



CURSO DE ESPECIALIZACIÓN EN INTELIGENCIA ARTIFICIAL Y BIGDATA

BIGDATA APLICADO

TAREA EVALUABLE 4.1

Autor: Carlos Sánchez Recio.
06 / 01 / 2025

Índice

Apartado 1: Películas y programas de Netflix	1
1 Listado de los 10 programas de TV con más de una temporada con mejor valoración, ordenados por valoración en orden decreciente.	2
2 Listado de los 10 años en los que sus programas de TV (según el año de lanzamiento) han tenido más votos, ordenados por número de votos, en orden decreciente.	2
3 Listado de los 10 directores con más películas, ordenados por número de películas en orden decreciente.	3
4 Listado de los 10 actores con mejor valoración media de sus películas, ordenados por valoración media en orden decreciente.	4

Apartado 1: Películas y programas de Netflix

En la actividad de aprendizaje de la entrega 3 (y en algunos ejemplos de esta entrega) trabajamos con el dataset de las películas y programas de TV mejor valoradas de Netflix, que puede encontrar en [Kaggle](#).

Recordemos que de los seis archivos que consta el dataset, nos interesan sólo dos:

- `raw_titles.csv`: que contiene la información de las películas (movies) y programas de TV (shows), incluyendo las series, de la plataforma Netflix, con el número de votos y puntuación en IMDB.
- `raw_credits.csv` que contiene la información de los actores y directores de todas las películas y programas

(Note:) también puedes descargar los archivos desde el repositorio del curso, donde ya se han empleado tabuladores como separadores de campos para evitar problemas con las importaciones: `raw_titles.csv` y `raw_credits.csv`.

Los archivos han sido descargados desde la URL del repositorio del curso y subidos a Hadoop mediante la siguiente secuencia de comandos:

```
1 wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/
2   refs/heads/main/raw_titles.csv
3 wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/
4   refs/heads/main/raw_credits.csv
5 hdfs dfs -mkdir movies
6 hdfs dfs -put raw_titles.csv movies/
7 hdfs dfs -put raw_credits.csv movies/
```

Una vez obtenidos los archivos y cargados en Hadoop, se crean las tablas con los datos de éstos mediante las siguientes sentencias sustituyendo la ruta a los archivos:

```
1 CREATE DATABASE movies;
2 USE movies;
3
4 CREATE TABLE titles(
5   index INT,
6   id STRING,
7   title STRING,
8   type STRING,
9   release_year INT,
10  age_certification STRING,
11  runtime INT,
12  genres STRING,
13  production_countries STRING,
14  seasons FLOAT,
15  imdb_id STRING,
16  imdb_score FLOAT,
17  imdb_votes FLOAT)
18 ROW FORMAT DELIMITED
19 FIELDS TERMINATED BY '\t'
20 TBLPROPERTIES ("skip.header.line.count"="1");
21
22 CREATE TABLE credits(
23   index INT,
24   person_id INT,
25   id STRING,
26   name STRING,
27   character STRING,
```

```

28  role STRING)
29  ROW FORMAT DELIMITED
30  FIELDS TERMINATED BY '\t'
31  TBLPROPERTIES ("skip.header.line.count"="1");
32
33  LOAD DATA LOCAL INPATH '/path/to/file/raw_titles.csv' INTO TABLE titles;
34  LOAD DATA LOCAL INPATH '/path/to/file/raw_credits.csv' INTO TABLE credits;

```

De esta forma los datos estarían cargados en las tablas. Este proceso se realizó en el bloque anterior. Los pasos necesarios para pasar los datos a Impala son los siguientes:

```

1  CREATE TABLE titles_parquet
2  STORED AS PARQUET
3  AS SELECT * FROM titles;
4
5  CREATE TABLE credits_parquet
6  STORED AS PARQUET
7  AS SELECT * FROM credits;

```

Como analistas de datos nos han pedido una serie de preguntas que debemos responder utilizando Apache Impala, ya sea desde Hue o desde el Shell. Son estas:

- 1 Listado de los 10 programas de TV con más de una temporada con mejor valoración, ordenados por valoración en orden decreciente.

```

1  SELECT *
2  FROM titles_parquet
3  WHERE seasons > 1 AND type = 'SHOW' AND imdb_score IS NOT NULL
4  ORDER BY imdb_score DESC
5  LIMIT 10;

```

index	id	title	type	release_year
1	656	ts160526	Khawatir	SHOW 2005
2	243	ts4	Breaking Bad	SHOW 2008
3	3827	ts90621	Kota Factory	SHOW 2019
4	259	ts3371	Avatar: The Last Airbender	SHOW 2005
5	1099	ts121189	Raja, Rasoi Aur Anya Kahaniyaan	SHOW 2014
6	917	ts20682	Attack on Titan	SHOW 2013
7	1263	ts52922	Leah Remini: Scientology and the Aftermath	SHOW 2016
8	717	ts32835	Hunter x Hunter	SHOW 2011
9	47	ts20681	Seinfeld	SHOW 1989
10	367	ts24028	Still Game	SHOW 2002

