IST - Labo 6

Paul Gillet & Kevin Auberson
Groupe : L06GrH
Date : 15.05.2024

TASK 1: EXPLORE NEW YORK CITY TAXI TRIP DATA

> Navigate to the TLC Trip Record Data website. The taxi commission publishes data on four types of cabs. Which are they?

- **Yellow Taxi Trip Records** (PARQUET)
- **Green Taxi Trip Records** (PARQUET)
- **For-Hire Vehicle Trip Records** (PARQUET)
- **High Volume For-Hire Vehicle Trip Records** (PARQUET)

> Find the PDF file with the data dictionary for the yellow cab data on web site. Does it contain the data types?

No, the type information is missing, there is the name field with a description.

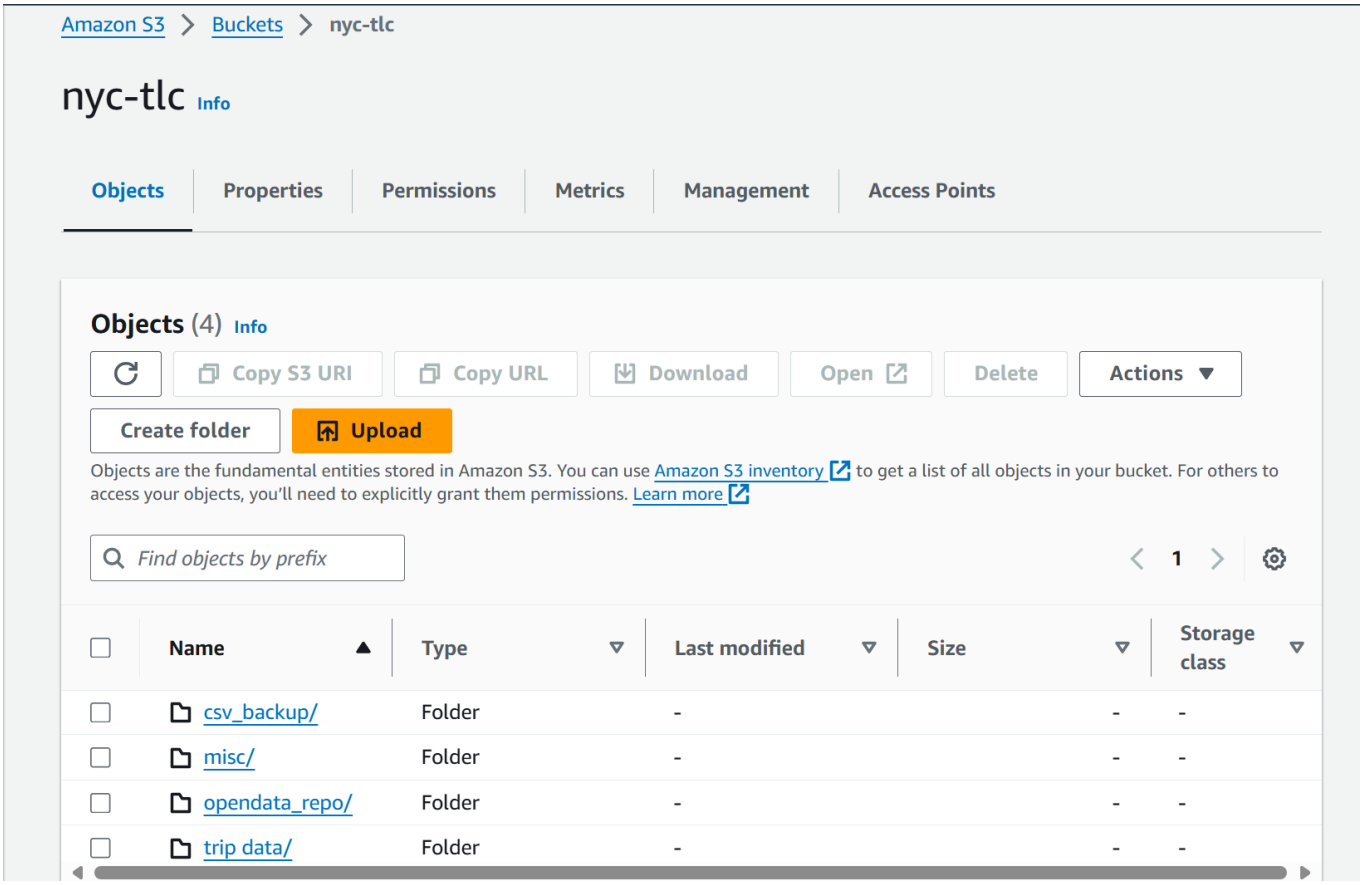
link to the PDF

> The yellow cab data is available in what types of files?

