**DBMS LAB PROJECT REPORT**

ON

**“BLOOD DONATION DATABASE MANAGEMENT SYSTEM”**

A PROJECT Report

Submitted in Partial Fulfillment of the Requirements

For the award of the Degree of

**Bachelor of Technology**

**in**

**Electronics & Computer Engineering (ECM)**

By

**BANDA ATHMIYA VARSHINI (21311A1937)**



**Department of Electronics & Computer Engineering**

**Sreenidhi Institute of Science & Technology (Autonomous)**

**2022-2023**

**DEPARTMENTOFELECTRONICS&COMPUTERENGINEERING**

**SREENIDHIINSTITUTEOFSCIENCE&TECHNOLOGY**

**(AUTONOMOUS)**



**CERTIFICATE**

This is to certify that DBMS lab project report entitled **“BLOOD DONATION DATABASEMANAGEMENT SYSTEM”** submitted by **BANDA ATHMIYA VARSHINI,** bearing roll no’s 21311A1937.Towards partial fullfilment for the award of Bachelors of Degree in Electronics & Computer Engineering from SreenidhiInstitueOf Science And Technology, Ghatkesar, Hyderabad is a record of Bonafide work done by my team, during academic year 2022-2023 under the guidance and the evaluation of faculty.

**MRS. K. KUSUMALATHA DR.D.MOHAN**

Assistant Professor HOD, ECM



**ACKNOWLEGEMENT**

I wish to extend my sincere gratitude to my lecturer**Mrs. K.KUSUMALATHA**,Department of Electronics & Computer Engineering. For her valuable guidance andencouragement which has been absolutely helpful in successful Completion of the Seminar Report/Document.

It is needed gratifying to have the privilege to express my deep sense of gratitude andappreciation to **Dr. D. MOHAN,** Head of ECM Department,forgiving me this opportunity.



**DECLARATION**

This is to certify that the DBMS Lab Project Report titled “**BLOOD DONATION DATABASE MANAGEMENT SYSTEM”** is a record work done by us in the Department of Electronics and Computer Engineering (ECM), Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad.

The report is based on the project work done entirely by us and not copied from any other source.

**BANDA ATHMIYA VARSHINI (21311A1937)**

**ABSTRACT**

The Blood Donation Database Management System is a comprehensive software solution designed to streamline and optimize the process of managing blood donation information, it aims to provide a centralized and secure platform for blood banks,hospitals,and other healthcare organizations to efficiently manage their blood donation records, donors, and inventory. The system ensures the availability of a steady and reliable blood supply, ultimately saving lives.

The Key features of the Blood Donation Database Management System include:

1. **Donor registration and Profile Management:The System allows potential donors to register their Personal information, medical history, and blood type. Donors can create profiles and update their details as needed, ensuring accurate and up-to-date records.**
2. **Blood Inventory Tracking:The system maintains a comprehensive inventory of available blood units, categorized by blood type, Rh factor and expiration dates. This real-time tracking prevents shortages and wastage of blood donation.**
3. **Donor Eligibility and Medical Screening: The system includes a medical screening process to assess donor eligibility based on predefined criteria and guidelines. This step ensures the safety of both donors and recipients.**
4. **Donor Rewards and Incentives: The system may offer a reward and incentives program to encourage regular donors and maintain a steady blood supply.**
5. **Hospitals and Healthcare Organizations: The system can integrate with hospitals and healthcare facilities to facilitate seamless blood transfusion request and updates, enabling rapid responses in emergencies.**
6. **Data Security and Privacy: The system incorporates robust security measures to protect sensitive donor information and comply with data privacy regulations.**

