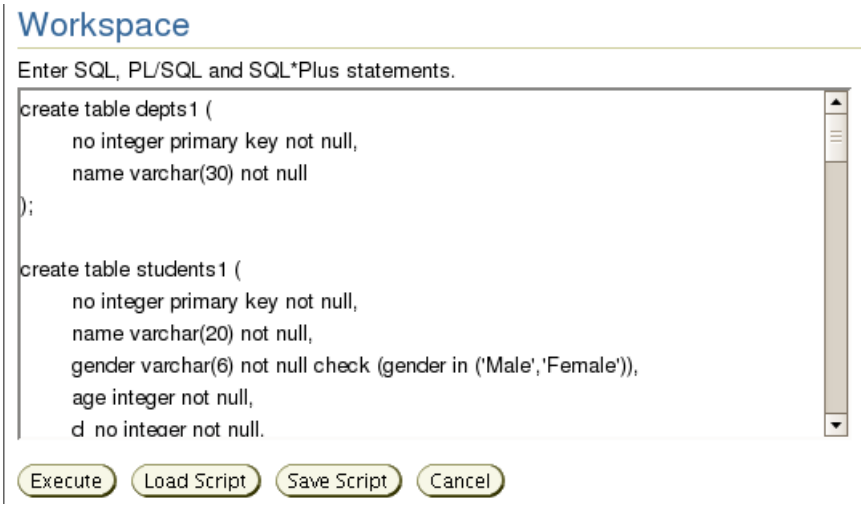
《数据库系统原理》实验报告（三）					
题目：上机实验课（三）交互式 SQL（DML）					
学号	1853790	姓名	庄镇华	日期	2020.10.27

实验环境：VMware 虚拟机 Red Hat5 系统下的 oracle 环境

实验步骤及结果截图：

1. 建立四表并查询表结构



students1 表

NO	NAME	GENDER	AGE	D_NO
200215120	Mike	Male	21	3
200215121	Tom	Male	20	1
200215122	Jerry	Female	19	1
200215123	Alice	Female	18	2
200215125	Bob	Male	19	3

courses1 表

NO	NAME	CREDIT	D_NO
1	Database	5	1
2	Mathematics	2	2
3	Information System	1	4
4	Operating System	6	1
5	Data Structure	4	1
6	Data Processing	2	4
7	PASCAL	3	1

depts1 表

NO	NAME
1	Computer Science
2	Mathematics
3	Architecture
4	Management

scores1 表

S_NO	C_NO	SCORE
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

2. 查所有年龄在 20 岁以下的学生姓名及其年龄（使用比较运算符）

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select name, age
from students1
where age < 20;
```

Execute Load Script Save Script Cancel

NAME	AGE
Jerry	19
Alice	18
Bob	19

3. 查询选 2 号课程(s_no='2')且成绩在 80--90 的学生号。(BETWEEN … AND …)

Workspace

Enter SQL, PL/SQL and SQL*Plus statements. Clear

```
select s_no
from scores1
where c_no = '2' and score between 80 and 90;
```

Execute Load Script Save Script Cancel

S_NO
200215121
200215122

4. 查姓名第二个字母是'o'的学生姓名

Workspace

Enter SQL, PL/SQL and SQL*Plus statements. Clear

```
select name
from students1
where name like '_o%';
```

Execute Load Script Save Script Cancel

NAME
Tom
Bob

5. 查询全体男学生的学号、系、年龄结果按所在的系升序排列，同一系中的学生按年龄降序排列。

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select no, d_no, age
from students1
where gender = 'Male'
order by d_no asc, age desc;
```

Execute Load Script Save Script Cancel

NO	D_NO	AGE
200215121	1	20
200215120	3	21
200215125	3	19

6. 查询女学生的总人数和平均年龄。

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select count(distinct no), avg(age)
from students1
where gender = 'Female';
```

Execute Load Script Save Script Cancel

COUNT(DISTINCTNO)	AVG(AGE)
2	18.5

7. 查询选修 3 号课程并及格【分数大于 60】的学生的最高分数、最低分及总分。

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select max(score), min(score), sum(score)
from scores1
where c_no = '3' and score > 60;
```

Execute Load Script Save Script Cancel

MAX(SCORE)	MIN(SCORE)	SUM(SCORE)
88	80	168

8. 向 Score 表中插入一条记录（200215123, 1, 70）

Enter SQL, PL/SQL and SQL*Plus statements.

```
insert into scores1(s_no, c_no, score)
      values(200215123,1,70);
```

S_NO	C_NO	SCORE
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80
200215123	1	70

9. 求每个学生（号）的平均成绩，并将其超过 80 分【HAVING AVG(score) > 80】的按学号输出【ORDER BY s_no】。

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select s_no, avg(score)
from scores1
group by s_no
having avg(score) > 80
order by s_no;
```

Execute Load Script Save Script Cancel

S_NO	AVG(SCORE)
200215121	88.3333333
200215122	85

10. 查询选修了课程 1 或者选修了课程 2 的学生姓名

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
(select distinct name
from scores1, students1
where c_no = '1' and scores1.s_no = students1.no)
   union
(select distinct name
from scores1, students1
where c_no = '2' and scores1.s_no = students1.no);
```

Execute Load Script Save Script Cancel

NAME
Alice
Jerry
Tom

11. 查询既选修了课程 1 又选修了课程 2 的学生姓名【intersect】

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
(select distinct name
from scores1, students1
where c_no = '1' and scores1.s_no = students1.no)
   intersect
(select distinct name
from scores1, students1
where c_no = '2' and scores1.s_no = students1.no);
```

Execute Load Script Save Script Cancel

NAME
Tom

NO.10:

```
select name from students1
where no in ((select s_no from scores1 where c_no='1')
intersect (select s_no from scores1 where c_no='2'));
```

12. 查询选修 Database 这门课最高分学生所在的系名

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select depts1.name
from scores1, students1, depts1
where scores1.s_no = students1.no
    and students1.d_no = depts1.no
    and score = (select max(score) from scores1);
```

Execute Load Script Save Script Cancel

NAME
Computer Science

```
NO.11:
        select name from depts1
        where no in (select d_no from students1
        where no in (select s_no from scores1
        where score=(select max(score) from scores1
        where c_no =(select no from courses1 where name ='Database'))));
```

13. 建立一个包含学生学号，姓名，年龄，以及所在系名的视图（赋予列名为 sno, sname, sage, deptname）

【create view】

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
create view stu_dept(sno, sname, sage, deptname)
    as (select students1.no, students1.name, students1.age, depts1.name
        from students1, depts1
        where students1.d_no = depts1.no);
select * from stu_dept;
```

Execute | Load Script | Save Script | Cancel

View created.

SNO	SNAME	SAGE	DEPTNAME
200215120	Mike	21	Architecture
200215121	Tom	20	Computer Science
200215122	Jerry	19	Computer Science
200215123	Alice	18	Mathematics
200215125	Bob	19	Architecture

出现的问题：

1. 在查询选修了课程 1 或者选修了课程 2 的学生姓名的时候，希望使用 natural join 将 scores1 和 students1 表单 no 相同的内容连接起来，但是得到的却是笛卡尔积的结果。

解决方案：

1. 因为 scores1 中的属性名称是 s_no，而 students1 中的属性名称是 no，二者不同，所以 natural join 没有参照的属性，因此和笛卡尔积等同，这时可以通过 on 或者 where 语句解决问题。