1.查找学号为 00002 没有选修的科目

select Cname from course  where

                          not exists               //找不到的组合，提交course

                               (select \* from SC where course.cno = cno and sno = ''00002'')

3. 选修了全部课程的学生姓名

select Sname from student

                           where NOT exists        //

                              (select  \* from course

                               where NOT exists      //不存在的提交给course

                                (select \* from SC where

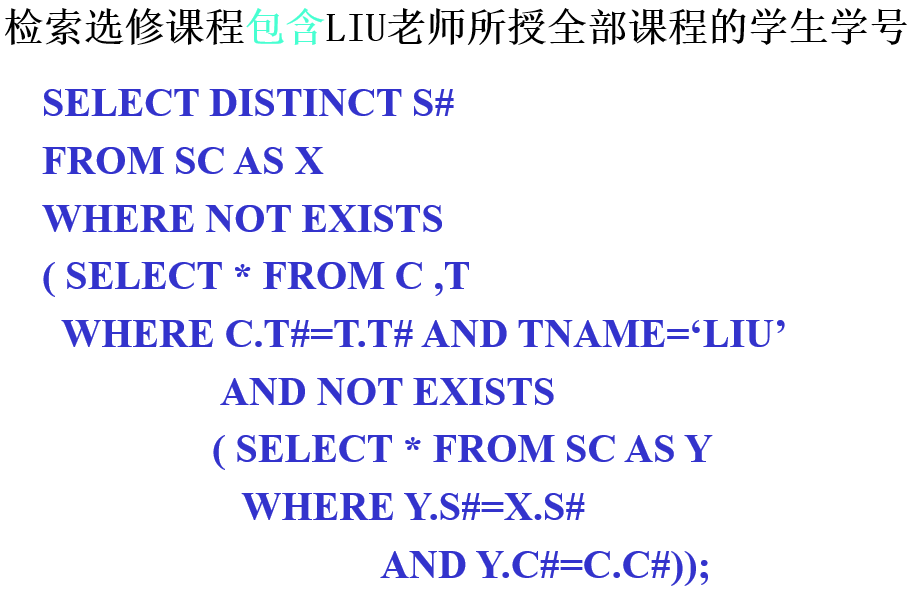
                                    Sno = student.sno  and cno = Course.Cno))   // 代入两个未知变量

select student.\* from student

where s\_id in

(select s\_id from score group by s\_id having count(c\_id)=(select count(distinct c\_id) from course));

4. 查询至少选修了学生95002选修的全部课程的学生名单。P111



5. 查询和01号同学学习的课程完全相同的其他同学的信息

select \* from student

where s\_id in(

select distinct s\_id from score where s\_id not in

(select s\_id from score where c\_id not in (select c\_id from score where s\_id='01'))

group by s\_id

having count(c\_id)=(select count(c\_id) from score where s\_id='01') and s\_id<>'01')

6.查询至少有一门课与学号为“01”的学生所学课程相同的学生的学号和姓名

select distinct student.s\_id, s\_name from student,score

where student.s\_id=score.s\_id

and score.c\_id in

(select c\_id from score where s\_id='01')//01学生的所有课

and student.s\_id!='01';

