

# 数据库实验报告 实验七 视图

姓名	学号	班级	课室
熊明	20305055	计科5班	D503

## 一、实验目的

熟悉SQL语言支持的有关视图的操作，能够熟练使用SQL语句来创建需要的视图，对视图进行查询和取消视图。

## 二、实验环境

数据库：Mysql

图形化工具：Navicat Premium 16

## 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. 通过实验考察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。

注：以下是PostgreSQL包含check-option可更新视图简介，更多内容可参考[PostgreSQL官网](#)

```
1  一、创建可更新视图：
2  create or replace view usa_city as
3      select city_id,city,country_id
4          from city
5          where country_id=103 order by city;
6
7  select * from usa_city;
8
9  更新视图插入数据：
10 insert into usa_city (city,country_id )values ('Birmingham', 102);
11 结果：插入成功，但是此插入的结果并不在视图usa_city中
12 插入的新行在视图中不可见。这可能会造成安全问题，为防止用户插入或更新通过视图不可见的行，在
   创建视图时可使用
13 WITH CHECK OPTION 子句。
14
```

```
15
16
17 二、创建有检查项的可更新视图
18 create or replace view usa_city as
19     select city_id,city,country_id
20         from city
21     where country_id=103 order by city
22     with check option;
23
24
25 更新视图插入数据:
26 insert into usa_city (city,country_id )values ('Birmingham', 102);
27 结果: > 错误:  新行违反了视图"usa_city"的检查选项
28
29 insert into usa_city (city,country_id )values ('Birenc', 103);
30 结果: 插入成功, 且只允许插入的数据满足原视图的where条件
31
32
33
34 三、检查项含local的可更新视图
35 (1)、创建一个可更新的基表视图
36 create or replace view usa_a as
37     select city_id,city,country_id
38         from city
39     where city like 'A%';
40
41 (2)、创建检查项含local的可更新视图
42 create or replace view usa_a_city as
43     select city_id,city,country_id
44         from usa_a
45     where country_id=103 order by city
46     with local check option;
47
48
49 更新视图插入数据:
50 insert into usa_a_city(city,country_id)values('Mirmin', 103);
51 结果: 插入成功, 因为usa_a_city视图只需要检查自身的插入数据是否满足where条件即可
52
53 insert into usa_a_city(city,country_id)values('Mirmin', 102);
54 结果: 插入失败, > 错误:  新行违反了视图"usa_a_city"的检查选项
55
56
57
58 (3)、创建检查项含cascaded的可更新视图
59 create or replace view usa_a_city as
60     select city_id,city,country_id
61         from usa_a
62     where country_id=103 order by city
63     with cascaded check option;
64
65 更新视图插入数据:
66 insert into usa_a_city(city,country_id)values('Meery', 103);
67 结果: 插入失败, > 错误:  新行违反了视图"usa_a"的检查选项
68 因为usa_a_city视图使用了cascaded级联检查, 即本身的where条件要满足同时也要满足基表视图的
   where条件
```

69

70

71 insert into usa\_a\_city(city,country\_id)values('Aeery', 103);

72 结果：插入成功，同时满足了本身视图的where条件，也满足了基表视图usa\_a的where条件

## 四、课内实验

1. 创建一个行列子集视图，给出选课成绩合格的学生的编号，所选课程号和该课程成绩

代码如下：

```
1 CREATE OR REPLACE VIEW t1(no,sid,cid,score) AS
2     SELECT no,sid,cid,score
3     FROM choices
4     WHERE score>=60
5     with check option
```

视图如下：

lab_database	NO	sid	cid	score
mysql	500000058	823069829	10037	76
performance_schema	500000253	829348273	10010	87
school	500001270	847061074	10025	92
表	500002143	860635914	10039	82
视图	500002371	829785562	10028	77
t1	500002715	822137137	10011	67
t2	500003262	826310502	10005	90
函数	500004144	817636568	10047	60
查询	500005935	801967882	10021	70
备份	500005987	875434315	10048	82
sys	500006353	830180555	10016	76
	500006805	848035070	10007	88
	500006824	834091581	10049	72
	500007004	809548802	10002	64
	500007637	833961570	10004	80
	500007901	894256303	10018	75
	500008311	826412145	10009	63
	500008620	836887807	10031	74
	500009030	804529880	10023	80
	500009612	835119325	10009	64
	500010028	836646619	10043	83
	500010629	867062490	10003	96
	500010697	897664264	10007	62
	500010915	894037661	10010	79
	500011058	838043180	10035	86
	500012602	836584689	10045	83
	500012952	835265794	10043	68
	500013110	878303907	10026	80
	500013643	856242358	10029	61
	500014353	820248718	10015	69

