# INSTAGRAM USER ANALYTICS

-YAADHAV R

## PROJECT DESCRIPTION :

THE OBJECTIVE OF THIS INSTAGRAM DATA ANALYSIS PROJECT IS TO EXTRACT MEANINGFUL INSIGHTS FORM USER INTERACTIONS AND ENGAGEMENT ON THE PLATFORM WHICH CAN THEN USED BY VARIOUS TEAMS WITHIN THE BUSINESS .HENCE THE TEAMS WILL BE ABLE TO INCREASE THE USER EXPERIENCES AND THERE BY HELPING THE BUSINESS TO GROW.

## TECH-STACK USED :

MYSQL WORKBENCH - VERSION 8.0.34 IS USED IN THIS PROJECT AS IT IS :

- IT A SIMPLE AND EASY TO USE SOFTWARE.
- IT IS FASTER COMPARED TO OTHER SOFTWARES DUE TO ITS SIMPLICITY.

## CONCEPTS APPLIED:

**DDL COMMAND USED:**

- CREATE

**DML COMMANDS USED :**

- SELECT
- INSERT

**OTHER COMMANDS USED:**

- FROM CLAUSE.
- WHERE CLAUSE.
- ORDER BY CLAUSE.
- GROUP BY CLAUSE.

**FUNCTIONS USED:**

- RANK()
- COUNT()
- OVER()
- MAX()

**OTHER CONCEPTS:**

- SUB-QURIES
- COMMON TABLE EXPRESSIONS
- DERIVED TABLES

# TASKS

## A) MARKETING ANALYSIS

## 1) LOYAL USER REWARD:

```
90
91    # FINDING THE 5 OLDEST USERS
92
93    SELECT sl_no, id, username, created_at
94    FROM (
95        SELECT *, RANK() OVER( ORDER BY created_at ) AS sl_no
96        FROM users
97    ) as ranked_table
98    WHERE sl_no<=5;
99
100
```

100% 11:104

Result Grid Filter Rows: Search Export:

| 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 |

### APPROACH

- ALL THE USERS IN THE ‘USERS’ TABLE WERE RANKED BASED ON THE ID CREATED DATE WHICH IS GIVEN AN ALIASE AS ‘RANKED_TABLE’.
- FROM THE ‘RANKED_TABLE’ TOP 5 OLDEST USERS WERE FILTERED BY USING A WHERE CLAUSE.

### INSIGHTS

- ALL THE 5 USERS ARE REAL ACCOUNTS AS THEY WERE NOT AMONG THE BOT ACCOUNTS LIST.
- THE FIRST USER WAS FOUND OUT BE AN INACTIVE USER.
- SO REWARDING THESE OLDEST USERS WITH A DIGITAL GIFT CARD, STATING THAT THEY ARE AMONG THE TOP 5 OLDEST USERS, MIGHT RESULT IN THEM POSTING THE CARDS ON THEIR ACCOUNTS.

## 2) INACTIVE USER ENGAGEMENT :

```sql
# IDENTITYFING INACTIVE USERS

SELECT id, username
FROM users
WHERE id NOT IN (
    SELECT DISTINCT user_id
    FROM photos
);
```

100% 12:115

Result Grid Filter Rows: Search Edit: Export/Import:

| 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.Mraz57 |
| 91 | Bethany20 |

### APPROACH

- A LIST OF DISTINCT USER IDS FROM THE ‘PHOTOS’ TABLE IS CREATED.
- THEN THE USERS IDS IN THIS LIST ARE FILTERED OUT OF THE ‘USERS’ TABLE.

### INSIGHTS

- OUT OF THE 26 INACTIVE USERS, 12 ARE FOUND OUT BE BOT/FAKE ACCOUNTS. SO THE ACTUAL REAL INACTIVE USERS ARE 14.
- SENDING PROMOTIONAL EMAIL OR DMS TO THESE USERS MAY ENCOURAGE THEM TO POST SOMETHING ON THEIR ACCOUNTS.
- CONDUCTING MORE CONTESTS MIGHT ALSO DRIVE THEM TO POST.

