数据库实验报告

实 验 人： 伍建霖 学 号： 20337251 日 期：

院（系）： 计算机学院 专业（班级）： 网络空间安全

联系方式： QQ 773542531

实验题目： 1.4 数据更新实验

一. 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。

二. 实验内容和要求

针对 TPC-H 数据库设计单个元组插入、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。

三. 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。

四. 实验工具

MySQL、SQL Server、Navicat

五. 实验过程

（1）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。如：values(30303030, '张三', '北京市',40,'010-51001199',0.00, 'Northeast', 'VIP Customer')

```
INSERT INTO customer
values(30303030, '张三', '北京市',40,'010-51001199',0.00, 'Northeast', 'VIP Customer')
```

将其结果查询后截屏如下：

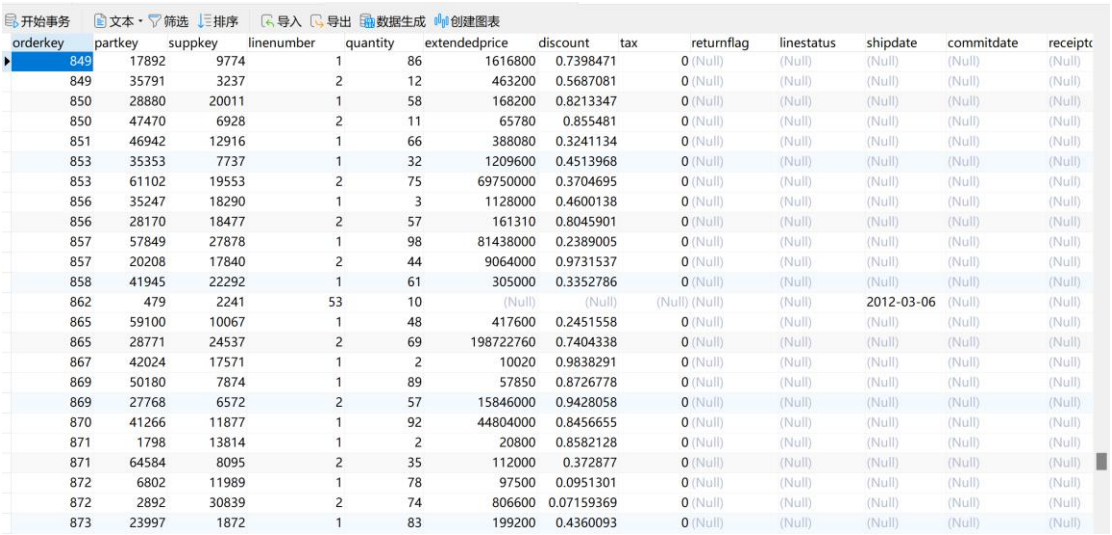
SELECT * FROM customer WHERE custkey= 30303030;

（2）insert 基本语句（插入部分列的数据） lineitem

插入一条订单记录，给出必要的几个字段值。如：(ORDERKEY,LINENUMBER,PARTKEY,SUPPKEY,QUANTITY,SHIPDATE)values(862,ROUND(RAND()*100,0),479,2241,10,'2012-3-6')

INSERT INTO lineitem (ORDERKEY,LINENUMBER,PARTKEY,SUPPKEY,QUANTITY,SHIPDATE)values(862,ROUND(RAND()*100,0),479,2241,10,'2012-3-6')

