**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

Кафедра математичної інформатики

**Звіт**

до лабораторної роботи на тему

**ФРАГМЕНТАРНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ТАБЛИЧНИМИ БАЗАМИ ДАНИХ**

Виконав студент 4-го курсу   
Білінська Софія Олегівна

Київ – 2021

**ЗМІСТ**

Постановка задачі 3

0)Функціональна специфікація СУТБД – діаграма прецедентів UML 4

2-3) Розробка локальної версії СУБД 8

10) REST web-сервіси 8

12) REST web-сервіси. Розробка OpenAPI Specification 8

18) Web-проект (AspNetWebApi) 8

24) Використання реляційної СУБД 8

19) Web-проект (ASP.NET MVC) 11

# Постановка задачі

Створити декілька варіантів реалізації системи управління табличними базами даних з використанням запропонованих технологій.

Загальні вимоги:

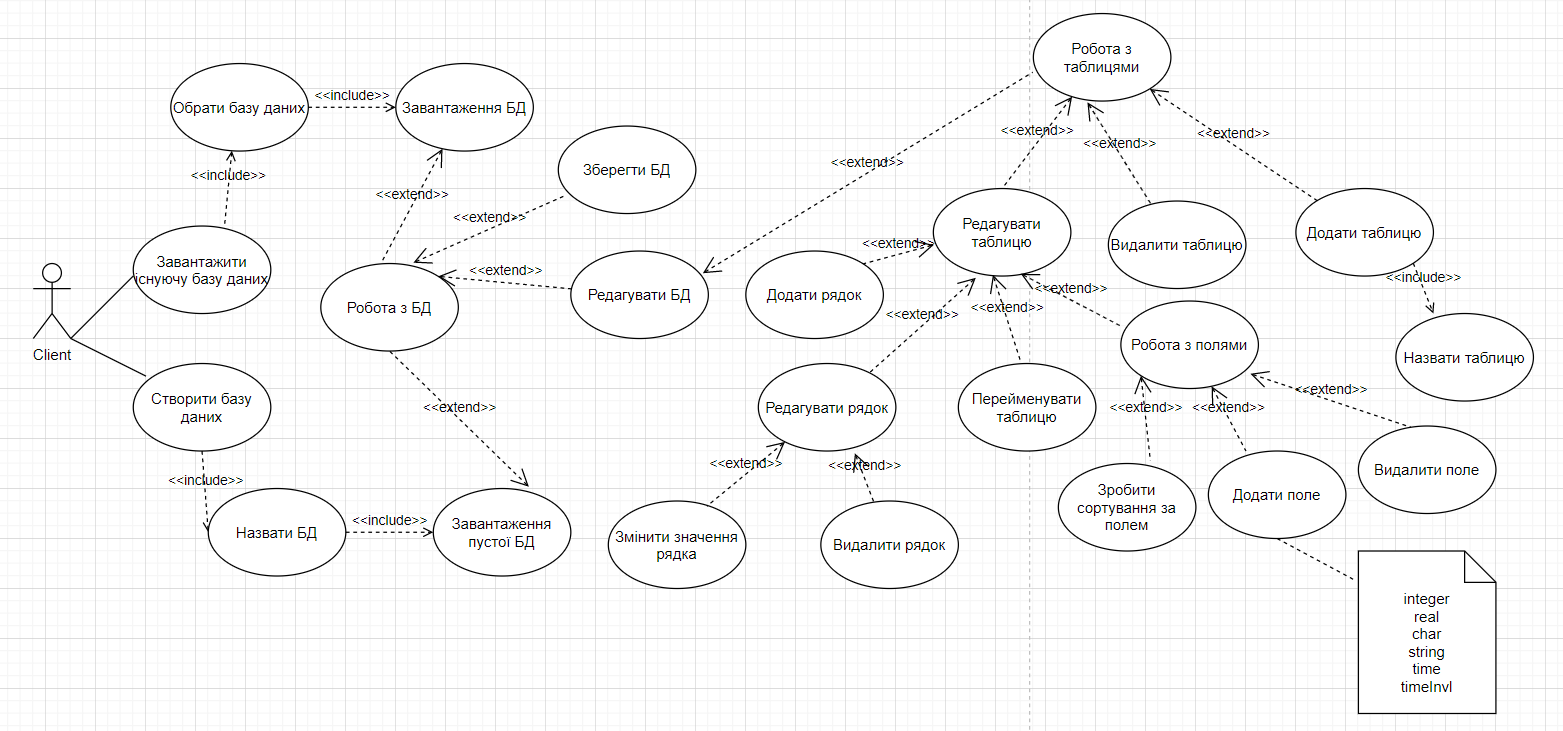
* кількість таблиць принципово не обмежена;
* кількість полів та записів у кожній таблиці принципово не обмежені;
* забезпечити підтримку типів integer, real, char, string
* реалізувати створення бази;
* реалізувати створення (з валідацією даних) та знищення таблиці з бази;
* реалізувати перегляд та редагування рядків таблиці;
* реалізувати збереження табличної бази на диску та, навпаки, її зчитування з диску.