The data is available in the type .parquet

> Find the copy of the data product in the Registry of Open Data on AWS. What is the bucket name? In which region is the bucket? Open the bucket in the S3 console.

Name of the bucket: nyc-tlc
Bucket's region: us-east-1



> In this lab we are going to use the yellow cab trip data. In which folder are the CSV files for yellow cabs? Does this folder only contain yellow cab data? In which folder are the Parquet files for yellow cabs? Does this folder only contain yellow cab data?

The CSVs are stored in the `opendata_repo/opendata_webconvert` folder which contain a folder for all 4 cab types. For the Parquet files, those are stored into the trip data folder, it also contains parquet files for all 4 types of cab

> Is Amazon's copy up-to-date compared to the original data product?

No, last update was in december 2022

TASK 2: CREATE AN ENTRY IN THE DATA CATALOG AND QUERY THE DATA

> What did you notice when you looked at the public AWS bucket of the NYC taxi trip data?

The data is organized into several folders, each representing different types of trip data such as yellow taxi trips, green taxi trips, and for-hire vehicles (FHV).
The data files are primarily in CSV format, which is widely used and easy to import into various data analysis tools and programming languages.
Some folders might also contain other formats such as Parquet, which is optimized for

performance and space efficiency, especially with big data processing frameworks like Apache Spark.

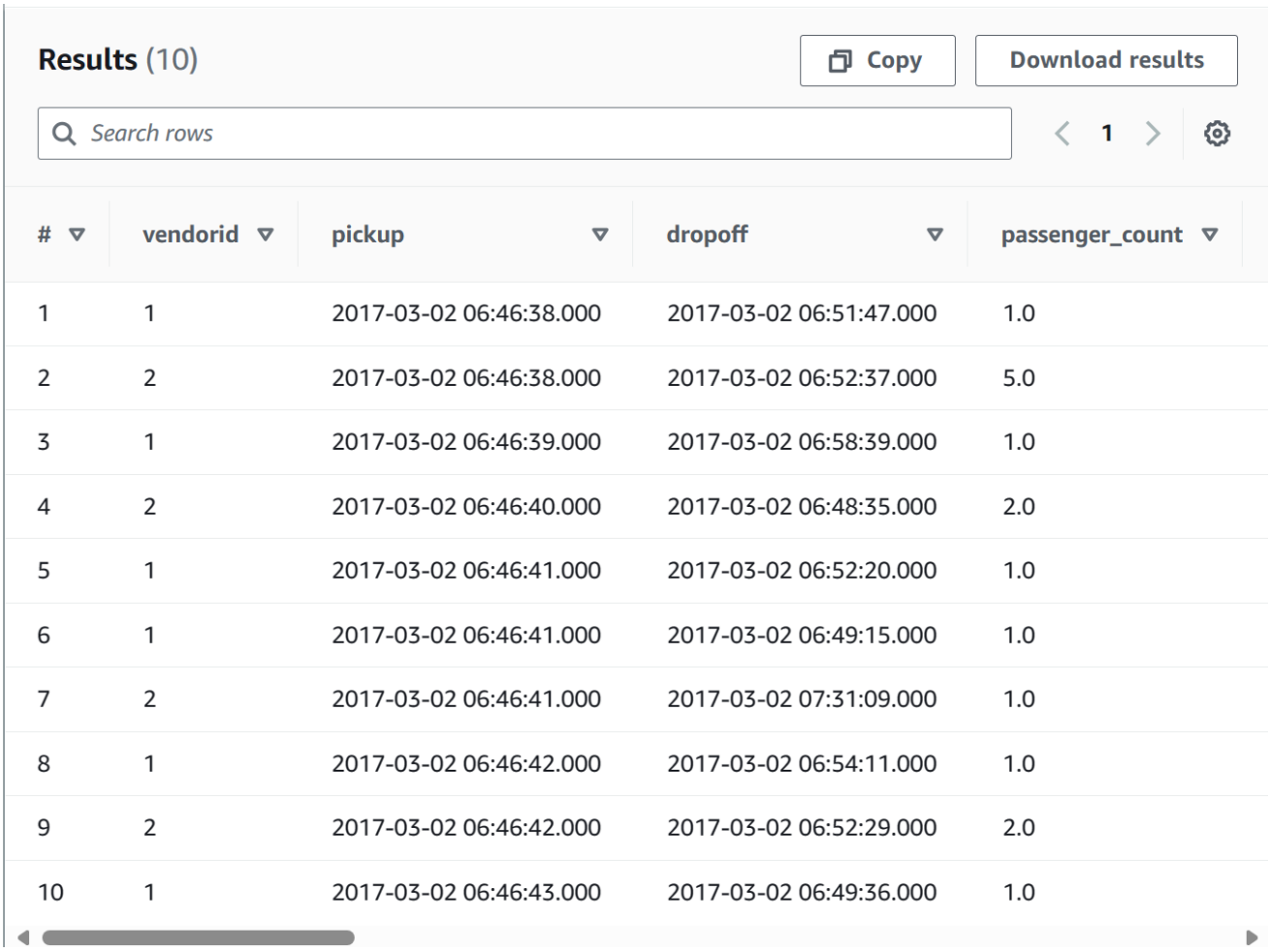
> What subset of the data does the folder contain? In what format?

