数据库复习

一、第3章 关系数据库标准语言SQL

1.数据定义

```
drop table Student;
create table Student(
    Sno char(5), -- 自动建立聚集索引
    Sname nvarchar(5) ,
    Sage tinyint ,
    Sdept nvarchar(20)
    )
    create  index sagsdept on Student(sdept)
    小测  创建成绩表（学号、课程号、成绩）
    建立学号聚集索引  成绩普通索引

    alter table Student add Ssex varchar(4)

    pRiNt 123*456

    数据类型   大数据存储  编程类型多了一个长度 String

    字符  char(n) 固定长度  varchar(n) 可变长度 text 长文本
          nchar(n)          nvarchar            ntext
          char > nchar > varchar > nvarchar
          长度n 尽可能小       性能 安全


    数字   int    tinyint 1字节 255  smallint 2字节 32676
                   int 4字节   bigint 8字节
           float  近似

    日期 datetime

    图片声音... 二级制流 存储成文件，将文件名存到数据库

    primary key  主键   1、不能为空  2、不能重复
                 char *int  nchar
    unique   唯一值
```

2.数据查询

```
select from where and or not
-- 代表注释
字符串 ''
* 表示所有列
表达式  as 别名
连接 join on
存在 exists(select * from  where )
剔除重复 distinct  整行
在集合(集合包含) in ()
```

模糊匹配 like % 任意长度字符 _ 单个字符 escape 转义符 短文本
长文本 全文索引 倒排索引 词是有限的，先将所有的词建立索引表 ，将包含词数据建立到索引表
空值 null is null is not null
排序 order by 列 asc/desc ，物理存储 聚集索引进行排序
取前n个记录 top (n) mysql limit n
全球唯一码 newid()
集函数 count(distinct 列/ *) 计数 sum(数值列) 汇总 avg(数值列) 平均值 max()最大值 min() 最小值
分组 group by 列 分组筛选 having
多表连接查询 笛卡尔积+选择 join on 左连接 left join 右连接 right join
嵌套查询 select from where group by having order by 嵌套在另一个查询块

练习 查询 学生的学号、姓名、学习榜样的姓名(本系平均成绩最高同学)

查询平均成绩最高的学生姓名

```
select Sname from Student where Sno in(
select top 1 Sno from SC group by Sno order by avg(Grade))
```

查询选修了2号课程学生的姓名

```
select Sname from Student join SC on Student.Sno=SC.Sno where Cno='2'
select Sname from Student where Sno in
(select Sno from SC where Cno='2')
```

查询选修了全部课程的学生信息

```
select * from Student where not exists(
select * from Course where not exists(select * from SC where Sno=Student.Sno and
Cno=Course.Cno))
```

查询未选修2号课程学生的姓名

```
select Sname from Student where Sno not in(select Sno from SC where Cno='2')

select Sname from Student join SC on Student.Sno=SC.Sno where Cno !='2'
```

查询学生 姓名、课程名、成绩

```
select (select Sname from Student where Sno=SC.Sno) as Sname,(select Cname from
Course where Cno=SC.Cno) as Cname,
Grade from SC
```

查询 选修了 以 数据库 为先行课的学生姓名、年龄、成绩、课程名

```
select Sname,Sage,Grade,c.Cname from Student as s join SC on s.Sno=Sc.Sno join
Course as c on SC.Cno=c.Cno join
Course as c1 on c.Cpno=c1.Cno where c1.Cname = '数据库'
```

查询课程信息 课程号、课程名、学分、先行课程名

```
select Cno,Cname,Ccredit,(select Cname from Course where Cno=c.Cpno) as Cpname
from Course as c


select c1.Cno,c1.Cname,c1.Ccredit,c2.Cname as CpName,c2.Ccredit as Cpcredit
 from Course as c1 left join Course as c2 on c1.Cpno=c2.Cno
```

查询所有课程被选情况 包括课程号，成绩，课程名，学分

```
select c.Cno,SC.Sno,SC.Grade,c.Cname,c.Ccredit from SC right join Course as c on
SC.Cno=c.Cno
```

查询学生姓名 课程名 和 成绩 成绩为优秀的

查询 Student 和 SC 的 笛卡尔积

查询学生信息 按照平均成绩排序

```
select * from Student order by (select avg(Grade) from SC where Sno=Student.Sno)
desc
```

查询IS系学生的学号和评价分

```
select Sno,avg(Grade)
from SC
group by Sno
having Sno in(select Sno from Student where Sdept='IS')
```

统计每个选课数量的学生人数

```
select CCount,count(*) from (
select (select count(*) from SC where Sno=Student.Sno) as CCount,Sno from
Student) as t
group by CCount

select * from
(
select top 2 * from
(select top 4 * from Student order by Sage desc) as t
order by Sage asc) as t1 order by SAge desc

select * from Student order by Sage Desc
```

查询学生的姓名、年龄、优秀门数、挂科门数、平均成绩

```
select Sname,Sage,(select count(*) from SC where Sno=Student.Sno and Grade>=90),
(select avg(Grade) from SC where Sno=Student.Sno)
from Student
```

查询所有学生的姓名和选课情况(课程号、成绩)

```
select s.Sno,Sname,Ssex,Cno,Grade from Student as s,SC where s.Sno = SC.Sno
select s.Sno,Sname,Ssex,Cno,Grade from Student as s left join SC on s.Sno=SC.Sno


select Sname,Cname,Grade from Student as s join SC on s.Sno = SC.Sno join Course
as c on SC.Cno=c.Cno
where Grade >=90
```

查询学生的姓名 课程号 成绩

```
select Sname,Cname,Grade from Student as s,SC,Course as c where s.Sno=SC.Sno and
SC.Cno=c.Cno  and Grade >=90
```

练习 查询保研同学的信息 没有挂科(<60) 按照必修课平均成绩前15%的同学
按照年龄排序 取50% 的 同学

```
select * from Student where Sno in(
select top((select count(*) from Student) * 50 / 100 ) Sno from SC -- 取50%
where Cno in (select Cno from Course where Ccredit>=4)  -- 必修课
and Sno not in(select Sno from SC where Grade <60) -- 没挂科
group by Sno
order by Avg(Grade))

select * from Student where Sno in(
```

```
select top((select count(*) from Student) * 50 / 100 ) Sno from SC -- 取50%
where Cno in (select Cno from Course where Ccredit>=4)  -- 必修课
group by Sno
having Sno not in(select Sno from SC where Grade <60) -- 没挂科
order by Avg(Grade))

select * from Student where Sno in(
select top((select count(*) from Student) * 50 / 100 ) Sno from SC -- 取50%
where Cno in (select Cno from Course where Ccredit>=4)  -- 必修课
group by Sno
order by Avg(Grade))
and Sno not in(select Sno from SC where Grade <60) -- 没挂科

order by Sage desc
(select count(*) from Student) * 50 / 100
```

