PROJECT 4 – PART 1 REPORT: Statistical Estimations

GROUP MEMBERS:

1. SHREYA DASGUPTA (UFID - 47016900)
2. YASH PATTARKINE (UFID - 28616005)

INSTRUCTIONS TO COMPILE THE CODE:

1. Unzip "shreyaDasgupta1_yashPattarkine2_p41.zip". Open terminal on the machine and go to the folder where the unzipped files are. Use command -- cd<pathname/foldername>
2. Command -- make clean.
3. Compile test.cc. Use command -- make test.out
4. Run test.out. For each of the queries, give the desired number. Use command -- ./test.out [0-11]. The output will be written to statistics.txt.
5. After compiling test.cc, run ./runTestCases.sh. The output for queries q1, q2, q5, q10 and q11 will be written to output41.txt
6. For gtests, compile it using -- make gtest.out.
7. Run gtest using -- ./gtest.out
8. For generating .bin files, compile a2test.cc by doing make a2test.cc and run it using ./a2test.cc.

BRIEF OVERVIEW OF ALL METHODS AND ATTRIBUTES OF Statistics CLASS:

In the first part of Project 4, we implement the 'Statistics' class, which acts as a container class, capable of storing statistical data about the attributes and relations of our database system and utilize these information during query-optimization for a better performance.

The methods we used in this class are as follows:

1. **AddRel()**: This method is used to add a new relation to the structure. We have used a map-data structure to keep track of all the relations in our system. The new relation is added with the number of tuples in it, as an information.

2. **AddAtt()**: This method simply adds the attribute to an already existing relation. The parameters here denote the name of the relation where the attribute is to be added, the name of the attribute itself and the total number of distinct values that the attribute has

for that relation. We have kept the provision of calling the AddAtt() method any number of times, and accordingly update our map-structure to reflect the updated data.

3. **CopyRel()**: This is used to copy one relation and rename the newly created relation with a different name, as suggested. We find the relation to be copied and then iterate through all its attributes using for loop, using a temporary relation-holder. Once the operation is finished, we create a new relation of designated name in our map-structure and put this copied relation (specified by the relation-holder) there.

4. **Write()**: We use this function to write the Statistics- object to a text file (Statistics.txt). We assume that the file path specified in the attribute to write from would not be empty. ‘Start’ and ‘End’ denote the beginning and ending of each relation, in our code. For each relation, we first print the relation name, and the number of tuples, then we print the count of designated attribute-names and the distinct numbers of each of them.

5. **Read()**: This function is the counter of the ‘Write’-method and reads back the text file to disk. In case the file to read from is non-existent we do not throw any error, instead the resulting file remains empty.

6. **Apply()**: This method is assuming what could happen if we performed a join among all the active relations in our map- data structure. It utilizes the statistical information and makes an estimate of the number of tuples and the number of distinct attributes in each of the relations that would be existing. We utilize the Estimate() – method (described below) for this. We make use of ParseTree extensively here. Also if the parse tree is null, we realize it as having no effect in the Statistics-object or results in pure cross-product of all the relations combining two or more partitions. After this method the result is realized in a single partition, and the state of Statistics-object is changed to reflect the adjusted effects of such a big join operation.

7. **Estimate()**: This method is exact replica of Apply() method with a few modifications. This also utilizes parse tree and is used in the Apply() method to come up with an estimate of the statistics resulting from joining of all of the relations. However, unlike Apply() method, we do not change the state of Statistics-object in this method, instead, we simply return the number of the tuples resulting from the joining all the relations in our database system.

8. The **copy-constructor** is used to deep-copy of the relations, and number of tuples and number of distinct attributes for each of them.

Additionally, we have added a **Class ‘ConvertToString’** with a method called ‘**Convert()**’ which converts a number to string. This method is utilized in Apply() and Estimate() methods of the Statistics -class.

FORMAT used for ‘Statistics.txt’:

For our Statistics.txt file, we have utilized the below format.

