CST4070 Week 19 Challenge Solution - SQL

Twitter Data Analysis in SQL

SQL Concepts Utilised:

1. Aggregate Functions (SUM, AVG, MIN, MAX, COUNT)
2. Joins
3. Window Functions (ROW_NUMBER(), RANK(), DENSE_RANK())
4. Common Table Expressions (CTE) using WITH clause
5. Filtering using WHERE and HAVING clauses
6. Ordering using ORDER BY clause
7. Pattern identification and analysis using LIKE keyword
8. CASE WHEN statements
9. Table and column Aliases
10. Relational and Arithmetic Operators

Flow of Analysis:

Dataset Preparation
Imported the followers, users and tweets tables.
Created follower_count table from followers tables, to get information about the number of followers for each user. \

FOLLOWER ANALYSIS:
1. Users with the highest number of followers
2. Follower count statistics (min, max, avg) by age group
3. Top 5 users in each age group with the highest number of followers
4. Users who follow back all their followers and their proportions
5. Whether young or older users generally follow users in their same age group
6. Top 10 most active users per age group
TWEETS ANALYSIS:
7. Total tweet count and average tweet count by age group. Percentage by which young people tweet more.
8. Proportions of total tweets with and without hashtags and by age group
9. Comparison of tweet lengths between young and old users
10. Longest and shortest tweet and the corresponding users
11. Tweets with funny sentiment (lol, haha, hehe, rofl, hahaha) and their distribution among different age groups
12. Proportion of tweets containing questions
13. Ratio of tweets containing links for different demographics
14. Top 10 most common words used in tweets and the age group that used these words
15. Top 10 most mentioned users in tweets
HASHTAGS ANALYSIS:
16. Top 10 hashtags (overall)
17. Popular hashtags among young users
18. Popular hashtags among old users
19. Most popular hashtags per day of the week
20. Most popular hashtags by month
TEMPORAL ANALYSIS:
21. Monthly tweet volume trends by age group
22. Comparison of tweet activity on weekdays vs weekends among different demographics
23. Most active hour for tweeting by age group
24. Average number of tweets per week by age group
25. Most active days for users with the highest followers

Insights from Analysis:

1. Five users are tied in the first spot for the highest number of followers, with 11 followers.
2. The average number of followers for young people is 4, which is slightly greater than the average number of old people, which is 3.
3. Only 16.61% of users follow back all their followers. i.e. 5181 users follow back all their followers, out of a total of 31185 users.
4. People prefer following their own age group people more (old-old = 33771, young-young = 62217). However, old people follow more young people, than young people following old people. i.e. 6125 old people follow young people. Whereas, only 2555 young people follow old people.
5. Young users (18003) tweet approximately 36.24% more than old users (13214).
6. Overall, only 4.09% of tweets had hashtags. The proportion of tweets that had hashtags were more in young people than old. Only, 2.57% of tweets used by old people had hashtags. In comparison, 5.22% of tweets used by young people had hashtags.
7. On an average, old people's tweets were longer (tweet length : 94) than young people (tweet length : 90).
8. The longest tweet belonged to user ShayMarie09 and the shortest tweet belonged to user yoyoskittles.
9. Young people tweet funny content more than twice as often as old people, indicating a strong preference for humor among younger users.
10. 9.96% of the total tweets contain questions. This indicates that a significant portion of users' tweets are intended to engage others, spark conversations, or seek information and feedback.
11. Young users include links in their tweets at a higher rate (6.57%) compared to older users (4.22%). This indicates that younger users are more likely to share external content, references, or promotional materials through their tweets.
12. In the most common words used in tweets, the words "music", "movies", and "love" suggests that younger users are likely to tweet about their personal interests, hobbies, and feelings. The words "family" suggests that older users may prioritize tweeting about their personal life and family-related topics.
13. Most of the top mentioned users are music artists or related to the music industry, indicating that music-related content is highly engaging on this platform.
14. Top mentioned hashtags shows the communities interest in music (#musicmonday), social engagement (#followfriday, #fb) and global events (#iranelection).
15. The hashtags popular among young people show their heavy interest in music (#music, #music4good). The hashtags popular among old people (#travel, #traveltuesday, #knitting) show their interest in travel, knitting and family.
16. Specific hashtags dominate particular days, like #musicmonday on Mondays and #followfriday on Fridays, suggesting thematic social media activities that users participate in.
17. The increasing use of #musicmonday from April (8) to June (300) suggests growing engagement with this hashtag over these months.
18. Both age groups show increasing tweet activity from April to June, with young users consistently tweeting more than older users. June shows the highest tweet activity for both age groups, indicating a potential seasonal peak in user engagement.
19. Young users are more active on both weekdays and weekends compared to older users, with a significant number of tweets on weekdays.

20. Weekend vs. Weekday Activity: Young users' tweet volume decreases by 15.68% on weekends, while older users' tweet volume only decreases by 2.22%, indicating that older users maintain a more consistent tweeting pattern throughout the week.
21. Young users are most active at night (10PM), while older users are most active early in the morning (6AM), reflecting different daily routines and peak engagement times.
22. Users with the highest followers are most active on weekends, particularly Sunday, suggesting that weekends are a prime time for engagement. There's still significant activity on other days like Tuesday

Dataset Preparation

```
# File location and type
file_location = "/FileStore/tables/tweets-11.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
tweets_df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(tweets_df)
```

Table   New result table: ON

	id	timestamp	text
1	8653	2009-04-06T21:21:5...	falling asleep. just heard about that tracy girl's body being found. how sad my heart breaks for that family.
2	12020	2009-04-06T21:22:4...	i have a sad feeling that dallas is not going to show up i gotta say though, you'd think more shows would use music from the game. mm...
3	23858	2009-04-06T21:25:5...	@statravelau just got ur newsletter, those fares really are unbelievable, shame i already booked and paid for mine
4	51844	2009-04-06T21:33:1...	@djalizay i really don't think people choose to be that way. but i think he chose not to accept my family's help he might be dead by now
5	52341	2009-04-06T21:33:2...	my mind and body are severely protesting this &quot;getting up&quot; thing. had nightmares to boot
6	82794	2009-04-06T21:41:2...	my thoughts are with sandra cantu's family at this difficult and sad time
7	84377	2009-04-06T21:41:5...	aww, sandra cantu is found dead in a suitcase missing children stories never seem to go good. poor family.
8	89668	2009-04-06T21:43:2...	stupid movies we watched... mirrors ugggggh... stooopeeed!!! rip off!
9	120701	2009-04-06T21:52:0...	&quot;on popular music&quot; by t.w.adorno is probably the most difficult reading ever prescribed, i'm actually struggling to continue
10	140666	2009-04-06T21:57:5...	poor sandra cantu &amp; the cantu family! my prayers go out to them! what a sick world we live in. she was only 8
11	159109	2009-04-06T22:02:5...	this earthquake in italy has me sadden. it's only three hours away from naples, where my family is
12	171912	2009-04-06T22:06:3...	@katebornstein which is pretty anti memorial tattoos but for all but the strictest, there's no official ban, just disapproving family
13	181838	2009-04-06T22:09:2...	i hate converting movies just to put em on my itouch
14	188426	2009-04-06T22:11:1...	just been playing with the new mobbler v0.4.0 and it adds some great new features, but won't play music on my e71, like v0.3.5 did
15	194477	2009-04-06T22:13:0...	heartbroken over little sandra. prayers are with the family.

