

# Database I

## PROJECT REPORT

To Be Presented To:  
Dr. Gamal Ibrahim Abdel Elshafy  
Eng. Sally Edward

|  
By:

Youssef Assem Mohammed	16P6064
Karim Walid Elhammady	16P3090
Ahmed Sameh Shahin	16P6063
Moataz Khalid Zakaria	16P8244
Mohammed Ehab Elsaheed	16P8160
Eslam Ahmed Genedy	17P6043



## Abstract

Database is absolutely an integral part of software system. The main aim of this project is providing information about the functioning of databases in a traditional banking system. Those databases should be able keep the day by day tally record as a complete banking. It can keep the information regarding account type, account opening form, deposits, and withdrawals. As well as keeping tabs on transactions, transaction reports, individual accounts opening forms, and group account.

## TABLE OF CONTENTS

### Contents

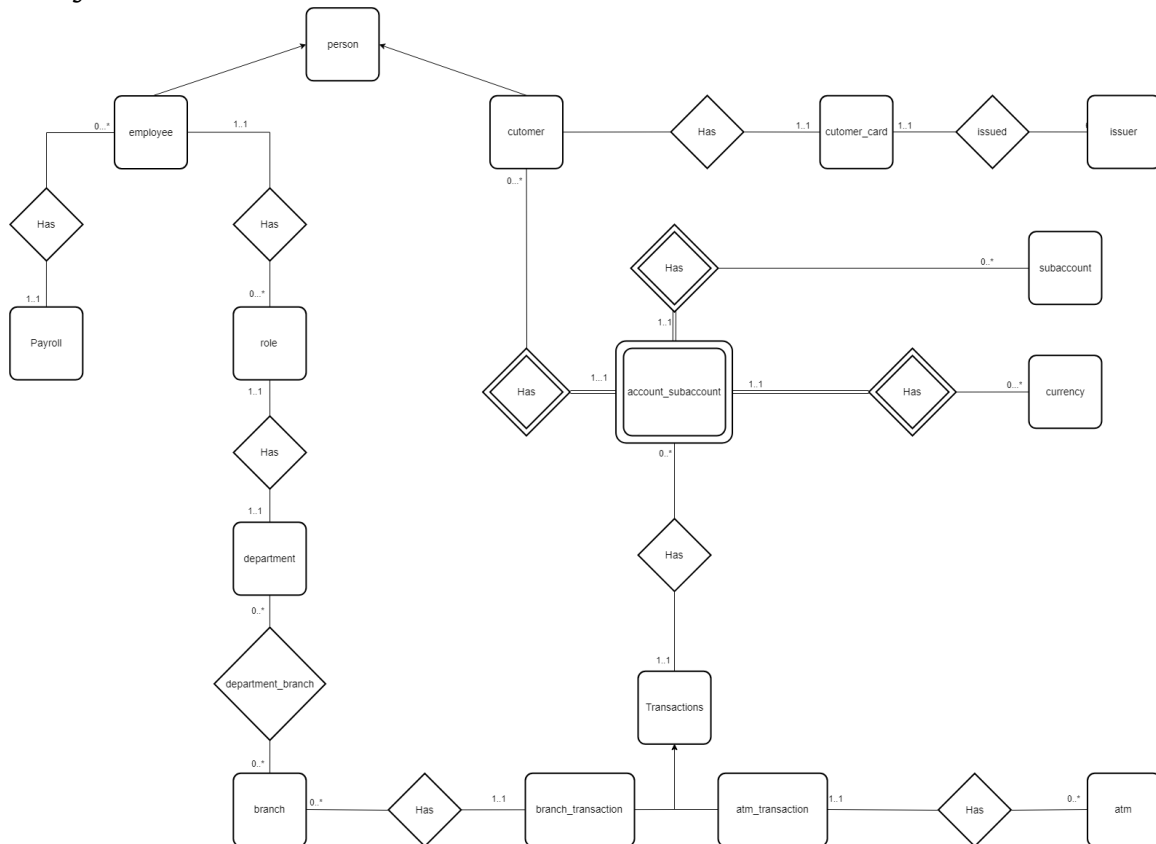
Abstract .....	2
The Code: .....	6
Log In.....	14
Registration .....	15
In employee panel: .....	15
In customer panel:.....	16
Services .....	16
New Subaccount Form.....	17
Issue Card Form .....	18
Withdraw Form.....	19
Deposit Form.....	20
Transfer Fund Form .....	21
Settings Form .....	22
View Transaction .....	23
Statistical Form .....	24
Issuers for Customers form.....	25
Query 1:.....	26
Query 2: .....	27
Query 3: .....	27
Query 4:.....	28
Query 5: .....	29
Query 6:.....	30
Query 7: .....	31

## TABLE OF FIGURES

Figure 1 ER diagram .....	5
Figure 2 Log-in Screen .....	14
Figure 3 Registration Screen.....	15
Figure 4 Registration Screen 3.....	15
Figure 5 Registration Screen 2 .....	15
Figure 6 Services Screen 2.....	16
Figure 7 Services Screen 1 .....	16
Figure 8 Services Screen 4 .....	16
Figure 9 Services Screen 3.....	16
Figure 10 New Accounts Screen.....	17
Figure 11 Issue Card Screen .....	18
Figure 12 Withdraw Screen.....	19
Figure 13 Deposit Screen .....	20
Figure 14 Funds Screen .....	21
Figure 15 Settings Screen.....	22
Figure 16 Transactions Screen .....	23
Figure 17 Statistical Screen.....	24
Figure 18 Issuers Screen .....	25

# ER Diagram

*Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities.*



*Figure 1 ER diagram*

Fully utilizing an ER Diagram in database engineering guarantees you to produce high quality database design to use in database creation, management and maintenance. An ER model also provides a means for communication.

