

期末机试

学号	姓名
20337025	崔璨明

注：以下实验均在school数据库上完成。

1、创建表CARD(cid,sid,money),其中cid为主键,具有唯一性约束, sid为外键, 参照STUDENTS表的sid字段。数据类型: cid为char(10),sid 为char(10),money为decimal(10,2)。

```
CREATE TABLE CARD (  
    cid CHAR(10) PRIMARY KEY,  
    sid CHAR(10) REFERENCES STUDENTS(sid),  
    money DECIMAL(10,2)  
);
```

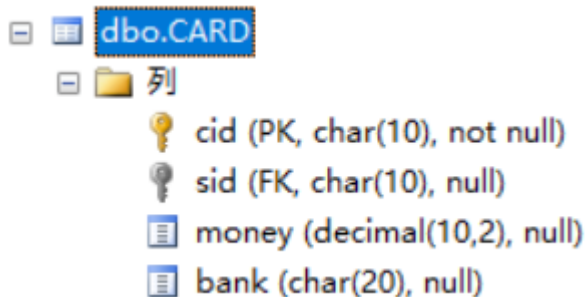
实验结果:



2、给表CARD增加一个属性列: bank, 数据类型为char(20)。

```
ALTER TABLE CARD ADD bank CHAR(20);
```

实验结果:



3、删除表CARD的属性bank。

```
ALTER TABLE CARD DROP COLUMN bank;
```

实验结果：



4、查询工资最低的教师姓名和开设的课程名称。

```
select tname,cname
from TEACHERS,CHOICES,COURSES
where salary=(select MIN(salary) from TEACHERS) and TEACHERS.tid=CHOICES.tid
and CHOICES.cid=COURSES.cid
```

实验结果：

	tname	cname
1	ossnnn	database
2	upxkrs	database
3	dfmpu	database
4	cpqsvt	database
5	elopzjnp	database
6	ueswxvfw	database
7	ooccbendq	database

5、查询课时与UML或C++的课时一样的课程名称。

```
select cname from COURSES
where COURSES.hour in(select hour from COURSES where
cname='UML' or cname='c++')
```

实验结果：

	cname
1	c++
2	uml
3	data structure
4	computer network
5	asp
6	struts
7	c#

✓ 查 | (local) (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:00 | 7 行

6、查询选修课程database但没有选修C++的学生编号（利用集合减运算）

```
select sname from STUDENTS
where sid in
(select sid from CHOICES
where cid =(select cid from COURSES where cname='database'))
except
select sname from STUDENTS
where sid in
(select sid from CHOICES
where cid =(select cid from COURSES where cname='C++'))
```

实验结果：

	sname
1	aaazoc
2	aagnlx
3	accuxh
4	acosvcqa
5	acqki
6	adlmyn
7	adnyg

✓ 查询已... | (local) (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:00 | 2918 行

7、查询选修database课程的成绩比名为“ruvldjlm ”的学生高的所有学生的编号。


```

select sid from CHOICES where
cid =(select cid from COURSES where cname='database')
and score>(
select score from CHOICES
where sid=(select sid from STUDENTS where sname='ruvldjlm')
and cid=(select cid from COURSES where cname='database')
)

```

实验结果：

结果		消息	
	sid		
1	842457951		
2	840045592		
3	822750091		
4	804312075		
5	857694294		
6	871441578		
7	883235106		

 查询已... | (local) (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:00 | 1051 行

8、查询database课程的平均成绩和最高成绩。


```

select AVG(score) as avg ,MAX(score) as max
from CHOICES
where cid =(select cid from COURSES where cname='database')