**The Blood Donation Management System is a user-friendly, scalable,and adaptable solution that addresses the challenges faced by blood banks and healthcare institutions in managing their blood donation processes. By promoting efficient management, accurate tracking, and streamlined operations, the system plays a vital role in supporting blood donation initiatives and saving lives in times of need.**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE-NO** |
| **01** | **INTRODUCTION** |  |
| **02** | **E-R DIAGRAM** |  |
| **03** | **INFORMATION OF ENTITIES** |  |
| **04** | **INFORMATION OF ATTRIBUTES** |  |
| **05** | **DATABASE SCHEMA** |  |
| **06** | **CREATION OF DATA** |  |
| **07** | **INSERTION OF DATA** |  |
| **08** | **QUERIES AND OUTPUTS** |  |
| **09** | **QUERIES AND OUTPUTS** |  |

**INTRODUCTION**

**Blood banks collect, store and provide collected blood to the patients who are in need of blood. The people who donate blood are called ‘donors. The banks then group the blood which they receive according to the blood groups. They also make sure that the blood is not contaminated. The main mission of the blood bank is to provide the blood to the hospitals and health care systems which saves the patient’s life. No hospital can maintain the health care system without pure and adequate blood. The major concern each blood bank has is to monitor the quality of the blood and monitor the people who donates the blood, that is ‘donors. But this a tough job. The existing system will not satisfy the need of maintaining quality blood and keep track of donors. To overcome all these Limitations we introduced a new system called ‘Blood Donation Management System’.**

The ‘Blood Bank Management System’ allows us to keep track of quality of blood and also

keeps track of available blood when requested by the acceptor. The existing systems are

Manual systems which are time consuming and not so effective. ‘Blood Bank Management

system’ automates the distribution of blood. This database consists of thousands of records

of each blood bank.

By using this system searching the available blood becomes easy and saves lot of time than

the manual system. It will hoard, operate, recover and analyse information concerned with

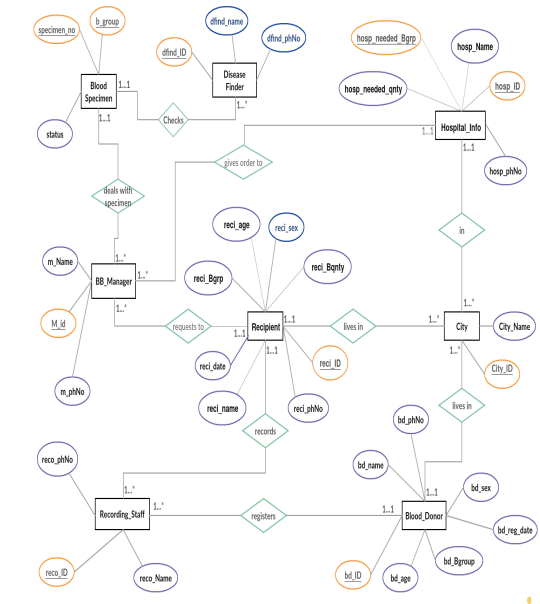
the administrative and inventory management within a blood bank. This system is developed

in a manner that it is manageable, time effective, cost effective, flexible and much man power

is not required.

**The ‘Blood Bank Management System’ allows us to keep track of quality of blood and also keeps track ofavailable blood when requested the acceptor. The existing systems are Manual systems which are time consuming and not so effective. ‘Blood Bank Management system’ automates the distribution of blood. This database consists of thousands of records of each blood bank. By using this system searching the available blood becomes easy and saves lot of time than the manual system. It will hoard, operate, recover and analyse information concerned with the administrative and inventory management within a blood bank. This system is developed in a manner that it is manageable, time effective, cost effective, flexible and much man power is not required.**

**E-R DIAGRAM**

****

**INFORMATION OF ENTITIES**

An entity is an “object” in the real world that is distinguishable from all other objects. An entity is a set of entities of the same type that share the same attributes.

The entities we have used in the Blood Bank Management System are:

1. Blood\_Donor Entity:

The donor is the person who donates blood.

1. Recipent Entity:

The Recipient is the person who receives blood from blood bank.

1. Blood bank\_manager Entity:

The blood bank manager is the person who takes care of the available blood samples in the blood bank, he is also responsible for handling blood requests from recipients and hospitals.

