----------------------------------------------------------------

--CIND 719 Assignment 1 DDL Chantal Sylvestre--

----------------------------------------------------------------

----------------------------------

DDL for Data Preparation

-----------------------------------

--Upload data onto VM, then copy files to Hadoop fs

hadoop fs -put root/lab/station\_data.csv /user/lab

hadoop fs -put /root/lab/trip\_data.csv /user/lab

--Create tables in Hive

CREATE DATABASE bike;

SHOW DATABASES;

USE bike;

--Create the tables

CREATE TABLE bike.station(

station\_id string,

station\_name string,

lat string,

long string,

dockcount int,

landmark string,

installation string)

row format delimited

fields terminated by ',' ;

CREATE TABLE bike.trip(

trip\_id int,

duration int,

start\_date string,

start\_station string,

start\_terminal string,

end\_date string,

end\_station string,

end\_terminal string,

bike\_num string,

sub\_type string,

zip int)

row format delimited

fields terminated by ',' ;

--Load data into tables

LOAD DATA INPATH '/user/lab/station\_data.csv'

OVERWRITE INTO TABLE bike.station;

DESCRIBE bike.station;

LOAD DATA INPATH '/user/lab/trip\_data.csv'

OVERWRITE INTO TABLE bike.trip;

DESCRIBE bike.trip;

--Creating join table on start station and station name

--and removing stations that start and end at the same station.

CREATE TABLE bike.joined AS

SELECT \*

FROM bike.trip a JOIN bike.station b

ON a.start\_station = b.station\_name AND a.start\_station <> a.end\_station;

-------------------------------------------------------------------------

Assignment 1 Hive Script for Questions Chantal Sylvestre

--------------------------------------------------------------------------

--Question 1

SELECT bike\_num, COUNT(trip\_id) AS cnt

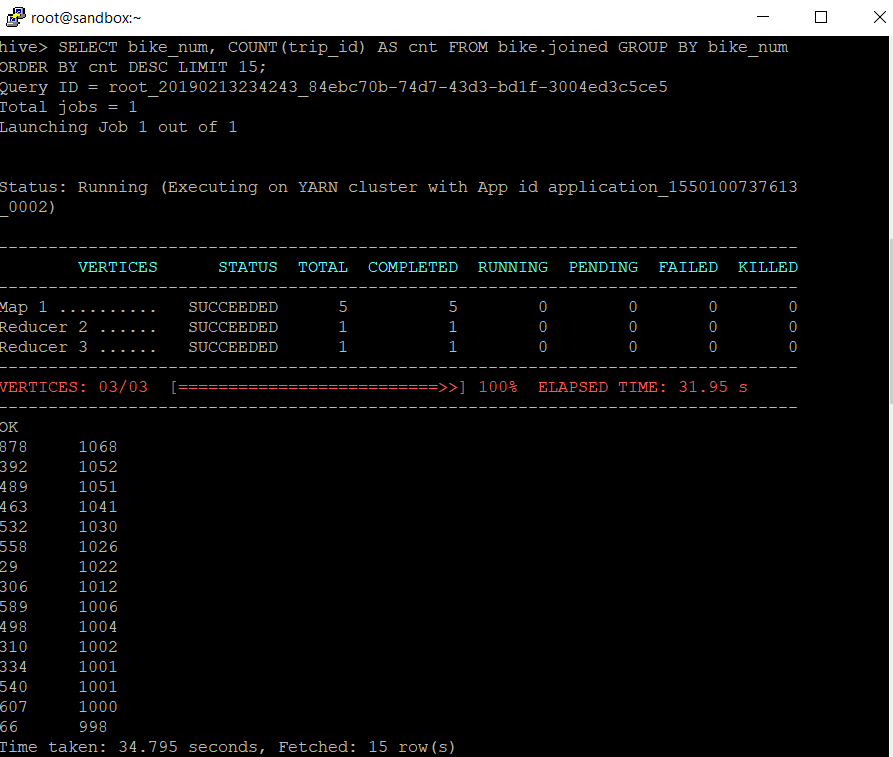
FROM bike.joined

GROUP BY bike\_num

ORDER BY cnt DESC

LIMIT 15;

