

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

Group No: 38

Muhammad Annus Shabbir – 24280015

Muhammad Atif - 24280067

GitHub Repo Link: https://github.com/annus-lums/ai601_assignment_2

1) Ingestion Script Execution Command:

```
for date in $(seq -f "%02g" 1 7); do  
    ./ingest_logs.sh "2023-09-$date"  
Done
```

2) Raw Tables in Hive

```
CREATE EXTERNAL TABLE raw_user_logs (  
    user_id INT,  
    content_id INT,  
    action STRING,  
    `timestamp` STRING,  
    device STRING,  
    region STRING,  
    session_id STRING  
)  
PARTITIONED BY (year INT, month INT, day INT)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  
WITH SERDEPROPERTIES (  
    "separatorChar" = ",",  
    "skip.header.line.count" = "1"  
)
```

STORED AS TEXTFILE

LOCATION '/raw/logs/';

SHOW TABLES;

DESCRIBE raw_user_logs;

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=1) LOCATION
'/raw/logs/2023/09/01/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=2) LOCATION
'/raw/logs/2023/09/02/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=3) LOCATION
'/raw/logs/2023/09/03/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=4) LOCATION
'/raw/logs/2023/09/04/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=5) LOCATION
'/raw/logs/2023/09/05/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=6) LOCATION
'/raw/logs/2023/09/06/';*

*ALTER TABLE raw_user_logs ADD PARTITION (year=2023, month=9, day=7) LOCATION
'/raw/logs/2023/09/07/';*

SHOW PARTITIONS raw_user_logs;

*SELECT * FROM raw_user_logs LIMIT 10;*

CREATE EXTERNAL TABLE raw_content_metadata (

content_id INT,

title STRING,

category STRING,

length INT,

artist STRING

)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

```

WITH SERDEPROPERTIES (
    "separatorChar" = ",",
    "skip.header.line.count" = "1"
)
STORED AS TEXTFILE
LOCATION '/raw/metadata/';

SHOW TABLES;
DESCRIBE raw_content_metadata;
SELECT * FROM raw_content_metadata LIMIT 10;

```

3) Star Schema Tables

```

CREATE TABLE fact_user_actions (
    user_id INT,
    content_id INT,
    action STRING,
    `timestamp` TIMESTAMP,
    device STRING,
    region STRING,
    session_id STRING
)
PARTITIONED BY (year INT, month INT, day INT)
STORED AS PARQUET;

CREATE TABLE dim_content (
    content_id INT,
    title STRING,
    category STRING,
    length INT,
    artist STRING

```

)

STORED AS PARQUET;

4) Transformation

INSERT OVERWRITE TABLE fact_user_actions PARTITION (year, month, day)

SELECT

user_id,

content_id,

action,

CAST(timestamp` AS TIMESTAMP),

device,

region,

session_id,

year,

month,

day

FROM raw_user_logs;

INSERT OVERWRITE TABLE dim_content

*SELECT * FROM raw_content_metadata;*

*SELECT * FROM dim_content LIMIT 10;*

SHOW PARTITIONS fact_user_actions;

5) Queries

Monthly active users by region:

```
SELECT region, COUNT(DISTINCT user_id) AS active_users
FROM fact_user_actions
GROUP BY region;
```

```
[hive>
[ >
[ >
[ > SELECT region, COUNT(DISTINCT user_id) AS active_users
[ > FROM fact_user_actions
[ > GROUP BY region;
Query ID = apple_20250311001826_c74b8685-182e-4ad7-9886-147885955565
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-03-11 00:18:28,305 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local156236339_0003
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 53970 HDFS Write: 31044 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
APAC      44
EU        41
US        44
Time taken: 1.876 seconds, Fetched: 3 row(s)
hive>
```

Top categories by play count:

```
SELECT d.category, COUNT(*) AS play_count
FROM fact_user_actions f
JOIN dim_content d
ON f.content_id = d.content_id
WHERE f.action = 'play'
GROUP BY d.category
ORDER BY play_count DESC
LIMIT 5;
```

```
2025-03-11 00:19:58,361 Stage-3 map = 100%, reduce = 100%
Ended Job = job_local127643440_0005
MapReduce Jobs Launched:
Stage-Stage-2: HDFS Read: 78554 HDFS Write: 31044 SUCCESS
Stage-Stage-3: HDFS Read: 78554 HDFS Write: 31044 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Jazz      7
Rock      6
News      6
Reggae    5
Electronic 5
Time taken: 20.915 seconds, Fetched: 5 row(s)
hive>
```

6) Short Write-Up

- fact_user_actions is partitioned by date (year, month, day) to improve query performance
- dim_content enables joins with the fact table and contains details of content.
- Parquet used for faster for aggregations and filtering
- The Monthly Active Users by Region query took 1.876 seconds
- The Top Categories by Play Count query took 20.915 seconds
- The pipeline execution time of various steps were few seconds.
- This Hive-based data pipeline enables efficient storage and querying of large-scale user interaction data. Parquet and Partitioning significantly improves performance, but complex joins on large tables still take time. Future optimizations can further enhance query speeds.