Crime Data Analysis Based On Hadoop Framework Using Hive Tool

Done By

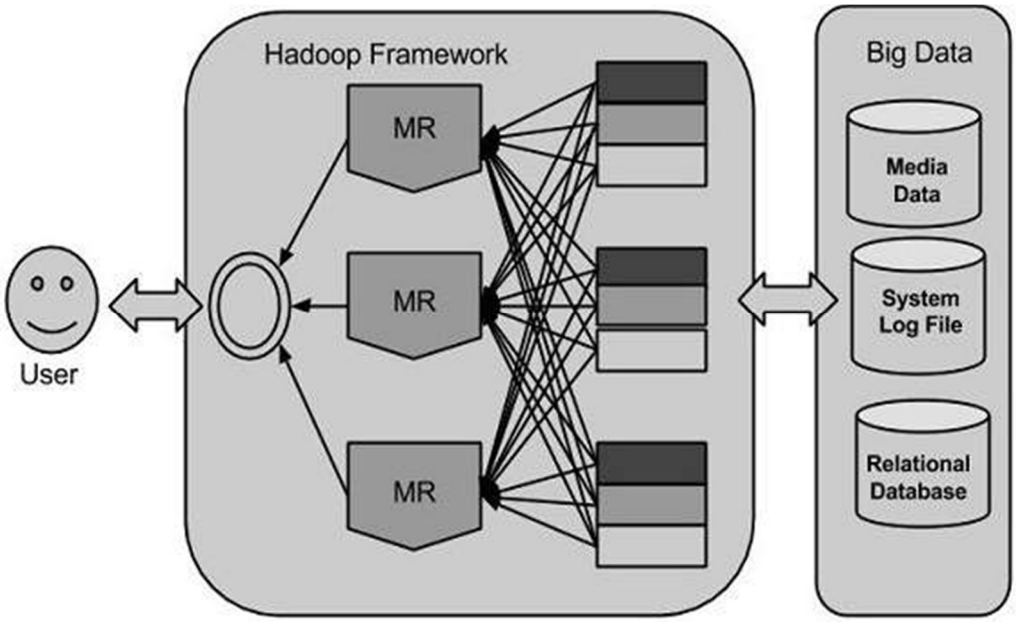
Tripti Singh

Hadoop :

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Doug Cutting, Mike Cafarella and team took the solution provided by Google and started an Open Source Project called HADOOP in 2005 and Doug named it after his son's toy elephant. Now Apache Hadoop is a registered trademark of the Apache Software Foundation.

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for a huge amounts of data.



Hadoop Architecture :

Hadoop framework includes following four modules:

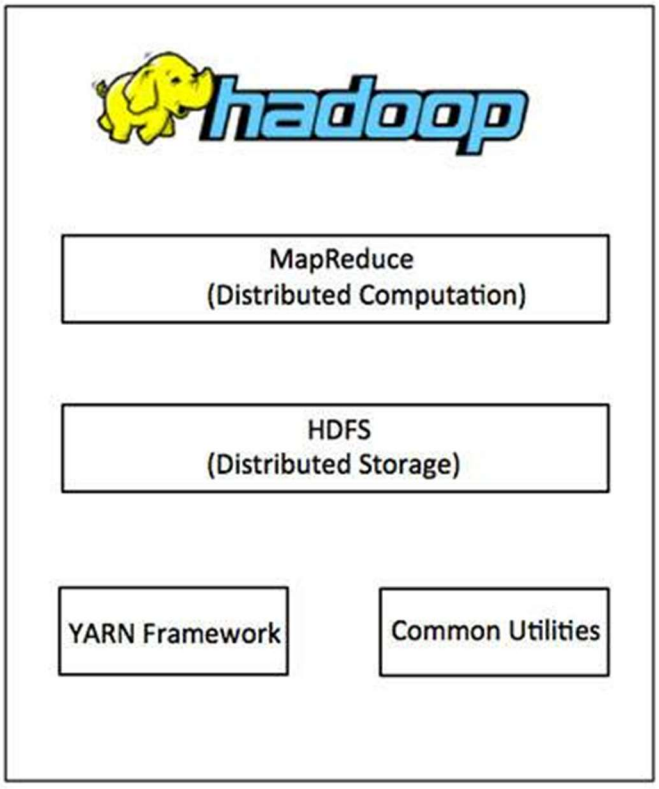
Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS™): A distributed file system that provides high-throughput access to application data.

Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop framework.



Since 2012, the term "Hadoop" often refers not just to the base modules mentioned above but also to the collection of additional software packages that can be installed on

top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Spark etc.

MapReduce :

Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

Phases in MapReduce :

A MapReduce job splits a large data set into independent chunks and organizes them into key, value pairs for parallel processing. A key-value pair (KVP) is a set of two linked data items: a key, which is a unique identifier for some item of data, and the value, which is either the data that is identified or a pointer to the location of that data. The mapping and reducing functions receive not just values, but (key, value) pairs.This parallel processing improves the speed and reliability of the cluster, returning solutions more quickly and with greater reliability.

Every MapReduce job consists of at-least three parts:

The driver

The Mapper

The Reducer

The Map Task:

This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).

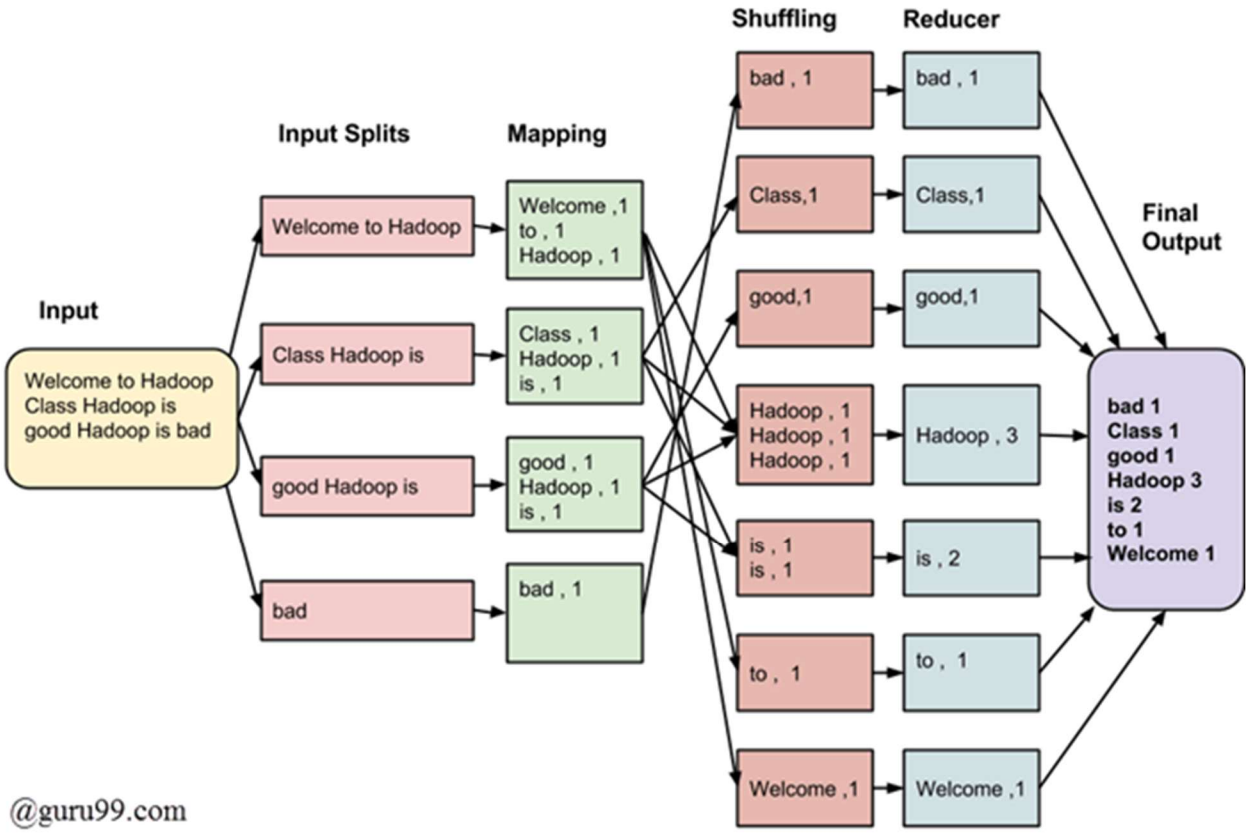
The Reduce Task:

This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.



Hadoop Distributed File System :

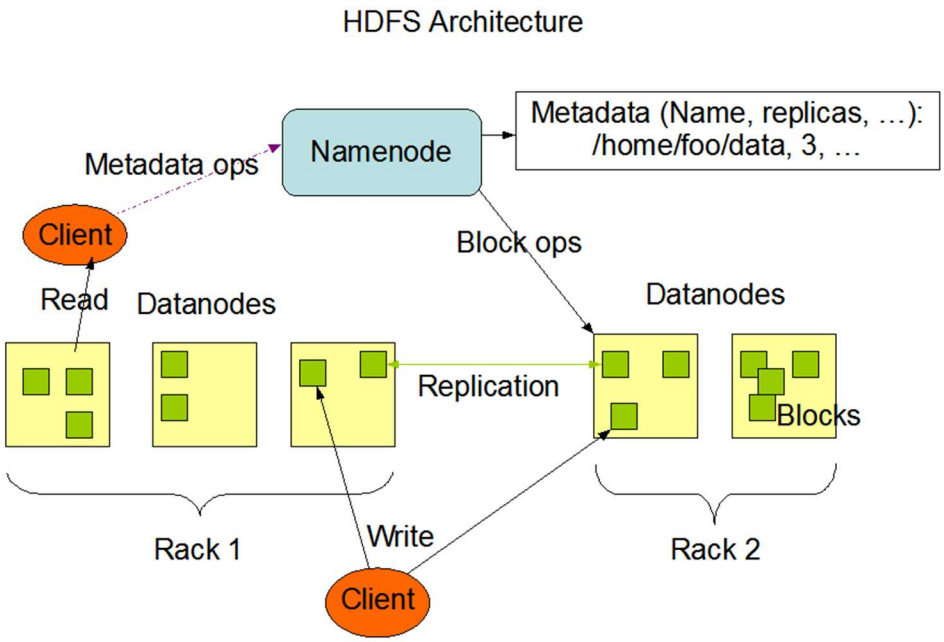
Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data.

A file in an HDFS namespace is split into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes takes care of read and write operation with the file system. They also take care of block creation, deletion and replication based on instruction given by NameNode.

HDFS provides a shell like any other file system and a list of commands are available to interact with the file system. These shell commands will be covered in a separate chapter along with appropriate examples.



Hive

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Hive is not :

A relational datab

A design for OnLine Transaction Processing (OLTP

A language for real-time queries and row-level updates

Features of Hive :

It stores schema in a database and processed data into HDFS.

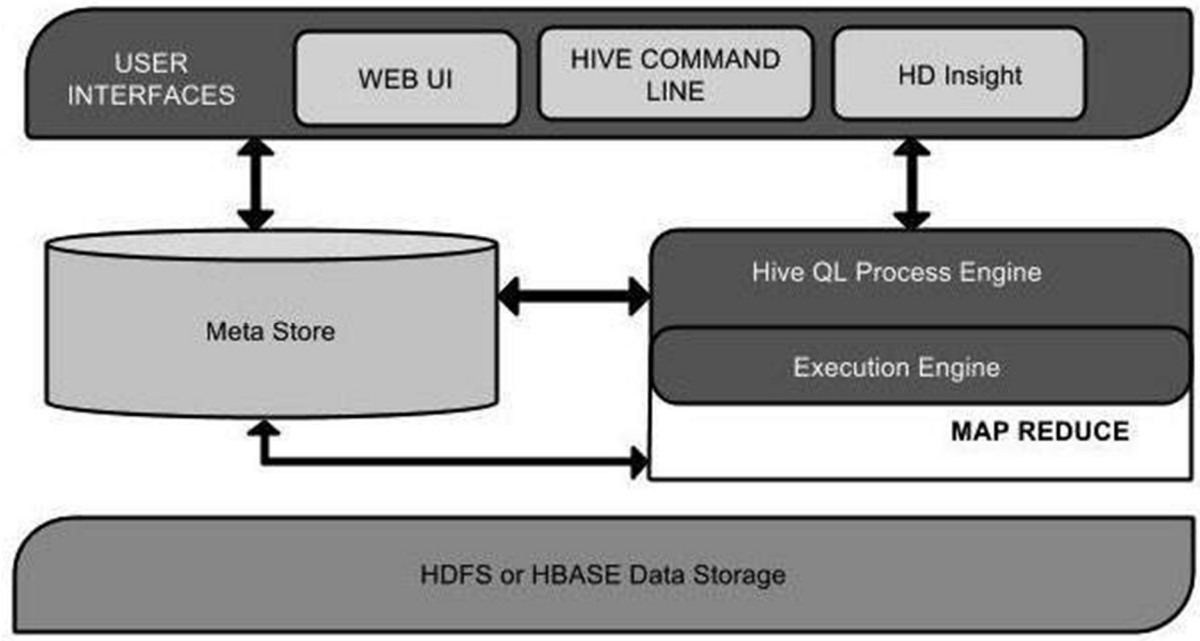
It is designed for OLAP.

It provides SQL type language for querying called HiveQL or HQL.

It is familiar, fast, scalable, and extensible.

Architecture of Hive :

The following component diagram depicts the architecture of Hive:



User Interface:

Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).

Meta Store:

Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

HiveQL Process Engine:

HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

Execution Engine:

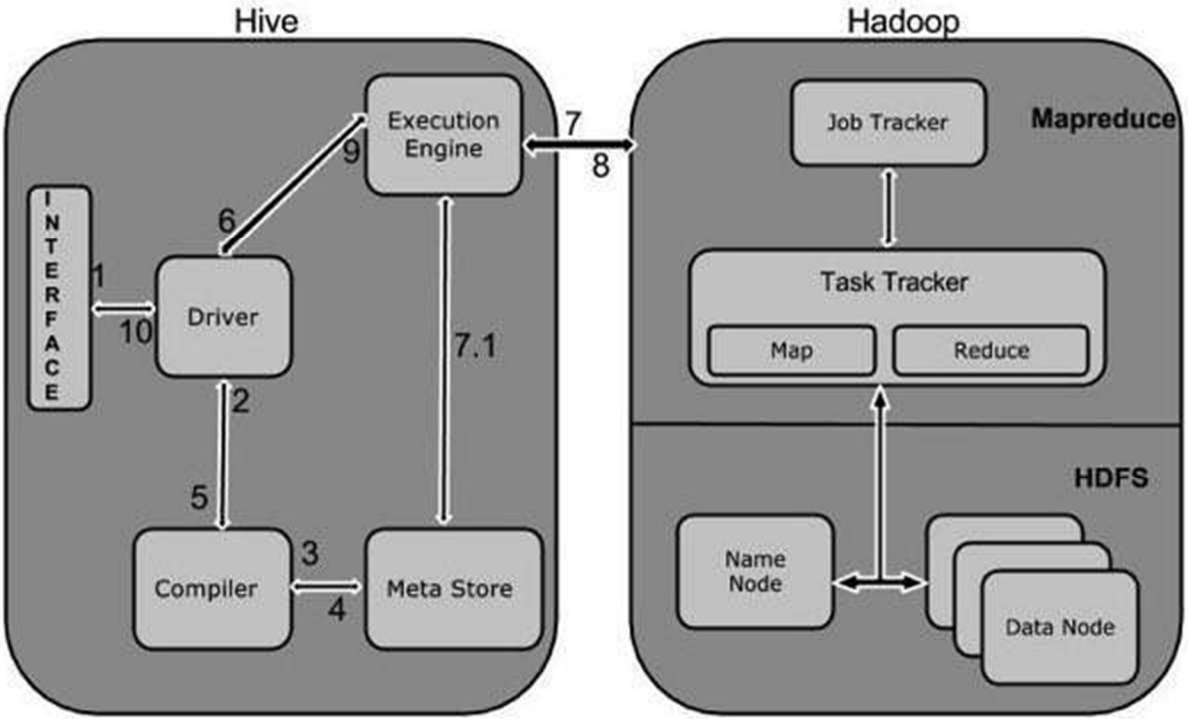
The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.

