**Project Name:** Blood Bank Management System

**Group Member:**

| Name | ID | Contribution |
|---|---|---|
| TAWSIF ZAWAD | 18-36327-1 | 25% |
| MOSAEB BIN MOZIB | 18-38598-2 | 25% |
| MD RAZIB MOLLAH | 20-42153-1 | 25% |
| ASIF IQBAL | 20-43187-1 | 25% |

**Course Name:** Advance Database Management System

**Section: B**

**Course Teacher:** Juena Ahmed Noshin.

# Contents

## Introduction:

The Blood Bank Management System is a web-based application that stores, processes, retrieves, and analyzes data about blood bank administration. It also supervises blood inventory management and other blood bank-related activities. The major goal of the blood bank management system is to keep track of blood, donors, blood groups, blood banks, and stock information. It keeps track of all information concerning blood, blood cells, stocks, and blood. Because the project is all done at the administrative level, only the administrator can see it. There are a number of scenarios where an urgent need for blood comes. At this critical time, the Online Blood Bank project aims at maintaining all the information related to blood donors. Through this application, any person who is interested in donating the blood can register himself. The number of persons who are in need of blood is increasing in large number day by day. In order to help people who are in need of blood, Online Blood Bank can be used effectively for getting the details of blood donors having the same blood group. With the help of Online Blood Bank people who are having the thought of donating blood get registered in Online Blood Bank giving their total details. The Blood bank management system is designed to provide blood for the people who are in need of blood. This system is planned to collect blood from many donators in short from various sources and distribute that blood to needy people who require blood.

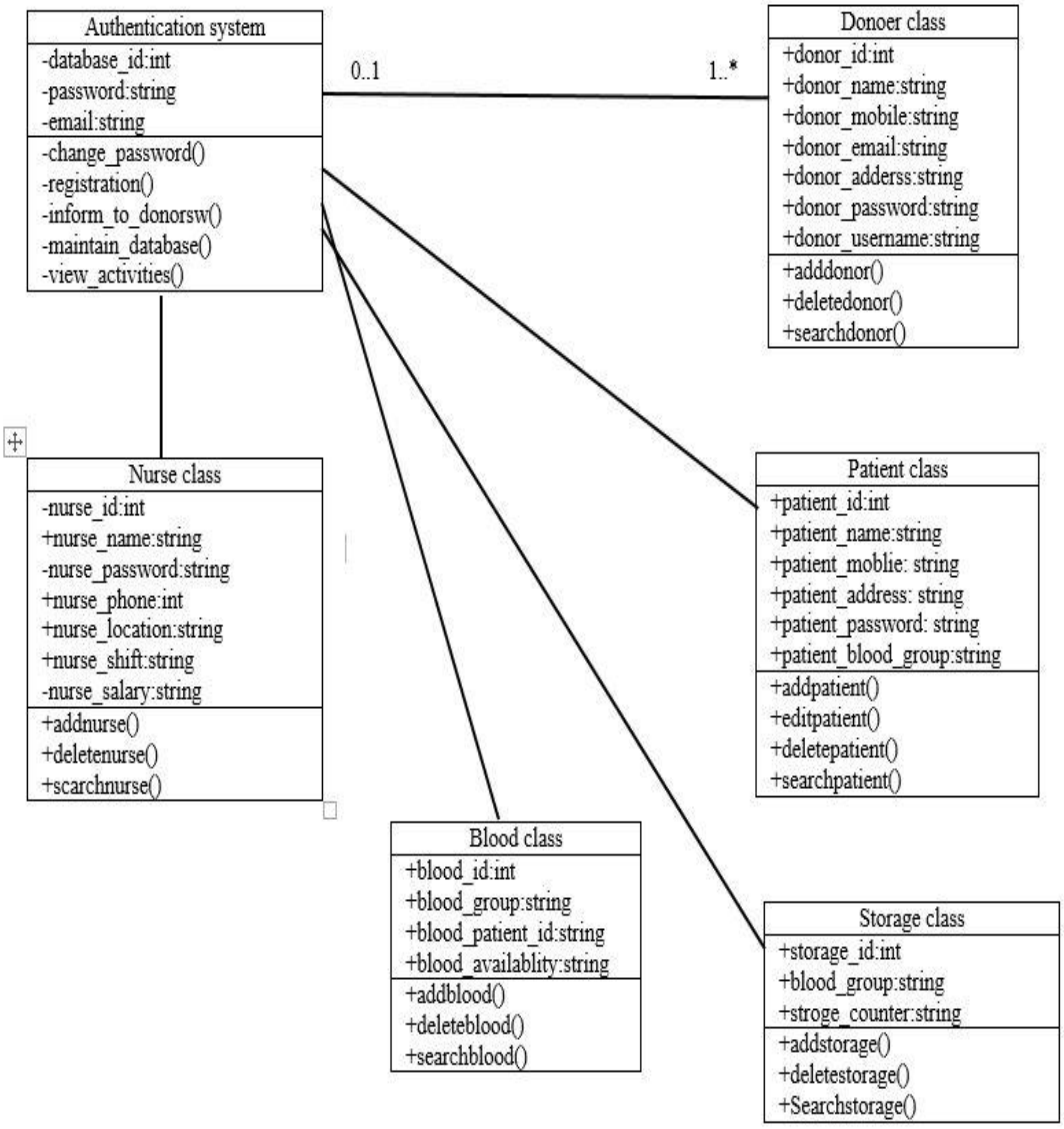
## Why we need this:

The primary purpose of the blood bank management system is to simplify and automate the process of looking for blood in the event of an emergency, as well as to keep track of blood donors, employees, and blood stocks in the bank.

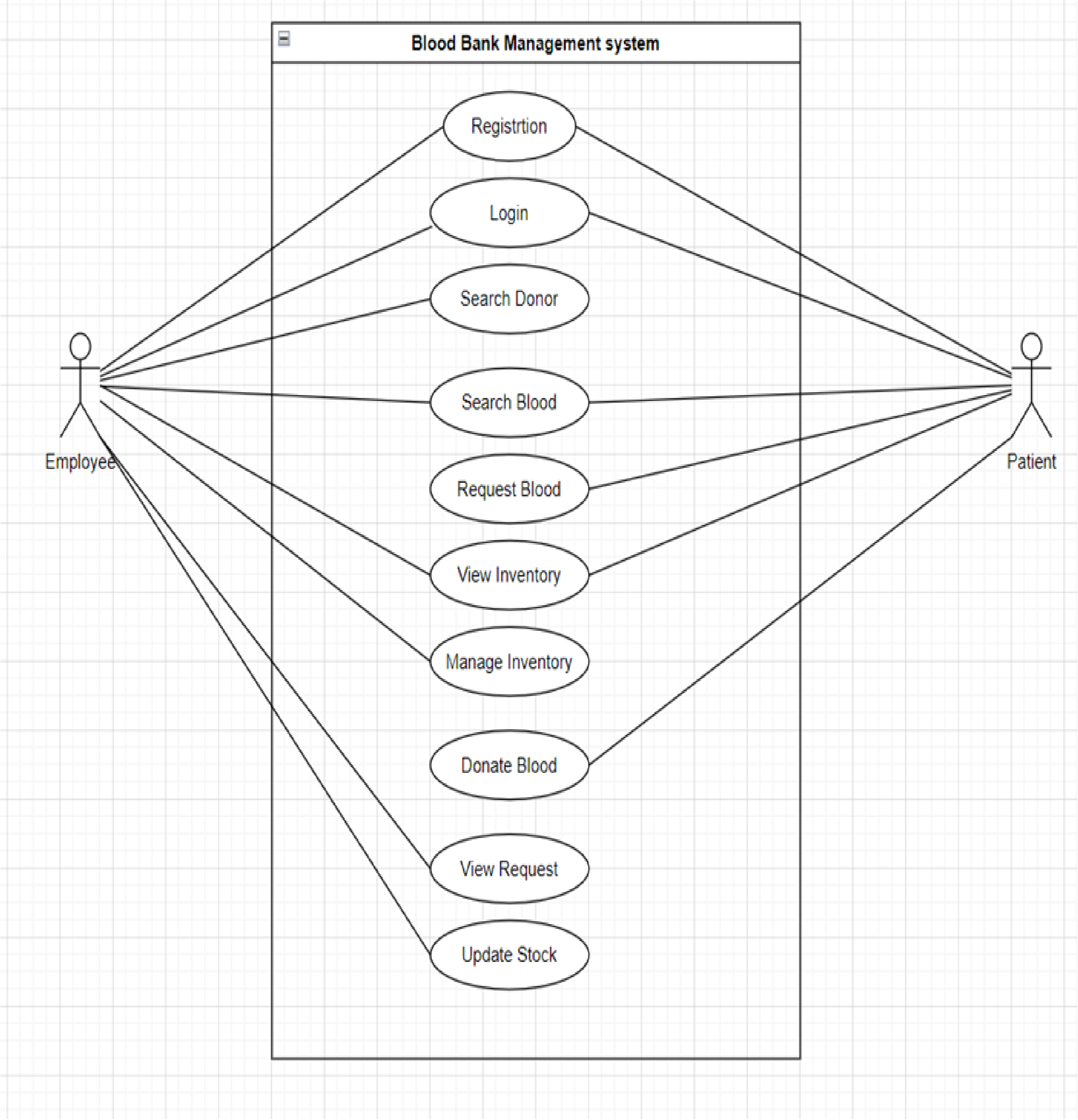
- To offer the blood bank with effective donor and blood stock management activities by recording donor and blood details.
- To provide a consolidated donor and blood stock database that is synchronized.
- To provide instant data and information storage and retrieval.

