

Magic Number: 97956

1.

CREATE DATABASE MyDb;

SHOW DATABASES;

CREATE EXTERNAL TABLE IF NOT EXISTS MYDB.foodplaces(

id INT Comment 'Restaurant ID (PK)',

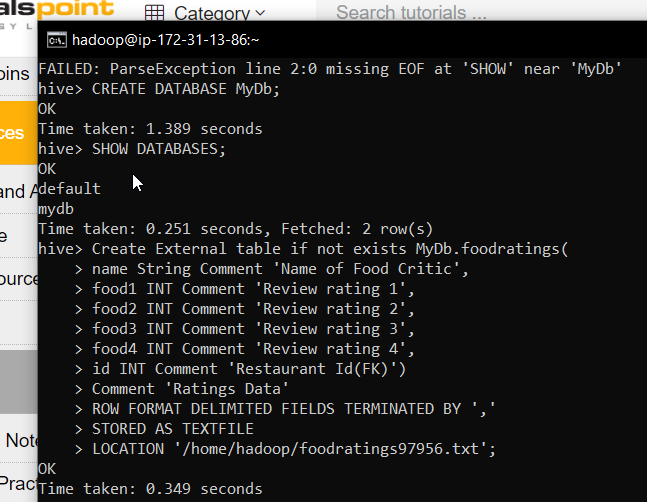
place STRING Comment 'Name of Res')

Comment 'Rest details'

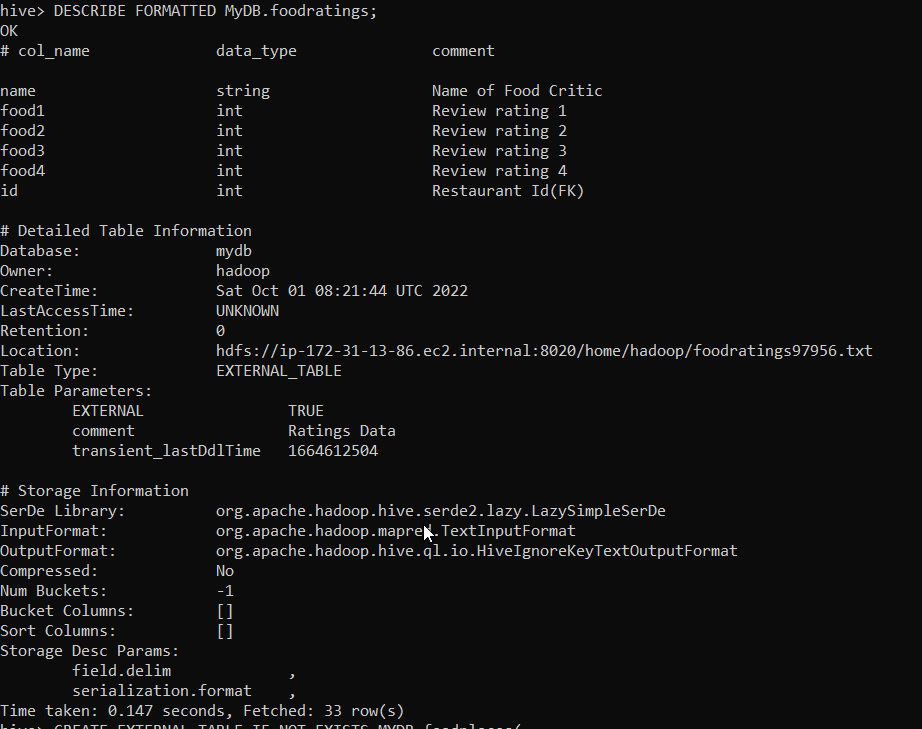
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/home/hadoop/foodplaces97956.txt';



DESCRIBE FORMATTED MyDb.foodratings;



CREATE EXTERNAL TABLE IF NOT EXISTS MYDB.foodplaces(

id INT Comment 'Restaurant ID (PK)',

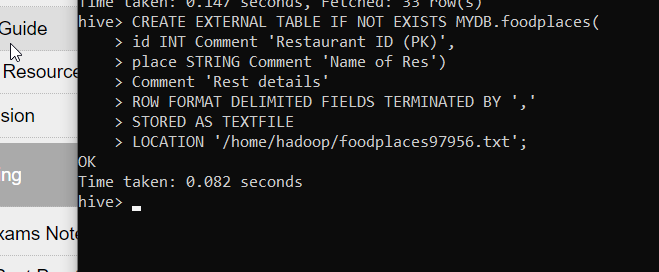
place STRING Comment 'Name of Res')

Comment 'Rest details'

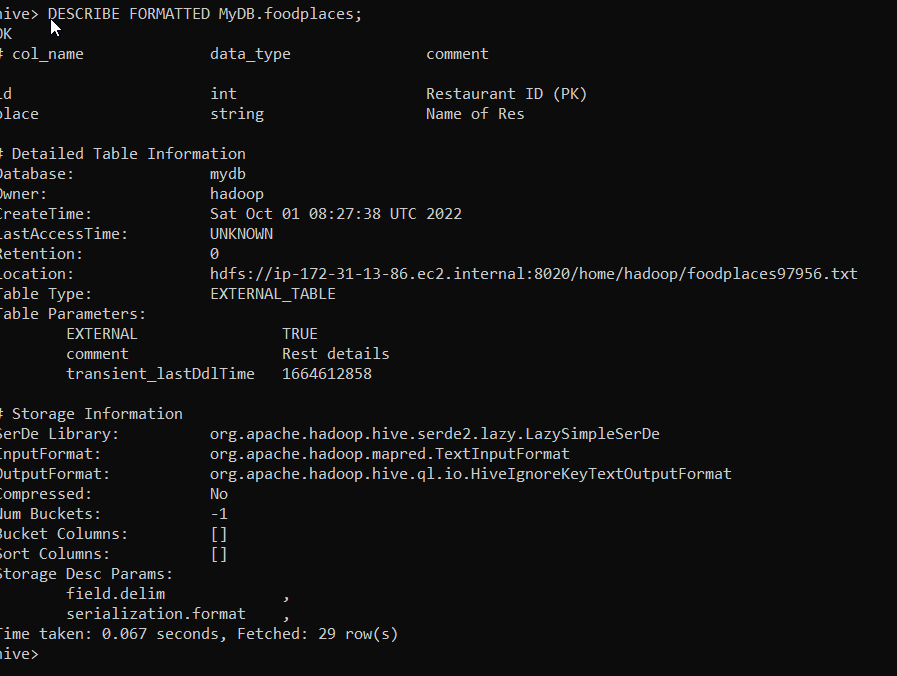
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/home/hadoop/foodplaces97956.txt';

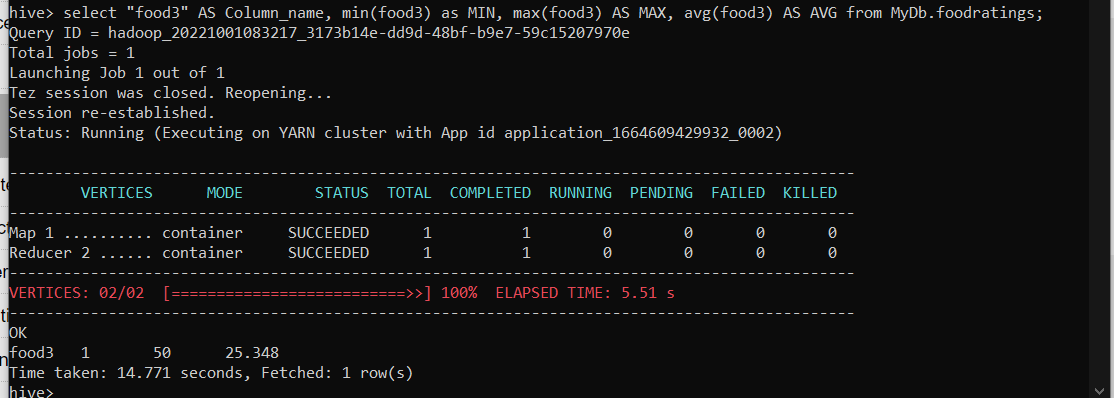


DESCRIBE FORMATTED MyDb.foodplaces;



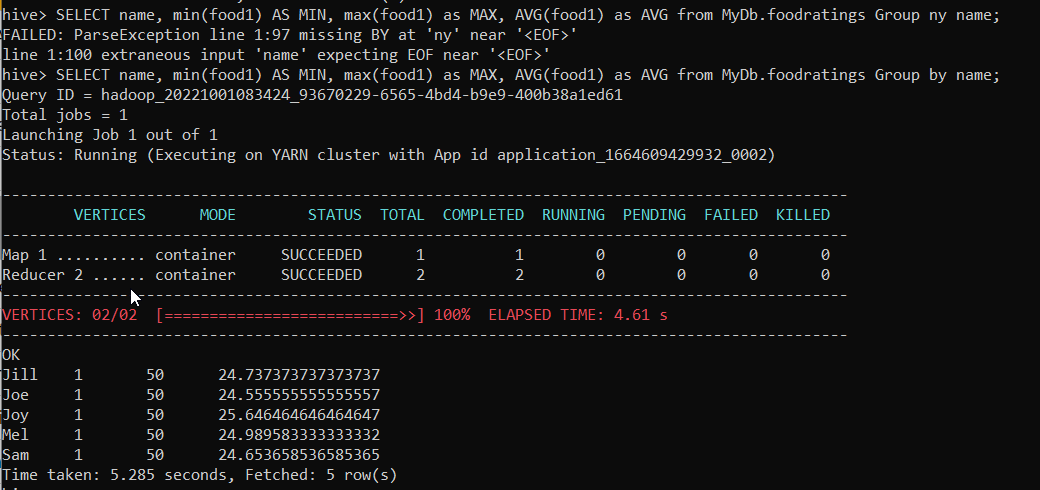
2.

SELECT name, min(food3) AS MIN, max(food3) as MAX, AVG(food3) as AVG from MyDb.foodratings;



3.

SELECT name, min(food1) AS MIN, max(food1) as MAX, AVG(food1) as AVG from MyDb.foodratings Group by name;



4.

CREATE EXTERNAL TABLE IF NOT EXISTS MyDb.foodratingspart (

food1 INT Comment 'Review rating 1',

food2 INT Comment 'Review rating 2',

food3 INT Comment 'Review rating 3',

food4 INT Comment 'Review rating 4',

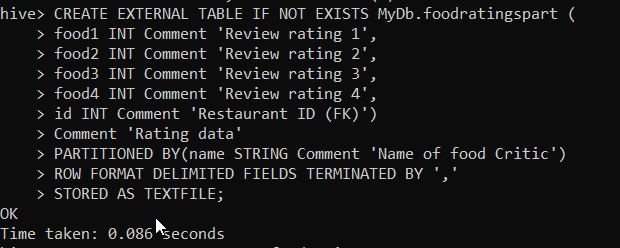
id INT Comment 'Restaurant ID (FK)')

Comment 'Rating data'

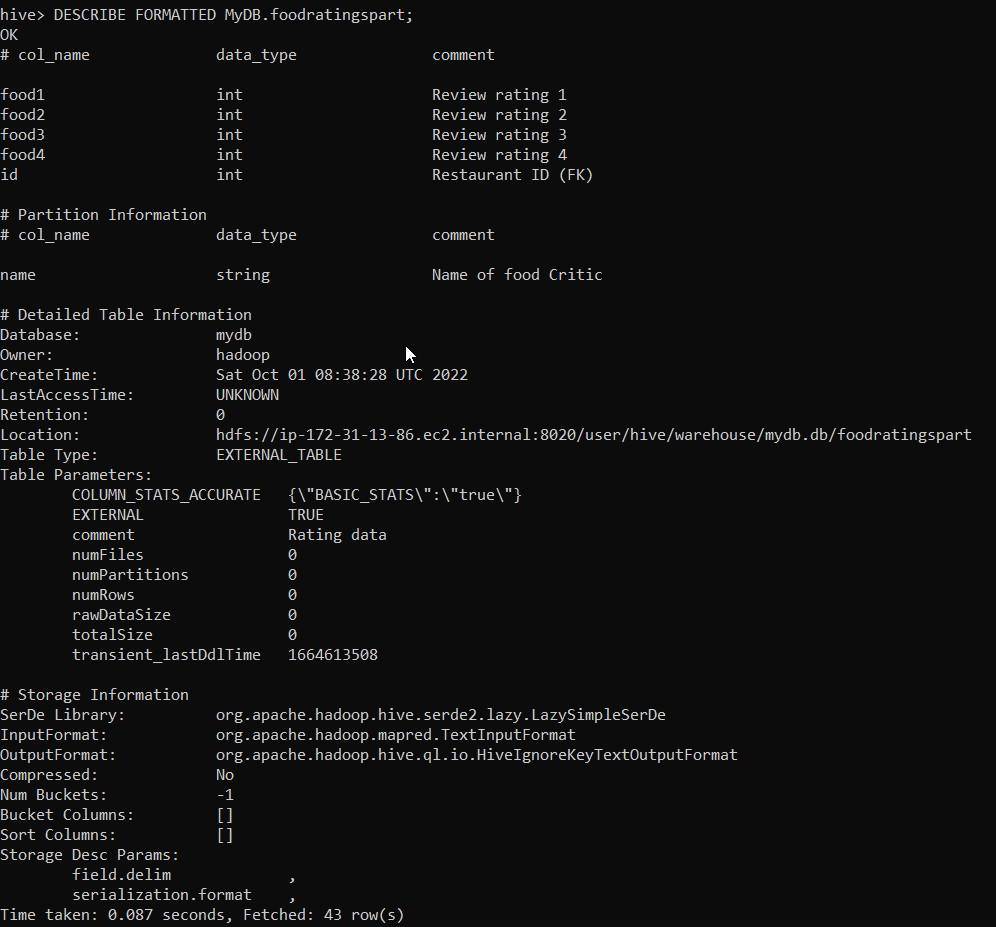
PARTITIONED BY(name STRING Comment 'Name of food Critic')

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE;



DESCRIBE FORMATTED MyDB.foodratingspart;



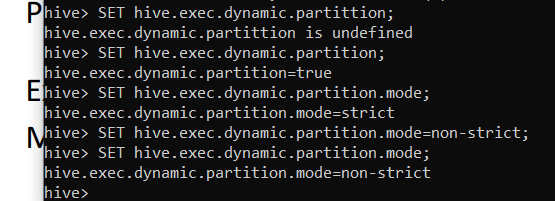
5.

Answer:

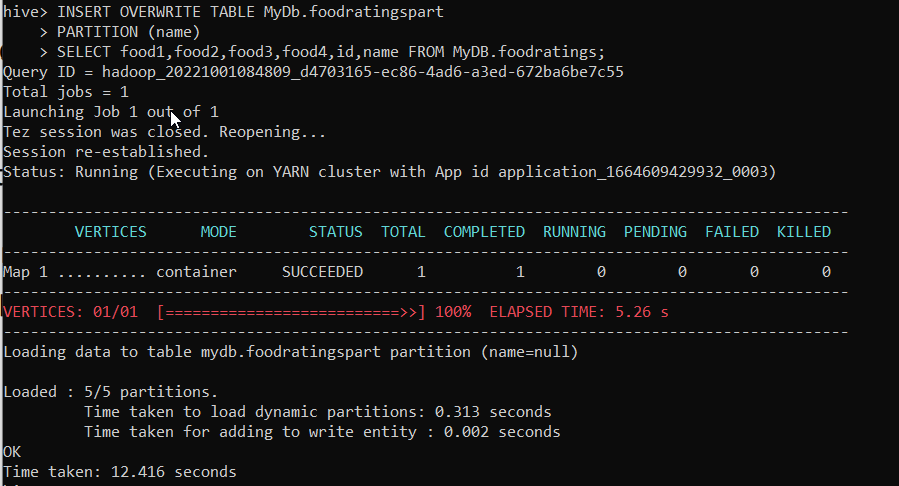
As number of critics are very low, partition on critic names would help to easy query part of data than partitioning data on number of places which are large.

6.

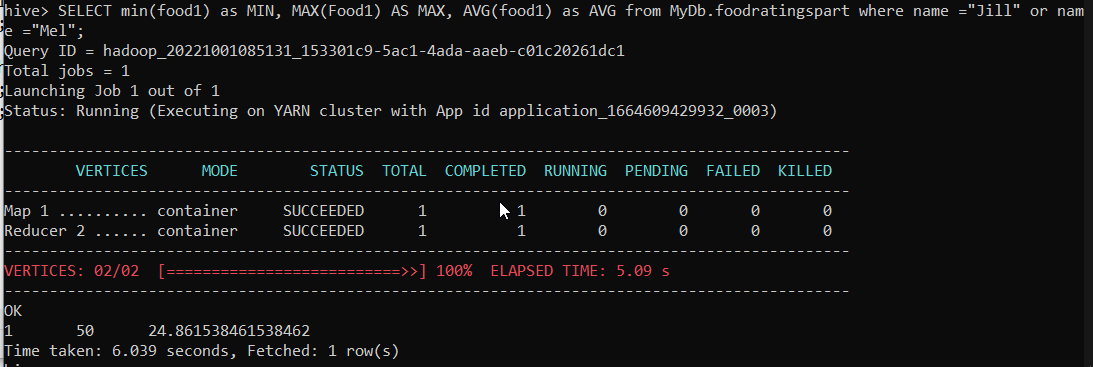
SET hive.exec.dynamic.partition.mode = non-strict;



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

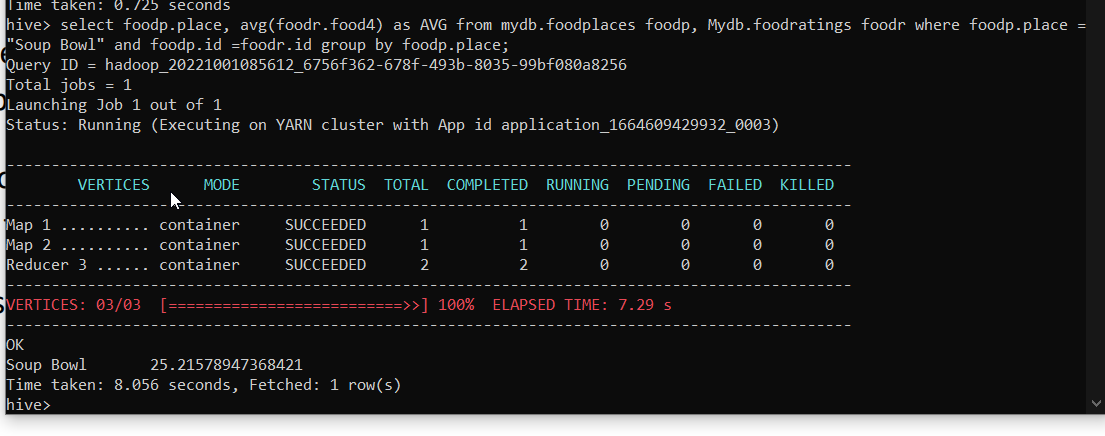


SELECT min(food1) as MIN, MAX(Food1) AS MAX, AVG(food1) as AVG from MyDb.foodratingspart where name ="Jill" or name ="Mel";



7.

select foodp.place, avg(foodr.food4) as AVG from mydb.foodplaces foodp, Mydb.foodratings foodr where foodp.place ="Soup Bowl" and foodp.id =foodr.id group by foodp.place;



8.

a. Row format is most useful when user has to access data with respect to row values and need to access many rows at a time. This format is used to read and write data optimally.

Column format is useful when the computation is focused on specific columns without the need to search row values. This format is used to read and compute optimally.

b.

The ability to breakdown a file into smaller parts which are not dependent on each other is called Splitability. Splitability on column file format can be done when the query computation is focused on one column. Which splits the data based on columns thus, making the computation optimized.

c.

Storing data of same type side by side allows us to compress better than stored in row by row.

d.

Parquet is used in Hadoop analytical database like (Implaca). It is specially used in analysing huge dataset with multiple columns for computations.