Assignment #3:

Create "customers" and "transactions" tables

Load the data from ~/futurence\_hadoop-pyspark/labs/dataset/retail

Perform the analysis mentioned below

1) No of transactions by customer

2) Total transaction amount by customer

3) Get top 3 customers by transaction amount

4) No of transactions by customer and mode of payment

5) Get top 3 cities which has more transactions

6) Get month wise highest transaction

7) Get sample transactions

0: jdbc:hive2://> CREATE TABLE Customers(cust\_id int,last\_name String,first\_name String,age int,profession String)

. . . . . . . . > comment "Customer details"

. . . . . . . . > ROW FORMAT DELIMITED

. . . . . . . . > FIELDS TERMINATED BY ","

. . . . . . . . > LINES TERMINATED BY "\n"

. . . . . . . . > STORED AS TEXTFILE;

OK

No rows affected (0.547 seconds)

0: jdbc:hive2://> load data local inpath '/home/tech/futurense\_hadoop-pyspark/labs/dataset/retail/customers.txt' overwrite into table customers;

0: jdbc:hive2://> CREATE TABLE Transactions (trans\_id int,trans\_date String, cust\_id int, amount double,category String,desc String,city String,state String,pymt\_mode String)

. . . . . . . . > comment "Transactions details"

. . . . . . . . > ROW FORMAT DELIMITED

. . . . . . . . > FIELDS TERMINATED BY ","

. . . . . . . . > LINES TERMINATED BY "\n"

. . . . . . . . > STORED AS TEXTFILE;

