# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

## A) HADOOP

Step1: Download the source file from CMS website (Center for Medical Services) where it has a dataset of all the insurance plan for throughout years.

For our Analysis, we took 2017 to 2021 dataset to compare and identify which insurance plan would be better for individual/family/employee.

The files to download are as follows:

1. Benefits and Cost Sharing PUF
2. Rate PUF
3. Plan Attributes PUF
4. Business Rules PUF
5. Service Area PUF
6. Network
7. Machine-readable URL PUF
8. Transparency in Coverage PUF

https://www.cms.gov/cciio/resources/data-resources/marketplace-puf

Step2: Upload the files in linux system:

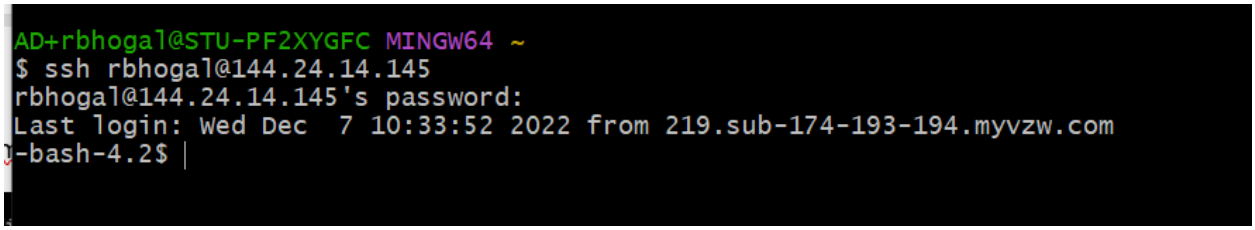
scp "filepath\file.csv" username@serveradd:/tmp


Step2: Transfer the file from linux to hdfs

**Log in to your hdfs server:**

Ssh username@serveradd

username@serveradd password:




cd /tmp

# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

ls /tmp (to view your uploaded files)

```
bash-4.2$ cd
bash-4.2$ pwd
home/rbhogal
bash-4.2$ cd /tmp
bash-4.2$ ls /tmp
```

```
machine_readble_url.csv
output
pymp-0o47ex
pymp-BIvM8m
pymp-cIeC9J
pymp-NLojTz
pymp-qCZG1h
pymp-Xzfkeo
report.json
snappy-1.0.5-5fd1aeda-b749-4081-b289-22c117049a00-libsnappyjava.so
systemd-private-3207b8db87a24ac9b7ceb8bf19353586-chronyd.service-W5XMXn
systemd-private-3207b8db87a24ac9b7ceb8bf19353586-cups.service-rmMLWK
systemd-private-3207b8db87a24ac9b7ceb8bf19353586-unified-monitoring-agent.s
```

**Make directory in hdfs**

hdfs dfs -mkdir Project/ ---(Project is main folder name)

hdfs dfs -mkdir Project/file --(file is subfolder)

hdfs dfs -put file.csv Project/file/

**To view folders and files:**

Hdfs dfs -ls

```
-bash-4.2$ hdfs dfs -ls
Found 3 items
drwx------   - rbhogal hdfs          0 2022-12-07 11:02 .Trash
drwxr-xr-x   - rbhogal hdfs          0 2022-11-10 02:35 .hiveJars
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 21:02 Health
```

Hdfs dfs -ls Health/

```
-bash-4.2$ hdfs dfs -ls Health/
Found 8 items
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:53 Health/benefits_cost_sha
ring
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:56 Health/business_rules
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:52 Health/machine_readble_u
rl
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:11 Health/network
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:18 Health/plan_attributes
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:32 Health/rate
drwxr-xr-x   - rbhogal hdfs          0 2022-11-29 22:19 Health/service_area
drwxr-xr-x   - rbhogal hdfs          0 2022-11-30 00:18 Health/transparency_in_c
overage
```

Step3: Once the files are copied from linux to hdfs, remove the files in linux :

rm -file.csv

# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

```
-bash-4.2$ rm machine_readble_url.csv
-bash-4.2$ ls /tmp
03d7e569-d376-4a64-b9c4-bbb8cca6cd35_resources
2417d95f-6634-4b31-87ca-e0ccfcdcd158_resources
2ea2bbcf-333c-45a5-a22a-04c6b9316ebf_resources
4bf07050-93e4-4012-a977-50b233a28eb6_resources
4bf2b422-c146-4140-92dd-377cb5e7a860_resources
76c1f38e-0efe-4571-99a4-f7d1d3460267_resources
```

Step4: Connect to Hive using command:

beeline;

```
-bash-4.2$ beeline;
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/odh/1.1.2/hive/lib/log4j-slf4j-impl-2.17.
1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/odh/1.1.2/hadoop/lib/slf4j-log4j12-1.7.25
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://bigdaiwn0.sub02180640120.trainingvcn.oraclevcn.com:21
81,bigdaimn0.sub02180640120.trainingvcn.oraclevcn.com:2181,bigdaiun0.sub02180640
120.trainingvcn.oraclevcn.com:2181/default;password=rbhogal;serviceDiscoveryMode
=zooKeeper;user=rbhogal;zooKeeperNamespace=hiveserver2
22/12/07 21:15:41 [main-EventThread]: ERROR imps.EnsembleTracker: Invalid config
 event received: {server.1=bigdaimn0.sub02180640120.trainingvcn.oraclevcn.com:28
88:3888:participant, version=0, server.3=bigdaiwn0.sub02180640120.trainingvcn.or
aclevcn.com:2888:3888:participant, server.2=bigdaiun0.sub02180640120.trainingvcn
```

# B) HIVE

Step5: use your database

use username;

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> use rbhogal;
INFO  : Compiling command(queryId=hive_20221207211901_1c895217-ebe8-4049-86d3-134e6dcf608b): use rbhogal
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20221207211901_1c895217-ebe8-4049-86d3-134e6dcf608b); Time taken: 0.025 s
econds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20221207211901_1c895217-ebe8-4049-86d3-134e6dcf608b): use rbhogal
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20221207211901_1c895217-ebe8-4049-86d3-134e6dcf608b); Time taken: 0.214 s
econds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
No rows affected (0.291 seconds)
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai>
```

Step6: See the existing tables

show tables;

```
0: jdbc:hive2://bigdaiwn0.sub02180640120.trai> show tables;
INFO  : Compiling command(queryId=hive_20221207211935_a5bf16a0-da8e-4d66-8a86-75eee62e2641): show tables
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserial
INFO  : Completed compiling command(queryId=hive_20221207211935_a5bf16a0-da8e-4d66-8a86-75eee62e2641); Time taken:
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20221207211935_a5bf16a0-da8e-4d66-8a86-75eee62e2641): show tables
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20221207211935_a5bf16a0-da8e-4d66-8a86-75eee62e2641); Time taken:
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+---------------------------+
|         tab_name          |
+---------------------------+
| a                         |
| b1                        |
| bb                        |
| benefits                  |
| benefits_cost_sharing     |
| br                        |
```

# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

Step7: Create new tables for all the csv files uploaded in hdfs.

Note: Since the file path is hdfs, we need to retain the Project folder as data gets populated from the hdfs to hive tables.

CREATE EXTERNAL TABLE if not exists table_name

(col1 datatype,

col2 datatype,

col3 datatype,

.

.

.

colN datatype

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

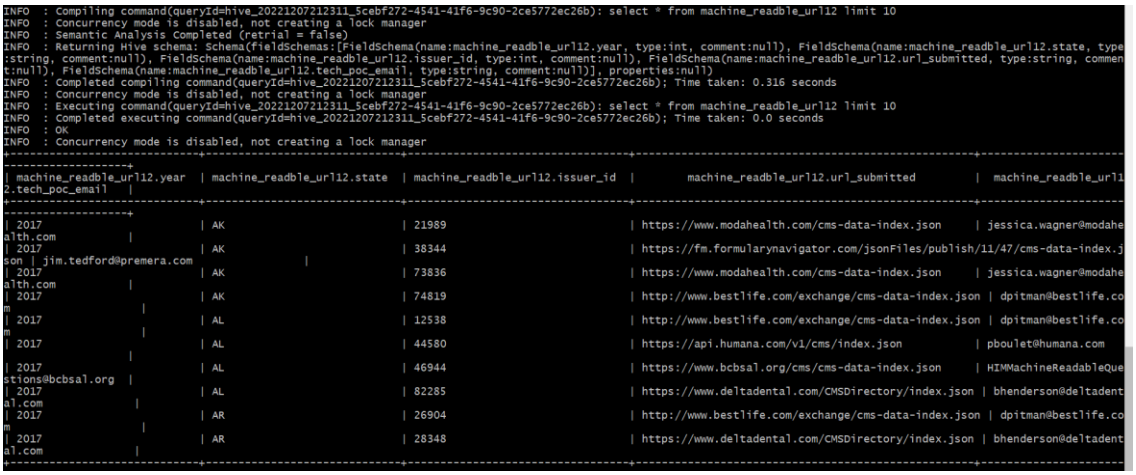
STORED AS TEXTFILE LOCATION '/user/username/Project/file/'  --file path of hdfs folder

TBLPROPERTIES ('skip.header.line.count'='1');


See the table contents

# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

Select * from table_name limit 10; Select * from machine_readble_url limit 10;




Step8: Once all the tables are created, we need to create summary tables

1) using joins

2) using group by clause

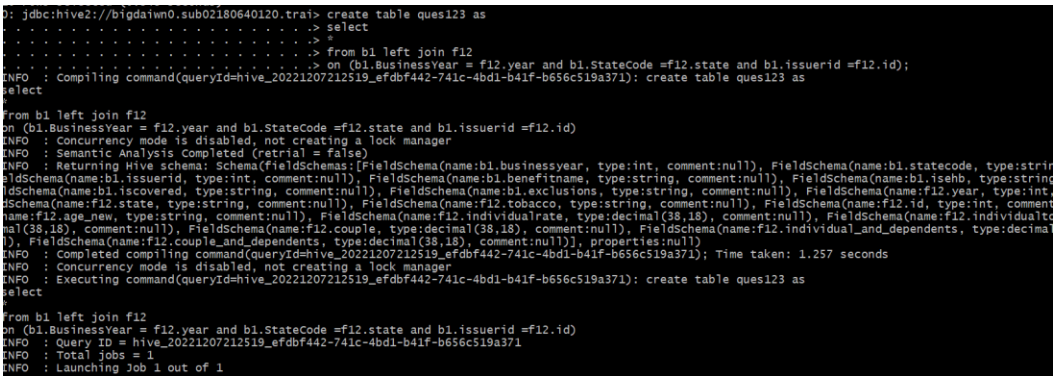
3) conditions (where clause, having clause ,case when or nested if-else statements, regex_expression)

ex:

create table ques123 as

select * from b1 left join f12

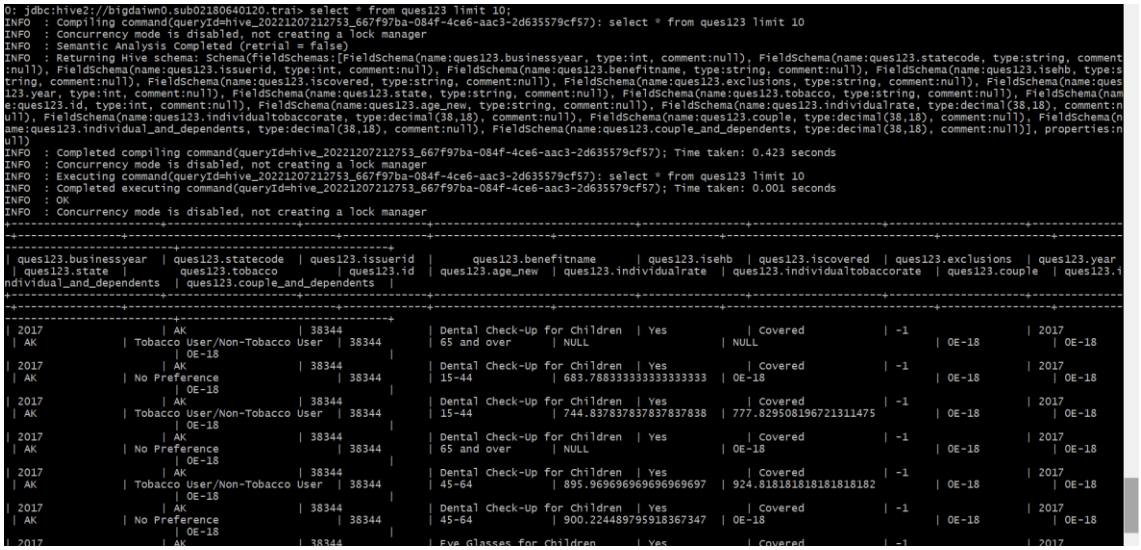
on (b1.BusinessYear = f12.year and b1.StateCode =f12.state and b1.issuerid =f12.id);




Step9: View the tables to check whether we are getting proper values or not.

select * from ques123 limit 10;

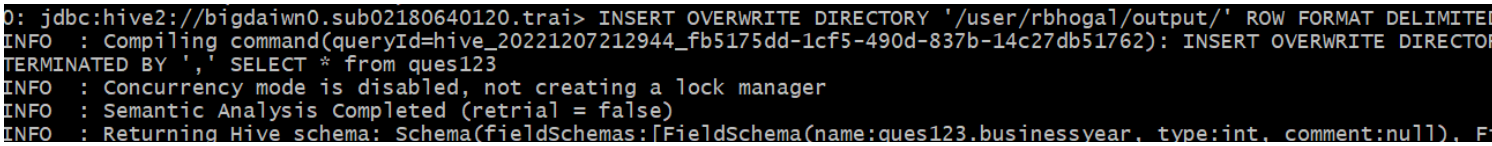
# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)




Step10: To download the summary files from hive into the hdfs system,

INSERT OVERWRITE DIRECTORY '/user/username/output/' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT * from ques123;

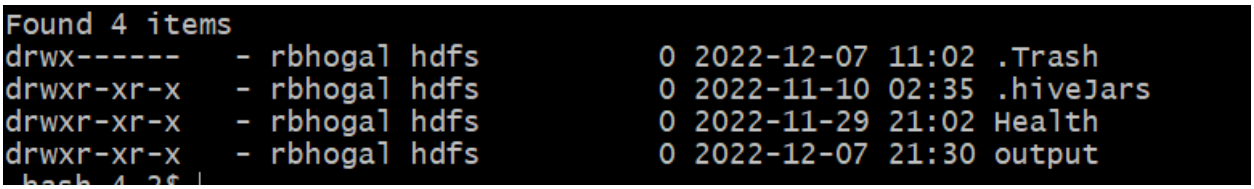
(Note: in your hdfs, folder of output would be created with summary table ques123 data)

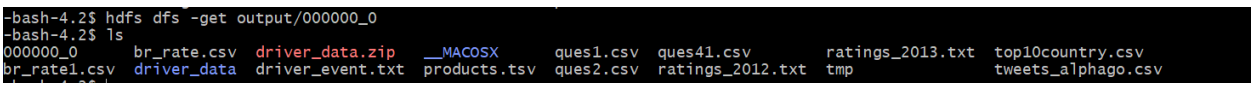



Step11: Once the file is stored in hdfs, download the file from hdfs to linux system.

hdfs dfs -get output/file

ex hdfs dfs -get output/000000_0



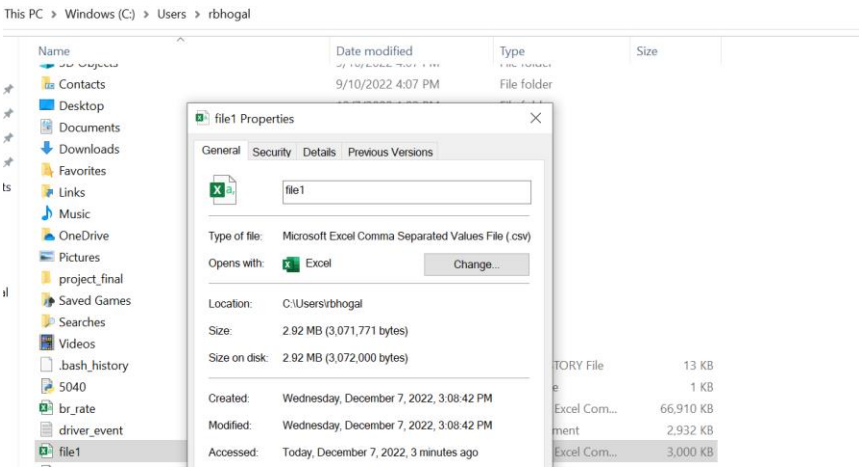

Step12: Download the file from linux to our local pc

scp username@serveradd:/home/username/000000_0 filename.csv

000000_0 will be saved as filename.csv or file1.csv as shown below in our C: Drive > Users > file1.csv

# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

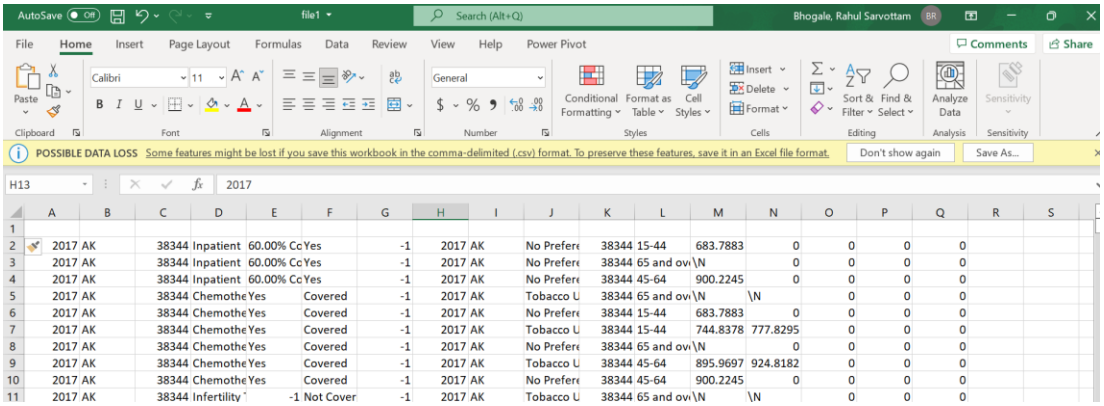
```
AD+rbhogal@STU-PF2XYGFC MINGW64 ~
$ scp rbhogal@144.24.14.145:/home/rbhogal/000000_0 file1.csv
rbhogal@144.24.14.145's password:
000000_0                                                    100% 3000KB
```

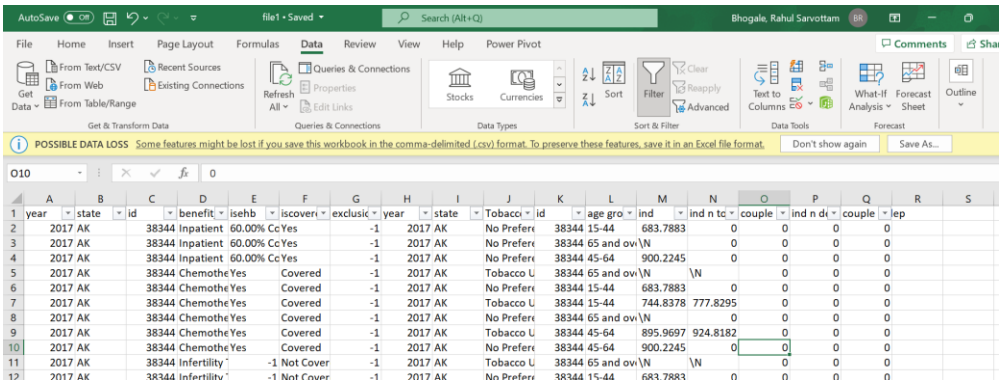



Step13: Once the file is downloaded, upload the file in any visualization tool(Excel/Powerbi/Tableau)

**(Note: Add the column name in the top row.)**

Analyze the data, through different filters, sorting and various visualization charts to solve your defined problem statement

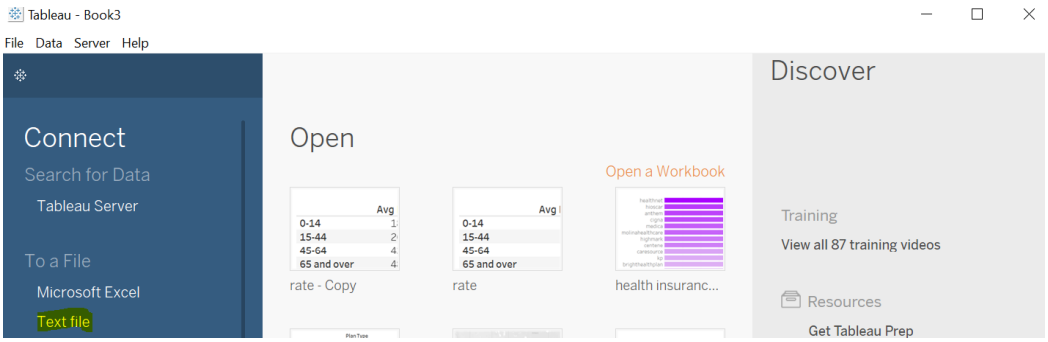


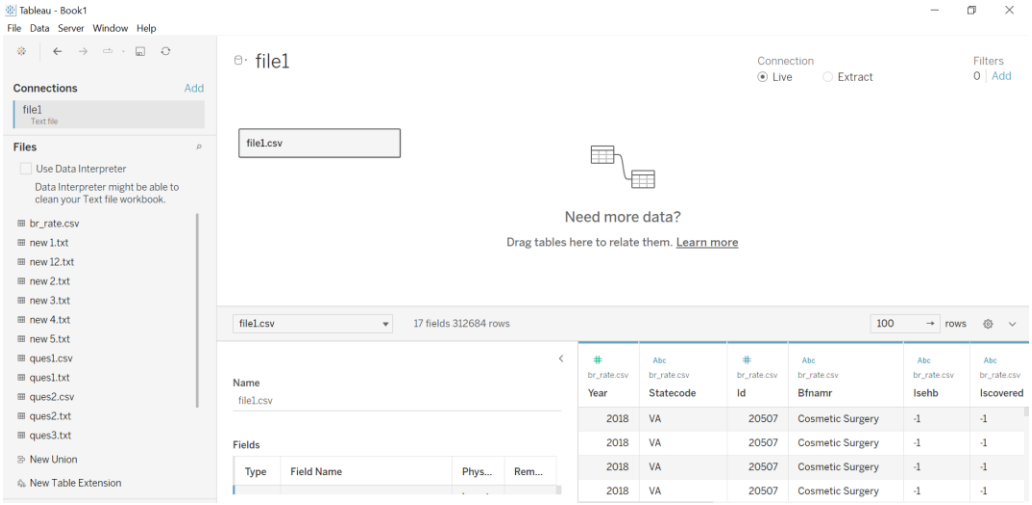
# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

Prepare the csv file for Tableau visualization.

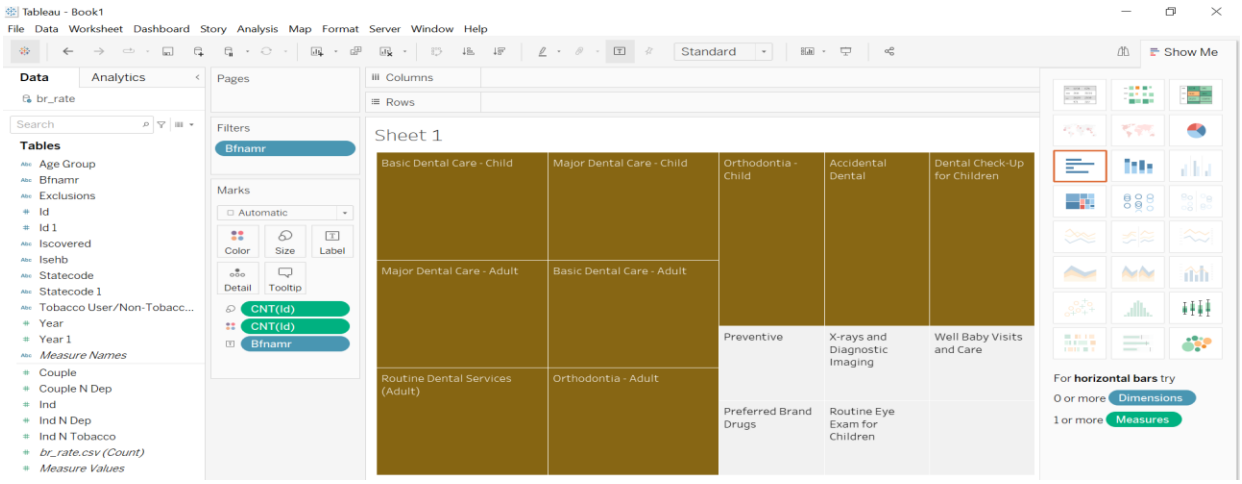
Upload the file through the data source connection




Once the connection is established, the file uploaded would look like.




Click on Sheet1, and build the dashboard similarly to this one where, bfnamr filtered as top 15 count using HeatMap/TreeMap Chart.



# HEALTH INSURANCE MARKETPLACE DATA ANALYSIS USING HADOOP & HIVE (Oracle Cloud)

Similarly, we can build a geospatial visualization as well.