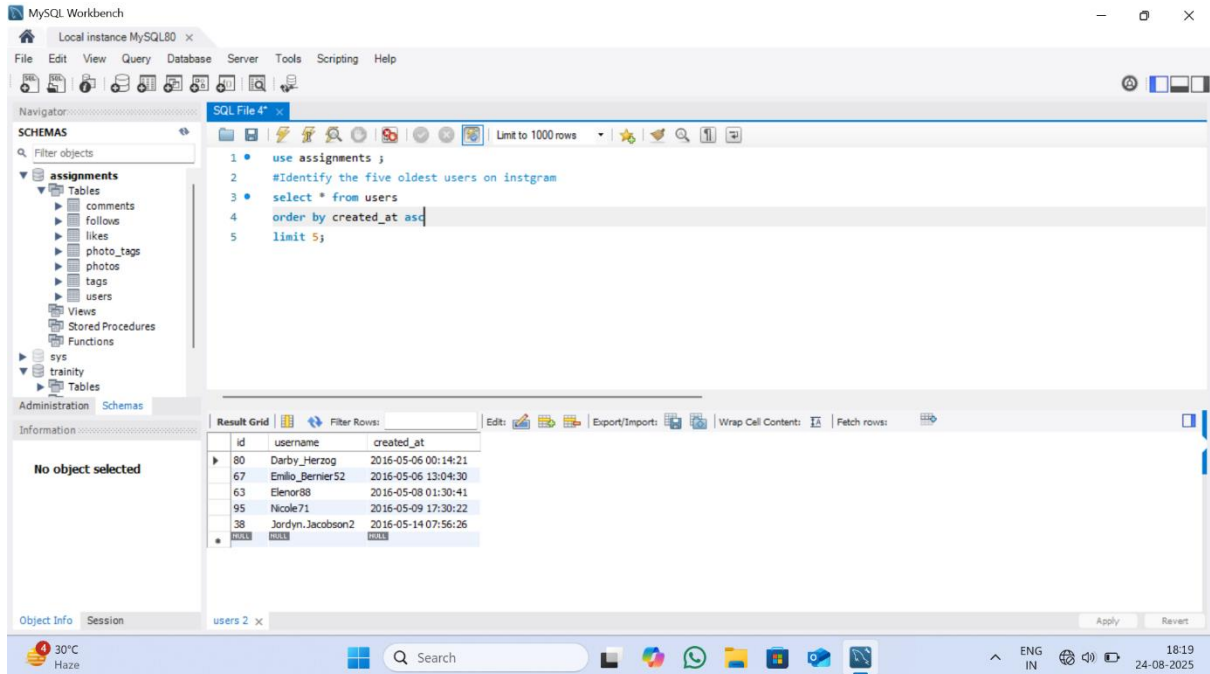


Instagram User Analytic

1. Identify the five oldest users on Instagram from the provide database.



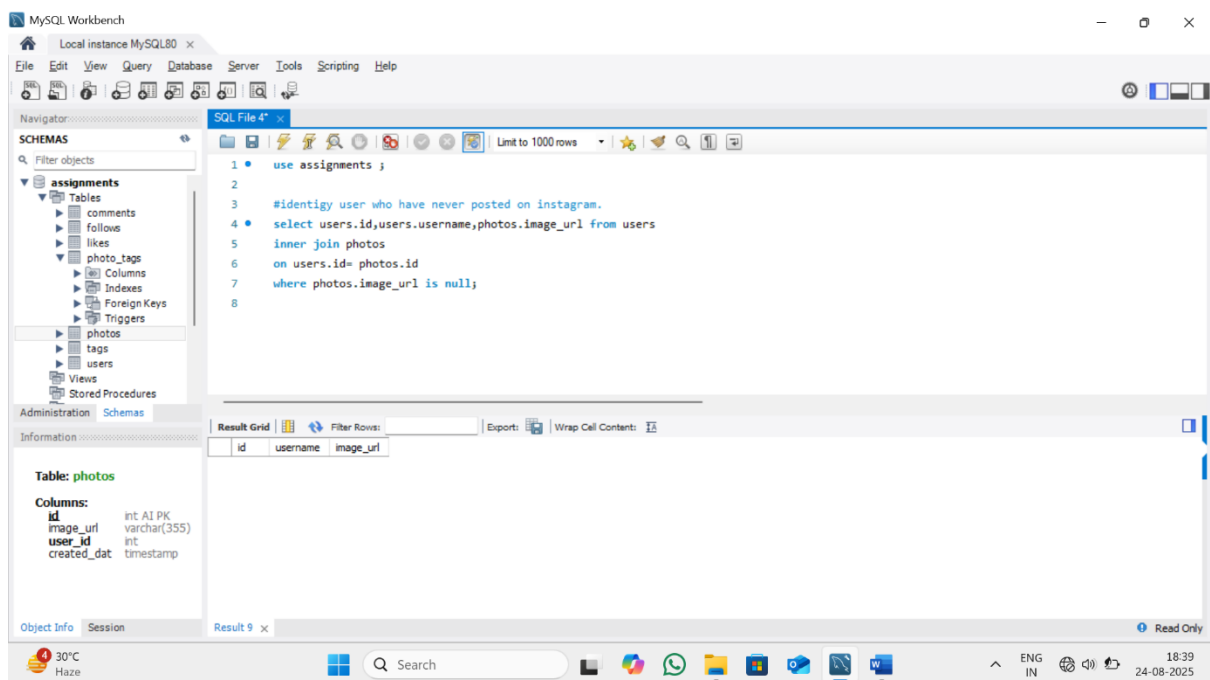
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use assignments ;
2 #Identify the five oldest users on instagram
3 select * from users
4 order by created_at asc
5 limit 5;
```

The Result Grid shows the following data:

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

2. Identify users who have never posted a single photo on Instagram.



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use assignments ;
2
3 #identigy user who have never posted on instagram.
4 select users.id,users.username,photos.image_url from users
5 inner join photos
6 on users.id= photos.id
7 where photos.image_url is null;
8
```

