**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: “Система аналізу цін споживчих товарів”

|  |  |  |
| --- | --- | --- |
| **Студент**  **групи** КП-03 | **Сіренко Вікторія Юріївна** | \_\_\_\_\_\_\_\_\_  (підпис) |
| **Викладач**  **к.т.н, доцент кафедри СПіСКС** | **Петрашенко А.В.** | \_\_\_\_\_\_\_\_\_  (підпис) |

Захищено з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2021**Анотація**

Метою виконання даної курсової роботиє набуття практичних навичок розробкисучасного програмного забезпечення, що взаємодіє з реляційними базами даних, а також здобуття навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації.

Під час виконання курсового проекту ми навчилися розробляти програмне забезпечення для реляційних баз даних, оволоділи основами використання СУБД, а також інструментальними засобами підтримки розробки додатків для подібних баз даних.

Темою даної курсової є система аналізу цін споживчих товарів.

Результатом даного проекту став додаток, який призначений для фільтрації , зберігання та аналізування поточних цін на товари за певною категорією, прогнозування потреби покупців та визначення певних цінових трендів.

**Зміст**

Анотація. 2

Зміст. 3

[Вступ. 4](#_Toc73702061)

[1. Аналіз інструментарію для виконання курсової роботи. 5](#_Toc73702062)

[2. Структура бази даних. 7](#_Toc73702064)

[3. Опис програмного забезпечення. 8](#_Toc73702071)

[3.1. Загальна структура програмного забезпечення. 8](#_Toc73702072)

[3.2. Опис модулів програмного забезпечення. 8](#_Toc73702073)

[3.3. Опис основних алгоритмів роботи. 9](#_Toc73702074)

4. Аналіз функціонування засобів реплікації. 11

[5. Аналіз функціонування засобів резервування/відновлення. 1](#_Toc73702080)2

[6. Аналіз результатів підвищення швидкодії запитів. 1](#_Toc73702081)2

[7. Опис результатів аналізу предметної галузі. 1](#_Toc73702081)5

[Висновки. 1](#_Toc73702082)5

[Література. 16](#_Toc73702081)

[Додатки. 1](#_Toc73702081)7

**Вступ**

В наш час, ціна грає важливу роль в покупці товарів і розвитку магазину. Адже, згідно статистичних даних:

* Більше ніж 80% покупців до того, як зробити замовлення, вивчають і порівнюють ціни у різних продавців.
* 60% покупців віддають перевагу тим магазинам, у яких “хороші” ціни. Вочевидь випливає висновок: для покупця, що вирішує, де зробити покупку, ціна – це основний критерій вибору.

Часто, дані про покупки та замовлення жодним чином не аналізуються, що призводить до втрати прибутків та неефективного ведення торгівлі.

Для вирішення цієї проблеми, було розроблено додаток, який призначений для фільтрації, зберігання та аналізування поточних цін на товари за певною категорією, прогнозування потреби покупців та визначення певних цінових трендів.

Метою розробки даного курсового проекту є набуття навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних, володіння основами використання СУБД, засобами аналізу великих обсягів даних, генерації, індексації таблиць бази даних, роботи з ними, зберігання та реплікації даних засобами СУБД.

**Аналіз інструментарію для виконання курсової роботи**

1. Обґрунтування вибору мови програмування:

Для реалізації даного програмного забезпечення була використана мова програмування - C#, оскільки ця мова дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків. C# підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію.

