

实验要求：

使用 ADO.NET 并任选一种程序设计语言进行编程。

实验报告需要说明所采用的方法、结果和总结（结果分析）。

实验方法部分说明采用的开发环境，包括操作系统、数据库管理系统及其版本、编程工具及其版本、和编程语言。如果使用 ADO.NET 则在实验方法部分回答下面的问题：

- (1) 使用哪种数据提供程序？
- (2) 使用的数据连接对象是哪一个？连接对象是如何建立的？最后生成的连接对象中的连接字符串是什么？代表什么含义？
- (3) 使用的数据适配器对象是什么？其中的查询或更新语句是什么？如果有参数则参数是如何处理的？
- (4) 使用的数据集对象是什么？数据集中有哪些数据表？数据表是由哪些适配器对象生成的？（或采用其它方法）。

实验方法中还需要给出手工添加的代码及对代码的说明。

实验结果部分给出程序运行的界面和操作的简单说明。

总结部分对实验过程中出现的问题进行分析，同时提出所开发的程序还有哪些可以改进的地方。

开发环境：

操作系统：Windows10 数据库版本：SQL server2016 编译器：Visual Studio2017
编程语言：ADO.NET

实验二：

实验内容：

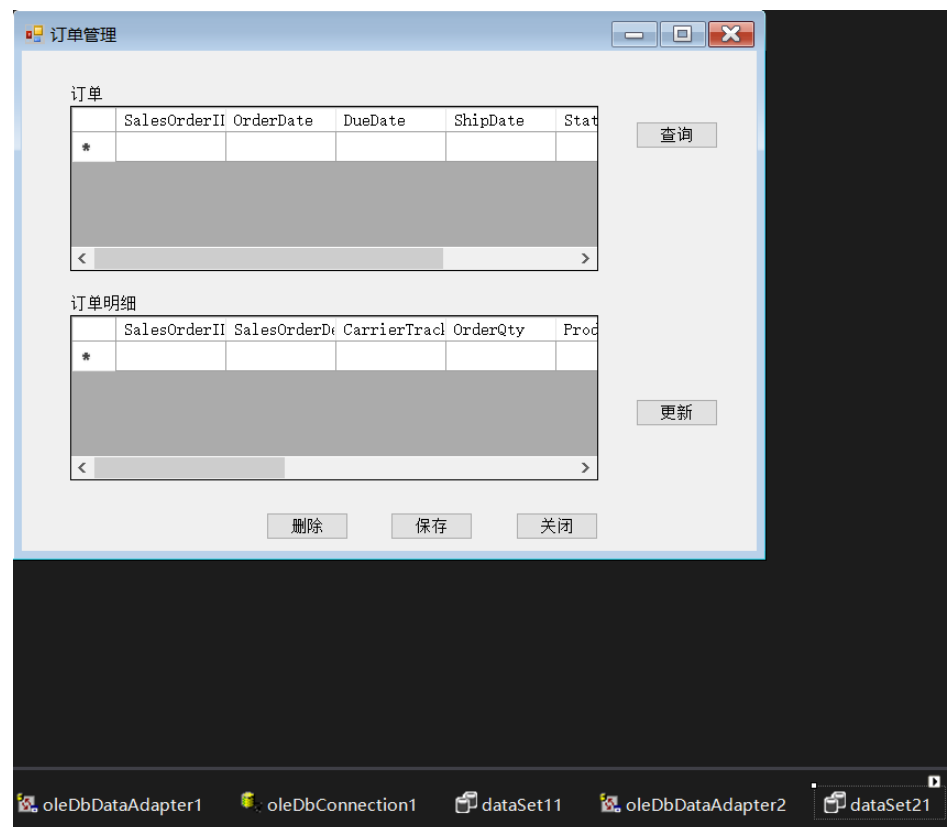
编程实现示例数据库中订单明细 SalesOrderDetail 的查询、插入、更新与删除功能。

