



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

INTRODUCTION TO DATABASE

FALL 2022-2023

Section: G Group: 05

PROJECT REPORT ON ORGAN BANK MANAGEMENT SYSTEM

Supervised By

Syed Nafiul Shefat

Submitted By:

NAME	ID
MD. SHOHANUR RAHMAN SHOHAN	22-46013-1
MAHMUDA KHATUN	22-47016-1
A. F. M. RAFIUL HASSAN	22-47048-1

TABLE OF CONTENTS

TOPICS	Page no.
1. Introduction	3
2. Scenario Description	3
3. Entity Relationship Diagram	4
4. Normalization	5-10
5. Total Table	11
6. Finalization	12
7. Constraints	13-15
8. Table Creation	16-19
9. Data Insertion	20-23
10.Query Test	24-27
11.Conclusion	28

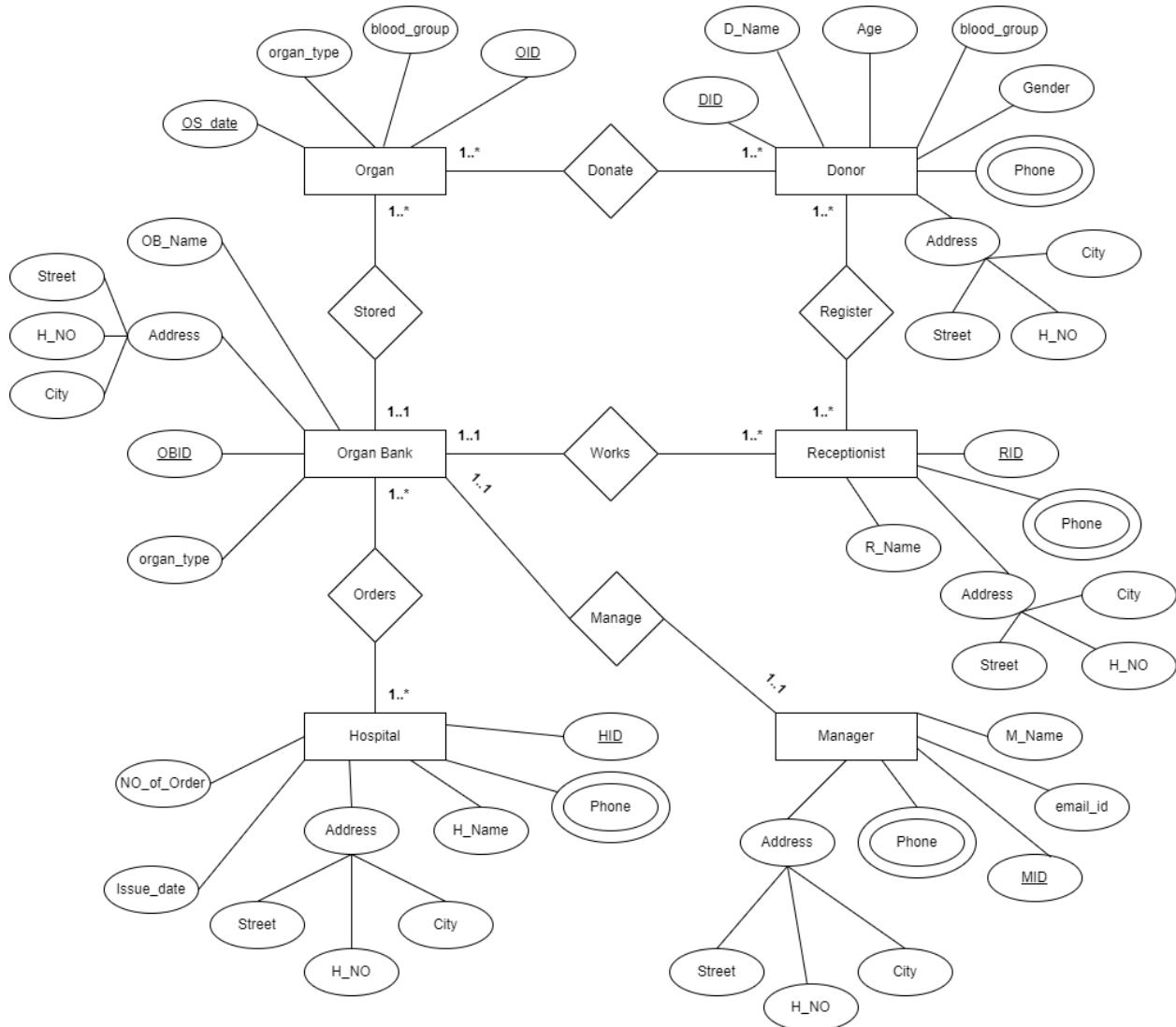
INTRODUCTION

The Organ Bank Management System is required for managing organ, designed for use by the Life Angel Organ Bank. The system is designed to connect with an extensive database of donor, organ, manager etc. details. These entities are organized the Organ Bank in hierachal structure which can be traversed briefly, giving commanders a visual summary of statuses through an Oracle 10g Database system interface.

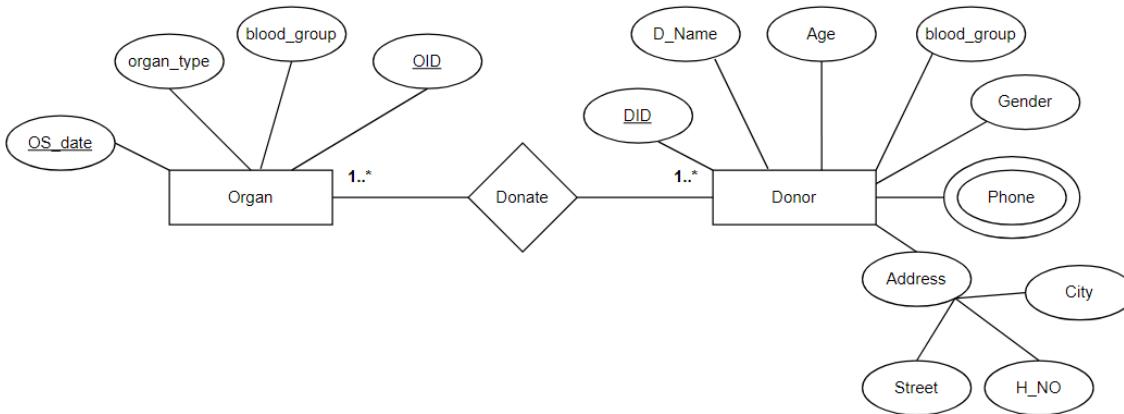
Scenario Description

An organ bank stores various types of organs. Many donors can donate different type of organ. A donor may donate organ more than once and he/she is identified by a donor id, name, age, gender, address, and phone number. The organ donated by the donor is distinguish by organ type, organ id, blood group and organ stored date. Before each donor donates his/her organ, he/she is required to register himself as a donor with the receptionist who works at the organ bank. The receptionist is characterized by receptionist id, name, address, and phone number. The organ banks receive orders for organ from many hospitals for emergency purposes and other surgical requirements and organ bank issues the same of required type of organ. The hospitals are identified by hospital id, name, address, organ issue date, orders, and phone number. Organ bank has organ bank id, name, address, and organ types stored. The organ bank is managed by the manager who is identified by manager id, name, e-mail id, address, and phone number. He/she is responsible for the proper management of the organ bank.