Додаткові типи та операція над таблицями – варіант 60:

6) типи «time» та «timelnvl»

0) сортування таблиці за одим полем

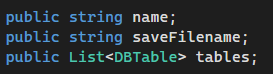
1. **Функціональна специфікація СУТБД – діаграма прецедентів UML**



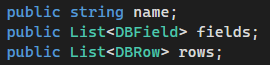
**2-3) Розробка локальної версії СУБД**

• Розробка класів

**Database -** клас бази даних, містить поле назви *Name* та колекцію таблиць *DBTables*:



**DBTable -** клас таблиці, містить поле назви *Name* та колекції рядків *DBRow* та стовпчиків *DBField*:



**DBRow -** клас рядку, містить колекцію значень *Elements* та колекцію *SerializableElement* :



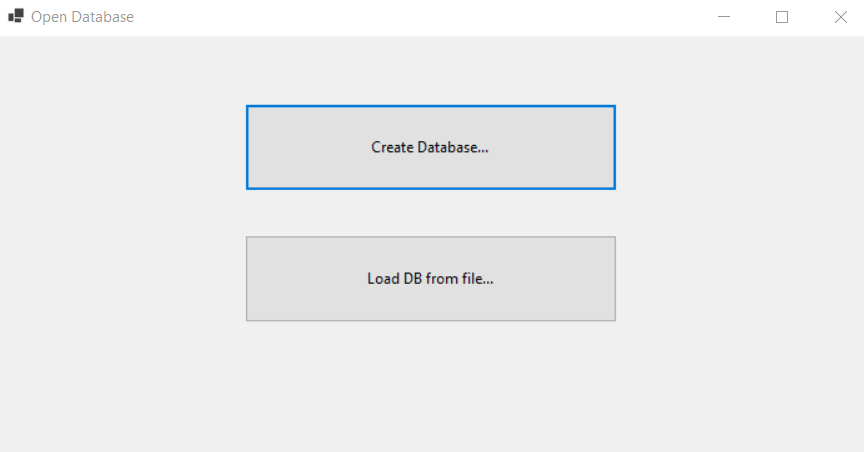
**DBField –** клас поля, містить *name* та *typeName*