实验方法：

1. 本实验采用 OLEDB 类数据提供程序，
2. 本实验使用的连接对象是 OleDbConnection1，通过数据适配器连接，生成的连接对象中的连接字符串是 Provider=SQLNCLI11;Data Source=localhost;Integrated Security=SSPI;Initial Catalog=AdventureWorks, 其中 SQLNCLI11 是客户端组件, localhost 是服务器, SSPI 是身份验证方式, AdventureWorks 是初始化数据库。
3. 本实验中使用了两个数据适配器 OleDbDataAdapter1 和 OleDbDataAdapter2, 查询语句分别是 SELECT SalesOrderID, OrderDate, DueDate, ShipDate, Status, SalesOrderNumber FROM Sales.SalesOrderHeader ; 和 SELECT * FROM Sales.SalesOrderDetail;
4. 本实验使用的数据集对象为 dataSet11 和 dataSet21, dataSet11 中有 Sales.SalesOrderHeader 表, 由 OleDbDataAdapter1 生成, dataSet21 中有 Sales.SalesOrderDetail 表, 由 OleDbDataAdapter2 生成。

实验过程：

首先通过拖动控件设计出下图所示的界面，包括两个 label,控件两个 dataGridView 控件(订单和订单明细)和五个 Button 控件（具体设计过程参考《数据库系统开发实验指导 3.1》的 7.1.2 节），然后创建两个数据适配器 OleDbDataAdapter1 和 OleDbDataAdapter2，对两个数据适配器分别生成一个数据集 dataSet11 和 dataSet21,然后订单表引用 dataSet11 中的填充的 SalesOrderHeader 的数据作为数据源，订单明细表把 dataSet21 中的填充的 SalesOrderDetail 的数据作为数据源。



界面设计好之后变点击界面空白处填充数据

```
34 private void Form1_Load(object sender, EventArgs e)
35 {
36     this.OleDbDataAdapter1.Fill(this.dataSet11);
37     this.OleDbDataAdapter2.Fill(this.dataSet21);
38 }
```

填充完毕后运行结果如下图

订单管理

订单

	SalesOrderID	OrderDate	DueDate
▶	43659	2011/5/31	2011/6/12
	43660	2011/5/31	2011/6/12
	43661	2011/5/31	2011/6/12

查询

订单明细

	SalesOrderID	SalesOrderDetail	CarrierTrackingID
▶	43659	1	4911-403C-98
	43659	2	4911-403C-98
	43659	3	4911-403C-98

插入

更新

删除 保存 关闭

设计查询按钮的功能代码

```

40 private void btnSelect_Click(object sender, EventArgs e)
41 {
42     //从订单表中获取当前行的SalesOrderID
43     string SalesOrderID = dgvOrder.Rows[this.dgvOrder.CurrentRow.Index].Cells[0].Value.ToString();
44     //修改查询命令
45     string whereStmt = "SalesOrderID=" + SalesOrderID;
46     String oldSelectStmt = oleDbDataAdapter2.SelectCommand.CommandText;
47
48     String selectStmt = oleDbDataAdapter2.SelectCommand.CommandText;
49     if (whereStmt.Length > 0) selectStmt += " WHERE " + whereStmt;
50
51     oleDbDataAdapter2.SelectCommand.CommandText=selectStmt;
52     //将订单明细表对应的数据集清空
53     dataSet21.Clear();
54     //使用新的查询命令填充数据集
55     oleDbDataAdapter2.Fill(dataSet21);
56     //把查询命令恢复
57     oleDbDataAdapter2.SelectCommand.CommandText = oldSelectStmt;
58 }

```

当点击查询按钮后，订单明细表中会显示订单表中当前行的 SalesOrderID 对应的相关订单明细情况

订单

	SalesOrderID	OrderDate	DueDate
	43659	2011/5/31	2011/6/12
▶	43660	2011/5/31	2011/6/12
	43661	2011/5/31	2011/6/12

查询

订单明细

	SalesOrderID	SalesOrderDetailID	CarrierTrackingID	OrderQty
▶	43660	13	6431-4D57-83	1
	43660	14	6431-4D57-83	1
•				

