Assignment 2: Building a Batch Analytics Pipeline on HDFS & Hive

Due Date: 11:59 PM 7th March

Scenario & Objectives

Your company, MediaCo, gathers large daily logs of user activity from a streaming platform (e.g., plays, skips, pauses). Your task is to design a batch analytics solution using HDFS for data storage and Hive for querying:

1. Ingest daily log files from a local directory into HDFS, organizing them by date.
2. Create Hive tables to store raw data (CSV/JSON) and a star schema (fact + dimension tables) for analytics.
3. Run analytical queries to generate insights (monthly usage, top content, average session times).

Data Description

1. **User Logs**: (user_id, content_id, action, timestamp, device, region, session_id, ...)
    - Arrives in CSV or JSON format.
    - Each day's logs in a local folder named YYYY-MM-DD.
2. **Content Metadata** (content_id, title, category, length, artist, ...)
    - Static reference data about each piece of content.

Core Requirements

1. **Ingestion Script**
    1. Write a shell script (e.g., ingest_logs.sh) that:
        - Accepts a date parameter (e.g., 2023-09-01).
        - Parses year/month/day.

        - Copies files into HDFS under a directory like /raw/logs/<year>/<month>/<day> and /raw/metadata/<year>/<month>/<day>
2. **Raw Tables in Hive**
    1. Create **external** tables pointing to /raw/logs and /raw/metadata.
    2. Partition by (year, month, day) for the log table so queries can filter by date.
3. **Star Schema**
    1. **Fact Table**: e.g., fact_user_actions storing user actions (partitioned by date).
    2. **Dimension Table**: e.g., dim_content storing content metadata.
    3. Store them in a **columnar format** (e.g., Parquet).
4. **Transformation**
    1. Use Hive SQL (INSERT OVERWRITE, CTAS) to move data from the raw tables to the star schema tables.
    2. Convert timestamps to proper types, if needed.
5. **Queries**
    1. Demonstrate **2–3** analytical queries:
        - E.g., “Monthly active users by region,” “Top categories by play count,” “Average session length weekly.”
    2. Include **group by**, **join** (fact + dimension), and **filters** on date partitions.
6. **Deliverables:** Please create a GitHub repository with 2 files and 1 folder. PDF file to be uploaded on LMS.
    1. **Input Data:** Create a folder named raw_data and put your generated input files here
    2. **Shell Ingestion Script**: Short .sh file name ingest_logs.sh
    3. **Hive DDL** for raw and star schema tables. The working queries should be included in the document.
    4. **Data Transformation** commands. The working queries should be included in the document.
    5. **Sample Queries** with results (Screenshots) to be included in the docs.
    6. **Short Write-Up** with the above queries and commands. Please explain the design choices and performance considerations. Especially including 1- how long the execution of the whole pipeline takes. 2- query execution times.

Grading / Assessment Criteria

- **Dataset generation:** Generate a reasonable dataset. Feel free to increase number of days.
- **Ingestion**: Correct partitioning, shell script usage.
- **Data Modeling**: Proper star schema (fact/dimension separation), partition columns.
- **Transformation**: Successful movement from raw CSV to Parquet, correct field typing.
- **SQL Queries**: Logical joins, aggregations, beneficial use of date partitions.
- **Write-Up**: Clear rationale for design, mention of potential performance optimizations.

Note: There might be vivas for this assignment so understand what you are doing!

Helping Resources

1. **Hive Documentation**:
    - https://cwiki.apache.org/confluence/display/Hive/Home
      Covers CREATE EXTERNAL TABLE, partitioning, INSERT OVERWRITE, SerDes for CSV/JSON, etc.
2. **HDFS Basics**:
    - https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html
    - https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html
      Explains file system commands (hdfs dfs -mkdir, -put, etc.).
    - Note: Please follow the Pseudo-Distributed Operation for the deployment of a single node cluster (https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html)
3. **Introduction to Shell Scripting:**
    - https://www.shellscript.sh/
4. **Dimensional Modeling**:
    - Ralph Kimball's "The Data Warehouse Toolkit" or numerous online articles about star schemas, fact and dimension design.
5. **CSV to Parquet** with Hive:
    - Example: https://docs.cloudera.com/documentation/enterprise/5-6-x/topics/cdh_ig_hive.html
      Illustrates how to store final data in a columnar format.
6. **Partitioning in Hive**:
    - https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL#LanguageManualDDL-PartitionedTables
      For dynamic partitioning settings and partition maintenance.

Using LLM for generating synthetic data (use any free LLM)

System / User Prompt

"Please generate **two separate CSV datasets** that I can use to simulate a streaming application's data in a data engineering assignment:

1) User Activity Logs

- Columns: user_id, content_id, action, timestamp, device, region, session_id
- Number of Rows: ~20–30 per day, for at least **7 different days** (e.g., 2023-09-01, 2023-09-02, 2023-09-03).
- Provide the logs in **CSV** format with a header row and valid data.
- The timestamp should be a full date+time (e.g., 2023-09-01 08:23:55).
- action: from {play, pause, skip, forward}, randomly assigned.
- device: from {mobile, desktop, tablet}.
- region: from {US, EU, APAC}, randomly assigned.
- session_id: short alphanumeric IDs, repeated occasionally for the same user's session.
- user_id: integer range ~100–200; content_id: integer range ~1000–1010.

2) Content Metadata

- Columns: content_id, title, category, length, artist
- ~8–12 rows total, with content_id matching the same range used in the logs (1000–1010).
- title: short text (e.g., “Summer Vibes”, “Rock Anthem”).
- category: {Pop, Rock, Podcast, News, Jazz, etc.}, pick randomly.
- length: integer representing total seconds or minutes (e.g., 180 for 3 minutes).
- artist: random short name (e.g., “DJ Alpha”, “The Beats”).
- Provide **separate CSV** output for this metadata file, also with a header row.

Output Format:

- Return two code blocks:
    1. The user activity logs for multiple days (with ~20–30 rows per day).
    2. The content metadata (8–12 rows).
- Use valid CSV syntax, comma-delimited, including header rows.

Make sure the content_id in the logs **overlaps** the content_id in the metadata so we can join them later.

Thank you!”

Tips/Notes:

- Tweak the **date range**, **row count**, or **field distributions**. We need at least 7 days of data.
- For **separate files** per day, ask LLM to generate each date’s logs **in a separate code block** or with a clear label.
- For realism, we want to ask for **variations** in user_id distribution, session_id formats, or location (region).

Good Luck!