

```
CREATE DATABASE instagram_clone;
```

```
USE instagram_clone;
```

TABLES:

1. Creating users table

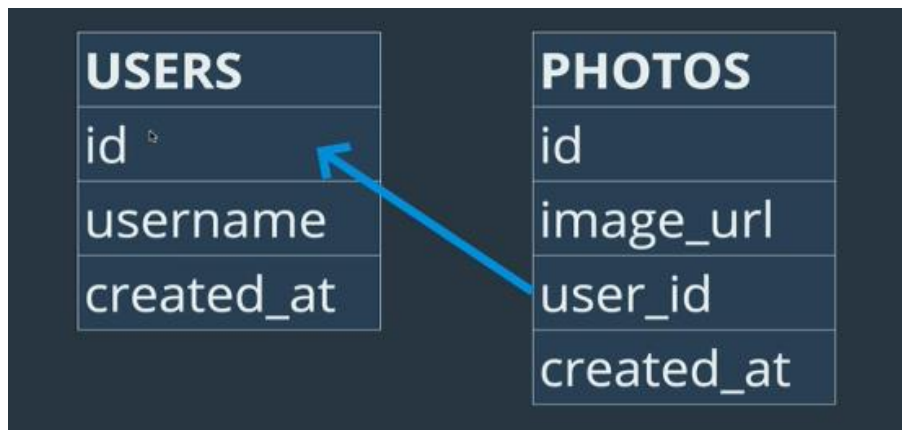
USERS
id
username
created_at

```
-- Creating table for users
CREATE TABLE users
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(255) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);
```

12 • DESC users;

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

2. Creating photos table:



```
-- Creating photos table
• CREATE TABLE photos
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    image_url VARCHAR(255) NOT NULL,
    user_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
```

```
23 • DESC photos;
```

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content: IA
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	image_url	varchar(255)	NO		NULL	
	user_id	int	NO	MUL	NULL	
	created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

3. Creating comments table:



```
-- Creating table for comments
CREATE TABLE comments
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    comment_text VARCHAR(255) NOT NULL,
    user_id INT NOT NULL,
    photo_id INT NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id),
    FOREIGN KEY(photo_id) REFERENCES photos(id)
);
```

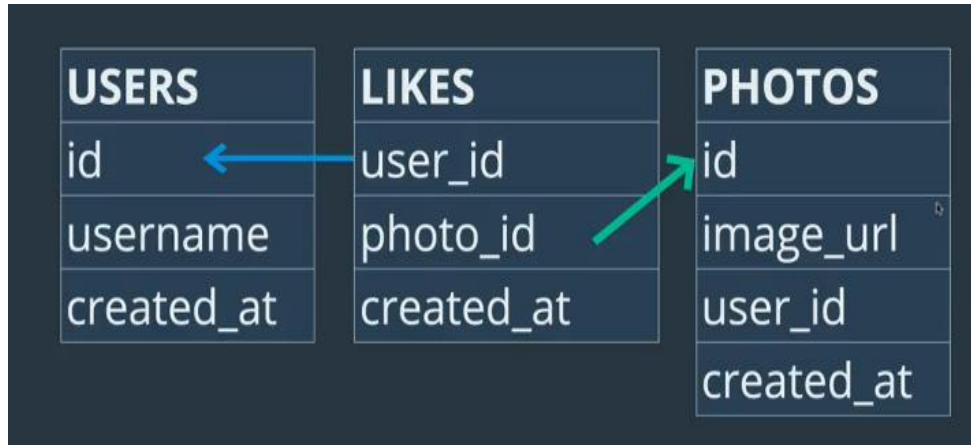
37 • DESC comments;

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	<small>HULL</small>	auto_increment
	comment_text	varchar(255)	NO		<small>HULL</small>	
	user_id	int	NO	MUL	<small>HULL</small>	
	photo_id	int	NO	MUL	<small>HULL</small>	
	created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

4. Creating table for likes:



```
-- Creating table for likes
CREATE TABLE likes
(
    user_id INT NOT NULL,
    photo_id INT 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) -- To make it such as a person can like a photo once.
);
```

50 • DESC likes;

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Field	Type	Null	Key	Default	Extra
▶	user_id	int	NO	PRI	NULL	
	photo_id	int	NO	PRI	NULL	
	created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

5. Creating table for followers



Eg:

Users		
ID	USERNAME	CREATED_AT
1	Tommy	2016-10-11
2	BlueCat	2017-01-10
3	ColtSteele	2017-04-04

Follows		
FOLLOWER_ID	FOLLOWEE_ID	CREATED_AT
3	2	2017-03-03
3	1	2017-04-04

```
-- Creating table for followers
CREATE TABLE follows
(
    follower_id INT NOT NULL,
    followee_id INT 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) -- To Check the following to and back only once
);
```

63 • DESC follows;

Field	Type	Null	Key	Default	Extra
follower_id	int	NO	PRI	NULL	
followee_id	int	NO	PRI	NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

6. Creating table for hashtags

Photos		
id	image_url	caption
1	'/ksjd97123'	'My cat'
2	'/098fsdskj'	'My meal'
3	'/87hghjkd'	'A Selfie'

