**What is Hive?**

Apache Hive is a Data warehouse system which is built to work on Hadoop. It 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. It provides a mechanism to project structure onto the data in Hadoop and to query that data using a SQL-like language called HiveQL (HQL).

Hive is used because the tables in Hive are similar to tables in a relational database. If you are familiar with SQL, it’s a cakewalk. Many users can simultaneously query the data using Hive-QL.

[Wish-to-Learn-Hive](http://www.edureka.co/big-data-and-hadoop?utm_source=blog&utm_medium=button&utm_campaign=hive-commands-wish-to-learn-hive)

**What is HQL?**

Hive defines a simple SQL-like query language to querying and managing large datasets called Hive-QL ( HQL ). It’s easy to use if you’re familiar with SQL Language. Hive allows programmers who are familiar with the language to write the custom MapReduce framework to perform more sophisticated analysis.

**Uses of Hive:**

1. The Apache Hive distributed storage.

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

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

4. By using Hive, we can access 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 Online transaction processing (OLTP ), it is only used for the Online Analytical Processing.

• Hive supports overwriting or apprehending data, but not updates and deletes.

• In Hive, sub queries are not supported.

**Why Hive is used inspite of Pig?**

The following are the reasons why Hive is used in spite of Pig’s availability:

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

**Components of Hive:**

**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, Deserializer gives instructions to hive on how to process a record.

**Hive Commands :**

**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.

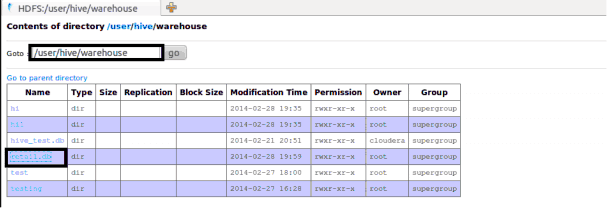
Go to Hive shell by giving the command sudo hive and enter the command**‘create database** **<data base name>’** to create the new database in the Hive.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/201.png)

To list out the databases in Hive warehouse, enter the command ‘**show databases’.**

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/21-1.png)

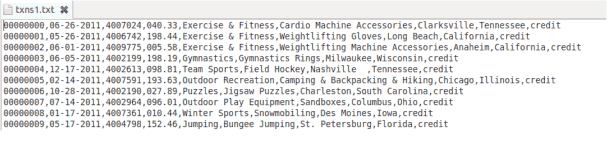
The database creates in a default location of the Hive warehouse. In Cloudera, Hive database store in a /user/hive/warehouse.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/22-1.png)

The command to use the database is**USE <data base name>**

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/23-1.png)

Copy the input data to HDFS from local by using the copy From Local command.

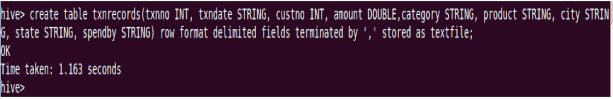
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/24-1.png)

Hive table

When we create a table in hive, it 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.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/26-1.png)

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

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/27-1.png)

**Data Manipulation Language (DML )**

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

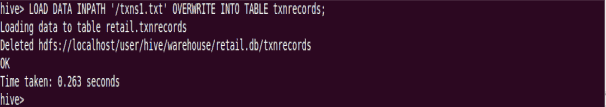
*Example :*

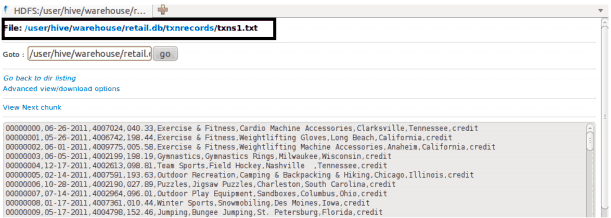
LOAD, INSERT Statements.

Syntax :

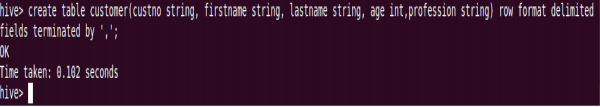
LOAD data <LOCAL> inpath <file path> into table [tablename]

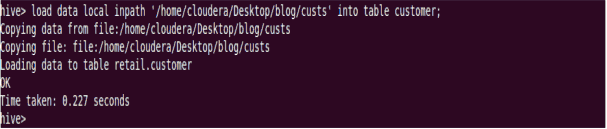
The Load operation is used to move the data into corresponding Hive table. If the keyword **local** is specified, then in the load command will give the local file system path. If the keyword local is not specified we have to use the HDFS path of the file.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/28-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/29-1.png)

*Here are some examples for the LOAD data LOCAL command*

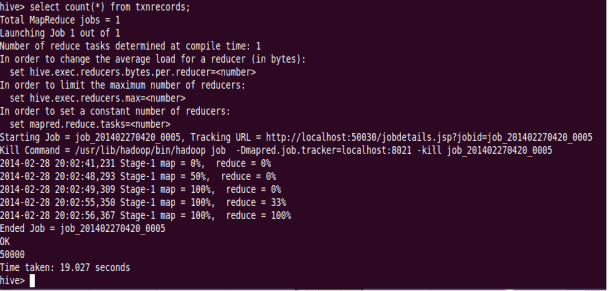
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/30-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/311-1.png)

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

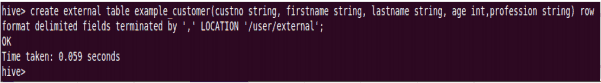
*Example to count number of records:*

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

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/32-1.png)

**‘create external’ Table :**

The **create external** keyword is used to create a table and provides a location where the 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.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/33-1.png)

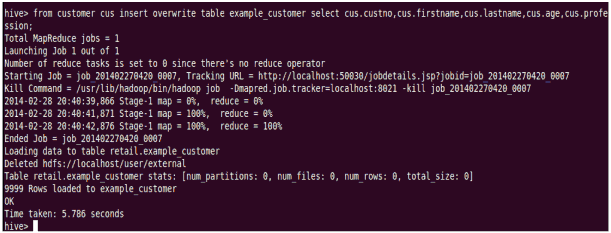
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/34-1.png)

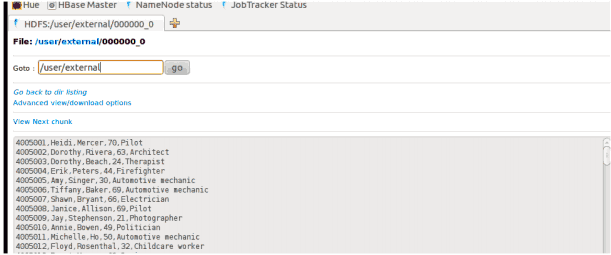
**Insert Command:**

The **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.

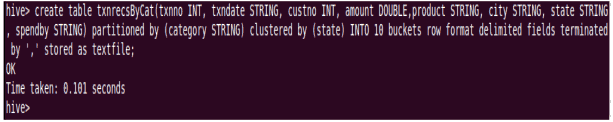
• INSERT INTO is used to append the data into existing data in a table. (Note: INSERT INTO syntax is work from the version 0.8)

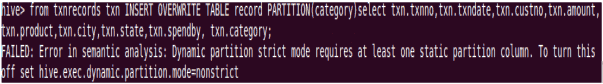
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/35-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/36-1.png)

**Example for ‘Partitioned By’ and ‘Clustered By’ Command :**

**‘Partitioned by**‘ is used to divided the table into the Partition and can be divided in to buckets by using the ‘**Clustered By**‘ command.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/37-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/38-1.png)

When we insert the data Hive throwing errors, the dynamic partition mode is strict and dynamic partition not enabled (by [Jeff](http://www.dresshead.com/dresshead-staff-profile-jeff-maurer/) at [dresshead website](http://www.dresshead.com/)). So we need to set the following parameters in Hive shell.

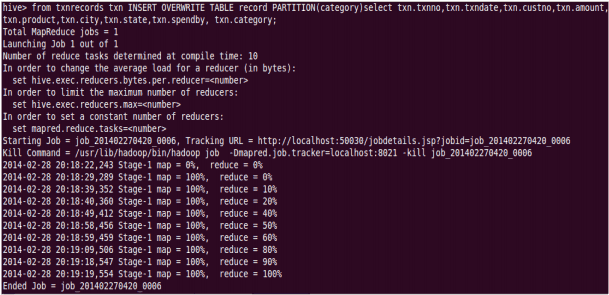
set hive.exec.dynamic.partition=true;

To enable dynamic partitions, by default, it’s false

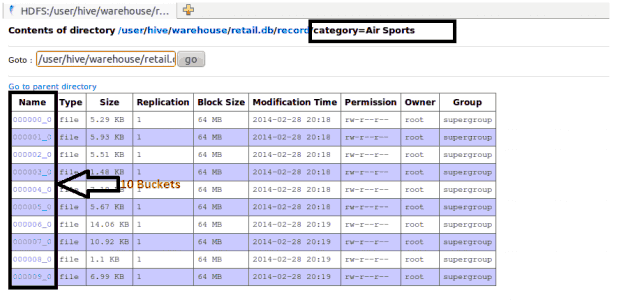
set hive.exec.dynamic.partition.mode=nonstrict;

[Dynamic Partitions](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/39-1.png)

