

数据库系统开发实验

实验二、触发器的设计与测试

- 班级：07111606
- 学号：1120161881
- 实验人：张启洋
- 实验日期：2018/09/14 - 2018/09/16

零、目录

数据库系统开发实验

实验二、触发器的设计与测试

零、目录

一、实验目的

二、实验环境

三、实验内容

(一) 建立修改信息记录表Production.ProductUpdateLog

(二) 建立存储过程Production.UpdatePrice

(三) 在表Sales.PriceUpdateTr建立触发器Sales.SalesOrderDetail

(四) 触发器测试

四、实验总结

(一) 总结

2018/09/16 19:42 实验人：张启洋 学号：1120161881

一、实验目的

通过该实验加深存储过程和触发器的基本概念的理解，掌握基本存储过程、触发器的管理和执行，能够使用游标编写较复杂的存储过程，为后面的系统开发打好基础

二、实验环境

- 服务器和实例名称：DWQETHINKPAD
- 操作系统：Windows 10 Home China
- 数据库服务器：Microsoft SQL Server 2017 Developer Edition
- 数据库客户端：SQL Server Management Studio v17.8.1

三、实验内容

（一）建立修改信息记录表Production.ProductUpdateLog

中文说明	列名	数据类型
记录编号	UpdateID	int
订单编号	SalesOrderID	int
订单明细编号	SalesOrderDetailID	int
产品编号	ProductID	int
产品的公开报价	ListPrice	money
修改前产品的单价	OldPrice	money
修改后产品的单价	NewPrice	money
修改者的登录名	LogName	varchar(50)

执行SQL语句，编号以及登录名设置为非空

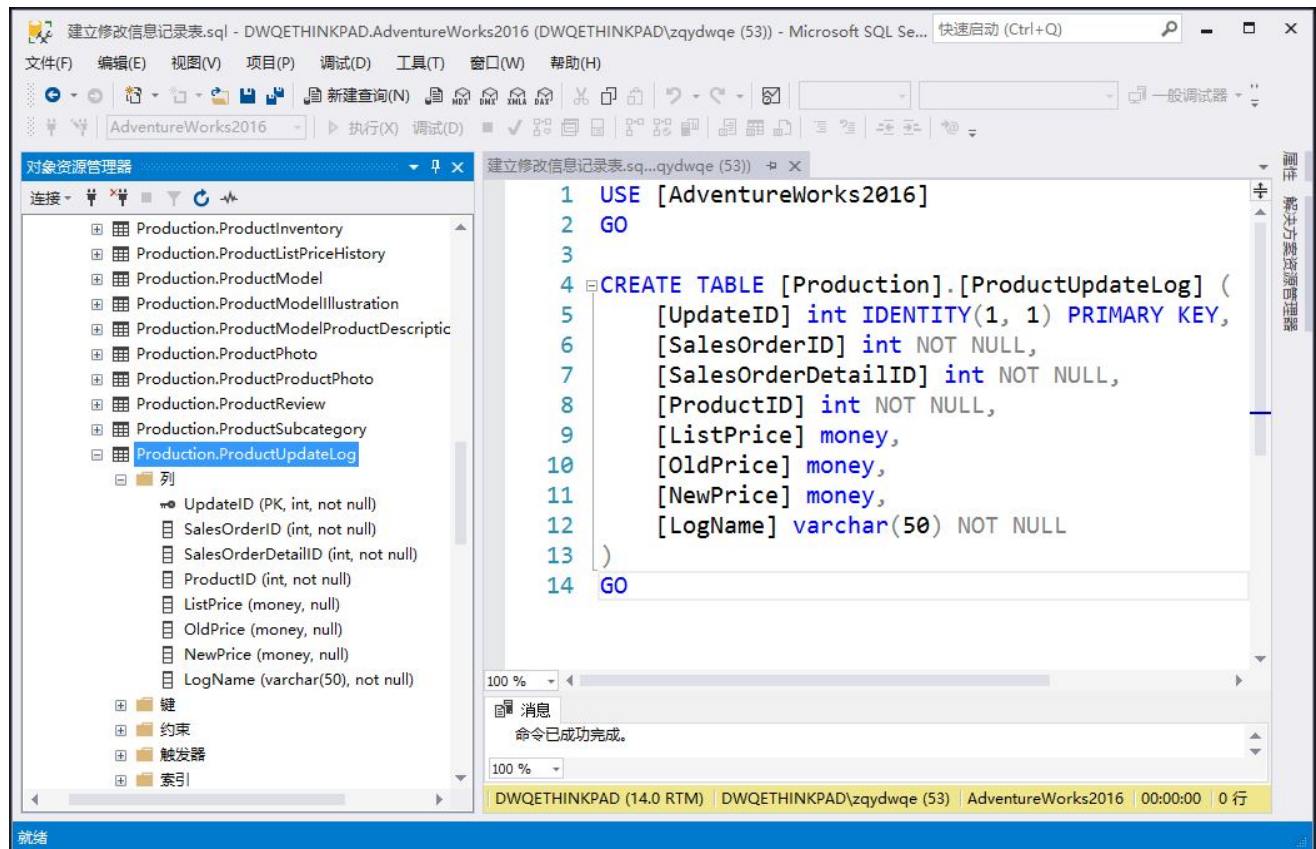
```
1 USE [AdventureWorks2016]
2 GO
3
4 CREATE TABLE [Production].[ProductUpdateLog] (
5     [UpdateID] int IDENTITY(1, 1) PRIMARY KEY,
6     [SalesOrderID] int NOT NULL,
7     [SalesOrderDetailID] int NOT NULL,
8     [ProductID] int NOT NULL,
9     [ListPrice] money,
10    [OldPrice] money,
```

```

11     [NewPrice] money,
12     [LogName] varchar(50) NOT NULL
13 )
14 GO