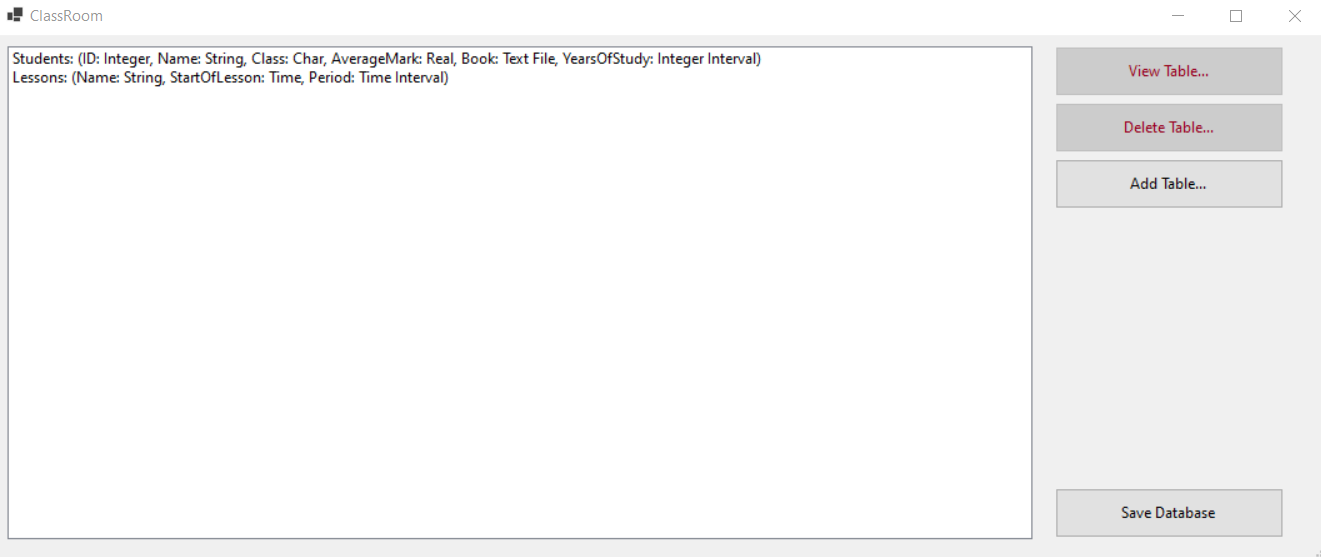


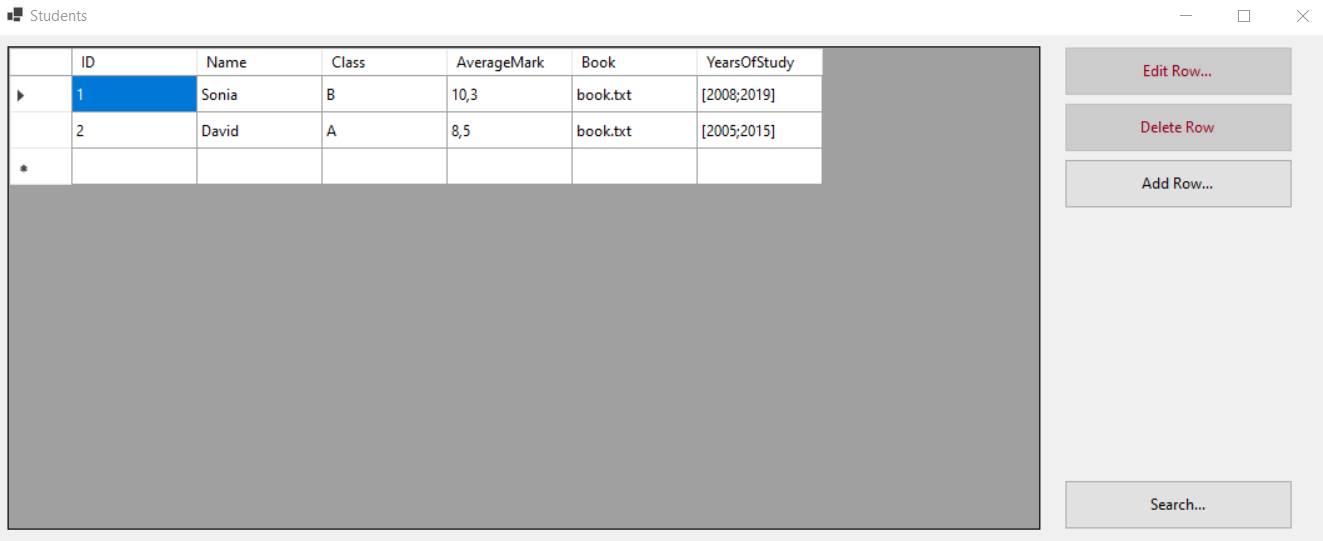
• Створення UML-діаграми класів

