**INFO20003 Database System Project2  
Juntong Angel Tan student ID: 987979**

**Q1**. Find all the ongoing/unfinished steps. Display the title of these steps and full names of the users who are taking these steps.

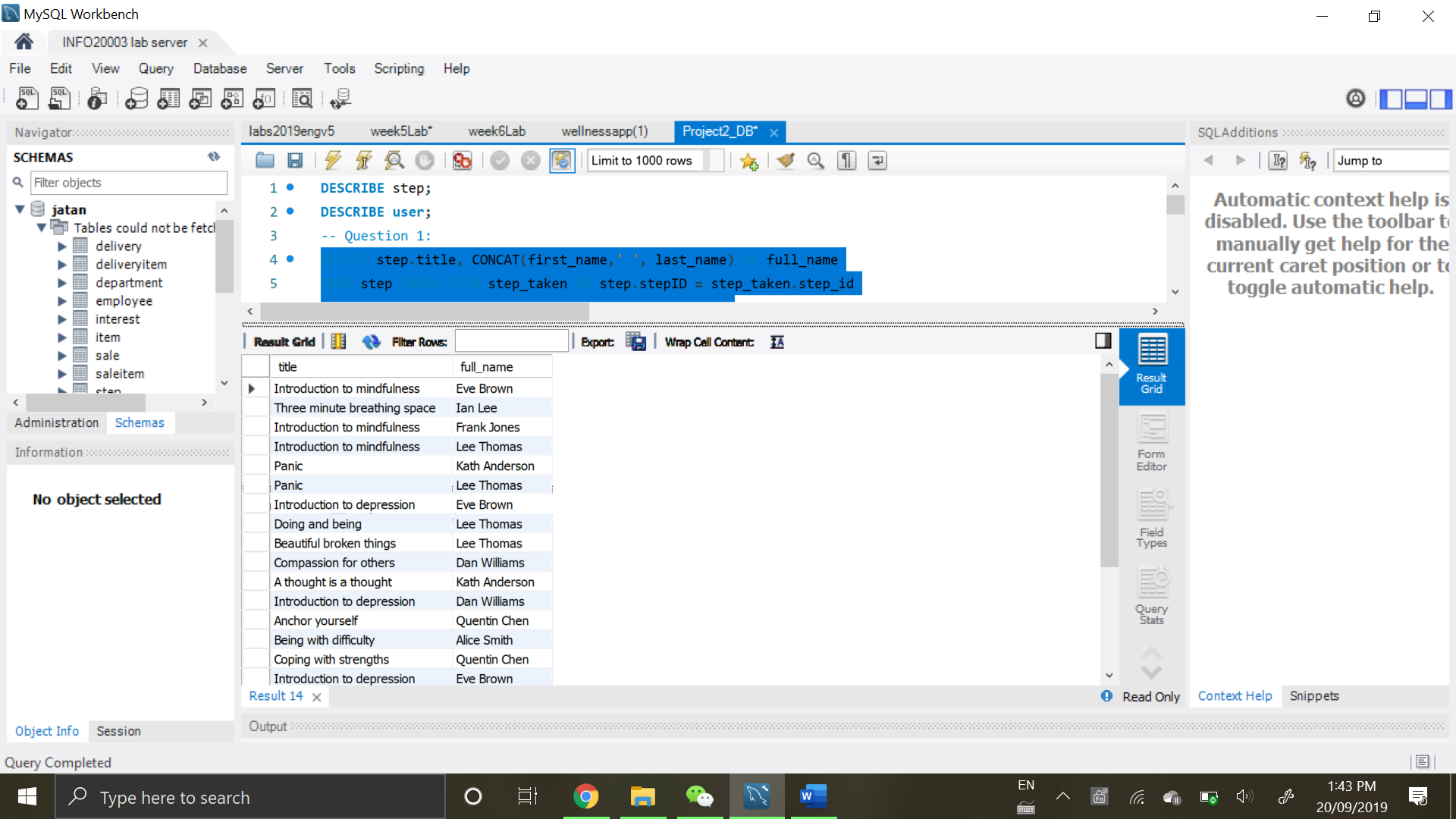
SELECT step.title, CONCAT(user.first\_name, ' ', user.last\_name) AS full\_name

FROM step

INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

INNER JOIN user ON step\_taken.user\_id = user.userID

WHERE step\_taken.when\_finished IS NULL;



