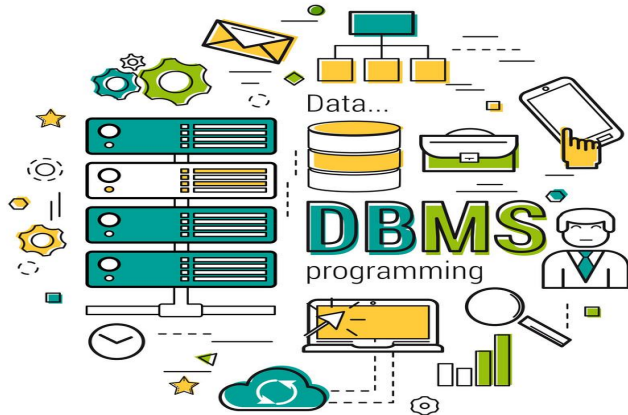


DBMS PROJECT ON:- **INSTAGRAM Clone Database**



Database Management System



INDEX

1. BRIEF INTRODUCTION ABOUT THE PROJECT.
2. E-R DIAGRAM AND ENTITIES AND RELATIONSHIPS TO RELATIONAL TABLE.
3. PERFORMING THE FOLLOWING:-
 - a. Creating a database.
 - b. Creating tables(with and without constraints)
 - c. Inserting/Updating/Deleting.
 - d. Viewing all tables in a database.
4. Performing the alter operation on tables:
Dropping/Truncating/Renaming tables.
5. Performing the operations involving functions :-
 - a. Simple queries
 - b. Simple queries with aggregate functions
 - c. Queries with aggregate functions (group by and having clause).
 - d. Queries involving - Date function ,string functions, math functions.
6. Perform the join and subqueries operations :
 - a. Join queries -Inner join and outer join.
 - b. Sub-queries -WITH-IN clause, with EXISTS clause.

1.INTRODUCTION ABOUT THE PROJECT.

This project is about a social media platform known as instagram clone. The application is a simple way to communicate with friends and family all over the world.

The basic use of the application are to add photos by the users. The photos are seen by the followers who are friend or family members. The photo are liked by the user. Friends and family members follow different user according to their choice and willingness. The Post Creator can add tags to the post in order to reach out more people. The viewers can add comments to post .

All the backend data process use MYSQL as its database.

3. CREATING A DATABASE

CODE :-

```
mysql> CREATE DATABASE ig_clone;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SHOW DATABASES;
```

Database
information_schema
book_shop
ig_clone
mysql
performance_schema
shirts_db
sys

```
7 rows in set (0.00 sec)
```

```
mysql> USE ig_clone;  
Database changed
```

OUTPUT :-

```
mysql> SELECT DATABASE();
```

DATABASE()
ig_clone

```
1 row in set (0.00 sec)
```

3.(b) CREATING TABLES

CODE :-

```
mysql> CREATE TABLE users (  
->  
->   id INTEGER AUTO_INCREMENT PRIMARY KEY,  
->  
->   username VARCHAR(255) UNIQUE NOT NULL,  
->  
->   created_at TIMESTAMP DEFAULT NOW()  
-> );  
Query OK, 0 rows affected (0.15 sec)  
  
mysql> CREATE TABLE photos (  
->  
->   id INTEGER AUTO_INCREMENT PRIMARY KEY,  
->  
->   image_url VARCHAR(255) NOT NULL,  
->  
->   user_id INTEGER NOT NULL,  
->  
->   created_at TIMESTAMP DEFAULT NOW(),  
->  
->   FOREIGN KEY(user_id) REFERENCES users(id)  
-> );  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> CREATE TABLE comments (  
->  
->   id INTEGER AUTO_INCREMENT PRIMARY KEY,  
->  
->   comment_text VARCHAR(255) NOT NULL,  
->  
->   photo_id INTEGER NOT NULL,  
->  
->   user_id INTEGER NOT NULL,  
->  
->   created_at TIMESTAMP DEFAULT NOW(),  
->  
->   FOREIGN KEY(photo_id) REFERENCES photos(id),  
->  
->   FOREIGN KEY(user_id) REFERENCES users(id)  
-> );  
Query OK, 0 rows affected (0.14 sec)  
  
mysql> CREATE TABLE likes (  
->  
->   user_id INTEGER NOT NULL,  
->  
->   photo_id INTEGER NOT NULL,  
->  
->   created_at TIMESTAMP DEFAULT NOW(),  
->  
->   FOREIGN KEY(user_id) REFERENCES users(id),  
->  
->   FOREIGN KEY(photo_id) REFERENCES photos(id),  
->  
->   PRIMARY KEY(user_id, photo_id)  
-> );  
Query OK, 0 rows affected (0.06 sec)
```

