**定义视图**

在EDUC数据库中，以Student、Course 和Student\_course 表为基础完成一下视图定义

1. 定义计算机系学生基本情况视图V\_Computer；

create view V\_Computer(Sno,Sname,Sage,Sdept,birthday)

as

select Sno,Sname,Sage,Sdept,birthday

from Students

where Sdept = 'CS';

1. 将Student、 Course 和Student\_course表中学生的学号，姓名，课程号，课程名，成绩定义为视图V\_S\_C\_G

create view V\_S\_C\_G(Sno,Sname,Cno,Cname,Grade)

as

select Students.Sno,Sname,Courses.Cno,Cname,Grade

from Students,Courses,student\_course

where Students.Sno = student\_course.Sno and

student\_course.Cno = Courses.Cno

1. 将各系学生人数，平均年龄定义为视图V\_NUM\_AVG

create view V\_NUM\_AVG(Sdept,avg\_age)

as

select Sdept,AVG(Sage)

from students

group by Sdept

1. 定义一个反映学生出生年份的视图V\_YEAR

create view V\_YEAR(Sname,year)

as

select Sname,birthday

from Students

1. 将各位学生选修课程的门数及平均成绩定义为视图V\_AVG\_S\_G

create view V\_AVG\_S\_G(Sno,grade\_num,grade\_avg)

as

select Student\_course.Sno,COUNT(Student\_course.Sno),AVG(grade)

from students,student\_course

where Students.Sno = Student\_course.Sno

group by Student\_course.Sno

1. 将各门课程的选修人数及平均成绩定义为视图V\_AVG\_C\_G

create view V\_AVG\_C\_G(Cname,Cnum,Avg\_grade)

as

select Cname,COUNT(Cname),AVG(grade)

from courses,student\_course

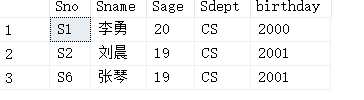
where courses.Cno = student\_course.Cno

group by Cname

**使用视图**

1. 查询以上所建的视图结果。

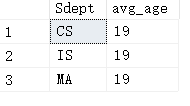
查询V\_Computer



查询V\_S\_C\_G



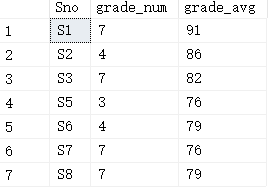
查询V\_NUM\_AVG



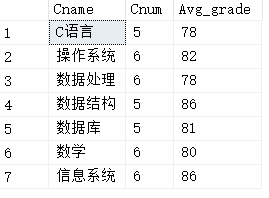
查询V\_YEAR



查询V\_AVG\_S\_G



查询V\_AVG\_C\_G



1. 查询平均成绩为90分以上的学生学号、姓名和成绩；

select Students.Sno,Sname,grade\_avg

from Students,V\_AVG\_S\_G

where Students.Sno = V\_AVG\_S\_G.Sno

and grade\_avg > 90



1. 查询各科成绩均大于平均成绩的学生学号、姓名、课程和成绩；

先建立一个表存储每个同学对于每个科目平均分的情况

use EDUC\_1

create table T\_S\_C\_G\_Avg(

Sno char(9),

Cno char(4),

Grade smallint,

GradeAvg int

)

insert into T\_S\_C\_G\_Avg(Sno,Cno,Grade,GradeAvg)

select Sno,Courses.Cno,Grade,V\_AVG\_C\_G.Avg\_grade

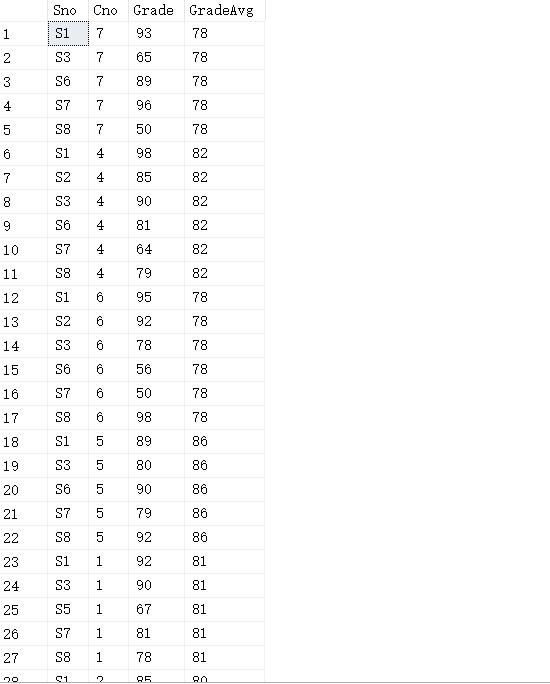
from (student\_course right join Courses

on student\_course.Cno = Courses.Cno)

right join V\_AVG\_C\_G

on V\_AVG\_C\_G.Cname = Courses.Cname

order by student\_course.Sno,Courses.Cno,V\_AVG\_C\_G.Avg\_grade asc



select distinct Sno

from T\_S\_C\_G\_Avg

where Sno not in(

select distinct Sno

from T\_S\_C\_G\_Avg

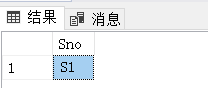
where Grade <= GradeAvg)