查询选修必修课程数超过2门的同学的所有课程的平均分

```
select Sno,avg(Grade) from SC
where Sno in(
    select Sno from SC where Cno in(select Cno from Course where Ccredit>=4)
    group by Sno having count(*) >=2)
group by Sno
order by avg(Grade) desc
```

查询年龄大 一半 的同学

```
select top (select (count(*)/3) from Student) * from Student
order by Sage desc
```

```
select Sno,avg(Grade) from SC
where Cno in(select Cno from Course where Ccredit >=4)
group by Sno
having count(*) >=2
```

查询每个学生的学号和平均成绩 结果按照成绩降序排列

```
select top 2 Sno,avg(Grade) from SC
group by Sno
order by avg(Grade) desc

select Cno,avg(Grade),max(Grade),min(Grade),count(*) from SC
group by Cno
```

查询每个学生的学号和必修课平均成绩 结果按照成绩降序排列

```
select top 2 Sno,avg(Grade) from SC
where Cno in(select Cno from Course where Ccredit >=4)
group by Sno
order by avg(Grade) desc
```

查询有2门以上课程优秀的学生信息

```
select * from Student where Sno in(
select Sno from SC where Grade >=90 group by Sno having count(*) >=2)
```

查询男生人数大于2的系集男生人数

```
select Sdept,count(*) from Student
where Ssex='男'
```

```
group by Sdept
having count(*)>=2

查询学生的人数
select count(*) from Student
查询CS系学生的人数
select count(*) from Student where Sdept='CS'
查询每个系学生的人数
select Sdept,count(*) from Student group by Sdept
查询每个系男生的人数
select Sdept,count(*) from Student
where Ssex='男'
group by Sdept
查询每个系选课的人数
select Sdept,count(*) from Student
where Sno in(select Sno from SC)
group by Sdept
查询每个系选必修课(4学分以上)的人数
select Sdept,count(*) from Student
where Sno in(select Sno from SC where Cno in(select Cno from Course where
Ccredit>=4))
group by Sdept
查询每个系选必修课(4学分以上)且成绩为优秀的人数
select Sdept,count(*) from Student
where Sno in(select Sno from SC where Cno in(select Cno from Course where
Ccredit>=4) and Grade>=90)
group by Sdept


select Sdept,count(*),max(Sage),avg(Sage) from Student group by Sdept


选修了 数据库 的最高分数
select max(Grade) from SC where Cno in(select Cno from Course where Cname='数据
库')
select min(Grade) from SC where Cno in(select Cno from Course where Cname='数据
库')
select avg(Grade) from SC where Cno in(select Cno from Course where Cname='数据
库')
查询CS系的最高成绩
select max(Grade) from SC where Sno in(select Sno from Student where Sdept='CS')

查询学生信息                  统计女生的个数
select * from Student    select count(*) from Student where Ssex='女'
统计学的最大年龄
select min(Sname) from Student
select max(Sage)- min(Sage) from Student



统计CS同学的总分
select sum(Grade) from SC where Sno in(select Sno from Student where Sdept='CS')

统计学生总年龄                        统计学生的总人数                  统计学生的平均年龄
select sum(Sage) from Student    select count(*) from Student   select
sum(Sage)/count(*) from Student
                                                                  select avg(Ssex)
from Student
```

统计平均成绩
```
select avg(Grade) from SC where Cno='2'
```

统计课程总学分
```
select sum(Ccredit) from Course
```

统计必修课程数
```
select count(*) from Course where Ccredit >=4
```
统计课程数
```
select count(*) from Course
```
统计性别个数
```
select count(distinct Ssex) from Student
```
统计系个数
```
select count(distinct Sdept) from Student
```

练习： 随机查询3个 姓王的女神 姓名和年龄 (女神所有课程都为优秀的女同学) 建议用 in
```
select top 1 Sname,Sage from Student where Ssex='女'
and Sno not in(select Sno from SC where Grade <90) and Sname like '王%'
and Sno in(select Sno from SC)
order by newid()
```

分页获取数据 每页2名学生
```
select top 2 * from Student order by Sno
```

随机抽取2个同学
```
select top 2 * from Student order by newid()
```

```
select newid()
select *,newid() as 排序列 from Student
select *,rand() as 排序列 from Student

select  *,25*78 from Student order by rand()
select  rand()
```

查询年龄最大的女同学
```
select top 1 * from Student
where Ssex = '女'
order by Sage desc
```

查询课程 按照学分排序 学分相同再按课程排序
```
select * from Course order by Ccredit desc,Cname desc

create clustered index idxc on Course(Ccredit)
```

查询有先行课的课程
```
select * from Course where  Cpno is not null
```

查询有 _ 的课程

```
-- select * from Course where Cname like 'DB\_%i__' escape '\'
```

查询姓刘的同学 名字是两个字
```
select * from Student where Sname not like '刘_'
select * from Student where Sname like '刘%'
```
查询名字包含敏 并且两个字的同学

```
select * from Student where Sname like '%敏%'



 不存在 课程 自己系同学选了 而自己没有以优秀成绩来选 必须要选修课程
 select * from Student as s where
 not exists(select * from Course -- 不存在课程
   where Cno in(select Cno from SC where Sno in(select Sno from Student where
Sdept=s.Sdept))  -- 自己系同学选了
   and not exists(select * from SC where Cno=Course.Cno and Sno=s.Sno and Grade
>=90) --自己没有以优秀成绩来选
   ) and exists(select * from SC where Sno=s.Sno )   --有选课

查询选修了数据库的姓名
select Sname from Student where  Sno not in(select Sno from SC where Cno
in(select Cno from Course where Cname='数据库'))
select Sname from Student where Exists(select * from SC where Sno=Student.Sno
and exists(select * from Course
where Cno=SC.Cno and Cname='数据库'))

查询选修了以数据库为先行课的课程的姓名
select Sname from Student where Sno in(
select Sno from SC where Cno in(
select Cno from Course where Cpno in(
select Cno from Course where Cname = '数据库'
)))



select Sname from Student where Sno in(
select Sno from SC where Cno='2'
)

查询学生CS和IS,MA系
select * from Student where Sdept = 'CS' or Sdept='IS' or Sdept = 'MA'
select * from Student where Sdept in('CS','IS','MA')


 练习 用SQL查询系的学霸信息  选修了自己系全部同学所选的课程且成绩为优秀
 逆否命题
 不存在 课程 自己系同学选了 而自己没有以优秀成绩来选 必须要选修课程
 select * from Student as s where
 not exists(select * from Course -- 不存在课程
   where exists(select * from SC where Cno=Course.Cno and
   exists(select * from Student where Sno=SC.Sno and Sdept=s.Sdept) -- 自己系同学选
了
   and not exists(select * from SC where Cno=Course.Cno and Sno=s.Sno and Grade
>=90) --自己没有以优秀成绩来选
   )) and exists(select * from SC where Sno=s.Sno )   --有选课
```

