Module – 4 (HIVE)

What is Hive?

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive.

Hive primitive data types: tinyint,smallint,int,bigint,string etc.

To switch off safe mode

```
Sudo  –u  hdfs  hdfs  dfsadmin  –safemode  leave
```

To create a data file

```
[cloudera@localhost ~]$ gedit data1;

1|ramu|20

2|krish|19

3|sita|20

4|john|21

5|syed|22
```

Save and Close

To connect to hive

```
[cloudera@localhost ~]$ hive
```

To create database

```
hive> create database db2;
```

(to drop database, drop database db2)

To see the list of databases

```
hive> show databases;
```

To make use of the database

```
hive> use db2;
```

To create table

```
hive> create table emp(id int,name string,age int)row format delimited fields
 terminated by '|' lines terminated by '\n' stored as textfile;
```

To see the list of tables

```
hive> show tables;
```

To see the structure of a table

```
hive> describe emp;
```

To load data into hive table from local system

```
hive> load data local inpath '/home/cloudera/data1' into table emp;
```

To see the content of the table

```
hive> select * from emp;
```

To rename the table

```
alter table emp rename to employee;
```

To check the table names

```
hive> show tables;
```

To add new columns to the table

```
hive> alter table employee add columns(city string,pincode int);
```

To see the structure of a table

```
hive> describe employee;
```

To rename the column in a table

```
hive> alter table employee change city addr string;
```

To see the structure of a table

```
hive> describe employee;
```

To select the rows based on condition

```
hive> select * from employee where age>19;
```

To create a table from another table

```
hive> create table empd as select * from employee;

hive> select * from empd;
```

To limit the number of rows to be displayed

```
hive> select * from employee limit 2;
```

To use built-in functions

```
hive> select upper(name) from employee;

hive> select count(id) from employee;

hive> select substr(name,1,3) from employee;
```

To store the output of analysis to some other table

```
hive> insert overwrite table empd select * from employee where age>20;

hive> select * from empd;
```

To join two tables

```
hive> select * from employee e join empd d on (e.id=d.id);

hive> select * from employee e left outer join empd d on (e.id=d.id);

hive> select * from employee e right outer join empd d on (e.id=d.id);
```

To create a view

```
hive> create view emp_v as select id,name from employee;

hive> select * from emp_v;
```

To store the output of analysis to HDFS file system

```
hive> insert overwrite directory '/user/cloudera/output2'

                    select * from employee where age>20;
```

To store the output of analysis to Local file system

```
hive> insert overwrite local directory '/home/cloudera/output1'
            select * from employee where age>20;
```

To quit from hive

```
hive> quit;
```

To check the output file in local system

```
[cloudera@localhost ~]$ ls output1;
[cloudera@localhost ~]$ cat output1/000000_0
```

To check the output file in Hadoop

```
[cloudera@localhost ~]$ hadoop fs -ls /user/cloudera/
[cloudera@localhost ~]$ hadoop fs -cat /user/cloudera/output1/000000_0
```

(or)

Through Browser

MODULE 4

HIVE

TO SWITCH OFF SAFE MODE

```
$ sudo   –u   hdfs   hdfs   dfsadmin   –safemode   leave
```

EXP 20: DDL Commands

To Create Two Files On Local Filesystem And Copy It To Hdfs Any Folder

```
[cloudera@localhost ~]$ gedit   emp.txt
1001|hari|d1|chennai|1986-12-10
1002|teja|d1|hyd|1987-01-21
1003|ram|d3|delhi|1986-02-11
1004|milind|d4|bang|1988-03-21
1005|jay|d2|bang|1988-03-22
1006|naveen|d4|hyd|1986-04-12
1007|naser|d1|hyd|1989-11-15
1008|rahul|d3|delhi|1990-12-23

[cloudera@localhost ~]$ gedit   d.txt
d1|research|A-block
d2|sales|A-block
d3|testing|B-block
d4|development|C-block

[cloudera@localhost ~]$ hadoop  fs   -put   emp.txt  /user/cloudera/batch3
[cloudera@localhost ~]$ hadoop   fs   -put   d.txt  /user/cloudera/batch3
```

CONNECT TO HIVE

```
[cloudera@localhost ~]$ hive
hive>
```

TO CREATE DATABSE

```
hive> ok;                          (OR)
hive> create database  if  not  exists  test;
```

TO LIST OUT DATABSES

```
hive> show databases;
```

TO DROP DATABSE

```
hive> drop database test;                        (OR)
hive> drop database if exists test;              (OR)
hive> drop database if exists test cascade;
```

NOTE!!!! [if exists] & [if not exists] doesn't show error if database already exists while creating time and database doesn't exists while dropping the same.
Without these options , errors displayed clearly.

