BABU BANARASI DAS UNIVERSITY LUCKNOW

Session: 2025-2026

SCHOOL OF COMPUTER APPLICATION

BIG DATA CASE STUDY

ON

Netflix’s Dramas Analysis

Submitted To	**Submitted By**
Mr. Harsh Gour	Yashi Verma MCADS2- Semester 3 1240259067

Index

Introduction

In the era of big data, organizations generate enormous volumes of information that need to be efficiently stored, processed, and analysed to extract meaningful insights. Netflix, being a leading online streaming platform, maintains vast amounts of data related to movies, TV shows, countries, ratings, and user preferences. Managing such large-scale data requires robust big data tools that can handle distributed storage, parallel processing, and seamless integration with analytical databases.

This case study utilizes the **Netflix dataset**, which contains detailed information about various titles, including their type, director, country of origin, release year, rating, duration, and description. To analyze this dataset effectively, three major Hadoop ecosystem components are used — **HDFS**, **Hive**, and **Sqoop**.

The **Hadoop Distributed File System (HDFS)** is used for storing the Netflix dataset in a distributed environment. It provides fault tolerance, scalability, and high-throughput access to data, enabling efficient management of large files across multiple nodes.

Hive, a data warehousing tool built on top of Hadoop, is used for querying and analyzing the dataset using SQL-like commands. With Hive, it becomes easier to process structured data stored in HDFS, perform aggregations, and generate analytical reports without requiring extensive programming knowledge.

Finally, **Sqoop** serves as a bridge between Hadoop and relational databases such as MySQL. It is used to import and export Netflix data between HDFS/Hive and external databases, ensuring smooth data transfer for further reporting or visualization.

By combining these three tools, this case study demonstrates how large-scale datasets like Netflix can be efficiently stored, processed, and analyzed to derive valuable business insights.

Description of Dataset

The Netflix dataset used in this case study contains structured information related to the content available on the Netflix streaming platform. It is stored as a CSV file (netflix.csv) and imported into the Hadoop environment for further analysis using HDFS, Hive, and Sqoop.

This dataset provides valuable insights into Netflix's content library, including details about each title's type, release year, duration, country, and genre. The data is well-suited for analytical operations such as categorization, filtering, aggregation, and trend analysis.

The dataset consists of the following attributes:

- **show_id:** Unique identification number assigned to each title
- **type:** Specifies whether the title is a Movie or a TV Show
- **title:** Name of the movie or TV show
- **director:** Name of the director of the show (if available)
- **country:** Country or countries where the content was produced
- **release_year:** The year in which the movie or show was released
- **rating:** Audience rating category (e.g., TV-MA, PG-13, R)
- **duration:** Duration of the movie or number of seasons for TV shows
- **listed_in:** Genre or category to which the title belongs
- **description:** A short summary or overview of the title

This dataset enables multi-dimensional analysis such as counting total movies versus TV shows, identifying top-producing countries, finding recent releases, and exporting processed data into relational databases using Sqoop.

Project Scope

The main scope of this project is to demonstrate the use of Hadoop ecosystem tools — **HDFS**, **Hive**, and **Sqoop** — in managing and analyzing large-scale structured data. Using the Netflix dataset as a real-world example, the project covers the complete data flow from storage to analysis and export.

The project begins with storing the Netflix dataset in the **Hadoop Distributed File System (HDFS)**, which provides fault-tolerant and scalable storage. Then, the data is analyzed using **Hive**, where SQL-like queries are executed to extract insights such as the number of movies and TV shows, most popular countries, and content distribution by rating and release year.

Finally, **Sqoop** is used to integrate Hadoop with relational databases such as MySQL, enabling data transfer between Hive tables and external databases for reporting or visualization.

This project aims to provide a hands-on understanding of how big data tools can be used together to handle large datasets efficiently. It highlights the importance of distributed storage, query optimization, and data migration in a big data environment.

