DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY , WARANGAL

DBMS Project

INSTAGRAM DATABASE MANAGEMENT SYSTEM

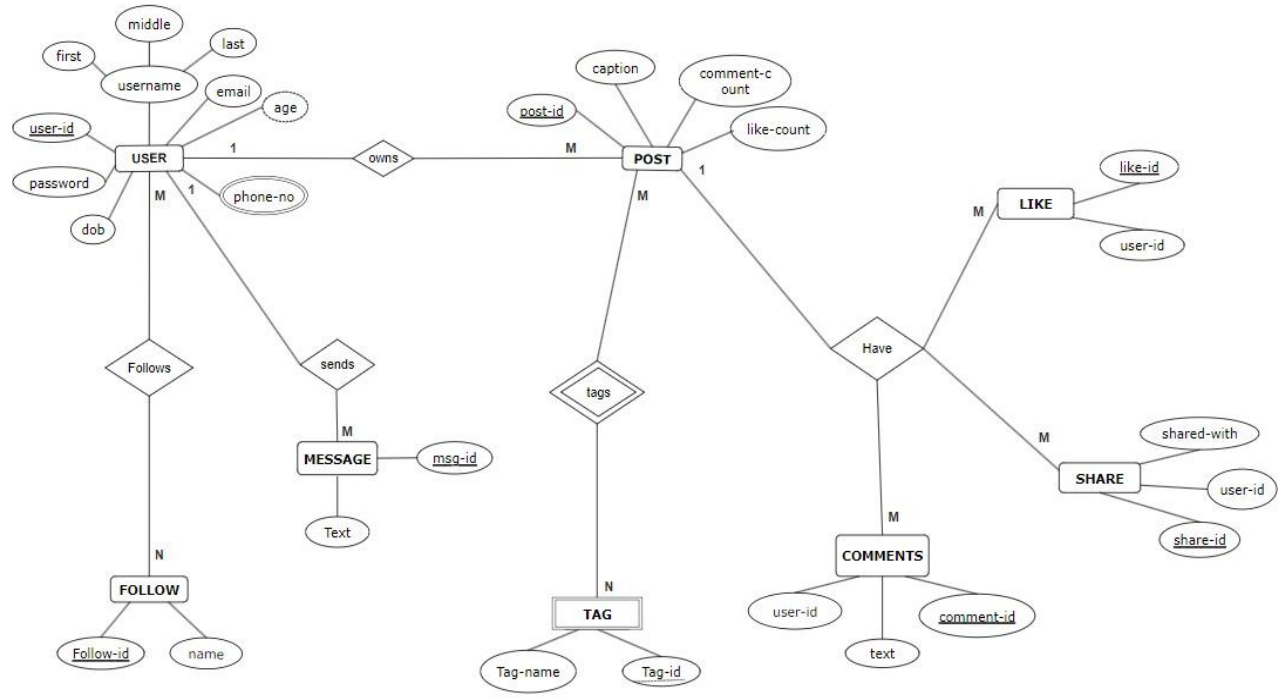
PREPARED BY -

AMAN RAJ : 22CSB0C02 (sec-A)
MANISH MEENA : 22CSB0C45(sec-B)

Problem Statement:

The purpose of this project is to develop a database management system for Instagram to store, manage, and utilize data related to its creators and viewers. This system should be able to store the information of the likes ,comments and share count of a post for a creator and also the number of post for a creator. FurtherMore The database will Show the relevant video of the viewers according to their likes and comment .This database will store the message information between two users with privacy.
The system should be secure and maintain minimal disruption for existing system.

ER DIAGRAM:



ER DIAGRAM ASSUMPTIONS:

1. The relationship between “user” and “Follow” is considered as a many to many relationship because one user can follows multiple user or one follow can
   Follows multiple users.

