**eg:Query the information of students.**

select \* from student;

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Q1: Query the information and course scores of students whose grades in the "01" course are higher than those in the "02" course**

**Q2:** **Query the information and course scores of students whose grades in the "01" course are lower than those in the "02" course**

**Q3:** **Query the student number, student name and average score of students whose average score is greater than or equal to 60 points**

-- No3

SELECT student.s\_id, s\_name, AVG(s\_score) AS avg\_score FROM student

JOIN score ON student.s\_id = score.s\_id

GROUP BY student.s\_id

HAVING avg\_score >= 60;

A screenshot of a computer

Description automatically generated

**Q4: Query the student number, name and average score of students whose average score is less than 60 points (including those with and without scores)**

**-- No4**

**SELECT student.s\_id, s\_name, AVG(s\_score) AS avg\_score FROM student**

**LEFT JOIN score ON student.s\_id = score.s\_id**

**GROUP BY student.s\_id**

**HAVING avg\_score <= 60 OR avg\_score IS NULL;**

A screenshot of a computer

Description automatically generated

**Q5: Query the student number, student name, total number of courses selected, and total grades of all courses (including those without selected courses) of all students**

-- No5

SELECT student.s\_id, student.s\_name,

COUNT(score.c\_id) AS classes,

SUM(score.s\_score) as total\_score

FROM student

LEFT JOIN score ON student.s\_id = score.s\_id

LEFT JOIN course ON course.c\_id = score.c\_id

GROUP BY student.s\_id;

A screenshot of a computer

Description automatically generated

**Q6: Query the number of teachers with the surname "Li"**

**-- No6**

**SELECT COUNT(t\_id) FROM teacher**

**WHERE t\_name LIKE 'li%';**

A screenshot of a computer

Description automatically generated

**Q7: Query the information of students who have studied with teacher "zhangsan"**

**-- No7**

**SELECT student.\* FROM student**

**LEFT JOIN score ON student.s\_id = score.s\_id**

**LEFT JOIN course ON course.c\_id = score.c\_id**

**LEFT JOIN teacher ON teacher.t\_id = course.t\_id**

**WHERE t\_name = 'zhangsan';**

A screenshot of a computer

Description automatically generated

**Q8: Query the information of students who have not studied under teacher "zhangsan"**

-- No8

SELECT \* FROM student

LEFT JOIN score ON student.s\_id = score.s\_id

WHERE c\_id != (SELECT c\_id FROM course

JOIN teacher teacher ON teacher.t\_id = course.t\_id

WHERE t\_name = 'zhangsan');

**Q9: Query the information of students who have studied the course numbered "01" and also studied the course numbered "02"**

**-- No9**

**SELECT DISTINCT s\_id, s\_name FROM (**

**SELECT student.\*, c\_id FROM student**

**LEFT JOIN score ON student.s\_id = score.s\_id**

**GROUP BY student.s\_id, c\_id**

**HAVING c\_id = '01' OR c\_id = '02') temp;**

A screenshot of a computer

Description automatically generated

**Q10: Query the information of students who have studied the course numbered "01" but not the course numbered "02"**

**Q11: Query information about students who have not completed all courses**

-- No11

SELECT student.\*, COUNT(student.s\_id) AS classes FROM student

LEFT JOIN score on student.s\_id = score.s\_id

GROUP BY student.s\_id

HAVING classes < 3;

A screenshot of a computer

Description automatically generated

**Q12: Query the information of students who have at least one course that is the same as the student with student number "01"**

**Q13: Query the information of other students who are studying the same course as the student numbered "01"**

**Q14: Query the names of students who have not studied any course taught by "zhangsan"**

**Q15: Query the student ID number, name and average grade of students who failed two or more courses (Even if you did not take the exam, your score will be recorded as zero, so it should be considered a failure.)**

-- No15

SELECT s\_id, s\_name, s\_brith, s\_sex FROM (

SELECT student.\*, COUNT(student.s\_id) AS classes FROM student

LEFT JOIN score ON

student.s\_id = score.s\_id

WHERE (s\_score < 60) OR s\_score IS NULL

GROUP BY student.s\_id) temp

;

A screenshot of a computer

Description automatically generated

**Q16: Query the student information of "01" course whose score is less than 60 and sorted by score in descending order**

**-- No16**

**SELECT \* FROM student**

**LEFT JOIN score ON**

**student.s\_id = score.s\_id**

**WHERE (c\_id = '01' AND s\_score < 60) OR s\_score IS NULL**

**ORDER BY s\_score DESC;**

A screenshot of a computer

Description automatically generated

**Q17: Displays the grades for all students in all courses and the average grade by average grade from highest to lowest (Students who did not take the exam will have a score of 0)**

