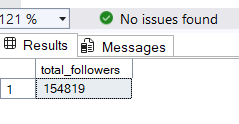
**Influencers SQL Queries**

**KPIs**

1. **Total Followers**

SELECT SUM(CAST(REPLACE(REPLACE(followers, 'm', '') , '.', '') AS BIGINT)) AS total\_followers

FROM influencers;

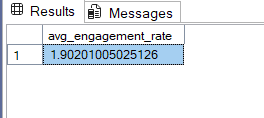
****

1. **Average Engagement Rate:**

SELECT AVG(TRY\_CAST(REPLACE([\_60\_day\_eng\_rate], '%', '') AS FLOAT)) AS avg\_engagement\_rate

FROM influencers

WHERE [\_60\_day\_eng\_rate] IS NOT NULL AND [\_60\_day\_eng\_rate] <> '';



1. **Average Likes per Post:**

SELECT

SUM(

CASE

WHEN total\_likes LIKE '%k' THEN TRY\_CAST(LEFT(total\_likes, LEN(total\_likes) - 1) AS FLOAT) \* 1000

WHEN total\_likes LIKE '%m' THEN TRY\_CAST(LEFT(total\_likes, LEN(total\_likes) - 1) AS FLOAT) \* 1000000

WHEN total\_likes LIKE '%b' THEN TRY\_CAST(LEFT(total\_likes, LEN(total\_likes) - 1) AS FLOAT) \* 1000000000

ELSE TRY\_CAST(total\_likes AS FLOAT)

END

) /

SUM(

CASE

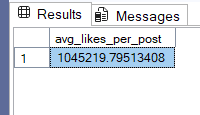
WHEN posts LIKE '%k' THEN TRY\_CAST(LEFT(posts, LEN(posts) - 1) AS FLOAT) \* 1000

ELSE TRY\_CAST(posts AS FLOAT)

END

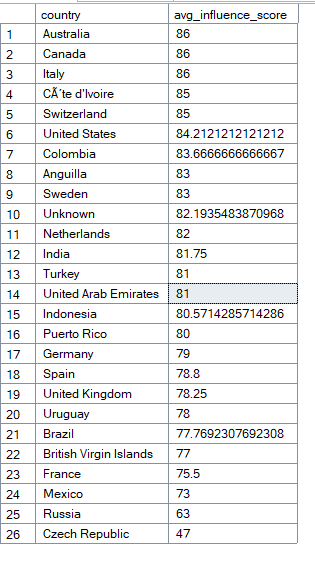
) AS avg\_likes\_per\_post

FROM influencers;



**4. Average Influence Score by Country**

SELECT country, AVG(TRY\_CAST(influence\_score AS FLOAT)) AS avg\_influence\_score

FROM influencers

GROUP BY country

ORDER BY avg\_influence\_score DESC;

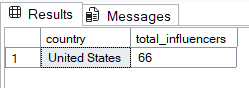
1. **Country with the Most Influencers :**

**SELECT TOP 1 country, COUNT(\*) AS total\_influencers**

**FROM influencers**

**GROUP BY country**

**ORDER BY total\_influencers DESC;**

****

Creating Calculated Fields

**🔹 1. Engagement Rate (ER)**

**Formula:** ([avg\_likes] / [followers]) \* 100

This metric standardizes how engaged an influencer’s audience is, relative to their follower count.

