REGISTER /usr/lib/pig/piggybank.jar;

prime = LOAD 'pig/prime.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') AS (url:chararray, title:chararray, type:chararray, genres:chararray, releaseYear:float, imdbId:chararray, imdbAverageRating:float, imdbNumVotes:float, availableCountries:chararray);

1. Mostrar el número de filas de datos que hay en total:  
     
   c\_prime= FILTER prime BY title IS NOT NULL;

total\_filas = FOREACH (GROUP c\_prime ALL) GENERATE COUNT(c\_prime);

DUMP total\_filas;

Resultado: 47167



1. Mostrar el número de series:  
     
   series = FILTER c\_prime BY type == ‘TV Series’;

total\_series = FOREACH (GROUP series ALL) GENERATE COUNT(series);

DUMP total\_series;

Resultado: 8332

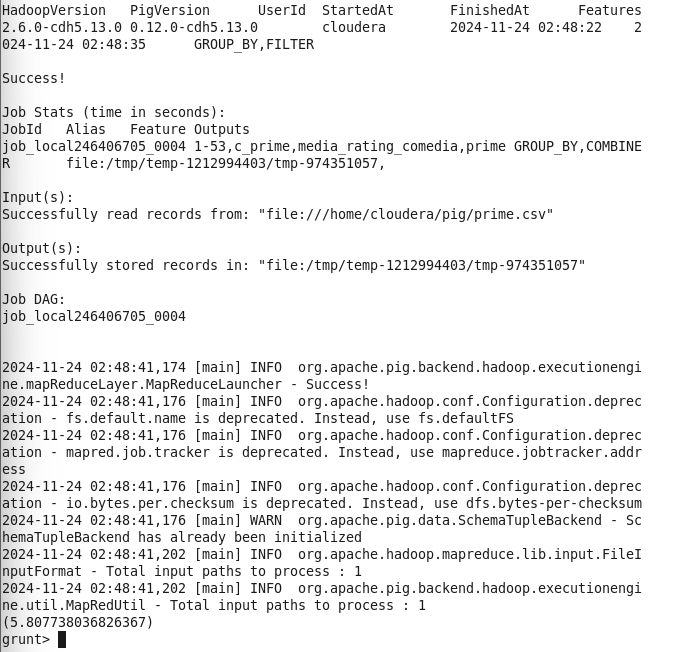


1. Mostrar la media de valoración en IMDB de las películas que solo tengan el género comedia:  
     
   comedia = FILTER c\_prime BY type == ‘Movie’ AND genres MATCHES ‘.\*Comedy.\*’*;*

media\_rating\_comedia = FOREACH (GROUP comedia ALL) GENERATE AVG(comedia.imdbAverageRating);

DUMP media\_rating\_comedia;

Resultado: 5.8077



1. Mostrar el total de votos en IMDB de las peliculas disponibles en España:

pelicula\_es = FILTER c\_prime BY type == ‘Movie’ AND availableCountries MATCHES ‘.\*ES.\*;

contador\_p\_es = FOREACH (GROUP pelicula\_es ALL) GENERATE COUNT(pelicula\_es);

DUMP contador\_p\_es;

Resultado: 3543

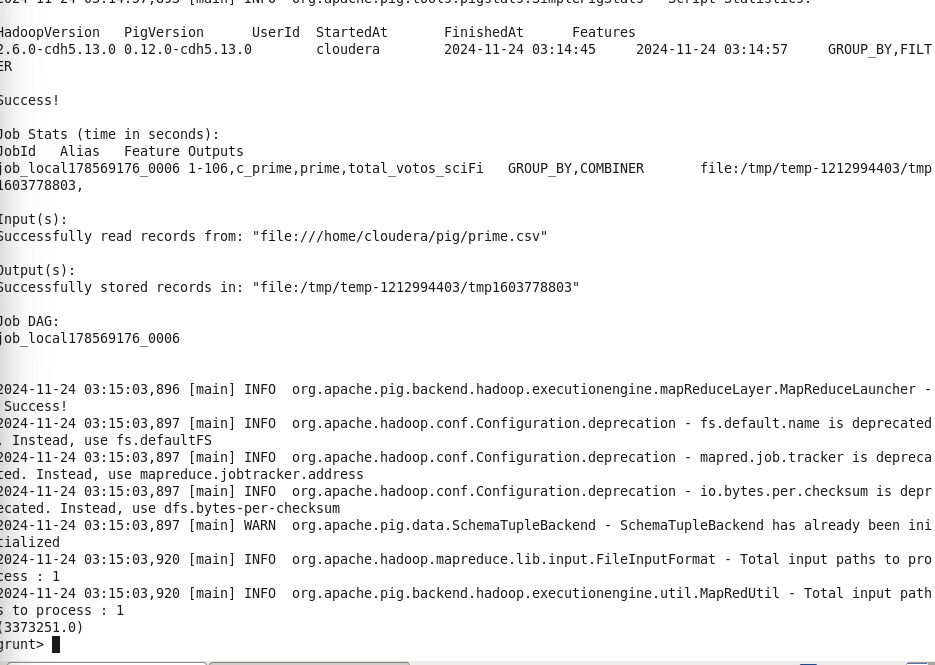


1. Mostrar el total de votos de IMDB en series de ciencia ficción:  
     
   sciFi = FILTER c\_prime BY type == ‘TV Series’ AND genres MATCHES ‘.\*Sci-Fi.\*;

