Top Hive Commands with Examples in HQL

Recommended by 189 users

Awanish (https://www.edureka.co/blog/author/awanish/) | Sep 05,2018

In this blog post, let's discuss top Hive commands with examples.

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.

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

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.

```
hive> create database retail;
OK
Time taken: 5.275 seconds
hive>
```

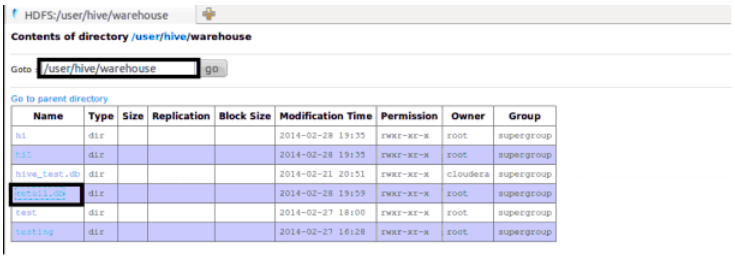
(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/201.png)

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

```
hive> show databases;
OK
default
retail
Time taken: 0.228 seconds
hive>
```

(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>**

```
hive> use retail;
OK
Time taken: 0.023 seconds
hive>
```

(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.

txns1.txt

```
00000000,06-26-2011,4007024,040.33,Exercise & Fitness,Cardio Machine Accessories,Clarksville,Tennessee,credit
00000001,05-26-2011,4006742,198.44,Exercise & Fitness,Weightlifting Gloves,Long Beach,California,credit
00000002,06-01-2011,4009775,005.58,Exercise & Fitness,Weightlifting Machine Accessories,Anaheim,California,credit
00000003,06-05-2011,4002199,198.19,Gymnastics,Gymnastics Rings,Milwaukee,Wisconsin,credit
00000004,12-17-2011,4002613,098.81,Team Sports,Field Hockey,Nashville ,Tennessee,credit
00000005,02-14-2011,4007591,193.63,Outdoor Recreation,Camping & Backpacking & Hiking,Chicago,Illinois,credit
00000006,10-28-2011,4002190,027.89,Puzzles,Jigsaw Puzzles,Charleston,South Carolina,credit
00000007,07-14-2011,4002964,096.01,Outdoor Play Equipment,Sandboxes,Columbus,Ohio,credit
00000008,01-17-2011,4007361,010.44,Winter Sports,Snowmobiling,Des Moines,Iowa,credit
00000009,05-17-2011,4004798,152.46,Jumping,Bungee Jumping,St. Petersburg,Florida,credit
```

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

```
cloudera@cloudera-vm:~$ hadoop dfs -copyFromLocal Desktop/blog/txns1.txt hdfs:/
cloudera@cloudera-vm:~$
```

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.

```
hive> create table txnrecords(txnno INT, txndate STRING, custno INT, amount DOUBLE,category STRING, product STRING, city STRIN
G, state STRING, spendby STRING) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 1.163 seconds
hive>
```

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

Describe provides information about the schema of the table.

```
txnno   int
txndate string
custno  int
amount  double
category        string
product string
city    string
state   string
spendby string
Time taken: 0.122 seconds
hive>
```

(https://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.

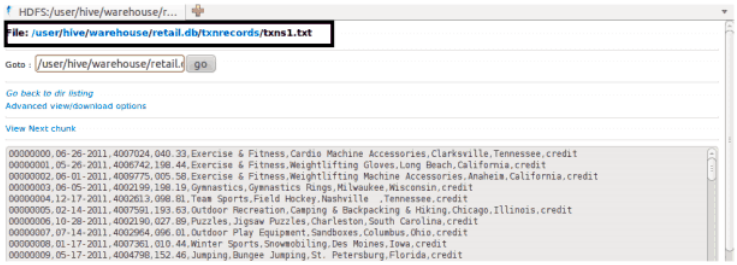
Syntax :

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.

```
hive> LOAD DATA INPATH '/txns1.txt' OVERWRITE INTO TABLE txnrecords;
Loading data to table retail.txnrecords
Deleted hdfs://localhost/user/hive/warehouse/retail.db/txnrecords
OK
Time taken: 0.263 seconds
hive>
```

