Contextualização:

Uma empresa de transportes particular que atua em Nova Iorque reuniu dados de corridas dos meses de maio e junho de 2020 para fazer o levantamento das características principais das corridas realizadas no período. O setor financeiro da empresa solicitou algumas informações:

Quantidade de passageiros por corrida
Valores máximo e mínimo de gorjeta no período
Valor médio da corrida
Média de distância percorrida nos meses de maio e junho

Fonte de Dados:

Os dados foram extraídos do Kaggle (https://www.kaggle.com/datasets/microize/newyork-yellow-taxi-trip-data-2020-2019), em fomato .csv.

Dicionário de Dados:

Descrições de colunas

- VendorID : Identificador do provedor de TPEP que fornece o registro.
  - 1 = Tecnologias Móveis Criativas, LLC
  - 2 = VeriFone Inc.
- tpep_pickup_datetime : A data e a hora em que o medidor foi ativado.
- tpep_dropoff_datetime : A data e a hora em que o medidor foi desligado.
- Passenger_count : O número de passageiros no veículo, conforme inserido pelo motorista.
- Trip_distance : A distância da viagem em milhas, conforme registrada pelo taxímetro.
- PULocationID : Zona de táxi TLC onde o taxímetro foi acionado.
- DOLocationID : Zona de táxi TLC onde o taxímetro foi desativado.
- RateCodeID : O código de tarifa aplicável no final da viagem.
  - 1 = Taxa padrão
  - 2 = JFK
  - 3 = Nova Iorque
  - 4 = Nassau ou Westchester
  - 5 = Tarifa negociada
  - 6 = Passeio em grupo
- Store_and_fwd_flag : indica se o registro da viagem foi armazenado na memória do veículo antes da transmissão ao fornecedor devido à falta de conexão com o servidor.
  - Y = Viagem de armazenamento e encaminhamento
  - N = Não é uma viagem de armazenamento e encaminhamento
- Payment_type : Como o passageiro pagou pela viagem, representado por um código numérico.
  - 1 = Cartão de crédito
  - 2 = Dinheiro

- 3 = Sem custo
- 4 = Disputa
- 5 = Desconhecido
- 6 = Viagem anulada

- Fare_amount : A tarifa calculada pelo taxímetro com base no tempo e na distância.
- Extra : Taxas adicionais, atualmente incluindo apenas as taxas de US$ 0,50 e US$ 1 para o horário de pico e pernoite.
- MTA_tax : Um imposto de US$ 0,50 adicionado automaticamente com base na taxa medida.
- Improvement_surcharge : Uma sobretaxa de US$ 0,30 adicionada no início da viagem, implementada desde 2015.
- Tip_amount : valores de gorjetas de cartão de crédito. (Observação: gorjetas em dinheiro não são registradas aqui.)
- Tolls_amount : Total de pedágios pagos durante a viagem.
- Total_amount : O valor total cobrado dos passageiros, excluindo gorjetas em dinheiro.

Scripts

Script de carga de dados para o HDFS

Os dados foram baixados e carregados no HDFS através do comando hdfs dfs -put na pasta ‘projeto’

```
mariana_cmenezes@projhadoop-m:~$ ls
movies.dat  projeto  yellow_tripdata_2019-06.csv.csupload  yellow_tripdata_2020-06.csv
mariana_cmenezes@projhadoop-m:~$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - mariana_cmenezes hadoop          0 2024-09-11 21:22 projeto
mariana_cmenezes@projhadoop-m:~$ hdfs dfs -put yellow_tripdata_2020-06.csv projeto
mariana_cmenezes@projhadoop-m:~$
```

Script de ingestão de dados no HIVE

1) Foi realizado o comando create database taxi para a criação do database
2) Foram criadas duas tabelas conforme o script abaixo:

CREATE EXTERNAL TABLE tb_dados_taxis (VendorID VARCHAR(3), tpep_pickup_datetime TIMESTAMP, tpep_dropoff_datetime TIMESTAMP, passenger_count SMALLINT, trip_distance DECIMAL(6,3),Rate_code_id SMALLINT, store_and_fwd_flag VARCHAR(1), PULocationID SMALLINT, DOLocationID SMALLINT, payment_type VARCHAR(3), fare_amount DECIMAL(6,2), extra DECIMAL(6,2), mta_tax DECIMAL (6,2), tip_amount decimal(6,2), tolls_amount DECIMAL(6,2), improvement_surcharge DECIMAL(6,2), total_amount DECIMAL(6,2), congestion_surcharge DECIMAL (6,2)) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

CREATE EXTERNAL TABLE tb_dados_taxis_maio (VendorID VARCHAR(3), tpep_pickup_datetime TIMESTAMP, tpep_dropoff_datetime TIMESTAMP, passenger_count SMALLINT, trip_distance DECIMAL(6,3),Rate_code_id SMALLINT, store_and_fwd_flag VARCHAR(1), PULocationID SMALLINT, DOLocationID SMALLINT, payment_type VARCHAR(3), fare_amount DECIMAL(6,2), extra DECIMAL(6,2), mta_tax DECIMAL (6,2), tip_amount decimal(6,2), tolls_amount DECIMAL(6,2), improvement_surcharge DECIMAL(6,2), total_amount DECIMAL(6,2), congestion_surcharge DECIMAL (6,2)) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

```
+----------------+    INFO  : Concurrency
| database_name  |    +-----------------+
+----------------+    |    tab_name     |
| default        |    +-----------------+
| taxi           |    | tb_dados_taxis  |
+----------------+    +-----------------+
2 rows selected (0.059 seconds)
0: jdbc:hive2://localhost:10000/default>
```

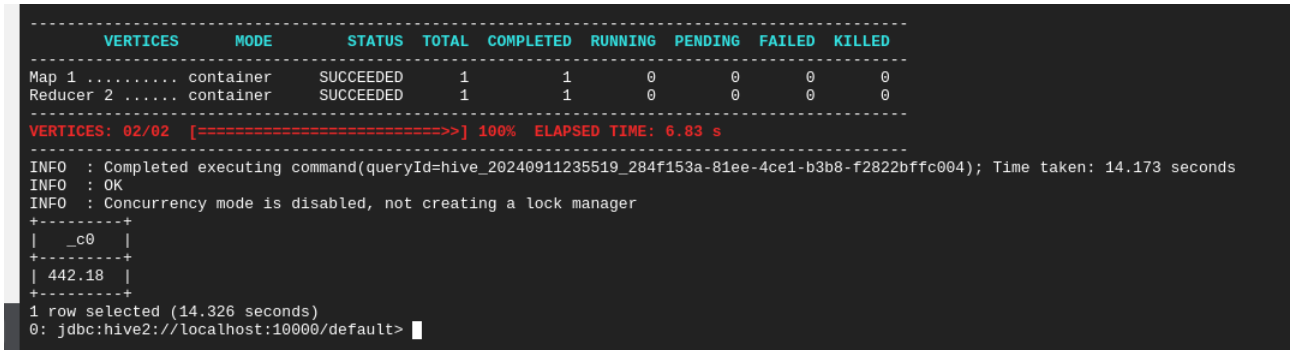
Comando de carga dos dados

```
verwrite into table tb_dados_taxis;
 LOAD DATA INPATH '/user/mariana_cmenezes/projeto' overwrite into table tb_dados_taxis
```

