BLOOD DONATION MANAGEMENT SYSTEM

TANAYA MAHALANABIS (22BCE1217)
Responsible for Introduction, Literature Work,ER diagram,PPT

KRIYA GADAG (22BCE1361)
Responsible for Problem description, Literature Work, Relational Schema

AYUSH GOSWAMI (22BCE5063)
(Responsible for Normalization,Literature Work, Future Plans

INTRODUCTION

The Blood Donation Management System (BDMS) is a comprehensive and efficient solution designed to streamline the complexities of blood donation processes. In the face of increasing demand for blood and its derivatives, managing the entire donation lifecycle becomes paramount. This DBMS project aims to create a robust system that facilitates the seamless coordination of blood donors, recipients, and blood banks. The system will encompass features such as donor registration, real-time inventory tracking, and automated notification systems. By leveraging database management principles, the BDMS seeks to optimize data storage, retrieval, and manipulation, ensuring accuracy, integrity, and accessibility of critical information. Through the implementation of this project, we aim to contribute to the enhancement of blood donation operations, ultimately saving lives and fostering a more efficient and organized healthcare ecosystem.

PROBLEM STATEMENT

The Blood Donation Management System (BDMS) project addresses the pressing need for an organized, efficient, and data-driven approach to managing blood donation processes. The current blood donation landscape is characterized by disparate systems and manual record-keeping, leading to inefficiencies in donor recruitment, inventory management, and distribution. The proposed BDMS aims to create a centralized database system with multiple attributes, including donor information, blood types, donation history, and inventory levels. This system will streamline the entire donation lifecycle, from donor registration to blood processing, storage, and distribution. The project will involve the design and implementation of a robust database management system to ensure data accuracy, integrity, and accessibility.

Once the database is populated with relevant data, the project will focus on data analysis to derive meaningful insights. By employing data analytics techniques, the team will extract valuable information regarding donation trends, seasonal variations, and donor demographics. This analysis will serve as a foundation for informed decision-making, allowing blood banks and healthcare institutions to optimize their resource allocation, donor outreach strategies, and emergency preparedness.

The project's ultimate goal is to propose future plans for the implementation of a data-driven blood donation management strategy. These plans will encompass recommendations for system enhancements, scalability, and integration with emerging technologies. The BDMS project seeks to contribute significantly to the advancement of blood donation practices, fostering a more proactive, responsive, and sustainable approach to meeting the ever-growing demand for blood products in healthcare systems.

OBJECTIVES

The primary objective of this project is to design and implement a BDMS that addresses the aforementioned challenges and achieves the following goals:

- Automate donor registration and management processes to streamline operations.
- Develop a centralized database system with multiple attributes to store donor information, blood inventory status, and recipient details.
- Implement efficient data retrieval and manipulation techniques for real-time access to critical information.
- Utilize data analytics to analyze donor demographics, donation patterns, and inventory trends for informed decision-making.
- Generate reports and visualizations to facilitate data-driven insights and future planning.
- Enhance communication and coordination between blood donors, recipients, and blood banks through automated notification systems.

PROBLEM DESCRIPTION

The Blood Donation Management System project aims to address the inefficiencies and challenges faced in managing blood donation processes. The current manual processes and lack of automation hinder the smooth functioning of blood banks and the availability of blood when needed. The proposed system will provide a comprehensive solution by incorporating the following features:

1. Automated Donor Registration: The system will automate the process of registering blood donors, simplifying the registration and updating of donor information.

2. Centralized Database System: A centralized database system will be developed to store donor information, blood inventory status, and recipient details. This centralized system will ensure efficient data management and retrieval.

3. Real-time Data Access: Efficient data retrieval and manipulation techniques will be implemented to ensure real-time access to critical information. This will enable quick retrieval of donor details, blood availability, and recipient information.

4. Data Analytics and Decision-making: The system will utilize data analytics techniques to analyze donor demographics, donation patterns, and inventory trends. This analysis will provide valuable insights for informed decision-making and resource planning.

PROBLEM DESCRIPTION

5. Reporting and Visualizations: The system will generate reports and visualizations to provide data-driven insights. Reports on donor statistics, blood inventory levels, and other relevant metrics will facilitate better planning and decision-making.

