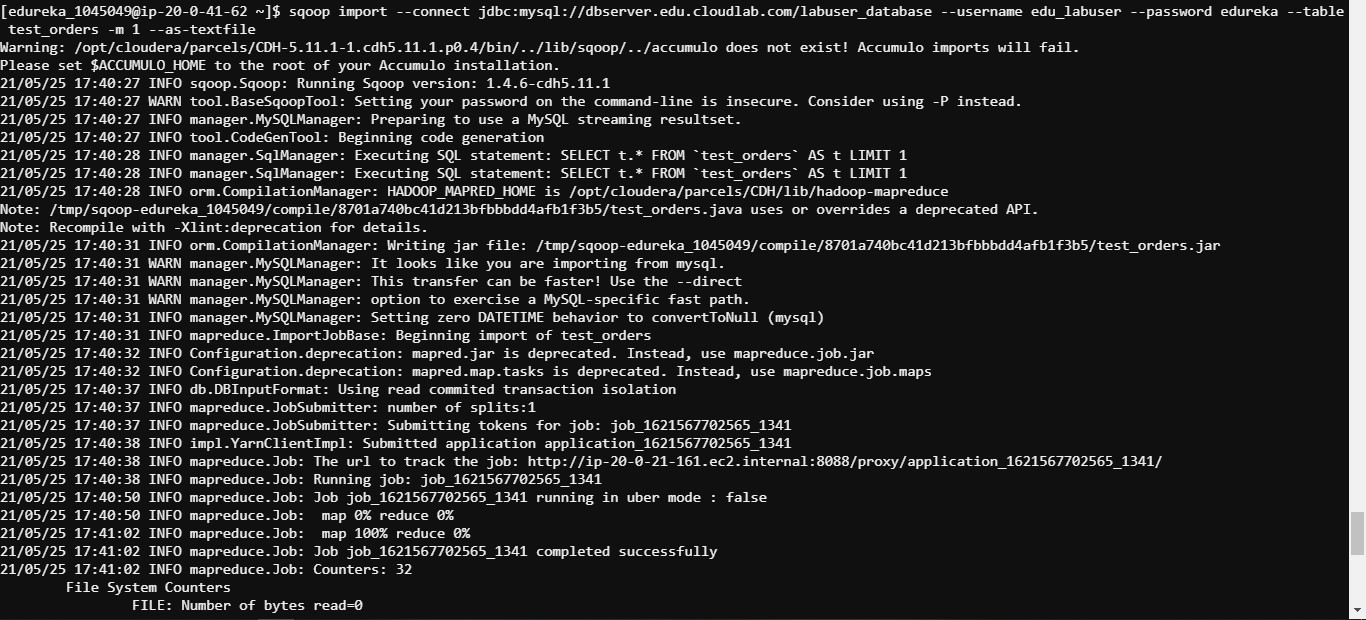
Q1 There are two tables named *test\_orders* and *test\_order\_items*, in MYSQL database, perform the following operation:

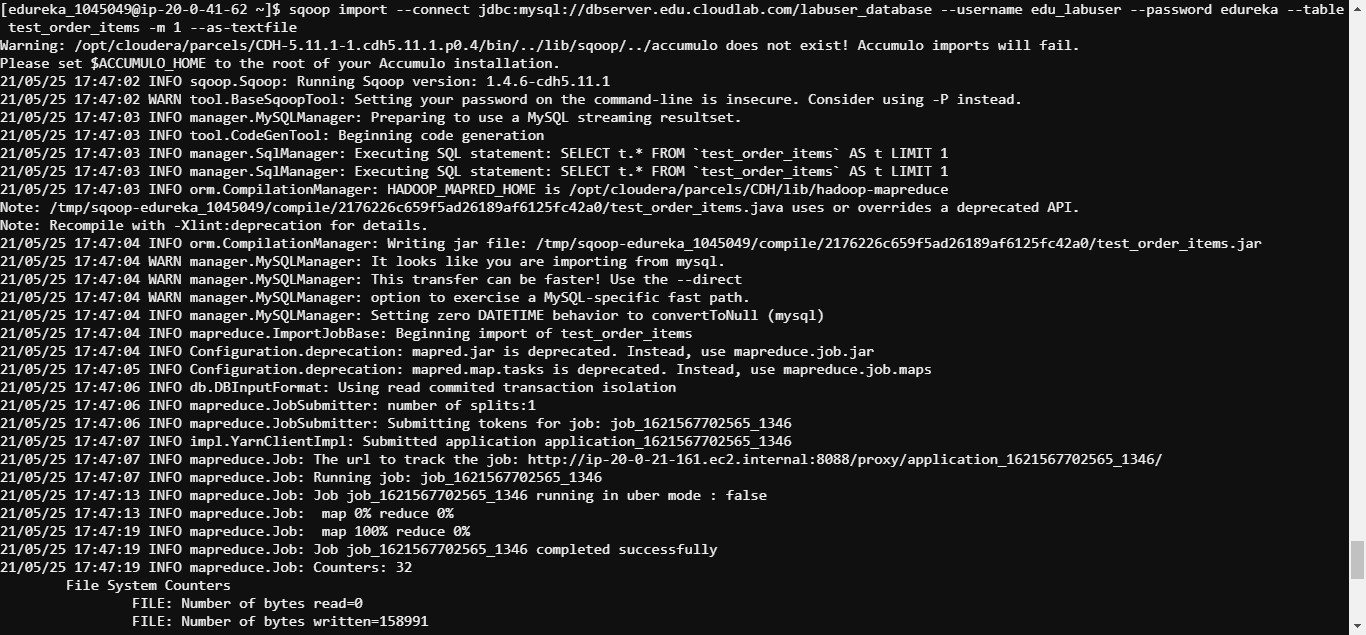
1. Import the table *test\_orders* into HDFS as a text file
2. Import the table *test\_order\_items* into HDFS as a text file
3. Import only those record from table *test\_orders* where the order status is ‘delivered’
4. Import only those record from the table *test\_order\_items* where the product price is less than 200

Output Format: Paste the commands used to solve this problem in the text box below. Also, attach a screenshot(s) of the code.

