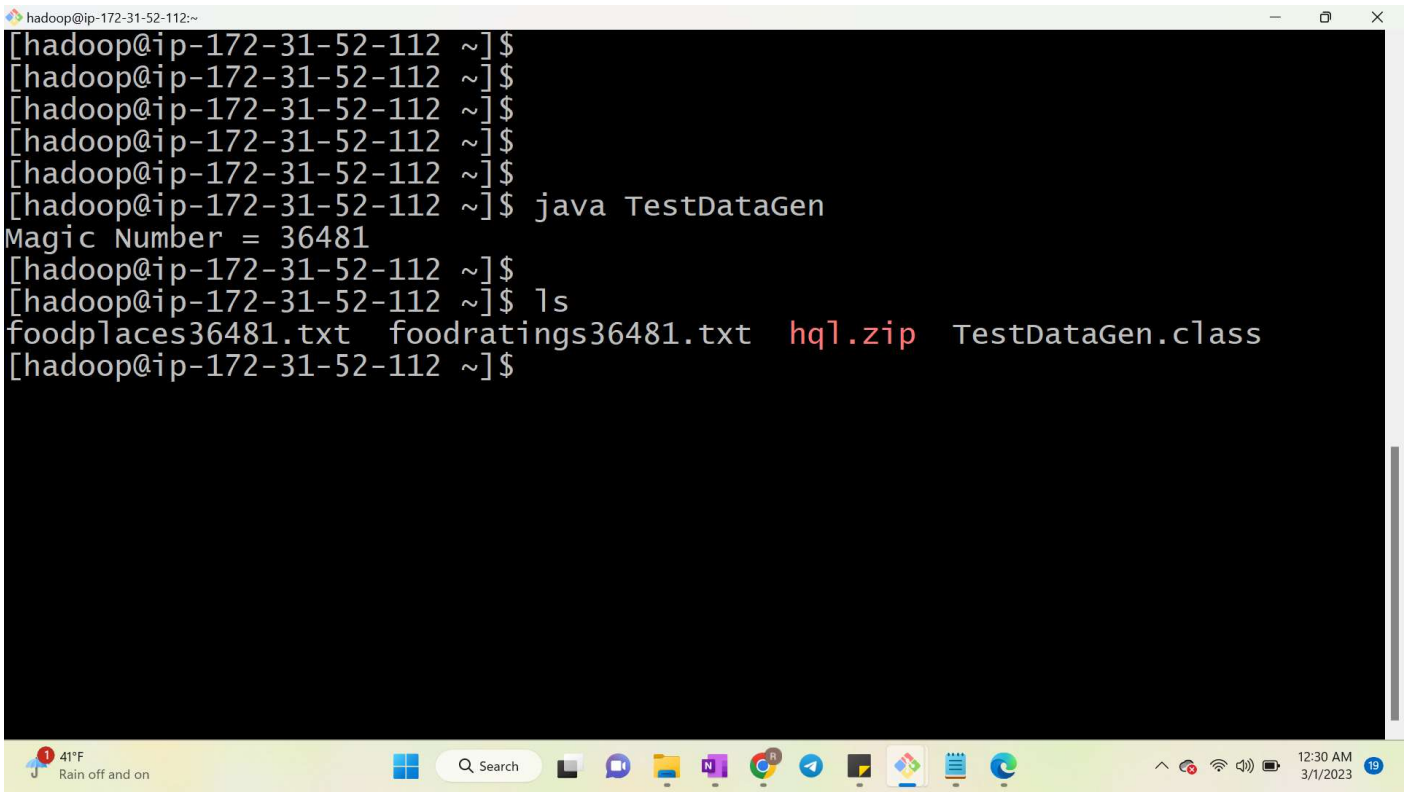
CWID: A20526972

CSP-554 Big Data Technologies

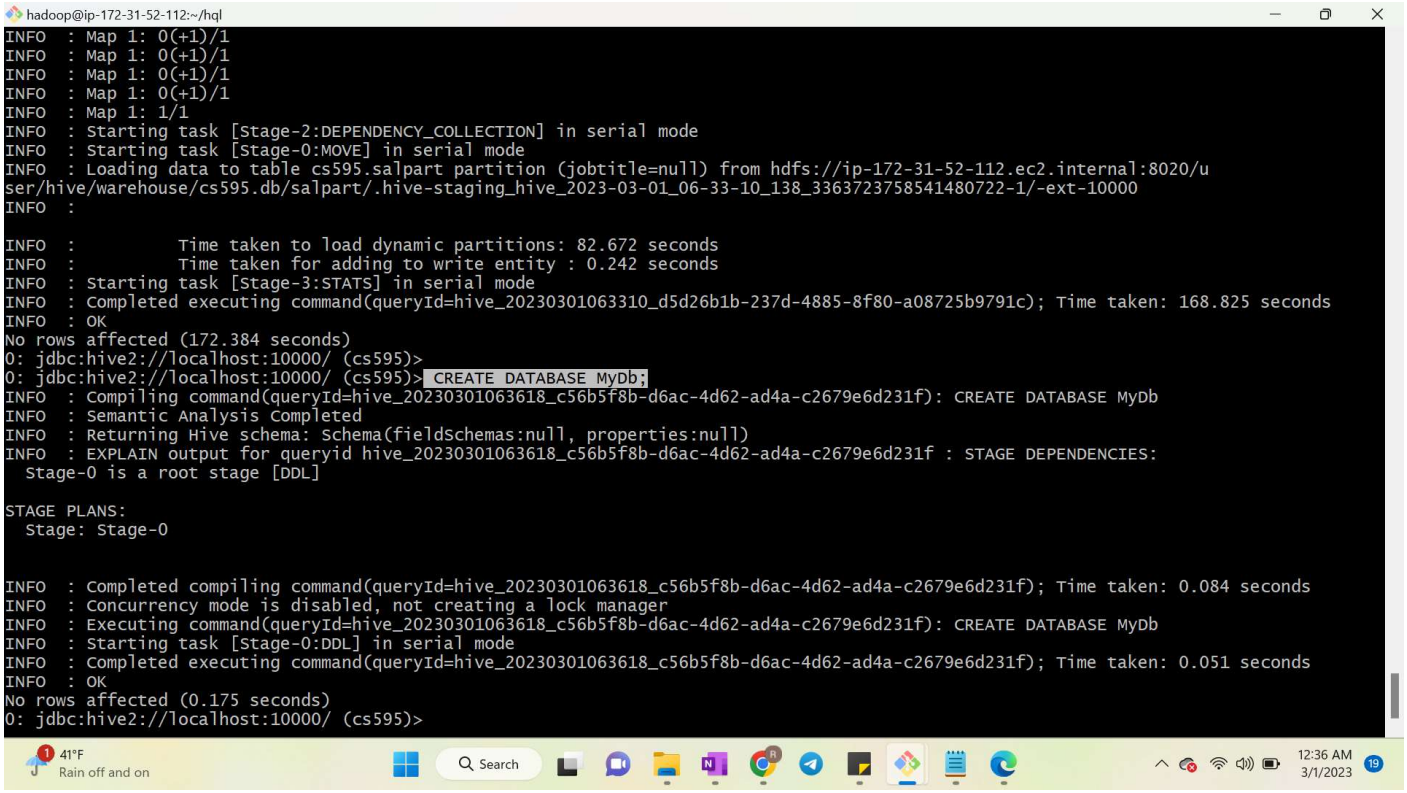
BAIRI ROHITH REDDY – Assignment #4

Magic Number : 36481

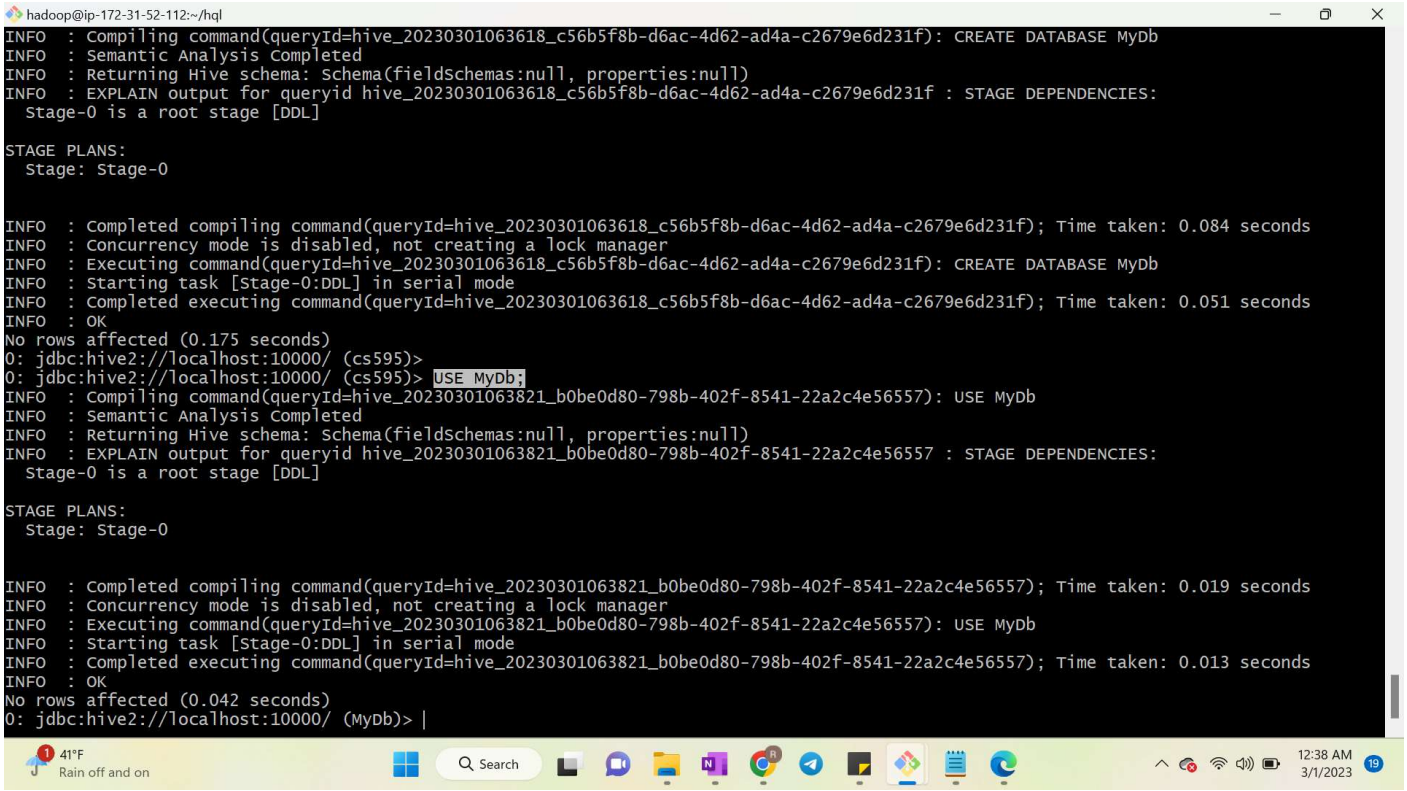


Exercise 1) Create a Hive database called “MyDb”.

