Instagram Insights Data Analysis

SQL Analysis on Instagram Engagement Metrics

Project Introduction

Project Goal

Analyze Instagram performance using SQL.

dim_date Table

Calendar metadata by date.

fact_account Table

Profile visits, new followers.

fact_content Table

Posts, reach, likes, shares.

Unique Post Types

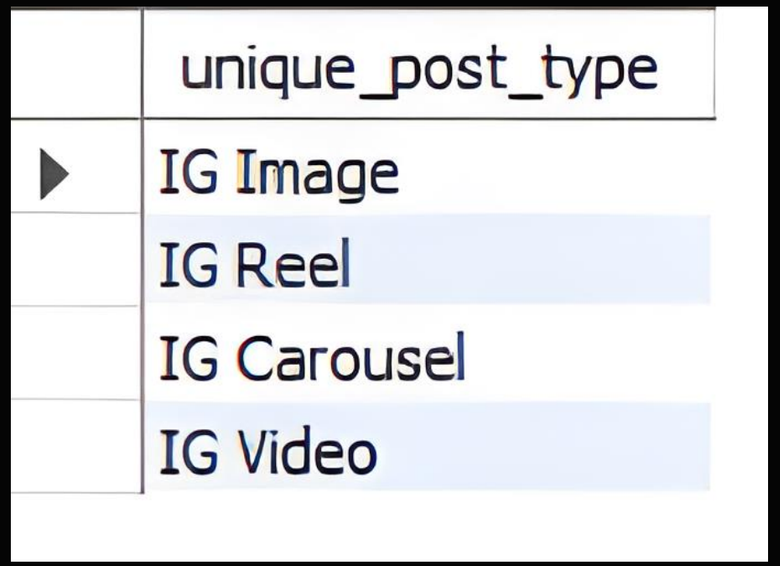
Question:

How many unique post types are in fact_content?

SQL Query:

```
select
		distinct(post_type) as unique_post_type
from gdb0120.fact_content;
```

This query identifies the distinct content formats used on Instagram, like photos, videos, or carousels. Understanding these types helps categorize content strategy.



unique_post_type
IG Image
IG Reel
IG Carousel
IG Video

Post Type Impressions

Question:

Highest and lowest impressions per post type?

SQL Query:

```
select
		post_type,
		max(impressions) as highest_impressions,
		min(impressions) as lowest_impressions
from gdb0120.fact_content
group by post_type;
```

post_type	highest_impressions	lowest_impressions
IG Image	129694	23367
IG Reel	339708	87570
IG Carousel	9677	3264
IG Video	73321	8741

This query reveals the performance range for each content format, highlighting which types can achieve viral reach and which might struggle with visibility.

Weekend Posts: March & April

Question:

Retrieve all posts made on weekends during March and April.

SQL Query:

```
select
		fc.*
from gdb0120.fact_content fc
join gdb0120.dim_dates dd on fc.date = dd.date
where dd.month_name in ('March','April')
and dd.weekday_or_weekend = 'Weekend';
```

