Q1 : How many unique post types are found in the 'fact\_content' table?

Sql\_Ans: SELECT DISTINCT post\_type FROM fact\_content;

Q2 : What are the highest and lowest recorded impressions for each post type?

Sql\_ans: SELECT

post\_type,

MAX(impressions) AS highest\_impressions,

MIN(impressions) AS lowest\_impressions

FROM fact\_content

GROUP BY post\_type;

Q3: Filter all the posts that were published on a weekend in the month of March and April and export them to a separate csv file.

Sql\_ans:

select fc.\* from fact\_content fc

join dim\_dates dd ON fc.date = dd.date

WHERE dd.weekday\_or\_weekend = 'weekend' and dd.month\_name in ('March','April');

Q4: Create a report to get the statistics for the account. The final output includes the following fields: • month\_name • total\_profile\_visits • total\_new\_followers

Sql\_ans:

SELECT

dd.month\_name AS 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

FIELD(dd.month\_name, 'January', 'February', 'March', 'April', 'May', 'June',

'July', 'August', 'September', 'October', 'November', 'December');

Q5: Write a CTE that calculates the total number of 'likes’ for each 'post\_category' during the month of 'July' and subsequently, arrange the 'post\_category' values in descending order according to their total likes.

Sql\_ans:

WITH likes\_per\_category AS (

SELECT

fc.post\_category AS 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 \*

FROM likes\_per\_category

ORDER BY total\_likes DESC;

Q6:Create a report that displays the unique post\_category names alongside their respective counts for each month. The output should have three columns: • month\_name • post\_category\_names • post\_category\_count

Sql\_ans:

SELECT

dd.month\_name AS month\_name,

GROUP\_CONCAT(DISTINCT fc.post\_category ORDER BY fc.post\_category) 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 FIELD(dd.month\_name, 'January','February','March','April','May','June',

'July','August','September','October','November','December');

Q7: What is the percentage breakdown of total reach by post type? The final output includes the following fields: • post\_type • total\_reach • reach\_percentage

Sql\_ans:  
SELECT

post\_type AS post\_type,

SUM(reach) AS total\_reach,

ROUND(SUM(reach) \* 100.0 / (SELECT SUM(reach) FROM fact\_content), 2) AS reach\_percentage

FROM fact\_content

GROUP BY post\_type

ORDER BY reach\_percentage DESC;

Q8:Create a report that includes the quarter, total comments, and total saves recorded for each post category. Assign the following quarter groupings: (January, February, March) → “Q1” (April, May, June) → “Q2” (July, August, September) → “Q3” The final output columns should consist of: • post\_category • quarter • total\_comments • total\_saves

Sql\_ans:

SELECT

fc.post\_category AS post\_category,

CASE dd.month\_name

WHEN 'January' THEN 'Q1'

WHEN 'February' THEN 'Q1'

WHEN 'March' THEN 'Q1'

WHEN 'April' THEN 'Q2'

WHEN 'May' THEN 'Q2'

WHEN 'June' THEN 'Q2'

WHEN 'July' THEN 'Q3'

WHEN 'August' THEN 'Q3'

WHEN 'September' THEN 'Q3'

ELSE 'Other'

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 post\_category, quarter

ORDER BY post\_category, quarter;

Q9: List the top three dates in each month with the highest number of new followers. The final output should include the following columns: • month • date • new\_followers

Sql\_ans:

WITH ranked\_followers AS (

SELECT

dd.month\_name AS month,

fa.date AS date,

fa.new\_followers,

RANK() OVER (PARTITION BY dd.month\_name ORDER BY fa.new\_followers DESC) AS 'rank'

FROM fact\_account fa

JOIN dim\_dates dd ON fa.date = dd.date

)

SELECT

month,

date,

new\_followers

FROM ranked\_followers

WHERE 'rank' <= 3

ORDER BY month, 'rank';

Q10:Create a stored procedure that takes the 'Week\_no' as input and generates a report displaying the total shares for each 'Post\_type'. The output of the procedure should consist of two columns: • post\_type • total\_shares

Sql\_ans:

DELIMITER //

CREATE PROCEDURE GetSharesByPostType(IN input\_week\_no VARCHAR(5))

BEGIN

SELECT

fc.post\_type AS 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;

END //

DELIMITER ;

CALL GetSharesByPostType('W4');