--创建用户  
create user han identified by han default tablespace  
users Temporary TABLESPACE Temp;  
grant connect,resource,dba to han; //授予用户han开发人员的权利

--------------------对表的操作--------------------------  
--创建表  
create table classes(  
       id number(9) not null primary key,  
       classname varchar2(40) not null  
)        
--查询表  
select \* from classes;

--删除表  
drop table students;

--修改表的名称  
rename alist\_table\_copy to alist\_table;

--显示表结构  
describe test --不对没查到

-----------------------对字段的操作-----------------------------------  
--增加列  
alter table test add address varchar2(40);

--删除列  
alter table test drop column address;

--修改列的名称  
alter table test modify address addresses varchar(40;

--修改列的属性  
alter table test modi

create table test1(  
       id number(9) primary key not null,  
       name varchar2(34)  
      )  
rename test2 to test;

--创建自增的序列  
create sequence class\_seq increment by 1 start with 1 MAXVALUE 999999 NOCYCLE NOCACHE;

select class\_seq.currval from dual

--插入数据  
insert into classes values(class\_seq.nextval,'软件一班')  
commit;

--更新数据  
update stu\_account set username='aaa' where count\_id=2;  
commit;

--创建唯一索引  
create unique index username on stu\_account(username);   --唯一索引不能插入相同的数据

--行锁 在新打开的对话中不能对此行进行操作  
select \* from stu\_account t where t.count\_id=2 for update; --行锁

--alter table stuinfo modify sty\_id to stu\_id;

alter table students drop constraint class\_fk;  
alter table students add constraint class\_fk foreign key (class\_id) references classes(id);--外键约束  
alter table stuinfo add constraint stu\_fk foreign key (stu\_id) references students(id) ON DELETE CASCADE;--外键约束,级联删除

alter table stuinfo drop constant stu\_fk;

insert into students values(stu\_seq.nextval,'张三',1,sysdate);

insert into stuinfo values(stu\_seq.currval,'威海');

select \* from stuinfo;

create table zhuce(  
       zc\_id number(9) not null primary key,  
       stu\_id number(9) not null,  
       zhucetime date default sysdate

)

create table feiyong (  
       fy\_id number(9) not null primary key,  
       stu\_id number(9) not null,  
       mx\_id number(9) not null,  
       yijiao number(7,2) not null default 0,  
       qianfei number(7,2) not null  
         
)

create talbe fymingxi(  
       mx\_id number(9) not null primary key,  
       feiyong number(7,2) not null,     //共7位数字，小数后有两位  
       class\_id number(9) not null  
}

create table card(  
       card\_id number(9) primary key,  
       stu\_id number(9) not null,  
       money number(7,2) not null default 0,  
       status number(1) not null default 0   --0表可用,1表挂失  
)

--链表查询

select c.classname||'\_'||s.stu\_name as 班级\_姓名,si.address from classes c,students s , stuinfo si where c.id=s.class\_id and s.id=si.stu\_id;   
insert into students values(stu\_seq.nextval,'李四',1,sysdate);  
insert into stuinfo values(stu\_seq.currval,'南京');

--函数  
select rownum,id,stu\_name from students t order by id asc;

--中间表实现多对多关联  
--（1   1， 1   n，n 1，n n ）

--1 n的描述   1的表不作处理   n的表有1表的字段  
--1 1的描述   主外键关联  
--n n的描述 中间表实现多对多关联

create　table course(  
         course\_id number(9) not null,  
         couser\_name varchar2(40) not null  
)  
alter table course to couse;  
create table stu\_couse(  
       stu\_couse\_id number(9) primary key,  
       stu\_id number(9) not null,  
       couse\_id number(9) not null

)

create unique index stu\_couse\_unq on stu\_couse(stu\_id,couse\_id); --唯一学生  
create sequence stu\_couse\_seq increment by 1 start with 1 MAXVALUE 999999 NOCYCLE NOCACHE;

create sequence couses\_seq increment by 1 start with 1 MAXVALUE 999999 NOCYCLE NOCACHE;  
insert into course values(couses\_seq.nextval,'计算机原理');  
insert into course values(couses\_seq.nextval,'编译原理');  
insert into course values(couses\_seq.nextval,'数据库原理');  
insert into course values(couses\_seq.nextval,'数据结构');  
insert into course values(couses\_seq.nextval,'计算机基础');  
insert into course values(couses\_seq.nextval,'C语言初步');  
commit;

insert into stu\_couse values(stu\_couse\_seq.nextval,1,1);  
insert into stu\_couse values(stu\_couse\_seq.nextval,1,3);  
insert into stu\_couse values(stu\_couse\_seq.nextval,1,5);  
insert into stu\_couse values(stu\_couse\_seq.nextval,1,5);

insert into stu\_couse values(stu\_couse\_seq.nextval,2,1);  
commit;  
select \* from stu\_couse;  
select \* from course;