Entity Relationship Diagram



NORMALIZATION



Donate:

UNF:

DID, D_NAME, AGE, BLOOD_GROUP, GENDER, PHONE,
H_NO, STREET, CITY, OID, BLOOD_GROUP, ORGAN_TYPE,
OS_DATE

1NF: PHONE is a multivalued attribute.

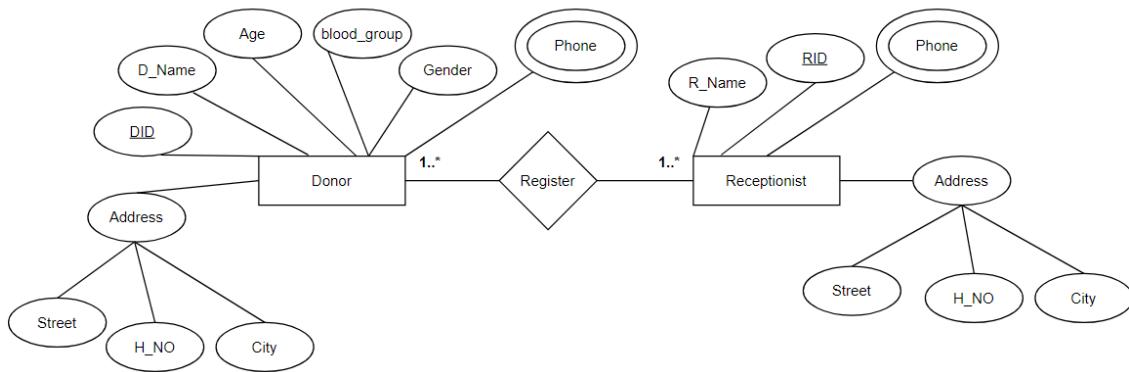
DID, D_NAME, AGE, BLOOD_GROUP, GENDER, H_NO,
STREET, CITY, OID, BLOOD_GROUP, ORGAN_TYPE,
OS_DATE

2NF:

1. DID, D_NAME, AGE, BLOOD_GROUP, GENDER, H_NO,
STREET, CITY, OID, PID
2. PHONE, PID
3. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE

3NF:

1. DID, D_NAME, AGE, BLOOD_GROUP, GENDER, OID, PID, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE



Register:

UNF:

DID, D_NAME, AGE, BLOOD_GROUP, GENDER, PHONE,
H_NO, STREET, CITY, RID, R_NAME, PHONE, H_NO, STREET,
CITY

1NF: PHONE is a multivalued attribute.

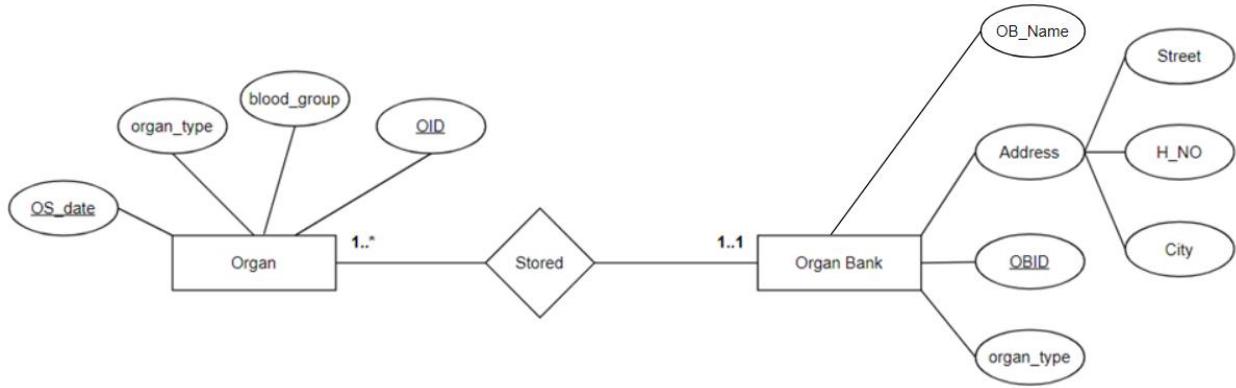
DID, D_NAME, AGE, BLOOD_GROUP, GENDER, H_NO,
STREET, CITY, RID, R_NAME, PHONE, H_NO, STREET, CITY

2NF:

1. DID, D_NAME, AGE, BLOOD_GROUP, GENDER, H_NO,
STREET, CITY, RID, PID
2. PID, PHONE
3. RID, R_NAME, H_NO, STREET, CITY, PID

3NF:

1. DID, D_NAME, AGE, BLOOD_GROUP, RID, PID, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. RID, R_NAME, PID, AID



Stored:

UNF:

OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID,
OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY

1NF: NO multivalued attribute. (SAME AS UNF)

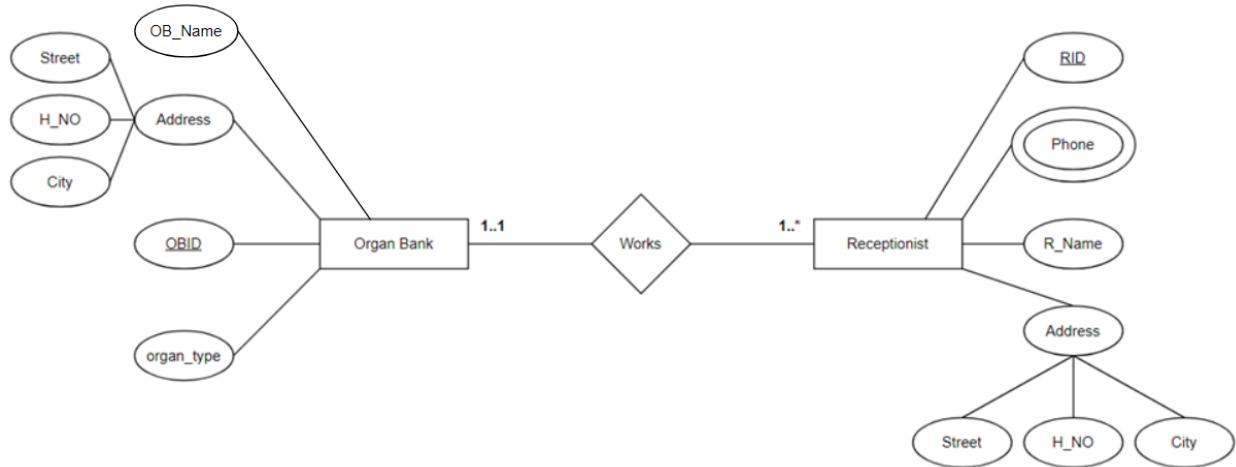
OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID,
OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY

2NF:

1. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID
2. OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY

3NF:

1. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID
2. OBID, OB_NAME, ORGAN_TYPE, AID
3. AID, H_NO, STREET, CITY



Works:

UNF:

OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, RID,
R_NAME, PHONE, H_NO, STREET, CITY

1NF: PHONE is a multivalued attribute.

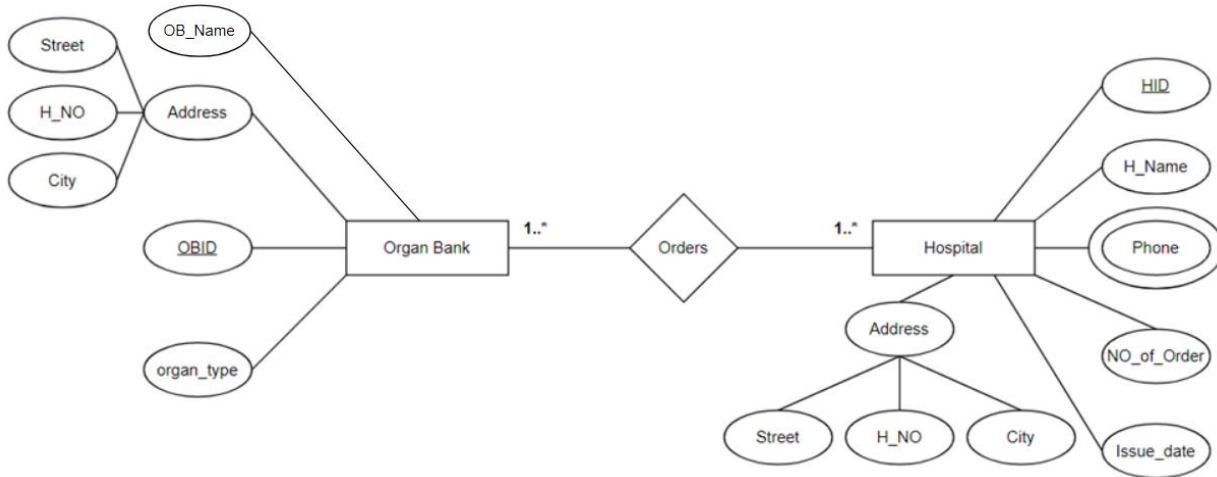
OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, RID,
R_NAME, H_NO, STREET, CITY

2NF:

1. OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY
2. PID, PHONE
3. RID, R_NAME, PHONE, H_NO, STREET, CITY, OBID, PID

3NF:

1. OBID, OB_NAME, ORGAN_TYPE, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. RID, R_NAME, OBID, PID, AID



Orders:

UNF:

OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, HID,
H_NAME, PHONE, NO_OF_ORDER, ISSUE_DATE, H_NO,
STREET, CITY

1NF: PHONE is a multivalued attribute.

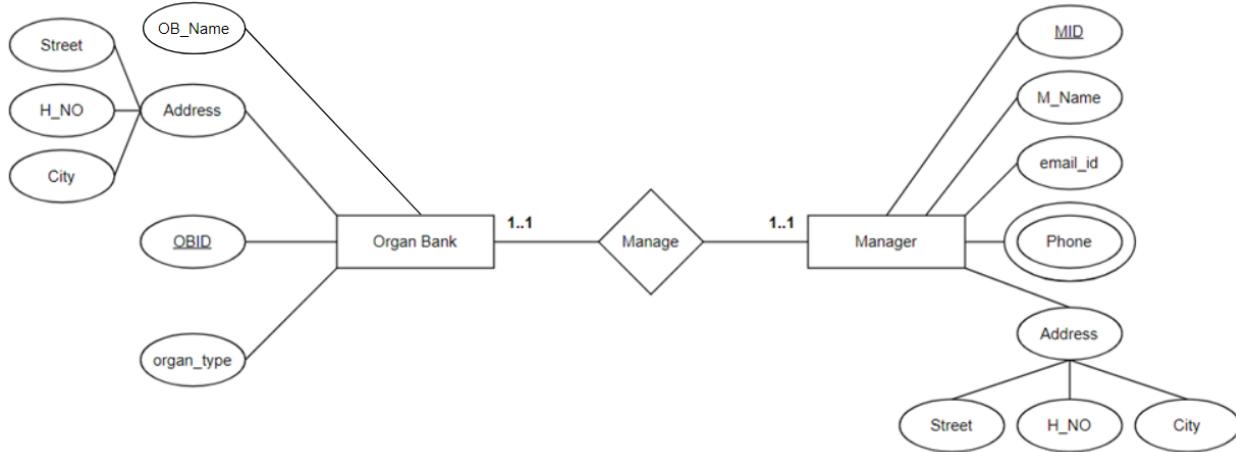
OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, HID,
H_NAME, NO_OF_ORDER, ISSUE_DATE, H_NO, STREET,
CITY

2NF:

1. OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY
2. PID, PHONE
3. HID, H_NAME, NO_OF_ORDER, ISSUE_DATE, H_NO, STREET,
CITY, OBID, PID

3NF:

1. OBID, OB_NAME, ORGAN_TYPE, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. HID, H_NAME, NO_OF_ORDER, ISSUE_DATE, OBID, PID, AID



Manage:

UNF:

OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, MID,
M_NAME, EMAIL_ID, PHONE, H_NO, STREET, CITY

1NF: PHONE is a multivalued attribute.

OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY, MID,
M_NAME, EMAIL_ID, H_NO, STREET, CITY

2NF:

1. OBID, OB_NAME, ORGAN_TYPE, H_NO, STREET, CITY
2. PID, PHONE
3. MID, M_NAME, EMAIL_ID, H_NO, STREET, CITY, OBID, PID

3NF:

1. OBID, OB_NAME, ORGAN_TYPE, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. MID, M_NAME, EMAIL_ID, OBID, PID, AID

TOTAL TABLE

1. DID, D_NAME, AGE, BLOOD_GROUP, GENDER, OID, PID, AID
2. PID, PHONE
3. AID, H_NO, STREET, CITY
4. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE
5. DID, D_NAME, AGE, BLOOD_GROUP, RID, PID, AID
6. PID, PHONE
7. AID, H_NO, STREET, CITY
8. RID, R_NAME, PID, AID
9. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID
10. OBID, OB_NAME, ORGAN_TYPE, AID
11. AID, H_NO, STREET, CITY
12. OBID, OB_NAME, ORGAN_TYPE, AID
13. PID, PHONE
14. AID, H_NO, STREET, CITY
15. RID, R_NAME, OBID, PID, AID
16. OBID, OB_NAME, ORGAN_TYPE, AID
17. PID, PHONE
18. AID, H_NO, STREET, CITY
19. HID, H_NAME, NO_OF_ORDER, ISSUE_DATE, OBID, PID, AID
20. OBID, OB_NAME, ORGAN_TYPE, AID
21. PID, PHONE
22. AID, H_NO, STREET, CITY
23. MID, M_NAME, EMAIL_ID, OBID, PID, AID

FINALIZATION

1. PID, PHONE
2. AID, H_NO, STREET, CITY
3. OBID, OB_NAME, AID
4. OID, BLOOD_GROUP, ORGAN_TYPE, OS_DATE, OBID
5. HID, H_NAME, NO_OF_ORDER, ISSUE_DATE, OBID, PID, AID
6. RID, R_NAME, OBID, PID, AID
7. DID, D_NAME, AGE, BLOOD_GROUP, GENDER, OID, PID, AID
8. MID, M_NAME, EMAIL_ID, OBID, PID, AID

Constraints

1. PHONE_NUMBER:

Column Name	Data Type	Constraints
PID	VARCHAR2(5)	PRIMARY KEY
PHONE	VARCHAR2(11)	NOT NULL

2. ADDRESS:

Column Name	Data Type	Constraints
AID	VARCHAR2(5)	PRIMARY KEY
H_NO	NUMBER (4,0)	NOT NULL
STREET_NUMBER	NUMBER (3,0)	NOT NULL
CITY	VARCHAR2(255)	NOT NULL

3. ORGAN_BANK:

Column Name	Data Type	Constraints
OBID	VARCHAR2(6)	PRIMARY KEY
OB_NAME	VARCHAR2(15)	NOT NULL
AID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from "ADDRESS"

4. ORGAN:

Column Name	Data Type	Constraints
OID	VARCHAR2(5)	PRIMARY KEY
BLOOD_GROUP	VARCHAR2(3)	CHECK, NOT NULL
ORGAN_TYPE	VARCHAR2(255)	NOT NULL
OS_DATE	DATE	NOT NULL
OBID	VARCHAR2(6)	NOT NULL, FOREIGN KEY from "ORGAN_BANK"

5. HOSPITAL:

Column Name	Data Type	Constraints
HID	VARCHAR2(5)	PRIMARY KEY
H_NAME	VARCHAR2(255)	NOT NULL
NO_OF_ORDER	NUMBER (3)	NOT NULL
ISSUE_DATE	DATE	NOT NULL
OBID	VARCHAR2(6)	NOT NULL, FOREIGN KEY from “ORGAN_BANK”
PID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “PHONE_NUMBER”
AID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “ADDRESS”

6. RECIPTIONIST:

Column Name	Data Type	Constraints
RID	VARCHAR2(5)	PRIMARY KEY
R_NAME	VARCHAR2(255)	NOT NULL
OBID	VARCHAR2(6)	NOT NULL, FOREIGN KEY from “ORGAN_BANK”
PID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “PHONE_NUMBER”
AID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “ADDRESS”

7. DONOR

Column Name	Data Type	Constraints
DID	VARCHAR2(5)	PRIMARY KEY
D_NAME	VARCHAR2(255)	DEFAULT
AGE	NUMBER (3)	NOT NULL
BLOOD_GROUP	VARCHAR2(3)	CHECK, NOT NULL
GENDER	VARCHAR2(255)	CHECK, NOT NULL
OID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “ORGAN”
PID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “PHONE_NUMBER”
AID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “ADDRESS”

8. MANAGER:

Column Name	Data Type	Constraints
MID	VARCHAR2(5)	PRIMARY KEY
M_NAME	VARCHAR2(255)	NOT NULL
EMAIL_ID	VARCHAR2(255)	NOT NULL
OBID	VARCHAR2(6)	NOT NULL, FOREIGN KEY from “ORGAN_BANK”
PID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “PHONE_NUMBER”
AID	VARCHAR2(5)	NOT NULL, FOREIGN KEY from “ADDRESS”

TABLE CREATION

1. PHONE_NUMBER:

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

```
CREATE TABLE PHONE_NUMBER
(
    PID VARCHAR2(5) PRIMARY KEY,
    PHONE VARCHAR2(11) NOT NULL
);
DESC PHONE_NUMBER;
```

Below the code, the results of the DESCRIBE command are shown in a table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PHONE_NUMBER	PID	Varchar2	5	-	-	1	-	-	-
PHONE_NUMBER	PHONE	Varchar2	11	-	-	-	-	-	-

At the bottom of the interface, the status bar indicates "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved".

2. ADDRESS:

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

```
CREATE TABLE ADDRESS
(
    AID VARCHAR2(5) PRIMARY KEY,
    H_NO NUMBER (4,0) NOT NULL,
    STREET NUMBER (3,0) NOT NULL,
    CITY VARCHAR2 (255) NOT NULL
);
DESC ADDRESS;
```

Below the code, the results of the DESCRIBE command are shown in a table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESS	AID	Varchar2	5	-	-	1	-	-	-
ADDRESS	H_NO	Number	-	4	0	-	-	-	-
ADDRESS	STREET	Number	-	3	0	-	-	-	-
ADDRESS	CITY	Varchar2	255	-	-	-	-	-	-

At the bottom of the interface, the status bar indicates "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved".

3. ORGAN_BANK:

The screenshot shows the Oracle Application Express interface. In the top navigation bar, it says "User SCOTT". Below that, the path "Home > SQL > SQL Commands" is shown. The main area contains the following SQL code:

```
CREATE TABLE ORGAN_BANK
(
    OBID VARCHAR2(6) PRIMARY KEY,
    OB_NAME VARCHAR2(255) NOT NULL,
    AID VARCHAR2(5) NOT NULL,
    FOREIGN KEY (AID) REFERENCES ADDRESS (AID)
);
```

Below the code, there is a blue link labeled "DESC ORGAN_BANK;". At the bottom of the SQL editor, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Describe" tab is selected. The results show the table structure:

Object Type	TABLE Object	ORGAN_BANK							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORGAN_BANK	OBID	VARCHAR2	6	-	-	1	-	-	-
	OB_NAME	VARCHAR2	255	-	-	-	-	-	-
	AID	VARCHAR2	5	-	-	-	-	-	-

At the bottom right of the interface, it says "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".