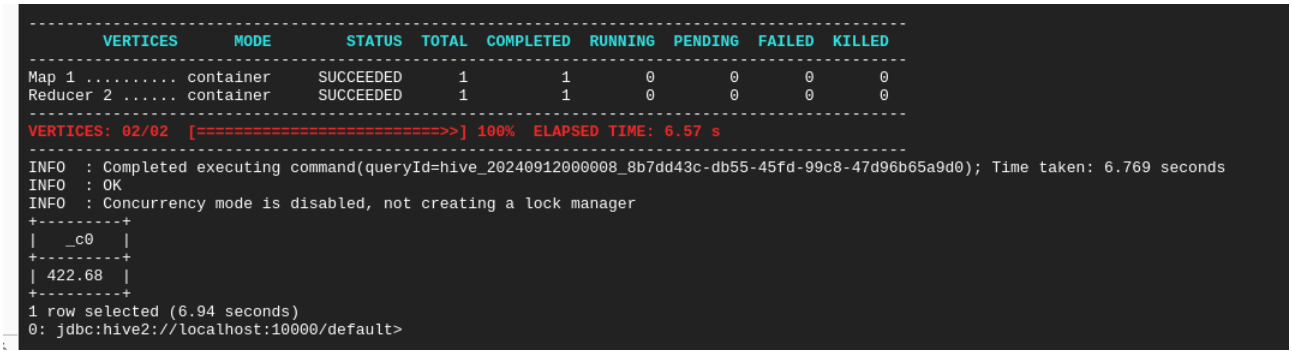
Consultas HQL

Valor máximo de gorjetas em cartão de crédito em maio e junho:

select max(tb_dados_taxis_maio.tip_amount) from tb_dados_taxis_maio;
select max(tb_dados_taxis.tip_amount) from tb_dados_taxis;



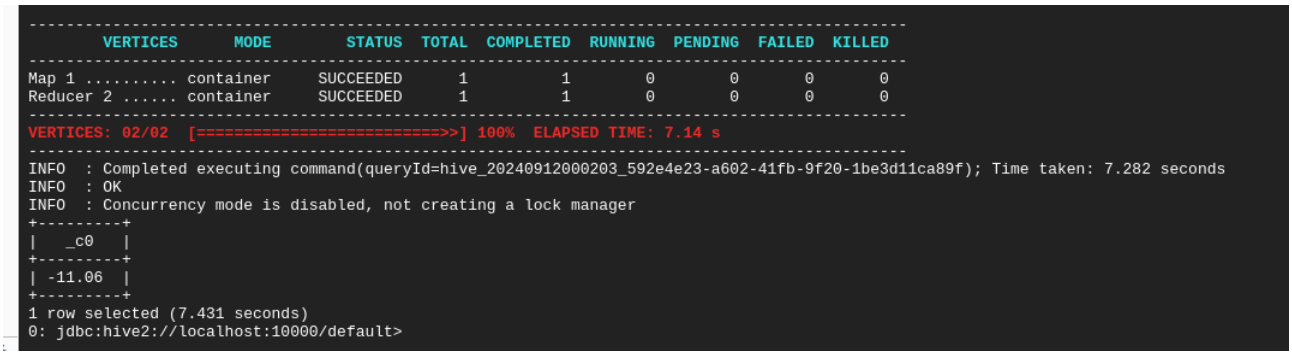
Valor máximo em maio: 422.2



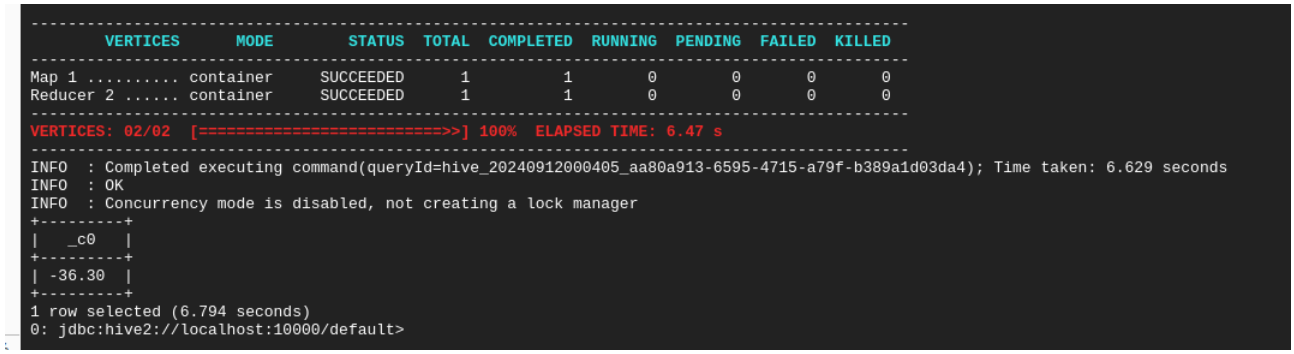
Valor máximo em junho: 422.7

Valor mínimo de gorjetas em cartão de crédito em maio e junho:

select min(tb_dados_taxis_maio.tip_amount) from tb_dados_taxis_maio;
select min(tb_dados_taxis.tip_amount) from tb_dados_taxis;



