SQL_ANALYSIS

PLACEMENT PREPARATION

Using the given dataset, I created the tables in MySQL workbench as follows

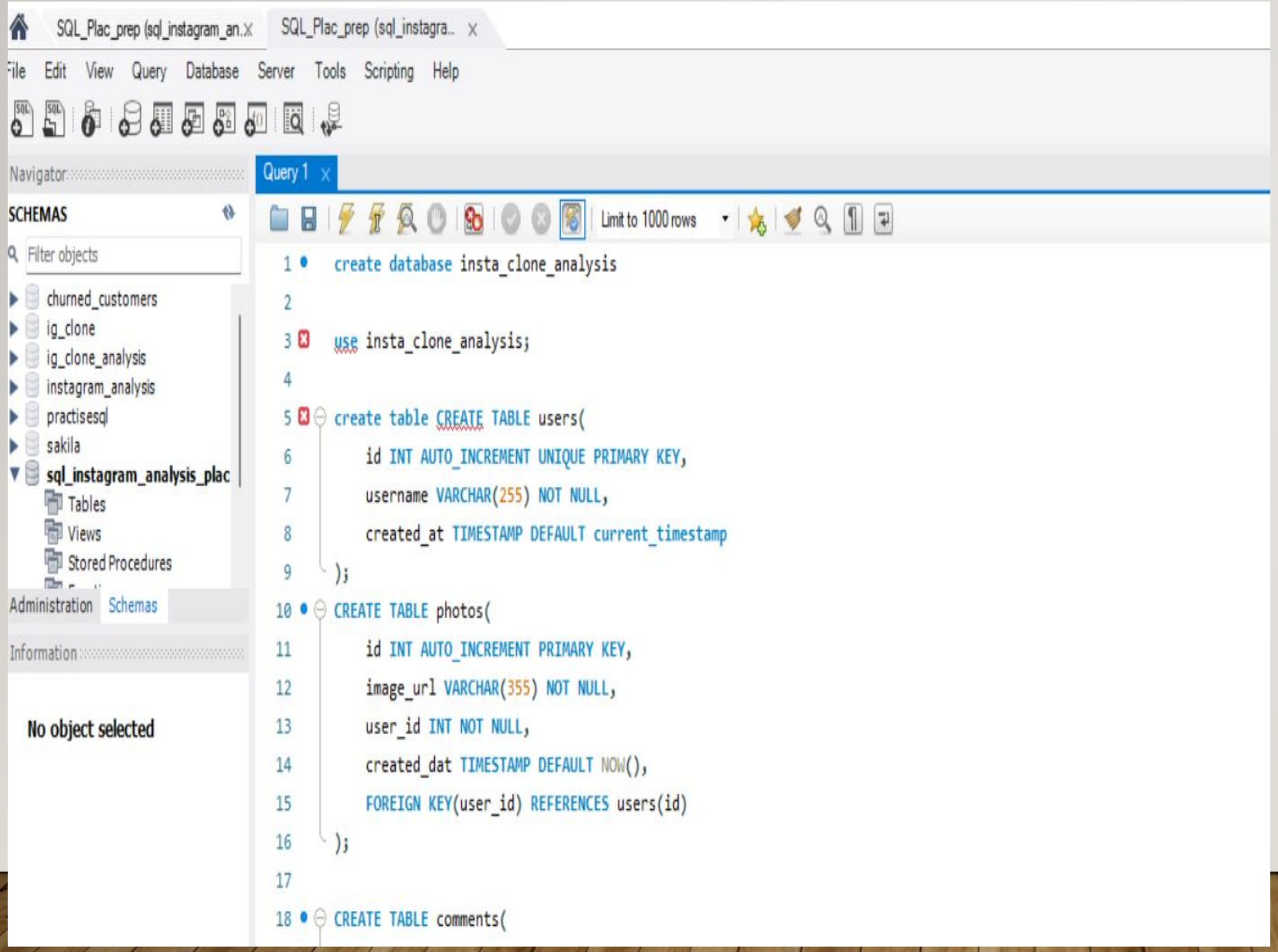
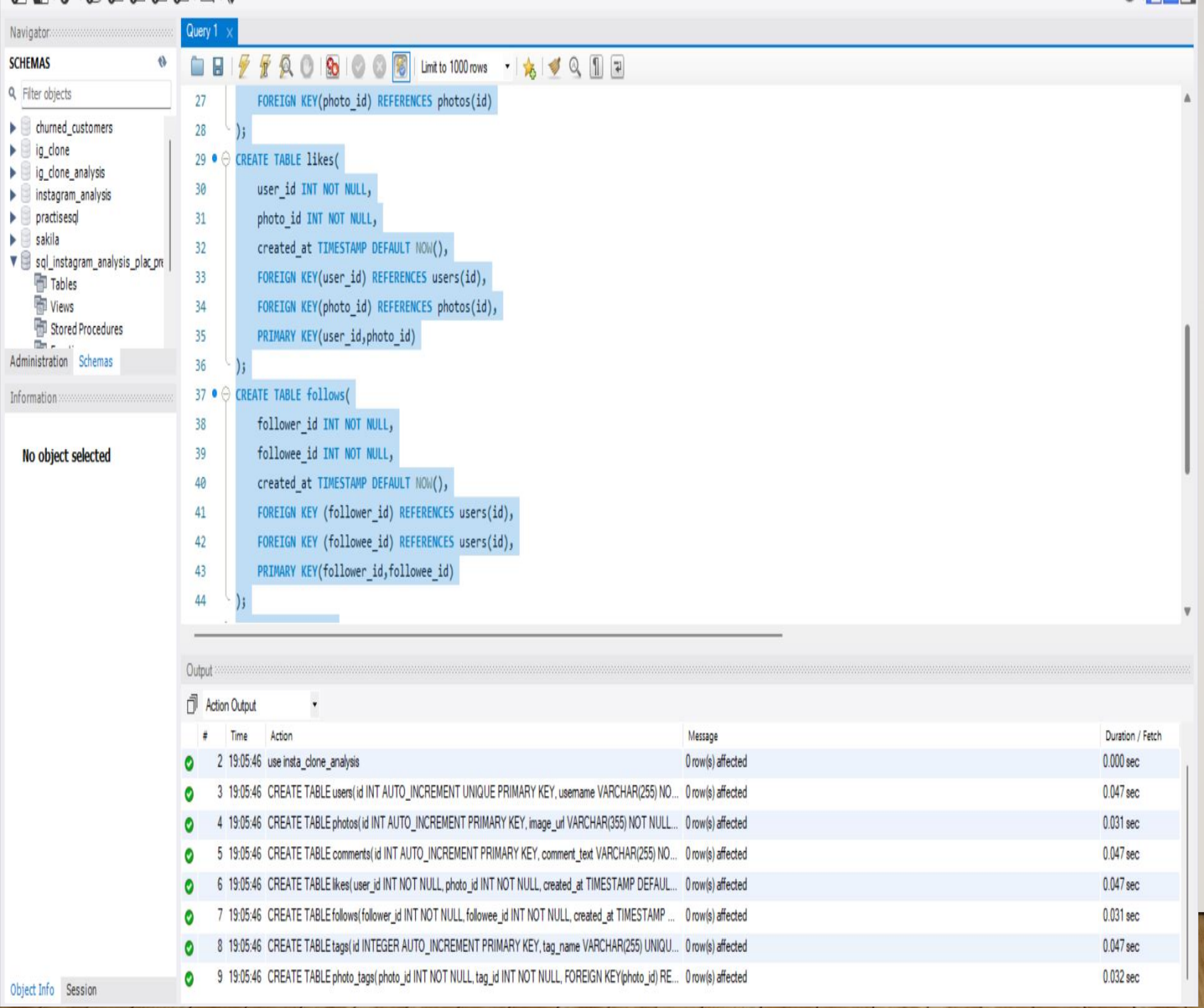
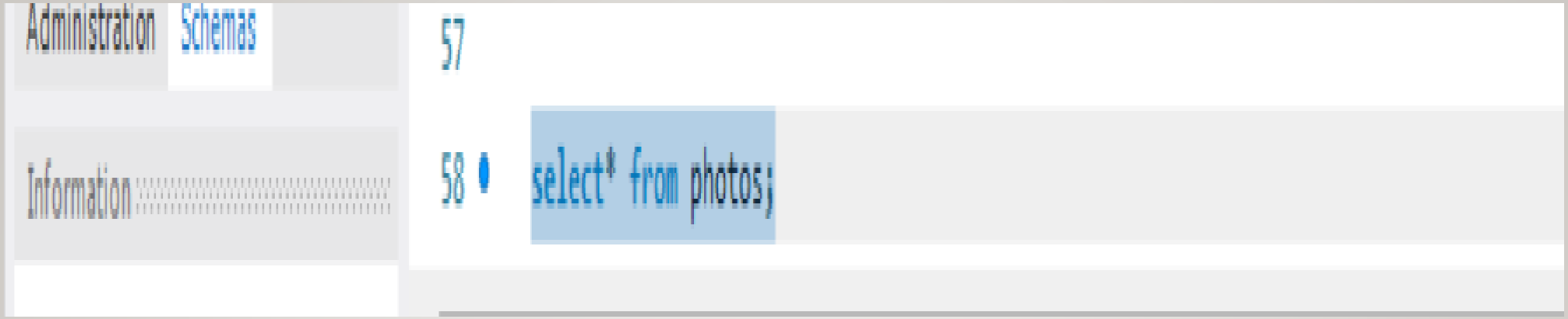


