Hive Illustration : Basics

* *To get started with the hive shell,*

hive

* *To check what is present in the HDFS,*

hadoop fs – ls

* *To create a directory in the current path (let’s say the name is ‘foo’),*

hadoop fs - mkdir foo

* *To create a database, in the hive shell (let’s say the name is ‘vin\_emp’),*

create database vin\_emp;

* *To see existing databases,*

show databases;

* *To start using the database,*

use vin\_emp;

* *To check the tables present in the database,*

show tables;

* *To come out of the hive shell,*

quit;

* *to list contents of the current working directory,*

ls

* *To create a directory,*

mkdir myhivedata

* *To navigate into that data,*

cd myhivedata

* *To check the present working directory*

pwd

* *To check the contents of a file (name of the file is employees.txt),*

cat employees.txt

* *To create a table within the hive shell,*

create table emp\_global(id int,name string,city string,continent string)

row format delimited

fields terminated by ‘ , ‘

stored as textfile ;

* *To check the tables,*

show tables ;

* *To query the table,*

select \* from emp\_global ;

* *To load data into the table,*

load local inpath ‘employees.txt’ into table emp\_global;

* *To drop all tables inside a database,*

drop database vin\_emp cascade,

* *To know the schema of the table,*

describe emp\_global ;

* *To drop a table,*

drop table emp\_global ;

Hive Illustration : External tables in hive

* *To create a database in a certain desired location,*

create database vin\_emp\_loc location ‘/user/cloudera/myhivedata’ ;

* *To copy a file from local file system to hdfs location,*

hadoop fs – put empglobal.csv empdata

* *To see the contents of the file,*

hadoop df – cat empdata/empglobal.csv

Hive Illustration : Loading different file formats

* *How to know what the table type is, whether internal or external,*

describe extended emp\_global ;

* *To load the data into orc table,*

insert into table emp\_global\_orc select \* from emp\_global ;

* *Create a table whose schema is exactly like an existing table,*

create table emp\_global\_seq LIKE emp\_global\_orc stored as sequencefile ;

Hive Illustration : Loading data into Hive tables

* *Create table only if another table of the same name doesn’t exist and an input multiple values in a single column using an array,*

create table if not exists sibling\_data (

name string, age int, country string, siblings array<string> )

row format delimited

fields terminated by ‘ , ‘

collection items terminated by ‘#’

lines terminated by ‘\n’

sorted as textfile ;

* *To create table with multiple inputs of different data type in a single column,*

create table auto\_details(company string, model string, fuel string,

basic\_specs struct<vehicle\_type : string, doors : int, gears : int>,

engine\_specs struct<cc : int, bhp : double>)

row format delimited

fields terminated by ‘ , ‘

collection items terminated by ‘#’ ;

Hive Illustration : Simple Operations on Hive tables

* *To rename an existing table,*

alter table auto\_details rename to auto\_table ;

* *To change the name of any column,*

alter table auto\_details change fuel fuel\_type string ;

* *To add a new column to an existing table,*

alter table auto\_details add columns (milage double) ;

* *To drop columns,(mention columns which need to remain inside the brackets after “replace” keyword)*

alter table auto\_details replace (company string, model string, fuel\_type string) ;

Hive Illustration : Query Operations on Hive tables

* *To create a table inside a desired pre-existing database, without navigating into the database first*

create table if not exists company.empdata (

empid int,

empname string,

salary double,

designation string,

department string,,

salary double,

designation string,

department string,

age int)

row format delimited

fields terminated by ‘ , ‘

lines terminated by ‘\n’

tblproperties(‘skip.header.line.count’ = ‘1’) ;

* *To select all columns and only those rows which satisfy a certain condition,*

select \* from empdata where department = “HR” ;

* *To select all columns and only those rows which satisfy more than one condition,*

select \* from empdata where department =”HR” and salary > 25000 ;

* *To select only desired columns and only those rows which satisfy more than one condition,*

select empname, age from empdata where department =”HR” and salary > 25000 ;

* *To select all columns and sort the rows based on a desired column,*

select \* from empdata order by salary ;

* *To select all columns and sort the rows based on a desired column in descending order,*

select \* from empdata order by salary desc ;

* *To count the total number of rows in the dataset,*

select count(\*) from empdata ;

* *To use ‘groupby’ to count number of rows based in each category of a certain column,*

select department, count(\*) from empdata group by department ;

* *To select all column but only those rows which do not have null value in a desired column,*

select \* from empdata where salary is not null ;

* *To select rows by matching a substring with a desired column value,*

select \* from empdata where designation rlike “Manager” or rlike “manager” or “Lead” ;

* *To find the average of a desired numerical column, grouped by a categorical column,*

select department, avg(salary) from empdata group by department;

Hive Illustration : Querying complex structures

* *To enable join operations in the hive shell,*

SET hive.auto.conveert.join = False;

* *To perform a join operation,*

select emp.empname, emp.salary from emp\_epf pf join empdata emp on (pf.empid = emp.empid) ;

* *To perform a left outer join operation,*

select emp.empname, emp.salary from emp\_epf pf left outer join empdata emp on (pf.empid = emp.empid) ;

* *To perform a right outer join operation,*

select emp.empname, emp.salary from emp\_epf pf right outer join empdata emp on (pf.empid = emp.empid) ;

* *To perform a full outer join operation,*

select emp.empname, emp.salary from emp\_epf pf full outer join empdata emp on (pf.empid = emp.empid) ;

Hive Illustration : Views

* *To create view,*

create view if not exists high\_sal as select \* from empdata where salary > 50000 ;

* *To query data from view,*

select \* from high\_sal ;

* *To see if view is created,*

show tables ;

* *To see the table type, (virtual or Managed),*

describe formatted high\_sal ;

* *To create a table, partitioned by a desired column,*

create table emp\_global\_part(id int, name string, city string, country string)

portioned by (continent string)

row format delimited

fields terminated by ‘ , ‘

stored as textfile ;