

数据库系统实验

实验一：数据库定义与操作语言

16 级计科教务 2 班

16337327

郑映雪

实验 1.1：数据库定义

实验题目

实验 1.1 数据库定义

实验目的

理解和掌握数据库 DDL 语言，能够熟练地使用 SQL DDL 语句创建、修改和删除数据库、模式和基本表。

实验重点和难点

实验重点：创建数据库、基本表

实验难点：创建基本表时，为不同的列选择合适的数据类型，正确创建表级和列级完整性约束，如列值是否允许为空、主码和外码等。注意：数据完整性约束可以在创建基本表时定义，也可以先创建表然后定义完整性约束。由于完整性约束的限制，被引用的表要先创建。

实验内容

定义数据库

建立 TPC-H 数据库模式，由零件表、供应商表、零件供应商联系表、顾客表、国家表、地区表、订单表和订单明细表组成。

它可以分为零件供应商和顾客订单两个子模式。

定义模式

在数据库中创建名为 Sales 的模式。

```
CREATE SCHEMA Sales;
```

定义基本表

定义地区表：

```
CREATE TABLE Region(  
    regionkey INTEGER PRIMARY KEY,  
    name CHAR(25),  
    comment VARCHAR(152),  
);
```

定义国家表：

```
CREATE TABLE Nation(  
    nationkey INTEGER PRIMARY KEY,  
    name CHAR(25),  
    regionkey INTEGER REFERENCES Region(regionkey),  
    comment VARCHAR(152),  
);
```

定义供应商基本表：

```
CREATE TABLE Supplier(  
    suppkey INTEGER PRIMARY KEY,  
    name CHAR(25),  
    address VARCHAR(40),  
    nationkey INTEGER REFERENCES Nation(nationkey),
```

```
    phone CHAR(15),  
    acctbal REAL,  
    comment VARCHAR(101)  
);
```

定义零件基本表:

```
CREATE TABLE Part(  
    partkey INTEGER PRIMARY KEY,  
    name VARCHAR(55),  
    mfgr CHAR(25),  
    brand CHAR(10),  
    type VARCHAR(25),  
    size INTEGER,  
    container CHAR(10),  
    retailprice REAL,  
    comment VARCHAR(23)  
);
```

定义零件供应联系表:

```
CREATE TABLE PartSupp(  
    partkey INTEGER REFERENCES Part(partkey),  
    supkey INTEGER REFERENCES Supplier(supkey),  
    availqty INTEGER,  
    supplycost REAL,  
    comment VARCHAR(199)  
    PRIMARY KEY(partkey,supkey)  
);
```

定义顾客表:

```
CREATE TABLE Customer(  
    custkey INTEGER PRIMARY KEY,  
    name VARCHAR(25),  
    address VARCHAR(40),  
    nationkey INTEGER REFERENCES Nation(nationkey),  
    phone CHAR(15),  
    acctbal REAL,  
    mktsegment CHAR(10),  
    comment VARCHAR(117)  
);
```

定义订单表:

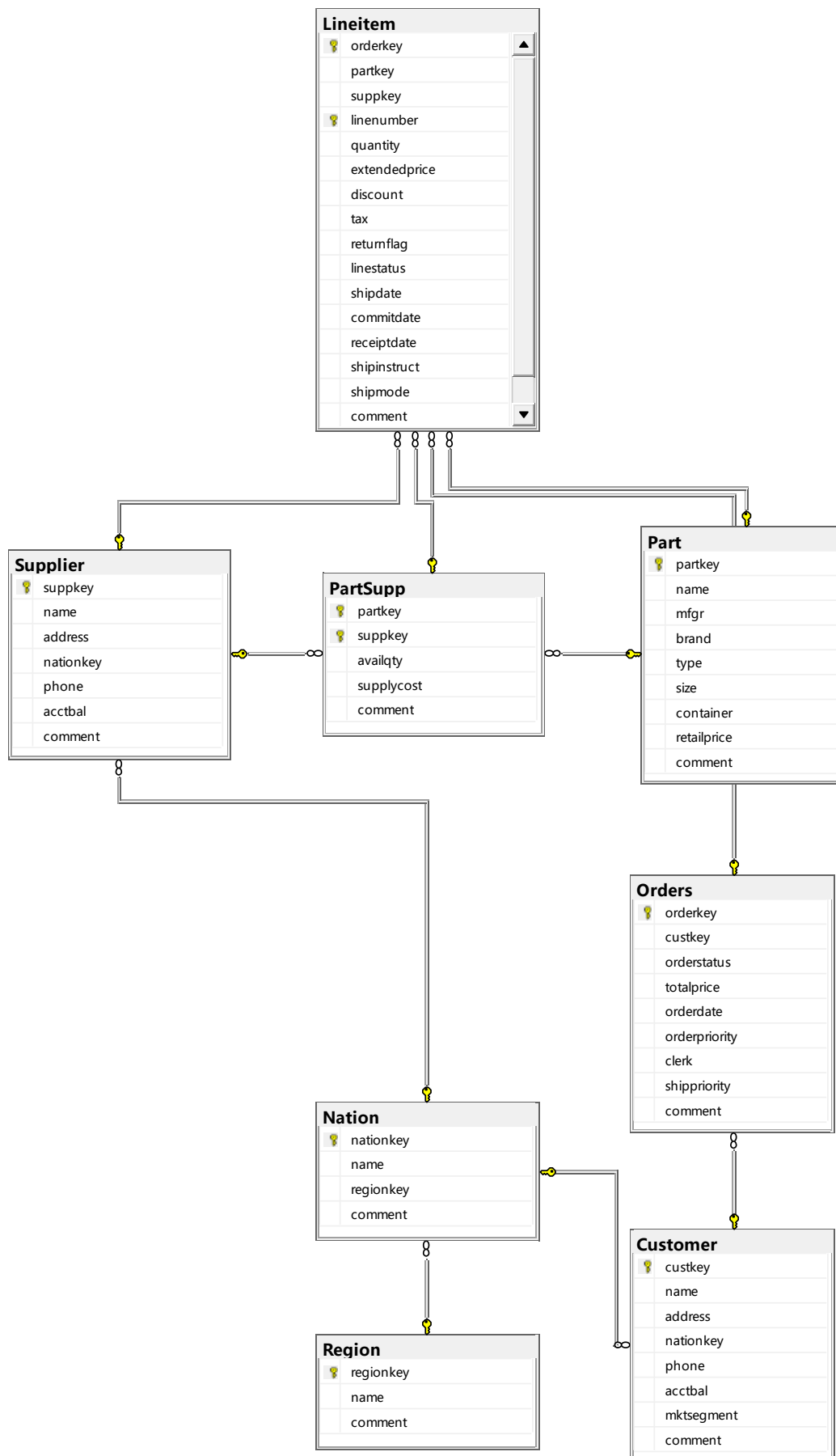
```
CREATE TABLE Orders(
    orderkey INTEGER PRIMARY KEY,
    custkey INTEGER REFERENCES Customer(custkey),
    orderstatus CHAR(1),
    totalprice REAL,
    orderdate DATE,
    orderpriority CHAR(15),
    clerk CHAR(15),
    shippriority INTEGER,
    comment VARCHAR(79)
);
```

