

## Лабораторна робота № 2

**Тема:** SqlDataReader та DataSet.

**Мета:** отримати практичні навички з використання методів класу SqlDataReader та DataSet.

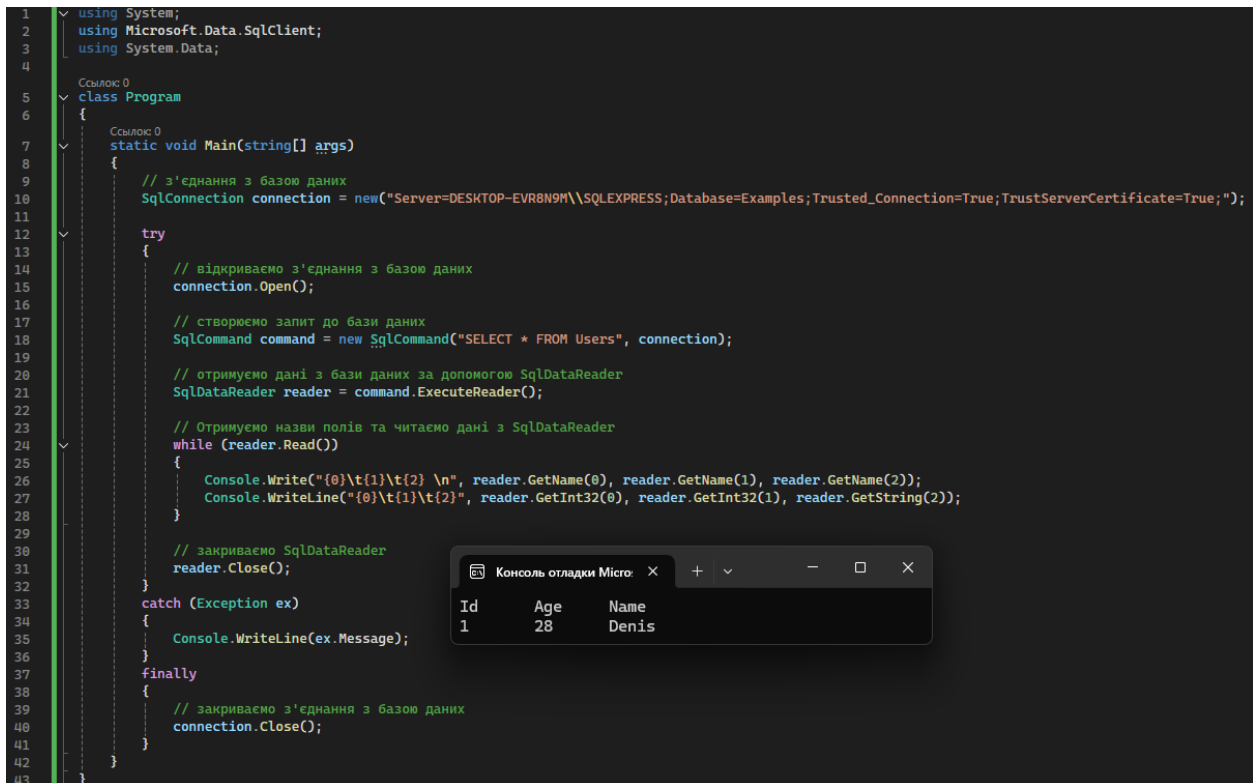
### Практична частина

Використовуючи базу даних з першої лабораторної роботи:

1. Створити програму, яка зчитує дані з бази даних MS SQL Server за допомогою об'єкту SqlDataReader та виводить їх на екран.
2. Завантажити дані з бази даних в DataSet: створіть програму, яка підключається до бази даних, використовуючи ADO.NET та завантажує дані з таблиці в DataSet. Потім виведіть ці дані на екран.
3. Змінити дані у DataSet: створіть програму, яка завантажує дані з бази даних в DataSet, дозволяє користувачу вносити зміни до даних та зберігати їх назад у базу даних.

### Завдання 1:

На рисунку 2.1 перевіряємо зчитування даних з бази даних до БД.



