

**Big Data Analytics Project – Project 3**  
**Comparison of Performance between Apache-Hive,**  
**Apache-Drill & Spark SQL**

**Name: Muhammad Ibrahim**  
**ERP: 15976**

# Machine Configuration

- MacBook Pro 2017
- Core i5
- 256 Flash Drive
- 16 GB RAM



## Project Video Link

[Final Video.mov](#)

Data Set Url:

<https://www.kaggle.com/datasets/emilyparrott/texas-education-agency-data-20122019>

Data Set Size: 2.39 GB

## Texas Education Agency Data 2012-2019

Data    Code (1)    Discussion (1)    Metadata

▲ 12 New Notebook

### About Dataset

#### Context

We at [Teaching Trust](#) are sharing the student achievement dataset that we have spent the past year gathering in partnership with [TCB Analytics](#). Our hope is that these data will drive insights and actions that advance public education. We are big believers in the collective impact of data #transparency.

Over the last year, Teaching Trust worked together with [TCB Analytics](#) to gather all publicly available data released by the [Texas Education Agency \(TEA\)](#) between 2012 and 2019. Multiple times each year, the TEA releases student academic achievement data, broadly referred to as STAAR (State of Texas Assessment of Academic Readiness) data, but each release is typically formatted differently (meaning that it is difficult to work with across multiple years). In downloading each and every data release, we were able to gather all the data in one place and begin the process of tidying it so that it is usable for longitudinal data analysis.

Since Teaching Trust is winding down operations over the next several months, we have decided to make this scraped data (and all the R code used to scrape and tidy it) publicly available via Kaggle Datasets. We are encouraging other education researchers and data scientists to use the data and code for their projects.

Texas is a large state, with over 5 million students (~10% of the nation's public school students), so we know that this could be useful for understanding and improving public education. In addition, other education advocates, nonprofits, and funders can use this data to understand their own impact. We chose Kaggle Datasets to make the data publicly available.

# Texas Education Agency Data 2012-2019

Data    Code (1)    Discussion (1)    Metadata

▲ 12 ▾ N

Education

Primary and Secondary Schools

## new\_csv (4 files)



tidy\_campprof\_enrollmn...  
119.26 MB



tidy\_campprof\_staff\_20...  
724.45 MB



tidy\_campstaar1\_2012t...  
555.63 MB



tidy\_campstaar2\_2013t...  
919.37 MB

## Data Definition i.e. Columns

[Create](#)

Home Competitions Datasets Code Discussions Courses More

RECENTLY VIEWED Texas Education Agen... Chicago crimes 2001... Starter: Smart meters i...

**Texas Education Agency Data 2012-2019**

Data    Code (1)    Discussion (1)    Metadata

▲ 12    New Notebook    Download (3 GB)    :

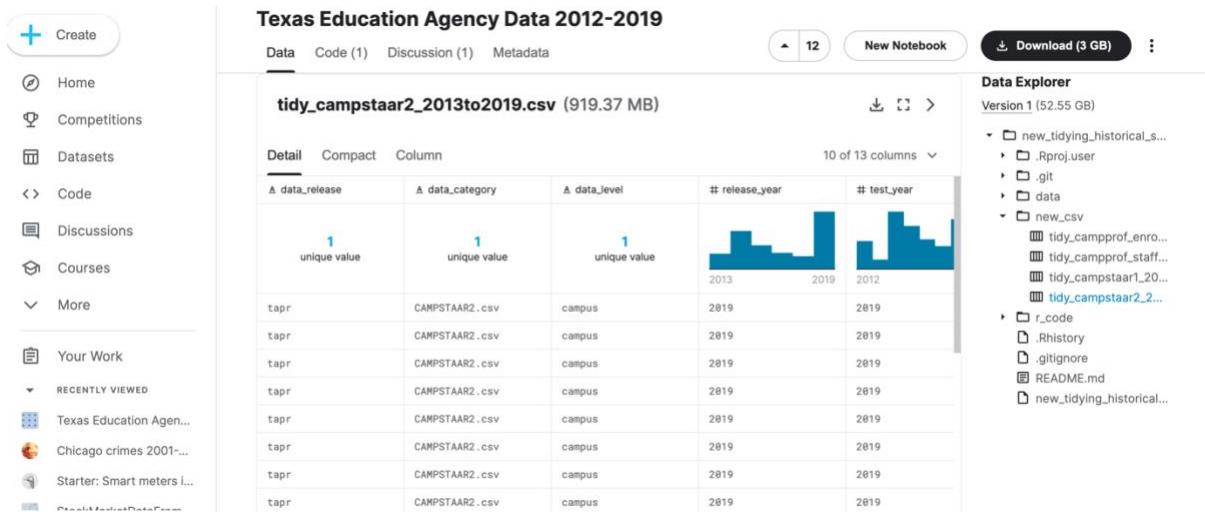
**tidy\_campprof\_enrollment\_2013\_to\_2019.csv (119.26 MB)**

Detail    Compact    Column    9 of 9 columns

# campus_number	data_release	data_level	data_category	release_year
1.90m	unique value	unique value	unique value	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013
1902001	tapr	campus	CAMPPROF.csv	2013

**Data Explorer**  
Version 1 (52.55 GB)

- new\_tidying\_historical\_s...
- .Rproj.user
- .git
- data
- new\_csv
  - tidy\_campprof\_enro...
  - tidy\_campprof\_staff...
  - tidy\_campstaar1\_20...
  - tidy\_campstaar2\_2...
- r\_code
- .Rhistory
- .gitignore
- README.md
- new\_tidying\_historical...

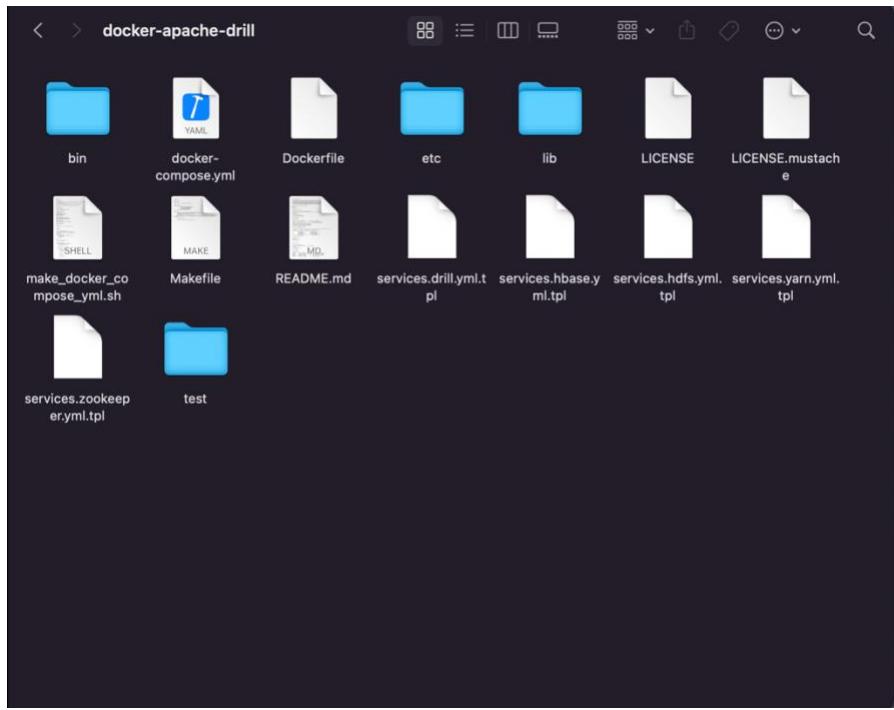


## Project Steps Details

### 1. Apache Drill

- Clone repository and go inside repository directory
  - `git clone https://github.com/smizy/docker-apache-drill.git`

```
BigDataProject -- bash -- 138x33
Muhammads-MacBook-Air-2:BigDataProject ibrahim$ git clone https://github.com/smizy/docker-apache-drill.git
```



- Create Virtual network on docker
  - docker network create network
  - The repo includes shell script run it and save results to create docker-compose file
  - `./make_docker_compose.yml sh hdfs drill > docker-compose.yml`
- Compose File after running shell Script

```
146  datanode-3:
147    container_name: datanode-3
148    networks: [ "vnet" ]
149    hostname: datanode-3.vnet
150    image: smizy/hadoop-base:2.7.7-alpine
151    expose: [ "50010", "50020", "50075" ]
152    environment:
153      - SERVICE_50010_NAME=datanode
154      - SERVICE_50020_IGNORE=true
155      - SERVICE_50075_IGNORE=true
156      -
157        HADOOP_ZOOKEEPER_QUORUM=zookeeper-1.vnet:2181,zookeeper-2
158          .vnet:2181,zookeeper-3.vnet:2181
159      - HADOOP_HEAPSIZE=1000
160
161    entrypoint: entrypoint.sh
162    command: datanode
163
164  drillbit-1:
165    container_name: drillbit-1
166    networks: [ "vnet" ]
167    hostname: drillbit-1.vnet
168    image: smizy/apache-drill:1.16.0-alpine
169    ports:
170      - 8047
171    depends_on: [ "zookeeper-1" ]
172    environment:
173      - SERVICE_8047_NAME=drillbit
174      - DRILL_HEAP=512M
175      - DRILL_MAX_DIRECT_MEMORY=1G
176      -
177        DRILL_ZOOKEEPER_QUORUM=zookeeper-1.vnet:2181,zookeeper-2
178          .vnet:2181,zookeeper-3.vnet:2181
```

