Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»
г. Екатеринбург

ОТЧЁТ

по лабораторной работе №4.

по дисциплине «Методы и инструменты анализа больших данных»

Ф.И.О.: Фадхил Фадхил Аббас Фадхил && Хасан Мухаммед Али

Институт: Институт радиоэлектроники и информационных технологий-РТФ

Направление:10.04.01

Группа: РИМ-201211

Направленность: Информационная безопасность

Преподаватель	_____	_____	С.Г. Мирвода
	(дата)	(подпись)	
Студент	_____	_____	Ф. Ф. Аббас
	(дата)	(подпись)	
Студент	_____	_____	Х. М. Али
	(дата)	(подпись)	

Екатеринбург 2021

Table of Contents

Задание 0

Задача подготовить полигон

Установить на свой кластер hadoop 3.3 СУБД HIVE

3.1.2 согласно инструкции и примеру

sudo wget https://dlcdn.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz

```
fadhil@ubuntu:~$ sudo wget https://dlcdn.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
[sudo] password for fadhil:
--2021-12-27 03:35:33--  https://dlcdn.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 278813748 (266M) [application/x-gzip]
Saving to: ‘apache-hive-3.1.2-bin.tar.gz’

apache-hive-3.1.2-bin.tar.gz        100%[=============================================================>] 265.90M   530KB/s    in 8m 9s

2021-12-27 03:43:44 (556 KB/s) - ‘apache-hive-3.1.2-bin.tar.gz’ saved [278813748/278813748]
```

tar -xzvf apache-hive-3.1.2-bin.tar.gz

sudo mv apache-hive-3.1.2-bin /usr/local/

```
fadhil@ubuntu:~$ sudo mv apache-hive-3.1.2-bin /usr/local/
drwxr-xr-x 14 root   root   4096 Dec 27 03:45 .
drwxr-xr-x 14 root   root   4096 Aug 19 03:32 ..
drwxrwxr-x 10 fadhil fadhil 4096 Dec 27 03:44 apache-hive-3.1.2-bin
drwxr-xr-x  2 root   root   4096 Aug 19 03:29 bin
drwxr-xr-x  2 root   root   4096 Aug 19 03:29 etc
drwxr-xr-x  2 root   root   4096 Aug 19 03:29 games
```

~ / .bashrc

```
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:PATH=$HADOOP_HOME/sbin
export HADOOP_CLASSPATH=${JAVA_HOME}/libs/tools.jar

export HIEV_HOME=/usr/local/apache-hive-3.1.2-bin
export CLASSPATH=$CLASSPATH:$HIEV_HOME/lib:$HADOOP_HOME/share/hadoop/common/lib
export PATH=$PATH:$HIEV_HOME/bin
# If not running interactively, don't do anything
```

Войти под пользователем hive и запустить консольную утилиту hive

```
root@ubuntu:/home/fadhil# hiveserver2
2021-12-27 04:25:51: Starting HiveServer2
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = f5df0759-5bc7-4e84-812d-bf9c6688fd62
Hive Session ID = 2ee7c1bb-00fc-4406-8823-649515b0d892
Hive Session ID = 86c14132-cfc3-44d4-956f-a9660f856204
Hive Session ID = b4cf257b-3cc0-4d68-ba70-db6b715f79b6
```

```
root@ubuntu:/home/fadhil# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 70fdeb23-b2e4-4c74-85a9-29b922e0db3b

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async:
true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, t
ez) or using Hive 1.X releases.
Hive Session ID = 8ebf25f6-274e-4f49-9c9c-8ca9126e5db6
```

Выполнить команду select version();

```
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 70fdeb23-b2e4-4c74-85a9-29b922e0db3b

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.2-bin/lib/
true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
ez) or using Hive 1.X releases.
Hive Session ID = 8ebf25f6-274e-4f49-9c9c-8ca9126e5db6
hive> select version();
OK
3.1.2 r8190d2be7b7165effa62bd21b7d60ef81fb0e4af
Time taken: 4.325 seconds, Fetched: 1 row(s)
hive>
```

Задание 1

Воспроизведите примеры из лекции и сохраните скрипт в свой репозиторий

```
hive> create table pokes(foo int, bar string);
OK
Time taken: 1.083 seconds
hive> create table invites(foo int, bar string) partitioned by (ds string);
OK
Time taken: 0.146 seconds
hive> show tables
    > ;
OK
invites
pokes
Time taken: 0.183 seconds, Fetched: 2 row(s)
hive>
```

```
hive> describe invites;
OK
foo                     int
bar                     string
ds                      string

# Partition Information
# col_name              data_type               comment
ds                      string
Time taken: 0.425 seconds, Fetched: 7 row(s)
hive>
```