## Class Diagram:

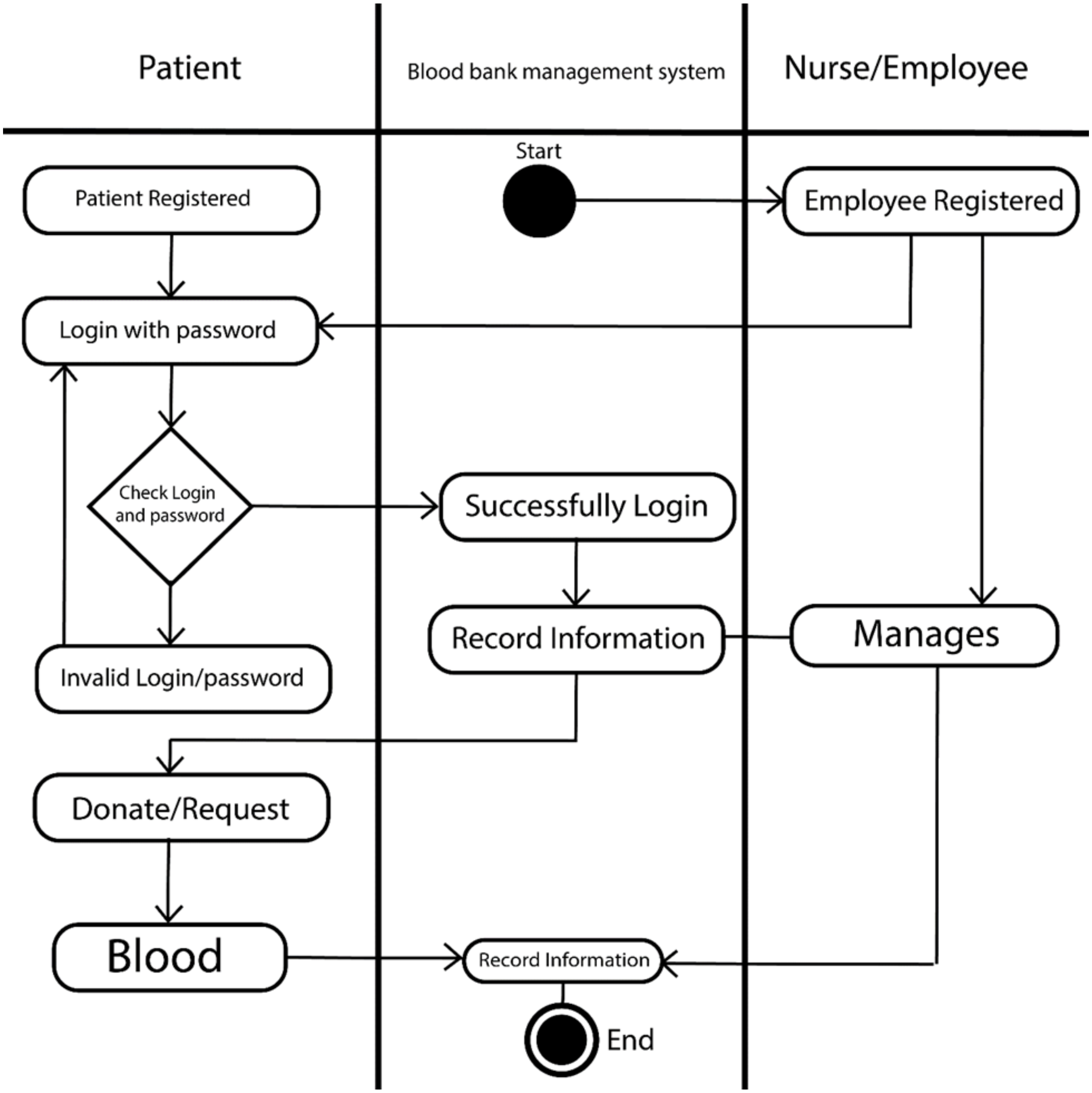
Class Diagram:



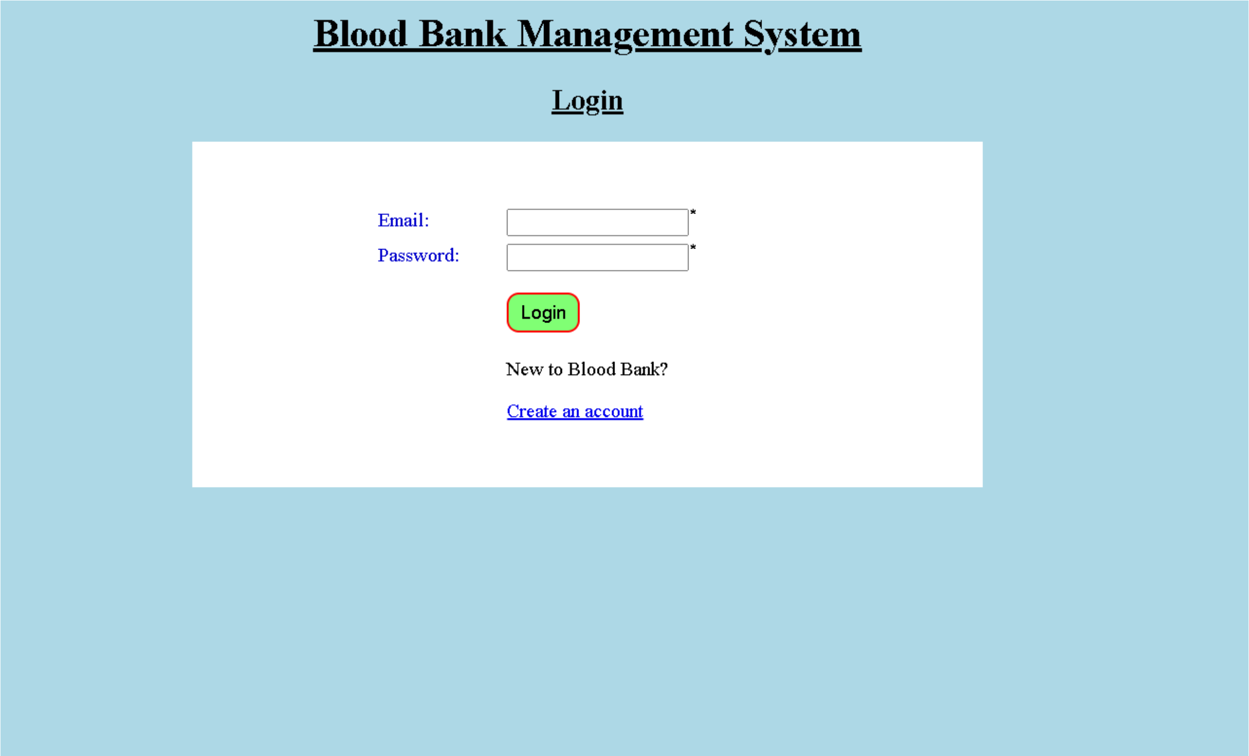
## Use Case Diagram:



## Activity Diagram:



## User Interface:



Blood Bank Management System

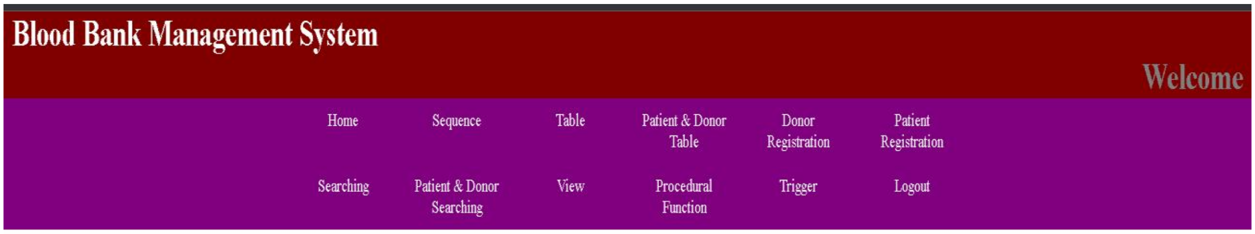
Login

Email: *

Password: *

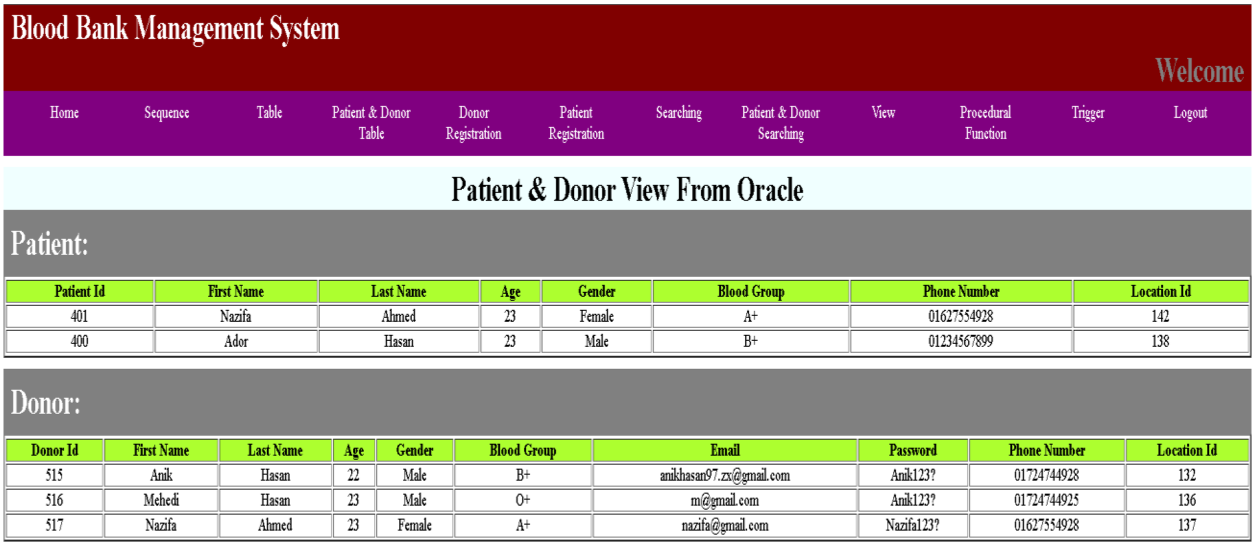
New to Blood Bank?