24 rows returned

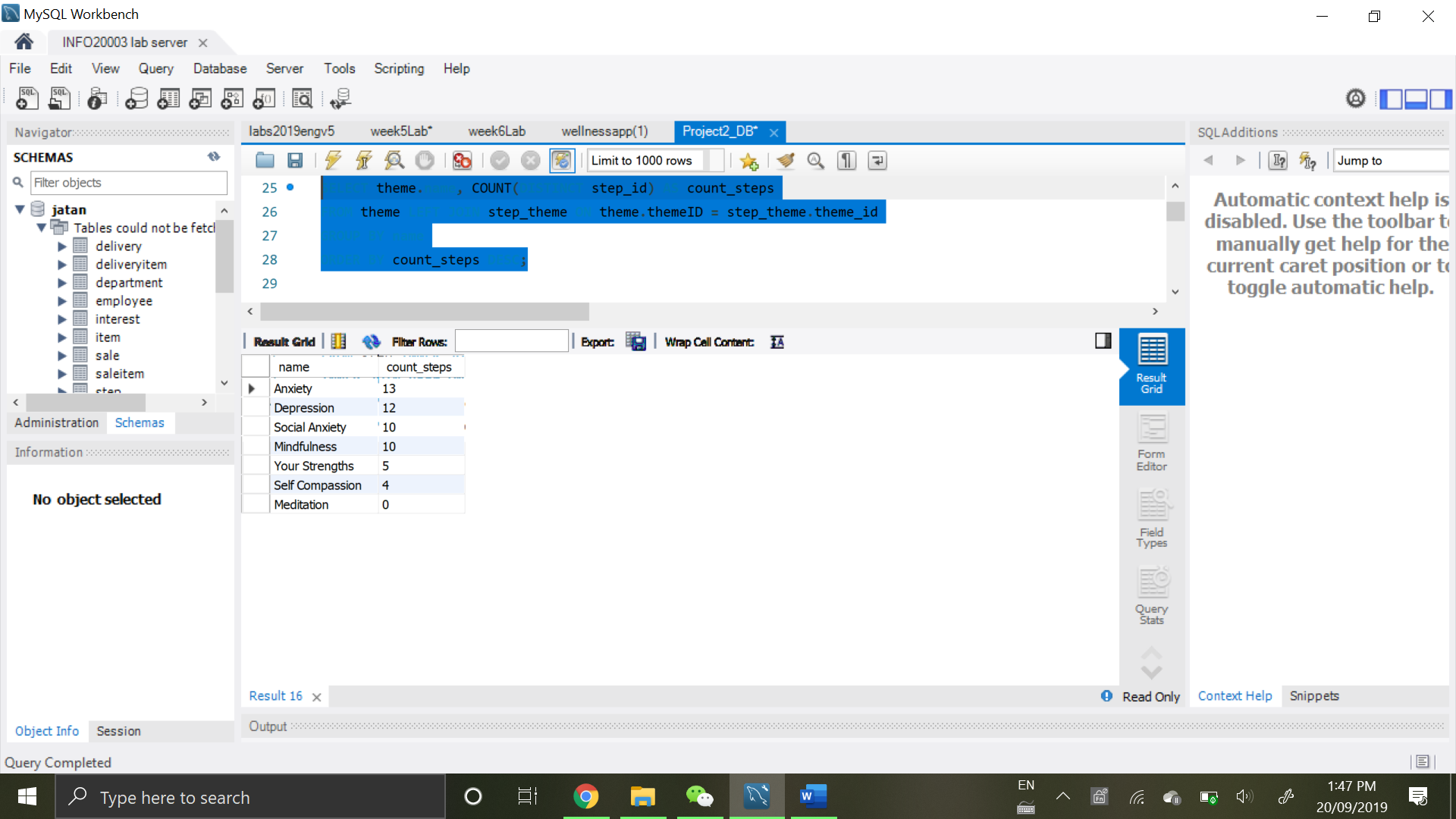
**Q2**. List the themes and the number of the steps associated with these themes. Display the theme name and number of associated steps sorted in descending order.

SELECT theme.name, COUNT(DISTINCT step\_id) AS count\_steps

FROM theme LEFT JOIN step\_theme ON theme.themeID = step\_theme.theme\_id

GROUP BY name

ORDER BY count\_steps DESC;



7 rows returned

**Q3**. Which step is the least popular based on the average rating given by users? Display the title and ID of the step and its average rating(formatted to 2 decimal places). Only include those steps which are rated by at least one user.

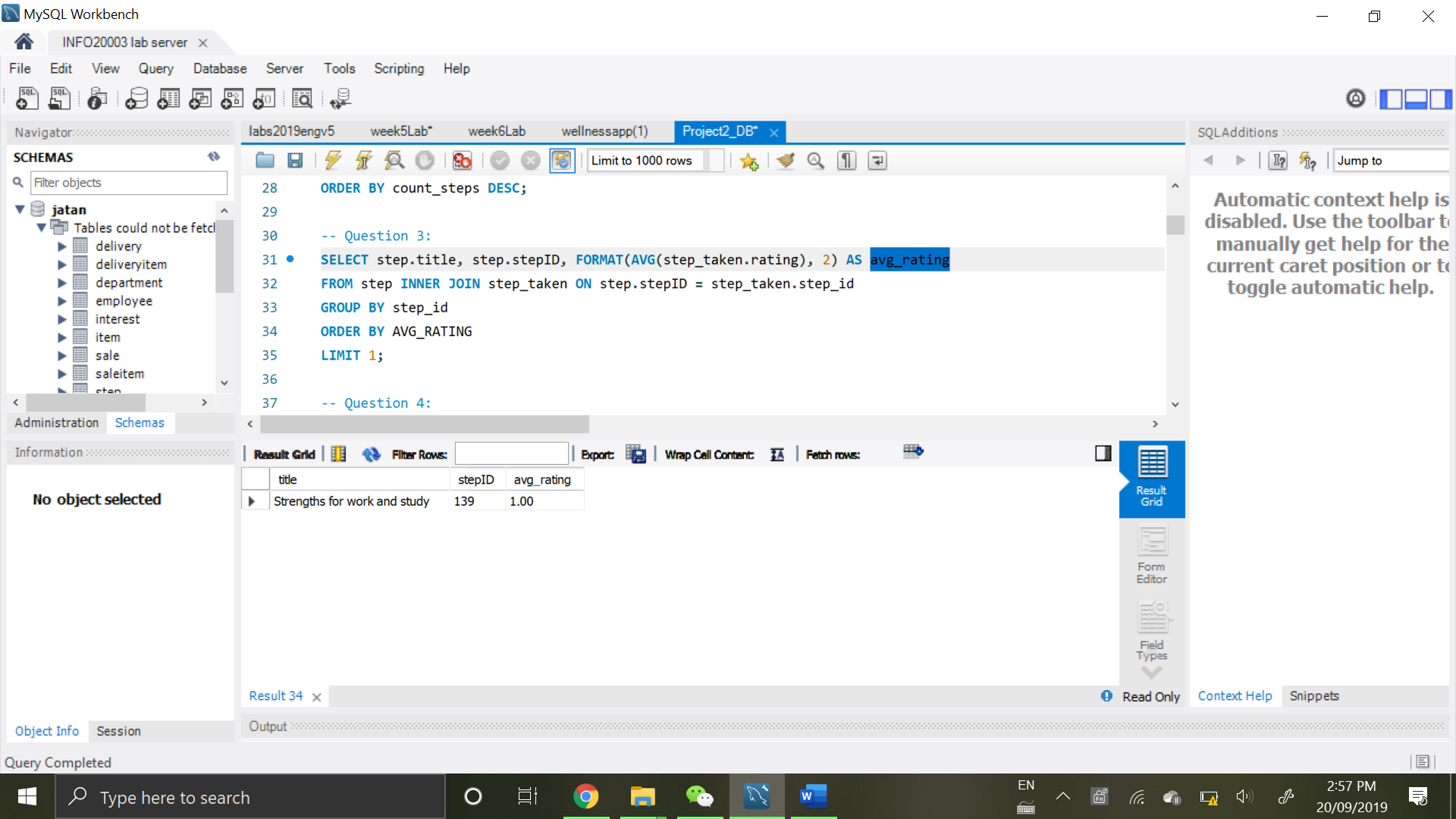
SELECT step.title, step.stepID, FORMAT(AVG(step\_taken.rating), 2) AS avg\_rating