```

使用IDENTITY语句将UpdateID设置为自增序列，并设置为主键



(二) 建立存储过程Production.UpdatePrice

执行SQL语句，建立存储过程，方便向修改信息记录表中插入数据

```

1 USE [AdventureWorks2016]
2 GO
3
4 IF OBJECT_ID('[Production].[UpdatePrice]', 'P') IS NOT NULL
5     DROP PROCEDURE [Production].[UpdatePrice]
6 GO
7
8 CREATE PROCEDURE [Production].[UpdatePrice]
9     @SalesOrderID int, @SalesOrderDetailID int, @ProductID int,
10     @ListPrice money, @OldPrice money, @NewPrice money,
11     @LogName varchar(50)
12 AS
13 INSERT INTO [Production].[ProductUpdateLog] (
14     [SalesOrderID], [SalesOrderDetailID], [ProductID],
15     [ListPrice], [OldPrice], [NewPrice],
16     [LogName]

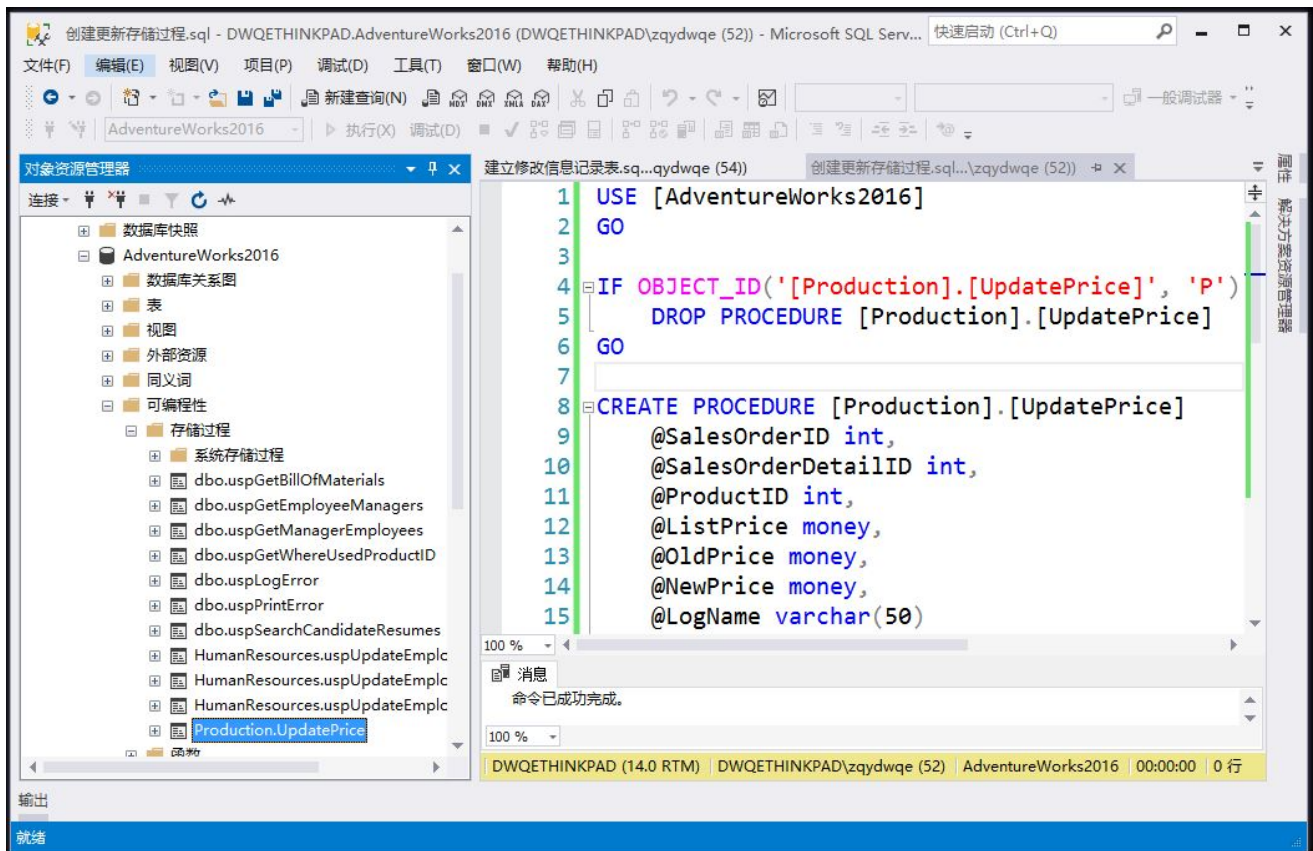
```

```

17 )
18 VALUES (
19     @SalesOrderID, @SalesOrderDetailID, @ProductID,
20     @ListPrice, @OldPrice, @NewPrice,
21     @LogName
22 )
23 GO