Create an account



Blood Bank Management System

Welcome



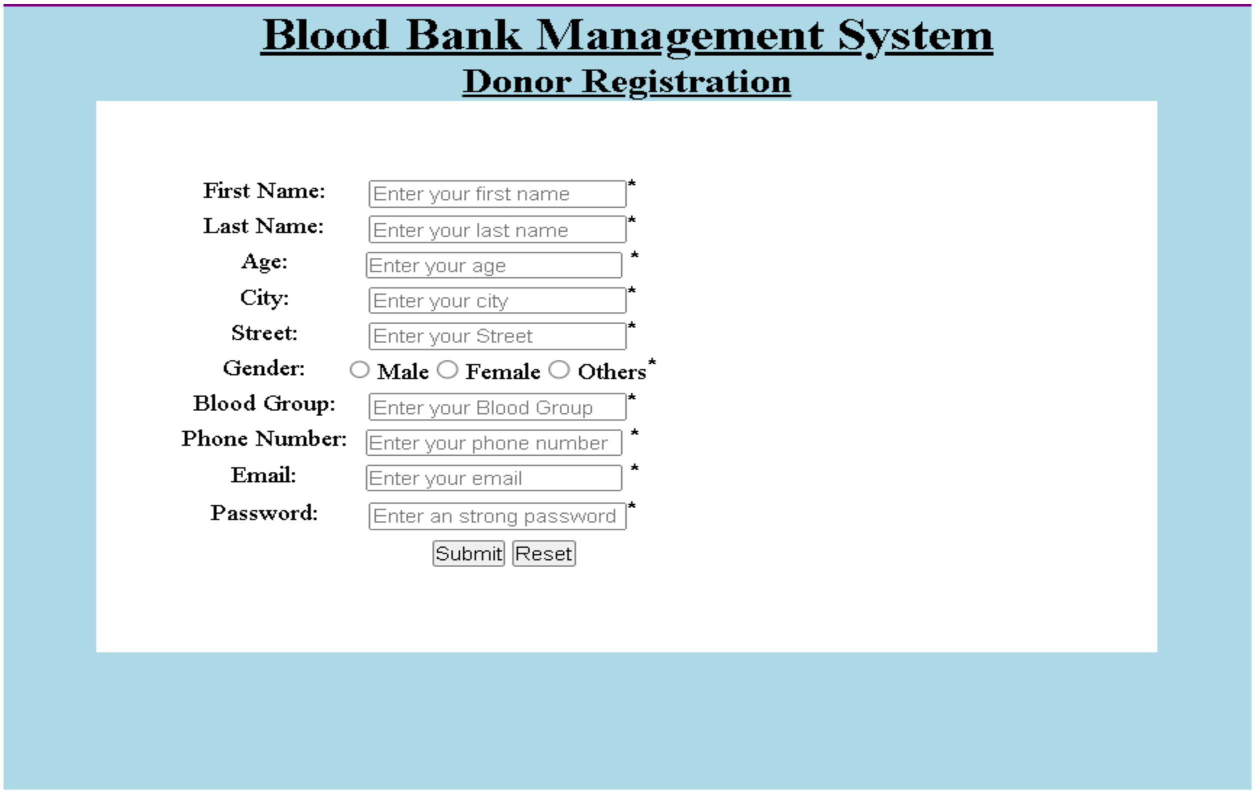
Patient & Donor View From Oracle

Patient:

| Patient Id | First Name | Last Name | Age | Gender | Blood Group | Phone Number | Location Id |
|---|---|---|---|---|---|---|---|
| 401 | Nazifa | Ahmed | 23 | Female | A+ | 01627554928 | 142 |
| 400 | Ador | Hasan | 23 | Male | B+ | 01234567899 | 138 |

Donor:

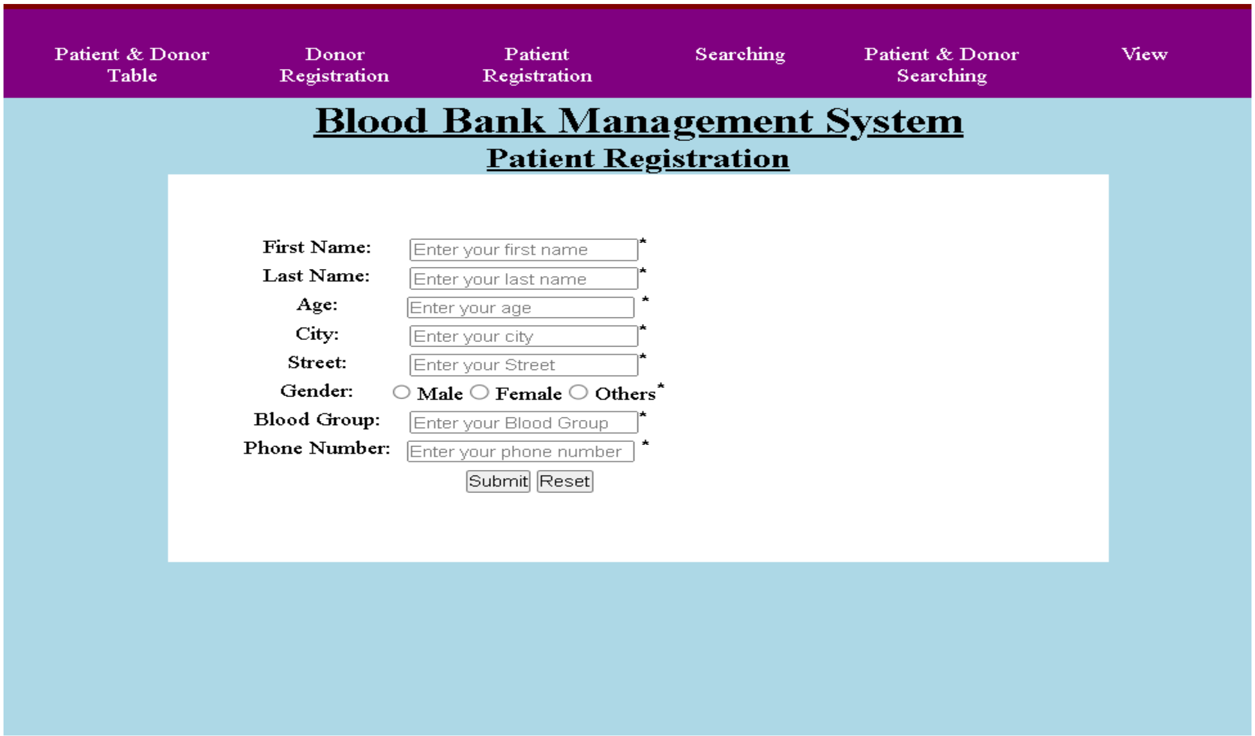
| Donor Id | First Name | Last Name | Age | Gender | Blood Group | Email | Password | Phone Number | Location Id |
|---|---|---|---|---|---|---|---|---|---|
| 515 | Anik | Hasan | 22 | Male | B+ | anikhasan97.zx@gmail.com | Anik123? | 01724744928 | 132 |
| 516 | Mehedi | Hasan | 23 | Male | O+ | m@gmail.com | Anik123? | 01724744925 | 136 |
| 517 | Nazifa | Ahmed | 23 | Female | A+ | nazifa@gmail.com | Nazifa123? | 01627554928 | 137 |



Blood Bank Management System

Donor Registration

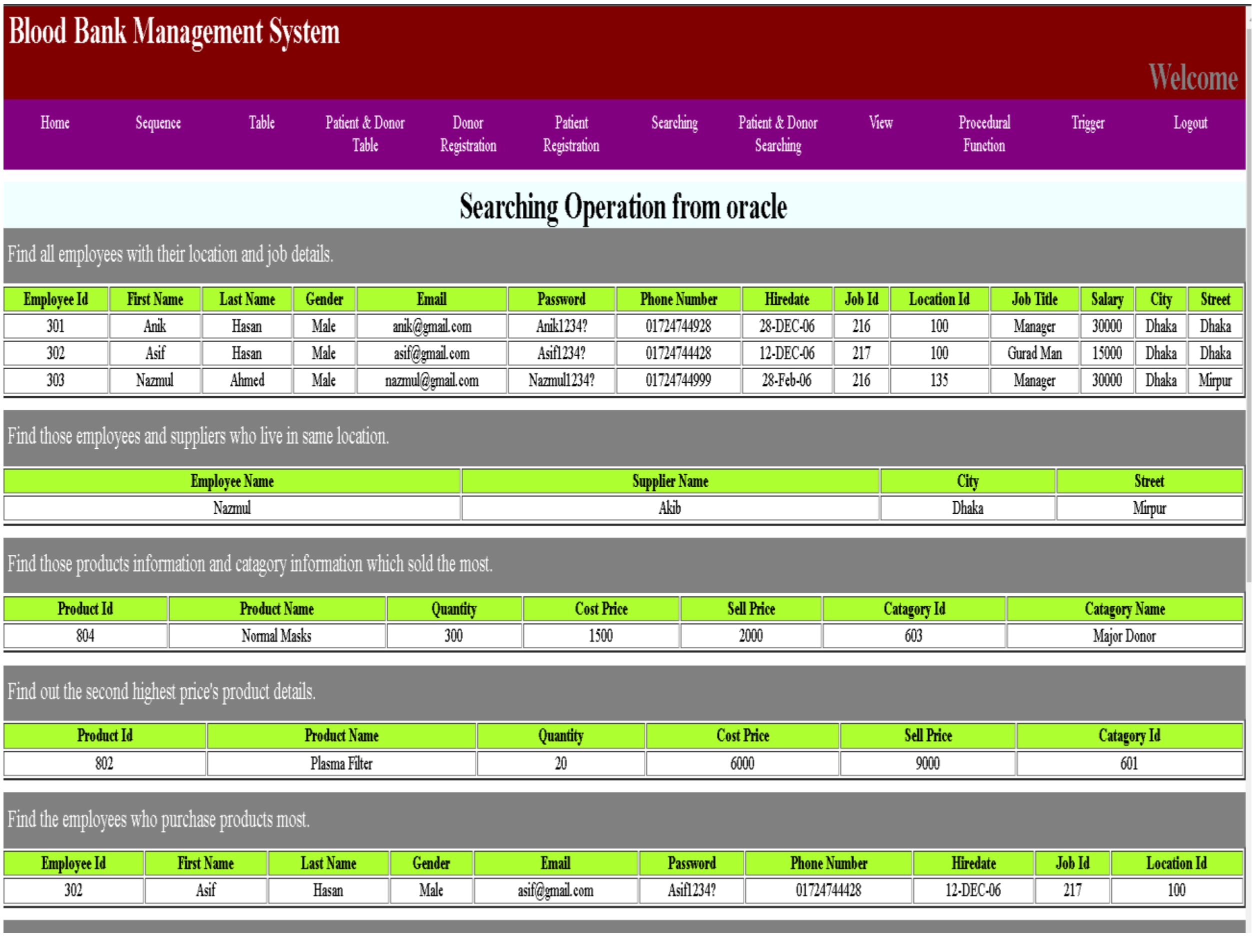
| | |
|---|---|
| First Name: | Enter your first name * |
| Last Name: | Enter your last name * |
| Age: | Enter your age * |
| City: | Enter your city * |
| Street: | Enter your Street * |
| Gender: | ○ Male ○ Female ○ Others * |
| Blood Group: | Enter your Blood Group * |
| Phone Number: | Enter your phone number * |
| Email: | Enter your email * |
| Password: | Enter an strong password * |



