


学号：031602523 姓名：刘宏岩 学院：数计学院 专业：计算机类


《数据库应用实践》实验一：数据库管理系统及其应用开发环境的创建使用

- 实验目的：

了解数据库应用开发环境的建立与使用；掌握 SQL 语言的使用；通过实践理解关系数据模型的相关概念；掌握数据库应用开发环境的使用；掌握创建、删除数据库的方法；掌握创建基本表、查看表属性、修改属性的方法；掌握向表中添加、删除以及修改数据的方法；掌握查询分析器的使用方法；掌握 SELECT 语句在单表查询中的应用；掌握复杂查询、多表查询的方法；掌握视图的使用方法；巩固数据库的基础知识。


- 实验环境：


 操作系统：Windows 10

 数据库管理系统：SQL sever 2017


- 实验内容：


1. 应用背景及设计的数据库名：

 背景：网店数据库系统

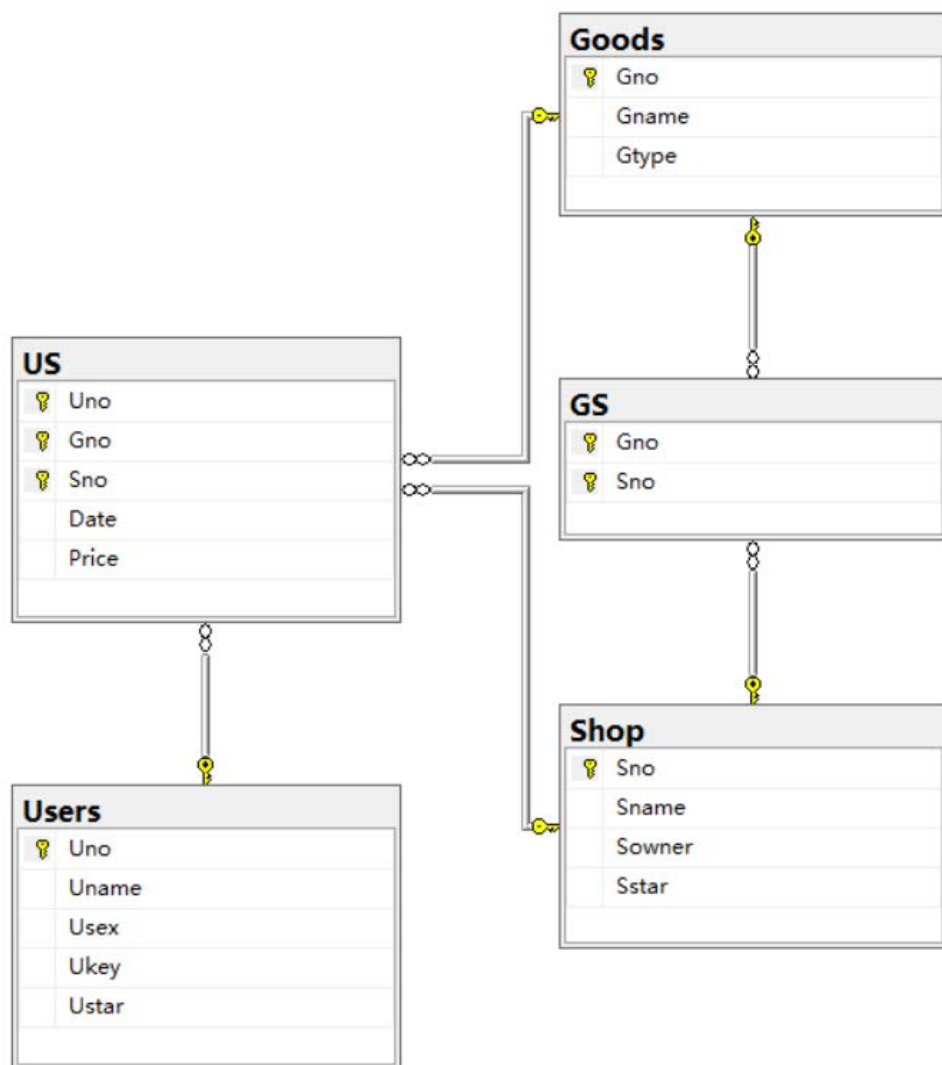
 数据库名：Ex1。

2. 所设计的各张表的说明（**红色**代表主码，**蓝色**代表外码）：

 店铺表 Shop (**Sno**,Sname,Sowner,Sstar): **店铺号**，店铺名，店铺主人，店铺星级。

 用户表 Users (**Uno**,Uname,Ukey,Usex,Ustar): **用户号**，用户名，密码，用户性别，用户信用星级。

- 商品表 Goods (Gno,Gname,Gtype): 商品号, 商品名, 商品类型。
- 商品店铺关系表 GS (Gno,Sno): 商品号, 店铺号。(备注: Gno, Sno 分别为外码。)
- 购买记录表 US (Uno,Sno,Gno,Date,Price): 用户账号, 店铺号, 商品号, 成交日期, 成交价。(备注: Uno, Gno, Sno 分别为外码。)
- UML 类图:



3. 创建一个应用数据库、创建并维护基本表的结构与数据。

1) 创建数据库。

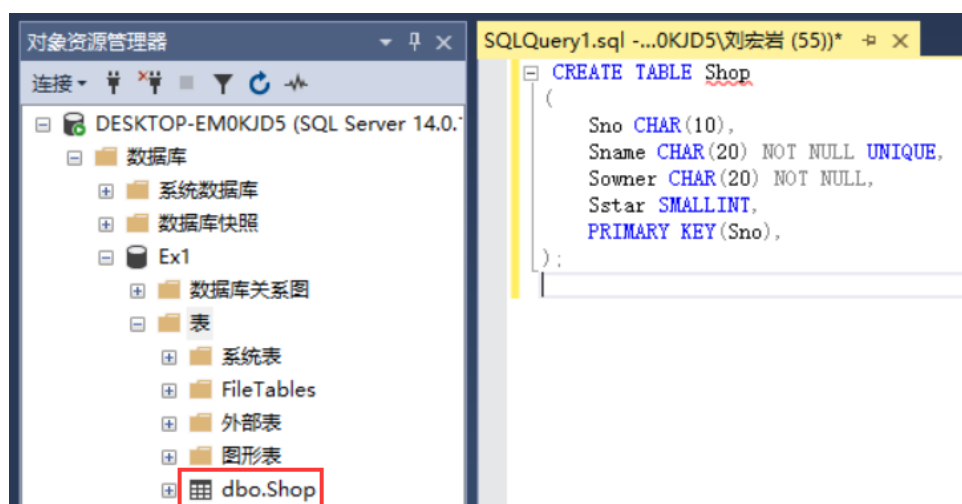
```
CREATE DATABASE Ex1
```



- 2) 在该数据库中创建至少 4 个相互关联的基本表，并设置主键、外键、自定义完整性约束（非空、唯一、默认值、check）。

创建店铺表

```
CREATE TABLE Shop
(
    Sno CHAR(10),
    Sname CHAR(20) NOT NULL UNIQUE,
    Sowner CHAR(20) NOT NULL,
    Sstar SMALLINT,
    PRIMARY KEY(Sno), #表级完整性约束约束条件, Sno 是主码。
);
```

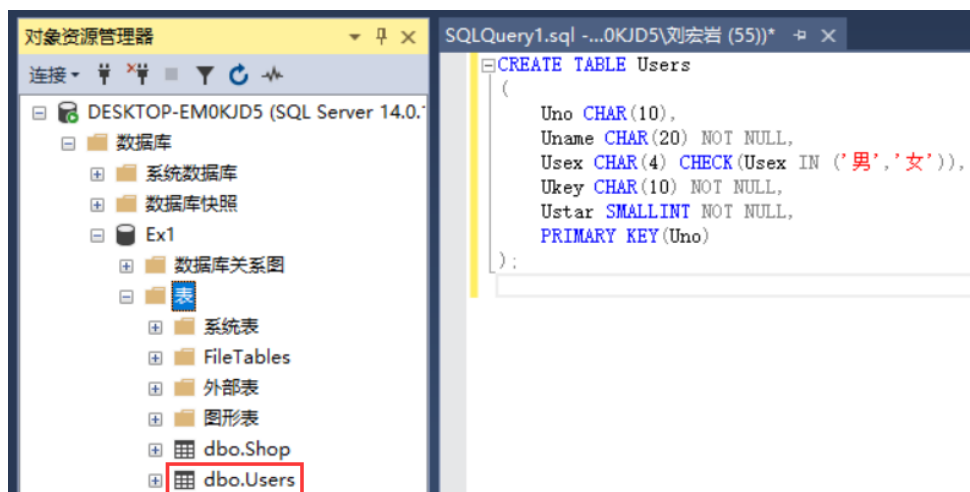


- 详细信息：

dbo.Shop	
列	
Sno (PK, char(10), not null)	
Sname (char(20), not null)	
Sowner (char(20), not null)	
Sstar (smallint, null)	

创建用户表

```
CREATE TABLE Users
(
    Uno CHAR(10),
    Uname CHAR(20) NOT NULL,
    Usex CHAR(4) CHECK(Usex IN ('男','女')), #check 约束
    Ukey CHAR(10) NOT NULL,
    Ustar SMALLINT NOT NULL,
    PRIMARY KEY(Uno) #表级完整性约束约束条件, Uno 是主码。
);
```



● 详细信息：

dbo.Users	
列	
Uno (PK, char(10), not null)	
Uname (char(20), not null)	
Usex (char(4), null)	
Ukey (char(10), not null)	
Ustar (smallint, not null)	

