## 实验四

一、实验方法

1.问题：

(1)使用哪种数据提供程序？

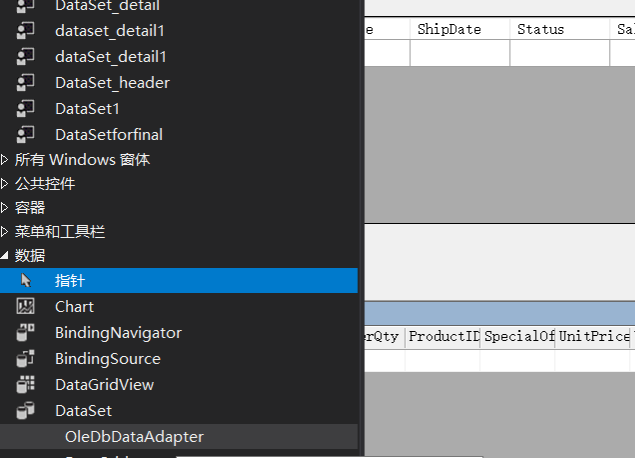
使用OLEDB .NET Framework作为数据提供程序

(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？

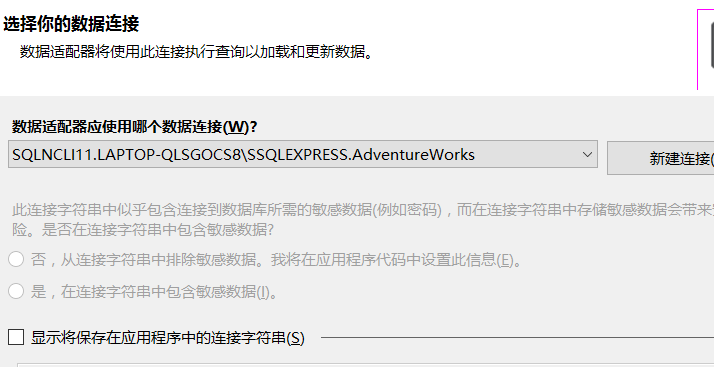
利用oleDbConnection1作为数据连接对象

连接对象的建立过程：

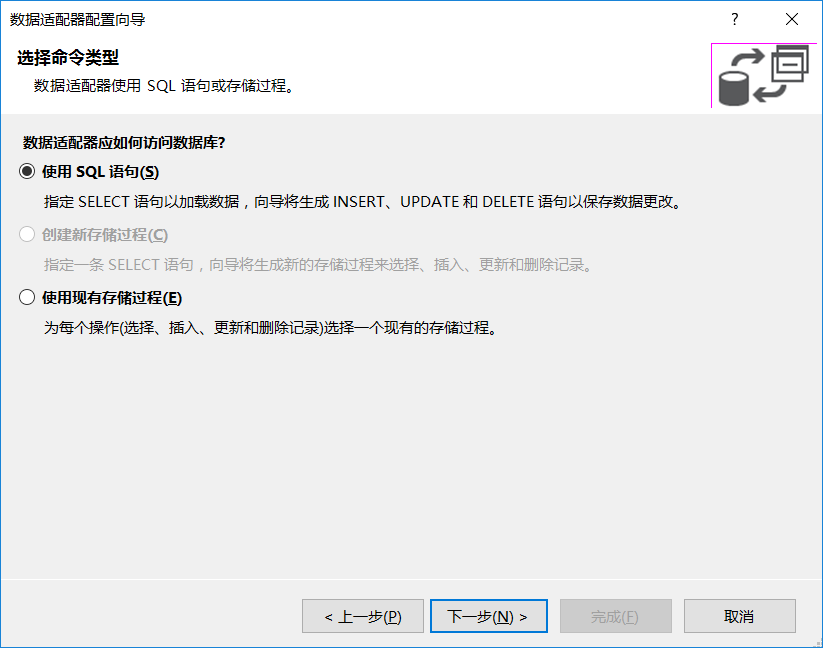
1)打开工具箱，选择“数据”一栏的OleDbDataAdapter，并拖动到窗口form上

