**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：** | | | **1.5 视图** | | | | | |

* + - 1. **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。

* + - 1. **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，创建视图和带 with check option的视图，并验证视图 with check option 选项的有效性。理解和掌握视图执行原理，掌握可更新视图和不可更新视图的区别。

* + - 1. **实验重点和难点**

实验重点：创建视图

实验难点：可更新视图和不可更新视图的区别，with check option 的验证。

* + - 1. **实验工具**

MySQL、SQL Server、Navicat

* + - 1. **实验过程**

（1）创建视图（省略视图列名）

创建一个‘上海黎顺服装经营部’供应商供应的零件视图 V\_DLMU\_PEATSUPP1，要求列出供应零件的编号，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

WHERE

P.Partkey = PS.partkey

AND PS.Suppkey IN (

SELECT

Suppkey

FROM

Supplier S

WHERE

S.NAME = '上海黎顺服装经营部')

查询该视图后截屏如下：

SELECT \* FROM V\_DLMU\_PARTSUPP1;



（2）创建视图（不能省略视图列名的情况）

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, NAME, Avgprice, AVGQUANTITY ) AS SELECT

O.Custkey,

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

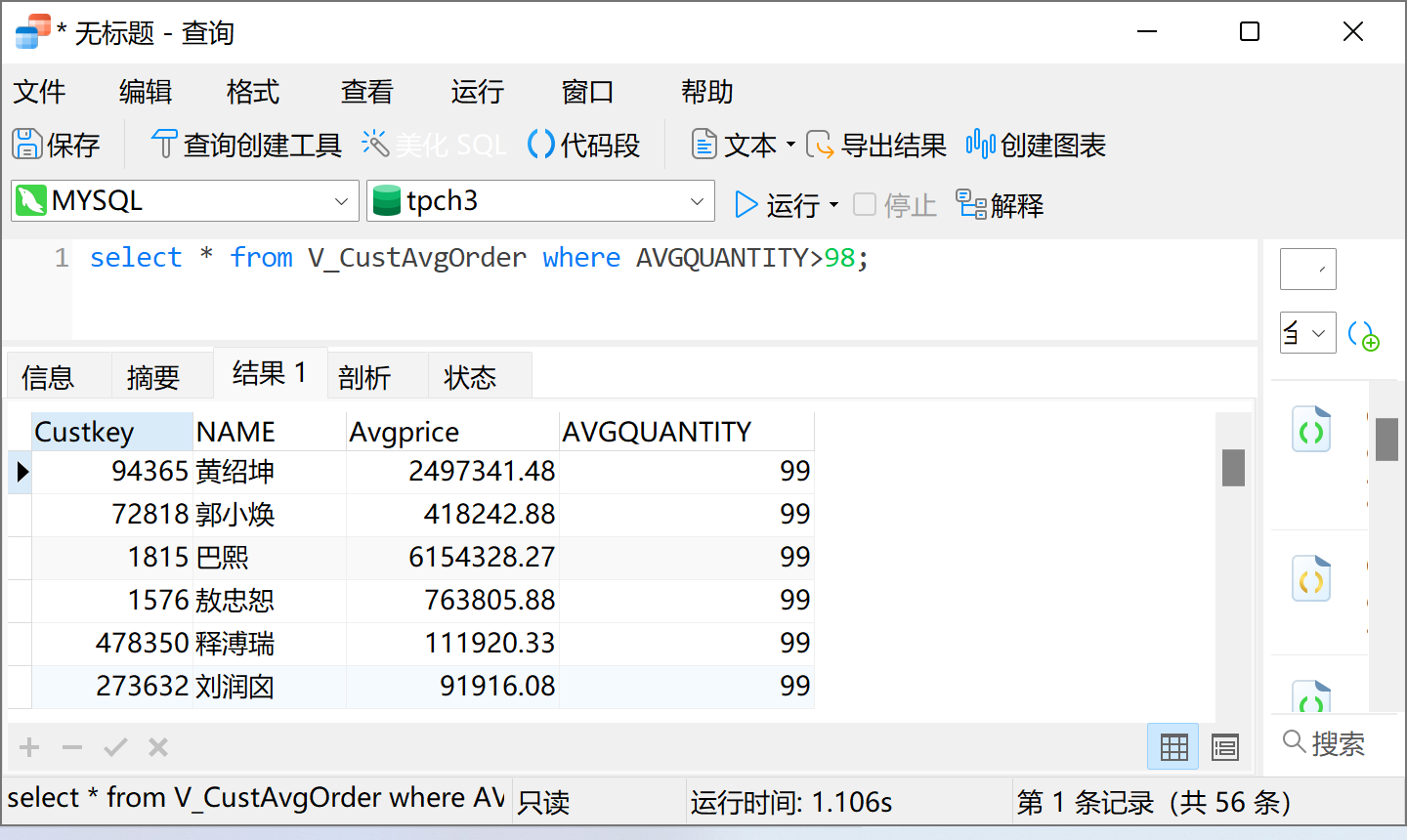
GROUP BY

O.Custkey,

C.NAME;

查询该视图中 AVGQUANTITY>98 的所有信息后部分截屏如下：

select \* from V\_CustAvgOrder where AVGQUANTITY>98;



