《数据库系统原理》实验报告（2）					
题目：交互式 SQL (2)					
学号	2152118	姓名	史君宝	日期	2023.10.28

实验环境：基于 docker 的 mariadb 数据库的自己创建的 mariadb_demo 容器。

实验步骤及结果截图：

（1）建立数据库：

```
MariaDB [(none)]> create database University;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| University         |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.000 sec)
```

```
MariaDB [(none)]> use University;
Database changed
MariaDB [University]>
```

（2）建立数据表：

```
MariaDB [University]> create table depts(
    -> no int not null,
    -> name varchar(30) not null,
    -> primary key(no));
Query OK, 0 rows affected (0.008 sec)
```

```
MariaDB [University]> create table students1( no int not null, name varchar(20) not null,
gender varchar(6) not null, check(gender = 'Male' or gender = 'Female'), age int not null,
 d_no int not null, primary key(no), constraint st_c_1     foreign key(d_no)     reference
s depts(no));
Query OK, 0 rows affected (0.014 sec)
```

```
MariaDB [University]> create table courses1( no int not null, name varchar(20) not null, c
redit int not null, d_no int not null, primary key(no), constraint co_c_1   foreign key(d_
no)     references depts(no));
Query OK, 0 rows affected (0.012 sec)
```

```
MariaDB [University]> create table scores1( s_no int not null, c_no int not null, score in
t not null, constraint sc_c_1   foreign key(s_no)     references students1(no), constraint
 sc_c_2 foreign key(c_no) references courses1(no));
Query OK, 0 rows affected (0.014 sec)
```

展示：

```
MariaDB [University]> show tables;
+----------------------+
| Tables_in_University |
+----------------------+
| courses1             |
| depts                |
| scores1              |
| students1            |
+----------------------+
4 rows in set (0.001 sec)
```

```
MariaDB [University]> desc courses1;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| no     | int(11)     | NO   | PRI | NULL    |       |
| name   | varchar(20) | NO   |     | NULL    |       |
| credit | int(11)     | NO   |     | NULL    |       |
| d_no   | int(11)     | NO   | MUL | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

（3）向数据表中导入数据

```
MariaDB [University]> insert into depts (no, name) values (1, 'Computer Science');
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into depts (no, name) values (2, 'Mathematics');
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into depts (no, name) values (3, 'Architecture');
Query OK, 1 row affected (0.001 sec)

MariaDB [University]> insert into depts (no, name) values (4, 'Management');
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [University]> select * from depts;
+----+------------------+
| no | name             |
+----+------------------+
|  1 | Computer Science |
|  2 | Mathematics      |
|  3 | Architecture     |
|  4 | Management       |
+----+------------------+
4 rows in set (0.001 sec)
```

```
MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (1,'Database',5,1)
;
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (2,'Mathematics',2
,2);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (3,'Information Sy
stem',1,4);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (4,'Operating Syst
em',6,1);
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (5,'Data Structure
',4,1);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (6,'Data Processin
g',2,4);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into courses1 (no,name,credit,d_no) values (7,'PASCAL',3,1);
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [University]> select * from courses1;
+----+--------------------+--------+------+
| 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 |
+----+--------------------+--------+------+
7 rows in set (0.000 sec)
```

```
MariaDB [University]> insert into students1(no,name,gender,age,d_no) values (200215120,'Mi
ke','Male',21,3);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into students1(no,name,gender,age,d_no) values (200215121,'To
m','Male',20,1);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into students1(no,name,gender,age,d_no) values (200215122,'Je
rry','Female',19,1);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into students1(no,name,gender,age,d_no) values (200215123,'Al
ice','Female',18,2);
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [University]> insert into students1(no,name,gender,age,d_no) values (200215124,'Bo
b','Male',19,3);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> select * from students1;
+-----------+-------+--------+-----+------+
| 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 |
| 200215124 | Bob   | Male   |  19 |    3 |
+-----------+-------+--------+-----+------+
5 rows in set (0.000 sec)
```

```
MariaDB [University]> insert into scores1(s_no,c_no,score) values (200215121,1,92);
Query OK, 1 row affected (0.004 sec)

MariaDB [University]> insert into scores1(s_no,c_no,score) values (200215121,2,85);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into scores1(s_no,c_no,score) values (200215121,3,88);
Query OK, 1 row affected (0.001 sec)

MariaDB [University]> insert into scores1(s_no,c_no,score) values (200215122,2,90);
Query OK, 1 row affected (0.002 sec)

MariaDB [University]> insert into scores1(s_no,c_no,score) values (200215122,3,80);
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [University]> select * from scores1;
+-----------+------+-------+
| s_no      | c_no | score |
+-----------+------+-------+
| 200215121 |    1 |    92 |
| 200215121 |    2 |    85 |
| 200215121 |    3 |    88 |
| 200215122 |    2 |    90 |
| 200215122 |    3 |    80 |
+-----------+------+-------+
5 rows in set (0.000 sec)
```

（4）NO.1 查所有年龄在 21 岁以下的学生姓名及其年龄（使用比较运算符）：

```
MariaDB [University]> select name,age from students1 where age<=21;
+-------+-----+
| name  | age |
+-------+-----+
| Mike  |  21 |
| Tom   |  20 |
| Jerry |  19 |
| Alice |  18 |
| Bob   |  19 |
+-------+-----+
5 rows in set (0.001 sec)
```

（5）NO.2 查询选 2 号课程(s_no='2')且成绩在 80--90 的学生号。(BETWEEN … AND …)