1. Обґрунтування вибору бібліотек та фреймворку:

* Фреймворк .NET - найпоширеніша версія середовища .NET. Він служить середовищем для підтримки, розробки і виконання розподілених додатків, що базуються на компонентах (елементах управління). Була використана бібліотека базових класів, яка містить в собі визначення різноманітних примітивів, якими можуть бути: потоки, графічні API-інтерфейси, реалізація баз даних, файловий ввід-вивід та інше. Було використане автоматичне управління пам’яттю, що є однією з найбільших переваг фреймворку .NET . А також, великою перевагою технології .NET є можливість запускати код на різних операційних системах та апаратних пристроях.
* ScottPlot - це бібліотека для побудови графіків із відкритим кодом для .NET, яка полегшує інтерактивне відображення великих наборів даних.
* ML.NET — це безкоштовна бібліотека машинного навчання програмного забезпечення для мов програмування C# і F# . ML.NET надає існуючим розробникам .NET аналітичні та передбачені можливості машинного навчання на основі моделі.
* Для роботи з MySQL нам знадобився «MySQL Connector / NET».

MySQL Connector / Net дозволяє розробникам створювати .NET-додатки, які з'єднуються з MySQL. MySQL Connector / Net реалізує повнофункціональний інтерфейс ADO.NET.

1. Обґрунтування вибору СУБД:

В даному проекті була використана реляційна СУБД - MySQL.

MySQL надає багатий набір функціональних можливостей, які підтримують безпечне середовище для зберігання, обслуговування і отримання даних.

Переваги MySQL:

1. Простота у використанні. MySQL досить легко інсталюється, а наявність безлічі плагінів і допоміжних додатків спрощує роботу з базами даних.
2. Великий функціонал. Система MySQL має практично всім необхідним інструментарієм, який може знадобитися в реалізації практично будь-якого проекту.
3. Безпека. Система спочатку створена таким чином, що безліч вбудованих функцій безпеки в ній працюють за замовчуванням.
4. Масштабованість. Будучи досить універсальної СУБД, MySQL в рівній мірі легко може бути використана для роботи і з малими, і з великими об'ємами даних.
5. Швидкість. Висока продуктивність системи забезпечується за рахунок спрощення деяких використовуваних в ній стандартів.
6. Обґрунтування вибору середовища розробки:

Розробка проводилась на операційній системі Windows 10, проте проект підтримує і інші платформи, зокрема Лінукс.

В якості IDE було обрано Visual Studio Code, яке надає можливості підключення та роботи з СУБД MySql.

**Структура бази даних**

База даних проекту складається з 4 таблиць:

І. customers – таблиця з даними споживачів:

* id (PK) – ідентифікатор споживача
* username – користувацьке ім’я споживача
* address –адреса
* phone\_number –номер телефону

ІІ. orders– таблиця, яка містить інформацію про замовлення:

* id (PK)- ідентифікатор замовлення
* order\_date –дата здійснення замовлення
* order\_price –ціна замовлення
* customer\_id (FK) – ідентифікатор покупця

ІІІ. products – таблиця з даними про продукти:

* id (PK) – ідентифікатор продукту
* product\_name – назва продукту
* price -ціна товару

ІV. purchases – допоміжна таблиця зв’язків:

* id (PK) – ідентифікатор таблиці
* product\_id (FK) – ідентифікатор продукту
* order\_id (FK) – ідентифікатор замовлення

Усі таблиці є нормалізованими до третьої нормальної форми.

Зв’язки між таблицями:

* Споживач-замовлення **one2many**
* Замовлення-продукт **many2many**

**Опис програмного забезпечення**

**Загальна структура програмного забезпечення.**

Програмні засоби містять наступні компоненти:

1. Підсистема попередньої обробки даних, що складається з:

1.1. Засобів генерації даних: присутня рандомізація даних, створено CRUD операції, що дозволяють взаємодіяти з даними таблиць.

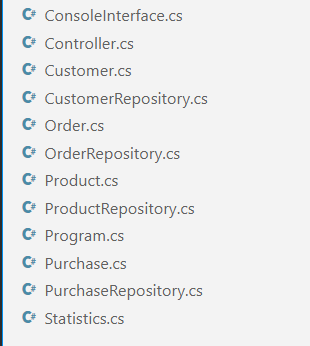
1.2. Засобів фільтрації та валідації даних.

