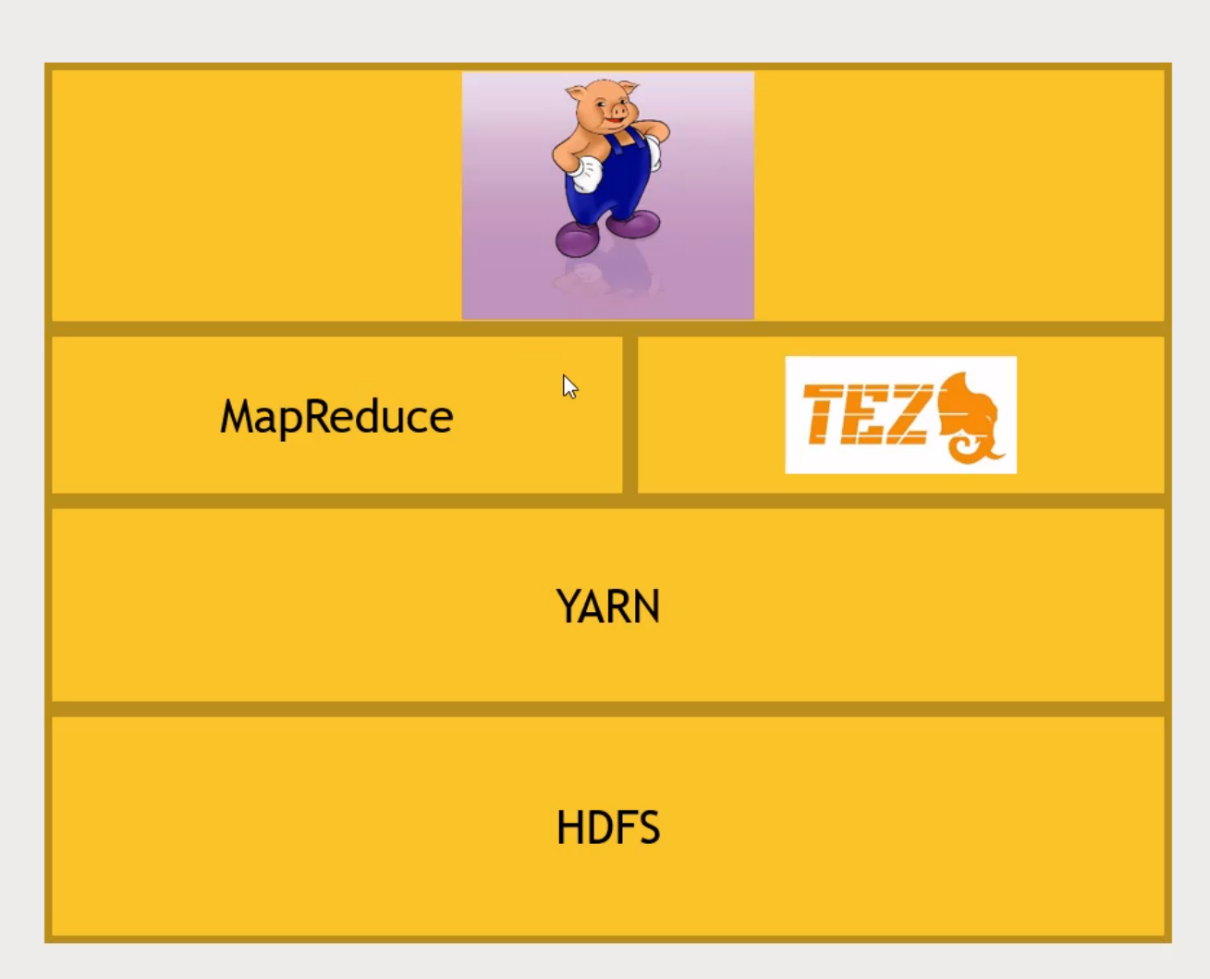
Pig – Scripting language, Create map reduce jobs without having to write Mapping and Reducing Functions

* Writing Mappers and reducers by hand takes a long time
* Pig Introduces Pig Latin, a scripting language that lets you use SQL like syntax to define your map and reduce steps
* Highly extensible with user-defined functions (UDF’s)

You can have custom function into Pig. Pig sits on top of Map Reduce which is on top of Yarn which is on top of HDFS

