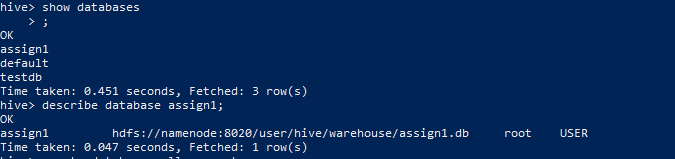
**Assignment 1**

- create a database named assign1



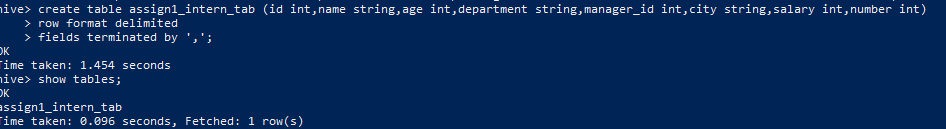
- What is the database path on HDFS?



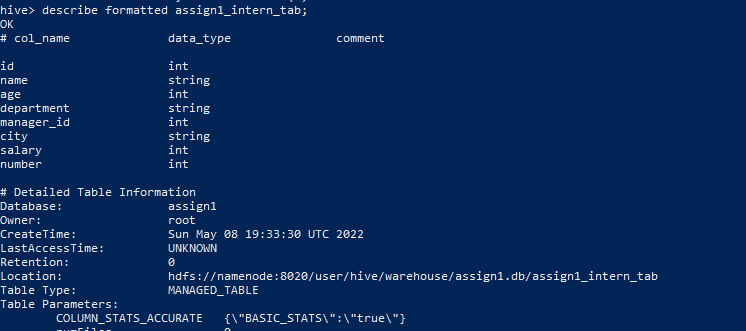
- create a database name assign1\_loc and set its location to /hp\_db/[db\_name]



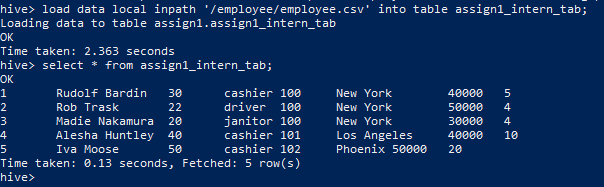
- create a hive managed table assign1\_intern\_tab inside the assign1 database with the right data types to host the data file employees



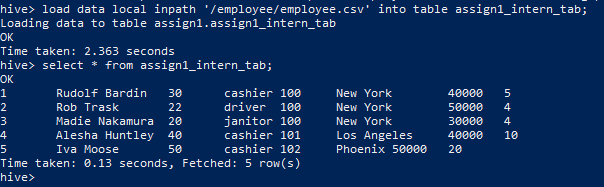
- What is the table path in HDFS?



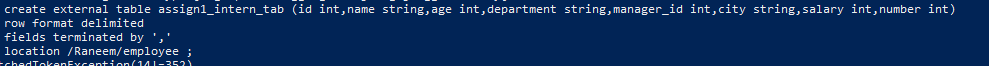
- load the data from the local file system into the table using two different commands



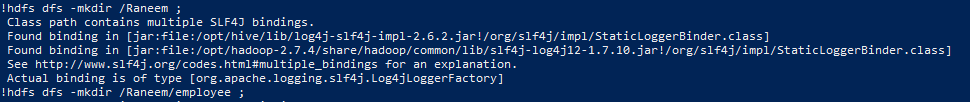
- select 10 records from the table as a sample to ensure the data was correctly loaded



- Create external table assign1\_intern\_tab inside the assign1\_loc database



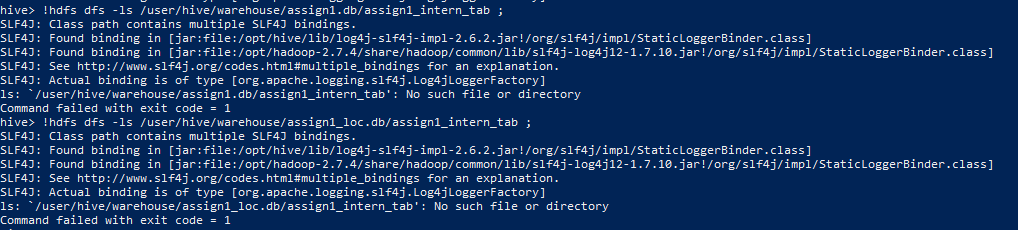
- What is the table path in HDFS?