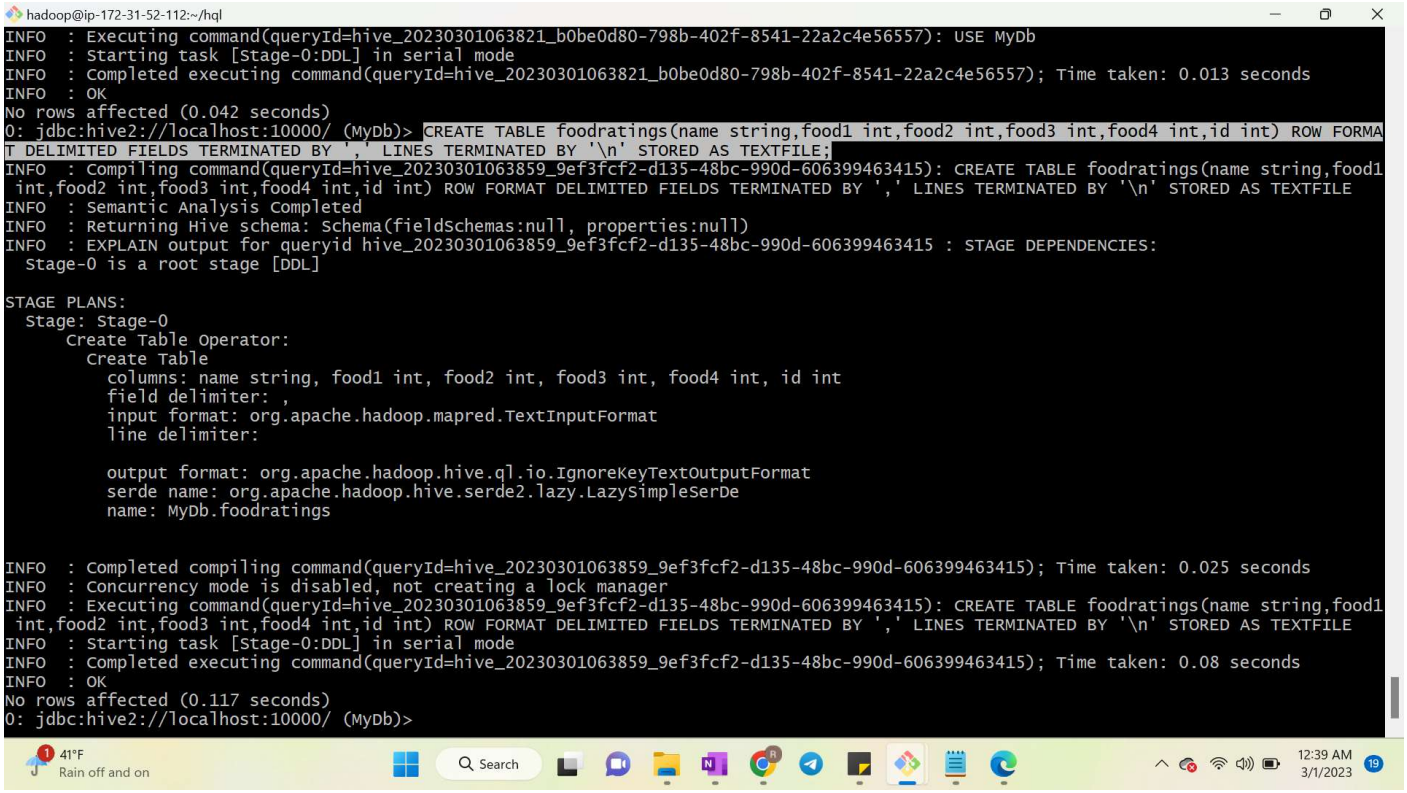
CREATE DATABASE MyDb;



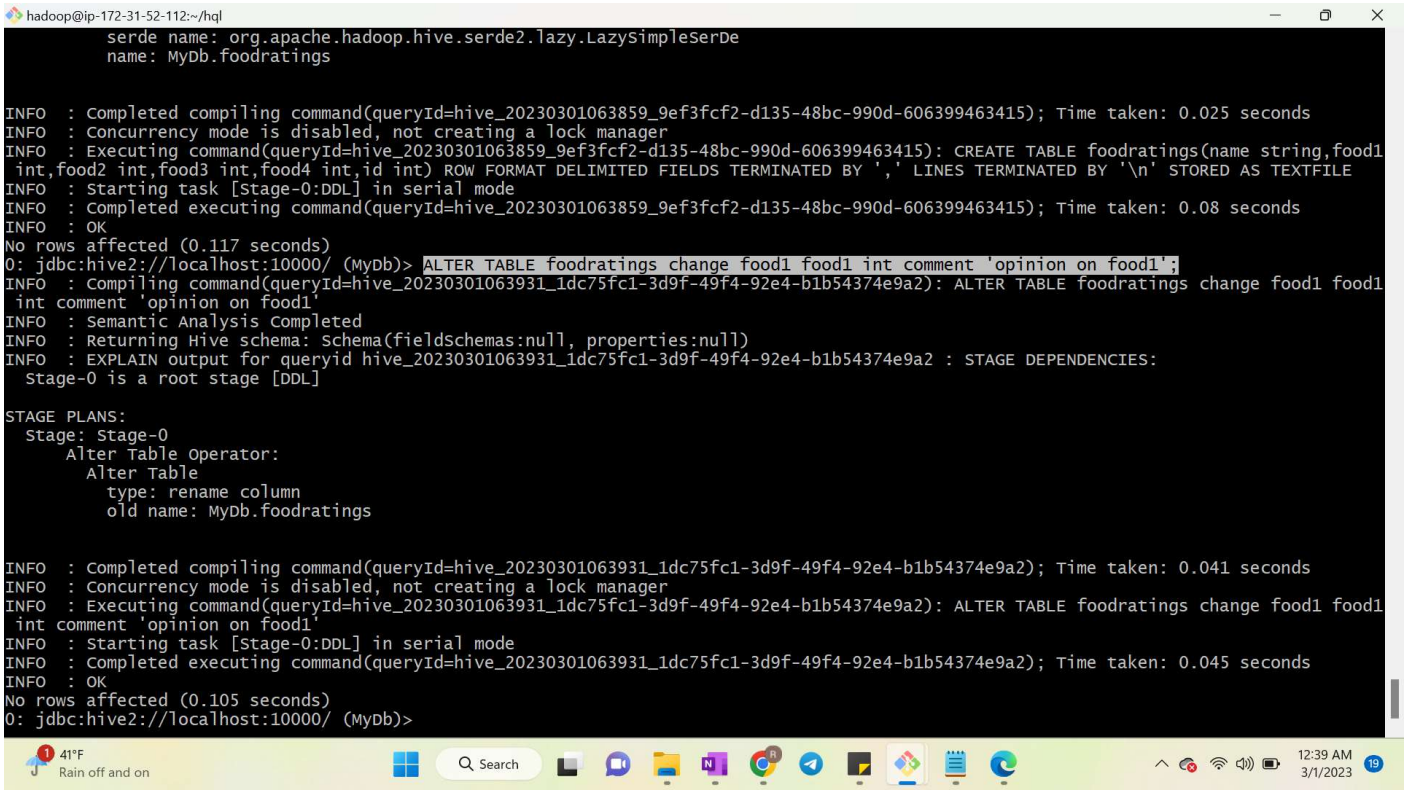
USE MyDB;



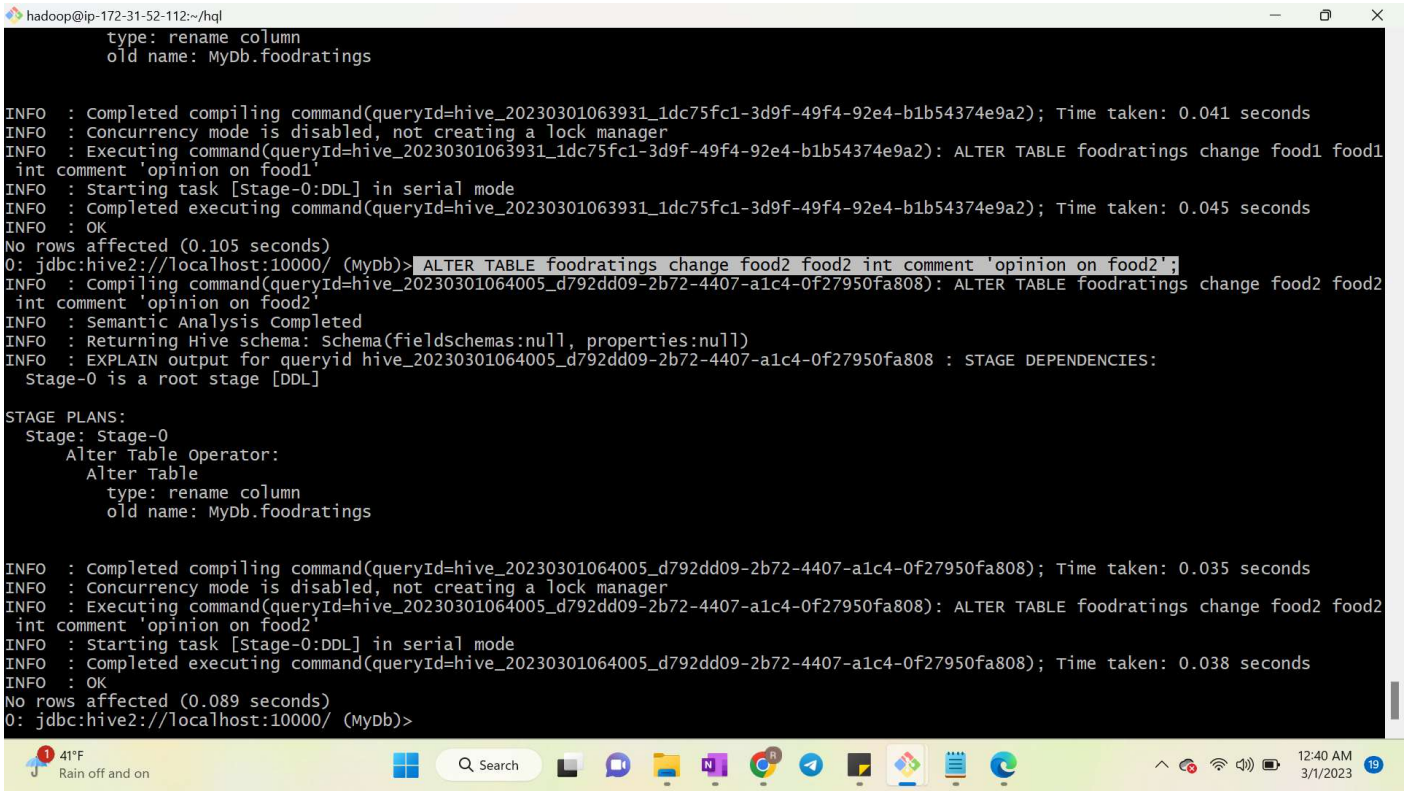
CREATE TABLE foodratings(name string,food1 int,food2 int,food3 int,food4 int,id int) row format delimited fields terminated by ',' lines terminated by '\n' stored as TEXTFILE;



