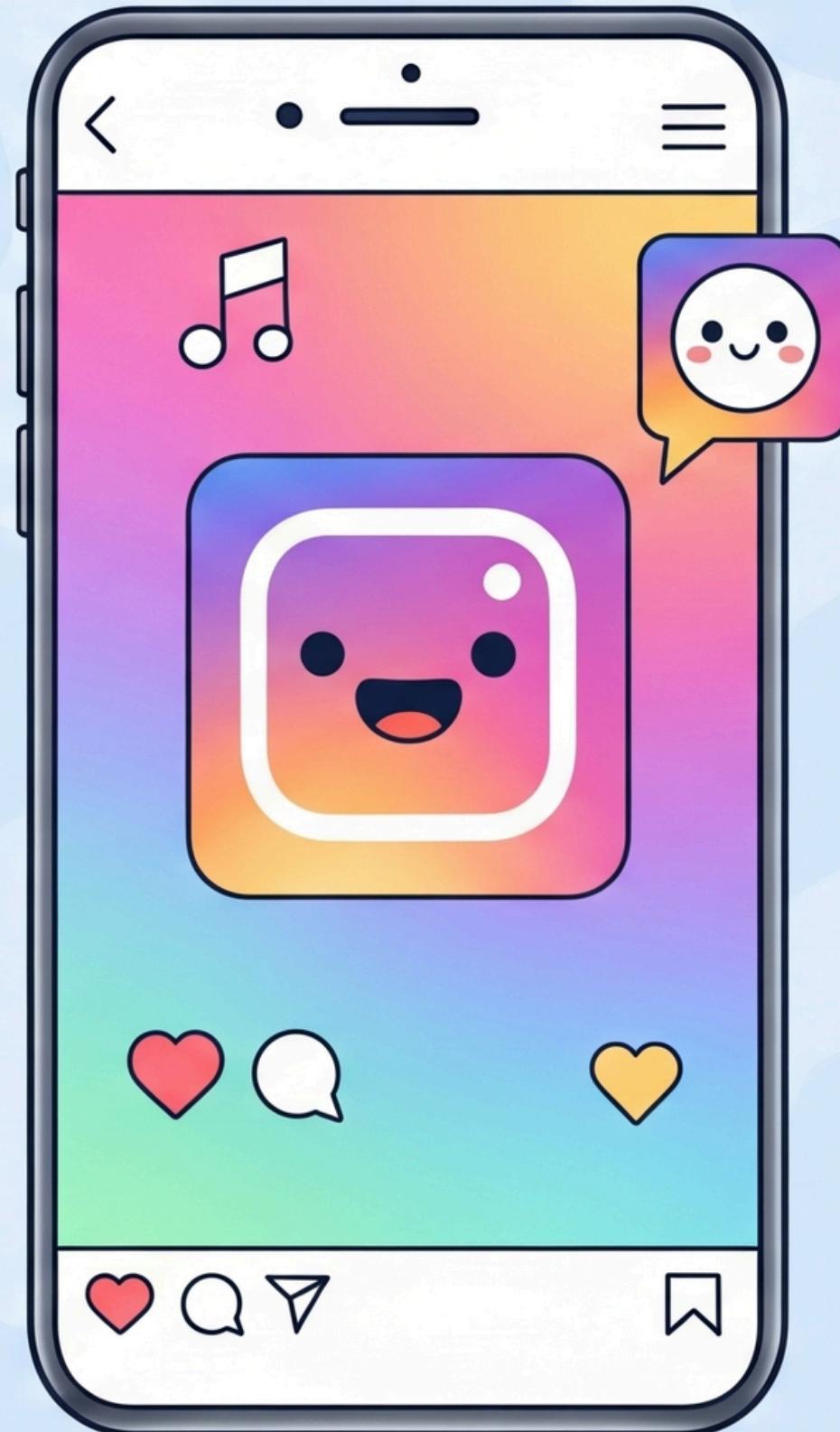


Decoding Tech Influence

A SQL-Based Analysis of Instagram Performance

presented by : Masihuzzaman Sayyed



The Objective

Analyse influencer data to answer 10 key business questions.

I

The Methodology

Employ SQL queries to extract, filter, and aggregate metrics.

2

The Outcome

Deliver actionable insights for content strategy and growth.

