INFO90002 S1 2019 Assignment 2 Example Solutions

1. List the full names (e.g. Alice Smith), as one column, of the users who have not taken any Steps yet.

Solution 1:

```
SELECT CONCAT(first_name, ' ', last_name) AS name
FROM user
WHERE user.id NOT IN (SELECT user_id FROM step_taken);
```

Solution 2:

```
SELECT CONCAT(first_name, ' ', last_name) AS name
FROM user
LEFT JOIN step_taken ON step_taken.user_id = user.id
WHERE step_taken.id IS NULL;
```

Total of 8 rows returned

name
Charlie Nguyen
Grace Wilson
Mallory White
Nick Johnson
Olivia Martin
Robbie Ryan
Sam Thompson
Tracey Young

2. List all of the IDs and titles of Steps that contain the substring ‘mindful’ in their title.

```
SELECT id, title
FROM step
WHERE title LIKE '%mindful%';
```

Total of 5 rows returned

id	title
57	Everyday mindfulness
67	Mindful movement
68	Mindful emotions
69	Mindful thoughts
111	Introduction to mindfulness
NULL	NULL

3. Provide a list of the titles of all Steps *completed* by user with id = 17. Do not show duplicates (list each title only once).

```
SELECT DISTINCT(title)
FROM step_taken INNER JOIN step ON step_taken.step_id = step.id
WHERE step_taken.user_id = 17 AND when_finished IS NOT NULL;
```

Total of 16 rows returned

title
Introduction to depression
A thought is a thought
Beautiful broken things
Chameleon dreams
Coping with strengths
Three minute breathing s...
Introduction to self compa...
Being with difficulty
Doing and being
Enhancing with strengths
Introduction to strengths
Connecting with strengths
What ifs
Triggers and warning signs
Body scan
Negative thought spirals

4. Provide a list of the titles of all Steps that have been taken more than two times along with a count of how many times.

```
SELECT title, COUNT(*)
FROM step INNER JOIN step_taken ON step.id = step_taken.step_id
GROUP BY step_taken.step_id, title HAVING COUNT(*) > 2;
```

Total of 11 rows returned

title	COUNT(*)
Being with difficulty	7
Introduction to mindfulness	4
Introduction to self compa...	4
Introduction to depression	9
Monsters, kittens and pop...	3
Beautiful broken things	3
Three minute breathing s...	8
Coping with strengths	5
Enhancing with strengths	3
Doing and being	10
A thought is a thought	6

5. Which Step(s), listed with columns id, title and the count of times taken, have been taken the greatest number of times?

Solution 1:

```
SELECT step.id, title, COUNT(*)
FROM step INNER JOIN step_taken ON step.id = step_taken.step_id
GROUP BY step.id, title HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM
step_taken GROUP BY step_id ORDER BY COUNT(*));
```

Solution 2:

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

Solution 3:

```
SELECT step_id, title, COUNT(step_id)  AS step_count
FROM (step_taken INNER JOIN step
ON step_taken.step_id=step.id)
GROUP BY step_id
HAVING COUNT(step_id) =
(Select MAX(total)
FROM (SELECT COUNT(step_id) AS Total
FROM step_taken
GROUP BY step_id) AS Results);
```

Total of 1 row returned

id	title	COUNT(*)
141	Doing and being	10

6. List each Step with the title column, along with a count of how many times that Step has been taken and the average rating received by the Step (formatted to 2 decimal places). Order the result by the average rating as a number in descending order.

