Big Data Architecture Lab 4

ZHANG Xin

Table of Contents

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

1. Set up MongoDB plugin

Below are the example quries from the toturials.

```
apache drill (mongo.test)> SELECT * FROM zips LIMIT 10;
```

```
+-------+--------------+-----------------------+---------+-------+
|  _id  |     city     |          loc          |   pop   | state |
+-------+--------------+-----------------------+---------+-------+
| 01007 | BELCHERTOWN  | [-72.410953,42.275103] | 10579.0 | MA    |
| 01001 | AGAWAM       | [-72.622739,42.070206] | 15338.0 | MA    |
| 01008 | BLANDFORD    | [-72.936114,42.182949] | 1240.0  | MA    |
| 01010 | BRIMFIELD    | [-72.188455,42.116543] | 3706.0  | MA    |
| 01011 | CHESTER      | [-72.988761,42.279421] | 1688.0  | MA    |
| 01012 | CHESTERFIELD | [-72.833309,42.38167]  | 177.0   | MA    |
| 01020 | CHICOPEE     | [-72.576142,42.176443] | 31495.0 | MA    |
| 01002 | CUSHMAN      | [-72.51565,42.377017]  | 36963.0 | MA    |
| 01022 | WESTOVER AFB | [-72.558657,42.196672] | 1764.0  | MA    |
```

```
| 01027 | MOUNT TOM    | [-72.679921,42.264319] | 16864.0 | MA    |
+-------+--------------+------------------------+---------+-------+
```

```
apache drill (mongo.test)> SELECT city, avg(pop) FROM zips GROUP BY city LIMIT 10;
```

```
+--------------+--------------------+
|     city     |       EXPR$1       |
+--------------+--------------------+
| BELCHERTOWN  | 10579.0            |
| AGAWAM       | 15338.0            |
| BLANDFORD    | 1240.0             |
| BRIMFIELD    | 2441.5             |
| CHESTER      | 7285.0952380952385 |
| CHESTERFIELD | 9988.857142857143  |
| CHICOPEE     | 27445.5            |
| CUSHMAN      | 18649.5            |
| WESTOVER AFB | 1764.0             |
| MOUNT TOM    | 16864.0            |
```

2. Import `structures-egalite-femmeshommes.json` dataset into MongoDB.

```
> mongoimport --db sef --file structures-egalite-femmeshommes.json --collection
sef --port 27017 --jsonArray
connected to: mongodb://localhost:27017/
114 document(s) imported successfully. 0 document(s) failed to import.
```

3. the number of organizations working for gender equality in Toulouse by their zip code in the descending order of size.

```
apache drill (mongo.sef)> SELECT s.fields.code_postal AS zip, count(*) AS count
FROM sef s WHERE s.fields.commune = 'Toulouse' GROUP BY s.fields.code_postal ORDER
BY count DESC;
```

```
+---------+-------+
|   zip   | count |
+---------+-------+
| null    | 26    |
| 31100.0 | 21    |
| 31000.0 | 16    |
| 31400.0 | 15    |
| 31300.0 | 13    |
| 31200.0 | 13    |
```

```
| 31500.0 | 10    |
+---------+-------+
7 rows selected (0.362 seconds)
```

4. Analyze the result of the aggregation query, Is the organizations' zip codes data complete?

We can see that there is a `null` value, so the organizations' zip codes data are not complete.

5. Save the result of the query into a Parquet file in `tmp` workspace using a default `dfs` plugin.

```
apache drill (mongo.sef)> alter session set `store.format`='parquet';
```

```
+------+-----------------------+
|  ok  |        summary        |
+------+-----------------------+
| true | store.format updated. |
+------+-----------------------+
1 row selected (0.088 seconds)
```

```
apache drill (mongo.sef)> CREATE TABLE dfs.tmp.`/stats/airport_data/` AS
. . . . . .semicolon> SELECT s.fields.code_postal AS `zip`, count(*) AS
`count` FROM sef s WHERE s.fields.commune = 'Toulouse' GROUP BY
s.fields.code_postal ORDER BY count DESC;
```

```
+----------+--------------------------+
| Fragment | Number of records written |
+----------+--------------------------+
| 0_0      | 7                        |
+----------+--------------------------+
1 row selected (1.711 seconds)
```