插入

更新

删除

保存

关闭

删除按钮设计代码如下：

```

60 private void btnDelete_Click(object sender, EventArgs e)
61 {
62     int index = -1;
63     foreach (DataRow dataRow in dataSet21.SalesOrderDetail.Rows)
64     {
65         DataRowState rowState = dataRow.RowState;
66         //如果该行已被删除，跳出
67         if (rowState == DataRowState.Deleted)
68             continue;
69         //否则，添加索引，执行删除操作
70         index++;
71         if (this.dgvOrderDetail.CurrentRow.Index == index)
72         {
73             dataRow.Delete();
74             break;
75         }
76     }
77 }

```

点击订单明细表中的某一行，然后点击删除，当前行便从数据集中删除

订单管理

订单

	SalesOrderID	OrderDate	DueDate
	43659	2011/5/31	2011/6/12
▶	43660	2011/5/31	2011/6/12
	43661	2011/5/31	2011/6/12

查询

订单明细

	SalesOrderID	SalesOrderDetailID	CarrierTrackingID	OrderQty
▶	43660	13	6431-4D57-83	1
•				

插入

更新

删除

保存

关闭

插入操作直接在表格下方插入即可

订单管理

— □ ×

订单

	SalesOrderID	OrderDate	DueDate
	43659	2011/5/31	2011/6/12
▶	43660	2011/5/31	2011/6/12
	43661	2011/5/31	2011/6/12

查询

订单明细

	SalesOrderID	SalesOrderDetail	CarrierTracking
	43660	13	6431-4D57-83
	43660	14	6431-4D57-83
✎	null	121318	

更新

删除

保存

关闭

更新按钮设计代码如下

```

79     private void btnUpdate_Click(object sender, EventArgs e)
80     {
81         try
82         {
83             this.oleDbDataAdapter2.Update(this.dataSet21.SalesOrderDetail);
84             MessageBox.Show("更新成功!");
85         }
86         catch (Exception ex)
87         {
88             MessageBox.Show("更新失败!\n" + ex.Message);
89         }
90     }

```

更新前

订单管理

订单

	SalesOrderID	OrderDate	DueDate
	43659	2011/5/31	2011/6/12
	43660	2011/5/31	2011/6/12
▶	43661	2011/5/31	2011/6/12

查询

订单明细

	alOfferID	UnitPrice	UnitPriceDiscow	LineT
▶		809.76	0.0000	809.76
		714.7043	0.0000	714.70
		714.7043	0.0000	1429.4

更新

删除 保存 关闭

更新后

订单管理

订单

	SalesOrderID	OrderDate	DueDate
	43659	2011/5/31	2011/6/12
	43660	2011/5/31	2011/6/12
▶	43661	2011/5/31	2011/6/12

查询

订单明细

	alOfferID	UnitPrice	UnitPriceDiscow	LineT
▶		800.00	0.0000	809.76
		714.7043	0.0000	714.70
		714.7043	0.0000	1429.4

更新

删除 保存 关闭

Form1.cs

```
private void
{
    this.ole
    this.ole
}
```

更新成功!

