1. HDFS AND HIVE

Problem Statement 1

Census Analytics is a project where you need to collect the data of people along with their incomes. As the census data is usually in large volume, the analysis of the data will be a cumbersome task. To overcome this, we will be using the Hadoop Ecosystem.

As a first step, you need to load the data into HDFS and create a table in Hive that can be used for querying the data. You have to create different types of tables and execute queries, as mentioned below and compare the time required for execution for different types of tables.

1 Download the dataset named **censusdata.csv** that is provided in your LMS.

2 Load the downloaded data into HDFS.

Ans2 Loaded the data using HDFS browser

3 Create an internal table in Hive to store the data

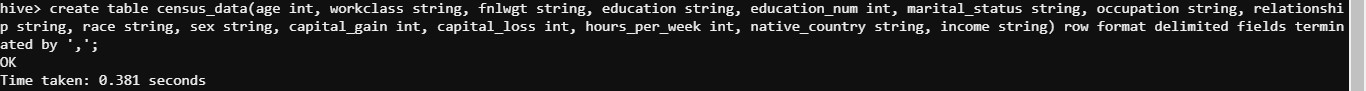
1. Create the table structure

create schema test\_tia13;

use test\_tia13;

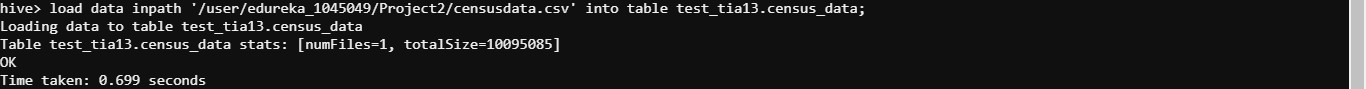
create table census\_data(age int, workclass string, fnlwgt string, education string, education\_num int, marital\_status string,

occupation string, relationship string, race string, sex string, capital\_gain int, capital\_loss int, hours\_per\_week int, native\_country string, income string) row format delimited fields terminated by ',';



1. Load the data from HDFS into the Hive table

load data inpath '/user/edureka\_1045049/Project2/censusdata.csv' into table test\_tia13.census\_data;

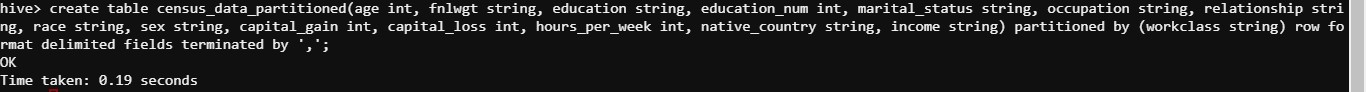


4 Create an internal table in Hive with partitions

1. Create a Partition Table in Hive using “workclass” as the Partition Key

create table census\_data\_partitioned(age int, fnlwgt string, education string, education\_num int, marital\_status string,

occupation string, relationship string, race string, sex string, capital\_gain int, capital\_loss int, hours\_per\_week int, native\_country string, income string) partitioned by (workclass string) row format delimited fields terminated by ',';