# Blood Bank Management System

## Patient Registration

| | |
|---|---|
| First Name: | Enter your first name * |
| Last Name: | Enter your last name * |
| Age: | Enter your age * |
| City: | Enter your city * |
| Street: | Enter your Street * |
| Gender: | ○ Male ○ Female ○ Others * |
| Blood Group: | Enter your Blood Group * |
| Phone Number: | Enter your phone number * |



## Searching Operation from oracle

Find all employees with their location and job details.

| Employee Id | First Name | Last Name | Gender | Email | Password | Phone Number | Hiredate | Job Id | Location Id | Job Title | Salary | City | Street |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 301 | Anik | Hasan | Male | anik@gmail.com | Anik1234? | 01724744928 | 28-DEC-06 | 216 | 100 | Manager | 30000 | Dhaka | Dhaka |
| 302 | Asif | Hasan | Male | asif@gmail.com | Asif1234? | 01724744428 | 12-DEC-06 | 217 | 100 | Gurad Man | 15000 | Dhaka | Dhaka |
| 303 | Nazmul | Ahmed | Male | nazmul@gmail.com | Nazmul1234? | 01724744999 | 28-Feb-06 | 216 | 135 | Manager | 30000 | Dhaka | Mirpur |

Find those employees and suppliers who live in same location.

| Employee Name | Supplier Name | City | Street |
|---|---|---|---|
| Nazmul | Akib | Dhaka | Mirpur |

Find those products information and catagory information which sold the most.

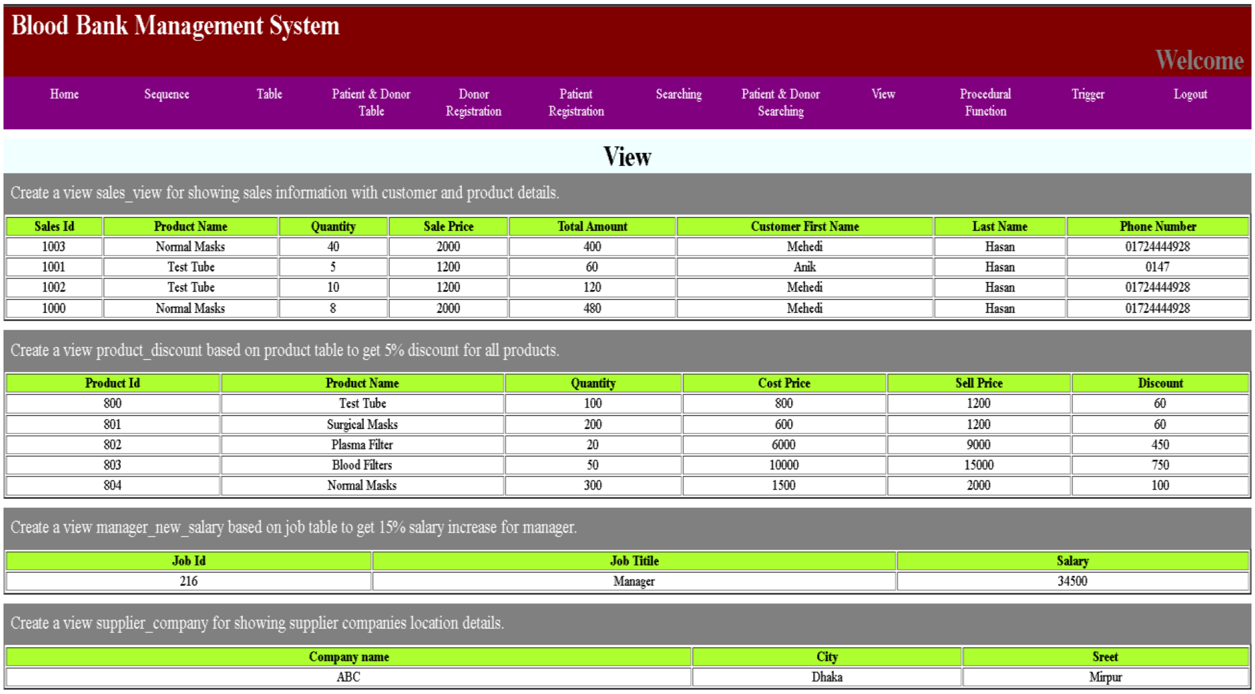
| Product Id | Product Name | Quantity | Cost Price | Sell Price | Catagory Id | Catagory Name |
|---|---|---|---|---|---|---|
| 804 | Normal Masks | 300 | 1500 | 2000 | 603 | Major Donor |

Find out the second highest price's product details.

| Product Id | Product Name | Quantity | Cost Price | Sell Price | Catagory Id |
|---|---|---|---|---|---|
| 802 | Plasma Filter | 20 | 6000 | 9000 | 601 |

Find the employees who purchase products most.

| Employee Id | First Name | Last Name | Gender | Email | Password | Phone Number | Hiredate | Job Id | Location Id |
|---|---|---|---|---|---|---|---|---|---|
| 302 | Asif | Hasan | Male | asif@gmail.com | Asif1234? | 01724744428 | 12-DEC-06 | 217 | 100 |



# View

Create a view sales_view for showing sales information with customer and product details.

| Sales Id | Product Name | Quantity | Sale Price | Total Amount | Customer First Name | Last Name | Phone Number |
|---|---|---|---|---|---|---|---|
| 1003 | Normal Masks | 40 | 2000 | 400 | Mehedi | Hasan | 01724444928 |
| 1001 | Test Tube | 5 | 1200 | 60 | Anik | Hasan | 0147 |
| 1002 | Test Tube | 10 | 1200 | 120 | Mehedi | Hasan | 01724444928 |
| 1000 | Normal Masks | 8 | 2000 | 480 | Mehedi | Hasan | 01724444928 |

Create a view product_discount based on product table to get 5% discount for all products.

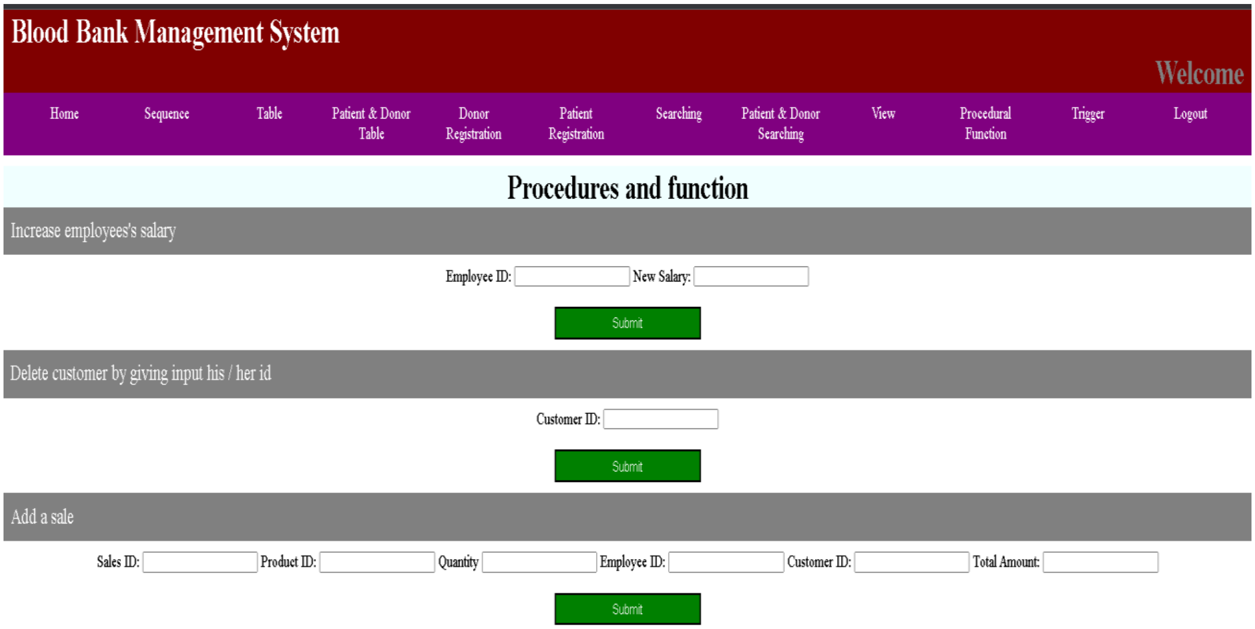
| Product Id | Product Name | Quantity | Cost Price | Sell Price | Discount |
|---|---|---|---|---|---|
| 800 | Test Tube | 100 | 800 | 1200 | 60 |
| 801 | Surgical Masks | 200 | 600 | 1200 | 60 |
| 802 | Plasma Filter | 20 | 6000 | 9000 | 450 |
| 803 | Blood Filters | 50 | 10000 | 15000 | 750 |
| 804 | Normal Masks | 300 | 1500 | 2000 | 100 |