Ans1 sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --table test\_orders -m 1 --as-textfile

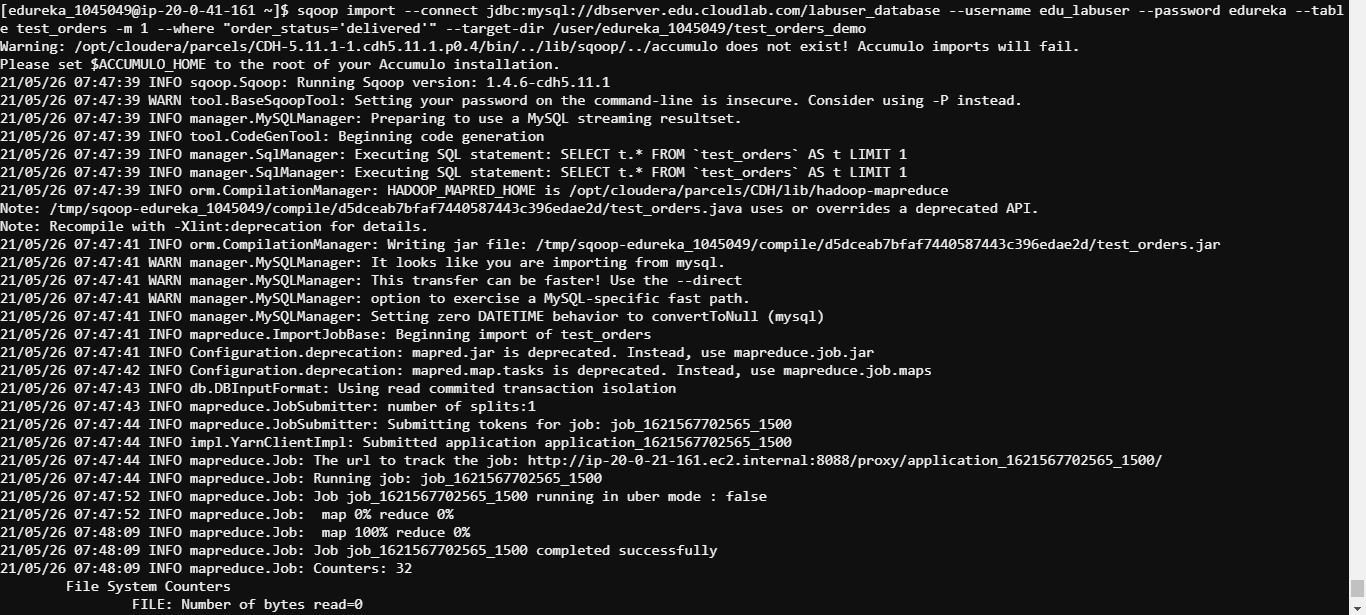


sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --table test\_order\_items -m 1 --as-textfile



sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --tabl

e test\_orders -m 1 --where "order\_status='delivered'" --target-dir /user/edureka\_1045049/test\_orders\_demo



sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --tabl

e test\_order\_items -m 1 --where "product\_price<200" --target-dir /user/edureka\_1045049/test\_order\_items\_demo



Q2 From the two tables mentioned in question 1, perform the following operations:

1. Write an import join statement, joining the tables *test\_orders* and *test\_order\_items* on *order\_id* and import the data in HDFS
2. Import the table *test\_orders* in append mode in and with 4 mappers
3. Export the result obtained from step 2 into a table in MySQL

