**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**

Answer:

SELECT t.\*, s.s\_score, s.c\_id

FROM student t

JOIN score s ON t.s\_id = s.c\_id

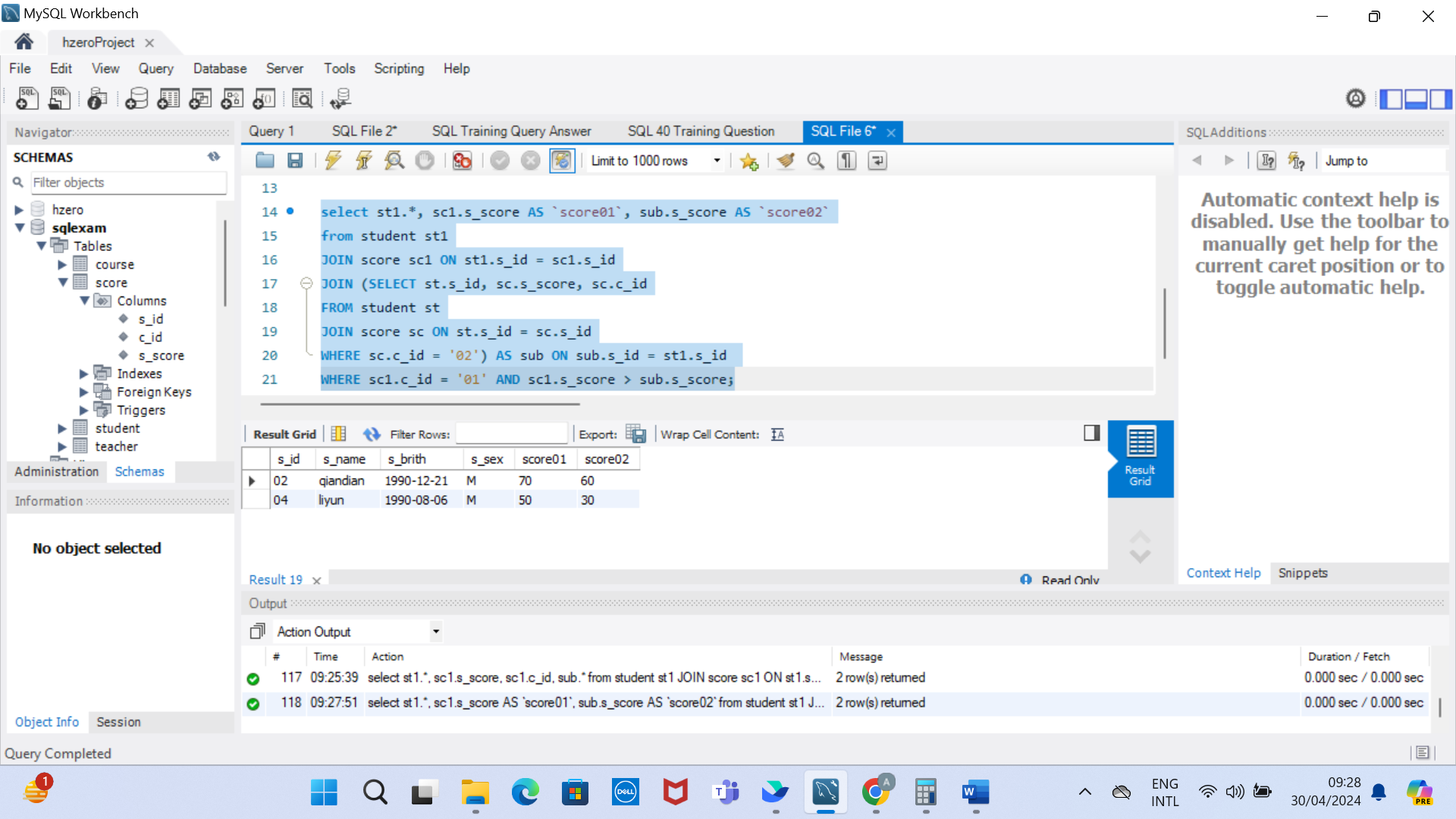
JOIN (SELECT st.\*, sc.s\_score, sc.c\_id

FROM student st

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

WHERE sc.c\_id = '02') AS sub ON sub.s\_id = t.s\_id

WHERE s.c\_id = '01' AND s.s\_score > sub.s\_score;