Create a view manager_new_salary based on job table to get 15% salary increase for manager.

| Job Id | Job Titile | Salary |
|---|---|---|
| 216 | Manager | 34500 |

Create a view supplier_company for showing supplier companies location details.

| Company name | City | Sreet |
|---|---|---|
| ABC | Dhaka | Mirpur |



# Procedures and function

Increase employees's salary

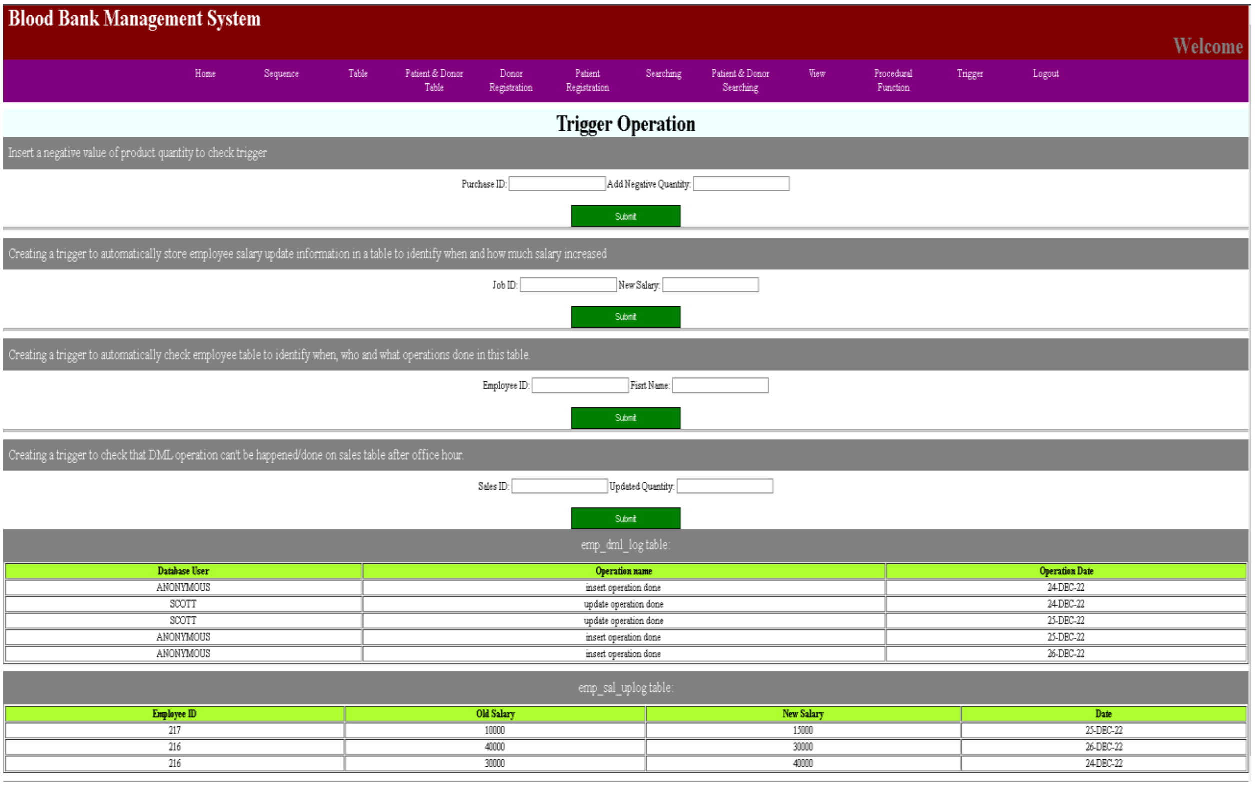
Employee ID: ______ New Salary: ______

Delete customer by giving input his / her id

Customer ID: ______

Add a sale

Sales ID: ______ Product ID: ______ Quantity ______ Employee ID: ______ Customer ID: ______ Total Amount: ______



## Trigger Operation

Insert a negative value of product quantity to check trigger

Purchase ID: [ ] Add Negative Quantity: [ ]

Submit

Creating a trigger to automatically store employee salary update information in a table to identify when and how much salary increased

Job ID: [ ] New Salary: [ ]

Submit

Creating a trigger to automatically check employee table to identify when, who and what operations done in this table.

Employee ID: [ ] First Name: [ ]

Submit

Creating a trigger to check that DML operation can't be happened/done on sales table after office hour.

Sales ID: [ ] Updated Quantity: [ ]

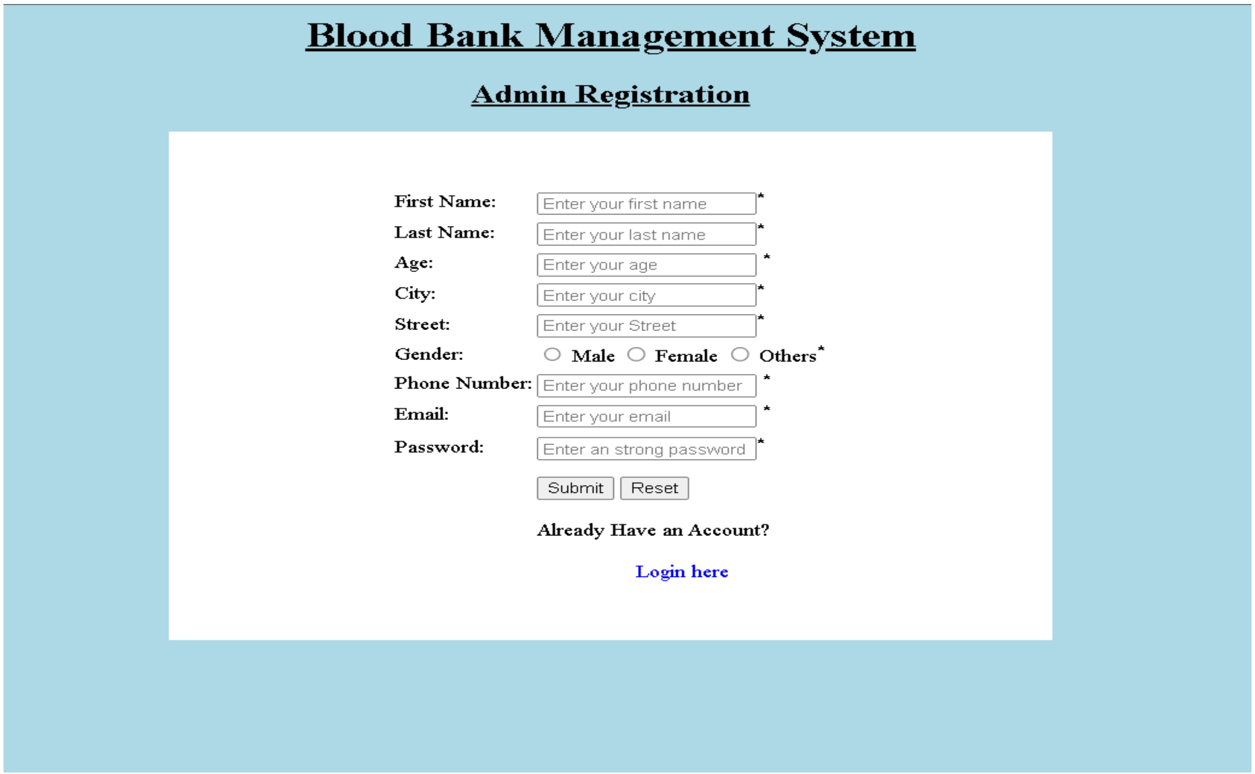
Submit

emp_dml_log table:

| Database User | Operation name | Operation Date |
|---|---|---|
| ANONYMOUS | insert operation done | 24-DEC-22 |
| SCOTT | update operation done | 24-DEC-22 |
| SCOTT | update operation done | 25-DEC-22 |
| ANONYMOUS | insert operation done | 25-DEC-22 |
| ANONYMOUS | insert operation done | 26-DEC-22 |

emp_sal_uplog table:

| Employee ID | Old Salary | New Salary | Date |
|---|---|---|---|
| 217 | 10000 | 15000 | 25-DEC-22 |
| 216 | 40000 | 30000 | 26-DEC-22 |
| 216 | 30000 | 40000 | 24-DEC-22 |



# Blood Bank Management System

## Admin Registration

First Name: [Enter your first name] *

Last Name: [Enter your last name] *

Age: [Enter your age] *

City: [Enter your city] *

Street: [Enter your Street] *

Gender: ○ Male ○ Female ○ Others *

Phone Number: [Enter your phone number] *

Email: [Enter your email] *

Password: [Enter an strong password] *

Submit Reset

Already Have an Account?

