BIG DATA TOOLS AND TECHNIQUES

ANALYSIS OF CLINICAL TRIAL DATA

M.SC DATA SCIENCE

PROJECT REPORT

SAKSHI SINGH

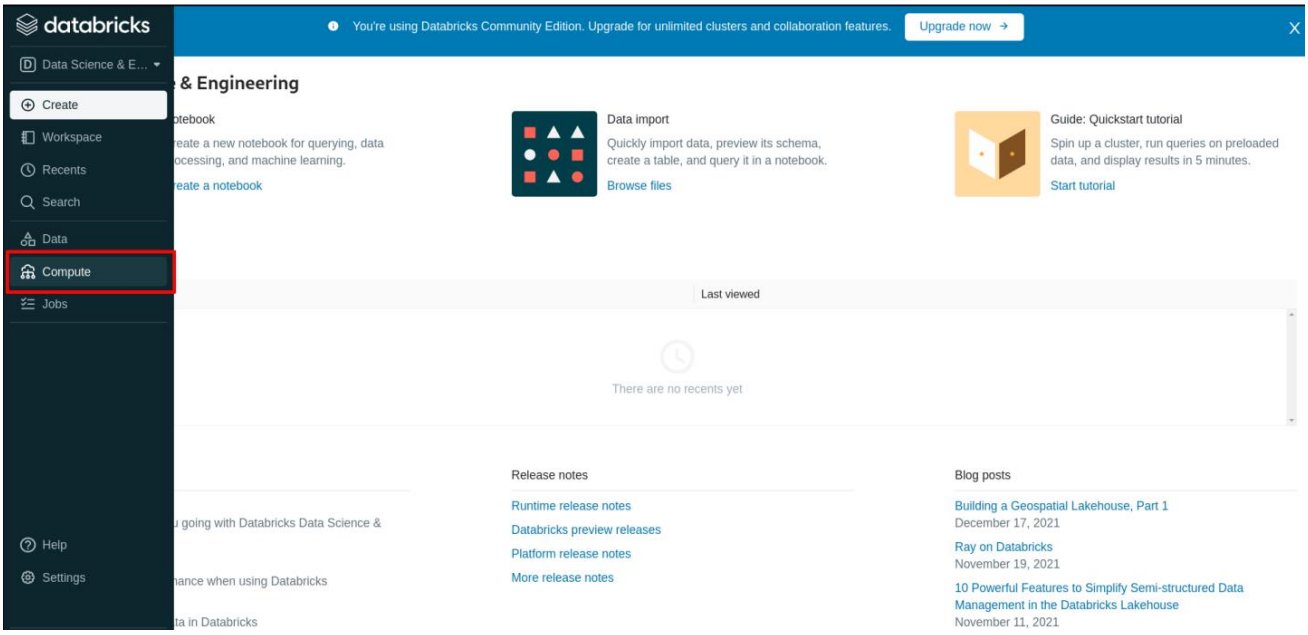
STUDENT ID - *@0065469*

Table of Contents

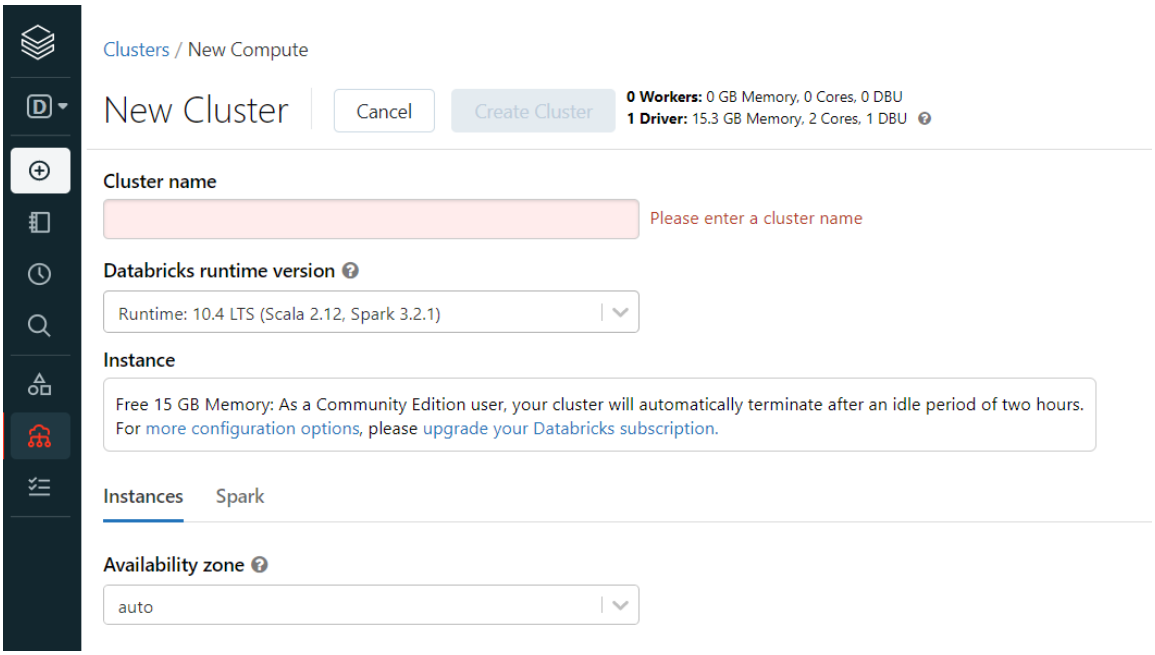
Accessing DataBricks:

For this project I am working on Databricks website. To working on Databricks, we need to attach Databricks clusters that use to run the data from various fields. After open the Databricks website, we will see this home page that could be seem below.

1. For accessing the machine, we need to create cluster. For that Click on 'COMPUTE' on the home page.

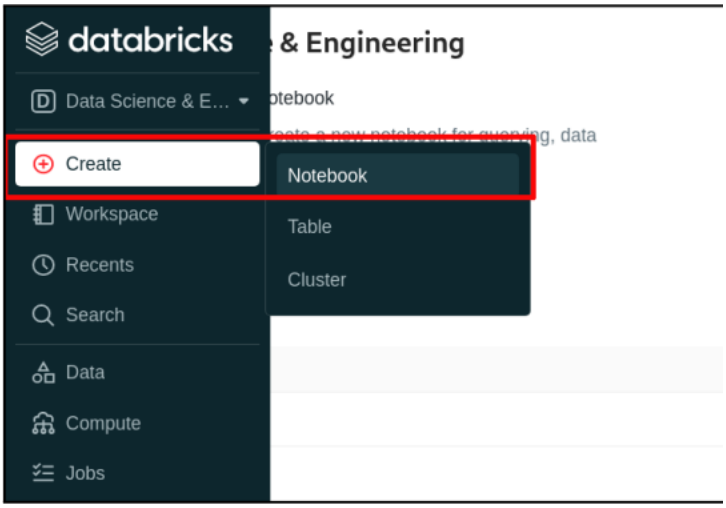


2. Click on create cluster.

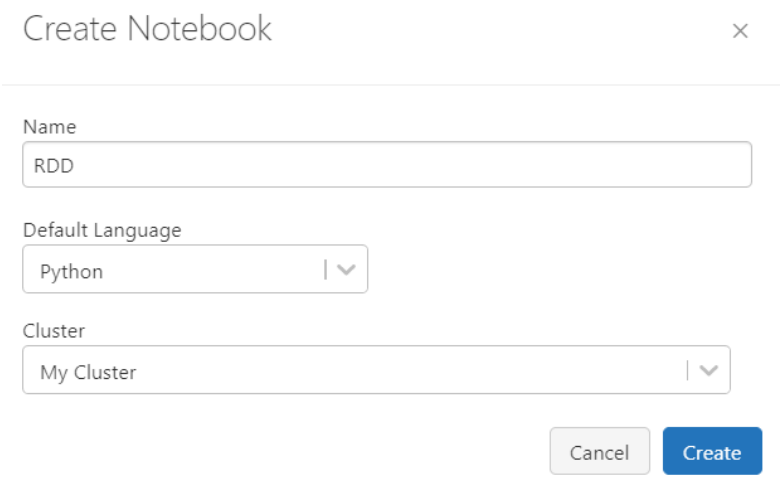


3. Give new name to the cluster.
4. Select most recent runtime version.
   Note - It takes few minutes to create the cluster.

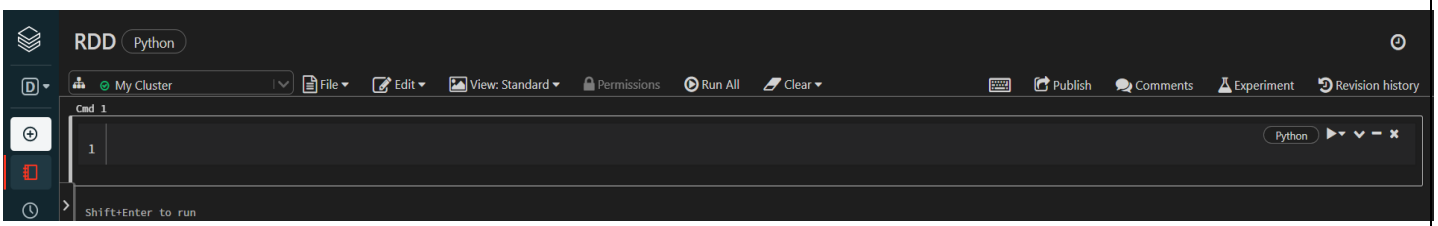
5. As we are working on PySpark and HiveQL so need to create new notebook. Click on Create on left side and select Notebook.