**Q18: Query the highest score, lowest score and average score of each subject: displayed in the following form: course ID, course name, highest score, lowest score, average score, passing rate, medium rate, good rate, excellent rate – passing is >=60 , medium is: 70-80, good is: 80-90, excellent is: >=90**

**-- No18**

**SELECT course.\*,**

**MAX(s\_score) AS highest\_score,**

**MIN(s\_score) AS lowest\_score,**

**AVG(s\_score) AS average\_score,**

**IF(**

**AVG(s\_score) >= 90, "Excellent",**

**IF(AVG(s\_score) > 80, "Good",**

**IF(AVG(s\_score) > 70, "Medium",**

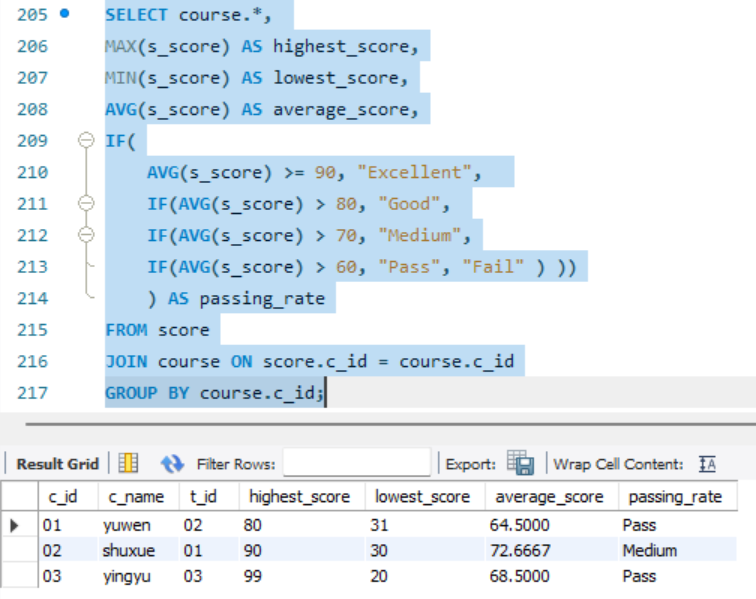
**IF(AVG(s\_score) > 60, "Pass", "Fail" ) ))**

**) AS passing\_rate**

**FROM score**

**JOIN course ON score.c\_id = course.c\_id**

**GROUP BY course.c\_id;**



**Q19: Query the student information, Sort by the scores of each subject and display the ranking**

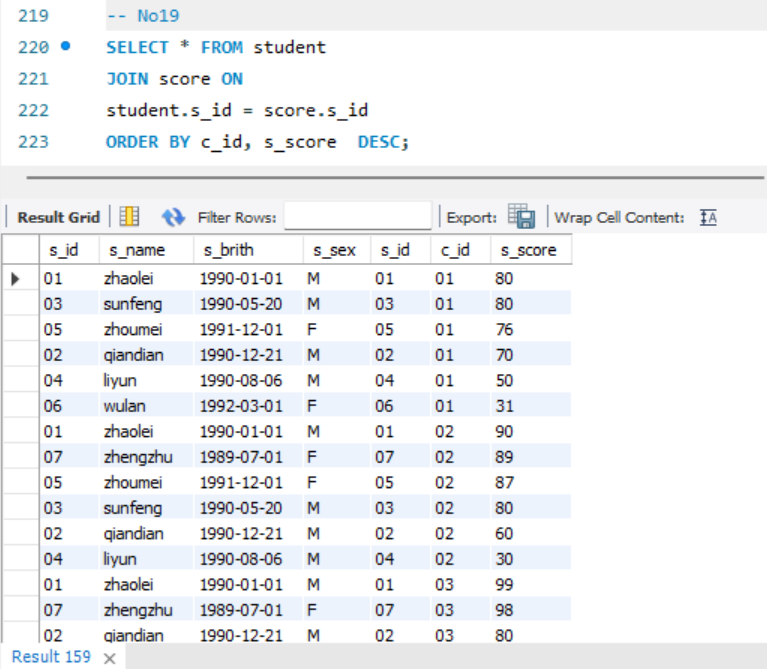
-- No19

SELECT \* FROM student

JOIN score ON

student.s\_id = score.s\_id

ORDER BY c\_id, s\_score DESC;



**Q20: Query students’ total scores and rank them**

-- No20

SELECT student.\*, SUM(s\_score) as total\_score FROM student

LEFT JOIN score ON

student.s\_id = score.s\_id

GROUP BY student.s\_id

ORDER BY total\_score DESC;

A screenshot of a computer

Description automatically generated