HDFS or HBASE:

Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Working of Hive :

The following diagram depicts the workflow between Hive and Hadoop.



1 : Execute Query

The Hive interface such as Command Line or Web UI sends query to Driver (any database driver such as JDBC, ODBC, etc.) to execute.

2 : Get Plan

The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.

3: Get Metadata

The compiler sends metadata request to Metastore (any database).

4: Send Metadata

Metastore sends metadata as a response to the compiler.

5: Send Plan

The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.

6: Execute Plan

The driver sends the execute plan to the execution engine.

7: Execute Job

Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.

7.1 : Metadata Ops

Meanwhile in execution, the execution engine can execute metadata operations with Metastore.

8: Fetch Result

The execution engine receives the results from Data nodes.

9: Send Results

The execution engine sends those resultant values to the driver.

10 : Send Results

The driver sends the results to Hive Interfaces.

Introduction of project

America for being number one to top the list of countries with highest crime rates. When we speak of American crimes, its reported on an annual basis in numbers of millions which include almost every crime committed that are known to man, every second,

every minute, every hour, every day, every week, every month and all year round which means to say that the crime cycle of Americans are around the clock non-stop. Even with the highest of trained police, best special forces and their crime prevention methods implemented and brutal force where needed, the crime never ends.

However, as a recent America Police campaign has articulated – “high Crime doesn’t mean we can not decrease it ”. With a view of generating results to help the American Police Force tackle crime effectively, we mined a set of crime data. Our findings are presented in this documentation. From our preliminary analysis we found that a large number of crimes occurred in residential areas. This is pertinent as America has a very high population density and a large proportion of the population live in government housing. Therefore, besides gearing our results towards the American Police Force, we sought to help the American Housing Development Board as well.

Due to the sensitive nature of crime data, I was given a censored version of the dataset. Our version was wiped of any sensitive details, which limited our analysis, but was a necessary step. We have attempted to mine the data for useful and relevant patterns, to substantiate our suggestions to the Singapore Police Force and Housing Development Board. We encountered limitations along the way, which are discussed later. We also brainstormed ideal circumstances in terms of data available to us, to construct an ideal crime prediction model.

Data Description

The data is primarily geospatial data with a few additional attributes describing the type of crime. To be systematic, we segmented the data into two types – record and ordered. The record data was represented by descriptions of the crime including the time and location of occurrence. The ordered data was represented by geospatial data that allowed us to plot the crimes on a map of America, for a visual overview of the data.

The dataset contained a number of records, ensuring an appropriately large size to mine. The dataset contained the information about the crimes that occurred in America from 2012-2015. The attributes are as follows:

Attribute name:	Format of data:	Data type:
ID	5584223	Bigint
Case_Number	HN386585	String
Date	06/05/2007 02:15:00 PM	String
Block	0000X N WESTERN AVE	String
IUCR	0430	String
Primary_Type	BATTERY	String
Description	AGGRAVATED: OTHER DANG WEAPON	String

Location_description	STREET	String
Arrest	False	String, Boolean
Domestic	False	Boolean
Beat	1332	Int
District	012	Int
Ward	2	Int
Community_Area	28	Int
FBI_Code	04B	String
X_Coordinate	1160411	Int
Y_Coordinate	1900033	Int
Year	2007	Int
Updated_On	04/15/2016 08:55:02 AM	String
Latitude	41.881391643	Double
Longitude	87.686437851	Double
Location	"(41.881391643, -87.686437851)"	String

Data analysis

$ start-all.sh

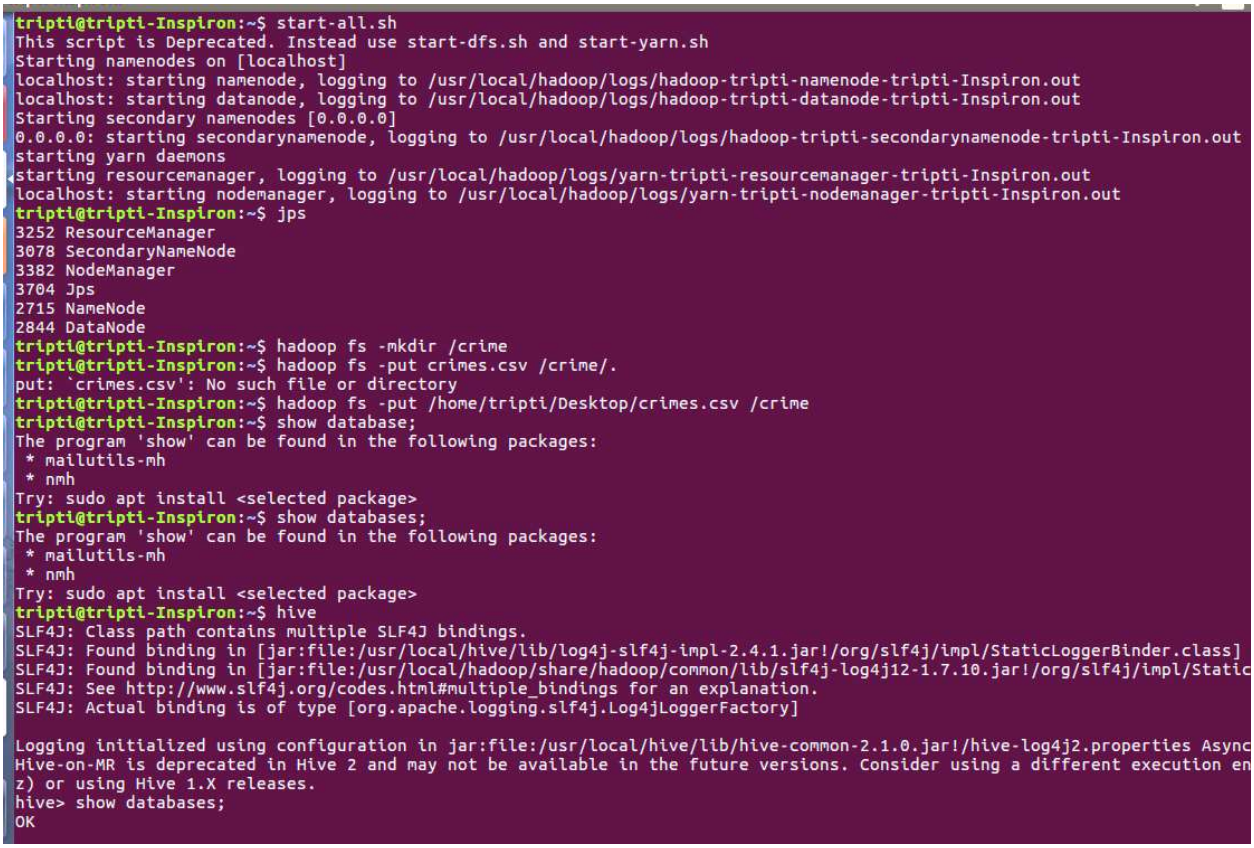
$ jps

$ hadoop fs -mkdir /crime

//this command is used for creating file directory in HDFS.

$ hadoop fs -put /home/tripti/Desktop/crimes.csv /crime/.

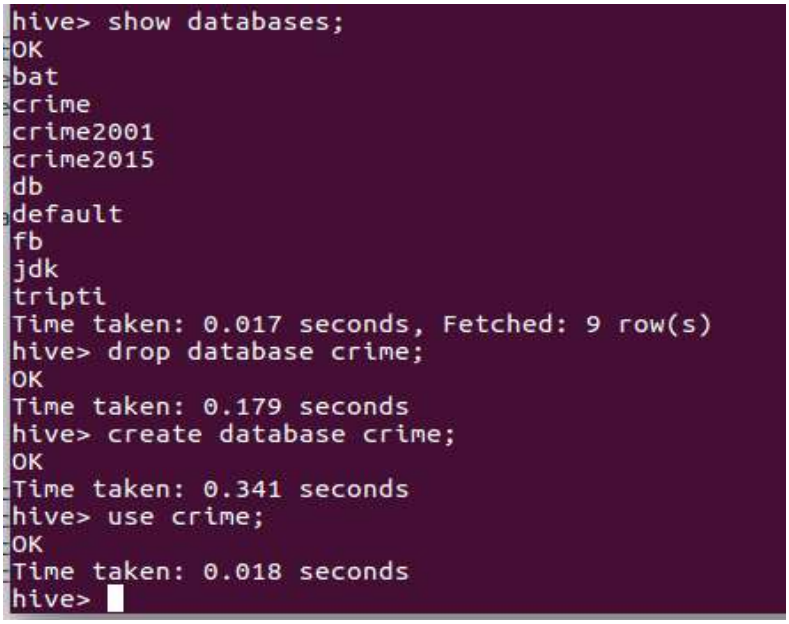
//This command is used for putting crimes.csv data from local to HDFS directory which is named as crime.



hive> show databases;

hive> create database crime;

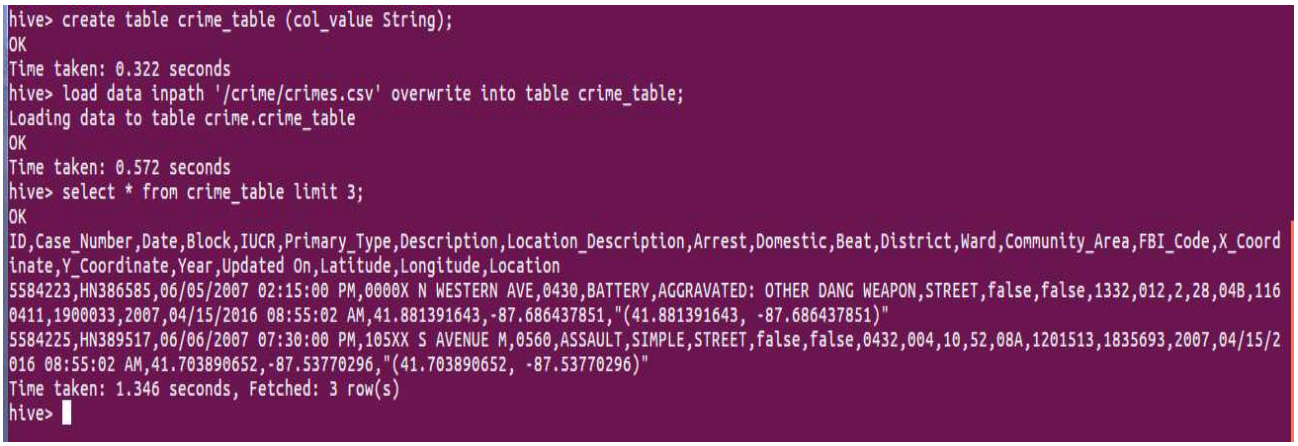
hive> use crime;



hive> create table crime_table (col_value String);

hive> load data inpath '/crime/crimes.csv' overwrite into table crime_table;

hive> select * from crime_table limit 3;



hive> create table crime1 (Case_Number String, Description String, Arrest String, Year int);

hive> insert overwrite table crime1 SELECT

regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number,

regexp_extract (col_value, '^(?:([^,]*)\,?){7}',1) Description,

regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) Arrest,

regexp_extract (col_value, '^(?:([^,]*)\,?){18}',1) Year from crime_table;

```
hive> create table crime1 (Case_Number String, Description String, Arrest String, Year int);
OK
Time taken: 0.121 seconds
hive> insert overwrite table crime1 SELECT
    > regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){7}',1) Description,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) Arrest,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){18}',1) Year from crime_table;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different ex
ecution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704125745_7fb36b92-7808-4b08-a4a0-8aa0f2b6e6fa
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
```

```
tripti@tripti-Inspiron: ~
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499151932442_0001, Tracking URL = http://tripti-Inspiron:8088/proxy/applicatio
n_1499151932442_0001/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0001
Hadoop job information for Stage-1: number of mappers: 6; number of reducers: 0
2017-07-04 12:58:03,131 Stage-1 map = 0%,  reduce = 0%
2017-07-04 12:59:04,121 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 38.32 sec
2017-07-04 12:59:59,845 Stage-1 map = 8%,  reduce = 0%, Cumulative CPU 130.99 sec
2017-07-04 13:00:00,954 Stage-1 map = 17%,  reduce = 0%, Cumulative CPU 132.03 sec
2017-07-04 13:00:02,002 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 133.19 sec
2017-07-04 13:00:03,037 Stage-1 map = 42%,  reduce = 0%, Cumulative CPU 134.72 sec
2017-07-04 13:00:07,200 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU 140.96 sec
2017-07-04 13:00:46,332 Stage-1 map = 58%,  reduce = 0%, Cumulative CPU 194.37 sec
2017-07-04 13:00:48,398 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 197.23 sec
2017-07-04 13:00:50,451 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 200.19 sec
2017-07-04 13:00:53,559 Stage-1 map = 92%,  reduce = 0%, Cumulative CPU 202.68 sec
2017-07-04 13:01:09,136 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 213.23 sec
MapReduce Total cumulative CPU time: 3 minutes 33 seconds 230 msec
Ended Job = job_1499151932442_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/crime.db/crime1/.hive-staging_h
ive_2017-07-04_12-57-45_219_5719128572028611698-1/-ext-10000
Loading data to table crime.crime1
MapReduce Jobs Launched:
Stage-Stage-1: Map: 6   Cumulative CPU: 213.42 sec   HDFS Read: 1478863916 HDFS Write: 230882071 S
UCCESS
Total MapReduce CPU Time Spent: 3 minutes 33 seconds 420 msec
OK
Time taken: 221.239 seconds
hive>
```

hive> select * from crime1 limit 10;

```
hive> select * from crime1 limit 10;
OK
Case_Number     Description     Arrest  NULL
HN386585        AGGRAVATED: OTHER DANG WEAPON   false   2007
HN389517        SIMPLE  false   2007
HN391344        $500 AND UNDER  false   2007
HN389729        TO LAND true    2007
HN389881        SIMPLE  false   2007
HN386855        TO VEHICLE      false   2007
HN391489        SIMPLE  true    2007
HN383578        $500 AND UNDER  true    2007
HN376323        TELEPHONE THREAT        false   2007
Time taken: 0.232 seconds, Fetched: 10 row(s)
hive>
```

hive> create table crime_part1 (Case_Number String) partitioned by (Year int);

hive> set hive.exec.dynamic.partition.mode=nonstrict;

hive> insert overwrite table crime_part1 partition (Year) select Case_Number, Year from crime1 where Year between 2000 and 2017;

```
hive> create table crime_part1 (Case_Number String) partitioned by (Year int);
OK
Time taken: 0.259 seconds
hive> set hive.exec.dynamic.partition.mode=nonstrict
    > ;
hive> insert overwrite table crime_part1 partition (Year) select Case_Number, Year  from crime1 wh
ere Year between 2000 and 2017;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
der using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704130442_f499fd7e-7d9d-4084-823f-d67e6a24fb0d
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499151932442_0002, Tracking URL = http://tripti-Inspiron:8088/proxy/applicatio
n_1499151932442_0002/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-07-04 13:04:55,019 Stage-1 map = 0%,  reduce = 0%
2017-07-04 13:05:05,540 Stage-1 map = 18%,  reduce = 0%, Cumulative CPU 10.76 sec
2017-07-04 13:05:08,684 Stage-1 map = 55%,  reduce = 0%, Cumulative CPU 14.26 sec
2017-07-04 13:05:11,795 Stage-1 map = 91%,  reduce = 0%, Cumulative CPU 17.41 sec
2017-07-04 13:05:14,906 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 18.29 sec
MapReduce Total cumulative CPU time: 18 seconds 290 msec
Ended Job = job_1499151932442_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
```

