Big Data Architecture Lab 4 Report

Task 1: querying MongoDB and saving results in Apache Parquet file format

1. Set up information

   First, we open a terminal 1 and start service:

   cd /Users/cherilyn/Downloads/apache-drill-1.16.0

   bin/drill-embedded

   Then open another terminal 2 and start the mongo server:

   cd /Users/cherilyn/Downloads/lab4_drill/

   mongod --dbpath ./server_1 --port 27018

   Open another terminal 3 and import the file:

   cd /Users/cherilyn/Downloads/lab4_drill/

   mongoimport -c egalite --port 27018 ./structures-egalite-femmeshommes.json --jsonArray

   After that all the commands are run in terminal 1.

   show databases;

   USE mongo.test;

   SHOW TABLES;

```
apache drill> show databases;
+---------------------+
|    SCHEMA_NAME      |
+---------------------+
| cp.default          |
| dfs.default         |
| dfs.root            |
| dfs.tmp             |
| information_schema  |
| mongo.admin         |
| mongo.config        |
| mongo.local         |
| mongo.test          |
| sys                 |
+---------------------+
10 rows selected (0.204 seconds)
apache drill> USE mongo.test;
+-------+-----------------------------------------+
|  ok   |                summary                  |
+-------+-----------------------------------------+
| true  | Default schema changed to [mongo.test]  |
+-------+-----------------------------------------+
1 row selected (0.096 seconds)
apache drill (mongo.test)> SHOW TABLES;
+--------------+-------------+
| TABLE_SCHEMA | TABLE_NAME  |
+--------------+-------------+
| mongo.test   | zips        |
| mongo.test   | egalite     |
+--------------+-------------+
2 rows selected (0.393 seconds)
```

2. The json file has been imported.

3. SELECT e.fields.code_postal AS zip_code, count(*) AS num
   FROM egalite e
   WHERE e.fields.commune = 'Toulouse'
   GROUP BY zip_code
   ORDER BY num DESC;

```
apache drill (mongo.test)> SELECT e.fields.code_postal as zip_code, count(*) AS num
. . . . . . . semicolon> FROM egalite e
. . . . . . . semicolon> WHERE e.fields.commune = 'Toulouse'
. . . . . . . semicolon> GROUP BY zip_code
. . . . . . . semicolon> ORDER BY num DESC;
+-----------+-----+
| zip_code  | num |
+-----------+-----+
| null      | 26  |
| 31100     | 21  |
| 31000     | 16  |
| 31400     | 15  |
| 31300     | 13  |
| 31200     | 13  |
| 31500     | 10  |
+-----------+-----+
7 rows selected (0.264 seconds)
```

4. From the result we can see there are 26 organizations whose zip codes are null. So, the zip codes data isn't complete.

5. CREATE TABLE dfs.tmp.query AS
   (SELECT e.fields.code_postal AS zip_code, count(*) AS num
   FROM egalite e
   WHERE e.fields.commune = 'Toulouse'
   GROUP BY zip_code
   ORDER BY num DESC
   );

6. SELECT* FROM dfs.tmp.query;

```
apache drill (mongo.test)> CREATE TABLE dfs.tmp.query AS
. . . . . semicolon> (SELECT e.fields.code_postal AS zip_code, count(*) AS num
. . . . . )> FROM egalite e
. . . . . )> WHERE e.fields.commune = 'Toulouse'
. . . . . )> GROUP BY zip_code
. . . . . )> ORDER BY num DESC
. . . . . )> );
+----------+---------------------------+
| Fragment | Number of records written |
+----------+---------------------------+
| 0_0      | 7                         |
+----------+---------------------------+
1 row selected (1.329 seconds)
apache drill (mongo.test)> select * from dfs.tmp.query;
+----------+-----+
| zip_code | num |
+----------+-----+
| null     | 26  |
| 31100    | 21  |
| 31000    | 16  |
| 31400    | 15  |
| 31300    | 13  |
| 31200    | 13  |
| 31500    | 10  |
+----------+-----+
7 rows selected (0.392 seconds)
```

Task 2: importing data in CSV and joining with data in Postgres

1. Open another terminal 4 to start Postgres:
   pg_ctl -D /usr/local/var/postgres -l /usr/local/var/postgres/server.log start

   We create a user named “root” and set a password:
   create user test superuser password '961011';

   And we create a data base named “data”:
   createdb data -O root -E UTF8 -e;

   We go to this database:
   psql -U root -d data -h 127.0.0.1;

