University of Tartu

Faculty of Science and Technology

Institute of Computer Science

Department of Information Technology

Movie Data Analysis

Project Part 3

Max Sebastian Segerkrantz, Kevin Christian Eriksson, Oto Pruul

Tartu

Project 3 additions

1. Apache Iceberg

- Integrated Apache Iceberg using a REST catalog backed by MinIO, with our Airflow pipeline responsible for creating and managing the Iceberg tables. Although the official ClickHouse OSS Docker image does not include support for the Iceberg table engine, we ensured that the Iceberg-managed data remained queryable from ClickHouse by exposing the underlying Parquet files stored in MinIO. Using ClickHouse's S3 table engine, we created an external read-only table that directly references the Parquet files produced by Iceberg.
- This approach preserves the required Airflow → Iceberg → ClickHouse while enabling ClickHouse to query Iceberg-managed data even without native Iceberg engine support.

2. Clickhouse users and roles

- Added a full access (**analytical_view_full**) and limited access (**analytical_view_limited**) pseudonymised view to be created along with the gold tables for Clickhouse RBAC demonstration. The views are based on the fact table data.

- Created a new limited access user and a full access user together with corresponding limited and full access roles.

- The full access user can use the full access view regarding movie performance, while the limited access user has 4 of the columns masked - movie_title, release_year, director_name, and vote_avg. Since none of the columns contain exactly sensitive data, these 4 were chosen for the task.

- **Pseudonymisation:**

1. **movie_title** - uses cityHash64 to hash the title
2. **release_year** - two last numbers of the year are masked with *
3. **director_name** - only the first name of the director is shown and anything else is replaced with five * symbols.
4. **vote_avg** - numeric rating is masked with a categorical rating -
   LOW (vote < 6), **MID (6 >= vote < 8)**, **HIGH (8 >= vote)** based on the numeric value.

3. OpenMetadata

OpenMetadata was connected to the ClickHouse warehouse and configured to ingest metadata from the gold layer.

The following were added:

- A ClickHouse service in OpenMetadata
- Ingestion pipeline to load tables and schemas
- Table-level and column-level descriptions for all fact and dimension tables
- Data quality tests for fact and director dimension tables

3.1 Documentation Added

Each dimension and fact table has table-level description and column-level descriptions.

This includes:

- dim_movie
- dim director
- dim_genre
- dim_production
- dim_date
- fact_movie_performance
- analytical_view_full
- analytical_view_limited

3.2 Tests Added

- **Not-Null Test (Fact Table)**
  Example: fact_movie_performance.movie_id and .fact_id IS NOT NULL
  Why: Foreign keys in fact tables must always be present. A missing key would break joins and produce incorrect analytics. This ensures referential integrity.

- **Unique Surrogate Key Test (Dimension Tables)**
  Example: dim_movie.movie_id and .imdb_dir_name_id IS UNIQUE
  Why: Surrogate keys must uniquely identify dimension rows. A duplicate movie or director key would produce incorrect aggregations and double counting.

4. Apache Superset

Apache superset was also connected to the ClickHouse warehouse.

A dashboard was created in Apache Superset using the analytical views.
The dashboard includes:

- Two charts visualizing average ratings by genre and directors.
- A filter (by genre)
- It answers business questions such as: Which genres have the highest average ratings? & Which directors consistently produce high-rated movies?

Group member's roles and contributions:

- Max Sebastian Segerkrantz (33%)
  Apache Iceberg and Minio implementation. Creating a newly modified and updated DAG with additional filtering on tmdb data. Implementing Iceberg with bronze layer data and making it queriable in Clickhouse through Minio. Testing and providing screenshots.

- Kevin Christian Eriksson (33%)
  Implementing OpenMetadata, adding column descriptors, data quality tests and testing with screenshots. Additionally, integrating Apache Superset and creating the dashboard.

- Oto Pruul (33%)
  Creating full access and pseudonymised views with masked data. Also creating corresponding users and roles for RBAC demonstration in ClickHouse. Testing with queries and screenshots.

NB!

When running the project, specifically the DAGs, it may happen that the computer runs out of memory and some tasks fail. To counteract this, when running DAGs in Airflow, disable Superset and Open Metadata containers as those take the most memory.

Furthermore, when testing the DAGs with a Mac system, the creating of the users didn't run successfully (permission problems that we just didn't know how to fix). That shouldn't block you from running everything else (like OMD or Superset), it just means that the users were not created. Running the task with a Windows machine worked.

In theory everything should be running smoothly.

LLM disclosure:

Similarly to project 2, ChatGPT and Google Gemini were used for debugging and troubleshooting issues. These tools assisted in identifying errors and optimizing code to ensure functionality. No direct code or text was copied from LLMs; they were used strictly as debugging and problem-solving aids.

No specific conversation links are provided, as the process involved numerous iterative debugging sessions and adjustments throughout.