**Database schema**

create database bikeShare;

use database bikeShare;

create table stationData(stationId int, name string, lat array<double>, long array<double>, dockcount int, landmark string, installation string)

row format delimited fields terminated by ',';

create table tripData(tripId int, duration int, startDate string, startStation string, startTerminal int, endDate string, endStation string, endTerminal int, bikeNum int, subType string, zipCode string) row format delimited fields terminated by ',';

**Question 1**

Script

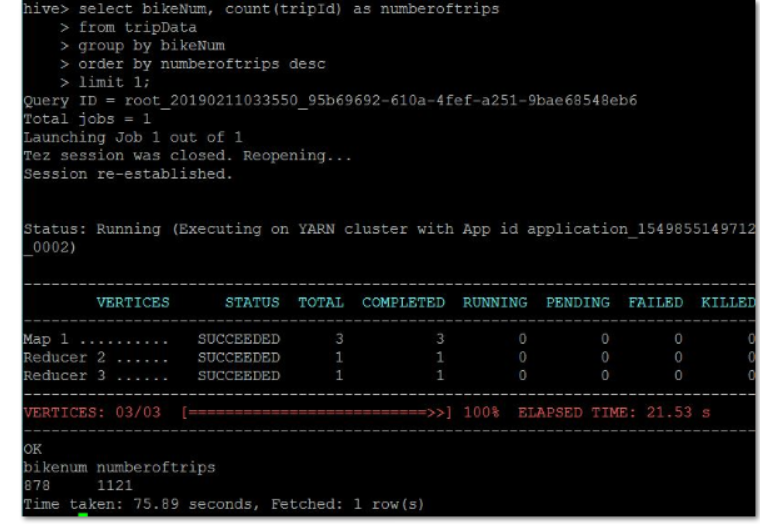
Select bikeNum, count(tripId) as numberoftrips

from tripData group by bikeNum

order by numberoftrips desc

limit 1;

Output



**Question 2**

Script

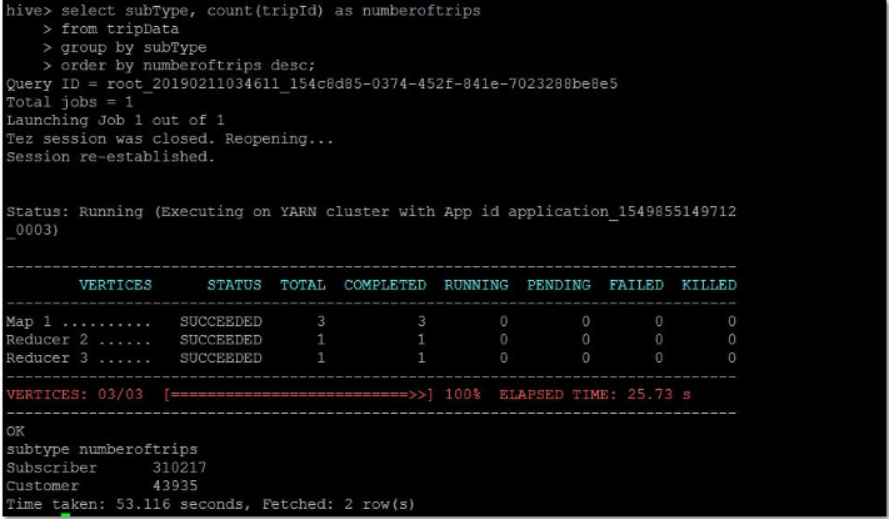
Select subType, count(tripId) as numberoftrips

from tripData

group by subtype

order by numberoftrips desc;

Output



**Question 3**

Script

--create the new table

create table connectedStations(startStation string, endStation string, tripNum int)

row format delimited fields terminated by ','

stored as textfile;

--insert output from query to new table

insert overwrite table connectedStations

select startStation, endStation, min(duration)

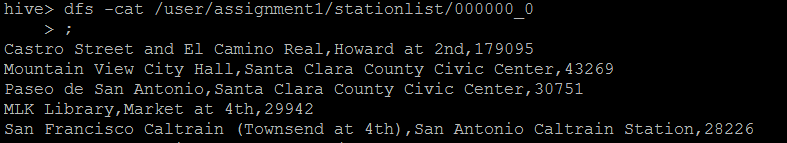
as minDuration from tripData group by startStation, endStation order by minDuration

desc;

--feeding the output of query to new table

insert overwrite directory '/user/assignment1/stationlist row format delimited fields terminated by ',' stored as textfile select \* from connectedStations;

