

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH  
(AIUB)**

FACULTY OF SCIENCE & TECHNOLOGY



*COURSE TITLE*  
**INTRODUCTION TO DATABASE(CSC208)**

Semester: Fall 24-25

Section [Y]

*Project Title*  
**BLOOD MANAGEMENT SYSTEM**

*Submitted to*  
**TAMANNA ZAMAN BRISTY**  
Faculty

*Submitted by*  
**Group: GROUP-1**

NAME	ID
MD. JANNATUL FAHAD	23-54618-3
SAYED ASHFAKUL ISLAM	23-54552-3
A.R.M.ZEHAD	23-54502-3
MD.SHIHAB SHIKDAR	23-54523-3

## **TABLE OF CONTENTS**

<b>TOPICS</b>	<b>PAGE NO.</b>
<b>TITLE PAGE</b>	<b>1</b>
<b>TABLE OF CONTENT</b>	<b>2</b>
<b>1. INTRODUCTION</b>	<b>3</b>
<b>2. CASE STUDY</b>	<b>3</b>
<b>3. ER DIAGRAM</b>	<b>4</b>
<b>4. NORMALIZATION</b>	<b>5-11</b>
<b>5. FINALIZATION</b>	<b>12-13</b>
<b>6. TABLE CREATION</b>	<b>14-20</b>
<b>7. DATA INSERTION</b>	<b>21-27</b>
<b>8. QUERY TEST</b>	<b>28-35</b>
<b>9. DB CONNECTION</b>	<b>36-44</b>

## Introduction

During the **Chowk Bazar(Dhaka) Fire** Tragedy, many people lose their life because there was a shortage of blood supply, even though donors wanted to give blood they could not do it as there was not enough information available, so we made this project to make the process easier of collecting and distributing blood to the blood banks, hospitals for the victims and ensure timely access to blood in emergency situations. The efficient coordination between Blood Savers, Blood Banks, Hospitals, Patients and Donors will save lives and provide critical support to those in need during such challenging times.

## Case Study

The **Blood saver** plays a vital role in this initiative, **arranging** blood from willing **Donors** and preserving it in the **Blood Bank**. These savers have the responsibility of gathering various DBLOODGRPs from multiple donors. However, it is important to acknowledge that there may be exceptional cases where arrangers fail to obtain blood from any donors. Each **Blood saver is identified by their unique Id, multiple contact number and name**. They actively arrange blood from numerous donors. On the other hand, **donors are required to provide essential information, including their DBLOODGRP, unique donor id, DSTATUS, various contact number, name, DOB, address and DAGE**. Each individual possesses a distinctive DBLOODGRP, and donors have the option to either contribute their blood to the blood bank or decline if they feel unfit to do so. **The Blood Bank, with its name and identified by ID and location**, serves as a crucial Storage facility for the arranged blood from the donors. Blood savers supply different DBLOODGRPs to the Blood Bank based on its specific requirements. Subsequently, the Blood Bank fulfills the demand for blood by distributing different DBLOODGRPs to various hospitals. **Hospitals, identified by their ID and location, NAME, multiple number** also play a critical role in this system. They can place orders for specific DBLOODGRPs from the Blood Bank according to their immediate needs. Alternatively, **hospitals may directly gather blood from donors** if necessary. **Patients seeking medical attention visit different hospitals** for check-ups and treatments. **Each patient is identified by their unique ID and name, number, location, DAGE, gender and DBLOODGRP**, ensuring the appropriate matching of blood when required. By creating this comprehensive Blood Donor Database, the Local Authority aims to facilitate a well-organized system for managing blood donation, storage, and distribution in response to emergencies like the CHOWK BAZAR DHAKA Fire Tragedy.

## ER Diagram

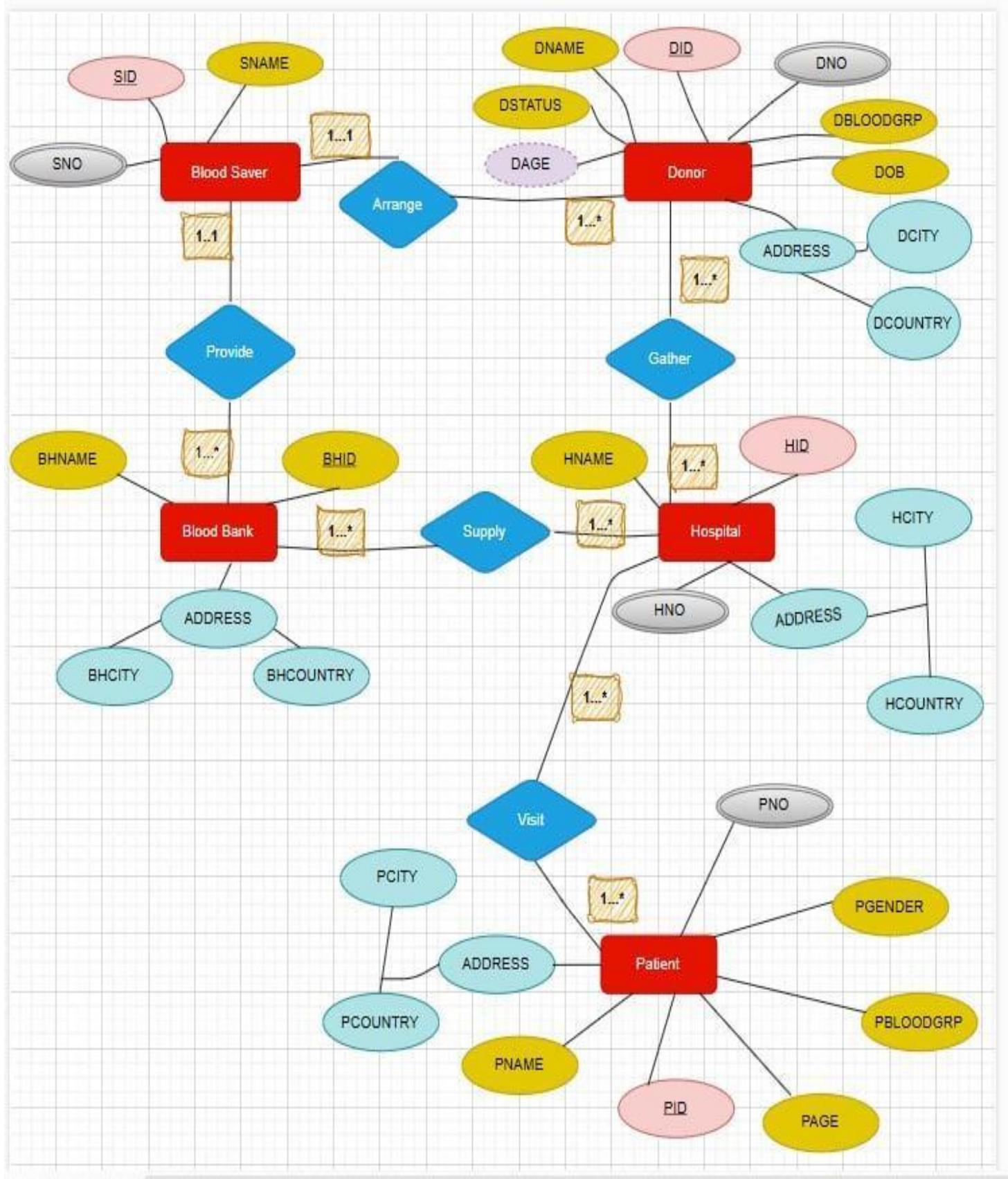


Figure 1 : Blood Donation Management System.

## Normalization

**Relation: One to Many ( A Blood Saver can arrange blood from many Donors )**

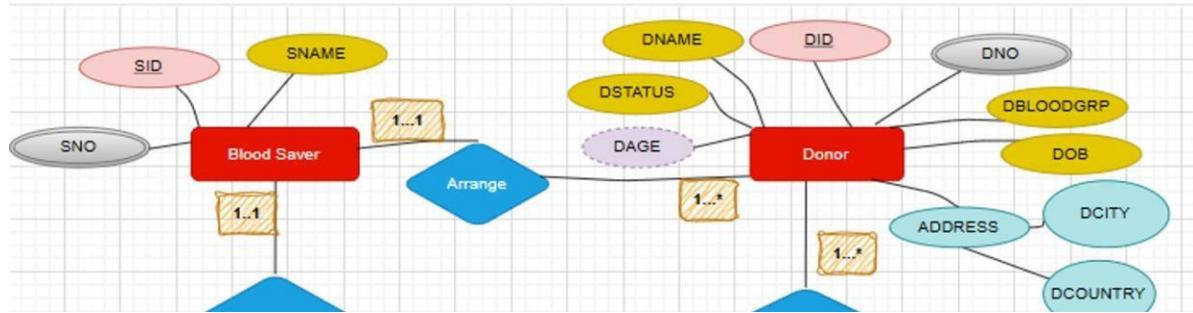


Figure 2 : ER Diagram for relation between Blood Saver and Donor.

**UNF:** SID, SNAME, DID, SNO, DNO, DNAME , DOB, DAGE, DBLOODGRP, DSTATUS, DCITY, DCOUNTRY.

**1NF:**

**Multivalued Attributes:** SNO, DNO

SID, SNAME, DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, DCITY, DCOUNTRY.

**2NF:**

1. DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, DCITY, DCOUNTRY, SID
2. SID, SNAME
3. DID, DNO
4. SID, SNO

**3NF:**

1. DID, DCITY, DCOUNTRY
2. DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID
3. SID, SNAME
4. DID, DNO
5. SID, SNO

**Relation: One-to-Many ( A blood Saver can provide blood to many Blood Bank)**

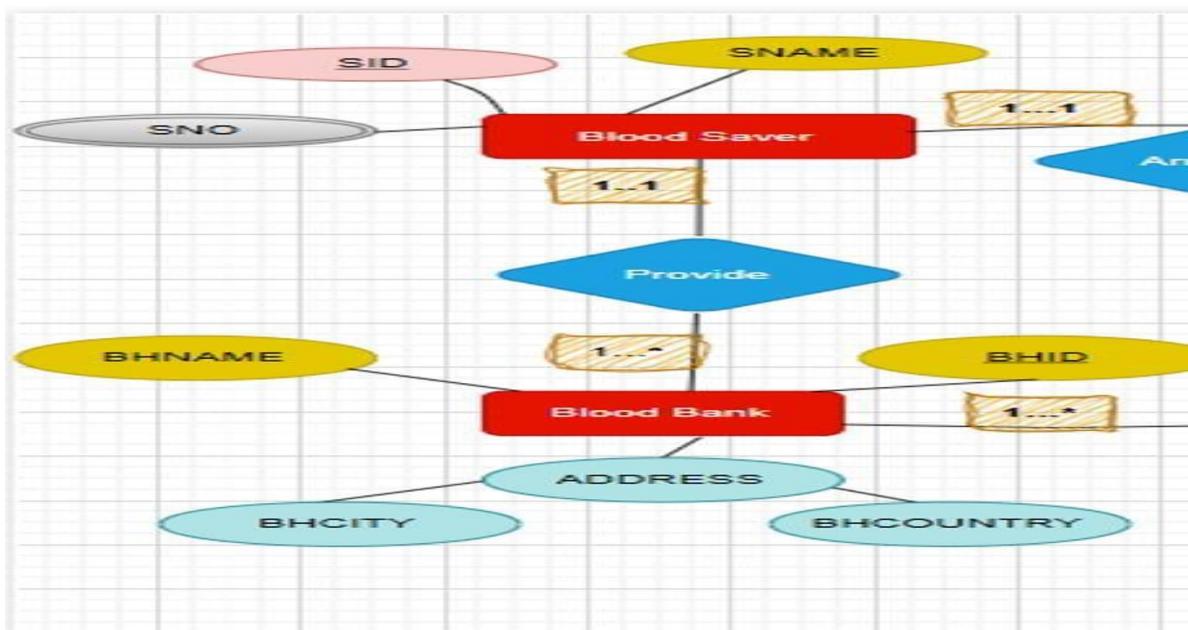


Figure 3 : ER Diagram for relation between Blood Saver and Blood Bank.