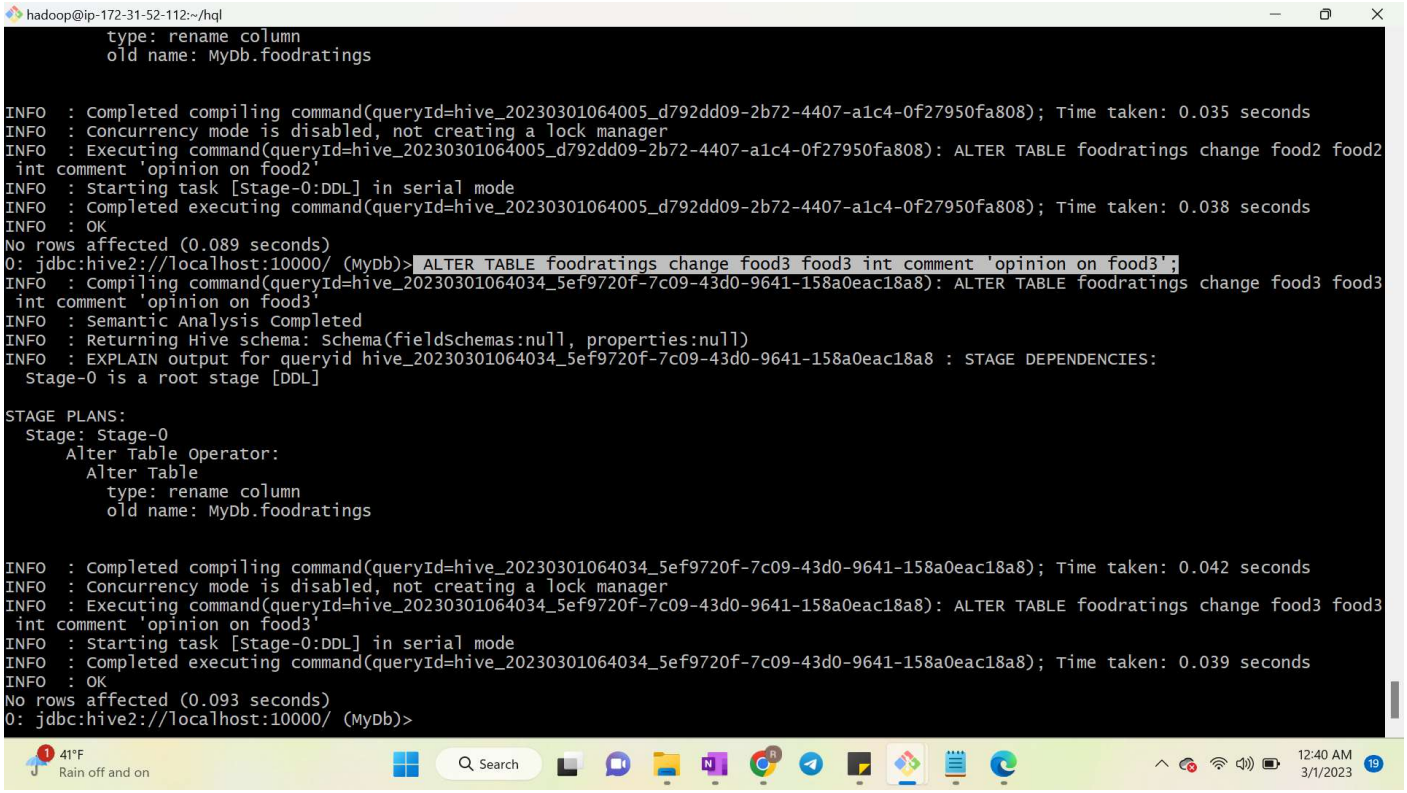
ALTER TABLE foodratings change food1 food1 int comment 'opinion on food1';



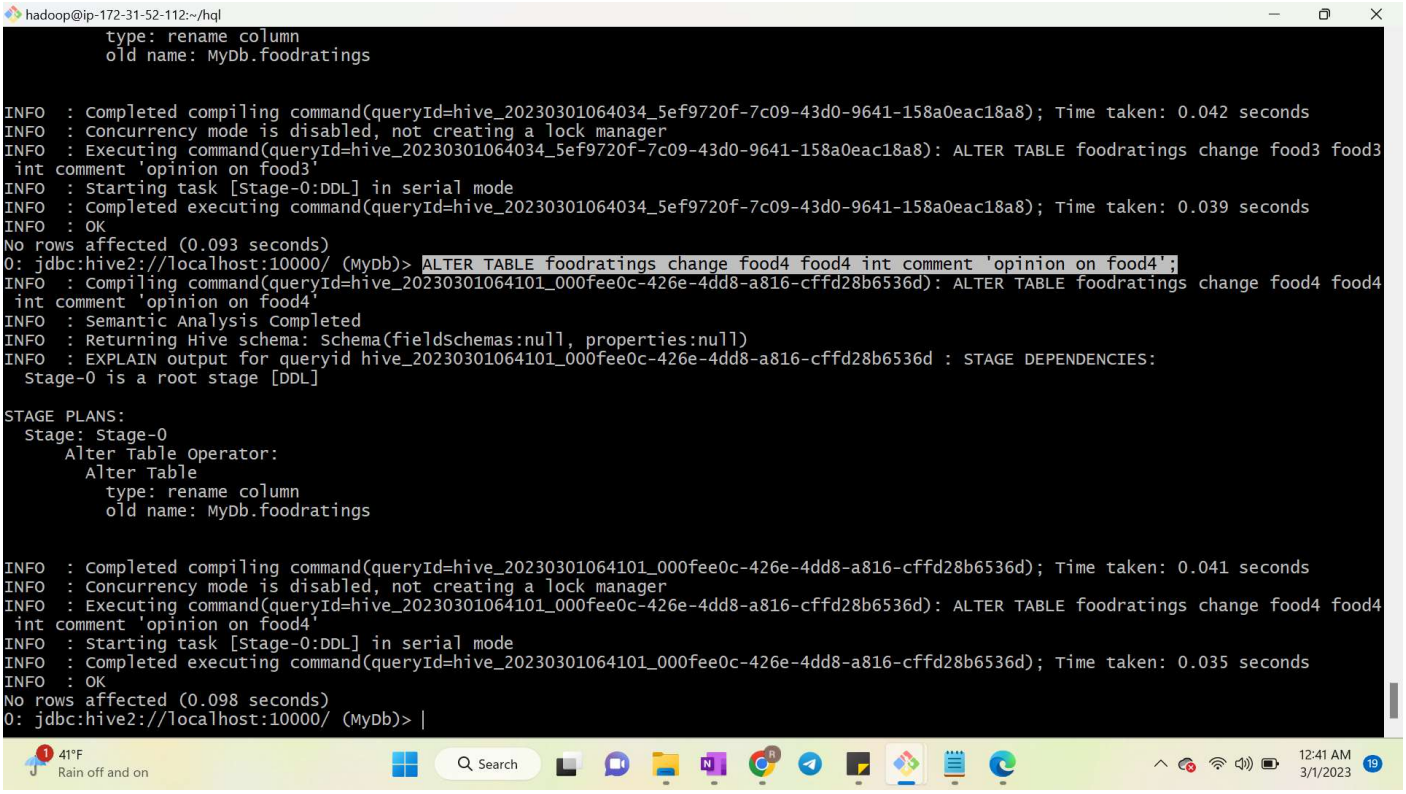
ALTER TABLE foodratings change food2 food2 int comment 'opinion on food2';



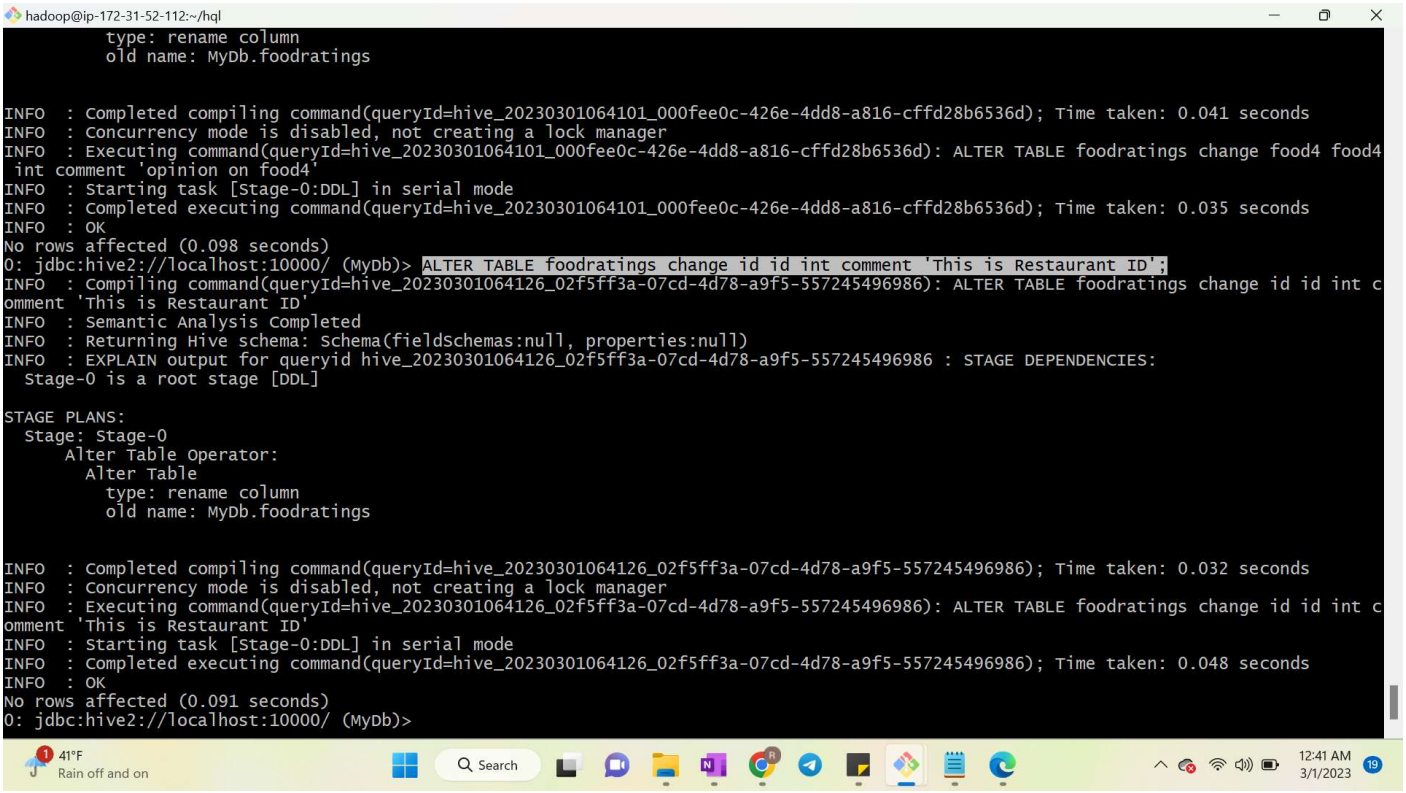
ALTER TABLE foodratings change food3 food3 int comment 'opinion on food3';