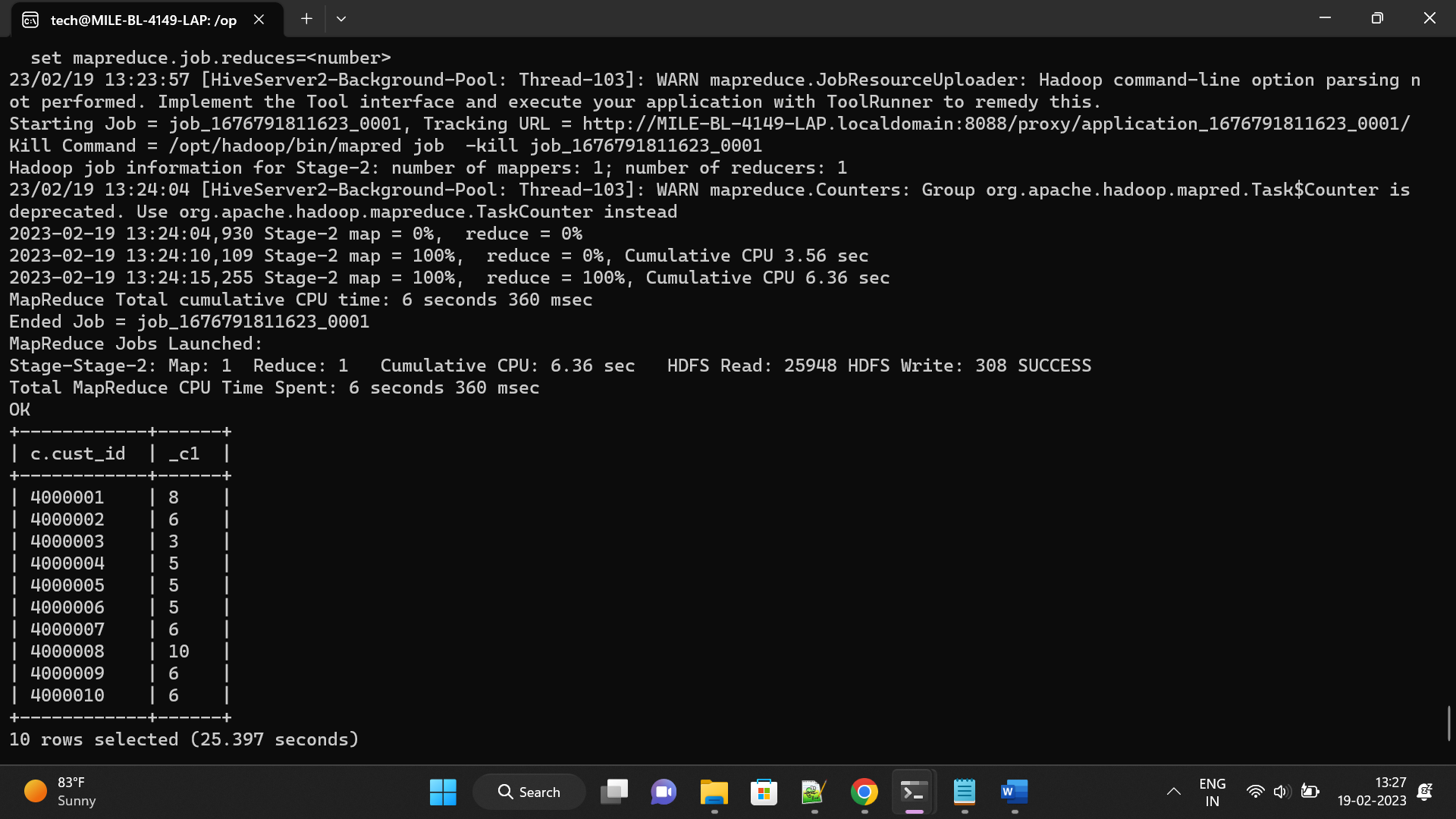
OK

No rows affected (0.071 seconds)

0: jdbc:hive2://> load data local inpath '/home/tech/futurense\_hadoop-pyspark/labs/dataset/retail/transactions.txt' overwrite into table transactions;

1) No of transactions by customer

0: jdbc:hive2://> SELECT c.cust\_id,count(t.trans\_id) FROM customers c join transactions t on c.cust\_id = t.cust\_id GROUP BY c.cust\_id;



