

学生（学号、姓名、性别、院系）

课程（课程号、课程名）

学习（学号、课程号、成绩）

讲授（课程号、教师号、教师姓名）

- 1 找出计算机系女同学名单
- 2 求选修课程号为2的学生名单
- 3 求选修数据库课程的学生姓名及院系
- 4 求选修了张宁老师所授全部课程的学生名单
- 5 求选修了张宁老师所授课程的学生名单
- 6 没有选修任何课程的学生名单及院系
- 7 没有被任何人选修的课程名

3.4 数据更新

一、插入数据

1 插入单个元组

INSERT

INTO <表名> [(<属性列1>[, <属性列2>...])]

VALUES (<常量1>[, ,<常量2>]...);

2 插入子查询结果

INSERT

INTO <表名> [(<属性列1>[, <属性列2>...])]

子查询;

```
INSERT
```

```
INTO Student (Sno,Sname,Ssex,Sage,Sdept)
```

```
VALUES('4','王磊' , '男' ,19,'计算机' );
```

指定列名

```
INSERT
```

```
INTO Student
```

```
VALUES('4','王磊' , '男' ,19,'计算机' );
```

不指定
列名

对每一个系，求学生的平均年龄，并存入数据库

```
CREATE TABLE Dept_age  
(Sdept CHAR(15),  
Avg_age SMALLINT);
```

建立新表

```
INSERT  
INTO Dept_age (Sdept, Avg_age)  
SELECT Sdept,AVG(Sage)  
FROM Student  
GROUP BY Sdept;
```

插入数据

二、删除数据

删除语句的一般格式:

DELETE

FROM <表名>

[WHERE <条件>];

DELETE语句删除表中的数据，
但表的定义仍然留在数据字典中

删除学号为1号的学生记录

```
DELETE FROM Student  
WHERE Sno='1';
```

删除单个
元组

删除计算机学院所有学生的选课记录

```
DELETE FROM SC  
WHERE Sno IN  
(SELECT Sno  
FROM Student  
WHERE Sdept='计算机' );
```

带子查询
的修改语
句

三、修改数据

修改操作语句的一般格式：

UPDATE <表名>

SET <列名>=<表达式> [, <列名>=<表达式>]
>]...

[**WHERE** <条件>];

将1号学生的年龄改为22岁

```
UPDATE Student  
SET Sage=22  
WHERE Sno='1';
```

修改单个
元组

将所有学生的年龄都增加1岁

```
UPDATE Student  
SET Sage=Sage+1;
```

修改多个
元组

将计算机学院全体学生的成绩置零

```
UPDATE SC  
SET Grade=0  
WHERE Sno IN  
  (SELECT Sno  
   FROM Student  
   WHERE Sdept='计算机' );
```

带子查询
的修改语
句

3.5 视图

视图： 从一个或几个基本表（或视图）导出的表（虚表）。

- 在数据库中只存放视图的定义，而不存放视图的数据；
- 视图一经定义，就可以和表一样被操作（查询和更新）。也可以在视图上再定义新的视图，但对视图的更新操作有一定得限制。

一、定义视图

1、建立视图

```
CREATE VIEW    < 视图名>[( <列名>[, <
列名>]...)]
```

```
AS    <子查询>
```

```
[WITH CHECK OPTION];
```

2、删除视图

```
DROP VIEW    <视图名>[CASCADE]
```

- 组成视图的属性列名或者全部省略，或者全部指定，没有第三种选择。
- 在下列情况下必须列出视图的属性：
 - 某个列不是单纯的属性名，而是聚集函数或表达式；
 - 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名字。

建立计算机学院学生的视图.

```
CREATE VIEW CS_VIEW  
AS  
SELECT Sno,Sname,Sdept  
FROM Student  
WHERE Sdept='计算机'  
  
WITH CHECK OPTION;
```

若视图从一个基本表导出，并且只是去掉了基本表中的部分行和列，但保留了原表的主码，称这类视图为“行列子视图”

建立计算机学院选修了1号课程且成绩在90分以上的学生视图，包含属性：学生编号、姓名和成绩。

```
CREATE VIEW CS_S1(Sno,Sname,Grade)
AS
SELECT Student.Sno,Sname,Grade
FROM Student,SC
WHERE Sdept='计算机'
AND Student.Sno=SC.Sno
AND Cno='1'
AND Grade>=90
```

视图不仅可以建立在单个表上，而且可以建立在多个表上。

建立计算机学院选修了1号课程且成绩在90分以上的学生视图，包含属性：学生编号、姓名和成绩..

Step1 :建立计算机学院选修1号课程的学生成绩视图。

```
CREATE VIEW CS_S21(Sno,Sname,Grade)
AS
SELECT Student.Sno,Sname,Grade
FROM Student,SC
WHERE Sdept='计算机'
AND Student.Sno=SC.Sno
AND Cno='1'
```

视图还可以建立在一个或多个已定义好的视图上，或建立在基本表与视图上。

Step2 :在视图CS_S21的基础上建立查询结果。

```
CREATE VIEW CS_S2  
AS  
SELECT Sno,Sname,Grade  
FROM CS_S21  
WHERE Grade>=90
```


定义一个反映学生出生年份的视图.

```
CREATE VIEW BT_S(Sno,Sname,Sbirth)
AS
SELECT Sno,Sname,2021-Sage
FROM Student
```

派生属性在基本表中并不实际存在，称为虚拟列，带虚拟列的视图称为
“带表达式的视图”

三、查询视图

查询计算机学院的视图中年龄小于20岁的学生.

```
SELECT Sno,Sage  
FROM CS_VIEW  
WHERE Sage<20
```

视图消解

(View Resolution)



```
SELECT Sno,Sage  
FROM Student  
WHERE Sage<20  
AND Sdept='计算机'
```

四、更新视图

更新视图是指通过视图来插入、删除和修改数据

将计算机学院学生视图CS_VIEW中学号为1号的学生姓名改为‘刘晨’。

```
UPDATE CS_VIEW  
SET Sname='刘晨'  
WHERE Sno='1'
```

不可更新的视图

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

此视图是不可更新的！

建立视图的优点

- 1、简化用户的操作
- 2、视图使用户能够以多种角度看待同一数据
- 3、视图对重构数据库提供了一定程度的逻辑独立性
- 4、视图能够对机密数据提供安全保护
- 5、适当地利用视图可以更清晰地表达查询

对每个学生查出他获得最高成绩的课程号.

不用视图怎么写?



Step1 :将学习表中的记录按照学生进行分组，统计每个学生的最高成绩，并生成视图。

```
CREATE VIEW MAXGrade
```

```
AS
```

```
SELECT Sno,MAX(Grade) AS Mgrade
```

```
FROM SC
```

```
GROUP BY Sno
```

Step2 :将试图MAXGrade与SC表连接，获得查询结果。

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

3.6 数据控制

授予权限语句

该语句基本形式为：

GRANT 〈权限〉 ON 〈数据对象〉 TO

〈授权者〉 [WITH GRANT OPTION]

〈数据对象〉 ::= 〈基表〉 | 〈视图〉 | 〈属性〉 | ...

〈授权者〉 ::= PUBLIC | 〈授权ID〉

其中PUBLIC是所有数据库用户的总称，若有WITH GRANT OPTION，则授权者可将此特权转授给其它〈授权者〉。

例如， 下列语句：

1) GRANT SELECT ON 表1 TO PUBLIC;

将对表1的SELECT特权授予所有的用户。

2) GRANT SELECT (列1, 列5) ON 表1 TO sam;

将对表1中1和5列的select特权授予sam。

3) GRANT UPDATE (列2, 列4) ON 表1 TO wang
WITH GRANT OPTION;

将对表1中2和4列的UPDATE特权授予wang， 并允许他将此特权转授给其它用户。

收回权限语句

```
REVOKE  〈权限〉 ON  〈数据对象〉 FROM  〈受权者〉  
      [ {,  〈受权者〉 } ]
```

REVOKE UPDATE ON 表1 FROM wang CASCADE 。

将对表1的UPDATE权限从wang处收回。