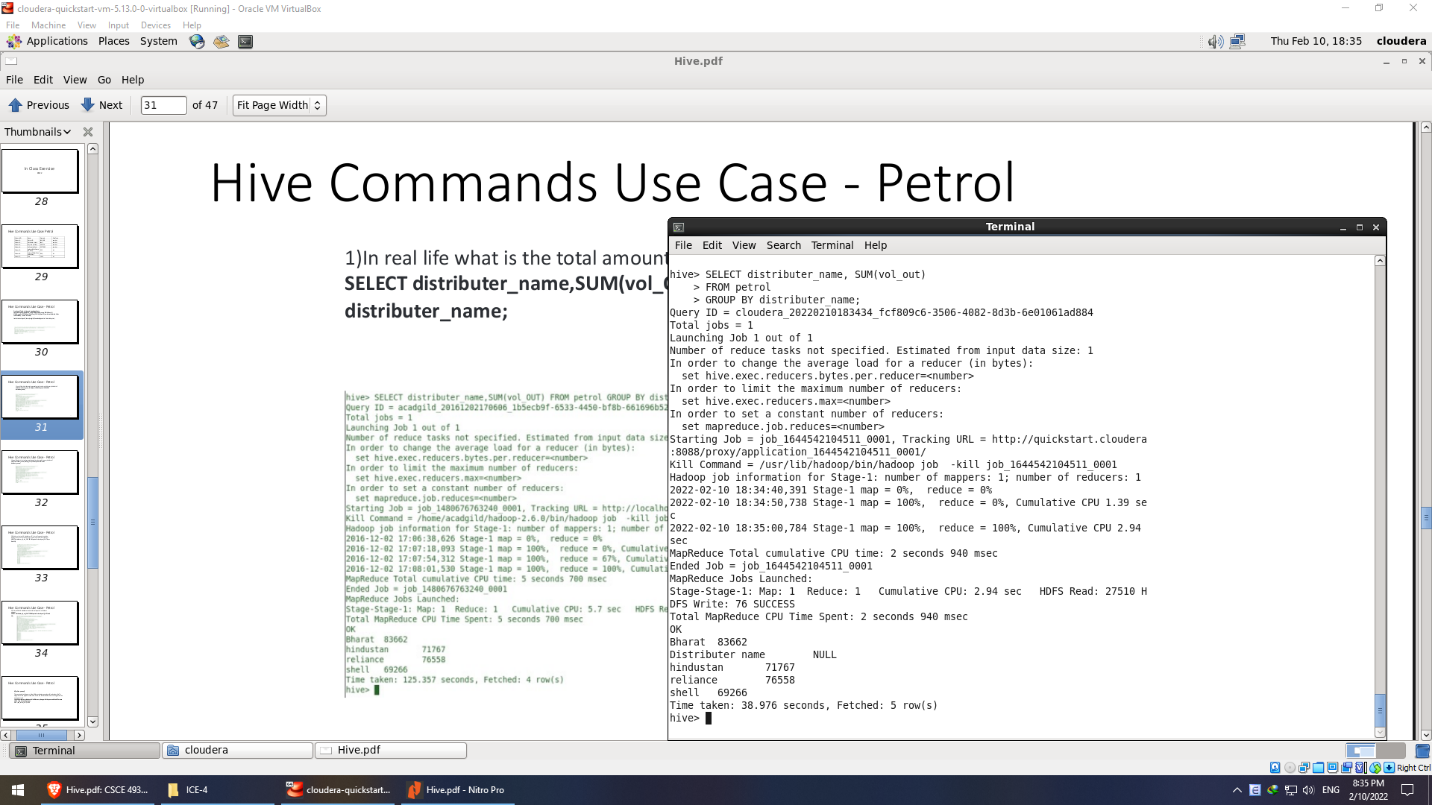


Create table petrol and load data from the text file to it.



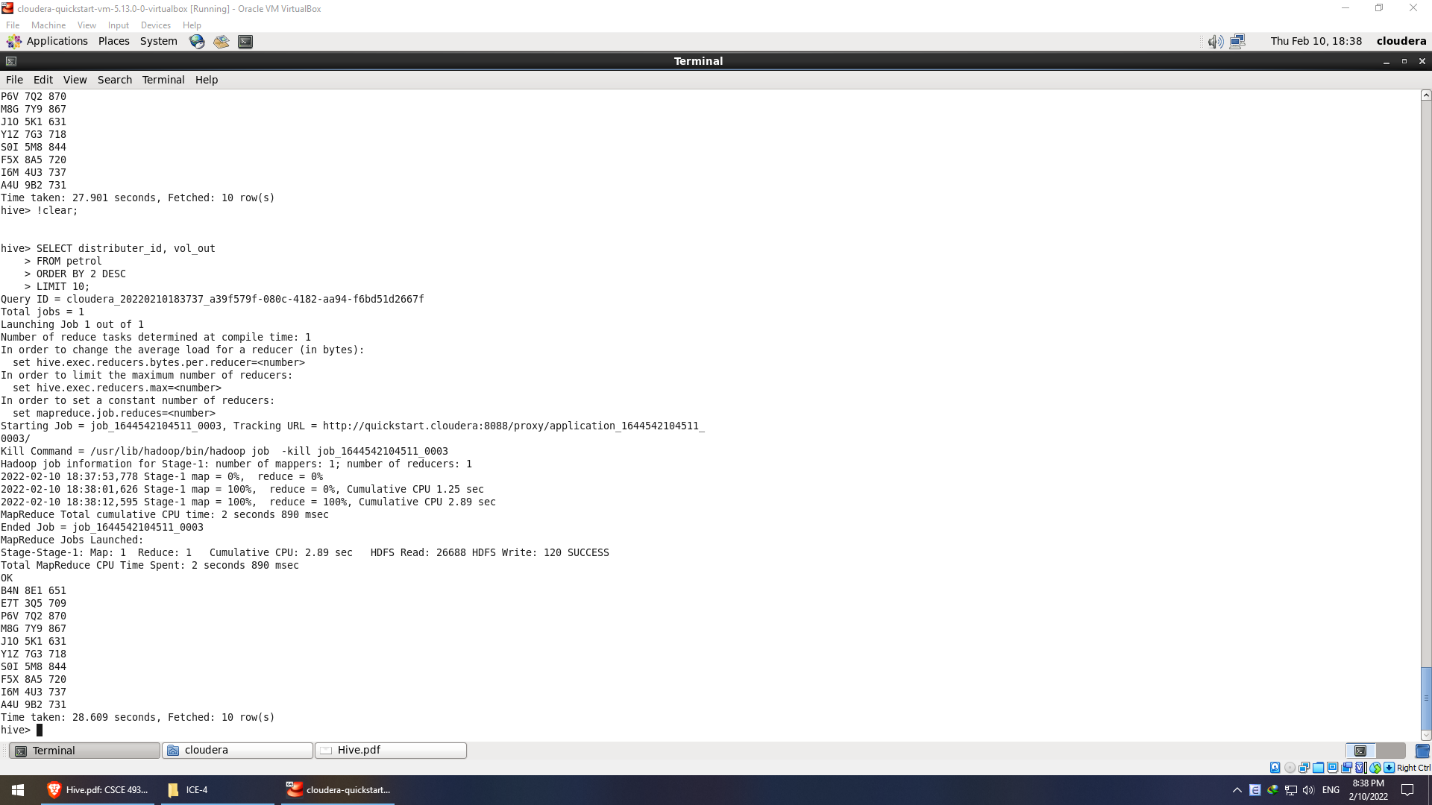
Find the total amount of petrol in volume sold by every distributer

