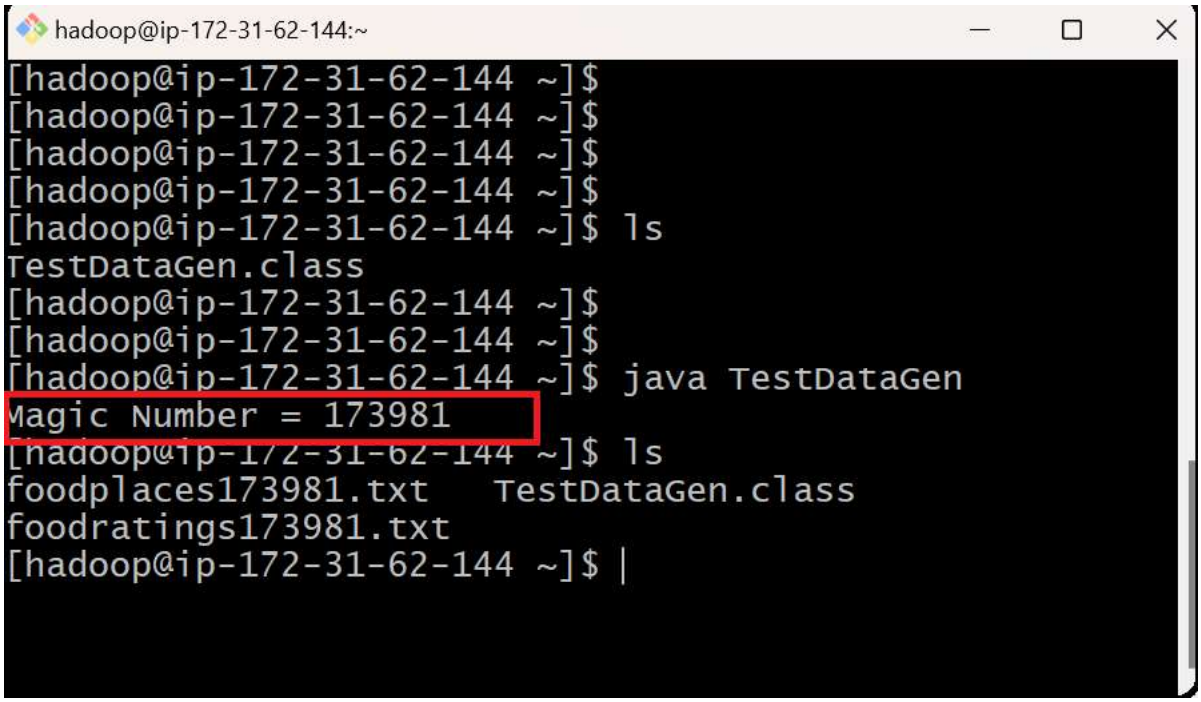
CWID: A20526972

CSP-554 Big Data Technologies

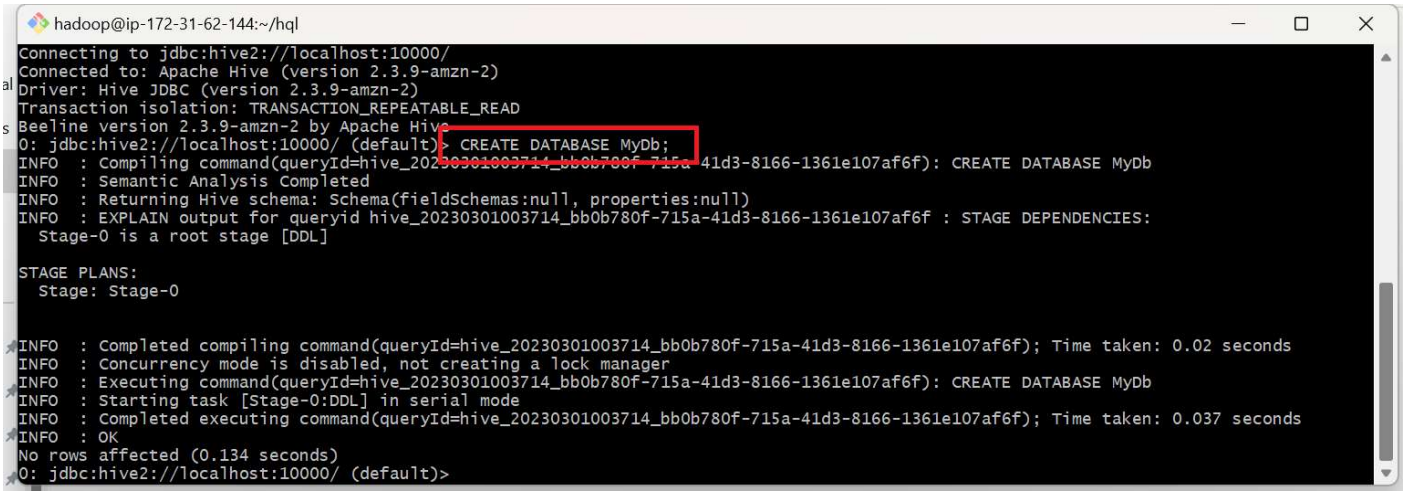
BAIRI ROHITH REDDY – Assignment #4

Magic Number :

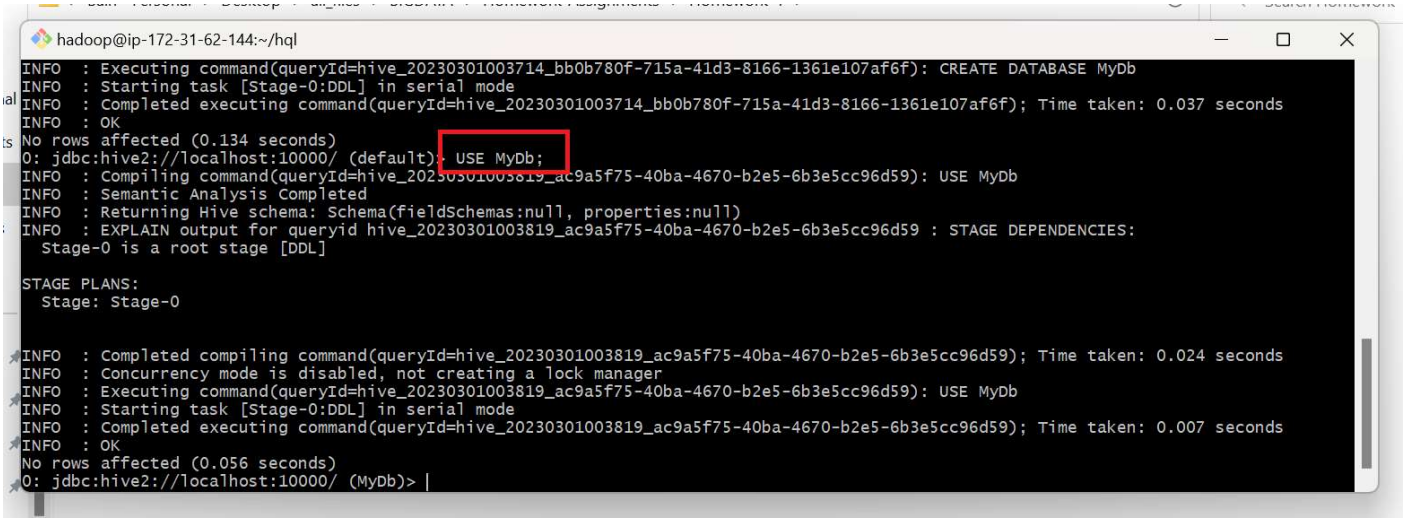


Exercise 1) **Create a Hive database called "MyDb".**

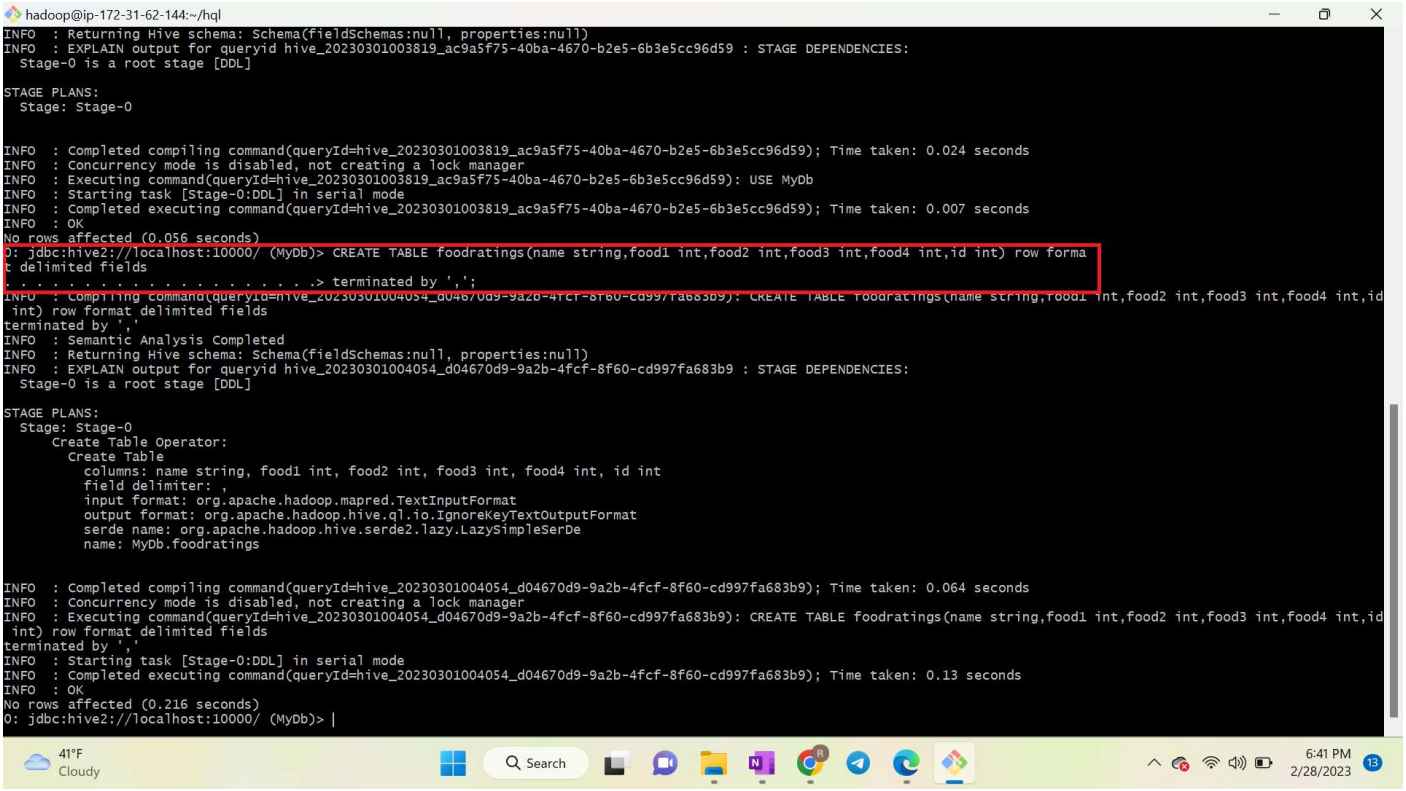
CREATE DATABASE MyDb;