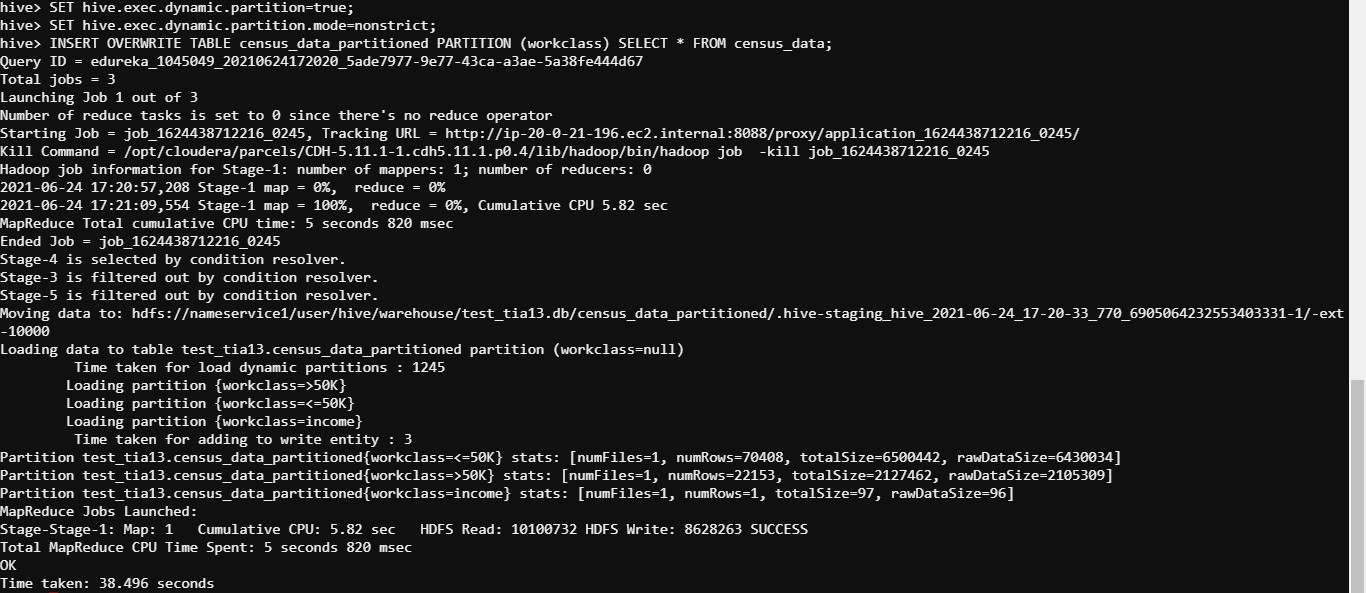


1. Load data from the staging table (Table created in Step 3) into this table

SET hive.exec.dynamic.partition=true;

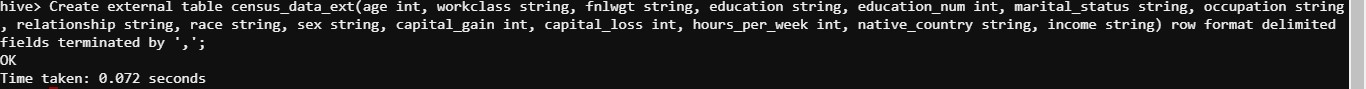
SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE census\_data\_partitioned PARTITION (workclass) SELECT \* FROM census\_data;

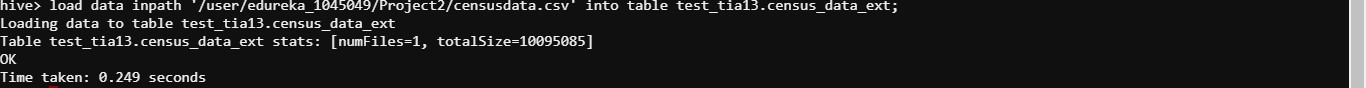


5 Create an external table in Hive to hold the same data stored in HDFS

Create external table census\_data\_ext(age int, workclass string, fnlwgt string, education string, education\_num int, marital\_status string, occupation string, relationship string, race string, sex string, capital\_gain int, capital\_loss int, hours\_per\_week int, native\_country string, income string) row format delimited fields terminated by ',';



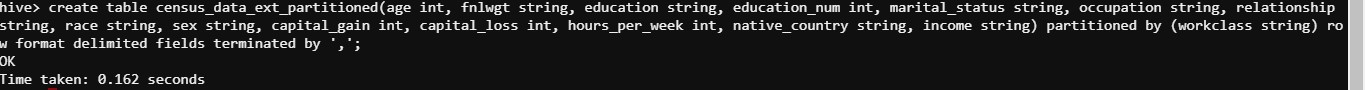
load data inpath '/user/edureka\_1045049/Project2/censusdata.csv' into table test\_tia13.census\_data\_ext;



6 Create an external table in Hive with partitions using “workclass” as Partition Key

create table census\_data\_ext\_partitioned(age int, fnlwgt string, education string, education\_num int, marital\_status string,

