

# Лабораторна робота № 1

**Тема:** Огляд технологій доступу до даних.

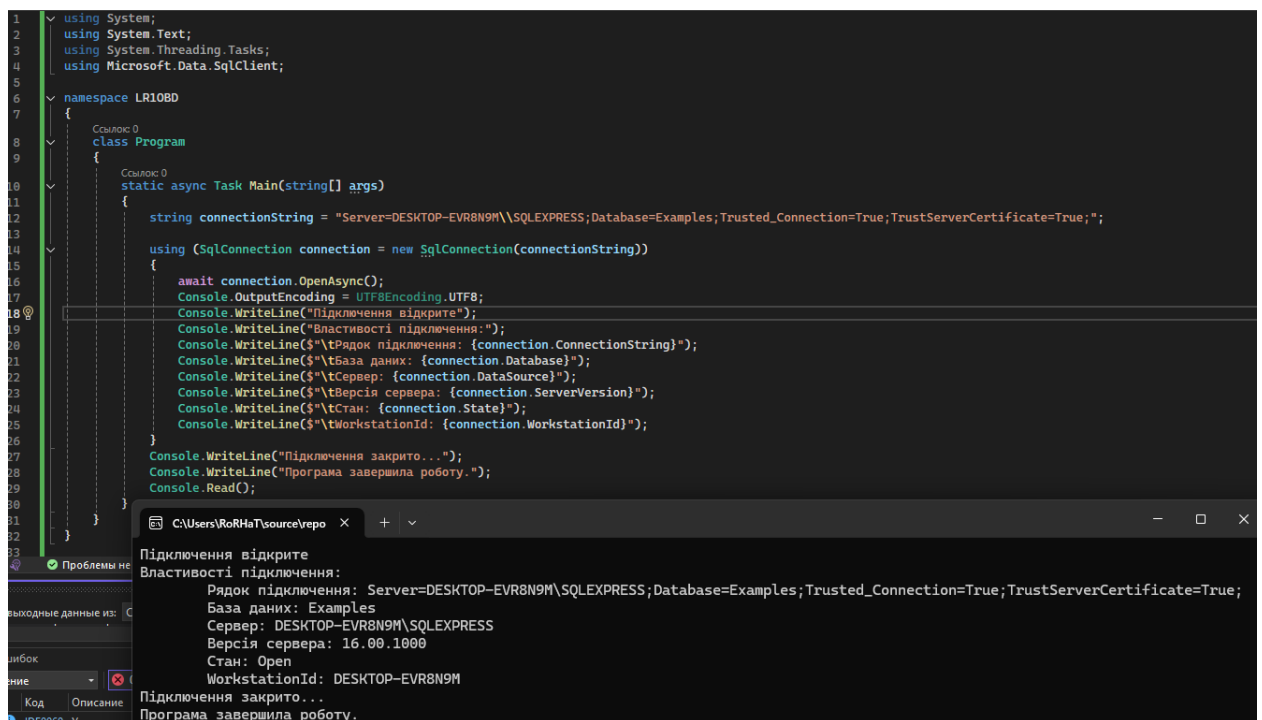
**Мета:** отримати практичні навички з використання технологій доступу до даних.

## Практична частина

1. Виконати практичні приклади які подані у практичній частині, створення, підключення до БД, створення таблиць, внесення даних, відповідно до індивідуального варіанту завдання.
2. Оформити звіт, де надати скріншоти роботи програми та відповідно код програми і створеної БД.
3. **Додатково:** Створити зв'язки між таблицями.

### Завдання 1-2:

На рисунку 1.1 перевіряємо підключення до БД.



```
1 using System;
2 using System.Text;
3 using System.Threading.Tasks;
4 using Microsoft.Data.SqlClient;
5
6 namespace LR10B0
7 {
8     class Program
9     {
10         static async Task Main(string[] args)
11         {
12             string connectionString = "Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
13
14             using (SqlConnection connection = new SqlConnection(connectionString))
15             {
16                 await connection.OpenAsync();
17                 Console.OutputEncoding = UTF8Encoding.UTF8;
18                 Console.WriteLine("Підключення відкрите");
19                 Console.WriteLine("Властивості підключення:");
20                 Console.WriteLine($"Рядок підключення: {connection.ConnectionString}");
21                 Console.WriteLine($"База даних: {connection.Database}");
22                 Console.WriteLine($"Сервер: {connection.DataSource}");
23                 Console.WriteLine($"Версія сервера: {connection.ServerVersion}");
24                 Console.WriteLine($"Стан: {connection.State}");
25                 Console.WriteLine($"WorkstationId: {connection.WorkstationId}");
26             }
27             Console.WriteLine("Підключення закрито...");
28             Console.WriteLine("Програма завершила роботу.");
29             Console.Read();
30         }
31     }
32 }
```

