1. 数据表

--1.学生表 Student(SId,Sname,Sage,Ssex)

--SId 学生编号,Sname 学生姓名,Sage 出生年月,Ssex 学生性别

--2.课程表 Course(CId,Cname,TId)

--CId 课程编号,Cname 课程名称,TId 教师编号

--3.教师表 Teacher(TId,Tname)

--TId 教师编号,Tname 教师姓名

--4.成绩表 SC(SId,CId,score)

--SId 学生编号,CId 课程编号,score 分数

2. sql 语句

2.1 学生表 SQL:

```
1.  create table Student(SId varchar(10),Sname varchar(10),Sage datetime,Ssex va
    rchar(10));
2.  insert into Student values('01' , '赵雷' , '1990-01-01' , '男');
3.  insert into Student values('02' , '钱电' , '1990-12-21' , '男');
4.  insert into Student values('03' , '孙风' , '1990-12-20' , '男');
5.  insert into Student values('04' , '李云' , '1990-12-06' , '男');
6.  insert into Student values('05' , '周梅' , '1991-12-01' , '女');
7.  insert into Student values('06' , '吴兰' , '1992-01-01' , '女');
8.  insert into Student values('07' , '郑竹' , '1989-01-01' , '女');
9.  insert into Student values('09' , '张三' , '2017-12-20' , '女');
10. insert into Student values('10' , '李四' , '2017-12-25' , '女');
11. insert into Student values('11' , '李四' , '2012-06-06' , '女');
12. insert into Student values('12' , '赵六' , '2013-06-13' , '女');
13. insert into Student values('13' , '孙七' , '2014-06-01' , '女');
```

2.2 科目表 SQL

```
create table Course(CId varchar(10),Cname nvarchar(10),TId varchar(10));
insert into Course values('01' , '语文' , '02');
insert into Course values('02' , '数学' , '01');
insert into Course values('03' , '英语' , '03');
```

2.3 教师表 SQL

```
create table Teacher(TId varchar(10),Tname varchar(10));
insert into Teacher values('01' , '张三');
insert into Teacher values('02' , '李四');
insert into Teacher values('03' , '王五');
```

2.4 成绩表 SQL

```
create table SC(SId varchar(10),CId varchar(10),score decimal(18,1));
insert into SC values('01' , '01' , 80);
insert into SC values('01' , '02' , 90);
insert into SC values('01' , '03' , 99);
insert into SC values('02' , '01' , 70);
insert into SC values('02' , '02' , 60);
insert into SC values('02' , '03' , 80);
insert into SC values('03' , '01' , 80);
insert into SC values('03' , '02' , 80);
insert into SC values('03' , '03' , 80);
insert into SC values('04' , '01' , 50);
insert into SC values('04' , '02' , 30);
insert into SC values('04' , '03' , 20);
insert into SC values('05' , '01' , 76);
insert into SC values('05' , '02' , 87);
insert into SC values('06' , '01' , 31);
insert into SC values('06' , '03' , 34);
insert into SC values('07' , '02' , 89);
insert into SC values('07' , '03' , 98);
```

3. 练习题

1.查询" 01 "课程比" 02 "课程成绩高的学生的信息及课程分数

1.1 查询同时存在" 01 "课程和" 02 "课程的情况

1.2 查询存在" 01 "课程但可能不存在" 02 "课程的情况(不存在时显示为 null)

1.3 查询不存在" 01 "课程但存在" 02 "课程的情况

2.查询平均成绩大于等于 60 分的同学的学生编号和学生姓名和平均成绩

3.查询在 SC 表存在成绩的学生信息

4.查询所有同学的学生编号、学生姓名、选课总数、所有课程的总成绩(没成绩的显示为 null)