(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

```
hive> create table customer(custno string, firstname string, lastname string, age int,profession string) row format delimited
fields terminated by ',';
OK
Time taken: 0.182 seconds
hive>
```

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

```
hive> load data local inpath '/home/cloudera/Desktop/blog/custs' into table customer;
Copying data from file:/home/cloudera/Desktop/blog/custs
Copying file: file:/home/cloudera/Desktop/blog/custs
Loading data to table retail.customer
OK
Time taken: 0.227 seconds
hive>
```

(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.

```
hive> select count(*) from txnrecords;
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201402270420_0005, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201402270420_0005
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:8021 -kill job_201402270420_0005
2014-02-28 20:02:41,231 Stage-1 map = 0%,  reduce = 0%
2014-02-28 20:02:48,293 Stage-1 map = 50%,  reduce = 0%
2014-02-28 20:02:49,309 Stage-1 map = 100%,  reduce = 0%
2014-02-28 20:02:55,350 Stage-1 map = 100%,  reduce = 33%
2014-02-28 20:02:56,367 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201402270420_0005
OK
50000
```

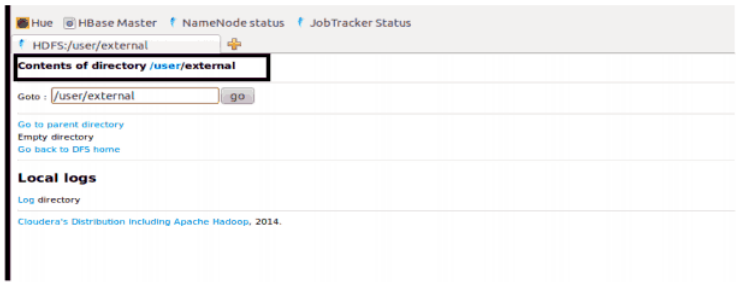
(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/32-1.png)

'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.

```
hive> create external table example_customer(custno string, firstname string, lastname string, age int,profession string) row
format delimited fields terminated by ',' LOCATION '/user/external';
OK
Time taken: 0.059 seconds
hive>
```

(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:

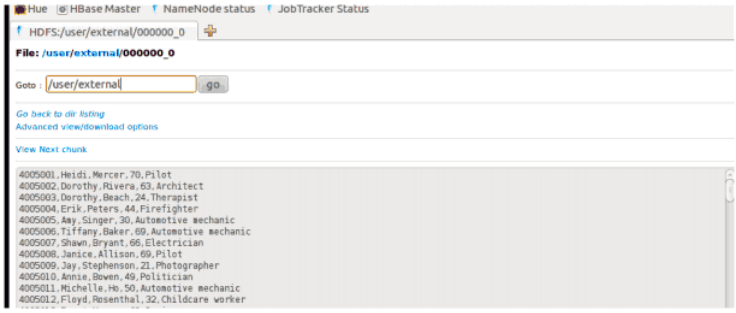
The **insert** command is used to load the data Hive table. Inserts can be done to a table or a partition.

• 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)

```
hive> from customer cus insert overwrite table example_customer select cus.custno,cus.firstname,cus.lastname,cus.age,cus.profe
ssion;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201402270420_0007, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201402270420_0007
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:8021 -kill job_201402270420_0007
2014-02-28 20:40:39,866 Stage-1 map = 0%,  reduce = 0%
2014-02-28 20:40:41,871 Stage-1 map = 100%,  reduce = 0%
2014-02-28 20:40:42,876 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201402270420_0007
Loading data to table retail.example_customer
Deleted hdfs://localhost/user/external
Table retail.example_customer stats: [num_partitions: 0, num_files: 0, num_rows: 0, total_size: 0]
9999 Rows loaded to example_customer
OK
Time taken: 5.786 seconds
hive>
```

(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.

```
hive> create table txnrecsByCat(txnno INT, txndate STRING, custno INT, amount DOUBLE,product STRING, city STRING, state STRING
, spendby STRING) partitioned by (category STRING) clustered by (state) INTO 10 buckets row format delimited fields terminated
 by ',' stored as textfile;
OK
Time taken: 0.101 seconds
hive>
```

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

```
hive> from txnrecords txn INSERT OVERWRITE TABLE record PARTITION(category)select txn.txnno,txn.txndate,txn.custno,txn.amount,
txn.product,txn.city,txn.state,txn.spendby, txn.category;
FAILED: Error in semantic analysis: Dynamic partition strict mode requires at least one static partition column. To turn this
off set hive.exec.dynamic.partition.mode=nonstrict
```

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

maurer/) at dresshead website (http://www.dresshead.com)). So we need to set the following parameters in Hive shell.

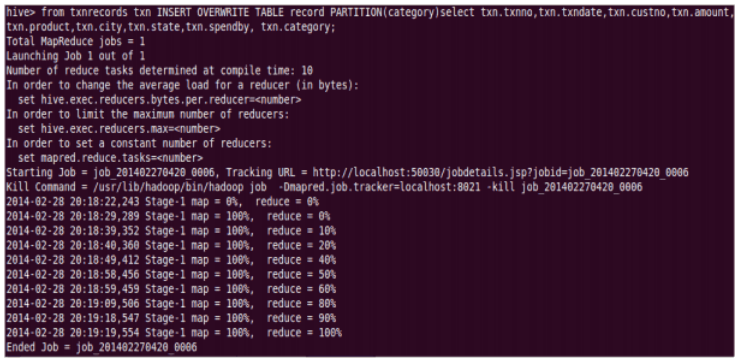
set hive.exec.dynamic.partition=true;

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

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

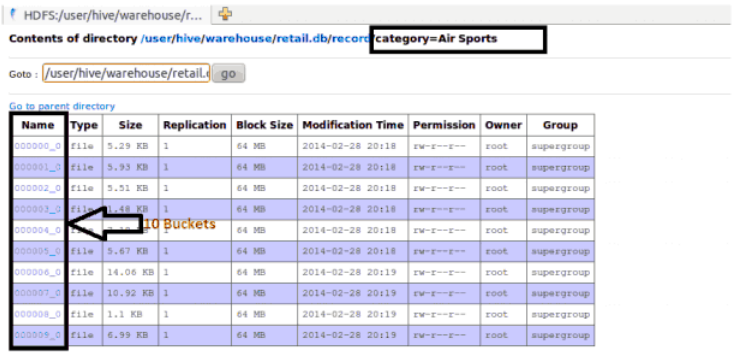
(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/39-1.png)

(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)

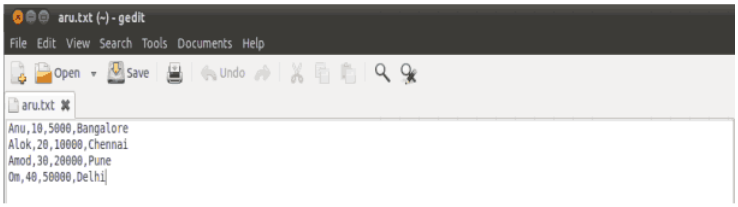
Partition is done by the category and can be divided in to buckets by using the 'Clustered By' command.



(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.

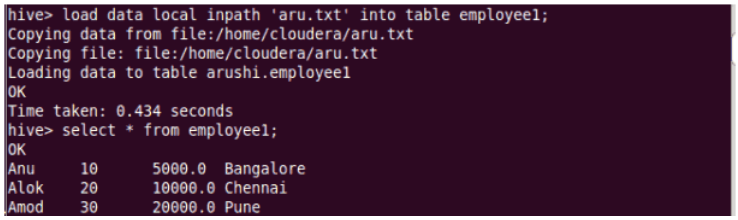
(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/43-1.png)



(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;

```
hive> select count(*) from employee1;
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0008, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0008
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0008
2013-12-11 00:58:36,125 Stage-1 map = 0%,  reduce = 0%
2013-12-11 00:58:39,154 Stage-1 map = 100%,  reduce = 0%
2013-12-11 00:58:46,204 Stage-1 map = 100%,  reduce = 33%
2013-12-11 00:58:47,214 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0008
OK
4
Time taken: 14.897 seconds
```

(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.

```
hive> select * from cate;
OK
category1       1000
category2       200
category1       1000
category3       5000
category2       200
category1       1000
category2       200
category1       1000
category2       200
category3       5000
Time taken: 0.219 seconds
```

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

```
hive> select count(distinct f1) from cate;
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0010, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0010
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0010
2013-12-11 01:04:07,180 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:04:09,190 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:04:16,224 Stage-1 map = 100%,  reduce = 33%
2013-12-11 01:04:17,231 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0010
OK
3
Time taken: 13.577 seconds
```

(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.

```
hive> select f1, sum(f2) from cate group by f1;
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0011, Tracking URL = http://localhost:50030/jobd
```

```
21 -kill job_201312102209_0011
2013-12-11 01:05:36,284 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:05:38,292 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:05:45,326 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0011
OK
category1       4000
category2       800
category3       10000
Time taken: 12.453 seconds
```

(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;

```
hive> create table result as select * from cate;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201312102209_0012, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0012
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0012
2013-12-11 01:09:44,943 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:09:46,957 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:09:47,970 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0012
Ended Job = 20115431, job is filtered out (removed at runtime).
Launching Job 2 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201312102209_0013, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0013
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0013
2013-12-11 01:09:50,216 Stage-2 map = 0%,  reduce = 0%
2013-12-11 01:09:51,224 Stage-2 map = 100%,  reduce = 0%
2013-12-11 01:09:52,230 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0013
```

(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'**

```
hive> create table mailid(name string, email string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.081 seconds
```

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

```
hive> select * from mailid;
OK
anu     anu@gmail.com
om      om@yahoo.com
Anu     anu@gmail.com
Om      om@yahoo.com
Alok    alok@gmail.com
Time taken: 0.126 seconds
```

(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.

```
hive> select a.name,a.age,a.salary,b.email from employee1 a
    > join mailid b on a.name = b.name;
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0017, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0017
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0017
2013-12-11 01:30:34,669 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:30:36,677 Stage-1 map = 67%,  reduce = 0%
2013-12-11 01:30:38,688 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:30:44,721 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0017
OK
Alok    20      10000.0 alok@gmail.com
Anu     10      5000.0  anu@gmail.com
Om      40      50000.0 om@yahoo.com
```

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

Left Outer Join:

```
hive> select a.name,a.age,a.salary,b.email from employee1 a
    > left outer join mailid b on a.name = b.name;
```

(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/54-1.png)

```
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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0018, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0018
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0018
2013-12-11 01:35:13,880 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:35:15,887 Stage-1 map = 67%,  reduce = 0%
2013-12-11 01:35:17,897 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:35:22,920 Stage-1 map = 100%,  reduce = 33%
2013-12-11 01:35:23,926 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0018
OK
Alok    20      10000.0 alok@gmail.com
Amod    30      20000.0 NULL
Anu     10      5000.0  anu@gmail.com
Om      40      50000.0 om@yahoo.com
Time taken: 13.464 seconds
```

(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.

```
hive> select a.name,a.age,a.salary,b.email from employee1 a
    > right outer join mailid b on a.name = b.name;
```

(https://cdn.edureka.co/blog/wp-content/uploads/2014/03/56-1.png)

```
Total MapReduce 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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0019, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0019
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0019
2013-12-11 01:37:53,768 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:37:55,775 Stage-1 map = 67%,  reduce = 0%
2013-12-11 01:37:57,789 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:38:03,817 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0019
OK
Alok    20      10000.0 alok@gmail.com
Anu     10      5000.0  anu@gmail.com
Om      40      50000.0 om@yahoo.com
NULL    NULL    NULL    anu@gmail.com
NULL    NULL    NULL    om@yahoo.com
```

(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.

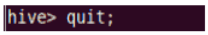
```
hive> select a.name,a.age,a.salary,b.email from employee1 a
    > full join mailid b on a.name = b.name;
```

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

```
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 mapred.reduce.tasks=<number>
Starting Job = job_201312102209_0020, Tracking URL = http://localhost:50030/jobd
etails.jsp?jobid=job_201312102209_0020
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:80
21 -kill job_201312102209_0020
2013-12-11 01:40:18,206 Stage-1 map = 0%,  reduce = 0%
2013-12-11 01:40:20,213 Stage-1 map = 67%,  reduce = 0%
2013-12-11 01:40:22,222 Stage-1 map = 100%,  reduce = 0%
2013-12-11 01:40:28,251 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201312102209_0020
OK
Alok    20      10000.0 alok@gmail.com
Amod    30      20000.0 NULL
```

(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.