Підключення відкрите  
Властивості підключення:  
Рядок підключення: Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted\_Connection=True;TrustServerCertificate=True;  
База даних: Examples  
Сервер: DESKTOP-EVR8N9M\\SQLEXPRESS  
Версія сервера: 16.00.1000  
Стан: Open  
WorkstationId: DESKTOP-EVR8N9M  
Підключення закрито...  
Програма завершила роботу.

Рисунок 1.1 – Підключення до бази даних

На рисунку 1.2 створюємо базу даних на сервері.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Microsoft.Data.SqlClient;
7
8 namespace ConsoleApp32
9 {
10     internal class Program
11     {
12         static async Task Main(string[] args)
13         {
14             string connectionString = "Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 await connection.OpenAsync(); // Відкриваємо підключення
19
20                 SqlCommand command = new SqlCommand();
21                 command.CommandText = "CREATE DATABASE Examples;"; // визначасмо виконувану команду (створення БД)
22                 command.Connection = connection; // визначасмо використане підключення
23                 await command.ExecuteNonQueryAsync(); // виконуємо команду
24                 Console.WriteLine("База даних створена");
25             }
26
27             Console.Read();
28         }
29     }
30 }
```

Рисунок 1.2 – Створення бази даних

На рисунку 1.3 створюємо таблицю до бази даних.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Microsoft.Data.SqlClient;
7
8 namespace LR10BD
9 {
10     internal class Program
11     {
12         static async Task Main(string[] args)
13         {
14             string connectionString = "Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 await connection.OpenAsync(); // Відкриваємо підключення
19
20                 SqlCommand command = new SqlCommand();
21                 command.CommandText = "CREATE TABLE Users (Id INT PRIMARY KEY IDENTITY, Age INT NOT NULL, Name NVARCHAR(100) NOT NULL);"; // визначасмо виконувану команду (створення таблиці)
22                 command.Connection = connection; // визначасмо використане підключення
23                 await command.ExecuteNonQueryAsync(); // виконуємо команду
24                 Console.WriteLine("Таблиця Users створена");
25             }
26
27             Console.Read();
28         }
29     }
30 }
```

Рисунок 1.3 – Створення таблиць до бази даних

На рисунку 1.4 додаємо тестові дані до таблиці.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Microsoft.Data.SqlClient;
7
8 namespace LR10BD
9 {
10     internal class Program
11     {
12         static async Task Main(string[] args)
13         {
14             string connectionString = "Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
15             string sqlExpression = "INSERT INTO Users (Name, Age) VALUES ('Denis', 20);"; // Створення локальної змінної, якій передано запит
16
17             using (SqlConnection connection = new SqlConnection(connectionString))
18             {
19                 await connection.OpenAsync(); // Відкриваємо підключення
20
21                 SqlCommand command = new SqlCommand(sqlExpression, connection);
22                 int number = await command.ExecuteNonQueryAsync(); // Виконує SQL-вираз і повертає кількість змінених записів
23                 Console.WriteLine($"Додано об'єктів: {number}");
24             }
25
26             Console.Read();
27         }
28     }
29 }
```

Рисунок 1.4 – Додання даних до таблиць

### Завдання 3:

На рисунку 1.5 створюємо другу таблицю та робимо зв'язки між таблицями.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using Microsoft.Data.SqlClient;
7
8 namespace LR10BD
9 {
10     internal class Program
11     {
12         static async Task Main(string[] args)
13         {
14             string connectionString = "Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
15
16             using (SqlConnection connection = new SqlConnection(connectionString))
17             {
18                 await connection.OpenAsync(); // Відкриваємо підключення
19
20                 // Створення таблиці Orders
21                 string createOrdersTable = "CREATE TABLE Orders (OrderId INT PRIMARY KEY IDENTITY, OrderDate DATETIME NOT NULL, UserId INT, FOREIGN KEY (UserId) REFERENCES Users(Id));";
22                 SqlCommand createOrdersCommand = new SqlCommand(createOrdersTable, connection);
23                 await createOrdersCommand.ExecuteNonQueryAsync();
24                 Console.WriteLine("Таблиця Orders створена");
25
26                 // Вставка даних в таблицю Orders
27                 string insertOrder = "INSERT INTO Orders (OrderDate, UserId) VALUES (GETDATE(), 1);";
28                 SqlCommand insertOrderCommand = new SqlCommand(insertOrder, connection);
29                 int orderNumber = await insertOrderCommand.ExecuteNonQueryAsync();
30                 Console.WriteLine($"Додано об'єктів в Orders: {orderNumber}");
31
32             }
33
34             Console.Read();
35         }
36     }
37 }
```

