SQL queries to express the below questions:

* What user posted this tweet?
* When did the user post this tweet?
* What tweets have this user posted in the past 24 hours?
* How many tweets have this user posted in the past 24 hours?
* When did this user join Twitter?
* What keywords/ hashtags are popular?
* What tweets are popular?

Query 1:

```
SELECT DISTINCT users . user_id, users . screen_name, users . name
FROM users INNER JOIN tweets ON users . user_id = tweets . user_id
WHERE tweets . tweet_id = '1591558499627958272'
```

$$\delta$$

$$\pi_{\text{users . user\_id, users . screen\_name, users . name}}$$

$$\sigma_{\text{tweets . tweet\_id = "1591558499627958272"}} (\text{users} \bowtie_{\text{users . user\_id = tweets . user\_id}} \text{tweets})$$

Query 2::

```
SELECT DISTINCT tweets . created_at FROM users INNER JOIN tweets ON users . user_id =
tweets . user_id WHERE tweets . tweet_id = '1591558499627958272'
```

$$\delta$$

$$\pi_{\text{tweets . created\_at}}$$

$$\sigma_{\text{tweets . tweet\_id = "1591558499627958272"}} (\text{users} \bowtie_{\text{users . user\_id = tweets . user\_id}} \text{tweets})$$

Query 3:

```
SELECT DISTINCT tweets.tweet_id,tweets.text,users.user_id FROM tweets INNER JOIN
users ON users.user_id=tweets.user_id WHERE tweets.user_id=1491776087684104193 AND
tweets.created_at>'2022-11-11 22:06:01'
```

$$\delta$$

$$\pi_{\text{tweets . tweet\_id, tweets . text, users . user\_id}}$$

$$\sigma_{\text{tweets . user\_id = 1491776087684104193 AND tweets . created\_at > "2022-11-11 22:06:01"}} (\text{tweets} \bowtie_{\text{users . user\_id = tweets . user\_id}} \text{users})$$

Query 4:
SELECT users.user_id,COUNT(tweets.tweet_id) as number_of_tweets FROM tweets INNER JOIN users ON users.user_id=tweets.user_id WHERE tweets.user_id=1491776087684104193 AND tweets.created_at>'2022-11-11 22:06:01'

$\pi_{\text{users . user\_id, COUNT (tweet\_id)} \rightarrow \text{number\_of\_tweets}}$

$\gamma_{\text{COUNT (tweet\_id)}}$

$\sigma_{\text{tweets . user\_id = 1491776087684104193 AND tweets . created\_at > "2022-11-11 22:06:01"}} (\text{tweets} \bowtie_{\text{users . user\_id = tweets . user\_id}} \text{users})$

Query 5:
SELECT DISTINCT users.user_id,users.created_at FROM users WHERE users.user_id=1491776087684104193

δ

$\pi_{\text{users . user\_id, users . created\_at}}$

$\sigma_{\text{users . user\_id = 1491776087684104193}} \text{users}$

Query 6:
SELECT DISTINCT hashtags . text, tweets . retweet_count
FROM hashtags INNER JOIN tweets ON tweets . tweet_id = hashtags . tweet_ids
WHERE hashtags . retweet_count > 3

δ

$\pi_{\text{hashtags . text, tweets . retweet\_count}}$

$\sigma_{\text{hashtags . retweet\_count > 3}} (\text{hashtags} \bowtie_{\text{tweets . tweet\_id = hashtags . tweet\_ids}} \text{tweets})$

Query 7:
SELECT tweet_id, text, retweet_count FROM tweets ORDER BY retweet_count DESC

$\tau_{\text{retweet\_count} \downarrow}$

$\pi_{\text{tweet\_id, text, retweet\_count}} \text{tweets}$

Career Recommendation System : Use Cases

Use Case: User can look for opening for their target job position Description: User can look for opening for a position named "Engineer"

Actor: User

Precondition: User should have a valid target position name

Steps:

Actor action: User request for list of job openings for his target position.

