

CONNECT SPHERE

BY

MANNARSH SINGH BALI
R024

DBMS PROJECT



INDEX

INTRODUCTION

STORYLINE

CONTRIBUTION

ER DIAGRAM

RELATION SCHEMA

QUIRES

LEARNING

CONCLUSION



STORYLINE



ConnectSphere: Revolutionizing Social Media Interactions

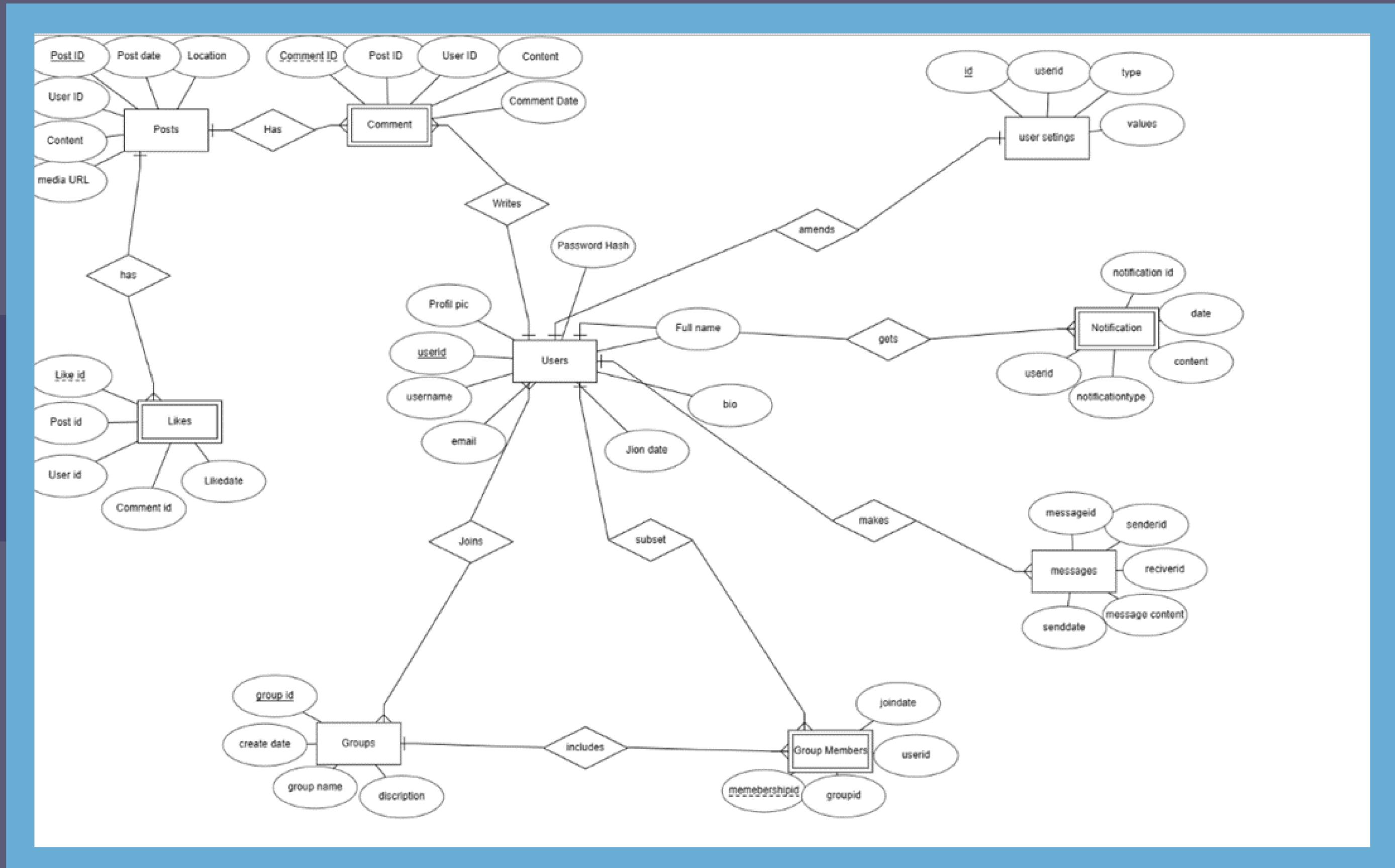
- **Innovative DBMS Project:** Tailored platform enhancing user engagement and interaction.
- **Centralized Social Hub:** Dynamic interactions, content sharing, and community building on a global scale.
- **Comprehensive Database Design:** Meticulously crafted to store user interactions, including posts, comments, likes, and more.
- **Empowering Users:** Seamless content creation, personalized connections, and group participation.
- **Advanced Features:** Privacy management, content recommendations, and real-time notifications for full control over online presence.
- **Goal:** To weave a more connected, accessible, and engaging digital social fabric.

