**What is Hive?**

Apache Hive is Data warehouse system built on the top of the Hadoop. Apache Hive is used to querying and managing large datasets residing in distributed storage. Before becoming a open source project of Apache Hadoop, Hive was originated in Facebook. Hive provides a mechanism to project structure onto the data in Hadoop and to query that data using a SQL-like language called HiveQL (HQL).

**What is HQL?**

Hive defines a simple SQL-like query language to to querying and managing large datasets is called Hive-QL ( HQL ), it is easy to use who are familiar with SQL Language. Hive allows the programmers who are familiar with the programming language to write the custom Mapreduce framework to perform more sophisticated analysis.

**Why Hive?**

The tables in Hive are similar to tables in a relational database, It is easy to use who are familiar with SQL. Many users can simultaneously query the data using Hive-QL.

**Uses of Hive?**

1. The Apache Hive distributed storage.

2. Hive provide the tools to enable easy data extract/transform/load (ETL)

3. It provide the structure on a variety of data formats.

4. By using Hive we can access to files stored in Hadoop Distributed File System (HDFS is used to querying and managing large datasets residing in) or in other data storage systems such as Apache HBase.

**Limitations of Hive?**

• Hive is not designed for the Online transaction processing (OLTP ), it is only used for the Online analytical processing.

• Hive supports overwriting (i.e. overwriting of hdfs file) or appending data, but not updates and deletes.

• In a Hive **Sub queries are** not supported.

**Why Hive when Pig is there?**

* Hive-QL is a declarative language line SQL, PigLatin is a data flow language.
* Pig: a dataflow language and environment for exploring very large datasets.
* Hive: a distributed data warehouse.

**Hive components:**

**Metastore :**

Hive stores the schema of the Hive tables in a Hive **Metastore**.

Metastore is used to hold all the information about the **tables and partitions** that are in the warehouse.

By default, the metastore is run in the same process as the Hive service and the default metastore is DerBy Database.

**SerDe :**

**Serializer, Deserialize**r it gives the instructions to hive on how to process a record.

**Hive Commands :**

**Hive-QL statements :**

**Data definition language (DDL )**

DDL statements are used to build and modify the tables and other objects in the database.

Example : CREATE, DROP, TRUNCATE, ALTER, SHOW, DESCRIBE Statements.

**Data Manipulation Language (DML )**

DML statements are used to retrieve, store, modify, delete, insert and update data in database.

Example : LOAD, INSERT Statements.

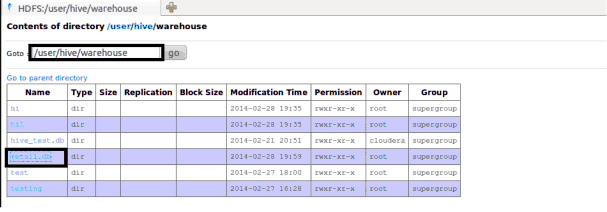
Go to Hive shell by giving the command sudo hive and enter the command **CREATE DATABASE** <data base name> to create the new database in the Hive.



To List out the databases in Hive warehouse enter the command **SHOW DATABASES.**



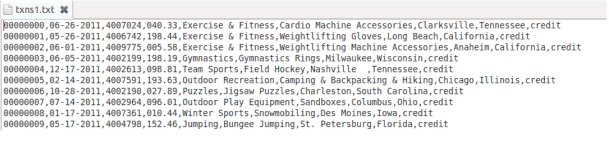
The database creates in a default location of the hive warehouse. In cloudera Hive database store in a /user/hive/warehouse.



Command to use the database in**USE <data base name>**



Copy the input data to HDFS from local by using the copyFromLocal command.

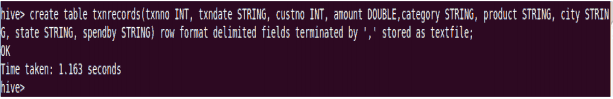


25

When we create table in hive it will creates in the default location of the hive warehouse. – “/user/hive/warehouse”, After creation of the table we can move the data from HDFS to hive table.