10,000+ rows | Truncated data due to row limit

```
# File location and type
file_location = "/FileStore/tables/users-10.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
users_df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(users_df)
```

Table   New result table: ON

	id	user	age
1	8653	hpfangirl94	old
2	12020	HybridMink	young
3	23858	driveaway2008	old
4	51844	lennytoups	old
5	52341	Jemimus	young
6	82794	hotrodlopez	old
7	84377	jenners101	old
8	89668	tracious	young
9	120701	calee01	young
10	140666	3WildBoys	old
11	159109	michellepolus	old
12	171912	A_Gael	old
13	181838	Huddy1124	young
14	188426	gerrymoth	young
15	194477	elohveee	old

10,000+ rows | Truncated data due to row limit

```
# File location and type
file_location = "/FileStore/tables/followers-9.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
followers_df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(followers_df)
```

Table

New result table: ON

	id	following
1	210569926	8653
2	709656306	8653
3	496010887	12020
4	544857586	12020
5	576431861	12020
6	707338535	12020
7	734089526	12020
8	412768991	13249
9	501230253	13249
10	522845974	13249
11	357662864	23858
12	600933485	23858
13	728061234	23858
14	530378660	33294
15	594162935	33294

10,000+ rows | Truncated data due to row limit

```
# Create a view or table

temp_table_name = "tweets"

tweets_df.createOrReplaceTempView(temp_table_name)

temp_table_name = "users"

users_df.createOrReplaceTempView(temp_table_name)

temp_table_name = "followers"

followers_df.createOrReplaceTempView(temp_table_name)
```

```
# Remove the existing directory if already exists
# performed to avoid errors in table creation while rerunning the notebook
dbutils.fs.rm("dbfs:/user/hive/warehouse/follower_count", recurse=True)
```

```
Out[7]: True
```

New Table Creation follower_count from followers table

```
%sql
-- Create follower_count table from followers table
CREATE TABLE IF NOT EXISTS follower_count AS
SELECT
    following AS user_id,
    COUNT(following) AS total_followers
FROM followers
GROUP BY following;
```

Query returned no results

```
%sql
-- Display follower_count table
SELECT *
FROM follower_count
ORDER BY total_followers
LIMIT 10;
```

Table

New result table: ON

	user_id	total_followers
1	529311560	1
2	780951940	1
3	580433400	1
4	217690947	1
5	582230907	1
6	89109524	1
7	589105801	1
8	594428561	1
9	743149576	1
10	598344714	1

10 rows

SQL Queries for Analysis:

FOLLOWERS ANALYSIS:

1. Users with the Highest Number of Followers:

```
%sql
-- users with the highest number of followers
WITH RankedFollowers AS (
    SELECT
        fc.user_id,
        fc.total_followers,
        u.user,
        DENSE_RANK() OVER (ORDER BY fc.total_followers DESC) AS rank
    FROM follower_count fc
    JOIN users u ON fc.user_id = u.id
)
SELECT
    user_id,
    user AS username,
    total_followers
FROM RankedFollowers
WHERE rank = 1;
```

Table   New result table: ON

	user_id	username	total_followers
1	503356722	gingerssnap	11
2	594082718	PatzIsDoomed	11
3	89312508	crumpet	11
4	517210582	VioletsCRUK	11
5	571862346	OmgitsJenna	11

5 rows

Inference:

Five users are tied in the first spot for highest number of followers, with 11 followers.

2. Followers Statistics:

```
%sql

-- Follower count statistics (min, max, avg) by age group

SELECT
    u.age,
    MIN(fc.total_followers) AS min_followers,
    MAX(fc.total_followers) AS max_followers,
    ROUND(AVG(fc.total_followers), 0) AS avg_followers
FROM follower_count fc
JOIN users u ON fc.user_id = u.id
GROUP BY u.age;
```

Table   New result table: ON

	age	min_followers	max_followers	avg_followers
1	old	1	11	3
2	young	1	11	4

2 rows

Inference:

The average number of followers for young people is 4, which is slightly greater than the average number of old people, which is 3.

3. Top 5 users with the most followers by age group

```
%sql

-- Top 5 users with the most followers by age group

WITH RankedFollowers AS (
    SELECT
        u.id,
        u.age,
        u.user,
        fc.total_followers,
        RANK() OVER (PARTITION BY u.age ORDER BY fc.total_followers DESC) AS rank
    FROM follower_count fc
    JOIN Users u ON fc.user_id = u.id
)
SELECT
    id AS user_id,
    user AS username,
    age,
    rank
FROM RankedFollowers
WHERE rank <= 5;
```

Table   New result table: ON

	user_id	username	age	rank
1	89312508	crumpet	old	1
2	413396969	CuzDaddySaidSo	old	2
3	520910418	DemiCyrus	old	2
4	584931117	srinitata	old	2
5	750222004	cnicolio	old	2
6	89511433	dizz02	old	2
7	465302286	AtomicShroom	old	2
8	488351162	bmichalk	old	2
9	596028868	velvet_grooves	old	2
10	503356722	gingerssnap	young	1
11	594082718	PatzIsDoomed	young	1
12	517210582	VioletsCRUK	young	1
13	571862346	OmgitsJenna	young	1
14	367521958	jodiekearns	young	5
15	495675668	P0rC3lainTrAmP	young	5

40 rows

Reason for 40 rows in the above query is due to the tie in rankings of users with the same number of followers.