1. Recording staff Entity:

The recording staff is a person who registers the blood donor and recipients.

1. Blood Specimen Entity:

In data base, under Blood Specimen entity we will store the information of blood samples which are available in the blood bank.

1. Disease Finder Entity:

In data base, under DiseaseFinder entity we will store the information of the doctor who checks the blood for any kind of contaminations.

1. Hospital\_Info Entity:

In the data base, under Hospital\_Info entity we will store the information of hospitals.

1. City Entity:

This entity will store the information of cities where donors, recipients and hospitals are present.

**INFORMATION OF ATTRIBUTES**

1. The Attributes of Blood Donor Entity are - bd\_ID, bd\_name, bd\_sex, bd\_age, bd\_Bgroup, bd\_reg\_date, bd\_phNo.

On donation a donor id (bd\_ID) is generated and used as primary key to identify the donor information. Other than that name, age, sex, blood group, phone number and registration dates will be stored in database under Blood\_Donor.

1. The Attributes of Recipient Entity are - reci\_ID, reci\_name, reci\_age, reci\_Bgrp, reci\_Bqnty ,reci\_sex, reci\_reg\_date, reci\_phNo.

When blood is given to a recipient a rericipient ID (reci\_ID) is generated and used as primary key for the recipient entity to indentify blood recipients information. Along with it name,age, sex, blood group (needed), blood quantity(needed), phone number, and registration dates are also stored in this data base.

1. The Attributes of BB Manager Entity are m\_ID,m\_name,m\_Phno.

Blood manager has a unique indentfication number (m\_ID) used as primary key along with name and phone number of blood bank manager will be stored in data base under BB\_Manager entity.

1. The Attributes of Recording Staff Entity are reco\_ID, reco\_Name, reco\_phNo.

It has reco\_ID which is primary key along with recoder’s name and recodrer’s phone number will also be stored in the data base under Recording\_Staff entity.

1. The Attributes of Blood Specimen Entity are specimen\_number, b\_group, status.

In this entity specimen\_number and b\_group together will be primary key along with status attribute which will show if the blood is contaminated on not.

1. The Attributes of DiseaseFinder Entity are dfind\_ID, dfind\_name, dfind\_PhNo.

Here, we have unique identification number (dfind\_ID) as primary key. Along with name and phone number of the doctor will also be stored under same entitity.

1. The Attributes of Hospital\_info Entity are hosp\_ID, hosp\_name, hosp\_needed\_Bgrp, hosp\_needed\_Bqnty.

In this hosp\_ID and hosp\_needed\_Bgrptoether makes the primary key. We will store hospital name and the blood quantity reqiured at the hospital.

1. The Attributes of city Entity are city\_ID, city\_name.

A unique identification number (City\_ID) will be used as primary key to indentify the information about the city. Along with ID city names will also be stored under this entity

**DATABASE SCHEMA**

**1.Donor Table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| bd\_id | Blood Donor’s Id | Number |
| bd\_Name | Blood Donor’s Name | varchar |
| bd\_age | Blood Donor’s Age | Number |
| bd\_sex | Blood Donor’s Sex | char |
| bd\_bgrp | Blood Donor’s blood group | varchar |
| bd\_regdate | Registration Date of Donor | date |
| reco\_id | Id of Recording Staff | number |
| city\_id | City Id | Number |

The relationship with Recording staff and Donor is 1 to many. That’s why primary key of Recording staff is used as a foreign key in Donor. The relationship with City and Donor is 1 to many. That’s why primary key of City is used as a foreign key in Donor.