4. ORGAN:

The screenshot shows the Oracle Application Express interface. In the top navigation bar, it says "User SCOTT". Below that, the path "Home > SQL > SQL Commands" is shown. The main area contains the following SQL code:

```
CREATE TABLE ORGAN
(
    OID VARCHAR2 (5) PRIMARY KEY,
    BLOOD_GROUP VARCHAR2 (3) CHECK (BLOOD_GROUP IN ('A+', 'B+', 'O+', 'AB+', 'A-', 'B-', 'O-', 'AB-')) NOT NULL,
    ORGAN_TYPE VARCHAR2 (255) NOT NULL,
    OS_DATE DATE NOT NULL,
    OBID VARCHAR2(6) NOT NULL,
    FOREIGN KEY (OBID) REFERENCES ORGAN_BANK (OBID)
);
```

Below the code, there is a blue link labeled "DESC ORGAN;". At the bottom of the SQL editor, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Describe" tab is selected. The results show the table structure:

Object Type	TABLE Object	ORGAN							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORGAN	OID	VARCHAR2	5	-	-	1	-	-	-
	BLOOD_GROUP	VARCHAR2	3	-	-	-	-	-	-
	ORGAN_TYPE	VARCHAR2	255	-	-	-	-	-	-
	OS_DATE	Date	7	-	-	-	-	-	-
	OBID	VARCHAR2	6	-	-	-	-	-	-

At the bottom right of the interface, it says "Application Express 2.1.0.0.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".

5. HOSPITAL:

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are three tabs labeled "SQL Commands". The middle tab is active and contains the SQL code for creating the HOSPITAL table. The bottom-right pane shows the table structure with columns HID, H_NAME, NO_OF_ORDER, ISSUE_DATE, OBID, PID, and AID. The bottom-left pane shows the system tray with weather information (21°C, Haze) and system icons.

```

CREATE TABLE HOSPITAL
(
    HID VARCHAR2 (5) PRIMARY KEY NOT NULL,
    H_NAME VARCHAR2 (255) NOT NULL,
    NO_OF_ORDER NUMBER (3) NOT NULL,
    ISSUE_DATE DATE NOT NULL,
    OBID VARCHAR2(6) NOT NULL,
    PID VARCHAR2 (5) NOT NULL,
    AID VARCHAR2 (5) NOT NULL,
    FOREIGN KEY (OBID) REFERENCES ORGAN_BANK (OBID),
    FOREIGN KEY (PID) REFERENCES PHONE_NUMBER (PID),
    FOREIGN KEY (AID) REFERENCES ADDRESS (AID)
);
DESC HOSPITAL;

```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HOSPITAL	HID	Varchar2	5	-	-	1	-	-	-
	H_NAME	Varchar2	255	-	-	-	-	-	-
	NO_OF_ORDER	Number	-	3	0	-	-	-	-
	ISSUE_DATE	Date	7	-	-	-	-	-	-
	OBID	Varchar2	6	-	-	-	-	-	-
	PID	Varchar2	5	-	-	-	-	-	-
	AID	Varchar2	5	-	-	-	-	-	-

1 - 7

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved. 11:13 PM 12/24/2022

6. RECEPTIONIST:

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are three tabs labeled "SQL Commands". The middle tab is active and contains the SQL code for creating the RECEPTIONIST table. The bottom-right pane shows the table structure with columns RID, R_NAME, OBID, PID, and AID. The bottom-left pane shows the system tray with weather information (21°C, Haze) and system icons.

```

CREATE TABLE RECEPTIONIST
(
    RID VARCHAR2(5) PRIMARY KEY,
    R_NAME VARCHAR2 (255) NOT NULL,
    OBID VARCHAR2(6) NOT NULL,
    PID VARCHAR2 (5) NOT NULL,
    AID VARCHAR2 (5) NOT NULL,
    FOREIGN KEY (OBID) REFERENCES ORGAN_BANK (OBID),
    FOREIGN KEY (PID) REFERENCES PHONE_NUMBER (PID),
    FOREIGN KEY (AID) REFERENCES ADDRESS (AID)
);
DESC RECEPTIONIST;

```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECEPTIONIST	RID	Varchar2	5	-	-	1	-	-	-
	R_NAME	Varchar2	255	-	-	-	-	-	-
	OBID	Varchar2	6	-	-	-	-	-	-
	PID	Varchar2	5	-	-	-	-	-	-
	AID	Varchar2	5	-	-	-	-	-	-

1 - 5

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved. 10:54 PM 12/24/2022

7. DONOR:

The screenshot shows the Oracle Application Express interface with three tabs open: 'SQL Commands' (active), 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL tab contains the following code:

```

CREATE TABLE DONOR
(
    DID VARCHAR2(5) PRIMARY KEY,
    D_NAME VARCHAR2(255) NOT NULL,
    AGE NUMBER (3) NOT NULL,
    BLOOD_GROUP VARCHAR2 (3) CHECK (BLOOD_GROUP IN ('A+', 'B+', 'O+', 'AB+', 'A-', 'B-', 'O-', 'AB-')) NOT NULL,
    GENDER VARCHAR2(255) CHECK (GENDER IN ('MALE', 'FEMALE', 'OTHER')) NOT NULL,
    OID VARCHAR2 (5) NOT NULL,
    PID VARCHAR2 (5) NOT NULL,
    AID VARCHAR2 (5) NOT NULL,
    FOREIGN KEY (OID) REFERENCES ORGAN(OID),
    FOREIGN KEY (PID) REFERENCES PHONE_NUMBER (PID),
    FOREIGN KEY (AID) REFERENCES ADDRESS (AID)
);
DESC DONOR;

```

The Results tab shows the table structure:

Object Type	TABLE Object	DONOR							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DONOR	DID	VARCHAR2	5	-	-	1	-	-	-
	D_NAME	VARCHAR2	255	-	-	-	-	-	-
	AGE	NUMBER	-	3	0	-	-	-	-
	BLOOD_GROUP	VARCHAR2	3	-	-	-	-	-	-
	GENDER	VARCHAR2	255	-	-	-	-	-	-
	OID	VARCHAR2	5	-	-	-	-	-	-
	PID	VARCHAR2	5	-	-	-	-	-	-
	AID	VARCHAR2	5	-	-	-	-	-	-

The status bar at the bottom right indicates '9:39 PM 12/24/2022'.