The ties can be seen in the query below

```
%sql
-- Check for ties in rankings within each age group
WITH RankedFollowers AS (
    SELECT
        u.id,
        u.age,
        fc.total_followers,
        RANK() OVER (PARTITION BY u.age ORDER BY fc.total_followers DESC) AS rank
    FROM follower_count fc
    JOIN Users u ON fc.user_id = u.id
)
SELECT
    age,
    rank,
    total_followers,
    COUNT(*) AS count
FROM RankedFollowers
GROUP BY age, rank, total_followers
ORDER BY age, rank;
```

Table   New result table: ON

	age	rank	total_followers	count
1	old	1	11	1
2	old	2	10	8
3	old	10	9	15
4	old	25	8	81
5	old	106	7	253
6	old	359	6	606
7	old	965	5	1362
8	old	2327	4	2355
9	old	4682	3	3182
10	old	7864	2	3002
11	old	10866	1	1835
12	young	1	11	4
13	young	5	10	27
14	young	32	9	89
15	young	121	8	254

22 rows

4. Users who follow back all their followers:

```
%sql
-- Users who follow back all their followers
WITH FollowerCounts AS (
    SELECT
        following AS user_id,
        COUNT(id) AS num_followers
    FROM Followers
    GROUP BY following
),
FollowingCounts AS (
    SELECT
        id AS user_id,
        COUNT(following) AS num_following
    FROM Followers
    GROUP BY id
),
FollowBackUsers AS (
    SELECT
        fc.user_id
    FROM FollowerCounts fc
    JOIN FollowingCounts fgc ON fc.user_id = fgc.user_id
    WHERE fc.num_followers = fgc.num_following
)
SELECT
    u.id AS user_id,
    u.user
FROM FollowBackUsers fbu
JOIN Users u ON fbu.user_id = u.id;
```

Table

New result table: ON

	user_id	user
1	91290149	dani_ellee
2	92031363	fetchmp3
3	209356608	bitoclass
4	218078424	Rose_H
5	286982617	ThatAngelGirl
6	359314984	bergenlarsen
7	412186055	daaym_mimi
8	414809099	BobCaton
9	496479516	jasonhurwitz
10	505160172	nmr71886
11	530563593	JackShockley
12	533605444	stephmcg
13	545738162	Netra
14	583111593	linalrae
15	598344714	scyrene

5,181 rows

Proportion of users who follow back all their followers

```
%sql
-- Users who follow back all their followers, proportion of such users, and total number of users
WITH FollowerCounts AS (
    SELECT
        following AS user_id,
        COUNT(id) AS num_followers
    FROM Followers
    GROUP BY following
),
FollowingCounts AS (
    SELECT
        id AS user_id,
        COUNT(following) AS num_following
    FROM Followers
    GROUP BY id
),
FollowBackUsers AS (
    SELECT
        fc.user_id
    FROM FollowerCounts fc
    JOIN FollowingCounts fgc ON fc.user_id = fgc.user_id
    WHERE fc.num_followers = fgc.num_following
),
FollowBackUserDetails AS (
    SELECT
        u.id AS user_id,
        u.user
    FROM FollowBackUsers fbu
    JOIN Users u ON fbu.user_id = u.id
)
SELECT
    COUNT(*) AS follow_back_users,
    (SELECT COUNT(*) FROM Users) AS total_users,
    CONCAT(ROUND((COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Users)), 2),"%") AS proportion_of_follow_back_users
FROM FollowBackUserDetails;
```

Table

New result table: ON

	follow_back_users	total_users	proportion_of_follow_back_users
1	5181	31185	16.61%

1 row

Inference

Only 16.61% of users follow back all their followers. i.e. 5181 users follow back all their followers, out of a total of 31185 users.

5. Whether young or older users generally follow users in their same age group

i.e. follow pattern of young and old users

```
%sql
-- Whether young or older users generally follow users in their same age group
SELECT
    u1.age AS follower_age,
    u2.age AS followed_age,
    COUNT(*) AS count_of_relationship
FROM Followers f
JOIN Users u1 ON f.following = u1.id
JOIN Users u2 ON f.id = u2.id
GROUP BY u1.age, u2.age;
```

Table

New result table: ON

	follower_age	followed_age	count_of_relationship
1	old	young	6125
2	old	old	33771
3	young	young	62217
4	young	old	2555

4 rows

Inference:

People prefer following their own age group people more (old-old = 33771, young-young = 62217). However, old people follow more young people, than young people following old people. i.e. 6125 old people follow young people. Whereas, only 2555 young people follow old people.

6. Top 10 most active users per age group

```
%sql
-- Top 10 most active users per age group
WITH UserActivity AS (
    SELECT
        u.id,
        u.user,
        u.age,
        COUNT(*) AS tweet_count,
        ROW_NUMBER() OVER (PARTITION BY u.age ORDER BY COUNT(*) DESC) AS rank
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    GROUP BY u.id, u.user, u.age
)
SELECT
    id AS user_id,
    user AS username,
    age
FROM UserActivity
WHERE rank <= 10;
```

Table

New result table: ON

	user_id	username	age
1	518031975	minxkitty	old
2	498601874	WhoahItsTanisha	old
3	521809040	louiseisanelf	old
4	366219601	DaveParris	old
5	534688550	rhueladams	old
6	497791396	Fire_at_will_xo	old
7	547692907	KymbaKat	old
8	585494868	ffmpaD	old
9	217899625	evAllTimeLow	old
10	598581868	hollyannaeree	old
11	515479964	ashleybella	young
12	719921443	Nidiamazing	young
13	580473461	Zwinky101	young
14	722063733	alex_and_brooke	young
15	544502662	toastylileskimo	young

20 rows

TWEETS ANALYSIS:

7. Total number of tweets and average number of tweets by age group. Percentage by which young people tweet more

```
%sql
-- Tweet count by age group
SELECT
    age,
    COUNT(*) AS tweet_count
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY age;
```

Table

New result table: ON

	age	tweet_count
1	old	13214
2	young	18003

2 rows

Percentage by which young people tweet more than old people

```
%sql
-- Percentage by which young people tweet more than old people
WITH TweetCounts AS (
    SELECT
        u.age,
        COUNT(*) AS tweet_count
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    GROUP BY u.age
),
YoungTweetCount AS (
    SELECT tweet_count FROM TweetCounts WHERE age = 'young'
),
OldTweetCount AS (
    SELECT tweet_count FROM TweetCounts WHERE age = 'old'
)
SELECT
    (SELECT tweet_count FROM YoungTweetCount) AS young_tweet_count,
    (SELECT tweet_count FROM OldTweetCount) AS old_tweet_count,
    ROUND(((SELECT tweet_count FROM YoungTweetCount) - (SELECT tweet_count FROM OldTweetCount)) * 100.0 / (SELECT tweet_count FROM OldTweetCount), 2) AS percentage_increase
FROM TweetCounts
LIMIT 1;
```