2) Total transaction amount by customer

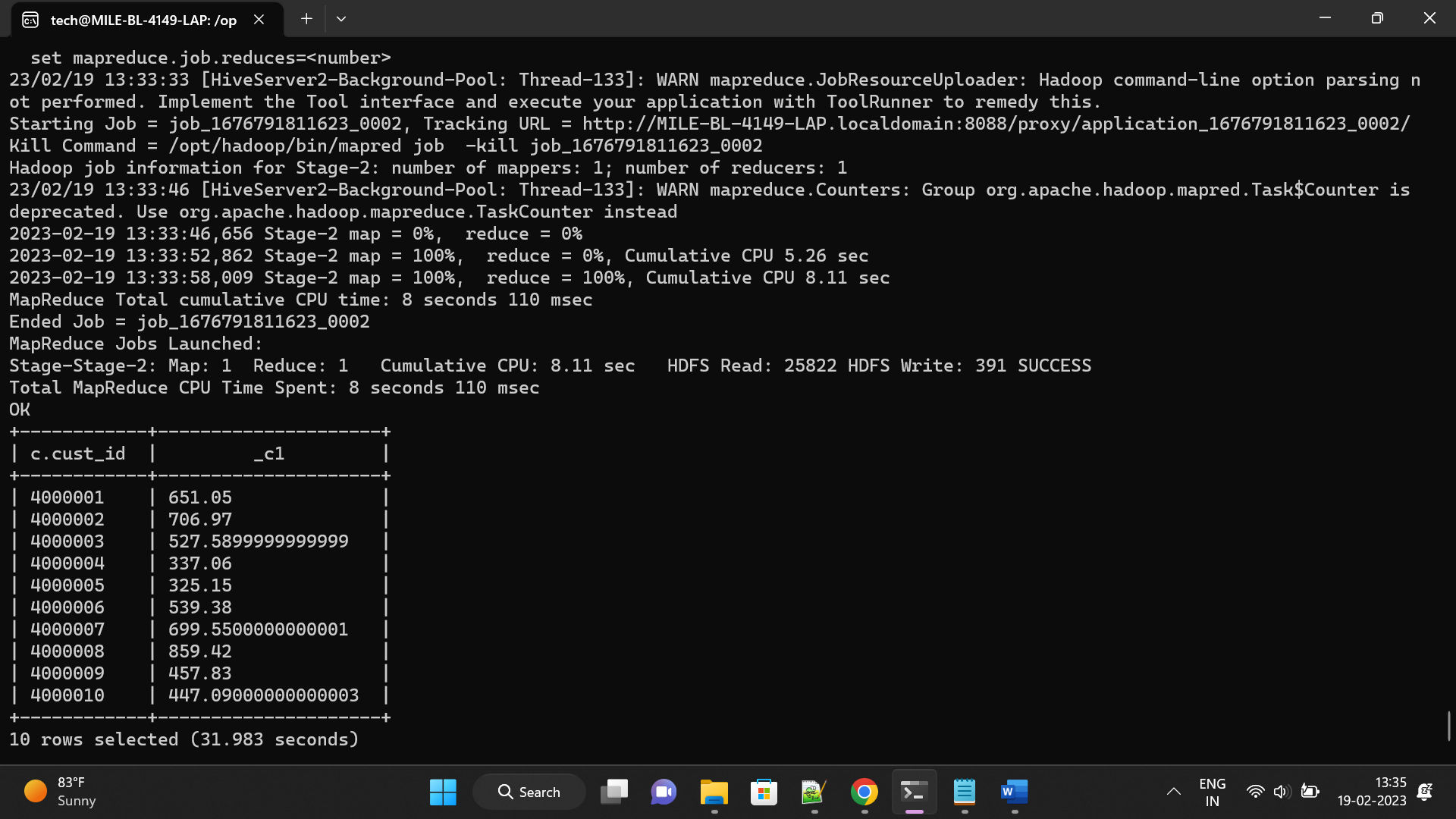
0: jdbc:hive2://> SELECT c.cust\_id, SUM(t.amount)

. . . . . . . . > FROM Customers c

. . . . . . . . > JOIN Transactions t

. . . . . . . . > ON c.cust\_id = t.cust\_id

. . . . . . . . > GROUP BY c.cust\_id;



3) Get top 3 customers by transaction amount

0: jdbc:hive2://> SELECT c.cust\_id, SUM(t.amount) AS Total\_Amount

. . . . . . . . > FROM Customers c

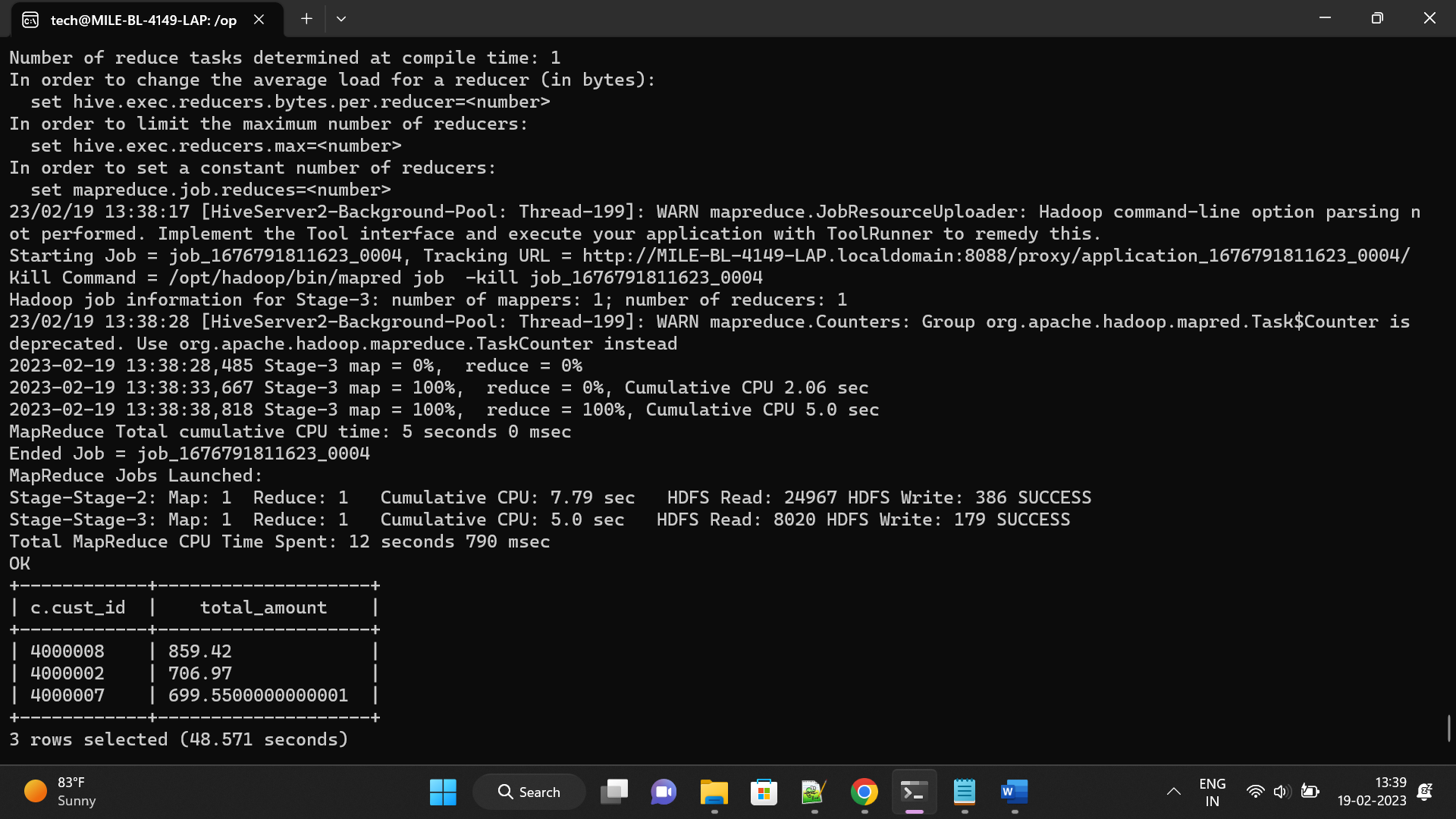
. . . . . . . . > INNER JOIN Transactions t

. . . . . . . . > ON c.cust\_id = t.cust\_id

. . . . . . . . > GROUP BY c.cust\_id

. . . . . . . . > ORDER BY Total\_Amount DESC

. . . . . . . . > LIMIT 3;

