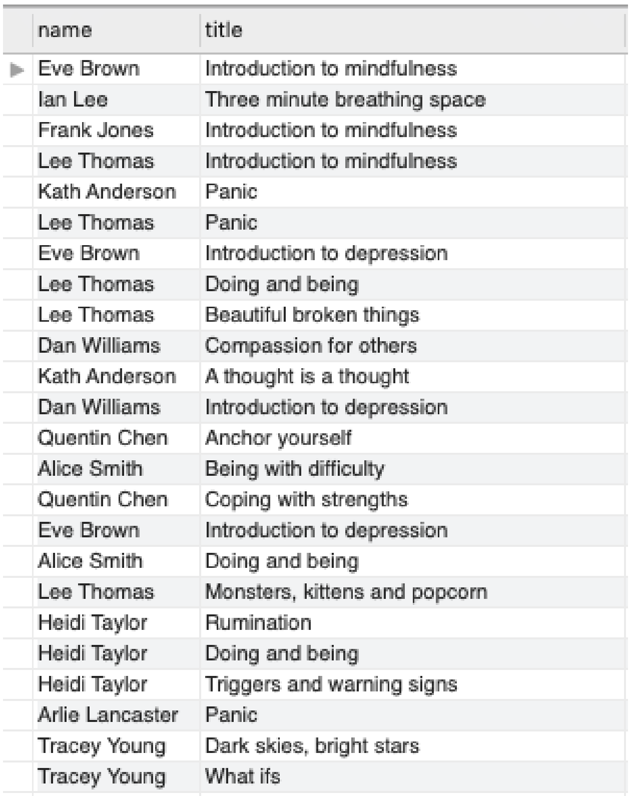
INFO20003 S2 2019 Assignment 2 Solutions

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

(1 mark)

```
SELECT CONCAT(first_name, ' ', last_name) AS name, title
FROM user INNER JOIN step_taken
      INNER JOIN step
ON user.userID = step_taken.user_id
      AND step_taken.step_id=step.stepID
WHERE when_finished IS NULL;
```



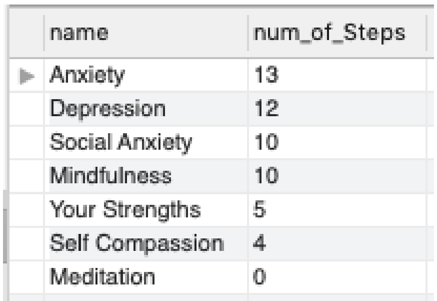
name	title
Eve Brown	Introduction to mindfulness
Ian Lee	Three minute breathing space
Frank Jones	Introduction to mindfulness
Lee Thomas	Introduction to mindfulness
Kath Anderson	Panic
Lee Thomas	Panic
Eve Brown	Introduction to depression
Lee Thomas	Doing and being
Lee Thomas	Beautiful broken things
Dan Williams	Compassion for others
Kath Anderson	A thought is a thought
Dan Williams	Introduction to depression
Quentin Chen	Anchor yourself
Alice Smith	Being with difficulty
Quentin Chen	Coping with strengths
Eve Brown	Introduction to depression
Alice Smith	Doing and being
Lee Thomas	Monsters, kittens and popcorn
Heidi Taylor	Rumination
Heidi Taylor	Doing and being
Heidi Taylor	Triggers and warning signs
Arlie Lancaster	Panic
Tracey Young	Dark skies, bright stars
Tracey Young	What ifs

24 rows returned

2. 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.

(1 mark)

```
SELECT theme.name, COUNT(step_id) AS num_of_Steps
FROM step_theme RIGHT JOIN theme
ON step_theme.theme_id=theme.themeID
GROUP BY theme.name
ORDER BY num_of_Steps DESC;
```



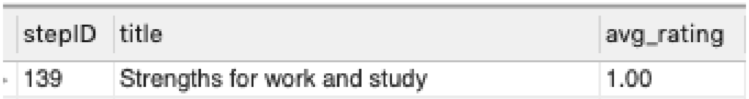
name	num_of_Steps
Anxiety	13
Depression	12
Social Anxiety	10
Mindfulness	10
Your Strengths	5
Self Compassion	4
Meditation	0

7 row returned

3. 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.

