SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

DATABASE MANAGEMENT SYSTEM PROJECT

BLOOD BANK MANAGEMENT SYSTEM.

Submitted to: Sindhu.S

PROJECT TEAM

1.Garagapati Prudhvija – RA2011042010126

2.Harshitha.T-RA2011042010140

3. phaneendra – RA2011042010143

AIM

The aim is to provide blood donation service to the city recently. Blood Bank Management System (BBMS) is a Web-based application that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank.

This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way.

Project Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle-free and corruption-free and make the system of blood bank management effective.

Blood Bank donation system can collect blood from many donators in short from various sources and distribute that blood to needy people who require blood.

PROPOSED WORK DETAILS

The proposed work consists of 6 tables that are interconnected. The team members work ontables and keep updating them by implementing queries.

DDL Statements:

In the context of SQL, data definition or data description language (DDL) is a syntaxfor creating and modifying database objects such as tables, indices, and users.

- CREATE to create a new table or database.
- ALTER for alteration.
- Truncate to delete data from the table.
- DROP to drop a table
- RENAME to rename a table.

1.Donor Table :-

It contains the details of donor along with ID,Name,age,sex,blood_type and phone number.Here,Donor_Id is the Primary key.

```
SQL> create table Donor(
  2  Donor_Id int,
  3  Name varchar(25),
  4  Age int,
  5  Sex varchar(10),
  6  Blood_Type varchar(5),
  7  Phonenumber int
  8  );
```

Table created.

2.DonatesBloodRecepient Table:-

It contains Details of the blood like cost,BloodAmoun,Blood_type,Bloodbag_number and Donor_Id .

```
SQL>  create table DonatesBloodRecepient(
  2  cost int,
  3  BloodAmount int,
  4  Blood_Type varchar(5),
  5  Bloodbag_number int,
  6  Donor_Id int
  7  );
```

Table created.

3. Recepient Table:-

It contains Recepient details like Name,Age,sex,Blood_Type.

```
SQL> create table Recepient(
  2  Name varchar(15),
  3  Age int,
  4  sex varchar(10),
  5  Blood_Type varchar(5)
  6  );
```

Table created.

4.Registers Table:-

It contains Details like Donor_Id and Emp_Id.

```
SQL> create table Registers(
  2  Donor_Id int,
  3  Emp_Id int
  4  );
```

Table created.

5. Staff Table:-

It contains details of staff like their Emp_Id,Name,address,Phone number , Salary.Here,Emp_id is primary key.

```
SQL> create table Staff(
  2  Emp_Id int,
  3  Name varchar(15),
  4  Address varchar(10),
  5  PhoneNumber int,
  6  Salary int
  7  );
```

Table created.

6. Bloodinventorymanages Table:

It contains details like quantity of Blood,blood_bag number and blood_Type.

```
SQL> create table Bloodinventorymanages(
  2  quantity int,
  3  Bloodbag_Number int,
  4  Blood_Type varchar(5)
  5  );
```

Table created.

```
SQL> Alter table Bloodinventorymanages
  2  add Orders int;
```

Table altered.

DML Statements

A data manipulation language (DML) is a computer programming language used foradding (inserting), deleting, and modifying (updating) data in a database.

INSERT – is used to insert data into a table.

UPDATE – is used to update existing data within a table.

DELETE – is used to delete records from a database table.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
  2  values(101,'Priyansh',23,'Male','B+',8452310932);
```

1 row created.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
  2  values(102,'Siri',22,'Female','O+',9923157832);
```

1 row created.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
  2  values(103,'Pavani',28,'Female','A-',8045692378);
```

1 row created.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
```

```
  2  values(104,'Raki',31,'Male','A+',9932567819);
```

1 row created.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
  2  values(105,'Krish',25,'Male','B-',9566164872);
```

1 row created.

```
SQL> insert into Donor(Donor_Id,Name,Age,Sex,Blood_Type,Phonenumber)
  2  values(106,'Kumari',30,'Female','AB-',8074001727);
```

1 row created.

```
SQL> select * from Donor;
```

```
  DONOR_ID NAME                           AGE SEX        BLOOD PHONENUMBER
---------- ------------------------ ---------- ---------- ----- -----------
       101 Priyansh                         23 Male       B+     8452310932
       102 Siri                             22 Female     O+     9923157832
       103 Pavani                           28 Female     A-     8045692378
       104 Raki                             31 Male       A+     9932567819
       105 Krish                            25 Male       B-     9566164872
       106 Kumari                           30 Female     AB-    8074001727

6 rows selected.
```

```
SQL> insert into
DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Do
nor_Id)
  2  values(850,250,'A-',021,103);
```

1 row created.

