Project Report: Database System Implementation (COP6726)

Project 4 Part 1: Statistical Estimation

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

I have done this project alone without any project partner. Hence, I am submitting it on my behalf.

1) **Compile and Run Instructions:** This program has been compiled, run and tested on Ubuntu 18.04 operating system. Before running the below commands please update all the paths in "**test.cat**" file. In test.cat accordingly for required bin files, catalogue file and tpch-ddbgen data.

Step.1 – To run a2test.out:
In order to load all the tables and generate the heap files we need to run **a2test.out**, we need to make sure that in a2test.h file filepaths like dbfile_dir, tpch_dir and catalog_path is updated correctly with right locations. please use below commands sequentially to run **a2test.out** and follow onscreen instruction to load the table files. Conce the a2test.out is compiled, it has to be run 8 times to load all 8 table files and generate heap files.

```
> make clean
> make a2test.out
> ./a2test.out
```

Step.2 - To run a4-1.out: In order to **a4-1.out** first we need to make sure that in "test.cat" has right path to dbfiles, tpch_dbgen files and catalog file, updated correctly with right locations. please use below commands sequentially to run **a4-1.out**.

```
> make a4-1.out
> ./a4-1.out (query number).            Ex.   ./a4-1.out   1
```

Step.3 - To run runTestCases.sh:
In order to run **runTestCases.sh** please use below commands sequentially.

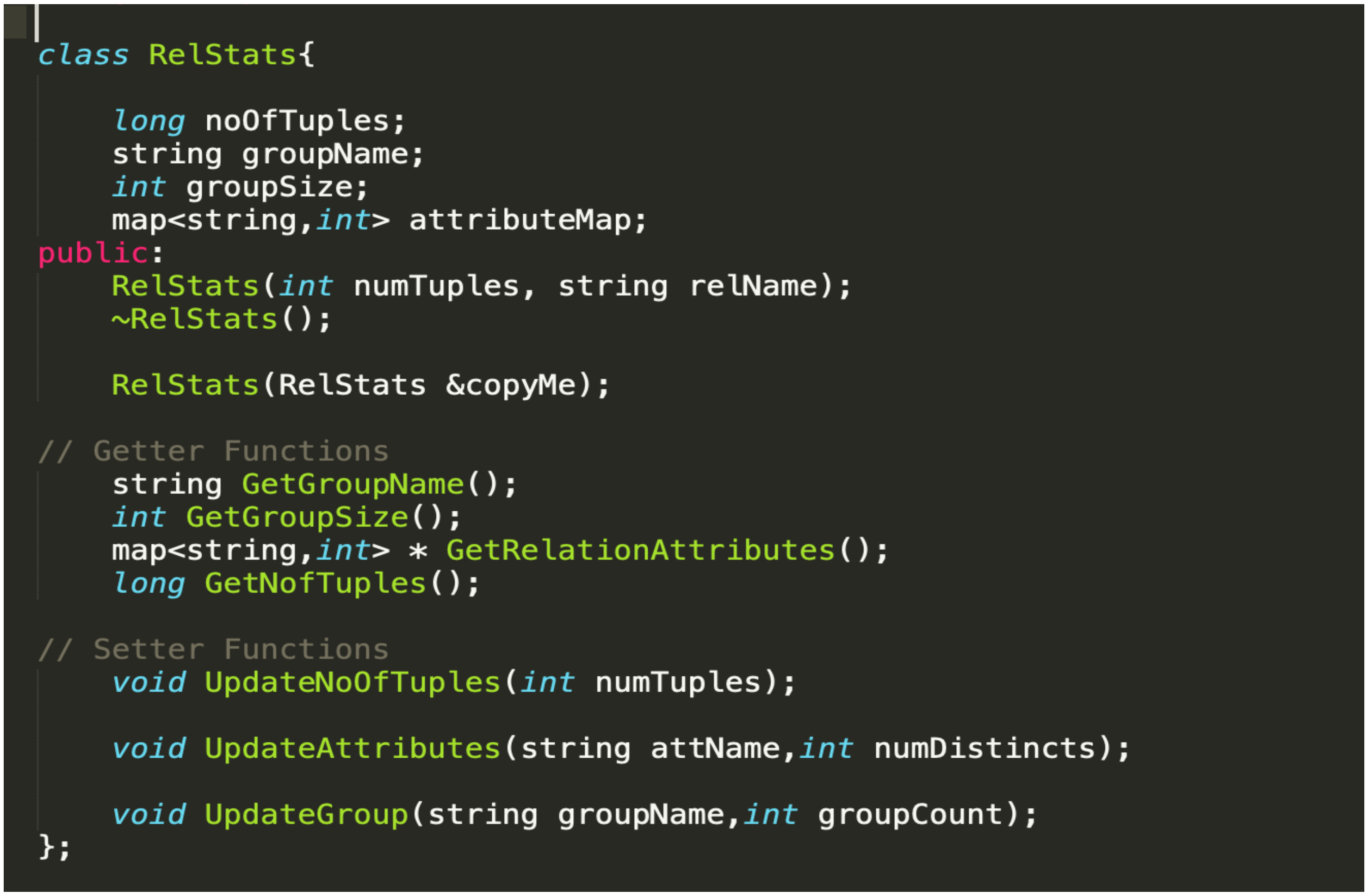
```
> make a4-1.out
> sh runTestCases.sh
```