2. 创建基于多个基表的视图，这个视图由学生姓名和其所选修的课程名及讲授该课程的教师姓名构成

代码如下：

```

1 CREATE OR REPLACE VIEW t2 AS
2     SELECT sname,cname,tname
3     FROM choices
4     LEFT JOIN students
5     ON students.sid = choices.sid
6     LEFT JOIN courses
7     ON courses.cid = choices.cid
8     LEFT JOIN teachers
9     ON teachers.tid = choices.tid

```

使用左连接，完整显示choices。结果如下：

information_schema	开始事务	文本	筛选	排序	导出	创建图表
lab_database	sname	cname	tname			
mysql	pxfys	software testing	upnhtksjg			
performance_schema	rflsleav	software engineering	pogyevqrj			
school	baqzmo	j2me	gkkogl			
表	qxkbh	embedded system	nbmma			
视图	xnhdjo	fortran	lgqcxr			
t1	bemgynei	architectonics	ihztiyd			
t2	qaxwe	distributed computing	vjkypqij			
函数	cqkrjkuf	c++	ijmrglzf			
查询	wzinemrs	computer interface	hkrbyvn			
备份	rcypjhsnc	corba	zfkzswqc			
sys	kvamveu	j2me	djeppj			
	rekmgdbo	data warehouse	jacyfyxg			
	mbfrhwz	data mining	fnuxmugx			
	kxqlbun	uml	iovegolq			
	xxamhe	project management	qgubszad			
	myosjmd	operating system	qxahwfc			
	bzbogqep	java	xniyxcpee			
	mamqnc	unix/linux	pclusw			
	ouvxyha	cryptology	utydr			
	mriky	internet	bucdn			
	sdqaqj	corba	gzjsiljd			
	brahrb	cryptology	qonhyx			
	ofsjtj	information security	zszverct			
	pcyacxqr	computer graphics	qompmmm			
	rvgbhmwaw	uml	svsquiuy			
	qeiygk	software engineering	qcgpu			
	ligga	computer virus	bnvhsjoem			
	nrxfdwz	compiling principle	bkmgg			
	whbjk	software interface	rdjgcmj			
	information security	information security	information security			

3. 创建带表达式的视图，由学生姓名、所选课程名和所有课程成绩都比原来多5分这几个属性组成代码如下：

```

1 CREATE OR REPLACE VIEW t3(sname,cname,socre) AS
2     SELECT sname,cname,score+5 AS score
3     FROM choices
4     LEFT JOIN students
5     ON students.sid = choices.sid
6     LEFT JOIN courses
7     ON courses.cid = choices.cid

```

结果如下：

lab_database		
mysql		
performance_schema		
school		
表		
视图		
t1		
t2		
t3		
函数		
查询		
备份		
sys		

  

sname	cname	socre
pxfys	software testing	81
rflsleav	software engineering	92
baqzmo	j2me	59
qxkbh	embeded system	97
xnhdjo	fortran	87
bemgynei	architectonics	82
qaxwe	distributed computing	72
cqkrjkuf	c++	95
wzinemrs	computer interface	65
rcypjhsnc	corba	(Null)
kvamveu	j2me	75
rekmgdbo	data warehouse	87
mbfrhwz	data mining	81
kxqlbun	uml	93
xxamhe	project management	77
myosjmd	operating system	69
bzbogqep	java	85
mamqnc	unix/linux	80
ouvxyha	cryptology	68
mriky	internet	79
sdqaqj	corba	85
brahrb	cryptology	69
ofsjtxj	information security	88
pcyacxqr	computer graphics	101
rvgbhmwaw	uml	67
qeiygk	software engineering	84
ligga	computer virus	91
nrxfdwz	compiling principle	(Null)

