**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**

SELECT s.s\_id, s.s\_name, s.s\_brith, s.s\_sex,

c1.c\_name AS course\_01,

c2.c\_name AS course\_02,

sc1.s\_score AS score\_01,

sc2.s\_score AS score\_02

FROM student s

JOIN

score sc1 ON s.s\_id = sc1.s\_id

JOIN

score sc2 ON s.s\_id = sc2.s\_id

JOIN

course c1 ON sc1.c\_id = c1.c\_id

JOIN

course c2 ON sc2.c\_id = c2.c\_id

WHERE

c1.c\_id = '01' AND c2.c\_id = '02' AND sc1.s\_score > sc2.s\_score;

A screenshot of a computer

Description automatically generated

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

SELECT s.s\_id, s.s\_name, s.s\_brith, s.s\_sex,

c1.c\_name AS course\_01,

c2.c\_name AS course\_02,

sc1.s\_score AS score\_01,

sc2.s\_score AS score\_02

FROM student s

JOIN

score sc1 ON s.s\_id = sc1.s\_id

JOIN

score sc2 ON s.s\_id = sc2.s\_id

JOIN

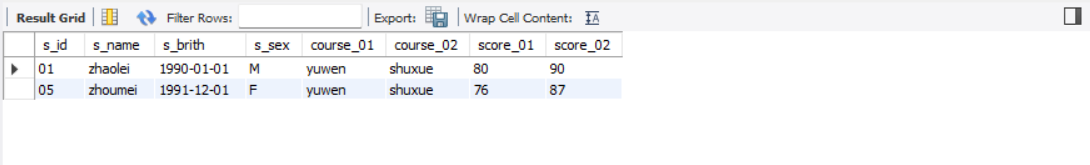
course c1 ON sc1.c\_id = c1.c\_id

JOIN

course c2 ON sc2.c\_id = c2.c\_id

WHERE

c1.c\_id = '01' AND c2.c\_id = '02' AND sc1.s\_score < sc2.s\_score;



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

SELECT s.s\_id, s.s\_name, AVG(sc.s\_score) AS average\_score

FROM student s

JOIN

score sc ON s.s\_id = sc.s\_id

GROUP BY

s.s\_id, s.s\_name

HAVING

AVG(sc.s\_score) >= 60;

A white background with black and white clouds

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)**

SELECT

s.s\_id,

s.s\_name,

IFNULL(AVG(sc.s\_score), 0) AS average\_score

FROM student s

LEFT JOIN

score sc ON s.s\_id = sc.s\_id

GROUP BY

s.s\_id, s.s\_name

HAVING

IFNULL(AVG(sc.s\_score), 0) < 60;

A white background with black and white clouds

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**

SELECT s.s\_id, s.s\_name, COUNT(DISTINCT sc.c\_id) AS total\_courses, SUM(IFNULL(sc.s\_score, 0)) AS total\_score

FROM student s

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

GROUP BY s.s\_id, s.s\_name;

A white background with black dots

Description automatically generated

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

SELECT COUNT(\*) FROM teacher

WHERE t\_name LIKE 'li%'



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

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

FROM student s

WHERE s.s\_id IN (

SELECT DISTINCT sc.s\_id

FROM score sc

JOIN course c ON sc.c\_id = c.c\_id

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

WHERE t.t\_name = 'zhangsan'

);



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

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

FROM student s

WHERE s.s\_id NOT IN (

SELECT DISTINCT sc.s\_id

FROM score sc

JOIN course c ON sc.c\_id = c.c\_id

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

WHERE t.t\_name = 'zhangsan'

);

A white background with black and white text

Description automatically generated with medium confidence

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

SELECT DISTINCT s.\*

FROM

student s

WHERE s.s\_id IN (

SELECT s\_id FROM score WHERE c\_id = '01'

) AND

s.s\_id IN (

SELECT s\_id FROM score WHERE c\_id = '02'

);

A white background with black and white clouds

Description automatically generated

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

SELECT DISTINCT s.\*

FROM student s

WHERE s.s\_id IN (

SELECT s\_id FROM score WHERE c\_id = '01'

) AND

s.s\_id NOT IN (

SELECT s\_id FROM score WHERE c\_id = '02'

);