The completely correct answer for full marks to this will also include Steps with step_taken counts of 0. This is achieved with a query such as the following:

```
SELECT title, COUNT(step_taken.step_id), CAST(AVG(rating) AS DECIMAL(3,2))
AS avg_rating
FROM step LEFT OUTER JOIN step_taken ON step.id = step_taken.step_id
GROUP BY step.id, title ORDER BY avg_rating DESC;
```

The following type of query that does not include Steps with step_taken counts of 0 will still get marks though.

```
SELECT title, COUNT(*), CAST(AVG(rating) AS DECIMAL(3,2)) AS avg_rating
FROM step INNER JOIN step_taken ON step.id = step_taken.step_id
GROUP BY step.id, title ORDER BY avg_rating DESC;
```

Total of 49 rows returned

title	COUNT(step_taken.step_id)	avg_rating
Anchor yourself	1	5.00
Introduction to strengths	2	5.00
Introduction to anxiety	1	5.00
Connecting with strengths	2	5.00
Unpopular mechanics	1	5.00
Mindful movement	1	5.00
Enhancing with strengths	3	4.67
Beautiful broken things	3	4.67
Rumination	2	4.50
Body scan	2	4.50
Compassion for others	2	4.50
Ice breaking and conver...	2	4.50
Negative thought spirals	2	4.50
Fortune telling	2	4.50
Doing and being	10	4.50
Introduction to social an...	2	4.50
Breathing relaxation	1	4.00

7. Provide a list of the titles of all Steps that have been taken by both Alice (id = 1) and Bob (id = 2), along with the combined number of times they have taken the Step.

Solution 1 approach using sub-selects:

```
SELECT title, (SELECT COUNT(*) FROM step_taken WHERE step_taken.step_id =
step.id AND step_taken.user_id IN (1,2)) AS combined_times_done
FROM step
WHERE step.id IN (SELECT step_taken.step_id FROM step_taken WHERE
step_taken.user_id = 1) AND step.id IN (SELECT step_taken.step_id FROM
step_taken WHERE step_taken.user_id = 2);
```

Solution 2 approach using reflexive joins:

```
SELECT title, COUNT(DISTINCT st1.id)
FROM step_taken st1
INNER JOIN step_taken st2 ON (st1.step_id = st2.step_id AND
((st1.user_id=1 AND st2.user_id=2) OR (st2.user_id=1 AND st1.user_id=2)))
INNER JOIN step ON st1.step_id = step.id
GROUP BY title;
```

Total of 3 rows returned

title	combined_times_done
Being with difficulty	2
Coping with strengths	2
Doing and being	3

8. List users older than or equal to 21 years of age, along with a count of how many other users they are following and a count of how many other users are following them. List the user's id, first name, last name, age, following count and followed count, and order the results by first name ascending, then last name ascending.

Solution 1:

```
SELECT id, first_name, last_name, TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS
Age, (SELECT COUNT(*) FROM user_follow WHERE following_user_id = user.id)
AS following_count, (SELECT COUNT(*) FROM user_follow WHERE
followed_user_id = user.id) AS followed_count
FROM user
WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) >= 21
ORDER BY first_name ASC, last_name ASC;
```

Solution 2:

```
SELECT user.id, user.first_name, user.last_name, TIMESTAMPDIFF(YEAR, DOB,
CURDATE()) AS Age, count(uf1.id), count(uf2.id)
FROM user
LEFT JOIN user_follow uf1 ON user.id = uf1.following_user_id
LEFT JOIN user_follow uf2 ON user.id = uf2.followed_user_id
WHERE TIMESTAMPDIFF(YEAR, DOB, CURDATE()) >= 21
GROUP BY user.id, user.first_name, user.last_name, user.DOB
ORDER BY first_name ASC, last_name ASC;
```

Total of 10 rows returned

id	first_name	last_name	Age	following_count	followed_count
5	Eve	Brown	23	3	0
6	Frank	Jones	21	0	1
8	Heidi	Taylor	22	0	2
9	Ian	Lee	21	0	0
10	Judy	Tran	23	0	1
11	Kath	Anderson	22	1	1
12	Lee	Thomas	21	0	1
16	Pat	Wang	24	1	0
17	Quentin	Chen	23	0	1
18	Robbie	Ryan	21	1	1

