMBER(10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

)



5 . PATIENT\_BLOOD\_INFO TABLE

QUERY;

CREATE TABLE PATIENT\_BLOOD\_INFO (

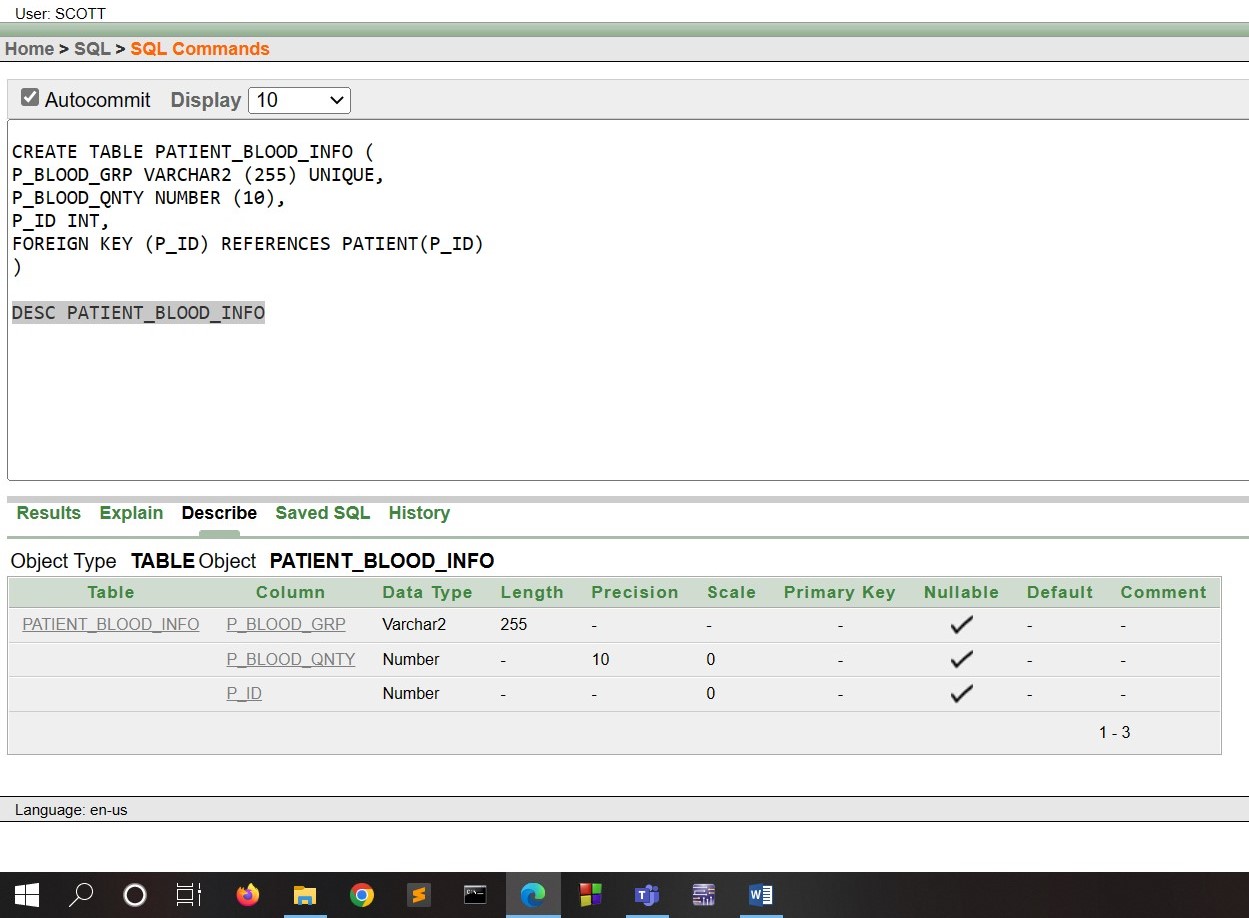
P\_BLOOD\_GRP VARCHAR2 (255) UNIQUE,

P\_BLOOD\_QNTY NUMBER (10),

P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)

)



6 . HOSPITAL TABLE

QUERY:

CREATE TABLE HOSPITAL (

HOSP\_ID NUMBER (10) PRIMARY KEY,

HOSP\_NAME VARCHAR2(255) NOT NULL,

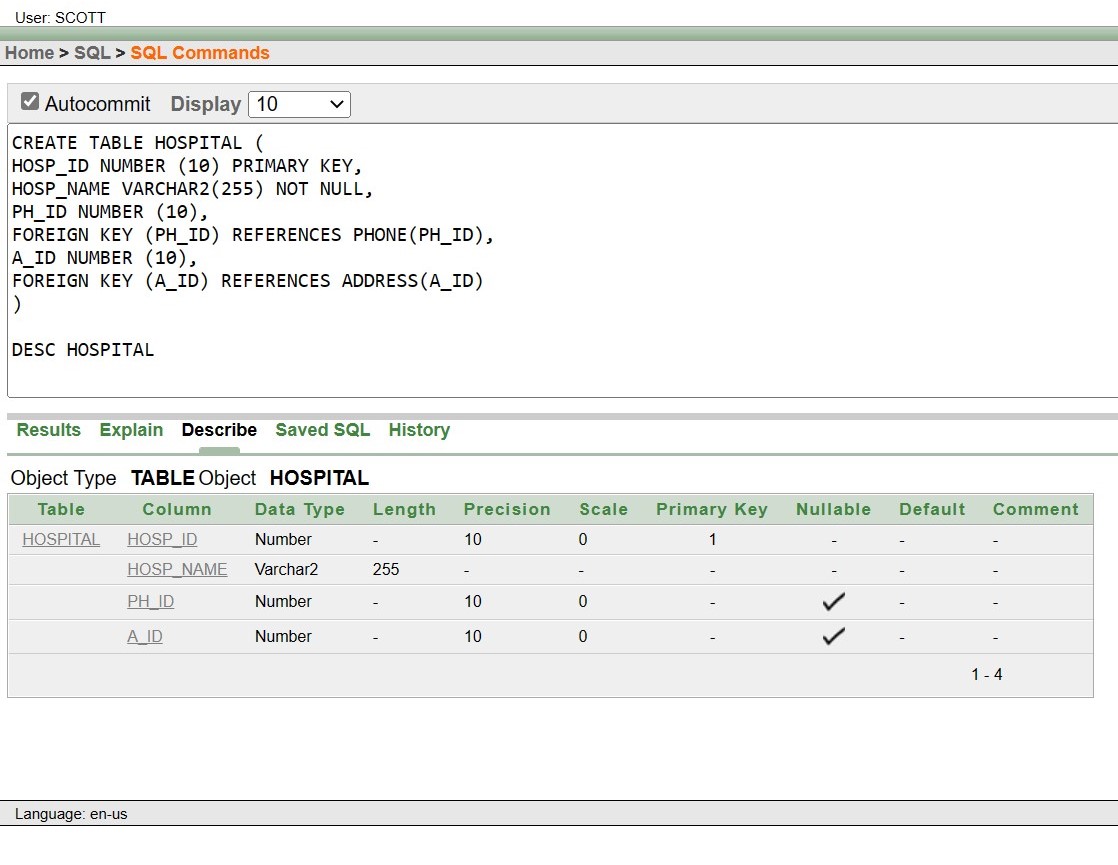
PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

)



7 . H\_NEEDED\_BLOOD TABLE

QUERY:

CREATE TABLE H\_NEEDED\_BLOOD (

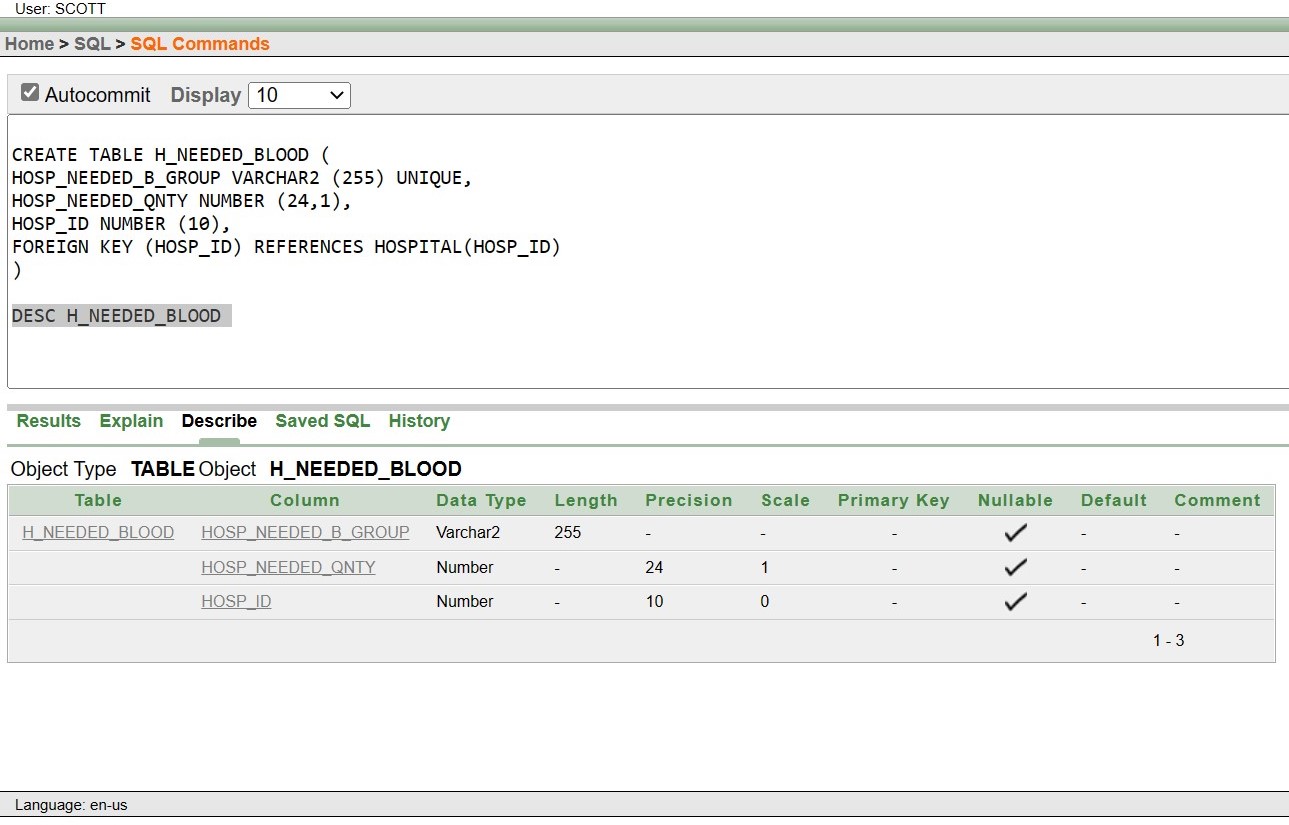
HOSP\_NEEDED\_B\_GROUP VARCHAR2 (255) UNIQUE,

HOSP\_NEEDED\_QNTY NUMBER (24,1),

HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)

)



8 . RECORDING\_STAFF TABLE

QUERY:

CREATE TABLE RECORDING\_STAFF (

R\_ID VARCHAR2(255) PRIMARY KEY,

R\_NAME VARCHAR2 (255) NOT NULL,

PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

BD\_ID INT,

FOREIGN KEY (BD\_ID) REFERENCES DONER(BD\_ID),

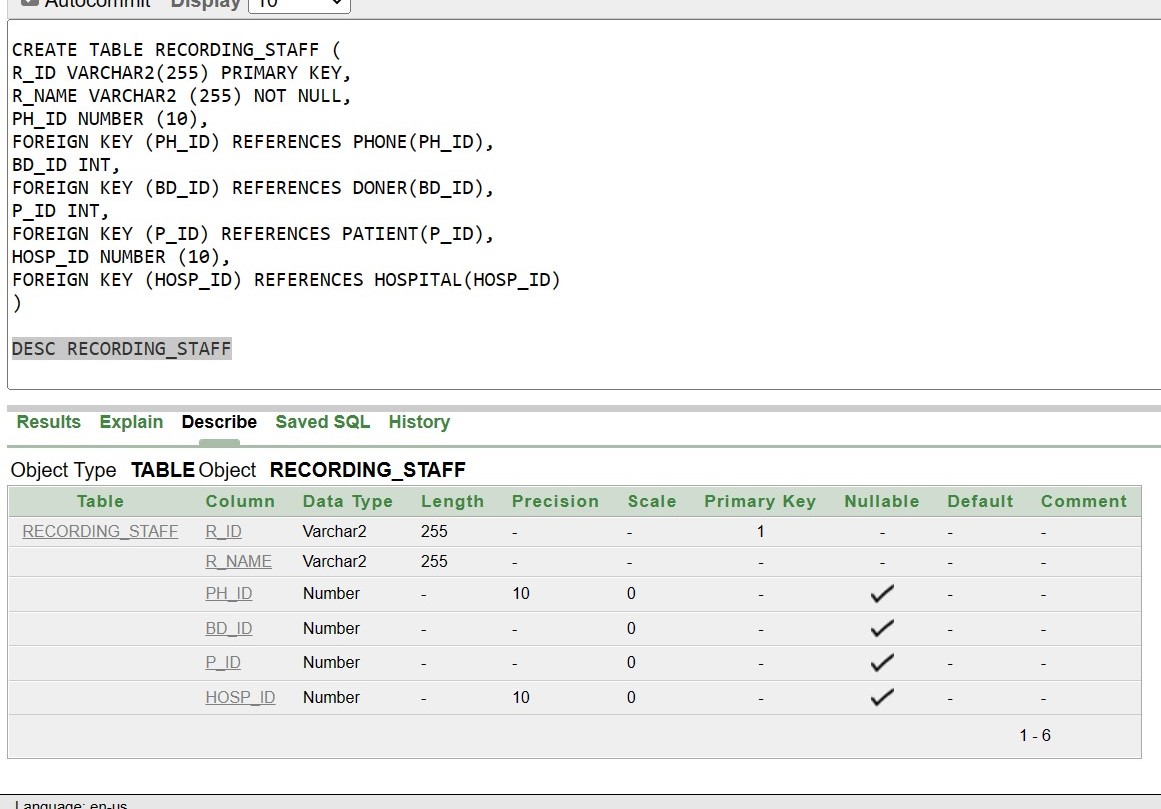
P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID),

HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)

)



9 . DISEASE TABLE

QUERY:

CREATE TABLE DISEASE (

D\_ID VARCHAR2(255) PRIMARY KEY,

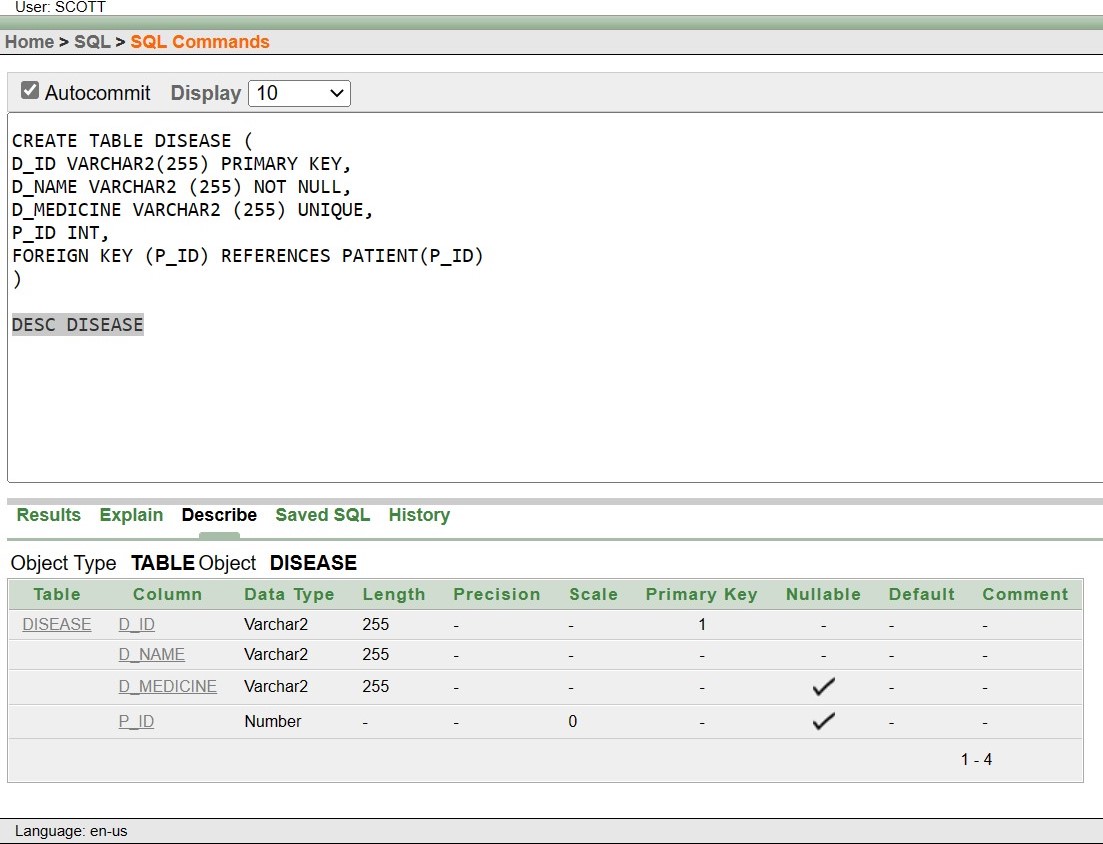
D\_NAME VARCHAR2 (255) NOT NULL,

D\_MEDICINE VARCHAR2 (255) UNIQUE,

P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)

)



10. DOCTOR TABLE

QUERY:

CREATE TABLE DOCTOR (

DR\_ID VARCHAR2 (255) PRIMARY KEY,

DR\_NAME VARCHAR2 (255) NOT NULL,

PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

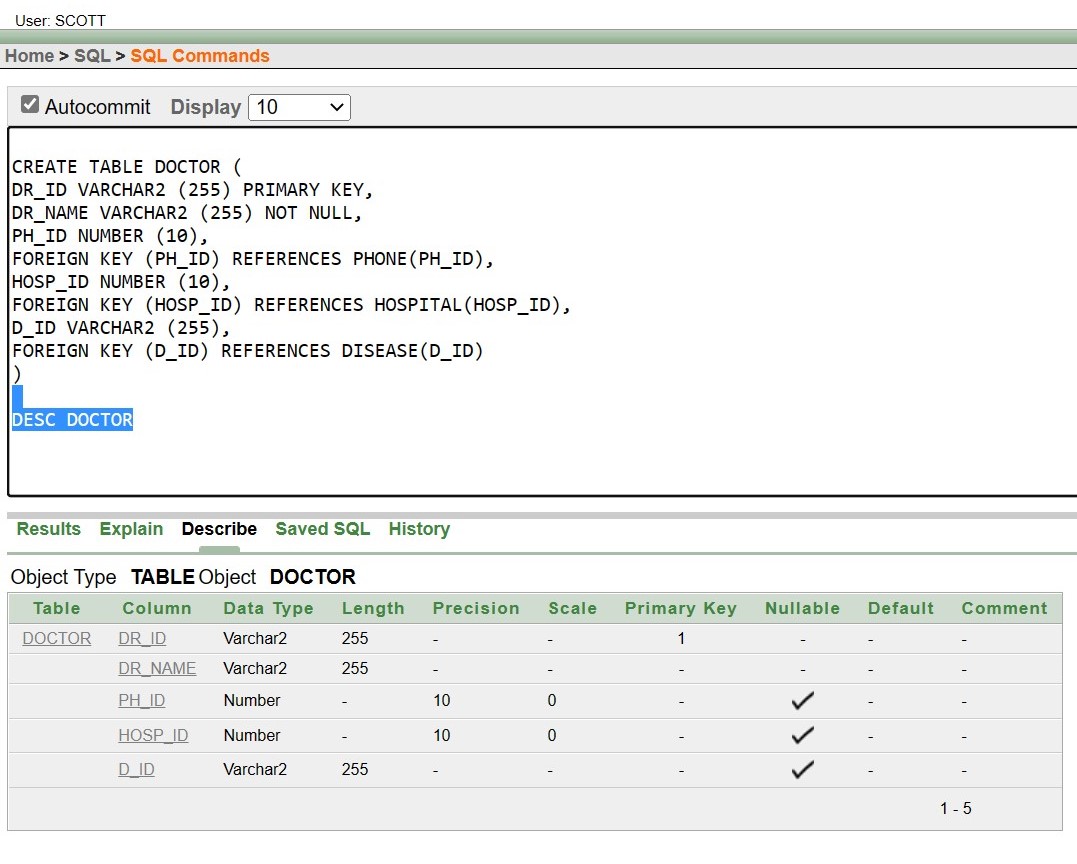
HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID),

D\_ID VARCHAR2 (255),

FOREIGN KEY (D\_ID) REFERENCES DISEASE(D\_ID)

)



11. MANAGER TABLE

QUERY:

CREATE TABLE MANAGER (

M\_ID NUMBER (10) PRIMARY KEY,

M\_NAME VARCHAR2(255) NOT NULL,

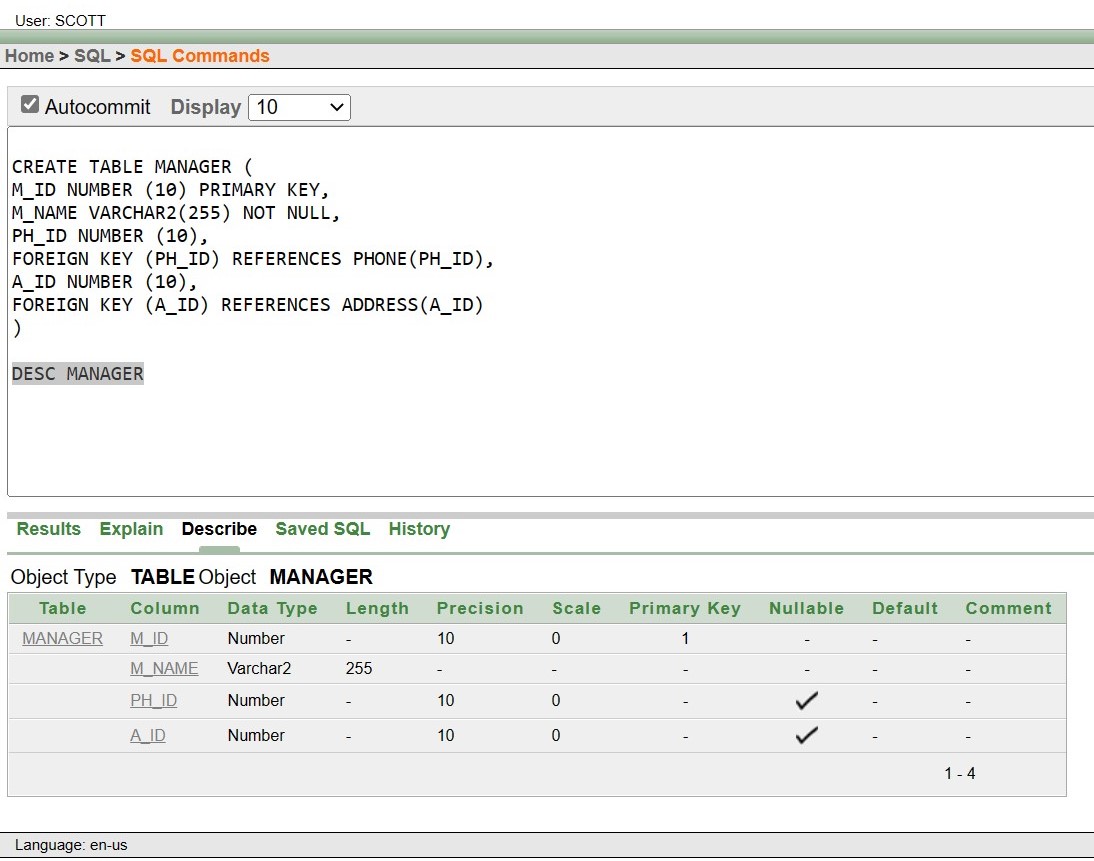
PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

)



12. GIVES\_ORDER TABLE

QUERY:

CREATE TABLE GIVES\_ORDER (

HOSP\_ID NUMBER (10) PRIMARY KEY,

HOSP\_NAME VARCHAR2(255) NOT NULL,

PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

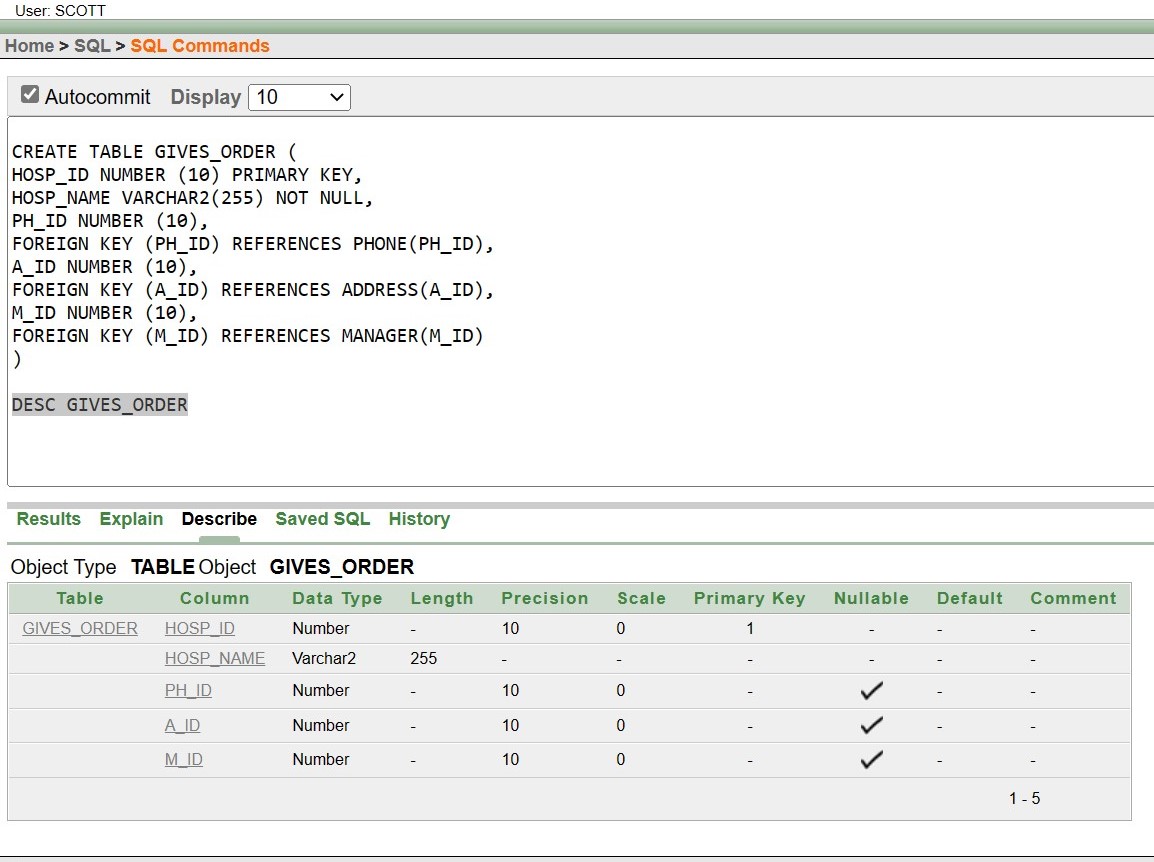
A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID),

M\_ID NUMBER (10),

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)

)



13. BLOOD\_BANK TABLE

QUERY:

CREATE TABLE BLOOD\_BANK (

BB\_ID VARCHAR2 (255) PRIMARY KEY,

BB\_NAME VARCHAR2 (255) NOT NULL,

PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

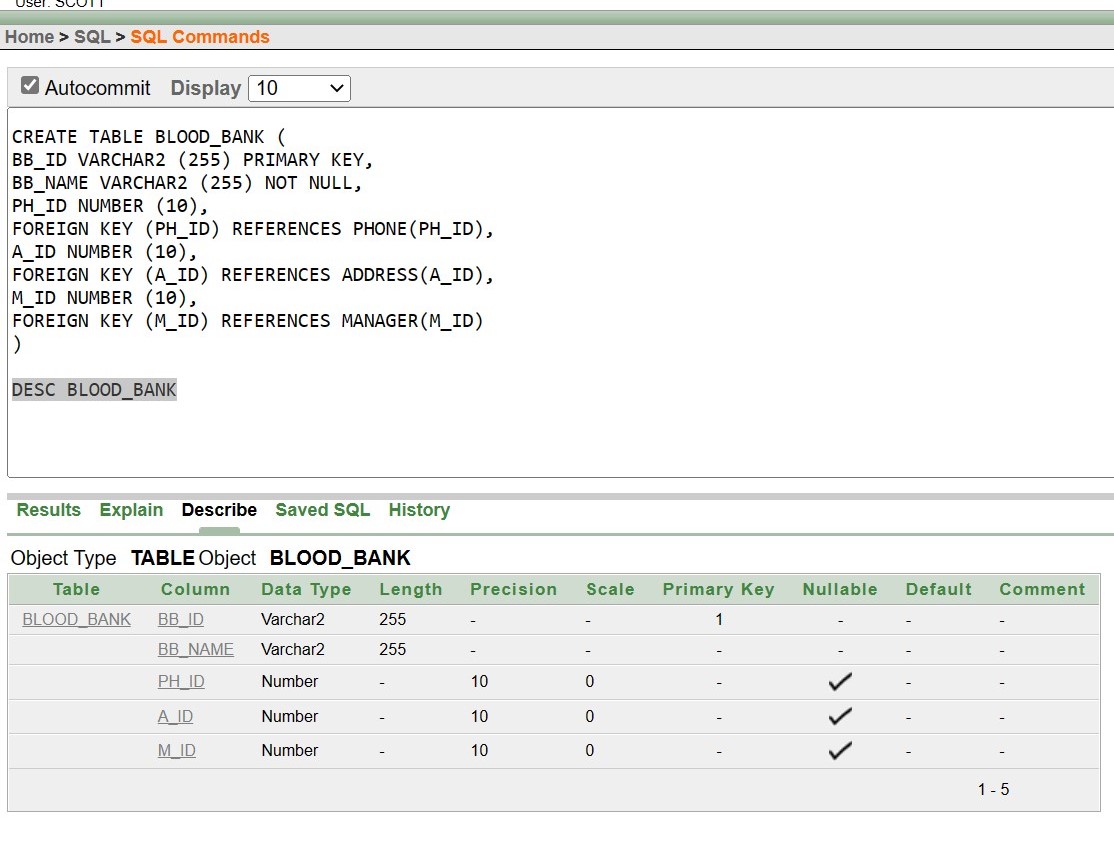
A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID),