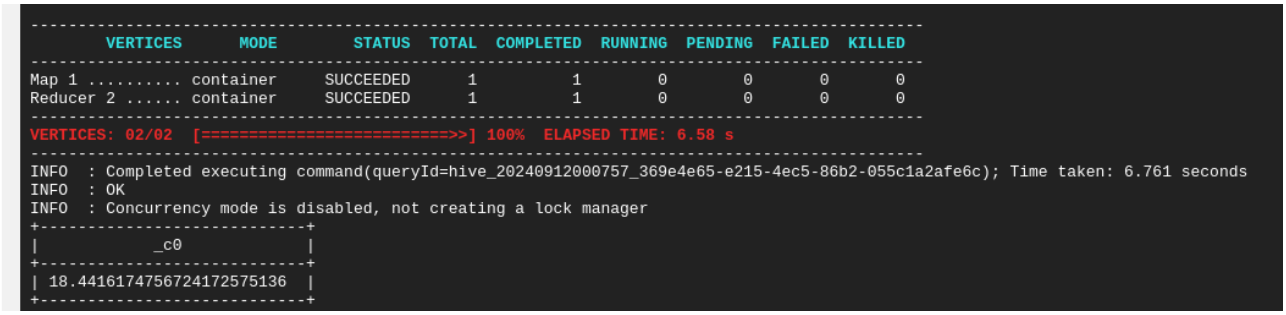
Valor mínimo em maio: 11.06



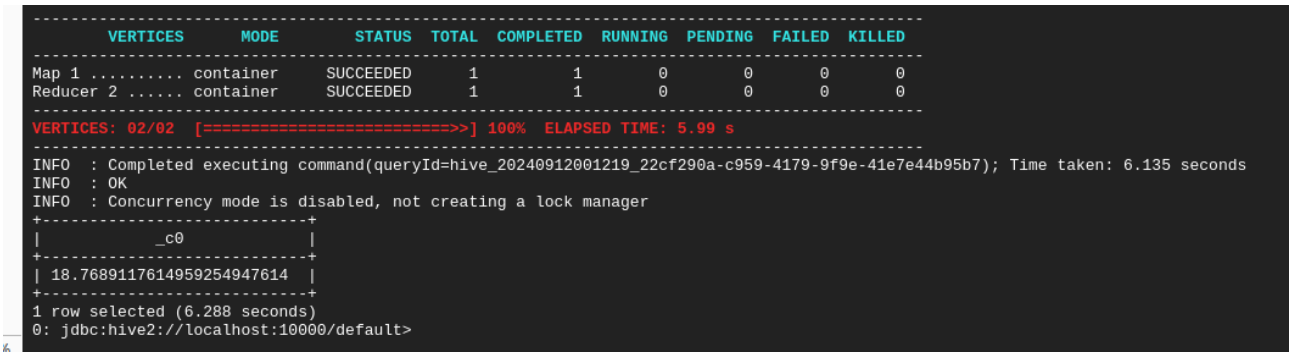
Valor mínimo em junho: 36.3

Valor médio das corridas em maio e junho:

query (maio): select avg(tb_dados_taxis_maio.total_amount) from tb_dados_taxis_maio
query (junho): select avg(tb_dados_taxis.total_amount) from tb_dados_taxis