# Physical Schema

*Physical schema is a term used in data management to describe how data is to be represented and stored (files, indices, et al.) in secondary storage using a particular database management system (DBMS) (e.g., Oracle RDBMS, Sybase SQL Server, etc.).*

THE CODE:

```
drop database if exists BANK;
```

```
create database BANK;
```

```
use BANK;
```

```
CREATE TABLE person (
```

```
    national_id INTEGER AUTO_INCREMENT,
```

```
    person_name VARCHAR(50),
```

```
    birth_date DATE,
```

```
    address VARCHAR(250),
```

```
    city VARCHAR(50),
```

```
    phone_number VARCHAR(15),
```

```
    email VARCHAR(25),
```

```
    PRIMARY KEY (national_id)
```

```
) ENGINE=INNODB;
```

```
CREATE TABLE sub_accounts (
```

```
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
```

```
    acc_name VARCHAR(50),
```

```
    description TINYTEXT,
```

```
    interest_rate DOUBLE,
```

```
    interest_interval INTEGER
```

```
) ENGINE=INNODB;
```

```
CREATE TABLE currencies (
```

```
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
```

```

    triple_code CHAR(3),
    exchange_rate DOUBLE
) ENGINE=INNODB;

CREATE TABLE issuers (
    id INTEGER AUTO_INCREMENT,
    issuerName VARCHAR(25),
    PRIMARY KEY (id)
) ENGINE=INNODB;

CREATE TABLE ATMs (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    address VARCHAR(50),
    balance DOUBLE
) ENGINE=INNODB;

CREATE TABLE branches (
    SWIFT VARCHAR(11) PRIMARY KEY,
    b_name VARCHAR(50),
    phone_number VARCHAR(15),
    postal_code VARCHAR(5),
    address VARCHAR(50),
    city VARCHAR(10),
    opening DATE
) ENGINE=INNODB;

CREATE TABLE departments (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    d_name VARCHAR(50)
) ENGINE=INNODB;

```



```

CREATE TABLE department_branch (
    branch_swift VARCHAR(11),
    dept_id INTEGER,
    CONSTRAINT FOREIGN KEY (dept_id)
        REFERENCES departments (id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (branch_SWIFT)
        REFERENCES branches (SWIFT)
        ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY(branch_swift,dept_id)
) ENGINE=INNODB;

CREATE TABLE customers (
    id int,
    account_id INTEGER PRIMARY KEY AUTO_INCREMENT,
    CONSTRAINT FOREIGN KEY (id)
        REFERENCES person (national_id)
        ON UPDATE CASCADE ON DELETE CASCADE
) ENGINE=INNODB;

CREATE TABLE roles (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    r_name VARCHAR(50),
    dept_id INTEGER,
    CONSTRAINT FOREIGN KEY (dept_id)
        REFERENCES departments (id)
        ON UPDATE CASCADE ON DELETE CASCADE
) ENGINE=INNODB;

```

```

CREATE TABLE employees (
    id INTEGER PRIMARY KEY,
    salary DOUBLE,
    role_id INTEGER,
    username varchar(45) UNIQUE,
    user_password varchar(45),
    CONSTRAINT FOREIGN KEY (id)
        REFERENCES person (national_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (role_id)
        REFERENCES roles (id)
        ON UPDATE CASCADE ON DELETE CASCADE
) ENGINE=INNODB;

CREATE TABLE accounts_sub_accounts (
    account_id INTEGER,
    subaccount_id INTEGER AUTO_INCREMENT,
    currency_id INTEGER,
    balance DOUBLE,
    PRIMARY KEY (account_id , subaccount_id , currency_id),
    CONSTRAINT FOREIGN KEY (account_id)
        REFERENCES customers (account_id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (subaccount_id)
        REFERENCES sub_accounts (id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (currency_id)

```

```

REFERENCES currencies (id)

ON UPDATE CASCADE ON DELETE CASCADE

) ENGINE=INNODB;

CREATE TABLE payroll (

id INTEGER PRIMARY KEY AUTO_INCREMENT,

employee_id INTEGER,

payment_date DATE,

bonuses DOUBLE,

penalties DOUBLE,

CONSTRAINT FOREIGN KEY (employee_id)

REFERENCES employees (id)

ON UPDATE CASCADE ON DELETE CASCADE

) ENGINE=INNODB;

create TABLE transactions (

id INTEGER PRIMARY KEY AUTO_INCREMENT,

from_account INTEGER,

from_subAccount INTEGER,

to_account INTEGER,

to_subAccount INTEGER,

amount DOUBLE,

t_time DATETIME,

CONSTRAINT FOREIGN KEY (from_account,from_subaccount)

REFERENCES accounts_sub_accounts (account_id,subaccount_id)

ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (to_account,to_subAccount)

REFERENCES accounts_sub_accounts (account_id,subaccount_id)

```

```

        ON UPDATE CASCADE ON DELETE CASCADE
    ) ENGINE=INNODB;

CREATE TABLE atm_transaction (
    atm_id INT,
    transaction_id INT,
    CONSTRAINT FOREIGN KEY (atm_id)
        REFERENCES atms (id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (transaction_id)
        REFERENCES transactions (id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY(atm_id,transaction_id)
) ENGINE=INNODB;

CREATE TABLE branch_transaction (
    branch_swift VARCHAR(11),
    transaction_id INT,
    CONSTRAINT FOREIGN KEY (branch_swift)
        REFERENCES branches (swift)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (transaction_id)
        REFERENCES transactions (id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY(branch_swift,transaction_id)
) ENGINE=INNODB;

CREATE TABLE customer_cards (
    card_number INTEGER PRIMARY KEY AUTO_INCREMENT,

```

```
account_id INTEGER,  
issuer_id INT,  
issue_date DATE,  
expirey_date DATE,  
daily_limit DOUBLE,  
card_type ENUM('debit', 'credit'),  
CVV VARCHAR(3),  
CONSTRAINT FOREIGN KEY (issuer_id)  
    REFERENCES issuers (id)  
    ON UPDATE CASCADE ON DELETE CASCADE,  
CONSTRAINT FOREIGN KEY (account_id)  
    REFERENCES customers (account_id)  
    ON UPDATE CASCADE ON DELETE CASCADE  
) ENGINE=INNODB;
```

