Assignment 4, Part 1:
Statistical Estimation

Group Members:

1. Dhiraj Mahesh Paryani, UFID: 1692 1261
2. Yogesh Laxman, UFID: 9451 2517

Instructions for compiling and running the code:

1. **Extracting folders:**
   a. Extract the contents of the folder. DhirajMaheshParyani_YogeshLaxman_p41.zip
   b. Open the terminal and navigate to the "Project" folder which is inside the extracted folder.

2. **Running the tests:**
   a. Run "make a4-1.out".
   b. To run individual test case, run "./a4-1.out {x}". Where {x} should be replaced by test case number i.e. any number from 1 to 11.
   c. Run "./runTestCases.sh" to run test cases from 1, 2, 5, 10 & 11, this will generate output in a file named "output41.txt".

Brief explanation of implementation:

This assignment deals with the implementation of Statistical Estimation. In this, we have developed the Statistics class from scratch which will be used by the query optimizer to choose a plan from different plans to execute the query.

Statistics class is developed from scratch to support Statistical estimation. This class mainly has several methods to store statistics of relations and attributes of the project. Which will be then used by Apply or Estimate method to do estimations.

Data members:

1. Name: groupNameToRelationMap
   Type: unordered_map<string, Relation>
   Description: This map is used to store relations corresponding to the group name.

2. Name: attNameToAttributeMap
   Type: unordered_map<string, Attribute>
   Description: This map stores the attributes corresponding to the attribute name.

3. Name: groupNameToSetOfRelationsMap
   Type: unordered_map<string, unordered_set<string> >
   Description: This map stores the relations corresponding to the group name.

4. Name: relNameToGroupNameMap
   Type: unordered_map<string, string>
   Description: This map gives mapping between relation name and group name.

Methods:

1. Statistics(Statistics &copyMe):
   This is the constructor method with a statistics object as the parameter. It creates a new copy of the parameter statistics object. This method simply deep copies all 4 maps of the Statistics class.

2. void AddRel(char *relName, int numTuples):
   This method adds relation into the statistics object, and if relation is already present in the object, it just updates the numTuples corresponding to that relation. While creating the new relation, it stores that relation into the new group. That is we make groups, each group containing a single relation.

3. void AddAtt(char *relName, char *attName, int numDistincts):
   This method adds an attribute into the statistics object, and if attribute is already present in the object, it just updates the numDistincts corresponding to that relation. It stores an attribute as relName.attName in "attNameToAttributeMap" map.

4. void CopyRel(char *oldName, char *newName):
   This method copies all the data of relation with oldName to create a new relation with newName. Inside which it copies all the data in 4 maps related to oldName relation to a newName relation.

5. void Read(char *fromWhere):
   This method reads the data from the "fromWhere" file and loads the current object with that data. Basically in file we store 4 maps of the Statistics class. This method reads that data from the file and loads all the 4 maps with that information.

6. void Write(char *fromWhere):
   This method writes the current object's data into the file "fromWhere". Basically it stores all the 4 maps of the current object into the file.

7. void Apply(struct AndList *parseTree, char *relNames[], int numToJoin):
   This method applies the join or selection operator and updates the current statistics object.
   It first starts by validating the current operation on relations and attributes given as the parameters. Validation is done the same as provided in the assignment requirement document.
   Then, each "and" of the parseTree condition is applied on the relations and estimates for the output number of tuples is updated in the object. Following are the rules used to estimate the number of output tuples after every end.
      i) For join:
      Output numTuples = numTuples(rel1)/numDistinct(rel1.att1)

* numTuples(rel2)/numDistinct(rel2.att1)
* min (numDistinct(rel1.att1), numDistinct(rel2.att1))

ii) Selection based on attribute value with = operation:
Output numTuples = numTuples(rel1) * 1/numDistinct(rel1.att1)

iii) Selection based on attribute value with > operation:
Output numTuples = numTuples(rel1) * ⅓

iv) Multiple "ORs" on different attributes of same table value with = operation:
Output numTuples =
numTuples(rel1) *[(1/numDistinct(rel1.att1) + 1/numDistinct(rel1.att2) + ….) - (1/numDistinct(rel1.att1)*1/numDistinct(rel1.att2) + ….)]

So, on...

8. double Estimate(struct AndList *parseTree, char **relNames, int numToJoin):
   This method applies the join or selection operator without applying updates to the current statistics object and return number of tuples in the output. This is done using creating another statistics class and calling apply method of that copied statistics class.

Statistics.txt file Format:

Information in statistics class is stored in the "Statistics.txt" file. As out Statistics class contains 4 maps. We just store that data in the file. So that the user can read this file to see the status of Statistics class. This file can be used for loading Statistics class with the data in it.
Following gives the information about format of the Statistics.txt file.

The first line contains the heading for groups i.e.
"********** Group Relations **********"
Second line gives the number of groups G. G groups follow. Each consists of one line containing two elements S and N separated by equals to (=) sign. Where S is the group name and N is the number of tuples.

After that it gives heading for attributes i.e.
"********** Attributes **********".
Next line gives the number of attributes NA. NA attributes follow. Each consists of one line containing two elements A and D separated by equals to (=) sign. Where A is the attribute name (relation name + "." + attribute name) and D is the number of distinct.

After that, it stores information about the map which gives mapping between group name and set of relations. It starts with storing heading for this i.e.
"********** GroupName to Relations **********"
Next line gives the number of mappings M. M mappings follow. Each consists of one line containing two elements G and R separated by equals to (=) sign. Where G is the group name and R is the list of relations separated by comma (",").