**UNF:** SID, SNAME, SNO, BHNAME, BHID, BHCITY, BHOUNTRY

**1NF:**

**Multivalued Attributes:** SNO

SID, SNAME, BHNAME, BHID, BHCITY, BHOUNTRY

**2NF:**

1. BHID, BHNAME, BHCITY, BHOUNTRY, SID

2. SID, SNAME

3. SID, SNO

**3NF:**

1. BHID, BHCITY, BHOUNTRY

2. BHID, BHNAME, SID

3. SID, SNAME

4. SID, SNO

## Relation: One-to-Many ( The Blood Bank supplies blood to Hospitals )

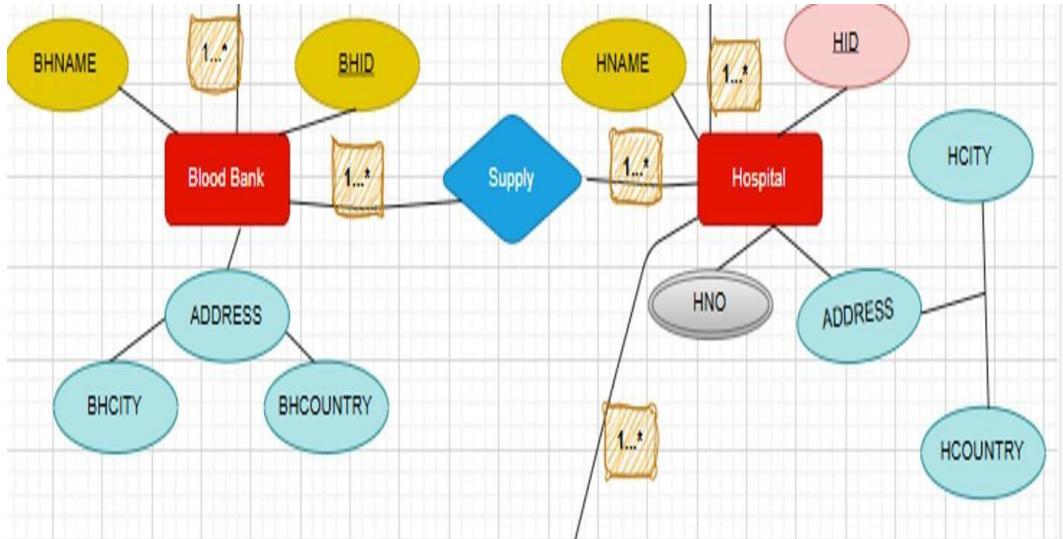


Figure 4 : Relation between Blood Bank and Hospital

**UNF:** BHNAME, **BHID**, BHCITY,BHCOUNTRY, HNAME, HNO,**HID**, HCITY, HCOUNTRY

**1NF:**

**Multivalued Attributes:** HNO

BHNAME, **BHID**, BHCITY,BHCOUNTRY,  
HNAME, **HID**, HCITY, HCOUNTRY

**2NF:**

1. **HID**, HNAME, DCITY, DCOUNTRY, **BHID**
2. **BHID**, BHNAME, BHCITY,BHCOUNTRY
3. **HID**, HNO

**3NF:**

- 1 . **HID** , HCITY,HCOUNTRY
2. **BHID**, BHCITY,BHCOUNTRY
3. **HID**, HNAME, **BHID**
4. **BHID**, BHNAME
5. **HID**, HNO

**Relation: Many-to-Many** (Patients seeking medical attention visit different hospitals for check-up and treatments)

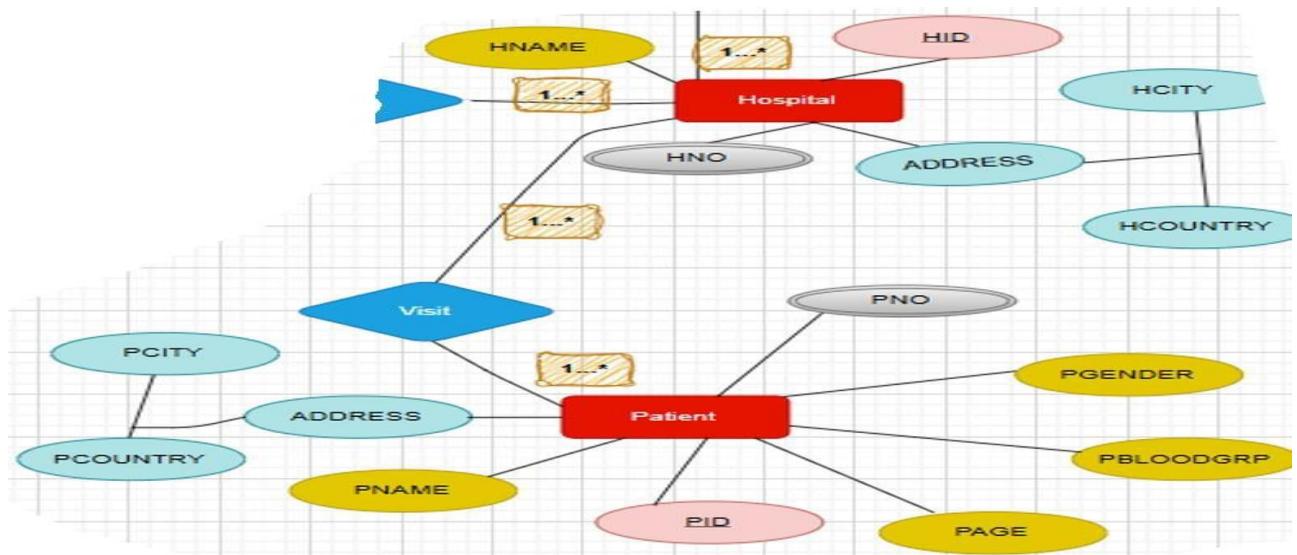


Figure 5 : ER Diagram for relation between Patient and Hospital.

**UNF:** HNAME, HID, HNO, HCITY, HCOUNTY, PNAME, PID,  
PNO, PCITY, PCOUNTRY, PBLOODGRP, PGENDER ,PAGE

**1NF:**

**Multivalued Attributes:** HNO, PNO

HNAME, HID, HCITY, HCOUNTY, PNAME, PID, PCITY, PCOUNTRY,  
PBLOODGRP ,PGENDER ,PAGE

**2NF:**

1. PID, PNAME, PCITY, PCOUNTRY, PBLOODGRP, PGENDER ,PAGE, HID
2. HID, HNAME, HCITY, HCOUNTY
3. PID, PNO
4. HID, HNO

**3NF:**

1. PID , PCITY,PCOUNTRY
2. HID, HCITY,HCOUNTY
3. PID, PNAME,PBLOODGRP,PGENDER,PAGE, HID
4. HID , HNAME,
5. PID, PNO
6. HID,HNO

**Relation: Many-to-Many (Hospitals can directly gather blood from Donors in emergency situations)**

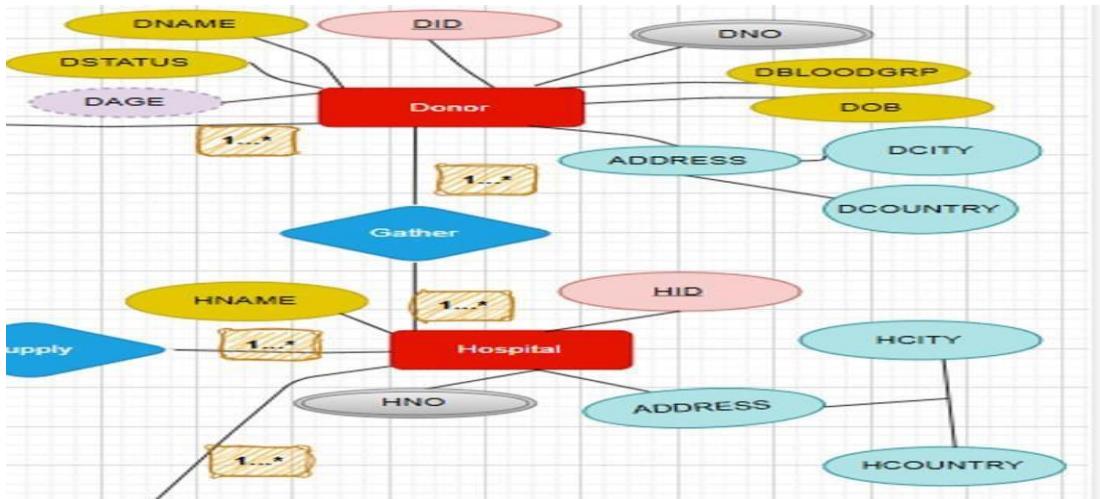


Figure 6 : ER Diagram for relation between Hospital and Donor.

