Uploaded hql.zip using “scp” and unzipped it

Running TestDataGen class to create Below Details:

Magic Number=227086

Foodplaces227086.txt

Foodratings22707086.txt

```
[hadoop@ip-172-31-48-80 ~]$ ls
hql  hql.zip  __MACOSX  TestDataGen.class
[hadoop@ip-172-31-48-80 ~]$ java TestDataGen.class
Error: Could not find or load main class TestDataGen.class
[hadoop@ip-172-31-48-80 ~]$ java TestDataGen
Magic Number = 227086
[hadoop@ip-172-31-48-80 ~]$
```

```
[hadoop@ip-172-31-48-80 ~]$ ls
foodplaces227086.txt   hql      __MACOSX
foodratings227086.txt  hql.zip  TestDataGen.class
[hadoop@ip-172-31-48-80 ~]$
```

Contents of Foodratings22707086.txt

```
[hadoop@ip-172-31-48-80 ~]$ cat foodratings227086.txt
Mel,36,43,5,16,2
Mel,47,25,31,39,1
Mel,27,48,18,35,1
Sam,32,47,13,31,1
Jill,7,39,40,44,2
Joe,3,16,42,34,3
Mel,45,46,11,13,2
Jill,40,26,26,26,2
```

Contents of Foodplaces227086.txt

```
[hadoop@ip-172-31-48-80 ~]$ cat foodplaces227086.txt
1,China Bistro
2,Atlantic
3,Food Town
4,Jake's
5,Soup Bowl
```

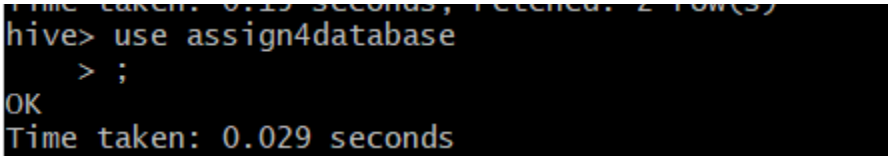
Creating Database: assign4Database

Command Used: create database assign4database;

```
hive> create database assign4Database;
OK
Time taken: 0.594 seconds
hive> show databases
    > ;
OK
assign4database
default
Time taken: 0.15 seconds, Fetched: 2 row(s)
hive>
```

Selecting DataBase: Selecting newly created database using below command

Command Used: use assign4database;



Printing Database Name: Printing Database to be sure of database working

Command Used: set hive.cli.print.current.db=true;

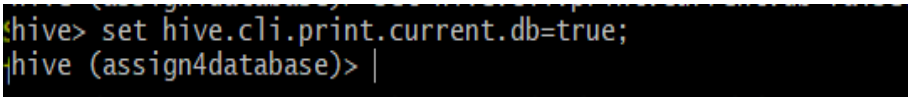
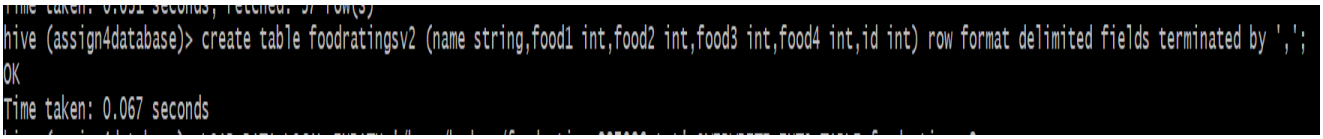
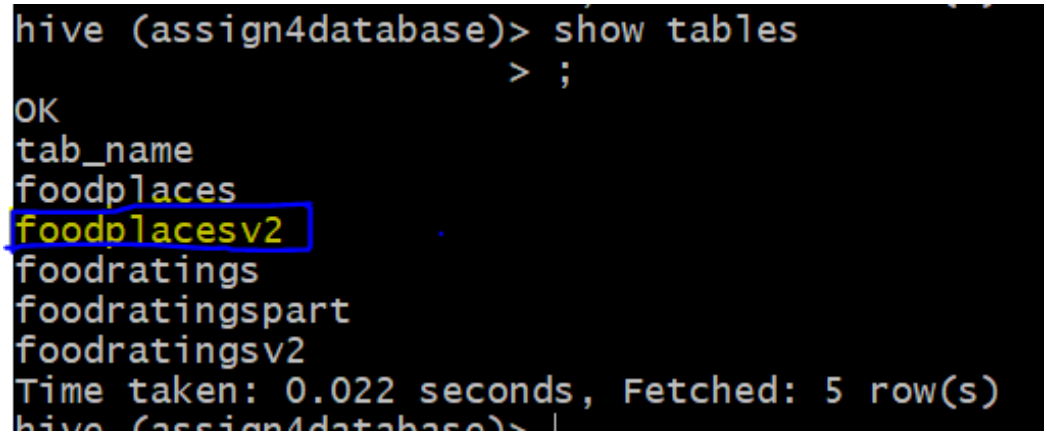