Step.4 - To run gtest (google tests):
In order to run **gtest** please use below commands sequentially to run googletests.

```
> make gtest.out
>./gtest
```

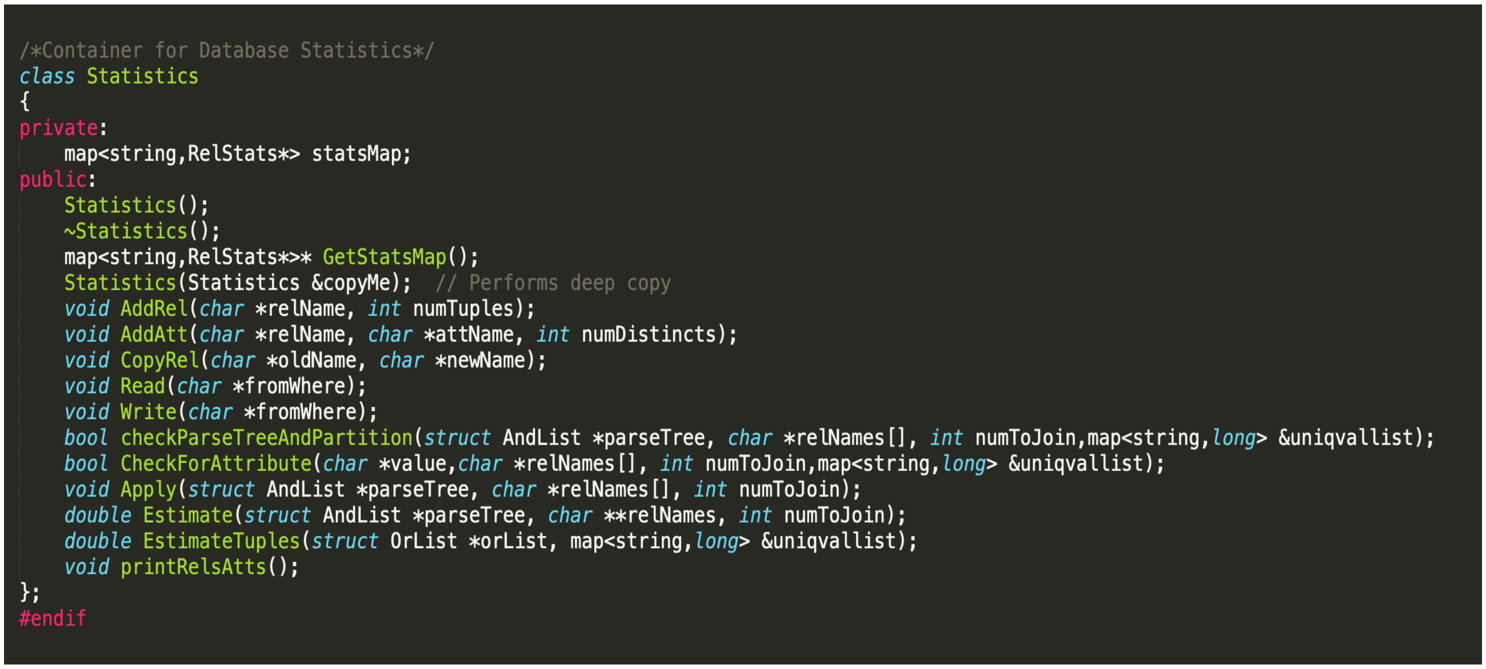
2) Project files, structure, classes and functions:

1) RelStats Class:



```
class RelStats{

    long noOfTuples;
    string groupName;
    int groupSize;
    map<string,int> attributeMap;
public:
    RelStats(int numTuples, string relName);
    ~RelStats();

    RelStats(RelStats &copyMe);

// Getter Functions
    string GetGroupName();
    int GetGroupSize();
    map<string,int> * GetRelationAttributes();
    long GetNofTuples();

// Setter Functions
    void UpdateNoOfTuples(int numTuples);

    void UpdateAttributes(string attName,int numDistincts);

    void UpdateGroup(string groupName,int groupCount);
};
```