定义订单明细表:

```
CREATE TABLE Lineitem(
    orderkey INTEGER REFERENCES Orders(orderkey),
    partkey INTEGER REFERENCES Part(partkey),
    supkey INTEGER REFERENCES Supplier(supkey),
    linenummer INTEGER,
    quantity REAL,
    extendedprice REAL,
    discount REAL,
    tax REAL,
    returnflag CHAR(1),
    linestatus CHAR(1),
    shipdate DATE,
    commitdate DATE,
    receiptdate DATE,
    shipinstruct CHAR(25),
    shipmode CHAR(10),
    comment VARCHAR(44),
    PRIMARY KEY(orderkey,linenummer),
    FOREIGN KEY(partkey,supkey) REFERENCES PartSupp(partkey,supkey)
);
```

实验结果

在 SQLServer 里运行定义的代码，并生成数据库关系图如下所示：



由上图知该关系表与书上关系表一致，关系创建成功。

实验总结

这次实验是数据库建立一些关系的初探。通过这次实验，我了解了关系的概念以及建立新表的方法。同时从官网上下载数据导入到空表中也花了我不少时间和精力去查资料 and 解决一些莫名其妙的问题（比如中文不能满足很多建表时的限制等），同时对 SQLServer 的使用更加熟练了。

实验 1.2：数据基本查询实验

实验题目

实验 1.2 数据基本查询实验

实验目的

掌握 SQL 程序设计的基本规范，熟练运用 SQL 语言实现数据基本查询，包括单表查询、分组统计查询和连接查询。

实验内容和要求

针对 TPC-H 数据库设计各种单表查询 SQL 语句、分组统计查询语句；设计单个表针对自身的连接查询，设计多个表的连接查询。理解和掌握 SQL 查询语句各个子句的特点和作用，按照 SQL 程序设计规范写出具体的 SQL 语句，并调试通过。

实验重点和难点

实验重点：分组统计查询、单表自身连接查询、多表连接查询。

实验难点：区分元组过滤条件和分组过滤条件；确定连接属性，正确设计连接条件。

实验操作及结果

导入数据

上一个实验只是创建了一些关系，它们只有属性并没有元组值，所以在进行后续的实验之前需要导入数据。一开始我没有在书上找到数据，以为要随机生成，后来在实验书的官方网站 <http://chinadb.ruc.edu.cn> 找到了已经生成好的数据。但是如何导入书上并没有教学，所以我搜索了一下批量导入 csv 文件的方法，语句如下：

```
BULK INSERT 关系名称
FROM '文件路径'
WITH(
FIELDTERMINATOR = ',',
ROWTERMINATOR = '\n'
);
```

但是导入数据后，发现书上很多变量的长度设定与官方本身给出的数据就不符，所以我修改了之前的变量限制长度。

最终导入的数据很成功，以其中一个关系 Lineitem 为例图示如下（其中 extendedprice 我按照书上的要求进行了批量的计算更新）：

	orderkey	partkey	suppkey	linenumber	quantity	extendedprice	discount	t
1	1	44930	18848	1	61	183	0	(
2	1	42522	13503	2	46	138	0	(
3	2	5001	15634	1	6	18	0	(
4	2	15802	29470	2	81	243	0	(
5	3	10672	26689	1	52	156	0	(
6	3	13973	4191	2	82	246	0	(
7	4	26000	1590	1	67	201	0	(
8	4	36680	24313	2	34	102	0	(

