西安电子科技大学

《数据库原理》 课程实验报告

实验名称 SQL 实践

姓名_____ 学号_____

一、实验目的

熟练掌握 SQL 的数据定义功能，数据查询功能，数据操作功能，包括定义表，定义主码和外码，修改表(增加属性，删除属性，修改属性类型)，删除表，向表中添加数据记录，查询表中内容等。

二、实验环境

MySQL 8.0

三、实验基本原理及步骤

实验原理：SQL 基本语法

实验步骤：根据实验题目要求，写出对应的 SQL 语句实现对应操作

四、实验数据记录(SQL 语句，运行过程及结果)

1. 实验任务：

使用 SQL 语句建立高考志愿数据库 college_data(包括以下 3 个表)，添加主码信息(其中标红的属性为主码)。其中，sID, enrollment, sizeHS 类型是整型，GPA 类型是浮点数，decision 类型是文本，其他属性类型是变长字符串。

学院表 College (CName,state, enrollment) 属性列表示:学院名称，所在州，招生情况

学生表 Student (sID，sName，GPA，sizeHS) 属性列表示:学生学号，学生姓名，绩点，高中规模

申请表 Apply (sID，cName，major, decision) 属性列表示:学生学号，申请大学名称，申请专业，是否被录取

实验结果：

```
CREATE TABLE College(
        cName VARCHAR(50) NOT NULL,
    state VARCHAR(50) NOT NULL,
    enrollment INTEGER NOT NULL,
    PRIMARY KEY (cName)
);

CREATE TABLE Student(
        sID INTEGER NOT NULL,
    sName VARCHAR(50) NOT NULL,
    GPA FLOAT NOT NULL,
    sizeHS INTEGER NOT NULL,
    PRIMARY KEY (sID)
);
```

```
CREATE TABLE Apply(
        sID INTEGER NOT NULL,
    cName VARCHAR(50) NOT NULL,
    major VARCHAR(50) NOT NULL,
    decision VARCHAR(50) NOT NULL,
    PRIMARY KEY (sID, cName, major)
);
```

代码运行前：

```
mysql> show tables;
Empty set (0.00 sec)
```

代码运行后：

```
mysql> show tables;
+------------------------+
| Tables_in_college_data |
+------------------------+
| Apply                  |
| College                |
| Student                |
+------------------------+
3 rows in set (0.01 sec)

mysql> desc College;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| cName      | varchar(50) | NO   | PRI | NULL    |       |
| state      | varchar(50) | NO   |     | NULL    |       |
| enrollment | int         | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> desc Student;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| sID    | int         | NO   | PRI | NULL    |       |
| sName  | varchar(50) | NO   |     | NULL    |       |
| GPA    | float       | NO   |     | NULL    |       |
| sizeHS | int         | NO   |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc Apply;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| sID      | int         | NO   | PRI | NULL    |       |
| cName    | varchar(50) | NO   | PRI | NULL    |       |
| major    | varchar(50) | NO   | PRI | NULL    |       |
| decision | char(5)     | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

2. 实验任务：

对各表添加数据

实验结果：

```
INSERT INTO College VALUES('Berkeley', 'CA', 36000);
INSERT INTO College VALUES('Cornell', 'NY', 21000);
INSERT INTO College VALUES('MIT', 'MA', 10000);
INSERT INTO College VALUES('Stanford', 'CA', 15000);

INSERT INTO Student VALUES(123, 'Amy', 3.9, 1000);
INSERT INTO Student VALUES(234, 'Bob', 3.6, 1500);
INSERT INTO Student VALUES(345, 'Craig', 3.5, 500);
INSERT INTO Student VALUES(456, 'Doris', 3.9, 1000);
INSERT INTO Student VALUES(543, 'Craig', 3.4, 2000);
INSERT INTO Student VALUES(567, 'Edward', 2.9, 2000);
INSERT INTO Student VALUES(654, 'Amy', 3.9, 1000);
INSERT INTO Student VALUES(678, 'Fay', 3.8, 200);
INSERT INTO Student VALUES(765, 'Jay', 2.9, 1500);
INSERT INTO Student VALUES(789, 'Gary', 3.4, 800);
INSERT INTO Student VALUES(876, 'Irene', 3.9, 400);
INSERT INTO Student VALUES(987, 'Helen', 3.7, 800);

INSERT INTO Apply VALUES(123, 'Berkeley', 'CS', 'Y');
INSERT INTO Apply VALUES(123, 'Cornell', 'EE', 'Y');
INSERT INTO Apply VALUES(123, 'Stanford', 'CS', 'Y');
INSERT INTO Apply VALUES(123, 'Stanford', 'EE', 'N');
INSERT INTO Apply VALUES(234, 'Berkeley', 'biology', 'N');
INSERT INTO Apply VALUES(345, 'Cornell', 'bioengineering', 'N');
INSERT INTO Apply VALUES(345, 'Cornell', 'CS', 'Y');
INSERT INTO Apply VALUES(345, 'Cornell', 'EE', 'N');
```

