Assignment 7.3

Problem Statement:

● Hive Data Definitions

● Hive Data Manipulations

● HiveQL Manipulations

Solution

● Hive Data Definitions

Hive Data Definition Language (DDL) is a subset of Hive SQL statements that describe the data structure in Hive by creating, deleting, or altering schema objects such as databases, tables, views, partitions, and buckets. Most Hive DDL statements start with the keywords CREATE, DROP, or ALTER. The syntax of Hive DDL is very similar to the DDL in SQL. The comments in Hive start from --

DDL Commands in Hive

1. CREATE Database,Table
2. DROP Database,Table
3. TRUNCATE Table
4. ALTER Database,Table
5. SHOW Databases,Tables,Table Properties,Partitions,Functions,Index
6. DESCRIBE Database, Table ,View

1. Create commands

Create Database in Hive

This DDL command in Hive is used for creating databases.

```
CREATE (DATABASE) [IF NOT EXISTS] database_name
  [COMMENT database_comment]
  [LOCATION hdfs_path]
  [WITH DBPROPERTIES (property_name=property_value, ...)];
```

for example

create database if not exists firstDB comment
"This is my hive database" location '/user/hive/warehouse/newdb'
with DBPROPERTIES ('createdby'='sundeep','createdfor'='acadgild');

```
hive> create database if not exists firstDB comment
    > "This is my hive database" location '/user/hive/warehouse/newdb'
    > with DBPROPERTIES ('createdby'='sundeep','createdfor'='acadgild');
OK
Time taken: 2.416 seconds
hive>
```

Use Database Command in Hive

This hive command is used to select a specific database for the session on which hive queries would be executed.

use firstDB;

```
hive> use firstDB;
OK
Time taken: 0.209 seconds
hive>
```

Create Table Command in Hive

Hive create table command is used to create a table in the existing database that is in use for a particular session.

```
CREATE  TABLE [IF NOT EXISTS] [db_name.]table_name    --
  [(col_name data_type [COMMENT col_comment], ...)]
  [COMMENT table_comment]
   [LOCATION hdfs_path]
```

for example

create table employee(
name string,
skill string,
rank int,
code string
)
row format delimited fields terminated by ',';
It will create the table employee with column name

```
hive> create table employee(
    > name string,
    > skill string,
    > rank int,
    > code string
    > )
    > row format delimited fields
    > terminated by ',';
OK
```

Create a table in hive by copying an existing table schema

Hive lets programmers create a new table by replicating the schema of an existing table but remember only the schema of the new table is replicated but not the data. When creating the new table, the location parameter can be specified.

CREATE TABLE [IF NOT EXISTS] [db_name.]table_name Like [db_name].existing_table [LOCATION hdfs_path]

2. Drop commands

Drop Database in Hive

This command is used for deleting an already created database in Hive and the syntax is as follows -

DROP (DATABASE) [IF EXISTS] database_name [RESTRICT|CASCADE];

```
hive> drop database if exists firstDB CASCADE;
OK
Time taken: 1.186 seconds
hive>
```

DROP Table Command in Hive

Drops the table and all the data associated with it in the Hive metastore.

DROP TABLE [IF EXISTS] table_name [PURGE];

3. Truncate commands

TRUNCATE Table Command in Hive

This hive command is used to truncate all the rows present in a table i.e. it deletes all the data from the Hive meta store and the data cannot be restored.