TABLE NAMES

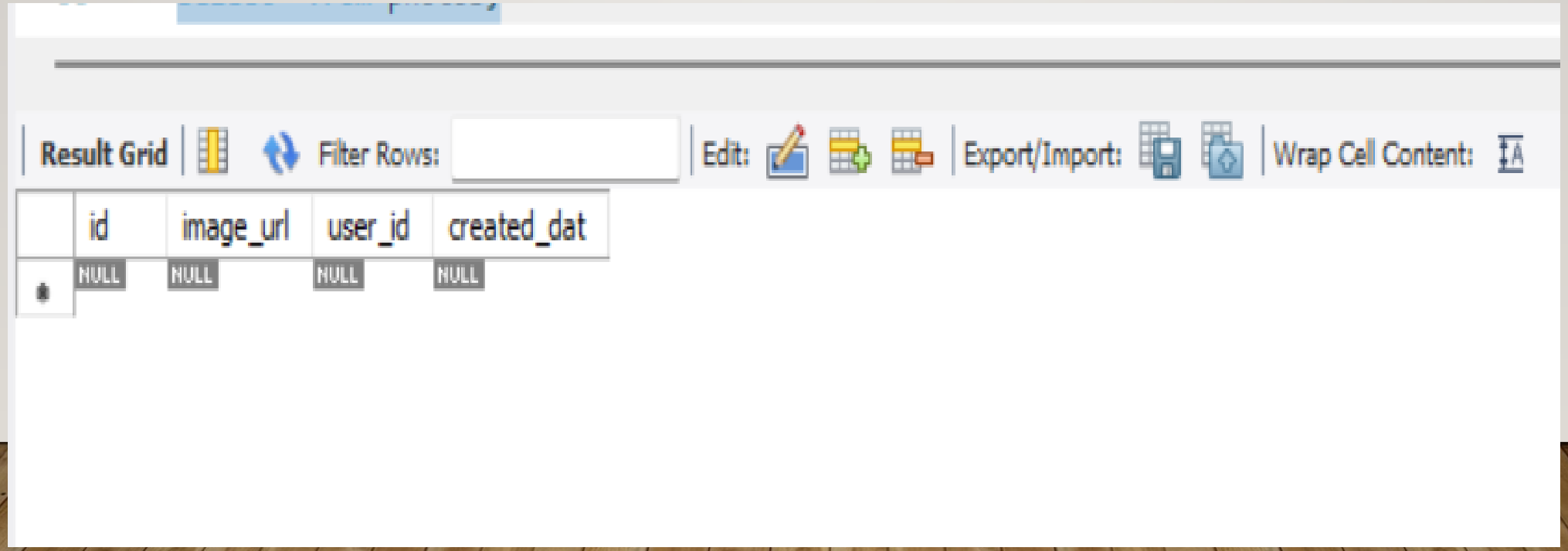
- USERS
- PHOTOS
- LIKES
- COMMENTS
- TAGS
- FOLLOWS
- PHOTO_TAGS