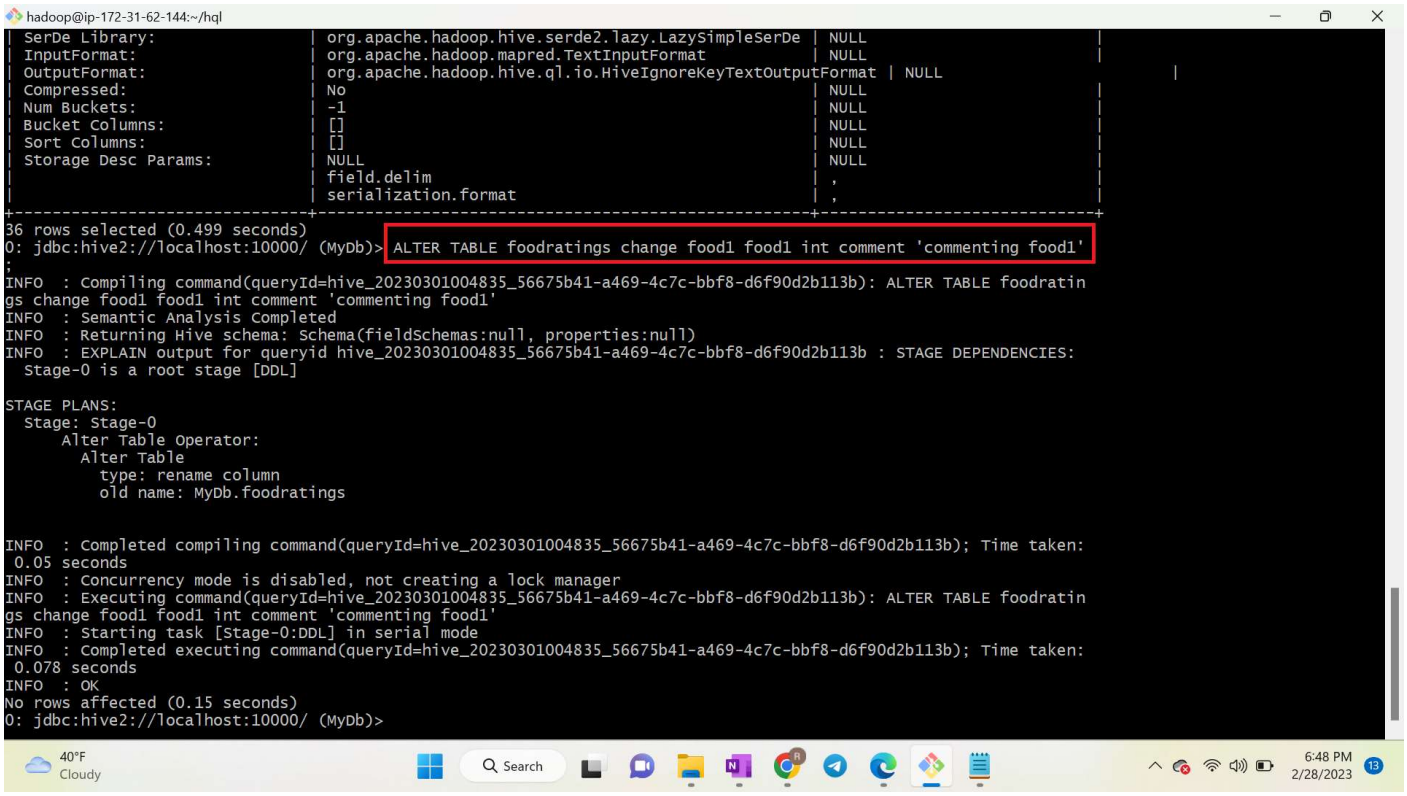
USE MyDB;



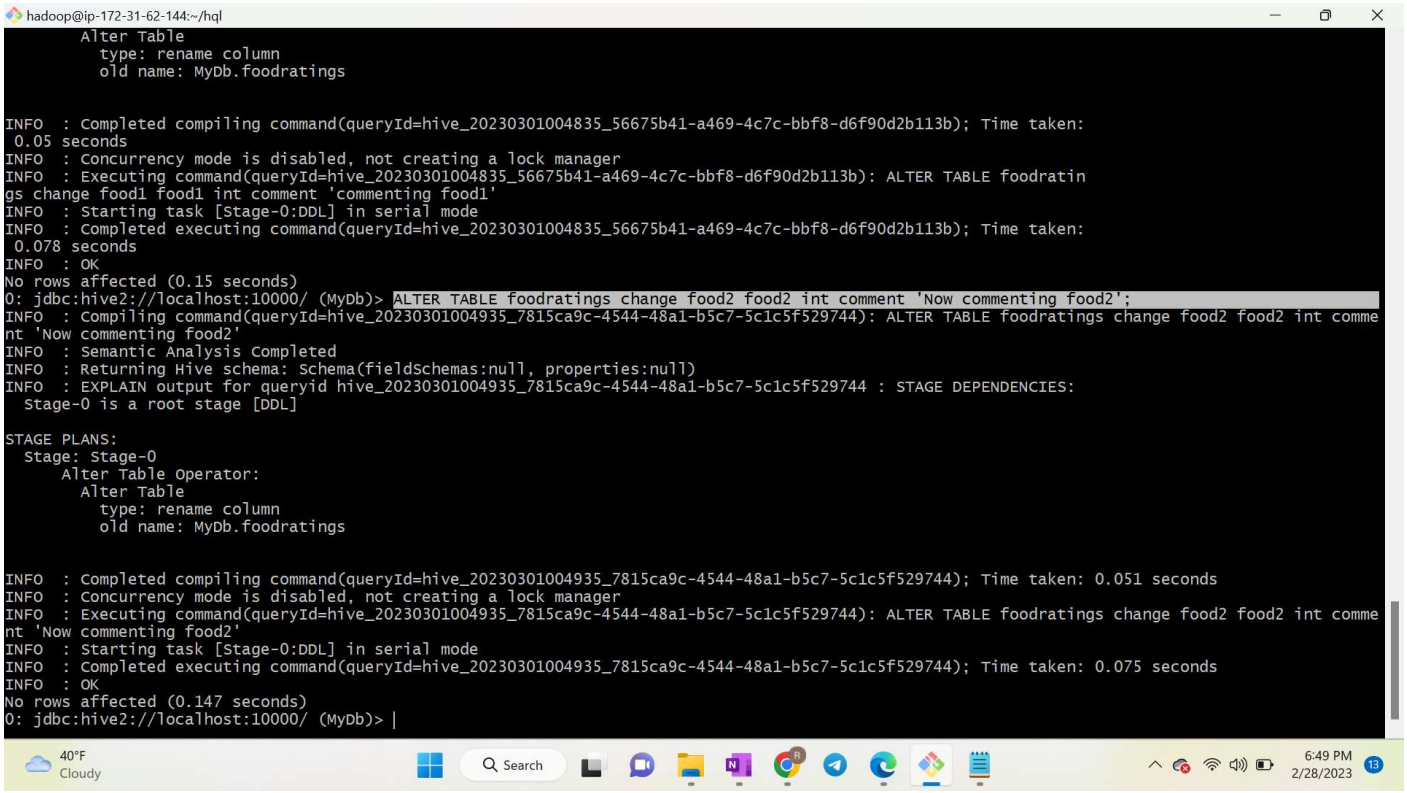
CREATE TABLE foodratings(name string,food1 int,food2 int,food3 int,food4 int,id int) row format delimited fields terminated by ',';