The following command creates a table with in location of**“/user/hive/warehouse/retail.db”**

Note : retail.db is the database created in the Hive warehouse.



**DESCRIBE** provides information about the schema of the table.

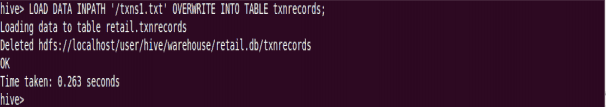


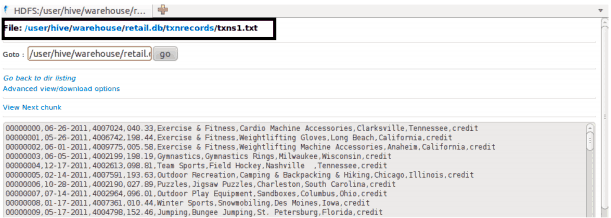
**LOAD** command :

Load operation is used to move the data into corresponding Hive table.

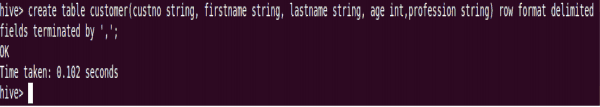
Syntax : **LOAD data <LOCAL> inpath <file path> into table [tablename]**

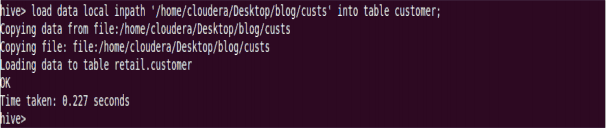
If the keyword LOCAL is specified, then in the load command will give the local file system path. If the keyword LOCAL is not specified we have to use the HDFS path of the file.





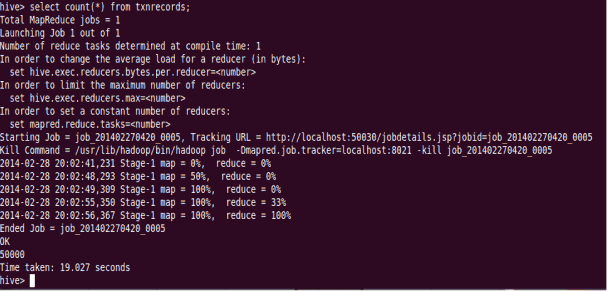
Example for the **LOAD** data **LOCAL** command





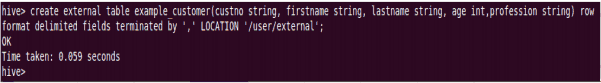
After loading the data into the Hive table we can apply the Data Manipulation Statements or aggregate functions retrieve the data.

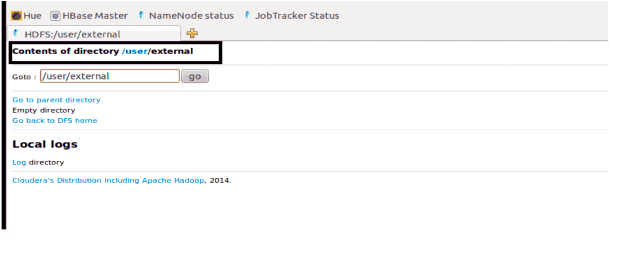
Example : count aggregate function is used count the total number of the records in a table.



**CREATE EXTERNAL table :**

The **CREATE EXTERNAL** keyword is used to create a table and provide a**LOCATION**where table will create so that Hive does not use a default location for this table. An EXTERNAL table points to any HDFS location for its storage, rather than default storage.



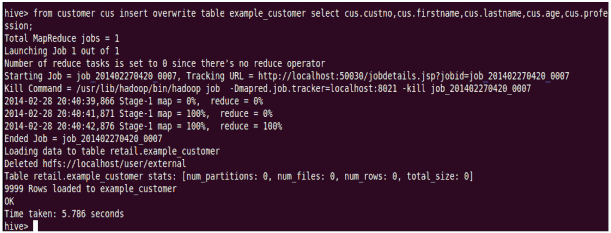


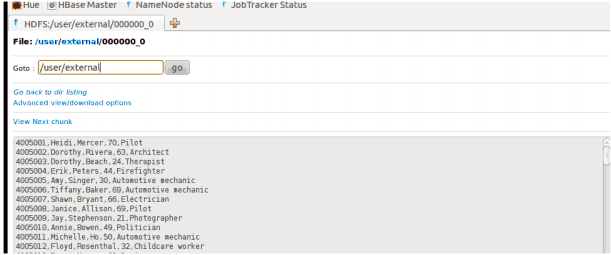
**INSERT** command is used to load the data Hive table.