(1 mark)

```
SELECT step.stepID, title, CAST(AVG(rating) AS DECIMAL(3,2)) AS avg_rating
FROM step INNER JOIN step_taken
ON step.stepID = step_taken.step_id
GROUP BY step.stepID, title
HAVING avg_rating IS NOT NULL
ORDER BY avg_rating
LIMIT 1;
```



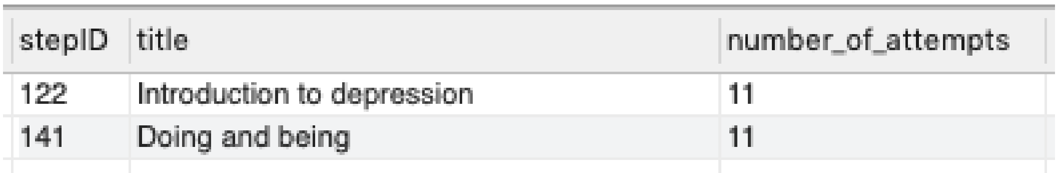
stepID	title	avg_rating
139	Strengths for work and study	1.00

1 row returned

4. 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.

(2 marks)

```
SELECT step.stepID, title, COUNT(*) as number_of_attempts
FROM step INNER JOIN step_taken
ON step.stepID = step_taken.step_id
GROUP BY step.stepID, title
HAVING number_of_attempts =
       (SELECT COUNT(*)
        FROM step_taken
        GROUP BY step_id
        ORDER BY COUNT(*) DESC
        LIMIT 1);
```



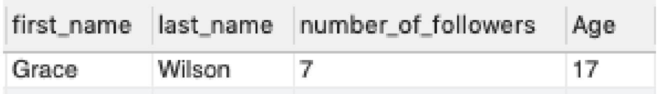
stepID	title	number_of_attempts
122	Introduction to depression	11
141	Doing and being	11

2 rows returned

5. 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.

(2 marks)

```
SELECT first_name, last_name, COUNT(*) AS number_of_followers,
TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS Age
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 user_follow.followed_user_id
ORDER BY COUNT(*) DESC
LIMIT 1;
```



first_name	last_name	number_of_followers	Age
Grace	Wilson	7	17

1 row returned

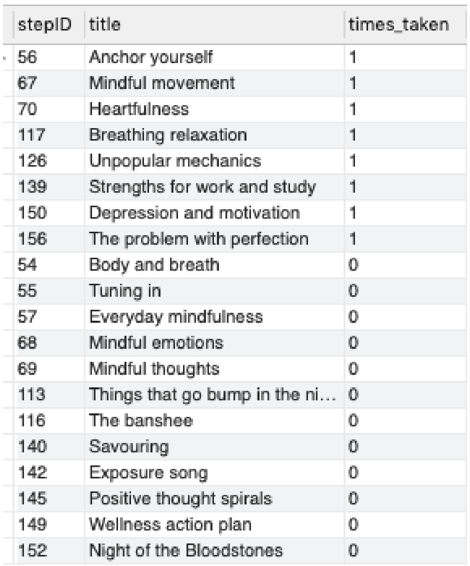
6. Find all steps that are never taken or are taken exactly once? Display the id and title of these steps along with the indication how many times the step has been taken (0 or 1).

(2 marks)

```
SELECT step.stepID, step.title, COUNT(step_taken.step_id) AS times_taken
FROM step LEFT OUTER JOIN step_taken
ON step.stepID = step_taken.step_id
GROUP BY step.stepID
HAVING COUNT(step_taken.step_id) <= 1;
```

Sample Solution 2:

```
SELECT step.stepID, step.title, COUNT(step_taken.step_id) AS times_taken
FROM step INNER JOIN step_taken
ON step.stepID = step_taken.step_id
GROUP BY step.stepID
HAVING COUNT(step_taken.step_id) = 1
UNION
SELECT step.stepID, step.title, 0
FROM step
WHERE stepID NOT IN
      (SELECT step_taken.step_id
       FROM step_taken);
```



stepID	title	times_taken
56	Anchor yourself	1
67	Mindful movement	1
70	Heartfulness	1
117	Breathing relaxation	1
126	Unpopular mechanics	1
139	Strengths for work and study	1
150	Depression and motivation	1
156	The problem with perfection	1
54	Body and breath	0
55	Tuning in	0
57	Everyday mindfulness	0
68	Mindful emotions	0
69	Mindful thoughts	0
113	Things that go bump in the ni…	0
116	The banshee	0
140	Savouring	0
142	Exposure song	0
145	Positive thought spirals	0
149	Wellness action plan	0
152	Night of the Bloodstones	0

20 rows returned

7. 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.