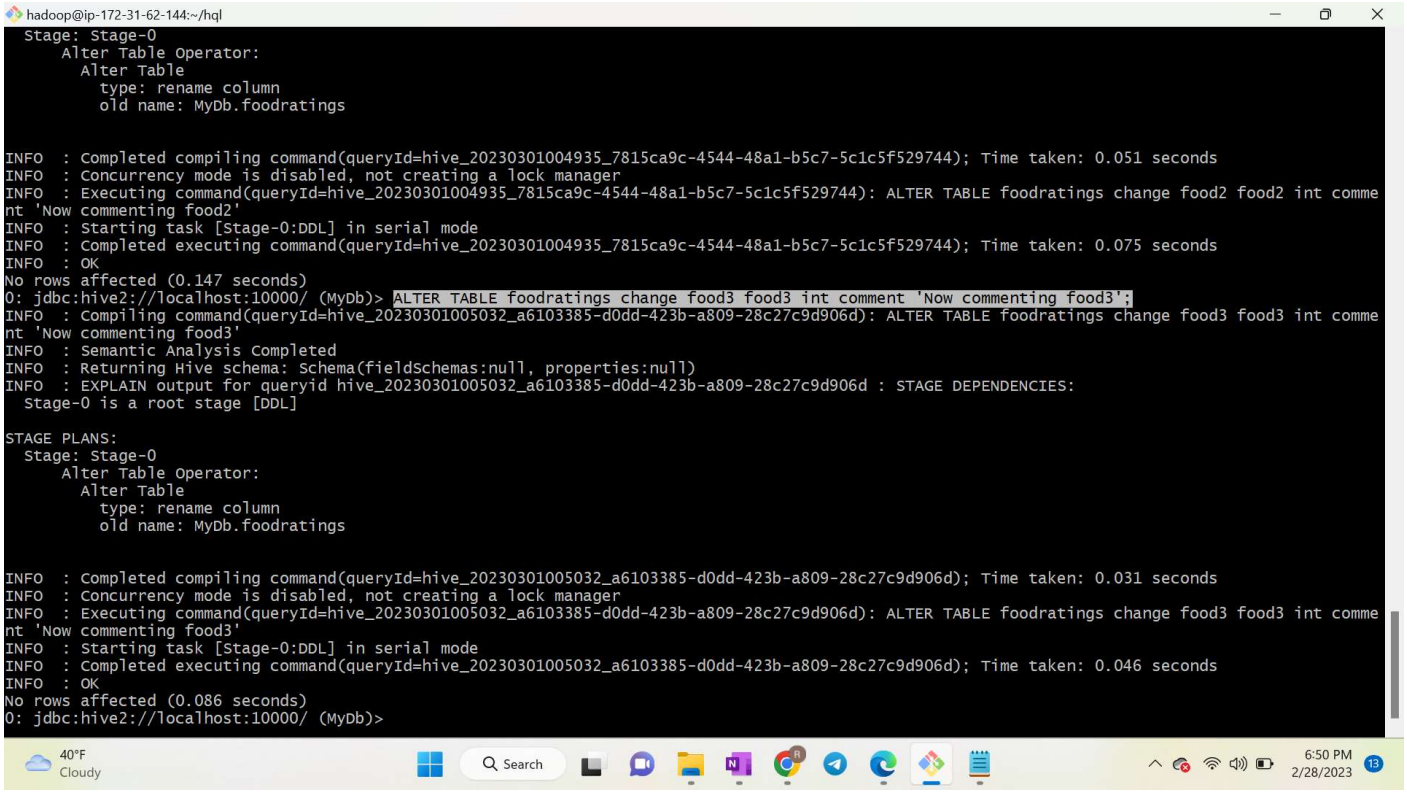
ALTER TABLE foodratings change food1 food1 int comment 'commenting food1';



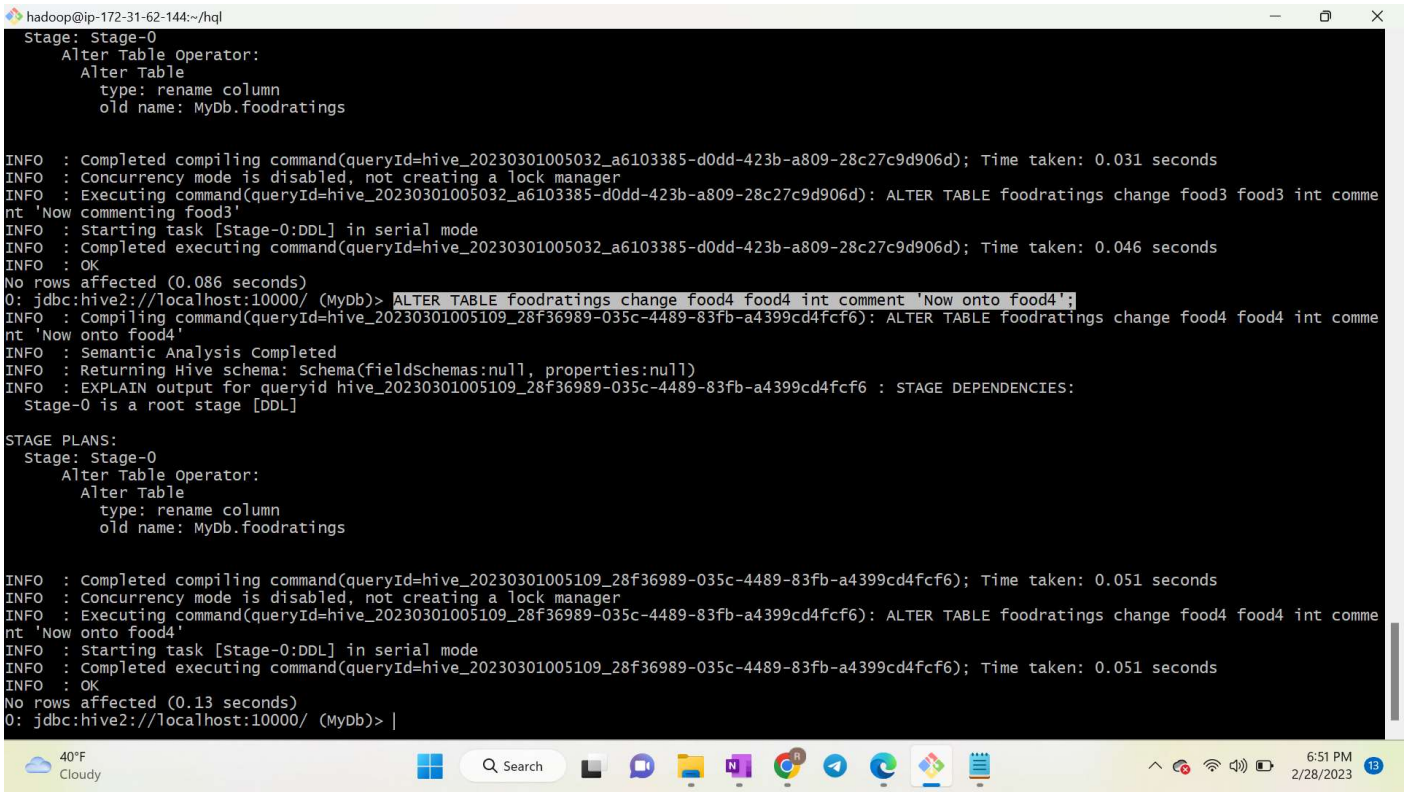
ALTER TABLE foodratings change food2 food2 int comment 'Now commenting food2';



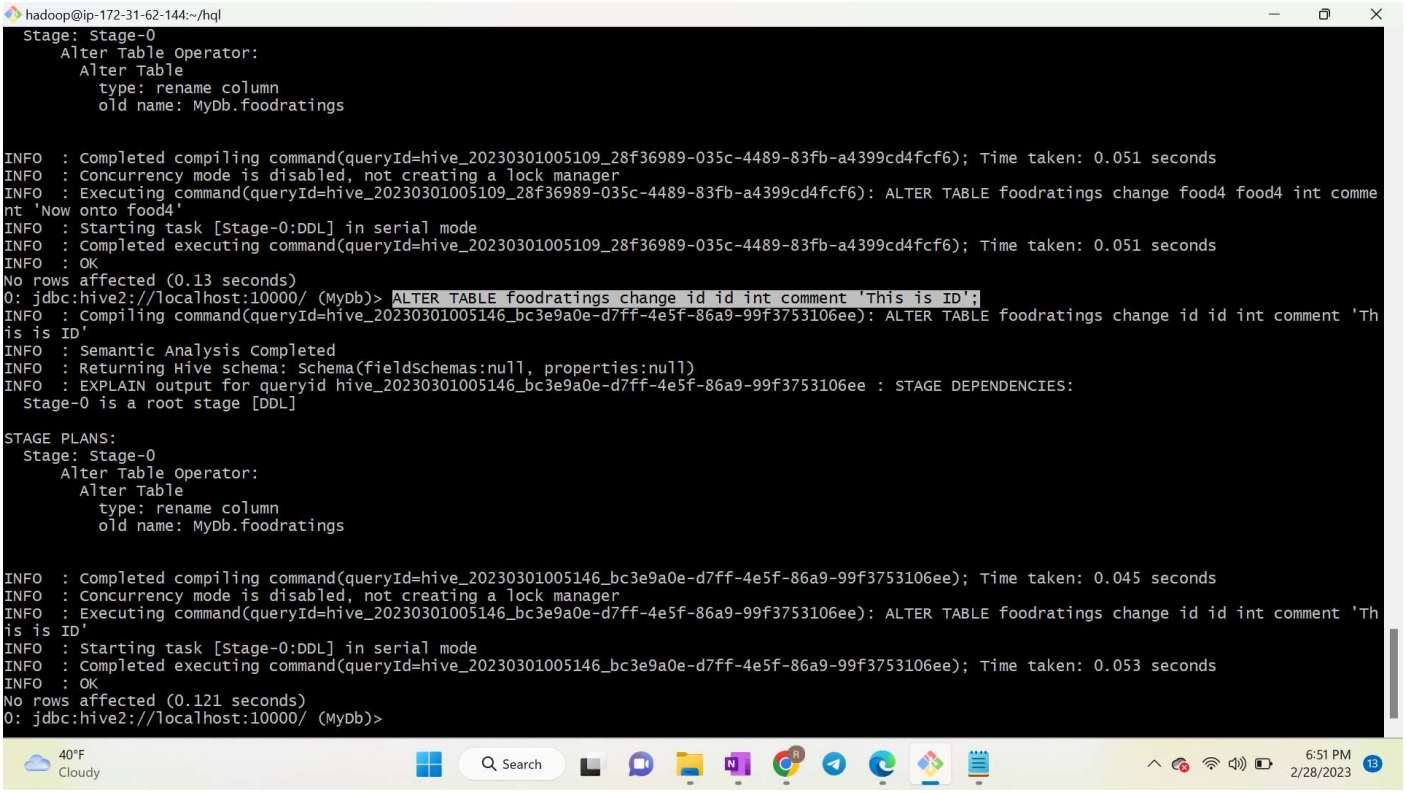
ALTER TABLE foodratings change food3 food3 int comment 'Now commenting food3';