(https://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.

Learn Big Data Hadoop Today

(https://www.edureka.co/big-data-and-hadoop)

Got a question for us? Please mention it in the comments section and we will get back to you.

Related Posts:

7 Ways Big Data Training Can Change Your Organization (https://www.edureka.co/blog/7-ways-big-data-training-can-change-your-organization/)

Get Started with Big Data & Hadoop (https://www.edureka.co/big-data-and-hadoop)

Hive Data Models (https://www.edureka.co/blog/hive-data-models/)

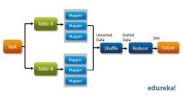
About Awanish (19 Posts (https://www.edureka.co/blog/author/awanish/))

Awanish is a Sr. Research Analyst at Edureka. He has rich expertise in Big Data technologies like Hadoop, Spark, Storm, Kafka, Flink. Awanish also holds good knowledge on Devops tools like Git, Jenkins, Docker, Puppet, Nagios. He is passionate about new technologies which provide realistic solutions.

Share on (https://www.linkedin.com/shareArticle?u=https://www.edureka.co/blog/hive-commands-with-examples) (https://plus.google.com/share?url=https://www.edureka.co/blog/hive-commands-with-examples) (https://www.linkedin.com/shareArticle?u=https://www.edureka.co/blog/hive-commands-with-examples)

PREVIOUS NEXT

Related Posts



Map Side Join Vs. Join
63.6K
(https://www.edureka.co/blog/map-side-join-vs-join/)

How to Run Hive Scripts?
63.5K
(https://www.edureka.co/blog/how-to-run-hive-scripts/)

Apache Hive Installation on Ubuntu
80.2K
(https://www.edureka.co/blog/apache-hive-installation-on-ubuntu)

How to Set Up Hadoop Cluster with HDFS High Availability
46.8K
(https://www.edureka.co/blog/how-to-set-up-hadoop-cluster-with-hdfs-high-availability/)

Browse Categories

Big Data NoSQL (https://www.edureka.co/blog/category/big-data-nosql/) Blockchain (https://www.edureka.co/blog/category/blockchain/)

Business Intelligence (https://www.edureka.co/blog/category/business-intelligence/) Cloud Computing (https://www.edureka.co/blog/category/cloud-computing/)

Finance (https://www.edureka.co/blog/category/finance/) Frameworks (https://www.edureka.co/blog/category/frameworks/)

Marketing (https://www.edureka.co/blog/category/marketing/) Mobile Development (https://www.edureka.co/blog/category/mobile-development/)

Operations (https://www.edureka.co/blog/category/operations/) Programming (https://www.edureka.co/blog/category/programming/)

Project Management (https://www.edureka.co/blog/category/project-management/) Robotic Process Automation (https://www.edureka.co/blog/category/robotic-process-automation/)

Success Story (https://www.edureka.co/blog/category/success-story/) Systems & Architecture (https://www.edureka.co/blog/category/systems-architecture/)

Systems Engineering (https://www.edureka.co/blog/category/systems-engineering/) Testing (https://www.edureka.co/blog/category/testing/)

Comments

60 Comments

Reply • Share ›

meea • a month ago
Hive supports overwriting or apprehending data.. Can you explain this
Reply • Share ›

Keerthana Reddy • 2 months ago
Awesome blog... Can you please send me some realtime projects to this mail: keerthanareddy1311@gmail.com
Reply • Share ›

Sivabalan Sethu • 7 months ago
It's really very helpful blog.. Can you please share one realtime project in hadoop , flow and architecture. sivabalan.sethu@gmail.com
Reply • Share ›