创建商品表

```
CREATE TABLE Goods
```

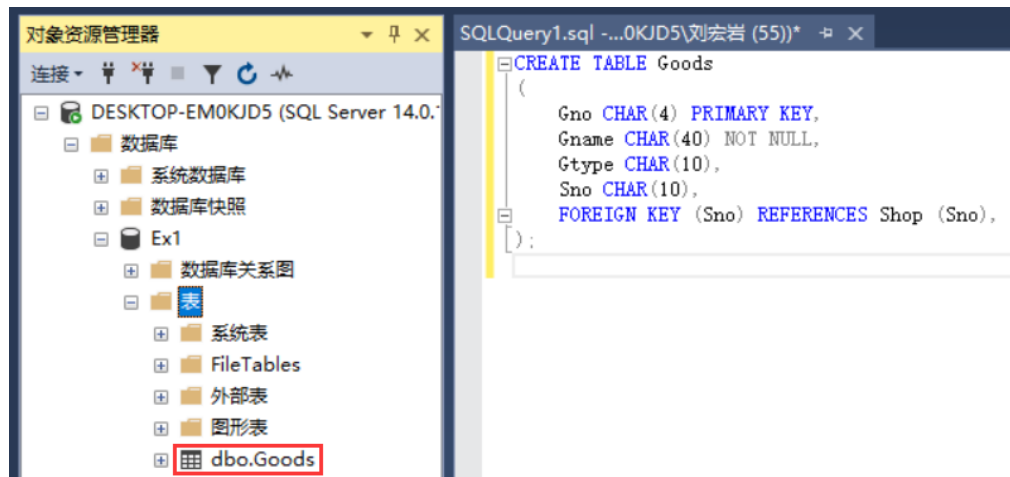
```
(
```

```
    Gno CHAR(4) PRIMARY KEY, #列级完整性约束条件, Gno是主码
```

```
    Gname CHAR(40) NOT NULL,
```

```
    Gtype CHAR(10),
```

```
);
```



● 详细信息：

dbo.Goods
列
Gno (PK, char(4), not null)
Gname (char(40), not null)
Gtype (char(10), null)

创建商品-店铺表

```
CREATE TABLE GS
```

```
(
```

```
    Gno CHAR(4),
```

```
    Sno CHAR(10),
```

```
    PRIMARY KEY(Gno,Sno), #组合主码
```

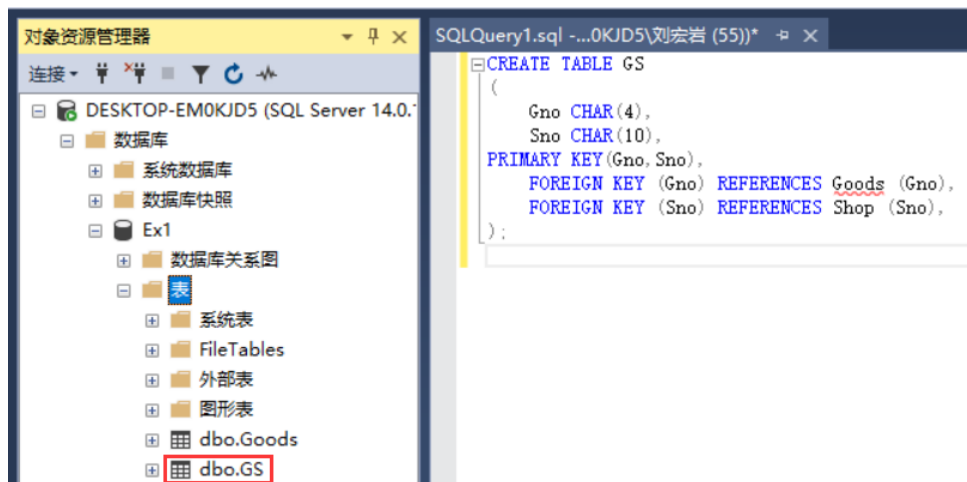
```
    FOREIGN KEY (Gno) REFERENCES Goods (Gno),
```

```
    #表级完整性约束条件, Gno 是外码, 被参照表是 Goods, 被参照属性是 Gno
```

```
    FOREIGN KEY (Sno) REFERENCES Shop (Sno),
```

```
    #表级完整性约束条件, Sno 是外码, 被参照表是 Shop, 被参照属性是 Sno
```

```
);
```



