# 实验报告-计信2109-张嘉辉

## 一、实验目的

1、掌握用户自定义类型的使用

2、掌握变量的分类及其使用。

3、掌握各种流程控制语句的使用。

4、掌握游标的使用。

5、掌握系统存储过程及自定义存储过程的使用。

6、掌握系统函数及用户自定义函数的使用。

7、掌握触发器的创建和使用

## 二、实验平台

操作系统：windows11

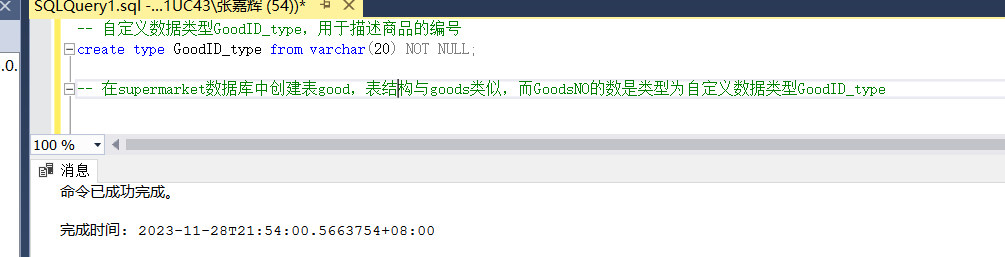
数据库：MySQL8

数据库管理平台：Navicat Premium 15

## 三、实验内容

**1、自定义数据类型GoodID\_type，用于描述商品的编号**

create type GoodID\_type from varchar(20) NOT NULL;

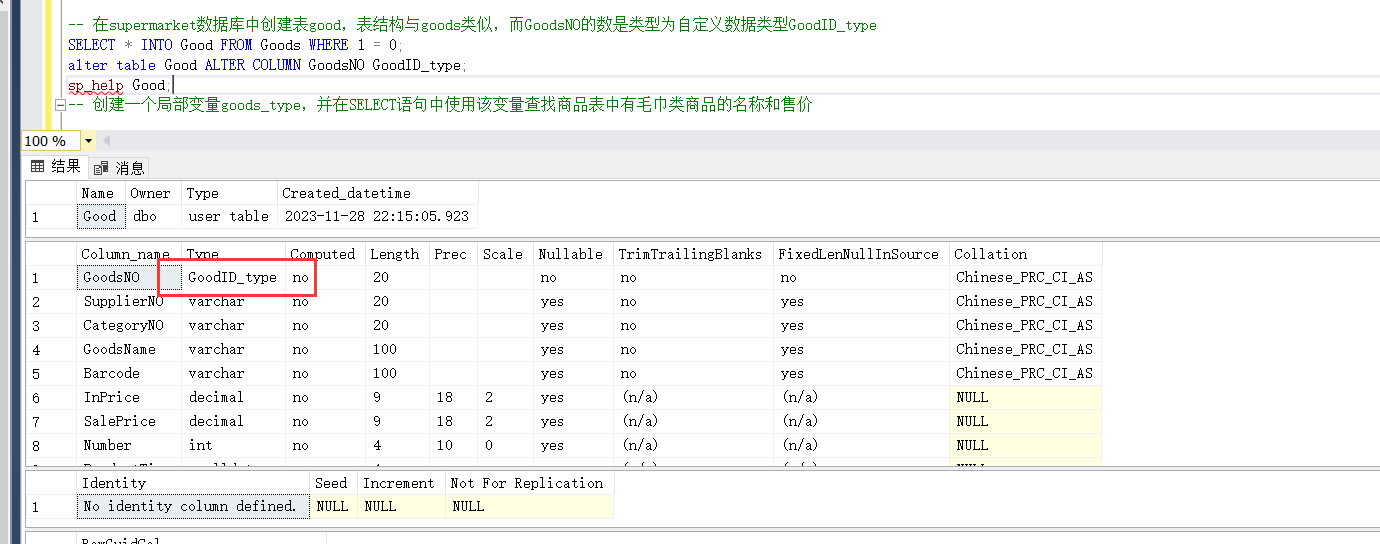


**2、在supermarket数据库中创建表good，表结构与goods类似，而GoodsNO的数是类型为自定义数据类型GoodID\_type**

SELECT \* INTO Good FROM Goods WHERE 1 = 0;

alter table Good ALTER COLUMN GoodsNO GoodID\_type;

sp\_help Good;