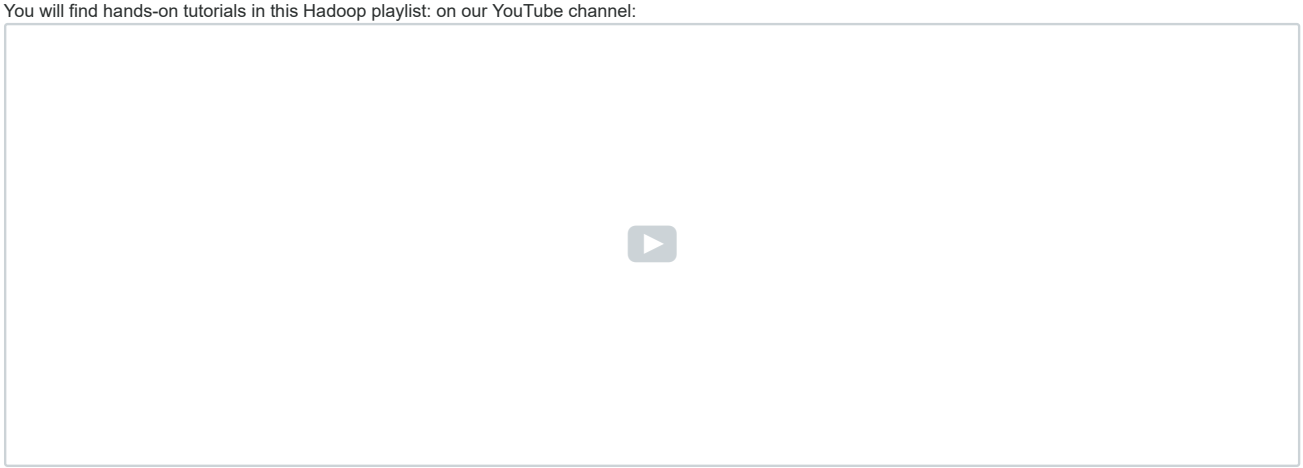
Ravi Kishore • a year ago
Super stuff.. Please help me with sample real time project in hadoop. My email ID - ravikishore223@gmail.com
Reply • Share ›

Sankeerna • a year ago
Hey.. Cool stuff... Got a good insight on hive.. Do you have any real time projects which you can share. This would be of great help to me in understanding hive better.
My mail id - sankeerna.lkmails@gmail.com..
Thanks in advance.
Reply • Share ›

Rathika Prabhu • a year ago
very understandable.watched your youtube video too.can u send me real projects related to this will be more useful.My email id is rathikarathinam@gmail.com
Reply • Share ›

Dhanashri Patil • a year ago
Very Nice blog, Can you please take one sample real time project in Hadoop and explain flow and architecture.
Plz mail to patildhanashri165@gmail.com
Reply • Share ›

EdurekaSupport Mod → Dhanashri Patil • a year ago
Hey Dhanashri, thanks for checking out our blog.
You will find hands-on tutorials in this Hadoop playlist: on our YouTube channel:



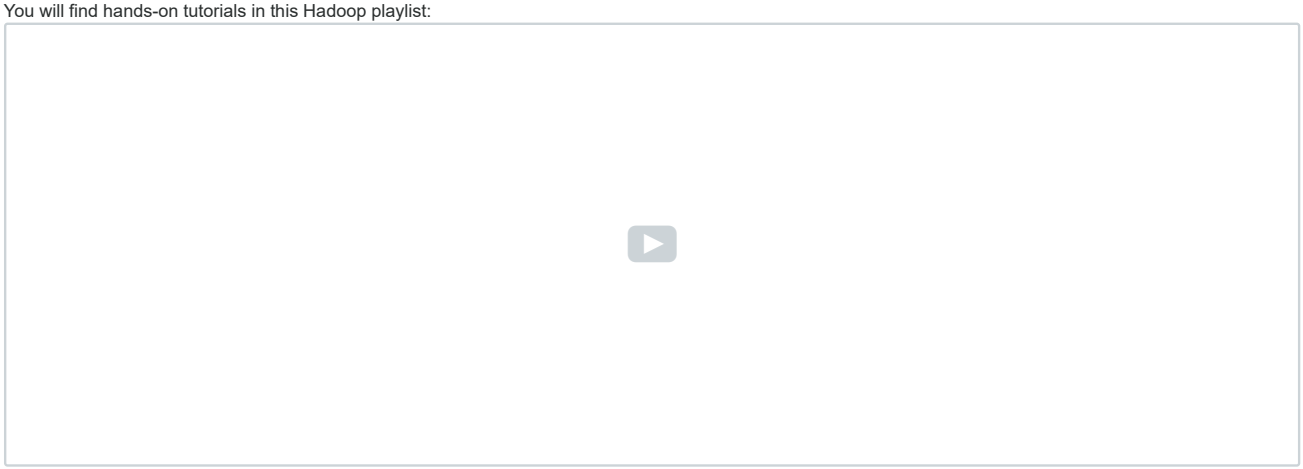
.
If you would like to avail practical training with real-time projects, you can can enroll into our Hadoop course here: https://www.edureka.co/big-....
Hope this helps. Cheers!
Reply • Share ›