单表查询（实现投影操作）

```
SELECT name,address,phone
FROM Supplier;
```

查询结果的部分内容图示如下：

SQLQuery1.sql ...UCS8JK7\映雪 (52))*			
SELECT name, address, phone FROM Supplier			
00 %			
结果 消息			
	name	address	phone
7503	深圳市享迪贸易有限公司	广东 深圳市 龙华镇、玉翠新村338号	13631382882
7504	明懿达电子有限公司	广东 深圳市 龙华镇宝山工业区	13632560960
7505	深圳市三秒通通信有限公司	广东 深圳市 龙华镇布龙公路鹤山工业区58号	13632729857
7506	双联塑胶机械(深圳)有限公司	广东 深圳市 龙华镇城市明珠花园	13662279160
7507	深圳市派诺杰实业有限公司...	广东 深圳市 龙华镇赤岭路53号B2楼(龙华...	13682640468
7508	深圳市派诺杰实业有限公司...	广东 深圳市 龙华镇赤岭头新村53号B栋2楼	13686415602
7509	深圳市宝安区龙华宏翔机械...	广东 深圳市 龙华镇赤岭头新一村16巷	13689559313
7510	深圳市凯凯模具机械厂	广东 深圳市 龙华镇大浪村八区30栋	13691695678
7511	深圳市邦凯电子有限公司市场部	广东 深圳市 龙华镇大浪村华荣路鸿富工业区	13692248256
7512	成都威士达粉末冶金有限公司	广东 深圳市 龙华镇大浪村黄麻埔工业区	13713547027
7513	深圳市强生电机有限公司	广东 深圳市 龙华镇大浪村玉壶工业区	13713720890
7514	深圳华康五金铁线制品厂	广东 深圳市 龙华镇大浪工业区 hk-sale@...	13713834135

单表查询（实现选择操作）

这里遇到 SQLServer 不支持 CURRENT_DATE 语句的问题，于是我把书上的语句修改成同样支持的 getdate()函数，又因为官方数据的年代比较久远，所以找不到最近一周的数据，我又修改成一周以外的。

```
SELECT *
```

```
FROM Orders
WHERE (getdate()-orderdate)>7 AND totalprice>1000;
```

部分结果如下：

结果		消息							
	orderkey	custkey	ordorstatus	totalprice	orderdate	orderpriority	cler	shippriority	commer
1	1	320728	NULL	335028.8	2014-05-14	NULL	NULL	NULL	NULL
2	2	518445	NULL	3.564664E+07	2014-09-19	NULL	NULL	NULL	NULL
3	3	277668	NULL	2.435218E+07	2014-02-21	NULL	NULL	NULL	NULL
4	4	112812	NULL	594650	2013-12-12	NULL	NULL	NULL	NULL
5	6	459256	NULL	235164.4	2014-04-20	NULL	NULL	NULL	NULL
6	9	407013	NULL	1495096	2014-05-18	NULL	NULL	NULL	NULL
7	10	527764	NULL	1.664025E+07	2014-01-22	NULL	NULL	NULL	NULL

查询已成功执行

DESKTOP-UICS8IK7\SOLEXPRESS

DESKTOP-UICS8IK7\映雪 (52)

不带分组过滤条件的分组统计查询

```
SELECT C.custkey,SUM(O.totalprice)
FROM Customer C,Orders O
WHERE C.custkey = O.custkey
GROUP BY C.custkey
```

结果如下所示。其中（无列名）即为 totalprice 的求和。

结果		消息
	custkey	(无列名)
1	8	472870.71875
2	125	140079104
3	139	624746.3125
4	152	0
5	172	1058524.25
6	190	0
7	205	0
8	215	0
9	317	0
10	340	5774512
11	351	0
12	423	3360427.5

带分组过滤条件的分组统计查询

查找订单平均金额超过 1000 元的顾客编号及其姓名。

```
SELECT C.custkey,MAX(C.name)
FROM Customer C,Orders O
WHERE C.custkey=O.custkey
GROUP BY C.custkey
```

```
HAVING AVG(O.totalprice)>1000;
```

部分结果如下:

	custkey	(无列名)
13	805	安大业
14	876	安国强
15	907	安洪溪
16	970	安金军
17	988	安娟
18	1007	安澜
19	1095	安期姓
20	1184	安素琴
21	1205	安天桂
22	1208	安田成美
23	1242	安文纲
24	1334	安延

单表自身连接查询

查询与“金仓集团”在同一个国家的供应商编号、名称和地址信息。

```
SELECT F.supkey,F.name,F.address  
FROM Supplier F,Supplier S  
WHERE F.nationkey=S.nationkey AND S.name='金仓集团';
```

然而结果是这样的:

	supkey	name	address

是一个空集，于是我在 Supplier 表里想找到是否金仓集团:

```
SELECT F.supkey,F.name,F.address  
FROM Supplier F  
WHERE F.name='金仓集团';
```

	supkey	name	address

依然没有，说明这个数据集里根本就没有 name 为金仓集团的供应商，查询结果为空集是必然的。

两表连接查询（普通连接）

查询供应价格大于零售价格的零件名、制造商名、零售价格和供应价格。

```
SELECT P.name,P.mfgr,P.retailprice,PS.supplycost
FROM Part P,PartSupp PS
WHERE P.retailprice>PS.supplycost;
```

查询部分结果如下，由于是笛卡尔积，查询时间很长：

	name	mfgr	retailprice	supplycost
1	软盘	江苏省铜山县阀门厂	1.7	0
2	软盘	江苏省铜山县阀门厂	1.7	0
3	软盘	江苏省铜山县阀门厂	1.7	0.721747
4	软盘	江苏省铜山县阀门厂	1.7	1.068301
5	软盘	江苏省铜山县阀门厂	1.7	1.414856
6	软盘	苏州阀门厂	1.7	0
7	软盘	苏州阀门厂	1.7	0
8	软盘	苏州阀门厂	1.7	0.721747
9	软盘	苏州阀门厂	1.7	1.068301
10	软盘	苏州阀门厂	1.7	1.414856
11	软盘	江苏省无锡市阀门厂	1.8	0
12	软盘	江苏省无锡市阀门厂	1.8	0