M\_ID NUMBER (10),

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)

)



14. DISEASE\_FINDER TABLE

QUERY:

CREATE TABLE DISEASE\_FINDER (

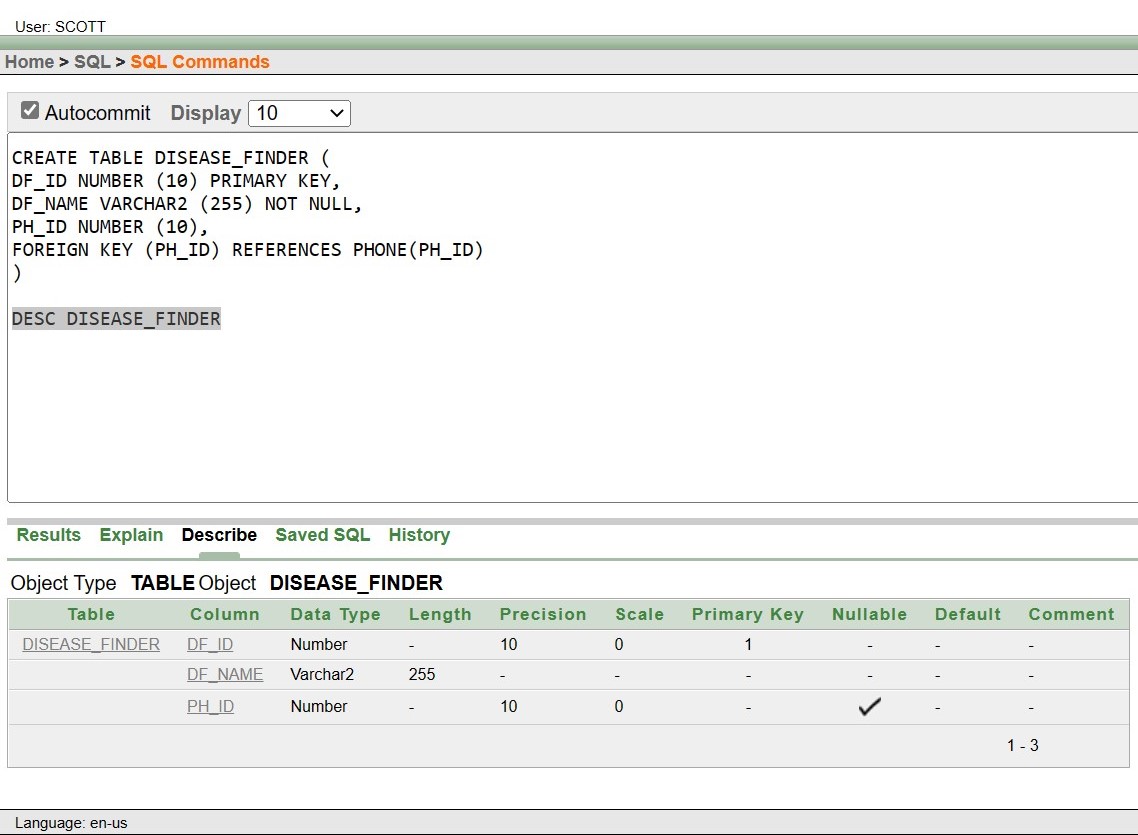
DF\_ID NUMBER (10) PRIMARY KEY,

DF\_NAME VARCHAR2 (255) NOT NULL,

PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

)



15. BLOOD\_GROUP TABLE

QUERY:

CREATE TABLE BLOOD\_GROUP (

B\_NUM NUMBER (10) PRIMARY KEY,

B\_GROUP VARCHAR2 (255) UNIQUE,

STATUS VARCHAR2 (255) DEFAULT 'HEALTHY',

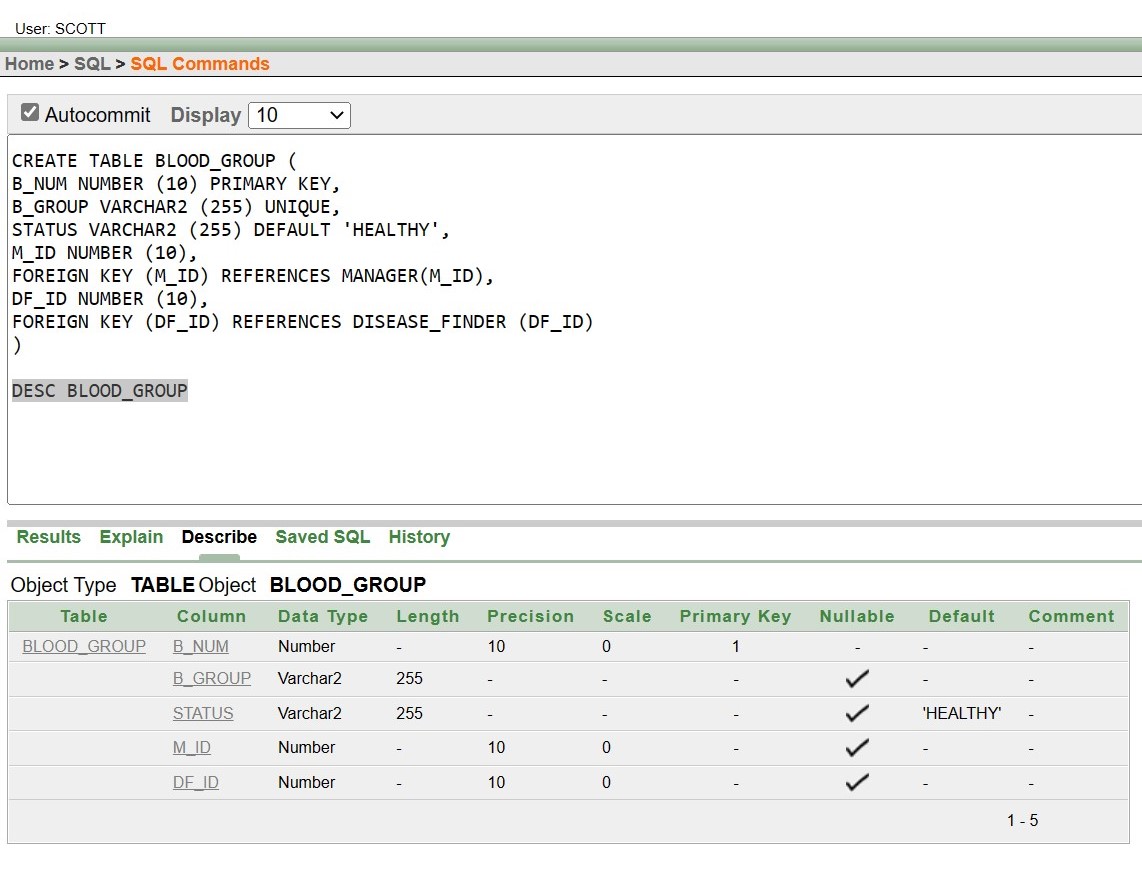
M\_ID NUMBER (10),

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID),

DF\_ID NUMBER (10),

FOREIGN KEY (DF\_ID) REFERENCES DISEASE\_FINDER (DF\_ID)

)



INSERT DATA INTO TABLE:

**1.PHONE**

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(60,'01728890324')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(61,'01531783245')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(62,'01936784328')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(63,'01728890325')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(64,'01728890326')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(65,'01728890327')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(66,'01728890328')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(67,'01728890329')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(68,'01728890330')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(69,'01728890331')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(70,'01728890332')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(71,'01728890333')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(72,'01728890334')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(73,'01728890335')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(74,'01728890336')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(75,'01728890337')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(76,'01728890338')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(77,'01728890339')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(78,'01728890340')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(79,'01728890341')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(80,'01728890342')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(81,'01728890343')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(82,'01728890344')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(83,'01728890345')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(84,'01728890346')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(85,'01728890347')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(87,'01728890348')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(88,'01728890349')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(89,'01728890350')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(90,'01728890351')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(91,'01728890352')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(92,'01728890353')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(93,'01728890354')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(94,'01728890355')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(95,'01728890356')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(96,'01728890357')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(97,'01728890358')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(98,'01728890359')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(99,'01728890360')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(100,'01728890361')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(101,'01728890362')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(102,'01728890363')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(103,'01728890364')

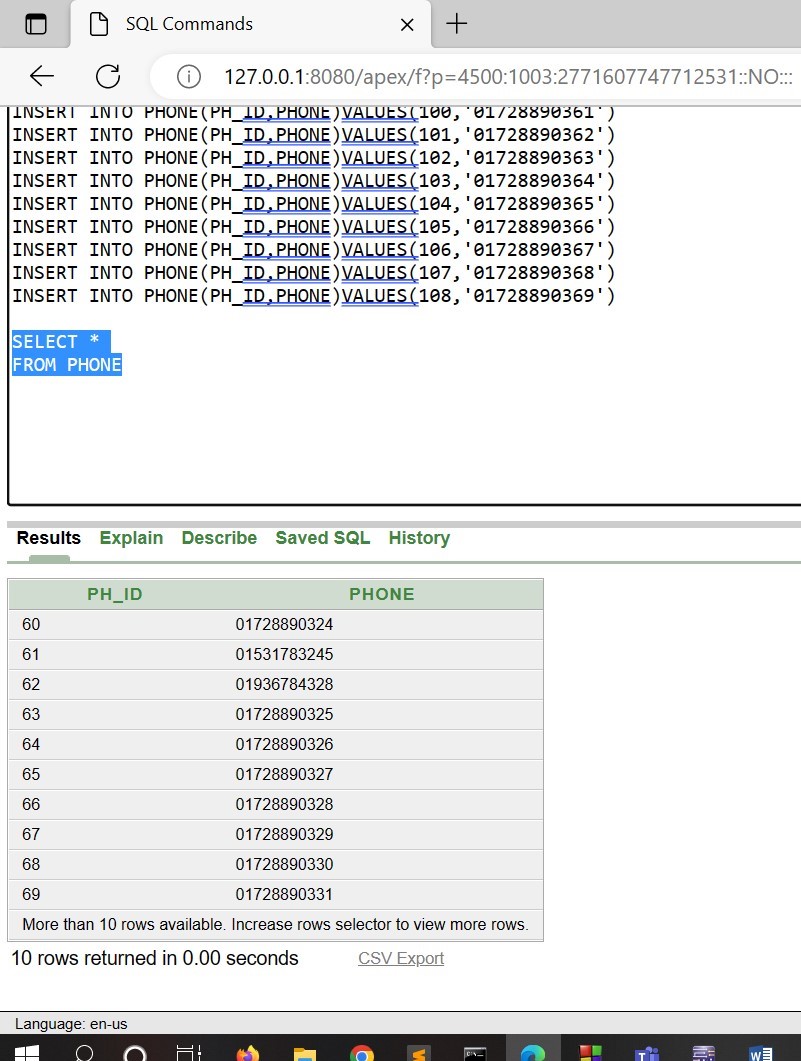
INSERT INTO PHONE(PH\_ID,PHONE)VALUES(104,'01728890365')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(105,'01728890366')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(106,'01728890367')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(107,'01728890368')

INSERT INTO PHONE(PH\_ID,PHONE)VALUES(108,'01728890369')



**2. ADDRESS:**

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223346,'DHAKA','BELLY ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223347,'DHAKA','OSMANI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223348,'DHAKA','KURIL ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223350,'DHAKA','KURATOLI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223351,'DHAKA','KALI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223352,'DHAKA','NSU ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223353,'DHAKA','NOBI NOGOR ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223354,'DHAKA','AIRPORT ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223355,'DHAKA','MUDDHA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223356,'DHAKA','PURAN DHAKA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223357,'BARISAL','BANGLA BAZAR ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223358,'DHAKA','KATPOTTI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223359,'DHAKA','AMTOLA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223360,'DHAKA','MODEL ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223362,'DHAKA','TALTOLI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223363,'DHAKA','HANGA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223364,'DHAKA','HATI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223367,'DHAKA','AQURIAM ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223368,'DHAKA','BELLY ROAD3')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223369,'KHULNA','FOLPOTTI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223370,'KHULNA','KAUNIA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223371,'KHULNA','BM COLLEGE ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223372,'KHULNA','BM SCHOOL ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223373,'KHULNA','AMTOLI ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223374,'KHULNA','BAGURA ROAD')

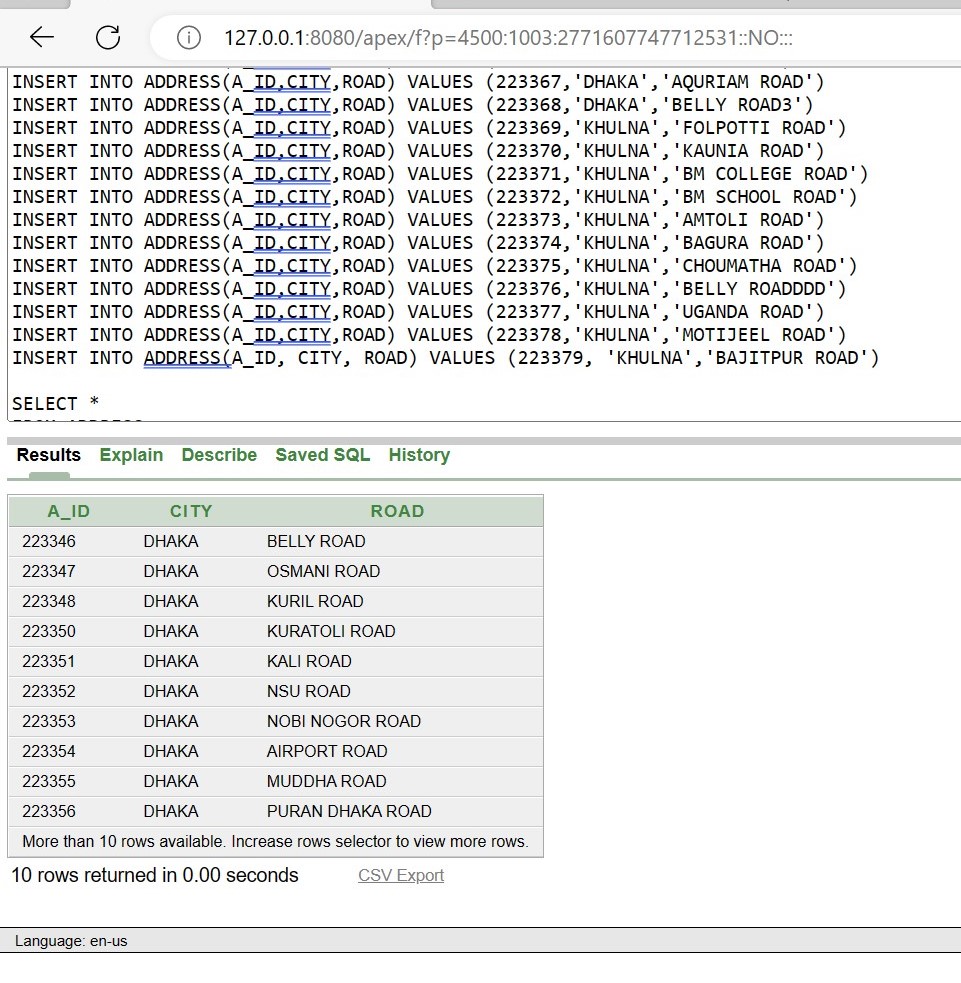
INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223375,'KHULNA','CHOUMATHA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223376,'KHULNA','BELLY ROADDDD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223377,'KHULNA','UGANDA ROAD')