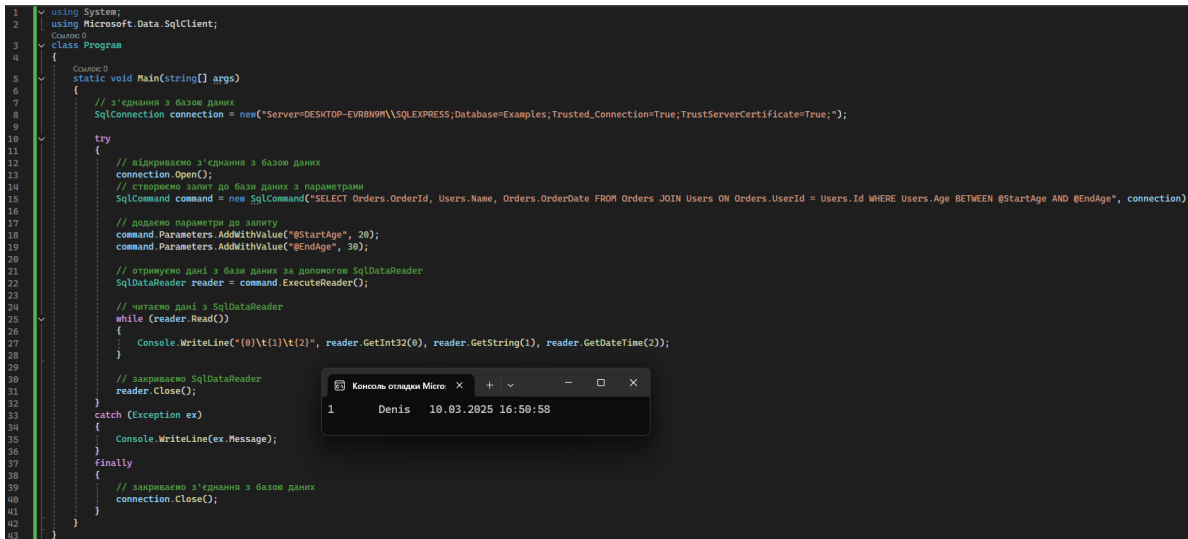
```
1 using System;
2 using Microsoft.Data.SqlClient;
3 using System.Data;
4
5 class Program
6 {
7     static void Main(string[] args)
8     {
9         // з'єднання з базою даних
10        SqlConnection connection = new SqlConnection("Server=DESKTOP-EVR8N9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;");
11
12        try
13        {
14            // відкриваємо з'єднання з базою даних
15            connection.Open();
16
17            // створюємо запит до бази даних
18            SqlCommand command = new SqlCommand("SELECT * FROM Users", connection);
19
20            // отримуємо дані з бази даних за допомогою SqlDataReader
21            SqlDataReader reader = command.ExecuteReader();
22
23            // Отримуємо назви полів та читаємо дані з SqlDataReader
24            while (reader.Read())
25            {
26                Console.WriteLine("{0}\t{1}\t{2} \n", reader.GetName(0), reader.GetName(1), reader.GetName(2));
27                Console.WriteLine("{0}\t{1}\t{2}", reader.GetInt32(0), reader.GetInt32(1), reader.GetString(2));
28            }
29
30            // закриваємо SqlDataReader
31            reader.Close();
32
33        } catch (Exception ex)
34        {
35            Console.WriteLine(ex.Message);
36        }
37        finally
38        {
39            // закриваємо з'єднання з базою даних
40            connection.Close();
41        }
42    }
43 }
```

Id	Age	Name
1	28	Denis

Рисунок 2.1 – Зчитування даних з БД

## Завдання 2:

На рисунку 2.2 підключення та завантаження даних БД.