**3、创建一个局部变量goods\_type，并在SELECT语句中使用该变量查找商品表中有毛巾类商品的名称和售价**

DECLARE @goods\_type NVARCHAR(50);

SET @goods\_type = '毛巾';

select GoodsName, SalePrice from Goods where CategoryNo in (

select CategoryNO from Category where CategoryName = @goods\_type

)



**4、判断商品表goods是否存在商品类型为 “白酒” 的商品，如果存在则显示该类到有商品信息，否则显示无此类商品**

CREATE PROCEDURE CheckWhiteWineGoods

AS

BEGIN

DECLARE @Count int;

DECLARE @CategoryID varchar(20);

SELECT @CategoryID = CategoryNO

FROM Category

WHERE CategoryName = '白酒';

SELECT @Count = COUNT(\*)

FROM Goods

WHERE CategoryNO = @CategoryID;

IF @Count > 0

BEGIN

-- 存在白酒类商品

SELECT GoodsName, SalePrice

FROM Goods

WHERE CategoryNO = @CategoryID;

END

ELSE

BEGIN

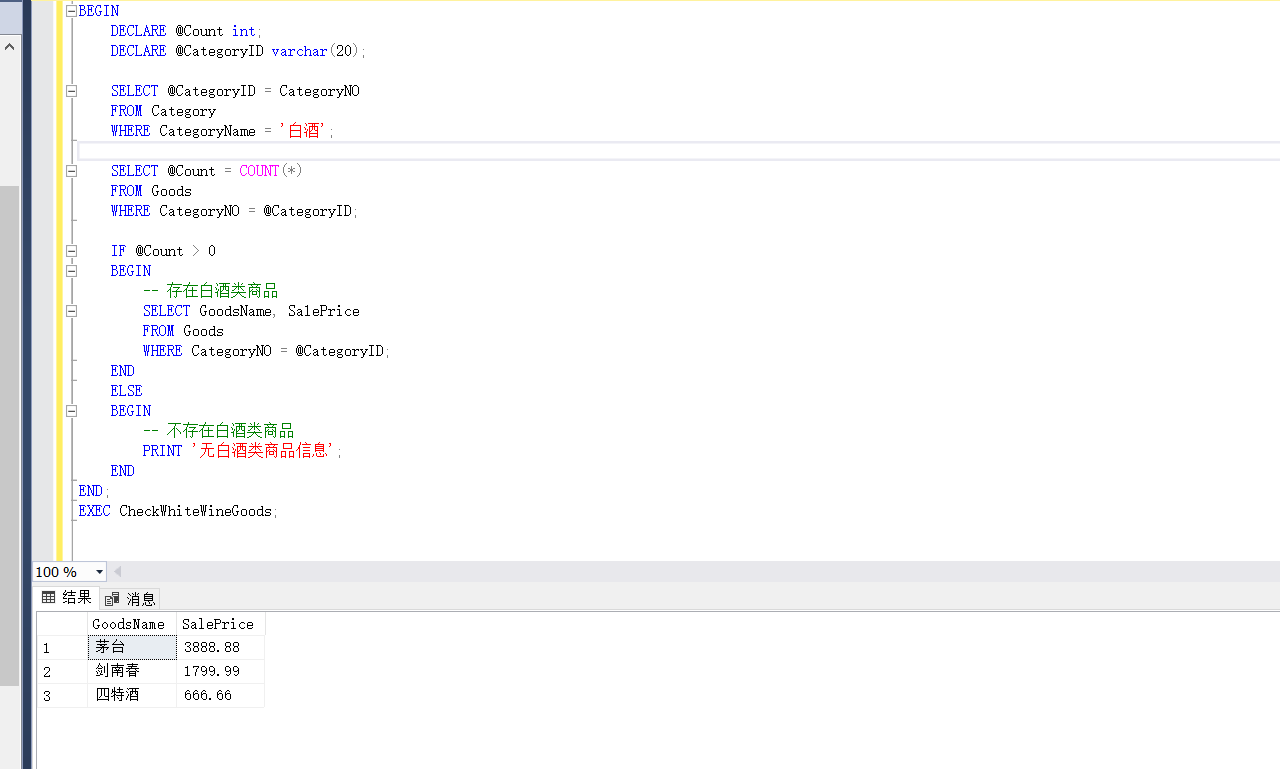
-- 不存在白酒类商品

PRINT '无白酒类商品信息';

END

END;

EXEC CheckWhiteWineGoods;



**5、如果商品表goods中存在商品数量小于10的情况，则将所有商品数量增加10，反复执行直到所有商品的数量都不小于10为止**

select \* from Goods

create procedure AddNumberGoods

as

begin

while exists (select 1 from Goods where Number < 10)

begin

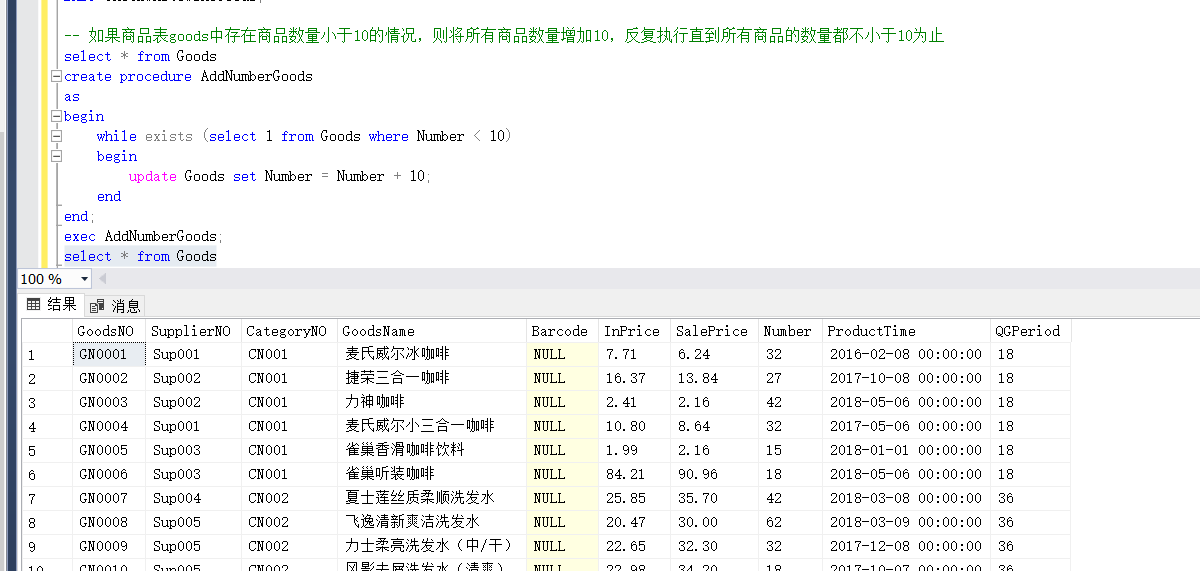
update Goods set Number = Number + 10;

end

end;

exec AddNumberGoods;

select \* from Goods



**6、声明一个游标，用于对 “饼干” 类商品的售价降价5%**

declare DiscountCursor cursor for

select GoodsNO, SalePrice from Goods where CategoryNO = (

select CategoryNO from Category where CategoryName = '饼干'

