CSE2004 – Database Management System

Project Report

Project Members:

Anas Ahmad Siddiqui

(20BPS1094)

Project Title:

Blood Bank Management System

For Storing and Managing data

Submitted to:

Dr. Balasundaram Ananthakrishnan

Contents covered in the Report: -

Abstract:

Blood Bank management is the storing and management of all data related to a blood bank. A person sitting on his chair in front of a computer can access all the data related to the blood bank. Unlike traditional data management that is carried out physically with effort of a person to write and read out certain data. The entire project has been developed keeping in view of the distributed client server technology, in mind. The project is to create e-information about the donor and organisation that are related to donating the blood. Through this application details of any person who is interested in donating the blood.

The project has been planned to be having the view of distributed architecture, with centralised storage of the database. The application for the storage of the data has been planned using the constructs of Oracle sql server and all the user interface have been designed using the HTML, CSS and JavaScript. The database connectivity is planned using the 'cx_Oracle' library of flask framework.

The schema has been normalised up to 3NF to eliminate all the anomalies that may arise due to the database transaction.

Keywords:

Blood bank management, Website, Oracle, ER Diagram, Schema, Database, RDBMS

Acknowledgement:

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the course of study.

I express my sincere gratitude to **Dr. Balasundaram Ananthakrishnan** for providing me an opportunity to undergo this project of developing blood bank management system.

I am thankful to for his support, cooperation and motivation provided to me during the project for constant inspiration, presence and blessings.

I also extend my sincere appreciation to who provided his valuable suggestions and precious time in accomplishing my Project report.

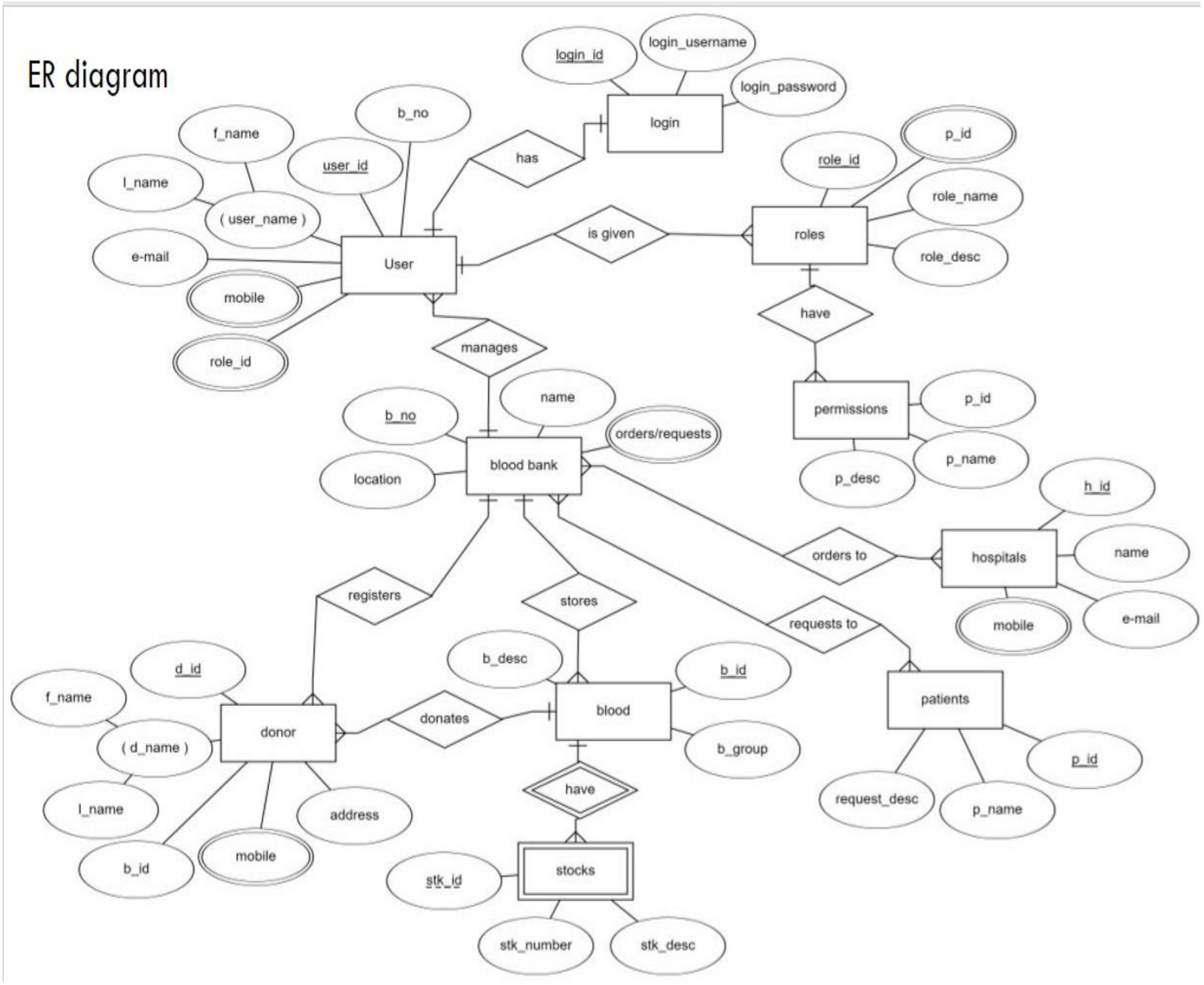
Anas Ahmad Siddiqui
(20BPS1094)

Introduction:

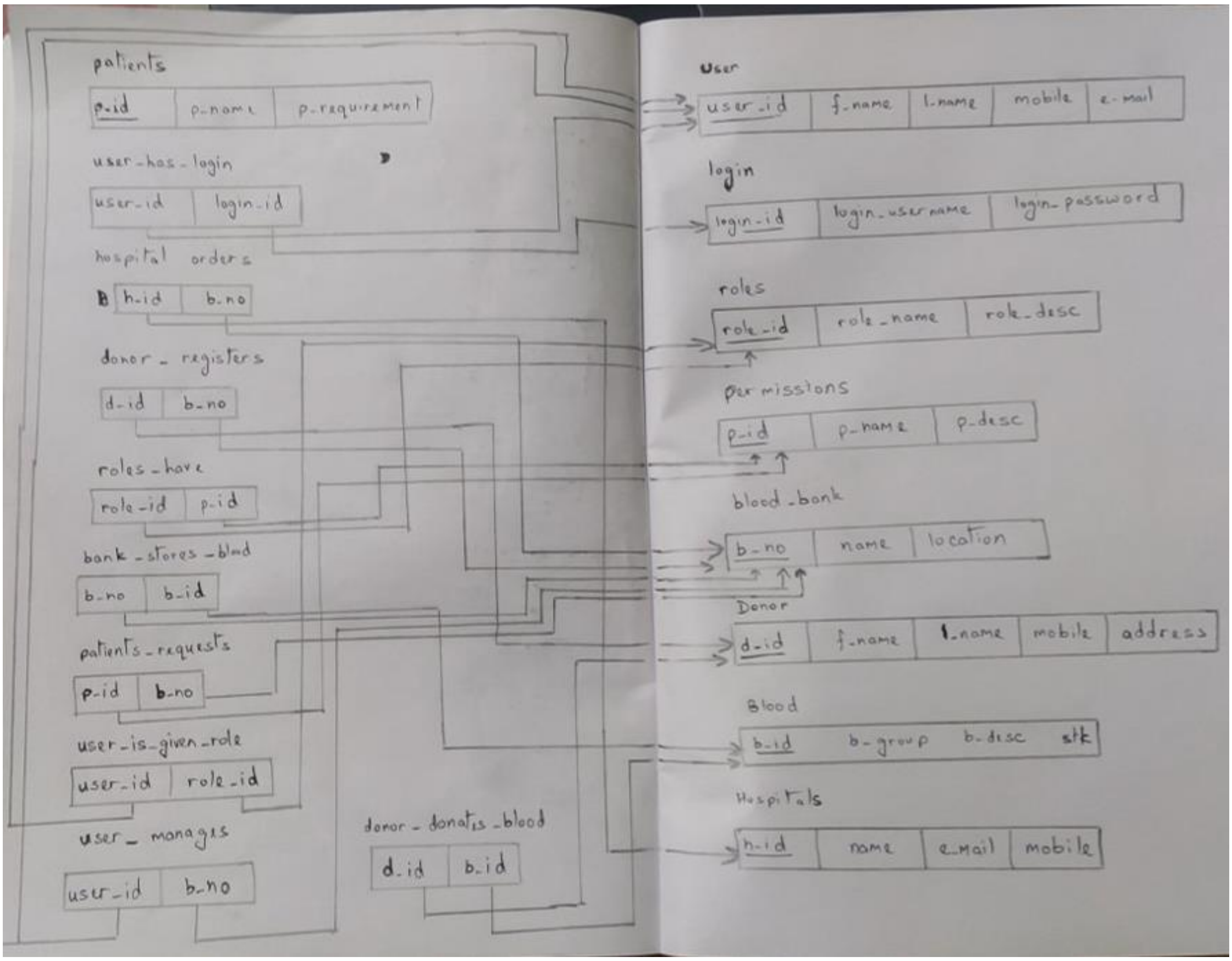
The main purpose to a database is to store information. Have a question about the stocks of blood? Check the database. Want to know about a donor? Check the database. By using a database, an application can ignore the bulk of data and focus more on the data required by the user.

