

数据库实验lab5

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

2017/10/20

Content

数据库实验lab5

Content

实验目的

句法汇总

实验内容

创建基于多个基表的视图

test1

test2

创建行列子集视图

test3

前三次实验的实验结果

基于视图的查询

test4

test5

test6

test7

test7实验结果分析

基于视图的更新

test8

视图的删除

test9

实验结果附录

test4的实验结果

test5的实验结果

test8的实验结果

实验目的

1. 熟悉SQL有关视图的操作
2. 能够使用SQL创建需要的视图，对视图进行查询以及删除操作

句法汇总

1. 创建视图的方式——test1
2. with check option的用法——test3
3. 视图的删除——test9

实验内容

创建基于多个基表的视图

test1

定义选课信息和课程名称的视图 VIEWC

```
create view viewc(no, sid, tid, cid, score, cname)
as select cis.no, cis.sid, cis.tid, cis.cid, cis.score, crs.cname
from CHOICES as cis, COURSES as crs
where cis.cid = crs.cid
```

思路比较简单，就不做分析了。

句法之创建视图的方式：

```
create view 视图名字（视图元素）
as 子查询语句 查询元素--元素之间对应
```

test2

定义学生姓名与选课信息的视图 VIEWS

```
create view views
(no, sid, tid, cid, score, sname)
as select
CHOICES.no, choices.sid, choices.tid, choices.cid, choices.score, students.sname
from CHOICES, STUDENTS
where choices.sid = STUDENTS.sid
```

考察方式同上一题，理解简单，句法同上。

创建行列子集视图

test3

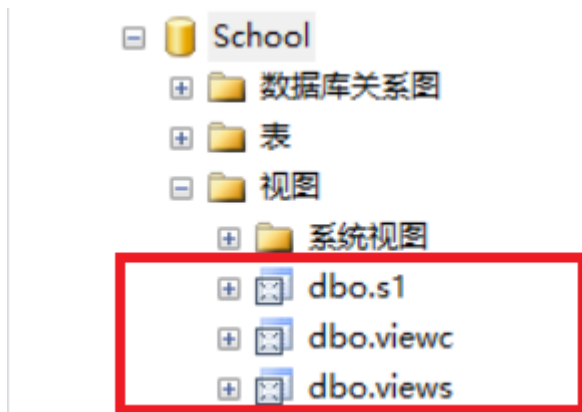
定义年级低于1998的学生的视图S1(SID, SNAME, GRADE)

```
create view s1
(sid, sname, grade)
as select sid, sname, grade
from STUDENTS
where grade > 1998
with check option
```

分析题意：基于基本table的视图创建，将特有的属性在select语句中查询出来，与视图的属性相互对照即可。

句法：with check option 设置约束条件。check条件会对之后的插入信息或者删除信息进行检查，对不符合条件的元组，禁止修改信息的操作。

前三次实验的实验结果



可以看到，在School库中的视图区域，增加了3个视图信息——正是笔者之前命名的3个视图。

基于视图的查询

test4

查询学生为“uxjof”的学生的选课信息

```
select *
from views
where sname = 'uxjof'
```

基于视图的查询与基于table的查询一样，句法也别无二致。

test5

查询选修课程“UML”的学生的编号和成绩

```
select *
from viewc
where cname = 'uml'
```

test6

向视图S1插入记录(60000001,Lily,2001)

```
insert
into s1
values(60000001, 'Lily', 2001)
```

由于插入的年份信息“2001”大于“1998”，所以可以插入。

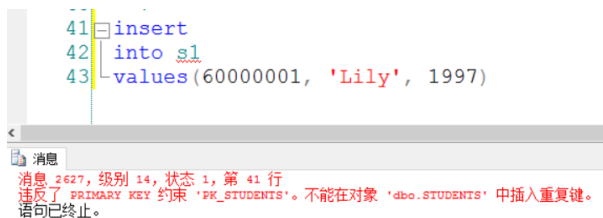
test7

定义包括更新和插入约束的视图S1,尝试向视图插入记录(60000001,Lily,1997)

```
insert
into s1
values(60000001, 'Lily', 1997)
```

test7实验结果分析

对插入操作的反馈结果如下：

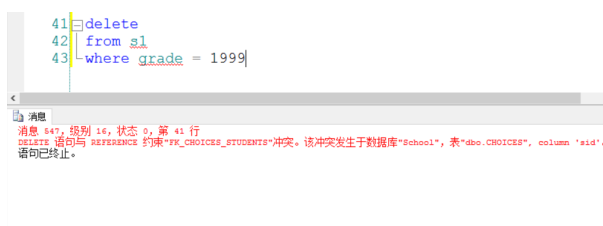


反馈结果显示，主键有重复信息——因为我们之前插入过一个学号为60000001的信息，grade为2001，所以本次插入以失败告终。

删除所有年级为1999的学生记录,讨论更新和插入约束带来的影响

```
delete
from s1
where grade = 1999
```

反馈结果如下：



实验结果分析：由于S1是students的行列子集视图，对于s1的操作将会影响到students表格的操作——而students的sid属性是作为choices的外键存在的，这种连带关系，使得删除操作不能正常执行。

基于视图的更新

test8

在视图 VIEWS中将姓名为“uxjof”的学生的选课成绩都加上5分

```
update views
set score = score + 5
where sname = 'uxjof'
```

基于视图的更新与基于table的更新一样。

视图的删除

test9

取消以上建立的所有视图

```
drop view s1
drop view viewc
drop view views
```

句法：drop view删除视图。后接视图名称即可，如果B视图基于A视图，而A被删除——此时B视图也会同时失效——需要手动显式删除。

实验结果附录

test4的实验结果

结果		消息				
	no	sid	tid	cid	score	sname
1	506978093	800023963	220667042	10046	94	uxjof
2	541221076	800023963	238341990	10018	84	uxjof
3	567316431	800023963	258375444	10037	98	uxjof

test5的实验结果

结果		消息				
	no	sid	tid	cid	score	sname
1	506978093	800023963	220667042	10046	94	uxjof
2	541221076	800023963	238341990	10018	84	uxjof
3	567316431	800023963	258375444	10037	98	uxjof

test8的实验结果

更新之前：

结果 消息						
	no	sid	tid	cid	score	sname
1	541221076	800023963	238341990	10018	84	uxjof
2	506978093	800023963	220667042	10046	94	uxjof
3	567316431	800023963	258375444	10037	98	uxjof

更新之后:

结果 消息						
	no	sid	tid	cid	score	sname
1	506978093	800023963	220667042	10046	99	uxjof
2	541221076	800023963	238341990	10018	89	uxjof
3	567316431	800023963	258375444	10037	103	uxjof