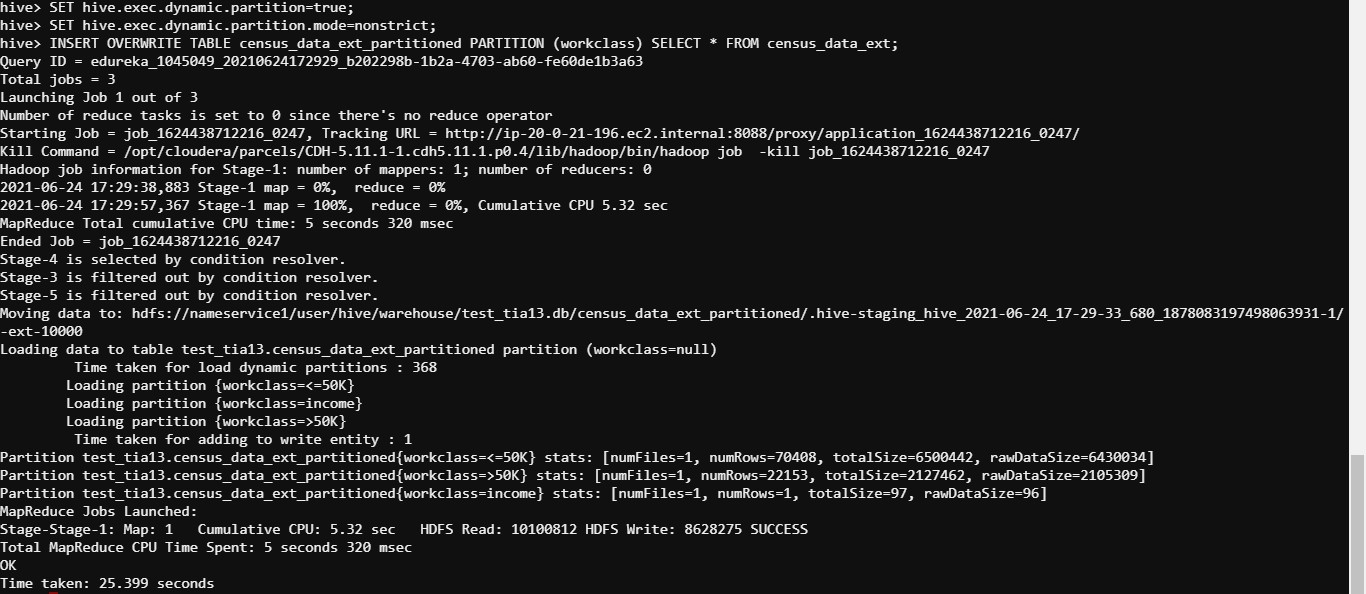
occupation string, relationship string, race string, sex string, capital\_gain int, capital\_loss int, hours\_per\_week int, native\_country string, income string) partitioned by (workclass string) row format delimited fields terminated by ',';



SET hive.exec.dynamic.partition=true;

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

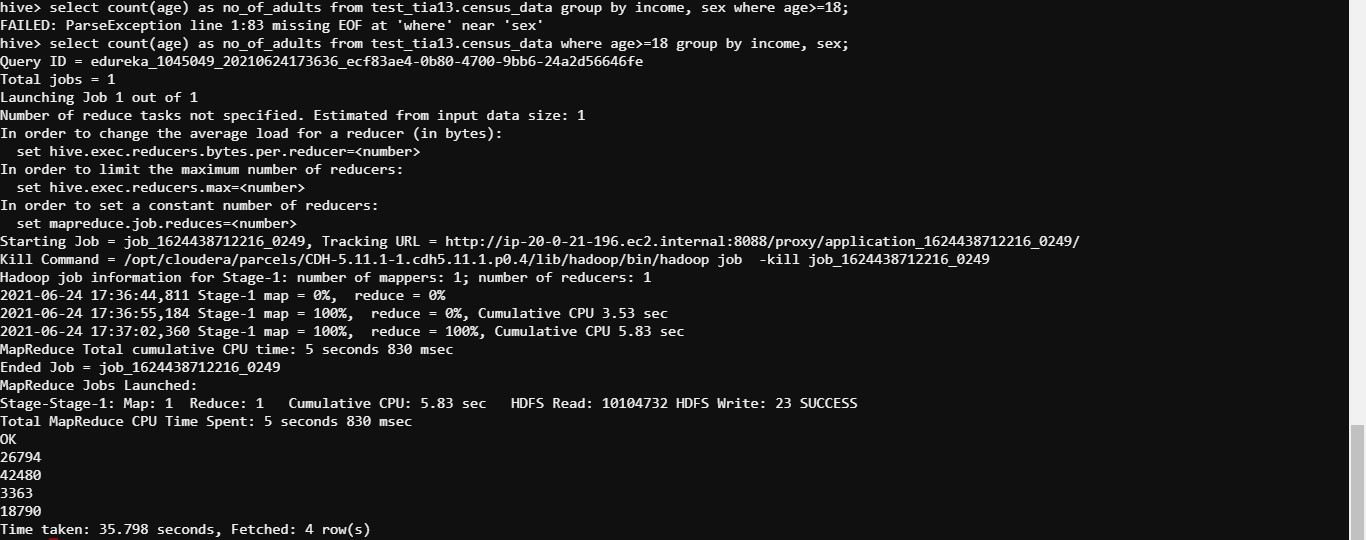
INSERT OVERWRITE TABLE census\_data\_ext\_partitioned PARTITION (workclass) SELECT \* FROM census\_data\_ext;