* In SELECT

Total amount of petrol in volume = sum(vol\_out)

Every distributer = distributer name

* FROM petrol : extract from table named petrol
* GROUP BY distributer\_name: makes sure the same distributers are grouped together

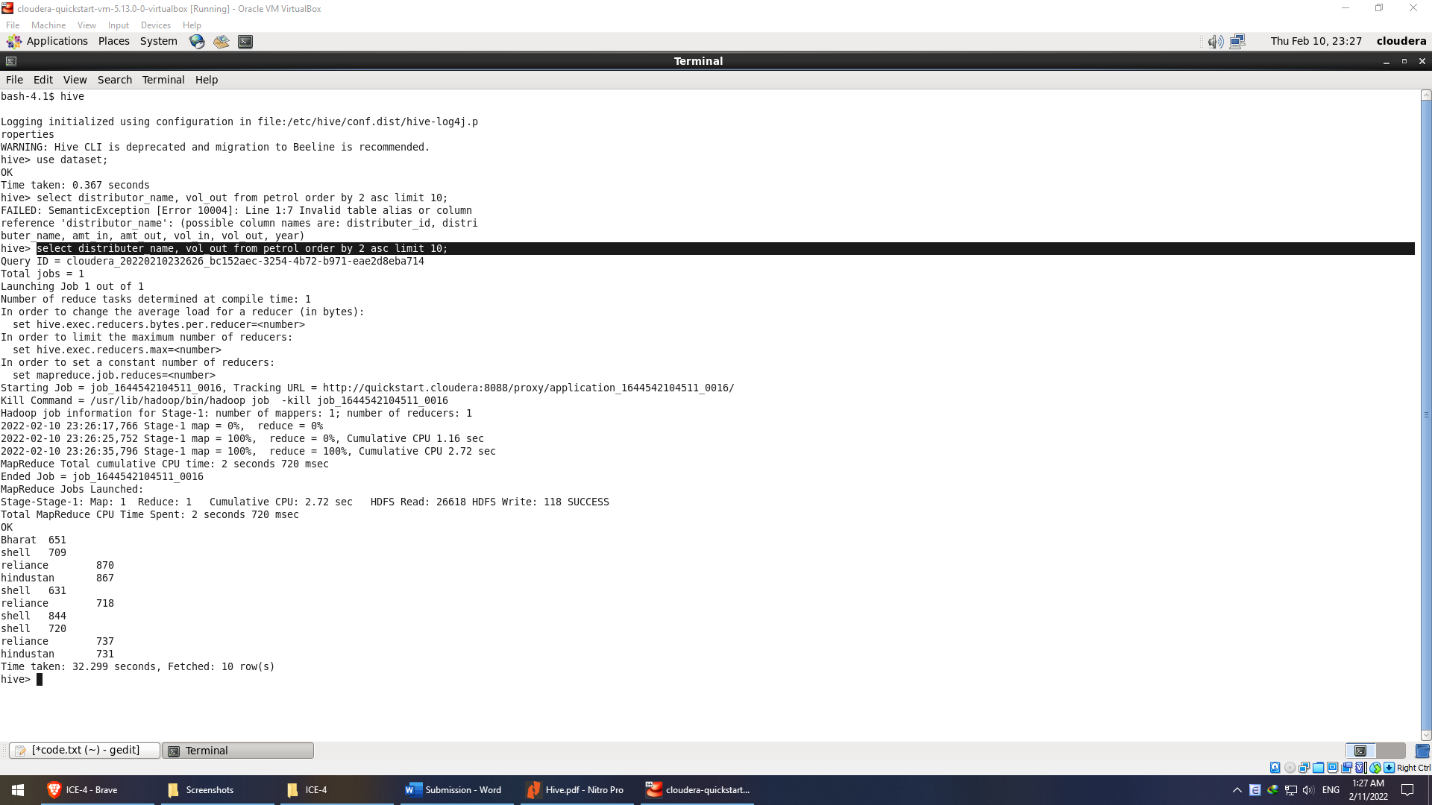


Top 10 distributors id for selling petrol and the amount of petrol sold in volume

* SELECT: extract the following features

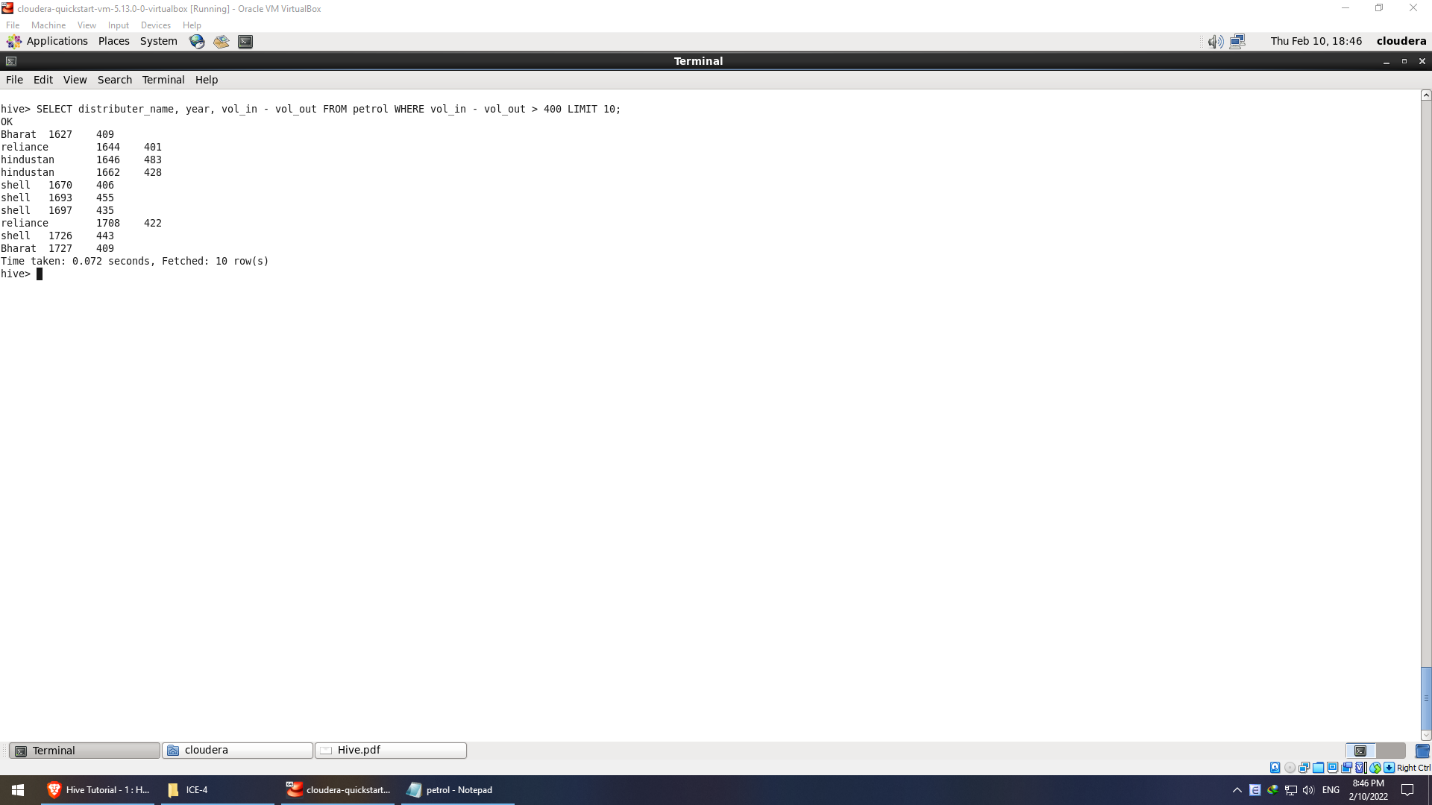
distributor\_id: distributor id, vol\_out: amount of petrol sold