TO MAKE USE OF THE DATABASE

```
hive> use    test;
```

Create Table Statement

Create Table is a statement used to create a table in Hive. The syntax and example are as follows:

Syntax

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name
[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment]
[ROW FORMAT row_format]
[STORED AS file_format]
```

TO CREATE TABLE

```
hive> create  table  emp(id int,name string,dept string,place string,dob string)
>comment  'this  is employee  table'
> row format delimited fields  terminated by '|' lines terminated by '\n'
>stored as textfile;
```

(OR) Type IN Single Line

```
hive> create  table  emp(id int,name string,dept string,place string,dob string) comment  'this  is employee  table'  row format delimited fields  terminated by '|' lines terminated by '\n' stored as textfile;
hive> create table department(did string,dname string,block string) comment 'this is department table' row format delimited fields terminated by '|' lines terminated by '\n' stored as textfile;
```

NOTE!!! You can mention just mention
Hive>USE test;
Hive> CREATE TABLE emp (…..) ….
(OR)
Hive> CREATE TABLE test.emp (……) ….

TO SEE THE LIST OF TABLES

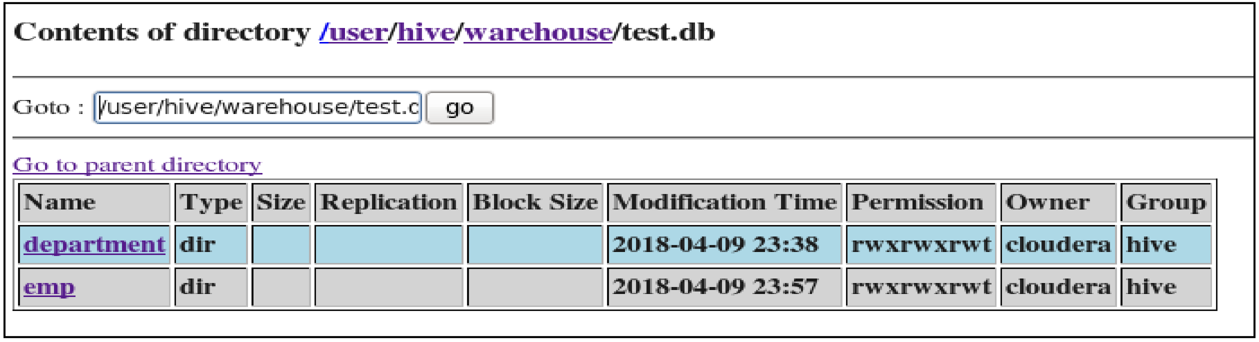
```
hive> show tables;
```

TO SEE THE STRUCTURE OF A TABLE

```
hive> describe  emp;
```

TO SEE THE STRUCTURE & METADATA INFORMATION OF TABLE

```
hive> describe formatted  emp;
hive> show create table emp;
```



Contents of directory /user/hive/warehouse/test.db

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
department	dir				2018-04-09 23:38	rwxrwxrwt	cloudera	hive
emp	dir				2018-04-09 23:57	rwxrwxrwt	cloudera	hive

Alter Table Statement

It is used to alter a table in Hive.

Syntax

The statement takes any of the following syntaxes based on what attributes we wish to modify in a table.

```
ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
ALTER TABLE name CHANGE column_name new_name new_type
ALTER TABLE name REPLACE COLUMNS (col_spec [, col_spec ...])
```

TO RENAME TABLE NAME

```
hive> alter table department rename to d;
hive> show tables;
d
emp
```

TO ADD ONE OR MORE COLUMNS TO THE TABLE

```
hive> alter table d add columns (estb_year int,rating smallint);
hive> describe d;
did     string
dname           string
block   string
estb_year       int
rating smallint
```

TO CHANGE COLUMN NAME OR ITS DATATYPE OR BOTH

