VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Big Data (CO1007)

Assignment Report

Applying Streaming SQL in Real-Time Social Media Interaction Tracking

---

Mentor: Thoại Nam
Student: Trần Hà Tuấn Kiệt – 2011493
Nguyễn Đức Thuỵ – 2012158

Ho Chi Minh City, 11/2024

Table of Contents

List of Figures

List of Tables

1 Introduction about Social media Interaction

Social Media Interaction refers to the ways users engage and communicate with content, brands, and other users on social media platforms. It encompasses all forms of engagement, such as likes, comments, shares, mentions, direct messages, and reactions to posts. These interactions create a dynamic, two-way communication channel between individuals, communities, and businesses.

Social media interaction encompasses various elements that enable engagement and communication among users, brands, and communities. Engagement types like likes, comments, shares, retweets, and mentions, which allow users to express opinions and amplify content. Direct communication, such as private messages and replies, facilitates personal connections, while user-generated content (UGC) like reviews and testimonials highlights the creative involvement of users. Additionally, community engagement through group discussions and forums fosters a sense of belonging and collaboration.

The importance of social media interaction lies in its ability to provide real-time feedback, offering valuable insights into audience opinions, preferences, and sentiments. It enhances brand awareness by increasing visibility and trust, while also allowing businesses to analyze content performance and measure campaign success. Social media interaction builds strong relationships between brands and their audiences, encourages loyalty, and helps identify emerging trends and consumer behaviors. Ultimately, it serves as a powerful tool for driving meaningful connections and strategic growth in the digital landscape.

This concept underscores the importance of leveraging unbounded data streams to stay competitive in the dynamic world of digital engagement. The accompanying vibrant image of individuals in creative poses reflects the dynamic and lively nature of social media platforms, aligning with the essence of constant interaction and engagement.

Let's take a look at a comparison between bounded dataset and unbounded dataset in data processing.