   Then create a table:
   CREATE TABLE crime(INCIDENT_NUMBER text, OFFENSE_CODE int, OFFENSE_CODE_GROUP text, OFFENSE_DESCRIPTION text, DISTRICT text, REPORTING_AREA text, SHOOTING text, OCCURRED_ON_DATE text, YEAR int, MONTH int, DAY_OF_WEEK text, HOUR int, UCR_PART text, STREET text, Lat double precision, Long double precision, Location text);

   Copy csv file to this table:
   COPY crime FROM '/Users/cherilyn/Downloads/lab4_drill/boston-crime-incident-reports-10k.csv' CSV HEADER;

2. Set up Postgres plugin

Configuration

```
{
  "type": "jdbc",
  "driver": "org.postgresql.Driver",
  "url": "jdbc:postgresql://127.0.0.1/data",
  "username": "root",
  "password": "961011",
  "caseInsensitiveTableNames": false,
  "enabled": true
}
```

3. We go back to terminal 1 and find the table

```
[apache drill (postgres)> show databases;
+---------------------------+
|        SCHEMA_NAME        |
+---------------------------+
| cp.default                |
| dfs.default               |
| dfs.root                  |
| dfs.tmp                   |
| information_schema        |
| postgres.data             |
| postgres.information_schema |
| postgres.pg_catalog       |
| postgres.public           |
| postgres                  |
| sys                       |
+---------------------------+
11 rows selected (30.162 seconds)
[apache drill (postgres)> use postgres.public;
+------+-----------------------------------------------+
|  ok  |                    summary                    |
+------+-----------------------------------------------+
| true | Default schema changed to [postgres.public]   |
+------+-----------------------------------------------+
1 row selected (0.097 seconds)
[apache drill (postgres.public)> show tables;
+-----------------+------------+
|  TABLE_SCHEMA   | TABLE_NAME |
+-----------------+------------+
| postgres.public | crime      |
+-----------------+------------+
1 row selected (30.159 seconds)
```

Now we can see the content of the dataset loaded to Postgres:

SELECT * FROM crime LIMIT 5;

```
apache drill (postgres.public)> SELECT * FROM crime LIMIT 5;
+-----------------+--------------+-----------------------+---------------------+------+-------+-------------+------+-------------+
| incident_number | offense_code |  offense_code_group   |  offense_description | district | reporting_area | shooting | occurred_on_date | year | month | day_of_week | hour | ucr_part | street | lat | long | location |
+-----------------+--------------+-----------------------+---------------------+------+-------+-------------+------+-------------+
| I192078648      | 3114         | Investigate Property  | INVESTIGATE PROPERTY | B3 | 427 | | 2019-09-29 06:39:00 | 2019 | 9 | Sunday | 6 | Part Three | WILMORE ST | 42.2779637 | -71.09246318 | (42.27796370, -71.09246318) |
| I192078647      | 3115         | Investigate Person    | INVESTIGATE PERSON | A1 | | | 2019-09-29 03:45:00 | 2019 | 9 | Sunday | 3 | Part Three | NASHUA ST | 42.36769032 | -71.06586347 | (42.36769032, -71.06586347) |
| I192078645      | 3301         | Verbal Disputes       | VERBAL DISPUTE | B3 | 450 | | 2019-09-29 06:00:00 | 2019 | 9 | Sunday | 6 | Part Three | ASPINWALL RD | 42.2918158 | -71.07244098 | (42.29181580, -71.07244098) |
| I192078642      | 3820         | Motor Vehicle Accident Response | M/V ACCIDENT INVOLVING PEDESTRIAN - INJURY | D4 | 269 | | 2019-09-29 05:50:00 | 2019 | 9 | Sunday | 5 | Part Three | ALBANY ST | null | null | (0.00000000, 0.00000000) |
| I192078640      | 3115         | Investigate Person    | INVESTIGATE PERSON | A7 | 28 | | 2019-09-29 01:30:00 | 2019 | 9 | Sunday | 1 | Part Three | PARIS ST | 42.37339168 | -71.03647779 | (42.37339168, -71.03647779) |
+-----------------+--------------+-----------------------+---------------------+------+-------+-------------+------+-------------+
5 rows selected (0.202 seconds)
```

And here is another query to find out the number of incidents in every day of week:

SELECT day_of_week, count(DISTINCT incident_number) as num

FROM crime

GROUP BY day_of_week;

```
apache drill (postgres.public)> SELECT day_of_week, count(DISTINCT incident_number) as num
. . . . . .semicolon> FROM crime
. . . . . .semicolon> GROUP BY day_of_week;
+-------------+------+
| day_of_week | num  |
+-------------+------+
| Friday      | 1405 |
| Monday      | 1189 |
| Saturday    | 1238 |
| Sunday      | 1006 |
| Thursday    | 1299 |
| Tuesday     | 1378 |
| Wednesday   | 1406 |
+-------------+------+
7 rows selected (0.512 seconds)
```