```

重新连接服务器后可以在左侧对象资源管理器的AdventureWorks2016数据库的可编程性-存储过程菜单下看到新创建的存储过程Production.UpdatePrice



(三) 在表Sales.PriceUpdateTr建立触发器Sales.SalesOrderDetail

由题目要求，修改订单明细表Sales.SalesOrderDetail中的记录时，如果修改记录中的产品单价UnitPrice大于产品公开报价（Production.ProductListPriceHistory中的ListPrice），则不能进行修改并抛出错误信息，否则，进行修改并将修改的有关信息写到Production.ProuctUpdateLog表中

执行SQL语句，在数据表Sales.SalesOrderDetail创建触发器Sales.PriceUpdateTr，首先删去已创建的重名触发器，然后创建INSTEAD OF触发器，根据题目要求，使用IF-ELSE语句，设定如果修改记录中的产品单价大于产品公开报价，则不能进行修改并使用RAISERROR抛出错误信息，否则，进行修改并将修改的有关信息写到Production.ProuctUpdateLog表中

```

1 USE [AdventureWorks2016]
2 GO
3 --删除已存在重名触发器
4 IF OBJECT_ID('[Sales].[PriceUpdateTr]', 'TR') IS NOT NULL

```

```

5      DROP TRIGGER [Sales].[PriceUpdateTr]
6  GO
7  --在表[Sales].[PriceUpdateTr]的UPADATE操作上, 创建架构为[Sales]的INSTEAD OF触发器
   [SalesOrderDetail]
8  CREATE TRIGGER [Sales].[PriceUpdateTr] ON [Sales].[SalesOrderDetail]
9  INSTEAD OF UPDATE
10 AS
11 --如果插入的UnitPrice大于公开报价ListPrice, 使用RAISERROR抛出错误信息, 并进行回滚
12 IF EXISTS (
13     SELECT [inserted].[UnitPrice]
14     FROM [Production].[ProductListPriceHistory], [inserted]
15     WHERE [inserted].[UnitPrice] > [Production].[ProductListPriceHistory].[ListPrice] AND
16           [inserted].[ProductID] = [Production].[ProductListPriceHistory].[ProductID] AND
17           [inserted].[ModifiedDate] >= [Production].[ProductListPriceHistory].[StartDate] AND
18           [inserted].[ModifiedDate] <= [Production].[ProductListPriceHistory].[EndDate]
19 )
20 BEGIN
21     RAISERROR('修改记录中产品单价UnitPrice不能大于产品公开报价ListPrice', 7, 1)
22     ROLLBACK TRANSACTION
23 END

