HIVE - LABS

**Step 1.** Connect to hive by typing “hive” at unix prompt and type “show tables;” to get the list tables

in hive catalog.

**Step 2.** Create a hive table as follows:

hive> create table session\_studentname (session\_id string,

user\_id string, url string, time\_spent int)

row format delimited fields terminated by ",";

**Step 3.** Make sure table is created and describe the table using desc on the table.

**Step 4.** Check if your co-students have created any of their tables.

**Step 5.** Create a datafile with 3 rows in unix as session\_studentname.txt in $HOME directory. The rows should have the respective column/data of the above table.

**Step 6.** Now, load your data into your Hive table.

load data local inpath 'path to your file' into table session\_studentname;

**Step 7.** Once it is loaded, browse the data in the table using simple select statements.

**Step 8.** Let’s do some join of tables. It is very similar to RDBMS world.

You can create another table as session\_studentname2 as select from your previous table.

i.e. create table session\_studentname2 as select \* from session\_studentname;

Now, you can join these two tables just to see how it works. You could skip this step if you are ok conceptually.

**Step 9.** Can you check in hdfs, where is your table stored in terms of directory structure? Do you see the directory with the same name as your table?

**Step 10.** Now, let’s alter your table and see what happens in the hdfs directory.

Use alter table TABLENAME to NEWTABLENAME;

Once you perform this task, please check what happens in the directory structure. Do you see your old directory anymore?

**Step 11.** We can see the execution path of a hive query before actually executing it using explain QUERY.

Run explain “YOUR CHOICE OF QUERY”

**Step 12.** Let’s create a view on your above table of your choice using standard sql syntax.

**Step 13.** Now it is more fun time. Let’s create an external table and that too using partition.

CREATE EXTERNAL TABLE session\_info\_studentname (

session\_id STRING, user\_id STRING, visit\_time BIGINT,

url STRING,

time\_spent INT

)

PARTITIONED BY (visit\_dt STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE

LOCATION 'hdfs path ';

Describe the table to see the # of columns. Do you see one additional column than you created?

Create a datafile with the above columns and put the file in hdfs directory of your choice

Let’s load some data into your session\_info\_studentname table as follows:

load data inpath 'your file in hdfs' overwrite into table session\_info\_studentname partition(visit\_dt='2012-03-01');

(set hive.exec.dynamic.partition.mode=nonstrict; set hive.exec.dynamic.partition=true;)