确定

String s
if (wher

oleDbDat
dataSet2
oleDbDat

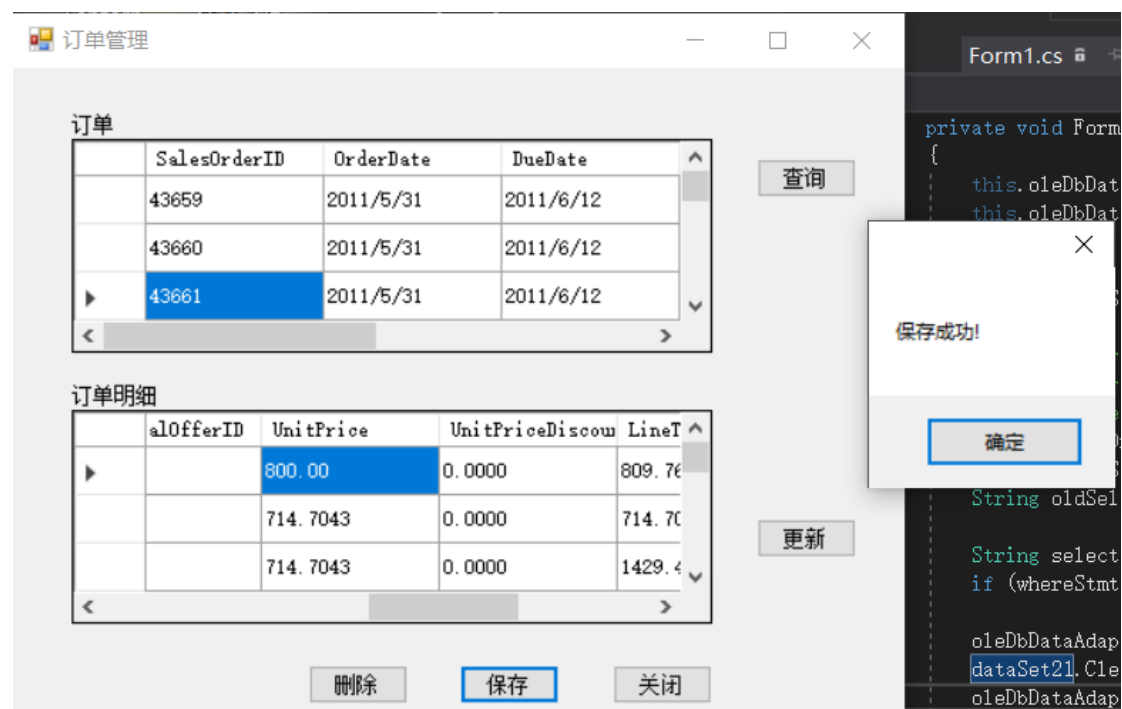
保存按钮设计代码

```

92     private void btnSave_Click(object sender, EventArgs e)
93     {
94         try
95         { //检查所有行rowguid列是否为空
96             foreach (DataRow dataRow in this.dataSet21.SalesOrderDetail.Rows)
97             {
98                 if (dataRow.RowState == DataRowState.Added)
99                 {
100                     if (dataRow["rowguid"].Equals(System.DBNull.Value))
101                     {
102                         dataRow["rowguid"] = Guid.NewGuid();
103                     }
104                 }
105             }
106             this.oleDbDataAdapter2.Update(this.dataSet21.SalesOrderDetail);
107             MessageBox.Show("保存成功!");
108         }
109         catch (Exception ex)
110         {
111             MessageBox.Show("保存失败!\n" + ex.Message);
112         }
113     }