This project is designed to store, process and retrieve information concerned with the administrative and inventory management within a blood bank. It aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. With this system the process of obtaining blood from a blood bank hassle-free and corruption-free and make the system of blood bank management effective.

ER Diagram:


ER diagram
login_id
login_username
login_password
login
has
b_no
f_name
user_id
l_name
(user_name)
e-mail
mobile
role_id
User
is given
roles
role_id
p_id
role_name
role_desc
have
permissions
p_id
p_name
p_desc
manages
name
b_no
orders/requests
location
blood bank
h_id
hospitals
name
mobile
e-mail
orders to
requests to
patients
p_id
p_name
request_desc
registers
stores
d_id
f_name
(d_name)
l_name
donor
address
mobile
b_id
donates
b_desc
blood
b_id
b_group
have
stocks
stk_id
stk_number
stk_desc

ER to Relational Mapping:



Implementation:

The schema was implemented by creating all the tables in the schema in the Oracle database.

The user table stores the personal information of all the users registered to using the website.

The login table stores the login username and password of the users registered.

Roles table stores all the roles and its description that can be granted to an user.

The permissions table describes all the permissions a role can have

The blood_bank table stores the name and locaton regarding a blood bank

The donor table stores information about the donors registered with a blood bank.

The blood table stores information about the stocks regarding a blood group.

The hospitals table stores information about the hospitals which request blood from the blood bank.

The patients table stores information about the patients who requests for blood from the blood bank.

The user_has_login table defines relationship between users table and login table.

The hospitals_orders table defines relationship between hospitals and blood_bank table.

The donor_registers table defines relationship between donor and blood_bank table.

The roles_have table defines relationship between roles and permissions table

The bank_stores_blood defines the relationship between blood_bank and blood table to describe about all the blood stored in the blood bank.

The patients_requests table defines the relationship between patients and blood table depicting the blood grouo that a patient requests for.

The user_is_given_role table defines the relationship between user and roles table depicting the role that a user will have.

The user_manages table defines relationship between user and blood_bank table depicting which blood_bank is managed by which user.

The donor_donates_blood table defines relationship between donor and blood table depicting the blood group of blood donated by a donor.

Experimental Setup:

Hardware Requirements:

- Processor: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
- Installed RAM: 8 GB DDR4
- OS: Windows 10 Home Single Language (19042.906)
- System type: 64-bit operating system, x64-based processor

Software Requirements:

- Backend:
  - Oracle 11g
  - Flask (Python)

- Frontend:
  - HTML
  - CSS
  - JavaScript

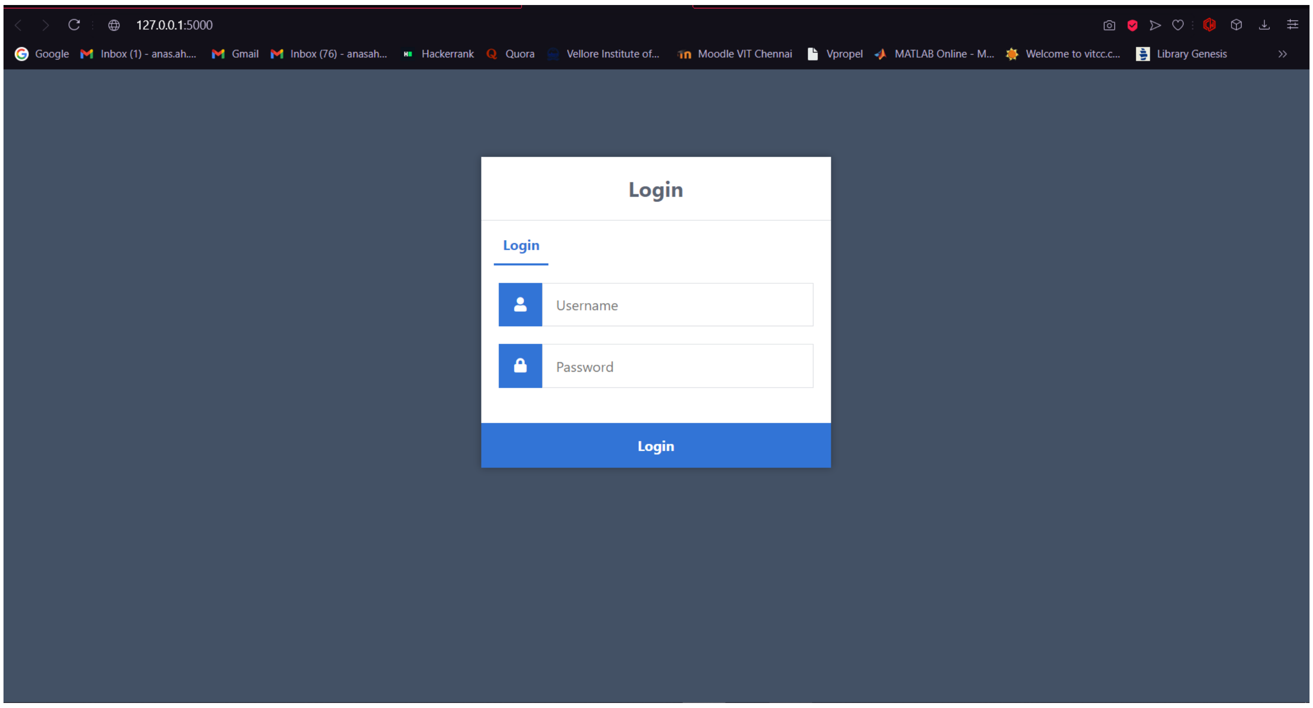
Experimental Result and Discussion:

A website has been developed the can connect to the database and perform certain transactions.

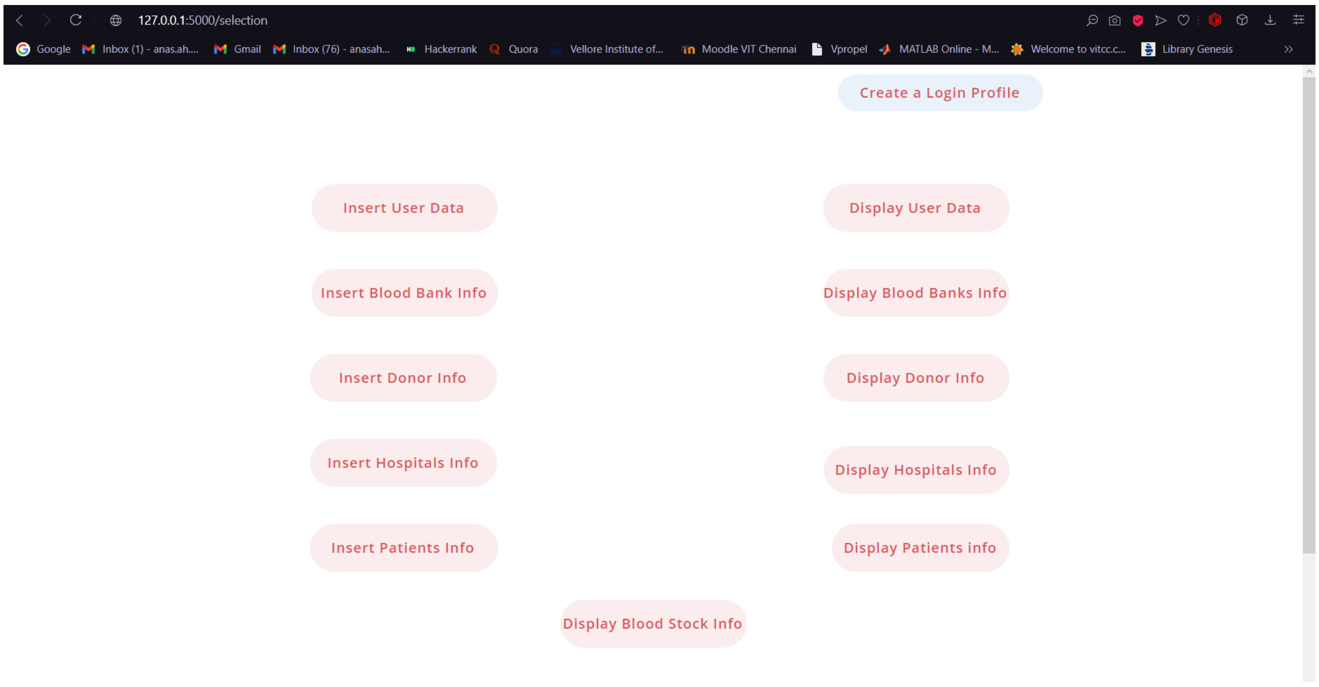
The user first encounters a login page which he/she can use to enter the website. Through the site he can manipulate and display every tablein the database.