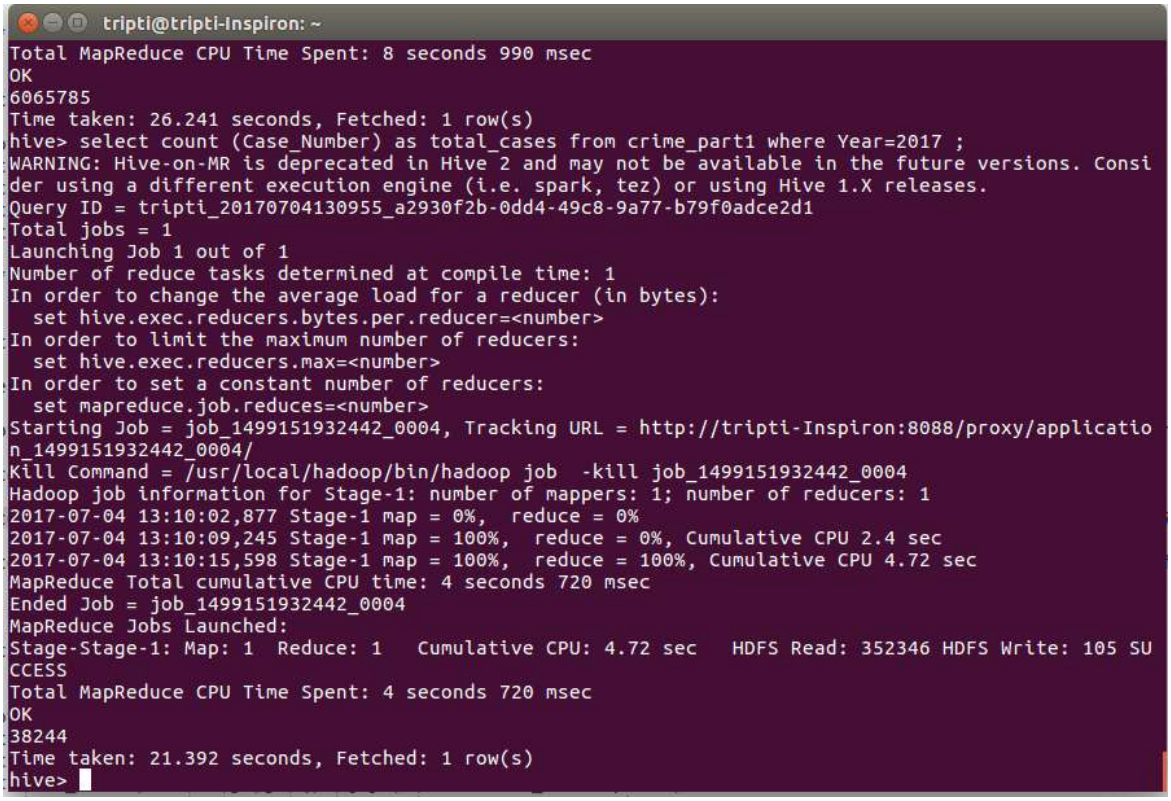
```
tripti@tripti-Inspiron: ~
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
der using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704130442_f499fd7e-7d9d-4084-823f-d67e6a24fb0d
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499151932442_0002, Tracking URL = http://tripti-Inspiron:8088/proxy/applicatio
n_1499151932442_0002/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-07-04 13:04:55,019 Stage-1 map = 0%,  reduce = 0%
2017-07-04 13:05:05,540 Stage-1 map = 18%,  reduce = 0%, Cumulative CPU 10.76 sec
2017-07-04 13:05:08,684 Stage-1 map = 55%,  reduce = 0%, Cumulative CPU 14.26 sec
2017-07-04 13:05:11,795 Stage-1 map = 91%,  reduce = 0%, Cumulative CPU 17.41 sec
2017-07-04 13:05:14,906 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 18.29 sec
MapReduce Total cumulative CPU time: 18 seconds 290 msec
Ended Job = job_1499151932442_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/crime.db/crime_part1/.hive-stag
ing_hive_2017-07-04_13-04-42_755_7477226428722552921-1/-ext-10000
Loading data to table crime.crime_part1 partition (year=null)

Loaded : 17/17 partitions.
        Time taken to load dynamic partitions: 6.435 seconds
        Time taken for adding to write entity : 0.003 seconds
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 18.29 sec   HDFS Read: 230886828 HDFS Write: 54129404 SUCC
ESS
Total MapReduce CPU Time Spent: 18 seconds 290 msec
OK
Time taken: 43.029 seconds
hive>
```

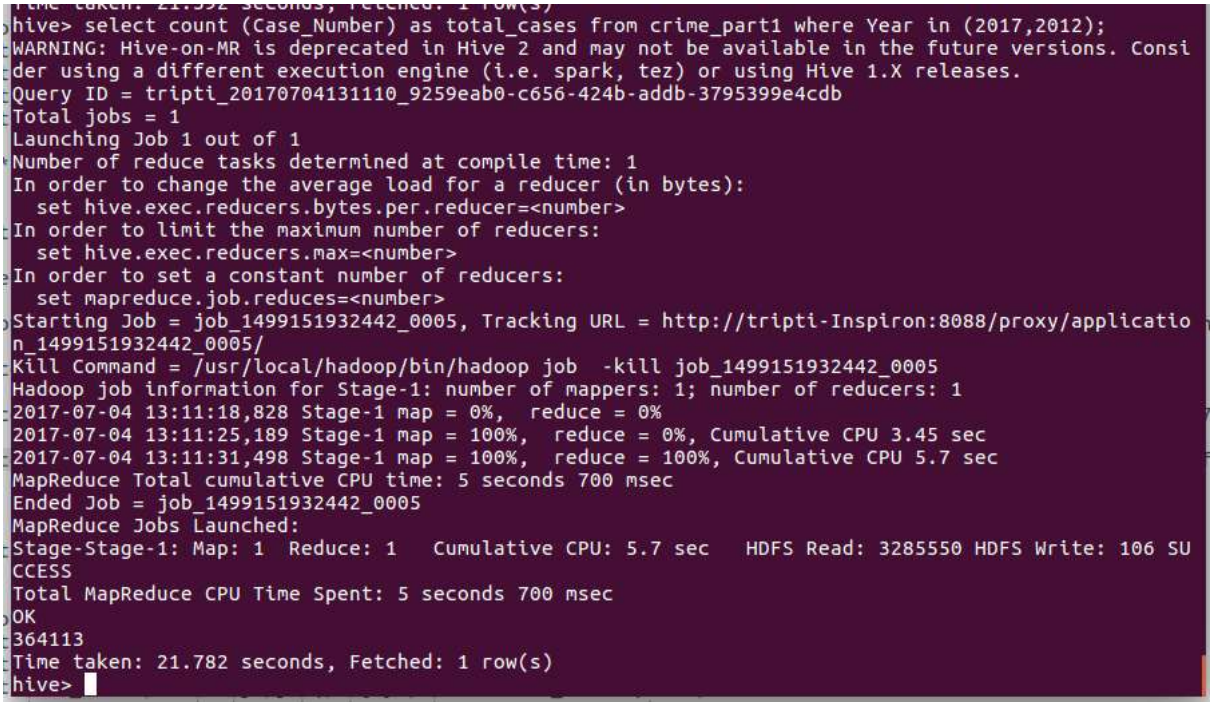
hive> select count (Case_Number) as total_cases from crime_part1 ;

```
tripti@tripti-Inspiron: ~
ESS
Total MapReduce CPU Time Spent: 18 seconds 290 msec
OK
Time taken: 43.029 seconds
hive> select count (Case_Number) as total_cases from crime_part1 ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
der using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704130649_fc987d56-f6ec-4d4d-8083-d008dca67fee
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499151932442_0003, Tracking URL = http://tripti-Inspiron:8088/proxy/applicatio
n_1499151932442_0003/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-07-04 13:06:58,239 Stage-1 map = 0%,  reduce = 0%
2017-07-04 13:07:07,703 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.83 sec
2017-07-04 13:07:14,071 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.99 sec
MapReduce Total cumulative CPU time: 8 seconds 990 msec
Ended Job = job_1499151932442_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 8.99 sec   HDFS Read: 54142730 HDFS Write: 107
SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 990 msec
OK
6065785
Time taken: 26.241 seconds, Fetched: 1 row(s)
hive>
```

hive> select count (Case_Number) as total_cases from crime_part1 where Year=2017 ;



hive> select count (Case_Number) as total_cases from crime_part1 where Year in (2017,2012);



hive> CREATE INDEX case2 ON TABLE crime1(case_number) AS 'COMPACT' WITH DEFERRED REBUILD;

hive> SHOW INDEX ON crime1;

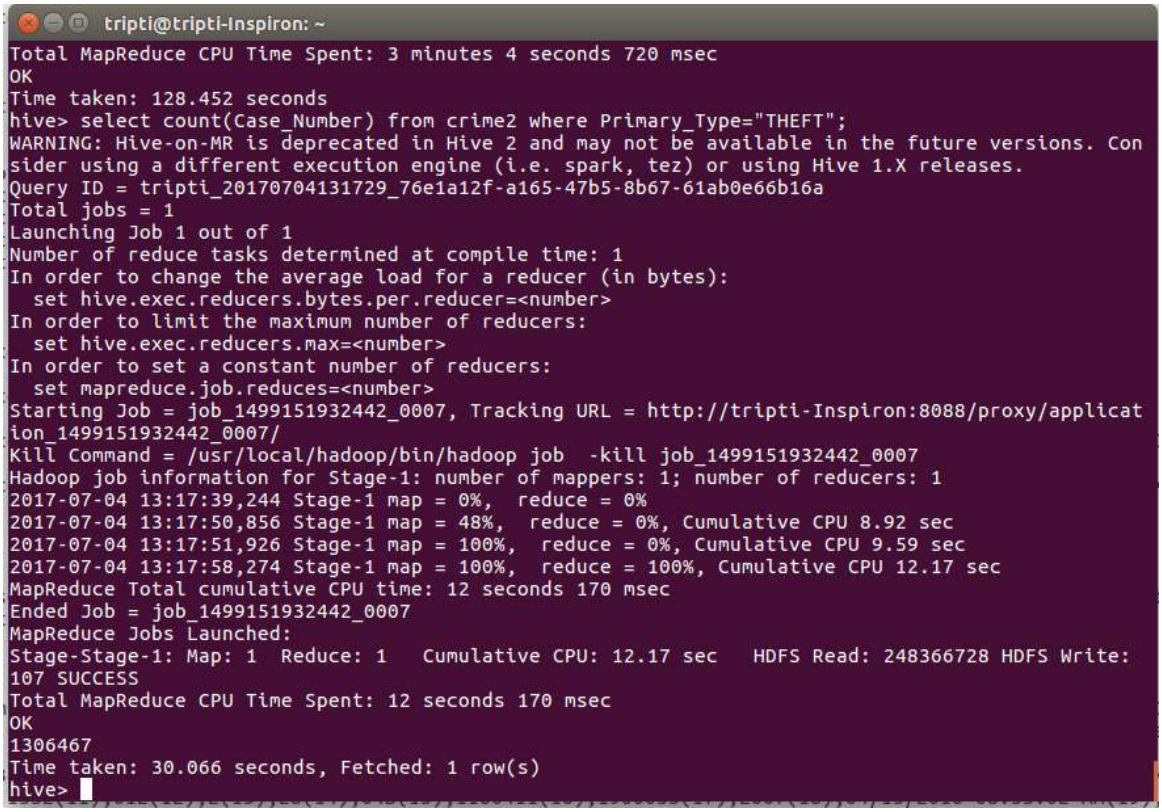
```
hive> CREATE INDEX case2 ON TABLE crime1(case_number)AS 'COMPACT' WITH DEFERRED
REBUILD;
OK
Time taken: 1.411 seconds
hive>  SHOW INDEX ON crime1;
OK
case1                   crime1                  case_number             crime__c
rime1_case1__    compact
case2                   crime1                  case_number             crime__c
rime1_case2__    compact
Time taken: 0.104 seconds, Fetched: 2 row(s)
hive>
```

hive> create table crime2 (Case_Number String, Primary_Type String, Location_Description String, ID int);

