

**CSC 785- Information Retrieval**  
**Project 3 – Blood Donation System**  
**Submitted By - Srijana Raut(101134199)**

**Question - Design a database: blood donation system and satisfy following:**

- A system in which data of Patient, data of donor, data of blood bank would be saved and will be interrelated with each other.
- DATA OF PATIENT: Patient Name, Patient Id, Patient Blood Group, Patient Disease
- DATA OF DONOR: Donor Name, Donor Id, Donor Blood Group, Donor Medical report, Donor Address, Donor Contact number
- DATA OF BLOOD BANK: Blood Bank Name, Blood Bank Address, Blood bank Donors name, Blood Bank Contact Number, Blood Bank Address

According to the Project Objective, following are the required screenshots of solution.

**1. Describe the properties of all relations.**

In this project, we created three tables. They are;

- **Patient Table**- The Patient table stores information about patients. It has the following attributes:
  - i. PatientId: A unique identifier for each patient.
  - ii. PatientName: The name of the patient.
  - iii. BloodGroup: The blood group of the patient.
  - iv. PatientDisease: Any disease or medical condition the patient may have.

```
mysql> CREATE TABLE Patient ( PatientId INT PRIMARY KEY, PatientName VARCHAR(255), BloodGroup VARCHAR(5), PatientDisease VARCHAR(255) );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> INSERT INTO Patient (PatientId, PatientName, BloodGroup, PatientDisease) VALUES
(1, 'John Doe', 'A+', 'None'), (2, 'Jane Smith', 'B-', 'Hypertension'), (3, 'Michael Johnson', 'O+', 'Diabetes'), (4, 'Emily Davis', 'AB+', 'None'), (5, 'KC Santosh', 'O-', 'Asthma'), (6, 'Srijana Raut', 'O-', 'Diabetes');
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
[mysql> select * from patient;
```

PatientId	PatientName	BloodGroup	PatientDisease
1	John Doe	A+	None
2	Jane Smith	B-	Hypertension
3	Michael Johnson	O+	Diabetes
4	Emily Davis	AB+	None
5	KC Santosh	O-	Asthma
6	Srijana Raut	O-	Diabetes

```
6 rows in set (0.00 sec)
```

**Note:** I modify the Table Patient by adding one column with DonorID as foreign Key to show the relation between the two tables which is shown below.

**Now the Final Table Looks like.**

```
[mysql> select * from Patient;
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
1	John Doe	A+	None	1
2	Jane Smith	B-	Hypertension	2
3	Michael Johnson	O+	Diabetes	3
4	Emily Davis	AB+	None	4
5	KC Santosh	O-	Asthma	5
6	Srijana Raut	O-	Diabetes	5

```
6 rows in set (0.00 sec)
```

- **Donor Table:** The Donor table stores information about individuals who are potential blood donors. It has the following attributes:
  - i. DonorId: A unique identifier for each donor.
  - ii. DonorName: The name of the donor.
  - iii. BloodGroup: The blood group of the donor.
  - iv. MedicalReport: This field is assumed to store the medical report of the donor, but it's set as NULL for now.
  - v. Address: The address of the donor.
  - vi. ContactNumber: The contact number of the donor.

```
mysql> INSERT INTO Donor (DonorId, DonorName, BloodGroup, MedicalReport, Address, ContactNumber)
-> VALUES
-> (1, 'Sarah Brown', 'A+', NULL, '123 Main St', '555-123-4567'),
-> (2, 'Robert Lee', 'B-', NULL, '456 Elm St', '555-987-6543'),
-> (3, 'Susan White', 'O+', NULL, '789 Oak St', '555-321-7890'),
-> (4, 'Sristi Raut', 'AB+', NULL, '101 Pine St', '555-555-5555'),
-> (5, 'Shreya Poudel', 'O-', NULL, '422 N Dakota', '984-123-8877');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Donor;
```

DonorId	DonorName	BloodGroup	MedicalReport	Address	ContactNumber
1	Sarah Brown	A+	NULL	123 Main St	555-123-4567
2	Robert Lee	B-	NULL	456 Elm St	555-987-6543
3	Susan White	O+	NULL	789 Oak St	555-321-7890
4	Sristi Raut	AB+	NULL	101 Pine St	555-555-5555
5	Shreya Poudel	O-	NULL	422 N Dakota	984-123-8877