**UNF:** HID, HNAME, HNO, HCITY, HCOUNTRY, DNAME, DID, DNO ,DOB, DAGE, DBLOODGROUP, DSTATUS, DCITY, DCOUNTRY

**1NF: Multivalued Attributes:** DNO, HNO

HID, HNAME, HCITY, HCOUNTRY, DNAME, DID, DOB, DAGE, DBLOODGROUP, DSTATUS, DCITY, DCOUNTRY

**2NF:**

1. **DID, DNAME, DOB, DAGE, DBLOODGROUP, DSTATUS, DCITY, DCOUNTRY, HID**
2. **HID, HNAME, HNO, HCITY, HCOUNTRY**
3. **HID, HNO**
4. **DID, DNO**

**3NF:**

1. **HID , HCITY,HCOUNTRY**
2. **DID, DCITY,DCOUNTRY**
3. **HID, HNAME**
4. **DID, DNAME,DOB,DAGE,DBLOODFROUP,DSTATUS HID**
5. **HID,HNO**
6. **DID,DNO**

## Finalization

1. ~~DID, DCITY, DCOUNTRY~~
2. ~~DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID~~
3. ~~SID, SNAME~~
4. ~~DID, DNO~~
5. ~~SID, SNO~~
6. ~~BHID, BH CITY, BH COUNTRY~~
7. ~~BHID, BH NAME, SID~~
8. ~~SID, SNAME~~
9. ~~SID, SNO~~
10. ~~HID, HCITY, HCOUNTRY~~
11. ~~BHID, BH CITY, BH COUNTRY~~
12. ~~HID, HNAME, BHID~~
13. ~~BHID, BH NAME~~
14. ~~HID, HNO~~
15. ~~PID, PCITY, PCOUNTRY~~
16. ~~HID, HCITY, HCOUNTRY~~
17. ~~PID, PNAME, PBLOODGRP, PGENDER, PAG  
E, HID~~
18. ~~HID, HNAME~~
19. ~~PID, PNO~~
20. ~~HID, HNO~~
21. ~~HID, HCITY, HCOUNTRY~~
22. ~~DID, DCITY, DCOUNTRY~~
23. ~~HID, HNAME~~
24. ~~DID,  
DNAME, DOB, DAGE, DBLOODFROUP, DTAT  
US, HID~~
25. ~~HID, HNO~~
26. ~~DID, DNO~~

## Final Table

1. DID, DCITY,DCOUNTRY
2. DID, DNAME, DOB, DAGE, DBLOODGRP,  
DSTATUS, SID,HID
3. SID, SNAME,SNO
4. DID, DNO
5. BHID, BH CITY, BH COUNTRY
6. BHID, BH NAME, SID
7. HID , HCITY,H COUNTRY
8. HID , HNO
9. PID , PCITY,PCOUNTRY
10. PID,PNAME,PBLOODGRP,PGENDER,PAGE,HID
11. HID,H NAME,BHID
12. PID,PNO

**Total finalized table: 12**

## Table Creation

StudentID1: 23-54618-3 Name: MD. JANNATUL FAHAD	StudentID3 : 23-54552-3 Name: SAYED ASHFAKUL ISLAM
StudentID2: 23-54502-3 Name: A.R.M.ZEHAD	StudentID4: 23-54523-3 Name: MD.SHIHAB SHIKDAR
<b>CO4:</b> Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
<b>PO-e-2:</b> Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

**Table 1:** Blood Saver

The screenshot shows the Oracle SQL developer interface. At the top, there are buttons for Autocommit (checked), Display (set to 10), Save, and Run. Below the buttons is a code editor containing the following SQL statements:

```
Create table Blood_Saver (SID varchar(10) primary key not null, SNAME varchar (255), SNO int);
desc Blood_Saver;
```

Below the code editor, there is a navigation bar with links: Results, Explain, Describe, Saved SQL, and History. The 'Describe' link is currently selected. Under 'Object Type', it shows 'TABLE Object BLOOD\_SAVER'. A detailed table below lists the columns:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comments
BLOOD_SAVER	SID	Varchar2	10	-	-	1	-	-	-
	SNAME	Varchar2	255	-	-	-	✓	-	-
	SNO	Number	-	-	0	-	✓	-	-

In the bottom right corner of the table area, there is a small text '1 - 3'.

**Table 2: Donor**

```
Create table Donor (DID varchar(10) not NULL Primary Key, DNAME varchar(50), DOB date, DAGE
Number(3),
DBLOODGRP varchar(5) Not Null, DSTATUS varchar(10));
desc Donor;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object DONOR

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DONOR	DID	Varchar2	10	-	-	1	-	-	-
	DNAME	Varchar2	50	-	-	-	✓	-	-
	DOB	Date	7	-	-	-	✓	-	-
	DAGE	Number	-	3	0	-	✓	-	-
	DBLOODGRP	Varchar2	5	-	-	-	-	-	-
	DSTATUS	Varchar2	10	-	-	-	✓	-	-

1 - 6

**Table 3: D\_S\_info**

```
Autocommit Display 10 Save Run
Create table D_S_info (DID varchar(10) not NULL Primary Key, DNAME varchar(50), DOB date, DAGE
Number(3),
DBLOODGRP varchar(5) Not Null, DSTATUS varchar(10), SID varchar(10));
Alter table D_S_info add constraint cons1 Foreign Key (SID) references Blood_Saver (SID);
desc D_S_info;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object D\_S\_INFO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
D_S_INFO	DID	Varchar2	10	-	-	1	-	-	-
	DNAME	Varchar2	50	-	-	-	✓	-	-
	DOB	Date	7	-	-	-	✓	-	-
	DAGE	Number	-	3	0	-	✓	-	-
	DBLOODGRP	Varchar2	5	-	-	-	-	-	-
	DSTATUS	Varchar2	10	-	-	-	✓	-	-
	SID	Varchar2	10	-	-	-	✓	-	-

1 - 7

**Table 4: Blood\_Bank**

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there is a checkbox for 'Autocommit' and a dropdown menu set to 'Display 10'. On the right side, there are 'Save' and 'Run' buttons. The main area contains the following SQL code:

```
CREATE TABLE Blood_Bank (
    BHID VARCHAR2(10) NOT NULL PRIMARY KEY,
    BHNAME VARCHAR2(20)
);

desc Blood_Bank;
```

Below the code, there is a navigation bar with links: Results, Explain, Describe, Saved SQL, and History. Under 'Object Type', it shows 'TABLE Object BLOOD\_BANK'. A detailed table follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BLOOD_BANK	BHID	Varchar2	10	-	-	1	-	-	-
	BHNAME	Varchar2	20	-	-	-	✓	-	-

1 - 2

**Table 5: BH\_ADDRESS**

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there is a checkbox for 'Autocommit' and a dropdown menu set to 'Display 10'. On the right side, there are 'Save' and 'Run' buttons. The main area contains the following SQL code:

```
Create table BH_ADDRESS (BHID varchar(10) not null Primary Key, BHCITY varchar(10), BH_COUNTRY
varchar(10));
desc BH_ADDRESS;
```

Below the code, there is a navigation bar with links: Results, Explain, Describe, Saved SQL, and History. Under 'Object Type', it shows 'TABLE Object BH\_ADDRESS'. A detailed table follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BH_ADDRESS	BHID	Varchar2	10	-	-	1	-	-	-
	BHCITY	Varchar2	10	-	-	-	✓	-	-
	BH_COUNTRY	Varchar2	10	-	-	-	✓	-	-

1 - 3

**Table 6: B\_S\_info**

The screenshot shows the MySQL Workbench interface with the following details:

- Autocommit:** Checked.
- Display:** Set to 10.
- SQL Editor Content:**

```
Create table B_S_info(BHID varchar(10) not null Primary Key, BHNAME varchar(20), SID
varchar(10));
Alter table B_S_info add constraint cons2 Foreign Key(SID) references Blood_Saver (SID);
desc B_S_info; W
```
- Buttons:** Save and Run.
- Results Tab:** Selected.
- Object Type:** TABLE Object B\_S\_INFO
- Table Definition:**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_S_INFO	BHID	Varchar2	10	-	-	1	-	-	-
	BHNAME	Varchar2	20	-	-	-	✓	-	-
	SID	Varchar2	10	-	-	-	✓	-	-
- Total Rows:** 1 - 3

**Table 7: Hospital**

The screenshot shows the MySQL Workbench interface with the following details:

- Autocommit:** Checked.
- Display:** Set to 10.
- SQL Editor Content:**

```
Create table Hospital (HID varchar(10) not null primary key, HNAME varchar(20), HNO int);
desc Hospital;
```
- Buttons:** Save and Run.
- Results Tab:** Selected.
- Object Type:** TABLE Object HOSPITAL
- Table Definition:**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
HOSPITAL	HID	Varchar2	10	-	-	1	-	-	-
	HNAME	Varchar2	20	-	-	-	✓	-	-
	HNO	Number	-	-	0	-	✓	-	-
- Total Rows:** 1 - 3

**Table 8: H\_ADDRESS**

Home > SQL > SQL Commands

Autocommit Display 10

```
Create table H_Address(HID varchar(10) not null primary key, HCITY varchar(20), HCOUNTRY VARCHAR(20));
DESC H_Address
```

Results Explain Describe Saved SQL History

Object Type TABLE Object H\_ADDRESS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
H_ADDRESS	HID	Varchar2	10	-	-	1	-	-	
	HCITY	Varchar2	20	-	-	-	✓	-	
	HCOUNTRY	Varchar2	20	-	-	-	✓	-	
									1-3

**Table 9: D\_H\_info**

Autocommit Display 10

```
Create table D_H_info (DID varchar(10) not null PRIMARY KEY, DNAME VARCHAR(50), DOB DATE,DAGE NUMBER(3),DBLOODGRP VARCHAR(5) NOT NULL, DSTATUS VARCHAR (10), HID VARCHAR (10));

ALTER TABLE D_H_info ADD CONSTRAINT CONS3 FOREIGN KEY (HID) REFERENCES Hospital (HID);
DESC D_H_info;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object D\_H\_INFO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
D_H_INFO	DID	Varchar2	10	-	-	1	-	-	
	DNAME	Varchar2	50	-	-	-	✓	-	
	DOB	Date	7	-	-	-	✓	-	
	DAGE	Number	-	3	0	-	✓	-	
	DBLOODGRP	Varchar2	5	-	-	-	-	-	
	DSTATUS	Varchar2	10	-	-	-	✓	-	
	HID	Varchar2	10	-	-	-	✓	-	
									1-7

Application Express 2.1.0.00.39

**Table 10: B\_H\_info**

Autocommit Display 10 Save Run

```
CREATE TABLE B_H_info (BHID VARCHAR (10) NOT NULL PRIMARY KEY, BHNAME VARCHAR(20), HID VARCHAR (10));
ALTER TABLE B_H_info ADD CONSTRAINT CONS4 FOREIGN KEY(HID) REFERENCES Hospital(HID);
DESC B_H_info;
```

---

Results Explain Describe Saved SQL History

Object Type TABLE Object B\_H\_INFO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
B_H_INFO	BHID	Varchar2	10	-	-	1	-	-	
	BHNAME	Varchar2	20	-	-	-	✓	-	
	HID	Varchar2	10	-	-	-	✓	-	

1 - 3

**Table 11: Patient**

Home > SQL > SQL Commands

Autocommit Display 10 ▼

```
CREATE TABLE Patient( PID VARCHAR(10) NOT NULL PRIMARY KEY, PNAME VARCHAR(20), PBLOODGRP VARCHAR (5), PGENDER VARCHAR(6));
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	PID	Varchar2	10	-	-	1	-	-	
	PNAME	Varchar2	20	-	-	-	✓	-	
	PBLOODGRP	Varchar2	5	-	-	-	✓	-	
	PGENDER	Varchar2	6	-	-	-	✓	-	

1 - 4

**Table 12: P\_ADDRESS**

Autocommit Display 10


---

```
CREATE TABLE P_ADDRESS(PID varchar(10) not null PRIMARY KEY, PCITY varchar(20), PCOUNTRY varchar(20));
desc P_ADDRESS;
```

---

Results
Explain
Describe
Saved SQL
History

---

Object Type	TABLE Object P_ADDRESS									
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
P_ADDRESS	PID	Varchar2	10	-	-	1	-	-	-	
	PCITY	Varchar2	20	-	-	-	✓	-	-	
	PCOUNTRY	Varchar2	20	-	-	-	✓	-	-	
								1 - 3		

**Table 13: P\_H\_info**

Autocommit Display 10
Save Run


---

```
CREATE TABLE P_H_info(PID VARCHAR(10) NOT NULL PRIMARY KEY, PNAME VARCHAR(20), PBLOODGRP VARCHAR (5), PGENDER VARCHAR(6), HID VARCHAR(10));
ALTER TABLE P_H_info ADD CONSTRAINT cons5 FOREIGN KEY(HID) REFERENCES Hospital(HID);
DESC P_H_info;
```

