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

2. Load the downloaded data into HDFS

Hdfs dfs –copyFromLocal <path>;

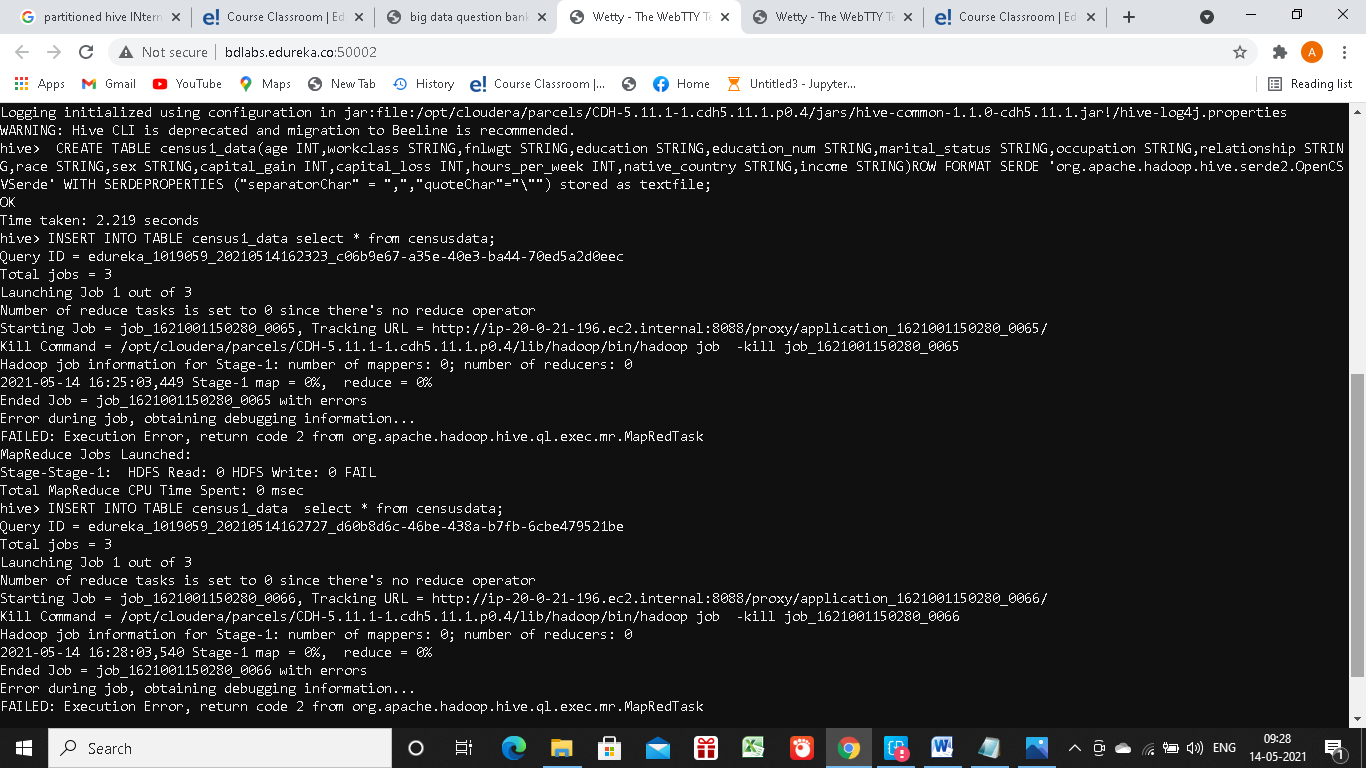
Uploaded the csv file censusdata.csv in hue.

* 1. 3. Create an internal table in Hive to store the data
  2. a. Create the table structure

hive;

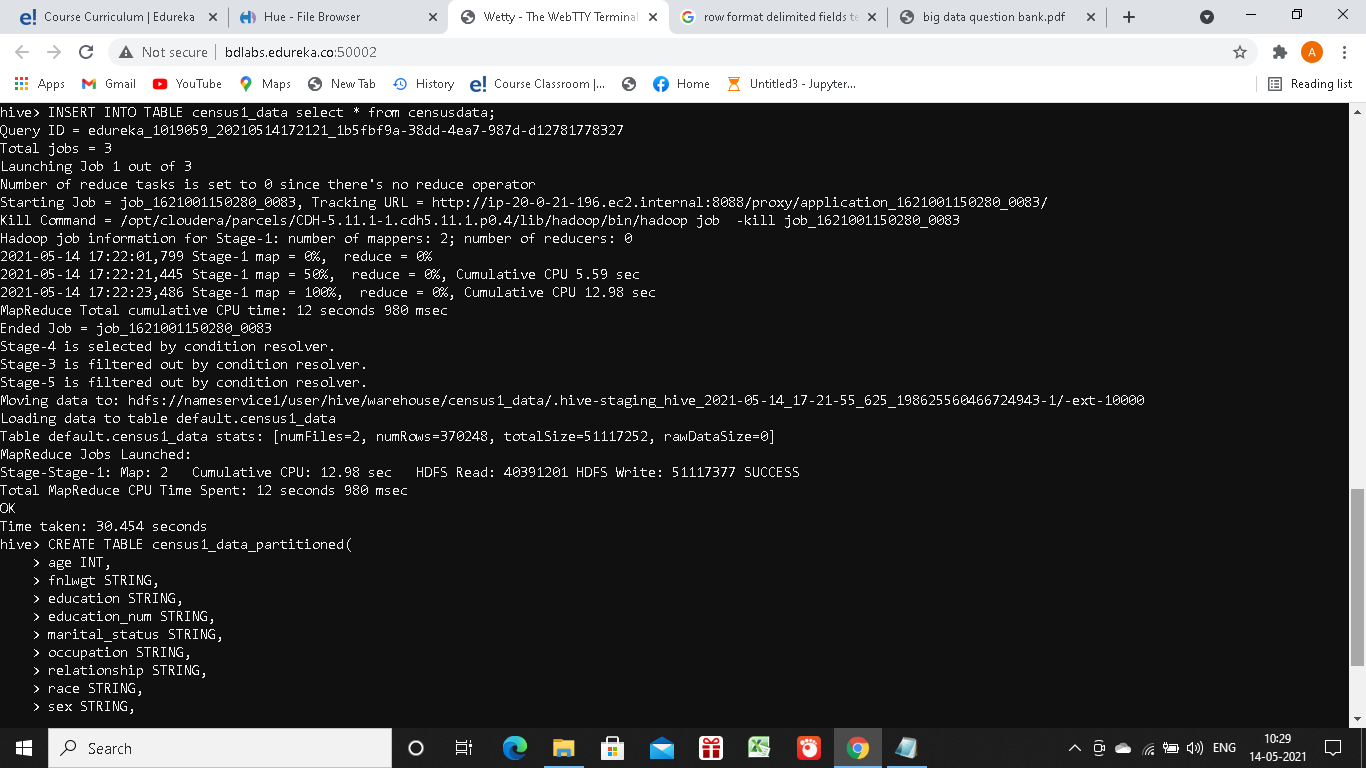
use Akash;

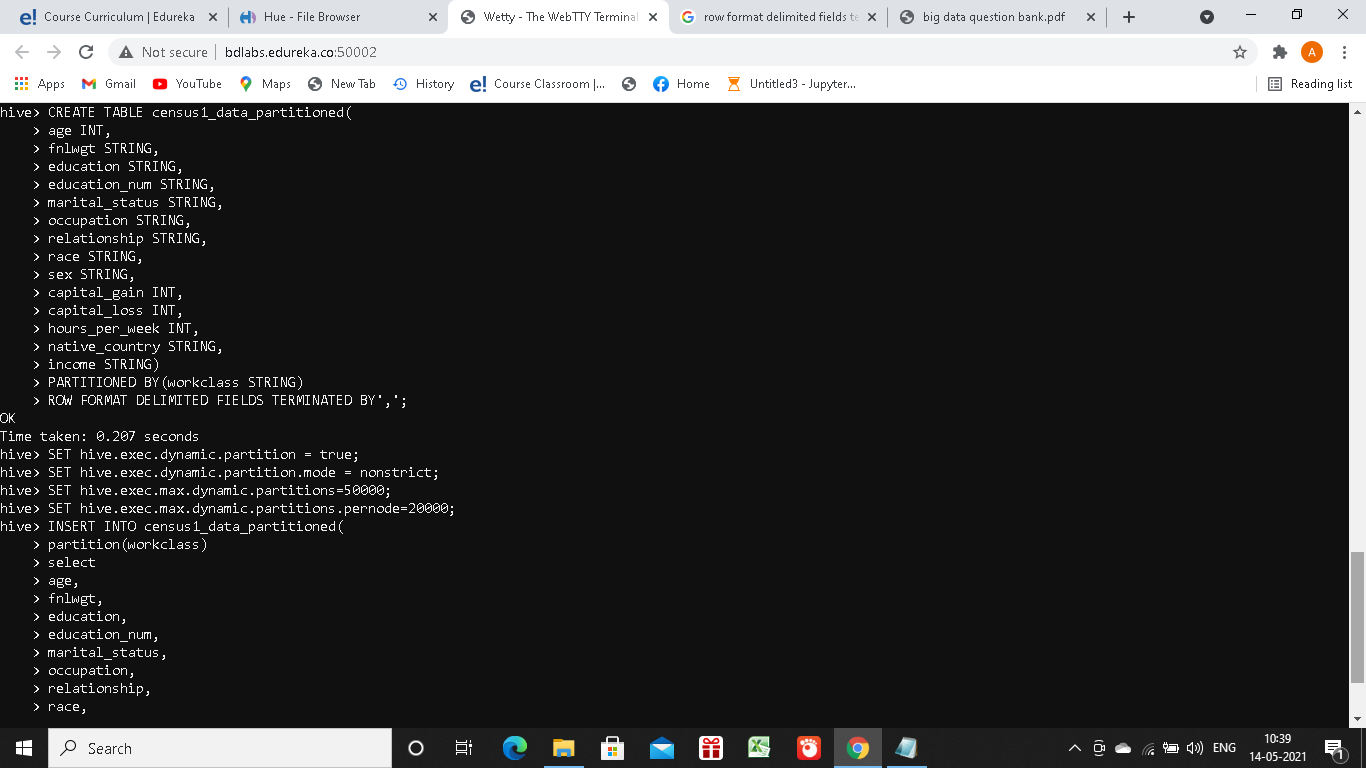
CREATE TABLE census1\_data(age INT, workclass STRING, fnlwgt STRING, education STRING,education-num INT ,marital-status STRING,occupation STRING ,relationship STRING,race STRING, sex STRING, capital-gain INT,capital-loss INT,hours-per-week INT,native-country STRING,income STRING) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separatorChar" = ",","quoteChar"="\"") stored as textfile;