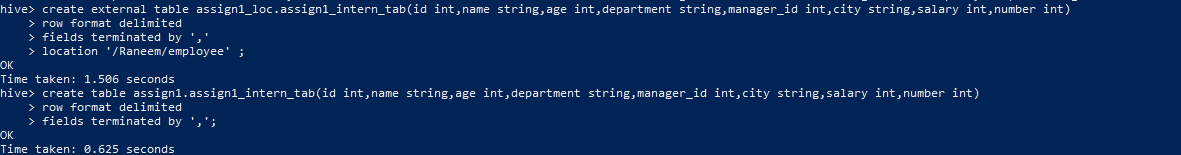
- move the data from local filesystem to the directory created in step 7

## 

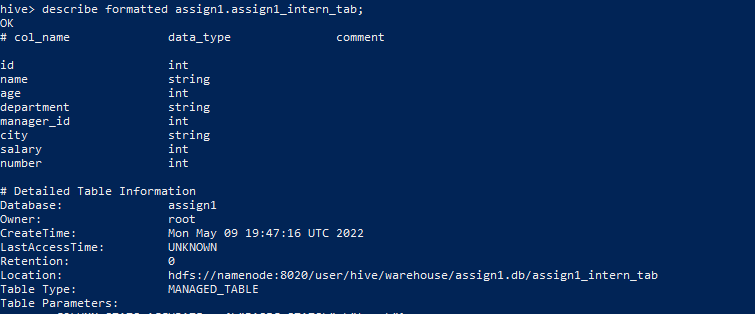
- drop both tables, is the data present after deletion or not?

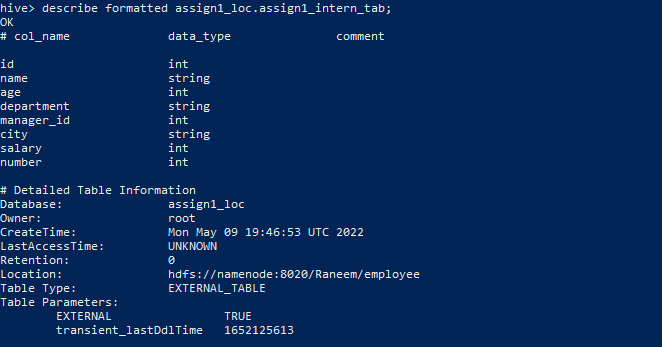


- recreate both tables

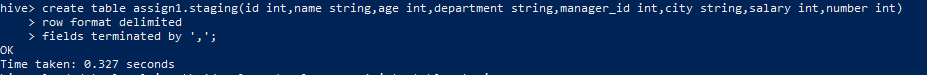


- list both table directories

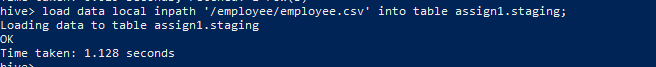




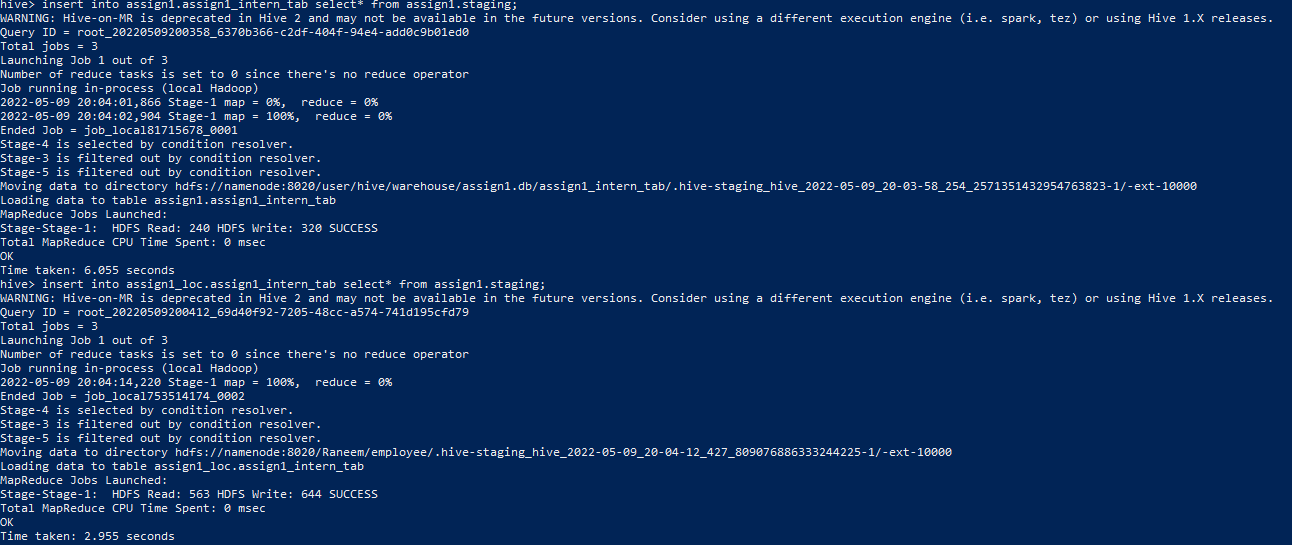
- create internal table 'staging' inside the assign1 database