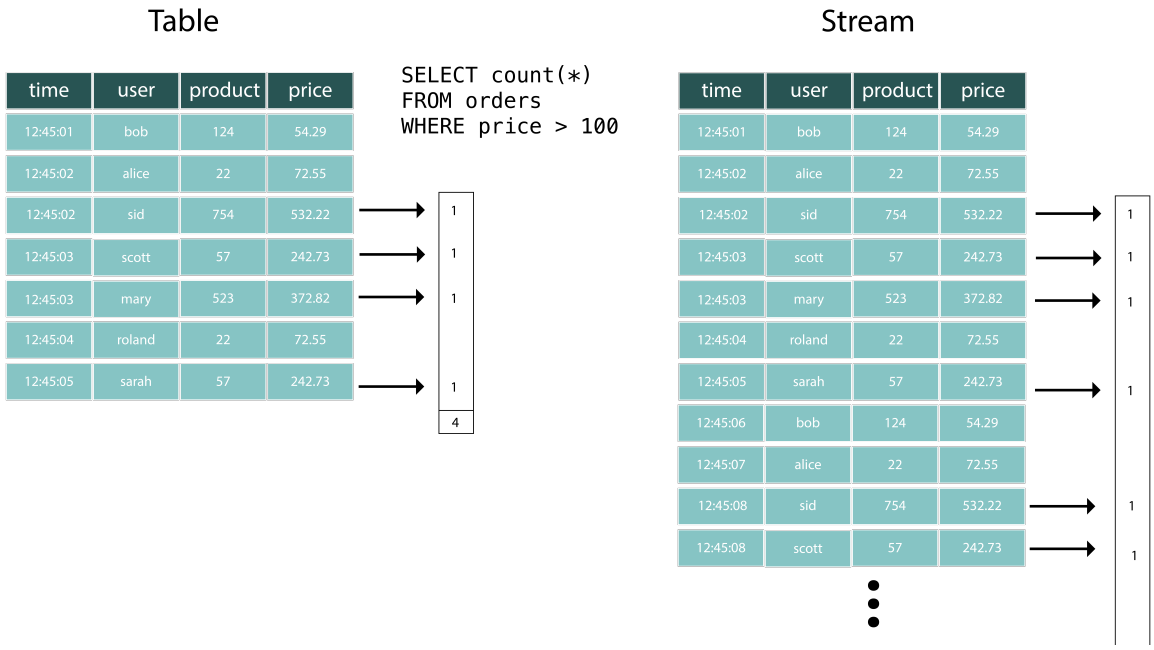


Figure 1: *Comparison between bounded dataset and unbounded dataset*

A table represents a bounded dataset, meaning it contains a finite and complete set of data at any given time. This bounded nature allows for aggregate or summary calculations, such as counting records or calculating averages. For example, the query **'SELECT count(*) FROM orders WHERE price > 100'** can aggregate the results in the table to produce a deterministic count because the data is static and complete.

On the other hand, **streams represent unbounded dataset**, where data continuously flows in over time. Since streams are ongoing, queries on streams never actually end; they continuously process incoming data. For example, when applying the same query to a stream, each incoming event is processed individually, providing incremental updates to the results. This behavior is suitable for scenarios such as real-time analytics or monitoring, where data is continuously generated and processed.

In short, the **Figure 1** demonstrates this distinction effectively, showing how tables aggregate finite data into a static result, while streams generate dynamic, continuously updated outputs. This concept is essential in understanding how modern data systems like databases, stream processors, and big data frameworks handle and process information.

2 Problem Statement

2.1 Impact

The lack of real-time insights from unbounded social data significantly hampers the ability of businesses and organizations to respond promptly to trends and audience engagement. In the fast-paced digital landscape, trends evolve rapidly, and audience preferences shift dynamically. Without continuous access to real-time data streams, it becomes difficult to monitor and analyze social interactions effectively. This delay in gaining actionable insights prevents organizations from adapting strategies, capitalizing on emerging opportunities, or addressing potential challenges in a timely manner. Ultimately, the inability to respond to real-time engagement can lead to missed opportunities, reduced competitiveness, and weakened connections with audiences, making it crucial to leverage tools that process unbounded data streams for effective decision-making and sustained growth.

2.2 Objectives

The objective is to **implement a real-time social media interaction tracking system for platforms** like Mastodon and X (formerly Twitter). This system will process and analyze unbounded streams of social media data, enabling organizations to monitor user interactions, engagement trends, and audience behaviors as they happen. By providing actionable insights in real time, it empowers businesses to stay responsive to emerging trends, adapt strategies effectively, and maintain relevance in the fast-paced digital landscape. This highlights the importance of leveraging advanced data processing technologies to manage the dynamic nature of social media, ensuring informed and timely decision-making.

2.3 Approach

2.3.1 Data Collection

Mastodon

Mastodon is a decentralized social media platform structured as a **network of independent servers**, commonly referred to as instances. Each instance operates independently, serv-

ing unique communities and interests, yet they remain interconnected through a federated network. This decentralized approach contrasts with centralized platforms, fostering diversity and community-specific experiences.

- **Structure:** Mastodon's ecosystem consists of multiple independent servers, each catering to distinct communities while participating in a larger, interconnected federation.
- **Data access:** Mastodon provides real-time data streams through its Server-Sent Events (SSE) API, enabling access to ongoing activities like posts, mentions, hashtags, and interactions.
- **Data types:** The platform generates various types of data, including posts, mentions, hashtags, and user interactions, making it a rich source for social media analytics and engagement monitoring.

Mastodon presents unique challenges due to its decentralized structure, where data is distributed across numerous independent servers (instances). Aggregating and analyzing data from the entire network is inherently complex and resource-intensive, requiring sophisticated solutions to handle the fragmented ecosystem. Additionally, Mastodon's strong emphasis on user privacy, with strict controls and regulations, further complicates data collection and processing while ensuring compliance with user preferences and privacy standards.

X (formerly Twitter)

X, formerly known as Twitter, operates as a centralized social media platform with a unified database, making it distinct from decentralized systems like Mastodon. This centralized structure provides a single, cohesive source of data, which simplifies access and analysis for users, developers, and organizations.

- **Structure:** X is built on a centralized database architecture, enabling streamlined data management and easier access to platform-wide information.
- **Data access:** The platform offers real-time streams through the X API, allowing developers to track activities such as tweets, retweets, likes, mentions, and trending topics in real time.

– **Data types:** X generates various types of valuable data, including tweets, retweets, likes, mentions, and trends, making it a vital source for social media analytics and insights.

X faces notable challenges in its data access and usage policies. Strict **rate limits on API usage** restrict the volume of data that can be collected within a specific timeframe, creating barriers to conducting comprehensive and large-scale analyses. Additionally, **privacy regulations and API restrictions** further complicate data collection and processing, as developers and researchers must navigate limitations designed to protect user data while ensuring compliance with legal and ethical standards. These challenges necessitate innovative solutions to effectively leverage the platform's data while respecting its constraints.

2.3.2 Concept

Streaming SQL is an extension of traditional SQL, specifically designed to handle unbounded, real-time data streams. Unlike standard SQL, which processes finite datasets stored in static tables, Streaming SQL operates on continuous data streams that are constantly being updated, enabling real-time analysis and processing.

Figure 2: *Concept of Streaming SQL*

Figure 3: *Concept of Streaming SQL*

Dataflow Semantics

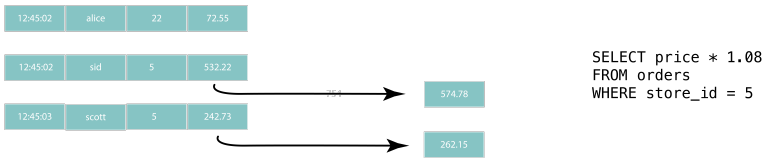


Figure 4: *Row-by-Row Processing*

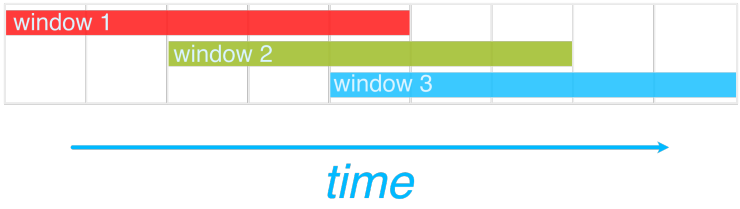


Figure 5: *Sliding windows*

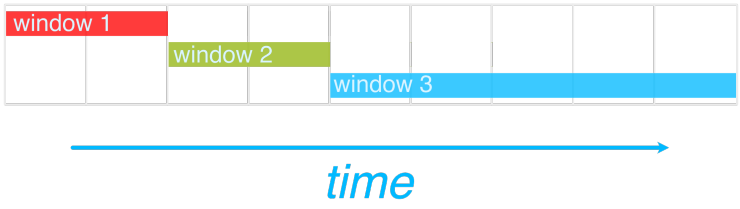
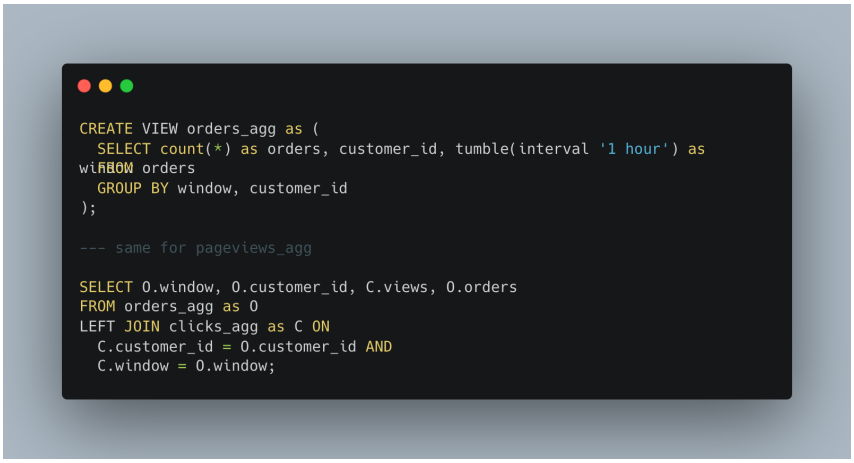


Figure 6: *Tumbling windows*



```
CREATE VIEW orders_agg as (
  SELECT count(*) as orders, customer_id, tumble(interval '1 hour') as
window
  FROM orders
  GROUP BY window, customer_id
);

--- same for pageviews_agg

SELECT O.window, O.customer_id, C.views, O.orders
FROM orders_agg as O
LEFT JOIN clicks_agg as C ON
  C.customer_id = O.customer_id AND
  C.window = O.window;
```

Figure 7: *Apply to Join Operation*

Update Semantics

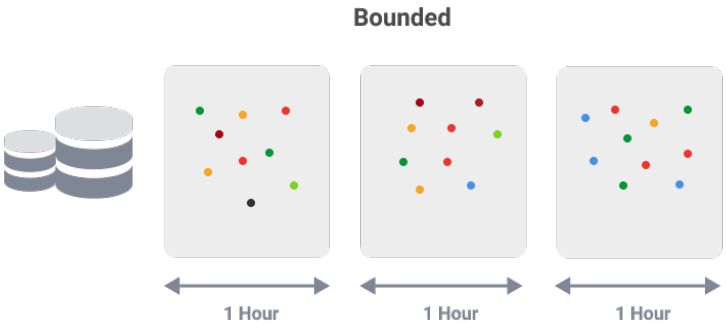


Figure 8: *Bounded Data in Time*

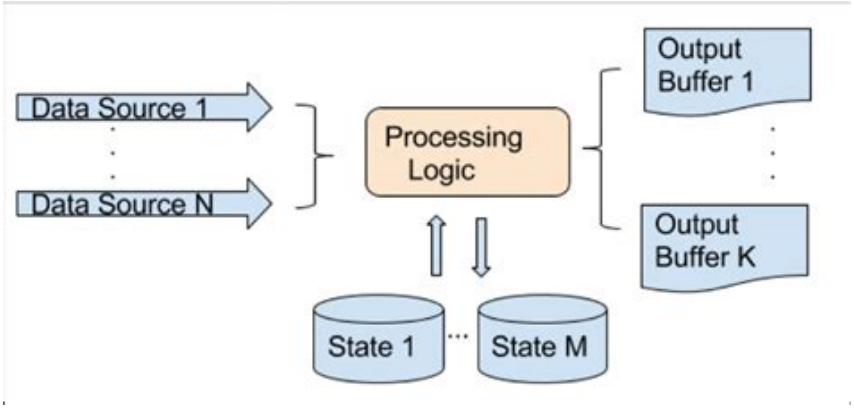


Figure 9: *processing logic of Incremental Updates*

2.3.3 Tool

Figure 10: *Arroyo*

We will incorporate **Arroyo**, a distributed stream processing engine developed in Rust, as part of our technology stack to handle real-time data streams efficiently. Arroyo is designed to support SQL-based queries, making it accessible for processing large volumes of unbounded data streams with ease. Its high scalability ensures that it can handle significant workloads while maintaining **exactly-once semantics**, which is critical for ensuring data accuracy and consistency. This tool will enable us to implement robust, real-time data processing pipelines, making it an ideal choice for applications requiring low-latency and fault-tolerant stream analytics.