Table

New result table: ON

	young_tweet_count	old_tweet_count	percentage_increase
1	18003	13214	36.24

1 row

Inference:

Young users (18003) tweet approximately 36.24% more than old users (13214).

```
%sql
-- Average tweet count by age group
WITH UserTweetCounts AS (
    SELECT
        id,
        COUNT(id) AS tweet_count
    FROM Tweets
    GROUP BY id
)
SELECT
    u.age,
    ROUND(AVG(utc.tweet_count),0) AS avg_tweet_count
FROM UserTweetCounts utc
JOIN Users u ON utc.id = u.id
GROUP BY u.age;
```

Table

New result table: ON

	age	avg_tweet_count
1	old	1
2	young	1

2 rows

8. Proportions of Tweets with and without hashtags/ and by age group

```
%sql
-- Proportions of tweets with and without hashtags
WITH HashtagCounts AS (
    SELECT
        SUM(CASE WHEN t.text LIKE '%#%' THEN 1 ELSE 0 END) AS tweets_with_hashtags,
        SUM(CASE WHEN t.text NOT LIKE '%#%' THEN 1 ELSE 0 END) AS tweets_without_hashtags,
        COUNT(*) AS total_tweets
    FROM Tweets t
    JOIN Users u ON t.id = u.id
)
SELECT
    tweets_with_hashtags,
    tweets_without_hashtags,
    CONCAT(ROUND((tweets_with_hashtags / total_tweets * 100),2), "%") AS proportion_of_tweets_with_hashtags,
    CONCAT(ROUND((tweets_without_hashtags / total_tweets * 100), 2), "%") AS proportion_of_tweets_without_hashtags
FROM HashtagCounts;
```

Table

New result table: ON

	tweets_with_hashtags	tweets_without_hashtags	proportion_of_tweets_with_hashtags	proportion_of_tweets_without_hashtags
1	1278	29939	4.09%	95.91%

1 row

Proportions of tweets with and without hashtags by age group

```
%sql
-- Proportions of tweets with and without hashtags by age group
WITH HashtagCounts AS (
    SELECT
        u.age,
        SUM(CASE WHEN t.text LIKE '%#%' THEN 1 ELSE 0 END) AS tweets_with_hashtags,
        SUM(CASE WHEN t.text NOT LIKE '%#%' THEN 1 ELSE 0 END) AS tweets_without_hashtags,
        COUNT(*) AS total_tweets
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    GROUP BY u.age
)
SELECT
    age,
    tweets_with_hashtags,
    tweets_without_hashtags,
    CONCAT(ROUND((tweets_with_hashtags / total_tweets * 100), 2), "%") AS proportion_of_tweets_with_hashtags,
    CONCAT(ROUND((tweets_without_hashtags / total_tweets * 100), 2), "%") AS proportion_of_tweets_without_hashtags
FROM HashtagCounts;
```

Table

New result table: ON

	age	tweets_with_hashtags	tweets_without_hashtags	proportion_of_tweets_with_hashtags	proportion_of_tweets_without_hashtags
1	old	339	12875	2.57%	97.43%
2	young	939	17064	5.22%	94.78%

2 rows

Inference:

Overall, only 4.09% of tweets had hashtags. The proportion of tweets that had hashtags were more in young people than old. Only, 2.57% of tweets used by old people had hashtags. In comparison, 5.22% of tweets used by young people had hashtags.

9. Comparison of tweet lengths between young and old users

```
%sql
-- Comparison of tweet lengths between young and old users
SELECT
    u.age,
    ROUND(AVG(LENGTH(t.text)),0) AS average_tweet_length
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY u.age;
```

Table

New result table: ON

	age	average_tweet_length
1	old	94
2	young	90

2 rows

Inference:

10. Longest and Shortest Tweets and the corresponding users

```
%sql
-- Longest and shortest tweet and their corresponding users
WITH TweetLengths AS (
    SELECT
        t.id,
        t.text,
        LENGTH(t.text) AS tweet_length
    FROM Tweets t
),
LongestTweet AS (
    SELECT
        u.id,
        u.user AS username,
        t.text AS tweet,
        t.tweet_length
    FROM TweetLengths t
    JOIN Users u ON t.id = u.id
    ORDER BY t.tweet_length DESC
    LIMIT 1
),
ShortestTweet AS (
    SELECT
        u.id,
        u.user AS username,
        t.text AS tweet,
        t.tweet_length
    FROM TweetLengths t
    JOIN Users u ON t.id = u.id
    ORDER BY t.tweet_length ASC
    LIMIT 1
)
SELECT * FROM LongestTweet
UNION ALL
SELECT * FROM ShortestTweet;
```

Table

New result table: ON

	id	username	tweet	tweet_length
1	531406044	ShayMarie09	› #musicmonday &quot;i love you&quot; faith evans....&quot;something that i like&quot; ryan leslie.....mario &quot;good one&quot;......	182
2	509953696	yoyoskittles	movies	7

2 rows

Inference:

The longest tweet belonged to user ShayMarie09 and the shortest tweet belonged to user yoyoskittles.

11. Tweets with funny sentiment and their distribution among the different age groups

```
%sql
-- Tweets with funny sentiment by age group
SELECT
    u.age,
    t.id,
    t.text AS tweet
FROM Tweets t
JOIN Users u ON t.id = u.id
WHERE t.text ILIKE '%lol%'
   OR t.text ILIKE '%haha%'
   OR t.text ILIKE '%hehe%'
   OR t.text ILIKE '%rofl%'
   OR t.text ILIKE '%hahaha%';
```

Table

New result table: ON

	age	id	tweet
1	old	423359	guess what? my dad is pregnant!!! lol nah, the doctor does have to give him an epidural for his chronic back pain, though.
2	young	1554347	wow....wrote 4 pages in one hour, while playing around with music downloads...lol...i knew i could write the paper!!! now math hw
3	old	2190198	@royalbluestuey haha. little scared of the public stage eh? i love it. i like to travel, just don't get a chance to do it very often.
4	old	80991087	@ricklondon hehe..making more competition for myself! .. i'm going to try to pen something for your cartoon of mariel &amp; family
5	young	86325569	at the movies. seeing hannah montana. i couldnt convince malena to see anything else. lol
6	young	88316091	@musicislife45 hahahah that may be true..... unfortunately we're at 3oh!5 now though...
7	old	88324116	@alicejong haha... i was painting downstairs while watching family channel. now i don't know if your up and i am slightly bored as well..
8	young	88633419	@musicalee i'm a loser! lol
9	young	88718209	@beanznkornbread lol that's because yall are but seriously what are you people doing this week. i'm on mission music super hard!
10	young	88813225	@malareignz they are now hahaha. i blush easy what can i say. i was listening to your music then the flicks. roro got red cheeks
11	young	89293056	heyya im so borde my sound card is fucked so i cant listen to music btw by that i mean mcr lol
12	young	90752295	last day at blockbuster... no more free movies. owell i get my weekends back woot! lol
13	young	91611685	@zaidah1 hahaha okay cooolness i think @achtungmusic isnt on anymore
14	old	91638344	@aplusk but lost 20 lbs in the 21 days still bit my whole family in the process hahaha
15	old	91666680	@mileycyrus don't leave us haha i'm part of your fan family, okay? good look at the hm premiere in germany, god bless you, luv ya guurl..