```

实验结果：

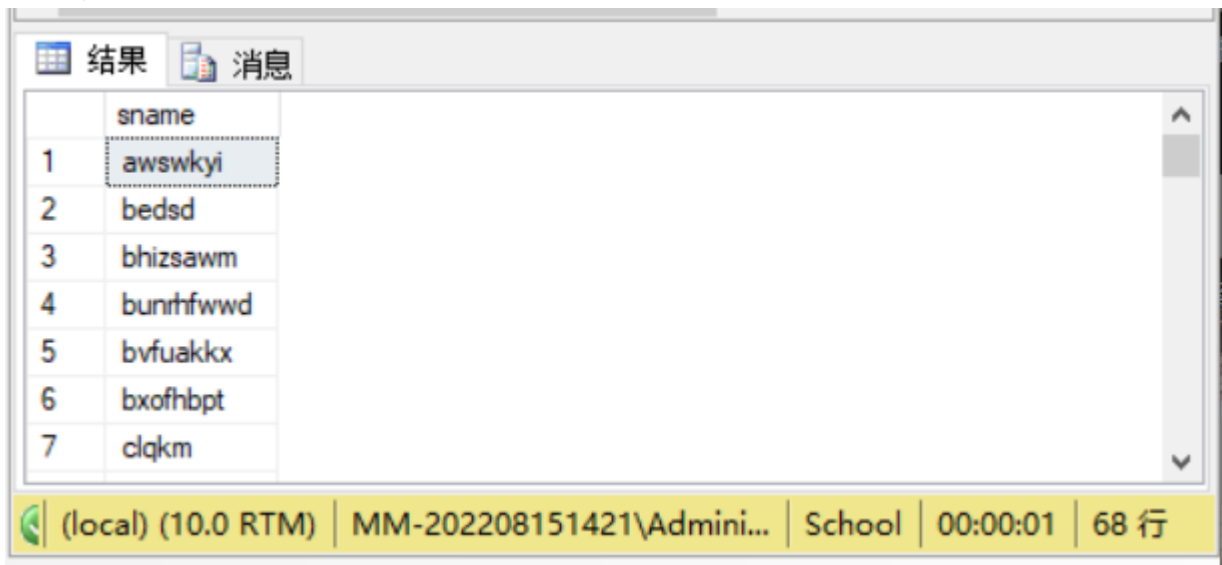
结果		消息	
	avg	max	
1	79	99	

 查询已成功... | (local) (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:00 | 1 行

9、查询选修course成绩最高的学生姓名。

```
select distinct sname
from students,CHOICES,COURSES
where CHOICES.cid=COURSES.cid and CHOICES.sid=STUDENTS.sid
      and COURSES.cname='database'
      and score=(select MAX(score)
                  from CHOICES, COURSES
                  where CHOICES.cid=COURSES.cid and
COURSES.cname='database')
```

实验结果：



The screenshot shows a SQL query result window with two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with two columns: 'sname' and an implicit index column. The table contains seven rows of student names. The status bar at the bottom indicates the connection is '(local) (10.0 RTM)', the user is 'MM-202208151421\Admini...', the database is 'School', the execution time is '00:00:01', and there are '68 行' (68 rows) in total.

	sname
1	awswkyi
2	bedsd
3	bhizawm
4	bunrhfwwd
5	bvfuaakkx
6	bxofhbpt
7	clqkm

(local) (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:01 | 68 行

10、查询没有选修course课程的学生姓名。

```
select sname from STUDENTS
where sid not in (select sid
from CHOICES where cid=
(select cid from courses where cname= 'database'))
```

结果		消息
	sname	
4	ogvmu	
5	uxqqbkjn	
6	zapyv	
7	zyuoh	
8	uwphrw	
9	aoaahudi	
10	fnvxgrisg	

查询已成功执行。 (local) (10.0 RTM) MM-202208151421\Admini... School 00:00:01 56849 行

11、查询选修了两门以上课程的学生姓名。

```
select sname,count(choices.cid)
from COURSES,CHOICES,STUDENTS
where CHOICES.cid=COURSES.cid and STUDENTS.sid=CHOICES.sid
group by sname
having COUNT(choices.cid)>=2
```

实验结果：

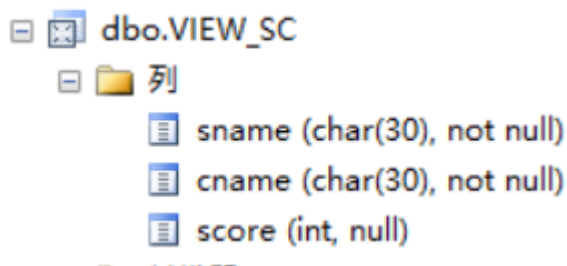
结果		消息
	sname	(无列名)
1	gvsybyb	3
2	kgtjt	2
3	kwzmpy	4
4	sktfrpi	4
5	ilalscn	3
6	dxwqw	4
7	orzqwkxn	2