3.数据更新+视图

```
对象更新 表 创建 create table  修改  alter table 删除 drop table
数据更新  新增 删除  修改
新增   insert into 表名(列...) values(值...)  值和列 一一对应
       insert into 表名 values(所有列的值)
       insert into 表名(列...) select from where 以子查询作为结果插入
```

```
       select ... into 表名 from where   先新建表  然后 将数据插入
删除   delete from 表名 where 条件
修改   update 表名  set 列名=值,... where 条件

选项  with 选型 option
视图 view  从不同角度看基础数据  基础数据如果发生变化 视图也会跟着变化
create view 视图名(列名...) as 视图主体(查询语句) with check option
alter view 视图名(列名...) as 视图主体(查询语句)  with check option
drop view 视图名
视图的意义  1、简化查询     2、安全控制
练习:  创建一个 CS系 必修课(4学分以上课程)为优秀的学生 视图
(姓名、年龄、平均成绩、优秀课门数、挂科门数、系) 不能修改系名
通过视图将 CS系必修课(4学分以上课程)为优秀 同学的年龄加1
select * from MASt update MASt set Sage = Sage + 10
alter view MASt(Sname,Sage,Grade,ECCout,BCCount,Sdept) as
          select Sname,Sage,(select avg(Grade) from SC where Sno=s.Sno),
          (select count(*) from SC where Sno=s.Sno and Grade >=90),
          (select count(*) from SC where Sno=s.Sno and Grade <60), Sdept
          from Student as s where Sdept = 'MA' and
          Sno not in(select Sno from SC
where Grade <90 and Cno in (select Cno from Course where Ccredit >=4))
and Sno in(select Sno from SC where Grade >=90 and Cno in(
select Cno from Course where Ccredit >=4))
with check option

查询必修课都为优秀的同学
select * from Student where Sno not in(select Sno from SC
where Grade <90 and Cno in (select Cno from Course where Ccredit >=4))
and Sno in(select Sno from SC where Grade >=90 and Cno in(
select Cno from Course where Ccredit >=4))

not in   in



drop view ISSt

学生姓名、课程名、学分、成绩、系
select * from vDeptAge
update vDeptAge set Age=35 where Dept='CS'
 delete from vDeptAge where Dept='CS'
 select * from CSStInfo  update CSStInfo set Ccredit=8 where Cname='数学'
select * from StInfo
create view CSStInfo as
select * from StInfo where Sdept='CS' with check option

update StInfo set Ccredit=6 where Cname='数学'
delete from StInfo where  Cname='数学'
select * from Course
insert into StInfo values('测学','测课',4,90,'IS')

create view StInfo(Sname,Cname,Ccredit,Grade,Sdept) as
select Sname ,Cname,Ccredit,Grade,Sdept from Student as s join SC
on s.Sno=SC.Sno join Course as c on SC.Cno=c.Cno


IS系辅导员 只查看自己管理的学生信息
alter view ISSt(Sno,Sname,Ssex,Sage,Sdept) as
```

```
select * from Student where Sdept='IS' with check option

create view ISSt(Sno,Sname,Ssex,Sage,Sdept) as
select * from Student where Sdept='IS'

select * from ISSt   delete from ISSt where Sno='95112'
update ISSt set Sage = 28
update ISSt set Sdept='CS' where Sno='95110'
insert into ISSt values('95110','王五','男','18','IS')

insert into ISSt values('95112','李五1','男','19','IS')

select * from Student


创建一个系平均年龄的视图
create view vDeptAge(Dept,Age) as
select Sdept,avg(Sage) from Student group by Sdept
select * from vDeptAge

select * from DeptAge

select * from Student update Student set Sage =  38
select Sdept,Avg(Sage) as Age into DeptAge from Student group by Sdept



练习：  将每个系 没有挂科 平均成绩最高同学  的 必修课(4学分以上) 成绩增加10分
    update SC set Grade = Grade + 10 where
        Cno in(select  Cno from Course where Ccredit>=4)  -- 课程限定条件
        and Sno in( -- 学生的限定条件
                select Sno from(
        select Sno,(select top 1 Sno from SC where Sno in(
        select Sno from Student where Sdept=s.Sdept)
        group by Sno order by avg(Grade) desc) as DSno
        from Student as s ) as t where Sno=DSno )


                select Sno from(   -- 学生的限定条件二
        select Sno,Sname,(select maxGrade from (
        select Sdept,max(avgGrade) as maxGrade from(
        select Sno,Sdept,(select avg(Grade) from SC where Sno=s.Sno) as avgGrade
        from Student as s) as t
        group by Sdept) as t1 where Sdept =s1.Sdept) as DMaxGrade,
        (select avg(Grade) from SC where Sno=s1.Sno) as AvgGrade
        from Student as s1) as t3 where DMaxGrade = AvgGrade
        )
        查询 学号、系、 平均分
        select Sno from(
        select Sno,(select top 1 Sno from SC where Sno in(
        select Sno from Student where Sdept=s.Sdept)
        group by Sno order by avg(Grade) desc) as DSno
        from Student as s ) as t where Sno=DSno

        查询 学号、姓名、系最高平均分、自己的平均分
        select Sno,Sname,DMaxGrade,AvgGrade from(
        select Sno,Sname,(select maxGrade from (
        select Sdept,max(avgGrade) as maxGrade from(
```

```
        select Sno,Sdept,(select avg(Grade) from SC where Sno=s.Sno) as avgGrade
        from Student as s) as t
        group by Sdept) as t1 where Sdept =s1.Sdept) as DMaxGrade,
        (select avg(Grade) from SC where Sno=s1.Sno) as AvgGrade
        from Student as s1) as t3 where DMaxGrade = AvgGrade


    select * from DeptAge
    将DeptAge cs系年龄加10
    update DeptAge set Age = 10 where Dept='CS'

     select * from Student
     将IS系学生的年龄统一设置为年龄的平均值
     update Student set Sage =(select avg(Sage) from Student) where Sdept='IS'
         将没选课的同学删除
         delete from Student where Sno not in(select Sno from SC)

        select * from FeSt;
             drop table FeSt
           delete from FeSt
        将女生信息表中 没选课的女生删除
        delete from FeSt where CCount = 0
        创建一个女生的信息表 包括学号、姓名、选课门数、平均成绩
        select Sno,Sname,(select count(*) from SC where Sno=s.Sno) as CCount,
        (select avg(Grade) from SC where Sno=s.Sno) as AvgGrade  into FeSt
        from Student as s where Ssex = '女'

        insert into DeptAge(Dept,Age)
               select Sdept,avg(Sage) from Student where Sdept is not null
               group by Sdept

        select * from DeptAge

     创建一个 系 平均年龄统计表
     create table DeptAge(
         Dept char(2) primary key,
         Age int
         )

     select * from Student
     insert into Student(Sno,Sname) values('95007','李四')
     insert into Student select '95009','王五1','女',21,'CS'
     select * from SC
     95004 选1号课程，分数是这门课程平均分
     insert into SC(Sno,Cno,Grade)
     select     '95004','1',avg(Grade) from SC where Cno='1'
```

二、第4章 数据库安全性

```
权限 (O,M)   数据对象 和 数据操作的组合  -> /<-  用户
1、数据对象   表 table  视图 view   列      行
2、数据操作   查询 select   新增 insert  删除 delete  修改 update
用户  当前用户 user    所有用户 public  print user select user as 用户名
授权 grant select(列...),update(列...),insert,delete on 表/视图 to 用户名
     同时授予可授权的权限 with grant option
```

```
回收 revoke select(列...),update(列...),insert,delete on 表/视图 from 用户名
     级联回收  cascade
视图授权  通过创建视图进行任意粒度的授权
练习  做一个通用授权策略 让所有用户能够 增删改查 自己管理的学生信息(假设有许多用户)
create view AllSt as
select *  from Student where user = Leader with check option
grant select,update,delete,insert on AllSt to public

select * from AllSt  select * from Student

create view U1St as
select * from Student where Leader = 'U1';
grant select,insert,update,delete on U1St to public
create view U1St as
select * from Student where Leader = 'U1';
grant select,insert,update,delete on U1St to U1

select * from U1St

grant select,update,delete,insert on ISSt to U2

select * from Student  alter table Student add Leader char(2)
update Student set Leader = 'U3' where Sdept='MA'

select * from ISSt
create view ISSt as
select * from Student where Sdept = 'IS' with check option

select user
select * from Student


grant select on Student to U2
授权 所有用户都可以 对 SC 进行所有操作
grant select,update,delete,insert on SC to public

grant select on Student to U1 with grant option
revoke select on Student from U1 cascade
grant select(Sno,Sname) on Student to U3
revoke select on Student from U3

grant select(Sno,Sname) on Student to U2
```

三、第5章 数据库完整性

```
数据库完整性
1、实体完整性 主码(主键)  primary key  1)主属性不能为空  2)主码值不能重复
2、参照完整性 外码(外键)  foreign key references 不为空时 值必须为 参照表中对应列的值
                级联操作  on delete,update cascade
3、用户自定义完整性  not null  非空  unique 唯一值   check(逻辑条件)
4、完整性约束条件 constraint 约束名  primary key  foreign key  check
          alter table 表名  add/drop constraint 约束名
5、触发器 trigger
create trigger 触发器名 on 表名 for insert,delete,update as 触发体
alter trigger  触发器名 on 表名 for insert,delete,update as 触发体
```

```
drop trigger 触发器名
触发体 正在操作的数据 inserted(new) 插入的数据(新数据)  deleted(old)删除的数据(旧数据)
 exists(select * from inserted/deleted where 主码 = 原主码)
练习：必修课(4学分以上)选课门数不超过10门,本系平均成绩最高的同学除外
create trigger SCIn on SC for insert as -- 如果选课超过要求 删除当前选课记录
delete from SC where exists(select * from inserted
where Sno=SC.Sno and Cno=SC.Cno -- 当前插入的数据
and Cno in(select Cno from Course where Ccredit >=4) -- 必修课
and Sno in(select Sno from SC where Cno in(select Cno from Course where
Ccredit>=4) group by Sno having count(*) >= 10) -- 必修课达到10门的学号
and Sno not in (select (select top 1 Sno from SC where Sno in(select Sno from
Student
where Sdept=s.Sdept) group by Sno order by avg(Grade) desc
) as Sno from Student as s))     -- 不是系平均成绩最高的同学
-- 候选操作
and Sno not in(select top 1 Sno from SC where Sno in(select Sno from Student
where
Sdept =(select Sdept from Student where Sno = inserted.Sno)) group by Sno order
by
avg(Grade) desc)


系平均成绩最高同学的学号 select Sno from
(
select distinct  Sdept, (select top 1 Sno from SC where Sno in(select Sno from
Student
where Sdept=s.Sdept) group by Sno order by avg(Grade) desc
) as Sno from Student as s) as t1

select Sdept,max(Grade) from (
select Sdept,Sno,(select avg(Grade) from SC where Sno=s.Sno) as Grade from
Student as s
) as t group by Sdept




每个同学只能选一门课
alter trigger StCLimit on SC for insert as
    delete from  SC where exists(select * from inserted where Sno=SC.Sno and
    Cno=SC.Cno and Sno in (select Sno from SC group by Sno having count(*)>=2))
    select * from SC

 create trigger StMath on Student for insert as
 insert into SC(Sno,Cno) select Sno,(select Cno from Course where
 Cname = '数学') from  inserted
 create trigger StDel on Student for delete
 update SC set Grade = 0 where Sno in(select Sno from deleted)

计算机系的成绩不能挂科如果小于60自动修改为60
create trigger CSStGrade on SC for insert,update as
    update SC set Grade = 60 where exists(select * from inserted where
    Sno =SC.Sno and Cno=SC.Cno and Grade < 60 and Sno in(
    select Sno from Student where Sdept = 'CS'))

    update SC set Grade = 30

女同学不大于20岁,如果大于20岁 自动修改为20岁
```

```
create trigger FStAge on Student for insert,update as
update Student set Sage = 20 where exists(select * from inserted
where Sno = Student.Sno and Sage > 20 and Ssex = '女')
select * from Student
update Student set SAge = 50
```

