

浙江工业大学

数据库原理及应用实验报告

(2021 级)



实验题目 实验 4 SQL 的视图

学生姓名 温家伟

学生学号 202103151422

学科(专业) 大数据分析 2101 班

所在学院 理学院

提交日期 2023 年 4 月 23 日

实验4、SQL的视图

4.1 实验目的

熟悉SQL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，对视图进行查询和删除视图。定义数据库外模式，并能使用所创建的视图实现数据管理。理解和掌握视图消解执行原理，掌握可更新视图和不可更新视图的区别。理解物化视图。

4.2 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—仅适用SQL Server DBMS】
- 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
- 全量物化视图CREATE MATERIALIZED VIEW

(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。

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。

4.3 实验步骤

- 创建一个行列子集视图CS_View，给出选课成绩合格的学生的编号、教师编号、所选课程号和该课程成绩，并查询该视图。

```
create view stuEmail AS select sno, sname, semail from students;
```

```
create view CS_view as select sno,tno,cno,score from STC where score>=60;
```

要允许视图可更新：需要在数据库系统中创建一条插入视图的规则

```
CREATE OR REPLACE RULE test_ins AS ON INSERT TO stuemail DO INSTEAD INSERT INTO students(sno, sname,semail) VALUES (new.sno, new.sname,new.semail);
```

```
ahweidb=# create view stuEmail AS select sno, sname, semail from students;
CREATE VIEW
ahweidb=# select * from stuEmail;
 sno | sname |      semail
-----+-----+-----
 S01 | 王建平 | WJP@zjut.edu.cn
 S02 | 刘华  | LH@zjut.edu.cn
 S03 | 范林军 | FLJ@zjut.edu.cn
 S04 | 李伟  | LW@zjut.edu.cn
 S26 | 黄河  | HUanghe@zjut.edu.cn
 S52 | 长江  | Changjiang@zjut.edu.cn
 S100 | 刘华  |
(7 rows)
```

```

ahweidb=# create view CS_view as select sno,tno,cno,score from STC where score>=60;
CREATE VIEW
ahweidb=# select * from CS_view;
 sno | tno | cno | score
-----+-----+-----+-----
 S01 | T01 | C01 | 83.0
 S01 | T03 | C03 | 85.0
 S02 | T01 | C01 | 75.0
 S02 | T05 | C05 | 70.0
 S02 | T04 | C06 | 83.0
 S02 | T05 | C07 | 90.0
 S02 | T01 | C08 | 83.0
 S02 | T02 | C09 | 77.0
 S02 | T07 | C10 | 83.0
 S02 | T06 | C11 | 88.0
 S03 | T01 | C08 | 63.0
 S03 | T02 | C02 | 93.0
 S03 | T01 | C01 | 78.0
 S04 | T06 | C06 | 89.0
 S04 | T05 | C05 | 93.0
 S26 | T04 | C04 | 86.0
 S52 | T07 | C10 | 91.0
 S52 | T06 | C11 | 90.0
 S52 | T01 | C08 | 64.0
 S52 | T02 | C09 | 81.0
(20 rows)

ahweidb=# CREATE OR REPLACE RULE test_ins AS ON INSERT TO stuemail DO INSTEAD INSERT INTO students(sno, sname,semail) VALUES (new.sno, new.sname,new.sema
il);
CREATE RULE