（3）创建视图（with check option）

使用 with check option，创建一个‘上海黎顺服装经营部’供应商供应的零件视图 V\_DLMU\_PARTSUPP2，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‘上海黎顺服装经营部’零件供应记录，验证 with check option 是否起作用。

1 创建视图

CREATE VIEW V\_DLMU\_PartSupp2

AS

SELECT Partkey,suppkey,Availqty,Supplycost

FROM PartSupp

WHERE Suppkey = (

SELECT Suppkey

FROM Supplier

WHERE NAME = '上海黎顺服装经营部')

WITH CHECK OPTION;

2 向该视图增加零件供应记录

* 情况1（values(58889,5048,704,77760)）：失败

INSERT INTO V\_DLMU\_Partsupp2 VALUES(58889,5048,704,77760);



* 情况2（values(58889,1,704,77760)）：成功

INSERT INTO V\_DLMU\_Partsupp2 VALUES(58889,1,704,77760);



select \* from V\_DLMU\_PARTSUPP2



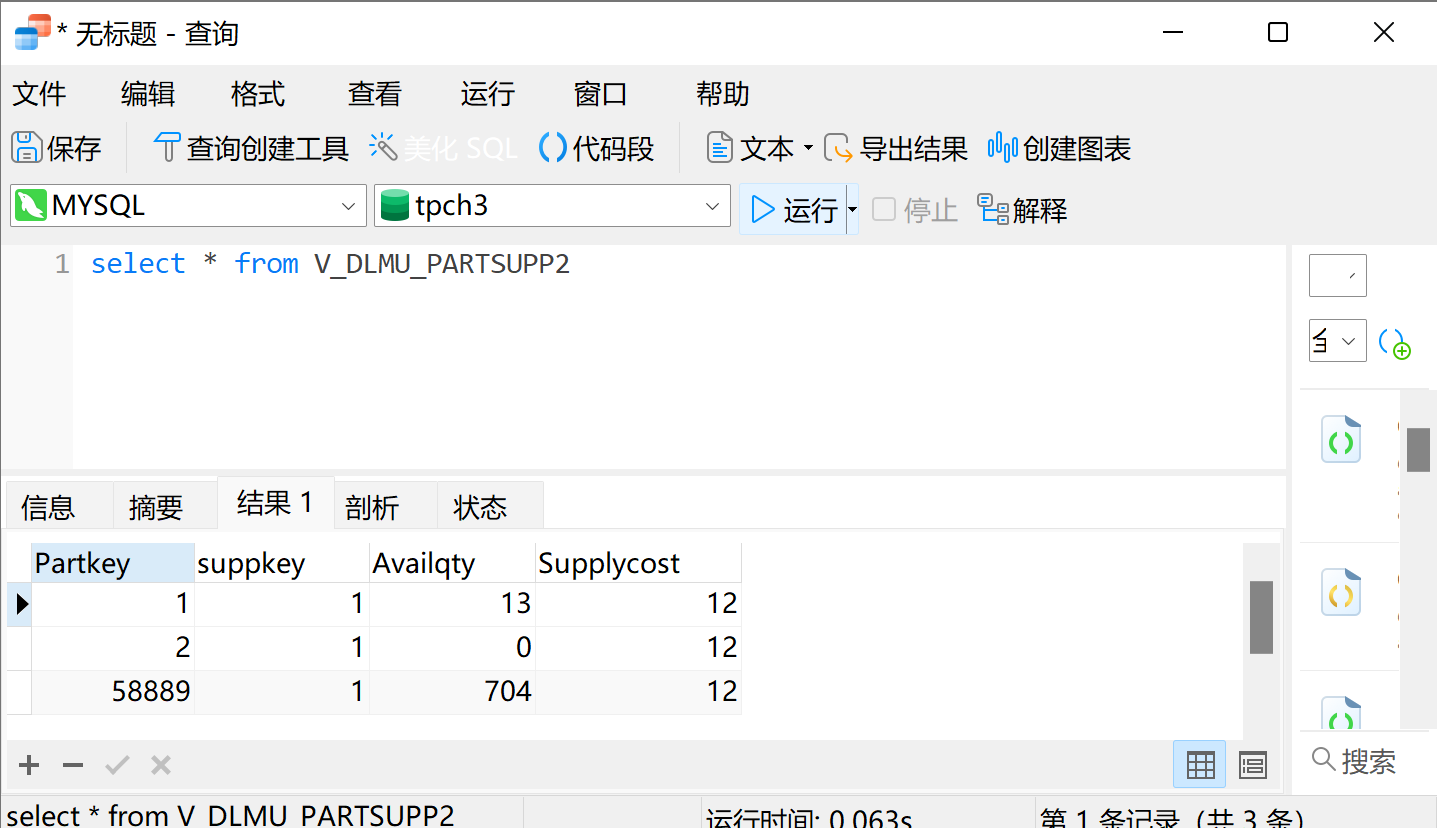
向该视图增加零件供应记录成功！

3 修改该视图的零件供应价格（set supplycost=12 where suppkey=1）

UPDATE V\_DLMU\_PartSupp2 SET Supplycost = 12

WHERE Suppkey = 1;

select \* from V\_DLMU\_PARTSUPP2



修改该视图的零件供应记录成功！

4 删除该视图中零件编号大于100的供应记录（where suppkey=1 and partkey>100）

DELETE FROM V\_DLMU\_PartSupp2

WHERE suppkey=1 and partkey>100;

select \* from V\_DLMU\_PARTSUPP2



删除该视图的零件供应记录成功！

说明：我们在创建视图时指定了 witch check option 关键字，这也就是说，更新后的每一条数据仍然要满足创建视图时指定的 where 条件。因为向视图增加记录，删除记录以及更新视图的零件供应记录都不与当初的 where 条件冲突，因此执行成功了。 即可验证 with check option 起了作用。