假设学生年龄很敏感 不能随意修改 如果修改了 记录 谁 什么时间 从... 改为 ...

```
create trigger AgeUpdaetLog on Student for update as
   insert into StAgeLog  select i.Sno,user,getdate(),d.Sage,i.Sage
   from inserted as i join deleted as d on i.Sno=d.Sno
select * from StAgeLog   select * from Student
update Student set Sage = 25 where Sno='95001'
grant select ,update on Student to public
```

```
alter trigger StInsert on Student for insert as
insert into StIn select *  from inserted
select * from StIn
INSERT INTO Student VALUES('95013','李勇3','男',21,'CS');

create table StAgeLog(
    Sno char(5),
    UserName varchar(10),
    updateTime datetime,
    oldAge int,
    newAge int
    )
select * from Student

create Table SC(
    Sno char(2),
    Cno char(2),
    check(not exists(select * from SC where
```

练习：创建系表(DeptNo,DeptName,DeptNum) 和学生表(Sno,Sname,Ssex,Sage,DeptNo)
系主码DeptNo,学生主码Sno 要求学生的系号必须是存在的系，如果系取消了学生也要删除，
男同学不大于18岁，女同学不大于20岁

```
insert into St1 values('02','张三1','女',29,'01')  select * from St1
insert into Dept values('01','计算机',500)  select * from Dept  delete from Dept
create table St1(
    Sno char(2) primary key,
    Sname varchar(10),
    Ssex char(2),
    Sage int,
    DeptNo char(2) foreign key references Dept(DeptNo) on delete cascade,
    check((Sage <=18 and Ssex='男') or (Sage <=20 and Ssex='女')))
 create table Dept(
    DeptNo char(2) primary key,
    DeptName varchar(10),
```

```
    DeptNum int)



create table StSeC(
    Sno char(2) foreign key references St(Sno) on update cascade,
    Cno char(2) ,
    Grade int,
    primary key(Sno,Cno)
    )
insert into StSeC values('02','01',89)
update StSec set Sno='03' where Sno = '04'   delete from StSeC where Sno='03'
select * from StSeC
select * from St
update St set Sno='05' where Sno = '02'
delete from St where Sno = '03'

drop table StSeC
create table StSeC(
    Sno char(2) foreign key references St(Sno) on update cascade,
    Cno char(2) ,
    Grade int,
    primary key(Sno,Cno)
    )
select * from St
alter table St add constraint stAge check(Sage >=18)
alter table St drop constraint stAge
insert into St values('03','张三',10)
insert into St(Sno,Sname,Sage) values('04','张三',20)
insert into St(Sno) values('05')
drop table St
create table St(
    Sno char(2) ,
    Sname varchar(10) ,
    Sage int
    )
```