- Now run docker-compose up -d to download and run container for name-node, data-node and Apache Drill on given ports

```
Muhammads-MacBook-Air-2:docker-apache-drill ibrahim$ ls
Dockerfile           README.md          lib               services.hdfs.yml.tpl
LICENSE              bin                make_docker_compose_yml.sh  services.yarn.yml.tpl
LICENSE.mustache     docker-compose.yml  services.drill.yml.tpl   services.zookeeper.yml.tpl
Makefile              etc                services.hbase.yml.tpl  test
Muhammads-MacBook-Air-2:docker-apache-drill ibrahim$ ls
Dockerfile           README.md          lib               services.hdfs.yml.tpl
LICENSE              bin                make_docker_compose_yml.sh  services.yarn.yml.tpl
LICENSE.mustache     docker-compose.yml  services.drill.yml.tpl   services.zookeeper.yml.tpl
Makefile              etc                services.hbase.yml.tpl  test
Muhammads-MacBook-Air-2:docker-apache-drill ibrahim$ docker-compose up -d
Starting namenode-2 ... done
Starting datanode-2 ... done
Starting journalnode-3 ... done
Starting zookeeper-1 ... done
Starting datanode-1 ... done
Starting zookeeper-3 ... done
Starting journalnode-2 ... done
Starting datanode-3 ... done
Starting zookeeper-2 ... done
Starting namenode-1 ... done
Starting journalnode-1 ... done
Starting drillbit-1 ... done
Muhammads-MacBook-Air-2:docker-apache-drill ibrahim$
```

 drillbit-1	<a href="#">smizy/apache-d...</a>	RUNNING	PORT: 54004
 journalnode-2	<a href="#">smizy/hadoop-b...</a>	RUNNING	
 zookeeper-1	<a href="#">smizy/zooke...e...</a>	RUNNING	
 journalnode-3	<a href="#">smizy/hadoop-b...</a>	RUNNING	
 zookeeper-2	<a href="#">smizy/zooke...e...</a>	RUNNING	
 zookeeper-3	<a href="#">smizy/zooke...e...</a>	RUNNING	
 journalnode-1	<a href="#">smizy/hadoop-b...</a>	RUNNING	
 namenode-1	<a href="#">smizy/hadoop-b...</a>	RUNNING	PORT: 54003
 namenode-2	<a href="#">smizy/hadoop-b...</a>	EXITED (1)	PORT: 0
 datanode-3	<a href="#">smizy/hadoop-b...</a>		

- Copy Data into datanode container using
  - docker cp tidy\_campstaar2\_2013to2019.csv datanode-1:/home/data
  - docker cp tidy\_campstaar1\_2012to2019.csv datanode-1:/share
  - docker cp tidy\_campstaar2\_2013to2019.csv spark-master:/home
  - docker cp tidy\_campprof\_enrollment\_2013\_to\_2019.csv spark-master:/home

## Data in Container

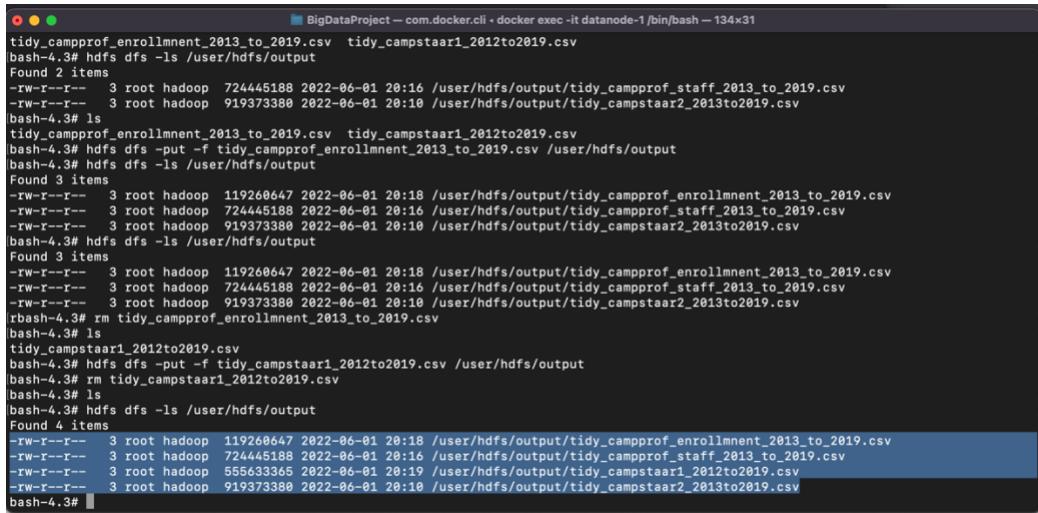
```

BigDataProject — com.docker.cli • docker exec -it datanode-1 /bin/bash — 134x31
[–find <path> ... <expression> ...]
[–get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
[–getfacl [-R] <path>]
[–getfattr [-R] {-n name | -d} [-e en] <path>]
[–getmerge [-nl] <src> <localdst>]
[–help [cmd ...]]
[–ls [-d] [-h] [-R] [<path> ...]]
[–mkdir [-p] <path> ...]
[–moveFromLocal <localsrc> ... <dst>]
[bash-4.3# hadoop fs -mkdir /user/data/
mkdir: '/user/data/': No such file or directory
[bash-4.3# hdfs dfs -mkdir -p /user/hdfs/output
bash-4.3# ls
tidy_campprof_enrollment_2013_to_2019.csv  tidy_campstaar1_2012to2019.csv
tidy_campprof_staff_2013_to_2019.csv  tidy_campstaar2_2013to2019.csv
[bash-4.3# hdfs dfs -put -f tidy_campstaar2_2013to2019.csv /user/hdfs/output
put: '/user/data/': No such file or directory
bash-4.3# hdfs dfs -put -f tidy_campstaar2_2013to2019.csv /user/hdfs/output
[bash-4.3# hdfs dfs -ls /user/hdfs/output
Found 1 items
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csv
[bash-4.3# ls
tidy_campprof_enrollment_2013_to_2019.csv  tidy_campstaar1_2012to2019.csv
tidy_campprof_staff_2013_to_2019.csv  tidy_campstaar2_2013to2019.csv
[bash-4.3# rm tidy_campstaar2_2013to2019.csv
[bash-4.3# ls
tidy_campprof_enrollment_2013_to_2019.csv  tidy_campprof_staff_2013_to_2019.csv      tidy_campstaar1_2012to2019.csv
[bash-4.3# unlink tidy_campstaar2_2013to2019.csv
unlink: can't remove file 'tidy_campstaar2_2013to2019.csv': No such file or directory
bash-4.3# hdfs dfs -put -f tidy_campprof_staff_2013_to_2019.csv /user/hdfs/output
[bash-4.3# hdfs dfs -ls /user/hdfs/output

```

- Copy Data in HDFS using
  - hdfs dfs -put -f tidy\_campprof\_enrollment\_2013\_to\_2019.csv /user/hdfs/output
  - hdfs dfs -put -f tidy\_campstaar1\_2012to2019.csv /user/hdfs/output
  - hdfs dfs -put -f tidy\_campprof\_staff\_2013\_to\_2019.csv /user/hdfs/output
  - hdfs dfs -put -f tidy\_campstaar2\_2013to2019.csv /user/hdfs/output

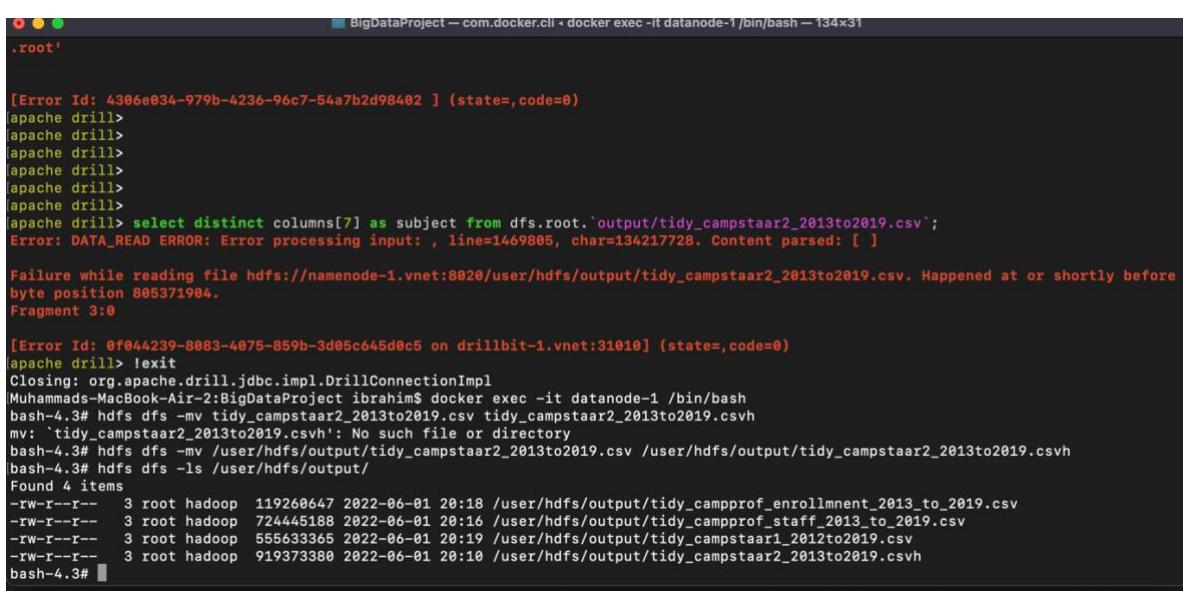
## Confirm Data transfer using ls command



```
BigDataProject — com.docker.cli - docker exec -it datanode-1 /bin/bash — 134x31
tidy_campprof_enrollment_2013_to_2019.csv tidy_campstaar1_2012to2019.csv
bash-4.3# hdfs dfs -ls /user/hdfs/output
Found 2 items
-rw-r--r-- 3 root hadoop 724445188 2022-06-01 20:16 /user/hdfs/output/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csv
bash-4.3# ls
tidy_campprof_enrollment_2013_to_2019.csv tidy_campstaar1_2012to2019.csv
bash-4.3# hdfs dfs -put -f tidy_campprof_enrollment_2013_to_2019.csv /user/hdfs/output
bash-4.3# hdfs dfs -ls /user/hdfs/output
Found 3 items
-rw-r--r-- 3 root hadoop 119260647 2022-06-01 20:18 /user/hdfs/output/tidy_campprof_enrollment_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 724445188 2022-06-01 20:16 /user/hdfs/output/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csv
bash-4.3# hdfs dfs -ls /user/hdfs/output
Found 3 items
-rw-r--r-- 3 root hadoop 119260647 2022-06-01 20:18 /user/hdfs/output/tidy_campprof_enrollment_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 724445188 2022-06-01 20:16 /user/hdfs/output/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csv
rbash-4.3# rm tidy_campprof_enrollment_2013_to_2019.csv
bash-4.3# ls
tidy_campstaar1_2012to2019.csv
bash-4.3# hdfs dfs -put -f tidy_campstaar1_2012to2019.csv /user/hdfs/output
bash-4.3# rm tidy_campstaar1_2012to2019.csv
bash-4.3# ls
bash-4.3# hdfs dfs -ls /user/hdfs/output
Found 4 items
-rw-r--r-- 3 root hadoop 119260647 2022-06-01 20:18 /user/hdfs/output/tidy_campprof_enrollment_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 724445188 2022-06-01 20:16 /user/hdfs/output/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 555633365 2022-06-01 20:19 /user/hdfs/output/tidy_campstaar1_2012to2019.csv
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csv
bash-4.3#
```

