Assignment 4, Part 2:
Query Compilation and Optimization

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_p412.zip
b. Open the terminal and navigate to the "Project" folder which is inside the extracted folder.

2. Update catalog path in test.cat:

Update catalog path in test.cat. Catalog path should be present in the first line.

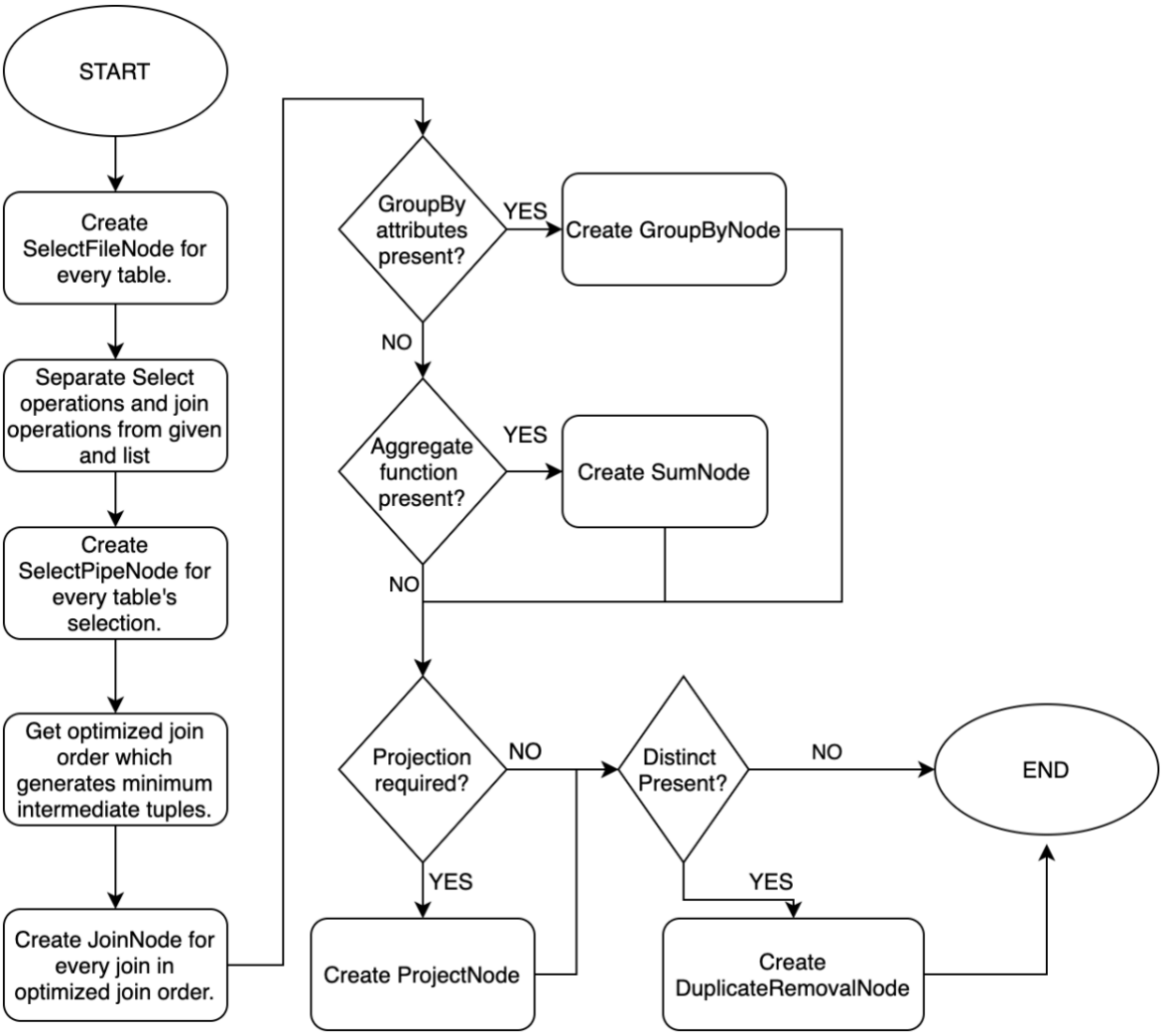
3. Running the tests:

a. Run "make a42.out".
b. Run "./a42.out". Then enter your query and press ctrl-D.
c. Run "./a42.out < {fileName}". This will take a query from file.
d. Run "./runtestcases42.sh". This will run and generate query plan for files tc1.sql, tc2.sql, tc3.sql, tc4.sql and tc5.sql and save the query plans in the file output42.txt.

Brief explanation of implementation:

This assignment generates an optimized query plan for the query. In this we have developed the QueryPlan class from scratch which is responsible for generating an optimized query plan in a tree corresponding to the query.
Tree node's class (structure) is given in QueryPlanNode. A tree node can be of seven different types i.e SelectFileNode, SelectPipeNode, ProjectNode, DuplicateRemovalNode, JoinNode, SumNode and GroupByNode. Every node type is a subclass of RelOpNode. Each node type corresponds to the corresponding relational operator.