FROM step INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

GROUP BY step\_id

ORDER BY avg\_rating

LIMIT 1;



1 row returned

**Q4**. Find the steps that are taken the greatest number of times. Display the ID, title and count of the times the step has been taken. In case of ties, display all the steps with the same number of times taken.

SELECT step.stepID, step.title, COUNT(\*) AS count\_of\_time

FROM step INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

GROUP BY stepID

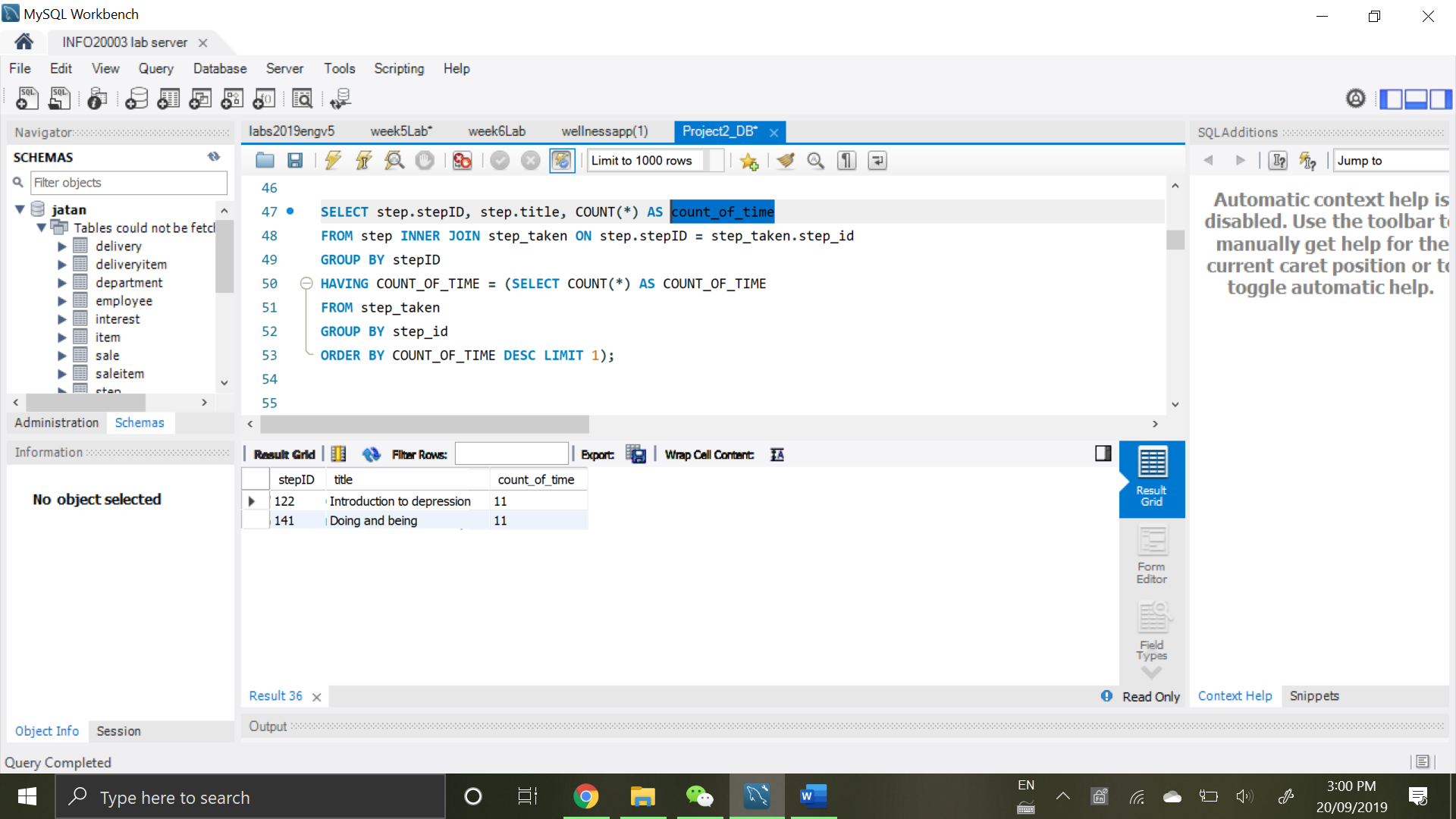
HAVING count\_of\_time = (SELECT COUNT(\*) AS count\_of\_time