Login here

## Scenario Description:

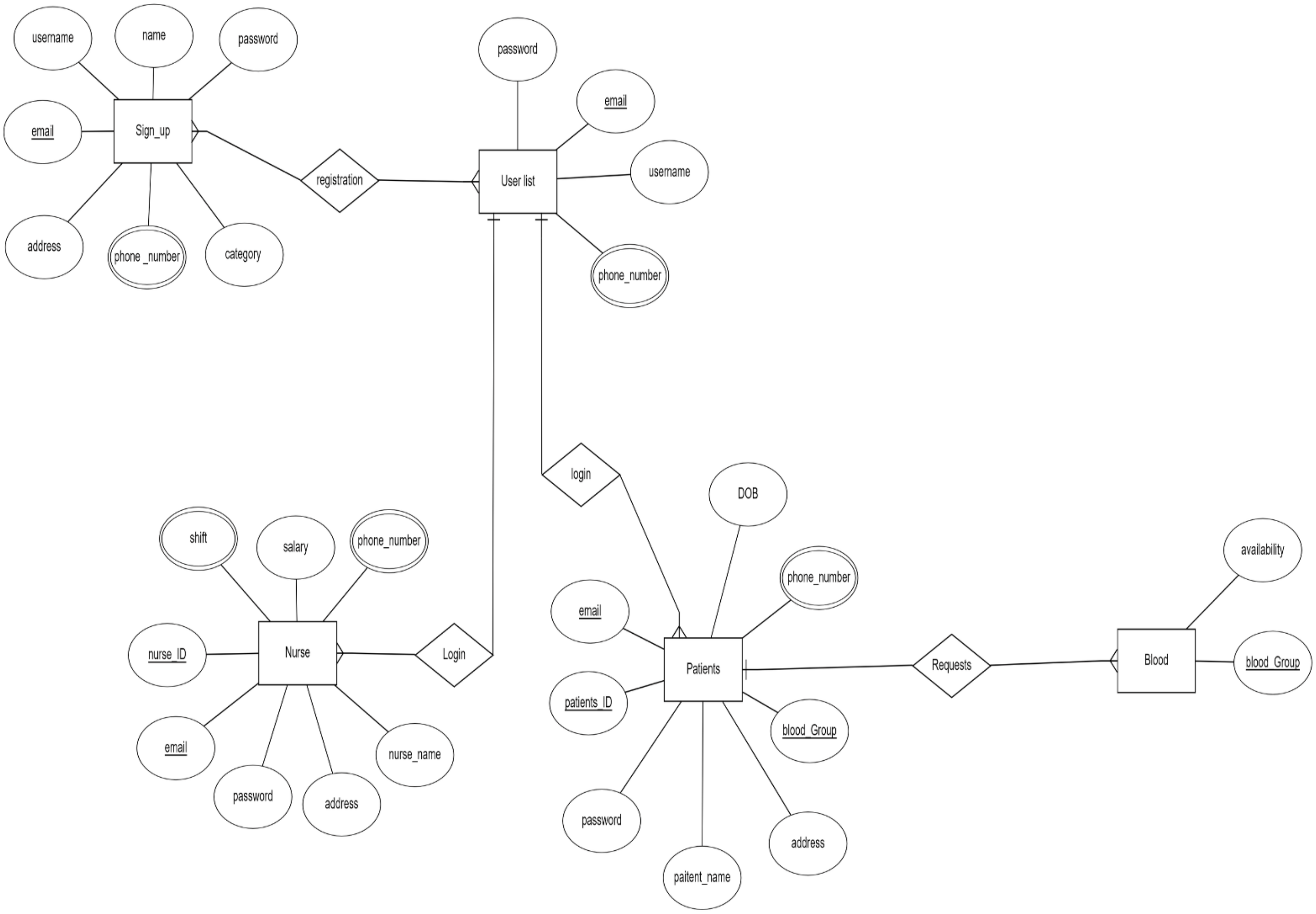
1. Blood Bank has a login system to secure authenticity. Nurse/Employee, Patients can login using unique id and password.

2. There are some nurses to manage Patients. Each Nurse have name, address, Phone number and unique id. Nurses work in two shifts day and night.

3. Patient can request and Donate blood. Before donating Patients address, name, blood group, phone and date of birth is recorded.

4. Blood is stored in storage. For that Blood group and amount is recorded.

## ER Diagram:


username
name
password
email
Sign_up
address
phone _number
category
registration
password
email
User list
username
phone_number
login
DOB
shift
salary
phone_number
availability
phone_number
email
nurse_ID
Nurse
Login
Patients
Requests
Blood
blood_Group
patients_ID
email
blood_Group
nurse_name
password
address
password
address
paitent_name

## Normallzation:

UNF

Login (username, email, password, phone_number, nurse_id, nurse_name, password, location, salary, shift, phone_number)

1NF

Phone_number & shift is a multi-valued attribute.

1. username, email, password, phone_number, nurse_id, nurse_name, email, password, address, salary, shift, phone_number

2NF

1. username, email, password, phone_number

2. nurse_id, nurse_name, email, password, address, salary, shift, phone_number

3NF

1. username, email, password, phone_number

2. nurse_id, nurse_name, address, email, salary, shift

Table Creation

1. username, email, password,

2. nurse_id, nurse_name, address, email, salary

3. email, Phone_number, shift

UNF

Login (username, email, password, phone_number, patient_id, patient_name, password, address, blood_group, DOB, phone_number)

1NF

Phone_number is a multi-value attribute

1. username, email, password, phone_number, patient_id, patient_name, email, password, address, blood_group, DOB, phone_number

2NF

1. username, email, password, phone_number

2. patient_id, patient_name, email, password, address, blood_group, DOB, phone_number

3NF

1. username, email, password, phone_number

2. patient_id, patient_name, email, password, address, blood_group, DOB, phone_number

Table Creation

1. username, email, password,

2. patient_id, patient_name, address, email, blood_group, DOB

3. email, Phone_number

UNF

Requests (patient_id, patient_name, password, address, blood_group, DOB, phone_number, blood_group, availability)

1NF

Phone_number is a multi-value attribute

1. patient_id, patient_name, password, address, blood_group, DOB, phone_number, blood_group, availability

2NF

1. patient_id, patient_name, password, address, blood_group, DOB, phone_number

2. blood_group, availability

3NF

1. patient_id, patient_name, password, address, blood_group, DOB, phone_number

2. blood_group, availability

Table Creation

1. patient_id, patient_name, password, address, blood_group, DOB, phone_number

2. blood_group, availability

UNF

Registration (username, name, password, address, phone_number, email, category, password, email, username, phone_number)

1NF

Phone_number is a multi-value attribute

1. username, name, password, address, phone_number, email, category, password, email, username, phone_number

2NF

1. username, name, password, address, phone_number, email, category

2. password, email, username, phone_number

3NF

1. username, name, password, address, phone_number, email, category

2. password, email, username, phone_number

Table Creation

1. username, name, password, address, email, category
2. email, phone_number

Temporary Table

1. username, email, password,
2. nurse_id, nurse_name, address, email, salary
3. email, Phone_number, shift
4. username, email, password
5. patient_id, patient_name, address, email, blood_group, DOB
6. email, Phone_number
7. patient_id, patient_name, password, address, blood_group, DOB, phone_number
8. blood_group, availability
9. username, name, password, address, email, category
10. email, phone_number

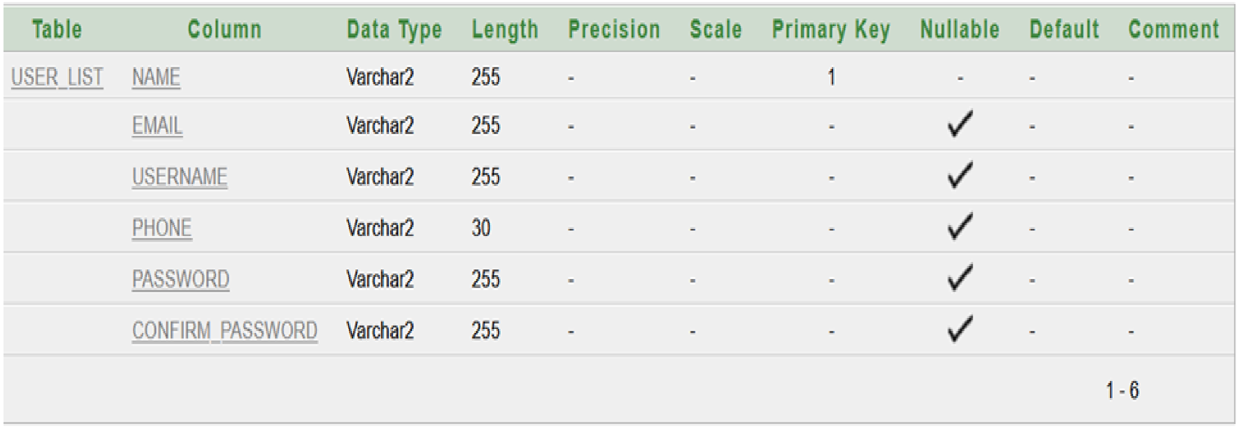
Final Table

1. username, name, password, address, email, category
2. nurse_id, nurse_name, address, email, salary
3. patient_id, patient_name, address, email, blood_group, DOB
4. email, Phone_number, shift
5. blood_group, availability

## Table Creation & Data Insertion:

# 1.Describe User_list:

CREATE TABLE User_list(Name varchar2(255) primary key,Email varchar2(255),UserName varchar2(255),Phone varchar2(30),Password varchar2(255),Confirm_Password varchar2(255));



| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| USER_LIST | NAME | Varchar2 | 255 | - | - | 1 | - | - | - |
| | EMAIL | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | USERNAME | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | PHONE | Varchar2 | 30 | - | - | - | ✓ | - | - |
| | PASSWORD | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | CONFIRM_PASSWORD | Varchar2 | 255 | - | - | - | ✓ | - | - |