Output

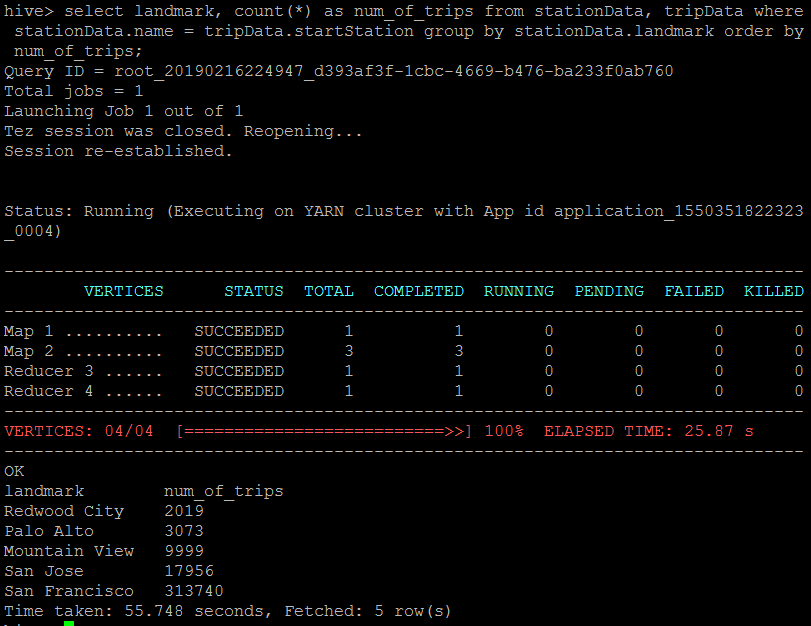


**Question 4**

Script

Select landmark, count(\*) as num\_of\_trips from stationData, tripData where stationData.name = tripData.startStation group by stationData.landmark order by num\_of\_trips;

Output

****

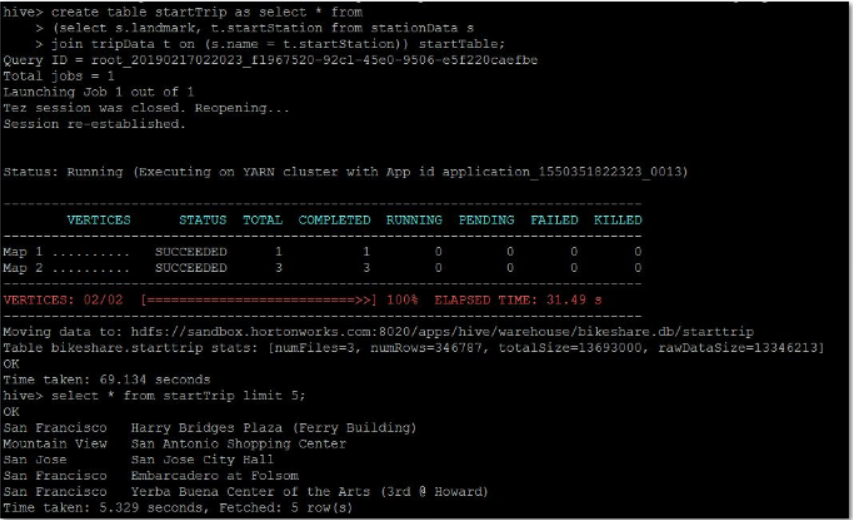
**Question 5**

Script

--create first table containing start station and start landmark

create table startTrip as select \* from stationData s

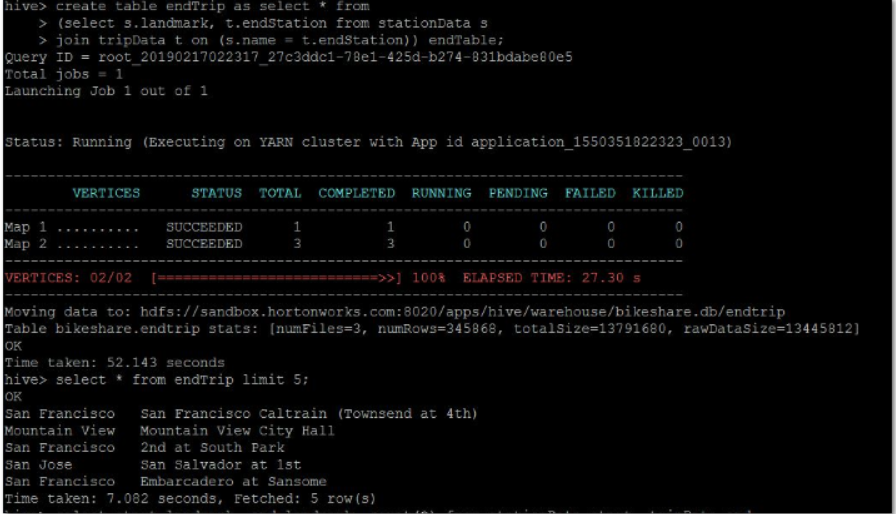
join tripData t on (s.name = t.startStation)) startTable;



--create second table containing end station and end landmark

create table endTrip as select \* from (select s.landmark, t.endStation from stationData s

join tripData t on (s.name = t.endStation)) endTable;



--display records where the end landmark does not equal start landmark

select startTrip.landmark from startTrip where startTrip.landmark not in (select endTrip.landmark from endTrip);