Photo_Tags		
photo_id	tag_id	
1	1	
1	2	
2	3	
2	6	
3	1	
3	4	
3	5	

Tags	
id	tag_name
1	'#cute'
2	'#pets'
3	'#microwave'
4	'#ego'
5	'#smile'
6	'#gross'

```
-- Creating tables for hashtags
CREATE TABLE tags
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    tag_name VARCHAR(255) UNIQUE,
    created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE photo_tags
(
    photo_id INT NOT NULL,
    tag_id INT NOT NULL,
    FOREIGN KEY(photo_id) REFERENCES photos(id),
    FOREIGN KEY(tag_id) REFERENCES tags(id),
    PRIMARY KEY (photo_id,tag_id)
);
```

```
82 • DESC tags;
```

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

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

```
84 • DESC photo_tags;
```

<

Result Grid

Filter Rows:

Export:

	Field	Type	Null	Key	Default	Extra
▶	photo_id	int	NO	PRI	NULL	
	tag_id	int	NO	PRI	NULL	

Done all the insertion with the data's from Insertion Data txt file.

Task 1

We want to reward our users who have been around the longest. Find the 5 oldest users.

```
102      -- TASK 1
103      -- We want to reward our users who have been around the longest. Find the 5 oldest users.
104      • SELECT * FROM users ORDER BY created_at LIMIT 5;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit: Export/Import: Wrap Cell Content: Fetch			
	id	username	created_at
▶	80	Darby_Herzog	2016-05-06 00:14:21
	67	Emilio_Bernier52	2016-05-06 13:04:30
	63	Elenor88	2016-05-08 01:30:41
	95	Nicole71	2016-05-09 17:30:22
	38	Jordyn.Jacobson2	2016-05-14 07:56:26

Task 2

What day of the week do most users register on? We need to figure out when to schedule an ad campaign.

```
107      -- Task 2
108      -- What day of the week do most users register on? We need to figure out when to schedule an ad campaign.
109      • SELECT
110          DAYNAME(created_at) AS Day,
111          COUNT(DAYNAME(created_at)) AS Total
112      FROM users
113      GROUP BY Day
114      ORDER BY Total DESC
115      LIMIT 2;
```

Result Grid		
Filter Rows: <input type="text"/>		
Export: Wrap Cell Content: Fetch rows:		
	Day	Total
▶	Thursday	16
	Sunday	16

TASK 3

We want to target our inactive users with an email campaign. Find the users who have never posted a photo

```
117 -- TASK 3
118 -- We want to target our inactive users with an email campaign. Find the users who have never posted a photo
119 • SELECT users.id,username
120 FROM users
121 LEFT JOIN photos
122 ON photos.user_id = users.id
123 WHERE image_url IS NULL
124 ORDER BY users.id;
```

id	username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz57
91	Bethany20

Total 26 users are inactive or never posted a photo.

TASK 4

We are running a new contest to see who can get the most likes on a single photo. Who won ???

```
126 -- Task 4
127 -- We are running a new contest to see who can get the most likes on a single photo. Who won ???
128 • SELECT
129     username AS Name,
130     photos.image_url AS 'Image',
131     COUNT(*) AS 'Total likes'
132 FROM photos
133 INNER JOIN likes
134     ON likes.photo_id = photos.id
135 INNER JOIN users
136     ON photos.user_id = users.id
137 GROUP BY photos.id
138 ORDER BY COUNT(*) DESC
139 LIMIT 1;
140
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	Name	Image	Total likes
▶	Zack_Kemmer93	https://jarret.name	48

TASK 5

The investors want to know how many times the average user posts.

```
142 -- Task 5
143 -- The investors want to know how many times the average user posts.
144 • SELECT ((SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users)) AS 'Average user post'; -- Total no:of post / Total no:od users
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Average user post
▶	2.5700

TASK 6

A brand wants to know which hashtags to use in a post. What are the top 5 most commonly used hashtags?

```
147 -- TASK 6
148 -- A brand wants to know which hashtags to use in a post. What are the top 5 most commonly used hashtags?
149 • SELECT
150     tag_name,
151     COUNT(tag_id) AS total_count
152 FROM tags
153 JOIN photo_tags
154     ON photo_tags.tag_id = tags.id
155 GROUP BY tag_name
156 ORDER BY total_count DESC
157 LIMIT 5;
```

tag_name	total_count
smile	59
beach	42
party	39
fun	38
concert	24

TASK 7

We have a small problem with bots on our site... Find users who have liked every single photo on the site?

```
160 -- TASK 7
161 -- We have a small problem with bots on our site... Find users who have liked every single photo on the site?
162 • SELECT
163     username,
164     COUNT(*) AS num_likes
165 FROM users
166 JOIN likes
167     ON users.id = likes.user_id
168 GROUP BY likes.user_id
169 HAVING num_likes = (SELECT COUNT(*) FROM photos);
```

username	num_likes
Aniya_Hackett	257
Jadyn81	257
Rocio33	257
Maxwell_Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike_Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bethany20	257