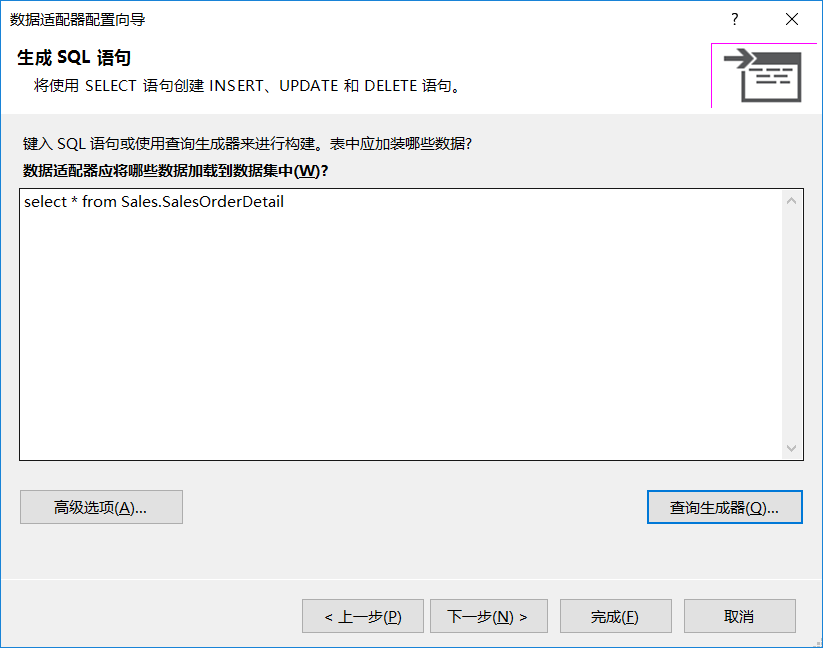


2)触发“数据适配器向导”，我们开启了数据适配器



3)设置使用SQL语句进行连接





Provider=SQLNCLI11;数据提供器

Data Source=LAPTOP-QLSGOCS8\SSQLEXPRESS;数据库实例名

Integrated Security=SSPI;使用微软的安全协议

Initial Catalog=AdventureWorks”。使用adventureworks数据库

1. 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？

oleDbDataAdapter1：

SELECT Sales.SalesOrderDetail.\* FROM Sales.SalesOrderDetail

oldDbDataAdapter2:

SELECT SalesOrderID, OrderDate, DueDate, ShipDate, Status, SalesOrderNumber FROM Sales.SalesOrderHeader”。

1. 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）。
2. dataSetforfinal2 , dataSetforfinal1。
3. 数据集中有表“SalesOrderDetail”和表“SalesOrderHeader”。“SalesOrderDetail”表是由适配器“oleDbDataAdapter1”生成的，“SalesOrderHeader”表是由适配器“oleDbDataAdapter2”生成的。

3.手工添加的代码及说明：

1）数据适配器加载函数，在加载的时候现实我们的填充效果

Private void Form\_load(object sender , EventArgs e)

{

This.oleDbdataAdapter1.Fill(this.datasetforme.salesorderdetail);

This.oleDbdataAdapter1.Fill(this.datasetforme.salesorderHeader);

}

2）删除对应的行

Private void button\_click(object sender , Eventargs e)

{

Int index = -1;

Foeach(datarow datarow in datasetforme.salesorderdetail.row)

{

Datarowstate rowstate = datarow.rowstate;

If(rowstate == datarowstate.deleted) continue;

Index ++;

If(this.dataGrid2.currentrowindex == index)

{

Datarow.delete();

Break;

}

}

}

1. 保存行

Private void button2\_click(object sender , eventargs e)

{

try

{

Foreach(datarow datarow in this.datasetforme.salesorderdetail.rows)

{

If(datarow.rowstate == datarowstate.added)

{

If(datarow[‘rowguid’].equals(system.dbnull.values))

{

Datarow[‘rowguid’] = guid.nowguid();

}

}

}

this.oleDbDataAdapter.Update(this.dataSetforme.SalesOrderDetail);

Messagebox.show(“save successfully!”)

}

Catch(Exception)

{

Message.show(“save failed!”)

