

**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 інтернет магазинів споживчих товарів  
(назва теми)**

**Студентки  
групи КП-03**

**Євтушенко Вікторії  
Павлівни  
(ПБ)**

\_\_\_\_\_  
(підпис)

**Викладач  
к.т.н, доцент кафедри СПіСКС  
Петрашенко А.В.**

\_\_\_\_\_  
(підпис)

**Захищено з оцінкою** \_\_\_\_\_

**Київ – 2021**

## Анотація

Метою розробки даного курсового проекту є набуття виконавицею практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних та навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у вигляді проектної документації.

У результаті виконання курсового проекту були освоєні навички розроблення програмного забезпечення для реляційних баз даних, володіння основами користування СУБД, засобами аналізу великих обсягів даних, індексації таблиць бази даних та роботи з ними.

Темою даного курсового проекту є система контролю успішності учнів навчального закладу.

Галуззю застосування даної розробки є заклади освіти, тобто школи та університети.

Результатами даного проекту став програмний додаток для роботи з наближеною до реальної базою даних навчального закладу.

## Зміст

### Зміст

Анотація.....	2
Вступ.....	4
Аналіз інструментарію при виконанні лабораторної роботи .....	5
Структура бази даних .....	6
Опис програмного забезпечення.....	7
Аналіз функціонування засобів реплікації.....	9
Аналіз функціонування засобів резервування .....	11
Аналіз результатів підвищення швидкодії виконання запитів ....	12
Опис результатів аналізу предметної галузі .....	13
Висновки.....	14
Література.....	15
Додаток 1 .....	16
Додаток 2 .....	16

## Вступ

Неможливо уявити сучасний світ без електронних систем зберігання та аналізу даних. Навчальні ж заклади, переважно молодшої та середньої освіти, у своїй більшості відстають від такого розкладу подій. Більшість даних зберігається на паперових версіях, що ускладнює можливість аналізу даних та визначення подальших дій з ними.

Саме для цю проблему вирішую додаток, розроблений у якості курсової роботи. Він дозволяє полегшити процес керування даних (внесення, видалення, перегляд, оновлення) їх аналіз та фільтрацію, а також забезпечує надання графічного їх представлення.

Галуззю застосування даної розробки є заклади освіти, тобто школи та університети.

Метою розробки даного курсового проекту є здобуття 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.

У результаті виконання курсового проекту отримано програмне забезпечення для реляційних баз даних.

Дані для аналізу були згенеровані псевдовипадковим чином за допомогою даних, взятих із датасетів, що розміщені їх авторами у вільному доступі, але також можуть бути згенеровані за допомогою відповідних бібліотек.

## Аналіз інструментарію при виконанні лабораторної роботи

### 1. Обґрунтування вибору мови програмування:

Мовою програмування для даного проекту було обрано C# через велику кількість відкритих бібліотек та фреймворків, що спрощує отримання бажаного результату.

Також, важливою перевагою C#, є її об'єкто-орієнтованість, що забезпечує зручне керування об'єктами бази даних та зв'язками між ними.

### 2. Обґрунтування вибору бібліотек та фреймворків

було використано наступні бібліотеки та фреймворки:

- MySqlConnection – забезпечує підключення до бази даних.
- ScottPlot – використовується для забезпечення ілюстративного матеріалу.

### 3. Обґрунтування вибору СУБД

Було обрано MySQL, як найбільш відповідну СУБД для даного проекту. Ця СУБД є масштабованою, має безліч визначених функцій і можливостей індексації та є реляційною, що дуже важливо для складної системи зв'язків сутностей таблиць БД шкільного закладу.

Дана СУБД надає можливості легко створювати і керувати індексами, тригерами, поточні реплікації та резервування БД.

### 4. Обґрунтування вибору середовища розробки

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

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

## Структура бази даних

База даних проекту складається з чотирьох таблиць:

1. students – таблиця, що містить інформацію про учнів.
  - id (PK) – унікальний ідентифікатор учня.
  - first\_name – ім'я учня.
  - last\_name – прізвище учня.
  - age – вік учня.
  - average\_score – середній бал учня.
2. subject – таблиця, що містить інформацію про предмети.
  - id (PK) – унікальний ідентифікатор предмету.
  - name – назва предмету.
3. teachers – таблиця, що містить інформацію про вчителів.
  - id (PK) – унікальний ідентифікатор вчителя.
  - first\_name – ім'я вчителя.
  - last\_name – прізвище вчителя.
  - subject\_id (FK) – ідентифікатор предмету, що викладає вчитель.
4. student\_to\_teacher – допоміжна таблиця, що забезпечує зв'язок “many-to-many” між об'єктами учня та вчителя.
  - id (PK) – унікальний ідентифікатор зв'язку.
  - student\_id (FK) – ідентифікатор учня.
  - teacher\_id (FK) – ідентифікатор вчителя.