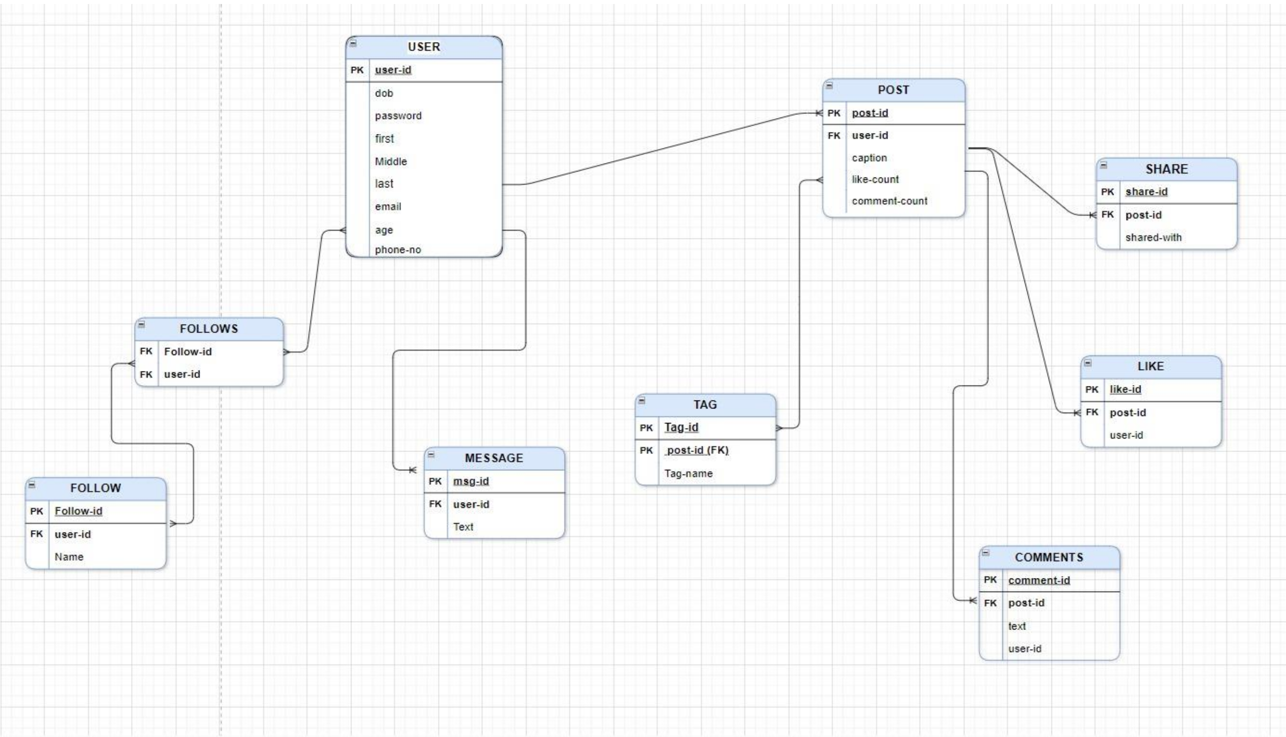
2. The relationship between “User” and “Post” is one to many relationship because one user can own multiple post but one post will have only one user.

3. The relationship between “post” and “like”,”share” and “comment” is considered as one to many relationship Because one post may have multiple likes,comments and share but a particular like,comment or share will belong to a single post.

4. The relationship between “Post” and “Tag” is considered as many to many as a single post can have multiple tag but it can also be possible that multiple post have same tag.

5. User and Message is considered to be one to many relationship as it is assumed that one user can send messages to multiple user but one message belongs to single user.

<u>RELATIONAL SCHEMA</u>:



NORMALISATION:

1. USER

Primary key=user_id

Which means that closure of the user_id will derive all the relation

ALL attributes are dependent on user_id so table is in 2NF(no partial dependency)

All attributes are dependent on user_id so table is in 3NF (no transitive dependency)

AS all relations are from user_id and it is a candidate key so it is in BCNF.