Vishal • a year ago
Very nice Blog...

• Reply • Share ›

EdurekaSupport Mod ➚ Vishal • a year ago

Hey Vishal, thanks for checking out our blog. We're glad you liked it.
You will find hands-on tutorials in this Hadoop playlist:



.
If you would like to avail practical training with real-time projects, you can can enroll into our Hadoop course here: https://www.edureka.co/big-.... We have also emailed a sample set.
Hope this helps. Cheers!

• Reply • Share ›

peeyush sharma • 2 years ago

Awesome blog... Can you please help me with some one sample realtime project in hadoop.

• Reply • Share ›

EdurekaSupport Mod ➚ peeyush sharma • 2 years ago

Hey Peeyush, thanks for checking out our blog. Could you please share your email address. Cheers!

• Reply • Share ›

suma malladi ➚ EdurekaSupport • 8 months ago

Can you please help me with some one sample realtime project in hadoop. Email address is sumachowdarym95@gmail.com

• Reply • Share ›

peeyush sharma ➚ EdurekaSupport • 2 years ago

Hi
Email address is sharmapeeyush3@gmail.com

• Reply • Share ›

purni • 2 years ago

Very helpful blog.. Can you please take one sample realtime project in hadoop and explain flow and architecture.

• Reply • Share ›

EdurekaSupport Mod ➚ purni • 2 years ago

Hey Purni, thanks for checking out our blog. Please share your contact email address with us and we will get in touch. Cheers!

• Reply • Share ›

purni ➚ EdurekaSupport • 2 years ago

purni_04@yahoomail.com

• Reply • Share ›

EdurekaSupport Mod ➚ purni • 2 years ago

Hi, we have sent some information to you. Cheers!

• Reply • Share ›

myspace ➚ EdurekaSupport • a year ago

Very Nice blog, Can you please take one sample realtime project in hadoop and explain flow and architecture.
plz mail to mailstomyspace@gmail.com

• Reply • Share ›

EdurekaSupport Mod ➚ myspace • a year ago

+myspace, thanks for checking out our blog. We're glad you liked it.
We have sent it. Cheers!

• Reply • Share ›

Rajiv • 2 years ago

Sir, I want to know how insurance company use big data to improve their customer experience using an example. if possible then plz explain flow diagram. like how many

sir pls answer??
Reply • Share ›

EdurekaSupport Mod → Rajiv • 2 years ago
Hey Rajiv, thanks for checking out our blog. Here are a couple of related blogs that you can refer to for the explanation:
https://goo.gl/EoTFe8
https://goo.gl/BxNwye
Please feel free to write to us if you have any questions. Hope this helps. Cheers!
Reply • Share ›

Rajiv • 2 years ago
very helpful blog..nice content..easy to understand ..Sir i want to know that is all data transfered in hadoop is in serialized form
Reply • Share ›

EdurekaSupport Mod → Rajiv • 2 years ago
Hey Rajiv, thanks for checking out our blog. With regard to your query, yes, the serialized data will be stored in Hive. To insert data into table, Hive creates an object by using Java. To transfer java objects over network, the data should be serialized. Each field serialized by using Object inspector and finally serialized data stored in Hive table. Hope this helps. Cheers!
Reply • Share ›

Rajiv → EdurekaSupport • 2 years ago
thanks sir for doubt clerification
Reply • Share ›

Vissu Kangarla • 2 years ago
please tell me the query how to split the information contained in the table in to another table
Reply • Share ›