四、第8章 数据库编程

```
1、基本输入(insert update delete) 输出 print  select  字符串 '' =等于
2、变量定义 Declare @变量名 类型   赋值  set @变量名 = 值  select @变量名 = 值 from
3、流程控制  ;区分语句   {begin    }end
if 条件 else    case when then when then end   while
4、函数 function  create function 函数名(参数 类型,...) returns 返回值类型 as
                  alter function 函数名(参数 类型,...)  returns 返回值类型 as
                  drop function 函数名
        函数调用  dbo.函数名(参数,...)
5、存储过程 procedure create/alter procedure 过程名(参数 类型=缺省值,...) as
                    drop procedure 过程名
    存储过程调用       execute 过程名 参数
6、临时表 #表名 当前会话临时表  ##表名 全局临时表
7、游标 Cursor
   1、定义  Declare 游标名 Cursor for select ... from
   2、打开  open 游标名
```

```
   3、取内容 Fetch Next from 游标名 into 变量名 @@Fetch_Status 0-未到达尾部 -1-到达尾部
     while(@@Fetch_Status = 0) begin
        ...,Fetch Next from 游标名 into 变量名,...
    end
   4、关闭 close 游标名 5、释放 deallocate 游标名

练习：  查询学生的信息   学号、姓名、选课情况(课程名:成绩,...)
                          1     李勇     数据库:90,操作系统:80,
    select Sno,Sname,dbo.GetSC(Sno) as 选课情况 from Student
    消息游标的方式输出
    declare StCursor Cursor for select Sno,Sname from Student;
    declare @Sno varchar(10),@Sname varchar(10);
    open StCursor;
    fetch Next from StCursor into @Sno,@Sname;
    while (@@Fetch_Status=0) begin
        declare @SC varchar(255),@Cname varchar(10),@Grade varchar(3);
        set @SC = '';
        Declare StSC Cursor for
        select Cname,Grade from SC join Course as c on SC.Cno=c.Cno where
Sno=@Sno;
        open StSC;
        Fetch Next from StSC into @Cname,@Grade;
        while (@@Fetch_Status = 0) begin
            set @SC=@SC + @Cname + ':' + @Grade + ',';
            Fetch Next from StSC into @Cname,@Grade;
        end; close StSC;  deallocate StSC;
        print @Sno + ';' + @Sname + ';' + @SC;
        fetch Next from StCursor into @Sno,@Sname;
    end;close StCursor;deallocate StCursor;



alter function GetSC(@Sno varchar(10)) returns varchar(255) as begin
    declare @SC varchar(255),@Cname varchar(10),@Grade varchar(3);
    set @SC = '';
    Declare StSC Cursor for
    select Cname,Grade from SC join Course as c on SC.Cno=c.Cno where Sno=@Sno;
    open StSC;
    Fetch Next from StSC into @Cname,@Grade;
    while (@@Fetch_Status = 0) begin
        set @SC=@SC + @Cname + ':' + @Grade + ',';
        Fetch Next from StSC into @Cname,@Grade;
    end;-- close StSC; -- deallocate StSC;
    return @SC;
end



编写一个存储过程 根据课程号 查询课程名、学分、先行课程名、被选情况(学生姓名:成绩,...)
                                  数据库    4     数据结构     李勇:90,刘晨:80,张三:75

select Cname,Ccredit,dbo.GetSelInfo(Cno) as 选课情况  from Course
create function GetSelInfo(@Cno varchar(10)) returns varchar(255) begin
Declare @SelInfo varchar(255);
    Declare SCCursor Cursor for select Sname,Grade
    from SC join Student as s on SC.Sno=s.Sno where Cno=@Cno;
    Declare @name varchar(10),@Grade varchar(10);
    open SCCursor;
```

```
    Fetch Next from SCCursor into @Name,@Grade;
    set @SelInfo = '';
    while(@@Fetch_Status =0) begin
        set @SelInfo = @SelInfo + @name + ':' + @Grade + ',';
        Fetch Next from SCCursor into @Name,@Grade;
    end; close SCCursor;deallocate SCCursor;
    return @SelInfo;
end;


   用消息的形式显示学生的信息    ...系男生...
   Declare StCursor Cursor for select Sdept,Ssex,Sname from Student;
   Declare @Sdept varchar(10),@Ssex varchar(2),@Sname varchar(8);
    print '0';print @@Fetch_status
   open StCursor;
   print '1';print @@Fetch_status
   Fetch Next from StCursor into @Sdept,@SSex,@Sname;
   print @Sdept +'系' + @Ssex + '生' + @Sname ;
   print '2';print @@Fetch_status
   while(@@FETCH_STATUS = 0) begin
    Fetch Next from StCursor into @Sdept,@Ssex,@Sname;
    print @Sdept +'系' + @Ssex + '生' + @Sname ;
      print '4';print @@Fetch_status
   end;  print '5';print @@Fetch_status
   close StCursor;deallocate StCursor;

   select * from Student


   print 'sdfsdf'123
    print @@Fetch_status

Declare CCursor Cursor for select Cno, Cname,Ccredit,(select Cname from Course
where
Cno = c.Cpno) as CpName   from Course as c;
Declare @Cno varchar(10), @Cname varchar(10),@Ccredit varchar(10),@CpName
varchar(10);
open CCursor;
Fetch Next from CCursor into @Cno, @Cname,@Ccredit,@CpName;
while(@@Fetch_Status=0) begin
    Declare @SelInfo varchar(255);
    Declare SCCursor Cursor for select Sname,Grade
    from SC join Student as s on SC.Sno=s.Sno where Cno=@Cno;
    Declare @name varchar(10),@Grade varchar(10);
    open SCCursor;
    Fetch Next from SCCursor into @Name,@Grade;
    set @SelInfo = '';
    while(@@Fetch_Status =0) begin
        set @SelInfo = @SelInfo + @name + ':' + @Grade + ',';
        Fetch Next from SCCursor into @Name,@Grade;
    end; close SCCursor;deallocate SCCursor;
    print @Cname +';' +@CCredit + ';' + @CpName + ';' + @SelInfo;
    Fetch Next from CCursor into @Cno, @Cname,@Ccredit,@CpName;
end;close CCursor;deallocate CCursor;
```

```
alter function GetStInfoByCno(@Cno varchar(10)) returns varchar(255) as begin
    Declare @StInfo varchar(255);
    select @StInfo = Sname + ':' + trim(Str(Grade)) from SC join Student as s on
SC.Sno=s.Sno
    where Cno = @Cno;
    return @StInfo;
end
print trim(Str(123))
print dbo.GetStInfoByCno('1')
select * from SC

练习  编写一个存储过程 根据课程号 查询选修学生的姓名、年龄、系、成绩
     调用存储过程  execute QuerySt '3'
     alter procedure QuerySt(@Cno varchar(10)='2') as begin
        select c.Cno,Cname, Sname,Sage,Sdept,GRade
        from Student as s join SC on s.Sno=SC.Sno join Course as c on
c.Cno=SC.Cno
        where SC.Cno = @Cno
    end


     alter procedure GetSt(@Sdept varchar(10)='IS') as begin
      select Sno,Sage,dbo.GetSName(Sname,Ssex) as 称谓 from Student
      where Sdept = @Sdept;
  end;
    Execute GetSt 'CS'
 编写一个存储过程 根据系 查询学生的学号、年龄、称谓(男同学 某先生  女同学 某女士)
 创建一个函数 根据姓名和性别返回称谓
 print dbo.GetSName('张三','中')
 create function GetSName(@Sname varchar(10),@Ssex char(2)) returns varchar(6)
as begin
    Declare @S varchar(6);
    if(@Ssex = '男') set @S= SUBSTRING(@Sname,1,1) + '先生'
    else set @S = SUBSTRING(@Sname,1,1) + '女士';
    return @S
 end;

 alter procedure GetSt(@Sdept varchar(10)='IS') as begin
    select Sno,Sname,Sage,Ssex into #temp from Student where Sdept = @Sdept;
    update #temp set Sname = substring(Sname,1,1) + '先生' where Ssex = '男';
    update #temp set Sname = SUBSTRING(Sname,1,1) + '女士' where Ssex = '女';
    select Sno,Sname,Sage  from #temp ;
    drop table #temp;
 end

 select Sno,Sname,Sage,Ssex,Sdept into #temp from Student;
 select * from ##temp2 ;
 drop table #temp ;
  alter procedure GetSt(@Sdept varchar(10)='IS') as begin
    declare @Sno char(5),@S varchar(10),@Sage int,@Sname varchar(10),@Ssex
char(2);
    select @Sno= Sno,@Sname = Sname,@Sage= Sage,@Ssex = Ssex from Student where
Sdept = @Sdept;
    if(@Ssex='男')
        set @Sname = substring(@Sname,1,1) + '先生'
    else
        set @Sname = substring(@Sname,1,1) + '女士';
    select @Sno,@Sname, @Sage;
```

```
 end
 print substring('数量大幅减少了',2,2)
 Declare @Sno char(5);
 select @Sno = Sno from Student order by Sno DEsc;
 print @Sno;


练习:    查询学生系、男生人数、不及格人数 (实现过程中用函数)
        select distinct Sdept,dbo.GetMCount(Sdept) as 男生数,dbo.GetCByDept(Sdept)
as 挂科
        from Student
        select Sdept,count(*),dbo.GetCByDept(Sdept) from Student
        where Ssex = '男' group by Sdept

   根据系返回 不及格门数
   alter function GetMCount(@Sdept varchar(10)) returns int as begin
    declare @MCount int;
    set @MCount = (select count(*) from Student where Sdept= @Sdept and
Ssex='男');
    return @MCount;
   end;
    create function GetCByDept(@Sdept varchar(10)) returns int as begin
        declare @Count int;
        select @Count= count(*)  from SC where Grade < 60 and Sno in(
        select Sno from Student where Sdept = @Sdept);
        return @Count;
    end

查询学生姓名及学习榜样(系平均成绩最高的同学)
select Sname,dbo.DESt(Sdept) from Student
函数 根据系返回平均成绩最高的同学 print dbo.DESt('MA')
create function DESt(@Sdept varchar(10)) returns varchar(20) as begin
    declare @Sname varchar(20),@Sno char(5);
    select top 1 @Sno = Sno from SC where Sno in(select Sno from Student
    where Sdept = @Sdept) group by Sno order by avg(Grade) desc ;
    select @Sname = Sname from Student where Sno = @Sno ;
    return @Sname;
end;


    print getdate()
    print dbo.MyAdd(1,2)
    create function MyAdd(@i int,@j int) returns int as begin
        return @i + @j;
    end

select getdate()


int myAdd(int j,int i) {
return i+j;
}
Declare @i int,@j int,@s varchar(10);
set @i = 10;
while (@i>0) begin
    print @i;
    set @i = @i - 1;
end;
```

```
select *,
case
    when Grade >= 85 then '优秀'
    when Grade <85 and Grade >=60 then '及格'
    when Grade <60 then '不及格'
end as DGrade
from SC



set @s = case(@i)
    when 1 then '优秀'
    when 2 then '良'
    when 3 then '一般'
end;
print @s;


if(@i >=5)
    print '大数'
else
    print '小';



-- set @i= (select count(*) from Student); -- 学生人数
select @i = count(*),@j = avg(Sage) from Student; -- @i 人数  @j 平均年龄
print @i;
print @j;
select * from Student

c语言 = 赋值  ==等于


create table St(
Sno char(2)
)

select * from Student where Sno = '95001'

int i,j;
i=3;

print 'hello world! '  select  'hello world! '
```

