# 数 据 库 原 理

# Theory of Database

李静

信息科学与技术学院

# 复习

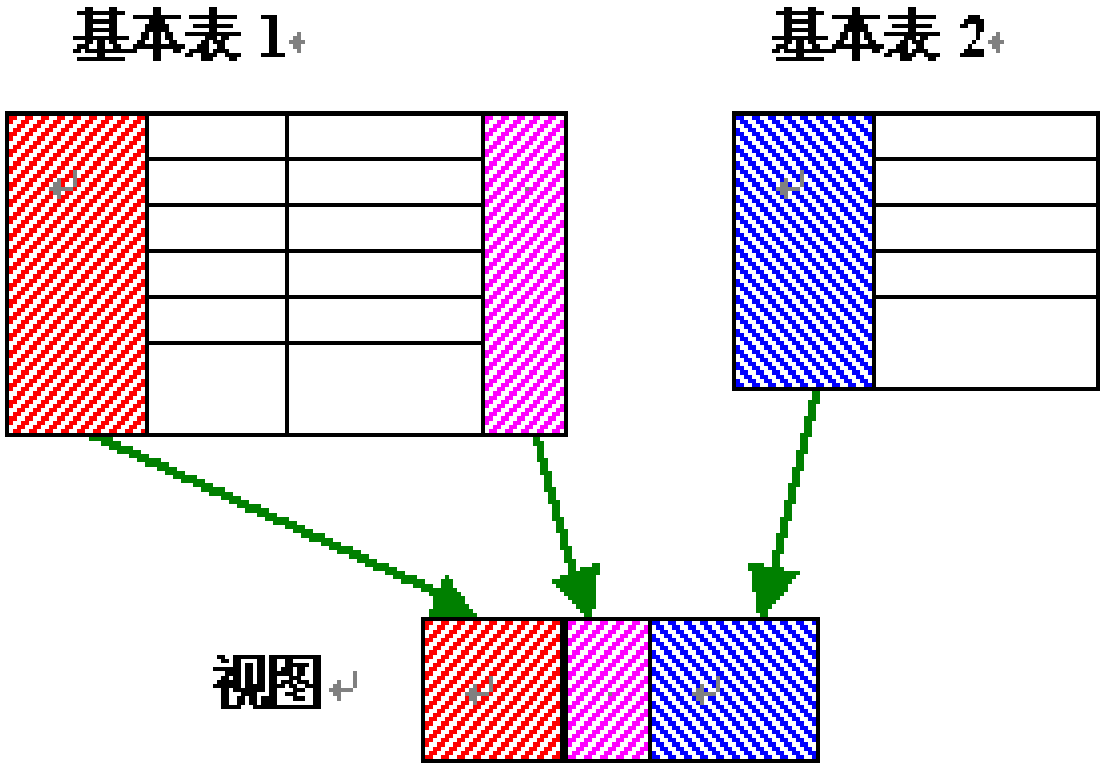
- 子查询（in/比较符/any|all/exists）
- 数据更改功能
  - 插入数据
  - 更新数据
  - 删除数据

# 第5章 视图

- 5.1 视图概念
- 5.2 定义视图
- 5.3 通过视图查询数据
- 5.4 修改和删除视图
- 5.5 视图的作用

# 5.1 视图概念

视图：由基本表构成的虚表（满足用户需求的表结构）



# 5.2 定义视图

**CREATE VIEW <视图名> [(视图列名表)]**

**AS**

**查询语句**

说明：

- 查询中通常不含ORDER BY和DISTINCT语句。
- 缺省时视图列名与查询列名相同。
- 查询的源表可以是已定义的视图。

# 说明

- 下列三种情况下不能省略视图列名：
  - 某个目标列是聚集函数或表达式列；
  - 多表连接查询时，在查询列表中有同名列；
  - 希望用新的更合适的列名。
- 视图的列名序列或者全部省略，或全部指定。

# 定义单源表视图

- 视图取自一个基本表的部分行、列，视图行列与基本表行列对应，
- 一般可看可改。

# 示例

## 例1．建立信息系学生的视图。

```
CREATE  VIEW   IS_Student
   AS
        SELECT  Sno, Sname, Sage
          FROM  Student
         WHERE  Sdept = '信息'
```

定义了视图后，可以和表一样，
使用SELECT语句访问它。

# 示例

例：

```
CREATE  VIEW  MYVIEW2  AS
 SELECT TITLE, ADVANCE,
PRICE * ROYALTY * YTD_SALES  AS
NEWPRICE
FROM  TITLES
WHERE  PRICE > $5
```

# 示例

例：

```
CREATE  VIEW  MYVIEW3  AS
 SELECT *
  FROM  TITLES
 WHERE  TITLE   LIKE  '%SALES%'
```

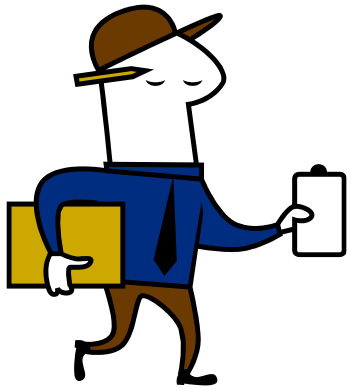
# 示例

例：

```
CREATE  VIEW  MYVIEW4  AS
SELECT TITLE_ID, PUB_ID, TITLE
FROM TITLES
WHERE TITLE LIKE  ‘%SALES%’
```

# 定义多源表视图

- 子查询源表多于一个，
- 一般可看不可改。



# 示例

例2．建立查询信息系选了C01课程的学生的视图，列出学号，姓名和成绩。

```
CREATE VIEW V_IS_S1（Sno, Sname, Grade）
    AS
      SELECT Student.Sno, Sname, Grade
          FROM Student join SC on
                    Student.Sno = SC.Sno
          WHERE Sdept = ‘信息系’
              AND  SC.Cno = ‘C01’
```

# 在已有视图上定义新视图

- 视图的数据源可以来自其它的视图。

# 示例

例3．利用例2建立的视图，建立信息系选了C01课程且成绩在90分以上的学生的视图。

```
CREATE VIEW V_IS_S2（Sno, Sname, Grade）
    AS
        SELECT Sno, Sname, Grade
            FROM V_IS_S1
            WHERE Grade >= 90
```

例2．建立查询信息系选了C01课程的学生的视图，列出学号，姓名和成绩。

# 示例

例4．利用例1所建的视图，建立查询信息系VB考试成绩大于等于80分的学生的姓名和成绩的视图。

```
CREATE VIEW V_IS_VB
AS
   SELECT Sname, Grade FROM  IS_Student   V
     JOIN SC ON V.Sno = SC.Sno
     JOIN Course C ON C.Cno = SC.Cno
     WHERE Cname = 'VB' AND Grade >= 80
```

例1．建立信息系学生的视图。

# 定义带表达式的视图

- 定义基本表时，为减少数据冗余，表中只存放基本数据。
- 由基本数据经过各种计算派生出的数据一般不存储。
- 由于视图中的数据并不实际存储，因此，可以在在视图中设置一些附加列来保存这些派生的数据。
- 由于这些附加列在基本表中并不实际存在，因此称这些列为虚拟列。
- 称包含虚拟列的视图为带表达式的视图。

# 示例

例5．定义一个查询学生学号、姓名和出生年份的视图。

```
CREATE VIEW  V_BirthYear
    (Sno, Sname, BirthYear)
AS
   SELECT Sno, Sname, 2020-Sage
      FROM Student
```

# 含分组统计信息的视图

- 子查询中含GROUP BY子句，视图行列由基本表行列得到，
- 数据只看不可改。

# 示例

例6．定义一个查询每个学生的学号及考试平均成绩的视图

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

# 5.3 通过视图查询数据

- 视图定义好后，可以对其进行查询，
- 通过视图查询数据同基本表一样。

# 示例

例7．利用5.2节例1建立的视图，查询信息系年龄小于等于20岁的学生。

```
SELECT Sno, Sname, Sage  FROM  IS_Student
  WHERE Sage <= 20
```

转换成相关基本表的等价查询

```
SELECT Sno, Sname, Sage FROM Student
        WHERE Sdept = '信息系'
        AND Sage <= 20
```

例1．建立信息系学生的视图。

# 示例

- 例8．查询信息系选修了“C01”的学生学号、姓名和年龄。

```
SELECT sc.Sno, Sname, Sage
FROM  IS_Student  JOIN  SC
    ON IS_Student.Sno = SC.Sno
WHERE Cno = 'C01'
```

# 示例

例9. 查询信息系学生的学号、姓名、所选课程的课程名。

```
SELECT v.Sno,Sname,Cname FROM  IS_Student v
   JOIN SC ON v.Sno = SC.Sno
   JOIN Course C ON C.Cno = SC.Cno
```

转换成相关基本表的等价查询

```
SELECT S.Sno,Sname,Cname FROM Student S
        JOIN SC ON S.Sno = SC.Sno
    JOIN Course C ON C.Cno = SC.Cno
        WHERE Sdept = '信息系'
```

# 示例

例10.利用5.2节例6建立的视图，查询考试平均成绩80分以上的学生的学号和平均成绩。

```
SELECT * FROM S_G  WHERE  AvgGrade > 80
```

不能直接转换为：

```
SELECT Sno, AVG(Grade) FROM SC
    WHERE  AVG(Grade) > 80
        GROUP BY Sno
```

而应该转换为：

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

例6. 定义一个查询每个学生学号及考试平均成绩的视图

# 修改视图

- 格式：

```
ALTER  VIEW  视图名
  [ ( 列名 [ ,...n ] ) ]
AS
  查询语句
```

# 示例

例11．修改例6定义的视图，使其统计每个学生的考试平均成绩和修课总门数。

```
ALTER VIEW S_G(Sno,AvgGrade,Count_Cno)
AS
  SELECT Sno, AVG(Grade), Count(*)
    FROM SC
   GROUP BY Sno
```

例6．定义一个查询每个学生学号及考试平均成绩的视图

# 更新视图

- 更新视图指通过视图来插入、删除和修改基本表中的数据。
- 视图不实际存放数据，对视图的更新，最终要转换为对基本表的更新。
  - 如果视图的定义中包含了表达式，或聚合运算，或消除重复值运算，则不能对视图进行更新操作。
- 对视图进行更新操作，其限制条件比较多
  - 建立视图的作用不是利用视图来更新数据库中的数据，而是简化用户的查询；
  - 达到一定程度的安全性保护；
  - 尽量不要对视图执行更新操作。

# 更新视图

例：在StudentView1991中，将学号为'0800004'同学的名字修改为'张小立'。

```
UPDATE StudentView1991
SET studentName='张小立'
WHERE studentNo='0800004‘
```

操作过程：

- 系统首先进行有效性检查，判断视图StudentView1991是否存在;
- 如果存在，则从系统的数据字典中取出该视图的定义；
- 将定义中的子查询与用户的查询结合起来，转换为基于基本表的修改：

```
UPDATE Student
SET studentName='张小立'
WHERE year(birthday)=1991 AND studentNo='0800004'
```

# 更新视图

例：在视图StudentView1991中将学号为'0800006'的同学记录删除。

```
DELETE FROM StudentView1991
WHERE studentNo='0800006'
```

- 系统将该操作转化为如下的操作：

```
DELETE FROM Student
WHERE year(birthday)=1991 AND
  studentNo='0800006'
```

# 更新视图

例：在视图SourceView中删除平均成绩大于80分的课程记录。

```
DELETE FROM SourceView
WHERE courseAvg>=80
```

- 该操作数据库管理系统拒绝执行
  - 因为视图SourceView包含了聚合运算，系统无法将该视图转化为对基本表的操作。
- 一般来讲，如果是行列子集视图，则可以对该视图进行更新操作；其它类型的视图，具体的数据库系统有具体的定义，一般不对其进行更新操作。

# 删除视图

- 格式：

DROP VIEW <视图名>

例：删除例1定义的IS_Student视图。

DROP VIEW IS_Student

# 5.5 视图的作用

- 简化数据查询语句
- 使用户能从多角度看待同一数据
- 提高了数据的安全性
- 提供了一定程度的逻辑独立性

# 作业

📖 P93 3题~6题