

INTERNET PROGRAMMING AND MOBILE PROGRAMMING CEF 440
TASK 6

DATABASE IMPLEMENTATION

Submitted By:

NAMES	MATRICULE
NDIKINTUM CARL NFON	FE20A073
NYENTY EYONG ARREHQUETTE	FE20A094
OBASIARREY M'ONEKE MARY ARREY-NJOK	FE20A095
OROCKTAKANG AGBORBEJA NTANGTANG	FE20A097
SALLE-NJUME MERYL EPOTE	FE20A102

Course Instructor
Dr Nkemeni Valery

TABLE OF CONTENT

DATABASE IMPLEMENTATION.....	1
Introduction.....	3
Entity-Relationship Diagram (ERD).....	3
Physical Database Schema.....	4
Database Relationships.....	5
Queries.....	6
Query to create User Table.....	6
Query to create Donations Table.....	6
Query to create Sold table.....	6
IMPORTATION OF DATA.....	7
TABLES.....	9
Conclusion.....	10

Introduction

Purpose:

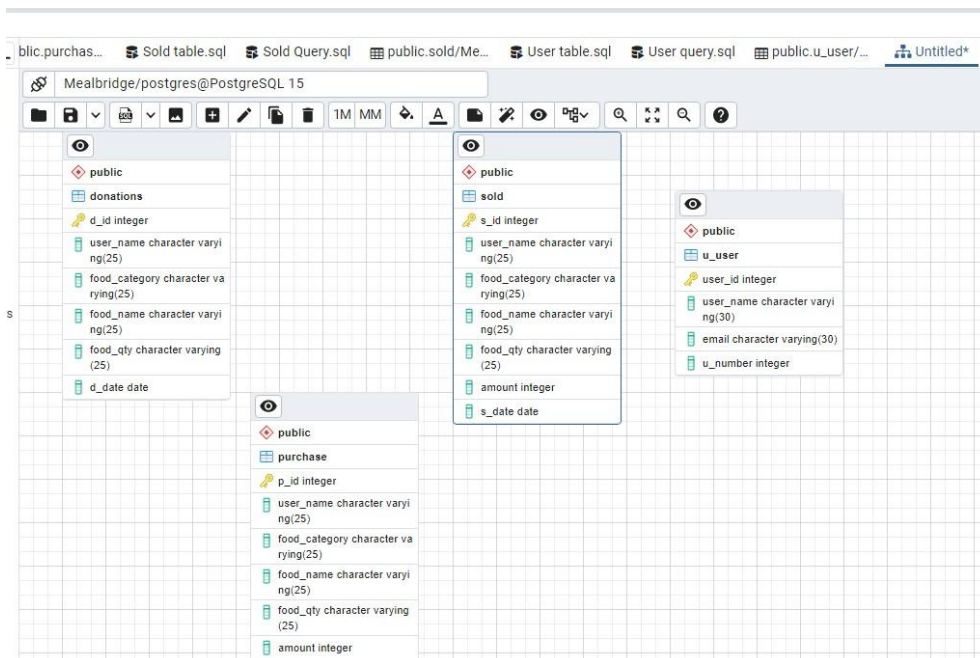
The purpose of this database is to support a food donation platform that allows users to donate food, sell food at discount prices, availability of food and request food.

Scope:

The database will store information related to users, food items and transactions.

Entity-Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) is a visual representation of the entities (objects), relationships between entities, and attributes (properties) of entities in the food donation platform database.



Physical Database Schema

a. User Table

```
CREATE TABLE U_user (  
  User_id INT NOT NULL PRIMARY KEY,  
  User_name VARCHAR(30) NOT NULL,  
  Email VARCHAR NOT NULL,  
  U_number int NOT NULL  
);
```

b. Donations Table

```
CREATE TABLE Donations(  
  D_id INT NOT NULL, PRIMARY KEY  
  User_name VARCHAR(25),  
  food_category VARCHAR(25),  
  food_name VARCHAR(25) NOT NULL,  
  food_qty VARCHAR(25),  
  D_date date  
);
```

