НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Кафедра автоматизованих систем обробки інформації і управління

**КУРСОВА РОБОТА**

з дисципліни

**«Бази даних»**

Виконав: *Бакун Нікіта*

Група: ІП-96

Керівник: ст.вик. Недашківський Є. А.

**Зміст**

[**ВСТУП** 3](#_Toc58883911)

[1.1 Постановка задачі 4](#_Toc58883912)

[1.2 Завдання 4](#_Toc58883913)

[**2. Основна частина** 5](#_Toc58883914)

[2.1 створення ER діаграми 5](#_Toc58883915)

[2.2 Створення діаграми класів 7](#_Toc58883916)

[**3. Створення таблиць та зв’язків між ними** 8](#_Toc58883917)

[3.1 Створення БД «Облік клієнтів банку» 8](#_Toc58883918)

[3.2 Заповнення бази даних 10](#_Toc58883919)

[**5.Створення об’єктів бази даних** 23](#_Toc58883920)

[5.1. Створення представлення 23](#_Toc58883921)

[5.2 Створення табличної функції 23](#_Toc58883922)

[5.3 Створення скалярної функції 24](#_Toc58883923)

[5.4 Створення збереженої процедури 24](#_Toc58883924)

[5.6 Використання «UNION» 25](#_Toc58883925)

[5.7 Використання «CASE» 25](#_Toc58883926)

[**6. Створення нового користувача** 26](#_Toc58883927)

[**ВИСНОВКИ** 27](#_Toc58883928)

[Список літератури: 28](#_Toc58883929)

# ВСТУП

База даних - це система певних матеріалів, що використовується для зручної обробки інформації з можливістю її пошуку та обробки. Особливе місце в курсовій роботі відведено вивченню мови програмування SQL. В рамках курсової роботи розгляну такі поняття, як представлення, процедури, функції, тригери, індексування даних. Також будуть розглянуті підходи до обробки даних.

Як ми знаємо, практично всі програмні продукти використовують бази даних, які необхідні в роботі як простих, розрахованих на одного користувача додатків, так і великих інформаційних систем. З кожним днем ​​потік інформації постійно збільшується, а при обробці великого обсягу даних виникає проблема швидкості доступу до даних. Очевидний спосіб підвищення швидкості роботи серверної частини, це підвищення обсягу обчислювальних ресурсів, проте такий спосіб має головний недолік у вигляді обмеженості фізичних розмірів.

Для більш зручного використання застосовують PostgreSQL - це система управління базами даних на платформі Unix. Вона включає в себе сортування та обробку данних. SQL, частина PostgreSQL, сумісна з "Structured Query Language" - найрозповсюдженішим стандартом мови, уживаної для доступу до БД.

Для ознайомлення з можливостями PostgreSQL я обрав тему курсової роботи «Облік клієнтів банку».

Саме цей проект допоміг мені розібратися в проектуванні БД, побудові діаграми ERі діаграми класів, її нормалізаціі, заповненні

Також важливою частиною цього проекту є створення інтерфейсу користувача, який допоможе швидше виконувати маніпуляції з БД. Дані форми розроблені для додавання сутностей в БД, редагування існуючих, видалення вже наявних та можливість виводу відсортованих данних за певними критеріями.

## 1.1 Постановка задачі

Метою даної роботи є створення БД «Облік клієнтів банку» у третій нормальній формі (нормалізація, якщо потрібно). Створення запитів та інтерфейсу користувача для того щоб( облік клієнтів банку)

## 1.2 Завдання

1. На основі предметної області в роботі потрібно реалізувати :
2. реляційну базу даних у третій нормальній формі. Структура узгоджена з керівником. База включає щонайменше шість таблиць з не менш ніж десятьма рядками у кожній для репрезентації даних. Базу даних доповнює застосунок, що дозволяє вивести на екран вміст однієї з таблиць.
3. весь попередній пункт, але застосунок дозволяє проводити фільтрацію виведення (наприклад вказавши діапазон дат, чи конкретне значення заданого атрибуту). Необхідно також додати можливість посторінкового виведення даних задавши кількість рядків на сторінку та можливість обрати номер сторінки. Передбачено інтерфейс користувача.
4. включно з попередніми пунктами, а також реалізація модифікації даних (додавання, редагування, видалення) через інтерфейс користувача.
5. включно з попередніми пунктами структура проекту на рівні бази даних повинна містити наступні об’єкти:
   * Представлення (view)
   * Табличну та скалярну функції
   * Збережену процедуру
   * Тригер
   * Принаймні одна з таблиць повинна передбачати значення «за замовчанням»
   * Принаймні в одному з вищеперечислених об’єктів повинні бути присутні оператори JOIN, UNION, CASE, використано під запит

6.включає всі попередні пункти. Передбачити окремого користувача для застосунку, що не матиме прав на редагування структури бази даних, проте матиме досить прав для роботи. Курсовий проект здано до дедлайну (початок сесії) та виконано у єдиному стилі написання коду (coding style). Відсутні зауваження щодо оформлення.

# **2. ОСНОВНА ЧАСТИНА**

## 2.1 створення ER діаграми

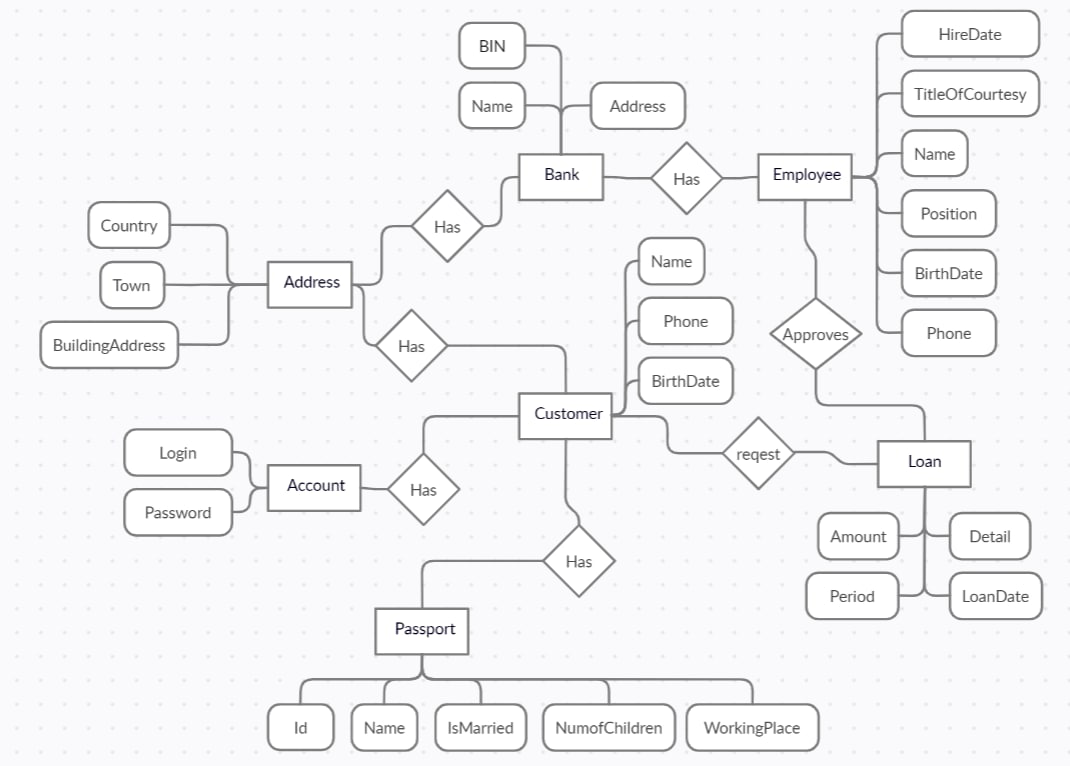
#### 

|  |  |
| --- | --- |
| **Сутність** | **Опис** |
| Address | Містить інформацію про адресу банку або клієнта |
| Bank | Включає інформацію про банк, його адреса та співробітники |
| Account | Інформація про акаунт користувача |
| Customer | Інформація про користувача, його адрес, акаунт, паспорт, кредит |
| Passport | Приватна інформація клієнта, яку беруть з паспорта |
| Loan | Ключова інформація про позику користувача , яку затверджує співробітник |
| Employee | Інформація про співробітника, позики які він одобрив та банк в якому працює |

*Таблиця 2.1 Опис сутностей*

|  |  |  |
| --- | --- | --- |
| **Перша сутність** | **Тип зв'язку** | **Друга сутність** |
| Address | Один до одного | Bank |
| Bank | Один до багатьох | Employee |
| Employee | Багато до багатьох | Loan |
| Loan | Один до багатьох | Customer |
| Customer | Один до одного | Address |
| Customer | Один до одного | Passport |
| Customer | Один до одного | Account |

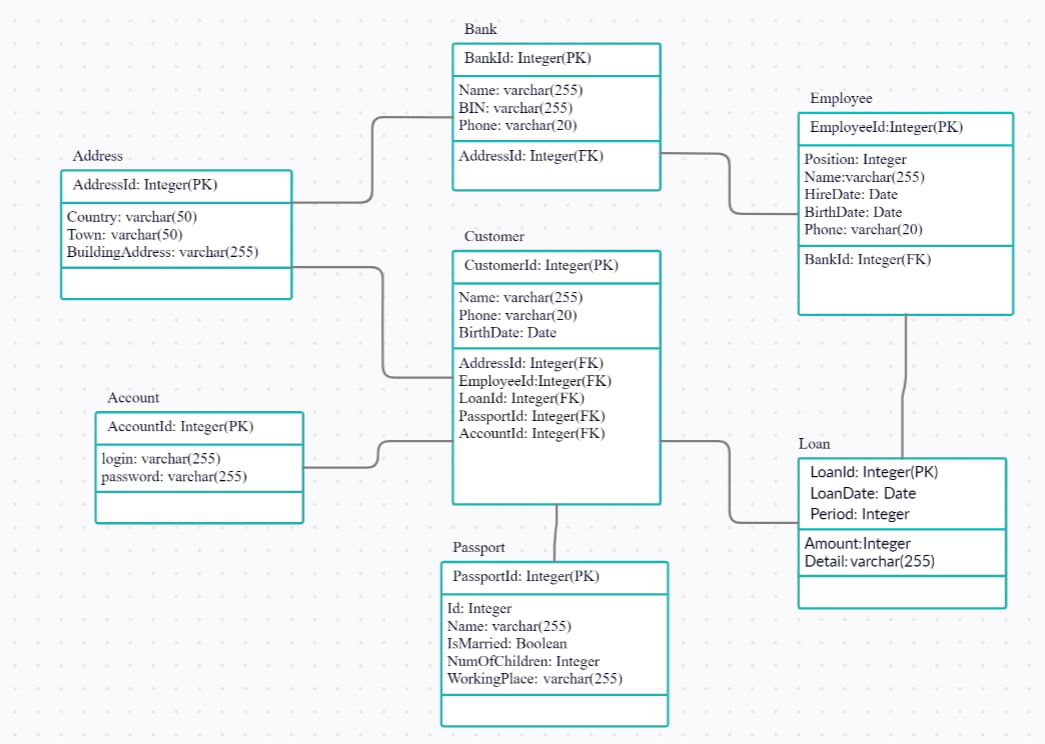
*Таблиця 2.2 Зв'язки між сутностями*



*Рисунок 2.1 ER діаграма «Облік клієнтів банку»*

## 2.2 Створення діаграми класів

На основі попердньої діаграми (ER діаграма «Облік клієнтів банку») побудуємо модель фізичної БД у третій нормальній формі



*Рисунок 2.2 Модель фізичної БД у третій нормальній формі*

# **3. Створення таблиць та зв’язків між ними**

## 3.1 Створення БД «Облік клієнтів банку»

1. **CREATE** **DATABASE** BankSystem

4. **CREATE** **TABLE** IF NOT EXISTS Address(
5. AddressId SERIAL **PRIMARY** **KEY** re,
6. Country **VARCHAR**(255) NOT NULL **DEFAULT** 'Ukraine',
7. Town **VARCHAR**(255) NOT NULL,
8. BuildingAddress **VARCHAR**(255) NOT NULL
9. );

12. **CREATE** **TABLE** IF NOT EXISTS Bank(
13. BankId SERIAL **PRIMARY** **KEY**,
14. BIN **VARCHAR**(255) NOT NULL,
15. Phone **VARCHAR**(255),
16. **Name** **VARCHAR**(255) **DEFAULT** 'NULL',
17. Address\_Id **Integer** NOT NULL,
18. **FOREIGN** **KEY**(Address\_Id) **REFERENCES** Address(AddressId)
19. );
21. **CREATE** **TABLE** IF NOT EXISTS Employee(
22. EmployeeId SERIAL **PRIMARY** **KEY**,
23. Position **Integer** NOT NULL,
24. **Name** **VARCHAR**(255) NOT NULL,
25. HireDate **DATE** NOT NULL,
26. BirthDate **DATE**,
27. Phone **VARCHAR**(20),
28. Bank\_Id **Integer** NOT NULL,
29. **FOREIGN** **KEY**(Bank\_Id) **REFERENCES** Bank(BankId)
30. );
32. **CREATE** **TABLE** IF NOT EXISTS Customer(
33. CustomerID  SERIAL **PRIMARY** **KEY**,
34. **Name** **VARCHAR**(255) NOT NULL,
35. Phone **VARCHAR**(20) **DEFAULT** 'NULL',
36. BirthDate **DATE** NOT NULL,
37. Address\_Id **Integer** NOT NULL,
38. **FOREIGN** **KEY**(Address\_Id) **REFERENCES** Address(AddressId),
39. Employee\_Id **Integer** NOT NULL,
40. **FOREIGN** **KEY**(Employee\_Id) **REFERENCES** Employee(EmployeeId),
41. Loan\_Id **Integer** NOT NULL,
42. **FOREIGN** **KEY**(Loan\_Id) **REFERENCES** Loan(LoanId),
43. Passport\_Id **Integer** NOT NULL,
44. **FOREIGN** **KEY**(Passport\_Id) **REFERENCES** Passport(PassportId),
45. Account\_Id **Integer** NOT NULL,
46. **FOREIGN** **KEY**(Account\_Id) **REFERENCES** Account(AccountId)
47. );
49. **CREATE** **TABLE** IF NOT EXISTS Account(
50. AccountID  SERIAL **PRIMARY** **KEY**,
51. Login **VARCHAR**(255) NOT NULL,
52. **Password** **VARCHAR**(255) NOT NULL
53. );

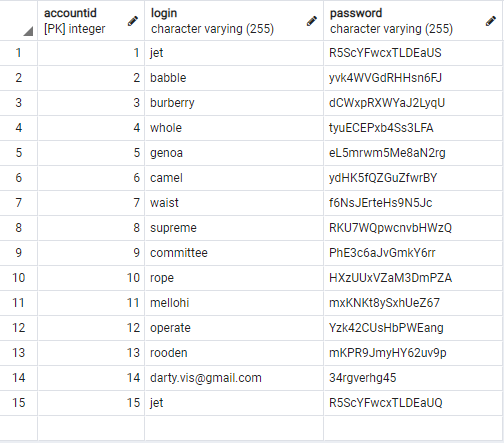
56. **CREATE** **TABLE** IF NOT EXISTS Passport(
57. PassportID  SERIAL **PRIMARY** **KEY**,
58. **Name** **VARCHAR**(255) NOT NULL,
59. IsMarried BOOL NOT NULL,
60. NumOfChildren **Integer** NOT NULL,
61. WorkingPlace **VARCHAR**(255) NOT NULL
62. );

65. **CREATE** **TABLE** IF NOT EXISTS Loan(
66. LoanId  SERIAL **PRIMARY** **KEY**,
67. LoanDate **Date** NOT NULL,
68. Period **Integer** NOT NULL,
69. Amount **Integer** NOT NULL,
70. Detail **VARCHAR**(255) NOT NULL
71. );

## 3.2 Заповнення бази даних

3.2.1 Заповнення таблиці «account»

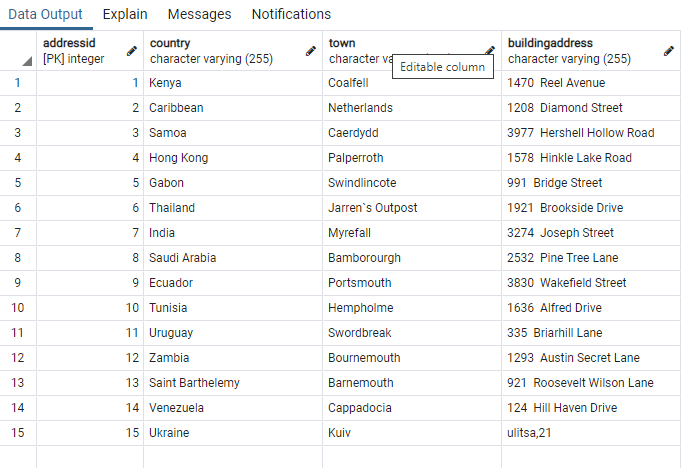
1. **INSERT** **INTO** account(login, **password**) **VALUES** ('babble','yvk4WVGdRHHsn6FJ');
2. **INSERT** **INTO** account(login, **password**) **VALUES** ('jet','R5ScYFwcxTLDEaUS');
3. **INSERT** **INTO** account(login, **password**) **VALUES** ('burberry','dCWxpRXWYaJ2LyqU');
4. **INSERT** **INTO** account(login, **password**) **VALUES** ('whole','tyuECEPxb4Ss3LFA');
5. **INSERT** **INTO** account(login, **password**) **VALUES** ('genoa','eL5mrwm5Me8aN2rg');
6. **INSERT** **INTO** account(login, **password**) **VALUES** ('camel','ydHK5fQZGuZfwrBY');
7. **INSERT** **INTO** account(login, **password**) **VALUES** ('waist','f6NsJErteHs9N5Jc');
8. **INSERT** **INTO** account(login, **password**) **VALUES** ('supreme','RKU7WQpwcnvbHWzQ');
9. **INSERT** **INTO** account(login, **password**) **VALUES** ('committee','PhE3c6aJvGmkY6rr');
10. **INSERT** **INTO** account(login, **password**) **VALUES** ('rope','HXzUUxVZaM3DmPZA');
11. **INSERT** **INTO** account(login, **password**) **VALUES** ('mellohi','mxKNKt8ySxhUeZ67');
12. **INSERT** **INTO** account(login, **password**) **VALUES** ('operate','Yzk42CUsHbPWEang');
13. **INSERT** **INTO** account(login, **password**) **VALUES** ('rooden','mKPR9JmyHY62uv9p');
15. **select** \* **from** account



*Рисунок 3.1 Заповнена таблиця «Account»*

3.2.2 Заповнення таблиці «address»

1. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Kenya','Coalfell','1470  Reel Avenue');
2. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Caribbean','Netherlands','1208  Diamond Street');
3. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Samoa','Caerdydd','3977  Hershell Hollow Road');
4. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Hong Kong','Palperroth','1578  Hinkle Lake Road');
5. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Gabon','Swindlincote','991  Bridge Street');
6. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Thailand','Jarren`s Outpost','1921  Brookside Drive');
7. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('India','Myrefall','3274  Joseph Street');
8. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Saudi Arabia','Bamborourgh','2532  Pine Tree Lane');
9. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Ecuador','Portsmouth','3830  Wakefield Street');
10. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Tunisia','Hempholme','1636  Alfred Drive');
11. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Uruguay','Swordbreak','335  Briarhill Lane');
12. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Zambia','Bournemouth','1293  Austin Secret Lane');
13. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Saint Barthelemy','Barnemouth','921  Roosevelt Wilson Lane');
14. **INSERT** **INTO** address(country, town, buildingaddress) **VALUES** ('Venezuela','Cappadocia','124  Hill Haven Drive');
16. **select** \* **from** address

 *Рисунок 3.2 Заповнена таблиця «address»*

3.2.3 Завнення таблиці «bank»

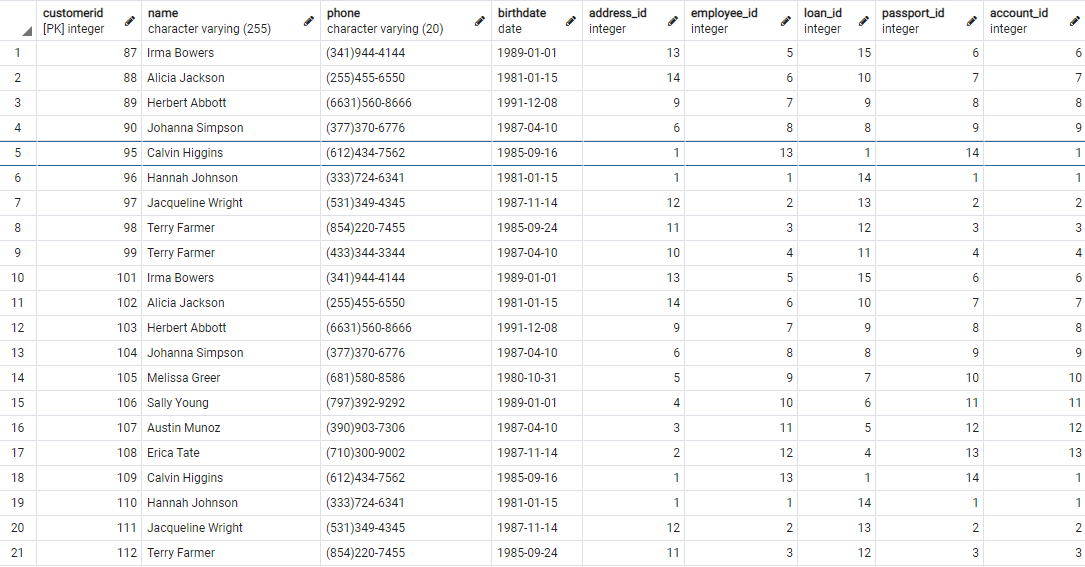
1. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5553 7573 6652 6312','(401) 521-4234','Mildred Reynolds',(**SELECT** addressid **from** address **WHERE** addressid='1'));
2. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4541 4205 8445 8820','(703) 806-5375','Amy Walters',(**SELECT** addressid **from** address **WHERE** addressid='2'));
3. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5864 4282 6750 5573','(623) 896-9257','Joseph Ball',(**SELECT** addressid **from** address **WHERE** addressid='3'));
4. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4133 1578 8320 0263','(813) 987-0950','Charlie Norman',(**SELECT** addressid **from** address **WHERE** addressid='4'));
5. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4343 1804 2237 1284','(570) 420-3055','Melissa Weber',(**SELECT** addressid **from** address **WHERE** addressid='5'));
6. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4152 6377 7244 2270','(336) 815-5622','Annie Reid',(**SELECT** addressid **from** address **WHERE** addressid='6'));
7. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5286 5615 2026 5815','(751) 544-7131','Antonia Jenkins',(**SELECT** addressid **from** address **WHERE** addressid='7'));
8. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5266 6131 7255 0178','(571) 588-0096','Ann Gordon',(**SELECT** addressid **from** address **WHERE** addressid='8'));
9. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4216 8758 4714 0735','(972) 493-1613','Israel Stanley',(**SELECT** addressid **from** address **WHERE** addressid='9'));
10. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4086 7525 1758 0481','(255) 706-9980','Ricky Lopez',(**SELECT** addressid **from** address **WHERE** addressid='10'));
11. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4786 3608 5567 3000','(999) 564-9528','Dawn Pope',(**SELECT** addressid **from** address **WHERE** addressid='11'));
12. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5621 7631 3572 6645','(224) 712-4239','Shane Taylor',(**SELECT** addressid **from** address **WHERE** addressid='12'));
13. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('4210 6742 6151 2081','(344) 703-4279','Franklin Mccarthy',(**SELECT** addressid **from** address **WHERE** addressid='13'));
14. **INSERT** **INTO** bank(bin, phone, **name**, address\_id) **VALUES** ('5718 7148 8333 6065','(234) 453-6549','Carroll Craig',(**SELECT** addressid **from** address **WHERE** addressid='14'));
16. **select** \* **from** bank



*Рисунок 3.3 Заповнена таблиця «bank»*

3.2.4 Заповнення таблиці «customer»

1. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Hannah Johnson','(333)724-6341','1981-01-15',(**SELECT** addressid **from** address **WHERE** addressid='1'),(**SELECT** employeeid **from** employee **WHERE** employeeid='1'),(**SELECT** loanid **from** loan **WHERE** loanid='14'),(**SELECT** passportid **from** passport **WHERE** passportid='1'),(**SELECT** accountid **from** account **WHERE** accountid='1'));
2. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Jacqueline Wright','(531)349-4345','1987-11-14',(**SELECT** addressid **from** address **WHERE** addressid='12'),(**SELECT** employeeid **from** employee **WHERE** employeeid='2'),(**SELECT** loanid **from** loan **WHERE** loanid='13'),(**SELECT** passportid **from** passport **WHERE** passportid='2'),(**SELECT** accountid **from** account **WHERE** accountid='2'));
3. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Terry Farmer','(854)220-7455','1985-09-24',(**SELECT** addressid **from** address **WHERE** addressid='11'),(**SELECT** employeeid **from** employee **WHERE** employeeid='3'),(**SELECT** loanid **from** loan **WHERE** loanid='12'),(**SELECT** passportid **from** passport **WHERE** passportid='3'),(**SELECT** accountid **from** account **WHERE** accountid='3'));
4. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Terry Farmer','(433)344-3344','1987-04-10',(**SELECT** addressid **from** address **WHERE** addressid='10'),(**SELECT** employeeid **from** employee **WHERE** employeeid='4'),(**SELECT** loanid **from** loan **WHERE** loanid='11'),(**SELECT** passportid **from** passport **WHERE** passportid='4'),(**SELECT** accountid **from** account **WHERE** accountid='4'));
5. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Fedir Molodiy','(249)934-3653','1997-04-04',(**SELECT** addressid **from** address **WHERE** addressid='8'),(**SELECT** employeeid **from** employee **WHERE** employeeid='14'),(**SELECT** loanid **from** loan **WHERE** loanid='1'),(**SELECT** passportid **from** passport **WHERE** passportid='5'),(**SELECT** accountid **from** account **WHERE** accountid='5'));
6. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Irma Bowers','(341)944-4144','1989-01-01',(**SELECT** addressid **from** address **WHERE** addressid='13'),(**SELECT** employeeid **from** employee **WHERE** employeeid='5'),(**SELECT** loanid **from** loan **WHERE** loanid='15'),(**SELECT** passportid **from** passport **WHERE** passportid='6'),(**SELECT** accountid **from** account **WHERE** accountid='6'));
7. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Alicia Jackson','(255)455-6550','1981-01-15',(**SELECT** addressid **from** address **WHERE** addressid='14'),(**SELECT** employeeid **from** employee **WHERE** employeeid='6'),(**SELECT** loanid **from** loan **WHERE** loanid='10'),(**SELECT** passportid **from** passport **WHERE** passportid='7'),(**SELECT** accountid **from** account **WHERE** accountid='7'));
8. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Herbert Abbott','(6631)560-8666','1991-12-08',(**SELECT** addressid **from** address **WHERE** addressid='9'),(**SELECT** employeeid **from** employee **WHERE** employeeid='7'),(**SELECT** loanid **from** loan **WHERE** loanid='9'),(**SELECT** passportid **from** passport **WHERE** passportid='8'),(**SELECT** accountid **from** account **WHERE** accountid='8'));
9. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Johanna Simpson','(377)370-6776','1987-04-10',(**SELECT** addressid **from** address **WHERE** addressid='6'),(**SELECT** employeeid **from** employee **WHERE** employeeid='8'),(**SELECT** loanid **from** loan **WHERE** loanid='8'),(**SELECT** passportid **from** passport **WHERE** passportid='9'),(**SELECT** accountid **from** account **WHERE** accountid='9'));
10. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Melissa Greer','(681)580-8586','1980-10-31',(**SELECT** addressid **from** address **WHERE** addressid='5'),(**SELECT** employeeid **from** employee **WHERE** employeeid='9'),(**SELECT** loanid **from** loan **WHERE** loanid='7'),(**SELECT** passportid **from** passport **WHERE** passportid='10'),(**SELECT** accountid **from** account **WHERE** accountid='10'));
11. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Sally Young','(797)392-9292','1989-01-01',(**SELECT** addressid **from** address **WHERE** addressid='4'),(**SELECT** employeeid **from** employee **WHERE** employeeid='10'),(**SELECT** loanid **from** loan **WHERE** loanid='6'),(**SELECT** passportid **from** passport **WHERE** passportid='11'),(**SELECT** accountid **from** account **WHERE** accountid='11'));
12. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Austin Munoz','(390)903-7306','1987-04-10',(**SELECT** addressid **from** address **WHERE** addressid='3'),(**SELECT** employeeid **from** employee **WHERE** employeeid='11'),(**SELECT** loanid **from** loan **WHERE** loanid='5'),(**SELECT** passportid **from** passport **WHERE** passportid='12'),(**SELECT** accountid **from** account **WHERE** accountid='12'));
13. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Erica Tate','(710)300-9002','1987-11-14',(**SELECT** addressid **from** address **WHERE** addressid='2'),(**SELECT** employeeid **from** employee **WHERE** employeeid='12'),(**SELECT** loanid **from** loan **WHERE** loanid='4'),(**SELECT** passportid **from** passport **WHERE** passportid='13'),(**SELECT** accountid **from** account **WHERE** accountid='13'));
14. **INSERT** **INTO** customer(**name**, phone, birthdate, address\_id, employee\_id, loan\_id, passport\_id, account\_id) **VALUES** ('Calvin Higgins','(612)434-7562','1985-09-16',(**SELECT** addressid **from** address **WHERE** addressid='1'),(**SELECT** employeeid **from** employee **WHERE** employeeid='13'),(**SELECT** loanid **from** loan **WHERE** loanid='3'),(**SELECT** passportid **from** passport **WHERE** passportid='14'),(**SELECT** accountid **from** account **WHERE** accountid='14'));



*Рисунок 3.4 Заповнена таблиця «customer»*

3.2.5 Заповнення таблиці «employee»

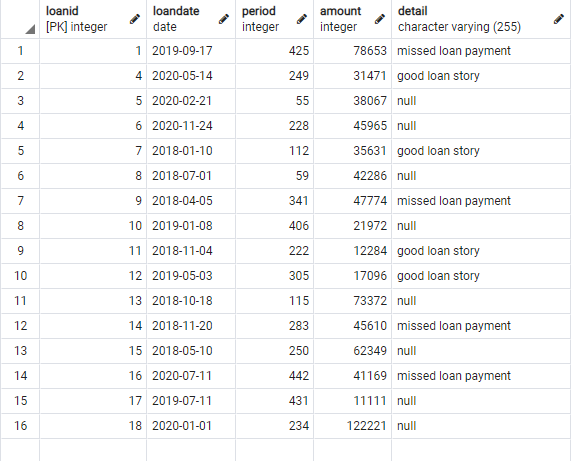
1. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (1,'Hannah Johnson','2003-06-04','1991-12-08','(303)729-6031',(**SELECT** bankid **from** bank **WHERE** bankid='1'));
2. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (2,'Jacqueline Wright','2005-01-31','1989-01-01','(202)475-6080',(**SELECT** bankid **from** bank **WHERE** bankid='2'));
3. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (2,'Terry Farmer','2005-03-31','1996-08-09','(848)230-7335',(**SELECT** bankid **from** bank **WHERE** bankid='3'));
4. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (3,'Irma Bowers','2006-12-07','1985-09-24','(310)901-7376',(**SELECT** bankid **from** bank **WHERE** bankid='4'));
5. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (4,'Alicia Jackson','2007-01-29','1980-10-31','(317)300-6416',(**SELECT** bankid **from** bank **WHERE** bankid='5'));
6. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (2,'Herbert Abbott','2007-10-30','1986-10-15','(601)475-7622',(**SELECT** bankid **from** bank **WHERE** bankid='6'));
7. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (3,'Johanna Simpson','2010-07-14','1981-01-15','(475)675-2063',(**SELECT** bankid **from** bank **WHERE** bankid='7'));
8. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (4,'Melissa Greer','2011-03-04','1987-04-10','(808)886-5986',(**SELECT** bankid **from** bank **WHERE** bankid='8'));
9. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (5,'Sally Young','2011-10-06','1988-12-15','(561)219-4181',(**SELECT** bankid **from** bank **WHERE** bankid='9'));
10. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (6,'Austin Munoz','2013-10-21','1987-11-14','(289)995-3711',(**SELECT** bankid **from** bank **WHERE** bankid='10'));
11. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (2,'Erica Tate','2014-07-09','1999-07-14','(425)332-3320',(**SELECT** bankid **from** bank **WHERE** bankid='11'));
12. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (3,'Calvin Higgins','2018-08-21','1985-09-16','(331)999-4150',(**SELECT** bankid **from** bank **WHERE** bankid='12'));
13. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (4, 'Fedir Molodiy','2019-01-01','1997-12-30','(631)500-8556',(**SELECT** bankid **from** bank **WHERE** bankid='13'));
14. **INSERT** **INTO** employee(position, **name**, hiredate, birthdate, phone, bank\_id) **VALUES** (5,'Lorenzo Manning','2019-04-22','1997-04-04','(717)312-9232',(**SELECT** bankid **from** bank **WHERE** bankid='14'));
16. **select** \* **from** employee



*Рисунок 3.5 Заповнена таблиця «employee»*

3.2.6 Заповнення таблиці «loan»

1. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2019-09-17',425,78653,'missed loan payment');
2. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2020-05-14',249,31471,'good loan story');
3. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2020-02-21',55,38067,'null');
4. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2020-11-24',228,45965,'null');
5. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-01-10',112,35631,'good loan story');
6. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-07-01',59,42286,'null');
7. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-04-05',341,47774,'missed loan payment');
8. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2019-01-08',406,21972,'null');
9. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-11-04',222,12284,'good loan story');
10. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2019-05-03',305,17096,'good loan story');
11. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-10-18',115,73372,'null');
12. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-11-20',283,45610,'missed loan payment');
13. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2018-05-10',250,62349,'null');
14. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2020-07-11',442,41169,'missed loan payment');
15. **INSERT** **INTO** loan(loandate, period, amount, detail) **VALUES** ('2019-07-11',431,11111,'null');
17. **select** \* **from** loan



*Рисунок 3.6 Заповнена таблиця «loan»*

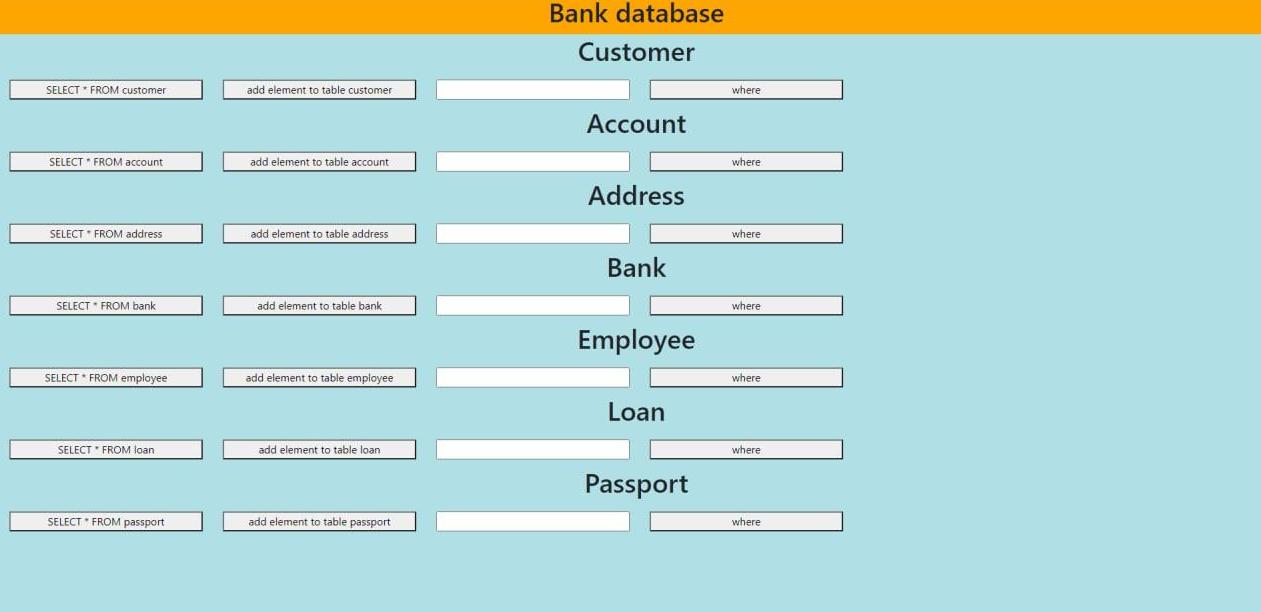
3.2.7 Заповнення таблиці «passport»

1. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Hannah Johnson', **true**,0,'Eco Focus');
2. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Jacqueline Wright', **true**,0,'Innovation Arch');
3. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Terry Farmer', **true**,0,'Strat Security');
4. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Irma Bowers', **true**,0,'Inspire Fitness Co');
5. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Fedir Molodiy',**false**,1,'Candor Corp');
6. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Alicia Jackson', **true**,1,'Cogent Data');
7. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Herbert Abbott', **true**,0,'Epic Adventure Inc');
8. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Johanna Simpson',**false**,1,'Sanguine Skincare');
9. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Melissa Greer',**false**,2,'Vortex Solar');
10. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Sally Young',**false**,2,'Admire Arts');
11. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Austin Munoz',**false**,0,'Bravura Inc');
12. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Erica Tate',**false**,0,'Bonefete Fun');
13. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Calvin Higgins',**false**,0,'Moxie Marketing');
14. **INSERT** **INTO** passport(**name**, ismarried, numofchildren, workingplace) **VALUES** ('Lorenzo Manning',**false**,0,'Zeal Wheels');
16. **select** \* **from** passport



*Рисунок 3.7 Заповнена таблиця «passport»*

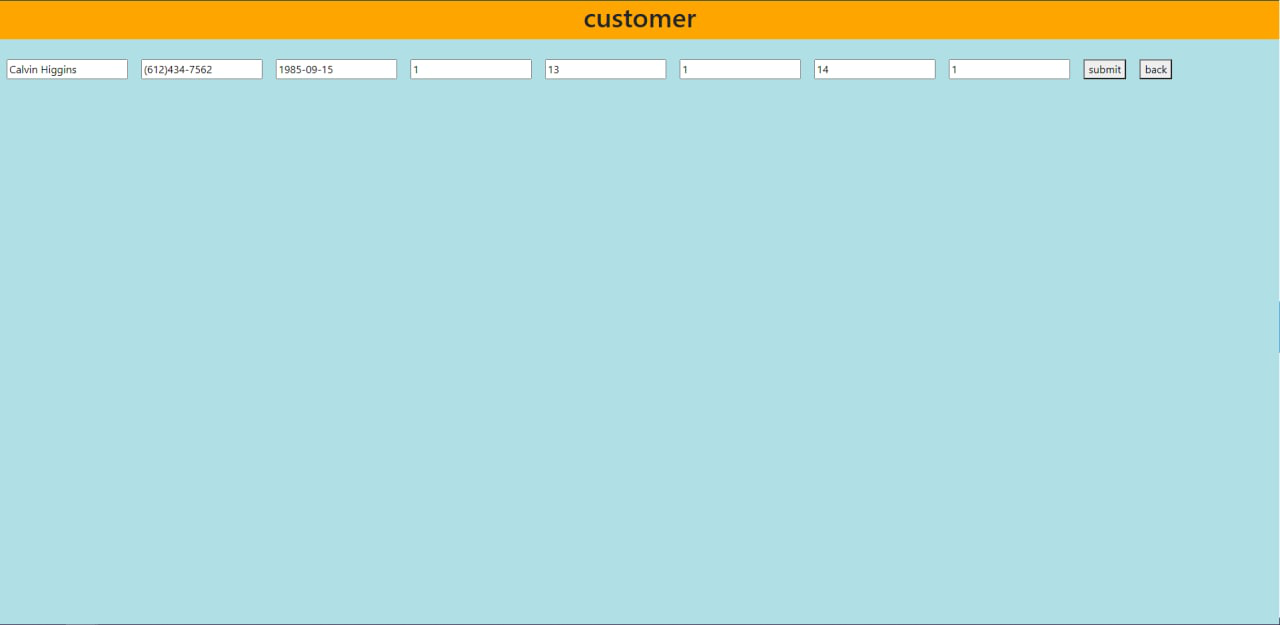
**4. Реалізація застосунку**



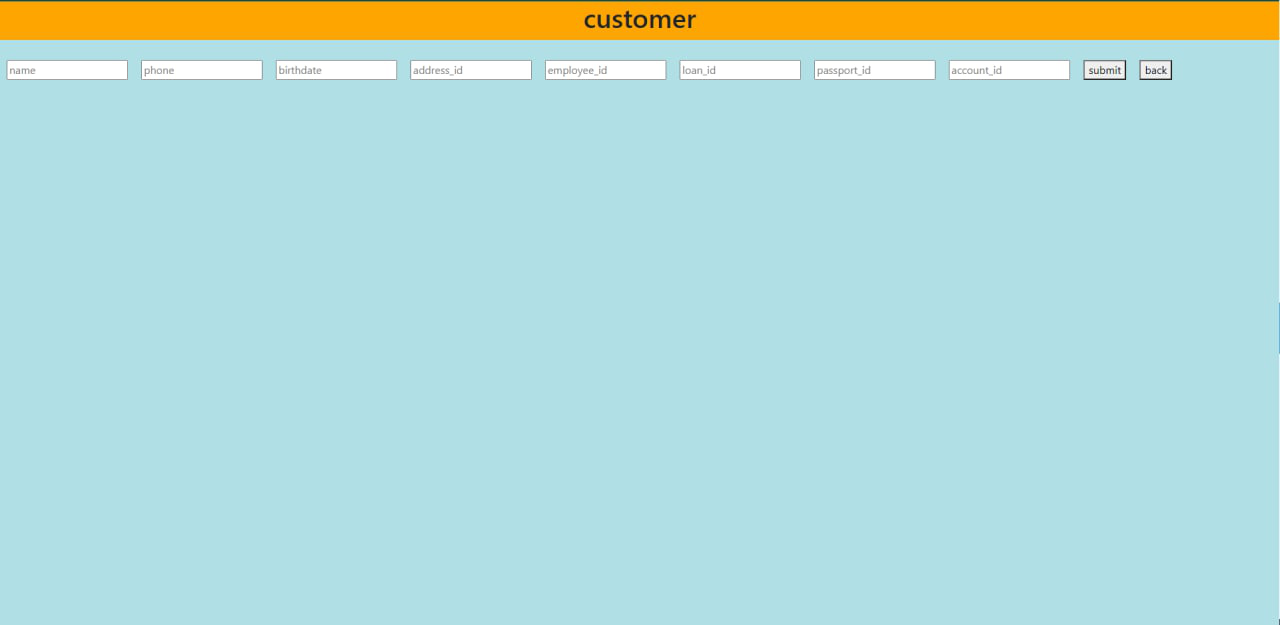
*Рисунок 4.1 Виведення головного меню в інтерфейсі користувача*



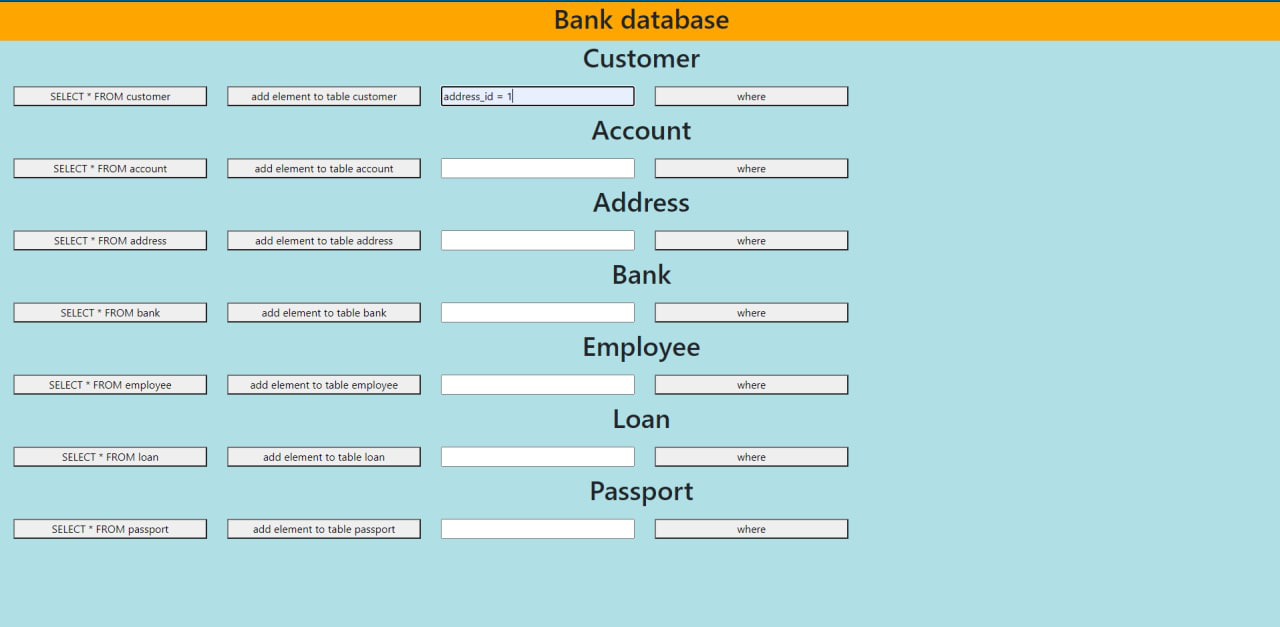
*Рисунок 4.2 Виведення однієї із таблиць*



*Рисунок 4.3 Редагування одного з рядків таблиці*



*Рисунок 4.4 Додавання до таблиці*



*Рисунок 4.5 Введення параметрів для фільтрації даних з таблиці*



*Рисунок 4.6 Виведення фільтрованої таблиці*

Також присутня можливість перегортування сторінок та перехід на задану. Видалення елементів відбувається при натисканні клавішi «delete»

# **5.Створення об’єктів бази даних**

## 5.1. Створення представлення

Створене для виведення тільки кредитів, які мають деталі

1. --представлення
2. **CREATE** **VIEW** loanWithDetails **AS**
3. **SELECT** loanid, period, amount, customerid, **name**, phone, employee\_id
4. **FROM** loan l
5. RIGHT JOIN customer cu **ON** cu.loan\_id = l.loanId AND  l.detail!='null'

## 5.2 Створення табличної функції

Дана функція створена для виведення клієнтів, які народилися в проміжку між заданими датами

1. --Таблична функція
2. **CREATE** OR REPLACE **FUNCTION** getCustomer(fromDate **DATE** **DEFAULT** NOW(), toDate **DATE** **DEFAULT** NOW())
3. **RETURNS**
4. **TABLE**("customerid" **int**, "birthdate" **DATE**)
5. **AS**
6. $BODY$
7. **BEGIN**
8. IF fromDate **IS** NOT NULL AND toDate **IS** NOT NULL **THEN**
9. **RETURN** QUERY
10. **SELECT** cu."customerid", cu."birthdate" **FROM** "customer" cu
11. **WHERE** cu."birthdate" >= fromDate AND cu."birthdate" <= toDate;
12. **END** IF;
13. **RETURN**;
14. **END**
15. $BODY$
16. LANGUAGE plpgsql;
18. **SELECT** \* **FROM** getCustomer('1900-01-01', '2020-12-13');

## 5.3 Створення скалярної функції

Дана функція потрібна для виведення кредитів, які мають термін коротший за заданий

1. --Скалярна функція
2. **CREATE** OR REPLACE **FUNCTION** getLoanLesThanPeriod(per **int**)
3. **RETURNS**
4. SETOF "loan"
5. **AS**
6. $BODY$
7. **RETURN** QUERY
8. **SELECT** \* **FROM** "loan"
9. **WHERE** "period" < $1;
10. **END**
11. $BODY$
12. LANGUAGE plpgsql;
14. **SELECT** \* **FROM** getLoanLesThanPeriod(5000);

## 5.4 Створення збереженої процедури

Використовується для зміни даних в таблиці «Account»

1. **CREATE** OR REPLACE **PROCEDURE** change(
2. reqid **int**,
3. log **VARCHAR**(255),
4. pass **VARCHAR**(255))
5. LANGUAGE plpgsql
6. **AS** $$
7. **BEGIN**
8. **UPDATE** account **SET** account.**password** = pass **where** account.accountid = reqid
9. **UPDATE** account **SET** account.login = log **where** account.accountid = reqid
11. **commit**;
12. **end**;$$
14. call change(1,'mynewlogin','mynewpassword')

#### 5.5 Створення трігеру

Даний трігер потрібен для зберігання даних в таблиці «LoanArchive» після видалення з «Loan»

1. ---Trigger
2. **CREATE** **TABLE** IF NOT EXISTS "LoanArchive"
3. (
4. "DeletedBy" TEXT,
5. "DeletionDateTime" **TIMESTAMP**,
6. LIKE loan
7. );
9. **CREATE** OR REPLACE **FUNCTION** dell()
10. **RETURNS** **TRIGGER**
11. **AS** $$
12. **BEGIN**
13. **INSERT** **INTO** "LoanArchive" (
14. "loanID",
15. "loandate",
16. "period",
17. "amount",
18. "detail",
19. "DeletionDateTime",
20. "DeletedBy"
21. ) **SELECT** OLD.\*, statement\_timestamp(), session\_user;
22. **RETURN** NEW;
23. **END**;
24. $$
25. LANGUAGE plpgsql;
27. **CREATE** **TRIGGER** dell
28. BEFORE **DELETE** **ON** loan
29. **FOR** EACH ROW
30. **EXECUTE** **PROCEDURE** dell();

## 5.6 Використання «UNION»

Створене для виведення повної інформації про кожного користувача

1. --use union
2. **select** \* **from** customer cus
3. **union** **select** \* account acc **where** acc.accountid = cus.account\_id
4. **union** **select** \* employee em **where** acc.accountid = cus.employee\_id

## 5.7 Використання «CASE»

Створене для аналізу таблиці «passport» та виведення інформації про максимальну суму позики даному клієнту

1. **select** \*, case
2. **when** p.numofchildren = 0 and p.ismarried = **false** **then** 'agree'
3. **when** p.numofchildren = 1 and p.ismarried = **false** **then** 'no more, than 5000$'
4. **when** p.numofchildren = 2 and p.ismarried = **false** **then** 'no more, than 500$'
5. **else** 'disagree'
6. **end**
7. **from** passport p

# **6. Створення нового користувача**

В даному етапі розробки я повинен створити користувача, який використовуватиме дану базу даних, та не матиме прав на його редагування.

1. --Користувач
2. **CREATE** ROLE userofBankDatabase;
3. **GRANT** **SELECT**, **TRIGGER**, **CONNECT**, **EXECUTE** **ON**
4. account,address,bank,customer,employee,loan,passport **TO** userofBankDatabase;

Я дозволив корисувачу виконувати команди «SELECT», «TRIGGER», «CONNECT», «EXECUTE». На мою думку цього досить бля роботи з бд не видаляючи елеменів

# **ВИСНОВКИ**

Актуальність курсової роботи по базах даних зумовлена ​​сучасними тенденціями розвитку інформаційних технологій, що вимагають безперервне підвищення продуктивності.

Завдяки даній роботі я ознайомився з структурою БД та створив базу даних для оптимізації роботи працівників банку.

На першому етапі створення БД «Облік клієнтів банку» було здійснено аналіз структури БД, також я ознайомився з можливостями PostgreSQL.

Наступним етапом роботи було створення ER діаграми та її нормалізація. Також розроблена модель фізичної БД, проаналізував зв’язки між таблицями.

На третьому етапі розробки я створив БД та провів тестування шляхом її заповнення.

Завершальним етапом розробки БД «Облік клієнтів банку» було створення зручного інтерфейсу користувача для роботи з БД.

В результаті виконання курсової роботи по БД слід провести її тестування шляхом заповнення, після чого створити запити і оцінити ефективність роботи бази даних.

В подальшому я маю намір покращити дану програму розробленими мною функціями, та інтегрувати їх в інтерфейс користувача для більшої зручності.

Недоліком даної роботи є ручне введення даних для фільтрації. У наступних версіях намагатимуся вирішити вказану проблему.

### Список літератури:

* 1. Ульман Дж. Основы систем баз данных. Пер.с англ. – М.: Финансы и статистика, 1983. – 334 стр.
  2. Хаммер М. Реинжиниринг корпорации : Манифест революции в бизнесе / Майкл Хаммер, Джеймс Чампи; перс англ. – М.: Манн, Иванов и Фербер, 2006. – 304 с. Третья нормальная форма. Материал из Википедии — свободной энциклопедии [Електронний ресурс] // Режим доступу: http://ru.wikipedia.org/wiki/Третья\_нормальная\_форма#
  3. Проектирование и реализация баз данных Microsoft SQL Server 2000. Учебный курс Microsoft / Пер.с англ. – 3-е изд. – М.: Издательско-торговый дом «Русская редакция»; СПб.: Питер, 2006. – 512 стр.: ил.
  4. Морган С. Проектирование и оптимизация доступа к базам данных Microsoft SQL Server 2005. Учебный курс Microsoft : Пер.с англ. / С.Морган, Т.Тернстрем. – М.: Издательство «Русская редакция», 2008. – 480 стр.: ил.
  5. Интегрированное управление производством: Организационные и технологические аспекты менеджмента предприятиями / В.И.Архангельский, И.М.Богаенко, Г.Г.Грабовский, М.О.Рюмшин; под ред. В.И.Архангельского. – К.: Техніка, 2005. – 328с.
  6. Фізична модель даних [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Фізична\_модель\_даних#Фізична\_схема
  7. Модель «сутність — зв'язок» [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Модель\_«сутність\_—\_зв’язок»