The website contains the following pages:

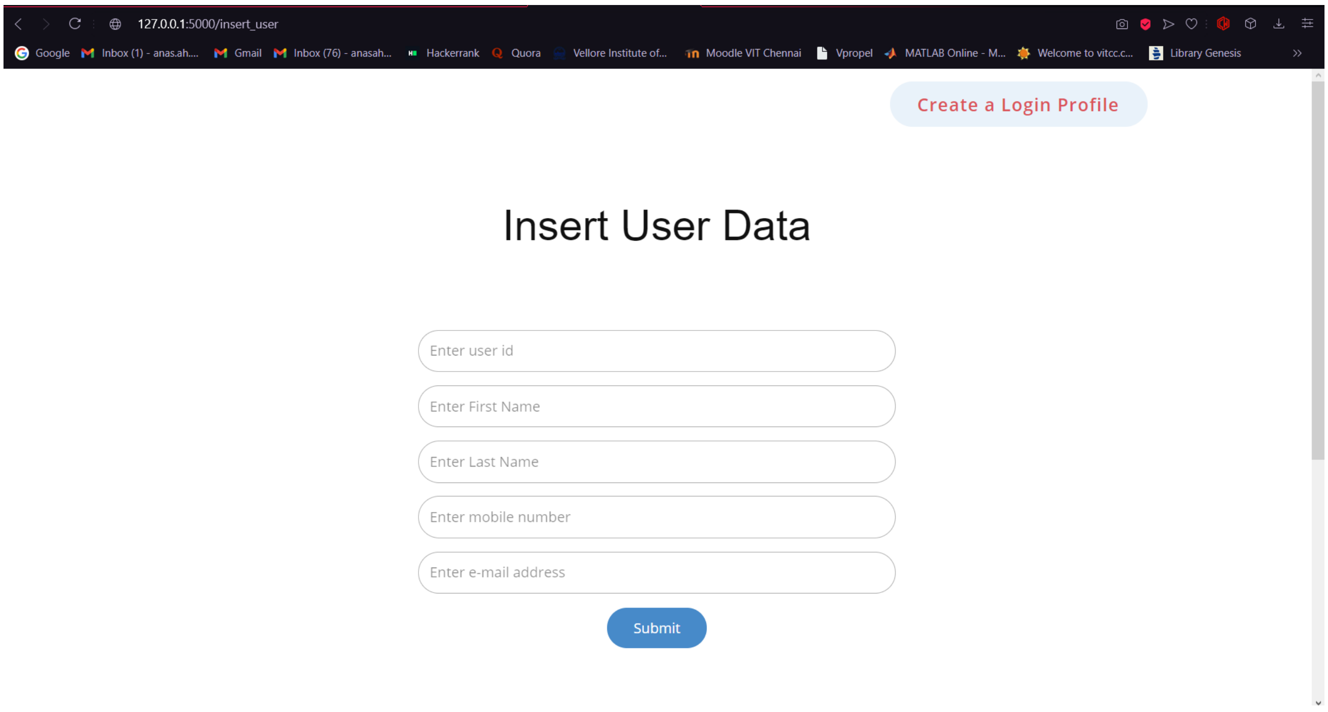
- **<u>Login Page:</u>** User enters the correct username and password to enter the site



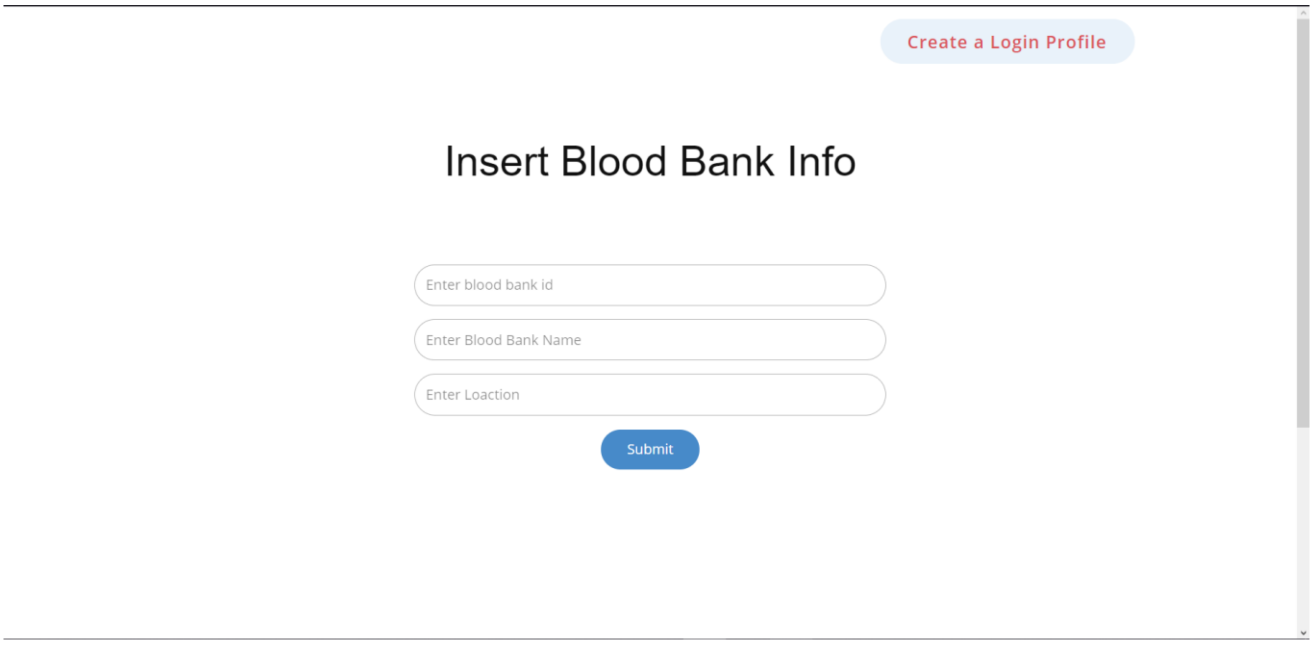
- **Operation Selection Page:** User selects the operation he wants to do.



- **Insert User Data:** Inserts the data into user_ table.

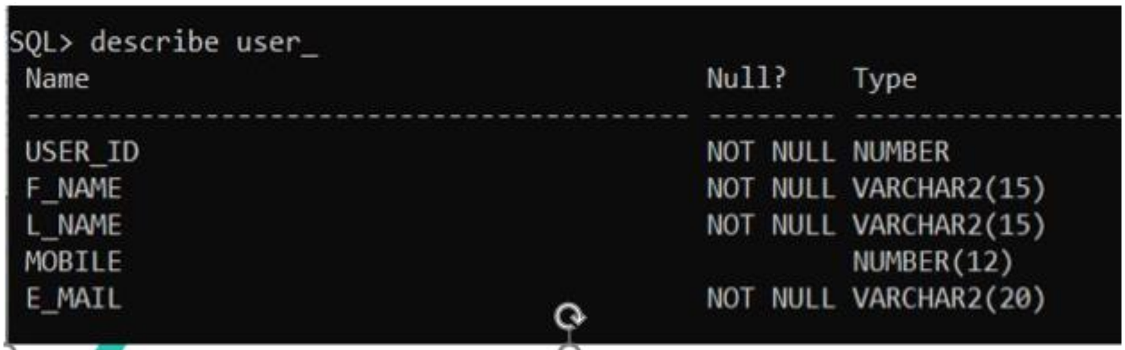


- **<u>Insert Blood Bank Info:</u>** inserts the data into blood_bank table

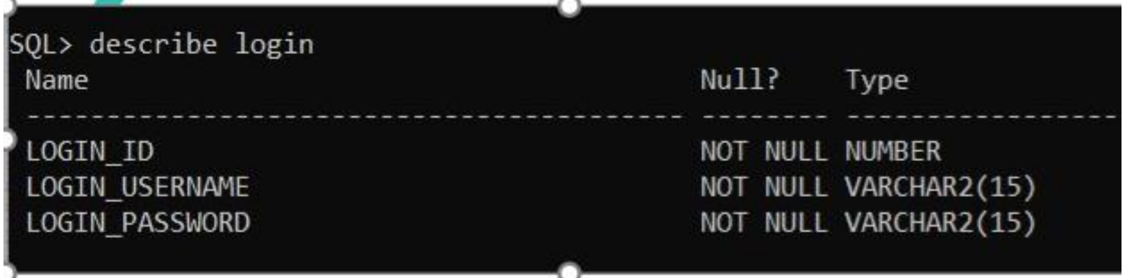


- **<u>Insert Donor Info:</u>** inserts the data into donor table.
- **<u>Insert Hospitals Info:</u>** inserts the data into hospitals table.
- **<u>Insert Patients Info</u>**: inserts data into patient table
- **<u>Display User data</u>**: displays all the columns data of user_ table
- **<u>Display Blood Banks Info:</u>** displays all the columns data of blood_bank table
- **<u>Display Donor Info</u>**: displays all the columns data of donor table
- **<u>Display Hospitals Info:</u>** displays all the columns data from hospitals table
- **<u>Display Patients Info:</u>** displays all the columns data from patients table
- **<u>Display Blood Stock Info</u>**: displays all the columns data from blood table