```
5 rows in set (0.00 sec)
```

- **BloodBank Table:** The BloodBank table stores information about blood banks or centers that collect and distribute blood. It has the following attributes:
  - i. BloodBankName: A unique identifier for each blood bank.
  - ii. Address: The address of the blood bank.
  - iii. DonorName: This field is intended to reference the name of the donor associated with the blood bank.

Here, while creating the table BloodBank, I got an error related to the foreign key constraint on the DonorName column. To modify an existing table in MySQL to add a UNIQUE constraint to a column, I have used the ALTER TABLE statement. In this case, I want to add a UNIQUE constraint to the DonorName column in the Donor table.

```
mysql> ALTER TABLE Donor
-> ADD CONSTRAINT UC_DonorName UNIQUE (DonorName);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE BloodBank (DonorBloodBankName VARCHAR(255), Address VARCHAR(255), DonorName VARCHAR(255),
ContactNumber VARCHAR(15), FOREIGN
KEY (DonorName) REFERENCES Donor(DonorName) );
Query OK, 0 rows affected (0.02 sec)
```

```
[mysql> select * from BloodBank;
```

BloodBankName	Address	DonorName	ContactNumber
City Blood Bank	123 Oak Ave	Sristi Raut	555-111-2222
Community Blood Center	456 Elm St	Sarah Brown	555-222-3333
County Blood Services	202 Birch St	Shreya Poudel	555-555-6666
National Blood Center	101 Maple St	Susan White	555-444-5555
Regional Blood Bank	789 Pine St	Robert Lee	555-333-4444

```
5 rows in set (0.01 sec)
```

## 2. Select the Specific rows and the columns.

It select only the two columns from Patient Table

```
[mysql> select PatientName, BloodGroup from Patient;
```

PatientName	BloodGroup
John Doe	A+
Jane Smith	B-
Michael Johnson	O+
KC Santosh	O-
Srijana Raut	O-

```
5 rows in set (0.00 sec)
```

The below query select the tuples from the Patient table who is o negative or Disease with diabetes.

```
[mysql> SELECT * from Patient WHERE BloodGroup = 'O-' OR PatientDisease='Diabetes';
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
3	Michael Johnson	O+	Diabetes	3
5	KC Santosh	O-	Asthma	5
6	Srijana Raut	O-	Diabetes	5

```
3 rows in set (0.01 sec)
```

## 3. Apply Search conditions with calculated Fields.

Example – To extract all the distinct Patients Disease from Patient table in first command and extracting all the data of patients having diabetes in the second command.

```
[mysql> select distinct PatientDisease from Patient;
+-----+
| PatientDisease |
+-----+
| None           |
| Hypertension   |
| Diabetes       |
| Asthma         |
+-----+
4 rows in set (0.00 sec)
```

```
[mysql> select * from Patient where PatientDisease='Diabetes';
+-----+-----+-----+-----+-----+
| PatientId | PatientName | BloodGroup | PatientDisease | DonorId |
+-----+-----+-----+-----+-----+
|          3 | Michael Johnson | O+         | Diabetes       |        3 |
|          6 | Srijana Raut   | O-         | Diabetes       |        5 |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

#### 4. Use a Pattern search.

To show the example of pattern search. I use the LIKE operator. The first query selects all the columns of patient whose name is something like KC.

The second command selects all columns for donors whose contact numbers start with '555-'.

