

**BIG DATA TOOL AND TECHNIQUES**

Module CRN Code: 50194

Programme Code: MST/DS1/F1, MST/DS1/P2

Assessor(s): Judita Preiss

Student Name:- DAKSHINAMURTHY SAI NANDAN PUJYAM

Student Id:- @00655520

Contents

[1 INTRODUCTION 4](#_Toc103286985)

[2 Setup 5](#_Toc103286986)

[2.1 DATABRICKS 6](#_Toc103286987)

[2.2 AWS 6](#_Toc103286988)

[3 Data Cleaning & Preparation 5](#_Toc103286989)

[3.1 DATABRICKS 5](#_Toc103286990)

[3.2 PYSPARK RDD 6](#_Toc103286991)

[3.3 DATAFRAME 7](#_Toc103286992)

[3.4 HIVE QL 9](#_Toc103286993)

[3.5 AWS 10](#_Toc103286994)

[4 Problem Answers 11](#_Toc103286995)

[4.1 Question 1 11](#_Toc103286996)

[4.1.1 AIM 11](#_Toc103286997)

[4.1.2 Assumptions made 11](#_Toc103286998)

[4.1.3 PYSPARK RDD 11](#_Toc103286999)

[4.1.4 DATAFRAMES 11](#_Toc103287000)

[4.1.5 HIVEQL 11](#_Toc103287001)

[4.1.6 AWS ATHENA 11](#_Toc103287002)

[4.1.7 Discussion of result 12](#_Toc103287003)

[4.2 Question 2 12](#_Toc103287004)

[4.2.1 Aim 12](#_Toc103287005)

[4.2.2 Assumptions made 12](#_Toc103287006)

[4.2.3 PYSPARK RDD 12](#_Toc103287007)

[4.2.4 DATAFRAMES 13](#_Toc103287008)

[4.2.5 HIVEQL 13](#_Toc103287009)

[4.2.6 AWS ATHENA 14](#_Toc103287010)

[4.2.7 Discussion of result 14](#_Toc103287011)

[4.3 Question 3 14](#_Toc103287012)

[4.3.1 Aim 14](#_Toc103287013)

[4.3.2 Assumptions made 14](#_Toc103287014)

[4.3.3 PYSPARK RDD 15](#_Toc103287015)

[4.3.4 DATAFRAMES 15](#_Toc103287016)

[4.3.5 HIVEQL 15](#_Toc103287017)

[4.3.6 AWS ATHENA 16](#_Toc103287018)

[4.3.7 Discussion of result 16](#_Toc103287019)

[4.4 Question 4 17](#_Toc103287020)

[4.4.1 Aim 17](#_Toc103287021)

[4.4.2 Assumptions made 17](#_Toc103287022)

[4.4.3 PYSPARK RDD 17](#_Toc103287023)

[4.4.4 DATAFRAMES 17](#_Toc103287024)

[4.4.5 HIVEQL 18](#_Toc103287025)

[4.4.6 AWS ATHENA 19](#_Toc103287026)

[4.4.7 Discussion of result 19](#_Toc103287027)

[4.5 Question 5 20](#_Toc103287028)

[4.5.1 Aim 20](#_Toc103287029)

[4.5.2 Assumptions made 20](#_Toc103287030)

[4.5.3 PYSPARK RDD 20](#_Toc103287031)

[4.5.4 DATAFRAMES 20](#_Toc103287032)

[4.5.5 HIVEQL 21](#_Toc103287033)

[4.5.6 AWS ATHENA 22](#_Toc103287034)

[4.5.7 Discussion of result 22](#_Toc103287035)

[4.6 Question 6 23](#_Toc103287036)

[4.6.1 Aim 23](#_Toc103287037)

[4.6.2 Assumptions made 23](#_Toc103287038)

[4.6.3 PYSPARK RDD 23](#_Toc103287039)

[4.6.4 DATAFRAMES 23](#_Toc103287040)

[4.6.5 HIVEQL 24](#_Toc103287041)

[4.6.6 AWS ATHENA 24](#_Toc103287042)

[4.6.7 Discussion of result 25](#_Toc103287043)

[4.7 Further analysis 1 25](#_Toc103287044)

[4.7.1 Aim 25](#_Toc103287045)

[4.7.2 Assumptions made 26](#_Toc103287046)

[4.7.3 DATAFRAMES 26](#_Toc103287047)

[4.7.4 Discussion of result 26](#_Toc103287048)

[4.8 Further analysis 2 27](#_Toc103287049)

[4.8.1 Aim 27](#_Toc103287050)

[4.8.2 Assumptions made 27](#_Toc103287051)

[4.8.3 DATAFRAMES 27](#_Toc103287052)

[4.8.4 Discussion of result 27](#_Toc103287053)

# INTRODUCTION

The 21st century has revolutionized the way we interact with the world, with almost 2.5 quintillion bytes of data being created every single day worldwide. Internet connectivity will increase the amount of data created as more people access it. Data is measured in 3V’s, variety, volume, and velocity, and if we are to process the data, we need some tools. By using big data tools, we can process large amounts of data and generate meaningful insights. With so much data being collected, it makes sense for companies to use this data to better understand their customers and their behaviour. Data Science has become increasingly popular in recent years because it provides useful insights that can help businesses improve their operations.

Although companies have been collecting a huge amount of data for decades, the concept of big data became popular only in the Early Mid 2000S. The companies recognized the amount of data collected on a daily basis and the importance of the effective use of these data.

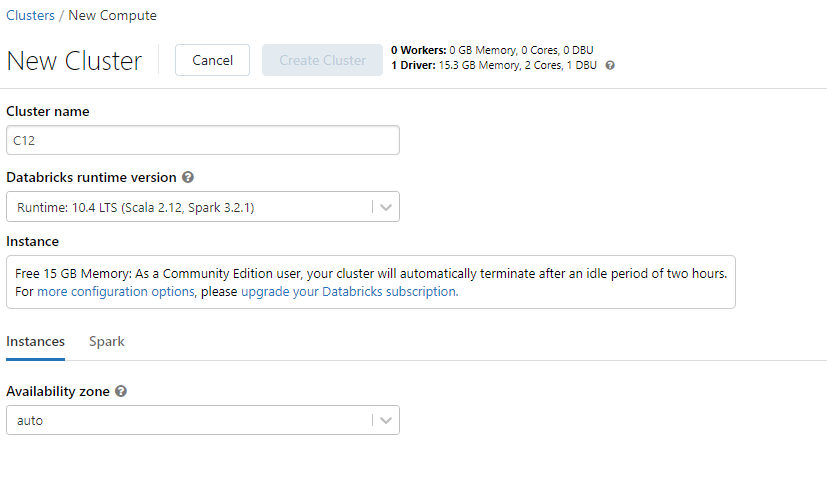
Companies like Google, Facebook, and Microsoft collect enormous amounts of data and use it for analysis, this data consists of structured and unstructured data. Big data is varied and has numerous tools to analyse and visualize it.

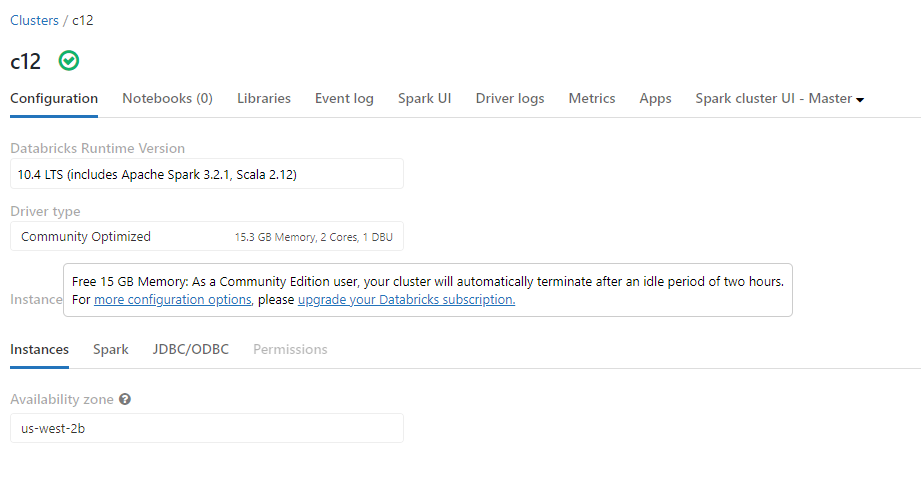
For our implementations we use SPARK, HiveQL, AWS Athena. We perform spark and hive implementations using data bricks.

# Setup

## DATABRICKS

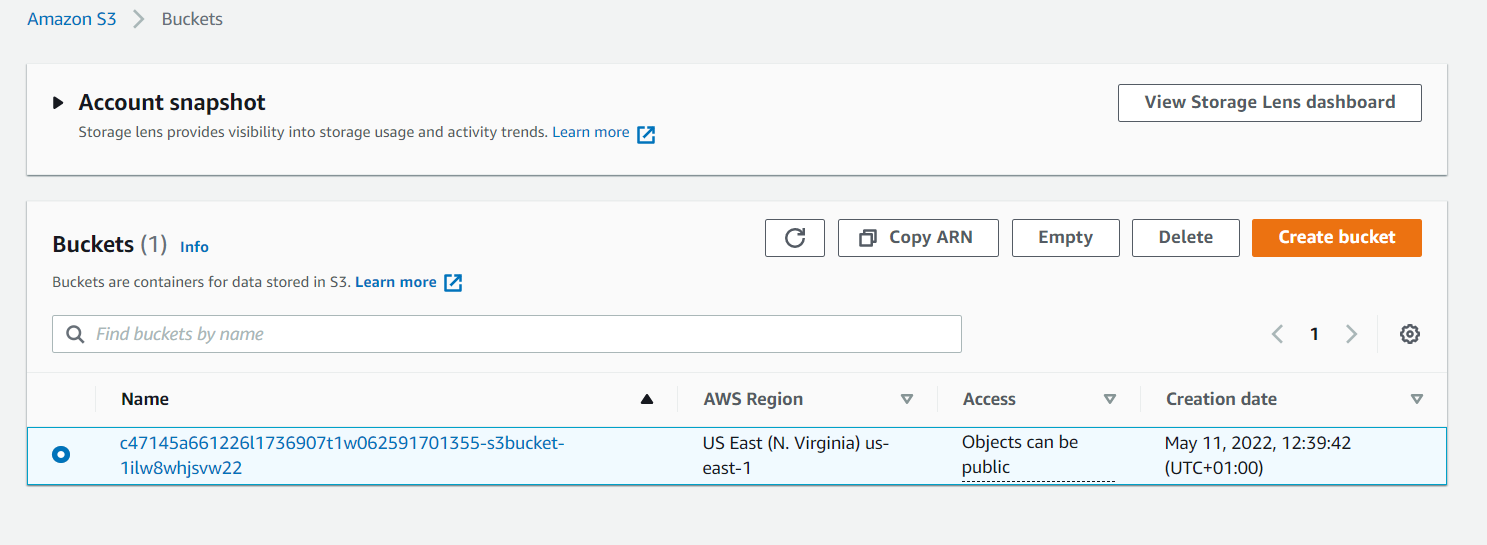
To perform tasks in Databricks initially we need to create a cluster by giving a name to cluster, Once the cluster is created, we get 15gb free memory to use which is more than enough for our implementations. we can use this cluster to perform the tasks.

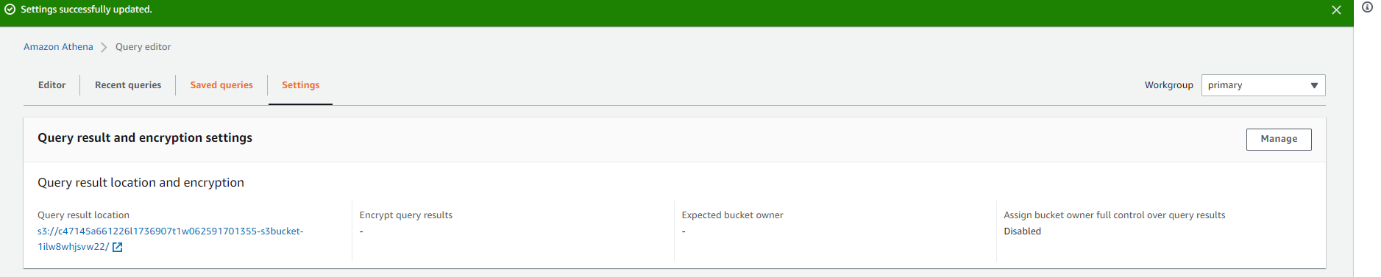
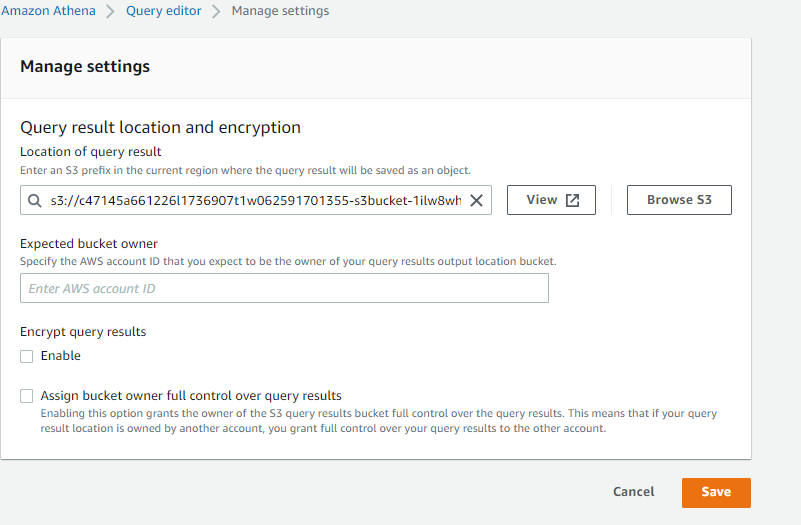




## AWS

In AWS initially we need to create a new bucket in s3. After the bucket is created, we need to open Athena and query editor manage settings we need to pass our s3 bucket path and click browse s3 and select s3 bucket from the location and click on connect, after the connection is made, we are good to perform tasks.



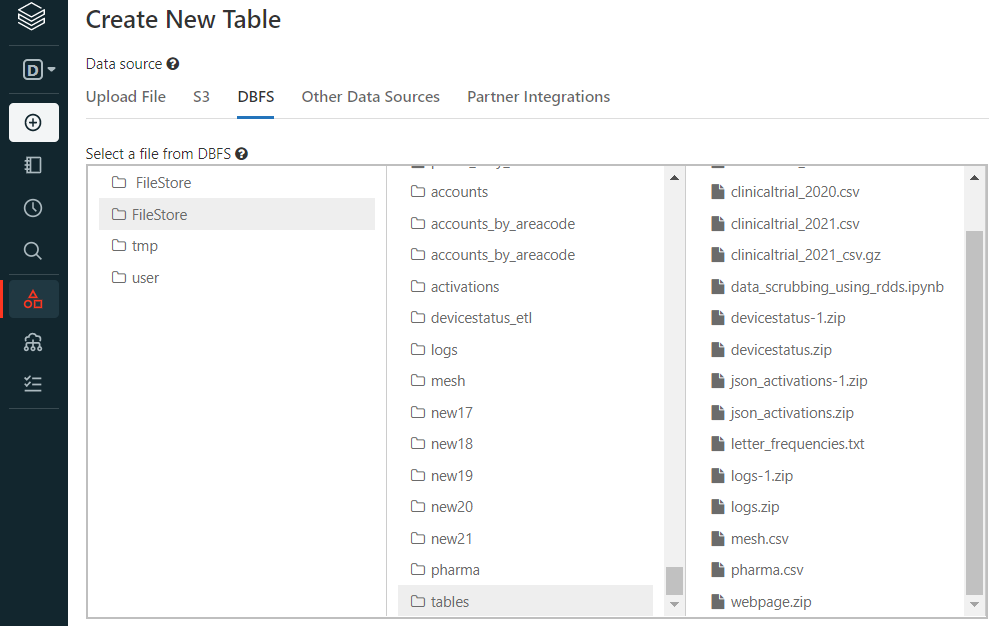


# Data Cleaning & Preparation

## DATABRICKS

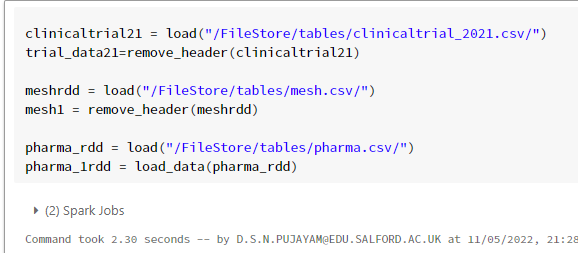
After the cluster is created, we need to upload the required data into Databricks Filestore (DBFS), For our analysis we are uploading clinicaltrial\_2021.csv.gz file to DBFS and unzip the file.





## **PYSPARK RDD**

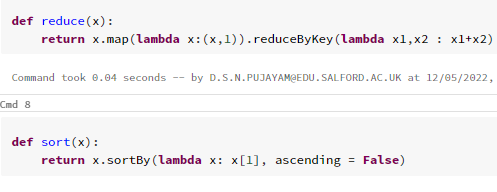
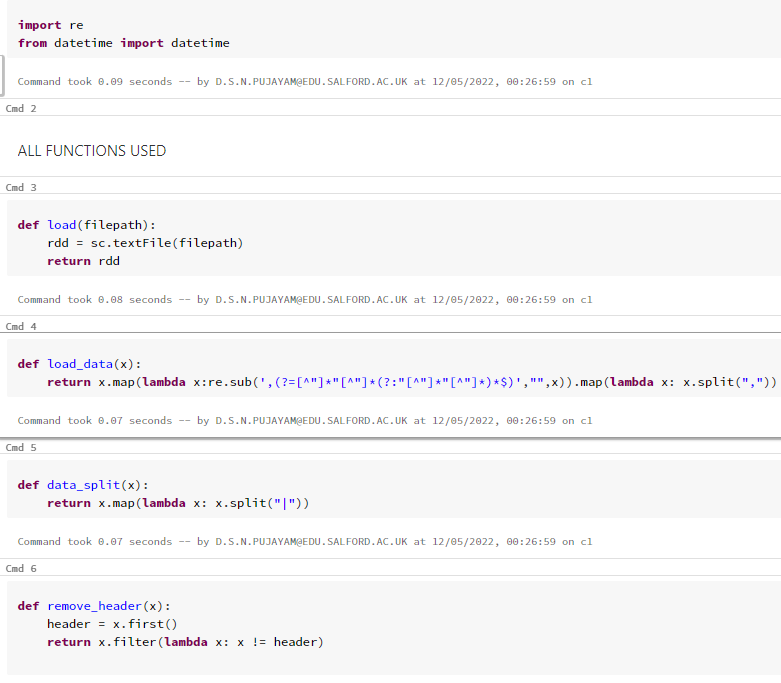
The data is being loaded to rdd using Sc text file method inside a load function and we are removing all the headers using remove header function.





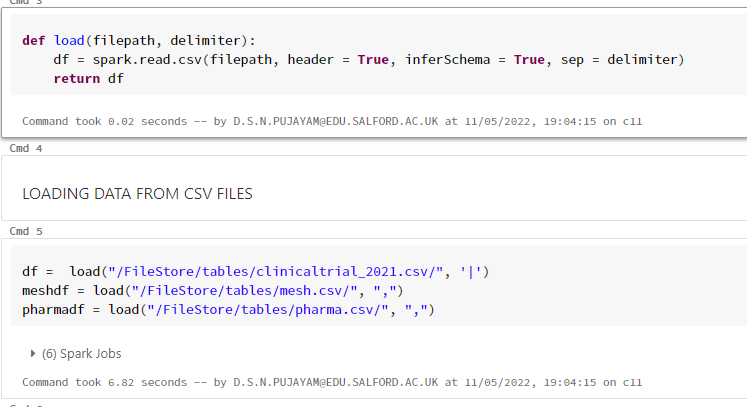
Here are the user defined functions which we will be using for our analysis

* Load () – used to load data to rdd by giving file path as parameter.
* Load\_data () – used to replace commas in between values and split the comma separated data.
* Data\_split () – used to split data by passing delimiter as parameter.
* Remove\_header () – used to remove first line from the rdd which is column names.
* Reduce () – used to get the frequencies for given column.
* Sort () – used to sort the elements in descending order.



## DATAFRAME

For data frame we are loading the data dynamically by a user-defined function that takes the file path as a parameter. The header looks for the first row of the data as the header, assumes the schema from the file, and uses the delimiter.



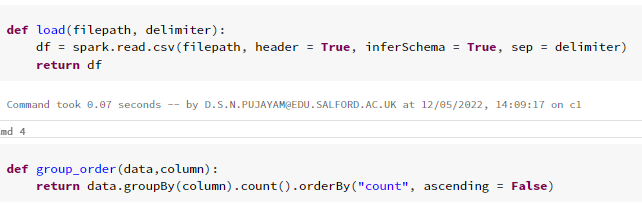
Here are system functions and user defined functions we will be using in our implementations.

We are importing split, explode, col, substring, to date functions from pyspark sql functions to perform various functionalities on the data set and generate meaningful results.

We imported pandas and matplotlib for plotting.

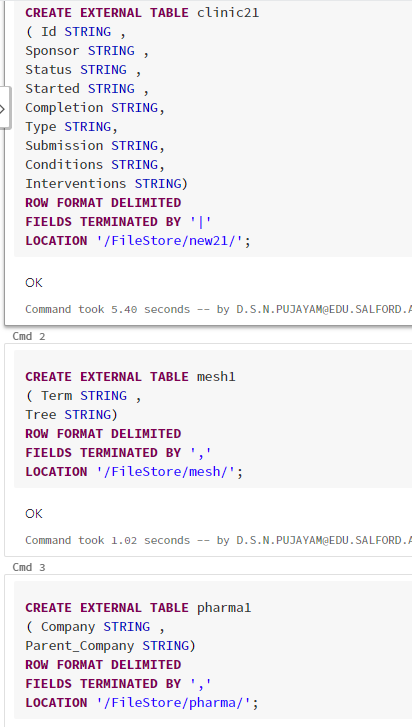


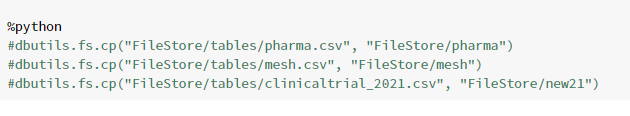
* load () – used to load data into data frame by passing file path and delimiter as parameters.
* group\_order () – used to group data and count values and order by the counts of the values in descending order.



## HIVE QL

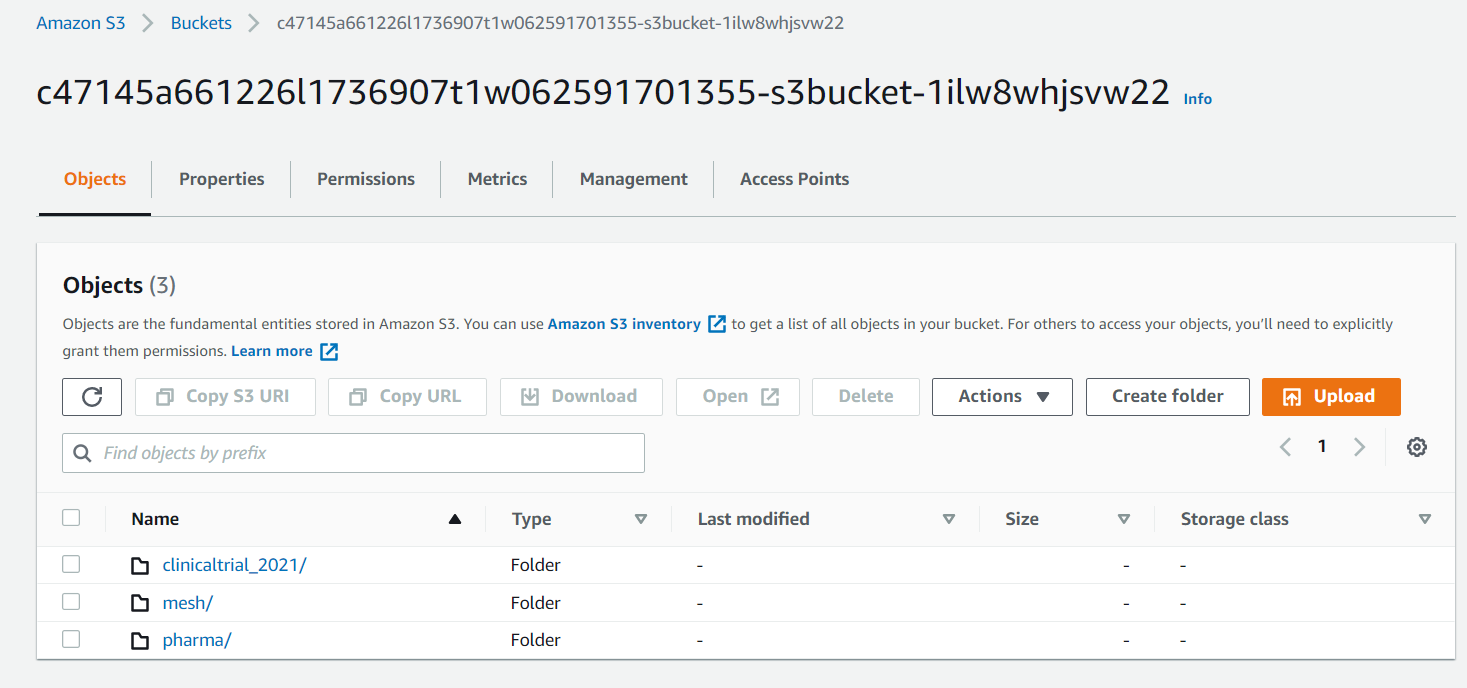
For hive we create tables using create external table and table name and providing column names and specify column type and also need declare the delimiter of the data and provide the location of the file to load.



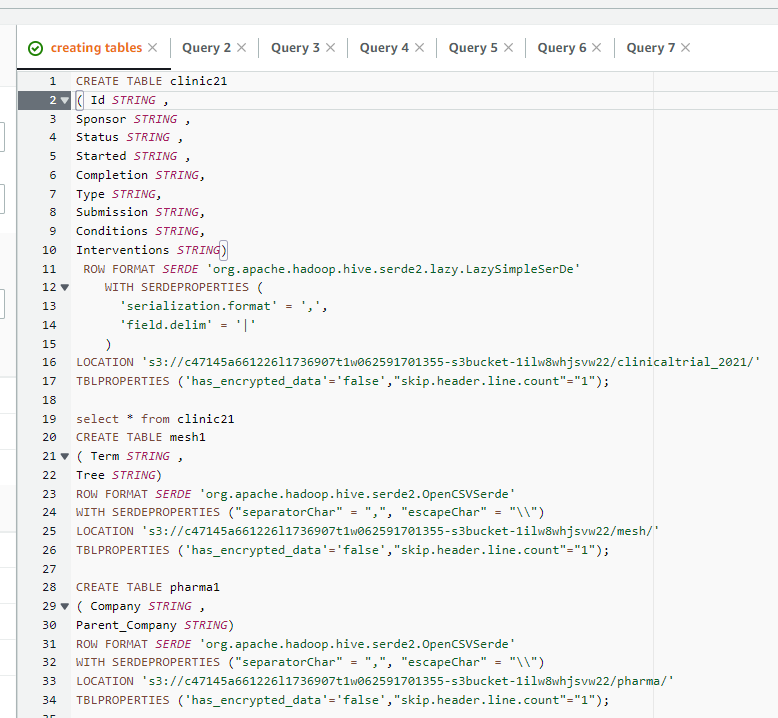


## AWS

Here we upload data into s3 after data is uploaded, we can write queries in Athena using the data.



We will now create tables in athena using the data we uploaded in s3, for.



As we have all the data prepared, Now we can solve the problem statements and perform further analysis to generate meaningful insights.

# Problem Answers

## Question 1

### AIM

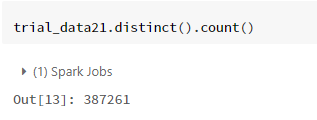
The number of studies in the dataset. You must ensure that you explicitly check distinct studies.

### Assumptions made

The data is available in csv file format, here we need to look into the data and separate the data by specifying the delimiter it contains.

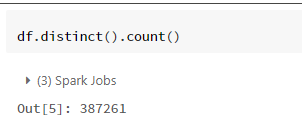
### PYSPARK RDD

Forthis implementation we take the clinical trial Rdd and execute distinct which takes unique elements and count function which is used to count the elements in the rdd.



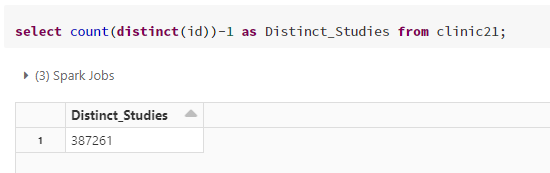
### DATAFRAMES

We are looking for number of distinct studies using data frame and performing distinct and count functions which takes unique elements and count the elements.



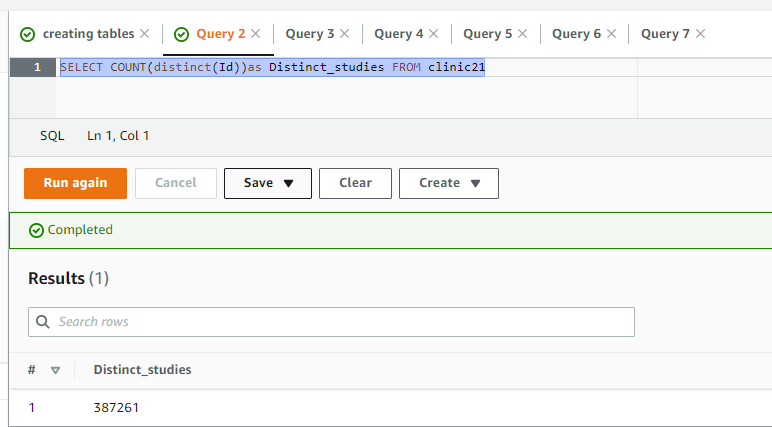
### HIVEQL

Here we are taking clinical trial data from hive table we already created by using select query we are performing distinct and count functions on unique id column to get number of distinct studies conducted.



### AWS ATHENA

We are examining clinical trial data table that we have already created using a select query we are performing the distinct and count functions on the unique ID.



### Discussion of result

****

In all the above implementations we got similar results for number of distinct studies that are conducted.

## Question 2

### Aim

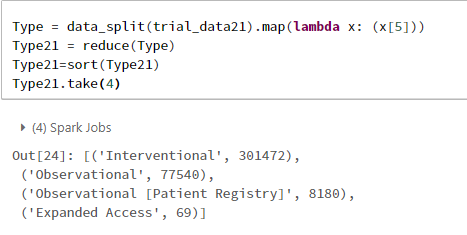
To list all the types contained in the Type column in the dataset along with the frequencies of each type and ordered from most frequent to least frequent.

### Assumptions made

For this analysis we need to process the type column from clinical trials data and check if there are any inconsistencies in the data.

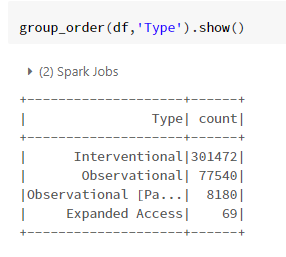
### PYSPARK RDD

Here we use user-defined functions and built-in functions to surround data and to obtain frequencies for each type.

****

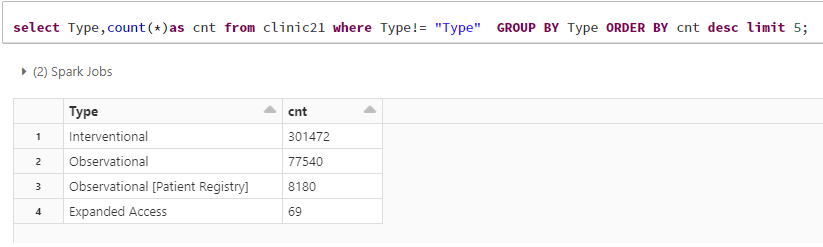
### DATAFRAMES

Here is the implementation to calculate frequencies of all study types in spark dataframe with user-defined function.

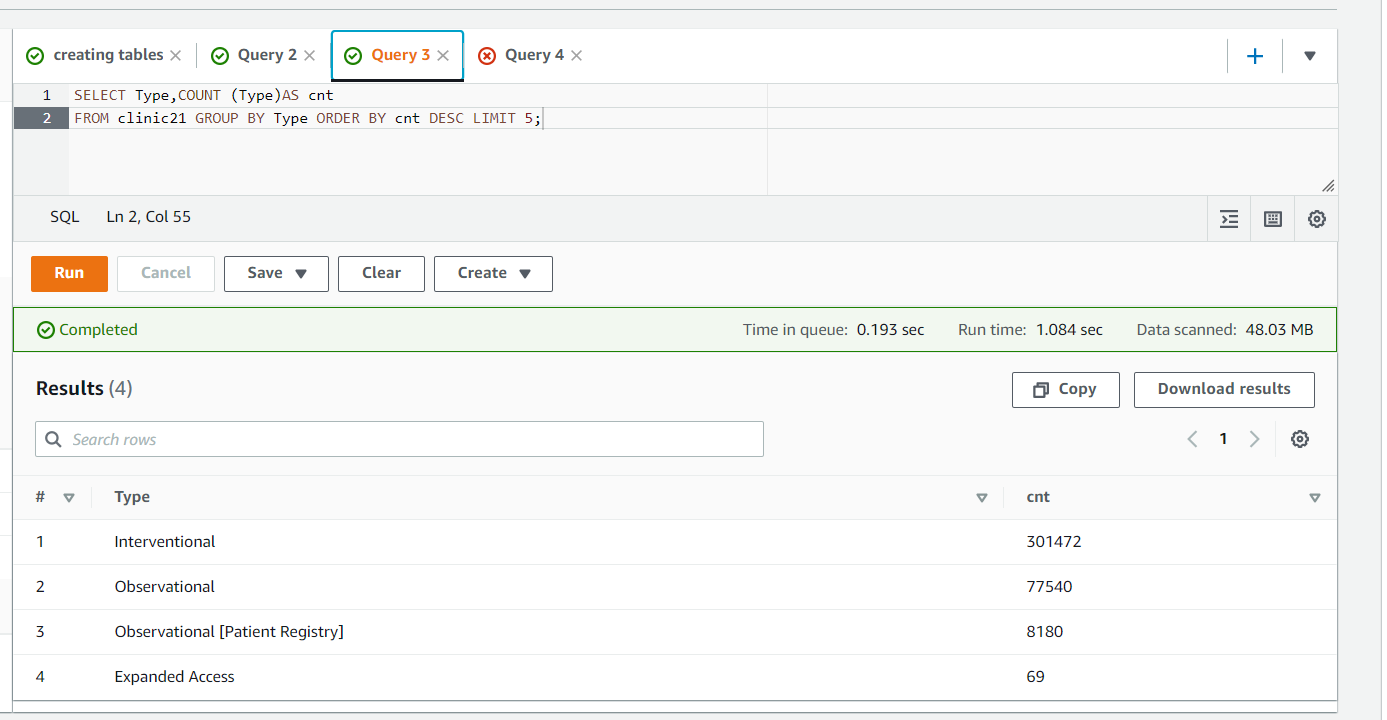
****

### HIVEQL

Using select query we select type column from table count the frequencies.

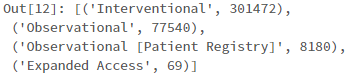
****

### AWS ATHENA

We use the select query to select the type column and count the number of occurrences of that type. We then group the data by using the groupby function, and order the data by using the orderby function. ****

### Discussion of result

After analysing type column from clinical trials 2021 dataset, we can identify that interventional type has the highest number of frequencies with count 301472 and next is observational type with 77540 frequencies.



## Question 3

### Aim

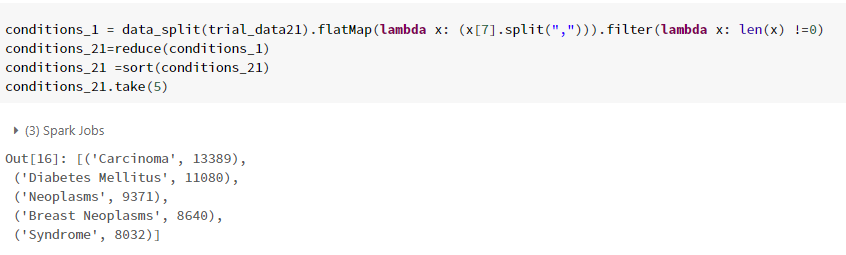
To find top 5 conditions from Conditions column with their frequencies.

### Assumptions made

The conditions column requires further processing and need to split it.

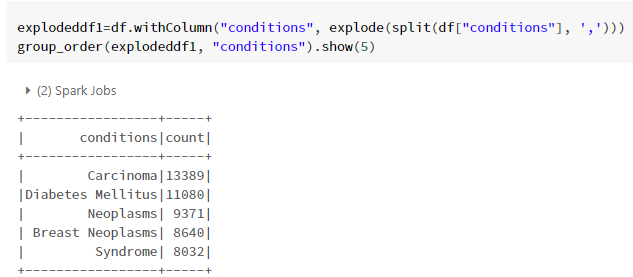
### PYSPARK RDD

In spark rdd we take the conditions column and we split the data and remove null values and perform reduce and sorting functions.



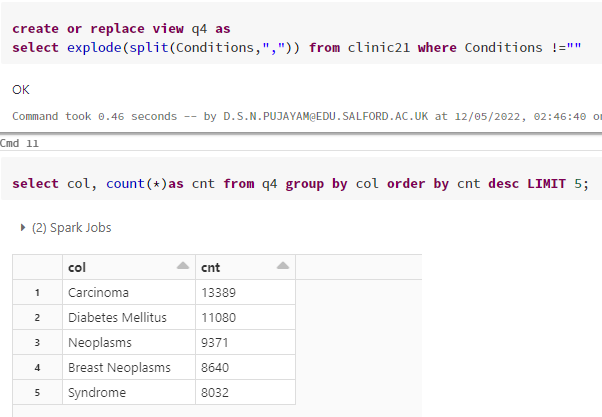
### DATAFRAMES

Here we are taking conditions column using with column function and performing split, explode and user defined functions to get top 5 frequencies.



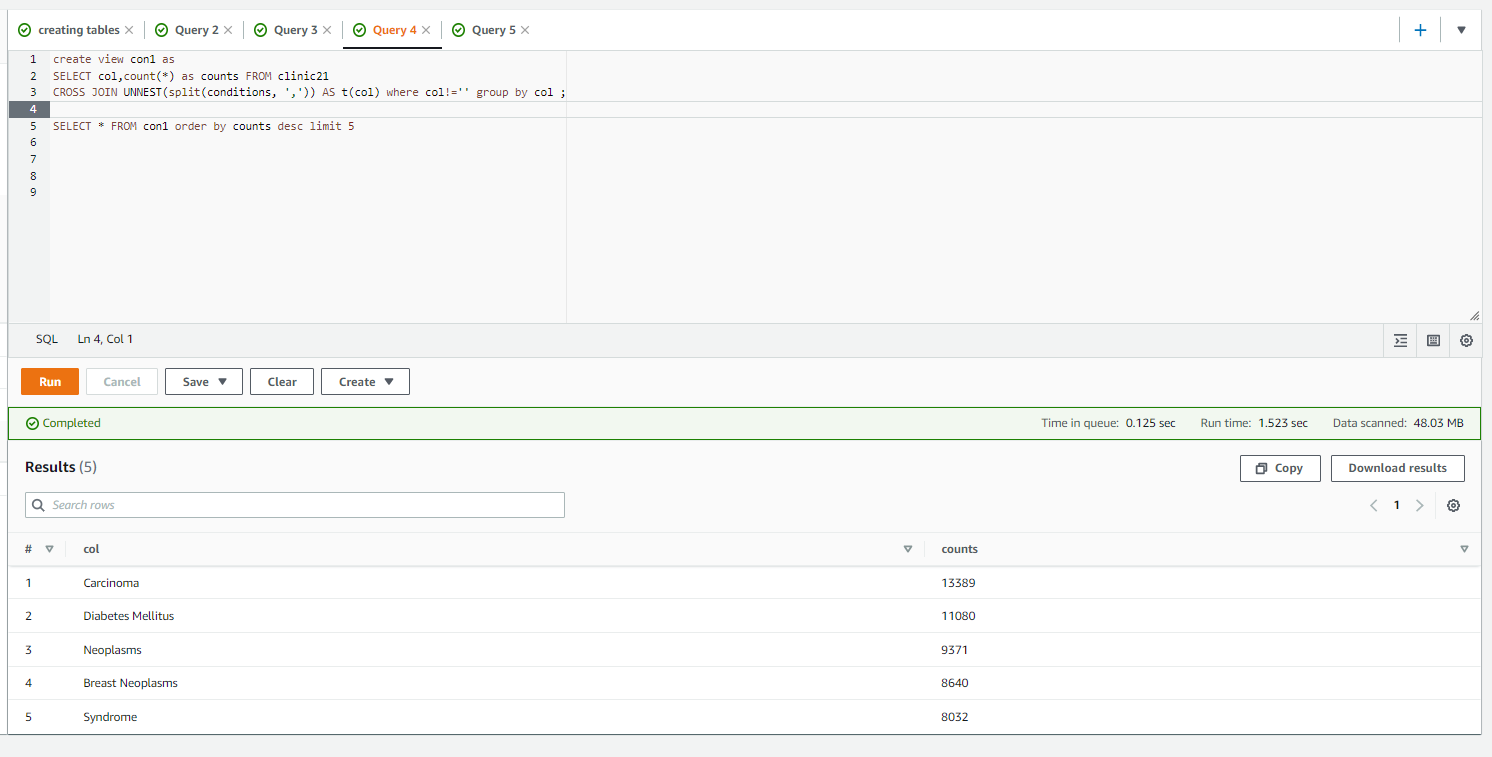
### HIVEQL

In hive we created a view which contains the processed conditions column, using the view we perform groupby and orderby functions to find the frequencies.



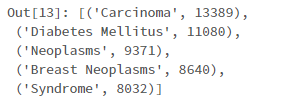
### AWS ATHENA

In athena we create a view which contains conditions column and count, we also use unnest and cross join functions to flatten the data.

****

### Discussion of result

After the analysis of the data and by seeing the results, cancer is the most common condition and diabetes to follow.



## Question 4

### Aim

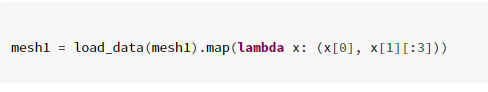
Each condition can be mapped to one or more hierarchy codes. The client wishes to know the 5 most frequent roots (i.e. the sequence of letters and numbers before the first full stop) after this is done.

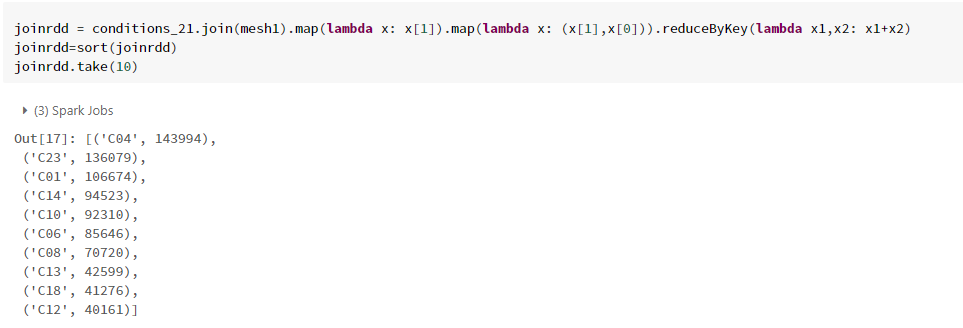
### Assumptions made

Assumed that we need to take the processed conditions column which can be taken from before analysis we did and we also require codes from mesh data set.

### PYSPARK RDD

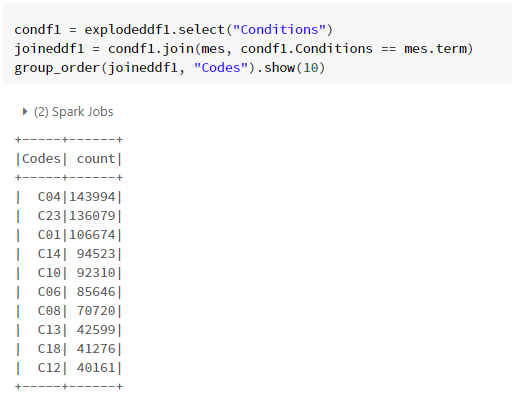
Here we are taking conditions column which is already processed and we are joining conditions column with codes column from mesh dataset and perform in-built and user-defined functions to generate frequencies of the matching codes.





### DATAFRAMES

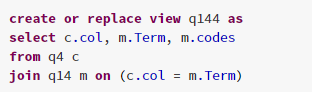
Here we are process mesh data and remove inconsistencies and take first 3 characters from codes column and join the mesh data with conditions column and perform in-built and user defined functions to generate frequencies.

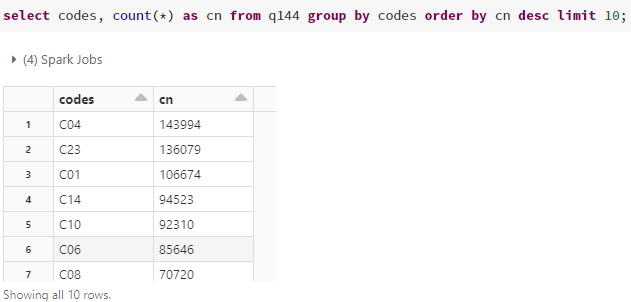


### HIVEQL

Here we take all the codes from mesh data and create a view and join the conditions column with codes in another view and perform built-in functions to generate frequencies.

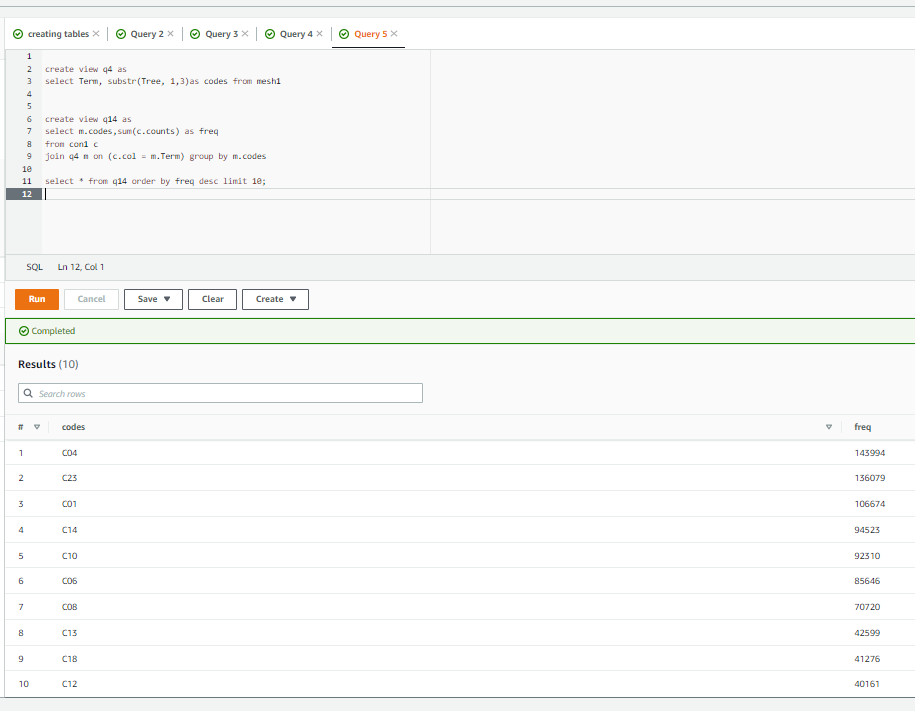






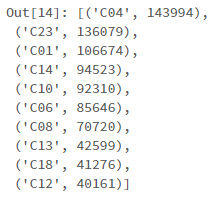
### AWS ATHENA

Here we select all the codes from mesh data and create a view and join it with conditions column. By performing built-in functions we generate frequencies for given data.



### Discussion of result

By the results we are evident that co4,c23,c01 codes have more than 100 thousand frequencies each.



## Question 5

### Aim

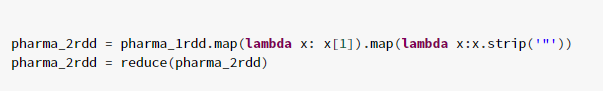
Find the 10 most common sponsors that are not pharmaceutical companies, along with the number of clinical trials they have sponsored.

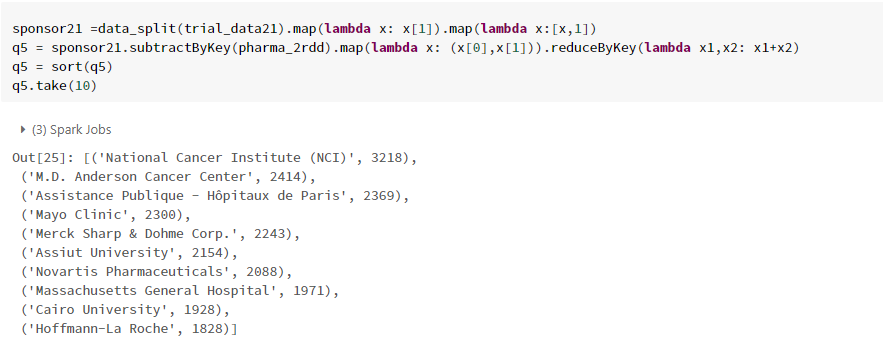
### Assumptions made

Assumed that we need to take sponsors column from clinical trial data and parent company column from pharma datasets.

### PYSPARK RDD

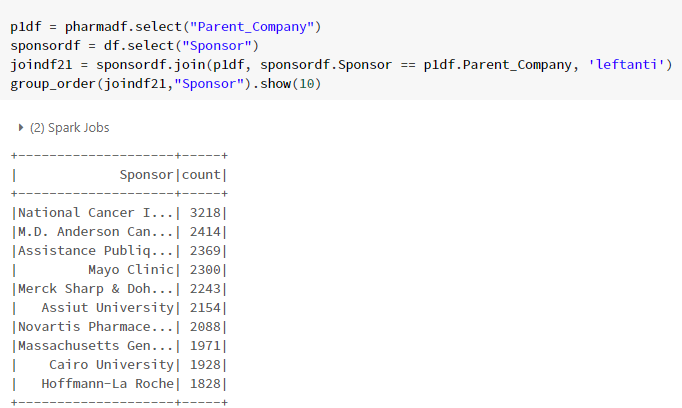
Here we are taking parent company column and removing inconsistencies in the data and joining with sponsors column from clinical trials and we perform subractbykey function to remove non matching values and further perform in-built and user defined functions to generate frequencies for non pharmaceutical companies.





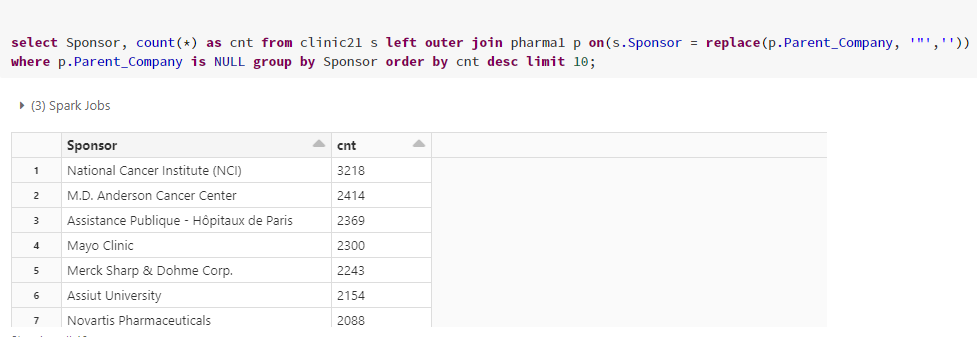
### DATAFRAMES

Here we are taking parent company from pharma data frame and join with sponsors column from clinical trials data and use leftanti function to get data which is not matching to generate frequencies.



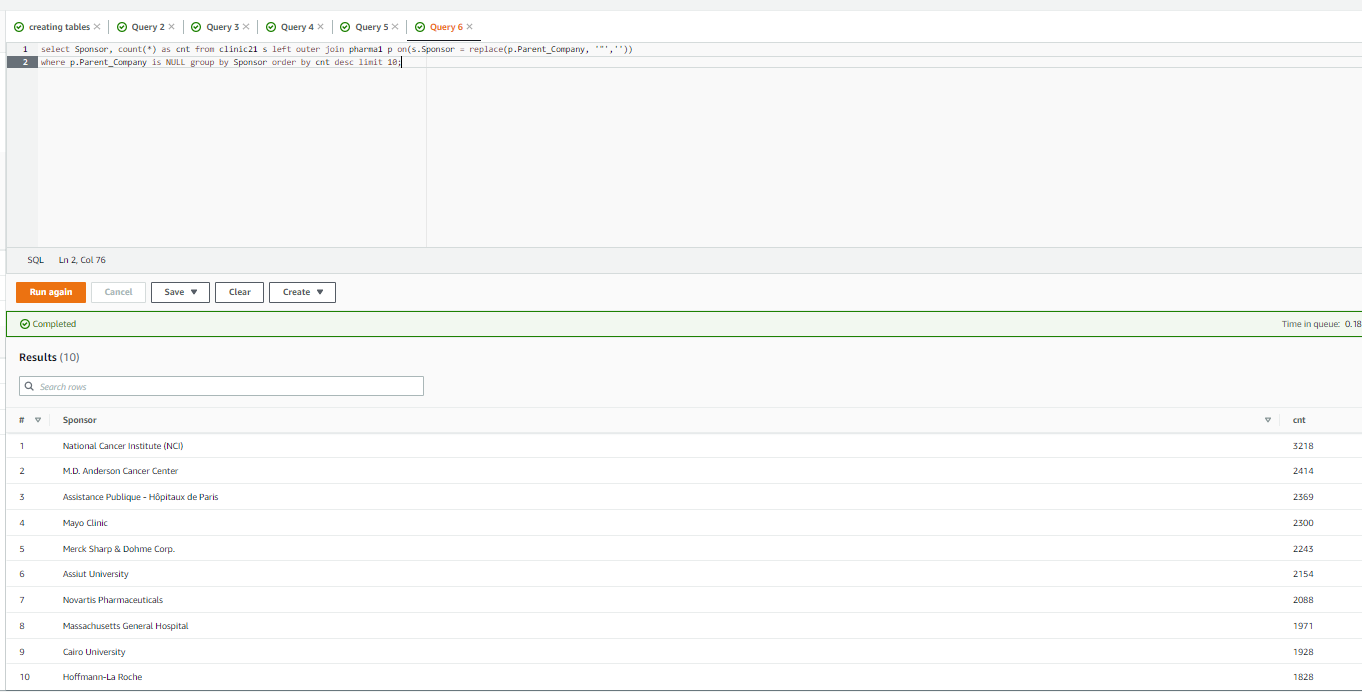
### HIVEQL

Here we are using left outer join function to join and get non matching values and perform built-in functions to generate frequencies.



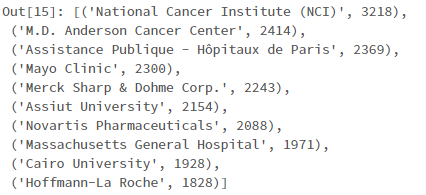
### AWS ATHENA

Here we are taking sponsor column from clinical trials data and joining it with parent company using left outer join function and grouping by sponsors and generating frequencies.



### Discussion of result

After analysing the data here are the top 10 non pharma companies that are sponsoring studies and National Cancer Institute(NCI) is sponsoring more number of studies with 3218 studies.



## Question 6

### Aim

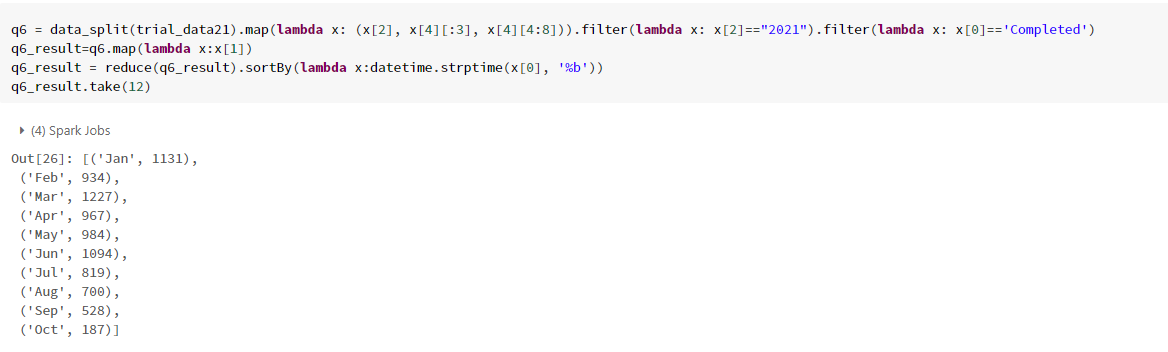
TO plot number of completed studies each month in a given year – 2021 and add visualization of all the values for each month.

### Assumptions made

Assumed that we need to take status column and completion date column and process the data to find frequencies.

### PYSPARK RDD

Here we are taking the required data and using built-in functions like map which applies to each element and filter function which is used to filter data and sort by which is used to sort elements through lambda expressions and user defined functions to generate frequencies.

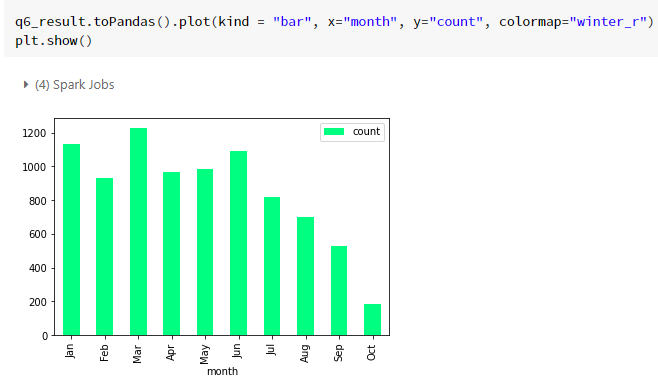


### DATAFRAMES

Here we are selecting status column and separating month and year using substr function from clinical trial data using built-in functions we are generating frequencies.



Below is the visualisation chart for studies completed for each month using pandas in this plotting we used bar chart, on x-axis we took months and count on y-axis



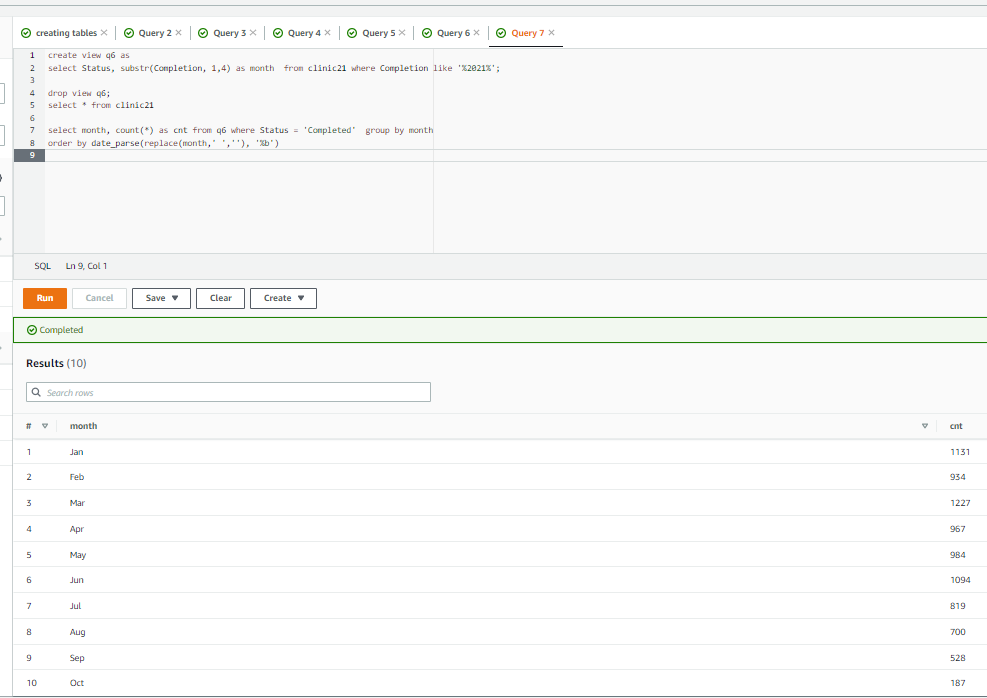
### HIVEQL

Here we create a view that contains the status column, the month-and-year column that we separated using the substr function on the completion column of clinical trial data using this view, we are retrieving the frequencies of all the studies completed for each Month in 2021 using where clause and order are by function.



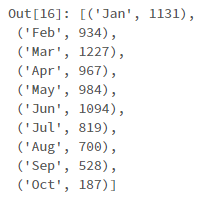
### AWS ATHENA

Here we are creating a view which contains Status column, month & year column which we separated using substr function from completion column from clinical trials data. With the help of this view, we were able to retrieve frequencies of all completed studies in the year 2021 based on a where clause and order by function.



### Discussion of result

After the analysis of this data we found that the total completed studies in the year 2021 is 8571 and the month march has highest studies completed with 1227 and October has least frequency with 187 studies.



## Further analysis 1

### Aim

To find frequencies of sponsoring countries.

### Assumptions made

Assumed that we need to take required columns and match them with clinical trials data to generate results.

### DATAFRAMES

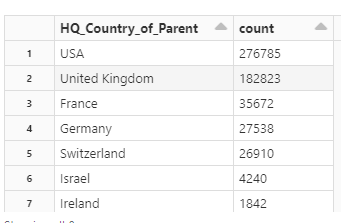
Here we are taking parent company column and country of parent column and join the data with sponsor column from clinical trials dataset where both values are matching, after joining the columns we use user defined function group\_order which is used for grouping the data and order by count to generate frequencies.

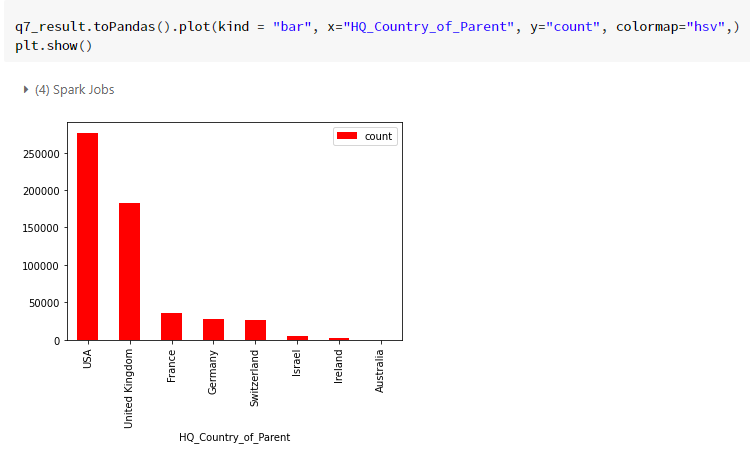




### Discussion of result

After analysing the data we are evident of the result showing countries and number of studies they are sponsoring. Here USA tops which sponsors 276785 studies. We also did plotting of the results we got using pandas bar chart, where we can see country on x-axis and number of studies on y-axis.





## Further analysis 2

### Aim

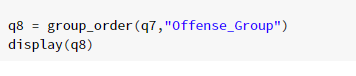
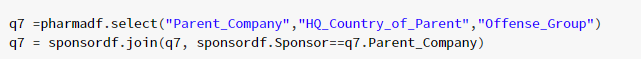
To find frequencies by Offence categories.

### Assumptions made

Assumed that we need to take required columns and match them with clinical trials data to generate results.

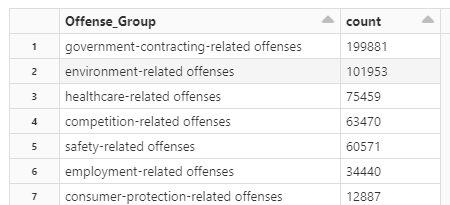
### DATAFRAMES

Here we are taking parent company column and offense group column from pharma data frame and joining with sponsors column from clinical trial data where sponsor values with parent company values, after joining we give these values in user defined function which groups by column and order by count and generate frequencies.

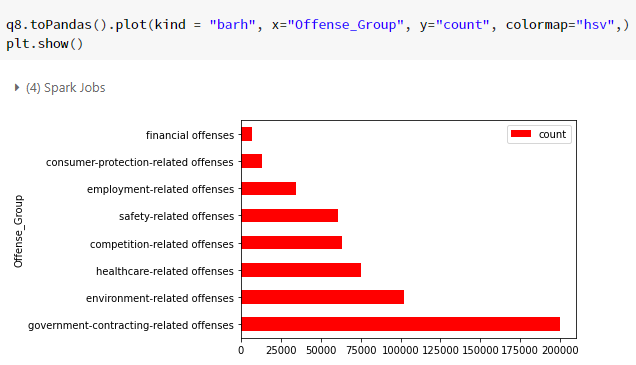


### Discussion of result

After the analysis we now have categories of offences that companies made and their frequencies. We can see from the results that most companies are not complying with government contracting laws.



Here is the horizontal bar chart plotted using pandas which displays offence group on x-axis and frequencies on y-axis.



## Further analysis 3

### Aim

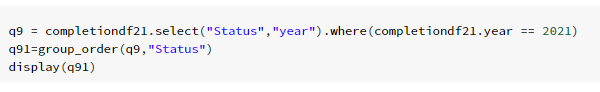
To find frequencies of studies for the year 2021.

### Assumptions made

Assumed that we need to take status of studies for year 2021.

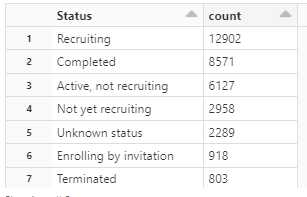
### DATAFRAMES

Here we are taking status column and filtering the values by using where clause to get 2021 studies, now we use user defined function group\_order which groups the values, counts them and orderby the counts.

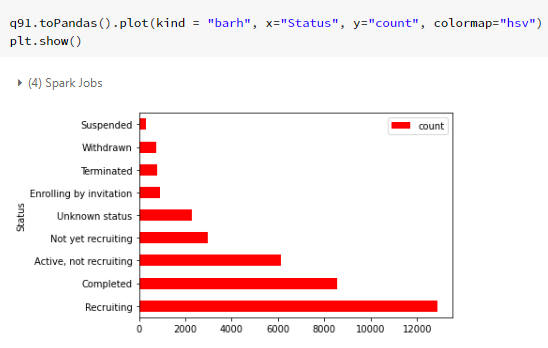


### Discussion of result

After the analysis of the data we can see that recruiting has the highest



We are plotting the data we analysed using pandas horizontal bar chart where status on x-axis and counts on y-axis.



## Further analysis 3

### Aim

To find frequencies of 10 most repeated diseases.

### Assumptions made

Assumed that we need to analyse the conditions column.

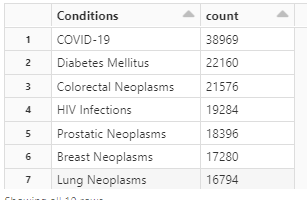
### DATAFRAMES

Here we are taking conditions column and filtering the values, now we use user defined function group\_order which groups the values, counts them and order by the counts.

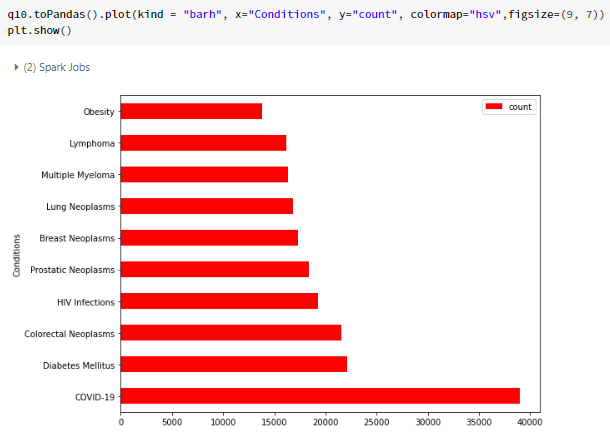


### Discussion of result

After the analysis of the data we found that COVID19 diseases are high in number with 38969 cases and Diabetes with 22160 cases.



We are plotting the most repeated diseases using pandas horizontal bar chart where diseases on x-axis and counts on y-axis.



# References

* **Stack overflow** - https://stackoverflow.com/questions/23205606/regex-to-remove-comma-between-double-quotes-notepad