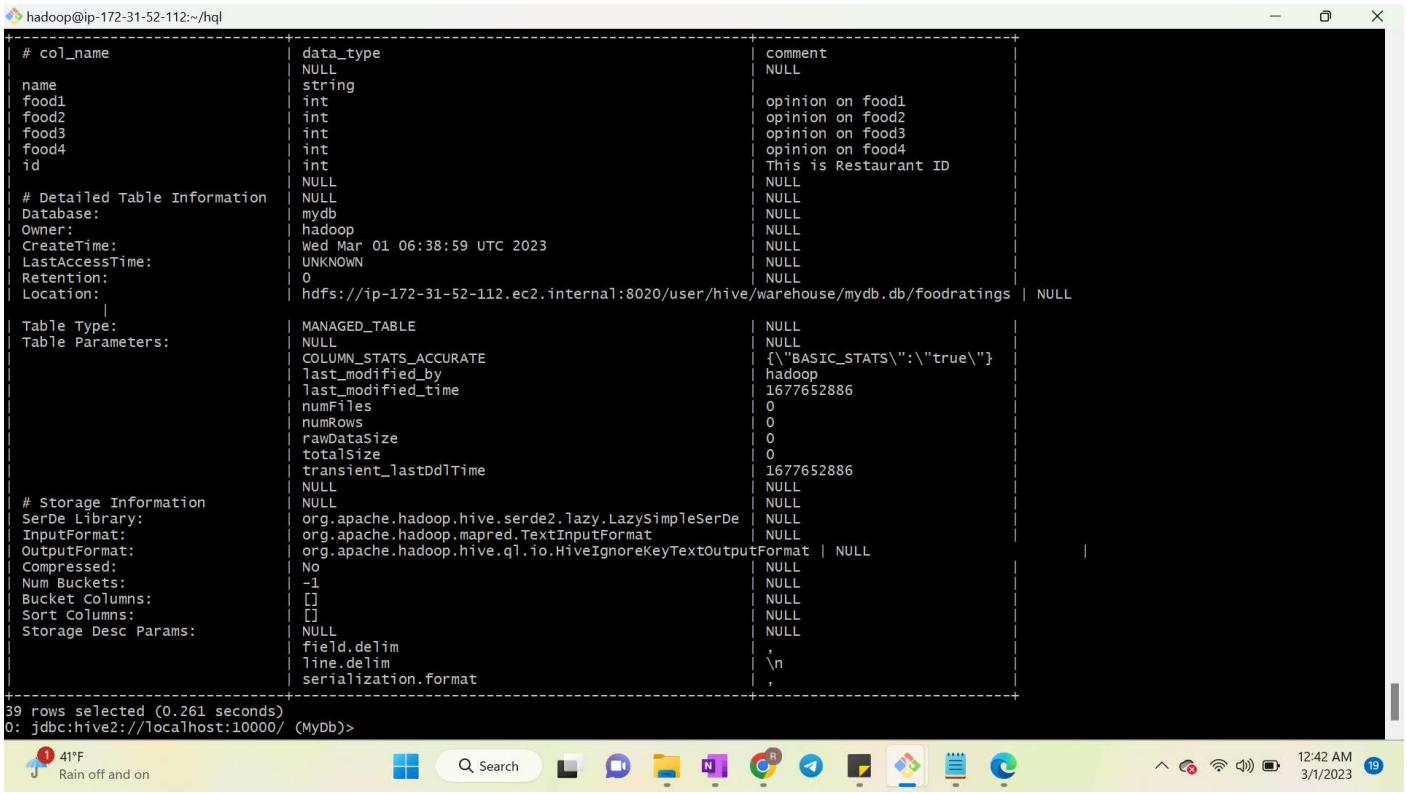
ALTER TABLE foodratings change food4 food4 int comment 'opinion on food4';



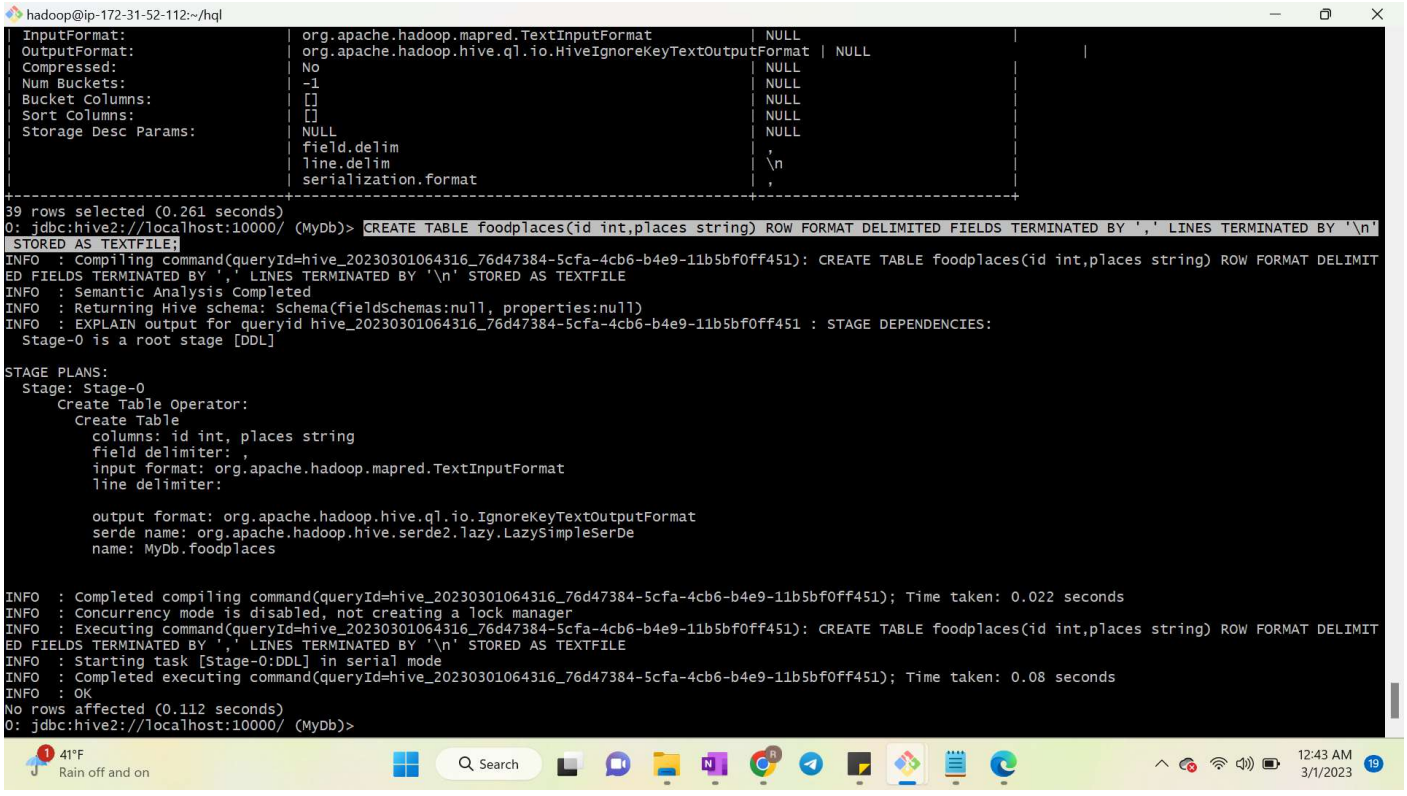
ALTER TABLE foodratings change id id int comment 'This is Restaurant ID';



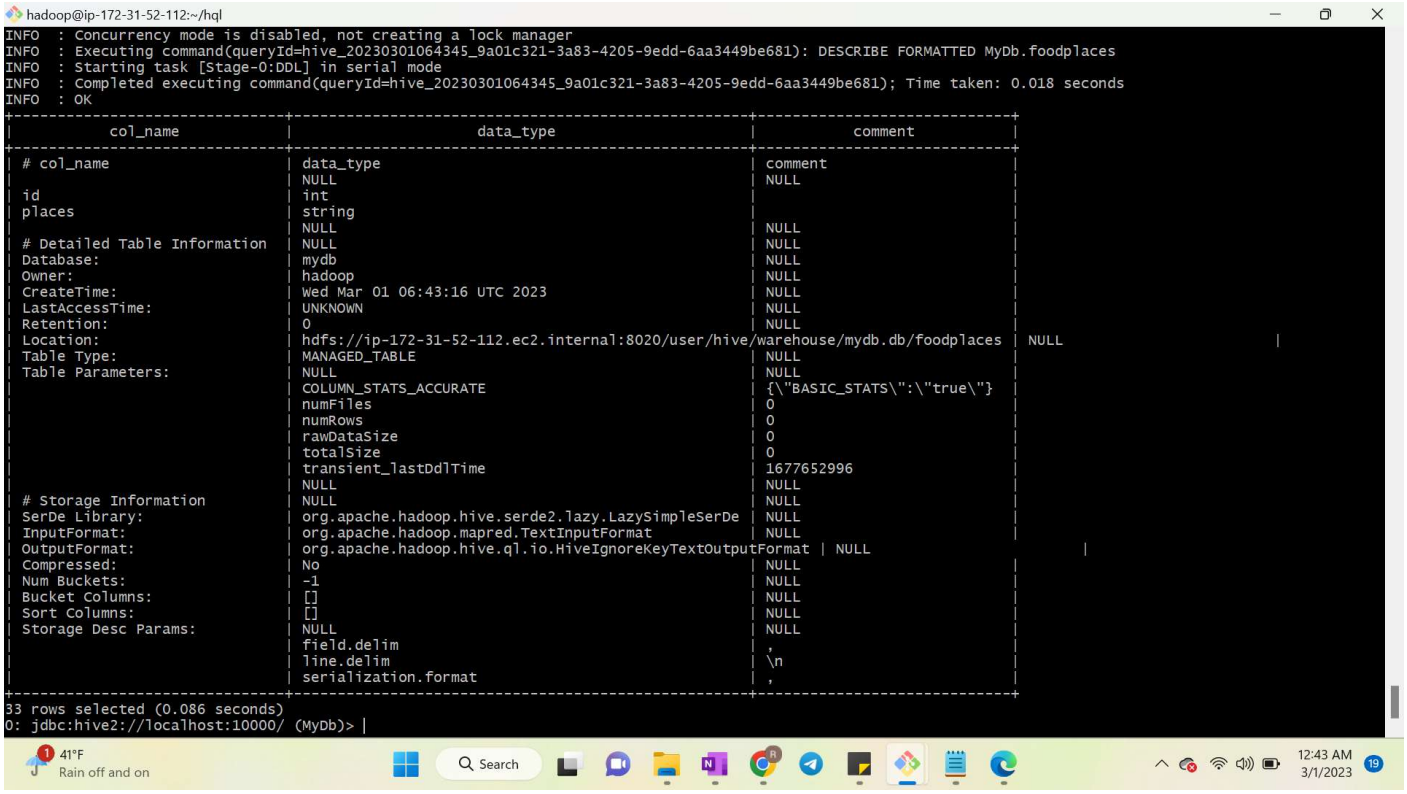
Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodratings;'



