

# 中山大学数据科学与计算机学院本科生实验报告

(2020学年秋季学期)

课程名称：数据库系统 任课教师：刘玉葆

年级&班级	18级 计科一班	专业(方向)	计科 (大数据与人工智能方向)
学号	18340014	姓名	陈嘉宁
电话	18475934419	Email	<a href="mailto:734311072@qq.com">734311072@qq.com</a>
开始日期	2020/10/23	完成日期	2020/10/24

## 一、实验题目

SQL的视图操作

## 二、实验目的

1. 熟悉SQL的有关视图的操作，能够使用SQL语句创建需要的视图，对视图进行查询和取消视图。

## 三、实验内容

1. 定义常见的视图形式，包括：
  - 行列子集视图
  - WITH CHECK OPTION的视图
  - 基于多个基表的视图
  - 基于视图的视图
  - 带表达式的视图
  - 分组视图
2. 考察 WITH CHECK OPTION这一语句在视图定义后产生的影响，包括对修改操作、删除操作、插入操作的影响
3. 讨论视图的数据更新情况，对子行列视图进行数据更新。
4. 使用DROP语句删除一个视图，由该视图导出的其他视图定义仍在数据字典中，但已不能使用，必须显式删除。同样的原因，删除基表时，由该基表导出的所有视图定义都必须显式删除。

## 四、实验过程

### 1. 定义选课信息和课程名称的视图 VIEWC

本题需要对 CHOICES 表和 COURSES 表进行自然连接，即  $CHOICES.cid = COURSES.cid$ 。定义视图所用的SQL语句为：

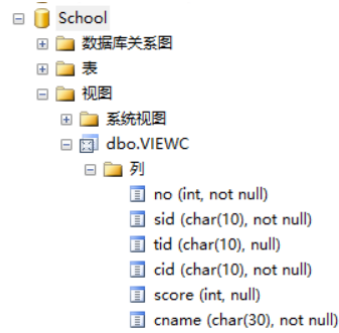
```
1 create view VIEWC
2 as select no,sid,tid,CHOICES.cid,score,cname
3 from CHOICES,COURSES
4 where CHOICES.cid = COURSES.cid
```

执行结果为：

```
SQLQuery1.sql - (I:\KLS675B\dell (58))*
create view VIEWC
as select no,sid,tid,CHOICES.cid,score,cname
from CHOICES,COURSES
where CHOICES.cid = COURSES.cid
```

消息  
命令已成功完成。

在该语句执行后，可以在 SCHOOL 数据库的“视图”项目下找到新建的视图 VIEWC：



## 2. 定义学生姓名与选课信息的视图 VIEWS

本题需要对 CHOICES 表和 SYUDENTS 表进行自然连接，即 CHOICES.sid = STUDENTS.sid。定义视图所用的SQL语句为：

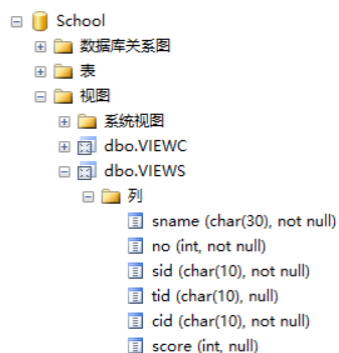
```
1 create view VIEWS
2 as select sname,no,STUDENTS.sid,tid,cid,score
3 from CHOICES,STUDENTS
4 where CHOICES.sid = STUDENTS.sid
```

执行结果为：

```
SQLQuery3.sql - (I:\KLS675B\dell (59)) SQLQuery1.sql - (I:\KLS675B\dell (58))*
create view VIEWS
as select sname,no,STUDENTS.sid,tid,cid,score
from CHOICES,STUDENTS
where CHOICES.sid = STUDENTS.sid
```

消息  
命令已成功完成。

在该语句执行后，可以在 SCHOOL 数据库的“视图”项目下找到新建的视图 VIEWS：



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

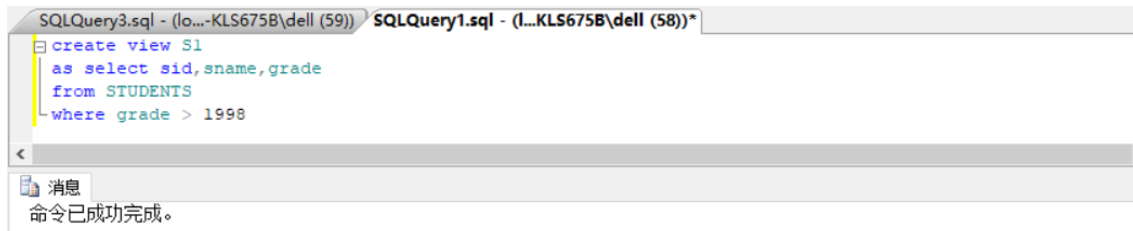
定义视图所用的SQL语句为：

```

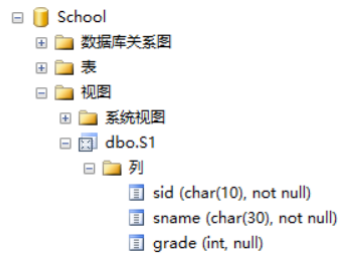
1 create view S1
2 as select sid,sname,grade
3 from STUDENTS
4 where grade > 1998

```

执行结果为：



在该语句执行后，可以在 SCHOOL 数据库的“视图”项目下找到新建的视图 S1：



#### 4. 查询学生为“uxjof”的学生的选课信息

查询语句为：

```

1 select *
2 from VIEWS
3 where sname = 'uxjof'

```

执行结果为：

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

#### 5. 查询选修课程“UML”的学生的编号和成绩：

查询语句为：

```

1 select sid,score
2 from VIEWC
3 where cname = 'UML'

```

执行结果为：

SQLQuery3.sql - (lo...-KLS675B\ dell (59))SQLQuery1.sql - (l...KLS675B\ dell (58))\*

```
select sid, score
from VIEWC
where cname = 'UML'
```

结果消息

	sid	score
1	848035070	88
2	897664264	62
3	898453203	91
4	846108663	86
5	827984677	86
6	823352185	NULL
7	882778410	69
8	884993242	74
9	806427512	96
10	848803637	85
11	839666024	92
12	844745062	96
13	818493215	88
14	815954009	62

由于查询结果数量过多，此处仅截取部分结果作为示意。

## 6. 向视图S1插入记录(60000001,Lily,2001):

插入语句为：

```

1 insert into s1
2 values('60000001','Lily',2001)

```

执行结果为：

SQLQuery3.sql - (lo...-KLS675B\ dell (59))		SQLQuery1.sql - (l...KLS675B\ dell (58))*	
<pre>insert into s1 values('60000001','Lily',2001)</pre>			
<div>消息</div>			
(1 行受影响)			

在执行插入语句后，可以在 SCHOOL 数据库中搜索到所插入的记录：

SQLQuery3.sql - (lo...-KLS675B\dell (59))SQLQuery1.sql - (l...KLS675B\dell (58))\*

```
select *  
from s1  
where sid = '60000001'
```

<

说明插入成功。

## 7. 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),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:

尝试插入记录(60000001,Lily,1997):

```

1 insert into s1
2 values('60000001','Lily',1997)

```

执行结果：

```
SQLQuery3.sql - (lo...-KLS675B\deli (59)) SQLQuery1.sql - (l...KLS675B\deli (58))*
insert into s1
values ('60000001', 'Lily', 1997)
```

消息  
消息 2627, 级别 14, 状态 1, 第 1 行  
违反了 PRIMARY KEY 约束 'PK\_STUDENTS'。不能在对象 'dbo.STUDENTS' 中插入重复键。  
语句已终止。

尝试删除所有年级为1999的学生记录：

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

执行结果为：

```
SQLQuery3.sql - (lo...-KLS675B\deli (59)) SQLQuery1.sql - (l...KLS675B\deli (58))*
delete from s1
where grade = 1999
```

消息  
消息 547, 级别 16, 状态 0, 第 1 行  
DELETE 语句与 REFERENCE 约束 'FK\_CHOICES\_STUDENTS' 冲突。该冲突发生于数据库 "School", 表 "dbo.CHICES", column 'sid'。  
语句已终止。

插入约束可以有效防止用户插入重复的键导致混乱；删除约束可以保证视图和数据库中数据的一致性。

## 8. 在视图 VIEWS中将姓名为“uxjof”的学生的选课成绩都加上5分：

在更新前，通过题目4可以得知姓名为“uxjof”的学生的选课情况如下：

```
SQLQuery3.sql - (lo...-KLS675B\deli (59)) SQLQuery1.sql - (l...KLS675B\deli (58))*
select *
from VIEWS
where sname = 'uxjof'
```

结果 消息

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

更新语句为：

```
1 update VIEWS
2 set score = score + 5
3 where sname = 'uxjof'
```

执行结果为：

```
SQLQuery3.sql - (lo...-KLS675B\deli (59)) SQLQuery1.sql - (l...KLS675B\deli (58))*
update VIEWS
set score = score + 5
where sname = 'uxjof'
```

消息  
(3 行受影响)

更新后，姓名为“uxjof”的学生的选课情况如下：

SQLQuery3.sql - (lo...-KLS675B\dell (59)) SQLQuery1.sql - (l...KLS675B\dell (58))\*

```

select *
from VIEWS
where sname = 'uxjof'

```

结果 消息

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

## 9. 取消以上建立的所有视图：

取消视图的语句为：

```

1 drop view VIEWC
2 drop view VIEWS
3 drop view S1

```

执行结果为：

SQLQuery3.sql - (lo...-KLS675B\dell (59)) SQLQuery1.sql - (l...KLS675B\dell (58))\*

```

drop view VIEWC
drop view VIEWS
drop view S1

```

消息

命令已成功完成。

取消上述视图后在 SCHOOL 数据库下不再有自定义视图：

- [-] School
  - [+] 数据库关系图
  - [+] 表
  - [+] 视图
  - [+] 系统视图
  - [+] 同义词

取消视图完成。