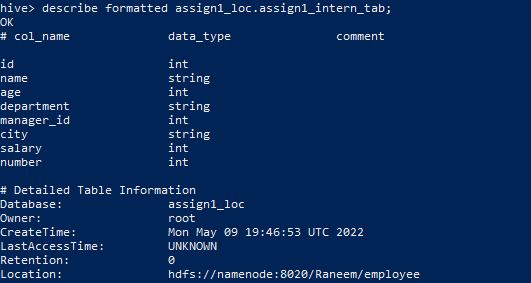
- load the staging table with the data from file employees

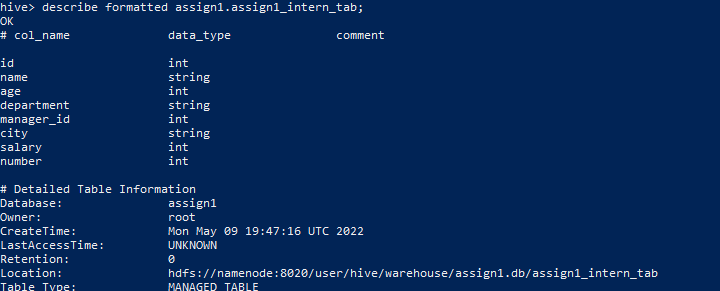


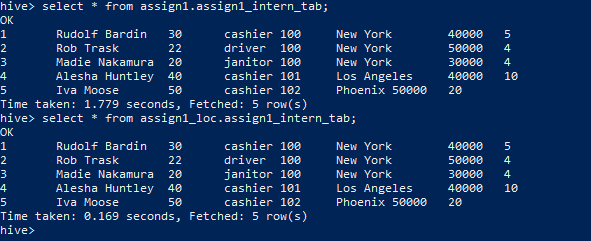
- load tables assign1\_intern\_tab and assign1\_extern\_tab from the staging table using INSERT SELECT statement