Apache drill is capable of querying text and CSV files, so I will query directly from CSV, changing extension to CSVH will allow us to use CSV Column names directly otherwise we have to use Column Index i.e. columns[2]

- Change CSV extension to CSVH using
  - `hdfs dfs mv tidy_campstaar2_2013to2019.csv tidy_campstaar2_2013to2019.csvh`



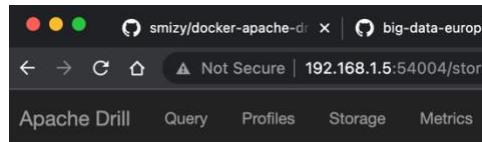
```
BigDataProject — com.docker.cli - docker exec -it datanode-1 /bin/bash — 134x31
.root'

[Error Id: 4306e034-979b-4236-96c7-54a7b2d98402 ] (state=,code=0)
apache drill>
apache drill> select distinct columns[7] as subject from dfs.root.`output/tidy_campstaar2_2013to2019.csv`;
Error: DATA_READ ERROR: Error processing input: , line=1469805, char=134217728. Content parsed: [ ]
Failure while reading file hdfs://namenode-1.vnet:8020/user/hdfs/output/tidy_campstaar2_2013to2019.csv. Happened at or shortly before byte position 805371904.
Fragment 3:0

[Error Id: 0f044239-8083-4075-859b-3d05c645d0c5 on drillbit-1.vnet:31010] (state=,code=0)
apache drill> !exit
Closing: org.apache.drill.jdbc.impl.DrillConnectionImpl
Muhammads-MacBook-Air-2:BigDataProject ibrahim$ docker exec -it datanode-1 /bin/bash
bash-4.3# hdfs dfs -mv tidy_campstaar2_2013to2019.csv tidy_campstaar2_2013to2019.csvh
mv: `tidy_campstaar2_2013to2019.csvh': No such file or directory
bash-4.3# hdfs dfs -mv /user/hdfs/output/tidy_campstaar2_2013to2019.csv /user/hdfs/output/tidy_campstaar2_2013to2019.csvh
bash-4.3# hdfs dfs -ls /user/hdfs/output/
Found 4 items
-rw-r--r-- 3 root hadoop 119260647 2022-06-01 20:18 /user/hdfs/output/tidy_campprof_enrollment_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 724445188 2022-06-01 20:16 /user/hdfs/output/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 3 root hadoop 555633365 2022-06-01 20:19 /user/hdfs/output/tidy_campstaar1_2012to2019.csv
-rw-r--r-- 3 root hadoop 919373380 2022-06-01 20:10 /user/hdfs/output/tidy_campstaar2_2013to2019.csvh
bash-4.3#
```

Change Apache Drill configuration to read data from HDFS root directory, open [Error! Hyperlink reference not valid.](#)  
i.e. <http://192.168.1.5:54004/storage/dfs>

```
{  
  "type": "file",  
  "enabled": true,  
  "connection": "hdfs://namenode-1.vnet:8020/",  
  "config": null,  
  "workspaces": {  
    "root": {  
      "location": "/user/hdfs",  
      "writable": false,  
      "defaultInputFormat": null  
    },  
    "tmp": {  
      "location": "/tmp",  
      "writable": true,  
      "defaultInputFormat": null  
    }  
  }  
}
```



Configuration

```
1  [{  
2   "type": "file",  
3   "connection": "hdfs://namenode-1.vnet:8020/",  
4   "config": null,  
5   "workspaces": {  
6     "tmp": {  
7       "location": "/user/hdfs",  
8       "writable": false,  
9       "defaultInputFormat": null,  
10      "allowAccessOutsideWorkspace": false  
11    },  
12    "root": {  
13      "location": "/user/hdfs",  
14      "writable": false,  
15      "defaultInputFormat": null,  
16      "allowAccessOutsideWorkspace": false  
17    },  
18    "formats": {  
19      "psv": {  
20        "type": "text",  
21        "extensions": [  
22          "tbl"  
23        ],  
24        "delimiter": "|"  
25      }  
26    }  
27  }  
28}  
29}
```

Back Update Disable Export Delete

- Open Drill Command Line using

## ○ docker exec -it drillbit-1 drill-conf

```
Muhammads-MacBook-Air-2:docker-apache-drill ibrahim$ docker exec -it drillbit-1 drill-conf
Apache Drill 1.16.0
"Say hello to my little Drill."
apache drill> █
```

## Queries in Apache Drill

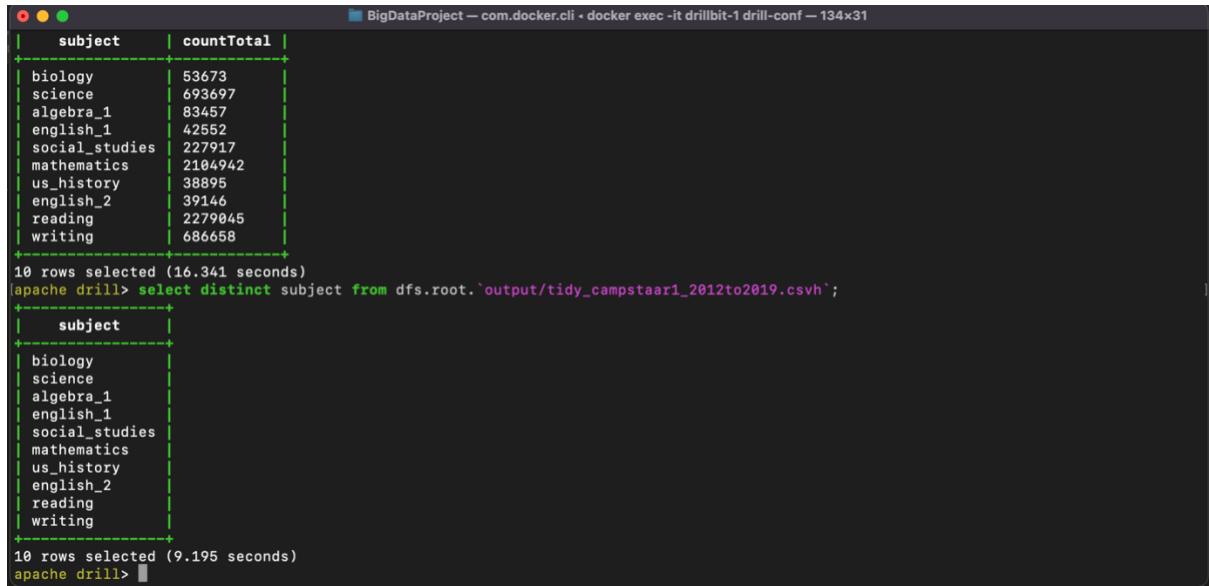
-> Query Data in drill using csv in HDFS root path

```
select * from dfs.root.`output/tidy_campstaar1_2012to2019.csv`;
```

```
BigDataProject — com.docker.cli • docker exec -it drillbit-1 drill-conf — 134x31
Apache Drill 1.16.0
"Keep your data close, but your Drillbits closer."
apache drill> select * from dfs.root.`output/tidy_campstaar1_2012to2019.csv` limit 20;
+-----+
| columns
+-----+
| [ "data_release", "data_category", "data_level", "release_year", "test_year", "campus_number", "grade_level", "subject", "proficiency", "demog", "numerator", "denominator", "new_rate" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "all_students", "41", "48", "85.42" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "all_students", "51", "57", "89.47" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "at_risk", "11", "17", "64.71" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "at_risk", "32", "38", "84.21" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "economic_disadvant", "17", "20", "85" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "economic_disadvant", "19", "20", "95" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "female", "23", "25", "92" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "female", "23", "25", "92" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "male", "18", "23", "78.26" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "male", "28", "32", "87.5" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "special_ed", "6", "12", "50" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "special_ed", "11", "14", "78.57" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "white", "38", "44", "86.36" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "white", "45", "51", "88.24" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "biology", "approaches", "all_students", "48", "52", "92.31" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "biology", "approaches", "all_students", "47", "48", "97.92" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "biology", "approaches", "at_risk", "13", "17", "76.47" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2014", "1902001", "eoc", "biology", "approaches", "at_risk", "28", "29", "96.55" ] |
| [ "tapr", "CAMPSTAAR1.csv", "campus", "2014", "2013", "1902001", "eoc", "biology", "approaches", "economic_disadvant", "18", "20", "90" ] |
+-----+
20 rows selected (1.083 seconds)
apache drill> █
```

Distinct Query:

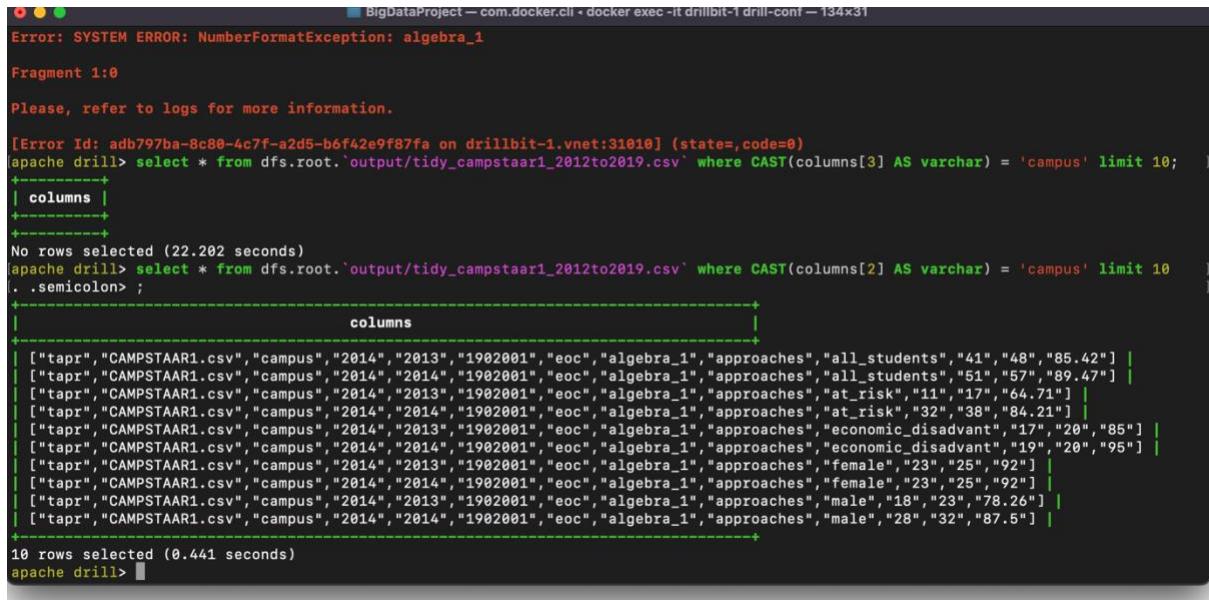
```
select distinct subject from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;
```



```
+----+-----+
| subject | countTotal |
+----+-----+
| biology | 53673 |
| science | 693697 |
| algebra_1 | 83457 |
| english_1 | 42552 |
| social_studies | 227917 |
| mathematics | 2184942 |
| us_history | 38895 |
| english_2 | 39146 |
| reading | 2279045 |
| writing | 686658 |
+----+-----+
10 rows selected (16.341 seconds)
apache drill> select distinct subject from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;
+----+
| subject |
+----+
| biology |
| science |
| algebra_1 |
| english_1 |
| social_studies |
| mathematics |
| us_history |
| english_2 |
| reading |
| writing |
+----+
10 rows selected (9.195 seconds)
apache drill>
```

## Where Clause:

```
select * from dfs.root.`output/tidy_campstaar1_2012to2019.csv` where
CAST(columns[2] AS varchar) = 'campus' limit 10
```



```
Error: SYSTEM ERROR: NumberFormatException: algebra_1
Fragment 1:0
Please, refer to logs for more information.

[Error Id: adb797ba-8c80-4c7f-a2d5-b6f42e9f87fa on drillbit-1.vnet:31010] (state=,code=0)
apache drill> select * from dfs.root.`output/tidy_campstaar1_2012to2019.csv` where CAST(columns[3] AS varchar) = 'campus' limit 10;
+-----+
| columns |
+-----+
+-----+
No rows selected (22.202 seconds)
apache drill> select * from dfs.root.`output/tidy_campstaar1_2012to2019.csv` where CAST(columns[2] AS varchar) = 'campus' limit 10
. . semicolon> ;
+-----+
| columns |
+-----+
[{"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "all_students", "41", "48", "85.42"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "all_students", "51", "57", "89.47"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "at_risk", "11", "17", "64.71"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "at_risk", "32", "38", "84.21"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "economic_disadvant", "17", "20", "85"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "economic_disadvant", "19", "20", "95"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "female", "23", "25", "92"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "female", "23", "25", "92"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2013", "1902001", "eoc", "algebra_1", "approaches", "male", "18", "23", "78.26"}, {"tapr": "CAMPSTAAR1.csv", "campus": "2014", "2014", "1902001", "eoc", "algebra_1", "approaches", "male", "28", "32", "87.5"}]
+-----+
10 rows selected (0.441 seconds)
apache drill>
```

## Count, Group by subjects

```
select subject, count(*) as countTotal from
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by subject;
```

```
"You told me to, Drill Sergeant!"
[Error Id: f42d668a-7b63-42ef-90ea-879f7143ffbe ] (state=,code=0)
[apache drill> select subject, count(*) as countTotal from dfs.root.`output/tidy_campstaar2_2013to2019.csvh`;
Error: VALIDATION ERROR: From line 1, column 8 to line 1, column 14: Expression 'subject' is not being grouped

[Error Id: be85488c-c654-4889-a8ff-fa9d45d432d1 on drillbit-1.vnet:31010] (state=,code=0)
[apache drill> select subject, count(*) as countTotal from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by subject;
+-----+
| subject | countTotal |
+-----+
| biology | 53673
| science | 693697
| algebra_1 | 83457
| english_1 | 42552
| social_studies | 227917
| mathematics | 2184942
| us_history | 38895
| english_2 | 39146
| reading | 2279045
| writing | 686658
+-----+
10 rows selected (11.499 seconds)
apache drill> 
```

## Order by DESC query – Year Wise Data

```
select release_year, count(*) as TotalCount from
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year order by
TotalCount DESC;
```

```
+-----+
| 2014 | 869282
| 2019 | 3524150
+-----+
6 rows selected (9.784 seconds)
[apache drill> select release_year, count(*) as TotalCount from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year
order by TotalCount ASC;
+-----+
| release_year | TotalCount |
+-----+
| 2015 | 319319
| 2013 | 368828
| 2017 | 476741
| 2018 | 691662
| 2014 | 869282
| 2019 | 3524150
+-----+
6 rows selected (8.883 seconds)
[apache drill> select release_year, count(*) as TotalCount from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year
order by TotalCount DESC;
+-----+
| release_year | TotalCount |
+-----+
| 2019 | 3524150
| 2014 | 869282
| 2018 | 691662
| 2017 | 476741
| 2013 | 368828
| 2015 | 319319
+-----+
6 rows selected (8.587 seconds)
apache drill> 
```

# Total Star Students Learning Mathematics -- Count, Where Mathematics, Result -> 2104942

```
select count(*) as countTotal from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where subject = 'mathematics'
```

```
BigDataProject — com.docker.cli + docker exec -it drillbit-1 drill-conf — 134x31  
10 rows selected (16.341 seconds)  
apache drill> select distinct subject from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;  
+-----+  
| subject |  
+-----+  
| biology |  
| science |  
| algebra_1 |  
| english_1 |  
| social_studies |  
| mathematics |  
| us_history |  
| english_2 |  
| reading |  
| writing |  
+-----+  
10 rows selected (9.195 seconds)  
apache drill> select subject, count(*) as countTotal from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` where subject = 'mathematics';  
Error: VALIDATION ERROR: From line 1, column 8 to line 1, column 14: Expression 'subject' is not being grouped  
  
[Error Id: 64ae32a9-0feb-4bcc-ae97-1535b24c93ba ] (state=,code=0)  
apache drill> select count(*) as countTotal from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` where subject = 'mathematics';  
+-----+  
| countTotal |  
+-----+  
| 2104942 |  
+-----+  
1 row selected (14.037 seconds)  
apache drill>
```

## Check which years data we have

```
select distinct release_year from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;
```

```
BigDataProject — com.docker.cli + docker exec -it drillbit-1 drill-conf — 134x31  
+-----+  
| release_year |  
+-----+  
| 2017 |  
| 2014 |  
| 2015 |  
| 2018 |  
| 2019 |  
| 2013 |  
+-----+  
6 rows selected (9.027 seconds)  
apache drill> select count(*) as TotalCount, release_year from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year;  
r;  
Error: VALIDATION ERROR: From line 1, column 32 to line 1, column 43: Expression 'release_year' is not being grouped  
  
[Error Id: e7e4ac68-ae80-47a5-80c5-be039acb6703 ] (state=,code=0)  
apache drill> select release_year, count(*) as TotalCount from dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year;  
+-----+-----+  
| release_year | TotalCount |  
+-----+-----+  
| 2017 | 476741 |  
| 2014 | 869282 |  
| 2015 | 319319 |  
| 2018 | 691662 |  
| 2019 | 3524150 |  
| 2013 | 368828 |  
+-----+-----+  
6 rows selected (8.57 seconds)  
apache drill>
```

# Following Queries were executed on Apache Drill to note performance

## 1. Distinct Subject

```
select distinct subject from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;  
9.195 seconds
```

## 2. Count, Group by subjects

```
select subject, count(*) as countTotal from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by subject;  
11.499 seconds
```

## 3. Total Star Students Learning Mathematics -- Count, Where Mathematics, Result -> 2104942

```
select count(*) as countTotal from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` where subject = 'mathematics'  
14.037 seconds
```

## 4. Check which years data we have

```
select distinct release_year from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;  
9.027 seconds
```

## 5. Total Count of data year wise

```
select release_year, count(*) as TotalCount from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year;  
8.57 seconds
```

## 6. which year have most data

```
select release_year, count(*) as TotalCount from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by release_year order by  
TotalCount DESC;  
8.587 seconds
```

## 7. Look at demographics of Star students

```
select distinct demog from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh`;  
7.234 seconds
```

8. Count female students -> 672545

```
select count(*) from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where demog = 'female';  
6.655 seconds
```

9. Count female students learning maths -> 221017

```
select count(*) from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where demog = 'female' and subject = 'mathematics';  
8.72 seconds
```

10. Count of male students

```
select count(*) from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where demog = 'male';  
6.929 seconds
```

