

BLOOD BANK MANAGEMENT SYSTEM

2CS402-Database Management System

Abstract

Implemented a Blood bank management system with required tables and attributes and relationships between different tables.

Taha Bhatia (21BCE027)
Jay Bhimani(21BCE029)
Chintan Detroja (21BCE047)
Dev Bachani(21BCE049)
Shyam Dhamecha(21BCE052)

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 donate the blood, that is 'donors'. But this is 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 a lot of time than the manual system. It will hoard, operate, recover and analyze 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 manpower is not required.

Existing Work:

There is a large body of research on the Blood Bank Management System, encompassing various aspects such as Blood Transactions between Hospitals and Blood Bank, Blood donations and Relationships between Inventory :

Blood Bank Management System:

<https://github.com/topics/blood-bank-management>

Article publication date: 6 December 2021

Issue publication date: 19 December 2022

Problem Statement:

This project aims to develop a Blood Bank Management System. A Blood Bank Management System can be used in any clinic, hospital, labs or any emergency situation which requires blood units for survival. Our system can be used to find required type of blood in emergency situations from either blood bank or even blood donors.

Current system uses a grapevine communication for finding blood in cases of emergency, may it be by a donor or blood bank. The intentions of proposing such a system is to abolish the panic caused during an emergency due to unavailability of blood.

Entity Relationship Diagram:

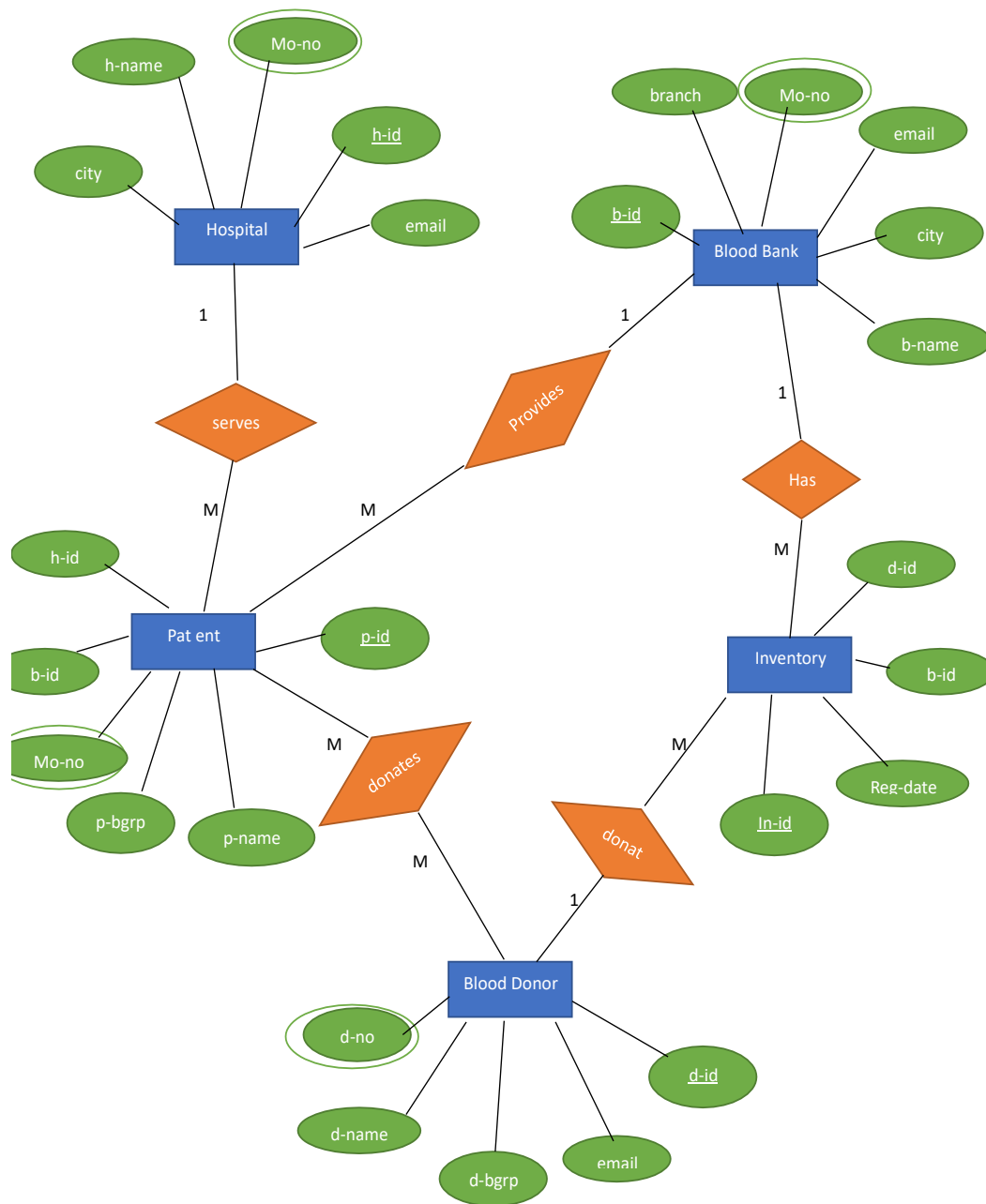


Table Description:

Blood Donor:

Blood Donor table can be used to store information about donors and their donations, Blood type, Blood Donor ID, email and contact.

