Table creation

```
create table users (
            user_id serial primary key,
            name varchar(50) not null,
            email varchar(50) unique not null,
            mobile varchar(20) unique not null
);

create table posts(
            post_id serial primary key,
            user_id integer not null,
            caption text,
            image_url varchar(200),
            created_at timestamp default current_timestamp,
            foreign key (user_id) references users(user_id)
);

create table comments(
            comment_id serial primary key,
            post_id integer not null,
            user_id integer not null,
            comment_text text not null,
            created_at timestamp default current_timestamp,
            foreign key (user_id) references users(user_id),
            foreign key (post_id) references posts(post_id)
);
```

```
create table followers(
	follower_id serial primary key,
	follower_user_id integer not null,
	user_id integer not null,
	created_at timestamp default current_timestamp,
	foreign key(user_id) references users(user_id),
	foreign key (follower_user_id) references users(user_id)
);

create table likes(
	like_id serial primary key,
	post_id integer not null,
	user_id integer not null,
	created_at timestamp default current_timestamp,
	foreign key(post_id) references posts(post_id),
	foreign key(user_id) references users(user_id)
)
```

Data insert in table

```
-- Inserting into Users table
INSERT INTO users (name, email, mobile)
VALUES
   ('John Smith', 'johnsmith@gmail.com', '1234567890'),
   ('Jane Doe', 'janedoe@yahoo.com', '0987654321'),
   ('Bob Johnson', 'bjohnson@gmail.com', '1112223333'),
   ('Alice Brown', 'abrown@yahoo.com', '7042746013'),
   ('Mike Davis', 'mdavis@gmail.com', '5556667777');
```

```
-- Inserting into Posts table
INSERT INTO posts (user_id, caption, image_url)
VALUES
   (1, 'Beautiful sunset', '<https://www.example.com/sunset.jpg>'),
   (2, 'My new puppy', '<https://www.example.com/puppy.jpg>'),
   (3, 'Delicious pizza', '<https://www.example.com/pizza.jpg>'),
   (4, 'Throwback to my vacation', '<https://www.example.com/vacation.jpg>'),
   (5, 'Amazing concert', '<https://www.example.com/concert.jpg>');

-- Inserting into Comments table
INSERT INTO comments (post_id, user_id, comment_text)
VALUES
   (1, 2, 'Wow! Stunning.'),
   (1, 3, 'Beautiful colors.'),
   (2, 1, 'What a cutie!'),
   (2, 4, 'Aww, I want one.'),
   (3, 5, 'Yum!'),
   (4, 1, 'Looks like an awesome trip.'),
   (5, 3, 'Wish I was there!');
```

```sql
-- Inserting into Likes table
INSERT INTO likes (post_id, user_id)
VALUES
    (1, 2),
    (1, 4),
    (2, 1),
    (2, 3),
    (3, 5),
    (4, 1),
    (4, 2),
    (4, 3),
    (5, 4),
    (5, 5);

-- Inserting into Followers table
INSERT INTO followers (user_id, follower_user_id)
VALUES
    (1, 2),
    (2, 1),
    (1, 3),
    (3, 1),
    (1, 4),
    (4, 1),
    (1, 5),
    (5, 1);
```

Analytics Example

-- Selecting all the posts where user_id is 1

```
SELECT *
FROM Posts
WHERE user_id = 1;
-- Selecting all the posts and ordering them by created_at in descending order SELECT * FROM Posts ORDER BY created_at DESC;
```

-- Counting the number of likes for each post and showing only the posts with more than 2 likes

```
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
HAVING COUNT(Likes.like_id) > 2;
```

-- Finding the 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;
```

```sql
-- Finding all the users who have commented on post_id 1
SELECT name
FROM Users
WHERE user_id IN (
    SELECT user_id
    FROM Comments
    WHERE post_id = 1
);
```

```sql
-- Ranking the posts based on the number of likes
SELECT post_id, num_likes, 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;
```

```sql
-- Finding all the posts and their comments using a Common Table Expression (CTE)
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;
```

-- Categorizing the posts based on the number of likes

```
SELECT
    post_id,
    CASE
        WHEN num_likes = 0 THEN 'No likes'
        WHEN num_likes < 5 THEN 'Few likes'
        WHEN num_likes < 10 THEN 'Some likes'
        ELSE 'Lots of likes'
    END AS like_category
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;
```

-- Finding all the posts created in the last month

```
SELECT *
FROM Posts
WHERE created_at >= CAST(DATE_TRUNC('month', CURRENT_TIMESTAMP - INTERVAL '1 month') AS DATE);
```