11. Count of male learning biology -> 6403

```
select count(*) from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where demog = 'male' and subject = 'biology'  
9.033 seconds
```

12. Count of asian students in year 2019 -> 63611

```
select count(*) from dfs.root.`output/tidy_campstaar1_2012to2019.csvh`  
where demog = 'asian' and release_year = '2019';  
8.036 seconds
```

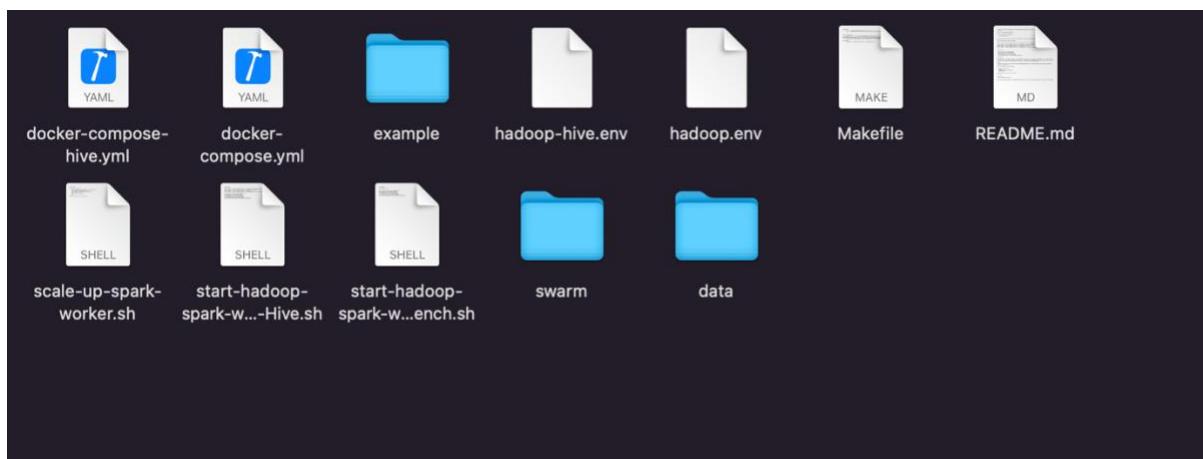
13. select which subject is mostly learned

```
select subject, count(*) as TotalCount from  
dfs.root.`output/tidy_campstaar1_2012to2019.csvh` group by subject order by  
TotalCount DESC  
7.505 seconds
```

## Project Steps Details

### Apache Spark & Hive

- Clone repository and go inside repository directory
- It includes Spark and Hive
  - git clone <https://github.com/big-data-europe/docker-hadoop-spark-workbench.git>



- Run Following one by one
  - docker-compose up -d
  - docker-compose -f docker-compose-hive.yml up -d namenode hive-metastore-postgresql
  - docker-compose -f docker-compose-hive.yml up -d datanode hive-metastore
  - docker-compose -f docker-compose-hive.yml up -d hive-server
  - docker-compose -f docker-compose-hive.yml up -d spark-master spark-worker spark-notebook hue

```

● ● ● docker-hadoop-spark-workbench — bash — 158x33
4654ee5f6659: Pull complete
88a820555dbf: Pull complete
14ead1715153: Pull complete
df32fa293aee: Pull complete
f0c9fa3e6fa9: Pull complete
ce0284e53103: Pull complete
cbdb8c564fcc: Pull complete
becd891f4d83: Pull complete
4305be6507a2: Pull complete
cd02abe11589: Pull complete
55b03a36f6bf: Pull complete
503ca7cfdec1: Pull complete
077ff5d65a2f0: Pull complete
16b679fd89e6: Pull complete
787b2d43064e: Pull complete
7dec2d090fb: Pull complete
914d86a00eba: Pull complete
a8ec9c576cb2: Pull complete
4714408faa44: Pull complete
ec27fa202f55: Pull complete
5687b41783e7: Pull complete
538fcac1f5af: Pull complete
8b1a2bf446c: Pull complete
cf63e94ebc3b: Pull complete
Digest: sha256:95b2b8b2e622b474851cb362934592fc22e520f608e891f9aa4be45032b23cc
Status: Downloaded newer image for bde2020/hdfs--filebrowser:3.11
Creating spark-notebook           ... done
Creating spark-master             ... done
Creating docker-hadoop-spark-workbench_hue_1 ... done
Creating namenode                 ... done
Creating docker-hadoop-spark-workbench_datanode_1 ... done
Creating docker-hadoop-spark-workbench_spark-worker_1 ... done
172-15-3-86:docker-hadoop-spark-workbench ibrahim$ 

```

## Data Node for Spark & Hive

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "DataNode" and displays the "DataNode on localhost:50010" interface.

**Cluster Information:**

Cluster ID:	CID-677ad518-5280-4120-b6eb-3d6c17d0445c
Version:	2.8.0

**Block Pools:**

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report
namenode:8020	BP-590204602-172.22.0.5-1653919979335	RUNNING	0s	18 minutes

**Volume Information:**

Directory	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/hadoop/dfs/data/current	532 KB	6.21 GB	0 B	0 B	1

## Name Node

The screenshot shows the HDFS NameNode Overview page. At the top, there is a navigation bar with tabs: Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The Overview tab is selected.

**Overview** 'namenode:8020' (active)

Started:	Mon May 30 19:13:10 +0500 2022
Version:	2.8.0, r91f2b7a13d1e97be65db92ddabc627cc29ac0009
Compiled:	Fri Mar 17 09:12:00 +0500 2017 by jdu from branch-2.8.0
Cluster ID:	CID-677ad518-5280-4120-b6eb-3d6c17d0445c
Block Pool ID:	BP-590204602-172.22.0.5-1653919979335

**Summary**

Security is off.  
Safemode is off.  
4 files and directories, 1 blocks = 5 total filesystem object(s).  
Heap Memory used 63.15 MB of 199.5 MB Heap Memory. Max Heap Memory is 889 MB.  
Non Heap Memory used 49.69 MB of 50.69 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	233.47 GB
DFS Used:	532 KB (0%)

## HDFS Browser

The screenshot shows the Hue File Browser. The URL is `localhost:8088/filebrowser/`. The interface includes a search bar, action buttons (Actions, Move to trash, Upload, New), and navigation buttons (History, Trash).

Home / user / ibrahim

Name	Size	User	Group	Permissions	Date
geolocation.csv	514.3 kB	ibrahim	ibrahim	-rw-r--r--	May 30, 2022 07:16 AM
..		root	supergroup	drwxr-xr-x	May 30, 2022 07:14 AM
.		ibrahim	ibrahim	drwxr-xr-x	May 30, 2022 07:16 AM

Show 45 of 1 items Page 1 of 1

## Copy Data Set to Container and then HDFS for Hive & Spark

```
docker cp tidy_campstaar2_2013to2019.csv spark-master:/home  
docker cp tidy_campstaar2_2013to2019.csv hive-server:/home
```

```
BigDataProject — com.docker.cli - docker exec -it spark-master /bin/sh — 149x24

Copy files/folders between a container and the local filesystem
172-15-3-86:BigDataProject ibrahim$ docker cp geolocation.csv hiver-server:/home/geolocation
Error: No such container:path: hiver-server:/home
172-15-3-86:BigDataProject ibrahim$ docker cp geolocation.csv hive-server:/home/geolocation
172-15-3-86:BigDataProject ibrahim$ docker exec -it hive-server /bin/sh
# cd ..
# ls
bin boot dev entrypoint.sh etc hadoop-data home lib lib64 media mnt opt proc root run sbin selinux srv sys tmp user usr var
# cd home
# ls
geolocation trucks
# cd geolocation
# ls
geolocation.csv
# hadoop fs -mkdir /user/data/
22/05/30 15:33:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# hadoop fs -put -f /home/geolocation /user/data/
22/05/30 15:33:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# hadoop fs -ls /user/data/geolocation
22/05/30 15:33:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r-- 1 root root 526677 2022-05-30 15:33 /user/data/geolocation/geolocation.csv
#
```

```
BigDataProject — com.docker.cli - docker exec -it hive-server /bin/sh — 134x24

# hadoop fs -mkdir /user/data/
22/05/31 08:24:29 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
mkdir: '/user/data': File exists
# hadoop fs -put -f /home/ /user/data/
22/05/31 08:25:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# hadoop fs -ls /user/data/
22/05/31 08:26:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x 1 root root 4096 2022-05-30 15:49 /user/data/geolocation
drwxr-xr-x 1 root root 4096 2022-05-31 08:25 /user/data/home
# hadoop fs -ls /user/data/home
22/05/31 08:26:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 6 items
drwxr-xr-x 1 root root 4096 2022-05-31 08:25 /user/data/home/geolocation
-rw-r--r-- 1 root root 119260647 2022-05-31 08:25 /user/data/home/tidy_campprof_enrollment_2013_to_2019.csv
-rw-r--r-- 1 root root 724445188 2022-05-31 08:25 /user/data/home/tidy_campprof_staff_2013_to_2019.csv
-rw-r--r-- 1 root root 555633365 2022-05-31 08:25 /user/data/home/tidy_campstaar1_2012to2019.csv
-rw-r--r-- 1 root root 919373380 2022-05-31 08:25 /user/data/home/tidy_campstaar2_2013to2019.csv
drwxr-xr-x 1 root root 4096 2022-05-31 08:25 /user/data/home/trucks
# /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000
```

```
BigDataProject — com.docker.cli - docker exec -it spark-master /bin/sh — 149x24

-----+
8,001 rows selected (5.747 seconds)
0: jdbc:hive2://localhost:10000> !exit
Closing: 0: jdbc:hive2://localhost:10000
# ls
geolocation.csv
# cd ..
# ls
geolocation trucks
# cd ..
# ls
bin boot dev entrypoint.sh etc hadoop-data home lib lib64 media mnt opt proc root run sbin selinux srv sys tmp user usr var
# cd home
# ls
geolocation trucks
# hadoop fs -ls /user/data/geolocation
22/05/30 15:49:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# hadoop fs -put -f /home/geolocation /user/data/
22/05/30 15:49:58 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# hadoop fs -ls /user/data/geolocation
22/05/30 15:50:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r-- 1 root root 526677 2022-05-30 15:49 /user/data/geolocation/geolocation.csv
#
```