# The Logical Schema

*A logical schema is a design-centric database structure built to meet your business requirements. It is a model that exists on a white board or in a diagramming tool. It is like the architect's drawings of your database.*

Legend:

- Bold Underlined => Primary Key (may compose of foreign keys)
- Only Bold => Foreign Key

person (**national\_id**, person\_name, birth\_date, address, city, phone\_number, email)

sub\_accounts (**id**, acc\_name, description, interest\_rate, interest\_interval)

currencies (**id**, triple\_code, exchange\_rate)

issuers (**id**, issuerName)

ATMs (**id**, address, balance)

branches (**SWIFT**, b\_name, phone\_number, postal\_code, address, city, opening)

departments (**id**, d\_name)

department\_branch (**branch swift**, **dept id**)

customers (**id**, account\_id)

roles (**id**, r\_name, **dept id**)

employees (**id**, salary, **role id**, username, user\_password)

accounts\_sub\_accounts (**account id**, **subaccount id**, **currency id**, balance)

payroll (**id**, **employee id**, payment\_date, bonuses, penalties)

transactions (**id**, **from\_account**, **from\_subAccount**, **to\_account**, **to\_subAccount**, amount, t\_time)

atm\_transaction (atm\_id, **transaction id**)

branch\_transaction (branch\_swift, **transaction id**)

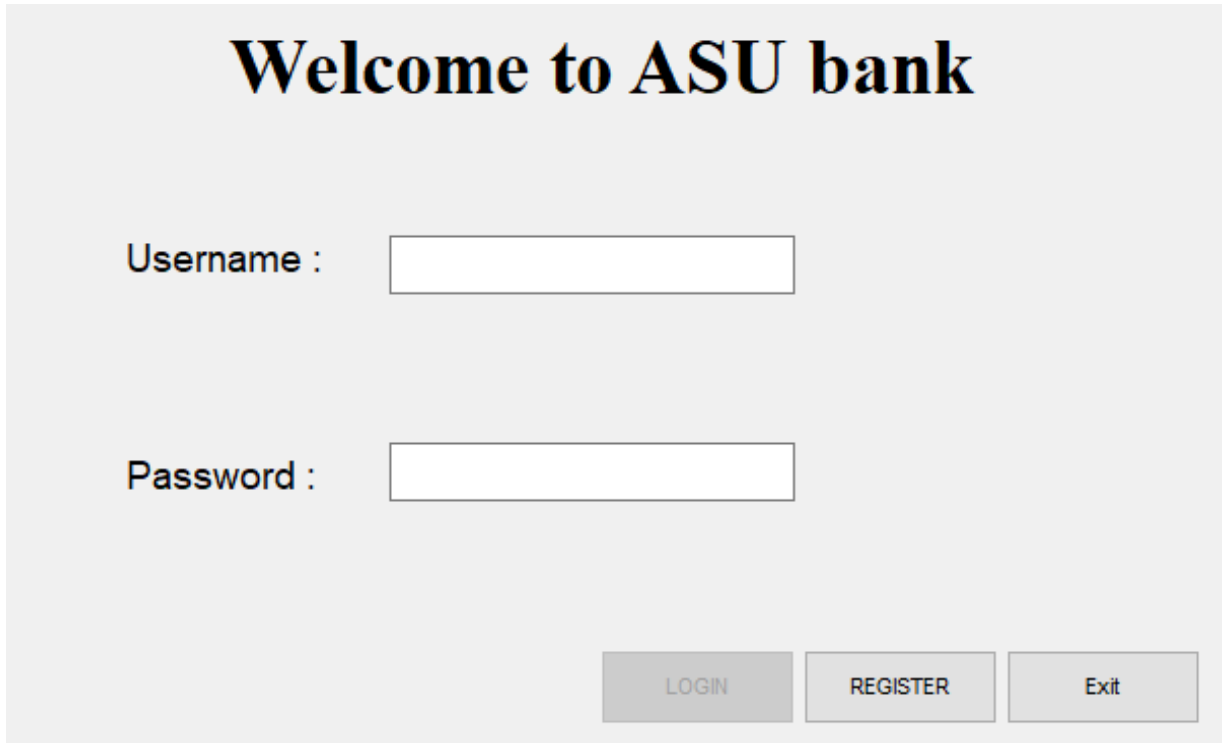
customer\_cards (**card number**, **account id**, **issuer id**, issue\_date, expiry\_date, daily\_limit, card\_type, CVV)

# User Guide

## Log In

The first screen all users will see is the login screen.

Employees can register or login to their accounts using this screen.

The image shows a login screen for ASU bank. At the top, the text "Welcome to ASU bank" is displayed in a large, bold, black serif font. Below this, there are two input fields. The first is labeled "Username :" in a black serif font, followed by a white rectangular input box with a thin black border. The second is labeled "Password :" in the same font, followed by a similar white rectangular input box. At the bottom right of the screen, there are three buttons: "LOGIN", "REGISTER", and "Exit". Each button is a light gray rectangle with a thin black border and the text is in a small, black, sans-serif font.

*Figure 2 Log-in Screen*

## Registration

The Registration form allows for both employee and customer registration.

The first tab takes personal information about the person registering in the bank.

The screenshot shows a web form titled "Registration". At the top right is a "Back" button. Below the title are two tabs: "Personal Information" (which is active) and "Additional Information". The form contains several input fields: "Name", "Address", "National ID", "City", "Phone Number", "E- mail", and "Birthdate". The "Birthdate" field is populated with "2018-12-20 01:56:00" and has a calendar icon on the right.

Figure 3 Registration Screen

The second tab gives the option either to register as an employee or a customer.