CONTRIBUTION

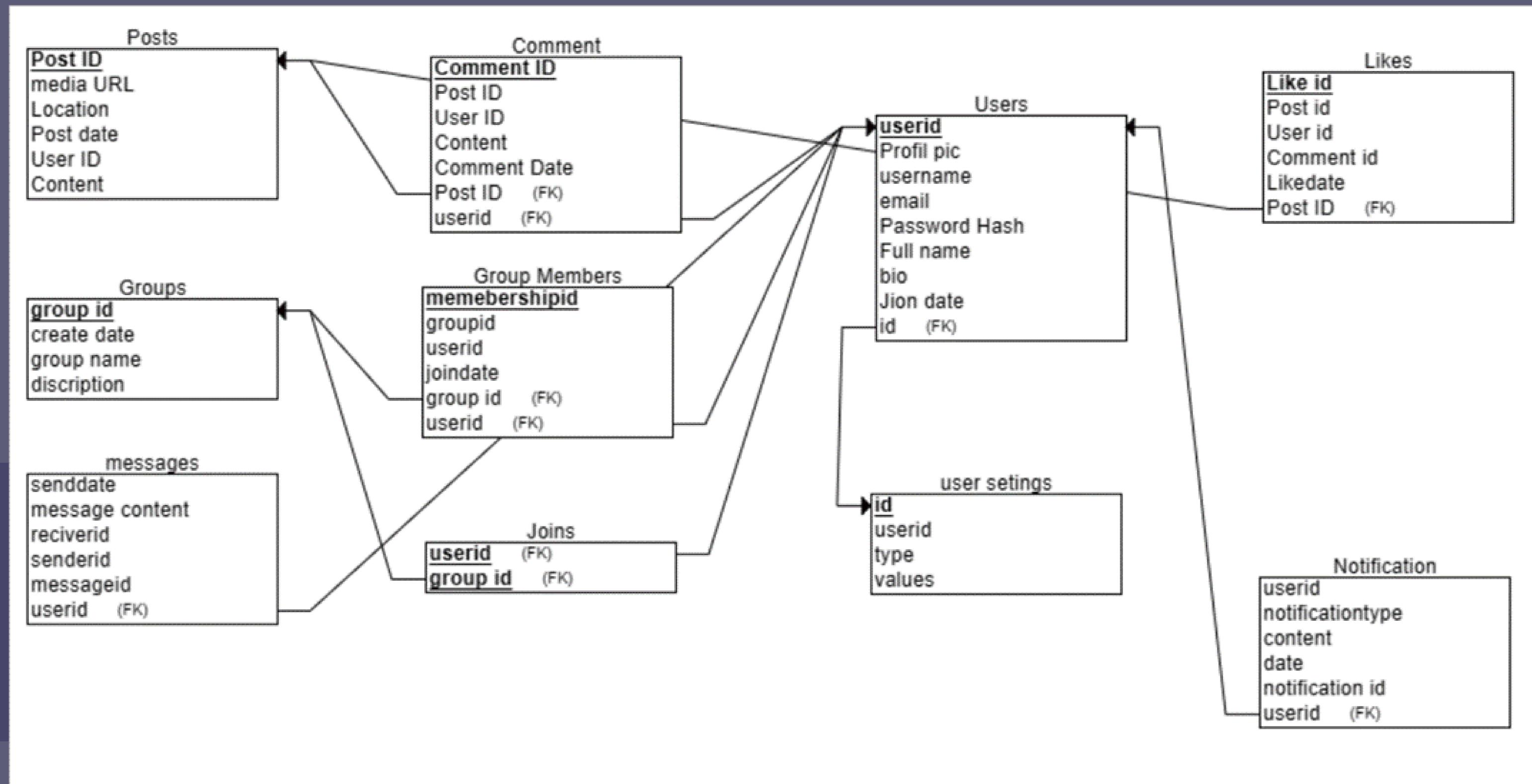
Group member	Contribution
Mannarsh Singh Bali	<ul style="list-style-type: none">• Data collection and organization• Relational model• Table Creation• Data Insertion• Attributes Tables• ER diagram



ER DIAGRAM



RELATIONAL SCHEMA



QUERIES

EASY	MEDIUM	HARD
<ul style="list-style-type: none">• List all users.	<ul style="list-style-type: none">• Count the number of posts each user has made.	<ul style="list-style-type: none">• List users who have not made any posts yet.
<ul style="list-style-type: none">• Find all posts made by a specific user (UserId = 1).	<ul style="list-style-type: none">• Find the total number of likes for each post.	<ul style="list-style-type: none">• Find the most recent post of each user.
<ul style="list-style-type: none">• Display all comments on a specific post (PostId = 2).	<ul style="list-style-type: none">• Retrieve user details along with the number of followers they have.	<ul style="list-style-type: none">• Identify users who follow back all their followers.
-----	<ul style="list-style-type: none">• List all likes on posts made by a specific user (UserId = 1).	-----



QUERIES

QUERY 1

Question: How can we list all the users in our social media platform?