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

Answer:

select st1.\*, sc1.s\_score AS `score01`, sub.s\_score AS `score02`

from student st1

JOIN score sc1 ON st1.s\_id = sc1.s\_id

JOIN (SELECT st.s\_id, sc.s\_score, sc.c\_id

FROM student st

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

WHERE sc.c\_id = '02') AS sub ON sub.s\_id = st1.s\_id

WHERE sc1.c\_id = '01' AND sc1.s\_score < sub.s\_score;

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT st.s\_id AS `student\_number`, st.s\_name AS `student\_name`, AVG(sc.s\_score) AS `average\_score`

FROM student st

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

GROUP BY `student\_number`, `student\_name`

HAVING `average\_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)**

Answer:

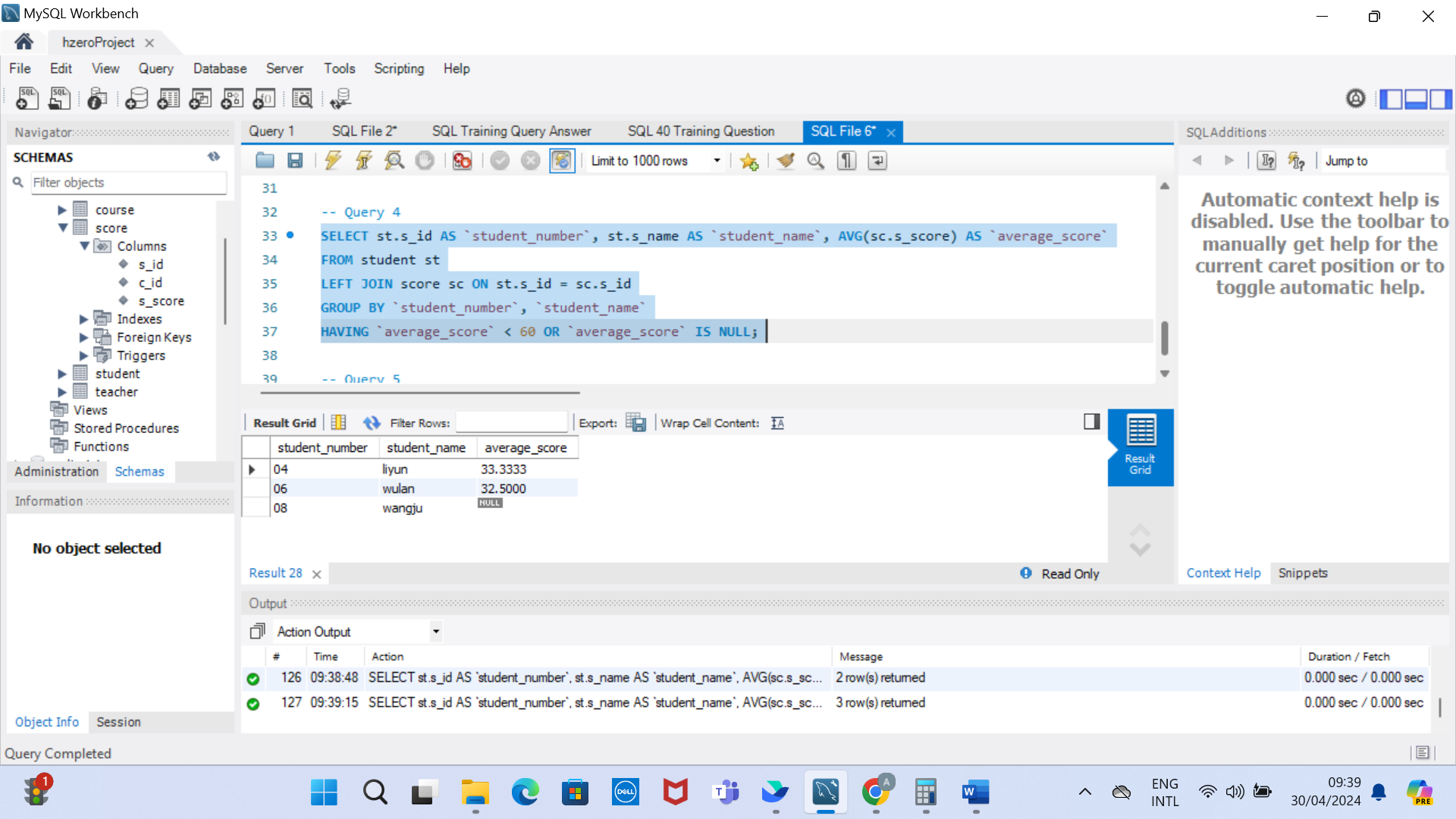
SELECT st.s\_id AS `student\_number`, st.s\_name AS `student\_name`, AVG(sc.s\_score) AS `average\_score`