* FROM petrol: name of the table
* ORDER BY 2 DESC: descending order by vol\_out (2 means the second position in SELECT)
* LIMIT 10: take top 10



10 bottom distributor names for selling petrol and the volume sold

* SELECT
  + distributor\_name: distributor name, vol\_out: the amount sold
* FROM petrol: name of the table
* ORDER BY 2 ASC: increasing order in terms of amount sold (from bottom)
* LIMIT 10: take 10 worst



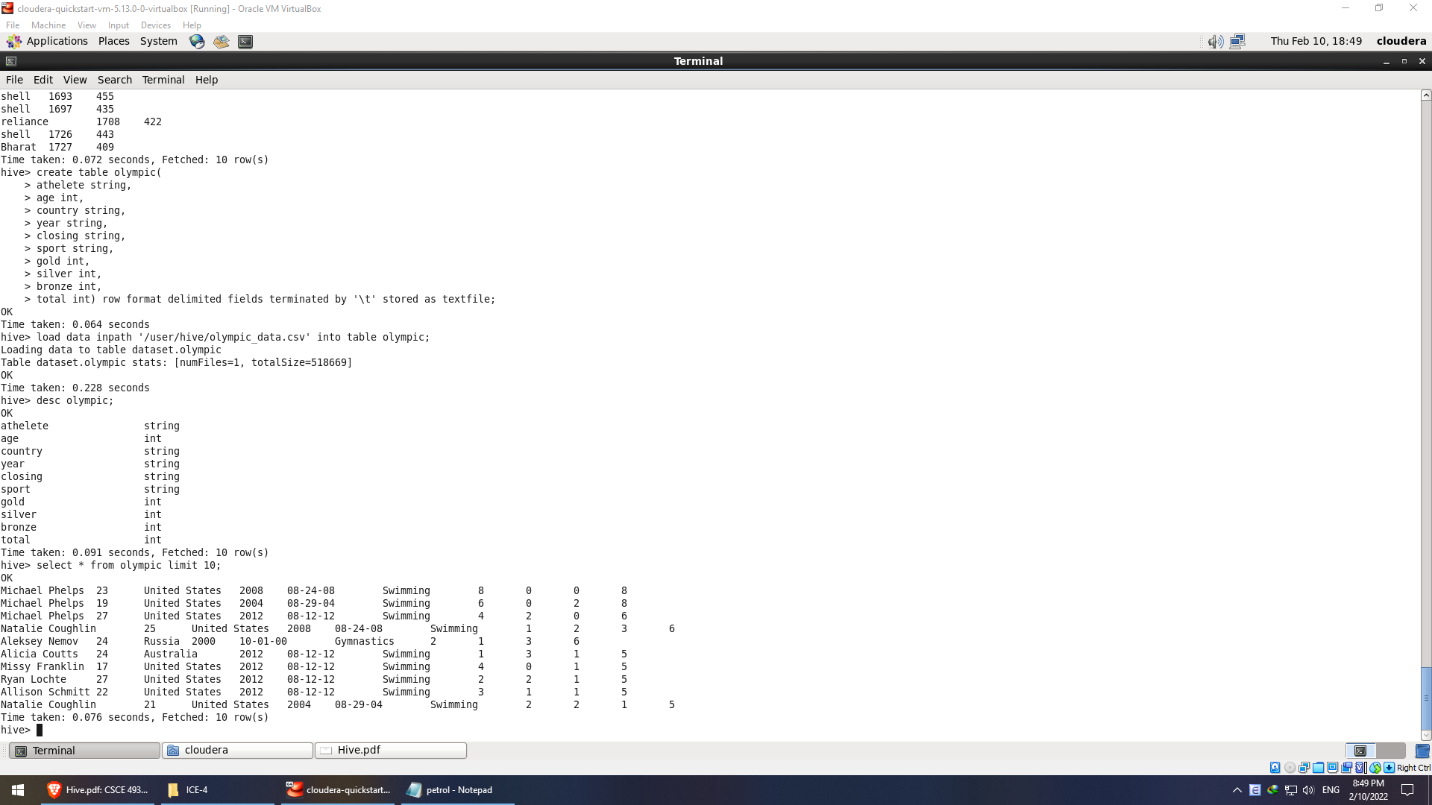
**Petrol 4:**

SELECT distributor\_name: name of distributor, year: year, vol\_in - vol\_out: the difference in volume