)

DECLARE @GoodsID varchar(20), @SalePrice DECIMAL(10, 2);

open DiscountCursor;

fetch next from DiscountCursor INTO @GoodsID, @SalePrice

while @@FETCH\_STATUS = 0

begin

UPDATE Goods

SET SalePrice = @SalePrice \* 0.95

WHERE GoodsNO = @GoodsID;

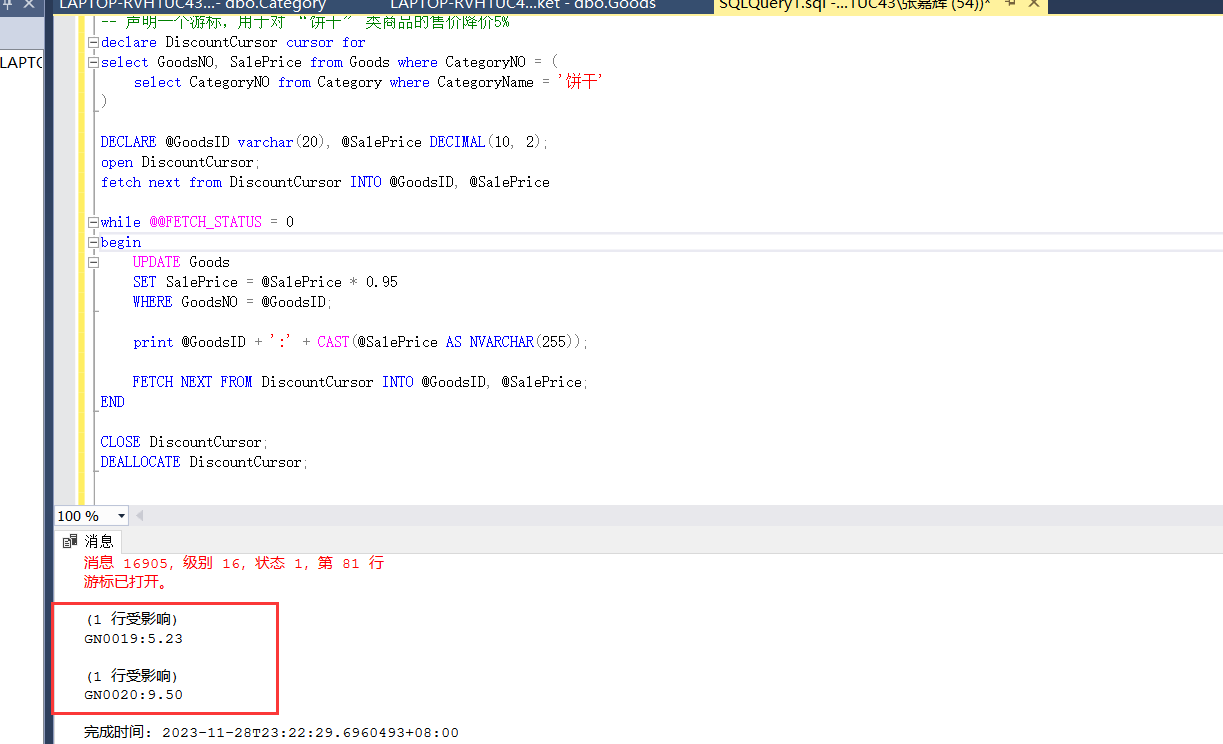
print @GoodsID + ':' + CAST(@SalePrice AS NVARCHAR(255));

FETCH NEXT FROM DiscountCursor INTO @GoodsID, @SalePrice;

END

CLOSE DiscountCursor;

DEALLOCATE DiscountCursor;



**7、创建一个有输入参数的存储过程，用于查询指定类别的所有商品信息。并执行该存储过程**

create procedure getGoodInfo

@name varchar(50)

as

begin

select \* from Goods where CategoryNO = (

select CategoryNO from Category where CategoryName = @name

);

end;

EXEC getGoodInfo @name = '饼干';



**8、创建一个有输入输出参数的存储过程，用于查询指定商品名的售价。并执行试存过程。**

create procedure getGoodSalePrice

@name varchar(50),

@sale DECIMAL(10, 2) output

as

begin

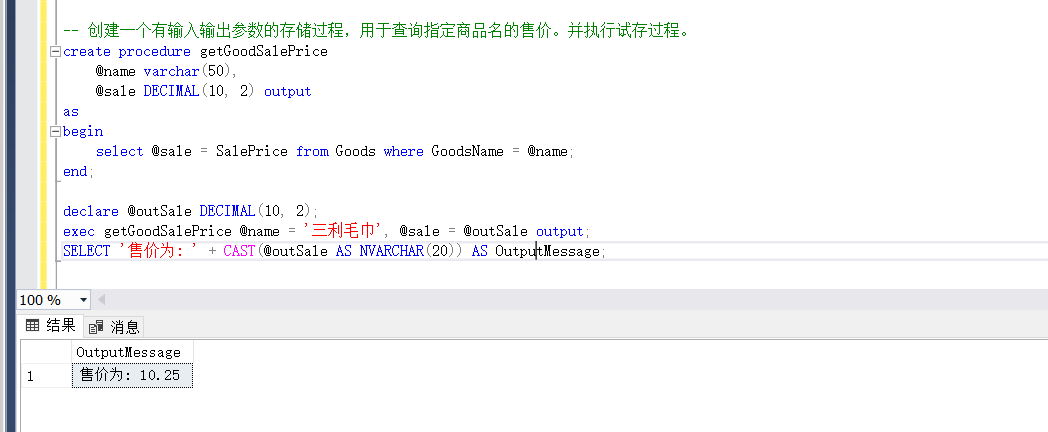
select @sale = SalePrice from Goods where GoodsName = @name;

end;

declare @outSale DECIMAL(10, 2);

exec getGoodSalePrice @name = '三利毛巾', @sale = @outSale output;

SELECT '售价为: ' + CAST(@outSale AS NVARCHAR(20)) AS OutputMessage;



**9、创建自定义函数，用于统计销售表salebill中某段时间内的销售情况。并调用该函数输出执行结果。**

create function SaleStatusByTimeFun

(

@start varchar(50),

@end varchar(50)

)

returns table

as

return

(

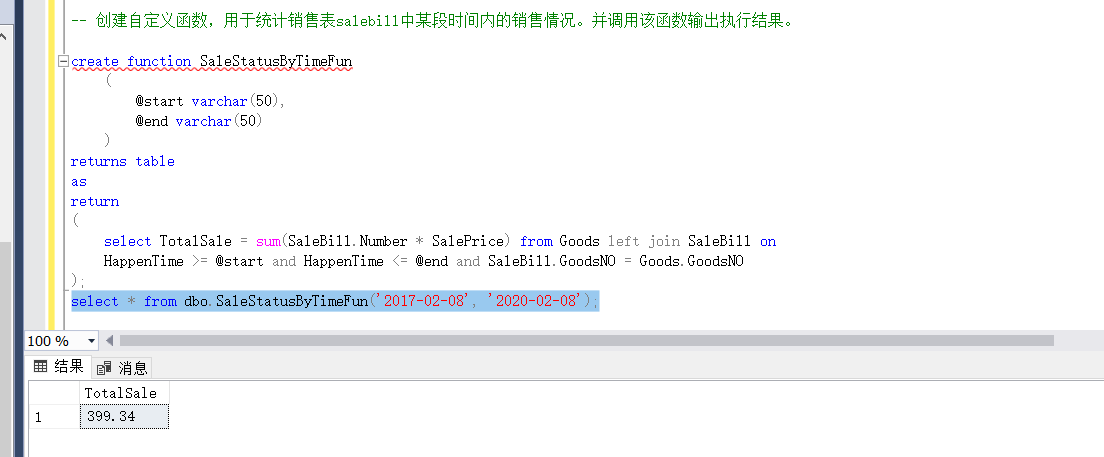
select TotalSale = sum(SaleBill.Number \* SalePrice) from Goods left join SaleBill on