}

}

1. 关闭按钮

Private void button3\_click(object sender , eventargs e)

{

Int flag = 0;

Try

{

Foreach(datarow datarow in this.dataforme.salesorderdetail.row)

{

If(datarow.rowstate == datarowstate.added || datarow.rowstate == datarowstate.deleted || datarow.rowstate == datarowstate.modified)

{

Flag = 1;

Break;

}

}

If(flag == 1)

{

Dialogresult result = messagebox.show(“still not save!”);

If(result == dialogresult.OK)

{

Application.exit();

}

}

Else

{

Application.exit();

}

}

Catch(exception e)

{

Messagebox.show(“close failed!\n”);

}

}

5）查询按钮，在其中对我们的查询的语句进行了替换，需要注意我们的替换之后需要替换回来

Private void salesorderheaderdatagridview\_cellcontentclick(object sender , datagridviewcelleventargs e)

{

String wherestmt = “”;

Wherestmt += “salesorderid=” + salesorderheaderdatagridview.row[salesorderheaderdatagridview.currentrow.index].cells[0].value.tostring() + “”;

String oldsalestmt = oledbdataadapter.selectcommand.commandtext;

String selectstmt = oledbdataadapter.selectcommand.commandtext;

Selectstmt += “where” + wherestmt;

Oledbdataadapter.selectcommand.commandtext = selectstmt;

Datasetforme.salesorderdetail.clear();

Oledbdataadapter.Fill(datasetforme.salesorderdetail);

Oledbdataadapter.selectcommand.commandtext = oldselectstmt;

