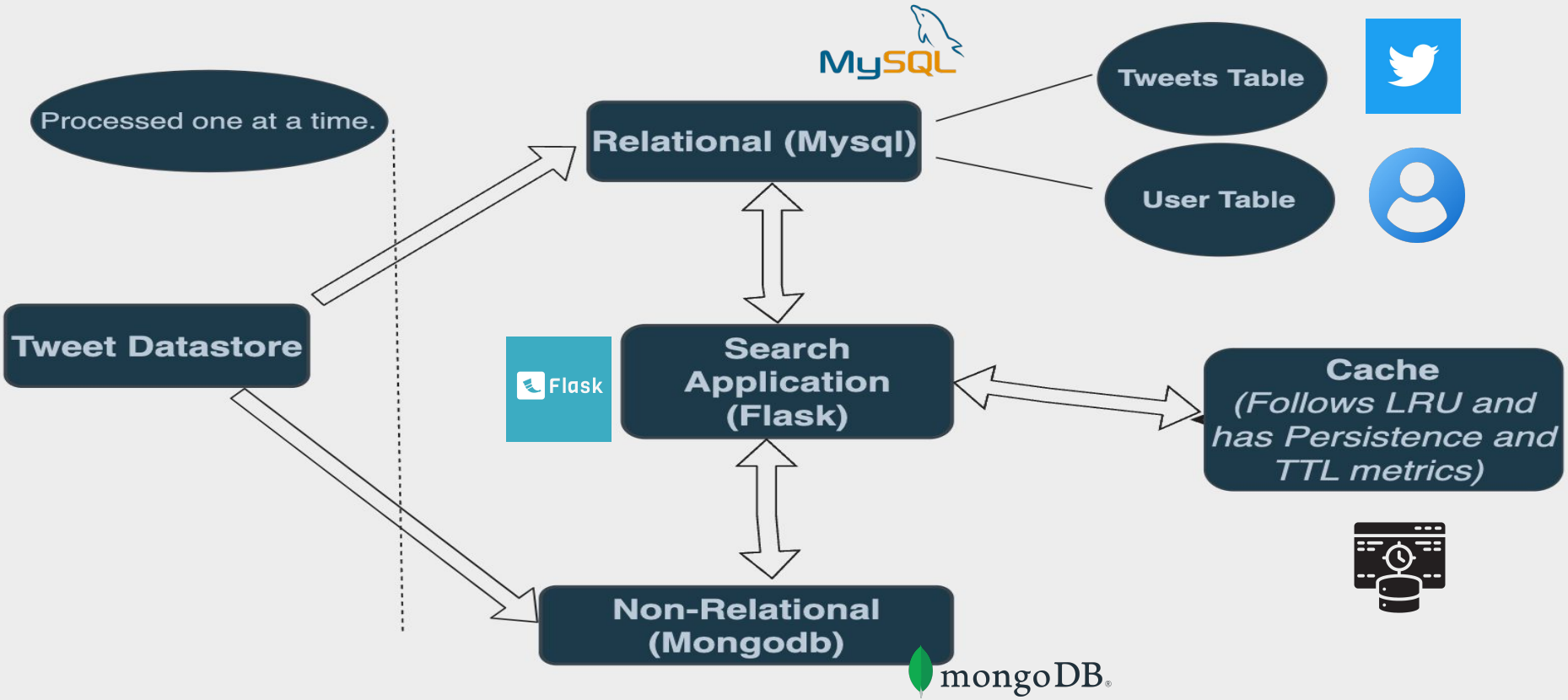
Final Project Presentation on Twitter Search Application

Team 13:

- Meiqiao Shi
- Sarthak Singh
- Phan Nguyen Huong Le
- Alvin Alex

Search Application Overview



Managing User Data with MySQL - Users Table

By: Sarthak Singh

1. Purpose & Importance:

- Store and analyze Twitter user demographics and behaviors.
- Essential for studying user connections and influence.

2. Key Fields:

- **user_id:** Unique identifier (Primary Key).
- **screen_name, name:** User's Twitter handle and real name.
- **location, followers_count, friends_count, statuses_count:** User demographics and engagement metrics.
- **created_at:** Account creation timestamp.

3. Data Integrity:

- Primary keys ensure data uniqueness.
- No Foreign Key constraint means while processing tweets one at a time we can update User Table without any restrictions.