FROM step\_taken

GROUP BY step\_id

ORDER BY count\_of\_time DESC

LIMIT 1);



2 rows returned

**Q5**. Who is the most followed user between age of 15 and 18? Display the age(as an integer), first name, and last name of such user along with the number of followers.  
(assumption: “*there will only be one user with the highest number of followers*” from discussion board )

SELECT FORMAT(TIMESTAMPDIFF(YEAR, DOB, CURDATE()), 0) AS age,

user.first\_name, user.last\_name, COUNT(\*) AS num\_followers

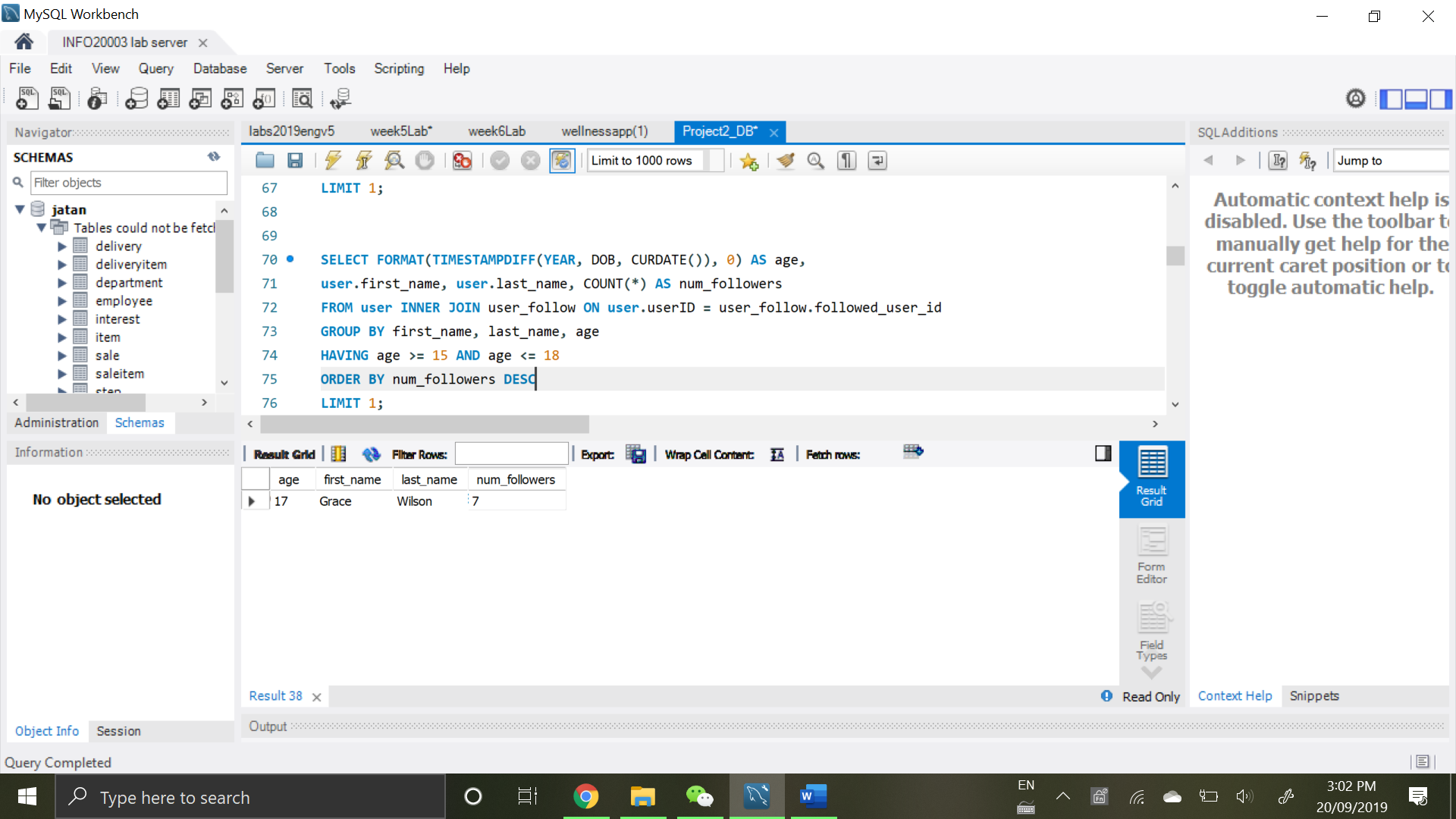
FROM user INNER JOIN user\_follow ON user.userID = user\_follow.followed\_user\_id

GROUP BY first\_name, last\_name, age

HAVING age >= 15 AND age <= 18

ORDER BY num\_followers DESC

LIMIT 1;