```
hive> alter table d change rating rate string;
hive> describe d;
did     string
dname           string
block   string
estb_year       int
rate    string

hive> alter table d change rate rate bigint;
hive> describe d;
did     string
dname           string
block   string
estb_year       int
rate    bigint
```

TO REPLACE COLUMNS

```
hive> alter  table  d  replace  columns (did string,dname string, block string);
hive> describe d;
did     string
dname           string
block   string

hive> alter table d replace columns (block string);
hive> describe d;
block   string

hive> select * from d;
d1
d2
d3
d4
```

//if you do REPLACE again, you will get the columns again I,e replace is not removing columns permanently

```
hive> alter  table  d  replace  columns (did string,dname string, block string);
hive> desc d;
```

```
did     string
dname           string
block   string
```

```
hive> select * from d;
d1      research        A-block
d2      sales   A-block
d3      testing         B-block
d4      development  C-block
d5      hr      A-block
```

TO DROP THE TABLE

```
hive> drop  table  if  exists  d;               (OR)
hive> drop   table   d;
```

EXP 21: Load, Insert of data

Load Data Statement

Generally, after creating a table in SQL, we can insert data using the Insert statement.

But in Hive, we can insert data using the LOAD DATA statement.

While inserting data into Hive, it is better to use LOAD DATA to store bulk records. There are two ways to load data: one is from local file system and second is from Hadoop file system.

Syntax

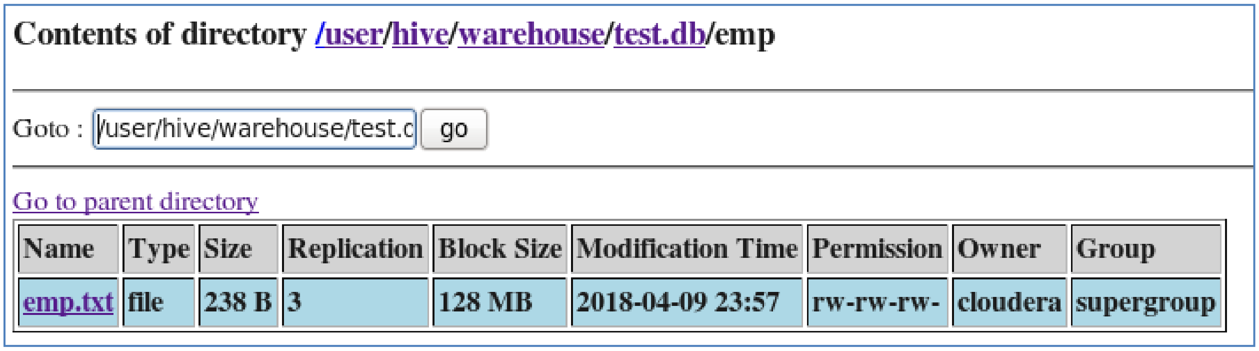
The syntax for load data is as follows:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename
[PARTITION  (partcol1=val1, partcol2=val2 ...)]
```

- LOCAL is identifier to specify the local path. It is optional.
- OVERWRITE is optional to overwrite the data in the table.
- PARTITION is optional used table is created with partitions.

TO LOAD FROM LOCAL FILESYSTEM

```
hive>  load  data  local  inpath  '/home/cloudera/emp.txt'  into  table emp;
hive> select * from emp;
1001    hari    d1      chennai         1986-12-10
1002    teja    d1      hyd     1987-01-21
1003    ram     d3      delhi   1986-02-11
1004    milind d4       bang    1988-03-21
1005    jay     d2      bang    1988-03-22
1006    naveen d4       hyd     1986-04-12
1007    naser   d1      hyd     1989-11-15
1008    rahul   d3      delhi   1990-12-23
```



Contents of directory /user/hive/warehouse/test.db/emp

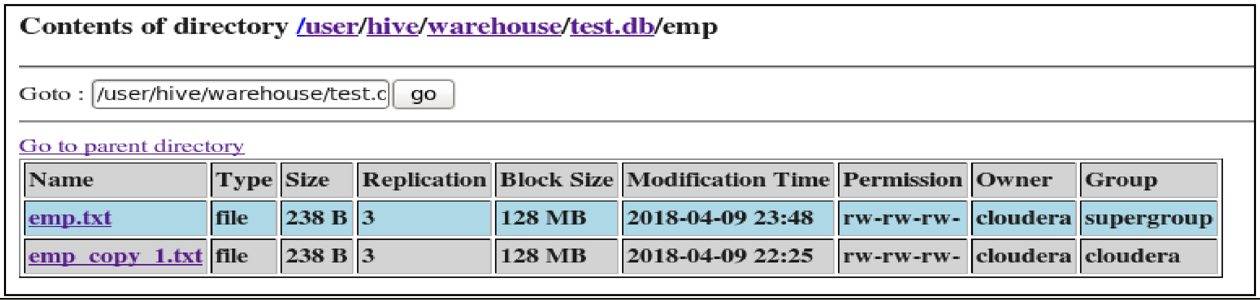
Goto : /user/hive/warehouse/test.c go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
emp.txt	file	238 B	3	128 MB	2018-04-09 23:57	rw-rw-rw-	cloudera	supergroup

TO LOAD FROM HADOOP FILE SYSTEM

```
hive> load data inpath '/user/cloudera/emp.txt' into table emp;
hive> select * from emp;
1001    hari    d1      chennai         1986-12-10
1002    teja    d1      hyd     1987-01-21
1003    ram     d3      delhi   1986-02-11
1004    milind d4       bang    1988-03-21
1005    jay     d2      bang    1988-03-22
1006    naveen d4       hyd     1986-04-12
1007    naser   d1      hyd     1989-11-15
1008    rahul   d3      delhi   1990-12-23
1001    hari    d1      chennai         1986-12-10
1002    teja    d1      hyd     1987-01-21
1003    ram     d3      delhi   1986-02-11
1004    milind d4       bang    1988-03-21
1005    jay     d2      bang    1988-03-22
1006    naveen d4       hyd     1986-04-12
1007    naser   d1      hyd     1989-11-15
1008    rahul   d3      delhi   1990-12-23
```



Contents of directory /user/hive/warehouse/test.db/emp

Goto : /user/hive/warehouse/test.c go

Go to parent directory

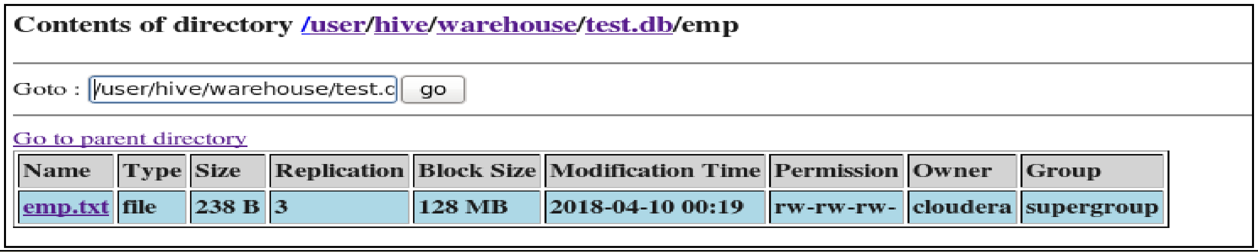
Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
emp.txt	file	238 B	3	128 MB	2018-04-09 23:48	rw-rw-rw-	cloudera	supergroup
emp_copy_1.txt	file	238 B	3	128 MB	2018-04-09 22:25	rw-rw-rw-	cloudera	cloudera

NOTE!!!!Here, Since no overwrite was used; the data got appended to same table. And in Hive/warehouse/test.db two copies of same content got generated.

TO LOAD USING OVERWRITE KEYWORD

```
hive> load data local inpath '/home/cloudera/emp.txt' overwrite into table
emp;
hive> select * from emp;
1001    hari    d1      chennai         NULL
1002    teja    d1      hyd     NULL
```

```
1003    ram     d3      delhi   NULL
1004    milind d4       bang    NULL
1005    jay     d2      bang    NULL
1006    naveen d4       hyd     NULL
1007    naser   d1      hyd     NULL
1008    rahul   d3      delhi   NULL
```