7. 查询平均成绩最高的同学的姓名

找出平均成绩大于等于所有人平均成绩的同学即可

SELECT student.sname, avg(sc.grade) AS avg\_grade

FROM student, sc

WHERE student.sno=sc.sno

GROUP BY student.sname

HAVING avg(sc.grade) >= all(

SELECT avg(grade) as avg

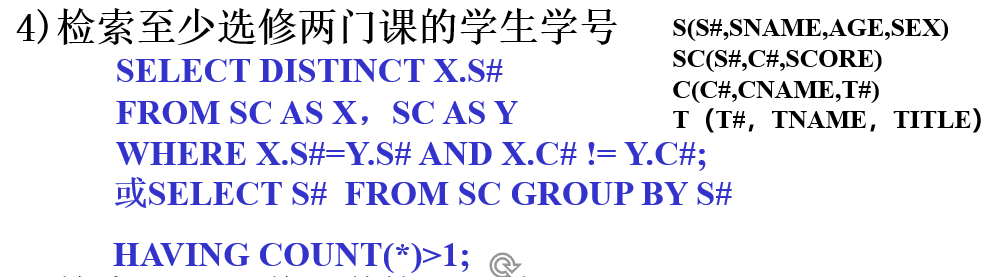
FROM sc

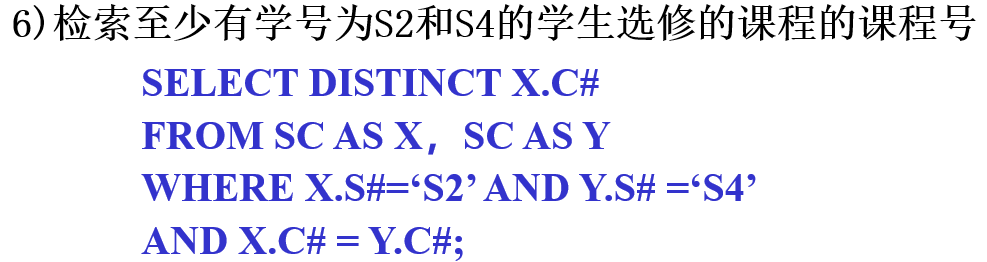
WHERE grade NOTNULL

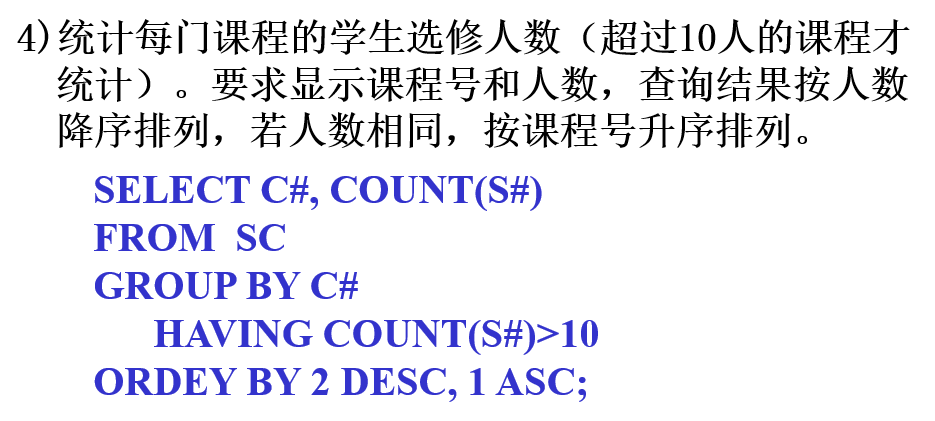
GROUP BY sc.sno);

8. 查出分数最高的前20位学生









**第六章**

**2NF还有什么问题？**

* + 一般地，如果把1NF关系模式 通过投影分解方法，消除非主属性对码的部分函数依赖，分解为多个2NF的关系模式。
  + 可以在一定程度上减轻 原1NF关系模式中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
  + 但是还不能完全消除关系模式中的各种异常情况和数据冗余

**3NF的一些性质**

* + **若R∈3NF，则R的每一个非主属性既不部分函数依赖于候选码也不传递函数依赖于候选码。**
  + **采用投影分解法将一个2NF的关系分解为多个3NF的关系，可以在一定程度上解决原2NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。并不能完全消除关系模式中的各种异常情况和数据冗余。**
  + **3NF等价定义：设F是关系模式R的FD集，如果对F中每个非平凡的FD X→Y，都有X是R的超键，或者Y的每个属性都是主属性，则称R是3NF的模式。**