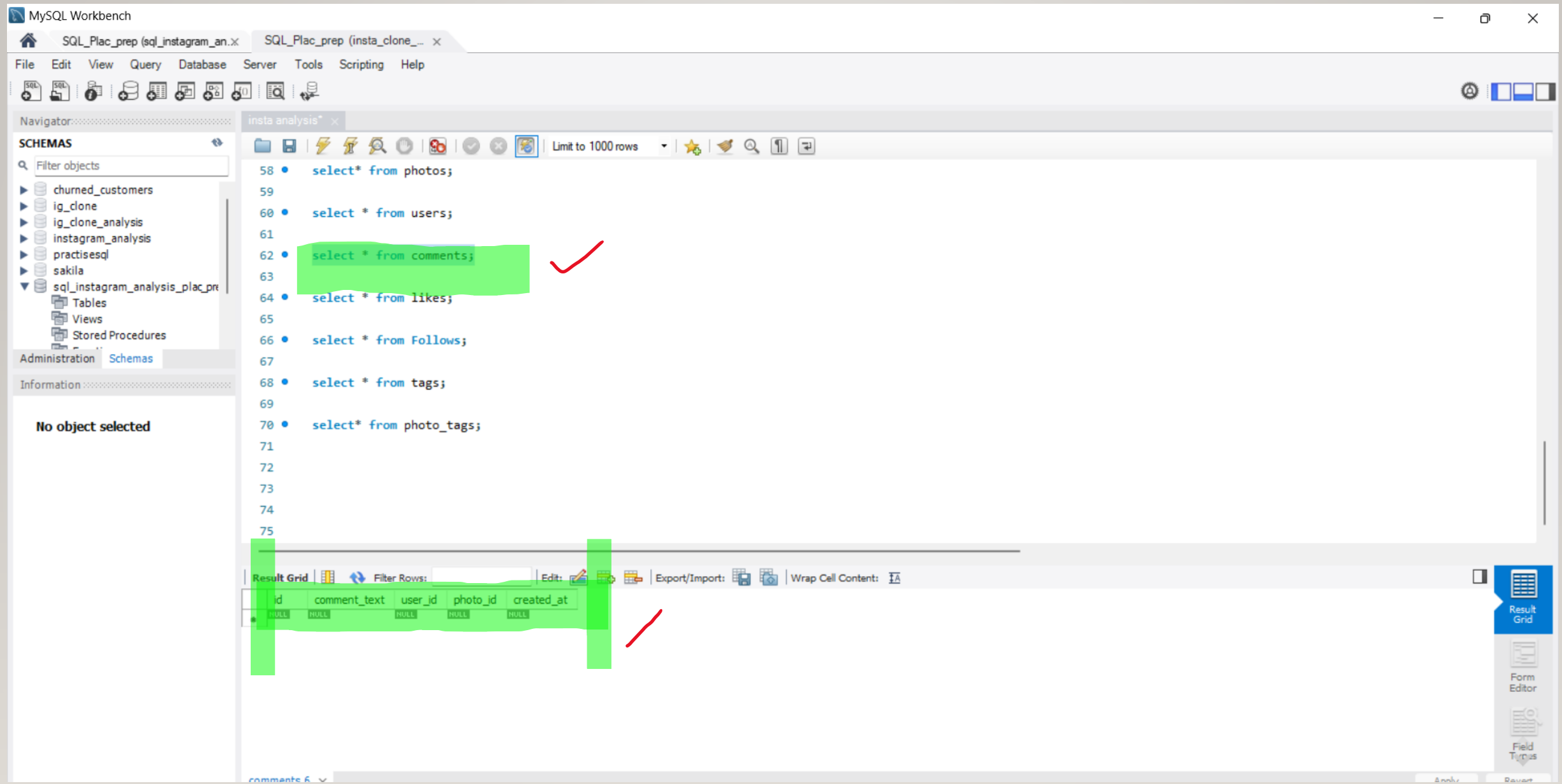


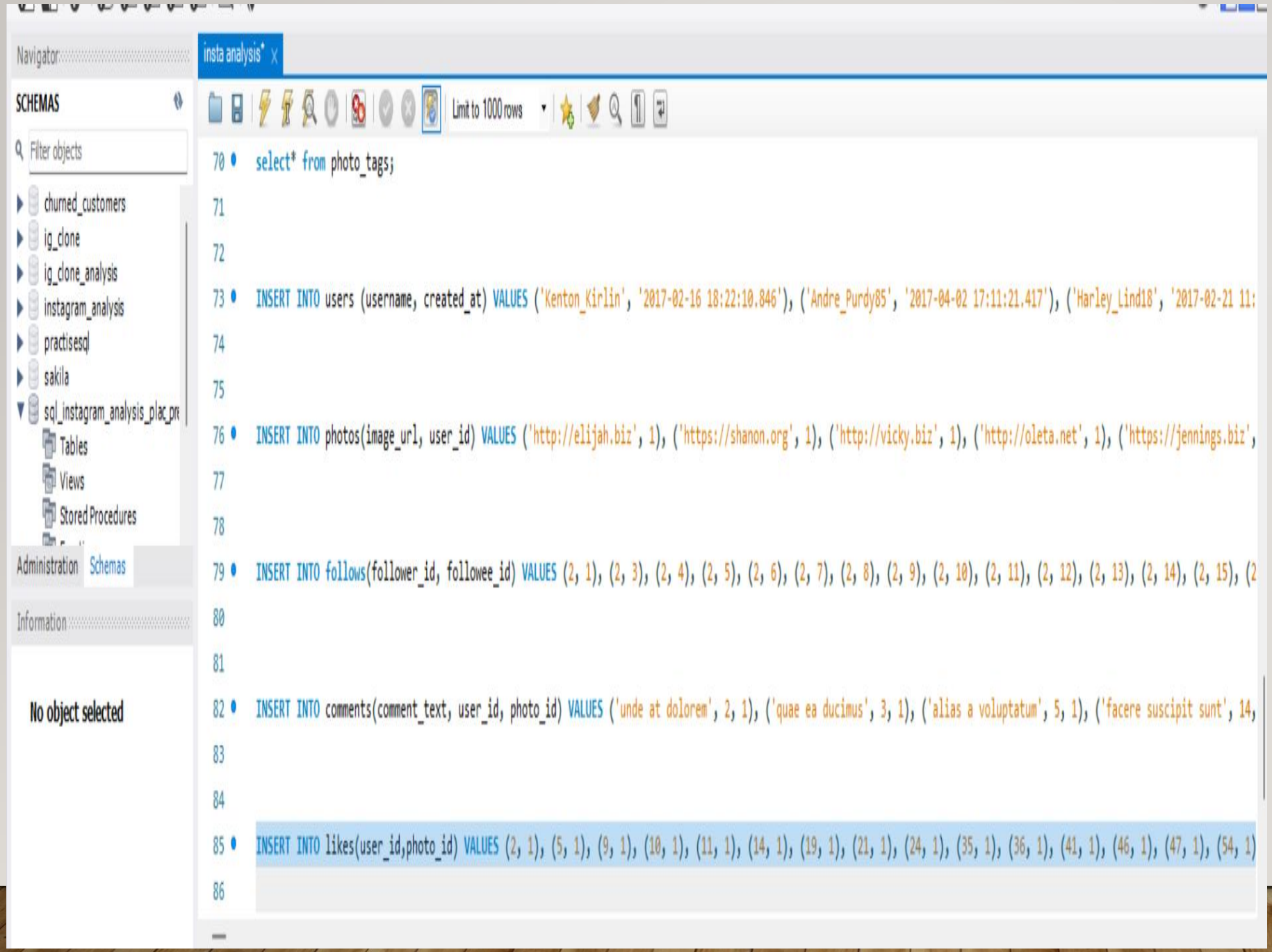
OUTPUTS AFTER TABLE CREATION WITHOUT ERROR