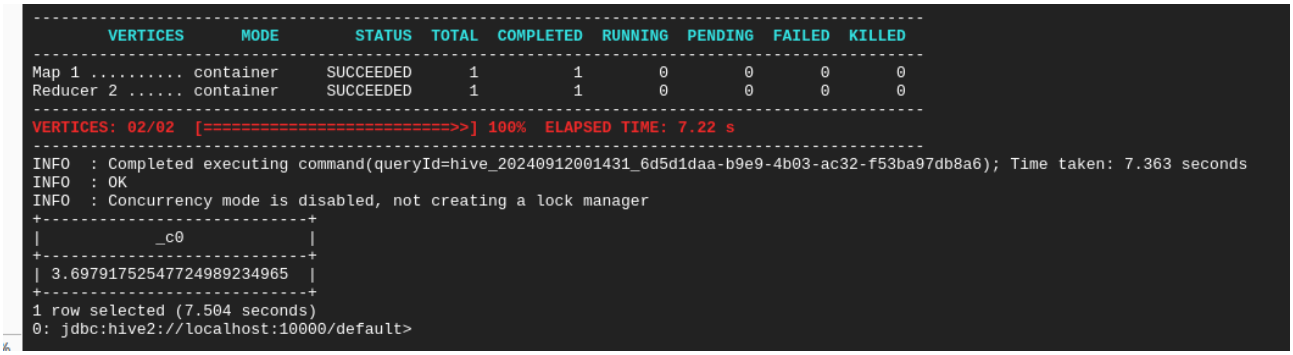
Valor médio em maio: 18.44



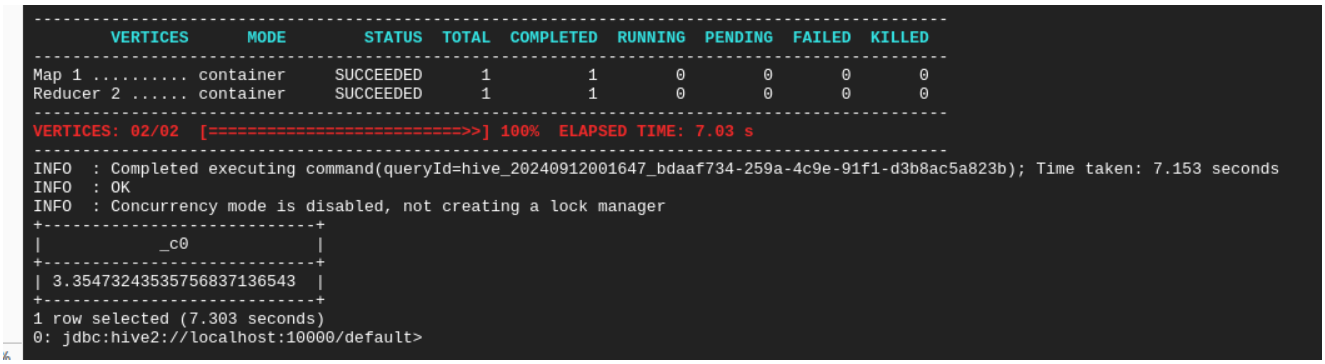
Valor médio em junho: 18.80

Média de distância das corridas em maio e junho

query(maio): select avg(tb_dados_taxis_maio.trip_distance) from tb_dados_taxis_maio
query(junho): select avg(tb_dados_taxis.trip_distance) from tb_dados_taxis