```

```

1  --如果符合要求, 修改数据, 并调用存储过程插入数据
2  ELSE
3      BEGIN
4          DECLARE
5              @SalesOrderID int, @SalesOrderDetailID int, @ProductID int,
6              @ListPrice money, @OldPrice money, @NewPrice money,
7              @LogName varchar(50)
8
9          SELECT @SalesOrderID = [SalesOrderID], @SalesOrderDetailID = [SalesOrderDetailID],
10             @ProductID = [ProductID], @NewPrice = [UnitPrice]
11          FROM [inserted]
12
13          SELECT @ListPrice = [ListPrice]
14          FROM [Production].[ProductListPriceHistory]
15          WHERE [Production].[ProductListPriceHistory].[ProductID] = (SELECT [ProductID] FROM
[inserted])
16          --修改数据
17          UPDATE [Sales].[SalesOrderDetail]
18          SET [UnitPrice] = @NewPrice
19          WHERE [Sales].[SalesOrderDetail].[SalesOrderID] = @SalesOrderID AND
20             [Sales].[SalesOrderDetail].[SalesOrderDetailID] = @SalesOrderDetailID
21
22          SELECT @OldPrice = [UnitPrice]
23          FROM [deleted]
24          WHERE [deleted].[ProductID] = (SELECT [ProductID] FROM [inserted])
25
26          SELECT @LogName = SYSTEM_USER
27          --执行存储过程, 插入数据
28          EXECUTE [Production].[UpdatePrice]
29              @SalesOrderID, @SalesOrderDetailID, @ProductID,
30              @ListPrice, @OldPrice, @NewPrice,

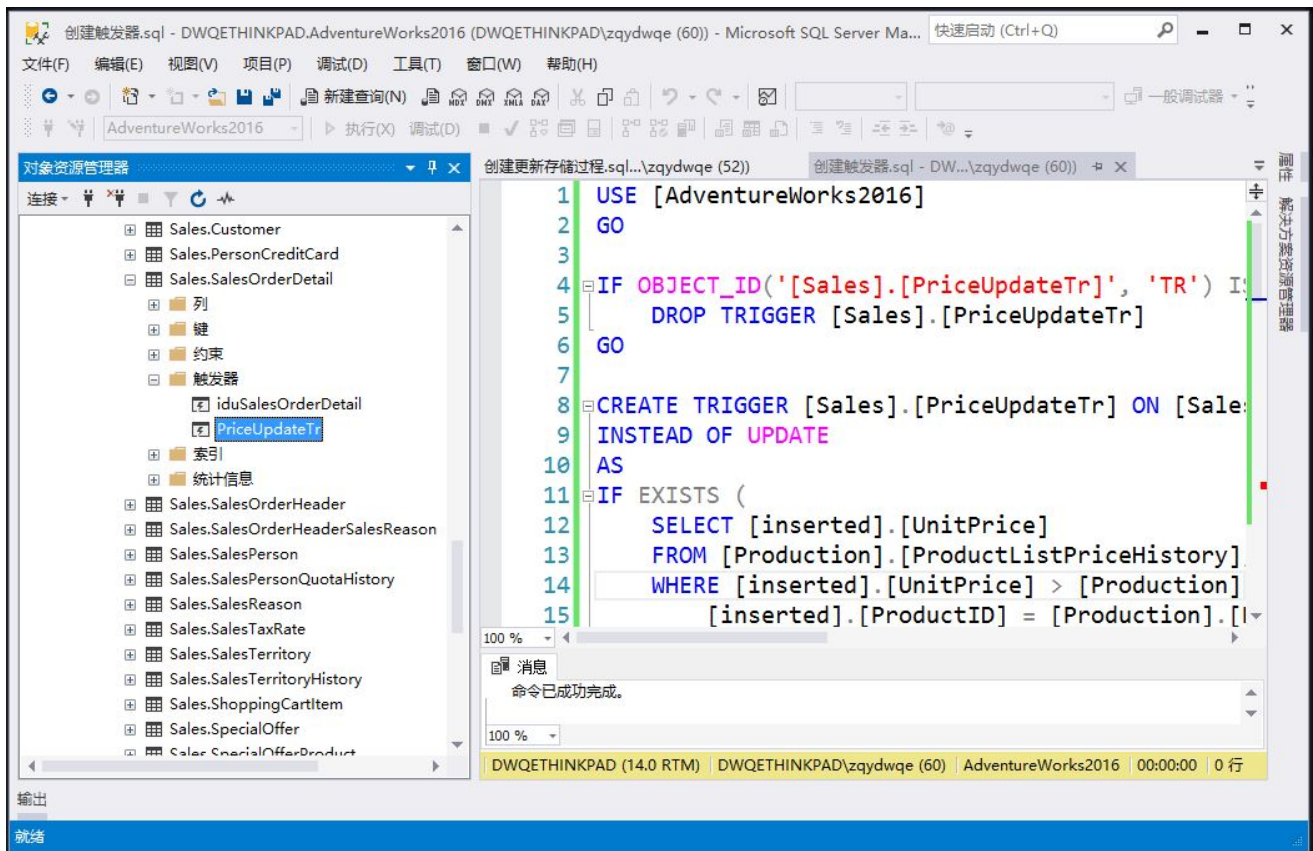
```

```

31         @LogName
32     END
33 GO