Structuring Tweet MetaData with MySQL - Tweets Table

1. **Purpose & Importance:**
   - Capture and analyze tweet interactions and engagement.
   - Crucial for studying information spread and user activity on Twitter.

2. **Key Fields:**
   - **tweet_id:** Unique identifier for each tweet (Primary Key).
   - **user_id, created_at:** Link to user and posting time.
   - **is_quote_status, quote_count, reply_count, retweet_count, favorite_count:** Metrics for user interactions.
   - **is_retweet, original_tweet_id:** Identifies retweets and links to original tweets for detailed analysis.

3. **Handling Retweets:**
   - Retweets are recorded by storing the original tweet ID.
   - This allows for accurate analysis of retweet behaviors—retweeting a retweet only increments the count for the original tweet.

By: Sarthak Singh

Non Relational Database- MongoDB

By Alvin Alex

- MongoDB was used for our non-relational database

- It was primarily used to store tweets and its associated data as a collection.
- The fields that were included were **tweet_id, text, hashtags, user: userid; name; screen_name, retweet_count, favorite_count, and created_at.**

- We decided to use 3 indexes. We did a **text index** on “text” to ensure faster searches.
- We did a **increasing index** on screen_name and hashtags to also ensure faster searches as the cost of increased data storage

- We inserted the tweets one at a time using collection.insert_one()
- Stored as an unordered collection

Cache Implementation

Strategy:

- ***Cache storage*:**
  - use python dict to store cache data, with keys and hashtags or userIDs and values as the data objects
- ***Eviction policy:***
  - Use ‘least recently used (LRU)’ to remove the least accessed items when the cache reaches its limit

By Phan Nguyen Huong Le

Persistent:

- Periodically serialize and save the state of cache to disk. On startup, deserialize the data to restore the cache state

Stale data handling:

- Entries in the cache can become stale if the data in the db is updated.
- Implemented a strategy to stale data and update/ remove as necessary

Expiry mechanism:

- Assign a time-to-live (TTL) to each cache entry. When an entry TTL expires, it should be considered for eviction

Cache Timing:

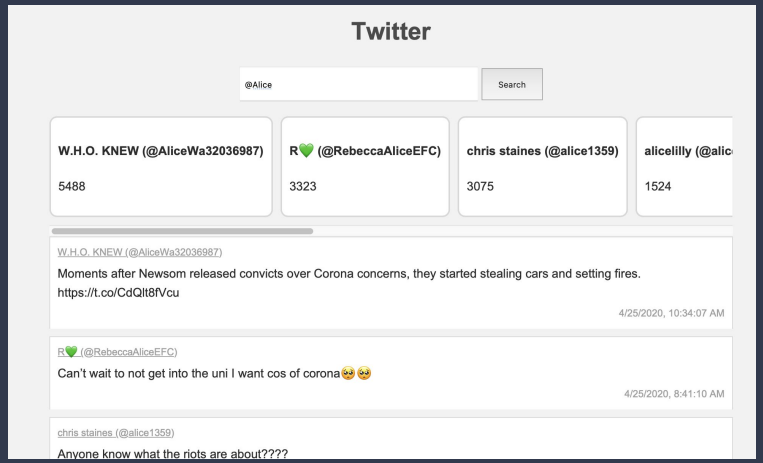
- Cache hit: takes 0.00 seconds
- Cache miss: takes roughly 0.10 seconds

```
19 INFO:root:Access took 0.000000 seconds
20 INFO:root:Access took 0.000000 seconds
21 INFO:root:Access took 0.000000 seconds
```

```
13 INFO:root:Database fetch for key 1249403767108668930 (miss) took 0.010637 seconds
14 INFO:root:Database fetch for key 1249403768023678982 (miss) took 0.010164 seconds
15 INFO:root:Database fetch for key 1249403769193779202 (miss) took 0.010381 seconds
```

Caching strategy

Search application implementation



- Creating the front-end part of the application on flask
- Ordered tweets by favorite count in decreasing order for relevance.
- Types of searches include by hashtag, by user, and by text
- If search starts with #, it searches by hashtag,
- If search starts with @, it searches by user
- If search starts with nothing, it searches by text
- Has top 10 tweets button and top 10 users sorted by followers
- also allows the time range selection

By Meiqiao Shi

Thank you