FROM student st

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

GROUP BY `student\_number`, `student\_name`

HAVING `average\_score` < 60 OR `average\_score` IS NULL;



**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**

Answer:

SELECT st.s\_id AS `student\_number`, st.s\_name AS `student\_name`, COUNT(sc.c\_id) AS `total\_course`,

SUM(sc.s\_score) AS `total\_grades`

FROM student st

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

GROUP BY `student\_number`, `student\_name`;

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT COUNT(t.t\_id) AS `number\_teacher\_with\_li`

FROM teacher t

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

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT st.\*, t.t\_name

FROM student st

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

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

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

WHERE t.t\_name = "zhangsan";

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT st1.\*

FROM student st1

WHERE st1.s\_id NOT IN

(SELECT st.s\_id

FROM student st

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

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

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

WHERE t.t\_name = "zhangsan");

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT st1.\*

FROM student st1

JOIN score sc1 ON st1.s\_id = sc1.s\_id

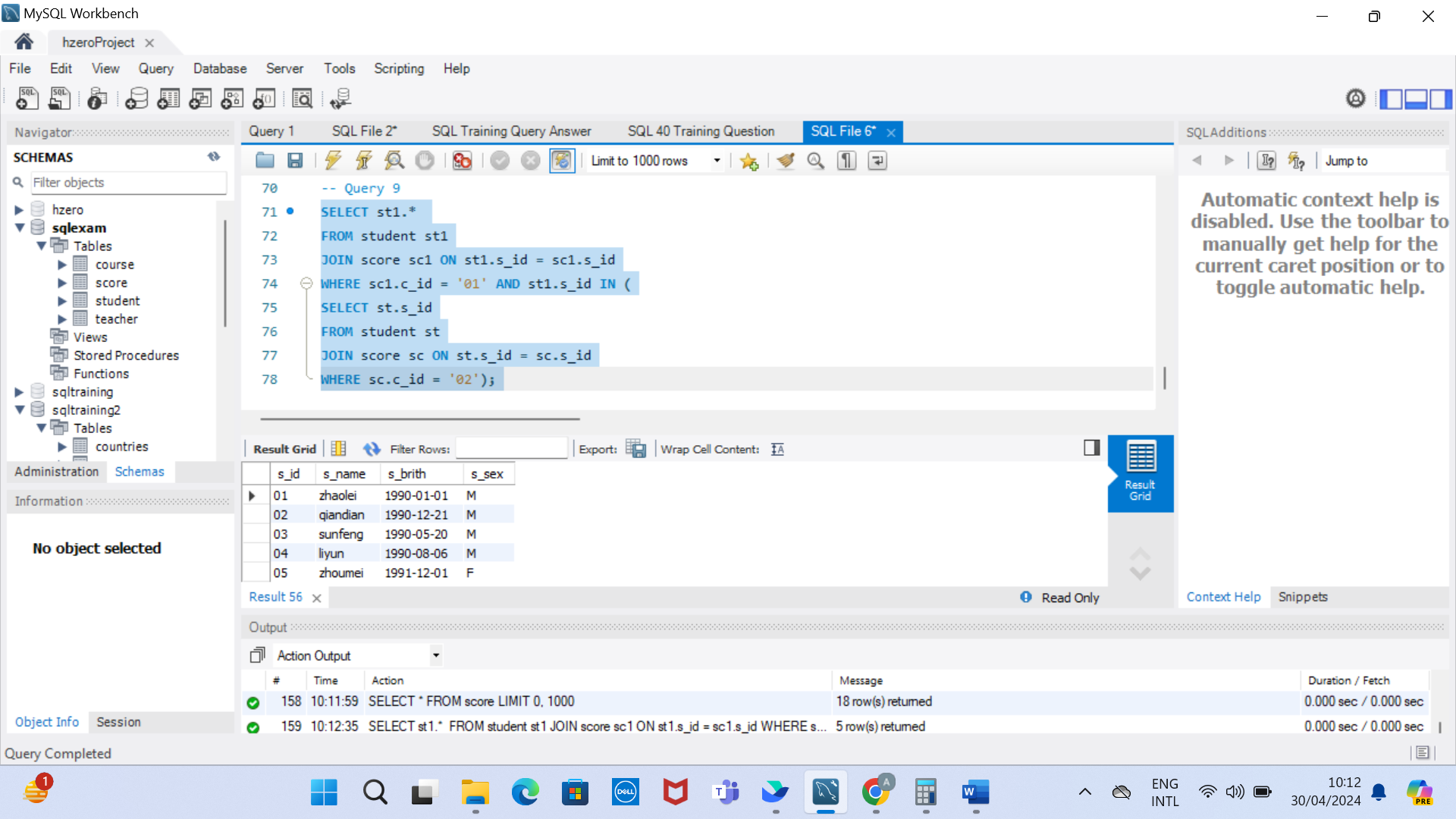
WHERE sc1.c\_id = '01' AND st1.s\_id IN (