```
INSERT INTO Apply VALUES(345, 'MIT', 'bioengineering', 'Y');
INSERT INTO Apply VALUES(543, 'MIT', 'CS', 'N');
INSERT INTO Apply VALUES(678, 'Stanford', 'history', 'Y');
INSERT INTO Apply VALUES(765, 'Cornell', 'history', 'N');
INSERT INTO Apply VALUES(765, 'Cornell', 'psychology', 'Y');
INSERT INTO Apply VALUES(765, 'Stanford', 'history', 'Y');
INSERT INTO Apply VALUES(876, 'MIT', 'biology', 'Y');
INSERT INTO Apply VALUES(876, 'MIT', 'marine biology', 'N');
INSERT INTO Apply VALUES(876, 'Stanford', 'CS', 'N');
INSERT INTO Apply VALUES(987, 'Berkeley', 'CS', 'Y');
INSERT INTO Apply VALUES(987, 'Stanford', 'CS', 'Y');
```

数据插入前：

```
mysql> select * from College;
Empty set (0.00 sec)

mysql> select * from Student;
Empty set (0.00 sec)

mysql> select * from Apply;
Empty set (0.00 sec)
```

数据插入后：

```
mysql> select * from College;
+----------+-------+------------+
| cName    | state | enrollment |
+----------+-------+------------+
| Berkeley | CA    |      36000 |
| Cornell  | NY    |      21000 |
| MIT      | MA    |      10000 |
| Stanford | CA    |      15000 |
+----------+-------+------------+
4 rows in set (0.00 sec)

mysql> select * from Student;
+-----+--------+-----+--------+
| sID | sName  | GPA | sizeHS |
+-----+--------+-----+--------+
| 123 | Amy    | 3.9 |   1000 |
| 234 | Bob    | 3.6 |   1500 |
| 345 | Craig  | 3.5 |    500 |
| 456 | Doris  | 3.9 |   1000 |
| 543 | Craig  | 3.4 |   2000 |
| 567 | Edward | 2.9 |   2000 |
| 654 | Amy    | 3.9 |   1000 |
| 678 | Fay    | 3.8 |    200 |
| 765 | Jay    | 2.9 |   1500 |
| 789 | Gary   | 3.4 |    800 |
| 876 | Irene  | 3.9 |    400 |
| 987 | Helen  | 3.7 |    800 |
+-----+--------+-----+--------+
12 rows in set (0.00 sec)
```

```
mysql> select * from Apply;
+-----+----------+----------------+----------+
| sID | cName    | major          | decision |
+-----+----------+----------------+----------+
| 123 | Berkeley | CS             | Y        |
| 123 | Cornell  | EE             | Y        |
| 123 | Stanford | CS             | Y        |
| 123 | Stanford | EE             | N        |
| 234 | Berkeley | biology        | N        |
| 345 | Cornell  | bioengineering | N        |
| 345 | Cornell  | CS             | Y        |
| 345 | Cornell  | EE             | N        |
| 345 | MIT      | bioengineering | Y        |
| 543 | MIT      | CS             | N        |
| 678 | Stanford | history        | Y        |
| 765 | Cornell  | history        | N        |
| 765 | Cornell  | psychology     | Y        |
| 765 | Stanford | history        | Y        |
| 876 | MIT      | biology        | Y        |
| 876 | MIT      | marine biology | N        |
| 876 | Stanford | CS             | N        |
| 987 | Berkeley | CS             | Y        |
| 987 | Stanford | CS             | Y        |
+-----+----------+----------------+----------+
19 rows in set (0.00 sec)
```

3. 实验任务：进行查询操作

(1)查询 GPA 大于 3.6 的学生学号和姓名

```
SELECT DISTINCT sID, sName FROM Student WHERE GPA > 3.6;
```

```
mysql> select sID, sName from Student where GPA > 3.6;
+-----+-------+
| sID | sName |
+-----+-------+
| 123 | Amy   |
| 456 | Doris |
| 654 | Amy   |
| 678 | Fay   |
| 876 | Irene |
| 987 | Helen |
+-----+-------+
6 rows in set (0.00 sec)
```

(2)查询所有学生姓名及申请专业

```
SELECT DISTINCT sName, major FROM Student JOIN Apply
WHERE Student.sID = Apply.sID;
```

```
mysql> select distinct sName, major from Student join Apply where Student.sID =
Apply.sID;
+-------+----------------+
| sName | major          |
+-------+----------------+
| Amy   | CS             |
| Amy   | EE             |
| Bob   | biology        |
| Craig | bioengineering |
| Craig | CS             |
| Craig | EE             |
| Fay   | history        |
| Jay   | history        |
| Jay   | psychology     |
| Irene | biology        |
| Irene | marine biology |
| Irene | CS             |
| Helen | CS             |
+-------+----------------+
13 rows in set (0.00 sec)
```

(3)查询所在高中规模不到 1000，申请了斯坦福大学 CS 专业的学生姓名、GPA 和申请结果

```
SELECT DISTINCT sName, GPA, decision FROM Student JOIN Apply
WHERE Student.sID = Apply.sID
AND sizeHS < 1000
AND cName = 'Stanford'
AND major = 'CS';
```

```
mysql> select distinct sName, GPA, decision from Student join Apply
    -> where Student.sID = Apply.sID
    -> and sizeHS < 1000
    -> and cName = 'Stanford'
    -> and major = 'CS';
+-------+-----+----------+
| sName | GPA | decision |
+-------+-----+----------+
| Irene | 3.9 | N        |
| Helen | 3.7 | Y        |
+-------+-----+----------+
2 rows in set (0.00 sec)
```

(4)查询具有 CS 专业、规模在 2000 人以上的学校名称