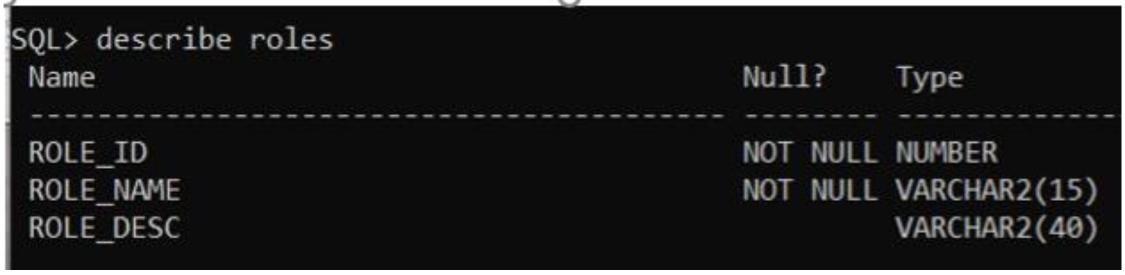
The schema that the database is based upon is described as follows:



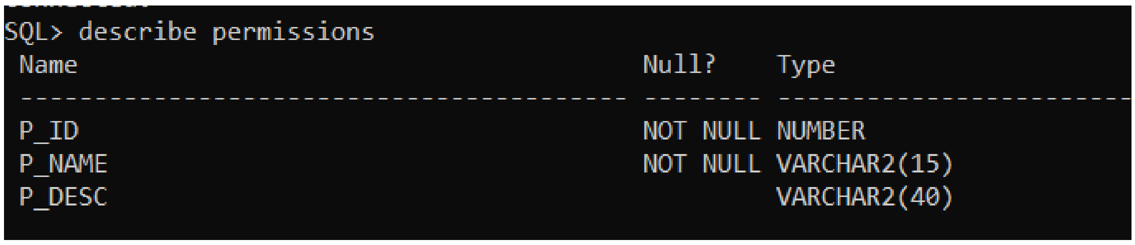
```
SQL> describe user_
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 USER_ID                                   NOT NULL NUMBER
 F_NAME                                    NOT NULL VARCHAR2(15)
 L_NAME                                    NOT NULL VARCHAR2(15)
 MOBILE                                             NUMBER(12)
 E_MAIL                                    NOT NULL VARCHAR2(20)
```



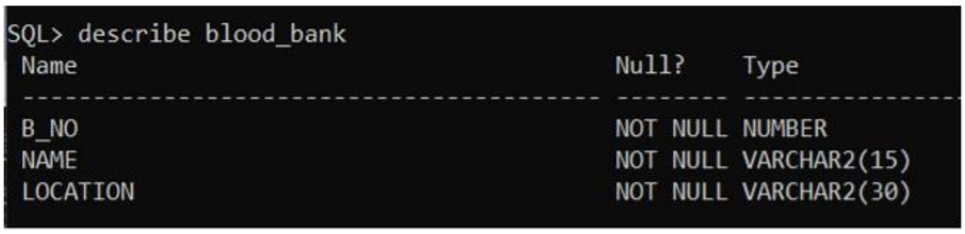
```
SQL> describe login
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 LOGIN_ID                                  NOT NULL NUMBER
 LOGIN_USERNAME                            NOT NULL VARCHAR2(15)
 LOGIN_PASSWORD                            NOT NULL VARCHAR2(15)
```



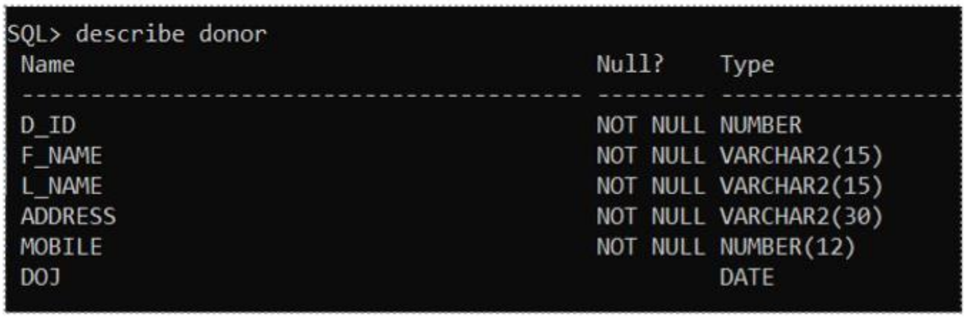
```
SQL> describe roles
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 ROLE_ID                                   NOT NULL NUMBER
 ROLE_NAME                                 NOT NULL VARCHAR2(15)
 ROLE_DESC                                          VARCHAR2(40)
```



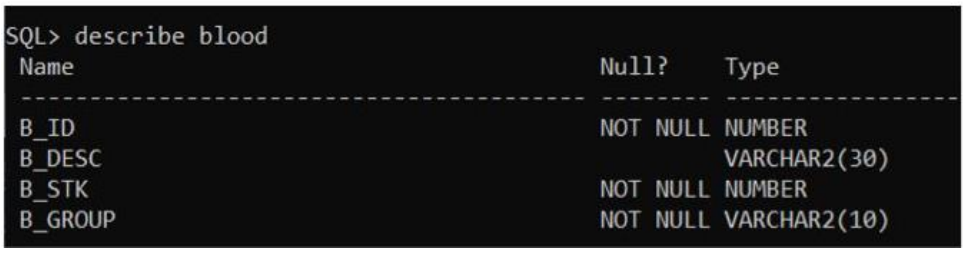
```
SQL> describe permissions
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                      NOT NULL NUMBER
 P_NAME                                    NOT NULL VARCHAR2(15)
 P_DESC                                             VARCHAR2(40)
```



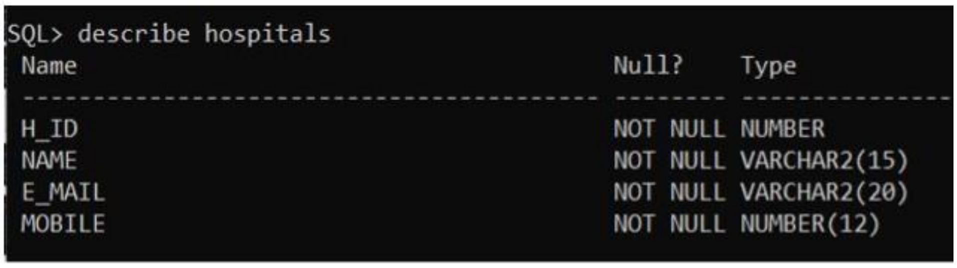
```
SQL> describe blood_bank
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 B_NO                                      NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(15)
 LOCATION                                  NOT NULL VARCHAR2(30)
```



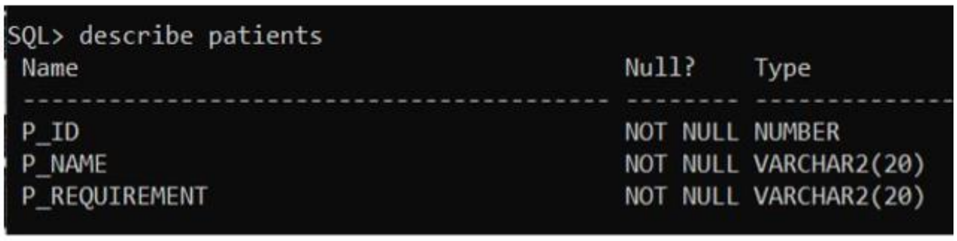
```
SQL> describe donor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 D_ID                                      NOT NULL NUMBER
 F_NAME                                    NOT NULL VARCHAR2(15)
 L_NAME                                    NOT NULL VARCHAR2(15)
 ADDRESS                                   NOT NULL VARCHAR2(30)
 MOBILE                                    NOT NULL NUMBER(12)
 DOJ                                                DATE
```