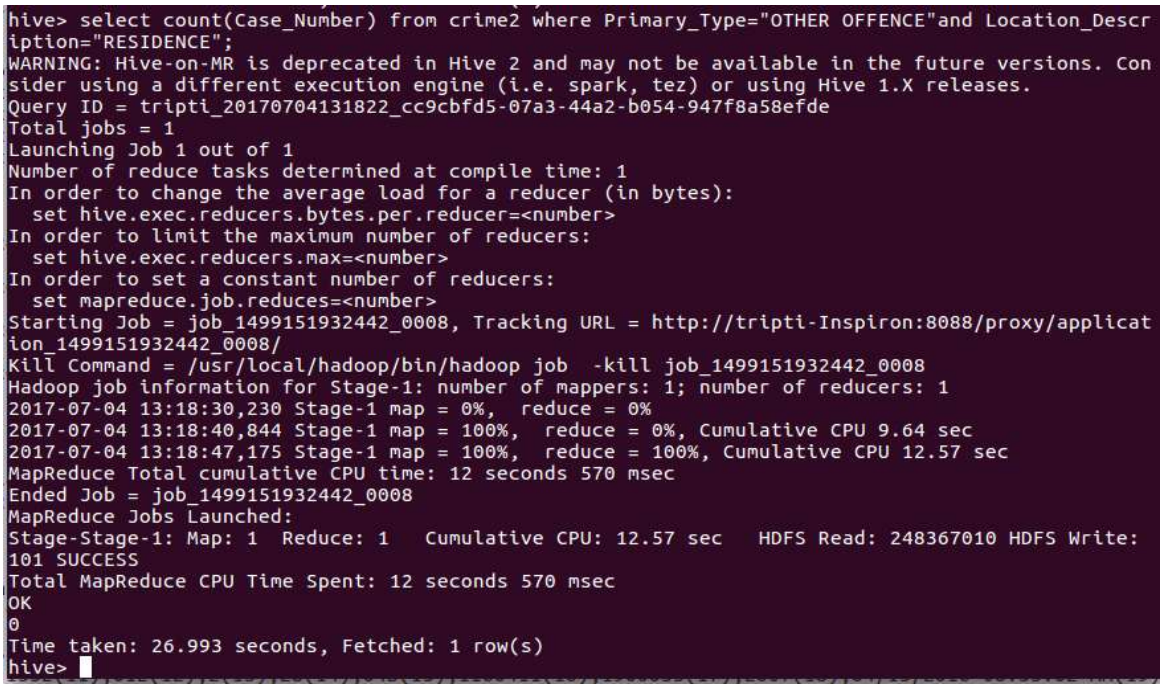
hive> insert overwrite table crime2 SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number, regexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type, regexp_extract (col_value, '^(?:([^,]*)\,?){8}',1) Location_Description, regexp_extract (col_value, '^(?:([^,]*)\,?){1}',1) ID from crime_table;

```
hive> create table crime2 (Case_Number String, Primary_Type String, Location_Description String, I
D int);
OK
Time taken: 0.15 seconds
hive> insert overwrite table crime2 SELECT
    > regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){8}',1) Location_Description,
    > regexp_extract (col_value, '^(?:([^,]*)\,?){1}',1) ID from crime_table;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consi
der using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704131226_cd336ed0-79fb-48f5-a780-6cacae206ed0
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499151932442_0006, Tracking URL = http://tripti-Inspiron:8088/proxy/applicatio
n_1499151932442_0006/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0006
Hadoop job information for Stage-1: number of mappers: 6; number of reducers: 0
2017-07-04 13:12:33,152 Stage-1 map = 0%,  reduce = 0%
2017-07-04 13:13:33,846 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 95.14 sec
2017-07-04 13:13:47,624 Stage-1 map = 58%,  reduce = 0%, Cumulative CPU 122.56 sec
2017-07-04 13:14:22,377 Stage-1 map = 92%,  reduce = 0%, Cumulative CPU 182.22 sec
2017-07-04 13:14:24,436 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 183.48 sec
MapReduce Total cumulative CPU time: 3 minutes 3 seconds 480 msec
Ended Job = job_1499151932442_0006
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/crime.db/crime2/.hive-staging_h
ive_2017-07-04_13-12-26_775_4595227895259065677-1/-ext-10000
```

hive> select count(Case_Number) from crime2 where Primary_Type="THEFT";



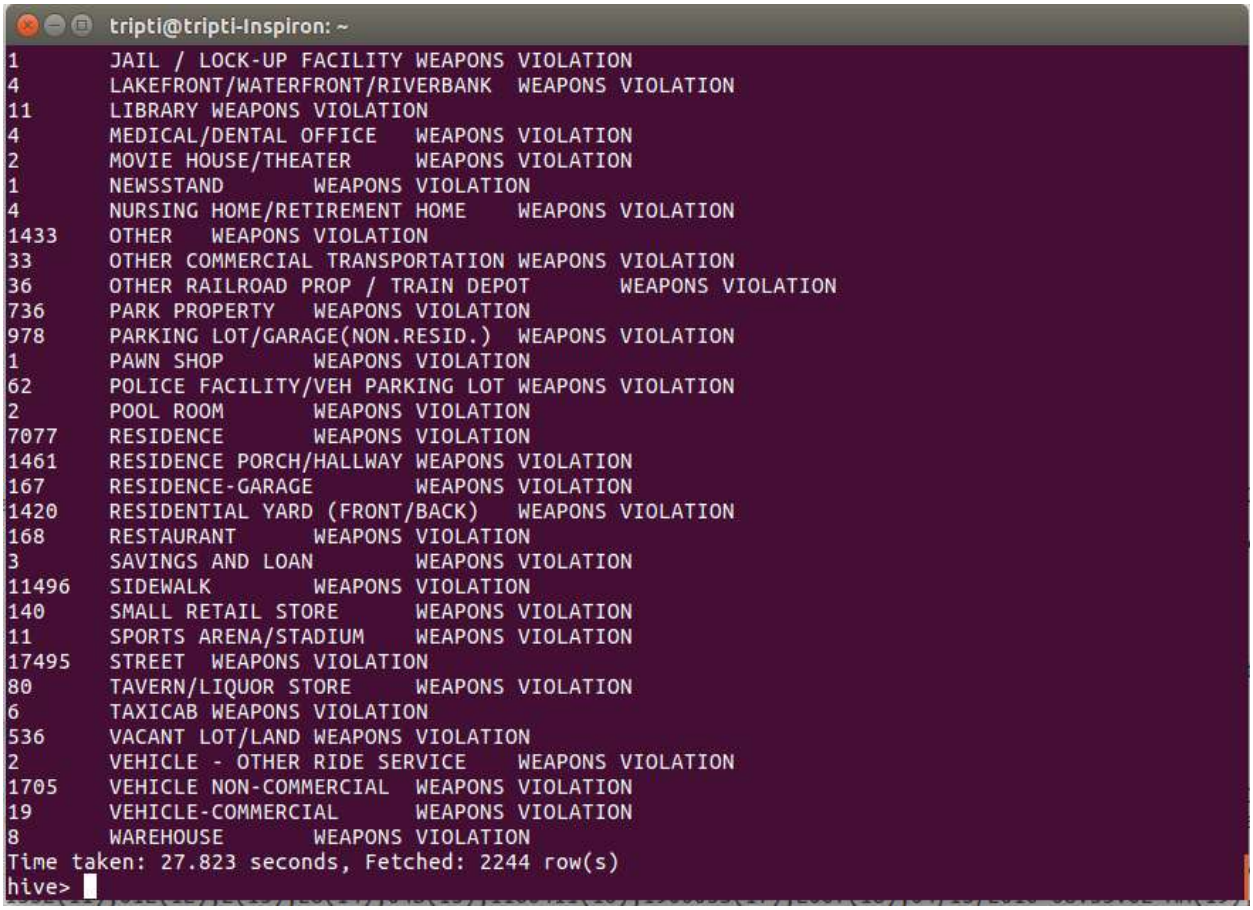
hive> select count(Case_Number) from crime2 where Primary_Type="OTHER OFFENCE"and Location_Description="RESIDENCE";



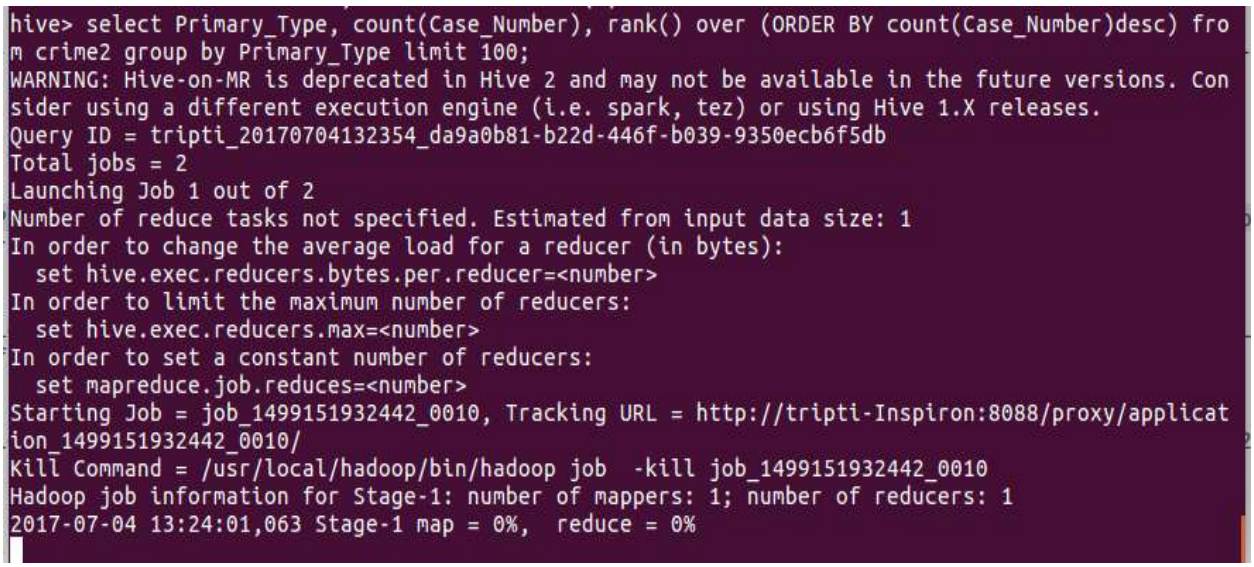
hive> select count(Case_Number),Location_Description,Primary_Type from crime2 group by Location_Description,Primary_Type;

```
hive> select count(Case_Number),Location_Description,Primary_Type from crime2 group by Location_
Description,Primary_Type;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Con
sider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170704132022_479842a2-b3fd-40a2-b68c-730252402fc8
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499151932442_0009, Tracking URL = http://tripti-Inspiron:8088/proxy/applicat
ion_1499151932442_0009/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-07-04 13:20:30,704 Stage-1 map = 0%,  reduce = 0%
2017-07-04 13:20:41,144 Stage-1 map = 48%,  reduce = 0%, Cumulative CPU 8.77 sec
2017-07-04 13:20:42,201 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.76 sec
2017-07-04 13:20:48,524 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.78 sec
MapReduce Total cumulative CPU time: 12 seconds 780 msec
Ended Job = job_1499151932442_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.78 sec   HDFS Read: 248367207 HDFS Write:
110722 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 780 msec
OK
241     "SCHOOL ARSON
```

```
tripti@tripti-Inspiron: ~
Total MapReduce CPU Time Spent: 12 seconds 780 msec
OK
241     "SCHOOL ARSON
199     ABANDONED BUILDING      ARSON
1       AIRPORT EXTERIOR - NON-SECURE AREA      ARSON
355     ALLEY   ARSON
2       ANIMAL HOSPITAL ARSON
808     APARTMENT       ARSON
1       APPLIANCE STORE ARSON
1       ATM (AUTOMATIC TELLER MACHINE)  ARSON
3       BANK    ARSON
33      BAR OR TAVERN   ARSON
12      BARBERSHOP      ARSON
2       BOAT/WATERCRAFT ARSON
5       CAR WASH        ARSON
60      CHA APARTMENT   ARSON
21      CHA HALLWAY/STAIRWELL/ELEVATOR  ARSON
14      CHA PARKING LOT/GROUNDS ARSON
49      CHURCH/SYNAGOGUE/PLACE OF WORSHIP       ARSON
2       CLEANING STORE  ARSON
8       COLLEGE/UNIVERSITY GROUNDS      ARSON
1       COLLEGE/UNIVERSITY RESIDENCE HALL       ARSON
53      COMMERCIAL / BUSINESS OFFICE    ARSON
34      CONSTRUCTION SITE       ARSON
19      CONVENIENCE STORE       ARSON
1       CREDIT UNION    ARSON
1       CTA BUS ARSON
4       CTA GARAGE / OTHER PROPERTY     ARSON
4       CTA PLATFORM    ARSON
1       CTA STATION     ARSON
2       CTA TRAIN       ARSON
6       CURRENCY EXCHANGE       ARSON
2       DAY CARE CENTER ARSON
3       DELIVERY TRUCK  ARSON
```



hive> select Primary_Type, count(Case_Number), rank() over (ORDER BY count(Case_Number)desc) from crime2 group by Primary_Type limit 100;



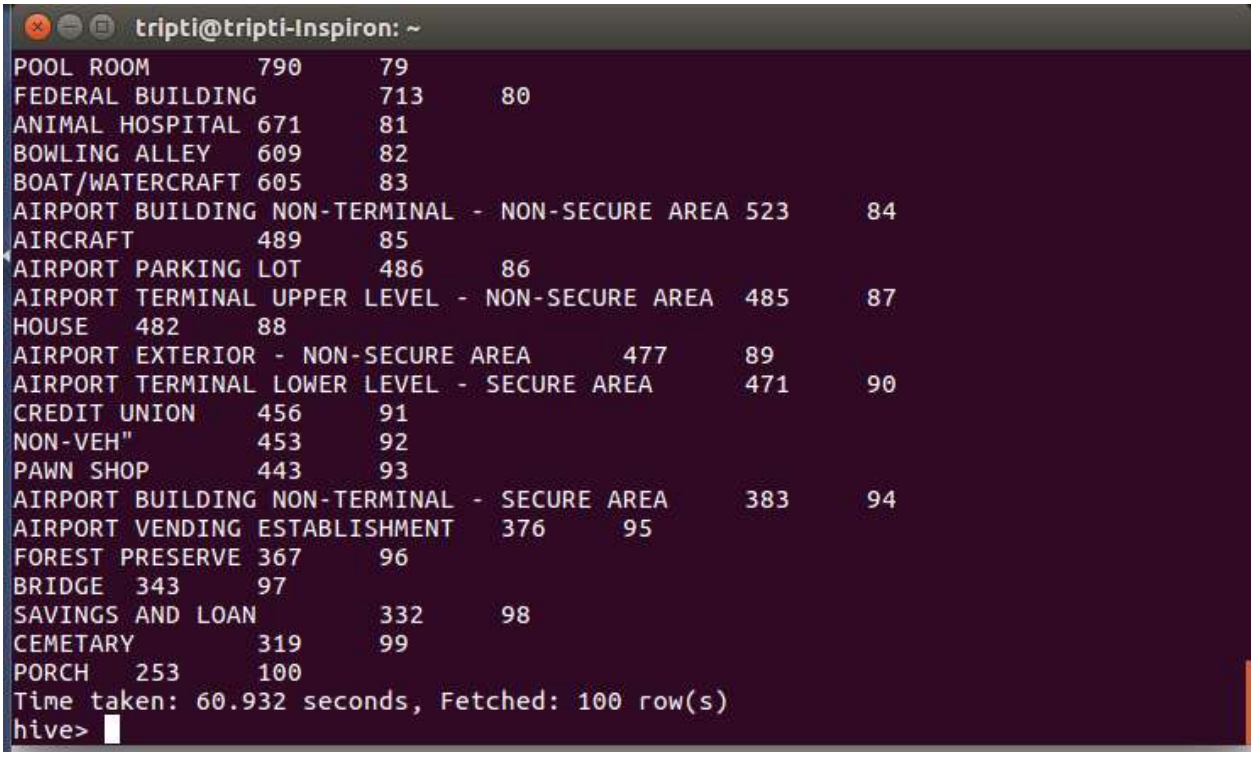
```
tripti@tripti-Inspiron: ~
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499151932442_0011, Tracking URL = http://tripti-Inspiron:8088/proxy/applicat
ion_1499151932442_0011/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499151932442_0011
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2017-07-04 13:24:30,724 Stage-2 map = 0%,  reduce = 0%
2017-07-04 13:24:35,092 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.21 sec
2017-07-04 13:24:41,413 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 3.54 sec
MapReduce Total cumulative CPU time: 3 seconds 540 msec
Ended Job = job_1499151932442_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.52 sec   HDFS Read: 248365576 HDFS Write:
1375 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.54 sec   HDFS Read: 9372 HDFS Write: 1376 S
UCCESS
Total MapReduce CPU Time Spent: 14 seconds 60 msec
OK
THEFT   1306467 1
BATTERY 1146260 2
CRIMINAL DAMAGE 722513  3
NARCOTICS       688885  4
OTHER OFFENSE   389143  5
ASSAULT 383983  6
BURGLARY        367048  7
MOTOR VEHICLE THEFT     296396  8
ROBBERY 237240  9
DECEPTIVE PRACTICE      227824  10
CRIMINAL TRESPASS       181608  11
```

```
tripti@tripti-Inspiron: ~
OTHER OFFENSE   389143  5
ASSAULT 383983  6
BURGLARY        367048  7
MOTOR VEHICLE THEFT     296396  8
ROBBERY 237240  9
DECEPTIVE PRACTICE      227824  10
CRIMINAL TRESPASS       181608  11
PROSTITUTION    67097   12
WEAPONS VIOLATION       61881   13
PUBLIC PEACE VIOLATION  45328   14
OFFENSE INVOLVING CHILDREN      40980   15
CRIM SEXUAL ASSAULT     24023   16
SEX OFFENSE     23048   17
GAMBLING        14045   18
LIQUOR LAW VIOLATION    13650   19
INTERFERENCE WITH PUBLIC OFFICER        13069   20
ARSON   10468   21
HOMICIDE        8334    22
KIDNAPPING      6354    23
INTIMIDATION    3658    24
STALKING        3037    25
OBSCENITY       422     26
PUBLIC INDECENCY        142     27
OTHER NARCOTIC VIOLATION        112     28
NON-CRIMINAL    97      29
CONCEALED CARRY LICENSE VIOLATION       95      30
NON - CRIMINAL  38      31
HUMAN TRAFFICKING       28      32
RITUALISM       23      33
NON-CRIMINAL (SUBJECT SPECIFIED)        5       34
DOMESTIC VIOLENCE       1       35
Primary_Type    1       35
Time taken: 49.239 seconds, Fetched: 36 row(s)
hive>
```