---

Results
Explain
Describe
Saved SQL
History

---

Object Type	TABLE Object P_H_INFO									
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
P_H_INFO	PID	Varchar2	10	-	-	1	-	-	-	
	PNAME	Varchar2	20	-	-	-	✓	-	-	
	PBLOODGRP	Varchar2	5	-	-	-	✓	-	-	
	PGENDER	Varchar2	6	-	-	-	✓	-	-	
	HID	Varchar2	10	-	-	-	✓	-	-	
								1 - 5		

## Data Insertion

**Table 1:** Blood Saver

Autocommit Display [10] ▾

Save
Run

```
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('45-789', 'Rahim', '01715678901');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('32-654', 'Karim', '01826547892');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('87-321', 'Hasan', '01934567890');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('29-876', 'Jamal', '01799887766');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('11-223', 'Salma', '01814567891');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('53-902', 'Amin', '01925467832');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('77-410', 'Nasir', '01676543210');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('66-304', 'Farida', '01387654321');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('90-123', 'Nayem', '01726547899');
INSERT INTO Blood_Saver (SID, SNAME, SNO) VALUES ('25-678', 'Munia', '01896543211');

SELECT * FROM Blood_Saver;
```

---

Results Explain Describe Saved SQL History

SID	SNAME	SNO
45-789	Rahim	1715678901
32-654	Karim	1826547892
87-321	Hasan	1934567890
29-876	Jamal	1799887766
11-223	Salma	1814567891
53-902	Amin	1925467832
77-410	Nasir	1676543210
66-304	Farida	1387654321
90-123	Nayem	1726547899
25-678	Munia	1896543211

10 rows returned in 0.00 seconds CSV Export

**Table 2:** Donor

Autocommit Display [10] ▾

Save
Run

Home > SQL > SQL Commands

```
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('21-345', 'Rafiqul', '12-Feb-84', 40, 'A+', 'Agree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('45-678', 'Shamima', '25-Jun-90', 33, 'AB+', 'Disagree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('38-902', 'Kamrul', '08-Aug-85', 38, 'B+', 'Agree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('19-563', 'Tariqul', '17-Sep-88', 35, 'O+', 'Disagree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('27-419', 'Nasima', '05-May-91', 32, 'B-', 'Agree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('66-210', 'Alamin', '14-Dec-83', 40, 'A-', 'Agree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('34-678', 'Sumaiya', '09-Jul-87', 36, 'AB-', 'Disagree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('78-112', 'Habib', '22-Oct-86', 37, 'B+', 'Agree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('50-003', 'Mizan', '30-Jan-92', 32, 'O+', 'Disagree');
INSERT INTO Donor (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS) VALUES ('91-876', 'Farzana', '07-Mar-89', 35, 'B-', 'Agree');

SELECT * FROM Donor;
```

---

Results Explain Describe Saved SQL History

DID	DNAME	DOB	DAGE	DBLOODGRP	DSTATUS
21-345	Rafiqul	12-FEB-84	40	A+	Agree
45-678	Shamima	25-JUN-90	33	AB+	Disagree
38-902	Kamrul	08-AUG-85	38	B+	Agree
19-563	Tariqul	17-SEP-88	35	O+	Disagree
27-419	Nasima	05-MAY-91	32	B-	Agree
66-210	Alamin	14-DEC-83	40	A-	Agree
34-678	Sumaiya	09-JUL-87	36	AB-	Disagree
78-112	Habib	22-OCT-86	37	B+	Agree
50-003	Mizan	30-JAN-92	32	O+	Disagree
91-876	Farzana	07-MAR-89	35	B-	Agree

### Table 3: D\_S\_info

Home > SQL > SQL Commands

Autocommit  ▾

```
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('21-345', 'Rafiquil', '12-Feb-84', 40, 'A+', 'Agree', '45-789');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('45-678', 'Shamima', '25-Jun-90', 33, 'AB+', 'Disagree', '32-654');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('38-902', 'Kamrul', '08-Aug-85', 38, 'B+', 'Agree', '87-321');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('19-563', 'Tariqul', '17-Sep-88', 35, 'O+', 'Disagree', '29-876');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('27-419', 'Nasima', '05-May-91', 32, 'B-', 'Agree', '11-223');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('66-210', 'Alamin', '14-Dec-83', 40, 'A-', 'Agree', '53-902');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('34-678', 'Sumaiya', '09-Jul-87', 36, 'AB-', 'Disagree', '77-410');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('78-112', 'Habib', '22-Oct-86', 37, 'B+', 'Agree', '66-304');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('50-003', 'Mizan', '30-Jan-92', 32, 'O+', 'Disagree', '90-123');
INSERT INTO D_S_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, SID) VALUES ('91-876', 'Farzana', '07-Mar-89', 35, 'B-', 'Agree', '25-678');

SELECT * FROM D_S_info;
```

Results Explain Describe Saved SQL History

DID	DNAME	DOB	DAGE	DBLOODGRP	DSTATUS	SID
21-345	Rafiquil	12-FEB-84	40	A+	Agree	45-789
45-678	Shamima	25-JUN-90	33	AB+	Disagree	32-654
38-902	Kamrul	08-AUG-85	38	B+	Agree	87-321
19-563	Tariqul	17-SEP-88	35	O+	Disagree	29-876
27-419	Nasima	05-MAY-91	32	B-	Agree	11-223
66-210	Alamin	14-DEC-83	40	A-	Agree	53-902
34-678	Sumaiya	09-JUL-87	36	AB-	Disagree	77-410
78-112	Habib	22-OCT-86	37	B+	Agree	66-304
50-003	Mizan	30-JAN-92	32	O+	Disagree	90-123
91-876	Farzana	07-MAR-89	35	B-	Agree	25-678

### Table 4: Blood\_Bank

INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('83-759', 'Sandhani Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('52-416', 'Quantum Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('96-842', 'Chinmoy Blood Center');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('31-679', 'Marvel Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('75-420', 'DMC Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('15-006', 'Labaid Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('16-007', 'CTG Blood Center');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('17-008', 'Rajshahi Blood Bank');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('18-009', 'Sylhet Blood Care');
INSERT INTO Blood\_Bank (BHID, BHNAME) VALUES ('19-100', 'Evercare Blood Home');

SELECT \* FROM Blood\_Bank;

Results Explain Describe Saved SQL History

BHID	BHNAME
83-759	Sandhani Blood Bank
52-416	Quantum Blood Bank
96-842	Chinmoy Blood Center
31-679	Marvel Blood Bank
75-420	DMC Blood Bank
15-006	Labaid Blood Bank
16-007	CTG Blood Center
17-008	Rajshahi Blood Bank
18-009	Sylhet Blood Care
19-100	Evercare Blood Home

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

## Table 5: BH\_ADDRESS

Autocommit Display 10		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('83-759', 'Dhaka', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('52-416', 'Chattogram', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('96-842', 'Sylhet', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('31-679', 'Khulna', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('75-420', 'Rajshahi', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('15-006', 'Barishal', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('16-007', 'Mymensingh', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('17-008', 'Rangpur', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('18-009', 'Comilla', 'Bangladesh');		
INSERT INTO BH_ADDRESS (BHID, BH CITY, BH COUNTRY) VALUES ('19-100', 'Tangail', 'Bangladesh');		
<b>SELECT * FROM BH_ADDRESS;</b>		

Results Explain Describe Saved SQL History

BHID	BH CITY	BH COUNTRY
83-759	Dhaka	Bangladesh
52-416	Chattogram	Bangladesh
96-842	Sylhet	Bangladesh
31-679	Khulna	Bangladesh
75-420	Rajshahi	Bangladesh
15-006	Barishal	Bangladesh
16-007	Mymensingh	Bangladesh
17-008	Rangpur	Bangladesh
18-009	Comilla	Bangladesh
19-100	Tangail	Bangladesh

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

## Table 6: B\_S\_info

Autocommit Display 10		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('83-759', 'Sandhani Blood Bank', '45-789');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('52-416', 'Quantum Blood Bank', '32-654');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('96-842', 'Chinmoy Blood Center', '87-321');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('31-679', 'Marvel Blood Bank', '29-876');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('75-420', 'DMC Blood Bank', '11-223');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('15-006', 'Labaid Blood Bank', '53-902');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('16-007', 'CTG Blood Center', '77-410');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('17-008', 'Rajshahi Blood Bank', '66-304');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('18-009', 'Sylhet Blood Care', '90-123');		
INSERT INTO B_S_info (BHID, BHNAME, SID) VALUES ('19-100', 'Evercare Blood Home', '25-678');		
<b>SELECT * FROM B_S_info;</b>		

Results Explain Describe Saved SQL History

BHID	BHNAME	SID
83-759	Sandhani Blood Bank	45-789
52-416	Quantum Blood Bank	32-654
96-842	Chinmoy Blood Center	87-321
31-679	Marvel Blood Bank	29-876
75-420	DMC Blood Bank	11-223
15-006	Labaid Blood Bank	53-902
16-007	CTG Blood Center	77-410
17-008	Rajshahi Blood Bank	66-304
18-009	Sylhet Blood Care	90-123
19-100	Evercare Blood Home	25-678

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

**Table 7: Hospital**

Results Explain Describe Saved SQL History		
<b>HID</b>	<b>HNAME</b>	<b>HNO</b>
20200	Labaid Hospital	1713333777
75420	Dhaka Hospital	1711123456
10582	Square Hospitals	1713147852
69318	Apollo Hospitals	1713147852
23764	United Hospital	1911888000
80173	Evercare Hospital	1815000111
20100	Bangladesh Hospital	1711996633
20300	Popular Hospital	1711808080
20400	Ibn Sina Hospital	1715108008
20500	Shodesh Hospital	1678901234

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

**Table 8: H\_ADDRESS**

Results Explain Describe Saved SQL History		
<b>HID</b>	<b>HCITY</b>	<b>HOUNTRY</b>
75420	Dhaka	Bangladesh
10582	Chittagong	Bangladesh
69318	Khulna	Bangladesh
23764	Rajshahi	Bangladesh
80173	Sylhet	Bangladesh
20100	Barishal	Bangladesh
20200	Rangpur	Bangladesh
20300	Mymensingh	Bangladesh
20400	Comilla	Bangladesh
20500	Narayanganj	Bangladesh

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

## Table 9: D\_H\_info

Autocommit Display 10 Save Run

```
INSERT INTO D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) VALUES('21-345', 'Rafiquil', '12-Feb-84', '40', 'A+', 'Agree', '75420');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('45-678', 'Shamima', '25-Jun-90', 33, 'AB+', 'Disagree', '10582');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('38-902', 'Kamrul', '08-Aug-85', 38, 'B+', 'Agree', '69318');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('19-563', 'Tariqul', '17-Sep-88', 35, 'O+', 'Disagree', '23764');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('27-419', 'Nasima', '05-May-91', 32, 'B-', 'Agree', '80173');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('66-210', 'Alamin', '14-Dec-83', 40, 'A-', 'Agree', '20100');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('34-678', 'Sumaiya', '09-Jul-87', 36, 'AB-', 'Disagree', '20200');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('78-112', 'Habib', '22-Oct-86', 37, 'B+', 'Agree', '20300');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('50-003', 'Mizan', '30-Jan-92', 32, 'O+', 'Disagree', '20400');
insert into D_H_info (DID, DNAME, DOB, DAGE, DBLOODGRP, DSTATUS, HID) values ('91-876', 'Farzana', '07-Mar-89', 35, 'B-', 'Agree', '20500');

select * from D_H_info;
```