SELECT

channel\_info,

TRY\_CAST(REPLACE(avg\_likes, 'm', '') AS FLOAT) \* 1000000 AS avg\_likes\_num,

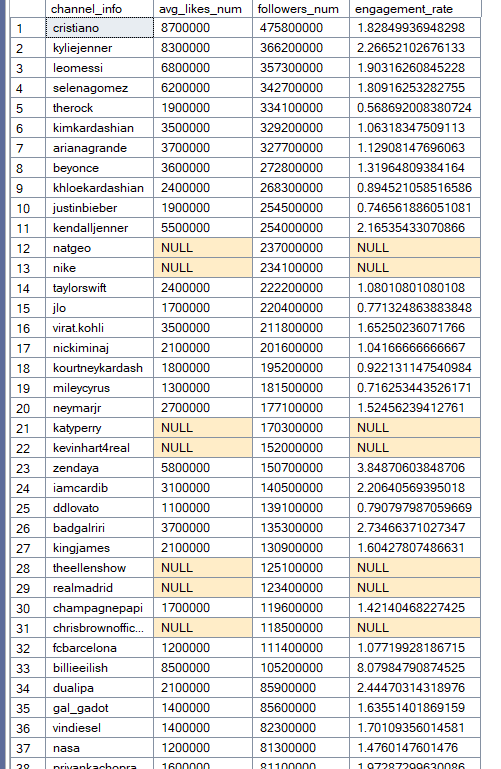
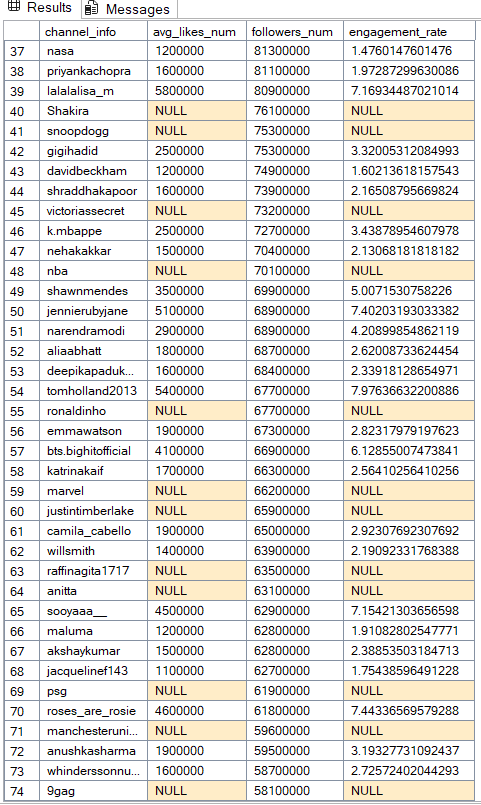
TRY\_CAST(REPLACE(followers, 'm', '') AS FLOAT) \* 1000000 AS followers\_num,

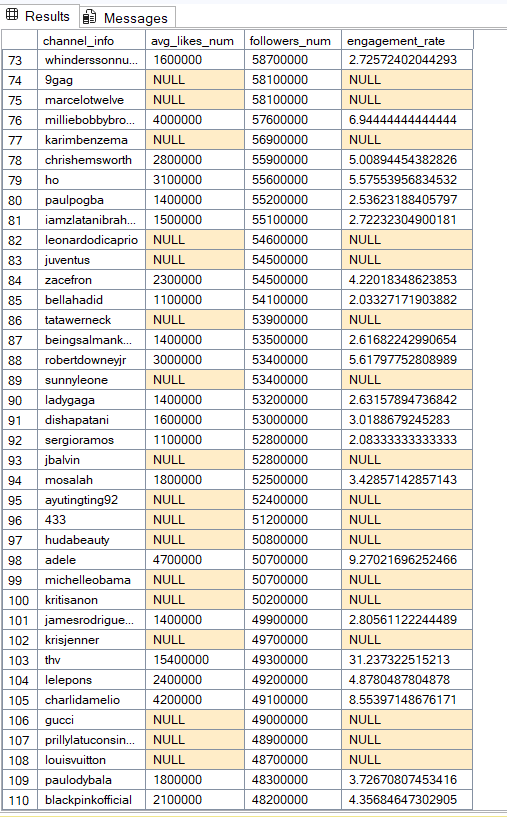
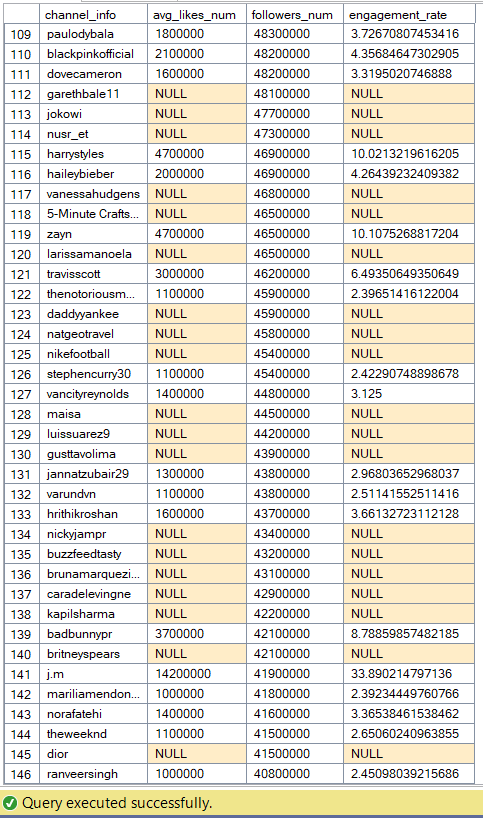
(TRY\_CAST(REPLACE(avg\_likes, 'm', '') AS FLOAT) \* 1000000) /