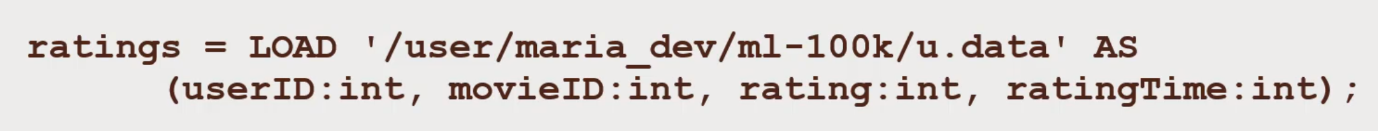
TEZ – organizing the jobs. Look at interdependency of the jobs and find the optimal way of getting it done

Running Pig

* Grunt (command line prompt)
* Script file name (run it from the command line)
* Ambari/Hue Right from the web browser

Find the Oldest 5-star Movies

Pig script, create a dataset.

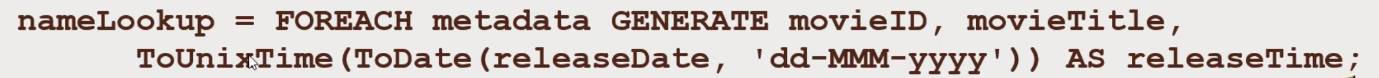
AS give it a schema. Load in Plain old Text data, and use AS as names and types.

We were writing a mapper.

Load the u.itemfile. Delimitor of | character.

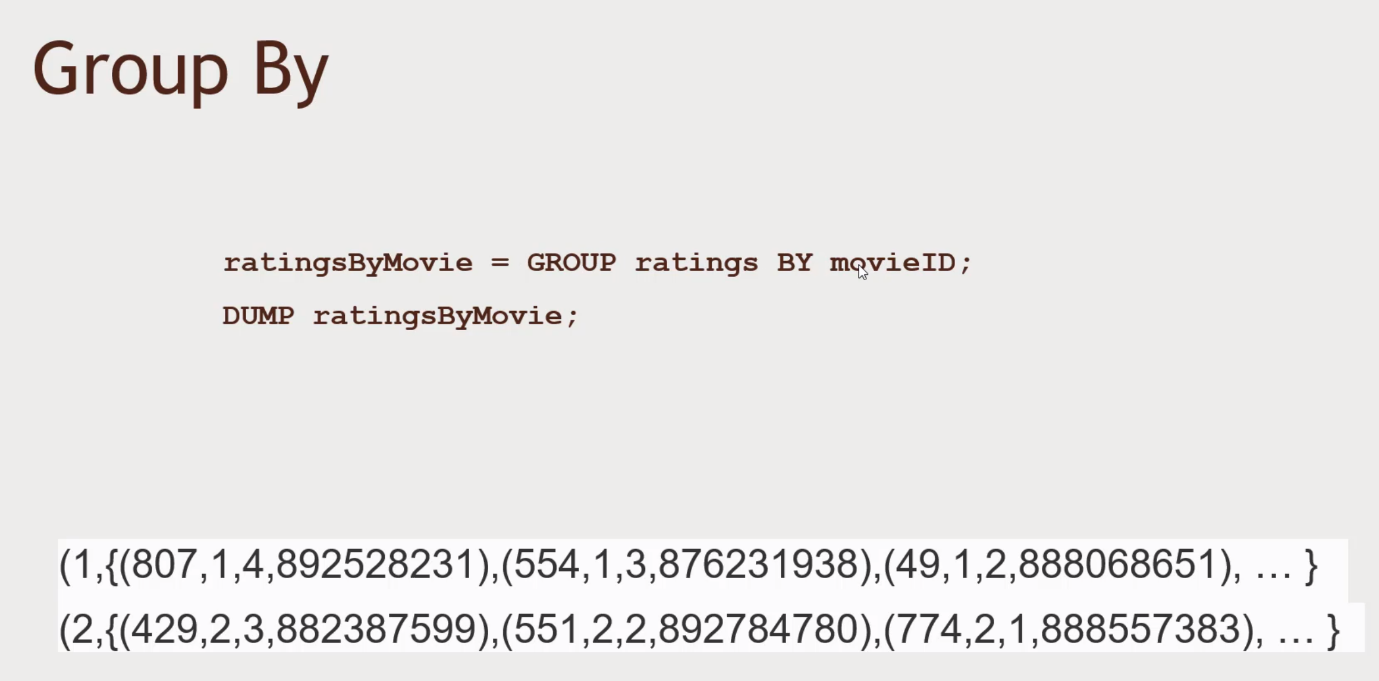
DUMP command DUMP out the contents of the entire relation. Give you a little view of what are the data consist of and etc.

