ACD\_BDD2.3\_Session\_7\_Assignment\_3

Explain the below concepts with an example in brief.

Hive Data Definitions

* CREATE DATABASE/SCHEMA, TABLE, VIEW, FUNCTION, INDEX

CREATE TABLE page\_view(viewTime INT, userid BIGINT,

      page\_url STRING, referrer\_url STRING,

      ip STRING COMMENT 'IP Address of the User')

 COMMENT 'This is the page view table'

 PARTITIONED BY(dt STRING, country STRING)

 ROW FORMAT DELIMITED

   FIELDS TERMINATED BY '\001'

STORED AS SEQUENCEFILE;

The above command creates a HIVE MANAGED table page\_view

* DROP DATABASE/SCHEMA, TABLE, VIEW, INDEX

DROP TABLE IF EXISTS page\_view

DROP TABLE removes metadata and data for this table. The data is actually moved to the .Trash/Current directory if Trash is configured  The metadata is completely lost.

When dropping an EXTERNAL table, data in the table will NOT be deleted from the file system.

* TRUNCATE TABLE

TRUNCATE TABLE page\_view

Removes all rows from a table or partition(s). The rows will be trashed if the filesystem Trash is enabled, otherwise they are deleted.Currently the target table should be native/managed table or an exception will be thrown. User can specify partial partition\_spec for truncating multiple partitions at once and omitting partition\_spec will truncate all partitions in the table.

* ALTER DATABASE/SCHEMA, TABLE, VIEW

ALTER TABLE table\_name RENAME TO new\_table\_name;

This statement lets you change the name of a table to a different name.  You can add columns/partitions, change SerDe, add table and SerDe properties, or rename the table itself. Similarly, alter table partition statements allow you change the properties of a specific partition in the named table.

The column change command will only modify Hive's metadata, and will not modify data. Users should make sure the actual data layout of the table/partition conforms with the metadata definition.

* MSCK REPAIR TABLE (or ALTER TABLE RECOVER PARTITIONS)
* SHOW DATABASES/SCHEMAS, TABLES, TBLPROPERTIES, VIEWS, PARTITIONS, FUNCTIONS, INDEX[ES], COLUMNS, CREATE TABLE

SHOW DATABASES or SHOW SCHEMAS lists all of the databases defined in the metastore. The uses of SCHEMAS and DATABASES are interchangeable – they mean the same thing.

SHOW TABLES lists all the base tables and views in the current database (or the one explicitly named using the IN clause) with names matching the optional regular expression. Wildcards in the regular expression can only be '\*' for any character(s) or '|' for a choice. Examples are 'page\_view', 'page\_v\*', '\*view|page\*', all which will match the 'page\_view' table. Matching tables are listed in alphabetical order. It is not an error if there are no matching tables found in metastore. If no regular expression is given then all tables in the selected database are listed.

SHOW VIEWS lists all the views in the current database.

SHOW PARTITIONS lists all the existing partitions for a given base table. Partitions are listed in alphabetical order.

SHOW TABLE EXTENDED will list information for all tables matching the given regular expression

* DESCRIBE DATABASE/SCHEMA, table\_name, view\_name

PARTITION statements are usually options of TABLE statements, except for SHOW PARTITIONS.

**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.

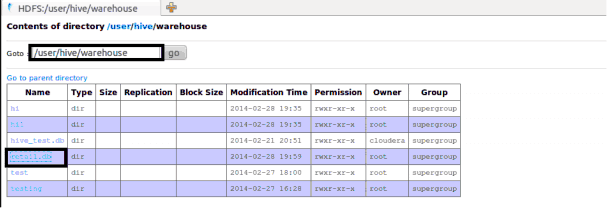
Go to Hive shell by giving the command sudo hive and enter the command**‘create database<data base** **name>’** to create the new database in the Hive.

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

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

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

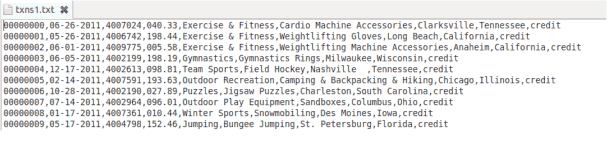
The database creates in a default location of the Hive warehouse. In Cloudera, Hive database store in a /user/hive/warehouse.

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

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

[](https://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.

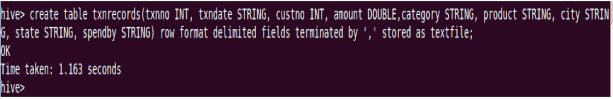
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/24-1.png)

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.