c. Sold table

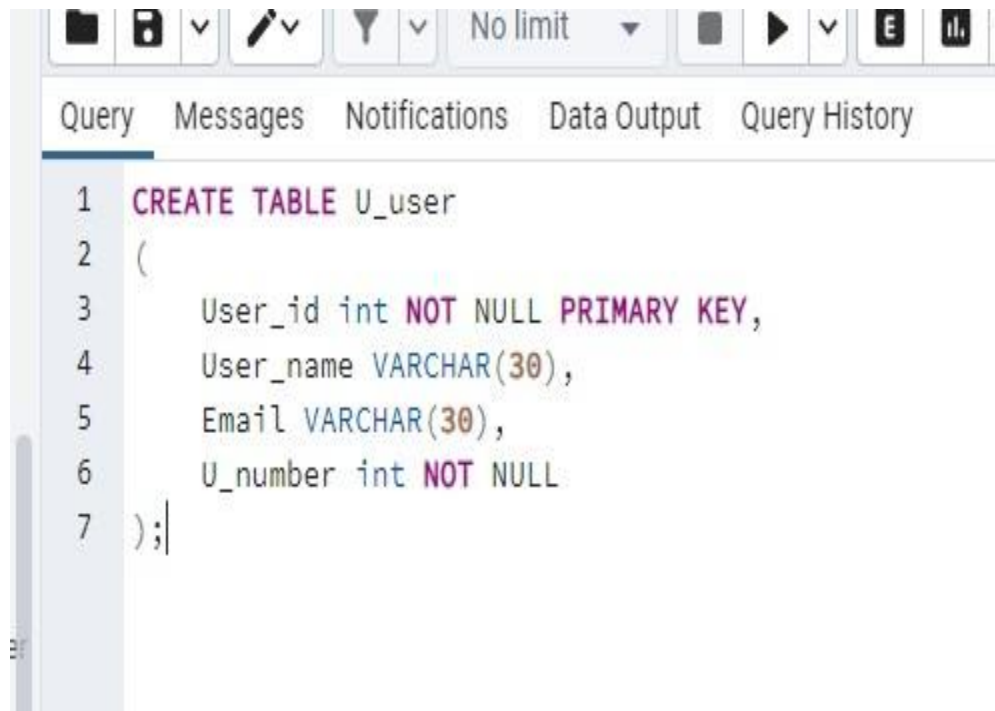
```
CREATE TABLE Sold(  
  s_id INT NOT NULL PRIMARY KEY,  
  user_name VARCHAR(25),  
  food_category VARCHAR(25),  
  food_name VARCHAR(25),  
  food_qty VARCHAR(25),  
  Amount INT NOT NULL,  
  s_date DATE  
);
```

Database Relationships

- One-to-Many:
 - Users to Food Items: One user can have multiple food items for donation or sale.
 - Users to Transactions: One user can have multiple transactions.

Queries

Query to create User Table



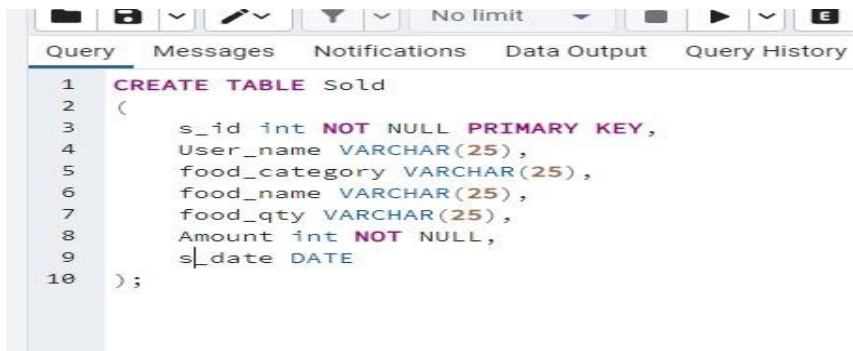
```
1 CREATE TABLE U_user
2 (
3     User_id int NOT NULL PRIMARY KEY,
4     User_name VARCHAR(30),
5     Email VARCHAR(30),
6     U_number int NOT NULL
7 );
```

The screenshot shows a database query editor interface. At the top, there is a toolbar with icons for file operations, editing, and execution. Below the toolbar is a tabbed interface with tabs for 'Query', 'Messages', 'Notifications', 'Data Output', and 'Query History'. The 'Query' tab is active, displaying a SQL query to create a table named 'U_user'. The query is written in a syntax-highlighted format, with keywords like 'CREATE TABLE', 'PRIMARY KEY', and 'NOT NULL' in purple, and data types like 'int' and 'VARCHAR' in blue. The table has four columns: 'User_id' (int, NOT NULL, PRIMARY KEY), 'User_name' (VARCHAR(30)), 'Email' (VARCHAR(30)), and 'U_number' (int, NOT NULL). The query is numbered 1 through 7 on the left margin.

Query to create Donations Table

```
CREATE TABLE IF NOT EXISTS public.donations
(
    d_id integer NOT NULL,
    user_name character varying(25) COLLATE pg_catalog."default",
    food_category character varying(25) COLLATE pg_catalog."default",
    food_name character varying(25) COLLATE pg_catalog."default",
    food_qty character varying(25) COLLATE pg_catalog."default",
    d_date date,
    CONSTRAINT donations_pkey PRIMARY KEY (d_id)
)
```

Query to create Sold table

A screenshot of a database query editor interface. The interface has a toolbar at the top with icons for file operations, a search icon, a filter icon, a dropdown menu showing 'No limit', and buttons for running the query and saving. Below the toolbar are tabs for 'Query', 'Messages', 'Notifications', 'Data Output', and 'Query History'. The 'Query' tab is active, showing a SQL script for creating a table named 'Sold'. The script is as follows:

```
1 CREATE TABLE Sold
2 (
3     s_id int NOT NULL PRIMARY KEY,
4     User_name VARCHAR(25),
5     food_category VARCHAR(25),
6     food_name VARCHAR(25),
7     food_qty VARCHAR(25),
8     Amount int NOT NULL,
9     s_date DATE
10 );
```