Photos table with columns we input, before inserting values into it.




MySQL Workbench
SQL_Plac_prep (sql_instagram_an...
SQL_Plac_prep (insta_clone_...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
churned_customers
ig_clone
ig_clone_analysis
instagram_analysis
practisesql
sakila
sql_instagram_analysis_plac_pre
Tables
Views
Stored Procedures
Administration Schemas
Information
No object selected
insta analysis*
Limit to 1000 rows
58 select* from photos;
59
60 select * from users;
61
62 select * from comments;
63
64 select * from likes;
65
66 select * from Follows;
67
68 select * from tags;
69
70 select* from photo_tags;
71
72
73
74
75
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id
comment_text
user_id
photo_id
created_at
NULL
Result Grid
Form Editor
Field Types
comments 6



INSERTING VALUES INTO TABLES USERS, PHOTOS, COMMENTS, LIKES, FOLLOWS, TAGS, PHOTO_TAGS.

Result Grid | Filter Rows: | Edit:

id	image_url	user_id	created_dat
1	http://elijah.biz	1	2024-06-21 20:40:13
2	https://shanon.org	1	2024-06-21 20:40:13
3	http://vicky.biz	1	2024-06-21 20:40:13
4	http://oleta.net	1	2024-06-21 20:40:13
5	https://jennings.biz	1	2024-06-21 20:40:13
6	https://quinn.biz	2	2024-06-21 20:40:13
7	https://selina.name	2	2024-06-21 20:40:13
8	http://malvina.org	2	2024-06-21 20:40:13
9	https://branson.biz	2	2024-06-21 20:40:13

Result Grid | Filter Rows: | Edit: | Export/Im

id	comment_text	user_id	photo_id	created_at
1	unde at dolorem	2	1	2024-06-21 20:40:33
2	quae ea ducimus	3	1	2024-06-21 20:40:33
3	alias a voluptatum	5	1	2024-06-21 20:40:33
4	facere suscipit sunt	14	1	2024-06-21 20:40:33
5	totam eligendi quaerat	17	1	2024-06-21 20:40:33
6	vitae quia aliquam	21	1	2024-06-21 20:40:33
7	exercitationem occaecati neque	24	1	2024-06-21 20:40:33
8	sint ad fugiat	31	1	2024-06-21 20:40:33
9	nesciunt aut nesciunt	36	1	2024-06-21 20:40:33
10	laudantium ut nostrum	41	1	2024-06-21 20:40:33

Result Grid | Filter

photo_id	tag_id
14	1
21	1
45	1
75	1
83	1
85	1
91	1
118	1
149	1
194	1

photo_tags 17

esult Grid | Filter Rows:

user_id	photo_id	created_at
2	1	2024-06-21 20:40:42
2	4	2024-06-21 20:40:42
2	8	2024-06-21 20:40:42
2	9	2024-06-21 20:40:42
2	10	2024-06-21 20:40:42
2	11	2024-06-21 20:40:42
2	12	2024-06-21 20:40:42
2	13	2024-06-21 20:40:42
2	15	2024-06-21 20:40:42
2	23	2024-06-21 20:40:42

es 14 ×

utput

Result Grid | Filter Rows:

follower_id	followee_id	created_at
2	1	2024-06-21 20:40:17
2	3	2024-06-21 20:40:17
2	4	2024-06-21 20:40:17
2	5	2024-06-21 20:40:17
2	6	2024-06-21 20:40:17
2	7	2024-06-21 20:40:17
2	8	2024-06-21 20:40:17
2	9	2024-06-21 20:40:17
2	10	2024-06-21 20:40:17
2	11	2024-06-21 20:40:17