CODE :-

```
mysql> CREATE TABLE follows (  
->  
->   follower_id INTEGER NOT NULL,  
->  
->   followee_id INTEGER NOT NULL,  
->  
->   created_at TIMESTAMP DEFAULT NOW(),  
->  
->   FOREIGN KEY(follower_id) REFERENCES users(id),  
->  
->   FOREIGN KEY(followee_id) REFERENCES users(id),  
->  
->   PRIMARY KEY(follower_id, followee_id)  
-> );  
Query OK, 0 rows affected (0.11 sec)  
  
mysql> CREATE TABLE tags (  
->  
->   id INTEGER AUTO_INCREMENT PRIMARY KEY,  
->  
->   tag_name VARCHAR(255) UNIQUE,  
->  
->   created_at TIMESTAMP DEFAULT NOW()  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> CREATE TABLE photo_tags (  
->  
->   photo_id INTEGER NOT NULL,  
->  
->   tag_id INTEGER NOT NULL,  
->  
->   FOREIGN KEY(photo_id) REFERENCES photos(id),  
->  
->   FOREIGN KEY(tag_id) REFERENCES tags(id),  
->  
->   PRIMARY KEY(photo_id, tag_id)  
-> );  
Query OK, 0 rows affected (0.08 sec)
```

3.(C)INSERTION

CODE :-

```
mysql> INSERT INTO users (username, created_at)
->
-> VALUES
->
-> ('Shivam_g10', '2017-03-10 18:22:10.846'),
->
-> ('Andrew_Purdy85', '2021-04-02 17:11:21.417'),
->
-> ('Hardik_Bhati18', '2020-02-21 11:12:32.574');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO tags(tag_name)
->
-> VALUES
->
-> ('sunset'), ('photography'),
->
-> ('sunrise'),('landscape'),
->
-> ('food'), ('foodie'),
->
-> ('delicious'), ('beauty'),
->
-> ('stunning'), ('dreamy'),
->
-> ('lol'), ('happy'),
->
-> ('fun'), ('style'), ('hair'),
->
-> ('fashion'), ('party'), ('concert'),
->
-> ('drunk'),('beach'), ('smile');
Query OK, 21 rows affected (0.02 sec)
Records: 21  Duplicates: 0  Warnings: 0
```

3.(C)UPDATING

BEFORE Updation :-

```
mysql> select * from photo_tags;
```

photo_id	tag_id
45	1
2	2
2	3
45	3
2	4
65	5

CODE :-

```
mysql> UPDATE photo_tags  
->  
-> SET tag_id = 12  
->  
-> WHERE photo_id = 45 AND tag_id=  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings:
```

AFTER Updation:-

4	11
4	12
24	12
34	12
45	12
64	12
1	13
4	13

3.(C)DELETION

Before deleting :-

	2		20	
	1		21	
	4		21	
	8		21	
	23		21	
+-----+				
33 rows in set (0.00 sec)				

CODE :-

```
mysql> DELETE FROM photo_tags
->
-> WHERE
->
-> photo_id = 23 AND tag_id = 21;
Query OK, 1 row affected (0.01 sec)
```