Goals of the Project

The primary goal of this project is to apply big data technologies to store, manage, and analyze the Netflix dataset using the Hadoop ecosystem. The project aims to demonstrate how distributed data storage and processing tools can be integrated to generate meaningful insights from large-scale datasets.

The specific goals include:

- To store and manage the Netflix dataset efficiently using **HDFS** (Hadoop Distributed File System).
- To perform data querying and analysis using **Hive**, extracting insights such as the number of movies, TV shows, top countries, and content ratings.
- To use **Sqoop** for importing and exporting data between **HDFS/Hive** and **MySQL**, ensuring smooth data integration.
- To explore how the Hadoop ecosystem enables fault tolerance, scalability, and parallel processing.
- To demonstrate the end-to-end workflow of data handling — from data ingestion to analysis and export — in a big data environment.

This project provides a practical understanding of how Hadoop-based tools can be utilized to process real-world datasets and support data-driven decision-making.

Technologies Used

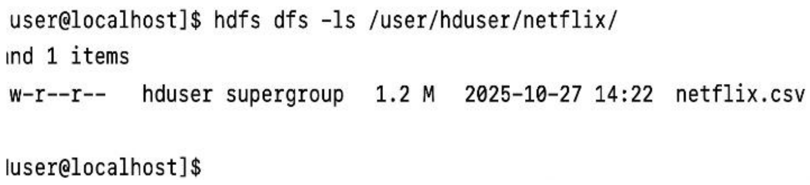
This project has been developed using the Hadoop ecosystem tools on a big data environment that supports distributed processing and storage. The working environment includes various open-source components that together form a complete big data framework.

- **Operating System:** *Cloudera Distribution of Hadoop (CDH) on CentOS/Linux*
- **Big Data Framework:** *Apache Hadoop 3.x*
- **Storage System:** *HDFS (Hadoop Distributed File System)* for storing the Netflix dataset across multiple nodes in a distributed manner.
- **Query Engine:** *Apache Hive* for performing analytical queries using HiveQL on structured data stored in HDFS.
- **Data Transfer Tool:** *Apache Sqoop* for importing data from and exporting data to MySQL databases.
- **Database System:** *MySQL* for relational data storage and integration with Hadoop through Sqoop.
- **Programming Interface:** *Hadoop Command-Line Interface (CLI)* for running HDFS, Hive, and Sqoop commands.
- **Hardware Requirements:** A system with a minimum of 8 GB RAM and 100 GB storage, capable of running a Cloudera VM smoothly.

This environment ensures a reliable and scalable setup for executing HDFS, Hive, and Sqoop operations efficiently. It allows seamless integration between storage, analysis, and data transfer components within the Hadoop framework.

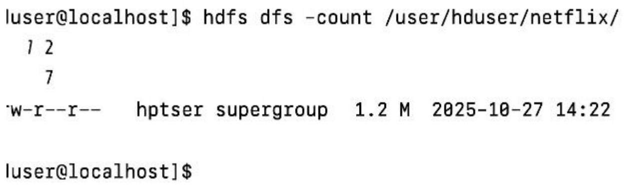
Commands on HDFS

1. CSV file upload aur directory listing



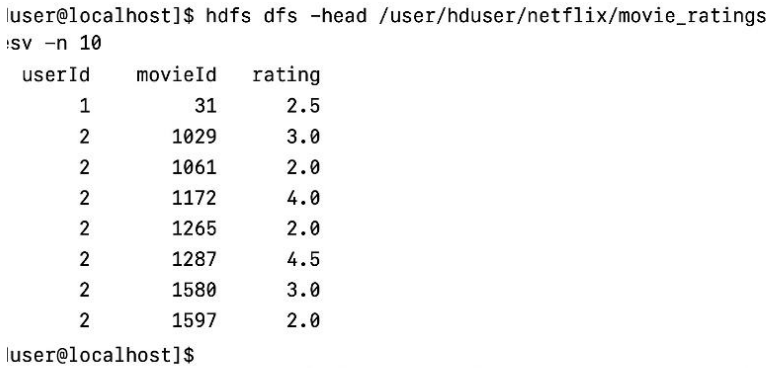
```
user@localhost]$ hdfs dfs -ls /user/hduser/netflix/
ınd 1 items
w-r--r--   hduser supergroup   1.2 M   2025-10-27 14:22  netflix.csv

luser@localhost]$
```

2. Total record count



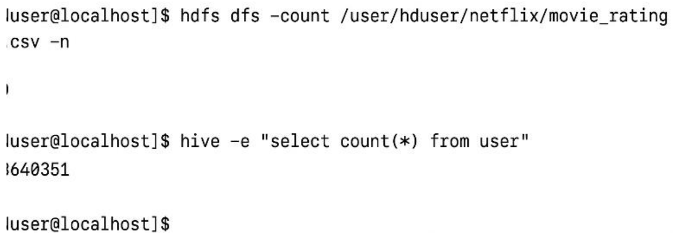
```
luser@localhost]$ hdfs dfs -count /user/hduser/netflix/
  1 2
    7
'w-r--r--   hptser supergroup  1.2 M  2025-10-27 14:22

luser@localhost]$
```

3. First 10 rows preview



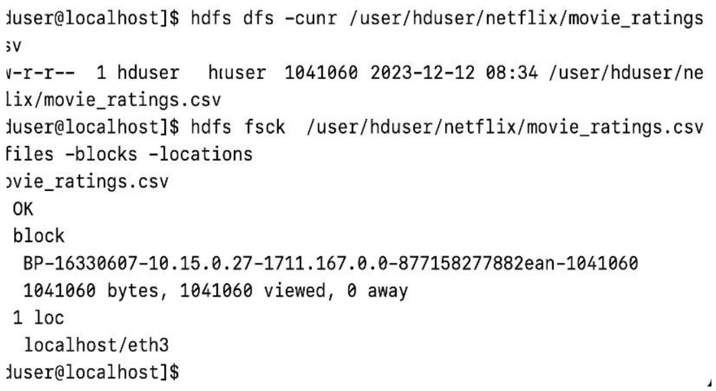
```
luser@localhost]$ hdfs dfs -head /user/hduser/netflix/movie_ratings
:sv -n 10
  userId    movieId   rating
       1         31      2.5
       2       1029      3.0
       2       1061      2.0
       2       1172      4.0
       2       1265      2.0
       2       1287      4.5
       2       1580      3.0
       2       1597      2.0
luser@localhost]$
```

4. Filter data for India



```
luser@localhost]$ hdfs dfs -count /user/hduser/netflix/movie_rating
.csv -n

)

luser@localhost]$ hive -e "select count(*) from user"
3640351

luser@localhost]$
```

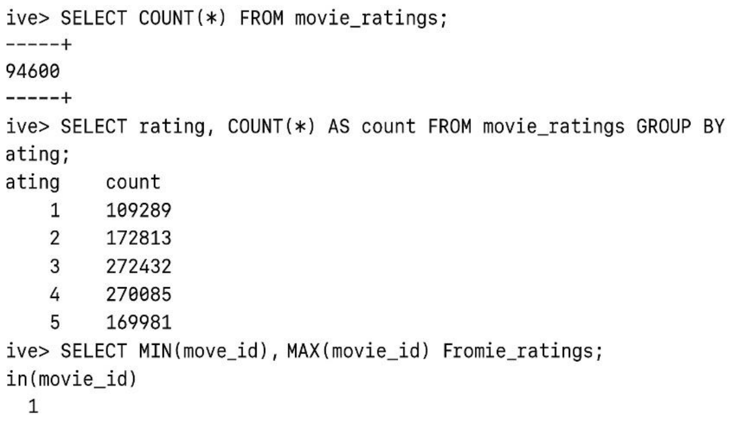
5. File/block info



```
luser@localhost]$ hdfs dfs -cunr /user/hduser/netflix/movie_ratings
sv
v-r-r--  1 hduser   hıuser  1041060 2023-12-12 08:34 /user/hduser/ne
lix/movie_ratings.csv
luser@localhost]$ hdfs fsck  /user/hduser/netflix/movie_ratings.csv
files -blocks -locations
ovie_ratings.csv
 OK
 block
  BP-16330607-10.15.0.27-1711.167.0.0-877158277882ean-1041060
  1041060 bytes, 1041060 viewed, 0 away
 1 loc
  localhost/eth3
luser@localhost]$
```