1,986 rows

```
%sql
-- Count of tweets with funny sentiment by age group
SELECT
    u.age,
    COUNT(*) AS funny_tweet_count
FROM Tweets t
JOIN Users u ON t.id = u.id
WHERE t.text ILIKE '%lol%'
   OR t.text ILIKE '%haha%'
   OR t.text ILIKE '%hehe%'
   OR t.text ILIKE '%rofl%'
   OR t.text ILIKE '%hahaha%'
GROUP BY u.age;
```

Table

New result table: ON

	age	funny_tweet_count
1	old	645
2	young	1341

2 rows

Inference:

Young people tweet funny content more than twice as often as old people, indicating a strong preference for humor among younger users.

12. Proportion of tweets containing questions

```
%sql
-- Count of tweets with questions, total tweet count, and proportion in %
SELECT
    COUNT(*) AS total_tweet_count,
    SUM(CASE WHEN t.text ILIKE '%?%' THEN 1 ELSE 0 END) AS question_tweet_count,
    CONCAT(ROUND((SUM(CASE WHEN t.text ILIKE '%?%' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)), 2), "%") AS proportion_of_question_tweets
FROM Tweets t
JOIN Users u ON t.id = u.id;
```

Table

New result table: ON

	total_tweet_count	question_tweet_count	proportion_of_question_tweets
1	31217	3109	9.96%

1 row

Inference:

9.96% of the total tweets contain questions. This indicates that a significant portion of users' tweets are intended to engage others, spark conversations, or seek information and feedback.

13. Ratio of links in tweets for different demographics

```
%sql
-- Ratio of tweets containing links by age group

SELECT
    u.age,
    COUNT(*) AS total_tweets_by_age,
    SUM(CASE WHEN t.text LIKE '%http%' THEN 1 ELSE 0 END) AS tweets_with_links,
    ROUND((SUM(CASE WHEN t.text LIKE '%http%' THEN 1 ELSE 0 END) * 1.0 / COUNT(*) *100 ),2) AS link_ratio
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY u.age;
```

Table

New result table: ON

	age	total_tweets_by_age	tweets_with_links	link_ratio
1	old	13214	557	4.22
2	young	18003	1182	6.57

2 rows

Inference:

Young users include links in their tweets at a higher rate (6.57%) compared to older users (4.22%). This indicates that younger users are more likely to share external content, references, or promotional materials through their tweets.

14. Top 10 most common words used in tweets and the age group that used these words

```
%sql

-- Top 10 most common words used excluding stop words

WITH StopWords AS (
    SELECT 'at' AS word UNION ALL
    SELECT 'the' UNION ALL
    SELECT 'i' UNION ALL
    SELECT 'to' UNION ALL
    SELECT 'in' UNION ALL
    SELECT 'on' UNION ALL
    SELECT 'and' UNION ALL
    SELECT 'or' UNION ALL
    SELECT 'a' UNION ALL
    SELECT 'is' UNION ALL
    SELECT 'of' UNION ALL
    SELECT 'it' UNION ALL
    SELECT 'for' UNION ALL
    SELECT 'with' UNION ALL
    SELECT 'as' UNION ALL
    SELECT 'this' UNION ALL
    SELECT 'by' UNION ALL
    SELECT 'that' UNION ALL
    SELECT 'an' UNION ALL
    SELECT 'be' UNION ALL
    SELECT 'are' UNION ALL
    SELECT 'was' UNION ALL
    SELECT 'were' UNION ALL
    SELECT 'which' UNION ALL
    SELECT 'but' UNION ALL
    SELECT 'if' UNION ALL
    SELECT 'you' UNION ALL
    SELECT 'not' UNION ALL
    SELECT 'we' UNION ALL
    SELECT 'they' UNION ALL
    SELECT 'from' UNION ALL
    SELECT 'at' UNION ALL
    SELECT 'he' UNION ALL
    SELECT 'she' UNION ALL
    SELECT 'has' UNION ALL
    SELECT 'have' UNION ALL
    SELECT 'had' UNION ALL
    SELECT 'will' UNION ALL
    SELECT 'shall' UNION ALL
    SELECT 'can' UNION ALL
    SELECT 'do' UNION ALL
    SELECT 'does' UNION ALL
    SELECT 'did' UNION ALL
    SELECT 'done'
),
CleanedTweets AS (
    SELECT
        id,
        LOWER(TRIM(REGEXP_REPLACE(text, '[^a-zA-Z\\s]', ''))) AS cleaned_text
    FROM Tweets
),
Words AS (
    SELECT
        id,
        EXPLODE(SPLIT(cleaned_text, '\\s+')) AS word
    FROM CleanedTweets
),
FilteredWords AS (
    SELECT
        w.id,
        w.word
    FROM Words w
    LEFT JOIN StopWords sw ON w.word = sw.word
    WHERE sw.word IS NULL AND w.word <> ''
)
SELECT
    w.word,
    COUNT(*) AS word_count,
    u.age
FROM FilteredWords w
JOIN Users u ON w.id = u.id
GROUP BY u.age, w.word
ORDER BY word_count DESC
LIMIT 10;
```

Table

New result table: ON

	word	word_count	age
1	music	8515	young
2	family	7317	old
3	my	4846	old
4	my	3959	young
5	movies	3476	young
6	im	1958	young
7	me	1930	young
8	so	1709	young
9	love	1575	young
10	just	1540	young

10 rows

Inference:

15. Top 10 most mentioned users in tweets

```
%sql
-- Top 10 most mentioned users in tweets
WITH Mentions AS (
    SELECT
        EXPLODE(SPLIT(t.text, ' ')) AS word
    FROM Tweets t
),
MentionCounts AS (
    SELECT
        word AS mention,
        COUNT(*) AS count
    FROM Mentions
    WHERE word LIKE '@%' AND LENGTH(word) > 1 -- Exclude singular '@'
    GROUP BY word
)
SELECT
    mention AS user_mentioned,
    count AS number_of_mentions
FROM MentionCounts
ORDER BY count DESC
LIMIT 10;
```

Table

New result table: ON

	user_mentioned	number_of_mentions
1	@ddlovato	127
2	@krisallenmusic	117
3	@mcflymusic	99
4	@mileycyrus	97
5	@alexandramusic	78
6	@jonasbrothers	55
7	@xomusicloverxo	54
8	@tommcfly	54
9	@xthemusic	49
10	@taylorswift13	49

10 rows

Inference:

Most of the top mentioned users are music artists or related to the music industry, indicating that music-related content is highly engaging on this platform.