Output after delting :-

	2		20	
	1		21	
	4		21	
	8		21	
+-----+				
32 rows in set (0.00 sec)				

3.(d)VIEWING ALL TABLES IN A DATABASE

CODE :-

```
mysql> SHOW TABLES;
+-----+
| Tables_in_ig_clone |
+-----+
| comments           |
| follows            |
| likes              |
| photo_tags         |
| photos             |
| tags               |
| users              |
+-----+
7 rows in set (0.00 sec)
```

Output :-

```
mysql> DESC comments;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
comment_text	varchar(255)	NO		NULL	
photo_id	int(11)	NO	MUL	NULL	
user_id	int(11)	NO	MUL	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

```
5 rows in set (0.00 sec)
```

```
mysql> DESC follows;
```

Field	Type	Null	Key	Default	Extra
follower_id	int(11)	NO	PRI	NULL	
followee_id	int(11)	NO	PRI	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

```
3 rows in set (0.00 sec)
```

Output :-

```
mysql> DESC likes;
```

Field	Type	Null	Key	Default	Extra
user_id	int(11)	NO	PRI	NULL	
photo_id	int(11)	NO	PRI	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

3 rows in set (0.00 sec)

```
mysql> DESC photos;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
image_url	varchar(255)	NO		NULL	
user_id	int(11)	NO	MUL	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

4 rows in set (0.00 sec)

```
mysql> DESC tags;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
tag_name	varchar(255)	YES	UNI	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

3 rows in set (0.00 sec)

Output :-

```
mysql> DESC photo_tags;
```

Field	Type	Null	Key	Default	Extra
photo_id	int(11)	NO	PRI	NULL	
tag_id	int(11)	NO	PRI	NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> DESC users;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(255)	NO	UNI	NULL	
created_at	timestamp	NO		CURRENT_TIMESTAMP	

```
3 rows in set (0.00 sec)
```

4. PERFORM OPERATION ON TABLES

Before data :-

```
mysql> select * from comments;
```

id	comment_text	photo_id	user_id	created_at
1	unde at dolore	1	2	2021-03-13 22:34:51
2	quae ea ducimus	1	3	2021-03-13 22:34:51
3	alias a voluptatum	1	5	2021-03-13 22:34:51
4	facere suscipit sunt	1	14	2021-03-13 22:34:51
5	totam eligendi quaerat	1	17	2021-03-13 22:34:51
6	vitae quia aliquam	1	21	2021-03-13 22:34:51
7	exercitationem occaecati neque	1	24	2021-03-13 22:34:51
8	sunt dolore reprehenderit	2	19	2021-03-13 22:34:51
9	omnis omnis nulla	2	21	2021-03-13 22:34:51
10	ullam in facilis	2	24	2021-03-13 22:34:51
11	vitae iusto labore	4	26	2021-03-13 22:34:51
12	atque minima fuga	12	18	2021-03-13 22:34:51
13	aut molestias accusamus	12	21	2021-03-13 22:34:51
14	et odit quidem	27	3	2021-03-13 22:34:51
15	labore et saepe	27	4	2021-03-13 22:34:51
16	et ipsam et	66	11	2021-03-13 22:34:51
17	minus dolore assumenda	66	12	2021-03-13 22:34:51

```
17 rows in set (0.00 sec)
```

4.PERFORM OPERATION ON TABLES

DROPPING :-

CODE :-

```
mysql> DROP TABLE comments;  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> select * from comments;  
ERROR 1146 (42S02): Table 'ig_clone.comments' doesn't exist
```

Truncating

CODE :-

```
mysql> TRUNCATE TABLE comments;  
Query OK, 0 rows affected (0.06 sec)  
  
mysql> select * from comments;  
Empty set (0.00 sec)
```

4.PERFORM OPERATION ON TABLES

RENAMING :-

RENAMING TYPE 1

CODE and OUTPUT :-