```
[mysql> SELECT * FROM Patient WHERE PatientName LIKE '%KC%';
+-----+-----+-----+-----+-----+
| PatientId | PatientName | BloodGroup | PatientDisease | DonorId |
+-----+-----+-----+-----+-----+
|          5 | KC Santosh  | O-         | Asthma         |        5 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT *
-> FROM Donor
-> WHERE ContactNumber LIKE '555-';
+-----+-----+-----+-----+-----+-----+
| DonorId | DonorName | BloodGroup | MedicalReport | Address | ContactNumber |
+-----+-----+-----+-----+-----+-----+
|        1 | Sarah Brown | A+         | NULL          | 123 Main St | 555-123-4567 |
|        2 | Robert Lee  | B-         | NULL          | 456 Elm St  | 555-987-6543 |
|        3 | Susan White | O+         | NULL          | 789 Oak St  | 555-321-7890 |
|        4 | Sristi Raut | AB+        | NULL          | 101 Pine St | 555-555-5555 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

#### 5. Select tuples based on ordering (but multiple columns);

To show the ordering, The first command shows the name of patient from patient table in ascending order/

The second command shows the DonorName from Donor table in descending order.

```
[mysql> select * from Patient ORDER BY PatientName ASC;
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
4	Emily Davis	AB+	None	4
2	Jane Smith	B-	Hypertension	2
1	John Doe	A+	None	1
5	KC Santosh	O-	Asthma	5
3	Michael Johnson	O+	Diabetes	3
6	Srijana Raut	O-	Diabetes	5

```
6 rows in set (0.00 sec)
```

```
[mysql> select * from BloodBank ORDER BY DonorName DESC;
```

BloodBankName	Address	DonorName	ContactNumber
National Blood Center	101 Maple St	Susan White	555-444-5555
City Blood Bank	123 Oak Ave	Sristi Raut	555-111-2222
County Blood Services	202 Birch St	Shreya Poudel	555-555-6666
Community Blood Center	456 Elm St	Sarah Brown	555-222-3333
Regional Blood Bank	789 Pine St	Robert Lee	555-333-4444

```
5 rows in set (0.00 sec)
```

## 6. Use nested queries.

The below query finds all patients who have the same blood group as a specific donor named "Sristi Raut."

```
mysql> SELECT *
-> FROM Patient
-> WHERE BloodGroup = (
->     SELECT BloodGroup
->     FROM Donor
->     WHERE DonorName = 'Sristi Raut');
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
4	Emily Davis	AB+	None	4

```
1 row in set (0.00 sec)
```

## 7. Use aggregated function.

We can use the various aggregated function like SUM, MIN, MAX. For the table donor, I am using AVG to calculates the total number of donors and the average contact number length:

```
mysql> SELECT
->     COUNT(*) AS TotalDonors,
->     AVG(LENGTH(ContactNumber)) AS AverageContactNumberLength
-> FROM Donor;
+-----+-----+
| TotalDonors | AverageContactNumberLength |
+-----+-----+
|          5 |                12.0000 |
+-----+-----+
1 row in set (0.05 sec)
```

## 8. Take multiple relations in a query.

Simple Query - The query will return the names of patients with the blood group 'O-' from the Patient table.

```
mysql> SELECT
->     PatientName
-> FROM
->     Patient
-> WHERE
->     BloodGroup = 'O-';
+-----+
| PatientName |
+-----+
| KC Santosh  |
| Srijana Raut |
+-----+
2 rows in set (0.00 sec)
```

**Using Join Operator** - Retrieving the names of patients who have received blood from donors with a blood group of 'O-' along with the blood group of each patient. Including the name of the blood bank organization associated with each donor.



```
mysql> SELECT
->     p.PatientName AS PatientName,
->     p.BloodGroup AS PatientBloodGroup,
->     d.DonorName AS DonorName,
->     bb.BloodBankName AS BloodBankName
-> FROM
->     Patient p
-> JOIN
->     Donor d ON p.DonorId = d.DonorId
-> JOIN
->     BloodBank bb ON d.DonorName = bb.DonorName
-> WHERE
->     d.BloodGroup = 'O-';
```

PatientName	PatientBloodGroup	DonorName	BloodBankName
KC Santosh	O-	Shreya Poudel	County Blood Services
Srijana Raut	O-	Shreya Poudel	County Blood Services

```
2 rows in set (0.01 sec)
```

## 9. Update specific columns and/or fields.

Updating the Patient table. The column name already created. Thus, are updated.

```
mysql> UPDATE Patient p
-> JOIN Donor d ON p.BloodGroup = d.BloodGroup
-> SET p.DonorId = d.DonorId;
Query OK, 6 rows affected (0.01 sec)
Rows matched: 6 Changed: 6 Warnings: 0
```

After Updating the table looks like.

```
mysql> select * from Patient;
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
1	John Doe	A+	None	1
2	Jane Smith	B-	Hypertension	2
3	Michael Johnson	O+	Diabetes	3
4	Emily Davis	AB+	None	4
5	KC Santosh	O-	Asthma	5
6	Srijana Raut	O-	Diabetes	5

```
6 rows in set (0.00 sec)
```

## 10. Drop specific columns and rows.

**Row Drop** – To show row drop, we will use Patient table.

Before dropping, the Patient table looks like.