The Result Grid shows the following data:

id	username	image_url
----	----------	-----------

The Table: photos section shows the following columns:

- id: int AI PK
- image_url: varchar(355)
- user_id: int
- created_at: timestamp

3. Determine the winner of the contest and provide their details to the team.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'assignments' expanded, showing tables like 'comments', 'follows', 'likes', 'photo_tags', and 'photos'. The main editor contains a SQL query:

```
1 use assignments ;
2 # The winner and her details .
3
4 select users.id,users.username,photos.user_id ,count(photo_id) as total_likes from users
5 inner join photos
6 on users.id=photos.user_id
7 inner join likes
8 on photos.user_id=likes.user_id
9 group by photos.user_id,users.id
10 order by total_likes desc
11 limit 1;
```

The 'Result Grid' at the bottom shows the following data:

id	username	user_id	total_likes
65	Adelle96	65	480

The 'Information' pane on the left shows details for the 'photos' table, including columns: id, image_url, user_id, and created_at.

4. Identify and suggest the top five most commonly used hashtags on the platform

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'assignments' expanded, showing tables like 'comments', 'follows', 'likes', 'photo_tags', and 'photos'. The main editor contains a SQL query:

```
1 use assignments ;
2 #Popular hashtags
3
4 select tags.id, tag_name,count(tag_name) as most_used from tags
5 inner join photo_tags
6 on tags.id = photo_tags.tag_id
7 group by tag_id
8 order by most_used desc
9 limit 5;
```

The 'Result Grid' at the bottom shows the following data:

id	tag_name	most_used
21	smile	59
20	beach	42
17	party	39
13	fun	38
18	concert	24

The 'Information' pane on the left shows details for the 'tags' table, including columns: id, tag_name, and created_at.

5. Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use assignments ;
2 #account creation
3
4 select dayname(created_at) as Day_created,
5 count(*) as total_acc from users
6 group by Day_created
7 order by total_acc desc
```

The result grid shows the following data:

Day_created	total_acc
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

The left sidebar shows the database schema with tables: comments, follows, likes, photo_tags, photos, tags, users, sys, trainity, and tables. The bottom status bar shows the temperature as 30°C and the date as 24-08-2025.

6. Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 use assignments ;
2 #avg count per person
3
4 select avg(avg_photo_count) as avg_count from(
5 select users.id,users.username ,count(photo_id) as avg_photo_count from photo_tags
6 inner join users
7 on users.id = photo_tags.photo_id
8 group by id
9 order by avg_photo_count desc) as final_count
10
11
```

The result grid shows the following data:

avg_count
2.6667

The left sidebar shows the database schema with tables: comments, follows, likes, photo_tags, photos, tags, users, sys, trainity, and tables. The bottom status bar shows the temperature as 30°C and the date as 24-08-2025.

7. Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with the 'assignments' database selected. The 'likes' table is highlighted under the 'Tables' folder. The main editor window shows a SQL query in 'SQL File 4'.

```
1 use assignments ;
2 # person liked every post
3
4 SELECT users.id, users.username
5 FROM users
6 inner join likes ON users.id = likes.user_id
7 GROUP BY users.id, users.username
8 HAVING COUNT(DISTINCT likes.photo_id) = (SELECT COUNT(*) FROM photos);
9
```

The 'Result Grid' at the bottom displays the results of the query, showing a list of users who have liked every photo. The table has two columns: 'id' and 'username'.

id	username
5	Aniya_Hackett
14	Jedyn81
21	Roco33
24	Maxwell.Halvorson
36	Olle_Ledner37
41	Mckenna17
54	Duane60
57	Julien_Schmidt
66	Mike_Auer39
71	Nia_Haag

The bottom status bar shows the system tray with the date and time: 22:34, 24-08-2025.