total\_votos\_sciFi = FOREACH (GROUP sciFI ALL) GENERATE SUM(sciFi.imdbNumVotes);

DUMP total\_votos\_sciFi;

Resultado: 3373251



1. Mostrar la valoración media en IMDB de todas las series y películas del año 2024:

cont\_2024 = FILTER c\_prime BY releaseYear == 2024;

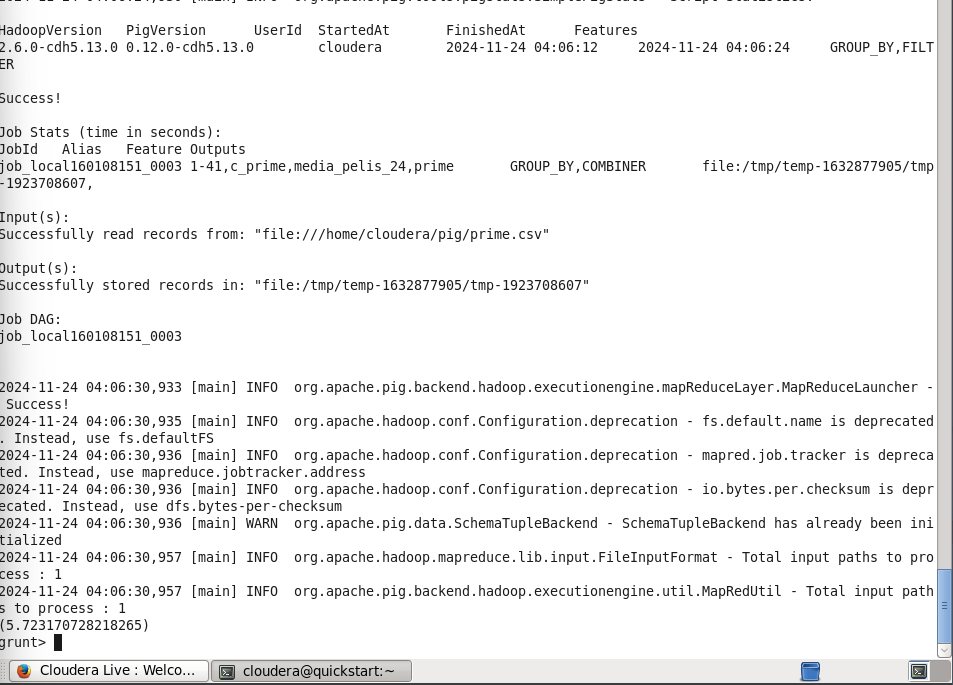
pelis\_2024 = FILTER cont\_2024 BY type == ‘Movie’;

series\_2024 = FILTER cont\_2024 BY type == ‘TV series’;

media\_pelis\_24 = FOREACH (GROUP pelis\_2024 ALL) GENERATE AVG(pelis\_2024.imdbAverageRating);

DUMP media\_pelis\_24;

