



## DEPARTMENT OF COMPUTER SCIENCE

FALL 2022-23

### DATABASE FINAL PROJECT REPORT

COURSE	INTRODUCTION TO DATABASE	SECTION	G	GROUP	04

### GROUP MEMBERS:

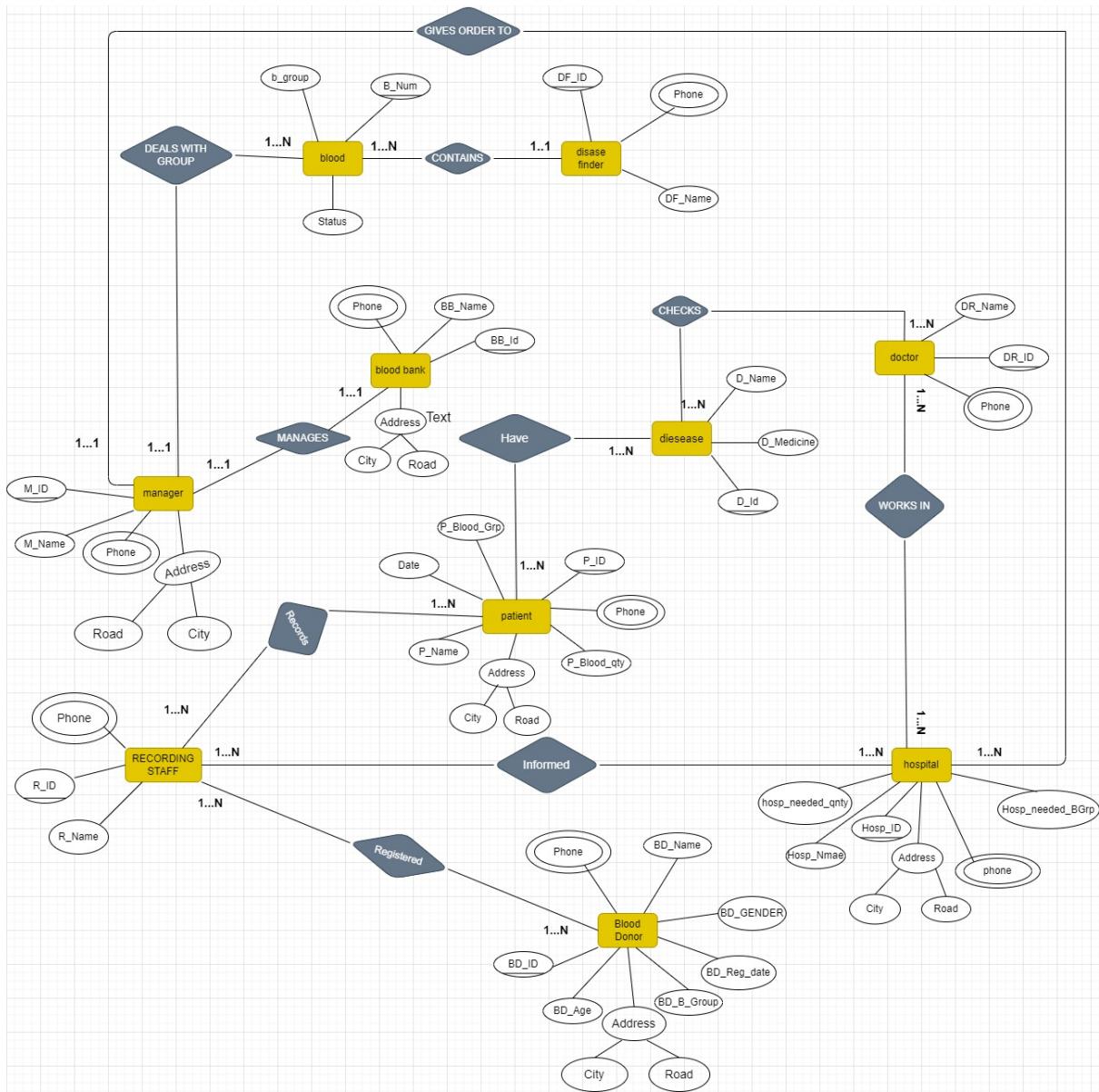
NAME	STUDENT ID
MD TAFHIMUL HAQUE SADI	22-47071-1
MD RAKIBUL ISLAM	22-47102-1
ABDULLAH AL HASIB	22-47055-1
SOUROB MISTRY	22-46077-1

## **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 identification 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 entity 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 entity.

