A Report on

Database Management

System in

Social Media

Submitted By:

Yash Dukhande (N060)

Under the guidance of

Prof .Shakila Shaikh

Shri Vile Parle Kelvani Mandal's

Mukesh Patel School of Technology and Management Engineering

Department of Computer Engineering

Vile Parle (w), Mumbai- 400056

Table Of Contents

1) Abstract

2) Introduction

3) Components

4) ER Diagram

5) Relational Model

6) Normalization

7) Queries

8) Self Learning

9) Challenges Faced

10) Conclusion

Abstract

This document dives deep into the intricacies of crafting an advanced database management system tailored specifically for social media platforms. Its core mission revolves around efficiently organizing, retaining, and managing the vast array of data stemming from user engagements within the realm of social media.

One of the pivotal focuses of this paper is on the design principles that underpin relational databases. It delves into how these principles, particularly normalization, serve as a strategic approach to streamline data retrieval processes and enhance storage efficiency by minimizing redundancy. By adhering to these principles, the overall performance of the database can be optimized, ensuring that data is stored logically and effectively.

Moreover, the document sheds light on the indispensable role of Structured Query Language (SQL) in manipulating databases. It emphasizes how SQL enables the execution of intricate queries vital for supporting key functionalities of social media platforms, including monitoring user behavior, managing post interactions such as likes and comments, and delineating social networks among users.

In elucidating the system's foundational design, the paper thoroughly examines the development of Entity-Relationship (ER) diagrams and associated schema diagrams. These visual representations serve not only as aids for conceptualization but also as essential tools for refining the structure of the database. By visually mapping out connections and dependencies between data entities, ER diagrams facilitate the optimization of the database for complex queries crucial to the platform's operation.

By adopting a holistic approach to database management, encompassing conceptual design through to SQL execution, the document furnishes developers and researchers with a robust blueprint for constructing scalable and efficient databases tailored to the demands of social media platforms. It bridges the gap between theory and practice by addressing both theoretical concepts and practical implementation strategies, offering valuable insights into building high-performing database systems capable of supporting the dynamic and data-intensive nature of social media interactions.

Introduction

In today's digital landscape, social media platforms have become bustling hubs of online activity, weaving together a rich tapestry of user-generated content and facilitating diverse interactions among users. This project is dedicated to crafting a database management system specifically tailored to meet the intricate demands unique to social media ecosystems. The envisioned system is engineered to adeptly handle the storage, organization, and retrieval of a broad spectrum of data types, spanning from detailed user profiles to the dynamic content of posts, and from the myriad of user interactions—such as likes, comments, and shares—to the extensive network of connections that define the social fabric of the platform.

At the core of this initiative lies the ambition to construct a database system that not only boasts robust architecture but also exhibits flexibility and efficiency in its operations. Such a system is indispensable for executing sophisticated data queries that power essential features of social media platforms, including content recommendation algorithms, trend analysis, and the monitoring of user engagement metrics. These functionalities are pivotal in refining the user experience, delivering personalized content, and fostering a deeper understanding of user behavior and preferences.

The ultimate objective of this project transcends mere data storage; it seeks to establish a repository that serves as a treasure trove of information, enabling intricate analytical operations and insights crucial for strategic decision-making and cultivating a vibrant social media environment. By ensuring the system's scalability, the project anticipates the burgeoning growth of data volumes and user interactions, ensuring the platform can adeptly manage an expanding universe of data without compromising on performance or user experience.

Realizing this vision demands a meticulous approach to database system design, incorporating advanced techniques in data modeling, system architecture, and query optimization. The system must be architected with foresight for extensibility, allowing for the seamless integration of new data types and interactions as the platform evolves. Furthermore, it must integrate robust data security measures to safeguard user information and uphold privacy compliance.

In summary, this project embodies a holistic endeavor to engineer a database management system that serves as the nucleus of the social media experience, capable of harnessing and interpreting the vast array of data generated by users. Through this endeavor, the project aims to establish the groundwork for a next-generation social media platform that is intelligent, responsive, and scalable, elevating the way individuals connect, share, and engage online.

Components

Table Bookmarks-

Column post_id: foreign key for every post

Column user_id:foreign key for every user Column
created_at: has timestamp for every bookmark

Table Comment_likes:

Column user_id:foreign key for every user

Column comment_id:foreign key for every comment Column
created_at: has timestamp for every comment_likes

Table Comments:

Column user_id:foreign key for every user

Column post_id:foreign key for every post

Column comment_id:foreign key for every comment

Column comment_text: shows the comments done by every user

Column created_at: has timestamp for every comment

Table Follows:

Column follower_id:primary key for every follower

Column followee_id:foreign key for every followee

Column created_at: has timestamp for every follows

Table Hashtag_follows:

Column user_id:foreign key for every user

Column hashtag_id:foreign key for every hashtag

Column created_at: has timestamp for every hashtag

Table Hashtags:

Column hashtag_name :name for every hashtag

Column hashtag_id:foreign key for every hashtag

Column created_at: has timestamp for every hashtag

Table login:

Column user_id:foreign key for every user

Column login_id:primary key for every hashtag

Ip: contains ips of every user

Column created_at: has timestamp for every login

Table photos:

Column photo_id:primary key for every photo

Column post_id:foreign key for every post

Column photo_url: contains url of every photo

Column size: contains size of photos

Column created_at: has timestamp for every photo
Table videos:
Column video_id:primary key for every video
Column post_id:foreign key for every post
Column video_url: contains url of every video
Column size: contains size of photos Column
created_at: has timestamp for every video