TRUNCATE TABLE [db_name].table_name

Usage of TRUNCATE Table in Hive

truncate table employee;

```
hive> select * from employee;
OK
Ramesh  Java    2       ABC
Suresh  Java    3       CDE
Time taken: 0.58 seconds, Fetched: 2 row(s)
hive>
```

```
hive> describe formatted employee;
OK
# col_name              data_type               comment

name                    string
skill                   string
rank                    int
code                    string

# Detailed Table Information
Database:               firstdb
Owner:                  acadgild
CreateTime:             Tue Oct 24 11:16:45 IST 2017
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://localhost:9000/user/hive/warehouse/newdb/employee

Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   true
        numFiles                2
```

```
        numRows                 2
        rawDataSize             34
        totalSize               36
        transient_lastDdlTime   1508826763

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.583 seconds, Fetched: 35 row(s)
hive> truncate table employee;
OK
Time taken: 0.84 seconds
hive> describe formatted employee;
OK
# col_name              data_type               comment

name                    string
skill                   string
rank                    int
code                    string

# Detailed Table Information
Database:               firstdb
Owner:                  acadgild
CreateTime:             Tue Oct 24 11:16:45 IST 2017
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://localhost:9000/user/hive/warehouse/newdb/employee

Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   false
        numFiles                2
        numRows                 -1
        rawDataSize             -1
        totalSize               36
        transient_lastDdlTime   1508827090

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.603 seconds, Fetched: 35 row(s)
hive>
```

4. Alter commands

Alter Database Command in Hive

Whenever the developers need to change the metadata of any of the databases, alter hive DDL command can be used as follows –

ALTER (DATABASE) database_name SET DBPROPERTIES (property_name=property_value, ...);

alter database firstDB set OWNER ROLE admin;

```
hive> alter database firstDB set OWNER ROLE admin;
OK
Time taken: 0.238 seconds
```

ALTER Table Command in Hive

Using ALTER Table command, the structure and metadata of the table can be modified even after the table has been created. Let's try to change the name of an existing table using the ALTER command –

ALTER TABLE [db_name].old_table_name RENAME TO [db_name].new_table_name;

ALTER TABLE employee RENAME TO employee_details;

```
hive> ALTER TABLE employee RENAME TO employee_details;
OK
Time taken: 0.84 seconds
```

4. Show commands

Show Database Command in Hive

Programmers can view the list of existing databases in the current schema.

Usage of Show Database Command

Show databases;

```
hive> Show databases;
OK
acadgild_db
custom
default
firstdb
Time taken: 0.534 seconds, Fetched: 4 row(s)
```

Show Table Command in Hive

Gives the list of existing tables in the current database schema.

Usage of Show tables Command

Show tables;

```
hive> show tables;
OK
employee
Time taken: 0.267 seconds, Fetched: 1 row(s)
```

5. Describe commands

DESCRIBE Table Command in Hive

Gives the information of a particular table and the syntax is as follows –

DESCRIBE [EXTENDED|FORMATTED] [db_name.] table_name[.col_name ([.field_name]

describe employee

```
hive> describe  employee;
OK
name                    string
skill                   string
rank                    int
code                    string
Time taken: 1.073 seconds, Fetched: 4 row(s)
```

describe formatted employee;

```
hive> describe formatted employee;
OK
# col_name              data_type               comment

name                    string
skill                   string
rank                    int
code                    string

# Detailed Table Information
Database:               firstdb
Owner:                  acadgild
CreateTime:             Tue Oct 24 11:16:45 IST 2017
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://localhost:9000/user/hive/warehouse/newdb/employee
Table Type:             MANAGED_TABLE
Table Parameters:
        transient_lastDdlTime   1508824005

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.972 seconds, Fetched: 30 row(s)
hive>
```

```
hive> describe extended employee;
OK
name                    string
skill                   string
rank                    int
code                    string

Detailed Table Information      Table(tableName:employee, dbName:firstdb, owner:
acadgild, createTime:1508824005, lastAccessTime:0, retention:0, sd:StorageDescri
ptor(cols:[FieldSchema(name:name, type:string, comment:null), FieldSchema(name:s
kill, type:string, comment:null), FieldSchema(name:rank, type:int, comment:null)
, FieldSchema(name:code, type:string, comment:null)], location:hdfs://localhost:
9000/user/hive/warehouse/newdb/employee, inputFormat:org.apache.hadoop.mapred.Te
xtInputFormat, outputFormat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutput
Format, compressed:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, serializ
ationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{field.d
elim=,, serialization.format=,}), bucketCols:[], sortCols:[], parameters:{}, ske
wedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedColValueLocation
Maps:{}), storedAsSubDirectories:false), partitionKeys:[], parameters:{transient
_lastDdlTime=1508824005}, viewOriginalText:null, viewExpandedText:null, tableTyp
e:MANAGED_TABLE)
Time taken: 0.932 seconds, Fetched: 6 row(s)
```

● Hive Data Manipulations

DML (Data Manipulation Language) commands in Hive are used for inserting and querying the data from hive tables once the structure and architecture of the database has been defined using the DDL.

Data can be loaded into Hive tables using –

- LOAD command
- Insert command

Usage of LOAD Command for Inserting Data Into Hive Tables

Syntax for Load Command in Hive

LOAD DATA [LOCAL] INPATH 'hdfsfilepath/localfilepath' [OVERWRITE] INTO TABLE existing_table_name

sample dataset 'emp_details.txt'
Amit,Big Data,1,BBSR
Venkat,Web Technology,2,BBSR
Aditya,DBA,1,BNG
Ravinder,Java,2,BBSR
Sunil,C#,1,BBSR
Anil,ASP,2,BNG
Mihir,Big Data,3,BBSR
Mohit,Java,1,BBSR

load data local inpath 'emp_details.txt' into table employee;

```
hive> select * from employee;
OK
Suresh  Java    3       CDE
Time taken: 0.289 seconds, Fetched: 1 row(s)
hive> load data local inpath 'emp_details.txt' into table employee;
Loading data to table firstdb.employee
Table firstdb.employee stats: [numFiles=2, numRows=0, totalSize=177, rawDataSize
=0]
OK
Time taken: 1.672 seconds
hive> select * from employee;
OK
Suresh  Java    3       CDE
Amit    Big Data        1       BBSR
Venkat  Web Technology  2       BBSR
Aditya  DBA     1       BNG
Ravinder        Java    2       BBSR
Sunil   C#      1       BBSR
Anil    ASP     2       BNG
Mihir   Big Data        3       BBSR
Mohit   Java    1       BBSR
Time taken: 0.305 seconds, Fetched: 9 row(s)
```

If the keyword LOCAL is not specified, then Hive will need absolute URI of the file. However, if local is specified then it assumes the following rules -

1. It will assume it's an HDFS path and will try to search for the file in HDFS.
2. If the path is not absolute, then hive will try to locate the file in the /user/ in HDFS.

Using the OVERWRITE keyword while importing means the data will be ingested i.e. it will delete old data and put new data otherwise it would just append the new data. The contents of the target table will be deleted and replaced by the files referred to by file path; otherwise the files referred by file path will be added to the table.

Usage of INSERT Command for Inserting Data Into Hive Tables

The INSERT statement lets you load data into a table from a query.

1.Using Values

Using Values command ,we can append more rows of data into existing table.

INSERT INTO table employee VALUES ('Punit','Java',2,'ABC');

```
hive>  INSERT INTO table employee VALUES ('Punit','Java',2,'ABC');
Query ID = acadgild_20171026022727_32af9e6c-a2b7-4624-9c5a-da2730e4415b
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_1508948047547_0012, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0012/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-10-26 02:27:27,030 Stage-1 map = 0%,  reduce = 0%
```

```
2017-10-26 02:27:43,158 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.79 se
c
MapReduce Total cumulative CPU time: 2 seconds 790 msec
Ended Job = job_1508948047547_0012
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: hdfs://localhost:9000/tmp/hive/acadgild/7b464010-279e-48d4-80f5-
77c38c22eb7f/hive_2017-10-26_02-27-07_812_6708197055589755238-1/-ext-10000
Loading data to table default.employee
Table default.employee stats: [numFiles=2, numRows=1, totalSize=176, rawDataSize
=16]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.79 sec   HDFS Read: 293 HDFS Write: 89
 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 790 msec
OK
Time taken: 37.991 seconds
```

2 Using Queries

You can also insert data using insert with select query output into a table.