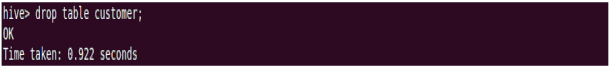
[Dynamic Partitions](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/40-1.png)

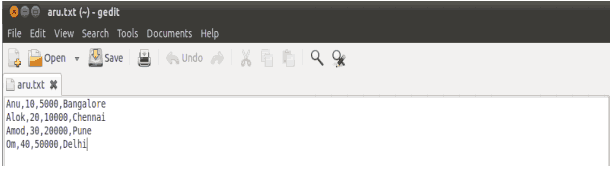
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/41-1.png)

Partition is done by the category and can be divided in to buckets by using the ‘Clustered By’ command.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/42-1.png)

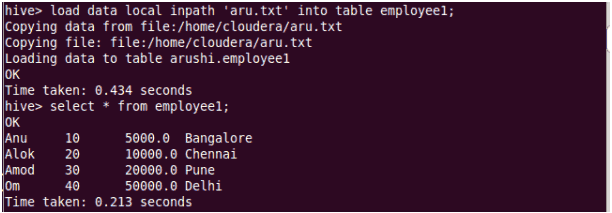
The ‘Drop Table’ statement deletes the data and metadata for a table. In the case of external tables, only the metadata is deleted.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/43-1.png)

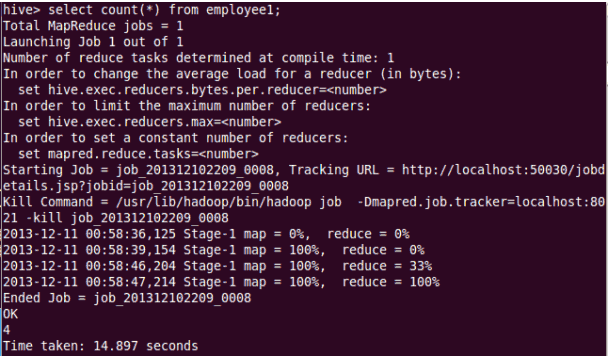
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/44-1.png)

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 and then we check employee1 table by using Select \* from table name command

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/45-1.png)

To count the number of records in table by using Select **count(\*)** from txnrecords;

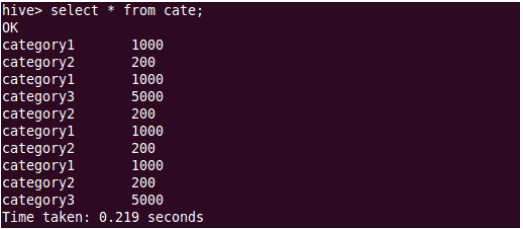
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/46-1.png)

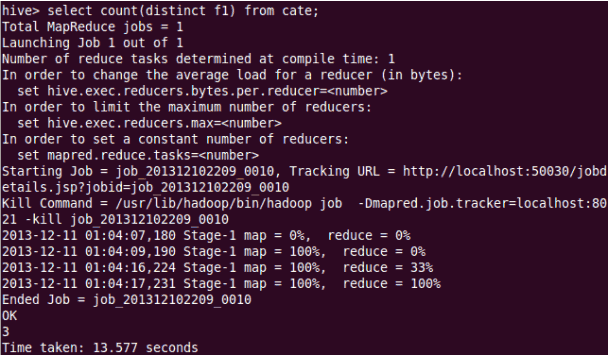
**Aggregation :**

Select count (DISTINCT category) from tablename;

This command will count the different category of ‘cate’ table. Here there are 3 different categories.

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

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/47-1.png)

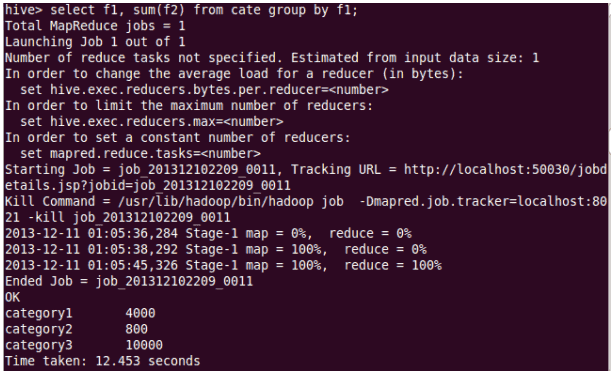
[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/48-1.png)