2) Statistics Class:



```
/*Container for Database Statistics*/
class Statistics
{
private:
    map<string,RelStats*> statsMap;
public:
    Statistics();
    ~Statistics();
    map<string,RelStats*>* GetStatsMap();
    Statistics(Statistics &copyMe);  // Performs deep copy
    void AddRel(char *relName, int numTuples);
    void AddAtt(char *relName, char *attName, int numDistincts);
    void CopyRel(char *oldName, char *newName);
    void Read(char *fromWhere);
    void Write(char *fromWhere);
    bool checkParseTreeAndPartition(struct AndList *parseTree, char *relNames[], int numToJoin,map<string,long> &uniqvallist);
    bool CheckForAttribute(char *value,char *relNames[], int numToJoin,map<string,long> &uniqvallist);
    void Apply(struct AndList *parseTree, char *relNames[], int numToJoin);
    double Estimate(struct AndList *parseTree, char **relNames, int numToJoin);
    double EstimateTuples(struct OrList *orList, map<string,long> &uniqvallist);
    void printRelsAtts();
};
#endif
```

3) **Statistics.txt:** This file statistics.txt stores the data and information which is recorded by the statistics class. This file could be read using read command to load the data into the program. Also, in case we need to write data in the statistics.txt file then it could be done using write command. The used format of statistics.txt file is explained below, please consider below example.

Before Join:

1) The first line "begin" denotes that new relation is started.
2) The second line format represents the relation name, number of tuples, group name and group count simultaneously in the same order.
3) Initially before joining the group name and relation name are same but after joining the group name is changed, it becomes the comma separated name of tables which is used by join and the group count becomes the count of tables in this case its 2.
4) Then next lines contain attributes associated with the relation followed by distinct count till the END tag appears.

```
BEGIN
lineitem 21433 ,lineitem 1
l_partkey 200000
l_shipinstruct 4
l_shipmode 7
END
BEGIN
part 21433 ,part 1
p_container 40
p_partkey 200000
END
```

After Join:

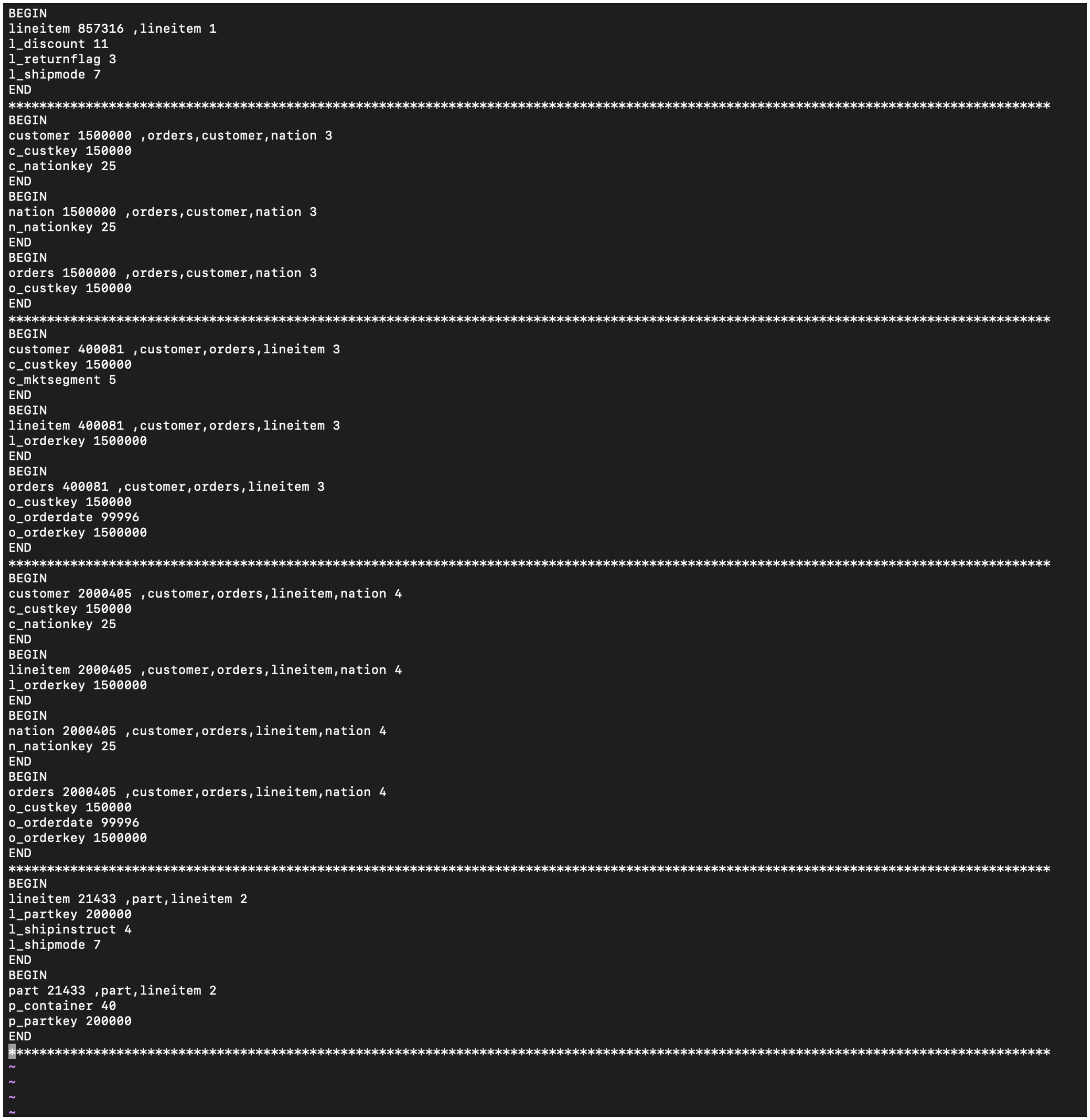
1) After join is performed then tuple count for all the relations are updated
2) After the join is performed the group name and group count for all the relations are updated as we can see after joining the group name is changed, it becomes the comma separated name of tables(part,lineitem) which is used by join and the group count becomes the count of tables in this case its 2.
3) Then next lines contain attributes associated with the relation followed by distinct count till the END tag appears.
4) The relations are not combined in one entity, the relations are stored separately with their group name and group count being the common identifying factor.

```
BEGIN
lineitem 21433 ,part,lineitem 2
l_partkey 200000
l_shipinstruct 4
l_shipmode 7
END
BEGIN
part 21433 ,part,lineitem 2
```