```
SQL> insert into
DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Do
nor_Id)
  2  values(1225,500,'AB-',043,106);
```

1 row created.

```
SQL> insert into
DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Do
nor_Id)
  2  values(850,250,'B+',051,101);
```

1 row created.

```
SQL> insert into
DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Do
nor_Id)
  2  values(1700,750,'O+',053,102);
```

1 row created.

```
SQL> insert into
DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Do
nor_Id)
  2  values(850,250,'B-',032,105);
```

1 row created.

SQL> insert into DonatesBloodRecepient(cost,BloodAmount,Blood_Type,Bloodbag_number,Donor_Id)

2 values(1700,750,'A+',071,104);

1 row created.

SQL> select * from DonatesBloodRecepient**;**

COST	BLOODAMOUNT	BLOOD	BLOODBAG_NUMBER	DONOR_ID
850	250	A-	21	103
1225	500	AB-	43	106
850	250	B+	51	101
1700	750	O+	53	102
850	250	B-	32	105
1700	750	A+	71	104

6 rows selected.

SQL> insert into Recepient(Name,Age,sex,Blood_Type)

2 values('Siva',24,'Male','B+');

1 row created.

SQL> insert into Recepient(Name,Age,sex,Blood_Type)

2 values('Prince',28,'Male','O+');

1 row created.

SQL> insert into Recepient(Name,Age,sex,Blood_Type)

2 values('Aishu',31,'Female','AB-');

1 row created.

SQL> insert into Recepient(Name,Age,sex,Blood_Type)

```
  2  values('Pavitra',26,'Female','A+');
```

1 row created.

```
SQL> insert into Recepient(Name,Age,sex,Blood_Type)
  2  values('Lucky',21,'Female','B-');
```

1 row created.

```
SQL> insert into Recepient(Name,Age,sex,Blood_Type)
  2  values('Vaishnavi',30,'Female','A-');
```

1 row created.

```
SQL> select * from Recepient;

NAME                    AGE SEX        BLOOD
--------------- ---------- ---------- -----
Siva                     24 Male       B+
Prince                   28 Male       O+
Aishu                    31 Female     AB-
Pavitra                  26 Female     A+
Lucky                    21 Female     B-
Vaishnavi                30 Female     A-

6 rows selected.

SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(104,19911);
```

1 row created.

```
SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(101,19912);
```

1 row created.

```
SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(106,19913);
```

1 row created.

```
SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(102,19914);
```

1 row created.

```
SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(105,19915);
```

1 row created.

```
SQL> insert into Registers(Donor_Id,Emp_Id)
  2  values(103,19916);
```

1 row created.

SQL> select * from Registers;

```
  DONOR_ID     EMP_ID
---------- ----------
       104      19911
       101      19912
       106      19913
       102      19914
       105      19915
       103      19916

6 rows selected.
```

SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19911,'Saranya','Vizag',8885523416,40000);

1 row created.
SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19912,'Ishaa','Orissa',9943567821,35000);

1 row created.

SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19913,'Sreenivas','Chennai',8321456986,42000);

1 row created.

SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19914,'Priyanka','Bangalore',9321599932,21000);

1 row created.

SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19915,'Reddy','Mumbai',9912345694,41050);

1 row created.

SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2 values(19916,'Sonika','Delhi',9123876124,27000);

1 row created.

SQL> select * from Staff;

```
    EMP_ID NAME            ADDRESS    PHONENUMBER     SALARY
---------- --------------- ---------- ----------- ----------
     19911 Saranya         Vizag       8885523416      40000
     19912 Ishaa           Orissa      9943567821      35000
     19913 Sreenivas       Chennai     8321456986      42000
     19914 Priyanka        Bangalore   9321599932      21000
     19915 Reddy           Mumbai      9912345694      41050
     19916 Sonika          Delhi       9123876124      27000

6 rows selected.
```

SQL> insert into Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(750,071,'A+');

1 row created.

SQL> insert into Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(250,021,'A-');

1 row created.

SQL> insert into Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(250,051,'B+');

1 row created.

SQL> insert into Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(250,032,'B+');

1 row created.

SQL> insert into Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(750,053,'O+');

1 row created.

SQL> insert into

Bloodinventorymanages(quantity,Bloodbag_Number,Blood_Type)
  2 values(500,043,'AB-');

1 row created.

SQL> select * from Bloodinventorymanages;