• Inserts can be done to a table or a partition.

• INSERT OVERWRITE is used to overwrite the existing data in the table or partition.

• INSERT INTO is used to append the data into existing data in a table. (Note: INSERT INTO syntax is work from the version 0.8)

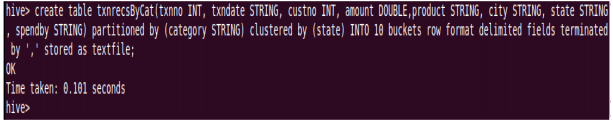


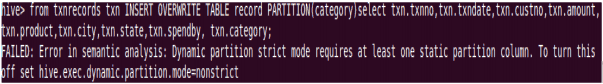


Example for the **PARTITIONED BY** and **CLUSTERED BY** command :

PARTITIONED BY is used to divided the table into the PARTITION and we can divide the

PARTITION in to Buckets by using the CLUSTERED BY command.





When we insert the data hive throwing errors, like dynamic partition mode is strict and dynamic partition not enabled. So we need to set the following parameters in hive shell

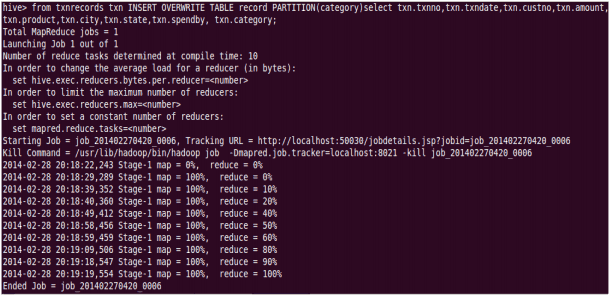
set hive.exec.dynamic.partition=true;

To enable dynamic partitions, by default it is false

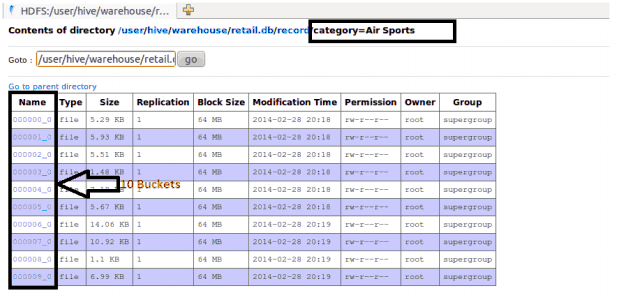
set hive.exec.dynamic.partition.mode=nonstrict;

39

40

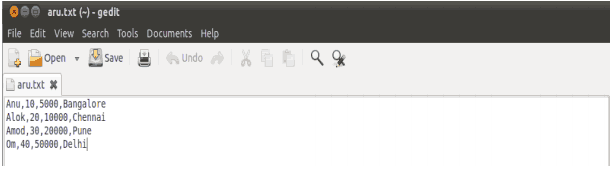


PARTITION is done by the category and each PARTITION we have created 10 buckets by applying the CLUSTER for the state.



The DROP TABLE statement deletes the data and metadata for a table. In the case of external tables, only the metadata is deleted.