Рисунок 1.5 – Створення зв'язків між таблицями

Створена протягом виконання лабораторної роботи база даних має наступний вигляд, зображений на рисунку 1.6.

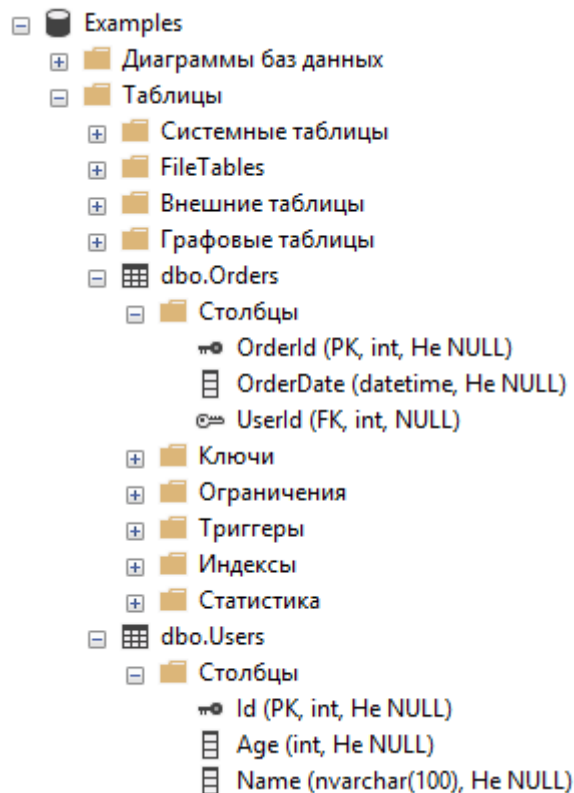


Рисунок 1.6 – Створена база даних Examples

## **Контрольні питання:**

### **1. Що таке технологія доступу до даних ADO.NET?**

ADO.NET (ActiveX Data Objects .NET) — це технологія, яка входить до складу платформи .NET і призначена для доступу до джерел даних (наприклад, баз даних SQL Server, Oracle, Excel, XML тощо).

### **2. Перерахуйте базові класи ADO.NET. Яке призначення кожного із них?**

Основні базові класи ADO.NET та їх призначення:

- `SqlConnection` – встановлює і управляє з'єднанням з базою даних SQL Server.
- `SqlCommand` – використовується для виконання SQL-запитів і команд (наприклад, SELECT, INSERT, UPDATE, DELETE).
- `SqlDataReader` – забезпечує швидке зчитування даних із бази у підключеному режимі (тільки вперед).
- `SqlDataAdapter` – слугує містком між `DataSet` і базою даних, дозволяє заповнювати об'єкти даних (наприклад, `DataSet`) і зберігати зміни назад у базу.
- `DataSet` – зберігає дані у пам'яті у відключеному режимі. Може містити декілька таблиць, зв'язків і обмежень.
- `DataTable` – представляє окрему таблицю у `DataSet`, яка може бути незалежною.
- `DataRow`,  `DataColumn` – представляють рядки і стовпці таблиці відповідно.

### **3. Опишіть призначення простору імен Microsoft.Data.SqlClient.**

Простір імен `Microsoft.Data.SqlClient` містить набір класів, які забезпечують доступ до бази даних Microsoft SQL Server із застосунків .NET.