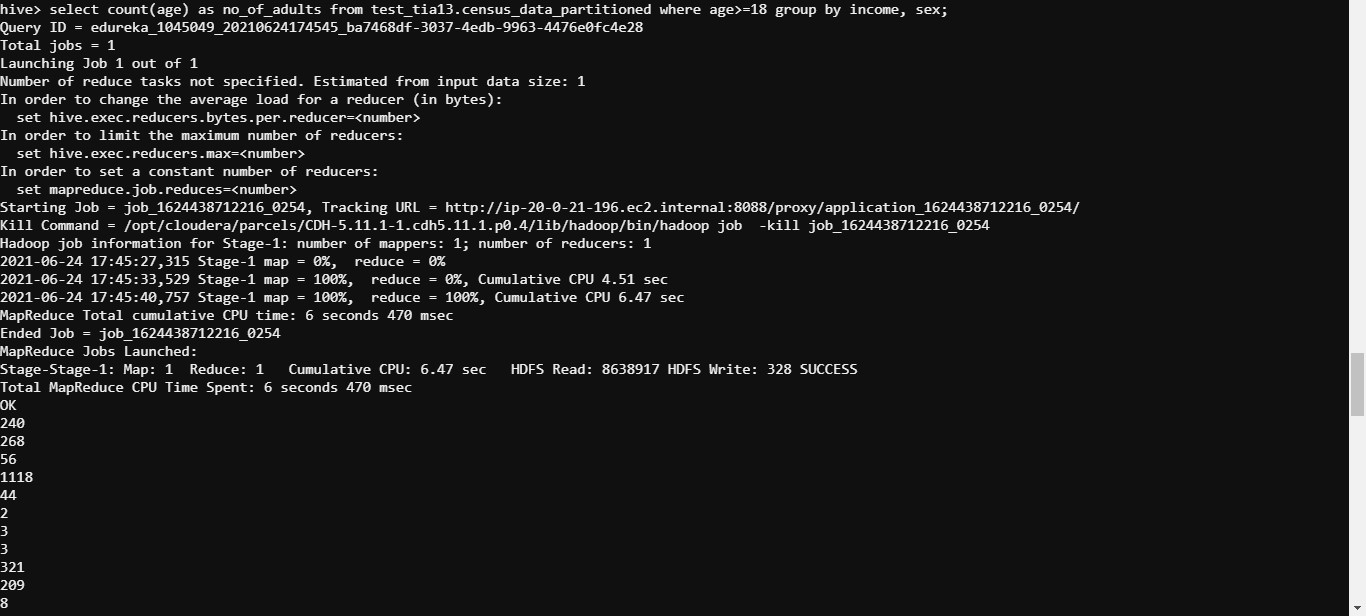
7 For each of the four tables created above, perform the following operations

1. Find out the number of adults based on income and gender. Note the time taken for getting the result

select count(age) as no\_of\_adults from test\_tia13.census\_data where age>=18 group by income, sex;

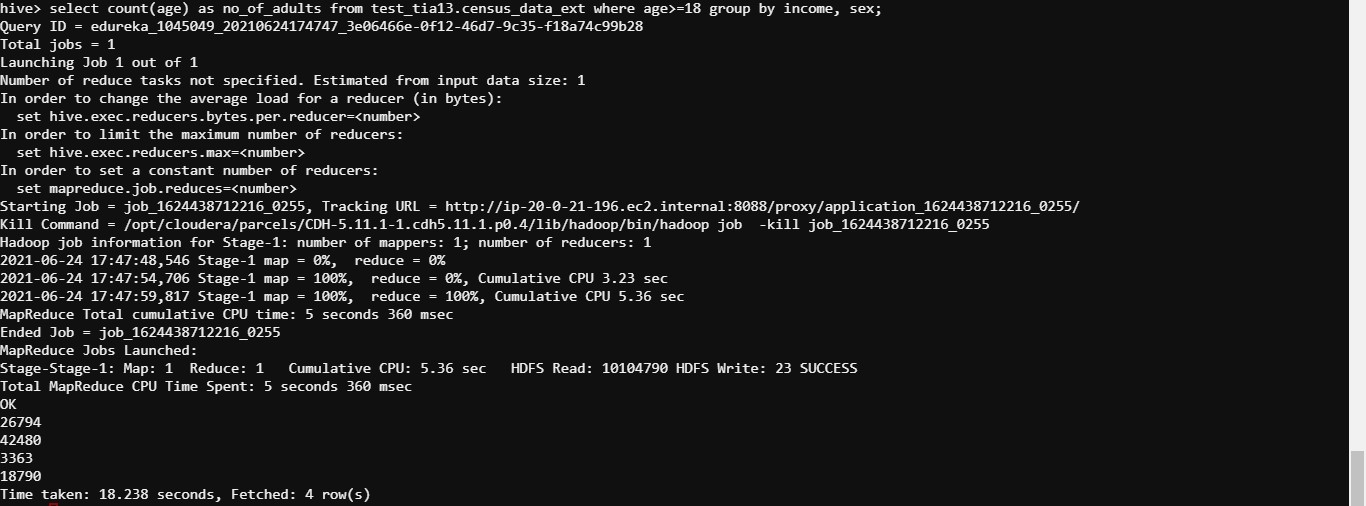


select count(age) as no\_of\_adults from test\_tia13.census\_data\_partitioned where age>=18 group by income, sex;