```
mysql> RENAME TABLE tags TO hash_tags;  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> SHOW TABLES;
```

Tables_in_ig_clone
follows
hash_tags
likes
photo_tags
photos
users

6 rows in set (0.00 sec)

RENAMING TYPE 2

CODE and OUTPUT :-

```
mysql> ALTER TABLE hash_tags RENAME tags;  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> SHOW TABLES;
```

Tables_in_ig_clone
follows
likes
photo_tags
photos
tags
users

6 rows in set (0.00 sec)

5.SIMPLE QUERIES

1. *QUERY For FINDING OLDEST 5 USERS.*

CODE AND OUTPUT :-

```
mysql> SELECT *  
->  
-> FROM users  
->  
-> ORDER BY created_at  
->  
-> LIMIT 5;
```

id	username	created_at
1	Shivam_g10	2010-03-10 18:22:11
26	Adelle96	2012-10-01 00:37:57
6	Tierra.Trantow	2012-10-03 12:49:21
22	Jayson65	2013-10-14 19:10:53
5	Maxwell201	2014-04-18 02:32:44

5 rows in set (0.00 sec)

5. SIMPLE QUERIES

2. QUERY For FINDING TOTAL NO. USERS.

CODE AND OUTPUT :-

```
mysql> SELECT  
->  
-> COUNT(*) AS No_of_Users  
->  
-> FROM users;  
+-----+  
| No_of_Users |  
+-----+  
|          28 |  
+-----+  
1 row in set (0.00 sec)
```

SIMPLE QUERIES WITH AGGREGATE FUNCTIONS

**QUERY For FINDING 2 MOST POPULAR
REGISTRATION DAYS.**

CODE AND OUTPUT :-

```
mysql> SELECT  
->  
->     DAYNAME(created_at) AS day,  
->  
->     COUNT(*) AS total  
->  
-> FROM users  
->  
-> GROUP BY day  
->  
-> ORDER BY total DESC  
->  
-> LIMIT 2;
```

day	total
Friday	7
Sunday	5

2 rows in set (0.08 sec)

QUERIES WITH AGGREGATE FUNCTIONS

QUERY For FINDING THE 5 MOST POPULAR HashTags

CODE AND OUTPUT :-

```
mysql> SELECT tags.tag_name,  
->  
->      Count(*) AS total  
->  
-> FROM   photo_tags  
->  
->      JOIN tags  
->  
->      ON photo_tags.tag_id = tags.id  
->  
-> GROUP BY tags.id  
->  
-> ORDER BY total DESC  
->  
-> LIMIT 5;
```

```
+-----+-----+  
| tag_name | total |  
+-----+-----+  
| fun      | 5     |  
| happy    | 5     |  
| party    | 3     |  
| smile    | 3     |  
| hair     | 2     |  
+-----+-----+  
5 rows in set (0.00 sec)
```

QUERIES WITH AGGREGATE FUNCTIONS

QUERY FOR FINDING ALL FOLLOWERS OF shivam_g10 (group by and having)

CODE :-

```
mysql> SELECT
->
-> username,
->
-> followee_id
->
-> FROM users
->
-> JOIN follows
->
-> ON follower_id = users.id
->
-> GROUP BY username
->
-> HAVING
->
-> followee_id = (SELECT id
->
->                      FROM users
->
->                      WHERE
->
->                      username = 'Shivam_g10');
```

QUERIES WITH AGGREGATE FUNCTIONS

QUERY FOR FINDING ALL FOLLOWER OF Shivam_g10 (group by and having)

OUTPUT :-

username	followee_id
Adelle96	1
Andrew_Purdy85	1
Aurelie71	1
Cesar93	1
Duane60	1
Elenor88	1
Emilio_Bernier52	1
Eveline95	1
Granville_Kutch	1
Hardik_Bhati18	1
Harrison.Beatty50	1
Jayson65	1
Julien_Schmidt	1
Maxwell201	1
Meggie_Doyle	1
Mike.Auer39	1
Peter.Stehr0	1
Sam52	1

18 rows in set (0.04 sec)

QUERIES INVOLVING DATE FUNCTIONS

QUERY FOR FINDING OUT DATE NAME AND MONTH WHEN USER SIGN UP.

CODE :-