（4）可更新的视图（行列子集视图）

创建一个‘上海黎顺服装经营部’供应商供应的零件视图 V\_DLMU\_PEATSUPP3，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‘上海黎顺服装经营部’零件供应记录，验证该视图是否是可更新的，并比较上述“（3）创建视图”实验任务与本任务结果有何异同？

1 创建视图

CREATE VIEW V\_DLMU\_PartSupp3

AS

SELECT partkey,suppkey,availqty,supplycost

FROM partsupp

WHERE suppkey=

(

SELECT suppkey

FROM supplier

WHERE name='上海黎顺服装经营部'

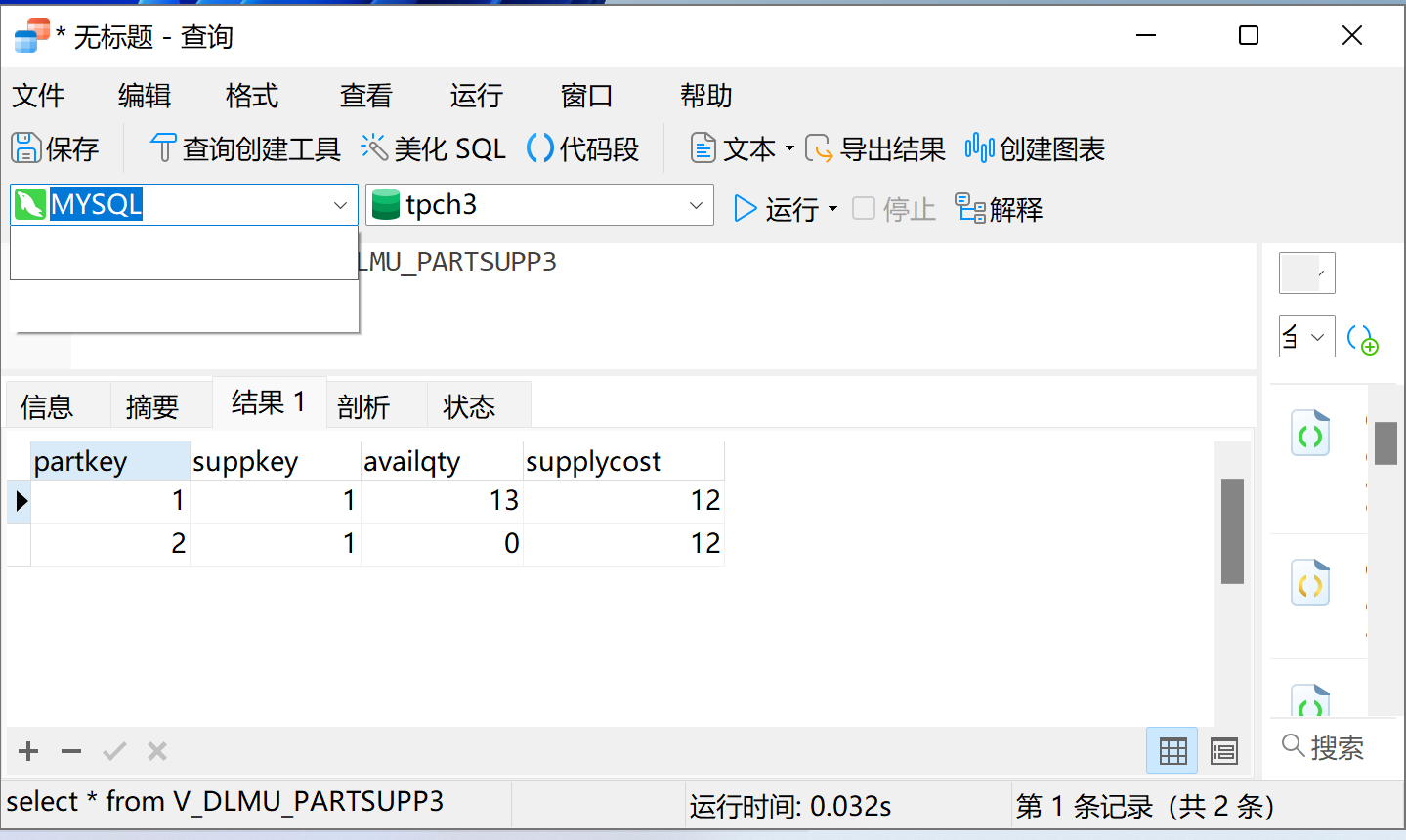
);