3



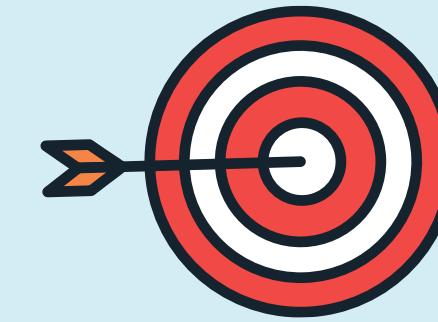
problem:

The first task is to determine how many unique post types are in the **fact_content** table.

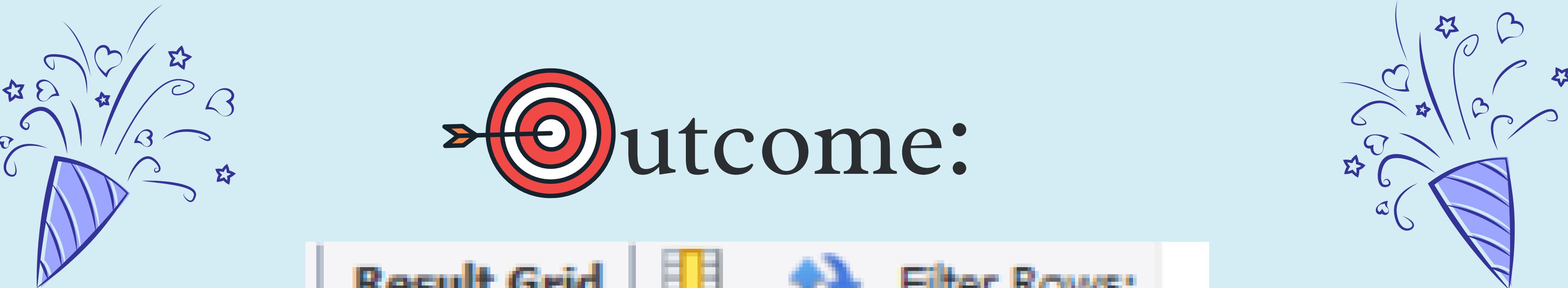


Query:

```
3 • SELECT  
4     COUNT(DISTINCT post_type) AS unique_post_types  
5 FROM  
6     fact_content;
```



outcome:



	unique_post_types
▶	4

The account utilises 4 unique post types, confirming a diverse content strategy is in place, which is a best practice for audience engagement.

problem:

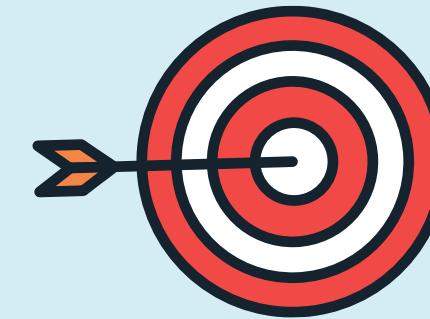


What are the highest and lowest recorded impressions for each post type?

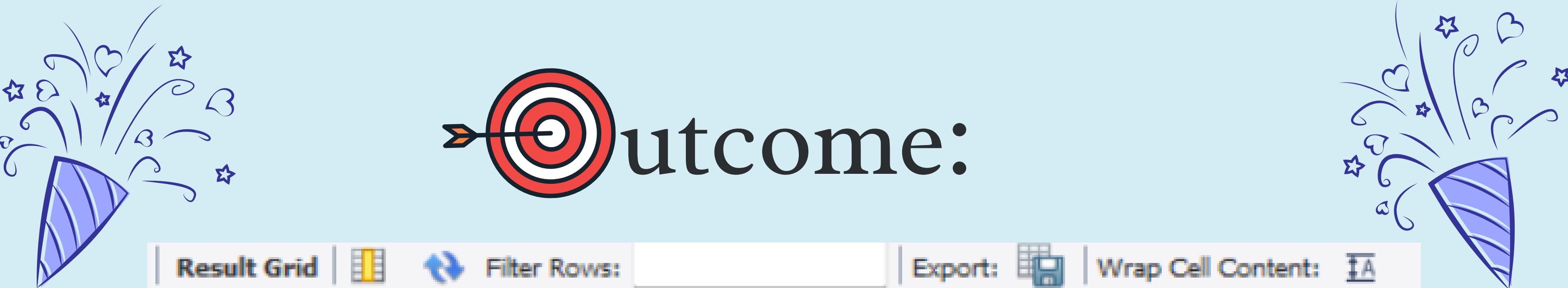
Query:



```
SELECT  
    post_type,  
    MAX(impressions) AS highest_impressions,  
    MIN(impressions) AS lowest_impressions  
FROM  
    fact_content  
GROUP BY  
    post_type;
```



outcome:



A screenshot of a data visualization tool showing a table of Instagram post types and their impression counts. The table has three columns: 'post_type', 'highest_impressions', and 'lowest_impressions'. A red arrow points to the second row, highlighting the IG Reel data.