4.1 查有成绩的学生信息

5.查询「李」姓老师的数量

6.查询学过「张三」老师授课的同学的信息

7.查询没有学全所有课程的同学的信息

8.查询至少有一门课与学号为" 01 "的同学所学相同的同学的信息

9.查询和" 01 "号的同学学习的课程 完全相同的其他同学的信息

10.查询没学过"张三"老师讲授的任一门课程的学生姓名

11.查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

12.检索" 01 "课程分数小于 60，按分数降序排列的学生信息

13.按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

14.查询各科成绩最高分、最低分和平均分：

15.以如下形式显示：课程 ID，课程 name，最高分，最低分，平均分，及格率，中等率，优良率，优秀率

及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

按各科成绩进行排序，并显示排名， Score 重复时保留名次空缺

15.1 按各科成绩进行排序，并显示排名， Score 重复时合并名次

16.查询学生的总成绩，并进行排名，总分重复时保留名次空缺

16.1 查询学生的总成绩，并进行排名，总分重复时不保留名次空缺

17.统计各科成绩各分数段人数：课程编号，课程名称，[100-85]，[85-70]，[70-60]，[60-0]及所占百分比

18.查询各科成绩前三名的记录

19.查询每门课程被选修的学生数

20.查询出只选修两门课程的学生学号和姓名

21.查询男生、女生人数

22.查询名字中含有「风」字的学生信息

23.查询同名同性学生名单，并统计同名人数

24.查询 1990 年出生的学生名单

25.查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号升序排列

26.查询平均成绩大于等于 85 的所有学生的学号、姓名和平均成绩

27.查询课程名称为「数学」，且分数低于 60 的学生姓名和分数

28.查询所有学生的课程及分数情况（存在学生没成绩，没选课的情况）

29.查询任何一门课程成绩在 70 分以上的姓名、课程名称和分数

30.查询不及格的课程

31.查询课程编号为 01 且课程成绩在 80 分以上的学生的学号和姓名

32.求每门课程的学生人数

33.成绩不重复，查询选修「张三」老师所授课程的学生中，成绩最高的学生信息及其成绩

34.成绩有重复的情况下，查询选修「张三」老师所授课程的学生中，成绩最高的学生信息及其成绩

35.查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩

36.查询每门功成绩最好的前两名

37.统计每门课程的学生选修人数（超过 5 人的课程才统计）。

38.检索至少选修两门课程的学生学号

39.查询选修了全部课程的学生信息

40.查询各学生的年龄，只按年份来算

41.按照出生日期来算，当前月日 < 出生年月的月日则，年龄减一

42.查询本周过生日的学生

43.查询下周过生日的学生

44.查询本月过生日的学生

45.查询下月过生日的学生

4. 答案

1.查询" 01 "课程比" 02 "课程成绩高的学生的信息及课程分数，因为需要全部的学生信息，则需要在 sc 表中得到符合条件的 SId 后与 student 表进行 join，可以 left join 也可以 right join。

```
select * from Student RIGHT JOIN (
    select t1.SId, class1, class2 from
        (select SId, score as class1 from sc where sc.CId = '01')as t1,
        (select SId, score as class2 from sc where sc.CId = '02')as t2
    where t1.SId = t2.SId AND t1.class1 > t2.class2
```

```
)r
on Student.SId = r.SId;
select * from  (
    select t1.SId, class1, class2
    from
        (SELECT SId, score as class1 FROM sc WHERE sc.CId = '01') AS t1,
        (SELECT SId, score as class2 FROM sc WHERE sc.CId = '02') AS t2
    where t1.SId = t2.SId and t1.class1 > t2.class2
) r
LEFT JOIN Student
ON Student.SId = r.SId;
```

1.1 查询同时存在" 01 "课程和" 02 "课程的情况

```
select * from
    (select * from sc where sc.CId = '01') as t1,
    (select * from sc where sc.CId = '02') as t2
where t1.SId = t2.SId;
```

1.2 查询存在" 01 "课程但可能不存在" 02 "课程的情况(不存在时显示为 null)

这一道就是明显需要使用 join 的情况了，02 可能不存在，即为 left join 的右侧或 right join 的左侧即可.

```
select * from
(select * from sc where sc.CId = '01') as t1
left join
(select * from sc where sc.CId = '02') as t2
on t1.SId = t2.SId;
select * from
(select * from sc where sc.CId = '02') as t2
right join
(select * from sc where sc.CId = '01') as t1
on t1.SId = t2.SId;
```

1.3 查询不存在" 01 "课程但存在" 02 "课程的情况

```
select * from sc
```

```
2.  where sc.SId not in (
3.      select SId from sc
4.      where sc.CId = '01'
5.  )
6.  AND sc.CId= '02';
```