6. Automated Communication and Coordination: The system will enhance communication and coordination between blood donors, recipients, and blood banks through automated notification systems. Donors can receive reminders for donation appointments, recipients can be notified about blood availability, and blood banks can communicate urgent requirements. By addressing these challenges and implementing these features, the Blood Donation Management System will streamline operations, improve efficiency, and contribute to saving lives by ensuring the availability of blood when needed.

INFORMATION OF ENTITIES

We'll have total 8 entities whose detailed description is given below :

1) BB_Manager(m_ID,m_name,m_phno)
2) Recipient(reci_ID,reci_name,reci_age,reci_sex,reci_bgrp,reci_bqnty,reci_date,reci_phno)
3) Blood_Donor(bd_ID,bd_name,bd_age,bd_sex,bd_bgrp,bd_reg_date,bd_phno)
4) City(city_ID,city_ID)
5) Hospital_Info(hosp_needed_bgrp,hosp_ID,hosp_name,hosp_needed_qnty,hosp_phno)
6) Blood_Specimen(specimen_no,bgrp,status)
7) Disease_Finder(dfind_ID,dfind_name,dfind_phno)
8) Recording_Staff(reco_ID,reco_name,reco_phno)

RELATION BETWEEN ENTITIES

1. City and Hospital_Info:

Relationship = “in”

Type of relation = 1 to many

Explanation = A city can have many hospital in it. One hospital will belong in one city.

2. City and Blood_Donor:

Relationship = “lives in”

Type of relation = 1 to many

Explanation = In a city, many donor can live. One donor will belong to one city.

3. City and Recipient:

Relationship = “lives in” Type of relation = 1 to many

Explanation = In a city, many recipient can live. One recipient will belong to one city.

RELATION BETWEEN ENTITIES

4. Recording_Staff and Donor:

Relationship = “registers”

Type of relation = 1 to many

Explanation = One recording staff can register many donors. One donor will register with one recording officer.

5. Recording_Staff and Recipient:

Relationship = “records”

Type of relation = 1 to many

Explanation = One recording staff can record many recipients. One recipient will be recorded by one recording officer.

6. Hospital_Info and BB_Manager:

Relationship = “gives order to”

RELATION BETWEEN ENTITIES

Type of relation = 1 to many

Explanation = One Blood bank manager can handle and process requests from many hospitals. One hospital will place request to on blood bank manager.

7. BB_Manager and Blood Specimen:

Relationship = “deals with specimen”

Type of relation = 1 to many

Explanation = One Blood bank manager can manage many blood specimen and one specimen will be managed by one manager.

RELATION BETWEEN ENTITIES

8. Recipient and BB_Manager:

Relationship = “requests to”

Type of relation = 1 to many

Explanation = One recipient can request blood to one manager and one manager can handle requests from many recipients.

9. Disease_finder and Blood Specimen:

Relationship = “checks”

Type of relation = 1 to many

Explanation = A disease finder can check many blood samples. One blood sample is checked by one disease finder.

RELATIONAL SCHEMA

City:

City_ID (Primary Key)

City_Name

Blood_Donor:

bd_ID (Primary Key)

City_ID (Foreign Key)

bd_name

bd_phno

bd_age

bd_sex

bd_grp

bd_reg_date

Recipient:

Reci_ID (Primary Key)

City_ID (Foreign Key)

Reci_Name

Reci_phno

Reci_sex

Reci_age

Reci_date

Reci_bgrp

reci_bqty

Recording_Staff:

reco_ID (Primary Key)

reco_Name

reco_phno

RELATIONAL SCHEMA

Hospital_Info:

Hosp_ID (Primary Key)

City_ID (Foreign Key)

Hosp_Name

Hosp_phno

Hosp_needed_qty

hosp_needed_bgrp

BB_Manager:

M_ID (Primary Key)

Hosp_ID (Foreign Key)

M_Name

m_phno

Blood_Specimen:

Specimen_Ino(Primary Key)

M_ID (Foreign Key)

B_group

status

Disease_Finder:

dfind_id (Primary Key)

dfind_name

dfind_phno

RELATIONAL SCHEMA

Donor Table:

Attribute Name	Description	Type
bd_id	Blood Donor's Id	int
bd_Name	Blood Donor's Name	varchar
bd_age	Blood Donor's Age	int
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	int
city_id	City Id	int

Recipient Table:

Attributes Name	Description	Type
reci_id	Recipient's Id	int
reci_Name	Recipient's Name	varchar
reci_age	Recipient's age	int
reci_sex	Recipient's sex	char
reci_bgrp	Recipient's blood group	varchar
reci_bqnty	Recipient's blood quantity	int
reci_reg_date	Recipient's registration date	date
reco_id	Recording Staff's Id	int
city_id	City's unique Id	int
M_id	Blood Bank Manager's Id	int

RELATIONAL SCHEMA

City Table:

Attributes Name	Description	Type
city_id	City's unique id	int
city_name	City's name	varchar

Recording Staff Table:

Attributes Name	Description	Type
reco_id	Recording Staff's id	int
reco_name	Recording Staff's Name	Varchar
reco_PhNo	Recording Staff's Phone number	bigint

RELATIONAL SCHEMA

Blood Specimen Table:

Attributes Name	**Description**	**Type**
specimen_No	Blood Sample's unique id	int
b_grp	Blood Group	varchar
status	Whether blood is pure or not?	int
M_id	Blood Bank Manager's id	int
dfind_id	Disease Finder's unique id	int

Disease Finder Table:

Attributes Name	**Description**	**Type**
dfind_id	Disease Finder's unique id	Int
dfind_name	Disease Finder's name	varchar
dfind_phNo	Disease Finder's phone number	bigint

RELATIONAL SCHEMA

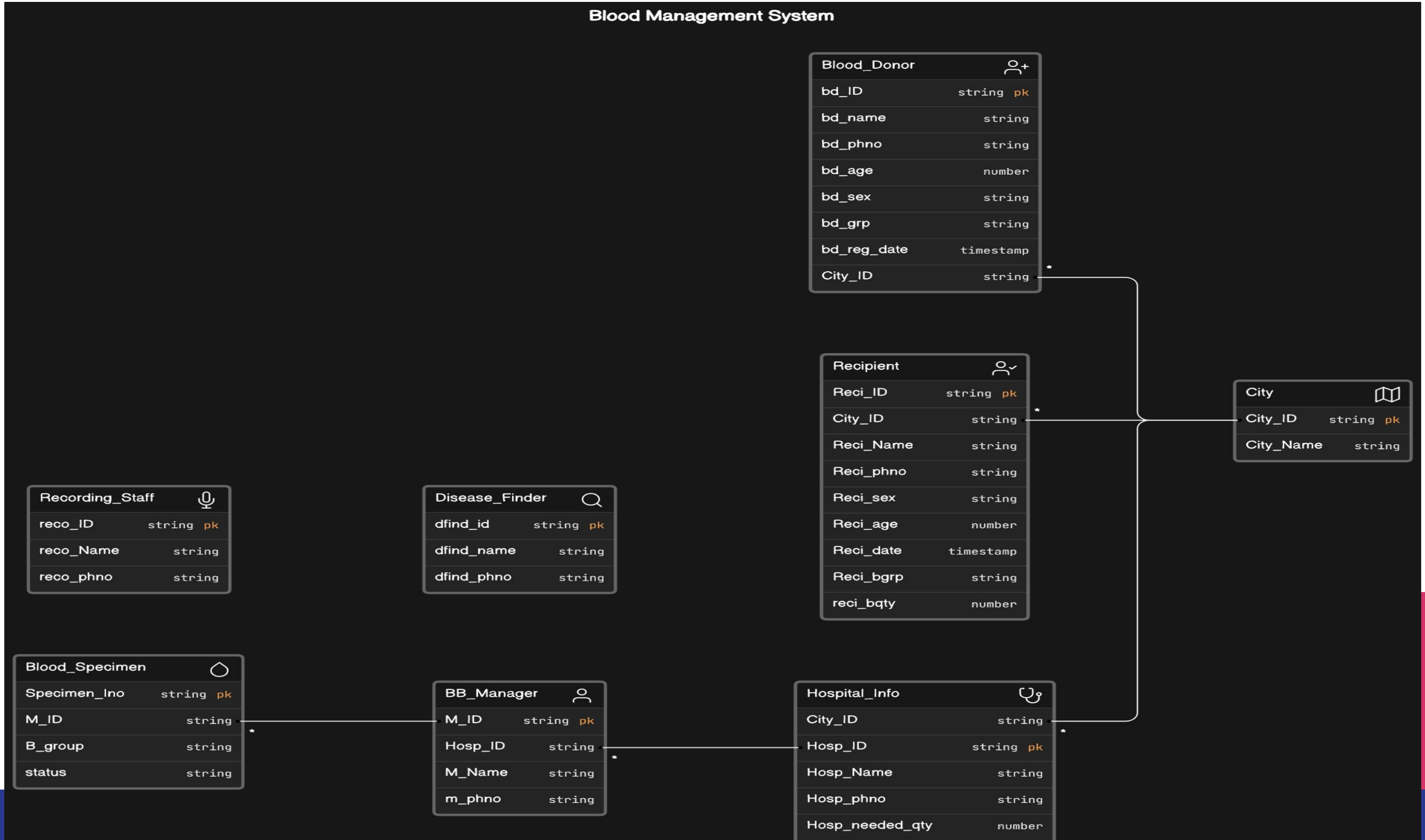
Blood Bank Manager Table:

Attributes Name	Description	Type
M_id	Blood Bank Manager's id	int
m_name	Blood Bank Manager's name	varchar
m_phNo	Blood Bank Manager's phone no	bigint

Hospital info Table:

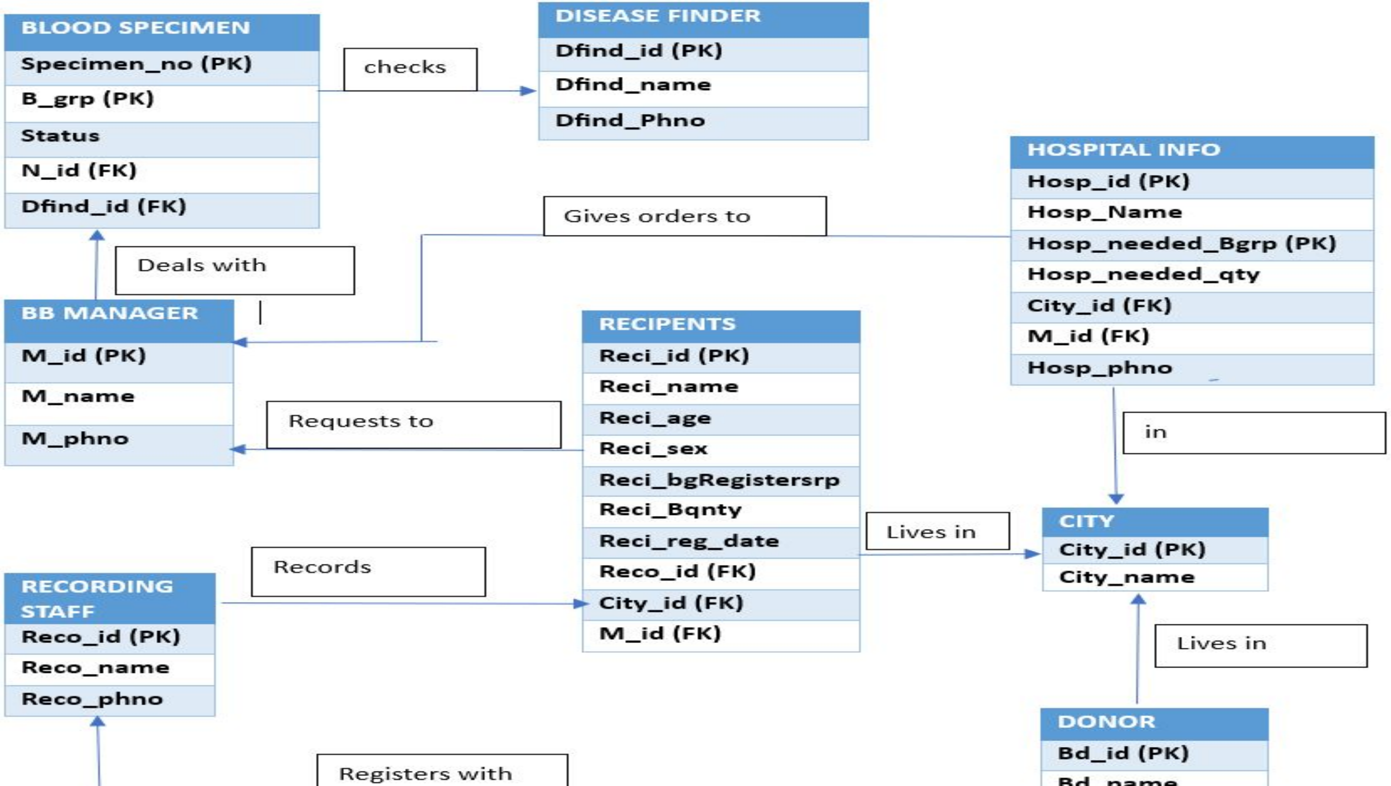
Attributes Name	Description	Type
hosp_id	Hospital's unique id	int
hosp_name	Hospital's name	varchar
hosp_needed_Bgrp	Blood group needed by hospital	varchar
hosp_needed_qnty	Quantity of blood group needed	int
city_id	City's unique id	int
M_id	Blood Bank Manger's id	int

