User Analytic

User analytics is a way of analyzing user data in order to give companies a clearer view of user cohorts. Typically, a business intelligence software is used to combine customer behavior data from web and mobile applications to create a holistic view of the user and the user experience.

Example

- Users were dropping out of the new user registration process on its portal for two reasons: Adding an extra space at the end of a name created an error message, and a third-party authentication tool was taking too long to load. The team shared that information with their development teams and the third-party partner to resolve the issues.

Project Description-

Instagram Project- SQL

You are working with the product team of Instagram and the product manager has asked you to provide insights on the questions asked by the management team.

Tech-Stack Used

MySQL installer community 8.0.30.0.msi was used for this project it contains:- -MySQL Workbench 8.0 CE -MySQL Shell -MySQL Command Line Client

Installation Steps:- https://www.youtube.com/watch?v=eq-e_n7lm2M

Approach-

I have used MY SQL Workbench which is a cross-platform, open-source, visual tool for database management. I have analyzed the database carefully. Observe all the tables, columns, rows, and relationship among all the tables and successfully created a Instagram database and Tables-Users, Photos, Comments , Likes, follows, Tags, junction table: Photos – Tags. After Inserting data into these table , Enhanced Entity-Relationship (EER) diagrams is used to provide a visual representation of the relationships among the tables in model and after that arranged the data in various queries to obtain desired output.

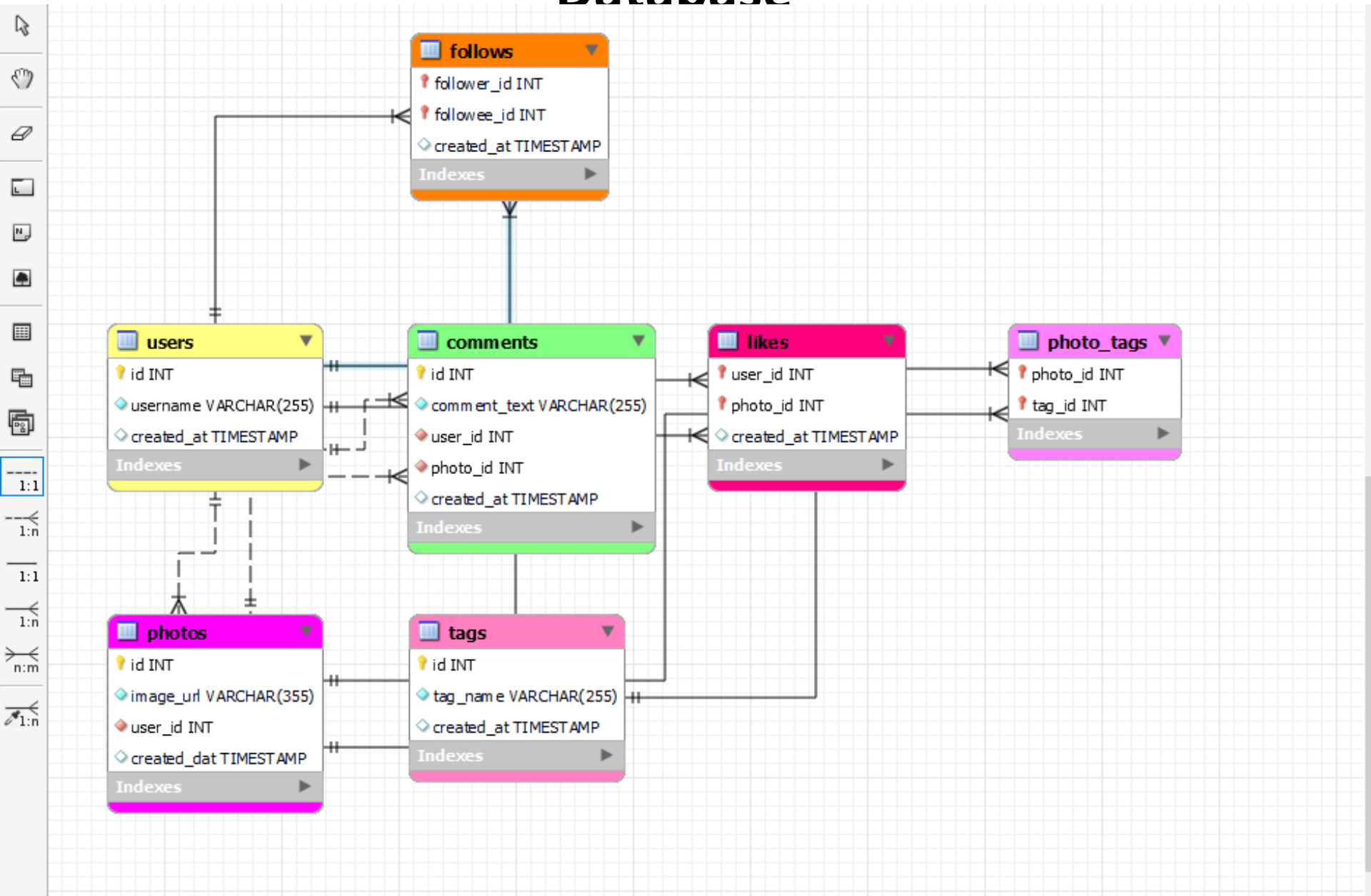
.

Creating a Database

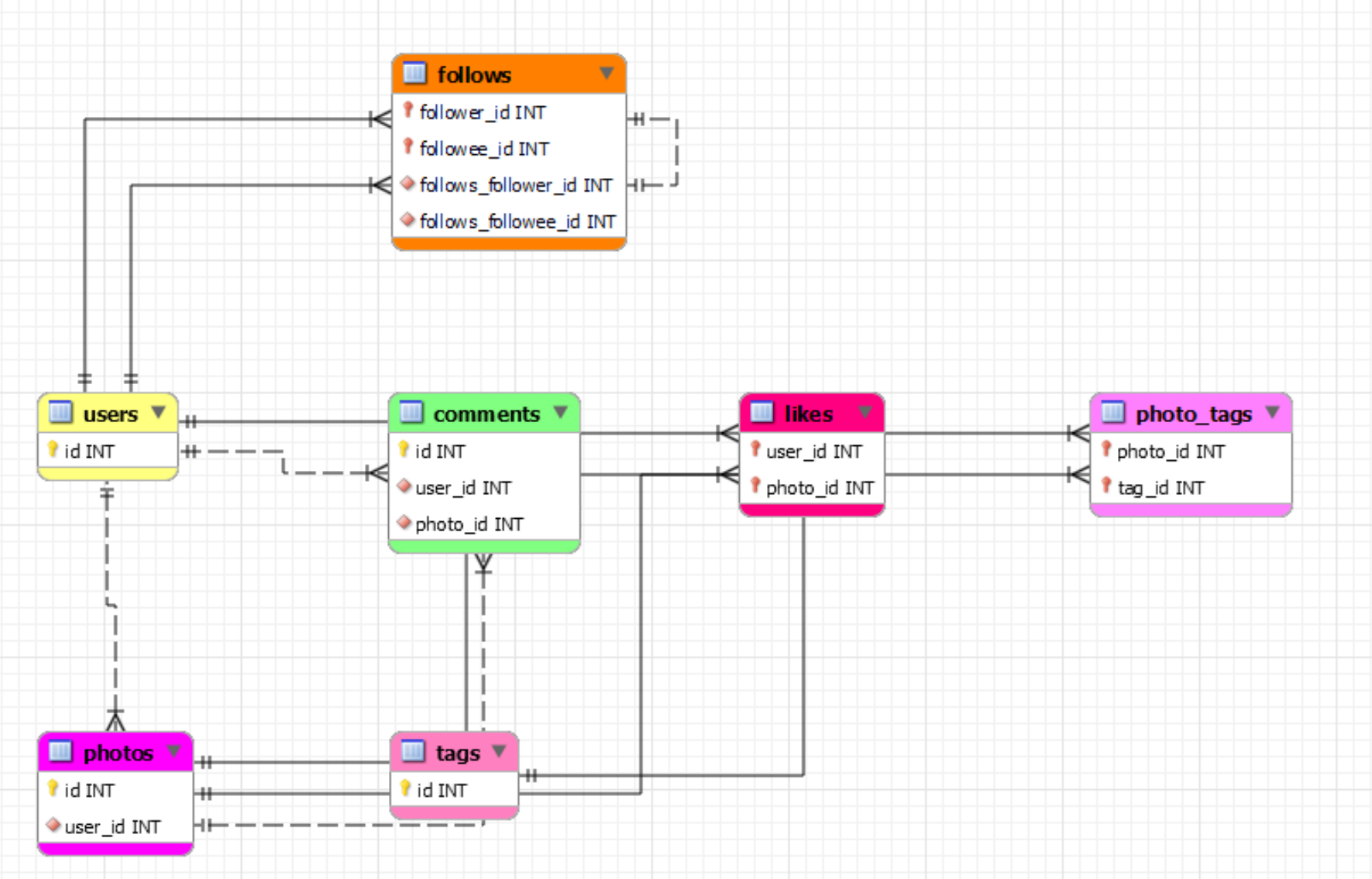
```sql
CREATE DATABASE Instagram2;
USE Instagram2;
CREATE TABLE users(
    id INT AUTO_INCREMENT UNIQUE PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);
/*Photos*/
CREATE TABLE photos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_url VARCHAR(355) NOT NULL,
    user_id INT NOT NULL,
    created_dat TIMESTAMP DEFAULT NOW(),
    FOREIGN KEY(user_id) REFERENCES users(id)
);
/*Comments*/
CREATE TABLE comments(
    id INT AUTO_INCREMENT PRIMARY KEY,
    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)
);
```

```sql
/*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)
);
/*follows*/
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)
);
/*Tags*/
CREATE TABLE tags(
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    tag_name VARCHAR(255) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT NOW()
);
```

EER Diagram for project of Instagram Database



Visual representation of Primary and foreign key only



Inserting values in a Database

```
INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-02-16 18:22:10.846'), ('Andre_Purdy85', '2017-04-02
```

```
INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('https://shanon.org', 1), ('http://vicky.biz', 1),
```

```
INSERT INTO follows(follower_id, followee_id) VALUES (2, 1), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10
```

```
INSERT INTO comments(comment_text, user_id, photo_id) VALUES ('unde at dolorem', 2, 1), ('quae ea ducimus', 3, 1), ('alias
```

```
INSERT INTO likes(user_id,photo_id) VALUES (2, 1), (5, 1), (9, 1), (10, 1), (11, 1), (14, 1), (19, 1), (21, 1), (24, 1), (3
```

```
INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'), ('landscape'), ('food'), ('foodie'), ('deliciou
```

Output

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	00:01:19	CREATE DATABASE Instagram	1 row(s) affected	0.047 sec
2	00:01:19	USE Instagram	0 row(s) affected	0.000 sec
3	00:01:19	CREATE TABLE users(id INT AUTO_INCREMENT UNIQUE PRIMARY KE...	0 row(s) affected	0.140 sec
4	00:01:19	CREATE TABLE photos(id INT AUTO_INCREMENT PRIMARY KEY, imag...	0 row(s) affected	0.063 sec
5	00:01:20	CREATE TABLE comments(id INT AUTO_INCREMENT PRIMARY KEY, c...	0 row(s) affected	0.047 sec
6	00:01:20	CREATE TABLE likes(user_id INT NOT NULL, photo_id INT NOT NULL, c...	0 row(s) affected	0.031 sec
7	00:01:20	CREATE TABLE follows(follower_id INT NOT NULL, followee_id INT NOT ...	0 row(s) affected	0.047 sec
8	00:01:20	CREATE TABLE tags(id INTEGER AUTO_INCREMENT PRIMARY KEY, t...	0 row(s) affected	0.031 sec
9	00:01:20	CREATE TABLE photo_tags(photo_id INT NOT NULL, tag_id INT NOT N...	0 row(s) affected	0.031 sec
10	00:01:20	INSERT INTO users (username, created_at) VALUES ('Kenton_Kirlin', '2017-...	100 row(s) affected Records: 100 Duplicates: 0 Warnings: 0	0.032 sec
11	00:01:20	INSERT INTO photos(image_url, user_id) VALUES ('http://elijah.biz', 1), ('htt...	257 row(s) affected Records: 257 Duplicates: 0 Warnings: 0	0.015 sec
12	00:01:20		7623 row(s) affected Records: 7623 Duplicates: 0 Warnings: 0	0.406 sec
13	00:01:20		7488 row(s) affected Records: 7488 Duplicates: 0 Warnings: 0	0.391 sec
14	00:01:21		8782 row(s) affected Records: 8782 Duplicates: 0 Warnings: 0	0.359 sec
15	00:01:21	INSERT INTO tags(tag_name) VALUES ('sunset'), ('photography'), ('sunrise'),...	21 row(s) affected Records: 21 Duplicates: 0 Warnings: 0	0.000 sec
16	00:01:21	INSERT INTO photo_tags(photo_id, tag_id) VALUES (1, 18), (1, 17), (1, 21),...	501 row(s) affected Records: 501 Duplicates: 0 Warnings: 0	0.047 sec

A detailed report answering the questions below :

(1) Marketing: The marketing team wants to launch some campaigns, and they need your help with the following:-

Perform Analysis:
Marketing

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

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

Query 1

Limit to 1000 rows

```
1  SELECT
2      id, username, created_at
3  FROM
4      users
5  ORDER BY created_at ASC
6  LIMIT 5;
7
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rc

id	username	created_at
95	Nicole71	2016-05-09 17:30:22
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
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

users 6

Output

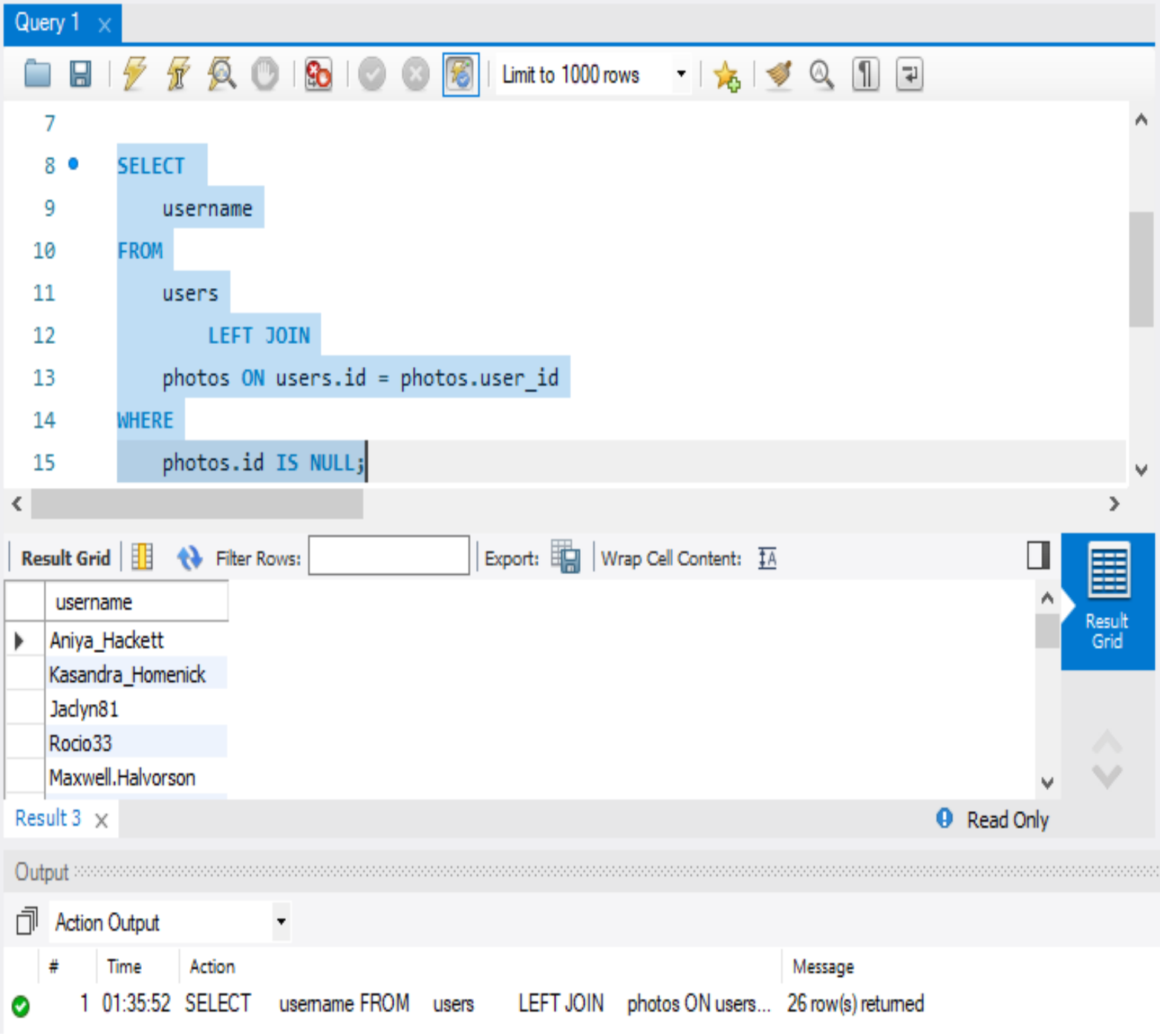
Action Output

#	Time	Action	Message
1	02:39:36	SELECT id, username, created_at FROM users ORDER BY created_at ASC LIMIT 5	5 row(s) returned

Perform Analysis:
Marketing

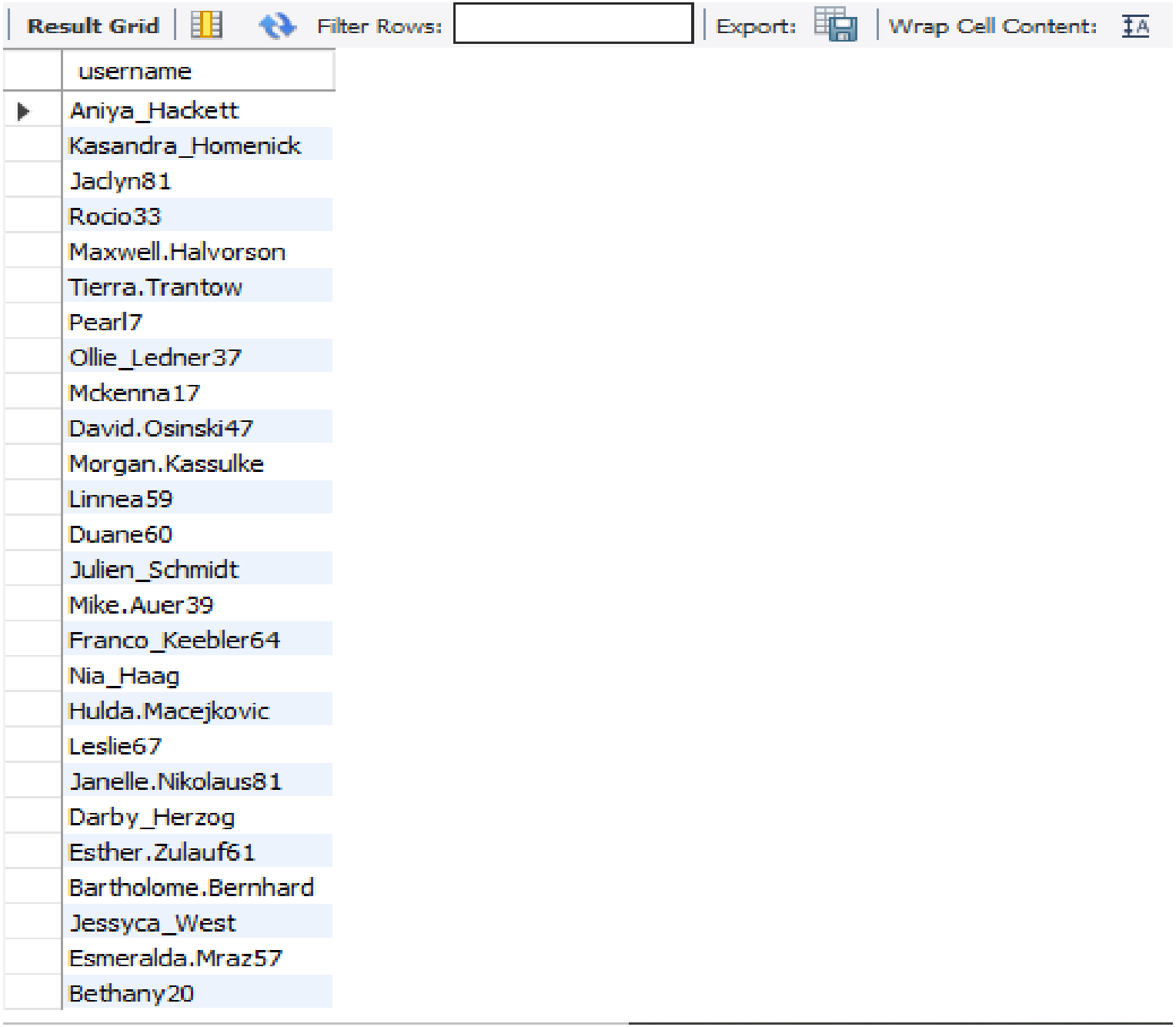
Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.

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



Complete User Name List:

Remind Inactive Users to Start Posting



username
Aniya_Hackett
Kasandra_Homenick
Jaclyn81
Rocio33
Maxwell.Halvorson
Tierra.Trantow
Pearl7
Ollie_Ledner37
Mckenna17
David.Osinski47
Morgan.Kassulke
Linnea59
Duane60
Julien_Schmidt
Mike.Auer39
Franco_Keebler64
Nia_Haag
Hulda.Macejkovic
Leslie67
Janelle.Nikolaus81
Darby_Herzog
Esther.Zulauf61
Bartholome.Bernhard
Jessyca_West
Esmeralda.Mraz57
Bethany20

Perform Analysis:
Marketing

Declaring Contest Winner: Th e 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.

Task: Identify the winner of the contest and provide their details

Query 1

Limit to 1000 rows

```
16  SELECT
17      users.username,
18      photos.id,
19      photos.image_url,
20      COUNT(*) AS total_likes
21  FROM
22      likes
23          JOIN
24      photos ON photos.id = likes.photo_id
25          JOIN
26      users ON users.id = likes.photo_id
27  GROUP BY photos.id
28  ORDER BY total_likes DESC
29  LIMIT 1;
```

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

username	id	image_url	total_likes
Kaley9	30	http://kenny.com	41

Result 8

Output

Action Output

#	Time	Action	Message
1	02:56:41	SELECT users.username, photos.id, photos.image_url, COUNT(*) AS total_likes FRO...	1 row(s) returned

Perform Analysis:

Marketing

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

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

Query 1

Limit to 1000 rows

```
30
31 •  SELECT
32        tag_name, COUNT(tag_name) AS total
33    FROM
34        tags
35            JOIN
36        photo_tags ON tags.id = photo_tags.tag_id
37    GROUP BY tags.id
38    ORDER BY total DESC
39    LIMIT 5;
```

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

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

Result 9

Output

Action Output

#	Time	Action	Message
1	03:13:31	SELECT tag_name, COUNT(tag_name) AS total FROM tags JOIN photo_tags ON t...	5 row(s) returned

Perform Analysis:

Marketing

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

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

Query 1

Limit to 1000 rows

```
40
41 •  SELECT
42        DATE_FORMAT(created_at, '%W') AS 'Week Day',
43        COUNT(*) AS 'Number of Registration'
44    FROM
45        users
46    GROUP BY 1
47    ORDER BY 2 DESC;
48
49
```

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

Week Day	Number of Registration
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Result 13

Output

Action Output

#	Time	Action		Message
1	03:25:26	SELECT	DATE_FORMAT(created_at, '%W') AS 'Week Day',	COUNT(*) AS 'Number of Re... 7 row(s) returned

(2) Investor Metrics: Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds

Perform Analysis: Investor Metrics:

```
74 •  SELECT ROUND((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users),2);
```

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

ROUND((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users),2)
2.57

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

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

Result 21

Output

Action Output

#	Time	Action	Message
1	03:46:47	SELECT ROUND((SELECT COUNT(*)FROM photos)/(SELECT COUNT(*) FROM users),2) LIMI...	1 row(s) returned

Perform Analysis: Investor Metrics:

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

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).

```
62  SELECT
63      users.id, username, COUNT(users.id) AS total_likes_by_user
64  FROM
65      users
66          JOIN
67      likes ON users.id = likes.user_id
68  GROUP BY users.id
69  HAVING total_likes_by_user = (SELECT
70          COUNT(*)
71      FROM
72          photos);
73
```

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

id	username	total_likes_by_user
5	Aniya_Hackett	257
14	Jaclyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257

Result 23

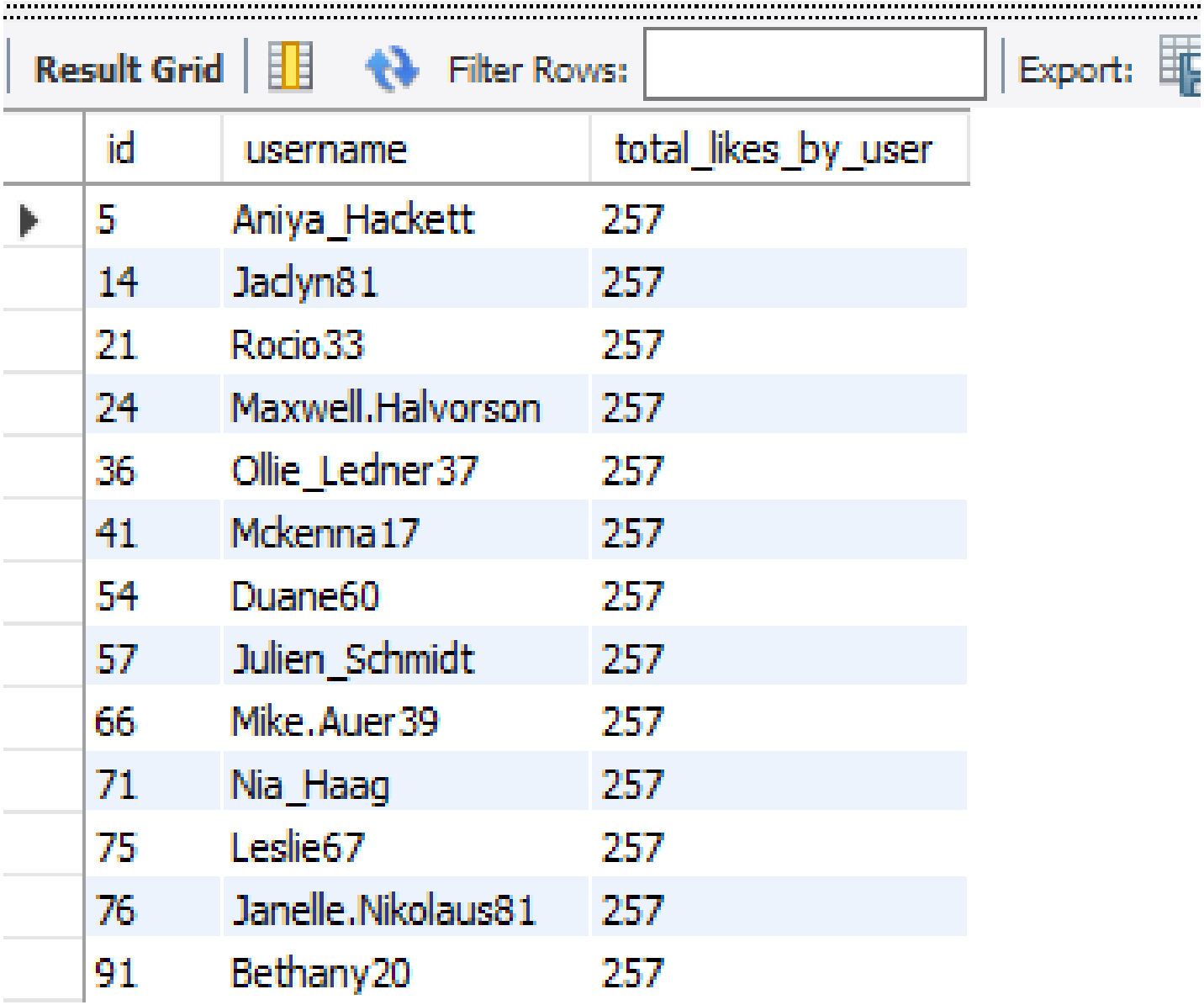
Output

Action Output

#	Time	Action	Message
1	03:56:29	SELECT users.id, username, COUNT(users.id) AS total_likes_by_user FROM users JO...	13 row(s) returned