HappenTime >= @start and HappenTime <= @end and SaleBill.GoodsNO = Goods.GoodsNO

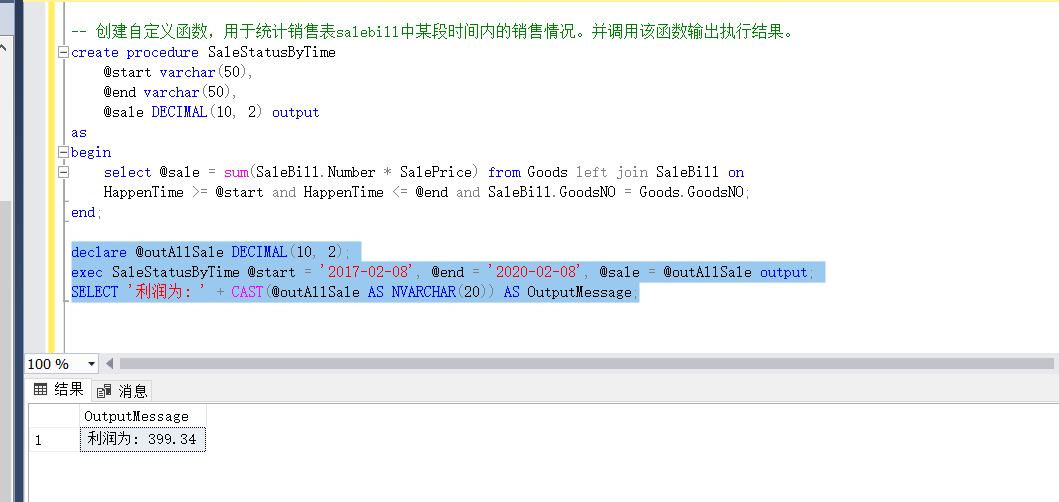
);

select \* from dbo.SaleStatusByTimeFun('2017-02-08', '2020-02-08');

**函数实现**



**存储过程实现**



**10、创建自定义函数，用于显示商品表goods中售价大于指定价格的商品信息，并调用该函数输出执行结果。**

CREATE FUNCTION GetGoodsByPrice

(@MinSalePrice DECIMAL(10, 2))

RETURNS TABLE

AS

RETURN

(

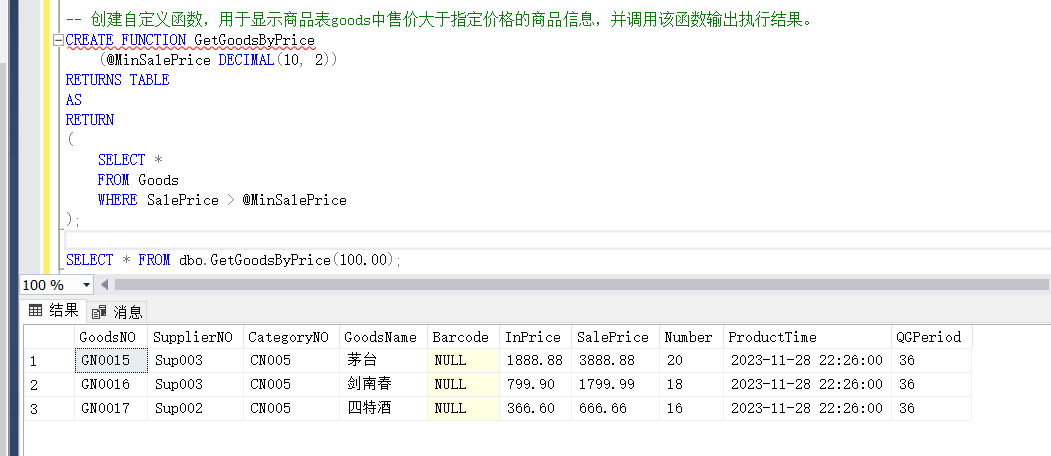
SELECT \*

FROM Goods

WHERE SalePrice > @MinSalePrice

);

SELECT \* FROM dbo.GetGoodsByPrice(100.00);



**11、创建一个触发器，向销售表salebill中插人一条记录时，这个触发器将更新商品表goods。goods表中数量为原有数量减去销售数量，如果库存数量小于10，则提示 “该商品数量小于10，低于安全库存量，请及时进货物” ；如果原有数量不足，则提示 “数量不足”**

CREATE TRIGGER UpdateGoodsOnSale

ON SaleBill

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

-- 更新商品表中的库存数量

UPDATE Goods

SET Number = Goods.Number - i.Number

FROM Goods

INNER JOIN inserted i ON Goods.GoodsNO = i.GoodsNO;

-- 检查库存数量是否低于10

IF EXISTS (SELECT 1 FROM Goods WHERE Number < 10)

BEGIN

PRINT '该商品数量小于10，低于安全库存量，请及时进货。';

END

-- 检查原有数量是否不足

IF EXISTS (SELECT 1 FROM Goods g INNER JOIN inserted i ON g.GoodsNO = i.GoodsNO WHERE g.Number < 0)

BEGIN

PRINT '数量不足。';

END

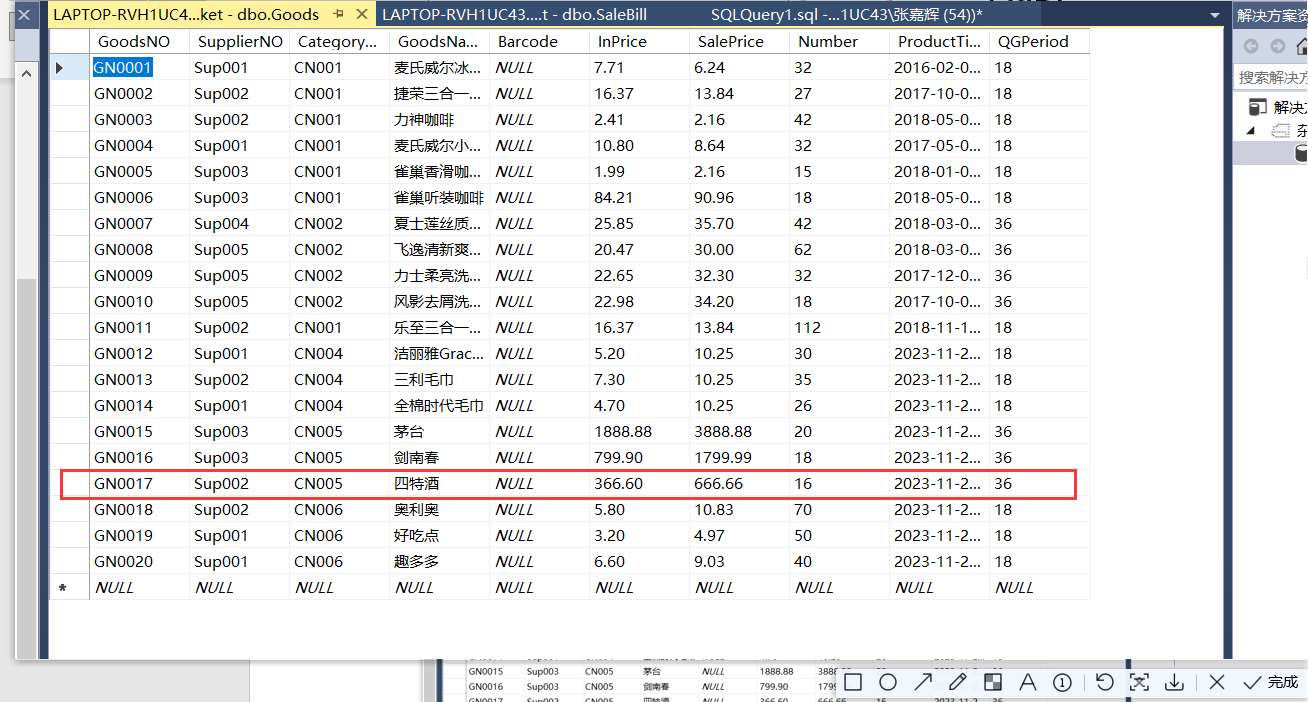
END;