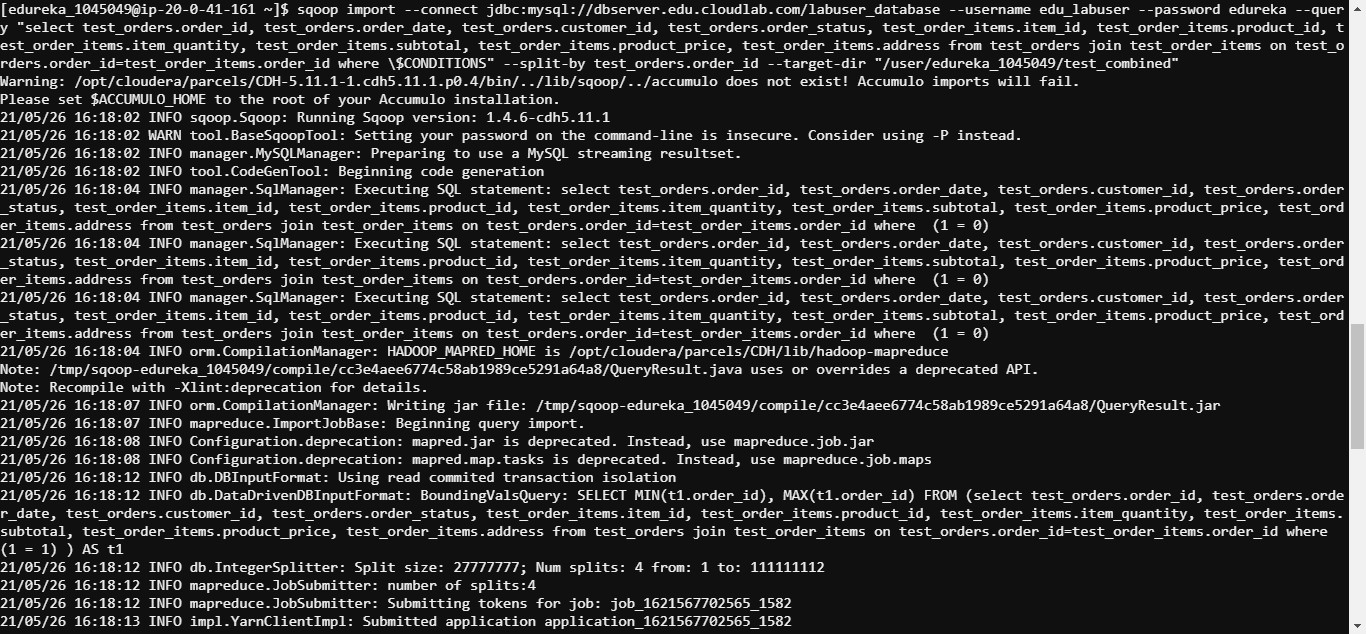
Output Format: Paste the commands used to solve this problem in the text box below. Also, attach a screenshot(s) of the code.

Ans2 sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --quer

y "select test\_orders.order\_id, test\_orders.order\_date, test\_orders.customer\_id, test\_orders.order\_status, test\_order\_items.item\_id, test\_order\_items.product\_id, t

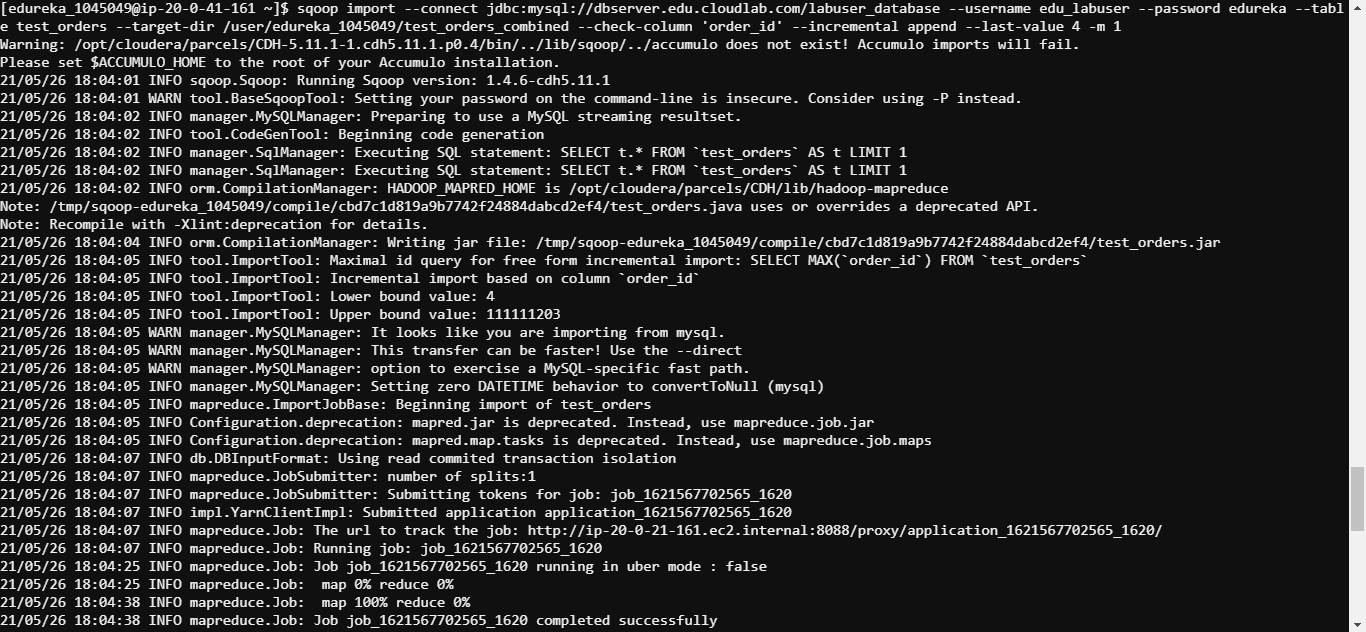
est\_order\_items.item\_quantity, test\_order\_items.subtotal, test\_order\_items.product\_price, test\_order\_items.address from test\_orders join test\_order\_items on test\_o

rders.order\_id=test\_order\_items.order\_id where \$CONDITIONS" --split-by test\_orders.order\_id --target-dir "/user/edureka\_1045049/test\_combined"



sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --tabl

e test\_orders --target-dir /user/edureka\_1045049/test\_orders\_combined --check-column 'order\_id' --incremental append --last-value 4 -m 1



sqoop eval --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --query "CREATE TABLE orders\_append\_test (order\_id

int(11), order\_date varchar(95), customer\_id int(11), order\_status varchar(50))" --username edu\_labuser --password edureka

mysql -u edu\_labuser -pedureka -h dbserver.edu.cloudlab.com

desc labuser\_database.orders\_append\_test;



sqoop export --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --tabl

e orders\_append\_test --export-dir /user/edureka\_1045049/test\_orders\_combined -m 1



Q3 From the two tables mentioned in question 1, perform the following operations:

1. Add a few rows in both the tables *test\_orders* and *test\_order\_items*, import these newly added rows in HDFS using the incremental append method
2. Create a hive table and import the table *test\_orders* directly in this hive table using sqoop hive import

Output Format: Paste the commands used to solve this problem in the text box below. Also, attach a screenshot(s) of the code.