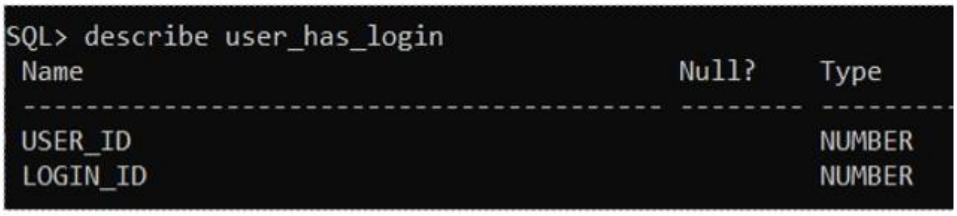
```
SQL> describe blood
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 B_ID                                      NOT NULL NUMBER
 B_DESC                                             VARCHAR2(30)
 B_STK                                     NOT NULL NUMBER
 B_GROUP                                   NOT NULL VARCHAR2(10)
```



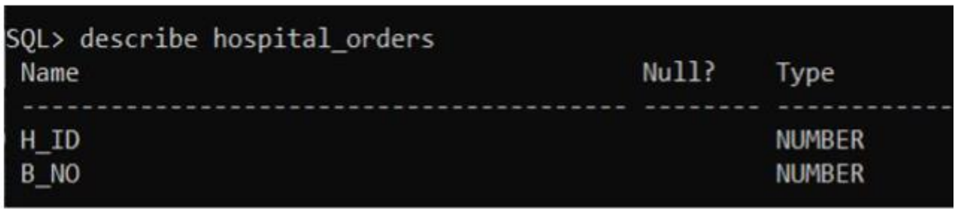
```
SQL> describe hospitals
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 H_ID                                      NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(15)
 E_MAIL                                    NOT NULL VARCHAR2(20)
 MOBILE                                    NOT NULL NUMBER(12)
```



```
SQL> describe patients
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                      NOT NULL NUMBER
 P_NAME                                    NOT NULL VARCHAR2(20)
 P_REQUIREMENT                             NOT NULL VARCHAR2(20)
```



```
SQL> describe user_has_login
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USER_ID                                            NUMBER
 LOGIN_ID                                           NUMBER
```



```
SQL> describe hospital_orders
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 H_ID                                               NUMBER
 B_NO                                               NUMBER
```

Conclusion and Future Work:

I was able to develop a database system the can store and manage all the data required by a blood bank management system.

I faced a few difficulties in finding the actual working of a blood bank and how & what data is actually stored and required. This information helped me in developing a schema best suited for the project.

A minimal front-end is used which can be improved and
the connection to the database can be implemented for the
whole website.

The login page only lets the user already registered to enter the site. As of now, the insertion task can be completely taken care from the website. Sign up page can be used to register new user. Though the display of data from the tables to website requires some more work and debugging which can be considered as future work for the project.

References:

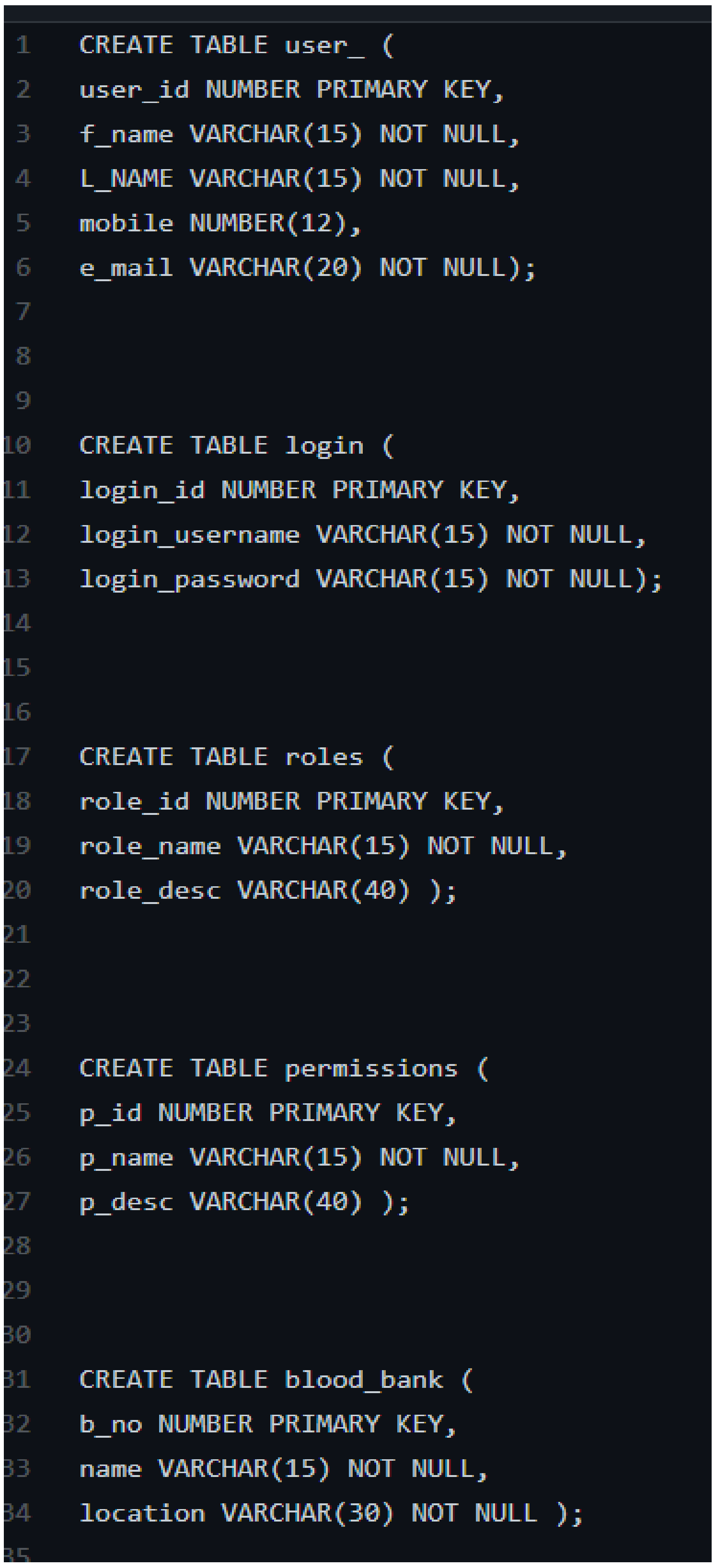
https://en.wikipedia.org/wiki/Blood_bank#:~:text=Most%20blood%20for%20transfusion%20is,the%20most%20commonly%20used%20product.

