BLOOD DONETION MANAGEMENT SYSTEM

Department of Computer Science Engineering (CSE)

B-Tech 3rd Semester

Database management system

Project Report

By

Trijeta Ghosh

INDEX

* Abstract 3
* Introduction 4
* Project Description 5-7
  + 1. Key points
    2. Benefit
    3. planning
* List of Entities and Attributes 8-9
* ER Diagram 10
* Creating Commands 11-17
* Conclusion 18

Abstract

The Blood Donation Management System (BDMS) represents a crucial innovation in healthcare technology aimed at optimizing the blood donation process. This project endeavors to develop a comprehensive software solution that addresses the complex requirements of blood banks, hospitals, and healthcare organizations. The BDMS encompasses key functionalities such as donor registration and management, appointment scheduling, donation screening and testing, inventory management, recipient management, and reporting and analytics. By leveraging advanced technologies and best practices in software development, the BDMS aims to streamline operations, improve efficiency, and ensure the safety and availability of blood products. Through intuitive user interfaces, robust database management, and real-time monitoring capabilities, stakeholders can make informed decisions and contribute to the noble cause of saving lives. The BDMS project is poised to revolutionize the blood donation ecosystem, making a tangible impact on healthcare outcomes and enhancing the well-being of communities worldwide.

Introduction

The Blood Donation Management System (BDMS) is a pivotal software solution designed to streamline the intricate process of blood donation, storage, and distribution. It serves as an indispensable tool for blood banks, hospitals, and healthcare organizations, empowering them to efficiently manage their blood donation operations and ensure a stable blood supply chain. Our project aims to develop a comprehensive BDMS that integrates various features and functionalities to address the complex requirements of blood donation management. Drawing inspiration from detailed descriptions provided earlier, our system will comprise key components such as donor registration and management, appointment scheduling, donation screening and testing, inventory management, recipient management, reporting, and analytics. Through intuitive user interfaces, robust database management, and real-time monitoring capabilities, the BDMS empowers stakeholders to make informed decisions, improve efficiency, and ultimately save lives. With a strong foundation built on comprehensive requirements analysis and strategic planning, we are poised to deliver a robust and impactful solution that will make a tangible difference in the lives of countless individuals.

The BDMS project revolves around the central objective of facilitating the entire blood donation lifecycle, from donor registration to blood transfusion. By leveraging advanced technologies and best practices in software development, our system aims to revolutionize the way blood donation activities are managed and coordinated. Through intuitive user interfaces, robust database management, and real-time monitoring capabilities, the BDMS empowers stakeholders to make informed decisions, improve efficiency, and ultimately save lives.

Project Description

The Blood Donation Management System (BDMS) is a comprehensive software solution designed to modernize and optimize the process of blood donation, storage, and distribution. It serves as a centralized platform for blood banks, hospitals, and healthcare organizations to manage their blood donation operations efficiently and effectively.

1. **Key Features:**

**1. Donor Management:** The BDMS facilitates the registration and management of blood donors. It stores donor information including name, contact details, blood type, and medical history. Donors can easily update their information and schedule donation appointments through the system.

**2. Appointment Scheduling:** Donors can schedule appointments for blood donation at their convenience. The system provides real-time availability and sends reminders to donors about their upcoming appointments, ensuring a steady supply of blood donations.

**3. Donation Screening and Testing:** Prior to accepting blood donations, the system conducts rigorous screening and testing procedures to ensure the safety and quality of donated blood. Donor eligibility is verified, and blood samples are tested for infectious diseases and compatibility.

**4. Inventory Management**: The BDMS tracks the inventory of blood and blood products in real-time. It monitors stock levels, expiration dates, and storage conditions to prevent wastage and ensure that an adequate supply of blood is available for transfusions.

**5. Recipient Management:** Healthcare facilities can place orders for blood products through the BDMS. The system manages recipient information, transfusion requirements, and delivery logistics to ensure timely and accurate distribution of blood products to patients in need.

**6. Reporting and Analytics:** The BDMS generates comprehensive reports and analytics on blood donation statistics, donor demographics, inventory levels, and usage trends. These insights help organizations optimize their blood donation processes and make data-driven decisions.