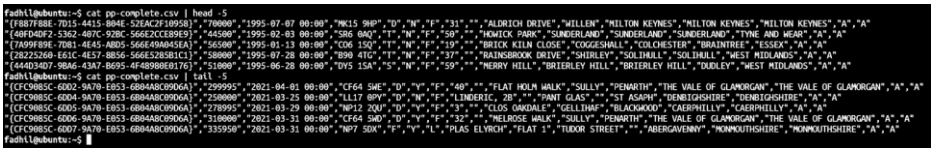
```
hive> ALTER TABLE invites ADD COLUMNS(col2 int comment 'my new col');
OK
Time taken: 0.315 seconds
hive> describe invites;
OK
foo                     int
bar                     string
col2                    int                     my new col
ds                      string

# Partition Information
# col_name              data_type               comment
ds                      string
Time taken: 0.223 seconds, Fetched: 8 row(s)
hive>
```

```
hive> drop table pokes
    > ;
OK
Time taken: 0.731 seconds
hive>
```

Задание 2

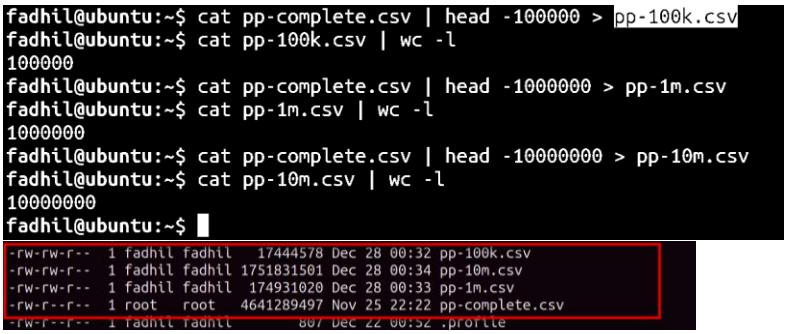
Загрузите тестовый массив данных в текущую папку (файл большой и в облаке, может качаться долго).

С помощью команд head и wc -l изучите его содержимо



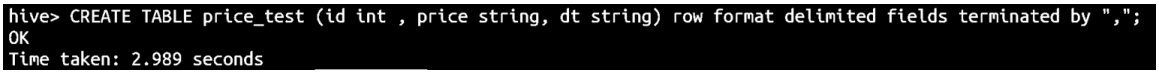
cat pp-complete.csv | head -100000 > pp-100k.csv

cat pp-100k.csv | wc -l

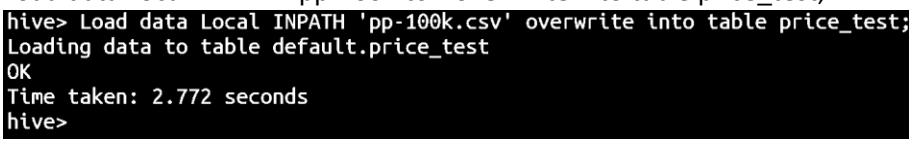


Создайте тестовую таблицу при помощи кода в примере 1 и загрузите в неё данные записав в отчёт скорость записи каждого файла (для каждого следующего файла таблицу можно удалять или создавать новую с другим именем), количество строк и скорость выполнения запроса count(*).

CREATE TABLE price_test (id int , price string, dt string) row format delimited fields terminated by ",";



Load data Local INPATH 'pp-100k.csv' overwrite into table price_test;



sudo cp pp-100k.csv /usr/local/

```
hive> SELECT * from price_test;
```

```
Time taken: 0.265 seconds, Fetched: 100000 row(s)
```

1m

CREATE TABLE price_test_1m (id int , price string, dt string) row format delimited fields terminated by ",";

```
hive> CREATE TABLE price_test_1m (id int , price string, dt string) row format delimited fields terminated by ",";
OK
Time taken: 1.596 seconds
```

Load data Local INPATH 'pp-1m.csv' overwrite into table price_test_1m;

```
hive> Load data Local INPATH 'pp-1m.csv' overwrite into table price_test_1m;
Loading data to table default.price_test_1m
OK
Time taken: 3.37 seconds
```

```
hive> SELECT * from price_test_1m;
```

```
Time taken: 5.602 seconds, Fetched: 1000000 row(s)
hive>
```

10m

CREATE TABLE price_test_10m (id int , price string, dt string) row format delimited fields terminated by ",";

```
hive>
    > CREATE TABLE price_test_10m (id int , price string, dt string) row format delimited fields terminated by ",";
OK
Time taken: 0.994 seconds
hive>
```

Load data Local INPATH 'pp-10m.csv' overwrite into table price_test_10m;

```
hive> Load data Local INPATH 'pp-10m.csv' overwrite into table price_test_10m;
Loading data to table default.price_test_10m
OK
Time taken: 34.282 seconds
hive>
```

```
hive> select * from price_test_10m;
```

```
Time taken: 0.207 seconds, Fetched: 10000000 row(s)
hive>
```

Задание 3

Дополнив оставшимися колонками пример ниже загрузите данные в таблицы HIVE, замерьте время загрузки и запишите в отчёт

```
CREATE TABLE price (
  id STRING,
  price INT,
  datetime DATE,
  postcode STRING,
```

```
  property_type STRING,
  new_build_flag STRING,
  tenure_type STRING,
  primary_addressable_object_name STRING,
  secondary_addressable_object_name STRING,
  street STRING,
  locality STRING,
  town_city STRING,
  district STRING,
  county STRING
  )
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("separatorChar" = ",", "quoteChar"="\"", "escapeChar"="\\")
STORED AS TEXTFILE;
```


```
LOAD DATA LOCAL INPATH 'opt/pp-complete.csv' OVERWRITE INTO TABLE price;
select * from price2 limit 10;
```

В итоговой таблице должно содержаться 16 колонок и 26_541_204 строк.

