

Лабораторна робота №4: SQLite - вбудована БД

Автор: Ясногородський Нікіта Вікторович, ТУ-12-22-Б1ІПЗ

Тема: SQLite - вбудована БД.

Мета: Розробити програму на мові C#, що задовольняє вимоги лабораторної роботи 3.1. Додаток створює БД і виконує кілька операторів вставки для початкового наповнення. Операції вибірки виконувати.

Виконання

```
using Microsoft.Data.Sqlite;
using System.Data;

class Program
{
    private static SqliteConnection connection;

    static void Main()
    {
        string connectionString = "Data Source=Program.db";
        connection = new SqliteConnection(connectionString);
        connection.Open();

        CreateTableA();
        CreateTableVeryLongNameOfTable();

        ExecuteSimpleSelectFromBothTables();
        ExecuteSelectWithAlias();
        ExecuteSelectWithCondition();
        ExecuteCrossJoin();
        ExecuteJoinWithCondition();
        ExecuteJoinWithTwoAsAndOneB();
        ExecuteLeftJoinAtoB();
        ExecuteLeftJoinBtoA();

        connection.Close();
    }

    static void CreateTableA()
    {
        string createTableACommand = "CREATE TABLE IF NOT EXISTS A (id INTEGER);";
        ExecuteNonQuery(createTableACommand);
        string insertDataACommand = "INSERT INTO A (id) VALUES (1), (2), (3), (4), (NULL);";
        ExecuteNonQuery(insertDataACommand);
    }
}
```

```

}

static void CreateTableVeryLongNameOfTable()
{
    string createTableCommand = "CREATE TABLE IF NOT EXISTS veryLongNameOfTable
(id INTEGER);";
    ExecuteNonQuery(createTableCommand);
    string insertDataCommand = "INSERT INTO veryLongNameOfTable (id) VALUES (1),
(2), (300), (NULL);";
    ExecuteNonQuery(insertDataCommand);
}

static void ExecuteSimpleSelectFromBothTables()
{
    string query = "SELECT * FROM A, veryLongNameOfTable;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteSelectWithAlias()
{
    string query = "SELECT A.id AS A_id, B.id AS B_id FROM A, veryLongNameOfTable
AS B;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteSelectWithCondition()
{
    string query = $"SELECT A.id AS A_id, B.id AS B_id FROM A, veryLongNameOfTable
AS B WHERE A.id = B.id;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteCrossJoin()
{
    string query = "SELECT A.id, B.id FROM A CROSS JOIN veryLongNameOfTable B;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteJoinWithCondition()
{
    string query = "SELECT A.id, B.id FROM A JOIN veryLongNameOfTable B ON A.id =
B.id;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteJoinWithTwoAsAndOneB()
{
    string query = "SELECT A1.id, A2.id, B.id FROM A A1 JOIN veryLongNameOfTable B
ON A1.id = B.id JOIN A A2 ON A1.id = A2.id AND A1.id != A2.id;";
    ExecuteQueryWithFormattedOutput(query);
}

```

```

static void ExecuteLeftJoinAtoB()
{
    string query = "SELECT A.id AS A_id, B.id AS B_id FROM A LEFT OUTER JOIN
veryLongNameOfTable B ON A.id = B.id;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteLeftJoinBtoA()
{
    string query = "SELECT B.id AS B_id, A.id AS A_id FROM veryLongNameOfTable B
LEFT OUTER JOIN A ON B.id = A.id;";
    ExecuteQueryWithFormattedOutput(query);
}

static void ExecuteNonQuery(string query)
{
    using (var command = new SqlCommand(query, connection))
    {
        command.ExecuteNonQuery();
    }
}

static void ExecuteQueryWithFormattedOutput(string query)
{
    using (var command = new SqlCommand(query, connection))
    {
        using (var reader = command.ExecuteReader())
        {
            var dataTable = new DataTable();
            dataTable.Load(reader);
            PrintDataTable(dataTable);
        }
    }
}

static void PrintDataTable(DataTable table)
{
    foreach (DataColumn col in table.Columns)
    {
        Console.Write($"{col.ColumnName,-20}");
    }
    Console.WriteLine();

    foreach (DataRow row in table.Rows)
    {
        foreach (var item in row.ItemArray)
        {
            Console.Write($"{item,-20}");
        }
        Console.WriteLine();
    }
}

```

```

    }
    Console.WriteLine();
}
}

```

```

1:[tmux]

~пу/Бази_даних/4 master !1 ?2 > dotnet build && dotnet run
> dotnet build && dotnet run
MSBuild version 17.9.6+a4ecab324 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
/Users/nickyasnogorodskyi/Documents/personal/labs/5_term/Бази_даних/4/Program.cs(6,37): warning CS8618: Non-nullable field 'id1' is declared as nullable. [/Users/nickyasnogorodskyi/Documents/personal/labs/5_term/Бази_даних/4/Program.cs]
4 → /Users/nickyasnogorodskyi/Documents/personal/labs/5_term/Бази_даних/4/bin/Debug/net8.0/4.dll

Build succeeded.

/Users/nickyasnogorodskyi/Documents/personal/labs/5_term/Бази_даних/4/Program.cs(6,37): warning CS8618: Non-nullable field 'id1' is declared as nullable. [/Users/nickyasnogorodskyi/Documents/personal/labs/5_term/Бази_даних/4/Program.cs]
1 Warning(s)
0 Error(s)

Time Elapsed 00:00:01.55
id            id1
1             1
1             2
1             300
1
2             1
2             2
2             300
2
3             1
3             2
3             300
3
4             1
4             2
4             300
4
           1
           2
           300

A_id          B_id
1             1
1             2
1             300
1
2             1

```

Висновок

Під час виконання лабораторної роботи №4 на тему "SQLite - вбудована БД", розробив програму на мові C#, що створює базу даних Program.db, наповнює таблиці даними, та виконує складні запити. Використано передові практики управління ресурсами та підтверджено ефективність SQLite як надійної системи управління базами даних.