

DBMS - Mini Project

BLOOD DONATION MANAGEMENT SYSTEM

Name -AYUSHI SOUMYA

SRN – PES1UG20CS097

V Semester

Section _B

Short Description and Scope of the Project

DESCRIPTION:-

Blood donation management system aims at storing data of a blood bank and performing CRUD operations on the same.

The main goal of the system is to project blood bank as well as donor data. The project is entirely administrative.

SCOPE:-

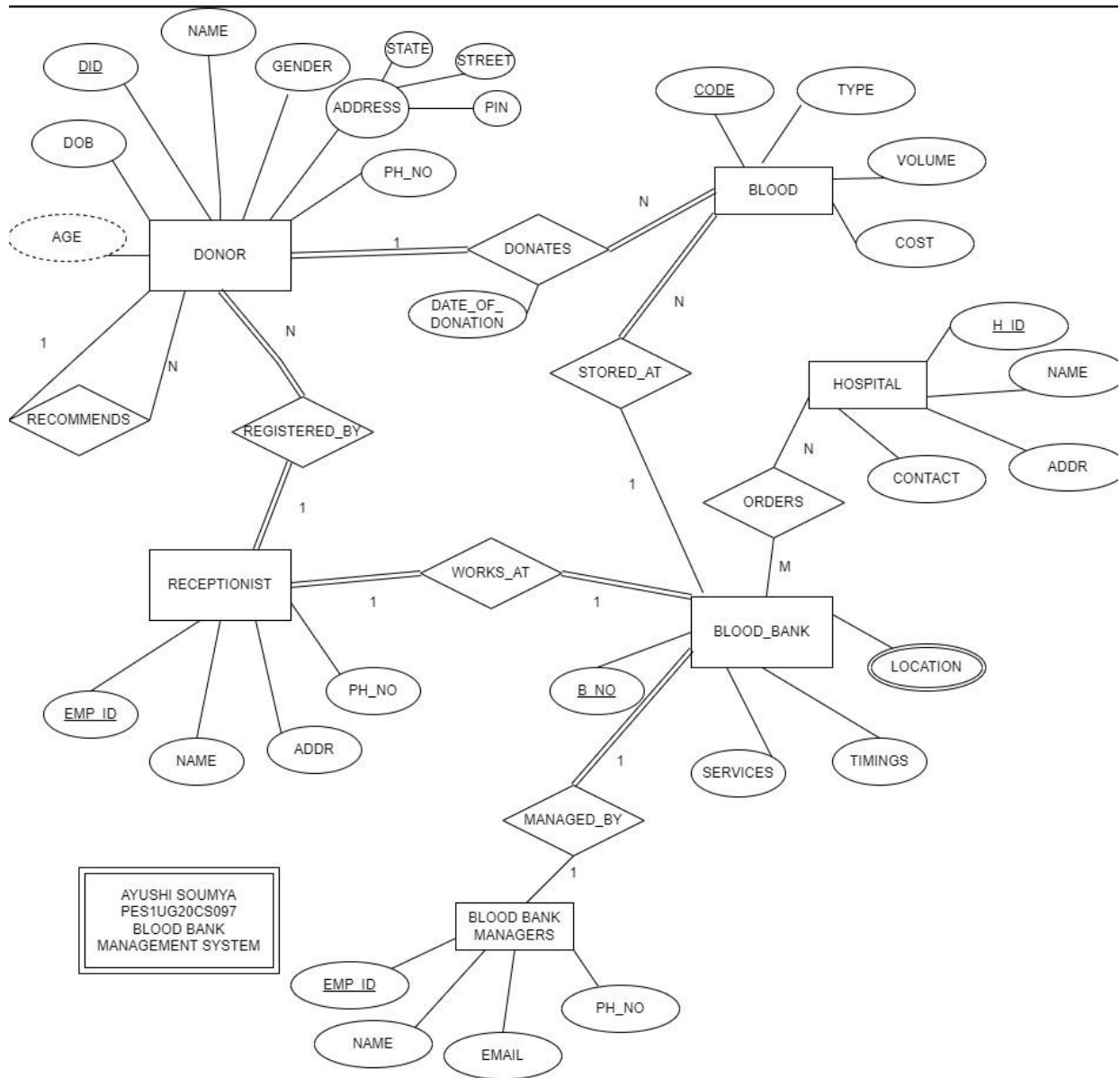
One important scope of this project is ease of availability of blood of all types to patients in need. This ensures providing information about the availability of blood from donors at the blood banks and hospitals easily to the people. This may save time as well as life.

The database contains many entities:

- Donor
- Blood
- Blood bank
- Hospital
- Blood bank manager
- Receptionist

The ERD and relational schema are given below:-

ER Diagram



Relational Schema


```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc blood_bank_manager;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11)       | NO   | PRI | NULL    |       |
| name   | varchar(255)  | YES  |     | NULL    |       |
| email  | varchar(255)  | YES  |     | NULL    |       |
| ph_no  | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.021 sec)
```

**create table blood_bank(B_No int NOT NULL, services
varchar(255),timings varchar(255),MGR_emp_id
int,PRIMARY KEY(B_No), FOREIGN KEY (MGR_emp_id)
references blood_bank_manager(emp_id));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc blood_bank;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| B_No       | int(11)       | NO   | PRI | NULL    |       |
| services   | varchar(255)  | YES  |     | NULL    |       |
| timings    | varchar(255)  | YES  |     | NULL    |       |
| MGR_emp_id | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.021 sec)
```

**create table receptionist(emp_id int NOT NULL, name
varchar(255) , email varchar(255), ph_no
varchar(255),B_no int,PRIMARY KEY(emp_id), FOREIGN
KEY(B_no) references blood_bank(B_No));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc receptionist;
```

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	NO	PRI	NULL	
name	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
ph_no	varchar(255)	YES		NULL	
B_no	int(11)	YES	MUL	NULL	

```
5 rows in set (0.020 sec)
```