2.POST

Primary key=post_id

Which means that closure of the post_id will derive all the relation

ALL attributes are dependent on post_id so table is in 2NF(no partial dependency)

All attributes are dependent on post_id so table is in 3NF (no transitive dependency)

AS all relations are from post_id and it is a candidate key so it is in BCNF.

3.FOLLOW

Primary key=follow_id

Which means that closure of the follow_id will derive all the relation

ALL attributes are dependent on follow_id so table is in 2NF(no partial dependency)

All attributes are dependent on follow_id so table is in 3NF (no transitive dependency)

AS all relations are from follow_id and it is a candidate key so it is in BCNF.

4.MESSAGE

Primary key=msg_id
Which means that closure of the msg_id will derive all the relation
ALL attributes are dependent on msg_id so table is in 2NF(no partial dependency)
All attributes are dependent on msg_id so table is in 3NF (no transitive dependency)
AS all relations are from msg_id and it is a candidate key so it is in BCNF.

5.TAG

Primary key=(tag_id,post_id)
Which means that closure of(tag_id,post_id) the will derive all the relation
ALL attributes are dependent on (tag_id,post_id) so table is in 2NF(no partial dependency)
All attributes are dependent on (tag_id,post_id) so table is in 3NF (no transitive dependency)
AS all relations are from (tag_id,post_id) and it is a candidate key so it is in BCNF.

6.COMMENTS

Primary key=comment_id
Which means that closure of the comment_id will derive all the relation
ALL attributes are dependent on comment_id so table is in 2NF(no partial dependency)
All attributes are dependent on comment_id so table is in 3NF (no transitive dependency)

AS all relations are from comment_id and it is a candidate key so it is in BCNF.

7.LIKE

Primary key=like_id
Which means that closure of the like_id will derive all the relation
ALL attributes are dependent on like_id so table is in 2NF(no partial dependency)
All attributes are dependent on like_id so table is in 3NF (no transitive dependency)
AS all relations are from like_id and it is a candidate key so it is in BCNF.

8.SHARE

Primary key=share_id
Which means that closure of the share_id will derive all the relation
ALL attributes are dependent on share_id so table is in 2NF(no partial dependency)
All attributes are dependent on share_id so table is in 3NF (no transitive dependency)
AS all relations are from share_id and it is a candidate key so it is in BCNF.