```
SELECT DISTINCT cName FROM College
WHERE enrollment >= 20000
AND cName IN (SELECT cName FROM Apply WHERE major = 'CS');
```

```
mysql> select distinct cName from College
    -> where enrollment >= 20000
    -> and cName in (select cName from Apply where major = 'CS');
+----------+
| cName    |
+----------+
| Berkeley |
| Cornell  |
+----------+
2 rows in set (0.01 sec)
```

(5)查询学生学号、姓名、绩点、申请学校，申请学校规模(按照申请学校分组，组内按绩点降序、学校规模升序排序)

```
SELECT DISTINCT Student.sID, Student.sName, Student.GPA, Apply.cName,
Student.sizeHS
FROM Student
JOIN Apply ON Student.sID = Apply.sID
GROUP BY Apply.cName, Student.sID, Student.sName, Student.GPA, Student.sizeHS
ORDER BY Apply.cName, Student.GPA DESC, Student.sizeHS;
```

```
mysql> select distinct Student.sID, Student.sName, Student.GPA, Apply.cName, Student.sizeHS
    -> from Student
    -> join Apply on Student.sID = Apply.sID
    -> group by Apply.cName, Student.sID, Student.sName, Student.GPA, Student.sizeHS
    -> order by Apply.cName, Student.GPA desc, Student.sizeHS;
+-----+-------+-----+----------+--------+
| sID | sName | GPA | cName    | sizeHS |
+-----+-------+-----+----------+--------+
| 123 | Amy   | 3.9 | Berkeley |   1000 |
| 987 | Helen | 3.7 | Berkeley |    800 |
| 234 | Bob   | 3.6 | Berkeley |   1500 |
| 123 | Amy   | 3.9 | Cornell  |   1000 |
| 345 | Craig | 3.5 | Cornell  |    500 |
| 765 | Jay   | 2.9 | Cornell  |   1500 |
| 876 | Irene | 3.9 | MIT      |    400 |
| 345 | Craig | 3.5 | MIT      |    500 |
| 543 | Craig | 3.4 | MIT      |   2000 |
| 876 | Irene | 3.9 | Stanford |    400 |
| 123 | Amy   | 3.9 | Stanford |   1000 |
| 678 | Fay   | 3.8 | Stanford |    200 |
| 987 | Helen | 3.7 | Stanford |    800 |
| 765 | Jay   | 2.9 | Stanford |   1500 |
+-----+-------+-----+----------+--------+
14 rows in set (0.00 sec)
```

(6)查询申请专业含有‘bio’字符的学生学号和申请专业

```
SELECT sID, major FROM Apply WHERE major LIKE '%bio%';
```

```
mysql> select sID, major from Apply where major like '%bio%';
+-----+----------------+
| sID | major          |
+-----+----------------+
| 234 | biology        |
| 345 | bioengineering |
| 345 | bioengineering |
| 876 | biology        |
| 876 | marine biology |
+-----+----------------+
5 rows in set (0.00 sec)
```

(7)查询具有相同绩点的学生信息对，输出他们的学号、姓名、GPA(S1.sID, S1.sName, S1.GPA, S2.sID, S2.sName, S2.GPA)

```
SELECT DISTINCT s1.sID, s1.sName, s1.GPA, s2.sID, s2.sName, s2.GPA
FROM Student s1 JOIN Student s2
WHERE s1.GPA = s2.GPA AND s1.sID < s2.sID;
```

```
mysql> select distinct s1.sID, s1.sName, s1.GPA, s2.sID, s2.sName, s2.GPA
    -> from Student s1 join Student s2
    -> where s1.GPA = s2.GPA and s1.sID < s2.sID;
+-----+--------+-----+-----+-------+-----+
| sID | sName  | GPA | sID | sName | GPA |
+-----+--------+-----+-----+-------+-----+
| 123 | Amy    | 3.9 | 456 | Doris | 3.9 |
| 123 | Amy    | 3.9 | 654 | Amy   | 3.9 |
| 123 | Amy    | 3.9 | 876 | Irene | 3.9 |
| 456 | Doris  | 3.9 | 654 | Amy   | 3.9 |
| 456 | Doris  | 3.9 | 876 | Irene | 3.9 |
| 543 | Craig  | 3.4 | 789 | Gary  | 3.4 |
| 567 | Edward | 2.9 | 765 | Jay   | 2.9 |
| 654 | Amy    | 3.9 | 876 | Irene | 3.9 |
+-----+--------+-----+-----+-------+-----+
8 rows in set (0.01 sec)
```

(8)查询同时申请了 CS 和 EE 专业的学生学号

```
SELECT DISTINCT sID FROM Apply AS A1
WHERE major = 'CS' AND EXISTS(
        SELECT * FROM Apply AS A2
    WHERE A1.sID = A2.sID AND A2.major = 'EE'
```

```
);
```

```
mysql> select distinct sID from Apply as A1
    -> where major = 'CS' and exists(
    -> select * from Apply as A2
    ->     where A1.sID = A2.sID and A2.major = 'EE'
    -> );
+-----+
| sID |
+-----+
| 123 |
| 345 |
+-----+
2 rows in set (0.00 sec)
```

(9) 查询申请了 CS 专业但是没有申请 EE 专业的学生学号的姓名

```
SELECT DISTINCT sID, sName FROM Student
WHERE EXISTS(
        SELECT * FROM Apply
    WHERE Student.sID = Apply.sID AND major = 'CS'
)AND NOT EXISTS(
        SELECT * FROM Apply
    WHERE Student.sID = Apply.sID AND major = 'EE'
);
```