(local) (10.0 RTM) MM-202208151421\Admini... School 00:00:00 44565 行

12、建立视图VIEW_SC，这个视图由学生姓名以及其选修的课程名称和相应分数构成。

```
CREATE VIEW VIEW_SC AS
SELECT S.sname AS sname, CO.cname, C.score
FROM (STUDENTS S
JOIN CHOICES C ON S.sid = C.sid)join COURSES CO on C.cid=CO.cid
```

实验结果：



13、利用视图VIEW_SC，查询分数大于90分数的学生姓名。

```
select cname from VIEW_SC
where score>90
```

实验结果：

	sname
1	fiiluomh
2	fiiluomh
3	uxqqbkjn
4	zapyv
5	hvyfafcj
6	ucsqywg
7	cxjnj

[local] (10.0 RTM) | MM-202208151421\Admini... | School | 00:00:00 | 32194 行

14、是否能利用视图VIEW_SC插入数据？为什么？（问答题）

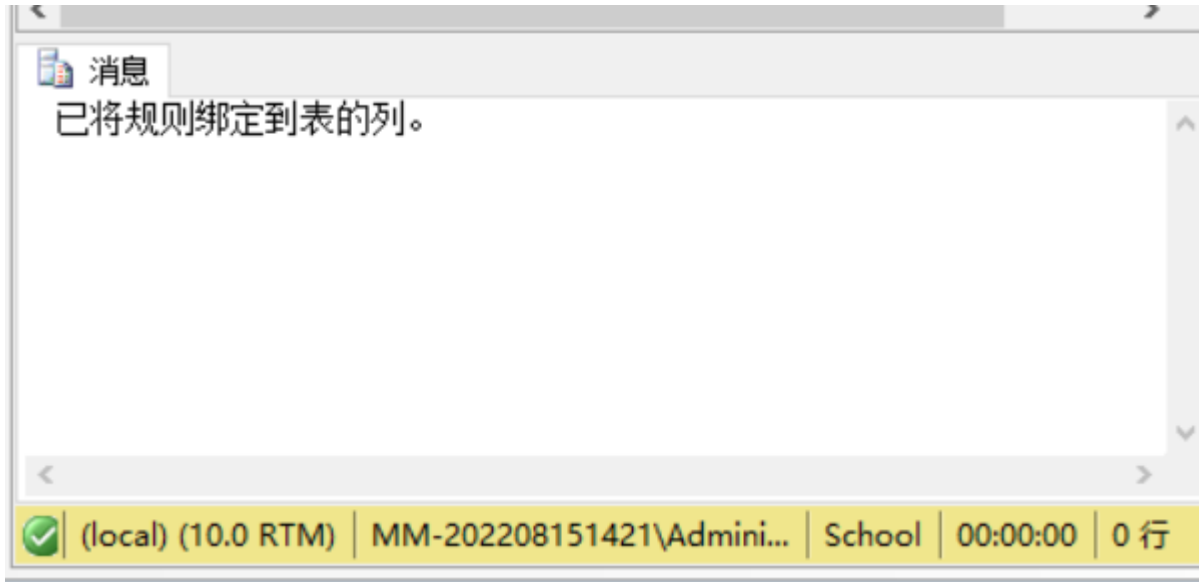
答：不能使用视图插入数据。因为视图是一种虚拟表，它通过将查询的结果存储在数据库中来提供数据，但不能用于存储数据。

视图的目的是提供一种方便的方式来查询数据库中的数据，而不是用于存储数据。因此，使用视图时，可以使用 `SELECT` 语句来查询数据，但不能使用 `INSERT`、`UPDATE` 或 `DELETE` 语句来插入、更新或删除数据。