This screenshot shows the "Registration" form with the "Additional Information" tab selected. It features two radio buttons: "Employee" (which is selected) and "Customer". A "Register" button is located to the right. Below these are four input fields: "Username", "Salary", "Password", and "Role ID" (which is a dropdown menu).

Figure 5 Registration Screen 2

This screenshot shows the "Registration" form with the "Additional Information" tab selected. It features two radio buttons: "Employee" and "Customer" (which is selected). A "Register" button is located to the right. Below these is a single input field labeled "Account ID".

Figure 4 Registration Screen 3

### IN EMPLOYEE PANEL:

- Username will be the employee's username to log into system.
- Password will be the employee's password to log into system.
- Salary is the employee's salary.
- Role ID is his role in the bank (Manager, Teller, Bookkeeper, financial manager).



IN CUSTOMER PANEL:

- Account ID text field will show the new customer's account ID (his account number) after the new creation.

## Services

After Login, all employees will go to the service page. However, according to each employee's role, the corresponding services will be available.

- Financial managers: can create sub account for customers and issue them card.
- Teller: allow customers to withdraw and deposit funds from and to their accounts.
- Bookkeeper: manages the transfer of funds between accounts.
- Manager: allow employees to adjust their personal information as well as their account username and password.

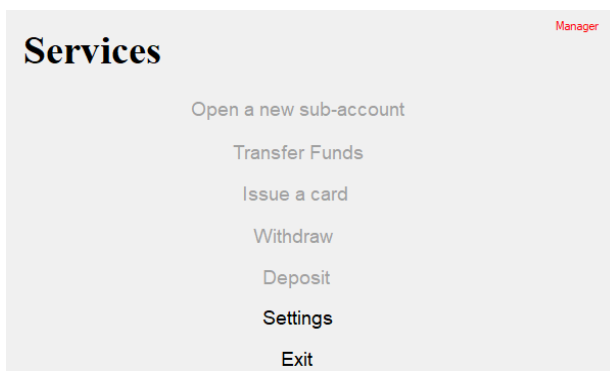


Figure 7 Services Screen 1

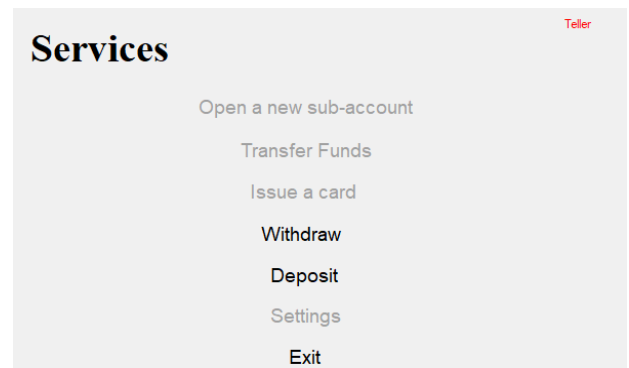


Figure 6 Services Screen 2

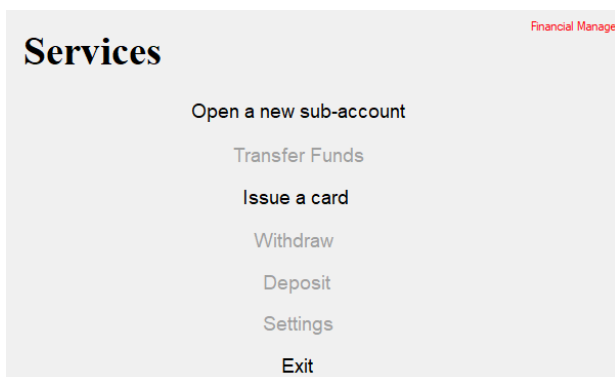


Figure 9 Services Screen 3

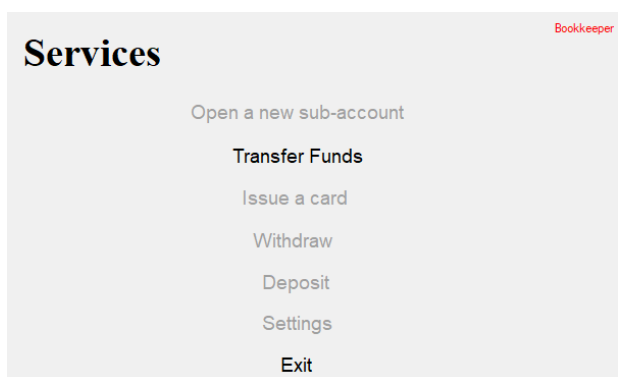
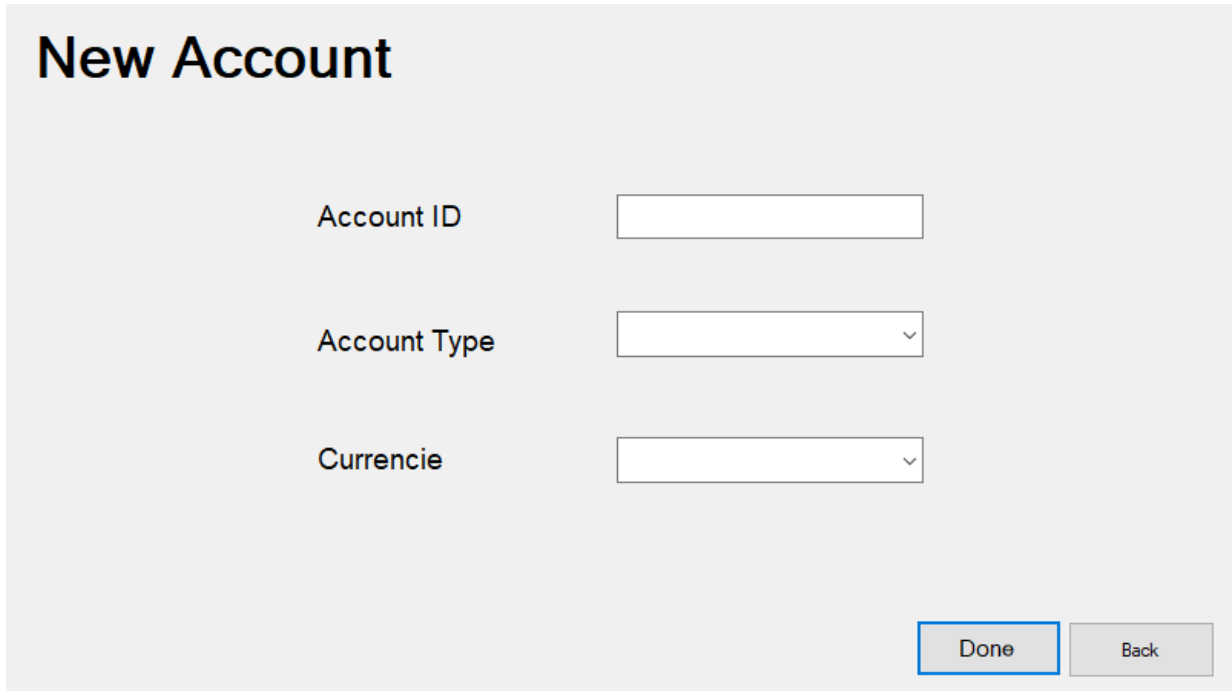