4. First, I didn't change the dfs plugin, the result is below:

```
apache drill (postgres.public)> SELECT *
. . . . . .semicolon> FROM dfs.`/Users/cherilyn/Downloads/lab4_drill
/boston-offense-codes-lookup.csv`
. . . . . .semicolon> LIMIT 10;
+-----------------------------------------------------------+
|                          columns                          |
+-----------------------------------------------------------+
| ["CODE","NAME\r"]                                         |
| ["612","LARCENY PURSE SNATCH - NO FORCE \r"]              |
| ["613","LARCENY SHOPLIFTING\r"]                           |
| ["615","LARCENY THEFT OF MV PARTS & ACCESSORIES\r"]       |
| ["1731","INCEST\r"]                                       |
| ["3111","LICENSE PREMISE VIOLATION\r"]                    |
| ["2646","LIQUOR - DRINKING IN PUBLIC\r"]                  |
| ["2204","LIQUOR LAW VIOLATION\r"]                         |
| ["3810","M/V ACCIDENT - INVOLVING  BICYCLE - INJURY\r"]   |
| ["3801","M/V ACCIDENT - OTHER\r"]                         |
+-----------------------------------------------------------+
10 rows selected (0.38 seconds)
```

I found that the code and name were together. So, I changed the dfs plugin. Put "extractHeader": true in csv in order to set the first line of csv file as the column name.

```
    "csv": {
      "type": "text",
      "extensions": [
        "csv"
      ],
      "extractHeader": true,
      "delimiter": ","
    },
```

Then I ran again this query:

SELECT *

FROM dfs.`/Users/cherilyn/Downloads/lab4_drill/boston-offense-codes-lookup.csv`

LIMIT 10;

```
apache drill (postgres.public)> SELECT *
. . . . . .semicolon> FROM dfs.`/Users/cherilyn/Downloads/lab4_drill/boston-offense-codes-lookup.csv`
. . . . . .semicolon> LIMIT 10;
+----------+--------------------------------------------------+
| col_CODE |                       NAME                       |
+----------+--------------------------------------------------+
          | LARCENY PURSE SNATCH - NO FORCE
                    |LIFTING
    |     | LARCENY THEFT OF MV PARTS & ACCESSORIES
                                |
                |E PREMISE VIOLATION
               |OR - DRINKING IN PUBLIC
                     |VIOLATION
 |3810     | M/V ACCIDENT - INVOLVING  BICYCLE - INJURY
                     |T - OTHER
         || M/V ACCIDENT - OTHER CITY VEHICLE
+----------+--------------------------------------------------+
10 rows selected (0.159 seconds)
```

Maybe because of the length of the name, it became strange. I tried a lot of solution but I still didn't succeed with it.

5. SELECT DISTINCT c.street
   FROM crime AS c, dfs.`/Users/cherilyn/Downloads/lab4_drill/boston-offense-codes-lookup.csv` AS l
   WHERE c.day_of_week = 'Monday'
   AND c.offense_code = CAST(l.col_CODE AS int)
   AND l.NAME LIKE '%FIRE%';

```
apache drill (postgres.public)> SELECT DISTINCT c.street
. . . . . .semicolon> FROM crime AS c, dfs.`/Users/cherilyn/Downloads/lab4_drill/boston-offense-codes-lookup.csv` AS l
. . . . . .semicolon> WHERE c.day_of_week = 'Monday'
. . . . . .semicolon> AND c.offense_code = CAST(l.col_CODE AS int)
. . . . . .semicolon> AND l.NAME like '%FIRE%';
+--------------------+
|       street       |
+--------------------+
| RIVER ST           |
| STRATTON ST        |
| METROPOLITAN AVE   |
| FAWNDALE RD        |
| TOVAR ST           |
| CAMBRIDGE ST       |
| ROWES WHRF         |
| MORTON ST          |
| PARKER ST          |
| GALLIVAN BLVD      |
| E INDIA ROW        |
|                    |
| BRIGHTON AVE       |
| ADAMS ST           |
| HENRY STERLING SQ  |
| CENTRE ST          |
| DUDLEY ST          |
| BROOKLINE AVE      |
| HARRISON AVE       |
| HAMMOND ST         |
| WASHINGTON ST      |
| BEACON ST          |
| CALLENDER ST       |
| BORDER ST          |
| W CONCORD ST       |
| ATLANTIC AVE       |
| NEWTON ST          |
| DALTON ST          |
| TREMONT ST         |
| LYFORD ST          |
+--------------------+
30 rows selected (0.827 seconds)
```