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

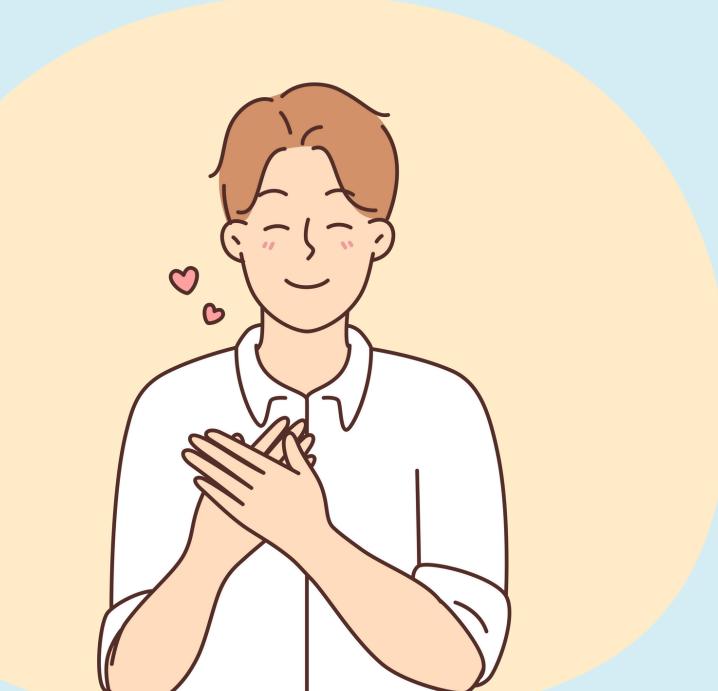
IG Reels demonstrate the highest potential for viral reach. Other formats, while valuable, serve a different purpose and have a much lower impression ceiling.



problem:

Filter all posts published on a weekend in March and April.

Query:



```
SELECT
    fc.*
FROM
    fact_content fc
JOIN
    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

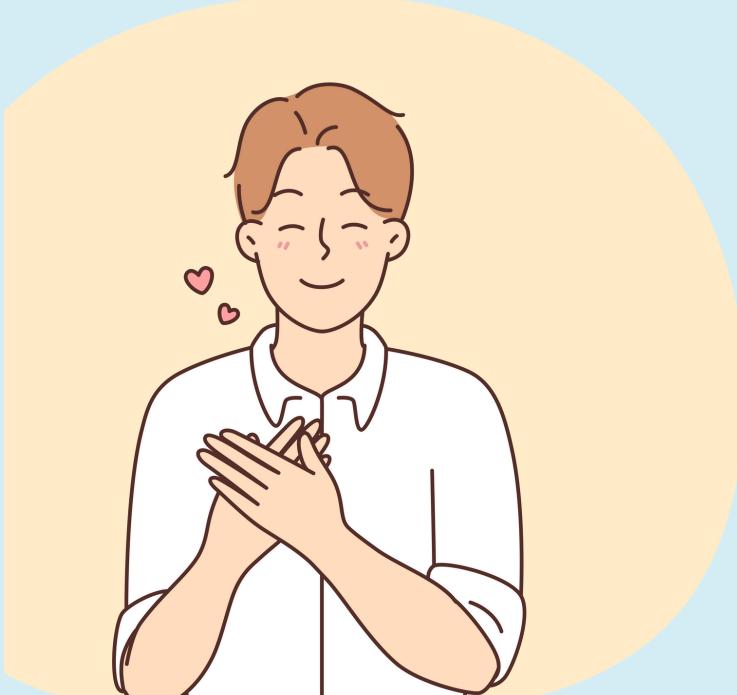
This query generates a targeted dataset. This list is the foundation for a deeper dive into what specific content performs best during peak audience availability on weekends.



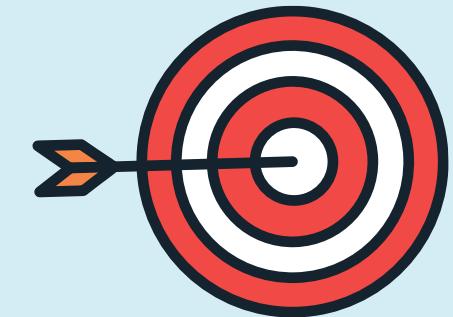
problem:

Create a report of total profile visits and new followers for each month.

Query:



```
SELECT
    dd.month_name,
    SUM(fa.profile_visits) AS total_profile_visits,
    SUM(fa.new_followers) AS total_new_followers
FROM
    fact_account fa
JOIN
    dim_dates dd ON fa.date = dd.date
GROUP BY
    dd.month_name
ORDER BY
    MONTH(fa.date); -- Order chronologically
```



outcome:

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

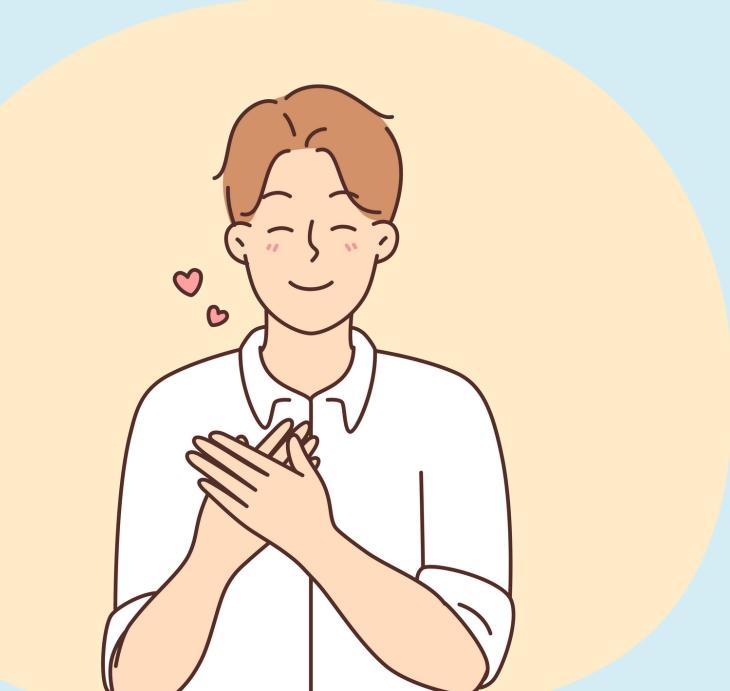
This report provides a high-level view of the account's growth trajectory, enabling stakeholders to immediately identify the strongest months, such as May and June, and ask deeper questions about what drove that success.



problem:

Identify the top-performing post categories for the month of 'July', measured by the total number of likes.

Query:



```
WITH JulyLikes AS (
    SELECT
        fc.post_category,
        SUM(fc.likes) AS total_likes
    FROM
        fact_content fc
    JOIN
        dim_dates dd ON fc.date = dd.date
    WHERE
        dd.month_name = 'July'
    GROUP BY
        fc.post_category
)
SELECT
    post_category,
    total_likes
FROM
    JulyLikes
ORDER BY
    total_likes DESC;
```



outcome:

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

The ranked list clearly shows what topics resonated most in July. If 'Tech Tips' is at the top, it's a strong, data-driven signal to produce more of this content to maximize engagement.



problem:

Report on the diversity of our content by showing the unique post categories and their count for each month.

Query:

```
SELECT
    dd.month_name,
    GROUP_CONCAT(DISTINCT fc.post_category ORDER BY fc.post_category SEPARATOR ', ') AS post_category_names,
    COUNT(DISTINCT fc.post_category) AS post_category_count
FROM
    fact_content fc
JOIN
    dim_dates dd ON fc.date = dd.date
GROUP BY
    dd.month_name
ORDER BY
    MONTH(fc.date);
```





month_name	post_category_names	post_category_count
January	Earphone, Mobile, Smartwatch	3
February	Earphone, Laptop, Mobile, Smartwatch	4
March	Earphone, Laptop, Mobile, Smartwatch	4
April	Earphone, Laptop, Mobile, Other Gadgets, Smartwatch	5
May	Earphone, Laptop, Mobile, Other Gadgets, Smartwatch, Tech Tips	6
June	Mobile, Other Gadgets, Smartwatch, Tech Tips	4
July	Earphone, Mobile, Other Gadgets, Smartwatch, Tech Tips	5
August	Earphone, Mobile, Other Gadgets, Smartwatch, Tech Tips	5
September	Mobile, Other Gadgets, Smartwatch, Tech Tips	4

This report allows us to visualize our content breadth over time. We can track if we're maintaining a broad appeal (e.g., 5 categories in April) or narrowing our focus (e.g., 4 in February)

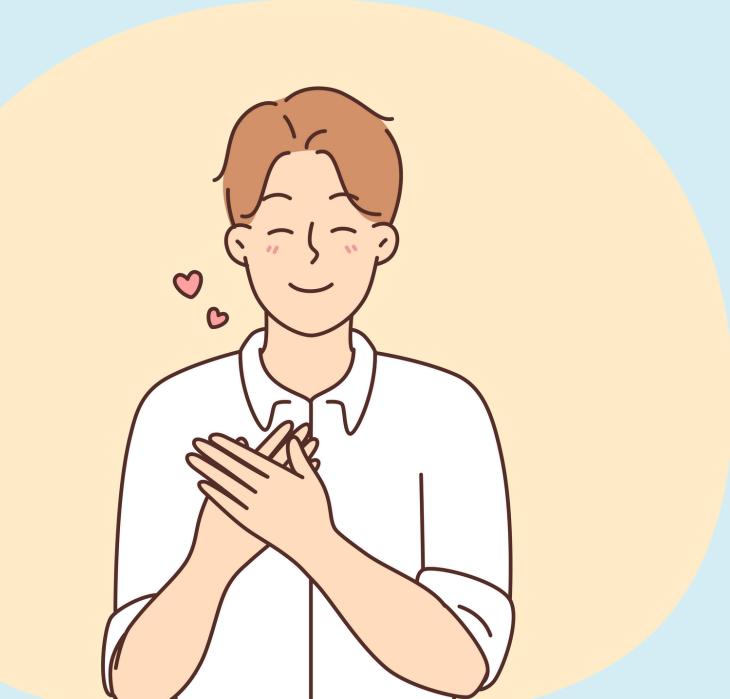
problem:

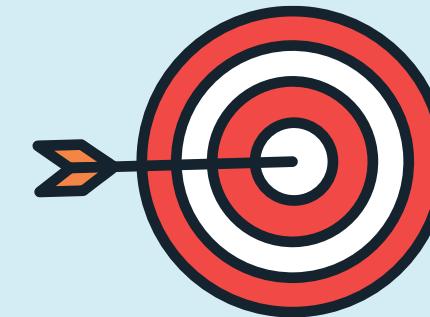
Determine the percentage breakdown of total reach by post type to understand the impact of each format



Query:

```
WITH PostReach AS (
    SELECT
        post_type,
        SUM(reach) AS total_reach
    FROM
        fact_content
    GROUP BY
        post_type
),
OverallReach AS (
    SELECT SUM(reach) AS overall_total_reach FROM fact_content
)
SELECT
    pr.post_type,
    pr.total_reach,
    (pr.total_reach / o.overall_total_reach) * 100 AS reach_percentage
FROM
    PostReach pr, OverallReach o
ORDER BY
    reach_percentage DESC;
```





outcome:

	post_type	total_reach	reach_percentage
▶	IG Reel	5379091	61.6286
	IG Image	1866381	21.3833
	IG Video	1422300	16.2954
	IG Carousel	60465	0.6928

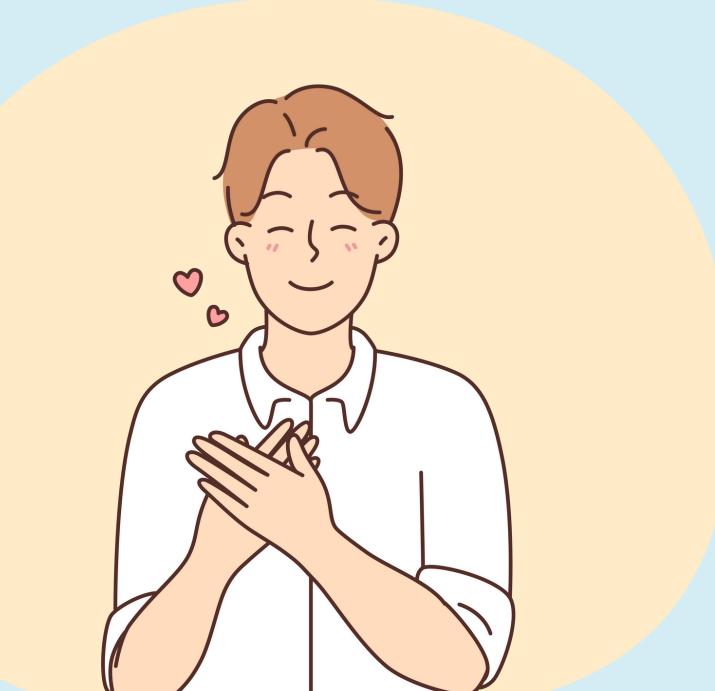
The data makes it clear that one format, likely IG Reels, is responsible for the vast majority of our total reach. This format is the primary engine for reaching new audiences and must be central to any growth strategy.



problem:

Create a quarterly report summarizing total comments and saves for each post category.

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'
    END AS quarter,
    SUM(fc.comments) AS total_comments,
    SUM(fc.saves) AS total_saves
FROM
    fact_content fc
JOIN
    dim_dates dd ON fc.date = dd.date
GROUP BY
    fc.post_category, quarter
ORDER BY
    fc.post_category, quarter;
```