1 - 6

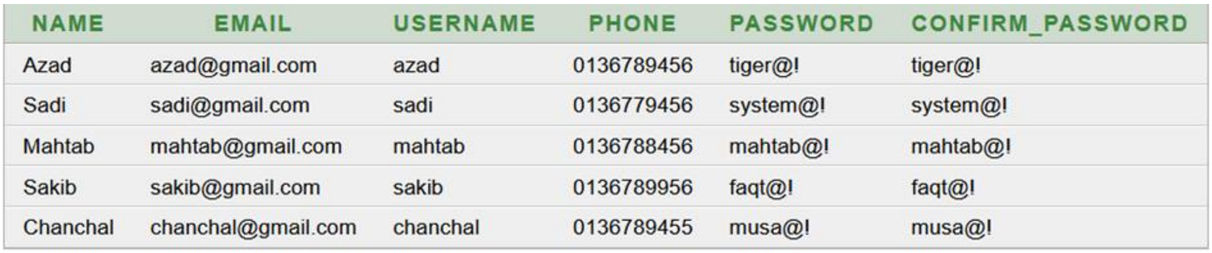
INSERT INTO User_list (Name,Email,UserName,Phone,Password,Confirm_Password)VALUES ('Azad', 'azad@gmail.com', 'azad', '0136789456', 'tiger@!', 'tiger@!');

INSERT INTO User_list (Name,Email,UserName,Phone,Password,Confirm_Password)VALUES ('Sadi', 'sadi@gmail.com', 'sadi', '0136779456', 'system@!', 'system@!');

INSERT INTO User_list (Name,Email,UserName,Phone,Password,Confirm_Password)VALUES ('Mahtab', 'mahtab@gmail.com', 'mahtab', '0136788456', 'mahtab@!', 'mahtab@!');

```
INSERT INTO User_list
(Name,Email,UserName,Phone,Password,Confirm_Password)VALUES
('Sakib', 'sakib@gmail.com', 'sakib', '0136789956', 'faqt@!', 'faqt@!');
```

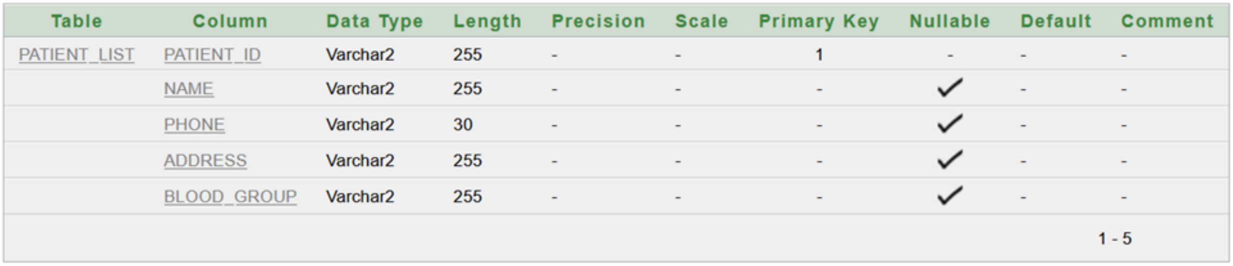
```
INSERT INTO User_list
(Name,Email,UserName,Phone,Password,Confirm_Password)VALUES
('Chanchal', 'chanchal@gmail.com', 'chanchal', '0136789455', 'musa@!',
'musa@!');
```



| NAME | EMAIL | USERNAME | PHONE | PASSWORD | CONFIRM_PASSWORD |
|---|---|---|---|---|---|
| Azad | azad@gmail.com | azad | 0136789456 | tiger@! | tiger@! |
| Sadi | sadi@gmail.com | sadi | 0136779456 | system@! | system@! |
| Mahtab | mahtab@gmail.com | mahtab | 0136788456 | mahtab@! | mahtab@! |
| Sakib | sakib@gmail.com | sakib | 0136789956 | faqt@! | faqt@! |
| Chanchal | chanchal@gmail.com | chanchal | 0136789455 | musa@! | musa@! |

## 2. Describe Patient_list:

```
CREATE TABLE Patient_list(Patient_ID varchar2(255) primary key,Name
varchar2(255),Phone varchar2(30),Address varchar2(255),Blood_Group
varchar2(255));
```

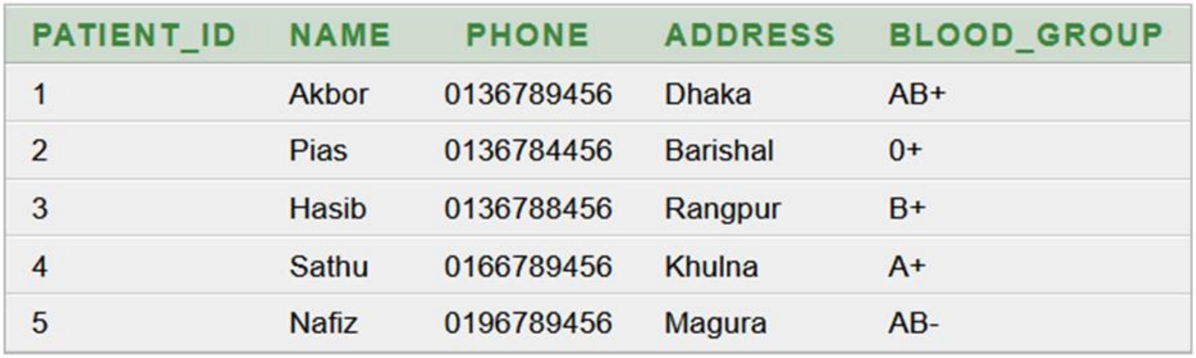


| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| PATIENT_LIST | PATIENT_ID | Varchar2 | 255 | - | - | 1 | - | - | - |
| | NAME | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | PHONE | Varchar2 | 30 | - | - | - | ✓ | - | - |
| | ADDRESS | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | BLOOD_GROUP | Varchar2 | 255 | - | - | - | ✓ | - | - |

1 - 5

```
INSERT INTO
Patient_list(Patient_ID,Name,Phone,Address,Blood_Group)VALUES ('1',
'Akbor', '0136789456', 'Dhaka', 'AB+');
```

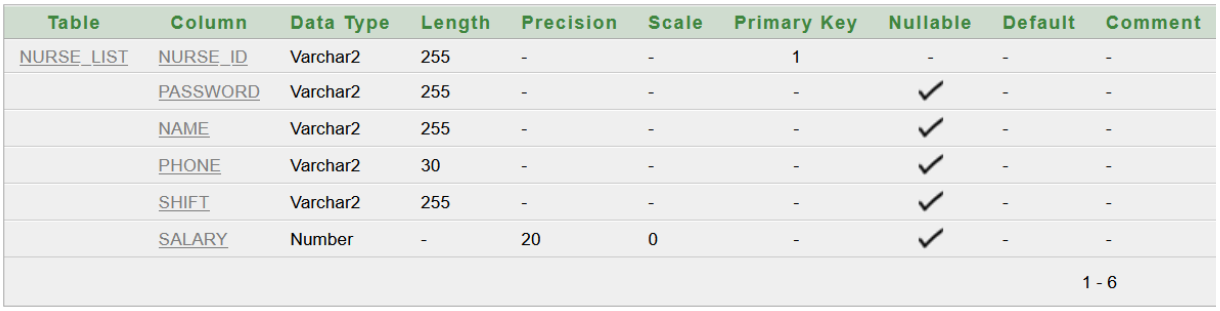
```
INSERT INTO Patient_list(Patient_ID,Name,Phone,Address,Blood_Group)VALUES ('2', 'Pias', '0136784456', 'Barishal', '0+');

INSERT INTO Patient_list(Patient_ID,Name,Phone,Address,Blood_Group)VALUES ('3', 'Hasib', '0136788456', 'Rangpur', 'B+');

INSERT INTO Patient_list(Patient_ID,Name,Phone,Address,Blood_Group)VALUES ('4', 'Sathu', '0166789456', 'Khulna', 'A+');

INSERT INTO Patient_list(Patient_ID,Name,Phone,Address,Blood_Group)VALUES ('5', 'Nafiz', '0196789456', 'Magura', 'AB-');
```



| PATIENT_ID | NAME | PHONE | ADDRESS | BLOOD_GROUP |
|---|---|---|---|---|
| 1 | Akbor | 0136789456 | Dhaka | AB+ |
| 2 | Pias | 0136784456 | Barishal | 0+ |
| 3 | Hasib | 0136788456 | Rangpur | B+ |
| 4 | Sathu | 0166789456 | Khulna | A+ |
| 5 | Nafiz | 0196789456 | Magura | AB- |

## 3. Describe Nurse_list:

```
CREATE TABLE Nurse_list(Nurse_ID varchar2(255) primary key,Password varchar2(255),Name varchar2(255),Phone varchar2(30),Shift varchar2(255),Salary number(20));
```



| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| NURSE_LIST | NURSE_ID | Varchar2 | 255 | - | - | 1 | - | - | - |
| | PASSWORD | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | NAME | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | PHONE | Varchar2 | 30 | - | - | - | ✓ | - | - |
| | SHIFT | Varchar2 | 255 | - | - | - | ✓ | - | - |
| | SALARY | Number | - | 20 | 0 | - | ✓ | - | - |