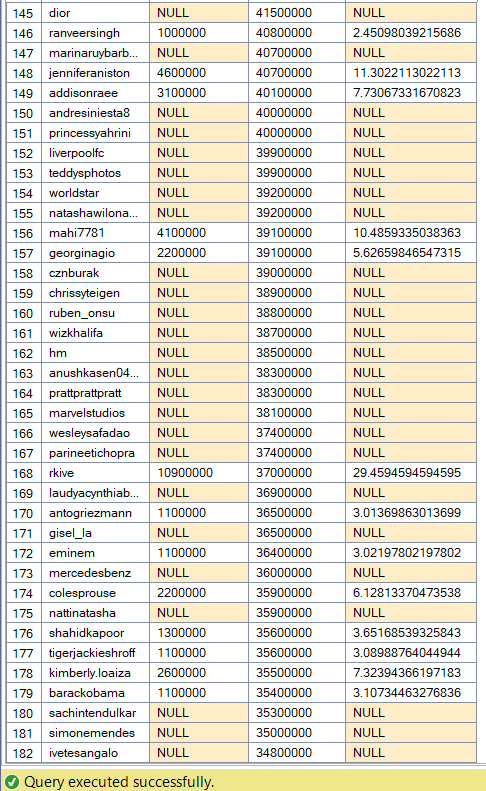
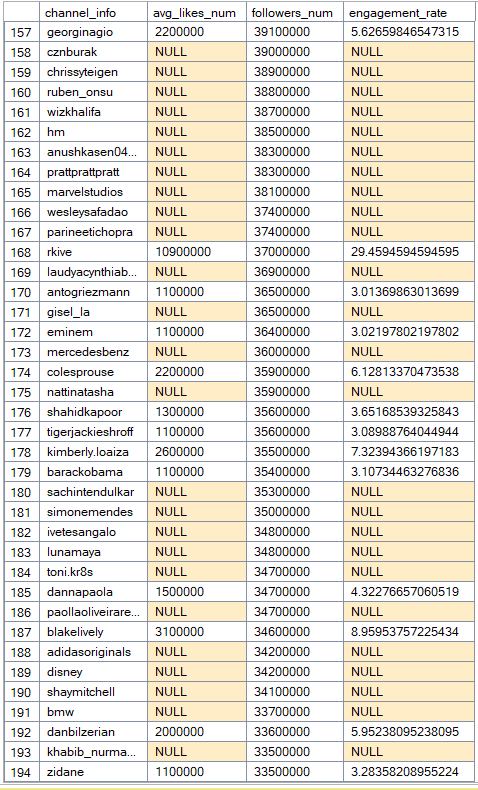
(TRY\_CAST(REPLACE(followers, 'm', '') AS FLOAT) \* 1000000) \* 100 AS engagement\_rate

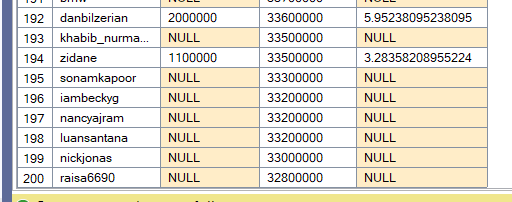
FROM influencers

WHERE avg\_likes IS NOT NULL AND followers IS NOT NULL;









**🔹 2. Growth Rate in New Post Likes**

**Formula:** ([new\_post\_avg\_like] - [avg\_likes]) / [avg\_likes] \* 100

This shows whether their likes are increasing or decreasing in recent posts.

**✅ SQL Query:**

SELECT

channel\_info,

TRY\_CAST(REPLACE(new\_post\_avg\_like, 'm', '') AS FLOAT) \* 1000000 AS new\_likes,

TRY\_CAST(REPLACE(avg\_likes, 'm', '') AS FLOAT) \* 1000000 AS old\_likes,

((TRY\_CAST(REPLACE(new\_post\_avg\_like, 'm', '') AS FLOAT) -

TRY\_CAST(REPLACE(avg\_likes, 'm', '') AS FLOAT)) /

TRY\_CAST(REPLACE(avg\_likes, 'm', '') AS FLOAT)) \* 100 AS growth\_rate

FROM influencers

WHERE new\_post\_avg\_like IS NOT NULL AND avg\_likes IS NOT NULL;

OUTPUT

THIS IS [HYPER LINK](file:///C:\Users\91620\Desktop\2.pdf)

**🔹 3. Like-to-Follower Ratio**

**Formula:** total\_likes / followers

Used to understand long-term performance — total engagement relative to audience size.

**✅ SQL Query:**

SELECT

channel\_info,

TRY\_CAST(REPLACE(total\_likes, 'b', '') AS FLOAT) \* 1000000000 AS total\_likes\_num,

TRY\_CAST(REPLACE(followers, 'm', '') AS FLOAT) \* 1000000 AS followers\_num,

(TRY\_CAST(REPLACE(total\_likes, 'b', '') AS FLOAT) \* 1000000000) /

(TRY\_CAST(REPLACE(followers, 'm', '') AS FLOAT) \* 1000000) AS like\_to\_follower\_ratio

FROM influencers

WHERE total\_likes IS NOT NULL AND followers IS NOT NULL;

OUTPUT

THIS IS [HYPER LINK](file:///C:\Users\91620\Desktop\3.pdf)