--select s.stu\_name,sc.couse\_id, c.couser\_name from students s,course c,stu\_couse sc where stu\_id=1

--select couse\_id from stu\_couse where stu\_id=1

select cl.classname,s.stu\_name,c.couser\_name from stu\_couse sc, students s,course c,classes cl where s.id=sc.stu\_id and sc.couse\_id=c.course\_id and s.class\_id=cl.id and s.id=1;

--班级——姓名  
select c.classname,s.stu\_name from students s,classes c where s.class\_id=c.id and s.id=2;

select \* from students s where s.id=2  
--班级——姓名——课程

select cl.classname,s.stu\_name,c.couse\_name from stu\_couse sc,students s,classes cl,couse c where sc.stu\_id=s.id and sc.couse\_id=c.couse\_id and s.id=26;

--sql 语句的写法，现写出关联到的表，然后写出要查找的字段，第三 写出关联条件   ，记住在写关联到的表时先写数据多的表，这样有助于提高sql的效率

select c.couser\_name,s.stu\_name from stu\_couse sc,students s,course c where c.course\_id=1 and c.course\_id=sc.couse\_id and sc.stu\_id=s.id;

select s.stu\_name from students s,stu\_couse sc where s.id=sc.stu\_id group by s.id,s.stu\_name;

select c.classname,count(sc.couse\_id) from stu\_couse sc,students s,classes c where s.class\_id=c.id and s.id=sc.stu\_id group by c.classname;

select s.stu\_name, count(sc.couse\_id) from stu\_couse sc,students s,classes cl where s.id=sc.stu\_id group by s.id,s.stu\_name having count(sc.stu\_couse\_id)>3;  
班级 学生 选课数量  
select cl.classname,count(sc.stu\_couse\_id) from stu\_couse sc,students s,classes cl where s.id=sc.stu\_id and s.class\_id=cl.id group by cl.classname;

--班级 学生 选课数量  
select cl.classname,s.stu\_name,count(sc.stu\_couse\_id) from stu\_couse sc,students s,classes cl where s.id=sc.stu\_id and s.class\_id=cl.id group by s.stu\_name;

select cl.classname,s.stu\_name,count(sc.stu\_couse\_id) from stu\_couse sc ,students s,classes cl where sc.stu\_id=s.id and s.class\_id=cl.id group by s.id;

select cl.classname,s.stu\_name,count(sc.stu\_couse\_id) from stu\_couse sc,students s,classes cl where sc.stu\_id=s.id and s.class\_id=cl.id group by s.stu\_name;  
--班级 学生 所选课程id 所选课程名称

--创建试图 目的把表联合起来 然后看成一个表，在与其他的联合进行查询   
create view xsxk as select cl.classname, s.stu\_name,c.couse\_id, c.couse\_name from stu\_couse sc,students s,classes cl,couse c where sc.stu\_id=s.id and sc.couse\_id=c.couse\_id and s.class\_id=cl.id;

select \* from xsxk

create view classstu as select s.id,c.classname,s.stu\_name from students s,classes c where c.id=s.class\_id;  
drop view classstu; --删除视图  
select \* from classstu;  
create view stu\_couse\_view as select s.id ,c.couse\_name from stu\_couse sc,students s,couse c where s.id=sc.stu\_id and sc.couse\_id=c.couse\_id;  
select \* from stu\_couse\_view;  
create view csc as select cs.classname,cs.stu\_name,scv.couse\_name from classstu cs,stu\_couse\_view scv where cs.id=scv.id;  
select \* from csc;

select \* from classes cross join students; --全连接，相当于select \* from classes,students;

select \* from classes cl left join students s on cl.id=s.class\_id; --左连接 不管左表有没有 都显示出来  
select \* from classes cl right join students s on cl.id=s.class\_id; --右连接  
select \* from classes cl full join students s on cl.id=s.class\_id; --全连接

insert into classes values(class\_seq.nextval,'软件四班');

create table sales(  
       nian varchar2(4),  
       yeji number(5)  
         