FROM petrol: name of table

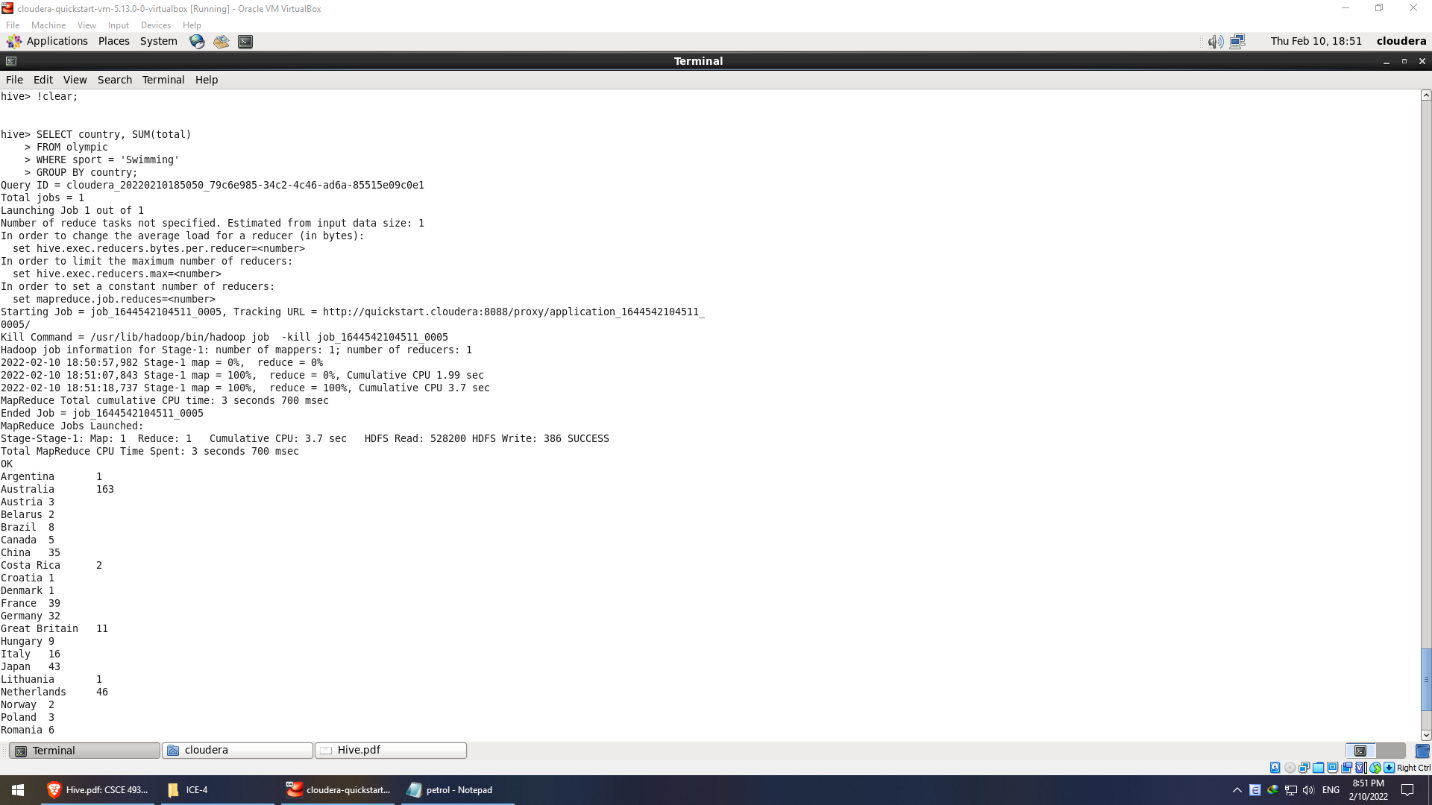
WHERE vol\_in – vol\_out > 400: condition that the difference has to be greater than 400

LIMIT 10: shows top 10 only



**Olympic:**

Create the table and load data from csv files



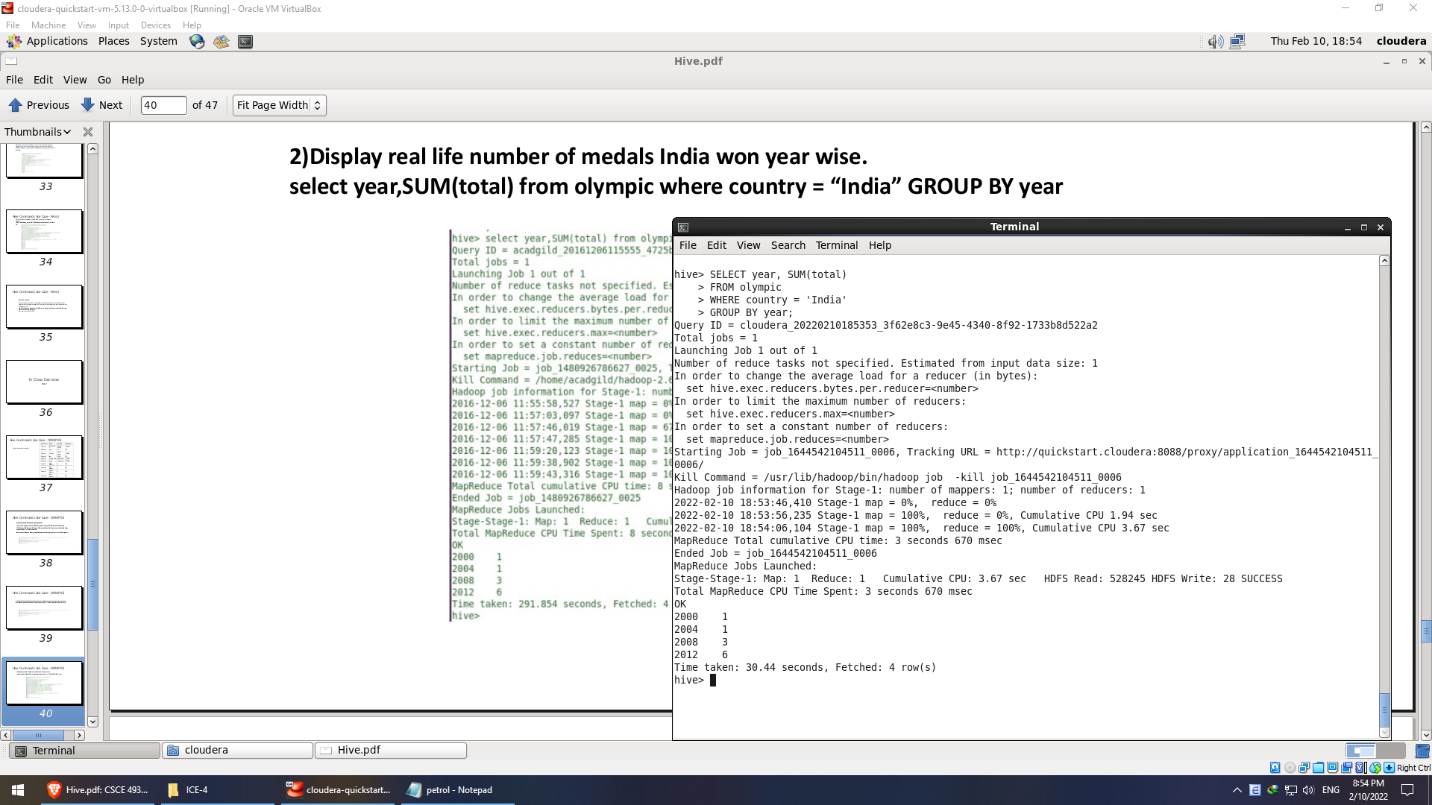
**Olympic 1:** List countries and their number of medals won in swimming