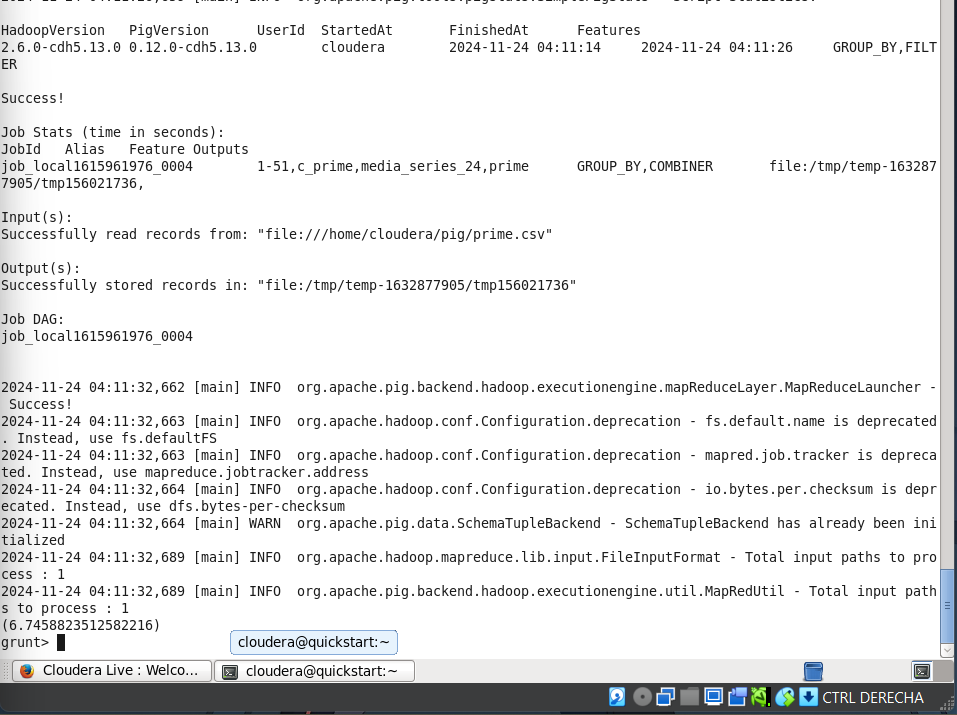
Resultado: 5.7231



media\_series\_24 = FOREACH (GROUP series\_2024 ALL) GENERATE AVG(series\_2024.imdbAverageRating);

DUMP media\_series\_24;

Resultado: 6.74588



1. Mostrar los 10 años en que más películas han sido lanzadas;  
     
   peliculas = FILTER c\_prime BY type == ‘Movie’;

peliculas\_x\_anyo = GROUP peliculas BY releaseYear;

cont\_x\_anyo = FOREACH peliculas\_x\_anyos GENERATE group AS year, COUNT(peliculas) AS count;

top\_10\_anyos = LIMIT (ORDER cont\_x\_anyo BY count DESC) 10;

DUMP top\_10\_anyos;

Resultado:

(2017.0, 2643)

(2016.0, 2425)

(2018.0, 2358)

(2015.0, 2252)

(2014.0, 2073)

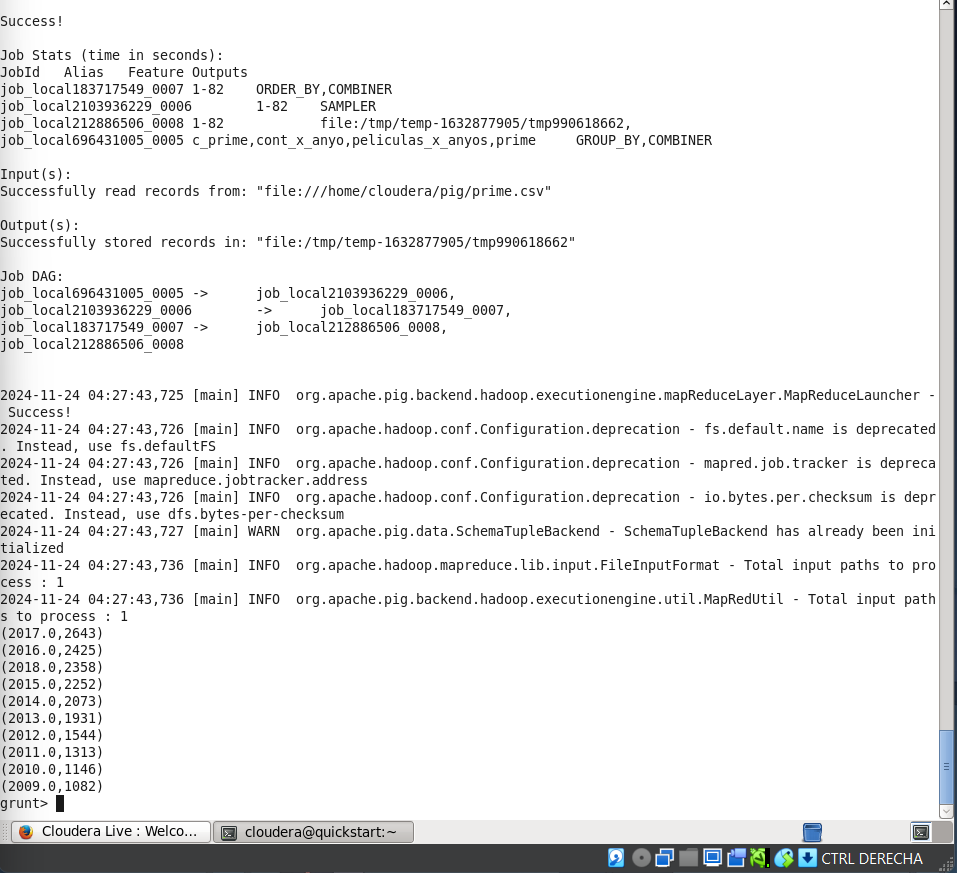
(2013.0, 1931)

(2012.0, 1544)