6.1. Количество загруженных строк данных

“select date_format(datetime, 'yyyy'),town_city,cast(avg(price) as INT) from price group by date_format(datetime, 'yyyy'),town_city;” > f1.txt

```
MapReduce Total cumulative CPU time: 33 minutes 8 seconds 710 msec
Ended Job = job_1640294990941_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 18  Reduce: 19   Cumulative CPU: 1988.71 sec   HDFS Read: 4641648970 HDFS Write: 2296 SUCCESS
Total MapReduce CPU Time Spent: 33 minutes 8 seconds 710 msec
OK
Time taken: 261.199 seconds, Fetched: 27 row(s)
```

27 lines (27 sloc) | 319 Bytes

```
1995    67931
1996    71506
1997    78532
1998    85436
1999    96037
2000    107483
2001    118885
2002    137942
2003    155888
2004    178886
2005    189352
2006    203528
2007    219378
2008    217056
2009    213419
2010    236109
2011    232804
2012    238366
2013    256923
2014    279938
2015    297266
2016    313222
2017    346095
2018    350275
2019    351488
```

6.2. Средняя цена за год

Hive “select date_format(datetime, 'yyyy'),town_city,cast(avg(price) as INT) from price group by date_format(datetime, 'yyyy'),town_city order by date_format(datetime, 'yyyy');” > f2.txt

Executable File | 31180 lines (31180 sloc) | 670 KB

```
1995    WELLINGBOROUGH  45870
1995    WEYBRIDGE       157201
1995    WHITSTABLE      61647
1995    WIGTON  52778
1995    AMLWCH  37322
1995    ASHBOURNE       86230
1995    ASHFORD 74976
1995    PRENTON 53984
1995    PICKERING       71060
1995    PENARTH 74589
1995    BACUP   32905
1995    BARNSLEY        43275
1995    BATTLE  91825
1995    BEAMINSTER      73240
1995    BETWS-Y-COED    37049
1995    BEWDLEY 72812
1995    BINGLEY 63478
1995    BOGNOR REGIS    66521
1995    BOREHAMWOOD     85596
1995    BOSTON  43451
1995    BOURNE  57090
1995    BRACKNELL       88567
1995    BUCKHURST HILL  106542
1995    CALLINGTON      58493
1995    CARNFORTH       73515
1995    CASTLE CARY     62556
```

```
MapReduce Total cumulative CPU time: 12 seconds 490 msec
Ended Job = job_1640294990941_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 18  Reduce: 19   Cumulative CPU: 1901.96 sec   HDFS Read: 4641643163 HDFS Write: 1128126 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 12.49 sec   HDFS Read: 1147174 HDFS Write: 1060693 SUCCESS
Total MapReduce CPU Time Spent: 31 minutes 54 seconds 450 msec
OK
Time taken: 250.612 seconds, Fetched: 31180 row(s)
```

6.3 Средняя цена за год в Городе 6.4 Самые дорогие районы

```
Hive “select district,cast(avg(price) as INT) from price group by district  order by cast(avg(price) as INT) DESC;” > f3.txt
```

Executable File | 463 lines (463 sloc) | 8.54 KB

```
CITY OF LONDON  1995179
KENSINGTON AND CHELSEA  1039831
CITY OF WESTMINSTER     1028504
CAMDEN  719088
HAMMERSMITH AND FULHAM  565830
BUCKINGHAMSHIRE 564427
ELMBRIDGE       497489
ISLINGTON       494469
RICHMOND UPON THAMES    471790
SOUTH BUCKS     457049
WANDSWORTH      438007
SOUTHWARK       413146
WEST NORTHAMPTONSHIRE   398455
TOWER HAMLETS   396889
CHILTERN        394007
WEST SUFFOLK    391442
WINDSOR AND MAIDENHEAD  386591
BOURNEMOUTH, CHRISTCHURCH AND POOLE     383726
HACKNEY 381256
BARNET  375548
MOLE VALLEY     367783
WAVERLEY        366398
DORSET  363360
ST ALBANS       362909
LAMBETH 360981
ST MARTIN'S     356250
```

```
MapReduce Total cumulative CPU time: 10 seconds 100 msec
Ended Job = job_1640294990941_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 18  Reduce: 19   Cumulative CPU: 332.3 sec   HDFS Read: 4641638708 HDFS Write: 17074 SUCCESS
Stage-Stage-2: Map: 3  Reduce: 1   Cumulative CPU: 10.1 sec   HDFS Read: 35532 HDFS Write: 14384 SUCCESS
Total MapReduce CPU Time Spent: 5 minutes 42 seconds 400 msec
OK
Time taken: 96.052 seconds, Fetched: 463 row(s)
```