INSERT INTO table employeeNew select * from employee;

```
hive> create table employeeNew(
    > name string,
    > skill string,
    > rank int,
    > code string
    > );
OK
Time taken: 0.825 seconds
hive> describe employeeNew;
OK
name                    string
skill                   string
rank                    int
code                    string
Time taken: 0.689 seconds, Fetched: 4 row(s)
```

```
hive> INSERT INTO table employeeNew select * from employee;
Query ID = acadgild_20171026022323_546f9e5d-ad9a-45b3-be4e-f21c361939cf
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_1508948047547_0011, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0011/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-10-26 02:23:38,244 Stage-1 map = 0%,  reduce = 0%
2017-10-26 02:23:54,448 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.58 se
c
MapReduce Total cumulative CPU time: 1 seconds 580 msec
Ended Job = job_1508948047547_0011
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
```

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-10-26 02:23:38,244 Stage-1 map = 0%,  reduce = 0%
2017-10-26 02:23:54,448 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.58 se
c
MapReduce Total cumulative CPU time: 1 seconds 580 msec
Ended Job = job_1508948047547_0011
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: hdfs://localhost:9000/tmp/hive/acadgild/7b464010-279e-48d4-80f5-
77c38c22eb7f/hive_2017-10-26_02-23-16_678_6695890311923289713-1/-ext-10000
Loading data to table default.employeenew
Table default.employeenew stats: [numFiles=1, numRows=8, totalSize=231, rawDataS
ize=223]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.58 sec   HDFS Read: 379 HDFS Write: 30
7 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 580 msec
OK
Time taken: 40.755 seconds
```

● HiveQL Manipulations

Hive provides SQL type querying language for the ETL purpose on top of Hadoop file system.

Hive Query language (HiveQL) provides SQL type environment in Hive to work with tables, databases, queries.

We can have a different type of Clauses associated with Hive to perform different type data manipulations and querying. For better connectivity with different nodes outside the environment. HIVE provide JDBC connectivity as well.

Syntax

SELECT [ALL | DISTINCT] select_expr, select_expr, ...

FROM table_reference

[WHERE where_condition]

[**GROUP BY** col_list]

[**HAVING** having_condition]

[**CLUSTER BY** col_list | [**DISTRIBUTE** BY col_list] [**SORT BY** col_list] [**ORDER BY** col_list]]]

[LIMIT number]

SELECT is the projection operator in SQL. The points are:

SELECT scans the table specified by the FROM clause

WHERE gives the condition of what to filter

GROUP BY gives a list of columns which specify how to aggregate the records

CLUSTER BY, DISTRIBUTE BY, SORT BY specify the sort order and algorithm

LIMIT specifies how many # of records to retrieve

here table used is emp_sal

```
hive> describe emp_sal;
OK
emp_id                  int
name                    string
sal                     int
dept                    string
Time taken: 0.875 seconds, Fetched: 4 row(s)
```

simple query to get all column data of table

```
hive> select * from emp_sal
    > ;
OK
1       Amit    105     Data Minning
2       Pankaj  85      Data Engineer
3       Kiran   110     Data Scientist
4       Arpitha 95      Data Engineer
5       Viraj   105     Data Mining
6       Smitha  80      Data Analyst
7       Supriya 90      Data Engineer
8       Vihan   120     Data Scientist
9       Emma    100     Data Engineer
10      Siddharath      100     Data Engineer
Time taken: 0.385 seconds, Fetched: 10 row(s)
```

GROUP BY Clauses

A GROUP BY clause is frequently used with aggregate functions, to group the result set by columns and apply aggregate functions over each group. Functions link count,max,avg can also be used to compute the grouping key.

To get department wise employee count

select dept,count(*) from emp_sal group by dept;

```
hive> select dept,count(*) from emp_sal group by dept;
Query ID = acadgild_20171026014242_f8e39caf-5071-42ba-b9ee-375a2b3ae02e
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_1508948047547_0005, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0005/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-26 01:42:22,844 Stage-1 map = 0%,  reduce = 0%
2017-10-26 01:42:43,777 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.93 se
c
2017-10-26 01:43:02,931 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.92
sec
MapReduce Total cumulative CPU time: 4 seconds 920 msec
Ended Job = job_1508948047547_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.92 sec   HDFS Read: 506 HDF
S Write: 77 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 920 msec
OK
Data Analyst    1
Data Engineer   5
Data Mining     1
Data Minning    1
Data Scientist  2
Time taken: 62.974 seconds, Fetched: 5 row(s)
```

HAVING Clauses

A HAVING clause lets you filter the groups produced by GROUP BY, by applying predicate operators to each groups.

To get department with number of employee having number of employee>1

select dept,count(*) from emp_sal group by dept having count(*)>1;

```
hive> select dept,count(*) from emp_sal group by dept having count(*)>1;
Query ID = acadgild_20171026014545_4759d23f-e23d-423a-8e91-64aedecb09b2
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_1508948047547_0006, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0006/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-26 01:45:44,419 Stage-1 map = 0%,  reduce = 0%
2017-10-26 01:45:55,821 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.4 sec
2017-10-26 01:46:12,646 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.6 s
ec
MapReduce Total cumulative CPU time: 4 seconds 600 msec
Ended Job = job_1508948047547_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.6 sec   HDFS Read: 506 HDFS
 Write: 33 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 600 msec
OK
Data Engineer   5
Data Scientist  2
Time taken: 46.608 seconds, Fetched: 2 row(s)
```

Order by:

The ORDER BY syntax in HiveQL is similar to the syntax of ORDER BY in SQL language.

Order by is the clause we use with "SELECT" statement in Hive queries, which guarantees total ordering of data. Order by clause use columns on Hive tables for grouping particular column values mentioned with Order by. For whatever the column name we are defining the order by clause the query will selects and display results by ascending or descending order the particular column values.

select * from emp_sal order by sal;

```
hive> select * from emp_sal order by sal;
Query ID = acadgild_20171026014949_e5d896c6-c474-4c15-b065-bc2b6ad3c317
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_1508948047547_0007, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0007/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-26 01:50:17,187 Stage-1 map = 0%,  reduce = 0%
2017-10-26 01:50:32,465 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.35 se
c
2017-10-26 01:50:49,177 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.63
sec
MapReduce Total cumulative CPU time: 3 seconds 630 msec
Ended Job = job_1508948047547_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.63 sec   HDFS Read: 506 HDF
S Write: 264 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 630 msec
OK
6       Smitha  80      Data Analyst
2       Pankaj  85      Data Engineer
7       Supriya 90      Data Engineer
4       Arpitha 95      Data Engineer
10      Siddharath      100     Data Engineer
9       Emma    100     Data Engineer
5       Viraj   105     Data Mining
1       Amit    105     Data Minning
3       Kiran   110     Data Scientist
8       Vihan   120     Data Scientist
Time taken: 53.217 seconds, Fetched: 10 row(s)
hive>
```

Sort by:

Sort by clause performs on column names of Hive tables to sort the output. We can mention DESC for sorting the order in descending order and mention ASC for Ascending order of the sort.

In this sort by it will sort the rows before feeding to the reducer. Always sort by depends on column types.

select * from emp_sal sort by sal;