Transform that movie info into something that you can actually use

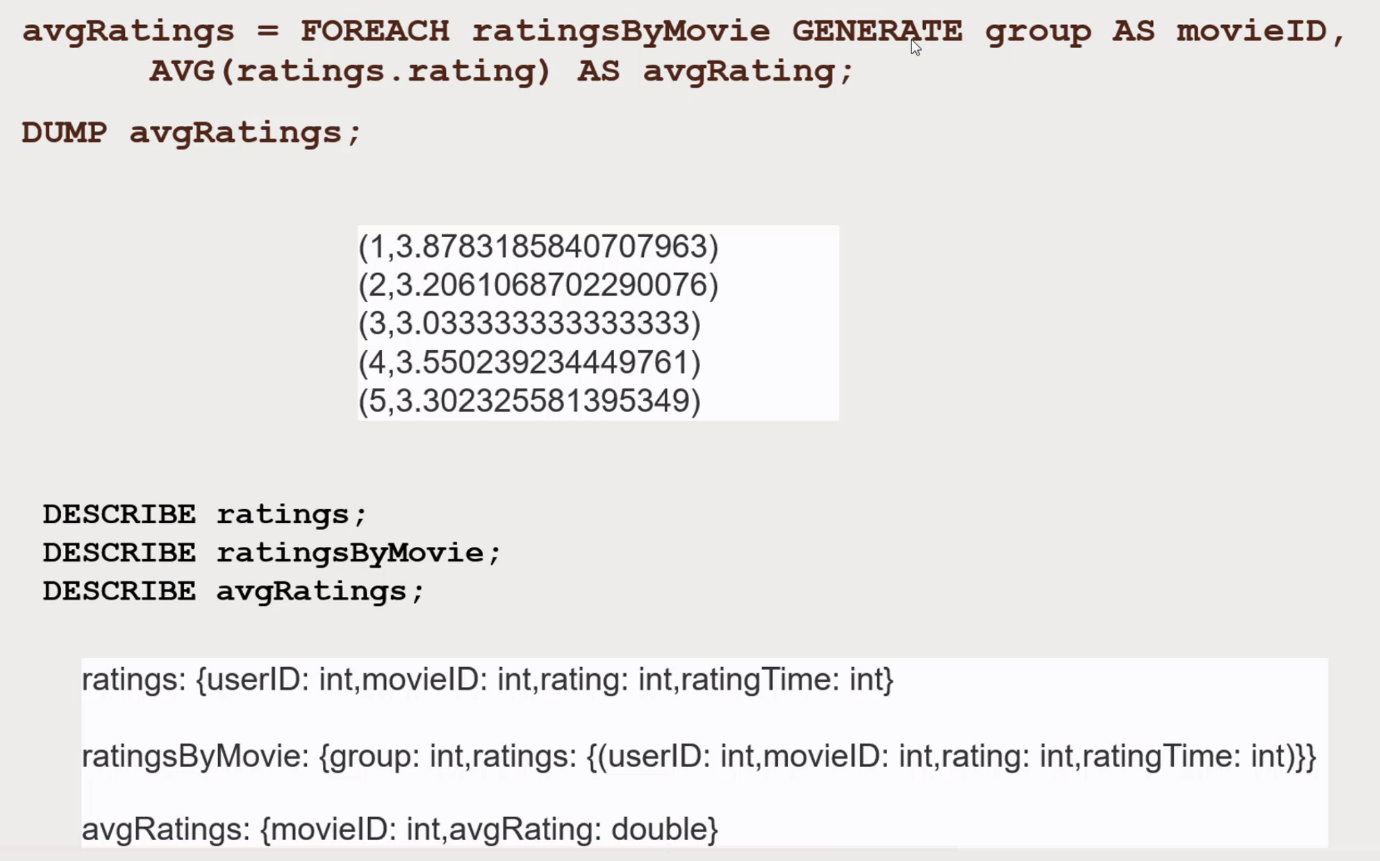


Apply the date time function to transform the date month year into a UnixTime stamp, seconds since 1st Jan 1970.