ER DIAGRAM

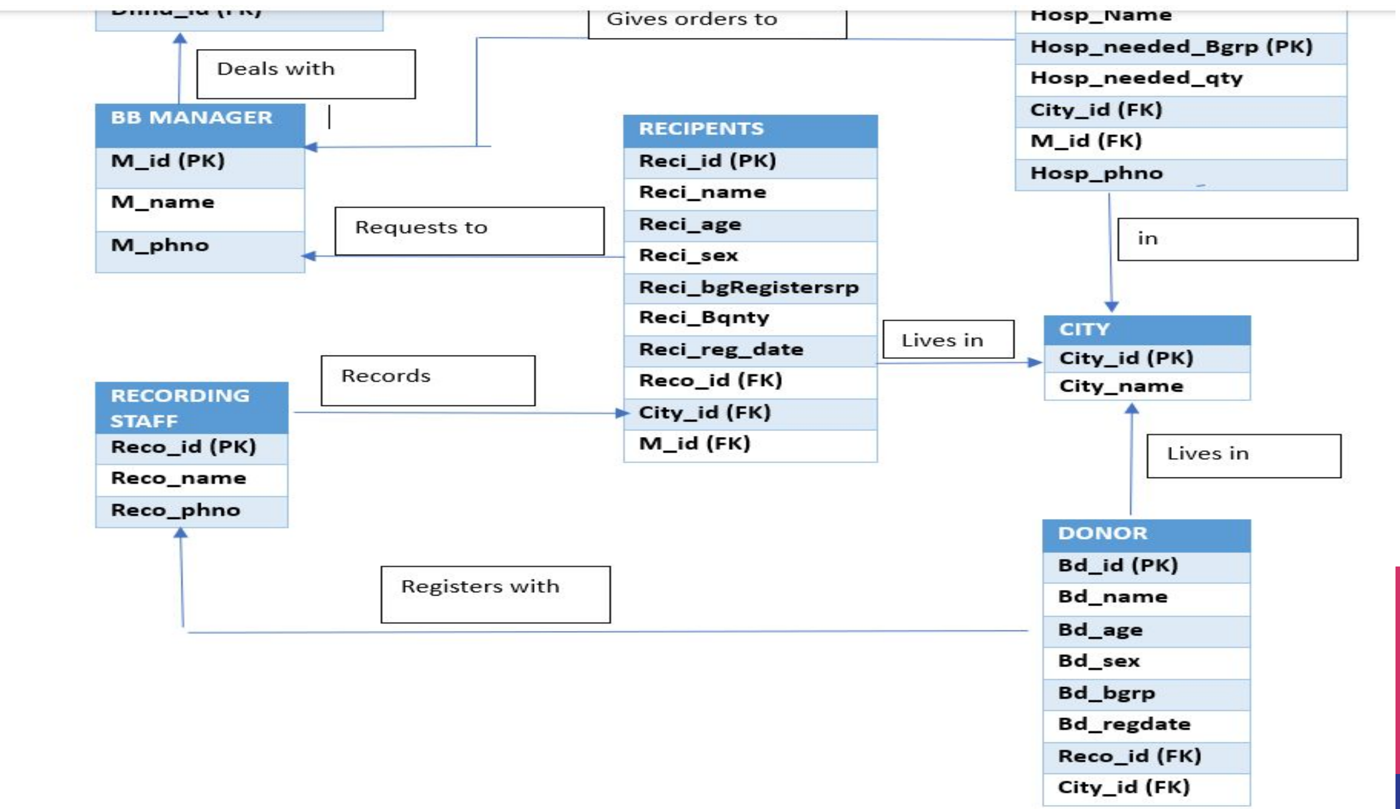


Blood Management System
Blood_Donor
bd_ID string pk
bd_name string
bd_phno string
bd_age number
bd_sex string
bd_grp string
bd_reg_date timestamp
City_ID string
Recipient
Reci_ID string pk
City_ID string
Reci_Name string
Reci_phno string
Reci_sex string
Reci_age number
Reci_date timestamp
Reci_bgrp string
reci_bqty number
City
City_ID string pk
City_Name string
Recording_Staff
reco_ID string pk
reco_Name string
reco_phno string
Disease_Finder
dfind_id string pk
dfind_name string
dfind_phno string
Blood_Specimen
Specimen_Ino string pk
M_ID string
B_group string
status string
BB_Manager
M_ID string pk
Hosp_ID string
M_Name string
m_phno string
Hospital_Info
City_ID string
Hosp_ID string pk
Hosp_Name string
Hosp_phno string
Hosp_needed_qty number

RELATIONAL MODEL



RELATIONAL MODEL



FUNCTIONAL DEPENDENCIES

City → Hospital_Info

City → Blood_Donor

City → Recipient

Recording_Staff → Donor

Recording_Staff → Recipient

Hospital_Info → BB_Manager

BB_Manager → Blood Specimen

NORMALIZATION

Normalization Rule Normalization rules are divided into the following normal forms:

1. First Normal Form

2. Second Normal Form

3. Third Normal Form

NORMALIZATION

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

NORMALIZATION

Second Normal Form (2NF)

For a table to be in the Second Normal Form

1. It should be in the First Normal form.