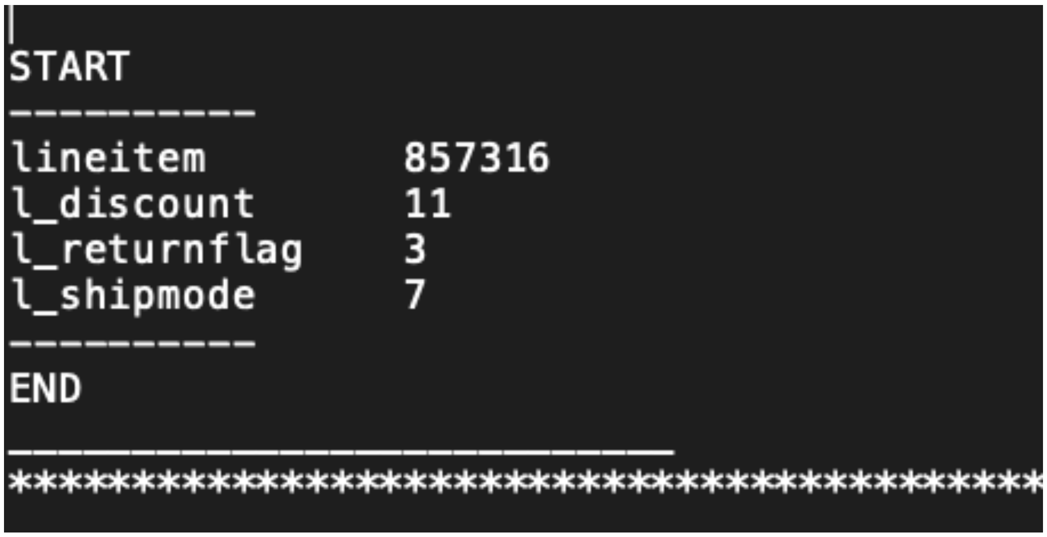
a. Each of the relations used in a specific query would be demarcated by ‘START’ to denote the beginning and ‘END’ to denote the ending.
b. In between the ‘START’ and ‘END’ demarcations, we first print the relation name and next to it the number of tuples is printed.
c. Then for each of the relation, we display the distinct attribute-names and their corresponding total count side-by-side.

A few Bug Fixes:

1. For Query-11, the word ‘Container’ is misspelt in line 541. Modified ‘p_conatiner’ to ‘p_container’.
2. For Query-5 and Query-10, the attribute ‘o_orderdate’ is added the relation named ‘Order’ using : s.AddAtt(relName[1], “o_orderdate”, 99996).
3. For Query-10, added : ‘yyparse();’ after line numbers 514 and 519.