System Responses: If the positon exists, the system will return a list of job openings posted.

Post Condition: List of job openings suggested

Alternate Path: The user request is not correct and system throws an error

Error: User information is incorrect

Use Case: User can look for openings posted by their dream company handle Description: Search for job posts posted by a particular user

Actor: User

Precondition: User should have a company name user is target

Steps:

Actor action: User request for list of job openings for his target position.

System Responses: If the company has posted job openings, the system will return the list.

Post Condition: List of job openings suggested

Alternate Path: The user request is not correct and system throws an error

Error: User information is incorrect

Use Case: User can look for openings posted within last 5 days and for a particular position
Description: Search for job posts posted within last 5 days

Actor: User

Precondition: User should have a valid target position name

Steps:

Actor action: User request for list of job openings for his target position.

System Responses: The system will return a list of job posts.

Post Condition: List of job openings suggested

Alternate Path: The user request is not correct and system throws an error

Error: User information is incorrect

Use Case: User can assess which job positions are more in demand Description: Search for job posts for different job positions

Actor: User

Precondition: User should have a valid target position name

Steps:

Actor action: User request for list of job openings for his target position.

System Responses: The system will return a count of job posts for a position.

Post Condition: List of job openings suggested

Alternate Path: The user request is not correct and system throws an error

Error: User information is incorrect

Use Case: User can assess which companies are posting more jobs

Description: Search for job posts for job positions by different companies

Actor: User

Precondition: User should have a valid target position name and target company

Steps:

Actor action: User request for list of job openings for his target position posted by copmany.

System Responses: The system will return a count of job posts posted by company handle.

Post Condition: Count of job openings suggested

Alternate Path: The user request is not correct and system throws an error

Error: User information is incorrect

Query 1:
SELECT jobs.job_title,jobs.description, jobs.poster, jobs.posted_at FROM jobs INNER JOIN tweets ON tweets.tweet_id=jobs.tweet_ids

$$\pi_{\text{jobs . job\_title, jobs . description, jobs . poster, jobs . posted\_at}} (\text{jobs} \bowtie_{\text{tweets . tweet\_id = jobs . tweet\_ids}} \text{tweets})$$

Query 2:
SELECT jobs.job_title,jobs.description, jobs.poster, jobs.posted_at FROM jobs WHERE poster='CareersInGovernment'

$$\pi_{\text{jobs . job\_title, jobs . description, jobs . poster, jobs . posted\_at}} \sigma_{\text{poster = "CareersInGovernment"}} \text{jobs}$$

Query 3:
SELECT jobs.job_title,jobs.description, jobs.poster, jobs.posted_at FROM jobs WHERE job_title='NURSE' and posted_at>'2022-11-11 22:28:10'

$\pi_{\text{jobs . job\_title, jobs . description, jobs . poster, jobs . posted\_at}}$
$\sigma_{\text{job\_title = "NURSE" AND posted\_at > "2022-11-11 22:28:10"}}$ jobs

Query 4:
SELECT jobs.job_title,COUNT(jobs.job_title) as number_of_postings,jobs.description, jobs.poster, jobs.posted_at FROM jobs GROUP BY job_title ORDER BY number_of_postings DESC

$\tau_{\text{number\_of\_postings} \downarrow}$
$\pi_{\text{jobs . job\_title, COUNT (job\_title)} \rightarrow \text{number\_of\_postings, jobs . description, jobs . poster, jobs . posted\_at}}$
$\gamma_{\text{job\_title, COUNT (job\_title)}}$ jobs

Query 5:
SELECT jobs.job_title,COUNT(jobs.job_title) as number_of_postings,jobs.description, jobs.poster, jobs.posted_at FROM jobs GROUP BY poster

$\pi_{\text{jobs . job\_title, COUNT (job\_title)} \rightarrow \text{number\_of\_postings, jobs . description, jobs . poster, jobs . posted\_at}}$
$\gamma_{\text{poster, COUNT (job\_title)}}$ jobs