);  
insert into sales values('2001',200);  
insert into sales values('2002',300);  
insert into sales values('2003',400);  
insert into sales values('2004',500);  
commit;  
select \* from sales;  
drop table sale;

select s1.nian,sum(s2.yeji) from sales s1,sales s2 where s1.nian>=s2.nian group by s1.nian order by s1.nian desc;

select s1.nian,sum(s2.yeji) from sales s1,sales s2 where s1.nian>=s2.nian group by s1.nian;

s  
       年       年业绩总和   
       2001     200  
       2002     500  
       2003     900  
       2004     1400

create table test1(  
       t\_id number(4)  
);

create table org(  
       org\_id number(9) not null primary key,  
       org\_name varchar2(40) not null,  
       parent\_id number(9)  
);

create sequence org\_seq increment by 1 start with 1 MAXVALUE 999999 NOCYCLE NOCACHE;  
drop sequence org\_seq;  
insert into org values(1,'华建集团',0);  
insert into org values(2,'华建集团一分公司',1);  
insert into org values(3,'华建集团二分公司',1);  
insert into org values(4,'华建集团财务部',1);  
insert into org values(5,'华建集团工程部',1);  
insert into org values(6,'华建集团一分公司财务处',2);  
insert into org values(7,'华建集团一分公司工程处',2);

select \* from org;  
--不正确 不能实现循环  
select b.org\_id , b.org\_name ,b.parent\_id from org a,org b where a.org\_id=7 and a.parent\_id=b.org\_id;  
select \* from org connect by prior parent\_id=org\_id start with org\_id=7 order by org\_id;  
select \* from org connect by prior org\_id=parent\_id start with org\_id=1 order by org\_id;

create table chengji(  
       cj\_id number(9) not null primary key,  
       stu\_cou\_id number(9) not null,  
       fen number(4,1)  
         
);  
insert into chengji values(1,1,62);  
insert into chengji values(2,2,90);  
insert into chengji values(3,3,85);  
insert into chengji values(4,4,45);  
insert into chengji values(5,5,68);  
insert into chengji values(6,6,87);  
commit;  
select \* from chengji;  
select \* from stu\_couse;  
--在oracle 中好像不适用 alter table chengji change stu\_cou\_id stu\_couse\_id;alter table shop\_jb change price1 price double;

学生姓名   平均分  
select s.stu\_name,avg(cj.fen) from stu\_couse sc,chengji cj,students s where s.id=sc.stu\_id and sc.stu\_couse\_id=cj.stu\_couse\_id group by s.id,s.stu\_name;  
select s.stu\_name from students s,stu\_couse sc,chengji cj where s.id=sc.stu\_id and sc.stu\_couse\_id=cj.stu\_couse\_id group by s.id,s.stu\_name;  
select s.stu\_name,cj.fen from students s,stu\_couse sc,chengji cj where s.id=sc.stu\_id and sc.stu\_couse\_id=cj.stu\_couse\_id and cj.fen>60;

学生姓名   科目   成绩  
select s.stu\_name,c.couse\_name,cj.fen from stu\_couse sc,students s,couse c,chengji cj where sc.stu\_id=s.id and sc.couse\_id=c.couse\_id and sc.stu\_couse\_id=cj.stu\_couse\_id and cj.fen>60 order by=;

select \* from stu\_couse;

--集合运算  
--选择了课程3的学生   union 选择了课程5的学生   并集  
--选择了课程3 或者 选择了课程5的学生  
select s.stu\_name from students s,couse c,stu\_couse sc where s.id=sc.stu\_id and sc.couse\_id=c.couse\_id and c.couse\_id=3  
union  
select s.stu\_name from students s,couse c,stu\_couse sc where s.id=sc.stu\_id and sc.couse\_id=c.couse\_id and c.couse\_id=5

--选择了课程3，5，2 的学生 intersect 选择课程1，2，4的学生    交集  
--求选择了课程 2 并且 选择了课程 3 的学生   交集  
select s.stu\_name from students s,couse c,stu\_couse sc where s.id=sc.stu\_id and sc.couse\_id=c.couse\_id and c.couse\_id=2  
intersect  
select s.stu\_name from students s,couse c,stu\_couse sc where s.id=sc.stu\_id and sc.couse\_id=c.couse\_id and c.couse\_id=3;

--选择了课程3，5，8的学生   minus 选择了课程1，7，8的学生   --差集  
-- 求所有课程的成绩都大于 60 的学生 差集  
select distinct(s.stu\_name) from stu\_couse sc,students s,couse c,chengji cj where sc.stu\_id=s.id and sc.couse\_id=c.couse\_id and sc.stu\_couse\_id=cj.stu\_couse\_id and cj.fen>60  
minus  
select distinct(s.stu\_name) from stu\_couse sc,students s,couse c,chengji cj where sc.stu\_id=s.id and sc.couse\_id=c.couse\_id and sc.stu\_couse\_id=cj.stu\_couse\_id and cj.fen<60;