CS 595 - Assignment 4

Name: Jagruti Vichare
Email ID: jvichare@hawk.iit.edu
CWID: A20378092

Command: java TestDataGen

Output:

```
[maria_dev@sandbox ~]$ java TestDataGen
Magic Number = 177653
```

Exercise 1

Step 1:
CREATE DATABASE mydb;

Step 2:
```
CREATE DATABASE IF NOT EXISTS mydb;
use mydb;
DROP TABLE IF EXISTS foodratings;
CREATE TABLE IF NOT EXISTS mydb.foodratings (
name STRING COMMENT 'Food critic name',
food1 INT COMMENT 'Food1 rating',
food2 INT COMMENT 'Food2 rating',
food3 INT COMMENT 'Food3 rating',
food4 INT COMMENT 'Food4 rating',
id INT COMMENT 'Restaurant ID')
COMMENT 'Description of the table foodratings'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

Step 3:
DESCRIBE FORMATTED MyDb.foodratings

Output:

```
OK
col_name        data_type       comment
# col_name              data_type               comment

name                    string                  Food critic name
food1                   int                     Food1 rating
food2                   int                     Food2 rating
food3                   int                     Food3 rating
food4                   int                     Food4 rating
id                      int                     Restaurant ID

# Detailed Table Information
Database:               mydb
Owner:                  maria_dev
CreateTime:             Sat Feb 03 21:42:05 UTC 2018
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/mydb.db/foodrating
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        comment                 Description of the table foodratings
        numFiles                0
        numRows                 0
        rawDataSize             0
        totalSize               0
        transient_lastDdlTime   1517694125

# 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: 0.721 seconds, Fetched: 38 row(s)
hive (mydb)>
```

<u>Step 4:</u>

```
CREATE DATABASE IF NOT EXISTS mydb;
use mydb;
DROP TABLE IF EXISTS foodplaces;
CREATE TABLE IF NOT EXISTS mydb.foodplaces (
id INT,
place STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

<u>Step 5:</u>

```
DESCRIBE FORMATTED MyDb.foodplaces
```

Output:

```
OK
col_name        data_type       comment
# col_name              data_type               comment

id                      int
place                   string

# Detailed Table Information
Database:               mydb
Owner:                  maria_dev
CreateTime:             Sat Feb 03 21:48:22 UTC 2018
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/mydb.db/foodplaces
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        numFiles                0
        numRows                 0
        rawDataSize             0
        totalSize               0
        transient_lastDdlTime   1517694502

# 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: 0.679 seconds, Fetched: 33 row(s)
hive (mydb)>
```

Exercise 2

Step 1:

LOAD DATA LOCAL INPATH './foodratings177653.txt' OVERWRITE INTO TABLE mydb.foodratings;

Step 2:

SELECT MIN(food3) AS Minimum, MAX(food3) AS Maximum, AVG(food3) AS Average from foodratings;

Magic number - 177653
Output:

```
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.


Status: Running (Executing on YARN cluster with App id application_1517685599581_0003)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 10.10 s
--------------------------------------------------------------------------------
OK
minimum maximum average
1       50      24.797
Time taken: 27.255 seconds, Fetched: 1 row(s)
hive (mydb)>
```

Exercise 3

Step 1:

SELECT name, MIN(food1) as Minimum, MAX(food1) as Maximum, AVG(food1) as Average FROM foodratings GROUP BY name;

Magic number - 177653
Output:

```
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1517685599581_0003)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 9.08 s
--------------------------------------------------------------------------------
OK
name    minimum maximum average
Jill    1       50      25.607843137254903
Joe     1       50      26.301507537688444
Joy     1       50      25.193069306930692
Mel     1       50      26.66315789473684
Sam     1       50      24.702439024390245
Time taken: 11.456 seconds, Fetched: 5 row(s)
hive (mydb)>
```

Exercise 4

Step 1:

CREATE DATABASE IF NOT EXISTS mydb;
use mydb;
DROP TABLE IF EXISTS foodratingspart;
CREATE 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;

Step 2:
DESCRIBE FORMATTED mydb.foodratingspart
Output:

```
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:               mydb
Owner:                  maria_dev
CreateTime:             Sat Feb 03 22:27:45 UTC 2018
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/mydb.db/foodratingspar
Table Type:             MANAGED_TABLE
Table Parameters:
        transient_lastDdlTime   1517696865

# 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: 0.611 seconds, Fetched: 36 row(s)
hive (mydb)>
```

Exercise 5

Step 1:

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

Step 2:

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

Step 3:

SELECT MIN(food2) as Minimum, MAX(food2) as Maximum, AVG(food2) as Average FROM foodratingspart WHERE name="Mel" OR name="Jill";

Output:

```
hive (mydb)> SELECT MIN(food2) as Minimum, MAX(food2) as Maximum, AVG(food2) as Average FROM foodratingspart WHERE name="Mel" OR name="Jil
Query ID = maria_dev_20180203225022_be9e32f0-6081-47a3-bec1-7ee65511b7a4
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1517685599581_0004)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 13.87 s
--------------------------------------------------------------------------------
OK
minimum maximum average
1       50      25.393401015228427
Time taken: 16.866 seconds, Fetched: 1 row(s)
hive (mydb)>
```

Exercise 6

Step 1:

LOAD DATA LOCAL INPATH './foodplaces177653.txt' OVERWRITE INTO TABLE mydb.foodplaces;

Step 2:

SELECT b.place, AVG(a.food4)
FROM foodratings a JOIN foodplaces b ON a.id = b.id
WHERE b.place = 'Soup Bowl'
GROUP BY b.place;

Output:

```
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.


Status: Running (Executing on YARN cluster with App id application_1517685599581_0005)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 3 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 22.14 s
--------------------------------------------------------------------------------
OK
b.place _c1
Soup Bowl       26.778761061946902
Time taken: 43.463 seconds, Fetched: 1 row(s)
hive (mydb)>
```

Exercise 7

Pig Latin: A Not-So-Foreign Language for Data Processing

Thus article describes a new language called Pig Latin deployed at Yahoo! which is a combination of declarative style of SQL and low level, procedural style of map-reduce. It is fully implemented and compiles Pig Latin expressions into a sequence of map-reduce jobs, and orchestrates the execution of these jobs on Hadoop, an open-source scalable map-reduce implementation. Pig has integrated novel debugging environment that leads to even higher productivity. It is an open-source, Apache-incubator project, and available for general use.

This article also shows comparative analysis of SQL and Pig and concludes that in Pig, it is easier for programmers to understand and control how their data processing task is executed. It also supports a flexible, fully nested data model, user-defined functions, and the ability to operate over plain input files without any schema information. It allows complex, non-atomic data types such as set, map, and tuple to occur as fields of a table.

Pig is meant for offline, ad-hoc, scan-centric workloads. This article also compares Pig against other data processing languages and systems and about its scope of improvement.