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

SELECT st.\*

FROM student st

WHERE (SELECT COUNT(DISTINCT c\_id) FROM course) >

(SELECT COUNT(DISTINCT c\_id) FROM score WHERE s\_id = st.s\_id);



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

SELECT DISTINCT s.\*

FROM

student s

JOIN

score sc ON s.s\_id = sc.s\_id

WHERE

sc.c\_id IN (SELECT c\_id FROM score WHERE s\_id = '01') AND s.s\_id NOT LIKE "01";

A white background with black and white clouds

Description automatically generated

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

SELECT DISTINCT s.\*

FROM student s

JOIN

score sc ON s.s\_id = sc.s\_id

WHERE

sc.c\_id IN (SELECT c\_id FROM score WHERE s\_id = '01') AND s.s\_id NOT LIKE '01';

A white background with black and white clouds

Description automatically generated

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

SELECT DISTINCT s.s\_name

FROM student s

WHERE

s.s\_id NOT IN (

SELECT s\_id

FROM score

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

WHERE course.t\_id = (

SELECT t\_id

FROM teacher

WHERE t\_name = 'zhangsan'

)

);

A white background with black dots

Description automatically generated

**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.)**

SELECT s.s\_id, s.s\_name, IFNULL(AVG(sc.s\_score), 0) AS average\_grade

FROM student s

LEFT JOIN

score sc ON s.s\_id = sc.s\_id

GROUP BY

s.s\_id, s.s\_name

HAVING

SUM(IFNULL(sc.s\_score, 0) < 60) >= 2;



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

SELECT s.s\_id, s.s\_name, sc.s\_score

FROM student s

JOIN

score sc ON s.s\_id = sc.s\_id

WHERE

sc.c\_id = '01' AND

sc.s\_score < 60

ORDER BY

sc.s\_score DESC;  
A white background with black and white clouds

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)**

SELECT s.s\_id, s.s\_name, c.c\_id, c.c\_name, IFNULL(sc.s\_score, 0) AS s\_score, IFNULL((SELECT AVG(s\_score) FROM score WHERE s\_id = s.s\_id), 0) AS average\_grade

FROM student s

CROSS JOIN

course c

LEFT JOIN

score sc ON s.s\_id = sc.s\_id AND c.c\_id = sc.c\_id

ORDER BY

average\_grade DESC;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**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**

SELECT c.c\_id, c.c\_name,

MAX(sc.s\_score) AS highest\_score,

MIN(sc.s\_score) AS lowest\_score,

AVG(sc.s\_score) AS average\_score,

AVG(CASE WHEN sc.s\_score >= 60 THEN 1 ELSE 0 END) \* 100 AS passing\_rate,

AVG(CASE WHEN sc.s\_score BETWEEN 70 AND 79.99 THEN 1 ELSE 0 END) \* 100 AS medium\_rate,

AVG(CASE WHEN sc.s\_score BETWEEN 80 AND 89.99 THEN 1 ELSE 0 END) \* 100 AS good\_rate,

AVG(CASE WHEN sc.s\_score >= 90 THEN 1 ELSE 0 END) \* 100 AS excellent\_rate

FROM course c

LEFT JOIN

score sc ON c.c\_id = sc.c\_id

GROUP BY

c.c\_id, c.c\_name;

A screenshot of a computer

Description automatically generated

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

SELECT s.s\_id, s.s\_name, sc.c\_id, sc.s\_score,

(

SELECT COUNT(DISTINCT sc2.s\_id) + 1

FROM score sc2

WHERE sc2.c\_id = sc.c\_id AND sc2.s\_score > sc.s\_score

) AS ranking

FROM student s

JOIN

score sc ON s.s\_id = sc.s\_id

ORDER BY

sc.c\_id, sc.s\_score DESC;

A white background with black and white clouds

Description automatically generated

A white background with black and white clouds

Description automatically generated

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

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

SUM(sc.s\_score) AS total\_score,

RANK() OVER (ORDER BY SUM(sc.s\_score) DESC) AS ranking

FROM student s

LEFT JOIN

score sc ON s.s\_id = sc.s\_id

GROUP BY

s.s\_id, s.s\_name

ORDER BY

total\_score DESC;

A white background with black and white clouds

Description automatically generated with medium confidence