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 required 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
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
5. R-ID, R-NAME, P-ID, PH-ID
6. PH-ID, PHONE
7. P-ID, P-NAME, PH-ID, DATE, A-ID
8. A-ID, CITY, ROAD
9. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID
10. R-ID,R-NAME,HOSP-ID,PH-ID
11. PH-ID, PHONE
12. HOSP-ID, HOSP-NAME,A-ID, PH-ID
13. A-ID,CITY,ROAD
14. HOSP-NEEDED-B-GRP,HOSP-NEEDED-QNTY,HOSP-ID
15. D-Id, D-NAME, D-MEDICINE, P-ID
16. P-ID, P-NAME, DATE, PH-ID, A-ID

17. PHONE, PH-ID
18. A-ID, CITY, ROAD
19. P-BLOOD-GRP, P-BLOOD-QNTY, P-ID
20. D-ID, D-NAME, D-MEDICINE
21. PH-ID, PHONE
22. DR-ID, DR-NAME, D-ID, PH-ID
23. DR-ID, DR-NAME, PH-ID, HOSP-ID
24. PH-ID, PHONE
25. HOSP-ID, HOSP-NAME, PH-ID, A-ID
26. A-ID, CITY, ROAD
27. HOSP-NEEDED-QNTY, HOSP-NEEDED-BGRP, HOSP-ID
28. M-ID, M-NAME, PH-ID, A-ID
29. PH-ID, PHONE
30. A-ID, CITY, ROAD
31. HOSP-ID, HOSP-NAME, PH-ID, A-ID, M-ID
32. HOSP-NEEDED-QNTY, HOSP-NEEDED-BGRP, HOSP-ID
33. M-ID, M-NAME, PH-ID, A-ID
34. CITY, ROAD, A-ID
35. PHONE, PH-ID
36. BB-ID, BB-NAME, PH-ID, A-ID, M-ID
38. M-ID, M\_NAME, PH-ID, A-ID
39. PH-ID, PHONE
40. A-ID, CITY, ROAD
41. B-NUM, B\_GROUP, STATUS, M-ID
42. DF-ID, DF-NAME, PH-ID
43. PH-ID, PHONE
44. 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
)
```

The screenshot shows a SQL command interface with the following details:

- Header:** User: SCOTT
- Breadcrumbs:** Home > SQL > SQL Commands
- Autocommit:** Checked
- Display:** 10
- SQL Commands:**

```
CREATE TABLE PHONE (
    PH_ID NUMBER (10) PRIMARY KEY,
    PHONE VARCHAR2(255) NOT NULL
)
DESC PHONE
```
- Results Tab:** Selected
- Table Definition:**

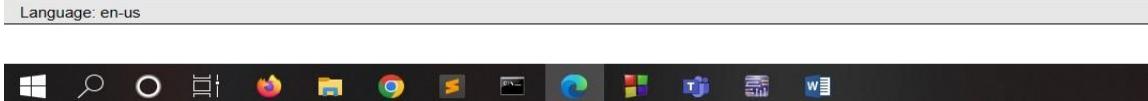
Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PHONE	PH_ID	Number	-	10	0	-	1	-	-	-
PHONE	PHONE	Varchar2	255	-	-	-	-	-	-	-

1 - 2

### 2 . ADDRESS TABLE

QUERY:

```
CREATE TABLE ADDRESS (
    A_ID NUMBER (10) PRIMARY KEY,
    CITY VARCHAR2 (255) NOT NULL,
    ROAD VARCHAR2(255) UNIQUE
)
```



**ORACLE® Database Express Edition**

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
CREATE TABLE ADDRESS (
A_ID NUMBER (10) PRIMARY KEY,
CITY VARCHAR2 (255) NOT NULL,
ROAD VARCHAR2(255) UNIQUE
)
```

```
DESC ADDRESS
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **ADDRESS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESS	A_ID	Number	-	10	0	1	-	-	-
	CITY	Varchar2	255	-	-	-	-	-	-
	ROAD	Varchar2	255	-	-	-	✓	-	-

1 - 3

Language: en-us

### 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))
```