https://flask.palletsprojects.com/en/1.1.x/quickstart/

https://docs.oracle.com/cd/E11882_01/index.htm

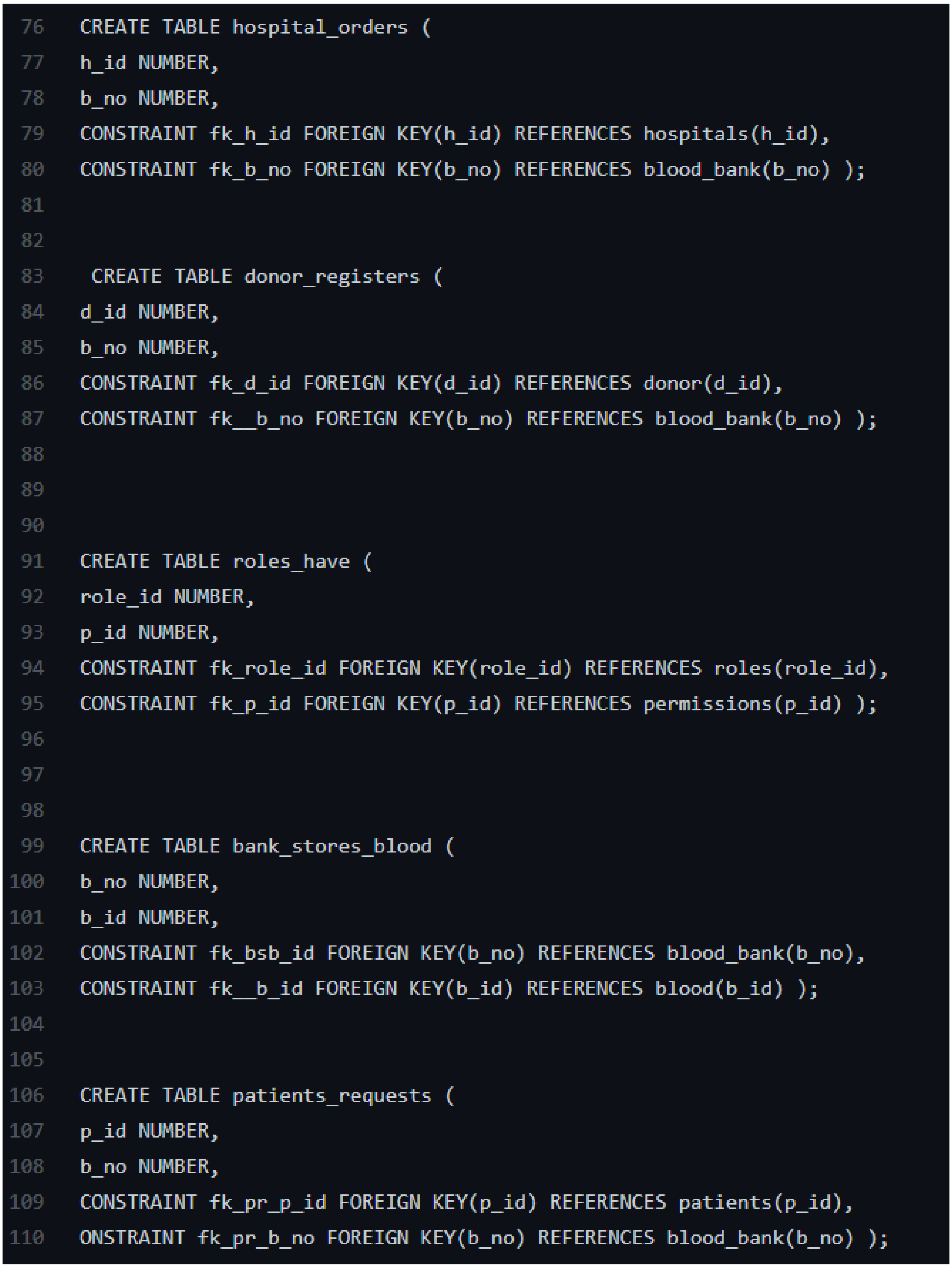
APPENDIX 1:

Create Table Queries:

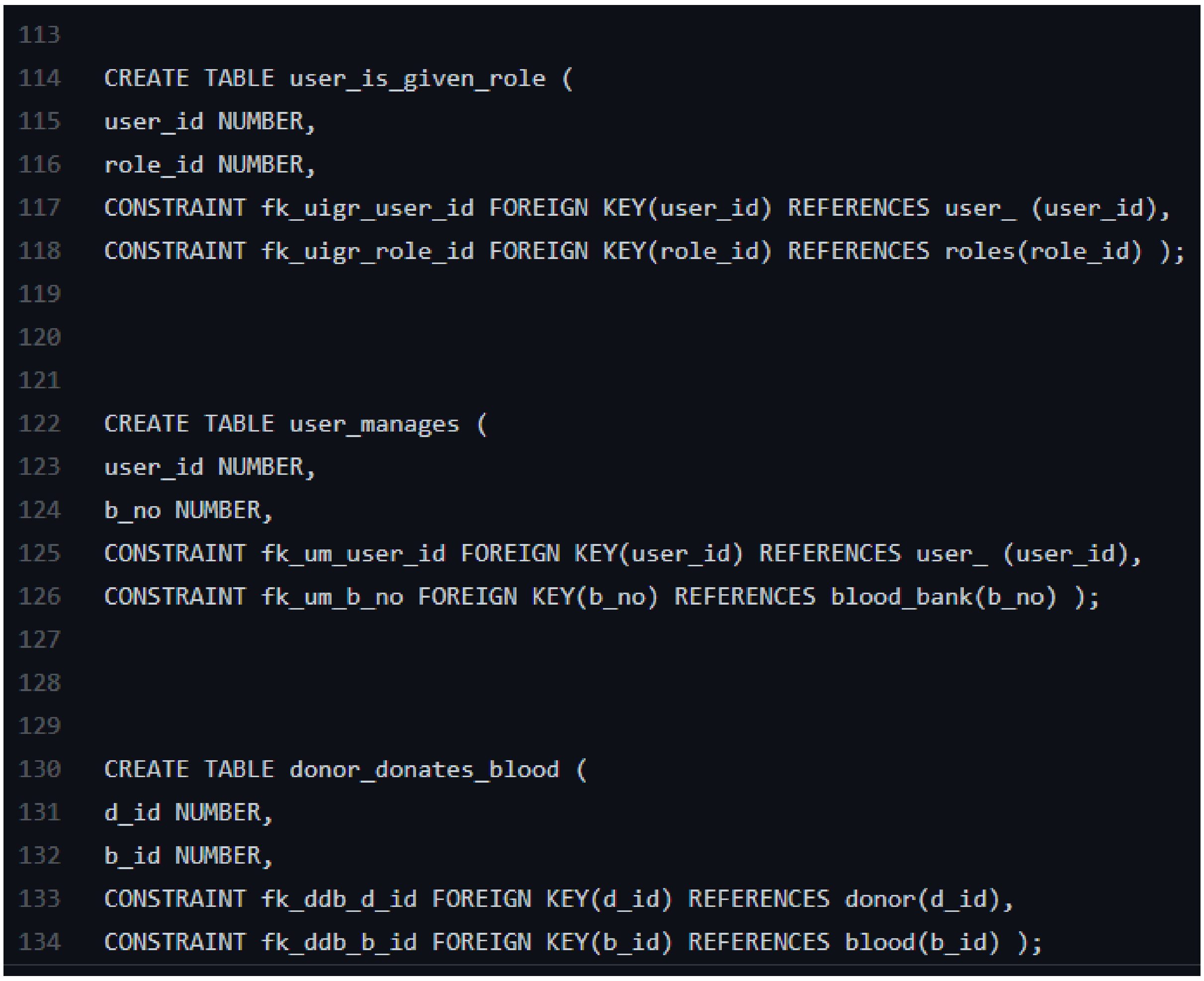


```
CREATE TABLE user_ (
user_id NUMBER PRIMARY KEY,
f_name VARCHAR(15) NOT NULL,
L_NAME VARCHAR(15) NOT NULL,
mobile NUMBER(12),
e_mail VARCHAR(20) NOT NULL);



CREATE TABLE login (
login_id NUMBER PRIMARY KEY,
login_username VARCHAR(15) NOT NULL,
login_password VARCHAR(15) NOT NULL);



CREATE TABLE roles (
role_id NUMBER PRIMARY KEY,
role_name VARCHAR(15) NOT NULL,
role_desc VARCHAR(40) );



CREATE TABLE permissions (
p_id NUMBER PRIMARY KEY,
p_name VARCHAR(15) NOT NULL,
p_desc VARCHAR(40) );



CREATE TABLE blood_bank (
b_no NUMBER PRIMARY KEY,
name VARCHAR(15) NOT NULL,
location VARCHAR(30) NOT NULL );
```

```
CREATE TABLE donor (
d_id NUMBER PRIMARY KEY,
f_name VARCHAR(15) NOT NULL,
l_name VARCHAR(15) NOT NULL,
address VARCHAR(30) NOT NULL,
mobile NUMBER(12) not NULL,
doj DATE );



CREATE TABLE blood (
b_id NUMBER PRIMARY KEY,
b_group VARCHAR(10) NOT NULL,
b_desc VARCHAR(30),
b_stk NUMBER NOT NULL );


CREATE TABLE hospitals (
h_id NUMBER PRIMARY KEY,
name VARCHAR(15) NOT NULL,
e_mail VARCHAR(20) NOT NULL,
mobile NUMBER(12) NOT NULL );


CREATE TABLE patients (
p_id NUMBER PRIMARY KEY,
p_name VARCHAR(20) NOT NULL,
p_requirement VARCHAR(20) NOT NULL );



CREATE TABLE user_has_login (
user_id NUMBER,
login_id NUMBER,
CONSTRAINT fk_user_id FOREIGN KEY(user_id) REFERENCES user_ (user_id),
CONSTRAINT fk_login_id FOREIGN KEY(login_id) REFERENCES login(login_id));
```