```

重新连接服务器，可在对象资源管理器中查看到创建的触发器



(四) 触发器测试

根据SalesOrderID和SalesOrderDetailID在表SalesOrderDetail中查找ProductID

触发器测试.sql - DWQETHINKPAD.AdventureWorks2016 (DWQETHINKPAD\zqydwqe (57)) - Microsoft SQL Server Ma... 快速启动 (Ctrl+Q)

文件(F) 编辑(E) 视图(V) 查询(Q) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)

对象资源管理器

- 连接
- DWQETHINKPAD (SQL Server 14.0.1000.169 - DWQETHINKPAD\zqydwqe (57))
 - 数据库
 - 系统数据库
 - 数据库快照
 - AdventureWorks2016
 - 安全性
 - 服务器对象
 - 复制
 - PolyBase
 - Always On 高可用性
 - 管理
 - Integration Services 目录
 - SQL Server 代理(已禁用代理 XP)
 - XEvent 探查器

触发器测试.sql - DW... \zqydwqe (57) 创建触发器.sql - DW... \zqydwqe (60)

```
1 USE [AdventureWorks2016]
2 GO
3
4 SELECT [SalesOrderID], [SalesOrderDetailID], [ProductID]
5 FROM [Sales].[SalesOrderDetail]
6 WHERE [SalesOrderID] = 43659 AND [SalesOrderDetailID] = 1
```

100 %

	SalesOrderID	SalesOrderDetailID	ProductID
1	43659	1	776

DWQETHINKPAD (14.0 RTM) | DWQETHINKPAD\zqydwqe (57) | AdventureWorks2016 | 00:00:00 | 1 行

输出

已保存的项

行 6 列 51 字符 51 Ins

根据ProductID在表ProductListPriceHistory查找UnitPrice, 发现价格为3374.99

触发器测试.sql - DWQETHINKPAD.AdventureWorks2016 (DWQETHINKPAD\zqydwqe (57)) - Microsoft SQL Server Ma... 快速启动 (Ctrl+Q)

文件(F) 编辑(E) 视图(V) 查询(Q) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)

对象资源管理器

- 连接
- DWQETHINKPAD (SQL Server 14.0.1000.169 - DWQETHINKPAD\zqydwqe (57))
 - 数据库
 - 系统数据库
 - 数据库快照
 - AdventureWorks2016
 - 安全性
 - 服务器对象
 - 复制
 - PolyBase
 - Always On 高可用性
 - 管理
 - Integration Services 目录
 - SQL Server 代理(已禁用代理 XP)
 - XEvent 探查器

触发器测试.sql - DW... \zqydwqe (57) 创建触发器.sql - DW... \zqydwqe (60)

```
1 USE [AdventureWorks2016]
2 GO
3
4 SELECT [ProductID], [ListPrice]
5 FROM [Production].[ProductListPriceHistory]
6 WHERE [ProductID] = 776
```