```
mysql> select * from Patient;
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
1	John Doe	A+	None	1
2	Jane Smith	B-	Hypertension	2
3	Michael Johnson	O+	Diabetes	3
4	Emily Davis	AB+	None	4
5	KC Santosh	O-	Asthma	5
6	Srijana Raut	O-	Diabetes	5

```
6 rows in set (0.00 sec)
```

Dropping specific rows of the information whose Blood group is AB+ and the resulting table.

```
mysql> DELETE FROM Patient WHERE BloodGroup = 'AB+';
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from Patient;
```

PatientId	PatientName	BloodGroup	PatientDisease	DonorId
1	John Doe	A+	None	1
2	Jane Smith	B-	Hypertension	2
3	Michael Johnson	O+	Diabetes	3
5	KC Santosh	O-	Asthma	5
6	Srijana Raut	O-	Diabetes	5

```
5 rows in set (0.01 sec)
```

**Column Drop** - In the Table Donor shows the operation column drop. At first the Donor table attributes are.

```
mysql> select * from Donor;
```

DonorId	DonorName	BloodGroup	MedicalReport	Address	ContactNumber
1	Sarah Brown	A+	NULL	123 Main St	555-123-4567
2	Robert Lee	B-	NULL	456 Elm St	555-987-6543
3	Susan White	O+	NULL	789 Oak St	555-321-7890
4	Sristi Raut	AB+	NULL	101 Pine St	555-555-5555
5	Shreya Poudel	O-	NULL	422 N Dakota	984-123-8877

```
5 rows in set (0.00 sec)
```

Drop the column called MedicalReport

```
mysql> ALTER TABLE Donor
-> DROP COLUMN MedicalReport;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Now, resulting table after performing the drop query.

```
mysql> select * From Donor;
```

DonorId	DonorName	BloodGroup	Address	ContactNumber
1	Sarah Brown	A+	123 Main St	555-123-4567
2	Robert Lee	B-	456 Elm St	555-987-6543
3	Susan White	O+	789 Oak St	555-321-7890
4	Sristi Raut	AB+	101 Pine St	555-555-5555
5	Shreya Poudel	O-	422 N Dakota	984-123-8877

```
5 rows in set (0.00 sec)
```

## 11. Create users and provide different views.

Creating users called srijana\_user1 and srijana\_user2 and differentiating the views with/without access to database which is shown in given example.

```
mysql> CREATE USER 'srijana_user1'@'localhost' IDENTIFIED BY 'password1';
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> SHOW GRANTS FOR 'srijana_user1'@'localhost';
```

Grants for srijana_user1@localhost
GRANT USAGE ON *.* TO `srijana_user1`@`localhost`
GRANT SELECT ON `project3_blood_donation_system`.`patientview` TO `srijana_user1`@`localhost`

```
2 rows in set (0.00 sec)
```

```
(base) srijanaraut@Srijanas-MBP ~ % mysql -u srijana_user1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| PROJECT3_BLOOD_DONATION_SYSTEM |
+-----+
3 rows in set (0.01 sec)

mysql> use PROJECT3_BLOOD_DONATION_SYSTEM;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT COUNT(*)
  -> FROM information_schema.tables
  -> WHERE table_schema = 'PROJECT3_BLOOD_DONATION_SYSTEM';
+-----+
| COUNT(*) |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM PatientView;
+-----+-----+-----+
| PatientId | PatientName | BloodGroup |
+-----+-----+-----+
|          1 | John Doe | A+ |
|          2 | Jane Smith | B- |
|          3 | Michael Johnson | O+ |
|          5 | KC Santosh | O- |
|          6 | Srijana Raut | O- |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