*SELECT *
FROM Users;*



OUTPUT

Output								
UserId	Username	Email	PasswordHash	FullName	Bio	ProfilePic	JoinDate	
1	johnDoe	john.doe@email.com	hashed_password_1	John Doe	Just a regular Joe.	profilepic1.jpg	2023-01-01	
2	janeDoe	jane.doe@email.com	hashed_password_2	Jane Doe	Love hiking and outdoor adventures.	profilepic2.jpg	2023-01-02	
3	coolUser123	cooluser123@email.com	hashed_password_3	Cool User	Always up for a cool adventure.	profilepic3.jpg	2023-01-03	
4	travelBug	travel.bug@email.com	hashed_password_4	Travel Bug	Exploring the world, one country at a time.	profilepic4.jpg	2023-01-04	
5	bookworm	bookworm@email.com	hashed_password_5	Book Worm	A room without books is like a body without a soul.	profilepic5.jpg	2023-01-05	
6	gamerGuy	gamer.guy@email.com	hashed_password_6	Gamer Guy	Living one game at a time.	profilepic6.jpg	2023-01-06	
7	codingNinja	coding.ninja@email.com	hashed_password_7	Coding Ninja	Writing code and fixing bugs.	profilepic7.jpg	2023-01-07	
8	fitnessFreak	fitness.freak@email.com	hashed_password_8	Fitness Freak	Sweat now, shine later.	profilepic8.jpg	2023-01-08	
9	foodie	foodie@email.com	hashed_password_9	Foodie	Life is short, eat	profilepic9.jpg	2023-01-09	

QUERIES

QUERY 2

QUESTION: HOW DO WE FIND ALL POSTS MADE BY THE USER WITH USERID = 1?

```
SELECT *  
FROM POSTS  
WHERE UserID = 1;
```

OUTPUT

Output

PostId	UserId	Content	MediaURL	PostDate	Location
1	1	Loving the new cafe in town!		2023-01-02 10:30:00	Town Center

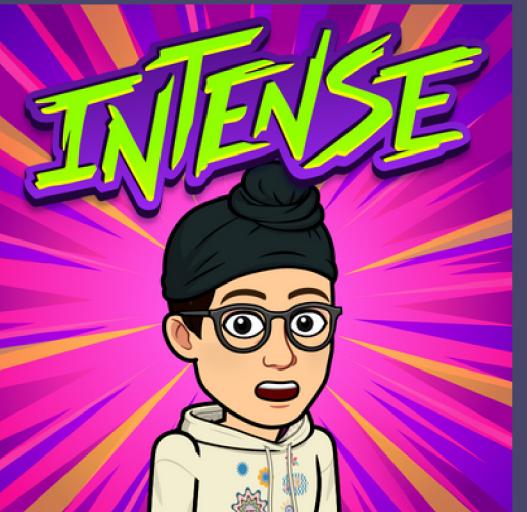


QUERIES

QUERY 3

Question: How can we count the total number of posts made by each user?

```
SELECT UserId,  
COUNT(*) AS NumberOfPosts  
FROM Posts  
GROUP BY UserId;
```



OUTPUT

UserId	NumberOfPosts
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

QUERIES

QUERY 4

Question: How do we find users who haven't made any posts yet?

```
SELECT *
FROM Users
WHERE UserId NOT IN(SELECT DISTINCT
UserId FROM Posts);
```



OUTPUT

Output

SQL query successfully executed. However, the result set is empty.