* 2. b. Load the data from HDFS into the Hive table

*INSERT INTO TABLE census1\_data select \* from censusdata;*

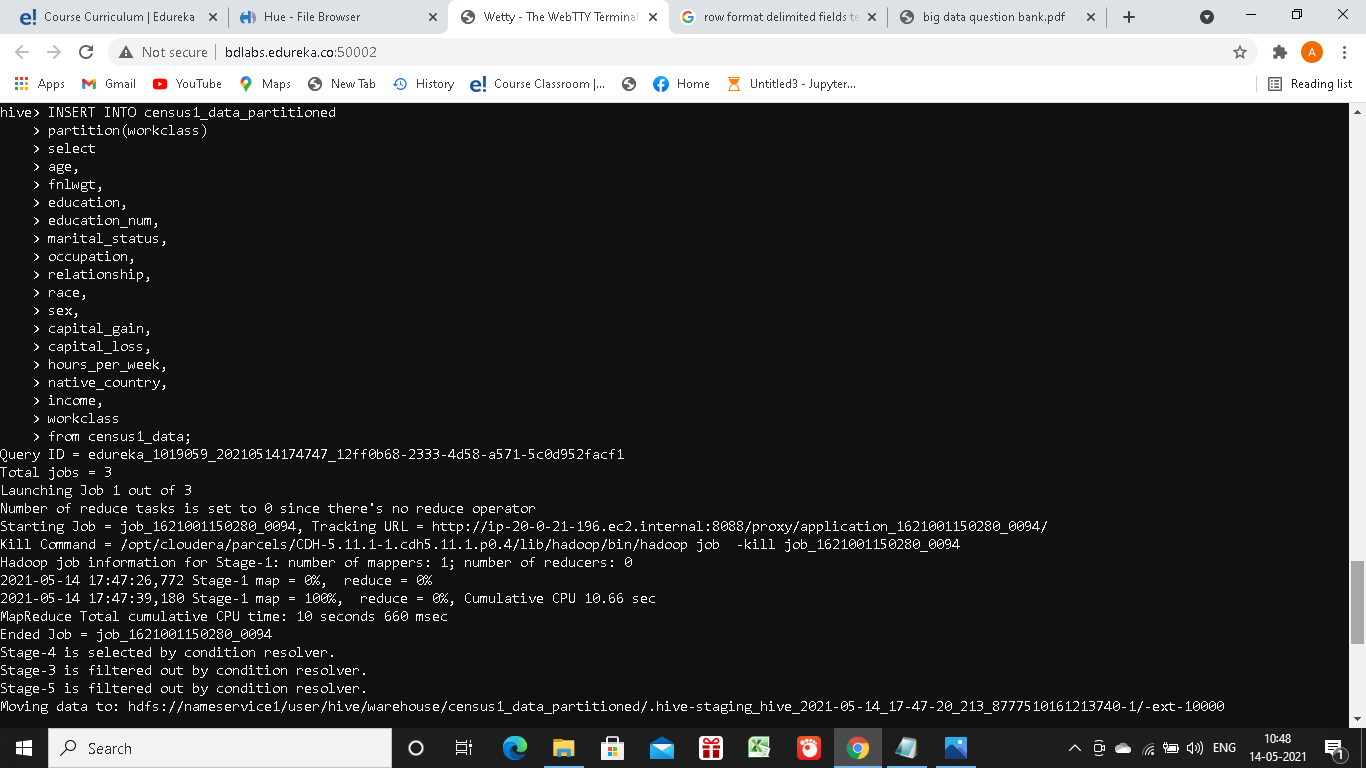


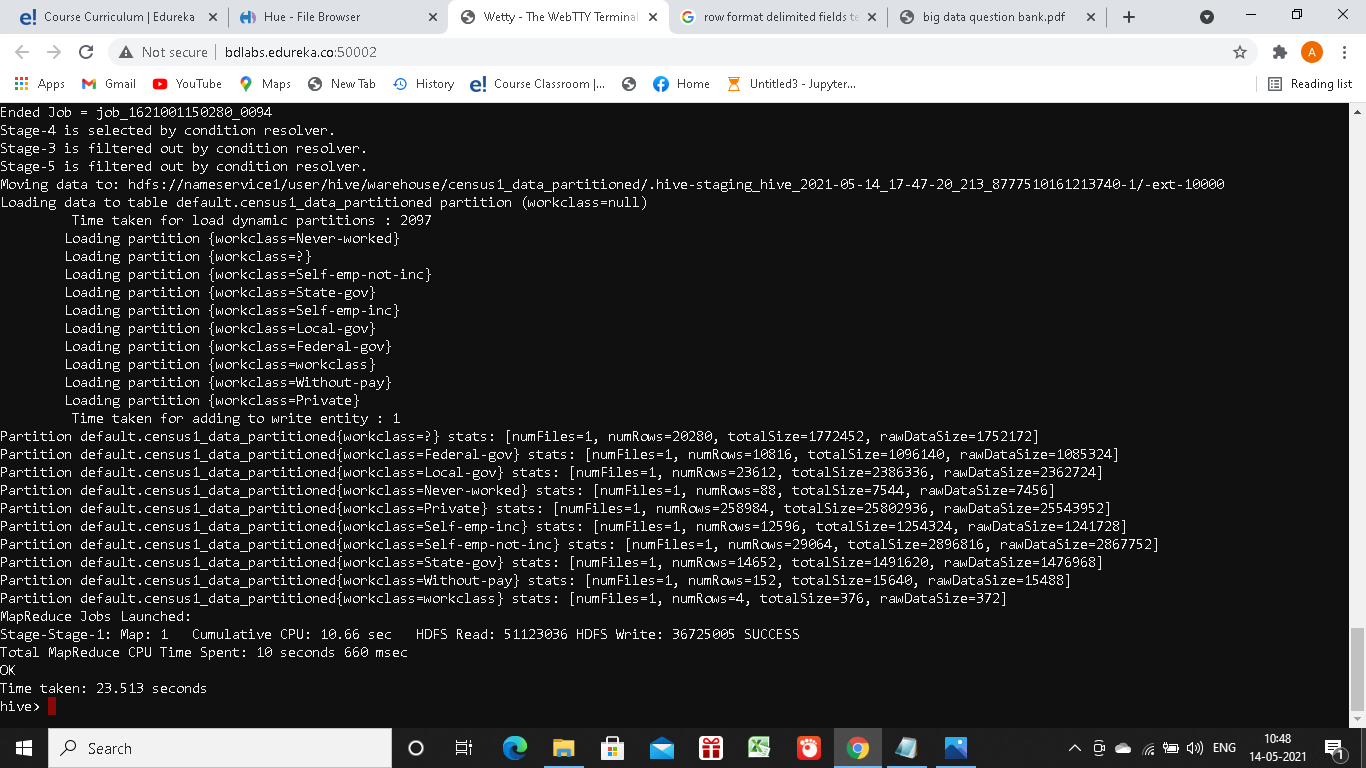
* 1. 4. Create an internal table in Hive with partitions
  2. a. Create a Partition Table in Hive using “workclass” as the Partition Key.
* Created another non partitioned table as above , named census with the same process and data.
* Inserted into the partitioned table census\_partition from census.
* censustable is later used for loading data into external nonpartitioned table in next question.
* CREATE TABLE census1\_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. 
  2. b. Load data from the staging table (Table created in Step 3) into this table

set hive.exec.dynamic.partition.mode=non-strict;

INSERT INTO census1\_data\_partitioned partition(workclass)