```
  QUANTITY BLOODBAG_NUMBER BLOOD     ORDERS
---------- --------------- ----- ----------
       750              71 A+
       250              21 A-
       250              51 B+
       250              32 B+
       750              53 O+
       500              43 AB-

6 rows selected.
```

SQL> insert into Bloodinventorymanages(Orders)
  2 values(2);

1 row created.

SQL> insert into Bloodinventorymanages(Orders)
  2 values(3);

1 row created.

SQL> insert into Bloodinventorymanages(Orders)
  2 values(1);

1 row created.

SQL> insert into Bloodinventorymanages(Orders)
  2 values(2);

1 row created.

SQL> insert into Bloodinventorymanages(Orders)
  2 values(2);

1 row created.

SQL> insert into Bloodinventorymanages(Orders)

2 values(2);

1 row created.

```
SQL> select * from Bloodinventorymanages;

  QUANTITY BLOODBAG_NUMBER BLOOD     ORDERS
---------- --------------- ----- ----------
       750              71 A+
       250              21 A-
       250              51 B+
       250              32 B+
       750              53 O+
       500              43 AB-
                                         2
                                         3
                                         1
                                         2
                                         2

  QUANTITY BLOODBAG_NUMBER BLOOD     ORDERS
---------- --------------- ----- ----------
                                         2

12 rows selected.
```

SQL> delete from Bloodinventorymanages
2 where orders=2;

4 rows deleted.

SQL> delete from Bloodinventorymanages
2 where orders=3;

1 row deleted.

SQL> delete from Bloodinventorymanages
2 where orders=1;

1 row deleted.

SQL> select * from Bloodinventorymanages;

```
  QUANTITY BLOODBAG_NUMBER BLOOD     ORDERS
---------- --------------- ----- ----------
       750              71 A+
       250              21 A-
       250              51 B+
       250              32 B+
       750              53 O+
       500              43 AB-

6 rows selected.
```

SQL> commit;

Commit complete.

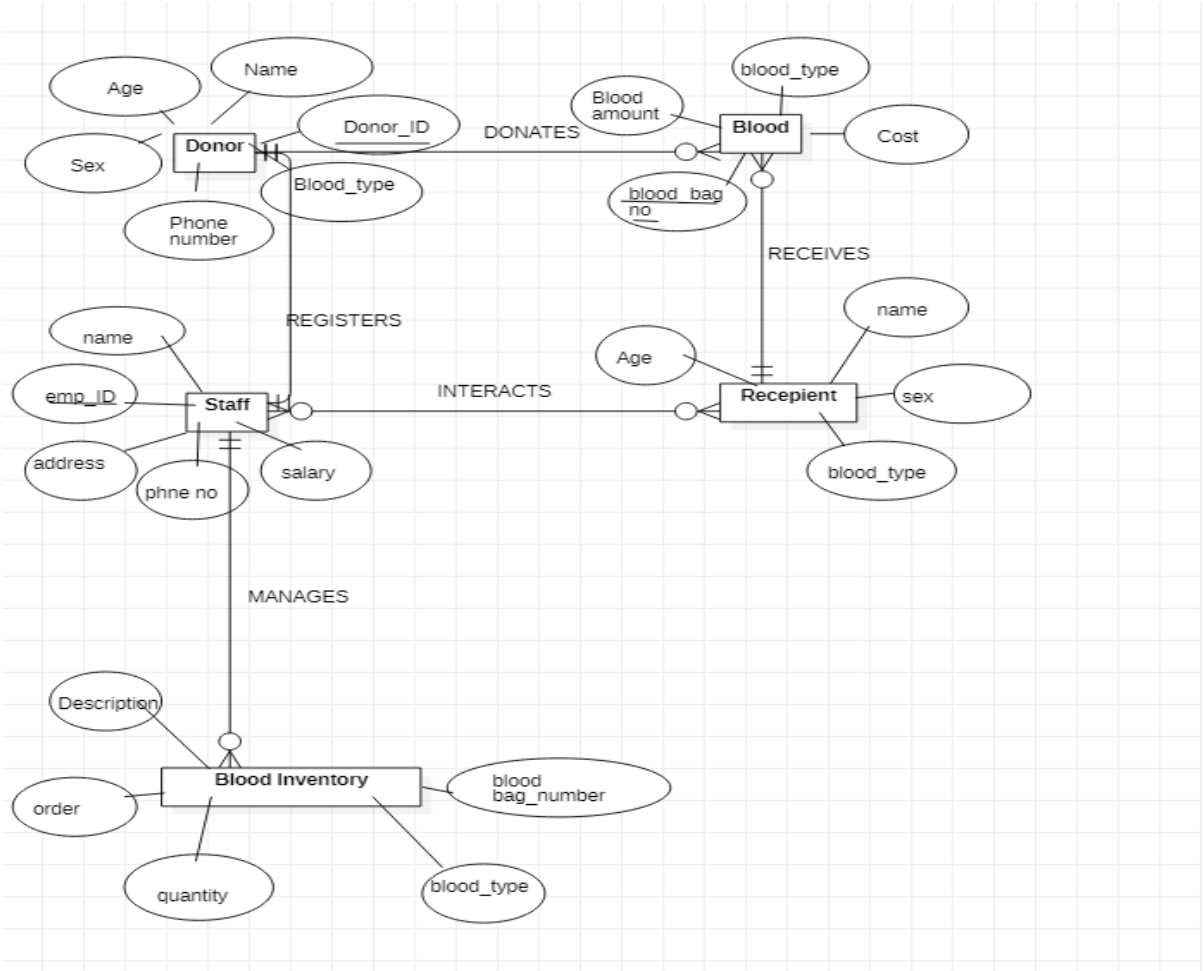
INBUILT FUNCTIONS:

A built-in function is a function that is already available in a programming language, application, or another tool that can be accessed by end users.