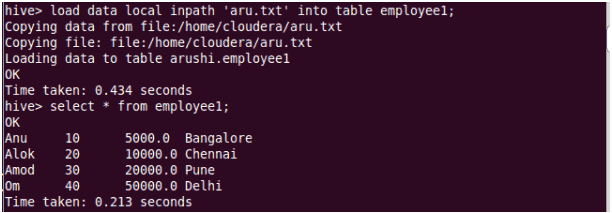


The DROP TABLE statement deletes the data and metadata for a table. In the case of external tables, only the metadata is deleted.

load data local inpath ‘aru.txt’ into table tablename

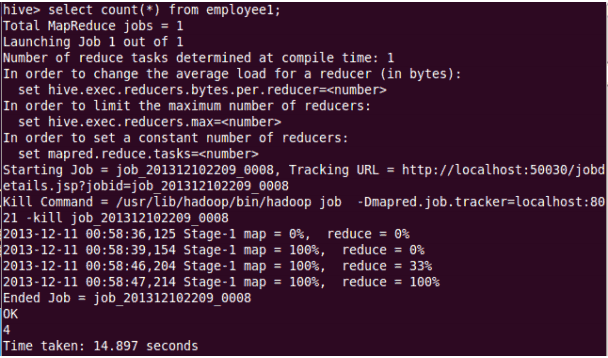
Then we check employee1 table by using

**Select \* from table** name command



To count the number of records in table by using

Select **count(\*)** from txnrecords;



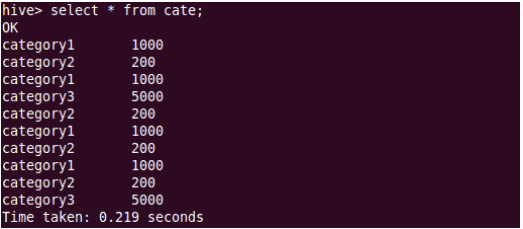
**Aggregation :**

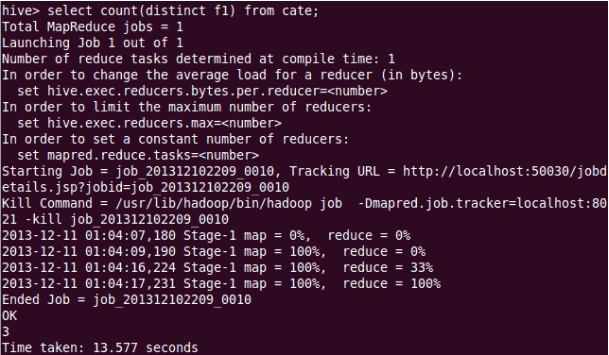
Select count (DISTINCT category) from tablename;

This command will count the different category of cate table

Here there are 3 different category.

Suppose there is another table cate where f1 is field name of category.



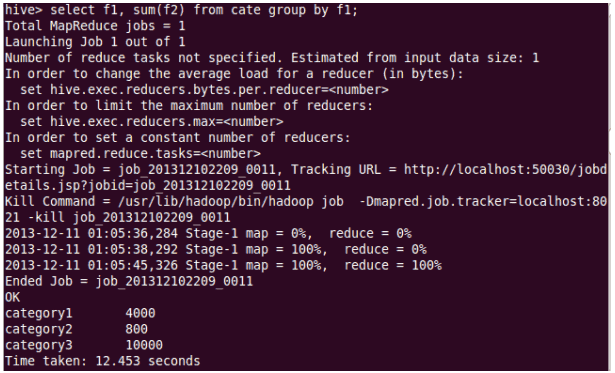


Grouping :

**GROUP** command is used to group the result-set by one or more columns.

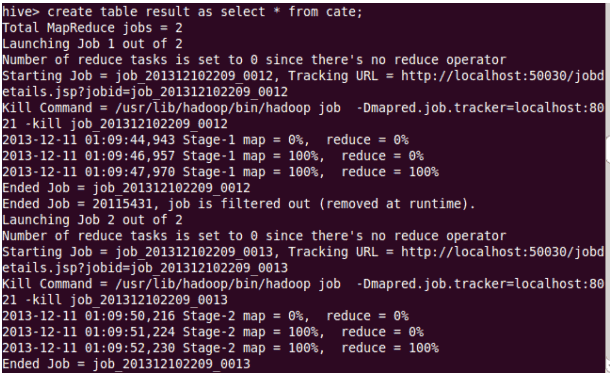
Select category, sum( amount) from txtrecords group by category

It calculate the amount of same category.