```
mysql> select distinct sID, sName from Student
    -> where exists(
    -> select * from Apply
    ->     where Student.sID = Apply.sID and major = 'CS'
    -> )and not exists(
    -> select * from Apply
    ->     where Student.sID = Apply.sID and major = 'EE'
    -> );
+-----+-------+
| sID | sName |
+-----+-------+
| 543 | Craig |
| 876 | Irene |
| 987 | Helen |
+-----+-------+
3 rows in set (0.00 sec)
```

(10) 查询申请了 CS 专业的学生学号的姓名

```
SELECT DISTINCT sID, sName FROM Student
WHERE EXISTS(
```

```
        SELECT * FROM Apply
    WHERE Student.sID = Apply.sID AND major = 'CS'
);
```

```
mysql> select distinct sID, sName from Student
    -> where exists(
    -> select * from Apply
    ->     where Student.sID = Apply.sID and major = 'CS'
    -> );
+-----+-------+
| sID | sName |
+-----+-------+
| 123 | Amy   |
| 345 | Craig |
| 543 | Craig |
| 876 | Irene |
| 987 | Helen |
+-----+-------+
5 rows in set (0.00 sec)
```

(11) 查询所在州有其他学校的学校名称和所在州

```
SELECT DISTINCT cName, state FROM College
WHERE state IN(
        SELECT state FROM College
    GROUP BY state HAVING COUNT(*) > 1
);
```

```
mysql> select distinct cName, state from College
    -> where state in(
    -> select state from College
    ->     group by state having count(*) > 1
    -> );
+----------+-------+
| cName    | state |
+----------+-------+
| Berkeley | CA    |
| Stanford | CA    |
+----------+-------+
2 rows in set (0.00 sec)
```

(12) 查询规模最大的学校的名称

```
SELECT cName FROM College
```

```
WHERE enrollment = (
        SELECT MAX(enrollment) FROM College
);
```

```
mysql> select cName from College
    -> where enrollment = (
    -> select max(enrollment) from College
    -> );
+----------+
| cName    |
+----------+
| Berkeley |
+----------+
1 row in set (0.00 sec)
```

(13)查询绩点最高的学生姓名和 GPA（不用子查询）

```
SELECT sName, GPA FROM Student
ORDER BY GPA DESC
LIMIT 1;
```

```
mysql> select sName, GPA from Student
    -> order by GPA desc
    -> limit 1;
+-------+-----+
| sName | GPA |
+-------+-----+
| Amy   | 3.9 |
+-------+-----+
1 row in set (0.00 sec)
```

(14)查询不是来自规模最小的高中的学生学号、姓名和高中规模

```
SELECT DISTINCT sID, sName, sizeHS FROM Student
WHERE sizeHS <> (
        SELECT MIN(sizeHS) FROM Student
);
```

```
mysql> select distinct sID, sName, sizeHS from Student
    -> where sizeHS <> (
    -> select min(sizeHS) from Student
    -> );
+-----+--------+--------+
| sID | sName  | sizeHS |
+-----+--------+--------+
| 123 | Amy    |   1000 |
| 234 | Bob    |   1500 |
| 345 | Craig  |    500 |
| 456 | Doris  |   1000 |
| 543 | Craig  |   2000 |
| 567 | Edward |   2000 |
| 654 | Amy    |   1000 |
| 765 | Jay    |   1500 |
| 789 | Gary   |    800 |
| 876 | Irene  |    400 |
| 987 | Helen  |    800 |
+-----+--------+--------+
11 rows in set (0.01 sec)
```

(15)查询每个学校的学校名称、所在州、以及申请者中 GPA 最高的学生的绩点

```
SELECT C.cName, C.state, MAX(S.GPA) AS max_GPA
FROM College C
JOIN Apply A ON C.cName = A.cName
JOIN Student S ON A.sID = S.sID
GROUP BY C.cName, C.state
```

```
mysql> select C.cName, C.state, max(S.GPA) as max_GPA
    -> from College C
    -> join Apply A on C.cName = A.cName
    -> join Student S on A.sID = S.sID
    -> group by C.cName, C.state
    -> ;
+----------+-------+---------+
| cName    | state | max_GPA |
+----------+-------+---------+
| Berkeley | CA    |     3.9 |
| Cornell  | NY    |     3.9 |
| Stanford | CA    |     3.9 |
| MIT      | MA    |     3.9 |
+----------+-------+---------+
4 rows in set (0.00 sec)
```

(16)查询学生姓名和他们申请的专业

```
SELECT DISTINCT S.sName, A.major FROM Student S
JOIN Apply A ON S.sID = A.sID;
```

```
mysql> select distinct S.sName, A.major from Student S
    -> join Apply A on S.sID = A.sID;
+-------+----------------+
| sName | major          |
+-------+----------------+
| Amy   | CS             |
| Amy   | EE             |
| Bob   | biology        |
| Craig | bioengineering |
| Craig | CS             |
| Craig | EE             |
| Fay   | history        |
| Jay   | history        |
| Jay   | psychology     |
| Irene | biology        |
| Irene | marine biology |
| Irene | CS             |
| Helen | CS             |
+-------+----------------+
13 rows in set (0.00 sec)
```

(17)查询学生姓名和成绩绩点（条件是申请了 Stanford 大学的 CS 专业，所在高中人数少于 1000）

