**Hbase Assignment1**

**Task 3:**

Create a table album in hive with the following data

hive> create database hivesavita;

hive> create table album(rowkey string,singer string,song string,duration string,musicdirector string)row format delimited fields terminate

d by ',';

created (notepad) singer.csv file for loading data

hive> load data local inpath 'singer.csv' into table album;

A white background with black text

Description automatically generated

hive> desc album;

A close-up of a computer code

Description automatically generated

hive> desc formatted album;

A screenshot of a computer

Description automatically generated

A black text on a white background

Description automatically generated

hive> select \* from album;



Created table in hive

1. Create a table in Hbase (using hive-hbase integration) with proper row key design efficiently to load the data from hive to hbase.

hive> CREATE TABLE AlbumToHBase(rowkey string,Singer string,Song string,Duration string,MusicDirector string)STORED BY 'org.apache.hadoop.h

ive.hbase.HBaseStorageHandler'WITH SERDEPROPERTIES ('hbase.columns.mapping' = ':key,New:Singer,New:Song,New:Duration,New:MusicDirector')TBLPROPERTIES ('hbase.table.name' = 'Savita\_db:singerinfo') ;

hive> insert overwrite table AlbumToHBase select \* from album;

hive> select \* from AlbumToHBase;

A screen shot of a song

Description automatically generated

import org.apache.hadoop.hbase.filter.SingleColumnValueFilter

import org.apache.hadoop.hbase.filter.CompareFilter

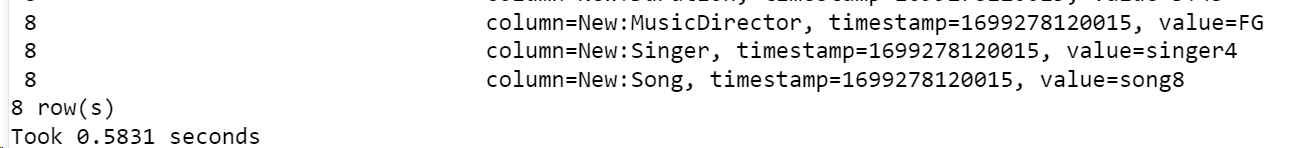
import org.apache.hadoop.hbase.filter.BinaryComparator

**There shoud not be any data loss when moving data from hive to hbase**

**Scan ‘koushikhbase:newalbum’**

A screenshot of a computer

Description automatically generated



1. There shoud not be any data loss when moving data from hive to hbase

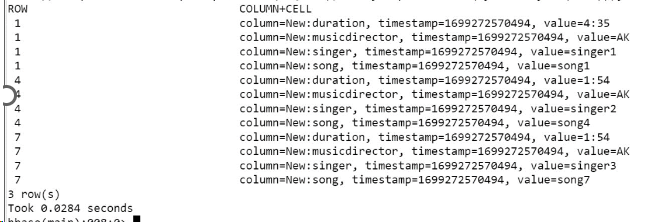
[mavricbdhoct07@ip-10-1-1-204 ~]$ hbase shell

hbase(main):003:0> scan 'Savita\_db:singerinfo'

hbase(main):014:0>

scan 'Savita\_db:singerinfo', { FILTER => SingleColumnValueFilter.new(Bytes.toBytes('New'), Bytes.toBytes('musicdirector'

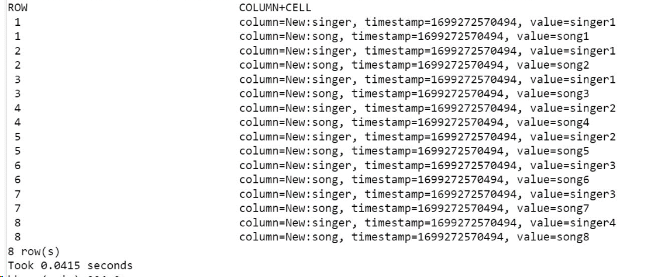
), CompareFilter::CompareOp.valueOf('EQUAL'),BinaryComparator.new(Bytes.toBytes('AK')))}



3. Run queries like 1) getall songs sung by a singer 2) Which singer sung for music director AK and what were the songs

1. **Get all songs sung by a singer**

hbase(main):006:0> scan 'Savita\_db:singerinfo', {COLUMN=>['New:singer','New:song']}



1. Which singer sung for music director AK and what were the songs

import org.apache.hadoop.hbase.filter.SingleColumnValueFilter

import org.apache.hadoop.hbase.filter.CompareFilter

import org.apache.hadoop.hbase.filter.BinaryComparator

hbase(main):005:0>

scan 'Savita\_db:singerinfo', {COLUMNS=>['New:singer','New:song','New:music\_director'],FILTER => SingleColumnValueFilter.new(Bytes.toBytes('New'), Bytes.toBytes('music\_director'),CompareFilter::CompareOp.value Of('EQUAL'),BinaryComparator.new(Bytes.toBytes('AK')))}

A screen shot of a computer

Description automatically generated

scan 'Savita\_db:singerinfo', {COLUMN=>['New:singer','New:song']}

