

**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 контролю успішності учня

**Студент
групи КП-01**

Беліцький О.С

(підпис)

**Асистент
кафедри СПіСКС**

Радченко К.О.

(підпис)

Захищено з оцінкою _____

Анотація.

У даній роботі було створено систему електронного журналу, де можна переглядати склад класів та оцінки учнів з різних навчальних дисциплін. Робота виконана на мові програмування C# версії 9.0, фреймворку .Net версії 5.0 та SQLite 6.0.1.

Вступ

Призначення роботи:

Робота є результатом вивчення дисципліни "Бази даних". В ній було використано CRUD-операції, засоби підвищення швидкодії, візуалізації даних, роботи з файлами баз даних. Було використано знання з нормальних форм баз даних, тригерів.

Актуальність роботи:

Наразі все більше популяризуються електронні журнали для шкіл і до сих пір немає одного додатку, який би підходив всім. Це означає, що питання зі зручності, швидкодії, доступності для електронних щоденників досі відкрите.

Загальна постановка завдання:

Об'єктом розробки у курсовій роботі є інформаційно-аналітична система (ІАС) широкого призначення, яка дозволяє отримувати інформацію або генерувати та здійснювати її обробку та аналіз.

ІАС, що розроблена: система контролю успішності учнів навчального закладу: збір, фільтрація та аналіз оцінок учнів різних вікових категорій за різними напрямками підготовки з метою прогнозування та вдосконалення навчального процесу.

ЗМІСТ

1. Аналіз інструментарію для виконання курсової роботи.
2. Структура бази даних.
3. Опис програмного забезпечення.
 - 3.1 Загальна структура програмного забезпечення.
 - 3.2 Опис модулів програмного забезпечення.
 - 3.3 Опис основних алгоритмів роботи.
4. Аналіз функціонування засобів резервування/відновлення.
5. Аналіз результатів підвищення швидкодії запитів.
6. Опис результатів аналізу предметної галузі.
7. Додатки

1. Аналіз інструментарію для виконання курсової роботи.

Для виконання курсової роботи буде використано SQLite та браузером, що дозволяє переглядати реляційні бази даних "DB Browser for SQLite". Вибір впав саме на цю СУБД, бо з нею я працював раніше та маю певні навички володіння SQLite та DB Browser for SQLite.

Вибір засобів розробки:

Сервер бази даних – SQLite.

Для реалізації даної системи буде використовуватися мова програмування C# 9.0 оскільки ця мова має засоби для роботи з реляційними СУБД та має об'єктно-орієнтований підхід до програмування - не потрібно тримати всю інформацію в голові, достатньо описати абстрактні конструкції і зв'язки між ними. В такому разі роботу програми можна описати принципом:

подати вхідні дані -> магія -> потрібний результат

Одним з найважливіших плюсів є наявність великої кількості бібліотек і шаблонів, що дозволяють не витрачати час на винахід свого власного велосипеда. Зокрема будуть використовуватися наступні бібліотеки:

- **Microsoft.Data.Sqlite** - це спрощений постачальник ADO.NET для SQLite. На основі цієї бібліотеки побудований постачальник Entity Framework Core для SQLite. Проте, її також можна використовувати незалежно або з іншими бібліотеками доступу до даних. *Можна використовувати System.Data.SQLite, але попередня бібліотека є більш оптимізованою та має кращу читабельність.*
- **System.IO** - простір імен. Містить типи, що дозволяють здійснювати читання і запис у файли і потоки даних, а також типи для базової підтримки файлів і папок.
- **System.Diagnostics** - простір імен. Надає класи, що дозволяють здійснювати взаємодію з системними процесами, журналами подій і лічильниками продуктивності.
- **System.Collections.Generic** - простір імен. Містить інтерфейси і класи, що визначають універсальні колекції, які дозволяють користувачам створювати строго типізовані колекції, що забезпечують підвищену продуктивність і безпеку типів у порівнянні з неуніверсальними строго типізованими колекціями.
- **ScottPlot** - це бібліотека інтерактивного креслення для .NET. Вона має власний пакет, що дозволяє створювати графіки в серверних середовищах або консольних додатках і зберігати їх у вигляді зображень. ScottPlot орієнтована на .NET Standard 2.0, тому її можна використовувати як в додатках .NET Core, так і в .NET Framework. API

ScottPlot заснований на аргументах (за зразком Matplotlib для Python), і більшість графіків можна створити за допомогою одного рядка коду (що настраюється за бажанням за допомогою додаткових аргументів).