Results Explain Describe Saved SQL History

DID	DNAME	DOB	DAGE	DBLOODGRP	DSTATUS	HID
21-345	Rafiquil	12-FEB-84	40	A+	Agree	75420
45-678	Shamima	25-JUN-90	33	AB+	Disagree	10582
38-902	Kamrul	08-AUG-85	38	B+	Agree	69318
19-563	Tariqul	17-SEP-88	35	O+	Disagree	23764
27-419	Nasima	05-MAY-91	32	B-	Agree	80173
66-210	Alamin	14-DEC-83	40	A-	Agree	20100
34-678	Sumaiya	09-JUL-87	36	AB-	Disagree	20200
78-112	Habib	22-OCT-86	37	B+	Agree	20300
50-003	Mizan	30-JAN-92	32	O+	Disagree	20400
91-876	Farzana	07-MAR-89	35	B-	Agree	20500

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

Application Express 2.1.0.00.39

## Table 10: B\_H\_info

Autocommit Display 10 Save Run

```
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('83-759', 'Sandhani Blood Bank', '75420');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('52-416', 'Quantum Blood Bank', '10582');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('96-842', 'Chinmoy Blood Center', '69318');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('31-679', 'Marvel Blood Bank', '23764');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('75-420', 'DMC Blood Bank', '80173');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('15-006', 'Labaid Blood Bank', '20100');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('16-007', 'CTG Blood Center', '20200');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('17-008', 'Rajshahi Blood Bank', '20300');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('18-009', 'Sylhet Blood Care', '20400');
INSERT INTO B_H_info (BHID, BHNAME, HID) VALUES ('19-100', 'Evercare Blood Home', '20500');

SELECT *FROM B_H_info;
```

Results Explain Describe Saved SQL History

BHID	BHNAME	HID
83-759	Sandhani Blood Bank	75420
52-416	Quantum Blood Bank	10582
96-842	Chinmoy Blood Center	69318
31-679	Marvel Blood Bank	23764
75-420	DMC Blood Bank	80173
15-006	Labaid Blood Bank	20100
16-007	CTG Blood Center	20200
17-008	Rajshahi Blood Bank	20300
18-009	Sylhet Blood Care	20400
19-100	Evercare Blood Home	20500

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

**Table 11: Patient**

```

 Autocommit Display 10 ▾
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('10-628', 'Anika', 'Male', 'AB+');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('79-538', 'Rahim', 'Male', 'A+');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('25-864', 'Sumaiya', 'Female', 'O-');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('67-931', 'Fahim', 'Female', 'B-');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('43-129', 'Ayesha', 'Female', 'A-');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('30-001', 'Nafis', 'Male', 'AB+');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('31-002', 'Taslima', 'Female', 'A+');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('32-003', 'Arif', 'Male', 'O-');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('33-004', 'Maliha', 'Female', 'B-');
INSERT INTO Patient (PID, PName, PGENDER, PBLOODGRP) values ('34-005', 'Imran', 'Female', 'A-');

select * from Patient;

```

Results Explain Describe Saved SQL History

PID	PNAME	PBLOODGRP	PGENDER
10-628	Anika	AB+	Male
79-538	Rahim	A+	Male
25-864	Sumaiya	O-	Female
67-931	Fahim	B-	Female
43-129	Ayesha	A-	Female
30-001	Nafis	AB+	Male
31-002	Taslima	A+	Female
32-003	Arif	O-	Male
33-004	Maliha	B-	Female
34-005	Imran	A-	Female

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

**Table 12: P\_ADDRESS**

```

 Autocommit Display 10 ▾
Save Run
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('10-628', 'Dhaka', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('79-538', 'Chittagong', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('25-864', 'Khulna', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('67-931', 'Rajshahi', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('43-129', 'Sylhet', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('30-001', 'Barishal', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('31-002', 'Rangpur', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('32-003', 'Mymensingh', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('33-004', 'Comilla', 'Bangladesh');
INSERT INTO P_ADDRESS (PID, PCity, PCOUNTRY) VALUES ('34-005', 'Narayanganj', 'Bangladesh');

SELECT * FROM P_ADDRESS;

```

Results Explain Describe Saved SQL History

PID	PCITY	PCOUNTRY
10-628	Dhaka	Bangladesh
79-538	Chittagong	Bangladesh
25-864	Khulna	Bangladesh
67-931	Rajshahi	Bangladesh
43-129	Sylhet	Bangladesh
30-001	Barishal	Bangladesh
31-002	Rangpur	Bangladesh
32-003	Mymensingh	Bangladesh
33-004	Comilla	Bangladesh
34-005	Narayanganj	Bangladesh

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

**Table 13: P\_H\_info**

Autocommit Display 10				
<pre> INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('10-628', 'Anika', 'Male', 'AB+', '75420'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('79-538', 'Rahim', 'Male', 'A+', '10582'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('25-864', 'Sumaiya', 'Female', 'O-', '69318'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('67-931', 'Fahim', 'Female', 'B-', '23764'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('43-129', 'Ayesha', 'Female', 'A-', '80173'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('30-001', 'Nafis', 'Male', 'AB+', '20100'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('31-002', 'Taslima', 'Female', 'A+', '20200'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('32-003', 'Arif', 'Male', 'O-', '20300'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('33-004', 'Maliha', 'Female', 'B-', '20400'); INSERT INTO P_H_info (PID, PName, PGENDER, PBLOODGRP, HID) VALUES ('34-005', 'Imran', 'Female', 'A-', '20500');  SELECT * FROM P_H_info; </pre>				
<a href="#">Results</a> <a href="#">Explain</a> <a href="#">Describe</a> <a href="#">Saved SQL</a> <a href="#">History</a>				
PID	PNAME	PBLOODGRP	PGENDER	HID
10-628	Anika	AB+	Male	75420
79-538	Rahim	A+	Male	10582
25-864	Sumaiya	O-	Female	69318
67-931	Fahim	B-	Female	23764
43-129	Ayesha	A-	Female	80173
30-001	Nafis	AB+	Male	20100
31-002	Taslima	A+	Female	20200
32-003	Arif	O-	Male	20300
33-004	Maliha	B-	Female	20400
34-005	Imran	A-	Female	20500

10 rows returned in 0.00 seconds

[CSV Export](#)

## Query Writing

### SIMPLE QUERY:

**Query 1 :** Display DNAME as Donor, DAGE and DBLOODGRP from Donor Table.

```
Autocommit Display 10
select DNAME as Donor, DAGE, DBLOODGRP from Donor;
```

**Results** **Explain** **Describe** **Saved SQL** **History**

DONOR	DAGE	DBLOODGRP
Rafiquil	40	A+
Shamima	33	AB+
Kamrul	38	B+
Tariqul	35	O+
Nasima	32	B-
Alamin	40	A-
Sumaiya	36	AB-
Habib	37	B+
Mizan	32	O+
Farzana	35	B-

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

**Query 2 :** Find all Female PID, PNAME and PBLOODGRP from Patient Table.

```
Autocommit Display 10
select PID as PATIENT_ID, PNAME as Patient, PBLOODGRP as BLOOD_GROUP from patient
where pgender='Female';
```

**Save** **Run**

**Results** **Explain** **Describe** **Saved SQL** **History**

PATIENT_ID	PATIENT	BLOOD_GROUP
25-864	Sumaiya	O-
67-931	Fahim	B-
43-129	Ayesha	A-
31-002	Taslima	A+
33-004	Maliha	B-
34-005	Imran	A-

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

## Aggregate Query:

Query 1 : Get DAGE of the (MIN) Blood\_donors from each DBLOODGRP.

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
Select min(dage) as Blood_Donor from donor group by DBLOODGRP
```

Results Explain Describe Saved SQL History

BLOOD_DONOR
33
32
37
32
36
40
40

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

Query 2 : Get DAGE of the (MAX) Agreed and Disagreed donors.

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
select max(dage) as Agreed_Donor from donor group by dstatus
```

Results Explain Describe Saved SQL History

AGREED_DONOR
36
40

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

## Single-Row Query:

**Query 1 :** Display the DNAME for donors whose DSTATUS is equal to 'Disagree' of DAGE 38.

The screenshot shows a MySQL query editor interface. At the top, there are buttons for 'Autocommit' (checked), 'Display' (set to 10), 'Save', and 'Run'. The SQL query is:

```
select DNAME as Donor, DSTATUS  
from Donor  
where DNAME=  
      (select DNAME from Donor where DAGE=38);
```

Below the query, the results are displayed in a table:

DONOR	DSTATUS
Kamrul	Agree

Text below the table indicates "1 rows returned in 0.00 seconds" and a "CSV Export" link.

**Query 2 :** Display DONOR DNAME and DAGE whose DAGE is greater than Kamrul.

The screenshot shows a MySQL query editor interface. At the top, there are buttons for 'Autocommit' (checked), 'Display' (set to 10), 'Save', and 'Run'. The SQL query is:

```
select dname as Donor, dage  
from donor  
where dage>  
(select dage from donor where dname='Kamrul');
```

Below the query, the results are displayed in a table:

DONOR	DAGE
Rafiqul	40
Alamin	40

Text below the table indicates "2 rows returned in 0.00 seconds" and a "CSV Export" link.

## Multiple-Row Query:

**Query 1 :** Display the DNAME, DOB of Any DAGE smaller than Rafiqul.

```
 Autocommit Display 10 ▾  
select dname as name, dob from donor where dage < any (select dage from donor where dname='Rafiqul');
```

---

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

NAME	DOB
Nasima	05-MAY-91
Mizan	30-JAN-92
Shamima	25-JUN-90
Tariqul	17-SEP-88
Farzana	07-MAR-89
Sumaiya	09-JUL-87
Habib	22-OCT-86
Kamrul	08-AUG-85

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

**Query 2 :** Show patient id, blood group and hospital id of the patient from hospital '75420' and patient id IN '10-628'.