1 row returned

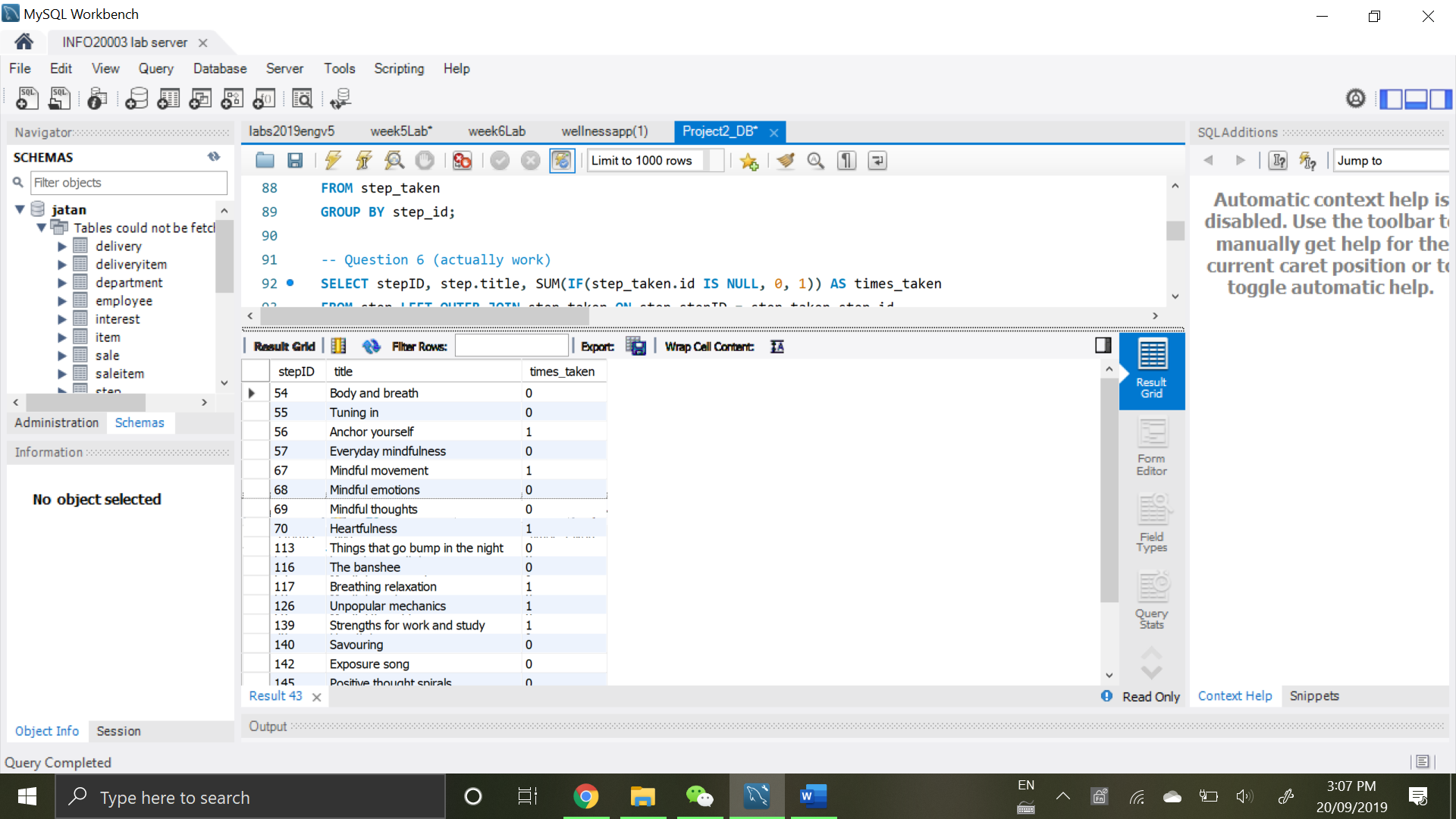
**Q6**. Find all steps that are never taken or are taken exactly once? Display the id and title of these steps along with the indication how many times the step has been taken(0 or 1)

SELECT stepID, step.title, SUM(IF(step\_taken.id IS NULL, 0, 1)) AS times\_taken

FROM step LEFT OUTER JOIN step\_taken ON step.stepID = step\_taken.step\_id

GROUP BY stepID, title

HAVING times\_taken = 0 OR times\_taken = 1;



20 rows returned

**Q7**. Find users who started taking step ‘Doing and being’ after they had started the step ‘Panic’ but have never completed ‘Panic’. Display the user ID, first name and last name.

SELECT DISTINCT user.userID, user.first\_name, user.last\_name

FROM user

INNER JOIN

(SELECT user\_id, when\_started, when\_finished

FROM step INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

WHERE step.title = 'Panic') AS step\_panic

ON user.userID = step\_panic.user\_id

INNER JOIN

(SELECT user\_id, when\_started

FROM step INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

WHERE step.title = 'Doing and being') AS step\_doing

ON step\_panic.user\_id = step\_doing.user\_id

WHERE timestampdiff(SECOND, step\_panic.when\_started, step\_doing.when\_started) > 0