**2.Recipient table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| reci\_id | Recipient’s Id | Number |
| reci\_Name | Recipient’s Name | varchar |
| reci\_age | Recipient’s age | number |
| reci\_sex | Recipient’s sex | char |
| reci\_bgrp | Recipient’s blood group | varchar |
| reci\_bqnty | Recipient’s blood quantity | number |
| reci\_reg\_date | Recipient’s registration Id | date |
| reco\_id | Recording Staff’s Id | number |
| city\_id | City’s unique Id | number |
| M\_id | Blood Bank Manager’s Id | number |

The relationship with Recording staff and Blood Recipient is 1 to many. That’s why primary key of Recording staff is used as a foreign key in Blood Recipient. The relationship with City and Blood Recipient is 1 to many. That’s why primary key of City is used as a foreign key in Blood Recipient. The relationship with Blood Bank Manager and Blood Recipient is 1 to many. That’s why primary key of Blood Specimen is used as a foreign key in Blood Recipient.

**3.City table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| city\_id | City’s unique id | int |
| city\_name | City’s name | varchar |

The relationship between City and Recipients, Donor, Hospital info are all of 1 to many. So that’s why Primary key of City is used as a Foreign key in Recipients, Donor and Hospital info.

**4.Recording Staff Table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| reco\_id | Recording Staff’s id | number |
| reco\_name | Recording Staff’s Name | varchar |
| reco\_PhNo | RecordingStaff’s Phone number | number |

The relationship between Recording Staff and Blood Donor, Recipients are all of 1 to many. That’s why the Primary key of Recording staff is used as a foreign key in Donor and Recipient.

**5.Blood Specimen Table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| specimen\_No | Blood Sample’s unique id | number |
| b\_grp | Blood Group | varchar |
| status | Whether blood is pure or not? | number |
| M\_id | Blood Bank Manager’s id | number |
| dfind\_id | Disease Finder’s unique id | number |

The relationship with Disease finder and Blood Specimen is 1 to many. That’s why Primary key of Disease finder is used as a Foreign key in Blood Specimen. The relationship with Blood Bank manager and Blood Specimen is 1 to many. That’s why primary key of Blood Bank manager is used as a foreign key in Blood Specimen.

6.**Disease Finder Table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| dfind\_id | Disease Finder’s unique id | number |
| dfind\_name | Disease Finder’s name | varchar |
| dfind\_phNo | Disease Finder’s Phone number | number |

The relationship with Disease finder and Blood Specimen is of 1 to many. Therefore, the Primary key of Disease finder is used as a foreign key in Blood Specimen.

**7.Blood Bank Manager Table**:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| M\_id | Blood Bank Manager’s id | number |
| m\_name | Blood Bank Manager’s name | varchar |
| M\_phno | Blood Bank Manager’s phone no | number |

The relationship between Blood Bank Manager and Blood Specimen, Recipient, Hospital info are all of 1 to many. So therefore, the Primary key of Blood Bank Manager is used as a foreign key in Blood Specimen, Recipient and Hospital info.

**8.Hospital info Table:**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Description | Type |
| hosp\_id | Hospital’s unique id | Number |
| hosp\_name | Hospital’s name | Varchar |
| hosp\_needed\_Bgrp | Blood group needed by hospital | Varchar |
| hosp\_needed\_qnty | Quantity of blood group needed | Number |
| City\_id | City’s unique id | Number |
| M\_id | Blood Bank Manger’s id | number |

The relationship with City and Hospital info is 1 to many. That’s why Primary key of City is used as a foreign key in Hospital info. The relationship with Blood Bank Manager and Hospital info is 1 to many. That’s why Primary key of Blood Bank manager is used as a foreign key in Hospital info.

**CREATION OF DATA**

**1]** CREATE TABLE BB\_Manager ( M\_id int NOT NULL, mName varchar(max) NOT NULL, m\_phNo bigint -- CONSTRAINT Mid\_pk PRIMARY KEY (M\_id) );

**2]** CREATE TABLE Blood\_Donor ( bd\_ID int NOT NULL, bd\_name varchar(max) NOT NULL, bd\_age varchar(max), bd\_sex varchar(max), bd\_Bgroup varchar(10), bd\_reg\_date date, reco\_ID int NOT NULL, City\_ID int NOT NULL -- CONSTRAINT bdID\_pk PRIMARY KEY (bd\_ID) );