date	post_category	post_type	video_duration	carousel_item_count	impressions	reach	shares	follows	likes	comments	saves
2023-03-04	Earphone	IG Video	291	0	12265	3668	69	92	327	7	18
2023-03-05	Smartwatch	IG Image	0	0	62770	18001	273	360	1194	28	76
2023-03-11	Mobile	IG Carousel	0	3	5899	1093	45	12	53	0	6
2023-03-12	Laptop	IG Image	0	0	79416	23474	327	259	1235	69	204
2023-03-18	Mobile	IG Carousel	0	3	9157	2254	67	58	55	6	15
2023-03-19	Smartwatch	IG Carousel	0	3	4146	1079	42	17	43	1	6
2023-03-25	Earphone	IG Reel	22	0	132284	66721	1093	1482	3622	83	695
2023-03-26	Mobile	IG Image	0	0	63425	26113	435	336	1994	68	179
2023-04-01	Mobile	IG Carousel	0	3	4549	1052	27	18	35	1	6
2023-04-02	Earphone	IG Video	163	0	54672	16126	172	182	938	22	81
2023-04-08	Other Gadgets	IG Video	258	0	37955	12663	204	164	753	31	63
2023-04-09	Mobile	IG Image	0	0	52278	14438	271	167	1393	36	44
2023-04-15	Laptop	IG Reel	30	0	123270	39850	296	1486	3926	101	1139
2023-04-16	Other Gadgets	IG Reel	29	0	115701	66829	937	929	5749	94	658
2023-04-22	Laptop	IG Video	172	0	33604	14682	255	349	1038	22	73
2023-04-23	Earphone	IG Video	229	0	36973	13629	224	244	929	30	68
2023-04-29	Earphone	IG Video	206	0	43526	11799	134	138	646	12	59
2023-04-30	Mobile	IG Reel	59	0	185017	63990	1010	2238	6039	94	330

This query helps analyze content performance during specific periods, crucial for seasonal campaigns and audience engagement patterns.

Monthly Account Statistics

Question:

Sum of profile visits and new followers per month.

SQL Query:

```
select
                dd.month_name as month_name,
    sum(fa.profile_visits) as total_profile_visits,
    sum(fa.new_followers) as total_new_followers
from gdb0120.fact_account fa
join gdb0120.dim_dates dd on fa.date = dd.date
group by dd.month_name
order by dd.month_name;
```

Tracks overall account growth and audience interest over time, providing monthly performance snapshots.

month_name	total_profile_visits	total_new_followers
April	29852	21799
August	42094	24371
February	20628	15254
January	26512	17053
July	54352	33302
June	103350	76942
March	23132	18285
May	106571	66984
September	41522	28523

July Likes Per Category

Question:

Total likes for each post category in July.

SQL Query:

```
with july_likes as (
select
                post_category,
   sum(likes) as total_likes
from gdb0120.fact_content fc
join gdb0120.dim_dates dd on fc.date = dd.date
where dd.month_name = 'July'
group by fc.post_category
)
select
                *
from july_likes
order by total_likes desc;
```

post_category	total_likes
Other Gadgets	26519
Tech Tips	20296
Mobile	16338
Earphone	14435
Smartwatch	3918

This query identifies top-performing content categories, helping refine content strategy for maximum audience appeal.

Unique Post Categories Per Month

Question:

Unique post categories and their count per month.

SQL Query:

```
SELECT
    dd.month_name,
    GROUP_CONCAT(DISTINCT fc.post_category ORDER BY fc.post_category
fc.post_category SEPARATOR ',') AS post_category_names,
    COUNT(DISTINCT fc.post_category) AS post_category_count
FROM gdb0120.fact_content fc
JOIN gdb0120.dim_dates dd
    ON fc.date = dd.date
where dd.month_name in ('April','February')
GROUP BY dd.month_name
ORDER BY
    STR_TO_DATE(dd.month_name, '%M');
```

month_name	post_category_names	post_category_count
April	Earphone,Laptop,Mobile,Other Gadgets,Smart...	5
February	Earphone,Laptop,Mobile,Smartwatch	4

This query reveals content diversity and consistency over time. Note: Use GROUP_CONCAT() for MySQL.

Reach Percentage by Post Type

Question:

Calculate reach percentage for each post type.

SQL Query:

```
WITH reach_total AS (
    SELECT SUM(reach) AS total_reach_all FROM gdb0120.fact_content
)
SELECT
    post_type,
    SUM(reach) AS total_reach,
    concat(ROUND(SUM(reach) * 100.0 / (SELECT total_reach_all FROM reach_total),
reach_total), 2),"%") AS reach_percentage
FROM gdb0120.fact_content
GROUP BY post_type;
```

Determines which post types are most effective at expanding audience reach. This is key for optimizing visibility.

post_type	total_reach	reach_percentage
IG Image	1866381	21.38%
IG Reel	5379091	61.63%
IG Carousel	60465	0.69%
IG Video	1422300	16.30%

Quarterly Comments & Saves

Question:

Total comments and saves per post category by quarter.

