

PROJECT 2: INSTAGRAM USER ANALYTICS

INTRODUCTION

This project aim to extract useful insights from raw data using MYSQL workbench and the findings could potentially influence the future development of performs and thereby increase your knowledge and deepen your understanding of MYSQL

(A)MARKETING ANALYSIS:

1. Loyal User Reward:

Task: Identify the five oldest users on Instagram from the provided database.

SQL QUERY: select * from users

 username

 order by created_at

 limit 5;

Result: The five oldest users selected for loyal user Reward are follow:

Sl No.	id	username	Created_at
1	80	Darby_Herzog	2016-05-06 00:14:21
2	67	Emilio_Bernier52	2016-05-06 13:04:30
3	63	Elenor88	2016-05-08 01:30:41
4	95	Nicole71	2016-05-09 17:30:22
5	38	Jordyn.Jacobson2	2016-05-14 07:56:26

The screenshot displays the MySQL Workbench interface. At the top, the SQL editor contains the query: `#TASK1: Identify the five oldest users on Instagram from the provided database.`
`select * from users`
`username`
`order by created_at`
`limit 5;`
Below the editor, the 'Result Grid' shows the output of the query. It lists five users with their IDs, usernames, and creation timestamps. The 'Output' pane at the bottom shows the execution log, indicating that the query was executed successfully and returned 5 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

Output:

#	Time	Action	Message
1	13:53:49	USE ig_clone	0 row(s) affected
2	13:58:17	select * from users username order by created_at limit 5	5 row(s) returned

2. Inactive User Engagement:

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

SQL QUERY: select users.id,(username) from users

left join photos

on users.id=photos.user_id

where photos.user_id is null;

Result: 26 users who have never posted a single photo on Instagram, Here's are some details of users:

Id	Username
5	Aniya_Hackett
7	Kasandra_Homenick
14	Jaclyn81
21	Rocio33
24	Maxwell.Halvorson
25	Tierra.Trantow
34	Pearl7
36	Ollie_Ledner37
41	Mckenna17
45	David.Osinski47
49	Morgan.Kassulke
53	Linnea59
54	Duane60
57	Julien_Schmidt
66	Mike.Auer39
68	Franco_Keebler64
71	Nia_Haag
74	Hulda.Macejkovic
75	Leslie67
76	Janelle.Nikolaus81
80	Darby_Herzog
81	Esther.Zulauf61
83	Bartholome.Bernhard
89	Jessyca_West
90	Esmeralda.Mraz5
91	Bethany20

```

30 # Task2: Identify users who have never posted a single photo on Instagram.
31
32 • select users.id, (username) from users
33 left join photos
34 on users.id=photos.user_id
35 where photos.user_id is null;
36

```

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

	id	username
▶	5	Aniya_Hackett
	7	Kassandra_Homenick
	14	Jadyn81
	21	Rocio33
	24	Maxwell.Halvorson
	25	Tierra.Trantow
	34	Pearl7
	36	Ollie_Ledner37
	41	Mckenna17
	45	David.Osinski47
	49	Morgan.Kassulke
	53	Linnea59
	54	Duane60
	57	Julien_Schmidt
	66	Mike.Auer39
	68	Franco_Keebler64
	71	Nia_Haag
	74	Hulda.Macejkovic
	75	Leslie67
	76	Janelle.Nikolaus81
	80	Darby_Herzog
	81	Fether.Wilma161

Result 6 x

3. Contest Winner Declaration:

Task: Determine the winner of the contest and provide their details to the team

SQL QUERY: SELECT users.username,
 photos.id,
 photos.image_url,
 COUNT(likes.user_id) AS total_likes
 FROM
 photos
 INNER JOIN
 likes ON likes.photo_id = photos.id
 INNER JOIN
 users ON photos.user_id = users.id
 GROUP BY

photos.id, users.username, photos.image_url

ORDER BY

total_likes DESC

LIMIT 1;

Result: Details of the contest winner:

username	id	image_url	total_likes
Zack_Kemmer93	145	https://jarret.name	48

```
61      #Task3: Determine the winner of the contest and provide their details to the team.
62
63
64      SELECT
65          users.username,
66          photos.id,
67          photos.image_url,
68          COUNT(likes.user_id) AS total_likes
69      FROM
70          photos
71      INNER JOIN
72          likes ON likes.photo_id = photos.id
73      INNER JOIN
74          users ON photos.user_id = users.id
75      GROUP BY
76          photos.id, users.username, photos.image_url
77      ORDER BY
78          total_likes DESC
79      LIMIT 1;
```

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

username	id	image_url	total_likes
Zack_Kemmer93	145	https://jarret.name	48

Result 7 x

4. Hashtag Research:

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

SQL QUERY: SELECT

tags.tag_name, COUNT(*) AS total_tags

```

FROM

photo_tags

JOIN

tags ON photo_tags.tag_id = tags.id

GROUP BY tags.id

ORDER BY total_tags DESC

LIMIT 5;

```

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

```

104      #Task:Identify and suggest the top five most commonly used hashtags on the platform.
105
106
107 •  SELECT
108     tags.tag_name, COUNT(*) AS total_tags
109   FROM
110     photo_tags
111  JOIN
112     tags ON photo_tags.tag_id = tags.id
113  GROUP BY tags.id
114  ORDER BY total_tags DESC
115  LIMIT 5;
116
117
118

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	tag_name	total_tags
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

Result 2

×

5. Ad Campaign Launch:

Task: Determine the day of the week when most users register on Instagram.

Provide insights on when to schedule an ad campaign.

SQL QUERY:

```
SELECT  
  
DAYNAME(created_at) AS day, COUNT(*) AS total_reg  
  
FROM  
  
users  
  
GROUP BY day  
  
ORDER BY total_reg DESC  
  
LIMIT 1;
```

Result: Day of the week when most users register on Instagram:

day	total_reg
Thursday	16

```
163 #Task: Determine the day of the week when most users register on Instagram.Provide insights on when to schedule an ad campaign.  
164  
165 • SELECT  
166 DAYNAME(created_at) AS day, COUNT(*) AS total_reg  
167 FROM  
168 users  
169 GROUP BY day  
170 ORDER BY total_reg DESC  
171 LIMIT 1;  
172  
173  
174  
175  
176  
177
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

day	total_reg
Thursday	16

(B) INVESTOR METRICS:

1. User Engagement:

Task: 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.

SQL QUERY: SELECT(

```
SELECT
count(*)
FROM
photos)
/
(
SELECT
count(*)
FROM
users) AS avg;
```

Result: Average number of posts per user on Instagram = 2.5700

176 #Tasks: 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.

177

178 SELECT(

179 SELECT

180 count(*)

181 FROM

182 photos)

183 /

184 (

185 SELECT

186 count(*)

187 FROM

188 users) AS avg;

Result Grid

avg
2.5700

2. Bots & Fake Accounts:

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

SQL QUERY: SELECT

```
users.username, COUNT(*) AS tot_num_likes
FROM
users
JOIN
likes ON users.id = likes.user_id
GROUP BY users.id
HAVING tot_num_likes = (SELECT
COUNT(*)
FROM
photos);
```

Result : Following are the fake users or potential bots in Instagram:

username	tot_num_likes
Anlya_Hackett	257
Jaclyn81	257
Rocio33	257
Maxwell.Halyvorson	257
Ollie_Ledner37	257
Mckennai17	257
Duane60	257
Jullen.Schmidt	257
Mike.Auer39	257
Nia_Haag	257
Leslie67	257
Janelle.Nikolaus81	257
Bathany20	257


```
93
94 • SELECT
95     users.username, COUNT(*) AS tot_num_likes
96 FROM
97     users
98 JOIN
99     likes ON users.id = likes.user_id
00 GROUP BY users.id
01 HAVING tot_num_likes = (SELECT
02     COUNT(*)
03
04 FROM
05     photos);
06
07
08
09
```

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

username	tot_num_likes
Aniya_Hackett	257
Jadyn81	257
Rocio33	257
Maxwell.Halvorson	257
Ollie_Ledner37	257
Mckenna17	257
Duane60	257
Julien_Schmidt	257
Mike.Auer39	257

result 6 x