1 - 6

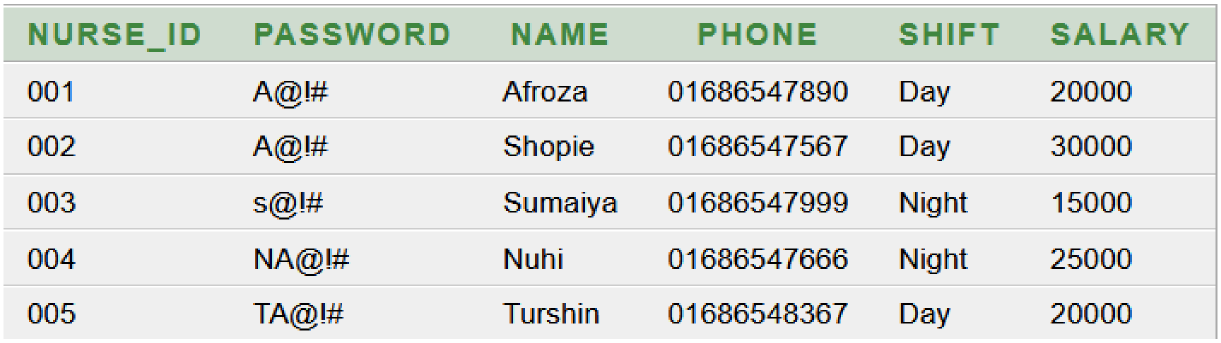
INSERT INTO Nurse_list(Nurse_ID ,Password ,Name ,Phone ,Shift,Salary )VALUES ('001', 'A@!#','Afroza', '01686547890', 'Day', '20000');

INSERT INTO Nurse_list(Nurse_ID ,Password ,Name ,Phone ,Shift,Salary )VALUES ('002', 'A@!#','Shopie', '01686547567', 'Day', '30000');

INSERT INTO Nurse_list(Nurse_ID ,Password ,Name ,Phone ,Shift,Salary )VALUES ('003', 's@!#','Sumaiya', '01686547999', 'Night', '15000');

INSERT INTO Nurse_list(Nurse_ID ,Password ,Name ,Phone ,Shift,Salary )VALUES ('004', 'NA@!#','Nuhi', '01686547666', 'Night', '25000');

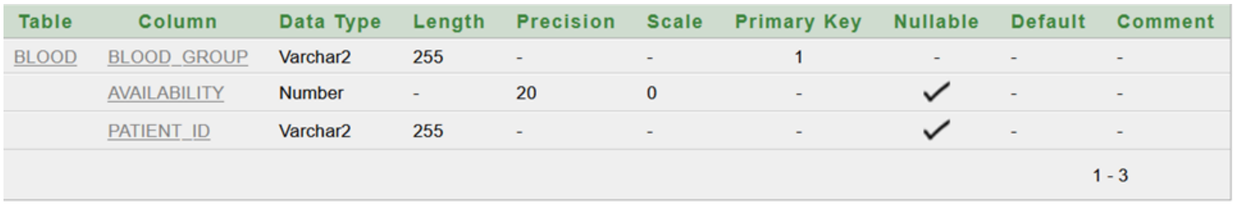
INSERT INTO Nurse_list(Nurse_ID ,Password ,Name ,Phone ,Shift,Salary )VALUES ('005', 'TA@!#','Turshin', '01686548367', 'Day', '20000');



| NURSE_ID | PASSWORD | NAME | PHONE | SHIFT | SALARY |
|---|---|---|---|---|---|
| 001 | A@!# | Afroza | 01686547890 | Day | 20000 |
| 002 | A@!# | Shopie | 01686547567 | Day | 30000 |
| 003 | s@!# | Sumaiya | 01686547999 | Night | 15000 |
| 004 | NA@!# | Nuhi | 01686547666 | Night | 25000 |
| 005 | TA@!# | Turshin | 01686548367 | Day | 20000 |

## 4.Describe Blood:

CREATE TABLE Blood(Blood_Group varchar2(255) primary key,Availability number(20),Patient_id varchar(255));



| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|---|---|---|---|---|---|---|---|---|---|
| BLOOD | BLOOD_GROUP | Varchar2 | 255 | - | - | 1 | - | - | - |
| | AVAILABILITY | Number | - | 20 | 0 | - | ✓ | - | - |
| | PATIENT_ID | Varchar2 | 255 | - | - | - | ✓ | - | - |

1 - 3

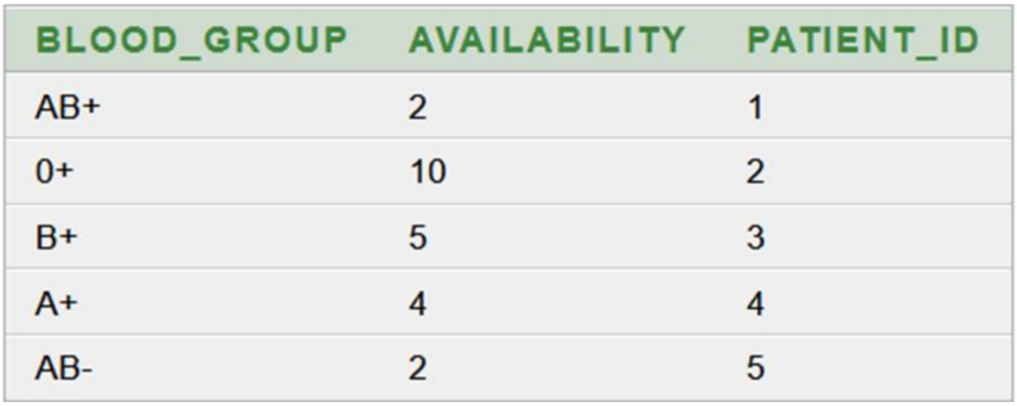
INSERT INTO Blood(Blood_Group ,Availability ,Patient_id)VALUES ('AB+', '2','1');

INSERT INTO Blood(Blood_Group ,Availability ,Patient_id)VALUES ('0+', '10','2');

INSERT INTO Blood(Blood_Group ,Availability ,Patient_id)VALUES ('B+', '5','3');

INSERT INTO Blood(Blood_Group ,Availability ,Patient_id)VALUES ('A+', '4','4');

INSERT INTO Blood(Blood_Group ,Availability ,Patient_id)VALUES ('AB-', '2','5');



| BLOOD_GROUP | AVAILABILITY | PATIENT_ID |
|---|---|---|
| AB+ | 2 | 1 |
| 0+ | 10 | 2 |
| B+ | 5 | 3 |
| A+ | 4 | 4 |
| AB- | 2 | 5 |

## Query Writing:

# **SQL:**

## **Single row:**

1. display the name and salary for Nurse_id '004'

    - select name, salary  FROM Nurse_list WHERE Nurse_ID='004';

## **Group Function:**

1. Display patient name and average blood group and group them by names

    - select patient_name,avg(blood_group) from patient_list group by patient_name;

## **Subquery:**

1. Display nurse name, salary whose salary is greater than Afroza salary from nurse _list table

    - select nurse_name ,nurse_salary from nurse_list where nurse_salary<(select nurse_salary from nurse_list where nurse_name ='Afroza');

## **View:**

1. create a view patientid that contains the details of patients with id greater than 2

- create view patientid as select patient_name, patient_id from patient_list where patient_id>2

## **Join:**

1. Joining the patient address from patient_list table and nurse name from nurse_list using EQUIJOIN as patient_list and nurse_list has direct relation.

- SELECT patient.patient_list, nurse.nurse_list FROM patient_list patient, nurse_list nurse WHERE patient.patient_id = nurse.nurse_id;

## **Synonym:**

1. Create a synonym that contains the details of patient name for the view of patient details.

- create synonym patientdetails for patient_name;

## **PL/SQL:**

### **Function:**

1. Create a function that returns the total number of patient in the patient list table.

-
  ```
  CREATE FUNCTION totalPatient
  RETURN number AS
  total number(2) := 0;
  ```

```
BEGIN
SELECT count(*) into total
FROM patient_list;
RETURN total;
END;
/
```

**Calling a Function:**

```
DECLARE
c number(2);
BEGIN
c := totalPatient();
dbms_output.put_line('Total no. of Patient: ' || c);
END;
/
```

## Trigger:

2. Create a trigger in such a way that whenever a row is deleted from the patient_list table an output ‘A Patient is Deleted’ is generated.

-
  ```
  CREATE TRIGGER patient_d
  after delete ON Patient_list
  FOR EACH ROW
  BEGIN
  dbms_output.put_line('A Patient is Deleted');
  END;
  /

  select * from Patient_list;
  DELETE FROM Patient_list WHERE patient_name = 'Nafiz';
  Rollback
  ```

## **Procedure:**

1. create a procedure to update the value of the column of shift from day to night in the nurse_list table.

-
  ```
  CREATE PROCEDURE adjust_shift(in_shift IN nurse_list.shift%TYPE)
  IS
  BEGIN
  UPDATE nurse_shift
  SET shift ='Day'
  WHERE shift= in_shift;
  END;

  begin
  adjust_shift('night');
  end

  select * from nurse_list;
  rollback
  ```

## **Record:**

1. Create a record that can output the name, phone and shift of the nurse whose id is 004 in the nurse_list table.

-
  ```
  declare
  dept_nurse list nurse%rowtype;

  begin
  select * into dept_nurse list from nurse
  ```

```
where nurse_id=004;
dbms_output.put_line('Nurse Name: ' || dept_nurse
list.nurse_name);
dbms_output.put_line('Nurse Phone: ' || dept_nurse
list.phone);
dbms_output.put_line('Nurse Shift: ' || dept_nurse
list.shift);
end
/
```

## Cursor:

1. Create a cursor that can output 1st patient’s name on the patient table.

-
  ```
  declare
  patientname Patient_list.patient_name%type;
  cursor patient_patient is
  select patient_name from Patient_list;

  begin
  open patient_patient;
  fetch patient_patient into patientname;
  dbms_output.put_line('1st Patient Name: ' || patientname);
  close patient_patient;
  end
  /
  ```

## Conclusion:

Technology is introducing new innovations day by day, thus reducing the time required to do things. The proposed system can be used to reduce the time required to deliver required blood to the needy in cases of emergency. The web application provides a way of communication and synchronization between the hospitals and the blood banks. It also provides them with the facility of communicating with the nearby donors in emergency. The database is a vital aspect of the system. The database of the hospitals and the blood banks must be checked for consistency on regular basis for smooth working of the system.