

# 实验六 视图定义和操作

## 计算机科学与技术

2021160291 李景昊

### 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.实验内容

1. 定义常见的视图形式，包括：
  1. 行列子集视图
  2. WITH CHECK OPTION 的视图
  3. 基于多个基表的视图
  4. 基于视图的视图
  5. 带表达式的视图
  6. 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

### 三.实验任务

数据库 School，在该数据库中存在四张表格，分别为：

学生表 STUDENTS（学号sid，学生姓名sname，email，年级grade，性别Sex，年龄age，系sdept）

教师表 TEACHERS（职工号tid，职工姓名tname，email，工资salary）

课程表 COURSES (课程号cid, 课程名称cname, 课时hour, 先行课Cpno)  
选课表 CHOICES (记录号no, 学号sid, 职工号tid, 课程号cid, 成绩sorce)

student

```
console x student
CREATE table STUDENT
(
  sid integer,
  sName varchar(10),
  email text,
  grade integer,
  sex varchar(10),
  age integer,
  sdept varchar(10)
);
```

WHERE ORDER BY sid

	sid	sname	email	grade	sex	age	sdept
1	1	John Doe	john.doe@example.com	1	Male	19	Computer Science
2	2	Jane Smith	jane.smith@example.com	2	Female	20	Mathematics
3	3	Mike Johnson	mike.johnson@example.com	3	Male	21	Physics
4	4	Emily Davis	emily.davis@example.com	1	Female	18	Biology
5	5	Chris Wilson	chris.wilson@example.com	4	Male	22	Chemistry
6	6	Amanda Lee	amanda.lee@example.com	2	Female	20	History
7	7	Robert Brown	robert.brown@example.com	3	Male	21	English
8	8	Sarah Hall	sarah.hall@example.com	4	Female	22	Economics
9	9	Michael Moore	michael.moore@example.com	1	Male	18	Psychology
10	10	Linda Clark	linda.clark@example.com	2	Female	20	Sociology

teacher

```
console x student
CREATE table TEACHER
(
  t_id integer,
  tName varchar(10),
  email text,
  salary integer
);
```

		WHERE	ORDER BY		
		t_id	tname	email	salary
1	1	Alice Smith	alice.smith@example.com	50000	
2	2	Bob Johnson	bob.johnson@example.com	55000	
3	3	Charlie Brown	charlie.brown@example.com	60000	
4	4	David Lee	david.lee@example.com	52000	
5	5	Ella Davis	ella.davis@example.com	58000	
6	6	Frank Wilson	frank.wilson@example.com	62000	
7	7	Grace Hall	grace.hall@example.com	54000	
8	8	Henry Clark	henry.clark@example.com	56000	
9	9	Ivy Moore	ivy.moore@example.com	59000	
10	10	Jack Taylor	jack.taylor@example.com	61000	

courses

```

1  ✓ CREATE table COURSES
2  (
3      cid          integer,
4      cName        varchar(10),
5      hour         integer,
6      Cpno         varchar(10)
7  );
8

```

		WHERE	ORDER BY		
		cid	cname	hour	cpno
1	1	Software Engineering	40	<null>	
2	2	Database Management	30	<null>	
3	3	Data Structures	35	<null>	
4	4	Algorithms	35	3	
5	5	Web Development	25	<null>	
6	6	Artificial Intelligence	45	3	
7	7	Computer Networks	30	<null>	
8	8	Operating Systems	35	3	
9	9	Software Testing	20	1	
10	10	Mobile App Development	30	5	

choice

```
✓ CREATE table CHOICES
(
  no integer,
  sid integer,
  t_id integer,
  cid integer,
  sorce float4
);
```

	no	sid	t_id	cid	score
1	1	1	1	1	90
2	2	2	2	2	60
3	3	3	3	3	92
4	4	4	4	4	88
5	5	5	5	5	40
6	6	6	6	6	78
7	7	7	7	7	86
8	8	8	8	8	55
9	9	9	9	9	89
10	10	10	10	10	94

(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。

```
1 ✓ CREATE VIEW PassChoices AS
2   SELECT c.sid, c.cid, c.score
3   FROM CHOICES c
4   WHERE c.score >= 60;
5
```

选课表

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

WHERE

	no	sid	t_id	cid	score
1	1	1	1	1	90
2	2	2	2	2	60
3	3	3	3	3	92
4	4	4	4	4	88
5	5	5	5	5	40
6	6	6	6	6	78
7	7	7	7	7	86
8	8	8	8	8	55
9	9	9	9	9	89
10	10	10	10	10	94

合格人数视图

console passchoices × courses

8 行

WHERE ORDER BY

	sid	cid	score
1	1	1	90
2	2	2	60
3	3	3	92
4	4	4	88
5	6	6	78
6	7	7	86
7	9	9	89
8	10	10	94

(2) 创建基于多个基表的视图SCT，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

## SCT视图

```
1 ✓ CREATE VIEW SCT AS
2 SELECT
3     s.sname AS student_name,
4     c.cname AS course_name,
5     t.tname AS teacher_name
6 FROM
7     CHOICES ch
8     JOIN STUDENT s ON ch.sid = s.sid
9     JOIN COURSES c ON ch.cid = c.cid
10    JOIN TEACHER t ON c.cid = t.t_id;
```

选课表，教师表，学生表和课程表都是顺序递增

	student_name	course_name	teacher_name
1	John Doe	Software Engineering	Alice Smith
2	Jane Smith	Database Management	Bob Johnson
3	Mike Johnson	Data Structures	Charlie Brown
4	Emily Davis	Algorithms	David Lee
5	Chris Wilson	Web Development	Ella Davis
6	Amanda Lee	Artificial Intelligence	Frank Wilson
7	Robert Brown	Computer Networks	Grace Hall
8	Sarah Hall	Operating Systems	Henry Clark
9	Michael Moore	Software Testing	Ivy Moore
10	Linda Clark	Mobile App Development	Jack Taylor

**(3) 创建带表达式的视图，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**

```

1 ✓ CREATE VIEW gradePlusFive AS
2 SELECT
3     s.sname AS student_name,
4     c.cname AS course_name,
5     (ch.score+5) as score
6 FROM
7     CHOICES ch
8     JOIN STUDENT s ON ch.sid = s.sid
9     JOIN COURSES c ON ch.cid = c.cid

```

加5分视图

	student_name	course_name	score
1	John Doe	Software Engineering	95
2	Jane Smith	Database Management	65
3	Mike Johnson	Data Structures	97
4	Emily Davis	Algorithms	93
5	Chris Wilson	Web Development	45
6	Amanda Lee	Artificial Intelligence	83
7	Robert Brown	Computer Networks	91
8	Sarah Hall	Operating Systems	60
9	Michael Moore	Software Testing	94
10	Linda Clark	Mobile App Development	99