INSERTING DATA INTO TABLES:

```
-- Create Users table

 CREATE TABLE Users (
    user_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(50),
    middle_name VARCHAR2(50),
    last_name VARCHAR2(50),
    age NUMBER,
    email VARCHAR2(100),
    phone_no VARCHAR2(15),
    password VARCHAR2(100),
    dob DATE
);

-- Insert into Users table

INSERT ALL
INTO Users VALUES (1, 'John', 'Doe', '', 25, 'john@example.com',
'1234567890', 'password123', TO_DATE('1999-01-01', 'YYYY-MM-DD'))
INTO Users VALUES (2, 'Alice', '', 'Smith', 30,
'alice@example.com', '9876543210', 'password456',
TO_DATE('1994-05-15', 'YYYY-MM-DD'))
INTO Users VALUES (3, 'Michael', 'A.', 'Johnson', 22,
'michael@example.com', '5551234567', 'password789',
TO_DATE('2002-11-30', 'YYYY-MM-DD'))
```

```
INTO Users VALUES (4, 'Emily', 'Grace', 'Brown', 28,
'emily@example.com', '9998887776', 'passwordabc',
TO_DATE('1996-08-20', 'YYYY-MM-DD'))
INTO Users VALUES (5, 'Sophia', 'L.', 'Anderson', 35,
'sophia@example.com', '1112223334', 'passworddef',
TO_DATE('1989-03-10', 'YYYY-MM-DD'))
INTO Users VALUES (6, 'David', 'E.', 'Martinez', 29,
'david@example.com', '4445556665', 'passwordghi',
TO_DATE('1993-07-05', 'YYYY-MM-DD'))
INTO Users VALUES (7, 'Emma', 'Rose', 'Taylor', 27,
'emma@example.com', '7778889998', 'passwordjkl',
TO_DATE('1997-09-25', 'YYYY-MM-DD'))
INTO Users VALUES (8, 'Daniel', '', 'Wilson', 31,
'daniel@example.com', '2223334447', 'passwordmno',
TO_DATE('1991-12-15', 'YYYY-MM-DD'))
INTO Users VALUES (9, 'Olivia', '', 'White', 26,
'olivia@example.com', '6667778889', 'passwordpqr',
TO_DATE('1995-04-05', 'YYYY-MM-DD'))
INTO Users VALUES (10, 'Matthew', 'J.', 'Lee', 32,
'matthew@example.com', '8889990001', 'passwordstu',
TO_DATE('1990-06-20', 'YYYY-MM-DD'))
SELECT * FROM DUAL;

-- Create Posts table

CREATE TABLE Posts (
   post_id NUMBER PRIMARY KEY,
   user_id NUMBER,
   caption VARCHAR2(500),
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
```

```sql
);

-- Insert into Posts table

INSERT ALL
INTO Posts VALUES (1, 1, 'Beautiful sunset view ')
INTO Posts VALUES (2, 2, 'Enjoying a delicious meal!')
INTO Posts VALUES (3, 3, 'Exploring new places! ')
INTO Posts VALUES (4, 4, 'Fitness goals! ')
INTO Posts VALUES (5, 5, 'Artistic inspiration ')
INTO Posts VALUES (6, 6, 'Live music concert! ')
INTO Posts VALUES (7, 7, 'Fashion show runway! ')
INTO Posts VALUES (8, 8, 'Latest technology trends!')
INTO Posts VALUES (9, 9, 'Reading a great book! ')
INTO Posts VALUES (10, 10, 'Happy moments with friends! ')
SELECT 1 FROM DUAL;

-- Create Like table

CREATE TABLE Like (
    like_id NUMBER PRIMARY KEY,
    user_id NUMBER,
    post_id NUMBER,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (post_id) REFERENCES Posts(post_id)
);

-- Insert into Like table
INSERT ALL
INTO Like VALUES (1, 1, 1)
INTO Like VALUES (2, 2, 2)
INTO Like VALUES (3, 3, 3)
INTO Like VALUES (4, 4, 4)
INTO Like VALUES (5, 5, 5)
INTO Like VALUES (6, 6, 6)
INTO Like VALUES (7, 7, 7)
INTO Like VALUES (8, 8, 8)
INTO Like VALUES (9, 9, 9)
INTO Like VALUES (10, 10, 10)
SELECT 1 FROM DUAL;

-- Create Share table

CREATE TABLE Share (
    post_id NUMBER,
    share_with VARCHAR2(100),
    user_id NUMBER,
    PRIMARY KEY (post_id, share_with),
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (post_id) REFERENCES Posts(post_id)
);

-- Insert into Share table

INSERT ALL
INTO Share VALUES (1, 'friend1@example.com', 1)
INTO Share VALUES (2, 'friend2@example.com', 2)
INTO Share VALUES (3, 'friend3@example.com', 3)