1. **Benefits:**

* **Improved Efficiency:** The BDMS streamlines blood donation operations, reducing manual paperwork and administrative overhead. This allows organizations to focus more on providing quality healthcare services to patients.
* **Enhanced Donor Experience:** Donors benefit from a user-friendly interface that simplifies the donation process and provides transparency on where their donated blood is being used.
* **Safety and Compliance:** The system ensures compliance with regulatory standards and best practices in blood donation management, minimizing the risk of transfusion-related complications.
* **Optimized Blood Supply:** By providing real-time visibility into blood inventory levels and demand patterns, the BDMS helps organizations optimize their blood supply chain and respond effectively to fluctuations in demand.

1. **Project Planning:**

**Requirements Gathering:**

* **Description:** Involves collecting and documenting the needs and expectations of stakeholders regarding the BDMS functionalities.
* **Importance:** Understanding the requirements ensures that the system meets the needs of blood banks, hospitals, donors, and recipients.

**System Design:**

* **Description:** Involves designing the overall structure, components, and interfaces of the BDMS based on the gathered requirements.
* **Importance:** A well-designed system ensures scalability, maintainability, and usability, leading to a more efficient and effective blood donation process.

**Development:**

* **Description:** Involves building the BDMS software according to the defined requirements and design specifications.
* **Importance**: Development transforms the conceptual design into a functional system that can be tested and deployed.

**Testing:**

* **Description:** Involves validating the functionality, performance, and security of the BDMS through various testing techniques.
* **Importance:** Testing ensures that the system works as intended, meets quality standards, and is free from defects or vulnerabilities.

**Deployment and Maintenance:**

* **Description:** Involves deploying the BDMS into production, providing user training, and maintaining the system post-deployment.
* **Importance:** Deployment ensures that the system is accessible to users, while maintenance ensures its continued functionality and performance over time.

List of Entities and Attributes

|  |  |
| --- | --- |
| Entities | Attributes |
| Donor | **DonerID**  FirstName  LastName  BloodGroup Age  Gender  ContactNumber  Email  Address |
| Doctor | **DoctorID**  FristName  LastName  Specialization  ContactNumber |
| Helper | **HelperID**  MainBranchID  FristName  LastName  Role  ContactNumber |

|  |  |
| --- | --- |
| MainBranch | **MainBranchID**  Name  Location  ContactNumber |
| SubBranch | **SubbranchID**  BranchName  Location  HelperContactNumber |

ER Diagram

contact

**Helper**

connect

include

checkup

contact

help

**MainBranch**

**SubBranch**

**Doners**

**Doctor**

Creating Commands

--crate database

CREATE DATABASE BDC

--information about blood donors

CREATE TABLE Donors (

DonorID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

BloodGroup VARCHAR(5),

Age INT,

Gender VARCHAR(10),

ContactNumber VARCHAR(15),

Email VARCHAR(100),

Address VARCHAR(255)

);

INSERT INTO Donors (DonorID, FirstName, LastName, BloodGroup, Age, Gender, ContactNumber, Email, Address)

VALUES

(1, 'Rahul', 'Gupta', 'O+', 35, 'Male', '9876543210', 'rahul@example.com', '123 Main St, New Delhi'),

(2, 'Priya', 'Sharma', 'A-', 28, 'Female', '9876543211', 'priya@example.com', '456 Park Ave, Mumbai'),

(3, 'Amit', 'Patel', 'B+', 45, 'Male', '9876543212', 'amit@example.com', '789 Elm St, Kolkata'),

(4, 'Sneha', 'Singh', 'AB+', 22, 'Female', '9876543213', 'sneha@example.com', '101 Oak St, Bangalore'),

(5, 'Ravi', 'Joshi', 'O-', 31, 'Male', '9876543214', 'ravi@example.com', '222 Pine St, Chennai'),

(6, 'Anita', 'Kumar', 'A+', 40, 'Female', '9876543215', 'anita@example.com', '333 Cedar St, Hyderabad'),

(7, 'Raj', 'Verma', 'B-', 29, 'Male', '9876543216', 'raj@example.com', '444 Maple St, Pune'),

(8, 'Neha', 'Agarwal', 'AB-', 37, 'Female', '9876543217', 'neha@example.com', '555 Birch St, Jaipur'),

