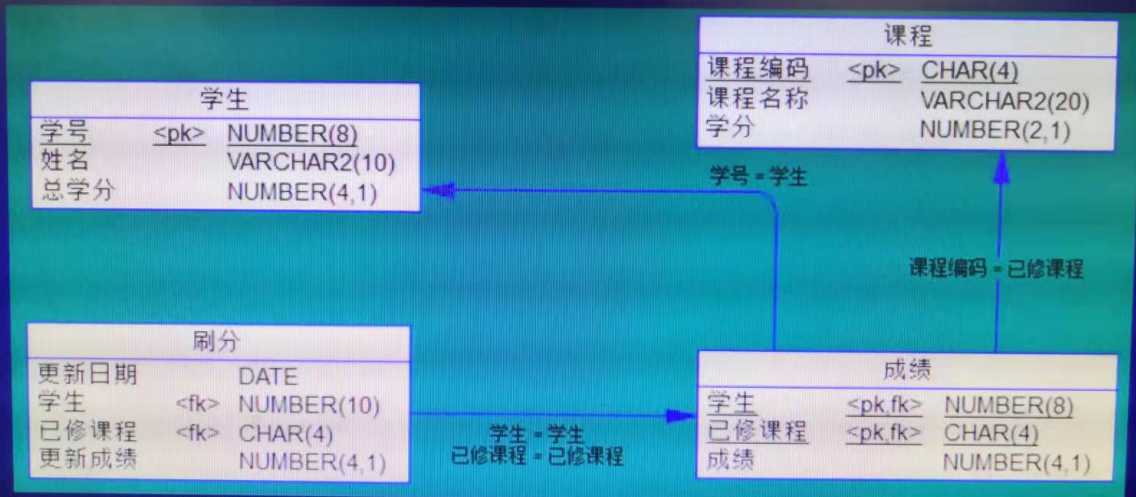


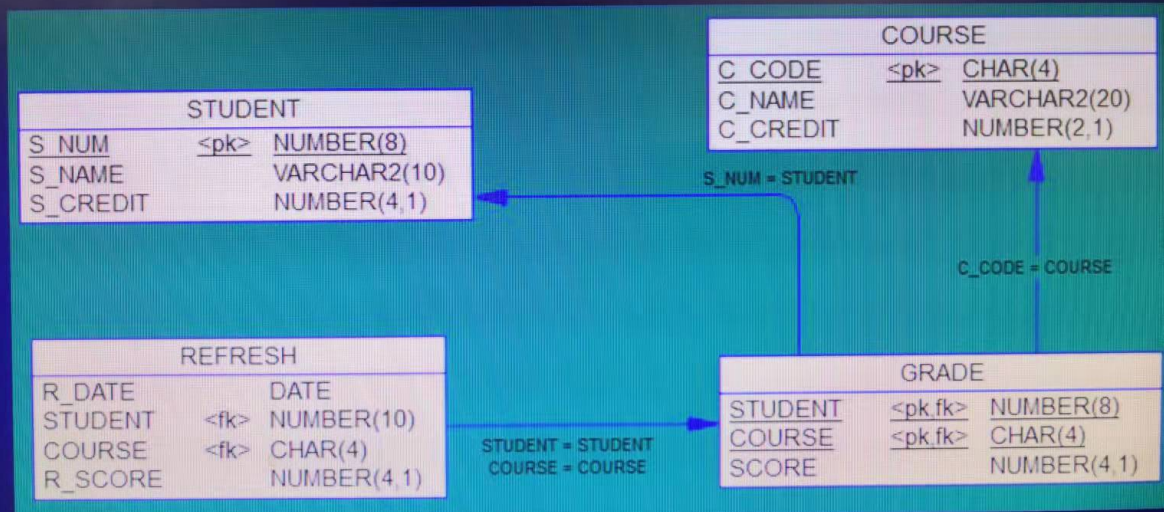
简化的学生成绩管理模型



ORACLE

综合练习三

简化的学生成绩管理模型



ORACLE

# 类比表结构SQL代码题

## 必不考题

序号	操作要求(困难版，一会全会) 大体看看就行，全是查询语句，不能全考这，毕竟要涉及 <b>12条SQL</b> 语句
0	创建学生刷分表REFRESH并关联外键FK (其他的表类似)  <pre>creat table REFRESH(r_date date,student number(10),course char(4),r_score number(4,1),foreign key(student) refenence GRADE(student),foreign key(course) refenence GRADE(course))</pre>
1	按照“学号”，“姓名”，“课程名称”，“成绩”的格式给出学生的成绩信息  <pre>select s.num as 学号, s.name as 姓名, c.name as 课程名称, g.score as 成绩 from STUDENT s,COURSE c,GARDE g where g.student = s.s_num and g.course = c.c_code</pre>
2	求出每门课程的平均成绩，并按照“课号”，“课程名称“，”平均成绩“的格式显示结果（按课号排序）  <pre>select g.course 课号,c.c_name 课程名称,g.ac 平均成绩 from (select course,trunc(avg(score), 1)) ac from grade group by course) g, course c where g.course=c.c_code</pre>
3	查询平均成绩在70分以下的所有学生，按照”学号“，”姓名“，”平均成绩“的格式输出  <pre>select s.num,s.name,avg(g.score) from student s,grade g where avg(s.score)&lt;70 and s.num = g.studnet</pre>
4	查询每门课程成绩最高的同学，并按照”学生“、”课程“、”最高成绩“的格式输出  <pre>select * from grade g where score = (select max(score) from grade where course = g.course);</pre>
5	查询每门课程成绩最低的同学，并按照”学生“、”课程“、”最高成绩“的格式输出