Autocommit

```
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))

DESC DONER
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DONER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DONER	BD_ID	Number	-	-	0	1	-	-	-
	BD_NAME	Varchar2	255	-	-	-	-	-	-
	BD_AGE	Number	-	10	0	-	✓	-	-
	BD_B_GROUP	Varchar2	255	-	-	-	✓	-	-
	BD_REG_DATE	Date	7	-	-	-	✓	-	-
	BD_GENDER	Varchar2	255	-	-	-	✓	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-

1 - 8

Language: en-us

#### 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)
)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)
)

DESC PATIENT
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PATIENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PATIENT	P_ID	Number	-	-	0	1	-	-	-
	P_NAME	Varchar2	255	-	-	-	-	-	-
	PA_DATE	Date	7	-	-	-	✓	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-

1 - 5

Language: en-us

## 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC PATIENT_BLOOD_INFO
```

**Results Explain Describe Saved SQL History**

Object Type **TABLE** Object **PATIENT\_BLOOD\_INFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PATIENT_BLOOD_INFO	P_BLOOD_GRP	Varchar2	255	-	-	-	✓	-	-
	P_BLOOD_QNTY	Number	-	10	0	-	✓	-	-
	P_ID	Number	-	-	0	-	✓	-	-

1 - 3

Language: en-us



## 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)
)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)
)

DESC HOSPITAL
```

Results Explain Describe Saved SQL History

Object Type TABLE Object HOSPITAL

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HOSPITAL	HOSP_ID	Number	-	10	0	1	-	-	-
	HOSP_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-

1 - 4

Language: en-us

## 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC H_NEEDED_BLOOD
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **H\_NEEDED\_BLOOD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
H_NEEDED_BLOOD	HOSP_NEEDED_B_GROUP	Varchar2	255	-	-	-	✓	-	-
	HOSP_NEEDED_QNTY	Number	-	24	1	-	✓	-	-
	HOSP_ID	Number	-	10	0	-	✓	-	-

1 - 3

Language: en-us

## 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)
)
```

```

AutoCommit Display [+] ▾
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)
)

DESC RECORDING_STAFF

Results Explain Describe Saved SQL History
Object Type TABLE Object RECORDING_STAFF
Table Column Data Type Length Precision Scale Primary Key Nullable Default Comment
RECORDING_STAFF R_ID Varchar2 255 - - 1 - - -
RECORDING_STAFF R_NAME Varchar2 255 - - - - - -
RECORDING_STAFF PH_ID Number - 10 0 - ✓ - - -
RECORDING_STAFF BD_ID Number - - 0 - ✓ - - -
RECORDING_STAFF P_ID Number - - 0 - ✓ - - -
RECORDING_STAFF HOSP_ID Number - 10 0 - ✓ - - -
1 - 6
Language: English

```

## 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)
)

```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)
```

```
DESC DISEASE
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DISEASE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DISEASE	D_ID	Varchar2	255	-	-	1	-	-	-
	D_NAME	Varchar2	255	-	-	-	-	-	-
	D_MEDICINE	Varchar2	255	-	-	-	✓	-	-
	P_ID	Number	-	-	0	-	✓	-	-
									1 - 4

Language: en-us

## 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC DOCTOR
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DOCTOR**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOCTOR	DR_ID	Varchar2	255	-	-	1	-	-	-
	DR_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	HOSP_ID	Number	-	10	0	-	✓	-	-
	D_ID	Varchar2	255	-	-	-	✓	-	-

1 - 5

## 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC MANAGER
```

Results Explain Describe Saved SQL History

Object Type	TABLE Object	MANAGER							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	M_ID	Number	-	10	0	1	-	-	-
	M_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-

1 - 4

Language: en-us

## 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC GIVES_ORDER
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **GIVES\_ORDER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GIVES_ORDER	HOSP_ID	Number	-	10	0	1	-	-	-
	HOSP_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 5

### 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)
)
```

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)
)

DESC BLOOD_BANK
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **BLOOD\_BANK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BLOOD_BANK	BB_ID	Varchar2	255	-	-	1	-	-	-
	BB_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-
	A_ID	Number	-	10	0	-	✓	-	-
	M_ID	Number	-	10	0	-	✓	-	-

1 - 5

#### 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)
)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)
)
```

```
DESC DISEASE_FINDER
```

Results Explain Describe Saved SQL History

Object Type TABLE Object DISEASE\_FINDER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DISEASE_FINDER	DF_ID	Number	-	10	0	1	-	-	-
	DF_NAME	Varchar2	255	-	-	-	-	-	-
	PH_ID	Number	-	10	0	-	✓	-	-