- List both directory tables and check if there is data or not



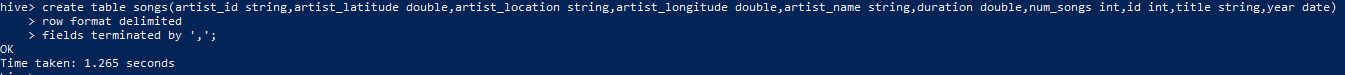




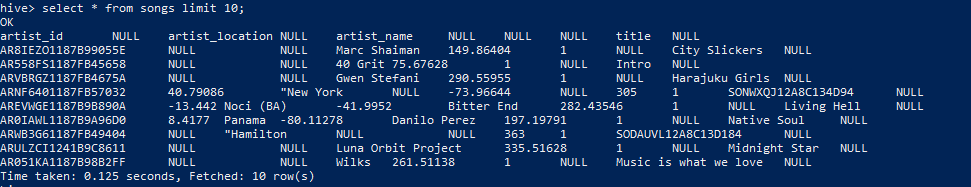
- count the lines inside the file songs



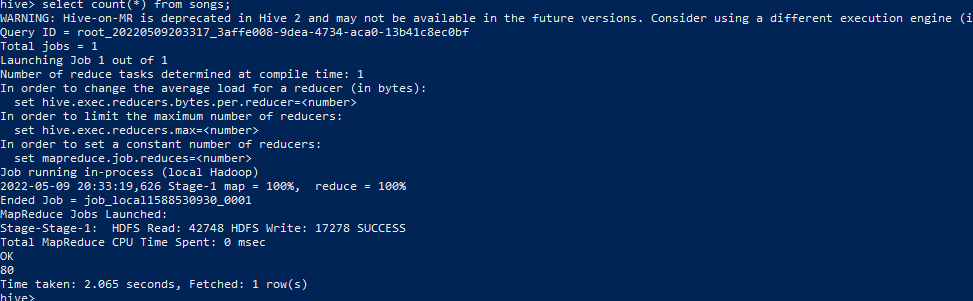
- create a table with the right types to host the data in file.



- select 10 records from the table to ensure it's loaded correctly



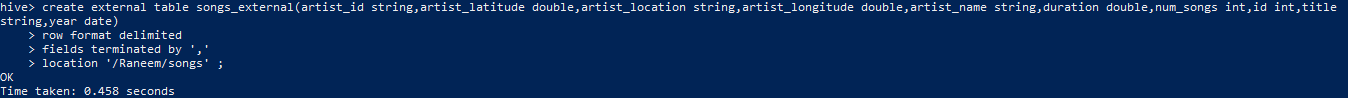
- count the number of records

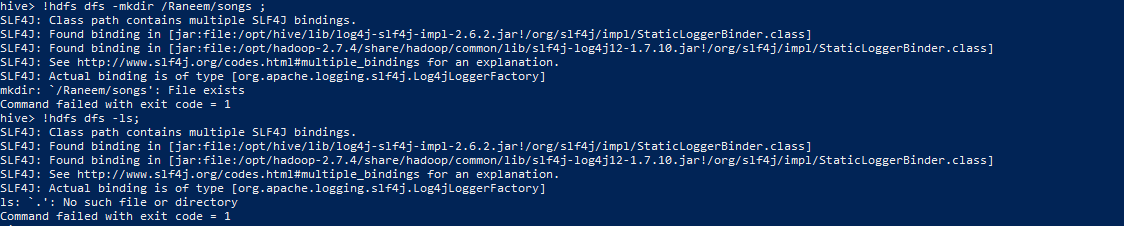


- is the hive count similar to the file count? is the data quality ok? If there is an issue, show how to resolve it

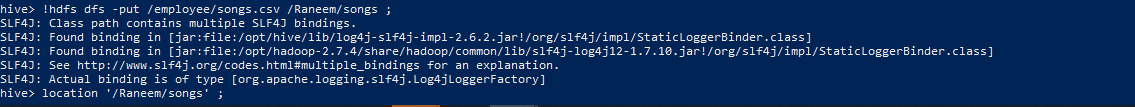
Yes similar, there is a lot of nulls in the data

- create external table ...... to host





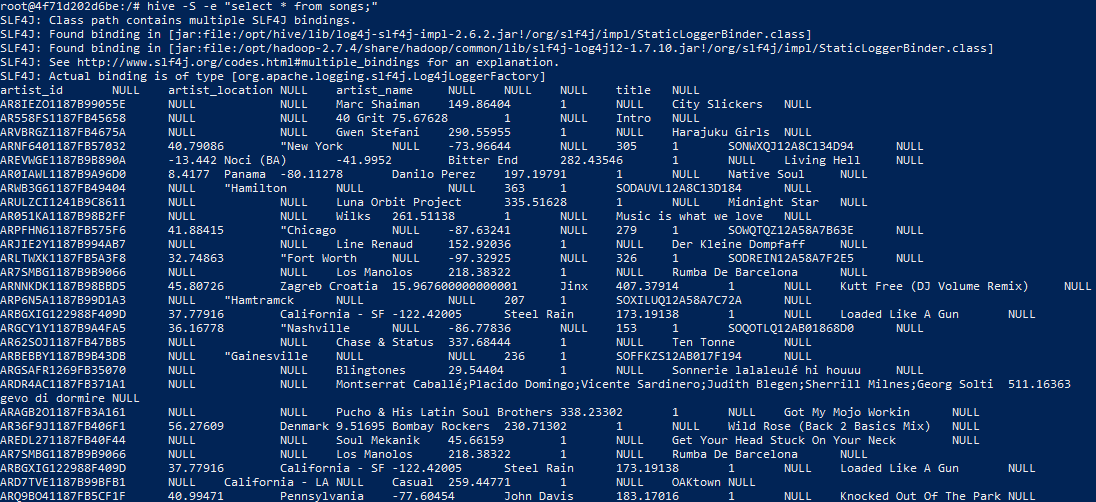
- load the table using put command



- is the data readable through the table? Why?