```

保存成功，数据集中变化的数据写入数据库



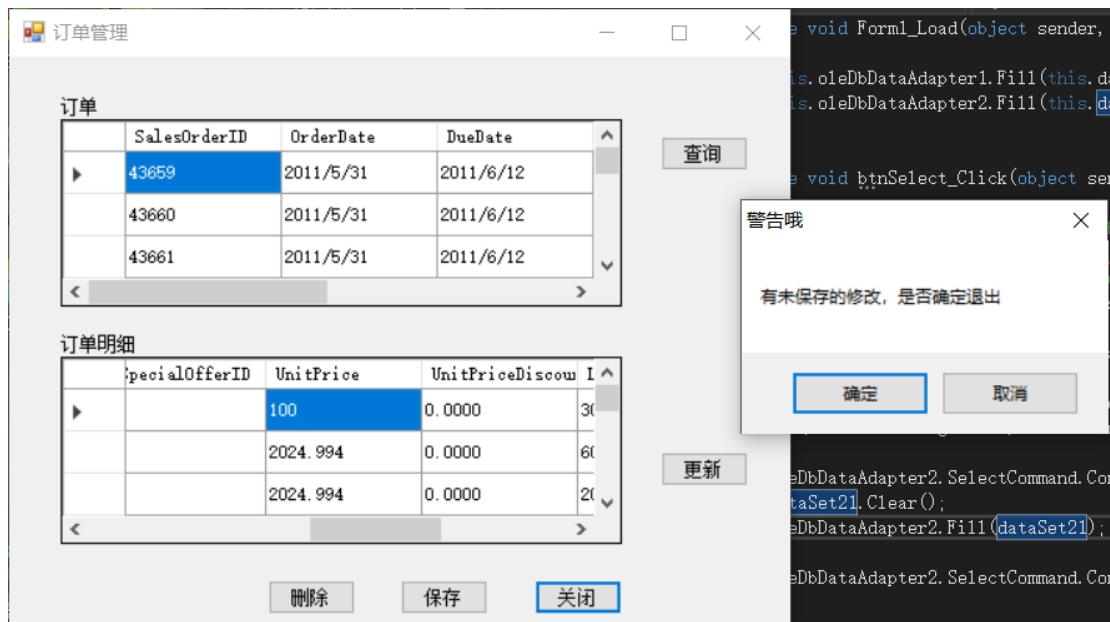
关闭按钮


```

115 private void btnClose_Click(object sender, EventArgs e)
116 {
117     if (dataSet21.HasChanges()) //判断有无未保存的修改
118     {
119         //弹出是否确认退出对话框
120         MessageBoxButtons warn = MessageBoxButtons.OKCancel;
121         DialogResult res = MessageBox.Show("有未保存的修改, "+ "是否确定退出", "警告哦", warn);
122         if (res == DialogResult.OK)
123         {
124             Application.Exit();
125         }
126         else
127         {
128             return;
129         }
130     } //若无未保存修改直接退出
131     else
132     {
133         Application.Exit();
134     }
135 }