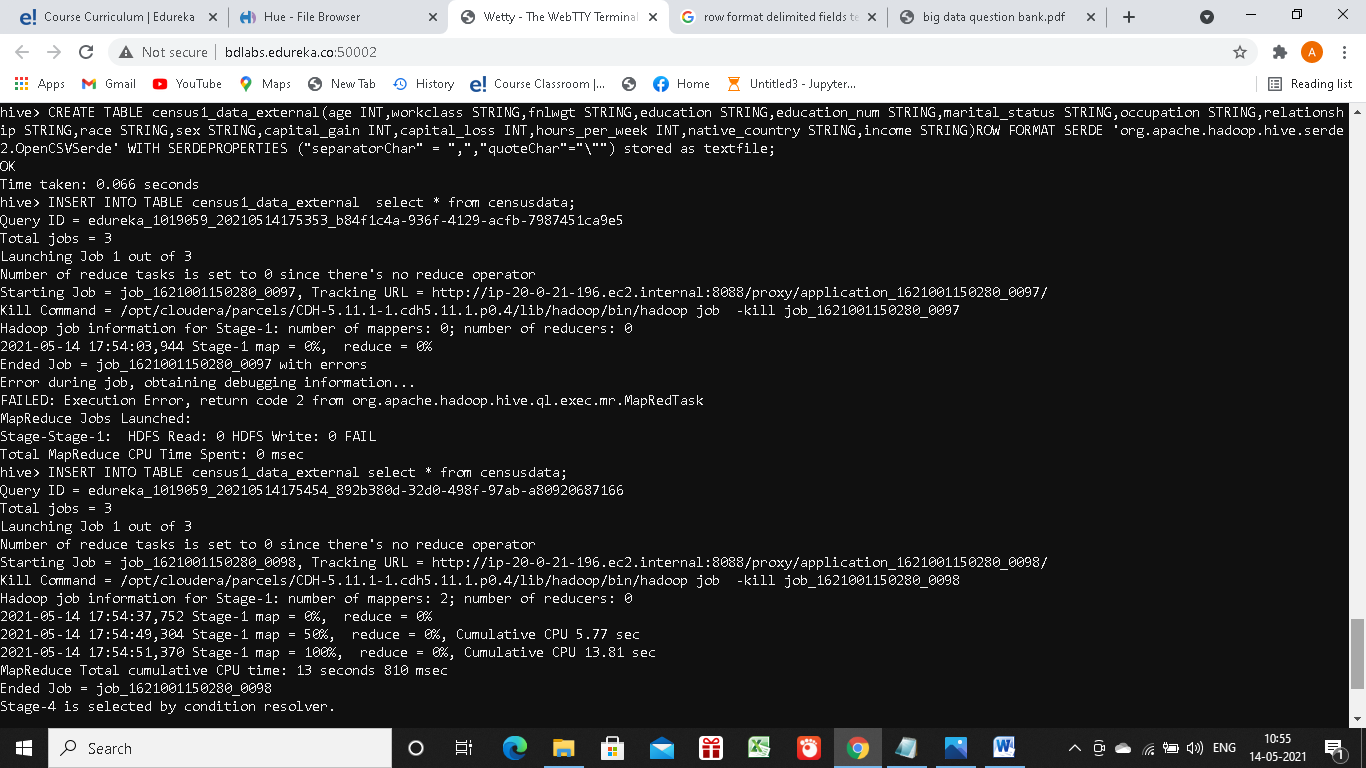
SELECT age,fnlwgt,education, `education-num`,`marital-status`,occupation,relationship,race,sex,`capital-gain`,`capital-loss`,`hours-per-week`,`native-country`,income,workclass FROM census1\_data;





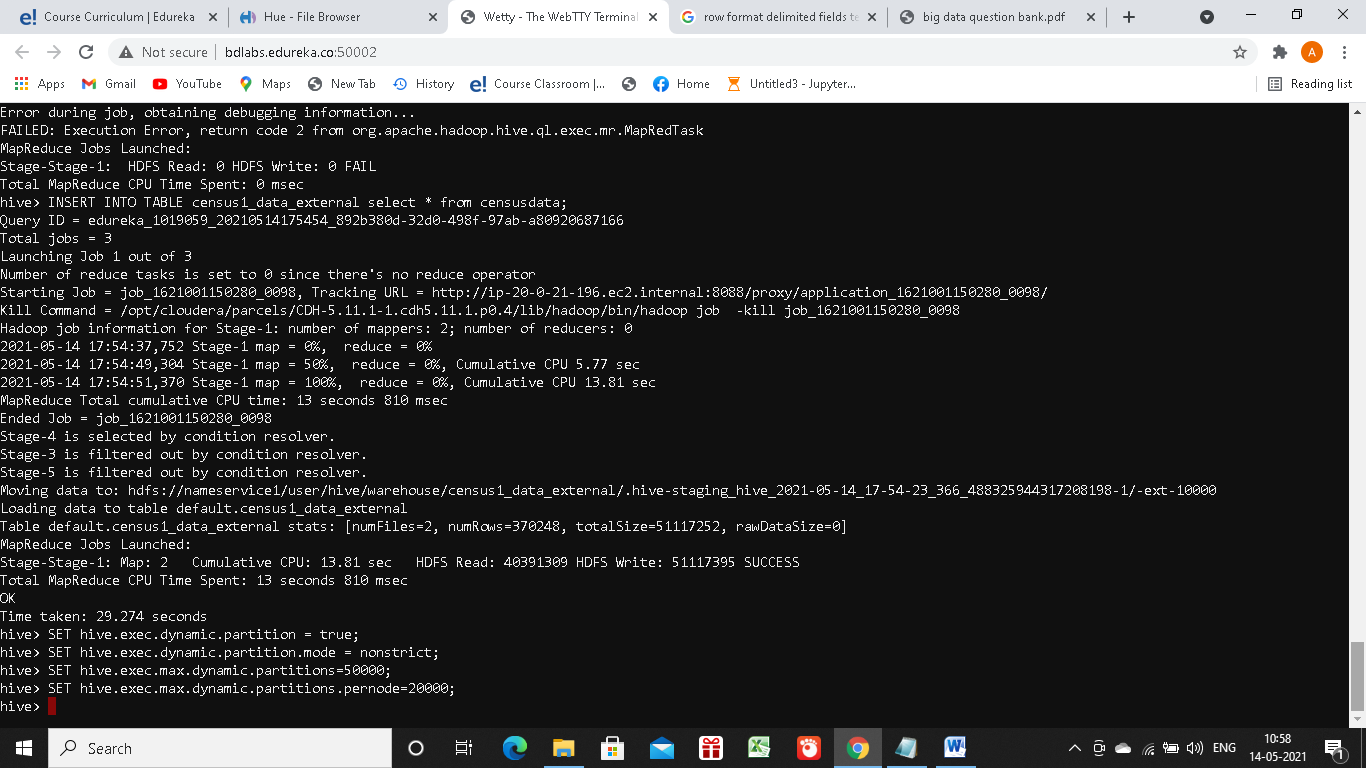
5. Create an external table in Hive to hold the same data stored in HDFS

CREATE TABLE census1\_data\_external(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 SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separatorChar" = ",","quoteChar"="\"") stored as textfile;