```

- 创建基于多个基本表的视图STC_View，这个视图由学生姓名和他所选修的课程名及讲授该课程的教师姓名构成，并查询该视图。从视图STC_View中查询所有选修课程“数据库原理及应用”的学生姓名。

```

ahweidb=# create view STC_view as select s.sname, c.cname, t.tname from Students s, Courses c, Teachers t, STC where s.sno = stc.sno and c.cno = stc.cno
and t.tno = stc.tno;
CREATE VIEW
ahweidb=# select * from STC_view;
 sname |      cname      | tname
-----+-----+-----
王建平 | C++              | 刘涛
王建平 | JAVA            | 张莹
刘华   | C++              | 刘涛
刘华   | UML              | 吴碧艳
刘华   | JAVA            | 张莹
刘华   | 算法分析与设计  | 张宁雅
刘华   | 数据库原理及应用 | 叶帅
刘华   | 数据结构与算法  | 张宁雅
刘华   | 计算机组成原理  | 叶帅
刘华   | 英语            | 刘涛
刘华   | 数字生活        | 吴碧艳
刘华   | 音乐鉴赏        | 程潜
刘华   | 体育1           | 杨光美
范林军 | 英语            | 刘涛
范林军 | UML              | 吴碧艳
范林军 | C++              | 刘涛
李伟   | 数据结构与算法  | 杨光美
李伟   | 数据库原理及应用 | 叶帅
黄河   | 音乐鉴赏        | 程潜
黄河   | 算法分析与设计  | 张宁雅
长江   | 音乐鉴赏        | 程潜
长江   | 体育1           | 杨光美
长江   | 数据库原理及应用 | 叶帅
长江   | 英语            | 刘涛
长江   | 数字生活        | 吴碧艳
(25 rows)

```

- 创建带表达式的视图EXP_View，由学生姓名及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，并查询该视图。

```

ahweidb=# Create view EXP_View(Name, CourseName, Score) AS SELECT S.sname, C.Cname, stc.score+5 from Students S, Courses C, STC
ahweidb=# WHERE S.sno=stc.sno AND C.cno=STC.cno;
CREATE VIEW
ahweidb=# select * from EXP_View;

```

name	coursename	score
王建平	C++	88.0
王建平	JAVA	90.0
刘华	C++	80.0
刘华	UML	50.0
刘华	JAVA	
刘华	算法分析与设计	
刘华	数据库原理及应用	75.0
刘华	数据结构与算法	88.0
刘华	计算机组成原理	95.0
刘华	英语	88.0
刘华	数字生活	82.0
刘华	音乐鉴赏	88.0
刘华	体育1	93.0
范林军	英语	68.0
范林军	UML	98.0
范林军	C++	83.0
李伟	数据结构与算法	94.0
李伟	数据库原理及应用	98.0
黄河	音乐鉴赏	50.0
黄河	算法分析与设计	91.0
长江	音乐鉴赏	96.0
长江	体育1	95.0
长江	数据库原理及应用	
长江	英语	69.0
长江	数字生活	86.0

```

(25 rows)

```

- 创建分组视图Group_View，将学生的学号及他的平均成绩定义为一个视图，并查询该视图。

```

ahweidb=# CREATE VIEW Group_View(sno,avgScore) AS SELECT sno,avg(score) from STC GROUP BY sno;
CREATE VIEW
ahweidb=# select * from Group_View;

```

sno	avgscore
S04	91.0000000000000000
S02	77.1111111111111111
S03	78.0000000000000000
S01	84.0000000000000000
S52	81.5000000000000000
S26	65.5000000000000000

```

(6 rows)

```

- 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VV_View，并查询该视图。

```

ahweidb=# CREATE VIEW VV_View(sno, CoursesSelected, avgscore) AS SELECT sno, count(cno), avg(score) FROM CS_view GROUP BY sno;
CREATE VIEW
ahweidb=# select * from VV_View;

```

sno	coursesselected	avgscore
S26	1	86.0000000000000000
S04	2	91.0000000000000000
S02	8	81.1250000000000000
S03	3	78.0000000000000000
S01	2	84.0000000000000000
S52	4	81.5000000000000000

```

(6 rows)

```

- 创建一个物化视图MV_CS_View，给出选课成绩合格的学生的编号、教师编号、所选课程号和该课程成绩，并查询该物化视图。向物化视图MV_CS_View中基表STC插入数据('S52','T05','C07',53)，并查询视图MV_CS_View。向物化视图MV_CS_View中基表STC插入数据('S52','T05','C03',83)，并查询视图MV_CS_View。

```

ahweidb=# create materialized view MV_CS_view as select sno, tno, cno, score from STC where score>=60
ahweidb=# WITH DATA;
SELECT 20