It contains all datas for yellow cabs during the year 2017 in the CSV format

> In the taxidata_grf database, create a new table yellow for the yellow cab data for 2017 in CSV format that is stored at s3://aws-tc-largeobjects/CUR-TF-200-ACBDFO-1/Lab2/yellow/

```
CREATE EXTERNAL TABLE `yellow`(
  `vendorid` int,
  `pickup` timestamp,
  `dropoff` timestamp,
  `passenger_count` float,
  `trip_distance` float,
  `ratecodeid` float,
  `store_and_fwd_flag` string,
  `pulocationid` int,
  `dolocationid` int,
  `payment_type` int,
  `fare_amount` float,
  `extra` float,
  `mta_tax` float,
  `tip_amount` float,
  `tolls_amount` float,
  `improvement_surcharge` float,
  `total_amount` float,
  `congestion_surcharge` float,
  `airport_fee` float)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://aws-tc-largeobjects/CUR-TF-200-ACBDFO-1/Lab2/yellow'
TBLPROPERTIES (
  'classification'='csv',
  'transient_lastDdlTime'='1714726862')
```

> Run a query that displays the first 10 records of the table



Results (10)

Copy | Download results

Search rows | 1

#	vendorid	pickup	dropoff	passenger_count
1	1	2017-03-02 06:46:38.000	2017-03-02 06:51:47.000	1.0
2	2	2017-03-02 06:46:38.000	2017-03-02 06:52:37.000	5.0
3	1	2017-03-02 06:46:39.000	2017-03-02 06:58:39.000	1.0
4	2	2017-03-02 06:46:40.000	2017-03-02 06:48:35.000	2.0
5	1	2017-03-02 06:46:41.000	2017-03-02 06:52:20.000	1.0
6	1	2017-03-02 06:46:41.000	2017-03-02 06:49:15.000	1.0
7	2	2017-03-02 06:46:41.000	2017-03-02 07:31:09.000	1.0
8	1	2017-03-02 06:46:42.000	2017-03-02 06:54:11.000	1.0
9	2	2017-03-02 06:46:42.000	2017-03-02 06:52:29.000	2.0
10	1	2017-03-02 06:46:43.000	2017-03-02 06:49:36.000	1.0

> Write down the run time and the volume of data scanned.

Run time: 1.629 sec
Data scanned: 1.71 MB

> How much did the last query cost?

5,00 USD par To de données analysées.