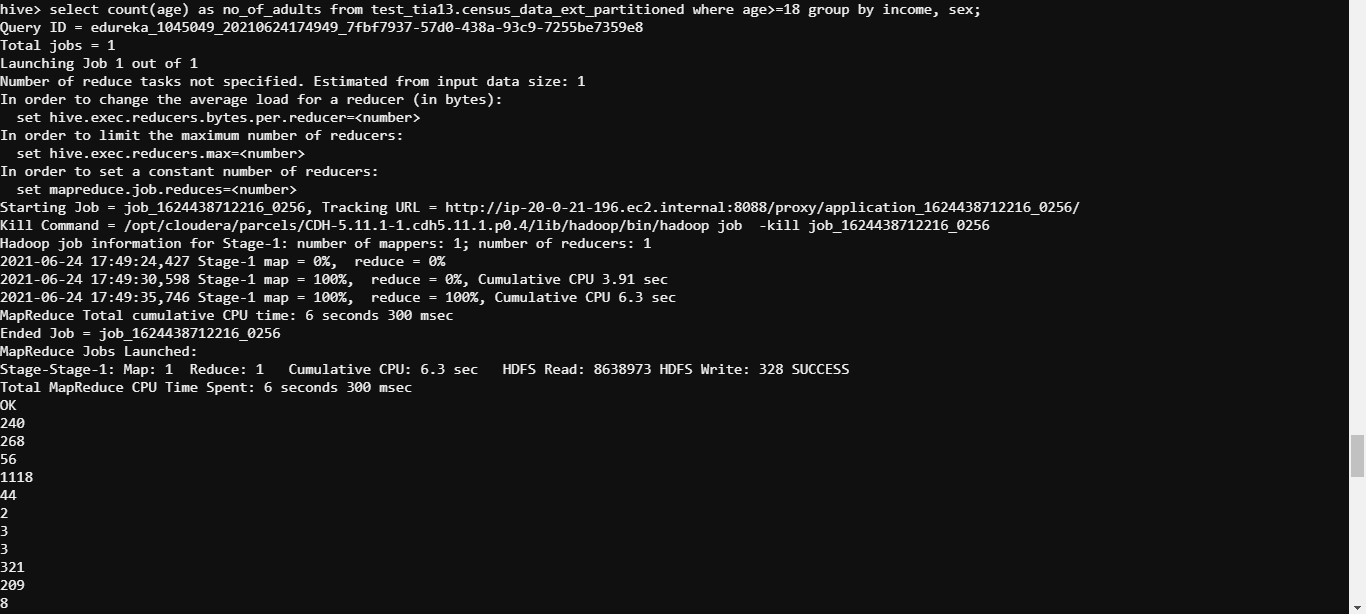




select count(age) as no\_of\_adults from test\_tia13.census\_data\_ext where age>=18 group by income, sex;



select count(age) as no\_of\_adults from test\_tia13.census\_data\_ext\_partitioned where age>=18 group by income, sex;