**3]** CREATE TABLE BloodSpecimen ( specimen\_number int NOT NULL, b\_group varchar(10) NOT NULL, status int, dfind\_ID int NOT NULL, M\_id int NOT NULL CONSTRAINT specimenumber\_pk PRIMARY KEY (specimen\_number) );

**4]** CREATE TABLE DiseaseFinder ( dfind\_ID int NOT NULL, dfind\_name varchar(max) NOT NULL, dfind\_PhNo bigint -- CONSTRAINT dfindID\_pk PRIMARY KEY (dfind\_ID) );

**5]** CREATE TABLE Hospital\_Info\_1 ( hosp\_ID int NOT NULL, hosp\_name varchar(max) NOT NULL, City\_ID int NOT NULL, M\_id int NOT NULL primary key(hosp\_ID) -- CONSTRAINT hospID\_pk PRIMARY KEY (hosp\_ID) );

**6]** CREATE TABLE Hospital\_Info\_2 ( hosp\_ID int NOT NULL, hosp\_name varchar(max) NOT NULL, hosp\_needed\_Bgrp varchar(10), hosp\_needed\_qnty int primary key(hosp\_ID,hosp\_needed\_Bgrp) -- CONSTRAINT hospID\_pk PRIMARY KEY (hosp\_ID) );

**7]** CREATE TABLE Recipient ( reci\_ID int NOT NULL, reci\_name varchar(max) NOT NULL, reci\_age varchar(max), reci\_Brgp varchar(max), reci\_Bqnty float, reco\_ID int NOT NULL, City\_ID int NOT NULL, M\_id int NOT NULL, reci\_sex varchar(max), reci\_reg\_date date -- CONSTRAINT reciid\_pk PRIMARY KEY (reci\_id) );

**8]** CREATE TABLE Recording\_Staff ( reco\_ID int NOT NULL, reco\_Name varchar(max) NOT NULL, reco\_phNo bigint -- CONSTRAINT recoID\_pk PRIMARY KEY (reco\_ID) );

**INSERTION OF DATA**

INSERT into BB\_Manager VALUES(102,'Jack', 4693959671), (103,'Peter', 4693959601), (104,'Mark', 4693959677), (105,'Jason', 4693957671);

INSERT into BB\_Manager VALUES(106,'Steve', 4694959671), (107,'Jason', 4695959671), (108,'Stella', 4663959671), (109,'Monika', 4673959671), (110,'John', 4693859671);

INSERT into Blood\_Donor VALUES(150221,'Mark',25,'M','B+','2015-12-17',101212,1100), (160011,'Abdul',35,'F','A+','2016-11-22',101212,1100),(160101,'Smith',22,'M','O+','2016-01-04',101312,1200),(150011,'Pat',29,'M','O+','2015-07-19',101412,1300), (150021,'Shyam',42,'F','A-','2015-12-24',101412,1300),(150121,'Dan',44,'M','AB+','2015-08-28',101212,1200),(160031,'Mike',33,'F','AB-','2016-02-06',101212,1400),(160301,'Elisa',31,'F','AB+','2016-09-10',101312,1200), (160091,'Carrol',24,'M','B-','2016-10-15',101312,1500),(160401,'Mark',29,'M','O-','2016-12-17',101212,1200);

INSERT into BloodSpecimen VALUES(1001, 'B+', 1,11,101), (1002, 'O+', 1,12,102), (1003, 'AB+', 1,11,102), (1004, 'O-', 1,13,103), (1005, 'A+', 0,14,101), (1006, 'A-', 1,13,104), (1007, 'AB-', 1,15,104), (1008, 'AB-', 0,11,105), (1009, 'B+', 1,13,105), (1010, 'O+', 0,12,105), (1011, 'O+', 1,13,103), (1012, 'O-', 1,14,102), (1013, 'B-', 1,14,102), (1014, 'AB+', 0,15,101);

INSERT into City VALUES(1200,'Austin'), (1300,'Irving'), (1400,'Houston'), (1500,'Richardson');