8. MANAGER:

The screenshot shows the Oracle Application Express interface with three tabs open: 'SQL Commands' (active), 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL tab contains the following code:

```

CREATE TABLE MANAGER
(
    MID VARCHAR2 (5) PRIMARY KEY NOT NULL,
    M_NAME VARCHAR2 (255) NOT NULL,
    EMAIL_ID VARCHAR2 (255) NOT NULL,
    OBID VARCHAR2(6) NOT NULL,
    PID VARCHAR2 (5) NOT NULL,
    AID VARCHAR2 (5) NOT NULL,
    FOREIGN KEY (OBID) REFERENCES ORGAN_BANK (OBID),
    FOREIGN KEY (PID) REFERENCES PHONE_NUMBER (PID),
    FOREIGN KEY (AID) REFERENCES ADDRESS (AID)
);
DESC MANAGER;

```

The Results tab shows the table structure:

Object Type	TABLE Object	MANAGER							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MID	VARCHAR2	5	-	-	1	-	-	-
	M_NAME	VARCHAR2	255	-	-	-	-	-	-
	EMAIL_ID	VARCHAR2	255	-	-	-	-	-	-
	OBID	VARCHAR2	6	-	-	-	-	-	-
	PID	VARCHAR2	5	-	-	-	-	-	-
	AID	VARCHAR2	5	-	-	-	-	-	-

The status bar at the bottom right indicates '11:11 PM 12/24/2022'.

DATA INSERTION

1. PHONE_NUMBER:

The screenshot shows the Oracle SQL Developer interface with a SQL Commands window. The code entered is an INSERT statement into the PHONE_NUMBER table, followed by a SELECT * FROM PHONE_NUMBER query. The results show 12 rows of data inserted, each consisting of a PID and a PHONE number.

```
INSERT INTO PHONE_NUMBER VALUES ('P-112', '01552308315')
INSERT INTO PHONE_NUMBER VALUES ('P-113', '01552308316')
INSERT INTO PHONE_NUMBER VALUES ('P-114', '01552308317')
INSERT INTO PHONE_NUMBER VALUES ('P-115', '01552308318')
INSERT INTO PHONE_NUMBER VALUES ('P-116', '01552308319')
INSERT INTO PHONE_NUMBER VALUES ('P-117', '01552308320')
INSERT INTO PHONE_NUMBER VALUES ('P-118', '01552308321')
INSERT INTO PHONE_NUMBER VALUES ('P-119', '01552308322')
INSERT INTO PHONE_NUMBER VALUES ('P-120', '01552308323')
INSERT INTO PHONE_NUMBER VALUES ('P-121', '01552308324')

SELECT *
FROM PHONE_NUMBER;
```

PID	PHONE
P-114	01552308317
P-115	01552308318
P-116	01552308319
P-117	01552308320
P-118	01552308321
P-119	01552308322
P-120	01552308323
P-121	01552308324
P-101	01456789011
P-102	01710923591

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

10 rows returned in 0.01 seconds CSV Export

2. ADDRESS:

The screenshot shows the Oracle SQL Developer interface with a SQL Commands window. The code entered is an INSERT statement into the ADDRESS table, followed by a SELECT * FROM ADDRESS query. The results show 12 rows of data inserted, each consisting of an AID, H_NO, STREET, and CITY.

```
INSERT INTO ADDRESS VALUES ('A-113', '5001', '601', 'RAJSHAHI')
INSERT INTO ADDRESS VALUES ('A-114', '5002', '609', 'RAJSHAHI')
INSERT INTO ADDRESS VALUES ('A-115', '5003', '609', 'RAJSHAHI')
INSERT INTO ADDRESS VALUES ('A-116', '5004', '618', 'RAJSHAHI')
INSERT INTO ADDRESS VALUES ('A-117', '2016', '411', 'DHAKA')
INSERT INTO ADDRESS VALUES ('A-118', '7001', '801', 'CHOTTOGRAM')
INSERT INTO ADDRESS VALUES ('A-119', '7008', '809', 'CHOTTOGRAM')
INSERT INTO ADDRESS VALUES ('A-120', '2018', '425', 'DHAKA')
INSERT INTO ADDRESS VALUES ('A-121', '2017', '428', 'DHAKA')
INSERT INTO ADDRESS VALUES ('A-122', '2022', '420', 'DHAKA')

SELECT *
FROM ADDRESS;
```

AID	H_NO	STREET	CITY
A-101	2001	401	DHAKA
A-102	2002	402	DHAKA
A-103	2003	401	DHAKA
A-104	2004	403	DHAKA
A-105	2005	401	DHAKA
A-106	2006	404	DHAKA
A-107	2007	404	DHAKA
A-108	2008	401	DHAKA
A-109	2009	405	DHAKA
A-110	2010	406	DHAKA

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

10 rows returned in 0.00 seconds CSV Export

3. ORGAN_BANK:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
INSERT INTO ORGAN_BANK VALUES ('OB-101', 'Life Angel Organ Bank', 'A-101')
SELECT * FROM ORGAN BANK;
```

The results show one row inserted:

OBID	OB_NAME	AID
OB-101	Life Angel Organ Bank	A-101

1 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.

4. ORGAN:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
INSERT INTO ORGAN VALUES ('O-101', 'A+', 'LIVER', '12-MAY-18', 'OB-101')
INSERT INTO ORGAN VALUES ('O-102', 'B+', 'EYE', '14-JUL-18', 'OB-101')
INSERT INTO ORGAN VALUES ('O-103', 'A-', 'KIDNEY', '18-JAN-19', 'OB-101')
INSERT INTO ORGAN VALUES ('O-104', 'O+', 'HEART', '18-JAN-19', 'OB-101')
INSERT INTO ORGAN VALUES ('O-105', 'AB+', 'LUNGS', '24-AUG-19', 'OB-101')
INSERT INTO ORGAN VALUES ('O-106', 'O-', 'PANCREAS', '17-DEC-19', 'OB-101')
INSERT INTO ORGAN VALUES ('O-107', 'B-', 'INTESTINE', '01-JAN-20', 'OB-101')

SELECT * FROM ORGAN
```

The results show seven rows inserted:

OID	BLOOD_GROUP	ORGAN_TYPE	OS_DATE	OBID
O-101	A+	LIVER	12-MAY-18	OB-101
O-102	B+	EYE	14-JUL-18	OB-101
O-103	A-	KIDNEY	18-JAN-19	OB-101
O-104	O+	HEART	18-JAN-19	OB-101
O-105	AB+	LUNGS	24-AUG-19	OB-101
O-106	O-	PANCREAS	17-DEC-19	OB-101
O-107	B-	INTESTINE	01-JAN-20	OB-101

7 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.