Table Creation : Created new table foodratingsv2

Command Used: CREATE TABLE <tableName> (<column1Name><Column1Datatype>,<column2Name><Column2Datatype>,….) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;





Adding Comments :Adding the comment to already created table. using the uderlying command

Command Used:ALTER TABLE foodratingsv2 CHANGE <ColumnName> <ColumnName> <ColumnDataType> COMMENT '<comments>' ;

```
hive (assign4database)> ALTER TABLE foodratingsv2 change name name string comment 'name comment';
OK
Time taken: 0.063 seconds
```

```
hive (assign4database)> ALTER TABLE foodratingsv2 change food4 food4 int comment 'Food4 comment';
OK
Time taken: 0.062 seconds
hive (assign4database)> ALTER TABLE foodratingsv2 change food3 food3 int comment 'Food3 comment';
OK
Time taken: 0.055 seconds
hive (assign4database)> ALTER TABLE foodratingsv2 change food2 food2 int comment 'Food2 comment';
OK
Time taken: 0.057 seconds
hive (assign4database)> ALTER TABLE foodratingsv2 change food1 food1 int comment 'Food1 comment';
OK
Time taken: 0.062 seconds
```

Table Description: Showing the table description using below command

Command used: describe formatted foodratingsv2;

```
hive (assign4database)> describe formatted foodratingsv2;
OK
col_name        data_type       comment
# col_name              data_type               comment

name                    string                  name comment
food1                   int                     Food1 comment
food2                   int                     Food2 comment
food3                   int                     Food3 comment
food4                   int                     Food4 comment
id                      int

# Detailed Table Information
Database:               assign4database
Owner:                  hadoop
CreateTime:             Sun Sep 20 23:37:53 UTC 2020
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://ip-172-31-48-80.ec2.internal:8020/user/hive/warehouse/assign4database.db/foodratingsv2
Table Type:             MANAGED_TABLE
Table Parameters:
        last_modified_by        hadoop
        last_modified_time      1600656182
        numFiles                1
        numRows                 0
        rawDataSize             0
        totalSize               17464
        transient_lastDdlTime   1600656182

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.028 seconds, Fetched: 37 row(s)
```

```
hive (assign4database)> create table foodplacesv2 (id int,place string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
```

Adding Comments :Adding the comment to already created table. using the uderlying command

Command Used:ALTER TABLE foodplacesv2 CHANGE <ColumnName> <ColumnName> <ColumnDataType> COMMENT '<comments>' ;

```
hive (assign4database)> describe formatted foodplacesv2;
OK
col_name        data_type       comment
# col_name              data_type               comment


id                      int                     this is id

place                   string                  this is place


# Detailed Table Information
Database:               assign4database
Owner:                  hadoop
CreateTime:             Mon Sep 21 01:57:18 UTC 2020
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://ip-172-31-48-80.ec2.internal:8020/
user/hive/warehouse/assign4database.db/foodplacesv2
Table Type:             MANAGED_TABLE
Table Parameters:
        last_modified_by        hadoop
        last_modified_time      1600656421
        numFiles                1
        numRows                 0
        rawDataSize             0
        totalSize               59
        transient_lastDdlTime   1600656421

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySi
mpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKe
yTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.046 seconds, Fetched: 33 row(s)
hive (assign4database)>
```

2) Magic Number=227086

Load Data: Loading Data from foodratings227086.txt to foodratingsv2 using below command.

Command used: LOAD DATA LOCAL INPATH '/home/Hadoop/foodratings227086.txt' OVERWRITE INTO TABLE foosratingsv2 ;

```
Time taken: 0.067 seconds
hive (assign4database)> LOAD DATA LOCAL INPATH '/home/hadoop/foodratings227086.txt' OVERWRITE INTO TABLE foodratingsv2;
Loading data to table assign4database.foodratingsv2
OK
Time taken: 0.894 seconds
```

Printing Values: Printing Min, Max, Avg of column food3 using below command.