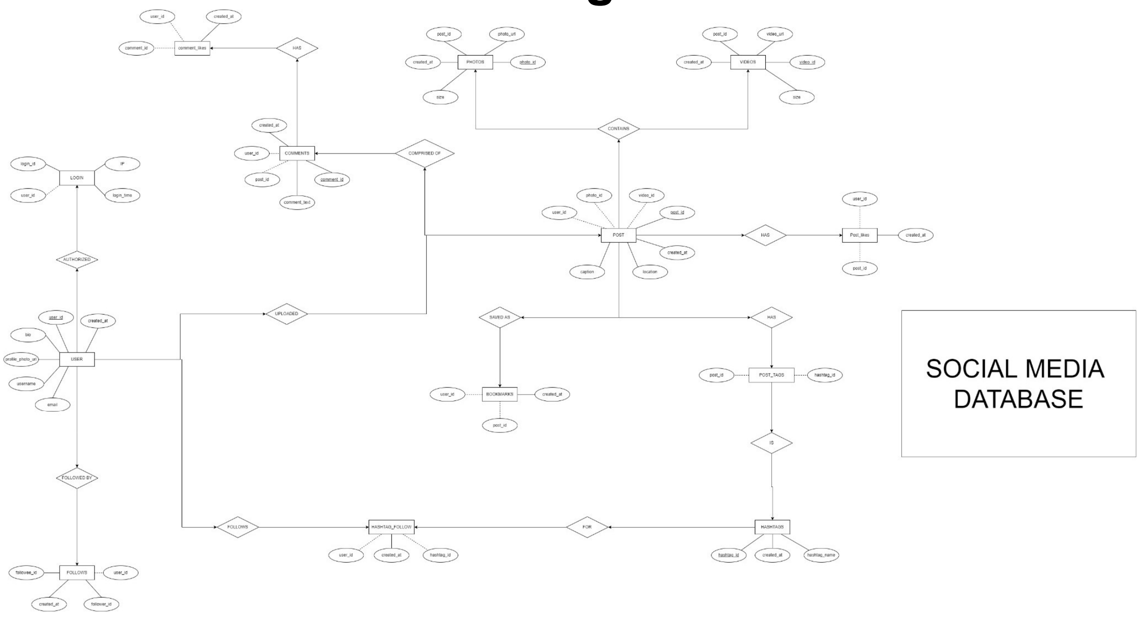
Table post:
Column video_id:primary key for every video
Column post_id:foreign key for every post
Column photo_id: foreign key for every photo
Column user_id: foreign key for every user
Column caption: contains caption of post
Column location:contains location of post
Column created_at: has timestamp for every post

Table Post_likes:
Column post_id: foreign key for every post
Column user_id:foreign key for every user
Column created_at: has timestamp for every post

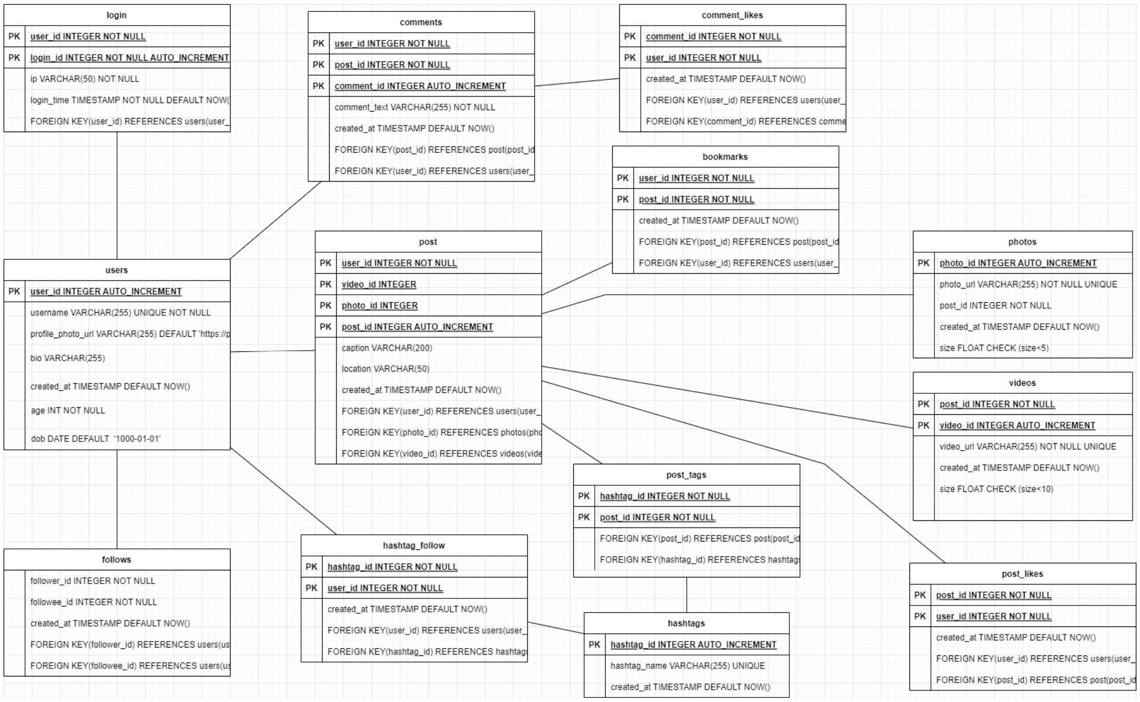
Table Post_likes:
Column post_id: foreign key for every post
Column hashtag_id:foreign key for every hashtag

Table users:
Column user_id: foreign key for every user
Column username: contains username of user
Column bio :contains bio of user
Column email: contains email of user
Column created_at: has timestamp for every post