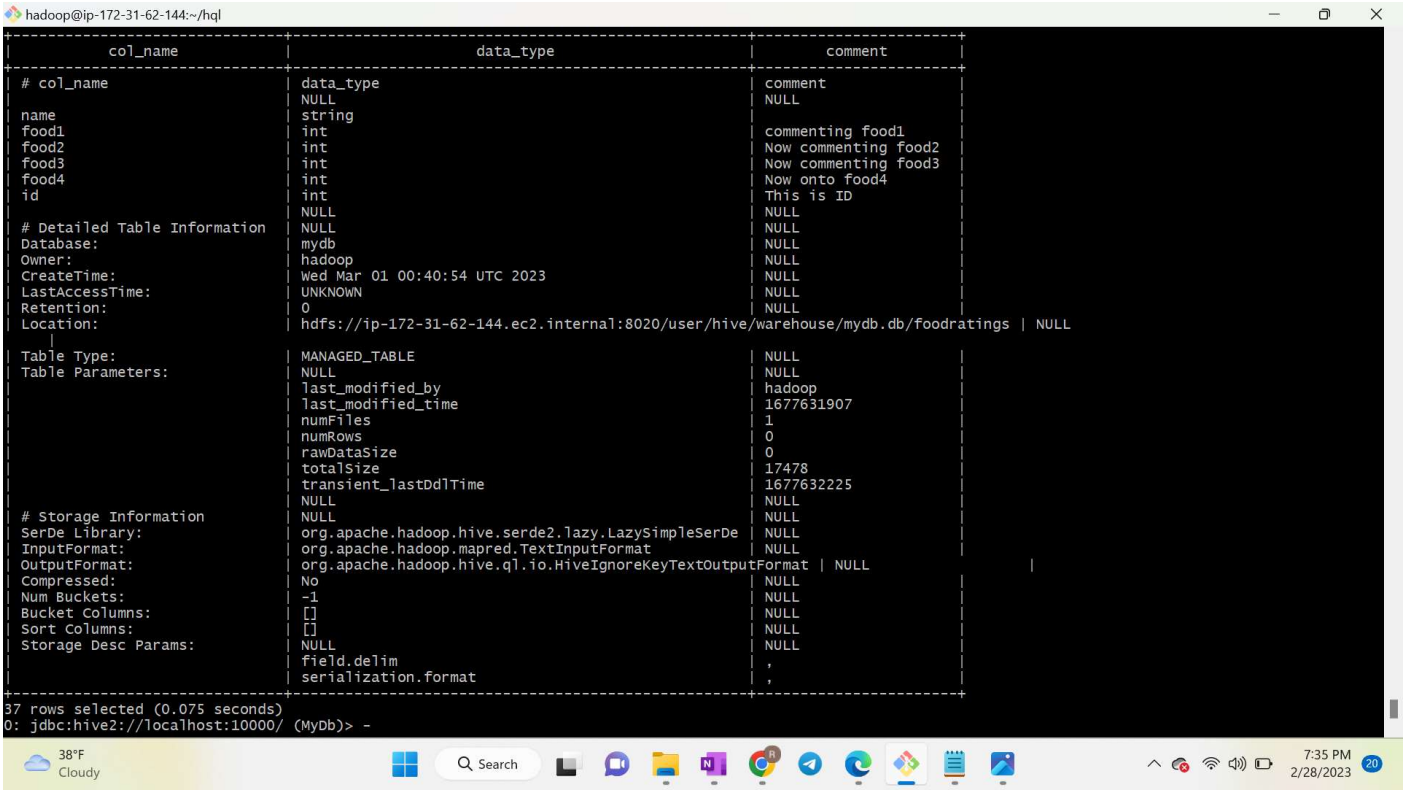
ALTER TABLE foodratings change food4 food4 int comment 'Now onto food4';



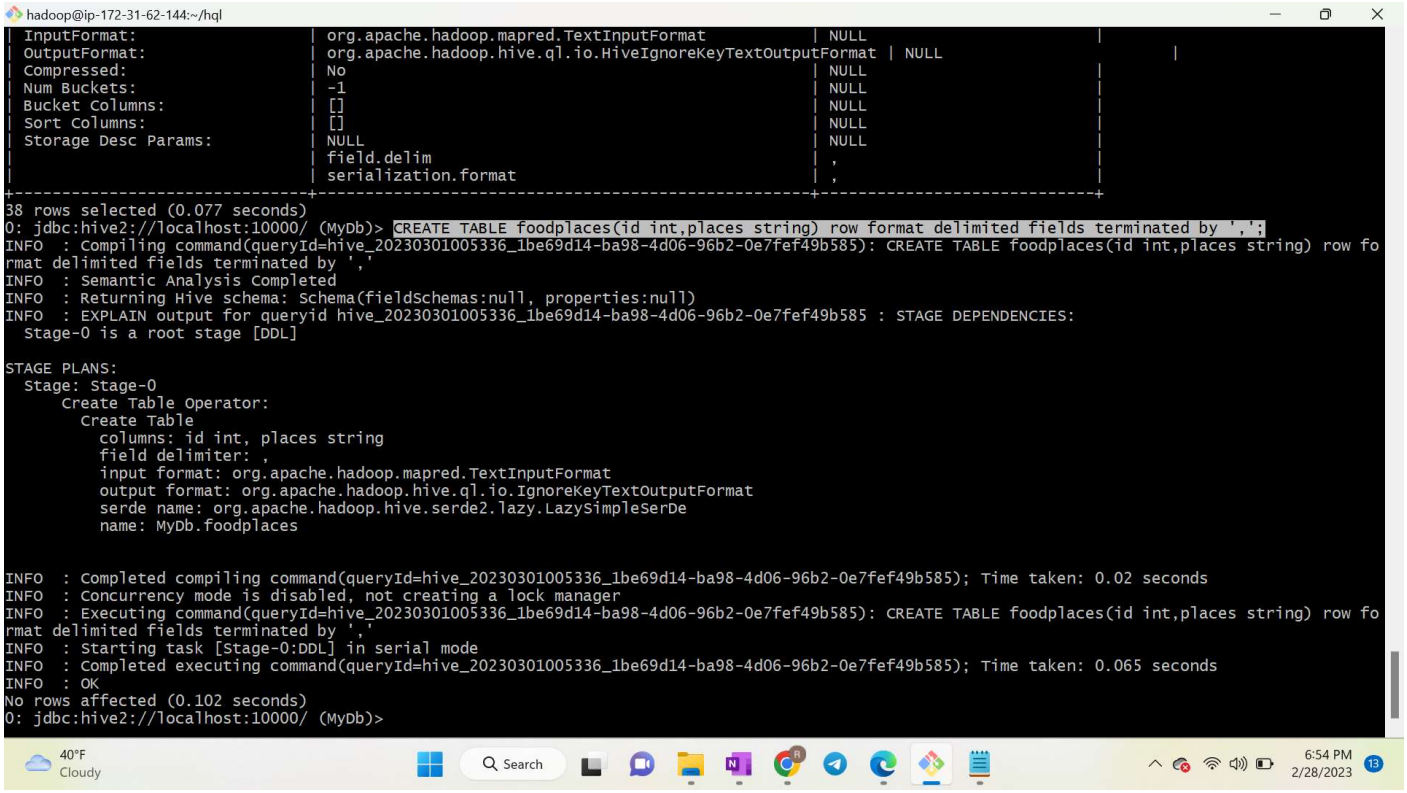
ALTER TABLE foodratings change id id int comment 'This is ID';



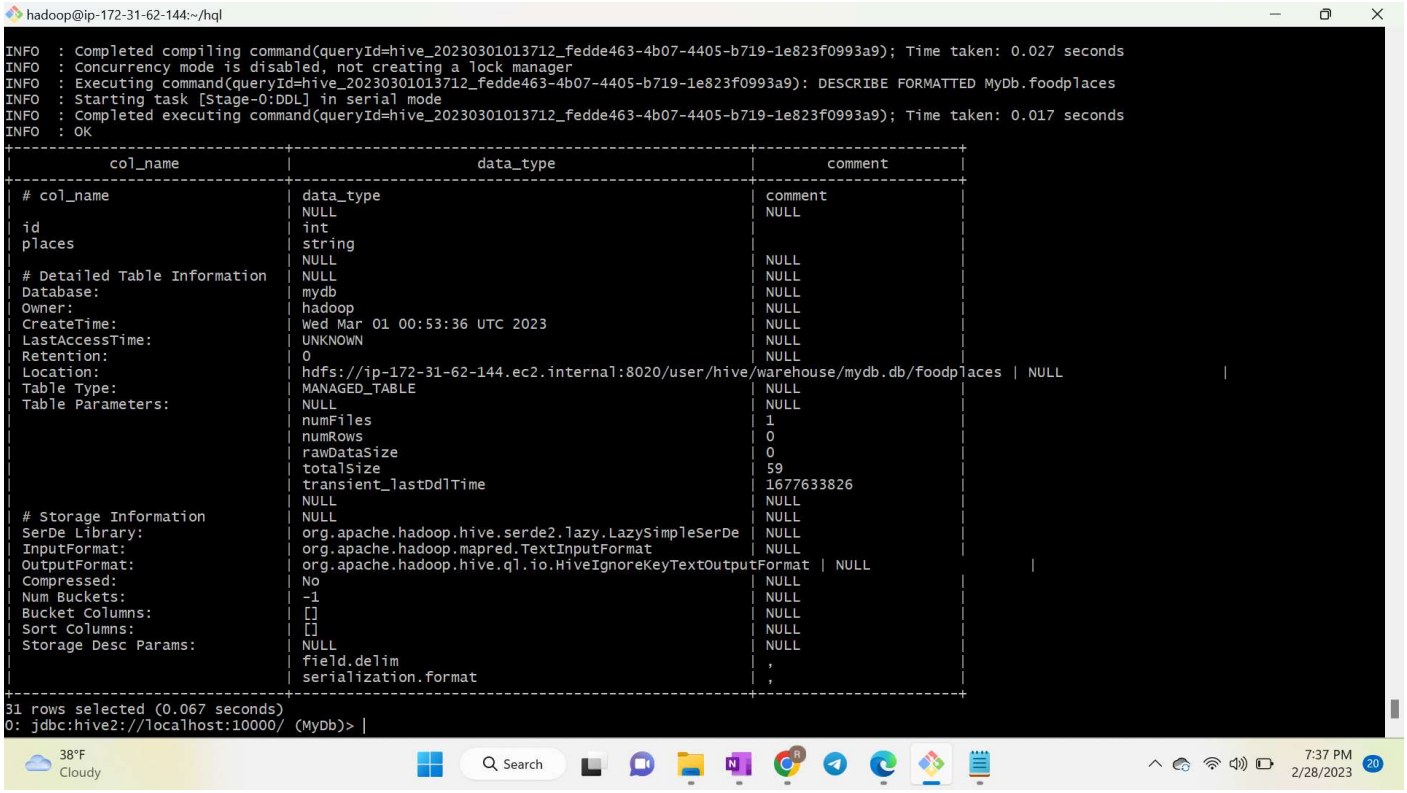
Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodratings;'