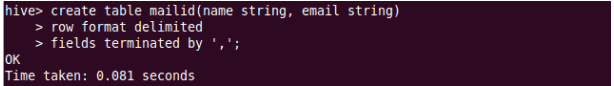
We store result of one table into another table.

**Create table newtablename** **as select \* from oldtablename;**



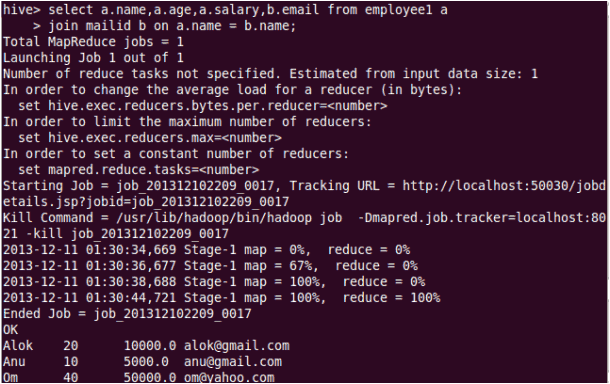
**JOIN command :**

We are creating one more table name with mailid



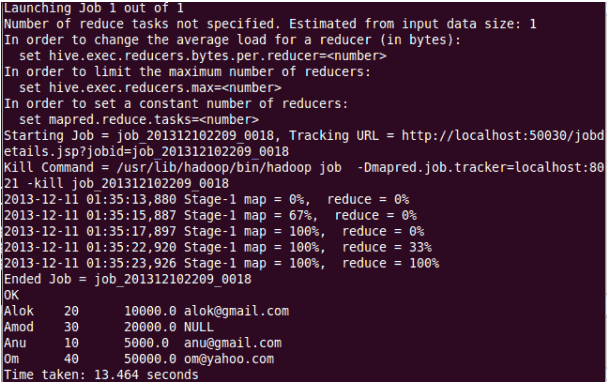


**Join operation**:- A **JOIN** is perform to combining fields from two tables by using values common to each.



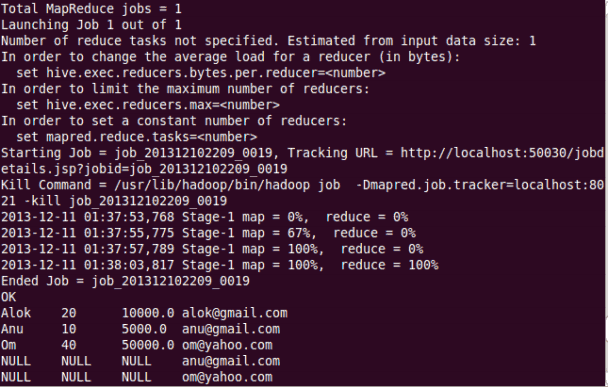
Left outer join:- The result of a left outer join (or simply left join) for tables A and B always contains all records of the “left” table (A), even if the join-condition does not find any matching record in the “right” table (B).

54



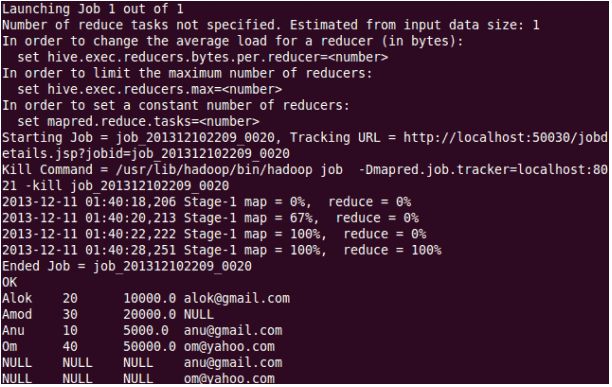
**Right outer join**:- A right outer join (or right join) closely resembles a left outer join, except with the treatment of the tables reversed. Every row from the “right” table (B) will appear in the joined table at least once.

56



**Full join**:- The joined table will contain all records from both tables, and fill in NULLs for missing matches on either side.

58



Once done with hive we can use quit command to exit from the hive shell.

60