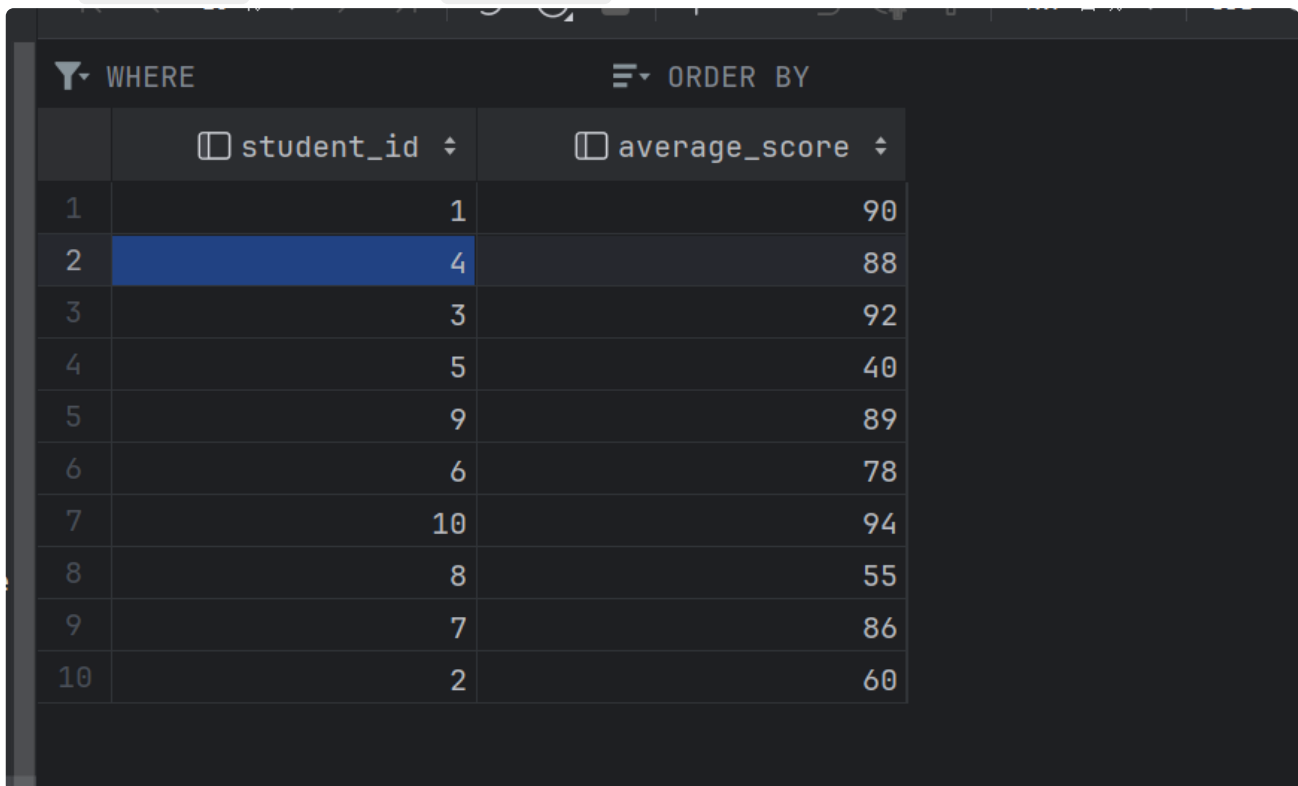
(4) 创建分组视图，将学生的学号及他的平均成绩定义为一个视图



## 使用 AVG 计算平均值

```
1 ✓ CREATE VIEW StudentAverageGrades AS
2 SELECT
3     s.sid AS student_id,
4     AVG(ch.score) AS average_score
5 FROM
6     STUDENT s
7     LEFT JOIN CHOICES ch ON s.sid = ch.sid
8 GROUP BY
9     s.sid;
```

使用 LEFT JOIN 连接两个表, LEFT JOIN 的结果中行的顺序通常是不确定的



	student_id	average_score
1	1	90
2	4	88
3	3	92
4	5	40
5	9	89
6	6	78
7	10	94
8	8	55
9	7	86
10	2	60

**(5) 创建一个基于视图的视图，基于题目 (1) 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**

```
console x passchoices studentaveragegrade
1 ✓ CREATE VIEW StudentSummary AS
2 SELECT
3     pc.sid,
4     COUNT(pc.cid) AS course_count,
5     AVG(pc.score) AS average_score
6 FROM
7     passchoices pc
8 GROUP BY
9     pc.sid;
10
```

学生选课数目和平均成绩

	sid	course_count	average_score
1	1	1	90
2	6	1	78
3	4	1	88
4	3	1	92
5	10	1	94
6	7	1	86
7	2	1	60
8	9	1	89

(6) 创建一个选修课程 software engineering 的学生姓名、学号的视图

```
1 ✓ CREATE VIEW choiceSE AS
2 SELECT
3     s.sid,
4     s.sname
5 FROM
6     student s
7 join
8     passchoices pc on s.sid = pc.sid
9 join
10    courses c on c.cid = pc.cid
11 where
12    c.cname = 'Software Engineering';
```

console choiceSE × passchoices

1 行

WHERE ORDER BY

	sid	sname
1	1	John Doe

(7) 在表 CHOICES上建立成绩合格的学生视图CS(sid, tid, cid, sorce), 通过视图CS插入元组 (600000000,823069829, 10010, 59) 。若视图的定义中存在 WITH CHECK OPTION 子句对插入操作有什么影响?

opengauss不支持with check option?

```
1  -- 创建视图 CS 并包含 WITH CHECK OPTION 子句
2  CREATE VIEW CS AS
3  SELECT
4      sid,
5      t_id,
6      cid,
7      score
8  FROM
9      CHOICES
10 WHERE
11     score >= 60
12 WITH CHECK OPTION;
13
14
```

[0A000] ERROR: WITH CHECK OPTION is not implemented

结论是：如果尝试插入一个符合视图筛选条件的元组，它将被接受，否则报错

**(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？**

这取决于score的格式和with check option字句是否允许修改，如果通过了，则会正确执行

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

和上述回答一样

**(10) 取消视图 SCT 和视图 CS**