CREATE TABLE foodplaces(id int,places string) row format delimited fields terminated by ',';



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

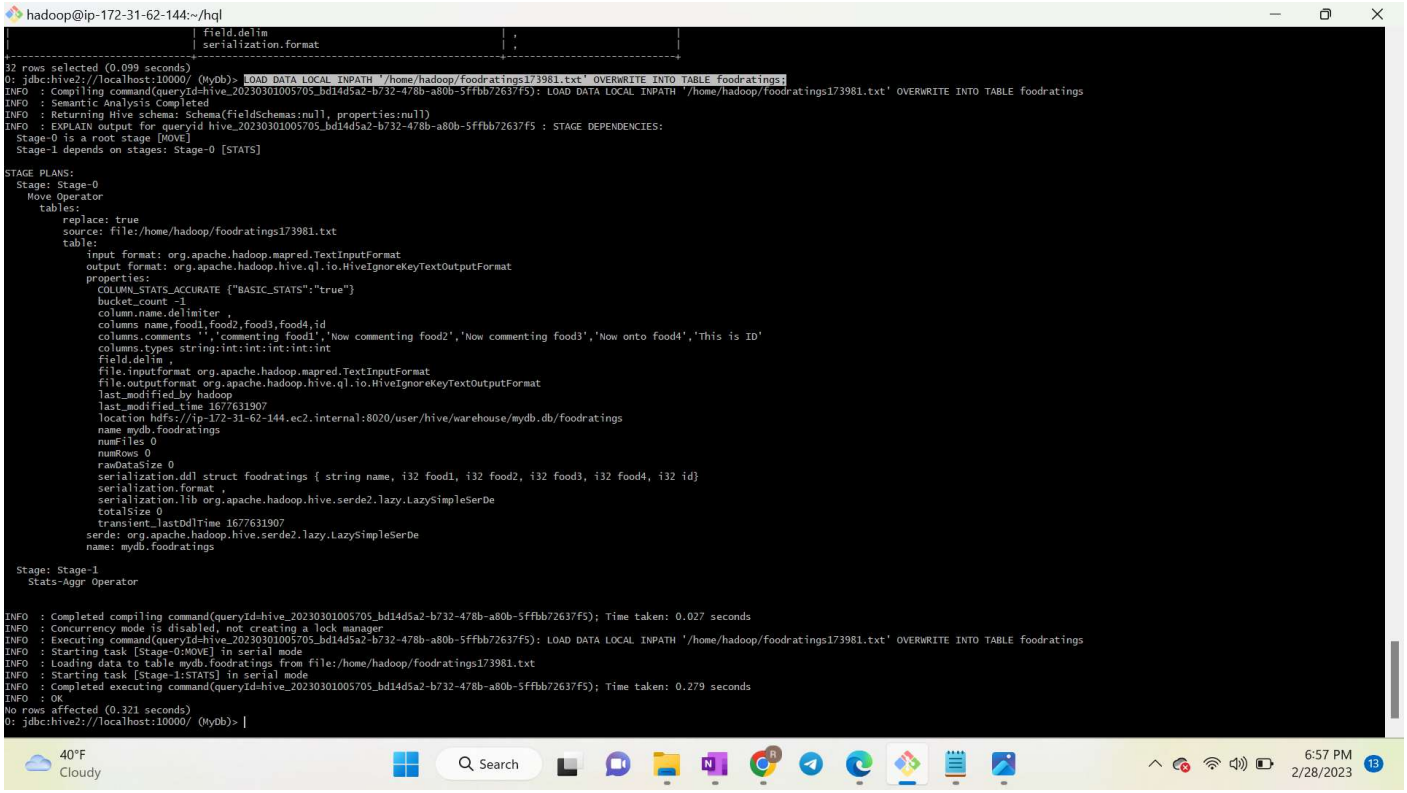


CWID: A20526972

Exercise 2)

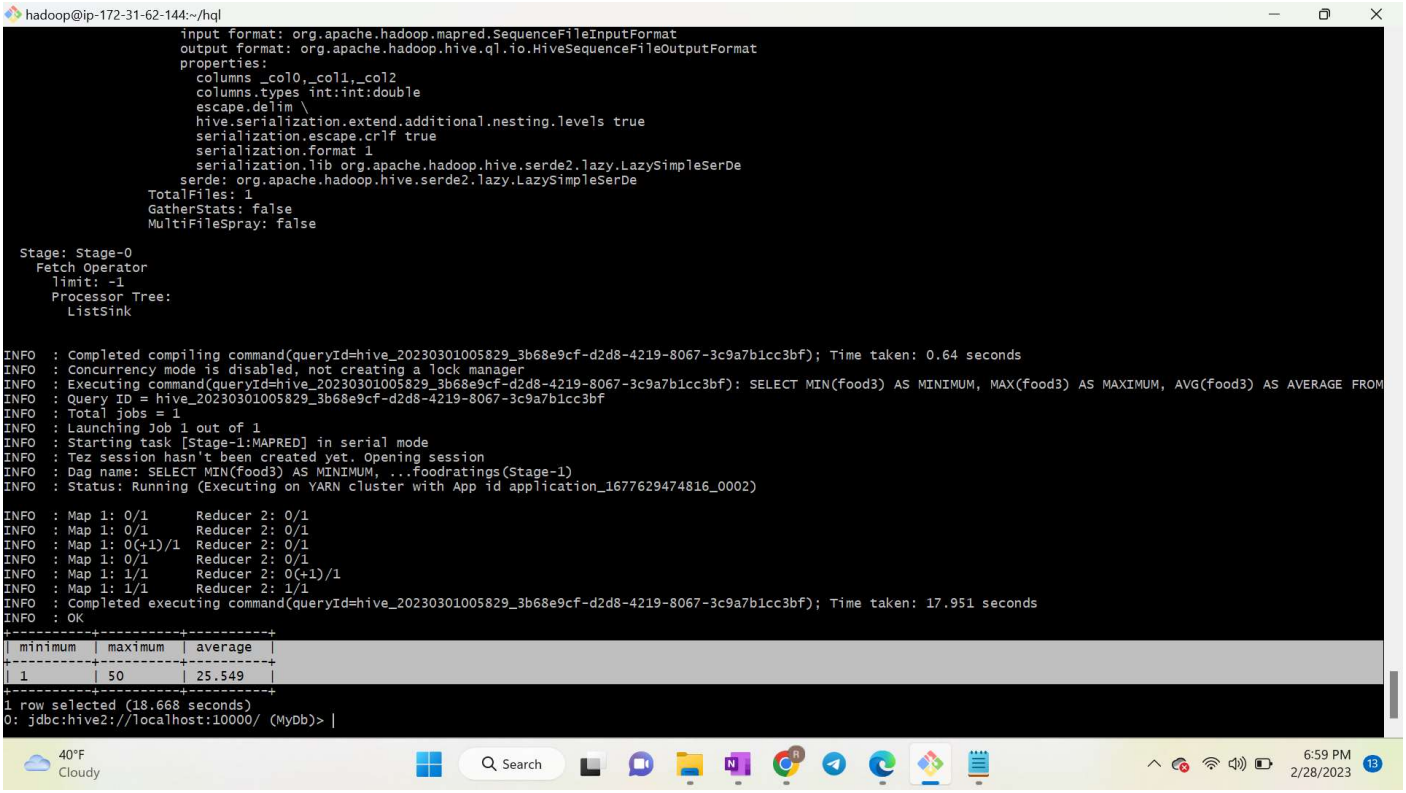
Load the foodratings.txt file created using TestDataGen from your local file system into the foodratings table.

