

# INSTAGRAM USERS ANALYTICS.



# 1. Rewarding Most Loyal Users: People who have been using the platform for the longest time.

Your Task: Find the 5 oldest users of the Instagram from the database provided

The screenshot shows the db-fiddle.com web interface. The top navigation bar includes a database icon, the text 'Database: MySQL v8.0', and buttons for 'Run', 'Save', 'Load Example', and 'Collaborate'. On the right, there are links for 'Sign in', 'Have any feedback?', and a Twitter icon.

On the left sidebar, there is a 'Fiddle Title' input field (50 characters remaining), a 'Fiddle Description' input field (300 characters remaining), a 'Private Fiddle' toggle switch (set to 'PRO'), and an 'Upgrade to PRO' button with a '50% OFF for Early Adopters' offer. At the bottom of the sidebar is a 'Show Keyboard Shortcuts' link and a Shutterstock advertisement.

The main area is divided into three panels:

- Schema SQL:** Contains the following SQL code:

```
1 id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,  
2 username VARCHAR(255) NOT NULL,  
3 created_at TIMESTAMP DEFAULT NOW()  
4 );  
5  
6 /*Photos*/  
7 CREATE TABLE photos(  
8 id INT AUTO_INCREMENT PRIMARY KEY,  
9 image_url VARCHAR(355) NOT NULL,  
10 user_id INT NOT NULL,  
11 created_at TIMESTAMP DEFAULT NOW(),  
12 FOREIGN KEY(user_id) REFERENCES users(id)  
13 );  
14  
15 /*Comments*/  
16 CREATE TABLE comments(  
17 id INT AUTO_INCREMENT PRIMARY KEY,  
18 user_id INT NOT NULL,  
19 comment_text VARCHAR(255) NOT NULL,  
20 created_at TIMESTAMP DEFAULT NOW(),  
21 FOREIGN KEY(user_id) REFERENCES users(id)  
22 );
```
- Query SQL:** Contains the following SQL code:

```
1 /** Rewarding Most Loyal Users: People who have been using the platform for the  
2 longest time.  
3 Your Task: Find the 5 oldest users of the Instagram from the database provided */  
4  
5 SELECT  
6 username,  
7 created_at  
8 FROM  
9 ig_clone.users  
10 ORDER BY created_at  
11 LIMIT 5;
```
- Results:** Displays the output of the query as a table with two columns: 'username' and 'created\_at'. The results are as follows:

username	created_at
Darby_Herzog	2016-05-06 00:14:21
Emilio_Bernier52	2016-05-06 13:04:30
Elenor88	2016-05-08 01:30:41
Nicole71	2016-05-09 17:30:22
Jordyn_Jacobson2	2016-05-14 07:56:26

## 2. Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.

Your Task: Find the users who have never posted a single photo on Instagram

The screenshot shows the DB Fiddle web application interface. The top navigation bar includes a 'Trainity Dashboard' link, a 'DB Fiddle - SQL Database Playground' tab, and a '+'. The main header bar displays the database as 'MySQL v8.0', along with 'Run', 'Save', 'Load Example', and 'Collaborate' buttons. On the right of the header are 'Sign in' and 'Have any feedback?' links, and a Twitter icon.

On the left sidebar, there is a 'Fiddle Title' field (50 characters remaining), a 'Fiddle Description' field (300 characters remaining), a 'Private Fiddle' toggle (currently off), and an 'Upgrade to PRO' button with a '50% OFF for Early Adopters' offer. Below this is a 'Show Keyboard Shortcuts' link and a 'shutterstock' advertisement for a free trial.

The main content area is divided into three sections:

- Schema SQL:** Contains the following SQL code:

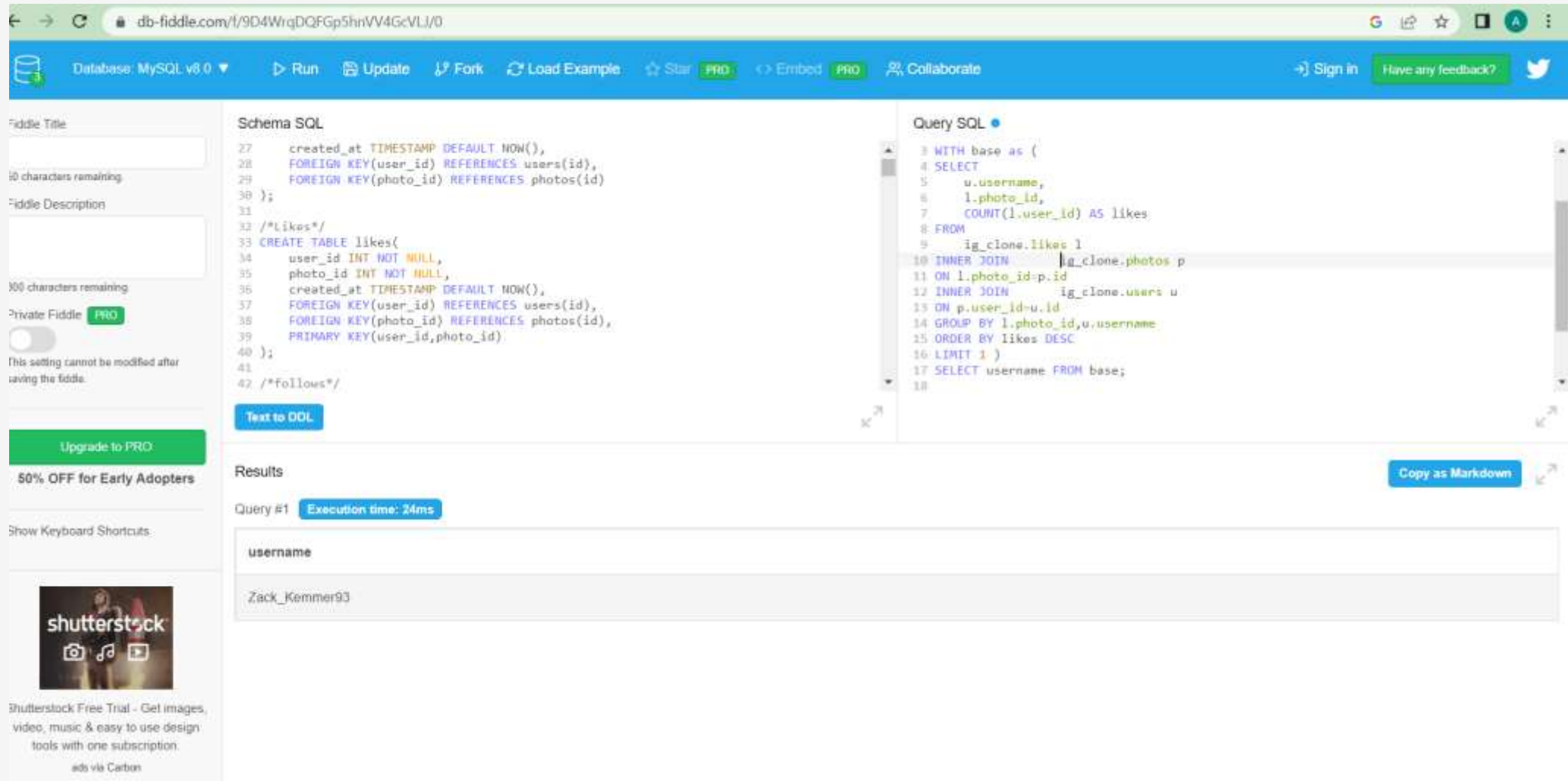
```
1 id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,  
2 username VARCHAR(255) NOT NULL,  
3 created_at TIMESTAMP DEFAULT NOW()  
4 );  
5  
6 /*Photos*/  
7 CREATE TABLE photos(  
8 id INT AUTO_INCREMENT PRIMARY KEY,  
9 image_url VARCHAR(355) NOT NULL,  
10 user_id INT NOT NULL,  
11 created_at TIMESTAMP DEFAULT NOW(),  
12 FOREIGN KEY(user_id) REFERENCES users(id)  
13 );  
14  
15 /*Comments*/  
16 CREATE TABLE comments(  
17
```
- Query SQL:** Contains the following SQL code:

```
1 /** Remind Inactive Users to Start Posting: By sending them promotional emails to  
2 post their 1st photo.  
3 Your Task: Find the users who have never posted a single photo on Instagram */  
4  
5 SELECT  
6 u.username  
7 FROM  
8 ig_clone.users u  
9 LEFT JOIN  
10 ig_clone.photos p  
11 ON u.id=p.user_id  
12 WHERE  
13 p.user_id is null  
14 ORDER BY  
15 u.username;
```
- Results:** Shows the execution of the query. The 'Query #1' status bar indicates 'Execution time: 1ms'. The results are displayed in a table with the following data:

username
Aniya_Hackett
Bartholome.Bernhard
Bethany20
Darby_Herzog
David_Osinski47

At the bottom of the interface, there is a footer with the text 'DB Fiddle - Crafted with ♥ by Status200 in the United Kingdom.' and a link to 'Terms of Use • Privacy • Cookie Policy • Status200 Ltd © 2018'.

**3.Declaring Contest Winner:** The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.  
Your Task: Identify the winner of the contest and provide their details to the team



The screenshot shows the db-fiddle.com interface. The left sidebar contains a 'Fiddle Title' field, a 'Fiddle Description' field, a 'Private Fiddle' toggle, and an 'Upgrade to PRO' button. The main area is divided into three panels: 'Schema SQL', 'Query SQL', and 'Results'.

**Schema SQL:**

```
27 created_at TIMESTAMP DEFAULT NOW(),
28 FOREIGN KEY(user_id) REFERENCES users(id),
29 FOREIGN KEY(photo_id) REFERENCES photos(id)
30 );
31
32 /*likes*/
33 CREATE TABLE likes(
34   user_id INT NOT NULL,
35   photo_id INT NOT NULL,
36   created_at TIMESTAMP DEFAULT NOW(),
37   FOREIGN KEY(user_id) REFERENCES users(id),
38   FOREIGN KEY(photo_id) REFERENCES photos(id),
39   PRIMARY KEY(user_id,photo_id)
40 );
41
42 /*follows*/
```

**Query SQL:**

```
3 WITH base as (
4 SELECT
5   u.username,
6   l.photo_id,
7   COUNT(l.user_id) AS likes
8 FROM
9   ig_clone.likes l
10 INNER JOIN ig_clone.photos p
11 ON l.photo_id=p.id
12 INNER JOIN ig_clone.users u
13 ON p.user_id=u.id
14 GROUP BY l.photo_id,u.username
15 ORDER BY likes DESC
16 LIMIT 1 )
17 SELECT username FROM base;
```

**Results:**

Query #1 Execution time: 24ms

username
Zack_Kemmer93

At the bottom of the sidebar, there is a Shutterstock Free Trial advertisement.

## 4.Hashtag Researching: A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.

Your Task: Identify and suggest the top 5 most commonly used hashtags on the platform

db-fiddle.com/l/9D4WrqDQFGp5hnVV4GcVLI/0

Database: MySQL v8.0

Run

Update

Fork

Load Example

Star

PRO

Embed

PRO

Collaborate

Sign in

Have any feedback?

Fiddle Title

50 characters remaining

Fiddle Description

300 characters remaining

Private Fiddle

PRO

Upgrade to PRO

50% OFF for Early Adopters

Show Keyboard Shortcuts

Schema SQL

```
1 CREATE TABLE tags(
2   id INTEGER AUTO_INCREMENT PRIMARY KEY,
3   tag_name VARCHAR(255) UNIQUE NOT NULL,
4   created_at TIMESTAMP DEFAULT NOW()
5 );
6
7 /*function table: Photos - Tags*/
8 CREATE TABLE photo_tags(
9   photo_id INT NOT NULL,
10  tag_id INT NOT NULL,
11  FOREIGN KEY(photo_id) REFERENCES photos(id),
12  FOREIGN KEY(tag_id) REFERENCES tags(id),
13  PRIMARY KEY(photo_id,tag_id)
14 );
```

Text to DDL

Query SQL

```
1 /** Hashtag Researching: A partner brand wants to know, which hashtags to use in
2   the post to reach the most people on the platform.
3   Your Task: Identify and suggest the top 5 most commonly used hashtags on the
4   platform **/
5
6 SELECT
7   t.tag_name,
8   COUNT(pt.photo_id) as Ctags
9 FROM
10  ig_clone.tags t
11 INNER JOIN
12  ig_clone.photo_tags pt
13 ON pt.tag_id=t.id
14 GROUP BY t.tag_name
15 ORDER BY Ctags DESC
16 Limit 5
```

Results

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

Copy as Markdown

## 5.Launch AD Campaign: The team wants to know, which day would be the best day to launch ADs.

Your Task: What day of the week do most users register on? Provide insights on when to schedule an ad campaign

Answer:- HERE, days count start from Monday-0, Tuesday-1,.....,Sunday-6 . Hence,Thursdays and Sunday are good for ad campaign.

db-fiddle.com/f/9D4WrqDQFGp5hnVV4GcVLJ/0

Database: MySQL v8.0 Run Update Fork Load Example Star PRO Embed PRO Collaborate Sign in Have any feedback?

Fiddle Title

50 characters remaining

Fiddle Description

300 characters remaining

Private Fiddle PRO

Upgrade to PRO

50% OFF for Early Adopters

Show Keyboard Shortcuts

Schema SQL

```
13:26:14.496'), ('Kassandra_Homenick', '2016-12-12 06:50:07.996'), ('Tobitha_Schamberger11', '2016-08-20 02:19:45.512'), ('Gus93', '2016-06-24 19:36:30.978'), ('Presley_McClure', '2016-08-07 16:25:48.561'), ('Justina_Gaylord27', '2017-05-04 16:32:15.577'), ('Dereck65', '2017-01-19 01:34:14.296'), ('Alexandro35', '2017-03-29 17:09:02.344'), ('Jaclyn81', '2017-02-06 23:29:16.394'), ('Billy52', '2016-10-05 14:10:20.453'), ('Annalise_McKenzie16', '2016-08-02 21:32:45.646'), ('Norbert_Carroll35', '2017-02-06 22:05:43.425'), ('Odessa2', '2016-10-21 18:16:56.390'), ('Hailes26', '2017-04-29 18:53:39.650'), ('Delpha_Kihn', '2016-08-31 02:42:30.288'), ('Macio33', '2017-01-23 11:51:15.467'), ('Kenneth64', '2016-12-27 09:48:17.380'), ('Eveline95', '2017-01-23 23:14:18.569'), ('Maxwell_Halvarson', '2017-04-18 02:32:43.597'), ('Tierra_Trantow', '2016-10-03 12:49:20.774'), ('Josianne_Friesen', '2016-06-07 12:47:00.703'), ('Darwin29', '2017-03-18 03:10:07.047'), ('Dario77', '2016-08-18 07:15:02.823'), ('Jaime53', '2016-09-11 18:51:56.965'), ('Kaley9', '2016-09-23 21:24:20.222'), ('Aiyana_Hoeger', '2016-09-29 20:28:12.457'), ('Irwin_Larson', '2016-08-26 19:36:22.199'), ('Yvette_Gottlieb91', '2016-11-14 12:32:01.405'), ('Pearl7', '2016-
```

Text to DDL

Query SQL

```
1 /** Launch AD Campaign: The team wants to know, which day would be the best day to launch ADs.
2 Your Task: What day of the week do most users register on? Provide insights on when to schedule an ad campaign */
3
4 SELECT
5     WEEKDAY(created_at),
6     COUNT(username) as num_users
7 FROM
8     ig_clone.users
9 GROUP BY WEEKDAY(created_at)
10 order by num_users DESC
11
12
13
```

Results

Query #1 Execution time: 1ms

WEEKDAY(created_at)	num_users
3	16
6	16
4	15
1	14
0	14

Copy as Markdown



B1.User Engagement: Are users still as active and post on Instagram or they are making fewer posts

Your Task: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users

Schema SQL ●

14), (111, 16), (112, 21), (112, 9), (112, 20), (116, 7), (117, 21), (118, 20),  
(118, 4), (118, 1), (120, 21), (120, 19), (120, 17), (121, 21), (121, 13), (121,  
11), (121, 12), (122, 15), (123, 17), (123, 14), (123, 15), (124, 15), (124, 16),  
(124, 17), (125, 11), (125, 12), (126, 2), (126, 3), (126, 4), (127, 21), (131, 5),  
(132, 19), (132, 17), (132, 13), (132, 18), (132, 21), (133, 15), (133, 17), (133,  
16), (133, 14), (134, 21), (134, 11), (134, 12), (134, 13), (135, 17), (136, 8),  
(136, 9), (136, 21), (136, 20), (136, 10), (137, 11), (137, 13), (137, 21), (137,  
12), (138, 9), (138, 20), (138, 10), (139, 5), (140, 11), (140, 12), (141, 17),  
(142, 21), (142, 20), (145, 18), (145, 13), (145, 19), (145, 17), (145, 21), (147,  
5), (148, 11), (149, 20), (149, 4), (149, 1), (149, 2), (152, 16), (152, 17), (152,  
15), (152, 14), (153, 15), (153, 16), (153, 17), (153, 14), (154, 21), (155, 10),  
(155, 21), (155, 20), (155, 9), (155, 8), (156, 6), (156, 5), (157, 13), (157, 11),  
(157, 12), (157, 21), (158, 15), (159, 5), (160, 5), (161, 11), (162, 15), (162,  
14), (162, 17), (163, 21), (163, 13), (163, 19), (166, 13), (166, 17), (168, 6),  
(168, 5), (169, 12), (171, 10), (171, 21), (171, 9), (171, 8), (171, 20), (172,  
15), (172, 16), (172, 14), (173, 16), (173, 15), (173, 17), (174, 5), (175, 17),

Text to DDL

Query SQL ●

```
4 SUM(photosid)/COUNT(usersid) as photos_per_user
5 FROM
6 (
7 SELECT
8     u.id AS usersid,
9     COUNT(p.id) AS photosid
10 FROM
11     ig_clone.users u
12 LEFT JOIN
13     ig_clone.photos p
14 ON u.id=p.user_id
15 GROUP BY u.id
16 )
17 base
18 WHERE photosid > 0
19
```

Results

Copy as Markdown

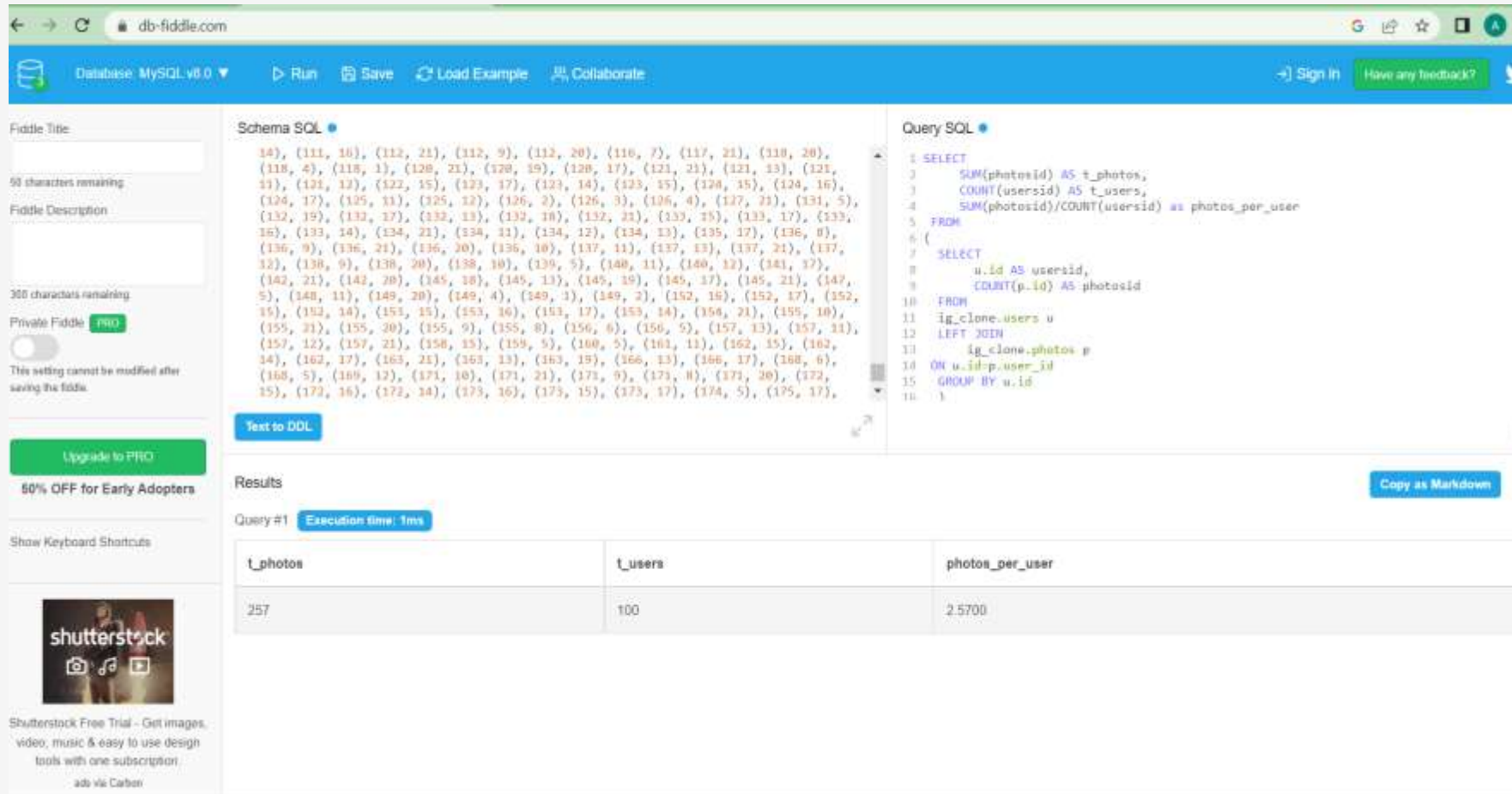
Query #1

Execution time: 1ms

t_photos	t_users	photos_per_user
257	74	3.4730

## B1.User Engagement: Are users still as active and post on Instagram or they are making fewer posts

Your Task: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users



The screenshot shows the db-fiddle.com web application. The interface includes a sidebar on the left with options like 'Fiddle Title', 'Fiddle Description', and 'Private Fiddle'. The main area is divided into three sections: 'Schema SQL' (containing a large block of sample data), 'Query SQL' (containing a SQL query to calculate average photos per user), and 'Results' (displaying the query output in a table).

**Schema SQL**

```
14), (111, 16), (112, 21), (112, 9), (112, 20), (116, 7), (117, 21), (118, 20),  
(118, 4), (118, 1), (120, 21), (120, 19), (120, 17), (121, 21), (121, 13), (121,  
21), (121, 12), (122, 15), (123, 17), (123, 14), (123, 15), (124, 15), (124, 16),  
(124, 17), (125, 11), (125, 12), (126, 2), (126, 3), (126, 4), (127, 21), (131, 5),  
(132, 19), (132, 17), (132, 13), (132, 18), (132, 22), (133, 15), (133, 17), (133,  
16), (133, 14), (134, 21), (134, 11), (134, 12), (134, 13), (135, 17), (136, 8),  
(136, 9), (136, 21), (136, 20), (136, 10), (137, 11), (137, 13), (137, 21), (137,  
12), (138, 9), (138, 20), (138, 10), (139, 5), (140, 11), (140, 12), (141, 17),  
(142, 21), (142, 20), (145, 18), (145, 13), (145, 19), (145, 17), (145, 21), (147,  
5), (148, 11), (149, 20), (149, 4), (149, 1), (149, 2), (152, 16), (152, 17), (152,  
15), (152, 14), (153, 15), (153, 16), (153, 17), (153, 14), (154, 21), (155, 10),  
(155, 21), (155, 20), (155, 9), (155, 8), (156, 6), (156, 5), (157, 13), (157, 11),  
(157, 12), (157, 21), (158, 15), (159, 5), (160, 5), (161, 11), (162, 15), (162,  
14), (162, 17), (163, 21), (163, 13), (163, 19), (166, 13), (166, 17), (168, 6),  
(168, 5), (169, 12), (171, 10), (171, 21), (171, 9), (171, 8), (171, 20), (172,  
15), (172, 16), (172, 14), (173, 16), (173, 15), (173, 17), (174, 5), (175, 17),
```

**Query SQL**

```
1 SELECT  
2   SUM(photosid) AS t_photos,  
3   COUNT(usersid) AS t_users,  
4   SUM(photosid)/COUNT(usersid) AS photos_per_user  
5 FROM  
6 (  
7   SELECT  
8     u.id AS usersid,  
9     COUNT(p.id) AS photosid  
10  FROM  
11    ig_clone.users u  
12  LEFT JOIN  
13    ig_clone.photos p  
14  ON u.id=p.user_id  
15  GROUP BY u.id  
16 )
```

**Results**

Query #1 Execution time: 1ms

t_photos	t_users	photos_per_user
257	100	2.5700



B2.Bots & Fake Accounts: The investors want to know if the platform is crowded with fake and dummy accounts

Your Task: Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this)

Fiddle Title

50 characters remaining

Fiddle Description


300 characters remaining

Private Fiddle: PRO

Upgrade to PRO

50% OFF for Early Adopters

Show Keyboard Shortcuts



Schema SQL

```
14), (111, 16), (112, 21), (112, 9), (112, 20), (116, 7), (117, 21), (118, 20),
(118, 4), (119, 1), (120, 21), (120, 19), (120, 17), (121, 21), (121, 13), (123,
11), (121, 12), (122, 15), (123, 17), (123, 14), (123, 15), (124, 15), (124, 16),
(124, 17), (125, 11), (125, 12), (126, 2), (126, 3), (126, 4), (127, 21), (131, 5),
(132, 19), (132, 17), (132, 13), (132, 10), (132, 21), (133, 19), (133, 17), (133,
16), (133, 14), (134, 21), (134, 11), (134, 12), (134, 13), (135, 17), (136, 0),
(136, 9), (136, 21), (136, 20), (136, 10), (137, 11), (137, 13), (137, 21), (137,
12), (138, 9), (138, 20), (138, 10), (138, 5), (140, 11), (140, 12), (141, 17),
(142, 21), (142, 20), (145, 18), (145, 13), (145, 19), (145, 17), (145, 21), (147,
5), (148, 11), (148, 20), (149, 4), (149, 1), (149, 2), (152, 16), (152, 17), (152,
15), (152, 14), (153, 15), (153, 16), (153, 17), (153, 14), (154, 21), (155, 10),
(155, 21), (155, 20), (155, 9), (155, 8), (156, 6), (156, 5), (157, 13), (157, 11),
(157, 12), (157, 21), (158, 15), (159, 5), (160, 5), (162, 11), (162, 15), (162,
14), (162, 17), (163, 21), (163, 13), (163, 19), (166, 13), (166, 17), (168, 6),
(168, 5), (169, 12), (171, 10), (171, 21), (171, 9), (171, 8), (171, 20), (172,
15), (172, 16), (172, 14), (173, 16), (173, 15), (173, 17), (174, 5), (175, 17),
```

Text to DDL

Query SQL

```
1 WITH pcount AS (
2   SELECT
3     user_id,
4     COUNT(photo_id) AS n_like
5   FROM
6     ig_clone.likes
7   GROUP BY
8     user_id
9   ORDER BY
10    n_like DESC
11 )
12 SELECT *
13 FROM pcount
14 WHERE
15   n_like=(SELECT COUNT(*) FROM ig_clone.photos)
16
```

Copy as Markdown

Results

Query #1 Execution time: 3ms

user_id	n_like
75	257
21	257
24	257
91	257
36	257

DB Fiddle - Created with by StatuS200 in the United Kingdom

Terms of Use - Privacy - Cookie Policy - StatuS200 Ltd © 2018