CSP 554 BIG DATA TECHNOLOGIES
MOHAMMED WASIM R D(A20497053)

ASSIGNMENT 4

Step 1: Assigning the key to private and connecting it to EMR

chmod 400/Users/mohammedwasimrd/Downloads/assign4.pem

ssh -i /Users/mohammedwasimrd/Downloads/assign4.pem hadoop@ec2-3-139-71-254.us-east-2.compute.amazonaws.com

```
Last login: Tue Sep 27 15:00:54 on ttys000
(base) mohammedwasimrd@dhcp66 ~ % chmod 400 /Users/mohammedwasimrd/Downloads/assign4.pem
(base) mohammedwasimrd@dhcp66 ~ % ssh -i /Users/mohammedwasimrd/Downloads/assign4.pem hadoop@ec2-3-139-71-254.us-east-2.compute.amazonaws.com
Last login: Tue Sep 27 19:52:32 2022 from 104.194.127.66

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
18 package(s) needed for security, out of 38 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M::::::M         M::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M:::::::M       M:::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M::::::::M     M::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

Step 2: Use scp to move the hql.zip demo file to your EMR master node (/home/hadoop) and decompress using unzip hql.zip

```
[hadoop@ip-172-31-26-9 ~]$ uzip hql.zip
-bash: uzip: command not found
[hadoop@ip-172-31-26-9 ~]$ unzip hql.zip
Archive:  hql.zip
   creating: hql/
  inflating: __MACOSX/._hql
  inflating: hql/salaries3.hql
  inflating: __MACOSX/hql/._salaries3.hql
  inflating: hql/salaries.hql
  inflating: __MACOSX/hql/._salaries.hql
  inflating: hql/salaries2.hql
  inflating: __MACOSX/hql/._salaries2.hql
  inflating: hql/demoreadme.txt
  inflating: __MACOSX/hql/._demoreadme.txt
  inflating: hql/loadsalaries.hql
  inflating: __MACOSX/hql/._loadsalaries.hql
  inflating: hql/basicsetup.hql
  inflating: __MACOSX/hql/._basicsetup.hql
  inflating: hql/partsetup.hql
  inflating: __MACOSX/hql/._partsetup.hql
  inflating: hql/Salaries.tsv
  inflating: __MACOSX/hql/._Salaries.tsv
[hadoop@ip-172-31-26-9 ~]$ cd /home/hadoop/hql
[hadoop@ip-172-31-26-9 hql]$
```

Step 3: scp the TestDataGen file over to the home directory (/home/hadoop) and execute the java code for magic number.

```
[hadoop@ip-172-31-26-9 ~]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop hdfsadmingroup       2189 2022-09-27 19:47 TestDataGen.class
[hadoop@ip-172-31-26-9 ~]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop hdfsadmingroup       2189 2022-09-27 19:47 TestDataGen.class
[hadoop@ip-172-31-26-9 ~]$ java TestDataGen
Magic Number = 189894
[hadoop@ip-172-31-26-9 ~]$
```

Magic Number : 189894

Step 4: It will also place the files foodratings<magic number>.txt and foodplaces<magic number>.txt in your VM home directory

```
[hadoop@ip-172-31-26-9 ~]$ ls
foodplaces189894.txt  foodratings189894.txt  hql      __MACOSX
foodplaces88502.txt   foodratings88502.txt   hql.zip  TestDataGen.class
[hadoop@ip-172-31-26-9 ~]$
```

Step 5: Starting up the hive by beeline

```
[hadoop@ip-172-31-26-9 ~]$ cd /home/hadoop/hql
[hadoop@ip-172-31-26-9 hql]$ beeline -u jdbc:hive2://localhost:10000/ -n hadoop -d org.apache.hive.jdbc.HiveDriver --showDbInPrompt
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 2.3.9-amzn-2)
Driver: Hive JDBC (version 2.3.9-amzn-2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.9-amzn-2 by Apache Hive
0: jdbc:hive2://localhost:10000/ (default)> source ./basicsetup.hql;
22/09/27 20:05:24 [main]: WARN conf.HiveConf: HiveConf of name hive.optimize.joinreducededuplication does not exist
22/09/27 20:05:24 [main]: WARN conf.HiveConf: HiveConf of name hive.server2.materializedviews.registry.impl does not exist
No rows affected (0.012 seconds)
No rows affected (0.021 seconds)
0: jdbc:hive2://localhost:10000/ (default)> source ./partsetup.hql;
22/09/27 20:06:06 [main]: WARN conf.HiveConf: HiveConf of name hive.optimize.joinreducededuplication does not exist
22/09/27 20:06:06 [main]: WARN conf.HiveConf: HiveConf of name hive.server2.materializedviews.registry.impl does not exist
No rows affected (0.018 seconds)
No rows affected (0.022 seconds)
No rows affected (0.011 seconds)
No rows affected (0.013 seconds)
0: jdbc:hive2://localhost:10000/ (default)> source ./salaries.hql;
22/09/27 20:06:15 [main]: WARN conf.HiveConf: HiveConf of name hive.optimize.joinreducededuplication does not exist
22/09/27 20:06:15 [main]: WARN conf.HiveConf: HiveConf of name hive.server2.materializedviews.registry.impl does not exist
INFO  : Compiling command(queryId=hive_20220927200615_7367817c-b9f4-4cf4-bf4c-66721352087c): CREATE DATABASE IF NOT EXISTS cs595
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927200615_7367817c-b9f4-4cf4-bf4c-66721352087c : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0
```

Step 6 : Create a Hive database called “MyDb”.

Create database MyDb;

```
0: jdbc:hive2://localhost:10000/ (cs595)> create database MyDb;
INFO  : Compiling command(queryId=hive_20220927202449_a9dbbfc9-f15c-4a45-a8aa-36634de15f0d): create database MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927202449_a9dbbfc9-f15c-4a45-a8aa-36634de15f0d : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0


INFO  : Completed compiling command(queryId=hive_20220927202449_a9dbbfc9-f15c-4a45-a8aa-36634de15f0d); Time taken: 0.083 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927202449_a9dbbfc9-f15c-4a45-a8aa-36634de15f0d): create database MyDb
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927202449_a9dbbfc9-f15c-4a45-a8aa-36634de15f0d); Time taken: 0.043 seconds
INFO  : OK
No rows affected (0.136 seconds)
0: jdbc:hive2://localhost:10000/ (cs595)> use MyDb;
INFO  : Compiling command(queryId=hive_20220927203208_2f86386e-e62f-4b2a-80f6-560160b5e7dc): use MyDb
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927203208_2f86386e-e62f-4b2a-80f6-560160b5e7dc : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0
```

Step 7 : In MyDb create a table with name foodratings

```
CREATE TABLE IF NOT EXISTS foodratings(

name STRING COMMENT 'Food Critic Name',
food1 INT COMMENT 'Ratings for food1',
food2 INT COMMENT 'Ratings for food2',
food3 INT COMMENT 'Ratings for food3',
food4 INT COMMENT 'Ratings for food4',
id INT COMMENT 'Food id'
)
COMMENT 'Food rating table'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

```
0: jdbc:hive2://localhost:10000/ (MyDb)>
0: jdbc:hive2://localhost:10000/ (MyDb)> CREATE TABLE IF NOT EXISTS foodratings(
. . . . . .> name STRING COMMENT 'Food Critic Name',
. . . . . .> food1 INT COMMENT 'Ratings for food1',
. . . . . .> food2 INT COMMENT 'Ratings for food2',
. . . . . .> food3 INT COMMENT 'Ratings for food3',
. . . . . .> food4 INT COMMENT 'Ratings for food4',
. . . . . .> id INT COMMENT 'Food id'
. . . . . .> )
. . . . . .> COMMENT 'Food rating table'
. . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE;
INFO  : Compiling command(queryId=hive_20220927210120_d7c4b8b5-8f55-4f9b-aa34-0558155a7ad0): CREATE TABLE IF NOT EXISTS foodratings(
name STRING COMMENT 'Food Critic Name',
food1 INT COMMENT 'Ratings for food1',
food2 INT COMMENT 'Ratings for food2',
food3 INT COMMENT 'Ratings for food3',
food4 INT COMMENT 'Ratings for food4',
id INT COMMENT 'Food id'
)
COMMENT 'Food rating table'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927210120_d7c4b8b5-8f55-4f9b-aa34-0558155a7ad0 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0
      Create Table Operator:
        Create Table
          columns: name string Food Critic Name, food1 int Ratings for food1, food2 int Ratings for food2, food3 int Ratings for food3, food4 int Ratings for food4, id int Food id
          comment: Food rating table
          field delimiter: ,
          if not exists: true
          input format: org.apache.hadoop.mapred.TextInputFormat
          output format: org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat
          serde name: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
          name: MyDb.foodratings


INFO  : Completed compiling command(queryId=hive_20220927210120_d7c4b8b5-8f55-4f9b-aa34-0558155a7ad0); Time taken: 0.055 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927210120_d7c4b8b5-8f55-4f9b-aa34-0558155a7ad0): CREATE TABLE IF NOT EXISTS foodratings(
name STRING COMMENT 'Food Critic Name',
food1 INT COMMENT 'Ratings for food1',
food2 INT COMMENT 'Ratings for food2',
food3 INT COMMENT 'Ratings for food3',
food4 INT COMMENT 'Ratings for food4',
id INT COMMENT 'Food id'
)
COMMENT 'Food rating table'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927210120_d7c4b8b5-8f55-4f9b-aa34-0558155a7ad0); Time taken: 0.12 seconds
INFO  : OK
No rows affected (0.206 seconds)
```

Step 8 : Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodratings

DESCRIBE FORMATTED MyDb.foodratings;

```
0: jdbc:hive2://localhost:10000/ (MyDb)> DESCRIBE FORMATTED MyDb.foodratings;
INFO  : Compiling command(queryId=hive_20220927210644_59e9ae11-9c2a-4922-b994-e6eed8adfb39): 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(nam
e:comment, type:string, comment:from deserializer)], properties:null)
INFO  : EXPLAIN output for queryid hive_20220927210644_59e9ae11-9c2a-4922-b994-e6eed8adfb39 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]
  Stage-1 depends on stages: Stage-0 [FETCH]

STAGE PLANS:
  Stage: Stage-0
      Describe Table Operator:
        Describe Table
          result file: file:/mnt/tmp/hive/5ee96600-1c84-4849-a39c-c9769060664f/hive_2022-09-27_21-06-44_938_4514556745365950548-1/-local-10000
          table: MyDb.foodratings

  Stage: Stage-1
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink


INFO  : Completed compiling command(queryId=hive_20220927210644_59e9ae11-9c2a-4922-b994-e6eed8adfb39); Time taken: 0.06 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927210644_59e9ae11-9c2a-4922-b994-e6eed8adfb39): DESCRIBE FORMATTED MyDb.foodratings
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927210644_59e9ae11-9c2a-4922-b994-e6eed8adfb39); Time taken: 0.118 seconds
INFO  : OK
```

```
+-------------------------------+----------------------------------------------------+-----------------------------+
|           col_name            |                     data_type                      |           comment           |
+-------------------------------+----------------------------------------------------+-----------------------------+
| # col_name                    | data_type                                          | comment                     |
|                               | NULL                                               | NULL                        |
| name                          | string                                             | Food Critic Name            |
| food1                         | int                                                | Ratings for food1           |
| food2                         | int                                                | Ratings for food2           |
| food3                         | int                                                | Ratings for food3           |
| food4                         | int                                                | Ratings for food4           |
| id                            | int                                                | Food id                     |
|                               | NULL                                               | NULL                        |
| # Detailed Table Information  | NULL                                               | NULL                        |
| Database:                     | mydb                                               | NULL                        |
| Owner:                        | hadoop                                             | NULL                        |
| CreateTime:                   | Tue Sep 27 21:01:20 UTC 2022                       | NULL                        |
| LastAccessTime:               | UNKNOWN                                            | NULL                        |
| Retention:                    | 0                                                  | NULL                        |
| Location:                     | hdfs://ip-172-31-26-9.us-east-2.compute.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                              | 1664312480                  |
|                               | 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 (0.308 seconds)
```

Step 9 : In MyDb create a table with name foodplaces

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

```
0: jdbc:hive2://localhost:10000/ (MyDb)> CREATE IF NOT EXISTS foodplaces (
. . . . . . . . . . . . . . . . . . . .> id INT,
. . . . . . . . . . . . . . . . . . . .> place String
. . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . . . . . . . . . . . . . . . .> STORED AS TEXTFILE;
Error: Error while compiling statement: FAILED: ParseException line 1:7 cannot recognize input near 'CREATE' 'IF' 'NOT' in ddl statement (state=42000,code=40000)
0: jdbc:hive2://localhost:10000/ (MyDb)>
0: jdbc:hive2://localhost:10000/ (MyDb)> CREATE TABLE IF NOT EXISTS foodplaces (
. . . . . . . . . . . . . . . . . . . .> id INT,
. . . . . . . . . . . . . . . . . . . .> place String
. . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . . . . . . . . . . . . . . . .> STORED AS TEXTFILE;
INFO  : Compiling command(queryId=hive_20220927211044_de0940fd-4ec1-41fe-8bfd-42f996b4f126): CREATE TABLE IF NOT EXISTS foodplaces (
id INT,
place String
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927211044_de0940fd-4ec1-41fe-8bfd-42f996b4f126 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0
      Create Table Operator:
        Create Table
          columns: id int, place string
          field delimiter: ,
          if not exists: true
          input format: org.apache.hadoop.mapred.TextInputFormat
          output format: org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat
          serde name: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
          name: MyDb.foodplaces


INFO  : Completed compiling command(queryId=hive_20220927211044_de0940fd-4ec1-41fe-8bfd-42f996b4f126); Time taken: 0.028 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927211044_de0940fd-4ec1-41fe-8bfd-42f996b4f126): CREATE TABLE IF NOT EXISTS foodplaces (
id INT,
place String
)
```

Step 10 : Execute a Hive command of DESCRIBE FORMATTED MyDb.foodplaces.

```
0: jdbc:hive2://localhost:10000/ (MyDb)> DESCRIBE FORMATTED MyDb.foodplaces;
INFO  : Compiling command(queryId=hive_20220927211118_b303cd2d-afb3-466e-a5c9-15c709b923e2): 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(nam
e:comment, type:string, comment:from deserializer)], properties:null)
INFO  : EXPLAIN output for queryid hive_20220927211118_b303cd2d-afb3-466e-a5c9-15c709b923e2 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]
  Stage-1 depends on stages: Stage-0 [FETCH]

STAGE PLANS:
  Stage: Stage-0
      Describe Table Operator:
        Describe Table
          result file: file:/mnt/tmp/hive/5ee96600-1c84-4849-a39c-c9769060664f/hive_2022-09-27_21-11-18_221_3618286039858724718-1/-local-10000
          table: MyDb.foodplaces

  Stage: Stage-1
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink


INFO  : Completed compiling command(queryId=hive_20220927211118_b303cd2d-afb3-466e-a5c9-15c709b923e2); Time taken: 0.037 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927211118_b303cd2d-afb3-466e-a5c9-15c709b923e2): DESCRIBE FORMATTED MyDb.foodplaces
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20220927211118_b303cd2d-afb3-466e-a5c9-15c709b923e2); Time taken: 0.014 seconds
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 21:10:44 UTC 2022                       | NULL                        |
| LastAccessTime:               | UNKNOWN                                            | NULL                        |
| Retention:                    | 0                                                  | NULL                        |
| Location:                     | hdfs://ip-172-31-26-9.us-east-2.compute.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                              | 1664313044                  |
|                               | 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                                        | ,                           |
```

Step 11 : Load the foodratings<magic number>.txt file created using TestDataGen from your local file system into the foodratings table.

LOAD DATA LOCAL INPATH '/home/hadoop/foodratings189894.txt'
OVERWRITE INTO TABLE foodratings;

```
0: jdbc:hive2://localhost:10000/ (MyDb)> LOAD DATA LOCAL INPATH '/home/hadoop/foodratings189894.txt'
. . . . . . . . . . . . . . . . . . . .> OVERWRITE INTO TABLE foodratings;
INFO  : Compiling command(queryId=hive_20220927213000_0dfcce12-5d34-455a-9eaf-a93dd22601f2): LOAD DATA LOCAL INPATH '/home/hadoop/foodratings189894.txt'
OVERWRITE INTO TABLE foodratings
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927213000_0dfcce12-5d34-455a-9eaf-a93dd22601f2 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [MOVE]
  Stage-1 depends on stages: Stage-0 [STATS]

STAGE PLANS:
  Stage: Stage-0
    Move Operator
      tables:
          replace: true
          source: file:/home/hadoop/foodratings189894.txt
          table:
              input format: org.apache.hadoop.mapred.TextInputFormat
              output format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
              properties:
                COLUMN_STATS_ACCURATE {"BASIC_STATS":"true"}
                bucket_count -1
                column.name.delimiter ,
                columns name,food1,food2,food3,food4,id
                columns.comments 'Food Critic Name','Ratings for food1','Ratings for food2','Ratings for food3','Ratings for food4','Food id'
                columns.types string:int:int:int:int:int
                comment Food rating table
                field.delim ,
                file.inputformat org.apache.hadoop.mapred.TextInputFormat
                file.outputformat org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
                location hdfs://ip-172-31-26-9.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodratings
                name mydb.foodratings
                numFiles 0
                numRows 0
                rawDataSize 0
                serialization.ddl struct foodratings { string name, i32 food1, i32 food2, i32 food3, i32 food4, i32 id}
                serialization.format ,
                serialization.lib org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
                totalSize 0
                transient_lastDdlTime 1664312480
              serde: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
              name: mydb.foodratings

  Stage: Stage-1
    Stats-Aggr Operator
```

Step 12: Execute a hive command to output the min, max and average of the values of the food3 column of the foodratings table

SELECT min(food3) AS min, max(food3) AS max, avg(food3) as average FROM foodratings;

```
0: jdbc:hive2://localhost:10000/ (MyDb)> SELECT min(food3) AS min, max(food3) AS max, avg(food3) as average FROM foodratings;
INFO  : Compiling command(queryId=hive_20220927213703_5e5e22ac-0fcf-479e-868d-17c726a33a32): SELECT min(food3) AS min, max(food3) AS max, avg(food3) as average FROM foodratings
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:average, type:double, comment:null)], pro
perties:null)
INFO  : EXPLAIN output for queryid hive_20220927213703_5e5e22ac-0fcf-479e-868d-17c726a33a32 : STAGE DEPENDENCIES:
  Stage-1 is a root stage [MAPRED]
  Stage-0 depends on stages: Stage-1 [FETCH]

STAGE PLANS:
  Stage: Stage-1
    Tez
      DagId: hive_20220927213703_5e5e22ac-0fcf-479e-868d-17c726a33a32:2
      Edges:
        Reducer 2 <- Map 1 (CUSTOM_SIMPLE_EDGE)



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
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_20220927213703_5e5e22ac-0fcf-479e-868d-17c726a33a32); Time taken: 14.417 seconds
INFO  : OK
+------+------+----------+
| min  | max  | average  |
+------+------+----------+
| 1    | 50   | 25.61    |
+------+------+----------+
1 row selected (15.713 seconds)
```

Step 13 : 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) AS min, max(food1) AS max, avg(food1) as average FROM foodratings GROUP BY name;

```
0: jdbc:hive2://localhost:10000/ (MyDb)> SELECT name,min(food1) AS min, max(food1) AS max, avg(food1) as average FROM foodratings GROUP BY name;
INFO  : Compiling command(queryId=hive_20220927214952_8296fbe1-d735-4792-b07d-9412f2a273fc): SELECT name,min(food1) AS min, max(food1) AS max, avg(food1) as average FROM 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), FieldSc
hema(name:average, type:double, comment:null)], properties:null)
INFO  : EXPLAIN output for queryid hive_20220927214952_8296fbe1-d735-4792-b07d-9412f2a273fc : STAGE DEPENDENCIES:
  Stage-1 is a root stage [MAPRED]
  Stage-0 depends on stages: Stage-1 [FETCH]

STAGE PLANS:
  Stage: Stage-1
    Tez
      DagId: hive_20220927214952_8296fbe1-d735-4792-b07d-9412f2a273fc:3
      Edges:
        Reducer 2 <- Map 1 (SIMPLE_EDGE)
      DagName:
      Vertices:
```

```
INFO  : Map 1: 0/1      Reducer 2: 0/2
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: 0/2
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/2
INFO  : Map 1: 1/1      Reducer 2: 1(+1)/2
INFO  : Map 1: 1/1      Reducer 2: 2/2
INFO  : Completed executing command(queryId=hive_20220927214952_8296fbe1-d735-4792-b07d-9412f2a273fc); Time taken: 15.136 seconds
INFO  : OK
+-------+------+------+---------------------+
| name  | min  | max  |       average       |
+-------+------+------+---------------------+
| Jill  | 1    | 50   | 25.35175879396985   |
| Joe   | 1    | 50   | 25.200934579439252  |
| Joy   | 1    | 50   | 23.354166666666668  |
| Mel   | 1    | 50   | 26.532994923857867  |
| Sam   | 1    | 50   | 25.12121212121212   |
+-------+------+------+---------------------+
5 rows selected (15.536 seconds)
```

Step 14 : In 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;
```

```
0: jdbc:hive2://localhost:10000/ (MyDb)> 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;
INFO  : Compiling command(queryId=hive_20220927215310_cb0c1891-f11b-4010-9fe1-4e9228658052): 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
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927215310_cb0c1891-f11b-4010-9fe1-4e9228658052 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [DDL]

STAGE PLANS:
  Stage: Stage-0
      Create Table Operator:
        Create Table
          columns: food1 int, food2 int, food3 int, food4 int, id int
          field delimiter: ,
          if not exists: true
          input format: org.apache.hadoop.mapred.TextInputFormat
          output format: org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat
          partition columns: name string
          serde name: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
          name: MyDb.foodratingspart


INFO  : Completed compiling command(queryId=hive_20220927215310_cb0c1891-f11b-4010-9fe1-4e9228658052); Time taken: 0.034 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927215310_cb0c1891-f11b-4010-9fe1-4e9228658052): CREATE TABLE IF NOT EXISTS foodratingspart (
food1 INT,
food2 INT,
food3 INT,
food4 INT,
id INT
)
PARTITIONED BY (name STRING)
```

Step 15: Execute a Hive command of 'DESCRIBE FORMATTED MyDb.foodratingspart;

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 21:53:10 UTC 2022                       | NULL                        |
| LastAccessTime:               | UNKNOWN                                            | NULL                        |
| Retention:                    | 0                                                  | NULL                        |
| Location:                     | hdfs://ip-172-31-26-9.us-east-2.compute.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                              | 1664315590                  |
|                               | 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 (0.172 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)> set hive.exec.dynamic.partition.mode=nonstrict;
No rows affected (0.006 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)> INSERT OVERWRITE TABLE foodratingspart
```

Step 16: 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.

As a result, in the example scenario, if we establish a partition table for the name because there are only 10 spots, then during a search operation, it will just look for the specific partition rather than the entire table's 10,000 entries in the database. This dynamic partition will save time and improve performance. This approach will only be effective if there are significantly fewer critics than there are locations. If not, this position will be eliminated.

Step 17: Configure Hive to allow dynamic partition creation. use a hive command to copy from MyDB.foodratings into MyDB.foodratingspart to create a partitioned table

set hive.exec.dynamic.partition.mode=nonstrict;

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

```
0: jdbc:hive2://localhost:10000/ (MyDb)> set hive.exec.dynamic.partition.mode=nonstrict;
No rows affected (0.006 seconds)
0: jdbc:hive2://localhost:10000/ (MyDb)> INSERT OVERWRITE TABLE foodratingspart
. . . . . .> PARTITION(name)
. . . . . .> SELECT food1, food2, food3, food4, id, name
. . . . . .> FROM foodratings;
INFO  : Compiling command(queryId=hive_20220927221348_900d79d5-1ca8-4b9d-918c-0a878757337d): INSERT OVERWRITE TABLE foodratingspart
PARTITION(name)
SELECT food1, food2, food3, food4, id, name
FROM 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), Fi
Schema(name:food4, type:int, comment:null), FieldSchema(name:id, type:int, comment:null), FieldSchema(name:name, type:string, comment:null)], properties:null)
INFO  : EXPLAIN output for queryid hive_20220927221348_900d79d5-1ca8-4b9d-918c-0a878757337d : STAGE DEPENDENCIES:
  Stage-1 is a root stage [MAPRED]
  Stage-2 depends on stages: Stage-1 [DEPENDENCY_COLLECTION]
  Stage-0 depends on stages: Stage-2 [MOVE]
  Stage-3 depends on stages: Stage-0 [STATS]

STAGE PLANS:
  Stage: Stage-1
    Tez
      DagId: hive_20220927221348_900d79d5-1ca8-4b9d-918c-0a878757337d:4
      Edges:
        Reducer 2 <- Map 1 (SIMPLE_EDGE)
      DagName:
      Vertices:
        Map 1
            Map Operator Tree:
                TableScan
                  alias: foodratings
                  Statistics: Num rows: 145 Data size: 17473 Basic stats: COMPLETE Column stats: NONE
                  GatherStats: false
                  Select Operator
                    expressions: food1 (type: int), food2 (type: int), food3 (type: int), food4 (type: int), id (type: int), name (type: string)
                    outputColumnNames: _col0, _col1, _col2, _col3, _col4, _col5
                    Statistics: Num rows: 145 Data size: 17473 Basic stats: COMPLETE Column stats: NONE
                    Reduce Output Operator
                      key expressions: _col5 (type: string)
                      null sort order: a
                      sort order: +
                      Map-reduce partition columns: _col5 (type: string)
                      Statistics: Num rows: 145 Data size: 17473 Basic stats: COMPLETE Column stats: NONE
                      tag: -1
                      value expressions: _col0 (type: int), _col1 (type: int), _col2 (type: int), _col3 (type: int), _col4 (type: int)


                directory: hdfs://ip-172-31-26-9.us-east-2.compute.internal:8020/tmp/hive/hadoop/5ee96600-1c84-4849-a39c-c9769060664f/hive_2022-09-27_22-16-41_10
taging_hive_2022-09-27_22-16-41_100_3215624262394850508-1/-ext-10002
                NumFilesPerFileSink: 1
                Statistics: Num rows: 1 Data size: 84 Basic stats: COMPLETE Column stats: NONE
                Stats Publishing Key Prefix: hdfs://ip-172-31-26-9.us-east-2.compute.internal:8020/tmp/hive/hadoop/5ee96600-1c84-4849-a39c-c9769060664f/hive_2022
/-mr-10001/.hive-staging_hive_2022-09-27_22-16-41_100_3215624262394850508-1/-ext-10002/
                table:
                    input format: org.apache.hadoop.mapred.SequenceFileInputFormat
                    output format: org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat
                    properties:
                      columns _col0,_col1,_col2
                      columns.types int:int:double
                      escape.delim \
                      hive.serialization.extend.additional.nesting.levels true
                      serialization.escape.crlf true
                      serialization.format 1
                      serialization.lib org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
                    serde: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
                TotalFiles: 1
                GatherStats: false
                MultiFileSpray: false

  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink


INFO  : Completed compiling command(queryId=hive_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9); Time taken: 0.759 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9): select min(food2) as min, max(food2) as max, avg(food2) as average
from foodratingspart
where name='Mel' or name = 'Jill'
INFO  : Query ID = hive_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9
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_1664307341610_0004)

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_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9); Time taken: 6.43 seconds
INFO  : OK
```

Step 18: Execute a hive command to output the min, max and average of the values of the food2

select min(food2) as min, max(food2) as max, avg(food2) as average
from foodratingspart
where name='Mel' or name = 'Jill';

```
0: jdbc:hive2://localhost:10000/ (MyDb)> select min(food2) as min, max(food2) as max, avg(food2) as average
. . . . . . . . . . . . . . . . . . . .> from foodratingspart
. . . . . . . . . . . . . . . . . . . .> where name='Mel' or name = 'Jill';
INFO  : Compiling command(queryId=hive_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9): select min(food2) as min, max(food2) as max, avg(food2) as average
from 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:average, type:double, comment:null)], pro
perties:null)
INFO  : EXPLAIN output for queryid hive_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9 : STAGE DEPENDENCIES:
  Stage-1 is a root stage [MAPRED]
  Stage-0 depends on stages: Stage-1 [FETCH]

STAGE PLANS:
  Stage: Stage-1


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_20220927221641_1fd180a9-1e07-4179-949a-07e588ee94d9); Time taken: 6.43 seconds
INFO  : OK
+------+------+---------------------+
| min  | max  |       average       |
+------+------+---------------------+
| 1    | 50   | 25.353535353535353  |
+------+------+---------------------+
1 row selected (7.213 seconds)
```

Step 19: Load the foodplaces<.magic number>.txt file created using TestDataGen from your local file system into the foodplaces table.

LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces189894.txt'
OVERWRITE INTO TABLE foodplaces;

```
0: jdbc:hive2://localhost:10000/ (MyDb)> LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces189894.txt'
. . . . . . . . . . . . . . . . . . . .> OVERWRITE INTO TABLE mydb.foodplaces;
INFO  : Compiling command(queryId=hive_20220927221849_b0358ee7-5cb6-4551-afae-fe6100e4deb0): LOAD DATA LOCAL INPATH '/home/hadoop/foodplaces189894.txt'
OVERWRITE INTO TABLE mydb.foodplaces
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : EXPLAIN output for queryid hive_20220927221849_b0358ee7-5cb6-4551-afae-fe6100e4deb0 : STAGE DEPENDENCIES:
  Stage-0 is a root stage [MOVE]
  Stage-1 depends on stages: Stage-0 [STATS]

STAGE PLANS:
  Stage: Stage-0
    Move Operator
      tables:
          replace: true
          source: file:/home/hadoop/foodplaces189894.txt
          table:
              input format: org.apache.hadoop.mapred.TextInputFormat
              output format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
              properties:
                COLUMN_STATS_ACCURATE {"BASIC_STATS":"true"}
                bucket_count -1
                column.name.delimiter ,
                columns id,place
                columns.comments
                columns.types int:string
                field.delim ,
                file.inputformat org.apache.hadoop.mapred.TextInputFormat
                file.outputformat org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
                location hdfs://ip-172-31-26-9.us-east-2.compute.internal:8020/user/hive/warehouse/mydb.db/foodplaces
                name mydb.foodplaces
                numFiles 0
                numRows 0
                rawDataSize 0
                serialization.ddl struct foodplaces { i32 id, string place}
                serialization.format ,
                serialization.lib org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
                totalSize 0
                transient_lastDdlTime 1664313044
              serde: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
              name: mydb.foodplaces
```

Step 20 : Use a join operation between the two tables (foodratings and foodplaces)

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

```
0: jdbc:hive2://localhost:10000/ (MyDb)> select fp.place, avg(fr.food4) as average
. . . . . .> from foodratings fr
. . . . . .> join foodplaces fp
. . . . . .> ON fp.id = fr.id
. . . . . .> where fp.place='Soup Bowl'
. . . . . .> group by fp.place;
INFO  : Compiling command(queryId=hive_20220927222037_f1fcbabc-9f87-4ec1-928a-830e9f0b06a5): select fp.place, avg(fr.food4) as average
from foodratings fr
join foodplaces fp
ON fp.id = fr.id
where fp.place='Soup Bowl'
group by fp.place
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:fp.place, type:string, comment:null), FieldSchema(name:average, type:double, comment:null)], properties:null)
INFO  : EXPLAIN output for queryid hive_20220927222037_f1fcbabc-9f87-4ec1-928a-830e9f0b06a5 : STAGE DEPENDENCIES:
  Stage-1 is a root stage [MAPRED]
  Stage-0 depends on stages: Stage-1 [FETCH]

STAGE PLANS:
  Stage: Stage-1
    Tez
      DagId: hive_20220927222037_f1fcbabc-9f87-4ec1-928a-830e9f0b06a5:6
      Edges:
        Map 1 <- Map 3 (BROADCAST_EDGE)
        Reducer 2 <- Map 1 (SIMPLE_EDGE)
      DagName:
      Vertices:
        Map 1
            Map Operator Tree:
                TableScan
                  alias: fr
                  Statistics: Num rows: 2184 Data size: 17473 Basic stats: COMPLETE Column stats: NONE

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      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: 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_20220927222037_f1fcbabc-9f87-4ec1-928a-830e9f0b06a5); Time taken: 12.765 seconds
INFO  : OK
+------------+---------------------+
|  fp.place  |       average       |
+------------+---------------------+
| Soup Bowl  | 23.858585858585858  |
+------------+---------------------+
1 row selected (13.178 seconds)
```

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?**

The greatest degree of analytics queries that just require a subset of columns to be analyzed over extremely huge data sets are when column-based storage is most useful. Row-based storage will be more appropriate for your needs if your queries require access to all or most of the columns of each row of data.

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

In order to process big datasets effectively, the task must typically be divided into components that can be outsourced to other processors. If a single column at a time is the focus of the query calculation, a column-based structure will be easier to split into different jobs. The row-columnar columnar formats include taking a batch of rows and storing them in columnar format. Then, these batches act as split boundaries

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

Compression uses encoding for frequently repeating data to achieve this reduction, Columnar data can achieve better compression rates than row-based data. Storing values by column, with the same type next to each other, allows you to do more efficient compression on them than if you’re storing rows of data. For example, storing all dates together in memory allows for more efficient compression than storing data of various types next to each other—such as string, number, date, string, date.

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

An analytics database for Hadoop called Apache Impala frequently uses Parquet. Wide datasets with many of columns make Parquet particularly skilled at analysis. Binary data is structured by "row group" within each Parquet file. The values of the data are arranged by column for each group of rows. The advantages of compression made possible by this are what we discussed previously. When reading a lot, Parquet is a wise choice.