Gives us the ability to join.

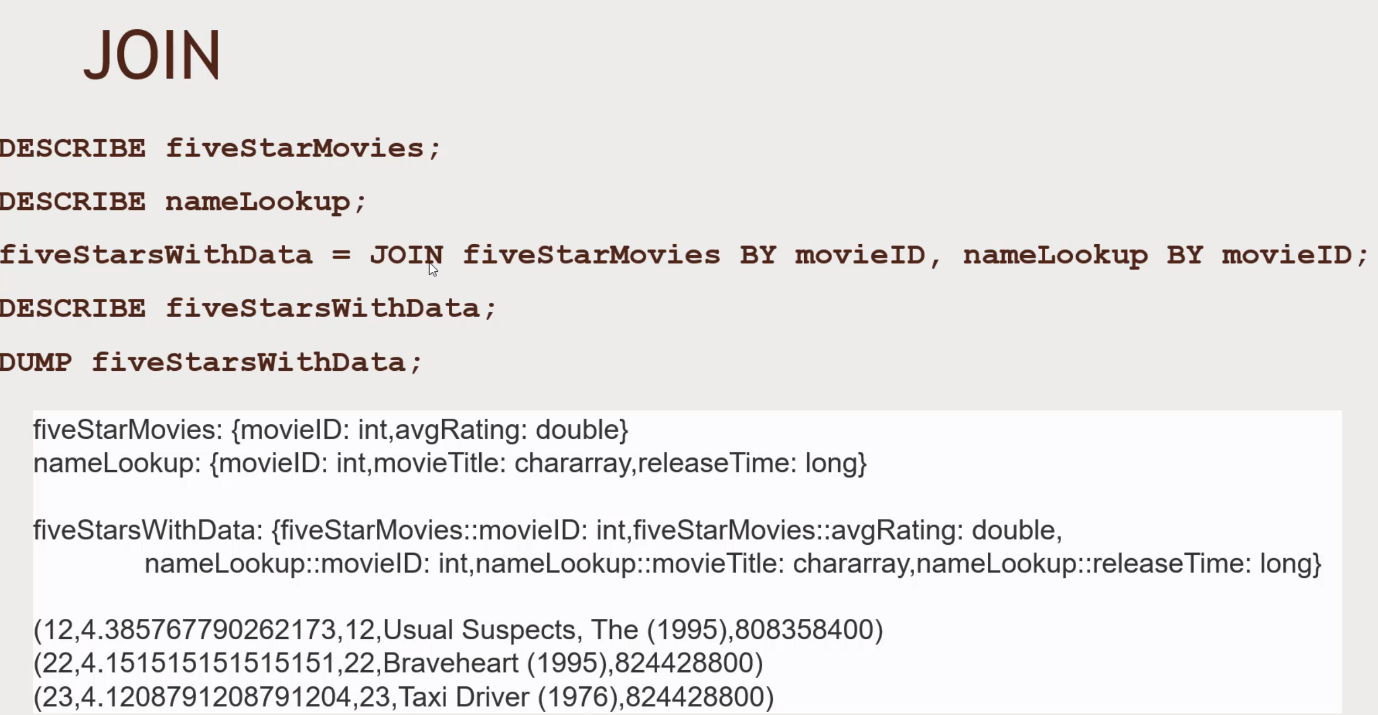
Figure out all the associated with each individual movie. Creates a bag that contains that contains the tuple of rows which contains the movie ID. It looks like a Reduce operation.

Compute average ratings

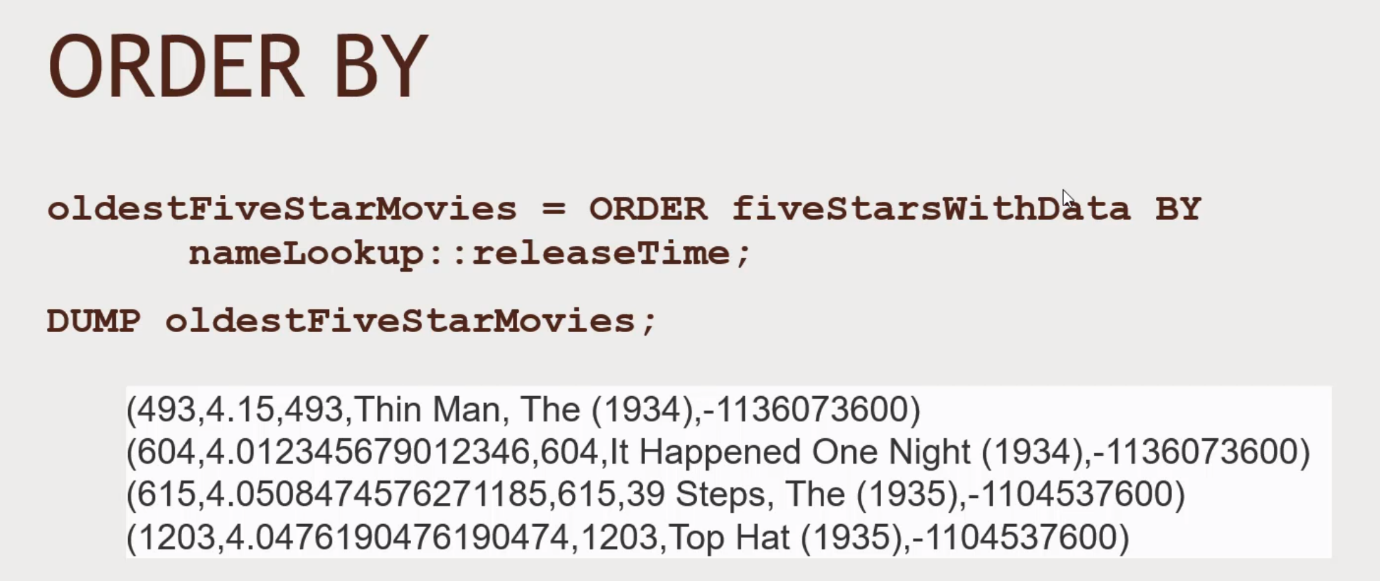
For each ratingsByMovie, generate new rows, generate a group as movieId, Average Rating.

Group of MovieIDs and rename that as MovieID. Contains the bags of rating, plug out all the rows of each individual rating.

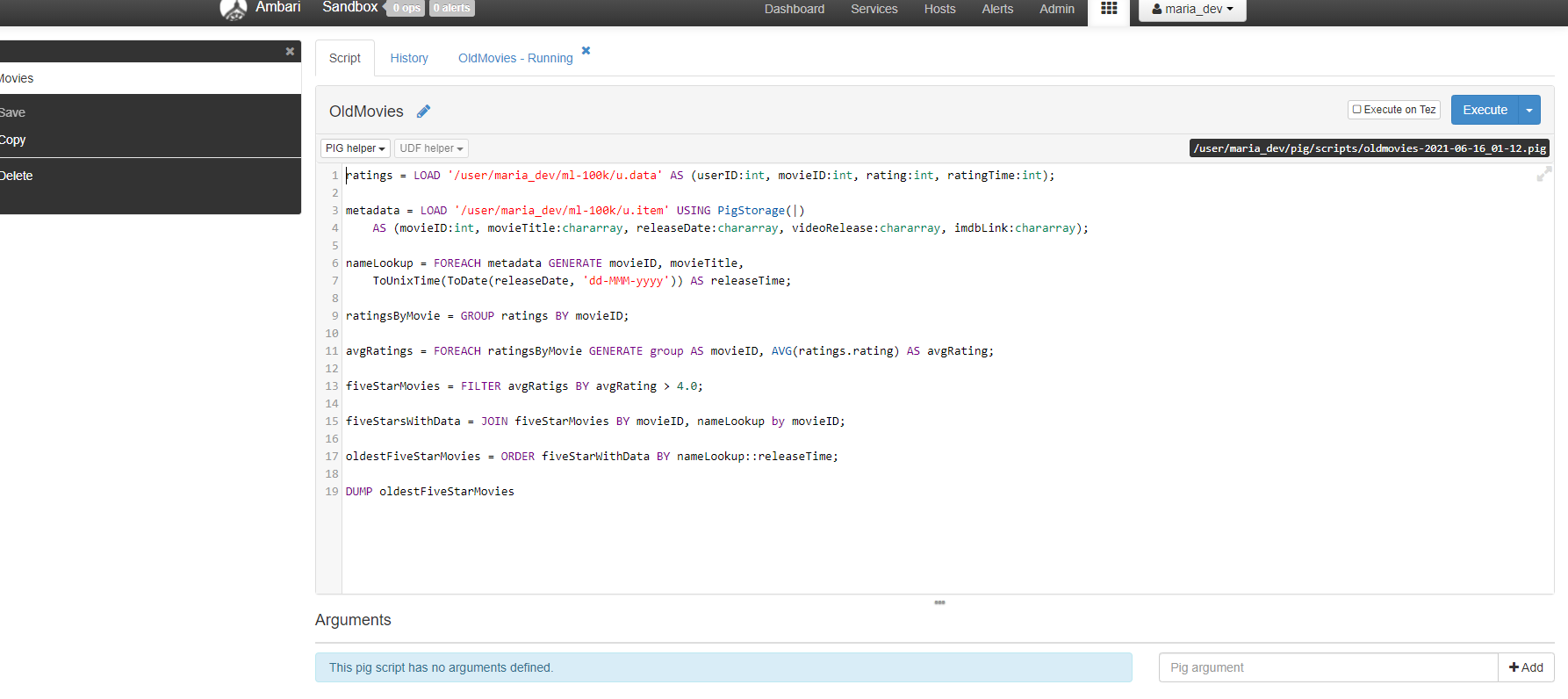
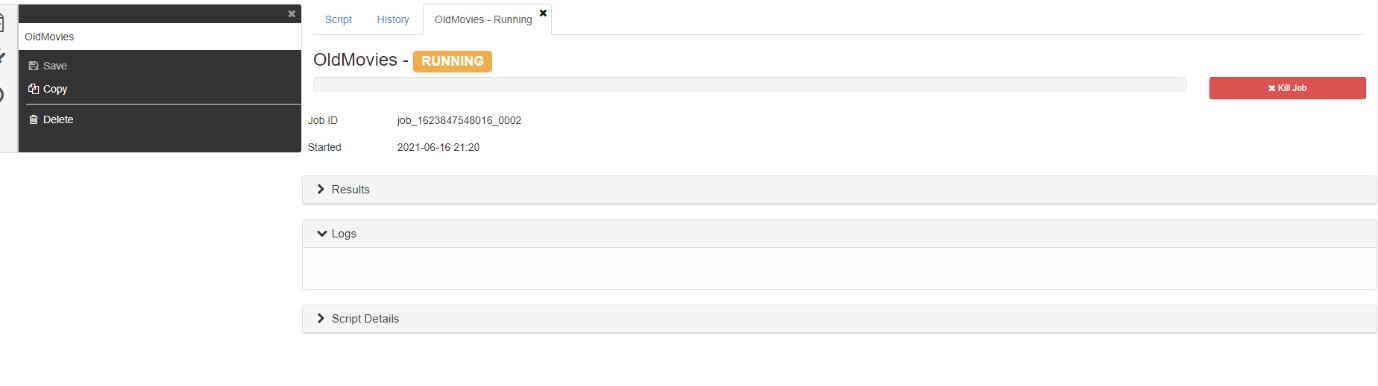
Join in the movie names to understand what the data means



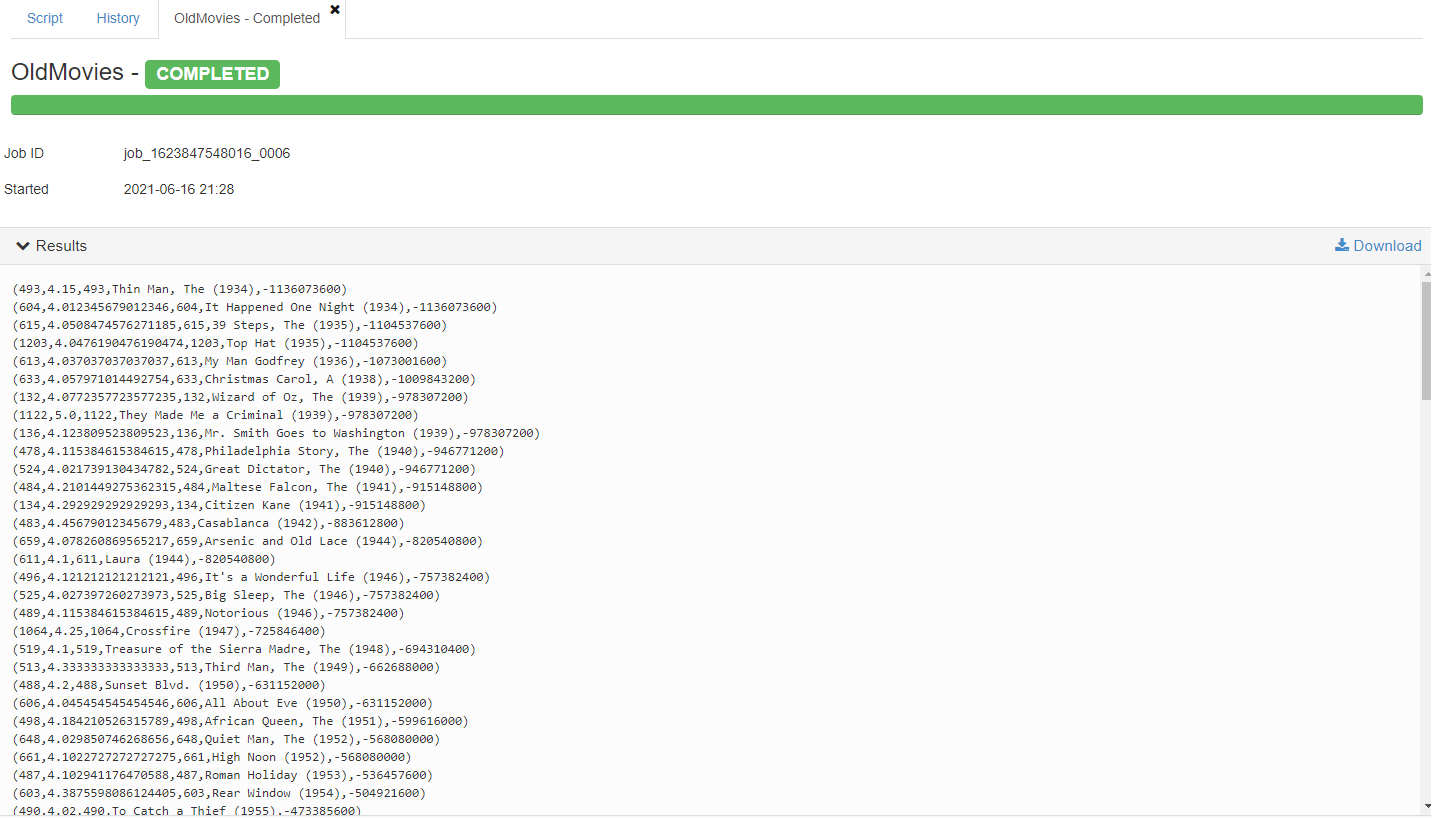
When you do a join you get the weird columns.

Order by the Release date.

Running the Pig Script

New tab will open up

Then it will finish up like this

To make the code run faster, you can use Tez – Directed acyclic graphs. Click on execute on Tez button and then execute again.

**More Pig Latin**: Dividng Deeping

* LOAD STORE DUMP
  + STORE ratings INTO “outRatings” USING PIGSTORAGE(‘:’):
* FILTER DISTINCT FOREACH/GENERATE MAPREDUCE STREAM SAMPLE
* JOIN COGROUP GROUP CROSS CUBE
* ORDER RANK LIMIT
* UNION SPLIT

Diagnostics

* DESCRIBE
* EXPLAIN – SQL EXPLAIN plan how
* ILLUSTRATE – what its doing

UDFs (Java defined Codes)

* REGISTER
* DEFINE
* IMPORT

Other fns and Loaders

* AVG ,CONCAT COUNT MAX MIN SIZE SUM
* PigStorage
* TextLoader
* JsonLoader
* AvroStorage - Serialization and Deserialisation
* ParquetLoader – Column oriented Data Format
* OrcStorage – Popular Compressed format
* HBaseStorage - interact with HBase

Programming Pig O’Reilly, DataFlow Scripting with Hadoop