Ans3 insert into test\_orders (order\_id, order\_date, customer\_id, order\_status) values (1328, '2021-05-27 17:05:09', 236791, 'pending');

insert into test\_orders (order\_id, order\_date, customer\_id, order\_status) values (1329, '2021-05-27 17:15:09', 295891, 'delivered');

insert into test\_orders (order\_id, order\_date, customer\_id, order\_status) values (1330, '2021-05-27 17:25:09', 156829, 'delayed');

insert into test\_orders (order\_id, order\_date, customer\_id, order\_status) values (1331, '2021-05-27 17:35:09', 568349, 'cancelled');

insert into test\_order\_items (item\_id, order\_id, product\_id, item\_quantity, subtotal, product\_price, address) values (56298, 1331, 832059

, 5, 177.5, 35.5, 'Chennai');

insert into test\_order\_items (item\_id, order\_id, product\_id, item\_quantity, subtotal, product\_price, address) values (95208, 1330, 724186

, 3, 3000, 1000, 'Bengaluru');

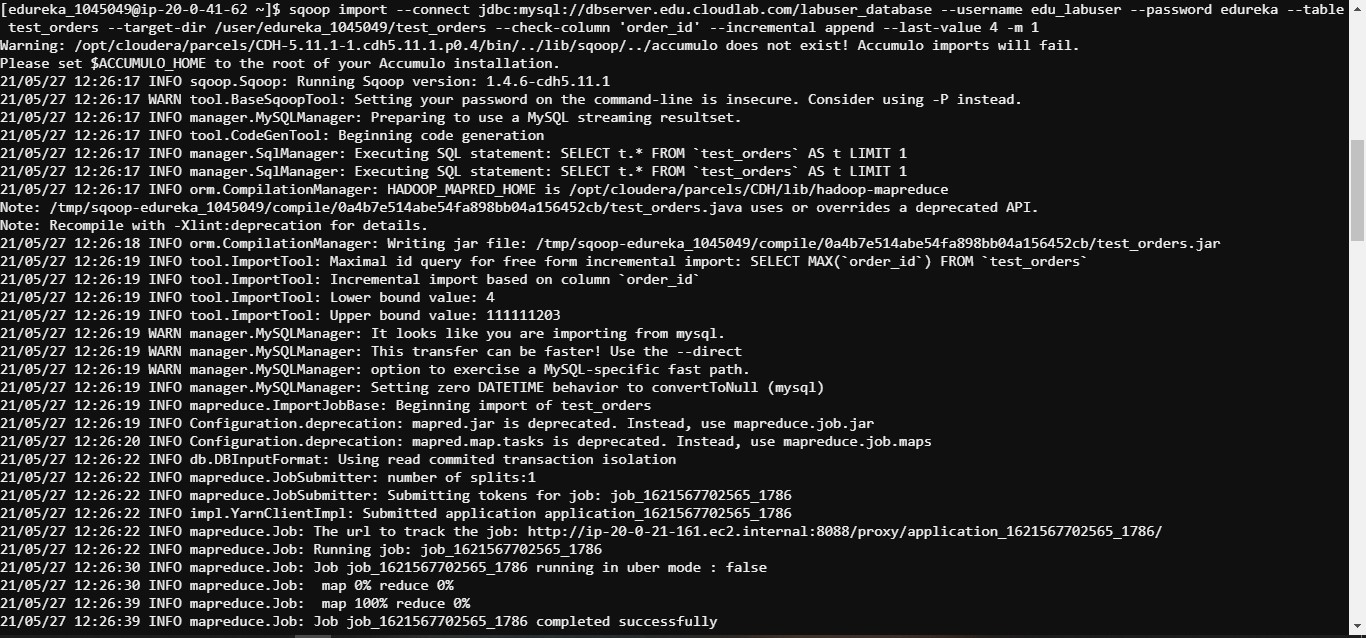
insert into test\_order\_items (item\_id, order\_id, product\_id, item\_quantity, subtotal, product\_price, address) values (32185, 1329, 423091

, 8, 2000, 250, 'Noida');