QUERIES

QUERY 5

Question: How can we retrieve the most recent post made by each user?

```
SELECT p.UserId, MAX(p.PostDate)  
as MostRecentPost  
FROM Posts p  
GROUP BY p.UserId;
```



OUTPUT

Output

UserId	MostRecentPost
1	2023-01-02 10:30:00
2	2023-01-03 09:15:00
3	2023-01-04 08:45:00
4	2023-01-05 17:20:00
5	2023-01-06 20:00:00
6	2023-01-07 13:00:00
7	2023-01-08 07:30:00
8	2023-01-09 19:00:00
9	2023-01-10 21:00:00
10	2023-01-11 15:00:00
11	2023-01-12 11:00:00
12	2023-01-13 18:30:00
13	2023-01-14 14:00:00
14	2023-01-15 22:00:00
15	2023-01-16 16:00:00

QUERIES

QUERY 6

Question: How do we show all comments for the post with PostId = 2?

```
SELECT *  
FROM Comments  
WHERE PostId = 2;
```

OUTPUT

Output				
CommentId	PostId	UserId	Content	CommentDate
2	2	1	Looks amazing! Have fun!	2023-01-03 10:00:00

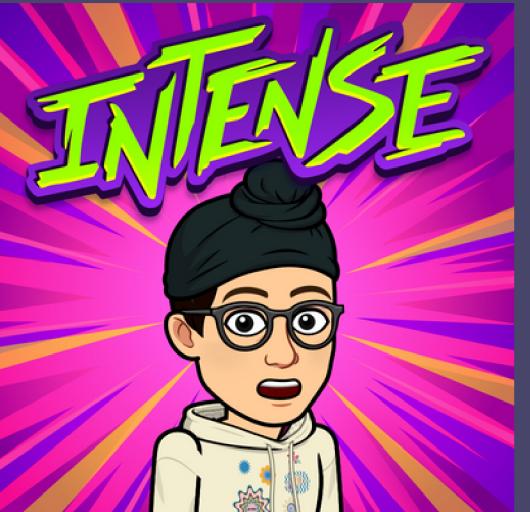


QUERIES

QUERY 7

Question: How can we list all likes for posts made by the user with UserId = 1?

```
SELECT l.*  
FROM Likes l  
JOIN Posts p ON l.PostId = p.PostId  
WHERE p.UserId = 1;
```



OUTPUT

Output

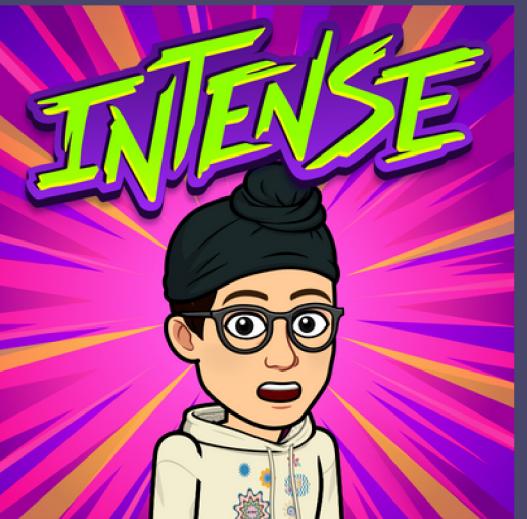
LikId	UserId	PostId	CommentId	LikeDate
1	2	1		2023-01-02 12:00:00

QUERIES

QUERY 8

Question: How can we count the total number of likes on each post?

```
SELECT PostId, COUNT(*) AS  
TotalLikes  
FROM Likes  
GROUP BY PostId;
```



OUTPUT

PostId	TotalLikes
7	7
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1

QUERIES

QUERY 9

Question: How do we find each user's details and their follower count?

```
SELECT u.*,
COUNT(f.FollowingUserId)
AS NumberOfFollowers
FROM Users u
LEFT JOIN Followers f ON
u.UserId = f.FollowedUserId
GROUP BY u.UserId;
```

OUTPUT

Output									
UserId	Username	Email	PasswordHash	FullName	Bio	ProfilePic	JoinDate	NumberOfFollowers	
1	johnDoe	john.doe@email.com	hashed_password_1	John Doe	Just a regular Joe.	profilepic1.jpg	2023-01-01	1	
2	janeDoe	jane.doe@email.com	hashed_password_2	Jane Doe	Love hiking and outdoor adventures.	profilepic2.jpg	2023-01-02	1	
3	coolUser123	cooluser123@email.com	hashed_password_3	Cool User	Always up for a cool adventure.	profilepic3.jpg	2023-01-03	1	
4	travelBug	travel.bug@email.com	hashed_password_4	Travel Bug	Exploring the world, one country at a time.	profilepic4.jpg	2023-01-04	1	
5	bookworm	bookworm@email.com	hashed_password_5	Book Worm	A room without books is like a body without a soul.	profilepic5.jpg	2023-01-05	1	



QUERIES

QUERY 10

Question: How can we find users who follow back every user that follows them?

```
SELECT DISTINCT f1.FollowedUserId  
FROM Followers f1  
INNER JOIN Followers f2 ON  
f1.FollowedUserId = f2.FollowingUserId  
AND f1.FollowingUserId =  
f2.FollowedUserId;
```