```
SELECT DISTINCT sName, GPA FROM Student
WHERE sizeHS < 1000
AND EXISTS(
    SELECT * FROM Apply
    WHERE major = 'CS' AND Student.sID = Apply.sID
);
```

```
mysql> select distinct sName, GPA from Student
    -> where sizeHS < 1000
    -> and exists(
    ->     select * from Apply
    ->     where major = 'CS' and Student.sID = Apply.sID
    -> );
+-------+-----+
| sName | GPA |
+-------+-----+
| Craig | 3.5 |
| Irene | 3.9 |
| Helen | 3.7 |
+-------+-----+
3 rows in set (0.00 sec)
```

(18)查询申请 CS 专业的学生的最低成绩绩点

```sql
SELECT MIN(GPA) AS min_GPA FROM Student S
JOIN Apply A ON S.sID = A.sID
WHERE A.major = 'CS';
```

```
mysql> select min(GPA) as min_GPA from Student S
    -> join Apply A on S.sID = A.sID
    -> where A.major = 'CS';
+---------+
| min_GPA |
+---------+
|     3.4 |
+---------+
1 row in set (0.00 sec)
```

(19)查询申请 CS 专业的学生的平均成绩绩点（不管他们申请多少次）

```sql
SELECT avg(DISTINCT GPA) AS avg_GPA FROM Student S
JOIN Apply A ON S.sID = A.sID
WHERE A.major = 'CS';
```

```
mysql> select avg(distinct GPA) as avg_GPA from Student S
    -> join Apply A on S.sID = A.sID
    -> where A.major = 'CS';
+--------------------+
| avg_GPA            |
+--------------------+
| 3.6250000596046448 |
+--------------------+
1 row in set (0.00 sec)
```

(20)查询申请 Cornell 大学的申请数（申请不同专业的同一个学生算一次申请）

```sql
SELECT COUNT(DISTINCT A.sID) AS application_num
FROM Apply A JOIN College C ON A.cName = C.cName
WHERE C.cName = 'Cornell';
```

```
mysql> select count(distinct A.sID) as application_num
    -> from Apply A join College C on A.cName = C.cName
    -> where C.cName = 'Cornell';
+-----------------+
| application_num |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.00 sec)
```

(21)查询学生信息（条件是与他具有相同 GPA 的学生人数等于与他所在高中规模相同的学生人数）

```
SELECT S.sID, S.sName, S.GPA, S.sizeHS FROM Student S
JOIN(
        SELECT GPA, COUNT(DISTINCT sID) AS cnt1
    FROM Student
    GROUP BY GPA
)T1 ON S.GPA = T1.GPA
JOIN(
        SELECT sizeHS, COUNT(DISTINCT sID) AS cnt2
    FROM Student
    GROUP BY sizeHS
)T2 ON S.sizeHS = T2.sizeHS
WHERE T1.cnt1 = T2.cnt2;
```

```
mysql> select S.sID, S.sName, S.GPA, S.sizeHS from Student S
    -> join(
    -> select GPA, count(distinct sID) as cnt1
    ->     from Student
    ->     group by GPA
    -> )T1 on S.GPA = T1.GPA
    -> join(
    -> select sizeHS, count(distinct sID) as cnt2
    ->     from Student
    ->     group by sizeHS
    -> )T2 on S.sizeHS = T2.sizeHS
    -> where T1.cnt1 = T2.cnt2;
+-----+--------+-----+--------+
| sID | sName  | GPA | sizeHS |
+-----+--------+-----+--------+
| 345 | Craig  | 3.5 |    500 |
| 543 | Craig  | 3.4 |   2000 |
| 567 | Edward | 2.9 |   2000 |
| 678 | Fay    | 3.8 |    200 |
| 765 | Jay    | 2.9 |   1500 |
| 789 | Gary   | 3.4 |    800 |
+-----+--------+-----+--------+
6 rows in set (0.00 sec)
```

(22) 查询申请 CS 专业的学生的平均绩点和申请非 CS 专业的学生的平均绩点的差值

```
SELECT avg(CASE WHEN major = 'CS' THEN GPA ELSE NULL END) AS avg_cs,
                avg(CASE WHEN major <> 'CS' THEN GPA ELSE NULL END) AS
avg_no_cs,
       avg(CASE WHEN major = 'CS' THEN GPA ELSE NULL END)-
       avg(CASE WHEN major <> 'CS' THEN GPA ELSE NULL END) AS diff
FROM Apply
JOIN Student ON Apply.sID = Student.sID;
```

```
mysql> select avg(case when major = 'CS' then GPA else NULL end) as avg_cs,
    ->
Display all 759 possibilities? (y or n)
    -> avg(case when major <> 'CS' then GPA else NULL end) as avg_no_cs,
    ->         avg(case when major = 'CS' then GPA else NULL end)-
    ->         avg(case when major <> 'CS' then GPA else NULL end) as diff
    -> from Apply
    -> join Student on Apply.sID = Student.sID;
+--------------------+--------------------+---------------------+
| avg_cs             | avg_no_cs          | diff                |
+--------------------+--------------------+---------------------+
| 3.7142857824053084 | 3.5166667103767395 | 0.19761907202856888 |
+--------------------+--------------------+---------------------+
1 row in set (0.00 sec)
```

(23) 查询每个大学的申请人数（同一个学生申请不同专业按照不同申请对待）