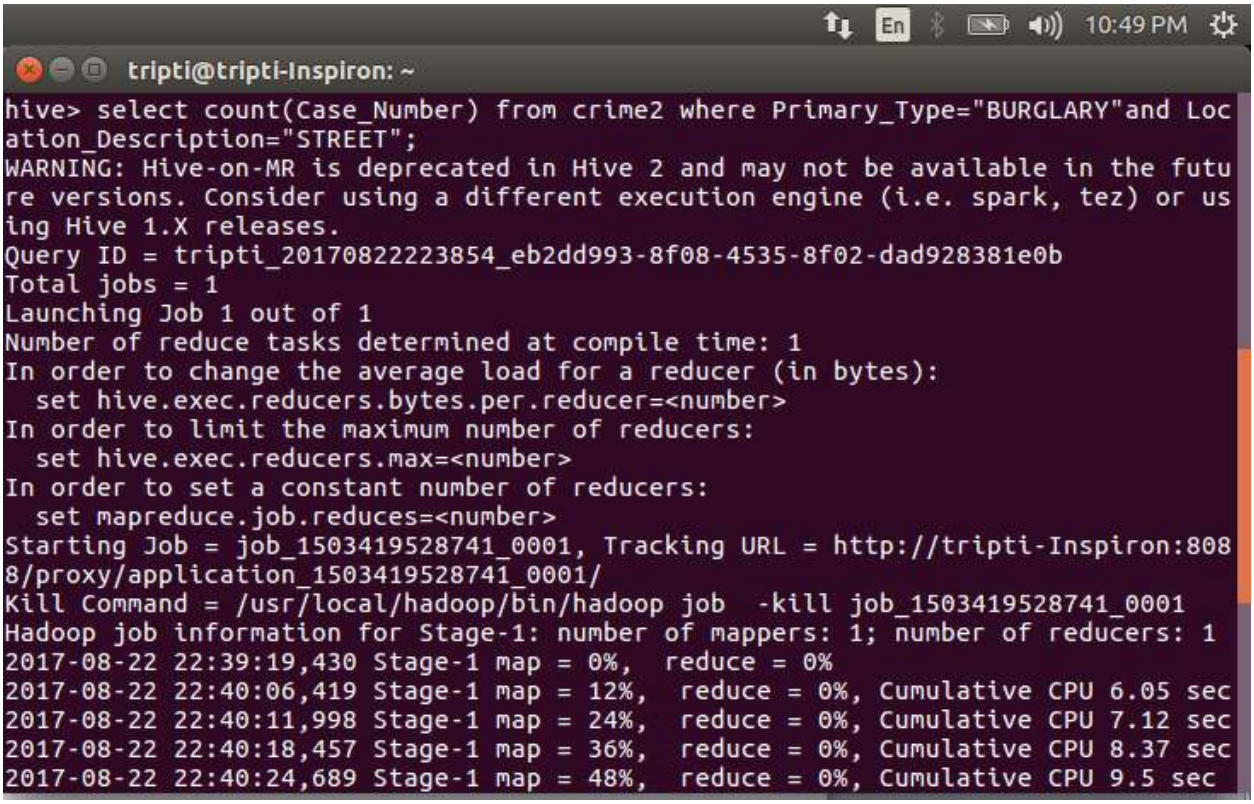
Hive> select Location_Description, count(Case_Number) from crime2 where Location_Description="STREET" group by Location_Description limit 100;

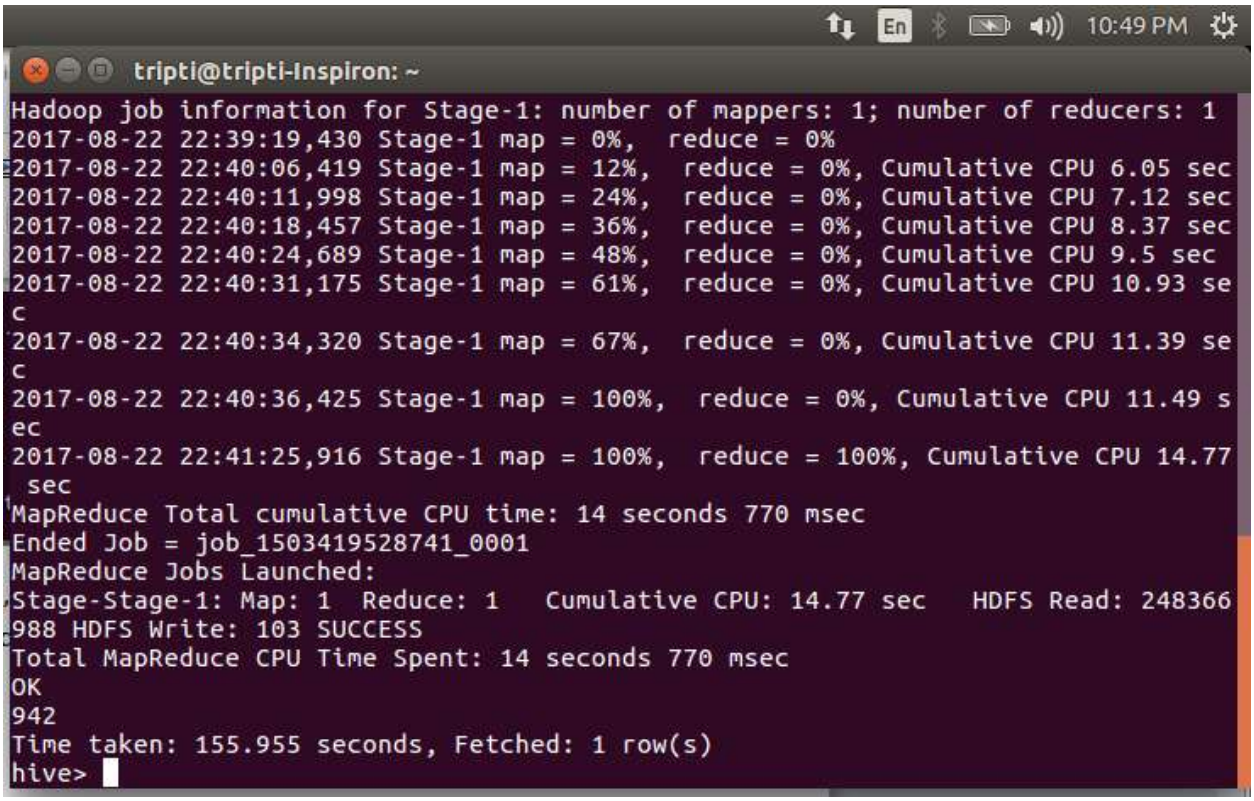
```
hive> select Location_Description, count(Case_Number) from crime2 where Location
_Description="STREET" group by Location_Description limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170704133958_80220aea-2cb9-4c7e-862d-c6dac9a9c75d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499155284503_0001, Tracking URL = http://tripti-Inspiron:808
8/proxy/application_1499155284503_0001/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499155284503_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-07-04 13:40:11,954 Stage-1 map = 0%,  reduce = 0%
```

```
tripti@tripti-Inspiron: ~
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2017-07-04 14:15:23,491 Stage-2 map = 0%,  reduce = 0%
2017-07-04 14:15:28,811 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.48 se
c
2017-07-04 14:15:35,206 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.08
sec
MapReduce Total cumulative CPU time: 4 seconds 80 msec
Ended Job = job_1499156353388_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.51 sec   HDFS Read: 248365
612 HDFS Write: 6462 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.08 sec   HDFS Read: 14490 H
DFS Write: 3992 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 590 msec
OK
STREET  1643607 1
RESIDENCE       1063060 2
APARTMENT       638999  3
SIDEWALK        627949  4
OTHER   234485  5
"SCHOOL 179826  6
PARKING LOT/GARAGE(NON.RESID.)  176025  7
ALLEY   141179  8
RESIDENCE-GARAGE        123954  9
```

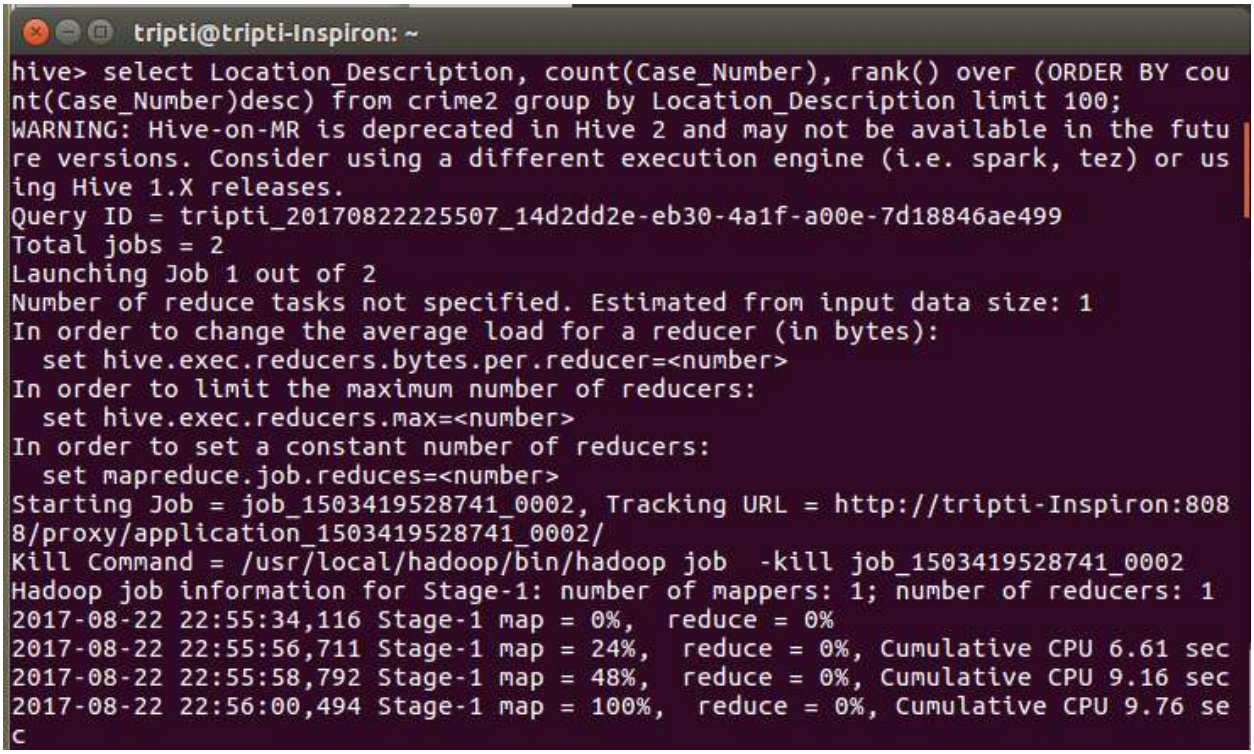


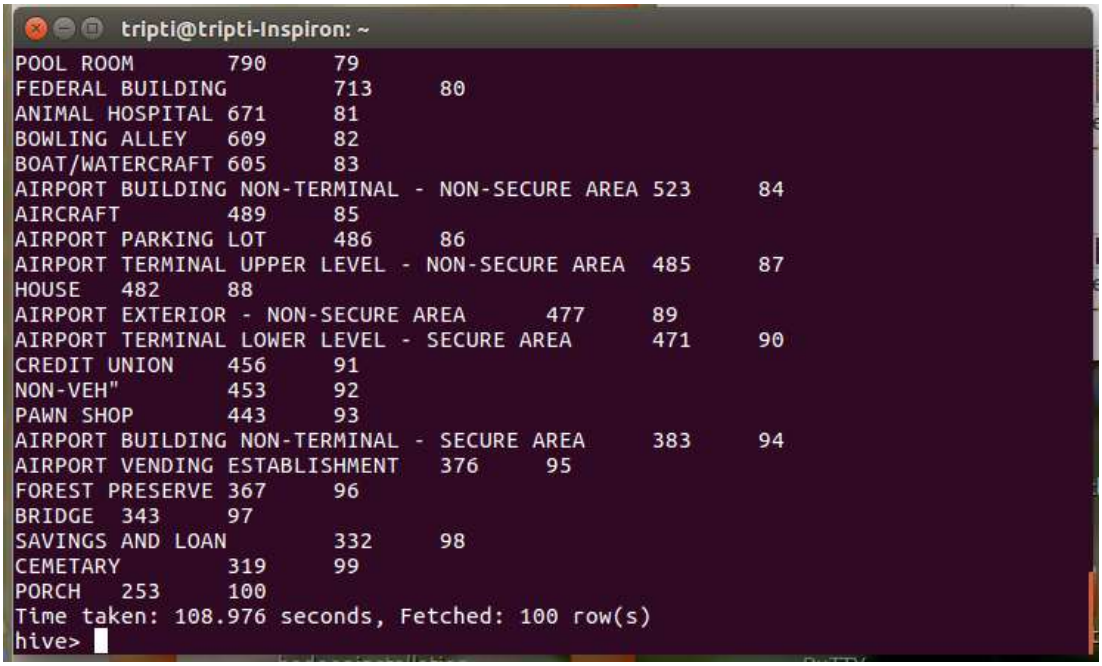
hive> select count(Case_Number) from crime2 where Primary_Type="BURGLARY"and Location_Description="STREET";