p_container 40
p_partkey 200000
END

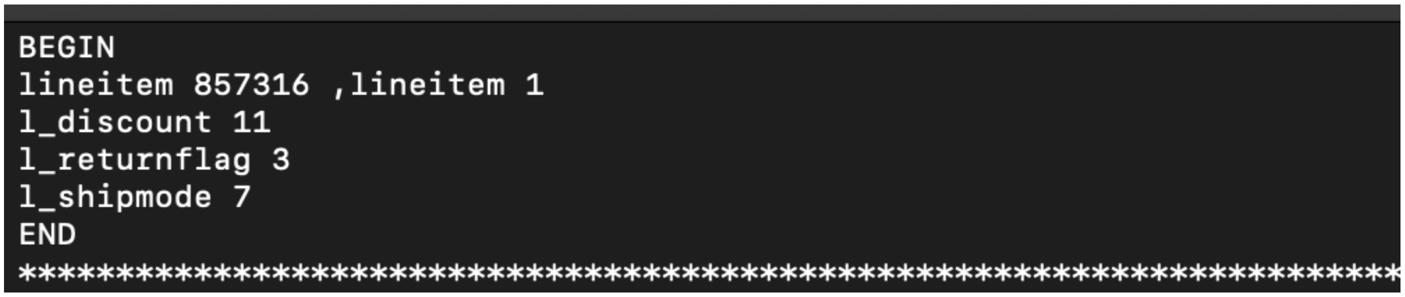
3) Program Run Results: All the screenshots are added in screenshots folder of this submission**.**

a) Results of "runTestCases.sh", Screenshot of Output41.txt:



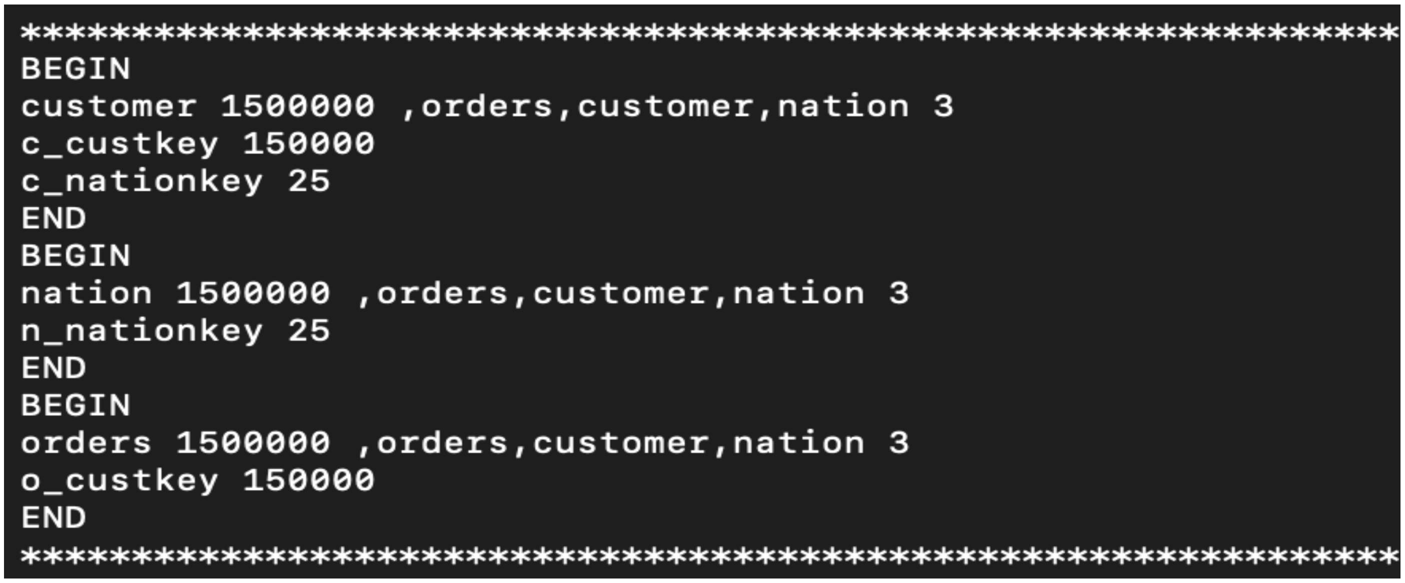
```
BEGIN
lineitem 857316 ,lineitem 1
l_discount 11
l_returnflag 3
l_shipmode 7
END
***********************************************************************************************************************************************
BEGIN
customer 1500000 ,orders,customer,nation 3
c_custkey 150000
c_nationkey 25
END
BEGIN
nation 1500000 ,orders,customer,nation 3
n_nationkey 25
END
BEGIN
orders 1500000 ,orders,customer,nation 3
o_custkey 150000
END
***********************************************************************************************************************************************
BEGIN
customer 400081 ,customer,orders,lineitem 3
c_custkey 150000
c_mktsegment 5
END
BEGIN
lineitem 400081 ,customer,orders,lineitem 3
l_orderkey 1500000
END
BEGIN
orders 400081 ,customer,orders,lineitem 3
o_custkey 150000
o_orderdate 99996
o_orderkey 1500000
END
***********************************************************************************************************************************************
BEGIN
customer 2000405 ,customer,orders,lineitem,nation 4
c_custkey 150000
c_nationkey 25
END
BEGIN
lineitem 2000405 ,customer,orders,lineitem,nation 4
l_orderkey 1500000
END
BEGIN
nation 2000405 ,customer,orders,lineitem,nation 4
n_nationkey 25
END
BEGIN
orders 2000405 ,customer,orders,lineitem,nation 4
o_custkey 150000
o_orderdate 99996
o_orderkey 1500000
END
***********************************************************************************************************************************************
BEGIN
lineitem 21433 ,part,lineitem 2
l_partkey 200000
l_shipinstruct 4
l_shipmode 7
END
BEGIN
part 21433 ,part,lineitem 2
p_container 40
p_partkey 200000
END
***********************************************************************************************************************************************
~
~
~
~
```

Query 1: Output Screenshot:



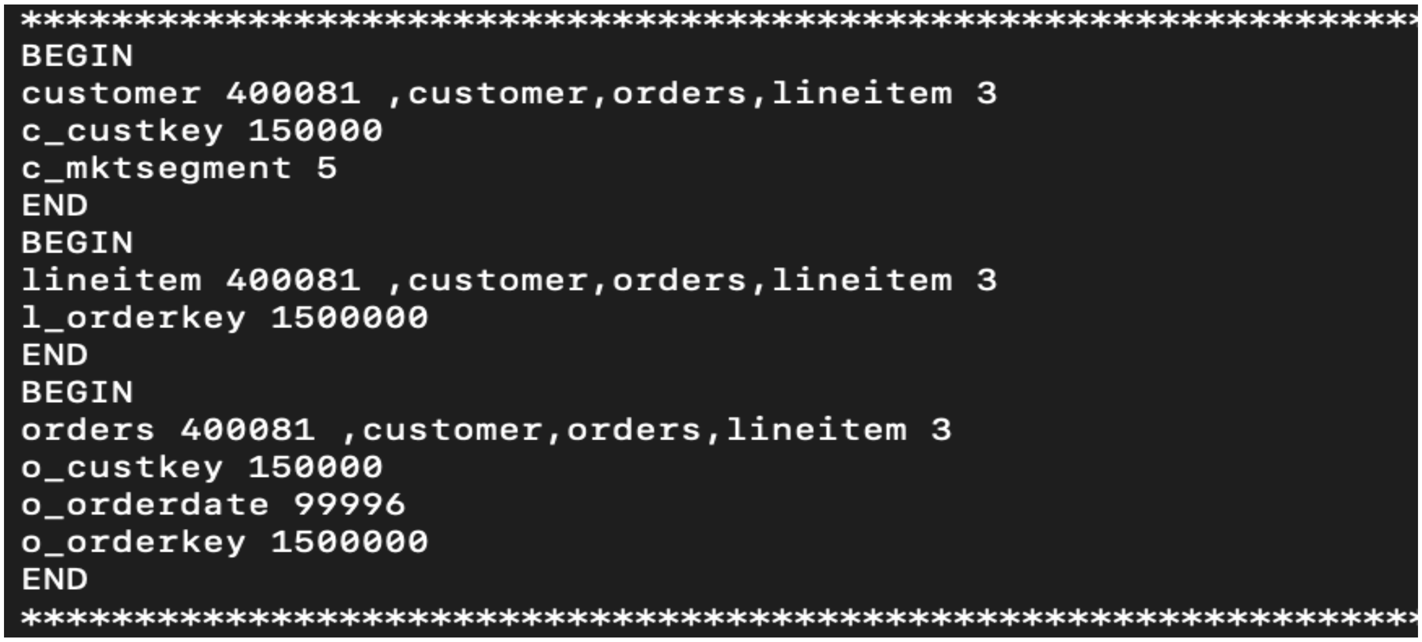
```
BEGIN
lineitem 857316 ,lineitem 1
l_discount 11
l_returnflag 3
l_shipmode 7
END
************************************************************************
```