9. For each (user, theme) pair such that user has taken some steps under the theme, provide a count of how many times a user has taken a step that is categorised under the theme. The output should consist of user ID, user first name, user last name, theme name and the count of steps taken.

```
SELECT user.id, first_name, last_name, theme.name, COUNT(*)
FROM user
INNER JOIN step_taken ON user.id = step_taken.user_id
INNER JOIN step ON step.id = step_taken.step_id
INNER JOIN step_theme on step.id = step_theme.step_id
INNER JOIN theme on step_theme.theme_id = theme.id
GROUP BY user.id, theme.id;
```

Total of 42 rows returned

id	first_name	last_name	name	COUNT(*)
1	Alice	Smith	Social Anxiety	1
1	Alice	Smith	Your Strengths	1
1	Alice	Smith	Depression	2
1	Alice	Smith	Anxiety	1
1	Alice	Smith	Self Compassion	1
2	Bob	Singh	Your Strengths	1
2	Bob	Singh	Depression	3
2	Bob	Singh	Anxiety	1
2	Bob	Singh	Self Compassion	2
4	Dan	Williams	Social Anxiety	2
4	Dan	Williams	Depression	3
4	Dan	Williams	Anxiety	2
4	Dan	Williams	Self Compassion	5
5	Eve	Brown	Depression	4
5	Eve	Brown	Mindfulness	1
6	Frank	Jones	Mindfulness	2
8	Heidi	Taylor	Depression	14

10. A) Provide a complete list of all user ID pairs such that the two users follow each other and share at least one interest. (Hint: MySQL has a CROSS JOIN operator, which returns the Cartesian product of rows from the joined tables)

```
SELECT user1.id, user2.id
FROM user AS user1 CROSS JOIN user AS user2
WHERE (SELECT COUNT(*) FROM user_follow WHERE following_user_id = user1.id
AND followed_user_id = user2.id)
AND (SELECT COUNT(*) FROM user_follow WHERE following_user_id = user2.id
AND followed_user_id = user1.id)
AND (SELECT COUNT(*) FROM user_interest WHERE user_interest.user_id =
user1.id AND user_interest.interest_id IN (SELECT interest_id FROM
user_interest WHERE user_id = user2.id));
```

Total of 2 rows returned

id	id
15	7
7	15

Note: Don’t penalise if they remove the duplicates

B) A query that returns a row for each instance in which two users, x and y, share an interest z. The result should consist of user x ID, user x first name, user y ID, user y first name and the interest name. The returned results should be such that only one of (x, y) or (y, x) is returned; for example, if row [1, Alice, 2, Bob, Tennis] is in the result set, then [2, Bob, 1, Alice, Tennis] should not be.

```
SELECT user1.id, user1.first_name, user2.id, user2.first_name,
interest.name
FROM user AS user1
INNER JOIN user_interest AS user_interest1 ON user1.id =
user_interest1.user_id
INNER JOIN user_interest AS user_interest2 ON user_interest1.interest_id =
user_interest2.interest_id
INNER JOIN user AS user2 ON user2.id = user_interest2.user_id
INNER JOIN interest ON user_interest1.interest_id = interest.id
WHERE user1.id > user2.id;
```

Total of 66 rows returned

id	first_name	id	first_name	name
7	Grace	1	Alice	Rock Climbing
9	Ian	1	Alice	Rock Climbing
9	Ian	7	Grace	Rock Climbing
14	Nick	1	Alice	Rock Climbing
14	Nick	7	Grace	Rock Climbing
14	Nick	9	Ian	Rock Climbing
6	Frank	2	Bob	The Simpsons
10	Judy	2	Bob	The Simpsons
10	Judy	6	Frank	The Simpsons
15	Olivia	2	Bob	The Simpsons
15	Olivia	6	Frank	The Simpsons
15	Olivia	10	Judy	The Simpsons
5	Eve	3	Charlie	Football
11	Kath	3	Charlie	Football
11	Kath	5	Eve	Football
16	Pat	3	Charlie	Football
16	Pat	5	Eve	Football