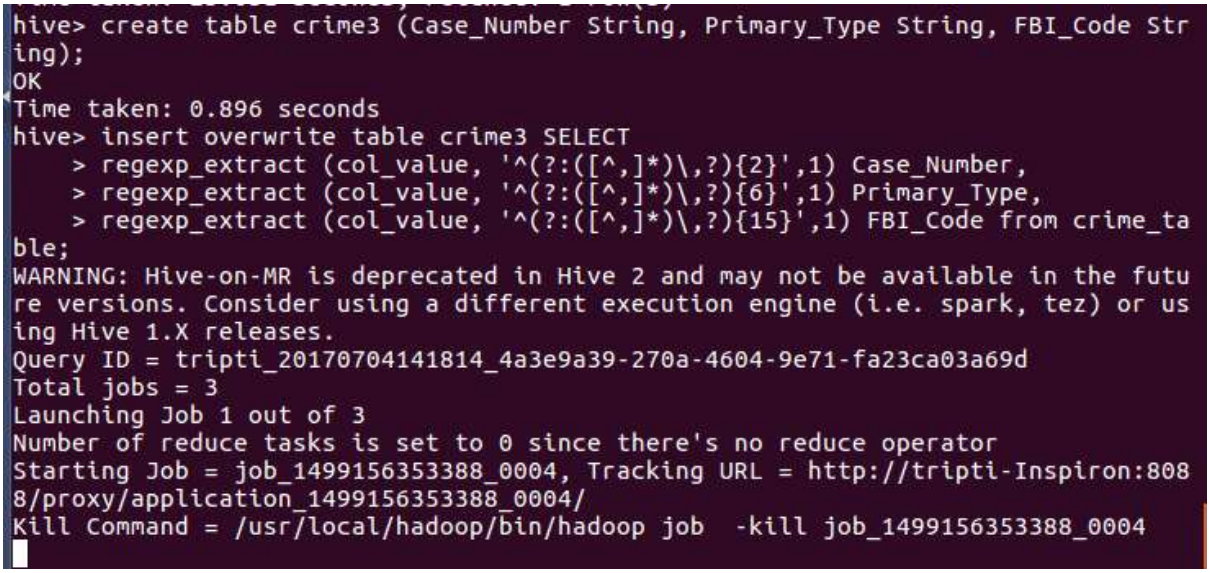
hive> select Location_Description, count(Case_Number), rank() over (ORDER BY count(Case_Number)desc) from crime2 group by Location_Description limit 100;





hive> create table crime4 (Case_Number String, Primary_Type String, Description String, Year int, Arrest String, FBI_Code String);

hive> insert overwrite table crime4 SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number, regexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type, regexp_extract (col_value, '^(?:([^,]*)\,?){7}',1) Description, regexp_extract (col_value, '^(?:([^,]*)\,?){18}',1) Year, regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) Arrest, regexp_extract (col_value, '^(?:([^,]*)\,?){15}',1) FBI_Code from crime_table;



hive> select FBI_Code, count(FBI_Code) as count from Crime3 group by FBI_Code;

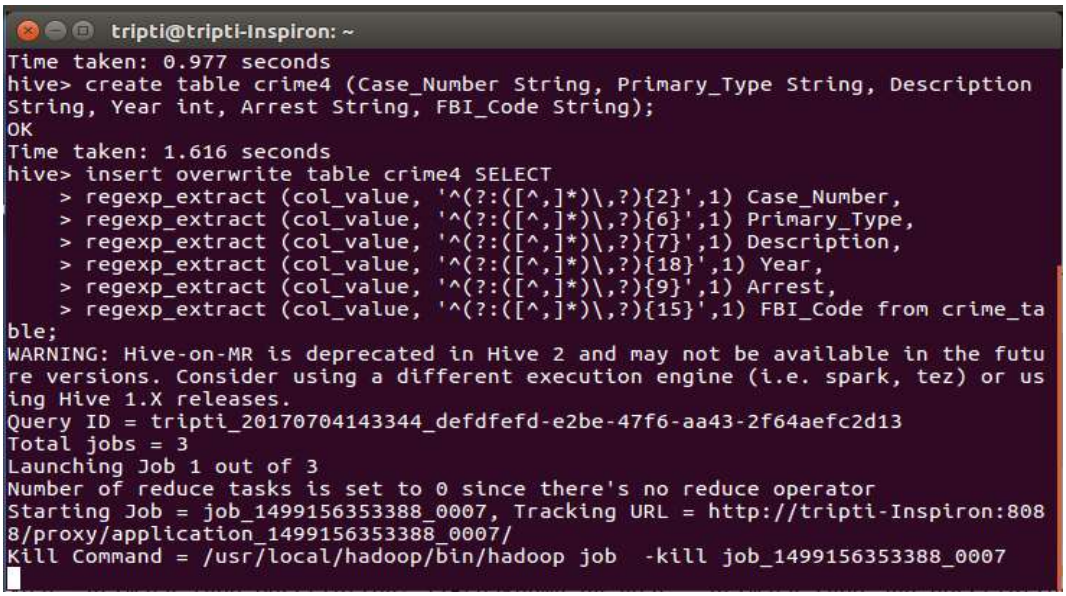
```
hive> select FBI_Code, count(FBI_Code) as count from Crime3 group by FBI_Code;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170704143055_46f86958-c716-4936-b221-bc577b887f30
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
```

```
tripti@tripti-Inspiron: ~
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499156353388_0006, Tracking URL = http://tripti-Inspiron:808
8/proxy/application_1499156353388_0006/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499156353388_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-07-04 14:31:22,492 Stage-1 map = 0%,  reduce = 0%
2017-07-04 14:31:34,383 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 8.63 se
c
2017-07-04 14:31:42,817 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.29
 sec
MapReduce Total cumulative CPU time: 11 seconds 290 msec
Ended Job = job_1499156353388_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.29 sec   HDFS Read: 145017
418 HDFS Write: 2933 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 290 msec
OK
        21832
0       1
005     1
008     1
018     1
```

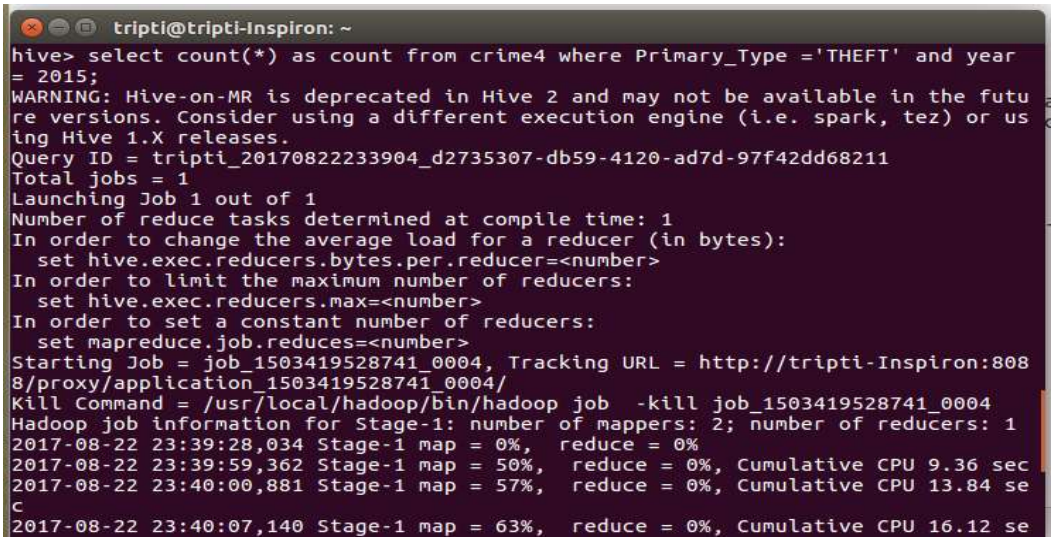
```
tripti@tripti-Inspiron: ~
60      6
61      60
62      21
63      5
64      107
65      1
66      58
67      9
68      11
69      11
7       5250
70      28
71      19
72      5
73      4
74      3
75      1
76      1062
77      20
8       9605
9       5959
FBI_Code        1
Time taken: 51.018 seconds, Fetched: 146 row(s)
hive>
```

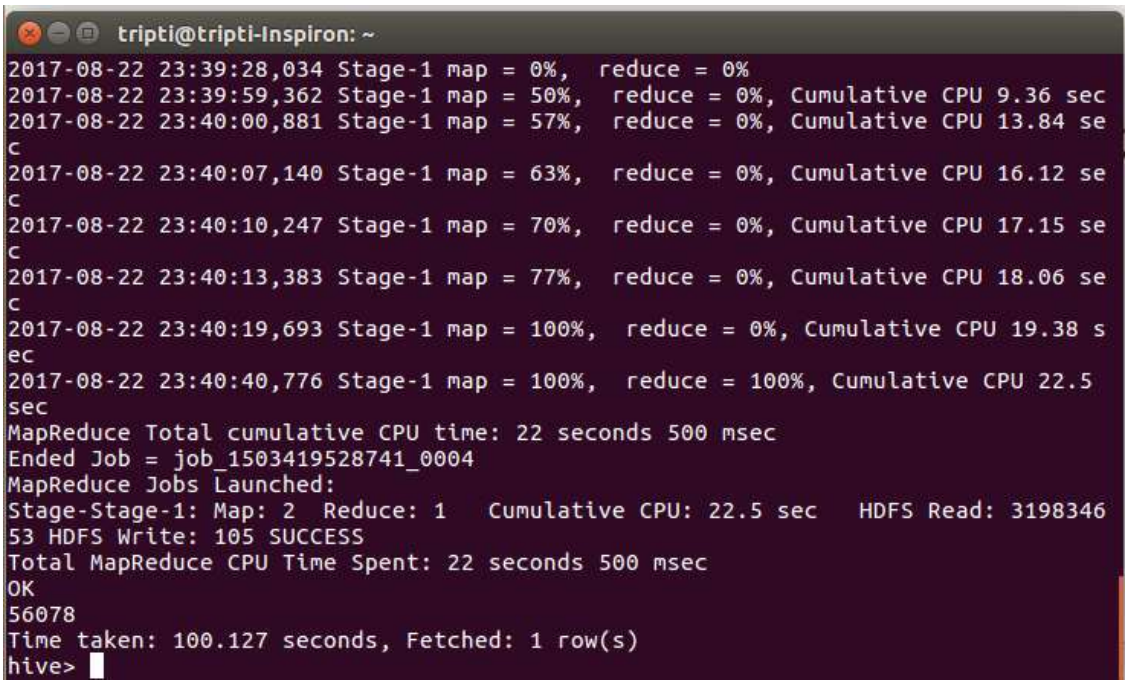
hive> create table crime4 (Case_Number String, Primary_Type String, Description String, Year int, Arrest String, FBI_Code String);

hive> insert overwrite table crime4 SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number, regexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type, regexp_extract (col_value, '^(?:([^,]*)\,?){7}',1) Description, regexp_extract (col_value, '^(?:([^,]*)\,?){18}',1) Year, regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) Arrest, regexp_extract (col_value, '^(?:([^,]*)\,?){15}',1) FBI_Code from crime_table;

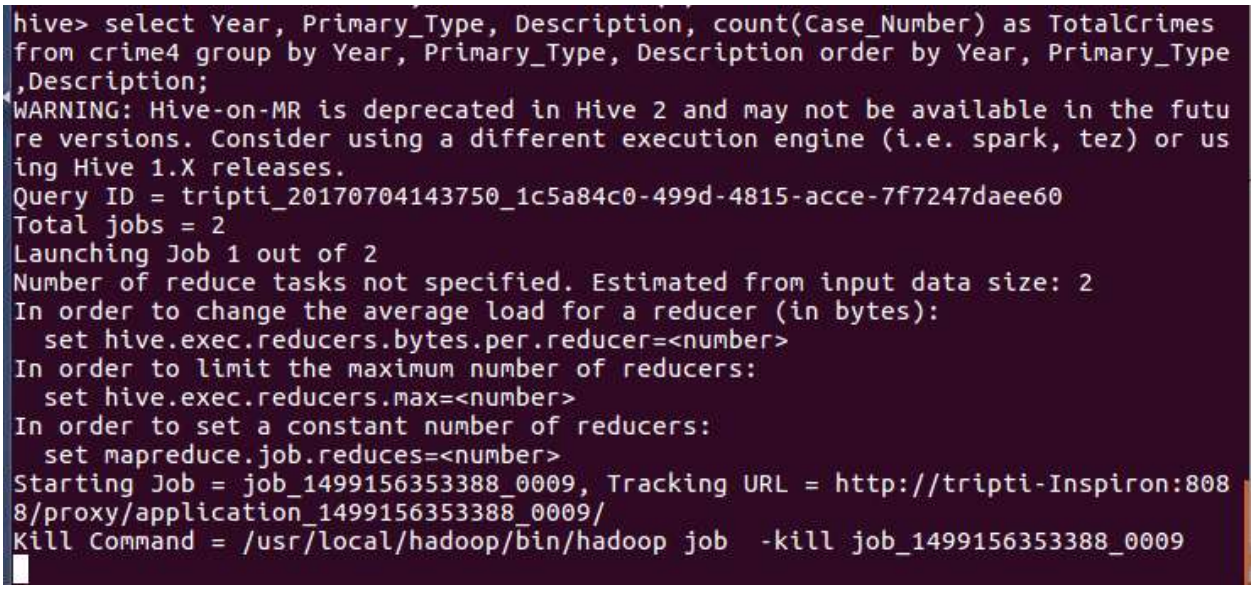


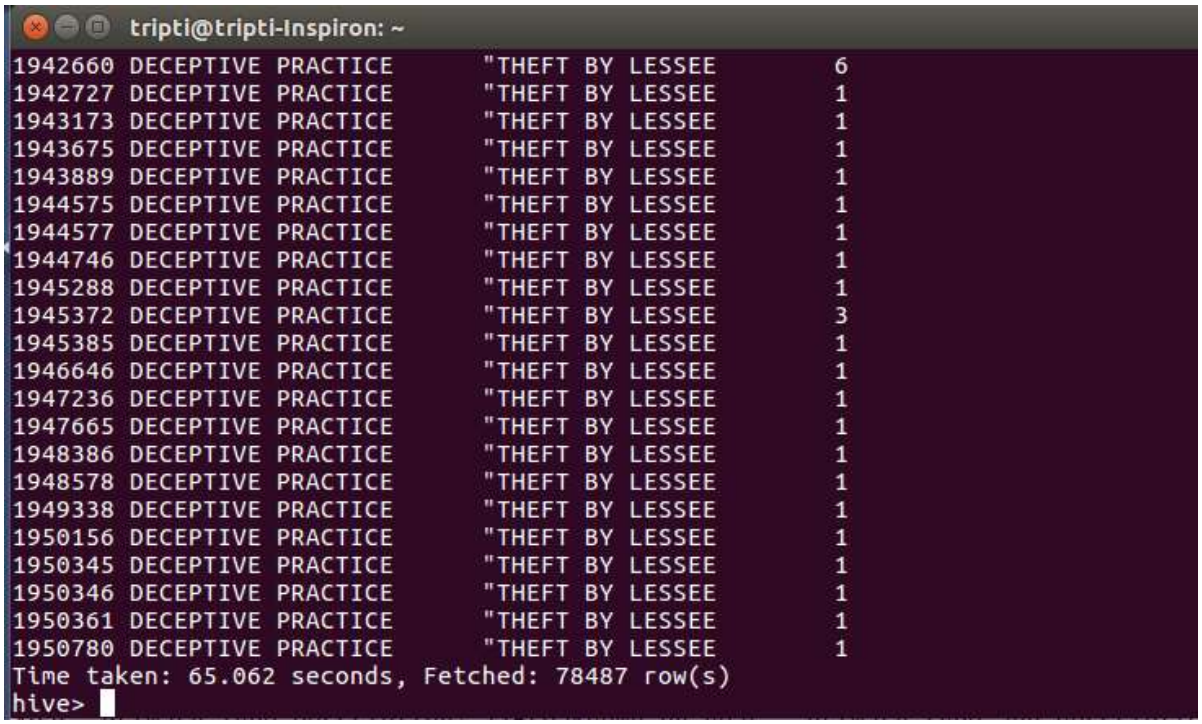
hive> select count(*) as count from crime4 where Primary_Type ='THEFT' and year = 2015;



