# 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 title AS step_title, CONCAT(first_name,' ', 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;

| step_title | full_name |
|---|---|
| Introduction to mindfulness | Eve Brown |
| Three minute breathing space | Ian Lee |
| Introduction to mindfulness | Frank Jones |
| Introduction to mindfulness | Lee Thomas |
| Panic | Kath Anderson |
| Panic | Lee Thomas |
| Introduction to depression | Eve Brown |
| Doing and being | Lee Thomas |
| Beautiful broken things | Lee Thomas |
| Compassion for others | Dan Williams |

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 name AS theme_name, COUNT(*) AS count_steps

FROM theme INNER JOIN step_theme ON theme.themeID = step_theme.theme_id

GROUP BY theme_name

ORDER BY count_steps DESC;

| theme_name | count_steps |
|---|---|
| Anxiety | 13 |
| Depression | 12 |
| Social Anxiety | 10 |
| Mindfulness | 10 |
| Your Strengths | 5 |
| Self Compassion | 4 |

6 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 title AS step_title, step_id, FORMAT(AVG(rating), 2) AS AVG_RATING

FROM step INNER JOIN step_taken ON step.stepID = step_taken.step_id

INNER JOIN user ON step_taken.user_id = user.userID

GROUP BY step_title, step_id

ORDER BY AVG_RATING

LIMIT 1;

| step_title | step_id | AVG_RATING |
|---|---|---|
| Strengths for work and study | 139 | 1.00 |

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 stepID, title AS step_title, COUNT(*) AS COUNT_OF_TIME

FROM step INNER JOIN step_taken ON step.stepID = step_taken.step_id

INNER JOIN user ON step_taken.user_id = user.userID

GROUP BY stepID, step_title

HAVING COUNT_OF_TIME = (SELECT COUNT(*) AS COUNT_OF_TIME

FROM step INNER JOIN step_taken ON step.stepID = step_taken.step_id

INNER JOIN user ON step_taken.user_id = user.userID

GROUP BY stepID

ORDER BY COUNT_OF_TIME DESC LIMIT 1);

| stepID | step_title | COUNT_OF_TIME |
|--------|-----------|---------------|
| 141 | Doing and being | 11 |
| 122 | Introduction to depression | 11 |

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 TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS age, first_name, last_name, COUNT(*) AS NUM_FOLLOWERS

FROM user INNER JOIN user_follow ON user.userID = user_follow.followed_user_id

WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) >= 15

AND TIMESTAMPDIFF(YEAR, DOB, CURDATE()) <= 18

GROUP BY first_name, last_name, age

ORDER BY NUM_FOLLOWERS DESC

LIMIT 1;

| age | first_name | last_name | NUM_FOLLOWERS |
|-----|------------|-----------|---------------|
| 17  | Grace      | Wilson    | 7             |

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, title AS 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, step_title

HAVING TIMES_TAKEN = 0 OR TIMES_TAKEN = 1;

| stepID | step_title | TIMES_TAKEN |
|--------|------------|-------------|
| 54 | Body and breath | 0 |
| 55 | Tuning in | 0 |
| 56 | Anchor yourself | 1 |
| 57 | Everyday mindfulness | 0 |
| 67 | Mindful movement | 1 |
| 68 | Mindful emotions | 0 |
| 69 | Mindful thoughts | 0 |
| 70 | Heartfulness | 1 |
| 113 | Things that go bump in the night | 0 |
| 116 | The banshee | 0 |

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);

| | userID | first_name | last_name |
|---|---|---|---|
| ▶ | 11 | Kath | Anderson |
| | 12 | Lee | Thomas |
| | 23 | Arlie | Lancaster |

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 stepID, title AS 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;

| | stepID | step_title | alice_counts | bob_counts |
|---|---|---|---|---|
| ▶ | 136 | Coping with strengths | 1 | 1 |

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 stepID, title AS step_title,

SUM(IF(most_interests.userID IS NOT NULL, 1, 0)) AS most_interests_count,

SUM(IF(second_most_interests.userID 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 userID, first_name, last_name

FROM user INNER JOIN user_interest ON user.userID = user_interest.user_id

INNER JOIN interest ON user_interest.interest_id = interest.interestID

GROUP BY userID

ORDER BY COUNT(interestID) DESC

LIMIT 1) AS most_interests

ON step_taken.user_id = most_interests.userID

LEFT OUTER JOIN

(SELECT userID, first_name, last_name

FROM user INNER JOIN user_interest ON user.userID = user_interest.user_id

INNER JOIN interest ON user_interest.interest_id = interest.interestID

GROUP BY userID

ORDER BY COUNT(interestID) DESC

LIMIT 1 OFFSET 1) AS second_most_interests

ON step_taken.user_id = second_most_interests.userID

GROUP BY step.stepID

HAVING most_interests_count > 0 AND second_most_interests_count > 0;

| stepID | step_title | most_interests_count | second_most_interests_count |
|--------|------------|----------------------|------------------------------|
| 127 | Navigating physical symptoms | 1 | 1 |
| 129 | Chameleon dreams | 1 | 1 |
| 130 | Perfectly rolled | 1 | 1 |
| 131 | Dark skies, bright stars | 1 | 1 |
| 133 | Beautiful broken things | 1 | 1 |
| 155 | What ifs | 1 | 1 |

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.

(Assume: now)

SELECT DISTINCT step_taken.user_id AS userID,

title AS step_title, COUNT(DISTINCT step_taken_other.user_id) 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 userID, step_title

HAVING (count_other_users-1) >= 5;

| | userID | step_title | count_other_users |
|---|---|---|---|
| ▶ | 1 | Doing and being | 7 |
| | 2 | Doing and being | 7 |
| | 8 | Doing and being | 7 |
| | 11 | Doing and being | 7 |
| | 12 | Doing and being | 7 |
| | 17 | Doing and being | 7 |
| | 23 | Doing and being | 7 |

7 rows returned