OUTPUT

Output

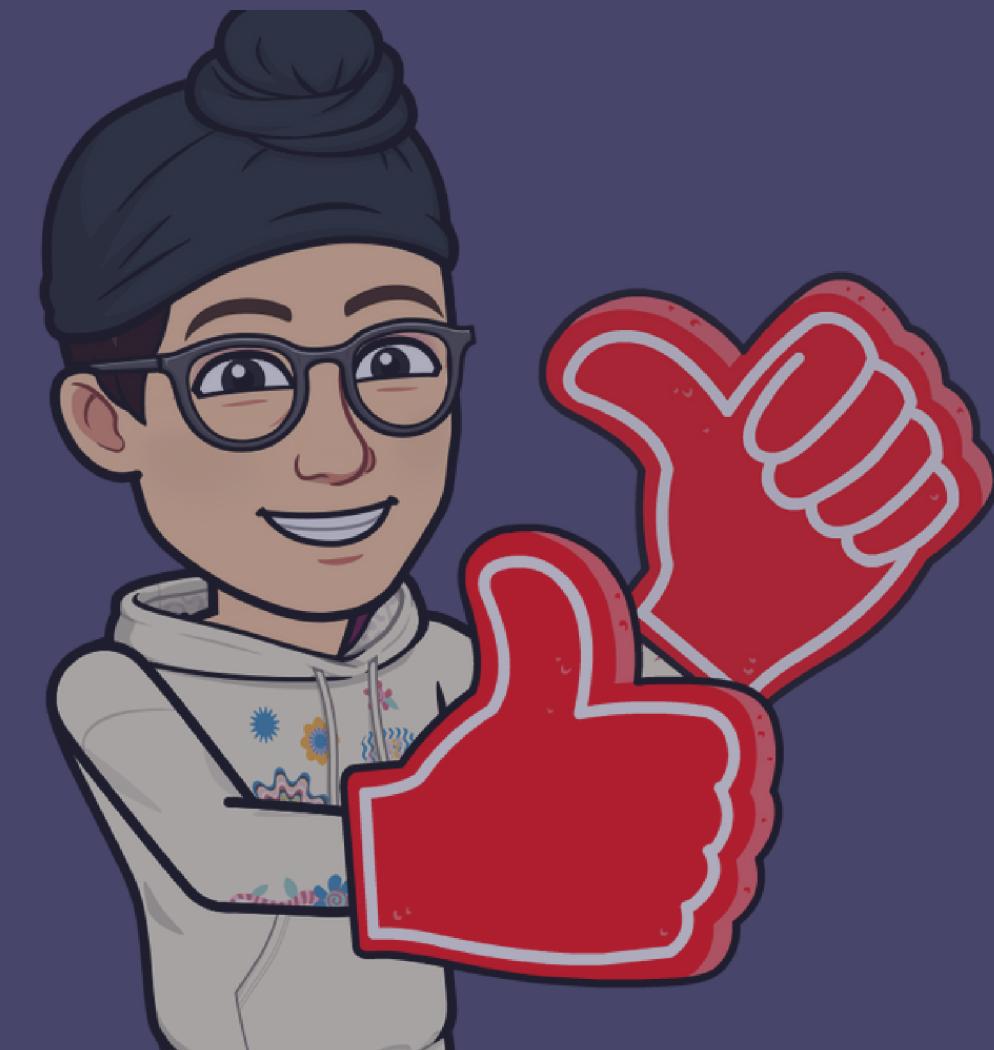
FollowedUserId
2
1

BEYOND LERANING

ADVANCED DATABASE DESIGN AND NORMALIZATION: GAINED A DEEPER UNDERSTANDING OF COMPLEX DATABASE CONCEPTS CRUCIAL FOR ENSURING DATA INTEGRITY AND OPTIMIZING EFFICIENCY IN A DYNAMIC SOCIAL MEDIA ENVIRONMENT.

BACKEND FOCUS AND STRUCTURED INFORMATION: SHIFTED EMPHASIS TO BACKEND DEVELOPMENT TO MANAGE AND FACILITATE COMPLEX INTERACTIONS LIKE USER CONNECTIONS AND CONTENT SHARING, HIGHLIGHTING THE IMPORTANCE OF WELL-ORGANIZED DATA.

CONTINUOUS LEARNING IN SOFTWARE DEVELOPMENT: THIS PROJECT UNDERSCORED THE NECESSITY OF ONGOING EDUCATION AND FLEXIBILITY IN APPLYING THEORETICAL KNOWLEDGE TO ADDRESS PRACTICAL, REAL-WORLD PROBLEMS EFFECTIVELY



THANK YOU