(2 marks)

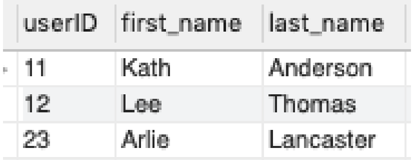
```
SELECT DISTINCT panic_starters.uid, panic_starters.ufn, panic_starters.uln
FROM
(SELECT user.userID AS uid, user.first_name AS ufn, user.last_name AS uln,
step_taken.when_started AS started
FROM user INNER JOIN step_taken
      ON user.userID = step_taken.user_id
      INNER JOIN step
      ON step_taken.step_id = step.stepID
      WHERE step.title =  'Panic'
AND step_taken.when_finished IS NULL) AS panic_starters
INNER JOIN step_taken
ON panic_starters.uid = step_taken.user_id
INNER JOIN step
ON step_taken.step_id = step.stepID
WHERE step.title = 'Doing and being' AND step_taken.when_started >
panic_starters.started;
```

Sample Solution 2:

```
SELECT DISTINCT user.userID, user.first_name, user.last_name
FROM
      (SELECT *
    FROM step INNER JOIN step_taken st
    ON step.stepID = st.step_id) AS st1
INNER JOIN
      (SELECT *
    FROM step INNER JOIN step_taken st
    ON step.stepID = st.step_id) AS st2
ON st1.user_id = st2.user_id
AND st2.title = "Doing and being" AND st1.title = "Panic"
AND st1.when_started < st2.when_started
INNER JOIN user
ON st1.user_id = user.userID
WHERE st1.when_finished IS NULL;
```

Sample Solution 3:

```
SELECT  userID, first_name, last_name
FROM user u INNER JOIN step_taken st1 INNER JOIN step s1
               ON u.userID=st1.user_id AND st1.step_id = s1.stepID
WHERE s1.title = 'Panic' AND st1.when_finished IS NULL AND when_started < ANY (
          SELECT when_started FROM step_taken st2 NATURAL JOIN step s2
          WHERE s2.title = 'Doing and being' AND st1.user_ID = st2.user_ID);
```



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

7 rows returned

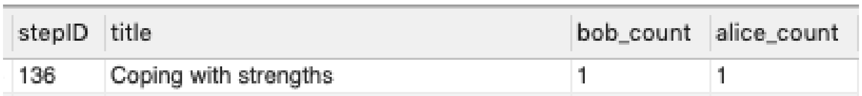
8. What finished steps were completed both by a user with first name "Alice" and a user with first name "Bob"? Display the ID and title of such steps along with the number of times each user has completed these steps.

(3 marks)

```
SELECT step.stepID, step.title, bob_steps.bob_count, alice_steps.alice_count
FROM
      (SELECT step.stepID as sid, COUNT(*) as bob_count
      FROM user
      INNER JOIN step_taken
      ON user.userID = step_taken.user_id
      INNER JOIN step
      ON step_taken.step_id = step.stepID
      WHERE user.first_name = 'Bob' AND step_taken.when_finished is not null
      GROUP BY step.stepID) as bob_steps
INNER JOIN step
ON bob_steps.sid = step.stepID
INNER JOIN
      (SELECT step.stepID as sid, COUNT(*) as alice_count
      FROM user
      INNER JOIN step_taken
      ON user.userID = step_taken.user_id
      INNER JOIN step
      ON step_taken.step_id = step.stepID
      WHERE user.first_name = 'Alice'
AND step_taken.when_finished is not null
      GROUP BY step.stepID) as alice_steps
ON alice_steps.sid = step.stepID;
```