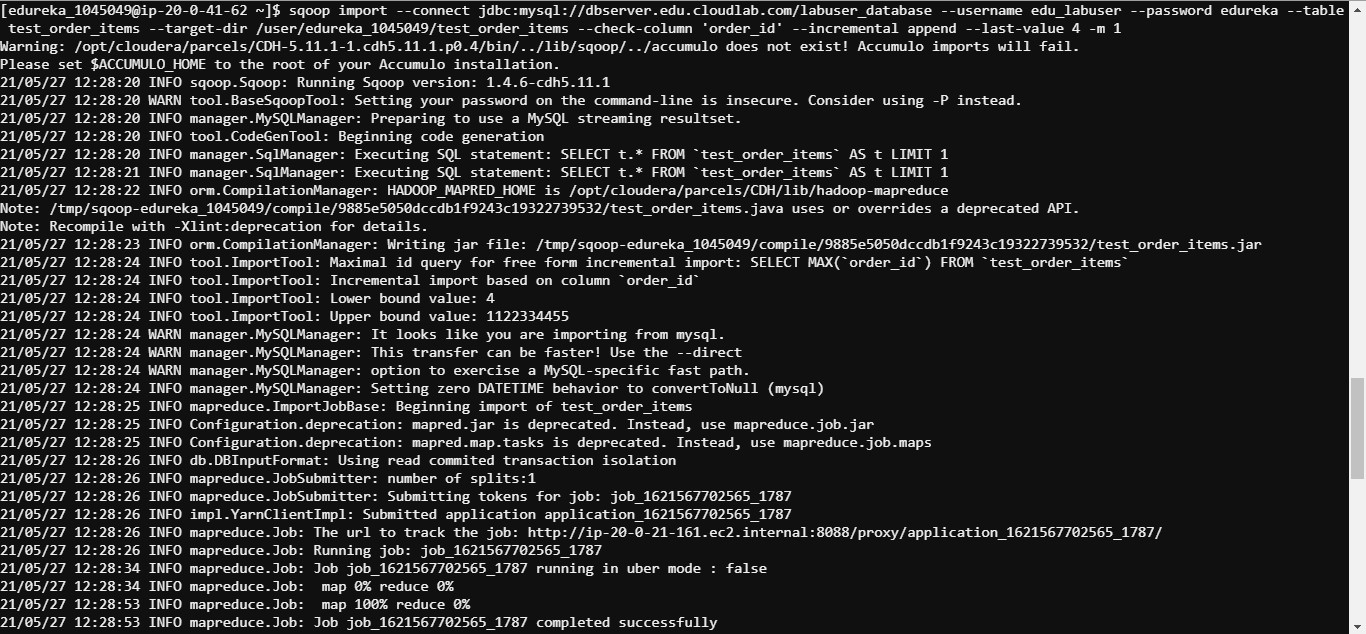
insert into test\_order\_items (item\_id, order\_id, product\_id, item\_quantity, subtotal, product\_price, address) values (37802, 1328, 925106

, 100, 5060, 50.6, 'Mumbai');

sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --table test\_orders --target-dir /user/edureka\_1045049/test\_orders --check-column 'order\_id' --incremental append --last-value 4 -m 1

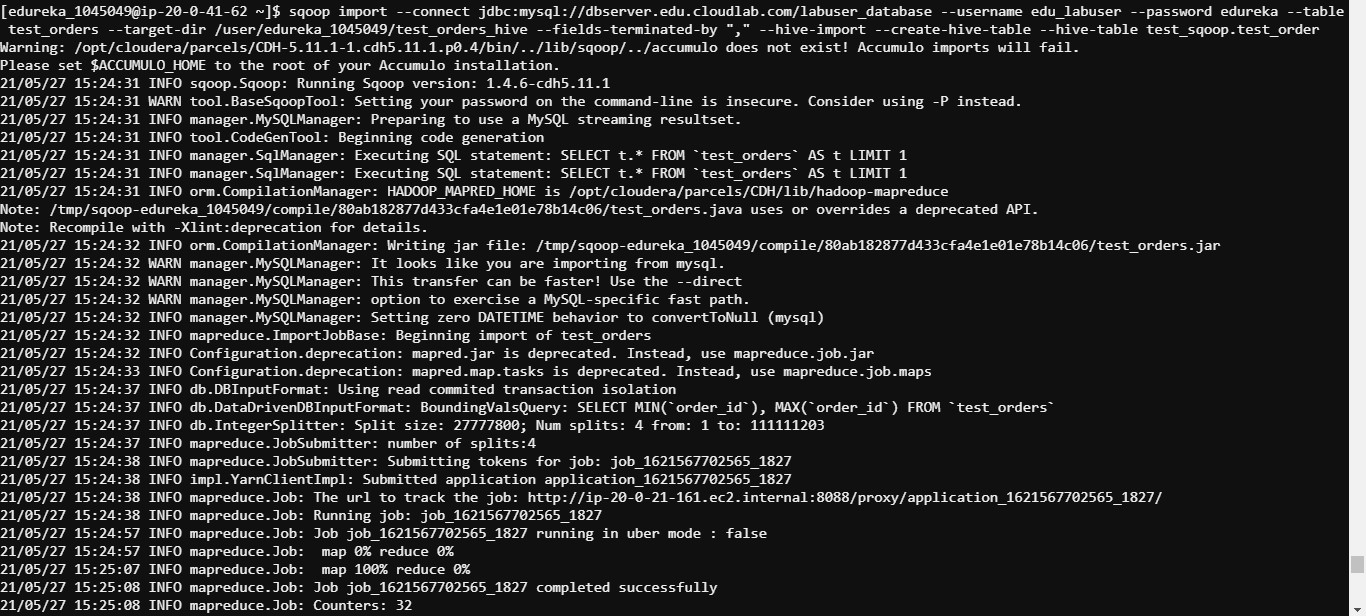


sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --table test\_order\_items --target-dir /user/edureka\_1045049/test\_order\_items --check-column 'order\_id' --incremental append --last-value 4 -m 1



sqoop import --connect jdbc:mysql://dbserver.edu.cloudlab.com/labuser\_database --username edu\_labuser --password edureka --table

test\_orders --target-dir /user/edureka\_1045049/test\_orders\_hive --fields-terminated-by "," --hive-import --create-hive-table --hive-table test\_sqoop.test\_order