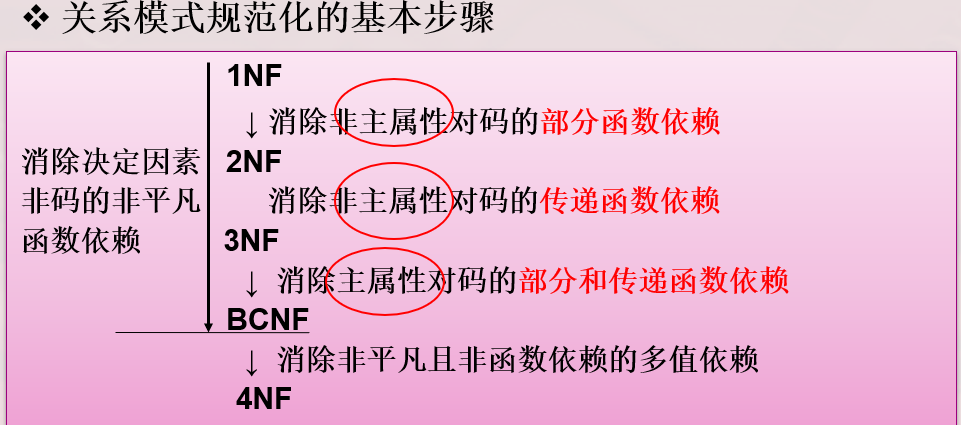
**BCNF的关系模式所具有的性质**

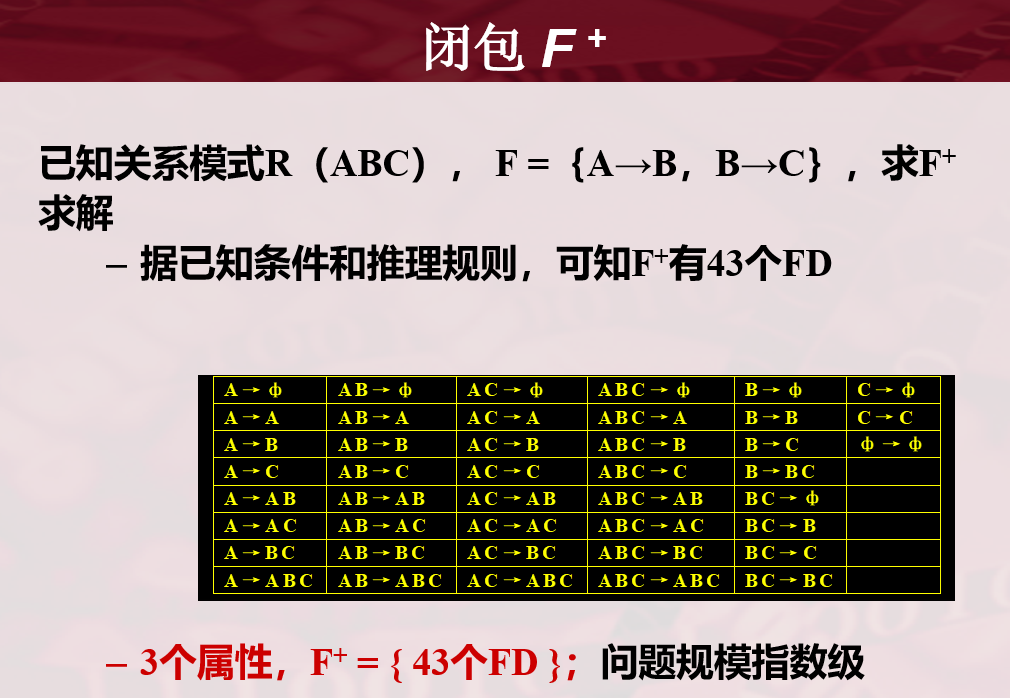
**⒈所有非主属性对每一个码都是完全函数依赖。**

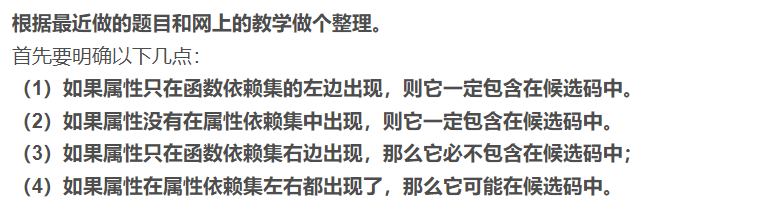
**⒉所有主属性对每一个不包含它的码也是完全函数依赖。**

**3.没有任何属性完全函数依赖于非码的任何一组属性。.**

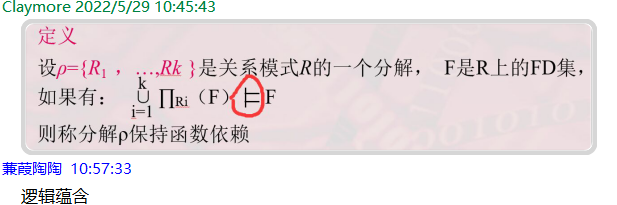
* **如果关系模式R∈BCNF，必定有R∈3NF**