Усі таблиці є приведеними до третьої нормальної форми.

## **Опис програмного забезпечення**

### **Загальна структура програмного забезпечення**

Програмні засоби містять наступні компоненти:

1. Підсистема попередньої обробки даних, що складається з:
  - 1.1. Засоби генерації даних. Є рандомізація даних, також було створено CRUD операції, що дозволяють додавати дані у таблиці.
  - 1.2. Засоби фільтрації та валідації даних.
2. База даних системи призначена для зберігання і аналізу інформації розробленої моніторингової системи.

### **Опис модулів програмного забезпечення**

Даний проект було оформлено згідно із структурою MVC (model – viewer – controller).

Модель (model) відповідає за підключення до бази даних, створення класів об'єктів та генерації даних.

Від подання (viewer) залежить інформація, що передається у інтерфейс, з яким працює користувач, у даному випадку – консольний.

Контроллер (controller) зв'язує між собою модель та подання. Він відповідає за прийняття та обробку даних, а далі – за виклик потрібного метода.

### **Опис основних алгоритмів роботи програмного забезпечення**

Операції генерації даних та їх виведення виконуються за допомогою згаданої вище бібліотеки MySqlConnectionor. Генерація виконується також за рахунок псевдовипадкового вибору даних із заздалегідь створених на основі датасету файлів. Валідація даних, введених користувачем, реалізована, в основному, завдяки методу TryParse() та перевірці записів таблиць, для чого були створені тригери.

```

CREATE
  DEFINER = 'root'@'localhost'
TRIGGER schooldb.trigger1
  BEFORE INSERT
  ON schooldb.students
  FOR EACH ROW
BEGIN
  IF(EXISTS(SELECT id from students WHERE first_name=NEW.first_name AND last_name=NEW.last_name AND age=NEW.age AND average_score = NEW.average_score ))
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Such student already exists';
END IF;
END

```

Тригер для перевірки перед вставкою для таблиці учнів

```

CREATE
  DEFINER = 'root'@'localhost'
TRIGGER schooldb.trigger2
  BEFORE INSERT
  ON schooldb.teachers
  FOR EACH ROW
BEGIN
  IF(EXISTS(SELECT id from teachers WHERE first_name=NEW.first_name AND last_name=NEW.last_name AND subject_id=NEW.subject_id))
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Such teacher already exists';
END IF;
END

```

Тригер для перевірки перед вставкою для таблиці вчителів



```

CREATE
  DEFINER = 'root'@'localhost'
TRIGGER schooldb.trigger3
  BEFORE INSERT
  ON schooldb.student_to_teacher
  FOR EACH ROW
BEGIN
  IF(EXISTS(SELECT id from first_student_to_teacher WHERE student_id=NEW.student_id AND teacher_id=NEW.teacher_id ))
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Such connection already exists';
END IF;
END

```

Тригер для перевірки перед вставкою для таблиці зв'язку між учнем та вчителем

```

CREATE
  DEFINER = 'root'@'localhost'
TRIGGER schooldb.trigger4
  BEFORE INSERT
  ON schooldb.subjects
  FOR EACH ROW
BEGIN
  IF(EXISTS(SELECT id from subjects WHERE name=NEW.name))
  THEN SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Such subject already exists';
END IF;
END