Not readable

- select [logic] from table [] through shell without accessing hive or beeline



- create a hive script that drop table if exists, creates it and load data with data.

- execute it from shell without accessing hive CLI /beeline

Hive -f script.hql

- What is a hive Temp table? how can you create it? why would someone use a temp table?

CREATE TEMPORARY TABLE statement we can create a temporary table in Hive which is used to store the data temporarily within an active session and the temporary tables get automatically removed when the active session end.

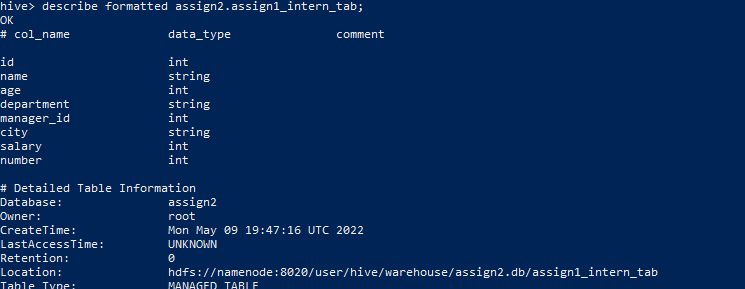
- move the table assign1\_intern\_tab from one database to another

use assign1;

alter table assign1\_intern\_tab rename to assign2.assign1\_intern\_tab

- check the table directory and list its components

describe formatted assign2.assign1\_intern\_tab;



**Assignment 2**

- Create a database named assign2

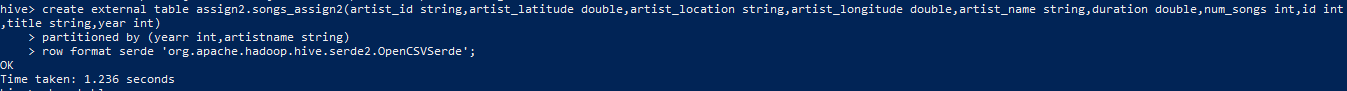
create database assign2;

- Create table for songs table partitioned by artist and year. ensure the right data types are selected and the right SERDEPROPERTIES are used

create external table assign2.songs\_assign2(artist\_id string,artist\_latitude double,artist\_location string,artist\_longitude double,artist\_name string,duration double,num\_songs int,id int,title string,year int)

partitioned by (yearr int,artistname string)

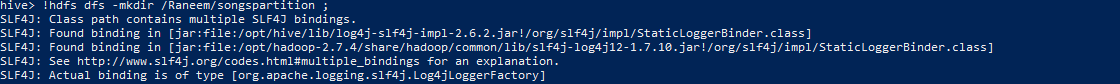
row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde';



- Load data into table HDFS directory using put command

!hdfs dfs -mkdir /Raneem/songspartition ;

!hdfs dfs -put /employee/songs.csv /Raneem/songspartition ;



- Run a SELECT check on the table, is there any data found? why?

No data found because we still did not add location to external table yet



- Add static partition using Alter and set partitions location in a separate directory from that of the table

!hdfs dfs -mkdir /Raneem/partition1 ;

alter table assign2.songs\_assign2 add partition(yearr = 2007,artistname = 'Raneem') location '/Raneem/partition1';



- Load data to the created partitions

!hdfs dfs -put /employee/songs.csv /Raneem/partition1 ;

- List the partition directories to check for presence of files

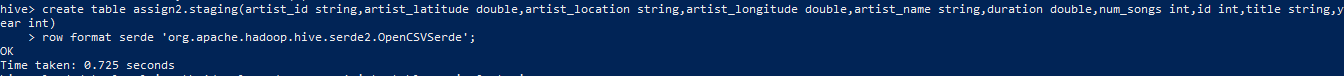
show partitions assign2.songs\_assign2;



- Create a staging table to host songs data

create table assign2.staging(artist\_id string,artist\_latitude double,artist\_location string,artist\_longitude double,artist\_name string,duration double,num\_songs int,id int,title string,year int)

row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde';



