COP-6726 DATABASE SYSTEM IMPLEMENTATION

Project - 4.1 Report

Group Member Names:

K Jaswanth Reddy - (UFID: 22719671)
M Dheeraj Kumar - (UFID: 82690922)

Running the Code:

1. Source code is in the folder “a4-1test”. It contains all the tbl and respective bin files for all tables.
2. Unit Testing is done and are presented in gtest.cc
3. To run main.cc, from project root execute command “make main” and then execute “./main”
4. Google test framework has been added in the directory “googletest”. This framework is needed to run gtest unit test cases.
5. All the gtest unit test cases have been added in the “test” folder. To run test.cc, from project root execute command “make a4-1.out” and then execute “./a4-1.out [0-11]”

Steps to run Tests(test.cc):

- Execute the “**make a4-1.out**” command from the project root. Then execute “**./a4-1.out [0-11]**”.

Steps to run gtests(gtests.cc):

- Execute the “**make gtests**” command from the project root. Then execute “**./gtests.out**”.

The following are the methods and their descriptions:

Statistics:

The above-mentioned Statistics class is used to add, compute, estimate, read and write the data analysis taken from the User and output the same to the output stream. Below are the list of the functions used in the project.

1. AddRel: This function takes relation name and number of tuples as input arguments and is used to perform addition of a relation specifying name and along with the number of tuples.
2. AddAtt: This function takes relation name, number of tuples and number of distinct elements as input arguments to perform addition of anew relation by assigning the distinct elements jus the above function.
3. CopyRel: This function takes old name and new name as input arguments and copies the relation and stores it under a new name.
4. Read: This function takes the pointer the begin the reading and once read it stores in an object.
5. Write: This function takes the pointer the begin the reading and writes the read content to the file.
6. Apply: This function takes a defined structure for the parse tree and names of the relations with the number of joins to apply the operation that the statistics uses to simulate the values of the defined operations.
7. Estimate: This function takes a defined structure of parse tree and relation names along with the number of join operations to evaluate the number of tuples that would result from the defined join operations and return the same to the caller.

STD:

The global functions from the compiler are overloaded with

1. ostream&operator<< and >>: This function takes the input as an argument of the ostream input and output to insert the number of attributes of the relation and size and puts the same to the output stream and add size of the relation present in the statistics with the estimated value to the output stream
2. istream&operator << and >>: This function takes the input as an argument from the istream to read the number of attributes present in the relation and store the statistics of the relation into the object.

Below is the format used in Statistics.txt:

1. Number in the first line indicates the number of relations in the query.
2. Name of the relation is specified in the second line.
3. Total Number of the tuples in the relation is specified in the third line.
    - **a.** If there exist multiple relations, they are listed down in the same order one after the other like {number of relations, relation name, number of tuples}
4. Fourth line contains the number of attributes for consideration.
5. Attribute names and distinct values for each attribute is defined.
6. Next line, contain names of all the relations of the query we are using.
7. Number of the tuples for the obtained relation is mentioned in the last line.

Output:

The following are the screenshots of results obtained by using 1 GB TPCH data generated from tpch-dbgen after executing all the test cases using the shell script “runTestCases.sh”. Below are the screenshots of the testcase:

output41.txt

Users > jaswanth > Desktop > Project 4.1 > output41.txt

```
1
lineitem
6001215
3
l_shipmode
1
l_discount
3.66667
l_returnflag
1
1
lineitem#
857316

**********************************************************************************************************
3
nation
25
1
n_nationkey
25
customer
150000
2
c_nationkey
25
c_custkey
150000
orders
1500000
1
o_custkey
150000
1
orders#customer#nation#
1.5e+06

**********************************************************************************************************
3
lineitem
6001215
1
l_orderkey
1.5e+06
orders
1500000
3
o_custkey
150000
o_orderkey
1.5e+06
o_orderdate
```

output41.txt

Users > jaswanth > Desktop > Project 4.1 > output41.txt

```
o_orderkey
1.5e+06
o_orderdate
-0.333333
customer
150000
2
c_mktsegment
1
c_custkey
150000
1
customer#orders#lineitem#
400081

**********************************************************************************************************
4
nation
25
1
n_nationkey
25
lineitem
6001215
1
l_orderkey
1.5e+06
orders
1500000
3
o_custkey
150000
o_orderkey
1.5e+06
o_orderdate
-0.333333
customer
150000
2
c_nationkey
25
c_custkey
150000
1
customer#orders#lineitem#nation#
2.0004e+06

**********************************************************************************************************
2
lineitem
6001215
3
```

output41.txt

Users > jaswanth > Desktop > Project 4.1 > output41.txt

```
2.0004e+06

**********
2
lineitem
6001215
3
l_shipmode
2
l_shipinstruct
1
l_partkey
200000
part
200000
2
p_container
2
p_partkey
200000
1
part#lineitem#
21432.9

**********

```

GTEST RESULTS:

Execute the "**make gtests**" command from the project root. Then execute "**./gtests.out**".

```
jaswanth — ubuntu@primary: ~/Home/Desktop/Project 4.1 — multipass ◂ multipass-gui.yuKNQU.command — 130×66
ubuntu@primary:~/Home/Desktop/Project 4.1$ ./gtests
[==========] Running 12 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 12 tests from STATISTICSTEST
[ RUN      ] STATISTICSTEST.READ_WRITE_TEST
Read and Write Test Started!
Read and Write Test Ended
[       OK ] STATISTICSTEST.READ_WRITE_TEST (10 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_1
Correctness Test 1 Started!
Correctness Test 1 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_1 (4 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_2
Correctness Test 2 Started!
Correctness Test 2 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_2 (7 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_3
Correctness Test 3 Started!
Correctness Test 3 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_3 (9 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_4
Correctness Test 4 Started!
n.n_regionkey not found!!
Correctness Test 4 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_4 (11 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_5
Correctness Test 5 Started!
Correctness Test 5 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_5 (5 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_6
Correctness Test 6 Started!
Correctness Test 6 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_6 (9 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_7
Correctness Test 7 Started!
Correctness Test 7 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_7 (7 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_8
Correctness Test 8 Started!
Correctness Test 8 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_8 (7 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_9
Correctness Test 9 Started!
Correctness Test 9 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_9 (7 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_10
Correctness Test 10 Started!
Correctness Test 10 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_10 (6 ms)
[ RUN      ] STATISTICSTEST.CORRECTNESS_TEST_11
Correctness Test 11 Started!
Correctness Test 11 Ended!
[       OK ] STATISTICSTEST.CORRECTNESS_TEST_11 (5 ms)
[----------] 12 tests from STATISTICSTEST (93 ms total)

[----------] Global test environment tear-down
[==========] 12 tests from 1 test suite ran. (95 ms total)
[  PASSED  ] 12 tests.
```

Result: All the test cases are Passed.

Conclusion: We have successfully implemented a class for *Statistics* used for query optimization and to decide among the query plans.