两表连接查询（自然连接）

查询供应价格大于零售价格的零件名、制造商名、零售价格和供应价格。

```
SELECT P.name,P.mfgr,P.retailprice,PS.supplycost
FROM Part P,PartSupp PS
WHERE P.partkey=PS.partkey
AND P.retailprice>PS.supplycost;
```

结果如下：

100 %				
结果 消息				
	name	mfg	retailprice	supplycost
1	竹炭空气清新篮	郑州市荥阳通用阀门厂	3	0
2	竹炭净化包	江西赣州阀门厂	3.5	0
3	佳能墨盒	佳能	100	27.14209
4	惠普墨盒	惠普	120277	5431.521
5	惠普墨盒	惠普	188	114.6433
6	佳能墨盒	佳能	550	76.96249
7	亚麻竹炭汽车座垫	江西赣州阀门厂	825	480.6001
8	吉祥如意	湖北省英山阀门厂	896	170.1073
9	羊绒汽车座垫	河北省隆尧县明珠阀门有限公司	1206	335.5703

三表连接查询

查询顾客“苏举库”订购的订单编号、总价及其订购的零件编号、数量和明细表。

```
SELECT O.orderkey,O.totalprice,L.partkey,L.quantity,L.extendedprice
FROM Customer C,Orders O,Lineitem L
WHERE C.custkey=O.custkey AND O.orderkey=L.orderkey AND C.name='苏举库';
```

查询结果为空集。

结果 消息					
	orderkey	totalprice	partkey	quantity	extendedprice

修改名字为陈容：

结果 消息					
	orderkey	totalprice	partkey	quantity	extendedprice
1	15213	4696320	27673	90	2493000
2	15213	4696320	4229	4	456000

实验总结

本次实验主要是数据库的查询操作。在经历了复杂的数据导入之后，终于可以进行正常的查询了。通过这次实验，我了解到很多书上和 SQLServer 中不同的地方，比如 CURRENT_DATE 在 SQLServer 中就是非法的，我查阅资料后发现 getdate 函数在

SQL Server 中可以获取当前时间，就进行了替换。另外对于不同的查询方式和组织方式，查询的效率不一样。比如在笛卡尔积里查找就卡到爆表，时间很久，所以要尽量使用效率较高的查询语句进行实验。

实验 1.3 数据高级查询实验

实验题目

实验 1.3 数据高级查询实验

实验目的

掌握 SQL 嵌套查询和集合查询等各种高级查询的设计方法等。

实验内容和要求

针对 TPC-H 数据库，正确分析用户查询要求，设计各种嵌套查询和集合查询。

实验重点和难点

实验重点：嵌套查询。

实验难点：相关子查询、多层 EXIST 嵌套查询。

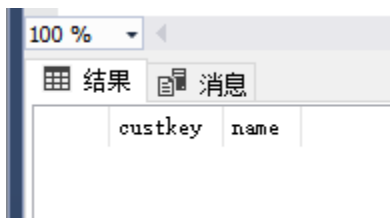
实验操作及结果

IN 嵌套查询

查询订购了“海大”制造的“船舶模拟驾驶舱”的顾客。

```
SELECT custkey,name
FROM Customer
WHERE custkey IN(
    SELECT O.custkey
    FROM Orders O,Lineitem L,PartSupp PS,Part P
    WHERE O.orderkey = L.orderkey AND
        L.partkey=PS.partkey AND
        L.supkey=PS.supkey AND
        P.partkey=PS.partkey AND
        P.mfgr='海大' AND P.name='船舶模拟驾驶舱'
);
```

结果如下：是空集。可见没有符合要求的元组。



The screenshot shows a database query result window. At the top, there is a zoom level of 100%. Below that, there are two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, and it displays a table with two columns: 'custkey' and 'name'. The table is empty, indicating that the query returned no results.

```
/*比较区别：*/
SELECT custkey,name
FROM Customer
WHERE custkey IN(
    SELECT O.custkey
    FROM Orders O,Lineitem L,Part P
    WHERE O.orderkey=L.orderkey AND
        L.partkey=P.partkey AND
        P.mfgr='海大' AND P.name='船舶模拟驾驶舱'
);
```

两个查询在一起比较，实际上没有区别。实际上是 orders natural join Lineitem natural join part，本质上与上一条查找查找出来的结果是一样的。

单层 EXISTS 嵌套查询

查询没有购买过“海大”制造的“船舶模拟驾驶舱”的顾客：

```
SELECT custkey,name
FROM Customer C
WHERE NOT EXISTS(
    SELECT O.custkey
    FROM Orders O,Lineitem L,Part P,PartSupp PS
    WHERE O.orderkey=C.custkey AND
        L.orderkey=O.orderkey AND
        L.partkey=PS.partkey AND
        L.supkey=PS.supkey AND
        PS.partkey=P.partkey AND
        P.mfgr='海大' AND P.name='船舶模拟驾驶舱'
);
```

结果自然是全部的顾客，因为上一条中查询结果为空集：



	custkey	name
	1	阿宝宝
:	2	阿爆
:	3	阿倍仲麻吕
:	4	阿比
:	5	阿碧
:	6	阿扁
:	7	阿炳
:	8	阿波罗
:	9	阿伯拉默夫

正在执行查询...