ER Diagram



Schema Diagram



	login
PK	user_id INTEGER NOT NULL
PK	login_id INTEGER NOT NULL AUTO_INCREMENT
	ip VARCHAR(50) NOT NULL
	login_time TIMESTAMP NOT NULL DEFAULT NOW(
	FOREIGN KEY(user_id) REFERENCES users(user_

	comments
PK	user_id INTEGER NOT NULL
PK	post_id INTEGER NOT NULL
PK	comment_id INTEGER AUTO_INCREMENT
	comment_text VARCHAR(255) NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(post_id) REFERENCES post(post_id
	FOREIGN KEY(user_id) REFERENCES users(user_

	comment_likes
PK	comment_id INTEGER NOT NULL
PK	user_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(user_id) REFERENCES users(user_
	FOREIGN KEY(comment_id) REFERENCES comme

	bookmarks
PK	user_id INTEGER NOT NULL
PK	post_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(post_id) REFERENCES post(post_id
	FOREIGN KEY(user_id) REFERENCES users(user_

	users
PK	user_id INTEGER AUTO_INCREMENT
	username VARCHAR(255) UNIQUE NOT NULL
	profile_photo_url VARCHAR(255) DEFAULT 'https://p
	bio VARCHAR(255)
	created_at TIMESTAMP DEFAULT NOW()
	age INT NOT NULL
	dob DATE DEFAULT '1000-01-01'

	post
PK	user_id INTEGER NOT NULL
PK	video_id INTEGER
PK	photo_id INTEGER
PK	post_id INTEGER AUTO_INCREMENT
	caption VARCHAR(200)
	location VARCHAR(50)
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(user_id) REFERENCES users(user_
	FOREIGN KEY(photo_id) REFERENCES photos(pho
	FOREIGN KEY(video_id) REFERENCES videos(vide

	photos
PK	photo_id INTEGER AUTO_INCREMENT
	photo_url VARCHAR(255) NOT NULL UNIQUE
	post_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	size FLOAT CHECK (size<5)

	videos
PK	post_id INTEGER NOT NULL
PK	video_id INTEGER AUTO_INCREMENT
	video_url VARCHAR(255) NOT NULL UNIQUE
	created_at TIMESTAMP DEFAULT NOW()
	size FLOAT CHECK (size<10)

	post_tags
PK	hashtag_id INTEGER NOT NULL
PK	post_id INTEGER NOT NULL
	FOREIGN KEY(post_id) REFERENCES post(post_id
	FOREIGN KEY(hashtag_id) REFERENCES hashtags

	follows
	follower_id INTEGER NOT NULL
	followee_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(follower_id) REFERENCES users(us
	FOREIGN KEY(followee_id) REFERENCES users(us

	hashtag_follow
PK	hashtag_id INTEGER NOT NULL
PK	user_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(user_id) REFERENCES users(user_
	FOREIGN KEY(hashtag_id) REFERENCES hashtags

	hashtags
PK	hashtag_id INTEGER AUTO_INCREMENT
	hashtag_name VARCHAR(255) UNIQUE
	created_at TIMESTAMP DEFAULT NOW()

	post_likes
PK	post_id INTEGER NOT NULL
PK	user_id INTEGER NOT NULL
	created_at TIMESTAMP DEFAULT NOW()
	FOREIGN KEY(user_id) REFERENCES users(user_
	FOREIGN KEY(post_id) REFERENCES post(post_id

Normalization

Normalization is a critical process in database design that structures a database in a way that reduces redundancy and dependency by decomposing tables. This methodology enhances the database's logical integrity and efficiency. The process of normalization in our social media database management system is meticulously carried out through several normalization forms—First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF). Each form serves a specific purpose in refining the database structure, making it more efficient and reliable for handling the complex data associated with social media platforms.

First Normal Form (1NF)

To achieve 1NF, the database tables are structured to ensure that each column contains atomic values, and each record holds a unique identifier. This means that the database does not allow multi-valued attributes or composite attributes; each field contains only the smallest possible data unit, enhancing query performance and data integrity. For example, in the users table, each user is identified by a unique user_id, and all other attributes like username, email, and dob contain single, indivisible values. This setup prevents anomalies and inconsistencies during data insertion, update, and deletion.

Second Normal Form (2NF)

Achieving 2NF involves restructuring the database to eliminate partial dependency; that is, ensuring that all attributes in a table are solely dependent on the primary key. This step requires that the database is already in 1NF. In our social media database, tables like post_likes and comment_likes are designed to reference posts and comments respectively through foreign keys. By doing so, we ensure that non-key columns (created_at timestamps in these tables) are fully functionally dependent on their primary keys (user_id and post_id or comment_id), eliminating partial dependencies and thereby enhancing data consistency.

Third Normal Form (3NF)

To reach 3NF, the database must be in 2NF, and all its tables must be designed to eliminate transitive dependency. This means that non-key attributes cannot depend on other non-key attributes. Each table's attributes must be directly dependent on the primary key alone. For instance, in our users table, attributes like username and email are directly dependent on user_id and not on any other non-key attribute. This structure ensures that the database does not store redundant data and that each piece of information is located in only one place, making the database more efficient and easier to maintain.