15、创建规则R1，确保插入的money值大于0，并将规则R1绑定到表CARD的money属性上。

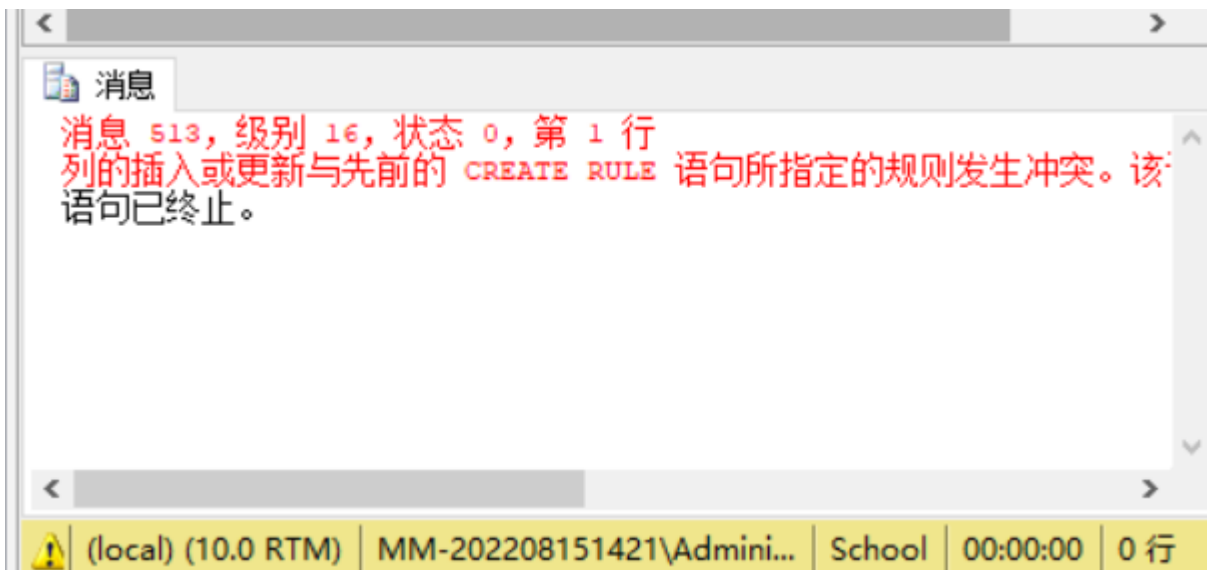
```
go
create rule R1 as @value>0
go
exec sp_bindrule R1,'CARD.[MONEY]';
```

实验结果：



16、在表CARD中插入一条违反规则R1的记录。

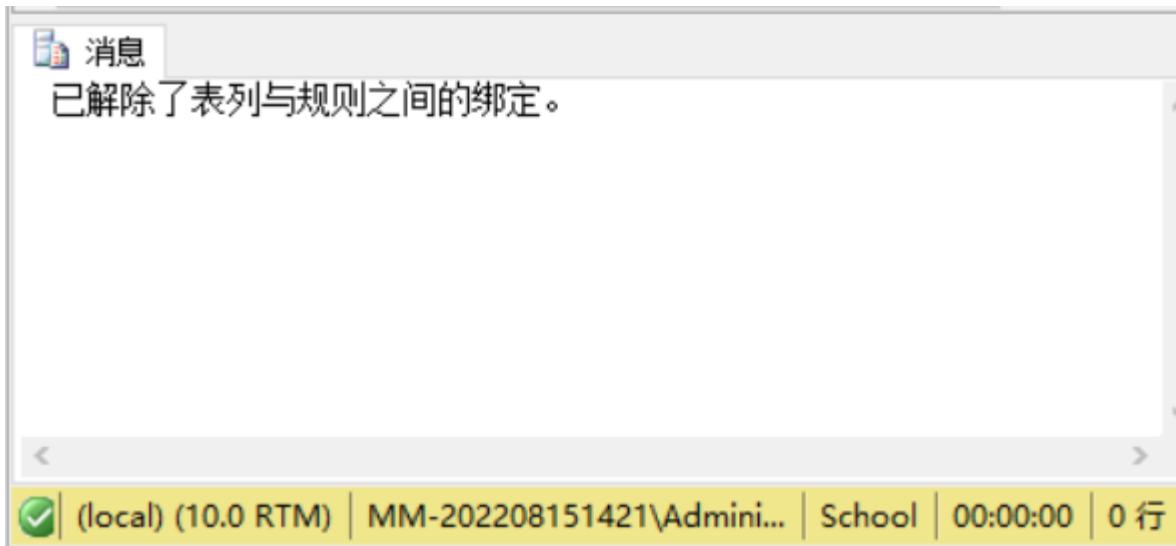
```
insert CARD values('10000000','60000000',-1)
```