双层 EXISTS 嵌套查询

查询至少购买过顾客“张三”购买过的全部零件的顾客姓名。

```
SELECT CA.name
FROM Customer CA
WHERE NOT EXISTS
(
    SELECT *
    FROM Customer CB,Orders OB,Lineitem LB
    WHERE CB.custkey=OB.custkey AND
        OB.orderkey=LB.orderkey AND
        CB.name='张三' AND
```



```

NOT EXISTS(
  SELECT *
  FROM Orders OC,Lineitem LC
  WHERE CA.custkey = OC.custkey AND
  OC.orderkey=LC.orderkey AND
  LB.supkey=LC.supkey AND
  LB.partkey=LC.partkey
)
);

```

结果如下所示:

	name
7	阿炳
8	阿波罗
9	阿伯拉默夫
10	阿博德
11	阿卜杜尔
12	阿卜杜
13	阿不都
14	阿不都
15	阿不都...

FROM 子句中的嵌套查询

查询订单平均金额超过 1 万元的顾客中的中国籍顾客信息。

```

SELECT C.*
FROM Customer C,(
  SELECT custkey
  FROM orders
  GROUP BY custkey
  HAVING AVG(totalprice)>10000) B,Nation N
WHERE C.custkey=B.custkey AND
C.nationkey=N.nationkey AND N.name='中国';

```

结果如下所示:

	custkey	name	address	nationkey	phone	acctbal	mktsegment	comment
1	13991	朝花社	NULL	40	NULL	NULL	NULL	NULL
2	25579	陈佑天	NULL	40	NULL	NULL	NULL	NULL
3	28811	程九林	NULL	40	NULL	NULL	NULL	NULL
4	34976	单莉莉	NULL	40	NULL	NULL	NULL	NULL
5	56843	傅立田	NULL	40	NULL	NULL	NULL	NULL
6	60516	高焕领	NULL	40	NULL	NULL	NULL	NULL

集合查询交、并、差

查询顾客“张三”和“李四”都订购过的零件信息。

```
(SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey=O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='张三')
INTERSECT
SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey = O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='李四';
```

```
SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey=O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='张三'
UNION
SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey = O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='李四';
```

```
SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey=O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='张三'
EXCEPT
SELECT P.*
FROM Customer C,Orders O,Lineitem L,PartSupp PS,Part P
WHERE C.custkey = O.custkey AND O.orderkey = L.orderkey AND
      L.supkey=PS.supkey AND L.partkey=PS.partkey AND
      PS.partkey = P.partkey AND C.name='李四';
```

我查看了一下顾客表，发现实验书的张三和李四都没有买过东西。所以结果都是空集，如下所示：

partkey	name	mfg	brand	type	size	container	retailprice	comment
---------	------	-----	-------	------	------	-----------	-------------	---------

实验总结

本次实验进行的是数据的高级查询，包括很重要的嵌套查询。但是 SQLServer 并不支持 natural join 操作，这给自然连接带来了一定的麻烦……同时我发现书上的很多表达在 SQLServer 里都需要修改，这是由于不同的 DBMS 导致不同语句的兼容性不同。

同时，我也明白了很多情况下一种查询有很多种不同的表达，但是不同的表达对执行查询的时间又较大的影响。我们应该具体问题具体分析，选择最合适的表达进行查询。要善于利用 SQL 本身语法里的很多便利之处，尽量简化查询从而达到最高的效率。

实验 1.4 数据更新实验

实验题目

实验 1.4 数据更新实验

实验目的

熟悉数据库的数据更新操作，能够使用 SQL 语句对数据库进行数据的插入、修改、删除操作。

实验内容和要求

针对 TPC-H 数据库设计单元组插入、批量数据插入、修改数据和删除数据等 SQL 语句。理解和掌握 INSERT、UPDATE 和 DELETE 语法结构的各个组成成分，结合嵌套 SQL 子查询，分别设计几种不同形式的插入、修改和删除数据的语句，并调试成功。

实验重点和难点

实验重点：插入、修改和删除数据的 SQL。

实验难点：与嵌套 SQL 子查询相结合的插入、修改和删除数据的 SQL 语句；利用一个表的数据来插入、修改和删除另外一个表的数据。

实验操作和结果

INSERT 基本语句（插入全部列的数据）

插入一条顾客记录，要求每列都给一个合理的值。

```
INSERT INTO Customer
VALUES(30000000,'张三','北京市',40,'010-51001199',0.00,'Northeast','VIP
Customer');
```

这里有一个修改，因为 custkey 为主键，所以不存在插入重复的 custkey 的操作，所以把 30 修改为 30000000。插入后我进行了搜索，结果如下：

结果		消息						
	custkey	name	address	nationkey	phone	acctbal	mktsegment	comment
1	30000000	张三	北京市	40	010-51001199	0	Northeast	VIP Customer

INSERT 基本语句（插入部分列的数据）

插入一条订单记录，给出必要的几个字段值。

```
INSERT INTO Lineitem(orderkey,Linenummer,partkey,suppkey,quantity,shipdate)
VALUES (4999,ROUND(RAND()*100,0),7531,10911,10,'2012-3-6');
```