Sample Solution 2:

```
SELECT step_id, title,
      COUNT(CASE WHEN first_name = 'Bob' THEN 1 END) AS Freq_Bob,
      COUNT(CASE WHEN first_name='Alice' THEN 1 END) AS Freq_Alice
FROM user INNER JOIN step_taken
INNER JOIN step
ON userID = user_id AND step_id = stepID
WHERE (first_name = 'Alice' OR first_name = 'Bob') AND
when_finished IS NOT NULL
GROUP BY step_id
HAVING Freq_Bob > 0 AND Freq_Alice > 0;
```



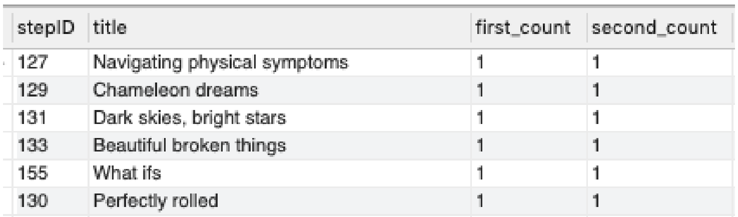
stepID	title	bob_count	alice_count
136	Coping with strengths	1	1

1 row returned

9. 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.

(3 marks)

```
SELECT step.stepID, step.title, first_steps.step_count as first_count,
second_steps.step_count as second_count
FROM
      (SELECT first_user.uid, step_taken.step_id as sid, COUNT(*) as step_count
      FROM (SELECT user_interest.user_id as uid, COUNT(*) as interest_count
             FROM user_interest
             GROUP BY uid
             ORDER BY interest_count DESC
             LIMIT 1) as first_user
      INNER JOIN step_taken
      ON step_taken.user_id = first_user.uid
      GROUP BY step_taken.user_id, step_taken.step_id
      ) as first_steps
INNER JOIN step
ON step.stepID = first_steps.sid
INNER JOIN
      (SELECT second_user.uid, step_taken.step_id as sid, COUNT(*) as step_count
      FROM step_taken
      INNER JOIN
    (SELECT user_interest.user_id as uid, COUNT(*) as interest_count
             FROM user_interest
             GROUP BY uid
             ORDER BY interest_count DESC
      LIMIT 1, 1) as second_user
      ON step_taken.user_id = second_user.uid
      GROUP BY step_taken.user_id, step_taken.step_id
      ) as second_steps
ON step.stepID = second_steps.sid;
```



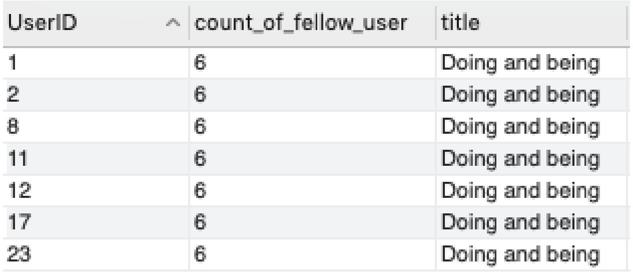
stepID	title	first_count	second_count
127	Navigating physical symptoms	1	1
129	Chameleon dreams	1	1
131	Dark skies, bright stars	1	1
133	Beautiful broken things	1	1
155	What ifs	1	1
130	Perfectly rolled	1	1

6 rows returned

10. 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.

(3 marks)

```
Select t.user_id1 AS UserID, count(t.user_id2) AS count_of_fellow_user, title
FROM
      (SELECT DISTINCT step_taken1.user_id AS user_id1,
                  step_taken2.user_id AS user_id2, step.title
      FROM step_taken AS step_taken1
            INNER JOIN step_taken AS step_taken2
            ON step_taken1.step_id = step_taken2.step_id
            AND step_taken1.user_id <> step_taken2.user_id
            INNER JOIN step ON step_taken1.step_id = step.stepID) AS t
GROUP BY UserID, title
HAVING count_of_fellow_user > 4
ORDER BY count_of_fellow_user DESC;
```



UserID	count_of_fellow_user	title
1	6	Doing and being
2	6	Doing and being
8	6	Doing and being
11	6	Doing and being
12	6	Doing and being
17	6	Doing and being
23	6	Doing and being

7 rows returned