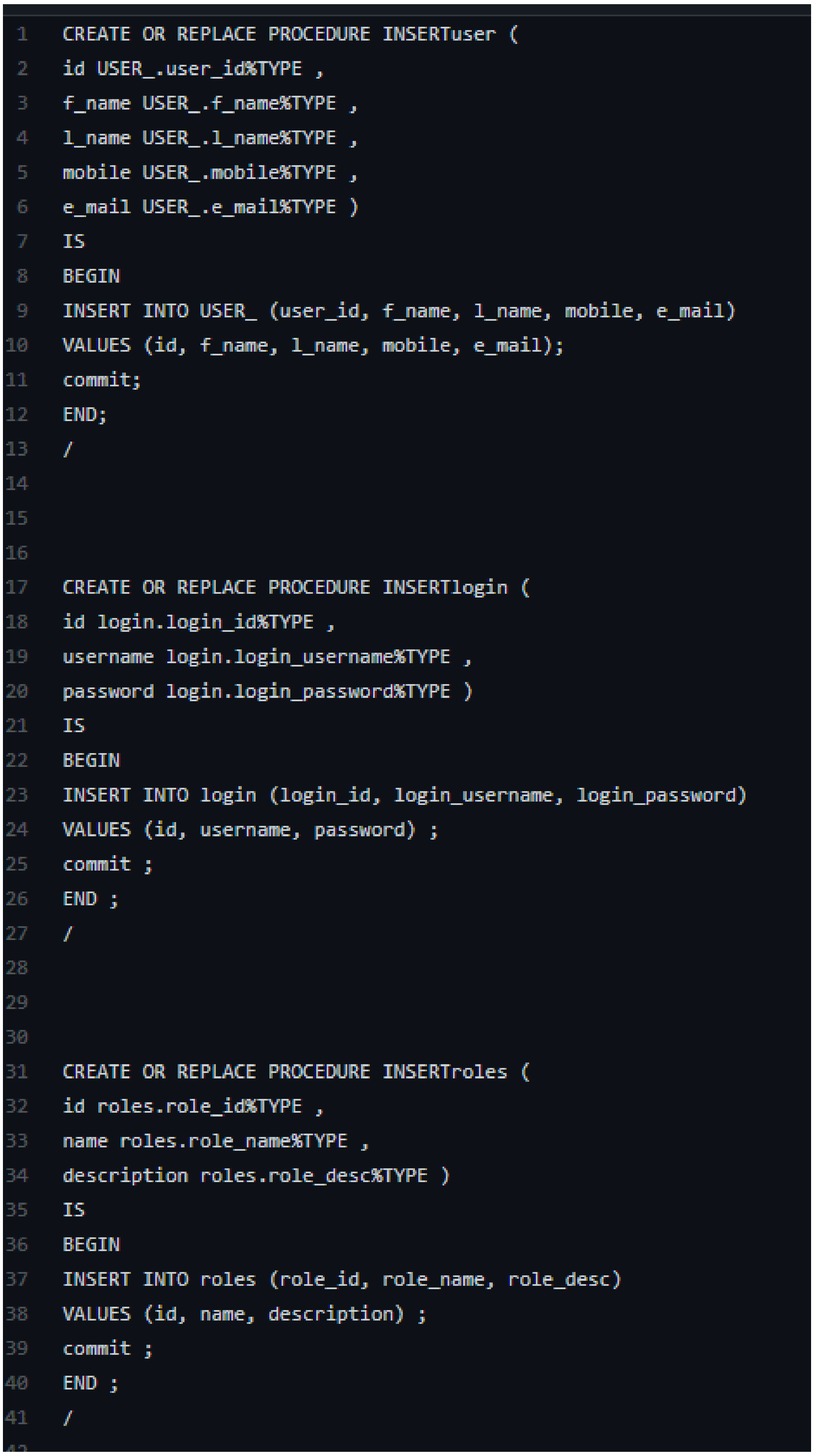


```
CREATE TABLE hospital_orders (
h_id NUMBER,
b_no NUMBER,
CONSTRAINT fk_h_id FOREIGN KEY(h_id) REFERENCES hospitals(h_id),
CONSTRAINT fk_b_no FOREIGN KEY(b_no) REFERENCES blood_bank(b_no) );


 CREATE TABLE donor_registers (
d_id NUMBER,
b_no NUMBER,
CONSTRAINT fk_d_id FOREIGN KEY(d_id) REFERENCES donor(d_id),
CONSTRAINT fk__b_no FOREIGN KEY(b_no) REFERENCES blood_bank(b_no) );



CREATE TABLE roles_have (
role_id NUMBER,
p_id NUMBER,
CONSTRAINT fk_role_id FOREIGN KEY(role_id) REFERENCES roles(role_id),
CONSTRAINT fk_p_id FOREIGN KEY(p_id) REFERENCES permissions(p_id) );



CREATE TABLE bank_stores_blood (
b_no NUMBER,
b_id NUMBER,
CONSTRAINT fk_bsb_id FOREIGN KEY(b_no) REFERENCES blood_bank(b_no),
CONSTRAINT fk__b_id FOREIGN KEY(b_id) REFERENCES blood(b_id) );


CREATE TABLE patients_requests (
p_id NUMBER,
b_no NUMBER,
CONSTRAINT fk_pr_p_id FOREIGN KEY(p_id) REFERENCES patients(p_id),
ONSTRAINT fk_pr_b_no FOREIGN KEY(b_no) REFERENCES blood_bank(b_no) );
```

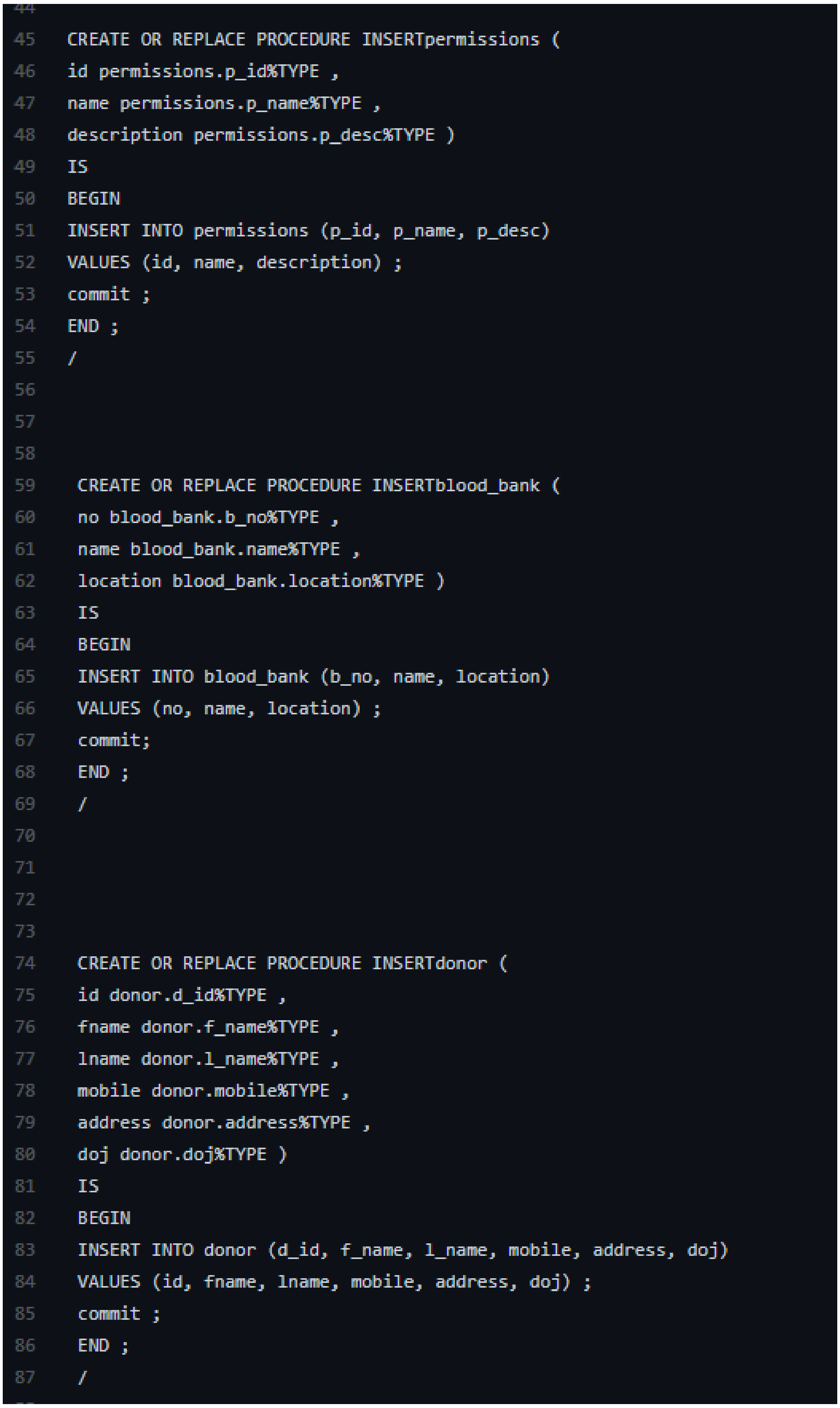


```
CREATE TABLE user_is_given_role (
user_id NUMBER,
role_id NUMBER,
CONSTRAINT fk_uigr_user_id FOREIGN KEY(user_id) REFERENCES user_ (user_id),
CONSTRAINT fk_uigr_role_id FOREIGN KEY(role_id) REFERENCES roles(role_id) );



CREATE TABLE user_manages (
user_id NUMBER,
b_no NUMBER,
CONSTRAINT fk_um_user_id FOREIGN KEY(user_id) REFERENCES user_ (user_id),
CONSTRAINT fk_um_b_no FOREIGN KEY(b_no) REFERENCES blood_bank(b_no) );



CREATE TABLE donor_donates_blood (
d_id NUMBER,
b_id NUMBER,
CONSTRAINT fk_ddb_d_id FOREIGN KEY(d_id) REFERENCES donor(d_id),
CONSTRAINT fk_ddb_b_id FOREIGN KEY(b_id) REFERENCES blood(b_id) );
```