Follows 15 ×

Output

Result Grid | Filter Rows: | Edi

id	tag_name	created_at
1	sunset	2024-06-21 20:40:58
2	photography	2024-06-21 20:40:58
3	sunrise	2024-06-21 20:40:58
4	landscape	2024-06-21 20:40:58
5	food	2024-06-21 20:40:58
6	foodie	2024-06-21 20:40:58
7	delicious	2024-06-21 20:40:58
8	beauty	2024-06-21 20:40:58
9	stunning	2024-06-21 20:40:58
10	dreamy	2024-06-21 20:40:58

MARKETING ANALYSIS

LOYAL USERS:

Top five oldest users.

```
12
13 •    select*from users;
14
15 •    SELECT * FROM users
16      ORDER BY created_at ASC
17      LIMIT 5;
18
```

Result Grid | Filter Rows:

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

INACTIVE USERS:

Users who have never posted a single photo.

```
19      ---- B.INACTIVE_USERS ------
20
21  •   select* from photos;
22
23  •   SELECT * FROM users u
24      LEFT JOIN photos p ON u.id = p.user_id
25      WHERE p.id IS NULL;
26
27
```

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

id	username	created_at	id	image_url	user_id	created_dat
5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
7	Kasandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
14	Jadyn81	2017-02-06 23:29:16	NULL	NULL	NULL	NULL
21	Rocio33	2017-01-23 11:51:15	NULL	NULL	NULL	NULL
24	Maxwell.Halvorson	2017-04-18 02:32:44	NULL	NULL	NULL	NULL
25	Tierra.Trantow	2016-10-03 12:49:21	NULL	NULL	NULL	NULL
34	Pearl7	2016-07-08 21:42:01	NULL	NULL	NULL	NULL
36	Ollie_Ledner37	2016-08-04 15:42:20	NULL	NULL	NULL	NULL
41	Mckenna17	2016-07-17 17:25:45	NULL	NULL	NULL	NULL
45	David.Osinski47	2017-02-05 21:23:37	NULL	NULL	NULL	NULL
49	Morgan.Kassulke	2016-10-30 12:42:31	NULL	NULL	NULL	NULL
53	Linnea59	2017-02-07 07:49:34	NULL	NULL	NULL	NULL
54	Duane60	2016-12-21 04:43:38	NULL	NULL	NULL	NULL
57	Julien_Schmidt	2017-02-02 23:12:48	NULL	NULL	NULL	NULL
66	Mike.Auer39	2016-07-01 17:36:15	NULL	NULL	NULL	NULL
68	Franco_Keebler64	2016-11-13 20:09:27	NULL	NULL	NULL	NULL

Contest winner declaration:

Users with the most likes in a single photo.

```sql
SELECT p.id, u.id, u.username, like_counts.like_count
FROM (
    SELECT photo_id, COUNT(user_id) AS like_count
    FROM likes
    GROUP BY photo_id
    ORDER BY like_count DESC
    LIMIT 1
) AS like_counts
JOIN photos p ON p.id = like_counts.photo_id
JOIN users u ON p.user_id = u.id;
```

Result Grid | Filter Rows:

id	id	username	like_count
145	52	Zack_Kemmer93	48

MOST POPULAR HASHTAG:

Top five HASHTAGS.

```
63
64 •    SELECT t.id, t.tag_name, tag_counts.tag_count
65      FROM (
66          SELECT tag_id, COUNT(photo_id) AS tag_count
67          FROM photo_tags
68          GROUP BY tag_id
69      ) AS tag_counts
70      JOIN tags t ON t.id = tag_counts.tag_id
71      ORDER BY tag_counts.tag_count DESC
72      LIMIT 5;
73
```