Contents of directory /user/hive/warehouse/test.db/emp

Goto : /user/hive/warehouse/test.c go

Go to parent directory

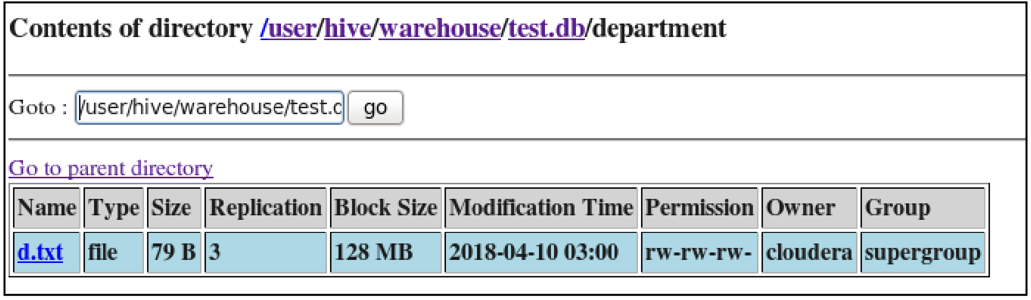
Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
emp.txt	file	238 B	3	128 MB	2018-04-10 00:19	rw-rw-rw-	cloudera	supergroup

NOTE!!! Once the data is loaded to hive table from hadoop filesystem, the file "emp.txt" no more exists in /user/cloudera I,e loading from hadoop filesystem is like cut and paste to hive; whereas its like copy & paste when loaded from local filesystem.

So , if you have loaded from hadoop filesytem once, then you can't load or load with overwrite to hive table from hadoop filesystem again…. Because you will get ERROR: "invalid path as file is cut already from that location."

hive> load data local inpath '/home/cloudera/d.txt' overwrite into table department;
hive> select * from department;

```
d1      research        A-block
d2      sales   A-block
d3      testingB-block
d4      development C-block
```



Contents of directory /user/hive/warehouse/test.db/department

Goto : /user/hive/warehouse/test.c go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
d.txt	file	79 B	3	128 MB	2018-04-10 03:00	rw-rw-rw-	cloudera	supergroup

EXP 22: BUILT_IN RELATIONAL OPERATORS

A=B, A<B, A<=B, A>B, A>=B, A IS NULL, A IS NOT NULL, A LIKE B, A!=B

BUILT_IN ARTHEMATIC OPERATORS

A+B, A-B, A*B, A/B, A%B, A&B, A|B,   A^B, ~A

A&B : bitwise and operation

A|B: bitwise or operation

A^B;bitwise XOR operation

~A: bitwise not operation

BUILT_IN LOGICAL OPERATORS

A AND B, A OR B, NOT A, A || B, A&&B,   !A

Q)Display details of employee whose employee id is greater than and equal to 1003 and doesn't come from hyd.

```
hive> select * from emp where id >=1003 and place !='hyd';                    (OR)
hive> select * from emp where id >=1003 and place not in ('hyd');
1003    ram     d3      delhi   1986-02-11
1004    milindd4        bang    1988-03-21
1005    jay     d2      bang    1988-03-22
1008    rahul   d3      delhi   1990-12-23
```

Q)Display details of department whose id is less than d2 or department name is development.

```
hive> select * from department where did<'d2' or dname='development';
        (OR)
hive> select * from department where did<'d2' or dname like 'development';
        (OR)
hive> select * from department where did<'d2' or dname like 'd%';
d1      research        A-block
d4      development C-block
```

Q)Display details of department whose department name's second letter is 'e'.

```
hive> select * from department where did<'d2' or dname like '_e%';
d1      research        A-block
d3      testingB-block
d4      development C-block
```

Q)Display total no:of employees ,minimum of their employee no, avg of their employee no,max of their employee, sum of their employee from employee dataset.

```
hive> select count(*),min(id),avg(id),max(id),sum(id) from emp;
8       1001    1004.5          1008    8036
```

Q) Display department id, count of employees in each department

```
hive> select count(*),dept from emp group by dept;
3       d1
1       d2
2       d3
2       d4
```

Q) Display department id,count of employees in each department and display rows those have count more than 2.

```
hive> select count(*),dept from emp group by dept having count(*)>2;
3       d1
```

Q) Display department id,count of employees in each department in descending order of count.

```
hive> select count(*) as c,dept from emp group by dept order by c desc;
3       d1
2       d3
2       d4
1       d2
```

Q) Display department id,count of employees in each department in descending order of count and display only first two rows.

```
hive> select count(*) as c,dept from emp group by dept order by c desc limit 2;
3       d1
2       d3
```