INTO Share VALUES (4, 'friend4@example.com', 4)
INTO Share VALUES (5, 'friend5@example.com', 5)
INTO Share VALUES (6, 'friend6@example.com', 6)
INTO Share VALUES (7, 'friend7@example.com', 7)
INTO Share VALUES (8, 'friend8@example.com', 8)
INTO Share VALUES (9, 'friend9@example.com', 9)
INTO Share VALUES (10, 'friend10@example.com', 10)
SELECT 1 FROM DUAL;

-- Create Comments table

CREATE TABLE Comments (
   comment_id NUMBER PRIMARY KEY,
   text VARCHAR2(500),
   user_id NUMBER,
   post_id NUMBER,
   FOREIGN KEY (user_id) REFERENCES Users(user_id),
   FOREIGN KEY (post_id) REFERENCES Posts(post_id)
);

-- Insert into Comments table

INSERT ALL
INTO Comments VALUES (1, 'Great photo!', 1, 1)
INTO Comments VALUES (2, 'Yum! Looks delicious!', 2, 2)
INTO Comments VALUES (3, 'Amazing view!', 3, 3)
INTO Comments VALUES (4, 'Keep it up!', 4, 4)
INTO Comments VALUES (5, 'Beautiful work!', 5, 5)
INTO Comments VALUES (6, 'Awesome performance!', 6, 6)
INTO Comments VALUES (7, 'Stunning outfits!', 7, 7)
INTO Comments VALUES (8, 'Impressive technology!', 8, 8)
INTO Comments VALUES (9, 'Love that book!', 9, 9)
INTO Comments VALUES (10, 'Fun times!', 10, 10)
SELECT 1 FROM DUAL;

-- Create Tag table

CREATE TABLE Tag (
    tag_id NUMBER ,
    tag_name VARCHAR2(50),
   post_id NUMBER ,
   FOREIGN KEY (post_id) REFERENCES Users(post_id),
   PRIMARY KEY(post_id,tag_id) REFERENCES Users(post_id)

    );

-- Insert into Tag table

INSERT ALL
INTO Tag VALUES (1, 'travel',1)
INTO Tag VALUES (2, 'food',2)
INTO Tag VALUES (3, 'nature',3)
INTO Tag VALUES (4, 'fitness',4)
INTO Tag VALUES (5, 'art',5)
INTO Tag VALUES (6, 'music',6)
INTO Tag VALUES (7, 'fashion',7)
INTO Tag VALUES (8, 'technology',8)
INTO Tag VALUES (9, 'books',9)
INTO Tag VALUES (10, 'friends',9)
SELECT 1 FROM DUAL;
```

```
-- Create Message table

CREATE TABLE Message (
   msg_id NUMBER PRIMARY KEY,
   user_id NUMBER,
   text VARCHAR2(1000),
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Insert into Message table

INSERT ALL
INTO Message VALUES (1, 1, 'Hey, how are you?')
INTO Message VALUES (2, 2, 'Let''s catch up sometime!')
INTO Message VALUES (3, 3, 'I have something to tell you.')
INTO Message VALUES (4, 4, 'Remember that trip we planned?')
INTO Message VALUES (5, 5, 'Just wanted to say hi!')
INTO Message VALUES (6, 6, 'Are you going to the concert?')
INTO Message VALUES (7, 7, 'Check out this new fashion trend!')
INTO Message VALUES (8, 8, 'Have you seen the latest tech
news?')
INTO Message VALUES (9, 9, 'I recommend this book!')
INTO Message VALUES (10, 10, 'Let''s hang out this weekend!')
SELECT 1 FROM DUAL;

-- Create Follow table

CREATE TABLE Follow (
   follow_id NUMBER PRIMARY KEY,
   name VARCHAR2(100)
);

-- Insert into Follow table

INSERT ALL
INTO Follow VALUES (1, 'Friend 1')
INTO Follow VALUES (2, 'Friend 2')
INTO Follow VALUES (3, 'Friend 3')
INTO Follow VALUES (4, 'Friend 4')
INTO Follow VALUES (5, 'Friend 5')
INTO Follow VALUES (6, 'Friend 6')
INTO Follow VALUES (7, 'Friend 7')
INTO Follow VALUES (8, 'Friend 8', 8)
INTO Follow VALUES (9, 'Friend 9', 9)
INTO Follow VALUES (10, 'Friend 10', 10)
SELECT 1 FROM DUAL;

-- Creating table FOLLOWS

Create table Follows(
  USER_ID NUMBER,
  FOLLOW_ID NUMBER
);

--inserting into FOLLOWS table

INSERT ALL
Into Follows values(1,1),
Into Follows values(2,2),
Into Follows values(1,2),
Into Follows values(2,3),
Into Follows values(4,1),
Into Follows values(5,6),
Into Follows values(1,8),
Into Follows values(7,1),
Into Follows values(1,10)
Select 1 from DUAL;
```