Result Grid | Filter Rows: | Export: | Wrap C

id	tag_name	tag_count
21	smile	59
20	beach	42
17	party	39
13	fun	38
5	food	24

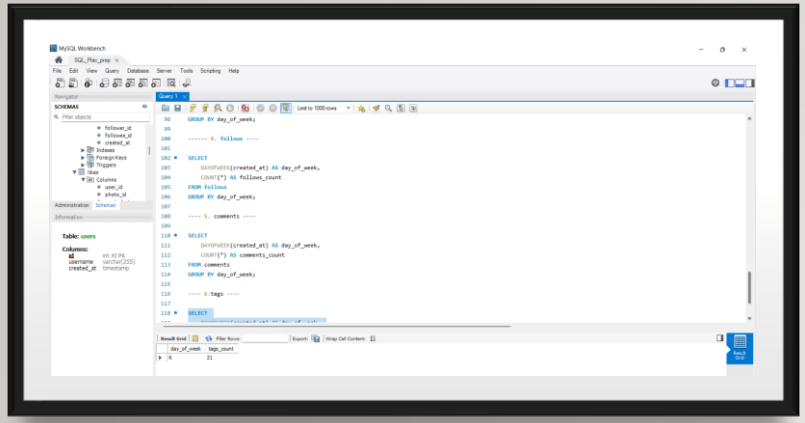
```
54 •    select* from tags;
55
56 •    SELECT tag_name, COUNT(*) AS tag_count
57      FROM tags
58      GROUP BY tag_name
59      ORDER BY tag_count DESC
60      LIMIT 5;
61
62
63
```

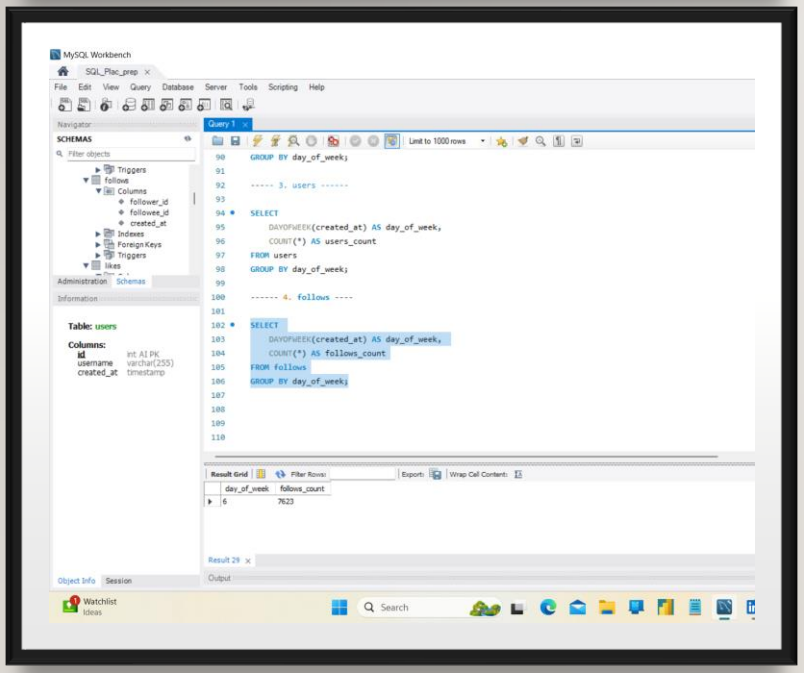
Result Grid | Filter Rows: | Export:

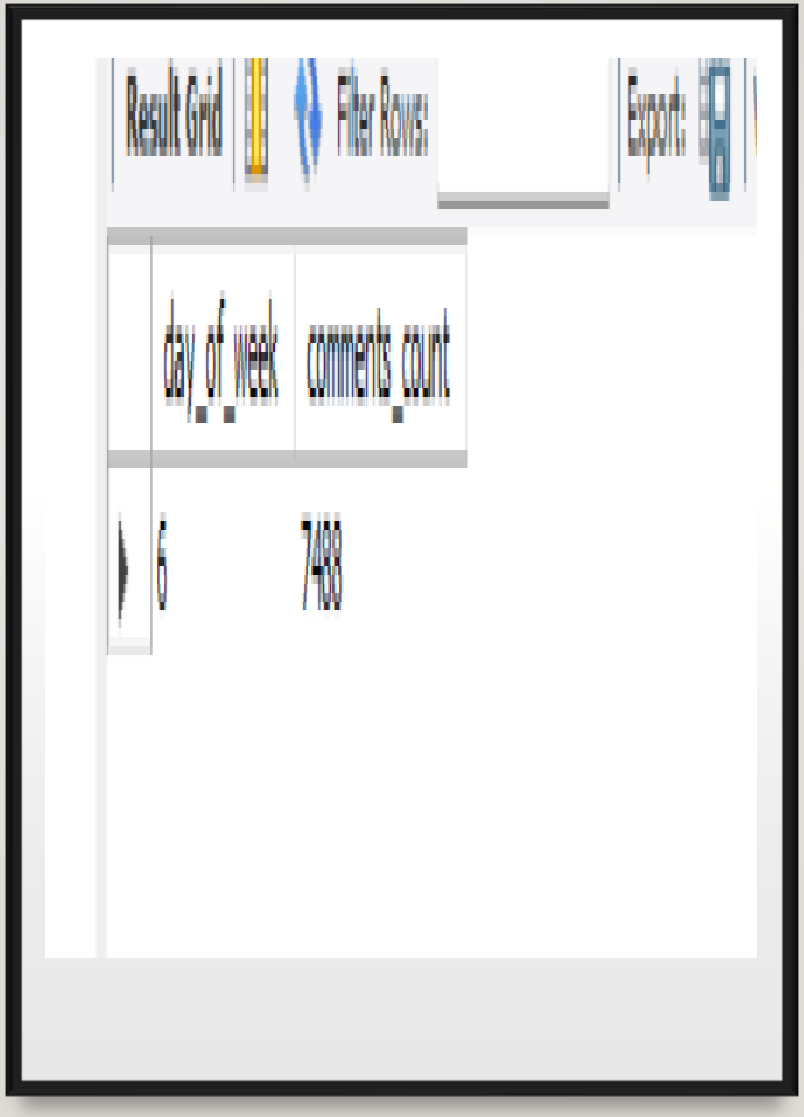
tag_name	tag_count
beach	1
beauty	1
concert	1
delicious	1
dreamy	1

