# Лабораторна робота 5: Знайомство з ORM на прикладі Entity Framework Core

### **Мета роботи:**

Навчитися використовувати EF Core для взаємодії з базою даних у C# проєктах.

### **Завдання:**

1. **Налаштувати базу даних**

На цьому етапі необхідно обрати базу даних (postgresql, mssql, mysql, sqlite) та встановити її.
Вийнятком є база даних SQLite яка не потребує встановлення та виконується у контексті поточного процесу.

Одним з способів встановсення баз даних з відкритим кодом (postgresql) є docker container.

docker pull postgres

docker run --name db -e POSTGRES\_PASSWORD=mysecretpassword -d -p 5432:5432 postgres

Щоб підключитися до PostgreSQL, можна використати будь-який клієнт, наприклад **psql** або інший інструмент (наприклад, PgAdmin або DBeaver). Visual Studio мітить вбудованих клієнт “Server Explorer” який може відображати дані з Postgres бази після встановлення відповідного драйверу (<https://odbc.postgresql.org/>)

Офіційний клієнт: https://www.pgadmin.org/

1. **Відкрийте проект WebAPI створений у 2й лабораторній роботі**
2. **Модифікувати модель даних** (додати поле з типом дати)

public class Student

{

 public int Id { get; set; }

 public string Name { get; set; }

 public DateTime DateOfBirth { get; set; }

 public string Email { get; set; }

}

1. **Створення контексту бази даних**

Створити клас контексту, який буде представляти базу даних

public class SchoolContext : DbContext

{

 public DbSet<Student> Students { get; set; }

 protected override void OnConfiguring(

DbContextOptionsBuilder optionsBuilder)

 {

 optionsBuilder.UseSqlServer(@"<connection string here>");

 }

}

1. **Міграції та створення бази даних**

Використовуючи команду **Add-Migration**, створити міграцію.

Застосувати міграцію для створення бази даних за допомогою команди **Update-Database**.

1. **Реалізувати необхідні операції з даними (CRUD)**

**Приклад додавання записів**:

using (var context = new SchoolContext())

{

 var student = new Student

 {

Name = "John Doe",

DateOfBirth = new DateTime(2000, 1, 1),

Email = "john@example.com"

 };

 context.Students.Add(student);

 context.SaveChanges();

}

**Приклад читання записів**:

using (var context = new SchoolContext())

{

 var students = context.Students.ToList();

 foreach (var student in students)

 {

 Console.WriteLine($"{student.Name} - {student.Email}");

 }

}

1. **Перевірити разом з UI створеним у попередній роботі**