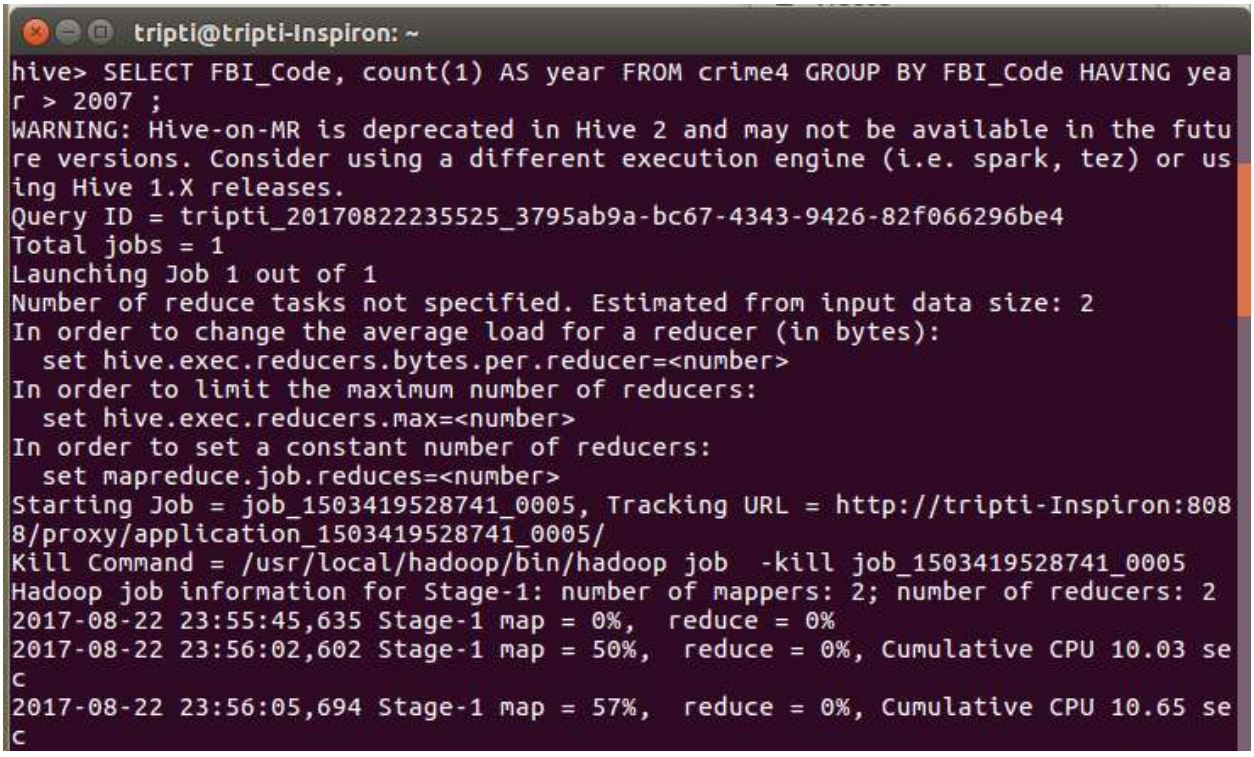


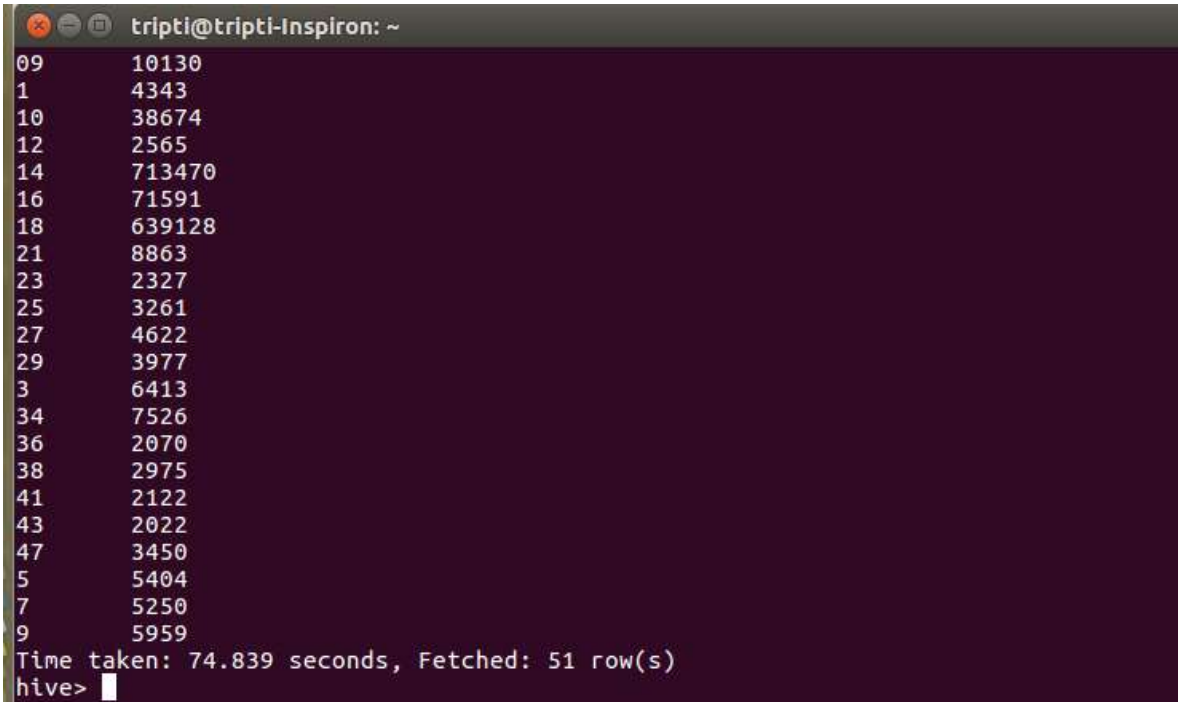
hive> select Year, Primary_Type, Description, count(Case_Number) as TotalCrimes from crime4 group by Year, Primary_Type, Description order by Year, Primary_Type,Description;





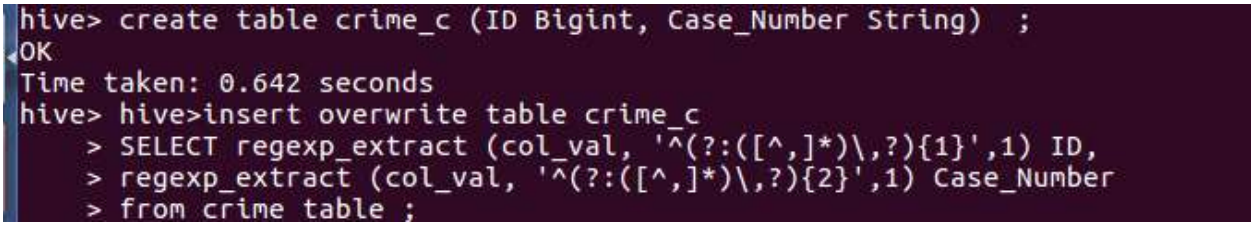
hive> SELECT FBI_Code, count(1) AS year FROM crime4 GROUP BY FBI_Code HAVING year > 2007 ;



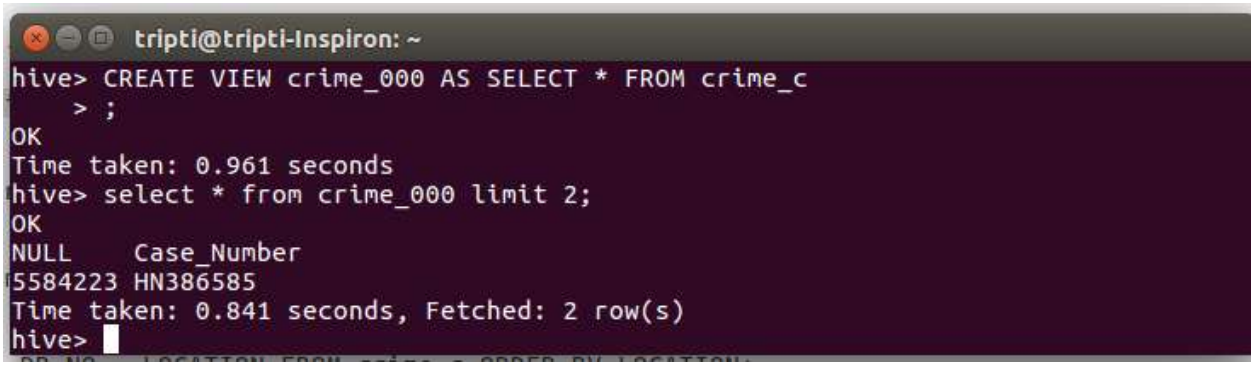


hive> create table crime_c (ID Bigint, Case_Number String) ;

hive>insert overwrite table crime_c SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){1}',1) ID, regexp_extract (col_value, '^(?:([^,]*)\,?){2}',1) Case_Number from crime_table ;



hive> CREATE VIEW crime_000 AS SELECT * FROM crime_c;



hive> SELECT ID, Case_Number FROM crime_c ORDER BY ID;

```
hive> Select ID, Case_Number from crime_c order by ID;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different ex
spark, tez) or using Hive 1.X releases.
Query ID = tripti_20170706132932_2b06ddca-f66f-4d88-a5f9-1813cea7c29f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499322985133_0003, Tracking URL = http://tripti-Inspiron:8088/proxy/application_1499322985133_0003/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499322985133_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-07-06 13:29:39,959 Stage-1 map = 0%,  reduce = 0%
2017-07-06 13:29:50,453 Stage-1 map = 24%,  reduce = 0%, Cumulative CPU 10.39 sec
2017-07-06 13:29:56,681 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 17.31 sec
2017-07-06 13:30:06,007 Stage-1 map = 61%,  reduce = 0%, Cumulative CPU 27.63 sec
2017-07-06 13:30:11,193 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 33.9 sec
2017-07-06 13:30:14,316 Stage-1 map = 91%,  reduce = 0%, Cumulative CPU 37.11 sec
2017-07-06 13:30:17,473 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 38.73 sec
```

```
10869840        JA177360
10869846        JA177517
10869859        JA177519
10869861        JA177281
10869870        JA177564
10869879        JA177037
10869894        JA177518
10869901        JA177351
10869931        JA177644
10869936        JA177625
10869945        JA177604
10869947        JA177520
10869971        JA177678
10870003        JA177687
10870029        JA177616
10870102        JA170694
10870104        JA177157
10870135        JA177820
10870180        JA177825
10870181        JA170655
10870187        JA176927
10870201        JA177832
10870203        JA177887
10870225        JA176809
10870270        JA177918
10870278        JA177967
10870292        JA177983
10870295        JA177971
10870313        JA169991
10870332        JA178019
10870352        JA178005
10870357        JA177960
Time taken: 69.619 seconds, Fetched: 6283303 row(s)
hive>
```

hive>create table crime_c2(Block String, IUCR String, Primary_Type String, Description String);

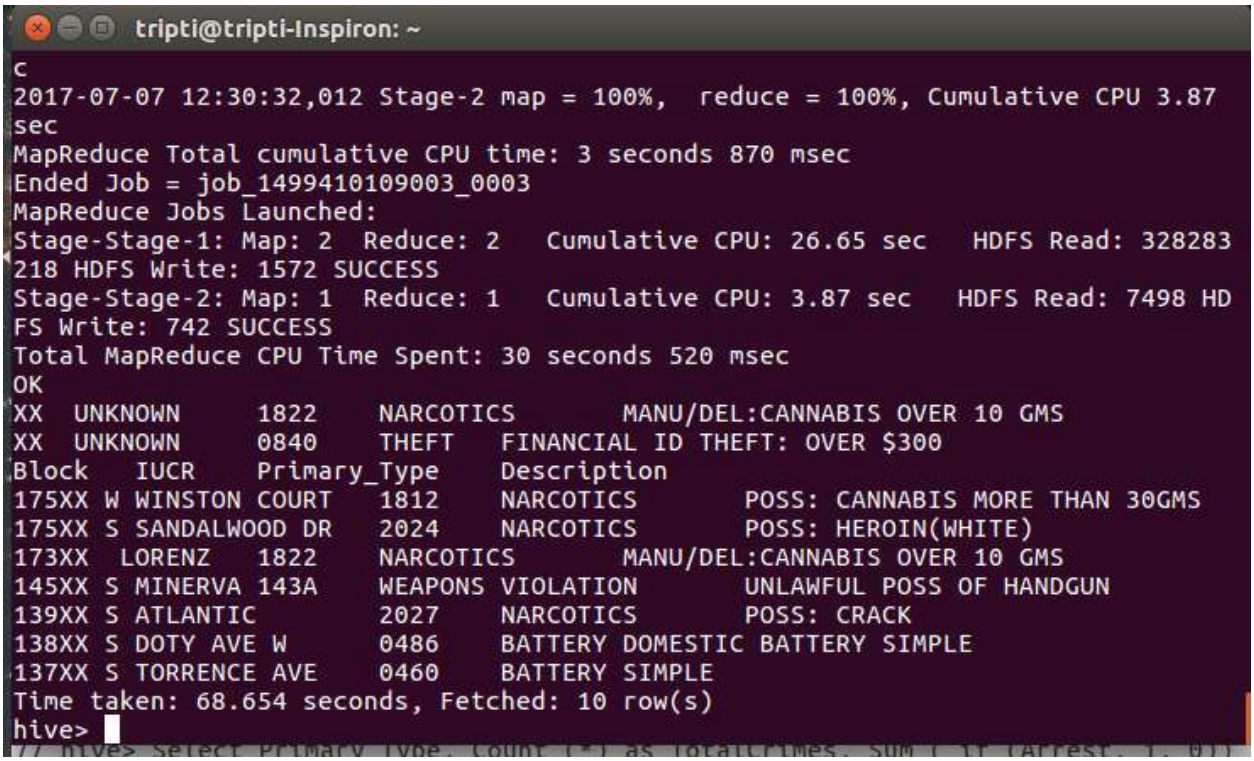
hive>insert overwrite table crime_C2 SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){4}',1) Block,regexp_extract (col_value, '^(?:([^,]*)\,?){5}',1)

IUCR,regexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type, regexp_extract (col_value, '^(?:([^,]*)\,?){7}',1) Description from crime_table ;