```
hive> select * from emp_sal sort by sal;
Query ID = acadgild_20171026015353_dd0b01de-6424-49da-9fdc-fa49966dc808
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_1508948047547_0008, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0008/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-26 01:53:38,836 Stage-1 map = 0%,  reduce = 0%
2017-10-26 01:53:52,849 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.33 se
c
2017-10-26 01:54:08,629 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.41
sec
MapReduce Total cumulative CPU time: 3 seconds 410 msec
Ended Job = job_1508948047547_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.41 sec   HDFS Read: 506 HDF
S Write: 264 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 410 msec
OK
6       Smitha  80      Data Analyst
2       Pankaj  85      Data Engineer
7       Supriya 90      Data Engineer
4       Arpitha 95      Data Engineer
10      Siddharath      100     Data Engineer
9       Emma    100     Data Engineer
5       Viraj   105     Data Mining
1       Amit    105     Data Minning
3       Kiran   110     Data Scientist
8       Vihan   120     Data Scientist
Time taken: 49.697 seconds, Fetched: 10 row(s)
```

Difference between Sort By and Order By

Hive supports SORT BY which sorts the data per reducer. The difference between "order by" and "sort by" is that the former guarantees total order in the output while the latter only guarantees ordering of the rows within a reducer. If there are more than one reducer, "sort by" may give partially ordered final results.

Distribute By:

Distribute BY clause used on tables present in Hive. Hive uses the columns in Distribute by to distribute the rows among reducers. All Distribute BY columns will go to the same reducer.

- It ensures each of N reducers gets non-overlapping ranges of column
- It doesn't sort the output of each reducer

select * from emp_sal distribute by sal;

```
Time taken: 57.808 seconds, Fetched: 10 row(s)
hive> select * from emp_sal distribute  by sal;
Query ID = acadgild_20171026020202_66c0f39c-4023-4e15-8488-29fa22951dca
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_1508948047547_0010, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0010/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0010
 Interrupting... Be patient, this might take some time.
 Press Ctrl+C again to kill JVM
 killing job with: job_1508948047547_0010
 Hadoop job information for Stage-1: number of mappers: 0; number of reducers: 0
 2017-10-26 02:02:35,515 Stage-1 map = 0%,  reduce = 0%
 Ended Job = job_1508948047547_0010 with errors
 Error during job, obtaining debugging information...
 FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.Ma
 pRedTask
 MapReduce Jobs Launched:
 Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 FAIL
 Total MapReduce CPU Time Spent: 0 msec
 hive>
```

Cluster By:

Cluster By used as an alternative for both Distribute BY and Sort BY clauses in Hive-QL.

Cluster BY clause used on tables present in Hive. Hive uses the columns in Cluster by to distribute the rows among reducers. Cluster BY columns will go to the multiple reducers. It ensures sorting orders of values present in multiple reducers

select * from emp_sal cluster by sal;

```
hive> select * from emp_sal cluster by sal;
Query ID = acadgild_20171026015959_8bbf950e-3d74-4fb5-b650-b39dfce312c4
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_1508948047547_0009, Tracking URL = http://localhost:8088/prox
y/application_1508948047547_0009/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_15089480475
47_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-26 01:59:22,157 Stage-1 map = 0%,  reduce = 0%
2017-10-26 01:59:41,588 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.41 se
c
2017-10-26 01:59:59,325 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.56
sec
MapReduce Total cumulative CPU time: 5 seconds 560 msec
Ended Job = job_1508948047547_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.56 sec   HDFS Read: 506 HDF
S Write: 264 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 560 msec
OK
6       Smitha  80      Data Analyst
2       Pankaj  85      Data Engineer
7       Supriya 90      Data Engineer
4       Arpitha 95      Data Engineer
10      Siddharath      100     Data Engineer
9       Emma    100     Data Engineer
5       Viraj   105     Data Mining
1       Amit    105     Data Minning
3       Kiran   110     Data Scientist
8       Vihan   120     Data Scientist
Time taken: 57.808 seconds, Fetched: 10 row(s)
```