五、第6章 关系数据理论+第7章 数据库设计

```
数据库设计不好存在
1、插入异常
2、删除异常
3、数据冗余
4、更新复杂
原因  属性 和 码 之间不合理关系
```

属性间关系　决定关系　函数依赖 ->
R(U,F)　　U-属性集　F-函数依赖集　最小依赖集
1、找出关系模式的函数依赖 F={?}　X->Y
2、找出关系模式的码　K-f>U　　L类+N类 有可能LR
通过属性闭包判断　码的属性叫主属性　其他属性叫非主属性　(码是指候选码 可能多个)
属性组 X判断是否为码:　1) X+ = U　2) 不存在X的真子集X`　X`+ = U
1)L类属性　只在函数依赖左边出现的属性
2)R类属性　只在函数依赖右边出现的属性
3)LR类属性　在函数依赖左右都出现的属性
4)N类属性　不在函数依赖出现的属性
3、范式理论　通过码与属性间关系 判断关系模式是否存在异常　证明 反证法
1) 2NF　不存在非主属性对码部分函数依赖　非主属性都完全函数依赖于码
　　单属性码关系模式一定满足2NF
2) 3NF　不存在非主属性对码传递函数依赖　传递函数依赖是公理 所有R不满足3NF?
3) BCNF 函数依赖的左边必包含码(最小函数依赖集)　　任意函数依赖左边包含码　F+
4、数据依赖的公理系统　函数依赖成立针对所有元组　函数依赖不成立举出2个元组的反例
A1 自反律 Y包含于X 则 X->Y　平凡的函数依赖
A2 增广率 X->Y 则 XZ->YZ
A3 传递率 X->Y,Y->Z 则 X->Z
推论:　函数依赖 左边可以合并　　右边可以分解与合并
5、函数依赖闭包 所有正确函数依赖集合 叫函数依赖的闭包　F+
6、属性闭包　属性X+ 所有由属性集X所决定的属性集(通过公理推到出决定关系的属性集)
Y 包含于 X+　　X->Y 成立　如何求X+
1)加入X的属性集 2)找现有属性集所能决定的属性集继续加入 3)如果有变化继续迭代
时间复杂度 |U|-|X| + 1
R(U,F)　F　A1A2A3 扩展许多函数依赖　F'　　F' 和 F+ 什么关系?　　F' = F+
F+ 通过 A1A2A3 收缩 Fmin　F+ 和 F 定价　F+ 和 Fmin 等价　F 和 Fmin 等价
7、F+ = G+　称F和G定价　　F包含于G+　G包含于F+
8、最小依赖集 Fmin
1)分右　将函数依赖右侧多个属性变为单个属性
2)去传　去掉能由其他函数依赖推导出来的函数依赖
3)分去左 将左边变成最简,去掉部分函数依赖 变为完全函数依赖
所有的处理都先转换为Fmin　求码　判断范式　范式分解
9、模式分解
1)正确 保持无损连接 分解后的关系模式 通过自然连接 能够恢复到原关系
判定算法: 1) 构造矩阵　属性作为列　分解作为行 在分解处填a
　　2) 根据函数依赖进行变换 对每个函数依赖 判断左边是否有相同行a,
　　如果有相同行a,右侧有a,则将这些行右侧均设置为a
　　3)如果有一行为a 则分解为无损连接(正确) 算法终止
　　4)如果每次变换有变换 则继续迭代 直到无变换

2)好　保持函数依赖　容易判断　保持函数依赖+码 保持无损连接
10、模式合并 码相同的关系模式要合并　合并方法 将多个关系模式的属性合在一起
变成一个新的关系模式 码不变 主属性不变 非主属性不变　范式不变
11、按照最小依赖集的函数依赖进行分解 一定达到3NF
12、将关系模式分解为3NF 保持函数依赖及无损连接的分解算法
1)求最小依赖集Fmin　a.分右　b.去传　c.分去左
2)按照函数依赖左边相同属性合并 进行模式分解　　保持函数依赖并且达到3NF
3)如果分解后的关系模式不包含原关系模式的码　　保持无损连接
则添加原关系模式的码作为一个新的分解
13、将关系模式分解为BCNF 保持无损连接
1)若R不属于BCNF, 存在X->Y,X不包含码
2)分解成XY 和 U-Y, XY是码是X 符合BCNF　分解无损链接
3)R = U-Y 继续进行BCNF判断分解

(Sno,Sdept,SdeptAddress)　不属于3NF　Sno->Sdept, Sdept->SdeptAddress
S1(Sno)　S2(Sdept)　S3(SdeptAddress)　　3NF　　不正确 不好
F=　　函数依赖　全丢

S1(Sno,SdeptAddress) S2(Sno,Sdept) 3NF 正确 不好
  Sno->SdeptAddress Sno->Sdept, 函数依赖 丢 Sdept->SdeptAddress

S1(Sno,SdeptAddress) S2(Sdept,SdeptAddress) 3NF 不正确 不好
  Sno->SdeptAddress Sdept->SdeptAddress 函数依赖 丢 Sno->Sdept
S1(Sno,Sdept) S2(Sdept,SdeptAddress) 3NF 正确 好
  保持函数依赖

(Sno,Sname,Bno,Bname) F={Sno->Sname,Bno->Bname} 码 (Sno,Bno)
R1(Sno,Sname) BCNF
R2(Bno,Bname) BCNF
R3(Sno,Bno) BCNF

R2(Sno,Bno,Bname) R1(Bno,Bname) BCNF R2(Sno,Bno) BCNF

S1(Sno,Sname) S2(Bno,Bname) 不正确 好 3NF BCNF
 Sno->Sname Bno->Bname

S1(Sno,Sname),S2(Bno,Bname) S3(Sno,Bno) 正确 好 3NF BCNF
Sno-Sname, Bno->Bname

练习 已知关系模式R<U，F>，其中•U={A，B，C，D，E}；•
F={A→BC，B→D，CD →E，D→C，AC→B}。
求 A+
1、(A)
2、(ABC)
3、(ABCD)
4、(ABCDE)
5、(ABCDE)
A+ =(ABCDE) = U A没有真子集 A 是码