Maio: média de 3,7 milhas

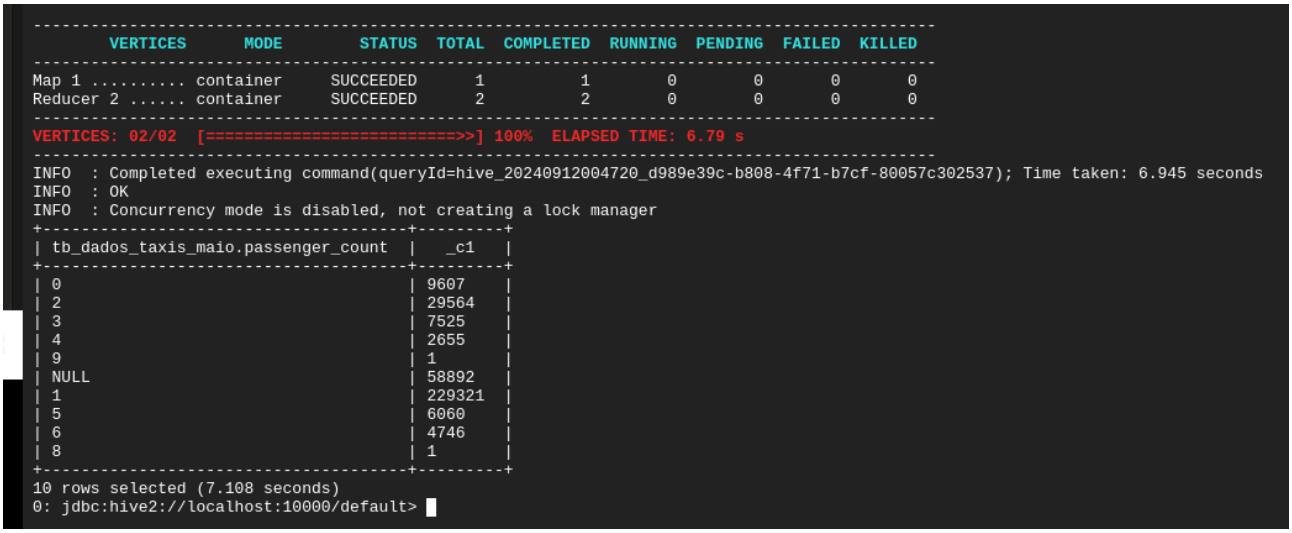


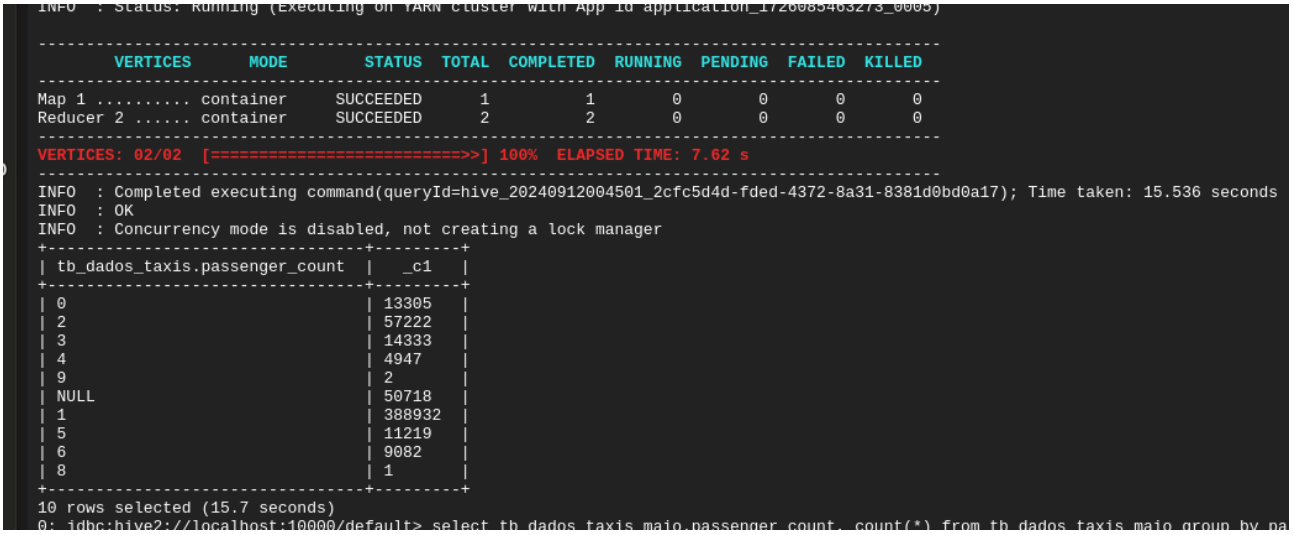
Junho: média de 3,5 milhas

Quantidade de passageiros por corrida em maio e junho

select tb_dados_taxis_maio.passenger_count, count (*) from tb_dados_taxis_maio
select tb_dados_taxis.passenger_count, count (*) from tb_dados_taxis

Observa-se que a maioria das corridas tem apenas um passageiro





CA

Proposta de evolução do projeto:

Fazer a integração com Spark para tratamento das bases de dados e utilização da Mllib para aplicação de modelos de machine learning em busca de correlações mais detalhadas entre as características das viagens. É possível também fazer a carga de todo histórico disponível no Kaggle para prever valores de corridas futuras com base na série temporal.

Este projeto está disponível em
https://github.com/maricmenezes/hadoop_projeto