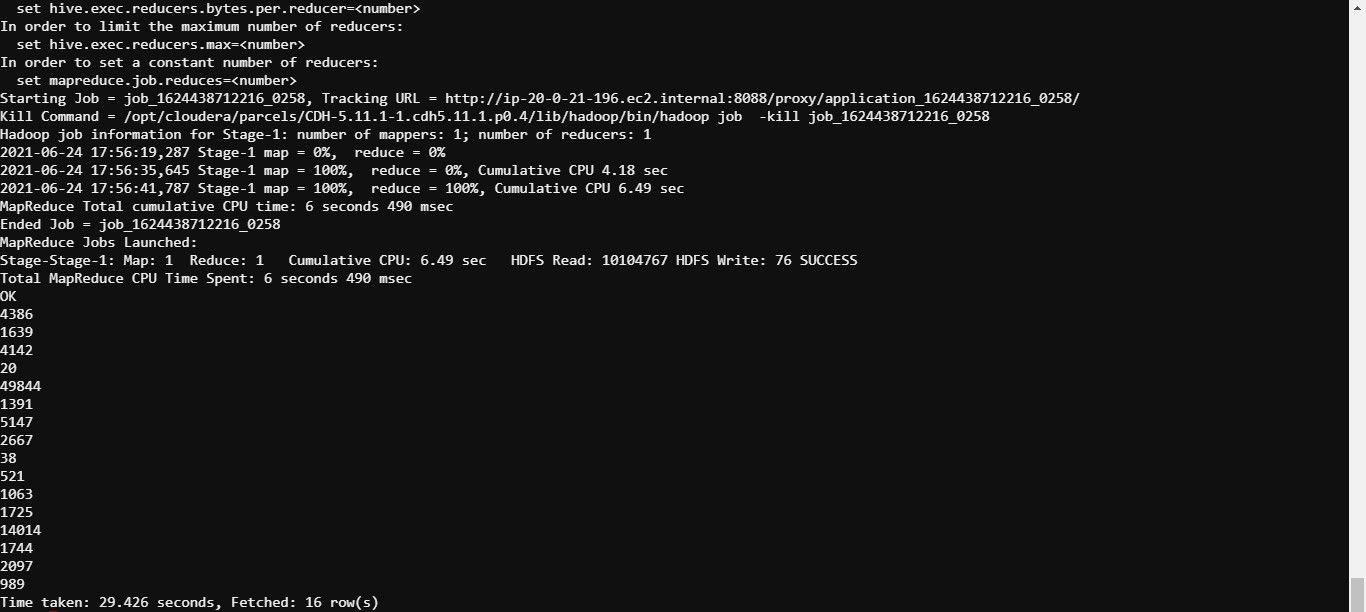




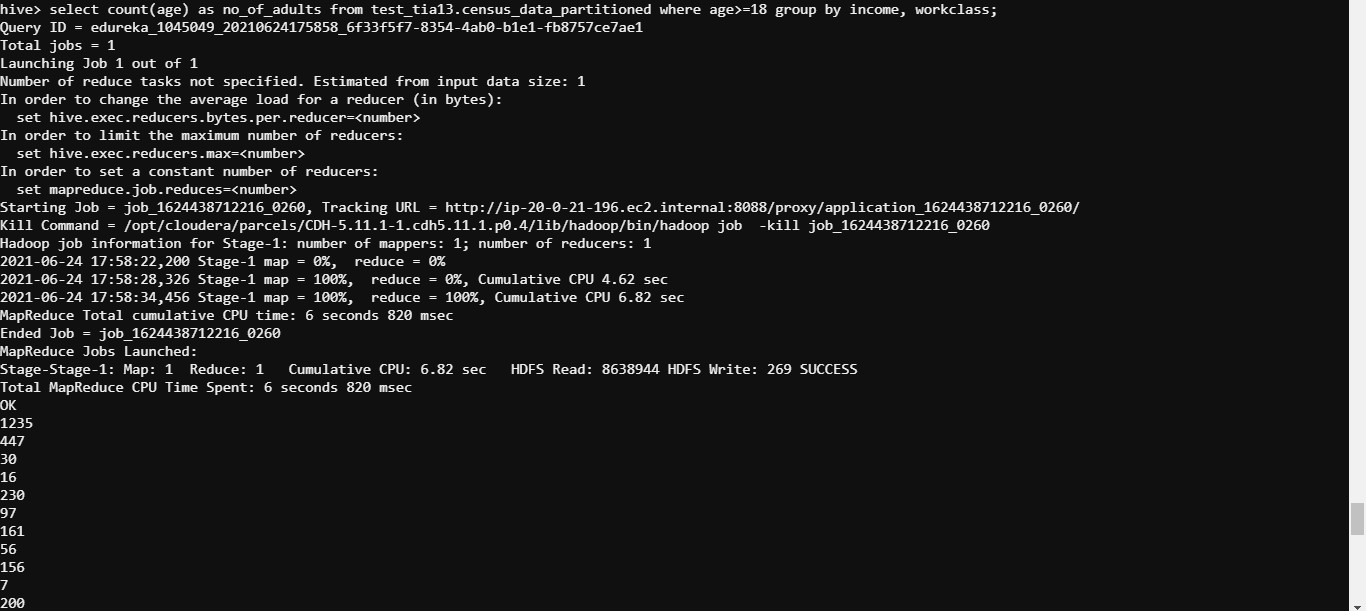
1. Find out the number of adults based on income and workclass. Note the time taken for getting the result

select count(age) as no\_of\_adults from test\_tia13.census\_data where age>=18 group by income, workclass;