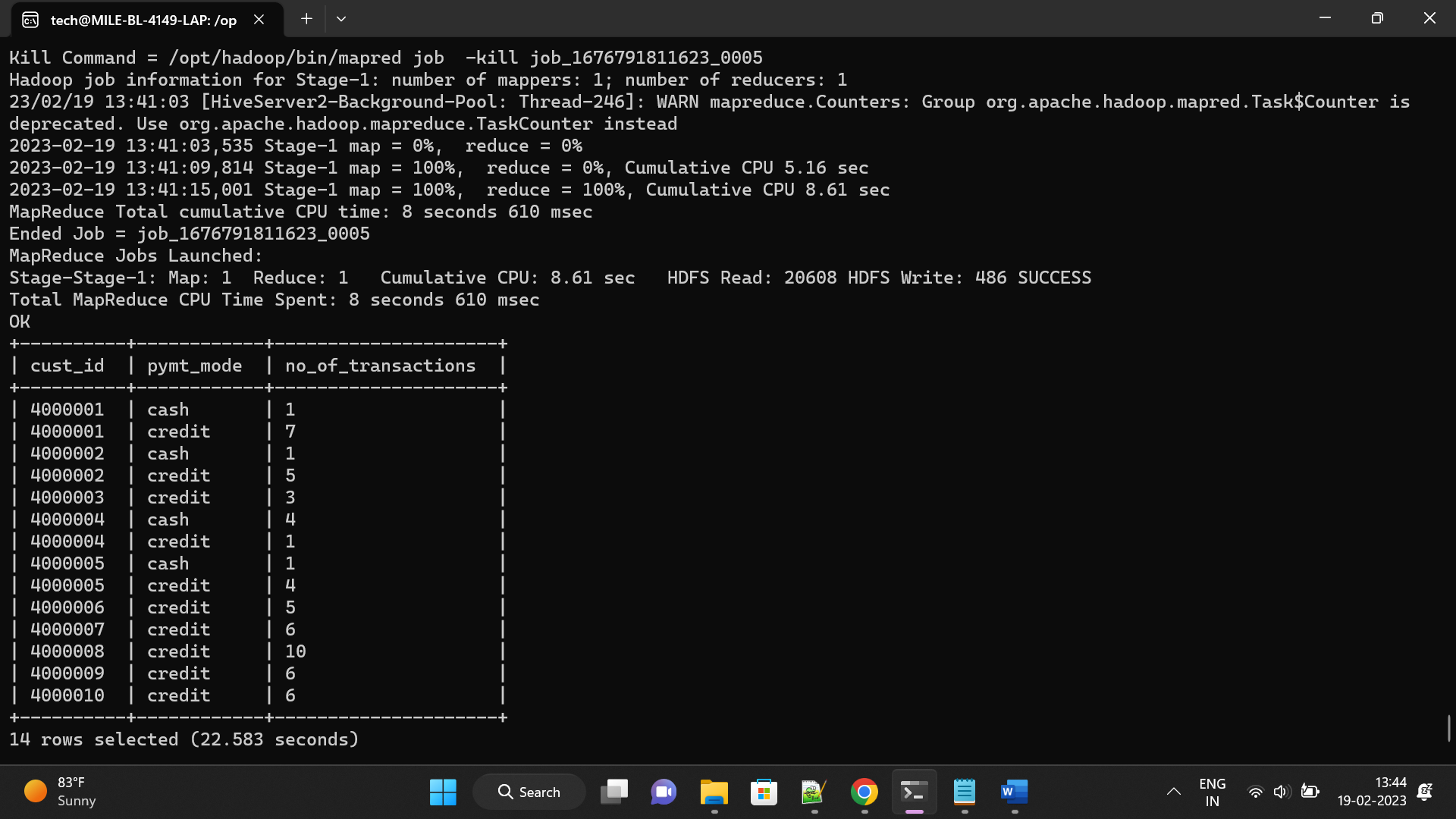


4) No of transactions by customer and mode of payment

0: jdbc:hive2://> SELECT cust\_id, pymt\_mode, COUNT(\*) AS No\_of\_transactions

. . . . . . . . > FROM Transactions

. . . . . . . . > GROUP BY cust\_id, pymt\_mode;