1 - 3

Language: en-us

## 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)
)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)
)
```

```
DESC BLOOD_GROUP
```

Results Explain Describe Saved SQL History

Object Type TABLE Object BLOOD\_GROUP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BLOOD_GROUP	B_NUM	Number	-	10	0	1	-	-	-
BLOOD_GROUP	B_GROUP	Varchar2	255	-	-	-	✓	-	-
BLOOD_GROUP	STATUS	Varchar2	255	-	-	-	✓	'HEALTHY'	-
BLOOD_GROUP	M_ID	Number	-	10	0	-	✓	-	-
BLOOD_GROUP	DF_ID	Number	-	10	0	-	✓	-	-

1 - 5

## 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')
```

SQL Commands

```

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')

SELECT *
FROM PHONE

```

Results Explain Describe Saved SQL History

PH_ID	PHONE
60	01728890324
61	01531783245
62	01936784328
63	01728890325
64	01728890326
65	01728890327
66	01728890328
67	01728890329
68	01728890330
69	01728890331

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us

## 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 ROADD')
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')

```

← ⏪ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:2771607747712531::NO:::

```

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 ROADD')
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')

SELECT *

```

**Results Explain Describe Saved SQL History**

A_ID	CITY	ROAD
223346	DHAKA	BELLY ROAD
223347	DHAKA	OSMANI ROAD
223348	DHAKA	KURIL ROAD
223350	DHAKA	KURATOLI ROAD
223351	DHAKA	KALI ROAD
223352	DHAKA	NSU ROAD
223353	DHAKA	NOBI NOGOR ROAD
223354	DHAKA	AIRPORT ROAD
223355	DHAKA	MUDDHA ROAD
223356	DHAKA	PURAN DHAKA ROAD

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us

### 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)
```

The screenshot shows the Oracle SQL developer interface. At the top, there's a toolbar with 'Autocommit' checked, a 'Display' dropdown set to 10, and a search icon. Below the toolbar is the SQL query window containing the six INSERT statements for the 'DONER' table. After the query, there's a 'SELECT \* FROM DONER' statement. At the bottom of the interface, there's a 'Results' tab where the data is displayed in a grid format.

BD_ID	BD_NAME	BD_AGE	BD_B_GROUP	BD_REG_DATE	BD_GENDER	PH_ID	A_ID
1	SHAMIM	47	B+	17-FEB-22	MALE	60	223346
2	SHAMIMA	27	A+	05-FEB-22	FEMALE	61	223347
3	KARIM	23	O-	10-FEB-22	MALE	62	223348
4	JARINA	34	AB+	16-FEB-22	FEMALE	63	223350
5	NAZIA	55	A-	19-FEB-22	FEMALE	64	223351
6	KABIR	26	O+	13-FEB-22	MALE	65	223352

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Copyright © 1999, 2006, Oracle

#### 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)
```

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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)

SELECT *
FROM PATIENT
```

Results Explain Describe Saved SQL History

P_ID	P_NAME	PA_DATE	PH_ID	A_ID
221	SAKIB	22-JAN-22	66	223353
222	RAKIB	22-JAN-22	67	223354
223	HASIB	22-JAN-22	68	223355
224	AKIB	22-JAN-22	69	223356
225	SADI	22-JAN-22	70	223357
226	SOURAV	22-JAN-22	71	223358

6 rows returned in 0.00 seconds

[CSV Export](#)

## 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)
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the six insert statements for the PATIENT\_BLOOD\_INFO table. Below the commands, a SELECT \* query is run against the same table. The results pane displays a grid of data with columns P\_BLOOD\_GRP, P\_BLOOD\_QNTY, and P\_ID, showing the inserted values.

P_BLOOD_GRP	P_BLOOD_QNTY	P_ID
O+	2	221
A+	12	222
B+	3	223
AB+	6	224
O-	1	225
A-	4	226

6 rows returned in 0.00 seconds [CSV Export](#)

## 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)
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is connected as SCOTT. In the SQL pane, the following SQL code is executed:

```

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)

SELECT *
FROM HOSPITAL

```

The Results pane displays the following table:

HOSP_ID	HOSP_NAME	PH_ID	A_ID
1	KURMITOLA A GENERAL HOSPITAL	72	223359
2	KURMITOLA B GENERAL HOSPITAL	73	223360
3	KURMITOLA C GENERAL HOSPITAL	74	223361
4	KURMITOLA D GENERAL HOSPITAL	75	223362
5	KURMITOLA E GENERAL HOSPITAL	76	223363
6	KURMITOLA F GENERAL HOSPITAL	77	223364

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.29

Whiteboard Fullscreen snip ...

