

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 .....	3
Установить на свой кластер hadoop 3.3 СУБД HIVE 3.1.2 согласно инструкции и примеру .....	3
Войти под пользователем hive и запустить консольную утилиту hive .....	3
Выполнить команду select version(); .....	4
Задание 1 .....	4
Воспроизведите примеры из лекции и сохраните скрипт в свой репозиторий .....	4
Задание 2 .....	4
Загрузите тестовый массив данных в текущую папку (файл большой и в облаке, может качаться долго). .....	5
С помощью команд head и wc -l изучите его содержимое .....	5
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(*). .....	5
Задание 3 .....	6
Дополнив оставшимися колонками пример ниже загрузите данные в таблицы HIVE, замерьте время загрузки и запишите в отчёт .....	6
В итоговой таблице должно содержаться 16 колонок и 26_541_204 строк. ....	7
6.1. Количество загруженных .....	7
6.2. Средняя цена за год .....	8
6.3 Средняя цена за год в Городе 6.4 Самые дорогие районы .....	9

## 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 -xzf 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:$HADOOP_HOME/sbin
export HADOOP_CLASSPATH=${JAVA_HOME}/libs/tools.jar

export HIVE_HOME=/usr/local/apache-hive-3.1.2-bin
export CLASSPATH=$CLASSPATH:$HIVE_HOME/lib:$HADOOP_HOME/share/hadoop/common/lib
export PATH=$PATH:$HIVE_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-80fc-4406-8823-649515b0d992
Hive Session ID = 86c14132-cfc3-44d4-956f-a9660f856204
Hive Session ID = b4cf257b-3cc0-4d68-ba76-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, tez) 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. Consider
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).

```
fadhil@ubuntu:~$ sudo wget http://prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-complete.csv
[sudo] password for fadhil:
--2021-12-27 07:03:32-- http://prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-complete.csv
Resolving prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com (prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com)... 52.218.61.228
Connecting to prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com (prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com)[52.218.61.228]:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-complete.csv [following]
--2021-12-27 07:03:34-- http://prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-complete.csv
Resolving prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com (prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com)... 52.218.40.22
Connecting to prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com (prod1.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com)[52.218.40.22]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 464128947 (4.3G) [text/csv]
Saving to: 'pp-complete.csv.1'

pp-complete.csv.1 95%[=====] 4.14G 1.49MB/s eta 2m 37
pp-complete.csv.1 100%[=====] 4.32G 1.41MB/s ln 63m 19s

2021-12-27 08:07:04 (1.17 MB/s) - 'pp-complete.csv.1' saved [464128947/464128947]
```

С помощью команд head и wc -l изучите его содержимое

```
fadhil@ubuntu:~$ cat pp-complete.csv | head -5
{"F88788E-7D15-4415-888E-52EACF1895B8","70000","1995-07-07 00:00","MK15 SHP","D","N","F","31","",,"ALDRICH DRIVE","WILLEN","MILTON KEYNES","MILTON KEYNES","MILTON KEYNES","A","A"
{"06FA0F2-5282-407E-920C-5662C2C89493","44500","1995-02-03 00:00","206 940","T","N","F","50","",,"NEMICK PIKE","SHERESLAND","SUNDERLAND","SUNDERLAND","TYNE AND WEAR","A","A"
{"7A09B9E-7D81-4E45-AD05-566849A0454A","56500","1995-01-13 00:00","COS 150","T","N","F","10","",,"BRICK KILN CLOSE","COGGESWALL","COLCHESTER","BRAINTREE","ESSEX","A","A"
{"28223268-E61C-4E57-8856-566528381C1","58000","1995-07-28 00:00","B99 410","T","N","F","37","",,"BAINGSBROOK DRIVE","SHIRLEY","SOLIHULL","SOLIHULL","WEST MIDLANDS","A","A"
{"4A02407-9846-4347-8095-4F499A83176","21000","1995-06-28 00:00","OVS 154","S","N","F","39","",,"HERRY HILL","BRIERLEY HILL","BRIERLEY HILL","DUDLEY","WEST MIDLANDS","A","A"
fadhil@ubuntu:~$ cat pp-complete.csv | tail -5
{"CF0985C-6D02-8A70-E853-6804ABC9D6A","299995","2021-04-01 00:00","CF64 SNE","D","N","F","40","",,"FLAT HOLM WALK","SULLY","PENARTH","THE VALE OF GLAMORGAN","THE VALE OF GLAMORGAN","A","A"
{"CF0985C-6D04-8A70-E853-6804ABC9D6A","250000","2021-03-25 00:00","LL17 00V","D","N","F","LINDBRIC","30","",,"PANT GLAS","ST ASAPH","DENBIGHSHIRE","DENBIGHSHIRE","A","A"
{"CF0985C-6D05-8A70-E853-6804ABC9D6A","278995","2021-03-29 00:00","NP12 20U","D","N","F","53","",,"CLOS GANDALE","GELLIDAF","BLACKBOD","CAERNELLILLY","CAERNELLILLY","A","A"
{"CF0985C-6D06-8A70-E853-6804ABC9D6A","310000","2021-03-31 00:00","CF64 SMO","D","N","F","32","",,"NELROSE WALK","SULLY","PENARTH","THE VALE OF GLAMORGAN","THE VALE OF GLAMORGAN","A","A"
{"CF0985C-6D07-8A70-E853-6804ABC9D6A","335950","2021-03-31 00:00","WPT 304","F","N","L","PLAS ELYCH","FLAT 1","TUDOR STREET","",,"ABERGAVENNY","MONMOUTHSHIRE","MONMOUTHSHIRE","A","A"
fadhil@ubuntu:~$
```