SELECT country: country name, sum(total): the total number of medals

FROM olympic: the name of the table

WHERE Sport = ‘Swimming’: won in swimming only

GROUP BY country: group countries by the same name



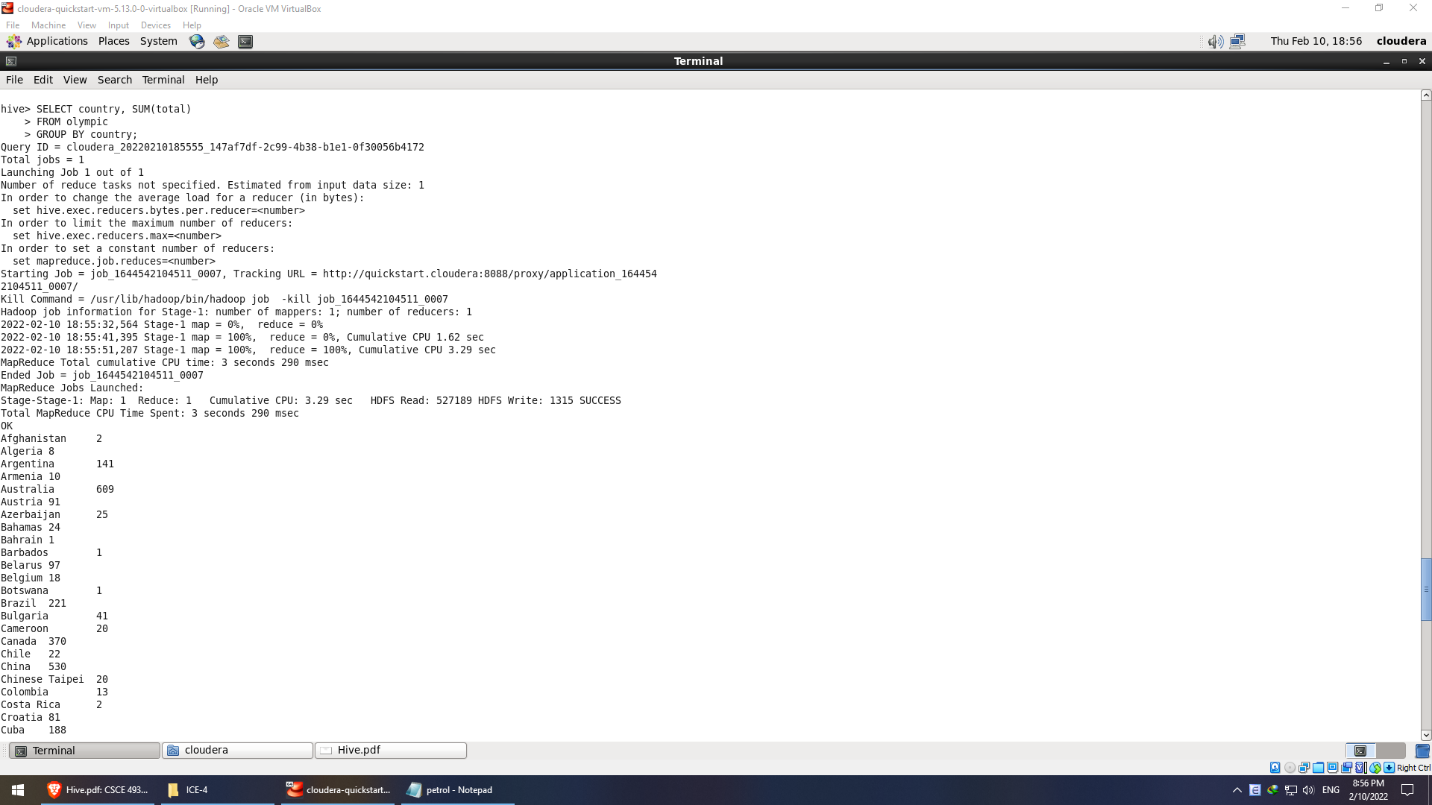
**Olympic 2:** Display the total number of medals that India won year wise

SELECT year: year the medal won, sum(total): the total of medals won

FROM Olympic: the name of the table

WHERE: country = ‘India’: only take India

GROUP BY year: group the same years



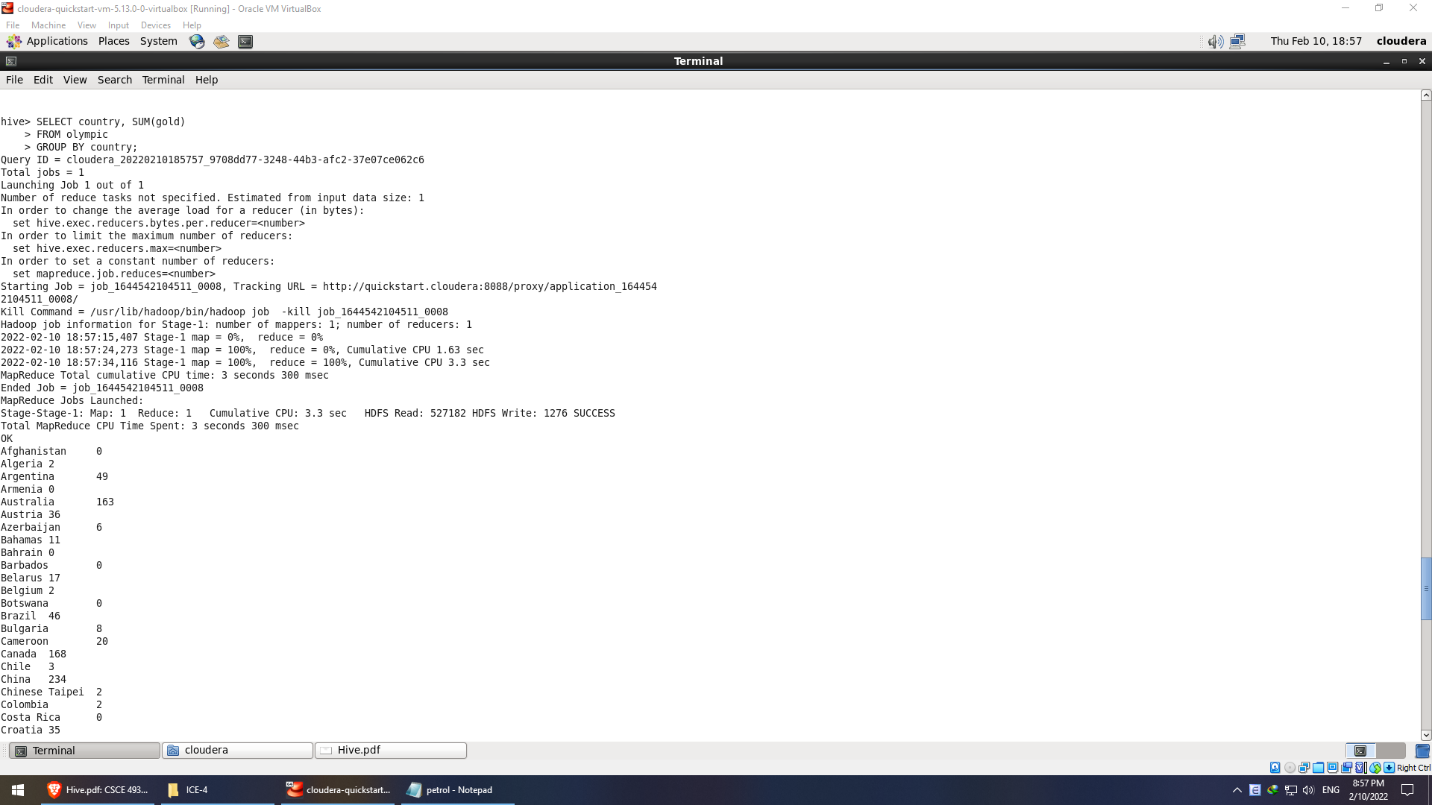
**Olympic 3:** Find the total number of medals each country won display the name along with total

medals.

SELECT country: country name, sum(total): the total of medals won

FROM Olympic: the name of the table

GROUP BY country: group the same countries

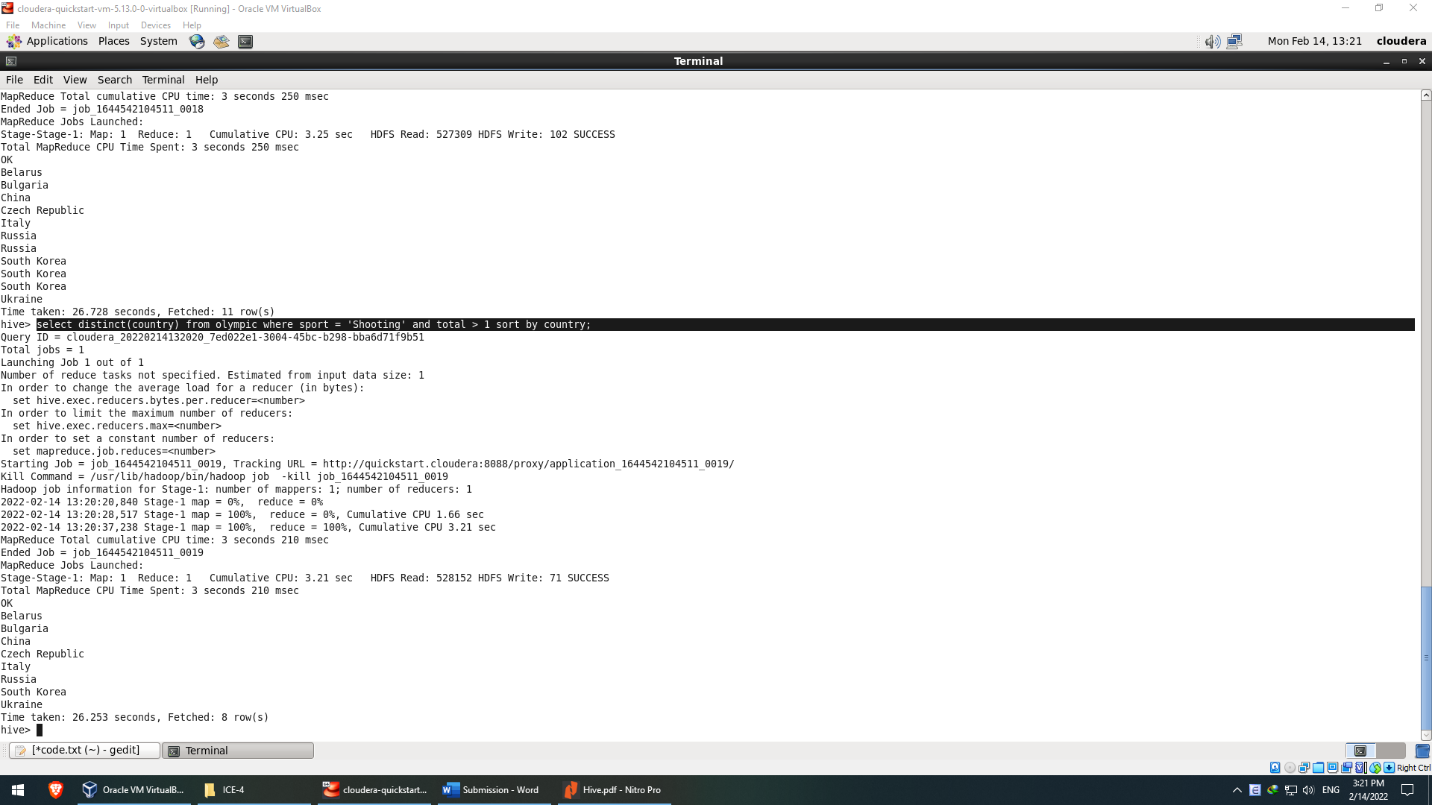


**Olympic 4**: Find the real life number of gold medals each country won.

SELECT country: country name, sum(gold): the total of gold medals won

FROM Olympic: the name of the table

GROUP BY country: group the same countries



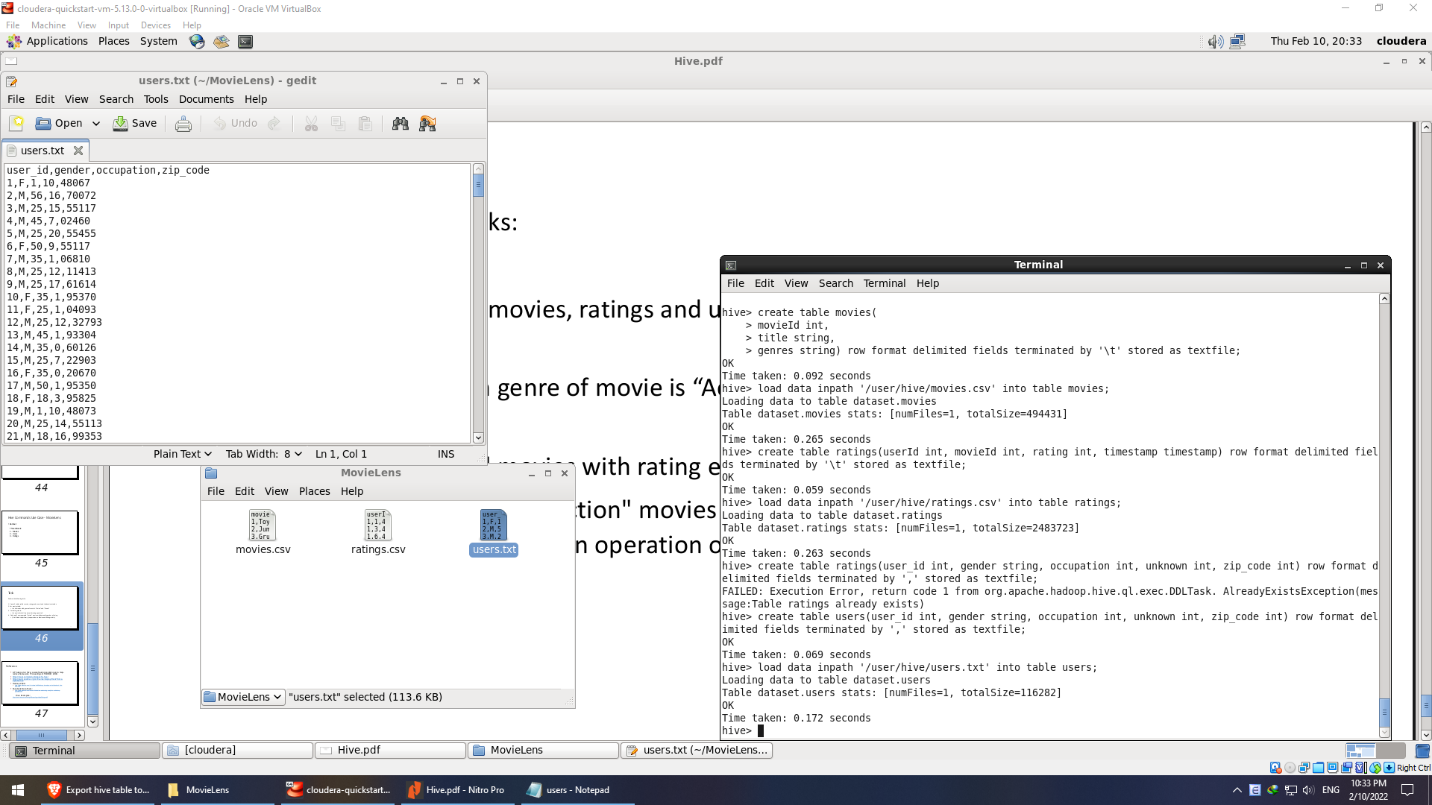
**Olympic 5:** Which country got medals for Shooting, year wise classification?