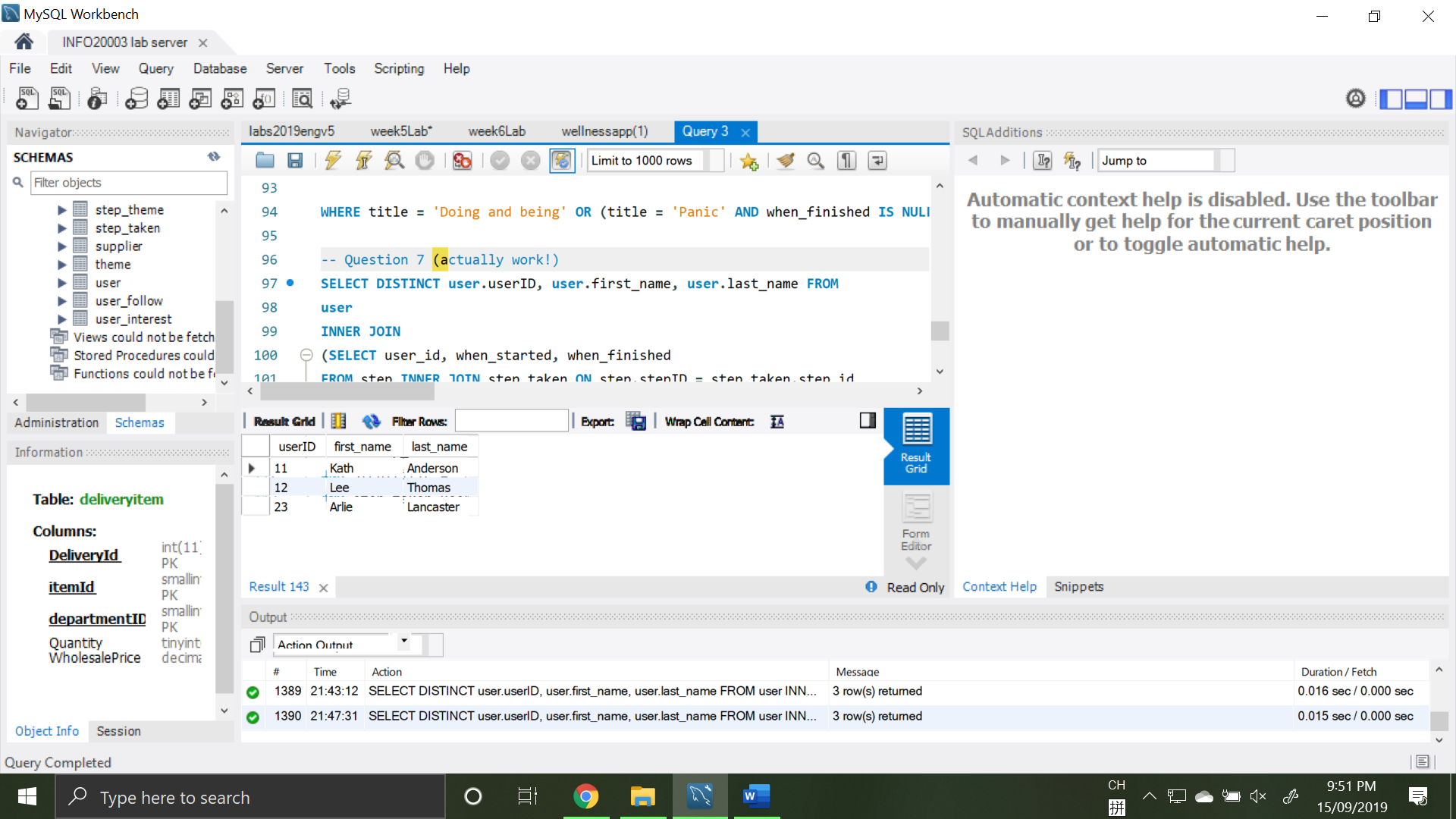
AND NOT EXISTS (SELECT 1 FROM step\_taken

INNER JOIN step ON step\_taken.step\_id = step.stepID

WHERE step\_taken.when\_finished IS NOT NULL

AND step.title = 'Panic'

AND step\_taken.user\_id = user.userID);



3 rows returned

**Q8.** What finished steps were completed both by a user with first name “Alice” and a user with first name “Bob”? Display the ID and title of such steps along with the number of times each user has completed these steps.

SELECT step.stepID, step.title,

SUM(IF(user.first\_name = 'Alice', 1, 0)) AS alice\_counts,

SUM(IF(user.first\_name = 'Bob', 1, 0)) AS bob\_counts

FROM step

INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

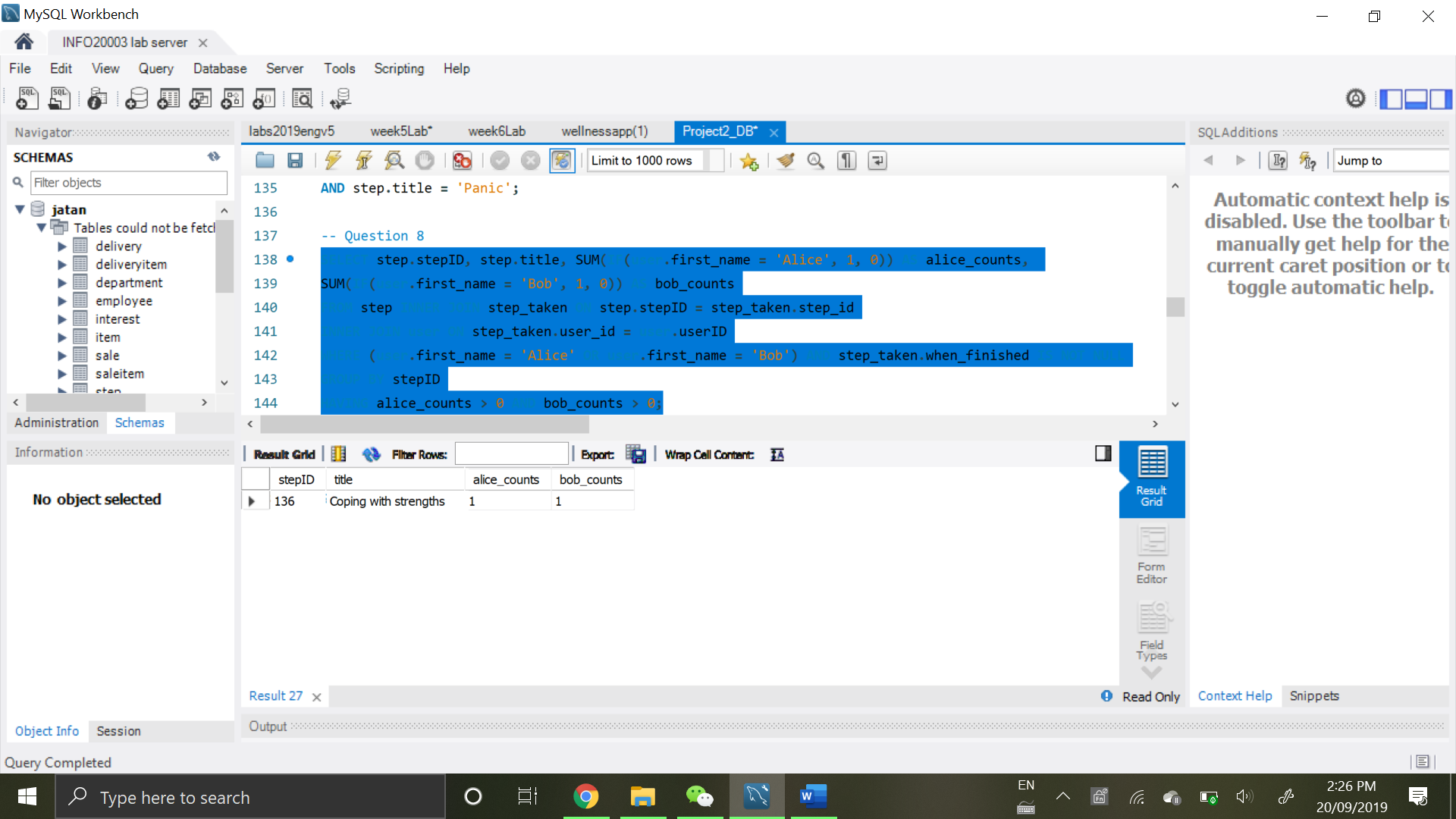
INNER JOIN user ON step\_taken.user\_id = user.userID

