Phase 4: Analytical DB POC

Group - Pi

Analytical Business questions

https://github.com/1captain0/stream-processing-workshop.git

Questions and results

Question 1: Which genre generates the highest revenue, and what is the average ticket price for events in that genre?

Tables: Streams, Artist, Ticket, Event

Query:

```
SELECT a.genre, SUM(t.price) as sum_ticket_price, AVG(t.price) AS avg_ticket_price
FROM stream s
JOIN artist a ON s.artistid = a.id
JOIN event e ON e.artistid = a.id
JOIN ticket t ON t.eventid = e.id
GROUP BY a.genre
ORDER BY sum_ticket_price
LIMIT 1;
```

Question 2 : Which artist events had the highest and lowest ticket sales compared to their capacity?

Tables: ETENRICH (enriched data from kafka streams - event + ticket + turnout)

Query:

```
WITH latest_event_turnouts AS (
  SELECT
        e2."timestamp" AS ts_group,
        e1.artistid,
        e1.eventid,
        e1.turnout,
        ROW_NUMBER() OVER (
        PARTITION BY e2."timestamp", e1.eventid
        ORDER BY e1."timestamp" DESC
        ) AS rn
  FROM etenrich e1
  JOIN etenrich e2
        ON e1."timestamp" <= e2."timestamp"
  WHERE e1.artistid = '368564877' AND e2.artistid =
'368564877'
),
filtered AS (
  SELECT * FROM latest_event_turnouts WHERE
rn = 1
),
min_turnouts AS (
  SELECT
        ts_group,
        artistid,
        MIN(turnout) AS min_turnout
  FROM filtered
  GROUP BY ts_group, artistid
),

max_turnouts AS (
  SELECT
        ts_group,
        artistid,
        MAX(turnout) AS max_turnout
  FROM filtered
  GROUP BY ts_group, artistid
),
worst_event AS (
  SELECT
        f.ts_group,
        f.artistid,
        f.eventid AS worst_eventid,
        f.turnout AS worst_turnout
  FROM filtered f
  JOIN min_turnouts m
        ON f.ts_group = m.ts_group
   AND f.artistid = m.artistid
   AND f.turnout = m.min_turnout_so_far
),

best_event AS (
  SELECT
        f.ts_group,
        f.artistid,
        f.eventid AS best_eventid,
        f.turnout AS best_turnout
  FROM filtered f
  JOIN max_turnouts m
        ON f.ts_group = m.ts_group
   AND f.artistid = m.artistid
   AND f.turnout = m.max_turnout_so_far
)
```

Retrospective Results

Streaming Aggregations vs Analytic Aggregations

Aspect	Streaming Aggregations	Analytic (OLAP) Aggregations
Latency	real-time	batch or ad-hoc
Data scope	Incremental, windowed slices only	Full-history scans
State	Stateful operators in-flights	Immutable snapshot tables with indexes
Flexibility	Fixed upfront (window definitions)	Ad-hoc (arbitrary GROUP BY, pivots, etc.)
Use cases	Continuous dashboards, alerts	Deep historical analysis, BI reporting

What went well?

Leverage Pinot’s Native SQL Joins

- Instead of building a separate "enrichment" pipeline, we defined our raw event and dimension tables in Pinot and used SQL joins to answer our business questions in a single step. (at least for the first query)

Understanding Streaming vs. Analytic Aggregation

- We developed a clear understanding (kind of) of the distinction between OLAP (Online Analytical Processing) and analytic queries in the context of Pinot. While OLAP solutions typically focus on pre-aggregated data in cubes for fast query performance, we learned how to leverage Pinot’s real-time capabilities to execute direct SQL queries on raw data tables, offering flexibility for both real-time and batch analytics.

What didn’t go well ?

Forming Business Questions

- The questions we formed for Phase 3 were mostly related to streaming aggregations, so we had to slightly adjust them to fit the requirements for Phase 4. We did not initially have the right questions for this type of aggregation from the previous phase and resulted in complex queries that may not be computationally efficient

Understanding the Trade-offs and differences

- Difficulty in understanding what the tradeoffs were between using stream processing vs directly querying in pinot from the tables. Deciding why should we choose one over the other even when if it’s possible in both scenarios and what should be the right way to go given we are presented with only a particular tools

Actions that can be taken

Query Platform & Aggregation Framework

- Think through the end-to-end query process, clearly categorize the different aggregation types to ensure they align with your business questions, and then decide whether to implement each aggregation step in Apache Pinot or in Kafka Streams, selecting the most appropriate platform for each.