将其结果查询后截屏如下：



orderkey	partkey	suppkey	linenumber	quantity	extendedprice	discount	tax	returnflag	linestatus	shipdate	commitdate	receipt
849	17892	9774	1	86	1616800	0.7398471	0	(Null)	(Null)	(Null)	(Null)	(Null)
849	35791	3237	2	12	463200	0.5687081	0	(Null)	(Null)	(Null)	(Null)	(Null)
850	28880	20011	1	58	168200	0.8213347	0	(Null)	(Null)	(Null)	(Null)	(Null)
850	47470	6928	2	11	65780	0.855481	0	(Null)	(Null)	(Null)	(Null)	(Null)
851	46942	12916	1	66	388080	0.3241134	0	(Null)	(Null)	(Null)	(Null)	(Null)
853	35353	7737	1	32	1209600	0.4513968	0	(Null)	(Null)	(Null)	(Null)	(Null)
853	61102	19553	2	75	69750000	0.3704695	0	(Null)	(Null)	(Null)	(Null)	(Null)
856	35247	18290	1	3	1128000	0.4600138	0	(Null)	(Null)	(Null)	(Null)	(Null)
856	28170	18477	2	57	161310	0.8045901	0	(Null)	(Null)	(Null)	(Null)	(Null)
857	57849	27878	1	98	81438000	0.2389005	0	(Null)	(Null)	(Null)	(Null)	(Null)
857	20208	17840	2	44	9064000	0.9731537	0	(Null)	(Null)	(Null)	(Null)	(Null)
858	41945	22292	1	61	305000	0.3352786	0	(Null)	(Null)	(Null)	(Null)	(Null)
862	479	2241	53	10	(Null)	(Null)	(Null)	(Null)	(Null)	2012-03-06	(Null)	(Null)
865	59100	10067	1	48	417600	0.2451558	0	(Null)	(Null)	(Null)	(Null)	(Null)
865	28771	24537	2	69	198722760	0.7404338	0	(Null)	(Null)	(Null)	(Null)	(Null)
867	42024	17571	1	2	10020	0.9838291	0	(Null)	(Null)	(Null)	(Null)	(Null)
869	50180	7874	1	89	57850	0.8726778	0	(Null)	(Null)	(Null)	(Null)	(Null)
869	27768	6572	2	57	15846000	0.9428058	0	(Null)	(Null)	(Null)	(Null)	(Null)
870	41266	11877	1	92	44804000	0.8456655	0	(Null)	(Null)	(Null)	(Null)	(Null)
871	1798	13814	1	2	20800	0.8582128	0	(Null)	(Null)	(Null)	(Null)	(Null)
871	64584	8095	2	35	112000	0.372877	0	(Null)	(Null)	(Null)	(Null)	(Null)
872	6802	11989	1	78	97500	0.0951301	0	(Null)	(Null)	(Null)	(Null)	(Null)
872	2892	30839	2	74	806600	0.07159369	0	(Null)	(Null)	(Null)	(Null)	(Null)
873	23997	1872	1	83	199200	0.4360093	0	(Null)	(Null)	(Null)	(Null)	(Null)

说明：由于在创建八个基本表的时候，已将所有列添加了约束：非空（不能为 NULL 值），因此在给 lineitem 插入部分列的时候会失败。

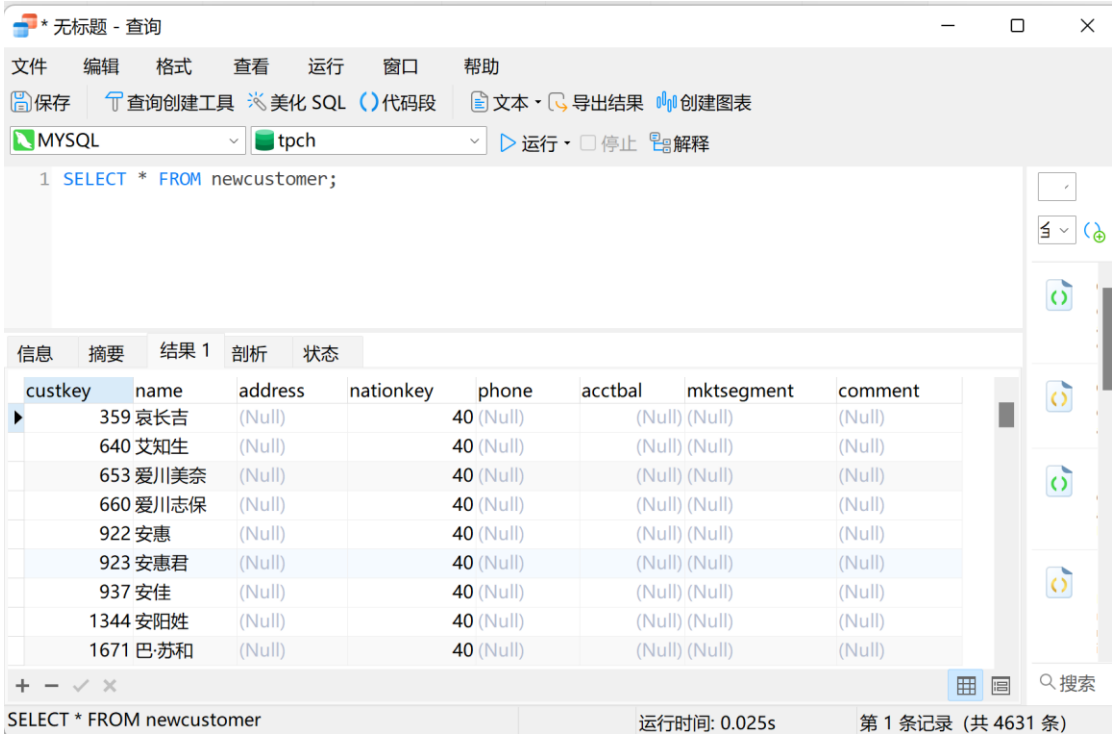
（3）批量数据 insert 语句

1. 创建一个新的顾客表 NewCustomer，把所有来自“中国”的顾客插入到这个表 NewCustomer

```
CREATE TABLE NewCustomer LIKE Customer;

INSERT INTO newcustomer

SELECT *

FROM customer

WHERE customer.nationkey = 40;
```

将其结果查询后部分截屏如下：

```
SELECT * FROM newcustomer;
```

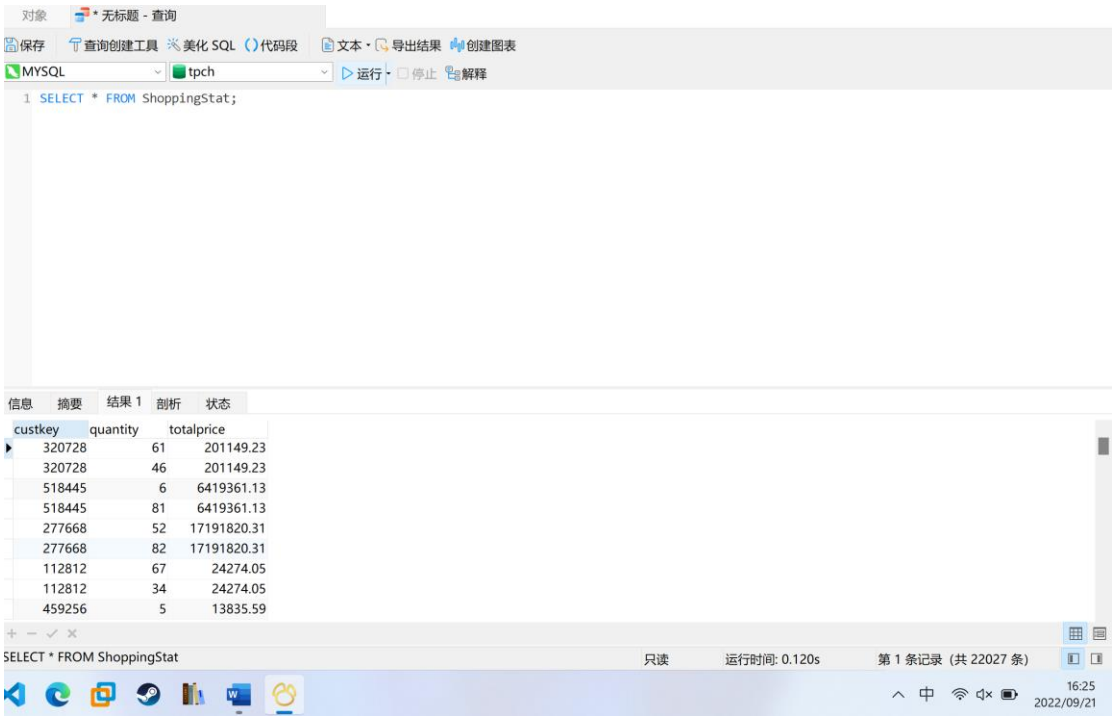


2. 创建一个顾客购物统计表，记录每个顾客及其购物总数和总价等信息。

```
CREATE TABLE ShoppingStat(custkey INTEGER, quantity REAL, totalprice REAL);

INSERT INTO shoppingstat

SELECT orders.custkey, lineitem.quantity, orders.totalprice

FROM orders, lineitem

WHERE orders.orderkey = lineitem.orderkey;
```