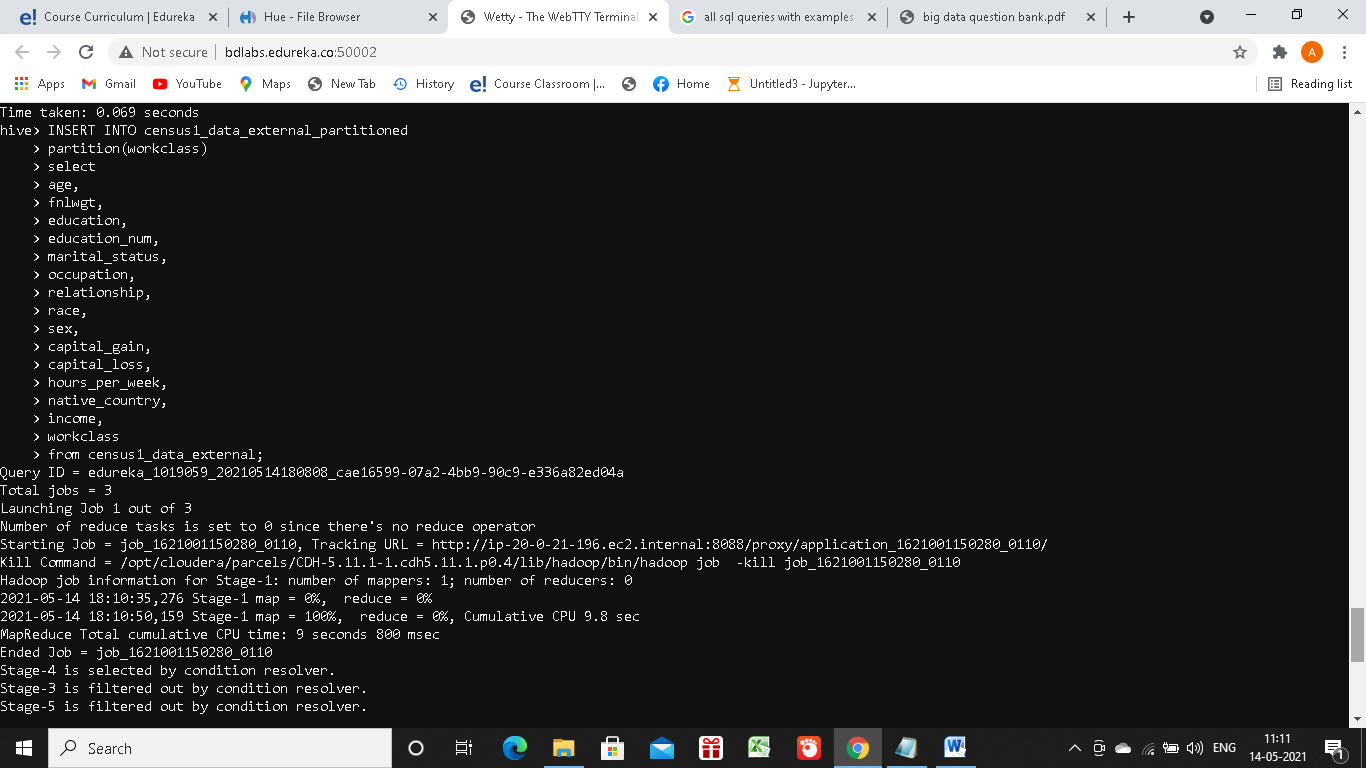
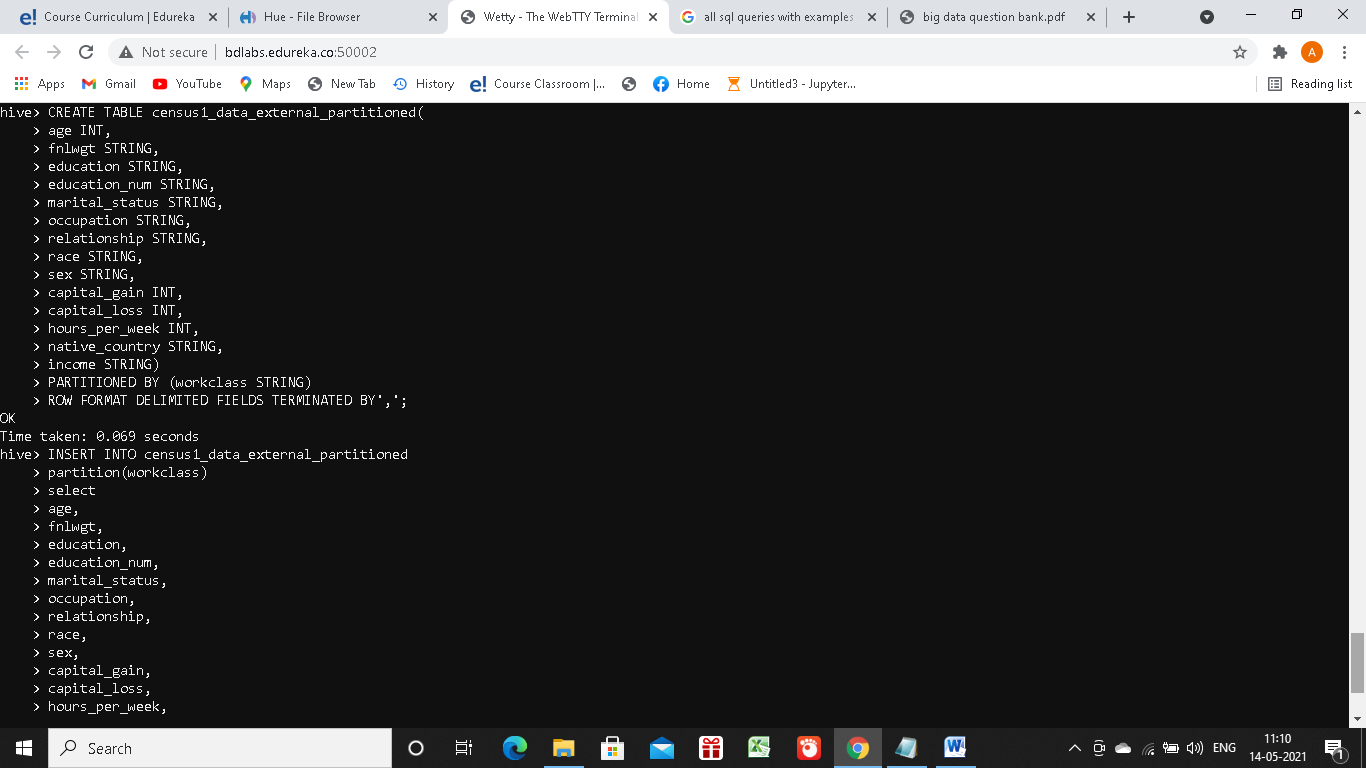
Load data into external table from internal table:

*INSERT INTO TABLE census1\_data \_external select \* from censusdata;*



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

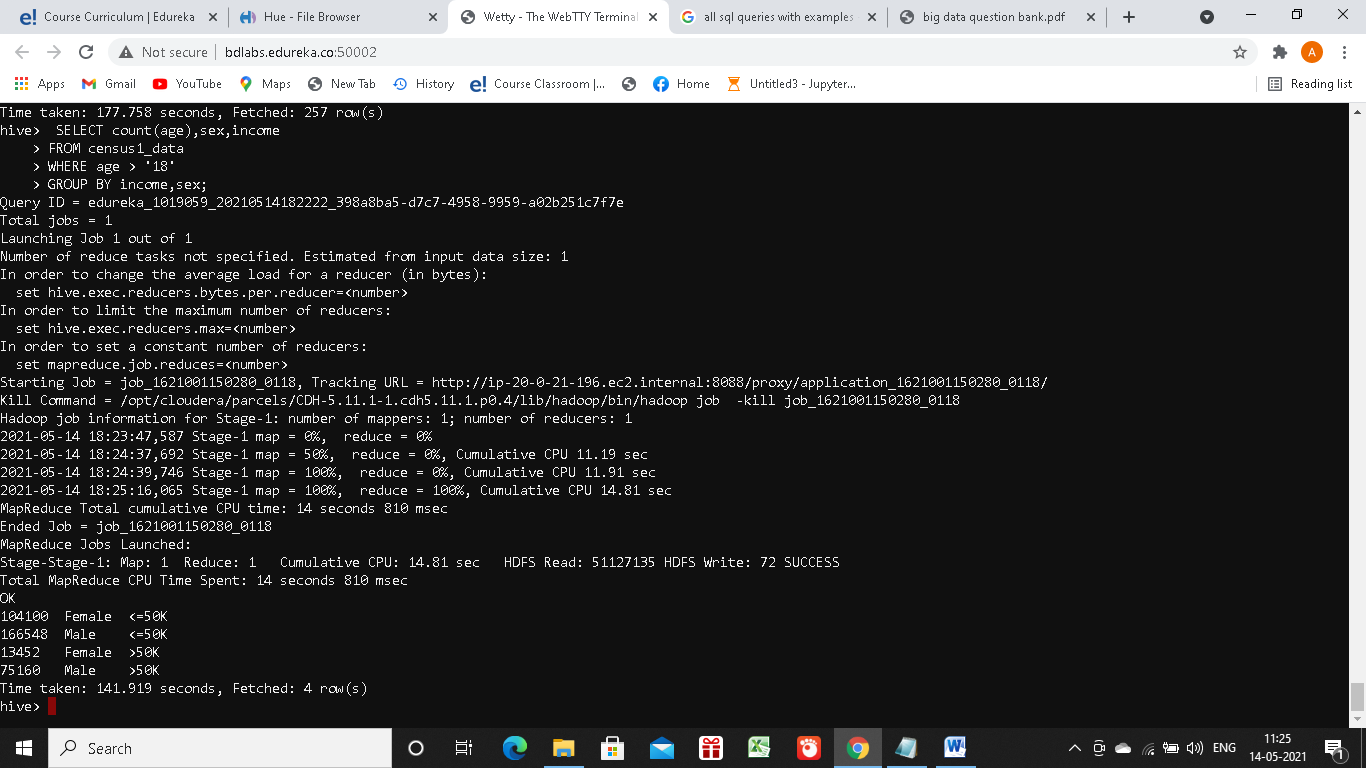
* CREATE TABLE census1\_data\_external\_partitioned(age INT,fnlwgt STRING, education STRING,education-num INT ,marital-status STRING,occupation STRING ,relationship STRING,race STRING, sex STRING, capital-gain INT,capital-loss INT,hours-per-week INT,native-country STRING,income STRING)PARTITIONED BY(workclass STRING)ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’;



7a. Find out the number of adults based on income and gender. Note the time taken for getting the result

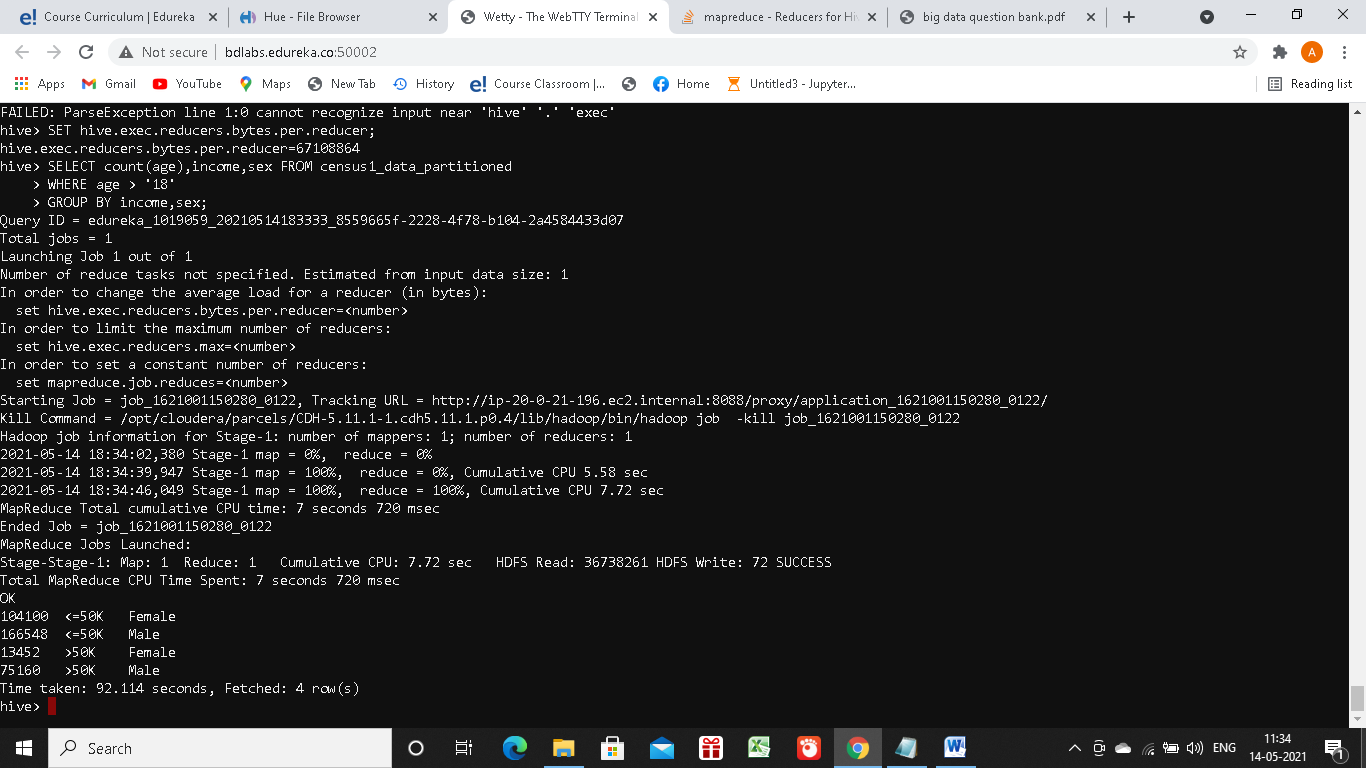
**internal table non-partitioned: 141.919 seconds**

Select count(age),sex,income From census1\_data WHERE age>’18’ GROUP BY income,sex;



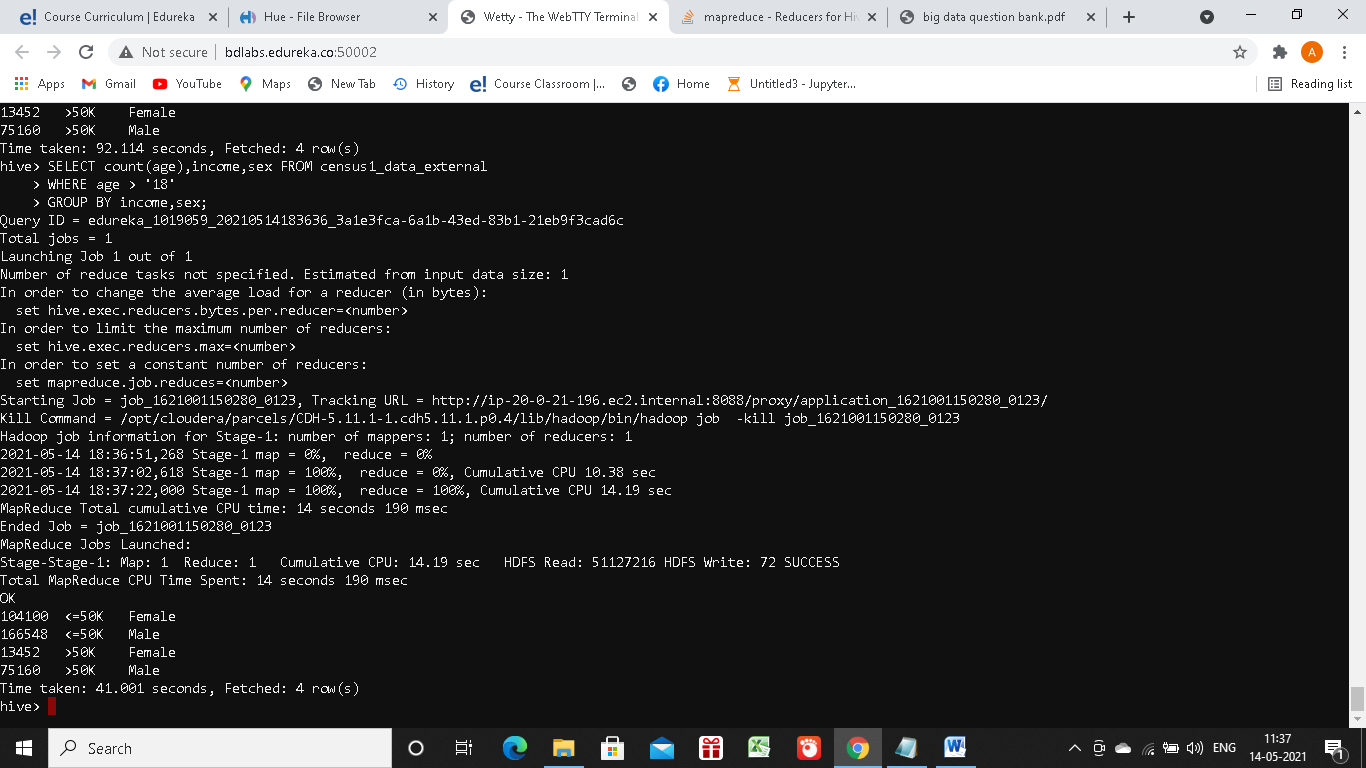
**internal table partitioned: 92.114 seconds**

Select count(age),sex,income From census1\_data \_partitioned WHERE age>’18’ GROUP BY income,sex;

****

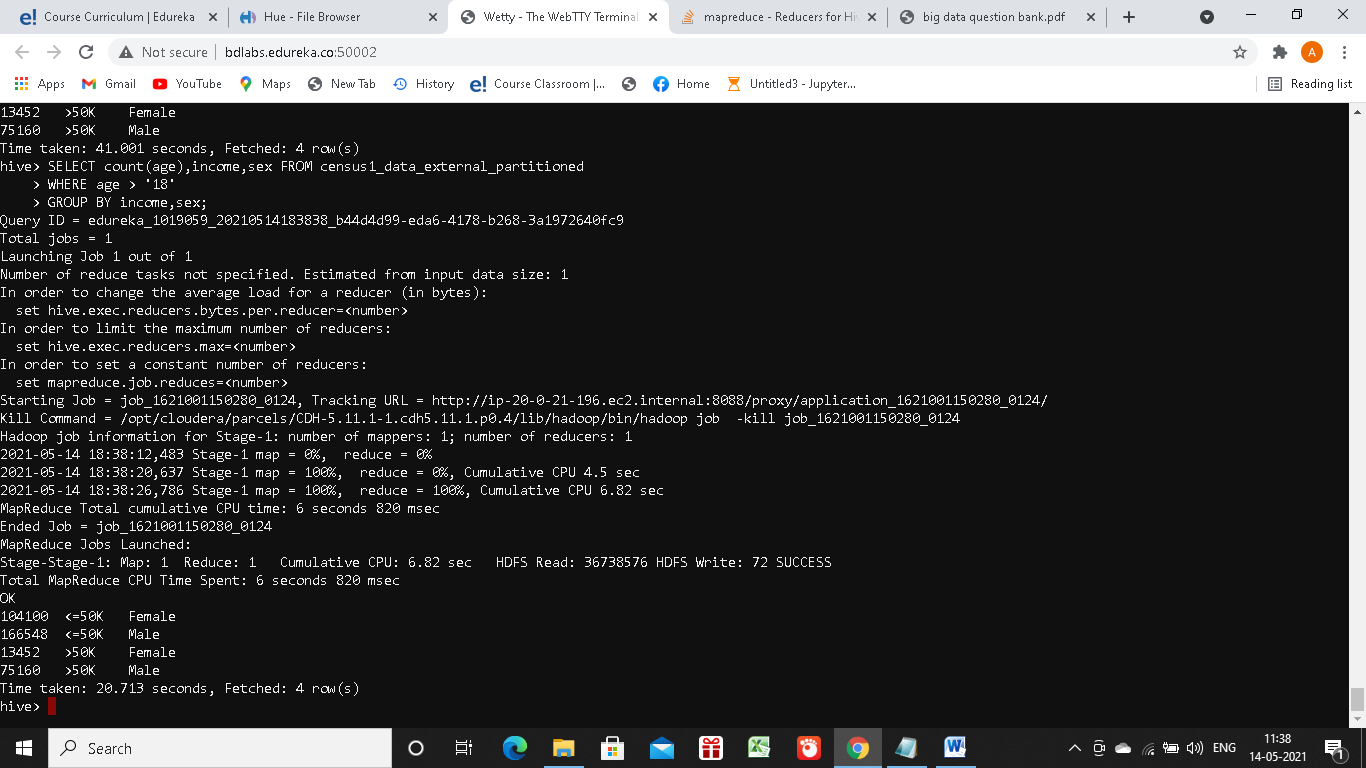
**external table: 41.001seconds**

Select count(age),sex,income From census1\_data \_partitioned WHERE age>’18’ GROUP BY income,sex;

****

**external table partitioned: 20.713seconds**

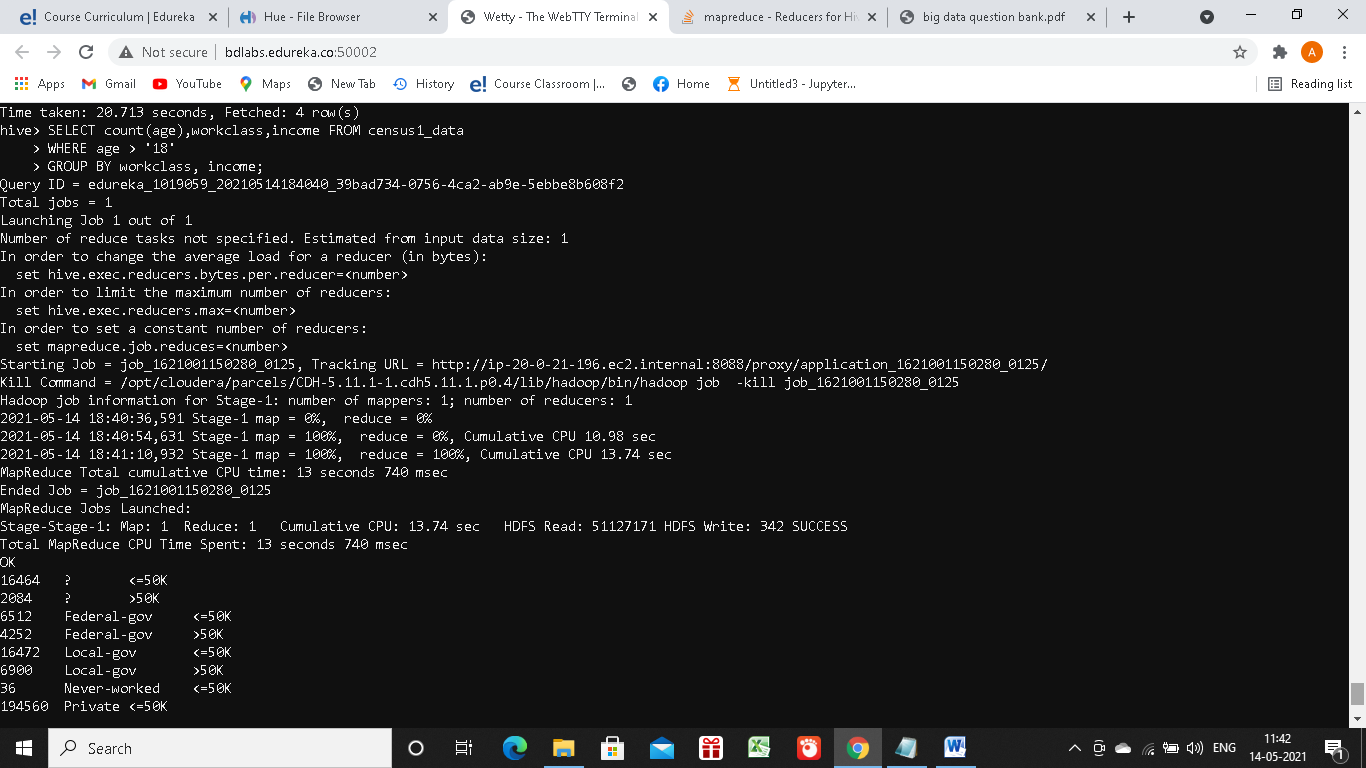
Select count(age),sex,income From census1\_data \_ external\_partitioned WHERE age>’18’ GROUP BY income,sex;

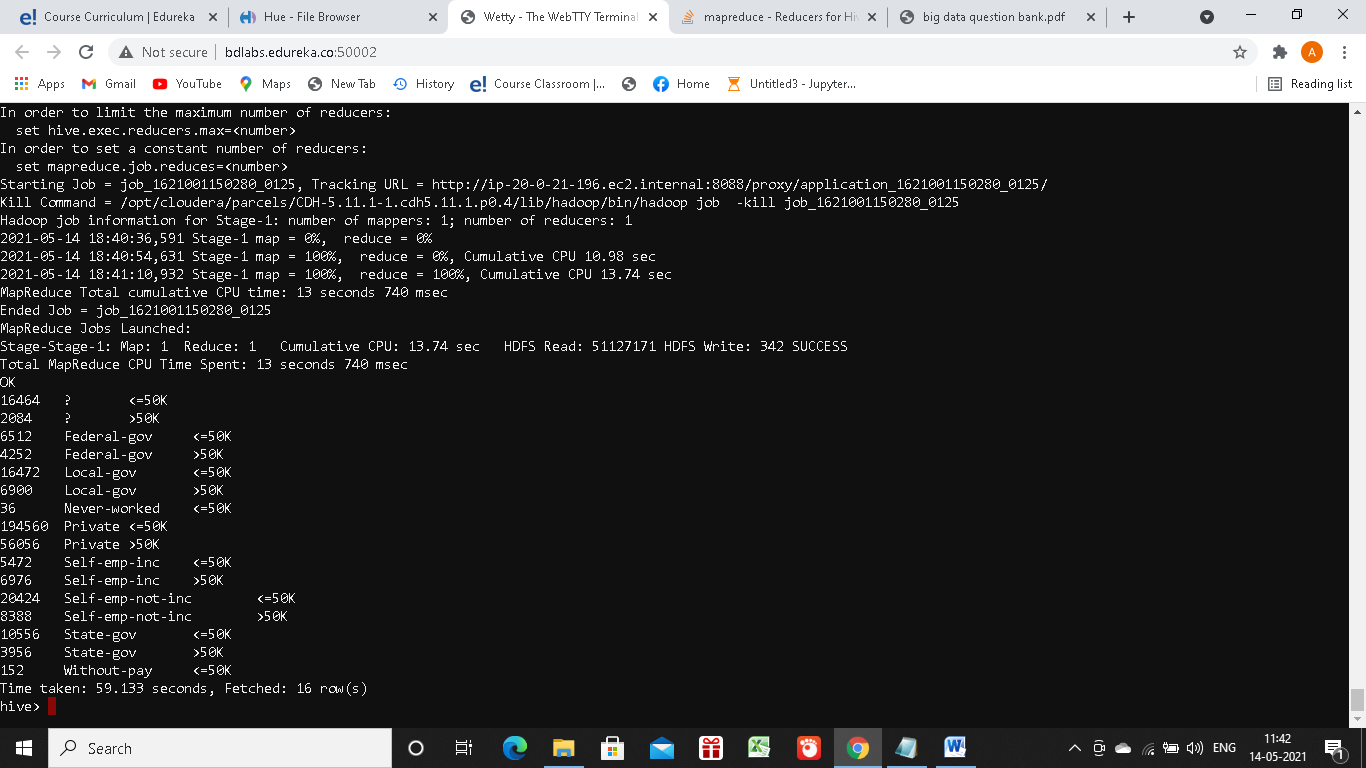


b Find out the number of adults based on income and workclass. Note the time taken for getting the result

**internal table non-partitioned: 59.133seconds**

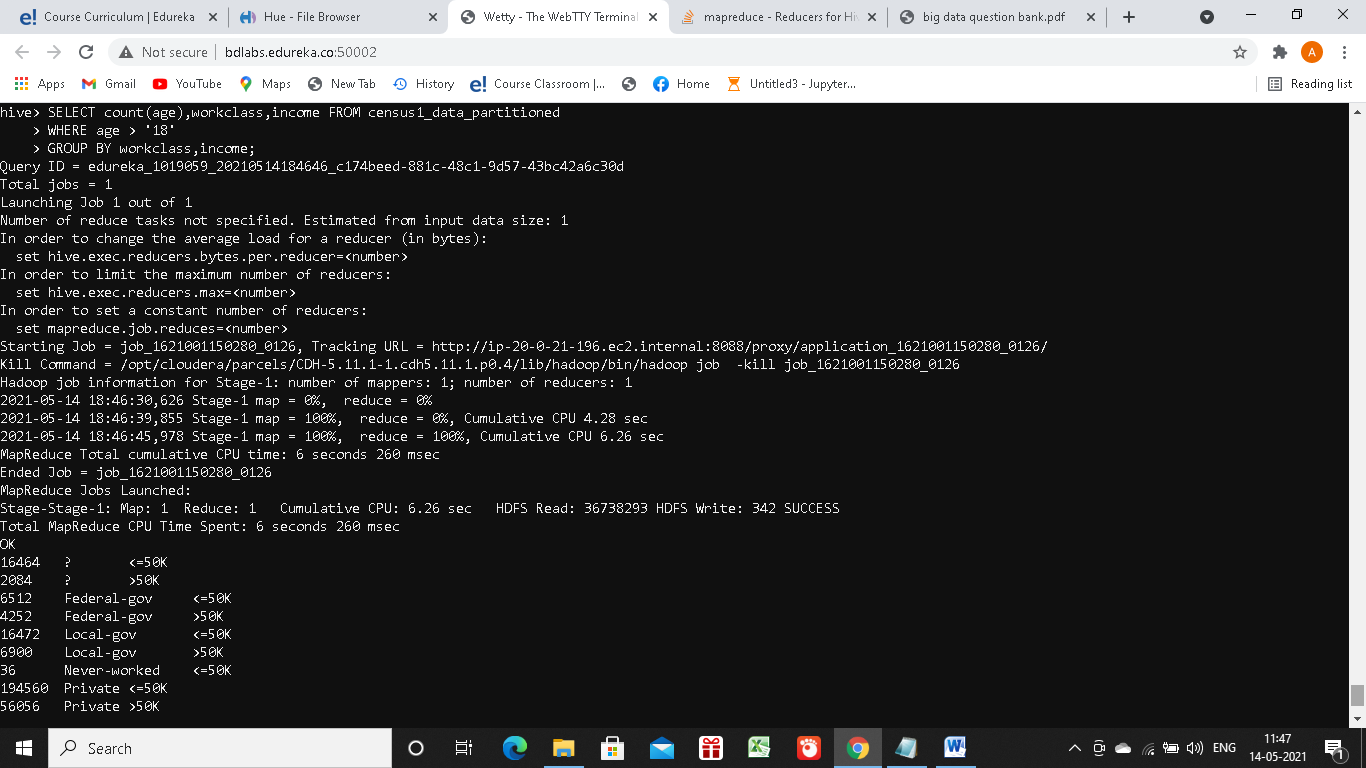
Select count(age),workclass,income From census1\_data WHERE age>’18’ GROUP BY income, workclass;

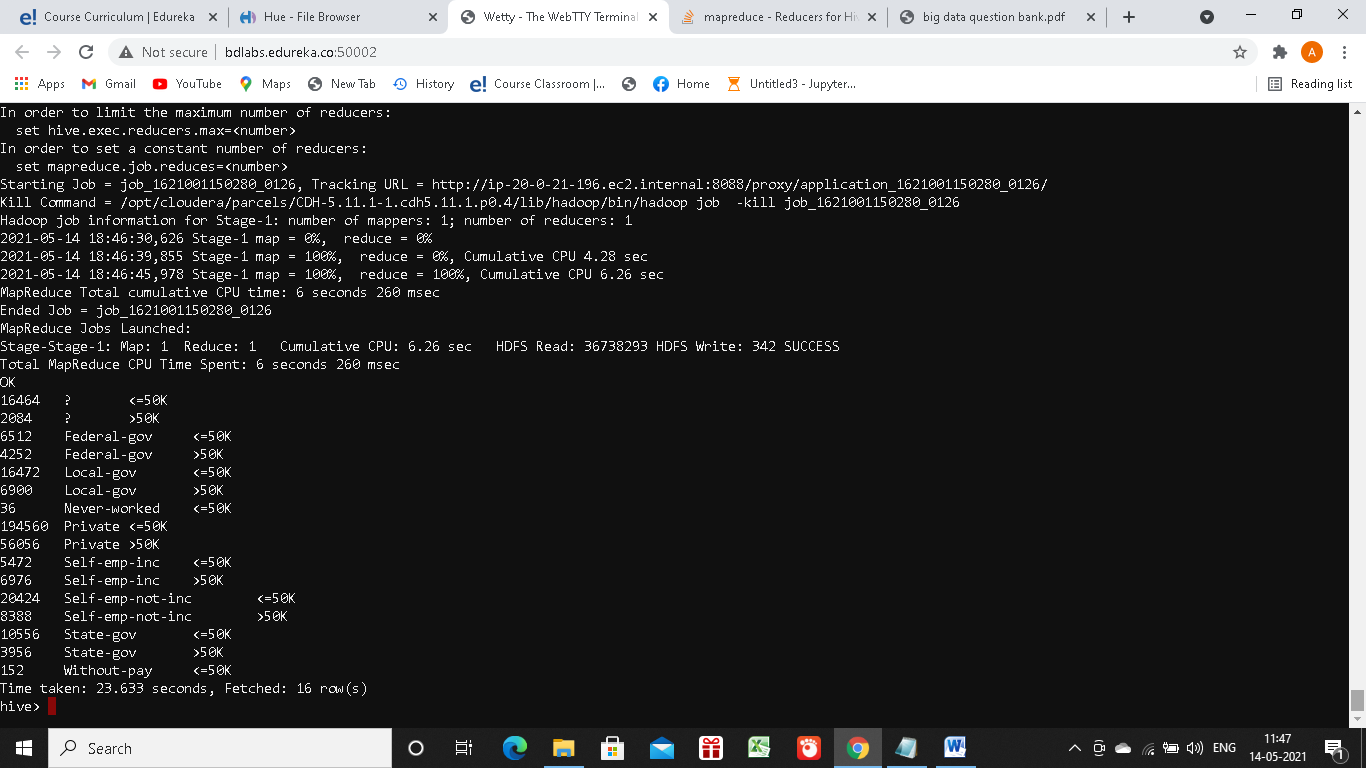




**internal table partitioned:23.633 seconds**

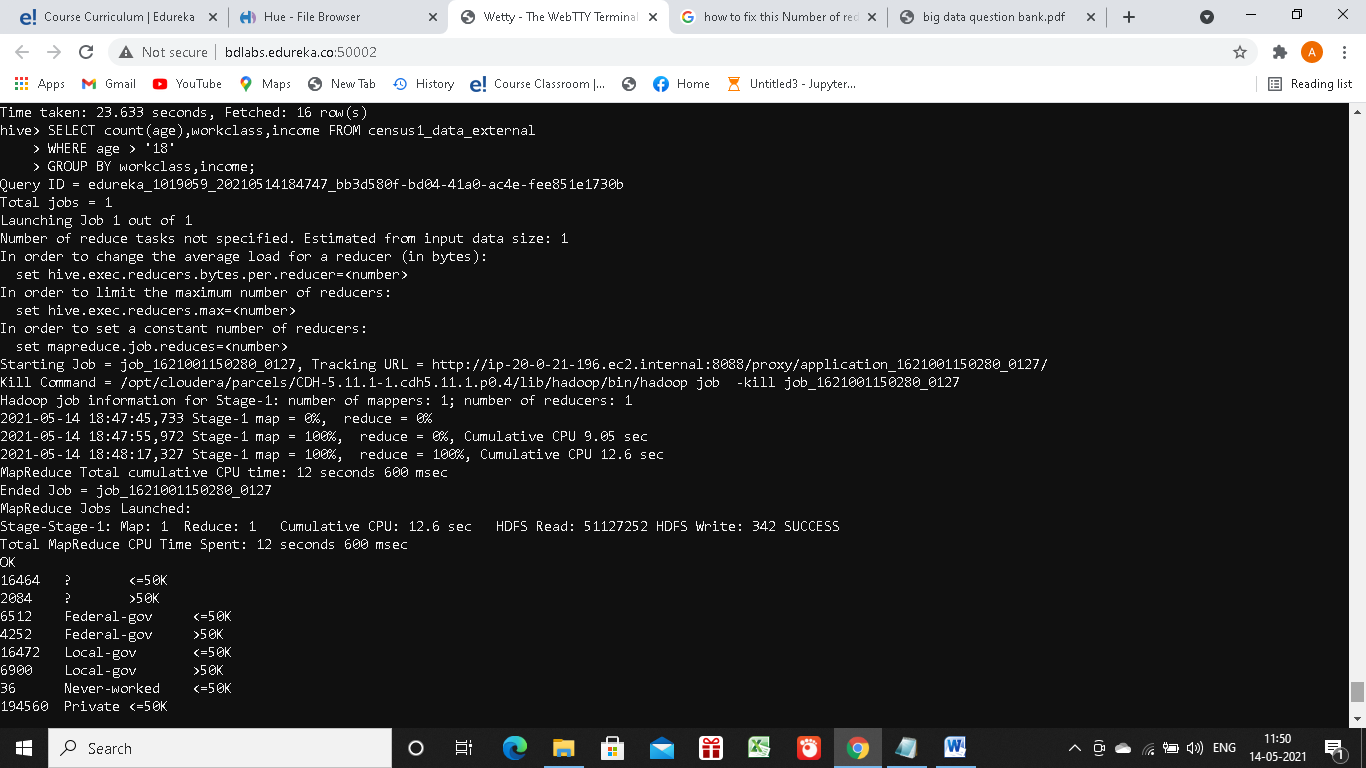
Select count(age),workclass,income From census1\_partitioned WHERE age>’18’ GROUP BY income, workclass;

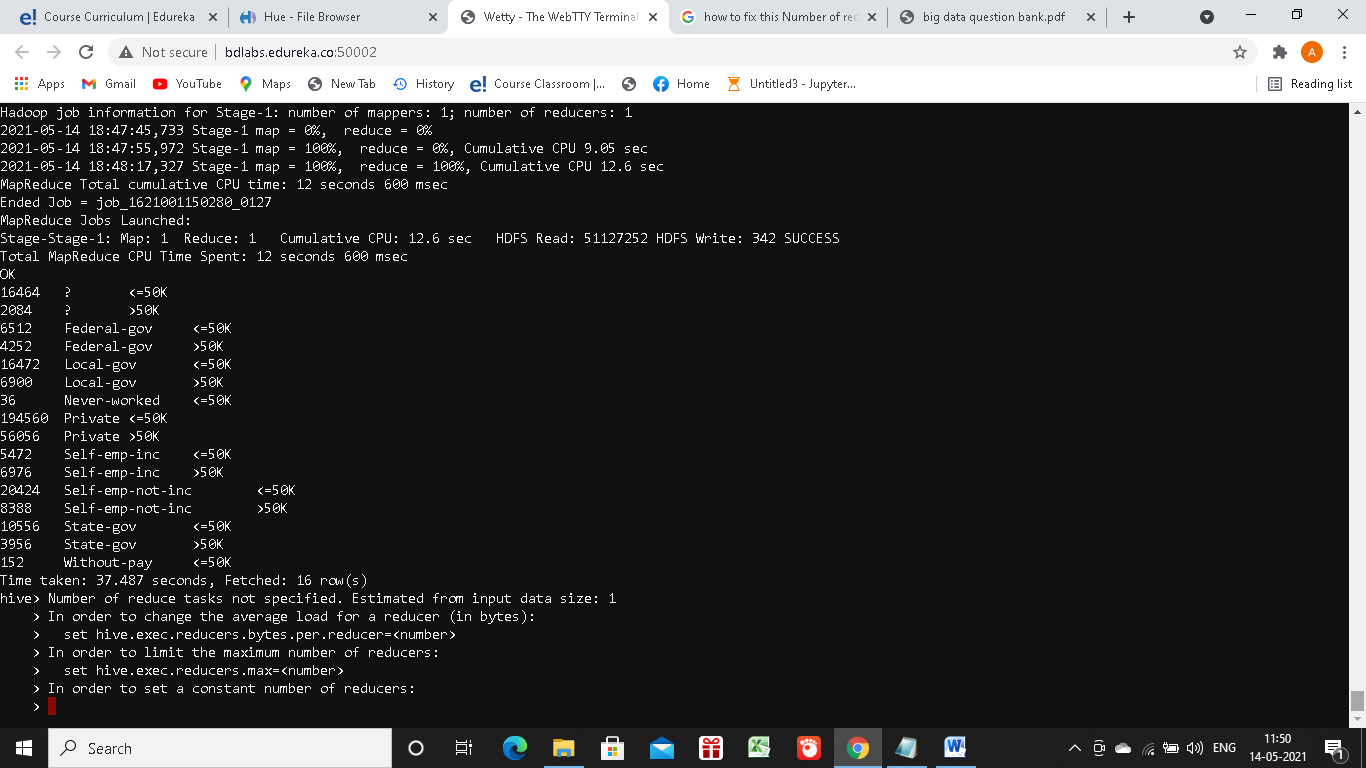




**external table non-partitioned:37.487 seconds**

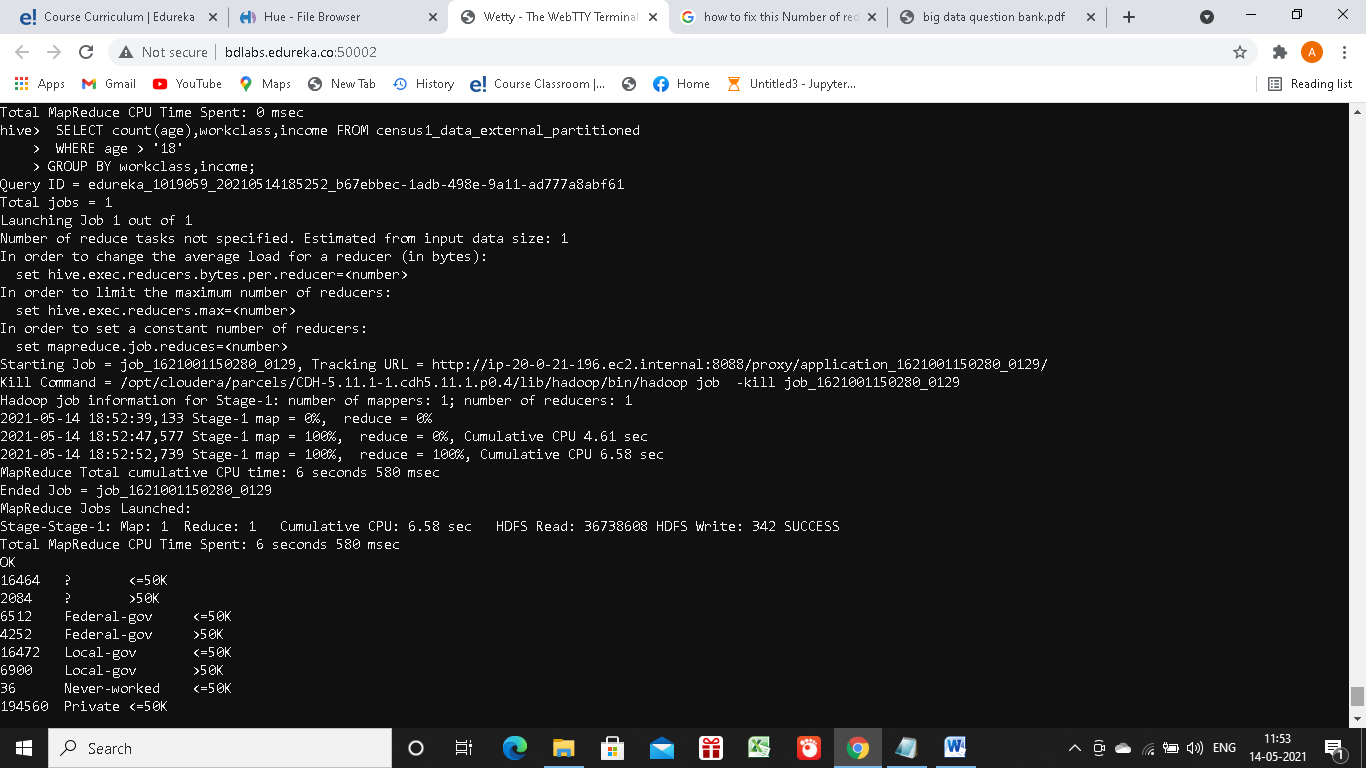
Select count(age),workclass,income From census1\_data \_external WHERE age>’18’ GROUP BY income, workclass;

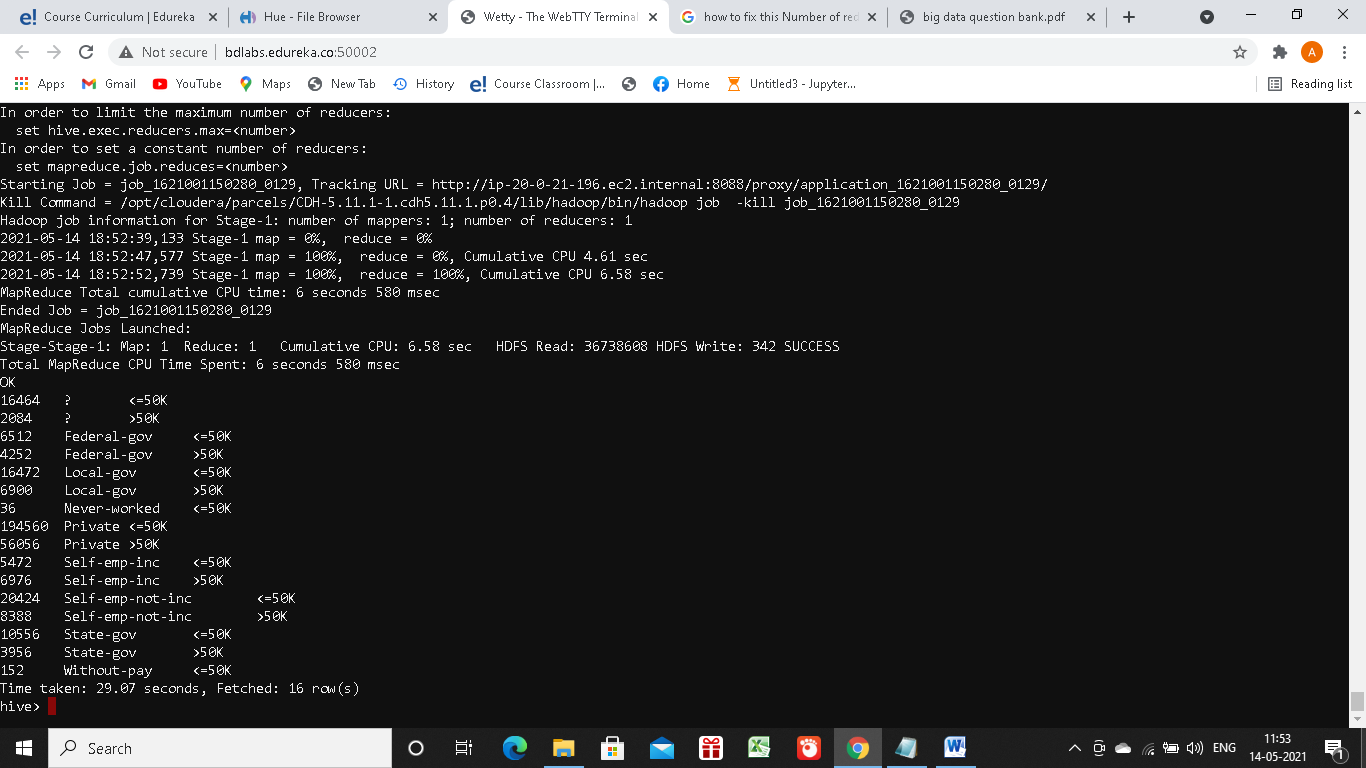




**external table partitioned:20.816 seconds**

Select count(age),workclass,income From census1\_data \_external \_partitioned WHERE age>’18’ GROUP BY income, workclass;





Comment : When we use Partition column in Query, that reduces query time significantly in case of partitioned tables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OBSERVATIONS IN BOTH QUERIES WRT TIME TAKEN IN THE 4 TABLES IN SECONDS | | | | |
| Query | Internal Table | Internal Partitioned Table | External Table | External Partitioned Table |
| Query1 | **141.919** | **92.114** | **41.001** | **28.787** |
| Query2 | **59.133** | **23.633** | **37.487** | **20.816** |

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 census1\_data;

Drop census1\_data\_external;

External Table: Whenever **we drop** the **external table**, then only the metadata associated with the **table** will get deleted, the **table** data remains untouched by **Hive**. It doesn't **delete** the **external** data.

Interanal Table: **Dropping** the **internal table** will delete the **table** data, as well as the metadata associated with the **table**. **Internal table** are like normal database **table** where data can be stored and queried on. On dropping these **tables** the data stored in them also gets deleted and data is lost forever.

External Table: