UNICAM
Università di Camerino
1336

STREAM OF REDDIT DATA

Technologies for Big Data Mangement

NICCOLÒ VACCA

INDEX

- Project description and objectives
- Methodologies and technologies being used
- Apache Flink vs Apache Spark
- Technical implementation
- Achieved results
- Future improvements

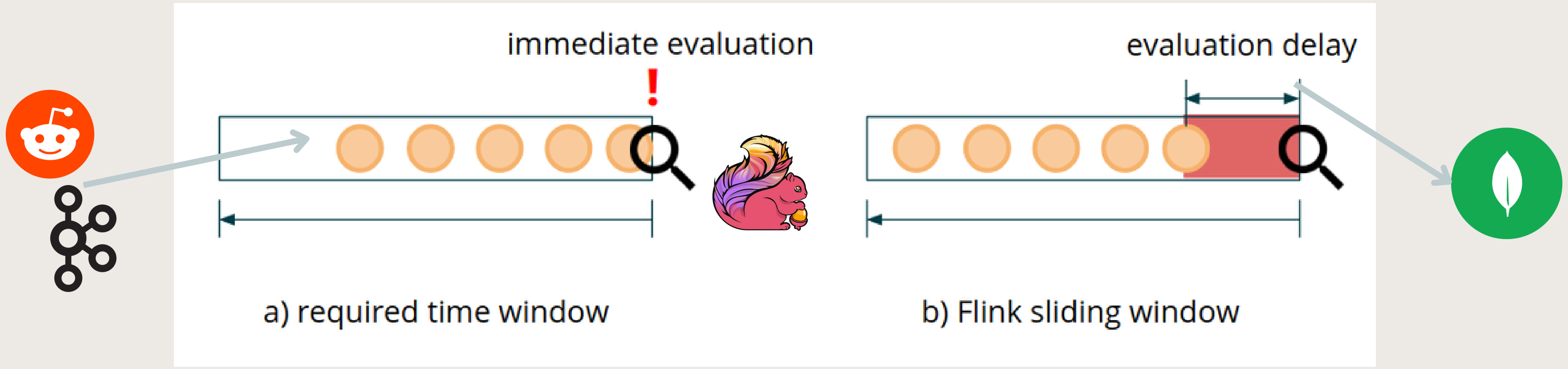
PROJECT DESCRIPTION AND OBJECTIVES

PROJECT DESCRIPTION AND OBJECTIVES

The **Streams of Reddit Data** project aims to retrieve real-time data from a **Reddit**'s subreddit decided by the user and to process this data within time windows.

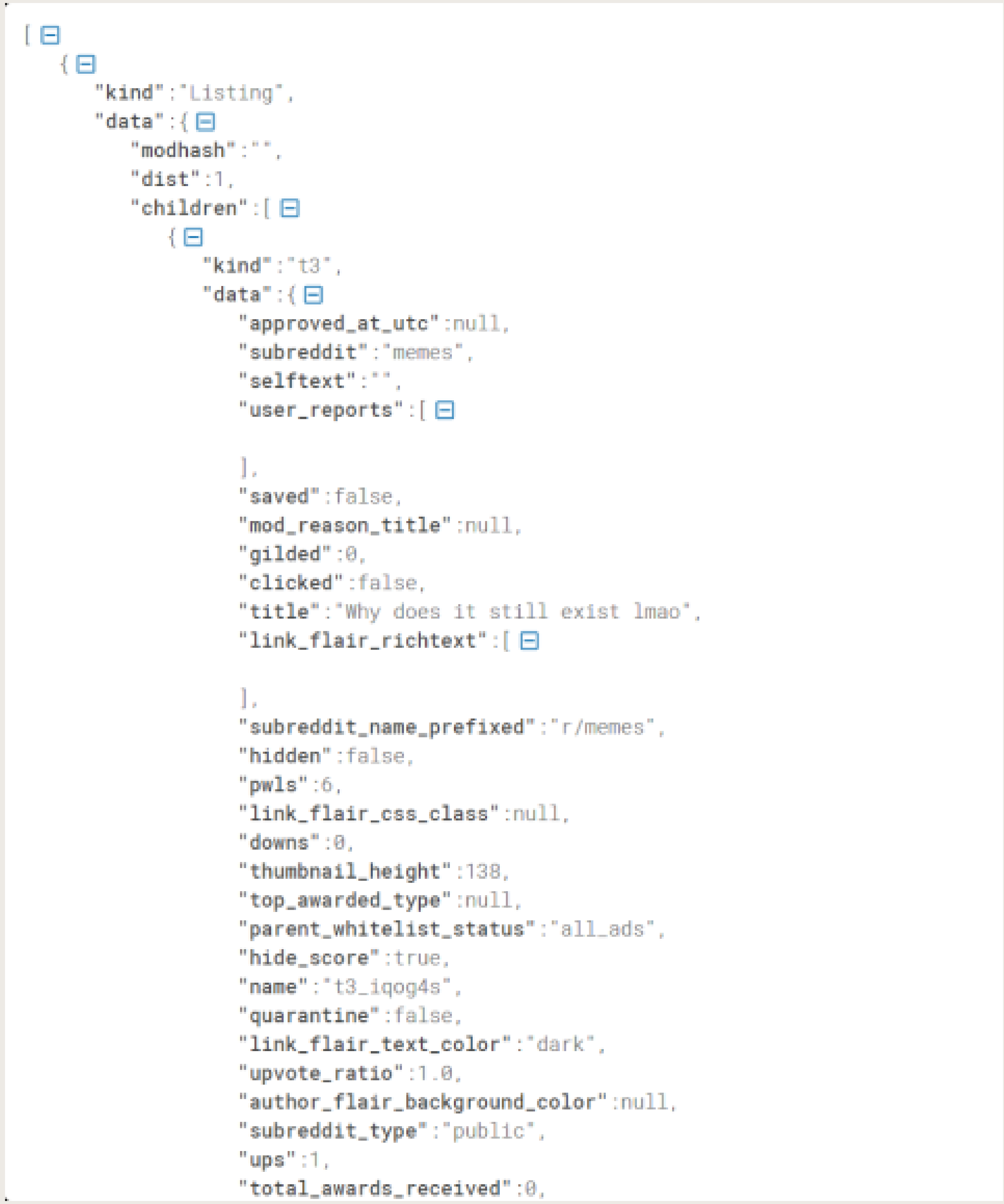
This means that computation is applied, once the **time window** closes, on all the data coming from the Reddit posts read within that specific window.

Both the raw and the computed data is stored into a **MongoDB** database, in two different collections.



PROJECT DESCRIPTION AND OBJECTIVES

REDDIT API - RESPONSE EXAMPLE

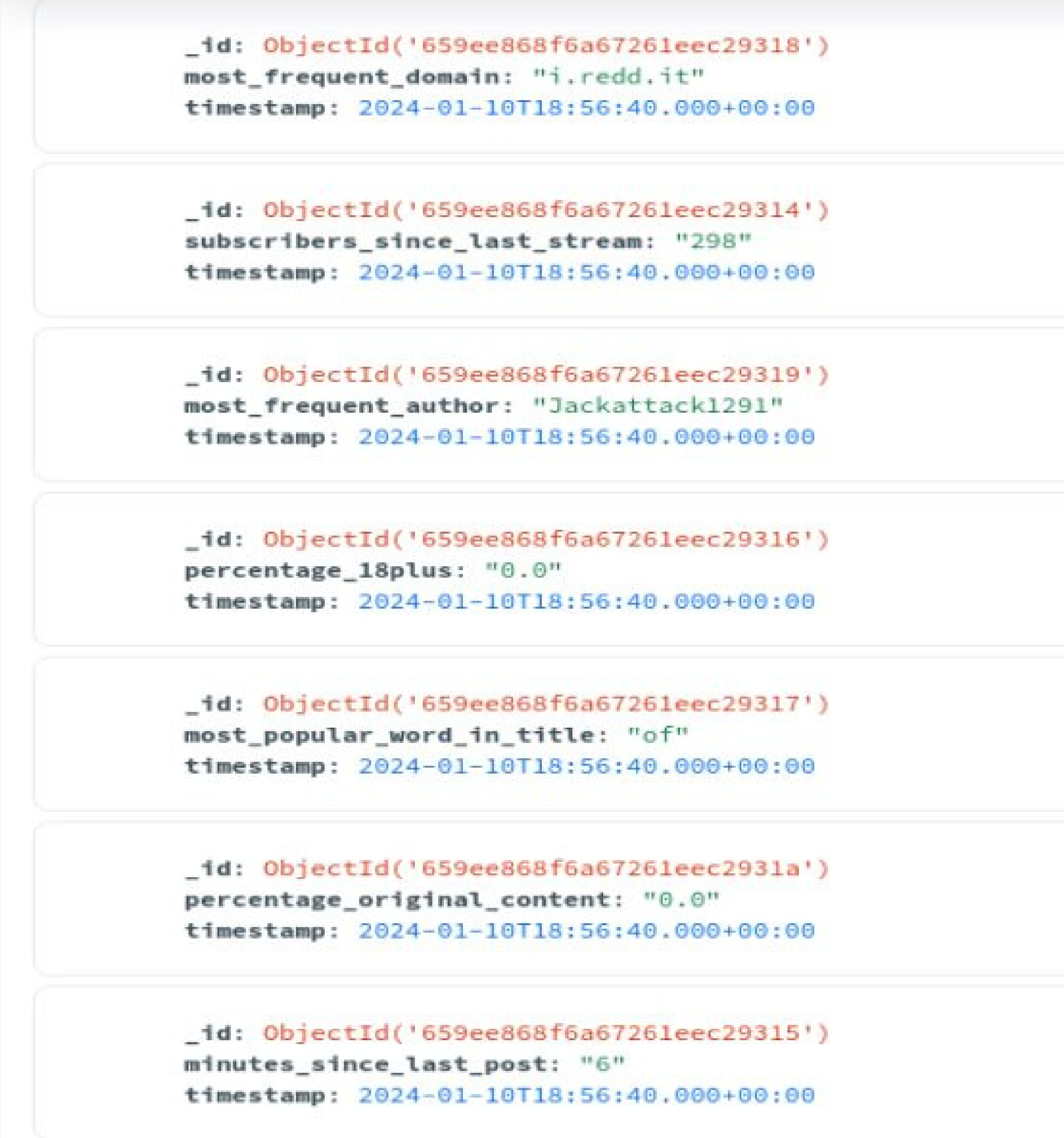


```
[ ⊟
   { ⊟
      "kind":"Listing",
      "data":{ ⊟
         "modhash":"",
         "dist":1,
         "children":[ ⊟
            { ⊟
               "kind":"t3",
               "data":{ ⊟
                  "approved_at_utc":null,
                  "subreddit":"memes",
                  "selftext":"",
                  "user_reports":[ ⊟

                  ],
                  "saved":false,
                  "mod_reason_title":null,
                  "gilded":0,
                  "clicked":false,
                  "title":"Why does it still exist lmao",
                  "link_flair_richtext":[ ⊟

                  ],
                  "subreddit_name_prefixed":"r/memes",
                  "hidden":false,
                  "pwls":6,
                  "link_flair_css_class":null,
                  "downs":0,
                  "thumbnail_height":138,
                  "top_awarded_type":null,
                  "parent_whitelist_status":"all_ads",
                  "hide_score":true,
                  "name":"t3_iqog4s",
                  "quarantine":false,
                  "link_flair_text_color":"dark",
                  "upvote_ratio":1.0,
                  "author_flair_background_color":null,
                  "subreddit_type":"public",
                  "ups":1,
                  "total_awards_received":0,
```