4. 创建分组视图，将学生的学号及其平均成绩定义为一个视图

代码如下：

```

1 CREATE OR REPLACE VIEW t4(sid,avg_socre) AS
2   SELECT sid,avg(any_value(score)) as avg_socre
3   FROM choices
4   GROUP BY sid

```

结果如下：

lab_database	开始事务	文本	筛选	排序
mysql	sid	avg_socre		
performance_schema	800001216	62.3333		
school	800002933	73.6667		
表	800005753	66.0000		
视图	800006682	89.0000		
t1	800006941	71.0000		
t2	800007595	74.0000		
t3	800008565	76.0000		
t4	800009026	88.5000		
函数	800009099	87.5000		
查询	800009249	75.5000		
备份	800010666	73.5000		
sys	800013889	84.5000		
	800014004	70.0000		
	800014678	71.0000		
	800014991	83.0000		
	800015244	(Null)		
	800015960	83.0000		
	800016416	75.6667		
	800016895	88.0000		
	800017442	75.0000		
	800017444	73.5000		
	800017669	70.0000		
	800017736	84.0000		
	800020409	69.0000		

5. 创建一个基于视图的视图，基于(1)中建立的视图，定义一个包括学生编号，学生所选课程数目和平均成绩的视图

代码如下：

```

1 CREATE OR REPLACE VIEW t5 AS
2     SELECT sid,count(cid) AS courses_count ,avg(score) AS avg_socre
3     FROM t1
4     GROUP BY sid

```

结果如下：

lab_database	
mysql	
performance_schema	
school	
表	
视图	
t1	
t2	
t3	
t4	
t5	
函数	
查询	
备份	
sys	

sid	courses_count	avg_socre
800001216	3	62.3333
800002933	3	73.6667
800005753	1	66.0000
800006682	3	89.0000
800006941	3	76.3333
800007595	3	74.0000
800008565	1	76.0000
800009026	2	88.5000
800009099	2	87.5000
800009249	2	75.5000
800010666	2	73.5000
800013889	4	84.5000
800014004	2	78.5000
800014678	1	71.0000
800014991	1	83.0000
800015960	1	83.0000
800016416	3	75.6667
800016895	3	88.0000
800017442	4	75.0000
800017444	3	80.3333
800017669	3	76.3333
800017736	4	84.0000
800020409	3	69.0000
800020890	3	76.0000
800022091	3	84.0000

6. 查询所有选修课程Software Engineering的学生姓名

```
1 SELECT sname
2 FROM choices
3 LEFT JOIN students
4 ON students.sid = choices.sid
5 WHERE cid = (
6 SELECT cid
7 FROM courses
8 WHERE cname = "Software Engineering")
```

部分结果如下：

sname
▶ rfslreav
qeiygk
xizbbugq
harwibwn
uxwpxjqjr
uyhrfe
okfbo
snldyau
jfekplki
eyozzi
liibh
yqeggyn
xwjubxj

7. 插入元组(600000000,823069829,10010,59)到视图CS中。若是在视图的定义中存在WITH CHECK OPTION子句对插入操作有什么影响?

- 当不加入with check option时:

不能插入, 因为cs(视图t1)是基于choice表, 而choices表中, choices(no,sid,cid,tid,score)中只有成绩可以为空, 其余值都为非空。而向视图插入数据时, 也会修改基本表, 但是该视图中没有tid元素, 故插入不成功, 会报如下错误:

**1423 - Field of view 'school.t1' underlying table doesn't have a default value**

- 当加入with check option时:

不能插入, 因为该元组成绩为59, 不符合t1表where筛选字段, 所以插入不成功。

8. 将视图CS (包含定义WITH CHECK OPTION)中, 所有课程编号为10010的课程的成绩都减去5分。这个操作数据库是否会正确执行, 为什么?如果加上5分(原来95分以上的不变)呢?

```

1  -- 减
2  UPDATE t1
3  SET score = score-5;
4
5  -- 加
6  UPDATE t1
7  SET score = score+5
8  WHERE score<=95;
```