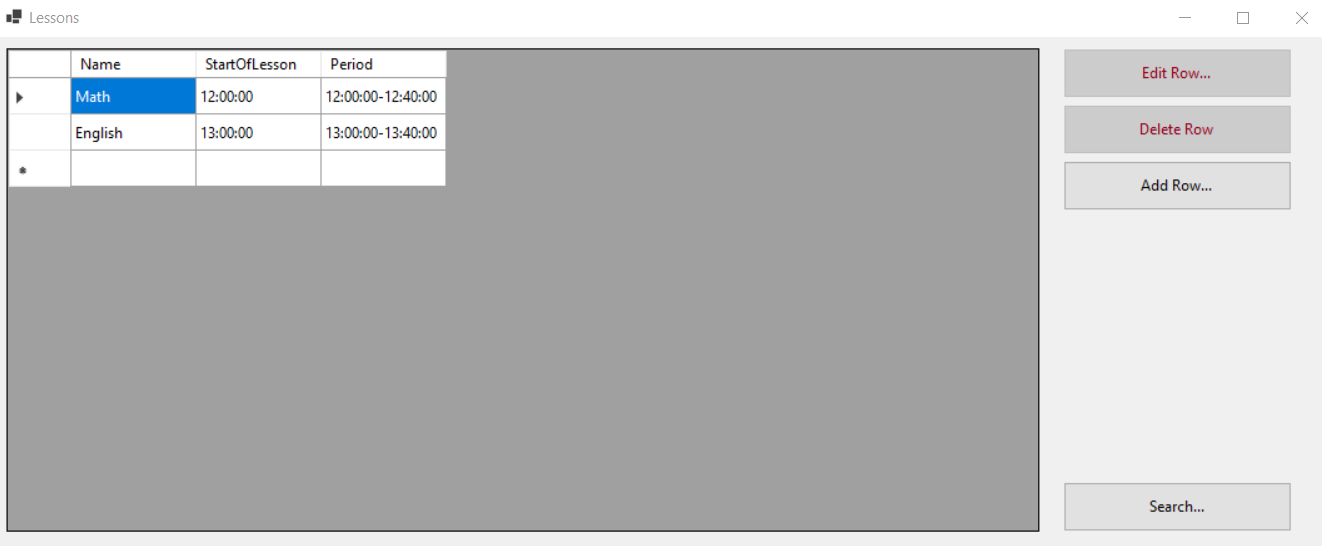


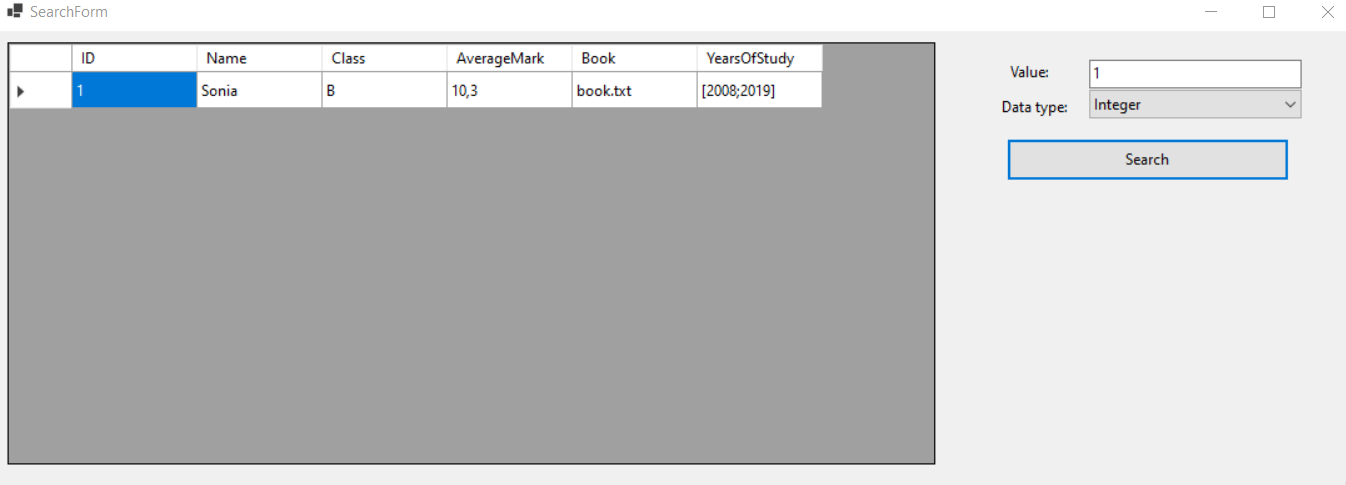
• Забезпечення інтерфейсу користувача на основі форм



