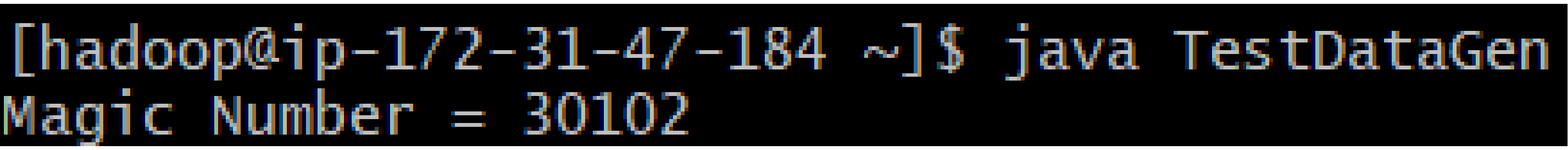
CSP554—Big Data Technologies

Assignment #4

Worth: 18 points



```
[hadoop@ip-172-31-47-184 ~]$ java TestDataGen
Magic Number = 30102
```

Exercise 1) 2 points

Create a Hive database called “MyDb”.

Note, after you do this the default database is still 'default." So unless you do something specific about this, if you create a table without qualifying it as belonging to MyDb (MyDb.sometable), it is created in the 'default' database. You can change the default database via a hive command. Try to discover which one and execute it now. Or when you create and use a table you must always qualify its name with the name of the database you created.

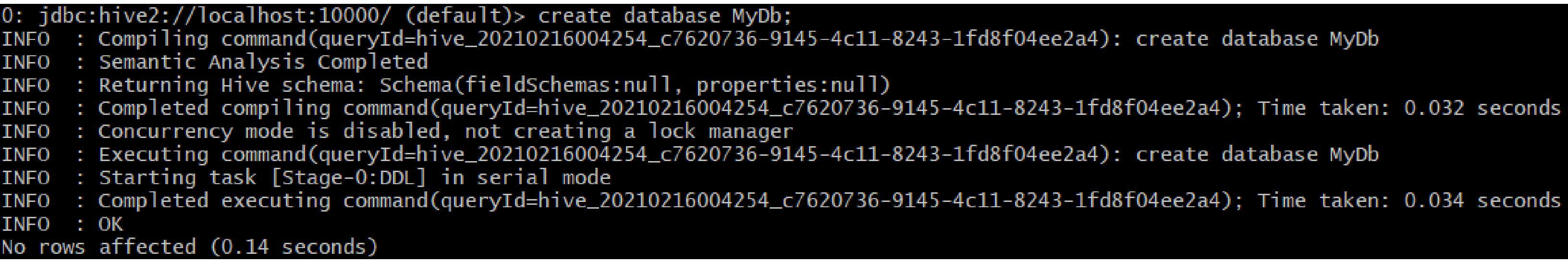
Now in MyDb create a table with name foodratings having six columns with the name of the first ‘name’ and the type of the first a string and the names of the remaining columns food1, food2, food3, food4 and id and indicate their types each as an integer. The table should have storage format TEXTFILE and column separator a “,”. That is the underlying format should be a CSV file. The table itself and each column should include a comment just to show me you know how to use comments (it does not matter what it says).

Execute a Hive command of ‘DESCRIBE FORMATTED MyDb.foodratings;’ and capture its output as one of the results of this exercise.

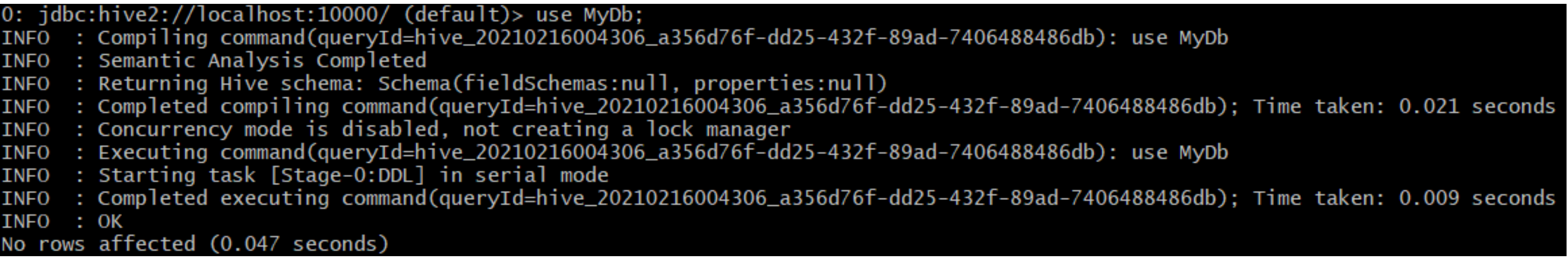
Then in MyDb create a table with name foodplaces having two columns with first called ‘id’ with the type of the first an integer, and the second column called ‘place’ with the type of the second a string. This table should also have storage format TEXTFILE and column separator a “,”. That is the underlying format should be a CSV file. No comments are needed for this table.

Execute a Hive command of ‘DESCRIBE FORMATTED MyDb.foodplaces’ and capture its output as another of the results of this exercise.

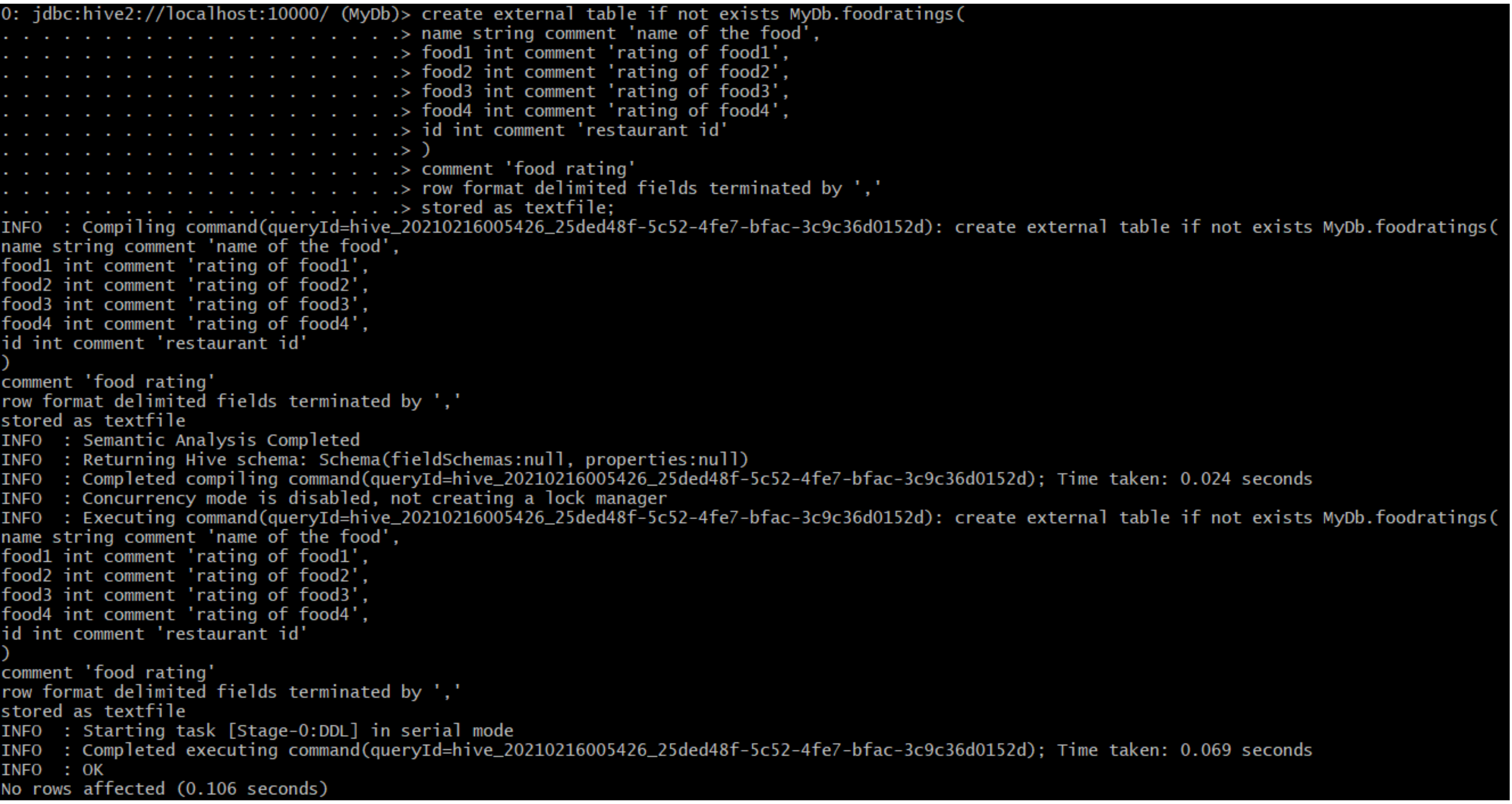
Magic Number: 30102



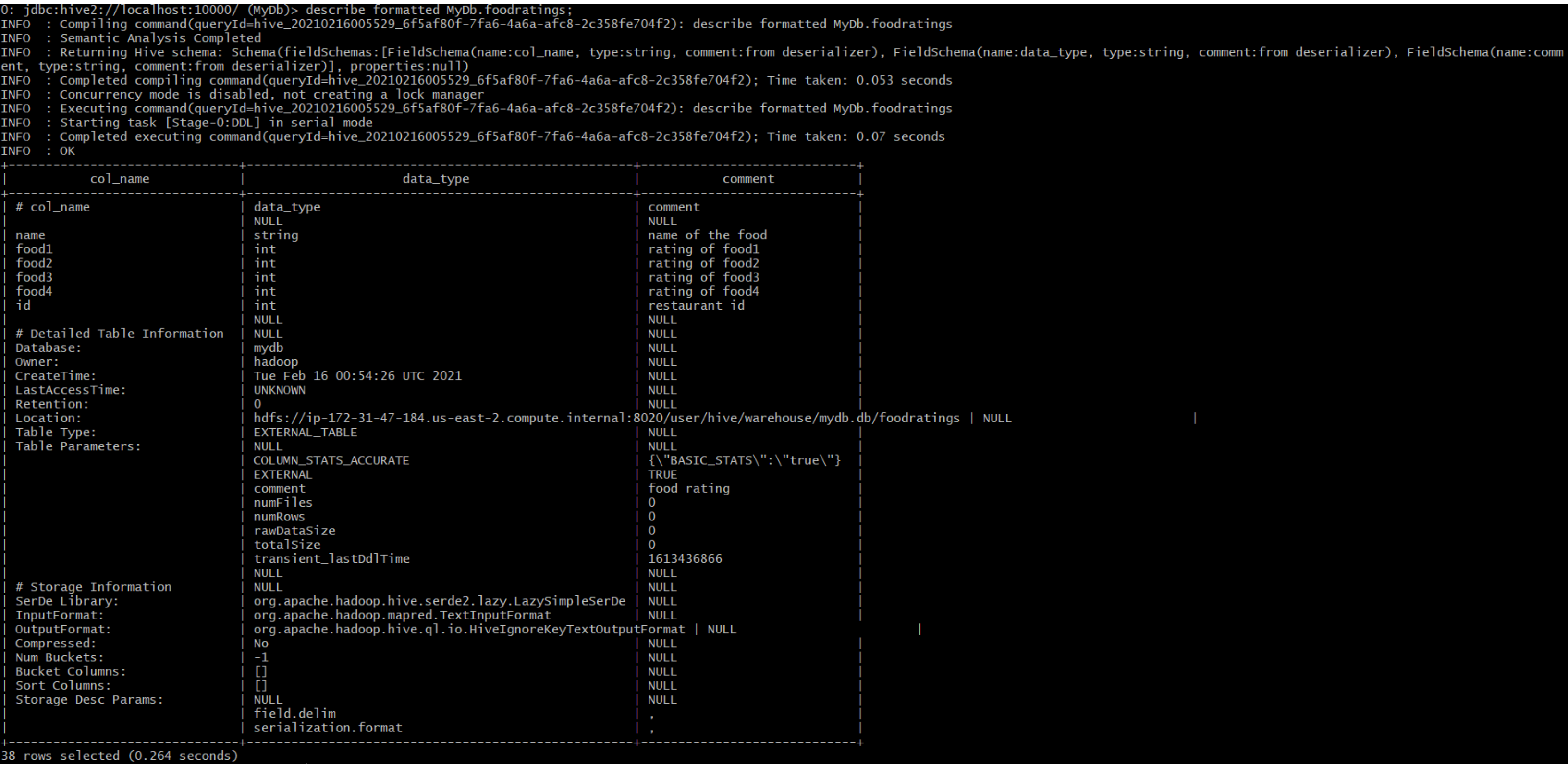
```
0: jdbc:hive2://localhost:10000/ (default)> create database MyDb;
INFO  : Compiling command(queryId=hive_20210216004254_c7620736-9145-4c11-8243-1fd8f04ee2a4): create database MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216004254_c7620736-9145-4c11-8243-1fd8f04ee2a4); Time taken: 0.032 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216004254_c7620736-9145-4c11-8243-1fd8f04ee2a4): create database MyDb
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216004254_c7620736-9145-4c11-8243-1fd8f04ee2a4); Time taken: 0.034 seconds
INFO  : OK
No rows affected (0.14 seconds)
```



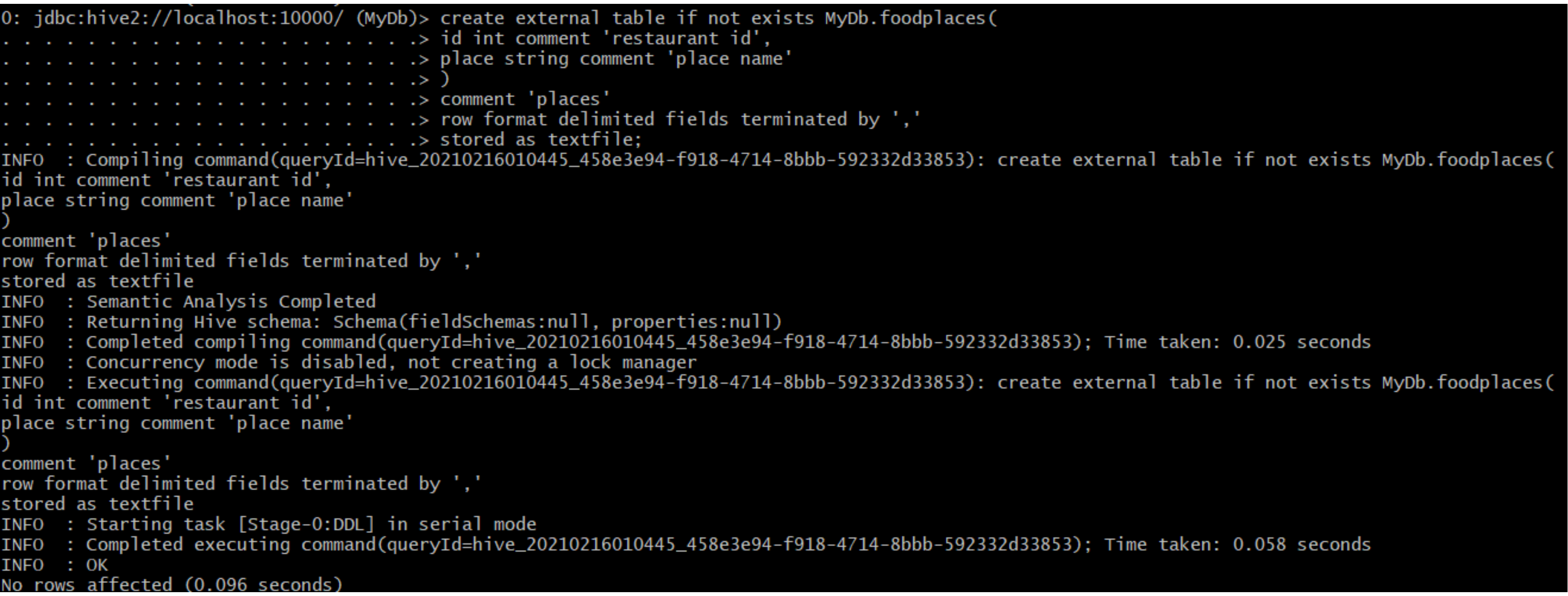
```
0: jdbc:hive2://localhost:10000/ (default)> use MyDb;
INFO  : Compiling command(queryId=hive_20210216004306_a356d76f-dd25-432f-89ad-7406488486db): use MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216004306_a356d76f-dd25-432f-89ad-7406488486db); Time taken: 0.021 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216004306_a356d76f-dd25-432f-89ad-7406488486db): use MyDb
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216004306_a356d76f-dd25-432f-89ad-7406488486db); Time taken: 0.009 seconds
INFO  : OK
No rows affected (0.047 seconds)
```



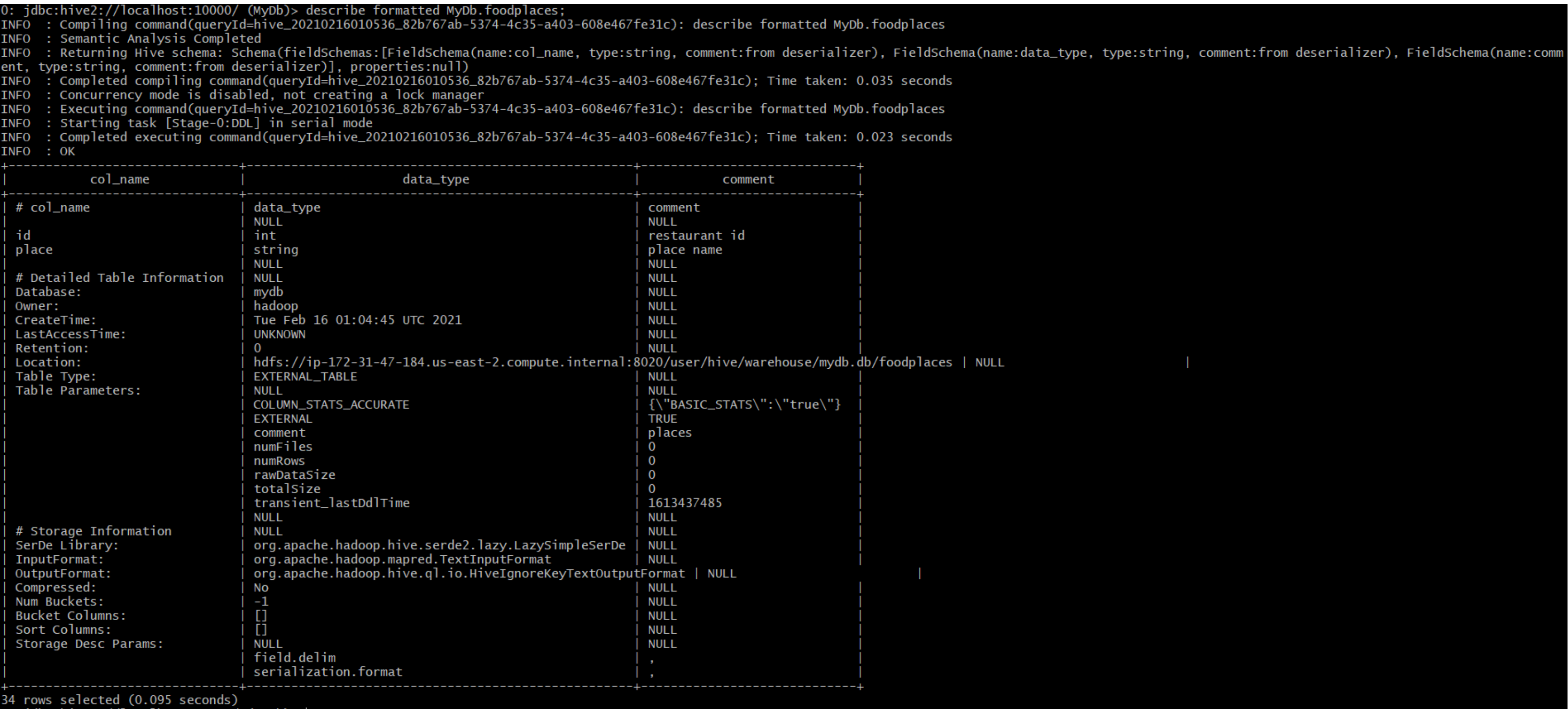
```
0: jdbc:hive2://localhost:10000/ (MyDb)> create external table if not exists MyDb.foodratings(
. . . . . . . . . . . . . . . . . . . .> name string comment 'name of the food',
. . . . . . . . . . . . . . . . . . . .> food1 int comment 'rating of food1',
. . . . . . . . . . . . . . . . . . . .> food2 int comment 'rating of food2',
. . . . . . . . . . . . . . . . . . . .> food3 int comment 'rating of food3',
. . . . . . . . . . . . . . . . . . . .> food4 int comment 'rating of food4',
. . . . . . . . . . . . . . . . . . . .> id int comment 'restaurant id'
. . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . .> comment 'food rating'
. . . . . . . . . . . . . . . . . . . .> row format delimited fields terminated by ','
. . . . . . . . . . . . . . . . . . . .> stored as textfile;
INFO  : Compiling command(queryId=hive_20210216005426_25ded48f-5c52-4fe7-bfac-3c9c36d0152d): create external table if not exists MyDb.foodratings(
name string comment 'name of the food',
food1 int comment 'rating of food1',
food2 int comment 'rating of food2',
food3 int comment 'rating of food3',
food4 int comment 'rating of food4',
id int comment 'restaurant id'
)
comment 'food rating'
row format delimited fields terminated by ','
stored as textfile
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216005426_25ded48f-5c52-4fe7-bfac-3c9c36d0152d); Time taken: 0.024 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216005426_25ded48f-5c52-4fe7-bfac-3c9c36d0152d): create external table if not exists MyDb.foodratings(
name string comment 'name of the food',
food1 int comment 'rating of food1',
food2 int comment 'rating of food2',
food3 int comment 'rating of food3',
food4 int comment 'rating of food4',
id int comment 'restaurant id'
)
comment 'food rating'
row format delimited fields terminated by ','
stored as textfile
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216005426_25ded48f-5c52-4fe7-bfac-3c9c36d0152d); Time taken: 0.069 seconds
INFO  : OK
No rows affected (0.106 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> describe formatted MyDb.foodratings;
INFO  : Compiling command(queryId=hive_20210216005529_6f5af80f-7fa6-4a6a-afc8-2c358fe704f2): describe formatted MyDb.foodratings
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comm
ent, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216005529_6f5af80f-7fa6-4a6a-afc8-2c358fe704f2); Time taken: 0.053 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216005529_6f5af80f-7fa6-4a6a-afc8-2c358fe704f2): describe formatted MyDb.foodratings
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216005529_6f5af80f-7fa6-4a6a-afc8-2c358fe704f2); Time taken: 0.07 seconds
INFO  : OK
+-------------------------------+----------------------------------------------------+----------------------------+
|           col_name            |                     data_type                      |          comment           |
+-------------------------------+----------------------------------------------------+----------------------------+
| # col_name                    | data_type                                          | comment                    |
|                               | NULL                                               | NULL                       |
| name                          | string                                             | name of the food           |
| food1                         | int                                                | rating of food1            |
| food2                         | int                                                | rating of food2            |
| food3                         | int                                                | rating of food3            |
| food4                         | int                                                | rating of food4            |
| id                            | int                                                | restaurant id              |
|                               | NULL                                               | NULL                       |
| # Detailed Table Information  | NULL                                               | NULL                       |
| Database:                     | mydb                                               | NULL                       |
| Owner:                        | hadoop                                             | NULL                       |
| CreateTime:                   | Tue Feb 16 00:54:26 UTC 2021                       | NULL                       |
| LastAccessTime:               | UNKNOWN                                            | NULL                       |
| Retention:                    | 0                                                  | NULL                       |
| Location:                     | hdfs://ip-172-31-47-184.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodratings | NULL                       |
| Table Type:                   | EXTERNAL_TABLE                                     | NULL                       |
| Table Parameters:             | NULL                                               | NULL                       |
|                               | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"} |
|                               | EXTERNAL                                           | TRUE                       |
|                               | comment                                            | food rating                |
|                               | numFiles                                           | 0                          |
|                               | numRows                                            | 0                          |
|                               | rawDataSize                                        | 0                          |
|                               | totalSize                                          | 0                          |
|                               | transient_lastDdlTime                              | 1613436866                 |
|                               | 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                               | ,                          |
+-------------------------------+----------------------------------------------------+----------------------------+
38 rows selected (0.264 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> create external table if not exists MyDb.foodplaces(
. . . . . . . . . . . . . . . . . . . .> id int comment 'restaurant id',
. . . . . . . . . . . . . . . . . . . .> place string comment 'place name'
. . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . .> comment 'places'
. . . . . . . . . . . . . . . . . . . .> row format delimited fields terminated by ','
. . . . . . . . . . . . . . . . . . . .> stored as textfile;
INFO  : Compiling command(queryId=hive_20210216010445_458e3e94-f918-4714-8bbb-592332d33853): create external table if not exists MyDb.foodplaces(
id int comment 'restaurant id',
place string comment 'place name'
)
comment 'places'
row format delimited fields terminated by ','
stored as textfile
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216010445_458e3e94-f918-4714-8bbb-592332d33853); Time taken: 0.025 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216010445_458e3e94-f918-4714-8bbb-592332d33853): create external table if not exists MyDb.foodplaces(
id int comment 'restaurant id',
place string comment 'place name'
)
comment 'places'
row format delimited fields terminated by ','
stored as textfile
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216010445_458e3e94-f918-4714-8bbb-592332d33853); Time taken: 0.058 seconds
INFO  : OK
No rows affected (0.096 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> describe formatted MyDb.foodplaces;
INFO  : Compiling command(queryId=hive_20210216010536_82b767ab-5374-4c35-a403-608e467fe31c): describe formatted MyDb.foodplaces
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comm
ent, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216010536_82b767ab-5374-4c35-a403-608e467fe31c); Time taken: 0.035 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216010536_82b767ab-5374-4c35-a403-608e467fe31c): describe formatted MyDb.foodplaces
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216010536_82b767ab-5374-4c35-a403-608e467fe31c); Time taken: 0.023 seconds
INFO  : OK
+------------------------------+----------------------------------------------------+---------------------------+
|           col_name           |                     data_type                      |          comment          |
+------------------------------+----------------------------------------------------+---------------------------+
| # col_name                   | data_type                                          | comment                   |
|                              | NULL                                               | NULL                      |
| id                           | int                                                | restaurant id             |
| place                        | string                                             | place name                |
|                              | NULL                                               | NULL                      |
| # Detailed Table Information | NULL                                               | NULL                      |
| Database:                    | mydb                                               | NULL                      |
| Owner:                       | hadoop                                             | NULL                      |
| CreateTime:                  | Tue Feb 16 01:04:45 UTC 2021                       | NULL                      |
| LastAccessTime:              | UNKNOWN                                            | NULL                      |
| Retention:                   | 0                                                  | NULL                      |
| Location:                    | hdfs://ip-172-31-47-184.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodplaces | NULL                      |
| Table Type:                  | EXTERNAL_TABLE                                     | NULL                      |
| Table Parameters:            | NULL                                               | NULL                      |
|                              | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"} |
|                              | EXTERNAL                                           | TRUE                      |
|                              | comment                                            | places                    |
|                              | numFiles                                           | 0                         |
|                              | numRows                                            | 0                         |
|                              | rawDataSize                                        | 0                         |
|                              | totalSize                                          | 0                         |
|                              | transient_lastDdlTime                              | 1613437485                |
|                              | 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                               | ,                         |
+------------------------------+----------------------------------------------------+---------------------------+
34 rows selected (0.095 seconds)
```

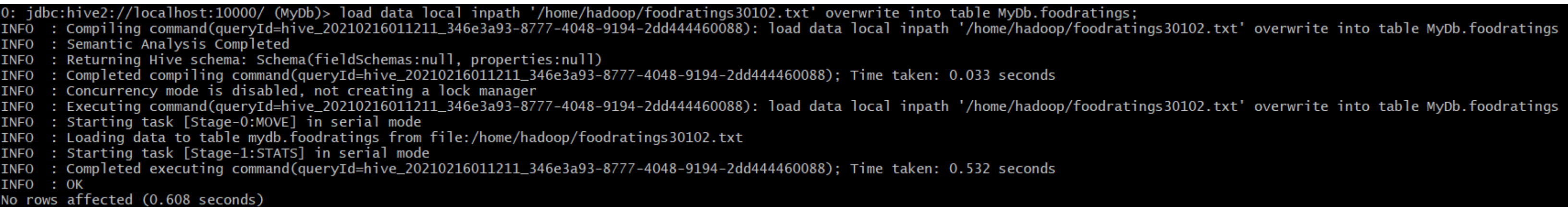
Exercise 2) 2 points

Load the foodratings<magic number>.txt file created using TestDataGen from your local file system into the foodratings table.

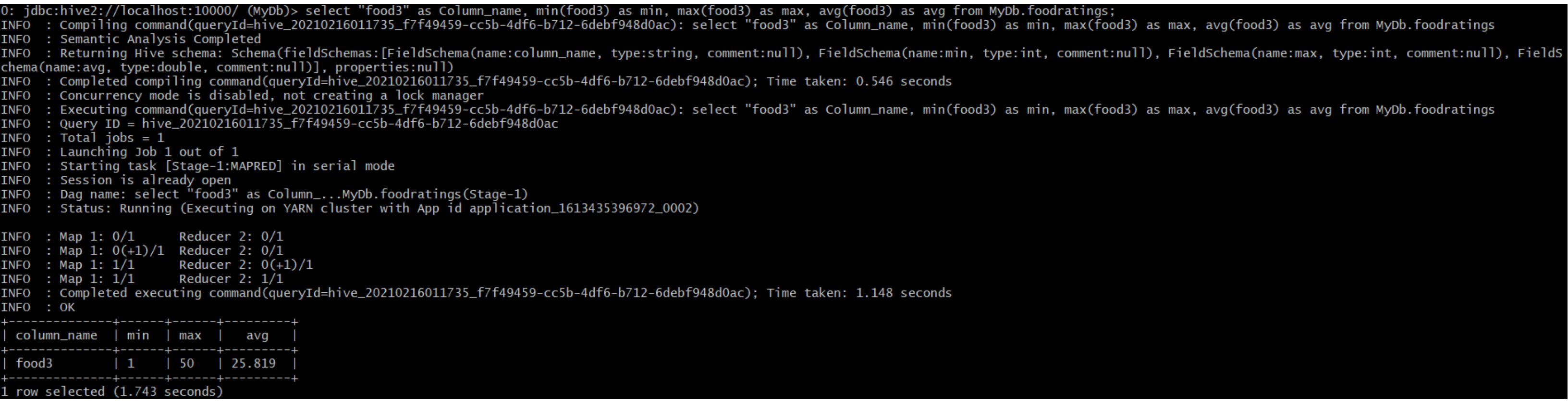
Execute a hive command to output the min, max and average of the values of the food3 column of the foodratings table. This should be one hive command, not three separate ones.

A copy of the hive command you wrote, the output of this query and the magic number are the result of this exercise.

Magic Number: 30102



```
0: jdbc:hive2://localhost:10000/ (MyDb)> load data local inpath '/home/hadoop/foodratings30102.txt' overwrite into table MyDb.foodratings;
INFO  : Compiling command(queryId=hive_20210216011211_346e3a93-8777-4048-9194-2dd444460088): load data local inpath '/home/hadoop/foodratings30102.txt' overwrite into table MyDb.foodratings
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216011211_346e3a93-8777-4048-9194-2dd444460088); Time taken: 0.033 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216011211_346e3a93-8777-4048-9194-2dd444460088): load data local inpath '/home/hadoop/foodratings30102.txt' overwrite into table MyDb.foodratings
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table mydb.foodratings from file:/home/hadoop/foodratings30102.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20210216011211_346e3a93-8777-4048-9194-2dd444460088); Time taken: 0.532 seconds
INFO  : OK
No rows affected (0.608 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> select "food3" as Column_name, min(food3) as min, max(food3) as max, avg(food3) as avg from MyDb.foodratings;
INFO  : Compiling command(queryId=hive_20210216011735_f7f49459-cc5b-4df6-b712-6debf948d0ac): select "food3" as Column_name, min(food3) as min, max(food3) as max, avg(food3) as avg from MyDb.foodratings
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:column_name, type:string, comment:null), FieldSchema(name:min, type:int, comment:null), FieldSchema(name:max, type:int, comment:null), FieldS
chema(name:avg, type:double, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216011735_f7f49459-cc5b-4df6-b712-6debf948d0ac); Time taken: 0.546 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216011735_f7f49459-cc5b-4df6-b712-6debf948d0ac): select "food3" as Column_name, min(food3) as min, max(food3) as max, avg(food3) as avg from MyDb.foodratings
INFO  : Query ID = hive_20210216011735_f7f49459-cc5b-4df6-b712-6debf948d0ac
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Session is already open
INFO  : Dag name: select "food3" as Column_...MyDb.foodratings(Stage-1)
INFO  : Status: Running (Executing on YARN cluster with App id application_1613435396972_0002)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Completed executing command(queryId=hive_20210216011735_f7f49459-cc5b-4df6-b712-6debf948d0ac); Time taken: 1.148 seconds
INFO  : OK
+--------------+------+------+---------+
| column_name  | min  | max  |   avg   |
+--------------+------+------+---------+
| food3        | 1    | 50   | 25.819  |
+--------------+------+------+---------+
1 row selected (1.743 seconds)
```

Exercise 3) 2 points

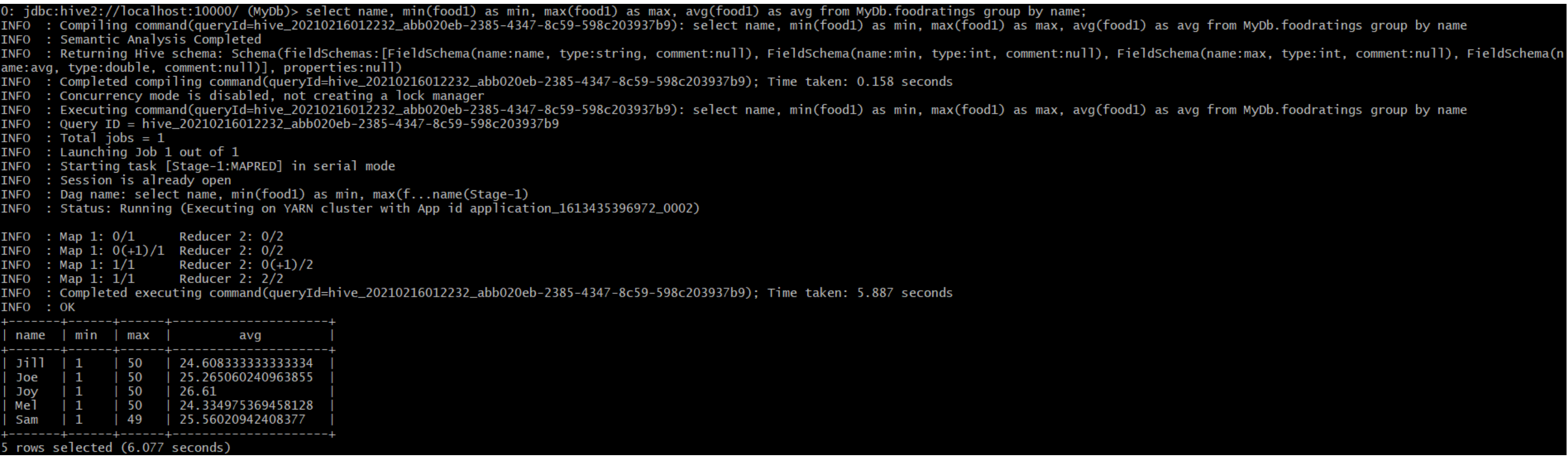
Execute a hive command to output the min, max and average of the values of the food1 column grouped by the first column ‘name’. This should be one hive command, not three separate ones.

The output should look something like:

Mel 10 20 15
Bill 20, 30, 24
…

A copy of the hive command you wrote, the output of this query and the magic number are the result of this exercise.

Magic Number: 30102



```
0: jdbc:hive2://localhost:10000/ (MyDb)> select name, min(food1) as min, max(food1) as max, avg(food1) as avg from MyDb.foodratings group by name;
INFO  : Compiling command(queryId=hive_20210216012232_abb020eb-2385-4347-8c59-598c203937b9): select name, min(food1) as min, max(food1) as max, avg(food1) as avg from MyDb.foodratings group by name
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:name, type:string, comment:null), FieldSchema(name:min, type:int, comment:null), FieldSchema(name:max, type:int, comment:null), FieldSchema(n
ame:avg, type:double, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216012232_abb020eb-2385-4347-8c59-598c203937b9); Time taken: 0.158 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216012232_abb020eb-2385-4347-8c59-598c203937b9): select name, min(food1) as min, max(food1) as max, avg(food1) as avg from MyDb.foodratings group by name
INFO  : Query ID = hive_20210216012232_abb020eb-2385-4347-8c59-598c203937b9
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Session is already open
INFO  : Dag name: select name, min(food1) as min, max(f...name(Stage-1)
INFO  : Status: Running (Executing on YARN cluster with App id application_1613435396972_0002)

INFO  : Map 1: 0/1      Reducer 2: 0/2
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/2
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/2
INFO  : Map 1: 1/1      Reducer 2: 2/2
INFO  : Completed executing command(queryId=hive_20210216012232_abb020eb-2385-4347-8c59-598c203937b9); Time taken: 5.887 seconds
INFO  : OK
+-------+------+------+---------------------+
| name  | min  | max  |         avg         |
+-------+------+------+---------------------+
| Jill  | 1    | 50   | 24.608333333333334  |
| Joe   | 1    | 50   | 25.265060240963855  |
| Joy   | 1    | 50   | 26.61               |
| Mel   | 1    | 50   | 24.334975369458128  |
| Sam   | 1    | 49   | 25.56020942408377   |
+-------+------+------+---------------------+
5 rows selected (6.077 seconds)
```

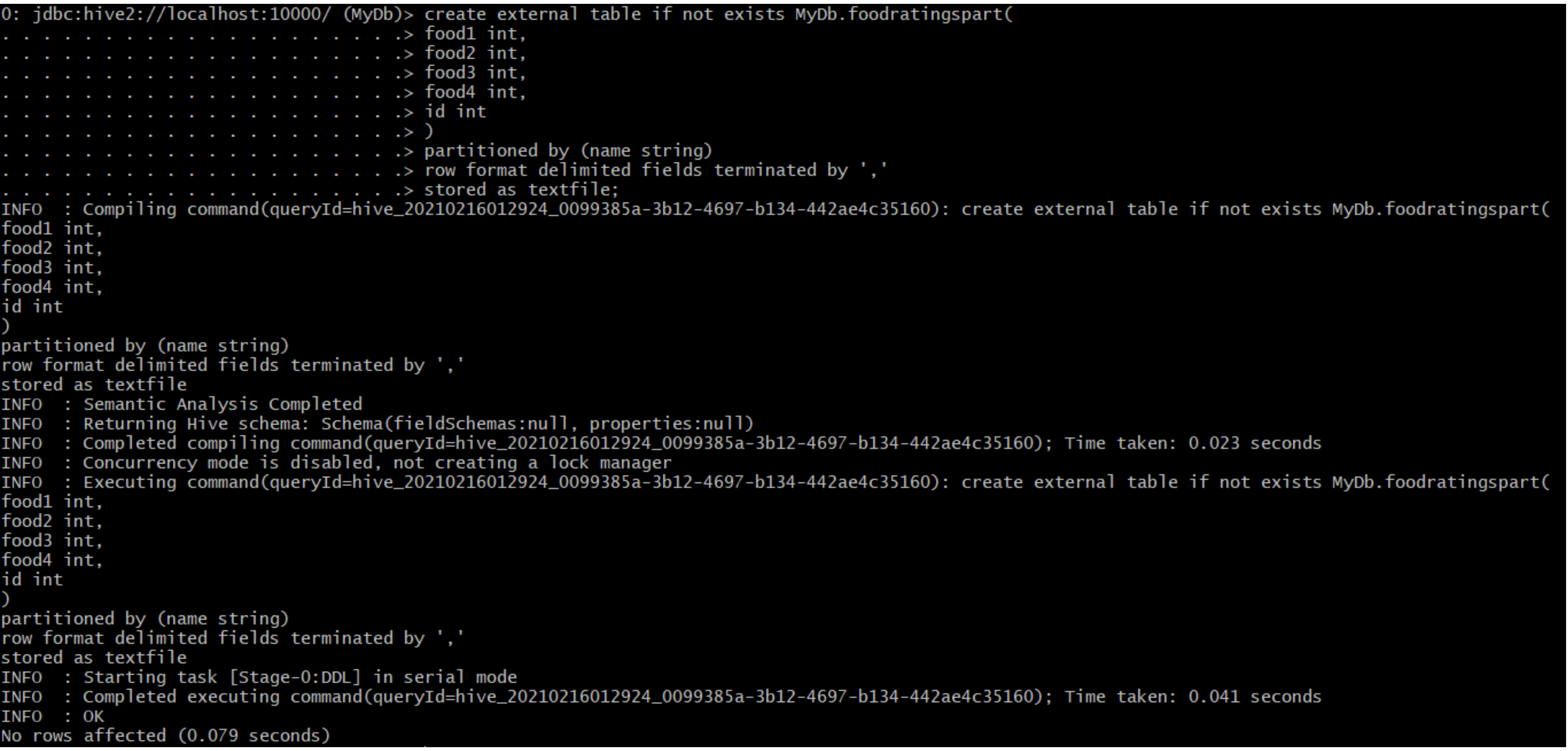
Exercise 4) 2 points

In MyDb create a partitioned table called ‘foodratingspart’

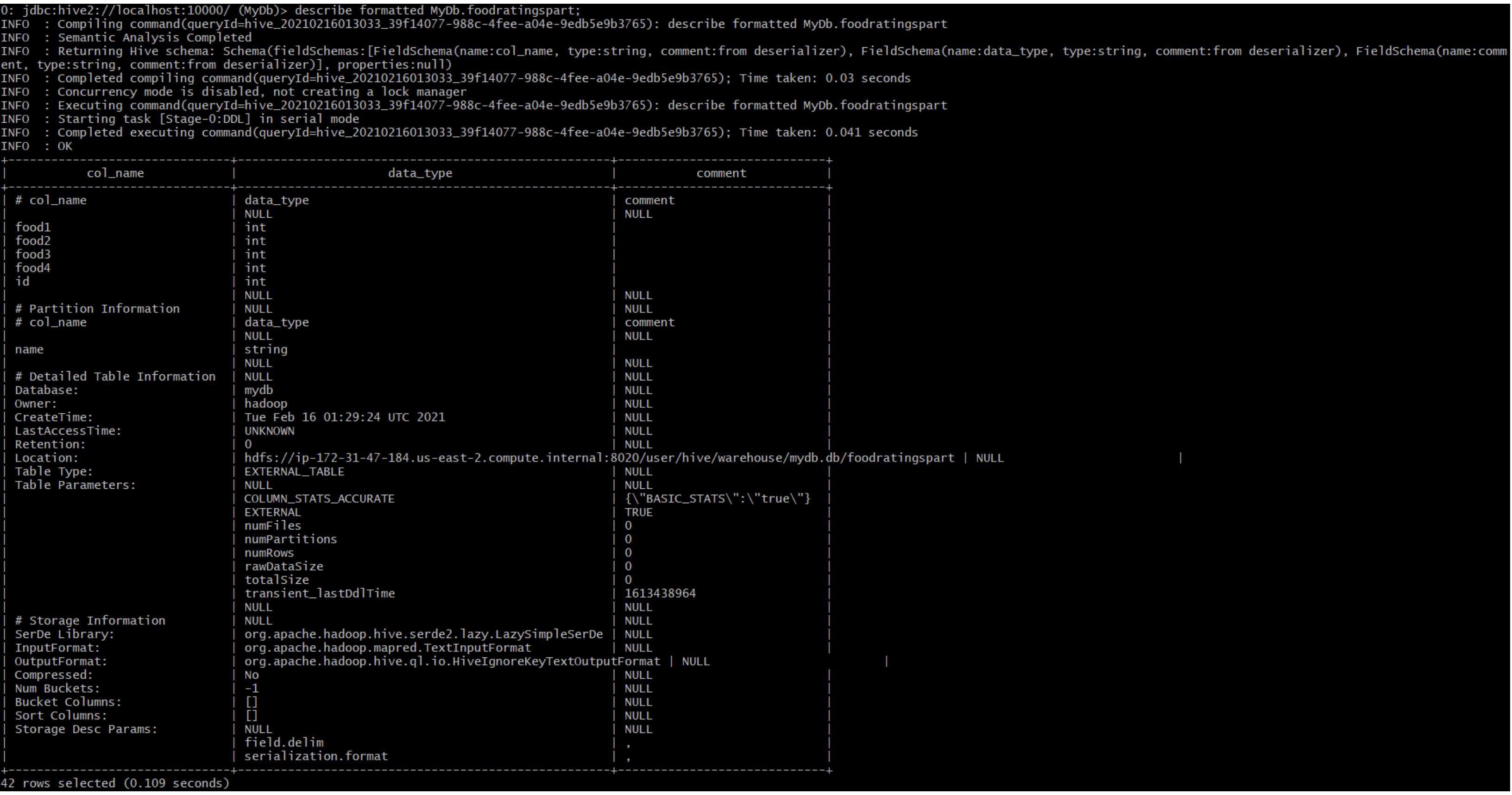
The partition field should be called ‘name’ and its type should be a string. The names of the non-partition columns should be food1, food2, food3, food4 and id and their types each an integer. The table should have storage format TEXTFILE and column separator a “,”. That is the underlying format should be a CSV file. No comments are needed for this table.

Execute a Hive command of ‘DESCRIBE FORMATTED MyDb.foodratingspart;’ and capture its output as the result of this exercise.

Magic Number: 30102



```
0: jdbc:hive2://localhost:10000/ (MyDb)> create external table if not exists MyDb.foodratingspart(
. . . . . . . . . . . . . . . . . . . . .> food1 int,
. . . . . . . . . . . . . . . . . . . . .> food2 int,
. . . . . . . . . . . . . . . . . . . . .> food3 int,
. . . . . . . . . . . . . . . . . . . . .> food4 int,
. . . . . . . . . . . . . . . . . . . . .> id int
. . . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . . .> partitioned by (name string)
. . . . . . . . . . . . . . . . . . . . .> row format delimited fields terminated by ','
. . . . . . . . . . . . . . . . . . . . .> stored as textfile;
INFO  : Compiling command(queryId=hive_20210216012924_0099385a-3b12-4697-b134-442ae4c35160): create external table if not exists MyDb.foodratingspart(
food1 int,
food2 int,
food3 int,
food4 int,
id int
)
partitioned by (name string)
row format delimited fields terminated by ','
stored as textfile
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216012924_0099385a-3b12-4697-b134-442ae4c35160); Time taken: 0.023 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216012924_0099385a-3b12-4697-b134-442ae4c35160): create external table if not exists MyDb.foodratingspart(
food1 int,
food2 int,
food3 int,
food4 int,
id int
)
partitioned by (name string)
row format delimited fields terminated by ','
stored as textfile
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216012924_0099385a-3b12-4697-b134-442ae4c35160); Time taken: 0.041 seconds
INFO  : OK
No rows affected (0.079 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> describe formatted MyDb.foodratingspart;
INFO  : Compiling command(queryId=hive_20210216013033_39f14077-988c-4fee-a04e-9edb5e9b3765): describe formatted MyDb.foodratingspart
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comm
ent, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216013033_39f14077-988c-4fee-a04e-9edb5e9b3765); Time taken: 0.03 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216013033_39f14077-988c-4fee-a04e-9edb5e9b3765): describe formatted MyDb.foodratingspart
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20210216013033_39f14077-988c-4fee-a04e-9edb5e9b3765); Time taken: 0.041 seconds
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 Feb 16 01:29:24 UTC 2021                       | NULL                       |
| LastAccessTime:              | UNKNOWN                                            | NULL                       |
| Retention:                   | 0                                                  | NULL                       |
| Location:                    | hdfs://ip-172-31-47-184.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodratingspart | NULL                       |
| Table Type:                  | EXTERNAL_TABLE                                     | NULL                       |
| Table Parameters:            | NULL                                               | NULL                       |
|                              | COLUMN_STATS_ACCURATE                              | {\"BASIC_STATS\":\"true\"} |
|                              | EXTERNAL                                           | TRUE                       |
|                              | numFiles                                           | 0                          |
|                              | numPartitions                                      | 0                          |
|                              | numRows                                            | 0                          |
|                              | rawDataSize                                        | 0                          |
|                              | totalSize                                          | 0                          |
|                              | transient_lastDdlTime                              | 1613438964                 |
|                              | 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                               | ,                          |
+------------------------------+----------------------------------------------------+----------------------------+
42 rows selected (0.109 seconds)
```

Exercise 5) 2 points

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 may lead to deterioration in efficiency. With Hive partitioning, the main thing is not to over-partition. In data loading and data retrieval, partitions improve the overhead.
- You are more likely to have small files if you build a very large number of partitions with small chunks of data in each partition.
- In hadoop, a few larger file numbers are normally much quicker than a small number of larger files.

Exercise 6) 2 points

Configure Hive to allow dynamic partition creation as described in the lecture. Now, use a hive command to copy from MyDB.foodratings into MyDB.foodratingspart to create a partitioned table from a non-partitioned one.

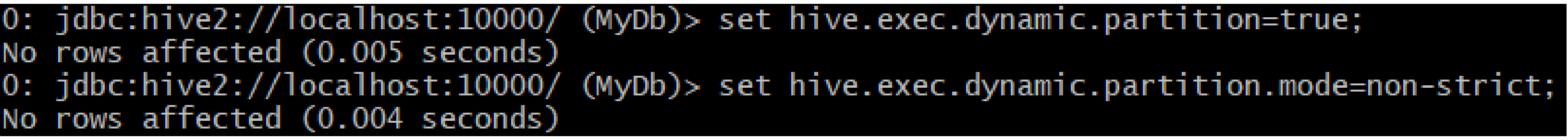
Hint: The ‘name’ column from MyDB.foodratings should be mentioned last in this command (whatever it is).

Provide a copy of the command you use to load the ‘foodratingspart’ table as a result of this exercise.

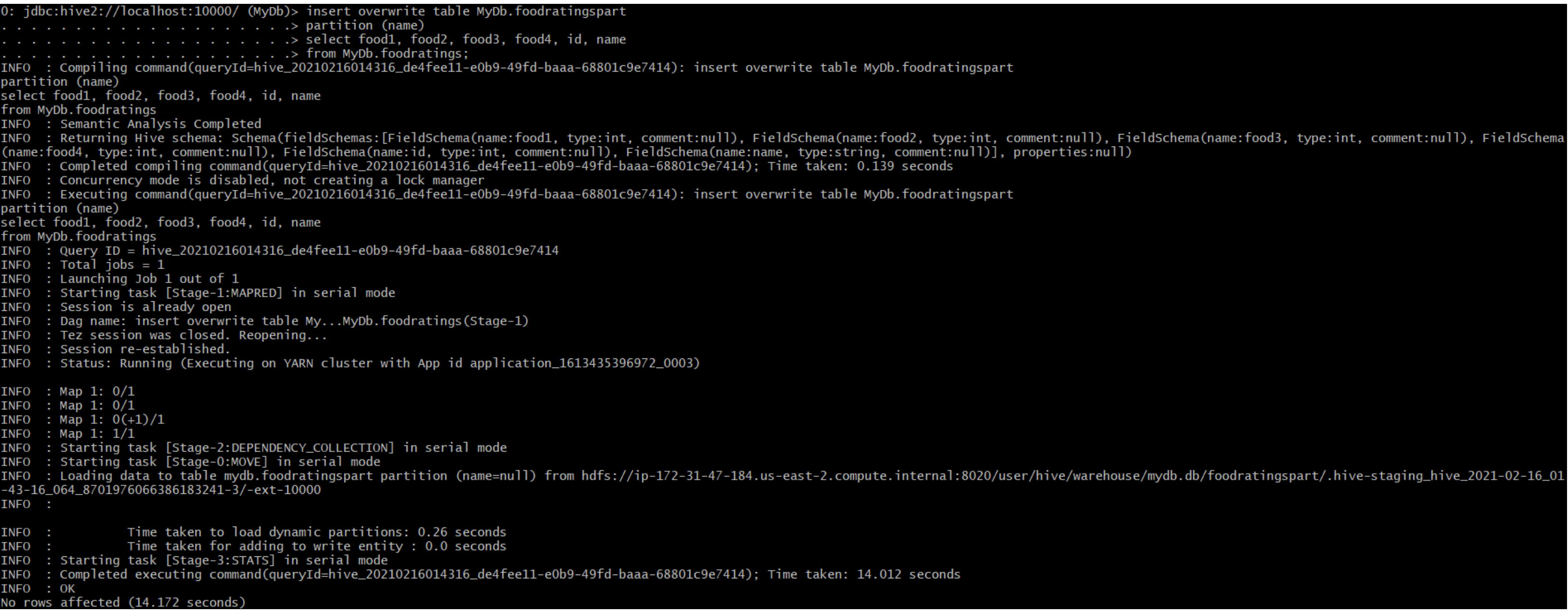
Execute a hive command to output the min, max and average of the values of the food2 column of MyDB.foodratingspart where the food critic ‘name’ is either Mel or Jill.

The query and the output of this query are other results of this exercise. It should look something like
10 20 15

Magic Number: 30102



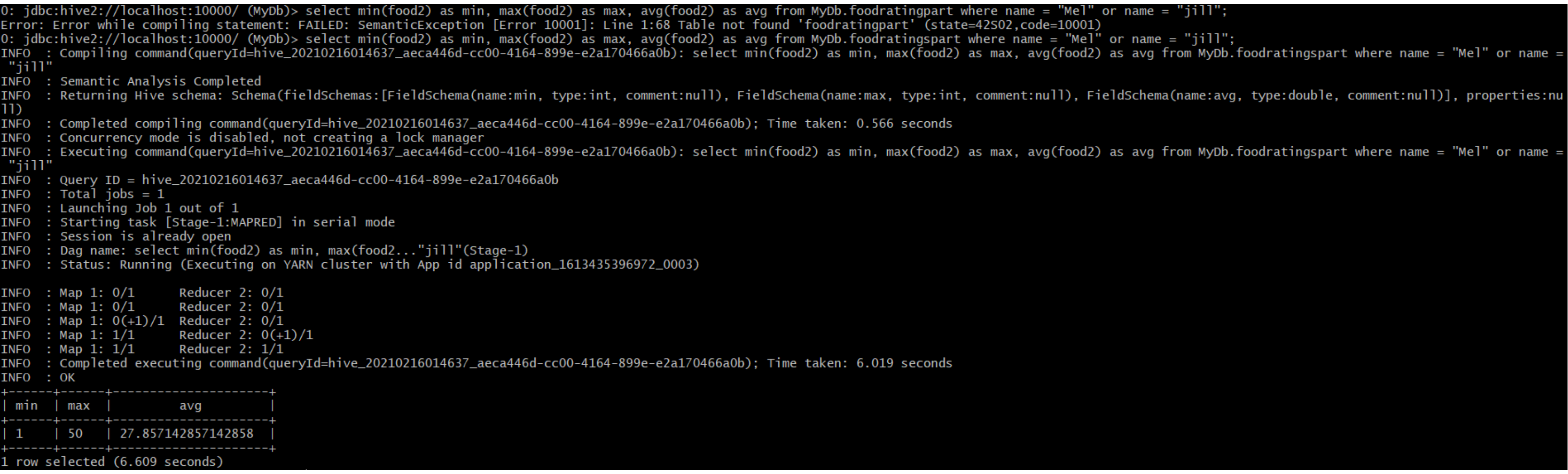
```
0: jdbc:hive2://localhost:10000/ (MyDb)> set hive.exec.dynamic.partition=true;
No rows affected (0.005 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)> set hive.exec.dynamic.partition.mode=non-strict;
No rows affected (0.004 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> insert overwrite table MyDb.foodratingspart
. . . . . . . . . . . . . . . . . . . .> partition (name)
. . . . . . . . . . . . . . . . . . . .> select food1, food2, food3, food4, id, name
. . . . . . . . . . . . . . . . . . . .> from MyDb.foodratings;
INFO  : Compiling command(queryId=hive_20210216014316_de4fee11-e0b9-49fd-baaa-68801c9e7414): insert overwrite table MyDb.foodratingspart
partition (name)
select food1, food2, food3, food4, id, name
from MyDb.foodratings
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:food1, type:int, comment:null), FieldSchema(name:food2, type:int, comment:null), FieldSchema(name:food3, type:int, comment:null), FieldSchema
(name:food4, type:int, comment:null), FieldSchema(name:id, type:int, comment:null), FieldSchema(name:name, type:string, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216014316_de4fee11-e0b9-49fd-baaa-68801c9e7414); Time taken: 0.139 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216014316_de4fee11-e0b9-49fd-baaa-68801c9e7414): insert overwrite table MyDb.foodratingspart
partition (name)
select food1, food2, food3, food4, id, name
from MyDb.foodratings
INFO  : Query ID = hive_20210216014316_de4fee11-e0b9-49fd-baaa-68801c9e7414
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Session is already open
INFO  : Dag name: insert overwrite table My...MyDb.foodratings(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  : Status: Running (Executing on YARN cluster with App id application_1613435396972_0003)

INFO  : Map 1: 0/1
INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
INFO  : Starting task [Stage-2:DEPENDENCY_COLLECTION] in serial mode
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table mydb.foodratingspart partition (name=null) from hdfs://ip-172-31-47-184.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodratingspart/.hive-staging_hive_2021-02-16_01
-43-16_064_8701976066386183241-3/-ext-10000
INFO  :

INFO  :          Time taken to load dynamic partitions: 0.26 seconds
INFO  :          Time taken for adding to write entity : 0.0 seconds
INFO  : Starting task [Stage-3:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20210216014316_de4fee11-e0b9-49fd-baaa-68801c9e7414); Time taken: 14.012 seconds
INFO  : OK
No rows affected (14.172 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> select min(food2) as min, max(food2) as max, avg(food2) as avg from MyDb.foodratingpart where name = "Mel" or name = "jill";
Error: Error while compiling statement: FAILED: SemanticException [Error 10001]: Line 1:68 Table not found 'foodratingpart' (state=42S02,code=10001)
0: jdbc:hive2://localhost:10000/ (MyDb)> select min(food2) as min, max(food2) as max, avg(food2) as avg from MyDb.foodratingspart where name = "Mel" or name = "jill";
INFO  : Compiling command(queryId=hive_20210216014637_aeca446d-cc00-4164-899e-e2a170466a0b): select min(food2) as min, max(food2) as max, avg(food2) as avg from MyDb.foodratingspart where name = "Mel" or name =
 "jill"
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:min, type:int, comment:null), FieldSchema(name:max, type:int, comment:null), FieldSchema(name:avg, type:double, comment:null)], properties:nu
ll)
INFO  : Completed compiling command(queryId=hive_20210216014637_aeca446d-cc00-4164-899e-e2a170466a0b); Time taken: 0.566 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216014637_aeca446d-cc00-4164-899e-e2a170466a0b): select min(food2) as min, max(food2) as max, avg(food2) as avg from MyDb.foodratingspart where name = "Mel" or name =
 "jill"
INFO  : Query ID = hive_20210216014637_aeca446d-cc00-4164-899e-e2a170466a0b
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Session is already open
INFO  : Dag name: select min(food2) as min, max(food2..."jill"(Stage-1)
INFO  : Status: Running (Executing on YARN cluster with App id application_1613435396972_0003)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Completed executing command(queryId=hive_20210216014637_aeca446d-cc00-4164-899e-e2a170466a0b); Time taken: 6.019 seconds
INFO  : OK
+------+------+---------------------+
| min  | max  |         avg         |
+------+------+---------------------+
| 1    | 50   | 27.857142857142858  |
+------+------+---------------------+
1 row selected (6.609 seconds)
```

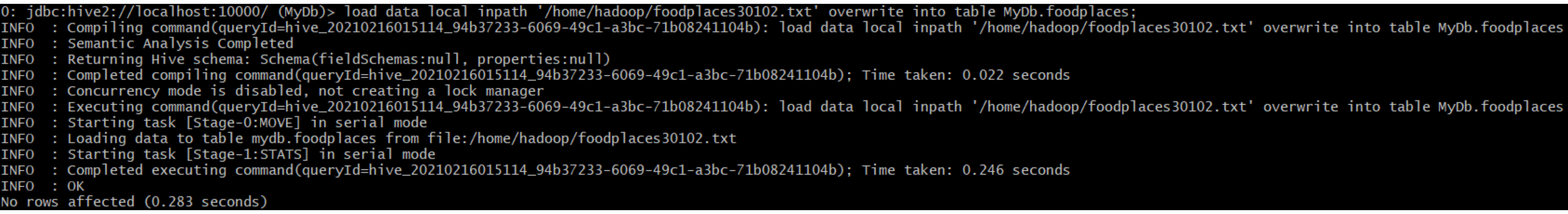
Exercise 7) 2 points

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

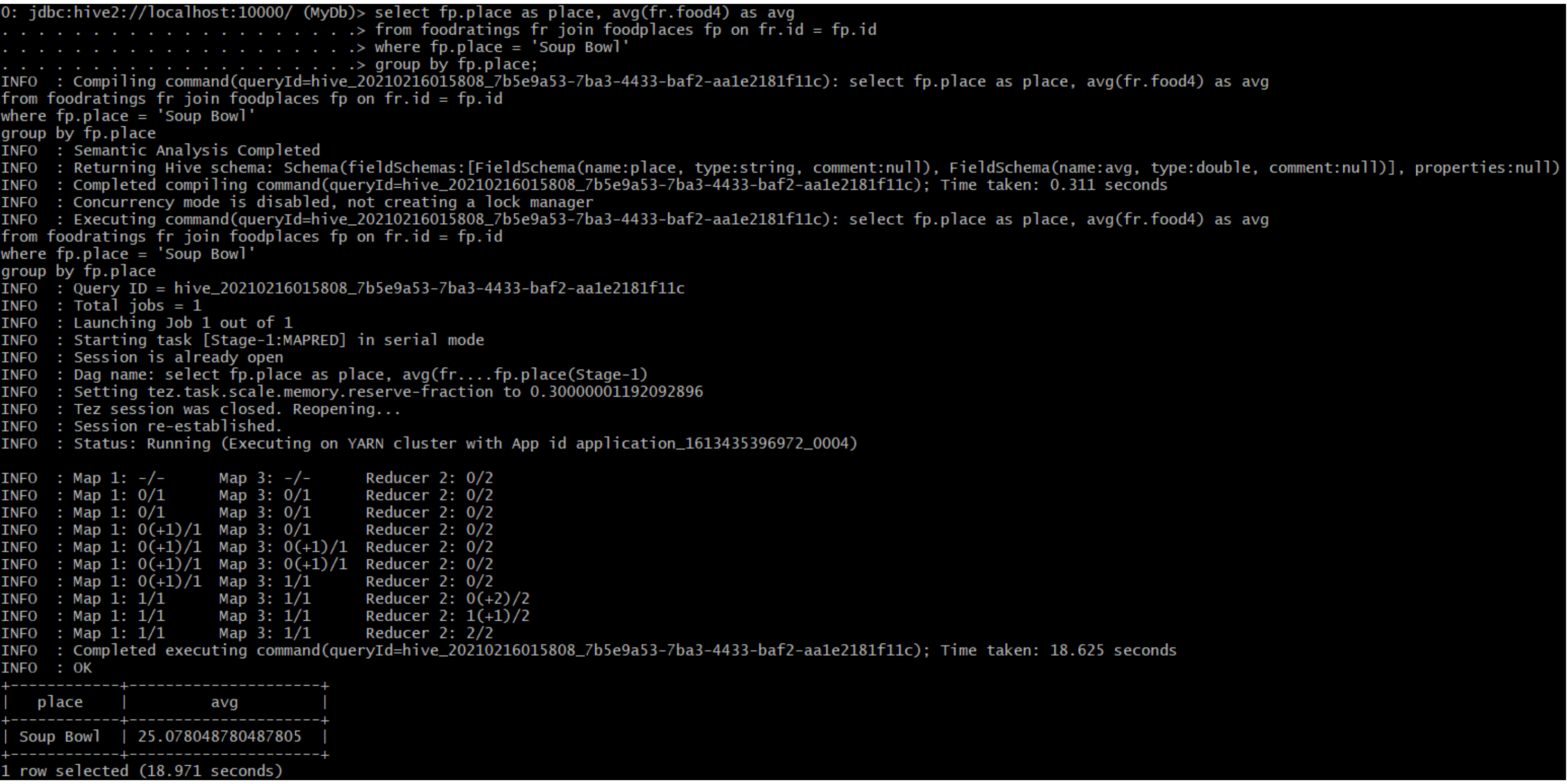
Use a join operation between the two tables (foodratings and foodplaces) to provide the average rating for field food4 for the restaurant ‘Soup Bowl’

The output of this query is the result of this exercise. It should look something like
Soup Bowl 20

Magic Number: 30102



```
0: jdbc:hive2://localhost:10000/ (MyDb)> load data local inpath '/home/hadoop/foodplaces30102.txt' overwrite into table MyDb.foodplaces;
INFO  : Compiling command(queryId=hive_20210216015114_94b37233-6069-49c1-a3bc-71b08241104b): load data local inpath '/home/hadoop/foodplaces30102.txt' overwrite into table MyDb.foodplaces
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20210216015114_94b37233-6069-49c1-a3bc-71b08241104b); Time taken: 0.022 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216015114_94b37233-6069-49c1-a3bc-71b08241104b): load data local inpath '/home/hadoop/foodplaces30102.txt' overwrite into table MyDb.foodplaces
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table mydb.foodplaces from file:/home/hadoop/foodplaces30102.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20210216015114_94b37233-6069-49c1-a3bc-71b08241104b); Time taken: 0.246 seconds
INFO  : OK
No rows affected (0.283 seconds)
```



```
0: jdbc:hive2://localhost:10000/ (MyDb)> select fp.place as place, avg(fr.food4) as avg
. . . . . . . . . . . . . . . . . . . .> from foodratings fr join foodplaces fp on fr.id = fp.id
. . . . . . . . . . . . . . . . . . . .> where fp.place = 'Soup Bowl'
. . . . . . . . . . . . . . . . . . . .> group by fp.place;
INFO  : Compiling command(queryId=hive_20210216015808_7b5e9a53-7ba3-4433-baf2-aa1e2181f11c): select fp.place as place, avg(fr.food4) as avg
from foodratings fr join foodplaces fp on fr.id = fp.id
where fp.place = 'Soup Bowl'
group by fp.place
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:place, type:string, comment:null), FieldSchema(name:avg, type:double, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20210216015808_7b5e9a53-7ba3-4433-baf2-aa1e2181f11c); Time taken: 0.311 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20210216015808_7b5e9a53-7ba3-4433-baf2-aa1e2181f11c): select fp.place as place, avg(fr.food4) as avg
from foodratings fr join foodplaces fp on fr.id = fp.id
where fp.place = 'Soup Bowl'
group by fp.place
INFO  : Query ID = hive_20210216015808_7b5e9a53-7ba3-4433-baf2-aa1e2181f11c
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Session is already open
INFO  : Dag name: select fp.place as place, avg(fr....fp.place(Stage-1)
INFO  : Setting tez.task.scale.memory.reserve-fraction to 0.30000001192092896
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  : Status: Running (Executing on YARN cluster with App id application_1613435396972_0004)

INFO  : Map 1: -/-      Map 3: -/-      Reducer 2: 0/2
INFO  : Map 1: 0/1      Map 3: 0/1      Reducer 2: 0/2
INFO  : Map 1: 0/1      Map 3: 0/1      Reducer 2: 0/2
INFO  : Map 1: 0(+1)/1  Map 3: 0/1      Reducer 2: 0/2
INFO  : Map 1: 0(+1)/1  Map 3: 0(+1)/1  Reducer 2: 0/2
INFO  : Map 1: 0(+1)/1  Map 3: 0(+1)/1  Reducer 2: 0/2
INFO  : Map 1: 0(+1)/1  Map 3: 1/1      Reducer 2: 0/2
INFO  : Map 1: 1/1      Map 3: 1/1      Reducer 2: 0(+2)/2
INFO  : Map 1: 1/1      Map 3: 1/1      Reducer 2: 1(+1)/2
INFO  : Map 1: 1/1      Map 3: 1/1      Reducer 2: 2/2
INFO  : Completed executing command(queryId=hive_20210216015808_7b5e9a53-7ba3-4433-baf2-aa1e2181f11c); Time taken: 18.625 seconds
INFO  : OK
+------------+---------------------+
|   place    |         avg         |
+------------+---------------------+
| Soup Bowl  | 25.078048780487805  |
+------------+---------------------+
1 row selected (18.971 seconds)
```

Exercise 8) 4 points

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?
    - The simplest type of the data table is the row format and is used in many applications, from web log files to highly structured database systems such as MySQL and Oracle. It is ideal for circumstances where simultaneous processing of the entire data row is required. When you want to use many of the fields associated with an entry, it is most helpful and you need to use many entries.
    - Data is stored sequentially by column, from top to bottom, not by row, left to right, in column formats. The sequential storing of data by column makes it easier to search the data faster since all related values are stored next to each other. It is also suitable for sparse data sets where empty values can be present.

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

- If the query calculation involves a single column at a time, a column format would be more vulnerable to breaking into separate jobs. In this paper, the columnar formats we address are row-columnar, meaning they take a batch of rows and store the batch in columnar format. Such lots then become independent boundaries.

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

- The column format is better than the row format to achieve better compression rates. Storing values by column, with the same form next to each other, helps you to compact them more efficiently than if you are storing data rows.

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

- Between quick data ingestion, fast random data search, and scalable data analytics, Apache Parquet has given very good versatility. The metadata for the Parquet file column is stored at the end of the file, which enables one-pass, quick writing. Details such as data types, compression/encoding, statistics, elements and more can be included in the metadata.