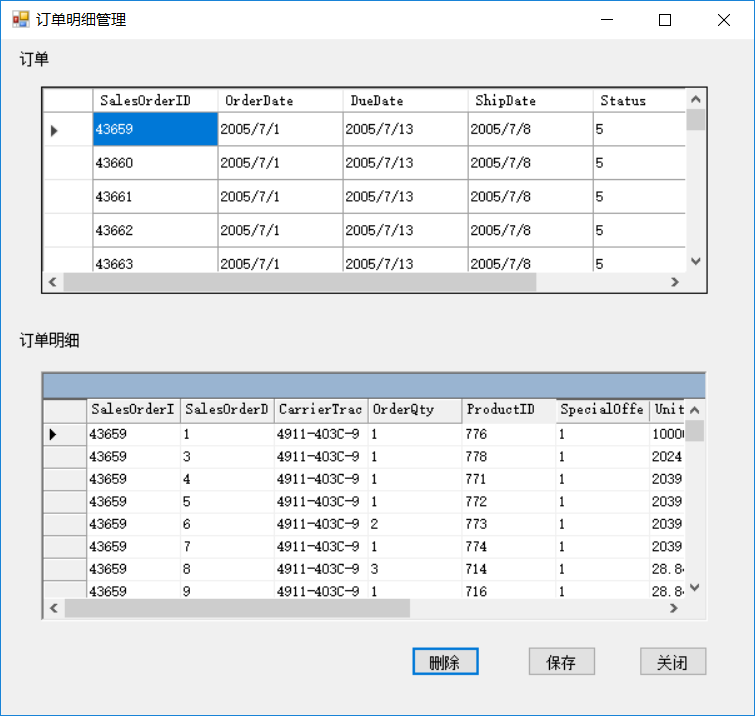
}

二、实验结果

1．程序运行界面。除了我们的datagrid的界面之外，还存在有

1. 删除
2. 保存
3. 关闭

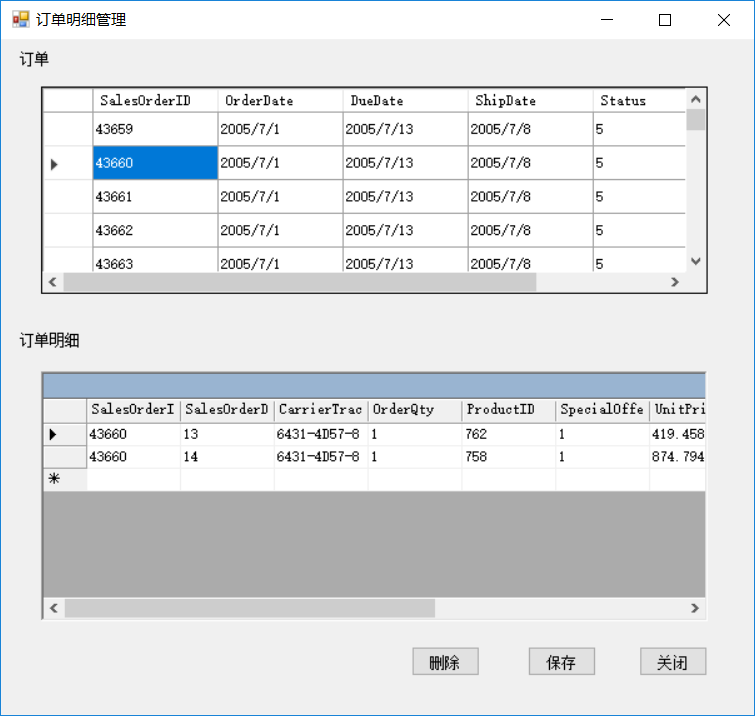
这三个按钮，这三个按钮实现应用 的三个不停的功能

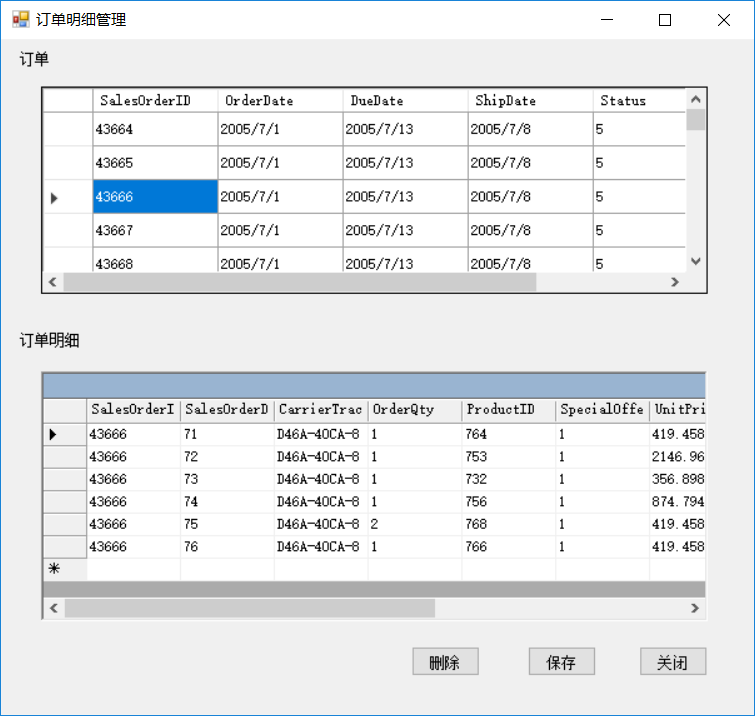


2．操作的简单说明。

1）查询功能：

1.订单明细表中的内容发生相应的变化。分别点击订单表“SalesOrderID = 43660”的行和“SalesOrderID = 43666”的行，我们的订单和订单明细两个表中间其实是不太一样的，我们点击订单之后会在订单明细中显示我们的订单的详细情况

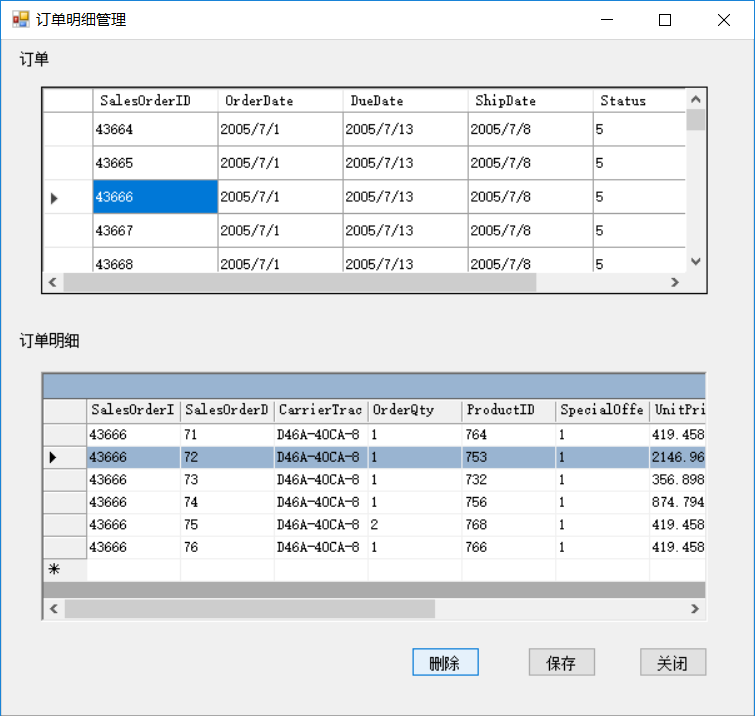




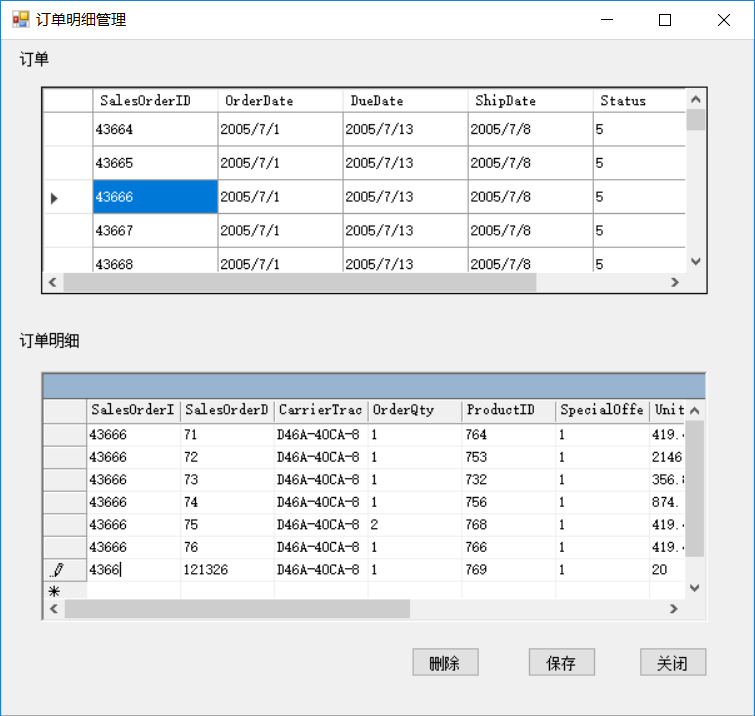
1. 删除功能的实现

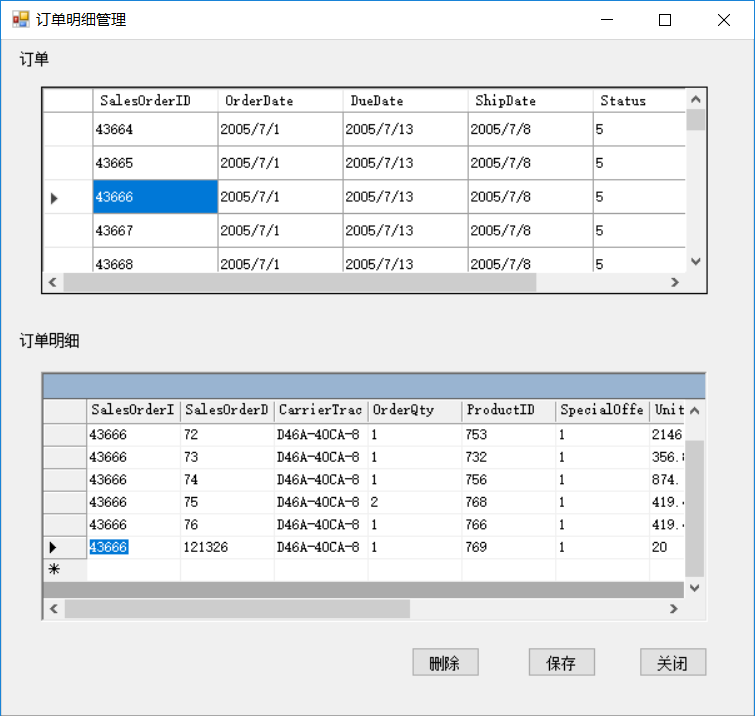
1.选取订单明细中的一行，点击删除，选择的订单明细从表中删除，但并不真正从数据库中删除。

2.删除订单明细表中“SalesOrderID = 43660，SalesOrderDetailID = 72”的行。点击“删除”按钮后。



1. 插入功能





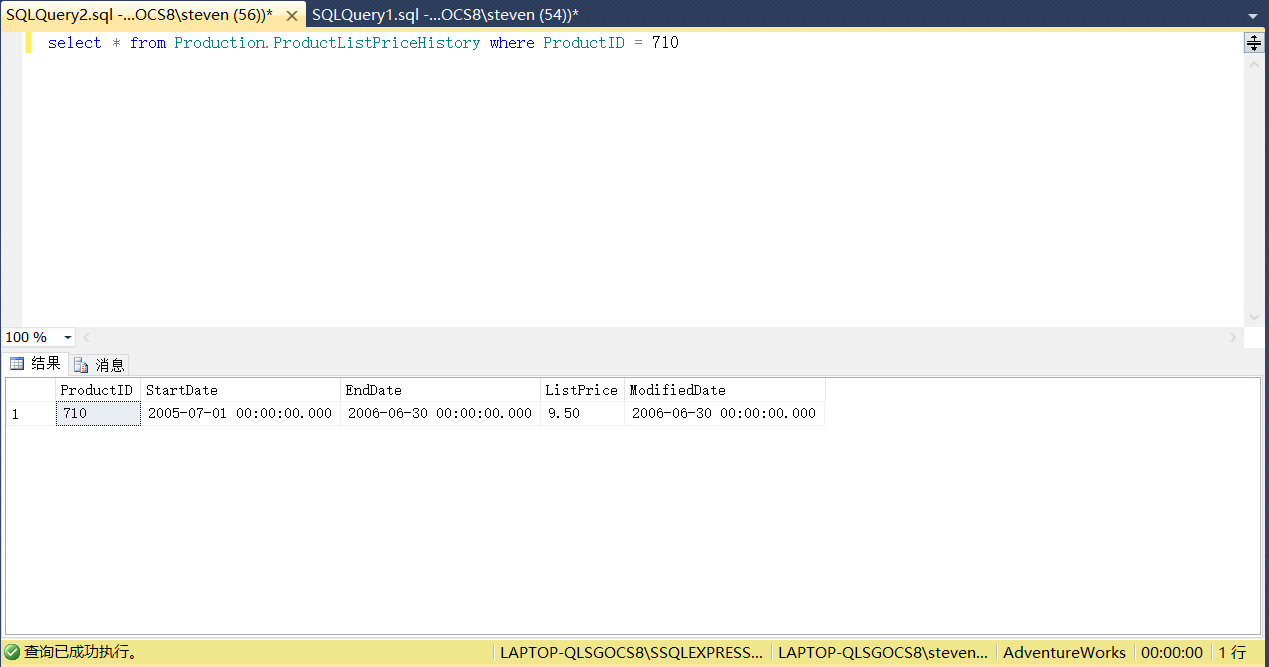
1. 更新功能的

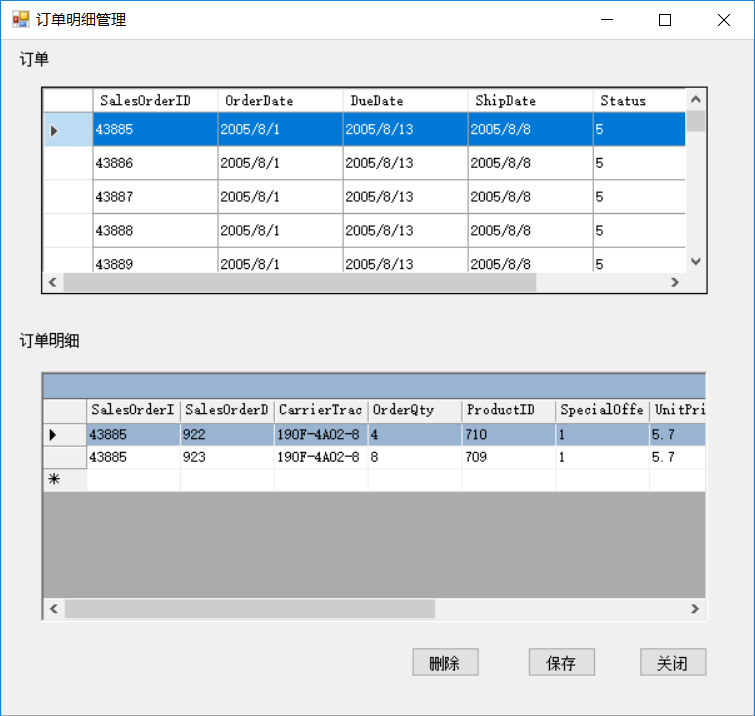
1. 可以修改订单明细表中的某一行的内容

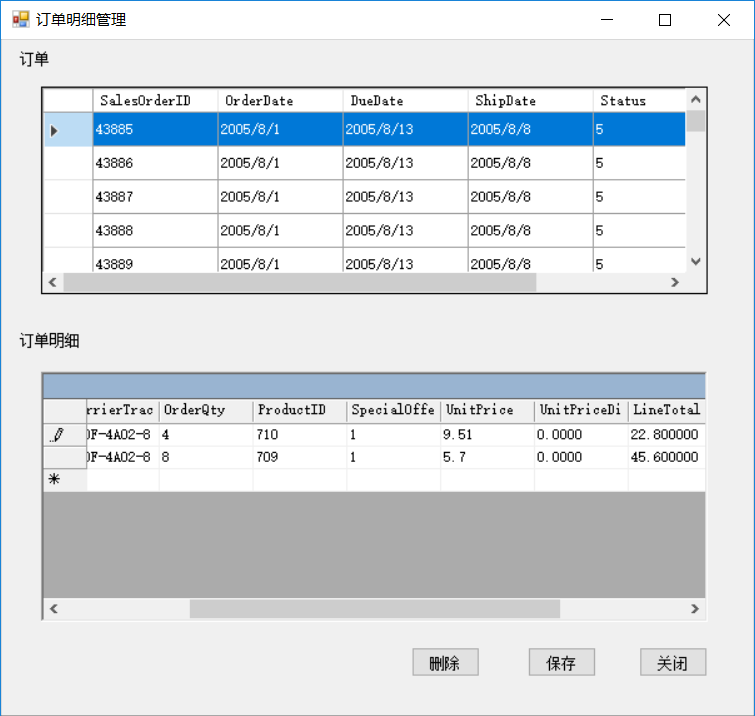
2. 得到Production.ProductListPriceHistory表的“ProductID == 710”的产品的历史价格和初始及结束时间

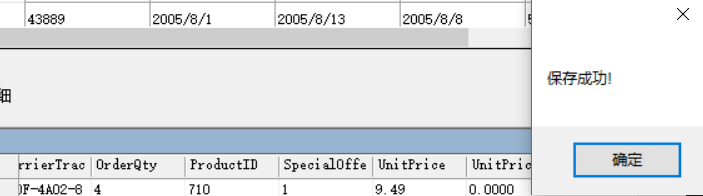
3.得到Sales.SalesOrderDetail表里“ProductID = 710”的产品的SalesOrderID和ModifiedDate