6. Name the Notebook. Select the required Default Language and the created Cluster. Then click on create.

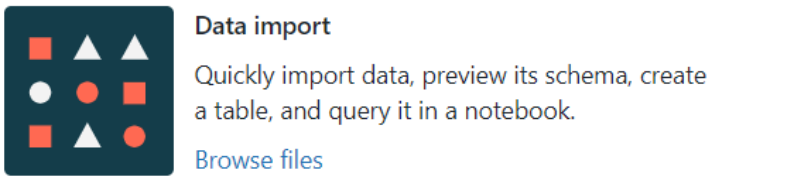


7. After accessing the Notebook, it will look like this.

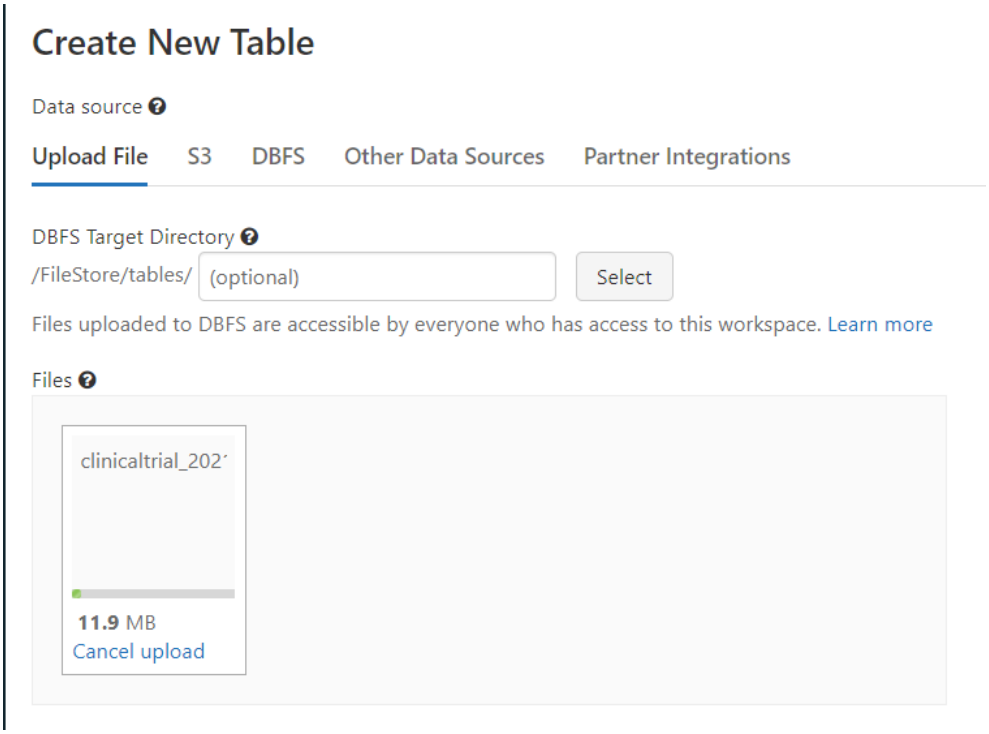


LOADING DATA:

1. First, we need to import data in Databricks. For that we need to browse files. Click on browse file on the home page.



2. Here the file will be downloaded.



3. The file will be placed in the location.

Description of any setup required: For this project I have used two Limitations that is

- PySpark
- HIVEQL
  For each script like PySpark and HiveQL, I have placed the snippet of each step below.

Data cleaning: I have cleaned the data for problem 4,5 and 6 as required. Whose scripts are attached in this report step wise.

Data preparation: Unzipping the File and steps are as follows.

Steps for unzip the file:

1. To check the list of the utility available along with the description, I have applied 'dbutils.fs.ls' command.
   Here I have applied ls command to check the list for each three files i.e., clinicaltrial_2019_csv.gz, clinicaltrial_2020_csv.gz, clinicaltrial_2021_csv.gz.

   RDD Implementation:

```
1  dbutils.fs.ls("FileStore/tables/clinicaltrial_2019_csv.gz")

Out[88]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2019_csv.gz', name='clinicaltrial_2019_csv.gz', size=10060669, modificationTime=1648830413000)]
Command took 0.09 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:37:56 PM on My Cluster

Cmd 2
1  dbutils.fs.ls("FileStore/tables/clinicaltrial_2020_csv.gz")

Out[89]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2020_csv.gz', name='clinicaltrial_2020_csv.gz', size=10981608, modificationTime=1650375979000)]
Command took 0.07 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:37:58 PM on My Cluster

Cmd 3
1  dbutils.fs.ls("FileStore/tables/clinicaltrial_2021_csv.gz")

Out[90]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2021_csv.gz', name='clinicaltrial_2021_csv.gz', size=11921810, modificationTime=1650376267000)]
Command took 0.05 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:38:01 PM on My Cluster
```

2. We need to extract this gz archive. As the dbutils does not provide an unzip command, so we need to copy the file to the driver node, using a shell command we extract the files and then put the extracted files back into DBFS. So, first copy all the files from DBFS to the /tmp directory on your driver node using dbutils.

3. Just to check the list of the temporary files in /tmp directory.

```
%sh
ls /tmp/
```

```
Rserv
RtmpyPciaR
chauffeur-daemon-params
chauffeur-daemon.pid
chauffeur-env.sh
custom-spark.conf
driver-daemon-params
driver-daemon.pid
driver-env.sh
hsperfdata_root
ipykernel-connection-ReplId-28bf0-826f4-33124-6.json
ipykernel-connection-ReplId-48f02-857a3-d570d-b.json
ipykernel-connection-ReplId-56163-4e5dc-360fb-e.json
ipykernel-connection-ReplId-78ec6-c7f23-9e38c-8.json
systemd-private-1e669e518c25444099dd12a8e29c20a1-apache2.service-mfRZYe
systemd-private-1e669e518c25444099dd12a8e29c20a1-ntp.service-wcBuzg
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-logind.service-A2fVRh
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-resolved.service-jddNVe
tmp.kVfimkUw4h
```

4. Copy the all files from FileStore to /tmp directory.

```
dbutils.fs.cp("/FileStore/tables/clinicaltrial_2019_csv.gz", "file:/tmp/")
```

```
Out[92]: True
Command took 0.47 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:38:30 PM on My Cluster
```

```
md 6
dbutils.fs.cp("/FileStore/tables/clinicaltrial_2020_csv.gz", "file:/tmp/")
```

```
Out[93]: True
Command took 0.41 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:38:31 PM on My Cluster
```

```
md 7
dbutils.fs.cp("/FileStore/tables/clinicaltrial_2021_csv.gz", "file:/tmp/")
```

```
Out[94]: True
Command took 0.46 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:38:33 PM on My Cluster
```

5. To verify the files has been copied.

```
%sh
ls /tmp
```

```
Rserv
RtmpyPciaR
chauffeur-daemon-params
chauffeur-daemon.pid
chauffeur-env.sh
clinicaltrial_2019_csv.gz
clinicaltrial_2020_csv.gz
clinicaltrial_2021_csv.gz
custom-spark.conf
driver-daemon-params
driver-daemon.pid
driver-env.sh
hsperfdata_root
ipykernel-connection-ReplId-28bf0-826f4-33124-6.json
ipykernel-connection-ReplId-48f02-857a3-d570d-b.json
ipykernel-connection-ReplId-56163-4e5dc-360fb-e.json
ipykernel-connection-ReplId-78ec6-c7f23-9e38c-8.json
systemd-private-1e669e518c25444099dd12a8e29c20a1-apache2.service-mfRZYe
systemd-private-1e669e518c25444099dd12a8e29c20a1-ntp.service-wcBuzg
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-logind.service-A2fVRh
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-resolved.service-iddNVe
```

Here we can see the files has been copied.

6. Used LINUX command to extract the files.

```
%sh
gunzip -d /tmp/ /tmp/clinicaltrial_2019_csv.gz
```

```
gzip: /tmp/ is a directory -- ignored
```

```
Command took 0.37 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:06 PM on My Cluster
```

```
md 10
```

```
%sh
gunzip -d /tmp/ /tmp/clinicaltrial_2020_csv.gz
```

```
gzip: /tmp/ is a directory -- ignored
```

```
Command took 0.40 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:09 PM on My Cluster
```

```
md 11
```

```
%sh
gunzip -d /tmp/ /tmp/clinicaltrial_2021_csv.gz
```

```
gzip: /tmp/ is a directory -- ignored
```

```
Command took 0.44 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:11 PM on My Cluster
```

7. Again, to verify the extracted files used /tmp command.

```
%sh
ls /tmp
```

```
Rserv
RtmpyPciaR
chauffeur-daemon-params
chauffeur-daemon.pid
chauffeur-env.sh
clinicaltrial_2019_csv
clinicaltrial_2020_csv
clinicaltrial_2021_csv
custom-spark.conf
driver-daemon-params
driver-daemon.pid
driver-env.sh
hsperfdata_root
ipykernel-connection-ReplId-28bf0-826f4-33124-6.json
ipykernel-connection-ReplId-48f02-857a3-d570d-b.json
ipykernel-connection-ReplId-56163-4e5dc-360fb-e.json
ipykernel-connection-ReplId-78ec6-c7f23-9e38c-8.json
systemd-private-1e669e518c25444099dd12a8e29c20a1-apache2.service-mfRZYe
systemd-private-1e669e518c25444099dd12a8e29c20a1-ntp.service-wcBuzg
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-logind.service-A2fVRh
systemd-private-1e669e518c25444099dd12a8e29c20a1-systemd-resolved.service-iddNVe
```

8. Now moved files back to the DBFS from /tmp.

```
dbutils.fs.mv("file:/tmp/clinicaltrial_2019_csv", "/FileStore/tables/clinicaltrial_2019.csv", True )
```

```
Out[100]: True
Command took 2.13 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:36 PM on My Cluster
```

```
Cmd 14
dbutils.fs.mv("file:/tmp/clinicaltrial_2020_csv", "/FileStore/tables/clinicaltrial_2020.csv", True )
Out[101]: True
Command took 2.13 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:40 PM on My Cluster
Cmd 15
dbutils.fs.mv("file:/tmp/clinicaltrial_2021_csv", "/FileStore/tables/clinicaltrial_2021.csv", True )
Out[102]: True
Command took 2.32 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:39:43 PM on My Cluster
```

9. After moving the files to DBFS, check the list of the files.

```
dbutils.fs.ls("FileStore/tables/clinicaltrial_2019.csv")
```

```
Out[103]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2019.csv', name='clinicaltrial_2019.csv', size=42400056, modificationTime=1652099978000)]
Command took 0.08 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:40:04 PM on My Cluster
```

```
Cmd 17
dbutils.fs.ls("FileStore/tables/clinicaltrial_2020.csv")
Out[104]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2020.csv', name='clinicaltrial_2020.csv', size=46318151, modificationTime=1652099982000)]
Command took 0.08 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:40:08 PM on My Cluster
Cmd 18
dbutils.fs.ls("FileStore/tables/clinicaltrial_2021.csv")
Out[105]: [FileInfo(path='dbfs:/FileStore/tables/clinicaltrial_2021.csv', name='clinicaltrial_2021.csv', size=50359696, modificationTime=1652099986000)]
Command took 0.08 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:40:10 PM on My Cluster
```