这里实验书上的本来的数据又是错误的，它违反了外键的约束，于是我又打开了数据集找了一行元组进行修改……插入成功后我进行查找得出刚刚插入的结果：

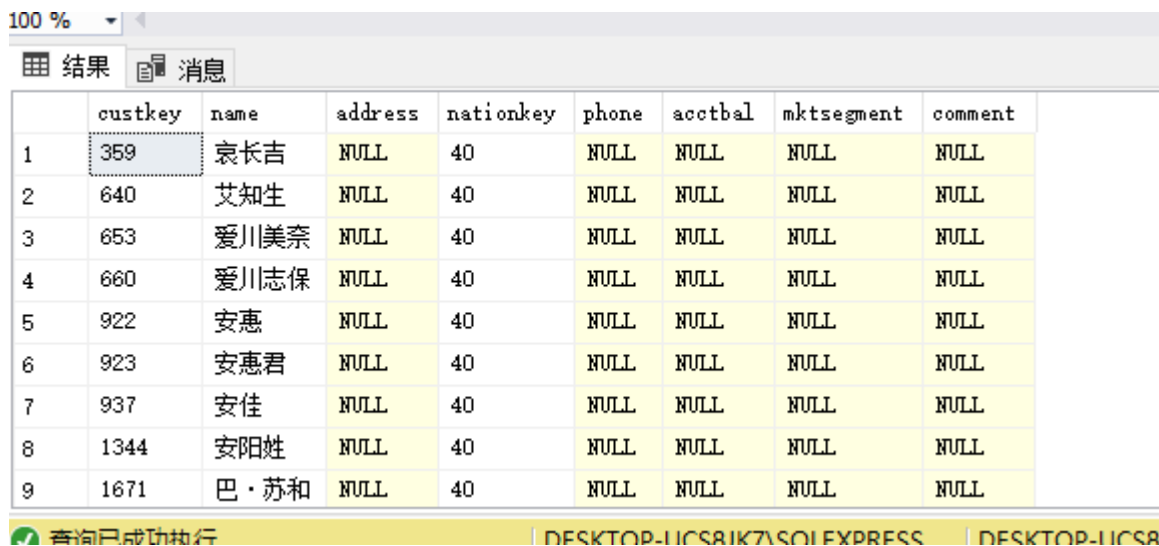
4999	7531	10911	11	10	NULL	NULL	NULL	NULL	NULL	2012-03-06	NULL
------	------	-------	----	----	------	------	------	------	------	------------	------

批量数据 INSERT 语句

①创建一个新的顾客表，把所有中国籍顾客插入到新的顾客表中。

```
SELECT * INTO NewCustomer FROM Customer;
DELETE
FROM NewCustomer;
INSERT INTO NewCustomer
SELECT C.*
FROM Customer C,Nation N
WHERE C.nationkey=N.nationkey AND N.name='中国';
```

这里又双叒是 SQL Server 不支持的语法，所以我查阅资料替换成支持的语法改写了这道题。结果如下：



	custkey	name	address	nationkey	phone	acctbal	mktsegment	comment
1	359	袁长吉	NULL	40	NULL	NULL	NULL	NULL
2	640	艾知生	NULL	40	NULL	NULL	NULL	NULL
3	653	爱川美奈	NULL	40	NULL	NULL	NULL	NULL
4	660	爱川志保	NULL	40	NULL	NULL	NULL	NULL
5	922	安惠	NULL	40	NULL	NULL	NULL	NULL
6	923	安惠君	NULL	40	NULL	NULL	NULL	NULL
7	937	安佳	NULL	40	NULL	NULL	NULL	NULL
8	1344	安阳姓	NULL	40	NULL	NULL	NULL	NULL
9	1671	巴·苏和	NULL	40	NULL	NULL	NULL	NULL

②创建一个顾客购物统计表，记录每个顾客及其购物总数和总价等信息。

```
CREATE TABLE ShoppingStat
(
```

```

    custkey INTEGER,
    quantity REAL,
    totalprice REAL
);
INSERT INTO ShoppingStat
SELECT C.custkey,Sum(L.quantity),Sum(O.totalprice)
FROM Customer C,Orders O,Lineitem L
WHERE C.custkey=O.custkey AND O.orderkey=L.orderkey
GROUP BY C.custkey;

```

结果如下:

	custkey	quantity	totalprice
1	8	99	945741.4
2	125	130	4.202373E+08
3	139	135	1249493
4	172	38	1058524
5	340	125	1.154902E+07
6	423	14	3360428
7	453	10	1728455
8	557	13	1772043
9	574	78	4805468

③倍增零件表的数据,多次重复执行,直到总记录数达到 50 万为止。

```

INSERT INTO Part
SELECT partkey+(SELECT COUNT(*)+1 FROM
Part),name,mfgr,brand,type,size,container,retailprice,comment
FROM Part;

```

这里实验书原句存在插入重复的问题,所以我又修改了一下,结果如下:

	partkey	name	mfgr	brand	type	size	container	retailprice	comment
1	1	竹炭空气清新篮	郑州市荥阳通用阀门厂	NULL	22cm×Φ19cm/只	NULL	NULL	3	NULL
2	2	竹炭净化包	江西赣州阀门厂	NULL	100g/包	NULL	NULL	3.5	NULL
3	3	小炭包	郑州市蝶阀厂	NULL	40g/包	NULL	NULL	4	NULL
4	4	无纺布鞋塞	河南省郑州市上街蝶阀厂	NULL	100g/对	NULL	NULL	5	NULL
5	5	小挂包	江苏省无锡市阀门厂	NULL	二个装/50g/包	NULL	NULL	5.2	NULL
6	6	居室除味宝	南通市疏水阀厂	NULL	100g/包	NULL	NULL	5.2	NULL
7	7	学生干爽鞋垫	苏州阀门厂	NULL	32码-45码	NULL	NULL	5.5	NULL
8	8	冰箱除味包	淮阴市清江化工阀门总厂	NULL	150g/盒	NULL	NULL	6	NULL
9	9	竹炭保暖鞋垫	山东省青岛高中压阀门厂	NULL	35码-47码	NULL	NULL	6	NULL

✓ 查询已成功执行。 | DESKTOP-UCS8JK7\SQLEXPRESS ... | DESKTOP-UCS8JK7\映雪 (54) | TPCB | 00:00:00 | 1000 行