4.保存成功。









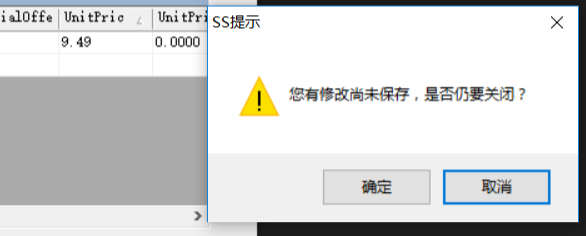
1. 保存功能

点击保存按钮时，将订单明细保存到数据库。。

1. 关闭功能

1.功能说明 : 点击关闭按钮时，如果订单明细有修改但没有保存，则提示是否关闭，如果不关闭则返回，否则关闭程序，如果没有修改，则直接关闭

2.下面截图显示没有保存的时候的关闭提醒机制



三、实验总结

1.实验过程分析：

经常在参考老师的实验报告来完成实验，但是实际发现，老师的说明和实验的实际上上还是存在很多的不一样支出需要实际的实验才可以发现，在实验过程中，尤其是在最后的保存按钮的实现上，我和周围的同学交流病查阅了相关的参考的官方文档最终才勉强得以完成相关的实验

2.改进的地方：

该应用的完成的时候，没有根据时间的要求来对我们的订单和合同进行查询，个人认为还可以在查询方面添加利用合同的日期来进行查询的思路

可以考虑的查询策略

1. 日期查询
2. 甲方查询
3. 乙方查询