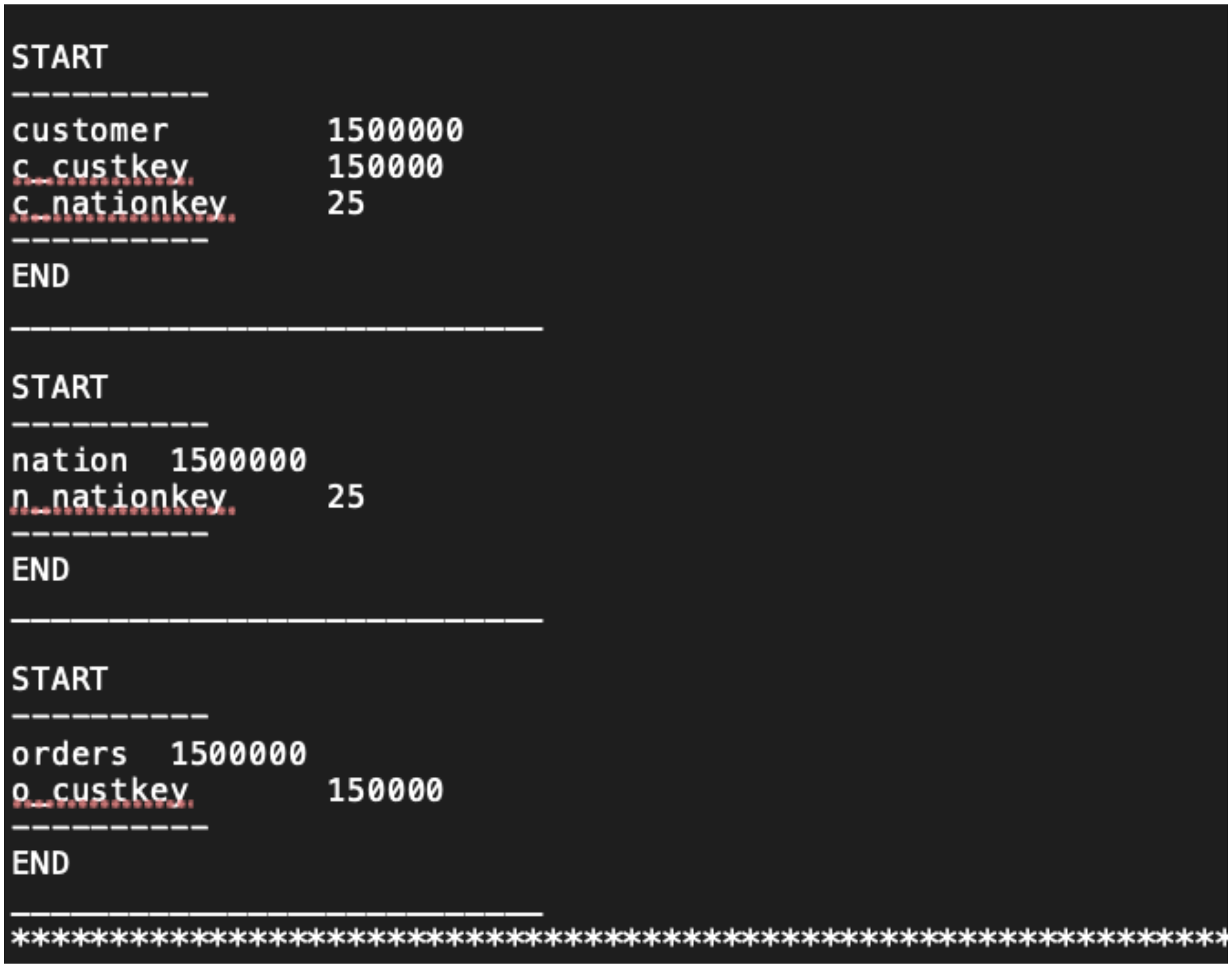
Output41.txt screen shots:

Q1:



```
|
START
----------
lineitem        857316
l_discount      11
l_returnflag    3
l_shipmode      7
----------
END
____________________________
****************************************
```

Q2:

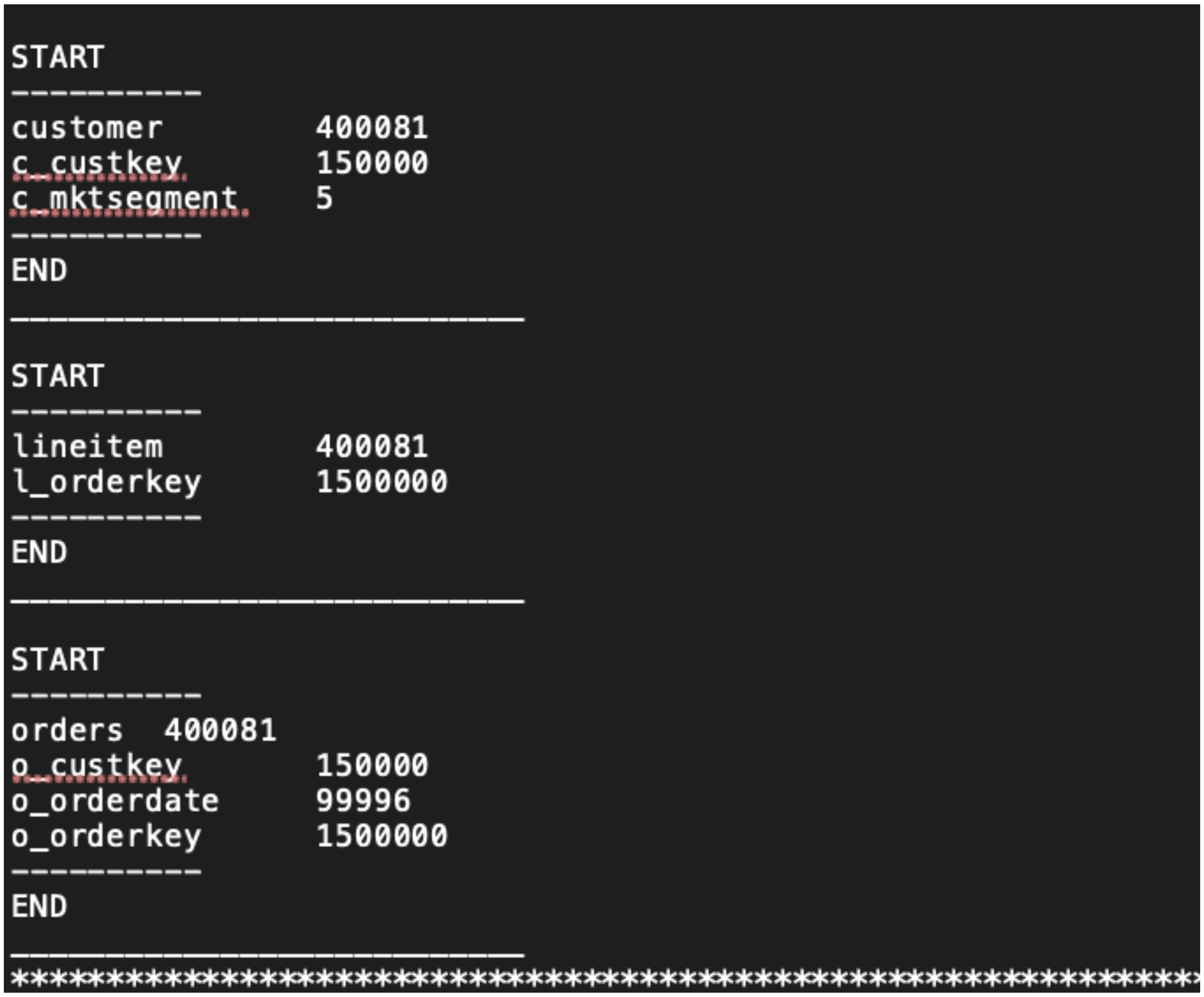


```
START
----------
customer        1500000
c_custkey       150000
c_nationkey     25
----------
END
___________________________

START
----------
nation  1500000
n_nationkey     25
----------
END
___________________________

START
----------
orders  1500000
o_custkey       150000
----------
END
___________________________
*************************************************************
```

Q5:

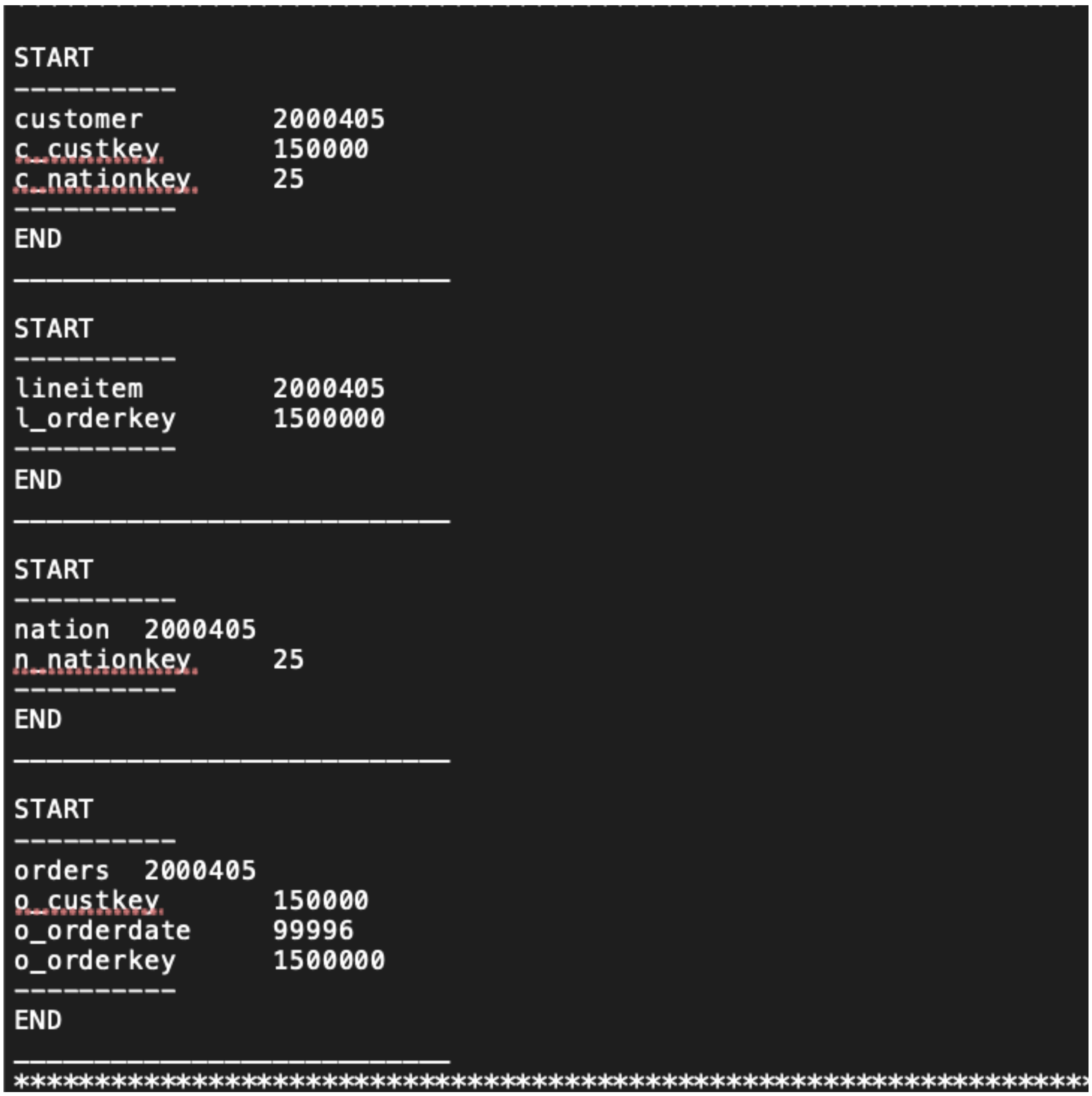


```
START
----------
customer        400081
c_custkey       150000
c_mktsegment    5
----------
END
___________________________

START
----------
lineitem        400081
l_orderkey      1500000
----------
END
___________________________

START
----------
orders  400081
o_custkey       150000
o_orderdate     99996
o_orderkey      1500000
----------
END
___________________________
*****************************************************************
```

Q10:



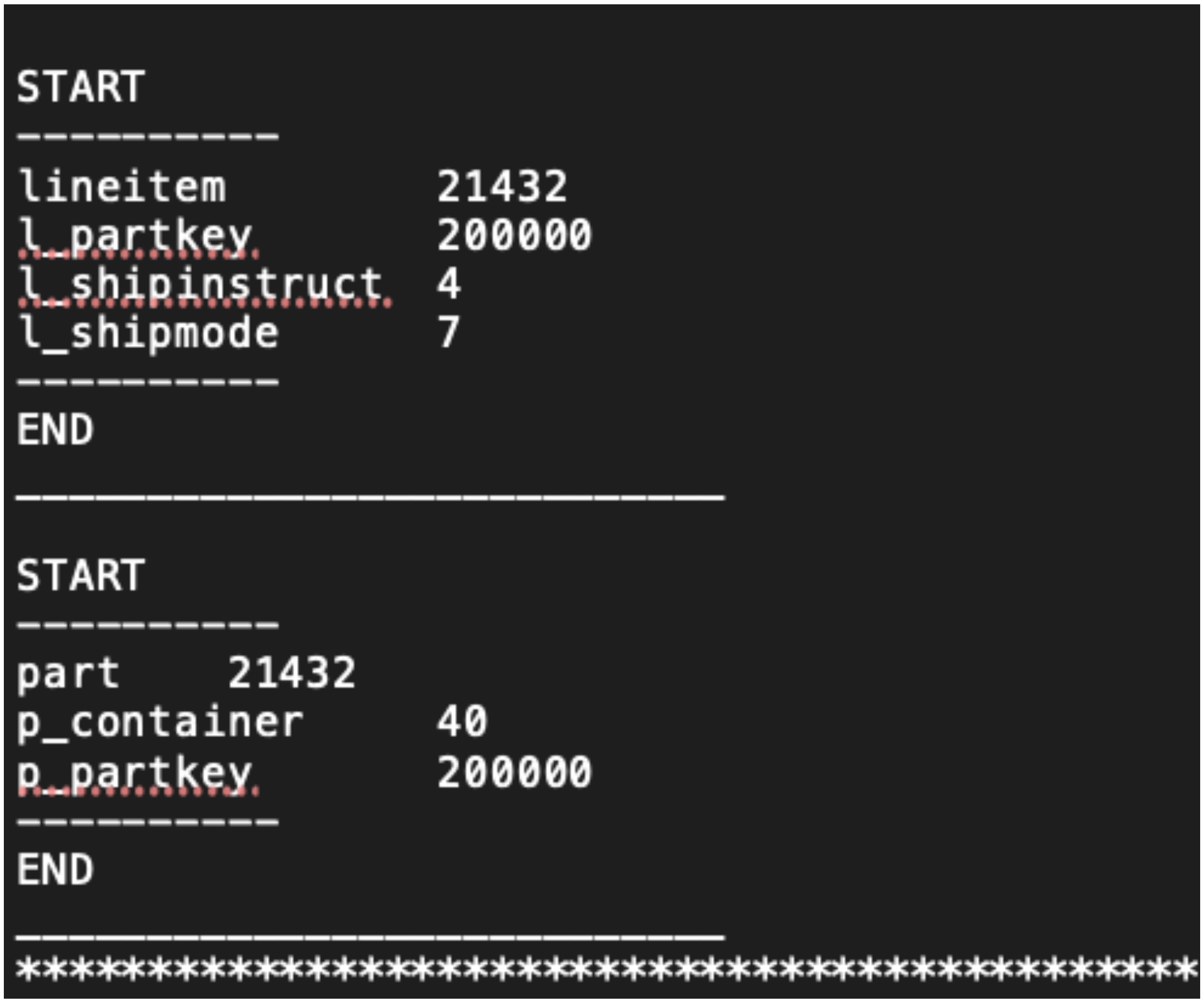
```
START
----------
customer        2000405
c_custkey       150000
c_nationkey     25
----------
END
___________________________

START
----------
lineitem        2000405
l_orderkey      1500000
----------
END
___________________________

START
----------
nation  2000405
n_nationkey     25
----------
END
___________________________

START
----------
orders  2000405
o_custkey       150000
o_orderdate     99996
o_orderkey      1500000
----------
END
___________________________
*******************************************************************
```