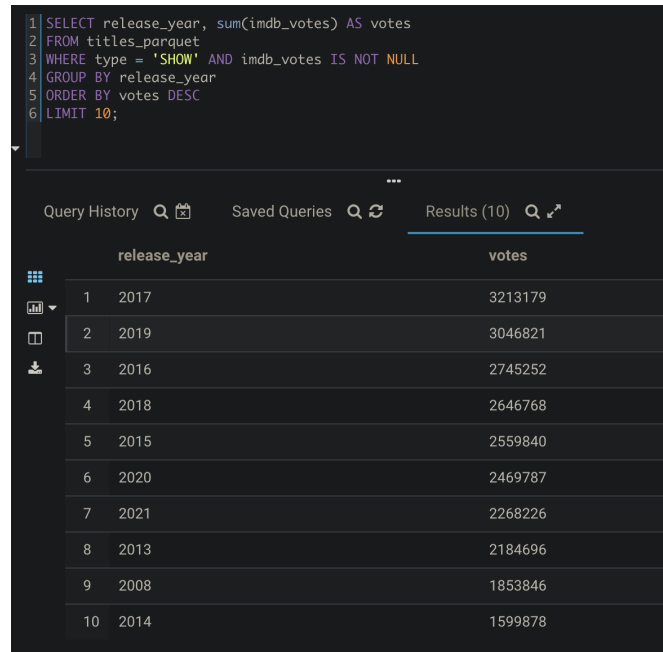
- 2 Listado de los 10 años en los que sus programas de TV (según el año de lanzamiento) han tenido más votos, ordenados por número de votos, en orden decreciente.

```

1  SELECT release_year, sum(imdb_votes) AS votes
2  FROM titles_parquet

```

```
3 WHERE type = 'SHOW' AND imdb_votes IS NOT NULL
4 GROUP BY release_year
5 ORDER BY votes DESC
6 LIMIT 10;
```



The screenshot shows a SQL query editor with a dark theme. The query is: `SELECT release_year, sum(imdb_votes) AS votes FROM titles_parquet WHERE type = 'SHOW' AND imdb_votes IS NOT NULL GROUP BY release_year ORDER BY votes DESC LIMIT 10;`. Below the query, there is a toolbar with 'Query History', 'Saved Queries', and 'Results (10)'. The results are displayed in a table with two columns: 'release_year' and 'votes'. The table contains 10 rows of data, ordered by votes in descending order.

	release_year	votes
1	2017	3213179
2	2019	3046821
3	2016	2745252
4	2018	2646768
5	2015	2559840
6	2020	2469787
7	2021	2268226
8	2013	2184696
9	2008	1853846
10	2014	1599878

3 Listado de los 10 directores con más películas, ordenados por número de películas en orden decreciente.

```
1 SELECT name, count(name) AS films_num
2 FROM credits_parquet
3 WHERE 'role' = 'DIRECTOR'
4 GROUP BY name
5 ORDER BY films_num DESC
6 LIMIT 10;
```

```

1 SELECT name, count(name) AS films_num
2 FROM credits_parquet
3 WHERE 'role' = 'DIRECTOR'
4 GROUP BY name
5 ORDER BY films_num DESC
6 LIMIT 10;

```

	name	films_num
1	Raúl Campos	21
2	Jan Suter	20
3	Jay Karas	16
4	Marcus Raboy	15
5	Ryan Polito	13
6	Jay Chapman	12
7	Cathy Garcia-Molina	12
8	Youssef Chahine	11
9	Justin G. Dyck	9
10	Umesh Mehra	8

4 Listado de los 10 actores con mejor valoración media de sus películas, ordenados por valoración media en orden decreciente.

```

1 SELECT c.name, AVG(t.imdb_score) AS imdb_avg
2 FROM credits_parquet AS c
3 JOIN titles_parquet AS t ON c.id = t.id
4 WHERE c.'role' = 'ACTOR' AND t.imdb_score IS NOT NULL
5 GROUP BY c.name
6 ORDER BY imdb_avg DESC
7 LIMIT 10;

```

```

1 SELECT c.name, AVG(t.imdb_score) AS imdb_avg
2 FROM credits_parquet AS c
3 JOIN titles_parquet AS t ON c.id = t.id
4 WHERE c.'role' = 'ACTOR' AND t.imdb_score IS NOT NULL
5 GROUP BY c.name
6 ORDER BY imdb_avg DESC
7 LIMIT 10;

```

	name	imdb_avg
1	Anna Gunn	9.5
2	Betsy Brandt	9.5
3	Zach Tyler	9.3000001907348633
4	Jessie Flower	9.3000001907348633
5	Cricket Leigh	9.3000001907348633
6	Lee Ji-ah	9.199998092651367
7	Son Sook	9.199998092651367
8	Zhang Feng Yi	9.199998092651367
9	He Kailang	9.199998092651367
10	Kim Seol	9.199998092651367