```
SQL> select sum(quantity) from Bloodinventorymanages;

SUM(QUANTITY)
-------------
         2750

SQL> select avg(salary) from Staff;

AVG(SALARY)
-----------
 34341.6667

SQL> select max(cost) from DonatesBloodRecepient;

 MAX(COST)
----------
      1700
```

```
SQL> select current_timestamp from Donor;

CURRENT_TIMESTAMP
---------------------------------------------------------------------------
13-JUN-22 08.50.22.305000 PM +05:30
13-JUN-22 08.50.22.305000 PM +05:30
13-JUN-22 08.50.22.305000 PM +05:30
13-JUN-22 08.50.22.305000 PM +05:30
13-JUN-22 08.50.22.305000 PM +05:30
13-JUN-22 08.50.22.305000 PM +05:30

6 rows selected.

SQL> select min(cost) from DonatesBloodRecepient;

 MIN(COST)
----------
       850
```

ER-Diagram:-



JOINS:-

A joins clause is used to combine rows from two or more tables, based on a related column between them.

Different Types of SQL Joins:

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table.

NATURAL JOIN: Returns all records based on the common columns in the two tables being joined.

SELF JOIN: A table is joined with itself.

```
SQL> select Donor.Blood_Type
  2  from Donor
  3  LEFT JOIN Recepient ON Donor.Blood_Type=Recepient.Blood_Type;

BLOOD
-----
B+
O+
AB-
A+
B-
A-

6 rows selected.
```

```
SQL> select DonatesBloodRecepient.cost,Donor.Name
  2  from DonatesBloodRecepient
  3   RIGHT JOIN Donor ON DonatesBloodRecepient.Donor_Id=Donor.Donor_Id;

      COST NAME
---------- -------------------------
       850 Pavani
      1225 Kumari
       850 Priyansh
      1700 Siri
       850 Krish
      1700 Raki

6 rows selected.
```

```
SQL> select Registers.Donor_Id,Staff.Name,Staff.Salary
  2  from Registers
  3  FULL JOIN Staff ON Registers.Emp_Id=Staff.Emp_Id;

  DONOR_ID NAME                SALARY
---------- --------------- ----------
       104 Saranya              40000
       101 Ishaa                35000
       106 Sreenivas            42000
       102 Priyanka             21000
       105 Reddy                41050
       103 Sonika               27000

6 rows selected.
```

```
SQL> select DonatesBloodRecepient.Blood_Type,DonatesBloodRecepient.Bloodbag_Number,Bloodinventorymanages.quantity
  2  from DonatesBloodRecepient
  3  INNER JOIN Bloodinventorymanages ON DonatesBloodRecepient.BloodAmount=Bloodinventorymanages.quantity;

BLOOD BLOODBAG_NUMBER   QUANTITY
----- --------------- ----------
O+                53        750
A+                71        750
A-                21        250
B+                51        250
B-                32        250
A-                21        250
B+                51        250
B-                32        250
A-                21        250
B+                51        250
B-                32        250