Result 23 ×

Output







AD CAMPAIGN LAUNCH

IT'S THE BEST DAY OF THE WEEK TO LAUNCH ADS.

Server Tools Scripting Help

Query 1

Limit to 1000 rows

```sql
SELECT
    day_of_week,
    SUM(likes_count) AS total_likes,
    SUM(photos_count) AS total_photos,
    SUM(comments_count) AS total_comments,
    SUM(follows_count) AS total_follows,
    SUM(tags_count) AS total_tags,
    (SUM(likes_count) + SUM(photos_count) + SUM(comments_count) + SUM(follows_count) + SUM(tags_count)) AS total_engagement
FROM (
    SELECT
        DAYOFWEEK(created_at) AS day_of_week,
        COUNT(*) AS likes_count,
        0 AS photos_count,
        0 AS comments_count,
        0 AS follows_count
```

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

day_of_week	total_likes	total_photos	total_comments	total_follows	total_tags	total_engagement
6	8782	257	7488	7623	21	24171

Result 32

Output

Search

- By the above analysis the 6th day of the week has the most number of engagements.
- So the 6th day of the week (FRIDAY) will be the best day to launch ads.

INVESTOR METRICS

- USER ENGAGEMENT:
- Average no. of posts per user on Instagram.

```
181            COUNT(*) AS tags_count
182        FROM tags
183        GROUP BY day_of_week
184    ) AS combined_counts
185    GROUP BY day_of_week
186    ORDER BY total_engagement DESC;
187
188    ----- INVESTOR METRICS ----
189
190    A. --- USER ENGAGEMENT ----
191
192    --- average no. of posts ----
193
194    SELECT COUNT(*) AS total_photos
```

Result Grid | Filter Rows: | Export: | Wrap Ce

total_photos
257

MySQL Workbench

SQL_Plac_prep

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- follower_id
- followee_id
- created_at
- Indexes
- Foreign Keys
- Triggers
- likes
  - Columns
    - user_id
    - photo_id

Administration Schemas

Information

Table: users

Columns:

id	int AI PK
username	varchar(255)
created_at	timestamp

Object Info Session

Query 1

Limit to 1000 rows

```
182     FROM tags
183     GROUP BY day_of_week
184 ) AS combined_counts
185 GROUP BY day_of_week
186 ORDER BY total_engagement DESC;
187
188 ----- INVESTOR METRICS ----
189
190 --- A. USER ENGAGEMENT ----
191
192 --- average no. of posts ----
193
194 SELECT COUNT(*) AS total_photos
195 FROM photos;
196
197 SELECT COUNT(*) AS total_users
198 FROM users;
199
200 SELECT
201     (SELECT COUNT(*) FROM photos) / (SELECT COUNT(*) FROM users) AS average_posts_per_user;
202
```

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

erage_posts_per_user
700

Result Grid

Result 35

Read Only

Output

JPY/INR -0.62%

Search

ENG IN

10:38 22-06-2024

BOTS AND FAKE ACCOUNTS:

Users who have liked every photo on the site as this is not typically possible for normal users.

```
202
203         ---- B. BOTS AND FAKE ACCOUNTS ---
204
205 •   SELECT u.id, u.username
206     FROM (
207         SELECT user_id, COUNT(photo_id) AS like_count
208         FROM likes
209         GROUP BY user_id
210     ) AS user_likes
211     JOIN users u ON u.id = user_likes.user_id
212     WHERE user_likes.like_count = (SELECT COUNT(*) FROM photos);
213
214
```

Result Grid | Filter Rows:

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

THANK YOU