Queries

Database creation and insertion of data ->

```
DROP database social_media;
CREATE DATABASE social_media;
USE social_media;

CREATE TABLE users ( user_id INTEGER
   AUTO_INCREMENT PRIMARY KEY, username
   VARCHAR(255) UNIQUE NOT NULL,
   profile_photo_url VARCHAR(255) DEFAULT 'https://picsum.photos/100',
   bio VARCHAR(255),
   created_at TIMESTAMP DEFAULT NOW()
);

ALTER TABLE users
ADD email VARCHAR(30) NOT NULL;

CREATE TABLE photos ( photo_id INTEGER
   AUTO_INCREMENT PRIMARY KEY, photo_url
   VARCHAR(255) NOT NULL UNIQUE, post_id
         INTEGER NOT NULL, created_at TIMESTAMP
   DEFAULT NOW(), size FLOAT CHECK (size<5)
);

CREATE TABLE videos ( video_id INTEGER
  AUTO_INCREMENT PRIMARY KEY, video_url
  VARCHAR(255) NOT NULL UNIQUE, post_id INTEGER
  NOT NULL, created_at TIMESTAMP DEFAULT NOW(),
  size FLOAT CHECK (size<10)
);

CREATE TABLE post ( post_id INTEGER
   AUTO_INCREMENT PRIMARY KEY, photo_id
   INTEGER, video_id INTEGER, user_id INTEGER NOT
   NULL, caption VARCHAR(200), location VARCHAR(50)
   ,
   created_at TIMESTAMP DEFAULT NOW(), FOREIGN
   KEY(user_id) REFERENCES users(user_id),
   FOREIGN KEY(photo_id) REFERENCES photos(photo_id),
   FOREIGN KEY(video_id) REFERENCES videos(video_id)
);
CREATE TABLE comments ( comment_id INTEGER
   AUTO_INCREMENT PRIMARY KEY, comment_text
   VARCHAR(255) NOT NULL, post_id INTEGER NOT NULL,
   user_id INTEGER NOT NULL, created_at TIMESTAMP
```

```
    DEFAULT NOW(), FOREIGN KEY(post_id) REFERENCES
    post(post_id),
    FOREIGN KEY(user_id) REFERENCES users(user_id)
);

CREATE TABLE post_likes ( user_id INTEGER NOT
    NULL, post_id INTEGER NOT NULL, created_at
    TIMESTAMP DEFAULT NOW(), FOREIGN
    KEY(user_id) REFERENCES users(user_id),
    FOREIGN KEY(post_id) REFERENCES post(post_id),
    PRIMARY KEY(user_id, post_id)
);

CREATE TABLE comment_likes ( user_id
    INTEGER NOT NULL, comment_id
    INTEGER NOT NULL, created_at
    TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(user_id),
    FOREIGN KEY(comment_id) REFERENCES comments(comment_id),
    PRIMARY KEY(user_id, comment_id)
);

CREATE TABLE follows ( follower_id
    INTEGER NOT NULL, followee_id
    INTEGER NOT NULL, created_at
    TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(follower_id) REFERENCES users(user_id),
    FOREIGN KEY(followee_id) REFERENCES users(user_id),
    PRIMARY KEY(follower_id, followee_id)
);

CREATE TABLE hashtags ( hashtag_id INTEGER
  AUTO_INCREMENT PRIMARY KEY, hashtag_name
  VARCHAR(255) UNIQUE, created_at TIMESTAMP
  DEFAULT NOW()
);

CREATE TABLE hashtag_follow (
        user_id INTEGER NOT NULL,
    hashtag_id INTEGER NOT NULL, created_at
    TIMESTAMP DEFAULT NOW(), FOREIGN
    KEY(user_id) REFERENCES users(user_id),
    FOREIGN KEY(hashtag_id) REFERENCES
    hashtags(hashtag_id), PRIMARY KEY(user_id,
    hashtag_id)
);
```

```
CREATE TABLE post_tags ( post_id
    INTEGER NOT NULL, hashtag_id
    INTEGER NOT NULL,
    FOREIGN KEY(post_id) REFERENCES post(post_id),
    FOREIGN KEY(hashtag_id) REFERENCES hashtags(hashtag_id),
    PRIMARY KEY(post_id, hashtag_id)
);

CREATE TABLE bookmarks ( post_id INTEGER NOT
  NULL, user_id INTEGER NOT NULL, created_at
  TIMESTAMP DEFAULT NOW(), FOREIGN
  KEY(post_id) REFERENCES post(post_id),
  FOREIGN KEY(user_id) REFERENCES users(user_id),
  PRIMARY KEY(user_id, post_id)
);

CREATE TABLE login ( login_id INTEGER NOT NULL
  AUTO_INCREMENT PRIMARY KEY, user_id INTEGER NOT
  NULL, ip VARCHAR(50) NOT NULL,
  login_time TIMESTAMP NOT NULL DEFAULT NOW(),
  FOREIGN KEY(user_id) REFERENCES users(user_id)
);

# QUERIES FOR INSERTION IN DATABASE
INSERT INTO users(username, profile_photo_url, bio,email) VALUES
("kanavphull","https://pbs.twimg.com/profile_images/1386885117428191232/70SyHOxP_40
0x400.jpg","Hedonist yet Spiritual || IT at NITJ 2024",'as1mobiles@gmail.com'),
('raj gupta' , '/klsadf893724:f//432' , 'soon to be updated','admin@1shopbuy.com'),
('Sahib Singh' ,
'https://pbs.twimg.com/profile_images/1465003815820693506/gbTJoe66_400x400.jpg' ,"Life
is a journey, It drives you MAD.|| IT NITJ'24",'12angeldesignworld@gmail.com');

INSERT INTO users(username, profile_photo_url, bio,email) VALUES
('Sakshi Warandani' ,
"https://vader.news/__export/1612817390103/sites/gadgets/img/2021/02/08/ian_somerhalde
r_vampires.jpg_246448593.jpg" , "NITJ wish me on 23 jan",'deepak@24sevenindia.com'),
("Omnicron Larson", "/sdfvsdf", "Heart Stealer",'101cartinfo@gmail.com'),
("dettol sharma", "/sdfvsdf", "Dettol Stealer",'the.yellow.gold@gmail.com'),
('sunil' , '/yisadf324//' , 'hotel manageemnt ','deepak@24sevenindia.com'),
('sanjay' , '/fduiahj43' , 'football lover','deepak@24sevenindia.com' ),
('Axel Sivert Anker' , '/adaskjnas','Norwegian','gazender.686@gmail.com'),
('Steven','/acdsccsdc', 'living life my way','sravi07@yahoo.com'),
('Jack','https://picsum.photos/100','Welcome To My Profile','contact@21fools.com'),
('Oliver','https://picsum.photos/101','Official Account','the.yellow.gold@gmail.com'),
('James','https://picsum.photos/102','Wish Me On 3 October','contact@21fools.com'),
('Charlie','https://picsum.photos/103','aap yha aae kisliye','sravi07@yahoo.com'),
('Harris','https://picsum.photos/104','Sanskari Ladka','pawan.modi1@gmail.com'),
('Lewis','https://picsum.photos/105','aapne bulaya isilye','as1mobiles@gmail.com'),
('Leo','https://picsum.photos/106','Gym Løvèr','pawan.modi1@gmail.com'),
('Noah','https://picsum.photos/107','aae hai toh kaam bi btiye','sunglasses.24@gmail.com'),
('Alfie','https://picsum.photos/108','Single','deepak@24sevenindia.com'),
('Rory','https://picsum.photos/109','phle zara aap muskurae','pawan.modi1@gmail.com'),
('Alexander','https://picsum.photos/110','Respect For
All','umesh.agarwal@24x7safebuy.com');


-- follows Database

INSERT INTO follows(follower_id, followee_id) VALUES (1, 1);
INSERT INTO follows(follower_id, followee_id) VALUES (2, 2);
INSERT INTO follows(follower_id, followee_id) VALUES (3, 9);
INSERT INTO follows(follower_id, followee_id) VALUES (4, 4);
INSERT INTO follows(follower_id, followee_id) VALUES (5, 19);
INSERT INTO follows(follower_id, followee_id) VALUES (6, 16);
INSERT INTO follows(follower_id, followee_id) VALUES (7, 12);
INSERT INTO follows(follower_id, followee_id) VALUES (8, 8);
INSERT INTO follows(follower_id, followee_id) VALUES (9, 9);
INSERT INTO follows(follower_id, followee_id) VALUES (10 , 10);
INSERT INTO follows(follower_id, followee_id) VALUES (11, 11);
INSERT INTO follows(follower_id, followee_id) VALUES (12, 15);
INSERT INTO follows(follower_id, followee_id) VALUES (13, 9);
INSERT INTO follows(follower_id, followee_id) VALUES (14, 14);
INSERT INTO follows(follower_id, followee_id) VALUES (15, 17);
INSERT INTO follows(follower_id, followee_id) VALUES (16, 16);
INSERT INTO follows(follower_id, followee_id) VALUES (17, 3);
INSERT INTO follows(follower_id, followee_id) VALUES (18, 18);
INSERT INTO follows(follower_id, followee_id) VALUES (19, 1);
INSERT INTO follows(follower_id, followee_id) VALUES (20, 20);
INSERT INTO follows(follower_id, followee_id) VALUES (21, 21);

-- HASHTAGS DATABASE

INSERT INTO HASHTAGS(hashtag_name) VALUES (' #joinbtsarmy');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #kisaansupport');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #fitnessfreak');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #runforunity');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #studentlivesmatter');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #cancellJEEiit');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #REOPEN colleges');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #party');
INSERT INTO
HASHTAGS(hashtag_name) VALUES (' #followme');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #christmas');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #socialmedia');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #family');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #festivesale');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #sunnyday');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #enjoy');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #weekendmasti');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #love');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #instagood');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #photooftheday');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #beautiful');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #fashion');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #tbt');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #happy');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #cute');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #followme');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #like4like');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #follow');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #me');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #picoftheday');
INSERT INTO HASHTAGS(hashtag_name) VALUES (' #selfie');
INSERT INTO HASHTAGS(hashtag_name) VALUES ('#GOGREEN ');


-- photo Database


INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/100', 26, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/101', 27, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/102', 28, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/103', 29, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/104', 30, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/105', 31, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/106', 32, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/107', 33, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/108', 34, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/109', 35, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/110', 36, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/111', 37, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/112', 38, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/113', 39, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/114', 40, 4);
INSERT INTO                              post_id, size) VALUE
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/115', 41, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/116', 42, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/117', 43, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/118', 44, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/119', 45, 1);
photos(photo_url,        ('https://picsum.photos/120', 46, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/121', 47, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/122', 48, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/123', 49, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/124', 50, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/125', 76, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/126', 77, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/127', 78, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/128', 79, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/129', 80, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/130', 81, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/131', 82, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/132', 83, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/133', 84, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/134', 85, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/135', 86, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/136', 87, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/137', 88, 4);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/138', 89, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/139', 90, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/140', 91, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/141', 92, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/142', 93, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/143', 94, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/144', 95, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/145', 96, 1);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/146', 97, 2);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/147', 98, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/148', 99, 3);
INSERT INTO photos(photo_url, post_id, size) VALUE ('https://picsum.photos/149', 100, 2);


-- video Database


INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=1TKJvWbZErY', 1, 1);
```

```
INSERT INTO                          post_id, size) VALUE
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=dcgeBa78WE8', 2, 8);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=vOfgVs6blGU', 3, 3);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=gDGbwhoWRBQ', 4, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=UAj7FB-BFGg', 5, 1);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=RzppsLjuSaI', 6, 3);
                    videos(video_url,
('https://www.youtube.com/watch?v=E1GLuxJ9mkU', 7, 3);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=tjrFQQjTI6c', 8, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=IwNSd7m2HRg', 9, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=4javFbk9Kn8', 10, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=BuX7TQc4a0E', 11, 4);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=4xx0YqaFalo', 12, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=n3LCQiuQn9o', 13, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=x9bmo0aPd0s', 14, 1);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=UkTWeGJewTQ', 15, 1);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=6GwUPaJh2Jg', 16, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=odHuMbTWIvU', 17, 4);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=XxvEchaofrs', 18, 8);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=3ZvSg5aU23E', 19, 6);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=yBJM2RbLefA', 20, 5);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=q6wb-EWR_lM', 21, 6);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=PcCDzONDVsA', 22, 1);
INSERT INTO videos(video_url, post_id, size) VALUE
INSERT INTO                          post_id, size) VALUE
('https://www.youtube.com/watch?v=2ne9HcY53AY', 23, 8);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=OJeynsIPj9I', 24, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=kRGjTgObzX0', 25, 4);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=btWZo8gUv-o', 51, 3);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=7j338SJZjoM', 52, 4);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=fidBadXy1dw', 53, 5);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=1iem1pT2MkQ', 54, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=4N0ew6JMlss', 55, 4); videos(video_url,
('https://www.youtube.com/watch?v=GXCdTSGNcOc', 56, 6);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=JFoJCMXzLLw', 57, 4);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=V-egEzLjnhc', 58, 5);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=6B5UK2GC3gY', 59, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=MVRRN6TABcs', 60, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=1ABkmrZxQkQ', 61, 5);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=cUz49dk86m8', 62, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=36BiplVD_Ng', 63, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=EoGYHDqbabw', 64, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=i1CmuuabIok', 65, 5);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=gurapeu6PBk', 66, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=coHgDPBMLKg', 67, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=racdVMrEghs', 68, 6);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=FKtbZtI0VJ0', 69, 9);
INSERT INTO videos(video_url, post_id, size) VALUE
INSERT INTO                      post_id, size) VALUE
('https://www.youtube.com/watch?v=bWqt7op1VpI', 70, 2);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=V_wXW4J73os', 71, 7);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=sHg9e9a_cYM', 72, 8);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=oaJJvO8Tte8', 73, 1);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=ancKcoTvdYY', 74, 3);
INSERT INTO videos(video_url, post_id, size) VALUE
('https://www.youtube.com/watch?v=n6kEYFPvtBY', 75, 8);


-- post_database


insert into post (post_id,photo_id,video_id,user_id,caption,location) values
(1,15,3,1,'HEY!!','VHA JHA KOI AATA JAATA NI'),
(2,21,47,2,'Live a good story.','The Red Fort, Delhi.'),
```

```sql
(3,45,6,3,'Escape the ordinary.','The Taj Mahal, Agra.'),
(4,7,37,4,'The best is yet to come.','Pangong Lake, Ladakh.'),
(5,4,22,5,'These are days we live for.','Valley of Flowers, Nainital.'),
(6,37,18,6,'Life happens, coffee helps.','Jaisalmer Fort, Jaisalmer.'),
(7,11,12,7,'Short sassy cute & classy.','Ruins of Hampi, Karnataka.'),
(8,50,30,8,'The future is bright.','Ghats at Varanasi, Uttar Pradesh.'),
(9,15,31,9,'Namastay in bed.','Backwaters, Kerala.'),
(10,10,44,10,'I have more issues than vogue.','abhayapuri'),
(11,8,32,11,'Life is short. Smile while you still have teeth.','achabbal'),
(12,9,25,12,'Ah, a perfectly captured selfie!','achalpur'),
(13,4,13,13,'Let’s just be who we are.','achhnera'),
(14,36,12,14,'One bad chapter doesn’t me','adari'),
(15,45,40,15,'Cinderella never asked for a prince.','adalaj'),
(16,39,17,16,'A selfie is worth a thousand words.','adilabad'),
(17,7,31,17,'Born to stand out with selfies.','adityana'),
(18,17,27,18,'I’m sorry I exist, here, a selfie.','pereyaapatna'),
(19,49,48,19,'….','adoni'),
(20,44,30,20,'dfgfsggf','adoor'),
(21,18,8,21,'4545','adyar');


-- post_tag Database

INSERT INTO post_tags(post_id, hashtag_id) VALUE (1, 13);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (2, 27);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (3, 20);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (4, 22);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (5, 22);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (6, 3);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (7, 14);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (8, 11);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (9, 1);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (10, 24);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (11, 7);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (12, 11);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (13, 8);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (14, 20);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (15, 28);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (16, 20);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (17, 5);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (18, 24);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (19, 4);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (20, 22);
INSERT INTO post_tags(post_id, hashtag_id) VALUE (21, 22);
INSERT INTO

-- post_likes

INSERT INTO POST_LIKES(user_id,post_id) VALUES (1,1); POST_LIKES(user_id,post_id)
VALUES (2,2); INSERT INTO POST_LIKES(user_id,post_id) VALUES (3,3);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (4,4);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (5,5);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (6,6);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (7,7);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (8,8);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (9,9);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (10,10);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (11,11);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (12,12);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (13,13);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (14,14);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (15,15);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (16,16);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (17,17);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (18,18);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (19,19);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (20,20);
INSERT INTO POST_LIKES(user_id,post_id) VALUES (21,21);

-- bookmarks

INSERT INTO bookmarks(post_id, user_id) VALUE (1, 1);
INSERT INTO bookmarks(post_id, user_id) VALUE (2, 2);
INSERT INTO bookmarks(post_id, user_id) VALUE (3, 3);
INSERT INTO bookmarks(post_id, user_id) VALUE (4, 4);
INSERT INTO bookmarks(post_id, user_id) VALUE (5, 5);
INSERT INTO bookmarks(post_id, user_id) VALUE (6, 6);
INSERT INTO bookmarks(post_id, user_id) VALUE (7, 7);
INSERT INTO bookmarks(post_id, user_id) VALUE (8, 8);
INSERT INTO bookmarks(post_id, user_id) VALUE (9, 9);
INSERT INTO bookmarks(post_id, user_id) VALUE (10, 10);
INSERT INTO bookmarks(post_id, user_id) VALUE (11, 11);
INSERT INTO bookmarks(post_id, user_id) VALUE (12, 12);
INSERT INTO bookmarks(post_id, user_id) VALUE (13, 13);
INSERT INTO bookmarks(post_id, user_id) VALUE (14, 14);
INSERT INTO bookmarks(post_id, user_id) VALUE (15, 15);
INSERT INTO bookmarks(post_id, user_id) VALUE (16, 16);
INSERT INTO bookmarks(post_id, user_id) VALUE (18, 17);
INSERT INTO bookmarks(post_id, user_id) VALUE (19, 19);
INSERT INTO bookmarks(post_id, user_id) VALUE (20, 20);
INSERT INTO


INSERT INTO bookmarks(post_id, user_id) VALUE (21, 21);


-- comment Database
                    COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',1,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',2,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',3,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',4,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',5,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',6,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',7,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',8,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',9,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',10,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',11,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',12,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',13,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',14,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',15,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',16,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',17,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',18,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',19,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',20,1);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great man',21,1);


INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('looking greate bhai',2,2);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('nice place keep enjoying',3,3);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('great bhai meetu soon ',4,4);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('aag lga di bhai',5,5);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('briallant keep working',6,6);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('soon will join you all',7,7);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('nice man !! loved it',8,8);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('bawnadar aayega abb ',9,9);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('boht tezz ho rhe ho ',10,10);
INSERT INTO


INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('bade acche ho
beta',11,11);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('rakh neeche rakh teri
toh',12,12);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('kaise ho bro',13,13);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('aag lga di h bss fire
extinguisher bulana pdega',14,14);
            COMMENTS(comment_text ,post_id,user_id) VALUES ('ek kahani h jo sabko
sunnani h jakne wako ki toh rooh bhi jaalani h',15,15);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('mast h bhai , mill
tabb btata hu kon mast h ',16,16);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('sahi lgg rha h bss
',17,17);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('I think this is the best
I’ve seen till now.',18,18);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES (' Not enough for me,
you are everything.',19,19);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES (' Just when I couldn’t
love you more. You posted this pic and my jaw dropped to the floor.',20,20);
INSERT INTO COMMENTS(comment_text ,post_id,user_id) VALUES ('You are a symbol of
beauty.',21,21);

-- comment likes database

INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(1 , 1);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(2 , 2);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(3 , 3);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(4 , 4);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(5 , 5);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(6 , 6);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(7 , 7);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(8 , 8);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(9 , 9);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(10 , 10);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(11 , 11);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(12 , 12);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(13 , 13);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(14 , 14);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(15 , 15);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(16 , 16);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(17 , 17);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(18 , 18);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(19 , 19);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(20 , 20);
INSERT INTO COMMENT_LIKES(user_id,comment_id) VALUES(21 , 21);
INSERT INTO

delete from comment_likes where user_id = 16;
-- hashtag_follow

insert into hashtag_follow (user_id,hashtag_id) values
(1,1),
(2,2),
(3,3),
(4,4),
(5,5),
(6,6),
(7,7),
(8,8),
(9,9),
(10,10),
(11,11),
(12,12),
(13,13),
(14,14),
(15,15),
(16,16),
(17,17),
(18,18),
(19,19),
(20,20),
(21,21);


-- LOGIN TABLE DATABASE

INSERT INTO LOGIN(user_id , ip) VALUES (1,'186.83.147.14');
INSERT INTO LOGIN(user_id , ip) VALUES (2,'95.43.246.66');
INSERT INTO LOGIN(user_id , ip) VALUES (3,'105.238.37.204');
INSERT INTO LOGIN(user_id , ip) VALUES (4,'13.120.97.136');
INSERT INTO LOGIN(user_id , ip) VALUES (5,'0.83.214.172');
INSERT INTO LOGIN(user_id , ip) VALUES (6,'20.182.93.222');
INSERT INTO LOGIN(user_id , ip) VALUES (7,'200.237.53.32');
INSERT INTO LOGIN(user_id , ip) VALUES (8,'41.81.231.81');
INSERT INTO LOGIN(user_id , ip) VALUES (9,'24.223.2.33');
INSERT INTO LOGIN(user_id , ip) VALUES (10,'8.168.37.68');
INSERT INTO LOGIN(user_id , ip) VALUES (11,'129.91.145.75');
INSERT INTO LOGIN(user_id , ip) VALUES (12,'8.65.175.204');
INSERT INTO LOGIN(user_id , ip) VALUES (13,'242.220.82.190');
INSERT INTO LOGIN(user_id , ip) VALUES (14,'107.137.170.154');
INSERT INTO LOGIN(user_id , ip) VALUES (15,'206.40.219.225');
INSERT INTO LOGIN(user_id , ip) VALUES (16,'136.186.80.29');
INSERT INTO LOGIN(user_id , ip) VALUES (17,'234.153.100.73');
INSERT INTO LOGIN(user_id , ip) VALUES (18,'137.168.133.16');
INSERT INTO LOGIN(user_id , ip) VALUES (19,'248.119.108.48');
INSERT INTO LOGIN(user_id , ip) VALUES (20,'92.178.211.66');
INSERT INTO LOGIN(user_id , ip) VALUES (21,'25.177.94.84');



select * from comments;
```

```
delete from follows where follower_id = 1;
delete from comments where user_id = 20;
```