- Load the data from the staging table into songs table partitions dynamically

load data local inpath '/employee/songs.csv' into table assign2.staging;

set hive.exec.dynamic.partition=true;

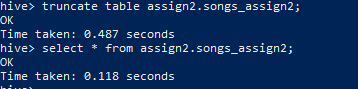
set hive.exec.dynamic.partition.mode=nonstrict;

from assign2.staging insert overwrite table assign2.songs\_assign2 partition(yearr=2008,artistname='Mark')select \* where year=2008 and artist\_name='Marc Shaiman';

- Truncate songs table and ensures no data in the table

truncate table assign2.songs\_assign2;

select \* from assign2.songs\_assign2;



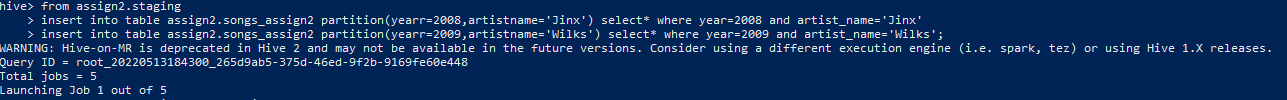
Truncate table songs;

- Use multi inserts to reload the data into the table fully dynamically

from assign2.staging

insert into table assign2.songs\_assign2 partition(yearr=2008,artistname='Jinx') select\* where year=2008 and artist\_name='Jinx'

insert into table assign2.songs\_assign2 partition(yearr=2009,artistname='Wilks') select\* where year=2009 and artist\_name='Wilks';



- Truncate



truncate table assign2.songs\_assign2;

- Use multi inserts to reload the data statically over year and dynamically by artist

- Use CREATE TABLE LIKE statement to create a table with a schema similar to the staging table. The new table should be able to read Avro files

CREATE TABLE like\_staging LIKE assign2.staging

stored as avro;

- Use CREATE TABLE LIKE statement to create a table with a schema similar to the staging table. The new table should be able to read Parquet files

CREATE TABLE like\_staging\_parquet LIKE assign2.staging

stored as Parquet;

- use the avro-tools getschema [avro\_file\_name] command to get the avro schema of the file.

-

**Assignment 3**

File to be used events.csv

1. Create a table with the right data types and SERDEPROPERTIES to host the data from the events.csv files

create table events(artist string,auth string,firstName string,gender string,itemInSession int,lastName string,length double,level string,location string,method string,page string,registration double,sessionId int,song string,status int,ts int,userAgent string,userId int)

row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde';

1. Load the file from local filesystem to the hive table using LOAD statement

load data local inpath '/employee/events.csv' into table events;

1. Select the user, session, first song and last song played per session

select userid,firstname,lastname,sessionid,first\_Value(song)over(partition by sessionid order by sessionid)

as first\_song

,last\_Value(song)over(partition by sessionid order by sessionid

rows between unbounded preceding and unbounded following) as last\_song

from events;

1. Rank users according to the number of distinct songs they played. If two users shared the same counts, they should have the same rank

Select x.userid,rank () over(order by x.c desc) as rank from(

select userid,count(distinct(song)) as c from events group by userid) x

order by rank desc ;

1. Rank users according to the number of distinct songs they played. If two users shared the same counts, each user should have his/her own number. Note that records indicating s a played song are those with column ‘page’ equals to NextPage

Select x.userid,row\_number () over(order by x.c desc) as rank from(

select userid,count(distinct(song)) as c from events where page=’NextSong’ group by userid) x

order by rank desc ;

1. In the same table, show the count of songs played per location and artists, per location only and the total count

select count(song),location,artist from events group by location,artist grouping sets((location,artist),location,());

1. In the same table, show the count of songs played per location and artists, per location only , per artist only and the total count
2. For each song played by a user, get the previous song and next song played. Get the count of each path, and fetch the top 10 paths found

Select lead(song)over(partition by userid order by ts),lag(song)over(partition by userid order by ts) from events;

1. Select userid, song ordered by userid, song, ts. The query should be written to run on a single reducer

SELECT userid,song,ts from events Distribute By userid sort by userid,song,ts;

1. Select userid, song ordered by userid, song, ts. The query should be written to run on a multiple reducers

SELECT userid,song from events order by userid,song,ts CLUSTER BY userid;