insert into SaleBill values('GN0017', 'SO1', GETDATE(), 3);

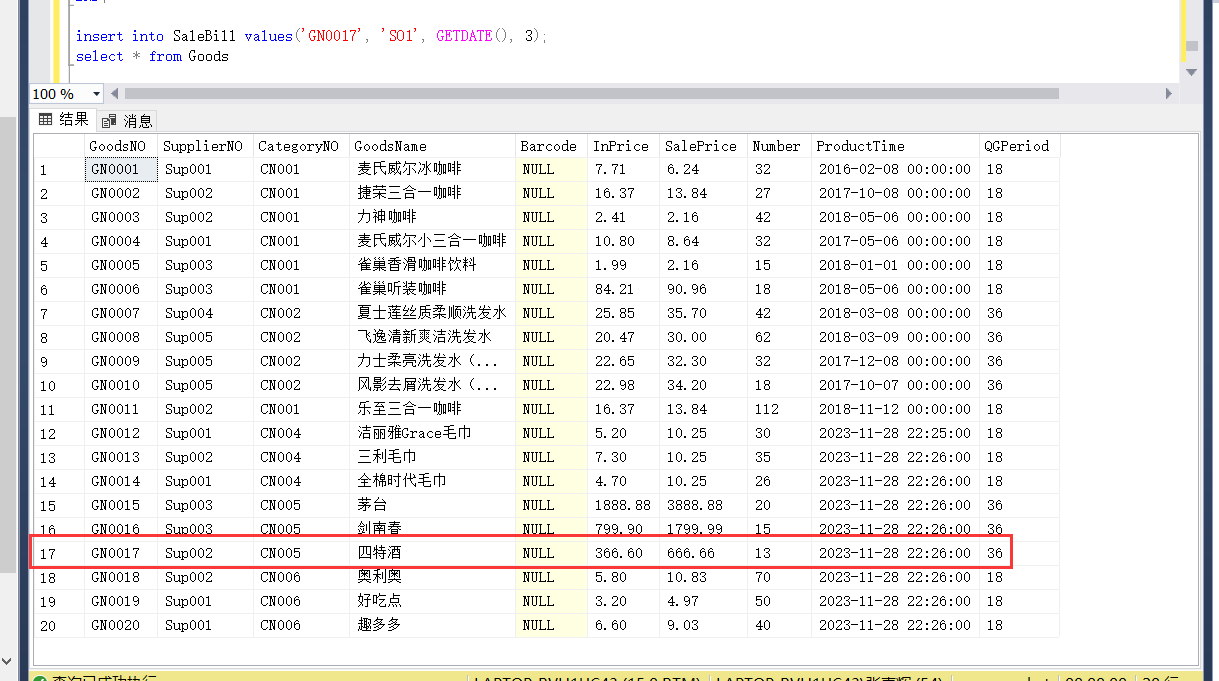
select \* from Goods



**我们先查看数据库中数据**



**在指定完插入之后，数据变化**



## 四、实验小结

经过这次实验，我了解了SQL Server的触发器，函数，游标，存储过程的处理方式