# Start Spark & Hive

- docker exec -it hive-server /bin/sh
  - /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000
- docker exec -it spark-master /bin/sh
  - /spark/bin/spark-shell –master=local spark://spark-master:7077

```
hadoop-2.7.1  hive
# exit
Muhammads-MacBook-Air-2:docker-hadoop-spark-workbench ibrahim$ docker exec -it 236dff95264409432c5b6e73a21f0568f7fc04dcfb94daa2971e4dd6bb0a90a6
a90a6 /bin/sh
Error: No such container: 236dff95264409432c5b6e73a21f0568f7fc04dcfb94daa2971e4dd6bb0a90a6
Muhammads-MacBook-Air-2:docker-hadoop-spark-workbench ibrahim$ docker exec -it spark-master /bin/sh
# ls
bin  dev   etc      finish-step.sh  home  lib64    media  opt   root  sbin  srv  tmp   var
boot  entrypoint.sh  execute-step.sh  hadoop-data  lib   master.sh  mnt   proc  run   spark  sys  usr  wait-for-step.sh
# /spark/bin/spark-shell –master=local spark://spark-master:7077
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/06/02 12:45:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/06/02 12:45:16 WARN metastore.ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 1.2.0
22/06/02 12:45:16 WARN metastore.ObjectStore: Failed to get database default, returning NoSuchObjectException
22/06/02 12:45:20 WARN metastore.ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://172.22.0.2:4040
Spark context available as 'sc' (master = local[*], app id = local-1654173903340).
Spark session available as 'spark'.
Welcome to

    /----/----/----/
   \  \ -\ \ -\ / \ / \
  /--/ .--\ / / / \ \ \
 /_/_/           version 2.1.2-SNAPSHOT

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_121)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

```
538fcca1f5af: Pull complete
8b1a2bf4f46c: Pull complete
cf63e94ebc3b: Pull complete
Digest: sha256:95b2b8b2e622b474851cb362934592fc22e528f608e891f9aa4be45832b23cc0
Status: Downloaded newer image for bde2020/hdfs-filebrowser:3.11
Creating spark-notebook ... done
Creating spark-master ... done
Creating docker-hadoop-spark-workbench_hue_1 ... done
Creating namenode ... done
Creating docker-hadoop-spark-workbench_datanode_1 ... done
Creating docker-hadoop-spark-workbench_spark-worker_1 ... done
172-15-3-86:docker-hadoop-spark-workbench ibrahim$ docker-compose -f docker-compose-hive.yml up -d namenode hive-metastore-postgresql
Pulling hive-metastore-postgresql (bde2020/hive-metastore-postgresql:2.1.0)...
2.1.0: Pulling from bde2020/hive-metastore-postgresql
5c99d4a42d1a8: Already exists
22337bfbd13a9: Already exists
c3961b297acc: Already exists
5a17453338b4: Already exists
6364e0d7a283: Already exists
58c25f5c8ddad: Already exists
f0e675ce88d9: Already exists
10f26cc68a34: Already exists
873d2c220bfff: Already exists
fd10fb78ded6: Already exists
ff1356ba118b: Already exists
e030b3f381c4: Pull complete
bd9a5d52d94c: Pull complete
87306d915983: Pull complete
Digest: sha256:2f09fad2f16bc71956668eb0a99769cb43afa2306abea55a65c00c48810416d63d
Status: Downloaded newer image for bde2020/hive-metastore-postgresql:2.1.0
Recreating namenode ... done
Creating docker-hadoop-spark-workbench_hive-metastore-postgresql_1 ... done
172-15-3-86:docker-hadoop-spark-workbench ibrahim$
```

# Running Queries on Hive

## Creating Tables by reading CSV from HDFS

LOAD DATA INPATH

```
'/user/data/home/tidy_campprof_enrollment_2013_to_2019.csv' INTO  
TABLE tbl_enrolment2013_2019;
```

```
LOAD DATA INPATH '/user/data/home/tidy_campstaar1_2012to2019.csv'  
INTO TABLE tbl_star2012_2019;
```

```
LOAD DATA INPATH '/user/data/home/tidy_campstaar2_2013to2019.csv'  
INTO TABLE tbl_star2013_2019
```

```
BigDataProject — com.docker.cli • docker exec -it hive-server /bin/sh — 134x24  
... .> subject string,  
... .> proficiency string,  
... .> demog string)  
... .> COMMENT 'Star 2012_2019 Table'  
... .> ROW FORMAT DELIMITED  
... .> FIELDS TERMINATED BY ',';  
No rows affected (0.261 seconds)  
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS tbl_star2013_2019 (  
... .> data_release string,  
... .> data_category string,  
... .> data_level decimal(5,0),  
... .> release_year decimal(5,0),  
... .> test_year string,  
... .> campus_number int,  
... .> grade_level string,  
... .> subject string,  
... .> proficiency string,  
... .> demog string)  
... .> COMMENT 'Star 2012_2019 Table'  
... .> ROW FORMAT DELIMITED  
... .> FIELDS TERMINATED BY ',';  
No rows affected (0.241 seconds)  
0: jdbc:hive2://localhost:10000> show tables  
... .> |
```

```
BigDataProject — com.docker.cli • docker exec -it hive-server /bin/sh — 134x24  
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS tbl_enrolment2013_2019 (  
... .> campus_number int,  
... .> data_release string,  
... .> data_level string,  
... .> data_category decimal(5,0),  
... .> release_year decimal(5,0),  
... .> demog string,  
... .> count int,  
... .> _percent int)  
... .> COMMENT 'Enrolment 2013 Table'  
... .> ROW FORMAT DELIMITED  
... .> FIELDS TERMINATED BY ',';  
Error: Error while compiling statement: FAILED: ParseException line 9:0 cannot recognize input near '_percent' 'int' ')' in column name  
e or primary key or foreign key (state=42000,code=40000)  
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS tbl_enrolment2013_2019 (  
... .> campus_number int,  
... .> data_release string,  
... .> data_level string,  
... .> data_category decimal(5,0),  
... .> release_year decimal(5,0),  
... .> demog string,  
... .> count int,  
... .> percentage int)  
... .> COMMENT 'Enrolment 2013 Table'
```

## Distinct Query

```
select distinct subject from tbl_star2012_2019;
```

```
e=400000)
0: jdbc:hive2://localhost:10000> select distinct subject from tbl_star2012_2019;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+
| subject |
+-----+
| biology |
| english_1 |
| social_studies |
| subject |
| writing |
| algebra_1 |
| english_2 |
| mathematics |
| reading |
| science |
| us_history |
+-----+
11 rows selected (15.757 seconds)
0: jdbc:hive2://localhost:10000> select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| subject | counttotal |
+-----+-----+
| biology | 53673 |
| english_1 | 42552 |
| social_studies | 227917 |
| subject | 1 |
| writing | 686658 |
| algebra_1 | 83457 |
| english_2 | 39146 |
+-----+
```

## Group by & Count (\*)

```
select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
```

```
e=400000)
0: jdbc:hive2://localhost:10000> select distinct subject from tbl_star2012_2019;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+
| subject |
+-----+
| biology |
| english_1 |
| social_studies |
| subject |
| writing |
| algebra_1 |
| english_2 |
| mathematics |
| reading |
| science |
| us_history |
+-----+
11 rows selected (15.757 seconds)
0: jdbc:hive2://localhost:10000> select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| subject | counttotal |
+-----+-----+
| biology | 53673 |
| english_1 | 42552 |
| social_studies | 227917 |
| subject | 1 |
| writing | 686658 |
| algebra_1 | 83457 |
| english_2 | 39146 |
+-----+
```

# Where Clause

```
docker-hadoop-spark-workbench -- com.docker.cli - docker exec -it hive-server /bin/sh - 138x33
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| subject | counttotal |
+-----+-----+
| biology | 53673 |
| english_1 | 42552 |
| social_studies | 227917 |
| subject | 1 |
| writing | 686658 |
| algebra_1 | 83457 |
| english_2 | 39146 |
| mathematics | 2104942 |
| reading | 2279045 |
| science | 693697 |
| us_history | 38895 |
+-----+-----+
11 rows selected (9.498 seconds)
0: jdbc:hive2://localhost:10000> Display all 560 possibilities? (y or n)
0: jdbc:hive2://localhost:10000> t(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics'
. . . . .
> ;
Error: Error while compiling statement: FAILED: ParseException line 1:0 cannot recognize input near 't' '(' '*' (state=42000,code=40000)
0: jdbc:hive2://localhost:10000>
0: jdbc:hive2://localhost:10000> select count(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| counttotal |
+-----+-----+
| 2104942 |
+-----+-----+
1 row selected (7.788 seconds)
0: jdbc:hive2://localhost:10000>
```

## Order by

```
select release_year, count(*) as TotalCount from tbl_star2012_2019  
group by release_year order by TotalCount DESC;
```

```
+-----+  
| 6403 |  
+-----+  
1 row selected (7.642 seconds)  
0: jdbc:hive2://localhost:10000> select count(*) from tbl_star2012_2019 where demog = 'asian' and release_year = '2019';  
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.  
+-----+  
| c0 |  
+-----+  
| 63611 |  
+-----+  
1 row selected (7.776 seconds)  
0: jdbc:hive2://localhost:10000> select subject, count(*) as TotalCount from tbl_star2012_2019 group by subject order by TotalCount DESC;  
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.  
+-----+  
| subject      | totalcount |  
+-----+  
| reading       | 2279045   |  
| mathematics   | 2104942   |  
| science        | 693697    |  
| writing        | 686688    |  
| social_studies | 227917    |  
| algebra_1     | 83457     |  
| biology        | 53673     |  
| english_1      | 42552     |  
| english_2      | 39146     |  
| us_history     | 38895     |  
| subject        | 1          |  
+-----+  
11 rows selected (11.293 seconds)  
0: jdbc:hive2://localhost:10000>
```

## Checking Records for Each Subject