Fmin 等价于 F
Fmin+ = F+

StInfo(U,F)
select * from StInfo 写完所有函数依赖 所有函数集合
Sname+ =(Sname)
Fmin={Sno->Sname,Sno->Sdept,Cno->Cname,Cno->Ccredit,(Sno,Cno)->Grade} (Sno,Cno)
按照函数依赖分解
R1(Sno,Sname,Sdept) R2(Cno,Cname,Ccredit),R3(Sno,Cno,Grade)

(Sdept,Cname)->Sdept,(Sno,Ccredit)->Sname,(Sno,Sname)->Sname,(Sno,Grade)->Sname

练习 已知R属于BCNF，证明R属于2NF

Sno,Sage -> Sname

```
Sno Sage->Sname
(Sno,Sge)->Sname

StInfo(Sno,Sname,Sage,Ssex,Sdept,SdeptAddress,SdeptInfo)
Sno->Sdept, Sdept->SdeptAddress    Sno->SdeptAddress



L类属性
(Sno,Sdet)->Sno


Fmin={Sno->Sname,Sno->Sage,Sno->Ssex,Sno->Sdept,Sdept->SdeptAddress,Sdept-
>SdeptInfo}
(Sname,Sno)->Sno
(Sno,Ssex)->(Ssex,Sdept)
(Sno,Ssex)->Ssex, Sno->Ssex
(Sno,Ssex)->Sdept Sno->Sdept
(Sage,Sno)->SdeptInfo
(Sage,Sname)->Sage


Sdept+ = (Sdept,SdeptAddress,SdeptInfo)
Sno+ = (Sno,Sname,Sage,Ssex,Sdept,SdeptAddress,SdeptInfo) =U
码   K+  = U
不存在 K的真子集 K'+ = U


Sno->Sdept,Sdept->SdeptAddress  传递率  Sno->SdeptAddress

F={Sno->Sname,Sno->Sage,Sno->Ssex,Sno->Sdept, Sdept->SdeptAddress,Sdept-
>SdeptInfo,
Sno->SdeptAddress,Sno->SdeptInfo}
S1(Sno,Sname,Sage,Ssex,Sdept)
S2(Sdept,SdeptAddress,SdeptInfo)
 Sno 是码
 (Sno,Cno)  是码
 (Sno,Cno,Grade)
 (Sno,Cno)
Sno->SdeptAddress,  Sno->Sdept, Sdept->SdeptAddress, Sdept -/> Sno 是传递

Sno->Sname,Sno->(Sno,Sname),(Sname,Sdept)->Sname ,(Sdept,Ccredit)->Sdept


St(Sno,Sname,Ssex,Sage) 假设不允许重名
Sno->Sname, Sno->Ssex,Sno->Sage  Sname->Ssex,Sname->Sage
Sno->Ssex  Sno->Sname, Sname->Ssex  Sname->Sno 不是传递

码  Sno
Sno -> Sname,Sno->Sdept
Sno->(Sname,Sdept)
Cno->(Cname,Ccredit)
Cno->Cname,Cno->Ccredit
Sno->Sname
(Sno,Cno) -> Sname
(Sno,Cno) ->Sno
select * from StInfo     码(Sno,Cno)   Sno->Sname   Sno不包含码 不满足BCNF
S1(Sno,Sname,Sdept)  码Sno
```

```
S2(Cno,Cname,Ccredit) 码Cno
S3(Sno,Cno,Grade)   码(Sno,Cno)

StInfo(U,F)
U=(Sno,Sname,Sdept,Cno,Cname,Grade,Ccredit)
F={Sno->Sname,Sno->Sdept,Cno->Cname,Cno->Ccredit,(Sno,Cno)->Grade}
L:Sno,Cno LR: N:
码 (Sno,Cno) - > Sno, Sno->Sname, Sno-/>(Sno,Cno)   (Sno,Cno)->Sname

(Sno,Cno) -p> Sname
(Sno,Cno) -p>   Sdept
(Sno,Cno) -p>  Cname
(Sno,Cno) -p>   Ccredit
(Sno,Cno)  -f>   Grade
(Sno,Cno) -p> Sno
(Sno,Cno) -p>   Cno
St(Sno,Sname,Sdept)   C(Cno,Cname,Ccredit)   SC(Sno,Cno,Grade)  属于2NF
码 Sno                  码Cno                  码(Sno,Cno) -f> Grade  (Sno,Cno)-
p>Sno



Student(U,F)
U={Sno,Sname,Ssex,Sage,Sdept,CompanyName,CTime)
F={Sno->Sname,Sno->Ssex,Sno->Sage,Sno->Sdept,(Sno,CompanyName)->CTime}
码(Sno,CompanyName)
(Sno,CompanyName) -p> Sname
(Sno,CompanyName) -p> Ssex
S1(Sno,Sname,Ssex,Sdept)  属于2NF
S2(Sno,CompanyName,CTime) 属于2NF


假设学生只能在一个公司实习一次


L类属性:Sno R类属性: LR:  N:CompanyName,CTime
码 (Sno,CompanyName,CTime)

练习:  写入Student 的码

码 Sno


(Ssex,Sdept) -> Ssex   Ssex - > Ssex



F={Sno->Sname,Sno->Ssex,Sno->Sage,Sno->Sdept,Bno->Bname}
L类:Sno Bno
N类:
(Sno,Bno)
K=Sno
Sno-f>U
(Sno,Sname) -p> U
Sno-f>U


--  Sname->Sno,Sname...}
```

```
select * from StInfo
F={Sno->Sname,Sno->Sdept,Cno->Cname,Cno->Ccredit,(Sno,Cno)->Grade}
L类：Sno,Cno
(Sno,Cno)-P>Sname
Sno-F>Sname

(Sno,Cno)->Sno  (


设计                开发          系统
ER                  sql + 语言 思想  OOD
关系模型(表 table) ？ 到底设计几张表    ？ 为什么设计成这几张表
select * from StInfo  码 primary key(Sno,Cno)
1、插入异常
2、删除异常
3、数据冗余
4、更新复杂


select * from Student select * from Course select * from SC
 select  s.Sno,Sname,Sdept ,c.Cno,Cname,Grade,Ccredit into StInfo from Student
as s join SC on s.Sno=SC.Sno join
 Course as c on SC.Cno=c.Cno
 drop table Stinfo
```