2. Структура бази даних.

База даних складається з чотирьох таблиць:

- 11A
- 11B
- math
- biology

11A та 11B - таблиці класів.

Math та Biology - таблиці предметів.

Таблиці класів мають індивідуальний номер учня, його прізвище і ім'я.

Таблиці предметів мають індивідуальний номер учня, його прізвище, ім'я, клас, в якому навчається учень, та його оцінка.

3. Опис програмного забезпечення.

3.1 Загальна структура програмного забезпечення.

У програмному забезпеченні створено CRUD-операції, операції змін оцінок, отримання всього списку студентів.

Забезпечено функцію генерування даних. Функція генерує імена студентів, їх оцінки по кожному з предметів (0 за замовчуванням). Функція не повинна генерувати пусті значення.

Також створено функції фільтрації даних по заданим параметрам та валідації цих самих даних. Проводиться структурування інформації з різних ресурсів, валідація, знаходження взаємозв'язку між учнем, предметом та оцінками.

Додано засоби підвищення швидкодії, резервування та відновлення даних.

Також система може виводити діаграми успішності учнів по предметам.

Вимоги до інтерфейсу користувача

Інтерфейс користувача – консольний додаток. Задачею інтерфейсу користувача є налаштування засобів та підсистем, запуск/завершення їх роботи, генерація звітної інформації (діаграм) у вигляді збережених файлів-зображень. Звітна інформація стосується візуалізації роботи засобів аналізу даних.

3.2 Опис модулів програмного забезпечення.

Програмне забезпечення створено за структурою MVC.

Модель - окремі файли роботи з таблицями класів, предметів.

Вид - подається у вигляді консольного виводу або окремих файлів за зображенням діаграм.

Контролер - дані, що вводить користувач у консоль через консольний додаток.

Модель представлена файлами репозиторіїв сутностей, лише які мають право звертатися до бази даних. Це файли 11ARepository.cs, 11BRepository.cs, MathRepository.cs, BiologyRepository.cs.

Вид представлений файлом plot.png, куди зберігаються діаграми та консольним виводом у функції Program().

Контролер представлений файлом Program.cs, тому ще цей файл є повністю консольним додатком.

3.3 Опис основних алгоритмів роботи.

Для класів:

Insert - дозволяє створити нову сутність.

Delete by id - видалення запису за id

Update by id - оновлення запису за id

Get by id - отримання запису за id

Get all - отримати всі записи

Generate - згенерувати випадкові сутності

Для предметів:

Change mark - зміна оцінки учня за id

Get all - отримати всі записи

Diagram - створити діаграму

Додатково:

backup - створює копію бази даних

change connection - змінює зв'язок на іншу базу даних

4. Аналіз функціонування засобів резервування/відновлення.

Резервування реалізовано за допомогою бібліотеки *System.IO*, що забезпечує роботу з файлами. Функція `backup` створює повну копію бази даних, з якою працює програма. Функція `change connection` дозволяє змінити під'єднання до копії, якщо вона існує і навпаки.

5. Аналіз результатів підвищення швидкодії запитів.

Для підвищення швидкодії були використані індекси SQLite. Покажемо декілька прикладів.