(9, 'Vikas', 'Yadav', 'O+', 25, 'Male', '9876543218', 'vikas@example.com', '666 Elm St, Ahmedabad'),

(10, 'Pooja', 'Das', 'A-', 33, 'Female', '9876543219', 'pooja@example.com', '777 Oak St, Surat'),

(11, 'Rahul', 'Shah', 'B+', 27, 'Male', '9876543220', 'rahul2@example.com', '888 Maple St, Lucknow'),

(12, 'Sunita', 'Rao', 'AB+', 39, 'Female', '9876543221', 'sunita@example.com', '999 Pine St, Chandigarh'),

(13, 'Amit', 'Singh', 'O-', 32, 'Male', '9876543222', 'amit2@example.com', '123 Cedar St, Patna'),

(14, 'Divya', 'Mishra', 'A+', 36, 'Female', '9876543223', 'divya@example.com', '456 Birch St, Nagpur'),

(15, 'Manoj', 'Gandhi', 'B-', 41, 'Male', '9876543224', 'manoj@example.com', '789 Maple St, Indore'),

(16, 'Shweta', 'Thakur', 'AB-', 24, 'Female', '9876543225', 'shweta@example.com', '101 Pine St, Varanasi'),

(17, 'Rajesh', 'Khan', 'O+', 30, 'Male', '9876543226', 'rajesh@example.com', '222 Cedar St, Bhopal'),

(18, 'Preeti', 'Sharma', 'A-', 26, 'Female', '9876543227', 'preeti@example.com', '333 Birch St, Ludhiana'),

(19, 'Vivek', 'Reddy', 'B+', 38, 'Male', '9876543228', 'vivek@example.com', '444 Maple St, Coimbatore'),

(20, 'Anjali', 'Rajput', 'AB+', 23, 'Female', '9876543229', 'anjali@example.com', '555 Oak St, Amritsar');

-- Table for Main Branches

CREATE TABLE MainBranch (

MainBranchID INT PRIMARY KEY,

Name VARCHAR(100),

Location VARCHAR(255),

ContactNumber VARCHAR(15)

);

INSERT INTO MainBranch (MainBranchID, Name, Location, ContactNumber)

VALUES (1, 'LifeSaver Drive', 'Kolkata', '033-654-2747');

-- Table for Helpers (Compounders, Volunteers, etc.)

CREATE TABLE Helpers (

HelperID INT PRIMARY KEY,

MainBranchID INT,

FirstName VARCHAR(10),

LastName VARCHAR(10),

Role VARCHAR(100),

ContactNumber VARCHAR(15),

FOREIGN KEY (MainBranchID) REFERENCES MainBranch(MainBranchID)

);

INSERT INTO Helpers (HelperID, MainBranchID, FirstName, LastName, Role, ContactNumber)

VALUES

(1, 1, 'Rajesh', 'Patel', 'Compounder', '9876543210'),

(2, 1, 'Neha', 'Sharma', 'Volunteer', '9876543211'),

(3, 1, 'Ramesh', 'Singh', 'Nurse', '9876543212'),

(4, 1, 'Priya', 'Gupta', 'Volunteer', '9876543213'),

(5, 1, 'Sanjay', 'Kumar', 'Compounder', '9876543214');

SELECT \*FROM Helpers

--Doctor details

-- Creating the table for doctor details

CREATE TABLE Doctors (

DoctorID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Specialization VARCHAR(100),

ContactNumber VARCHAR(15)

);

INSERT INTO Doctors VALUES

(1, 'Dr. Manoj', 'Sharma', 'Hematologist (Blood Cancer)', '9876543220'),

(2, 'Dr. Priyanka', 'Verma', 'Endocrinologist (Diabetes)', '9876543221'),

(3, 'Dr. Rahul', 'Singh', 'Cardiovascular Hematologist', '9876543222'),

(4, 'Dr. Sneha', 'Patil', 'Hematologist (Thrombophilia)', '9876543223'),

(5, 'Dr. Rohan', 'Mehta', 'Hematologist (Hemophilia)', '9876543224'),

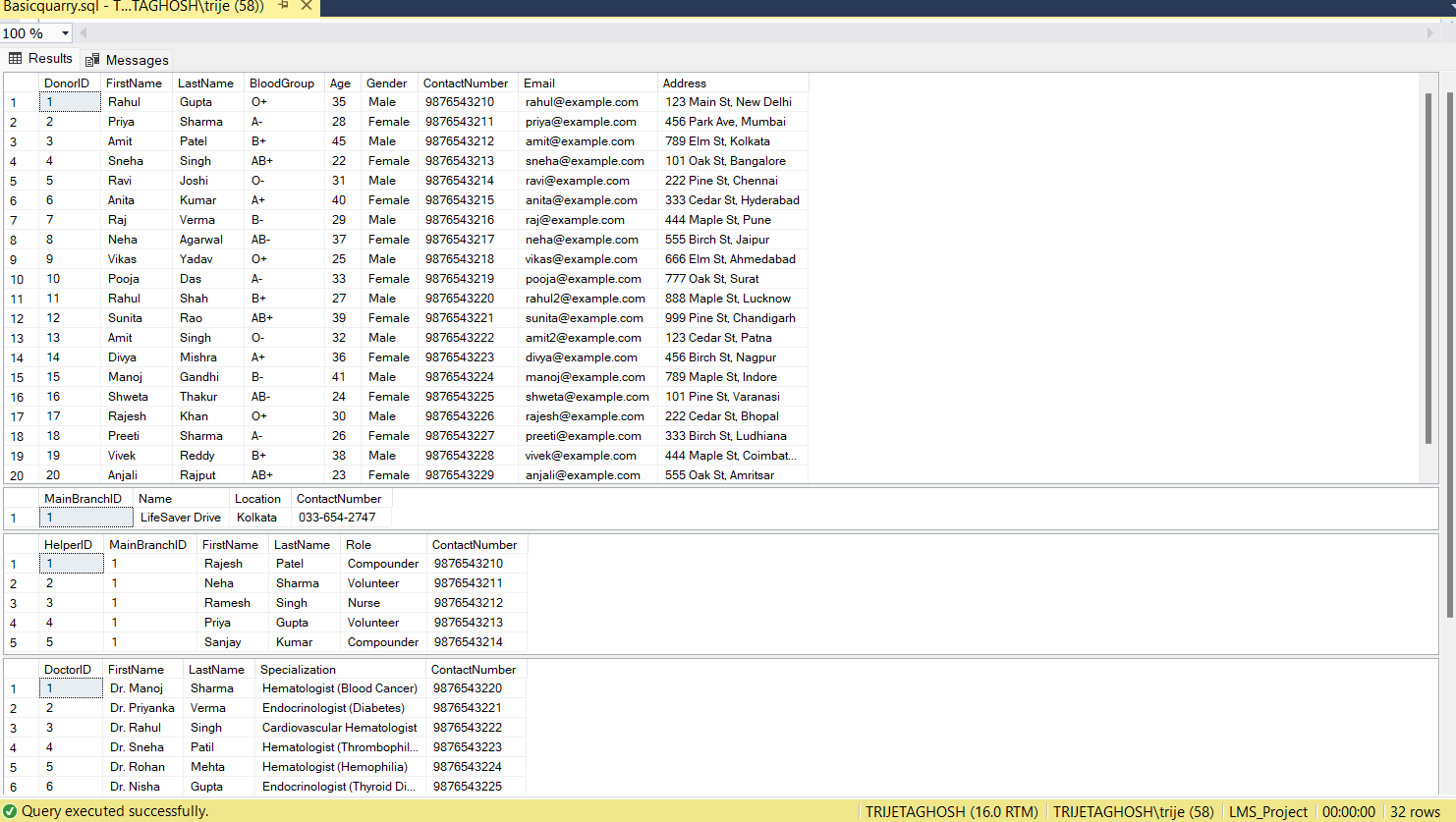
(6, 'Dr. Nisha', 'Gupta', 'Endocrinologist (Thyroid Disorders)', '9876543225');

SELECT\*FROM Donors;

SELECT\*FROM MainBranch;

SELECT \*FROM Helpers;

SELECT\*FROM Doctors;

Output

Relational Table

CREATE TABLE SubBranch (

SubBranchID INT PRIMARY KEY,

BranchName VARCHAR(100),

Location VARCHAR(255)

);

-- Inserting data into the SubBranch table

INSERT INTO SubBranch (SubBranchID, BranchName, Location)

VALUES

(1, 'LifeSaver Drive\_1', 'Howrah'),

(2, 'LifeSaver Drive\_2', 'Salt Lake City'),

(3, 'LifeSaver Drive\_3', 'New Town');

-- Add new column HelperContactNumber to Doctors\_SubBranch table

ALTER TABLE SubBranch

ADD HelperContactNumber VARCHAR(15);

UPDATE SubBranch

SET HelperContactNumber = '9876543210'

WHERE SubBranchID = 1;

UPDATE SubBranch

SET HelperContactNumber = '9876543211'

WHERE SubBranchID = 2;

UPDATE SubBranch

SET HelperContactNumber = '9876543212'

WHERE SubBranchID = 3;

-- Creating the table for doctors assigned to sub-branches

CREATE TABLE Doctors\_SubBranch (

DoctorID INT,

SubBranchID INT,

PRIMARY KEY (DoctorID, SubBranchID),

FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID),

FOREIGN KEY (SubBranchID) REFERENCES SubBranch(SubBranchID)

);

-- Inserting data into the Doctors\_SubBranch table

INSERT INTO Doctors\_SubBranch (DoctorID, SubBranchID)

VALUES

(1, 2), -- Doctor 1 assigned to SubBranch A

(2, 1), -- Doctor 2 assigned to SubBranch A

(3, 1), -- Doctor 3 assigned to SubBranch B

(4, 2), -- Doctor 4 assigned to SubBranch B

(5, 3), -- Doctor 5 assigned to SubBranch C

(6, 3); -- Doctor 6 assigned to SubBranch C

-- Creating the table for helpers assigned to doctors

CREATE TABLE Helpers\_Doctors (

HelperID INT PRIMARY KEY,

DoctorID INT,

FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)

);

-- Inserting data into the Helpers\_Doctors table

INSERT INTO Helpers\_Doctors (HelperID, DoctorID)

VALUES

(1, 2), -- Helper 1 assigned to Doctor 1

(2, 1), -- Helper 2 assigned to Doctor 2

(3, 3), -- Helper 3 assigned to Doctor 3

(4, 4), -- Helper 4 assigned to Doctor 4

(5, 5), -- Helper 5 assigned to Doctor 5

(6, 6); -- Helper 6 assigned to Doctor 6

-- Creating the Donor\_Branch table

CREATE TABLE Donor\_Branch (

DonorID INT,

SubBranchID INT,

PRIMARY KEY (DonorID, SubBranchID),

FOREIGN KEY (DonorID) REFERENCES Donors(DonorID),

FOREIGN KEY (SubBranchID) REFERENCES SubBranch(SubBranchID)

);

-- Inserting values into the Donor\_Branch table

INSERT INTO Donor\_Branch (DonorID, SubBranchID)