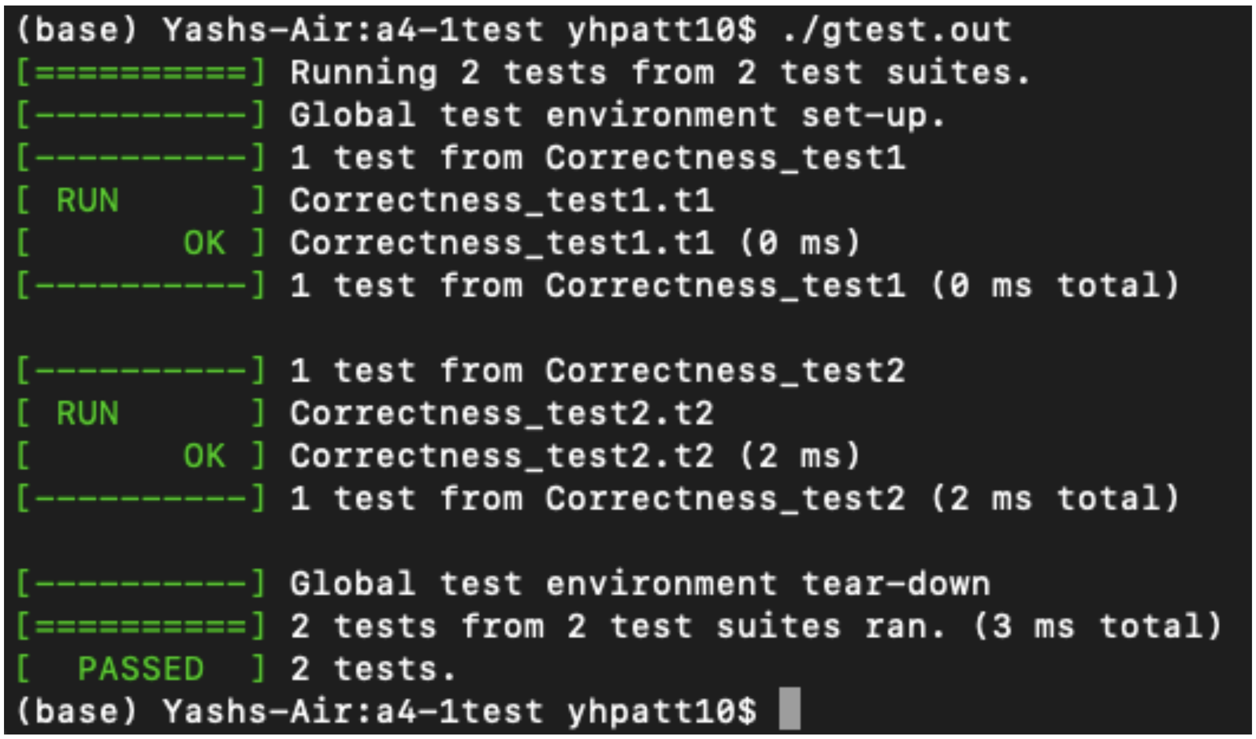
Q11:



```
START
----------
lineitem        21432
l_partkey       200000
l_shipinstruct  4
l_shipmode      7
----------
END
___________________________

START
----------
part    21432
p_container     40
p_partkey       200000
----------
END
___________________________
**********************************************
```

Gtest Results:



```
(base) Yashs-Air:a4-1test yhpatt10$ ./gtest.out
[==========] Running 2 tests from 2 test suites.
[----------] Global test environment set-up.
[----------] 1 test from Correctness_test1
[ RUN      ] Correctness_test1.t1
[       OK ] Correctness_test1.t1 (0 ms)
[----------] 1 test from Correctness_test1 (0 ms total)

[----------] 1 test from Correctness_test2
[ RUN      ] Correctness_test2.t2
[       OK ] Correctness_test2.t2 (2 ms)
[----------] 1 test from Correctness_test2 (2 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 2 test suites ran. (3 ms total)
[  PASSED  ] 2 tests.
(base) Yashs-Air:a4-1test yhpatt10$
```