CREATE TABLE foodplaces(id int,places string) row format delimited fields terminated by ',' lines terminated by '\n' stored as TEXTFILE;



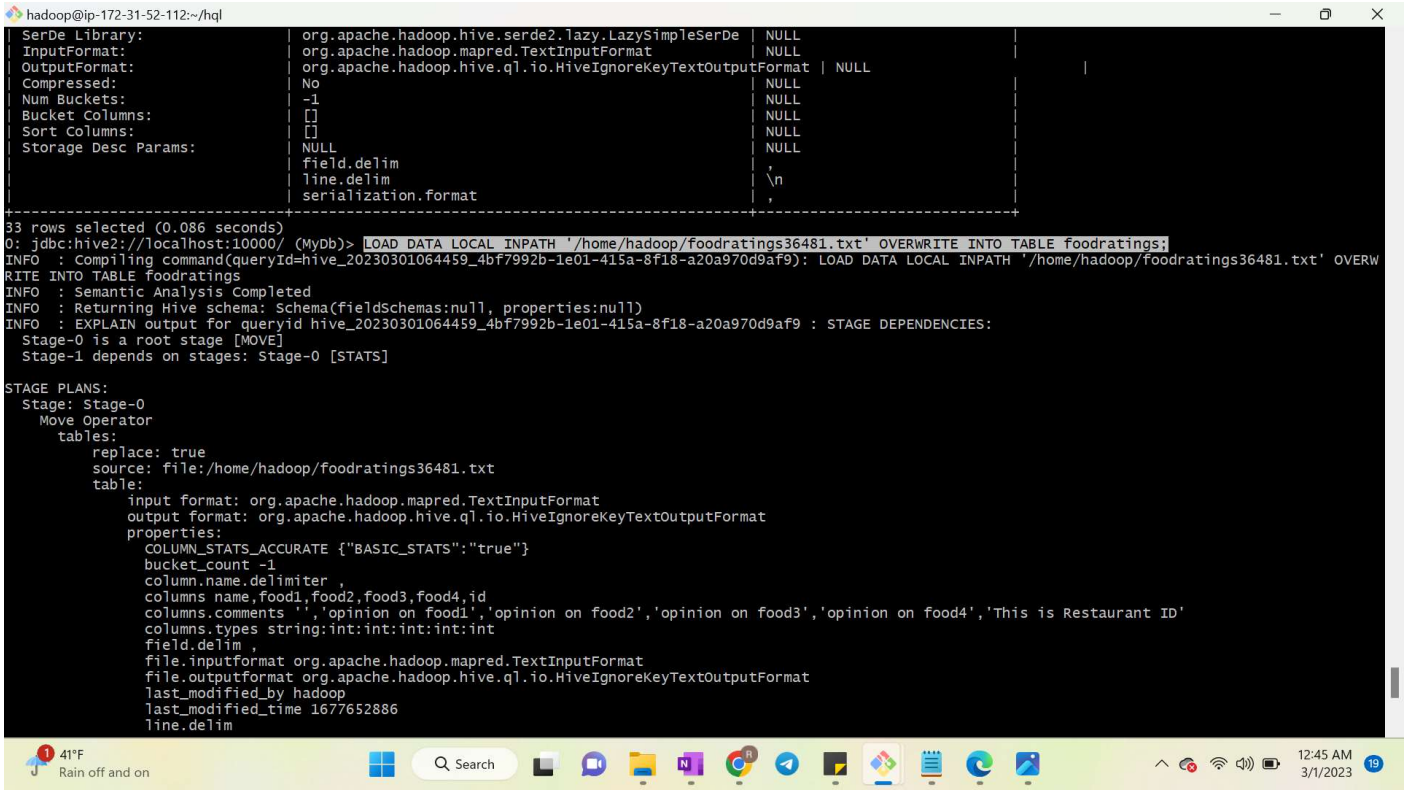
Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodplaces'



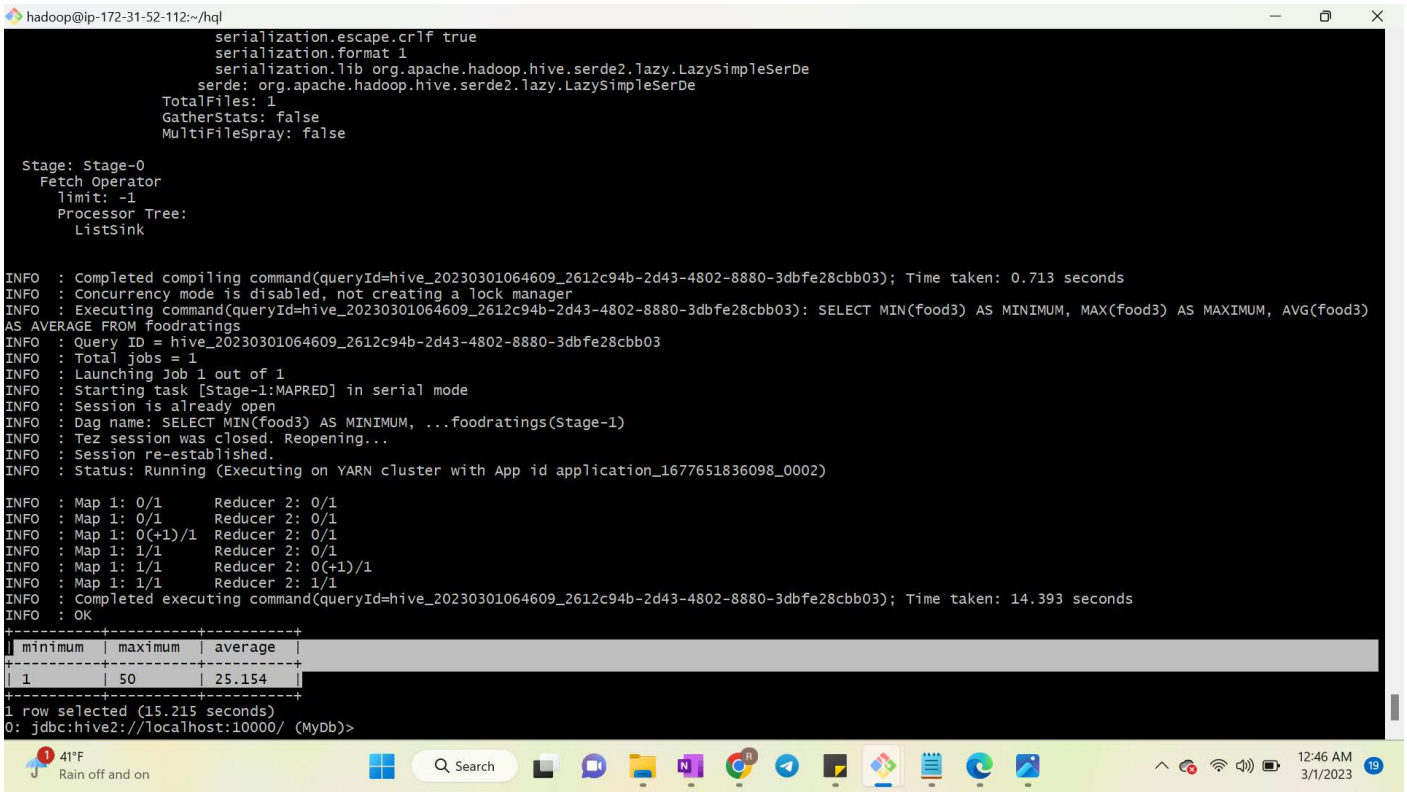
Exercise 2)

Load the foodratings.txt file created using TestDataGen from your local file system into the foodratings table.

LOAD DATA LOCAL INPATH '/home/hadoop/foodratings36481.txt' OVERWRITE INTO TABLE foodratings;



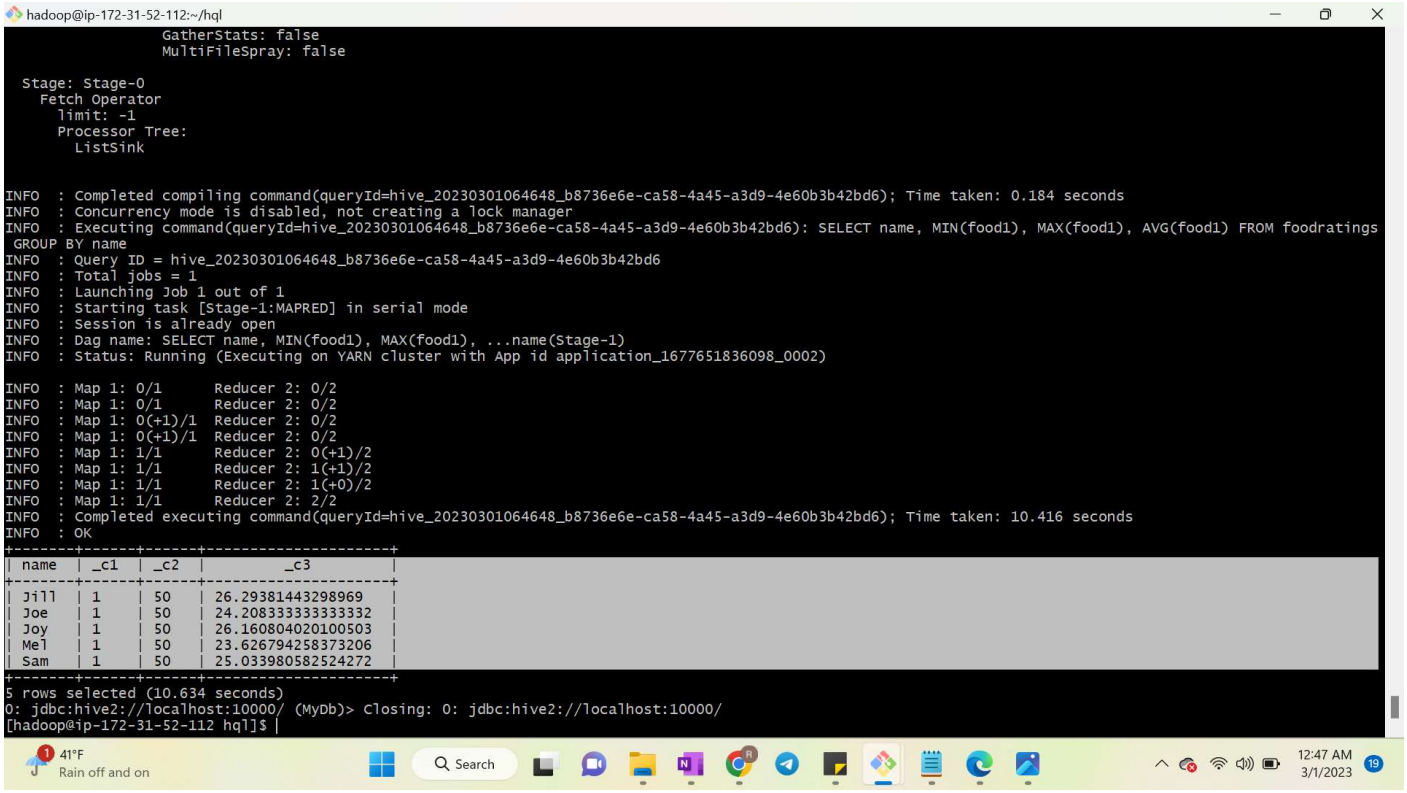
SELECT MIN(food3) AS MINIMUM, MAX(food3) AS MAXIMUM, AVG(food3) AS AVERAGE FROM foodratings;



Exercise 3)

Execute a hive command to output the min, max and average of the values of the food1 column grouped by the first column 'name'.