5. HOSPITAL:

User SCOTT
Home > SQL > SQL Commands

```
SQL Commands
127.0.1.8080/apex/f?p=4500:1003:3158032473254069:NO::
```

Autocommit

```
INSERT INTO HOSPITAL VALUES ('H-101', 'Southeast Hospital', 2, '10-FEB-18', 'OB-101', 'P-115', 'A-107')
INSERT INTO HOSPITAL VALUES ('H-102', 'Northern Hospital', 1, '07-JUN-18', 'OB-101', 'P-116', 'A-108')
INSERT INTO HOSPITAL VALUES ('H-103', 'XYZ Hospital', 1, '18-DEC-18', 'OB-101', 'P-117', 'A-109')
INSERT INTO HOSPITAL VALUES ('H-104', 'ABCD Hospital', 2, '17-JAN-19', 'OB-101', 'P-118', 'A-110')
INSERT INTO HOSPITAL VALUES ('H-105', 'ALPHA Hospital', 1, '16-AUG-19', 'OB-101', 'P-119', 'A-111')
INSERT INTO HOSPITAL VALUES ('H-106', 'GAMMA Hospital', 1, '19-SEP-19', 'OB-101', 'P-120', 'A-112')

SELECT *
FROM HOSPITAL;
```

HID	H_NAME	NO_OF_ORDER	ISSUE_DATE	OBID	PID	AID
H-101	Southeast Hospital	2	10-FEB-18	OB-101	P-115	A-107
H-102	Northern Hospital	1	07-JUN-18	OB-101	P-116	A-108
H-103	XYZ Hospital	1	18-DEC-18	OB-101	P-117	A-109
H-104	ABCD Hospital	2	17-JAN-19	OB-101	P-118	A-110
H-105	ALPHA Hospital	1	16-AUG-19	OB-101	P-119	A-111
H-106	GAMMA Hospital	1	19-SEP-19	OB-101	P-120	A-112

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

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

6. RECEPTIONIST:

User SCOTT
Home > SQL > SQL Commands

Autocommit

```
SQL Commands
127.0.1.8080/apex/f?p=4500:1003:3158032473254069:NO::
```

ORACLE® Database Express Edition

```
INSERT INTO RECEPTIONIST VALUES ('R-101', 'ALINA AMIN', 'OB-101', 'P-111', 'A-103')
INSERT INTO RECEPTIONIST VALUES ('R-102', 'RAKIB KHAN', 'OB-101', 'P-112', 'A-104')
INSERT INTO RECEPTIONIST VALUES ('R-103', 'RAFSAN SABAB', 'OB-101', 'P-113', 'A-105')
INSERT INTO RECEPTIONIST VALUES ('R-104', 'ANTIK MAHMUD', 'OB-101', 'P-114', 'A-106')

SELECT *
FROM RECEPTIONIST;
```

RID	R_NAME	OBID	PID	AID
R-101	ALINA AMIN	OB-101	P-111	A-103
R-102	RAKIB KHAN	OB-101	P-112	A-104
R-103	RAFSAN SABAB	OB-101	P-113	A-105
R-104	ANTIK MAHMUD	OB-101	P-114	A-106

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

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

7. DONOR:

The screenshot shows the Oracle Application Express interface. In the top-left pane, there are three tabs: 'SQL Commands' (selected), 'Autocommit' (checkbox checked), and 'Display' (dropdown set to 10). The SQL code entered is:

```
INSERT INTO DONOR VALUES ('D-101', 'ALIMA BEGUM', 70, 'A+', 'FEMALE', 'O-101', 'P-101', 'A-118')
INSERT INTO DONOR VALUES ('D-102', 'RAHIM KHAN', 81, 'B+', 'MALE', 'O-102', 'P-102', 'A-117')
INSERT INTO DONOR VALUES ('D-103', 'ATEF ALI', 64, 'O-', 'MALE', 'O-106', 'P-103', 'A-119')
INSERT INTO DONOR VALUES ('D-104', 'MAHABUB ALOM', 53, 'B-', 'MALE', 'O-107', 'P-104', 'A-120')
INSERT INTO DONOR VALUES ('D-105', 'MOURI MORION', 75, 'A+', 'FEMALE', 'O-101', 'P-105', 'A-121')
INSERT INTO DONOR VALUES ('D-106', 'ALIF IMRAN', 68, 'AB+', 'MALE', 'O-105', 'P-107', 'A-122')
INSERT INTO DONOR VALUES ('D-107', 'ABU RAYHAN', 73, 'A+', 'MALE', 'O-103', 'P-108', 'A-113')
INSERT INTO DONOR VALUES ('D-108', 'AZMAL FUAD', 77, 'B+', 'MALE', 'O-102', 'P-108', 'A-114')
INSERT INTO DONOR VALUES ('D-109', 'AZIZ AHMED', 71, 'A+', 'MALE', 'O-101', 'P-109', 'A-115')
INSERT INTO DONOR VALUES ('D-110', 'SAIF ALI', 80, 'O+', 'MALE', 'O-104', 'P-110', 'A-116')
INSERT INTO DONOR VALUES ('D-111', 'ALIMA BEGUM', 70, 'A+', 'FEMALE', 'O-101', 'P-101', 'A-118')

SELECT *
FROM DONOR
```

In the bottom-left pane, the results are displayed in a table:

DID	D_NAME	AGE	BLOOD_GROUP	GENDER	OID	PID	AID
D-102	RAHIM KHAN	81	B+	MALE	O-102	P-102	A-117
D-103	ATEF ALI	64	O-	MALE	O-103	P-103	A-119
D-104	MAHABUB ALOM	53	B-	MALE	O-107	P-104	A-120
D-105	MOURI MORION	75	A+	FEMALE	O-101	P-105	A-121
D-106	ALIF IMRAN	68	AB+	MALE	O-105	P-107	A-122
D-107	ABU RAYHAN	73	A+	MALE	O-103	P-107	A-113
D-108	AZMAL FUAD	77	B+	MALE	O-102	P-108	A-114
D-109	AZIZ AHMED	71	A+	MALE	O-101	P-109	A-115
D-110	SAIF ALI	80	O+	MALE	O-104	P-110	A-116
D-111	ALIMA BEGUM	70	A+	FEMALE	O-101	P-101	A-118

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

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

8. MANAGER:

The screenshot shows the Oracle Database Express Edition interface. In the top-left pane, there are three tabs: 'SQL Commands' (selected), 'Autocommit' (checkbox checked), and 'Display' (dropdown set to 10). The SQL code entered is:

```
INSERT INTO MANAGER VALUES ('M-101', 'ABDUR RAHMAN', 'manager.laob@gmail.com', 'OB-101', 'P-121', 'A-102')

SELECT *
FROM MANAGER;
```

In the bottom-left pane, the results are displayed in a table:

MID	M_NAME	EMAIL_ID	OBID	PID	AID
M-101	ABDUR RAHMAN	manager.laob@gmail.com	OB-101	P-121	A-102

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

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

QUERY TEST

Basic Query

1. Show the address info whose location (city) in CHOTTAGRAM.

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

```
SELECT *
FROM ADDRESS
WHERE CITY = 'CHOTTOGRAM';
```

The results pane displays a table with three columns: AID, H_NO, and STREET. The data shows two rows where the CITY is 'CHOTTOGRAM'.