100 %

	ProductID	ListPrice
1	776	3374.99

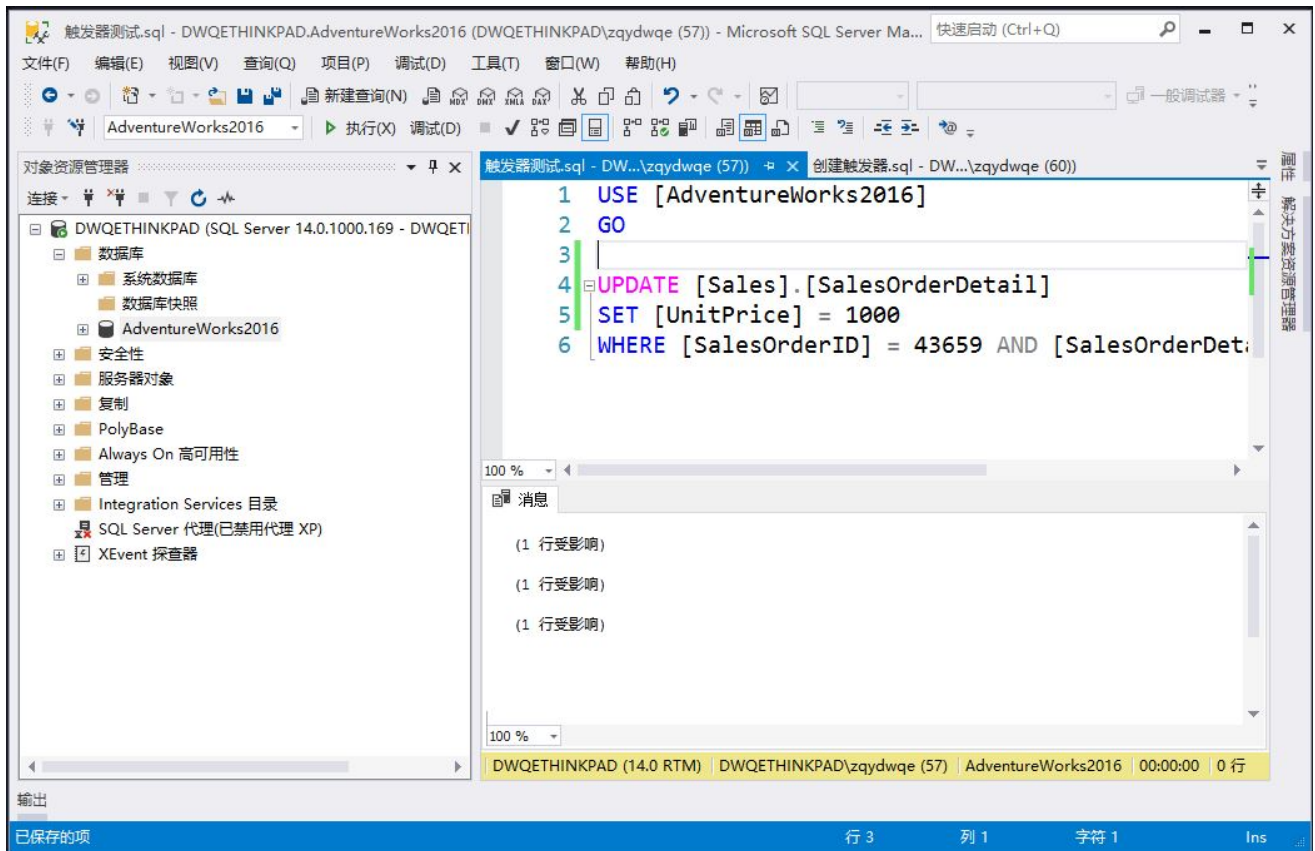
DWQETHINKPAD (14.0 RTM) | DWQETHINKPAD\zqydwqe (57) | AdventureWorks2016 | 00:00:00 | 1 行