Q4 *Dataset: World\_Bank\_Indicators\_noheader.csv*

Using the dataset present at the HDFS path

'hdfs:///bigdatads/common\_folder/assignment2/World\_Bank\_Indicators\_noheader.csv', perform the following activities on Hive:

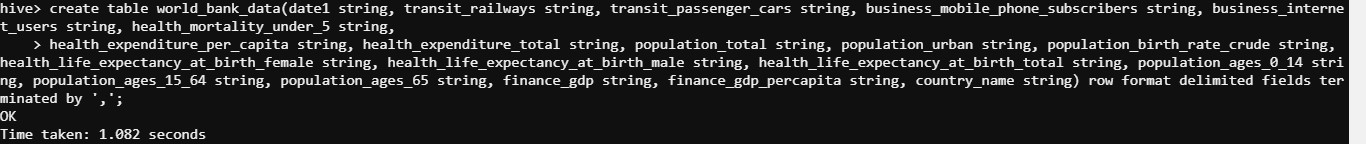
1. Create a hive internal table named *world\_bank\_data* which is not partitioned
2. Create one more (internal) table named *world\_bank\_data\_partitioned* partitioned on *Country\_Name*
3. Insert the dataset given in table *world\_bank\_data*
4. Insert data in table *world\_bank\_data\_partitioned* from table *world\_bank\_data* using dynamic partitioning

Output Format: Paste the commands used to solve this problem in the text box below. Also, attach a screenshot(s) of the code.

Ans4

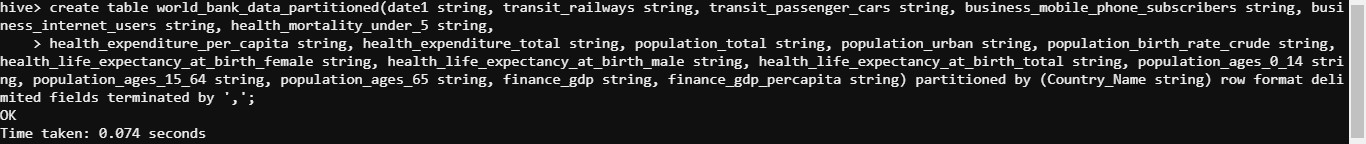
create table world\_bank\_data(date1 string, transit\_railways string, transit\_passenger\_cars string, business\_mobile\_phone\_subscribers string, business\_internet\_users string, health\_mortality\_under\_5 string,

health\_expenditure\_per\_capita string, health\_expenditure\_total string, population\_total string, population\_urban string, population\_birth\_rate\_crude string, health\_life\_expectancy\_at\_birth\_female string, health\_life\_expectancy\_at\_birth\_male string, health\_life\_expectancy\_at\_birth\_total string, population\_ages\_0\_14 string, population\_ages\_15\_64 string, population\_ages\_65 string, finance\_gdp string, finance\_gdp\_percapita string, country\_name string) row format delimited fields terminated by ',';



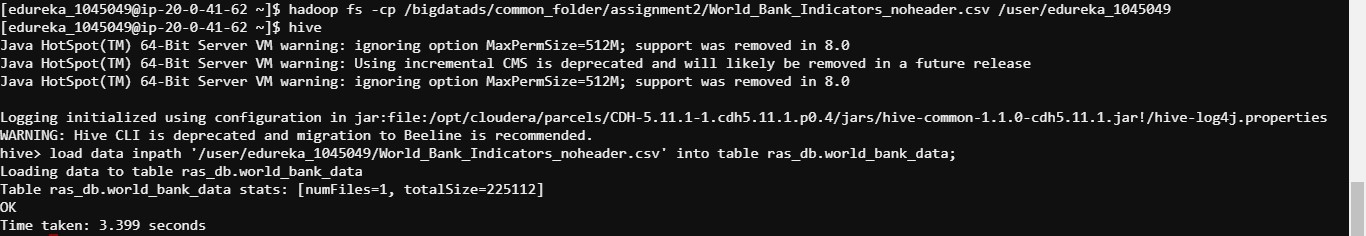
create table world\_bank\_data\_partitioned(date1 string, transit\_railways string, transit\_passenger\_cars string, business\_mobile\_phone\_subscribers string, business\_internet\_users string, health\_mortality\_under\_5 string,

health\_expenditure\_per\_capita string, health\_expenditure\_total string, population\_total string, population\_urban string, population\_birth\_rate\_crude string, health\_life\_expectancy\_at\_birth\_female string, health\_life\_expectancy\_at\_birth\_male string, health\_life\_expectancy\_at\_birth\_total string, population\_ages\_0\_14 string, population\_ages\_15\_64 string, population\_ages\_65 string, finance\_gdp string, finance\_gdp\_percapita string) partitioned by (Country\_Name string) row format delimited fields terminated by ',';



hadoop fs -cp /bigdatads/common\_folder/assignment2/World\_Bank\_Indicators\_noheader.csv /user/edureka\_1045049

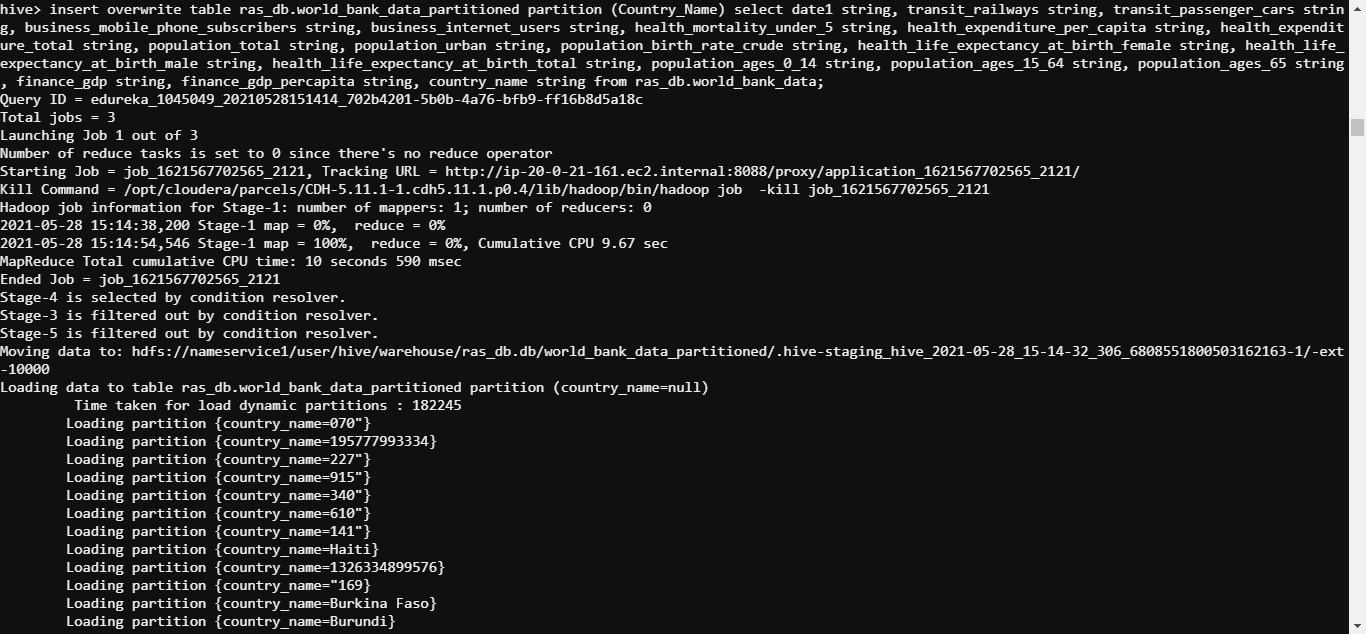
load data inpath '/user/edureka\_1045049/World\_Bank\_Indicators\_noheader.csv' into table ras\_db.world\_bank\_data;



SET hive.exec.max.dynamic.partitions=100000;

SET hive.exec.max.dynamic.partitions.pernode=100000;

insert overwrite table ras\_db.world\_bank\_data\_partitioned partition (Country\_Name) select date1 string, transit\_railways string, transit\_passenger\_cars string, business\_mobile\_phone\_subscribers string, business\_internet\_users string, health\_mortality\_under\_5 string, health\_expenditure\_per\_capita string, health\_expenditure\_total string, population\_total string, population\_urban string, population\_birth\_rate\_crude string, health\_life\_expectancy\_at\_birth\_female string, health\_life\_expectancy\_at\_birth\_male string, health\_life\_expectancy\_at\_birth\_total string, population\_ages\_0\_14 string, population\_ages\_15\_64 string, population\_ages\_65 string, finance\_gdp string, finance\_gdp\_percapita string, country\_name string from ras\_db.world\_bank\_data;



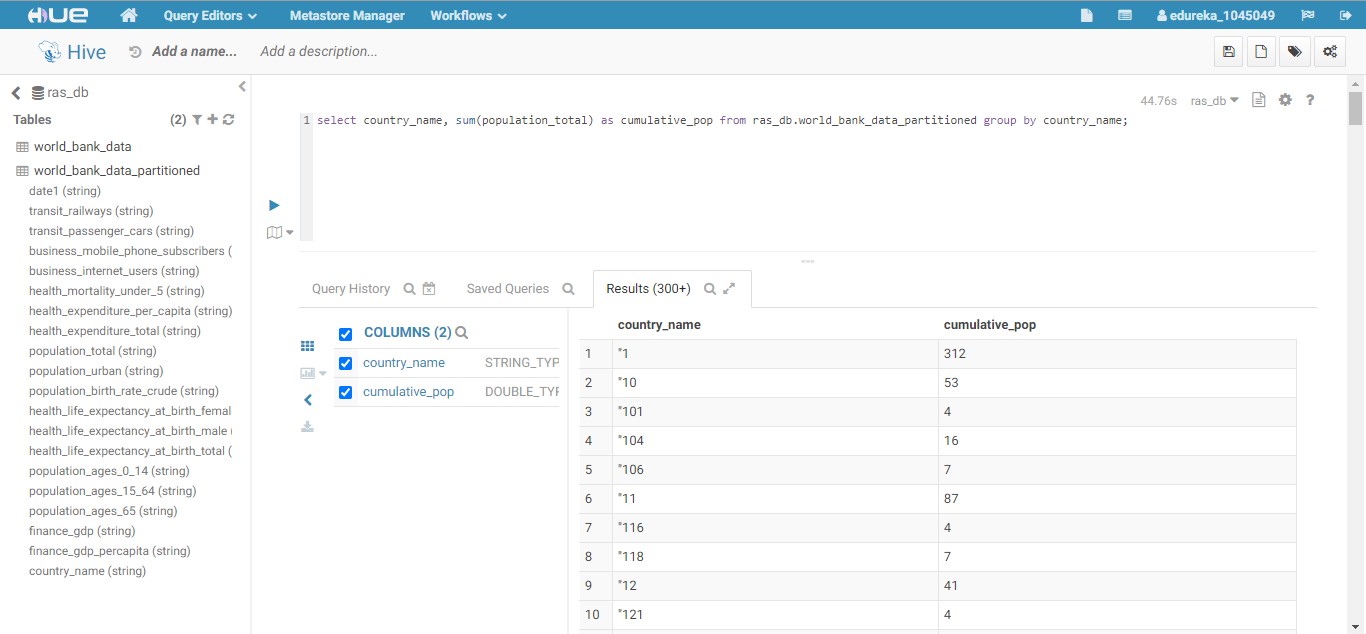
Q5 *Dataset: World\_Bank\_Indicators\_noheader.csv*

