Introduction to Database: Final Report

Blood Bank Project

Course Number: CCCS215 (26303)

Themar Yassir

In today's fast-paced medical landscape, the efficient management of blood resources stands as a critical pillar in ensuring timely and adequate healthcare services. Blood banks serve as life-saving repositories, and their operational effectiveness relies heavily on sophisticated database systems. This report delves into the design, implementation, and benefits of a state-of-the-art database program tailored specifically for blood banks. The intricate nature of blood management necessitates a robust, secure, and user-friendly database solution, which plays a pivotal role in streamlining processes, enhancing traceability, and ultimately saving lives. This report aims to explore the functionalities, advantages, and potential impact of this dedicated database program within the context of blood bank operations.

In this report, we'll show you our special database program made just for blood banks. First, we'll explain how everything fits together using diagrams called Entity-Relationship (E-R) diagrams. Then, we'll talk about the technical side using SQL (Structured Query Language), which helps us build the actual parts of the database. This detailed walkthrough will explain how our database is set up, specifically to suit the needs of managing a blood bank. We'll show you how our program goes from ideas to real-life use in the database, so you can understand how well it works and how it makes managing blood bank tasks easier and more efficient.

1. E-R/EE-R Diagram

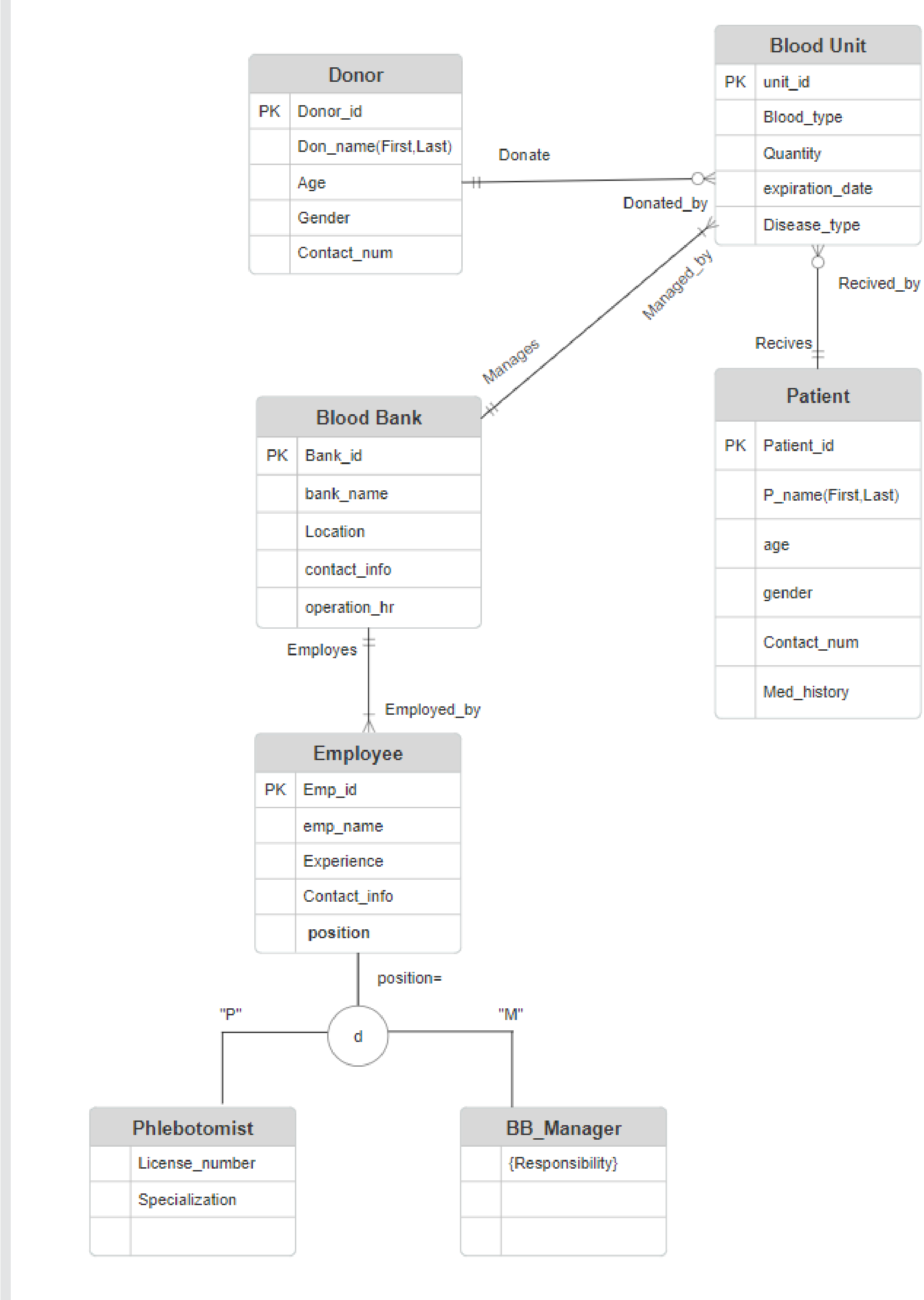


Figure 1: E-R Diagram demonstrating Blood Bank

1.1 Entities of E-R/EE-R

The diagram illustrates the interconnected entities crucial to the effective functioning of blood bank system:

1. **Donor:** This entity represents individuals who generously provide blood donations to the blood bank. Donors play a vital role in the process of maintaining a sufficient blood supply for patients in need.

2. **Blood Unit:** This entity refers to the individual units of blood collected from donors. Each blood unit undergoes testing and storage within the blood bank before being distributed for transfusions to patients.

3. **Patient:** Individuals in need of blood transfusions or related medical care, for whom suitable blood units are provided by the blood bank.

4. **Blood Bank:** The central entity governing the management, storage, and distribution of blood units to cater to patient needs.

5. **Employee:** A superclass entity encompassing the personnel engaged in various roles within the blood bank system.

   5.a. **Phlebotomist:** Specialized professionals trained to safely collect blood from donors. They ensure the hygienic extraction of blood units.

   5.b. **BB Manager** (Blood Bank Manager): A managerial role overseeing the blood bank's operations, responsible for resource management, regulatory compliance, and implementing effective strategies.

The diagram showcases the relationships between these entities, highlighting the flow and interaction within the blood bank system. The Employee entity serves as the superclass, encompassing broader roles within the blood bank structure, including both Phlebotomist and BB Manager. This hierarchical relationship signifies that both the Phlebotomist and BB Manager roles inherit attributes and responsibilities from the Employee superclass, demonstrating their specialized roles within the blood bank's workforce.

1.2 System Analysis:

One donor can donate multiple blood units, and each blood unit is donated by one donor. Each donor instance will be assigned a unique donor ID, donor name, age, gender, and contact number. And every blood unit has a unique ID, Blood type, Quantity, Expiration date, and Disease type.

A patient may receive multiple blood units if needed, and each blood unit is received by only one patient. Every patient instance must contain a unique patient ID, Name, Age, Gender, Contact information, and Medical history.

A blood bank can manage more than one blood unit, and every blood unit is managed by one bank. Each blood bank contains a unique bank ID, Name, Location, Contact information, and Operational hours.

A blood bank can employ many employees, yet each employee must be employed by only one blood bank. An Employee must have their own position, either BB Manager, a Phlebotomist, or Other. An employee must have their own unique ID number, Name, Experience, Contact information, and Position. Each employee must have their own position, and the positions can be a BB Manager or a Phlebotomist. a BB Manager instance must have all Employee attributes as well as responsibilities, and a Phlebotomist must also have all Employee attributes along with their License Number and Specialization.

2. Normalization

Normalizing the Entity-Relationship (ER) diagram stands as a fundamental step in refining the architecture of the blood bank's database system, poised to improve its overall functionality. At its core, the ER diagram serves as a visual map outlining the key entities—donors, blood units, patients, blood bank, and employee(including the phlebotomist and the BB manager)—interconnected through defined relationships. However, a robust database system demands more than just a graphical representation.

Normalization steps in as the intricate process of fine-tuning this representation to eliminate inefficiencies and enhance the database's performance. Within the blood bank context, where the management of diverse data—ranging from donor information to intricate inventory details—is paramount, normalization becomes imperative. Reducing redundancies, minimizing anomalies, and optimizing the arrangement of data not only heightens operational efficiency but also ensures accuracy and consistency in information retrieval and maintenance.

DONOR

Donor_id	firstName	lastName	age	gender	Contact_Num

BLOOD_UNIT

Unit_id	bloodType	quantity	expirationDate	diseaseType	Donor_id	Bank_id	Patient_id

BLOOD_BANK

Bank_id	bankName	bankLocation	contact_Info	OperationHours

PATIENT

Patient_id	firstName	lastName	age	gender	Contact_Num	medhistory

EMPLOYEE

Emp_id	empName	experience	contactInfo	position	Bank_id

PHLEBOTOMIST

P_Emp_id	licenseNumber	specialization

BB_MANAGER

M_Emp_id

M_RESPONSIBILITY

M_Emp_id	Responsibility

Here are all the logical schemas from our E-R/EE-R diagram before normalization, and here are all the steps we took to normalize it so it can be ready for usage in physical schema:

INTRODUCTION TO DATABASE: FINAL REPORT

Normalization:

-1NF:

- There are no repeating groups
- A unique key has been identified for each relation
- All attributes are functionally dependent on all or part of the key

DONOR(Donor_id, firstName, lastName, age, gender, contact_Num)

BLOOD_UNIT(Unit_id, bloodType, quantity, expirationDate, diseaseType, Donor_id#, Bank_id#, Patient_id#)

BLOOD_BANK(Bank_id, bankName, bankLocation, contact_Info, operationHours)

PATIENT(Patient_id, firstName, lastName, age, gender, contact_Num, medHistory)

EMPLOYEE(Emp_id, empName, experience, contactInfo, position ,Bank_id#)

PHLEBOTOMIST (P_Emp_id, licenseNumber, specialization)

BB_MANAGER(M_Emp_id)

M_RESPONSIBILITY(M_Emp_id, Responsibility)

-2NF:

- Already in 2NF, as there's no partial dependencies.

-3NF:

- There's transitive dependencies in BLOOD_UNIT as diseaseType depends on bloodType

DONOR(Donor_id, firstName, lastName, age, gender, contact_Num)

BLOOD_UNIT(Unit_id, bloodType, quantity, expirationDate, Donor_id#, Bank_id#, Patient_id#)

BLOOD_DISEASE(bloodType, diseaseType)

INTRODUCTION TO DATABASE: FINAL REPORT

BLOOD_BANK(<u>Bank_id</u>, bankName, bankLocation, contact_tInfo, operation_hr)

PATIENT(<u>Patient_id,</u> firstName, lastName, age, gender, contact_Num, medHistory)

EMPLOYEE(<u>Emp_id</u>, empName, experience, contact_tInfo, position,Bank_id#)

PHLEBOTOMIST (<u>P_Emp_id</u>, licenseNumber, specialization)

BB_MANAGER(<u>M_Emp_id</u>)

M_RESPONSIBILITY(<u>M_Emp_id</u>, <u>Responsibility</u>).

After normalizing our schemas, here are the Functional Dependencies:

Donor_id	→	First Name,last Name ,Age,Gender, contact_num
Unit_id	→	Blood_type,Quantity, expiration_date
BloodType	→	diseaseType
Patient_id	→	First Name,Last Name,age,gender, contact_num,Med_history
Bank_id	→	bank_name,Location, contact_info,operation_hr
Emp_id	→	emp_name, Experience,contact_info,position

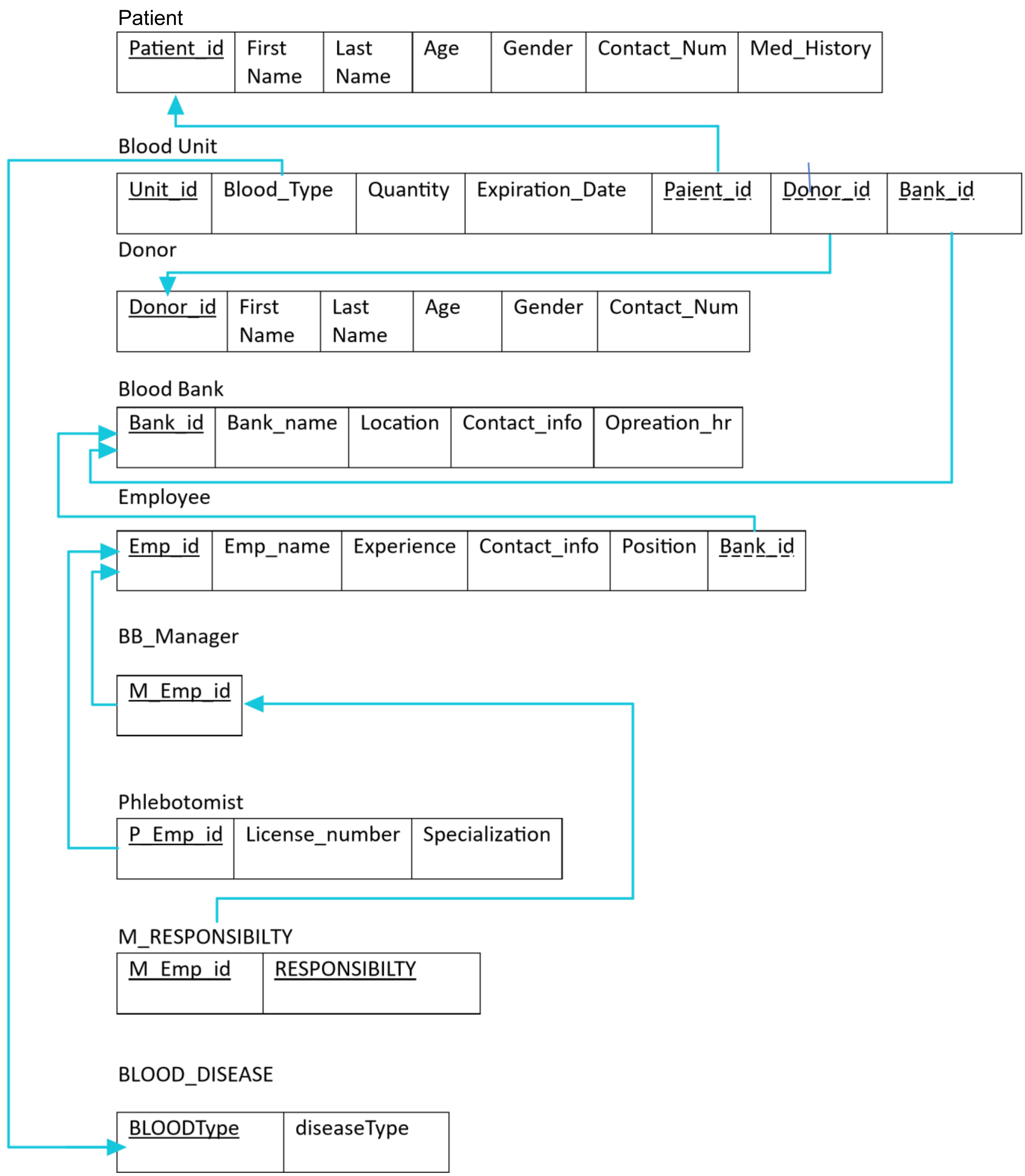


Figure 2: Logical schema after normalization

3. Physical Implementation: SQL

Designing and implementing a Blood Bank Database involves translating the logical schema into a physical implementation using SQL (Structured Query Language). This process includes creating tables, defining relationships, and establishing constraints to ensure data integrity and efficient query handling.

The SQL-based physical implementation for a Blood Bank Database transforms the conceptual design into tangible tables and relationships within a relational database management system (RDBMS). SQL provides a standardized language to interact with the database and execute commands for creating, manipulating, and querying data.

The implementation begins with the creation of tables, defining their structure, including columns, data types, constraints, and relationships. Foreign keys are employed to establish links between tables, ensuring referential integrity across different entities. Indexes may also be utilized to optimize query performance by facilitating faster data retrieval.

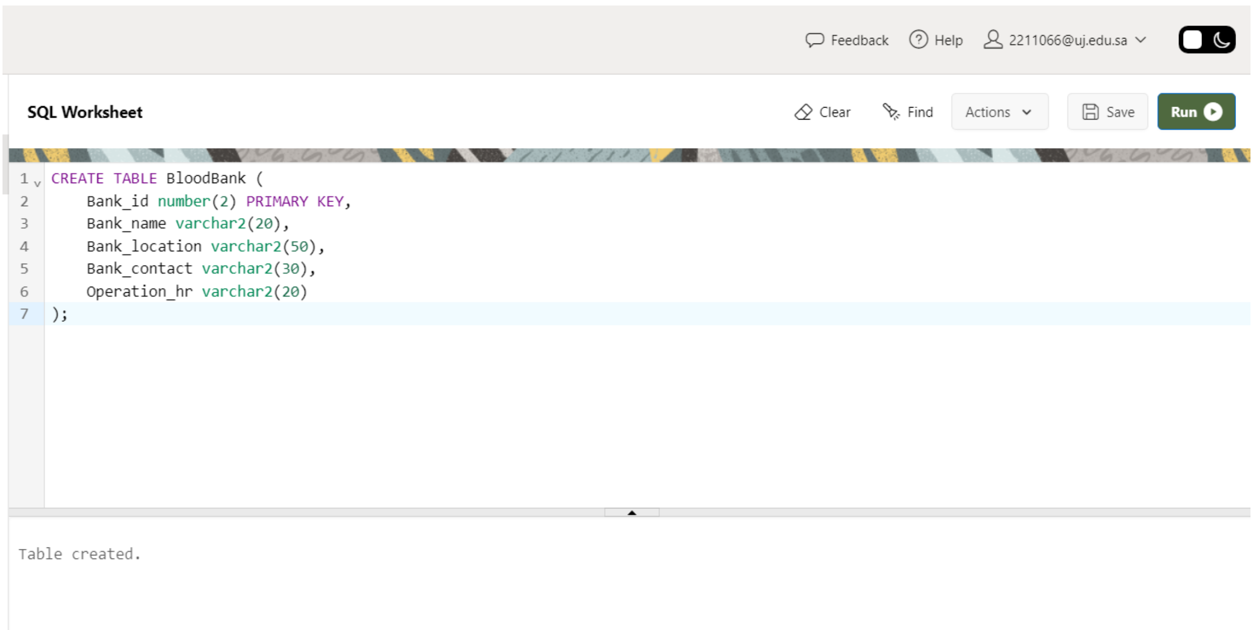
Constraints such as primary keys, foreign keys, unique constraints, and check constraints play a crucial role in maintaining data accuracy and consistency. They enforce rules on the data to prevent invalid entries and maintain the relationships defined in the logical schema.

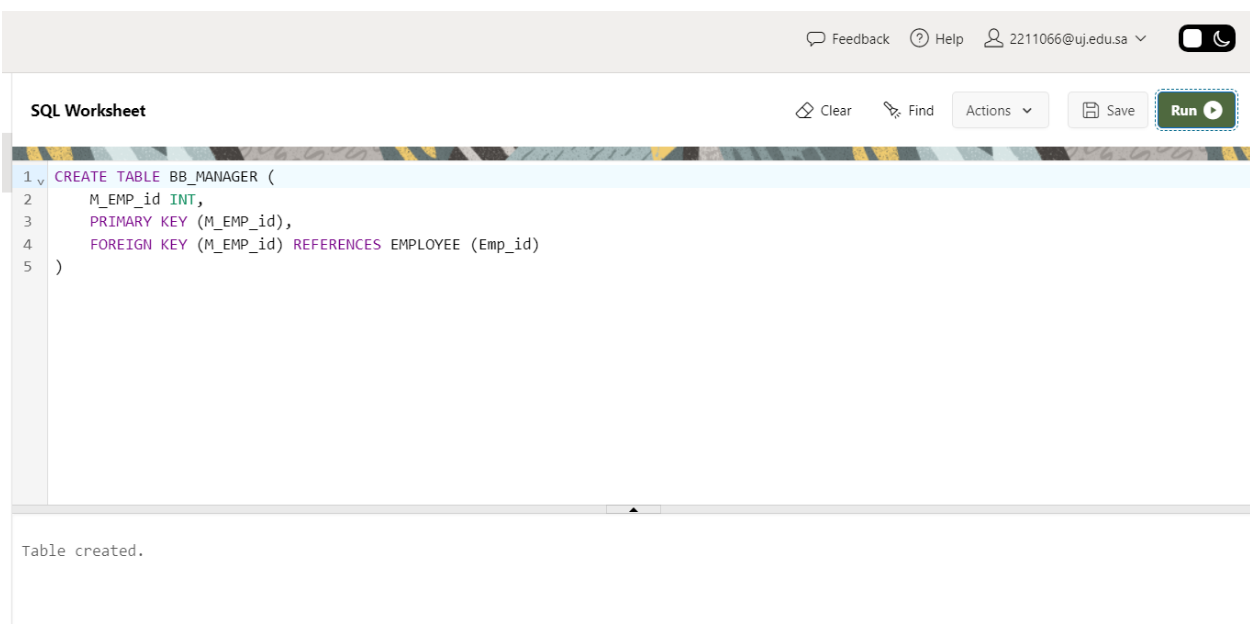
As well as stored procedures that might be incorporated to enhance functionality and automate certain database operations. Stored procedures in a Blood Bank Database play a pivotal role in streamlining database operations and improving efficiency. They encapsulate a set of SQL statements that can be executed as a single unit, enhancing functionality, security, and maintenance within the database.

Next, we'll demonstrate the practical application of these schemas in the physical database environment:

3.1 Creating Tables

In this section, we will demonstrate every table created in our database.





INTRODUCTION TO DATABASE: FINAL REPORT

SQL Worksheet

```
CREATE TABLE PATIENT(
    Patient_id Number (2) PRIMARY KEY,firstName VARCHAR(50),
    lastName VARCHAR(50),age INT,gender VARCHAR(10),
    contact_Num VARCHAR(15),medHistoty VARCHAR(200));
```

SQL Worksheet

```
CREATE TABLE Donor (
    id INT PRIMARY KEY,
    firstName VARCHAR(50),
    lastName VARCHAR(50),
    age INT,
    gender VARCHAR(10),
    contac_tNum VARCHAR(15)
)
```

Table created.

SQL Worksheet

```
CREATE TABLE BLOOD_UNIT (
    Unit_id NUMBER(8),
    bloodType VARCHAR(3),
    quantity VARCHAR(10),
    expirationdate DATE,
    Donor_id NUMBER(22),
    BANK_ID NUMBER(22),
    PATIENT_ID NUMBER(22),
    PRIMARY KEY (Unit_id, bloodType),
    FOREIGN KEY (Donor_id) REFERENCES DONOR(ID),
    FOREIGN KEY (BANK_ID) REFERENCES BLOODBANK(BANK_ID),
    FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(PATIENT_ID)
)
```

Table created.

Feedback Help 2211066@uj.edu.sa

SQL Worksheet Clear Find Actions Save Run

```sql
CREATE TABLE EMPLOYEE (
    Emp_id number(2),
    emp_name varchar2(15) not null,
    experience varchar2(20),
    contact_info number(10) not null,
    position varchar2(15),
    bank_id number(2),
    CONSTRAINT emp_pk PRIMARY KEY (Emp_id),
    CONSTRAINT bank_fk FOREIGN KEY (bank_id) REFERENCES BloodBank (Bank_id)
);
```

```
Table created.
```

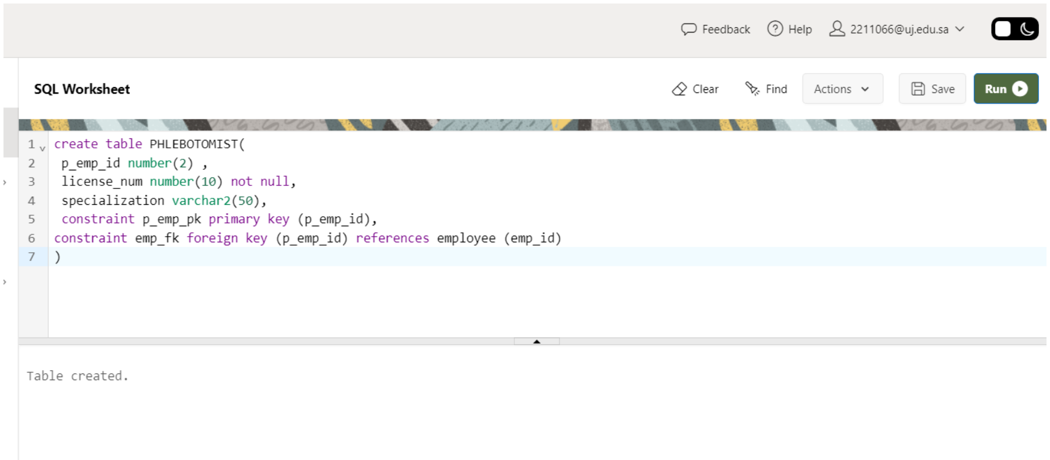
Feedback Help 2211066@uj.edu.sa

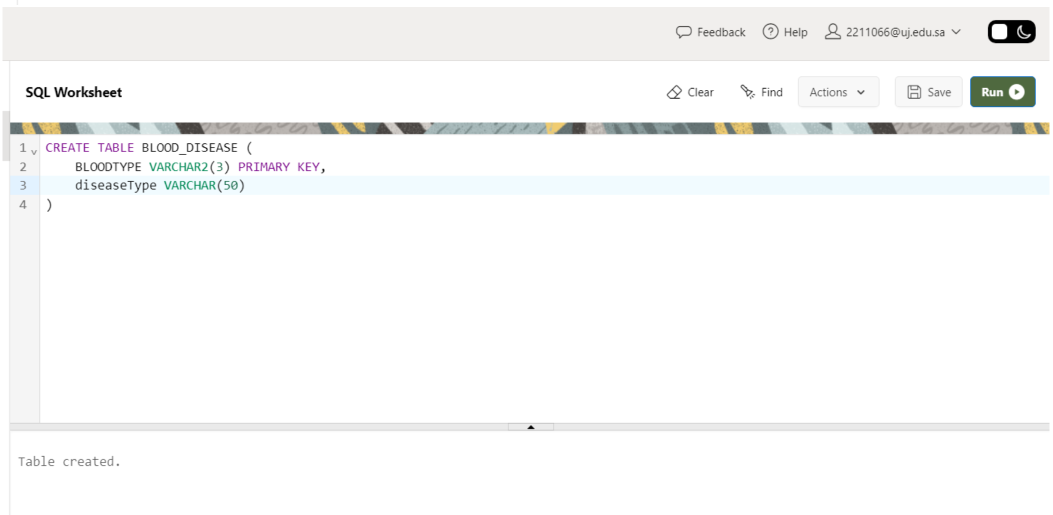
SQL Worksheet Clear Find Actions Save Run

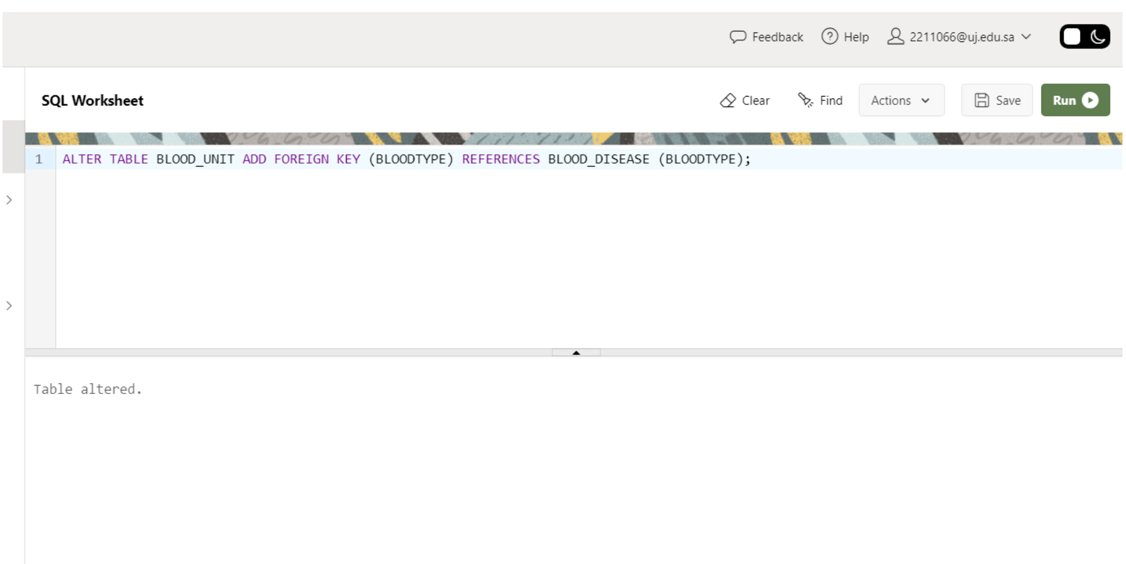
```sql
CREATE TABLE Emp_Responsibility (
    M_Emp_ID NUMBER(8),
    Responsibility VARCHAR(50),
    PRIMARY KEY (M_Emp_ID, Responsibility),
    FOREIGN KEY (M_Emp_ID) REFERENCES BB_Manager (M_Emp_ID)
)
```

```
Table created.
```

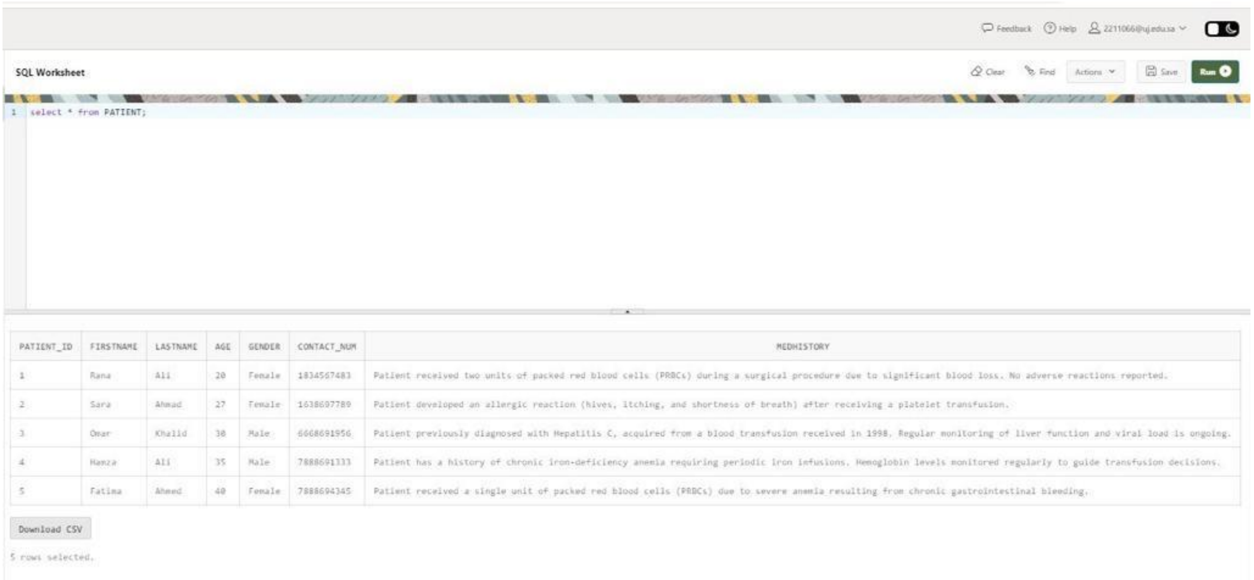
INTRODUCTION TO DATABASE: FINAL REPORT







3.2 Data Inserted into Schemas

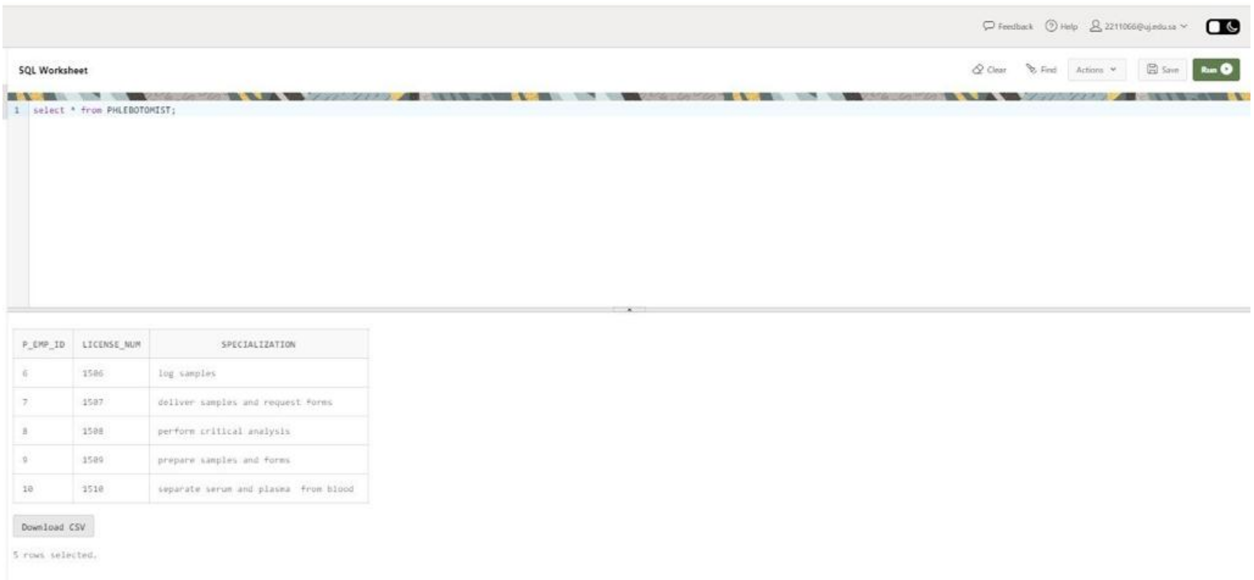



```sql
select * from PATIENT;
```

PATIENT_ID	FIRSTNAME	LASTNAME	AGE	GENDER	CONTACT_NUM	MEDHISTORY
1	Rana	Ali	20	Female	1834567483	Patient received two units of packed red blood cells (PRBCs) during a surgical procedure due to significant blood loss. No adverse reactions reported.
2	Sara	Ahmad	27	Female	1638697789	Patient developed an allergic reaction (hives, itching, and shortness of breath) after receiving a platelet transfusion.
3	Omar	Khalid	30	Male	6668691956	Patient previously diagnosed with Hepatitis C, acquired from a blood transfusion received in 1998. Regular monitoring of liver function and viral load is ongoing.
4	Hamza	Ali	35	Male	7888691333	Patient has a history of chronic iron-deficiency anemia requiring periodic iron infusions. Hemoglobin levels monitored regularly to guide transfusion decisions.
5	Fatima	Ahmed	40	Female	7888694345	Patient received a single unit of packed red blood cells (PRBCs) due to severe anemia resulting from chronic gastrointestinal bleeding.

Download CSV

5 rows selected.




```sql
select * from PHLEBOTOMIST;
```

P_EMP_ID	LICENSE_NUM	SPECIALIZATION
6	1506	log samples
7	1507	deliver samples and request forms
8	1508	perform critical analysis
9	1509	prepare samples and forms
10	1510	separate serum and plasma from blood

Download CSV

5 rows selected.

```
select * from emp_responsibility;
```

M_EMP_ID	RESPONSIBILITY
1	Obtains blood specimens
2	Maintains specimen integrity
3	prepare samples and forms
4	Verifies patient identification
5	prepare samples and forms

Download CSV

5 rows selected.

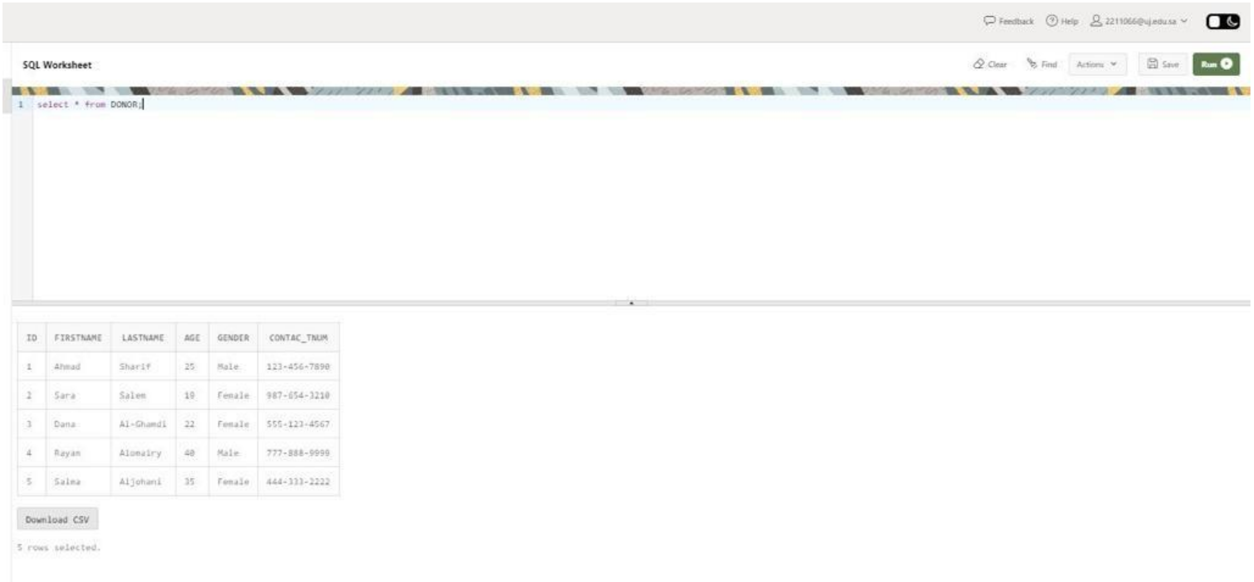
Feedback Help 2211066@uj.edu.sa

SQL Worksheet

Clear Find Actions Save Run

```
select * from EMPLOYEE;
```

EMP_ID	EMP_NAME	EXPERIENCE	CONTACT_INFO	POSITION	BANK_ID
1	jay	volunteering as EMT	135673359	BB_Manager	1
2	lily	pharmacy techT	345623411	BB_Manager	1
3	john	working as physician	547896573	BB_Manager	1
4	Faisal	volunteering as EMT	456789432	BB_Manager	3
5	Thelma	pharmacy tech	678954325	Phlebotomist	4
6	Mohamed	pharmacy tech	45678943	Phlebotomist	4
7	Jain	volunteering	342213479	Phlebotomist	4
8	fahad	working in hospital	114589056	BB_Manager	1
9	yassir	volunteering	984456874	Phlebotomist	5
10	Kevin	volunteering	567342214	Phlebotomist	4



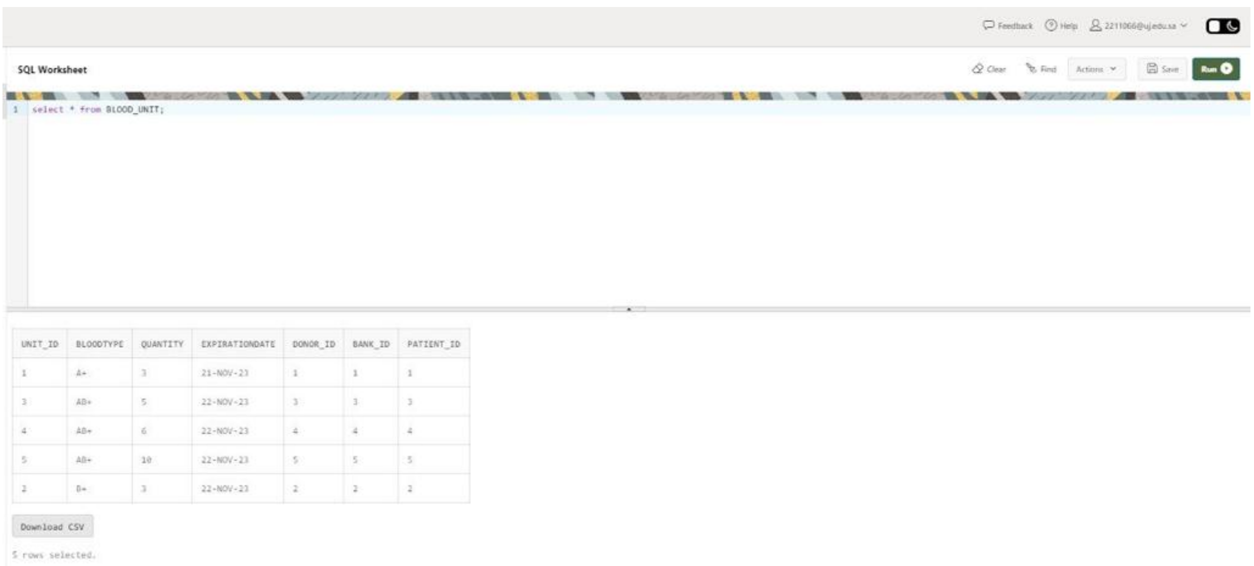
SQL Worksheet

```
1 select * from DONOR;
```

ID	FIRSTNAME	LASTNAME	AGE	GENDER	CONTAC_TNUM
1	Ahmad	Sharif	25	Male	123-456-7890
2	Sara	Salem	18	Female	987-654-3210
3	Dana	Al-Ghamdi	22	Female	555-123-4567
4	Rayan	Alomairy	40	Male	777-888-9999
5	Salma	Aljohani	35	Female	444-333-2222

Download CSV

5 rows selected.



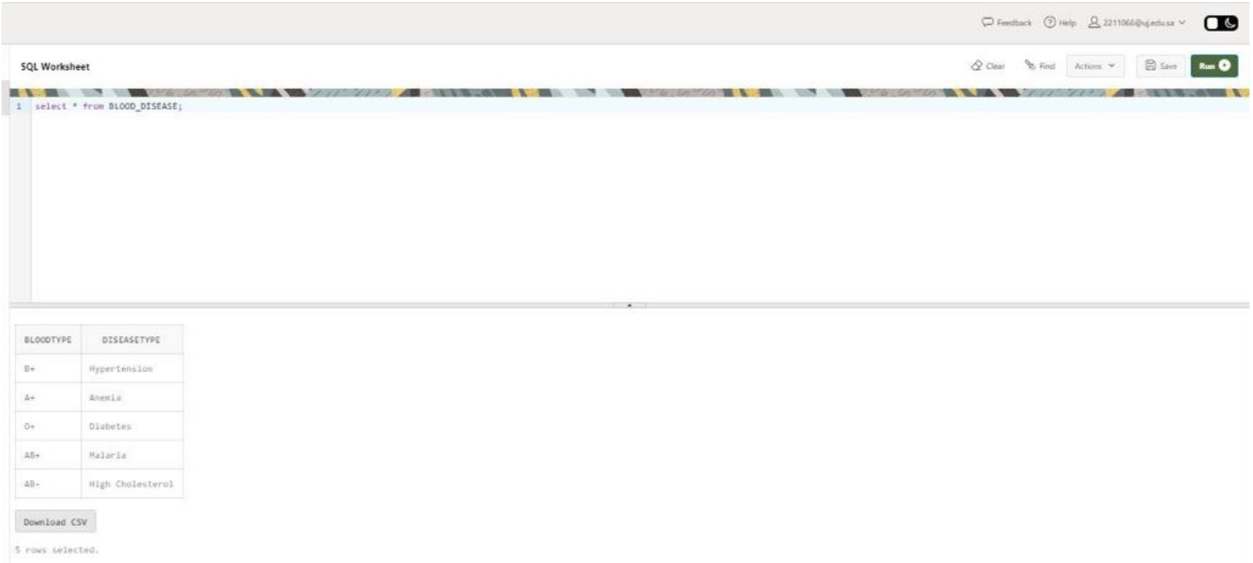
SQL Worksheet

```
1 select * from BLOOD_UNIT;
```

UNIT_ID	BLOODTYPE	QUANTITY	EXPIRATIONDATE	DONOR_ID	BANK_ID	PATIENT_ID
1	A+	3	21-NOV-23	1	1	1
3	AB+	5	22-NOV-23	3	3	3
4	AB+	6	22-NOV-23	4	4	4
5	AB+	10	22-NOV-23	5	5	5
2	B+	3	22-NOV-23	2	2	2

Download CSV

5 rows selected.



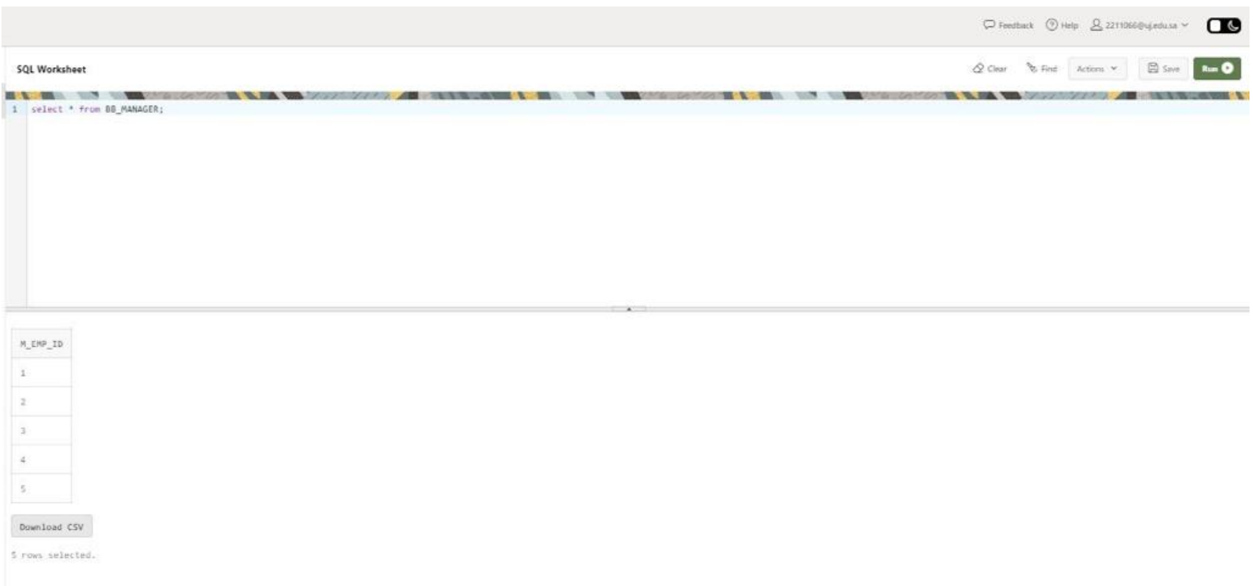
SQL Worksheet

```sql
select * from BLOOD_DISEASE;
```

BLOODTYPE	DISEASETYPE
B+	Hypertension
A+	Anemia
O+	Diabetes
AB+	Malaria
AB-	High Cholesterol

Download CSV

5 rows selected.



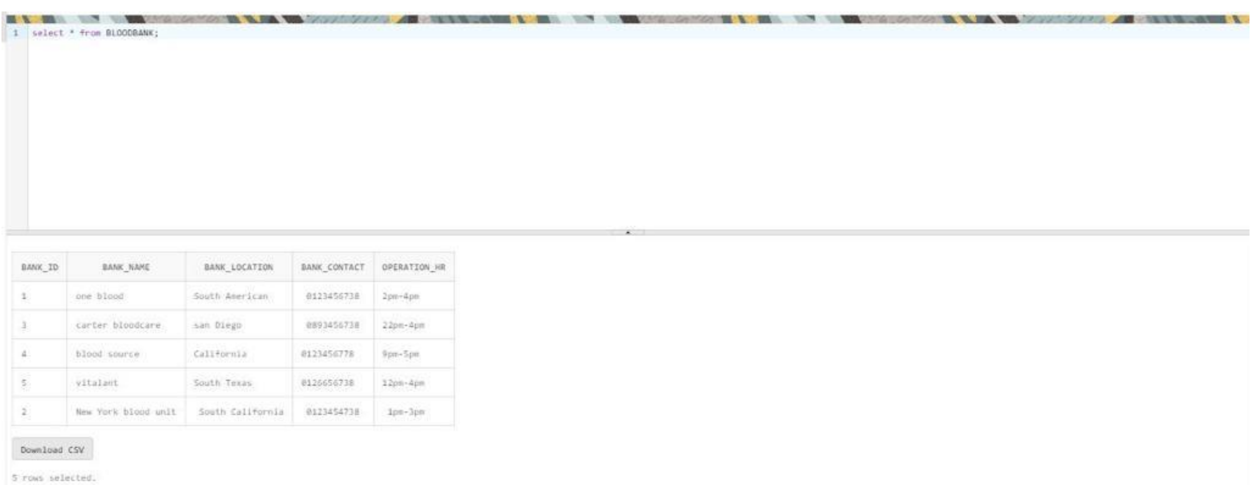
SQL Worksheet

```sql
select * from BB_MANAGER;
```

M_EMP_ID
1
2
3
4
5

Download CSV

5 rows selected.



```sql
select * from BLOODBANK;
```

BANK_ID	BANK_NAME	BANK_LOCATION	BANK_CONTACT	OPERATION_HR
1	one blood	South American	0123456738	2pm-4pm
3	carter bloodcare	san Diego	0893456738	22pm-4pm
4	blood source	California	0123456778	9pm-5pm
5	vitalant	South Texas	0126656738	12pm-4pm
2	New York blood unit	South California	0123454738	1pm-3pm

Download CSV

5 rows selected.

3.3 Stored Procedures

SQL Worksheet

```sql
CREATE OR REPLACE PROCEDURE Employee_Bank(
  R_BankID  IN EMPLOYEE.BANK_ID%TYPE,
  R_Position IN EMPLOYEE.POSITION%TYPE
)
IS
  CURSOR EB IS
    SELECT R.EMP_ID, EMP_NAME, EXPERIENCE, BANK_ID, POSITION
    FROM EMPLOYEE R
    WHERE R.BANK_ID = R_BankID AND R.POSITION = R_Position;
BEGIN
  FOR rec IN EB LOOP
    DBMS_OUTPUT.PUT_LINE(rec.EMP_ID || ' ' || rec.EMP_NAME || ' ' || rec.EXPERIENCE || ' ' || rec.BANK_ID || ' ' || rec.POSITION);
  END LOOP;
END;
```

```
Procedure created.
```

SQL Worksheet

```sql
execute Employee_Bank(3,'BB_Manager');
```

```
Statement processed.
4 Faisal volunteering as EMT 3 BB_Manager
```

SQL Worksheet

```sql
CREATE OR REPLACE PROCEDURE UpdatebloodQuantity (
P_Unit_id IN NUMBER,
P_bloodType IN VARCHAR2,
P_QuantityDelta IN NUMBER
)
AS
BEGIN
UPDATE BLOOD_UNIT
SET quantity = TO_CHAR(TO_NUMBER(quantity) + P_QuantityDelta)
WHERE Unit_id = p_Unit_id AND bloodType = p_bloodType;
COMMIT;
END;
/
```

```
Procedure created.
```

Before Update:

SQL Worksheet

```sql
select * from BLOOD_UNIT;
```

UNIT_ID	BLOODTYPE	QUANTITY	EXPIRATIONDATE	DONOR_ID	BANK_ID	PATIENT_ID
1	A+	3	21-NOV-23	1	1	1
3	AB+	5	22-NOV-23	3	3	3
4	AB+	6	22-NOV-23	4	4	4
5	AB+	10	22-NOV-23	5	5	5
2	B+	3	22-NOV-23	2	2	2

Download CSV

5 rows selected.

SQL Worksheet

Clear Find Actions Save Run

```sql
EXECUTE UpdateBloodQuantity(4,'AB+',2);
```

Statement processed.

After Update:

SQL Worksheet

Clear Find Actions Save Run

```sql
select*from BLOOD_UNIT;
```

UNIT_ID	BLOODTYPE	QUANTITY	EXPIRATIONDATE	DONOR_ID	BANK_ID	PATIENT_ID
3	AB+	5	22-NOV-23	3	3	3
4	AB+	8	22-NOV-23	4	4	4
1	A+	3	21-NOV-23	1	1	1
5	AB+	10	22-NOV-23	5	5	5
2	B+	3	22-NOV-23	2	2	2

Download CSV

3.4 Screenshots of All Tables After Population And Results

```sql
select DONOR.ID,DONOR.FIRSTNAME,DONOR.AGE,BLOOD_UNIT.UNIT_ID,BLOOD_UNIT.BLOODTYPE
FROM DONOR
INNER JOIN BLOOD_UNIT ON DONOR.ID=BLOOD_UNIT.DONOR_ID
```

ID	FIRSTNAME	AGE	UNIT_ID	BLOODTYPE
1	Ahmad	25	1	A+
3	Dana	22	3	AB+
4	Rayan	40	4	AB+
5	Salma	35	5	AB+
2	Sara	19	2	B+

Download CSV

```sql
SELECT BD.BLOODTYPE, BD.diseaseType
FROM BLOOD_DISEASE BD
WHERE BD.BLOODTYPE = 'AB+'
  AND EXISTS (
    SELECT 1
    FROM PATIENT P
    WHERE BD.BLOODTYPE = BD.BLOODTYPE
);
```

BLOODTYPE	DISEASETYPE
AB+	Malaria

Download CSV

```sql
SELECT PATIENT_ID, FIRSTNAME, LASTNAME, AGE, MEDHISTORY
FROM PATIENT
WHERE AGE IN (
    SELECT AGE
    FROM PATIENT
    WHERE AGE < 30
)
ORDER BY AGE DESC
```

PATIENT_ID	FIRSTNAME	LASTNAME	AGE	MEDHISTORY
2	Sara	Ahmad	27	Patient developed an allergic reaction (hives, itching, and shortness of breath) after receiving a platelet transfusion.
1	Rana	Ali	20	Patient received two units of packed red blood cells (PRBCs) during a surgical procedure due to significant blood loss. No adverse reactions reported.

Download CSV

2 rows selected.

```sql
SELECT gender, COUNT(*) as TotalDonors
FROM Donor
GROUP BY gender
ORDER BY gender, AVG(age) DESC;
```

GENDER	TOTALDONORS
Female	3
Male	2

Download CSV

2 rows selected.

```sql
SELECT ID ,FIRSTNAME,AGE
FROM DONOR WHERE AGE <(select AVG(AGE)FROM DONOR)
```

ID	FIRSTNAME	AGE
1	Ahmad	25
2	Sara	19
3	Dana	22

Download CSV

3 rows selected.

```
SELECT AVG(QUANTITY) As  q FROM BLOOD_UNIT where BLOODTYPE='AB+'
```

Q
7

Download CSV