Data manipulation queries ->

```
-- 1. Location of User
SELECT * FROM post
WHERE location IN ('agra' ,'adyar','adari');


-- 2. Most Followed Hashtag
SELECT hashtag_name AS 'Hashtags', COUNT(hashtag_follow.hashtag_id) AS 'Total
Follows'
FROM hashtag_follow, hashtags
WHERE hashtags.hashtag_id = hashtag_follow.hashtag_id
GROUP BY hashtag_follow.hashtag_id
ORDER BY COUNT(hashtag_follow.hashtag_id) DESC LIMIT 5;

-- 3. Most Used Hashtags
SELECT
        hashtag_name AS 'Trending Hashtags',
   COUNT(post_tags.hashtag_id) AS 'Times Used'
FROM hashtags,post_tags
WHERE hashtags.hashtag_id = post_tags.hashtag_id
GROUP BY post_tags.hashtag_id
ORDER BY COUNT(post_tags.hashtag_id) DESC LIMIT 10;


-- 4. Most Inactive User
SELECT user_id, username AS 'Most Inactive User'
FROM users
WHERE user_id NOT IN (SELECT user_id FROM post);



-- 6. Average post per user
SELECT ROUND((COUNT(post_id) / COUNT(DISTINCT user_id) ),2) AS 'Average Post per
User'
FROM post;

-- 7. no. of login by per user
SELECT user_id, email, username, login.login_id AS login_number
FROM users
NATURAL JOIN login;


-- 8. User who liked every single post (CHECK FOR BOT)
```

```
SELECT username, Count(*) AS num_likes
FROM users
INNER JOIN post_likes ON users.user_id = post_likes.user_id
GROUP BY post_likes.user_id
HAVING num_likes = (SELECT Count(*) FROM   post);

-- 9. User Never Comment
SELECT user_id, username AS 'User Never Comment'
FROM users
WHERE user_id NOT IN (SELECT user_id FROM comments);

-- 10. User who commented on every post (CHECK FOR BOT)
SELECT username, Count(*) AS num_comment
FROM users
INNER JOIN comments ON users.user_id = comments.user_id
GROUP BY comments.user_id
HAVING num_comment = (SELECT Count(*) FROM comments);


-- 11. User Not Followed by anyone
SELECT user_id, username AS 'User Not Followed by anyone'
FROM users
WHERE user_id NOT IN (SELECT followee_id FROM follows);

-- 12. User Not Following Anyone
SELECT user_id, username AS 'User Not Following Anyone'
FROM users
WHERE user_id NOT IN (SELECT follower_id FROM follows);

-- 13. Posted ATLEAST ONCE
SELECT user_id, COUNT(user_id) AS post_count FROM post
GROUP BY user_id
HAVING post_count > 0
ORDER BY COUNT(user_id) DESC;


-- 14. Followers > 10
CREATE OR REPLACE VIEW GRREATER_10 AS
SELECT followee_id, COUNT(follower_id) AS follower_count FROM follows
GROUP BY followee_id
HAVING follower_count > 1
ORDER BY COUNT(follower_id) DESC;

#DROP VIEW GRREATER_10;
```

```sql
SELECT * FROM GRREATER_10;

-- 15. Any specific word in comment
SELECT * FROM comments
WHERE comment_text REGEXP'love';


-- 16. Longest captions in post
SELECT user_id, caption, LENGTH(post.caption) AS caption_length FROM post
ORDER BY caption_length DESC LIMIT 5;

-- 17. user engagement analysis
SELECT u.user_id, u.username,
       COUNT(DISTINCT p.post_id) AS posts_created,
       COUNT(DISTINCT c.comment_id) AS comments_made
FROM users u
LEFT JOIN post p ON u.user_id = p.user_id
LEFT JOIN comments c ON u.user_id = c.user_id
LEFT JOIN post_likes l ON u.user_id = l.user_id
GROUP BY u.user_id, u.username
ORDER BY (posts_created + comments_made ) DESC;


-- 18. content analysis based on location
SELECT p.location,
       COUNT(*) AS total_posts,
       AVG(LENGTH(p.caption)) AS avg_caption_length,
       COUNT(DISTINCT ht.hashtag_id) AS total_hashtags_used
FROM post p
LEFT JOIN post_tags pt ON p.post_id = pt.post_id
LEFT JOIN hashtags ht ON pt.hashtag_id = ht.hashtag_id
GROUP BY p.location
ORDER BY total_posts DESC;

-- 19. identify influential users
WITH follower_counts AS (
  SELECT follower_id, COUNT(*) AS follower_count
  FROM follows
  GROUP BY follower_id
),
user_engagement AS (
  SELECT u.user_id,
         COUNT(DISTINCT p.post_id) AS posts_created,
         COUNT(DISTINCT c.comment_id) AS comments_made
  FROM users u
  LEFT JOIN post p ON u.user_id = p.user_id
  LEFT JOIN comments c ON u.user_id = c.user_id
  LEFT JOIN post_likes l ON u.user_id = l.user_id
  GROUP BY u.user_id
)
SELECT ue.user_id, fc.follower_count, #users.useranme,
      (ue.posts_created + ue.comments_made ) AS total_engagement
FROM user_engagement ue
INNER JOIN follower_counts fc ON ue.user_id = fc.follower_id
ORDER BY total_engagement DESC, fc.follower_count DESC
LIMIT 10;
-- 20. Analyze popular post by time
SELECT DATE(p.created_at) AS post_date,
      COUNT(*) AS posts_created,
      AVG(LENGTH(p.caption)) AS avg_caption_length
FROM post p
LEFT JOIN post_likes l ON p.post_id = l.post_id
GROUP BY DATE(p.created_at)
ORDER BY post_date DESC
LIMIT 7;

-- 21. Identify trending topics based on commments
WITH comment_words AS (
  SELECT comment_id,
        SUBSTRING_INDEX(comment_text, ' ', 3) AS first_three_words
  FROM comments
)
SELECT cw.first_three_words, COUNT(*) AS mentions
FROM comment_words cw
GROUP BY cw.first_three_words
ORDER BY mentions DESC
LIMIT 10;

-- 22.
SELECT u.username, COUNT(f.follower_id) AS followers,
  SUM(p.likes + p.comments + p.shares) / COUNT(p.post_id) AS avg_engagement
FROM users u
LEFT JOIN followers f ON u.user_id = f.follower_id
LEFT JOIN post p ON u.user_id = p.user_id
GROUP BY u.username
ORDER BY (followers * avg_engagement) DESC;

-- 23.Analyze the most frequent locations used in posts
```

```
SELECT location, COUNT(*) AS post_count
FROM post
WHERE location IS NOT NULL
GROUP BY location
ORDER BY post_count DESC
LIMIT 10;

-- 24. User Engagement by Time
SELECT DATE(created_at) AS post_date, COUNT(*) AS post_count
FROM post
GROUP BY DATE(created_at)
ORDER BY post_date;
```

Self Learning

The project emphasized self learning, particularly in mastering SQL and understanding the complexities of database design. Addressing challenges like query optimization and data security required research and trial and error. This led us to develop a deeper understanding of how databases can be scaled up and secured.

Challenges Faced

A social media platform stores diverse data (users, posts, comments, likes, etc.). We needed to design tables with appropriate relationships (foreign keys) to avoid data redundancy while ensuring efficient querying. Doing so was one of the challenges we faced. Also, needed to keep in mind about keeping our database normalized Thorough testing of database design and queries was crucial to ensure functionality and catch errors. Develop a testing strategy that includes various scenarios took a large amount of time for us.

Conclusion

We successfully created a database management system for social media from initial design to implementation. We outlined our project’s workflow, including the schema and technologies used as Draw.io and MySQL Workbench. We delved into complex queries and understood how to face challenges along the way. We applied our theoretical knowledge to make a practical application.