

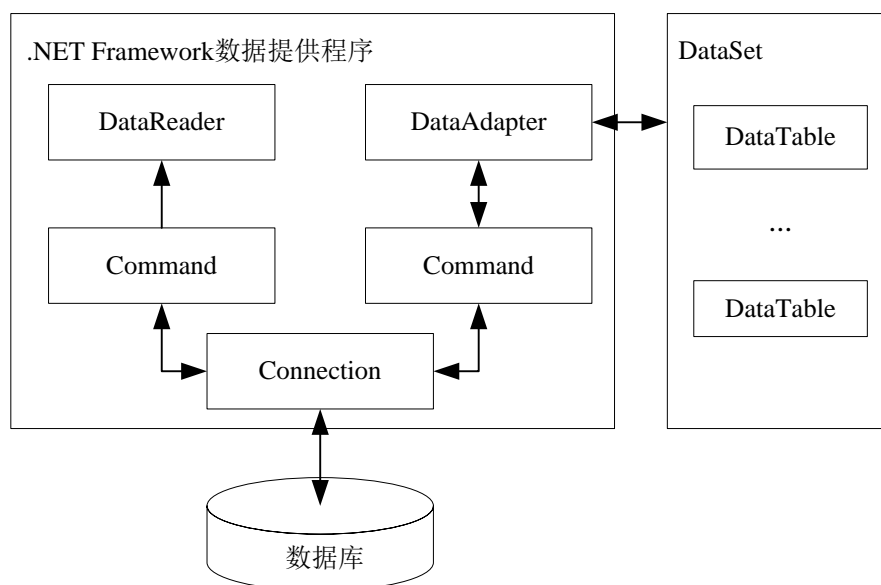
Lab9 ADO.NET

学习目标:

- 1.掌握在 VS 中操作 MySQL 数据库的方法
- 2.掌握非断开式连接数据库方法

一、ADO.NET

ADO.NET 是微软.NET 框架的组件，是在基于.NET 平台的应用程序中用于访问数据源的技术，也是.NET Framework 提供的数据库访问的类库。应用程序可以使用 ADO.NET 连接到这些数据源，并检索和更新所包含的数据。ADO.NET 体系结构如下图所示，其中包含两大核心组件，即.NET Framework 数据提供程序和 DataSet。



.NET Framework 数据提供程序包含以下 4 个核心类：

- (1) Connection：建立与数据源的连接。
- (2) Command：对数据源执行操作命令，用于修改数据、查询数据和运行存储过程等。
- (3) DataReader：从数据源获取只读只进的数据流。
- (4) DataAdapter：用数据源数据填充 DataSet，并可以处理数据更新。

二、ADO.NET 数据库访问操作

- (1) 在解决方案管理器中添加对 MySQL 的引用
- (2) 在代码窗口中添加引用语句：`using MySql.Data.MySqlClient;`

三、非断开式访问数据库

ADO.NET 提供了两种访问数据库的方式，即非断开式和断开式。

非断开式的访问数据库，在取得数据库连接之后将一直保持与数据库的连接，直到执行关闭连接的操作。具体步骤如下：

- (1) 通过数据库连接类（Connection）连接到数据库。
- (2) 通过数据库命令类（Command）在数据库上执行 SQL 语句，实现对数据库的插入（Insert）、删除（Delete）、更新（Update）以及查询（Select）等操作。
- (3) 如果是查询操作，可以通过数据库读取器类（DataReader）进行数据记录的向前只读操作。
- (4) 数据库操作完成后，再通过连接类（Connection）关闭当前的数据连接。

在非断开式访问数据库过程中，由于数据库的客户连接数量有限，因此，应尽量缩短与数据库操作的时间，数据库操作一旦完成，应及时关闭对应的数据库连接。

四、非断开式访问数据库练习

1、使用 Connection 对象连接数据库

① 直接构造

```
string conStr = "server=localhost;user id=root;password=123456;database=Hotel";
```

② 配置文件中存储连接字符串

步骤：（1）配置文件代码

```
<connectionStrings >
```

```
    <add name="conStr" connectionString="server=localhost;user  
id=root;password=123456;database=hotel"/>
```

```
</connectionStrings>
```

（2）给项目添加引用（3）代码页using System.Configuration;

（4）从配置文件中获取连接字符串

```
string conStr = ConfigurationManager.ConnectionStrings  
["conStr"].ConnectionString;
```

2、编写代码实现从 Hotel 数据库的 user 表中读取全部用户数据。

参考代码如下：

```
MySqlConnection con = new MySqlConnection(conStr);  
MySqlCommand cmd = new MySqlCommand("select * from user", con);  
con.Open();  
MySqlDataReader reader = cmd.ExecuteReader();  
while (reader.Read())  
{  
    //可使用多种方法输出  
    Console.Write(reader[0].ToString() + " ");  
    Console.Write(reader["userPwd"].ToString() + " ");  
    Console.WriteLine(reader.GetString(2));  
}  
reader.Close();  
con.Close();
```

3、编写代码实现在 Hotel 数据库的 user 表添加一条用户数据。

参考代码：

```
MySqlCommand cmd = new MySqlCommand("insert into user(userName,userPwd,role)  
values(@userName,@userPwd,@role)", con);  
cmd.Parameters.AddWithValue("@userName", "xiaoming");  
cmd.Parameters.AddWithValue("@userPwd", "123");  
cmd.Parameters.AddWithValue("@role", 0);  
cmd.ExecuteNonQuery();
```

4、编写代码实现修改一条数据。

5、编写代码实现删除一条数据。

五、用户管理设计（下次课练习使用）

1.创建如下图所示窗体

控件属性设置

控件	属性	属性值
Form1	Text	用户管理
label1	Text	用户名:
textBox1	Name	txtName
label2	Text	密码:
textBox2	Name	txtPwd
label3	Text	权限:
textBox3	Name	txtRole
dataGridView1	Name	dgvUser
button1	Name	btnSelect
	Text	查询
button2	Name	btnInsert
	Text	增加
button3	Name	btnUpdate
	Text	修改
button4	Name	btnDelete
	Text	删除