SELECT st.s\_id

FROM student st

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

WHERE sc.c\_id = '02');



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

Answer:

SELECT st1.\*

FROM student st1

JOIN score sc1 ON st1.s\_id = sc1.s\_id

WHERE sc1.c\_id = '01' AND st1.s\_id NOT IN (

SELECT st.s\_id

FROM student st

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

WHERE sc.c\_id = '02');

A screenshot of a computer

Description automatically generated

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

Answer:

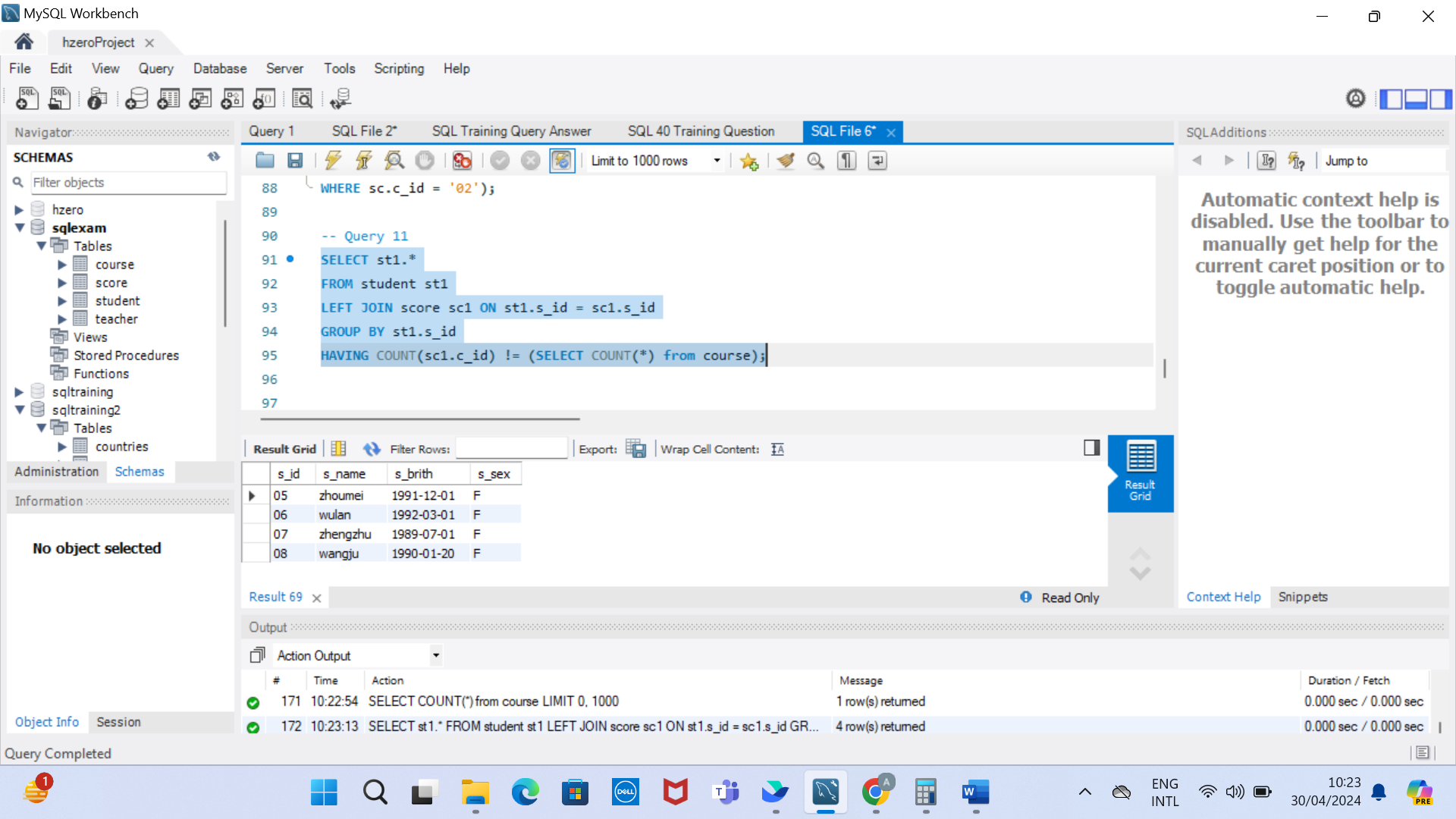
SELECT st1.\*

FROM student st1

LEFT JOIN score sc1 ON st1.s\_id = sc1.s\_id

GROUP BY st1.s\_id

HAVING COUNT(sc1.c\_id) != (SELECT COUNT(\*) from course);



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

Answer:

SELECT DISTINCT st.\*

FROM student st

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

WHERE sc.c\_id IN (

SELECT sc1.c\_id

FROM score sc1

WHERE sc1.s\_id = '01');

A screenshot of a computer

Description automatically generated

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

Answer:

SELECT st.\*

FROM student st

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

WHERE sc.c\_id IN (

SELECT sc1.c\_id

FROM score sc1

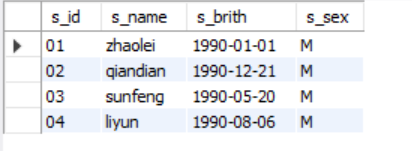
WHERE sc1.s\_id = '01')

GROUP BY st.s\_id

HAVING COUNT(st.s\_id) = (SELECT COUNT(sc1.c\_id)

FROM score sc1

WHERE sc1.s\_id = '01');



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

Answer:

SELECT st1.\*

FROM student st1

WHERE st1.s\_id NOT IN

(SELECT st.s\_id

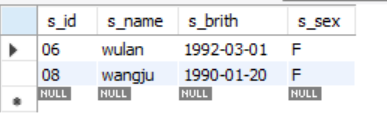
FROM student st