```

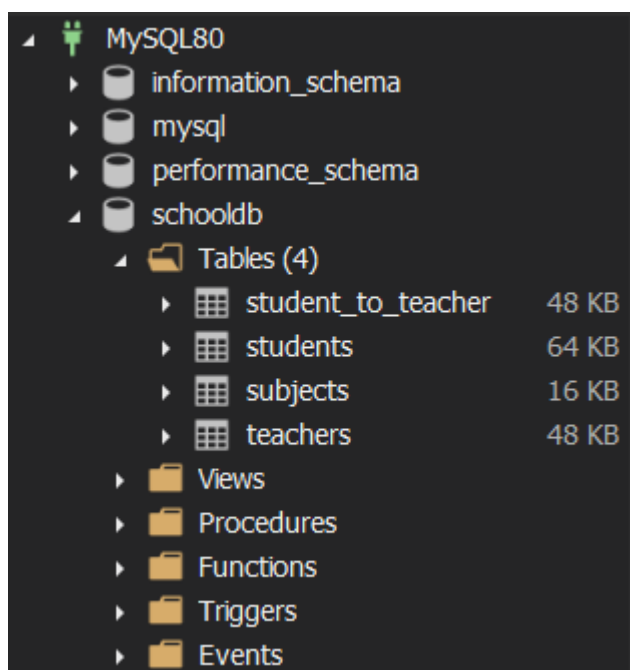
Тригер для перевірки перед вставкою для таблиці предметів

## Аналіз функціонування засобів реплікації

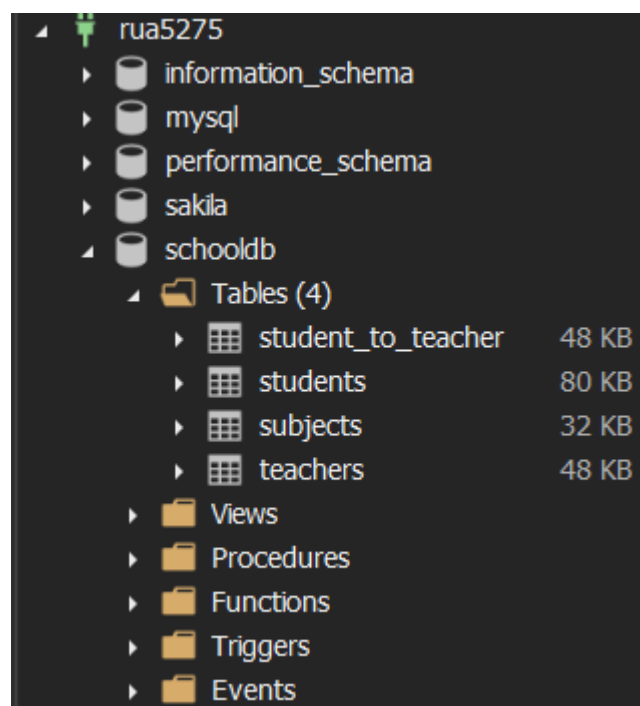
Реплікація – техніка масштабування даних, механізм розподілу даних за вузлами, що дозволяє зберігати копії тих самих даних на різних вузлах мережі з метою прискорення пошуку і підвищення стійкості до відмов. Відношення чи фрагмент є реплікованим, якщо його копії зберігаються на двох або більше вузлах.

У MySQL реплікація досягається шляхом створення серверу на іншому порті. Була налаштована Master-Slave реплікація, за якої зчитування даних відбувається на slave-сервер, а запис (та, у деяких випадках, зчитування) – на master.

У випадку технічних проблем, slave бере на себе обов'язки серверу master і стає первинним сервером для роботи.



Первинний сервер (master)



Вторинний сервер (slave)

## Аналіз функціонування засобів резервування

Механізм резервації був реалізований за допомогою стандартної утиліти MySQL `mysqldump`. Вона дозволяє серверу бази даних генерувати текстовий файл з SQL-командами, що потім відновлюють базу даних на вказаному сервері у тому вигляді, у якому вона була на момент створення файлу. Також резервація була застосована при створенні вторинного серверу під час реплікації.

Такий спосіб має безліч переваг, серед яких відсутність зайвої роботи із файлом, невелика кількість дій та незалежність від версії MySQL.

Були використані наступні команди:

Створення копії:	<code>mysqldump -uroot -p schooldb &gt; D:\victo\kpi\data\schooldb.sql</code>
Відновлення даних:	<code>mysql -u vic -h 192.168.1.109 -p schooldb &lt; D:\victo\kpi\data\schooldb.sql</code>

## Аналіз результатів підвищення швидкодії виконання запитів

Для підвищення швидкодії виконання запитів у таблицях були створені індекси. Вони використовувалися у запитах, пов'язаних із пошуком. За замовчанням, у MySQL можливо створити індекси типу B-Tree (збалансоване дерево). Переваги B-Tree полягають у наступному:

- зберігає ключі в упорядкованому порядку для послідовного проходження;
- використовує ієрархічний індекс, щоб мінімізувати кількість читань з диска;
- використовує частково повні блоки для прискорення вставок і видалення;
- зберігає індекс збалансованим за допомогою рекурсивного алгоритму;
- мінімізує відходи, переконавшись, що внутрішні вузли заповнені принаймні наполовину.

Збалансоване дерево також може обробляти довільну кількість вставок і видалення.

Нижче наведені приклади роботи запитів з таблицею до додання індексу та після. Варто зазначити, що запити відбувалися на таблицях з невеликою кількістю попередньо створених даних, але навіть так можна побачити, що різниця доволі значна.

```
ALTER TABLE schooldb.students  
ADD INDEX IDX_students_average_score(average_score);
```

Приклад запиту на створення індексу у таблиці студентів

✓ Query executed successfully. 0:00:00.166

Час виконання до створення індексу

✓ Query executed successfully. 0:00:00.012

Час виконання після створення індексу

## Опис результатів аналізу предметної галузі

### *1. Генерація даних*

Генерація даних здійснена за допомогою функцій генерування псевдо-випадкових даних, оснований на інформації з датасетів.

### *2. Аналіз даних*

За допомогою вищезгаданої бібліотеки ScottPlot, було забезпечено візуальне представлення діаграми, що характеризує БД.

### *3. Валідація та фільтрація даних*

Забезпечена всередині універсальних модулів контролера та представлення. Валідація працює за допомогою перевірки правильності типу введених даних методом TryParse().

### *4. Пошук корисних даних*

Забезпечено створенням відповідних запитів до бази даних у модулі.

## Висновки

В процесі виконання даної курсової роботи були опановані практичні навички з роботи із великим обсягом даних за допомогою мови програмування C# та СУБД MySQL, освоєно їх реплікацію, створення резервних копій та отримання цієї інформації різними шляхами.

Було створено програмне забезпечення, в якому було реалізовано виконання таких пунктів:

1. Генерація псевдорандомізованих даних.
2. Статистичний аналіз даних за допомогою бібліотек C# і .NET.
3. Візуалізація даних завдяки бібліотеці C#.
4. Консольний інтерфейс.

В ході виконання даного курсового проекту було досягнуто поставленої мети: виконавиця набула практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з реляційними базами даних та навичок оформлення відповідного текстового, програмного та ілюстративного матеріалу у вигляді проектної документації.

У результаті виконання курсового проекту було написано програмне забезпечення для релятивістських баз даних, опановано основи використання СУБД, а також, було закріплено вміння користуватися інструментальними засобами аналізу, візуалізації та роботи з великими масивами даних.

## Література

1. MySQL [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://www.mysql.com/>
2. Документація C# [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://docs.microsoft.com/en-us/dotnet/csharp/>
3. Документація .NET [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://docs.microsoft.com/ru-ru/dotnet/core/introduction>
4. VisualStudio [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://visualstudio.microsoft.com/ru/>
5. MySQL Connector [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://dev.mysql.com/doc/visual-studio/en/>
6. ScottPlot [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://scottplot.net/>
7. Replication in MYSQL [Електронний ресурс]. – Режим доступу до ресурсу:  
<https://dev.mysql.com/doc/refman/8.0/en/replication.html#:~:text=Replication%20enables%20data%20from%20one,receive%20updates%20from%20a%20source>