Complete User Name List:

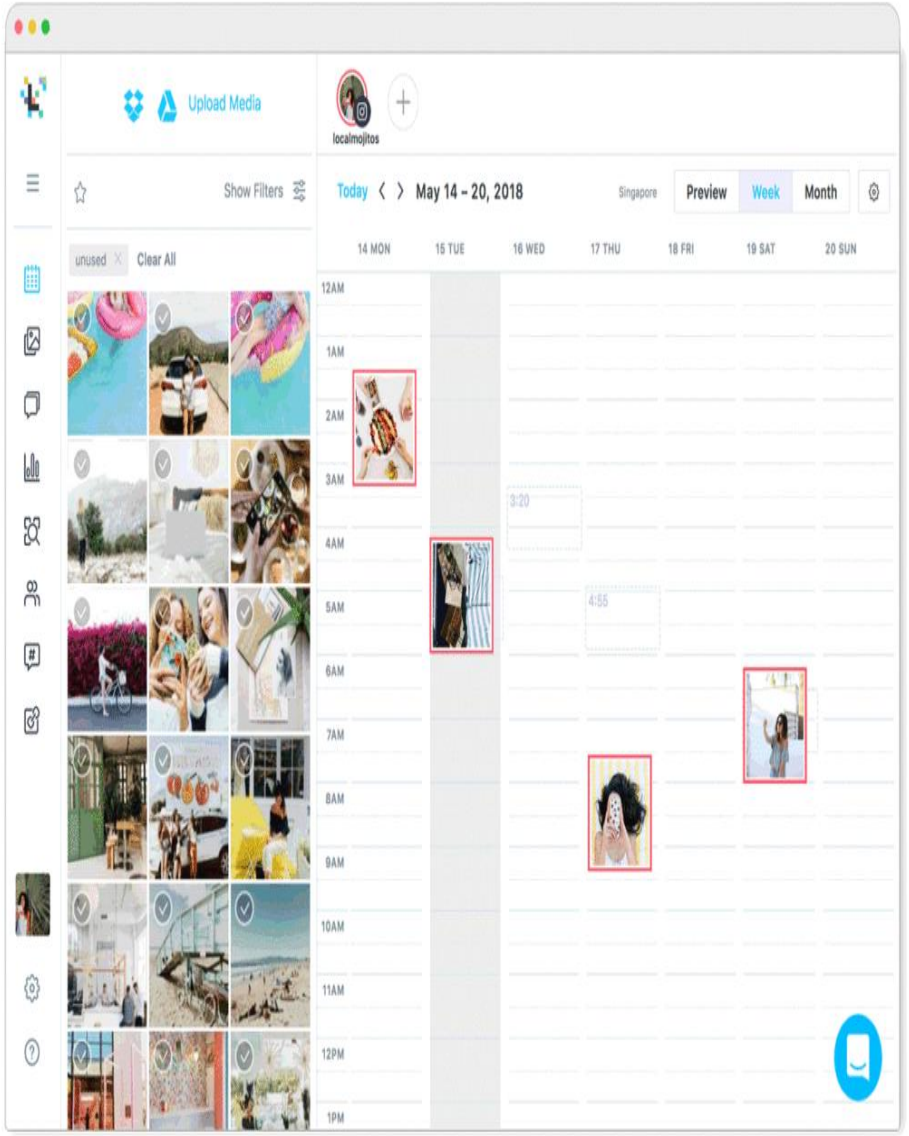
Bots & Fake Accounts



id	username	total_likes_by_user
5	Aniya_Hackett	257
14	Jadyn81	257
21	Rocio33	257
24	Maxwell.Halvorson	257
36	Ollie_Ledner37	257
41	Mckenna17	257
54	Duane60	257
57	Julien_Schmidt	257
66	Mike.Auer39	257
71	Nia_Haag	257
75	Leslie67	257
76	Janelle.Nikolaus81	257
91	Bethany20	257

Result

- I have gained knowledge of various SQL functions which helped me to solve the questions asked in this project. Following are the functions I used: Sorting functions- TOP, , GROUP BY, HAVING, ORDER BY, JOIN,LEFT JOIN
- Aggregate function- COUNT



THANKS