```

若还有未保存的修改，则会跳出确认对话框，否则直接关闭



实验三：

实验内容：

使用示例数据库 AdventureWorks 完成购物车的增、删、改、查功能。

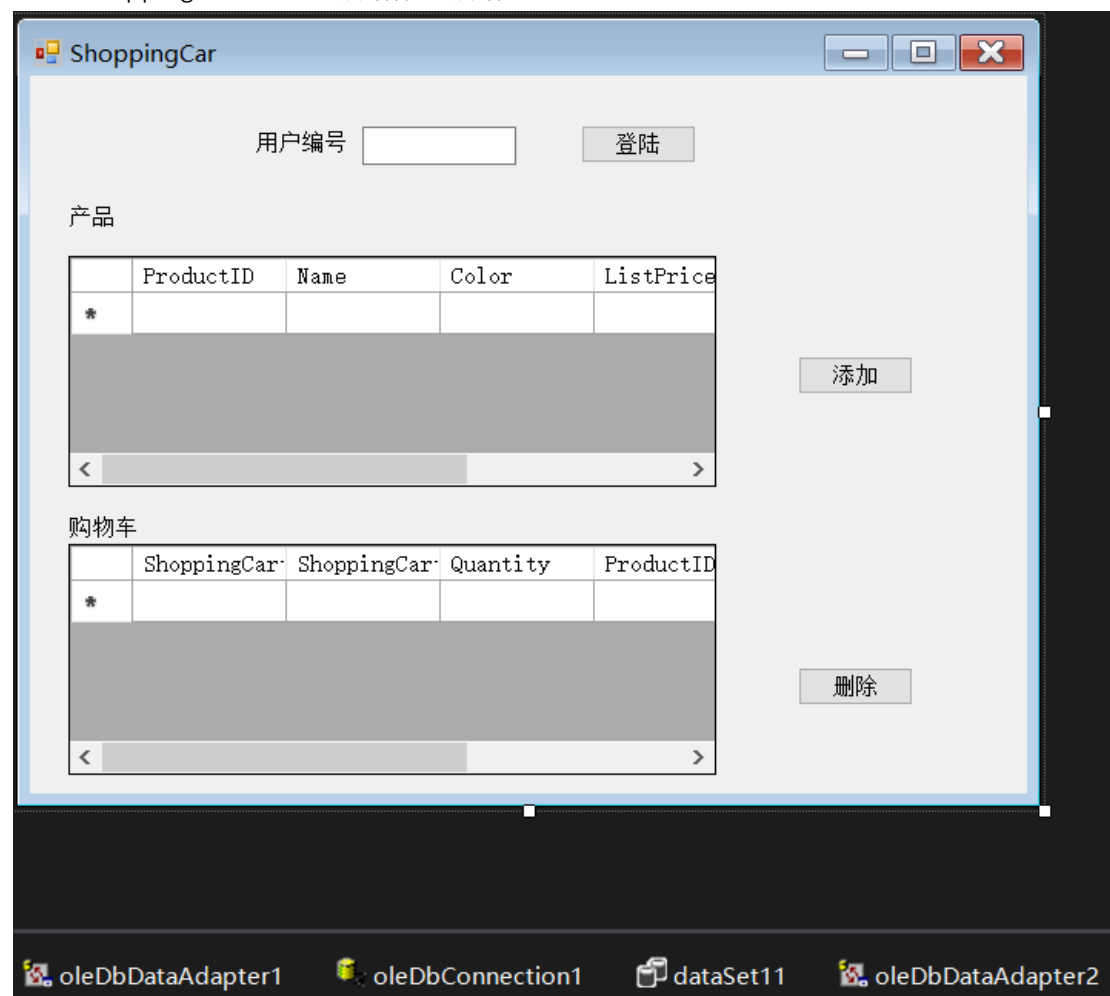
实验方法：

1. 本实验采用 OLEDB 类数据提供程序，
2. 本实验使用的连接对象是 OleDbConnection1，通过数据适配器连接，生成的连接对象中的连接字符串是 Provider=SQLNCLI11;Data Source=localhost;Integrated Security=SSPI;Initial Catalog=AdventureWorks，其中 SQLNCLI11 是客户端组件，localhost 是服务器，SSPI 是身份验证方式，AdventureWorks 是初始化数据库。
3. 本实验中使用了两个数据适配器 OleDbDataAdapter1 和 OleDbDataAdapter2，查询语句分别是 SELECT ProductID, Name, Color, ListPrice, Weight, (SELECT SUM(Quantity) AS Expr FROM Production.ProductInventory AS I WHERE (I.ProductID = P.ProductID)) AS Quantity FROM Production.Product AS P; 和 SELECT * FROM Sales.ShoppingCartItem;

4. 本实验使用的数据集对象为 dataSet11 和 dataSet21,dataSet11 中有 Production.Product 表, 由 OleDbDataAdapter1 生成, dataSet21 中有 Sales.ShoppingCartItem 表, 由 OleDbDataAdapter2 生成.

实验过程:

首先通过拖动控件设计出下图所示的界面, 包括一个 label 控件, 两个 dataGridView 控件(产品和购物车)和四个 Button 控件 (具体设计过程参考《数据库系统开发实验指导 3.1》的 7.1.2 节), 然后创建两个数据适配器 OleDbDataAdapter1 和 OleDbDataAdapter2, 对两个数据适配器分别生成一个数据集 dataSet11 和 dataSet21,然后订单表引用 dataSet11 中的填充的 Production.Product 的数据作为数据源, 订单明细表把 dataSet21 中的填充的 Sales.ShoppingCartItem 的数据作为数据源。



还未登陆时, 只填充 dataSet11

```
19 private void Form1_Load(object sender, EventArgs e)
20 {
21     OleDbDataAdapter1.Fill(dataSet11);
22 }
```

界面一开始只显示产品列表

ShoppingCar

用户编号

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

购物车

	ShoppingCartIter	ShoppingCartID	Quantity
*			

登陆按钮的代码，根据输入的用户 ID 构造查询语句，并根据查询语句填充数据集 dataSet21

```

23 private void btnLogIn_Click(object sender, EventArgs e)
24 {
25     if (tbxUserID.Text.Length == 0)
26     {
27         MessageBox.Show("input your userID!");
28     }
29     else
30     {
31         int user_id = Int32.Parse(tbxUserID.Text);
32         //修改查询命令
33         string whereStmt = " WHERE ShoppingCartID=" + user_id;
34         String oldSelectStmt = oleDbDataAdapter2.SelectCommand.CommandText;
35         String selectStmt = oleDbDataAdapter2.SelectCommand.CommandText;
36         selectStmt += whereStmt;
37         oleDbDataAdapter2.SelectCommand.CommandText = selectStmt;
38         //通过新的查询命令填充购物车列表对应的数据集
39         dataSet21.Clear();
40         oleDbDataAdapter2.Fill(dataSet21);
41         //将查询命令恢复
42         oleDbDataAdapter2.SelectCommand.CommandText = oldSelectStmt;
43     }
44 }