La requête ayant fait 1.71 MB = 1.71e-6 TB

Elle a donc couté 1.71e-6 TB * 5$ = 0.00000855$

TASK 3: OPTIMISE THE QUERY BY SCANNING ONLY A PARTITION OF THE DATA

> Create a new table that contains just the data for January 2017 called jan, stored at s3://aws-tc-largeobjects/CUR-TF-200-ACBDFO-1/Lab2/January2017/

```
CREATE EXTERNAL TABLE `jan`(
  `vendorid` int,
```

```
  `pickup` timestamp,
  `dropoff` timestamp,
  `passenger_count` float,
  `trip_distance` float,
  `ratecodeid` float,
  `store_and_fwd_flag` string,
  `pulocationid` int,
  `dolocationid` int,
  `payment_type` int,
  `fare_amount` float,
  `extra` float,
  `mta_tax` float,
  `tip_amount` float,
  `tolls_amount` float,
  `improvement_surcharge` float,
  `total_amount` float,
  `congestion_surcharge` float,
  `airport_fee` float)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://aws-tc-largeobjects/CUR-TF-200-ACBDFO-1/Lab2/January2017/'
TBLPROPERTIES (
  'classification'='csv',
  'transient_lastDdlTime'='1714726862')
```

> Run the following query on the data for the whole of 2017 that summarises trips in January 2017:

Data scanned: 9.32 GB

> Run the following query on the data for January 2017 only:

Data scanned: 815.30 MB

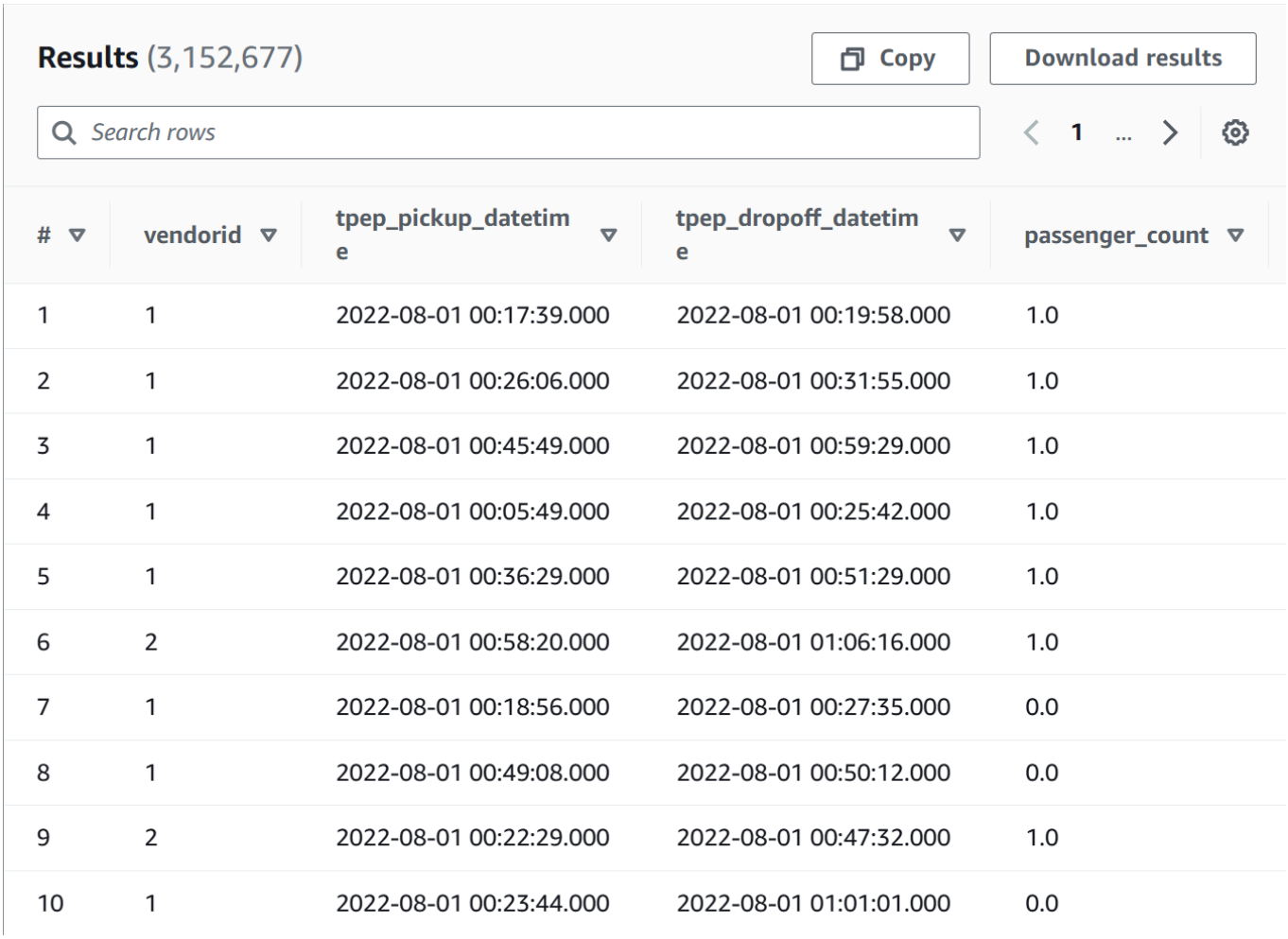
TASK 4: CREATE A PARTITIONED TABLE IN THE DATA CATALOG WITH A GLUE CRAWLER

> What virtual columns were added to the schema?

The columns 'year' and 'month' were added

> In Athena write a query that makes use of the added virtual columns to restrict the scanning of data to August of 2022.

```
SELECT * FROM partyellow WHERE year = '2022' AND month = '08';
```



Results (3,152,677) Copy Download results

Search rows 1 ...

#	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count
1	1	2022-08-01 00:17:39.000	2022-08-01 00:19:58.000	1.0
2	1	2022-08-01 00:26:06.000	2022-08-01 00:31:55.000	1.0
3	1	2022-08-01 00:45:49.000	2022-08-01 00:59:29.000	1.0
4	1	2022-08-01 00:05:49.000	2022-08-01 00:25:42.000	1.0
5	1	2022-08-01 00:36:29.000	2022-08-01 00:51:29.000	1.0
6	2	2022-08-01 00:58:20.000	2022-08-01 01:06:16.000	1.0
7	1	2022-08-01 00:18:56.000	2022-08-01 00:27:35.000	0.0
8	1	2022-08-01 00:49:08.000	2022-08-01 00:50:12.000	0.0
9	2	2022-08-01 00:22:29.000	2022-08-01 00:47:32.000	1.0
10	1	2022-08-01 00:23:44.000	2022-08-01 01:01:01.000	0.0

> How much data was scanned?

Amount of data scanned: 47.40 MB

TASK 5: EXPLORE AND TRANSFORM DATA WITH GLUE DATABREW

For all this question we configure the dataset with a sample First 500 rows.

> What is the largest distance travelled (approximately)?

21.75 miles

> What is the biggest tip given by a passenger (approximately)?

20$

> How big are the generated files?

97.6 MB and 81.3 MB

> Show the schema of the generated Parquet files in the report

We get this schema with the following Python script

```
import pandas as pd

# Read the Parquet file into a DataFrame
df = pd.read_parquet('yellow-tripdata-2022-
gri_07May2024_1715098650560_part00000.parquet')

# Get the DataFrame's schema
schema = df.dtypes

# Print the schema
print(schema)
```

```
VendorID                        int32
tpep_pickup_datetime     datetime64[ns]
tpep_dropoff_datetime    datetime64[ns]
passenger_count                 float32
trip_distance                   float32
RatecodeID                      float32
store_and_fwd_flag               object
PULocationID                      int32
DOLocationID                      int32
payment_type                      int32
fare_amount                     float32
extra                           float32
mta_tax                         float32
tip_amount                      float32
tolls_amount                    float32
improvement_surcharge           float32
total_amount                    float32
congestion_surcharge            float32
dtype: object
```

TASK 6: SCENARIO

First of all we would suggest to have a standardized date format, as we can see on the picture below, we can easily see to which year the data comes from but it's impossible to see when is the year the data is from.

The structure is to change, to implement regional analysis we should divide the data let's say with a 1st layer by continent then by country then by region