**create table Donor(DID int NOT NULL,name
varchar(255),gender varchar(25),DOB DATE,Ph_no
varchar(10),state varchar(255),city varchar(255),pin
varchar(6),recomm_did int,receptionist_emp_id int,
PRIMARY KEY(DID), FOREIGN KEY (recomm_did)
references Donor(DID), FOREIGN KEY
(receptionist_emp_id) references receptionist(emp_id));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc Donor;
```

Field	Type	Null	Key	Default	Extra
DID	int(11)	NO	PRI	NULL	
name	varchar(255)	YES		NULL	
gender	varchar(25)	YES		NULL	
DOB	date	YES		NULL	
Ph_no	varchar(10)	YES		NULL	
state	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
pin	varchar(6)	YES		NULL	
recomm_did	int(11)	YES	MUL	NULL	
receptionist_emp_id	int(11)	YES	MUL	NULL	

```
10 rows in set (0.021 sec)
```

**create table hospital(h_id int NOT NULL,name
varchar(255),address varchar(255), contact int(10),
PRIMARY KEY(h_id));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc hospital;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| h_id  | int(11)       | NO   | PRI | NULL    |       |
| name  | varchar(255)  | YES  |     | NULL    |       |
| address | varchar(255) | YES  |     | NULL    |       |
| contact | int(10)      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.023 sec)
```

**create table blood(code int NOT NULL, type
varchar(10),volume int,cost int,DID int,date_of_donation
DATE,b_b_no int, PRIMARY KEY(code), FOREIGN KEY(DID)
references Donor(DID),FOREIGN KEY(b_b_no) references
blood_bank(B_No));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc blood;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| code           | int(11)       | NO   | PRI | NULL    |       |
| type           | varchar(10)   | YES  |     | NULL    |       |
| volume         | int(11)       | YES  |     | NULL    |       |
| cost           | int(11)       | YES  |     | NULL    |       |
| DID            | int(11)       | YES  | MUL | NULL    |       |
| date_of_donation | date         | YES  |     | NULL    |       |
| b_b_no         | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.022 sec)
```

**create table orders(h_id int NOT NULL, b_no int NOT
NULL, PRIMARY KEY(b_no , h_id), FOREIGN KEY(h_id)
references hospital(h_id),FOREIGN KEY(b_no) references
blood_bank(B_No));**


```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| h_id  | int(11)   | NO   | PRI | NULL    |       |
| b_no  | int(11)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.023 sec)
```

**create table blood_bank_location(b_no int NOT NULL,b_location
varchar(255), PRIMARY KEY(b_no, b_location),FOREIGN KEY (b_no)
references blood_bank(B_No));**

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> desc blood_bank_location;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| b_no       | int(11)       | NO   | PRI | NULL    |       |
| b_location | varchar(255)  | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.024 sec)
```

Tables in our database:

```
+-----+
| Tables_in_dbms_project_pes1ug20cs097 |
+-----+
| blood |
| blood_bank |
| blood_bank_location |
| blood_bank_manager |
| donor |
| hospital |
| orders |
| receptionist |
+-----+
```

Populating the Database

Loading data from the csv files.

blood_bank_manager

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> LOAD DATA INFILE "blood_bank_manager.csv" into table blood_bank_manager
-> COLUMNS TERMINATED BY ','
-> OPTIONALLY ENCLOSED BY '"'
-> ESCAPED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 3 rows affected, 3 warnings (0.010 sec)
Records: 3 Deleted: 0 Skipped: 0 Warnings: 3

MariaDB [DBMS_PROJECT_PES1UG20CS097]> SELECT * FROM blood_bank_manager;
+-----+-----+-----+-----+
| emp_id | name   | email                | ph_no |
+-----+-----+-----+-----+
| 1106   | Abhishek | abhi@gmail.com       | 9178253790 |
| 1107   | Shourya | shourya1@gmail.com   | 9876543210 |
| 1108   | Shreya  | shreya123@gmail.com  | 6758942456 |
+-----+-----+-----+-----+
3 rows in set (0.007 sec)
```

blood_bank

```
MariaDB [DBMS_PROJECT_PES1UG20CS097]> LOAD DATA INFILE "blood_bank.csv" into table blood_bank
-> COLUMNS TERMINATED BY ','
-> OPTIONALLY ENCLOSED BY '"'
-> ESCAPED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 4 rows affected, 4 warnings (0.007 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 4

MariaDB [DBMS_PROJECT_PES1UG20CS097]> select * from blood_bank;
+-----+-----+-----+-----+
| B_No | services                | timings    | MGR_emp_id |
+-----+-----+-----+-----+
| 1    | donation,purchase,tests | all times  | 1106       |
| 2    | donation,test           | 11:00-5:00 | 1106       |
| 3    | donation,test,purchase  | all times  | 1107       |
| 4    | donation,test           | 11:00-5:00 | 1108       |
+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

receptionist.csv

```
Command Prompt - mysql -u root

MariaDB [DBMS_PROJECT_PES1UG20CS097]> LOAD DATA INFILE "receptionist.csv" into table receptionist
-> COLUMNS TERMINATED BY ','
-> OPTIONALLY ENCLOSED BY '"'
-> ESCAPED BY '\\'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES;
Query OK, 4 rows affected, 4 warnings (0.006 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 4

MariaDB [DBMS_PROJECT_PES1UG20CS097]> select * from receptionist;
+-----+-----+-----+-----+-----+
| emp_id | name  | email                | ph_no  | B_no |
+-----+-----+-----+-----+-----+
| 1111   | John  | john567@gmail.com    | 8796374975 | 1   |
| 1112   | Mary  | mary2001@gmail.com   | 8356254778 | 2   |
| 1113   | Selena | s123leena@gmail.com  | 9756385622 | 3   |
| 1114   | Veronica | ver567@gmail.com    | 9423745610 | 4   |
+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

Similarly we insert values in all the tables .

Our database is then ready to run queries and extract information.

Join Queries

Showcase at least 2 join queries

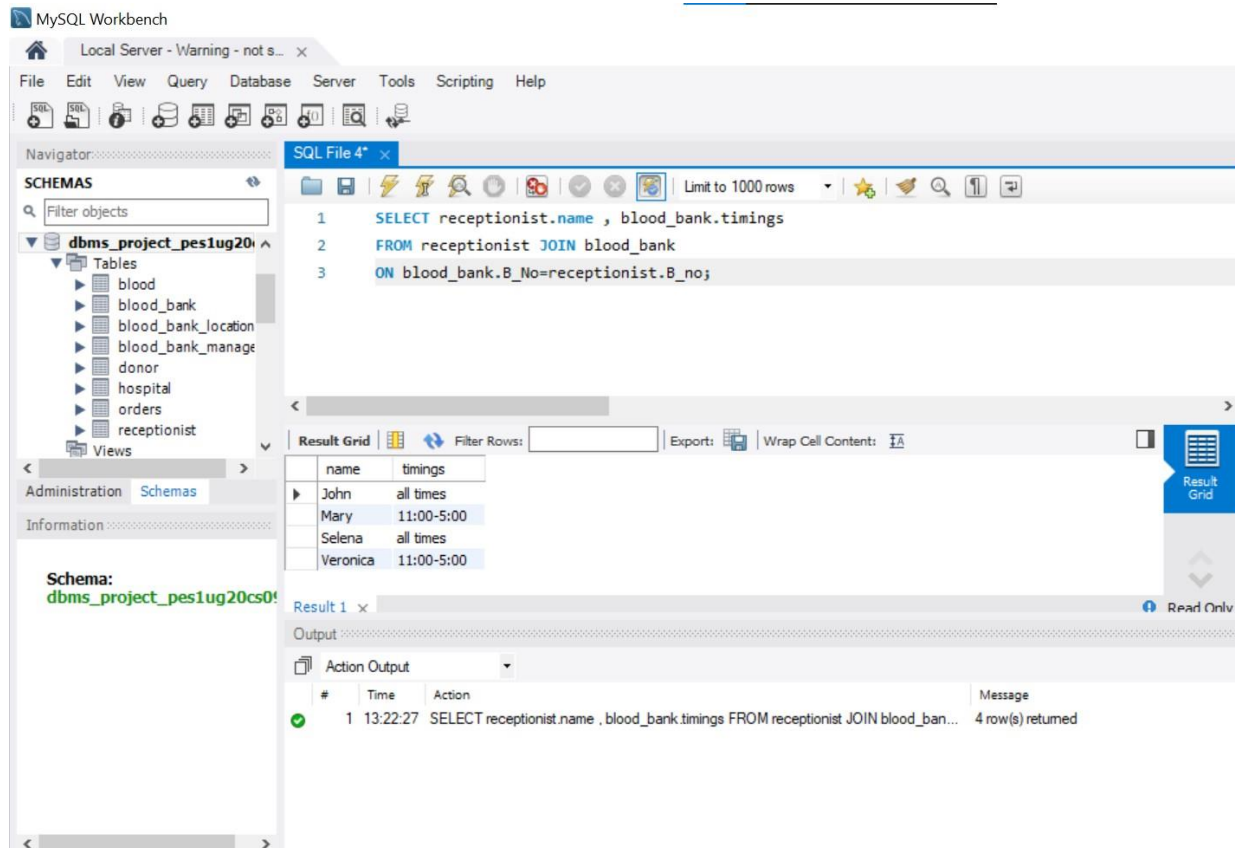
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

Display the name of the receptionist and the timings when she /he will be available in the blood banks.

```
SELECT receptionist.name , blood_bank.timings
```

```
FROM receptionist JOIN blood_bank
```

```
ON blood_bank.B_No=receptionist.B_no;
```



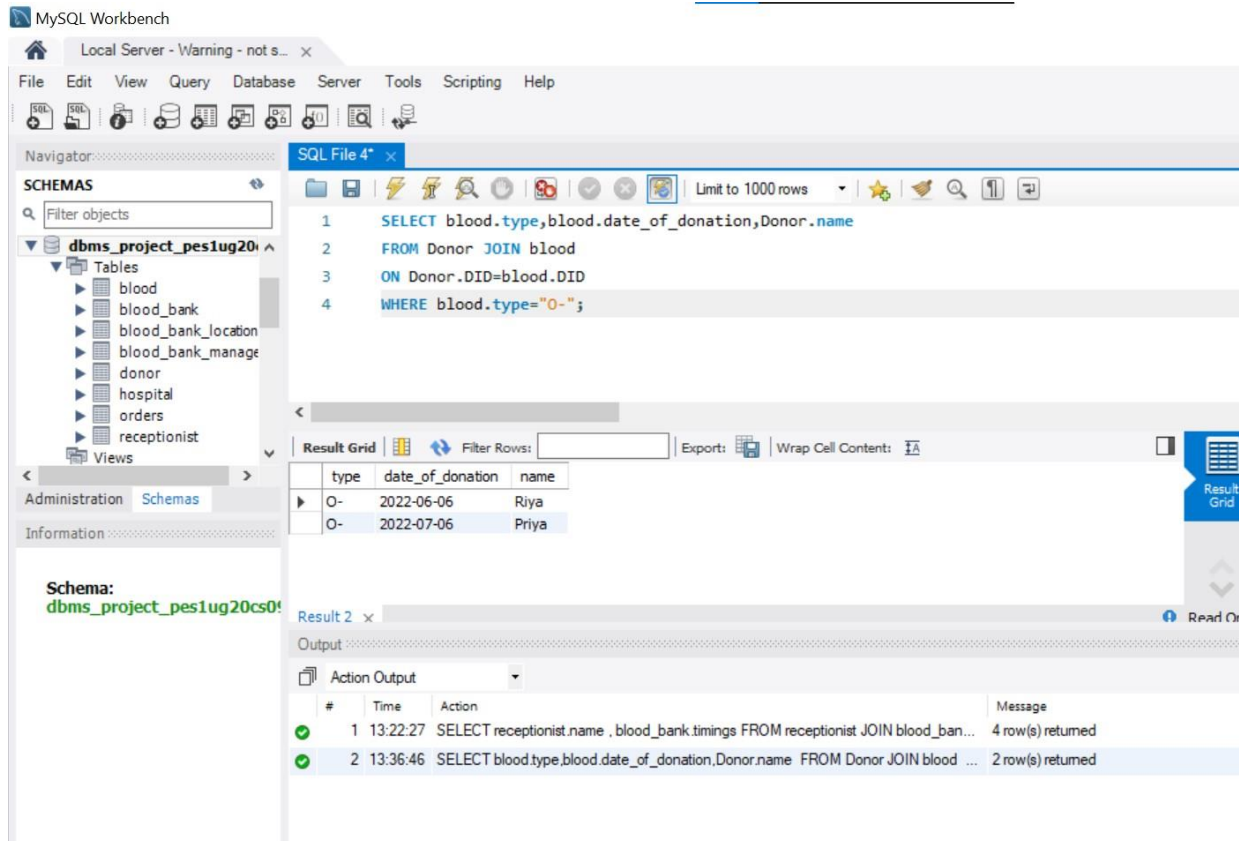
Display the date of donation, blood type and the donor name of all the O- blood groups.

```
SELECT blood.type,blood.date_of_donation,Donor.name
```

```
FROM Donor JOIN blood
```

```
ON Donor.DID=blood.DID
```

```
WHERE blood.type="O-";
```



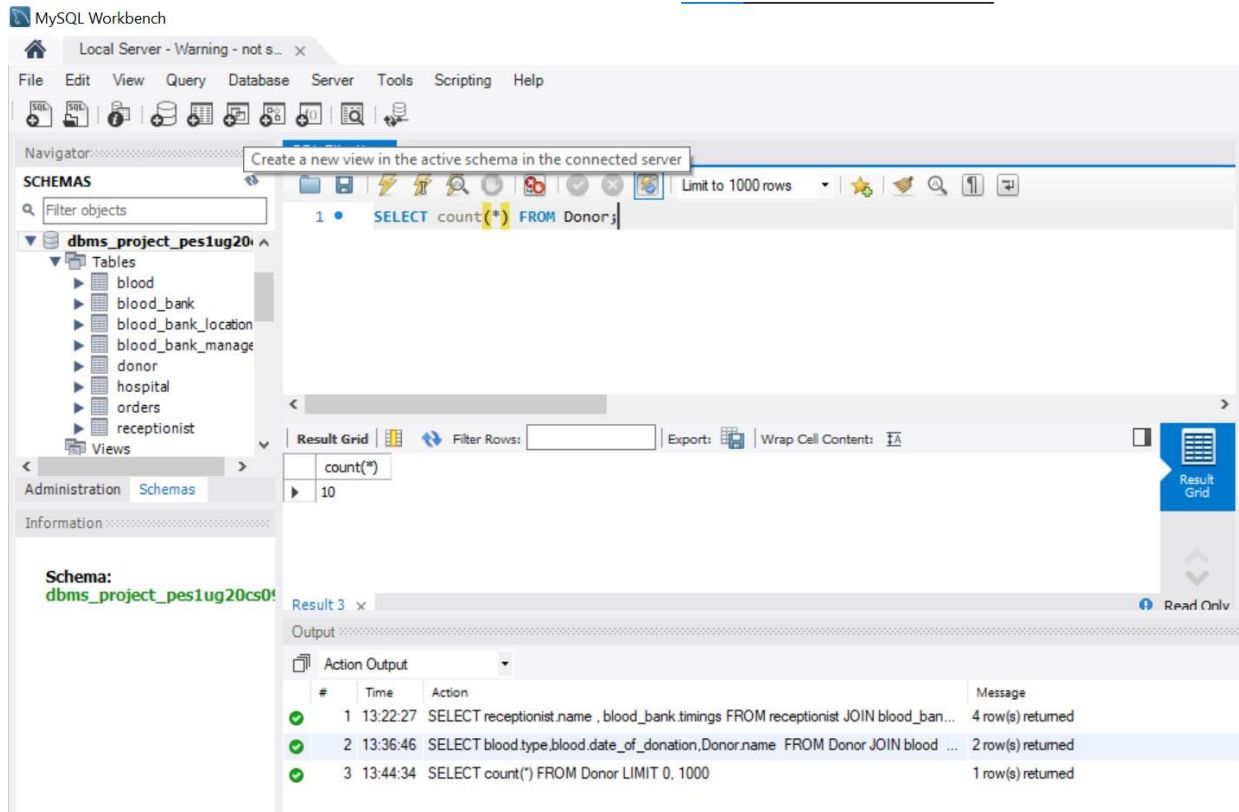
Aggregate Functions

Showcase at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

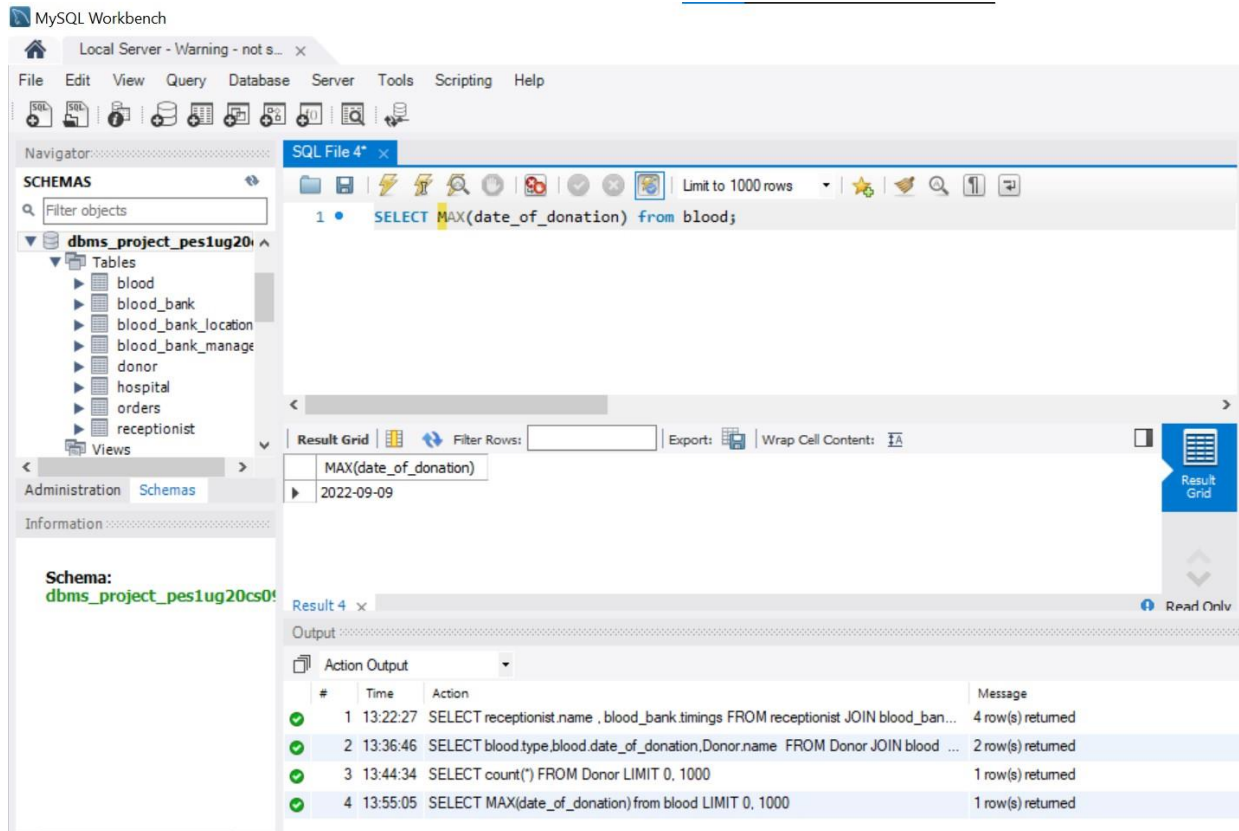
Aggregate functions- count, min, max , avg ;

Count the number of donors



Displaying the latest donation:

```
SELECT MAX(date_of_donation) from blood;
```



Set Operations

Showcase at least 4 Set Operations queries

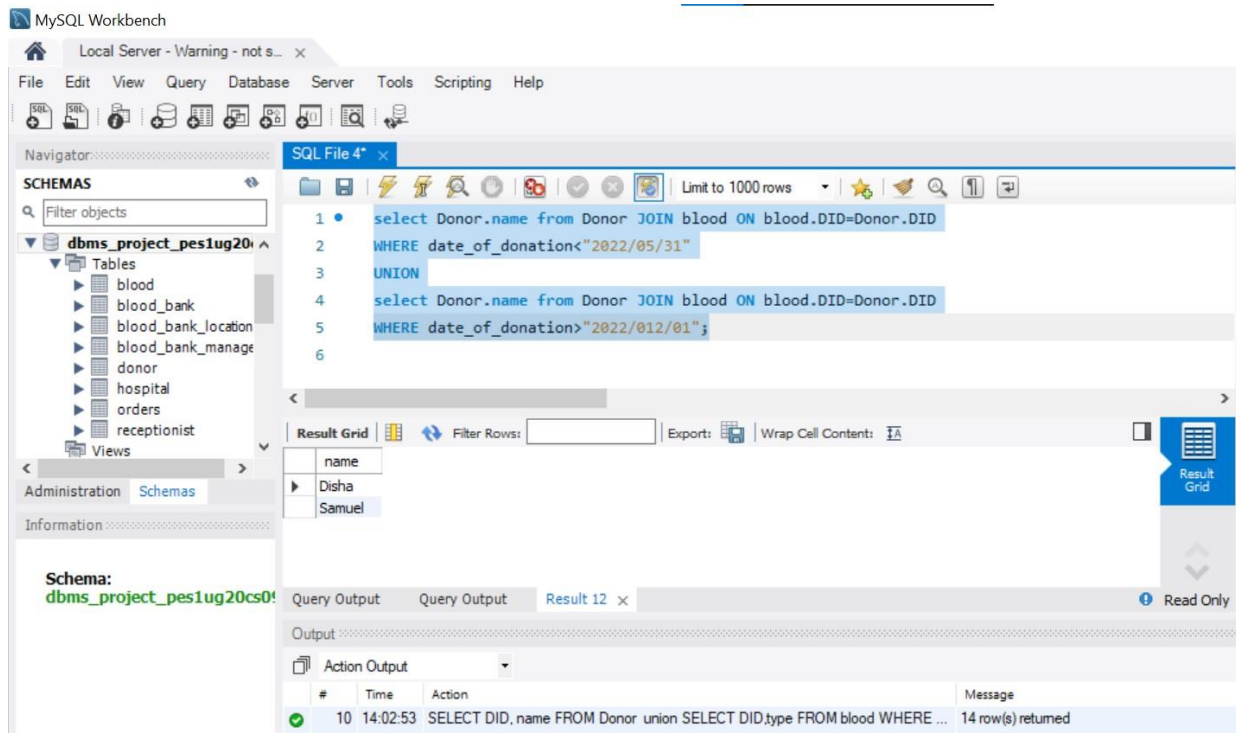
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

UNION, INTERSECT, MINUS

UNION:

Find list of all donors who donated between feb and may 2022:

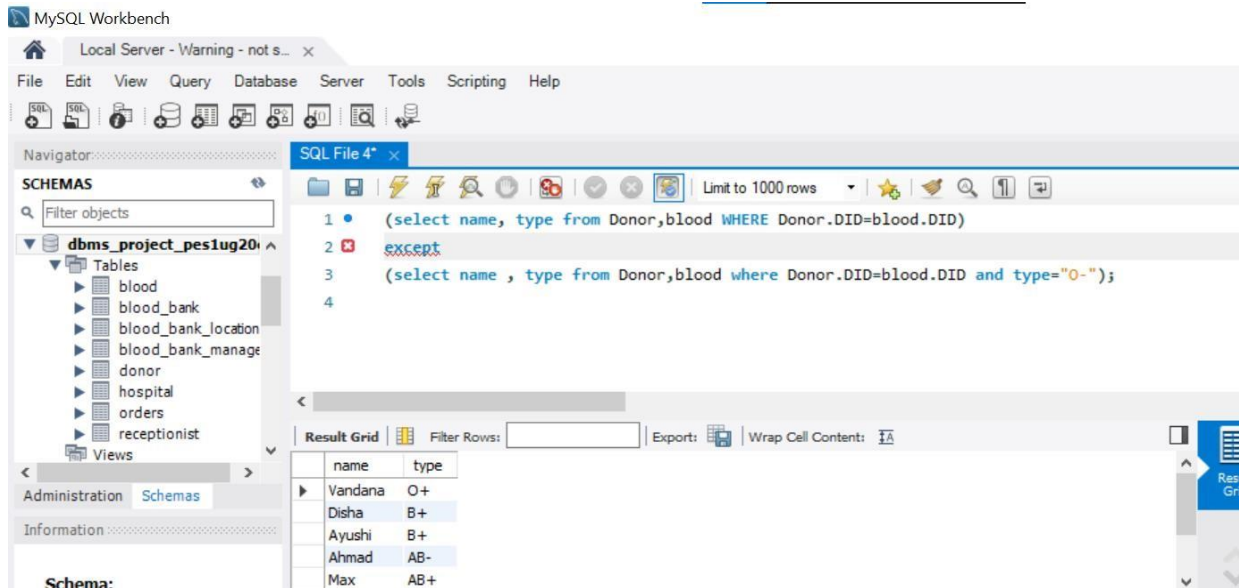
```
select Donor.name from Donor JOIN blood ON blood.DID=Donor.DID
WHERE date_of_donation<"2022/05/31" UNION select Donor.name
from Donor JOIN blood ON blood.DID=Donor.DID
WHERE date_of_donation>"2022/012/01";
```



Find all donors who aren't O-

MINUS-

```
(select name, type from Donor,blood WHERE Donor.DID=blood.DID) except
(select name , type from Donor,blood where Donor.DID=blood.DID and type="O-");
```

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

FUNCTION:

Write a function to find the availability of a particular type of blood .if the date of donation is before july 2022 then return a message that the blood is expired

CODE:

DELIMITER \$\$

```

CREATE DEFINER=`root`@`localhost` FUNCTION `availability`(type_of_blood varchar(10))
RETURNS varchar(50) CHARSET utf8mb4

```

DETERMINISTIC

```

BEGIN

DECLARE RESULT VARCHAR(50);

DECLARE num int;

SELECT count(*) into num from blood where type=type_of_blood
and date_of_donation>"2022/05/01"; if(num>0)

THEN

SET RESULT="AVAILABLE";

else

SET RESULT="NOT AVAILABLE";

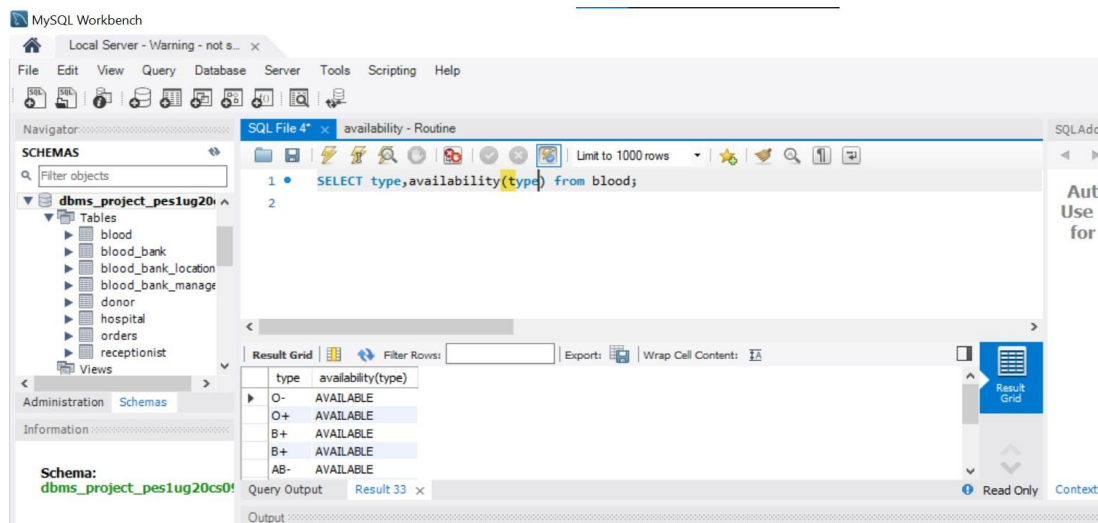
END IF;

RETURN RESULT;

END;

$$ DELIMITER

```



PROCEDURES:

Updating the cost of given type by 200 more.

Code:

Delimiter \$\$

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `new_cost`(IN type_of_blood varchar(10), IN cost int)
```

```
BEGIN
```

```
    DECLARE new_cost int; set new_cost=cost+200; update  
    blood set cost=new_cost where type_of_blood=type;
```

```
END;
```

```
$$ DELIMITER
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the database 'dbms_project_pes1ug20' selected. The main window shows a SQL query in the 'SQL File 4' editor:

```
1 • SELECT * FROM BLOOD WHERE TYPE="O-";  
2 • CALL new_cost("O-",700);  
3 • SELECT * FROM BLOOD WHERE TYPE="O-";  
4  
5  
6
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has columns: code, type, volume, cost, DID, date_of_donation, and b_b_no. The results are as follows:

code	type	volume	cost	DID	date_of_donation	b_b_no
301	O-	1	700	1501	2022-06-06	3
309	O-	1	700	1509	2022-07-06	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

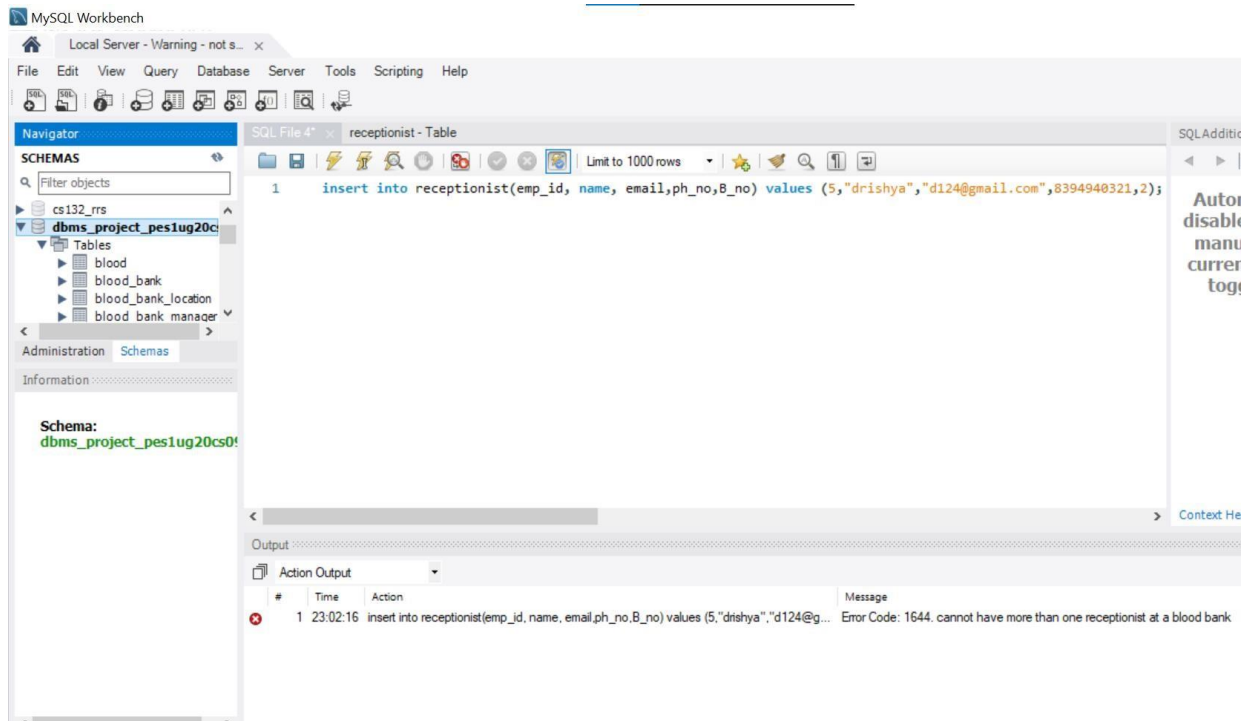
The bottom status bar shows 'Query Output' and 'BLOOD 36'.

Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

Trigger to display error message when more than one receptionist is inserted for a blood_bank

```
CREATE DEFINER='root'@'localhost' TRIGGER BeforeInsert
BEFORE INSERT
ON receptionist FOR EACH ROW
BEGIN
    DECLARE err_msg VARCHAR(255);
    DECLARE n int;
    SET err_msg=('cannot have more than one receptionist at a blood bank');
    select count(B_no) into n from receptionist where B_no=B_no; if(n>1)
THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = err_msg;
END IF;
END
```



CURSORS:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `proc_employee`()
```

```
BEGIN
```

```
    DECLARE emp_id VARCHAR(50);
```

```
    DECLARE name VARCHAR(50);
```

```
    DECLARE var_check_row INTEGER DEFAULT 0;
```

```
    DECLARE c1 CURSOR FOR SELECT emp_id, name FROM blood_bank_manager;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND set var_check_row=1;
```

```
    OPEN c1; get_emp : LOOP
```

```
        FETCH c1 INTO emp_id,name;
```

```
        IF var_check_row=1 THEN
```

```
            LEAVE get_emp;
```

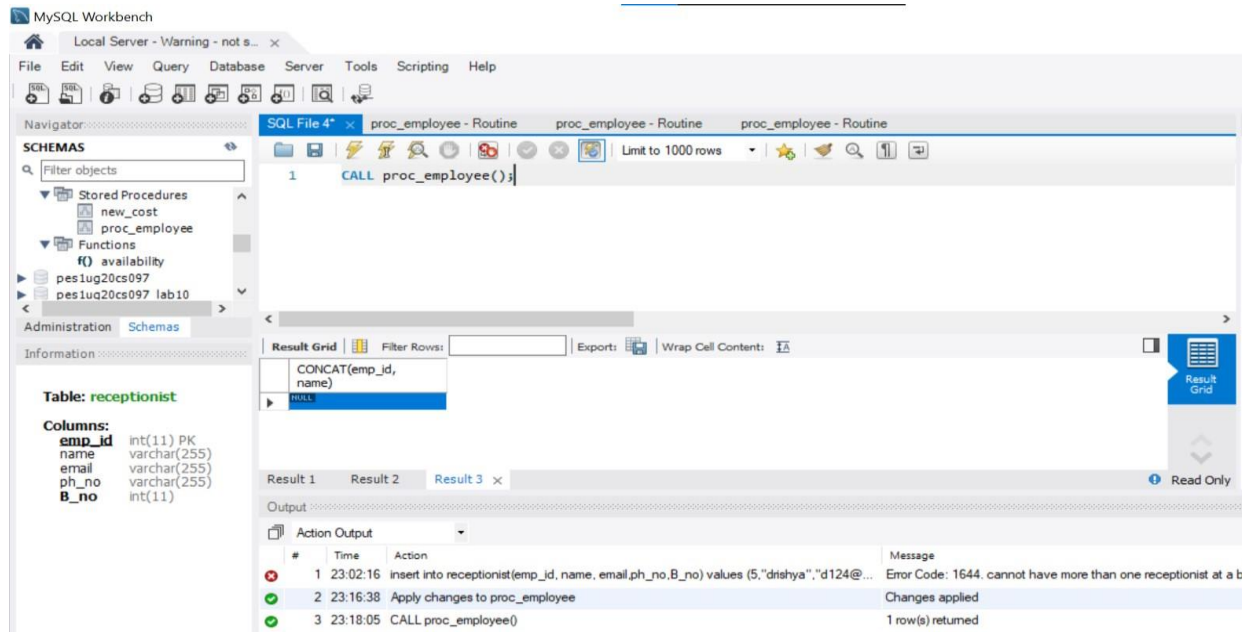
```
        END IF;
```

```
        SELECT CONCAT(emp_id, name);
```

```
    END LOOP get_emp;
```

```
    CLOSE c1;
```

```
END
```

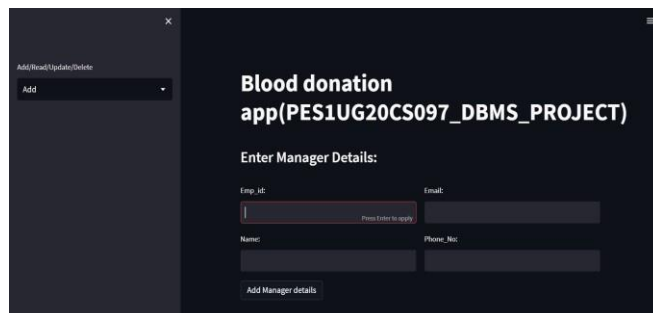


Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

GUI



1. PERFORMING CRUD OPERATIONS USING GUI:

ADDITION:

Add/Read/Update/Delete

Add

Blood donation app(PES1UG20CS097_DBMS_PROJECT)

Enter Manager Details:

Emp_id:

1109

Email:

shre34@gmail.com

Name:

shreyash

Phone_No:

9766454220

Add Manager details

Successfully added Manager: {} going to {}

READ:

Add/Read/Update/Delete

Read

Blood donation app(PES1UG20CS097_DBMS_PROJECT)

Read Manager details

View all Manager

	emp_id	name	email	ph_no
0	1106	Abhishek	abhi@gmail.com	9178253790
1	1107	Shourya	shourya1@gmail.com	9876543210
2	1108	Shreya	shreya123@gmail.com	6758942456
3	1109	shreyash	shre34@gmail.com	9766454220

MODIFICATION:

Add/Read/Update/Delete

Edit

app(F210G20C3097_DBMS_PROJECT)

Update Manager details

Current Data

	emp_id	name	email	ph_no
0	1106	Abhishek	abhi@gmail.com	9178253790
1	1107	Shourya	shourya1@gmail.com	9876543210
2	1108	Shreya	shreya123@gmail.com	6758942456

Email_id:

1108

email

shreya1234@gmail.com

Update email

Successfully updated

DELETION:

Add/Read/Update/Delete

Remove

Delete Manager

Current data

emp_id

1109

Do you want to delete ::1109

Delete Manager

manager has been deleted successfully

Updated data

2. Window to accept and run queries:

×

Add/Read/Update/Delete/Query

Query

Blood donation app(PES1UG20CS097_DBMS_PROJECT)

query and extract data from manager table

Enter Query:

select * from blood_bank_manager where emp_id<1108;

Run Query

	0	1	2	3
0	1106	Abhishek	abhi@gmail.com	9178253790
1	1107	Shourya	shourya1@gmail.com	9876543210

MODIFICATIONS ASKED TO BE DONE:

- 1.) CRUD OPERATIONS ON ALL THE MAIN TABLES:

×

About/Add/Read/Update/Delete/Query/DonorList

Add

Blood donation app(PES1UG20CS097_DBMS_PROJECT)

Add Manager Details: ▾

Add Donor Details: ▾

Add Blood_bank Details: ▾

Made with Streamlit

About/Add/Read/Update/Delete/Query/DonorList

Read

View all manager

	0	1	2	3
0	1106	Abhishek	abhi@gmail.com	9178253790
1	1107	Shourya	shourya1@gmail.com	9876543210
2	1108	Shreya	sh123@gmail.com	6758942456

View all donor

	0	1	2	3	4	5	6	7	8
0	1501	Riya	F	1997-01-02	9744665578	karnataka	bangalore	560065	<NA>
1	1502	Vandana	F	2001-01-02	8364463364	goa	panjim	234589	1,501.0000
2	1503	Disha	F	1995-03-06	6478392740	andhra pradesh	vijaywada	561103	1,502.0000
3	1504	Ayushi	F	2000-12-12	8344592472	karnataka	bangalore	560034	1,503.0000
4	1505	Ahmad	M	1996-08-03	7763849921	karnataka	tumkur	560012	1,504.0000

2.) Creating a Find Donor Page which takes the blood group as input and displays the donor details:

About/Add/Read/Update/Delete/Query/DonorList

Find Donor

Blood donation app(PES1UG20CS097_DBMS_PROJECT)

Donor List

Enter blood type:

O+

Get Donor List

	DONOR_ID	name	ph_no
0	1502	Vandana	8364463364
1	1507	Samuel	8000122910