INSTAGRAM DATA MODELLING AND ANALYSIS

1) Update the caption of post_id =3

```
UPDATE POSTS SET caption ='Scrumptious Pizza' where post_id = 3;
```

	post_id [PK] integer	user_id integer	caption text	image_url character varying	created_at timestamp without time zone
1	1	1	Beautiful sunset	<https://www.example.com/sunset.jpg>	2023-08-14 12:36:42.726085
2	2	2	My new puppy	<https://www.example.com/puppy.jpg>	2023-08-14 12:36:42.726085
3	4	4	Throwback to my vacation	<https://www.example.com/vacation.jpg>	2023-08-14 12:36:42.726085
4	5	5	Amazing concert	<https://www.example.com/concert.jpg>	2023-08-14 12:36:42.726085
5	3	3	Scrumptious Pizza	<https://www.example.com/pizza.jpg>	2023-08-14 12:36:42.726085
6	6	1	Sureal	<https://www.example.com/scenary.jpg>	2023-08-18 11:44:01.576135

2) Count number of likes for each post.

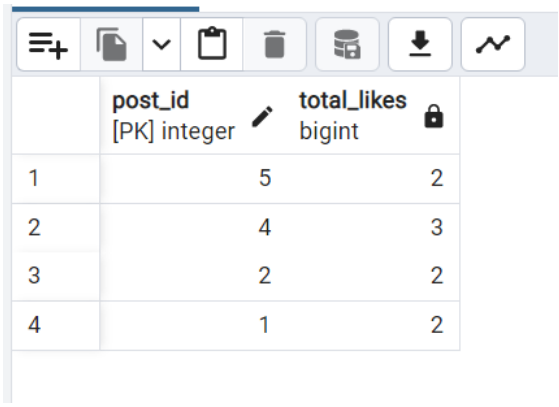
```
select p.post_id,count(l.like_id) as total_likes
from posts p left join likes l on p.post_id = l.post_id
group by p.post_id ;
```

(Left join has been used instead of inner join because we want to display even those posts which have 0 likes)

	post_id [PK] integer	total_likes bigint
1	3	1
2	5	2
3	4	3
4	6	0
5	2	2
6	1	2

3) Show only the posts with more than 2 likes.

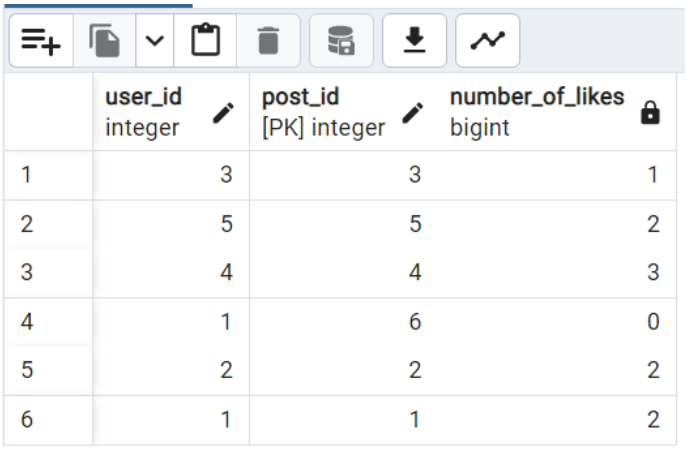
```
select p.post_id,count(l.like_id) as total_likes
from posts p left join likes l on p.post_id = l.post_id
group by p.post_id having count(l.like_id) >=2;
```



	post_id [PK] integer	total_likes bigint
1	5	2
2	4	3
3	2	2
4	1	2

4) Count number of likes for each post done by a user

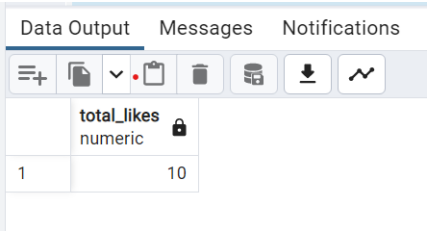
```
select p.user_id ,p.post_id,count(l.user_id) as number_of_likes from posts p left join likes l
on
p.post_id=l.post_id group by p.user_id,p.post_id;
```



	user_id integer	post_id [PK] integer	number_of_likes bigint
1	3	3	1
2	5	5	2
3	4	4	3
4	1	6	0
5	2	2	2
6	1	1	2

5) Find total number of likes for all posts

```
SELECT SUM(num_likes) AS total_likes
FROM (
SELECT COUNT(Likes.like_id) AS num_likes
 FROM Posts
 LEFT JOIN Likes ON Posts.post_id = Likes.post_id
 GROUP BY Posts.post_id
 ) as likes_by_post;
```



6) Find all the users who have commented on post_id =1

```
/*  1) Using joins*/

select users.user_id,users.name from users join comments on users.user_id= comments.user_id where comments.post_id=1;


/*  2) Using subquery */
select user_id,name from users where user_id in (
select user_id from comments where post_id=1);

/* 3) Using CTE  */

WITH UserComments AS (
    SELECT user_id
    FROM comments
    WHERE post_id = 1
)
SELECT u.user_id, u.name
FROM users u
JOIN UserComments uc ON u.user_id = uc.user_id;
```

	user_id [PK] integer	name character varying
1	2	Jane Doe
2	3	Bob Johnson

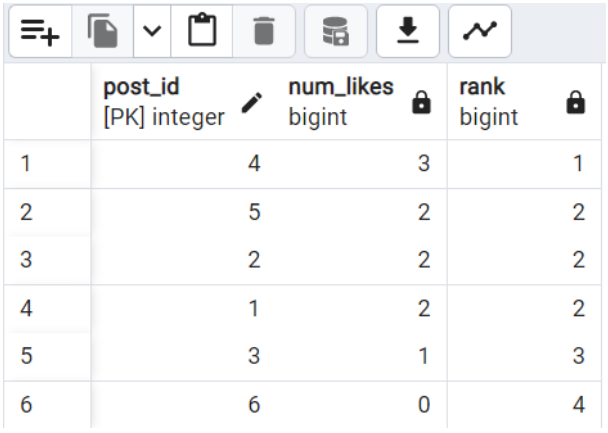
7) Rank the posts based on number of lines

```
SELECT post_id, num_likes, DENSE_RANK() OVER (ORDER BY num_likes DESC) AS rank
FROM (
    SELECT Posts.post_id, COUNT(Likes.like_id) AS num_likes
    FROM Posts
    LEFT JOIN Likes ON Posts.post_id = Likes.post_id
```

```
    GROUP BY Posts.post_id
) AS likes_by_post;


/* USING CTE */

WITH cte AS
(   SELECT Posts.post_id, COUNT(Likes.like_id) AS num_likes
    FROM Posts
    LEFT JOIN Likes ON Posts.post_id = Likes.post_id
    GROUP BY Posts.post_id
 )
  select  post_id,num_likes,  DENSE_RANK()  OVER(ORDER  BY  num_likes  DESC)  as
rank_by_likes
 from cte;
```



	post_id [PK] integer	num_likes bigint	rank bigint
1	4	3	1
2	5	2	2
3	2	2	2
4	1	2	2
5	3	1	3
6	6	0	4

8) Find all the posts and their comments using a common table expression

```
WITH post_comments AS (
    SELECT Posts.post_id, Posts.caption, Comments.comment_text
    FROM Posts
    LEFT JOIN Comments ON Posts.post_id = Comments.post_id
)
SELECT *
FROM post_comments;
```

	post_id integer	caption text	comment_text text
1	1	Beautiful sunset	Wow! Stunning.
2	1	Beautiful sunset	Beautiful colors.
3	2	My new puppy	What a cutie!
4	2	My new puppy	Aww, I want one.
5	3	Scrumptious Pizza	Yum!
6	4	Throwback to my vacation	Looks like an awesome trip.
7	5	Amazing concert	Wish I was there!
8	6	Sureal	[null]

9) Categorize the posts based on the number of likes

```
WITH CTE AS
( SELECT p.post_id,count(l.like_id) number_likes
  FROM posts as p
  JOIN likes as l on p.post_id = l.post_id
  GROUP BY p.post_id
)
SELECT post_id,number_likes,
CASE WHEN number_likes <=2 THEN 'low_likes'
WHEN number_likes =2 THEN 'few likes'
WHEN number_likes > 2 THEN 'lot of likes'
ELSE 'no data'
END like_category
FROM CTE;
```

	post_id [PK] integer	number_likes bigint	like_category text
1	3	1	low_likes
2	5	2	low_likes
3	4	3	lot of likes
4	2	2	low_likes
5	1	2	low_likes