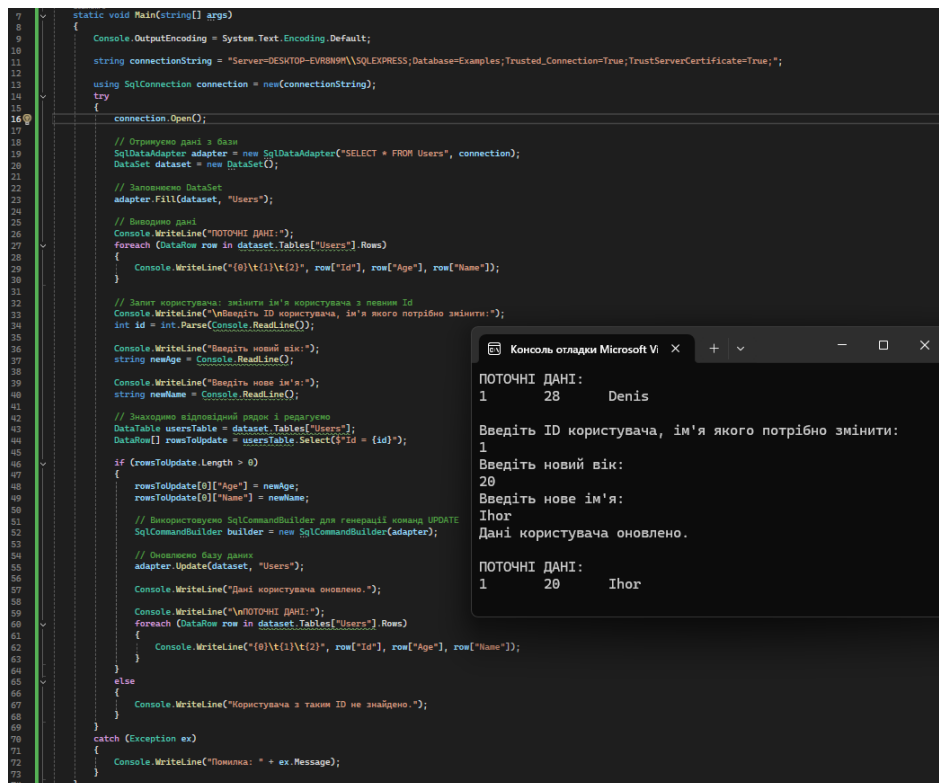


```
1 using System;
2 using Microsoft.Data.SqlClient;
3
4 namespace Program
5 {
6     static void Main(string[] args)
7     {
8         // з'єднання з базов даних
9         SqlConnection connection = new("Server=DESKTOP-EVRBN9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;");
10
11         try
12         {
13             // відкриваємо з'єднання з базов даних
14             connection.Open();
15             // створимо запит до бази даних з параметрами
16             SqlCommand command = new SqlCommand("SELECT Orders.OrderId, Users.Name, Orders.OrderDate FROM Orders JOIN Users ON Orders.UserId = Users.Id WHERE Users.Age BETWEEN @StartAge AND @EndAge", connection);
17             // додаємо параметри до запиту
18             command.Parameters.AddWithValue("@StartAge", 20);
19             command.Parameters.AddWithValue("@EndAge", 30);
20
21             // отримемо дані з бази даних за допомогою SqlDataReader
22             SqlDataReader reader = command.ExecuteReader();
23             // читаємо дані з SqlDataReader
24             while (reader.Read())
25             {
26                 Console.WriteLine("{0}\t{1}\t{2}", reader.GetInt32(0), reader.GetString(1), reader.GetDateTime(2));
27             }
28             // закриваємо SqlDataReader
29             reader.Close();
30         }
31         catch (Exception ex)
32         {
33             Console.WriteLine(ex.Message);
34         }
35         finally
36         {
37             // закриваємо з'єднання з базов даних
38             connection.Close();
39         }
40     }
41 }
```