HASHTAG ANALYSIS:

16. Top 10 most popular hastags:

```
%sql
-- Find the top 10 hashtags
WITH Hashtags AS (
    -- Extract hashtags from the text
    SELECT
        id,
        EXPLODE(SPLIT(text, ' ')) AS word
    FROM Tweets
),
FilteredHashtags AS (
    -- Filter out only the words that start with a hashtag
    SELECT
        word AS hashtag
    FROM Hashtags
    WHERE word LIKE '#%'
)
SELECT
    hashtag,
    COUNT(*) AS hashtag_count
FROM FilteredHashtags
GROUP BY hashtag
ORDER BY hashtag_count DESC
LIMIT 10;
```

Table

New result table: ON

	hashtag	hashtag_count
1	#musicmonday	397
2	#music	68
3	#followfriday	45
4	#fb	43
5	#familyforce5	32
6	#iranelection	27
7	#travel	15
8	#ff	14
9	#myweakness	13
10	#music4good	11

10 rows

Inference:

Top mentioned hashtags shows the communities interest in music (#musicmonday), social engagement (#followfriday, #fb) and global events (#iranelection).

17. Hashtags popular among young users

```
%sql
-- Find the top 10 hashtags used by younger users
WITH YoungerUsers AS (
    SELECT id
    FROM Users
    WHERE age = 'young'
),
Hashtags AS (
    -- Extract hashtags from the tweet text
    SELECT
        t.id,
        EXPLODE(SPLIT(t.text, ' ')) AS word
    FROM Tweets t
    JOIN YoungerUsers yu ON t.id = yu.id
),
FilteredHashtags AS (
    -- Filter out only the words that start with a hashtag
    SELECT
        word AS hashtag
    FROM Hashtags
    WHERE word LIKE '#%'
)
SELECT
    hashtag,
    COUNT(*) AS hashtag_count
FROM FilteredHashtags
GROUP BY hashtag
ORDER BY hashtag_count DESC
LIMIT 10;
```

Table   New result table: ON

	hashtag	hashtag_count
1	#musicmonday	397
2	#music	68
3	#followfriday	26
4	#iranelection	26
5	#myweakness	12
6	#music4good	11
7	#fb	11
8	#	9
9	#1	8
10	#inappropriatemovi...	8

10 rows

18. Hashtags Popular among old users

```
%sql
-- Find the top 10 hashtags used by older users
WITH OlderUsers AS (
    SELECT id
    FROM Users
    WHERE age = 'old'
),
Hashtags AS (
    -- Extract hashtags from the tweet text
    SELECT
        t.id,
        EXPLODE(SPLIT(t.text, ' ')) AS word
    FROM Tweets t
    JOIN OlderUsers ou ON t.id = ou.id
),
FilteredHashtags AS (
    -- Filter out only the words that start with a hashtag
    SELECT
        word AS hashtag
    FROM Hashtags
    WHERE word LIKE '#%'
)
SELECT
    hashtag,
    COUNT(*) AS hashtag_count
FROM FilteredHashtags
GROUP BY hashtag
ORDER BY hashtag_count DESC
LIMIT 10;
```

Table   New result table: ON

	hashtag	hashtag_count
1	#fb	32
2	#familyforce5	32
3	#followfriday	19
4	#travel	15
5	#ff	6
6	#30secondstom...	6
7	#family	5
8	#traveltuesday	5

9	#bsb	4
10	#knitting	4

10 rows

Inference:

The hashtags popular among young people show their heavy interest in music (#music, #music4good). The hashtags popular among old people (#travel, #traveltuesday, #knitting) show their interest in travel, knitting and family.

19. Most popular hashtags by day of week

```
%sql
-- Find the most popular hashtag per day of the week
WITH Hashtags AS (
    -- Extract hashtags from the tweet text and get the day of the week
    SELECT
        t.id,
        EXPLODE(SPLIT(t.text, ' ')) AS word,
        CASE
            WHEN DAYOFWEEK(t.timestamp) = 1 THEN 'Sunday'
            WHEN DAYOFWEEK(t.timestamp) = 2 THEN 'Monday'
            WHEN DAYOFWEEK(t.timestamp) = 3 THEN 'Tuesday'
            WHEN DAYOFWEEK(t.timestamp) = 4 THEN 'Wednesday'
            WHEN DAYOFWEEK(t.timestamp) = 5 THEN 'Thursday'
            WHEN DAYOFWEEK(t.timestamp) = 6 THEN 'Friday'
            ELSE 'Saturday'
        END AS day_of_week
    FROM Tweets t
),
FilteredHashtags AS (
    -- Filter out only the words that start with a hashtag
    SELECT
        day_of_week,
        word AS hashtag
    FROM Hashtags
    WHERE word LIKE '#%'
),
RankedHashtags AS (
    -- Rank the hashtags based on their counts for each day of the week
    SELECT
        day_of_week,
        hashtag,
        COUNT(*) AS hashtag_count,
        ROW_NUMBER() OVER (PARTITION BY day_of_week ORDER BY COUNT(*) DESC) AS rank
    FROM FilteredHashtags
    GROUP BY day_of_week, hashtag
)
-- Select the top-ranked hashtag for each day of the week
SELECT
    day_of_week,
    hashtag,
    hashtag_count
FROM RankedHashtags
WHERE rank = 1
ORDER BY
    CASE
        WHEN day_of_week = 'Sunday' THEN 1
        WHEN day_of_week = 'Monday' THEN 2
        WHEN day_of_week = 'Tuesday' THEN 3
        WHEN day_of_week = 'Wednesday' THEN 4
        WHEN day_of_week = 'Thursday' THEN 5
        WHEN day_of_week = 'Friday' THEN 6
        ELSE 7
    END;
```

Table

New result table: ON

	day_of_week	hashtag	hashtag_count
1	Sunday	#familyforce5	25
2	Monday	#musicmonday	374
3	Tuesday	#inappropriatemovi...	8
4	Wednesday	#bsb	5
5	Thursday	#followfriday	6
6	Friday	#followfriday	35
7	Saturday	#fb	13

7 rows

Inference:

Specific hashtags dominate particular days, like #musicmonday on Mondays and #followfriday on Fridays, suggesting thematic social media activities that users participate in.