● 详细信息：

dbo.GS

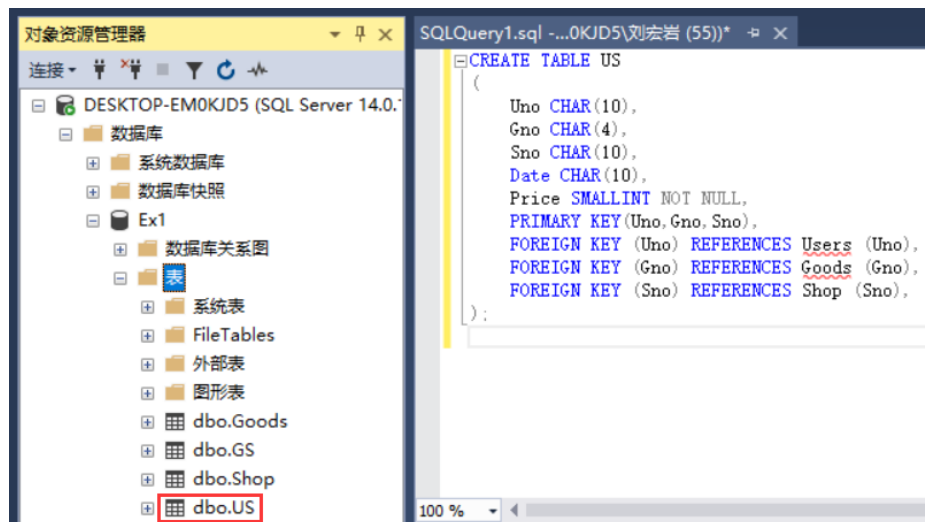
列

Gno (PK, FK, char(4), not null)

Sno (PK, FK, char(10), not null)

创建购买记录表：

```
CREATE TABLE US
(
    Uno CHAR(10),
    Gno CHAR(4),
    Sno CHAR(10),
    Date CHAR(10),
    Price SMALLINT NOT NULL,
    PRIMARY KEY(Uno,Gno,Sno), #组合主码
    FOREIGN KEY (Uno) REFERENCES Users (Uno),
    #表级完整性约束条件，Uno 是外码，被参照表是 Users, 被参照属性是 Uno
    FOREIGN KEY (Gno) REFERENCES Goods (Gno),
    #表级完整性约束条件，Gno 是外码，被参照表是 Goods, 被参照属性是 Gno
    FOREIGN KEY (Sno) REFERENCES Shop (Sno),
    #表级完整性约束条件，Sno 是外码，被参照表是 Shop, 被参照属性是 Sno
);
```



- 详细信息：

dbo.US
列
Uno (PK, FK, char(10), not null)
Gno (PK, FK, char(4), not null)
Sno (PK, FK, char(10), not null)
Date (char(10), null)
Price (smallint, not null)

3) 维护基本表的结构

🚦 用户、店铺星级在 1-5 之间，添加 CONSTRAINT 约束：

```
ALTER TABLE Shop
ADD CONSTRAINT C1 CHECK (Sstar>=1 AND Sstar<=5);
```

```
ALTER TABLE Users
ADD CONSTRAINT C2 CHECK (Ustar>=1 AND Ustar<=5);
```

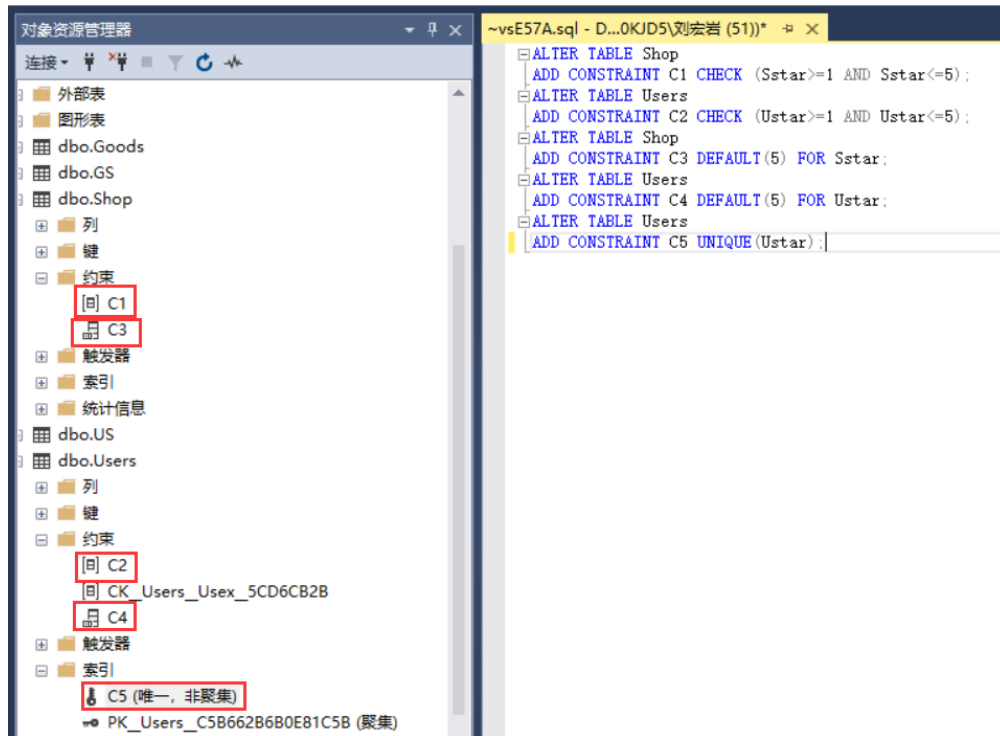
🚦 用户、店铺表中，设置默认星级为 5：

```
ALTER TABLE Shop
ADD CONSTRAINT C3 DEFAULT(5) FOR Sstar;
```

```
ALTER TABLE Users
ADD CONSTRAINT C4 DEFAULT(5) FOR Ustar;
```