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

```
cat pp-100k.csv | wc -l
```

```
fadhil@ubuntu:~$ cat pp-complete.csv | head -100000 > pp-100k.csv
fadhil@ubuntu:~$ cat pp-100k.csv | wc -l
100000
fadhil@ubuntu:~$ cat pp-complete.csv | head -1000000 > pp-1m.csv
fadhil@ubuntu:~$ cat pp-1m.csv | wc -l
1000000
fadhil@ubuntu:~$ cat pp-complete.csv | head -10000000 > pp-10m.csv
fadhil@ubuntu:~$ cat pp-10m.csv | wc -l
10000000
fadhil@ubuntu:~$
```

```
-rw-rw-r-- 1 fadhil fadhil 17444578 Dec 28 00:32 pp-100k.csv
-rw-rw-r-- 1 fadhil fadhil 1751831501 Dec 28 00:34 pp-10m.csv
-rw-rw-r-- 1 fadhil fadhil 174931020 Dec 28 00:33 pp-1m.csv
-rw-r--r-- 1 root root 4641289497 Nov 25 22:22 pp-complete.csv
-rw-r--r-- 1 fadhil fadhil 807 Dec 28 00:32 .profile
```

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 ",";

```
hive> CREATE TABLE price_test (id int , price string, dt string) row format delimited fields terminated by ",";
OK
Time taken: 2.989 seconds
```

Load data Local INPATH 'pp-100k.csv' overwrite into table price\_test;

```
hive> Load data Local INPATH 'pp-100k.csv' overwrite into table price_test;
Loading data to table default.price_test
OK
Time taken: 2.772 seconds
hive>
```

```
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;

```

```

hive> 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;
OK
Time taken: 1.186 seconds
hive>

```

```

hive> describe price2;
OK
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
Time taken: 2.195 seconds, Fetched: 14 row(s)
hive>

```

```

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

```

```

hive> select * from price2 limit 10;
OK
{"F8B7F8BE-7D15-4415-804E-52EAC2F1995B"},"70000","1995-07-07 00:00","M13 9HP","D","N","F","31","1","ALDRICH DRIVE","WILLEN","MILTON KEYNES","MILTON KEYNES","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL NULL
{"48FD4DF2-5362-497C-92BC-566E2CCEB9E9"},"44500","1995-02-03 00:00","SR6 8AQ","T","N","F","58","1","HOWICK PARK","SUNDERLAND","SUNDERLAND","TYNE AND WEAR","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"7A99F09E-7D81-4E45-ABD5-566E49A85EA"},"56500","1995-01-13 00:00","CO6 1SQ","T","N","F","19","1","BRICK KILN CLOSE","COGGESHALL","COLCHESTER","BRAINTREE","ESSEX","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"28225260-E61C-4E57-8B56-566E5285B1C1"},"58000","1995-07-28 00:00","B90 4TG","T","N","F","37","1","RAINSBROOK DRIVE","SHIRLEY","SOLIHULL","SOLIHULL","WEST MIDLANDS","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"44403407-9846-43A7-B695-4F48980E1763"},"51000","1995-06-28 00:00","DY5 1SA","S","N","F","59","1","MERRY HILL","BRIERLEY HILL","BRIERLEY HILL","DUDLEY","WEST MIDLANDS","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"28225260-E61C-4E57-8B56-566E5285B1C1"},"17000","1995-03-10 00:00","S65 1QJ","T","N","L","22","1","DENMAN STREET","ROTHERHAM","ROTHERHAM","ROTHERHAM","SOUTH YORKSHIRE","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
{"4E76CA93-18CC-4399-BF63-4F48A2857D43"},"17000","1995-03-10 00:00","S65 1QJ","T","N","L","22","1","DENMAN STREET","ROTHERHAM","ROTHERHAM","ROTHERHAM","SOUTH YORKSHIRE","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"709FB471-3690-4945-A0D6-4F48C65A0A6D"},"58000","1995-04-28 00:00","PE7 3AL","D","Y","F","4","1","BROOK LANE","FARCET","PETERBOROUGH","PETERBOROUGH","CAMBRIDGESHIRE","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"5FA6862E-5378-4278-8C47-5A060540596D"},"19500","1995-01-27 00:00","SK10 2QM","T","N","L","38","1","GARDEN STREET","MACCLESFIELD","MACCLESFIELD","MACCLESFIELD","CHESHIRE","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"E787710AD-ED1A-4811-AB99-5A06140510AD"},"20000","1995-01-16 00:00","S46 5AY","D","N","F","592","1","CLYDACH ROAD","HIVSTAME","SHANSEA","SHANSEA","SHANSEA","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL NULL
{"1D7BF83E-53A7-4813-A37C-5A06247A09A8"},"137500","1995-03-31 00:00","NR2 2NQ","D","N","F","26","1","LINE TREE ROAD","NORWICH","NORWICH","NORWICH","NORFOLK","A","A" NULL NULL NULL NULL NULL NULL NULL NULL NULL NULL
NULL
Time taken: 1.641 seconds, Fetched: 10 row(s)

```

В итоговой таблице должно содержаться 16 колонок и 26\_541\_204 строк.

```

Ended Job = job_1640286116539_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 18 Reduce: 1 Cumulative CPU: 299.4 sec HDFS Read: 4641575673 HDFS Write: 108 SUCCESS
Total MapReduce CPU Time Spent: 4 minutes 59 seconds 400 msec
OK
26541204
Time taken: 161.226 seconds, Fetched: 1 row(s)

```

## 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		
1	1995	67931
2	1996	71506
3	1997	78532
4	1998	85436
5	1999	96037
6	2000	107483
7	2001	118885
8	2002	137942
9	2003	155888
10	2004	178886
11	2005	189352
12	2006	203528
13	2007	219378
14	2008	217056
15	2009	213419
16	2010	236109
17	2011	232804
18	2012	238366
19	2013	256923
20	2014	279938
21	2015	297266
22	2016	313222
23	2017	346095
24	2018	350275
25	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			
1	1995	WELLINGBOROUGH	45870
2	1995	WEYBRIDGE	157201
3	1995	WHITSTABLE	61647
4	1995	WIGTON	52778
5	1995	AMLWCH	37322
6	1995	ASHBOURNE	86230
7	1995	ASHFORD	74976
8	1995	PRENTON	53984
9	1995	PICKERING	71060
10	1995	PENARTH	74589
11	1995	BACUP	32905
12	1995	BARNSELY	43275
13	1995	BATTLE	91825
14	1995	BEAMINSTER	73240
15	1995	BETWS-Y-COED	37049
16	1995	BEWDLEY	72812
17	1995	BINGLEY	63478
18	1995	BOGNOR REGIS	66521
19	1995	BOREHAMWOOD	85596
20	1995	BOSTON	43451
21	1995	BOURNE	57090
22	1995	BRACKNELL	88567
23	1995	BUCKHURST HILL	106542
24	1995	CALLINGTON	58493
25	1995	CARNFORTH	73515
26	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
1 CITY OF LONDON 1995179
2 KENSINGTON AND CHELSEA 1039831
3 CITY OF WESTMINSTER 1028504
4 CAMDEN 719088
5 HAMMERSMITH AND FULHAM 565830
6 BUCKINGHAMSHIRE 564427
7 ELMBRIDGE 497489
8 ISLINGTON 494469
9 RICHMOND UPON THAMES 471790
10 SOUTH BUCKS 457049
11 WANDSWORTH 438007
12 SOUTHWARK 413146
13 WEST NORTHAMPTONSHIRE 398455
14 TOWER HAMLETS 396889
15 CHILTERN 394007
16 WEST SUFFOLK 391442
17 WINDSOR AND MAIDENHEAD 386591
18 BOURNEMOUTH, CHRISTCHURCH AND POOLE 383726
19 HACKNEY 381256
20 BARNET 375548
21 MOLE VALLEY 367783
22 WAVERLEY 366398
23 DORSET 363360
24 ST ALBANS 362909
25 LAMBETH 360981
26 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)

```