* **如果关系模式R∈3NF，不一定有 R∈BCNF**
* **如果一个关系数据库中的所有关系模式都属于BCNF，那么在函数依赖范畴内，它已实现了模式的彻底分解，达到了最高的规范化程度，消除了操作异常诸多问题。**







候选码不一定只有一个

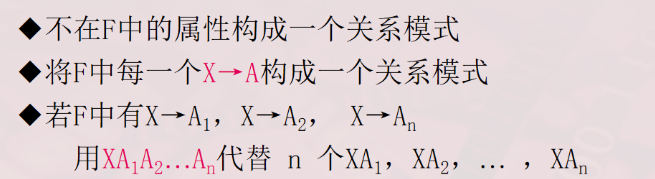


如果一个分解具有无损连接性，则它能够保证不丢失信息。

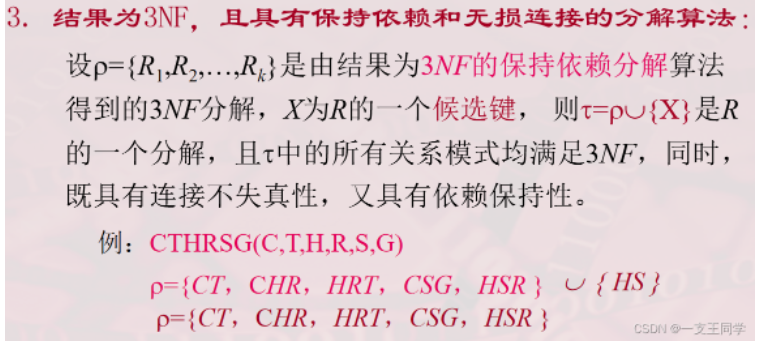
如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况。