**Grouping :**

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

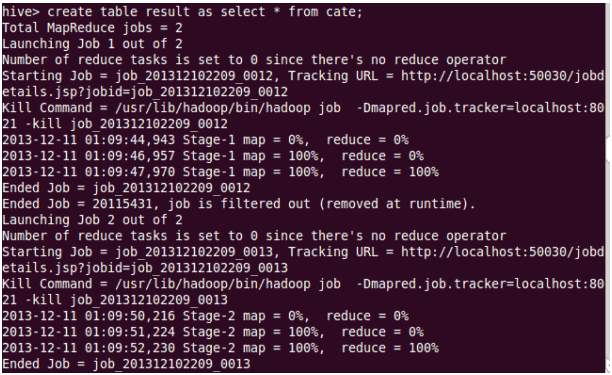
Select category, sum( amount) from txt records group by category

It calculates the amount of same category.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/49-11.png)

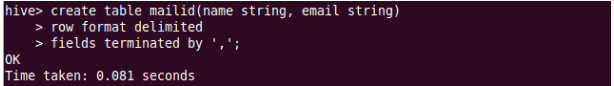
The result one table is stored in to another table.

Create table newtablename as select \* from oldtablename;

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/50-1.png)

**Join Command :**

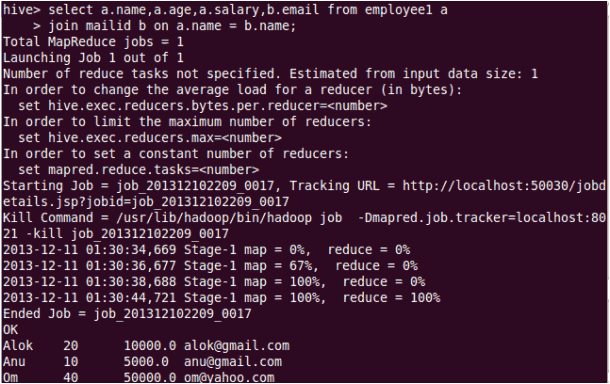
Here one more table is created in the name**‘mailid’**

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/51-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/52-1.png)

**Join Operation**:

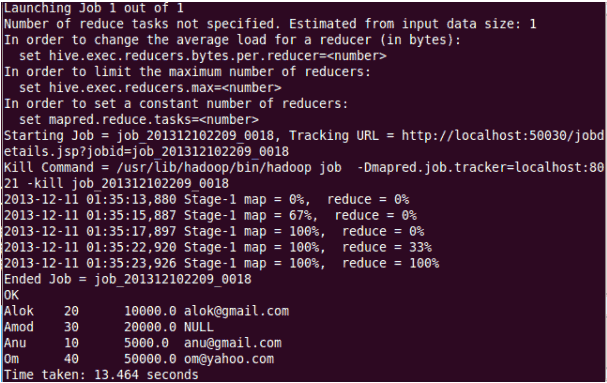
A Join operation is performed to combining fields from two tables by using values common to each.

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/53-1.png)

**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).

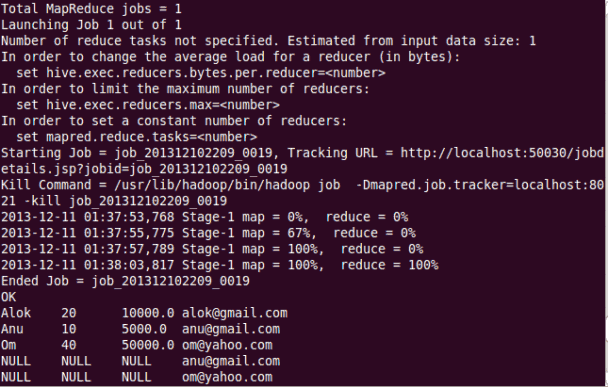
[Left Outer Join](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/54-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/55-1.png)

**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.

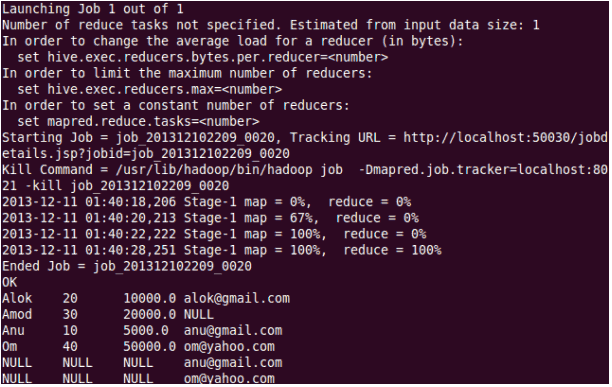
[Right Outer Join](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/56-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/57-1.png)

**Full Join**:

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

[Full Join](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/58-1.png)

[](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/59-1.png)

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

[Exiting from Hive](http://cdn.edureka.co/blog/wp-content/uploads/2014/03/60-1.png)

Hive is just a part of the big puzzle called Big Data and Hadoop. Hadoop is much more than just Hive. Click below to see what other skills you should master in Hadoop.