

Лабораторна робота №4: SQLite - вбудована БД

Автор: Матушевич Ярослав, студент ТУ-8-21-Б1ІПЗ(4.6здс)

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 System;
using Microsoft.Data.Sqlite;

class Program
{
    static void Main(string[] args)
    {
        string connectionString = "Data Source=lab4.db";

        using (var connection = new SqliteConnection(connectionString))
        {
            connection.Open();

            CreateTableA(connection);
            CreateTableVeryLongNameOfTable(connection);

            // Виконуємо різні SELECT запити з JOIN
            PerformSimpleSelectFromBothTables(connection);
            PerformSelectWithAlias(connection);
            PerformSelectWithCondition(connection, "B");
            PerformCrossJoin(connection);
            PerformJoinWithCondition(connection);
            PerformJoinWithTwoAsAndOneB(connection);
            PerformLeftOuterJoin(connection, true); // A LEFT JOIN B
            PerformLeftOuterJoin(connection, false); // B LEFT JOIN A
        }
    }

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

```

}

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

static void PerformSimpleSelectFromBothTables(SqliteConnection connection)
{
    string query = "SELECT * FROM A, veryLongNameOfTable;";
    ExecuteQuery(connection, query);
}

static void PerformSelectWithAlias(SqliteConnection connection)
{
    string query = "SELECT A.id, B.id FROM A, veryLongNameOfTable AS B;";
    ExecuteQuery(connection, query);
}

static void PerformSelectWithCondition(SqliteConnection connection, string alias)
{
    string query = $"SELECT A.id AS A_id, {alias}.id AS {alias}_id FROM A,
veryLongNameOfTable AS {alias} WHERE A.id = {alias}.id;";
    ExecuteQuery(connection, query);
}

static void PerformCrossJoin(SqliteConnection connection)
{
    string query = "SELECT A.id, B.id FROM A CROSS JOIN veryLongNameOfTable B;";
    ExecuteQuery(connection, query);
}

static void PerformJoinWithCondition(SqliteConnection connection)
{
    string query = "SELECT A.id, B.id FROM A JOIN veryLongNameOfTable B ON A.id =
B.id;";
    ExecuteQuery(connection, query);
}

static void PerformJoinWithTwoAsAndOneB(SqliteConnection connection)
{
    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;";
    ExecuteQuery(connection, query);
}

```

```

static void PerformLeftOuterJoin(SqliteConnection connection, bool aLeftJoinB)
{
    string query;
    if (aLeftJoinB)
    {
        query = "SELECT A.id, B.id FROM A LEFT OUTER JOIN veryLongNameOfTable B ON
A.id = B.id;";
    }
    else
    {
        query = "SELECT B.id, A.id FROM veryLongNameOfTable B LEFT OUTER JOIN A ON
B.id = A.id;";
    }
    ExecuteQuery(connection, query);
}

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

static void ExecuteQuery(SqliteConnection connection, string query)
{
    using (var command = new SqliteCommand(query, connection))
    {
        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                for (int i = 0; i < reader.FieldCount; i++)
                {
                    Console.Write($"{reader.GetValue(i)}\t");
                }
                Console.WriteLine();
            }
        }
    }
}

```

```
ymatushevych.appwell@YMATUSHEVYCH 4 % dotnet build && dotnet run
MSBuild version 17.9.6+a4ecab324 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
4 -> /Users/ymatushevych.appwell/Desktop/baza danych/4/bin/Debug/net8.0/4.dll

Build succeeded.
[ 0 Warning(s)
  0 Error(s)

Time Elapsed 00:00:00.95
```

Висновок

У ході виконання лабораторної роботи №4 на тему "SQLite - вбудована БД", я, Матушевич Ярослав, студент групи ТУ-8-21-Б1ПЗ(4.6зс), розробив програму на мові C#, що відповідає всім вимогам задачі лабораторної роботи 3.1. Програма успішно створює базу даних lab4.db та виконує оператори вставки для формування початкових даних у таблицях A та veryLongNameOfTable, заповнюючи їх вказаними значеннями, включно з NULL.

Окрім створення та наповнення таблиць даними, було виконано різноманітні оператори вибірки з використанням різних типів JOIN (CROSS JOIN, LEFT JOIN тощо), що дозволило демонструвати можливості SQLite для здійснення складних запитів до бази даних. Це підкреслило ефективність SQLite як легковагової, але потужної системи управління базами даних, здатної задовольняти потреби навіть у складних операторах вибірки.

Реалізація програми також включала в себе застосування передових практик роботи з базами даних, таких як коректне управління ресурсами через використання конструкції using, що забезпечує автоматичне закриття з'єднань та інших ресурсів, тим самим запобігаючи витоку пам'яті.

Проведена робота засвідчила, що SQLite є ідеальним вибором для проектів, де необхідна легка та надійна вбудована база даних. Вона пропонує широкий спектр можливостей для управління даними без потреби у складному налаштуванні чи адмініструванні, що робить її відмінним рішенням для розробки десктопних, мобільних або вбудованих додатків.

Виконання цієї лабораторної роботи не тільки дозволило глибше зрозуміти принципи роботи з реляційними базами даних та їх застосування на практиці, але й покращило навички програмування на C# та роботи з бібліотекою Microsoft.Data.Sqlite.