BLOOD BLOODBAG_NUMBER   QUANTITY
----- --------------- ----------
O+                53        750
A+                71        750
AB-               43        500
```

```
SQL> select Donor.Donor_Id,Donor.Phonenumber,Recepient.Blood_Type
  2  from Donor
  3  cross join Recepient;

  DONOR_ID PHONENUMBER BLOOD
---------- ----------- -----
       101  8452310932 B+
       101  8452310932 O+
       101  8452310932 AB-
       101  8452310932 A+
       101  8452310932 B-
       101  8452310932 A-
       102  9923157832 B+
       102  9923157832 O+
       102  9923157832 AB-
       102  9923157832 A+
       102  9923157832 B-

  DONOR_ID PHONENUMBER BLOOD
---------- ----------- -----
       102  9923157832 A-
       103  8045692378 B+
       103  8045692378 O+
       103  8045692378 AB-
       103  8045692378 A+
       103  8045692378 B-
       103  8045692378 A-
       104  9932567819 B+
       104  9932567819 O+
       104  9932567819 AB-
       104  9932567819 A+
```

```
  DONOR_ID PHONENUMBER BLOOD
---------- ----------- -----
       104  9932567819 B-
       104  9932567819 A-
       105  9566164872 B+
       105  9566164872 O+
       105  9566164872 AB-
       105  9566164872 A+
       105  9566164872 B-
       105  9566164872 A-
       106  8074001727 B+
       106  8074001727 O+
       106  8074001727 AB-

  DONOR_ID PHONENUMBER BLOOD
---------- ----------- -----
       106  8074001727 A+
       106  8074001727 B-
       106  8074001727 A-

36 rows selected.
```

SUBQUERIES:-

A query within a query is known as subquery**.**

```
SQL> select Registers.Donor_Id,Registers.Emp_Id,Staff.Name,Staff.Salary
  2  from Registers,Staff where Registers.Emp_Id=Staff.Emp_Id;

  DONOR_ID     EMP_ID NAME                SALARY
---------- ---------- --------------- ----------
       104      19911 Saranya              40000
       101      19912 Ishaa                35000
       106      19913 Sreenivas            42000
       102      19914 Priyanka             21000
       105      19915 Reddy                41050
       103      19916 Sonika               27000

6 rows selected.

SQL> select Emp_Id,Name,Salary from Staff where Salary>=(select Salary from Staff where Name='Sonika');

    EMP_ID NAME                SALARY
---------- --------------- ----------
     19911 Saranya              40000
     19912 Ishaa                35000
     19913 Sreenivas            42000
     19915 Reddy                41050
     19916 Sonika               27000

SQL> select Name, Age from Recepient where Sex in (select Sex from Recepient where Sex='Female');

NAME                    AGE
--------------- ----------
Aishu                   31
Pavitra                 26
Lucky                   21
Vaishnavi               30
```

```
SQL> Select Name,Age,Sex,Blood_Type from Donor where age=(select min(age) from Donor );

NAME                            AGE SEX        BLOOD
------------------------- ---------- ---------- -----
Siri                            22 Female     O+

SQL> select Donor_ID,Blood_Type from DonatesBloodRecepient where BloodAmount = any(select BloodAmount from DonatesBloodRecepient where BloodAmount>250);

  DONOR_ID BLOOD
---------- -----
       106 AB-
       102 O+
       104 A+
```

PL/SQL PROGRAMS:

PROCEDURES-

```
SQL> CREATE OR REPLACE PROCEDURE today_is AS
  2  BEGIN
  3  DBMS_OUTPUT.PUT_LINE( 'Today is ' || TO_CHAR(SYSDATE, 'DL') );
  4  END today_is;
  5  /

Procedure created.

SQL> BEGIN
  2    today_is();
  3  END;
  4  /

PL/SQL procedure successfully completed.
Today is Tuesday,June 14, 2022
```

FUNCTIONS:

```
SQL> create or replace function
  2  count_rows (Donor in varchar2)
  3  return number
  4  as
  5    l_count number;
  6  begin
  7   execute immediate
  8    'select count(*)
  9     from '|| Donor
 10   into l_count;
 11
 12     return l_count;
 13  end;
 14  /

Function created.
SQL> select count(*) as numberofdonors
  2  from Donor;

NUMBEROFDONORS
--------------
             6
```

TRIGGERS:

```
SQL>  CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE DELETE OR INSERT OR UPDATE ON Staff
  3  FOR EACH ROW
  4  WHEN (NEW.Emp_Id>0)
  5  DECLARE
  6     sal_diff number;
  7  BEGIN
  8     sal_diff := :NEW.salary  - :OLD.salary;
  9     dbms_output.put_line('Old salary: ' || :OLD.salary);
 10     dbms_output.put_line('New salary: ' || :NEW.salary);
 11     dbms_output.put_line('Salary difference: ' || sal_diff);
 12  END;
 13  /

Trigger created.
SQL> insert into Staff(Emp_Id,Name,Address,PhoneNumber,Salary)
  2  values(19919,'Lohi','Delhi',9912376231,41900);

1 row created.
Old salary:
New salary: 41900
Salary difference:
```

IMPLICIT CURSORS:-

```
SQL> DECLARE
  2  total_rows number(2);
  3  BEGIN
  4
  5  UPDATE Staff
  6  SET salary = salary + 500;
  7  IF sql%notfound THEN
  8  dbms_output.put_line('no Staff selected');
  9  ELSIF sql%found THEN
 10    total_rows := sql%rowcount;
 11  dbms_output.put_line( total_rows: || ' Staff selected');
 12  END IF;
 13   END;
 14  /

PL/SQL procedure successfully completed.
Statement Processed
total_rows: 7 Staff selected
```

EXPLICIT CURSORS:-

```
SQL> DECLARE
  2  c_Name Recepient.Name%type;
  3  c_Age Recepient.Age%type;
  4  c_Sex Recepient.Sex%type;
  5  CURSOR c_Recepient is
  6     SELECT Name,Age,Sex FROM Recepient;
  7     BEGIN
  8  OPEN c_Recepient;
  9  LOOP
 10  FETCH c_Recepient into c_Name,c_Age,c_Sex;
 11     EXIT WHEN c_Recepient%notfound;
 12    dbms_output.put_line(c_Name || ' ' || c_Age || ' ' || c_Sex);
 13  END LOOP;
 14  CLOSE c_Recepient;
 15  END;
 16  /

PL/SQL procedure successfully completed.
Statement Processed
NAME                   AGE SEX
-------------- ---------- ----------
Siva                    24 Male
Prince                  28 Male
Aishu                   31 Female
Pavitra                 26 Female
Lucky                   21 Female
Vaishnavi               30 Female
```

SET OPERATORS:-

```
SQL> select Blood_Type from Donor
  2  union all
  3  select Blood_Type from Recepient;

BLOOD
-----
B+
O+
A-
A+
B-
AB-
B+
O+
AB-
A+
B-

BLOOD
-----
A-

SQL> select Blood_Type from Donor
  2  union
  3  select Blood_Type from Recepient;

BLOOD
-----
A+
A-
AB-
B+
B-
O+
```

```
SQL> select Emp_Id from Registers
  2  intersect
  3  select Emp_Id from Staff;

    EMP_ID
----------
     19911
     19912
     19913
     19914
     19915
     19916

6 rows selected.


SQL> select Emp_Id from Staff
  2  minus
  3  select Emp_Id from Registers;

    EMP_ID
----------
     19919
1 row selected.
```

VIEWS:-

```
SQL> create view donordetails as
  2  select Name,Age,Blood_Type
  3  from Donor;

View created.

SQL> select * from donordetails;

NAME                            AGE BLOOD
------------------------- ---------- -----
Priyansh                         23 B+
Siri                             22 O+
Pavani                           28 A-
Raki                             31 A+
Krish                            25 B-
Kumari                           30 AB-

6 rows selected.

SQL> update donordetails
  2  SET Name='Surya',Age=29
  3  where Blood_Type='O+';

1 row updated.

SQL> select * from donordetails;

NAME                            AGE BLOOD
------------------------- ---------- -----
Priyansh                         23 B+
Surya                            29 O+
Pavani                           28 A-
Raki                             31 A+
Krish                            25 B-
Kumari                           30 AB-
```

```
SQL> delete from donordetails
  2  where Name='Raki';

1 row deleted.

SQL> select * from donordetails;

NAME                            AGE BLOOD
------------------------- ---------- -----
Priyansh                         23 B+
Surya                            29 O+
Pavani                           28 A-
Krish                            25 B-
Kumari                           30 AB-

SQL> drop view donordetails;

View dropped.

SQL> select * from donordetails;
select * from donordetails
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>spool off;
```

RESULTS:-

Successfully implemented Blood bank management system Database with all the necessary sql queries.