```

```

ahweidb=# select * from MV_CS_view;
 sno | tno | cno | score
-----+-----+-----+-----
S01  | T01 | C01 | 83.0
S01  | T03 | C03 | 85.0
S02  | T01 | C01 | 75.0
S02  | T05 | C05 | 70.0
S02  | T04 | C06 | 83.0
S02  | T05 | C07 | 90.0
S02  | T01 | C08 | 83.0
S02  | T02 | C09 | 77.0
S02  | T07 | C10 | 83.0
S02  | T06 | C11 | 88.0
S03  | T01 | C08 | 63.0
S03  | T02 | C02 | 93.0
S03  | T01 | C01 | 78.0
S04  | T06 | C06 | 89.0
S04  | T05 | C05 | 93.0
S26  | T04 | C04 | 86.0
S52  | T07 | C10 | 91.0
S52  | T06 | C11 | 90.0
S52  | T01 | C08 | 64.0
S52  | T02 | C09 | 81.0
(20 rows)

```

```

ahweidb=# insert into stc values ('S52','T05','C07',53);
INSERT 0 1
ahweidb=# select * from MV_CS_view;
 sno | tno | cno | score
-----+-----+-----+-----
 S01 | T01 | C01 | 83.0
 S01 | T03 | C03 | 85.0
 S02 | T01 | C01 | 75.0
 S02 | T05 | C05 | 70.0
 S02 | T04 | C06 | 83.0
 S02 | T05 | C07 | 90.0
 S02 | T01 | C08 | 83.0
 S02 | T02 | C09 | 77.0
 S02 | T07 | C10 | 83.0
 S02 | T06 | C11 | 88.0
 S03 | T01 | C08 | 63.0
 S03 | T02 | C02 | 93.0
 S03 | T01 | C01 | 78.0
 S04 | T06 | C06 | 89.0
 S04 | T05 | C05 | 93.0
 S26 | T04 | C04 | 86.0
 S52 | T07 | C10 | 91.0
 S52 | T06 | C11 | 90.0
 S52 | T01 | C08 | 64.0
 S52 | T02 | C09 | 81.0
(20 rows)

```

```

ahweidb=# insert into stc values ('S52','T05','C03',83);
INSERT 0 1
ahweidb=# select * from MV_CS_view;
 sno | tno | cno | score
-----+-----+-----+-----
 S01 | T01 | C01 | 83.0
 S01 | T03 | C03 | 85.0
 S02 | T01 | C01 | 75.0
 S02 | T05 | C05 | 70.0
 S02 | T04 | C06 | 83.0
 S02 | T05 | C07 | 90.0
 S02 | T01 | C08 | 83.0
 S02 | T02 | C09 | 77.0
 S02 | T07 | C10 | 83.0
 S02 | T06 | C11 | 88.0
 S03 | T01 | C08 | 63.0
 S03 | T02 | C02 | 93.0
 S03 | T01 | C01 | 78.0
 S04 | T06 | C06 | 89.0
 S04 | T05 | C05 | 93.0
 S26 | T04 | C04 | 86.0
 S52 | T07 | C10 | 91.0
 S52 | T06 | C11 | 90.0
 S52 | T01 | C08 | 64.0
 S52 | T02 | C09 | 81.0
(20 rows)

```

```

ahweidb=# REFRESH MATERIALIZED VIEW MV_CS_view;
REFRESH MATERIALIZED VIEW
ahweidb=# select * from MV_CS_view;
 sno | tno | cno | score
-----+-----+-----+-----
 S01 | T01 | C01 | 83.0
 S01 | T03 | C03 | 85.0
 S02 | T01 | C01 | 75.0
 S02 | T05 | C05 | 70.0
 S02 | T04 | C06 | 83.0
 S02 | T05 | C07 | 90.0
 S02 | T01 | C08 | 83.0
 S02 | T02 | C09 | 77.0
 S02 | T07 | C10 | 83.0
 S02 | T06 | C11 | 88.0
 S03 | T01 | C08 | 63.0
 S03 | T02 | C02 | 93.0
 S03 | T01 | C01 | 78.0
 S04 | T06 | C06 | 89.0
 S04 | T05 | C05 | 93.0
 S26 | T04 | C04 | 86.0
 S52 | T07 | C10 | 91.0
 S52 | T06 | C11 | 90.0
 S52 | T01 | C08 | 64.0
 S52 | T02 | C09 | 81.0
 S52 | T05 | C03 | 83.0
(21 rows)