LOAD DATA LOCAL INPATH '/home/hadoop/foodratings76011.txt' OVERWRITE INTO TABLE foodratings;



CWID: A20526972

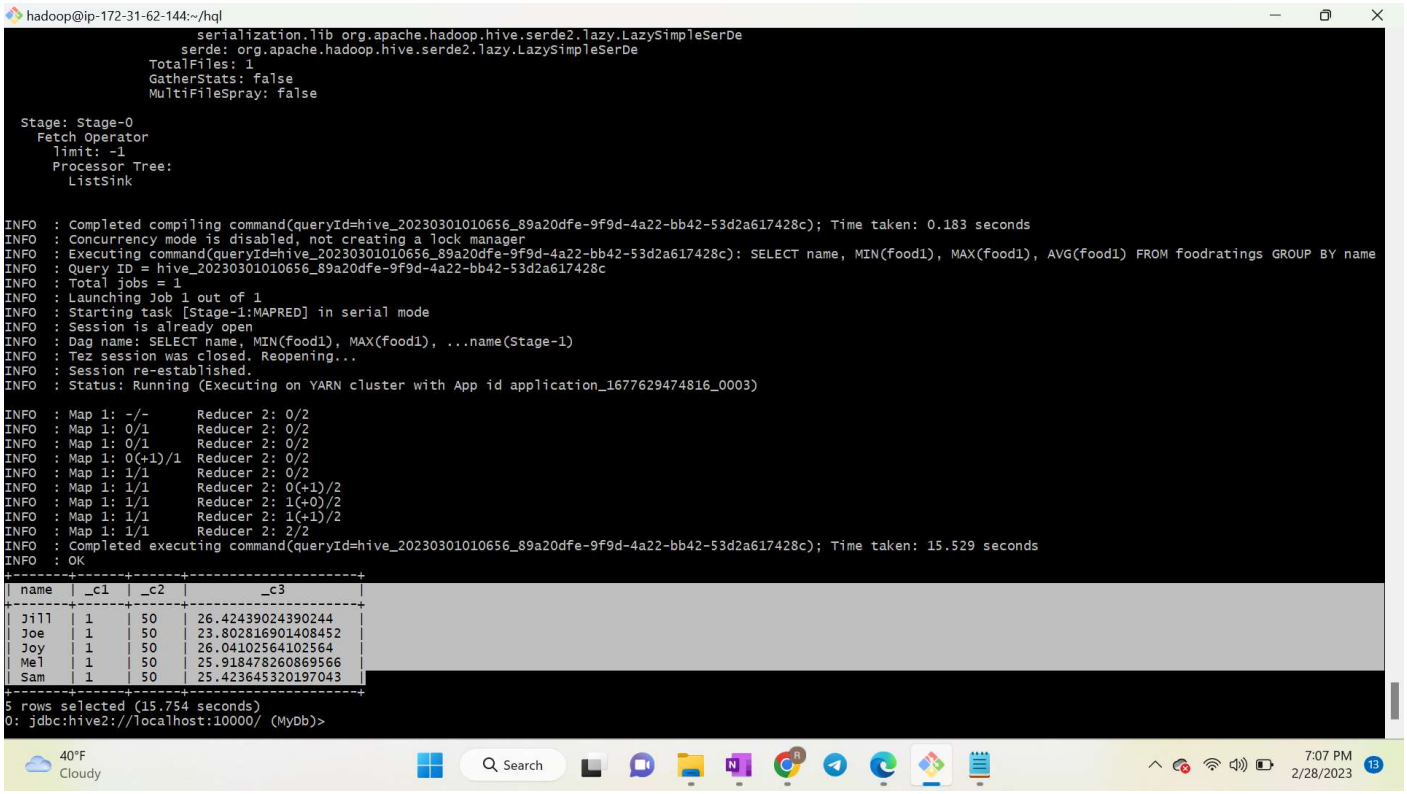
SELECT MIN(food3) AS MINIMUM, MAX(food3) AS MAXIMUM, AVG(food3) AS AVERAGE FROM foodratings;



Exercise 3)

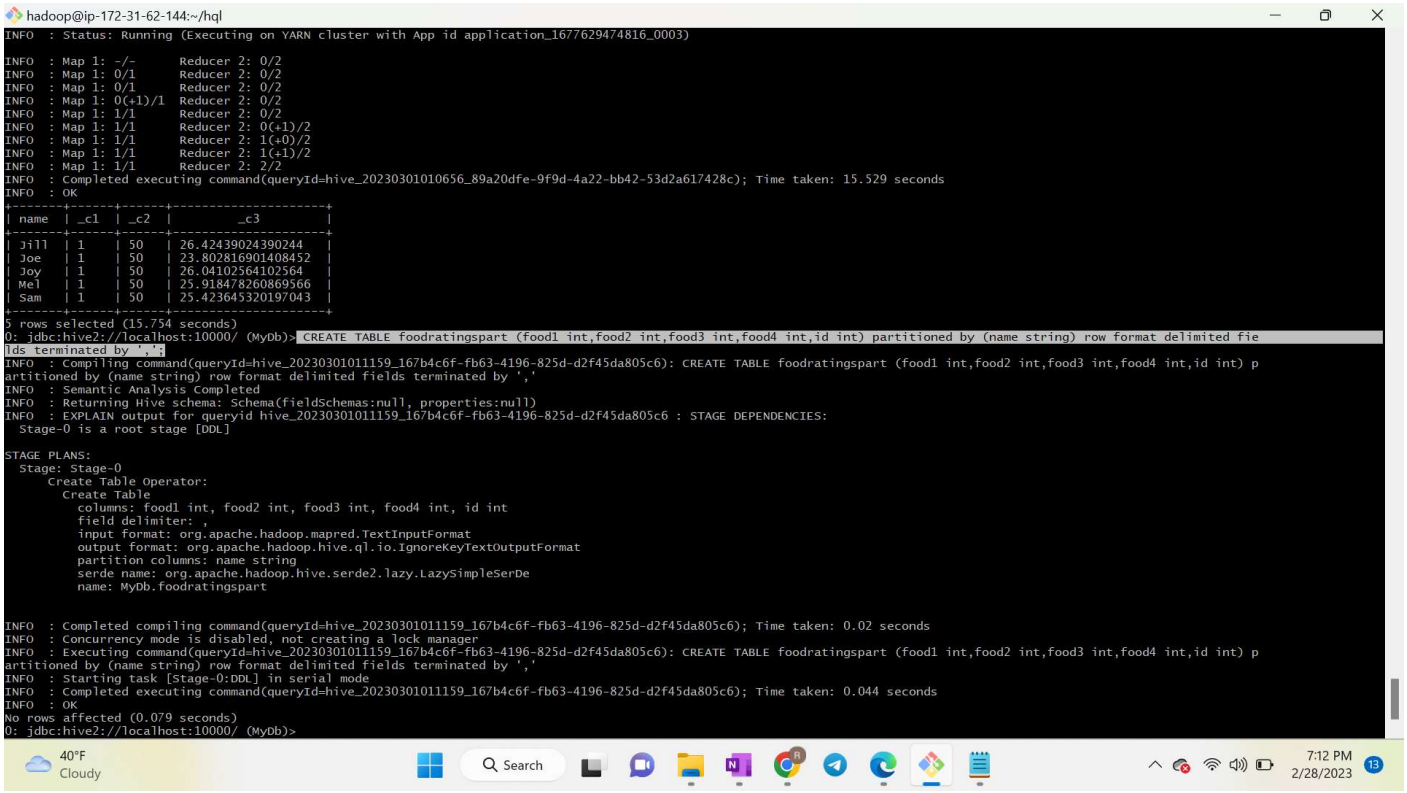
Execute a hive command to output the min, max and average of the values of the food1 column grouped by the first column 'name'.

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

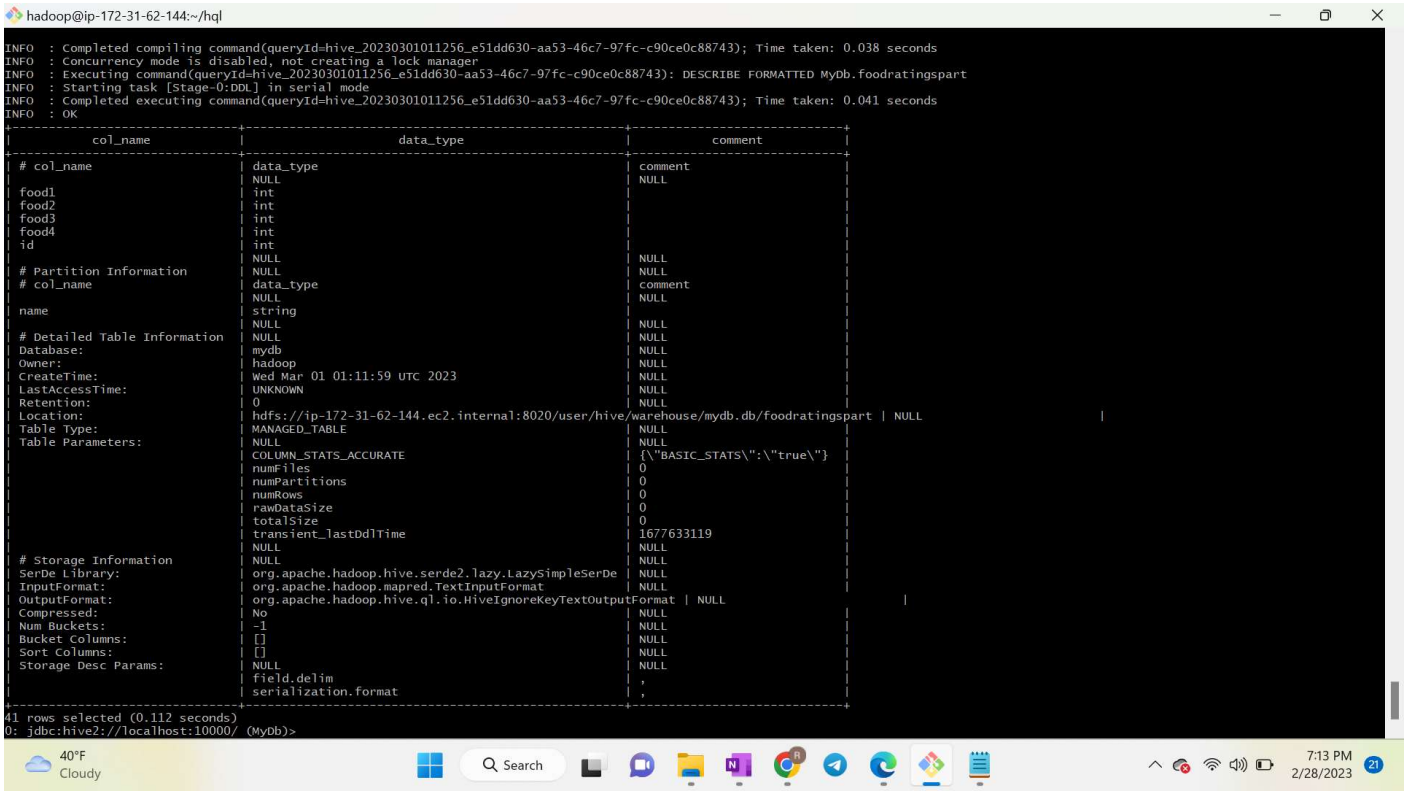


Exercise 4)

In MyDb create a partitioned table called ‘foodratingspart’.