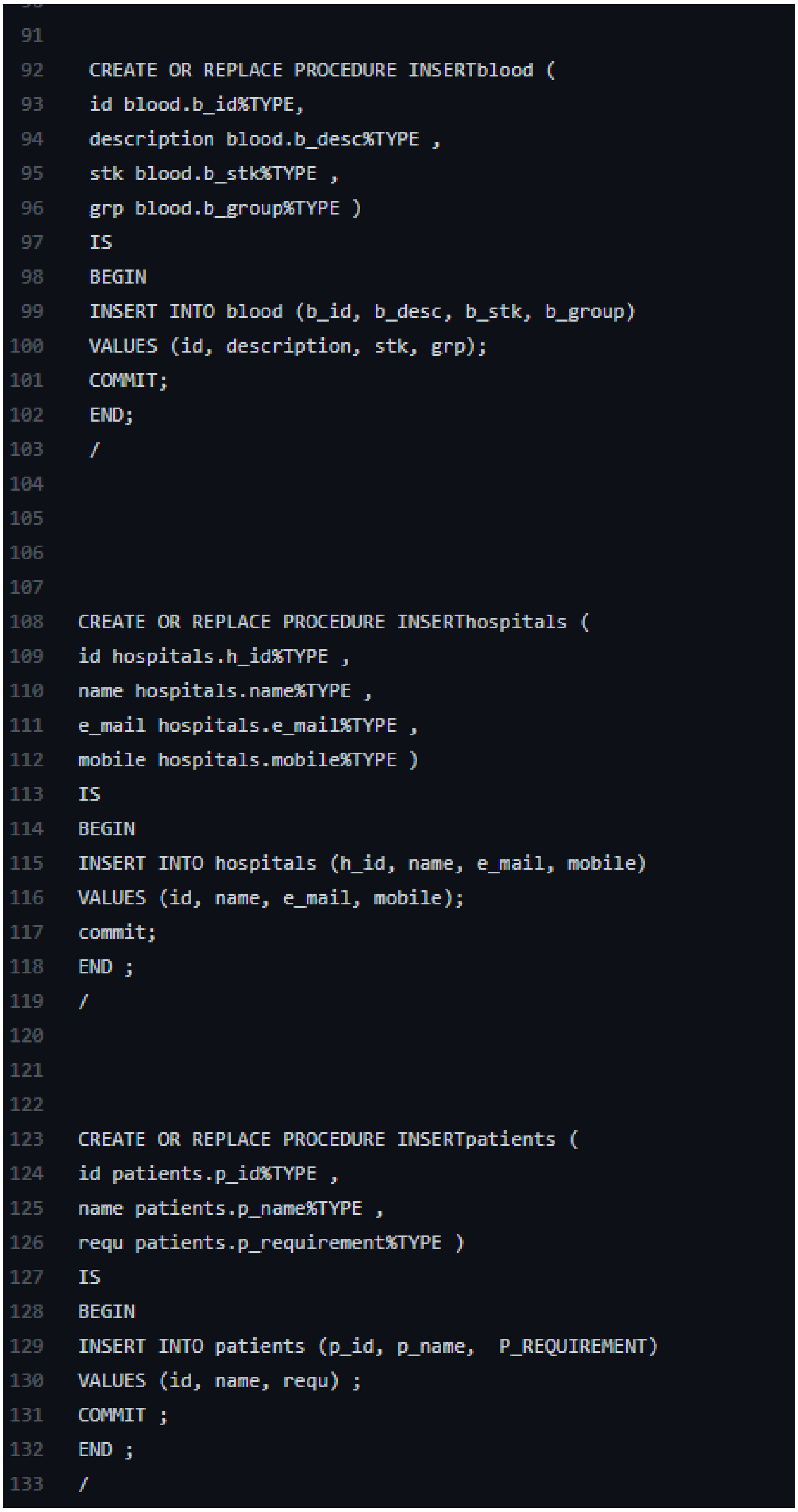
Sample Insert Table Queries:



```
CREATE OR REPLACE PROCEDURE INSERTuser (
id USER_.user_id%TYPE ,
f_name USER_.f_name%TYPE ,
l_name USER_.l_name%TYPE ,
mobile USER_.mobile%TYPE ,
e_mail USER_.e_mail%TYPE )
IS
BEGIN
INSERT INTO USER_ (user_id, f_name, l_name, mobile, e_mail)
VALUES (id, f_name, l_name, mobile, e_mail);
commit;
END;
/



CREATE OR REPLACE PROCEDURE INSERTlogin (
id login.login_id%TYPE ,
username login.login_username%TYPE ,
password login.login_password%TYPE )
IS
BEGIN
INSERT INTO login (login_id, login_username, login_password)
VALUES (id, username, password) ;
commit ;
END ;
/



CREATE OR REPLACE PROCEDURE INSERTroles (
id roles.role_id%TYPE ,
name roles.role_name%TYPE ,
description roles.role_desc%TYPE )
IS
BEGIN
INSERT INTO roles (role_id, role_name, role_desc)
VALUES (id, name, description) ;
commit ;
END ;
/
```



```
CREATE OR REPLACE PROCEDURE INSERTpermissions (
id permissions.p_id%TYPE ,
name permissions.p_name%TYPE ,
description permissions.p_desc%TYPE )
IS
BEGIN
INSERT INTO permissions (p_id, p_name, p_desc)
VALUES (id, name, description) ;
commit ;
END ;
/



 CREATE OR REPLACE PROCEDURE INSERTblood_bank (
 no blood_bank.b_no%TYPE ,
 name blood_bank.name%TYPE ,
 location blood_bank.location%TYPE )
 IS
 BEGIN
 INSERT INTO blood_bank (b_no, name, location)
 VALUES (no, name, location) ;
 commit;
 END ;
 /




 CREATE OR REPLACE PROCEDURE INSERTdonor (
 id donor.d_id%TYPE ,
 fname donor.f_name%TYPE ,
 lname donor.l_name%TYPE ,
 mobile donor.mobile%TYPE ,
 address donor.address%TYPE ,
 doj donor.doj%TYPE )
 IS
 BEGIN
 INSERT INTO donor (d_id, f_name, l_name, mobile, address, doj)
 VALUES (id, fname, lname, mobile, address, doj) ;
 commit ;
 END ;
 /
```



```
 CREATE OR REPLACE PROCEDURE INSERTblood (
 id blood.b_id%TYPE,
 description blood.b_desc%TYPE ,
 stk blood.b_stk%TYPE ,
 grp blood.b_group%TYPE )
 IS
 BEGIN
 INSERT INTO blood (b_id, b_desc, b_stk, b_group)
 VALUES (id, description, stk, grp);
 COMMIT;
 END;
 /




CREATE OR REPLACE PROCEDURE INSERThospitals (
id hospitals.h_id%TYPE ,
name hospitals.name%TYPE ,
e_mail hospitals.e_mail%TYPE ,
mobile hospitals.mobile%TYPE )
IS
BEGIN
INSERT INTO hospitals (h_id, name, e_mail, mobile)
VALUES (id, name, e_mail, mobile);
commit;
END ;
/



CREATE OR REPLACE PROCEDURE INSERTpatients (
id patients.p_id%TYPE ,
name patients.p_name%TYPE ,
requ patients.p_requirement%TYPE )
IS
BEGIN
INSERT INTO patients (p_id, p_name,  P_REQUIREMENT)
VALUES (id, name, requ) ;
COMMIT ;
END ;
/
```

APPENDIX 2:

Table Creation Files Download:

https://drive.google.com/file/d/1h_drVHE00-HaGr-4Inj5pURGuT-gUCa6/view?usp=sharing

Data Insertion Files Download:

https://drive.google.com/file/d/1TRHjw5j9b3PWByxvoQawYjB-QK_iW99e/view?usp=sharing

Website Source Code Files Download:

https://github.com/Anas-Ahmad-Siddiqui/blood_bank_management_sys.git