

Lab10 ADO.NET

学习目标:

1.掌握断开式连接数据库方法

一、断开式连接数据库

DataAdapter 对象（当使用 MySQL 数据库时，为 MySqlConnectionAdapter 对象）是用来连接数据源与数据集 DataSet 的，而在一个 DataSet 对象实例中，可以包含多个 DataTable，而一个 DataTable 可以包含多个 DataRow。DataAdapter 的主要方法描述如下。

①Fill()方法：用于将数据源的查询结果添加到数据集。

②Update()方法：用于将数据集中数据存储到数据源中。

1.使用 DataAdapter 对象查询数据

使用 DataAdapter 对象查询数据时，首先建立数据库连接，然后利用 select 语句和数据连接建立 DataAdapter 对象，并使用 DataAdapter 对象的 Fill()方法把查询结果放在 DataSet 对象的一个数据表中，最后可使用数据控件展示数据。

2.使用 DataAdapter 对象增加、修改和删除数据

使用 DataAdapter 对象增加数据时，首先建立数据库连接，然后利用 select 语句和数据连接建立 DataAdapter 对象，并建立 SqlCommandBuilder 对象自动生成 DataAdapter 的 SqlCommand 命令，并使用 DataAdapter 对象的 Fill()方法把查询结果放在 DataSet 对象的一个数据表中，并实现对 DataTable 对象中数据的增加、修改或删除，并通过 DataAdapter 对象的 Update()方法向数据库提交数据。

【注意】只有当 DataAdapter 操作单个数据库表时，才可以利用 SqlCommandBuilder 对象自动生成 DataAdapter 的 DeleteCommand、InsertCommand 和 UpdateCommand。为了自动生成命令，必须设置 SelectCommand 属性，SelectCommand 检索表架构以此确定自动生成的 INSERT、UPDATE 和 DELETE 语句的语法。

二、用户管理项目

- (1) 打开Lab9中创建的用户管理项目
- (2) 为项目添加Mysql引用
- (3) 为项目添加Configuration引用
- (4) 为配置文件App.config设置连接字符串的值，代码如下。

```
<connectionStrings >
  <add name="conStr" connectionString="server=localhost;user
id=root;password=123456;database=studentmanage;SslMode = none;" />
</connectionStrings>
```

- (5) 为项目添加引用，代码如下。

```
using MySql.Data.MySqlClient;
using System.Configuration;
```

- (6) 在Form1类中添加成员变量conStr，用于保存数据库连接字符串，并创建 MySqlConnection、MySqlDataAdapter、DataSet和MySqlCommandBuilder对象，代码如下。

```
string conStr = ConfigurationManager.ConnectionStrings["conStr"].ConnectionString;
MySqlConnection con;
MySqlDataAdapter da;
DataSet ds;
MySqlCommandBuilder builder;
```

- (7) 双击“查询”按钮，为其添加单击事件处理程序，程序代码如下。

```
private void btnSelect_Click(object sender, EventArgs e)
```

```

{
    con = new MySqlConnection(conStr);
    da = new MySqlDataAdapter("select * from user", con);
    ds = new DataSet();
    da.Fill(ds, "user");
    dgvUser.DataSource = ds.Tables["user"];
}

```

【注意】DataGridView控件是常用的数据控件，它以表格的形式显示数据源中的数据。在DataGridView控件中进行数据绑定时，只需设置DataSource属性。例如上述代码中语句，dgvStudentScore.DataSource = ds.Tables["user"];是将数据集中“user”表的数据绑定到DataGridView控件中。

(8) 双击“增加”按钮，为其添加单击事件处理程序，程序代码如下。

```

private void btnInsert_Click(object sender, EventArgs e)
{
    builder = new MySqlCommandBuilder(da);
    DataRow r1 = ds.Tables["user"].NewRow();
    r1[0] = txtName.Text;
    r1[1] = txtPwd.Text;
    r1[2] = txtRole.Text;
    ds.Tables[0].Rows.Add(r1);
    da.Update(ds, "user");
    dgvUser.DataSource = ds.Tables["user"];
}

```

(9) 双击表格dgvUser，为其添加单元格单击事件处理程序，程序代码如下。

```

private void dgvStudentScore_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    txtNum.Text = dgvStudentScore.CurrentRow.Cells["userName"].Value.ToString();
    txtName.Text = dgvStudentScore.CurrentRow.Cells["userPwd"].Value.ToString();
    txtScore.Text = dgvStudentScore.CurrentRow.Cells["role"].Value.ToString();
    txtName.Enabled=false;
}

```

以上代码的功能是，当用户点击表格中的某一个单元格时，将单元格中对应一行数据显示在相应的文本框中。

(10) 双击“修改”按钮，为其添加单击事件处理程序，程序代码如下。

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    builder = new MySqlCommandBuilder(da);
    DataRowCollection rows = ds.Tables["user"].Rows;
    DataRow row;
    for (int i = 0; i < rows.Count; i++)
    {
        row = rows[i];
        if (row["userName"].ToString() == txtName.Text)

```

```
        {  
            row["userPwd"] = txtPwd.Text;  
            row["role"] = Convert.ToInt32(txtRole.Text);  
        }  
    }  
    dgvUser.DataSource = ds.Tables["user"];  
    da.Update(ds, "user");  
}
```

- (11) 双击“删除”按钮，为其添加单击事件处理程序。
- (12) 在解决方案资源管理器中右击项目，将其设为启动项目。
- (13) 编译并运行。