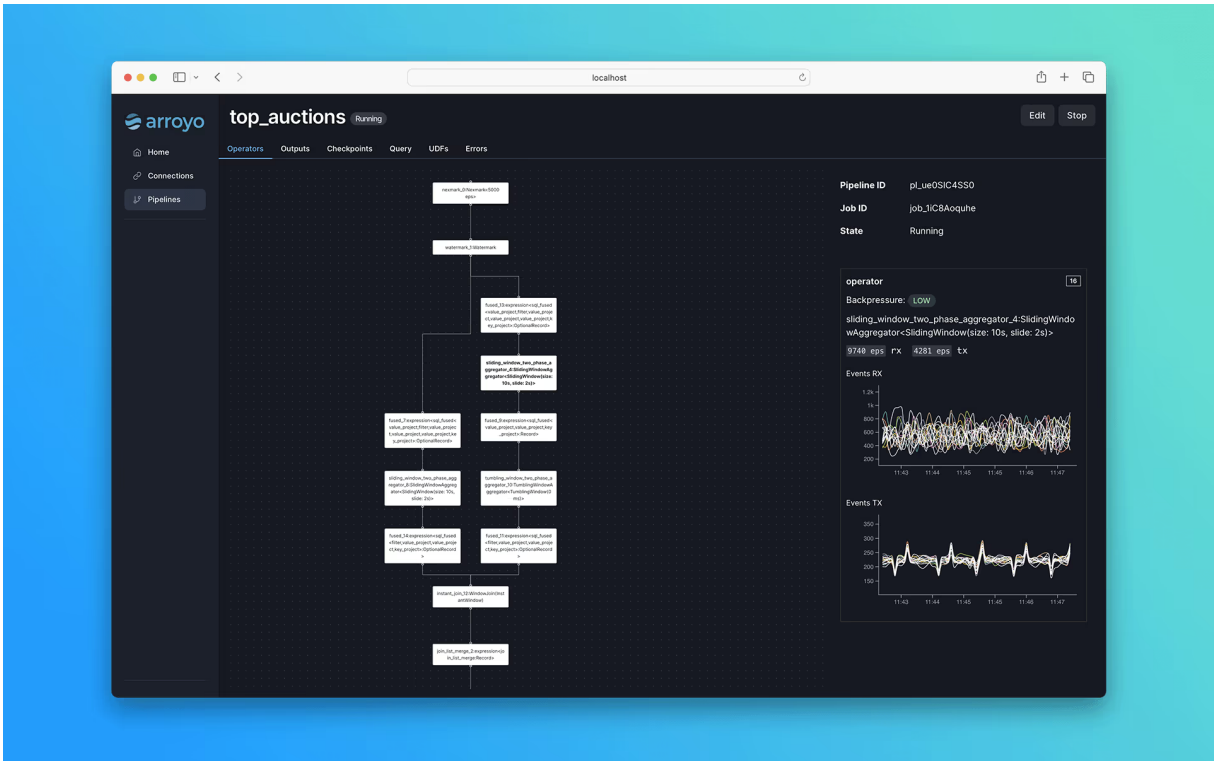


Figure 11: *dashboard*

3 Implementation

3.1 Set Up Infrastructure

3.2 Define Data Pipeline

3.3 Develop Key Tracking Metrics

3.4 Build dashboard

4 Conclusion

References

[1] Ra'ed M. Al-Khatib, Mohammed Al-Betar, Mohammed Awadallah, Khalid Nahar, Mohammed Abu Shquier, Ahmad Manasrah, and Ahmad Doumi. Mga-tsp: Modernized genetic algorithm for the traveling salesman problem. *International Journal of Reasoning-based Intelligent Systems*, 11:1, 01 2019. doi: 10.1504/IJRIS.2019.10019776.

[2] Annu Lambora, Kunal Gupta, and Kriti Chopra. Genetic algorithm- a literature review. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 380–384, 2019. doi: 10.1109/COMITCon.2019.8862255.