UPDATE 语句（修改部分记录的部分列值）

“金仓集团”供应的所有零件的供应成本价下降 10%。

```
UPDATE PartSupp
SET supplycost=supplycost *0.9
WHERE suppkey=(
    SELECT suppkey
    FROM Supplier
    WHERE name='上海黎顺服装经营部'
);
```

之前的实验有提到过“金仓集团”并不存在在官方提供的供应商的表中，所以我换成了一个表里存在的供应商。

结果					消息			
	partkey	suppkey	availqty	supplycos		suppkey	name	address
1	1	1	13	0				
2	2	1	0	0	1	1	上海黎顺服装经营部	上海市

UPDATE 语句（利用一个表中的数据修改另外一个表中的数据）

```
UPDATE Lineitem
SET Lineitem.extendedprice = P.retailprice * Lineitem.quantity
FROM Part P
WHERE Lineitem.partkey=P.partkey
```

结果如下：

结果		消息								
	orderkey	partkey	suppkey	linenumber	quantity	extendedprice	discount	tax	returnflag	lin
1	1	44930	18848	1	61	33489	0	0.08714417	NULL	NUL
2	1	42522	13503	2	46	234600	0	0.272896	NULL	NUL
3	2	5001	15634	1	6	7020000	0	0.1504321	NULL	NUL
4	2	15802	29470	2	81	1.4013E+07	0	0.967502	NULL	NUL
5	3	10672	26689	1	52	7488000	0	0.248013	NULL	NUL
6	3	13973	4191	2	82	1.3284E+07	0	0.1297093	NULL	NUL
7	4	26000	1590	1	67	173530	0	0.8972544	NULL	NUL
8	4	36680	24313	2	34	135932	0	0.9525895	NULL	NUL

DELETE 基本语句（删除给定条件的所有记录）

删除顾客曹操的所有订单记录。

```
DELETE FROM Lineitem
WHERE orderkey IN (
    SELECT orderkey
    FROM Orders O, Customer C
    WHERE O.custkey = C.custkey AND C.name='曹操'
);
DELETE FROM Orders
WHERE custkey = (
    SELECT custkey
    FROM Customer
    WHERE name='曹操'
);
```

这里原本书上是张三，但数据库中张三并没有买任何东西……所以我选择了删除买了东西的“曹操”的记录，终于由“0 行记录受影响”变成：

— 1 —
(1 行受影响)

曹操的记录被删除了。

实验总结

插入和删除以及更新这个实验做得我内牛满面。几乎每一条书上的语句的语法都需要我去查资料修改（不同 DBMS 支持的语句不一样真的是一个坑）。不过，通过对这些语句的修改，我对主键、外键约束的限制有了更深的了解。同时也了解到了 SQL Server 中对于某些操作的特定表达，以后使用不同的 DBMS 仍然要注意语句表达不同的问题。同时在插入的时候要注意外键、主键的限制，在删除的时候要根据表之间的关系注意顺序。

实验 1.5 视图实验

实验题目

实验 1.5 视图实验

实验目的

熟悉 SQL 语言有关视图的操作，能够熟练使用 SQL 语句来创建需要的视图，定义数据库外模式，并能使用所创建的视图实现数据管理。

实验内容和要求

针对给定的数据库模式，以及相应的应用需求，创建视图、创建带 WITH CHECK OPTION 的视图，并验证视图 WITH CHECK OPTION 选项的有效性。理解和掌握视图消解执行原理，掌握可更新视图和不可更新视图的区别。

实验重点和难点

实验重点：创建视图。

实验难点：可更新的视图和不可更新的视图之区别， WITH CHECK OPTION 的验证。

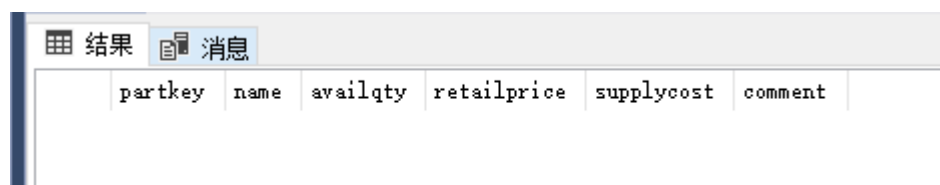
实验操作和结果

创建视图（省略视图列名）

创建一个“海大汽配”供应商供应的零件视图 V_DLMU_PartSupp1,要求列出供应商的编号、零件名称、可用数量、零售价格、供应价格和备注等信息。

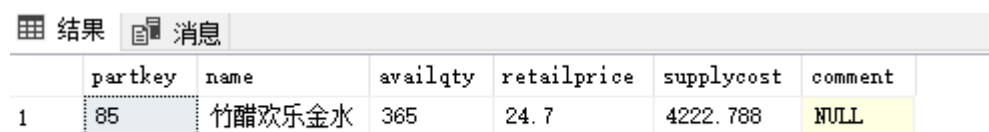
```
CREATE VIEW V_DLMU_PARTSUPP1 AS
SELECT P.partkey,P.name,PS.availqty,P.retailprice,PS.supplycost,P.comment
FROM Part P,PartSupp PS,Supplier S
WHERE P.partkey=PS.partkey AND S.supkey=PS.supkey AND S.name='海大汽配';
```

由于“海大汽配”的零件为空，所以创建的视图是空集：



The screenshot shows a database interface with two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying an empty table with the following columns: partkey, name, availqty, retailprice, supplycost, and comment. The table is currently empty, reflecting the fact that no parts are supplied by '海大汽配'.

