

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»  
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
**Кафедра системного програмування та спеціалізованих  
комп'ютерних систем**

**Лабораторна робота №2**  
з дисципліни

**«Бази даних і засоби управління»**

Тема: «Створення додатку бази даних, орієнтованого на  
взаємодію з СУБД PostgreSQL»

Виконав: студент III курсу

ФПМ групи КВ-82

Любич І. Д.

Перевірив: Павловський В. І.

*Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.*

*Загальне завдання роботи полягає у наступному:*

1. Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
3. Забезпечити реалізацію пошуку за декількома атрибутами з 2-х та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

*Деталізоване завдання:*

1. Забезпечити можливість введення/редагування/вилучення даних у таблицях бази даних з можливістю контролю відповідності типів даних атрибутів таблиць (рядків, чисел, дати/часу). Для контролю пропонується два варіанти:

а) контроль при введенні - валідація даних;

б) перехоплення помилок (**try...except**) від сервера PostgreSQL при виконанні відповідної команди SQL.

Особливу увагу варто звернути на дані таблиць, що мають зв'язок 1:N.

З боку батьківської таблиці необхідно контролювати **вилучення (ON DELETE)** рядків за умови наявності даних у підлеглий таблиці.

З боку підлеглої таблиці варто контролювати наявність відповідного рядка у батьківській таблиці при виконанні **внесення** до неї нових даних.

Унеможливити виведення програмою на екрані системних помилок PostgreSQL шляхом їх перехоплення і адекватної обробки.

Внесення даних виконується користувачем у консольному вікні програми.

2. Забезпечити можливість автоматичної генерації великої кількості даних у таблицях за допомогою вбудованих у PostgreSQL функцій роботи з псевдовипадковими числами. Дані мають бути згенерованими **не програмою, а відповідним SQL-запитом!** Кількість даних для генерування має вводити користувач з клавіатури.

3. Для реалізації багатокритеріального пошуку необхідно підготувати 3 запити, що включають дані з декількох таблиць і фільтрують рядки за 3-4 атрибутами цих таблиць. Забезпечити можливість введення конкретних значень констант для фільтрації з клавіатури користувачем. Після

виведення даних вивести час виконання запиту у мілісекундах. Перевірити швидкість роботи запитів на попередньо згенерованих даних.

4. Програмний код організувати згідно шаблону Model-View-Controller (MVC). Приклад організації коду згідно шаблону доступний [за даним посиланням](#). Модель, подання (представлення) та контролер мають бути реалізовані у окремих файлах. Для доступу до бази даних використовувати **лише мову SQL** (без ORM).

### Нормалізована модель даних

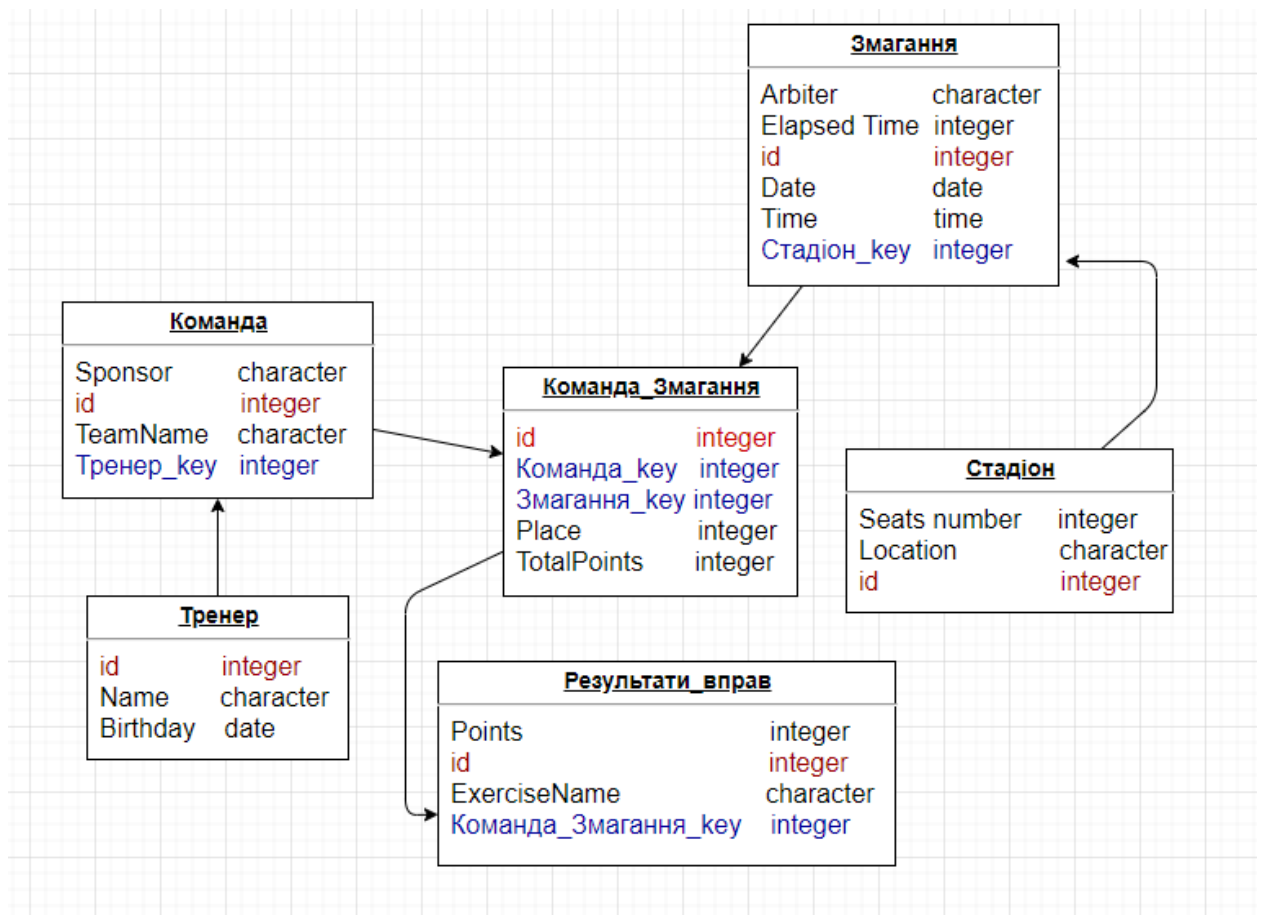


Рис. 1 – нормалізована модель БД

Зміни у порівнянні з першою лабораторною роботою:

1. Змінений зв'язок між сутностями Стадіон і Змагання з N:M до 1:N.
2. Атрибут Date and Time таблиці Змагання перероблено на 2 сутності.
3. Два зовнішні ключі таблиці Результати вправ (Команди та Змагання) замінено на один – таблиці Команда\_Змагання.

## Середовище розробки та налаштування підключення до бази даних

Для виконання лабораторної роботи використовувалась мова програмування C# та IDE Visual Studio 2019.

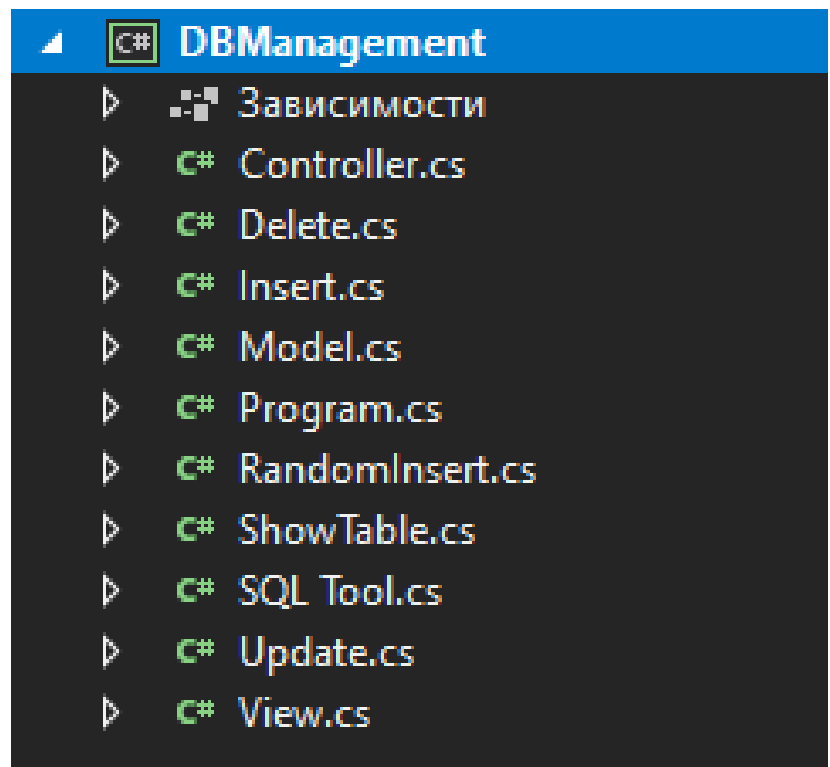
Для підключення до серверу бази даних PostgreSQL використовувався пакет Npgsql. Для цього створений метод, який з'єднується з базою даних.

```
private void connectDB()
{
    var cs = $"Host=localhost;Username=postgres;Password={password};Database=DB_Lab1";
    con = new NpgsqlConnection(cs);
    con.Open();
    cmd = new NpgsqlCommand();
    cmd.Connection = con;
}
```

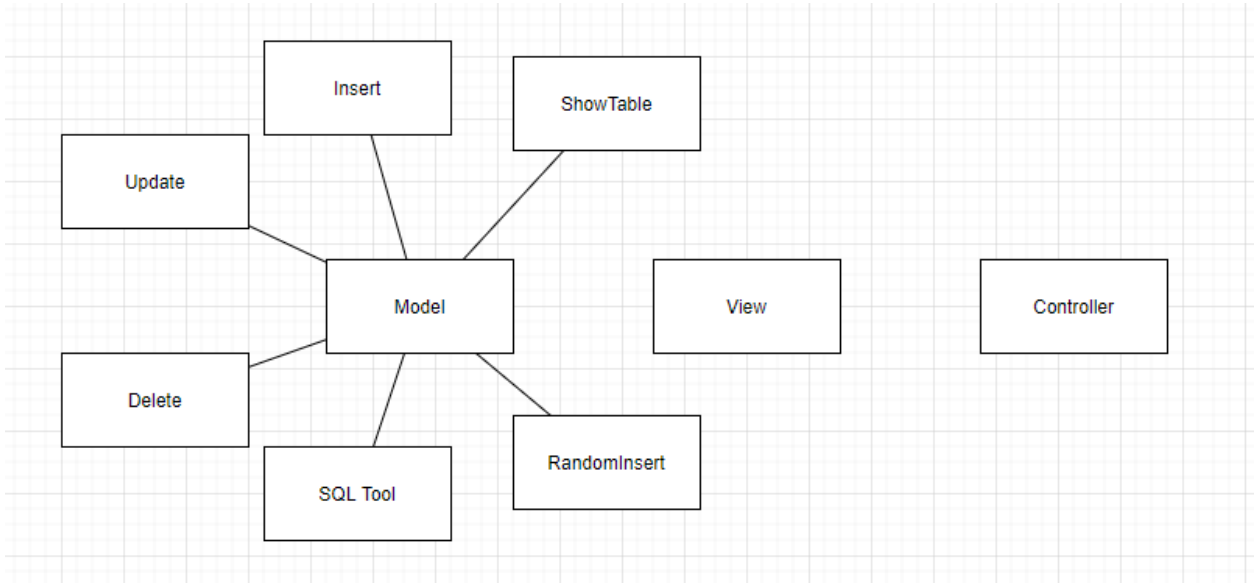
Надалі для керування БД відбуватиметься за допомогою змінної “cmd”.

## Опис структури програми

Програма містить 10 модулів, серед яких основні: **Program** (початок програми), **Model**, **View**, **Controller** – відповідають за дані, інтерфейс та обробку інформації відповідно. Інші 6 модулів виконують лише одну визначену задачу: ShowTable – вивід таблиць, Insert – внесення даних, Update – оновлення даних, Delete – видалення записів з таблиць, SQL Tool – параметризовані запити до БД, RandomInsert – рандомізоване заповнення таблиці «Тренер».



## Структура MVC у файлах



## Структура меню програми

### Головне меню

```
1. View Db content
2. Insert
3. Update
4. Delete
5. SQL tool
6. Random insert to "Тренер"
7. Exit
```

### Меню вибору таблиці

```
1. Команда
2. Тренер
3. Стадіон
4. Змагання
5. Результати_вправ
6. Команда_Змагання
0. Back
```

### Меню вибору запиту

```
1. Пошук команд та їх тренерів за назвою команди, спонсором та датою народження тренера команди
2. Пошук стадіонів та змагань за кількістю місць, на яких заплановані змагання у заданий проміжок дати та часу
3. Пошук команд і змагань за загальною кількістю очок, тривалістю змагань і арбітром
0. Back
```

### Меню вибору кількості рандомізованих рядків

```
Number of randomized records:
```

```
_
```

## Відповіді на деталізовані завдання

1. Ілюстрації обробки виняткових ситуацій (помилки) при введенні/вилученні та валідації даних.

Обробка виняткових ситуацій при введенні (insert) та вилученні даних (delete) виконується за допомогою блоку try-catch. При введенні помилкових даних Команда SQL буде проігнорована, а в меню користувача з'явиться відповідне повідомлення.

```
public void ExecuteQuery(NpgsqlCommand _cmd)
{
    try
    {
        _cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Помилка перехоплена");
    }
}
```

Рис. 2 - Обробка помилок з сервера SQL

Приклад додавання рядка з неіснуючим зовнішнім ключем:

```
1. Команда
2. Тренер
3. Стадіон
4. Змагання
5. Результати_вправ
6. Команда_Змагання
0. Back
1
TeamName: T1
Sponsor: Sp1
Coach ID: 100500
Помилка перехоплена
```

Приклад видалення рядка, ключ якого є зовнішнім ключем іншої таблиці:

```
1. Команда
2. Тренер
3. Стадіон
4. Змагання
5. Результати_вправ
6. Команда_Змагання
0. Back
2
Record ID to delete: 2
Помилка перехоплена
```

Приклад уведення рядка з полем, тип якого не відповідає дійсному:

```
1. Команда
2. Тренер
3. Стадіон
4. Змагання
5. Результати_вправ
6. Команда_Змагання
0. Back
2
Name: Jack
Birthday:
Day: 32
Month: 5
Year: 1991
Помилка перехоплена
```

2. Копії екрану з фрагментами згенерованих даних таблиць (таблиця Тренер)

Data Output				Explain	Messages	Notifications
	id [PK] integer		Name character varying		Birthday date	
1	0	Ivan Gromov	1979-05-10			
2	1	Denis Denisov	1977-06-20			
3	2	Alexander Ivanyuk	1965-06-20			
4	3	Vlad Nosko	1977-07-30			
5	4	ibs	2001-12-20			
6	5	123	0232-02-02			

Таблиця «Тренер» до генерування.

```
Number of randomized records:
32
```

Уведення кількості рядків для генерування.

	<div>id</div> <div>[PK] integer</div>	<div>Name</div> <div>character varying</div>	<div>Birthday</div> <div>date</div>				
1	0	Ivan Gromov	1979-05-10				
2	1	Denis Denisov	1977-06-20				
3	2	Alexander Ivanyuk	1965-06-20				
4	3	Vlad Nosko	1977-07-30				
5	4	ibs	2001-12-20				
6	5	123	0232-02-02				
7	60	nopqrstuvw	2014-01-12				
8	61	ghijklmnop	2014-01-10				
9	62	WXYZabcdef	2014-01-19	24	77	BCDEFGHIJK	2014-01-17
10	63	nopqrstuvw	2014-01-11	25	78	JKLMNOPQRS	2014-01-17
11	64	QRSTUVWXYZ	2014-01-17	26	79	FGHIJKLMNO	2014-01-18
12	65	z	2014-01-17	27	80	WXYZabcdef	2014-01-11
13	66	stuvwxyz	2014-01-10	28	81	FGHIJKLMNO	2014-01-17
14	67	TUVWXYZabc	2014-01-18	29	82	stuvwxyz	2014-01-15
15	68	fghijklmno	2014-01-14	30	83	EFGHIJKLMN	2014-01-11
16	69	DEFGHIJKLM	2014-01-10	31	84	NOPQRSTUVWXYZ	2014-01-13
17	70	wxyz	2014-01-11	32	85	RSTUVWXYZa	2014-01-18
18	71	abcdefghij	2014-01-12	33	86	VWXYZabcde	2014-01-15
19	72	UVWXYZabcd	2014-01-19	34	87	ijklmnopqrs	2014-01-17
20	73	z	2014-01-10	35	88	vWXYZ	2014-01-10
21	74	CDEFGHIJKL	2014-01-14	36	89	defghijklm	2014-01-11
22	75	pqrstuvwxy	2014-01-19	37	90	stuvwxyz	2014-01-14
23	76	XYZabcdefg	2014-01-18	38	91	WXYZabcdef	2014-01-17
24	77	BCDEFGHIJK	2014-01-17				

Таблиця «Тренер» після генерування.



### 3. Ілюстрації введення пошукових запитів та результати їх виконання.

1. Пошук команд та їх тренерів за назвою команди, спонсором та датою народження тренера команди
2. Пошук стадіонів та змагань за кількістю місць, на яких заплановані змагання у заданий проміжок дати та часу
3. Пошук команд і змагань за загальною кількістю очок, тривалістю змагань і арбітром
0. Back

```
Пошук команд та їх тренерів за назвою команди, спонсором та датою народження тренера команди
TeamName like: %
Sponsor like: %
Проміжок дати народження
Нижня границя
Day: 03
Month: 03
Year: 1964
Верхня границя
Day: 03
Month: 03
Year: 2012
Team = TeamOne, Sponsor = Sponsor1, CoachName = Denis Denisov, CoachBirthday = 1977-06-20
Team = TeamTwo, Sponsor = Sponsor2, CoachName = Alexander Ivanyuk, CoachBirthday = 1965-06-20
Team = TeamThree, Sponsor = Sponsor3, CoachName = Vlad Nosko, CoachBirthday = 1977-07-30
Execution time: 3
```

```
Пошук стадіонів та змагань за кількістю місць, на яких заплановані змагання у заданий проміжок дати та часу
Мінімальна кількість місць: 10
Максимальна кількість місць: 11212
Проміжок дати
Нижня границя
Day: 1
Month: 1
Year: 1999
Верхня границя
Day: 5
Month: 5
Year: 2055
Мінімальний час початку: 2:30
Максимальний час початку: 23:43
id = 2, Seats number = 200, Date = 2020-10-10, Time = 15:40:00
id = 3, Seats number = 400, Date = 2020-12-11, Time = 18:00:00
id = 2, Seats number = 200, Date = 2005-04-23, Time = 23:40:00
id = 3, Seats number = 400, Date = 2020-12-20, Time = 20:30:00
Execution time: 2
```

```
Пошук команд і змагань за загальною кількістю очок, тривалістю змагань і арбітром
Мінімальна кількість очок: 110
Максимальна кількість очок: 160
Мінімальна тривалість змагання: 20
Максимальна тривалість змагання: 95
Arbiter like: %
Team = TeamOne, Competition ID = 72, Total Points = 111, Duration = 60, Arbiter = Arbiter1
Team = TeamOne, Competition ID = 73, Total Points = 155, Duration = 90, Arbiter = Arbiter2
Team = TeamTwo, Competition ID = 73, Total Points = 143, Duration = 90, Arbiter = Arbiter2
Team = TeamThree, Competition ID = 73, Total Points = 123, Duration = 90, Arbiter = Arbiter2
Execution time: 1
```

# 4. Ілюстрації програмного коду на Github.

Search or jump to...

[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)

Unwatch

Star

Fork

A1xaT / Database

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

master1 branch0 tags

Go to file

Add file

Code

AtxaT Lab 2#870b571 minute ago9 commits

Lab1Lab1 report27 days ago

Lab2Lab 21 minute ago

README.mdUpdate README.md2 months ago

README.md

### Database

Database & Management tools labs

Liubchych Ilyia KV-82

About

Database and management system labs

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Search or jump to...

[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)

Unwatch

Star

Fork

A1xaT / Database

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

masterDatabase / Lab2 /

Go to file

Add file

AtxaT Lab 2#5a86891 minute agoHistory

..

Controller.csLab 21 minute ago

Delete.csLab 23 hours ago

Insert.csLab 23 hours ago

Model.csLab 23 hours ago

Program.csLab 23 hours ago

README.mdLab 21 minute ago

RandomInsert.csLab 23 hours ago

SQL Tools.csLab 23 hours ago

ShowTable.csLab 23 hours ago

Update.csLab 23 hours ago

Views.csLab 23 hours ago

README.md

Лабораторна робота №2 Тема: Створення додатку бази даних, орієнтованого на взаємодію з СУБД PostgreSQL Структура бази даних:

Зображення