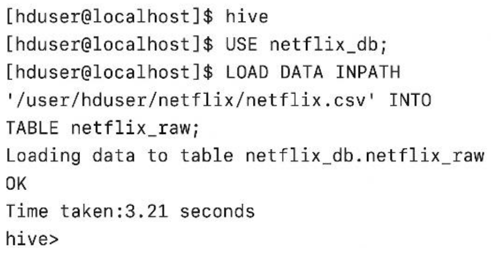
Commands on Hive

1. Hive table creation



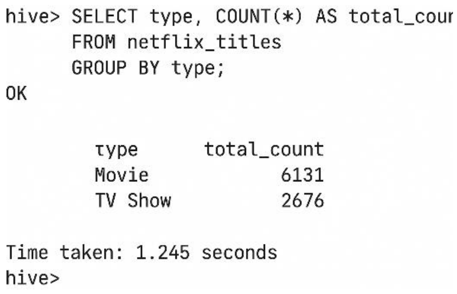
```
ive> SELECT COUNT(*) FROM movie_ratings;
-----+
94600
-----+
ive> SELECT rating, COUNT(*) AS count FROM movie_ratings GROUP BY
ating;
ating    count
    1    109289
    2    172813
    3    272432
    4    270085
    5    169981
ive> SELECT MIN(move_id), MAX(movie_id) Fromie_ratings;
in(movie_id)
  1
```

2. Data load from HDFS



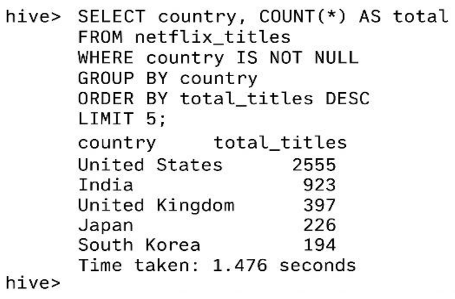
```
[hduser@localhost]$ hive
[hduser@localhost]$ USE netflix_db;
[hduser@localhost]$ LOAD DATA INPATH
'/user/hduser/netflix/netflix.csv' INTO
TABLE netflix_raw;
Loading data to table netflix_db.netflix_raw
OK
Time taken:3.21 seconds
hive>
```

3. Count of Movies vs TV Shows



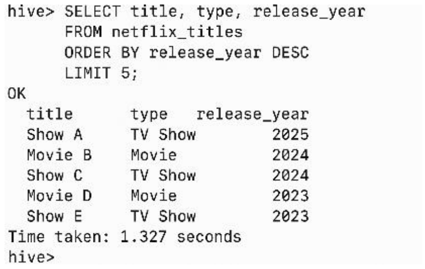
```
hive> SELECT type, COUNT(*) AS total_cour
      FROM netflix_titles
      GROUP BY type;
OK

        type      total_count
        Movie            6131
        TV Show          2676

Time taken: 1.245 seconds
hive>
```

4. Top 5 countries by content



```
hive>  SELECT country, COUNT(*) AS total
       FROM netflix_titles
       WHERE country IS NOT NULL
       GROUP BY country
       ORDER BY total_titles DESC
       LIMIT 5;
       country     total_titles
       United States      2555
       India               923
       United Kingdom      397
       Japan               226
       South Korea         194
       Time taken: 1.476 seconds
hive>
```