VALUES

(1, 1), -- Donor 1 assigned to SubBranch 1

(2, 1), -- Donor 2 assigned to SubBranch 1

(3, 1), -- Donor 3 assigned to SubBranch 1

(4, 2), -- Donor 4 assigned to SubBranch 2

(5, 2), -- Donor 5 assigned to SubBranch 2

(6, 2), -- Donor 6 assigned to SubBranch 2

(7, 3), -- Donor 7 assigned to SubBranch 3

(8, 3), -- Donor 8 assigned to SubBranch 3

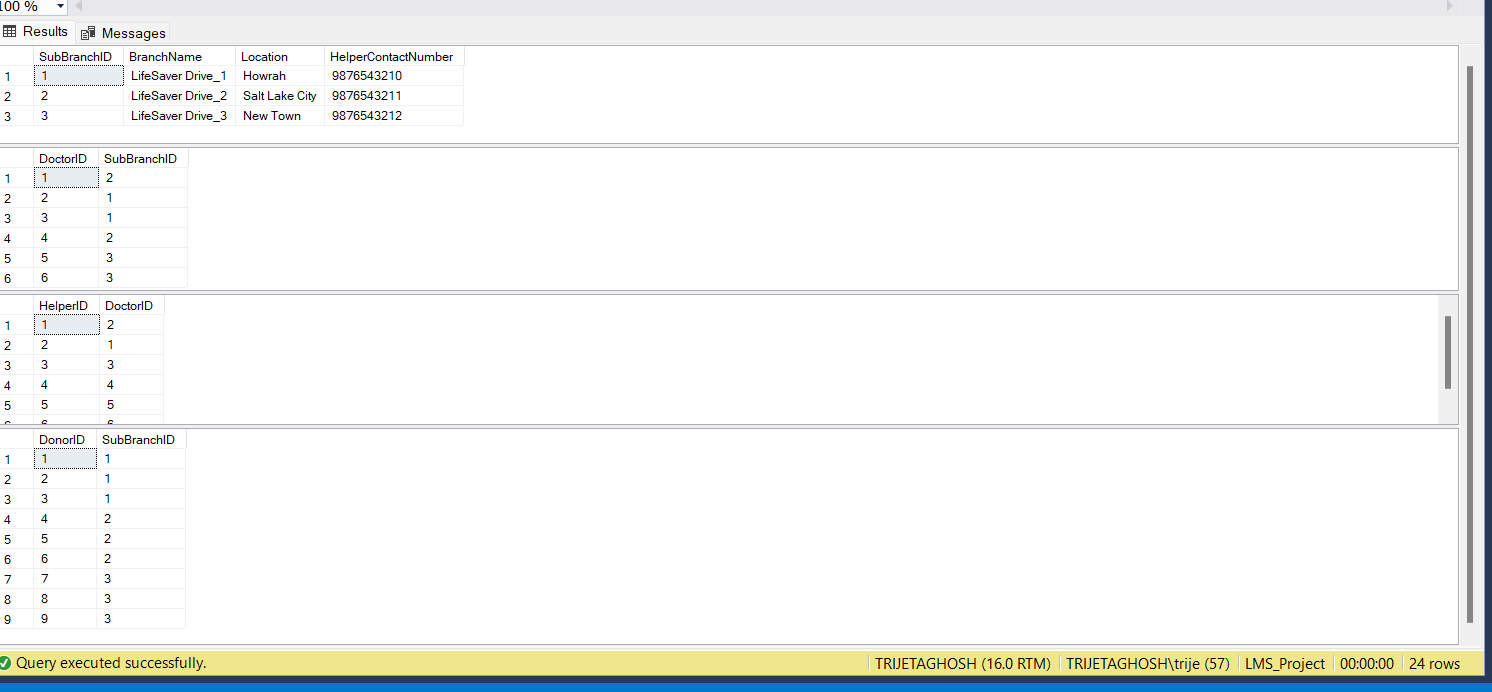
(9, 3); -- Donor 9 assigned to SubBranch 3

SELECT\*FROM SubBranch;

SELECT\* FROM Doctors\_SubBranch;

SELECT\*FROM Helpers\_Doctors;

SELECT\*FROM Donor\_Branch;

OUTPUT

Some Related Quary problems

**--Find the FirstNmae,and LastName whose blood group is AB-**

SELECT FirstName,LastName,Age

FROM Donor

WHERE BloodGRoup='AB-'

**--Find the names of donors, their blood group,Age, the names of the doctor and helper, and the sub-branch name where the blood group is O+.**

SELECT

d.DonorID,

d.FirstName AS DonorFirstName,

d.LastName AS DonorLastName,

d.BloodGroup AS DonorBloodGroup,

doc.FirstName AS DoctorFirstName,

doc.LastName AS DoctorLastName,

h.FirstName AS HelperFirstName,

h.LastName AS HelperLastName,

sb.BranchName AS SubBranchName

FROM

Donor d

JOIN

Donor\_Branch db ON d.DonorID = db.DonorID

JOIN

SubBranch sb ON db.SubBranchID = sb.SubBranchID

JOIN

Doctors\_SubBranch dsb ON db.SubBranchID = dsb.SubBranchID

JOIN

Doctors doc ON dsb.DoctorID = doc.DoctorID

JOIN

Helpers\_Doctors hd ON doc.DoctorID = hd.DoctorID

JOIN

Helpers h ON hd.HelperID = h.HelperID

WHERE

d.BloodGroup = 'O+' AND sb.BranchName ='LifeSaver Drive\_1';

**--Find all donor names, ages, blood groups, and their assigned doctor names along with the branch they are assigned to.**