INSERT INTO ADDRESS(A\_ID,CITY,ROAD) VALUES (223378,'KHULNA','MOTIJEEL ROAD')

INSERT INTO ADDRESS(A\_ID, CITY, ROAD) VALUES (223379, 'KHULNA','BAJITPUR ROAD')



3. DONER:

INSERT INTO

DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(1,'SHAMIM',47,'B+','17-FEB-2022','MALE',60,223346)

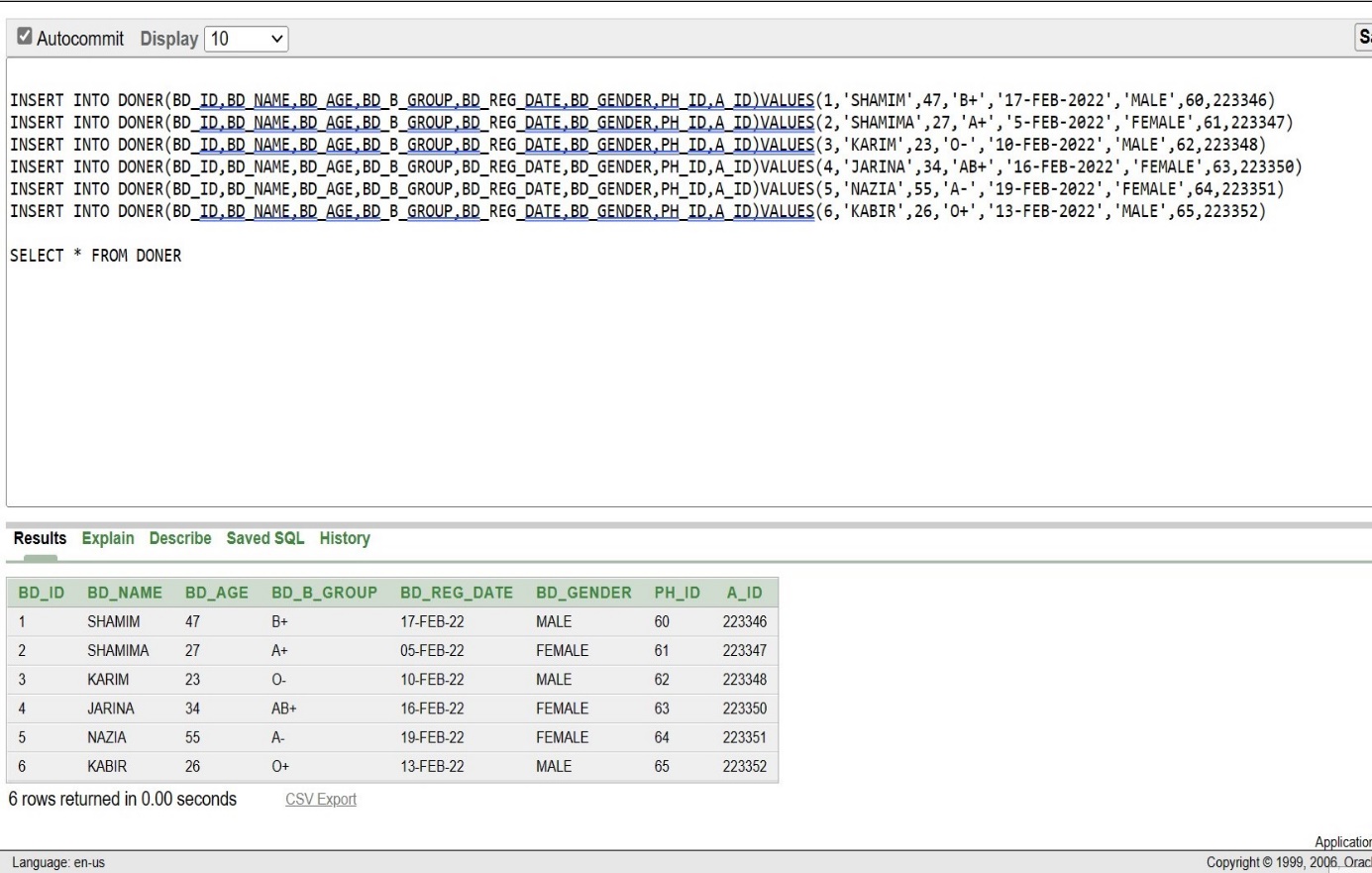
INSERT INTO DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(2,'SHAMIMA',27,'A+','5-FEB-2022','FEMALE',61,223347)

INSERT INTO DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(3,'KARIM',23,'O-','10-FEB-2022','MALE',62,223348)

INSERT INTO DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(4,'JARINA',34,'AB+','16-FEB-2022','FEMALE',63,223350)

INSERT INTO DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(5,'NAZIA',55,'A-','19-FEB-2022','FEMALE',64,223351)

INSERT INTO DONER(BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP,BD\_REG\_DATE,BD\_GENDER,PH\_ID,A\_ID)VALUES(6,'KABIR',26,'O+','13-FEB-2022','MALE',65,223352)



4. PATIENT:

INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (221,'SAKIB','22-JAN-22',66, 223353)

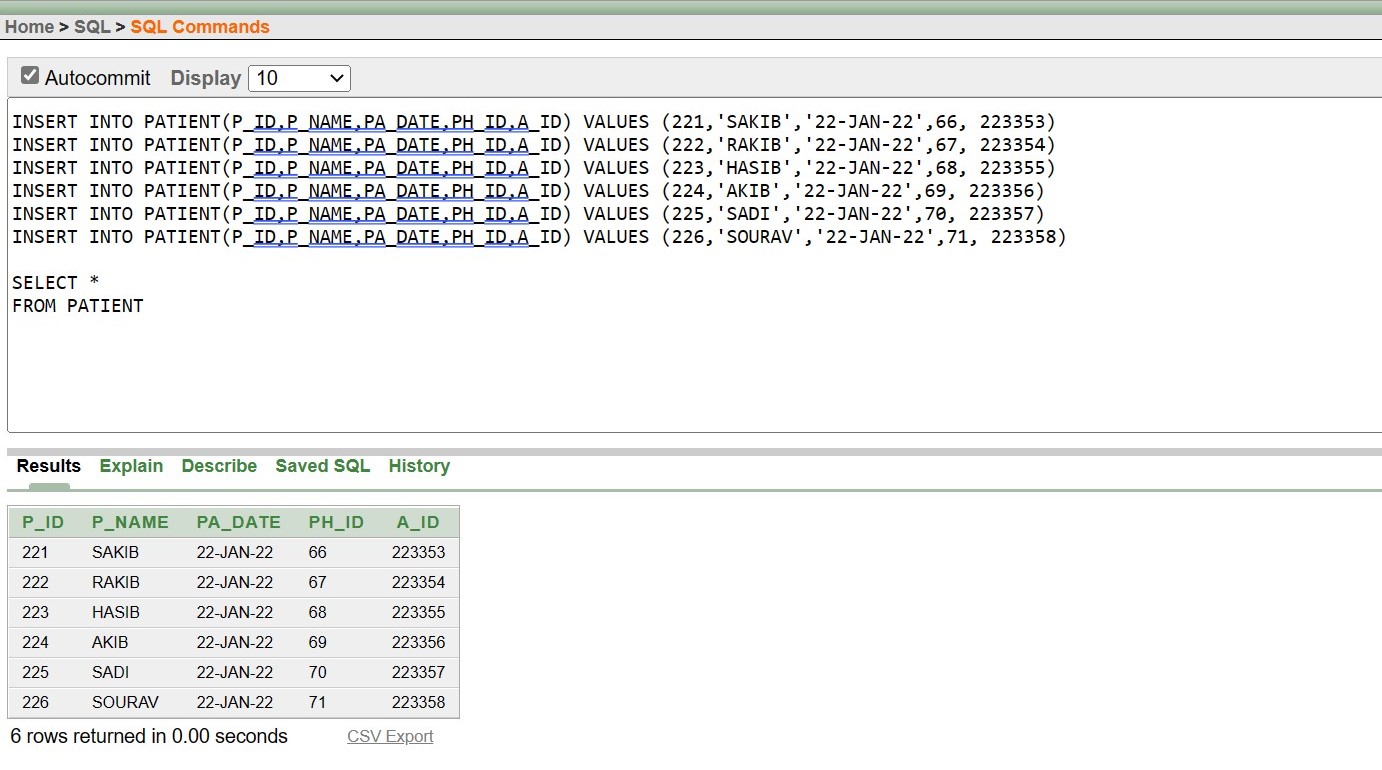
INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (222,'RAKIB','22-JAN-22',67, 223354)

INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (223,'HASIB','22-JAN-22',68, 223355)

INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (224,'AKIB','22-JAN-22',69, 223356)

INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (225,'SADI','22-JAN-22',70, 223357)

INSERT INTO PATIENT(P\_ID,P\_NAME,PA\_DATE,PH\_ID,A\_ID) VALUES (226,'SOURAV','22-JAN-22',71, 223358)



5. PATIENT\_BLOOD\_INFO:

INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('O+',2,221)

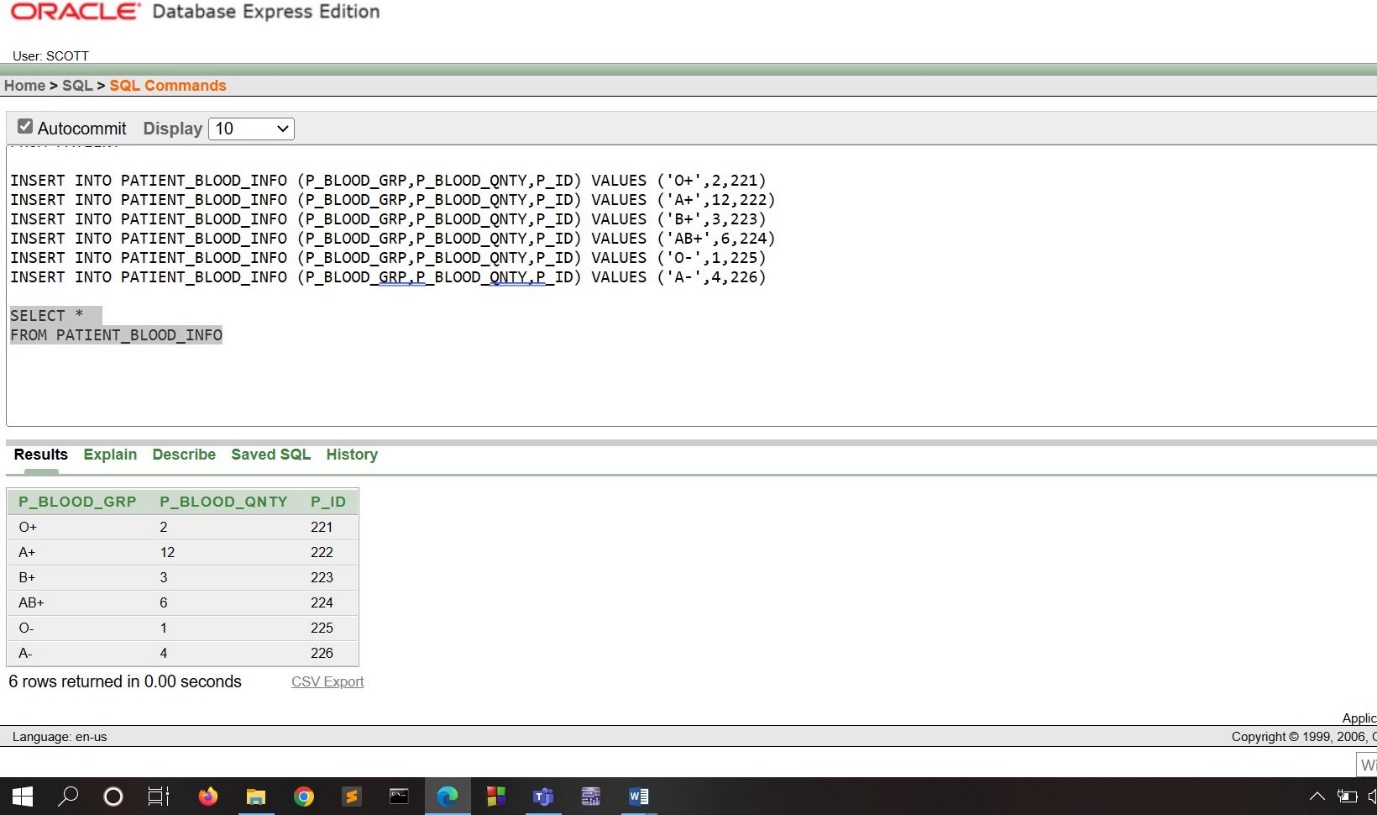
INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('A+',12,222)

INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('B+',3,223)

INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('AB+',6,224)

INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('O-',1,225)

INSERT INTO PATIENT\_BLOOD\_INFO (P\_BLOOD\_GRP,P\_BLOOD\_QNTY,P\_ID) VALUES ('A-',4,226)



6. HOSPITAL:

INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(1,'KURMITOLA A GENERAL HOSPITAL',72,223359)

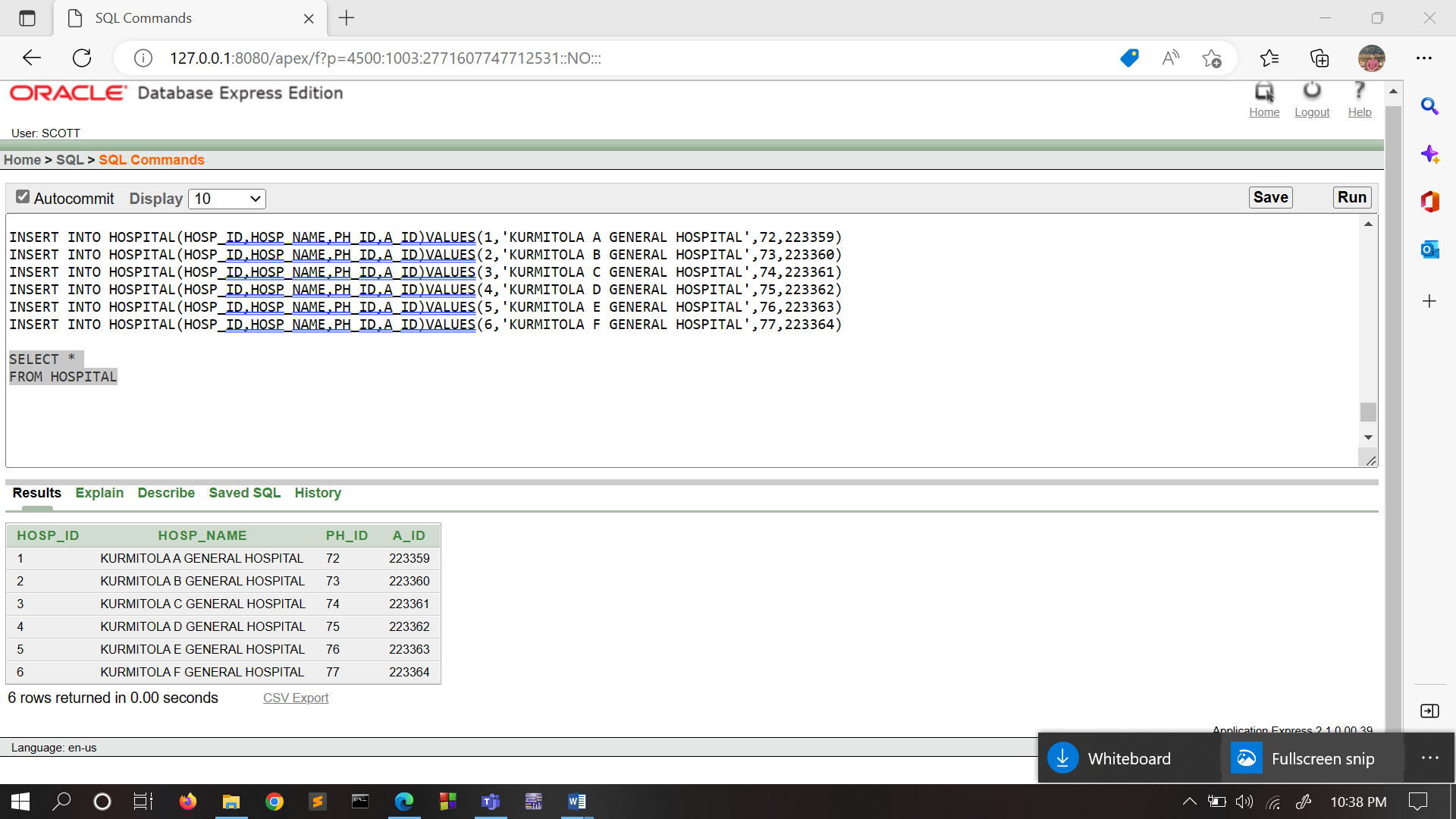
INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(2,'KURMITOLA B GENERAL HOSPITAL',73,223360)

INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(3,'KURMITOLA C GENERAL HOSPITAL',74,223361)

INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(4,'KURMITOLA D GENERAL HOSPITAL',75,223362)

INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(5,'KURMITOLA E GENERAL HOSPITAL',76,223363)

INSERT INTO HOSPITAL(HOSP\_ID,HOSP\_NAME,PH\_ID,A\_ID)VALUES(6,'KURMITOLA F GENERAL HOSPITAL',77,223364)

****

7. H-NEEDED-BLOOD:

INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('AB-',2,1)

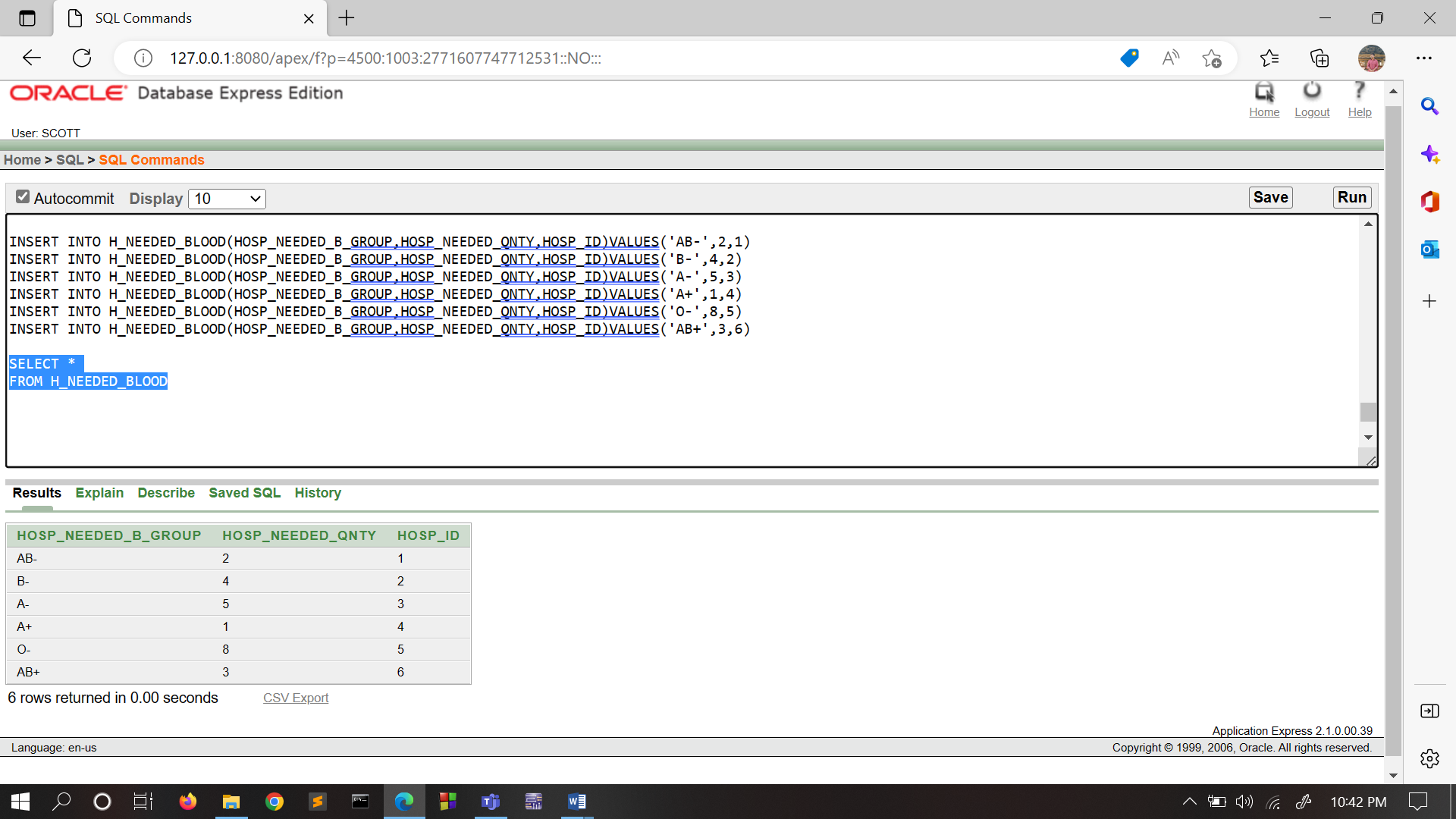
INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('B-',4,2)

INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('A-',5,3)

INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('A+',1,4)

INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('O-',8,5)

INSERT INTO H\_NEEDED\_BLOOD(HOSP\_NEEDED\_B\_GROUP,HOSP\_NEEDED\_QNTY,HOSP\_ID)VALUES('AB+',3,6)



• The relationship with Blood Bank manager and Blood Specimen is 1

to many. That’s why primary

key of Blood Bank manager is used as a foreign key in Blood Specimen

.

Disease Finder Table:

• The relationship with Disease finder and Blood Specimen is of 1 to

many. Therefore, the primary key of Disease finder is used as a foreign

key in Blood Specimen.

Blood Bank Manager Table:

• The relationship between Blood Bank Manager and Blood Specimen,

Recipient, Hospital info are all of 1 to many. So therefore, the primary

key of Blood Bank Manager is used as a foreign key in Blood Specimen,

Recipient and Hospital info.

Hospital info Table:

• The relationship with City and Hospital info is 1 to many. That’s why

primary key of City is used as a foreign key in Hospital info.

• The relationship with Blood Bank Manager and Hospital info is 1 to

many. That’s why primary key of Blood Bank manager is used as a

foreign key in Hospital info.

8. RECORDING\_STAFF:

INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-31','KUDDUS',78,1,221,1)

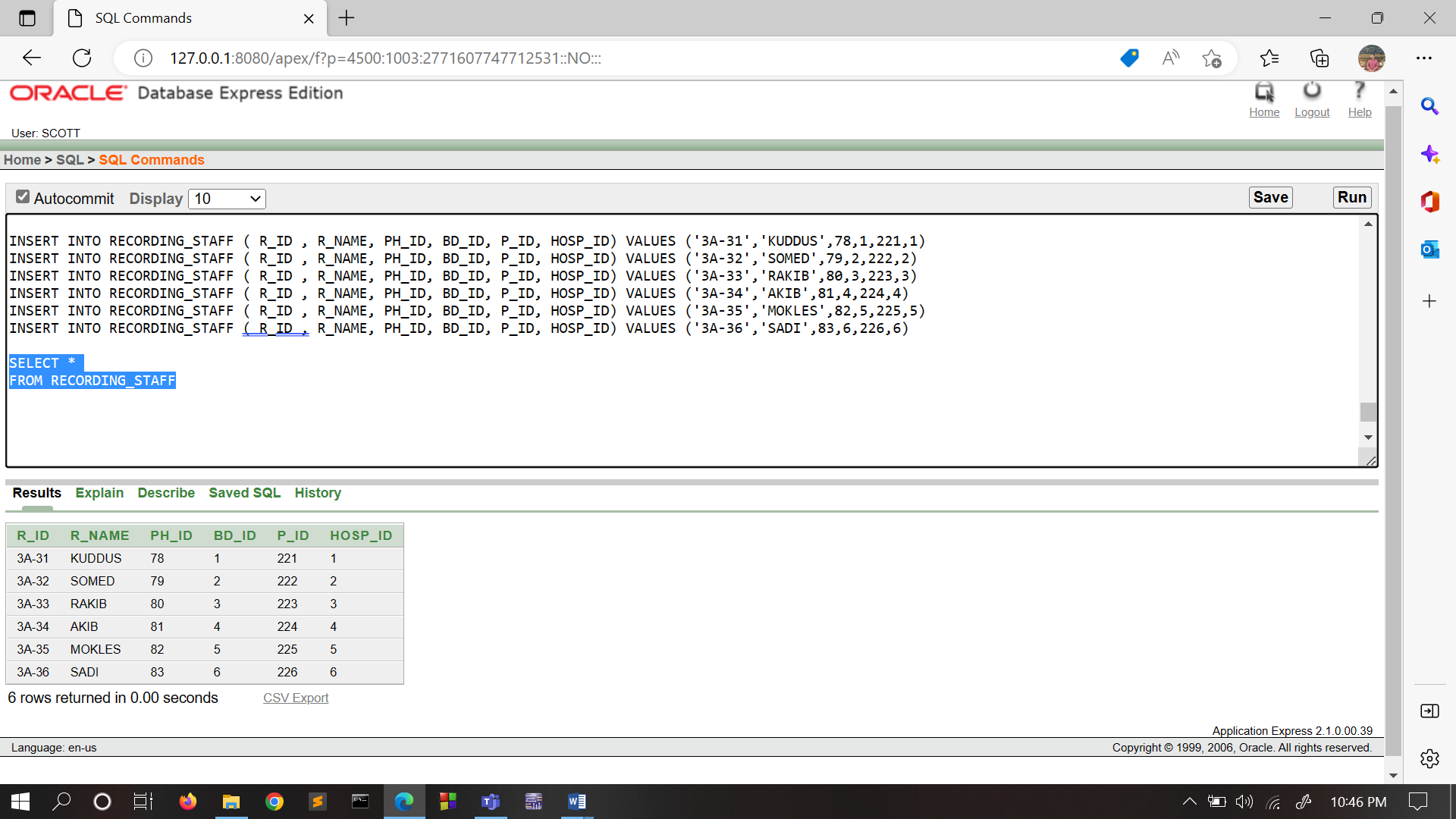
INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-32','SOMED',79,2,222,2)

INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-33','RAKIB',80,3,223,3)

INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-34','AKIB',81,4,224,4)

INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-35','MOKLES',82,5,225,5)

INSERT INTO RECORDING\_STAFF ( R\_ID , R\_NAME, PH\_ID, BD\_ID, P\_ID, HOSP\_ID) VALUES ('3A-36','SADI',83,6,226,6)



9. DISEASE:

INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-01','TYPHOID','Ciprofloxacin',221)

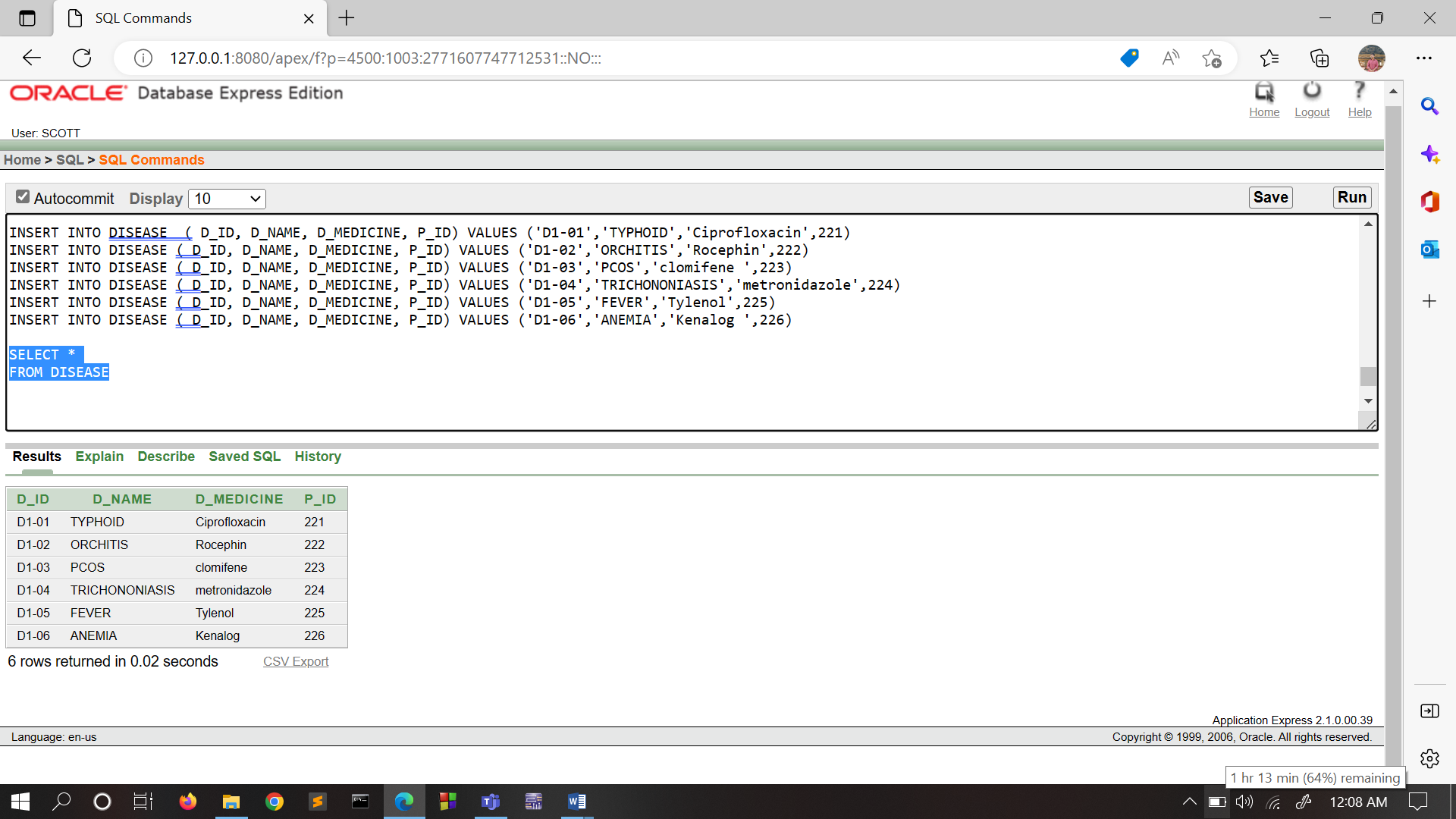
INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-02','ORCHITIS','Rocephin',222)

INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-03','PCOS','clomifene ',223)

INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-04','TRICHONONIASIS','metronidazole',224)

INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-05','FEVER','Tylenol',225)

INSERT INTO DISEASE ( D\_ID, D\_NAME, D\_MEDICINE, P\_ID) VALUES ('D1-06','ANEMIA','Kenalog ',226)



10. DOCTOR:

INSERT INTO DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(1,'KAMRUZZAMAN',108,1,'D1-01')

INSERT INTO DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(2,'ATAUR',107,2,'D1-02')

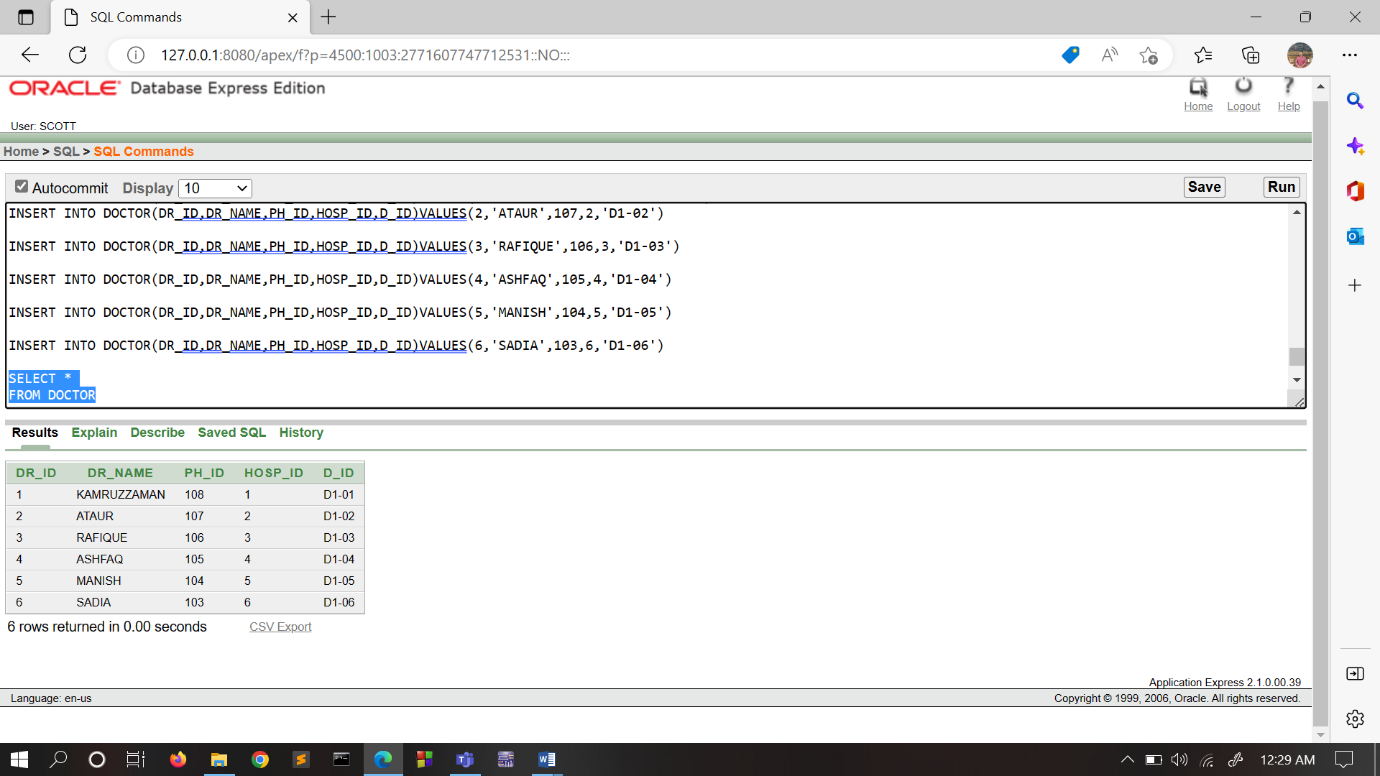
INSERT INTO DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(3,'RAFIQUE',106,3,'D1-03')

INSERT INTO

DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(4,'ASHFAQ',105,4,'D1-04')

INSERT INTO DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(5,'MANISH',104,5,'D1-05')

INSERT INTO DOCTOR(DR\_ID,DR\_NAME,PH\_ID,HOSP\_ID,D\_ID)VALUES(6,'SADIA',103,6,'D1-06')



11. MANGER:

INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (1, 'RAFIQUL', 84, 223367)

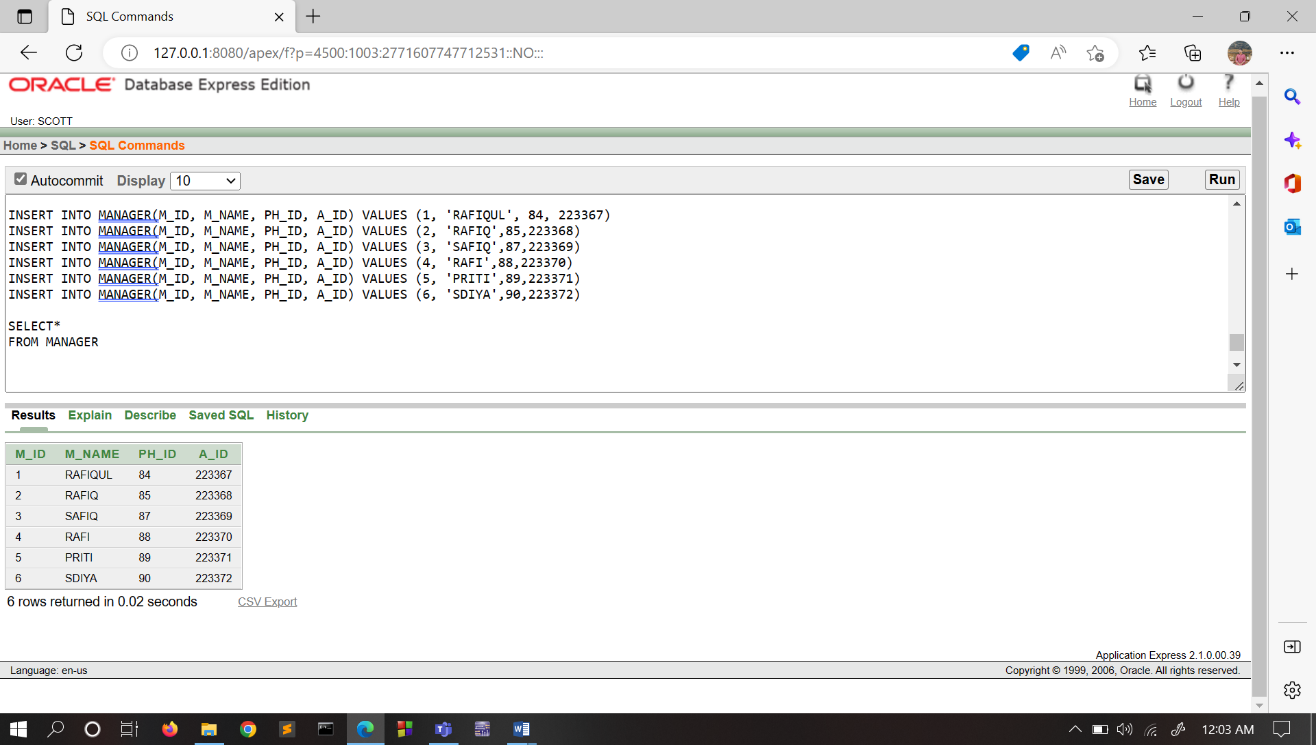
INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (2, 'RAFIQ',85,223368)

INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (3, 'SAFIQ',87,223369)

INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (4, 'RAFI',88,223370)

INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (5, 'PRITI',89,223371)

INSERT INTO MANAGER(M\_ID, M\_NAME, PH\_ID, A\_ID) VALUES (6, 'SDIYA',90,223372)



12. GIVES\_ORDER\_TO

INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (2,'KURMITOLA B GENERAL HOSPITAL',91,223360,1)

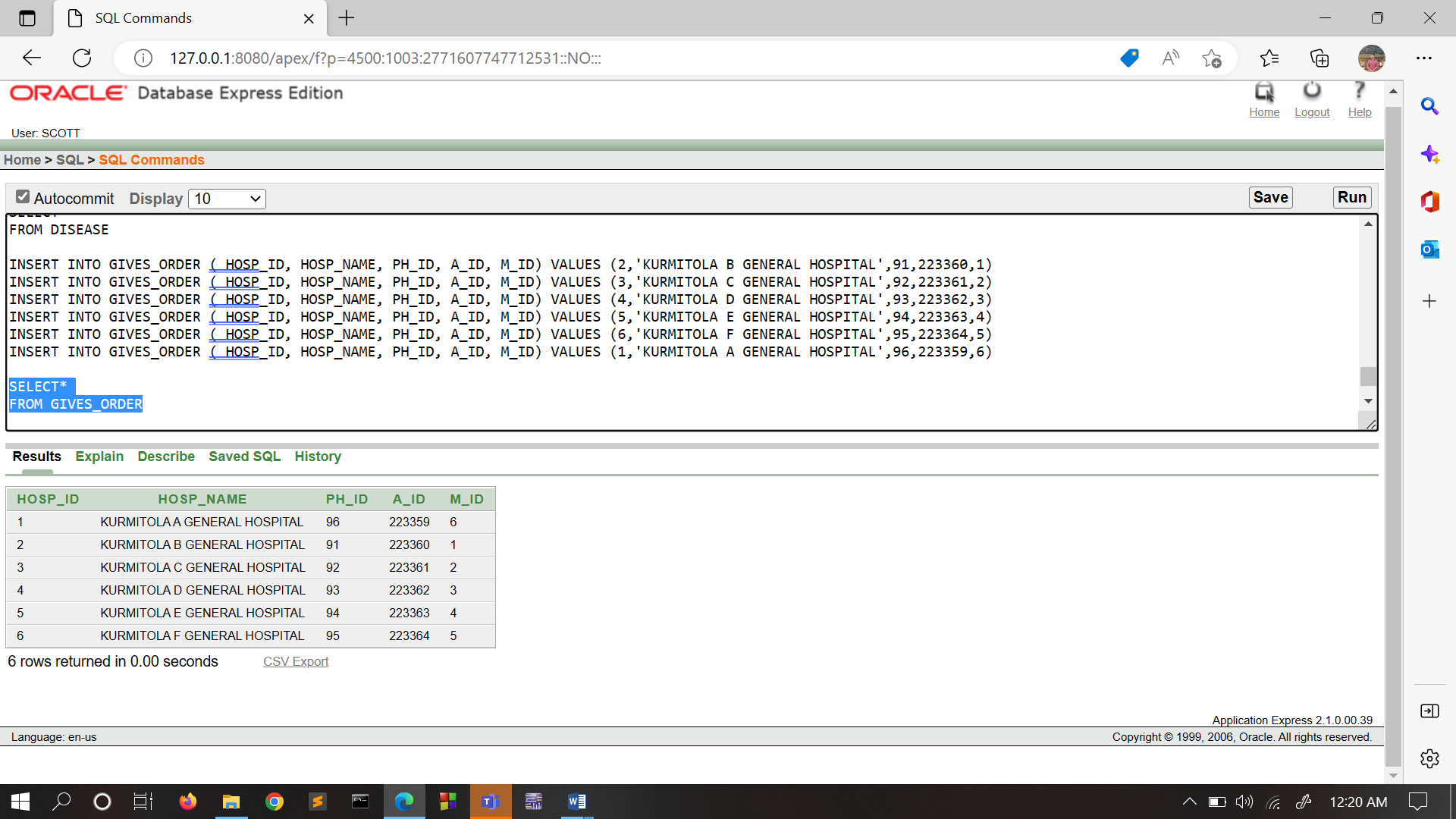
INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (3,'KURMITOLA C GENERAL HOSPITAL',92,223361,2)

INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (4,'KURMITOLA D GENERAL HOSPITAL',93,223362,3)

INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (5,'KURMITOLA E GENERAL HOSPITAL',94,223363,4)

INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (6,'KURMITOLA F GENERAL HOSPITAL',95,223364,5)

INSERT INTO GIVES\_ORDER ( HOSP\_ID, HOSP\_NAME, PH\_ID, A\_ID, M\_ID) VALUES (1,'KURMITOLA A GENERAL HOSPITAL',96,223359,6)



13. BLOOD BANK

INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (1,'SHONDHANI 1',102,223378,1)

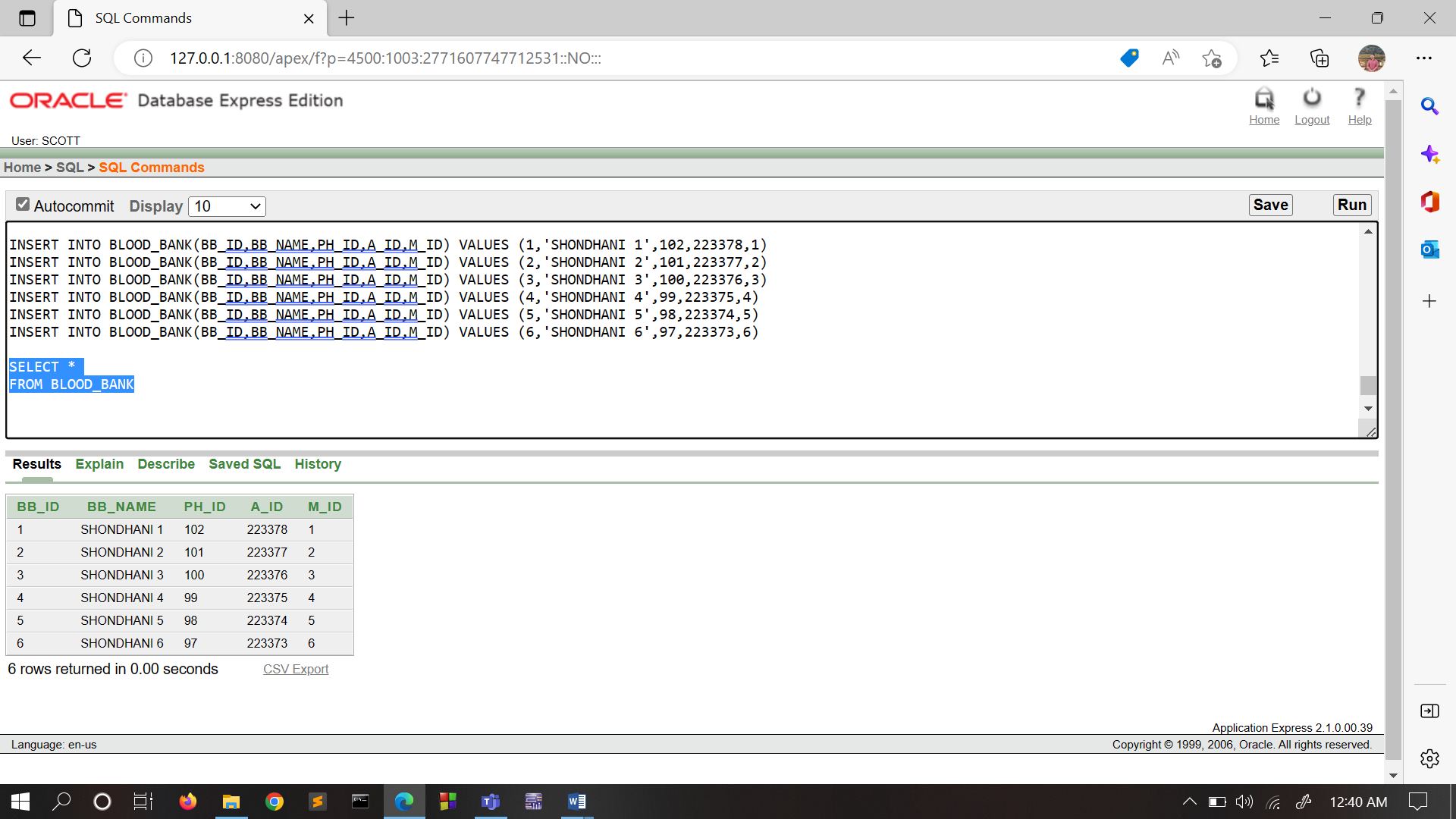
INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (2,'SHONDHANI 2',101,223377,2)

INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (3,'SHONDHANI 3',100,223376,3)

INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (4,'SHONDHANI 4',99,223375,4)

INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (5,'SHONDHANI 5',98,223374,5)

INSERT INTO BLOOD\_BANK(BB\_ID,BB\_NAME,PH\_ID,A\_ID,M\_ID) VALUES (6,'SHONDHANI 6',97,223373,6)



14. BLOOD FINDER

INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (812,'SAKIB',97)

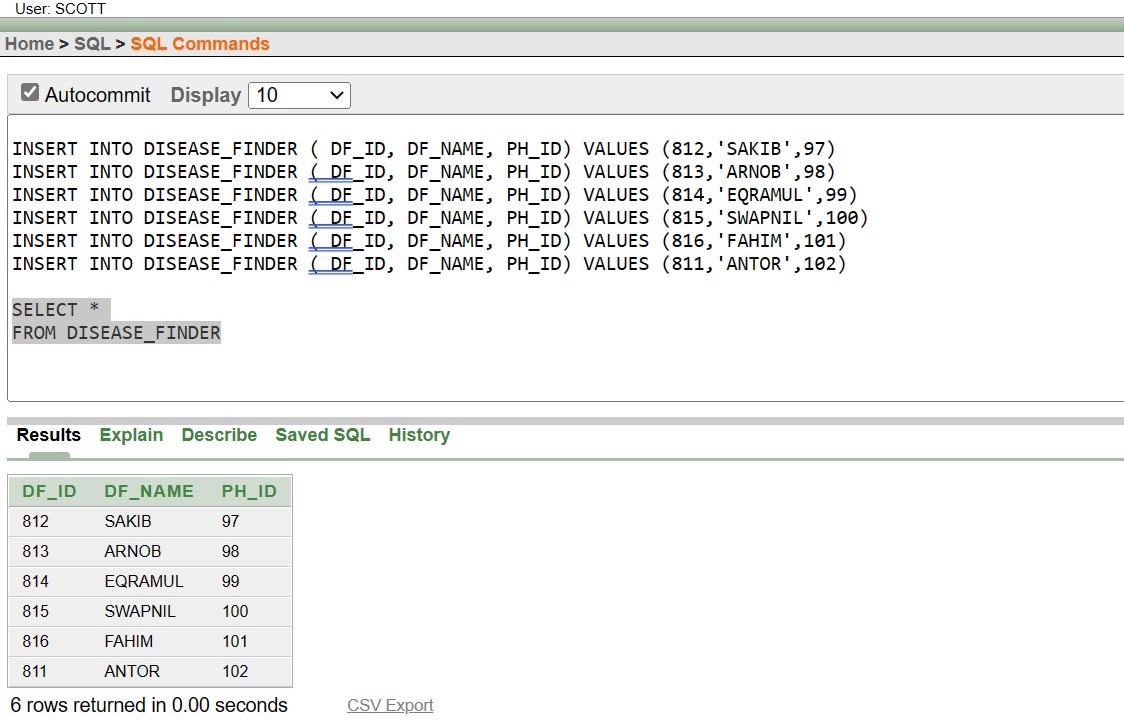
INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (813,'ARNOB',98)

INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (814,'EQRAMUL',99)

INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (815,'SWAPNIL',100)

INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (816,'FAHIM',101)

INSERT INTO DISEASE\_FINDER ( DF\_ID, DF\_NAME, PH\_ID) VALUES (811,'ANTOR',102)



15. BLOOD GROUP

INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,M\_ID,DF\_ID)VALUES(1,'A-',1,812)

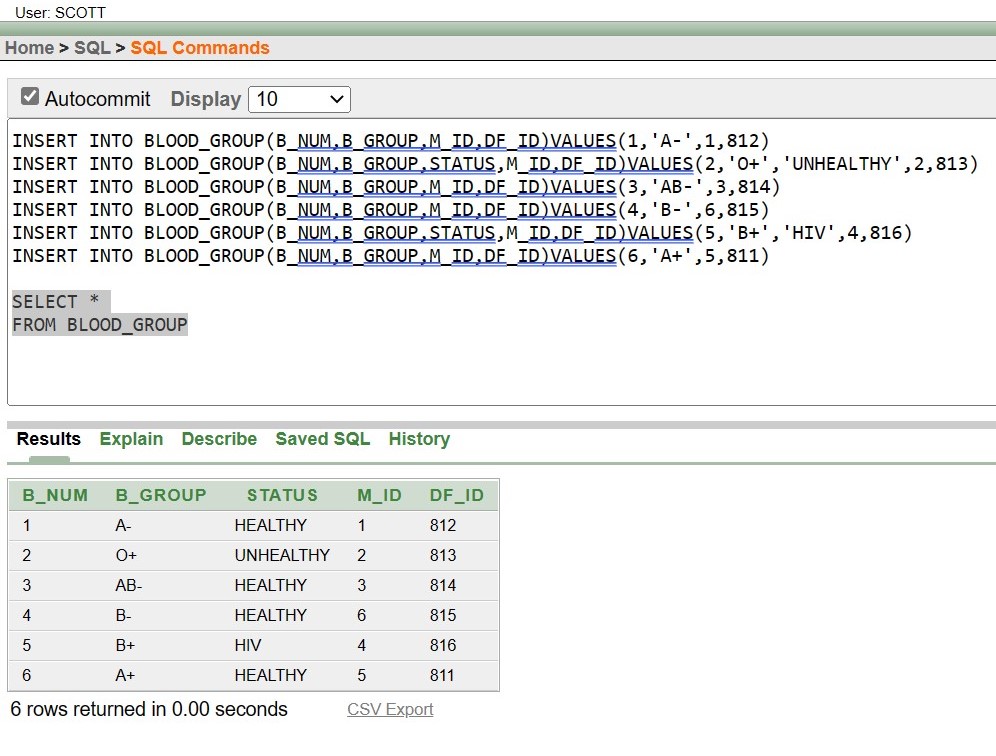
INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,STATUS,M\_ID,DF\_ID)VALUES(2,'O+','UNHEALTHY',2,813)

INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,M\_ID,DF\_ID)VALUES(3,'AB-',3,814)

INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,M\_ID,DF\_ID)VALUES(4,'B-',6,815)

INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,STATUS,M\_ID,DF\_ID)VALUES(5,'B+','HIV',4,816)

INSERT INTO BLOOD\_GROUP(B\_NUM,B\_GROUP,M\_ID,DF\_ID)VALUES(6,'A+',5,811)



**CONSTRAINTS**

**1.PHONE**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| PH\_ID | NUMBER (10) | PRIMARY KEY |
| PHONE | VARCHAR2(255) | NOT NULL |

**QUERY:**

1. PH\_ID NUMBER (10) PRIMARY KEY
2. PHONE VARCHAR2(255) NOT NULL

**2.ADDRESS**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| A\_ID | NUMBER (10) | PRIMARY KEY |
| CITY | VARCHAR2(255) | NOT NULL |
| ROAD | VARCHAR2(255) | UNIQUE |

**QUERY:**

1. A\_ID NUMBER (10) PRIMARY KEY
2. CITY VARCHAR2 (255) NOT NULL
3. ROAD VARCHAR2(255) UNIQUE

**3.DONAR**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| BD\_ID | INT | PRIMARY KEY |
| BD\_NAME | VARCHAR2 (255) | NOT NULL |
| BD\_AGE | NUMBER(10) | CHECK(BD\_AGE>18) |
| BD\_B\_GROUP | VARCHAR2(255) | UNIQUE |
| BD\_REG\_DATE | DATE |  |
| BD\_GENDER | VARCHAR2(255) |  |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |

**QUERY:**

1. BD\_ID INT PRIMARY KEY
2. BD\_NAME VARCHAR2 (255) NOT NULL
3. BD\_AGE NUMBER (10) CHECK (BD\_AGE>18)
4. BD\_B\_GROUP VARCHAR2 (255) UNIQUE
5. PH\_ID NUMBER(10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),

1. A\_ID NUMBER(10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

**4.PATIENT**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| P\_ID | INT | PRIMARY KEY |
| P\_NAME | VARCHAR2 (255) | NOT NULL |
| PA\_DATE | DATE |  |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |

**QUERY:**

1. P\_ID INT PRIMARY KEY
2. P\_NAME VARCHAR2 (255) NOT NULL
3. PH\_ID NUMBER(10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. A\_ID NUMBER(10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

**PATIENT\_BLOOD\_INFO**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| P\_BLOOD\_GRP | VARCHAR2 (255) | UNIQUE |
| P\_BLOOD\_QNTY | NUMBER(10) |  |
| P\_ID | INT | FOREIGN KEY FROM ‘PATIENT’ TABLE |

**QUERY:**

1. P\_BLOOD\_GRP VARCHAR2 (255) UNIQUE
2. P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)

**HOSPTIAL**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| HOSP\_ID | INT | PRIMARY KEY |
| HOSP\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |

**QUERY:**

1. HOSP\_ID NUMBER (10) PRIMARY KEY
2. HOSP\_NAME VARCHAR2(255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

**H\_NEEDED\_BLOOD**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| HOSP\_NEEDED\_B\_GROUP | VARCHAR2 (255) | UNIQUE |
| HOSP\_NEEDED\_QNTY | NUMBER(24,1) |  |
| HOSP\_ID | NUMBER (10) | FOREIGN KEY FROM ‘HOSPITAL’ TABLE |

**QUERY:**

1. HOSP\_NEEDED\_B\_GROUP VARCHAR2 (255) UNIQUE
2. HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)

RECORDING\_STAFF:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| R\_ID | VARCHAR2 (255) | PRIMARY KEY |
| R\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER(10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| BD\_ID | VARCHAR2(255) | FOREIGN KEY FROM ‘DONER’ TABLE |
| P\_ID | DATE | FOREIGN KEY FROM ‘PATIENT’ TABLE |
| HOSP\_ID | VARCHAR2(255) | FOREIGN KEY FROM ‘HOSPITAL’ TABLE |

**QUERY:**

1. R\_ID VARCHAR2(255) PRIMARY KEY
2. R\_NAME VARCHAR2 (255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. BD\_ID INT ,

FOREIGN KEY (BD\_ID) REFERENCES DONER(BD\_ID)

1. P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)

1. HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)

**DISEASE**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| D\_ID | VARCHAR2 (255) | PRIMARY KEY |
| D\_NAME | VARCHAR2 (255) | NOT NULL |
| D\_MEDICINE | VARCHAR2 (255) | UNIQUE |
| P\_ID | INT | FOREIGN KEY FROM ‘PATIENT’ TABLE |

**QUERY:**

1. D\_ID VARCHAR2(255) PRIMARY KEY
2. D\_NAME VARCHAR2 (255) NOT NULL
3. P\_ID INT,

FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)

DOCTOR:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| DR\_ID | VARCHAR2 (255) | PRIMARY KEY |
| DR\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER(10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| HOSP\_ID | NUMBER(10) | FOREIGN KEY FROM ‘HOSPITAL’ TABLE |
| D\_ID | DATE | FOREIGN KEY FROM ‘DISEASE’ TABLE |

**QUERY:**

1. DR\_ID VARCHAR2 (255) PRIMARY KEY
2. DR\_NAME VARCHAR2 (255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. HOSP\_ID NUMBER (10),

FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)

1. D\_ID VARCHAR2 (255),

FOREIGN KEY (D\_ID) REFERENCES DISEASE(D\_ID)

**MANAGER**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| M\_ID | NUMBER(10) | PRIMARY KEY |
| M\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |

**QUERY:**

1. M\_ID NUMBER (10) PRIMARY KEY
2. M\_NAME VARCHAR2(255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

GIVES\_ORDER:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| HOSP\_ID | INT | PRIMARY KEY |
| HOSP\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |
| M\_ID | NUMBER (10) | FOREIGN KEY FROM ‘MANAGER’ TABLE |

**QUERY:**

1. HOSP\_ID NUMBER (10) PRIMARY KEY
2. HOSP\_NAME VARCHAR2(255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. A\_ID NUMBER (10),

  FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

1. M\_ID NUMBER (10),

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)

BLOOD\_BANK:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| BB\_ID | VARCHAR2 (255) | PRIMARY KEY |
| BB\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |
| A\_ID | NUMBER(10) | FOREIGN KEY FROM ‘ADDRESS’ TABLE |
| M\_ID | NUMBER (10) | FOREIGN KEY FROM ‘MANAGER’ TABLE |

**QUERY:**

1. BB\_ID VARCHAR2 (255) PRIMARY KEY
2. BB\_NAME VARCHAR2 (255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

1. A\_ID NUMBER (10),

FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)

1. M\_ID NUMBER (10),

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)

DISEASE\_FINDER:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| DF\_ID | NUMBER(10) | PRIMARY KEY |
| DF\_NAME | VARCHAR2 (255) | NOT NULL |
| PH\_ID | NUMBER (10) | FOREIGN KEY FROM ‘PHONES’ TABLE |

1. DF\_ID NUMBER (10) PRIMARY KEY
2. DF\_NAME VARCHAR2 (255) NOT NULL
3. PH\_ID NUMBER (10),

FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)

**BLOOD\_GROUP**:

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | DATA TYPE | CONSTRAINT |
| B\_NUM | NUMBER(10) | PRIMARY KEY |
| B\_GROUP | VARCHAR2 (255) | NOT NULL |
| STATUS | VARCHAR2 (255) | DEFAULT 'HEALTHY' |
| M\_ID | NUMBER (10) | FOREIGN KEY FROM ‘MANAGER’ TABLE |
| DF\_ID | NUMBER(10) | FOREIGN KEY FROM ‘DISEASE\_FINDER’ TABLE |

QUERY:

1. B\_NUM NUMBER (10) PRIMARY KEY
2. B\_GROUP VARCHAR2 (255) UNIQUE
3. STATUS VARCHAR2 (255) DEFAULT 'HEALTHY'
4. M\_ID NUMBER (10)

FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)

1. DF\_ID NUMBER(10)

FOREIGN KEY (DF\_ID) REFERENCES DISEASE\_FINDER (DF\_ID)

SAMPLE QUERY:

1. SELECT ALL THE INFORMATION OF PATIENT WHOSE ID IS EQUAL TO SAKIB

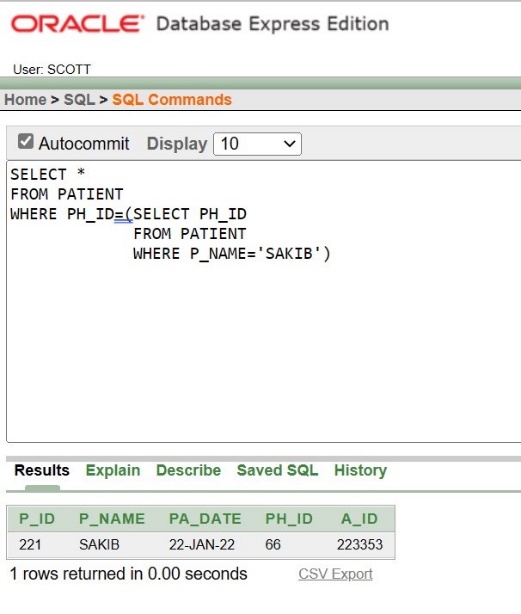
SELECT \*

FROM PATIENT

WHERE PH\_ID=(SELECT PH\_ID

FROM PATIENT

WHERE P\_NAME='SAKIB')



2. SHOW ALL THE DONER ID, NAME, AGE, BLOOD GROUP THOSE ID IS GREATER THAN THE BLOOD DONER WHOSE ID IS 2

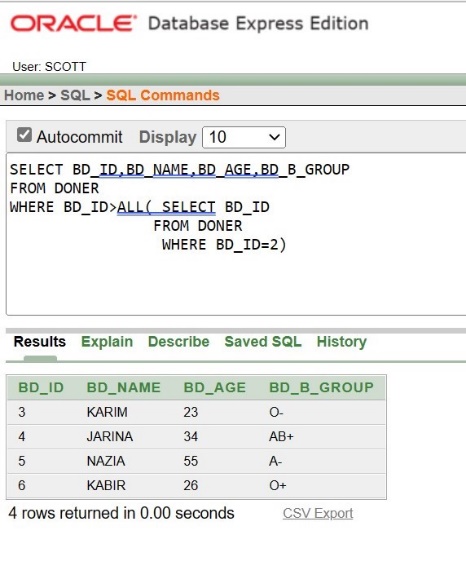
SELECT BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP

FROM DONER

WHERE BD\_ID>ALL( SELECT BD\_ID

FROM DONER

WHERE BD\_ID=2)

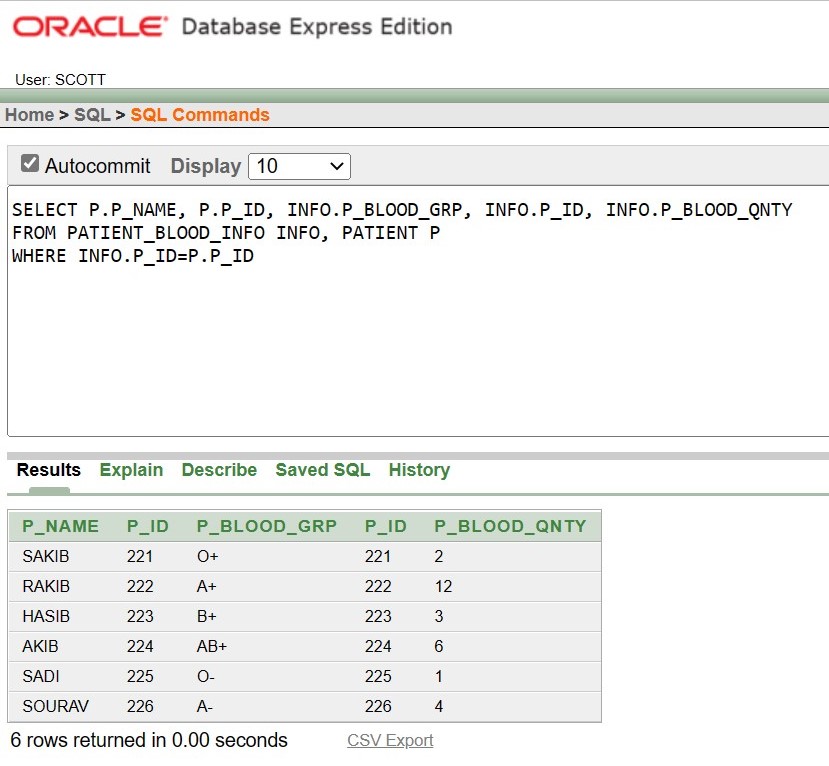


3. SHOW PATIENT NAME, PATIENT ID AND PATIENT BLOOD GROUP, QUANTITY USING ALIAS

SELECT P.P\_NAME, P.P\_ID, INFO.P\_BLOOD\_GRP, INFO.P\_ID, INFO.P\_BLOOD\_QNTY

FROM PATIENT\_BLOOD\_INFO INFO, PATIENT P

WHERE INFO.P\_ID=P.P\_ID

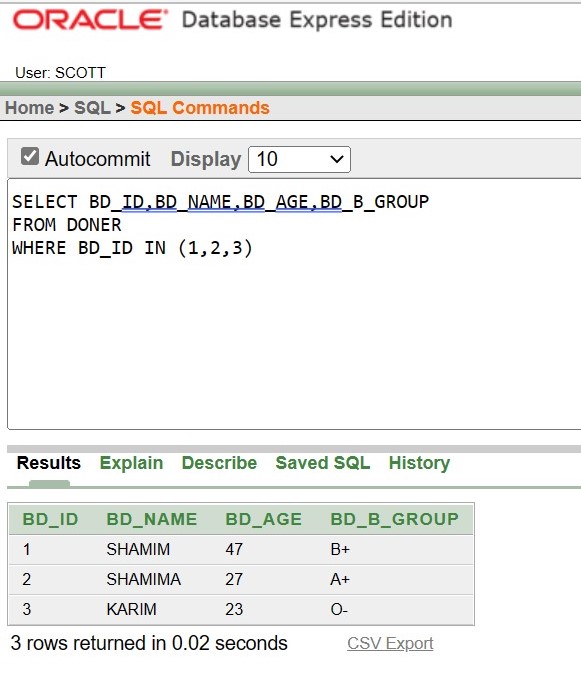


4. SHOW DONER ID, NAME, AGE, BLOOD GROUP WHOSE ID IN 1,2,3

SELECT BD\_ID,BD\_NAME,BD\_AGE,BD\_B\_GROUP

FROM DONER

WHERE BD\_ID IN (1,2,3)



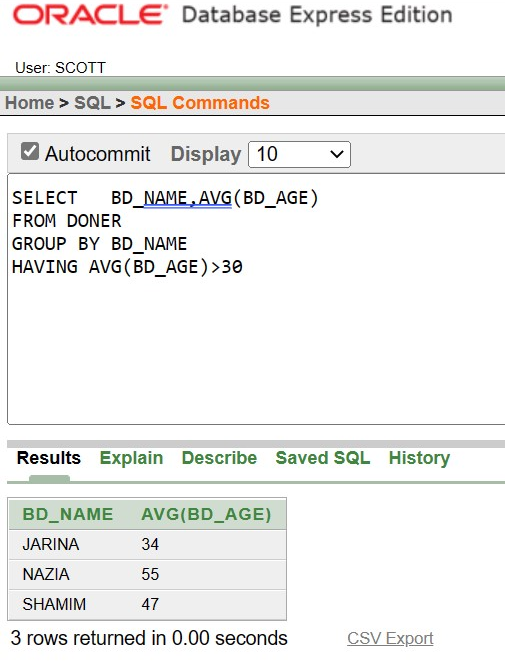
5. USING GROUP BY FUNCTION OF NAME AND SHOW DONER NAME AND AVERAGE OF THE AGE WHOSE AVERAGE AGE IS GREATER THAN 30

SELECT BD\_NAME,AVG(BD\_AGE)

FROM DONER

GROUP BY BD\_NAME

HAVING AVG(BD\_AGE)>30



6. CREATE A VIEW BETWEEN PATIENT AND PATIENT BLOOD QUANTITY AND SHOW PATIENT NAME, ID, BLOOD GROUP, BLOOD QUANTITY

CREATE VIEW INFO

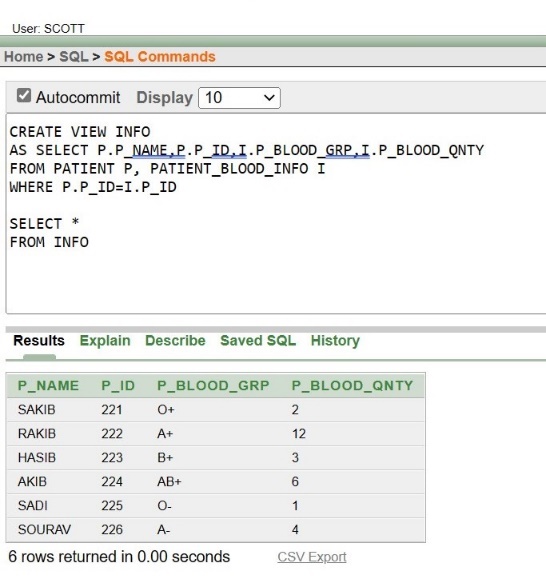
AS SELECT P.P\_NAME,P.P\_ID,I.P\_BLOOD\_GRP,I.P\_BLOOD\_QNTY

FROM PATIENT P, PATIENT\_BLOOD\_INFO I

WHERE P.P\_ID=I.P\_ID

SELECT \*

FROM INFO



7. CREATE A VIEW OF DONER TABLE AND SHOW ID, NAME, AGE WHOSE ID IN 1,2,3

CREATE VIEW INFOR

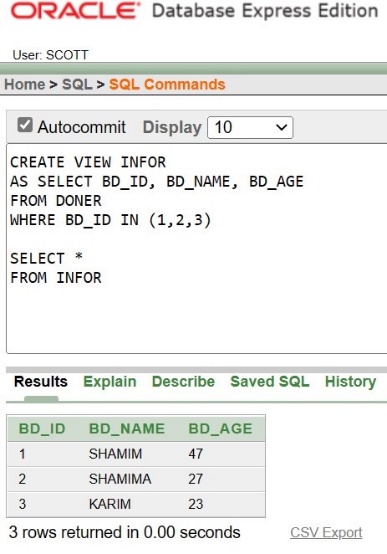
AS SELECT BD\_ID, BD\_NAME, BD\_AGE

FROM DONER

WHERE BD\_ID IN (1,2,3)

SELECT \*

FROM INFOR



8. SHOW ALL THE INFORMATION ABOUT DONER, INCLUDING ADDRESS.

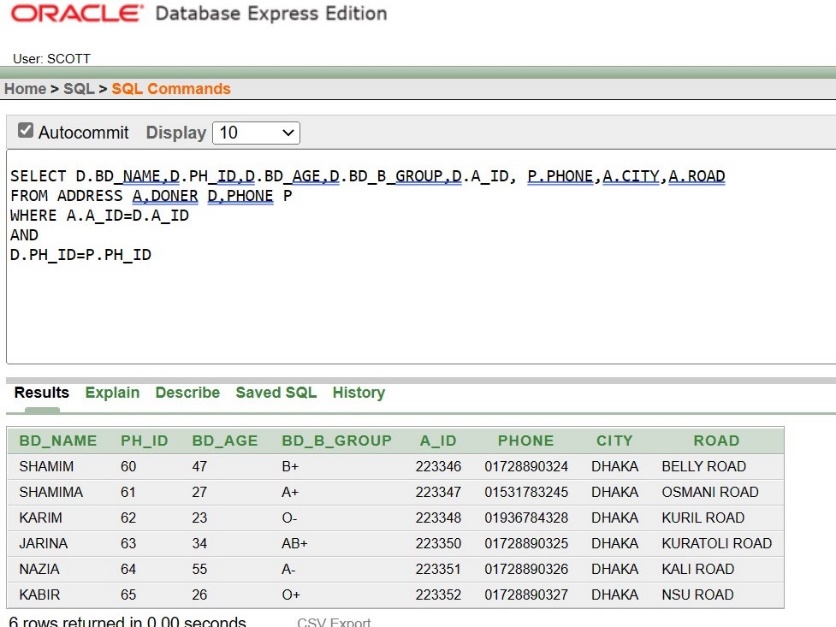
SELECT D.BD\_NAME,D.PH\_ID,D.BD\_AGE,D.BD\_B\_GROUP,D.A\_ID, P.PHONE,A.CITY,A.ROAD

FROM ADDRESS A,DONER D,PHONE P

WHERE A.A\_ID=D.A\_ID

AND

D.PH\_ID=P.PH\_ID

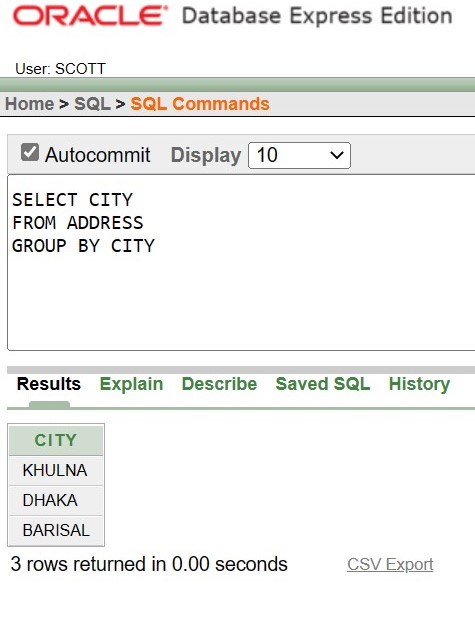


9. SHOW THE CITY OF ADDRESS TABLE USING GROUP BY FUNCTION

SELECT CITY

FROM ADDRESS

GROUP BY CITY

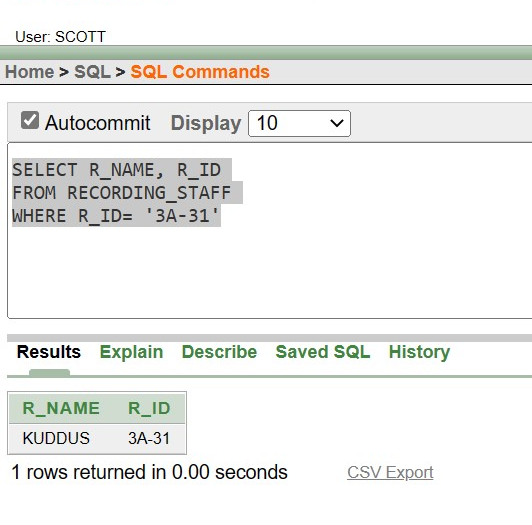


10. SHOW THE RECORDING STAFF NAME WHOSE ID 3A-31

SELECT R\_NAME, R\_ID

FROM RECORDING\_STAFF

WHERE R\_ID= '3A-31'



THANK YOU

• The relationship with Blood Bank manager and Blood Specimen is 1

to many. That’s why primary

key of Blood Bank manager is used as a foreign key in Blood Specimen

.

Disease Finder Table:

• The relationship with Disease finder and Blood Specimen is of 1 to

many. Therefore, the primary key of Disease finder is used as a foreign

key in Blood Specimen.

Blood Bank Manager Table:

• The relationship between Blood Bank Manager and Blood Specimen,

Recipient, Hospital info are all of 1 to many. So therefore, the primary

key of Blood Bank Manager is used as a foreign key in Blood Specimen,

Recipient and Hospital info.

Hospital info Table:

• The relationship with City and Hospital info is 1 to many. That’s why

primary key of City is used as a foreign key in Hospital info.

• The relationship with Blood Bank Manager and Hospital info is 1 to

many. That’s why primary key of Blood Bank manager is used as a

foreign key in Hospital info