Home > SQL > [SQL Commands](#)

```
 Autocommit Display 10 ▾  
select PID, PBLOODGRP, HID from P_H_info where HID = '75420' and PID IN (select PID from P_H_info where PID='10-628');
```

---

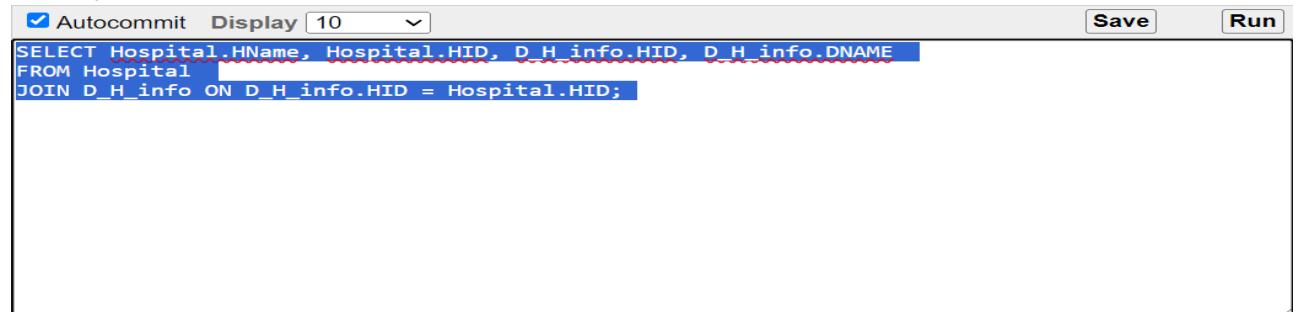
[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

PID	PBLOODGRP	HID
10-628	AB+	75420

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

## Joining Equijoin

**Query 1 :** List all the HNAME where the donor donates and DNAME.



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

- Autocommit is checked.
- Display is set to 10.
- Save and Run buttons are visible.
- The SQL query is:

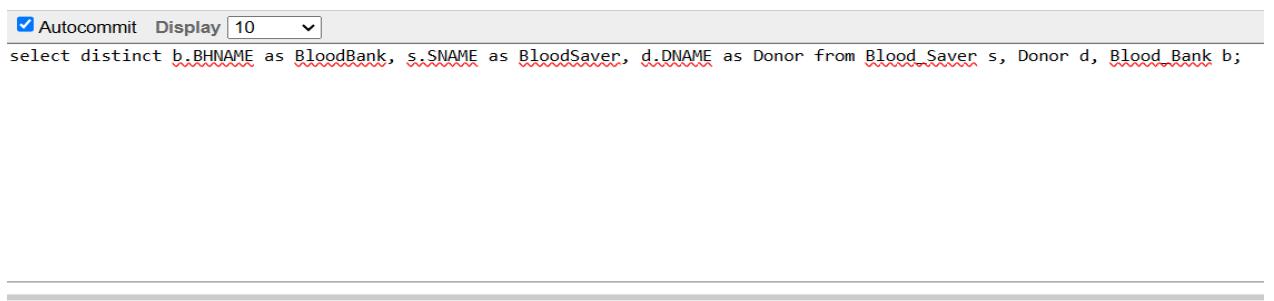
```
SELECT Hospital.HName, Hospital.HID, D_H_info.HID, D_H_info.DNAME
FROM Hospital
JOIN D_H_info ON D_H_info.HID = Hospital.HID;
```
- The results table has columns: HNAME, HID, HID, and DNAME.
- The data in the results table is:

HNAME	HID	HID	DNAME
Dhaka Hospital	75420	75420	Rafiqul
Square Hospitals	10582	10582	Shamima
Apollo Hospitals	69318	69318	Kamrul
United Hospital	23764	23764	Tariqul
Evercare Hospital	80173	80173	Nasima
Bangladesh Hospital	20100	20100	Alamin
Labaid Hospital	20200	20200	Sumaiya
Popular Hospital	20300	20300	Habib
Ibn Sina Hospital	20400	20400	Mizan
Shodesh Hospital	20500	20500	Farzana

- 10 rows returned in 0.00 seconds.
- CSV Export link is present.

## Non-Equijoin

**Query 1 :** Show all the BHNAME and SNAME who collects and supplies blood.



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

- Autocommit is checked.
- Display is set to 10.
- The SQL query is:

```
select distinct b.BHNAME as BloodBank, s.SNAME as BloodSaver, d.DNAME as Donor from Blood_Saver s, Donor d, Blood_Bank b;
```
- The results table has columns: BLOODBANK, BLOODSAVER, and DONOR.
- The data in the results table is:

BLOODBANK	BLOODSAVER	DONOR
Marvel Blood Bank	Rahim	Rafiqul
Marvel Blood Bank	Rahim	Shamima
Sandhani Blood Bank	Rahim	Kamrul
CTG Blood Center	Rahim	Kamrul
Chinmoy Blood Center	Rahim	Tariqul
Sandhani Blood Bank	Rahim	Nasima
Rajshahi Blood Bank	Rahim	Sumaiya
Quantum Blood Bank	Rahim	Habib
CTG Blood Center	Rahim	Habib
Evercare Blood Home	Rahim	Habib

- More than 10 rows available. Increase rows selector to view more rows.
- 10 rows returned in 0.01 seconds.
- CSV Export link is present.

## Outer Join

**Query 1 :** Display DNAME,HNAME , HID, DID from both of the tables.

Home > SQL > **SQL Commands**

Autocommit **Display** 10

```
select d.DNAME as DONOR_NAME, h.HID as HOSPITAL_ID, h.HNAME as HOSPITAL_NAME, d.DID  
as DONOR_ID from hospital h, d_h_info d  
where d.HID (+) = h.HID;
```

**Results Explain Describe Saved SQL History**

DONOR_NAME	HOSPITAL_ID	HOSPITAL_NAME	DONOR_ID
Rafiquil	75420	Dhaka Hospital	21-345
Shamima	10582	Square Hospitals	45-678
Kamrul	69318	Apollo Hospitals	38-902
Tariqul	23764	United Hospital	19-563
Nasima	80173	Evercare Hospital	27-419
Alamin	20100	Bangladesh Hospital	66-210
Sumaiya	20200	Labaid Hospital	34-678
Habib	20300	Popular Hospital	78-112
Mizan	20400	Ibn Sina Hospital	50-003
Farzana	20500	Shodesh Hospital	91-876

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

## Self Join

**Query 1 :** Saver id who collects blood from the agreed donors.

Autocommit **Display** 10

```
select Blood_Saver.SID || ' collects blood from ' || donor.DNAME  
from d_s_info Blood_Saver, d_s_info donor
```

**Results Explain Describe Saved SQL History**

BLOOD_SAVER.SID  'COLLECTSBLOODFROM'  DONOR.DNAME
45-789 collects blood from Rafiquil
45-789 collects blood from Shamima
45-789 collects blood from Kamrul
45-789 collects blood from Tariqul
45-789 collects blood from Nasima
45-789 collects blood from Alamin
45-789 collects blood from Sumaiya
45-789 collects blood from Habib
45-789 collects blood from Mizan
45-789 collects blood from Farzana

More than 10 rows available. Increase rows selector to view more rows.  
10 rows returned in 0.00 seconds [CSV Export](#)

## View

**Query 1 :** Create a view named Under\_40\_Donor\_View that contains DID, DNAME and DAGE younger than 40 years and Display.

```
Create VIEW Under_40_Donor_View AS select DID as ID, DNAME as Name, DAGE from Donor where dage<40;  
SELECT *from Under_40_Donor_View;  
desc Under_40_Donor_View;
```

Results Explain Describe Saved SQL History

Object Type **VIEW** Object **UNDER\_40\_DONOR\_VIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
UNDER_40_DONOR_VIEW	ID	Varchar2	10	-	-	-	-	-	-
	NAME	Varchar2	50	-	-	-	✓	-	-
	DAGE	Number	-	3	0	-	✓	-	-

1 - 3

Results Explain Describe Saved SQL History

ID	NAME	DAGE
45-678	Shamima	33
38-902	Kamrul	38
19-563	Tariqul	35
27-419	Nasima	32
34-678	Sumaiya	36
78-112	Habib	37
50-003	Mizan	32
91-876	Farzana	35

8 rows returned in 0.00 seconds

[CSV Export](#)

## Complex view with two tables

**Query 1 :** Create a view named Bloodbank\_Distribution\_View based on BHNAME and HNAME from both of the table Blood\_Bank and Hospital.

```
Create view Blood_bank_Distribution_View AS
select distinct b.BHNAME as Blood_Bank, h.HNAME as Hospital
from Blood_Bank b, hospital h;

select *from Blood_bank_Distribution_View;
desc Blood_bank_Distribution_View;
```

Results Explain Describe Saved SQL History

Object Type VIEW Object BLOOD\_BANK\_DISTRIBUTION\_VIEW

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BLOOD_BANK_DISTRIBUTION_VIEW	BLOOD_BANK	Varchar2	20	-	-	-	✓	-	-
	HOSPITAL	Varchar2	20	-	-	-	✓	-	-

1 - 2

Autocommit Display 10 ▾

```
Create view Blood_bank_Distribution_View AS
select distinct b.BHNAME as Blood_Bank, h.HNAME as Hospital
from Blood_Bank b, hospital h;

select *from Blood_bank_Distribution_View;
```

Results Explain Describe Saved SQL History

BLOOD_BANK	HOSPITAL
Sandhani Blood Bank	Dhaka Hospital
Sandhani Blood Bank	Square Hospitals
Quantum Blood Bank	Evercare Hospital
Quantum Blood Bank	Ibn Sina Hospital
Chinmoy Blood Center	Shodesh Hospital
Marvel Blood Bank	Evercare Hospital
Marvel Blood Bank	Popular Hospital
Rajshahi Blood Bank	Shodesh Hospital
Sylhet Blood Care	Ibn Sina Hospital
Evercare Blood Home	Labaid Hospital

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

## **DATABASE CONNECTION**

### **1) MD. JANNATUL FAHAD**

Software Installation and JAR File Setup:

1. Downloaded xampp and installed
2. Downloaded jdbc driver
3. Added mysql connector jar file under lib folder in my project

### **Database and Table Setup**

I Launched XAMPP and start both the Apache and MySQL modules from the XAMPP Control Panel. I

Open phpMyAdmin by clicking on the "Admin" button to the MySQL service in XAMPP and accessed

"<http://localhost:8080/phpmyadmin>" to create a database named "Hospital.". Inside this database, I created a table called " Hospital " and added columns according to the project requirements.

Then I inserted some data into the table.

HID	HNAME	HNO
10582	Square Hospital	1713147852
20100	Bangladesh Hospital	1711996633
20200	Labaid Hospital	1713333777
20300	Popular Hospital	1711808080
20400	Ibn Sina Hospital	1715108008
20500	Shodesh Hospital	1678901234
23764	United Hospitals	1911888000
69318	Apollo Hospitals	1713147852
75420	Dhaka Hospital	1711123456
80173	Evercare Hospitals	1815000111

Now we need an IDE.