Figure 8 Services Screen 4

## New Subaccount Form

In this form the financial manager can create a new sub account for a customer.

The customer must tell the financial manager his account ID and what type of account he wants (checking, saving, dividend and current account) and which currency he wants his account in.

The image shows a screenshot of a web form titled "New Account" in a large, bold, black font at the top left. Below the title, there are three input fields arranged vertically. The first field is labeled "Account ID" and is a simple text box. The second field is labeled "Account Type" and is a dropdown menu with a small downward arrow on the right. The third field is labeled "Currencie" and is also a dropdown menu with a small downward arrow on the right. At the bottom right of the form, there are two buttons: a "Done" button with a blue border and a "Back" button with a gray border. The entire form is set against a light gray background.

**New Account**

Account ID

Account Type

Currencie

[Done](#) [Back](#)

*Figure 10 New Accounts Screen*

## Issue Card Form

When a customer needs to create a new card, he has to go to the financial manager who will ask him for his Account ID and which issuer he want to create the card with (visa, Mastercard, American Express, chase, discover, Citibank, capital one) , ,the card duration ( 3 , 5 or 7 years),Daily limit for card and finally the card type(debit or credit).

**Issue Card**

Account ID

Issuer

Issue Date  Card Duration

Expiry Date

Daily Limit

Card Type

Figure 11 Issue Card Screen

## Withdraw Form

The teller is responsible for depositing and withdrawing money from customers' accounts.

In this Form, the customer tells the teller his account number and which account he wants to withdraw form and the amount to withdraw.

The image shows a web form titled "Withdraw" in a large, bold, black font. Below the title, there are four input fields arranged vertically. The first field is labeled "Account ID" and is a standard text input box. The second field is labeled "Account Type" and is a dropdown menu with a small downward arrow on the right. The third field is labeled "Currencie" (note the spelling) and is also a dropdown menu with a small downward arrow. The fourth field is labeled "Amount" and is a standard text input box. At the bottom right of the form, there are two buttons: a blue button with the text "Withdraw" and a grey button with the text "Back".

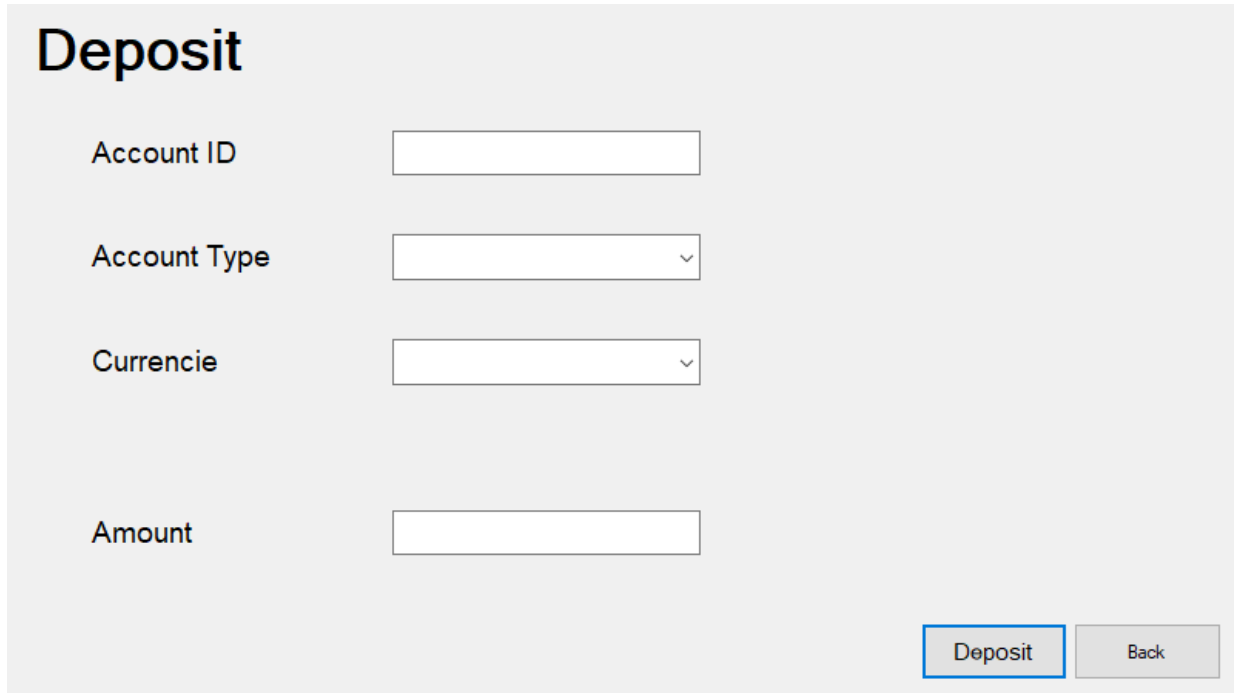
Account ID	<input type="text"/>
Account Type	<input type="text"/>
Currencie	<input type="text"/>
Amount	<input type="text"/>

WithdrawBack

*Figure 12 Withdraw Screen*

## Deposit Form

In this Form, the customer tells the teller his account number and which account he wants to deposit in and the amount to deposit.

A screenshot of a web-based deposit form. The form has a light gray background and a title 'Deposit' in bold black text at the top left. Below the title, there are four input fields arranged vertically. The first field is labeled 'Account ID' and is a text input. The second field is labeled 'Account Type' and is a dropdown menu with a small downward arrow. The third field is labeled 'Currencie' and is a dropdown menu with a small downward arrow. The fourth field is labeled 'Amount' and is a text input. At the bottom right of the form, there are two buttons: 'Deposit' and 'Back'. The 'Deposit' button is highlighted with a blue border, while the 'Back' button is a standard gray button.

**Deposit**

Account ID

Account Type

Currencie

Amount

[Deposit](#) [Back](#)

*Figure 13 Deposit Screen*

## Transfer Fund Form

In this form, the bookkeeper can transfer fund from an account to another

# Transfer form

From	Account	<input type="text"/>	Sub account	<input type="text"/>
To	Account	<input type="text"/>	Sub account	<input type="text"/>

Amount

Authorization code : LO2B12

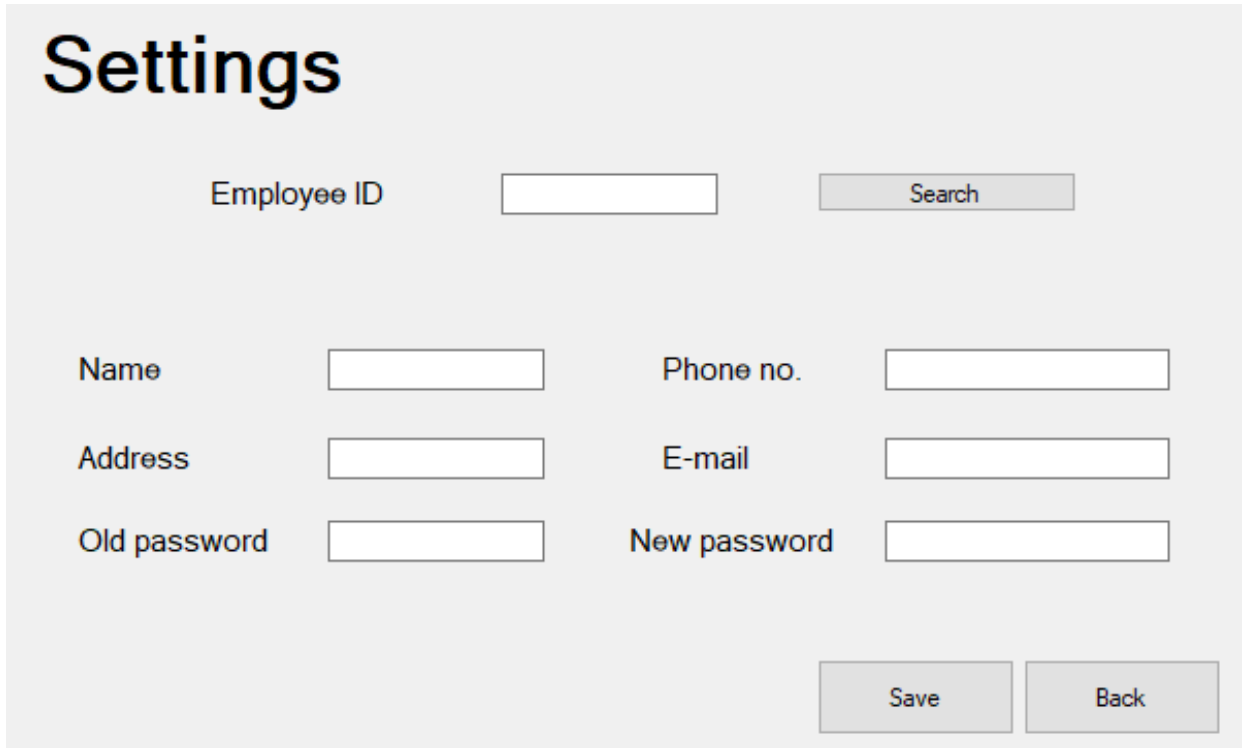
Transfer

Back

Figure 14 Funds Screen

## Settings Form

The manager is the only individual with access to the settings forms, he can change personnel information as well as username and password of employee account.



The screenshot shows a web-based 'Settings' form. At the top left, the word 'Settings' is displayed in a large, bold, black font. Below this, there is a search section with the label 'Employee ID' followed by a white rectangular input field and a grey button labeled 'Search'. The main body of the form contains six input fields arranged in three rows. The first row has 'Name' and 'Phone no.' labels. The second row has 'Address' and 'E-mail' labels. The third row has 'Old password' and 'New password' labels. Each label is followed by a white rectangular input field. At the bottom right of the form, there are two grey buttons: 'Save' and 'Back'.