Second User – Srijana\_user2

```
mysql> CREATE USER 'srijana_user2'@'localhost' IDENTIFIED BY 'password2';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SHOW GRANTS FOR 'srijana_user2'@'localhost';
+-----+
| Grants for srijana_user2@localhost |
+-----+
| GRANT USAGE ON *.* TO `srijana_user2`@`localhost` |
+-----+
1 row in set (0.00 sec)
```

```
Last login: Mon Oct  2 09:22:04 on ttys001
(base) srijanaraut@Srijanas-MBP ~ % mysql -u srijana_user2 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
+-----+
2 rows in set (0.00 sec)

mysql> █
```

## 12.grant privileges (global and local) for specific users.

Here, we provide the global privileges as Select, insert to Srijana\_user1.

```

[mysql> GRANT SELECT, INSERT ON PROJECT3_BLOOD_DONATION_SYSTEM.* TO 'srijana_user1'@'localhost';
Query OK, 0 rows affected (0.00 sec)

[mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

[mysql> SHOW GRANTS FOR 'srijana_user1'@'localhost';
+-----+
| Grants for srijana_user1@localhost |
+-----+
| GRANT USAGE ON *.* TO `srijana_user1`@`localhost` |
| GRANT SELECT, INSERT ON `project3_blood_donation_system`.* TO `srijana_user1`@`localhost` |
| GRANT SELECT ON `project3_blood_donation_system`.`patientview` TO `srijana_user1`@`localhost` |
+-----+
3 rows in set (0.00 sec)

mysql> █

```

Here, we provide the local privileges as update, delete to Srijana\_user2.

```

[mysql> GRANT UPDATE,DELETE ON PROJECT3_BLOOD_DONATION_SYSTEM.* TO 'srijana_user2'@'localhost';
Query OK, 0 rows affected (0.01 sec)

[mysql> FLUSH PRIVILEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL s
erver version for the right syntax to use near 'PRIVILEGES' at line 1
[mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

[mysql> SHOW GRANTS FOR 'srijana_user2'@'localhost';
+-----+
| Grants for srijana_user2@localhost |
+-----+
| GRANT USAGE ON *.* TO `srijana_user2`@`localhost` |
| GRANT UPDATE, DELETE ON `project3_blood_donation_system`.* TO `srijana_user2`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql> █

```

### 13.backup the database (complete database and specific relations from a particular database); and

Backing up the complete database

```

(base) srijanaraut@Srijanas-MBP ~ % mysqldump -h localhost -uroot -p PROJECT3_BLOOD_DONATION_SYSTEM> ~/complete_backup.sql
Enter password:

```

Backing up the database with only two tables Donor and Patient.

```

(base) srijanaraut@Srijanas-MBP ~ % mysqldump -h localhost -uroot -p PROJECT3_BLOOD_DONATION_SYSTEM --tables Patient Donor > backup_tables.sql
Enter password:
(base) srijanaraut@Srijanas-MBP ~ % █

```

### 14. import database(s) that has(have) been already backed up.

To import database, I am using the backup sql file.

```

(base) srijanaraut@Srijanas-MBP ~ % mysql -u root -p PROJECT3_BLOOD_DONATION_SYSTEM < ~/backup_tables.sql
Enter password:
(base) srijanaraut@Srijanas-MBP ~ % █

```

### **Findings while doing Creating the Project 3**

1. Every table can have (but doesn't have to have) a primary key. The primary key of one table may also help to identify records in other tables.
  - a. Blood Bank table has no primary key i.e., ID are not mentioned.
  - b. The Blood\_Bank table references the Donor table through the Donor\_Name field. However, using the donor's name as a foreign key may not be the most effective way to establish this relationship because names are not guaranteed to be unique.
2. To associate as a foreign key that column should hold uniqueness in the table from where it is associated. We have DonorName in Blood Bank table which is the foreign key of Donor table. Thus, it should be unique constraint so that we can reference to foreign key as DonorName table.
3. No Direct relation between Patient's table and the Donor's table.
4. No Direct relation between Patient's table and the Blood\_Bank table.