序号      操作要求(困难版，一会全会)  
大体看看就行，全是查询语句，不能全考这，毕竟要涉及**12条SQL**语句

---

```
select * from grade g
where score = (select min(score) from grade where course =
g.course);
select * from grade g
where (course, score) in (select course,min(score) from
grade group by student)
```

---

6            查询那些课程成绩高于课程平均成绩的同学及其课程成绩

---

```
select * from grade g
where score>(select avg(score) from grade where course =
g.course)
```

---

7            根据成绩登记表中的成绩，查询每个同学总学分（成绩大于60分获得对应课程的学分）

---

```
select g.student,sum(c.c_credit)
from grade g,course c
where g.course = c.c_code and g.score>60 group by student
```

---

8            根据成绩登记表中的成绩，叫每个学生的总学分更新至学生表

---

```
update student s
set s_credit =(select sum(c.c_credit) from grade g,course c
where g.course = c.c_code and g.score>=60 and g.student =
s.s_sum);
```

---

9            查询平均成绩最高的课程（或学生）及其平均成绩

---

```
select course, avg(score)
from grade
having avg(score) = (select max(avg(score)) from grade
group by course))
```

---

10           指定课程编号，给出该课程成绩最高的三位同学，并按”学生“，”成绩“的格式输出

---

```
select *
from (select * from grade where course =&crs order by
score)
where rownum<=3;
```

---

11           指定课程编号，给出成绩在第n到第m名之间的学生

---

序号	操作要求(困难版，一会全会)
	大体看看就行，全是查询语句，不能全考这，毕竟要涉及 <b>12条SQL语句</b>
	<pre>select * from (select rownum r, s.* from (select student, score from grade where course = \$crs order by score desc) s) where r&gt;=n and r&lt;=m;</pre>
12	给出所有课程成绩最高的三位同学（成绩处于前三名）的信息，按”课程“、“学生“、“成绩“的格式输出
	<pre>select * from (select course,student,score from grade order by score) where rownum&lt;=3;</pre>

## 必考题：基于12条SQL基本语句

序号	SQL语句	操作要求
<b>DDL</b>		
1	创建表 create	创建个带主键外键的表
	<pre>create table (数据 数据类型, constraint 主键名 primary key(字段名) foreign key(字段名) reference (另一个表名) (另一个表的 字段名))</pre>	
2	修改表 alter	增加删除字段（列），增加主，外键
	<pre>alter table 表名 add 字段名 数据类型 alter table 表名 add constraint (键名) primary key(字段名) alter table 表名 add constraint (键名) foreign key(本表字段) references (另一个表名) (另一个表的字段 名)</pre>	
3	删除数据 库对象 drop	没遇到过
<b>DML</b>		
4	查询 select	太多了，什么查询最高分，最低分，平均分，前几名的

序号	SQL语句	操作要求
5	更新表中的数据 update	根据成绩登记表中的成绩，叫每个学生的总学分更新至学生表
	<pre>update student s set s_credit =(select sum(c.c_credit) from grade g,course c where g.course = c.c_code and g.score&gt;=60 and g.student = s.s_sum);</pre>	
6	插入行 insert	加入一个新同学，直接加，或者把别表的数据插入这个表
	<pre>insert into 表名 values() insert into 表名 select 要从别的表添加的字段 from 另一个表 where 条件</pre>	
7	删除数据 delete	delete from 表名 where 条件
8	清空表 truncate	truncate table 表名
<b>DCL</b>		
9	提交 commit	commit
10	撤销 rollback	rollback
11	给权限 grant	grant 权限名 on 表名 to 用户名
12	收权限 revoke	revoke 权限名 on 表名 from 用户名

## 编写对应的PL/SQL语句

### 必考题

设计一个跟踪学生成绩变更的触发器Trigger

```
create table trial(
change_date date,
```

```

student number(8)
course char(6)
score1 number(4,1)
score2 number(4,1)
);

create or replace trigger score_change
after update of score on grade
for each row
begin
    insert into trail
values(sysdate,:new.student,:new.course,:old.score,:new.score);
end;

```

编写一个计算学生学分的**函数Function**（已知学号，计算总学分）

```

create or replace function get_credit(s_num number)
return number
as credit number(5, 1);
begin
    select sum(c_credit) into credits from grade g,course c
    where g.course = c.c_code
    and g.score>60
    and g.student=s_num;
    return nvl(credits, 0);
exception
    when no_data_found then
        dbms_output.put_line("没有此学生的成绩信息。");
end;

```

编写一个更新学分的**存储过程procedure**

```

create or replace procedure upd_credits
as
cursor c_stu is select * from student for update;
stu c_stu%rowtype;
begin
    open c_stu
    loop
        fetch c_stu into stu;
        exit when c_stu%notfound;
        update student set s_credit=get_credit(stu.s_num)
        where current of c_stu;
    end loop;
    commit;
    close c_stu;
end;

```

# 参数和数据字典题

## 列举3~5个数据字典视图

1. USER\_USER: 主要描述当前用户的信息
2. USER\_TABLES: 主要描述当前用户拥有的所有表的信息
3. USER\_OBJECTS: 记录了当前用户的所有对象的信息
4. USER\_TAB\_PRIVS: 该视图主要是存储当前用户下对所有表的权限信息
5. USER\_SOURCE :包含了系统中对象的原码

## 列举3~5个参数

1. AUDIT\_FILE\_DEST: 定义Oracle存储审计文件的路径
2. DB\_FILES: 设置数据库最大文件个数
3. SESSIONS: 设置系统可创建的最大会话数
4. SGA\_MAX\_SIZE: 用于设置实例的SGA的大小
5. OPEN\_CURSORS: 设置一个会话最多可以同时打开多少游标

声明：必不必考纯属虚构，仅供学习参考，其他用途与作者无关。