Рисунок 2.2 – Підключення до бази даних

## Завдання 3:

На рисунку 2.3 створена програма, яка завантажує дані з бази даних в DataSet, дозволяє користувачу вносити зміни до даних та зберігати їх назад у базу даних.



```
1 static void Main(string[] args)
2 {
3     Console.OutputEncoding = System.Text.Encoding.Default;
4
5     string connectionString = "Server=DESKTOP-EVRBN9M\\SQLEXPRESS;Database=Examples;Trusted_Connection=True;TrustServerCertificate=True;";
6
7     using SqlConnection connection = new(connectionString);
8
9     try
10     {
11         connection.Open();
12
13         // Отримуємо дані з бази
14         SqlDataAdapter adapter = new SqlDataAdapter("SELECT * FROM Users", connection);
15         DataSet dataset = new DataSet();
16
17         // Заповнюємо DataSet
18         adapter.Fill(dataset, "Users");
19
20         // Виводимо дані
21         Console.WriteLine("ПОТОЧНІ ДАНІ:");
22         foreach (DataRow row in dataset.Tables["Users"].Rows)
23         {
24             Console.WriteLine("{0}\t{1}\t{2}", row["Id"], row["Age"], row["Name"]);
25         }
26
27         // Запит користувача: змінити ім'я користувача з певним ID
28         Console.WriteLine("\nВведіть ID користувача, ім'я якого потрібно змінити:");
29         int id = int.Parse(Console.ReadLine());
30
31         Console.WriteLine("Введіть новий вік:");
32         string newAge = Console.ReadLine();
33
34         Console.WriteLine("Введіть нове ім'я:");
35         string newName = Console.ReadLine();
36
37         // Знаходимо відповідний рядок у DataSet
38         DataTable usersTable = dataset.Tables["Users"];
39         DataRow[] rowsToUpdate = usersTable.Select($"Id = {id}");
40
41         if (rowsToUpdate.Length > 0)
42         {
43             rowsToUpdate[0]["Age"] = newAge;
44             rowsToUpdate[0]["Name"] = newName;
45
46             // Використовуємо SqlCommandBuilder для генерації команд UPDATE
47             SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
48
49             // Оновлюємо базу даних
50             adapter.Update(dataset, "Users");
51
52             Console.WriteLine("Дані користувача оновлено.");
53
54             Console.WriteLine("\nПОТОЧНІ ДАНІ:");
55             foreach (DataRow row in dataset.Tables["Users"].Rows)
56             {
57                 Console.WriteLine("{0}\t{1}\t{2}", row["Id"], row["Age"], row["Name"]);
58             }
59         }
60         else
61         {
62             Console.WriteLine("Користувача з таким ID не знайдено.");
63         }
64     }
65     catch (Exception ex)
66     {
67         Console.WriteLine("Помилка: " + ex.Message);
68     }
69 }
```

Рисунок 2.3 – Програма завантаження та зміни даних

На рисунку 2.4 зображені зміни даних у базі після виконання програми.

	Id	Age	Name
1	1	20	Ihor

Рисунок 2.4 – Оновлені дані в базі даних

### Контрольні питання:

#### 1. В чому полягає різниця між DataSet та SqlDataReader?

- DataSet — це об'єкт у ADO.NET, який дозволяє зберігати дані в оперативній пам'яті у відключеному режимі. Він може містити кілька таблиць (DataTable), зв'язків, обмежень і підтримує навігацію, редагування та збереження змін. Підходить для складних сценаріїв з маніпуляцією даними без постійного підключення до бази.

- SqlDataReader — це об'єкт, який забезпечує зчитування даних з бази даних у підключеному режимі. Працює швидко, але тільки для послідовного зчитування (вперед), без можливості редагування або зберігання даних. Підходить для простої, ефективної обробки великих обсягів даних.

#### 2. Поясніть, навіщо використовується цей фрагмент коду: `adapter.Fill(dataset);`

Цей фрагмент коду використовується для заповнення об'єкта DataSet даними з бази даних.

Об'єкт SqlDataAdapter виконує SQL-запит, отримує результати з бази, і методом Fill() передає їх у DataSet. Це дозволяє зберігати дані у відключеному режимі та працювати з ними без активного з'єднання з БД.

#### 3. Перерахуйте основні класи ADO.NET. Яке призначення кожного із них?

Клас	Призначення
SqlConnection	Встановлює і управляє з'єднанням з базою даних SQL Server.
SqlCommand	Виконує SQL-команди (SELECT, INSERT, UPDATE, DELETE).

Продовження таблиці

SqlDataReader	Читає дані з бази в підключеному режимі. Швидкий і ефективний.
SqlDataAdapter	Міст між базою даних і об'єктами, такими як DataSet. Дозволяє завантажувати та оновлювати дані.
DataSet	Відключене сховище даних у пам'яті. Може містити кілька таблиць і зв'язків.
DataTable	Представляє одну таблицю даних у DataSet.
DataRow, DataColumn	Представляють рядки та стовпці відповідно у DataTable.
SqlParameter	Використовується для передачі параметрів до SQL-запитів, зокрема для захисту від SQL-ін'єкцій.