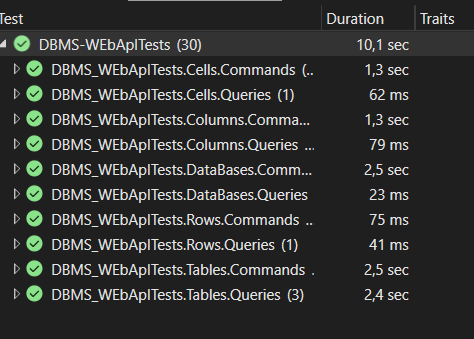








Unit-тести:



**10) REST web-сервіси**

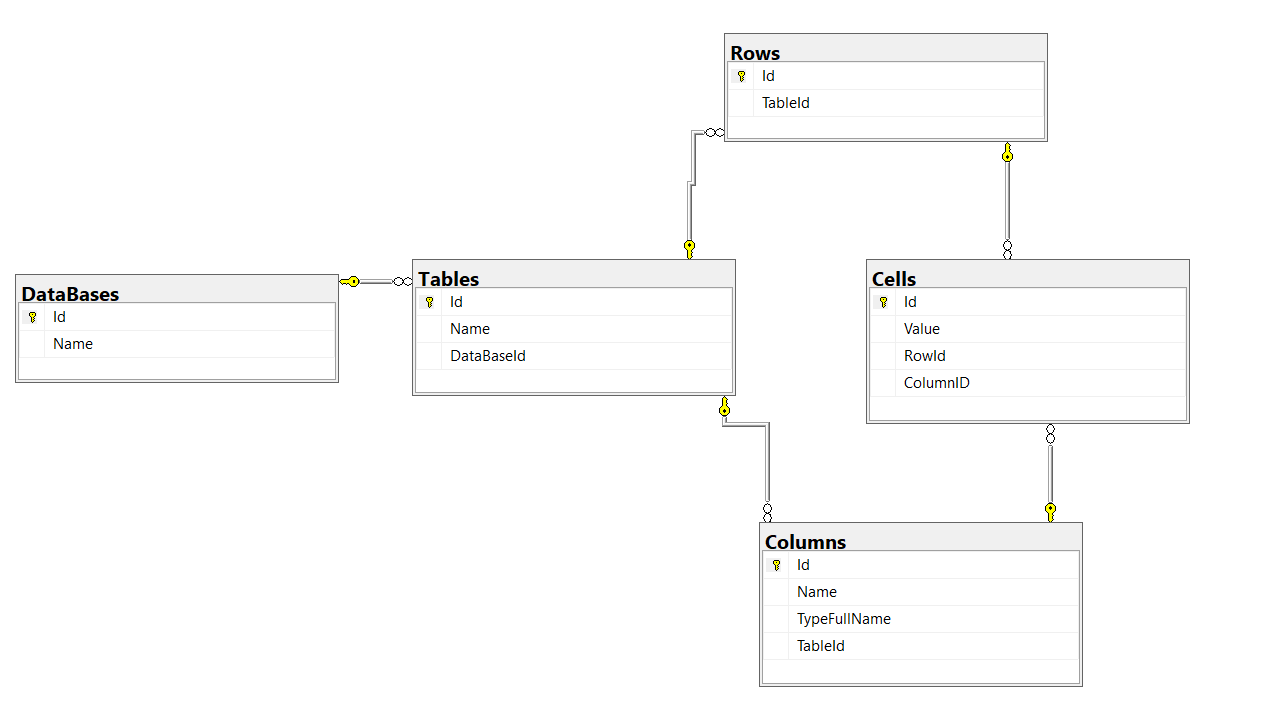
**12) REST web-сервіси. Розробка OpenAPI Specification**

**18) Web-проект (AspNetWebApi)**

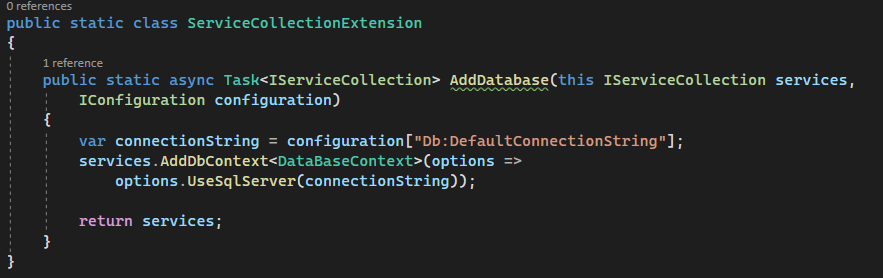
**24) Використання реляційної СУБД**

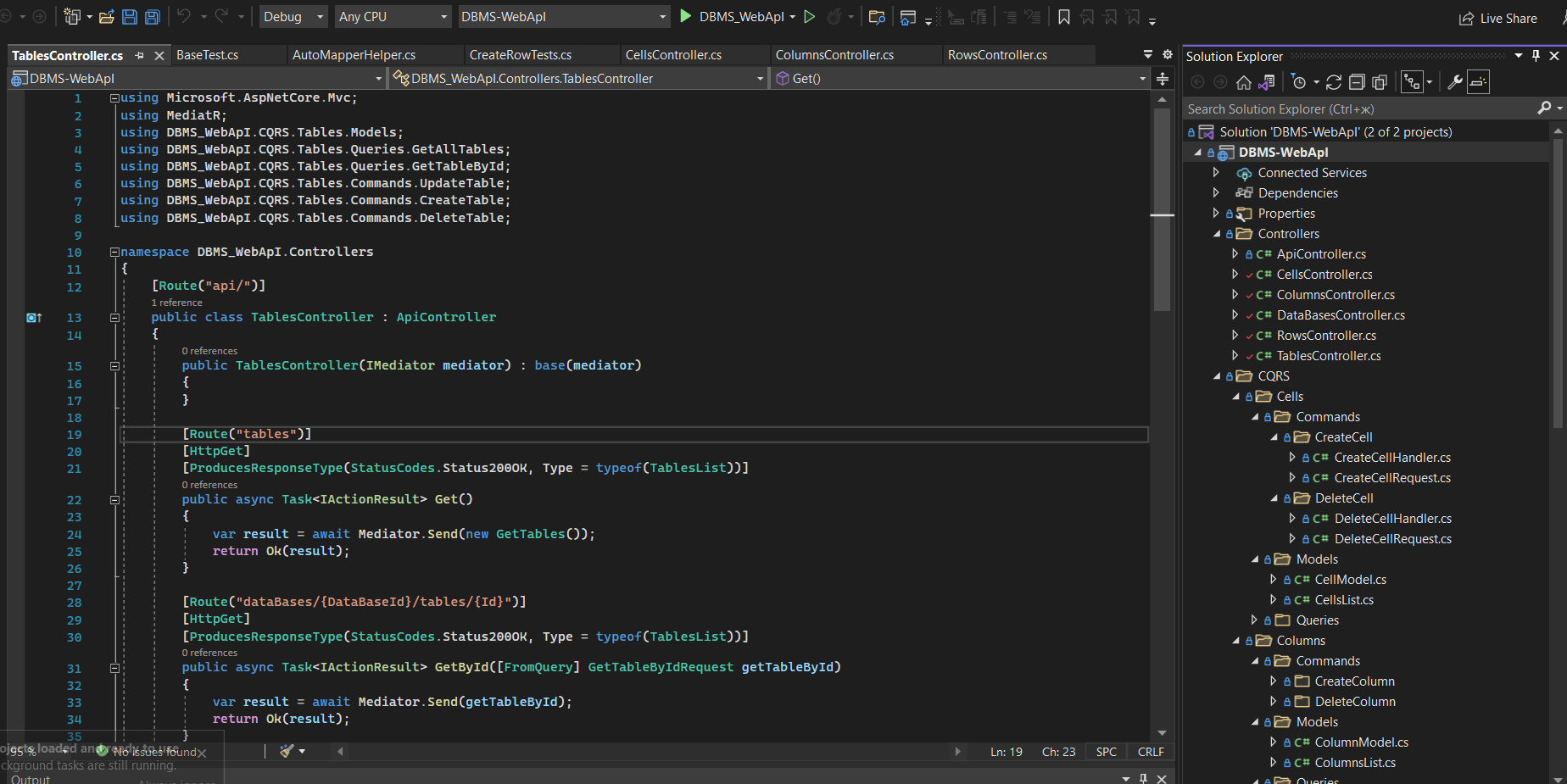
Створимо проект за шаблоном ASP.NET Web API та напишемо відповідні REST API методи для GET, POST, PUT та DELETE-запитів в API-контролерах. Також в роботі дотриманий шаблон програмування CQRS. В якості OpenAPI Specification використано Swagger/Swashbuckle. В якості СУБД візьмемо MS SQL Server.

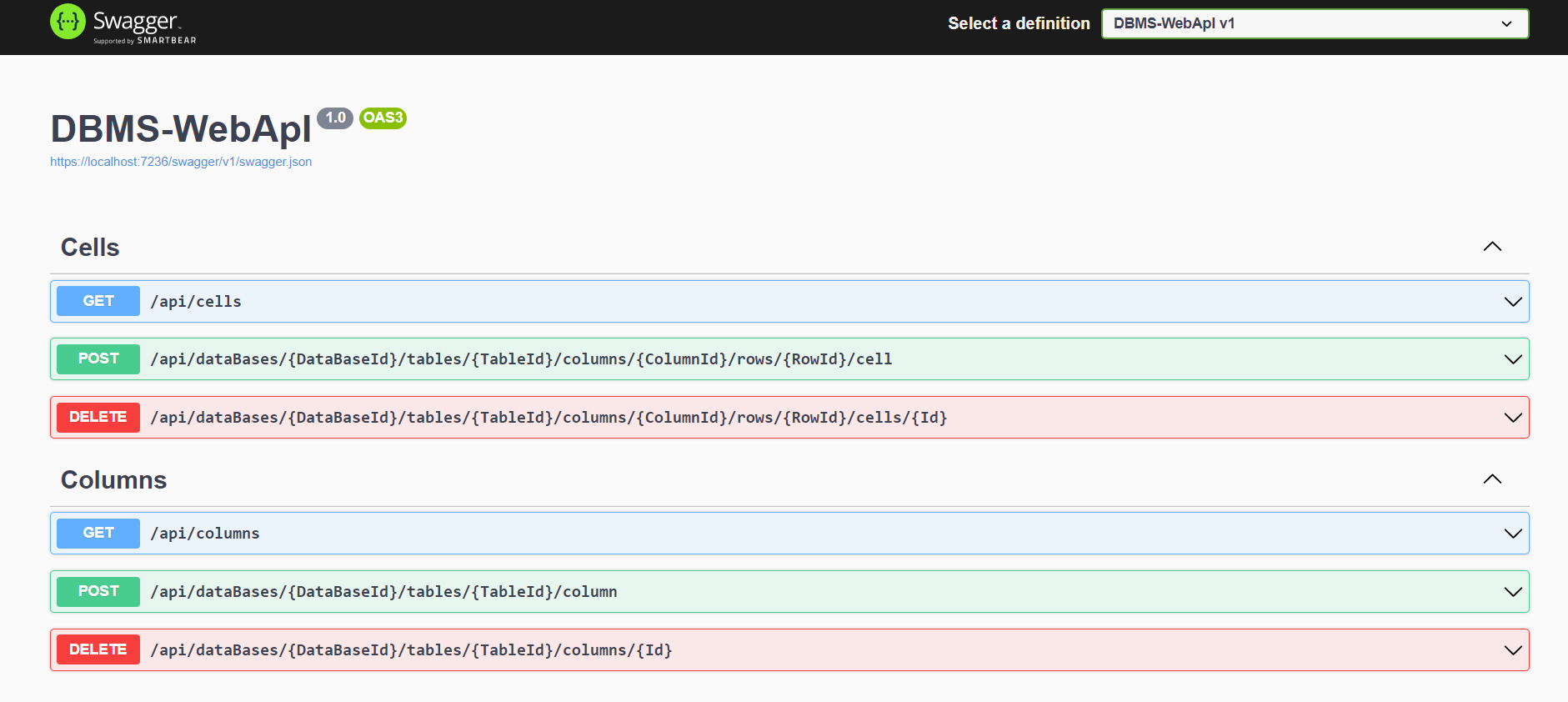
Діаграма таблиць і їх зв’язків у MS SQL Server:



Підключення до БД:







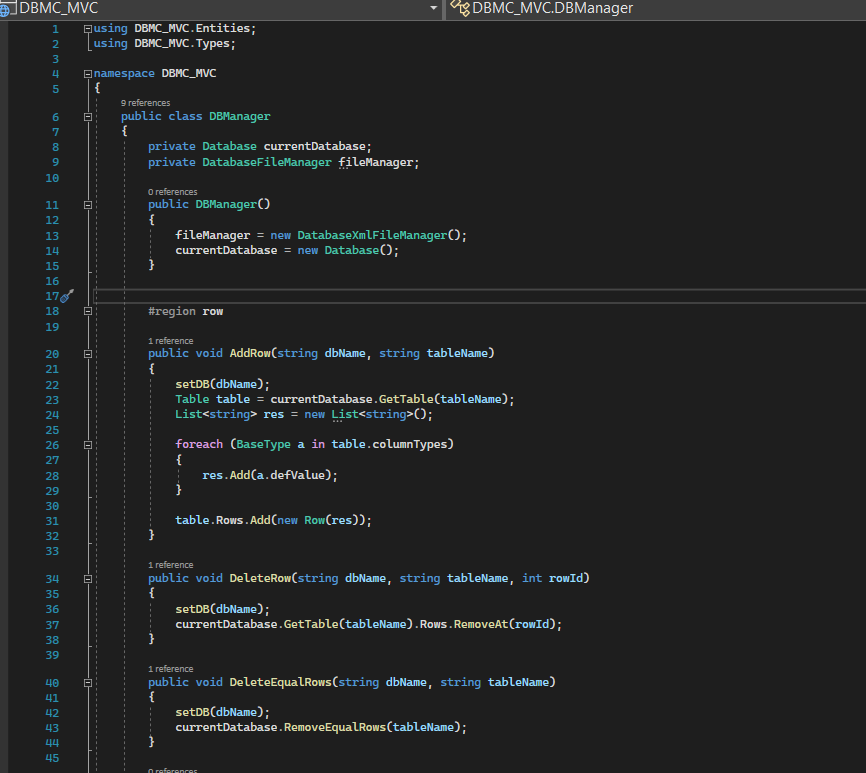
Приклад запиту GET:



**19) Web-проект (ASP.NET MVC)**

В рамках даного шаблону потрібно забезпечити класи моделей, класи-контролери та представлення для веб-сторінок. Для виконання CRUD-операцій над цими сутностями нам знадобляться контролери

**DatabaseManager** – клас для керування всіма операціями над базою даних, реалізований за патерном проектування *Singleton*:



Вигляд інтерфейсу користувача:

