

Лабораторна робота №8

Робота з базами даних із використанням ADO.NET

План

1. Підключення SQLite до проекту.
2. Приклад програми.
3. Завдання.

1. Для роботи з SQLite, в першу чергу, необхідно до проекту підключити відповідні пакети NuGet (NuGet – система управління пакетами для .NET Framework). Зробити це можна так:

- 1) створити порожню консольну програму;
- 2) натиснути правою кнопкою миші у вікні «Solution Explorer» на проект (Рис. 1);

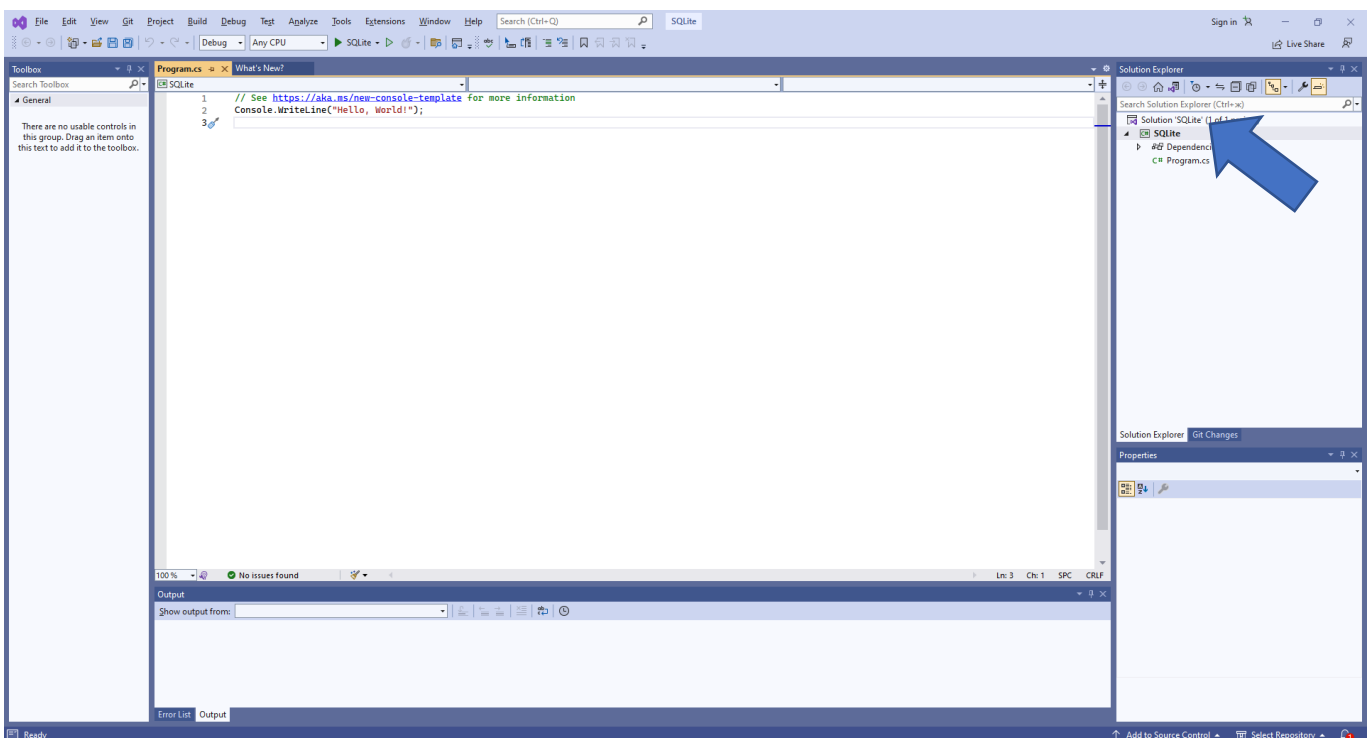


Рис. 1. Вікно «Solution Explorer»

3) у контекстному меню, що з'явилося, вибрати команду «Manage NuGet Packages for Solution...»;

4) ввести в пошукове поле рядок «sqlite»; вибрати та встановити пакет «System.Data.SQLite» (Рис. 2).

Якщо все було зроблено правильно, всі необхідні допоміжні пакети для SQLite будуть завантажені і встановлені автоматично.

2. Розглянемо приклад програми, що реалізує роботу з найпростішою базою даних студентів. Вона складається з трьох таблиць: Dep – списку факультетів; Spe – списку спеціальностей та Std – списку студентів (Рис. 3).