2 向该视图增加零件供应记录

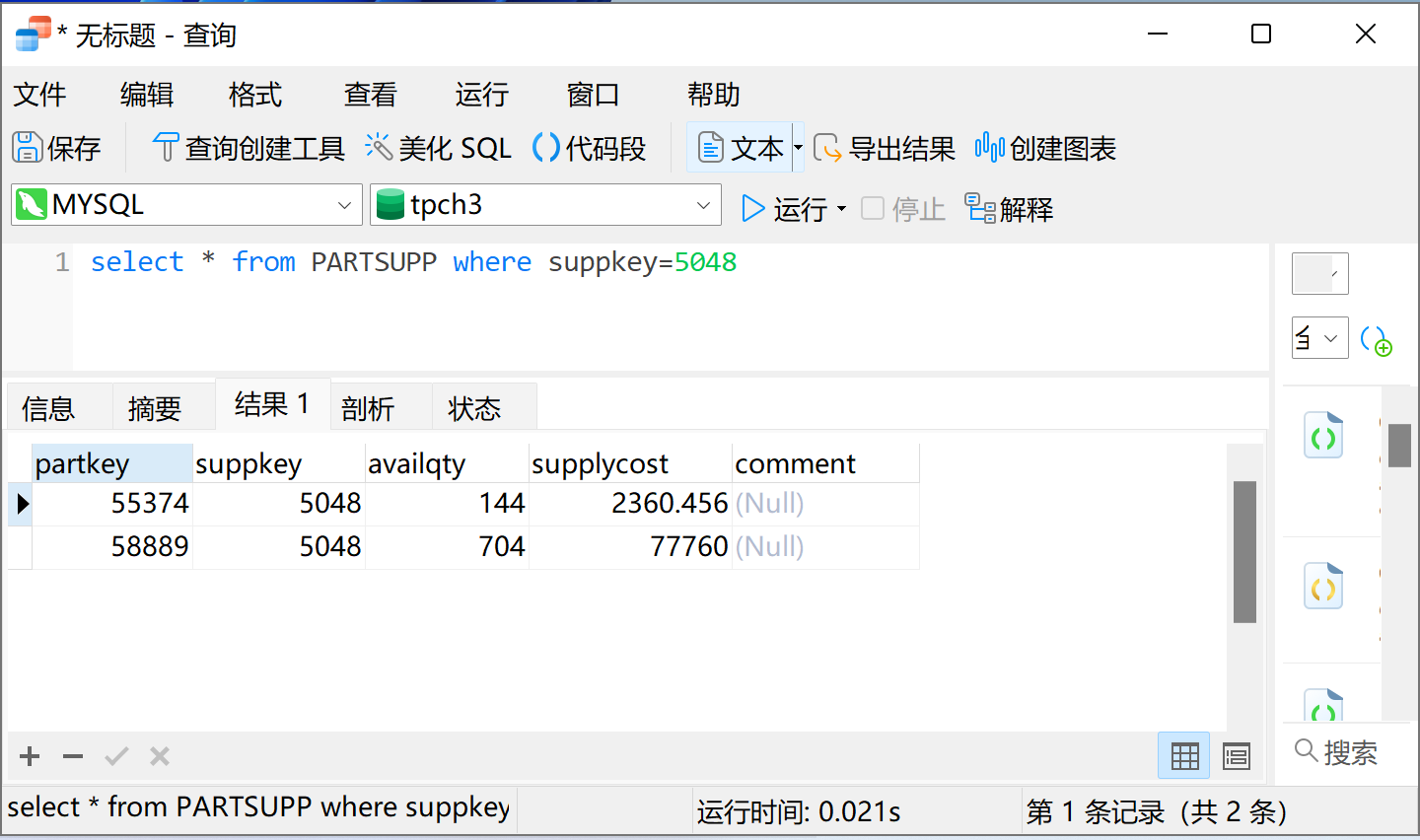
* 情况1（values(58889,5048,704,77760)）：成功，但不能通过视图查询到

INSERT INTO v\_dlmu\_partsupp3 VALUES(58889,5048,704,77760)

select \* from V\_DLMU\_PARTSUPP3



select \* from PARTSUPP where suppkey=5048



* 情况2（values(58889,1,704,77760)）：成功，并且通过视图可以查询到

INSERT INTO v\_dlmu\_partsupp3 VALUES(58889,1,704,77760)

select \* from V\_DLMU\_PARTSUPP3



向该视图增加零件供应记录成功！

3 修改该视图的零件供应价格（set supplycost=20 where suppkey=1）

UPDATE V\_DLMU\_PartSupp3

SET supplycost=20

WHERE partkey=1;

select \* from V\_DLMU\_PARTSUPP3



修改该视图的零件供应记录成功！

4、删除该视图中零件编号大于100的供应记录（where suppkey=1 and partkey>100）

DELETE FROM V\_DLMU\_PartSupp3

WHERE suppkey=1 and partkey>100;

