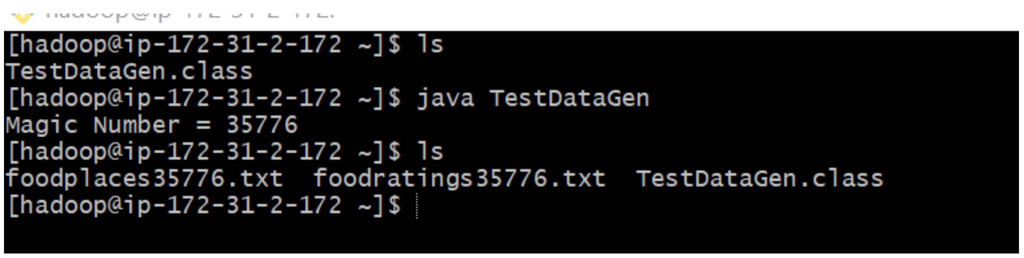
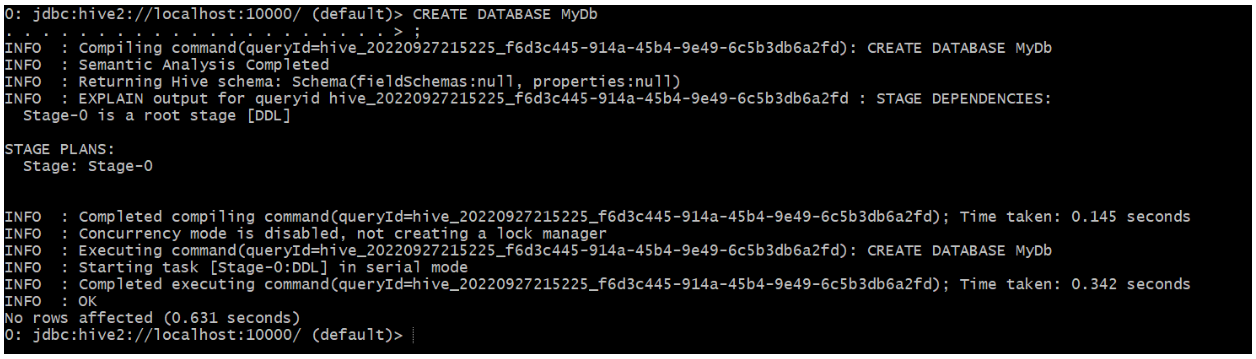
```
Magic Number = 35776
```



```
[hadoop@ip-172-31-2-172 ~]$ ls
TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$ java TestDataGen
Magic Number = 35776
[hadoop@ip-172-31-2-172 ~]$ ls
foodplaces35776.txt  foodratings35776.txt  TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$
```

Exercise 1) 2 points Create a Hive database called "MyDb"

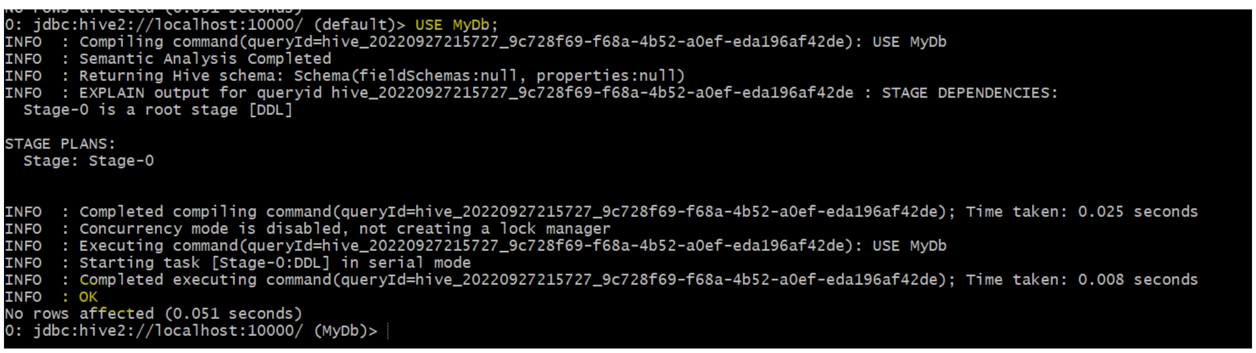
CREATE DATABASE MyDb;



```
0: jdbc:hive2://localhost:10000/ (default)> CREATE DATABASE MyDb
. . . . . . . . . . . . . . . . . . . . . .> ;
INFO  : Compiling command(queryId=hive_20220927215225_f6d3c445-914a-45b4-9e49-6c5b3db6a2fd): CREATE DATABASE MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927215225_f6d3c445-914a-45b4-9e49-6c5b3db6a2fd : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0


INFO  : Completed compiling command(queryId=hive_20220927215225_f6d3c445-914a-45b4-9e49-6c5b3db6a2fd); Time taken: 0.145 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927215225_f6d3c445-914a-45b4-9e49-6c5b3db6a2fd): CREATE DATABASE MyDb
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927215225_f6d3c445-914a-45b4-9e49-6c5b3db6a2fd); Time taken: 0.342 seconds
INFO  : OK
No rows affected (0.631 seconds)
0: jdbc:hive2://localhost:10000/ (default)>
```

To change from default to MyDb:

USE MyDb;

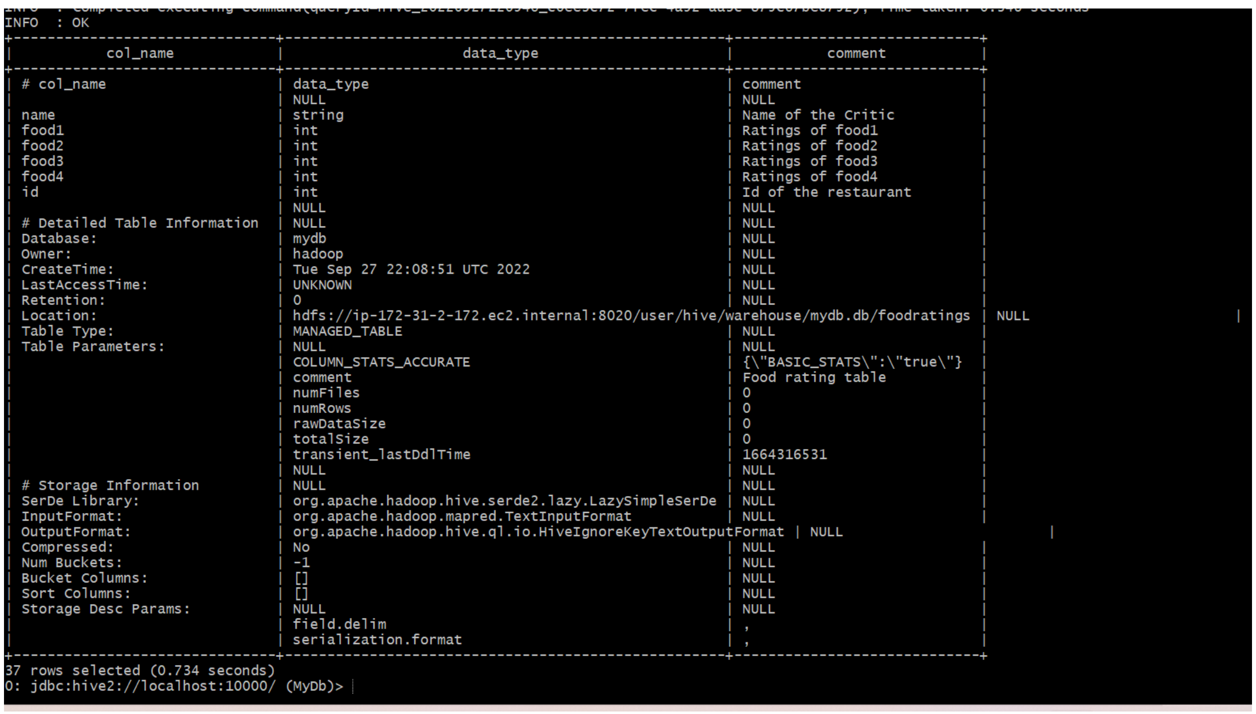


```
0: jdbc:hive2://localhost:10000/ (default)> USE MyDb;
INFO  : Compiling command(queryId=hive_20220927215727_9c728f69-f68a-4b52-a0ef-eda196af42de): USE MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927215727_9c728f69-f68a-4b52-a0ef-eda196af42de : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0


INFO  : Completed compiling command(queryId=hive_20220927215727_9c728f69-f68a-4b52-a0ef-eda196af42de); Time taken: 0.025 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927215727_9c728f69-f68a-4b52-a0ef-eda196af42de): USE MyDb
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927215727_9c728f69-f68a-4b52-a0ef-eda196af42de); Time taken: 0.008 seconds
INFO  : OK
No rows affected (0.051 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

Creating "foodratings" table:

```
CREATE TABLE IF NOT EXISTS foodratings (
    name STRING COMMENT 'Name of the Critic',
    food1 INT COMMENT 'Ratings of food1',
food2 INT COMMENT 'Ratings of food2',
food3 INT COMMENT 'Ratings of food3',
food4 INT COMMENT 'Ratings of food4',
id INT COMMENT 'Id of the restaurant ')
COMMENT 'Food rating table'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

<u>DESCRIBE FORMATTED MyDb.foodratings;</u>

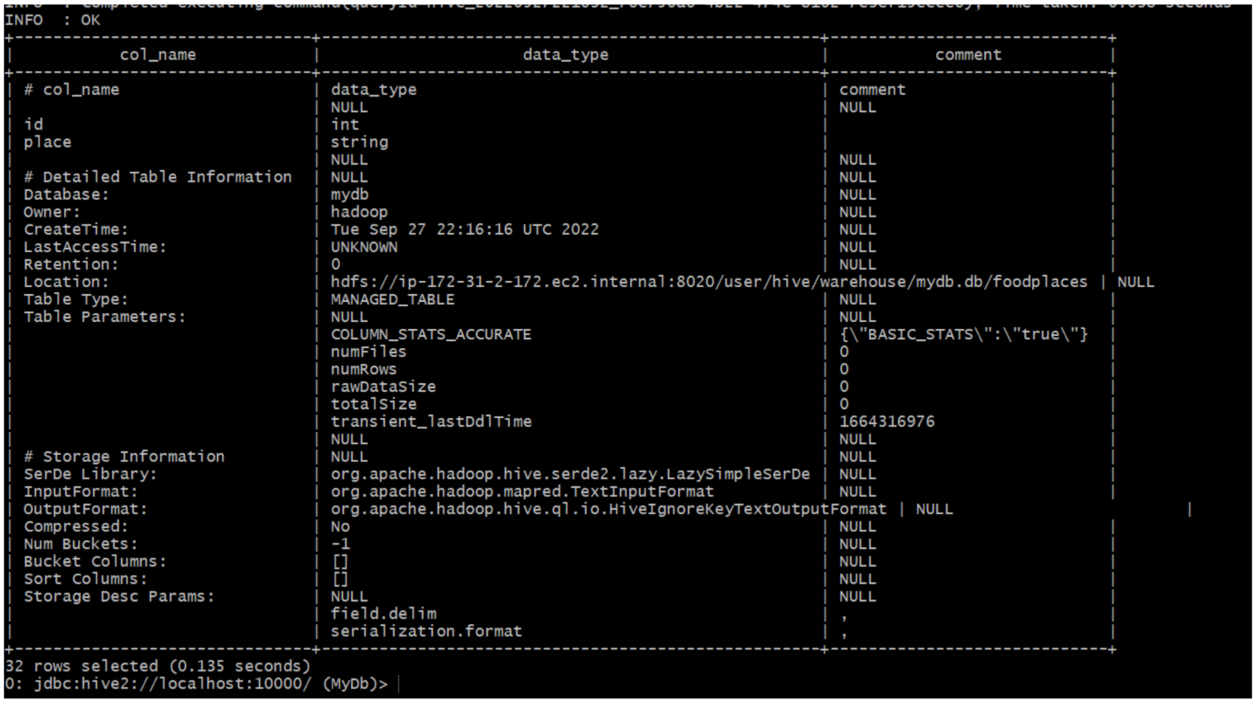


```
INFO  : OK
+-------------------------------+----------------------------------------------------+-----------------------------+
|           col_name            |                     data_type                      |           comment           |
+-------------------------------+----------------------------------------------------+-----------------------------+
| # col_name                    | data_type                                          | comment                     |
|                               | NULL                                               | NULL                        |
| name                          | string                                             | Name of the Critic          |
| food1                         | int                                                | Ratings of food1            |
| food2                         | int                                                | Ratings of food2            |
| food3                         | int                                                | Ratings of food3            |
| food4                         | int                                                | Ratings of food4            |
| id                            | int                                                | Id of the restaurant        |
|                               | NULL                                               | NULL                        |
| # Detailed Table Information  | NULL                                               | NULL                        |
| Database:                     | mydb                                               | NULL                        |
| Owner:                        | hadoop                                             | NULL                        |
| CreateTime:                   | Tue Sep 27 22:08:51 UTC 2022                       | NULL                        |
| LastAccessTime:               | UNKNOWN                                            | NULL                        |
| Retention:                    | 0                                                  | NULL                        |
| Location:                     | hdfs://ip-172-31-2-172.ec2.internal:8020/user/hive/warehouse/mydb.db/foodratings | NULL                        |
| Table Type:                   | MANAGED_TABLE                                      | NULL                        |
| Table Parameters:             | NULL                                               | NULL                        |
|                               | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"}  |
|                               | comment                                            | Food rating table           |
|                               | numFiles                                           | 0                           |
|                               | numRows                                            | 0                           |
|                               | rawDataSize                                        | 0                           |
|                               | totalSize                                          | 0                           |
|                               | transient_lastDdlTime                              | 1664316531                  |
|                               | NULL                                               | NULL                        |
| # Storage Information         | NULL                                               | NULL                        |
| SerDe Library:                | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL                        |
| InputFormat:                  | org.apache.hadoop.mapred.TextInputFormat           | NULL                        |
| OutputFormat:                 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat | NULL                        |
| Compressed:                   | No                                                 | NULL                        |
| Num Buckets:                  | -1                                                 | NULL                        |
| Bucket Columns:               | []                                                 | NULL                        |
| Sort Columns:                 | []                                                 | NULL                        |
| Storage Desc Params:          | NULL                                               | NULL                        |
|                               | field.delim                                        | ,                           |
|                               | serialization.format                               | ,                           |
+-------------------------------+----------------------------------------------------+-----------------------------+
37 rows selected (0.734 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

<u>Creating “foodplaces” table:</u>

```
CREATE TABLE IF NOT EXISTS foodplaces (
        id INT,
        place String
        )
        ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
        STORED AS TEXTFILE;
```

DESCRIBE FORMATTED MyDb.foodplaces;



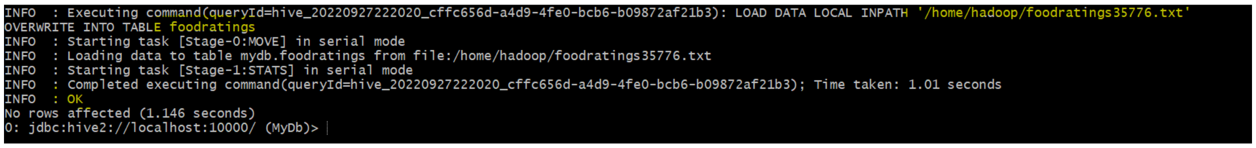
```
INFO  : OK
+-------------------------------+----------------------------------------------------+-----------------------------+
|           col_name            |                     data_type                      |           comment           |
+-------------------------------+----------------------------------------------------+-----------------------------+
| # col_name                    | data_type                                          | comment                     |
|                               | NULL                                               | NULL                        |
| id                            | int                                                |                             |
| place                         | string                                             |                             |
|                               | NULL                                               | NULL                        |
| # Detailed Table Information  | NULL                                               | NULL                        |
| Database:                     | mydb                                               | NULL                        |
| Owner:                        | hadoop                                             | NULL                        |
| CreateTime:                   | Tue Sep 27 22:16:16 UTC 2022                       | NULL                        |
| LastAccessTime:               | UNKNOWN                                            | NULL                        |
| Retention:                    | 0                                                  | NULL                        |
| Location:                     | hdfs://ip-172-31-2-172.ec2.internal:8020/user/hive/warehouse/mydb.db/foodplaces | NULL |
| Table Type:                   | MANAGED_TABLE                                      | NULL                        |
| Table Parameters:             | NULL                                               | NULL                        |
|                               | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"}  |
|                               | numFiles                                           | 0                           |
|                               | numRows                                            | 0                           |
|                               | rawDataSize                                        | 0                           |
|                               | totalSize                                          | 0                           |
|                               | transient_lastDdlTime                              | 1664316976                  |
|                               | NULL                                               | NULL                        |
| # Storage Information         | NULL                                               | NULL                        |
| SerDe Library:                | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL                        |
| InputFormat:                  | org.apache.hadoop.mapred.TextInputFormat           | NULL                        |
| OutputFormat:                 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat | NULL                |
| Compressed:                   | No                                                 | NULL                        |
| Num Buckets:                  | -1                                                 | NULL                        |
| Bucket Columns:               | []                                                 | NULL                        |
| Sort Columns:                 | []                                                 | NULL                        |
| Storage Desc Params:          | NULL                                               | NULL                        |
|                               | field.delim                                        | ,                           |
|                               | serialization.format                               | ,                           |
+-------------------------------+----------------------------------------------------+-----------------------------+
32 rows selected (0.135 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

Exercise 2) 2 points

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

LOAD DATA LOCAL INPATH '/home/hadoop/foodratings35776.txt'

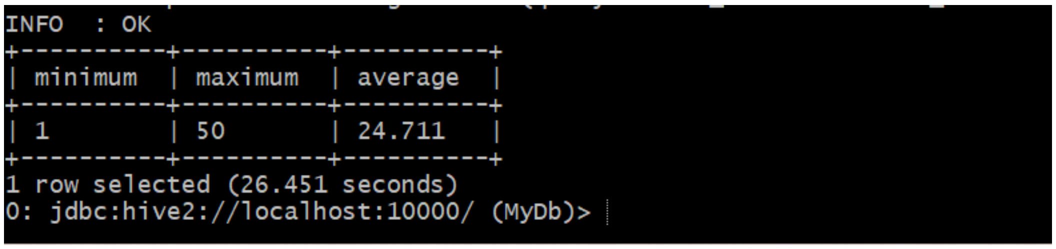
OVERWRITE INTO TABLE foodratings;

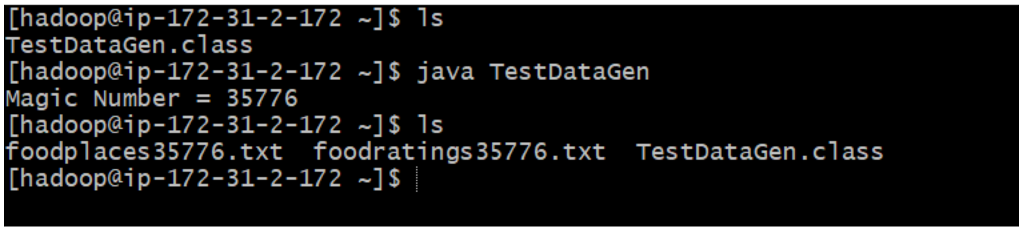


```
INFO  : Executing command(queryId=hive_20220927222020_cffc656d-a4d9-4fe0-bcb6-b09872af21b3): LOAD DATA LOCAL INPATH '/home/hadoop/foodratings35776.txt'
OVERWRITE INTO TABLE foodratings
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table mydb.foodratings from file:/home/hadoop/foodratings35776.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20220927222020_cffc656d-a4d9-4fe0-bcb6-b09872af21b3); Time taken: 1.01 seconds
INFO  : OK
No rows affected (1.146 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

Execute a hive command to output the min, max and average of the values of the food3 column of the foodratings table. This should be one hive command, not three separate ones.

Select min(food3) as minimum, max(food3) as maximum, avg(food3) as average from foodratings;


```
INFO  : OK
+----------+----------+----------+
| minimum  | maximum  | average  |
+----------+----------+----------+
| 1        | 50       | 24.711   |
+----------+----------+----------+
1 row selected (26.451 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

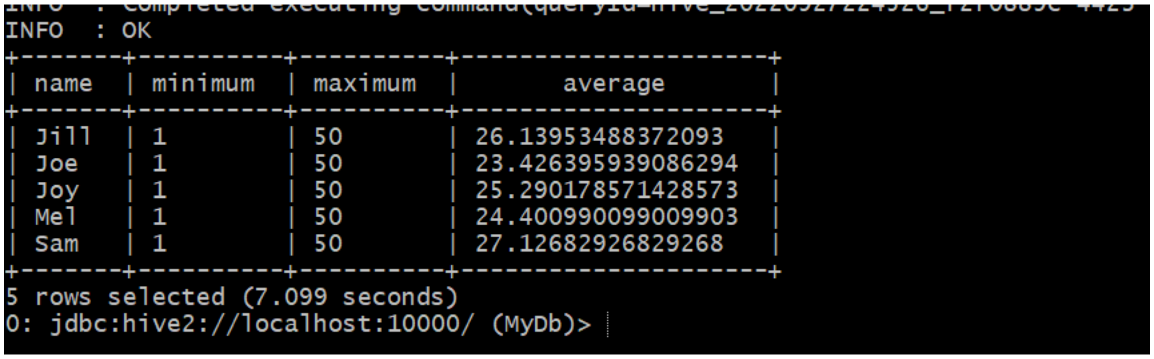

```
[hadoop@ip-172-31-2-172 ~]$ ls
TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$ java TestDataGen
Magic Number = 35776
[hadoop@ip-172-31-2-172 ~]$ ls
foodplaces35776.txt  foodratings35776.txt  TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$
```

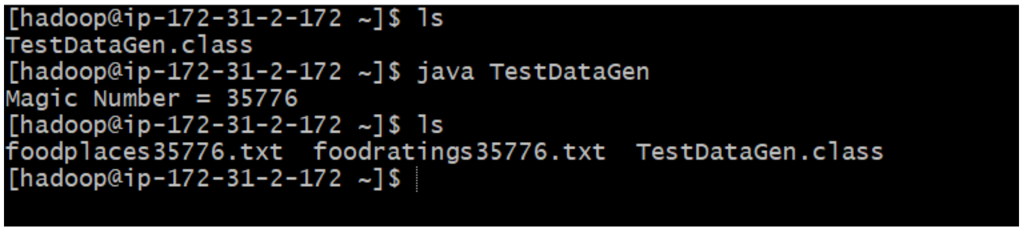
```
Magic Number = 35776
```

Exercise 3) 2 points

Execute a hive command to output the min, max and average of the values of the food1 column grouped by the first column 'name'. This should be one hive command, not three separate ones.

Select name, min(food1) as minimum, max(food1) as maximum, avg(food1) as average from foodratings group by name;


```
INFO  : OK
+-------+----------+----------+---------------------+
| name  | minimum  | maximum  |       average       |
+-------+----------+----------+---------------------+
| Jill  | 1        | 50       | 26.13953488372093   |
| Joe   | 1        | 50       | 23.426395939086294  |
| Joy   | 1        | 50       | 25.290178571428573  |
| Mel   | 1        | 50       | 24.400990099009903  |
| Sam   | 1        | 50       | 27.12682926829268   |
+-------+----------+----------+---------------------+
5 rows selected (7.099 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```


```
[hadoop@ip-172-31-2-172 ~]$ ls
TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$ java TestDataGen
Magic Number = 35776
[hadoop@ip-172-31-2-172 ~]$ ls
foodplaces35776.txt  foodratings35776.txt  TestDataGen.class
[hadoop@ip-172-31-2-172 ~]$
```

Exercise 4) 2 points

MyDb create a partitioned table called 'foodratingspart':

CREATE TABLE IF NOT EXISTS foodratingspart (

food1 INT,

food2 INT,

food3 INT,

food4 INT,

id INT

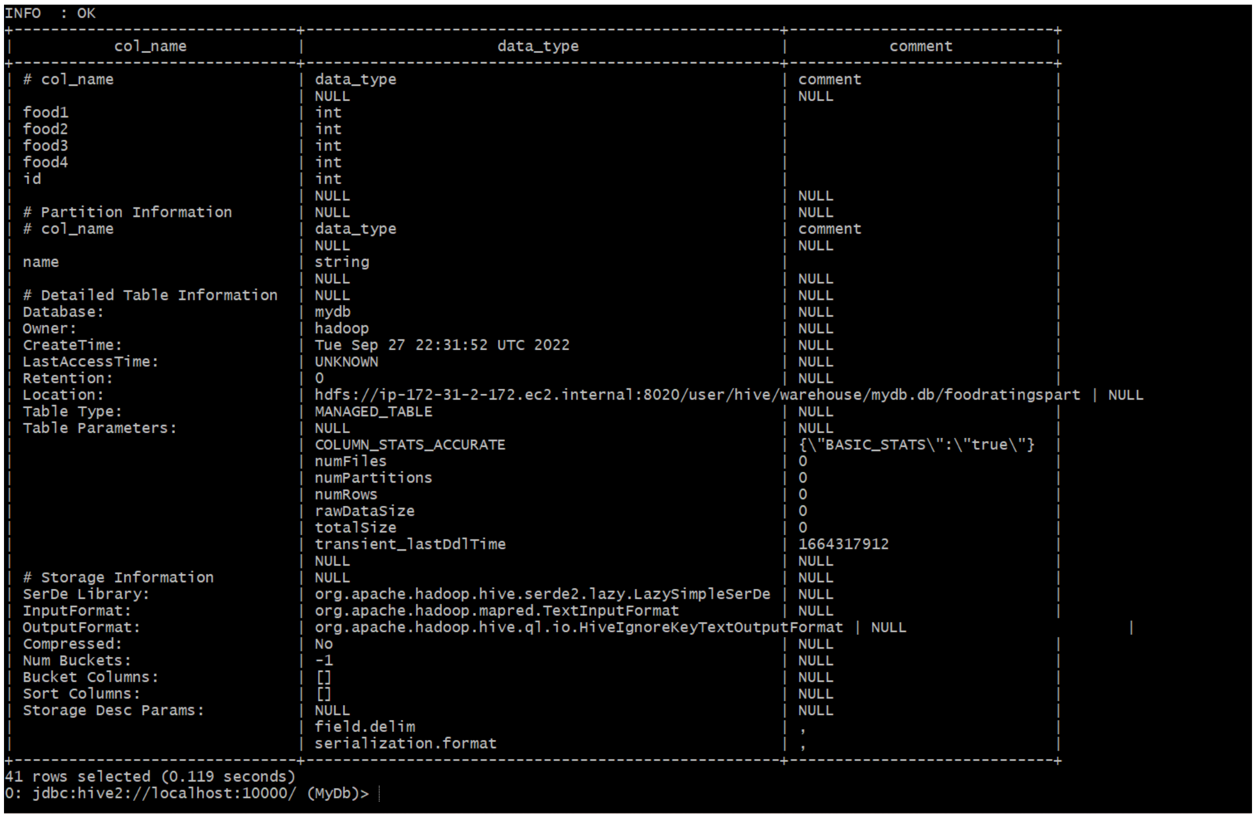
)

PARTITIONED BY (name STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

DESCRIBE FORMATTED MyDb.foodratingspart;



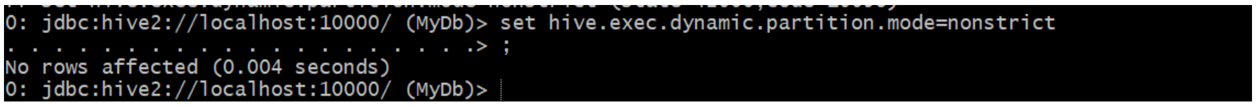
```
INFO  : OK
+-------------------------------+----------------------------------------------------+------------------------------+
|           col_name            |                     data_type                      |           comment            |
+-------------------------------+----------------------------------------------------+------------------------------+
| # col_name                    | data_type                                          | comment                      |
|                               | NULL                                               | NULL                         |
| food1                         | int                                                |                              |
| food2                         | int                                                |                              |
| food3                         | int                                                |                              |
| food4                         | int                                                |                              |
| id                            | int                                                |                              |
|                               | NULL                                               | NULL                         |
| # Partition Information       | NULL                                               | NULL                         |
| # col_name                    | data_type                                          | comment                      |
|                               | NULL                                               | NULL                         |
| name                          | string                                             |                              |
|                               | NULL                                               | NULL                         |
| # Detailed Table Information  | NULL                                               | NULL                         |
| Database:                     | mydb                                               | NULL                         |
| Owner:                        | hadoop                                             | NULL                         |
| CreateTime:                   | Tue Sep 27 22:31:52 UTC 2022                       | NULL                         |
| LastAccessTime:               | UNKNOWN                                            | NULL                         |
| Retention:                    | 0                                                  | NULL                         |
| Location:                     | hdfs://ip-172-31-2-172.ec2.internal:8020/user/hive/warehouse/mydb.db/foodratingspart | NULL |
| Table Type:                   | MANAGED_TABLE                                      | NULL                         |
| Table Parameters:             | NULL                                               | NULL                         |
|                               | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"}   |
|                               | numFiles                                           | 0                            |
|                               | numPartitions                                      | 0                            |
|                               | numRows                                            | 0                            |
|                               | rawDataSize                                        | 0                            |
|                               | totalSize                                          | 0                            |
|                               | transient_lastDdlTime                              | 1664317912                   |
|                               | NULL                                               | NULL                         |
| # Storage Information         | NULL                                               | NULL                         |
| SerDe Library:                | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL                         |
| InputFormat:                  | org.apache.hadoop.mapred.TextInputFormat           | NULL                         |
| OutputFormat:                 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat | NULL                 |
| Compressed:                   | No                                                 | NULL                         |
| Num Buckets:                  | -1                                                 | NULL                         |
| Bucket Columns:               | []                                                 | NULL                         |
| Sort Columns:                 | []                                                 | NULL                         |
| Storage Desc Params:          | NULL                                               | NULL                         |
|                               | field.delim                                        | ,                            |
|                               | serialization.format                               | ,                            |
+-------------------------------+----------------------------------------------------+------------------------------+
41 rows selected (0.119 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

Exercise 5) 2 points

<u>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.</u>

When we build a partition table, the partition column determines how many rows sets are saved in the partitioned table. In our situation, utilizing critic name will result in fewer than 10 partitions being formed in the partitioned dataset, however using place id will result in more than 10,000 groups (data sets) being present in the partitioned table, which will again affect the performance of the table.

<u>Exercise 6) 2 points</u>



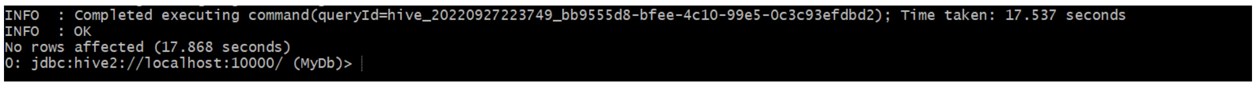
```
0: jdbc:hive2://localhost:10000/ (MyDb)> set hive.exec.dynamic.partition.mode=nonstrict
. . . . . . . . . . . . . . . . . . . . .> ;
No rows affected (0.004 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

INSERT OVERWRITE TABLE foodratingspart

PARTITION (name)

SELECT food1, food2, food3, food4, id, name

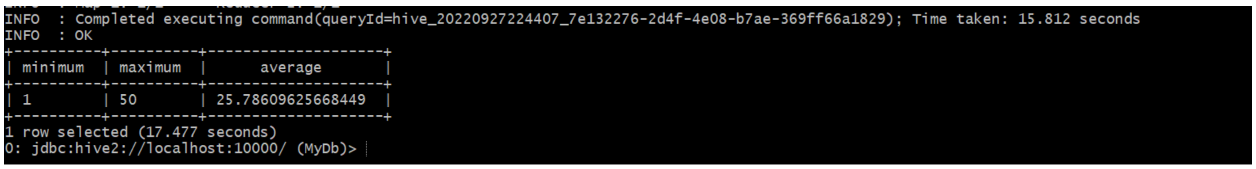
FROM foodratings;



```
INFO  : Completed executing command(queryId=hive_20220927223749_bb9555d8-bfee-4c10-99e5-0c3c93efdbd2); Time taken: 17.537 seconds
INFO  : OK
No rows affected (17.868 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

<u>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</u>

Select min(food2) as minimum, max(food2) as maximum, avg(food2) as average from foodratingspart where name = 'Mel' OR name ='Jill' ;



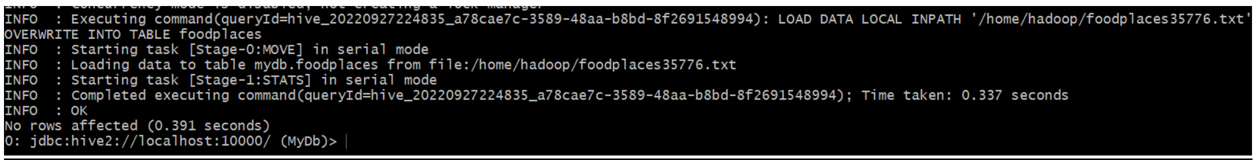
```
INFO  : Completed executing command(queryId=hive_20220927224407_7e132276-2d4f-4e08-b7ae-369ff66a1829); Time taken: 15.812 seconds
INFO  : OK
+----------+----------+--------------------+
| minimum  | maximum  |      average       |
+----------+----------+--------------------+
| 1        | 50       | 25.78609625668449  |
+----------+----------+--------------------+
1 row selected (17.477 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

<u>Exercise 7) 2 points</u>

<u>Load the foodplaces.txt file created using TestDataGen from your local file system into the foodplaces table</u>

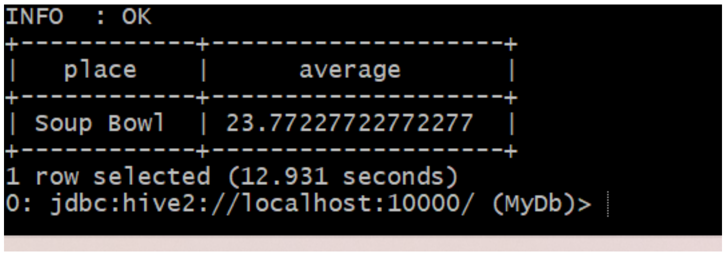
LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces35776.txt'

OVERWRITE INTO TABLE foodplaces;



```
INFO  : Executing command(queryId=hive_20220927224835_a78cae7c-3589-48aa-b8bd-8f2691548994): LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces35776.txt'
OVERWRITE INTO TABLE foodplaces
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table mydb.foodplaces from file:/home/hadoop/foodplaces35776.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20220927224835_a78cae7c-3589-48aa-b8bd-8f2691548994); Time taken: 0.337 seconds
INFO  : OK
No rows affected (0.391 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

Select fp.place as place, avg(food4) as average from foodratings fr join foodplaces fp on fr.id = fp.id where fp.place = 'Soup Bowl' group by fp.place;



```
INFO  : OK
+------------+--------------------+
|   place    |      average       |
+------------+--------------------+
| Soup Bowl  | 23.77227722772277  |
+------------+--------------------+
1 row selected (12.931 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)>
```

<u>Exercise 8) 4 points</u>

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?**

   Depending on our goals, I suppose. For instance, a row-based style is appropriate if we need to access all or most of each row's data as well as several rows. Column-based formats are appropriate if certain operations must only be performed on a certain subset of columns. For instance, all we need is the values from the Salary column to be processed in order to determine the average salary of all the employees. However, row-based storage will be the appropriate format if we need to obtain all personal information about an employee or employees.

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

   Splittability refers to the ability to divide or separate the column files into numerous small logical files for processing records concurrently. In order to process data in parallel, which is essential to boosting the processing speed of the data, it is crucial to arrange the data in this fashion.

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

   Values from one column will be kept next to one another in a column format. Then the compression rate will be better because everything will be of the same datatype. Comparatively speaking, compression on row-based storage will be less effective because data stored next to each other in row format will be of different datatypes.

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

When we need to study large datasets with several columns, it is the ideal option. In this case, each Parquet data file will include binary data organized into row groups, with the values of the columns stored adjacent to one another for each row group. Compression is simple as a result. Heavy data loads can be read using Parquet.