🚦 用户表中，用户名唯一：

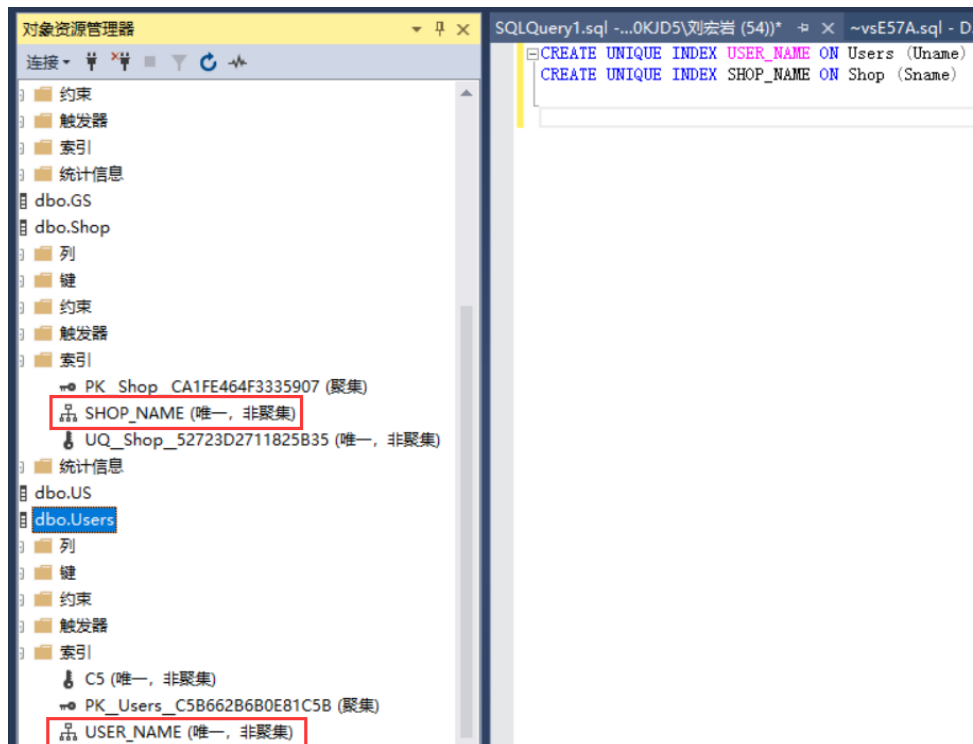
```
ALTER TABLE Users  
ADD CONSTRAINT C5 UNIQUE (Uname);
```




4) 创建并查看索引对象。

🚦 为店铺表、用户表创建索引：

```
CREATE UNIQUE INDEX USER_NAME ON Users (Uname);  
CREATE UNIQUE INDEX SHOP_NAME ON Shop (Sname);
```

5) 向表中录入若干数据，并维护表中数据。

 商品表插入数据：

```
INSERT
INTO Goods
VALUES('G001','数据库系统概论','书籍');
INSERT
INTO Goods
VALUES('G002','华为平板电脑','电子产品');
INSERT
INTO Goods
VALUES('G003','iPhoneX','电子产品');
INSERT
INTO Goods
VALUES('G004','NIKE 跑鞋','运动装备');
INSERT
INTO Goods
VALUES('G005','美的洗衣机','家用电器');
INSERT
INTO Goods
VALUES('G006','香奈儿洗面奶','护肤品');
INSERT
INTO Goods
VALUES('G007','YSL 口红','化妆品');
```

	Gno	Gname	Gtype
1	G001	数据库系统概论	书籍
2	G002	华为平板电脑	电子产品
3	G003	IPhoneX	电子产品
4	G004	NIKE跑鞋	运动装备
5	G005	美的洗衣机	日用电器
6	G006	香奈儿洗面奶	护肤品
7	G007	YSL口红	化妆品