查询平均成绩大于等于 60 分的同学的学生编号和学生姓名和平均成绩，这里只用根据学生 ID 把成绩分组，对分组中的 score 求平均值，最后在选取结果中 AVG 大于 60 的即可. 注意，这里必须要给计算得到的 AVG 结果一个 alias.（AS ss）得到学生信息的时候既可以用 join 也可以用一般的联合搜索

```
1.  select student.SId,sname,ss from student,(
2.      select SId, AVG(score) as ss from sc
3.      GROUP BY SId
4.      HAVING AVG(score)> 60
5.      )r
6.  where student.sid = r.sid;
7.  select Student.SId, Student.Sname, r.ss from Student right join(
8.        select SId, AVG(score) AS ss from sc
9.        GROUP BY SId
10.       HAVING AVG(score)> 60
11. )r on Student.SId = r.SId;
12. select s.SId,ss,Sname from(
13. select SId, AVG(score) as ss from sc
14. GROUP BY SId
15. HAVING AVG(score)> 60
16. )r left join
17. (select Student.SId, Student.Sname from
18. Student)s on s.SId = r.SId;
```

查询在 SC 表存在成绩的学生信息

```
1.  select DISTINCT student.*
2.  from student,scwhere student.SId=sc.SId
```

4.查询所有同学的学生编号、学生姓名、选课总数、所有课程的成绩总和联合查询不会显示没选课的学生：

```
1.  select student.sid, student.sname,r.coursenumber,r.scoresumfrom student,
2.  (select sc.sid, sum(sc.score) as scoresum, count(sc.cid) as coursenumber fro
    m sc
3.  group by sc.sid)r
4.  where student.sid = r.sid;
```

如要显示没选课的学生(显示为 NULL)，需要使用 join:

```
select s.sid, s.sname,r.coursenumber,r.scoresumfrom (
    (select student.sid,student.sname
    from student
    )s
    left join
    (select
        sc.sid, sum(sc.score) as scoresum, count(sc.cid) as coursenumber
        from sc
        group by sc.sid
    )r
   on s.sid = r.sid
);
```

4.1 查有成绩的学生信息

这一题涉及到 in 和 exists 的用法，在这种小表中，两种方法的效率都差不多，但是请参考 SQL 查询中 in 和 exists 的区别分析，当表 2 的记录数量非常大的时候，选用 exists 比 in 要高效很多.EXISTS 用于检查子查询是否至少会返回一行数据，该子查询实际上并不返回任何数据，而是返回值 True 或 False.

结论：IN()适合 B 表比 A 表数据小的情况

结论：EXISTS()适合 B 表比 A 表数据大的情况

```
select * from student where exists (select sc.sid from sc where student.sid = sc.sid);
select * from student
where student.sid in (select sc.sid from sc);
```

查询「李」姓老师的数量

```
select count(*)
from teacherwhere tname like '李%';
```

查询学过「张三」老师授课的同学的信息，多表联合查询

```
select student.* from student,teacher,course,sc
where
    student.sid = sc.sid
    and course.cid=sc.cid
    and course.tid = teacher.tid
    and tname = '张三';
```

查询没有学全所有课程的同学的信息，因为有学生什么课都没有选，反向思考，先查询选了所有课的学生，再选择这些人之外的学生.

```
select * from student
where student.sid not in (
  select sc.sid from sc
  group by sc.sid
  having count(sc.cid)= (select count(cid) from course)
);
```

查询至少有一门课与学号为" 01 "的同学所学相同的同学的信息

这个用联合查询也可以，但是逻辑不清楚，我觉得较为清楚的逻辑是这样的：从 sc 表查询 01 同学的所有选课 cid--从 sc 表查询所有同学的 sid 如果其 cid 在前面的结果中--从 student 表查询所有学生信息如果 sid 在前面的结果中

```
select * from student where student.sid in (
    select sc.sid from sc
    where sc.cid in(
        select sc.cid from sc
        where sc.sid = '01'
    )
);
```

10.查询没学过"张三"老师讲授的任一门课程的学生姓名

仍然还是嵌套，三层嵌套， 或者多表联合查询

```
select * from student
    where student.sid not in(
        select sc.sid from sc where sc.cid in(
            select course.cid from course where course.tid in(
                select teacher.tid from teacher where tname = "张三"
            )
        )
    );
select * from student
where student.sid not in(
    select sc.sid from sc,course,teacher
    where
        sc.cid = course.cid
        and course.tid = teacher.tid
        and teacher.tname= "张三"
```

```
16. );
```

11.查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

从 SC 表中选取 score 小于 60 的，并 group by sid，having count 大于 1

```
1.  select student.sid, student.sname, AVG(sc.score) from student,sc
2.  where
3.      student.sid = sc.sid and sc.score<60
4.  group by sc.sid
5.  having count(*)>1;
```

检索" 01 "课程分数小于 60，按分数降序排列的学生信息

双表联合查询，在查询最后可以设置排序方式，语法为 ORDER BY ***** DESC\ASC;

```
1.  select student.*, sc.score from student, sc
2.  where student.sid = sc.sidand sc.score < 60and cid = "01"
3.  ORDER BY sc.score DESC;
```

按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

```
1.  select *  from sc
2.  left join (
3.      select sid,avg(score) as avscore from sc
4.      group by sid
5.      )r
6.  on sc.sid = r.sid
7.  order by avscore desc;
```

查询各科成绩最高分、最低分和平均分：

以如下形式显示：课程 ID，课程 name，最高分，最低分，平均分，及格率，中等率，优良率，优秀率

及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

```
1.  select
2.  sc.CId ,
3.  max(sc.score)as 最高分,
4.  min(sc.score)as 最低分,
5.  AVG(sc.score)as 平均分,
6.  count(*)as 选修人数,
7.  sum(case when sc.score>=60 then 1 else 0 end )/count(*)as 及格率,
```

```
8.  sum(case when sc.score>=70 and sc.score<80 then 1 else 0 end )/count(*)as 中等率,
9.  sum(case when sc.score>=80 and sc.score<90 then 1 else 0 end )/count(*)as 优良率,
10. sum(case when sc.score>=90 then 1 else 0 end )/count(*)as 优秀率
11. from sc
12. GROUP BY sc.CId
13. ORDER BY count(*)DESC, sc.CId ASC
```

按各科成绩进行排序，并显示排名， Score 重复时保留名次空缺

这一道题有点 tricky，可以用变量，但也有更为简单的方法，即自交（左交）

用 sc 中的 score 和自己进行对比，来计算“比当前分数高的分数有几个”。

```
1.  select a.cid, a.sid, a.score, count(b.score)+1 as rankfrom sc as a
2.  left join sc as b
3.  on a.score<b.score and a.cid = b.cid
4.  group by a.cid, a.sid,a.score
5.  order by a.cid, rank ASC;
```

查询学生的总成绩，并进行排名，总分重复时不保留名次空缺

这里主要学习一下使用变量。在 SQL 里面变量用@来标识。

```
1.  set @crank=0;select q.sid, total, @crank := @crank +1 as rank from(
2.  select sc.sid, sum(sc.score) as total from sc
3.  group by sc.sid
4.  order by total desc)q;
```

统计各科成绩各分数段人数：课程编号，课程名称，[100-85]，[85-70]，[70-60]，[60-0]及所占百分比

有时候觉得自己真是死脑筋。group by 以后的查询结果无法使用别名，所以不要想着先单表 group by 计算出结果再从第二张表里添上课程信息，而应该先将两张表 join 在一起得到所有想要的属性再对这张总表进行统计计算。这里就不算百分比了，道理相同。

注意一下，用 case when 返回 1 以后的统计不是用 count 而是 sum

```
1.  select course.cname, course.cid,
2.  sum(case when sc.score<=100 and sc.score>85 then 1 else 0 end) as "[100-85]",
3.  sum(case when sc.score<=85 and sc.score>70 then 1 else 0 end) as "[85-70]",
```

```
    sum(case when sc.score<=70 and sc.score>60 then 1 else 0 end) as "[70-60]",
    sum(case when sc.score<=60 and sc.score>0 then 1 else 0 end) as "[60-0]"
    from sc left join course
    on sc.cid = course.cid
    group by sc.cid;
```

查询各科成绩前三名的记录

大坑比。mysql 不能 group by 了以后取 limit，所以不要想着讨巧了，我快被这一题气死了。思路有两种，第一种比较暴力，计算比自己分数大的记录有几条，如果小于 3 就 select，因为对前三名来说不会有 3 个及以上的分数比自己大了，最后再对所有 select 到的结果按照分数和课程编号排名即可。

```
select * from sc
where (
select count(*) from sc as a
where sc.cid = a.cid and sc.score<a.score
)< 3
order by cid asc, sc.score desc;
```

第二种比较灵巧一些，用自身左交，但是有点难以理解。

先用自己交自己，条件为 a.cid = b.cid and a.score<b.score，其实就是列出同一门课内所有分数比较的情况。

想要查看完整的表可以

```
select * from sc a
left join sc b on a.cid = b.cid and a.score<b.score
order by a.cid,a.score;
```

结果

查看，发现结果是 47 行的一个表，列出了类似 01 号课里“30 分小于 50，也小于 70，也小于 80，也小于 90” “50 分小于 70，小于 80，小于 90”.....

所以理论上，对任何一门课来说，分数最高的那三个记录，在这张大表里，通过 a.sid 和 a.cid 可以联合确定这个同学的这门课的这个分数究竟比多少个其他记录高/低，

如果这个特定的 a.sid 和 a.cid 组合出现在这张表里的次数少于 3 个，那就意味着这个组合

（学号+课号+分数）是这门课里排名前三的。

所以下面这个计算中 having count 部分其实 count()或者任意其他列都可以，这里制定了

一个列只是因为比 count()运行速度上更快。

```
select a.sid,a.cid,a.score from sc a
left join sc b on a.cid = b.cid and a.score<b.score
group by a.cid, a.sid
having count(b.cid)<3
order by a.cid;
```

查询每门课程被选修的学生数

```
select cid, count(sid) from sc
group by cid;
```

查询出只选修两门课程的学生学号和姓名

嵌套查询

```
select student.sid, student.sname from student
where student.sid in
(select sc.sid from sc
group by sc.sid
having count(sc.cid)=2
);
```

联合查询

```
select student.SId,student.Sname
from sc,studentwhere student.SId=sc.SId
GROUP BY sc.SId
HAVING count(*)=2;
```

21.查询男生、女生人数

```
select ssex, count(*) from student
group by ssex;
```

查询名字中含有「风」字的学生信息

```
select *
from student where student.Sname like '%风%'
```

23.查询同名学生名单，并统计同名人数

找到同名的名字并统计个数

```
select sname, count(*) from student
group by sname
having count(*)>1;
```

嵌套查询列出同名的全部学生的信息

```
select * from student
where sname in (
select sname from student
group by sname
having count(*)>1
);
```

24.查询 1990 年出生的学生名单

```
select *
from studentwhere YEAR(student.Sage)=1990;
```

25.查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号升序排列

```
select sc.cid, course.cname, AVG(SC.SCORE) as average from sc, course
where sc.cid = course.cid
group by sc.cid
order by average desc,cid asc;
```

26.查询平均成绩大于等于 85 的所有学生的学号、姓名和平均成绩

having 也可以用来截取结果表，在这里就先得到平均成绩总表，再截取 AVG 大于 85 的即可.

```
select student.sid, student.sname, AVG(sc.score) as aver from student, sc
where student.sid = sc.sid
group by sc.sid
having aver > 85;
```

查询课程名称为「数学」，且分数低于 60 的学生姓名和分数

```
select student.sname, sc.score from student, sc, course
where student.sid = sc.sidand course.cid = sc.cidand course.cname = " 数学
"and sc.score < 60;
```

查询所有学生的课程及分数情况（存在学生没成绩，没选课的情况）

```
select student.sname, cid, score from student
left join sc
on student.sid = sc.sid;
```

查询任何一门课程成绩在 70 分以上的姓名、课程名称和分数

```
select student.sname, course.cname,sc.score from student,course,sc
where sc.score>70and student.sid = sc.sidand sc.cid = course.cid;
```

