

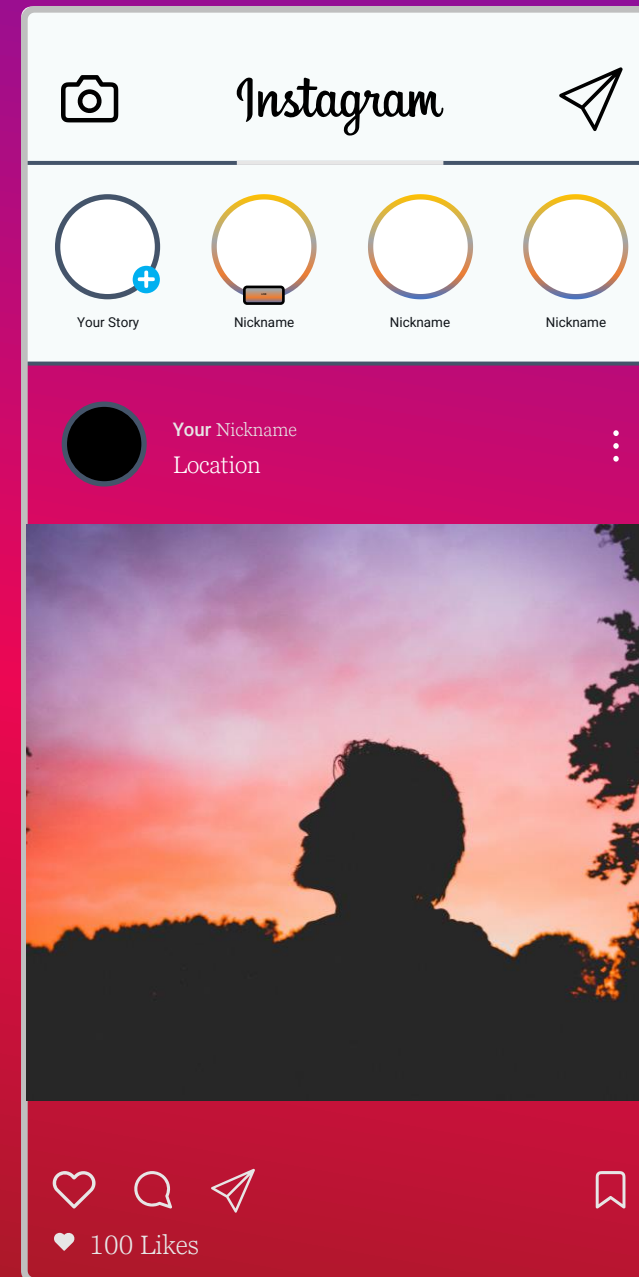


Instagram

Instagram Database Clone

My name is Avantika Rajpurohit .

In this project I used my creative thinking and made this Instagram Database Clone in which I have also solved Instagram related Queries using SQL.



Interesting

Instagram Schema

USER TABLE

```
CREATE TABLE USER(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(255) NOT NULL,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

PHOTOS TABLE

```
CREATE TABLE PHOTOS(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  image_url VARCHAR(255) NOT NULL ,  
  user_id INT NOT NULL,  
  created_at TIMESTAMP DEFAULT NOW(),  
  FOREIGN KEY(user_id) REFERENCE USER(id)  
);
```

COMMENT TABLE

```
CREATE TABLE COMMENTS(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  comment_text VARCHAR(255) NOT NULL,  
  photo_id INT NOT NULL,  
  user_id INT NOT NULL  
  created_at TIMESTAMP DEFAULT NOW(),  
  FOREIGN KEY(user_id) REFERENCE USER(id),  
  FOREIGN KEY(photo_id) REFERENCE PHOTOS(id)  
);
```



Interesting

Instagram Schema



LIKES TABLE

```
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)  
);
```

FOLLOWS TABLE