2. And, it should not have Partial Dependency.

NORMALIZATION

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.

2. And, it doesn't have Transitive Dependency.

NORMALIZATION OF BLOOD BANK DATABASE

To normalize the tables, we can apply the normalization rules and decompose the tables into smaller tables to eliminate partial and transitive dependencies.

City and Hospital_Info:

City → Hospital_Info

Normalized table:

City (City_ID, City_Name)

Hospital_Info (Hospital_ID, City_ID, Hospital_Name, Address)

NORMALIZATION OF BLOOD BANK DATABASE

City and Blood_Donor:

City → Blood_Donor

Normalized table:

City (City_ID, City_Name)

Blood_Donor (Donor_ID, City_ID, Donor_Name, Address, Blood_Type)

City and Recipient:

City → Recipient

Normalized table:

City (City_ID, City_Name)

Recipient (Recipient_ID, City_ID, Recipient_Name, Address)

NORMALIZATION OF BLOOD BANK DATABASE

Recording_Staff and Donor:

Recording_Staff → Donor

Normalized table:

Recording_Staff (Staff_ID, Staff_Name)

Donor (Donor_ID, Staff_ID, Donor_Name, Address, Blood_Type)

Recording_Staff and Recipient:

Recording_Staff → Recipient

Normalized table:

Recording_Staff (Staff_ID, Staff_Name)

Recipient (Recipient_ID, Staff_ID, Recipient_Name, Address)

NORMALIZATION OF BLOOD BANK DATABASE

Recording_Staff and Donor:

Recording_Staff → Donor

Normalized table:

Recording_Staff (Staff_ID, Staff_Name)

Donor (Donor_ID, Staff_ID, Donor_Name, Address, Blood_Type)