将名字换为有信息记录的“上海天责投资咨询有限公司”，视图中有了数据：



The screenshot shows the same database interface, but now the '结果' tab displays a single row of data. The columns are partkey, name, availqty, retailprice, supplycost, and comment. The data row is as follows:

	partkey	name	availqty	retailprice	supplycost	comment
1	85	竹醋欢乐金水	365	24.7	4222.788	NULL

创建视图（不能省略列名的情况）

创建一个视图 V_CustAvgOrder，按顾客统计平均每个订单的购买金额和零件数量，要求输出顾客编号，姓名，平均购买金额和平均购买零件数量。

```
CREATE VIEW V_CustAvgOrder(custkey,cname,avgprice,avgquantity) AS
SELECT C.custkey,MAX (C.name),AVG(O.totalprice),AVG(L.quantity)
FROM Customer C,Orders O,Lineitem L
WHERE C.custkey=O.custkey AND L.orderkey=O.orderkey
GROUP BY C.custkey
```

视图的结果如下所示：

	custkey	cname	avgprice	avgquantity
1	8	阿波罗	472870.71875	49.5
2	125	阿盖郡主	140079104	43.33333333333333
3	139	阿怪	624746.3125	67.5
4	172	阿坚	1058524.25	38
5	340	阿永	5774512	62.5
6	423	艾蒿	3360427.5	14
7	453	艾锦亮	1728455.375	10
8	557	艾微儿	1772043.25	13
9	574	艾芜	1601822.65234375	26

创建视图 (WITH CHECK OPTION)

使用 WITH CHECK OPTION, 创建一个“海大汽配”供应商供应的零件视图, 要求列出供应零件的编号、可用数量和供应价格等信息。然后通过该视图分别增加、删除和修改一条“海大汽配”零件供应记录, 验证 WITH CHECK OPTION 是否起作用。

```
CREATE VIEW V_DLMU_PartSupp2
AS
SELECT partkey,suppkey,availqty,supplycost
FROM PartSupp
WHERE suppkey = (
    SELECT suppkey
    FROM Supplier
    WHERE name='海大汽配'
)
WITH CHECK OPTION;

INSERT INTO V_DLMU_PartSupp2
VALUES(58889,5048,704,77760);

UPDATE V_DLMU_PartSupp2
SET supplycost = 12
WHERE suppkey=58889;

DELETE FROM V_DLMU_PartSupp2
WHERE suppkey=58889;
```

结果如下:

```
消息 550, 级别 16, 状态 1, 第 1 行
试图进行的插入或更新已失败, 原因是目标视图或者目标视图所跨越的某一视图指定了 WITH CHECK OPTION, 而该操作的一个或多个结果行又不符合该条件。
语句已终止。
```

可见 WITH CHECK OPTION 生效，对于此视图无法进行插入删除和更新操作。

可更新的视图（行列子集视图）

再次创建上述视图，但不添加 WITH CHECK OPTION。

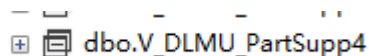
```
CREATE VIEW V_DLMU_PartSupp4
AS
SELECT partkey,suppkey,availqty,supplycost
FROM PartSupp
WHERE suppkey = (
    SELECT suppkey
    FROM Supplier
    WHERE name='海大汽配'
);

INSERT INTO V_DLMU_PartSupp4
VALUES(58889,5048,704,77760);

UPDATE V_DLMU_PartSupp4
SET supplycost = 12
WHERE suppkey=58889;

DELETE FROM V_DLMU_PartSupp4
WHERE suppkey=58889;
```

创建视图成功：



执行插入操作：

	partkey	suppkey	availqty	supplycost	comment
1	58889	1622	550	3823.194	NULL
2	58889	5048	704	77760	NULL
3	58889	23865	469	7608.931	NULL

不可更新的视图

(2) 中创建的视图是可更新的吗？

```
INSERT INTO V_CustAvgOrder
VALUES(100000,NULL,20,2000);
```

(2) 中创建的视图，有计算列，即有 MAX、AVG 等计算操作，所以是不可更新的，结果自然是不可更新：

消息 4406, 级别 16, 状态 1, 第 1 行
对视图或函数 'V_CustAvgOrder' 的更新或插入失败，因其包含派生域或常量域。

删除视图

```
CREATE VIEW V_CustOrd(custkey,cname,qty,extprice)
AS
SELECT C.custkey,C.name,L.quantity,L.extendedprice
FROM Customer C,Orders O,Lineitem L
WHERE C.custkey=O.custkey AND O.orderkey=L.orderkey;

CREATE VIEW V_CustAvgOrder(custkey,cname,avgqty,avgprice)
AS
SELECT custkey,MAX(cname),AVG(qty),AVG(extprice)
FROM V_CustOrd
GROUP BY custkey;

DROP VIEW V_CustOrd RESTRICT;
DROP VIEW V_CustOrd CASCADE;
```

创建 V_CustOrd 和 V_CustAvgOrder 成功：



系统视图

- dbo.V_CustAvgOrder
- dbo.V_CustOrd
- dbo.V_DLMU_PartSu
- dbo.V_DLMU_PartSu

但是 sql server 不支持删除时 restrict 和 cascade，如果删去 restrict，发现可以直接删除。可见这个 DBMS 默认是可以级联删除的。

实验总结

通过视图实验，我对视图的使用和各个情况有了更深的理解。但是也再一次感受到不同 DBMS 支持的语句不同的现象，比如实验最后的 restrict 和 cascade，代表不可级联

删除和可删除，但是 SQL Server 不支持这个语法。视图的本质其实是一个虚表，主要是方便查询，如果需要进行大范围的改动和插入，我还是认为直接在数据集上操作是最佳的。