EdurekaSupport Mod → Vissu Kangarla • 2 years ago
Hey Vissu, thanks for checking out our blog. You can insert existing data from a table into another table by using select query.
INSERT INTO table new_table select columnlist FROM secondtable where soma_conditions;
Hope this helps. Cheers!
Reply • Share ›

Gopesh kumar • 2 years ago
i became much acquainted with HIVE with help of this blog.
Reply • Share ›

EdurekaSupport Mod → Gopesh kumar • 2 years ago
We are happy we could help you understand Hive concepts. DO keep checking back in for more videos on all your favourite topics.
Reply • Share ›

Gopesh kumar • 2 years ago
Nice blog ....i helped me a lot.
Reply • Share ›

EdurekaSupport Mod → Gopesh kumar • 2 years ago
We are so happy we could help. Do keep checking back in for more blogs on all your favourite topics!
Reply • Share ›

Mridula Bvs • 3 years ago
This blog is wonderful and very informative.
Can I also get some information regarding MLlib and kafka?
Thanks in advance :)
Reply • Share ›

EdurekaSupport Mod → Mridula Bvs • 3 years ago
Hi Mridula,
Thank you for the positive feedback. Keep following our blog for more interesting and informative posts.
We would recommend that you get in touch with us for further clarification on Apache Kafka by contacting our sales team on +91-8880862004 (India) or 1800 275 9730 (US toll free). You can mail us on sales@edureka.co.
Reply • Share ›

Sabarinadh • 3 years ago
nice blog.. this blog helps beginners to work and understand the HIVE commands. very useful... :)
Reply • Share ›

EdurekaSupport Mod → Sabarinadh • 3 years ago
Hi Sabarinadh,
Thank you for your positive feedback. We hope that you will find our blog useful in future as well.
Keep visiting the Edureka Blog page for latest posts on this link: http://www.edureka.co/blog/
Reply • Share ›

Reply • Share ›

EdurekaSupport Mod ➚ Justin Ag • 3 years ago
Hi Justin, let us know your area of expertise and also do send couple of samples to content@edureka.co We will get back to you after reviewing it.
Reply • Share ›

Prateek Pathak • 3 years ago
Very Informative! Thanks
This should be tightly coupled with HBase also running on Hadoop
Reply • Share ›

EdurekaSupport Mod ➚ Prateek Pathak • 3 years ago
Thanks Prateek!
Reply • Share ›

Load more comments

ALSO ON HTTPS://WWW.EDUREKA.CO/BLOG/

Cybersecurity Tools You Must Know – Tools for Cyber Threats
1 comment • 2 months ago
Gvsm Chaithanya — what are the industrial applications for information security

Top 10 Reasons Why You Should Learn Selenium
1 comment • 6 months ago
Partho — Since there is no Global Certification in Selenium what is the benchmark then.....

Microservices Tutorial – Learn all about Microservices with Example
203 comments • 6 months ago
anibxl abril — Excellent Article. Can you please send source code to my email : anibxlabril@gmail.com

Top 10 Trending Technologies To Master In 2018
2 comments • 9 months ago
Mohammed Innat — Cool

Related Blogs


(https://www.side-join-vs-join/)

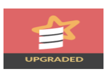

(https://www.to-run-hive-scripts/)
How to Run Hive Scripts? (https://www.edureka.co/blog/how-to-run-hive-scripts/)

(https://www.hive-installation-on-ubuntu)
Apache Hive Installation on Ubuntu (https://www.edureka.co/blog/apache-hive-installation-on-ubuntu)

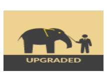

(https://www.questions/top-50-hadoop-interview-questions-2016/)
Top 50 Hadoop Interview Questions You Must Prepare In 2018 (https://www.edureka.co/blog/interview-questions/top-50-hadoop-interview-questions-2016/)

Big Data Analytics Courses


(/big-data-and-hadoop)
Big Data Hadoop Certification Training (/big-data-and-hadoop)



(/apache-spark-scala-training)
Apache Spark and Scala Certification Training (/apache-spark-scala-training)



(/hadoop-admin)
Hadoop Administration Certification Training (/hadoop-admin)


(/splunk)
Splunk Power User & Admin Certification Training (/splunk)