EXP 23: TO JOIN TWO TABLES

```
hive> select  *  from emp e join department  d on (e.dept=d.did);
001     hari    d1      chennai         1986-12-10      d1      research        A-block
1002    teja    d1      hyd     1987-01-21      d1      research        A-block
1007    naser   d1      hyd     1989-11-15      d1      research        A-block
1005    jay     d2      bang    1988-03-22      d2      sales   A-block
1003    ram     d3      delhi   1986-02-11      d3      testing B-block
1008    rahul   d3      delhi   1990-12-23      d3      testing B-block
1004    milind d4       bang    1988-03-21      d4      development     C-block
1006    naveen d4       hyd     1986-04-12      d4      development     C-block
hive> select  *  from emp e left outer join department  d on (e.dept=d.did);
1001    hari    d1      chennai         1986-12-10      d1      research        A-block
1002    teja    d1      hyd     1987-01-21      d1      research        A-block
1007    naser   d1      hyd     1989-11-15      d1      research        A-block
1005    jay     d2      bang    1988-03-22      d2      sales   A-block
1003    ram     d3      delhi   1986-02-11      d3      testing B-block
1008    rahul   d3      delhi   1990-12-23      d3      testing B-block
1004    milind d4       bang    1988-03-21      d4      development     C-block
1006    naveen d4       hyd     1986-04-12      d4      development     C-block
1009    jay     d6      hyd     1988-07-19      null    null            null
```

NOTE!!if d6 department not there ,then no matching on right side table values

```
hive> select  *  from emp e right outer  join department  d on (e.dept=d.did);
1001    hari    d1      chennai         1986-12-10      d1      research        A-block
1002    teja    d1      hyd     1987-01-21      d1      research        A-block
1007    naser   d1      hyd     1989-11-15      d1      research        A-block
1005    jay     d2      bang    1988-03-22      d2      sales   A-block
1003    ram     d3      delhi   1986-02-11      d3      testingB-block
1008    rahul   d3      delhi   1990-12-23      d3      testingB-block
1004    milindd4        bang    1988-03-21      d4      development C-block
1006    naveen          d4      hyd     1986-04-12      d4      development C-block
NULL NULL NULL NULL NULL d5     hr      A-block
```

EXP 24: TO CREATE A VIEW

```
hive> create  view  emp_v  as select id,name from emp where id>1003;
hive> select  * from emp_v;
1004    milind
1005    jay
1006    naveen
1007    naser
1008    rahul
```

TO DROP THE VIEW

```
hive>drop view emp_v;
```

To Use Built-In Functions

```
hive> select  upper(name)  from emp;
HARI
TEJA
RAM
MILIND
JAY
NAVEEN
NASER
RAHUL
```

```
hive> select  count(id)  from  emp;
8
```

```
hive> select  substr(name,1,3)  from  emp;
har
tej
ram
mil
jay
nav
nas
```

```
rah
```

```
hive> select  substr(name,2)  from  emp;
ari
eja
am
ilind
ay
aveen
aser
ahul
```

```
hive> select  substr(name,3,2)  from  emp;
ri
ja
m
li
y
ve
se
hu
```

Syntax: substr(string,starting index,no of character)

Note: if no of characters not mentioned then it returns from the start position to the end of the string

TO CREATE A TABLE FROM ANOTHER TABLE

```
hive> create table abc  as select  * from emp;
hive> select  *  from  abc;
```

TO STORE THE OUTPUT OF ANALYSIS TO SOME OTHER TABLE

```
hive> insert overwrite  table  abc  select * from emp where id>1003;
hive> select * from abc;
004     Milind D4      Bang    1988-03-21
1005    Jay    D2      Bang    1988-03-22
1006    Naveen         D4      Hyd     1986-04-12
1007    Naser  D1      Hyd     1989-11-15
1008    Rahul  D3      Delhi   1990-12-23
```

Note:schema should match(No.of column should match)

To Store The Output Of Analysis To Hdfs File System

```
Hive> Insert  Overwrite  Directory  '/User/Cloudera/Output1'  Select  *  From  Emp Where Id>1003;
```

Note!! *Where Output1 Is A New Directory In /User/Cloudera, Which Will Get Created Automatically*

To Check The Output File In Hadoop File System

```
[Cloudera@Localhost ~]$ Hadoop   Fs   -Ls   /User/Cloudera/Output1
 [Cloudera@Localhost ~]$ Hadoop   Fs   -Cat   /User/Cloudera/Output1/00000_0
```

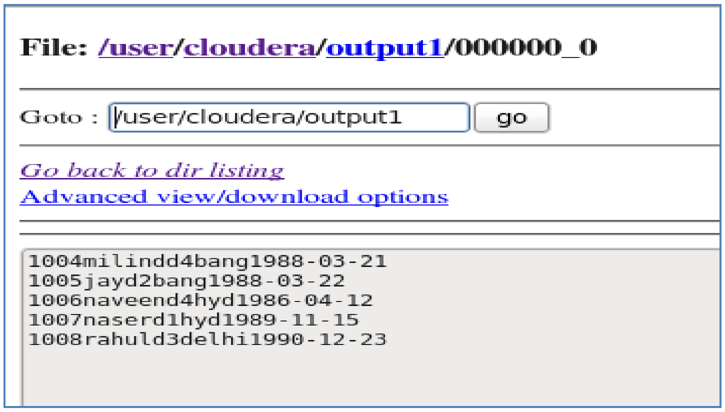
(Or)

```
Hive> Dfs   -Ls  /User/Cloudera/Output1;
-Rw-R--R--        3   Cloudera   Supergroup                 149    2018-04-17    03:32
/User/Cloudera/Output1/000000_0
```

```
Hive> Dfs   -Cat   /User/Cloudera/Output1/0*;
1004milindd4bang1988-03-21
1005jayd2bang1988-03-22
1006naveend4hyd1986-04-12
1007naserd1hyd1989-11-15
1008rahuld3delhi1990-12-23
```

(Or)

Use Browser Of Your Vm To Go To The Location And Verify The File



File: /user/cloudera/output1/000000_0

Goto : /user/cloudera/output1 go

Go back to dir listing
Advanced view/download options

```
1004milindd4bang1988-03-21
1005jayd2bang1988-03-22
1006naveend4hyd1986-04-12
1007naserd1hyd1989-11-15
1008rahuld3delhi1990-12-23
```

TO STORE THE OUTPUT OF ANALYSIS TO LOCAL FILE SYSTEM

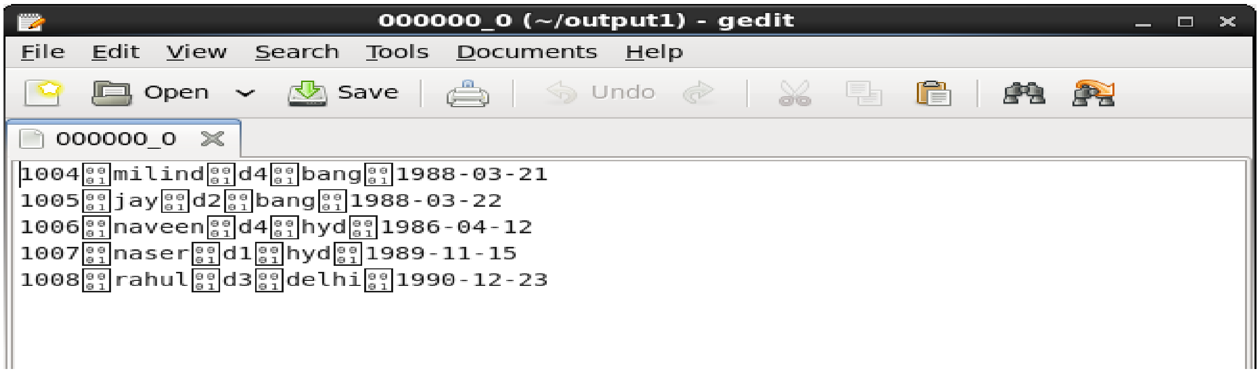
```
hive> insert  overwrite  local  directory  '/home/cloudera/output1'  select * from
employee where id>1003;
```

TO CHECK THE OUTPUT FILE IN LOCAL SYSTEM

○
```
[cloudera@localhost ~]$ ls
[cloudera@localhost ~]$cd   output1
[cloudera@localhost ~]$ cat   000000_0            (OR)  $gedit   000000_0
```

(OR)

○
```
[cloudera@localhost ~]$ ls  output1
 [cloudera@localhost ~]$ cat  output1/000000_0
```



```
1004milindd4bang1988-03-21
1005jayd2bang1988-03-22
1006naveend4hyd1986-04-12
1007naserd1hyd1989-11-15
1008rahuld3delhi1990-12-23
```

TO QUIT FROM HIVE

hive> quit;