Recording_Staff and Recipient:

Recording_Staff → Recipient

Normalized table:

Recording_Staff (Staff_ID, Staff_Name)

Recipient (Recipient_ID, Staff_ID, Recipient_Name, Address)

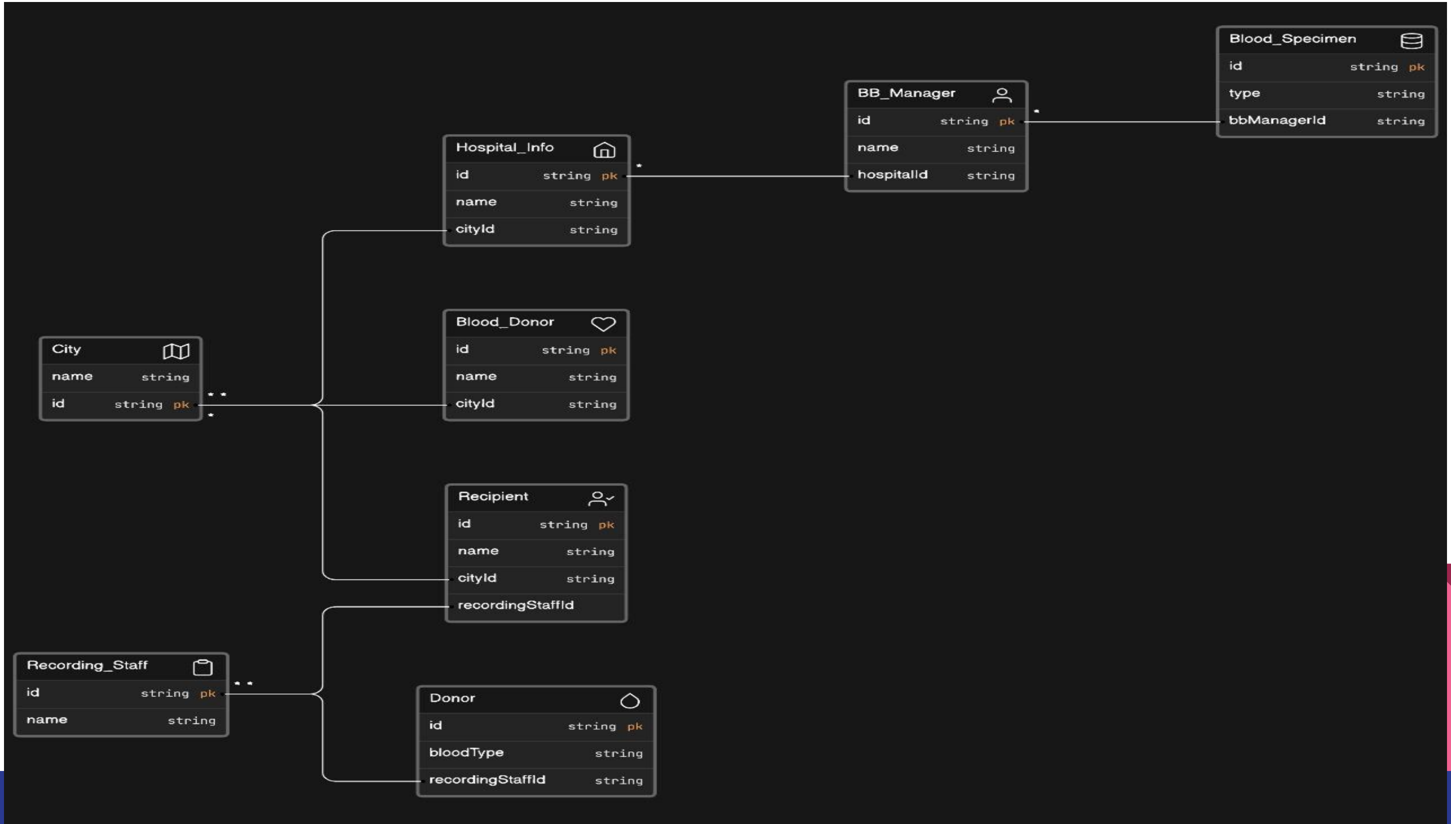
NORMALIZATION OF BLOOD BANK DATABASE

After normalization, the ER diagram would show the relationships between the entities and their attributes, with the primary keys and foreign keys identified. The ER diagram would show the relationships between the entities as one-to-many, with the primary key on the "one" side and the foreign key on the "many" side.