🚦 用户表插入数据：

```

INSERT
INTO Users(Uno,Uname,Ukey,Usex,Ustar)
VALUES('U001','余锦昌','yjc1','男',1);
INSERT
INTO Users(Uno,Uname,Ukey,Usex,Ustar)
VALUES('U002','郑云涛','dcc1','男',2);
INSERT
INTO Users(Uno,Uname,Ukey,Usex,Ustar)
VALUES('U003','郑得华','zdh1','男',3);
INSERT
INTO Users(Uno,Uname,Ukey,Usex,Ustar)
VALUES('U004','许云涛','xyt1','女',4);
INSERT
INTO Users(Uno,Uname,Ukey,Usex)
VALUES('U005','房子涵','fzh1','男');
INSERT
INTO Users(Uno,Uname,Ukey,Usex)
VALUES('U006','陈星屹','cxy1','男');

```

	Uno	Uname	Usex	Ukey	Ustar
1	U001	余锦昌	男	yjc1	1
2	U002	郑云涛	男	dcc1	2
3	U003	郑得华	男	zdh1	3
4	U004	许云涛	女	xyt1	4
5	U005	房子涵	男	fzh1	5
6	U006	陈星屹	男	cxy1	5

🚦 店铺表插入数据：

```
INSERT
INTO Shop(Sno,Sname,Sowner,Sstar)
VALUES('S001','广群图书专卖','余广群',1);
INSERT
INTO Shop(Sno,Sname,Sowner,Sstar)
VALUES('S002','永安数码','郑永安',2);
INSERT
INTO Shop(Sno,Sname,Sowner,Sstar)
VALUES('S003','CC 运动','TCC',3);
INSERT
INTO Shop(Sno,Sname,Sowner)
VALUES('S004','咕咕鸡电器','咕咕鸡');
INSERT
INTO Shop(Sno,Sname,Sowner)
VALUES('S005','菊儿美妆','郑菊');
```

	Sno	Sname	Sowner	Sstar
1	S001	广群图书专卖	余广群	1
2	S002	永安数码	郑永安	2
3	S003	CC运动	TCC	3
4	S004	咕咕鸡电器	咕咕鸡	5
5	S005	菊儿美妆	郑菊	5

🚦 商品-店铺表插入数据：

```
INSERT
INTO GS(Gno,Sno)
VALUES('G001','S001');
INSERT
INTO GS(Gno,Sno)
VALUES('G002','S002');
INSERT
INTO GS(Gno,Sno)
VALUES('G003','S002');
INSERT
INTO GS(Gno,Sno)
VALUES('G004','S003');
INSERT
INTO GS(Gno,Sno)
VALUES('G005','S004');
INSERT
INTO GS(Gno,Sno)
VALUES('G006','S005');
INSERT
INTO GS(Gno,Sno)
VALUES('G007','S005');
```

	Gno	Sno
1	G001	S001
2	G002	S002
3	G003	S002
4	G004	S003
5	G005	S004
6	G006	S005
7	G007	S005

🚦 购买记录表插入数据:

```

INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U001','S001','G001','2018.5.1','42');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U001','S002','G002','2018.5.2','2000');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U002','S004','G005','2018.4.25','1500');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U003','S002','G003','2018.3.4','8000');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U004','S005','G006','2018.4.10','200');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U004','S005','G007','2018.4.10','300');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U005','S001','G001','2018.4.19','42');
INSERT
INTO US (Uno,Sno,Gno,Date,Price)
VALUES('U006','S002','G002','2018.4.18','2000');

```

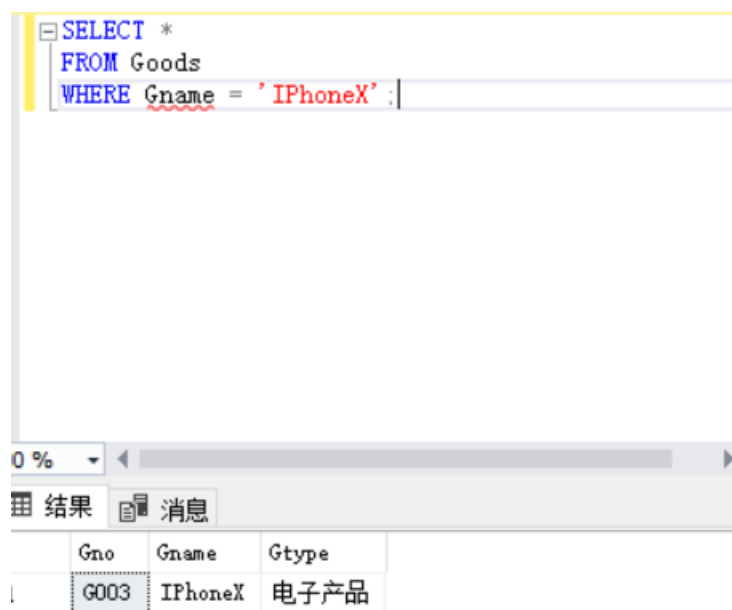
	Uno	Gno	Sno	Date	Price
1	U001	G001	S001	2018.5.1	42
2	U001	G002	S002	2018.5.2	2000
3	U002	G005	S004	2018.4.25	1500
4	U003	G003	S002	2018.3.4	8000
5	U004	G006	S005	2018.4.10	200
6	U004	G007	S005	2018.4.10	300
7	U005	G001	S001	2018.4.19	42
8	U006	G002	S002	2018.4.18	2000