将其结果查询后截屏如下：

```
SELECT * FROM ShoppingStat;
```



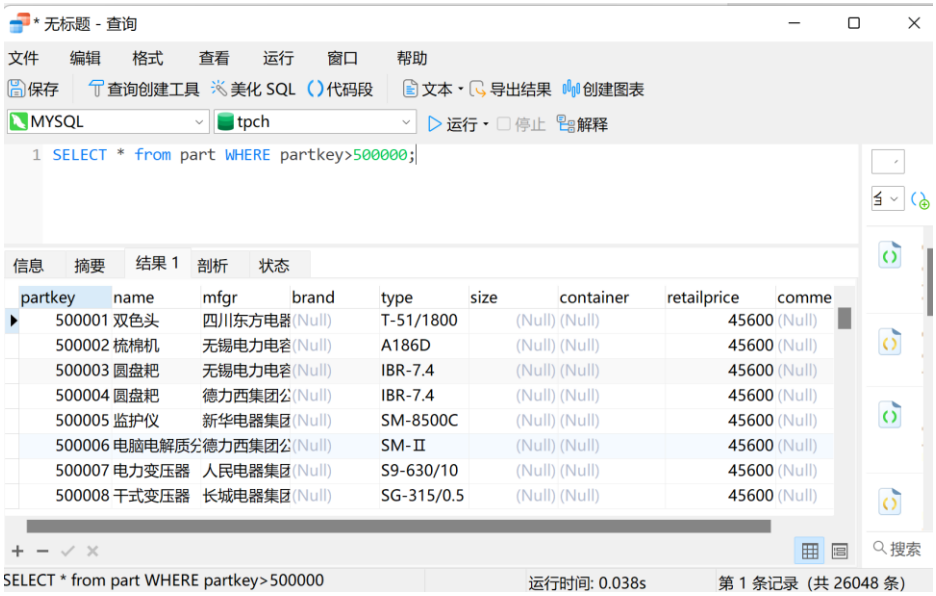
3. 倍增零件表的数据，多次重复执行，直到总记录数达到 50 万为止。

```
insert into PART
select PARTKEY+(select max(partkey) from PART), NAME, MFGR, BRAND, TYPE, SIZE,
CONTAINER, RETAILPRICE, COMMENT
from PART;
```

多次执行上述语句后，观察表格可以发现，其最终结果满足要求。

将其结果查询后截屏如下：

```
SELECT * from part WHERE partkey>500000;
```



（4）update 语句（修改部分记录的部分列值）

将“双思 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 in (SELECT suppkey FROM Supplier WHERE name='双思 集团') ;
```

部分查询结果截屏如下：

```
select *
from PartSupp
WHERE suppkey in (SELECT suppkey FROM Supplier WHERE name='双思 集团') ;
```

1 更新前：

2 更新后：

比较两者的 SUPPLYCOST 可以发现，SUPPLYCOST 的供应成本价较 update 前确实下降

了 10%，其最终结果满足要求。

（5）update 语句（利用一个表的数据修改另外一个表的数据）

利用 part 表中的零售价格来修改 lineitem 中的 extendedprice，其中 extendedprice = part.retailprice*quantity。

```
update LINEITEM L INNER JOIN PART P USING(partkey)
set L.EXTENDEDPRICE = P.RETAILPRICE * L.QUANTITY
```

或者

```
UPDATE Lineitem
SET extendedprice = quantity * (select retailprice from part where partkey=Lineitem.partkey);
```

（6）delete 基本语句（删除给定条件的所有记录）

删除顾客“阿波罗”的所有订单记录。

```
DELETE FROM lineitem
 WHERE orderkey
  IN (SELECT orderkey
        FROM orders WHERE custkey
         IN(SELECT custkey
          FROM customer
           WHERE name='阿波罗'));
DELETE FROM orders
 WHERE orders.custkey
  IN (SELECT custkey
   FROM customer
    WHERE name='阿波罗');
```

再查询‘阿波罗’的订单记录，结果如下：

```
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='阿波罗';
```

说明：结果表明，已经找不到该顾客的订单记录，说明删除该顾客的订单记录成功。

六. 与实验结果相关的文件

无

七. 实验总结