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

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

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

Hive Data Manipulations

**Data Manipulation Language (DML )**

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

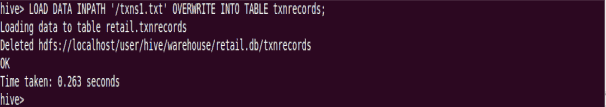
Example :

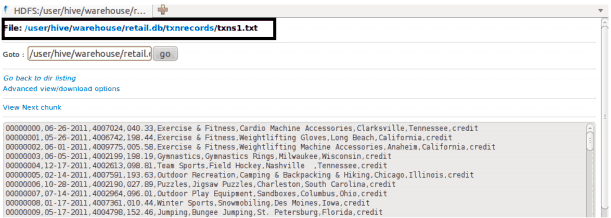
LOAD, INSERT Statements.

Syntax :

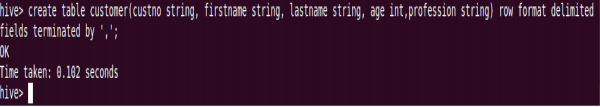
LOAD data <LOCAL> inpath <file path> into table [tablename]

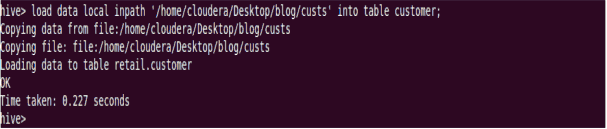
The Load operation is used to move the data into corresponding Hive table. If the keyword **local** is specified, then in the load command will give the local file system path. If the keyword local is not specified we have to use the HDFS path of the file.

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

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

Here are some examples for the LOAD data LOCAL command

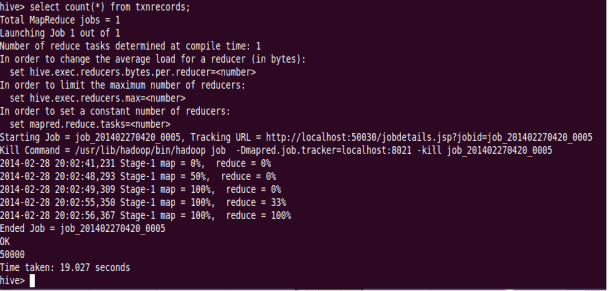
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/30-1.png)

[](https://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.

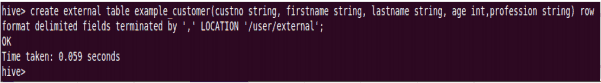
Example to count number of records:

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

[](https://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.

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

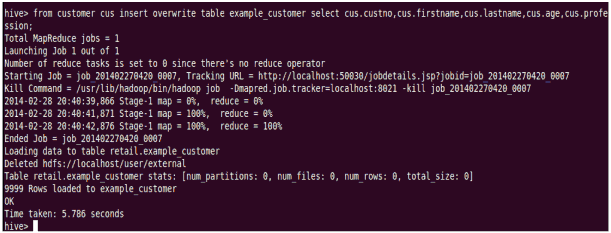
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/34-1.png)

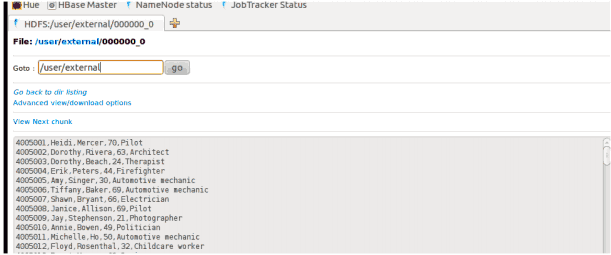
**Insert Command:**

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.

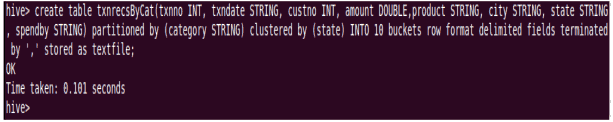
• INSERT INTO is used to append the data into existing data in a table. (Note: INSERT INTO syntax is work from the version 0.8)

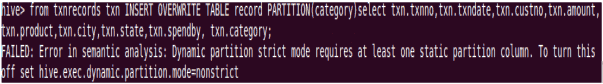
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/35-1.png)

[](https://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.

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

[](https://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.

