《数据库系统原理》实验报告（2）					
题目：交互式 SQL（2）					
学号		姓名		日期	2023.10.17

实验环境：

Docker MariaDB

处理器：11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz　　3.11 GHz

实验步骤及结果截图：

1. 建立 table

```
Query OK, 0 rows affected (0.009 sec)

MariaDB [db_1024]> create table depts1 (
    ->   no integer not null auto_increment,
    ->   name varchar(30) not null,
    ->   primary key (no)
    -> );
Query OK, 0 rows affected (0.008 sec)

MariaDB [db_1024]> create table students1 (
    ->   no integer not null auto_increment,
    ->   name varchar(20) not null,
    ->   gender varchar(6) not null check (gender='Male' or gender='Female'),
    ->   age integer not null,
    ->   d_no integer not null,
    ->   primary key (no),
    ->   constraint st_c_1 foreign key (d_no) references depts1(no)
    -> );
Query OK, 0 rows affected (0.014 sec)

MariaDB [db_1024]> create table courses1 (
    ->   no integer not null auto_increment,
    ->   name varchar(20) not null,
    ->   credit integer not null,
    ->   d_no integer not null,
    ->   primary key (no),
    ->   constraint co_c_1 foreign key (d_no) references depts1(no)
    -> );
Query OK, 0 rows affected (0.013 sec)

MariaDB [db_1024]> create table scores1 (
    ->   s_no integer not null auto_increment,
    ->   c_no integer not null,
    ->   score integer 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 (0.013 sec)
```

2. 插入表单项

2.1 插入 depts1 表单项

```
MariaDB [db_1024]> insert into depts1 (no,name) values (1, 'Computer Science');
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into depts1 (no,name) values (2, 'Mathematics');
Query OK, 1 row affected (0.003 sec)

MariaDB [db_1024]> insert into depts1 (no,name) values (3, 'Architecture');
Query OK, 1 row affected (0.002 sec)

MariaDB [db_1024]> insert into depts1 (no,name) values (4, 'Management');
Query OK, 1 row affected (0.002 sec)
```

2.2 插入 course1 表单项

```
MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (1, 'DataBase', 5, 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (2, 'Mathematics', 2, 2);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (3, 'Information System', 1,
 4);
Query OK, 1 row affected (0.003 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (4, 'Operating System', 6, 1
);
Query OK, 1 row affected (0.003 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (5, 'Data Structure', 4, 1);

Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (6, 'Data Processing', 2, 4)
;
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into courses1 (no, name, credit, d_no) values (7, 'PASCAL', 3, 1);
Query OK, 1 row affected (0.001 sec)
```

2.3 插入 students1 表单项

```
MariaDB [db_1024]> insert into students1 (no, name, gender, age, d_no) values (200215120, 'Mike', 'Male', 21, 3);
Query OK, 1 row affected (0.003 sec)

MariaDB [db_1024]> insert into students1 (no, name, gender, age, d_no) values (200215121, 'Tom', 'Male', 20, 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into students1 (no, name, gender, age, d_no) values (200215122, 'Jerry', 'Female', 19, 1);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into students1 (no, name, gender, age, d_no) values (200215123, 'Alice', 'Female', 18, 2);
Query OK, 1 row affected (0.002 sec)

MariaDB [db_1024]> insert into students1 (no, name, gender, age, d_no) values (200215125, 'Bob', 'Male', 19, 3);
Query OK, 1 row affected (0.001 sec)
```

2.4 插入 scores1 表单项

```
MariaDB [db_1024]> insert into scores1 (s_no, c_no, score) values(200215121, 1, 92);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into scores1 (s_no, c_no, score) values(200215121, 2, 85);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into scores1 (s_no, c_no, score) values(200215121, 3, 88);
Query OK, 1 row affected (0.003 sec)

MariaDB [db_1024]> insert into scores1 (s_no, c_no, score) values(200215122, 2, 90);
Query OK, 1 row affected (0.001 sec)

MariaDB [db_1024]> insert into scores1 (s_no, c_no, score) values(200215122, 3, 80);
Query OK, 1 row affected (0.002 sec)
```

2.5 展示表项

depts1:

```
MariaDB [db_1024]> select * from depts1
    -> ;
+----+------------------+
| no | name             |
+----+------------------+
|  1 | Computer Science |
|  2 | Mathematics      |
|  3 | Architecture     |
|  4 | Management       |
+----+------------------+
4 rows in set (0.000 sec)
```

scores1:

```
MariaDB [db_1024]> 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)
```

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)
```

students1:

```
MariaDB [db_1024]> 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 |
| 200215125 | Bob   | Male   |  19 |    3 |
+-----------+-------+--------+-----+------+
5 rows in set (0.000 sec)
```

3. 问题一: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 [db_1024]> select name, age
    -> from students1
    -> where age < 21;
+-------+-----+
| name  | age |
+-------+-----+
| Tom   |  20 |
| Jerry |  19 |
| Alice |  18 |
| Bob   |  19 |
+-------+-----+
4 rows in set (0.000 sec)
```

4. 问题二:NO.2 查询选 2 号课程(c_no='2')且成绩在 80--90 的学生号。(BETWEEN … AND …)

```
MariaDB [db_1024]> select s_no
    -> from scores1
    -> where c_no = '2' and score between 80 and 90;
+-----------+
| s_no      |
+-----------+
| 200215121 |
| 200215122 |
+-----------+
2 rows in set (0.000 sec)
```

5. 问题三:NO.3 查姓名第二个字母是'e'的学生姓名

```
MariaDB [db_1024]> select name
    -> from students1
    -> where name like '_e%';
+-------+
| name  |
+-------+
| Jerry |
+-------+
1 row in set (0.000 sec)
```

6. 问题四: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 [db_1024]> select stu.no as student_ID, dept.name as department_name, stu.age as student_age
    -> from students1 as stu inner join depts1 as dept on stu.d_no = dept.no
    -> where stu.gender = 'Male'
    -> order by dept.name asc, stu.age desc;
+------------+-----------------+-------------+
| student_ID | department_name  | student_age |
+------------+-----------------+-------------+
|  200215120 | Architecture     |          21 |
|  200215125 | Architecture     |          19 |
|  200215121 | Computer Science |          20 |
+------------+-----------------+-------------+
3 rows in set (0.001 sec)
```

7. 问题五:NO.5 查询女学生的总人数和平均年龄.