```
SELECT cName, COUNT(sID) AS application_num
FROM Apply
GROUP BY cName;
```

```
mysql> SELECT cName, COUNT(sID) AS application_num
    -> FROM Apply
    -> GROUP BY cName;
+----------+-----------------+
| cName    | application_num |
+----------+-----------------+
| Berkeley |               3 |
| Cornell  |               6 |
| Stanford |               6 |
| MIT      |               4 |
+----------+-----------------+
4 rows in set (0.00 sec)
```

(24) 查询所有大学每个专业申请人的最低 GPA 和最高 GPA 的最大差值

```
SELECT A.cName, A.major, TRUNCATE((T.maxGPA - T.minGPA),1) AS maxDiffGPA
FROM Apply A JOIN(
        SELECT cName, major, MIN(GPA) AS minGPA, MAX(GPA) AS maxGPA
    FROM Student S JOIN Apply A ON S.sID = A.sID
    GROUP BY cName, major
) T ON A.cName = T.cName AND A.major = T.major
GROUP BY A.cName, A.major
ORDER BY A.cName;
```

```
mysql> select A.cName, A.major, truncate((T.maxGPA - T.minGPA),1) as maxDiffGPA
    -> from Apply A join(
    -> select cName, major, min(GPA) as minGPA, max(GPA) as maxGPA
    ->     from Student S join Apply A on S.sID = A.sID
    ->     group by cName, major
    -> ) T on A.cName = T.cName and A.major = T.major
    -> group by A.cName, A.major
    -> order by A.cName;
+----------+----------------+------------+
| cName    | major          | maxDiffGPA |
+----------+----------------+------------+
| Berkeley | biology        |          0 |
| Berkeley | CS             |        0.2 |
| Cornell  | bioengineering |          0 |
| Cornell  | CS             |          0 |
| Cornell  | EE             |        0.4 |
| Cornell  | history        |          0 |
| Cornell  | psychology     |          0 |
| MIT      | bioengineering |          0 |
| MIT      | biology        |          0 |
| MIT      | CS             |          0 |
| MIT      | marine biology |          0 |
| Stanford | CS             |        0.2 |
| Stanford | EE             |          0 |
| Stanford | history        |        0.8 |
+----------+----------------+------------+
14 rows in set (0.00 sec)
```

(25) 查询每个学生申请的学校个数（包括没有申请任何学习哦啊的学生，输出学生学号和申请学校个数）

```
SELECT S.sID, COUNT(A.cName) AS application_num
FROM Student S LEFT JOIN Apply A ON S.sID = A.sID
GROUP BY S.sID;
```

```
mysql> select S.sID, count(A.cName) as application_num
    -> from Student S left join Apply A on S.sID = A.sID
    -> group by S.sID;
+-----+-----------------+
| sID | application_num |
+-----+-----------------+
| 123 |               4 |
| 234 |               1 |
| 345 |               4 |
| 456 |               0 |
| 543 |               1 |
| 567 |               0 |
| 654 |               0 |
| 678 |               1 |
| 765 |               3 |
| 789 |               0 |
| 876 |               3 |
| 987 |               2 |
+-----+-----------------+
12 rows in set (0.00 sec)
```

(26)查询申请者少于 5 的大学名称（申请者，不是申请数）

```sql
SELECT A.cName FROM Apply A
GROUP BY A.cName
HAVING COUNT(DISTINCT A.sID) < 5;
```

```
mysql> select A.cName from Apply A
    -> group by A.cName
    -> having count(distinct A.sID) < 5;
+----------+
| cName    |
+----------+
| Berkeley |
| Cornell  |
| MIT      |
+----------+
3 rows in set (0.00 sec)
```

(27)查询申请者最高 GPA 低于所有学生平均 GPA 的专业

```sql
SELECT A.major
FROM(
        SELECT avg(GPA) AS avg_GPA
    FROM Student
) AS S,(
        SELECT major, MAX(GPA) AS max_GPA
```

```
    FROM Apply, Student
    WHERE Apply.sID = Student.sID
    GROUP BY major
) AS A
WHERE A.max_GPA < S.avg_GPA;
```

```
mysql> select A.major
    -> from(
    -> select avg(GPA) as avg_GPA
    ->     from Student
    -> ) as S,(
    -> select major, max(GPA) as max_GPA
    ->     from Apply, Student
    ->     where Apply.sID = Student.sID
    ->     group by major
    -> ) as A
    -> where A.max_GPA < S.avg_GPA;
+----------------+
| major          |
+----------------+
| bioengineering |
| psychology     |
+----------------+
2 rows in set (0.00 sec)
```

4. 实验任务：进行数据修改操作

(1) 在 college 中插入一条数据，学校为'Carnegie Mellon'，所在州'PA'，入学人数 11500；

```
INSERT INTO College VALUES('Carnegie Mellon', 'PA', 11500);
```

插入前：

```
mysql> select * from College;
+----------+-------+------------+
| cName    | state | enrollment |
+----------+-------+------------+
| Berkeley | CA    |      36000 |
| Cornell  | NY    |      21000 |
| MIT      | MA    |      10000 |
| Stanford | CA    |      15000 |
+----------+-------+------------+
4 rows in set (0.00 sec)
```

插入后：