```
MariaDB [University]> select s_no, c_no, score from scores1 where c_no=2 and score between 80 and 90;
+-----------+------+-------+
| s_no      | c_no | score |
+-----------+------+-------+
| 200215121 |    2 |    85 |
| 200215122 |    2 |    90 |
+-----------+------+-------+
2 rows in set (0.000 sec)
```

（6）NO.3 查姓名第二个字母是'e'的学生姓名

```
MariaDB [University]> select name from students1 where name like '_e%';
+-------+
| name  |
+-------+
| Jerry |
+-------+
1 row in set (0.001 sec)
```

（7）NO.4 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。

```
MariaDB [University]> select s.no, d.name,s.age from students1 as s join depts as d on s.d_no = d.no  where s.gender = 'Male' order by
d.name ASC,s.age DESC;
+-----------+------------------+-----+
| no        | name             | age |
+-----------+------------------+-----+
| 200215120 | Architecture     |  21 |
| 200215124 | Architecture     |  19 |
| 200215121 | Computer Science |  20 |
+-----------+------------------+-----+
3 rows in set (0.001 sec)
```

（8）NO.5 查询女学生的总人数和平均年龄。

```
MariaDB [University]> select count(no) as total_count, avg(age) as avg_age from students1 where gender = 'Female';
+-------------+---------+
| total_count | avg_age |
+-------------+---------+
|           2 | 18.5000 |
+-------------+---------+
1 row in set (0.001 sec)
```

（9）NO.6　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。

```
MariaDB [University]> select max(score) as max_score, min(score) as min_score, sum(score) from scores1 where c_no = 3 and score>60;
+-----------+-----------+------------+
| max_score | min_score | sum(score) |
+-----------+-----------+------------+
|        88 |        80 |        168 |
+-----------+-----------+------------+
1 row in set (0.001 sec)
```

（10）NO.7　向 Score 表中插入一条记录（200215123,1,72）

```
MariaDB [University]> insert into scores1(s_no, c_no, score) values (200215123, 1, 72)
    -> ;
Query OK, 1 row affected (0.005 sec)
```

（11）NO.8 求每个学生(号)的平均成绩,并将其超过 75 分【HAVING AVG(score) > 75】的按学号输出【ORDER BY s_no】。

```
MariaDB [University]> select s_no, avg(score) as avg_score from scores1 group by s_no having avg(score)>75 order by s_no;
+-----------+-----------+
| s_no      | avg_score |
+-----------+-----------+
| 200215121 |   88.3333 |
| 200215122 |   85.0000 |
+-----------+-----------+
2 rows in set (0.001 sec)
```

（12）NO.9 查询选修了课程 1 或者选修了课程 2 的学生姓名

```
MariaDB [University]> select name from students1 where no in (select s_no from scores1 where c_no = 1 or c_no = 2);
+-------+
| name  |
+-------+
| Tom   |
| Alice |
| Jerry |
+-------+
3 rows in set (0.001 sec)
```

（13）NO.10 查询既选修了课程 1 又选修了课程 2 的学生姓名【mysql 模拟 intersect: 用 DISTINCT,INNER JOIN 或 DISTINCT,WHERE 等方式，可以实现交集操作即可】

```
MariaDB [University]> select distinct name from students1 as s inner join scores1 as sc on s.no = sc.s_no and (sc.c_no = 1 or sc.c_no =
2);
+-------+
| name  |
+-------+
| Tom   |
| Alice |
| Jerry |
+-------+
3 rows in set (0.001 sec)
```

（14）NO.11 查询选修 Database 这门课最高分学生所在的系名

```
MariaDB [University]> select d.name as d_name from depts as d inner join students1 as s on d.no = s.d_no inner join scores1 as sc on s.
no = sc.s_no inner join courses1 as c on sc.c_no = c.no  where c.name = 'Database' order by sc.score DESC limit 1;
+------------------+
| d_name           |
+------------------+
| Computer Science |
+------------------+
1 row in set (0.001 sec)
```

（15）NO.12 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】

```
MariaDB [University]> create view student_info as select s.no as s_no, s.name as s_name, s.age as s_age, d.name as d_name from students
1 as s join depts as d on s.d_no = d.no;
Query OK, 0 rows affected (0.016 sec)
```

```
MariaDB [University]> select * from student_info;
+-----------+--------+-------+------------------+
| s_no      | s_name | s_age | d_name           |
+-----------+--------+-------+------------------+
| 200215120 | Mike   |    21 | Architecture     |
| 200215121 | Tom    |    20 | Computer Science |
| 200215122 | Jerry  |    19 | Computer Science |
| 200215123 | Alice  |    18 | Mathematics      |
| 200215124 | Bob    |    19 | Architecture     |
+-----------+--------+-------+------------------+
5 rows in set (0.002 sec)
```

出现的问题：

在整个实验过程中并未出现比较大的问题，主要问题就是对于一些内容不太确定，比如在其中 join 中的知识点就有很多，这些在上课并未学到，用起来不是特别的得心应手，需要查找一下资料才能进一步完成。比如自然连接和内连接等等，还需要进一步的学习。

解决方案：

以上出现的问题都不是比较严重的问题，通过在网上查找资料，比如 CSDN、百度等众多的学习工具，就可以顺利的解决了。