select \* from V\_DLMU\_PARTSUPP3



删除该视图的零件供应记录成功！

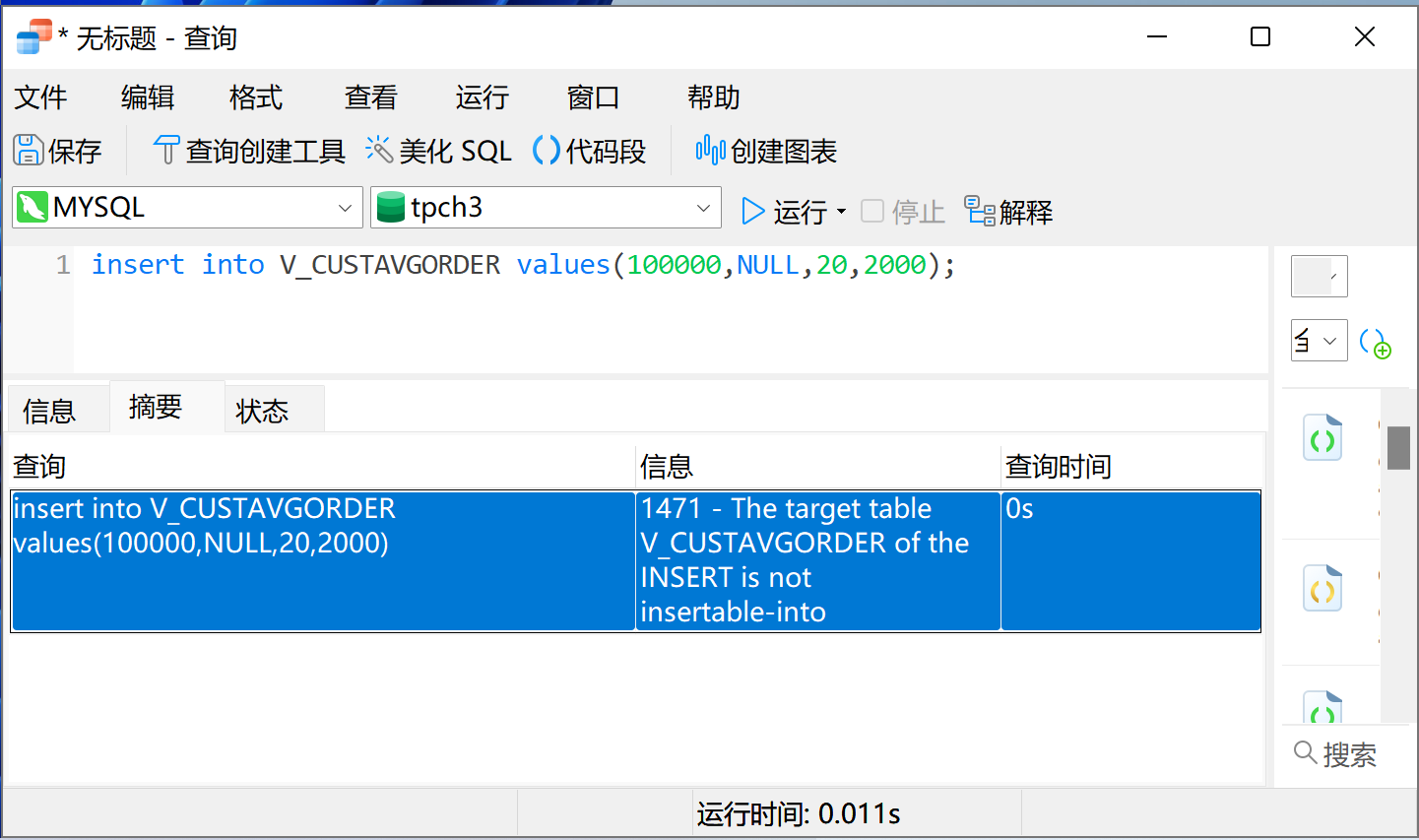
（5）不可更新的视图

（2）中创建的视图是可更新的吗？通过 SQL 更新语句加以验证，并说明原因。

答：

insert into V\_CUSTAVGORDER values(100000,NULL,20,2000);

因为所创建的视图对其属性值进行了计算的其他形式上的改变，而对视图的更改最终表现为对表的更改而表中不存在视图的某一属性，或属性的性质不相同，则无法更改，这是一种视图机制。V\_CustAvgOrder定义了平均购买金额和平均购买零件数量，但是原来的表项中并没有，对它们的更改无法对应到基本表上，所以该视图不可以更新。



（6）删除视图（ restrict/Cascade）

创建顾客订购零件明细视图 V\_CustOrd,要求列出顾客编号、姓名、购买零件数，金额， 然后在该视图的基础上，再创建（2）的视图 V\_CustAvOrder1,然后使用 restrict 选项删除 视图 V\_CustOrd,观察现象并解释原因。利用 cascade 选项删除视图 V\_CustOrd,观察现象并检查 V\_CustAvgOrder 是否存在，解释原因。

1、创建顾客订购零件明细视图 V\_CustOrd

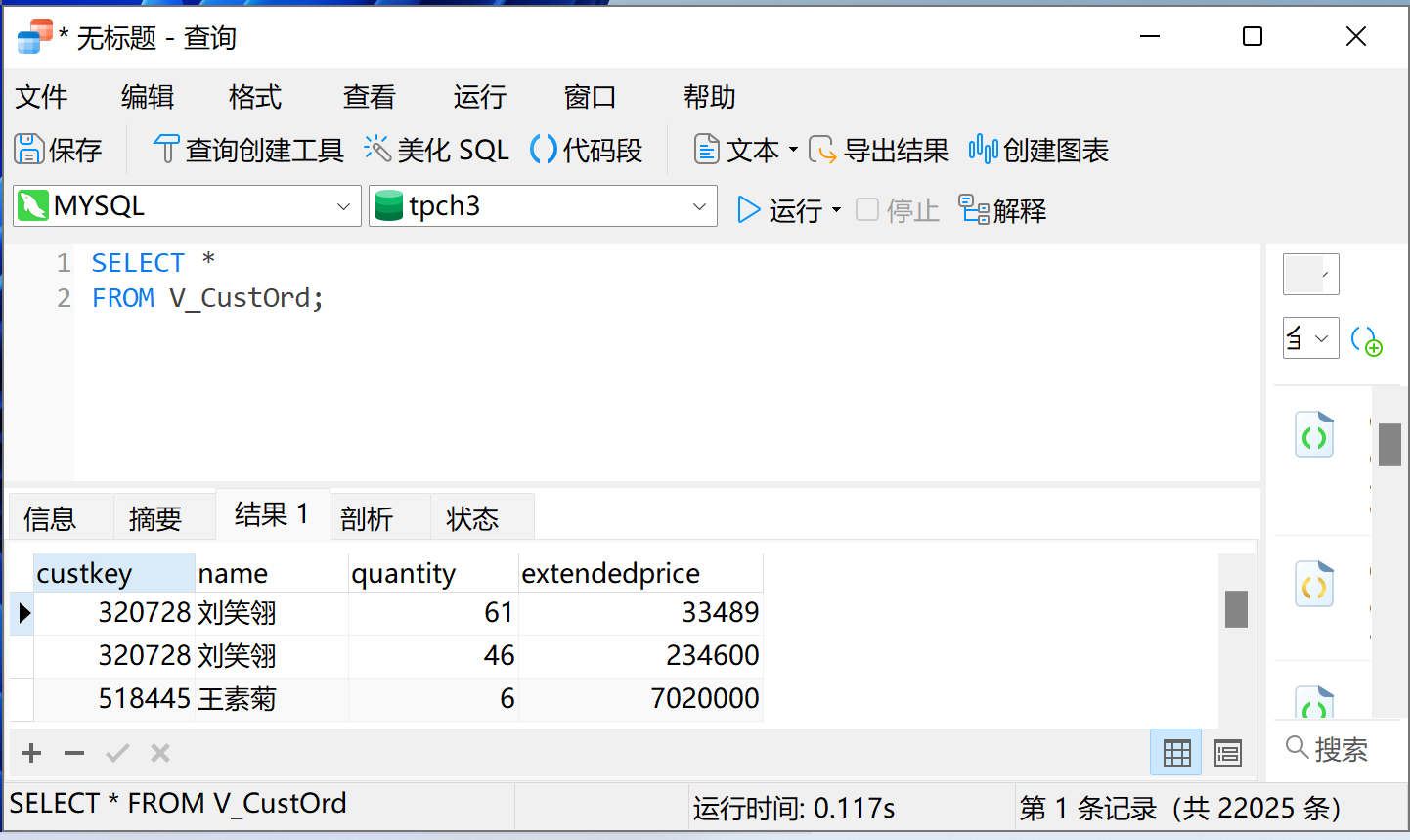
CREATE VIEW V\_CustOrd(custkey,name,quantity,extendedprice)

AS

SELECT customer.custkey,customer.name,quantity,extendedprice

FROM customer,orders,lineitem

WHERE customer.custkey=orders.custkey AND orders.orderkey=lineitem.orderkey;



2、基于视图 V\_CustOrd 创建视图 V\_CustAvOrder1

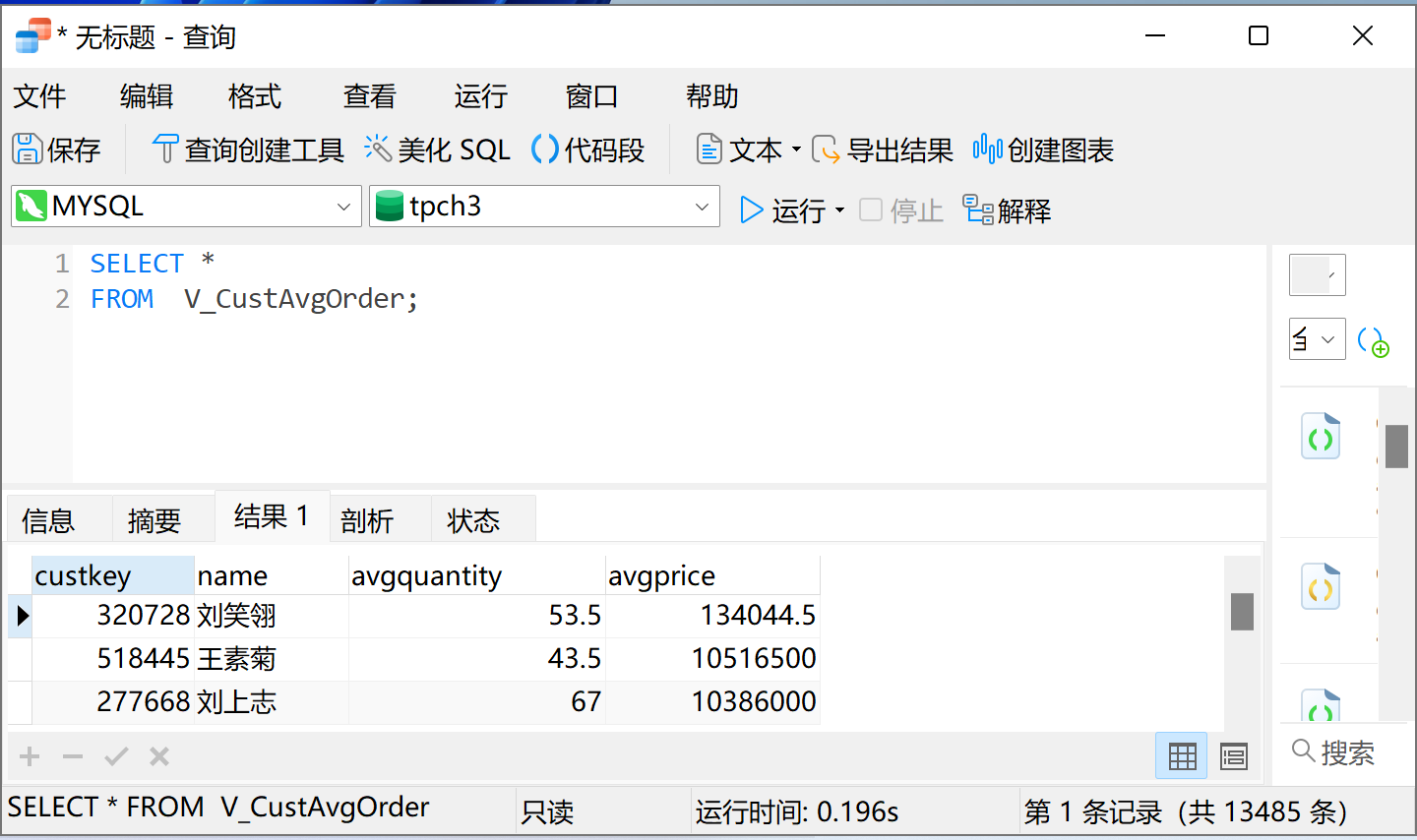
CREATE VIEW V\_CustAvgOrder(custkey,name,avgquantity,avgprice)