Following flow chart gives implementation details:



Results (output42.txt):

```
TC1
PRINTING TREE POST ORDER:
 ***********
SELECT FILE operation
Output Pipe 1
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING

SELECTION CNF :
( n.n_name = UNITED STATES )

 ***********
PROJECT operation
Input Pipe 1
Output Pipe 2
Output Schema:
        Att n.n_nationkey: INT

**********
```

```
**********
TC2
PRINTING TREE POST ORDER:
 ***********
SELECT FILE operation
Output Pipe 2
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING

SELECTION CNF :
( n.n_nationkey > 5 )

 ***********
SELECT FILE operation
Output Pipe 1
Output Schema:
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING
```

```
 ***********
JOIN operation
Input Pipe 2
Input Pipe 1
Output Pipe 3
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

CNF:
( n.n_regionkey = r.r_regionkey )

 ***********
PROJECT operation
Input Pipe 3
Output Pipe 4
Output Schema:
        Att n.n_name: STRING

**********
```

```
TC3
PRINTING TREE POST ORDER:
 ***********
SELECT FILE operation
Output Pipe 2
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING

SELECTION CNF :
( n.n_name = UNITED STATES )

 ***********
SELECT FILE operation
Output Pipe 1
Output Schema:
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

 ***********
JOIN operation
Input Pipe 2
Input Pipe 1
Output Pipe 3
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

CNF:
( n.n_regionkey = r.r_regionkey )

 ***********
SUM operation
Input Pipe 3
Output Pipe 4
Output Schema:
        Att SUM: DOUBLE

FUNCTION
(n.n_nationkey)

Distinct Function: 0

****************************************
```

```
TC4
PRINTING TREE POST ORDER:
 ***********
SELECT FILE operation
Output Pipe 2
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING

SELECTION CNF :
( n.n_name = UNITED STATES )

 ***********
SELECT FILE operation
Output Pipe 1
Output Schema:
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

 ***********
JOIN operation
Input Pipe 2
Input Pipe 1
Output Pipe 3
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

CNF:
( n.n_regionkey = r.r_regionkey )

 ***********
GROUP BY operation
Input Pipe 3
Output Pipe 4
Output Schema:
        Att SUM: DOUBLE
        Att n.n_regionkey: INT

GROUPING ON
n.n_regionkey

FUNCTION
(n.n_regionkey)

Distinct Function: 0

 ***********
PROJECT operation
Input Pipe 4
Output Pipe 5
Output Schema:
        Att SUM: DOUBLE

****************************************
```

```
****************************************
TC5
PRINTING TREE POST ORDER:
 ***********
SELECT FILE operation
Output Pipe 3
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING

SELECTION CNF :
( n.n_nationkey > 10 )

 ***********
SELECT FILE operation
Output Pipe 2
Output Schema:
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

 ***********
JOIN operation
Input Pipe 3
Input Pipe 2
Output Pipe 4
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING

CNF:
( n.n_regionkey = r.r_regionkey )

 ***********
SELECT FILE operation
Output Pipe 1
Output Schema:
        Att c.c_custkey: INT
        Att c.c_name: STRING
        Att c.c_address: STRING
        Att c.c_nationkey: INT
        Att c.c_phone: STRING
        Att c.c_acctbal: DOUBLE
        Att c.c_mktsegment: STRING
        Att c.c_comment: STRING
```

```
 ***********
JOIN operation
Input Pipe 4
Input Pipe 1
Output Pipe 5
Output Schema:
        Att n.n_nationkey: INT
        Att n.n_name: STRING
        Att n.n_regionkey: INT
        Att n.n_comment: STRING
        Att r.r_regionkey: INT
        Att r.r_name: STRING
        Att r.r_comment: STRING
        Att c.c_custkey: INT
        Att c.c_name: STRING
        Att c.c_address: STRING
        Att c.c_nationkey: INT
        Att c.c_phone: STRING
        Att c.c_acctbal: DOUBLE
        Att c.c_mktsegment: STRING
        Att c.c_comment: STRING

CNF:
( n.n_nationkey = c.c_nationkey )

 ***********
GROUP BY operation
Input Pipe 5
Output Pipe 6
Output Schema:
        Att SUM: DOUBLE
        Att r.r_regionkey: INT

GROUPING ON
r.r_regionkey

FUNCTION
((n.n_nationkey + r.r_regionkey))

Distinct Function: 1

 ***********
PROJECT operation
Input Pipe 6
Output Pipe 7
Output Schema:
        Att SUM: DOUBLE

****************************************
```

GTests:

Instructions to run:

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

Details:

Following are the details of 2 written GTests:

1. HeapPermutation: This tests HeapPermutation method which is used to find all permutations of joins.
2. QueryPlanNode: This tests the constructor of RelOpNode.

Output:

```
g++ -O2 -Wno-deprecated -o gTestQueryPlan.out QueryPlan.o
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from QueryPlanGTests
[ RUN      ] QueryPlanGTests.HeapPermutation
[       OK ] QueryPlanGTests.HeapPermutation (0 ms)
[ RUN      ] QueryPlanGTests.QueryPlanNode
[       OK ] QueryPlanGTests.QueryPlanNode (0 ms)
[----------] 2 tests from QueryPlanGTests (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. Distinct on aggregate was not implemented. This assignment assumes that is already handled in previous assignments. Hence Distinct on aggregate should be included in the “Relation Operators” assignment.