The Blood Donor Id help identify the customer, and their Contact details. Donors can be distinguished on basis of Id. Overall, this table is essential for efficient management of blood transactions.

<i>Attribute</i>	<i>Data Type</i>	<i>Description</i>
<u>D_ID</u>	Integer	Unique identifier for each Donor.
D_name	Varchar (20)	The name of the Donor.
D_phno	Bigint	The contact number.
D_btype	Varchar (5)	Blood Group of Blood Donor
D_email	Varchar (20)	The email of the Blood Donor.

The donor is the person who donates blood, on donation a donor id (bd_ID) is generated and used as primary key to identify the donor information. Other than that name, age , blood group, phone number and registration dates will be stored in database under Blood_Donor entity.

Blood Bank:

A blood bank collects, separates, tests and stores blood until a patient needs it. If you've donated blood before, you've been part of the blood banking process. The first blood bank in the United States was established in 1937 by Dr. Bernard Fantus at Cook County Hospital in Chicago.

<i>Attribute</i>	<i>Data Type</i>	<i>Description</i>
<i><u>B_Id</u></i>	Integer	Unique identifier for each BloodBank.
<i>BCity</i>	Varchar (15)	City of Blood Bank
<i>Branch</i>	Varchar (15)	Branch/Area of Blood .
<i>B_Contact</i>	Varchar (50)	The date on which the order was shipped.
<i>B_email</i>	Varchar (20)	The email of Bloodbank.
<i>B_name</i>	Varchar (20)	Name of Blood Bank

Patient:

<i>Attribute</i>	<i>Data Type</i>	<i>Description</i>
<u>P_Id</u>	Integer	Unique identifier for each Patient.
Pname	Varchar(20)	Patient's name
P-Bgrp	Varchar(5)	Blood group of patient .
P_Contact	Bigint	Contact of Patient.
P_emial	Varchar(20)	The email of Bloodbank.
B_id(Foreign Key)	Integer	Blood bank id as a foreign key
H_id(Foreign Key)	Integer	Hospital Id as a foreign key

The Patient is the person who receives blood from blood bank, when blood is given to a patient a patient ID (P_ID) is generated and used as primary key for the recipient entity to identify blood recipients information. Along with it name ,age, blood group (needed), blood quantity(needed) , phone number are also stored in the data base under recipient entity.

Hospital:

<i>Attribute</i>	<i>Data Type</i>	<i>Description</i>
<i><u>H_id</u></i>	Integer	Unique identification number of Hospital
<i>Hname</i>	Varchar (20)	Hospital Name
<i>Hemail</i>	Varchar(20)	Email of Hospital.
<i>H_phno</i>	bigint	Contact number of hospital
<i>Hcity</i>	Varchar (10)	City of Hospital

In the data base, under Hospital_Info entity we will store the information of hospitals. In this hosp_ID and hosp_needed_Bgrp together makes the primary key. We will store hospital name and the blood quantity required at the hospital.

Inventory:

Blood inventory management demands a fine balance between ensuring blood availability and keeping wastage to a minimum. Identifying and analyzing various factors that contribute to wastage will provide an insight into ideal inventory management.

<i>Attribute</i>	<i>Data Type</i>	<i>Description</i>
<i><u>In_id</u></i>	Integer	Unique identification number of records in inventory
<i>Reg-Date</i>	VarChar (15)	Date on which blood is registered in blood bank
<i>B-id(Foreign key)</i>	Integer	Blood bank Id as a Foreign key.
<i>D-id(Foreign key)</i>	integer	Blood Donor Id as a foreign key.

Queries:

1. Show details of patient which need 'B+' blood and is admitted to hospital 'Berger and Sons'

Implementation:

```
select Pname,P_btype,P_contact_number,patient.h_id,hcity,hname from patient
left outer join hospital
on
patient.h_id=hospital.h_id
where patient.p_btype='B+' and hospital.hname='Berger and Sons';
```

```
mysql> select * from patient
-> inner join hospital
-> on
-> patient.h_id=hospital.h_id
-> where patient.p_btype='B+' and hospital.hname='Berger and Sons';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| P_id | Pname          | P_btype | P_contact_number | h_id | h_id | hname          | hemail          | h_contact_number | hcity          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 204  | Joshua Rodriguez | B+      | 8888888888      | 1104 | 1104 | Berger and Sons | nicole37@example.net | 9555555556      | Thompsonchester |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2. Patient 'Sarah Nicholson' was injured and needs Blood urgently.
Show details of donors who can donate blood.

Implementation:

```
select D_id,Dname,D_contact_number,D_btype,P_btype from blood_donor
natural join patient
where patient.pname='Sarah Nicholson' and
(patient.p_btype=Blood_Donor.D_btype or Blood_Donor.D_btype='O-');
```

```
mysql> select D_id,Dname,D_contact_number,D_btype,P_btype from blood_donor
-> natural join patient
-> where patient.pname='Sarah Nicholson' and (patient.p_btype=Blood_Donor.D_btype or Blood_Donor.D_btype='O-');
+-----+-----+-----+-----+-----+
| D_id | Dname          | D_contact_number | D_btype | P_btype |
+-----+-----+-----+-----+-----+
| 109  | Bobby Smith    | 8921012345      | O-      | B-      |
| 110  | Tanya Gomez    | 8712265431      | B-      | B-      |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3. A person have had an very bad accident and is in serious condition near city 'Holloway PLC' and need blood very urgently. Show details of nearby Blood Bank details.

Implementation:

select hname,Bname,Bcity,B_contact_number,B_branch,P_id from bloodbank
natural join patient natural join hospital
where hospital.hname='Holloway PLC' and hcity=Bcity;

```
mysql> select hname,Bname,B_city,B_contact_number,B_branch,P_id from bloodbank natural join patient natural join hospital
-> where hospital.hname='Holloway PLC' and hcity=Bcity;
ERROR 1054 (42S22): Unknown column 'B_city' in 'field list'
mysql> select hname,Bname,Bcity,B_contact_number,B_branch,P_id from bloodbank natural join patient natural join hospital
-> where hospital.hname='Holloway PLC' and hcity=Bcity;
```

hname	Bname	Bcity	B_contact_number	B_branch	P_id
Holloway PLC	Williams	Port	9090898989	East Branch	207

1 row in set (0.00 sec)

```
mysql> select *from Blood_Donor
-> Natural join patient
-> where patient.p_btype='AB-' and (Blood_Donor.d_btype='AB-' or Blood_donor.d_btype='A-' or Blood_donor.d_btype='B-' or Blood_donor.d_btype='O-');
```

D_id	Dname	D_Contact_number	D_btype	P_id	Pname	P_btype	P_contact_number	h_id
192	Paige Howard	9123456780	A-	200	Jill Singh	AB-	9123456789	1100
194	John Duarte	7123456012	AB-	200	Jill Singh	AB-	9123456789	1100
195	Randall Ramos	8791221091	AB-	200	Jill Singh	AB-	9123456789	1100
197	Brandon Cook	7891228987	AB-	200	Jill Singh	AB-	9123456789	1100
199	Bobby Smith	8921012345	O-	200	Jill Singh	AB-	9123456789	1100
110	Tanya Gomez	8712265431	B-	200	Jill Singh	AB-	9123456789	1100

```
6 rows in set (0.00 sec)
```

Learning Outcome:

Knowledge of database design: Developing a blood bank management system requires designing a database schema to store various types of data such as donor information, blood type, inventory management, and more. This project will help you gain knowledge of database design principles and implementation.

Understanding of software development lifecycle: Developing a blood bank management system involves various stages of software development lifecycle, including requirements gathering, designing, coding, testing, and deployment. You will gain practical knowledge of software development methodologies and practices.

Familiarity with programming languages: Developing a blood bank management system requires programming skills. Depending on the tools and technologies used, you may need to learn programming languages such as Java, Python, PHP, or C#.

Knowledge of web development: If you are building a web-based blood bank management system, you will need to have a solid understanding of web development technologies such as HTML, CSS, and JavaScript. You may also need to learn a web application framework such as React, Angular, or Vue.

Understanding of security considerations: A blood bank management system will handle sensitive information such as donor details and medical records. As such, security is of utmost importance. You will learn about security considerations, such as user authentication, data encryption, and secure communication protocols.

Teamwork and communication skills: Developing a blood bank management system is typically a team effort. You will need to work with others to ensure that the project is completed on time and to the required specifications. You will learn about teamwork and communication skills as you work with your team members to complete the project.

Project management skills: Building a blood bank management system is a complex project that requires proper planning, resource allocation, and management. You will learn about project management skills such as task scheduling, risk management, and quality assurance.

Overall, developing a blood bank management system is a challenging but rewarding project that will help you gain practical knowledge and skills in various areas of software development.