```
MariaDB [db_1024]> select COUNT(gender), AVG(age)
    -> from students1
    -> where gender = 'Female';
+---------------+----------+
| COUNT(gender) | AVG(age) |
+---------------+----------+
|             2 |  18.5000 |
+---------------+----------+
1 row in set (0.000 sec)
```

8. 问题六: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 [db_1024]> select max(sco.score), min(sco.score), sum(sco.score)
    -> from students1 as stu inner join scores1 as sco on stu.no = sco.s_no
    -> where sco.c_no = '3' and sco.score > 60;
+----------------+----------------+----------------+
| max(sco.score) | min(sco.score) | sum(sco.score) |
+----------------+----------------+----------------+
|             88 |             80 |            168 |
+----------------+----------------+----------------+
1 row in set (0.001 sec)
```

9. 问题七”NO.7　向 Score 表中插入一条记录 200215123,1,72

```
MariaDB [db_1024]> INSERT INTO scores1 (s_no, c_no, score)
    -> VALUES (200215123, 1, 72);
Query OK, 1 row affected (0.002 sec)

MariaDB [db_1024]> select * from scores1
    -> ;
+-----------+------+-------+
| s_no      | c_no | score |
+-----------+------+-------+
| 200215121 |    1 |    92 |
| 200215121 |    2 |    85 |
| 200215121 |    3 |    88 |
| 200215122 |    2 |    90 |
| 200215122 |    3 |    80 |
| 200215123 |    1 |    72 |
+-----------+------+-------+
6 rows in set (0.000 sec)
```

10. 问题八: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 [db_1024]> 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 (0.000 sec)
```

11. 问题九:NO.9 查询选修了课程 1 或者选修了课程 2 的学生姓名

```
MariaDB [db_1024]> select name
    -> from students1
    -> where no in (
    ->   select distinct s_no
    ->   from scores1
    ->   where c_no = '1' or c_no = '2');
+-------+
| name  |
+-------+
| Tom   |
| Alice |
| Jerry |
+-------+
3 rows in set (0.001 sec)
```

12. 问题十: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 [db_1024]> select name
    -> from students1
    -> where no in (
    ->     select distinct s_no
    ->     from scores1
    ->     where c_no = '1' and s_no in (
    ->     select distinct s_no
    ->     from scores1
    ->     where c_no = '2'
    ->   )
    -> );
+------+
| name |
+------+
| Tom  |
+------+
1 row in set (0.001 sec)
```

13. 问题十一:NO.11 查询选修 Database 这门课最高分学生所在的系名

```
MariaDB [db_1024]> select dept.name
    -> from students1 as stu inner join depts1 as dept on stu.d_no = dept.no
    -> where stu.no in (
    ->   select s_no
    ->   from scores1
    ->   where c_no = '1' and score in (
    ->     select max(score)
    ->     from scores1
    ->     where c_no = '1'
    ->   )
    -> );
+------------------+
| name             |
+------------------+
| Computer Science |
+------------------+
1 row in set (0.001 sec)
```

14. 问题十二: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 [db_1024]> create view student_view as select s.no as sno, s.name as sname, s.age as sage, d
.name as deptname from students1 s inner join depts1 d on s.d_no=d.no;
Query OK, 0 rows affected (0.007 sec)

MariaDB [db_1024]> select * from student_view;
+-----------+-------+------+-----------------+
| 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    |
+-----------+-------+------+-----------------+
5 rows in set (0.001 sec)
```

出现的问题：

1. **Inner join 之后的表中有重复的表单项导致 ambiguous 的问题**

```
ERROR 1052 (23000): Column 'name' in field list is ambiguous
MariaDB [db_1024]> select stu.no, dept.name, stu.age
    -> from students1 as stu inner join depts1 as dept on students1.d_no = depts1.no
    -> where gender = 'Male';
```

解决方案：

1. **将 inner join 的两个表单起别名，通过别名来访问 inner join 之后的表单，可以有效避免在新表单当中 ambiguous 的问题。**

```
MariaDB [db_1024]> select stu.no as student_ID, dept.name as department_name, stu.age as student_age

    -> from students1 as stu join depts1 as dept on stu.d_no = dept.no
    -> where stu.gender = 'Male',
    -> order by dept.name asc, stu.age desc;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near '
order by dept.name asc, stu.age desc' at line 3
MariaDB [db_1024]> select stu.no as student_ID, dept.name as department_name, stu.age as student_age

    -> from students1 as stu inner join depts1 as dept on stu.d_no = dept.no
    -> where stu.gender = 'Male'
    -> order by dept.name asc, stu.age desc;
+------------+-----------------+-------------+
| student_ID | department_name | student_age |
+------------+-----------------+-------------+
|  200215120 | Architecture    |          21 |
|  200215125 | Architecture    |          19 |
|  200215121 | Computer Science |          20 |
+------------+-----------------+-------------+
3 rows in set (0.001 sec)
```