PROJECT DESCRIPTION AND OBJECTIVES

COMPUTATIONS

- Word that appears most frequently in “*title*”
- “*author*” that appears most frequently
- “*domain*” that appears most frequently
- Percentage of “*is_original_content*” being true
- Percentage of “*over_18*” being true
- Difference between last and first “*subreddit_subscribers*”
- Time difference between last and second-last “*created_utc*”



```
_id: ObjectId('659ee868f6a67261eec29318')
most_frequent_domain: "i.redd.it"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec29314')
subscribers_since_last_stream: "298"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec29319')
most_frequent_author: "Jackattack1291"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec29316')
percentage_18plus: "0.0"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec29317')
most_popular_word_in_title: "of"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec2931a')
percentage_original_content: "0.0"
timestamp: 2024-01-10T18:56:40.000+00:00

_id: ObjectId('659ee868f6a67261eec29315')
minutes_since_last_post: "6"
timestamp: 2024-01-10T18:56:40.000+00:00
```

METHODOLOGIES AND TECHNOLOGIES BEING USED

METHODOLOGIES AND TECHNOLOGIES BEING USED

KEY ASPECTS

- **Data Fetching**: Utilizes the Reddit API, authenticated with OAuth tokens, through the Java-based RedditDataFetcher class.
- **Data Streaming**: Employs Apache Flink and Kafka to stream data continously.
- **Data Processing**: Implements windowed computations within the Apache Flink application.
- **Data Storage**: Utilizes MongoDB, with separate collections for raw data ('*reddit-data*') and computed results ('*reddit-computed-data*').
- **Execution Environment**: Where to run the services used - in this case they run on a Virtual Machine instantiated through Oracle VM VirtualBox.

METHODOLOGIES AND TECHNOLOGIES BEING USED

FOCUS ON APACHE FLINK AND APACHE KAFKA

- **Flink** allows you to perform windowed operations on the streaming data, enabling computations over specified time intervals.
- **Flink** integrates with **Kafka**, acting as both a source and a sink. It consumes data from the Kafka topic ('*reddit*') and produces computed results back into Kafka.
- **Kafka** allows the RedditDataFetcher class to publish data to the '*reddit*' topic.
- **Kafka** acts as a buffer, decoupling the data source (Reddit API) from the data processing engine (Flink). This buffering ensures that data is available for Flink even if there are issues in the data source.