4. 数据库查询，视图使用

1) 单表查询。

🚦 查询 “iPhoneX” 的商品信息：

```
SELECT *  
FROM Goods  
WHERE Gname = 'iPhoneX';
```



2) 多表连接查询并排序输出。

🚦 查询店铺号为 “S002” 的店铺所包含的商品名：

```
SELECT Goods.Gname  
FROM GS, Goods, Shop  
WHERE Shop.Sno = 'S002' AND Shop.Sno = GS.Sno AND Goods.Gno = GS.Gno;
```

<pre> SELECT Goods.Gname FROM GS, Goods, Shop WHERE Shop.Sno = 'S002' AND Shop.Sno = GS.Sno AND Goods.Gno = GS.Gno; </pre>							
100 %	<div> <div>结果</div> <div>消息</div> </div> <table> <tr> <th></th><th>Gname</th></tr> <tr> <td>1</td><td>华为平板电脑</td></tr> <tr> <td>2</td><td>iPhoneX</td></tr> </table>		Gname	1	华为平板电脑	2	iPhoneX
	Gname						
1	华为平板电脑						
2	iPhoneX						

3) 使用聚集函数的查询。

🚦 查询每个用户的消费总额：

```

SELECT Uname,SUM(Price)
FROM Users,US
WHERE Users.Uno = US.Uno;
GROUP BY Users.Uname;

```

SQLQuery1.sql - ...0KJD5\刘宏岩 (54))*

```
SELECT Uname, SUM(Price)
FROM Users, US
WHERE Users.Uno = US.Uno
GROUP BY Users.Uname;
```

100 %

结果

消息

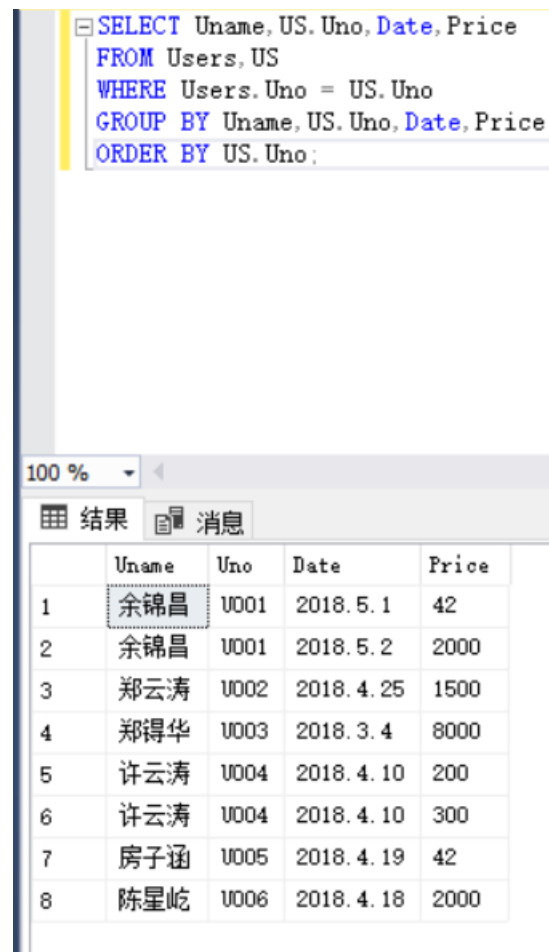
	Uname	(无列名)
1	陈星屹	2000
2	房子涵	42
3	许云涛	500
4	余锦昌	2042
5	郑得华	8000
6	郑云涛	1500

4) 分组查询。

按照用户名分组，查询每个用户的消费记录：

```
SELECT Uname,US.Uno,Date,Price
FROM Users,US
WHERE Users.Uno = US.Uno
GROUP BY Uname,US.Uno,Date,Price
ORDER BY US.Uno;
```

特别注意：Select 指定的每一列都应该出现在 Group By 子句中，除非对这一列使用了聚合函数。



The screenshot shows a SQL query editor with the following query:

```
SELECT Uname, US. Uno, Date, Price
FROM Users, US
WHERE Users. Uno = US. Uno
GROUP BY Uname, US. Uno, Date, Price
ORDER BY US. Uno;
```

Below the query editor, the results are displayed in a table with 8 rows and 5 columns: Uname, Uno, Date, and Price. The first row is highlighted with a dashed border.

