

实验七 视图

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。

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](#)

一、创建可更新视图：

```
create or replace view usa_city as
select city_id,city,country_id
from city
where country_id=103 order by city;

select * from usa_city;
```

更新视图插入数据：

```
insert into usa_city (city,country_id )values ('Birmingham', 102);
```

结果：插入成功，但是此插入的结果并不在视图usa_city中

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

WITH CHECK OPTION 子句。

二、创建有检查项的可更新视图

```
create or replace view usa_city as
select city_id,city,country_id
from city
where country_id=103 order by city
with check option;
```

更新视图插入数据:

```
insert into usa_city (city,country_id )values ('Birmingham', 102);
```

结果: > 错误: 新行违反了视图"usa_city"的检查选项

```
insert into usa_city (city,country_id )values ('Birenc', 103);
```

结果: 插入成功, 且只允许插入的数据满足原视图的where条件

三、检查项含local的可更新视图

(1)、创建一个可更新的基表视图

```
create or replace view usa_a as
select city_id,city,country_id
from city
where city like 'A%';
```

(2)、创建检查项含local的可更新视图

```
create or replace view usa_a_city as
select city_id,city,country_id
from usa_a
where country_id=103 order by city
with local check option;
```

更新视图插入数据:

```
insert into usa_a_city(city,country_id)values('Mirmin', 103);
```

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

```
insert into usa_a_city(city,country_id)values('Mirmin', 102);
```

结果: 插入失败, > 错误: 新行违反了视图"usa_a_city"的检查选项

(3)、创建检查项含cascaded的可更新视图

```
create or replace view usa_a_city as
select city_id,city,country_id
from usa_a
where country_id=103 order by city
with cascaded check option;
```

更新视图插入数据:

```
insert into usa_a_city(city,country_id)values('Meery', 103);
```

结果: 插入失败, > 错误: 新行违反了视图"usa_a"的检查选项

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

```
insert into usa_a_city(city,country_id)values('Aeery', 103);
```

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

课内实验

要求:

以school数据库为例(与之前实验的数据同), 在该数据库中存在4张表格, 分别为:

STUDENTS(sid,sname,email,grade)

TEACHERS(tid,tname,email,salary)

COURSES(cid,cname,hour)

CHOICES(no,sid,tid,cid,score)

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
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
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
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
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
6. 查询所有选修课程Software Engineering的学生姓名
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?
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?
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?
10. 取消视图SCT和视图CS

自我实践

1. 定义选课信息和课程名称的视图VIEWC
2. 定义学生姓名与选课信息的视图VIEWS
3. 定义年级低于1998的学生的视图S1(SID,SNAME,GRADE)
4. 查询学生为"uxjof"的学生的选课信息
5. 查询选修课程"UML"的学生的编号和成绩
6. 向视图S1插入记录("600000001,Lily,2001")
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("600000001,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
8. 在视图VIEWS中将姓名为"uxjof"的学生的选课成绩都加上5分
9. 取消以上建立的所有视图

实验报告提交要求

实验报告(课内实验+自我实践)至少包括以下内容: ①实验目的; ②实验环境; ③实验内容、步骤、结果和实验过程中出现的问题; ④实验过程中主要使用的SQL语句。

注意: 本学期理论课和实验课作业(练习)迟交政策: 截止日期后两周内补交分数打九折, 期末考试前补交分数打八折。