Note: More than required fields are selected in SQL queries to provide context to the information retrieved.

Use case - 1

Use Case: Register a user in twitter database

Description: User is added in the user entity relationship.

Actor: User

Precondition: When a user tweets something about yelp, its basic information needs to be collected

Actor action: User request for adding user information.

System Responses: If user information is correct, it's added to the database, use case ends.

Post Condition: User information successfully added.

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

Error: User information is incorrect

SQL

```
INSERT INTO users
(user_id, username, location, created_at)
VALUES (112900604, "TheRealNekaRay", "Maryland, USA", '2010-02-10 02:18:13');
```

Relational Algebra

Relational algebra representation is not possible, since the operation needed in the use case is INSERT

Use Case -2

Use Case: Add a tweet information

Description: User makes a tweet with “#yelp”, its information is added to the database

Actors: User

Precondition: User must have a unique Twitter handle to tweet

Steps:

Actor action : User tweets about yelp review along with the yelp review URL

System Responses: The tweet information is collected and stored.

Post Condition: An tweet is added to hashtag table for the yelp review the user tweeted.

Alternate Path: There is no url to the yelp review.

Error: URL not available

SQL

INSERT INTO hashtag

(tweet_id, user_id, text, url, retweets, likes, time)

VALUES (1589845173340708865, 931223747985575936, 'Checkout we\'re providing #bulk #data of 3k plus #business #categories\n\n#plumbing #roofing #landscaping #realtors #hotels #restaurants #leads #leadsgeneration #emailslist #doctors #dentists #lawyers #socialmedia #python #java #php #yelp #googlemaps #ml #AI\n\nhttps://t.co/xO23syY7dI https://t.co/KnRSRphnmw', 'https://t.co/xO23syY7dI', 5, 5, '2022-11-08 05:00:01');

Relational Algebra

Relational algebra representation is not possible, since the operation needed in the use case is INSERT

Use Case - 3

Use Case: View the past tweet already tweeted by a user

Description: view a user's past tweets

Actors: User

Precondition: User must have made a previous tweet

Steps:

Actor action – User views previous tweets posted by a user from its username

System Responses – past user tweets will be displayed

Post Condition: user must have previous tweets

SQL

```
SELECT
users.username,
timeline.text,
timeline.created_at
FROM timeline
JOIN users ON timeline.user_id = users.user_id
WHERE username = "prplehaiz";
```

Relational Algebra

$\pi_{\text{timeline.text}}(\sigma_{\text{username = "prplehaiz"}}(\text{timeline.user\_id} \bowtie \text{users.user\_id}))$

Use Case - 4

Use Case: View the tweets which are most retweeted (more 5)

Description: User tweets the products above a particular price

Actor: User

Precondition:

Steps:

Actor action: User views the tweets above a particular retweets count.

System Responses: the list of tweets above a retweet count are displayed

Post Condition: system displays the list of tweets for the condition

SQL

```
SELECT
username,
text,
url,
likes,
retweets
FROM hashtag
JOIN users ON hashtag.user_id = users.user_id
WHERE likes >= 5 or retweets >= 5;
```

Relational Algebra

$\pi_{\text{hashtag.text}}(\sigma_{\text{username = "prplehaiz"}}(\text{hashtag.user\_id} \bowtie \text{users.user\_id}))$

Use Case - 5

Use Case: User account creation date

Description: User views the account the account creation date

Actor: User

Precondition: User must have a account created

Steps:

Actor action: User views the account creation date

System Responses: Displays the user account creation date

Alternate Path: There are no account created

Error: User account not found.

SQL

```
SELECT
username,
created_at AS account_created_date
FROM users
WHERE username = "cursortek";
```

Relational Algebra

ρ (account_created_date ,$\pi_{\text{created\_at}}$ ($\sigma_{\text{username = " cursortek "}}$(users)))