AS

SELECT custkey,name,AVG(quantity),AVG(extendedprice)

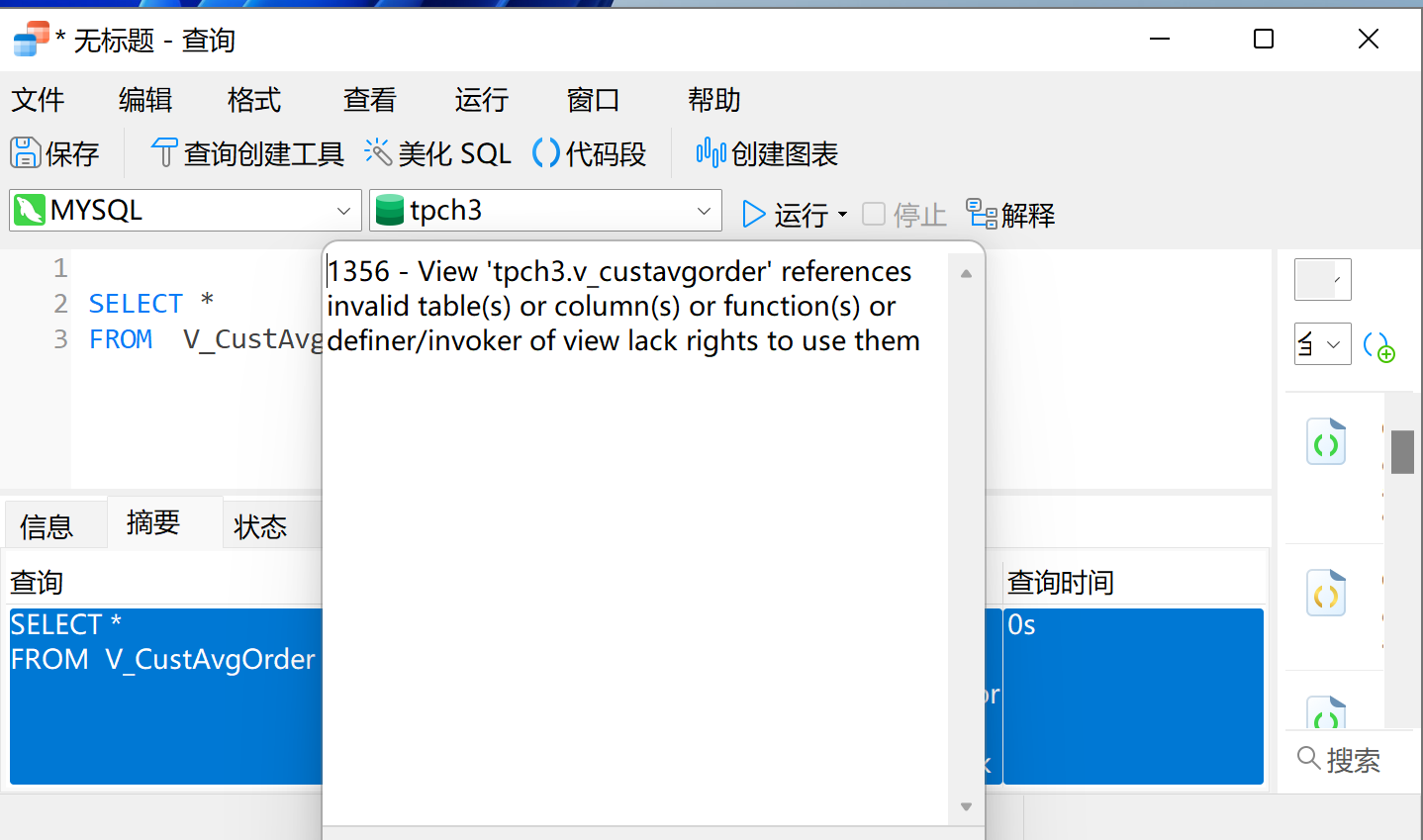
FROM V\_CustOrd

GROUP BY custkey,name;