```
mysql> SELECT  
->  
-> username,  
->  
-> DAYNAME(created_at) as day_name,  
->  
-> DAYOFMONTH(created_at) as Day_of_Month  
->  
-> FROM users;
```

QUERIES INVOLVING DATE FUNCTIONS

QUERY FOR FINDING OUT DATE NAME AND MONTH WHEN USER SIGN UP.

OUTPUT :-

username	day_name	Day_of_Month
Shivam_g10	Wednesday	10
Andrew_Purdy85	Friday	2
Hardik_Bhati18	Friday	21
Eveline95	Friday	23
Maxwell201	Friday	18
Tierra.Trantow	Wednesday	3
Malinda_Streich	Saturday	9
Harrison.Beatty50	Friday	2
Granville_Kutch	Sunday	26
Morgan.Kassulke	Sunday	30
Gerard79	Tuesday	23
Mariano_Koch3	Monday	17
Zack_Kemmer93	Sunday	1
Linnea59	Tuesday	7
Duane60	Wednesday	21
Meggie_Doyle	Saturday	4
Peter.Stehr0	Monday	22
Julien_Schmidt	Thursday	2
Aurelie71	Tuesday	31
Cesar93	Tuesday	18
Sam52	Thursday	30
Jayson65	Monday	14
Ressie_Stanton46	Sunday	20
Elenor88	Sunday	8
Florence99	Thursday	6
Adelle96	Monday	1
Mike.Auer39	Friday	1
Emilio_Bernier52	Friday	6

28 rows in set (0.00 sec)

QUERIES INVOLVING STRING FUNCTIONS

QUERY FOR FINDING OUT CHARACTER LENGTH FOR DIFFERENT USERNAME

CODE :-

```
mysql> SELECT  
->  
-> username,  
->  
-> CHAR_LENGTH(username) as length  
->  
-> FROM users  
->  
-> ORDER BY length;
```

QUERIES INVOLVING MATH FUNCTIONS

QUERY FOR FINDING OUT NUMBER OF
PHOTOS UPLOADED BY DIFFERENT
USERS:-

OUTPUT :-

username	No_of_pics
Adelle96	5
Andrew_Purdy85	2
Aurelie71	2
Cesar93	1
Duane60	3
Emilio_Bernier52	2
Eveline95	3
Gerard79	5
Granville_Kutch	2
Hardik_Bhati18	3
Harrison.Beatty50	4
Jayson65	1
Julien_Schmidt	1
Mariano_Koch3	4
Meggie_Doyle	2
Mike.Auer39	1
Morgan.Kassulke	2
Peter.Stehr0	2
Ressie_Stanton46	8
Shivam_g10	5
Tierra.Trantow	5
Zack_Kemmer93	3

22 rows in set (0.00 sec)

Join queries (Outer Joins)

**QUERY For Identifying
Inactive Users (users with no
photos) :-**

CODE AND OUTPUT :-

```
mysql> SELECT username  
->  
-> FROM users  
->  
-> LEFT JOIN photos  
->  
->     ON users.id = photos.user_id  
->  
-> WHERE photos.id IS NULL;
```

username
Elenor88
Florence99
Linnea59
Malinda_Streich
Maxwell201
Sam52

6 rows in set (0.00 sec)

Join queries (Inner Joins)

QUERY For Identifying most popular photo (and user who created it) :-

```
mysql> SELECT
->
->     username,
->     photos.id,
->     photos.image_url,
->     COUNT(*) AS total
->
-> FROM photos
->
-> INNER JOIN likes
->
->     ON likes.photo_id = photos.id
->
-> INNER JOIN users
->
->     ON photos.user_id = users.id
->
-> GROUP BY photos.id
->
-> ORDER BY total DESC
->
-> LIMIT 1;
```

username	id	image_url	total
Shivam_g10	1	http://elijah.biz	9

1 row in set (0.00 sec)

SubQueries with exist clause

Query For Finding tag id of a tag name used for different photos by users.