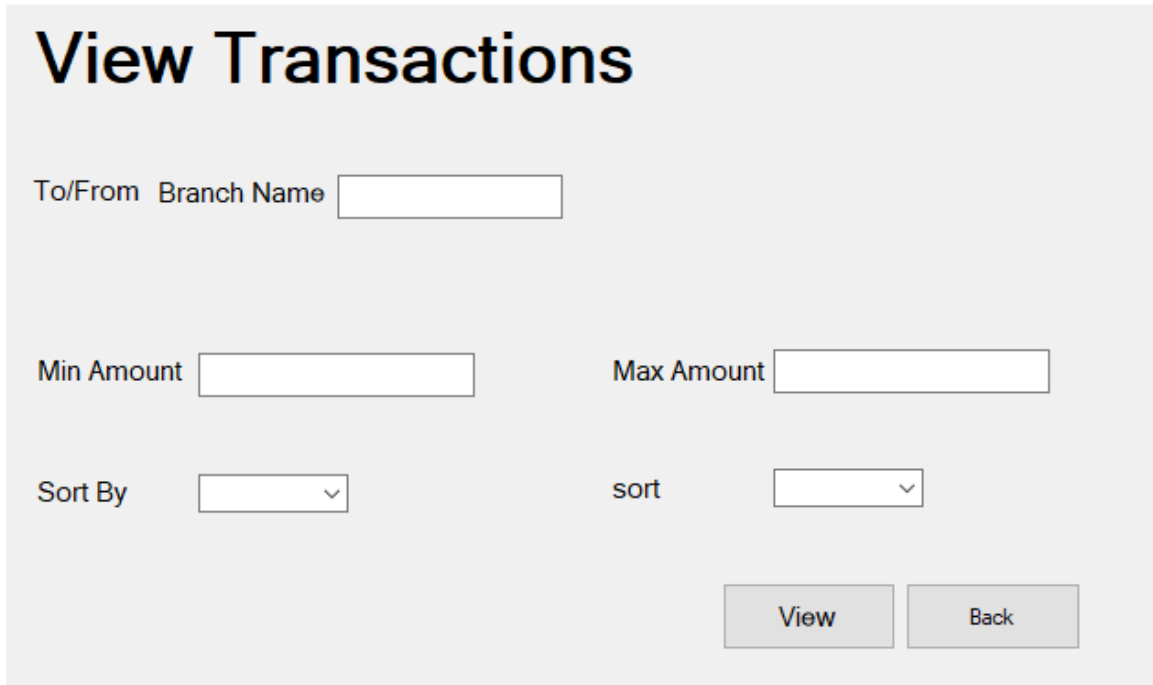
<b>Settings</b>			
Employee ID	<input type="text"/>	<input type="button" value="Search"/>	
Name	<input type="text"/>	Phone no.	<input type="text"/>
Address	<input type="text"/>	E-mail	<input type="text"/>
Old password	<input type="text"/>	New password	<input type="text"/>
		<input type="button" value="Save"/>	<input type="button" value="Back"/>

*Figure 15 Settings Screen*

## View Transaction

In this form, the employee can view all transactions in a specific branch and filter them according to the amount.

He can also filter sort them by customer name or by the transaction amount and sort them accordingly.