Execute a Hive command of ‘DESCRIBE FORMATTED MyDb.foodratingspart;’



Exercise 5)

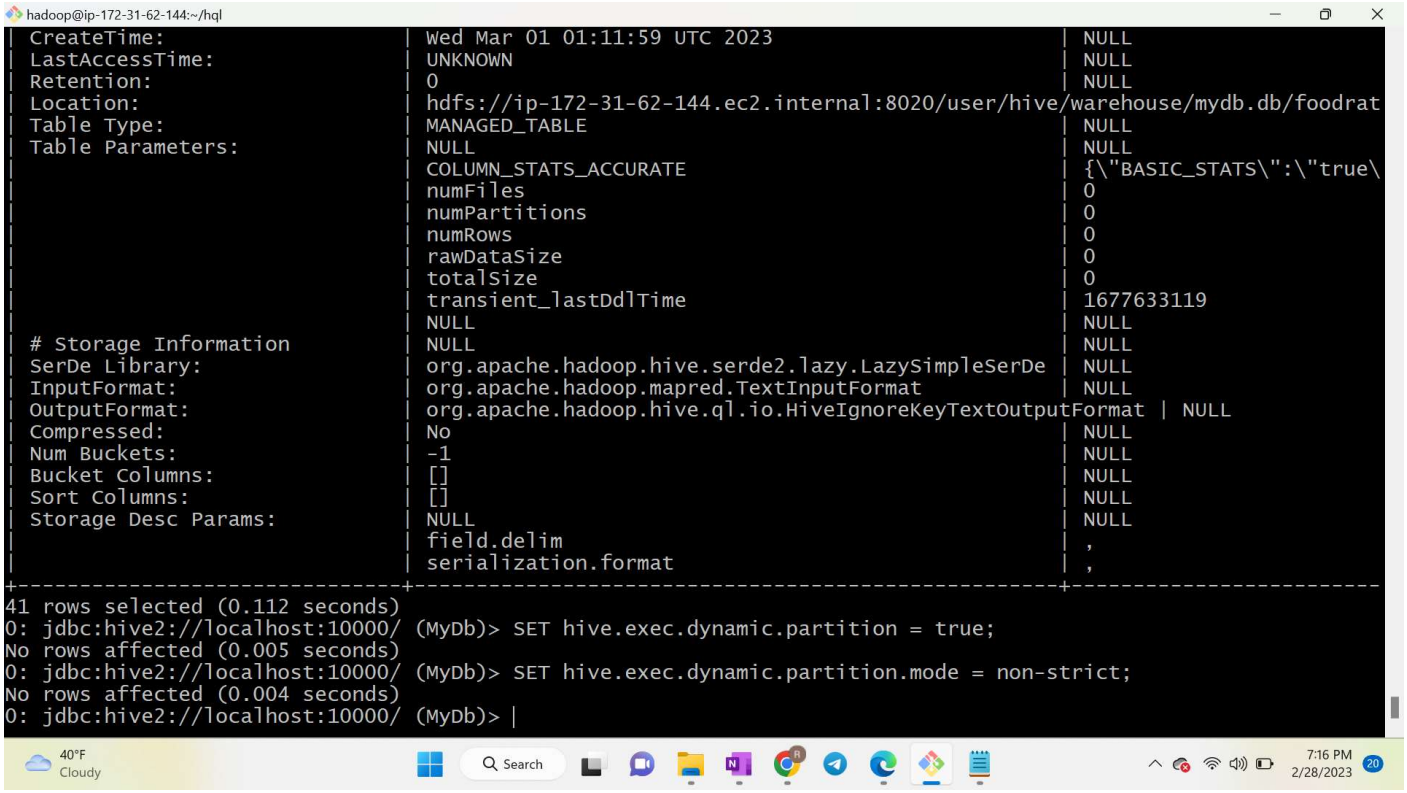
Assume that the number of food critics is relatively small, say less than 10 and the number places to eat is very large, say more than 10,000. In a few short sentences explain why using the (critic) name is a good choice for a partition field while using the place id is not.

Exercise 6)

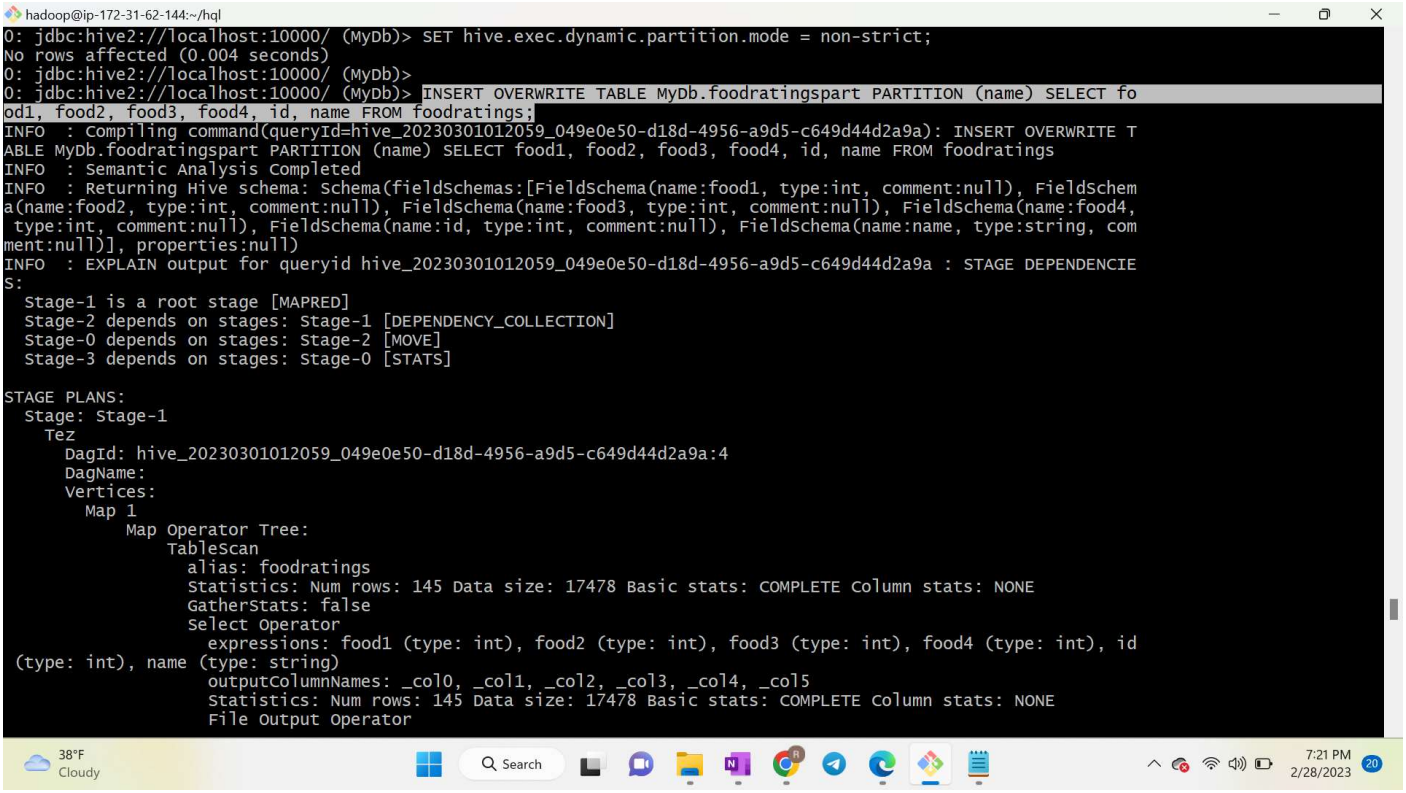
Execute a hive command to output the min, max and average of the values of the food2 column of MyDB.foodratingspart where the food critic 'name' is either Mel or Jill.

