

.net中连接数据库太麻烦? 试试文件型数据库SQLite吧

作者: 微信公众号:【架构师老卢】

11-22 7:39

167



**概述:** SQLite 是一种轻量级的嵌入式数据库引擎,它在 .NET 中被广泛使用。SQLite 是一个零配置的数据库引擎,不需要服务器,可以直接在应用程序中使用。下面是一个简单的示例,演示如何在 .NET 中使用 SQLite,并提供了常见的查询、增加、修改和删除功能。

SQLite 是一种轻量级的嵌入式数据库引擎,它在 .NET 中被广泛使用。SQLite 是一个零配置的数据库引擎,不需要服务器,可以直接在应用程序中使用。下面是一个简单的示例,演示如何在 .NET 中使用 SQLite,并提供了常见的查询、增加、修改和删除功能。

首先,你需要在项目中安装 `System.Data.SQLite` 包。你可以使用 NuGet 包管理器或通过 Package Manager Console 执行以下命令:

```
1 | Install-Package System.Data.SQLite
```

接下来,创建一个 C# 文件,例如 `SQLiteExample.cs`,并添加以下代码:

```
1 using System;
2 using System.Data.SQLite;
3
4 class Program
5 {
6     static void Main()
7     {
8         // 指定数据库文件路径
9         string dbFilePath = "sample.db";
10
11        // 连接字符串
12        string connectionString = $"Data Source={dbFilePath};Version=3;";
13
14        // 创建数据库连接
15        using (SQLiteConnection connection = new SQLiteConnection(connectionString))
16        {
17            connection.Open();
18
19            // 创建表
20            CreateTable(connection);
21
22            // 插入数据
23            InsertData(connection, "John Doe", 30);
24
25            // 查询数据
26            QueryData(connection);
27
28            // 更新数据
29            UpdateData(connection, 1, "Updated Name", 35);
30
31            // 查询更新后的数据
32            QueryData(connection);
33
34            // 删除数据
35            DeleteData(connection, 1);
36
37            // 查询删除后的数据
38            QueryData(connection);
39        }
40    }
41
42    static void CreateTable(SQLiteConnection connection)
43    {
44        using (SQLiteCommand command = new SQLiteCommand(
45            "CREATE TABLE IF NOT EXISTS Users (Id INTEGER PRIMARY KEY AUTOINCREMENT, Name TEXT, Age INTEGER);", connection))
46        {
47            command.ExecuteNonQuery();
48        }
49
50        Console.WriteLine("Table created or already exists.");
51    }
52
53    static void InsertData(SQLiteConnection connection, string name, int age)
54    {
55        using (SQLiteCommand command = new SQLiteCommand(
56            "INSERT INTO Users (Name, Age) VALUES (@Name, @Age);", connection))
57        {
58            command.Parameters.AddWithValue("@Name", name);
59            command.Parameters.AddWithValue("@Age", age);
60
61            command.ExecuteNonQuery();
62        }
63
64        Console.WriteLine("Data inserted.");
65    }
66
67    static void QueryData(SQLiteConnection connection)
68    {
69        using (SQLiteCommand command = new SQLiteCommand(
70            "SELECT * FROM Users;", connection))
71        {
72            using (SQLiteDataReader reader = command.ExecuteReader())
73            {
74                Console.WriteLine("Id\tName\tAge");
75                while (reader.Read())
76                {
77                    Console.WriteLine($"{reader["Id"]}\t{reader["Name"]}\t{reader["Age"]}");
78                }
79            }
80        }
81        Console.WriteLine("Data queried.");
82    }
83
84    static void UpdateData(SQLiteConnection connection, int id, string name, int age)
85    {
86        using (SQLiteCommand command = new SQLiteCommand(
87            "UPDATE Users SET Name=@Name, Age=@Age WHERE Id=@Id;", connection))
88        {
89            command.Parameters.AddWithValue("@Name", name);
90            command.Parameters.AddWithValue("@Age", age);
91            command.Parameters.AddWithValue("@Id", id);
92
93            command.ExecuteNonQuery();
94        }
95
96        Console.WriteLine("Data updated.");
97    }
98
99    static void DeleteData(SQLiteConnection connection, int id)
100    {
101        using (SQLiteCommand command = new SQLiteCommand(
102            "DELETE FROM Users WHERE Id=@Id;", connection))
103        {
104            command.Parameters.AddWithValue("@Id", id);
105
106            command.ExecuteNonQuery();
107        }
108
109        Console.WriteLine("Data deleted.");
110    }
111 }
```

请注意,上述示例假设你已经安装了 `System.Data.SQLite` 包,并且在项目目录中创建了一个名为 `sample.db` 的 SQLite 数据库文件。在实际应用中,你需要根据自己的需求修改数据库文件路径和连接字符串。

这个示例演示了如何创建表、插入数据、查询数据、更新数据和删除数据。你可以根据具体的应用场景和需求进行修改和扩展。