## 3) CONTEST WINNER DECLARATION :

```sql
# DETERMINING THE MOST LIKED POST

WITH likes_count AS (
    SELECT photo_id, COUNT(user_id) AS no_of_likes
    FROM likes
    GROUP BY photo_id
)

, max_liked AS (
    SELECT user_id, photo_id, no_of_likes
    FROM likes_count
    INNER JOIN photos
    ON likes_count.photo_id = photos.id
    AND no_of_likes = (SELECT MAX(no_of_likes) FROM likes_count)
)

SELECT id, username, max_liked.photo_id, max_liked.no_of_likes
FROM users
INNER JOIN max_liked
ON users.id = max_liked.user_id;
```

100% 1:134

Result Grid Filter Rows: Search Export:

| id | username | photo_id | no_of_likes |
|---|---|---|---|
| 52 | Zack_Kemmer93 | 145 | 48 |

## APPROACH

- FIRST A WITH CLAUSE CTE AS 'LIKES_COUNT' WHICH CONTAINS PHOTO_ID AND ITS RESPECTIVE LIKES COUNT IS CREATED.
- PHOTO_ID WITH MAXIMUM LIKES IS FILTERED FROM 'LIKES_COUNT' AND THE RESPECTIVE USER_ID IS JOINED TO IT FORM 'PHOTOS' TABLE ON PHOTO_ID.
- THIS IS AGAIN JOINED WITH 'USERS' TABLE ON USER_ID TO GET THE USERNAME .

## INSIGHTS

- DECLARING THE IDENTIFIED WINNER ON THE OFFICIAL ACCOUNT OF INSTAGRAM WOULD BE REWARDING.
- ALSO REWARDING THE WINNER WITH OFFICIAL INSTAGRAM MERCHANDISE AND GOODIES WOULD ENCOURAGE MORE USERS ENGAGEMENT IN THE UPCOMING CONTESTS.

# 4) HASHTAG RESEARCH :

```sql
# IDENTIFYING THE 5 MOST POPULAR HASHTAGS

WITH hashtags AS (
    SELECT tag_id, COUNT(*) AS tag_count
    FROM photo_tags
    GROUP BY tag_id
)

SELECT sl_no, tag_id, tag_name, tag_count
FROM (
    SELECT *, RANK() OVER(ORDER BY tag_count DESC) AS sl_no
    FROM hashtags
    INNER JOIN tags
    ON tags.id=hashtags.tag_id
) AS most_popular
WHERE sl_no<=5;
```

100% 48:154

Result Grid Filter Rows: Search Export:

| sl_no | tag_id | tag_name | tag_count |
|---|---|---|---|
| 1 | 21 | smile | 59 |
| 2 | 20 | beach | 42 |
| 3 | 17 | party | 39 |
| 4 | 13 | fun | 38 |
| 5 | 5 | food | 24 |
| 5 | 11 | lol | 24 |
| 5 | 18 | concert | 24 |

## APPROACH

- FIRST A WITH CLAUSE CTE WITH ALIASE ‘HASHTAGS’ WHICH CONTAINS TAG_ID AND ITS RESPECTIVE TAGS COUNT IS CREATED.
- THE TAGS GETS RANKED BY ITS HIGHEST TAG_COUNT AND JOINED WITH ‘TAGS’ TABLE ON TAG_ID.
- FINALLY THE TOP 5 TAGS HAVE BEEN FILTERED WITH A WHERE CLAUSE.

## INSIGHTS

- THESE HASHTAGS CAN BE PROVIDED TO THE UPCOMING INFLUENCERS AS THIS WOULD HELP THEM IN MAXIMIZING REACH FOR THEIR POSTS.
- INCREASING THE FEED REACH FOR THESE SPECIFIC TAGS WOULD RESULT IN MORE REACH OF POSTS WITH THESE TAGS

# 5) AD CAMPAGIAN LAUNCH :

```sql
# DETERMINING THE DAY OF WEEK WHEN MOST USERS REGISTERED

WITH day_count_table AS (
    SELECT DAYOFWEEK(created_at) AS day_of_week, COUNT(*) AS day_count
    FROM users
    GROUP BY day_of_week
    ORDER BY day_of_week
)

SELECT *
FROM day_count_table
WHERE day_count=( SELECT MAX(day_count) FROM day_count_table );
```

100% 1:166

Result Grid Filter Rows: Search Export:

| day_of_week | day_count |
|---|---|
| 1 | 16 |
| 5 | 16 |

## APPROACH

- WITH THE HELP OF DAYOFWEEK() FUNCTION, DAY_OF_WEEK AND ITS COUNT IS EXTRACTED FROM CREATED_AT IN THE 'USERS' TABLE .
- THEN THE DAY_OF_WEEK WITH MAXIMUM VALUE IS FILTERED.