```
select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
```

```
e=40000)
0: jdbc:hive2://localhost:10000> select distinct subject from tbl_star2012_2019;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+
| subject |
+-----+
| biology |
| english_1 |
| social_studies |
| subject |
| writing |
| algebra_1 |
| english_2 |
| mathematics |
| reading |
| science |
| us_history |
+-----+
11 rows selected (15.757 seconds)
0: jdbc:hive2://localhost:10000> select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| subject | counttotal |
+-----+-----+
| biology | 53673 |
| english_1 | 42552 |
| social_studies | 227917 |
| subject | 1 |
| writing | 686658 |
| algebra_1 | 83457 |
| english_2 | 39146 |
+-----+
```

## Looking at how many records belongs to each year

```
select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year;
```

```
+-----+-----+
1 row selected (7.788 seconds)
0: jdbc:hive2://localhost:10000> select distinct release_year from tbl_star2012_2019;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+
| release_year |
+-----+
| NULL |
| 2013 |
| 2019 |
| 2015 |
| 2018 |
| 2014 |
| 2017 |
+-----+
7 rows selected (14.804 seconds)
0: jdbc:hive2://localhost:10000> select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. tez, spark) or using Hive 1.X releases.
+-----+-----+
| release_year | totalcount |
+-----+-----+
| NULL | 1 |
| 2013 | 368828 |
| 2019 | 3524150 |
| 2015 | 319319 |
| 2018 | 691662 |
| 2014 | 869282 |
| 2017 | 476741 |
+-----+
7 rows selected (14.692 seconds)
0: jdbc:hive2://localhost:10000> ■
```

# Following Queries are executed to note performance of Apache Hive

## 1. Distinct Subject

```
select distinct subject from tbl_star2012_2019;
```

15.757 seconds

## 2. Count, Group by subjects

```
select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;
```

9.498 seconds

## 3. Total Star Students Learning Mathematics -- Count, Where Mathematics, Result -> 2104942

```
select count(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics'
```

7.788 seconds

## 4. Check which years data we have

```
select distinct release_year from tbl_star2012_2019;
```

14.804 seconds

## 5. Total Count of data year wise

```
select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year;
```

14.692 seconds

## 6. which year have most data

```
select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year order by TotalCount DESC;
```

16.159 seconds

## 7. Look at demographics of Star students

```
select distinct demog from tbl_star2012_2019;
```

8.727 seconds

## 8. Count female students -> 672545

```
select count(*) from tbl_star2012_2019 where demog = 'female';
```

6.689 seconds

9. Count female students learning maths -> 221017

```
select count(*) from tbl_star2012_2019 where demog = 'female' and subject = 'mathematics';  
8.935 seconds
```

10. Count of male students

```
select count(*) from tbl_star2012_2019 where demog = 'male';  
6.953
```

11. Count of male learning biology -> 6403

```
select count(*) from tbl_star2012_2019 where demog = 'male' and subject = 'biology'  
7.642 seconds
```

12. Count of asian students in year 2019 -> 63611

```
select count(*) from tbl_star2012_2019 where demog = 'asian' and release_year = '2019';  
7.776 seconds
```

13. select which subject is mostly learned

```
select subject, count(*) as TotalCount from tbl_star2012_2019 group by subject order by TotalCount DESC  
11.293 seconds
```

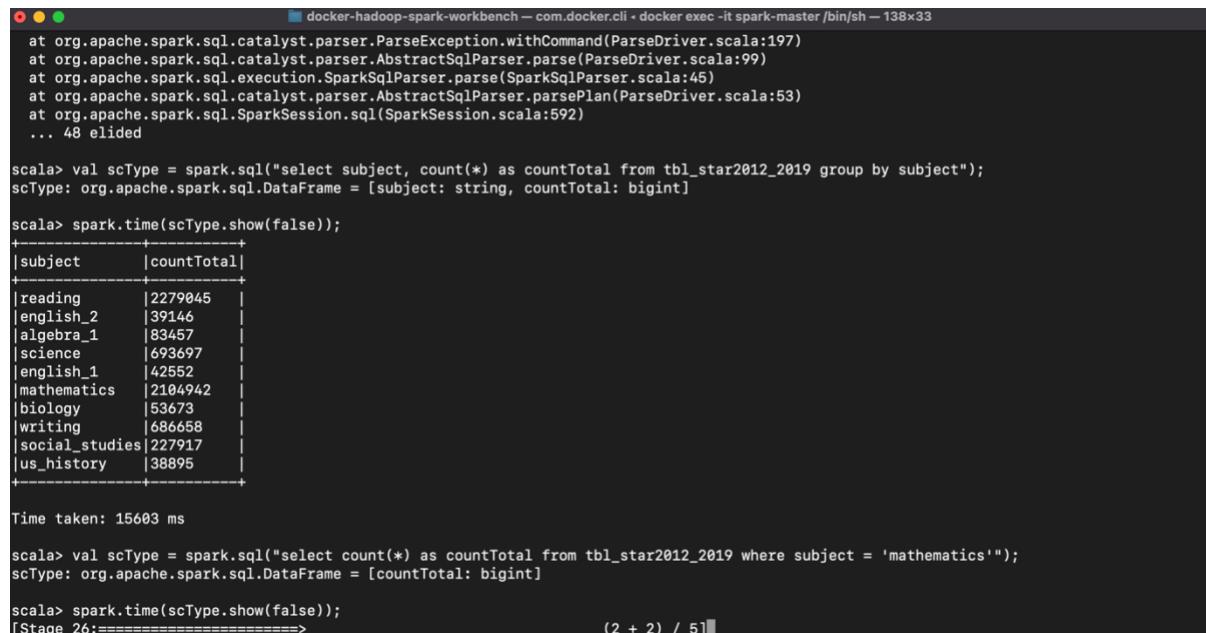
# Query on Apache Spark

## Config required to write queries

```
val df =  
spark.read.text("hdfs://namenode:8020/user/home/tidy_campstaar1_2012to2019.csv")  
  
df.createOrReplaceTempView("tidy_campstaar1_2012to2019")  
  
spark.time(spark.sql("select distinct subject from tidy_campstaar1_2012to2019"))
```

## Count, Group by subjects

```
select subject, count(*) as countTotal from tbl_star2012_2019 group by  
subject;
```



```
docker-hadoop-spark-workbench — com.docker.cli - docker exec -it spark-master /bin/sh — 138x33  
at org.apache.spark.sql.catalyst.parser.ParseException.withCommand(ParseDriver.scala:197)  
at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parse(ParseDriver.scala:99)  
at org.apache.spark.sql.execution.SparkSqlParser.parse(SparkSqlParser.scala:45)  
at org.apache.spark.sql.catalyst.parser.AbstractSqlParser.parsePlan(ParseDriver.scala:53)  
at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:592)  
... 48 elided  
  
scala> val scType = spark.sql("select subject, count(*) as countTotal from tbl_star2012_2019 group by subject");  
scType: org.apache.spark.sql.DataFrame = [subject: string, countTotal: bigint]  
  
scala> spark.time(scType.show(false));  
+-----+-----+  
|subject |countTotal|  
+-----+-----+  
|reading |2279845 |  
|english_2 |39146 |  
|algebra_1 |83457 |  
|science |693697 |  
|english_1 |42552 |  
|mathematics |2104942 |  
|biology |53673 |  
|writing |6866568 |  
|social_studies|227917 |  
|us_history |38895 |  
+-----+-----+  
Time taken: 15603 ms  
  
scala> val scType = spark.sql("select count(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics'");  
scType: org.apache.spark.sql.DataFrame = [countTotal: bigint]  
  
scala> spark.time(scType.show(false));  
[Stage 26:=====] (2 + 2) / 5
```

## Total Count of data year wise

```
select release_year, count(*) as TotalCount from tbl_star2012_2019 group by  
release_year;
```

```

scala> val scType = spark.sql("select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year");
scType: org.apache.spark.sql.DataFrame = [release_year: string, TotalCount: bigint]

scala> spark.time(scType.show(false));
+-----+-----+
|release_year|TotalCount|
+-----+-----+
|2019        |3524150   |
|2017        |476741    |
|2014        |869282   |
|2013        |368828   |
|2018        |691662   |
|2015        |319319   |
+-----+-----+

Time taken: 17325 ms

scala> val scType = spark.sql("select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year order by TotalCount DESC");
scType: org.apache.spark.sql.DataFrame = [release_year: string, TotalCount: bigint]

scala> spark.time(scType.show(false));
+-----+-----+
|release_year|TotalCount|
+-----+-----+
|2019        |3524150   |
|2014        |869282   |
|2018        |691662   |
|2017        |476741   |
|2013        |368828   |
|2015        |319319   |
+-----+-----+

```

## Total Star Students Learning Mathematics -- Count, Where Mathematics, Result -> 2104942

select count(\*) as countTotal from tbl\_star2012\_2019 where subject = 'mathematics'

```

scala> val scType = spark.sql("select subject, count(*) as countTotal from tbl_star2012_2019 group by subject");
scType: org.apache.spark.sql.DataFrame = [subject: string, countTotal: bigint]

scala> spark.time(scType.show(false));
+-----+-----+
|subject      |countTotal|
+-----+-----+
|reading      |2279045   |
|english_2    |39146    |
|algebra_1    |83457    |
|science      |693697   |
|english_1    |42552    |
|mathematics  |2104942   |
|biology      |53673    |
|writing      |686658   |
|social_studies|227917   |
|us_history   |38895    |
+-----+-----+

Time taken: 15603 ms

scala> val scType = spark.sql("select count(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics'");
scType: org.apache.spark.sql.DataFrame = [countTotal: bigint]

scala> spark.time(scType.show(false));
+-----+
|countTotal|
+-----+
|2104942  |
+-----+