Створимо і перевіримо роботу індексу для таблички класу 11A:

SQL 1

```
1 CREATE INDEX surname_name
2 ON _11A (surname, name);
```

Execution finished without errors.
Result: запит успішно виконаний. Зайняло 2мс
At line 1:
CREATE INDEX surname_name
ON _11A (surname, name);

SQL 1

```
1 EXPLAIN QUERY PLAN
2 SELECT
3   surname,
4   name
5 FROM
6   _11A
7 WHERE
8   surname = 'Іванов' AND name = 'Іван'
```

id	parent	notused	detail
1	2	0	0 SEARCH TABLE _11A USING COVERING INDEX surname_name (surname=? AND name=?)

Видно, що база даних виконує пошук саме з використанням створеного індексу.

Якщо програмно порівняти звернення для пошуку без індексу і з індексом, то результат буде наступним:

<div>10 Elapsed: 00:00:00.9599367</div> <div>Пошук 10-го учня зайняв майже секунду без індексу.</div>	<div>11 Elapsed: 00:00:00.2810704</div> <div>Пошук 11-го учня зайняв менше 0,3 секунди з індексом.</div>
---	--

Аналогічно створений індекс для 11В класу.

Індекс для предмету на прикладі біології за аналогічною логікою:

SQL 1

```
1  EXPLAIN QUERY PLAN
2  SELECT
3      surname,
4      name,
5      class,
6      mark
7  FROM
8      biology
9  WHERE
10     surname = 'Іванов' AND
11     name = 'Іван' AND
12     class = '11A' AND
13     mark = 11;
```

id	parent	notused	detail
1	2	0	0 SEARCH TABLE biology USING COVERING INDEX biologyIndex (surname=? AND name=? AND class=? AND mark=?)

Execution finished without errors.
Result: 1 строк возвращено за 13мс
At line 1:
EXPLAIN QUERY PLAN
SELECT
 surname,
 name,
 class,
 mark
FROM
 biology
WHERE
 surname = 'Іванов' AND
 name = 'Іван' AND
 class = '11A' AND
 mark = 11;

Тест для отримання всіх записів:

Elapsed: 00:00:00.0222677

Без індексу

Тест для отримання всіх записів:

Elapsed: 00:00:00.0205262

З індексом

Видно, що тут різниця не така критична, але для бази даних з великою кількістю записів ця різниця буде суттєвою.

6. Опис результатів аналізу предметної галузі.

Аналіз даних відбувається для створення діаграм, де діаграма виступає висновком аналізу даних. Функція отримує всі оцінки з предмета, та показує кількість кожної оцінки(див. додаток).

Висновки

Виконавши дану курсову роботу було створено систему електронного журналу. Ця система є прототипом справжнього додатку електронного журналу та вміщує в себе основні його функції.

Важливо, що в ході роботи над цим проєктом я набув кращих навичок володіння базами даних, вчасності, за допомогою SQLite та глибше розумію роботу баз даних, її внутрішньої логіки та зв'язків.

Метою розробки було створення бази даних, що буде мати записи про успішність учнів навчального закладу, та створення інтерфейсу користувача, що буде задовольняти функції збору, фільтрації та аналізу оцінок учнів різних вікових категорій за різними напрямками підготовки з метою прогнозування та вдосконалення навчального процесу. Я вважаю, що мета була виконана.

Компіляція всього коду відбувалася за допомогою утиліти dotnet в операційній системі Linux дистрибутив Ubuntu.

Література

- 1) SQLite: Light Database System, Bhosale S. T.Patil T.Patil P.
International Journal of Computer Science and Mobile Computing (2015)
- 2) A Structural Concept and Tool for Forensic Data Analysis and Recovery of Deleted SQLite Records, Meng C.Baier H., Digital Investigation (2019)
- 3) DB Browser for SQLite, Sqlitebrowser, sqlitebrowser.org (2016)
- 4) Recovery of SQLite Data Using Expired Indexes, Ramisch F. Rieger M.,
Proceedings - 9th International Conference on IT Security Incident Management and IT Forensics, IMF 2015 (2015).

7. Додатки

Приклади діаграм для предметів біологія та математика:

