```
-- 1.查询课程编号为“01”的课程比“02”的课程成绩高的所有学生的学号（重点）
SELECT DISTINCT s_id FROM score
JOIN (SELECT s_id, s_score AS a_score FROM score WHERE c_id = 01) a USING(s_id)
JOIN (SELECT s_id, s_score AS b_score FROM score WHERE c_id = 02) b USING(s_id)
WHERE a.a_score > b.b_score;
```

```
-- 2、查询平均成绩大于 60 分的学生的学号和平均成绩（重点）
SELECT s_id, avg(s_score) AS avg_score
FROM score
GROUP BY s_id HAVING avg_score>60;
```

```
-- 3、查询所有学生的学号、姓名、选课数、总成绩（不重要）
SELECT s_id, s_name, count(c_id) AS "选课数", sum(case when s_score is null then 0 else s_score end)AS "总成绩"
FROM student
LEFT JOIN score USING (s_id)
GROUP BY s_id,s_name;
```

```
-- 4、查询姓“猴”的老师的个数（不重要）
```

```
SELECT count(t_id)

from teacher

WHERE t_name like "猴%";


-- 5、查询没学过“张三”老师课的学生的学号、姓名（重点）

SELECT s_id, s_name

FROM student

WHERE s_id NOT IN

(

SELECT s_id FROM score

    JOIN course USING(c_id)

    JOIN teacher USING(t_id)

WHERE t_name = "张三");


-- 6、查询学过“张三”老师所教的所有课的同学的学号、姓名（重点）

SELECT s.s_id, s.s_name

FROM student s

JOIN score sc USING(s_id)

WHERE sc.c_id IN

(SELECT c_id

FROM teacher

LEFT JOIN course USING(t_id)
WHERE t_name = "张三");

-- 7、查询学过编号为“01”的课程并且也学过编号为“02”的课程的学生的学号、姓名（重点）
SELECT DISTINCT s_id FROM score
RIGHT JOIN (SELECT s_id FROM score WHERE c_id = 01) a USING(s_id)
RIGHT JOIN (SELECT s_id FROM score WHERE c_id = 02) b USING(s_id);

-- 8、查询课程编号为“02”的总成绩（不重点）
SELECT sum(s_score)
FROM score
WHERE c_id = '02';

-- 9、查询所有课程成绩小于 60 分的学生的学号、姓名
-- 无成绩不算在内的
SELECT
    DISTINCT s_id,
      s_name
FROM student
JOIN score USING(s_id)
WHERE s_score < 60;
```

```
--  无成绩也算低于 60

SELECT

    DISTINCT s_id,

      s_name

FROM student

LEFT JOIN score USING(s_id)

WHERE ifnull(s_score,0) < 60;



-- 10.查询没有学全所有课的学生的学号、姓名(重点)

SELECT

    s_id,

      s_name

FROM student

LEFT JOIN score sc USING(s_id)

GROUP BY s_id, s_name

HAVING count(sc.c_id) < (SELECT count(c_id) FROM course);


-- 11、查询至少有一门课与学号为“01”的学生所学课程相同的学生的

学号和姓名（重点）

SELECT

    DISTINCT s_id,
```

```sql
        s_name
FROM student
JOIN score USING (s_id)
WHERE c_id in
(
SELECT c_id FROM score
WHERE s_id = '01'
)
AND s_id != '01';

-- 12.查询和“01”号同学所学课程完全相同的其他同学的学号(重点)
SELECT s_id, s_name FROM student
JOIN score USING(s_id)
WHERE   s_id != 01
        AND s_id NOT IN(
            SELECT DISTINCT s_id FROM score WHERE c_id NOT IN
            (SELECT c_id FROM score
                WHERE s_id = '01')
                )
GROUP BY s_id
HAVING COUNT(DISTINCT c_id) = (SELECT COUNT(DISTINCT c_id)
FROM score WHERE s_id = '01');
```

-- 15、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（重点）

```
SELECT
    DISTINCT s_id,
     s_name,
     avg(s_score)
FROM student
JOIN score USING(s_id)
WHERE s_score < 60
GROUP BY s_id, s_name
HAVING count(c_id) >= 2;
```

-- 16、检索"01"课程分数小于 60，按分数降序排列的学生信息（和34 题重复，不重点）

```
SELECT *
FROM student
JOIN score USING(s_id)
WHERE c_id = '01' AND s_score < 60
ORDER BY s_score DESC;
```

-- 17、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(重重点)

```
-- 方法一
SELECT
    s_id, a.语文, b.数学, c.英语, d.平均成绩
FROM (SELECT s_id, avg(s_score) "平均成绩" FROM score GROUP BY s_id) d
LEFT JOIN(SELECT s_id, s_score '语文' FROM score WHERE c_id = '01') a USING(s_id)
LEFT JOIN(SELECT s_id, s_score '数学' FROM score WHERE c_id = '02') b USING(s_id)
LEFT JOIN(SELECT s_id, s_score '英语' FROM score WHERE c_id = '03') c USING(s_id)
ORDER BY d.平均成绩  DESC;
-- 方法 2.
SELECT
    s_id,
      max(if(c_id = '01',s_score,NULL)) '语文',
      max(if(c_id = '02',s_score,NULL)) '数学',
      max(if(c_id = '03',s_score,NULL)) '英语',
      avg(s_score)
FROM score
GROUP BY s_id
```

```
ORDER BY AVG(s_score) DESC;


-- 18.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 (超级重点)
SELECT s.c_id,
c.c_name,
  max(s.s_score),
  min(s.s_score),
  avg(s.s_score),
  sum(if(s.s_score>= 60,1,0)) / count(s_id) AS 及格率,
  sum(if(s.s_score BETWEEN 70 and 79,1,0)) / count(s_id) AS 中等,
  sum(if(s.s_score BETWEEN 80 and 89,1,0)) / count(s_id) AS 优良,
  sum(if(s.s_score>= 90,1,0)) / count(s_id) AS 优秀
FROM score as s
JOIN course as c USING(c_id)
GROUP BY c_id;
-- 19、按各科成绩进行排序，并显示排名(重点 row_number)
SELECT
      s_id,
      c_id,
      s_score,
      ROW_NUMBER() OVER (ORDER BY s_score DESC)
FROM score;

-- 20、查询学生的总成绩并进行排名（不重点）
SELECT
    s_id,
      sum(s_score)
FROM score
GROUP BY s_id
ORDER BY sum(s_score) DESC;

-- 21 、查询不同老师所教不同课程平均分从高到低显示(不重点)
SELECT
    t_id,
      t_name,
      avg(s_score)
FROM teacher
JOIN course USING(t_id)
JOIN score USING(c_id)
GROUP BY t_id
```

```
ORDER BY avg(s_score) DESC;

-- 22、查询所有课程的成绩第 2 名到第 3 名的学生信息及该课程成绩（重要）
SELECT *
FROM (SELECT st.s_id, st.s_name, s_birth, st.s_sex, c_id, s_score, row_number()
          over(partition by c_id ORDER BY s_score DESC) m
            FROM Score sc INNER JOIN student st USING(s_id) ) a
WHERE m in (2,3) ;

-- 23、使用分段[100-85],[85-70],[70-60],[<60]来统计各科成绩,
--          分别统计各分数段人数：课程 ID 和课程名称(重点和 18 题类似)
SELECT
   DISTINCT c_id,
    c_name,
    sum(if(s_score>85,1,0)) '85 以上',
    sum(if(s_score BETWEEN 70 AND 84,1,0)) '85 到 70',
    sum(if(s_score BETWEEN 60 AND 69,1,0)) '60-70',
    sum(if(s_score < 60,1,0)) '<60'
FROM score
```

```
JOIN course USING(c_id)
GROUP BY c_id;

-- 24、查询学生平均成绩及其名次（同 19 题，重点）
SELECT
    s_id,
      s_name,
      avg(s_score),
      row_number()over(ORDER BY avg(s_score) DESC)
FROM student
JOIN score USING (s_id)
GROUP BY s_id;

-- 25、查询各科成绩前三名的记录（不考虑成绩并列情况）
SELECT *
FROM (SELECT c_id ,st.s_id ,s_score, st.s_name,
row_number () over( partition by c_id ORDER BY s_score DESC) AS 'ranking'
from    score sc
INNER JOIN student st
ON sc.s_id =st.s_id) a
WHERE ranking <4 ;
```

```
-- 26、查询每门课程被选修的学生数(不重点)
SELECT
    c_id,
      c_name,
      count(c_id)
FROM score
JOIN course USING(c_id)
GROUP BY c_id;
```

```
-- 27、 查询出只有两门课程的全部学生的学号和姓名(不重点)
SELECT
    s_id,
      s_name
FROM student
JOIN score USING (s_id)
GROUP BY s_id
HAVING count(c_id)=2;
```

```
-- 28、查询男生、女生人数(不重点)
SELECT
    s_sex,
    count(s_sex)
FROM student
GROUP BY s_sex;

-- 29  查询名字中含有"风"字的学生信息（不重点）
SELECT
    *
FROM student
WHERE s_name LIKE "%风%";

-- 31、查询 1990 年出生的学生名单（重点 year）
SELECT *
FROM student
WHERE YEAR(s_birth) = '1990';

-- 32、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（不重要）
SELECT
    s_id,
      s_name,
      avg(s_score)
FROM student
```

```
JOIN score USING(s_id)
GROUP BY s_id
HAVING avg(s_score) >= 85;

-- 33、查询每门课程的平均成绩，结果按平均成绩升序排序，平均成绩相同时，按课程号降序排列（不重要）
SELECT
    c_id,
      avg(s_score)
FROM score
GROUP BY c_id
ORDER BY avg(s_score), c_id DESC;

-- 34、查询课程名称为"数学"，且分数低于 60 的学生姓名和分数（不重点）
SELECT
    s_name,
      s_score
FROM student
JOIN score USING(s_id)
JOIN course USING(c_id)
WHERE c_name = '数学'
```

```
AND s_score < 60;

-- 35、查询所有学生的课程及分数情况（重点）
SELECT
    s_id,
     max(if(c_id = '02',s_score,NULL)) '语文',
     max(if(c_id = '01',s_score,NULL)) '语文',
     max(if(c_id = '03',s_score,NULL)) '英语',
     avg(s_score)
FROM score
GROUP BY s_id;

-- 36、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（重点）
SELECT
     s_name,
     c_name,
     s_score
FROM student
JOIN score USING (s_id)
JOIN course USING (c_id)
WHERE s_score > 70;
```

```
-- 37、查询不及格的课程并按课程号从大到小排列(不重点)
SELECT
    s_id,
     s_name,
     c_name,
     s_score
FROM student
JOIN score USING (s_id)
JOIN course USING (c_id)
WHERE s_score < 60
ORDER BY c_id DESC;
```

```
-- 38、查询课程编号为 03 且课程成绩在 80 分以上的学生的学号和姓名（不重要）
SELECT
    s_id,
     s_name
FROM student
JOIN score USING (s_id)
WHERE s_score > 80 AND c_id = '03';
```

```
-- 39、求每门课程的学生人数（不重要）
SELECT
    c_id,
      count(s_id)
FROM score
GROUP BY c_id;

-- 40、查询选修“张三”老师所授课程的学生中成绩最高的学生姓名及其成绩（重要 top）
SELECT
    s_name,
      s_score
FROM student
JOIN score USING (s_id)
JOIN course USING (c_id)
JOIN teacher USING (t_id)
WHERE t_name = '张三'
ORDER BY s_score DESC
LIMIT 1;

-- 41.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  （重点）
```

```
SELECT
    s_id, c_id,s_score
FROM student JOIN score USING(s_id)
LEFT JOIN(SELECT s_id, s_score 'scre_01' FROM score WHERE c_id = '01')
a USING(s_id)
LEFT JOIN(SELECT s_id, s_score 'scre_02' FROM score WHERE c_id = '02')
b USING(s_id)
LEFT JOIN(SELECT s_id, s_score 'scre_03' FROM score WHERE c_id = '03')
c USING(s_id)
WHERE a.scre_01 = b.scre_02 AND b.scre_02 = c.scre_03;
```

```
-- 43、统计每门课程的学生选修人数。
--            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（不重要）
SELECT
    c_id,
      count(s_id)
FROM score
GROUP BY c_id
ORDER BY count(s_id) DESC, c_id;
```

```
-- 44、检索至少选修两门课程的学生学号（不重要）
SELECT
    s_id,
      count(c_id)
FROM score
GROUP BY s_id
HAVING count(s_id) >=2;
```

```
-- 45、 查询选修了全部课程的学生信息（重点划红线地方）
SELECT s.*
FROM student s
JOIN score USING (s_id)
GROUP BY s_id
HAVING count(c_id) = (SELECT count(c_id) FROM course);
```

```
-- 46、查询各学生的年龄
SELECT
    s_name,
    FLOOR(DATEDIFF(curdate(), s_birth)/365) AS age
FROM student;
```

```
-- 47、查询本周过生日的学生
```

```sql
SELECT
    s_name,
    s_birth
FROM student
WHERE    week(concat(YEAR(CURDATE()),"-",DATE_FORMAT(s_birth, '%m-%d')),1) = week(CURDATE(),1);

-- 48、查询下周过生日的学生
SELECT
    s_name,
    s_birth
FROM student
WHERE    week(concat(YEAR(CURDATE()),"-",DATE_FORMAT(s_birth, '%m-%d')),1) = week(CURDATE(),1)+1;

-- 49、查询本月过生日的学生
SELECT *
FROM student
WHERE MONTH(s_birth) = MONTH(now());

-- 50、查询下月过生日的学生
SELECT *
```

```
FROM student
WHERE MONTH(s_birth) = MONTH(now())+1;
```