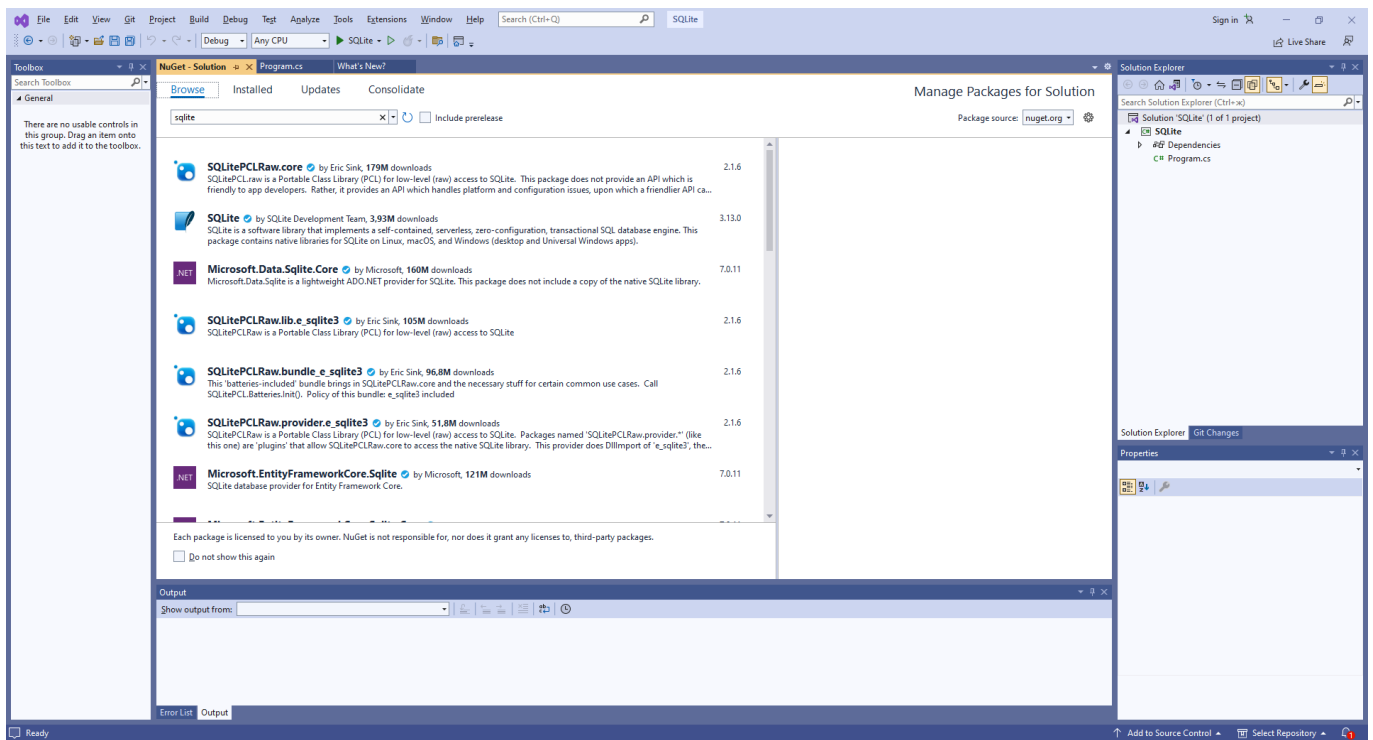


Рис. 2. Вікно «NuGet Solution»

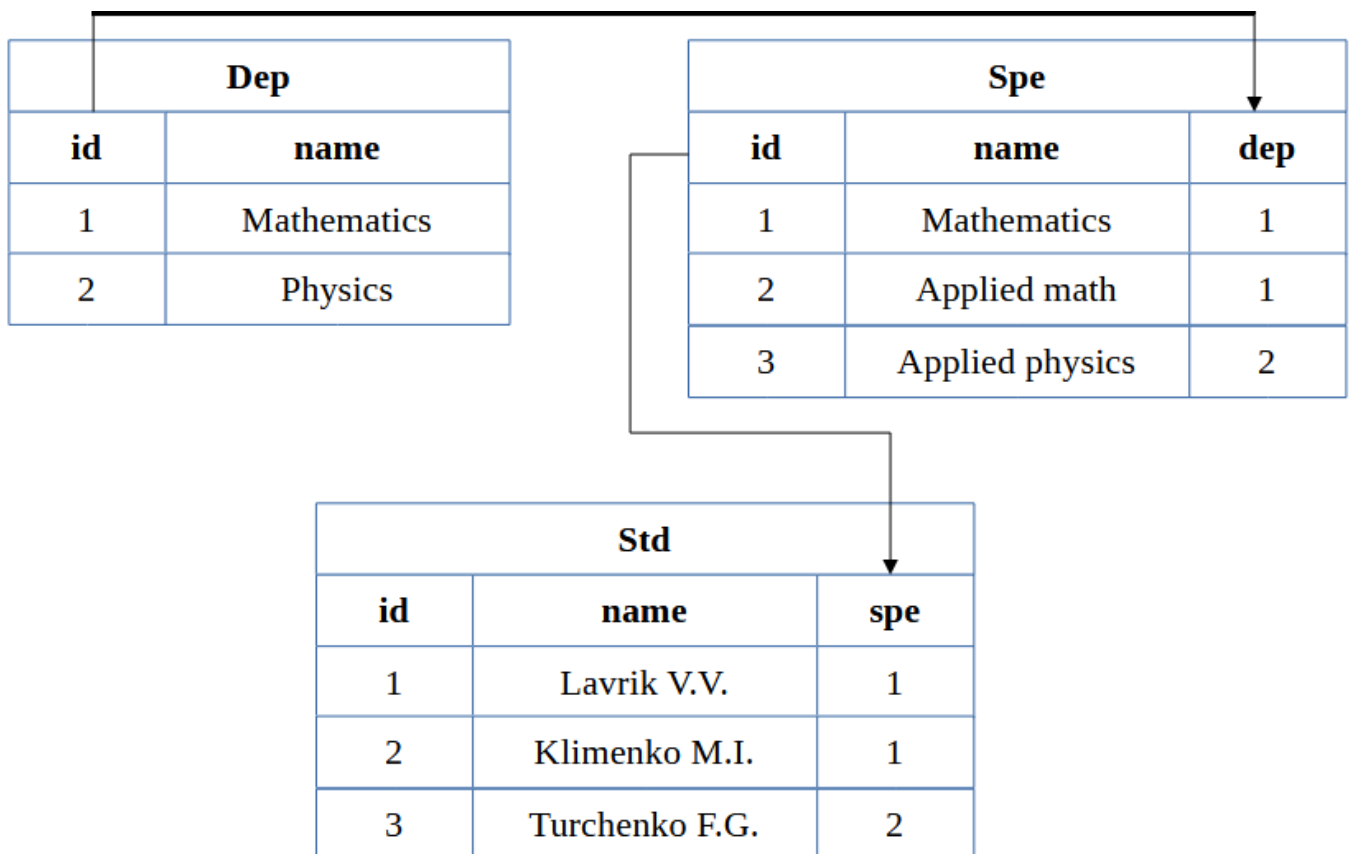


Рис. 3. Структура бази даних «Студент»

```
// Створення БД
using System;
using System.Data.SQLite;
using System.IO;
```

```

static SQLiteConnection CreateDatabase(string dbName)
{
    SQLiteConnection connect;
    SQLiteCommand command = new SQLiteCommand();
    string commandTextDep = "CREATE TABLE Dep (id INTEGER PRIMARY KEY " +
        "AUTOINCREMENT NOT NULL, name NVARCHAR(128))",
        commandTextSpe = "CREATE TABLE Spe (id INTEGER PRIMARY KEY " +
        "AUTOINCREMENT NOT NULL, name NVARCHAR(128), " +
        "dep INTEGER, FOREIGN KEY (dep) REFERENCES Dep(id))",
        commandTextStd = "CREATE TABLE Std (id INTEGER PRIMARY KEY " +
        "AUTOINCREMENT NOT NULL, name NVARCHAR(128), " +
        "spe INTEGER, FOREIGN KEY (spe) REFERENCES Spe(id))";
    SQLiteConnection.CreateFile(dbName);

    // Створити з'єднання та відкрити його
    connect = new SQLiteConnection("Data Source = " + dbName);
    connect.Open();

    command.Connection = connect;
    // Запит на створення таблиці факультетів
    command.CommandText = commandTextDep;
    command.ExecuteNonQuery();
    // Запит на створення таблиці спеціальностей
    command.CommandText = commandTextSpe;
    command.ExecuteNonQuery();
    // Запит на створення таблиці студентів
    command.CommandText = commandTextStd;
    command.ExecuteNonQuery();

    return connect;
}
// Відкриття БД
static SQLiteConnection OpenDatabase(string dbName)
{
    SQLiteConnection connect;

    // Створити БД у разі її відсутності
    if (!File.Exists(dbName))
        connect = CreateDatabase(dbName);
    else
    {
        connect = new SQLiteConnection("Data Source = " + dbName);
        connect.Open();
    }
    return connect;
}
// Закриття БД
static void CloseDatabase(SQLiteConnection connect)
{
    connect.Close();
}

// Запис у БД інформації
// Читання інформації з БД