6. Run a query to display the content of the Parquet file.

```
apache drill (mongo.sef)> SELECT * FROM
dfs.`C:/tmp/stats/airport_data/0_0_0.parquet`;
```

```
+---------+-------+
|   zip   | count |
+---------+-------+
| null    | 26    |
```

```
| 31100.0 | 21    |
| 31000.0 | 16    |
| 31400.0 | 15    |
| 31300.0 | 13    |
| 31200.0 | 13    |
| 31500.0 | 10    |
+---------+-------+
7 rows selected (0.401 seconds)
```

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

1. Import boston-crime-incident-reports-10k.csv dataset into Postgres.

```
postgres=# CREATE DATABASE reports;
CREATE DATABASE
```

```
postgres=$ \connect reports
Vous êtes maintenant connecté à la base de données « reports » en tant
qu'utilisateur « postgres ».
```

```
postgres=$ CREATE TABLE reports(incident_number varchar NOT NULL, offense_code
varchar, offense_code_group varchar, offense_description varchar, district
varchar, reporting_area varchar, shooting varchar, occurred_on_date varchar, year
varchar, month varchar, day_of_week varchar, hour varchar, ucr_part varchar,
street varchar, lat varchar, long varchar, location varchar);
CREATE TABLE
```

```
postgres=% COPY reports FROM
'C:\Temp\Dossiers_Cours\Big_Data_Architectures\tp4\lab_3_dataset\boston-crime-
incident-reports-10k.csv' WITH CSV HEADER;
COPY 9999
```

The `CSV HEADER` option is used in order to ignore the first line headers.

2. Set up Postgres plugin

```
{
  "type": "jdbc",
  "driver": "org.postgresql.Driver",
  "url": "jdbc:postgresql://localhost:5432/reports",
  "username": "postgres",
  "password": "0122",
```

```
  "caseInsensitiveTableNames": false,
  "enabled": true
}
```

3. Run a query to display the content of the dataset

```
apache drill> SELECT * FROM psql.public.reports 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    |
|              | (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 (0.303 seconds)
```

4. Run a query to display the content of `boston-offense-codes-lookup.csv` file in Apache Drill(without loading it to Postgres).

Before start, I've modified the configuration of `dfs` on the way to read `csv` files in order to ignore the first HEADER line.

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

Here I've added `"skipFirstLine": true` Then the query

```
apache drill> SELECT columns[0] as code, columns[1] as name FROM
dfs.`C:\Temp\Dossiers_Cours\Big_Data_Architectures\tp4\lab_3_dataset\boston-
offense-codes-lookup.csv` LIMIT 5;
```

```
+------+----------------------------------------+
| code |                  name                  |
+------+----------------------------------------+
| 612  | LARCENY PURSE SNATCH - NO FORCE         |
| 613  | LARCENY SHOPLIFTING                     |
| 615  | LARCENY THEFT OF MV PARTS & ACCESSORIES |
| 1731 | INCEST                                  |
| 3111 | LICENSE PREMISE VIOLATION               |
+------+----------------------------------------+
5 rows selected (0.296 seconds)
```

5. Find all the distinct street names mentioned in reports such that their code name in a lookup CSV file contains "FIRE" and they refer to Monday.

So here I used two `WITH` clause to get the two tables stored in different places, and then I have run my query on top of them.

```
apache drill> WITH l AS (SELECT columns[0] as code, columns[1] as name FROM
dfs.`C:\Temp\Dossiers_Cours\Big_Data_Architectures\tp4\lab_3_dataset\boston-
offense-codes-lookup.csv` ), r AS (SELECT * FROM psql.public.reports) SELECT
DISTINCT r.street FROM l, r WHERE r.offense_code = l.code AND l.name LIKE '%FIRE%'
AND r.day_of_week = 'Monday' AND r.street IS NOT NULL;
```

```
+------------------+
|      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 (0.791 seconds)
```