减操作会失败, 因为可能会出现score减5以后小于60的情况。

**1369 - CHECK OPTION failed 'school.t1'**

加操作正常运行

```

UPDATE t1
SET score = score+5
WHERE score<=95
```

Affected rows: 215561

9. 在视图CS (包含定义WITH CHECK OPTION)删除编号为804529880学生的记录, 会产生什么结果?



```
1 DELETE FROM t1
2 WHERE sid = '804529880'
```

正常删除

DELETE FROM t1 WHERE sid = '804529880'	Affected rows: 5
---	------------------

## 10. 取消视图SCT和视图CS

```
1 DROP VIEW t1;
2 DROP VIEW t2;
3 DROP VIEW t3;
4 DROP VIEW t4;
5 DROP VIEW t5;
```

重新加载choices表后继续自我实践

# 五、自我实践

## 1. 定义选课信息和课程名称的视图VIEWC

```
1 CREATE OR REPLACE VIEW viewc AS
2     SELECT choices.*, cname
3     FROM choices
4     LEFT JOIN courses
5     ON courses.cid = choices.cid;
```

lab_database	开始事务	文本	筛选	排序	导出	创建图表
mysql	no	sid	tid	cid	score	cname
performance_schema	500000058	823069829	249596497	10037	76	software testing
school	500000253	829348273	202560416	10010	87	software engineering
表	500000984	850955252	234145610	10021	54	j2me
courses	500001270	847061074	292043491	10025	92	embeded system
students	500002143	860635914	238811498	10039	82	fortran
teachers	500002371	829785562	273189968	10028	77	architectonics
视图	500002715	822137137	218922066	10011	67	distributed computing
viewc	500003262	826310502	267846042	10005	90	c++
函数	500004144	817636568	253205179	10047	60	computer interface
查询	500005519	813520169	226385492	10023	(Null)	corba
备份	500005935	801967882	234419511	10021	70	j2me
sys	500005987	875434315	223646385	10048	82	data warehouse
	500006353	830180555	218440500	10016	76	data mining
	500006805	848035070	208952048	10007	88	uml
	500006824	834091581	201353263	10049	72	project management
	500007004	809548802	210870137	10002	64	operating system
	500007637	833961570	209860626	10004	80	java
	500007901	894256303	209831209	10018	75	unix/linux
	500008311	826412145	200207128	10009	63	cryptology
	500008620	836887807	294083883	10031	74	internet
	500009030	804529880	208801701	10023	80	corba

## 2. 定义学生姓名与选课信息的视图VIEWS

```
1 CREATE OR REPLACE VIEW views AS
2     SELECT choices.*, sname
3     FROM choices
4     LEFT JOIN students
5     ON students.sid = choices.sid;
```

lab_database	开始事务	文本	筛选	排序	导出	创建图表
mysql	no	sid	tid	cid	score	sname
performance_schema	500000058	823069829	249596497	10037	76	pxfys
school	500000253	829348273	202560416	10010	87	rfsleav
表	500000984	850955252	234145610	10021	54	baqzmo
courses	500001270	847061074	292043491	10025	92	qxkbh
students	500002143	860635914	238811498	10039	82	xnhdjo
teachers	500002371	829785562	273189968	10028	77	bemgynei
视图	500002715	822137137	218922066	10011	67	qaxwe
viewc	500003262	826310502	267846042	10005	90	cqkrjkuf
views	500004144	817636568	253205179	10047	60	wzinemrs
函数	500005519	813520169	226385492	10023	(Null)	rcypjhsnc
查询	500005935	801967882	234419511	10021	70	kvamveu
备份	500005987	875434315	223646385	10048	82	rekmgdbo
sys	500006353	830180555	218440500	10016	76	mbfrhwz
	500006805	848035070	208952048	10007	88	kxqlbun
	500006824	834091581	201353263	10049	72	xxamhe
	500007004	809548802	210870137	10002	64	myosjmd
	500007637	833961570	209860626	10004	80	bzbogqep
	500007901	894256303	209831209	10018	75	mamqnc
	500008311	826412145	200207128	10009	63	ouvxywha
	500008620	836887807	294083883	10031	74	mriky
	500009030	804529880	208801701	10023	80	sdqaqj

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

```

1 CREATE OR REPLACE VIEW s1(sid,sname,grade) AS
2     SELECT sid,sname,grade
3     FROM students
4     WHERE grade < 1998

```

information_schema	开始事务	文本	筛选	排序	导出
lab_database	sid	sname	grade		
mysql	800001216	gfxrgs	1992		
performance_schema	800005753	waqcj	1992		
school	800006682	fiiluommh	1992		
表	800006941	ogvmu	1995		
courses	800007595	uxqqbkjn	1997		
students	800009099	zapyv	1992		
teachers	800009249	zyuoh	1991		
视图	800010666	uwphrw	1992		
s1	800014004	aoaahudi	1994		
viewc	800014678	fnvxgrisg	1996		
views	800014991	mztqyvc	1994		
函数	800015960	yqjhake	1995		
查询	800016416	hvylafcj	1997		
备份	800016895	ucsqywg	1991		
sys	800017442	vocojnnml	1997		
	800017669	lkclbq	1995		
	800020409	yuitb	1993		
	800022091	kdthgvooo	1994		
	800022243	ssst	1996		
	800023031	aqyty	1991		
	800023963	uxjof	1996		
	800025157	axgrcq	1995		

### 4. 查询学生为“uxjof”的学生的选课信息

```

1 SELECT *
2 FROM choices
3 WHERE sid = (
4 SELECT sid
5 FROM students
6 WHERE sname = "uxjof");

```

no	sid	tid	cid	score
506978093	800023963	220667042	10046	94
541221076	800023963	238341990	10018	84
567316431	800023963	258375444	10037	98

5. 查询选修课程“UML”的学生的编号和成绩

```

1 SELECT sid,score
2 FROM choices
3 WHERE cid = (
4 SELECT cid
5 FROM courses
6 WHERE cname = "UML");

```

部分结果如下：

sid	score
848035070	88
897664264	62
898453203	91
846108663	86
827984677	86
823352185	(Null)
882778410	69
884993242	74
806427512	96
848803637	85
839666024	92
844745062	96
818493215	88
815954009	62
829224326	80

6. 向视图S1插入记录("60000001",Lily,2001")

因为没写with check option, 故会插入成功, 但是视图不会显示

```

1 INSERT INTO s1(sid,sname,grade) VALUES("60000001","Lily",2001)

```

INSERT INTO s1(sid,sname,grade) VALUES("60000001","Lily",2001)

Affected rows: 1

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

定义包括更新和插入约束的视图S1：

```
1 CREATE OR REPLACE VIEW s1(sid,sname,grade) AS
2     SELECT sid,sname,grade
3     FROM students
4     WHERE grade < 1998
5     with check option
```

插入语句：

```
1 INSERT INTO s1(sid,sname,grade) VALUES("60000001","Lily",2001)
```

会报错，因为2001大于1998，会被check option给拒掉。

1369 - CHECK OPTION failed 'school.s1'

删除：

```
1 DELETE FROM s1
2 WHERE grade = 1999
```

运行成功，但是影响行数为0：

```
DELETE FROM s1
WHERE grade = 1999
```

Affected rows: 0

8. 在视图VIEWS中将姓名为“uxjof”的学生的选课成绩都加上5分

使用如下代码：

```
1 UPDATE views
2 SET score = score+5
3 WHERE sname = "uxjof"
```

添加失败，报错如下：

1288 - The target table views of the UPDATE is not updatable

这是因为views视图是由choices和students表连接生成的，对其进行更新操作是不允许的。

1. **缺少表的主键或唯一约束**：在update语句中，如果目标表没有主键或唯一约束，那么更新操作可能会导致数据不一致性。
2. **使用子查询或视图**：如果UPDATE语句中使用了子查询或视图作为目标表，则可能会导致目标表不可更新。因为子查询或视图的结果集不能直接更新。
3. **使用多表连接**：如果UPDATE语句中使用了多个表进行连接，并且目标表与其他表之间存在关联，那么可能会导致目标表不可更新。这通常是由于连接条件不正确或连接不唯一的结果导致的。
4. **使用触发器**：如果在目标表上定义了触发器，并且触发器的操作导致目标表不可更新，那么更新操作将失败。

9. 取消以上建立的所有视图

```
1 DROP VIEW viewc;  
2 DROP VIEW views;  
3 DROP VIEW s1;
```