--The answer is bike number 878 with 1,068 trips



--Question 2

SELECT sub\_type, COUNT(trip\_id) AS cnt

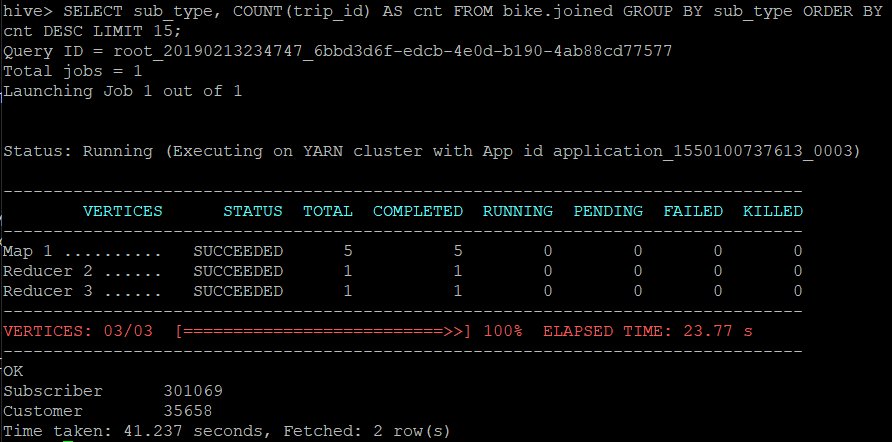
FROM bike.joined

GROUP BY sub\_type

ORDER BY cnt DESC

LIMIT 15;

--The answer is subscriber 301,069 rides and customer 35,658 rides



--Question 3

--Create external table in assignment 1 folder on Hadoop fs for data to go into as comma separated textfile

CREATE EXTERNAL TABLE bike.stationlist (

start\_station string,

end\_station string,

duration int)

row format delimited

fields terminated by ','

stored as textfile

location '/user/assignment1/stationlist';

--Insert data into external table

INSERT OVERWRITE TABLE bike.stationlist

SELECT start\_station, end\_station, MIN(duration) as duration

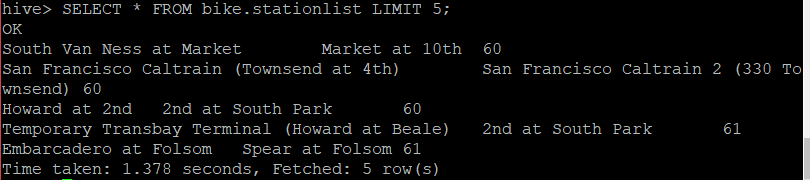
FROM bike.joined

GROUP BY start\_station, end\_station

ORDER BY duration;

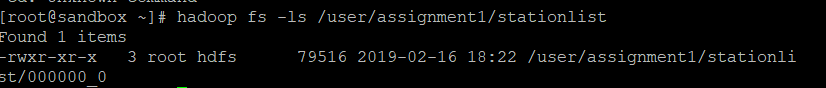
--First 5 lines of output file

SELECT \* FROM bike.stationlist LIMIT 5;



--Directory files under /user/assignment1/stationlist

Hadoop fs –ls /user/assignment/stationlist



--Question 4

SELECT landmark, COUNT(trip\_id) AS num\_of\_trips

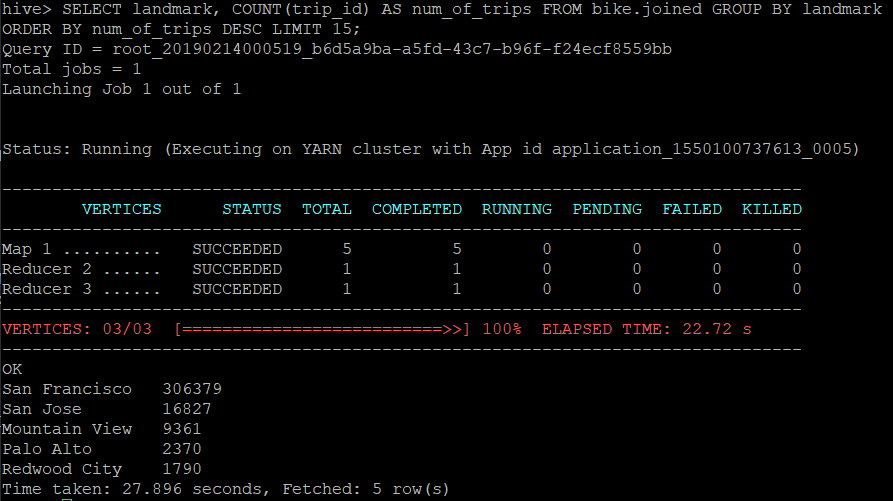
FROM bike.joined

GROUP BY landmark

ORDER BY num\_of\_trips DESC

LIMIT 15;

--San Francisco has the highest value with 306,379 trips originating from that landmark.



--Question 5

--Create table to join end\_terminal to landmark in addition to start\_station to landmark from the previous join so that there is a start landmark and an end landmark

CREATE TABLE bike.crossing AS

SELECT a.landmark as start\_landmark, b.landmark as end\_landmark, a.trip\_id

FROM bike.joined a JOIN bike.station b

ON a.end\_terminal = b.station\_id;

--SELECT statement to show number of trips that originate in one landmark and end in another by counting trip ids and ordered from largest to smallest

SELECT start\_landmark, end\_landmark, COUNT(trip\_id) AS num\_of\_trips

FROM bike.crossing

WHERE start\_landmark <> end\_landmark

GROUP BY start\_landmark, end\_landmark

ORDER BY num\_of\_trips DESC

LIMIT 15;