Before that, we need to understand the steps of DB connection

- ➔ Register Driver
- ➔ Connect the DB
- ➔ Create statement
- ➔ Execute the query () in the resultstmtnt
- ➔ Connection close ()

# Database Connection Code

C:\Users\Mohammed Fahad\Downloads\My Sql Driver\Object.java - Notepad++

The screenshot shows a Notepad++ window with the title "Object.java". The code is a Java program that connects to a MySQL database named "Hospital" using the JDBC driver. It prints the details of all hospitals in the database to the console. The code includes imports, a main method, try-catch blocks for exceptions, and comments explaining the logic.

```
1 import java.sql.*;
2
3 public class Object {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver");
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hospital", "root", "");
9             System.out.println("connected");
10            Statement st = conn.createStatement();
11            ResultSet rs = st.executeQuery("select * from Hospital");//Executing the SQL Query
12            while (rs.next()) {
13                System.out.println("HID=" + rs.getString(1));
14                System.out.println("HNAME=" + rs.getString(2));
15                System.out.println("HNO=" + rs.getInt(3));
16
17            }
18        } catch (Exception s) {
19            System.out.println(s);
20        }
21    }
22
23 }
24
25
26
27 //javac -cp Driver.jar;. Object.java
28 //java -cp Driver.jar;. Object
29
30
31 /* System.out.println("SID=" + rs.getInt(1) + " First Name=" + rs.getString(2) + " Last Name=" + rs.getString(3) + " Department=" + rs.getString(4));// Processing the ResultSet*/
```

## Output

Microsoft Windows [Version 10.0.26100.2894]  
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mohammed Fahad\Downloads\My Sql Driver>javac -cp Driver.jar;. Object.java

C:\Users\Mohammed Fahad\Downloads\My Sql Driver>java -cp Driver.jar;. Object

connected  
HID=10582  
HNAME=Square Hospital  
HNO=1713147852  
HID=20100  
HNAME=Bangladesh Hospital  
HNO=1711996633  
HID=20200  
HNAME=Labaid Hospital  
HNO=1713333777  
HID=20300  
HNAME=Popular Hospital  
HNO=1711808888  
HID=20400  
HNAME=Ibn Sina Hospital  
HNO=1715108098  
HID=20500  
HNAME=Shodesh Hospital  
HNO=1678901234  
HID=23764  
HNAME=United Hospitals  
HNO=1911888900  
HID=69318  
HNAME=Apollo Hospitals  
HNO=1713147852  
HID=75420  
HNAME=Dhaka Hospital  
HNO=1711123456  
HID=80173  
HNAME=Evercare Hospitals  
HNO=18150900111

C:\Users\Mohammed Fahad\Downloads\My Sql Driver>

The screenshot shows a Windows Command Prompt window titled "cmd.exe" with the path "C:\Windows\System32\cmd.exe". The user has navigated to their Downloads folder and run the Java command to execute the "Object.java" program. The output lists the details of various hospitals from the "Hospital" database, including their HID, HNAME, and HNO.

## 2) SAYED ASHFAKUL ISLAM

Software Installation and JAR File Setup:

1. Downloaded xampp and installed
2. Downloaded Beekeeper Studio.
2. Downloaded jdbc driver
4. Added mysql connector jar file under lib folder in my project

### Database and Table Setup

I Launched XAMPP and start only the MySQL modules from the XAMPP Control Panel. Then I opened **Beekeeper Studio** and made a mysql connection using “root” as username and then accessed localhost:3306-Beekeeper Studio to create a database named "bloodmanagement". Inside this database, I created a table called "blood\_bank" and added columns according to the project requirements. Then I inserted some data into the table.

	DID	varchar(255)	DNAME	varchar(255)	DOB	date	DAGE	varchar(255)	DBLOODGRP	varchar(255)	DSTATUS	varchar(255)
1	19-563		Tariqul		1988-09-17		35		O+		Disagree	
2	21-345		Rafiqul		1984-02-12		40		A+		Agree	
3	27-419		Nasima		1991-05-05		32		B-		Agree	
4	34-678		Sumaiya		1987-07-09		36		AB-		Disagree	
5	38-902		Kamrul		1985-08-08		38		B+		Agree	
6	45-678		Shamima		1990-06-25		33		AB+		Disagree	
7	50-003		Mizan		1992-01-30		32		O+		Disagree	
8	66-210		Alamin		1983-12-14		40		A-		Agree	
9	78-112		Habib		1986-10-22		37		B+		Agree	
10	91-876		Farzana		1989-03-07		35		B-		Agree	

Now we need an IDE.

Before that, we need to understand the steps of DB connection

- ➔ Register Driver
- ➔ Connect the DB
- ➔ Create statement
- ➔ Execute the query () in the resultstmt
- ➔ Connection close ()

# Database Connection Code

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** Code, File, Edit, Selection, View, Go, Run, Terminal, Window, Help.
- Toolbar:** Includes icons for file operations, search, and other common functions.
- Search Bar:** A search bar labeled "Search" is located at the top right.
- Code Area:** The main window displays the content of `Object.java`. The code uses JDBC to connect to a MySQL database named "newdb" running on localhost port 3306. It retrieves all records from a table and prints each row's fields to the console.
- Status Bar:** Shows the line number (Ln 34), column number (Col 37), spaces (Spaces: 4), encoding (UTF-8), and file type (CRLF) for the current file.
- Taskbar:** The taskbar at the bottom shows various application icons.

```
1 import java.sql.*;
2
3 public class Object {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName(className:"com.mysql.cj.jdbc.Driver"); // register jdbc driver of mysql,Driver Registration
8             //This line ensures that the MySQL JDBC driver is loaded and registered with the JDBC API. It allows your application to connect to the MySQL database.
9             Connection conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/donor", user:"root", password:""); //Establishing a Connection
10            //This line establishes a connection to the database named newdb running on localhost at the default MySQL port 3306.
11            //DriverManager.getConnection() returns a Connection object which represents the session with the database.
12            System.out.println(x:"connected");//printing that database is connected
13            Statement st = conn.createStatement(); //Creating a Statement which allows you to send SQL queries to the database.
14            ResultSet rs = st.executeQuery(sql:"select * from donor");//Executing the SQL Query
15            while (rs.next()) { //moves the cursor to the next row of the result set. If there are no more rows, it returns false, terminating the loop
16                System.out.println("DID=" + rs.getString(columnIndex:1));
17                System.out.println("DNAME=" + rs.getString(columnIndex:2));
18                System.out.println("DOB=" + rs.getString(columnIndex:3));
19                System.out.println("DAGE=" + rs.getInt(columnIndex:4));
20                System.out.println("DBLOODGRP=" + rs.getString(columnIndex:5));
21                System.out.println("DSTATUS=" + rs.getString(columnIndex:6));
22
23            }
24
25            // Connection.close(); //This line is commented out, which means the connection is not being closed after use. This could lead to resource leaks.
26            // Ideally, you should close the connection after completing your operations. You can do this either by explicitly calling conn.close() or by using try-with-resources.
27        } catch (Exception s) {
28            System.out.println(s);
29        }
30    }
31 }
```

## Output

The screenshot shows a Windows Command Prompt window with the following details:

- Title Bar:** UTM, File, Edit, View, Window, Help.
- Taskbar:** Shows the taskbar with various application icons.
- Code Area:** The command prompt shows the execution of the Java code and its output. The output lists multiple rows of data, each representing a record from the database table, with columns DNAME, DOB, DAGE, DBLOODGRP, and DSTATUS.

```
C:\Users\Ashfaque\Downloads\DBCODE\DBCODE>java -cp Driver.jar;. Object
connected
DID=19-563
DNAME=Tariqul
DOB=1988-09-17
DAGE=35
DBLOODGRP=0+
DSTATUS=Disagree
DID=21-345
DNAME=Rafiqul
DOB=1984-02-12
DAGE=40
DBLOODGRP=A+
DSTATUS=Agree
DID=27-419
DNAME=Nasima
DOB=1991-05-05
DAGE=32
DBLOODGRP=B-
DSTATUS=Agree
DID=34-678
DNAME=Sumaiya
DOB=1987-07-09
DAGE=36
DBLOODGRP=AB-
DSTATUS=Disagree
DID=38-902
DNAME=Kamrul
DOB=1985-08-08
DAGE=38
DBLOODGRP=B+
DSTATUS=Agree
DID=45-678
DNAME=Shamima
DOB=1990-06-25
```

### **3) A.R.M. ZEHAD**

Software Installation and JAR File Setup:

1. Downloaded xampp and installed
2. Downloaded jdbc driver
3. Added mysql connector jar file under lib folder in my project

### **Database and Table Setup**

I Launched XAMPP and start both the Apache and MySQL modules from the XAMPP Control Panel. I

Open phpMyAdmin by clicking on the "Admin" button to the MySQL service in XAMPP and accessed

"<http://localhost:8080/phpmyadmin>" to create a database named "Hospital.". Inside this database, I created a table called " patient " and added columns according to the project requirements.

Then I inserted some data into the table.

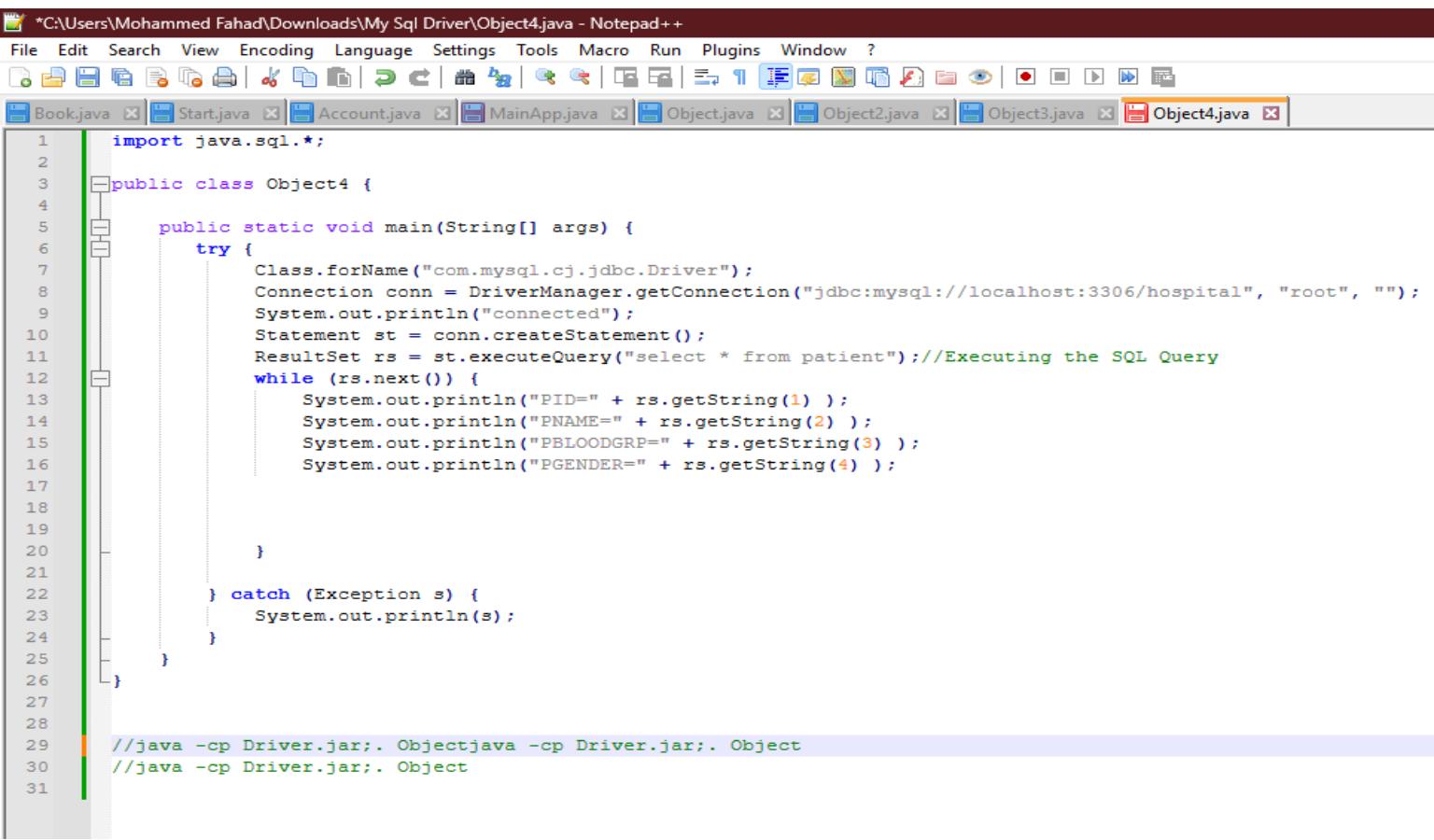
PID	PNAME	PBLOODGRP	PGENDER
10-628	Anika	AB+	Male
25-864	Sumaiya	O-	Female
30-001	Nafis	AB+	Male
31-002	Taslima	A+	Female
32-003	Arif	O-	Male
33-004	Maliha	B-	Female
34-005	Imran	A-	Female
43-129	Ayesha	A-	Female
67-931	Fahim	B-	Female
79-538	Rahim	A+	Male

Now we need an IDE.