5) Get top 3 cities which has more transactions

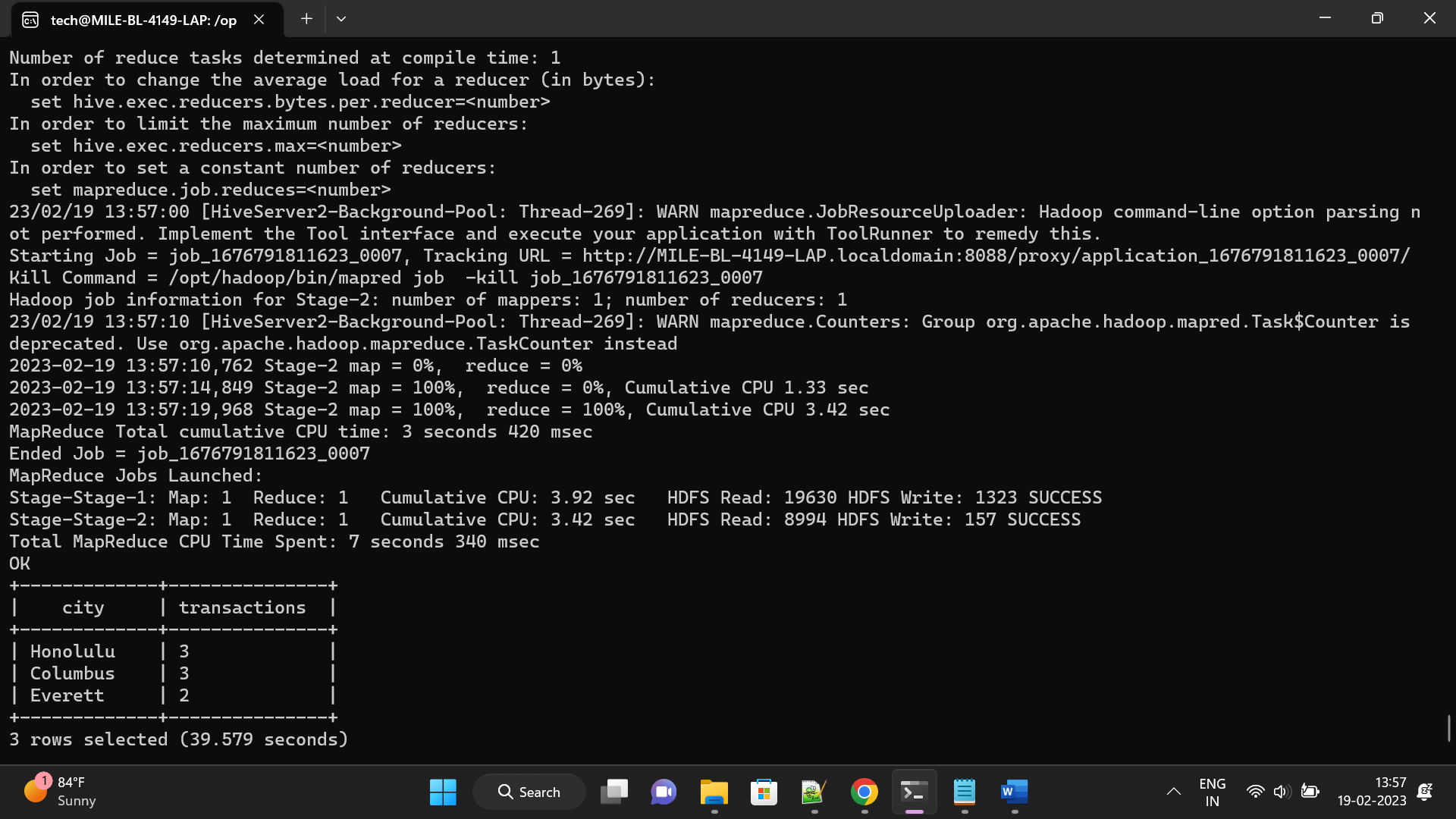
0: jdbc:hive2://> SELECT city, COUNT(trans\_id) as Transactions

. . . . . . . . > FROM Transactions

. . . . . . . . > GROUP BY city

. . . . . . . . > ORDER BY Transactions DESC

. . . . . . . . > LIMIT 3;