## Додаток 1

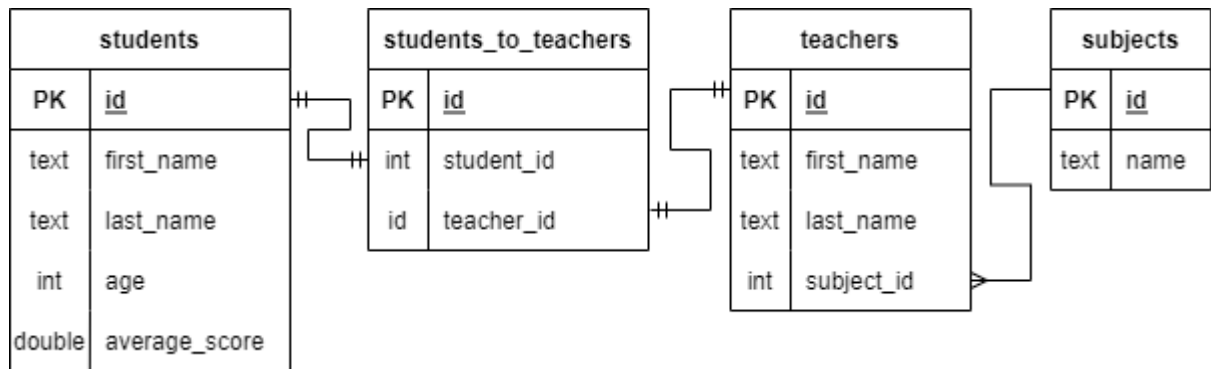


Рис. 1. Структура бази даних

## Додаток 2

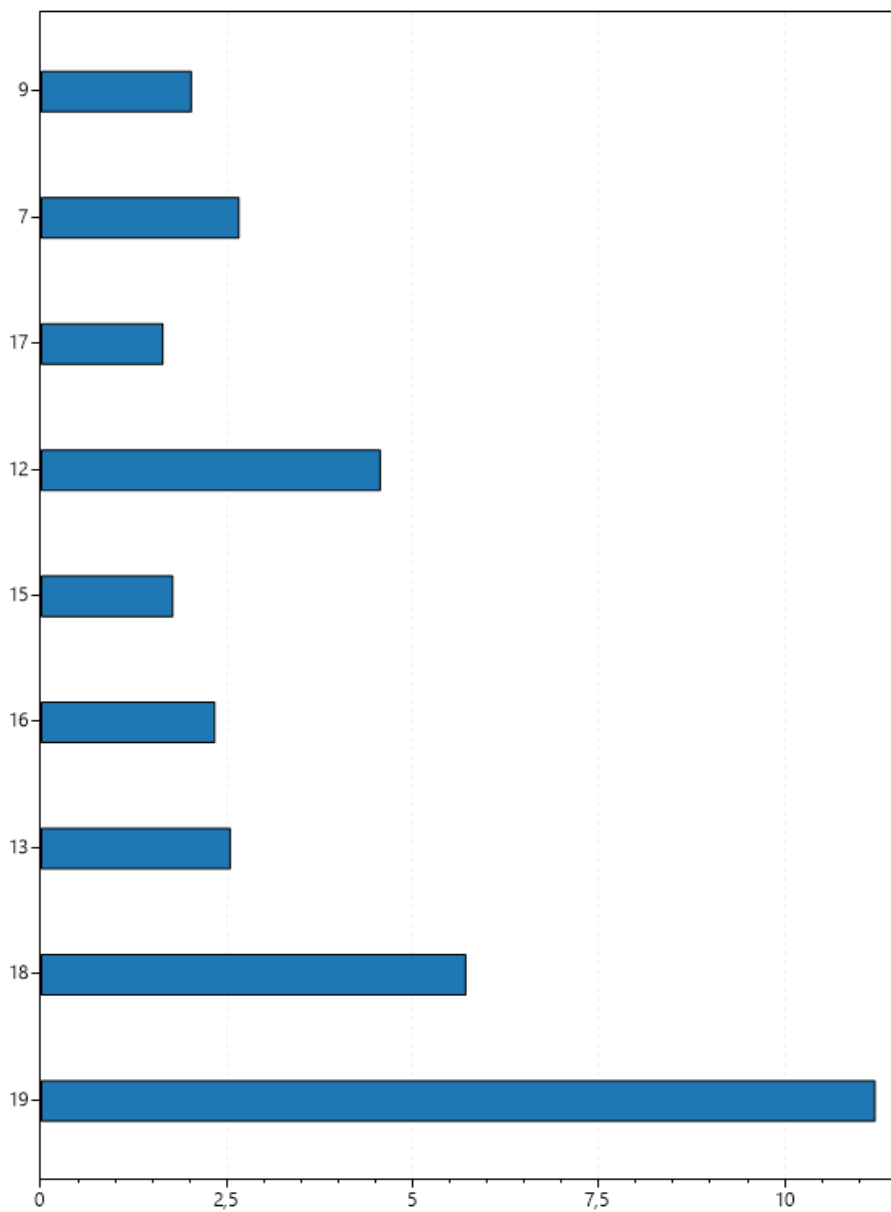


Рис. 2. Діаграма відношення віку учнів та їх успішності