



南京大学软件学院
NANJING UNIVERSITY · SOFTWARE INSTITUTE



数据管理基础

ch22 视图

Software Institute
Nanjing University
Bei Jia

- 视图的特点

- 虚表，是从一个或几个基本表（或视图）导出的表
- 只存放视图的定义，不存放视图对应的数据
- 基表中的数据发生变化，从视图中查询出的数据也随之改变

定义视图-建立视图 1

- 语句格式

CREATE VIEW

<视图名> [(<列名> [,<列名>]...)]

AS <子查询>

[WITH CHECK OPTION];

- WITH CHECK OPTION

- 对视图进行UPDATE, INSERT和DELETE操作时要保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）

- 子查询可以是任意的SELECT语句，是否可以含有ORDER BY子句和DISTINCT短语，则决定具体系统的实现。



定义视图-建立视图 2

- 组成视图的属性列名：全部省略或全部指定

- 全部省略:

- ◆ 由子查询中SELECT目标列中的诸字段组成

- 明确指定视图的所有列名:

- ◆ 某个目标列是聚集函数或列表达式
- ◆ 多表连接时选出了几个同名列作为视图的字段
- ◆ 需要在视图中为某个列启用新的更合适的名字



定义视图-建立视图 3

- 关系数据库管理系统执行CREATE VIEW语句时只是把视图定义存入数据字典，并不执行其中的SELECT语句。
- 在对视图查询时，按视图的定义从基本表中将数据查出。
- [例3.84] 建立信息系学生的视图。

```
CREATE VIEW IS_Student
```

```
AS
```

```
SELECT Sno,Sname,Sage
```

```
FROM Student
```

```
WHERE Sdept= 'IS';
```



定义视图-建立视图 4

- [例3.85]建立信息系学生的视图,并要求进行修改和插入操作时仍需保证该视图只有信息系的学生。

```
CREATE VIEW IS_Student  
AS  
SELECT Sno,Sname,Sage  
FROM Student  
WHERE Sdept= 'IS'  
WITH CHECK OPTION;
```

- 定义IS_Student视图时加上了WITH CHECK OPTION子句,对该视图进行插入、修改和删除操作时,RDBMS会自动加上Sdept='IS'的条件。
- 若一个视图是从单个基本表导出的,并且只是去掉了基本表的某些行和某些列,但保留了主码,我们称这类视图为行列子集视图。
 - IS_Student视图就是一个行列子集视图。



定义视图-建立视图 5

- 基于多个基表的视图
- [例3.86] 建立信息系选修了1号课程的学生的视图（包括学号、姓名、成绩）。

```
CREATE VIEW IS_S1(Sno,Sname,Grade)
```

```
AS
```

```
SELECT Student.Sno,Sname,Grade
```

```
FROM Student,SC
```

```
WHERE Sdept= 'IS' AND
```

```
Student.Sno=SC.Sno AND
```

```
SC.Cno= '1';
```



定义视图-建立视图 6

- 基于视图的视图
- [例3.87] 建立信息系选修了1号课程且成绩在90分以上的学生的视图。
- 带表达式的视图
- [例3.88] 定义一个反映学生出生年份的视图。

```
CREATE VIEW IS_S2  
AS  
SELECT  
Sno,Sname,Grade  
FROM IS_S1  
WHERE Grade>=90;
```

```
CREATE VIEW  
BT_S(Sno,Sname,Sbirth)  
AS  
SELECT  
Sno,Sname,2014-Sage  
FROM Student;
```



定义视图-建立视图 7

- 分组视图
- [例3.89] 将学生的学号及平均成绩定义为一个视图

```
CREAT VIEW  
S_G(Sno,Gavg)  
AS  
SELECT  
Sno,AVG(Grade)  
FROM SC  
GROUP BY Sno;
```

- [例3.90]将Student表中所有女生记录定义为一个视图

```
CREATE VIEW  
F_Student(F_Sno,name,sex,age,dept)  
AS  
SELECT * /*不指定属性列*/  
FROM Student  
WHERE Ssex='女';
```

- 缺点:
 - 修改基表Student的结构后, Student表与F_Student视图 的映象关系被破坏, 导致该视图不能正确工作。



定义视图-删除视图 1

- 语句的格式:

`DROP VIEW <视图名>[CASCADE];`

- 该语句从数据字典中删除指定的视图定义
- 如果该视图上还导出了其他视图，使用
CASCADE级联删除语句，把该视图和由它导出的所有视图一起删除
- 删除基表时，由该基表导出的所有视图定义都必须显式地使用DROP VIEW语句删除



定义视图-删除视图 2

- [例3.91] 删除视图BT_S和IS_S1

DROP VIEW BT_S; /*成功执行*/

DROP VIEW IS_S1; /*拒绝执行*/

- 要删除IS_S1, 需使用级联删除:

DROP VIEW IS_S1 CASCADE;



查询视图 1

- 用户角度：查询视图与查询基本表相同
- 关系数据库管理系统实现视图查询的方法
 - 视图消解法 (View Resolution)
 - ◆ 进行有效性检查
 - ◆ 转换成等价的对基本表的查询
 - ◆ 执行修正后的查询



查询视图 2

- [例3.92] 在信息系学生的视图中找出年龄小于20岁的学生。

```
SELECT Sno,Sage
FROM IS_Student
WHERE Sage<20;
```

- 视图消解转换后的查询语句为:

```
SELECT Sno,Sage
FROM Student
WHERE Sdept= 'IS' AND Sage<20;
```

- [例3.93] 查询选修了1号课程的信息系学生

```
SELECT IS_Student.Sno,Sname
FROM IS_Student,SC
WHERE IS_Student.Sno =SC.Sno AND
SC.Cno= '1';
```



查询视图 3

- 视图消解法的局限

- 有些情况下，视图消解法不能生成正确的查询。

- [例3.94]在S_G视图中查询平均成绩在90分以上的学生学号和平均成绩

```
SELECT *  
FROM S_G  
WHERE Gavg>=90;
```

- S_G视图的子查询定义：

```
CREATE VIEW S_G (Sno,Gavg)  
AS  
SELECT Sno,AVG(Grade)  
FROM SC  
GROUP BY Sno;
```

- 错误：

```
SELECT Sno, AVG(Grade)  
FROM SC  
WHERE AVG(Grade)>=90  
GROUP BY Sno;
```

- 正确：

```
SELECT Sno,AVG(Grade)  
FROM SC  
GROUP BY Sno  
HAVING AVG(Grade)>=90;
```

- 或者

```
SELECT *  
FROM (SELECT Sno,AVG(Grade)  
FROM SC  
GROUP BY Sno) AS S_G(Sno,Gavg)  
WHERE Gavg>=90;
```



更新视图 1

- [例3.95] 将信息系学生视图IS_Student中学号"201215122"的学生姓名改为"刘辰"。

```
UPDATE IS_Student  
SET Sname= '刘辰'  
WHERE Sno= ' 201215122 ';
```

- 转换后的语句:

```
UPDATE Student  
SET Sname= '刘辰'  
WHERE Sno= ' 201215122 ' AND Sdept= 'IS';
```



更新视图 2

- [例3.96] 向信息系学生视图IS_S中插入一个新的学生记录，其中学号为“201215129”，姓名为“赵新”，年龄为20岁

INSERT

INTO IS_Student

VALUES('201215129','赵新',20);

- 转换为对基本表的更新:

INSERT

INTO Student(Sno,Sname,Sage,Sdept)

VALUES('200215129 ','赵新',20,'IS');



更新视图 3

- [例3.97]删除信息系学生视图IS_Student中学号为"201215129"的记录

DELETE

FROM IS_Student

WHERE Sno= ' 201215129 ';

- 转换为对基本表的更新:

DELETE

FROM Student

WHERE Sno= ' 201215129 ' AND Sdept= 'IS';



更新视图 4

- 允许对行列子集视图进行更新
- 对其他类型视图的更新不同系统有不同限制
 - 更新视图的限制：一些视图是不可更新的，因为对这些视图的更新不能唯一地有意义地转换成对相应基本表的更新
- 例：例3.89定义的视图S_G为不可更新视图。

```
UPDATE S_G
SET   Gavg=90
WHERE Sno= '201215121';
```
- 这个对视图的更新无法转换成对基本表SC的更新



更新视图 5

- 例：将SC中成绩在平均成绩之上的元组定义成一个视图

```
CREATE VIEW GOOD_SC
```

```
AS
```

```
SELECT Sno,Cno,Grade
```

```
FROM SC
```

```
WHERE Grade >
```

```
(SELECT AVG(Grade)
```

```
FROM SC);
```

- 一个不允许更新的视图上定义的视图也不允许更新



视图的作用-

视图能够简化用户的操作

- 当视图中数据不是直接来自基本表时，定义视图能够简化用户的操作
 - 基于多张表连接形成的视图
 - 基于复杂嵌套查询的视图
 - 含导出属性的视图



视图的作用-

视图使用户能以多种角度看待同一数据

- 视图机制能使不同用户以不同方式看待同一数据,
- 适应数据库共享的需要



视图的作用-

视图对重构数据库提供了一定程度的逻辑独立性

- 例：学生关系Student(Sno,Sname,Ssex,Sage,Sdept)
 - “垂直”地分成两个基本表：
SX(Sno,Sname,Sage)
SY(Sno,Ssex,Sdept)
- 通过建立一个视图Student：

```
CREATE VIEW Student(Sno,Sname,Ssex,Sage,Sdept)
AS
SELECT SX.Sno,SX.Sname,SY.Ssex,SX.Sage,SY.Sdept
FROM SX,SY
WHERE SX.Sno=SY.Sno;
```
- 使用户的外模式保持不变，用户的应用程序通过视图仍然能够查找数据
- 由于对视图的更新是有条件的，因此应用程序中修改数据的语句可能仍会因基本表结构的改变而改变。



视图的作用-

视图能够对机密数据提供安全保护

- 对不同用户定义不同视图，使每个用户只能看到他有权看到的数据



视图的作用-

适当的利用视图可以更清晰的表达查询

- 经常需要执行这样的查询“对每个同学找出他获得最高成绩的课程号”。
- 可以先定义一个视图，求出每个同学获得的最高成绩

```
CREATE VIEW VMGRADE  
AS  
SELECT Sno, MAX(Grade) Mgrade  
FROM SC  
GROUP BY Sno;
```

- 然后用如下的查询语句完成查询：

```
SELECT SC.Sno,Cno  
FROM SC,VMGRADE  
WHERE SC.Sno=VMGRADE.Sno AND  
SC.Grade=VMGRADE .Mgrade;
```