```

```

static void PrintStudent(SQLiteConnection connect)
{
    var command = new SQLiteCommand("SELECT Std.id, Std.name, Spe.name, " +
        "Dep.name FROM Std, Dep, Spe " +
        "WHERE Std.spe = Spe.id AND " +
        "Spe.dep = Dep.id " +
        "ORDER BY Dep.name, Spe.name", connect);
    var res = command.ExecuteReader();

    Console.WriteLine("{0, 4}\t{1, 20}\t{2, 20}\t{3, 20}", "Id", "Name", "Spe", "Dep");
    while (res.Read())
        Console.WriteLine("{0, 4}\t{1, 20}\t{2, 20}\t{3, 20}", res[0], res[1], res[2], res[3]);
}
// Додавання факультету
static void InsertDep(SQLiteConnection connect, string DepName)
{
    SQLiteCommand command = new SQLiteCommand(String.Format("INSERT INTO Dep (name) " +
        "VALUES('{0}')", DepName),
        connect);

    command.ExecuteNonQuery();
}
// Додавання спеціальності
static void InsertSpe(SQLiteConnection connect, string SpeName, string Dep)
{
    int id = GetId(connect, "Dep", Dep);
    SQLiteCommand command = new SQLiteCommand(String.Format("INSERT INTO Spe (name, dep) "
+
        "VALUES('{0}', {1})", SpeName, id),
        connect);

    command.ExecuteNonQuery();
}
// Додавання студента
static void InsertStd(SQLiteConnection connect, string Name, string SpeName)
{
    int spe = GetId(connect, "Spe", SpeName);
    SQLiteCommand command = new SQLiteCommand(String.Format("INSERT INTO Std (name, spe) "
+
        "VALUES('{0}', {1})", Name, spe),
        connect);

    command.ExecuteNonQuery();
}
// Пошук ключа за значенням
static int GetId(SQLiteConnection connect, string Table, string Value)
{
    var command = new SQLiteCommand(String.Format("SELECT id FROM {0} WHERE name = '{1}'",
        Table, Value), connect);

    var res = command.ExecuteReader();
    int ret = -1;

```

```

while (res.Read())
{
    ret = Convert.ToInt32(res[0]);
    break;
}
if (ret == -1)
    throw new Exception(String.Format("Bad key value: {0}", Value));
return ret;
}

var connect = OpenDatabase("d:/student.db"); // Відкриття БД
SQLiteTransaction transaction = connect.BeginTransaction(); // Запуск транзакції

try
{
    // Додавання факультетів
    InsertDep(connect, "Mathematics");
    InsertDep(connect, "Physics");

    // Додавання спеціальностей
    InsertSpe(connect, "Mathematics", "Mathematics");
    InsertSpe(connect, "Applied mathematics", "Mathematics");
    InsertSpe(connect, "Physics", "Physics");

    // Додавання студентів
    InsertStd(connect, "Trump D.", "Mathematics");
    InsertStd(connect, "Biden J.", "Applied mathematics");
    InsertStd(connect, "Obama B.", "Physics");

    // Виведення інформації про студентів
    PrintStudent(connect);

    transaction.Commit(); // Фіксація транзакції
}
catch (Exception e)
{
    transaction.Rollback(); // Відкат транзакції
    Console.WriteLine(e.Message);
}
connect.Close(); // Закриття БД
Console.ReadKey();

```

3. Використовуючи наведений вище приклад, написати програму, що реалізує роботу з базою даних:

- 1 варіант – «Телефонний довідник» (з урахуванням кількох номерів у абонента);
- 2 варіант – «Домашня відеотека»;
- 3 варіант – «Електронний журнал успішності»;
- 4 варіант – «Електронний каталог книг»;
- 5 варіант – «Відділ кадрів підприємства».

Передбачити виведення на екран ПІБ та номери групи студента, який здає роботу.