WHERE (user.first\_name = 'Alice' OR user.first\_name = 'Bob')

AND step\_taken.when\_finished IS NOT NULL

GROUP BY stepID

HAVING alice\_counts > 0 AND bob\_counts > 0;



1 row returned

**Q9.** Find the top two users with the highest number of interests. For those two users, find out the common steps taken by both of them. Display the titles of the common steps they have taken and the number of times those steps are taken by each user.

SELECT step.title,

SUM(IF(most\_interests.user\_id IS NOT NULL, 1, 0)) AS most\_interests\_count,

SUM(IF(second\_most\_interests.user\_id IS NOT NULL, 1, 0)) AS second\_most\_interests\_count

FROM step

INNER JOIN step\_taken ON step.stepID = step\_taken.step\_id

LEFT OUTER JOIN

(SELECT user\_id

FROM user\_interest

GROUP BY user\_id

ORDER BY COUNT(\*) DESC

LIMIT 1) AS most\_interests

ON step\_taken.user\_id = most\_interests.user\_id

LEFT OUTER JOIN

(SELECT user\_id

FROM user\_interest

GROUP BY user\_id

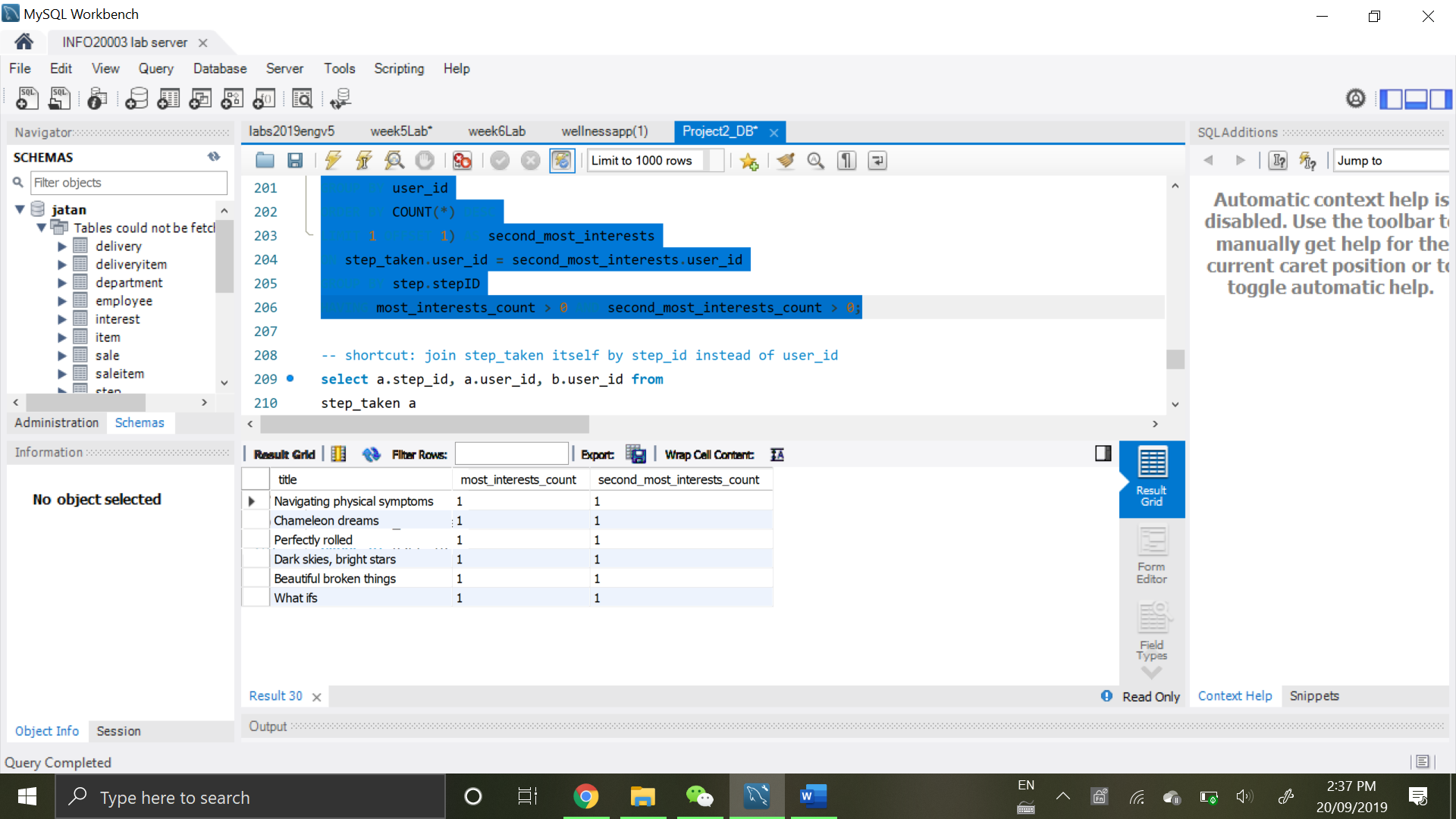