INSERT into City VALUES(1600,'Plano'), (1700,'Frisco'), (1800,'Arlington'), (1900,'San Antonio'), (2000,'Tyler');

INSERT into DiseaseFinder VALUES(11,'Peter',4693804223), (12,'Park',4693804223), (13,'Jerry',4693804223), (14,'Mark',4693804223), (15,'Monika',4693804223);

INSERT into DiseaseFinder VALUES(16,'Ram',4693804123), (17,'Swathi',4693804223), (18,'Gautham',4693804323), (19,'Ashwin',4693804423), (20,'Yash',4693804523);

INSERT into Hospital\_Info\_1 VALUES(1,'MayoClinic',1100,101), (2,'CleavelandClinic',1200,103), (3,'NYU',1300,103);

INSERT into Hospital\_Info\_1 VALUES(4,'Baylor',1400,104), (5,'Charlton',1800,103), (6,'Greenoaks',1300,106), (7,'Forestpark',1300,102), (8,'Parkland',1200,106), (9,'Pinecreek',1500,109), (10,'WalnutHill',1700,105);

INSERT into Hospital\_Info\_2 VALUES(1,'MayoClinic','A+',20), (1,'MayoClinic','AB+',0), (1,'MayoClinic','A-',40), (1,'MayoClinic','B-',10), (1,'MayoClinic','AB-',20);

INSERT into Hospital\_Info\_2 VALUES(1,'MayoClinic','A+',20), (1,'MayoClinic','AB+',0), (1,'MayoClinic','A-',40), (1,'MayoClinic','B-',10), (1,'MayoClinic','AB-',20);

INSERT into Hospital\_Info\_2 VALUES(3,'NYU','A+',0), (3,'NYU','AB+',0), (3,'NYU','A-',0), (3,'NYU','B-',20), P a g e | 24 (3,'NYU','B+',10), (3,'NYU','AB-',0);

INSERT into Hospital\_Info\_2 VALUES(4,'Baylor','A+',10), (5,'Charlton','B+',30), (4,'Baylor','A-',40), (7,'Forestpark','B-',40), (8,'Parkland','B+',10), (9,'Pinecreek','AB-',20);

INSERT into Recipient VALUES(10001,'Mark',25,'B+',1.5,101212,1100,101,'M','2015-12-17'), (10002,'Dan',60,'A+',1,101312,1100,102,'M','2015-12-16'),

(10003,'Steve',35,'AB+',0.5,101312,1200,102,'M','2015-10-17'),(10004,'Parker',66,'B+',1,101212,1300,104,'M','2016-11-17'),(10005,'Jason',53,'B-',1,101412,1400,105,'M','2015-04-17'), (10006,'Preetham',45,'O+',1.5,101512,1500,105,'M','2015-12-17'), (10007,'Swetha',22,'AB-',1,101212,1500,101,'F','2015-05-17');

INSERT into Recipient VALUES(10008,'Swathi',25,'B+',2,101412,1300,103,'F','2015-12-14'), (10009,'Lance',30,'A+',1.5,101312,1100,104,'M','2015-02-16'),

(10010,'Marsh',25,'AB+',3.5,101212,1200,107,'M','2016-10-17');

INSERT into Recording\_Staff VALUES(101212,'Walcot',4045806553), (101312,'Henry',4045806553), (101412,'Silva',4045806553), (101512,'Adrian',4045806553), (101612,'Mark',4045806553);

INSERT into Recording\_Staff VALUES(101712,'Abdul',4045816553), (101812,'Jerry',4045826553), (101912,'Tim',4045836553), (101012,'Lekha',4044846553), (101112,'Mark',4045856553);

**SAMPLE QUERIES AND OUTPUTS**

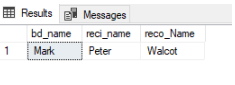
**1. Create a View of recipients and donors names having the same blood group registered on the same date.**

QUERY:

CREATEVIEWBlood\_Recipient\_SameBGrp; AS select Blood\_Donor.bd\_name,Recipient.reci\_name,reco\_Name from Recording\_Staff inner join Blood\_Donor on Recording\_Staff.reco\_ID = Blood\_Donor.reco\_ID inner join Recipient on Recording\_Staff.reco\_ID = Recipient.reco\_ID where Blood\_Donor.bd\_Bgroup = Recipient.reci\_Brgp and Blood\_Donor.bd\_reg\_date = Recipient.reci\_reg\_date

select \* from Blood\_Recipient\_SameBGrp;

OUTPUT:

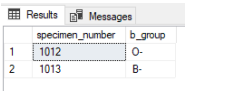


**2. Show the blood specimen verified by disease finder Mark which are pure (status=1).**

**QUERY:**

Select specimen\_number,b\_group from BloodSpecimen,DiseaseFinder WHERE BloodSpecimen.dfind\_ID= DiseaseFinder.dfind\_ID AND dfind\_name='Mark' AND status=1

OUTPUT:

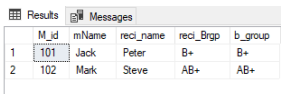


**3.Show the pure blood specimen handled by BB\_Manager who also handles a recipient needing the same blood group along with the details of the BB\_Manager and Recipient.**

**QUERY:**

SelectBB\_Manager.M\_id,mName,Recipient.reci\_name, Recipient.reci\_Brgp,b\_group from BB\_Manager,Recipient,BloodSpecimen where Recipient.M\_id = BloodSpecimen.M\_id and Recipient.reci\_Brgp = BloodSpecimen.b\_group and status = 1

OUTPUT:

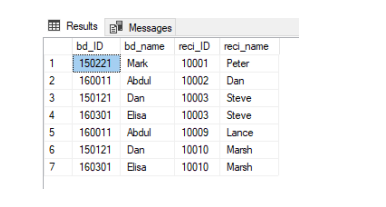


**4. Show the donors having the same blood groups required by the recipient staying in the same city along with recipient details.**

**QUERY:**

Select bd\_ID,bd\_name,reci\_ID,reci\_name FROM Blood\_Donor,Recipient WHERE bd\_Bgroup=reci\_Brgp AND Blood\_Donor.City\_ID= Recipient.City\_ID

OUTPUT:

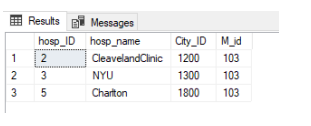


**5. Display the information of Hospital\_Info\_1 handled by BB\_Manager whose ID is 103:**

**QUERY:**

Select hosp\_ID,hosp\_name , City\_ID, HOspital\_Info\_1.M\_id from Hospital\_Info\_1,BB\_Manager where BB\_Manager.M\_id=Hospital\_Info\_1.M\_id and BB\_Manager.M\_id=103

OUTPUT:

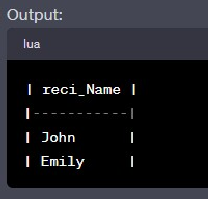


**6.Find all recipients whose age is less than 30 and blood group is 'A+'.**

**QUERY:**

SELECT reci\_Name FROM Recipient WHERE reci\_age < 30 AND reci\_bgrp = 'A+';

**OUTPUT:**

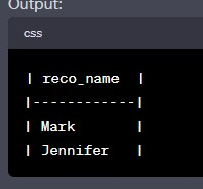


**7. Find the recording staff members who have recorded more than 100 donors.**

**QUERY:**

SELECT reco\_name FROM Recording\_Staff WHERE reco\_id IN ( SELECT reco\_id FROM Donor

GROUP BY reco\_id HAVING COUNT(\*) > 100

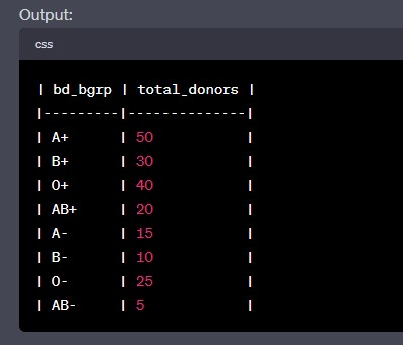
);