Finally, it stores information about the map which gives mapping between relation name and group name. It starts with storing heading for this i.e.
"********** Relation Name to Group Name **********"
Next line gives the number of mappings M. M mappings follow. Each consists of one line containing two elements R and G separated by equals to (=) character. Where R is the relation name and G is the group name.

Example: Suppose we have 4 relations and their corresponding attributes in the Statistics class. And 2 tables are already joined. This information in the Statistics.txt file will look like the following.

```
********** Group Relations **********
3
rel3&rel4=3200
rel1=100
rel2=200
********** Attributes **********
4
rel4.att=100
```

rel3.att=100
rel1.att=100
rel2.att=100
********** GroupName to Relations **********
3
rel3&rel4=rel4,rel3
rel1=rel1
rel2=rel2
********** Relation Name to Group Name **********
4
rel4=rel3&rel4
rel3=rel3&rel4
rel1=rel1
rel2=rel2

Results:

output41.txt

Query 1:

```
********** Group Relations **********
1
lineitem=857316
********** Attributes **********
3
lineitem.l_shipmode=7
lineitem.l_returnflag=3
lineitem.l_discount=11
********** GroupName to Relations **********
1
lineitem=lineitem
********** Relation Name to Group Name **********
1
lineitem=lineitem
**********
```

Query 2:

```
**********
********** Group Relations **********
1
customer&orders&nation=1.5e+06
********** Attributes **********
4
nation.n_nationkey=25
customer.c_nationkey=25
orders.o_custkey=150000
customer.c_custkey=150000
********** GroupName to Relations **********
1
customer&orders&nation=nation,orders,customer
********** Relation Name to Group Name **********
3
nation=customer&orders&nation
orders=customer&orders&nation
customer=customer&orders&nation
**********
```

Query 5:

```
**********
********** Group Relations **********
1
lineitem&customer&orders=400081
********** Attributes **********
6
lineitem.l_orderkey=1500000
orders.o_orderdate=99996
orders.o_custkey=150000
orders.o_orderkey=1500000
customer.c_custkey=150000
customer.c_mktsegment=5
********** GroupName to Relations **********
1
lineitem&customer&orders=customer,lineitem,orders
********** Relation Name to Group Name **********
3
lineitem=lineitem&customer&orders
customer=lineitem&customer&orders
orders=lineitem&customer&orders
**********
```

Query 10:

```
**********************************************************************************************************************************************************
*************** Group Relations ****************
1
lineitem&customer&orders&nation=2.00041e+06
*************** Attributes *****************
7
customer.c_nationkey=25
orders.o_custkey=150000
nation.n_nationkey=25
orders.o_orderkey=1500000
orders.o_orderdate=99996
customer.c_custkey=150000
lineitem.l_orderkey=1500000
***************** GroupName to Relations ***************
1
lineitem&customer&orders&nation=nation,orders,customer,lineitem
******************* Relation Name to Group Name ****************
4
nation=lineitem&customer&orders&nation
lineitem=lineitem&customer&orders&nation
customer=lineitem&customer&orders&nation
orders=lineitem&customer&orders&nation
**********************************************************************************************************************************************************
```

Query 11:

```
**********************************************************************************************************************************************************
*************** Group Relations ****************
1
lineitem&part=21432.9
*************** Attributes *****************
5
lineitem.l_shipmode=7
lineitem.l_shipinstruct=4
lineitem.l_partkey=200000
part.p_partkey=200000
part.p_container=40
***************** GroupName to Relations ***************
1
lineitem&part=part,lineitem
******************* Relation Name to Group Name ****************
2
lineitem=lineitem&part
part=lineitem&part
**********************************************************************************************************************************************************
```

GTests:

Instructions to run:

1. Run “make gTestStatistics.out”.
2. Run “./gTestStatistics.out”.

Details:

Following are the details of 2 written GTests:

1. TestAddRel: This tests the AddRel method of Statistics class. This first tests whether a relation is getting correctly added in the Statistics class. And then tests the same by calling the same method with a different number of tuples.

2. TestAddAtt: This tests the AddAtt method of Statistics class. This first test whether an attribute is getting correctly added in the Statistics class. And then tests the same by calling the same method with a different number of distincts.

Output:

```
g++ -O2 -Wno-deprecated -g -c gtests/StatisticsGTests.cc
g++ -O2 -Wno-deprecated -o gTestStatistics.out Statistics.o y.tab.o lex.yy.o StatisticsGTests.o -ll -lgtest
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from StatisticsGTests
[ RUN      ] StatisticsGTests.TestAddRel
[       OK ] StatisticsGTests.TestAddRel (0 ms)
[ RUN      ] StatisticsGTests.TestAddAtt
[       OK ] StatisticsGTests.TestAddAtt (0 ms)
[----------] 2 tests from StatisticsGTests (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran. (0 ms total)
[  PASSED  ] 2 tests.
dhirajmaheshparyani@Dhirajs-MBP Project %
```

Bugs:

1. **Test.cc bugs**

   i. Spelling mistake in q3 line number 201. “S_nationey” should be “s_nationkey”.

   ii. Wrong relations given as parameters in q4. Corresponding changes are done in the test.cc file.

   iii. No attribute added “o_orderdate” for relation “orders” in q5.
   Added: s.AddAtt(relName[1], "o_orderdate",99996);

   iv. No attribute added “l_receiptdate” for relation “lineitem” in q7.
   Added: s.AddAtt(relName[1], "l_receiptdate",198455);

   v. No attribute added “o_orderdate” for relation “orders” in q10.
   Added: s.AddAtt(relName[1], "o_orderdate",99996);

   vi. There should be “yyparse();” after line 514 & 519 in q10.

   vii. Spelling mistake in q11 line number 541. “p_conatiner” should be “p_container”.