20. Most Popular Hashtag by month

```
%sql
-- Find the most popular hashtag per month
WITH Hashtags AS (
    -- Extract hashtags from the tweet text and get the month name
    SELECT
        t.id,
        EXPLODE(SPLIT(t.text, ' ')) AS word,
        CASE
            WHEN MONTH(t.timestamp) = 1 THEN 'January'
            WHEN MONTH(t.timestamp) = 2 THEN 'February'
            WHEN MONTH(t.timestamp) = 3 THEN 'March'
            WHEN MONTH(t.timestamp) = 4 THEN 'April'
            WHEN MONTH(t.timestamp) = 5 THEN 'May'
            WHEN MONTH(t.timestamp) = 6 THEN 'June'
            WHEN MONTH(t.timestamp) = 7 THEN 'July'
            WHEN MONTH(t.timestamp) = 8 THEN 'August'
            WHEN MONTH(t.timestamp) = 9 THEN 'September'
            WHEN MONTH(t.timestamp) = 10 THEN 'October'
            WHEN MONTH(t.timestamp) = 11 THEN 'November'
            ELSE 'December'
        END AS month_name
    FROM Tweets t
),
FilteredHashtags AS (
    -- Filter out only the words that start with a hashtag
    SELECT
        month_name,
        word AS hashtag
    FROM Hashtags
    WHERE word LIKE '#%'
),
RankedHashtags AS (
    -- Rank the hashtags based on their counts for each month
    SELECT
        month_name,
        hashtag,
        COUNT(*) AS hashtag_count,
        ROW_NUMBER() OVER (PARTITION BY month_name ORDER BY COUNT(*) DESC) AS rank
    FROM FilteredHashtags
    GROUP BY month_name, hashtag
)
-- Select the top-ranked hashtag for each month
SELECT
    month_name,
    hashtag,
    hashtag_count
FROM RankedHashtags
WHERE rank =1
ORDER BY hashtag_count;
```

Table New result table: ON

	month_name	hashtag	hashtag_count
1	April	#musicmonday	8
2	May	#musicmonday	89
3	June	#musicmonday	300

3 rows

Inference:

The increasing use of #musicmonday from April (8) to June (300) suggests growing engagement with this hashtag over these months.

TEMPORAL ANALYSIS

21. Monthly tweet volume trends by age group

```
%sql
-- Monthly tweet volume trends by age group
SELECT
    u.age,
    YEAR(t.timestamp) AS year,
    CASE
    WHEN MONTH(t.timestamp) = 1 THEN 'January'
    WHEN MONTH(t.timestamp) = 2 THEN 'February'
    WHEN MONTH(t.timestamp) = 3 THEN 'March'
    WHEN MONTH(t.timestamp) = 4 THEN 'April'
    WHEN MONTH(t.timestamp) = 5 THEN 'May'
    WHEN MONTH(t.timestamp) = 6 THEN 'June'
    WHEN MONTH(t.timestamp) = 7 THEN 'July'
    WHEN MONTH(t.timestamp) = 8 THEN 'August'
    WHEN MONTH(t.timestamp) = 9 THEN 'September'
    WHEN MONTH(t.timestamp) = 10 THEN 'October'
    WHEN MONTH(t.timestamp) = 11 THEN 'November'
    ELSE 'December'
    END AS month_name,
    COUNT(*) AS tweet_count
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY u.age, YEAR(t.timestamp), MONTH(t.timestamp)
ORDER BY
    u.age,
    YEAR(t.timestamp), MONTH(t.timestamp);
```

Table New result table: ON

	age	year	month_name	tweet_count
1	old	2009	April	847
2	old	2009	May	5215
3	old	2009	June	7152
4	young	2009	April	1206
5	young	2009	May	7009
6	young	2009	June	9788

6 rows

Inference:

Both age groups show increasing tweet activity from April to June, with young users consistently tweeting more than older users. June shows the highest tweet activity for both age groups, indicating a potential seasonal peak in user engagement.

22. Average number of tweets per week by age group

```
%sql
-- Average number of tweets per week by age group
SELECT
    u.age,
    ROUND(COUNT(t.id) / COUNT(DISTINCT CONCAT(YEAR(t.timestamp), '-', WEEKOFYEAR(t.timestamp))), 0) AS avg_tweets_per_week
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY u.age;
```

Table   New result table: ON

	age	avg_tweets_per_week
1	old	1101
2	young	1500

2 rows

23. Comparison of tweet activity on weekdays and weekends among different demographic

```
%sql
-- Comparison of tweet activity on weekdays vs weekends by age group
SELECT
    u.age,
    CASE WHEN DAYOFWEEK(t.timestamp) IN (1, 7) THEN 'Weekend' ELSE 'Weekday' END AS day_type,
    COUNT(*) AS tweet_count
FROM Tweets t
JOIN Users u ON t.id = u.id
GROUP BY u.age, day_type;
```

Table   New result table: ON

	age	day_type	tweet_count
1	old	Weekend	6533
2	young	Weekend	8236
3	old	Weekday	6681
4	young	Weekday	9767

4 rows

Inference:

Young users are more active on both weekdays and weekends compared to older users, with a significant number of tweets on weekdays.

```
%sql
-- Percentage of activity change in weekends compared to weekdays by age group
WITH ActivityByDayType AS (
    SELECT
        u.age,
        CASE WHEN DAYOFWEEK(t.timestamp) IN (1, 7) THEN 'Weekend' ELSE 'Weekday' END AS day_type,
        COUNT(*) AS tweet_count
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    GROUP BY u.age, day_type
),
TotalActivity AS (
    SELECT
        age,
        SUM(CASE WHEN day_type = 'Weekday' THEN tweet_count ELSE 0 END) AS weekday_count,
        SUM(CASE WHEN day_type = 'Weekend' THEN tweet_count ELSE 0 END) AS weekend_count
    FROM ActivityByDayType
    GROUP BY age
)
SELECT
    age,
    ROUND((weekend_count - weekday_count) / weekday_count * 100,2) AS percentage_change
FROM TotalActivity;
```

Table

New result table: ON

	age	percentage_change
1	old	-2.22
2	young	-15.68

2 rows

Inference:

Weekend vs. Weekday Activity: Young users' tweet volume decreases by 15.68% on weekends, while older users' tweet volume only decreases by 2.22%, indicating that older users maintain a more consistent tweeting pattern throughout the week.