```

ShoppingCar

用户编号

14951

登陆

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

添加

购物车

	ShoppingCartIter	ShoppingCartID	Quantity
▶	2	14951	3
•			

删除

添加产品到购物车代码

```

45 private void btnAdd_Click(object sender, EventArgs e)
46 {
47     //获取用户ID
48     int user_id = Int32.Parse(tbxUserID.Text);
49     //获取产品ID
50     int product_id = (int)dataSet11.Product[dgvProduct.CurrentRow.Index]["ProductID"];
51     OleDbCommand command = new OleDbCommand();
52     command.Connection = this.oleDbConnection1;
53     this.oleDbConnection1.Open();
54     OleDbTransaction trans = OleDbConnection1.BeginTransaction();
55     //创建一个事务
56     try
57     {
58         command.Transaction = trans;
59         string command_text = "EXEC add_to_shoppingcart " + product_id + "," + user_id;
60         command.CommandText = command_text;
61         command.ExecuteNonQuery();
62         trans.Commit();
63     }
64     catch (Exception ex)
65     {
66         trans.Rollback(); //如果出错则回滚
67         MessageBox.Show(ex.ToString());
68     }
69     finally
70     {
71         this.oleDbConnection1.Close();
72     }
73     btnLogIn_Click(sender, e);
74 }

```

将 ProductID 为 1 的产品加入购物车

ShoppingCar

用户编号14951 登陆

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

添加

购物车

	ShoppingCartID	Quantity	ProductID
▶	14951	3	862
	14951	1	1
•			

删除

再加入两次，Quantity 变为 3

ShoppingCar

用户编号14951 登陆

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

添加

购物车

	ShoppingCartID	Quantity	ProductID
	14951	3	862
	14951	3	1
▶*			

删除

将产品从购物车移除代码

```
76 private void btnRemove_Click(object sender, EventArgs e)
77 {
78     //获取用户ID
79     int user_id = Int32.Parse(tbxUserID.Text);
80     //获取要移出的产品ID
81     int product_id = (int)dataSet11.Product[dgvProduct.CurrentRow.Index]["ProductID"];
82     OleDbCommand command = new OleDbCommand();
83     command.Connection = this.oleDbConnection1;
84     this.oleDbConnection1.Open();
85     OleDbTransaction trans = OleDbConnection1.BeginTransaction();
86     //构造一个事务
87     try
88     {
89         command.Transaction = trans;
90         string command_text = "EXEC remove_from_shoppingcart " + product_id + "," + user_id; command.CommandText = command_text;
91         command.ExecuteNonQuery();
92         trans.Commit();
93     }
94     catch (Exception ex)
95     {
96         trans.Rollback(); //如果出错则回滚
97         MessageBox.Show(ex.ToString());
98     }
99     finally
100     {
101         this.oleDbConnection1.Close();
102     }
103     btnLogin_Click(sender, e);
104 }
```

将 ProuctID 为 862 的产品从购物车中移除两次，Quantity 变为 1

ShoppingCar

用户编号 14951 登陆

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

添加

购物车

	ShoppingCartID	Quantity	ProductID
▶	14951	1	862
	14951	3	1
•			

删除

再移除一次，该产品消失

ShoppingCar

用户编号14951

登陆

产品

	ProductID	Name	Color
▶	1	Adjustable Race	
	2	Bearing Ball	
	3	BB Ball Bearing	

添加

购物车

	ShoppingCartID	Quantity	ProductID
▶	14951	3	1
*			

删除