- This command the returned the maximum number of bytes of the given file.

```
1  dbutils.fs.head("FileStore/tables/clinicaltrial_2019.csv")
```

```
[Truncated to first 65536 bytes]
Out[53]: "Id|Sponsor|Status|Start|Completion|Type|Submission|Conditions|Interventions\r\nNCT02758028|The University of Hong Kong|Recruiting|Aug 2005|Nov 2021|Interventional|Apr 20
16||\r\nNCT02751957|Duke University|Completed|Jul 2016|Jul 2020|Interventional|Apr 2016|Autistic Disorder,Autism Spectrum Disorder|\r\nNCT02758483|Universidade Federal do Rio de J
aneiro|Completed|Mar 2017|Jan 2018|Interventional|Apr 2016|Diabetes Mellitus|\r\nNCT02759848|Istanbul Medeniyet University|Completed|Jan 2012|Dec 2014|Observational|May 2016|Tuber
culosis,Lung Diseases,Pulmonary Disease|\r\nNCT02758860|University of Roma La Sapienza|Active, not recruiting|Jun 2016|Sep 2020|Observational [Patient Registry]|Apr 2016|Diverticu
lar Diseases,Diverticulum,Diverticulosis|\r\nNCT02757209|Consorzio Futuro in Ricerca|Completed|Apr 2016|Jan 2018|Interventional|Apr 2016|Asthma|Fluticasone,Xhance,Budesonide,Formo
terol Fumarate,Salmeterol Xinafoate\r\nNCT02752438|Ankara University|Unknown status|May 2016|Jul 2017|Observational [Patient Registry]|Apr 2016|Hypoventilation|\r\nNCT02753543|Rui
jin Hospital|Unknown status|Nov 2015|Nov 2019|Interventional|Apr 2016|Lymphoma|\r\nNCT02757508|Washington University School of Medicine|Completed|Mar 2016|Jul 2017|Interventional|
Apr 2016||Vitamins\r\nNCT02753530|Orphazyme|Completed|Aug 2017|Jan 2021|Interventional|Apr 2016|Myositis|\r\nNCT02754817|Novo Nordisk A/S|Completed|Apr 2016|Oct 2016|Observational
|Apr 2016|Diabetes Mellitus|Liraglutide,Xultophy\r\nNCT02759276|Daniel Alexandre Bottino|Completed|May 2015|Dec 2015|Observational|Apr 2016|Hypertension|\r\nNCT02750956|Bulent Ece
vit University|Completed|Jun 2015|Mar 2016|Observational|Apr 2016|Periodontal Diseases|\r\nNCT02752113|Institut für Pharmakologie und Präventive Medizin|Completed|Apr 2016|May 201
9|Interventional|Apr 2016|Diabetes Mellitus|Metformin,Empagliflozin,Linagliptin\r\nNCT02752698|The Third Xiangya Hospital of Central South University|Active, not recruiting|Jan 20
15|Dec 2021|Interventional|Jun 2015|Appendicitis,Stomach Ulcer,Cholecystolithiasis,Cholelithiasis,Gallstones|\r\nNCT02755779|Tel Aviv Medical Center|Unknown status|Jun 2016|Jun 20
17|Observational|Apr 2016|||\r\nNCT02750384|Medicines for Malaria Venture|Terminated|May 2016|Jul 2016|Interventional|Apr 2016|||\r\nNCT02754609|James Cook University, Queensland, A
ustralia|Completed|Sep 2016|Oct 2019|Interventional|Apr 2016|Hookworm Infections,Celiac Disease|\r\nNCT02755701|Soonchunhyang University Hospital|Unknown status|Jul 2016|Dec 2018|
Interventional|Apr 2016|Ascites|\r\nNCT02751762|Member Companies of the Opioid PMR Consortium|Recruiting|Nov 2017|Oct 2022|Observational|Apr 2016|Chronic Pain,Substance-Related Di
sorders,Opioid-Related Disorders,Narcotic-Related Disorders,Behavior|\r\nNCT02756299|Marmara University|Completed|Jun 2014|Apr 2015|Interventional|Apr 2016|Sleep Apnea Syndromes,S
leep Apnea|\r\nNCT02750709|Cycle Pharmaceuticals Ltd.|Completed|Oct 2015|Jan 2016|Interventional|Apr 2016|Tyrosinemias|Nitisinone\r\nNCT02753907|Yonsei University|Completed|Jun 20
15||Interventional|Apr 2016|||\r\nNCT02755467|Cutera Inc.|Completed|May 2016|Apr 2017|Interventional|Apr 2016|Hemangioma|\r\nNCT02755298|University of Zurich|Completed|Oct 2016|Nov
2020|Interventional|Mar 2016|Hypertension|Acetazolamide\r\nNCT02759614|Guangdong Association of Clinical Trials|Unknown status|Apr 2016|Jun 2019|Interventional|Mar 2016|Carcinoma|
Bevacizumab,Erlotinib Hydrochloride\r\nNCT02752815|Ruijin Hospital|Unknown status|Apr 2016|Jun 2020|Interventional|Apr 2016|Lymphoma|Prednisone,Cyclophosphamide,Rituximab,Vincris%
```

```
1  dbutils.fs.head("FileStore/tables/clinicaltrial_2020.csv")
```

```
[Truncated to first 65536 bytes]
Out[54]: "Id|Sponsor|Status|Start|Completion|Type|Submission|Conditions|Interventions\r\nNCT02758028|The University of Hong Kong|Recruiting|Aug 2005|Nov 2021|Interventional|Apr 20
16||\r\nNCT02751957|Duke University|Completed|Jul 2016|Jul 2020|Interventional|Apr 2016|Autistic Disorder,Autism Spectrum Disorder|\r\nNCT02758483|Universidade Federal do Rio de J
aneiro|Completed|Mar 2017|Jan 2018|Interventional|Apr 2016|Diabetes Mellitus|\r\nNCT02759848|Istanbul Medeniyet University|Completed|Jan 2012|Dec 2014|Observational|May 2016|Tuber
culosis,Lung Diseases,Pulmonary Disease|\r\nNCT02758860|University of Roma La Sapienza|Active, not recruiting|Jun 2016|Sep 2020|Observational [Patient Registry]|Apr 2016|Diverticu
lar Diseases,Diverticulum,Diverticulosis|\r\nNCT02757209|Consorzio Futuro in Ricerca|Completed|Apr 2016|Jan 2018|Interventional|Apr 2016|Asthma|Fluticasone,Xhance,Budesonide,Formo
terol Fumarate,Salmeterol Xinafoate\r\nNCT02752438|Ankara University|Unknown status|May 2016|Jul 2017|Observational [Patient Registry]|Apr 2016|Hypoventilation|\r\nNCT02753543|Rui
jin Hospital|Unknown status|Nov 2015|Nov 2019|Interventional|Apr 2016|Lymphoma|\r\nNCT02757508|Washington University School of Medicine|Completed|Mar 2016|Jul 2017|Interventional|
Apr 2016||Vitamins\r\nNCT02753530|Orphazyme|Completed|Aug 2017|Jan 2021|Interventional|Apr 2016|Myositis|\r\nNCT02754817|Novo Nordisk A/S|Completed|Apr 2016|Oct 2016|Observational
|Apr 2016|Diabetes Mellitus|Liraglutide,Xultophy\r\nNCT02759276|Daniel Alexandre Bottino|Completed|May 2015|Dec 2015|Observational|Apr 2016|Hypertension|\r\nNCT02750956|Bulent Ece
vit University|Completed|Jun 2015|Mar 2016|Observational|Apr 2016|Periodontal Diseases|\r\nNCT02752113|Institut für Pharmakologie und Präventive Medizin|Completed|Apr 2016|May 201
9|Interventional|Apr 2016|Diabetes Mellitus|Metformin,Empagliflozin,Linagliptin\r\nNCT02752698|The Third Xiangya Hospital of Central South University|Active, not recruiting|Jan 20
15|Dec 2021|Interventional|Jun 2015|Appendicitis,Stomach Ulcer,Cholecystolithiasis,Cholelithiasis,Gallstones|\r\nNCT02755779|Tel Aviv Medical Center|Unknown status|Jun 2016|Jun 20
17|Observational|Apr 2016|||\r\nNCT02750384|Medicines for Malaria Venture|Terminated|May 2016|Jul 2016|Interventional|Apr 2016|||\r\nNCT02754609|James Cook University, Queensland, A
ustralia|Completed|Sep 2016|Oct 2019|Interventional|Apr 2016|Hookworm Infections,Celiac Disease|\r\nNCT02755701|Soonchunhyang University Hospital|Unknown status|Jul 2016|Dec 2018|
Interventional|Apr 2016|Ascites|\r\nNCT02751762|Member Companies of the Opioid PMR Consortium|Recruiting|Nov 2017|Oct 2022|Observational|Apr 2016|Chronic Pain,Substance-Related Di
sorders,Opioid-Related Disorders,Narcotic-Related Disorders,Behavior|\r\nNCT02756299|Marmara University|Completed|Jun 2014|Apr 2015|Interventional|Apr 2016|Sleep Apnea Syndromes,S
leep Apnea|\r\nNCT02750709|Cycle Pharmaceuticals Ltd.|Completed|Oct 2015|Jan 2016|Interventional|Apr 2016|Tyrosinemias|Nitisinone\r\nNCT02753907|Yonsei University|Completed|Jun 20
15||Interventional|Apr 2016|||\r\nNCT02755467|Cutera Inc.|Completed|May 2016|Apr 2017|Interventional|Apr 2016|Hemangioma|\r\nNCT02755298|University of Zurich|Completed|Oct 2016|Nov
2020|Interventional|Mar 2016|Hypertension|Acetazolamide\r\nNCT02759614|Guangdong Association of Clinical Trials|Unknown status|Apr 2016|Jun 2019|Interventional|Mar 2016|Carcinoma|
Bevacizumab,Erlotinib Hydrochloride\r\nNCT02752815|Ruijin Hospital|Unknown status|Apr 2016|Jun 2020|Interventional|Apr 2016|Lymphoma|Prednisone,Cyclophosphamide,Rituximab,Vincris%
```

```
1  dbutils.fs.head("FileStore/tables/clinicaltrial_2020.csv")
```

```
[Truncated to first 65536 bytes]
Out[55]: "Id|Sponsor|Status|Start|Completion|Type|Submission|Conditions|Interventions\r\nNCT02758028|The University of Hong Kong|Recruiting|Aug 2005|Nov 2021|Interventional|Apr 20
16||\r\nNCT02751957|Duke University|Completed|Jul 2016|Jul 2020|Interventional|Apr 2016|Autistic Disorder,Autism Spectrum Disorder|\r\nNCT02758483|Universidade Federal do Rio de J
aneiro|Completed|Mar 2017|Jan 2018|Interventional|Apr 2016|Diabetes Mellitus|\r\nNCT02759848|Istanbul Medeniyet University|Completed|Jan 2012|Dec 2014|Observational|May 2016|Tuber
culosis,Lung Diseases,Pulmonary Disease|\r\nNCT02758860|University of Roma La Sapienza|Active, not recruiting|Jun 2016|Sep 2020|Observational [Patient Registry]|Apr 2016|Diverticu
lar Diseases,Diverticulum,Diverticulosis|\r\nNCT02757209|Consorzio Futuro in Ricerca|Completed|Apr 2016|Jan 2018|Interventional|Apr 2016|Asthma|Fluticasone,Xhance,Budesonide,Formo
terol Fumarate,Salmeterol Xinafoate\r\nNCT02752438|Ankara University|Unknown status|May 2016|Jul 2017|Observational [Patient Registry]|Apr 2016|Hypoventilation|\r\nNCT02753543|Rui
jin Hospital|Unknown status|Nov 2015|Nov 2019|Interventional|Apr 2016|Lymphoma|\r\nNCT02757508|Washington University School of Medicine|Completed|Mar 2016|Jul 2017|Interventional|
Apr 2016||Vitamins\r\nNCT02753530|Orphazyme|Completed|Aug 2017|Jan 2021|Interventional|Apr 2016|Myositis|\r\nNCT02754817|Novo Nordisk A/S|Completed|Apr 2016|Oct 2016|Observational
|Apr 2016|Diabetes Mellitus|Liraglutide,Xultophy\r\nNCT02759276|Daniel Alexandre Bottino|Completed|May 2015|Dec 2015|Observational|Apr 2016|Hypertension|\r\nNCT02750956|Bulent Ece
vit University|Completed|Jun 2015|Mar 2016|Observational|Apr 2016|Periodontal Diseases|\r\nNCT02752113|Institut für Pharmakologie und Präventive Medizin|Completed|Apr 2016|May 201
9|Interventional|Apr 2016|Diabetes Mellitus|Metformin,Empagliflozin,Linagliptin\r\nNCT02752698|The Third Xiangya Hospital of Central South University|Active, not recruiting|Jan 20
15|Dec 2021|Interventional|Jun 2015|Appendicitis,Stomach Ulcer,Cholecystolithiasis,Cholelithiasis,Gallstones|\r\nNCT02755779|Tel Aviv Medical Center|Unknown status|Jun 2016|Jun 20
17|Observational|Apr 2016|||\r\nNCT02750384|Medicines for Malaria Venture|Terminated|May 2016|Jul 2016|Interventional|Apr 2016|||\r\nNCT02754609|James Cook University, Queensland, A
ustralia|Completed|Sep 2016|Oct 2019|Interventional|Apr 2016|Hookworm Infections,Celiac Disease|\r\nNCT02755701|Soonchunhyang University Hospital|Unknown status|Jul 2016|Dec 2018|
Interventional|Apr 2016|Ascites|\r\nNCT02751762|Member Companies of the Opioid PMR Consortium|Recruiting|Nov 2017|Oct 2022|Observational|Apr 2016|Chronic Pain,Substance-Related Di
sorders,Opioid-Related Disorders,Narcotic-Related Disorders,Behavior|\r\nNCT02756299|Marmara University|Completed|Jun 2014|Apr 2015|Interventional|Apr 2016|Sleep Apnea Syndromes,S
leep Apnea|\r\nNCT02750709|Cycle Pharmaceuticals Ltd.|Completed|Oct 2015|Jan 2016|Interventional|Apr 2016|Tyrosinemias|Nitisinone\r\nNCT02753907|Yonsei University|Completed|Jun 20
15||Interventional|Apr 2016|||\r\nNCT02755467|Cutera Inc.|Completed|May 2016|Apr 2017|Interventional|Apr 2016|Hemangioma|\r\nNCT02755298|University of Zurich|Completed|Oct 2016|Nov
2020|Interventional|Mar 2016|Hypertension|Acetazolamide\r\nNCT02759614|Guangdong Association of Clinical Trials|Unknown status|Apr 2016|Jun 2019|Interventional|Mar 2016|Carcinoma|
Bevacizumab,Erlotinib Hydrochloride\r\nNCT02752815|Ruijin Hospital|Unknown status|Apr 2016|Jun 2020|Interventional|Apr 2016|Lymphoma|Prednisone,Cyclophosphamide,Rituximab,Vincris%
```

Problem Solving:

Here I am working on 'clinicaltrial_2021_csv.gz', 'mesh.csv' and 'pharma.csv' data so the following implementations are based on the client's requirement. The two implementations have been made 'RDD' and 'HiveQL' with different script and codes and their snippet are shown below with each problem statement.

1. The number of studies in the dataset. You must ensure that you explicitly check distinct studies.
   Assumption made: After observing all the data, it includes some duplicate values as well. So, for that I thought to use data cleansing method after reading the data.
   RDD Implementations:

   - Here firstly, I created the hierarchical file that contains directories and files.

```
fileroot = "clinicaltrial_2021"
import os
os.environ['fileroot'] = fileroot
```

   - To check the count of the dataset. First, we need to create RDD naming RDD21 using 'SparkContext.text' to read the files. From the 'fileroot' file.

```
RDD21 = sc.textFile("dbfs:/FileStore/tables/"+fileroot+".csv")
```

   - As we need to check the count of the dataset so here, we do not need to count header. That's why we applied the following code so that it does not count header after that.

```
header = RDD21.first()
RDD21 = RDD21.filter(lambda x:x !=  header)
```

```
▸ (1) Spark Jobs
Command took 1.04 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

   - So, these are the number of studies in the dataset.

```
RDD21.count()
```

```
▸ (1) Spark Jobs
Out[56]: 387261
Command took 1.86 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

   Result: Got distinct studies from the dataset.

HIVEQL implementation: Hive query language is a platform where we analyse the structured data. For the same question I have created the implementation on HIVEQL as well.

- First, I have created database. Creating database is required to import the any kind of data then we can use it to analyse the scripts.

```
1  CREATE DATABASE if not exists PROJECT;
```

```
OK
Command took 35.29 seconds -- by S.Singh12@edu.salford.ac.uk at 5/11/2022, 4:22:29 PM on PROJECT
```

- Then I have used the database 'PROJECT' to import the csv files.

```
1  USE PROJECT;
```

```
OK
Command took 0.94 seconds -- by S.Singh12@edu.salford.ac.uk at 5/11/2022, 4:22:29 PM on PROJECT
```

- Then I have created a table named 'clinicaltrial_2021'. To imported the data from csv file we used delimited '|' to split the fields and verified the result using 'SELECT' command. Here first row had created duplicate values as header. To remove those duplicate values, I used 'WHERE' clause.

```
1  DROP TABLE IF EXISTS clinicaltrial_2021;
2  CREATE External TABLE if not exists clinicaltrial_2021(Id STRING,
3                Sponsor STRING, Status STRING,
4                Start STRING, Completion STRING,
5                Type STRING ,Submission STRING,
6                Conditions STRING, Interventions STRING)
7  USING CSV
8  OPTIONS (path "dbfs:/FileStore/tables/clinicaltrial_2021.csv",
9         delimiter "|");
10 SELECT * FROM clinicaltrial_2021
11 where Id<>'Id';
```

▸ (1) Spark Jobs

	Id	Sponsor	Status	Start	Completion	Type
1	NCT02758028	The University of Hong Kong	Recruiting	Aug 2005	Nov 2021	Interv
2	NCT02751957	Duke University	Completed	Jul 2016	Jul 2020	Interv
3	NCT02758483	Universidade Federal do Rio de Janeiro	Completed	Mar 2017	Jan 2018	Interv
4	NCT02759848	Istanbul Medeniyet University	Completed	Jan 2012	Dec 2014	Obser
5	NCT02758860	University of Roma La Sapienza	Active, not recruiting	Jun 2016	Sep 2020	Obser
6	NCT02757209	Consorzio Futuro in Ricerca	Completed	Apr 2016	Jan 2018	Interv
7	NCT02752438	Ankara University	Unknown status	May 2016	Jul 2017	Obser

Truncated results, showing first 1000 rows.
Click to re-execute with maximum result limits.

Command took 3.76 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:28:49 AM on My Cluster

- So, the following are the total number of studies in the dataset.

```
1  SELECT COUNT(*) TOTAL FROM clinicaltrial_2021 WHERE Id<>'Id';
```

▸ (2) Spark Jobs

	TOTAL
1	387261

Showing all 1 rows.

---

2. You should list all the types (as contained in the Type column) of studies in the dataset along with the frequencies of each type. These should be ordered from most frequent to least frequent.
   Assumption made: The dataset in the file is in CSV format, so first I assumed to separate the columns. Then apply the suitable code to get the resultant answer.

   RDD Implementation:

   - In this command I have used the split function. In order to split the strings of the column, we need split function. So, I have replaced the delimiter '|' by splitting it by ','.

```
1  #Replaced the delimiter "|" with ","
2  splitRDD21 = RDD21.map(lambda line: line.replace("|",",").split(","))
```

Command took 0.03 seconds -- by S.Singh12@edu.salford.ac.uk at 5/10/2022, 10:38:45 PM on My Cluster

   - For verification.
     Here it shows the result in the form of array.

```
1  splitRDD21.take(2)
```

▸ (1) Spark Jobs

```
Out[58]: [['NCT02758028',
  'The University of Hong Kong',
  'Recruiting',
  'Aug 2005',
  'Nov 2021',
  'Interventional',
  'Apr 2016',
  '',
  ''],
 ['NCT02751957',
  'Duke University',
  'Completed',
  'Jul 2016',
  'Jul 2020',
  'Interventional',
  'Apr 2016',
  'Autistic Disorder',
  'Autism Spectrum Disorder',
  '']]
```

Command took 0.93 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster

- So here we need list of the types of 'Type' column with their count and must be sorted in highest frequency to lowest. As we need to count of each type so I have used 'reduceByKey' function where it selects one type and if it gets the same type then it will be added further.

```
#used reduceByKey to combine the same value and sort in descending order.
TypeRDD21= splitRDD21.map(lambda x: x[5])
TypeRDD21.map(lambda a:(a,1)).reduceByKey(lambda b1,b2:b1+b2).sortBy(lambda b:b[1],ascending =False).take(4)
```

```
▸ (4) Spark Jobs
Out[106]: [('Interventional', 301472),
 ('Observational', 77540),
 ('Observational [Patient Registry]', 8180),
 ('Expanded Access', 69)]
Command took 3.75 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 4:39:00 PM on My Cluster
```

HIVE Implementation:

- As the question above, I have selected the 'Type' column with count as 'Total' from table 'clinicaltrial_2021' that I have created. Again, I have used 'WHERE' clause because it duplicates the header. So, to remove this by using 'WHERE' clause.

```
SELECT Type, COUNT(Type) AS TOTAL
FROM clinicaltrial_2021
WHERE TYPE IS NOT NULL AND TYPE <>'Type'
GROUP BY clinicaltrial_2021.Type
ORDER BY TOTAL DESC;
```

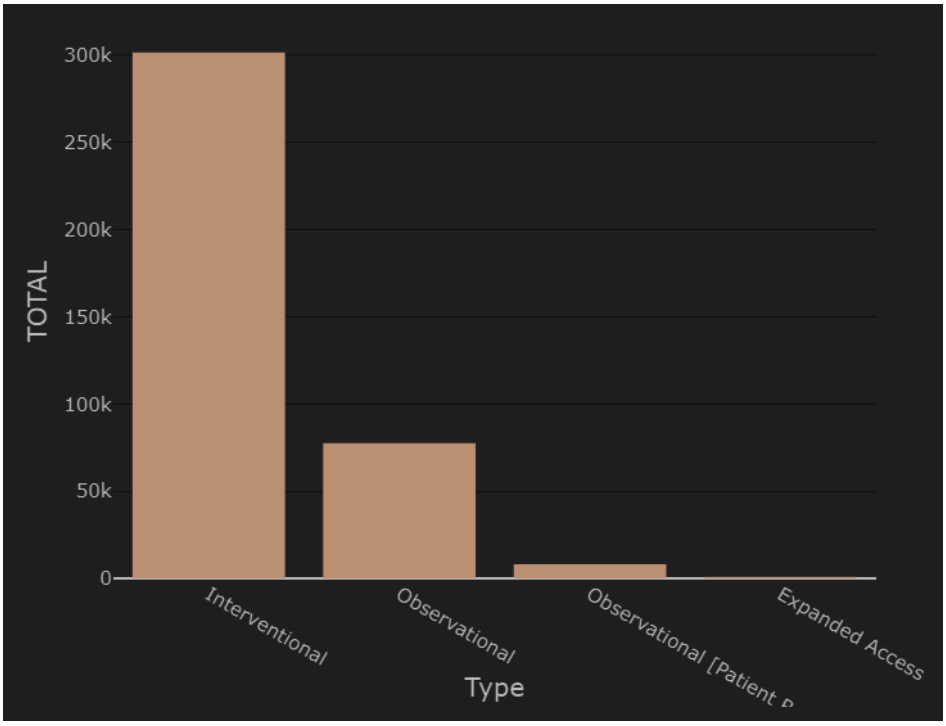
▸ (2) Spark Jobs

	Type	TOTAL
1	Interventional	301472
2	Observational	77540
3	Observational [Patient Registry]	8180
4	Expanded Access	69

Showing all 4 rows.

Command took 6.21 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:29:07 AM on My Cluster

The above result is as per the required question and visualization of the report can be seen in the following bar graph.



Result: The result shows the five different types of 'Type'. The top most type is Interventional and least is Expanded access which has huge difference of 301,403. While looking at bar graph that clearly shows the value of Expanded access as negligible. Though, we can say the most used type is 'Interventional'.

---

3. The top 5 conditions (from Conditions) with their frequencies.
Assumption made: This is almost same as the above problem. Where I need to separate the column and lookup for the clients' requirement.

RDD Implementation:

- Here we need top 5 count of the 'Conditions' column. As in the query,
  First split the column using delimiter '|'
  Remove header count.
  Then substitute '|' with ',' and searched for the number of counts of that value. After this, merge the count by adding using 'reduceByKey' and sorted in descending order.

```
#Used flatMap to produce required value along with reduceByKey and SortByKey
New = RDD21.map(lambda c: c.split('|'))
head =New.first()
NEW1 = New.filter(lambda r:  r != head)
DJ = New.flatMap(lambda c: c[7].split(',')).map(lambda y:[y,1]).reduceByKey(lambda a1,a2:a1+a2).map(lambda a:(a[1],a[0])).sortByKey(ascending = False).map(lambda a:(a[1],a[0])).filter(lambda s:(s[0]!=''))
DJ.take(5)
```

```
▸ (4) Spark Jobs
Out[60]: [('Carcinoma', 13389),
 ('Diabetes Mellitus', 11080),
 ('Neoplasms', 9371),
 ('Breast Neoplasms', 8640),
 ('Syndrome', 8032)]
Command took 5.45 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

HIVEQL Implementations:

- In this query, here I used 'Lateral view' to expand the array into a row. When we use lateral view with explode it will show the result as follows.
  As you can see, I have used 'HAVING' clause. It is due to I got the top row having value '65131' which I did not need. So, remove this from the row by using 'HAVING' clause.

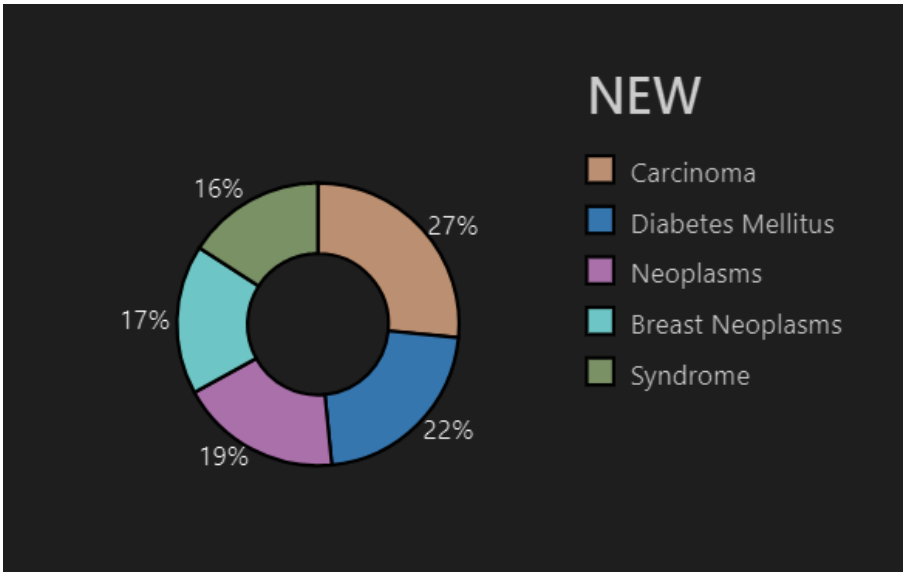
```
1  SELECT NEW, COUNT(NEW) AS TOTAL
2  FROM clinicaltrial_2021 lateral view explode(split(Conditions,',')) Conditions AS NEW
3  GROUP BY NEW
4  HAVING TOTAL<65131
5  ORDER BY TOTAL DESC
6  LIMIT 5
```

▸ (2) Spark Jobs

	NEW	TOTAL
1	Carcinoma	13389
2	Diabetes Mellitus	11080
3	Neoplasms	9371
4	Breast Neoplasms	8640
5	Syndrome	8032

Showing all 5 rows.

Command took 3.80 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:29:25 AM on My Cluster



Result: Pie chart visualization for this result which shows the values in percentage is quite helpful to understand the data. Maximum number of conditions we could get in 'CARCINOMA' with 27%. However, with this data, least are 'SYNDROME'. The maximum difference can be seen in between 'CARCINOMA' and 'BREAST NEOPLASM' i.e., 5%. Else we can see the alikeness in the values of other three.

---

4. 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.

   Assumption made: To make the result desirable, I will the use the ETL process that extract the data and refine it. Use the codes that help me for cleansing the files.

   RDD Implementations:

   - To read the file I have used the 'sc.textFile'.

```
1  MESH =  sc.textFile("FileStore/tables/mesh.csv")
```

```
Command took 0.08 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

   - Then I used split command, where I replaced blank with ',' and '.' With ','. Then map function runs lambda over the list. And see top 5 values.

```
1  #Replaced and split  " "(blank) and "." with ","
2  MESH_Split = MESH.map(lambda t: t.replace('"','').replace('.',',').split(',')).map(lambda l: (l[0],l[1]))
3  MESH_Split.take(5)
```

```
▸ (1) Spark Jobs

Out[62]: [('term', 'tree'),
 ('Calcimycin', 'D03'),
 ('A-23187', 'D03'),
 ('Temefos', 'D02'),
 ('Temefos', 'D02')]

Command took 0.55 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- check the count after removing header count

```
1  header = MESH_Split.first()
2  RDD21_count1 = MESH_Split.filter(lambda r :r!=header)
3  RDD21_count1.count()

▸ (2) Spark Jobs

Out[63]: 124284

Command took 1.45 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- After removing the header check the result.

```
1  #removed header count and split the data and check the five results
2  MESH_Header = MESH.first()
3  MESH_Header_Remove = MESH.filter(lambda Header: Header!= MESH_Header)
4  MESH_Split = MESH_Header_Remove.map(lambda y:(y.split(",")[0], y.split(",")[1].split(".")[0]))
5  MESH_Split.take(5)
```

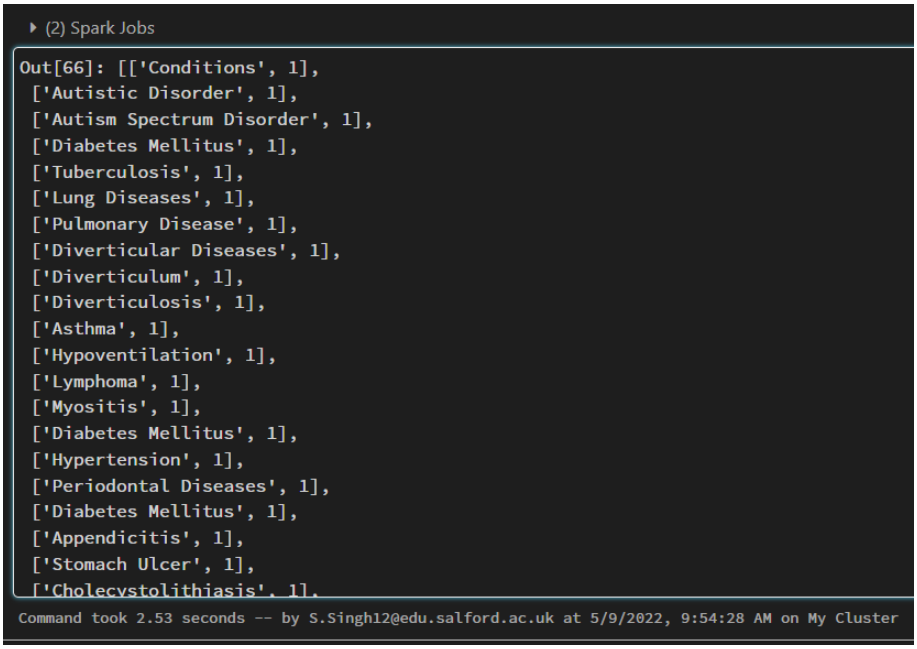
```
▸ (2) Spark Jobs

Out[64]: [('Calcimycin', 'D03'),
 ('A-23187', 'D03'),
 ('Temefos', 'D02'),
 ('Temefos', 'D02'),
 ('Temefos', 'D02')]

Command took 1.25 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- Check ten outcomes using this command.

```
1  MESH_Split.take(10)

▸ (1) Spark Jobs

Out[65]: [('Calcimycin', 'D03'),
 ('A-23187', 'D03'),
 ('Temefos', 'D02'),
 ('Temefos', 'D02'),
 ('Temefos', 'D02'),
 ('Abate', 'D02'),
 ('Abate', 'D02'),
 ('Abate', 'D02'),
 ('Difos', 'D02'),
 ('Difos', 'D02')]

Command took 0.67 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

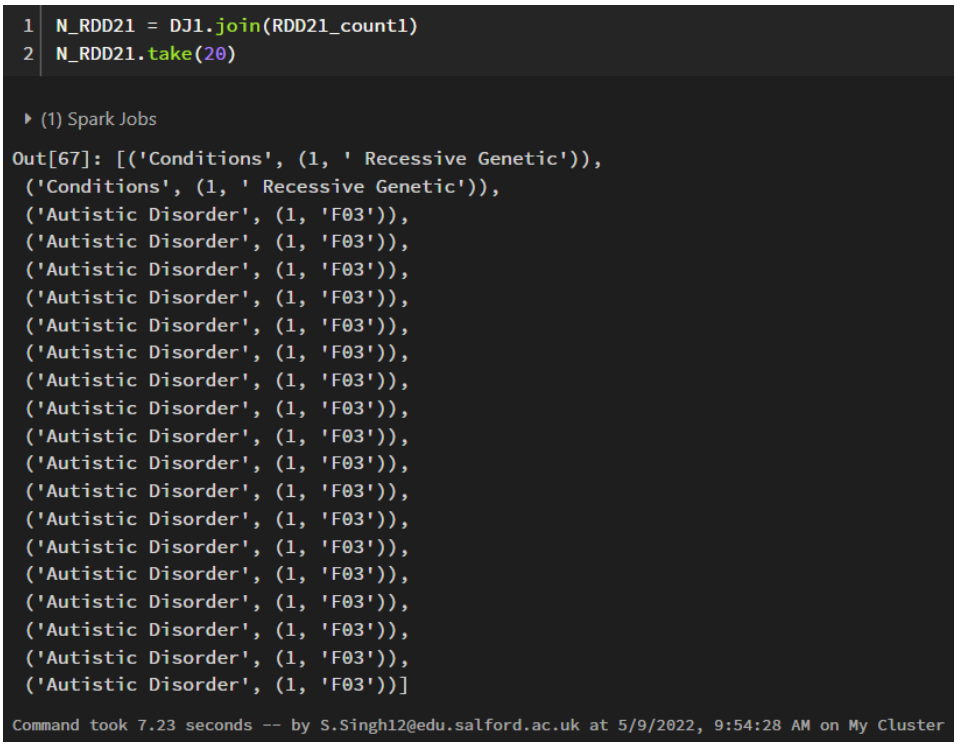
- Looking at the client's requirement, I have analysed that we need to combine two files. For that first, sort the data as per the requirement.

```
New2 = RDD21.map(lambda c: c.split('|'))
head =New2.first()
NEW1 = New2.filter(lambda r:  r != head)
DJ1 = New2.map(lambda c: c[7]).flatMap(lambda y : y.split(",")).filter(lambda y: y != "").map(lambda y:[y,1])
DJ1.take(100)
```



```
▸ (2) Spark Jobs
Out[66]: [['Conditions', 1],
 ['Autistic Disorder', 1],
 ['Autism Spectrum Disorder', 1],
 ['Diabetes Mellitus', 1],
 ['Tuberculosis', 1],
 ['Lung Diseases', 1],
 ['Pulmonary Disease', 1],
 ['Diverticular Diseases', 1],
 ['Diverticulum', 1],
 ['Diverticulosis', 1],
 ['Asthma', 1],
 ['Hypoventilation', 1],
 ['Lymphoma', 1],
 ['Myositis', 1],
 ['Diabetes Mellitus', 1],
 ['Hypertension', 1],
 ['Periodontal Diseases', 1],
 ['Diabetes Mellitus', 1],
 ['Appendicitis', 1],
 ['Stomach Ulcer', 1],
 ['Cholecystolithiasis', 1],
Command took 2.53 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- Then I combined the files using Joins.



```
N_RDD21 = DJ1.join(RDD21_count1)
N_RDD21.take(20)

▸ (1) Spark Jobs
Out[67]: [('Conditions', (1, ' Recessive Genetic')),
 ('Conditions', (1, ' Recessive Genetic')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03')),
 ('Autistic Disorder', (1, 'F03'))]
Command took 7.23 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- To clarify, suppose your clinical trial data was:
  NCT01, ...,"Disease_A,Disease_B",
  NCT02, ..., Disease_B,
  And the mesh file contained:
  Disease_A A01.01 C23.02
  Disease_B B01.34.56
  The result would be
  B01 2
  A01 1
  C23 1

The above similar requirement needed by the client. So, this the next command I have applied. That give the following result.

```
#Join two tables for required result
N_RDD2021 = DJ1.join(MESH_Split).map(lambda a: (a[1][1], 1)).reduceByKey(lambda v1, v2: v1+ v2).sortBy(lambda y: -y[1])
N_RDD2021.take(5)
```

```
▸ (3) Spark Jobs
Out[68]: [('C04', 143994),
 ('C23', 136079),
 ('C01', 106674),
 ('C14', 94523),
 ('C10', 92310)]
Command took 9.71 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

HIVE Implementation:

- Here I have created a table ‘mesh’ with the required fields and load the data into it and observe the result.

```
CREATE External TABLE if not exists mesh
                    (TERM STRING, TREE STRING)
USING CSV
OPTIONS (path "dbfs:/FileStore/tables/mesh.csv",
        header "true");
SELECT * FROM mesh;
```

▸ (1) Spark Jobs

	TERM	TREE
1	Calcimycin	D03.633.100.221.173
2	A-23187	D03.633.100.221.173
3	Temefos	D02.705.400.625.800
4	Temefos	D02.705.539.345.800
5	Temefos	D02.886.300.692.800
6	Abate	D02.705.400.625.800
7	Abate	D02.705.539.345.800

Truncated results, showing first 1000 rows.

- Again, I created a new table named ‘SP_STRING’ to put some specific values in particular column.

```
1  DROP TABLE IF EXISTS SP_STRING;
2  CREATE OR REPLACE TABLE SP_STRING(
3  SELECT TERM,TREE,SUBSTRING(TREE,1,3) AS NEW1
4  FROM mesh);
```

▸ (7) Spark Jobs

Query returned no results

- Check the result for the above query. ‘WHERE’ clause used to remove the duplicity.

```
1  SELECT * FROM SP_STRING
2  WHERE TERM <> 'term';
```

▸ (1) Spark Jobs

	TERM	TREE	NEW1
1	Calcimycin	D03.633.100.221.173	D03
2	A-23187	D03.633.100.221.173	D03
3	Temefos	D02.705.400.625.800	D02
4	Temefos	D02.705.539.345.800	D02
5	Temefos	D02.886.300.692.800	D02
6	Abate	D02.705.400.625.800	D02
7	Abate	D02.705.539.345.800	D02

Truncated results, showing first 1000 rows.

Command took 1.35 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 10:19:00 PM on My Cluster

- According to the client’s requirement, we need to join the two tables. For that I have created another table ‘SP_EX’ including required columns. Used ‘Lateral View’ with explode to list the array.

```
1  DROP TABLE IF EXISTS SP_EX;
2  CREATE OR REPLACE TABLE SP_EX(SELECT Id,Conditions from clinicaltrial_2021
3  lateral view explode(split(Conditions,',')) Conditions AS NEW);
4  SELECT * FROM SP_EX;
```

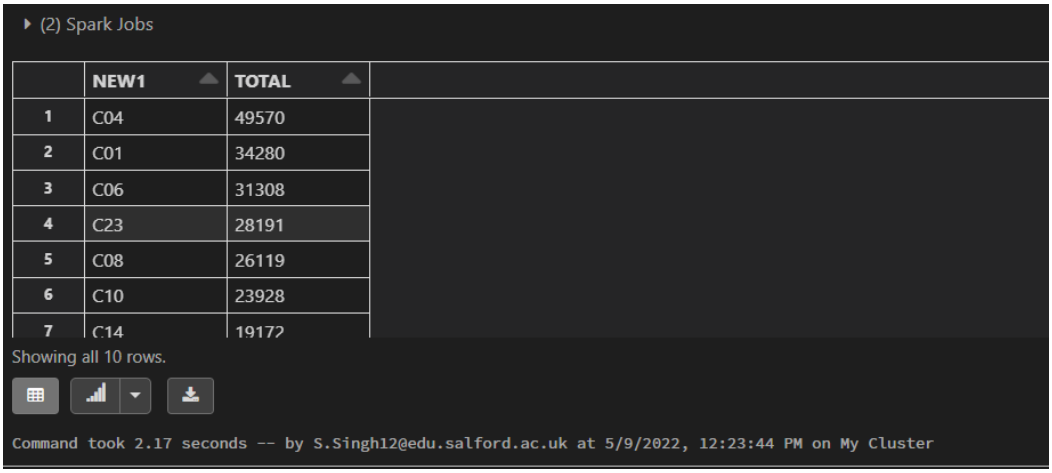
▸ (5) Spark Jobs

	Id	Conditions
1	Id	Conditions
2	NCT02751957	Autistic Disorder,Autism Spectrum Disorder
3	NCT02751957	Autistic Disorder,Autism Spectrum Disorder
4	NCT02758483	Diabetes Mellitus
5	NCT02759848	Tuberculosis,Lung Diseases,Pulmonary Disease
6	NCT02759848	Tuberculosis,Lung Diseases,Pulmonary Disease
7	NCT02759848	Tuberculosis,Lung Diseases,Pulmonary Disease

Truncated results, showing first 1000 rows.

- With this query I am trying to fetch the data, while using join for lookup values.

```
SELECT SP_STRING.NEW1,COUNT(SP_STRING.NEW1) AS TOTAL
From SP_STRING
INNER JOIN SP_EX ON SP_EX.Conditions = SP_STRING.TERM
GROUP BY SP_STRING.NEW1
ORDER BY TOTAL DESC
LIMIT 10
```



As per the requirement of client for this problem statement, I have created the table 'mesh' and import the data from 'mesh.csv' file. When I identify the result of csv file then I got to know that I need to segregate the data on basis of delimiter ',' so I applied the same. According to the problem statement, I need to merge the data of columns named 'conditions' and 'Term' from two different tables by creating the relationship. At the end, I clubbed the data on the basis of their counts and created the summarization report. However, in HiveQL the result shows the data discrepancies. Even after floating the data in the same process as PySpark.

---

5. Find the 10 most common sponsors that are not pharmaceutical companies, along with the number of clinical trials they have sponsored. *Hint:* For a basic implementation, you can assume that the *Parent Company* column contains all possible pharmaceutical companies.

   Assumption made: Basically, here, we are looking for the results that are connected through two tables. So, first I need to refine the files and lookup for the desired fields and then join them to make desirable result.

RDD Implementations:

- This script provides the data of two field and that has been split by delimiter '|'.

```
#Check first result after splitting
splitRDD21 = RDD21.map(lambda x:x.split('|'))
JAN = splitRDD21.map(lambda l:l[1])
JAN.take(2)
```

```
▸ (1) Spark Jobs
Out[32]: ['Sponsor', 'The University of Hong Kong']
```

- In this requirement, I am working on new table and sort table. Here, I check the result for a row.

```
#Check the result of csv file after splitting the values with delimiter
PHARMA = sc.textFile("/FileStore/tables/pharma.csv")
N_PHARMA = PHARMA.map(lambda x: x.replace('"','').split(','))
N_PHARMA.take(2)
```

```
▸ (1) Spark Jobs
Out[72]: [['Company',
  'Parent_Company',
  'Penalty_Amount',
  'Subtraction_From_Penalty',
  'Penalty_Amount_Adjusted_For_Eliminating_Multiple_Counting',
  'Penalty_Year',
  'Penalty_Date',
  'Offense_Group',
  'Primary_Offense',
  'Secondary_Offense',
  'Description',
  'Level_of_Government',
  'Action_Type',
  'Agency',
  'Civil/Criminal',
  'Prosecution_Agreement',
  'Court',
  'Case_ID',
  'Private_Litigation_Case_Title',
  'Lawsuit_Resolution',
  'Facility_State',
Command took 0.52 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- Just for the verification, I have used this script that shows the value of ‘Parent_Company’ field.

```
1  PHARMA21 = N_PHARMA.map(lambda l:l[1])
2  PHARMA21.take(2)
```

▸ (1) Spark Jobs

```
Out[73]: ['Parent_Company', 'Abbott Laboratories']
```

Command took 0.32 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster

- After analysing, according to the client’s further needs, I have to work on two tables that ‘Clinicaltrail_2021’ and ‘Pharma’ which helps me to give the desired result. So firstly, I implemented as follows.

```
1  sponsors = RDD21.map(lambda c: c.split('|'))
2  head = sponsors.first()
3  sponsors = sponsors.filter(lambda row:  row != head)
4  SPONSORS = sponsors.map(lambda x: x[1]).map(lambda y: (y, 1))
```

▸ (1) Spark Jobs

Command took 1.17 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster

- Same script for the ‘Pharma’ table as I need it to join the two columns.

```
1  PHARMA = sc.textFile('dbfs:/FileStore/tables/pharma.csv')
2  PHARMAhead = PHARMA.first()
3  PHARMA = PHARMA.filter(lambda row:  row != PHARMAhead)
4  Pharmacetical = PHARMA.map(lambda a: a.split(',')).map(lambda b: (b[1].replace('"', ''))).map(lambda c: (c,1))
```

▸ (1) Spark Jobs

Command took 0.36 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster

- So, this the final requirement which shows the result of 10 common Sponsors that are not pharmaceutical companies. Here this is done by the joining two files.

```
#Merge two files to get resultant value
PHARMA_21 = SPONSORS.leftOuterJoin(Pharmacetical).filter(lambda i : i[1][1] == None).map(lambda j: (j[0], 1)).reduceByKey(lambda k1, k2: k1+ k2).sortBy(lambda l: -l[1])
PHARMA_21.take(10)
```

```
▸ (3) Spark Jobs
Out[76]: [('National Cancer Institute (NCI)', 3218),
 ('M.D. Anderson Cancer Center', 2414),
 ('Assistance Publique - Hôpitaux de Paris', 2369),
 ('Mayo Clinic', 2300),
 ('Merck Sharp & Dohme Corp.', 2243),
 ('Assiut University', 2154),
 ('Novartis Pharmaceuticals', 2088),
 ('Massachusetts General Hospital', 1971),
 ('Cairo University', 1928),
 ('Hoffmann-La Roche', 1828)]
Command took 6.64 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

The result gives a clear description of the companies from maximum frequencies to lowest.

HIVEQL Implementations:

- In HIVEQL, first, I created a table named 'PHARMA' with this script which includes the following fields from the following path that added the data and check the result for the first five rows.

```
DROP TABLE IF EXISTS PHARMA;
CREATE External TABLE IF NOT EXISTS PHARMA(Company STRING,Parent_Company STRING,
Penalty_Amount STRING, Subtraction_From_Penalty STRING,
Penalty_Amount_Adjusted_For_Eliminating_Multiple_Counting STRING,
Penalty_Year STRING, Penalty_Date STRING, Offense_Group STRING,
Primary_Offense STRING, Secondary_Offense STRING, Description STRING,
Level_of_Government STRING, Action_Type STRING,
Agency STRING, Civil_Criminal STRING,
Prosecution_Agreement STRING,Court STRING,
Case_ID STRING,Private_Litigation_Case_Title STRING,
Lawsuit_Resolution STRING,Facility_State STRING,
City STRING,Address STRING,Zip STRING,
NAICS_Code STRING,NAICS_Translation STRING,
HQ_Country_of_Parent STRING,HQ_State_of_Parent STRING,
Ownership_Structure STRING,
Parent_Company_Stock_Ticker STRING,
Major_Industry_of_Parent STRING,
Specific_Industry_of_Parent STRING,
Info_Source STRING,
Notes STRING
) USING CSV
OPTIONS (path "dbfs:/FileStore/tables/pharma.csv",
        delimiter ",",
        header "true")
        ;
SELECT * FROM PHARMA
LIMIt 5;
```

▸ (1) Spark Jobs

	Company	Parent_Company	Penalty_Amount	Subtraction_From_Penalty	Penalty_Amount_Adjusted_For_Eliminating_Multiple_Counting	Penalty_Year	Penalty_Da
1	Abbott Laboratories	Abbott Laboratories	$5,475,000	$0	$5,475,000	2013	20131227
2	Abbott Laboratories Inc.	AbbVie	$1,500,000,000	$0	$1,500,000,000	2012	20120507
3	Abbott Laboratories Inc.	AbbVie	$126,500,000	$0	$126,500,000	2010	20101207
4	Abbott Laboratories Puerto Rico, Inc.	Abbott Laboratories	$49,045	$0	$49,045	2009	20090305
5	Acclarent Inc.	Johnson & Johnson	$18,000,000	$0	$18,000,000	2016	20160722

Showing all 5 rows.

Command took 2.01 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 1:59:35 PM on My Cluster

- As per the requirement, I need to create two tables to fetch the desired fields from two different tables. So, I have created first table 'PHARMAS' having filed 'Parent_Company'

```
1  DROP TABLE IF EXISTS PHARMAS;
2  CREATE EXTERNAL TABLE IF NOT EXISTS PHARMAS(Parent_Company STRING)
3  USING CSV
4  LOCATION 'dbfs:/FileStore/tables/pharma.csv';
5  SELECT * FROM PHARMAS
```

▸ (1) Spark Jobs

	Parent_Company
1	Company
2	Abbott Laboratories
3	Abbott Laboratories Inc.
4	Abbott Laboratories Inc.
5	Abbott Laboratories Puerto Rico, Inc.
6	Acclarent Inc.
7	Advanced Medical Optics

Showing all 969 rows.

Command took 1.40 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 2:08:55 PM on My Cluster

- Create another table named 'NEW' and select field 'Sponsor'.

```
1  DROP TABLE IF EXISTS NEW;
2  CREATE OR REPLACE TABLE NEW(SELECT Sponsor FROM clinicaltrial_2021);
3  SELECT * FROM NEW
```

(5) Spark Jobs

	Sponsor
1	Sponsor
2	The University of Hong Kong
3	Duke University
4	Universidade Federal do Rio de Janeiro
5	Istanbul Medeniyet University
6	University of Roma La Sapienza
7	Consorzio Futuro in Ricerca

Truncated results, showing first 1000 rows.

Command took 11.38 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 2:24:34 PM on My Cluster

- The desired results are as follows for which I have joined these above tables to make the result as required.

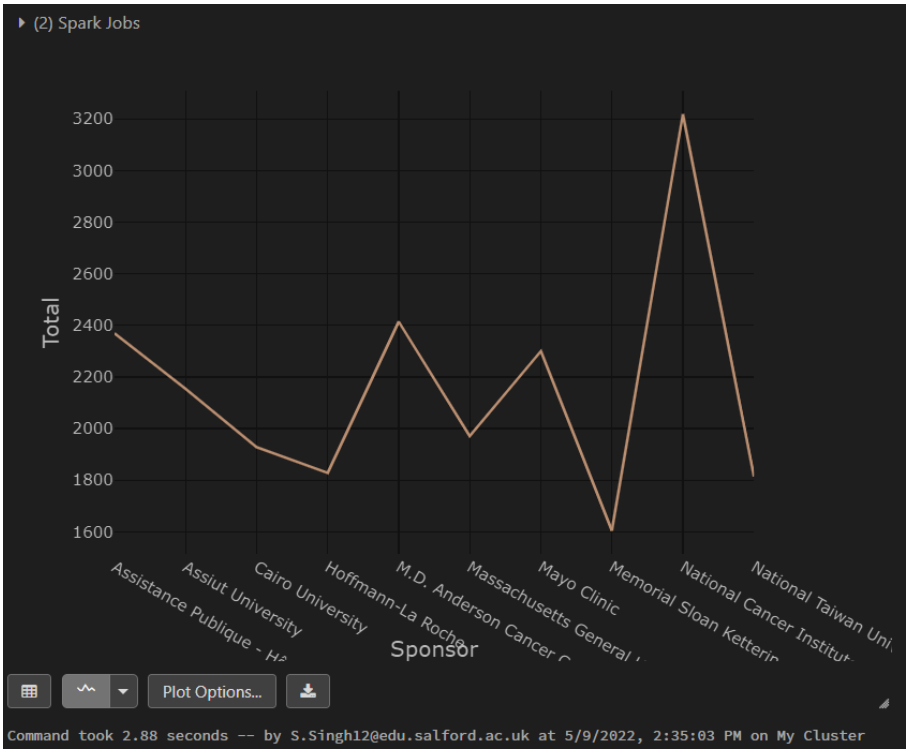
```
1  SELECT Sponsor, COUNT(Sponsor) AS Total
2  From NEW
3  Left Join PHARMAS ON NEW.Sponsor = PHARMAS.Parent_company
4  WHERE Parent_Company IS NULL
5  Group By Sponsor
6  ORDER BY TOTAL DESC
7  LIMIT 10
```

(2) Spark Jobs

	Sponsor	Total
1	National Cancer Institute (NCI)	3218
2	M.D. Anderson Cancer Center	2414
3	Assistance Publique - Hôpitaux de Paris	2369
4	Mayo Clinic	2300
5	Assiut University	2154
6	Massachusetts General Hospital	1971
7	Cairo University	1928

Showing all 10 rows.

Command took 2.88 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 2:35:03 PM on My Cluster



The line graph for the above script clearly shows the different frequencies for each company.

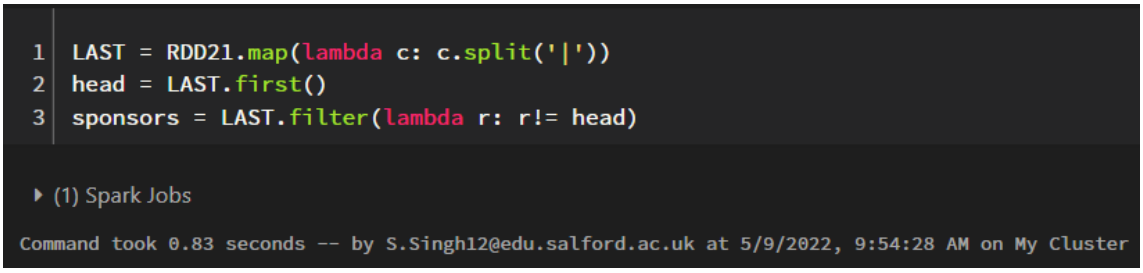
Result: While sorted in order, the frequencies for different non-pharmaceutical companies are fluctuated. Maximum hike of frequency goes up to 3218. Other than this are quite low in numbers.

6. Plot number of completed studies each month in a given year – for the submission dataset, the year is 2021. You need to include your visualization as well as a table of all the values you have plotted for each month.

   Assumption made: In the dataset, completed studies includes the other year as well. But we are looking for the 2021 dataset. To collect the studies of 2021, I tried to fetch the data of 2021 and then find the number of values for that.

   RDD Implementations:

   - As per this script, use string split function to segregate the data from the file and first row is similar with header so that row has been filter out with the command.



- In this code, the data of 2021 has been collected those shows completed status.

```
#Filtered out data of completed status of year 2021
completed = LAST.filter(lambda g:g[2]=='Completed').map(lambda h:h[4].split(' ')).filter(lambda i:i[0]!='').filter(lambda j:j[1]=='2021').map(lambda f:(f[0],1)).reduceByKey(lambda k1,k2:k1+k2).sortBy(lambda z:z[0])
completed.collect()
```

```
▸ (3) Spark Jobs
Out[78]: [('Apr', 967),
 ('Aug', 700),
 ('Feb', 934),
 ('Jan', 1131),
 ('Jul', 819),
 ('Jun', 1094),
 ('Mar', 1227),
 ('May', 984),
 ('Oct', 187),
 ('Sep', 528)]
Command took 2.91 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

- While importing the calendar, the data is sorted as per the month's order.

```
import calendar
x = {a:z for z,a in enumerate(calendar.month_abbr[1:],1)}
N_RDD21=completed.sortBy(lambda a: x.get(a[0]))
N_RDD21.collect()
```

```
▸ (3) Spark Jobs
Out[79]: [('Jan', 1131),
 ('Feb', 934),
 ('Mar', 1227),
 ('Apr', 967),
 ('May', 984),
 ('Jun', 1094),
 ('Jul', 819),
 ('Aug', 700),
 ('Sep', 528),
 ('Oct', 187)]
Command took 0.60 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 9:54:28 AM on My Cluster
```

HIVEQL Implementation:

- Created a new table ‘ST_COM’ to split the data into different fields as per their ‘STATUS’ and identify the result on the basis of given table.

```
DROP TABLE IF EXISTS ST_COM;
CREATE OR REPLACE TABLE ST_COM
(SELECT SPLIT(Completion,' ')[0] as MONTH ,SPLIT(Completion,' ')[1] as YEAR,STATUS
FROM TEX);
Select * from ST_COM
```

(5) Spark Jobs

	MONTH	YEAR	STATUS
1	Completion	null	Status
2	Nov	2021	Recruiting
3	Jul	2020	Completed
4	Jan	2018	Completed
5	Dec	2014	Completed
6	Sep	2020	Active, not recruiting
7	Jan	2018	Completed

Truncated results, showing first 1000 rows.

Command took 11.47 seconds -- by S.Singh12@edu.salford.ac.uk at 5/9/2022, 2:39:50 PM on My Cluster

- In these codes, I have cleaned the data for the final requirement.

```
SELECT Month,COUNT(MONTH)AS TOTAL
FROM ST_COM
WHERE YEAR == 2021 AND STATUS == 'Completed'
GROUP BY Month
```

(2) Spark Jobs

	Month	TOTAL
1	Oct	187
2	Sep	528
3	Aug	700
4	May	984
5	Jun	1094
6	Feb	934
7	Mar	1227

Showing all 10 rows.

- Finally, the result has been sorted Month wise in which I used the ‘Case’ that helps to make final result sorted.

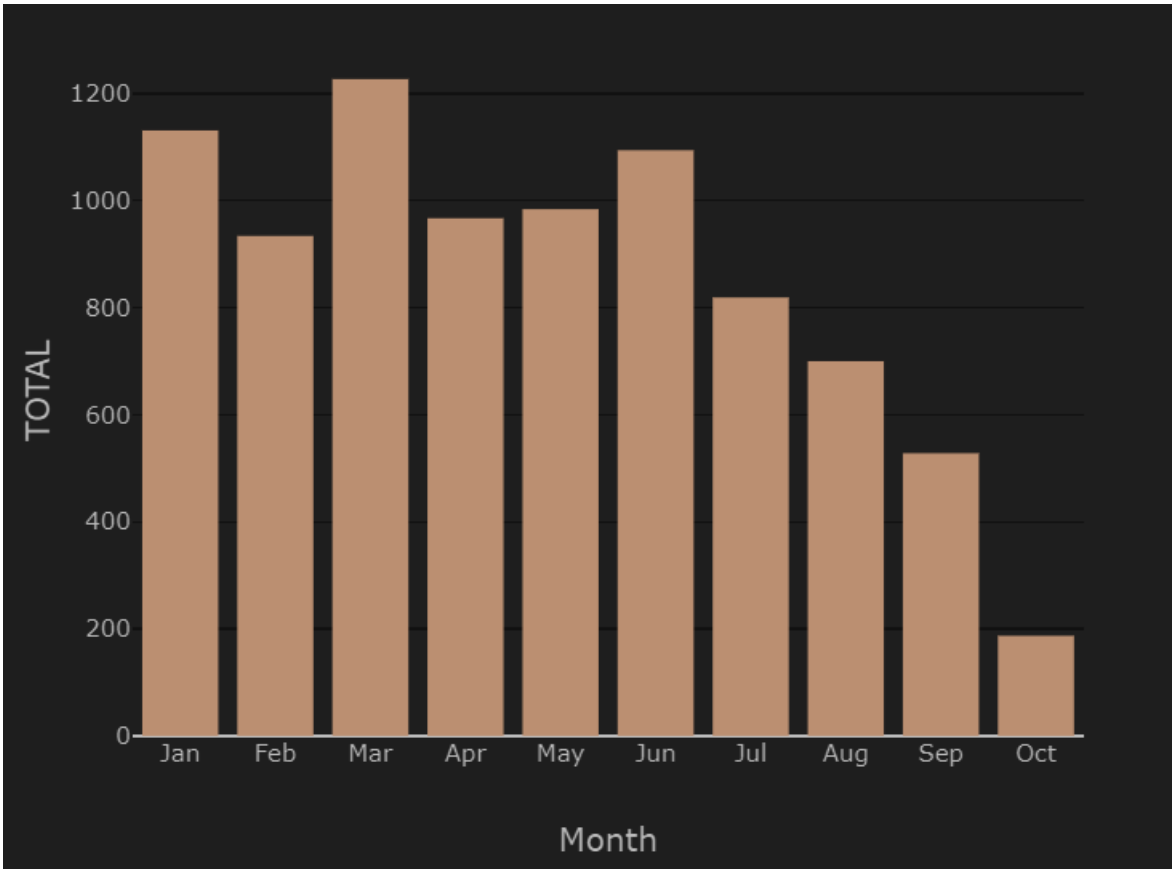
```
SELECT Month,COUNT(MONTH)AS TOTAL
FROM ST_COM
WHERE YEAR == 2021 AND STATUS == 'Completed'
GROUP BY Month
order by CASE
           Month WHEN 'Jan' THEN '01'
                 WHEN 'Feb' THEN '02'
                 WHEN 'Mar' THEN '03'
                 WHEN 'Apr' THEN '04'
                 WHEN 'May' THEN '05'
                 WHEN 'Jun' THEN '06'
                 WHEN 'Jul' THEN '07'
                 WHEN 'Aug' THEN '08'
                 WHEN 'Sep' THEN '09'
                 WHEN 'Oct' THEN '10'
                 WHEN 'Nov' THEN '11'
                 ELSE '12'
                 END
```

▸ (2) Spark Jobs

	Month	TOTAL
1	Jan	1131
2	Feb	934
3	Mar	1227
4	Apr	967
5	May	984
6	Jun	1094
7	Jul	819

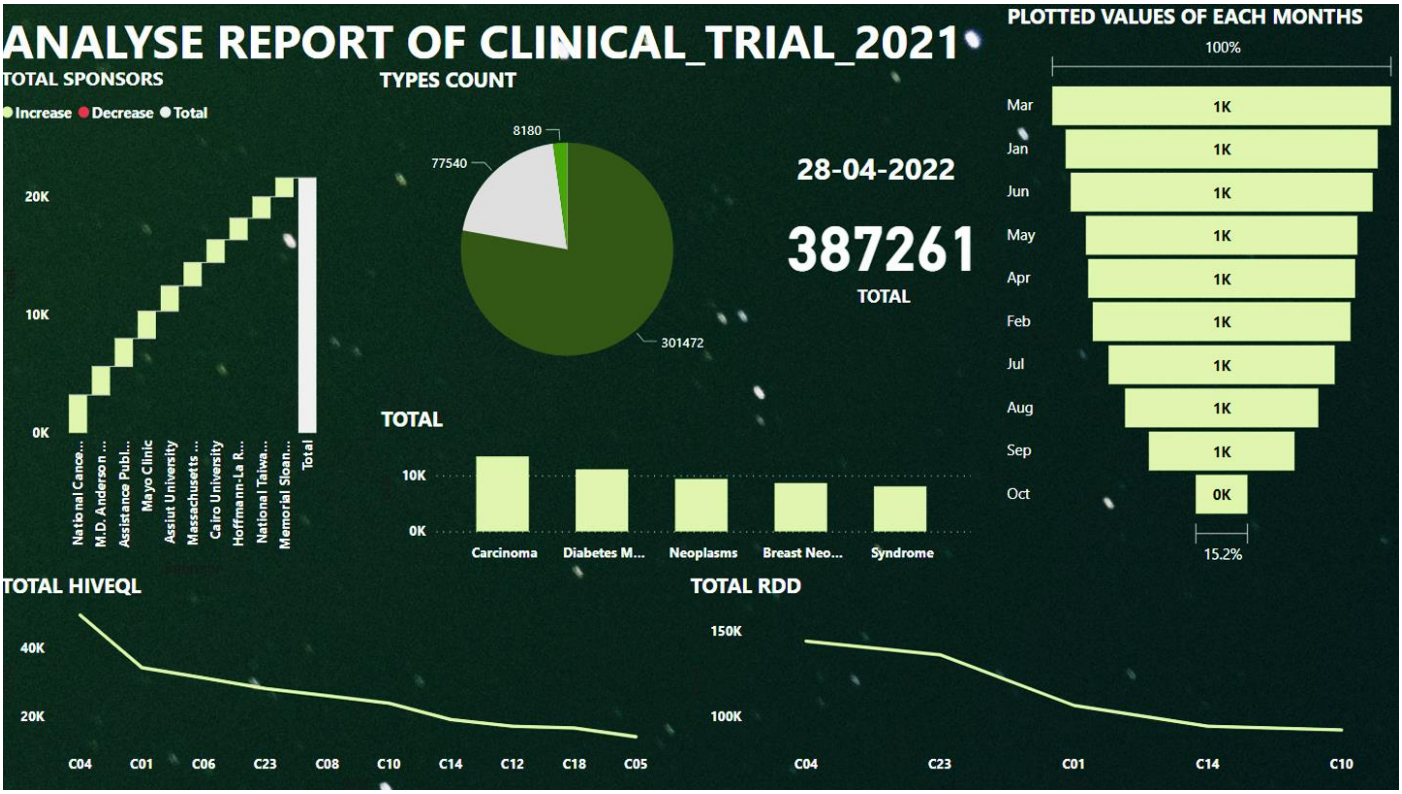
Showing all 10 rows.

Command took 1.38 seconds -- by S.Singh12@edu.salford.ac.uk at 5/12/2022,



Result: The raw table above gives the value month wise and clearly be seen in Bar graph. The studies are fluctuating from January to June but after that it goes down till October.

POWER BI DASHBOARD SNIPET



All problem Statement has been examined in form of visualization using Card, Line, Bar, Funnel and Waterfall Graph in 'POWER BI'.

1. The card shows the total numbers of studies in the dataset.
2. Pie describes the counts of different types of 'Type'.
3. The bar graph illustrates the counts of distinct top 5 conditions.
4. The line graph, differentiate the total codes in HiveQL and PySpark.
5. The waterfall explains 10 most common Sponsors.
6. Final, Funnel graph gives information about completed studies during each month of 2021.

Summary

During the working on this project, I have learnt various data cleansing techniques using PySpark and HiveQL. I also dig out the raw data and get the resultant as meaningful insights. So that if somebody wants to read the insights they can directly read from the visualizations. I had been trying to make the SOP (statement of purpose) and make the Set of Instructions. Following that, this process shoed be followed in the similar kind of data and it will reduce the repetitive work. The matrix can be read on that similar kind of files while using this set of instruction.

As we have followed the ETL (Extract Transform and Load) process, in the initial step of ETL, I have imported the non-identical data in PySpark and HiveQL. In the intermediate level, the data transformation has been made using different functionality and pull out the rightful information as per the requirements and trying to achieve the best informative results. All the tables and graphs give the clear visualization of data.