2. База даних системи призначена для зберігання і аналізу інформації розробленої моніторингової системи.

3. Засоби аналізу даних.

**Опис модулів програмного забезпечення**

Програма містить такі модулі :



ConsoleInterface – дозволяє користувачу обрати необхідну команду.

Controller – реалізує логіку додатку.

Customer, Order, Product, Purchase – окремі модулі для кожного класу, який відповідає певній сутності бази даних: customers, orders, products, purchases.

CustomerRepository, OrderRepository, ProductRepository, PurchaseRepository – ці модулі взаємодіють з базою даних та дозволяють програмі отримувати необхідну інформацію.

Statistics – модуль, який виконує статичний аналіз даних та на основі отриманої інформації створює графіки.

**Опис основних алгоритмів роботи**

Для взаємодії з даними для кожної таблиці було реалізовано 4 основні функції управління даними  «створення, читання, оновлення і вилучення» (CRUD).

Наведемо приклад CRUD операцій для таблиці споживачів:

Створення:

|  |
| --- |
| 1. *//adds a new customer to the database* 2. **public** **ulong** Insert(Customer customer) 3. { 4. connection.Open(); 6. MySqlCommand command = connection.CreateCommand(); 7. command.CommandText = 8. @" 9. INSERT INTO customers (username, address, phone\_number) 10. VALUES (@username, @address, @phone\_number); 11. SELECT LAST\_INSERT\_ID();"; 12. command.Parameters.AddWithValue("@username", customer.username); 13. command.Parameters.AddWithValue("@address", customer.address); 14. command.Parameters.AddWithValue("@phone\_number", customer.phone\_number); 15. **ulong** newId = (**ulong**)command.ExecuteScalar(); 16. connection.Close(); 17. **return** newId; 18. } |

Читання:

|  |
| --- |
| 1. *//finds the customer in the database by his username  and returns it* 2. **public** Customer GetCustomer(**string** username) 3. { 4. connection.Open(); 5. MySqlCommand command = connection.CreateCommand(); 6. command.CommandText = @"SELECT \* FROM customer WHERE username = @username "; 7. command.Parameters.AddWithValue("@username", username); 8. MySqlDataReader reader = command.ExecuteReader(); 9. **if** (reader.Read()) 10. { 11. Customer customer = ParseCustomer(reader); 12. connection.Close(); 13. **return** customer; 14. } 15. reader.Close(); 16. connection.Close(); 17. **return** **null**; 19. } |

Оновлення:

|  |
| --- |
| 1. *//updates the customer* 2. **public** **bool** Update(Customer customer, **long** customerId) 3. { 4. connection.Open(); 5. MySqlCommand command = connection.CreateCommand(); 6. command.CommandText = @"UPDATE customers SET username=@username , address=@address , phone\_number=@phone\_number WHERE id=@id"; 7. command.Parameters.AddWithValue("@id", customerId); 8. command.Parameters.AddWithValue("@username", customer.username); 9. command.Parameters.AddWithValue("@address", customer.address); 10. command.Parameters.AddWithValue("@phone\_number", customer.phone\_number); 11. **int** nChanged = command.ExecuteNonQuery(); 12. connection.Close(); 13. **return** nChanged == 1; 15. } |

Видалення:

|  |
| --- |
| 1. *//deletes the customer by his ID* 2. **public** **bool** Delete(**long** id) 3. { 4. connection.Open(); 5. MySqlCommand command = connection.CreateCommand(); 6. command.CommandText = @"DELETE FROM customers WHERE id=@id"; 7. command.Parameters.AddWithValue("@id", id); 8. **int** nChanges = command.ExecuteNonQuery(); 9. connection.Close(); 10. **return** nChanges == 1; 11. } |

Наявна валідація даних, які вводить користувач, а також присутня фільтрація даних.