**8.Count the number of donors in each blood group.**

**QUERY:**

SELECT bd\_bgrp, COUNT(\*) AS total\_donors FROM Donor

GROUP BY bd\_bgrp;

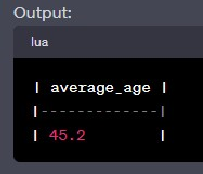


1. **Calculate the average age of recipients.**

**QUERY:**

SELECT AVG(reci\_age) AS average\_age FROM Recipient;

**OUTPUT:**

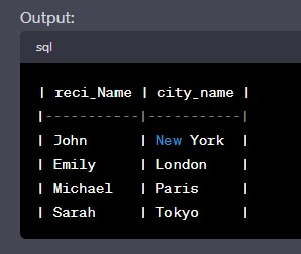


**10. Retrieve the names of recipients along with the city they belong to.**

**QUERY:**

` SELECT r.reci\_Name, c.city\_name FROM Recipient r

Inner JOIN City c ON r.city\_id = c.city\_id;

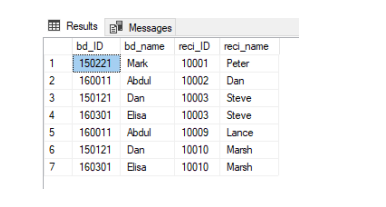


**11.Show the donors having the same blood groups required by the recipient staying in the same city along with recipient details.**

**QUERY:**

Select bd\_ID,bd\_name,reci\_ID,reci\_name FROM Blood\_Donor,Recipient WHERE bd\_Bgroup=reci\_Brgp AND Blood\_Donor.City\_ID= Recipient.City\_ID

**OUTPUT:**



**CONCLUSION**

Our project well addressed the limitations of the existing system. We designed well organized database management system which is a challenging job in this era. We have built a database for a Blood Bank using Microsoft SQL Server. Before implementing the database, in the design phase, we have explored various features, operations of a blood bank to figure out required entities, attributes and the relationship among entities to make an efficient Entity Relationship Diagram (ERD). Using Microsoft SQL Server we have created the tables for our database and inserted some sample values in the tables. Finally, we have executed sample queries on our database to check its performance to retrieve useful information accurately and speedily. In conclusion, the Blood Donation Database Management System (DBMS) project is a comprehensive and efficient solution for managing blood donation records. Through the development of this system, various features and functionalities have been implemented to ensure smooth operations and effective management of blood donation data. The primary objective of the DBMS project was to create a centralized platform for storing, organizing, and retrieving information related to blood donors, recipients, and inventory. This system enables quick and accurate tracking of blood donations, facilitating the matching of donors with recipients in need. By implementing a user-friendly interface, the DBMS project ensures ease of use for both administrators and end-users. Donors can easily register and update their information, while recipients can search for compatible donors based on their specific requirements. The system also provides real-time inventory management, ensuring the availability of blood units and reducing wastage. the security of sensitive data was a top priority throughout the project. Access controls and authentication mechanisms were implemented to safeguard the privacy of donor and recipient information. Regular backups and disaster recovery plans were put in place to ensure data integrity and availability.

Overall, the Blood Donation DBMS project provides numerous benefits. It streamlines the blood donation process, enhances coordination between donors and recipients, and improves the overall efficiency of blood banks and organizations involved in managing blood supplies. With its user-friendly interface, robust features, and emphasis on data security, this DBMS project is a valuable tool for enhancing the blood donation ecosystem and saving lives.

**REFERENCE**

<https://www.w3schools.com/>

<https://sqlzoo.net/>

<https://db-engines.com/>

<https://docs.oracle.com/en/database/>

<https://dev.mysql.com/doc/>

<https://www.oracle.com/database/>

RARE BLOO

D GROUP

ACCEPTOR