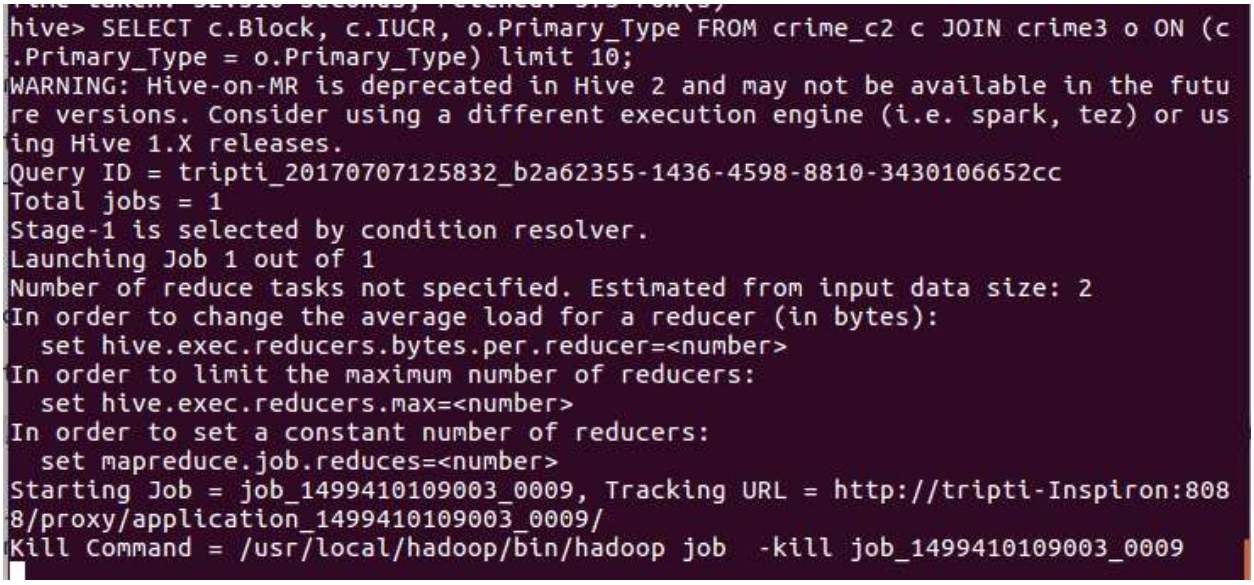
```
hive> create table crime_c2(Block String, IUCR String, Primary_Type String, Desc
ription String);
OK
Time taken: 0.441 seconds
hive> insert overwrite table crime_C2 SELECT regexp_extract (col_value, '^(?:([^
,]*)\,?){4}',1) Block,regexp_extract (col_value, '^(?:([^,]*)\,?){5}',1) IUCR,re
gexp_extract (col_value, '^(?:([^,]*)\,?){6}',1) Primary_Type, regexp_extract (c
ol_value, '^(?:([^,]*)\,?){7}',1) Description from crime_table ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170707095443_cfc21c90-37fc-4d30-a67c-23ae7603722c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499397806097_0002, Tracking URL = http://tripti-Inspiron:808
8/proxy/application_1499397806097_0002/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499397806097_0002
Hadoop job information for Stage-1: number of mappers: 6; number of reducers: 0
2017-07-07 09:54:58,628 Stage-1 map = 0%,  reduce = 0%
2017-07-07 09:57:35,517 Stage-1 map = 0%,  reduce = 0%
2017-07-07 09:59:07,836 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 15.79 sec
2017-07-07 09:59:26,489 Stage-1 map = 33%,  reduce = 0%, Cumulative CPU 155.38 s
```

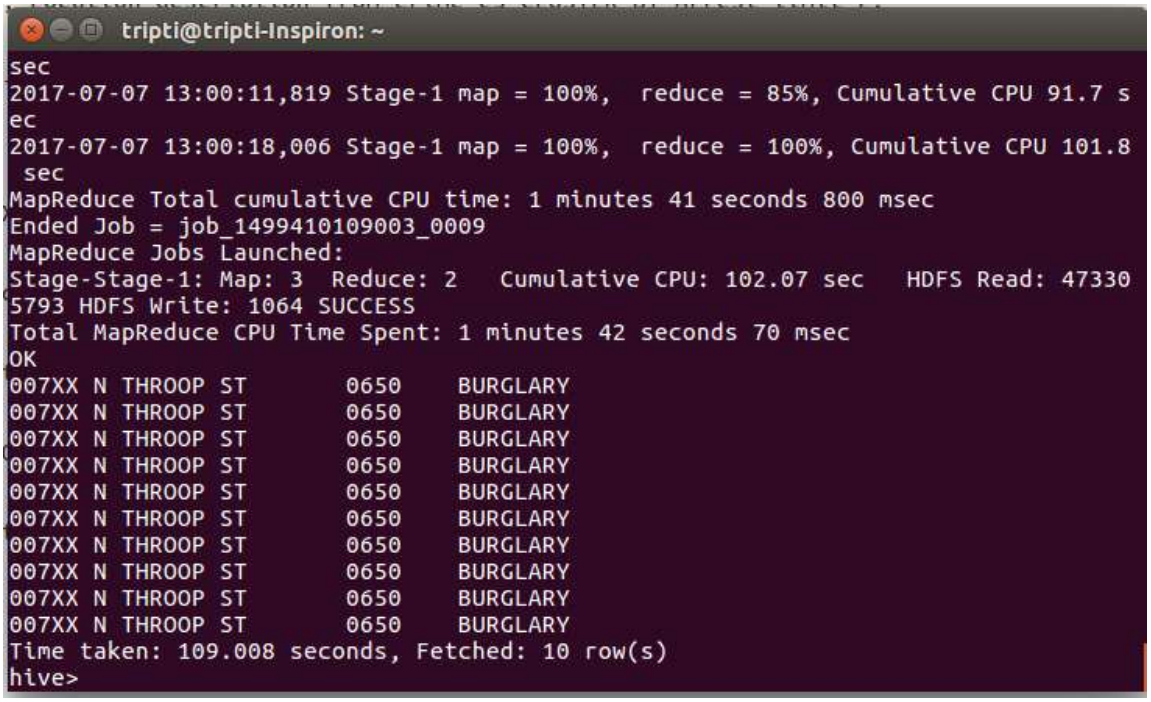
hive>SELECT * from crime_c2 SORT BY Block DESC limit 10;

```
hive> SELECT * from crime_c2 SORT BY Block DESC limit 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170707122924_086bec52-d8f5-4bc6-85f2-5f48f7e8c443
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 2
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1499410109003_0002, Tracking URL = http://tripti-Inspiron:808
8/proxy/application_1499410109003_0002/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499410109003_0002
```

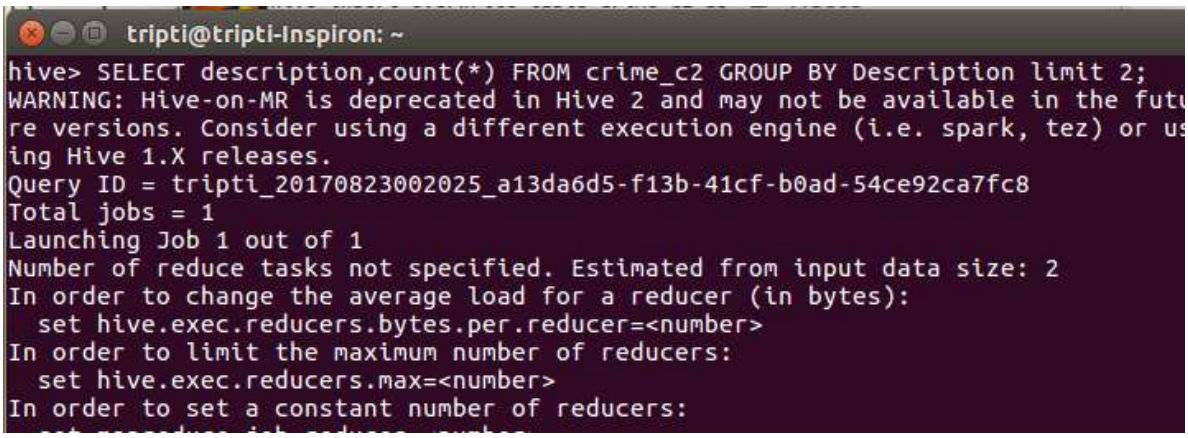


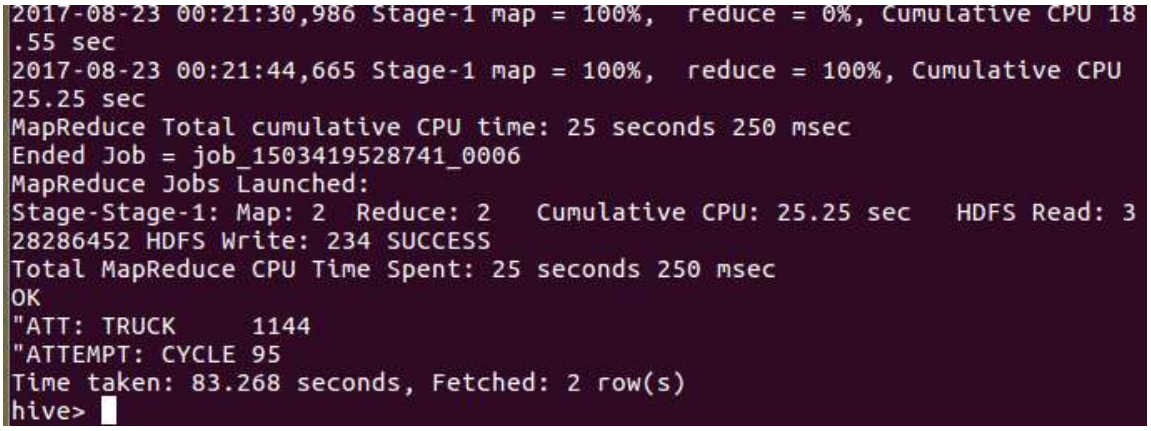
hive>SELECT c.Block, c.IUCR, o.Primary_Type FROM crime_c2 c JOIN crime3 o ON (c.Primary_Type = o.Primary_Type);





Hive> SELECT description,count(*) FROM crime_c2 GROUP BY Description limit 2;





hive>create table crime_c3(Location_description String, Arrest Boolean, Domestic Boolean, Beat Int);

hive> insert overwrite table crime_c3 SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){8}',1) Location_description,regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) Arrest,regexp_extract (col_value, '^(?:([^,]*)\,?){10}',1) Domestic, regexp_extract (col_value, '^(?:([^,]*)\,?){11}',1) Beat from crime_table ;

```
hive> create table crime_c3(Location_description String, Arrest Boolean, Domesti
c Boolean, Beat Int);
OK
Time taken: 0.266 seconds
hive> insert overwrite table crime_c3 SELECT regexp_extract (col_value, '^(?:([^
,]*)\,?){8}',1) Location_description,regexp_extract (col_value, '^(?:([^,]*)\,?)
{9}',1) Arrest,regexp_extract (col_value, '^(?:([^,]*)\,?){10}',1) Domestic, reg
exp_extract (col_value, '^(?:([^,]*)\,?){11}',1) Beat from crime_table ;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170707101128_ab7b518f-92b1-4616-b1fe-1550a7228330
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1499397806097_0004, Tracking URL = http://tripti-Inspiron:808
8/proxy/application_1499397806097_0004/
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1499397806097_0004
```

Hive> SELECT Arrest, Location_description from crime_c3 CLUSTER BY Arrest limit 2;

```
hive> SELECT  Arrest, Location_description from crime_c3 CLUSTER BY Arrest limit
 2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170707124150_d2f744d6-87ca-4ede-ab9c-81a8697082aa
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
```

```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.6 sec   HDFS Read: 1638421
09 HDFS Write: 163 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 3.4 sec   HDFS Read: 5437 HDF
S Write: 152 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 0 msec
OK
true    Location_Description
true    RESIDENCE
Time taken: 54.39 seconds, Fetched: 2 row(s)
hive>
```

hive>SELECT * from crime_c3 where Domestic=true limit 4;

```
hive> SELECT * from crime_c3 where Domestic=true limit 4;
OK
Location_Description    true    true    NULL
STREET  true    true    1332
STREET  true    true    432
STREET  true    true    733
Time taken: 0.545 seconds, Fetched: 4 row(s)
hive>
```

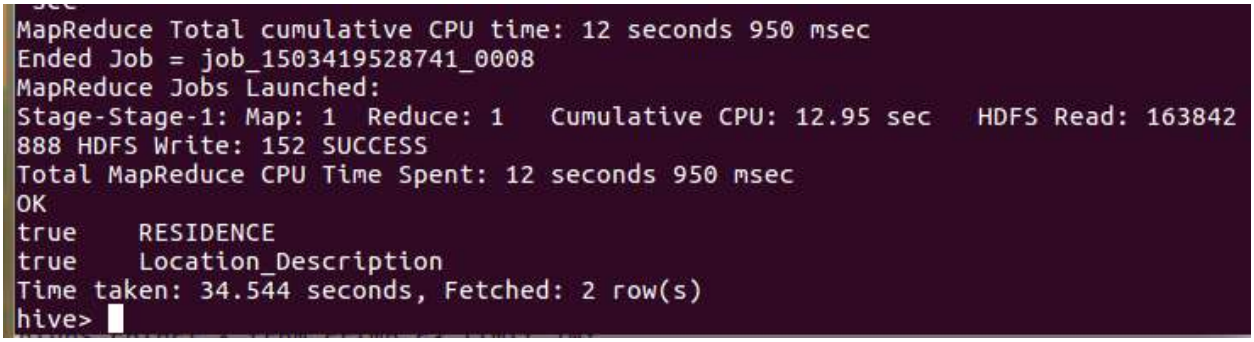
hive>SELECT Arrest , Location_Description, Domestic FROM crime_c3 ORDER BY Location_Description limit 10;

```
tripti@tripti-Inspiron: ~
hive> SELECT Arrest , Location_Description, Domestic FROM crime_c3 ORDER BY Loca
tion_Description limit 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170823004040_718c6fbe-0fc5-4b0f-a8d1-a41dca3242cf
Total jobs = 1
Launching Job 1 out of 1
```

```
MapReduce Total cumulative CPU time: 13 seconds 740 msec
Ended Job = job_1503419528741_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.74 sec   HDFS Read: 163843
272 HDFS Write: 317 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 740 msec
OK
true            true
true            true
true            true
true            true
true            true
true            true
true            true
true            true
true            true
Time taken: 44.965 seconds, Fetched: 10 row(s)
hive>
```

hive>SELECT Arrest, Location_description from crime_c3 DISTRIBUTE BY Arrest limit 2;

```
hive> SELECT  Arrest, Location_description from crime_c3 DISTRIBUTE BY Arrest li
mit 2;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the futu
re versions. Consider using a different execution engine (i.e. spark, tez) or us
ing Hive 1.X releases.
Query ID = tripti_20170823004316_fc07163b-7b99-4346-a03d-f1b7331d8819
Total jobs = 1
Launching Job 1 out of 1
```


MapReduce Total cumulative CPU time: 12 seconds 950 msec
Ended Job = job_1503419528741_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 12.95 sec HDFS Read: 163842
888 HDFS Write: 152 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 950 msec
OK
true RESIDENCE
true Location_Description
Time taken: 34.544 seconds, Fetched: 2 row(s)
hive>