输出

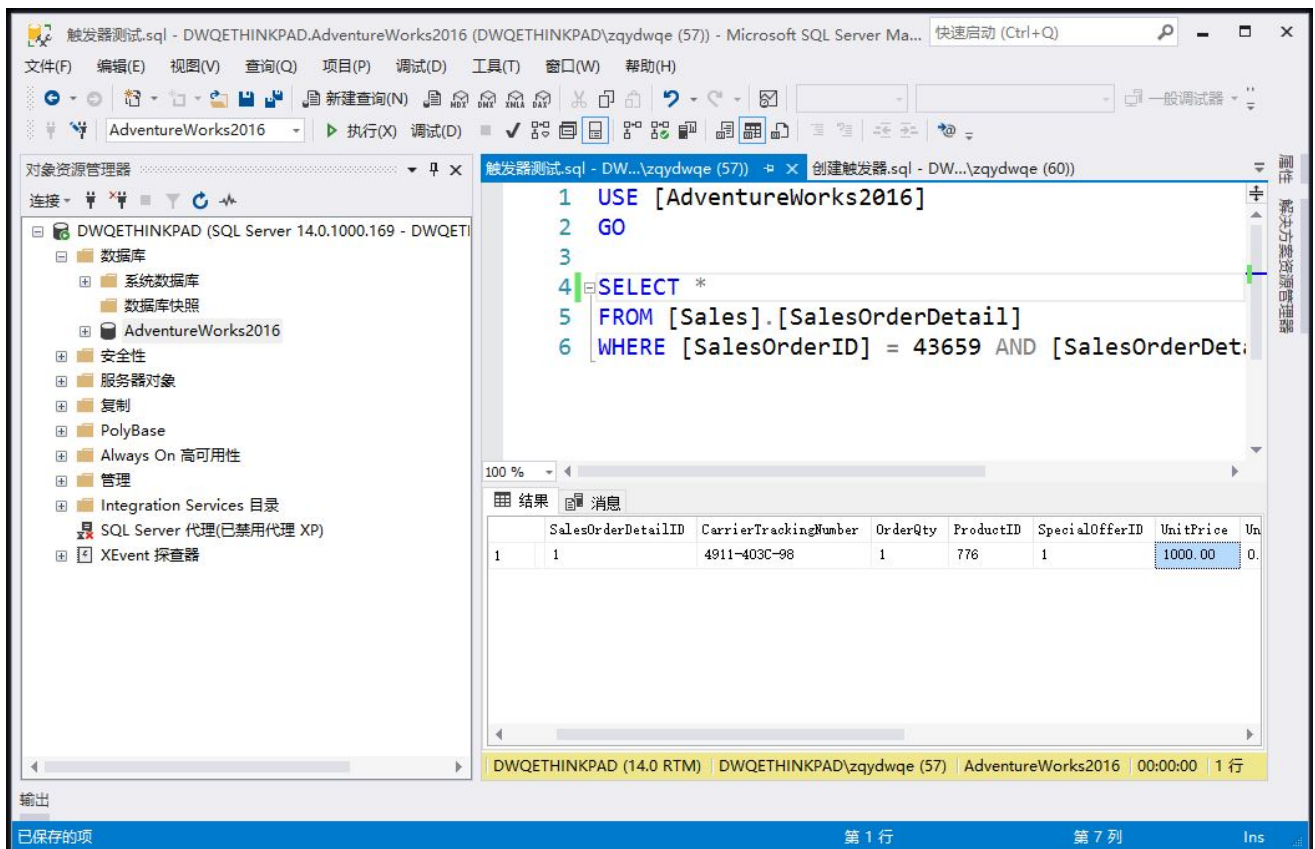
已保存的项

第 1 行 第 1 列 Ins

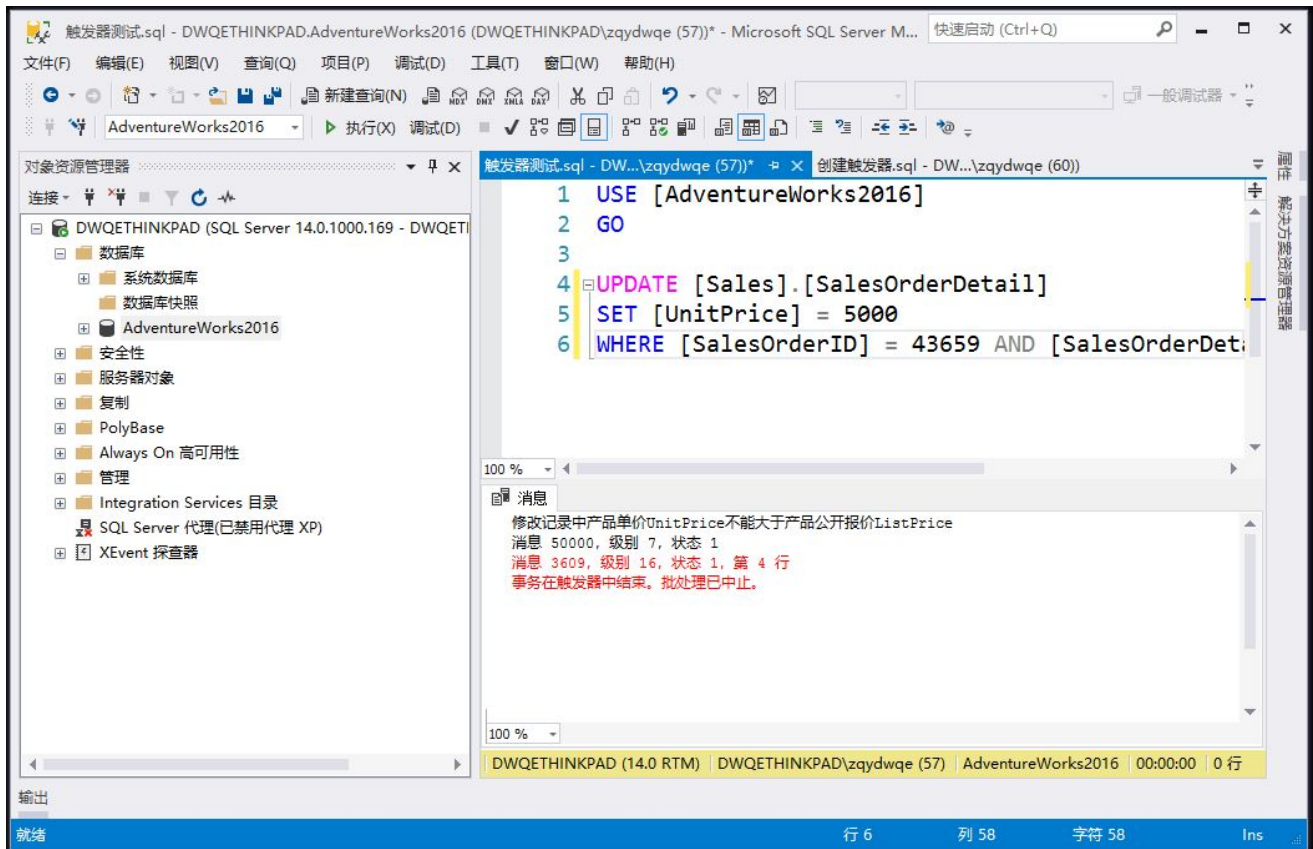
修改Sales.SalesOrderDetail中UnitPrice修改为1000, 修改操作成功



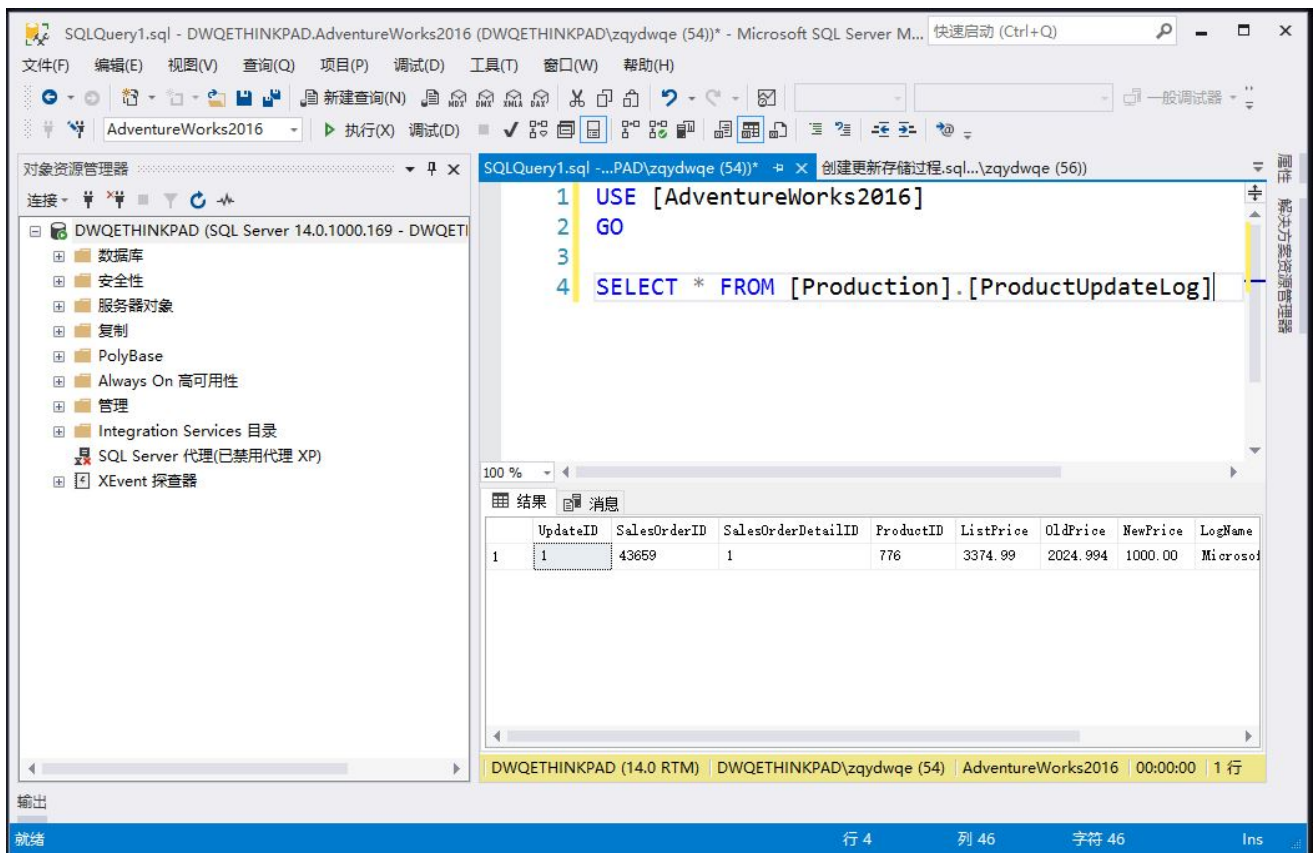
查询Sales.SalesOrderDetail中的信息，价格被成功修改



修改Sales.SalesOrderDetail中UnitPrice修改为5000，修改操作失败，RAISERROR抛出错误信息



查询Production.ProductUpdateLog表, 发现只有第一次修改成功的记录



四、实验总结

(一) 总结

通过"实验二、触发器的设计与测试", 对存储过程和触发器有了深入的认识, 掌握了存储过程及触发器的设计与测试, 为以后的实验打下基础

2018/09/16 19:42

实验人: 张启洋

学号: 1120161881