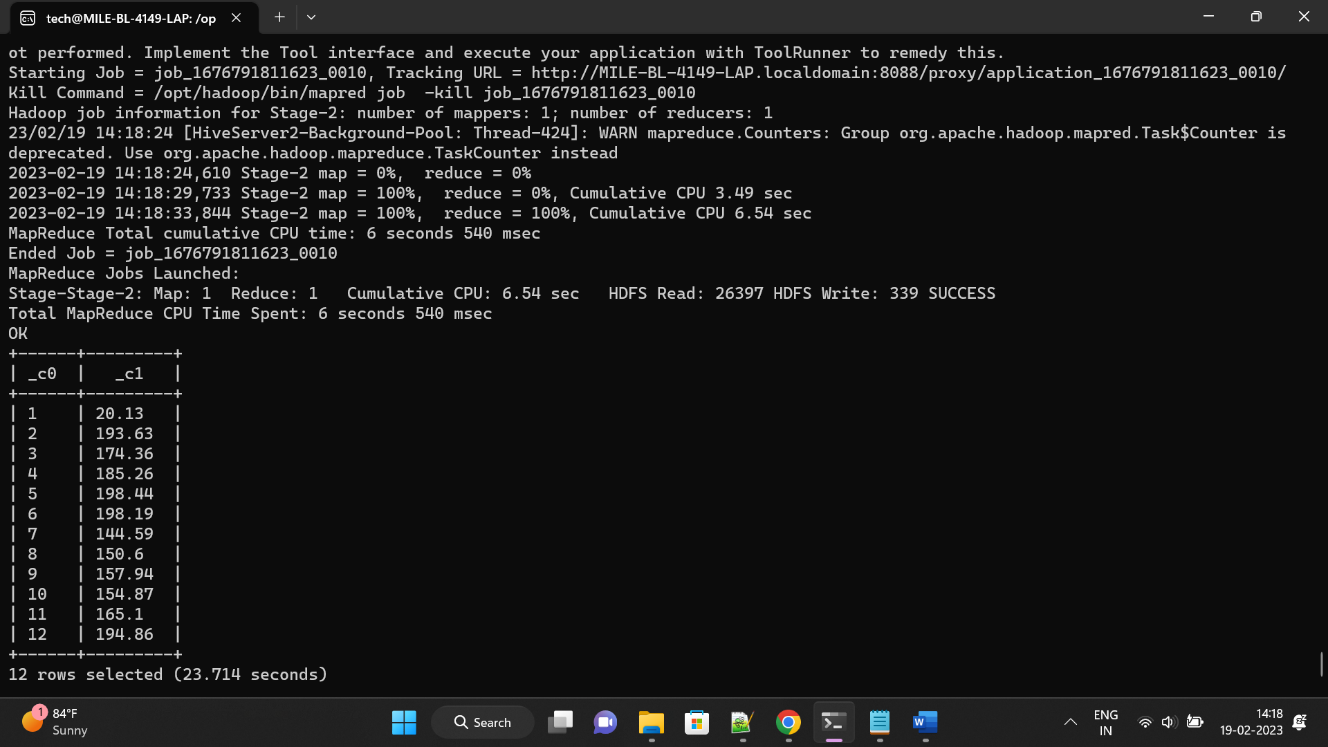
6) Get month wise highest transaction

0: jdbc:hive2://> CREATE TABLE mytime(trans\_id int, trans\_date date);

0: jdbc:hive2://> INSERT INTO TABLE mytime SELECT trans\_id, from\_unixtime(unix\_timestamp(trans\_date, 'MM-dd-yyyy')) FROM Transactions;

0: jdbc:hive2://> SELECT MONTH(mytime.trans\_date), MAX(amount)

. . . . . . . . > FROM Transactions JOIN mytime ON Transactions.trans\_id = mytime.trans\_id

. . . . . . . . > GROUP BY MONTH(mytime.trans\_date);

7) Get sample transactions

0: jdbc:hive2://> SELECT \* FROM Transactions LIMIT 10;

OK

