HBASE

🡪 su hbase : This takes you to the root of hbase in order if there is a permission issue .

🡪 hbase shell: this command is used to enter into the command line of hbase shell.

* Creating tables in Hbase (key:value based, it doesnt require any dataType int the column)

🡪 create ‘employee’,’personaldata,’professionaldata’

This above statement create employee table with two column family

🡪 put ‘employee’,’r1’,’personaldata:name’,’Aditya’

🡪put’emplyee’,’r1’,’personaldata:city’,’california’

🡪put 'employee','r1','professionaldata:salary','10000'

🡪put 'employee','r1','professionaldata:designation','manager'

here now r1(unique identifier for entire row) is the row key and we have to insert the first column name under first column family and the data associated with that column name in the r1 in the first column family

so here (name) is the column name under the column family and (Aditya) is the data in that column.

After every input you have to enter the line and use put command again.

The benefit of hbase is there is no limit for anything you can add N number of columns under the personaldata or professional data using the put statements.

🡪 scan 'employee'

output:

ROW COLUMN+CELL

r1 column=personaldata:city, timestamp=1465802982533, value=california

r1 column=personaldata:name, timestamp=1465802712609, value=Aditya

r1 column=professionaldata:designation, timestamp=1465807061527, value=man

ager

r1 column=professionaldata:salary, timestamp=1465808559028, value=10000

🡪 get ‘employee’,’r1’,{COLUMN => ‘professionaldata:designation’} 🡪 COLUMN is in caps

COLUMN CELL

professionaldata:designa timestamp=1465807061527, value=manager

tion

🡪 get 'employee' ,'r1',{COLUMN => 'personaldata:city'}

COLUMN CELL

personaldata:city timestamp=1465802982533, value=california

get the value from the timestamp so in this case we can also get the old value by giving the timestamp constraint

🡪 get ‘employee’ ,’r1’,{COLUMN => ‘professionaldata:salary’,TIMESTAMP = 1465808559028}

output:

COLUMN CELL

professionaldata:salary timestamp=1465808559028, value=10000

This code intergrates HIVE with HBASE 🡪 Here the table name is employee which is made in hbase and whose data is being imported to the hiveemp table made in hive so we can perform all the necessary and required operations on hive using HiveQL commands

create EXTERNAL table hiveemp(rkey string , name string , city string , design string, salary string)

STORED BY ‘org.apache.hadoop.hive.hbase.HBaseStorageHandler’

WITH SERDEPROPERTIES ( ‘hbase.columns.mapping’=’:key,personaldata:name,personaldata:city,professionaldata: design,professionaldata:salary’)

TBLPROPERTIES (‘hbase.table.name’=’employee’);

Select \* from hiveemp;

Output:

r1 Aditya

Now we should display the column name of the columns in order to have a clear view of the columns.

set hive.cli.print.header=true; 🡪 This will add header

output :

select \* from hivehbase;

hivehbase.rkey hivehbase.name

r1 Aditya

so now we have the column name which one belongs to which part.

Using hive and HBASE with cloudeara .

In hive query editor under Settings we have to add :

Key 🡪 Hbase.zookeper.quorum

Value🡪localhost

create EXTERNAL table hbase\_too\_cool(rkey string , name string , city string , design string, salary string)

STORED BY ‘org.apache.hadoop.hive.hbase.HBaseStorageHandler’

WITH SERDEPROPERTIES ( ‘hbase.columns.mapping’=’:key,personaldata:name,personaldata:city,professionaldata: design,professionaldata:salary’)

TBLPROPERTIES (‘hbase.table.name’=’employee’);

This will create a table under hbase : then we can use ‘put’ command to store the data under hbase table