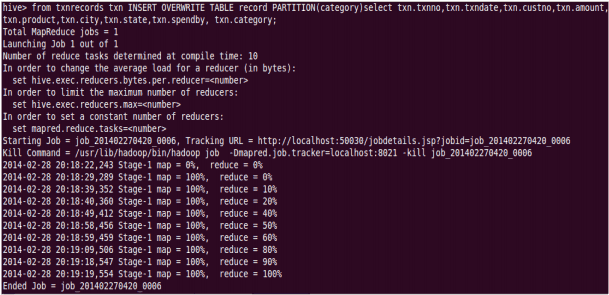
set hive.exec.dynamic.partition=true;

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

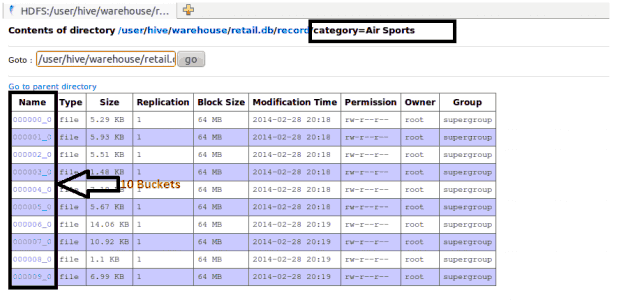
set hive.exec.dynamic.partition.mode=nonstrict;

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

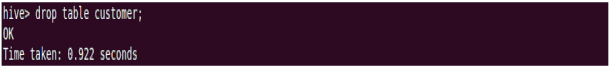
[Dynamic Partitions](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/40-1.png)

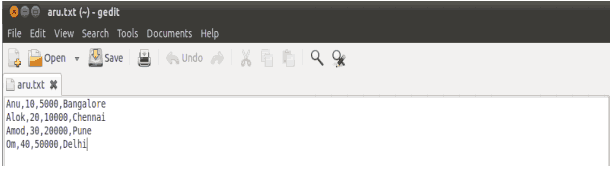
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/41-1.png)

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

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

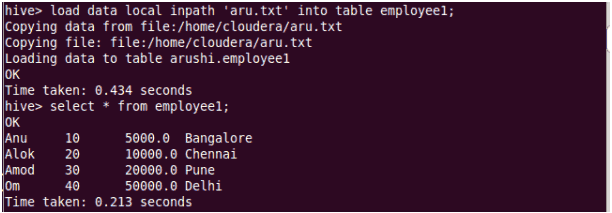
The ‘Drop Table’ statement deletes the data and metadata for a table. In the case of external tables, only the metadata is deleted.

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

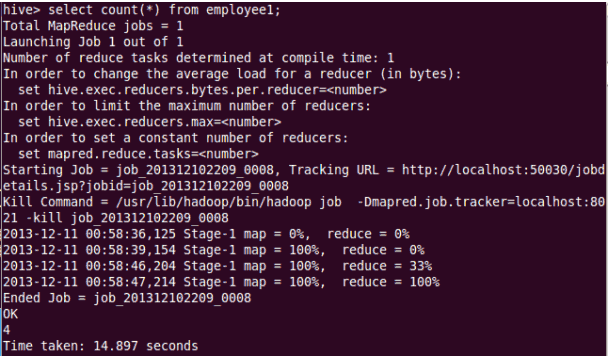
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/44-1.png)

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

[](https://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;

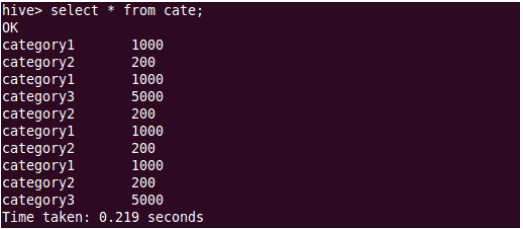
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/46-1.png)

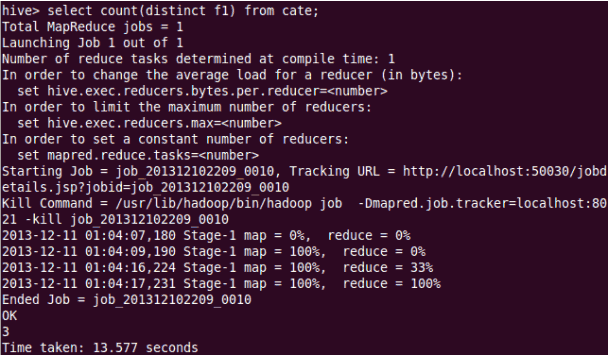
## ****Aggregation :****

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.

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

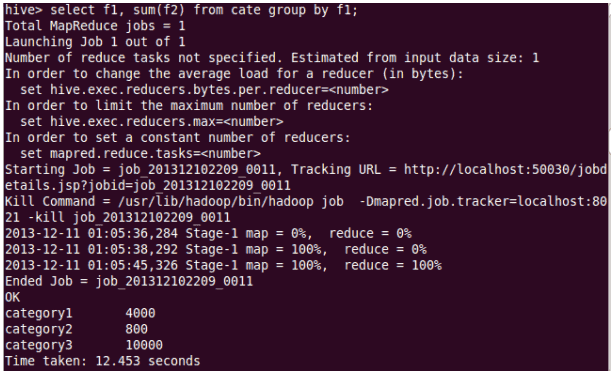
[](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/48-1.png)