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

This quarterly view reveals broader engagement trends by smoothing out monthly fluctuations. We can spot patterns like a steady increase in 'saves' for 'Smartwatch' content quarter-over-quarter, indicating growing long-term interest.



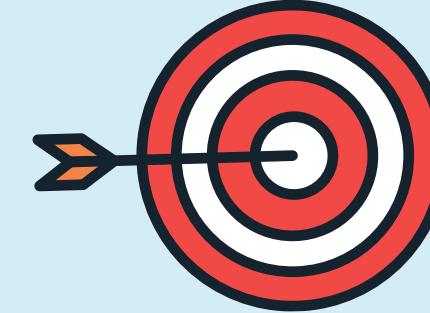
problem:

For each month, identify the top three dates that generated the highest number of new followers

Query:



```
WITH RankedFollowers AS (
    SELECT
        dd.month_name AS month,
        fa.date,
        fa.new_followers,
        DENSE_RANK() OVER(PARTITION BY dd.month_name ORDER BY fa.new_followers DESC) as date_rank
    FROM
        fact_account fa
    JOIN
        dim_dates dd ON fa.date = dd.date
)
SELECT
    month,
    date,
    new_followers
FROM
    RankedFollowers
WHERE
    date_rank <= 3
ORDER BY
    MONTH(date), new_followers DESC;
```



outcome:

	month	date	new_followers
▶	January	2023-01-30	3186
	January	2023-01-03	2959
	January	2023-01-23	1003
	February	2023-02-01	4106
	February	2023-02-24	2383
	February	2023-02-02	1989
	March	2023-03-21	5421
	March	2023-03-28	2513
	March	2023-03-25	2356
	April	2023-04-25	3736
	April	2023-04-30	2753
	April	2023-04-06	2500
	May	2023-05-08	8872
	May	2023-05-20	6169
	May	2023-05-12	6051

This report creates a direct, data-backed link between a specific date and, therefore a specific post and its impact on audience growth. It allows us to answer, "What exact piece of content caused our biggest follower gains?"



problem:

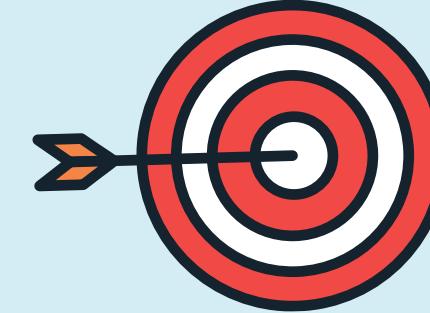
Develop a reusable tool—a stored procedure—that generates a report of total shares for a given week number

Query:



```
CREATE PROCEDURE GetSharesByWeek(IN input_week_no VARCHAR(255))
BEGIN
    SELECT
        fc.post_type,
        SUM(fc.shares) AS total_shares
    FROM
        fact_content fc
    JOIN
        dim_dates dd ON fc.date = dd.date
    WHERE
        dd.week_no = input_week_no
    GROUP BY
        fc.post_type
    ORDER BY
        total_shares DESC;
END$$

DELIMITER ;
```



outcome:

The result here is not just data, but a tool. This stored procedure democratizes data, empowering the marketing team to get performance reports on-demand without needing to write SQL. This shows foresight in making analysis repeatable and accessible.



THANK YOU!

presented by : Masihuzzaman Sayyed