SET hive.exec.dynamic.partition = true;

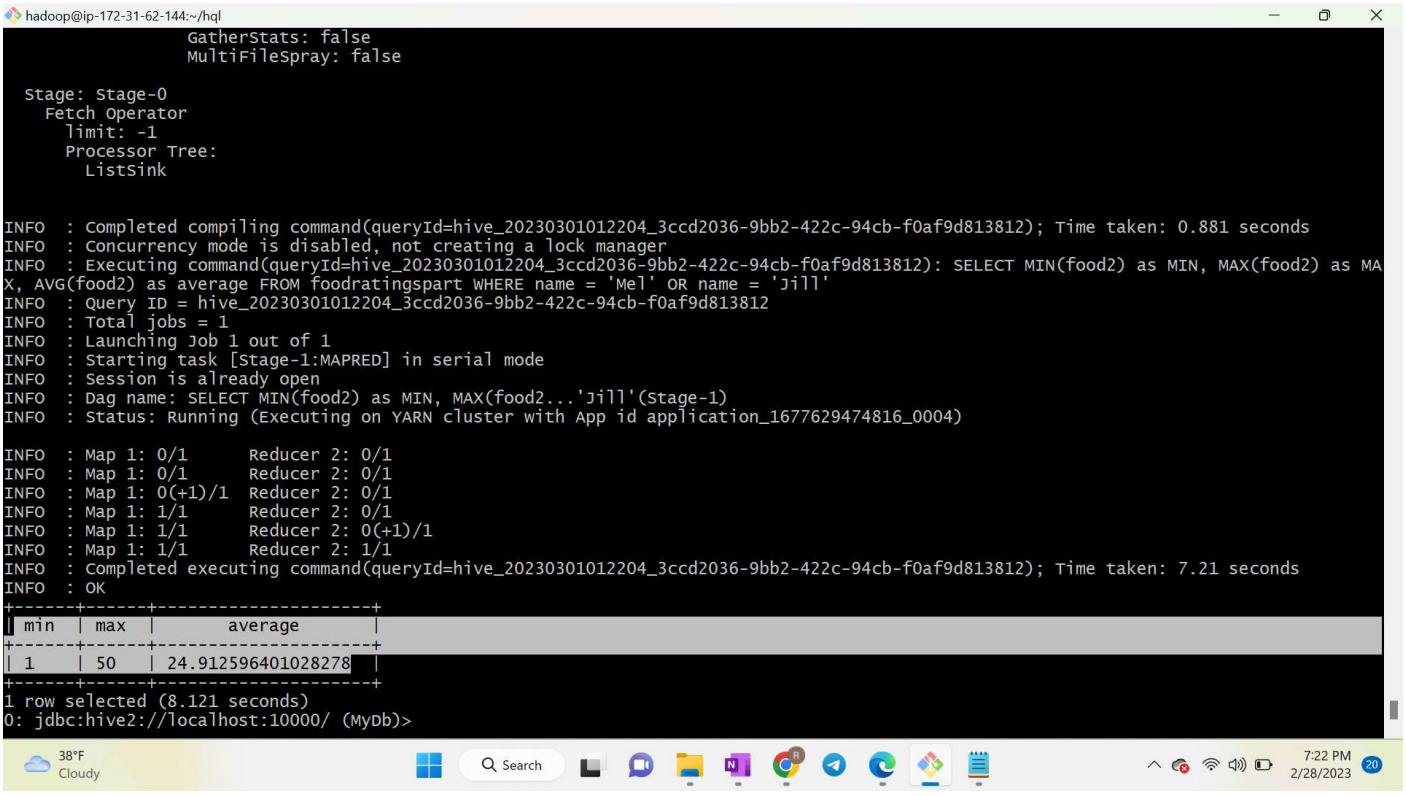
SET hive.exec.dynamic.partition.mode = non-strict;



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



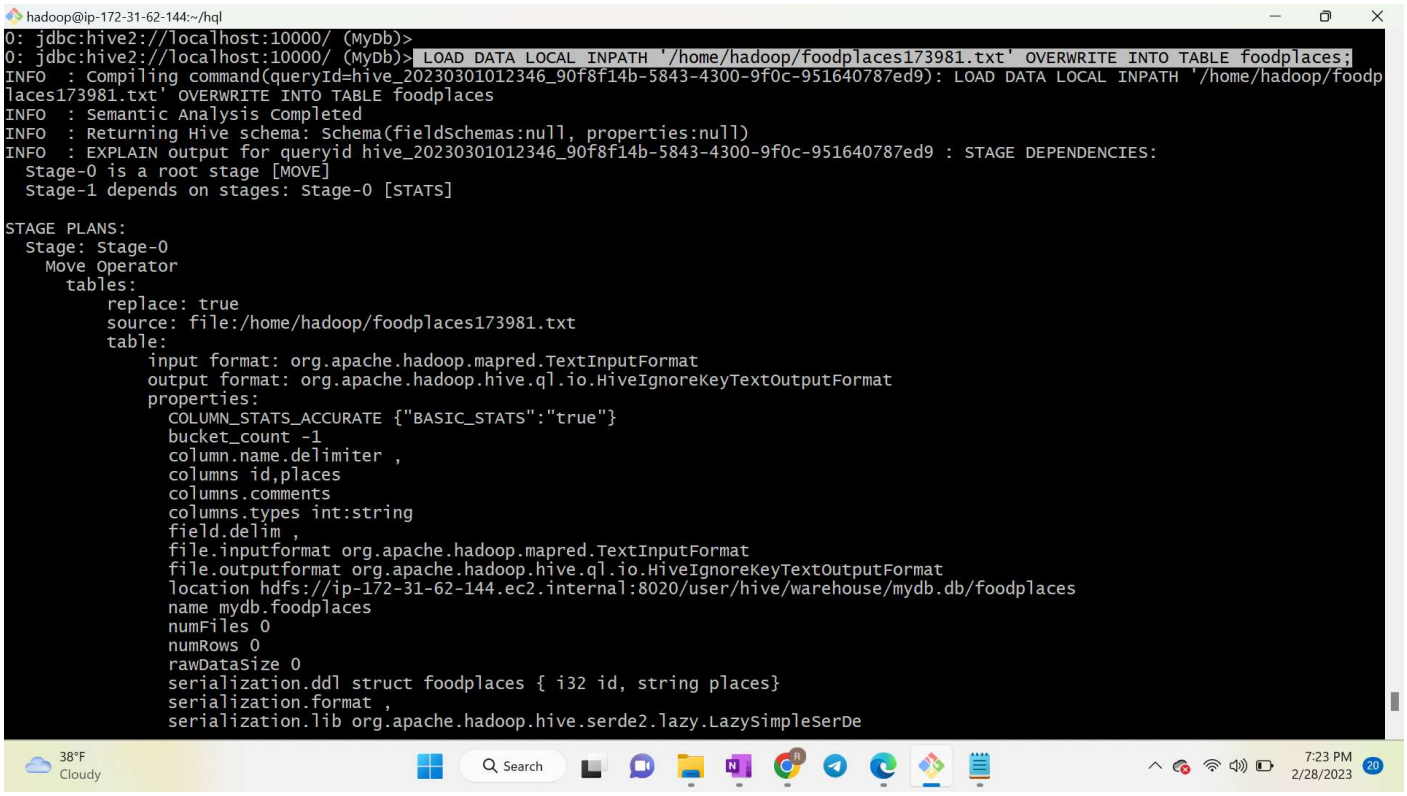
SELECT MIN(food2) as MIN, MAX(food2) as MAX, AVG(food2) as average FROM foodratingspart WHERE name = 'Mel' OR name = 'Jill';



Exercise 7)

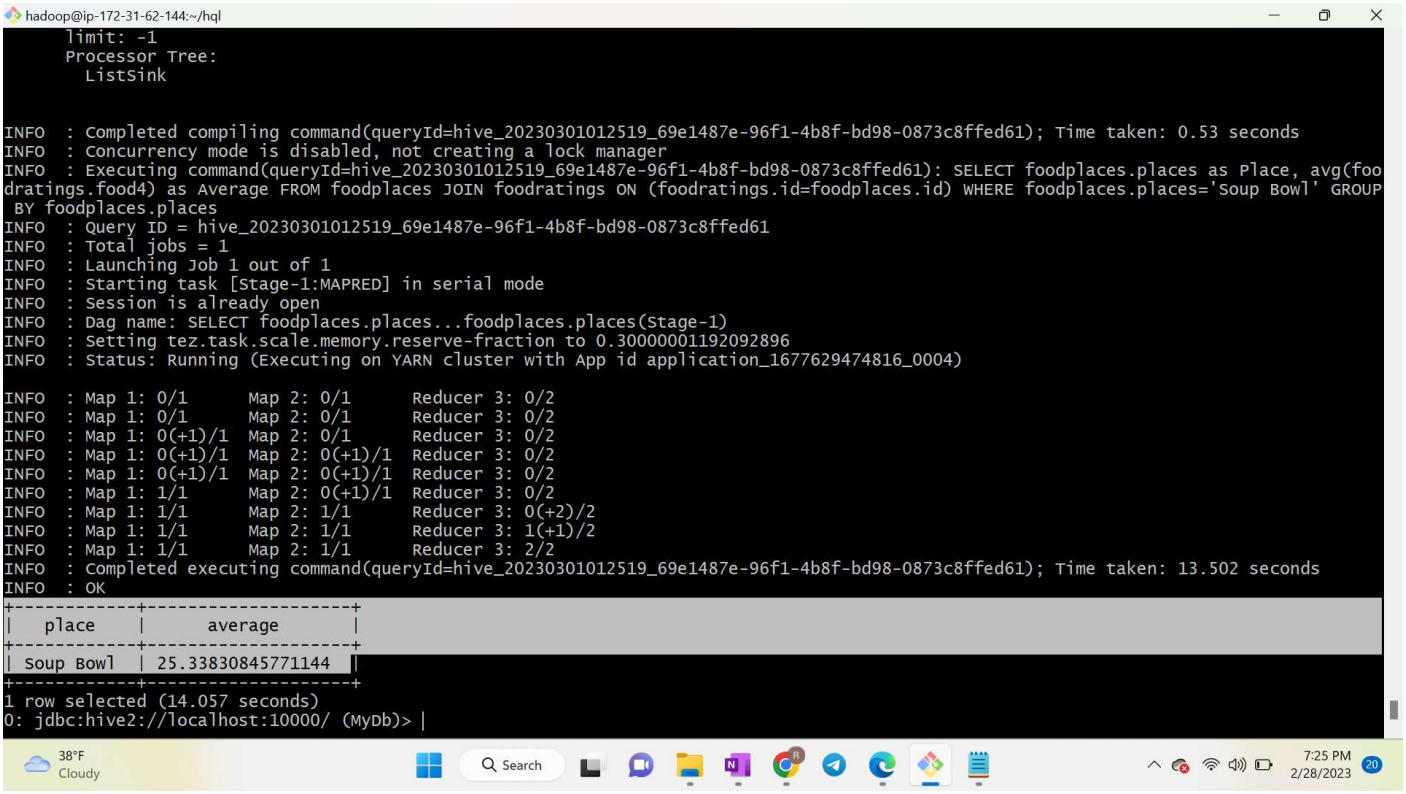
Load the foodplaces.txt file created using TestDataGen from your local file system into the foodplaces table.

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?**

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

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

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