Command Used: select min(food3) as min, max(food3) as max, avg(food3) as avg from foodratingsv2;

```
Time taken: 0.344 seconds, Fetched: 1 row(s)
hive (assign4database)> select min(food3) as min,max(food3) as max,avg(food3) as avg from foodratingsv2;
Query ID = hadoop_20200920234354_cc61c1b6-b80e-4c07-8c0d-2032cf2624e7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1600630486906_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.57 s
--------------------------------------------------------------------------------
OK
min     max     avg
1       50      25.903
Time taken: 6.119 seconds, Fetched: 1 row(s)
```

3) Magic Number=227086

Printing Values: Printing Min, Max, Avg of column food3 grouped by name using below command.

Command Used: select name, min(food1) as min, max(food1) as max, avg(food1) as avg from foodratingsv2 group by name;

```
hive (assign4database)> select name, min(food1) as min,max(food1) as max,avg(food1) as avg from foodratingsv2 group by name;
Query ID = hadoop_20200920235051_30c6df15-d9de-46c1-bbc3-b3c32abfed27
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1600630486906_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.40 s
--------------------------------------------------------------------------------
OK
name    min     max     avg
Jill    1       50      26.54696132596685
Joe     1       50      25.68
Joy     1       50      23.175675675675677
Mel     1       50      26.014492753623188
Sam     1       50      26.905263157894737
Time taken: 7.232 seconds, Fetched: 5 row(s)
```

4)

Creating Table: creating table using below command.

Command Used: CREATE TABLE <tableName> (<column1Name><Column1Datatype>,<column2Name><Column2Datatype>,….) PARTIONED BY (<columnName><ColumnDatatype>) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’ ;

```
hive (assign4database)> create table foodratingspart (food1 int,food2 int,food3
int,food4 int,id int) partitioned by (name string) ROW FORMAT DELIMITED FIELDS T
ERMINATED BY ',';
OK
Time taken: 0.07 seconds
```

```
hive (assign4database)> describe formatted foodratingspart
                      > ;
OK
col_name        data_type       comment
# col_name              data_type               comment

food1                   int
food2                   int
food3                   int
food4                   int
id                      int

# Partition Information
# col_name              data_type               comment

name                    string

# Detailed Table Information
Database:               assign4database
Owner:                  hadoop
CreateTime:             Mon Sep 21 00:33:27 UTC 2020
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://ip-172-31-48-80.ec2.internal:8020/user/hive/wareh
ouse/assign4database.db/foodratingspart
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        numFiles                0
        numPartitions           0
        numRows                 0
        rawDataSize             0
        totalSize               0
        transient_lastDdlTime   1600648407

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.063 seconds, Fetched: 41 row(s)
hive (assign4database)>
```

5)

As the number of critics is relatively small it helps in organizing data efficiently over the partition feature.

6)

Copying Data from table-> portioned table: Copying the data from normal table to the Partitioned table

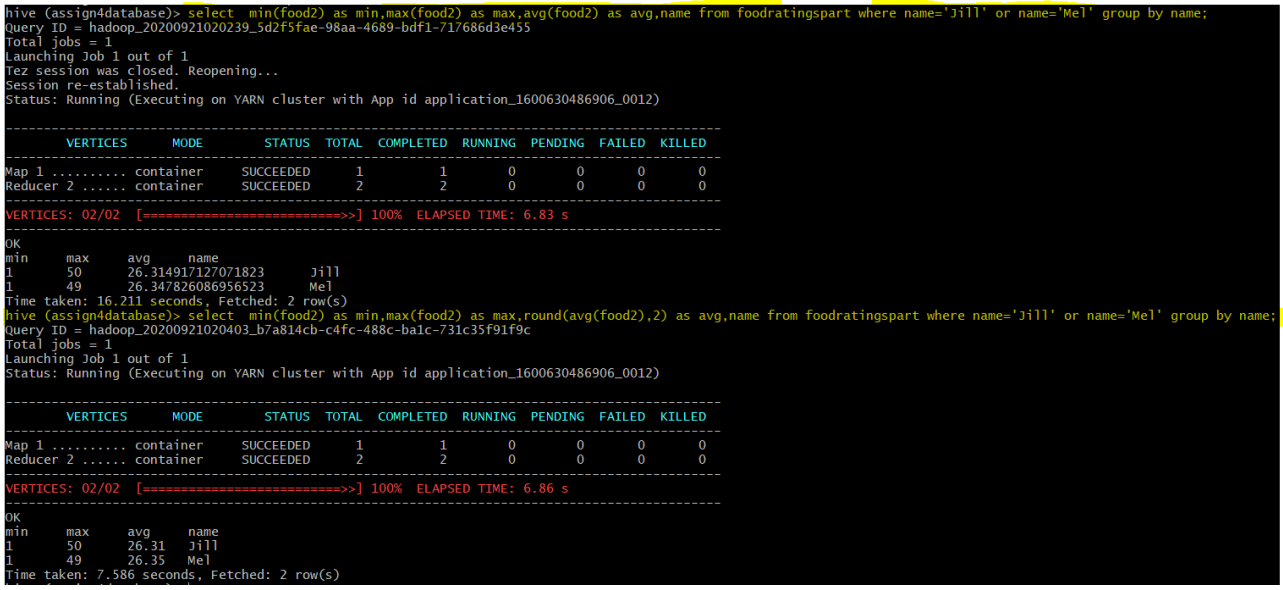
Command used: INSERT OVERWRITE TABLE foodratingspart PARTITION (name) SELECT food1,food2,food3,food4,id ,name FROM foodratingsv2 ;

```
hive (assign4database)> INSERT OVERWRITE TABLE foodratingspart PARTITION (name)
SELECT food1,food2,food3,food4,id,name FROM foodratingsv2;
Query ID = hadoop_20200921005721_1fec22ff-1479-42c5-8ec3-5abf2bde49a4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1600630486906
_0008)

Map 1: 0/1      Reducer 2: 0/2
Map 1: 0/1      Reducer 2: 0/2
Map 1: 0(+1)/1  Reducer 2: 0/2
Map 1: 0/1      Reducer 2: 0/2
Map 1: 1/1      Reducer 2: 0(+1)/2
Map 1: 1/1      Reducer 2: 1(+0)/2
Map 1: 1/1      Reducer 2: 1(+1)/2
Map 1: 1/1      Reducer 2: 2/2
Loading data to table assign4database.foodratingspart partition (name=null)


        Time taken to load dynamic partitions: 0.671 seconds
        Time taken for adding to write entity : 0.003 seconds
OK
```

```
hive (assign4database)> select * from foodratingspart;
OK
40      34      48      45      2       Jill
7       39      40      44      2       Jill
40      26      26      26      2       Jill
20      28      37      13      1       Jill
35      32      19      37      2       Jill
3       30      2       26      4       Jill
6       20      39      5       5       Jill
8       49      28      50      2       Jill
39      7       50      45      5       Jill
36      10      5       27      3       Jill
45      11      30      6       2       Jill
9       38      25      29      2       Jill
48      44      6       44      1       Jill
14      21      43      40      2       Jill
14      43      49      20      4       Jill
44      43      8       27      2       Jill
45      14      27      47      5       Jill
38      30      11      17      2       Jill
41      32      17      24      2       Jill
26      44      24      27      2       Jill
25      44      26      20      1       Jill
48      25      45      28      4       Jill
6       40      18      20      1       Jill
4       45      20      21      2       Jill
```

Calculating following Statistics: calculating the avg, Min, Max using following command

Command used: SELECT min (food2 AS min, max(food2) AS max, avg(food2) AS avg, name FROM foodratingpart WHERE name='Jill' OR name='Mel' GROUP BY name;



7)

Joining Tables and Finding Avg: Joining tables foodratingv2, foodplacesv2 and finding average value of food4 column when place is equal to 'Soup Bowl'.

Command Used: SELECT fp.place, avg(fr.food4) FROM foodplacesv2 fp JOIN foodratingsv2 fr ON (fr.id=fp.id) WHERE fp.place ='Soup Bowl' GROUP BY fp.place ;

8)

a) Row data format is chosen when your query requires to access almost all the columns in the row. Column based format is chosen when we are doing analytics query that require only some columns of the data.
b) Breaking down of the data into smaller records that can be handled independently is called splitability. It used to process large volumes of data efficiently. It usually requires breaking the job up into parts that can be farmed out to separate processors. In fact, large-scale parallelization of processing is key to performance. For example, if each file in your dataset contains one massive XML structure or JSON record, the files will not be “splittable”, i.e. decomposable into smaller records that can be handled independently.
c) Data stored in column format can achieve better compression rates than row-based data. Storing values by column, with the same type next to each other, allows the user to do more efficient compression on them than if you’re storing rows of data.
d) Parquet is especially adept at analyzing wide datasets with many columns. Each Parquet file contains binary data organized by “row group.” For each row group, the data values are organized by column. Parquet is a good choice for read-heavy workloads.