APACHE FLINK VS APACHE SPARK

APACHE FLINK VS APACHE SPARK

Flink	APACHE Spark™
1. **Exactly-once** semantic	1. Exactly-once semantic is **not guaranteed**
2. Processes data in **real**-time	2. Processes data in **near**-real-time
3. **Can** recognize the temporal order of received messages	3. **Can't** recognize the temporal order of received messages
4. Designed for **stream processing**	4. Designed mainly for **batch processing**
5. Supports mainly **Java and Scala**	5. Supports **many** programming languages
6. **Doesn't support** graph processing or machine learning	6. **Supports** graph processing and machine learning
7. Distributed Architecture with a JobManager coordinating tasks across multiple TaskManagers	7. Distributed Architecture with a driver program coordinating tasks across worker nodes

TECHNICAL IMPLEMENTATION

TECHNICAL IMPLEMENTATION

- **RedditDataFetcher Class**: Fetches data from Reddit API.
  - **methods**: *getAccessToken*, *makeRedditApiRequest*, *isAccessTokenValid*.

- **Producer Class**: Streams Reddit data into Kafka.
  - **methods**: *StreamProducer*, *StreamGenerator*, *createStringProducer*.

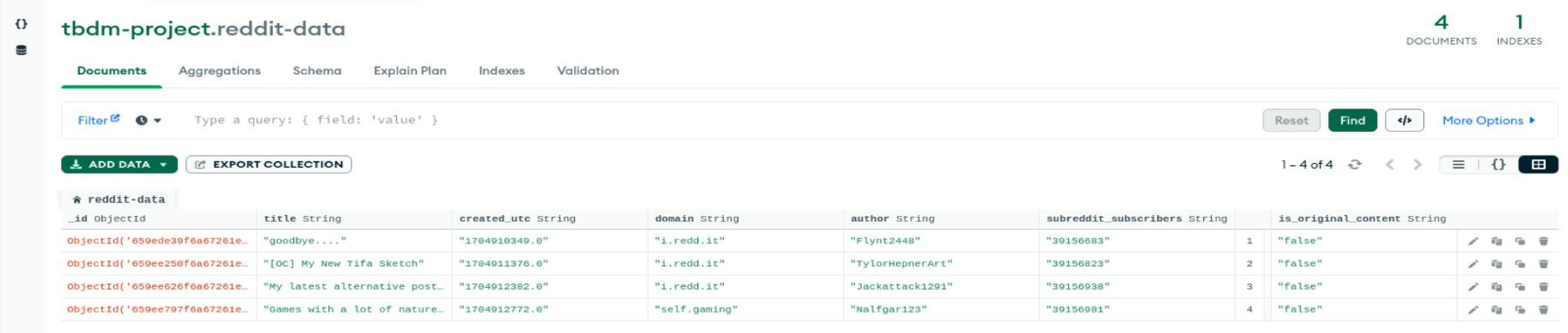
- **Consumer Class**: Consumes data from Kafka, applies windowed computations, and sinks results into MongoDB.
  - **methods**: *consumeFromKafka*, methods for data processing.

- **MongoDB Manager Classes**: *MongoDBSink* and *MongoDBSinkComputed* serve to write raw and computed data into MongoDB, respectively.

ACHIEVED RESULTS

ACHIEVED RESULTS

The application retrieve real-time data from Reddit posts of a specific subreddit; it also creates insights starting from this raw data, including the identification of the most frequent words in titles, top authors, percentage of original content, and more. Raw data is stored alongside computed results, giving the possibility to check for what has been gathered and eventually implement data visualization techniques.



_id ObjectId	title String	created_utc String	domain String	author String	subreddit_subscribers String		is_original_content String
ObjectId('659ede39f6a67261e…	"goodbye...."	"1704910349.0"	"i.redd.it"	"Flynt2448"	"39156683"	1	"false"
ObjectId('659ee250f6a67261e…	"[OC] My New Tifa Sketch"	"1704911376.0"	"i.redd.it"	"TylorHepnerArt"	"39156823"	2	"false"
ObjectId('659ee626f6a67261e…	"My latest alternative post…	"1704912382.0"	"i.redd.it"	"Jackattack1291"	"39156938"	3	"false"
ObjectId('659ee797f6a67261e…	"Games with a lot of nature…	"1704912772.0"	"self.gaming"	"Nalfgar123"	"39156981"	4	"false"

FUTURE IMPROVEMENTS

FUTURE IMPROVEMENTS

- Exclude **insignificant words** from the word count in titles (*of*, *the*, *a*, *my*, etc.)
- **Add metrics and analyses** for more comprehensive insights
- Add other **computation that doesn’t use time windows** and processes data as soon as it’s read
- Implement **data visualization**: for example using Tableau, Looker or MongoDB Charts

THANK YOU