# 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!');

-- 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;

**-- 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

);

**-- 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;

**-- 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);