TABLES AFTER NORMALIZATION

	A	B	C	D	E	F	G
	Table Name	Column Name(s)	Data Type	Primary Key	Foreign Key(s)		
	City	City_ID	Integer	City_ID (Primary)			
	City	City_Name	Varchar(255)				
	Hospital_Info	Hospital_ID	Integer	Hospital_ID (Prir	City_ID		
	Hospital_Info	City_ID	Integer		City.City_ID		
	Hospital_Info	Hospital_Name	Varchar(255)				
	Hospital_Info	Address	Text				
	BB_Manager	BB_Manager_ID	Integer	BB_Manager_ID	Hospital_ID		
	BB_Manager	Hospital_ID	Integer		Hospital_Info.Hospital_ID		
	BB_Manager	Manager_Name	Varchar(255)				
	Blood_Specimer	Specimen_ID	Integer	Specimen_ID (P	BB_Manager_ID		
	Blood_Specimer	BB_Manager_ID	Integer		BB_Manager.BB_Manager_ID		
	Blood_Specimer	Blood_Type	Varchar(10)				
	Blood_Specimer	Collection_Date	Date				
	Recipient	Recipient_ID	Integer	Recipient_ID (Pr	City_ID		
	Recipient	City_ID	Integer		City.City_ID		
	Recipient	Recipient_Name	Varchar(255)				
	Recipient	Additional_Info	Text				
	Donation/Treatm	Donation/Treatm	Integer	Donation/Treatm	Recording_Staff_ID, Donor_ID/Recipient_ID		
	Donation/Treatm	Recording_Staff_	Integer		Recording_Staff.Recording_Staff_ID		
	Donation/Treatm	Donor_ID	Integer		(Null allowed or check constraint)		
	Donation/Treatm	Recipient_ID	Integer		(Null allowed or check constraint)		
	Recording_Staff	Recording_Staff_	Integer	Recording_Staff_ID (Primary)			
	Recording_Staff	Staff_Name	Varchar(255)				

ER DIAGRAM AFTER NORMALIZATION



FUTURE PLANS

The future plan for the Blood Donation Management System involves continuous improvement and expansion to enhance its functionality and usability. Here are some key aspects of the future plan:

1. Integration with Online Platforms: The system can be integrated with online platforms, such as websites or mobile apps, to enable online registration of donors, appointment scheduling, and blood donation campaigns. This will make it more convenient for donors to participate and contribute to the cause.

2. Enhanced Data Analytics: The system can be further developed to incorporate advanced data analytics techniques. This would include predictive analytics to forecast blood demand, machine learning algorithms to identify potential donors, and data visualization tools to provide more comprehensive insights for decisionmaking.

FUTURE PLANS

3. Integration with Healthcare Systems: Integration with healthcare systems, such as hospitals and clinics, can be explored to enable seamless sharing of donor and recipient information. This would facilitate efficient coordination between blood banks and healthcare providers, ensuring timely availability of blood for patients in need.

4. Expansion to Multiple Locations: The system can be expanded to cater to multiple blood banks and donation centers across different locations. This would enable centralized management of blood donation processes and efficient utilization of blood inventory on a larger scale.

5. Enhanced Security and Privacy Measures: Strengthening the security and privacy measures of the system will be a priority. Implementation of advanced encryption techniques, access control mechanisms, and regular security audits will ensure the confidentiality and integrity of donor and recipient data.

FUTURE PLANS

6. Continuous User Feedback and System Optimization: Regular feedback from users, including donors, recipients, and blood bank staff, will be collected to identify areas for improvement. User experience enhancements, system optimizations, and bug fixes will be implemented based on the feedback received.

By implementing these future plans, the Blood Donation Management System will continue to evolve and provide an efficient and effective solution for managing blood donation processes. It will contribute to saving lives by ensuring the availability of blood when needed, improving coordination between stakeholders, and making data-driven decisions for better resource planning