```
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)  
);
```

TAGS TABLE

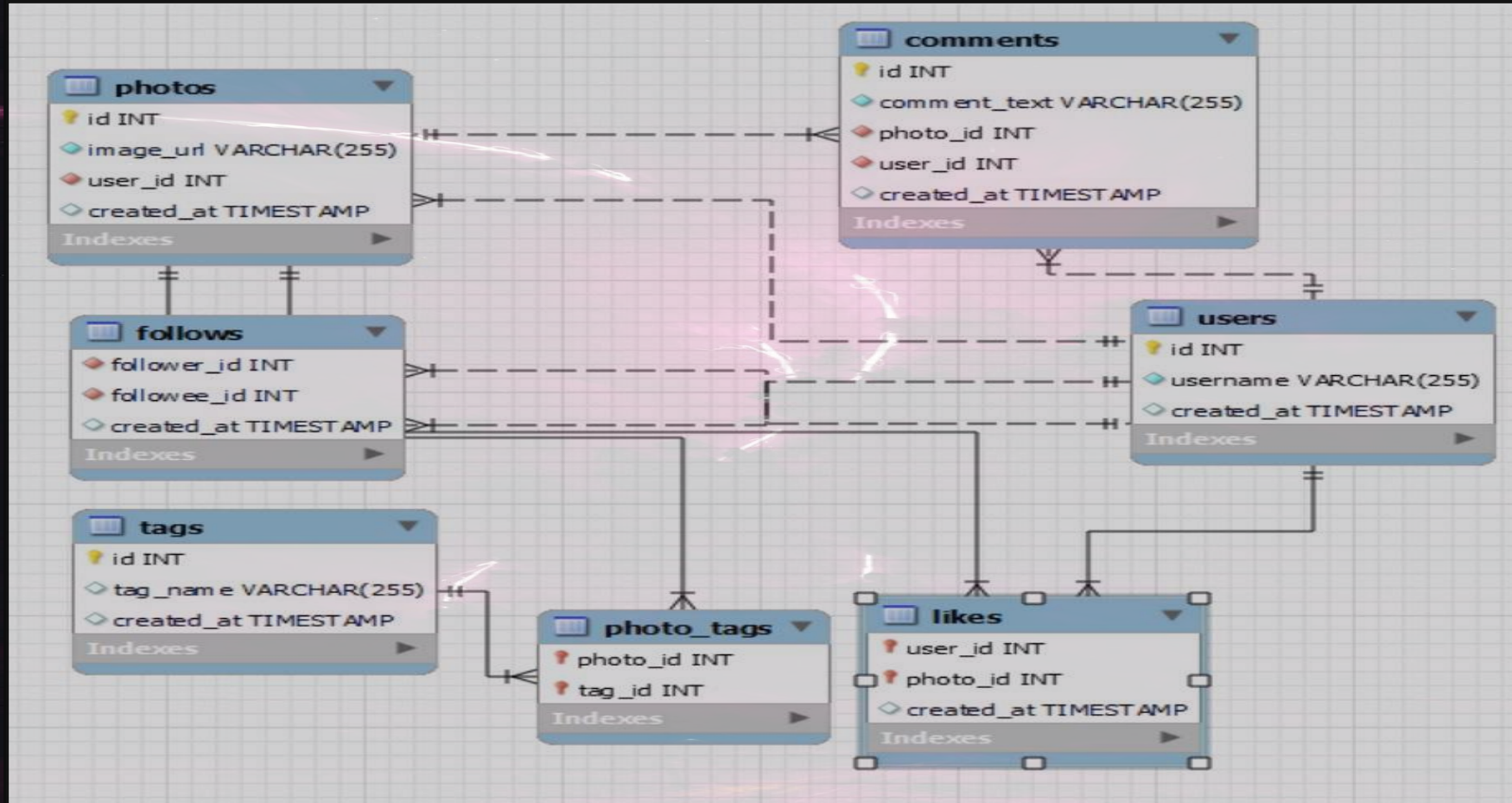
```
CREATE TABLE TAGS(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  tag_name VARCHAR(255) UNIQUE,  
  created_at TIMESTAMP DEFAULT NOW()  
);
```

PHOTO TAGS TABLE

```
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)  
);
```

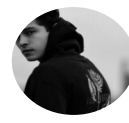

ER Diagram of Schema

Our work



135 likes





Example Data :

```
INSERT INTO USERS (username, created_at)
VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'),
       ('Andre_Purdy85', '2017-04-02 17:11:21.417'),
       ('Harley_Lind18', '2017-02-21 11:12:32.574'),
       ('Arely_Bogan63', '2016-08-13 01:28:43.085'),
       ('Aniya_Hackett', '2016-12-07 01:04:39.298');
```

```
INSERT INTO COMMENTS(comment_text, user_id, photo_id)
VALUES ('unde at dolore', 2, 1),
       ('quae ea ducimus', 3, 1),
       ('alias a voluptatum', 5, 1),
       ('facere suscipit sunt', 14, 1),
       ('totam eligendi quaerat', 17, 1);
```

```
INSERT INTO PHOTOS(image_url, user_id)
VALUES ('http://elijah.biz', 1),
       ('https://shanon.org', 2),
       ('http://vicky.biz', 2),
       ('http://oleta.net', 3),
       ('https://jennings.biz', 1);
```

```
INSERT INTO LIKES(user_id, photo_id)
VALUES (2, 1), (5, 1), (9, 1), (10, 1),
       (52, 42), (54, 42), (56, 42),
       (57, 42), (62, 42) (11, 1), (14, 1),
       (19, 1), (40, 9), (14, 20), (16, 20);
```





Example Data :

```
INSERT INTO FOLLOWS(follower_id, followee_id)
VALUES (2, 1), (2, 3), (2, 4), (2, 21),
       (12, 5), (12, 6), (12, 7), (12, 8),
       (23, 21), (23, 14), (23, 15), (23, 16),
       (3, 17), (3, 18), (3, 19), (3, 20),
       (14, 32), (14, 33), (14, 34), (14, 35);
```

```
INSERT INTO TAGS(tag_name)
VALUES ('sunset'), ('photography'), ('sunrise'),
       ('landscape'), ('food'), ('foodie'),
       ('delicious'), ('beauty'), ('stunning'),
       ('dreamy'), ('lol'), ('happy'), ('fun');
```

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



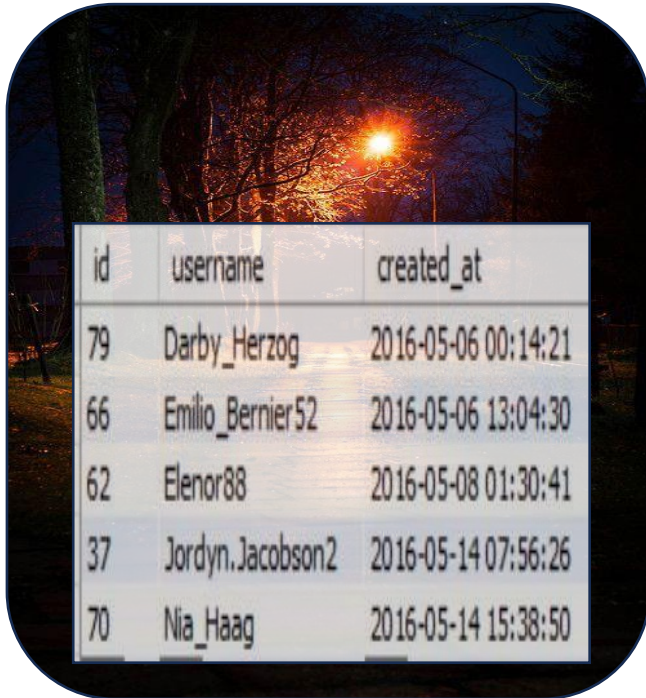
1947 likes



Our work



Hareley_lind18



id	username	created_at
79	Darby_Herzog	2016-05-06 00:14:21
66	Emilio_Bernier52	2016-05-06 13:04:30
62	Elenor88	2016-05-08 01:30:41
37	Jordyn.Jacobson2	2016-05-14 07:56:26
70	Nia_Haag	2016-05-14 15:38:50

FIND THE 5 OLDEST USERS:

SELECT

*

FROM

USERS

ORDER BY created_at

LIMIT 5;



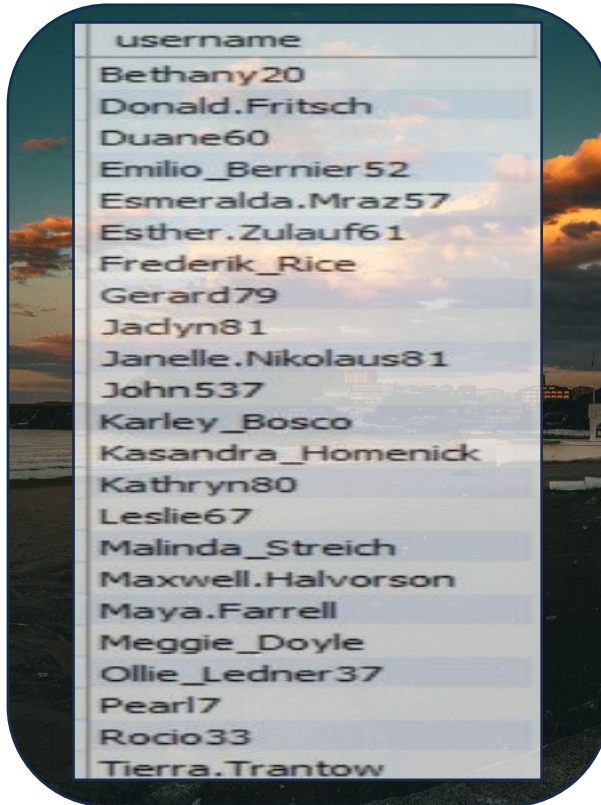
2015 likes



Our work



Annalise.McKenzie16



username
Bethany20
Donald.Fritsch
Duane60
Emilio_Bernier52
Esmeralda.Mraz57
Esther.Zulauf61
Frederik_Rice
Gerard79
Jadyn81
Janelle.Nikolaus81
John537
Karley_Bosco
Kassandra_Homenick
Kathryn80
Leslie67
Malinda_Streich
Maxwell.Halvorson
Maya.Farrell
Meggie_Doyle
Ollie_Ledner37
Pearl7
Rocio33
Tierra.Trantow

IDENTIFY THE INACTIVE USER(Users with no photos):

```
SELECT  username
FROM
    USERS
    LEFT JOIN
    PHOTOS ON USERS.id = PHOTOS.user_id
WHERE
    PHOTOS.id IS NULL;
```



215 likes



Our work



Nia_Haag



MOST POPULAR REGISTRATION DAY:

SELECT

DAYNAME(created_at) AS DAYS,
COUNT(*) AS TOTAL

FROM

USERS

GROUP BY DAYS

ORDER BY TOTAL DESC

LIMIT 2;



89 likes



Our work



Tabitha_Schamberger11



username	id	image_url	TOTAL
Linnea59	145	https://jarret.name	48
Mike.Auer39	182	https://dorcas.biz	43
Harrison.Beatty50	127	https://celestine.name	43
David.Osinski47	123	http://shannon.org	42
Presley_McClure	30	http://kenny.com	41
Florence99	174	https://delbert.net	41
Jaylan.Lakin	192	https://anahi.info	41
Peter.Stehr0	147	https://adela.com	41
Delpha.Kihn	61	https://dejon.name	41
Annalise.McKenzie16	52	https://hershel.com	41

IDENTIFY THE TOP 10 MOST LIKED PHOTOS (and users who created them):

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 10;



2035 likes



Our work



Keenan.Schamberger60



tag_name	total
smile	59
beach	42
party	39
fun	37
lol	24

FIVE MOST COMMONLY USED HASHTAGS :

```
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 ;
```



2035 likes



Our work



Presley_McClure

username	num_likes
Aniya_Hackett	255
Aurelie71	255
Donald.Fritsch	255
Emilio_Bernier52	255
Frederik_Rice	255
Jadyn81	255
Janelle.Nikolaus81	255
Kathryn80	255
Maxwell.Halvorson	255
Maya.Farrell	255
Meggie_Doyle	255
Ollie_Ledner37	255
Rocio33	255

FIND THE BOTS (USER WHO HAVE LIKED EVERY SINGLE PHOTO) :

```
SELECT
    username,
    count(*) AS num_likes
FROM
    USERS
    INNER JOIN
    LIKES ON USERS.id = LIKES.user_id
GROUP BY LIKES.user_id
HAVING num_likes = (SELECT count(*)
                    FROM photos);
```



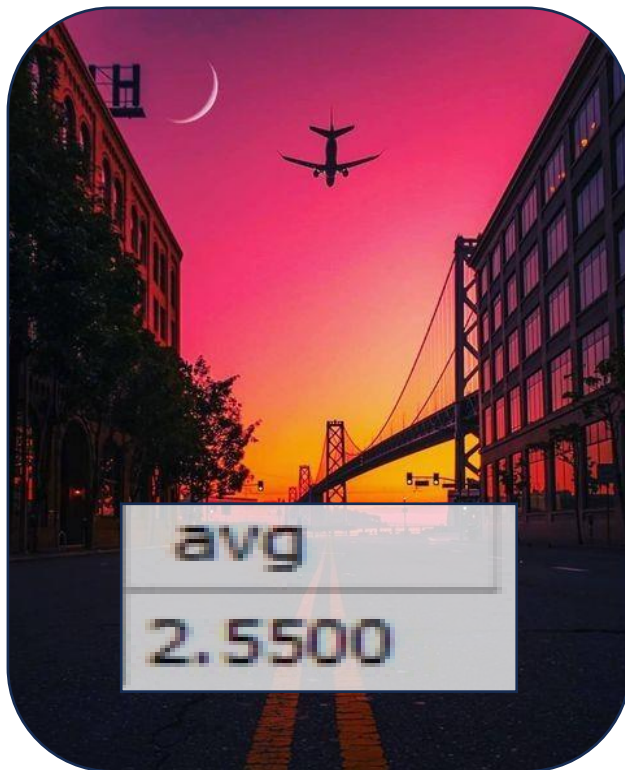
235 likes



Our work



Darwin29



CALCULATE AVGRAGE NUMBER OF PHOTOS PER USER :

```
SELECT  
(SELECT Count(*)  
FROM PHOTOS) / (SELECT Count(*)  
FROM USERS) AS AVG;
```



1035 likes

Our work



Maxwell.Halvorson

id	tag_name	total
1	sunset	5
8	beauty	4
7	delicious	4
5	food	4
6	foodie	4
4	landscape	4
2	photography	4
21	smile	4
11	lol	3
17	party	3
13	fun	2
14	style	2
10	dreamy	1
19	drunk	1
12	happy	1
9	stunning	1
3	sunrise	1

NUMBERS OF TAGS USED BY USERS :

```
SELECT
    id,
    tag_name,
    COUNT(tag_id) AS TOTAL
FROM
    PHOTO_TAGS
    JOIN
    TAGS ON PHOTO_TAGS.photo_id = TAGS.id
GROUP BY photo_id
ORDER BY TOTAL DESC;
```



135 likes





Instagram

THANK YOU!

avantikaraj7360@gmail.com