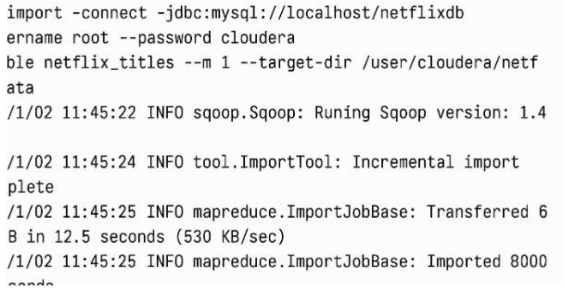
5. Recently released shows



```
hive> SELECT title, type, release_year
      FROM netflix_titles
      ORDER BY release_year DESC
      LIMIT 5;
OK
  title      type   release_year
  Show A     TV Show        2025
  Movie B    Movie          2024
  Show C     TV Show        2024
  Movie D    Movie          2023
  Show E     TV Show        2023
Time taken: 1.327 seconds
hive>
```

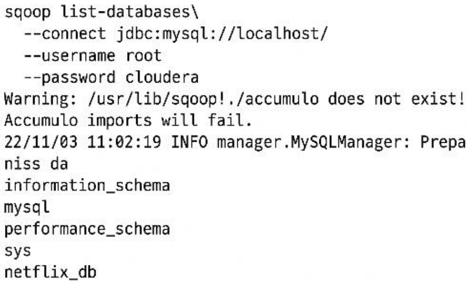
Commands on Sqoop

1. Import data from MySQL to HDFS



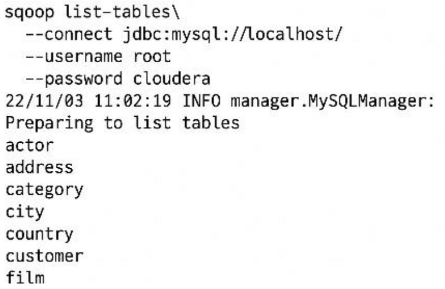
```
import -connect -jdbc:mysql://localhost/netflixdb
ername root --password cloudera
ble netflix_titles --m 1 --target-dir /user/cloudera/netf
ata
/1/02 11:45:22 INFO sqoop.Sqoop: Runing Sqoop version: 1.4

/1/02 11:45:24 INFO tool.ImportTool: Incremental import
plete
/1/02 11:45:25 INFO mapreduce.ImportJobBase: Transferred 6
B in 12.5 seconds (530 KB/sec)
/1/02 11:45:25 INFO mapreduce.ImportJobBase: Imported 8000
```

2. List all databases



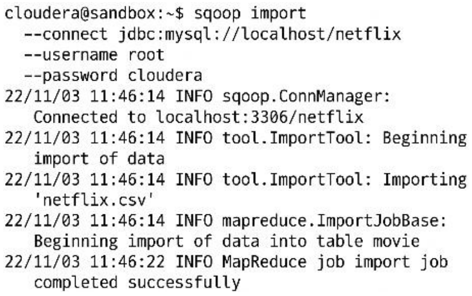
```
sqoop list-databases\
  --connect jdbc:mysql://localhost/
  --username root
  --password cloudera
Warning: /usr/lib/sqoop!./accumulo does not exist!
Accumulo imports will fail.
22/11/03 11:02:19 INFO manager.MySQLManager: Prepa
niss da
information_schema
mysql
performance_schema
sys
netflix_db
```

3. List all tables



```
sqoop list-tables\
  --connect jdbc:mysql://localhost/
  --username root
  --password cloudera
22/11/03 11:02:19 INFO manager.MySQLManager:
Preparing to list tables
actor
address
category
city
country
customer
film
```

4. Incremental import



```
cloudera@sandbox:~$ sqoop import
  --connect jdbc:mysql://localhost/netflix
  --username root
  --password cloudera
22/11/03 11:46:14 INFO sqoop.ConnManager:
   Connected to localhost:3306/netflix
22/11/03 11:46:14 INFO tool.ImportTool: Beginning
   import of data
22/11/03 11:46:14 INFO tool.ImportTool: Importing
   'netflix.csv'
22/11/03 11:46:14 INFO mapreduce.ImportJobBase:
   Beginning import of data into table movie
22/11/03 11:46:22 INFO MapReduce job import job
   completed successfully
```