(2011.0, 1313)

(2010.0, 1146)

(2009.0, 1082)



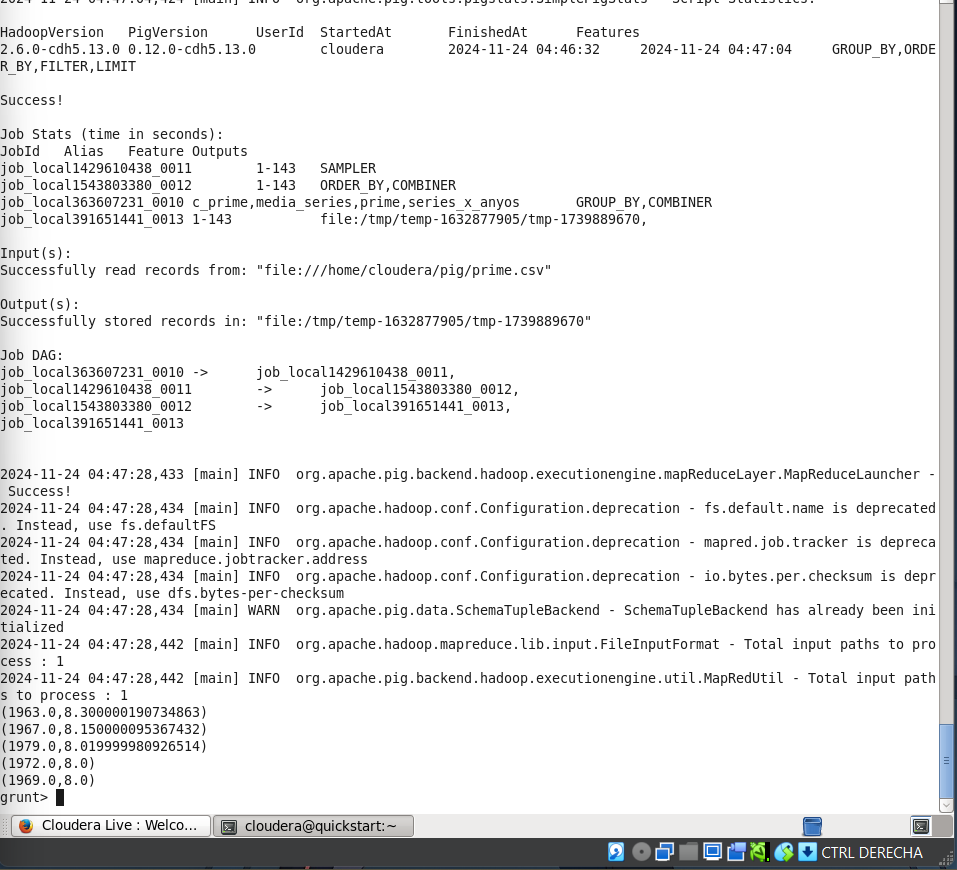
1. Mostrar los 5 años que tienen una mejor valoración en IMDB de series de drama:  
     
   series\_drama = FILTER c\_prime BY type == ‘TV Series’ AND genres MATCHES ‘.\*Drama.\*;  
     
   series\_x\_anyos = GROUP series\_drama BY releaseYear;  
     
   media\_series = FOREACH series\_x\_anyos GENERATE group AS year, AVG(series\_drama.imdbAverageRating) AS avg\_rating;  
     
   top5 = LIMIT (ORDER media\_series BY avg\_rating DESC) 5;  
     
   DUMP top5;  
     
   Resultado:  
   (1963.0, 8.3000)

(1967.0, 8.1500)

(1979.0, 8.0199)

(1972.0, 8.0)

(1969.0, 8.0)



1. Exportar a un fichero csv con los requisitos requeridos:

series\_recientes = FILTER c\_prime BY type == ‘TV Series’ AND releaseYear >= 2020;

top10\_series = LIMIT (ORDER series\_recientes BY imdbAverageRating DESC) 10;

export\_datos = FOREACH top10\_series GENERATE title, releaseYear, imdbAverageRating, imdbNumVotes;

STORE export\_datos INTO ‘pig/top10\_series.csv’ USING PigStorage(‘,’);



1. Cargar el fichero exportado y mostrar el total de votos en IMDB de las series que contiene:

datos\_exportados = LOAD 'pig/top10\_series.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') AS (title:chararray, releaseYear:float, imdbAverageRating:float, imdbNumVotes:float);

top10\_filtro = FILTER datos\_exportados BY imdbNumVotes IS NOT NULL;  
  
total\_votos = FOREACH (GROUP top10\_filtro ALL) GENERATE SUM(top10\_filtro.imdbNumVotes);

DUMP total\_votos;

Resultado: Me da un resultado el cual seguramente esté equivocado el cual es (314802.39999961853):