SELECT country: country name,

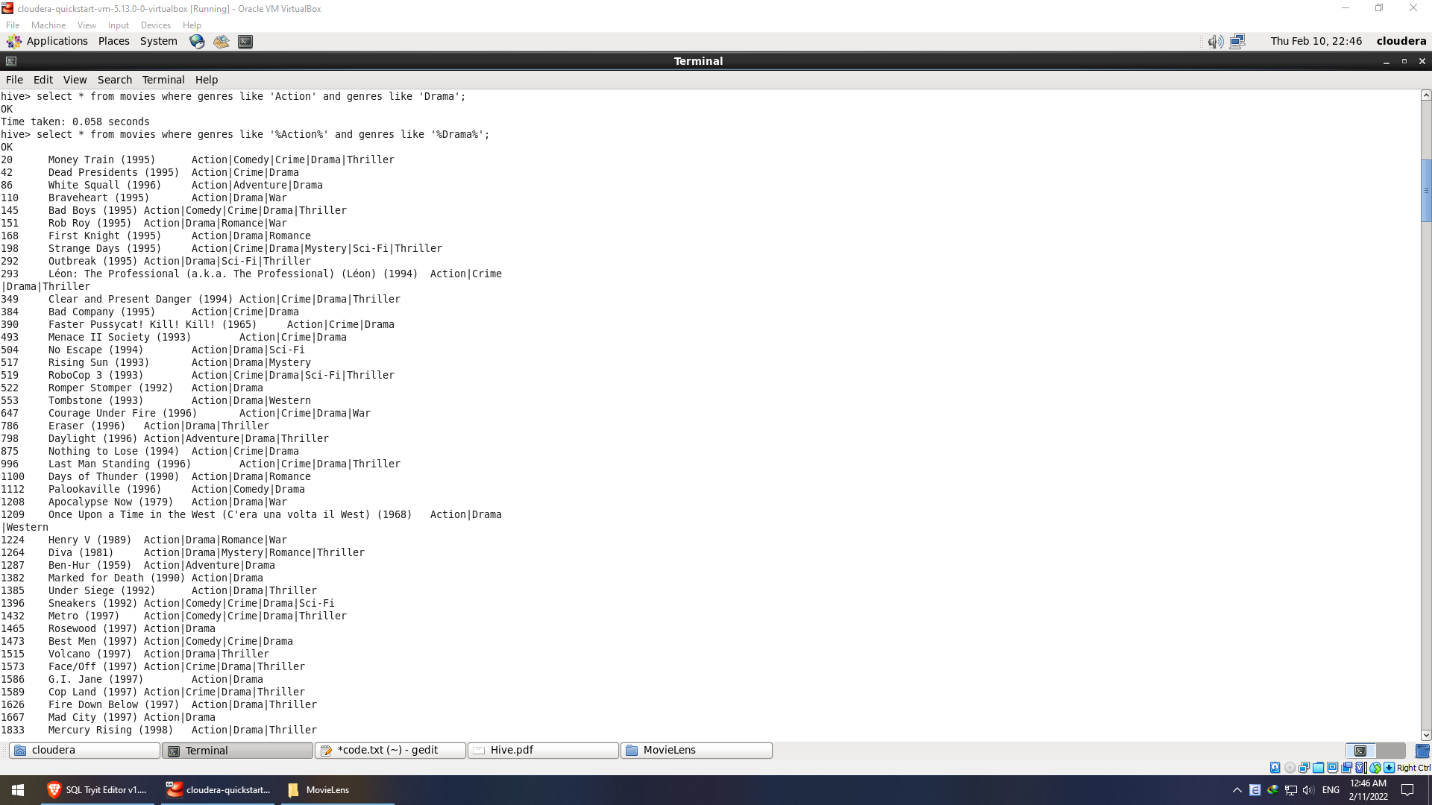
FROM Olympic: the name of the table

WHERE Sport = ‘Shooting’: Only extract Shooting as sport

GROUP BY did not work with year here, strangely.



**Part 3:** Create 3 tables and load local files to Hive

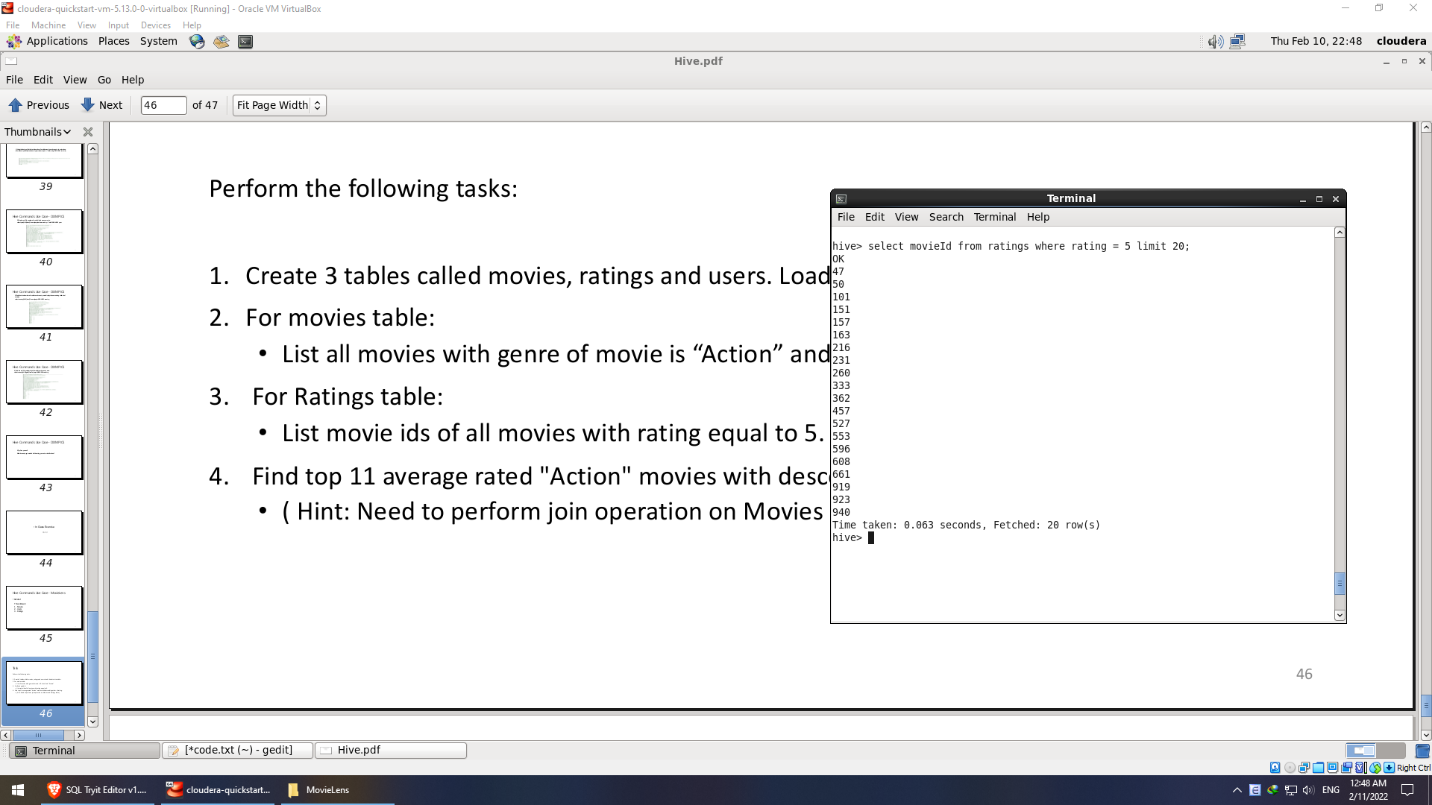


**List all movies with genre of movie is “Action” and “Drama”**

SELECT \*: all attributes

FROM movies: the table name

WHERE genres like “%Action%” and genres like “%Drama%”: As genres have multiple values separated by symbol “|”, LIKE makes sure as long as genres have the word, and %% means the particular searched values can be at any position.



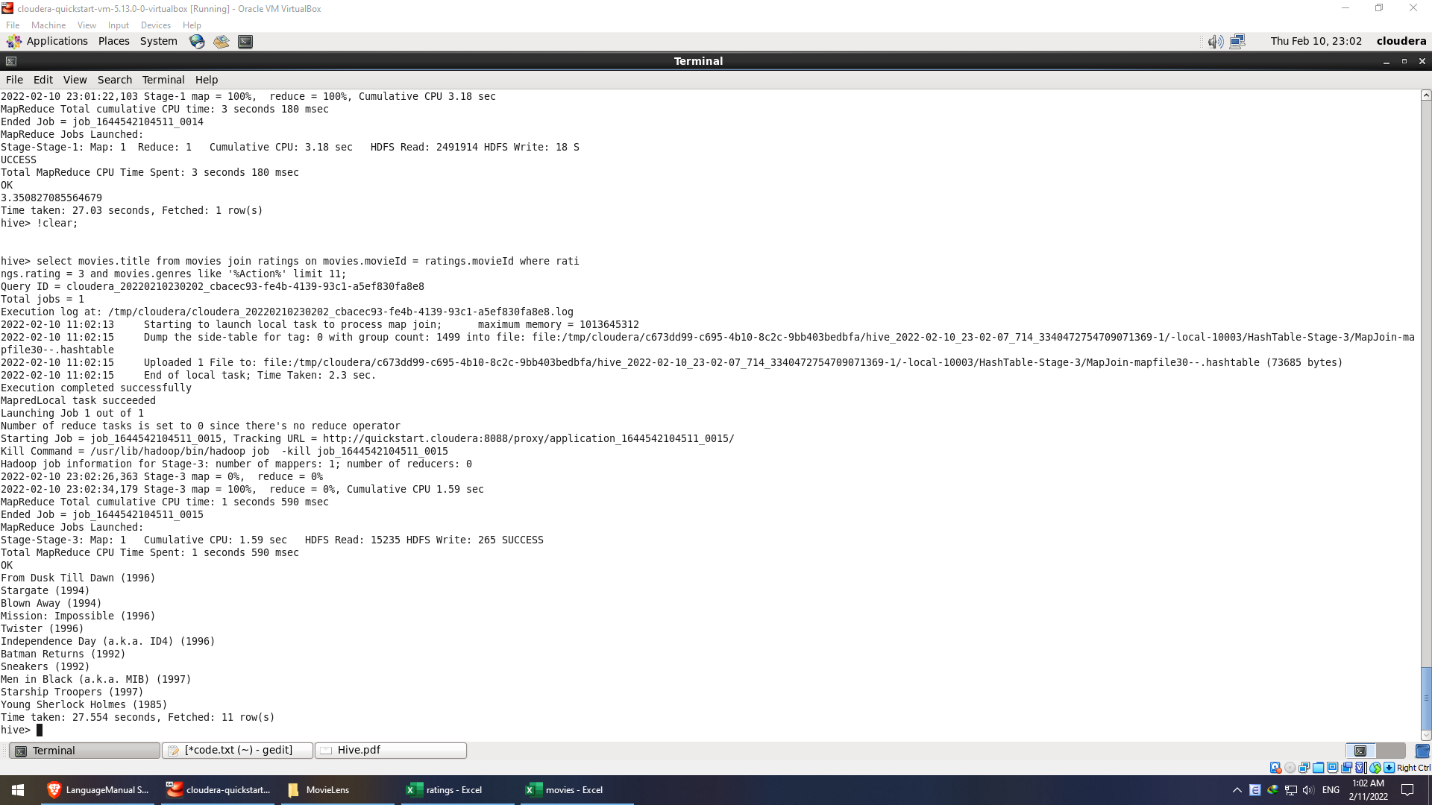
**List movie ids of all movies with rating equal to 5.**

SELECT movie\_id: movie id

FROM ratings: name of the table

WHERE rating = 5: only extract movies with rating = 5

LIMIT 20: I added it because the list was long.



**Find top 11 average rated "Action" movies with descending order of rating.**

SELECT movies.title: attribute title from table movies

FROM movies: name of the table

JOIN ratings: name of the joined table

ON movies.movieId = ratings.movieId: movieId in movies and movieId in ratings are the common part for the tables to join

WHERE

ratings.rating = 3: rating of the movie is average rated (from 1 to 5)

movies.genres like “%Action%”: genres can be any other but it has to have “Action” positioned any where in the line

LIMIT 11: find top 11

As you can see in the next screenshot, function order by did not work with ratings.rating