17、解绑表CARD上规则R1的绑定。


```
exec sp_unbindrule 'CARD.[MONEY]'
```

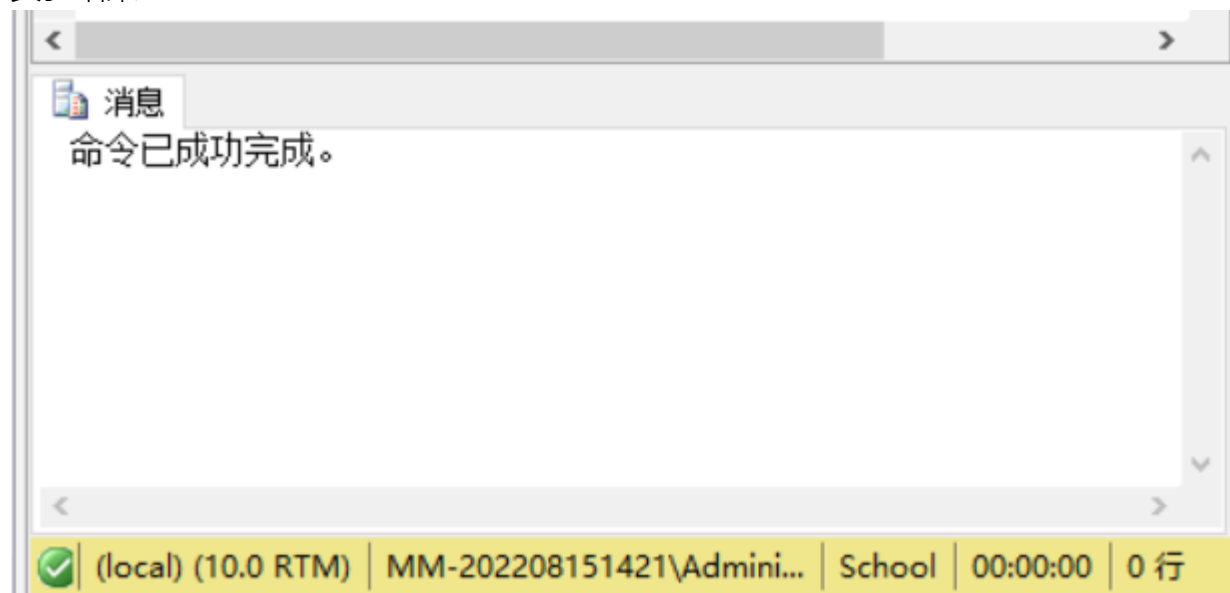
实验结果：



18、为表STUDENTS建立触发器T1，禁止删除学号为800015960的记录。

```
CREATE TRIGGER T1
ON STUDENTS
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT * FROM deleted WHERE sid = 800015960)
    BEGIN
        RAISERROR('Cannot delete student with ID 800015960', 16, 1)
        ROLLBACK TRANSACTION
    END
END;
```