30.查询存在不及格的课程

可以用 group by 来取唯一，也可以用 distinct

```
select cid from scwhere score< 60
group by cid;
select DISTINCT sc.CIdfrom scwhere sc.score <60;
```

31.查询课程编号为 01 且课程成绩在 80 分及以上的学生的学号和姓名

```
select student.sid,student.sname from student,sc
where cid="01"and score>=80and student.sid = sc.sid;
```

求每门课程的学生人数

```
select sc.CId,count(*) as 学生人数 from sc
GROUP BY sc.CId;
```

成绩不重复，查询选修「张三」老师所授课程的学生中，成绩最高的学生信息及其成绩

用 having max()理论上也是对的，但是下面那种按分数排序然后取 limit 1 的更直观可靠

```
select student.*, sc.score, sc.cid from student, teacher, course,sc
where teacher.tid = course.tidand sc.sid = student.sidand sc.cid = course.ci
dand teacher.tname = "张三"
having max(sc.score);
select student.*, sc.score, sc.cid from student, teacher, course,sc
where teacher.tid = course.tidand sc.sid = student.sidand sc.cid = course.ci
dand teacher.tname = "张三"
order by score desc
limit 1;
```

成绩有重复的情况下，查询选修「张三」老师所授课程的学生中，成绩最高的学生信息及其

成绩

为了验证这一题，先修改原始数据

```
UPDATE sc SET score=90where sid = "07"
and cid ="02";
```

这样张三老师教的 02 号课就有两个学生同时获得 90 的最高分了。

这道题的思路继续上一题，我们已经查询到了符合限定条件的最高分了，这个时候只用比较

这张表，找到全部 score 等于这个最高分的记录就可，看起来有点繁复。

```
select student.*, sc.score, sc.cid from student, teacher, course,sc
where teacher.tid = course.tidand sc.sid = student.sidand sc.cid = course.ci
dand teacher.tname = "张三"and sc.score = (
    select Max(sc.score)
    from sc,student, teacher, course
    where teacher.tid = course.tid
    and sc.sid = student.sid
    and sc.cid = course.cid
    and teacher.tname = "张三"
);
```

查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩

同上，在这里用了 inner join 后会有概念是重复的记录：“01 课与 03 课” = “03 课与 01 课”，所以这里取唯一可以直接用 group by

```
select  a.cid, a.sid,  a.score from sc as a
inner join
sc as b
on a.sid = b.sidand a.cid != b.cidand a.score = b.score
group by cid, sid;
```

36.查询每门功成绩最好的前两名

同上 19 题

```
select a.sid,a.cid,a.score from sc as a
left join sc as b
on a.cid = b.cid and a.score<b.score
group by a.cid, a.sid
having count(b.cid)<2
order by a.cid;
```

37.统计每门课程的学生选修人数（超过 5 人的课程才统计）

```
select sc.cid, count(sid) as cc from sc
group by cid
having cc >5;
```

38.检索至少选修两门课程的学生学号

```
select sid, count(cid) as cc from sc
group by sid
having cc>=2;
```

查询选修了全部课程的学生信息

```
select student.*from sc ,student
```

```
2.  where sc.SId=student.SId
3.  GROUP BY sc.SId
4.  HAVING count(*) = (select DISTINCT count(*) from course )
```

40.查询各学生的年龄，只按年份来算

不想做，一般都用 41 题的方法精确到天

按照出生日期来算，当前月日 < 出生年月的月日则，年龄减一

```
1.  select student.SId as 学生编号,student.Sname  as  学生姓名,
2.  TIMESTAMPDIFF(YEAR,student.Sage,CURDATE()) as 学生年龄 from student
```

42.查询本周过生日的学生

```
1.  select *
2.  from student where WEEKOFYEAR(student.Sage)=WEEKOFYEAR(CURDATE());
```

查询下周过生日的学生

```
1.  select *
2.  from student where WEEKOFYEAR(student.Sage)=WEEKOFYEAR(CURDATE())+1;
```

44.查询本月过生日的学生

```
1.  select *
2.  from student where MONTH(student.Sage)=MONTH(CURDATE());
```

45.查询下月过生日的学生

```
1.  select *
2.  from student where MONTH(student.Sage)=MONTH(CURDATE())+1;
```