```

当向基表插入数据时，视图并不会自动更新。这是因为视图是基于 SELECT 语句生成的虚拟表，它们只是对基表的一种查询结果的展现，而不是实际存储数据的表。如果想要更新视图，需要重新运行 SELECT 语句来生成新的视图结果。

- 创建一个带 `WITH CHECK OPTION` 子句的视图 `CS_View_opt`，给出选课成绩合格的学生的编号、教师编号、所选课程号和该课程成绩，并查询该视图。

```

mysql> create view CS_View_opt as select STC.sno, STC.tno, STC.cno, STC.score from STC inner join Courses on STC.cno = Courses.cno where score >= 60 with check option;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from CS_View_opt;
+-----+-----+-----+-----+
| sno | tno | cno | score |
+-----+-----+-----+-----+
| S01 | T01 | C01 | 83.0 |
| S01 | T03 | C03 | 85.0 |
| S02 | T01 | C01 | 75.0 |
| S02 | T05 | C05 | 70.0 |
| S02 | T04 | C06 | 83.0 |
| S02 | T05 | C07 | 90.0 |
| S02 | T01 | C08 | 83.0 |
| S02 | T02 | C09 | 77.0 |
| S02 | T07 | C10 | 83.0 |
| S02 | T06 | C11 | 88.0 |
| S03 | T01 | C08 | 63.0 |
| S03 | T02 | C02 | 93.0 |
| S03 | T01 | C01 | 78.0 |
| S04 | T06 | C06 | 89.0 |
| S04 | T05 | C05 | 93.0 |
| S26 | T04 | C04 | 86.0 |
| S52 | T07 | C10 | 91.0 |
| S52 | T06 | C11 | 90.0 |
| S52 | T01 | C08 | 64.0 |
| S52 | T02 | C09 | 81.0 |
+-----+-----+-----+-----+
20 rows in set (0.00 sec)

```

- 删除视图 `SCT_View`、视图 `CS_View` 和视图 `CS_View_opt`，删除物化视图 `MV_CS_View`。思考物化视图与普通视图的区别。


```
ahweidb=# drop materialized view mv_cs_view;  
DROP MATERIALIZED VIEW  
ahweidb=# drop view stc_view, cs_view;  
ERROR:  cannot drop desired object(s) because other objects depend on them  
DETAIL:  view vv_view depends on view cs_view  
HINT:  Use DROP ... CASCADE to drop the dependent objects too.  
ahweidb=# drop view vv_view;  
DROP VIEW  
ahweidb=# drop view stc_view, cs_view;  
DROP VIEW  
ahweidb=#
```

物化视图和普通视图都是数据库中的视图，但是它们的定义和使用方式有所不同。

普通视图是一种虚拟表，它是基于一个或多个表的查询结果而创建的。普通视图可以用来简化复杂的查询，过滤数据，以及保护数据安全性。

物化视图也是一种表，但是它不是虚拟的，而是实际存在于数据库中的物理表。物化视图的数据是预先计算好并存储在物理表中的，因此查询速度更快。但是，物化视图需要占用更多的存储空间，并且在底层数据发生变化时需要手动刷新。

在某些情况下，物化视图可以替代普通视图来提高查询性能。例如，在大型数据仓库中，当查询的数据量非常庞大时，使用物化视图可以显著提高查询效率。同时，物化视图也可以用来缓存常用查询结果，以减少查询的响应时间。

因此，物化视图和普通视图并不是互相排斥的关系，而是根据不同的需求选择使用不同类型的视图。

4.4 实验要求

(1) 实验之前请仔细阅读实验总体要求与说明指导书

(2) 在DataStudio的SQL终端环境中，完成以上实验(1)-(7)(9)步所有SQL操作，在SQL Server中操作完成实验(8)步并将其中粗体步骤的操作窗口剪贴到实验报告中。