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

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

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

WHERE t.t\_name = "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.)**

Answer:

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

Answer:

SELECT st.\*

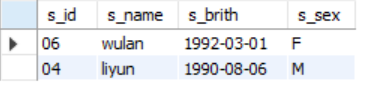
FROM student st

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

WHERE sc.c\_id = '01' AND

sc.s\_score < 60

ORDER BY sc.s\_score;



**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**

Answer:

SELECT cr.c\_id, cr.c\_name, MAX(sc.s\_score) AS `Highest Score`, MIN(sc.s\_score) AS `Lowest Score`,

AVG(sc.s\_score) AS `Average Score`,

(CASE

WHEN AVG(sc.s\_score) >= 60 THEN 'Passing Rate'

WHEN AVG(sc.s\_score) >= 70 AND AVG(sc.s\_score) < 80 THEN 'Medium Rate'

WHEN AVG(sc.s\_score) >= 80 AND AVG(sc.s\_score) < 90 THEN 'Good Rate'

WHEN AVG(sc.s\_score) >= 90 THEN 'Excellent Rate'

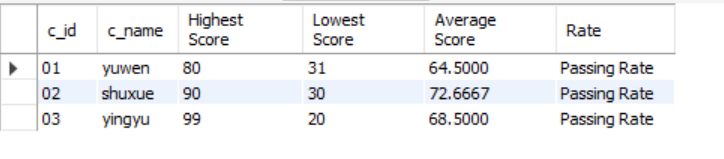
ELSE 'No Rate'

END) AS `Rate`

FROM score sc

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

GROUP BY cr.c\_id, cr.c\_name;



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

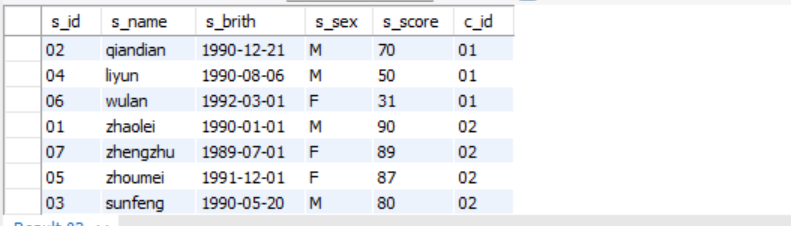
Answer:

SELECT st.\*,sc.s\_score,sc.c\_id

FROM student st

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

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



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

SELECT st.s\_id, SUM(sc.s\_score) AS `Total Score`

FROM student st

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

GROUP BY st.s\_id

ORDER BY `Total Score` DESC