24. Most active hour for tweeting by age group

```
%sql
WITH TweetHours AS (
    SELECT
        u.age,
        HOUR(t.timestamp) AS hour,
        COUNT(*) AS tweet_count,
        RANK() OVER (PARTITION BY u.age ORDER BY COUNT(*) DESC) AS rank
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    GROUP BY u.age, HOUR(t.timestamp)
)
SELECT
    age,
    CASE
        WHEN hour = 0 THEN '12AM'
        WHEN hour < 12 THEN CONCAT(hour, 'AM')
        WHEN hour = 12 THEN '12PM'
        ELSE CONCAT(hour - 12, 'PM')
    END AS hour,
    tweet_count
FROM TweetHours
WHERE rank = 1;
```

Table

New result table: ON

	age	hour	tweet_count
1	old	6AM	755
2	young	10PM	1047

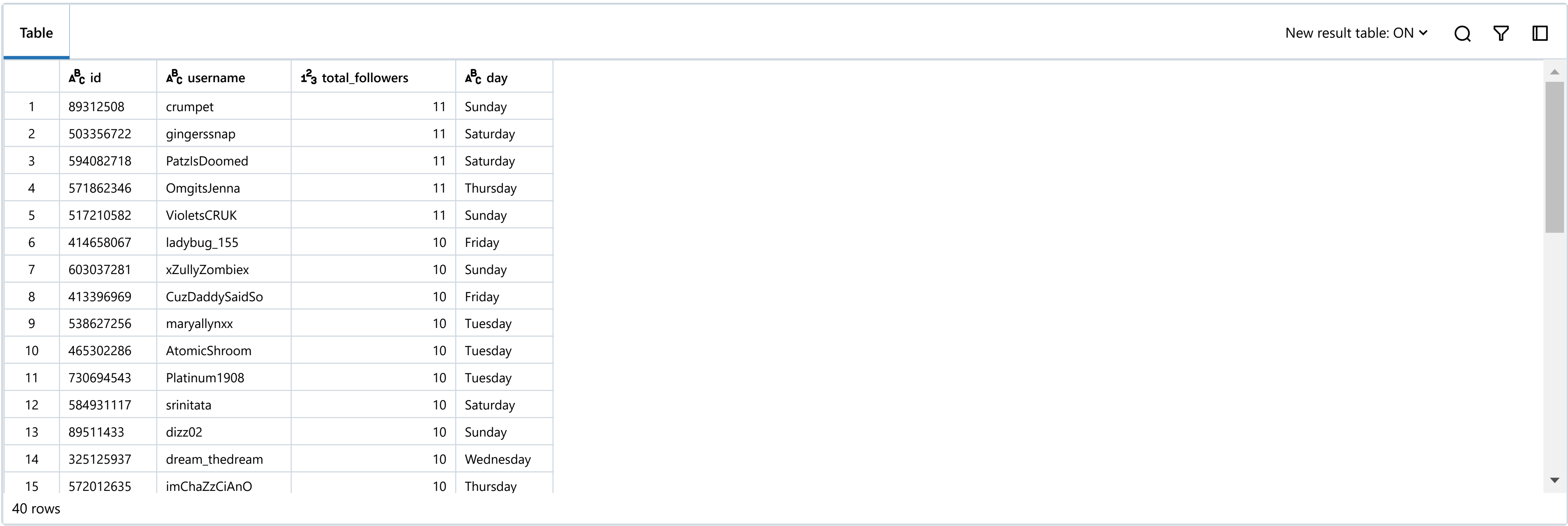
2 rows

Inference:

Young users are most active at night (10PM), while older users are most active early in the morning (6AM), reflecting different daily routines and peak engagement times.

25. Most active days for users with the highest followers

```
%sql
-- Most active days for users with the highest followers
WITH TopFollowers AS (
    SELECT
        fc.user_id,
        fc.total_followers,
        RANK() OVER (ORDER BY fc.total_followers DESC) AS follower_rank
    FROM follower_count fc
    JOIN Users u ON fc.user_id = u.id
),
UserActivity AS (
    SELECT
        u.id,
        u.user AS username,
        tf.total_followers,
        CASE
            WHEN DAYOFWEEK(t.timestamp) = 1 THEN 'Sunday'
            WHEN DAYOFWEEK(t.timestamp) = 2 THEN 'Monday'
            WHEN DAYOFWEEK(t.timestamp) = 3 THEN 'Tuesday'
            WHEN DAYOFWEEK(t.timestamp) = 4 THEN 'Wednesday'
            WHEN DAYOFWEEK(t.timestamp) = 5 THEN 'Thursday'
            WHEN DAYOFWEEK(t.timestamp) = 6 THEN 'Friday'
            WHEN DAYOFWEEK(t.timestamp) = 7 THEN 'Saturday'
            ELSE NULL
        END AS day
    FROM Tweets t
    JOIN Users u ON t.id = u.id
    JOIN TopFollowers tf ON u.id = tf.user_id
    WHERE tf.follower_rank <= 10
    GROUP BY u.id, u.user, tf.total_followers, day
)
SELECT
    id,
    username,
    total_followers,
    day
FROM UserActivity
ORDER BY total_followers DESC;
```



	id	username	total_followers	day
1	89312508	crumpet	11	Sunday
2	503356722	gingerssnap	11	Saturday
3	594082718	PatzIsDoomed	11	Saturday
4	571862346	OmgitsJenna	11	Thursday
5	517210582	VioletsCRUK	11	Sunday
6	414658067	ladybug_155	10	Friday
7	603037281	xZullyZombiex	10	Sunday
8	413396969	CuzDaddySaidSo	10	Friday
9	538627256	maryallynxx	10	Tuesday
10	465302286	AtomicShroom	10	Tuesday
11	730694543	Platinum1908	10	Tuesday
12	584931117	srinitata	10	Saturday
13	89511433	dizz02	10	Sunday
14	325125937	dream_thedream	10	Wednesday
15	572012635	imChaZzCiAnO	10	Thursday

40 rows

Inference:

Users with the highest followers are most active on weekends, particularly Sunday, suggesting that weekends are a prime time for engagement. There's still significant activity on other days like Tuesday and Thursday, indicating that high-followership users maintain a consistent presence throughout the week.