## ****Grouping :****

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

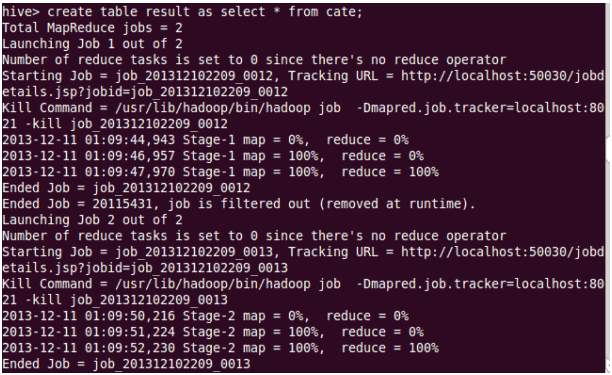
Select category, sum( amount) from txt records group by category

It calculates the amount of same category.

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

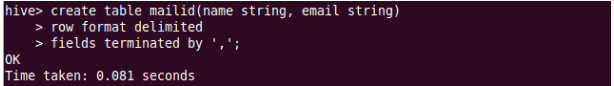
The result one table is stored in to another table.

Create table newtablename as select \* from oldtablename;

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

## ****Join Command :****

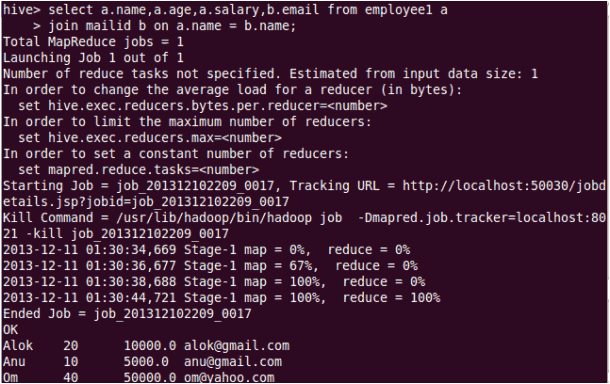
Here one more table is created in the name**‘mailid’**

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

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

## ****Join Operation****:

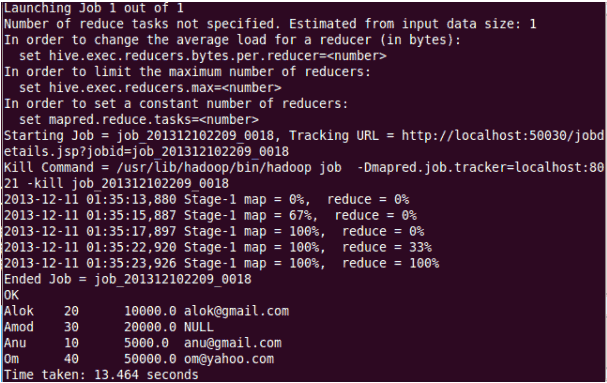
A Join operation is performed to combining fields from two tables by using values common to each.

[](https://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).

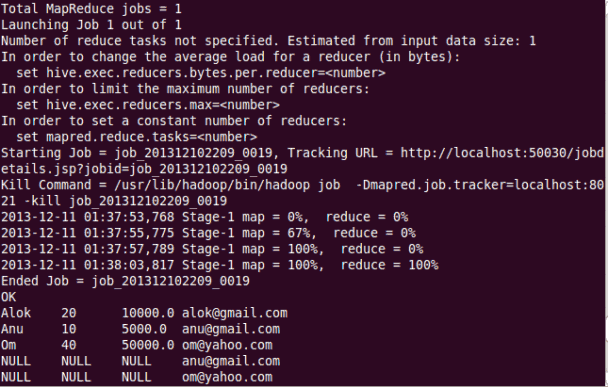
[Left Outer Join](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/54-1.png)

[](https://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.

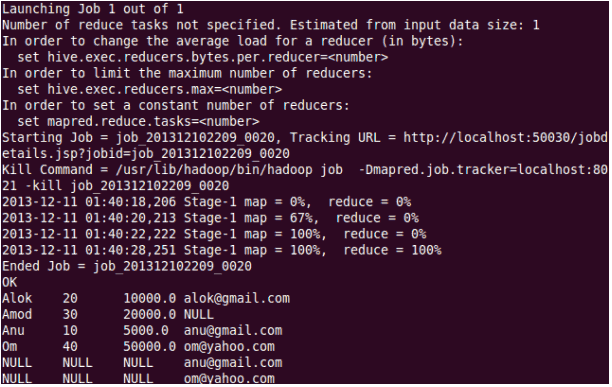
[Right Outer Join](https://cdn.edureka.co/blog/wp-content/uploads/2014/03/56-1.png)

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

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

HiveQL Manipulations

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.