AID	H_NO	STREET
A-118	7001	801
A-119	7008	809

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

At the bottom of the interface, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved".

2. Show organ id, organ type, organ stored date where heart doesn't exist

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

```
SELECT OID, ORGAN_TYPE, OS_DATE
FROM ORGAN
WHERE ORGAN_TYPE NOT IN 'HEART'
```

The results pane displays a table with three columns: OID, ORGAN_TYPE, and OS_DATE. The data shows seven rows of organs other than 'HEART'.

OID	ORGAN_TYPE	OS_DATE
O-101	LIVER	12-MAY-18
O-102	EYE	14-JUL-18
O-103	KIDNEY	18-JAN-19
O-105	LUNGS	24-AUG-19
O-106	PANCREAS	17-DEC-19
O-107	INTESTINE	01-JAN-20

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

At the bottom of the interface, it says "Application Express 2.1.0.0.39 Copyright © 1999, 2008, Oracle. All rights reserved".

Sub Query

1. Display the donor id, age, blood group, gender whose age is greater than donor id = D-106

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

```
SELECT DID, AGE, BLOOD_GROUP, GENDER
FROM DONOR
WHERE AGE > (SELECT AGE
               FROM DONOR
              WHERE DID = 'D-106')
```

The Results tab displays the following data:

DID	AGE	BLOOD_GROUP	GENDER
D-102	81	B+	MALE
D-105	75	A+	FEMALE
D-107	73	A-	MALE
D-108	77	B+	MALE
D-109	71	A+	MALE
D-110	80	O+	MALE
D-101	70	A+	FEMALE

7 rows returned in 0.00 seconds

2. Display organ id, organ type and age whose age is between 50 to 65

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

```
SELECT O_OID, O_ORGAN_TYPE, D.AGE
FROM ORGAN_O_ DONOR D
WHERE D.AGE <= ANY (SELECT AGE
                     FROM DONOR
                    WHERE AGE BETWEEN 50 AND 65)
```

The Results tab displays the following data:

O_OID	O_ORGAN_TYPE	AGE
O-101	LIVER	64
O-102	EYE	64
O-103	KIDNEY	64
O-104	HEART	64
O-105	LUNGS	64
O-106	PANCREAS	64
O-107	INTESTINE	64
O-101	LIVER	53
O-102	EYE	53
O-103	KIDNEY	53

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

10 rows returned in 0.00 seconds

Joining

1. Show the donor id, donor name, donor age those are from the city RAJSHAHI

The screenshot shows the Oracle Application Express interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:8724909175693::NO:::. The page title is "SQL Commands". The SQL query is:

```
SELECT D.DID, D.D_NAME, D.AGE, A.CITY
FROM DONOR D, ADDRESS A
WHERE D.AID = A.AID AND A.CITY = 'RAJSHAHI'
```

The results table shows four rows:

DID	D_NAME	AGE	CITY
D-107	ABU RAYHAN	73	RAJSHAHI
D-108	AZMAL FUAD	77	RAJSHAHI
D-109	AZIZ AHMED	71	RAJSHAHI
D-110	SAIF ALI	80	RAJSHAHI

4 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.

2. Show organ bank name, hospital name, issue date where issue date between 07-JUN-18 to 16-AUG-19

The screenshot shows the Oracle Application Express interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:8724909175693::NO:::. The page title is "SQL Commands". The SQL query is:

```
SELECT OB.OB_NAME, H.H_NAME, H.ISSUE_DATE
FROM HOSPITAL H, ORGAN_BANK OB
WHERE H.OBID = OB.OBID AND ISSUE_DATE BETWEEN '07-JUN-18' AND '16-AUG-19'
```

The results table shows four rows:

OB_NAME	H_NAME	ISSUE_DATE
Life Angel Organ Bank	Northern Hospital	07-JUN-18
Life Angel Organ Bank	XYZ Hospital	18-DEC-18
Life Angel Organ Bank	ABCD Hospital	17-JAN-19
Life Angel Organ Bank	ALPHA Hospital	16-AUG-19

4 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.

Create View

1. Create a view of DID, D_NAME, PID and AID from DONOR table where gender is male

```
CREATE VIEW MALE_VIEW
AS SELECT DID, D_NAME, PID, AID
FROM DONOR
WHERE GENDER = 'MALE'

DESC MALE_VIEW
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MALE_VIEW	DID	VARCHAR2	5	-	-	-	-	-	-
	D_NAME	VARCHAR2	255	-	-	-	✓	-	-
	PID	VARCHAR2	5	-	-	-	-	-	-
	AID	VARCHAR2	5	-	-	-	-	-	-

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

2. Create a view of OID, ORGAN_TYPE, and OS_DATE form ORGAN table where organ is EYE

```
CREATE VIEW ORGAN_VIEW
AS SELECT OID, ORGAN_TYPE, OS_DATE
FROM ORGAN
WHERE ORGAN_TYPE = 'EYE'

DESC ORGAN_VIEW
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORGAN_VIEW	OID	VARCHAR2	5	-	-	-	-	-	-
	ORGAN_TYPE	VARCHAR2	255	-	-	-	-	-	-
	OS_DATE	DATE	7	-	-	-	-	-	-

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

CONCLUSION

Our Database project is about Organ Bank Management System. While doing this project we had learnt many things. When we were doing ER Diagram, we faced some difficulties in arranging everything. Then comes normalization. It was a big challenge for us because there are too many rows. After completing normalization, we created our all tables. Then while inserting values we faced some difficulties with foreign keys. Also, we used Oracle 10G and it wasn't supporting multiple lines, so we had to insert every line manually. Then Query part and it was fun. Overall, we had many new experiences while doing this project. Solving every problem felt like a great achievement for us. After finishing this project, we have a clearer concept in Databases.

THANK YOU