

Практика. Работа с HIVE

1. Развернем контейнер с HIVE

Перейду с папку с контейнером

```
cd Downloads/ docker-hive-master
```

и подниму контейнер

```
docker-compose up -d
```

```
max@max:~/Downloads/docker-hive-master$ docker-compose up -d
Starting docker-hive-master_datanode_1 ... done
Starting docker-hive-master_hive-server_1 ... done
Starting docker-hive-master_hive-metastore-postgresql_1 ... done
Starting docker-hive-master_namenode_1 ... done
Starting docker-hive-master_presto-coordinator_1 ... done
Starting docker-hive-master_hive-metastore_1 ... done
```

2. Идем в бобра. -> Запускаем DBeaver

Скачать датасет с кaggла: <https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city>

uber-raw-data-janjune-15.csv

3. Закину его на namenod:

```
docker cp "/home/max/Downloads/hive_db/uber-raw-data-janjune-15.csv" docker-hive-master_namenode_1:
```

```
max@max:~/Downloads/docker-hive-master$ docker cp "/home/max/Downloads/hive_db/uber-raw-data-janjune-15.csv" docker-hive-master_namenode_1:/
max@max:~/Downloads/docker-hive-master$
```

4. Запускаю namenod, куда скопировали файл

```
docker exec -it docker-hive-master_namenode_1 bash
```

ls - увидим, что uber-raw-data-janjune-15.csv лежит в папке

```
root@5d686970546f:/# ls
airports.csv  entrypoint.sh  home  mnt  run  sys  var
bin           etc            lib   opt  run.sh  tmp
boot         hadoop        lib64  proc  sbin  uber-raw-data-janjune-15.csv
dev          hadoop-data   media  root  srv   usr
```

В контейнере создам папку «/my_testdata»

```
hdfs dfs -mkdir /my_testdata
```

и закину туда файл uber-raw-data-janjune-15.csv

```
hdfs dfs -put uber-raw-data-janjune-15.csv /my_testdata
```

```
root@5d686970546f:/# hdfs dfs -put uber-raw-data-janjune-15.csv /my_testdata
root@5d686970546f:/# hdfs dfs -ls /my_testdata
Found 1 items
-rw-r--r--  3 root supergroup  551672691 2021-12-19 09:04 /my_testdata/uber-raw-data-janjune-15.csv
```

Удалю файл файл uber-raw-data-janjune-15.csv из контейнера (он уже внутри hdfs)

ВЗЯТЬ файл из контейнера и закинуть его в нашу систему:

Сохранили в контейнере файл

```
hdfs dfs -get /user/hive/warehouse/mydb.db/uber_data_ex_pq/000000_0
```

Скопировал в локалку (в Downloads)

```
docker cp 5d686970546f:/000000_0 ~/Downloads
```

Прочитал Pandas`ом. Попробовал вычисления всякие – все работает

```
In [1]: import pandas as pd

In [2]: df = pd.read_parquet('000000_0')

In [3]: df.shape
Out[3]: (6943783, 4)

In [4]: df.head(5)
Out[4]:
```

	dispatching_base_num	pickup_date	affiliated_base_num	locationid
0	B02617	2015-05-17 09:47:00	B02617	141
1	B02617	2015-05-17 09:47:00	B02617	65
2	B02617	2015-05-17 09:47:00	B02617	100
3	B02617	2015-05-17 09:47:00	B02774	80
4	B02617	2015-05-17 09:47:00	B02617	90

```
In [ ]:
```

Browse Directory

/my_testdata

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxr-xr-x	root	supergroup	100.76 MB	12/19/2021, 8:59:44 PM	3	128 MB	000000_0

Hadoop, 2017.

```
--- SET parquet.compression=SNAPPY;
SET parquet.compression=SNAPPY;
SET hive.exec.compress.output=true;
SET mapred.output.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
SET mapred.output.compression.type=BLOCK;

CREATE TABLE uber_testsnappy_pq
(
  Dispatching_base_num string,
  Pickup_date timestamp,
  Affiliated_base_num string,
  locationID int
)
stored as PARQUET Location "/my_testdata";

insert overwrite table uber_testsnappy_pq -- => 28.9s
select * from uber_data_ex;

show CREATE table uber_testsnappy_pq;
```

Type	Text	Duration (ms)	Rows	Res
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	3,429	[2]	
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	2,812		Succ
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	2,788		Succ
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	4,785		Succ
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	3,867		Succ
SQL / User	insert overwrite table uber_testsnappy_pq -- => 28.9s	4,726		Succ

ORC с orc.compress="gzip" не записалась

```
CREATE TABLE uber_testsnappy_orc
(
  Dispatching_base_num string,
  Pickup_date timestamp,
  Affiliated_base_num string,
  locationID int
)

STORED AS ORC
TBLPROPERTIES("orc.compress"="gzip");

insert overwrite table uber_testsnappy_orc -- => 1.4s
select * from uber_data_ex;
```

Statistics 1 x

insert overwrite table uber_testsnappy_ Enter a SQL expression to filter results (use Ctrl+Space)

SQL Error [2] [08501]: org.apache.hive.service.cli.HiveSQLException: Error while processing statement: FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
at org.apache.hive.service.cli.operation.Operation.toSQLException(Operation.java:380)
at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:257)
at org.apache.hive.service.cli.operation.SQLOperation.access\$800(SQLOperation.java:91)
at org.apache.hive.service.cli.operation.SQLOperation\$BackgroundWork\$1.run(SQLOperation.java:348)

Save Cancel Script

SQL Error [2] [08501]: org.apache.hive.servi

Query Manager x

Type	Text	Duration (ms)	Rows
19:21:1 SQL / User	insert overwrite table uber_testsnappy_orc -- => 1.4s	1,389	