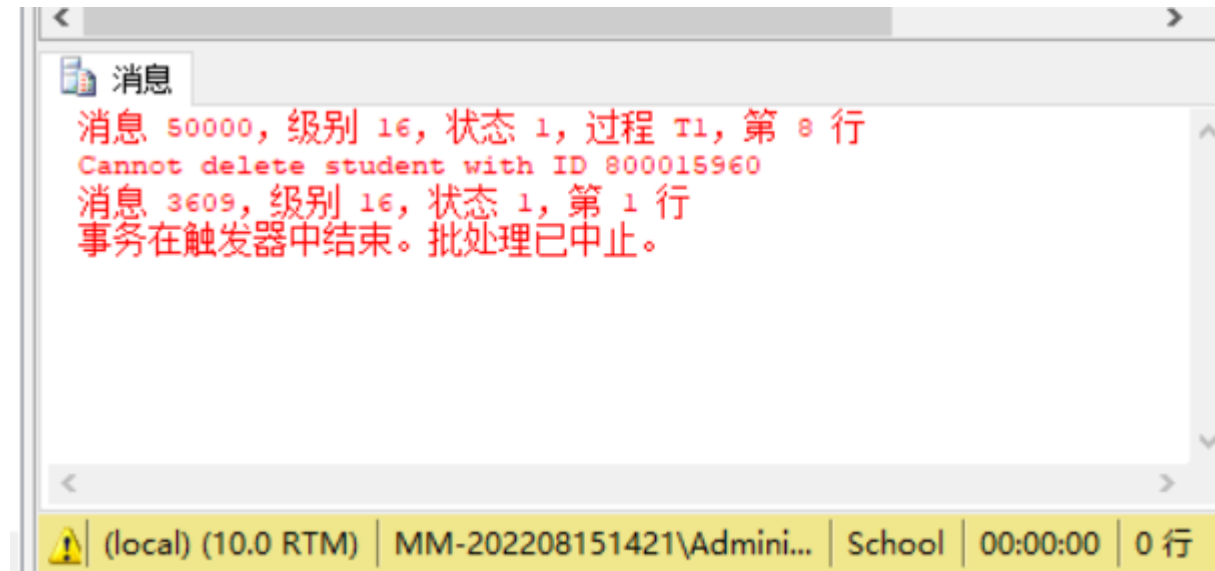
实验结果：



19、演示违反触发器T1的操作。

```
DELETE FROM STUDENTS WHERE sid = 800015960;
```

实验结果：



20、编写一个嵌套事务。外层修改STUDENTS表某记录，内层在COURSES表插入一条记录。演示内层插入操作失败后，外层修改操作回滚。


```
select 'before transaction' as hint, @@TRANCOUNT AS TRANSACTIONCOUNT
BEGIN TRAN
SELECT 'THE FIRST TRANSACTION STARTS' AS HINT, @@TRANCOUNT AS TRANSACTIONCOUNT
```

```

UPDATE STUDENTS SET SNAME='MIKE' WHERE SID='60000000 '
BEGIN TRAN
    INSERT COURSES VALUES('1000','1',100)
    SELECT 'THE SECOND TRANSACTION STARTS:' AS HINT, @@TRANCOUNT AS
TRANSACTIONCOUNT
    COMMIT TRAN
    SELECT 'THE SECOND TRANSACTION COMMITS' AS HINT, @@TRANCOUNT AS
TRANSACTIONCOUNT
ROLLBACK TRAN
    SELECT 'THE FIRST TRANSACTION STARTS' AS HINT, @@TRANCOUNT AS
TRANSACTIONCOUNT

```

实验结果：

结果	消息				
1	<table> <tr> <th>hint</th><th>TRANSACTIONCOUNT</th></tr> <tr> <td>before transaction</td><td>0</td></tr> </table>	hint	TRANSACTIONCOUNT	before transaction	0
hint	TRANSACTIONCOUNT				
before transaction	0				
1	<table> <tr> <th>HINT</th><th>TRANSACTIONCOUNT</th></tr> <tr> <td>THE FIRST TRANSACTION STARTS</td><td>1</td></tr> </table>	HINT	TRANSACTIONCOUNT	THE FIRST TRANSACTION STARTS	1
HINT	TRANSACTIONCOUNT				
THE FIRST TRANSACTION STARTS	1				
1	<table> <tr> <th>HINT</th><th>TRANSACTIONCOUNT</th></tr> <tr> <td>THE SECOND TRANSACTION STARTS:</td><td>2</td></tr> </table>	HINT	TRANSACTIONCOUNT	THE SECOND TRANSACTION STARTS:	2
HINT	TRANSACTIONCOUNT				
THE SECOND TRANSACTION STARTS:	2				
1	<table> <tr> <th>HINT</th><th>TRANSACTIONCOUNT</th></tr> <tr> <td>THE SECOND TRANSACTION COMMITS</td><td>1</td></tr> </table>	HINT	TRANSACTIONCOUNT	THE SECOND TRANSACTION COMMITS	1
HINT	TRANSACTIONCOUNT				
THE SECOND TRANSACTION COMMITS	1				
1	<table> <tr> <th>HINT</th><th>TRANSACTIONCOUNT</th></tr> <tr> <td>THE FIRST TRANSACTION STARTS</td><td>0</td></tr> </table>	HINT	TRANSACTIONCOUNT	THE FIRST TRANSACTION STARTS	0
HINT	TRANSACTIONCOUNT				
THE FIRST TRANSACTION STARTS	0				
 查询已成功执行。					