SQL Query:

```
SELECT
    fc.post_category,
    CASE
        WHEN dd.month_name IN ('January', 'February', 'March') THEN 'Q1'
        WHEN dd.month_name IN ('April', 'May', 'June') THEN 'Q2'
        WHEN dd.month_name IN ('July', 'August', 'September') THEN 'Q3'
        WHEN dd.month_name IN ('October', 'November', 'December') THEN 'Q4'
'Q4'
    END AS quarter,
    SUM(fc.comments) AS total_comments,
    SUM(fc.saves) AS total_saves
FROM gdb0120.fact_content fc
JOIN gdb0120.dim_dates dd ON fc.date = dd.date
GROUP BY fc.post_category, quarter
ORDER BY quarter, post_category;
```

Analyzes quarterly audience engagement, identifying seasonal patterns in comments and saves. This is crucial for strategic content planning.

post_category	quarter	total_comments	total_saves
Earphone	Q1	351	2230
Laptop	Q1	418	2837
Mobile	Q1	1836	9843
Smartwatch	Q1	600	2860
Earphone	Q2	589	3602
Laptop	Q2	452	2248
Mobile	Q2	2313	17207
Other Gadgets	Q2	1622	12041
Smartwatch	Q2	1358	12581
Tech Tips	Q2	2201	17649
Earphone	Q3	427	3247
Mobile	Q3	1134	5285
Other Gadgets	Q3	964	4457
Smartwatch	Q3	971	3326
Tech Tips	Q3	1596	12976

Top 3 New Followers

Question:

Top three dates in each month with the highest number of new followers.

SQL Query:

```
WITH ranked_followers AS (
    SELECT
        dd.month_name,
        fa.date,
        fa.new_followers,
        ROW_NUMBER() OVER (PARTITION BY dd.month_name ORDER BY
fa.new_followers DESC) AS rnk
    FROM gdb0120.fact_account fa
    JOIN gdb0120.dim_dates dd ON fa.date = dd.date
)
SELECT
    month_name,
    date,
    new_followers
FROM ranked_followers
WHERE rnk <= 3;
```

month_name	date	new_followers
April	2023-04-25	3736
April	2023-04-30	2753
April	2023-04-06	2500
August	2023-08-23	2074
August	2023-08-21	1783
August	2023-08-06	1687
February	2023-02-01	4106
February	2023-02-24	2383
February	2023-02-02	1989
January	2023-01-30	3186
January	2023-01-03	2959
January	2023-01-23	1003
July	2023-07-08	3716
July	2023-07-15	3364
July	2023-07-28	2344
June	2023-06-30	8804
June	2023-06-03	8802
June	2023-06-21	7033
March	2023-03-21	5421
March	2023-03-28	2513
March	2023-03-25	2356
May	2023-05-08	8872
May	2023-05-20	6169
May	2023-05-12	6051

Generate Post type & Shares by Week No

Question:

Takes the 'Week_no' as input and generates a report displaying the total shares for each 'Post_type'

SQL Query:

```
SELECT
    fc.post_type,
    SUM(fc.shares) AS total_shares
FROM gdb0120.fact_content fc
JOIN gdb0120.dim_dates dd ON fc.date = dd.date
WHERE dd.week_no = week_param
GROUP BY fc.post_type;
```

W1

post_type	total_shares
IG Image	1134
IG Reel	659
IG Carousel	40

W3

post_type	total_shares
IG Carousel	111
IG Video	318
IG Image	198

W2

post_type	total_shares
IG Image	861
IG Video	410
IG Carousel	42

W4

post_type	total_shares
IG Video	843
IG Reel	1112
IG Image	924

Thank you