**View Transactions**

To/From Branch Name

Min Amount  Max Amount

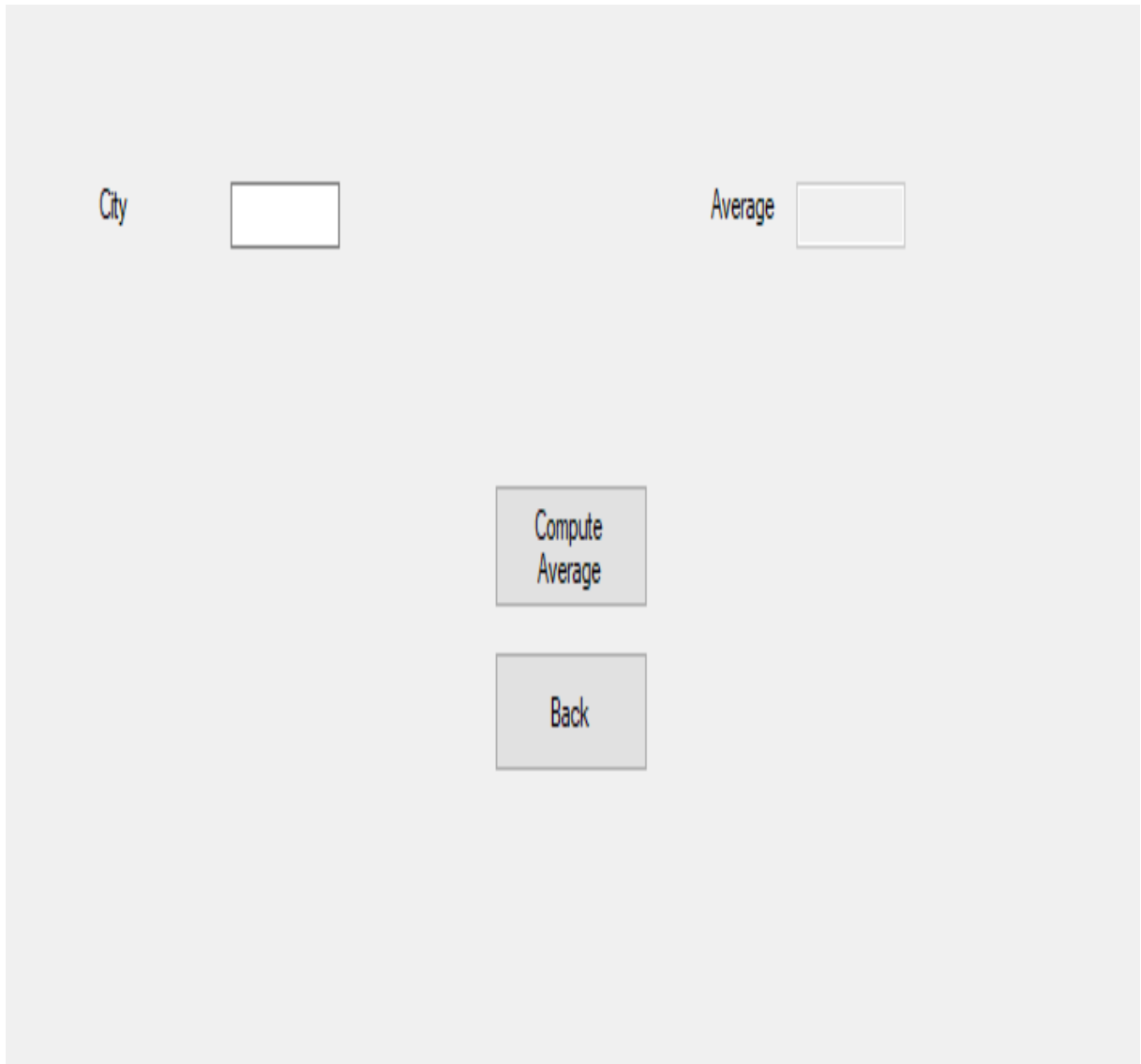
Sort By  sort

*Figure 16 Transactions Screen*



## Statistical Form

This form provides statistical information, it computes the average of the balances in a specific city.



The image shows a graphical user interface for a statistical form. It features a light gray background. At the top left, the label "City" is positioned next to a white rectangular input field. To the right of this, the label "Average" is positioned next to another white rectangular input field. Below these two input fields, centered horizontally, are two buttons. The first button is labeled "Compute Average" and the second button, located directly below the first, is labeled "Back". Both buttons have a light gray background and a thin black border.

*Figure 17 Statistical Screen*

## Issuers for Customers form

In this form the employee can view all customers in a specific city holding a card (debit or credit) from a specific issuer.

# Issuers For Customers in a city

Issuer

Customer's City

*Figure 18 Issuers Screen*

# Complex Queries

## Query 1:

*Gets national ID, name of customers living in a specified city with cards issued by a specified issuer*

SELECT

national\_id, person\_name

FROM

person

WHERE

national\_id IN (SELECT

id

FROM

customers

WHERE

account\_id IN (SELECT

account\_id

FROM

customer\_cards

WHERE

issuer\_id IN (SELECT

id

FROM

issuers

WHERE

```
issuename LIKE 'Visa'))))  
  
AND city LIKE 'Giza';
```

### Query 2:

*Find average of balances for all customers living in a specified city.*

```
SELECT  
  
    AVG(balance) AS Average  
  
FROM  
  
    accounts_sub_accounts  
  
WHERE  
  
    account_id IN (SELECT  
  
        account_id  
  
    FROM  
  
        customers  
  
    WHERE  
  
        id IN (SELECT  
  
            national_id  
  
        FROM  
  
            person  
  
        WHERE  
  
            city LIKE 'Aswan')));
```

### Query 3:

*Gets national ID, name, “from” account, “to” account, amount of transactions done at a specified branch with options to specify either or both minimum/maximum of amount and the option to order the result ascendingly or descendingly by person\_name or amount (same applies to queries 4 to 7).*

```
SELECT  
  
    national_id, person_name, from_account, to_account, amount
```

```

FROM
    (person
    INNER JOIN customers ON person.national_id = customers.id
    INNER JOIN (SELECT
        *
    FROM
        transactions AS Tfrom
    WHERE
        Tfrom.id IN (SELECT
            transaction_id
        FROM
            branch_transaction
        WHERE
            branch_transaction.transaction_id IN (SELECT
                SWIFT
            FROM
                branches
            WHERE
                b_name LIKE 'AlexPlaza')))) AS T ON T.from_account =
customers.account_id)
WHERE
    amount >= 500 AND amount <= 5000

```

#### Query 4:

```

SELECT
    national_id, person_name, from_account, to_account, amount
FROM

```

```

(person
INNER JOIN customers ON person.national_id = customers.id
INNER JOIN (SELECT
    *
FROM
    transactions AS Tfrom
WHERE
    Tfrom.id IN (SELECT
        transaction_id
    FROM
        branch_transaction
    WHERE
        branch_transaction.transaction_id IN (SELECT
            SWIFT
        FROM
            branches
        WHERE
            b_name LIKE 'AlexPlaza')))) AS T ON T.from_account =
customers.account_id)
WHERE
    amount >= 500 AND amount <= 5000
ORDER BY person_name ASC

```

### Query 5:

```

SELECT
    national_id, person_name, from_account, to_account, amount
FROM

```

```

(person
INNER JOIN customers ON person.national_id = customers.id
INNER JOIN (SELECT
    *
FROM
    transactions AS Tfrom
WHERE
    Tfrom.id IN (SELECT
        transaction_id
    FROM
        branch_transaction
    WHERE
        branch_transaction.transaction_id IN (SELECT
            SWIFT
        FROM
            branches
        WHERE
            b_name LIKE 'AlexPlaza')))) AS T ON T.from_account =
customers.account_id)

```

### Query 6:

```

SELECT
    national_id, person_name, from_account, to_account, amount
FROM
    (person
    INNER JOIN customers ON person.national_id = customers.id
    INNER JOIN (SELECT

```

```

*

FROM

    transactions AS Tfrom

WHERE

    Tfrom.id IN (SELECT

        transaction_id

    FROM

        branch_transaction

    WHERE

        branch_transaction.transaction_id IN (SELECT

            SWIFT

        FROM

            branches

        WHERE

            b_name LIKE 'AlexPlaza')) AS T ON T.from_account =
customers.account_id)

WHERE

    amount >= 500

```

### Query 7:

```

SELECT

    national_id, person_name, from_account, to_account, amount

FROM

    (person

    INNER JOIN customers ON person.national_id = customers.id

    INNER JOIN (SELECT

        *

```



```
FROM
    transactions AS Tfrom
WHERE
    Tfrom.id IN (SELECT
        transaction_id
    FROM
        branch_transaction
    WHERE
        branch_transaction.transaction_id IN (SELECT
            SWIFT
        FROM
            branches
        WHERE
            b_name LIKE 'AlexPlaza')) AS T ON T.from_account =
customers.account_id)
WHERE
    amount <= 5000
ORDER BY person_name ASC
```