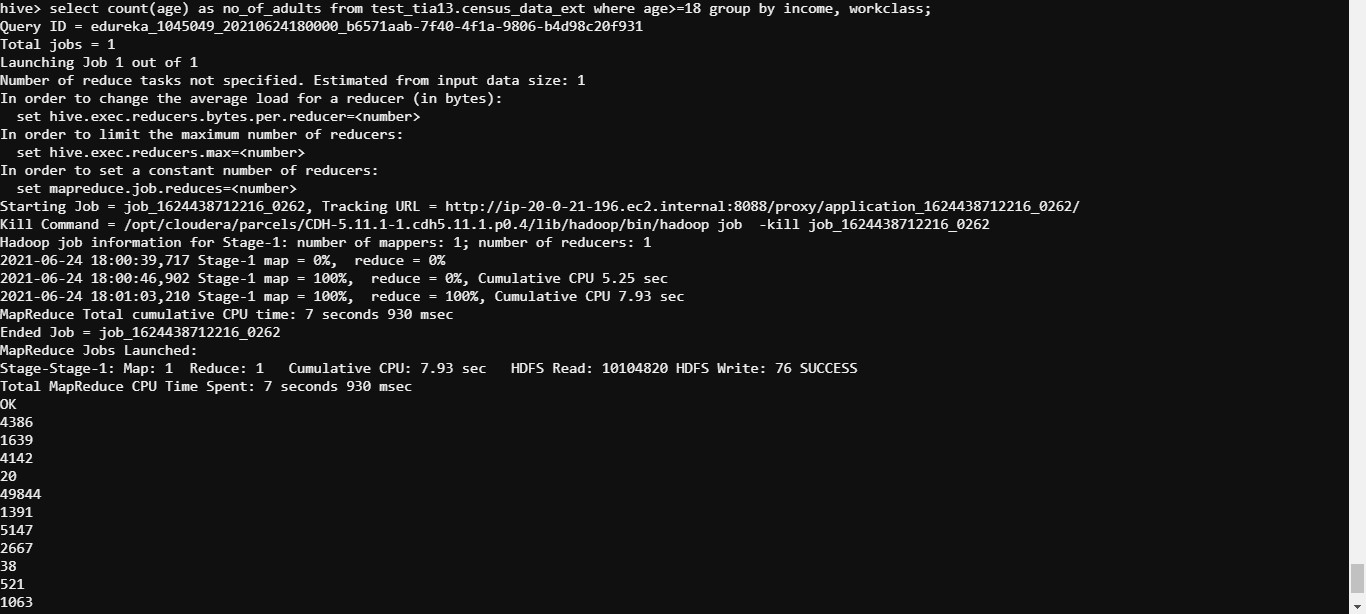


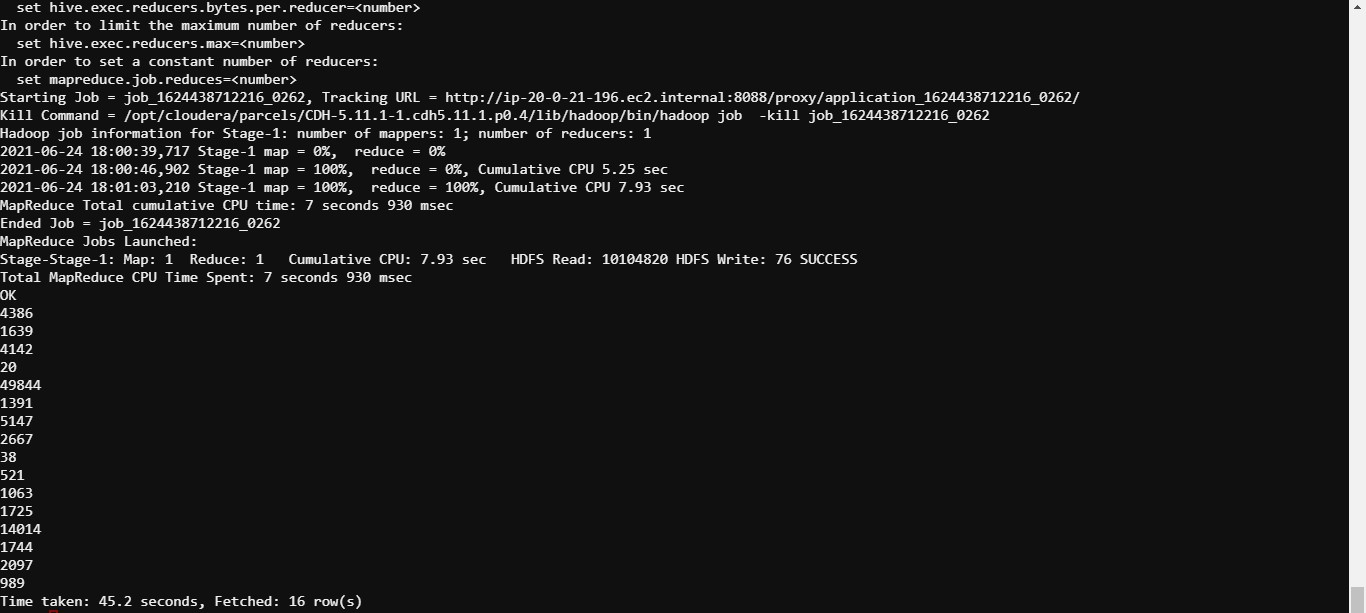
select count(age) as no\_of\_adults from test\_tia13.census\_data\_partitioned where age>=18 group by income, workclass;



