VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

FINAL PROJECT REPORT

BLOOD DONATION MANAGEMENT

Group members:

NAME	STUDENT ID	CONTRIBUTIONS
Trần Huỳnh Đức Tài	ITDSIU20132	110%
Huỳnh Thanh Thảo	ITDSIU20126	120%
Nguyễn Minh Quân	ITDSIU20094	110%
Phan Thế Thiện	ITDSIU20084	50%
Nguyễn Dương Tấn Đạt	ITDSIU20096	110%

Course: Principles of Database Management
Lecturer: Nguyen Thi Thuy Loan

Ho Chi Minh City, Vietnam
2023

TABLE OF CONTENTS:

ΧIII. Figure

I. Introduction

A blood donation is a procedure in which a person voluntarily consents to have blood drawn for use in transfusions that will later be necessary in hospitals for medical procedures that call for them. Whole blood donations are possible (blood that is taken directly from the body), or of certain blood elements including red blood cells, white blood cells, plasma, and platelets.

Blood banks frequently take part in the blood donation process as well as other activities like monitoring stockpiles, approving blood requests, and updating donor information.

The aim of this project is to establish a blood bank information system that focuses on creating an online system that is accessible for both donors and administrators, as well as strengthening blood banks in Vietnam. Donors can access information about their previous blood donations, such as blood test results and contribution patterns, to help them plan their upcoming donations, and they can also update their personal information through the system.

II. Overview

The blood donation management project is a system designed to manage the flow of blood from donors to recipients. It utilizes the principles of database management to create a centralized and efficient blood donation management system that can be used by blood banks, hospitals, and other healthcare facilities.

The project includes features such as donor registration, blood collection, inventory management, and blood distribution. By implementing a centralized database management system, the project aims to improve the safety, accuracy, efficiency, and effectiveness of blood donation and transfusion processes.

III. Goal

The system aims to achieve two things: to provide convenience while optimizing blood donation and distribution processes, and to ensure the creation of a comprehensive and efficient database schema, ensuring data accuracy and integrity. Moreover, by upgrading the online database system, we believe that we can improve the availability and quality of blood for patients in need while streamlining the entire process from donor to recipient through the use of database management principles. As a result, our system not only encourages customers to use the blood donation system to save time and effort and facilitate data collection and information retrieval for blood donation history but also for procedural blood collection procedures.

IV. Initial Approach

Blood donation management is designed to track the workflow of blood donation and transfusion processes. The following entities are typically included in the system to facilitate this tracking:

Donor:

- Donor ID
- Donor Name
- D_Blood Type
- Medical Report
- Personal Information
    - Day of Birth
    - Sex
    - D_Address
    - D_Number
    - Weight
    - Age

The “donors” entity includes information about the donor with unique IDs, such as their name, address, personal information, and blood type. Donors can register with the system and provide information about their medical history, which can help determine if they are eligible to donate blood.

Donates:

- Date
- Blood ID
- Donor ID

Donates: This component of the system manages the process of collecting, testing, and storing blood donations. Once a donor is eligible to donate, they are directed to the donation area, where their blood is collected and sent to the laboratory for testing. If the donation passes all necessary tests, it is stored in a blood bank for future use.

Blood:

- Blood ID
- B_Blood type
- Volume

Blood: This component of a system that tracks the inventory of blood products from blood donors, for processing and storage in blood banks

Staff:

- Staff ID
- S_Name
- S_Address
- S_Number

Staff: The staff is responsible for three jobs: managing the donor's registration process, working at Hospital and Blood Bank.

Blood bank:

- Blood Bank ID
- BB_Name
- BB_Address
- BB_Number
- BB_Blood Type
- Operating Hours

The initial approach includes data fields related to the Blood Bank component of the blood donation management project, including unique IDs for the blood bank, blood units, and Staff, operating hours and blood type information. These data fields help manage the blood supply efficiently and ensure its safety and accessibility

Order:

- Hospital ID
- Blood Bank ID

Orders: Hospitals order blood from blood banks as needed for transfusion. The system will need to manage these orders, track the delivery of blood units, and ensure that the right blood type is delivered to the right patient.

Hospital:

- Hospital ID
- H_Name
- H_Address
- H_Number

The hospital's initial approach was to maintain a database of patient information, including ID set a unique, name, address, number. This helps to ensure that the patient receives the correct transfusion and that the blood supply is used efficiently. In addition, the Hospital receives blood products from blood banks upon request. They use blood for patients who need blood transfusions during surgery or to manage various medical conditions.

Patient:

- Patient ID
- P_Name
- P_Blood Type
- P_Number
- Medical Conditions

The initial approach for the patient is to maintain a comprehensive database of patient information, including patient ID set a unique, name, blood type, number, medical condition, and blood type. This helps healthcare professionals to quickly access the necessary information to ensure that the correct blood type is transfused to each patient and to identify any potential risks or complications associated with the transfusion.

V. Timeline and Contribution

Stage	Week	Task	Member	Due
1. Project analysis and planning	1	Research information about blood donation and current blood donation management	All	Feb 12th
		Collect data for analysis	All	
		Search documents and references related to the project.	Thien, Dat	
		Discuss tools used to communicate between members, IDEs for project implementation, and project management.	Thao, Tai	
		Determine the right goals and methods to best manage and complete the project	Dat, Quan	
	2	Define functional and non-functional requirements of the project	Thao	Feb 19th
		Complete the timeline for the whole project	All	

		Discuss and determine the appropriate database management system	All	
		Review stage 1	All	
2. Diagram design	3	Define use cases and actors for the system	Thao	Feb 26th
		Define in detail the relationships between use cases and actors	Thao, Tai	
		Complete the proposal for the project.	All	
		Submit proposal	Thao	
	4-5	Design relational models diagram	Quan, Dat	Mar 12th
		Design class and ERD diagram	Thao	
		User interface design with key functions.	Tai, Thien	
		Review stage 2	All	
3. Complete application	5-8	Create database tables, define their relationships and constraints	Thien, Dat	April 23rd
		Query each function	Tai, Thao	
		App interface design	Quan, Tai	
		Connect the database to the app	Quan	
		Review stage 3	All	
4. Review the	8-10	Test all functions	Tai, Thao	May

project and application.		Code review and fix bugs	Tai, Quan	3rd
		Prepare slides and documents	Dat, Thien	
		Complete the report	Thao, Quan	
		Review stage 4 and submit the report	All	
	Last 1-3 week	Presentation	All	May 8th - 22nd

Table 1. Timeline and contribution

VI. Design Entity-Relationship Diagram

1. Determine the Entities

- Donor
- Blood
- Blood Bank
- Hospital
- Patient
- Staff

2. *Determine Attributes for Entities*

- Donor
  - Donor ID
  - Donor Name
  - D_Blood Type
  - Medical Report
  - Personal Information
    - Day of Birth
    - Sex
    - D_Address
    - D_Number

        - Weight
        - Age

- Blood
    - Blood ID
    - Volume
    - B_Blood Type

- Staff
    - Staff ID
    - S_Name
    - S_Address
    - S_Number

- Blood Bank
    - Blood Bank ID
    - BB_Name
    - BB_Address
    - BB_Number
    - BB_Blood Type
    - Operating Hours

- Hospital
    - Hospital ID
    - H_Name
    - H_Address
    - H_Number

- Patient
    - Patient ID
    - P_Name
    - P_Blood Type
    - P_Number

+ Medical Conditions

3. *Define the Relationships Between Entities*

- Relationship between Donor and Blood is ‘Donates’.
- Relationship between Donor and Staff is ‘Registers’.
- Relationship between Blood Bank and Blood is ‘Stored’.
- Relationship between Blood Bank and Staff is ‘Emoloyed_by’.
- Relationship between Blood Bank and Hospital is ‘Orders’.
- Relationship between Hospital and Patient is ‘Delivers’.
- Relationship between Staff and Hospital is ‘Work_at’.

4. *Add Binary relationships*

Cardinality is a mathematical term, and as stated above, it tells us the number of interactions entities have with each other. Cardinality is simply a number ratio expressed in symbols, like one-to-one or one-to-many.

Cardinality	**Symbol**	**Description**
One-to-one	<1..1>	One instance of the first entity can correspond to only one instance of the second entity.
One-to-many	<1..N>	One instance of the first entity can correspond to more than one instance of the second entity.
Many-to-one	<N..1>	More than one instance of the first entity can correspond to the same one instance of the second entity.
Many-to-many	<N..M>	More than one instance of the first entity can correspond to more than one instance of the second entity.

Table 2. Type of Cardinality

In our project the cardinalities are given:

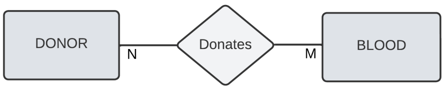
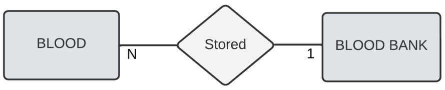
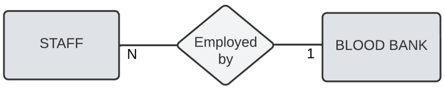
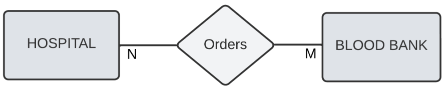
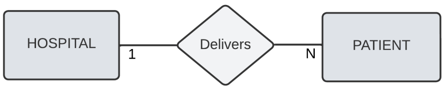
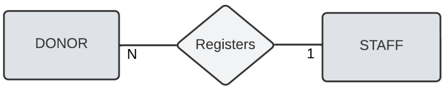
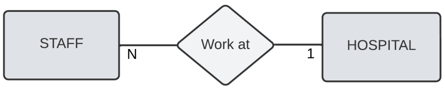
Cardinality	Description
	Many DONORS donate BLOOD for many bags.
	Many BLOOD bags are stored in one BLOOD BANK.
	One BLOOD BANK employs many STAFF.
	Many HOSPITALS can order many BLOOD BANK.
	One HOSPITAL can deliver to many PATIENTS.
	Many DONOR register at one STAFF.
	Many STAFF work at one HOSPITAL.

Table 3. Cartinality in project

5. *Implement the Entity - Relationship Diagram*

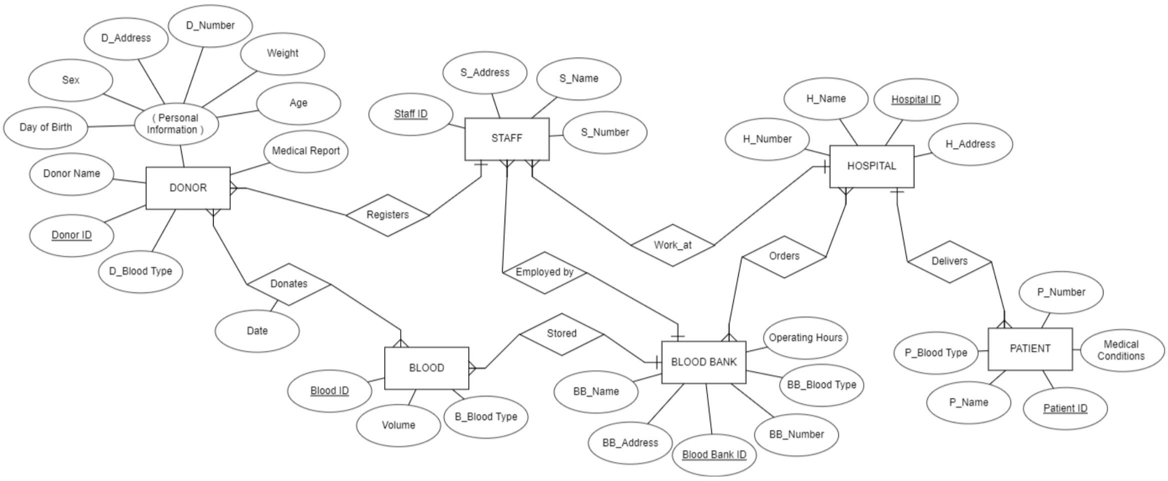


Figure 1. Entity - Relationship Diagram

6. *Translate Entity - Relationship Diagram to Relational Database*

DONOR (<u>Donor ID</u>, D_Blood Type, D_Name, D_Number, Day of Birthday, Sex, D_Address, Weight, Age, Medical Report, Staff ID)

BLOOD (<u>Blood ID</u>, B_Blood Type, volume, Blood Bank ID)

STAFF (<u>Staff ID</u>, S_Name, S_Number, S_Address, Blood Bank ID, Hospital ID)

BLOOD_BANK (<u>Blood Bank ID</u>, BB_Blood Type, BB_Address, Operating Hours, BB_Name)

HOSPITAL (<u>Hospital ID</u>, H_Name, H_Address, H_Numer)

PATIENT (Patient ID, P_Name, P_Blood Type, Medical Conditions, P_Number, Hospital ID)

Donates (Donor ID, Blood ID, Date)

Orders (Hospital ID , Blood Bank ID)

VII. Convert ER Diagram to Relational Model

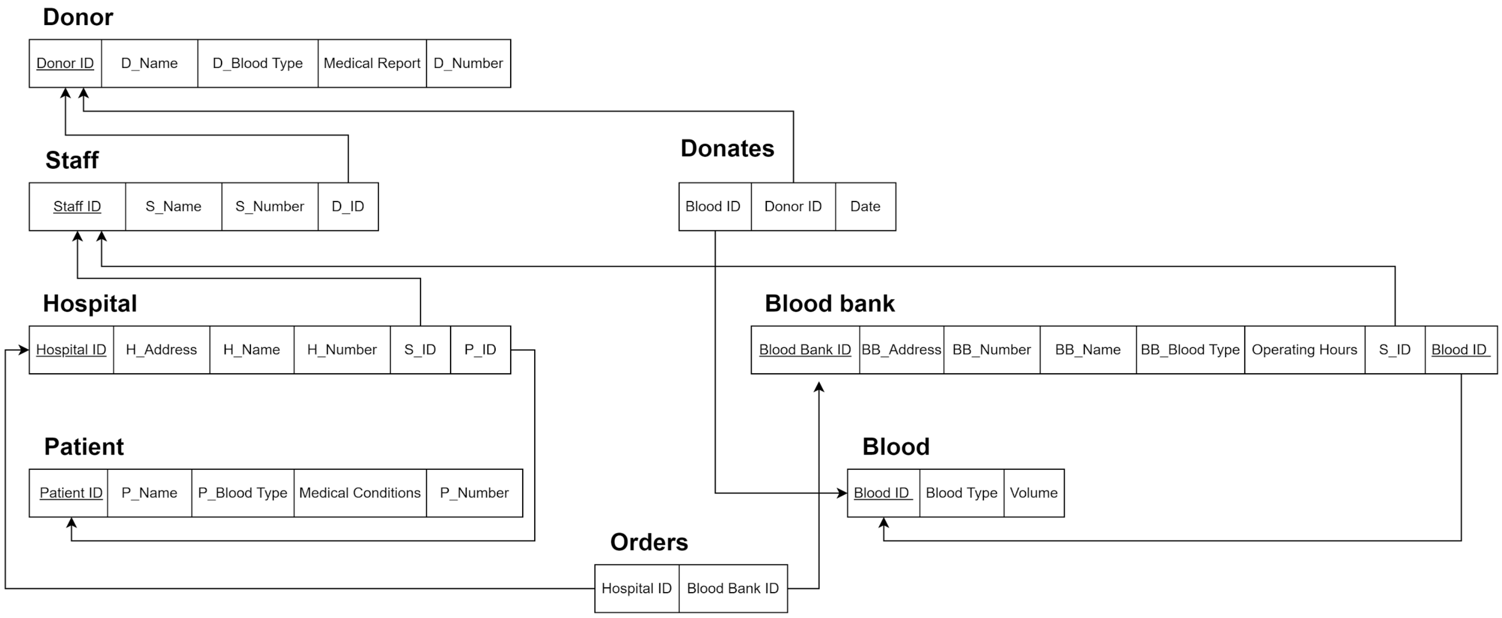


Figure 2. Relational Model

- Step 1: Regular Entity Types

1. For each regular/strong entity type, create a corresponding relation that includes all the simple attributes (includes simple attributes of composite relations)

2. Choose one of the key attributes as primary
- If composite, the simple attributes together form the primary key

3. Any remaining key attributes are kept as secondary unique keys (these will be useful for physical tuning with reference to indexing analysis)

- Step 2: Weak Entity Types

    1. For each weak entity type, create a corresponding relation that includes all the simple attributes

    2. Add as a foreign key all of the primary key attribute(s) in the entity corresponding to the owner entity type

    3. The primary key is the combination of all the primary key attributes from the owner and the partial key of the weak entity, if any

- Step 3: Mapping Binary 1-to-1

    1. Choose one relation as S, the other T
        - Better if S has total participation (reduces number of NULL values)

    2. Add to S all the simple attributes of the relationship

    3. Add as a foreign key in S the primary key attributes of T

- Step 4: Binary 1-to-N

    1. Choose the A relation as the type at the N - side of the relationship, other is T

    2. Add as a foreign key to S all of the primary key attribute(s) of T

- Step 5: Binary M-to-N

    1. Create a new relation S (termed: relationship relation)
       In some ERD dialects, actually drawn in

2. Add as foreign keys the primary keys of both relations; their combination forms the primary key of S

3. Add any simple attributes of the M : N relationship to S

- Step 6: Multivalued Attributes

    1. Create a new relation S

    2. Add as foreign keys the primary keys of the corresponding relation

    3. Add the attribute to S (if composite, the simple attributes); the combination of all attributes in S forms the primary key

- Step 7: Specialization/Generalization

    A. Multiple relations - subclass and superclass
        - Usually works (assumes unique id at parent)

    B. Multiples relations - subclass only
        - Should only be used for disjoint

    C. Single relation with one type attribute
        - Only for disjoint, can result in many NULLs

    D. Single relation with multiple type attributes
        - Better for overlapping, could be disjoint

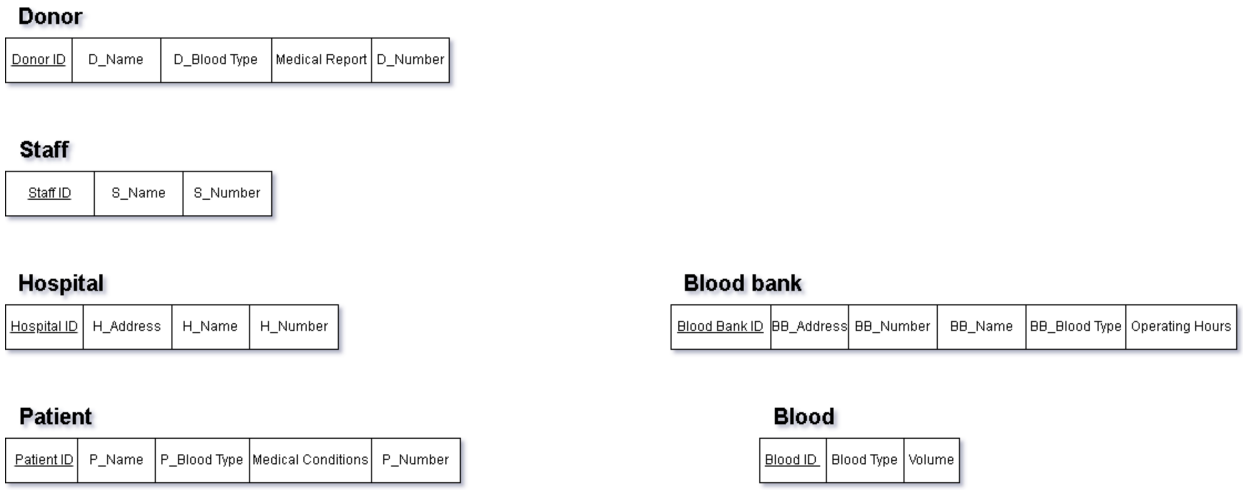
In this project, base on Entity - Relationships Diagram, we removed step 2 (Weak Entity Types), step 3 (Mapping Binary 1-to-1), step 6 (Multivalued Attributes), and step 7 (Specialization or Generalization) because the component in the above step is not included in our Entity - Relationships Diagram . The steps in our project are performed in the following order:

- Step 1: Regular Entity Types

  1. For each regular/strong entity type, create a corresponding relation that includes all the simple attributes (includes: Donor, Staff, Hospital, Patient, Blood Bank, Blood)

  2. Choose one of the key attributes as primary
     - Donor : Donor ID
     - Staff : Staff ID
     - Hospital: Hospital ID
     - Patient: Patient ID
     - Blood Bank: Blood Bank ID
     - Blood: Blood ID

  3. Any remaining key attributes are kept as secondary unique keys (these will be useful for physical tuning with reference to indexing analysis)



Donor

Donor ID	D_Name	D_Blood Type	Medical Report	D_Number

Staff

Staff ID	S_Name	S_Number

Hospital

Hospital ID	H_Address	H_Name	H_Number

Blood bank

Blood Bank ID	BB_Address	BB_Number	BB_Name	BB_Blood Type	Operating Hours

Patient

Patient ID	P_Name	P_Blood Type	Medical Conditions	P_Number

Blood

Blood ID	Blood Type	Volume

Figure 3. Relational Model in Step #1

- Step 4: Binary 1-to-N

    1. Choose the A relation as the type at the N - side of the relationship, other is T
    2. Add as a foreign key to S all of the primary key attribute(s) of T

- Choose entity “Staff” as the N-side of the relationship, which means that staff can have many relationships with entity “Donor”. Then add a foreign key (D_ID) to the "Staff " entity that references the primary key attribute(s) of the "Donor" entity (Donor ID).

- Similarly, we would repeat this process for the entity “Hospital” can have many relationships with other entities (Staff and Patient) with foreign key B_ID to the “Staff” and P_ID to the “Patient”. The entity “Blood Bank” has a relationship with the entity “Blood” throught the foreign key S_ID was added to “Blood Bank”.

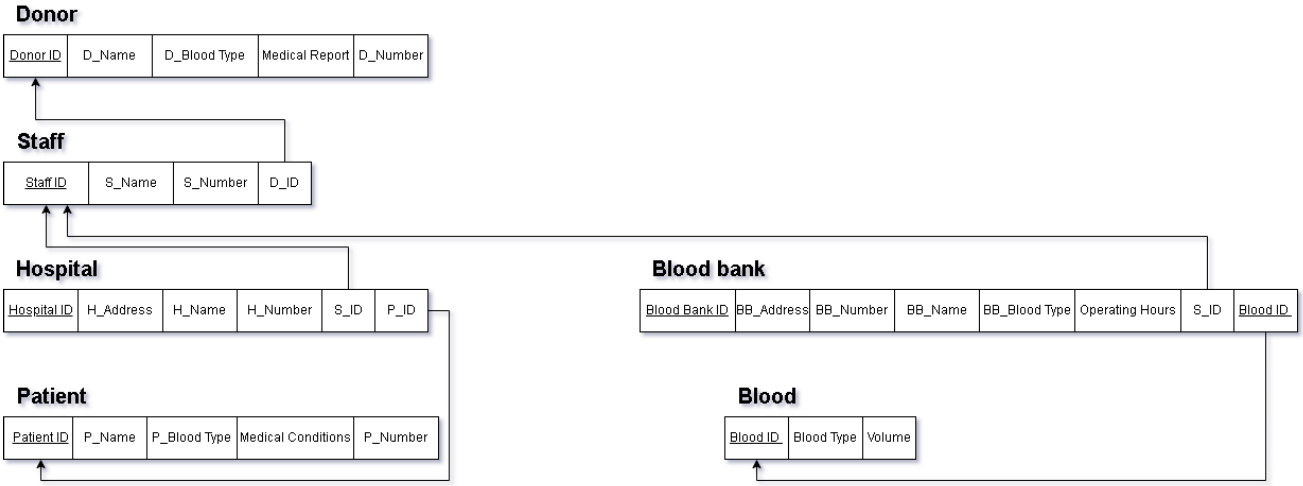


Figure 4. Relational Model in Step #4

- Step 5: Binary M-to-N

    1. Create two new relations are Orders and Donates.

    2. Add as foreign keys the primary keys of both relations; their combination forms the primary key of S

- In Orders relation, we add two foreign keys here: Blood ID (in Blood entity) and Donor ID (in Donor entity).

- With Donates relation, we add two foreign keys here: Hospital ID (in Hospital entity) and Blood Bank ID (in Blood Bank entity).

3. Add any simple attributes of the M : N relationship to S

   - Orders relation: draw two connections. The first connection comes from Blood ID to Blood ID in the Blood entity. The second is between Donor ID and Donor ID in the Donor entity.

   - Donates relation: Forge two interconnections: the first one between the Hospital ID of the Hospital entity and the Hospital ID. The second link should be established between the Blood Bank ID and the Blood Bank ID of the Blood Bank entity.

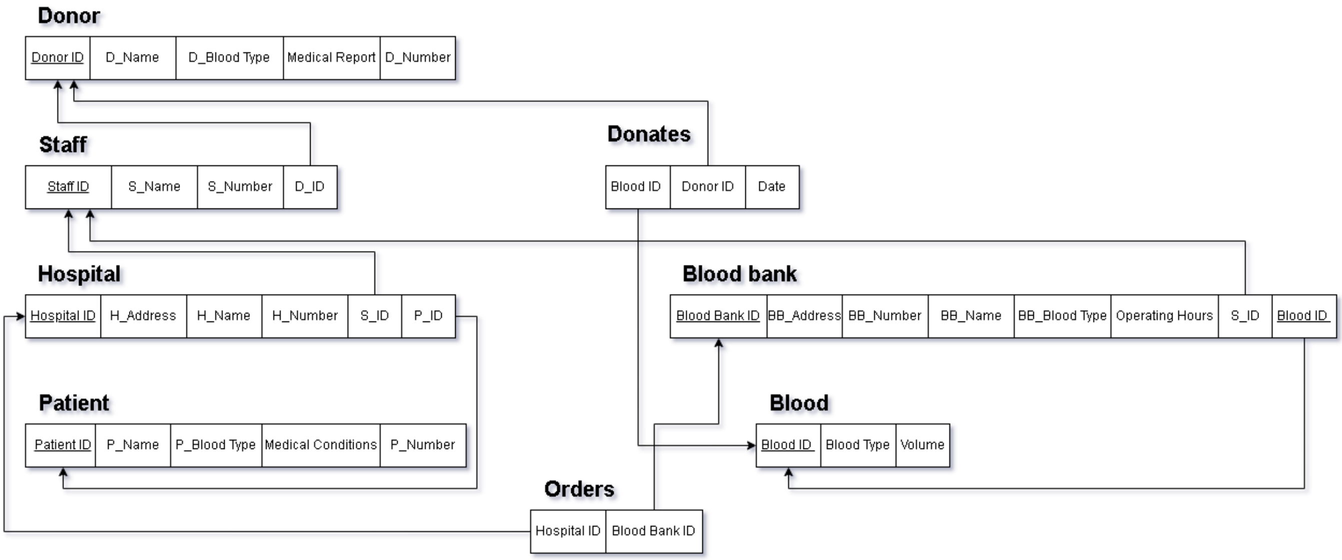
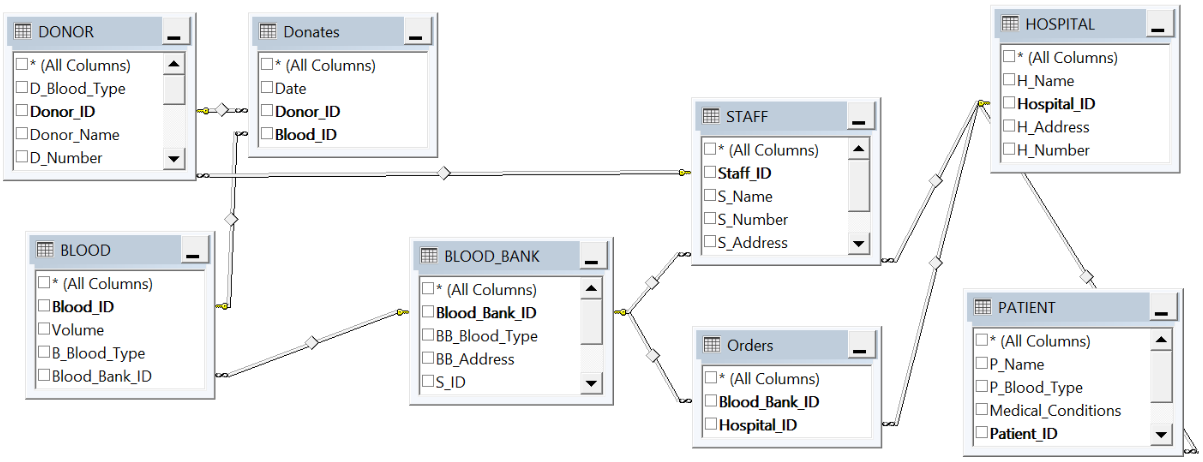


Figure 5. Relational Model in Step #5

VIII. SQL Server Database

https://drive.google.com/drive/folders/1vA-pLAz3r_HwTKnIeLVVK_1T4hj0QOzl?usp=sharing

IX. Create View



X. Query questions

1. How many female donors are over 40 years old?

- **SQL:**

SELECT COUNT (*)FROM donor
WHERE sex = 'Female' AND age > 40

- **Relational algebra:**

$\pi_{COUNT\,(*)}$

$\gamma_{COUNT\,(*)}$

$\sigma_{sex\,=\,\text{"Female"}\ AND\ age\,>\,40}\ donor$

- **Tree Diagram:**

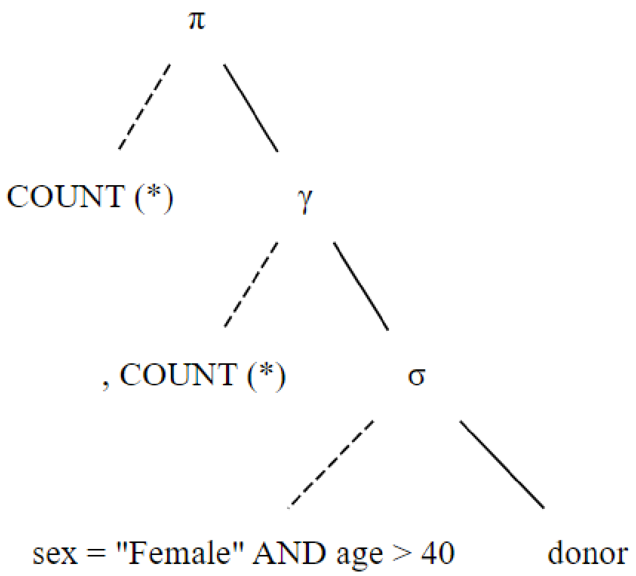


Figure 6. Tree Diagram in Question #1

2. Find the address of the staff at "Yet Another Town Hospital".

- **SQL**

SELECT s_address
FROM staff INNER JOIN hospital ON staff.hospital_id = hospital.hospital_id
WHERE h_name = 'Yet Another Town Hospital'

- **Relational algebra:**

$$\pi_{s\_\text{address}}$$

$$\sigma_{h\_\text{name} = \text{"Yet } \textit{Another Town Hospital}\text{"}} (\text{staff} \bowtie_{staff\,.\,hospital\_\text{id} = hospital\,.\,hospital\_\text{id}} \mathit{hospital})$$

- **Tree Diagram**

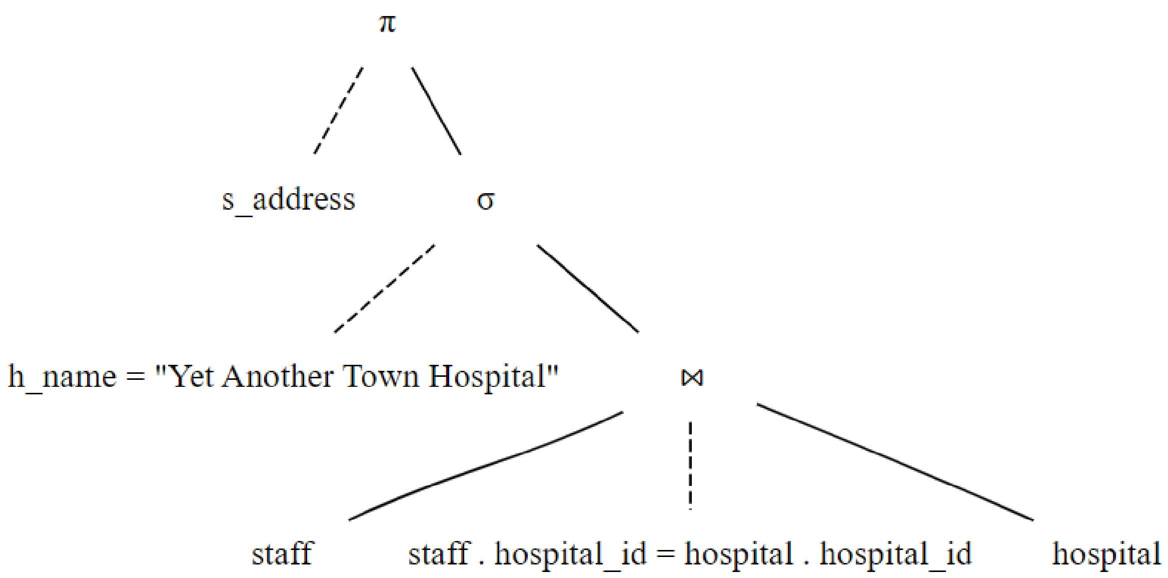

π
s_address
σ
h_name = "Yet Another Town Hospital"
⋈
staff
staff . hospital_id = hospital . hospital_id
hospital

Figure 7. Tree Diagram in Question #2

3. Find all blood types that 'Anytown Blood Bank’ has.

- **SQL:**

```
SELECT DISTINCT bb_blood_type
FROM blood_bank AS bb
 WHERE bb_name = 'Anytown Blood Bank'
```

- **Relational algebra:**

δ

$\pi_{BB\_\mathrm{Blood\_Type}}$

$\sigma_{BB\_\mathrm{Name} = \text{"Anytown } \textit{Blood Bank}\text{"}}$

$\rho_{BB}\, BLOOD\_BANK$

- **Tree Diagram:**

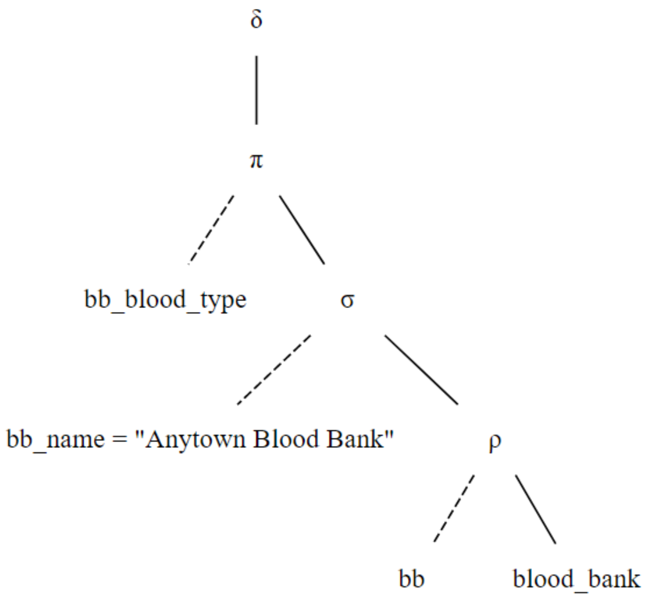


Figure 8. Tree Diagram in Question #3

4. Find the donor's name and the blood volume that donate on "2023-4-22".

- **SQL:**

```
SELECT donor_name, volume
FROM donor INNER JOIN donates ON donor . donor_id = donates . donor_id
INNER JOIN blood ON donates . blood_id = blood . blood_id
 WHERE date = '2023-04-22'
```

- **Relational algebra:**

$\pi_{donor\_name,\ volume}$
$\sigma_{date = "2023-04-22"}(\text{donor} \bowtie_{donor\ .\ donor\_id = donates\ .\ donor\_id} donates \bowtie_{donates\ .\ blood\_id = blood\ .\ blood\_id} blood)$

- **Tree Diagram**

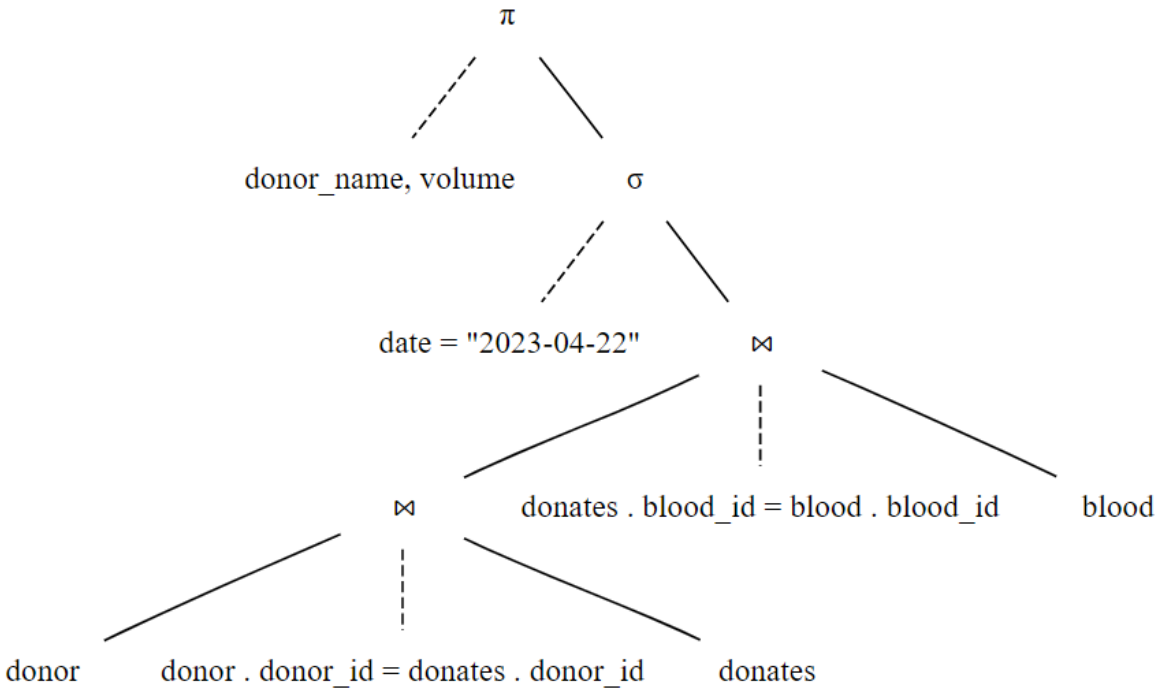

π
donor_name, volume
σ
date = "2023-04-22"
⋈
⋈
donates . blood_id = blood . blood_id
blood
donor
donor . donor_id = donates . donor_id
donates

Figure 9. Tree Diagram in Question #4

5. Find the patient blood type and staff phone number at "789 Elm St, Yet Another Town USA".

- **SQL:**

SELECT p_blood_type, s_number
FROM patient INNER JOIN hospital ON patient . hospital_id = hospital . hospital_id INNER JOIN staff ON patient . hospital_id = staff . hospital_id
WHERE h_address = '789 Elm St, Yet Another Town USA'

- **Relational algebra:**

$$\pi_{p\_blood\_type,\ s\_number}\ \sigma_{h\_address = \text{"789 } Elm\ St,\ Yet\ Another\ Town\ USA\text{"}}\ (\text{patient} \bowtie_{patient\ .\ hospital\_id = hospital\ .\ hospital\_id}\ hospital \bowtie_{patient\ .\ hospital\_id = staff\ .\ hospital\_id}\ staff)$$

- **Tree Diagram:**

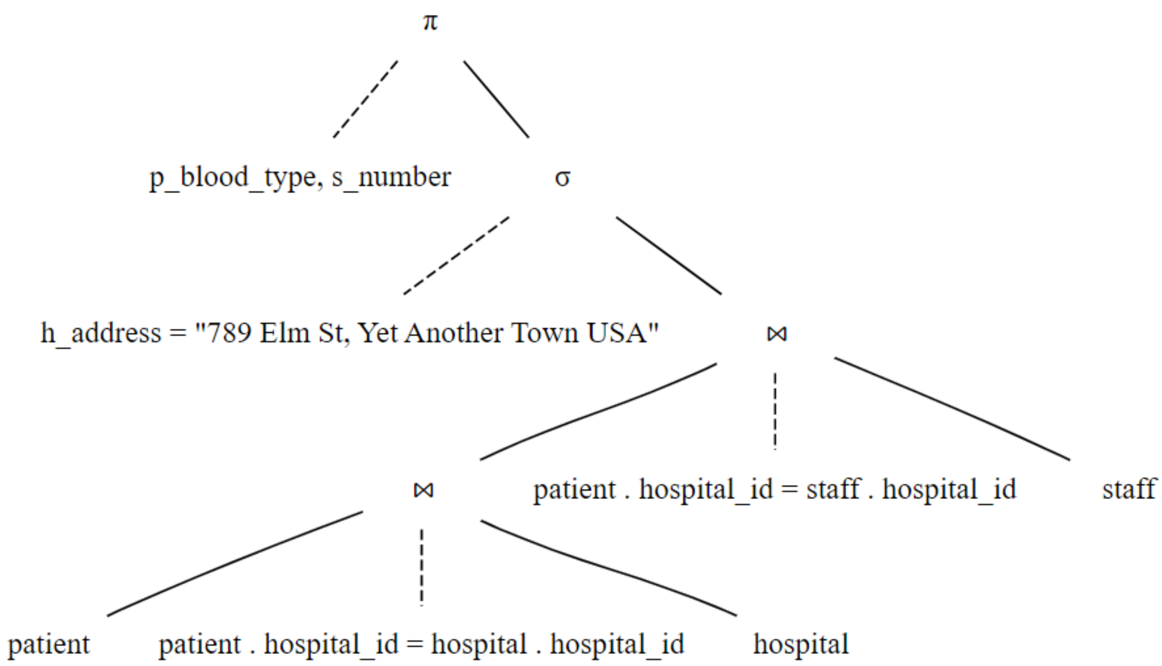

π
p_blood_type, s_number
σ
h_address = "789 Elm St, Yet Another Town USA"
⋈
⋈
patient . hospital_id = staff . hospital_id
staff
patient
patient . hospital_id = hospital . hospital_id
hospital

Figure 10. Tree Diagram in Question #5

6. Find male donors with "Allergies to pollen" that have the staffs work at "Anytown Blood bank" and "Anytown Hospital" taken care of.

- **SQL:**

```
SELECT donor_name
FROM donor INNER JOIN staff ON donor . staff_id = staff . staff_id INNER
JOIN blood_bank ON staff . blood_bank_id = blood_bank . blood_bank_id
INNER JOIN hospital ON staff . hospital_id = hospital . hospital_id
```

WHERE medical_report = 'Allergies to pollen' AND bb_name = 'Anytown Blood bank' AND h_name = 'Anytown Hospital'

- **Relational Algebra**

π donor_name
σ medical_report = "Allergies to pollen" AND bb_name = "Anytown Blood bank" AND h_name = "Anytown Hospital" (donor ⋈ donor . staff_id = staff . staff_id staff ⋈ staff . blood_bank_id = blood_bank . blood_bank_id blood_bank ⋈ staff . hospital_id = hospital . hospital_id hospital)

- **Tree Diagram**

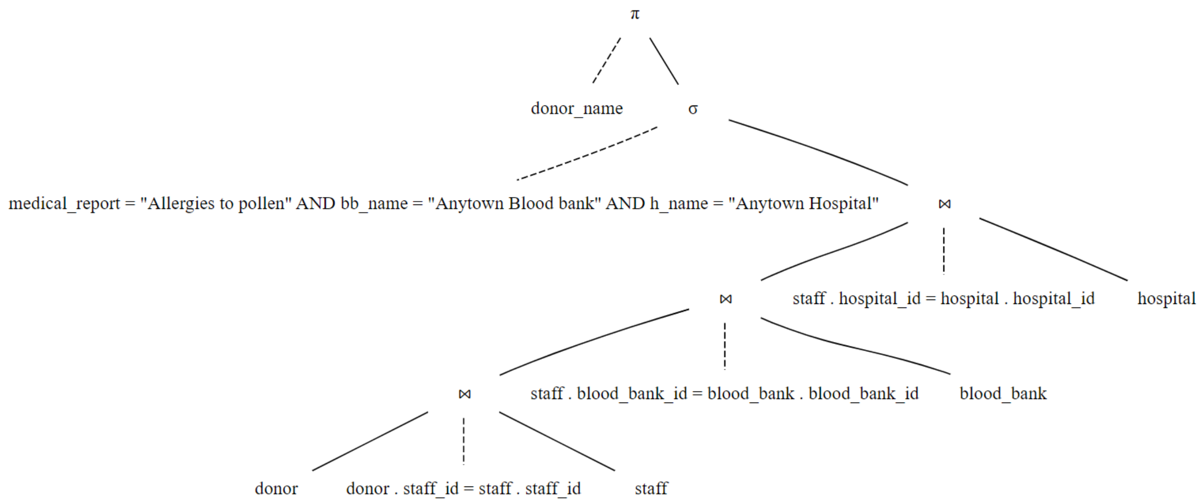

π
donor_name
σ
medical_report = "Allergies to pollen" AND bb_name = "Anytown Blood bank" AND h_name = "Anytown Hospital"
⋈
⋈
staff . hospital_id = hospital . hospital_id
hospital
⋈
staff . blood_bank_id = blood_bank . blood_bank_id
blood_bank
donor
donor . staff_id = staff . staff_id
staff

Figure 11. Tree Diagram in Question #6

XI. Application

- ❖ Home

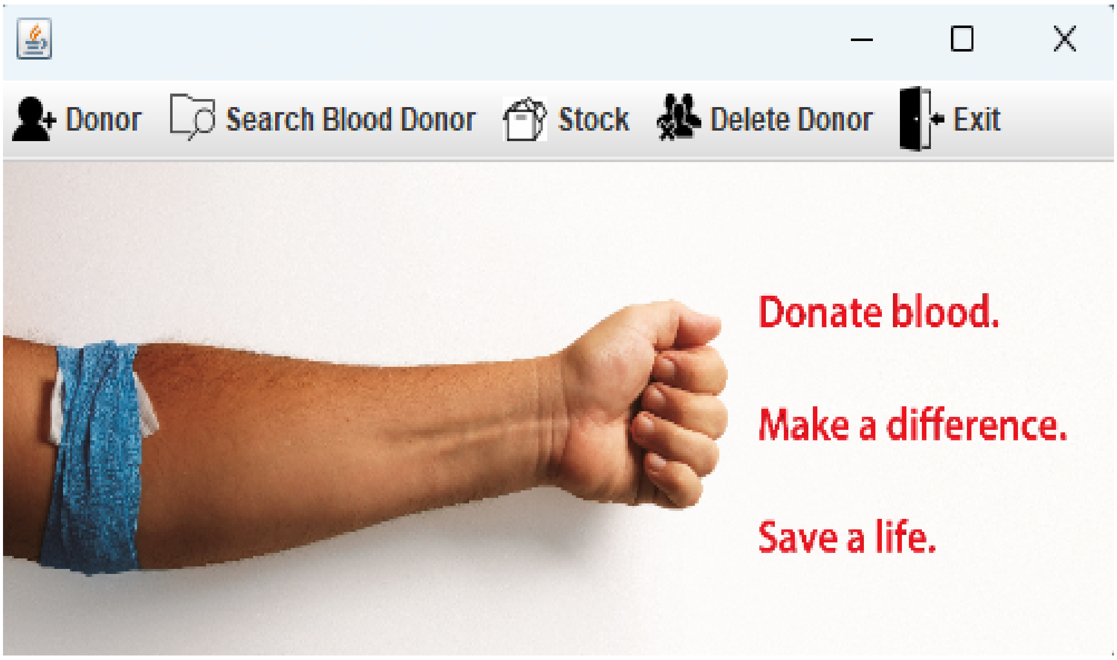
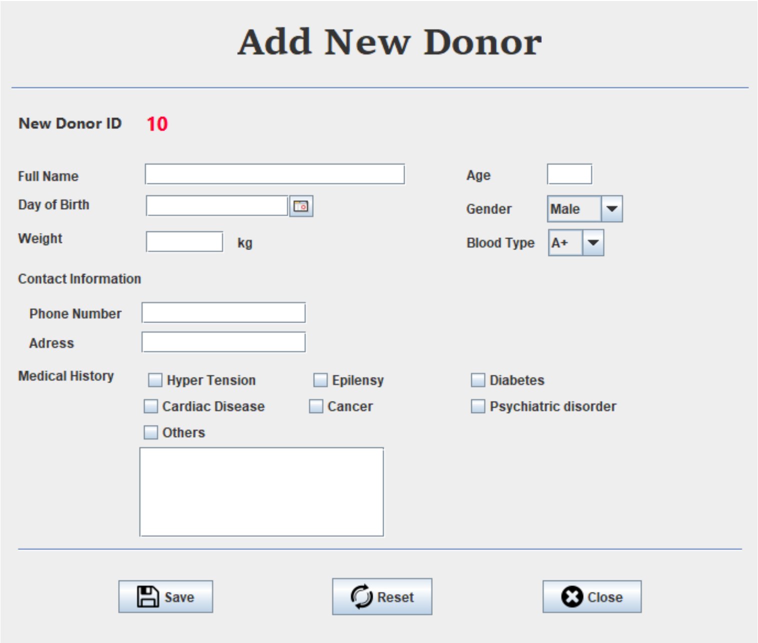


Figure 12. Application Software Interface

- ❖ Donor

➢ Add new Donor



New Donor ID 10

Gender Male

Blood Type A+

Figure 13. Add New Donor Interface

➢ Update Details

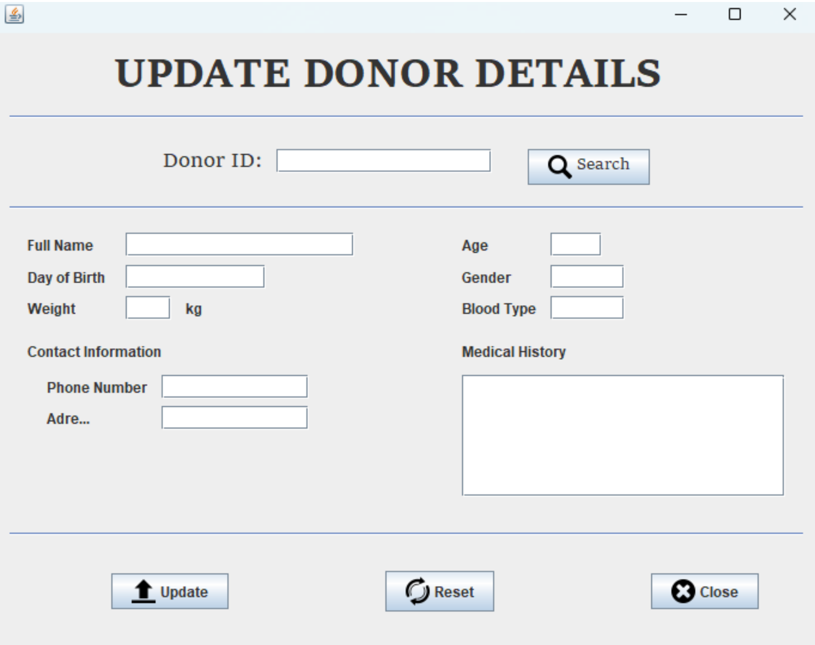
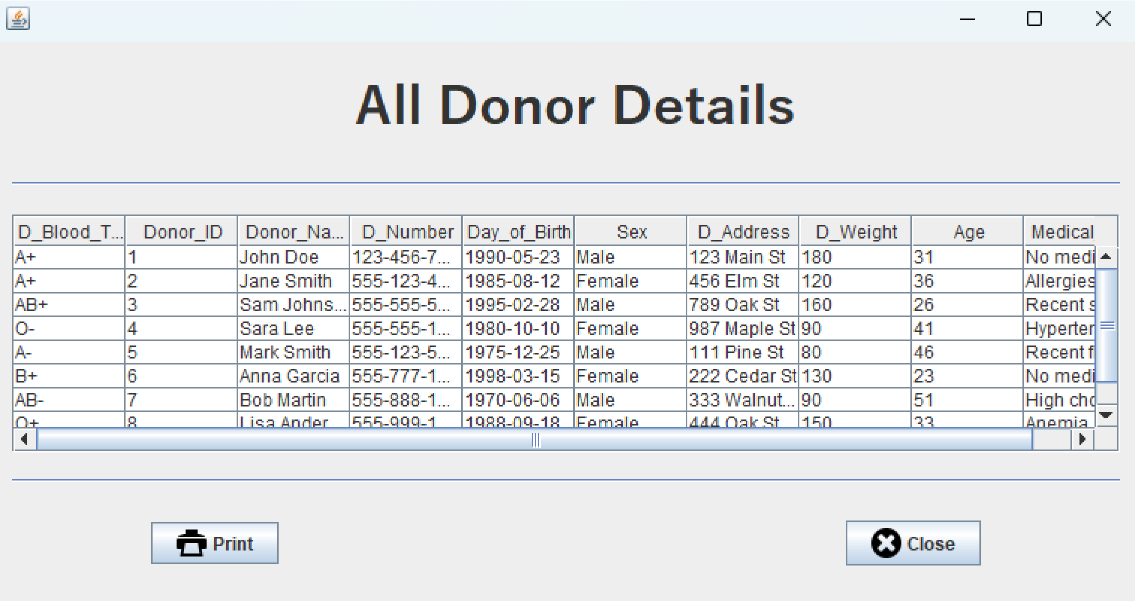


Figure 14. Update Donor Detail Interface

➢ All Donor Details



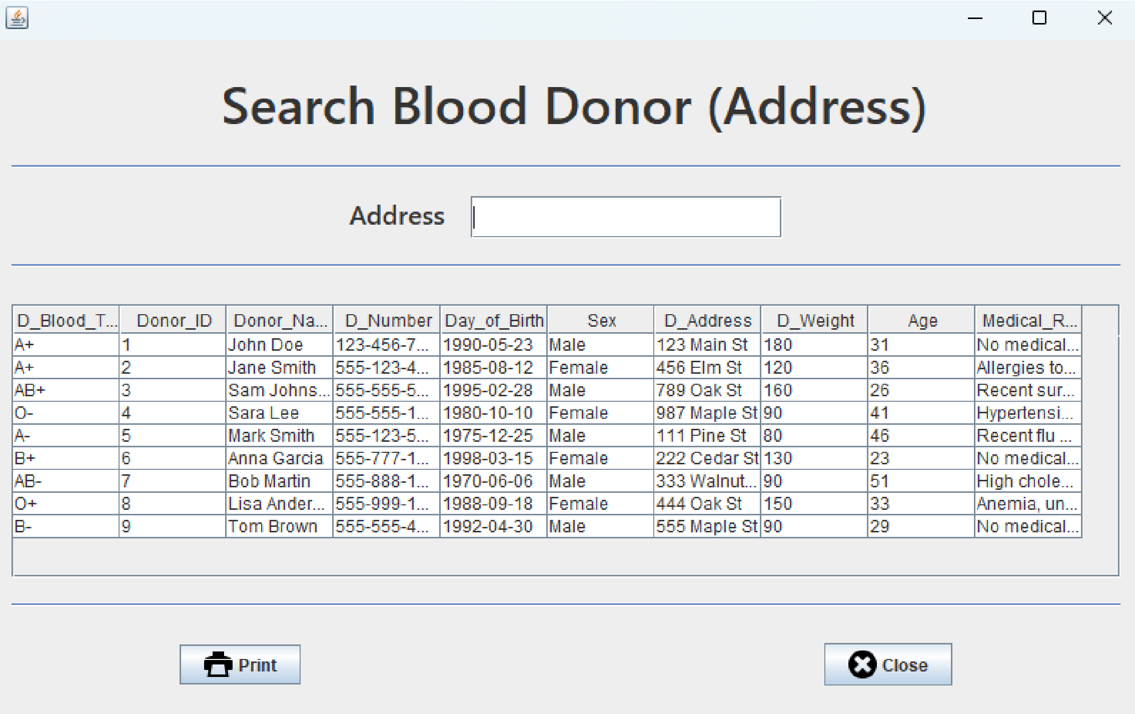
All Donor Details

D_Blood_T...	Donor_ID	Donor_Na...	D_Number	Day_of_Birth	Sex	D_Address	D_Weight	Age	Medical
A+	1	John Doe	123-456-7...	1990-05-23	Male	123 Main St	180	31	No medi
A+	2	Jane Smith	555-123-4...	1985-08-12	Female	456 Elm St	120	36	Allergies
AB+	3	Sam Johns...	555-555-5...	1995-02-28	Male	789 Oak St	160	26	Recent s
O-	4	Sara Lee	555-555-1...	1980-10-10	Female	987 Maple St	90	41	Hyperter
A-	5	Mark Smith	555-123-5...	1975-12-25	Male	111 Pine St	80	46	Recent f
B+	6	Anna Garcia	555-777-1...	1998-03-15	Female	222 Cedar St	130	23	No medi
AB-	7	Bob Martin	555-888-1...	1970-06-06	Male	333 Walnut...	90	51	High chc
O+	8	Lisa Ander	555-999-1	1988-09-18	Female	444 Oak St	150	33	Anemia

Figure 15. All Donor Details Screen

❖ Search Blood Donor

➢ Location



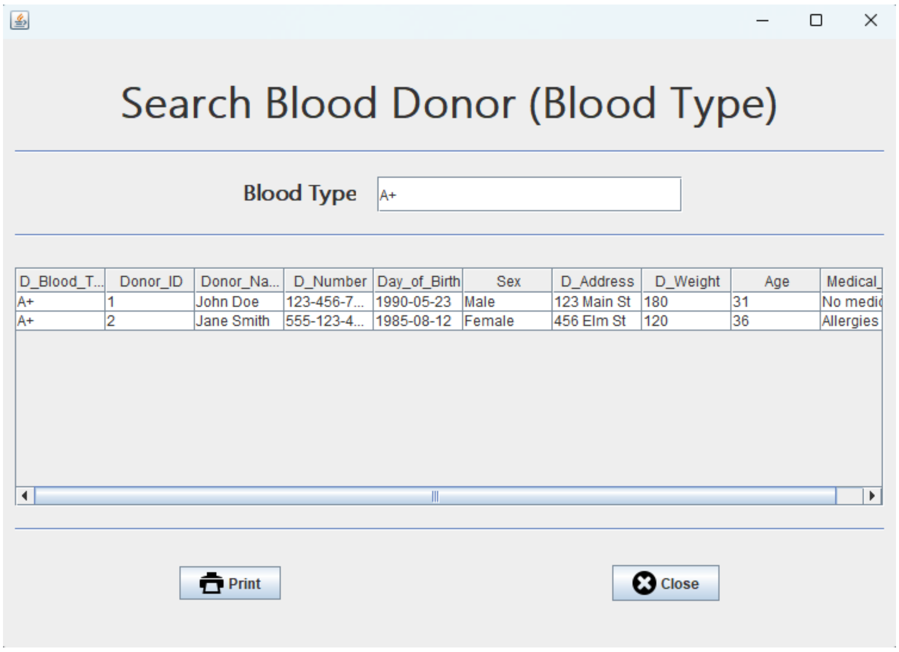
Search Blood Donor (Address)

Address

D_Blood_T...	Donor_ID	Donor_Na...	D_Number	Day_of_Birth	Sex	D_Address	D_Weight	Age	Medical_R...
A+	1	John Doe	123-456-7...	1990-05-23	Male	123 Main St	180	31	No medical...
A+	2	Jane Smith	555-123-4...	1985-08-12	Female	456 Elm St	120	36	Allergies to...
AB+	3	Sam Johns...	555-555-5...	1995-02-28	Male	789 Oak St	160	26	Recent sur...
O-	4	Sara Lee	555-555-1...	1980-10-10	Female	987 Maple St	90	41	Hypertensi...
A-	5	Mark Smith	555-123-5...	1975-12-25	Male	111 Pine St	80	46	Recent flu ...
B+	6	Anna Garcia	555-777-1...	1998-03-15	Female	222 Cedar St	130	23	No medical...
AB-	7	Bob Martin	555-888-1...	1970-06-06	Male	333 Walnut...	90	51	High chole...
O+	8	Lisa Ander...	555-999-1...	1988-09-18	Female	444 Oak St	150	33	Anemia, un...
B-	9	Tom Brown	555-555-4...	1992-04-30	Male	555 Maple St	90	29	No medical...

Figure 16. Search Blood Donor (Address) Screen

➢ Blood Group



Search Blood Donor (Blood Type)

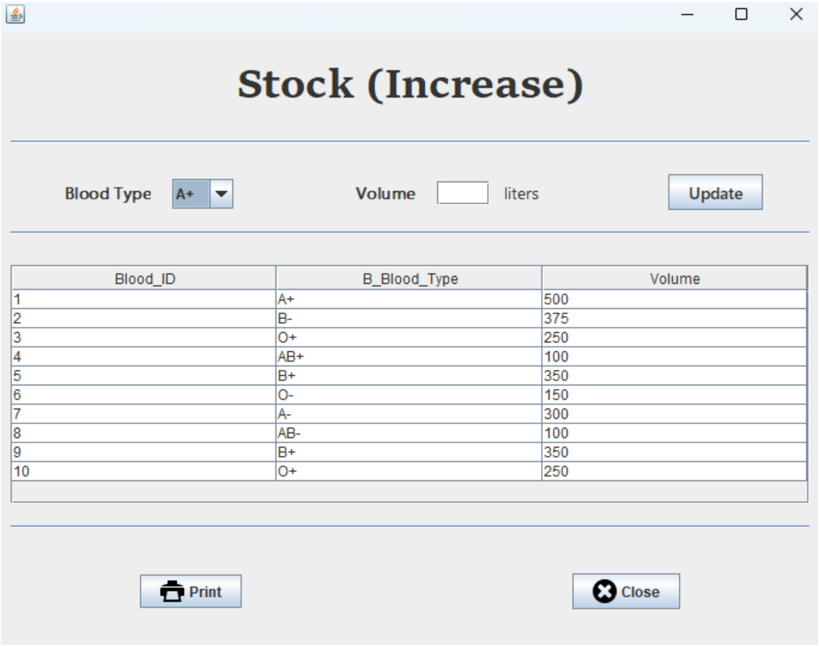
Blood Type: A+

D_Blood_T...	Donor_ID	Donor_Na...	D_Number	Day_of_Birth	Sex	D_Address	D_Weight	Age	Medical_
A+	1	John Doe	123-456-7...	1990-05-23	Male	123 Main St	180	31	No medic
A+	2	Jane Smith	555-123-4...	1985-08-12	Female	456 Elm St	120	36	Allergies

Figure 17. Search Blood Donor (Blood Type) Screen

❖ Stock

➢ Increase



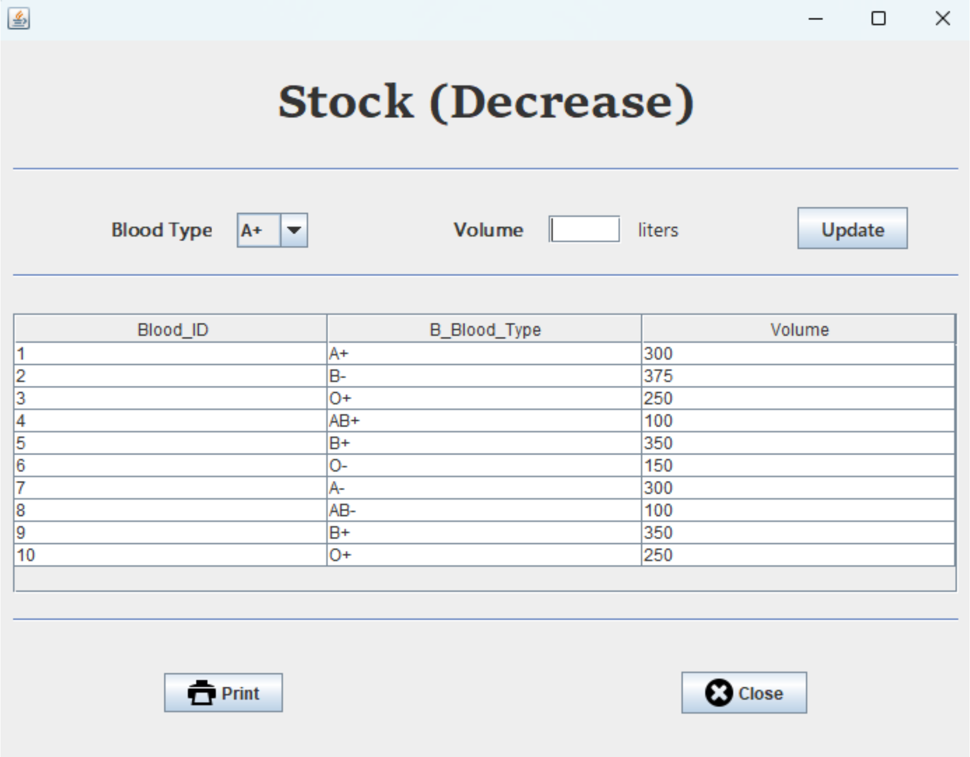
Stock (Increase)

Blood Type: A+ Volume: liters

Blood_ID	B_Blood_Type	Volume
1	A+	500
2	B-	375
3	O+	250
4	AB+	100
5	B+	350
6	O-	150
7	A-	300
8	AB-	100
9	B+	350
10	O+	250

Figure 18. Stock (Increase) Screen

- Decrease



Stock (Decrease)

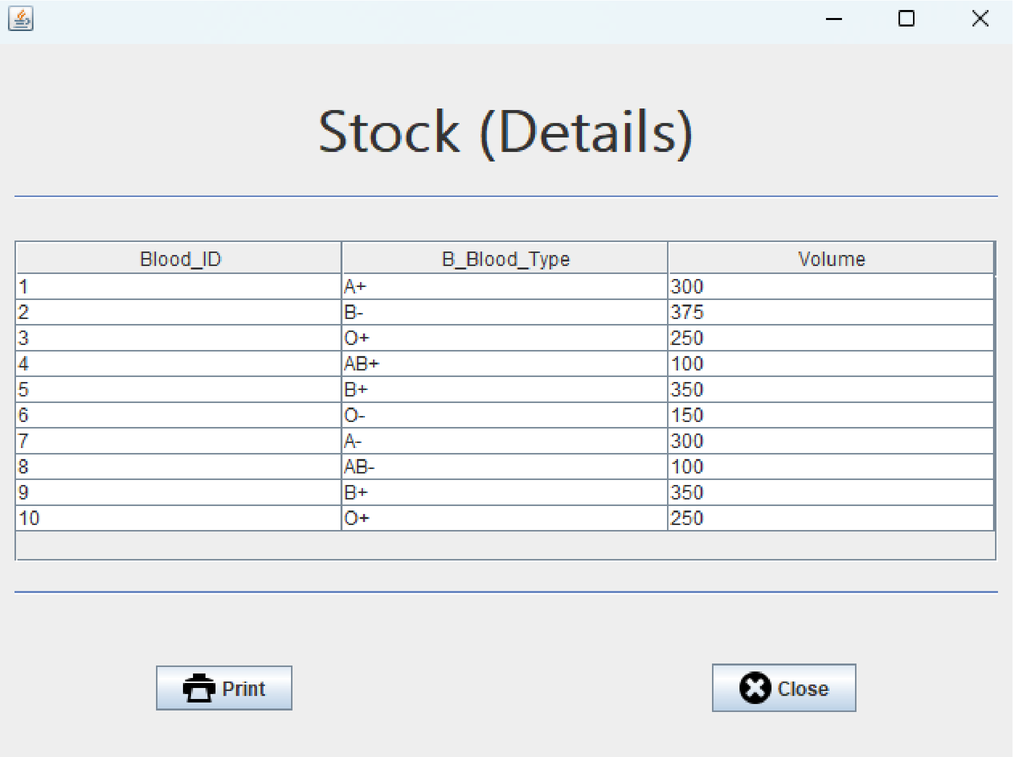
Blood Type: A+

Volume: liters

Blood_ID	B_Blood_Type	Volume
1	A+	300
2	B-	375
3	O+	250
4	AB+	100
5	B+	350
6	O-	150
7	A-	300
8	AB-	100
9	B+	350
10	O+	250

Figure 19. Stock (Decrease) Screen

- Details

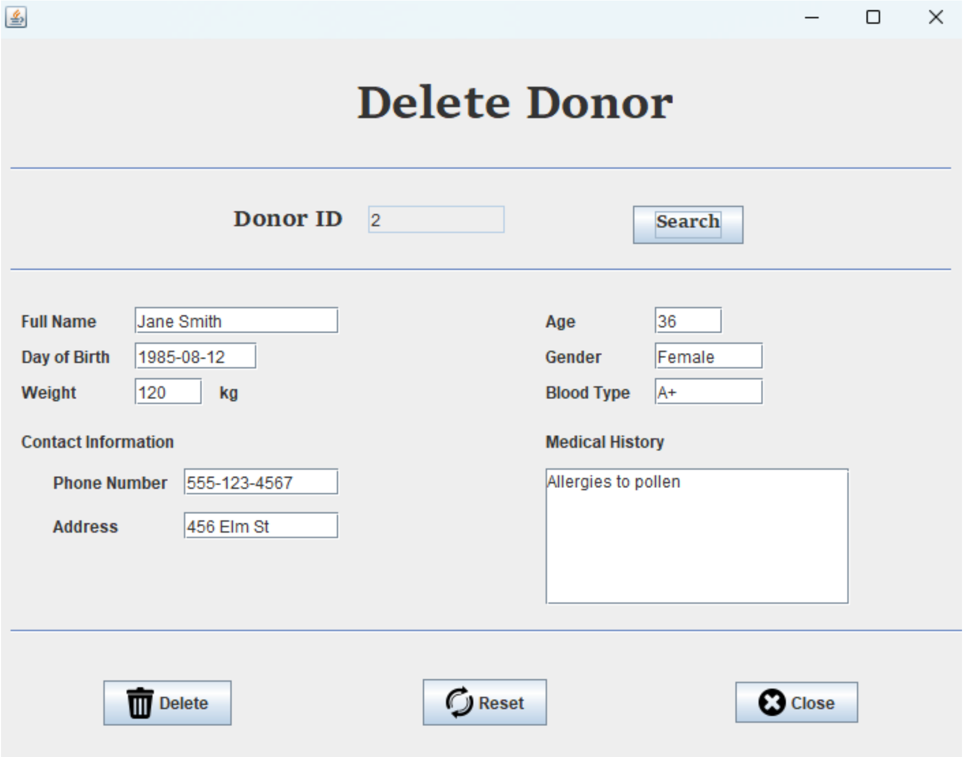


Stock (Details)

Blood_ID	B_Blood_Type	Volume
1	A+	300
2	B-	375
3	O+	250
4	AB+	100
5	B+	350
6	O-	150
7	A-	300
8	AB-	100
9	B+	350
10	O+	250

Figure 20. Stock (Details) Interface

- Delete Donor



Delete Donor

Donor ID: 2

Full Name: Jane Smith

Day of Birth: 1985-08-12

Weight: 120 kg

Age: 36

Gender: Female

Blood Type: A+

Contact Information

Phone Number: 555-123-4567

Address: 456 Elm St

Medical History: Allergies to pollen

Figure 21. Delete Donor Interface

XII. Conclusion

Through the project, we can understand in a simple and complete way how to build a project for Principles of Database Management. Gain more knowledge about the blood donation process and relationships between entities, understand more about attributes, and binary relationships, how to collect data, how to draw an entity-relationship diagram and a relationship model; and how to translate from an entity- diagram to a relational database.

XIII. Future work

For the future plan, we are going to complete and develop the project in the best direction. Moreover, make a website that makes it simple for people to register to donate blood and look up information on blood donation (address, blood donation benefits, learn about the steps to prepare before donating blood, etc). Simplify paperwork by creating an app so that everyone can keep track of their blood donation history, blood type (for those who have donated blood before), pre-donation health, blood donation records, etc.).

In order to make it simpler for staff to manage three tasks (managing the registration form in the donor object, managing the list of information in the patient in need of blood transfusion in the hospital object, managing the amount of blood in the warehouse, and receiving the list of blood orders from hospitals), I want to develop more management software for the blood bank and hospital. The scope of our project is strictly limited to developing the application for managing donor and blood warehouses. Creating new websites is essential since they are easily available and serve as a platform for attracting potential funders, easily for view and update the information.

XIV. GitHub

ITDSIU20094/Blood_Donation_Management (github.com)

XV. Reference

List of references:

1. Instant Edu Help. (2021). Gym Management System Project for Final Year Students. Retrieved March 11, 2023, from https://instanteduhelp.com/gym-management-system-project-for-final-year-students/
2. Gliffy. (n.d.). How to Draw an Entity Relationship Diagram. Retrieved March 11, 2023, from

https://www.gliffy.com/blog/how-to-draw-an-entity-relationship-diagram

3. Bui, H. (2022). ERD (Entity – Relationship Diagram) là gì – Cách thiết kế mô hình thực thể [ERD (Entity-Relationship Diagram) - What is it and how to design an entity model]. Retrieved March 11, 2023, from https://blog.cole.vn/entity-relationship-diagram-erd/
4. GeeksforGeeks. (n.d.). Relational Model in DBMS. Retrieved March 11, 2023, from https://www.geeksforgeeks.org/relational-model-in-dbms/
5. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database system concepts (6th ed.). McGraw-Hill Education. Chapter 2: Introduction to the Relational Model, pp. 37-64; Chapter 6: Database Design Using the E-R Model, pp. 241-302; Chapter 7: Relational Database Design, pp. 303-360.
6. Hoffer, J. A., Ramesh, V., & Topi, H. (2019). Modern database management (12th ed.). Pearson Education. Chapter 4: Logical Database Design and the Relational Model, pp. 191-241. Chapter 5: Physical Database Design and Performance, pp. 242-296.
7. Elmasri, R., & Navathe, S. B. (2010). Fundamentals of database systems (7th ed.,). Pearson Education. Chapter 3: Data modeling using the entity-relationship (ER) model, pp. 59 - 105. Chapter 4: The Enhanced Entity–Relationship (EER) Model, pp. 107 - 146. Retrieved from https://archive.org/details/dbms-by-elmasri-and-navathe/mode/2up
8. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). Chapter 4: High-Level Database Models, pp.125-200. In Database Systems: The Complete Book (2nd ed., pp. 121-155). Pearson Education.
9. Ramakrishnan, R., & Gehrke, J. (2003). The Entity-Relationship Model, pp. 57-99. In Database Management Systems (3rd ed.). McGraw-Hill, from https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxoYW5ub2RncmFwaHxneDoyNWJmY2MxZWMyN2Y2OTY