From the two tables created in question 4, perform the following operations in Hive:

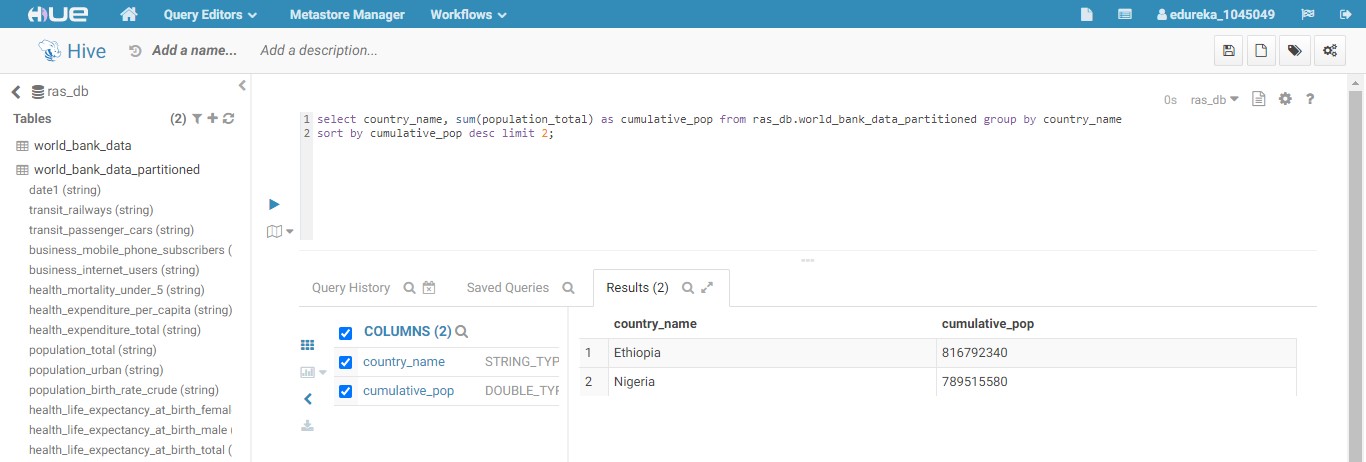
1. Find out the cumulative population of countries over the decade from the partitioned table
2. Find out the second most populous country from the partitioned table
3. Find out the population of the country Afghanistan from the table and compare the difference in timings taken by the query

Output Format: Paste the commands used to solve this problem in the text box below. Also, attach a screenshot(s) of the code.

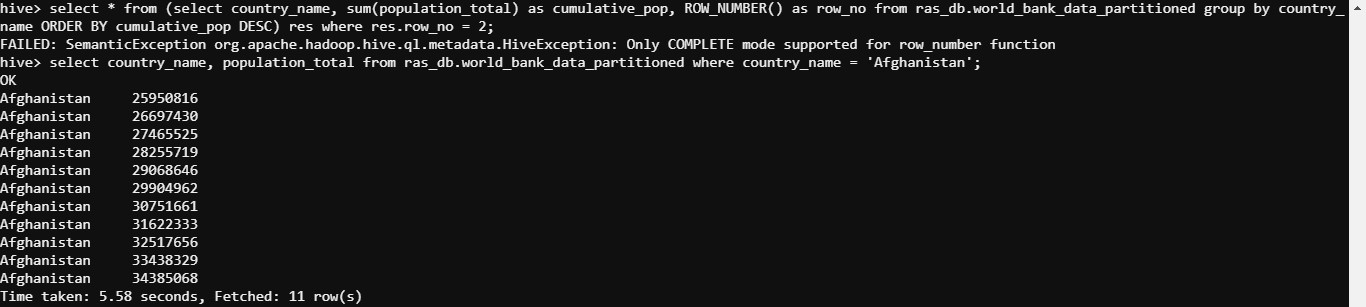
Ans5 select country\_name, sum(population\_total) as cumulative\_pop from ras\_db.world\_bank\_data\_partitioned group by country\_name;



select country\_name, sum(population\_total) as cumulative\_pop from ras\_db.world\_bank\_data\_partitioned group by country\_name sort by cumulative\_pop desc limit 2;



select country\_name, population\_total from ras\_db.world\_bank\_data\_partitioned where country\_name = 'Afghanistan';



select country\_name, population\_total from ras\_db.world\_bank\_data where country\_name = 'Afghanistan';

