Step 1:

Download the data files from

<https://github.com/imrankhan88/Dataengineering_bootcamp_docs/tree/main/data/hive>

and copy all downloaded files to home dir.

Use below commands to download

Please run these commands from ubuntu or sandbox control. These commands will download the data files into your home dir.

| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/customer.txt |
| --- |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/file1 |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/file11 |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/orders.txt |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/real\_state.csv |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/realstatewh.csv |
| wget https://github.com/imrankhan88/Dataengineering\_bootcamp\_docs/blob/main/data/hive/salesman.txt |

Create a blank text file on your machine with name as **customer.txt** in the current working directory(i.e home directory).

Copy below table data into this customer.txt.

| 1,rahul,2010-01-01,2010-01-01 01:01:01  2,rakesh,2010-01-02,2010-01-02 01:01:01  ,rahul,,2010-01-01 01:01:01  4,rakesh,2010-01-02,2010-01-02 01:01:01 |
| --- |

Step 2:

Create **/data/test/text** dir on hdfs-(Run below command on terminal).

| hadoop fs -mkdir -p **/data/test/text** |
| --- |

Put the text file on HDFS in the create directory-(Run below command on terminal).

| hadoop fs -put **customer.txt /data/test/text** |
| --- |

Step3:

Open hive console-(Run below command on terminal).

| hive |
| --- |

(You will get Hive prompt- **‘**hive>’)

**Create new database in hive**

| create database xyz; |
| --- |

**Switch Database in hive**

| use xyz; |
| --- |

Step4:

**Create an external Table over text file**-(Run below command on hive console).

| CREATE external TABLE customer(id int, name string, dob date, time1 timestamp) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '**/data/test/text**'; |
| --- |

Step5: Check table data-(Run below command on hive console).

| select \* from customer; |
| --- |

Expected output

| 1 rahul 2010-01-01 2010-01-01 01:01:01  2 rakesh 2010-01-02 2010-01-02 01:01:01  **NULL** rahul **NULL** 2010-01-01 01:01:01  4 rakesh 2010-01-02 2010-01-02 01:01:01 |
| --- |

**Other useful queries:-**

| show databases; |
| --- |
| show tables; |
| describe customer; |

**Create managed Table**-(Run below command on hive console).

| create table emp(empId int, empName String, doj date); |
| --- |

**Insert data:**-(Run below create table command on hive console).

| insert into emp values(1,'vishal','2009-01-01'); |
| --- |

*(Note:-MapReduce job will run for data insertion).*

**Read data from one table and insert into another**

1. Create a new table name emp1 and insert data into it from emp.-(Run below commands on hive console).

| create table emp1(empId int, empName String, doj date);  INSERT OVERWRITE TABLE emp1 select \* from emp;  INSERT INTO TABLE emp1 select \* from emp; |
| --- |

**Try these commands as well**

| select \* from emp where empId=1;  select count(\*) from customer; |
| --- |

**Count queries:-**

| Select count (DISTINCT name) from customer;  Select sum(id) from customer group by dob; |
| --- |

**Join:-**

| Select a.empname,b.name from emp a **join** customer b on a.empname=b.name;  Select a.empname,b.name from emp a **left outer join** customer b on a.empname=b.name;  Select a.empname,b.name from emp a **right outer join** customer b on a.empname=b.name;  Select a.empname,b.name from emp a **outer join** customer b on a.empname=b.name; |
| --- |

**Drop a particular table:-**

| Drop Table emp2; |
| --- |

**Create dynamic partitioned table and insert data into it from Non-partitioned table:-**

| CREATE TABLE **partition\_date**(column1 string) **partitioned by** (day string, event string);  CREATE TABLE non\_partitioned\_date1(column1 string, day string,event string);  insert into non\_partitioned\_date1 values('abc', '2000-01-01','e1');  insert into non\_partitioned\_date1 values('abc', '2000-01-02','e1');  insert into non\_partitioned\_date1 values('abc', '1999-12-20','e1');  insert into non\_partitioned\_date1 values('abc', '2000-01-01','e1');  set hive.exec.dynamic.partition.mode=nonstrict;  insert overwrite table partition\_date partition(day,b) select \* from non\_partitioned\_date1; |
| --- |

**Drop multiple partitions from table which have multiple partitions columns**

| **//drop single partition in single command**  **ALTER TABLE partition\_date DROP PARTITION(day =** 2000-01-01**,** event **= 'e1');**  **// drop multiple partitions in single commad**  **ALTER TABLE partition\_date DROP PARTITION (day =** '2000-01-02'**,** event **= 'e1) , PARTITION (day =** '1999-12-20'**,** event **= 'e1);** |
| --- |

**Create Buckets in hive**

| create table input\_table (Street string,  City string,  Zip string,  State string,  Beds string,  Baths string,  Sq\_feet int,  flat\_type string,  Price int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;  load data local inpath 'realstatewh.csv' into table input\_table;  SET hive.enforce.bucketing = true;  set hive.exec.dynamic.partition.mode=nonstrict;  create table bucket\_table(Street string,  Zip string,  State string,  Beds string,  Baths string,  Sq\_feet int,  flat\_type string,  Price int) partitioned by(**city** string) clustered by (**street**) into 4 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';  insert into table bucket\_table partition(city) select street,zip,state,beds,baths,sq\_feet,flat\_type,price,**city** from input\_table; |
| --- |

Note:- In above query we are partitioning bucket\_table on column (city), So it should be the last column in source select statement. I have marked this RED in the above query.

show create table input\_table;

describe formatted input\_table;

Show partitions bucket\_table;

**Ordering commands**

| create table employee(id bigint, name string,age int, salary bigint) partitioned by (department string);  insert into table employee partition(department='HR') values(1,'aarti',28,55000),(2,'shakshi',22,60000),(3,'mahesh',25,25000);  insert into table employee partition(department='BIGDATA') values(10001,'rajesh',29,50000),(10002,'rahul',23,250000),(10003,'dinesh',35,70000);  select \* from employee order by id; |
| --- |

**Convert text table to parquet table**

| create table employee\_parquet(id bigint, name string,age int, salary bigint) STORED AS PARQUET; |
| --- |

| insert intoemployee\_parquet select id,name,age,salary from employee; |
| --- |

| drop table employee; |
| --- |

**Quit the hive terminal:-**

| quit; |
| --- |

**Static partitioning example:-**

| hadoop fs -mkdir -p /data/test/static  hadoop fs -mkdir -p /data/test/static/year=2010  hadoop fs -mkdir -p /data/test/static/year=2011  hadoop fs -put file1 /data/test/static/year=2010  hadoop fs -put file11 /data/test/static/year=2011  Open hive console if it is not already opened - run command  hive  ……..  …….  hive> Now execute all below commands  set hive.mapred.mode = strict;  create external table userdata(userId string, userName String) partitioned by (year string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/data/test/static' ;    **ALTER TABLE userdata ADD PARTITION (year = '2010') LOCATION '/data/test/static/year=2010';**  **ALTER TABLE userdata ADD PARTITION (year = '2011') LOCATION '/data/test/static/year=2011';**    hive> select \* from userdata;  10 rahul 2010  11 deepak 2011  select \* from userdata where year=2010;  OK  10 rahul 2010 |
| --- |

**Show table partitions**

| **show partitions userdata;** |
| --- |