Наведемо приклад фільтрації даних: функція фільтрації даних покупця за його користувацьким ім’ям:

|  |
| --- |
| 1. **public** List<Customer> FindCustomer(**string** user\_name) 2. { 3. connection.Open(); 4. MySqlCommand command = connection.CreateCommand(); 5. command.CommandText = @"SELECT \* FROM customers WHERE username LIKE CONCAT('%', @username, '%'); "; 6. command.Parameters.AddWithValue("@username", user\_name); 7. MySqlDataReader reader = command.ExecuteReader(); 8. List<Customer> customersList = new List<Customer>(); 9. **while** (reader.Read()) 10. { 11. Customer customer = ParseCustomer(reader); 12. customersList.**Add**(customer); 13. } 14. reader.Close(); 15. connection.Close(); 16. **return** customersList; 18. } |

Також для кожної з таблиць було реалізовано генерацію псевдовипадкових даних за допомогою SQL запитів:

Приклад функції для генерування псевдовипадкових даних користувача:

|  |
| --- |
| 1. **public** **void** Generate(**int** numberOfCustomers) 2. { 3. connection.Open(); 4. MySqlCommand command = connection.CreateCommand(); 5. command.CommandText = @"INSERT into customers (username, address, phone\_number) 6. SELECT SUBSTR(MD5(RAND()), 1, 8) AS randomString, 7. SUBSTR(MD5(RAND()), 1, 8) AS randomString, 8. LPAD(FLOOR(RAND() \* 9999999999.99), 10, '0') AS randomString "; 9. **int** res = command.ExecuteNonQuery(); 10. connection.Close(); 11. } |

Також для прогнозування кількості покупок товару за його ціною було використано алгоритм лінійної регресії. Лінійна регресія – одна з найважливіших та широко використовуваних технік регресії. Однією з його переваг є легкість інтерпретації результатів. Для реалізації лінійної регресії була використана бібліотека ML.NET .

**Аналіз функціонування засобів реплікації**

Реплікація - одна з технік масштабування баз даних. Ця техніка полягає в тому, що дані з одного сервера бази даних постійно копіюються (реплікуються) на один або кілька інших (звані репліками).

Реплікація бази даних MySQL типу Master-Slave використовується для більш спрощеного обслуговування множинних копій даних БД MySQL копіюючи дані автоматично з бази даних Master на базу даних Slave.

**Аналіз функціонування засобів резервування/відновлення**

Для резервування та відновлення бази даних було використано утиліту mysqldump. Mysqldump - це ефективний інструмент створення резервної копії бази даних MySQL. Він дозволяє створити \*.sql файл із сукупністю (дампом) всіх таблиць та даних основної бази даних.

За допомогою mysqldump, можна створювати резервну копію локальних баз даних, так і відновлювати їх на віддалених базах даних.

Для створення резервної копії бази даних, необхідно виконати команду:

|  |
| --- |
| C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -uroot -p16842778 db3>db3\_mydump.sql |

Отримали резервну копію бази даних «db3»:



Для відновлення бази даних необхідно виконати команду:

|  |
| --- |
| C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -uroot -p16842778 db3<C:\Program Files\MySQL\MySQL Server 8.0\bin\db3\_mydump.sql |

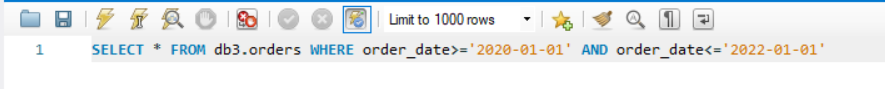
**Аналіз результатів підвищення швидкодії запитів**

Для підвищення швидкодії SQL запитів були використані індекси.

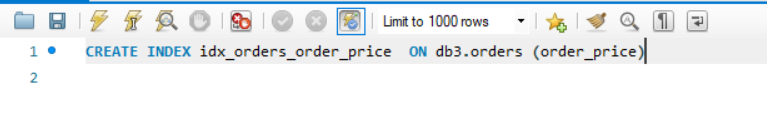
Для підвищення швидкодії фільтрації замовлень за датою та фільтрації продуктів за ціною був використаний індекс типу B-tree. За допомогою B-tree можна проіндексувати будь-які дані, які можуть бути відсортовані, тобто для яких застосовні операції порівняння більше/менш/рівно. Сюди можна віднести числа, рядки, дати та час, логічний тип та будь-які дані, які можна ними закодувати.

Проаналізуємо підвищення швидкодії виконання запитів фільтрації замовлень за датою:

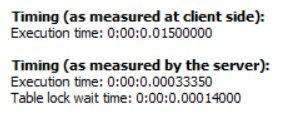
Команда для аналізу запиту:



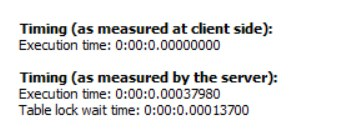
Команда для створення індексу:

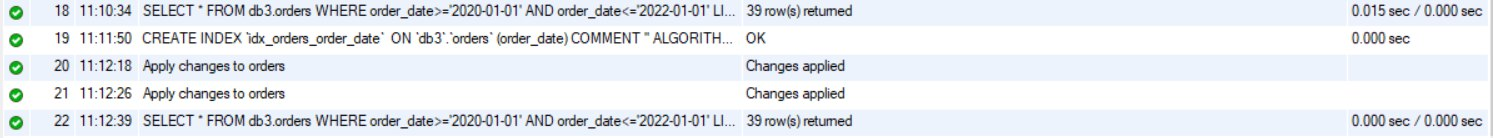


Спочатку ми здійснили фільтрацію замовлень без використання індексу: запит виконався за 0.015 секунд:



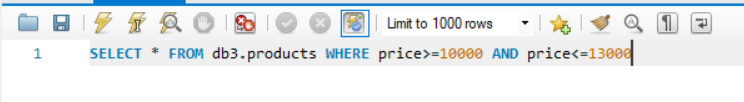
Потім ми створили B-tree індекс та здійснили той же запит: запит з використанням індексу виконався за 0.000 секунд.



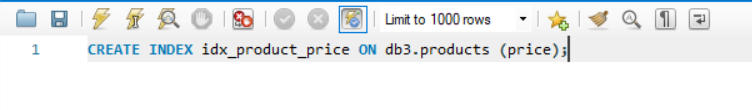


Проаналізуємо підвищення швидкодії виконання запиту фільтрації продуктів за ціною :

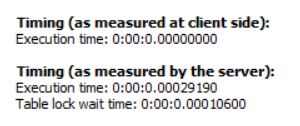
Команда для аналізу запиту:



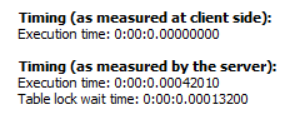
Команда для створення індексу:

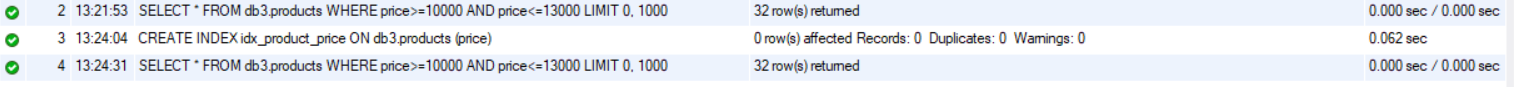


Спочатку ми здійснили фільтрацію продуктів за ціною без використання індексу: запит виконався за 0.000 секунд:



Потім ми створили B-tree індекс та здійснили той же запит: запит з використанням індексу виконався за 0.000 секунд.





Можемо підсумувати, що швидкість пошуку у таблиці orders збільшилась з використанням B-tree індексу. Для цього ми порівняли швидкість виконання до створення і після створення. А швидкість пошуку у таблиці products залишилась сталою.

**Опис результатів аналізу предметної галузі**

1. Генерація даних

Генерація псевдо-випадкових даних здійснена за допомогою SQL запитів, замість інформації з дата сетів.

1. Аналіз даних

За допомогою бібліотеки ScottPlot, було забезпечено візуальне представлення графіків, що представляють кількість замовлень за певний період часу та рейтинг найпопулярніших товарів.

1. Валідація та фільтрація даних

Фільтрація дає змогу знаходити “корисні” дані, а за допомогою валідації здійснюється перевірка вхідних даних на відповідність заданим структурам даних та вимогам до них.

**Висновки**

Особливу увагу під час розроблення даного програмного продукту було приділено розробленню бази даних, що містить логічні зв’язки між таблицями, та реалізації зрозумілого та зручного інтерфейсу користувача.

Розробка виконана у повному обсязі, всі вимоги враховані, продукт протестовано.

Використання розробленого додатку дозволить зберігати та аналізувати поточні ціни на товари за певною категорією, прогнозувати потреби покупців та визначати певні цінові тренди.

В ході виконання даного курсового проекту було досягнуто

поставленої мети: було набуто практичних навичок розробки сучасного

програмного забезпечення, що взаємодіє з SQL базами даних, а також

були здобуті навички оформлення відповідного текстового, програмного та

ілюстративного матеріалу у формі проектної документації.

**Література**

* + - 1. С# [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/C_Sharp>
      2. .NET Framework [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/.NET_Framework>
      3. ScottPlot [Електронний ресурс]. – Режим доступу до ресурсу: <https://scottplot.net/>
      4. ML.NET [Електронний ресурс]. – Режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>
      5. MySQL [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MySQL>
      6. Налаштування створення резервних копій БД. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.sqlshack.com/how-to-backup-and-restore-mysql-databases-using-the-mysqldump-command/>

**Додатки**

**Додаток 1:**

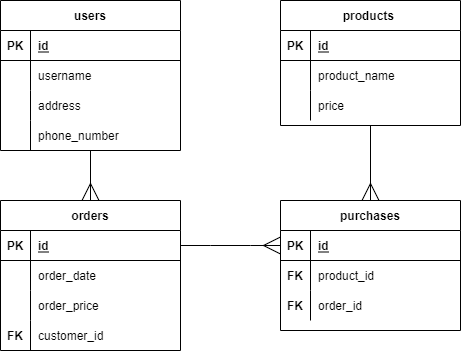


Рис. Структура бази даних

**Додаток 2:**

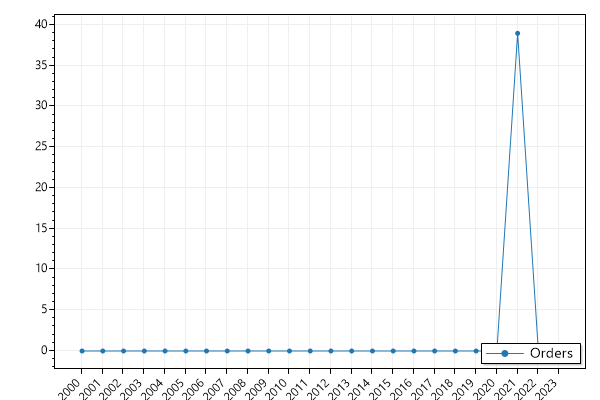


Рис. Графік замовлень за період 2000-2023 р.р.

**Додаток 3:**

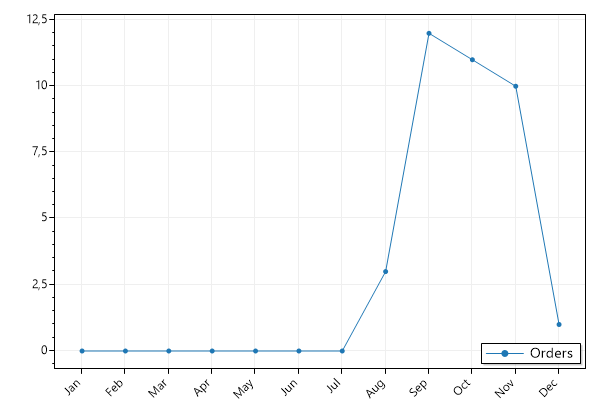


Рис. Графік замовлень за 2021 рік

**Додаток 4:**

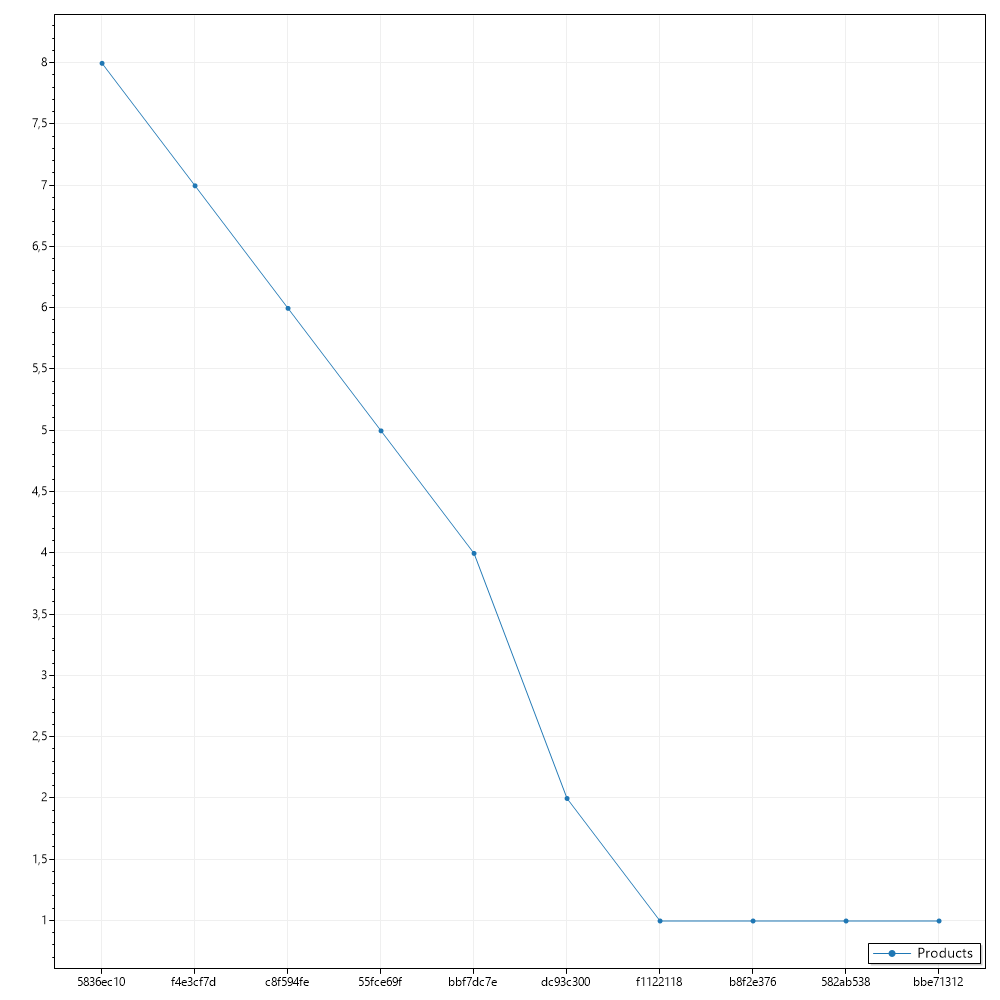


Рис. Графік десяти найпопулярніших товарів