Time taken: 12686 ms

scala> val scType = spark.sql("select distinct release_year from tbl_star2012_2019");

```

## Count female students learning maths -> 221017

select count(\*) from tbl\_star2012\_2019 where demog = 'female' and subject = 'mathematics';

```
docker-hadoop-spark-workbench — com.docker.cli - docker exec -it spark-master /bin/sh — 138x33

scala> spark.time(scType.show(false));
+-----+
|count(1)|
+-----+
|672545 |
+-----+

Time taken: 12846 ms

scala>

scala> val scType = spark.sql("select count(*) from tbl_star2012_2019 where demog = 'female' and subject = 'mathematics'");
scType: org.apache.spark.sql.DataFrame = [count(1): bigint]

scala> spark.time(scType.show(false));
+-----+
|count(1)|
+-----+
|221017 |
+-----+

Time taken: 13316 ms

scala> val scType = spark.sql("select count(*) from tbl_star2012_2019 where demog = 'male'");
scType: org.apache.spark.sql.DataFrame = [count(1): bigint]

scala> spark.time(scType.show(false));
+-----+
|count(1)|
+-----+
|670664 |
+-----+
```

## Following Queries were Executed on Spark

### 1. Distinct Subject

select distinct subject from tbl\_star2012\_2019

35.538 ms

### 2. Count, Group by subjects

select subject, count(\*) as countTotal from tbl\_star2012\_2019 group by subject;

15.603 ms

### 3. Total Star Students Learning Mathematics -- Count, Where Mathematics, Result -> 21.04942

select count(\*) as countTotal from tbl\_star2012\_2019 where subject = 'mathematics'

12.686 ms

### 4. Check which years data we have

select distinct release\_year from tbl\_star2012\_2019

13.914 ms

### 5. Total Count of data year wise

select release\_year, count(\*) as TotalCount from tbl\_star2012\_2019 group by release\_year;

17.325 ms

6. which year have most data

```
select release_year, count(*) as TotalCount from tbl_star2012_2019 group by  
release_year order by TotalCount DESC;
```

15.701 ms

7. Look at demographics of Star students

```
select distinct demog from tbl_star2012_2019;
```

14.060 ms

8. Count female students -> 672545

```
select count(*) from tbl_star2012_2019 where demog = 'female';
```

12.846 ms

9. Count female students learning maths -> 221017

```
select count(*) from tbl_star2012_2019 where demog = 'female' and subject =  
'mathematics';
```

13.316 ms

10. Count of male students

```
select count(*) from tbl_star2012_2019 where demog = 'male';
```

13.167 ms

11. Count of male learning biology -> 6403

```
select count(*) from tbl_star2012_2019 where demog = 'male' and subject =  
'biology'
```

17.000 ms

12. Count of asian students in year 2019 -> 63611

```
select count(*) from tbl_star2012_2019 where demog = 'asian' and  
release_year = '2019';
```

14.073 ms

13. select which subject is mostly learned

```
select subject, count(*) as TotalCount from tbl_star2012_2019 group by  
subject order by TotalCount DESC
```

16.680 ms

# Following CSV file was generated for Performance Comparison between above Three

The screenshot shows a Microsoft Excel spreadsheet titled "DrillVsHiveVsSpark". The table has four columns: A, B, C, and D. Column A contains 14 SQL queries. Columns B, C, and D contain execution times for Drill, Hive, and Spark respectively. The data is as follows:

A	B	C	D
1 Query	9.195	15.757	35.538
2 select distinct subject from tbl_star2012_2019;	11.499	9.498	15.603
3 select subject, count(*) as countTotal from tbl_star2012_2019 group by subject;	14.037	7.788	12.686
4 select count(*) as countTotal from tbl_star2012_2019 where subject = 'mathematics'	9.027	14.804	13.914
5 select distinct release_year from tbl_star2012_2019;	8.57	14.692	17.325
6 select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year;	8.587	16.159	15.701
7 select release_year, count(*) as TotalCount from tbl_star2012_2019 group by release_year order by TotalCount DESC;	7.234	8.727	14.06
8 select distinct demog from tbl_star2012_2019;	6.655	6.689	12.846
9 select count(*) from tbl_star2012_2019 where demog = 'female';	8.72	8.935	13.316
10 select count(*) from tbl_star2012_2019 where demog = 'female' and subject = 'mathematics';	6.929	6.953	13.167
11 select count(*) from tbl_star2012_2019 where demog = 'male';	9.033	7.642	17
12 select count(*) from tbl_star2012_2019 where demog = 'male' and subject = 'biology'	8.036	7.776	14.073
13 select count(*) from tbl_star2012_2019 where demog = 'asian' and release_year = '2019';	7.505	11.293	16.68
14 select subject, count(*) as TotalCount from tbl_star2012_2019 group by subject order by TotalCount DESC			
15			

## Docker Compose File for Apache Hive and Spark

```
version: '2'
services:
  namenode:
    image: bde2020/hadoop-namenode:1.1.0-hadoop2.8-java8
    container_name: namenode
    volumes:
      - ./data/namenode:/hadoop/dfs/name
    environment:
      - CLUSTER_NAME=test
    env_file:
      - ./hadoop-hive.env
    ports:
      - 50070:50070
  datanode:
    image: bde2020/hadoop-datanode:1.1.0-hadoop2.8-java8
    depends_on:
      - namenode
    volumes:
      - ./data/datanode:/hadoop/dfs/data
    env_file:
      - ./hadoop-hive.env
    ports:
      - 50075:50075
  hive-server:
    image: bde2020/hive:2.1.0-postgresql-metastore
    container_name: hive-server
    env_file:
      - ./hadoop-hive.env
    environment:
      - "HIVE_CORE_CONF_javax_jdo_option_ConnectionURL=jdbc:postgresql://hive-metastore/metastore"
    ports:
      - "10000:10000"
  hive-metastore:
    image: bde2020/hive:2.1.0-postgresql-metastore
    container_name: hive-metastore
    env_file:
      - ./hadoop-hive.env
    command: /opt/hive/bin/hive --service metastore
  hive-metastore-postgresql:
    image: bde2020/hive-metastore-postgresql:2.1.0
  spark-master:
    image: bde2020/spark-master:2.1.0-hadoop2.8-hive-java8
    container_name: spark-master
    ports:
      - 8080:8080
      - 7077:7077
    env_file:
```

# Docker Compose for Apache Drill

```
namenode-1:
  container_name: namenode-1
  networks: ["vnet"]
  hostname: namenode-1.vnet
  image: smizy/hadoop-base:2.7.7-alpine
  expose: ["8020"]
  ports: ["50070"]
  environment:
    - SERVICE_8020_NAME=namenode
    - SERVICE_50070_IGNORE=true
    - HADOOP_ZOOKEEPER_QUORUM=zookeeper-1.vnet:2181,zookeeper-2.vnet:2181,zookeeper-3.vnet:2181
    - HADOOP_HEAPSIZE=1000

  entrypoint: entrypoint.sh
  command: namenode-1

namenode-2:
  container_name: namenode-2
  networks: ["vnet"]
  hostname: namenode-2.vnet
  image: smizy/hadoop-base:2.7.7-alpine
  expose: ["8020"]
  ports: ["50070"]
  environment:
    - SERVICE_8020_NAME=namenode
    - SERVICE_50070_IGNORE=true
    - HADOOP_ZOOKEEPER_QUORUM=zookeeper-1.vnet:2181,zookeeper-2.vnet:2181,zookeeper-3.vnet:2181
    - HADOOP_HEAPSIZE=1000

  entrypoint: entrypoint.sh
  command: namenode-2

datanode-1:
  container_name: datanode-1
  networks: ["vnet"]
  hostname: datanode-1.vnet
  image: smizy/hadoop-base:2.7.7-alpine
  expose: ["50010", "50020", "50075"]
  environment:
    - SERVICE_50010_NAME=datanode
    - SERVICE_50020_IGNORE=true
    - SERVICE_50075_IGNORE=true
    - HADOOP_ZOOKEEPER_QUORUM=zookeeper-1.vnet:2181,zookeeper-2.vnet:2181,zookeeper-3.vnet:2181
    - HADOOP_HEAPSIZE=1000

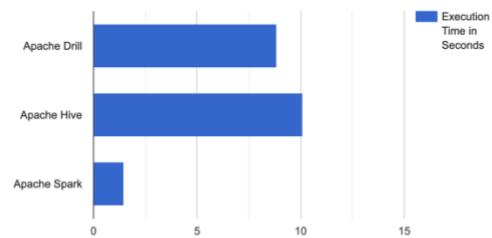
  entrypoint: entrypoint.sh
  command: datanode
```

# Project Front End using Python & Flask

## Performance Comparison Between Apache Hive, Apache Drill and Spark

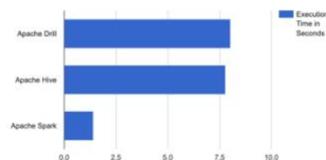
Total Queries Executed: 13

Query: Average Query Time of All Queries on Particular tools

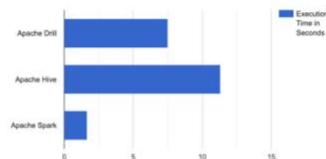


Query: select distinct subject from tbl\_star2012\_2019;

Query: select count(\*) from tbl\_star2012\_2019 where demog = 'asian' and release\_year = '2019';



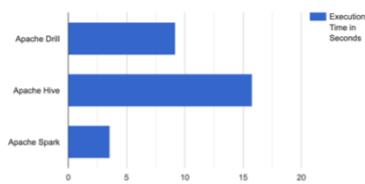
Query: select subject count(\*) as TotalCount from tbl\_star2012\_2019 group by subject order by TotalCount DESC



@ Project Made By M.Ibrahim Arain - ERP: 15976 – IBA Karachi

---

Query: select distinct subject from tbl\_star2012\_2019;



Query: select subject count(\*) as countTotal from tbl\_star2012\_2019 group by subject;