Before that, we need to understand the steps of DB connection

- ➔ Register Driver
- ➔ Connect the DB
- ➔ Create statement
- ➔ Execute the query () in the resultstmt
- ➔ Connection close ()

## Database Connection Code

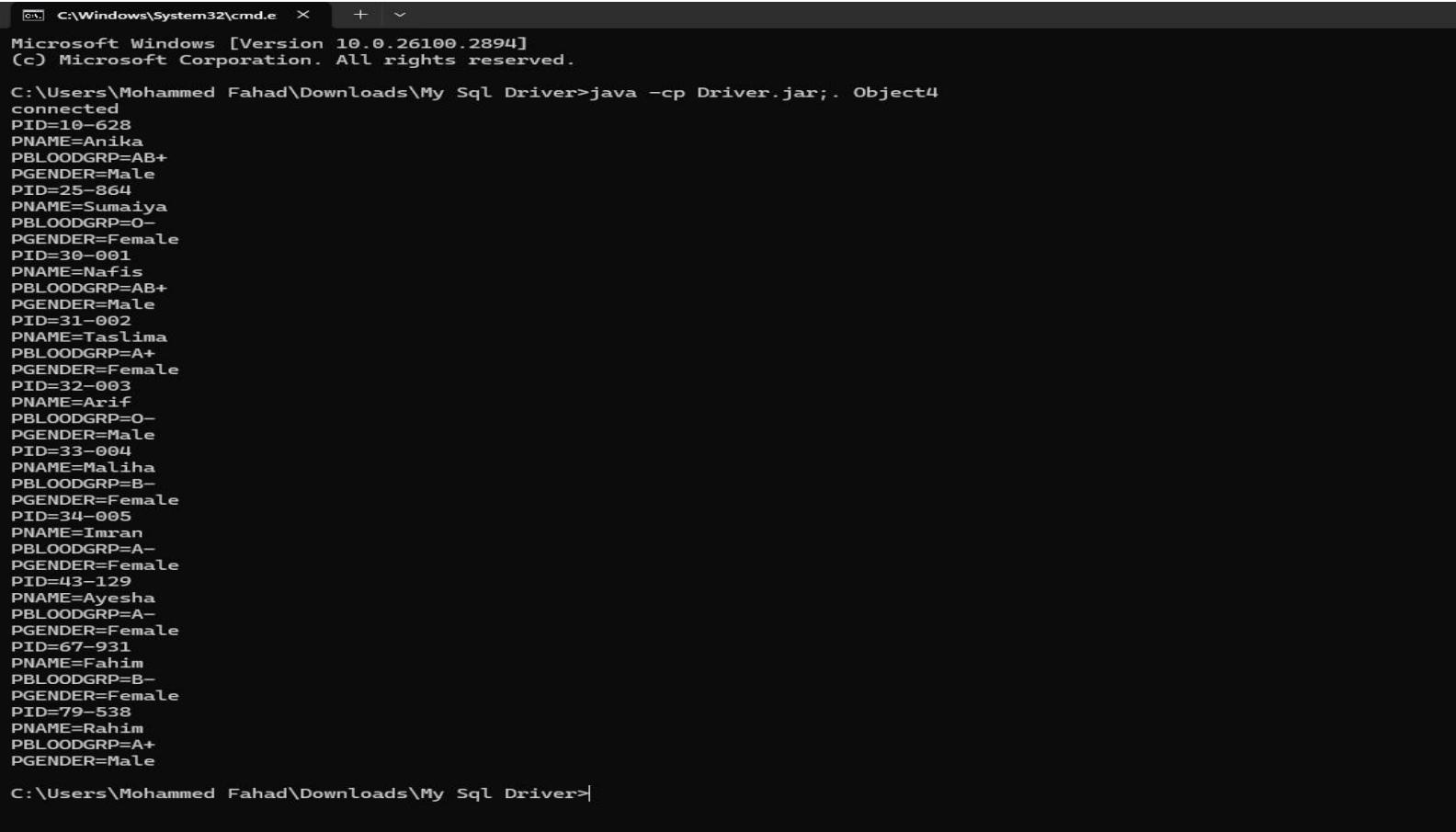


The screenshot shows a Notepad++ window with the title bar "C:\Users\Mohammed Fahad\Downloads\My Sql Driver\Object4.java - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?.

The code in the editor is as follows:

```
1 import java.sql.*;
2
3 public class Object4 {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver");
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital", "root", "");
9             System.out.println("connected");
10            Statement st = conn.createStatement();
11            ResultSet rs = st.executeQuery("select * from patient");//Executing the SQL Query
12            while (rs.next()) {
13                System.out.println("PID=" + rs.getString(1));
14                System.out.println("PNAME=" + rs.getString(2));
15                System.out.println("PBLOODGRP=" + rs.getString(3));
16                System.out.println("PGENDER=" + rs.getString(4));
17
18            }
19
20        } catch (Exception s) {
21            System.out.println(s);
22        }
23    }
24
25 }
26
27
28
29 //java -cp Driver.jar;. Objectjava -cp Driver.jar;. Object
30 //java -cp Driver.jar;. Object
31
```

## OUTPUT



The screenshot shows a Windows Command Prompt window titled "cmd C:\Windows\System32\cmd.e". The command entered is "C:\Users\Mohammed Fahad\Downloads\My Sql Driver>java -cp Driver.jar;. Object4". The output displayed is the result of executing the Java code, which prints the details of patients from the "patient" table in the "hospital" database.

```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mohammed Fahad\Downloads\My Sql Driver>java -cp Driver.jar;. Object4
connected
PID=10-628
PNAME=Anika
PBLOODGRP=AB+
PGENDER=Male
PID=25-864
PNAME=Sumaiya
PBLOODGRP=O-
PGENDER=Female
PID=30-001
PNAME=Nafis
PBLOODGRP=AB+
PGENDER=Male
PID=31-002
PNAME=Taslima
PBLOODGRP=A+
PGENDER=Female
PID=32-003
PNAME=Arif
PBLOODGRP=O-
PGENDER=Male
PID=33-004
PNAME=Maliha
PBLOODGRP=B-
PGENDER=Female
PID=34-005
PNAME=Imran
PBLOODGRP=A-
PGENDER=Female
PID=43-129
PNAME=Ayesha
PBLOODGRP=A-
PGENDER=Female
PID=67-931
PNAME=Fahim
PBLOODGRP=B-
PGENDER=Female
PID=79-538
PNAME=Rahim
PBLOODGRP=A+
PGENDER=Male
```

## 4) MD.SHIHAB SHIKDAR

Software Installation and JAR File Setup:

1. Downloaded xampp and installed
2. Downloaded jdbc driver
3. Added mysql connector jar file under lib folder in my project

### **Database and Table Setup**

I Launched XAMPP and start both the Apache and MySQL modules from the XAMPP Control Panel. I

Open phpMyAdmin by clicking on the "Admin" button to the MySQL service in XAMPP and accessed

"<http://localhost:8080/phpmyadmin>" to create a database named "Hospital.". Inside this database, I created a table called " blood\_saver " and added columns according to the project requirements.

Then I inserted some data into the table.

SID	SNAME	SNO
11-223	Salma	1814567891
25-678	Munia	1896543211
29-876	Jamal	1799887766
32-654	Karim	1826547892
45-789	Rahim	1715678901
53-902	Amin	1925467832
66-304	Farida	1387654321
77-410	Nasir	1676543210
87-321	Hasan	1934567890
90-123	Nayem	1726547899

Now we need an IDE.

Before that, we need to understand the steps of DB connection

- ➔ Register Driver
- ➔ Connect the DB
- ➔ Create statement
- ➔ Execute the query () in the resultstmtnt
- ➔ Connection close ()

## Database Connection Code

The screenshot shows a Notepad++ window with the title bar "C:\Users\Mohammed Fahad\Downloads\My Sql Driver\Object3.java - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. Below the menu is a toolbar with various icons. The main editor area contains Java code for connecting to a MySQL database and executing a query:

```
1 import java.sql.*;
2
3 public class Object3 {
4
5     public static void main(String[] args) {
6         try {
7             Class.forName("com.mysql.cj.jdbc.Driver");
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital", "root", "");
9             System.out.println("connected");
10            Statement st = conn.createStatement();
11            ResultSet rs = st.executeQuery("select * from blood_saver");//Executing the SQL Query
12            while (rs.next()) {
13                System.out.println("SID=" + rs.getString(1));
14                System.out.println("SNAME=" + rs.getString(2));
15                System.out.println("SNO=" + rs.getString(3));
16
17            }
18
19        } catch (Exception s) {
20            System.out.println(s);
21        }
22    }
23
24
25
26
27
28 //javac -cp Driver.jar;. Object.java
29 //java -cp Driver.jar;. Object
30 }
```

## OUTPUT

The screenshot shows a Windows Command Prompt window titled "C:\Windows\System32\cmd.e" with the command prompt ">". The output of the Java application is displayed, showing multiple rows of data extracted from the "blood\_saver" table:

```
SNO=1814567891
SID=25-678
SNAME=Munia
SNO=1896543211
SID=29-876
SNAME=Jamal
SNO=1799887766
SID=32-654
SNAME=Karim
SNO=1826547892
SID=45-789
SNAME=Rahim
SNO=1715678901
SID=53-902
SNAME=Amin
SNO=1925467832
SID=66-304
SNAME=Farida
SNO=1387654321
SID=77-410
SNAME=Nasir
SNO=1676543210
SID=87-321
SNAME=Hasan
SNO=1934567890
SID=90-123
SNAME=Nayem
SNO=1726547899
```

The command prompt at the bottom shows the path "C:\Users\Mohammed Fahad\Downloads\My Sql Driver>".

## Conclusion

The project named Blood Donation Management System was highly needed as an proactive approach to deal with emergencies for the availability of blood in time, just like the Chowk Bazar Dhaka Fire Tragedy. It efficiently coordinates donors, blood banks, hospitals, and patients to help in easy blood collection, storage, and distribution. Application of database normalization and optimized queries enhances performance and reliability. The project will not only address critical gaps in the emergency supply of blood but will also set up a sustainable model for future crisis responses. Above all, this is a testimony to the power of technology in saving lives and bringing better healthcare accessibility.