算法6.3首先对F极小化处理

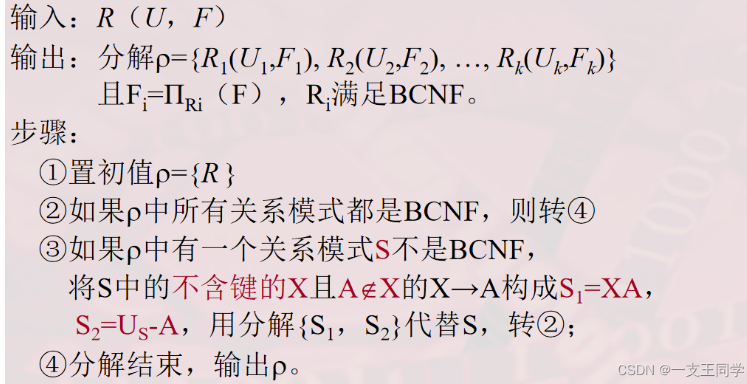


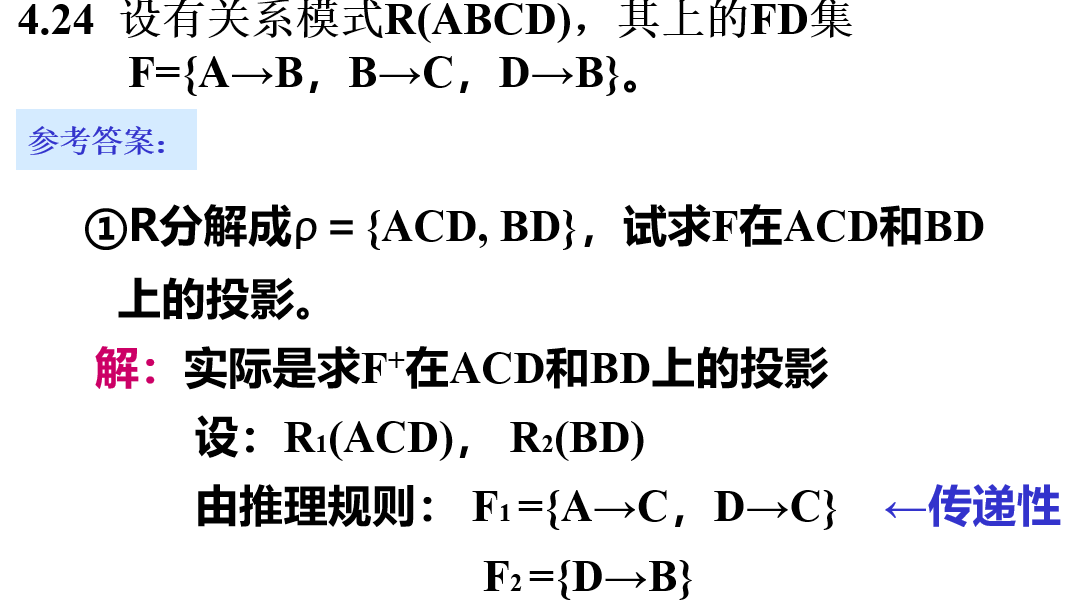
算法6.4



 就是在 “算法6.3” 的基础上，多 ∪ 一个候选码。图中因为 “HSR” 包含了 “HS”，所以 并 之前和 并 之后结果一样。

算法6.5





投影定义p194

**第七章**

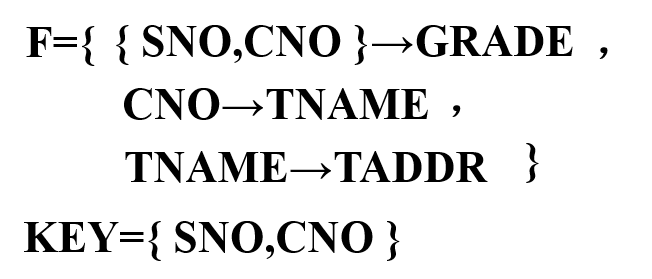
**根据需求设计数据库：题型1画出ER图，设计概念模式；题型2：写出基本函数依赖集和候选键**

**设有关系模式R(SNO, CNO, GRADE, TNAME, TADDR)，其属性分别表示学生学号、课程号、成绩、任课教师名、教师地址。规定：每个学生一门课只有一个成绩；每门课只有一个教师任教；每个教师只有一个地址，所有教师不同名。**

**(1) 试写出关系模式R基本的函数依赖和候选键。**

**(2) 试把R分解成2NF模式集，并说明理由。**

**(3) 试把R分解成3NF模式集，并说明理由。**



**第10章**

**(1) 检查点机制的主要思想：**

**\*运行过程中由DBMS每隔一定时间在日志中设置一个检查点**

**\*在检查点时刻，**

**－ 把将上一个检查点后已commit的事务对DB的更新**

**写入磁盘**

**－ 在日志文件中登入<checkpoint>记录**

**－ 故障恢复时，只有最近一个检查点之后到发生故障时**

**的事务需要恢复**

**(2) COMMIT与检查点时刻操作的协调：**

**－在检查点时刻，写入磁盘的是上一个检查点后已**

**commit的事务对DB的更新**

**－ 故障恢复时，重做的是从最近一个检查点之后到**

**发生故障时已commit的事务，这些事务被正向扫描写入**

**重做队列，并重新实施更新操作**