IMPORTATION OF DATA

We used 'MOCKAROO' to import and generate data into the different tables

User table

```
INSERT INTO
U_user (User_id, User_name, Email, U_number)
VALUES
('001', 'Samantha', 'saman@gmail.com', '676876543'),
('002', 'Sam', 'sam@gmail.com', '676876543'),
('003', 'John', 'john@gmail.com', '676876543'),
('004', 'Ruth', 'ruth@gmail.com', '676876543'),
('005', 'James', 'james@gmail.com', '676876543'),
('006', 'Casandra', 'casy@gmail.com', '676876543'),
('007', 'Cedrick', 'ced@gmail.com', '676876543'),
('008', 'Micheal', 'mic@gmail.com', '676876543'),
('009', 'Joshua', 'jos@gmail.com', '676876543');
```

Donations Table

Query Query History



```
1 INSERT INTO
2 donations(d_id, user_name, food_category, food_name, food_qty, d_date)
3 VALUES
4 ('001', 'Samantha', 'vegetarian', 'Bitter leaf', '2 bundles', '2022-02-09')
5 ('002', 'Jules', 'carnavarian', 'Sharwama', '2', '2022-02-09'),
6 ('003', 'Peace', 'vegetarian', 'Bitter leaf', '2 bundles', '2022-02-09'),
7 ('004', 'Jules', 'carnavarian', 'Sharwama', '2', '2022-02-09'),
8 ('005', 'Peace', 'vegetarian', 'Bitter leaf', '2 bundles', '2022-02-09'),
9 ('006', 'Jules', 'carnavarian', 'Sharwama', '2', '2022-02-09'),
10 ('007', 'Charity', 'canivarian', 'Sharwama', '2', '2022-02-09');
11
```

Sold Table

```
Query Messages Notifications Data Output Query History
1 INSERT INTO
2 sold(s_id, user_name, food_category, food_name, food_qty, amount, s_date)
3 VALUES
4 ('001', 'Samantha', 'vegetarian', 'Bitter leaf', '2 bundles', '400', '2022-02-09'),
5 ('002', 'Jules', 'carnavarian', 'Sharwama', '2', '600', '2022-02-09'),
6 ('003', 'Peace', 'vegetarian', 'Bitter leaf', '2 bundles', '900', '2022-02-09'),
7 ('004', 'Jules', 'carnavarian', 'Sharwama', '2', '500', '2022-02-09'),
8 ('005', 'Peace', 'vegetarian', 'Bitter leaf', '2 bundles', '650', '2022-02-09'),
9 ('006', 'Jules', 'carnavarian', 'Sharwama', '2', '800', '2022-02-09'),
10 ('007', 'Charity', 'canivarian', 'Sharwama', '2', '900', '2022-02-09');
```

TABLES

User Table

	user_id [PK] integer	user_name character varying (30)	email character varying (30)	u_number integer
1	1	Samantha	saman@gmail.com	676876543
2	2	Sam	sam@gmail.com	676876543
3	3	JOhn	john@gmail.com	676876543
4	4	Ruth	ruth@gmail.com	676876543
5	5	James	james@gmail.com	676876543
6	6	Casandra	casy@gmail.com	676876543
7	7	Cedrick	ced@gmail.com	676876543
8	8	Micheal	mic@gmail.com	676876543
9	9	Joshua	jos@gmail.com	676876543

Donations Table

```
1 SELECT * FROM public.donations
2 ORDER BY d_id ASC
```

Output Messages

d_id	user_name	food_category	food_name	food_qty	d_date
1	Samantha	vegetarian	Bitter leaf	2 bundles	2022-02-09
2	Jules	carnavarian	Sharwama	2	2022-02-09
3	Peace	vegetarian	Bitter leaf	2 bundles	2022-02-09
4	Jules	carnavarian	Sharwama	2	2022-02-09
5	Peace	vegetarian	Bitter leaf	2 bundles	2022-02-09
6	Jules	carnavarian	Sharwama	2	2022-02-09
7	Charity	canivarian	Sharwama	2	2022-02-09

Sold Table

Query Messages Notifications Data Output Query History

	s_id	user_name	food_category	food_name	food_qty	amount	s_date
1	1	Samantha	vegetarian	Bitter leaf	2 bundles	400	2022-02-09
2	2	Jules	carnavarian	Sharwama	2	600	2022-02-09
3	3	Peace	vegetarian	Bitter leaf	2 bundles	900	2022-02-09
4	4	Jules	carnavarian	Sharwama	2	500	2022-02-09
5	5	Peace	vegetarian	Bitter leaf	2 bundles	650	2022-02-09
6	6	Jules	carnavarian	Sharwama	2	800	2022-02-09
7	7	Charity	canivarian	Sharwama	2	900	2022-02-09

Conclusion

- The database design outlined above provides a foundation for building a food donation platform with essential features and functionalities. It allows users to interact with the system, donate or sell food items conveniently.