## INSIGHTS

- SUNDAY AND THURSDAY ARE THE DAYS THAT ARE FOUND TO BE HAVING HIGHEST NUMBER OF USER REGISTRATION
- SO LAUNCHING AD CAMPAIGN ON THESE DAYS WOULD BE IDEAL AS IT MIGHT GET A MORE REACH.
- USAGE OF THE ABOVE MENTIONED HASHTAG WILL ALSO HELP IN MAXIMIZING THE REACH

# B) INVESTOR METRICS :

## 1) USER ENGAGEMENT :

```
164
165     # CALCULATING AVG POSTS PER USER
166
167     SELECT
168         ROUND(COUNT(*) / COUNT(DISTINCT user_id), 3) AS active_users_avg,
169         ROUND(COUNT(*) / (SELECT COUNT(id) FROM users), 3) AS total_users_avg
170     FROM
171         photos;
172
173
```

100% 22:176

Result Grid | Filter Rows: Search | Export:

| active_users_avg | total_users_avg |
|---|---|
| 3.473 | 2.570 |

### APPROACH

- COUNT OF DISTINCT USER_ID AND COUNT ID FROM 'USERS' TABLE ARE DIVIDED FROM THE TOTAL COUNT OF PHOTOS IN THE 'PHOTOS' TABLE AND THE RESULTS ARE DISPLAYED AS ACTIVE USERS AVERAGE AND TOTAL USERS AVERAGE RESPECTIVELY.

### INSIGHTS

- THE AVERAGE POST PER USER IS 2.570, ALL THE 100 USERS ARE TAKEN CONSIDERATION FOR CALCULATING THIS NUMBER.
- THE AVERAGE INCREASES TO 3.473 LEAVING OUT THE INACTIVE AND BOT ACCOUNTS IN CONSIDERATIONS.
- SO THE ACTIVE USERS ARE CONSTANTLY ENGAGING IN THE PLATFORM.
- AS REGULAR CONTESTS ARE CONDUCTED THIS NUMBER WILL ONLY INCREASE IN TIME.
- ONLY THE NO OF BOT/FAKE ACCOUNTS IS QUIET CONCERNING.

## 2) BOTS AND FAKE ACCOUNTS :

```sql
# IDENTIFYING BOT ACCOUNTS

WITH liked_table AS (
    SELECT user_id, COUNT(user_id) AS liked_count
    FROM likes
    GROUP BY user_id
)

SELECT
    id AS user_id, username AS bot_accounts
FROM users
INNER JOIN liked_table
ON users.id = liked_table.user_id
WHERE liked_count = (SELECT COUNT(*) FROM photos);
```

100% 1:189

Result Grid Filter Rows: Search Export:

| user_id | bot_accounts |
|---|---|
| 5 | Aniya_Hackett |
| 14 | Jaclyn81 |
| 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 |

## APPROACH

- TOTAL COUNT OF PHOTOS EACH USERS LIKED AS LIKED_COUNT AND THEIR USER_ID IS CREATED AS A WITH CLAUSE CTE.
- THEN IT IS JOINED WITH ‘USERS’ TABLE ON USER_ID AND THE IDS WEHRE LIKED_COUNT IS EQUAL TO THE TOTAL NUMBER OF PHOTOS UPLOADED FORM ‘PHOTOS’ TABLE ARE FILTERED.

## INSIGHTS

- THERE ARE 12 BOT/FAKE ACCOUNTS OUT OF THE TOTAL 100 ACCOUNTS WHICH LIKED ALL THE PHOTOS POSTED.
- THIS MIGHT REDUCE AS THE TEAMS WILL BE WORKING ON IT.

## RESULT:

- FIRST OF ALL THIS PROJECT HELPED TO UNDERSTAND THE CONCEPTS I LEARNED IN BETTER AND INTRESTING WAY.
- SO NOW FEEL CONFIDENT IN APPLYING MYSQL SKILLS AS I WAS ABLE TO COMPLETE ALL THE GIVEN TASKS.
- IT WAS SO EXICTING TO ANALYZE THE OUTPUTS AND DERIVE INSIGHTS FROM IT.
- OVERALL IT WAS GREAT TO EXPERIENCE TO APPLY THE SKILLS AND LEARN ALONG THE WAY.
- LOOKING FORWARD TO FACE THE UPCOMING CHALLENGES WITH CONFIDENCE AND EXICTMENT.

## DRIVE LINK OF SQL FILE:

HTTPS://DRIVE.GOOGLE.COM/FILE/D/19R9QGFEEMO3EKJKKP_QZBKLL2UAPMNOD/VIEW?USP=SHARE_LINK