## 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)

```

SQL Commands

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit  Display 10  Save  Run

```
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)

SELECT *
FROM H_NEEDED_BLOOD
```

Results Explain Describe Saved SQL History

HOSP_NEEDED_B_GROUP	HOSP_NEEDED_QNTY	HOSP_ID
AB-	2	1
B-	4	2
A-	5	3
A+	1	4
O-	8	5
AB+	3	6

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## 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)
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor window contains the following code:

```

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)

SELECT *
FROM DISEASE

```

The results pane displays the following table:

D_ID	D_NAME	D_MEDICINE	P_ID
D1-01	TYPHOID	Ciprofloxacin	221
D1-02	ORCHITIS	Rocephin	222
D1-03	PCOS	clomifene	223
D1-04	TRICHONONIASIS	metronidazole	224
D1-05	FEVER	Tylenol	225
D1-06	ANEMIA	Kenalog	226

6 rows returned in 0.02 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## 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')

```

SQL Commands

User SCOTT

Home > SQL > SQL Commands

```
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')

SELECT *
FROM DOCTOR
```

Results

DR_ID	DR_NAME	PH_ID	HOSP_ID	D_ID
1	KAMRUZZAMAN	108	1	D1-01
2	ATAUR	107	2	D1-02
3	RAFIQUE	106	3	D1-03
4	ASHFAQ	105	4	D1-04
5	MANISH	104	5	D1-05
6	SADIA	103	6	D1-06

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## 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)
```

SQL Commands

User SCOTT

Home > SQL > SQL Commands

```
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)

SELECT *
FROM MANAGER
```

Results

M_ID	M_NAME	PH_ID	A_ID
1	RAFIQUL	84	223367
2	RAFIQ	85	223368
3	SAFIQ	87	223369
4	RAFI	88	223370
5	PRITI	89	223371
6	SDIYA	90	223372

6 rows returned in 0.02 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

## 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)
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
FROM DISEASE  
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)  
  
SELECT *  
FROM GIVES_ORDER
```

The results section displays the following table:

HOSP_ID	HOSP_NAME	PH_ID	A_ID	M_ID
1	KURMITOLA A GENERAL HOSPITAL	96	223359	6
2	KURMITOLA B GENERAL HOSPITAL	91	223360	1
3	KURMITOLA C GENERAL HOSPITAL	92	223361	2
4	KURMITOLA D GENERAL HOSPITAL	93	223362	3
5	KURMITOLA E GENERAL HOSPITAL	94	223363	4
6	KURMITOLA F GENERAL HOSPITAL	95	223364	5

6 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

### 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)
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2771607747712531::NO:::. The session user is SCOTT. The SQL command window contains the six INSERT statements and a SELECT \* query from the BLOOD\_BANK table. The results window displays a table with 6 rows, matching the data inserted.

BB_ID	BB_NAME	PH_ID	A_ID	M_ID
1	SHONDHANI 1	102	223378	1
2	SHONDHANI 2	101	223377	2
3	SHONDHANI 3	100	223376	3
4	SHONDHANI 4	99	223375	4
5	SHONDHANI 5	98	223374	5
6	SHONDHANI 6	97	223373	6

6 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1  
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

## 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)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)

SELECT *
FROM DISEASE_FINDER
```

Results Explain Describe Saved SQL History

DF_ID	DF_NAME	PH_ID
812	SAKIB	97
813	ARNOB	98
814	EQRAMUL	99
815	SWAPNIL	100
816	FAHIM	101
811	ANTOR	102

6 rows returned in 0.00 seconds [CSV Export](#)

## 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)
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
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)
```

```
SELECT *
FROM BLOOD_GROUP
```

Results Explain Describe Saved SQL History

B_NUM	B_GROUP	STATUS	M_ID	DF_ID
1	A-	HEALTHY	1	812
2	O+	UNHEALTHY	2	813
3	AB-	HEALTHY	3	814
4	B-	HEALTHY	6	815
5	B+	HIV	4	816
6	A+	HEALTHY	5	811