```
mysql> SELECT tag_name,id
->
-> FROM tags
->
-> WHERE EXISTS
-> (SELECT photo_id
-> FROM photo_tags
-> WHERE
-> photo_tags.tag_id = tags.id );
```

tag_name	id
beach	20
beauty	8
concert	18
dreamy	10
drunk	19
food	5
foodie	6
fun	13
hair	15
happy	12
landscape	4
lol	11
party	17
photography	2
smile	21
style	14
sunrise	3
sunset	1

18 rows in set (0.00 sec)

Subquery with 'in' clause

Query For Finding comment id and comments which contains " et " in between their texts .

```
mysql> SELECT id,  
->  
-> comment_text  
->  
-> FROM comments  
->  
-> WHERE comment_text  
->  
->         IN (SELECT comment_text  
->  
->             FROM comments  
->  
->             WHERE comment_text Like '%et%');
```

id	comment_text
14	et odit quidem
15	labore et saepe
16	et ipsam et

3 rows in set (0.00 sec)

View without Check condition

***Query For Creating a View Table of
Inactive Users (Users who have
never Posted a photo) .***

```
mysql> CREATE VIEW inactive_users AS
->
-> SELECT username , users.id
->
-> FROM users
->
-> LEFT JOIN photos
->
-> ON users.id=photos.user_id
->
-> WHERE photos.id IS NULL;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_ig_clone |
+-----+
| follows             |
| inactive_users      |
| likes               |
| photo_tags          |
| photos              |
| tags                |
| users               |
+-----+
7 rows in set (0.00 sec)
```

View with Check condition

***Query For Creating a View
Table of Users who Signed
up on the very 1st day of
any month .***

```
mysql> CREATE OR REPLACE VIEW 1stDay_SignUps AS
->
-> SELECT
->
-> username,
->
-> DAYNAME(created_at) as day_name,
->
-> DAYOFMONTH(created_at) as Day_of_Month
->
-> FROM users
->
-> WHERE DAYOFMONTH(created_at) = 1
->
-> WITH CHECK OPTION;
Query OK, 0 rows affected (0.01 sec)
```


1stDay_SignUps View Table Created.

```
mysql> show tables;
+-----+
| Tables_in_ig_clone |
+-----+
| 1stDay_SignUps      |
| follows             |
| inactive_users      |
| likes               |
| photo_tags          |
| photos              |
| tags                |
| users               |
+-----+
8 rows in set (0.00 sec)
```

Selecting From View Table

```
mysql> SELECT *  
->  
-> FROM inactive_users;  
  
+-----+-----+  
| username | id |  
+-----+-----+  
| Elenor88 | 24 |  
| Florence99 | 25 |  
| Linnea59 | 14 |  
| Malinda_Streich | 7 |  
| Maxwell201 | 5 |  
| Sam52 | 21 |  
+-----+-----+  
6 rows in set (0.00 sec)
```

```
mysql> SELECT *  
->  
-> FROM 1stDay_SignUps;  
  
+-----+-----+-----+  
| username | day_name | Day_of_Month |  
+-----+-----+-----+  
| Zack_Kemmer93 | Sunday | 1 |  
| Adelle96 | Monday | 1 |  
| Mike.Auer39 | Friday | 1 |  
+-----+-----+-----+  
3 rows in set (0.03 sec)
```

Dropping a View Table

```
mysql> DROP VIEW 1stDay_SignUps;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show tables;
```

Tables_in_ig_clone
follows
inactive_users
likes
photo_tags
photos
tags
users

```
7 rows in set (0.00 sec)
```