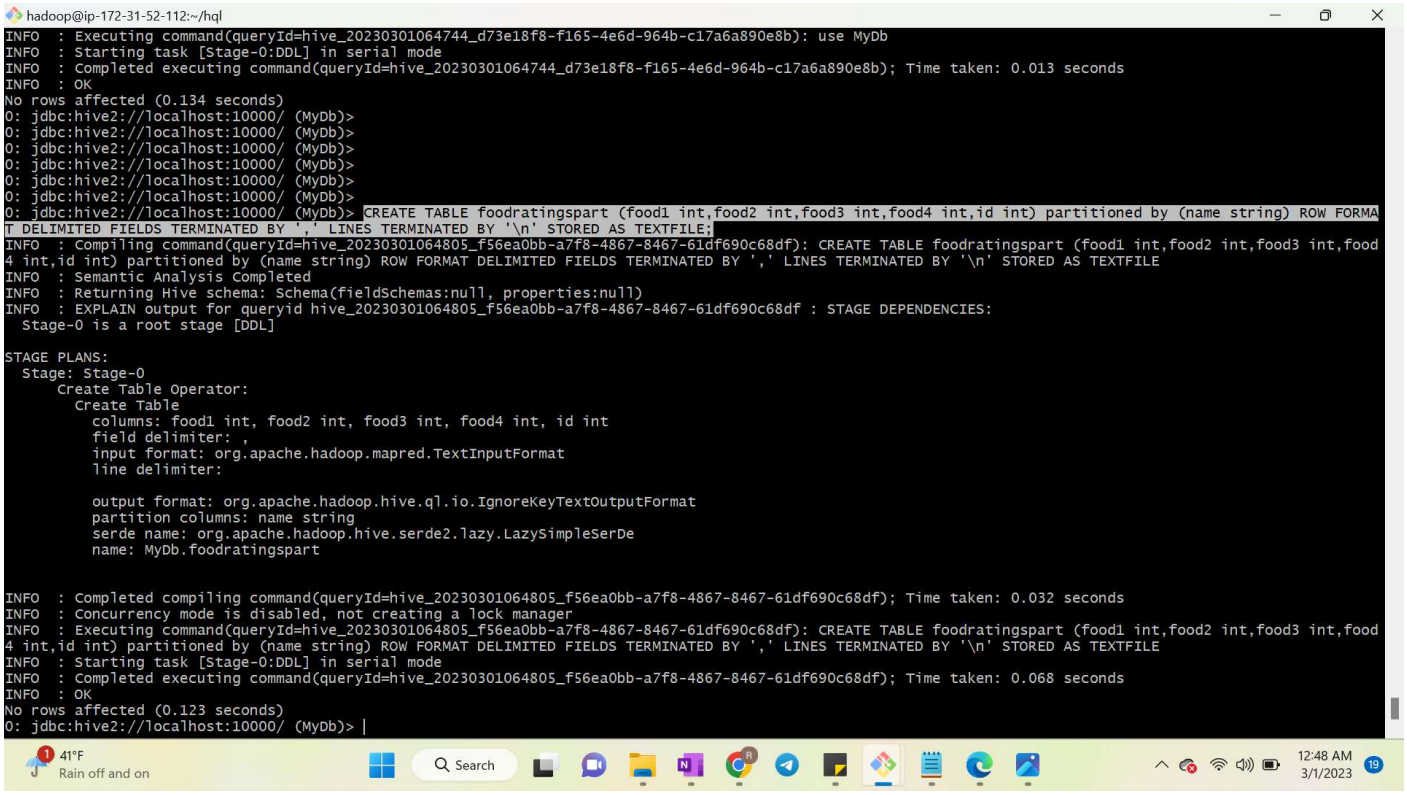
SELECT name, MIN(food1), MAX(food1), AVG(food1) FROM foodratings GROUP BY name;



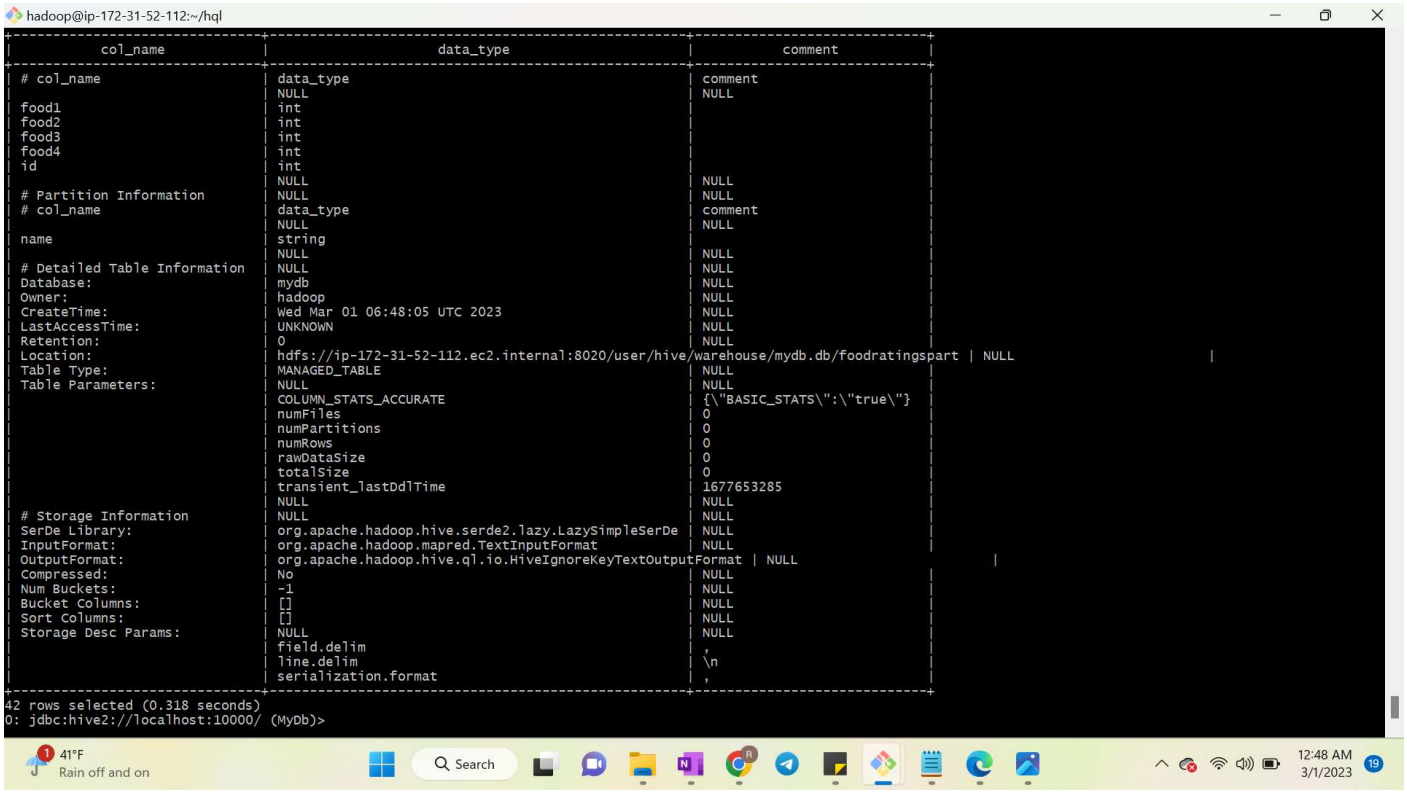
Exercise 4)

In MyDb create a partitioned table called 'foodratingspart'.

CREATE TABLE foodratingspart (food1 int,food2 int,food3 int, food4 int,id int) partitioned by (name string) ROW FORMAT DELIMITED BY ',' LINES DELIMITED BY '\n' STORED AS TEXTFILE;



Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodratingspart;'



Exercise 5)

Assume that the number of food critics is relatively small, say less than 10 and the number places to eat is very large, say more than 10,000. In a few short sentences explain why using the (critic) name is a good choice for a partition field while using the place id is not.

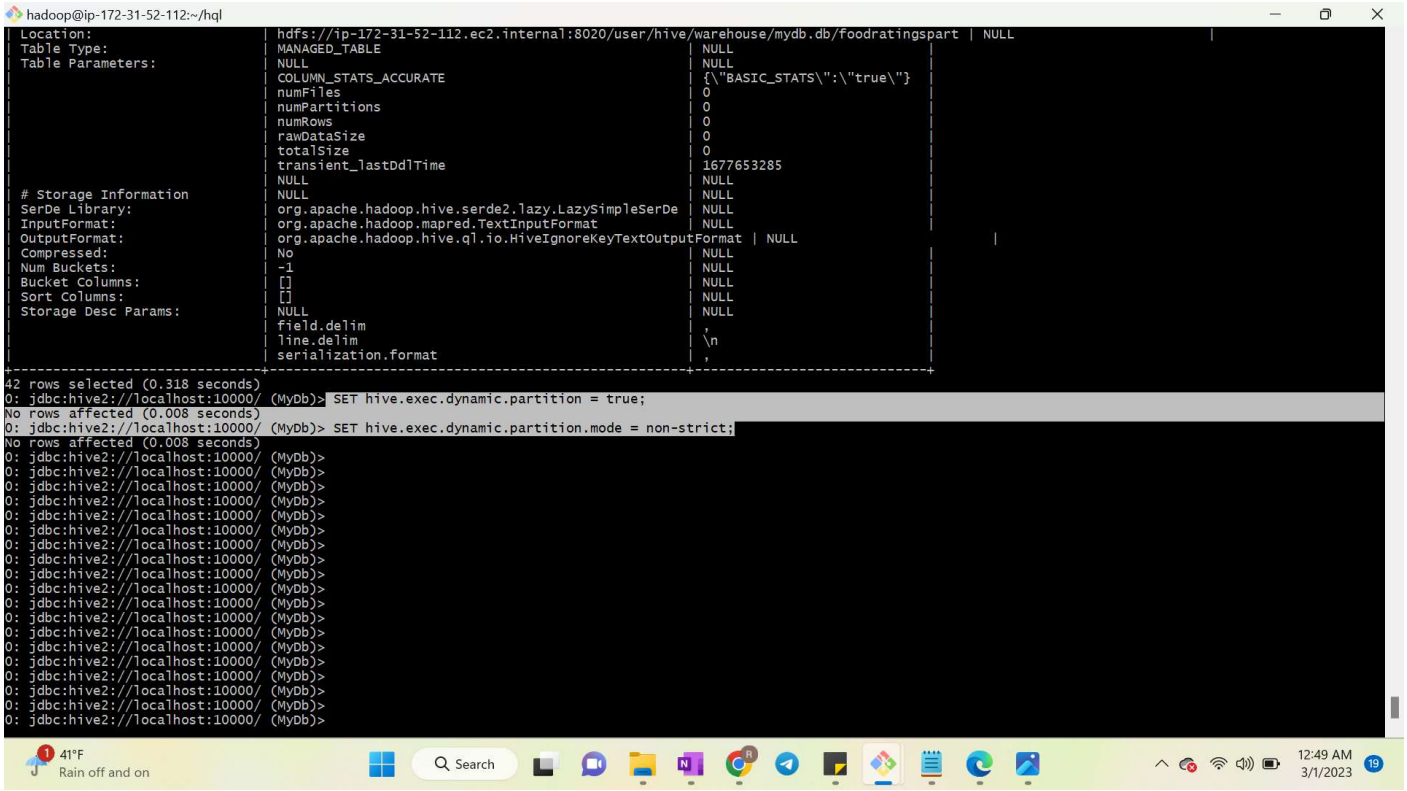
Partitioning is done to improve the query performance, so the column we choose to do partition with must be minimal. As stated the food critics are lesser in number and hence the number partitions are limited and are less than 10 (in this case as the number of critics are less than 10). But if we choose to partition on place id, then the number partitions will be huge and this will increase the expense of loading and retrieving data which results in performance degradation due to longer execution times.

Exercise 6)

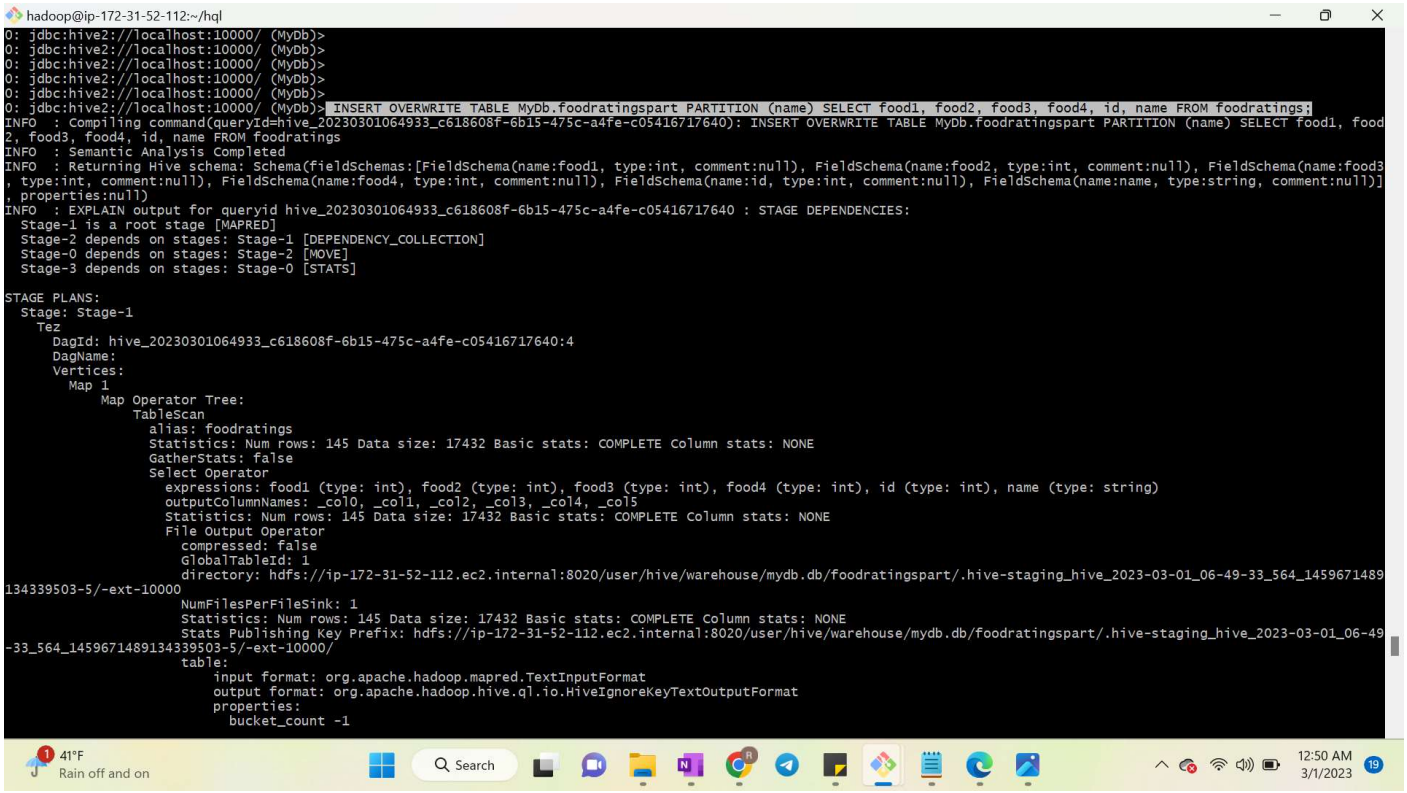
Execute a hive command to output the min, max and average of the values of the food2 column of MyDB.foodratingspart where the food critic 'name' is either Mel or Jill.

SET hive.exec.dynamic.partition = true;

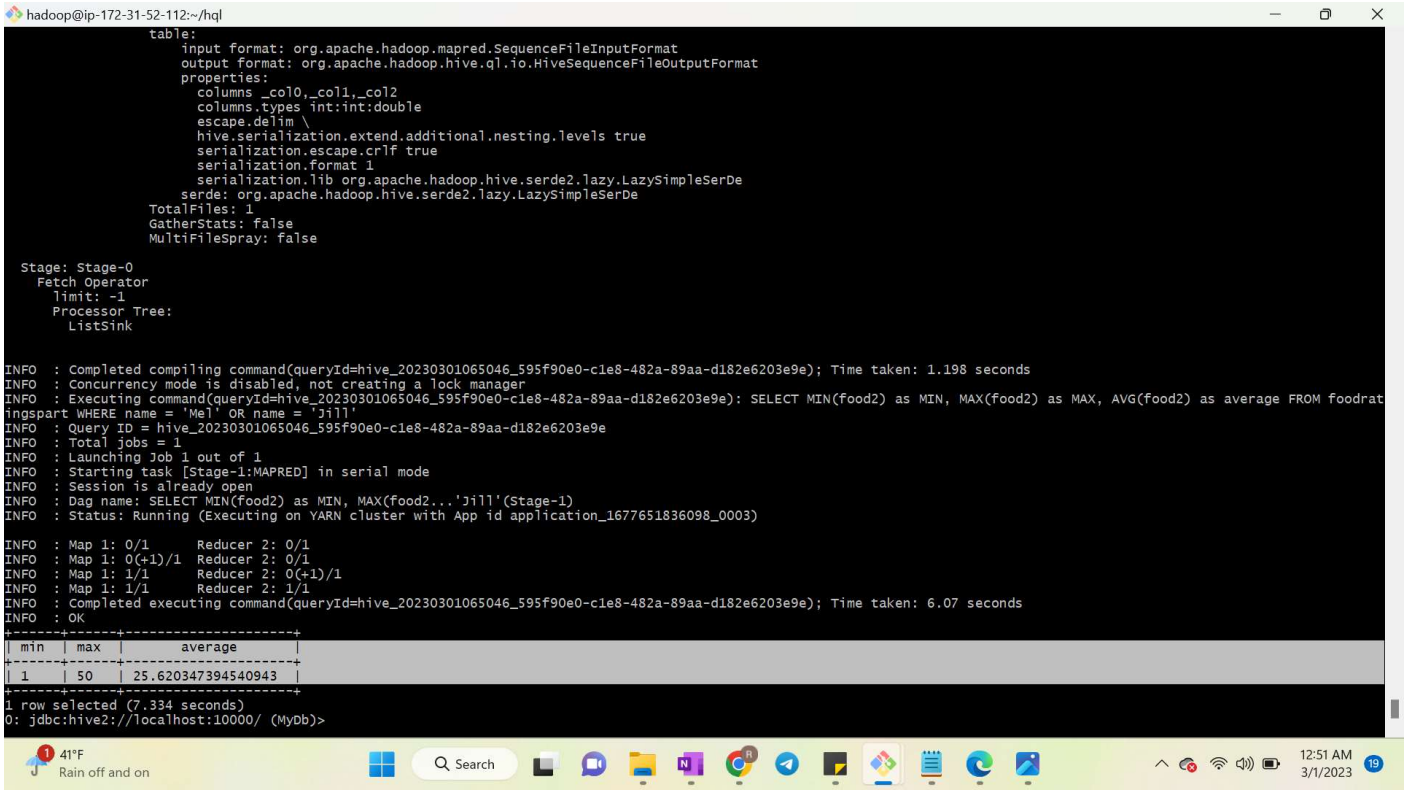
SET hive.exec.dynamic.partition.mode = non-strict;



INSERT OVERWRITE TABLE MyDb.foodratingspart PARTITION (name) SELECT food1, food2, food3, food4, id, name FROM foodratings;



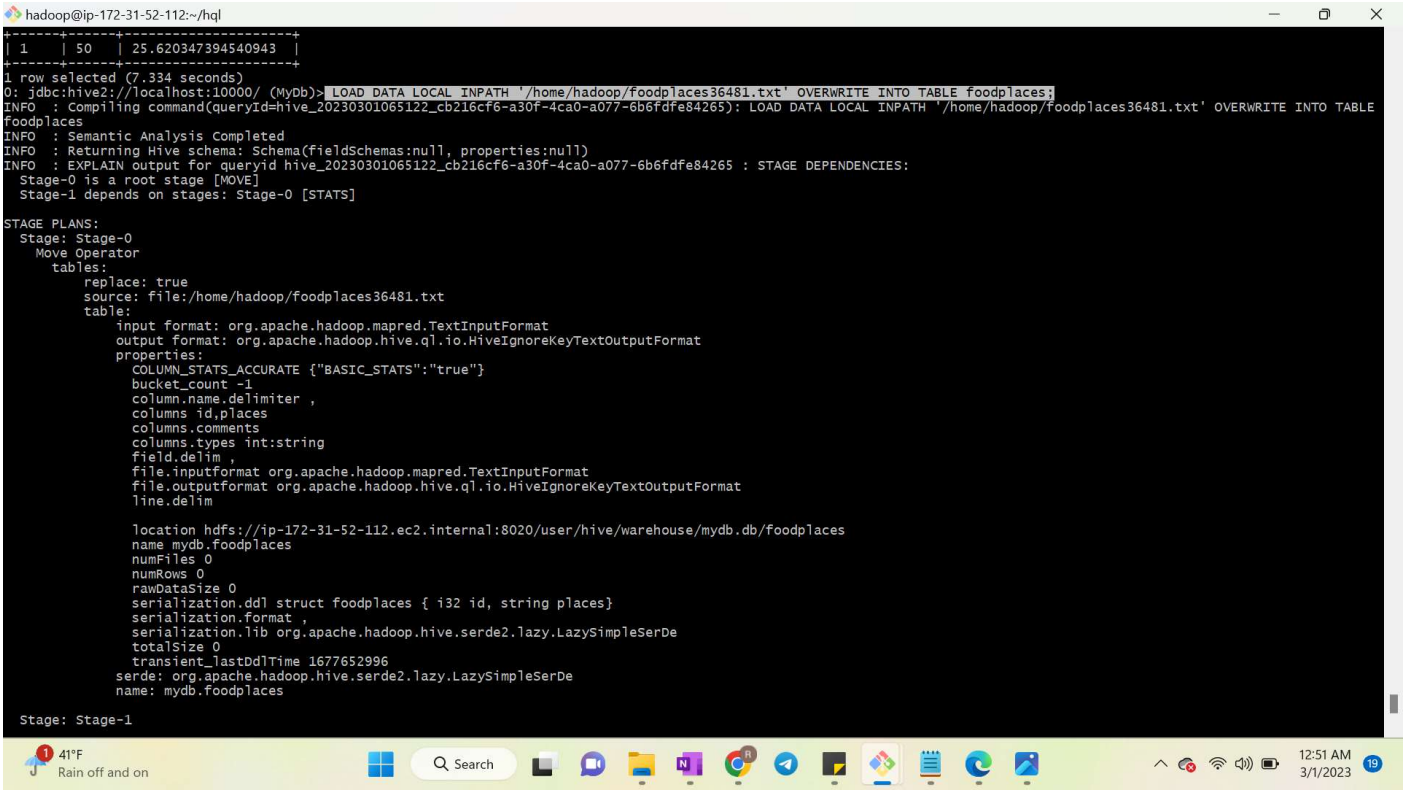
SELECT MIN(food2) as MIN, MAX(food2) as MAX, AVG(food2) as average FROM foodratingspart WHERE name = 'Mel' OR name = 'Jill';



Exercise 7)

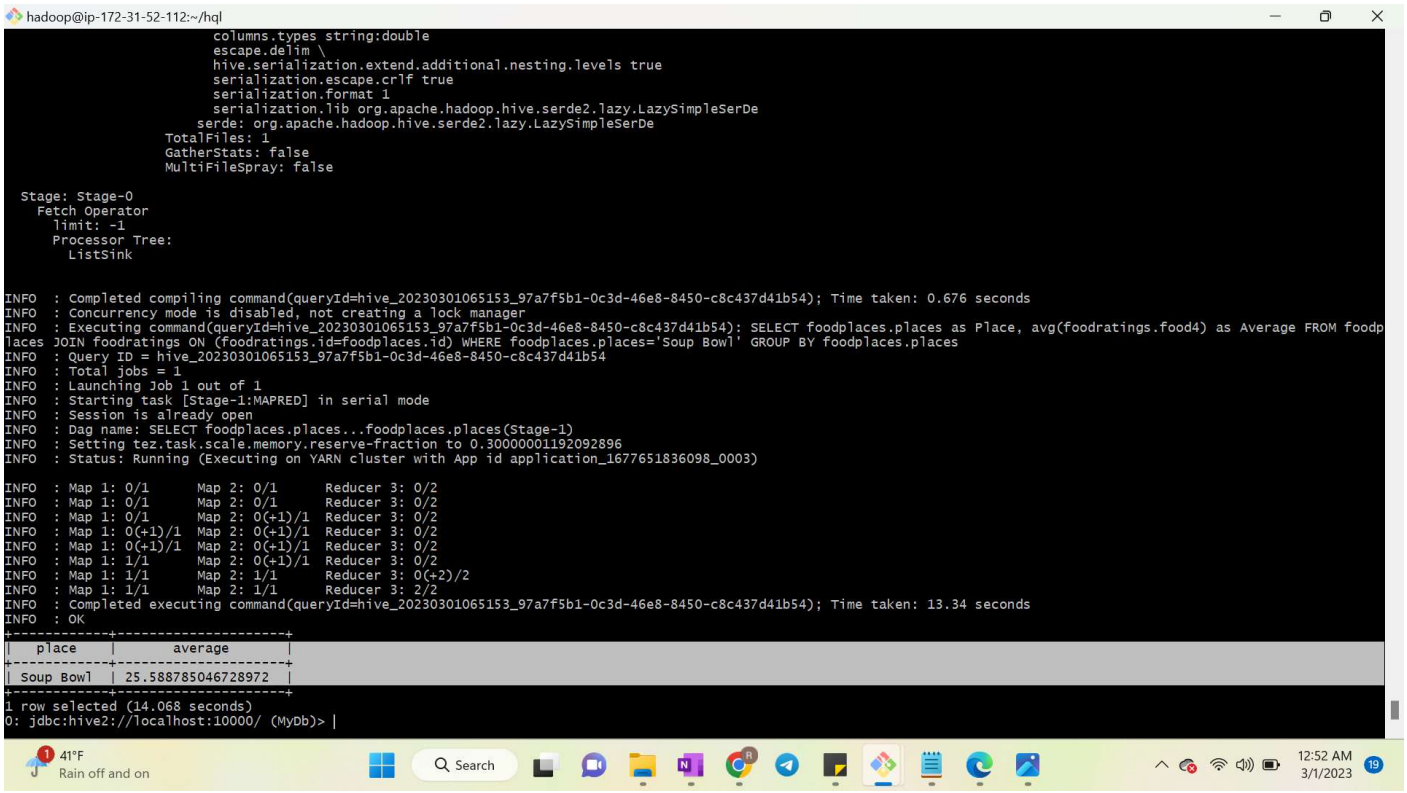
Load the foodplaces.txt file created using TestDataGen from your local file system into the foodplaces table.

LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces76011.txt' OVERWRITE INTO TABLE foodplaces;



Use a join operation between the two tables (foodratings and foodplaces) to provide the average rating for field food4 for the restaurant 'Soup Bowl'.

SELECT foodplaces.places as Place, avg(foodratings.food4) as Average FROM foodplaces JOIN foodratings ON (foodratings.id=foodplaces.id) WHERE foodplaces.places='Soup Bowl' GROUP BY foodplaces.places;



Exercise 8)

Read the article "An Introduction to Big Data Formats" found on the blackboard in section "Articles" and provide short (2 to 4 sentence) answers to the following questions:

a) **When is the most important consideration when choosing a row format and when a column format for your big data file?**

Row format is preferred when the query needs access to all the columns in the table.

When, only the data from a specific columns is needed, Column format is employed. Most analytical queries in big data require only a subset of columns from the relations, so using column format here is appropriate.

b) **What is "splittability" for a column file format and why is it important when processing large volumes of data?**

'Splittablility', refers the ability to split larger jobs into smaller jobs for parallel execution. Column file format stores the larger files in smaller chunks of columns for efficient querying and processing. Since, processing of large volumes of data could be very slow or impossible on a single machine. Therefore, splittability is a critical feature that makes column file formats an effective solution to this problem.

c) **What can files stored in column format achieve better compression than those stored in row format?**

Column file format can achieve better compression because of the data is stored column by column and these columns are of single data type allowing for better compression whereas the row format has all the columns with different data types making it difficult for compression.
Files stored in column formats achieve better query performance as the query has less data to search from where as the row format has a lot of columns resulting in longer processing time.

d) **Under what circumstances would it be the best choice to use the "Parquet" column file format**

Columnar file storage format of 'Parquet' allows for efficient compression, splittable data, faster and improved query performance.
Parquet is extremely useful for distributed computing frameworks such as Hadoop, which allows for parallel-processing across multiple nodes in a cluster. It is very helpful for processing of big data datasets which have numerous columns, where storage efficiency, query performance and data compression are important.

CWID: A20526972

DOCUMENTATION:-

STEP BY STEP WORKFLOW OF HOMEWORK-4:-

In bash shell-2/ normal bash shell:-

scp -i C:/Users/rohit/OneDrive/Desktop/all_files/BIGDATA/emrkey-pair.cer C:/Users/rohit/OneDrive/Desktop/all_files/BIGDATA/Homework-Assignments/Homework-4/TestDataGen.class hadoop@<MASTER-NODE-DNS>/home/hadoop

scp -i C:/Users/rohit/OneDrive/Desktop/all_files/BIGDATA/emrkey-pair.cer C:/Users/rohit/OneDrive/Desktop/all_files/BIGDATA/Homework-Assignments/Homework-4/hql.zip hadoop@<MASTER-NODE-DNS>:/home/hadoop

In hadoop cmd/ hadoop shell:-

ssh -i C:\Users\rohit\OneDrive\Desktop\all_files\BIGDATA\emrkey-pair.cer hadoop@<MASTER-NODE-DNS>

ls

java TestDataGen

ls

unzip hql.zip

cd /home/hadoop/hql

beeline -u jdbc:hive2://localhost:10000/ -n hadoop -d org.apache.hive.jdbc.HiveDriver --showDbInPrompt

source ./basicsetup.hql;

source ./partsetup.hql;

source ./salaries.hql;

source ./loadsalaries.hql;

source ./salaries2.hql;

source ./salaries3.hql;

1. Question

CREATE DATABASE MyDb;

USE MyDb;

CREATE TABLE foodratings(name string,food1 int,food2 int,food3 int,food4 int,id int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

ALTER TABLE foodratings change food1 food1 int comment 'opinion on food1';

ALTER TABLE foodratings change food2 food2 int comment 'opinion on food2';

ALTER TABLE foodratings change food3 food3 int comment 'opinion on food3';

ALTER TABLE foodratings change food4 food4 int comment 'opinion on food4';

ALTER TABLE foodratings change id id int comment 'This is Restaurant ID';

DESCRIBE FORMATTED MyDb.foodratings;

CREATE TABLE foodplaces(id int,places string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

DESCRIBE FORMATTED MyDb.foodplaces;

2. Question

LOAD DATA LOCAL INPATH '/home/hadoop/foodratings36481.txt' OVERWRITE INTO TABLE foodratings;

SELECT MIN(food3) AS MINIMUM, MAX(food3) AS MAXIMUM, AVG(food3) AS AVERAGE FROM foodratings;

3. Question

SELECT name, MIN(food1), MAX(food1), AVG(food1) FROM foodratings GROUP BY name;

4. question

CREATE TABLE foodratingspart (food1 int,food2 int,food3 int,food4 int,id int) partitioned by (name string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;

DESCRIBE FORMATTED MyDb.foodratingspart;

5. Question answer the theory question.

6. question

SET hive.exec.dynamic.partition = true;

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

INSERT OVERWRITE TABLE MyDb.foodratingspart PARTITION (name) SELECT food1, food2, food3, food4, id, name FROM foodratings;

SELECT MIN(food2) as MIN, MAX(food2) as MAX, AVG(food2) as average FROM foodratingspart WHERE name = 'Mel' OR name = 'Jill';

7. Question

LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces36481.txt' OVERWRITE INTO TABLE foodplaces;

SELECT foodplaces.places as Place, avg(foodratings.food4) as Average FROM foodplaces JOIN foodratings ON (foodratings.id=foodplaces.id) WHERE foodplaces.places='Soup Bowl' GROUP BY foodplaces.places;

8. Question Answer the following questions about the data formats used in Big data