Query 2: Output Screenshot:



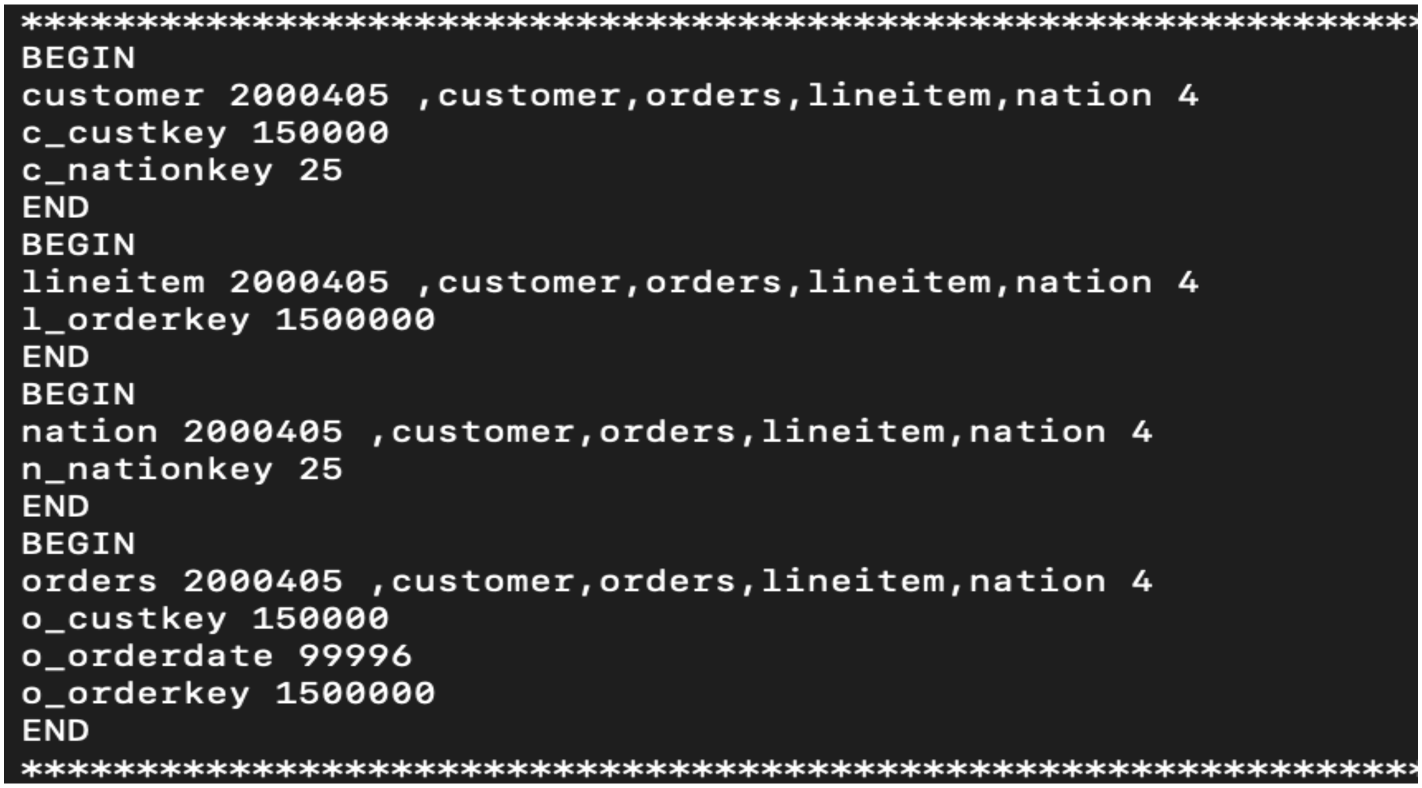
```
*****************************************************************
BEGIN
customer 1500000 ,orders,customer,nation 3
c_custkey 150000
c_nationkey 25
END
BEGIN
nation 1500000 ,orders,customer,nation 3
n_nationkey 25
END
BEGIN
orders 1500000 ,orders,customer,nation 3
o_custkey 150000
END
*****************************************************************
```

Query 5: Output Screenshot:



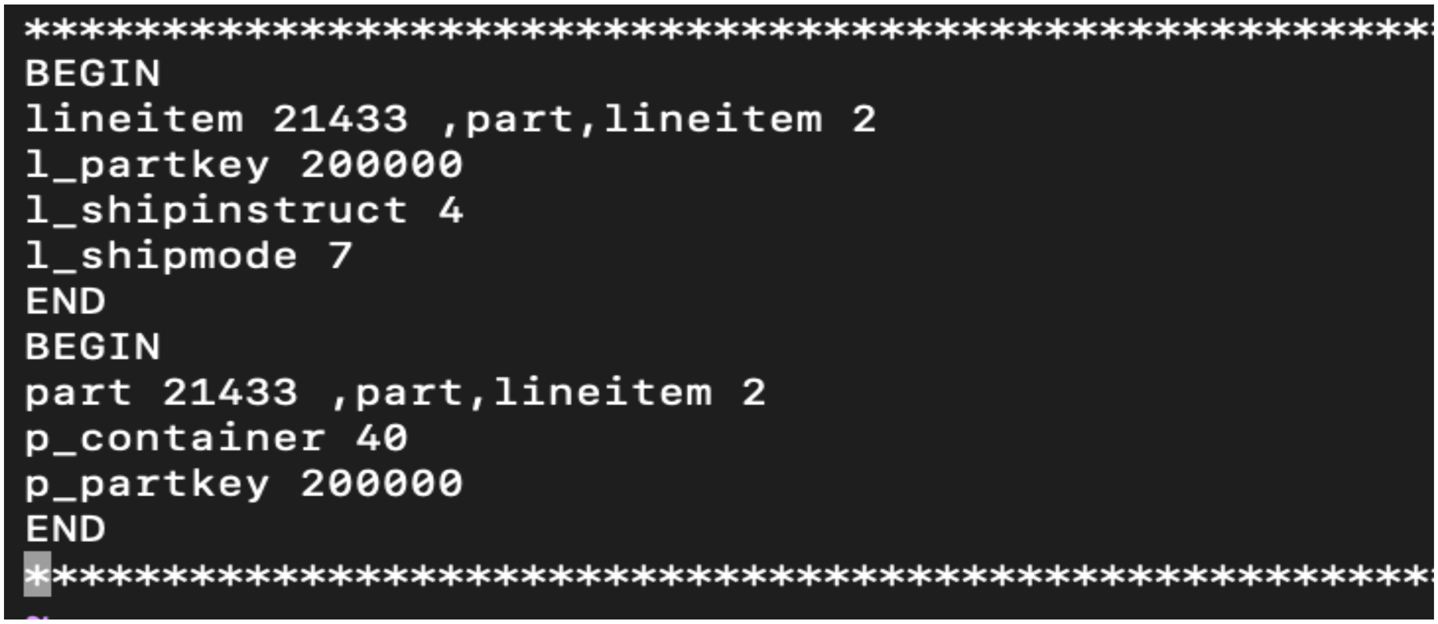
```
*****************************************************************
BEGIN
customer 400081 ,customer,orders,lineitem 3
c_custkey 150000
c_mktsegment 5
END
BEGIN
lineitem 400081 ,customer,orders,lineitem 3
l_orderkey 1500000
END
BEGIN
orders 400081 ,customer,orders,lineitem 3
o_custkey 150000
o_orderdate 99996
o_orderkey 1500000
END
*****************************************************************
```

Query 10: Output Screenshot:



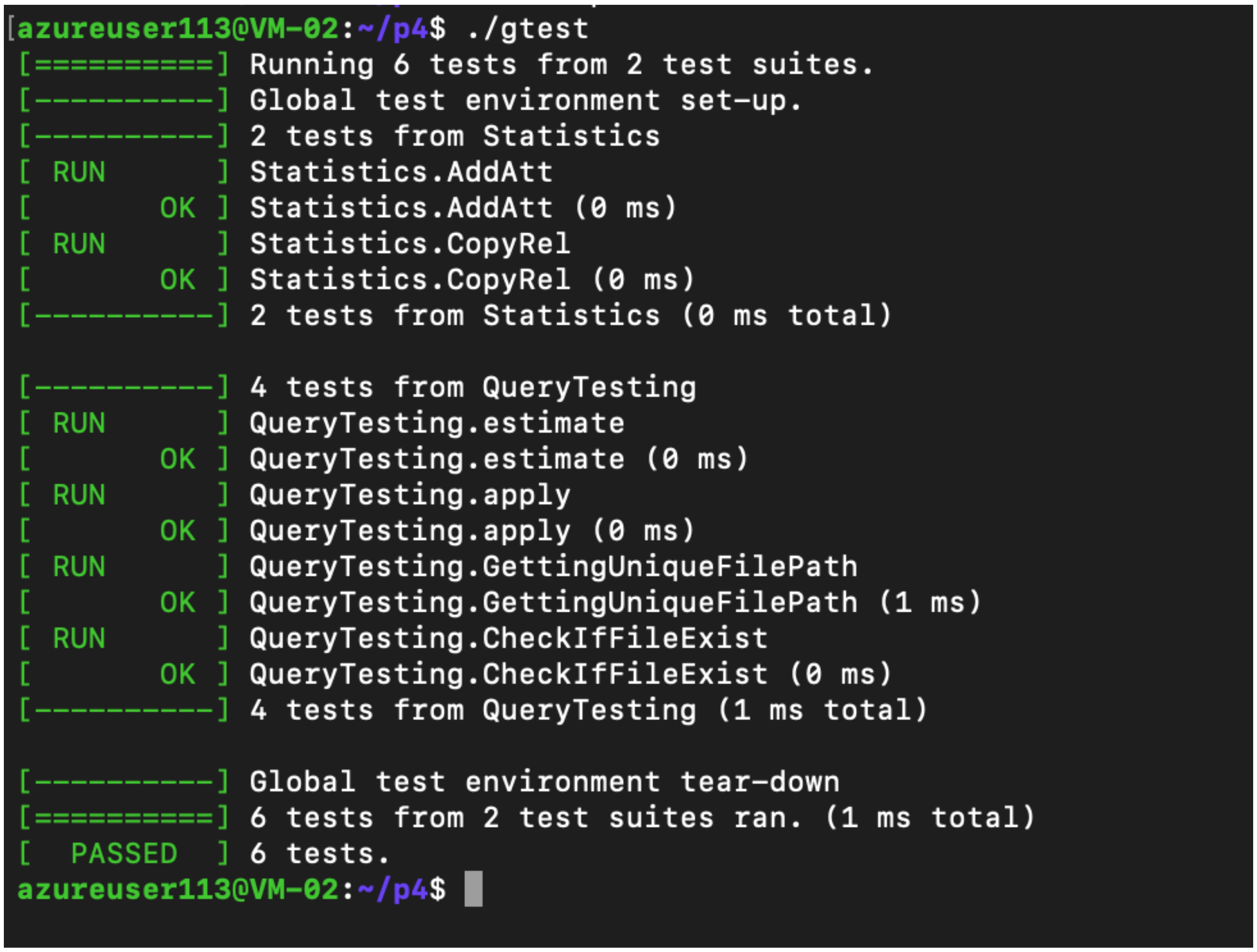
```
*************************************************************
BEGIN
customer 2000405 ,customer,orders,lineitem,nation 4
c_custkey 150000
c_nationkey 25
END
BEGIN
lineitem 2000405 ,customer,orders,lineitem,nation 4
l_orderkey 1500000
END
BEGIN
nation 2000405 ,customer,orders,lineitem,nation 4
n_nationkey 25
END
BEGIN
orders 2000405 ,customer,orders,lineitem,nation 4
o_custkey 150000
o_orderdate 99996
o_orderkey 1500000
END
*************************************************************
```

Query 11: Output Screenshot:



```
*****************************************************
BEGIN
lineitem 21433 ,part,lineitem 2
l_partkey 200000
l_shipinstruct 4
l_shipmode 7
END
BEGIN
part 21433 ,part,lineitem 2
p_container 40
p_partkey 200000
END
*****************************************************
```

b) Gtest Results: Wrote total 6 gtest cases. Below is the screenshot of output.



```
[azureuser113@VM-02:~/p4$ ./gtest
 [==========] Running 6 tests from 2 test suites.
 [----------] Global test environment set-up.
 [----------] 2 tests from Statistics
 [ RUN      ] Statistics.AddAtt
 [       OK ] Statistics.AddAtt (0 ms)
 [ RUN      ] Statistics.CopyRel
 [       OK ] Statistics.CopyRel (0 ms)
 [----------] 2 tests from Statistics (0 ms total)

 [----------] 4 tests from QueryTesting
 [ RUN      ] QueryTesting.estimate
 [       OK ] QueryTesting.estimate (0 ms)
 [ RUN      ] QueryTesting.apply
 [       OK ] QueryTesting.apply (0 ms)
 [ RUN      ] QueryTesting.GettingUniqueFilePath
 [       OK ] QueryTesting.GettingUniqueFilePath (1 ms)
 [ RUN      ] QueryTesting.CheckIfFileExist
 [       OK ] QueryTesting.CheckIfFileExist (0 ms)
 [----------] 4 tests from QueryTesting (1 ms total)

 [----------] Global test environment tear-down
 [==========] 6 tests from 2 test suites ran. (1 ms total)
 [  PASSED  ] 6 tests.
 azureuser113@VM-02:~/p4$
```

4) Observed Bugs:

1) Number of attributes for Merging and Projecting of the Record in the Join and GroupBy queries are not provided which can be passed as a parameter or can be calculated by the following formula. Offset to first attribute- sizeof(int)/sizeof(int)where Offset to first attribute is present in the record.