```
mysql> insert into College values('Carnegie Mellon', 'PA', 11500);
Query OK, 1 row affected (0.00 sec)

mysql> select * from College;
+-----------------+-------+------------+
| cName           | state | enrollment |
+-----------------+-------+------------+
| Berkeley        | CA    |      36000 |
| Carnegie Mellon | PA    |      11500 |
| Cornell         | NY    |      21000 |
| MIT             | MA    |      10000 |
| Stanford        | CA    |      15000 |
+-----------------+-------+------------+
5 rows in set (0.00 sec)
```

(2) 在 Apply 表中插入数据：没有申请任何学校的学生将申请 Carnegie Mellon 大学的 CS 专业；

将申请信息插入 apply 表中(decision 设置为空值)

```
INSERT INTO Apply(sID, cName, major, decision)
SELECT S.sID, 'Carnegie Mellon', 'CS', NULL
FROM Student S
WHERE NOT EXISTS(
        SELECT *
    FROM Apply A
    WHERE A.sID = S.sID
);
```

修改前：

```
mysql> select * from Apply;
+-----+----------+----------------+----------+
| sID | cName    | major          | decision |
+-----+----------+----------------+----------+
| 123 | Berkeley | CS             | Y        |
| 123 | Cornell  | EE             | Y        |
| 123 | Stanford | CS             | Y        |
| 123 | Stanford | EE             | N        |
| 234 | Berkeley | biology        | N        |
| 345 | Cornell  | bioengineering | N        |
| 345 | Cornell  | CS             | Y        |
| 345 | Cornell  | EE             | N        |
| 345 | MIT      | bioengineering | Y        |
| 543 | MIT      | CS             | N        |
| 678 | Stanford | history        | Y        |
| 765 | Cornell  | history        | N        |
| 765 | Cornell  | psychology     | Y        |
| 765 | Stanford | history        | Y        |
| 876 | MIT      | biology        | Y        |
| 876 | MIT      | marine biology | N        |
| 876 | Stanford | CS             | N        |
| 987 | Berkeley | CS             | Y        |
| 987 | Stanford | CS             | Y        |
+-----+----------+----------------+----------+
19 rows in set (0.00 sec)
```

修改后：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | CS             | NULL     |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | CS             | NULL     |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
23 rows in set (0.00 sec)
```

(3) 在 Apply 表中插入数据:允许所有申请 EE 专业被拒绝的学生进入卡内基梅隆大学 EE 专业(decision 值为‘Y’);

```sql
INSERT INTO Apply(sID, cName, major, decision)
SELECT sID, 'Carnegie Mellon', 'EE', 'Y'
FROM Apply
WHERE major = 'EE' AND decision = 'N';
```

修改前：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | CS             | NULL     |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | CS             | NULL     |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
23 rows in set (0.00 sec)
```

修改后：

```
mysql> select * from Apply
    -> ;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Carnegie Mellon | EE             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Carnegie Mellon | EE             | Y        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | CS             | NULL     |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | CS             | NULL     |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
25 rows in set (0.00 sec)
```

(4)从学生表中删除申请了 2 个以上专业的学生信息

```
DELETE FROM Student
WHERE sID IN(
        SELECT sID FROM Apply
```

```
    GROUP BY sID HAVING COUNT(DISTINCT major) > 2
);
```

修改前：

```
mysql> select * from Student;
+-----+--------+-----+--------+
| sID | sName  | GPA | sizeHS |
+-----+--------+-----+--------+
| 123 | Amy    | 3.9 |   1000 |
| 234 | Bob    | 3.6 |   1500 |
| 345 | Craig  | 3.5 |    500 |
| 456 | Doris  | 3.9 |   1000 |
| 543 | Craig  | 3.4 |   2000 |
| 567 | Edward | 2.9 |   2000 |
| 654 | Amy    | 3.9 |   1000 |
| 678 | Fay    | 3.8 |    200 |
| 765 | Jay    | 2.9 |   1500 |
| 789 | Gary   | 3.4 |    800 |
| 876 | Irene  | 3.9 |    400 |
| 987 | Helen  | 3.7 |    800 |
+-----+--------+-----+--------+
12 rows in set (0.00 sec)
```

修改后：

```
mysql> DELETE FROM Student
    -> WHERE sID IN(
    -> SELECT sID FROM Apply
    ->     GROUP BY sID HAVING COUNT(DISTINCT major) > 2
    -> );
Query OK, 2 rows affected (0.00 sec)

mysql> select * from Student;
+-----+--------+-----+--------+
| sID | sName  | GPA | sizeHS |
+-----+--------+-----+--------+
| 123 | Amy    | 3.9 |   1000 |
| 234 | Bob    | 3.6 |   1500 |
| 456 | Doris  | 3.9 |   1000 |
| 543 | Craig  | 3.4 |   2000 |
| 567 | Edward | 2.9 |   2000 |
| 654 | Amy    | 3.9 |   1000 |
| 678 | Fay    | 3.8 |    200 |
| 765 | Jay    | 2.9 |   1500 |
| 789 | Gary   | 3.4 |    800 |
| 987 | Helen  | 3.7 |    800 |
+-----+--------+-----+--------+
10 rows in set (0.00 sec)
```

(5)从 college 表中删除没有 CS 专业申请记录的大学

```
DELETE FROM College
WHERE NOT EXISTS(
        SELECT * FROM Apply
    WHERE Apply.cName = College.cName
```