* + 1. 使用 restrict 选项删除视图 V\_CustOrd

DROP VIEW V\_CustOrd RESTRICT;



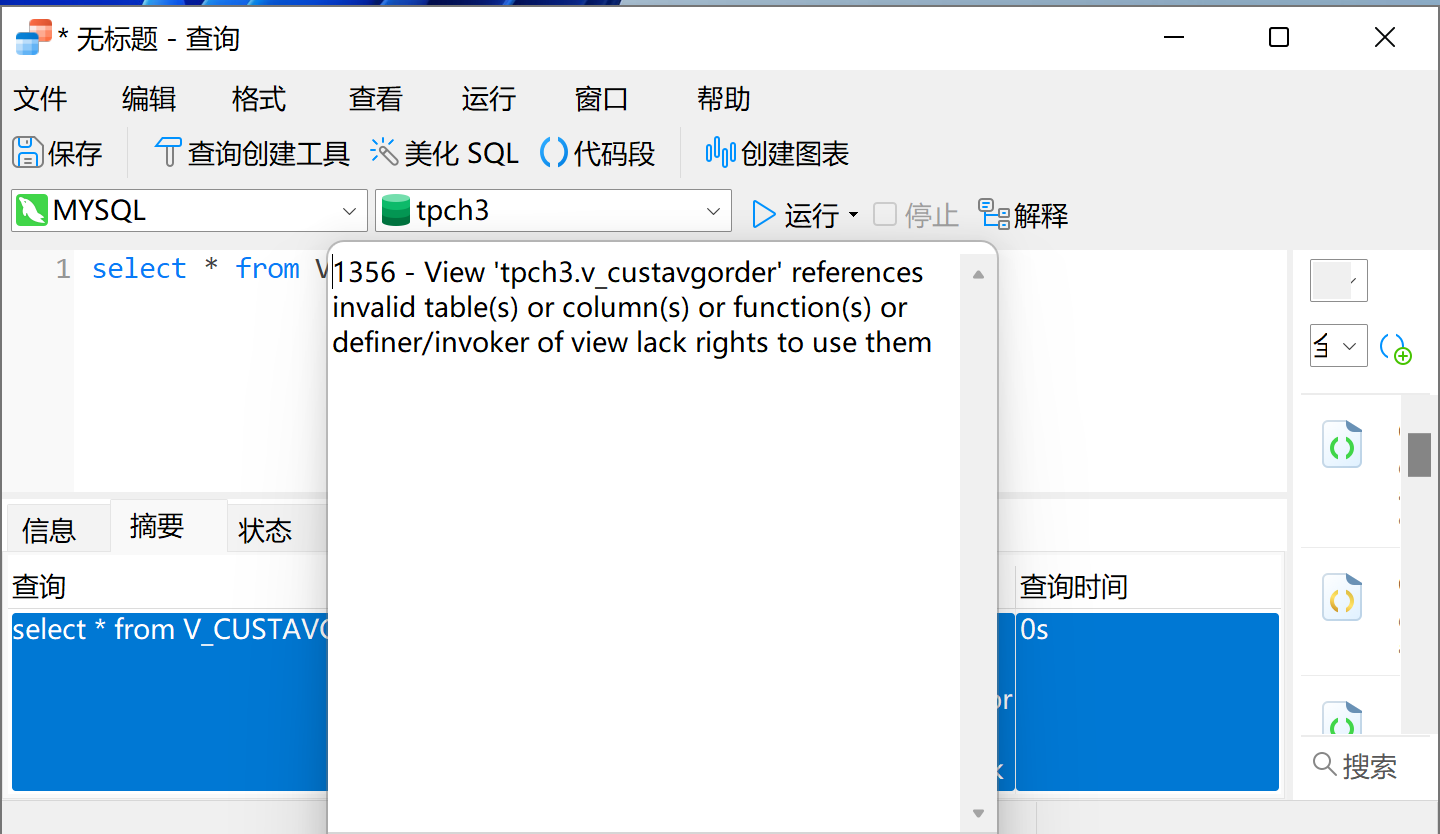
视图删除后视图的定义将从数据字典中删除，如果在该视图上还导出了其他视图，删除这些视图需要级联删除，遗憾的是，SQL Server2008并不支持级联删除，不过可以先删除视图 V\_CustAvgOrder，然后再删除视图V\_CustOrd。

4、使用 cascade 选项删除视图 V\_CustOrd

先做“1、创建顾客订购零件明细视图 V\_CustOrd”

drop view V\_CUSTORD cascade;

select \* from V\_CUSTAVGORDER1;



解释同restrict

* + - 1. **与实验结果相关的文件**

无

* + - 1. **实验总结**