	Uname	Uno	Date	Price
1	余锦昌	U001	2018. 5. 1	42
2	余锦昌	U001	2018. 5. 2	2000
3	郑云涛	U002	2018. 4. 25	1500
4	郑得华	U003	2018. 3. 4	8000
5	许云涛	U004	2018. 4. 10	200
6	许云涛	U004	2018. 4. 10	300
7	房子涵	U005	2018. 4. 19	42
8	陈星屹	U006	2018. 4. 18	2000

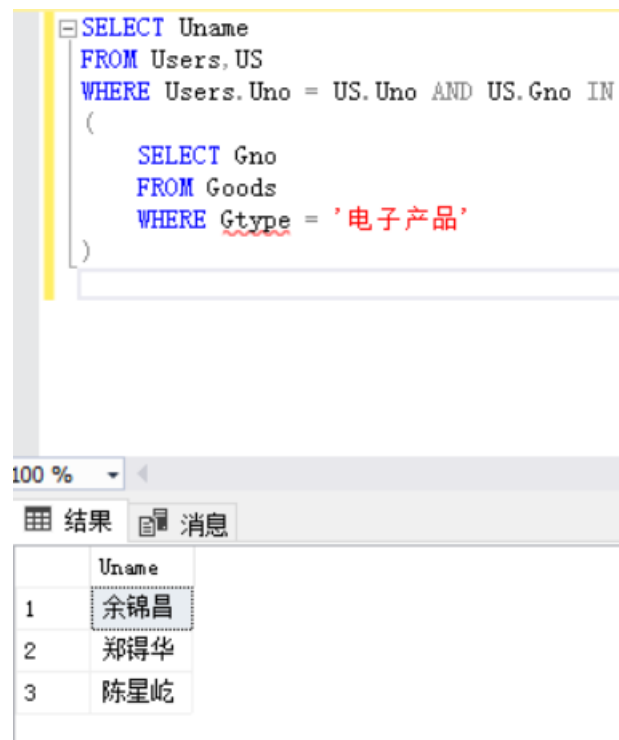
5) 嵌套查询。

查询所有购买过电子产品的人：

```

SELECT Uname
FROM Users,US
WHERE Users.Uno = US.Uno AND US.Gno IN
(
    SELECT Gno
    FROM Goods
    WHERE Gtype = '电子产品'
)
);

```



6) 创建并查询视图。

✚ 由于用户密码是敏感信息，当商家需要查用户信息和用户星级时，就必须要通过视图查看

✚ 创建用户信息视图：

```

CREATE VIEW USER_INFORMATION
AS
SELECT Uno,Uname,Usex,Ustar
FROM Users

```


	U _{no}	U _{name}	U _{sex}	U _{star}
1	U001	余锦昌	男	1
2	U002	郑云涛	男	2
3	U003	郑得华	男	3
4	U004	许云涛	女	4
5	U005	房子涵	男	5
6	U006	陈星屹	男	5

✚ 由于用户的购买记录会经常的被使用，所以建立视图可以减少连接时所耗费的资源。

✚ 建立消费记录视图：

```
CREATE VIEW RECORD
AS
SELECT Uname,Gname,Sname,Date,Price
FROM Users,US,Goods,Shop
WHERE Users.Uno = US.Uno AND US.Gno = Goods.Gno AND US.Sno = Shop.Sno
```

	U _{name}	G _{name}	S _{name}	Date	Price
1	余锦昌	数据库系统概论	广群图书专卖	2018.5.1	42
2	余锦昌	华为平板电脑	永安数码	2018.5.2	2000
3	郑云涛	美的洗衣机	咕咕鸡电器	2018.4.25	1500
4	郑得华	IPhoneX	永安数码	2018.3.4	8000
5	许云涛	香奈儿洗面奶	菊儿美妆	2018.4.10	200
6	许云涛	YSL口红	菊儿美妆	2018.4.10	300
7	房子涵	数据库系统概论	广群图书专卖	2018.4.19	42
8	陈星屹	华为平板电脑	永安数码	2018.4.18	2000

5. 实验心得体会

通过本次实验，把 SPOC 平台上学到的知识运用到实践中去，不单单是建立数据库库以及执行查询操作。还要考虑到实际情况等因素。我熟悉了 SQL sever 的使用规则，对理论知识有了更加深刻的记忆和理解。

本次实验让我认识到，学习知识不光要停留在理论，能做到理论与实际的结合才是真正的强者。我也要朝着这个方向努力。Learning by doing!

实践期间遇到的问题，通过查阅博客的方式解决。提高了自己独立解决问题的能力。

这份实验报告做了很久，到了最后这部分自己的体会会有很多。我个人很注重实践。不只是数据课程，其他课程的实践环节我都很重视。合格的实验报告，是对自己的一个交代，也是对老师辛苦工作的一个交代!