```
    AND Apply.major = 'CS'
);
```

实验前：

```
mysql> select * from college;
+-----------------+-------+------------+
| cName           | state | enrollment |
+-----------------+-------+------------+
| Berkeley        | CA    |      36000 |
| Carnegie Mellon | PA    |      11500 |
| Cornell         | NY    |      21000 |
| MIT             | MA    |      10000 |
| Stanford        | CA    |      15000 |
+-----------------+-------+------------+
5 rows in set (0.00 sec)
```

实验后：

```
mysql> delete from College
    -> where not exists(
    -> select * from Apply
    ->     where Apply.cName = College.cName
    ->     and Apply.major = 'CS'
    -> );
Query OK, 0 rows affected (0.00 sec)

mysql> select * from college;
+-----------------+-------+------------+
| cName           | state | enrollment |
+-----------------+-------+------------+
| Berkeley        | CA    |      36000 |
| Carnegie Mellon | PA    |      11500 |
| Cornell         | NY    |      21000 |
| MIT             | MA    |      10000 |
| Stanford        | CA    |      15000 |
+-----------------+-------+------------+
5 rows in set (0.00 sec)
```

(6)更新 Apply 表：录取申请 Carnegie Mellon 大学、GPA 小于 3.6 的学生，录取专业为 economics；

```
UPDATE Apply
SET decision = 'Y', major = 'economics'
WHERE cName = 'Carnegie Mellon'
AND sID IN(SELECT sID
FROM Student WHERE GPA < 3.6
);
```

修改前：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Carnegie Mellon | EE             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Carnegie Mellon | EE             | Y        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | CS             | NULL     |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | CS             | NULL     |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
25 rows in set (0.00 sec)
```

修改后：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Carnegie Mellon | EE             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Carnegie Mellon | EE             | Y        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | economics      | Y        |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | economics      | Y        |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
25 rows in set (0.00 sec)
```

(7) 更新 Apply 表:将申请 EE 专业具有最高 GPA 学生的专业更新为 CSE。

```
SET @maxGPA := (
  SELECT MAX(s.GPA)
  FROM Apply a
  JOIN Student s ON a.sID = s.sID
```

```
  WHERE a.major = 'EE'
);

UPDATE Apply a
JOIN Student s ON a.sID = s.sID
SET a.major = 'CSE'
WHERE a.major = 'EE' AND s.GPA = @maxGPA;
```

修改前：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Carnegie Mellon | EE             | Y        |
| 123 | Cornell         | EE             | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | EE             | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Carnegie Mellon | EE             | Y        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | economics      | Y        |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | economics      | Y        |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
25 rows in set (0.00 sec)
```

修改后：

```
mysql> select * from Apply;
+-----+-----------------+----------------+----------+
| sID | cName           | major          | decision |
+-----+-----------------+----------------+----------+
| 123 | Berkeley        | CS             | Y        |
| 123 | Carnegie Mellon | CSE            | Y        |
| 123 | Cornell         | CSE            | Y        |
| 123 | Stanford        | CS             | Y        |
| 123 | Stanford        | CSE            | N        |
| 234 | Berkeley        | biology        | N        |
| 345 | Carnegie Mellon | EE             | Y        |
| 345 | Cornell         | bioengineering | N        |
| 345 | Cornell         | CS             | Y        |
| 345 | Cornell         | EE             | N        |
| 345 | MIT             | bioengineering | Y        |
| 456 | Carnegie Mellon | CS             | NULL     |
| 543 | MIT             | CS             | N        |
| 567 | Carnegie Mellon | economics      | Y        |
| 654 | Carnegie Mellon | CS             | NULL     |
| 678 | Stanford        | history        | Y        |
| 765 | Cornell         | history        | N        |
| 765 | Cornell         | psychology     | Y        |
| 765 | Stanford        | history        | Y        |
| 789 | Carnegie Mellon | economics      | Y        |
| 876 | MIT             | biology        | Y        |
| 876 | MIT             | marine biology | N        |
| 876 | Stanford        | CS             | N        |
| 987 | Berkeley        | CS             | Y        |
| 987 | Stanford        | CS             | Y        |
+-----+-----------------+----------------+----------+
25 rows in set (0.00 sec)
```

问题记录：

(1)在 3.5 中，报错：未指定表中的列。

发现原因是对于 sID 这个变量，我未指定是哪个表中的。因此相同列名必须指定是哪个表中的列，否则会引起歧义

(2)在 3.13,要求不使用子查询，通过资料查询可以先排序，后使用 limit 语句。

但在实际操作过程中，发现当多个人的 GPA 都为 maxGPA 时，无法无法判断到底有多少人，因此无法正确输出所有 GPA 为 maxGPA 的学生信息。

(3) 在 3.21 中,由于使用 cnt 时未 group by，报错 In aggregated query without GROUP BY

查询资料发现，当聚集函数和非聚集函数出现在一起时，必须要将非聚集函数进行 group by

(4)在 3.24，mysql 进行减法以后，无法得到完全正确的结果，会出现好几位小数

对结果进行四舍五入，取一位小数即可

(5)在 4.7 中,报错 You can't specify target table 'Apply' for update in FROM clause。

发现不能先 select 出同一表中的某些值，再 update 这个表(在同一语句中)。于是首先通过 sql 变量求出申请 EE 专业的学生的最高 GPA，保存在@maxGPA 中，再进行 Apply 表的更新。