6 rows returned in 0.00 seconds

[CSV Export](#)

## CONSTRAINTS

### 1.PHONE:

COLUMN NAME	DATA TYPE	CONSTRAINT
PH_ID	NUMBER (10)	PRIMARY KEY
PHONE	VARCHAR2(255)	NOT NULL

### QUERY:

```
i.PH_ID NUMBER (10) PRIMARY KEY
ii.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:

```
i.A_ID NUMBER (10) PRIMARY KEY
ii. CITY VARCHAR2 (255) NOT NULL
iii.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:**

- i.BD\_ID INT PRIMARY KEY
- ii.BD\_NAME VARCHAR2 (255) NOT NULL
- iii.BD\_AGE NUMBER (10) CHECK (BD\_AGE>18)
- iv.BD\_B\_GROUP VARCHAR2 (255) UNIQUE
- v.PH\_ID NUMBER(10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID),
- vi.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:**

- i.P\_ID INT PRIMARY KEY
- ii.P\_NAME VARCHAR2 (255) NOT NULL
- iii.PH\_ID NUMBER(10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)
- iv.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:**

- i.P\_BLOOD\_GRP VARCHAR2 (255) UNIQUE
- ii.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:**

- i.HOSP\_ID NUMBER (10) PRIMARY KEY
- ii.HOSP\_NAME VARCHAR2(255) NOT NULL
- iii.PH\_ID NUMBER (10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)
- iv.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:**

- i.HOSP\_NEEDED\_B\_GROUP VARCHAR2 (255) UNIQUE
- ii.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:**

i.R\_ID VARCHAR2(255) PRIMARY KEY  
 ii.R\_NAME VARCHAR2 (255) NOT NULL  
 iii.PH\_ID NUMBER (10),  
     FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)  
 iv.BD\_ID INT ,  
     FOREIGN KEY (BD\_ID) REFERENCES DONER(BD\_ID)  
 v.P\_ID INT,  
     FOREIGN KEY (P\_ID) REFERENCES PATIENT(P\_ID)  
 vi.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:**

i.D\_ID VARCHAR2(255) PRIMARY KEY  
 ii.D\_NAME VARCHAR2 (255) NOT NULL  
 iii.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:**

i.DR\_ID VARCHAR2 (255) PRIMARY KEY  
 ii.DR\_NAME VARCHAR2 (255) NOT NULL  
 iii.PH\_ID NUMBER (10),  
     FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)  
 iv.HOSP\_ID NUMBER (10),  
     FOREIGN KEY (HOSP\_ID) REFERENCES HOSPITAL(HOSP\_ID)  
 v.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:**

- i.M\_ID NUMBER (10) PRIMARY KEY
- ii.M\_NAME VARCHAR2(255) NOT NULL
- iii.PH\_ID NUMBER (10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)
- iv.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:**

- i.HOSP\_ID NUMBER (10) PRIMARY KEY
- ii.HOSP\_NAME VARCHAR2(255) NOT NULL
- iii.PH\_ID NUMBER (10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)
- iv.A\_ID NUMBER (10),  
FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)
- v.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:**

i.BB\_ID VARCHAR2 (255) PRIMARY KEY  
ii.BB\_NAME VARCHAR2 (255) NOT NULL  
iii.PH\_ID NUMBER (10),  
FOREIGN KEY (PH\_ID) REFERENCES PHONE(PH\_ID)  
iv.A\_ID NUMBER (10),  
FOREIGN KEY (A\_ID) REFERENCES ADDRESS(A\_ID)  
v.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

i.DF\_ID NUMBER (10) PRIMARY KEY  
ii.DF\_NAME VARCHAR2 (255) NOT NULL  
iii.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:**

i.B\_NUM NUMBER (10) PRIMARY KEY  
ii.B\_GROUP VARCHAR2 (255) UNIQUE  
iii.STATUS VARCHAR2 (255) DEFAULT 'HEALTHY'  
iv.M\_ID NUMBER (10)  
FOREIGN KEY (M\_ID) REFERENCES MANAGER(M\_ID)  
vi. 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')
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following query:

```
SELECT *
FROM PATIENT
WHERE PH_ID=(SELECT PH_ID
              FROM PATIENT
            WHERE P_NAME='SAKIB')
```

The results pane shows a single row returned in 0.00 seconds:

P_ID	P_NAME	PA_DATE	PH_ID	A_ID
221	SAKIB	22-JAN-22	66	223353

CSV Export

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)
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following query:

```
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)
```

The results pane shows four rows returned in 0.00 seconds:

BD_ID	BD_NAME	BD_AGE	BD_B_GROUP
3	KARIM	23	O-
4	JARINA	34	AB+
5	NAZIA	55	A-
6	KABIR	26	O+

CSV Export

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

**ORACLE® Database Express Edition**

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
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
```

**Results Explain Describe Saved SQL History**

P_NAME	P_ID	P_BLOOD_GRP	P_ID	P_BLOOD_QNTY
SAKIB	221	O+	221	2
RAKIB	222	A+	222	12
HASIB	223	B+	223	3
AKIB	224	AB+	224	6
SADI	225	O-	225	1
SOURAV	226	A-	226	4

6 rows returned in 0.00 seconds [CSV Export](#)

#### 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)
```

**ORACLE® Database Express Edition**

User: SCOTT

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
SELECT BD_ID,BD_NAME,BD_AGE,BD_B_GROUP
FROM DONER
WHERE BD_ID IN (1,2,3)
```

**Results Explain Describe Saved SQL History**

BD_ID	BD_NAME	BD_AGE	BD_B_GROUP
1	SHAMIM	47	B+
2	SHAMIMA	27	A+
3	KARIM	23	O-

3 rows returned in 0.02 seconds [CSV Export](#)

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
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
SELECT BD_NAME, AVG(BD_AGE)
FROM DONER
GROUP BY BD_NAME
HAVING AVG(BD_AGE)>30
```

The results section displays a table with three rows:

BD_NAME	AVG(BD_AGE)
JARINA	34
NAZIA	55
SHAMIM	47

Below the table, it says "3 rows returned in 0.00 seconds" and there is a "CSV Export" link.

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
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
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
```

The results section displays a table with six rows:

P_NAME	P_ID	P_BLOOD_GRP	P_BLOOD_QNTY
SAKIB	221	O+	2
RAKIB	222	A+	12
HASIB	223	B+	3
AKIB	224	AB+	6
SADI	225	O-	1
SOURAV	226	A-	4

Below the table, it says "6 rows returned in 0.00 seconds" and there is a "CSV Export" link.

## 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
```

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE VIEW INFOR  
AS SELECT BD_ID, BD_NAME, BD_AGE  
FROM DONER  
WHERE BD_ID IN (1,2,3)  
  
SELECT *  
FROM INFOR
```

Results Explain Describe Saved SQL History

BD_ID	BD_NAME	BD_AGE
1	SHAMIM	47
2	SHAMIMA	27
3	KARIM	23

3 rows returned in 0.00 seconds [CSV Export](#)

## 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
```

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
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
```

Results Explain Describe Saved SQL History

BD_NAME	PH_ID	BD_AGE	BD_B_GROUP	A_ID	PHONE	CITY	ROAD
SHAMIM	60	47	B+	223346	01728890324	DHAKA	BELLY ROAD
SHAMIMA	61	27	A+	223347	01531783245	DHAKA	OSMANI ROAD
KARIM	62	23	O-	223348	01936784328	DHAKA	KURIL ROAD
JARINA	63	34	AB+	223350	01728890325	DHAKA	KURATOLI ROAD
NAZIA	64	55	A-	223351	01728890326	DHAKA	KALI ROAD
KABIR	65	26	O+	223352	01728890327	DHAKA	NSU ROAD

6 rows returned in 0.00 seconds [CSV Export](#)

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

```
SELECT CITY  
FROM ADDRESS  
GROUP BY CITY
```

**ORACLE® Database Express Edition**

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT CITY  
FROM ADDRESS  
GROUP BY CITY
```

Results Explain Describe Saved SQL History

CITY
KHULNA
DHAKA
BARISAL

3 rows returned in 0.00 seconds [CSV Export](#)

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

```
SELECT R_NAME, R_ID  
FROM RECORDING_STAFF  
WHERE R_ID= '3A-31'
```

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
SELECT R_NAME, R_ID  
FROM RECORDING_STAFF  
WHERE R_ID= '3A-31'
```

Results Explain Describe Saved SQL History

R_NAME	R_ID
KUDDUS	3A-31

1 rows returned in 0.00 seconds [CSV Export](#)

*THANK YOU*