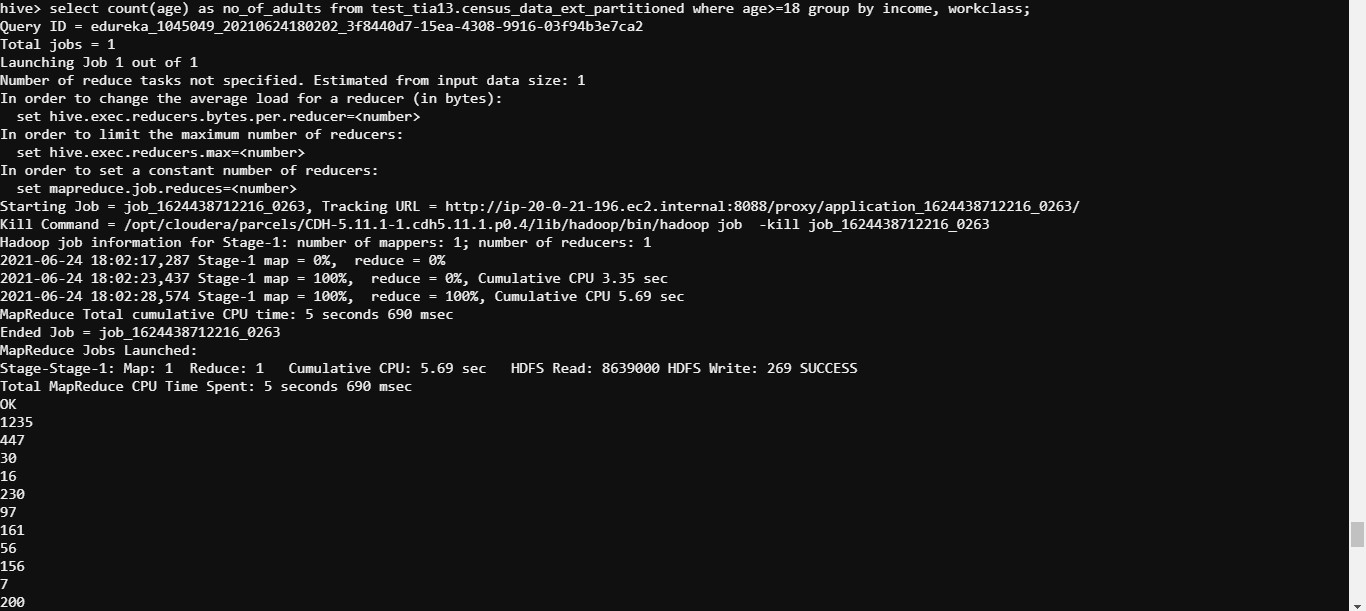


select count(age) as no\_of\_adults from test\_tia13.census\_data\_ext where age>=18 group by income, workclass;





select count(age) as no\_of\_adults from test\_tia13.census\_data\_ext\_partitioned where age>=18 group by income, workclass;





1. Write your observations by comparing the time taken for executing the commands between:
   1. Internal & External Tables

Time taken while working with Internal Tables - 35.798secs, 19.501secs, 29.426secs, 20.613secs;

Avg time taken while working on Internal Tables - 26.3345secs;

Time taken while working with External Tables - 18.238secs, 17.722secs, 45.2secs, 21.24secs;

Avg time taken while working on External Tables - 25.6secs;

Time taken by Internal Tables is more compared to time taken by External Tables.

* 1. Partitioned & Non-partitioned Tables

Time taken while working with Partitioned Tables - 19.501secs, 17.722secs, 20.613secs, 21.24secs;

Avg time taken while working on Partitioned Tables - 19.769secs;

Time taken while working with Non-partitioned Tables - 35.798secs, 18.238secs, 29.426secs, 45.2secs;

Avg time taken while working on Non-partitioned Tables - 32.1655secs;

Time taken by Non-partitioned Tables is more compared to time taken by Partitioned Tables.

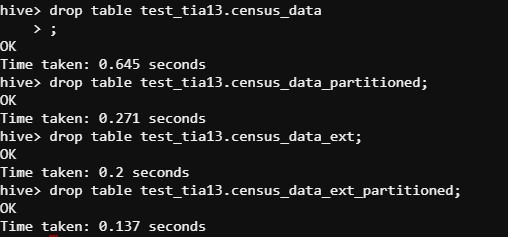
8 Delete the internal as well as external tables. Comment on the effect on data and metadata after the deletion is performed for both internal and external tables.

drop table test\_tia13.census\_data;

drop table test\_tia13.census\_data\_partitioned;

drop table test\_tia13.census\_data\_ext;

drop table test\_tia13.census\_data\_ext\_partitioned;



Both data and metadata(schema) gets deleted from the hive server once the tables are dropped in internal tables whereas metadata(schema) gets deleted from the hive server once the tables are dropped in external tables.