ORDER BY COUNT(\*) DESC

LIMIT 1 OFFSET 1) AS second\_most\_interests

ON step\_taken.user\_id = second\_most\_interests.user\_id

GROUP BY step.stepID

HAVING most\_interests\_count > 0 AND second\_most\_interests\_count > 0;



6 rows returned

**Q10.** For each user taking a step, calculate how many other users have taken the same step. We are only interested in the cases where the step is performed by at least 5 other users. Display the user ID, number of other users (at least 5 other users) who are taking the same step and the title of the taken step.

SELECT DISTINCT step\_taken.user\_id, step.title,

COUNT(DISTINCT step\_taken\_other.user\_id)-1 AS count\_other\_users

FROM step

INNER JOIN step\_taken

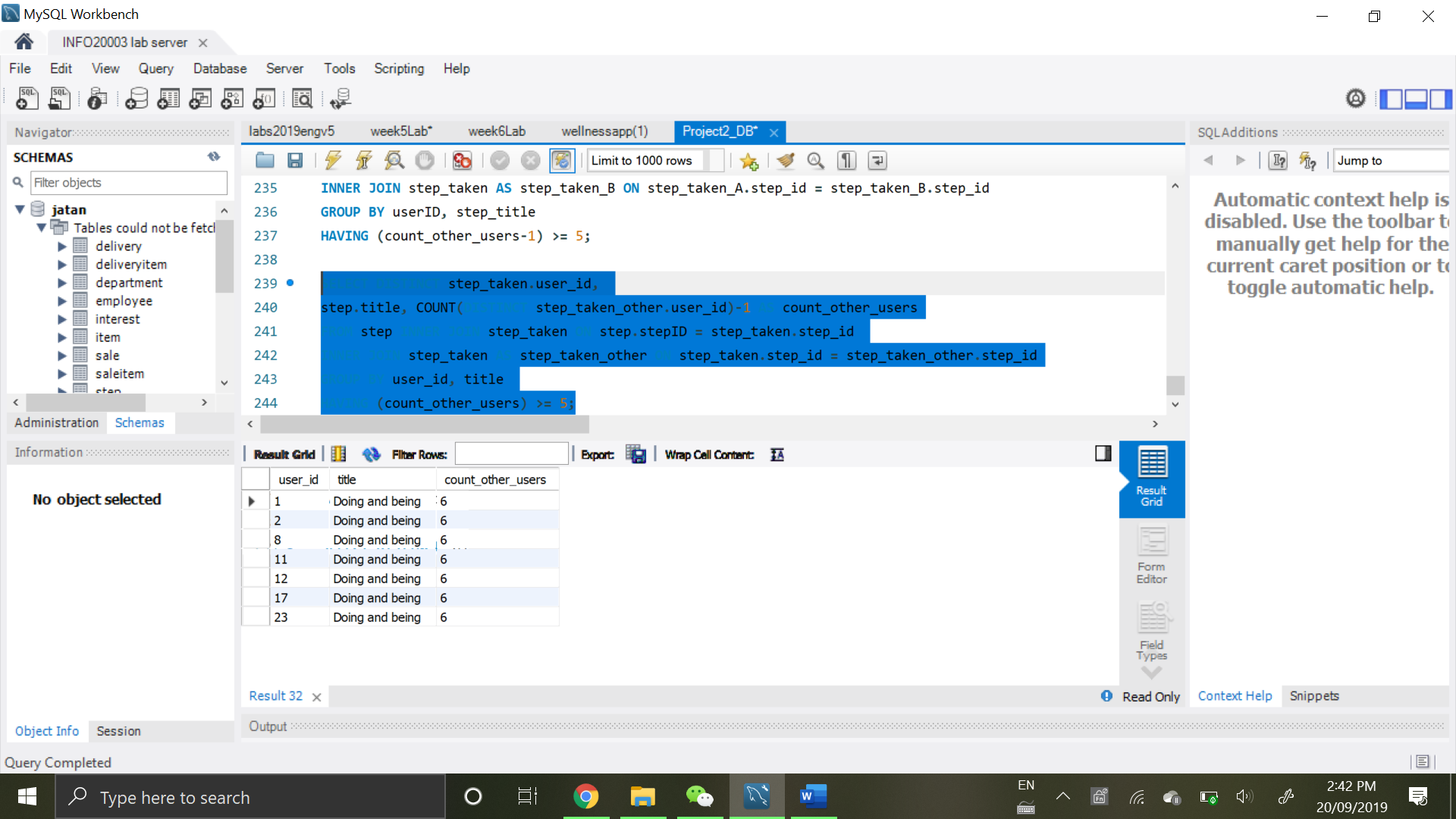
ON step.stepID = step\_taken.step\_id

INNER JOIN step\_taken AS step\_taken\_other

ON step\_taken.step\_id = step\_taken\_other.step\_id

GROUP BY user\_id, title

HAVING (count\_other\_users) >= 5;



7 rows returned