子查询查询学生成绩小于科目平均分

如果满足这个条件，说明，这个学生有至少一门成绩低于平均分数。

则只要学生的学号不在子查询结果范围内则满足条件。

查询结果：



1. 按系统计各系平均成绩在80分以上的人数，结果按降序排列；

select Students.Sdept,count(Students.Sdept)as num

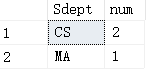
from V\_AVG\_S\_G,Students

where V\_AVG\_S\_G.Sno = Students.Sno and

V\_AVG\_S\_G.grade\_avg >80

group by Students.Sdept

order by num desc



**修改视图**

1. 通过视图V\_Computer，分别将学号为“S1”和“S4”的学生姓名更改为“S1\_MMM”,”S4\_MMM” 并查询结果;

update V\_Computer

set Sno = 'S1\_MMM'

where Sno ='S1';

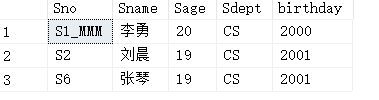
update V\_Computer

set Sno = 'S4\_MMM'

where Sno ='S4';

因为视图V\_Computer没有Sno为S4

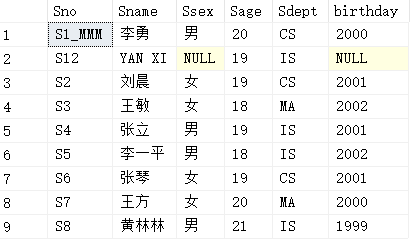
所以只更新了一条数据



1. 通过视图V\_Computer，新增加一个学生记录 ('S12','YAN XI',19,'IS')，并查询结果

因为插入的学生记录不是计算机系的学生，在视图V\_Computer中不显示，但是会同步更新数据到表students中。

Insert into V\_Computer (Sno,Sname,Sage,Sdept) values ('S12','YAN XI',19,'IS')

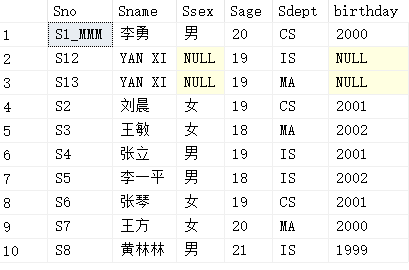


1. 通过视图V\_Computer，新增加一个学生记录 ('S13','YAN XI',19,'MA')，并查询结果

因为插入的学生记录不是计算机系的学生，在视图V\_Computer中不显示，但是会同步更新数据到表students中。

insert into V\_Computer(Sno,Sname,Sage,Sdept)

values('S13','YAN XI',19,'MA')



1. 通过视图V\_Computer，删除学号为“S12”和“S3”的学生信息，并查询结果

视图中不包含学号为“S12”和“S3”的学生，

所以对原表students没有影响。

1. 要通过视图V\_S\_C\_G，将学号为“S12”的姓名改为“S12\_MMM”，是否可以实现？并说明原因

不能实现。

因为插入S12学生的记录时，没有相应的更新学生的成绩，所以表V\_S\_C\_G不存在该学生的记录，所以不能修改姓名。

1. 要通过视图V\_AVG\_S\_G，将学号为“S1”的平均成绩改为90分，是否可以实现？并说明原因

不能实现。

* 1. 学生的平均成绩是根据查询语句中的avg求的平均值，直接修改会影响数据的一致性。
  2. 在第一问时，将S1修改为S1\_MMM,相应的在视图V\_AVG\_S\_G中带有S1的记录消失，无法改正。

**思考**

如何通过视图实现程序的逻辑独立性。

答：1.尽可能的减少表与表之间的连接，减少联系。

2.如果想要更新一个表的内容，而会影响另一个表的数据，那么最好建立外键约束。

3.在建立视图时，最好不要使用约束条件。不要使用order by，因为原本表数据是无序的。

**使用索引**

1. 分别建立以下索引(如果不能成功建立,请分析原因)

(1) 在student表的sname列上建立普通降序索引。

create index S\_idx

on students(Sname desc)

(2) 在course表的cname列上建立唯一索引。

create unique index C\_idx

on Courses(Cname)

(3) 在student\_course表的sno列上建立聚集索引。

create clustered index S\_idx

on student\_course(Sno)

(4) 在student\_course表的sno(升序), tid(升序)和score(降序)三列上建立一个普通索引。

create index S\_T\_S\_idx

on student\_course(Sno asc,Cno asc,grade desc)

2. 删除索引

将student\_course表的sno列上的聚集索引删掉。

drop index student\_course.S\_idx