SELECT

d.FirstName AS DonorFirstName,

d.LastName AS DonorLastName,

d.Age AS DonorAge,

d.BloodGroup AS DonorBloodGroup,

doc.FirstName AS DoctorFirstName,

doc.LastName AS DoctorLastName,

sb.BranchName AS AssignedBranch

FROM

Donor d

JOIN

Donor\_Branch db ON d.DonorID = db.DonorID

JOIN

SubBranch sb ON db.SubBranchID = sb.SubBranchID

JOIN

Doctors\_SubBranch dsb ON db.SubBranchID = dsb.SubBranchID

JOIN

Doctors doc ON dsb.DoctorID = doc.DoctorID;

**--Find Doctor FirstName,Specialization,Contactnum and their helpers,and their contactnumber in asc order**

SELECT

doc.FirstName AS DoctorFirstName,

doc.Specialization,

doc.ContactNumber AS DoctorContactNumber,

h.FirstName AS HelperFirstName,

h.ContactNumber AS HelperContactNumber

FROM

Doctors AS doc

INNER JOIN

Helpers\_Doctors AS hd ON doc.DoctorID = hd.DoctorID

INNER JOIN

Helpers AS h ON hd.HelperID = h.HelperID

ORDER BY

doc.FirstName ASC;

**--Find the total number of donors in each blood group:**

SELECT BloodGroup, COUNT(\*) AS TotalDonors

FROM Donor

GROUP BY BloodGroup;

**--Find the average age of donors for each blood group:**

SELECT BloodGroup, AVG(Age) AS AverageAge

FROM Donor

GROUP BY BloodGroup;

**--Find the sub-branches with the highest number of donors:**

SELECT sb.BranchName, COUNT(\*) AS TotalDonors

FROM SubBranch sb

JOIN Donor\_Branch db ON sb.SubBranchID = db.SubBranchID

GROUP BY sb.BranchName

ORDER BY TotalDonors DESC;

**--Find 5 max age doner name**

SELECT TOP 5 \* FROM Donor

ORDER BY DonorID DESC;

**--Name and Bloodgroup who has rare blood**

SELECT FirstName,LastName,BloodGroup

FROM Donor

WHERE BloodGroup NOT IN ('A+', 'B+', 'AB+', 'O+')

ORDER BY DonorID DESC;

**--Retrieve SubBranches with Helper Contact Numbers in Howrah:**

SELECT SubBranchID, BranchName, Location, HelperContactNumber

FROM SubBranch;

**--Update Helper Contact Number for a Specific SubBranch:**

UPDATE SubBranch

SET HelperContactNumber = '9876543213'

WHERE SubBranchID = 3;

**--Retrieve the details of doctors along with their associated sub-branches.**

SELECT

d.DoctorID,

d.FirstName AS DoctorFirstName,

d.LastName AS DoctorLastName,

d.Specialization,

sb.BranchName,

sb.Location

FROM

Doctors d

INNER JOIN

Doctors\_SubBranch ds ON d.DoctorID = ds.DoctorID

INNER JOIN

SubBranch sb ON ds.SubBranchID = sb.SubBranchID;

**-- List all donors along with their associated sub-branch locations**

SELECT

d.FirstName AS DonorFirstName,

d.LastName AS DonorLastName,

d.BloodGroup,

sb.Location

FROM

Donors d

INNER JOIN

Donor\_Branch db ON d.DonorID = db.DonorID

INNER JOIN

SubBranch sb ON db.SubBranchID = sb.SubBranchID;

Conclusion

the Blood Donation Management System (BDMS) represents a pivotal advancement in healthcare technology, with its streamlined processes, optimized resource management, and enhanced donor engagement. By digitizing donor registration, appointment scheduling, and donation tracking, the BDMS reduces administrative burdens and ensures a smoother experience for both donors and healthcare providers. Real-time monitoring of blood inventory levels and donor availability enables proactive resource allocation, minimizing shortages and reducing wastage. The system's comprehensive donor profiles and personalized communication strategies foster stronger donor engagement, leading to increased donor retention rates. Through data-driven insights on donor demographics, donation patterns, and inventory levels, the BDMS empowers healthcare administrators to make informed decisions, enhance strategic planning, and optimize resource allocation. Ultimately, the BDMS contributes to improving public health outcomes by ensuring the availability of safe and adequate blood supplies, thereby saving lives and enhancing patient care