Project Report 4.2

Query Compilation and Optimization

Arpan Banerjee
UFID: 9359-9083
arpanbanerjee@ufl.edu

Krutantak Patil
UFID: 5615-6343
Krutantakb.patil@ufl.edu

1) Steps to compile and run - code, tests and gtests

(Assuming .tbl files would be provided to us in same directory)

i. **make a42.out** – Command to compile a42.cc into executable.
ii. **./a42.out** – Command to run a42.out.
iii. **make gtest.out** – Command to build the gtests.
iv. **./runGtestCases.sh** – Command to run the gtests.

2) Implementation of Query Planner

i. **class Optimizer**

a. This is the main class of the optimizer for which the constructor is called from a42.cc.
b. Stores the statistics and processes all the operations evaluating the best order.
c. Constructor - *Optimizer(Statistics* stats)* - The constructor performs the following tasks in this order -
    i. *constructLeafNodes()*
    ii. *processJoins()*
    iii. *ProcessSums()*
    iv. *processProjects()*
    v. *processDistinct()*
    vi. *processWrite()*
    vii. *printNodes()*

ii. **class OptimizerNode**

a. Encapsulates a node (operation) in the query planning tree.
b. This is the base class that all nodes inherit from.
c. Provides a skeleton infrastructure including pipes, schema and print functionality.
d. Constructors -
    i. OptimizerNode(const string& op, Schema* outSchema, Statistics* stats);
    ii. OptimizerNode(const string& op, Schema* outSchema, char* relation, Statistics* stats);
    iii. OptimizerNode(const string& op, Schema* outSchema, char* relations[], size_t num, Statistics* stats);
e. Member functions
    i. virtual void print(ostream& os = cout) const;
    ii. virtual void printAnnot(ostream& os = cout) const = 0;
    iii. virtual void printPipe(ostream& os) const = 0;
    iv. virtual void printChildren(ostream& os) const = 0;

iii. **class LeafNode**

a. Inherits OptimizerNode to represent leaf nodes in the tree.
b. These are effectively SelectFile operations, also storing a CNF for selection.
c. *hasCNF()* - Returns true if a CNF is used for selection.

iv. **class UnaryNode**

a. Inherits Optimizer Node and is used to represent nodes with one children such as Project, Dedup etc.
b. Contains a pointer to its child node.
c. Nodes that inherit from UnaryNode -
    i. ProjectNode
    ii. DedupNode
    iii. GroupByNode
    iv. SelectPipeNode
    v. SumNode
    vi. WriteNode

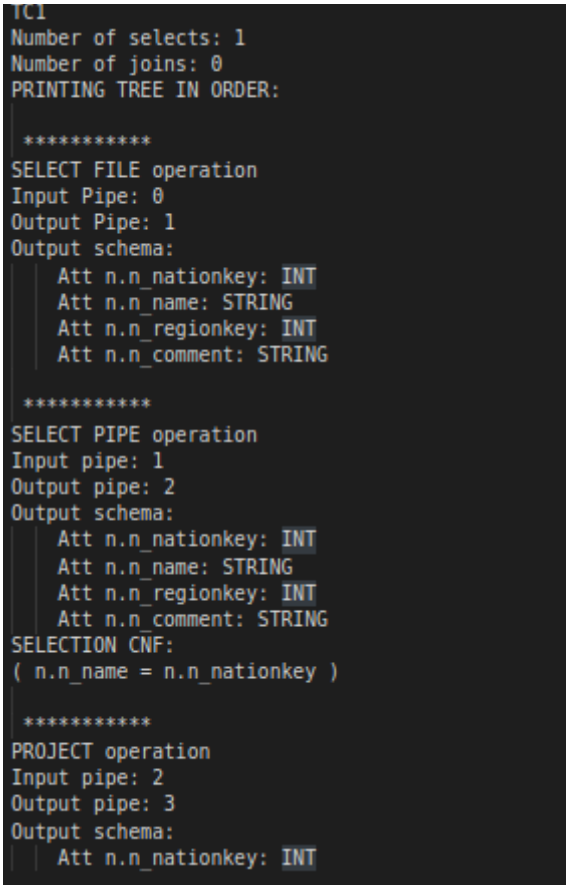
v. **class BinaryNode**

a. Inherits Optimizer Node and represents nodes with two children.
b. Stores pipe ids and pointers for both children.
c. Nodes that inherit from BinaryNode -
    i. JoinNode

3) ./test.out results for 1GB data

i. **Query 1 –**

```
SELECT n.n_nationkey
FROM nation AS n
WHERE (n.n_name = 'UNITED STATES')
```



ii. **Query 2 –**

SELECT n.n_name
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey > 5)

```
TC2
Number of selects: 1
Number of joins: 1
PRINTING TREE IN ORDER:

 ***********
SELECT FILE operation
Input Pipe: 0
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

 ***********
SELECT PIPE operation
Input pipe: 2
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 > n.n_nationkey )

 ***********
SELECT FILE operation
Input Pipe: 0
Output Pipe: 1
Output schema:
    Att r.r_regionkey: INT
    Att r.r_name: STRING
    Att r.r_comment: STRING

 ***********
JOIN operation
Left input pipe: 3
Right input pipe: 1
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 )

 ***********
PROJECT operation
Input pipe: 4
Output pipe: 5
Output schema:
    Att n.n_name: STRING
```

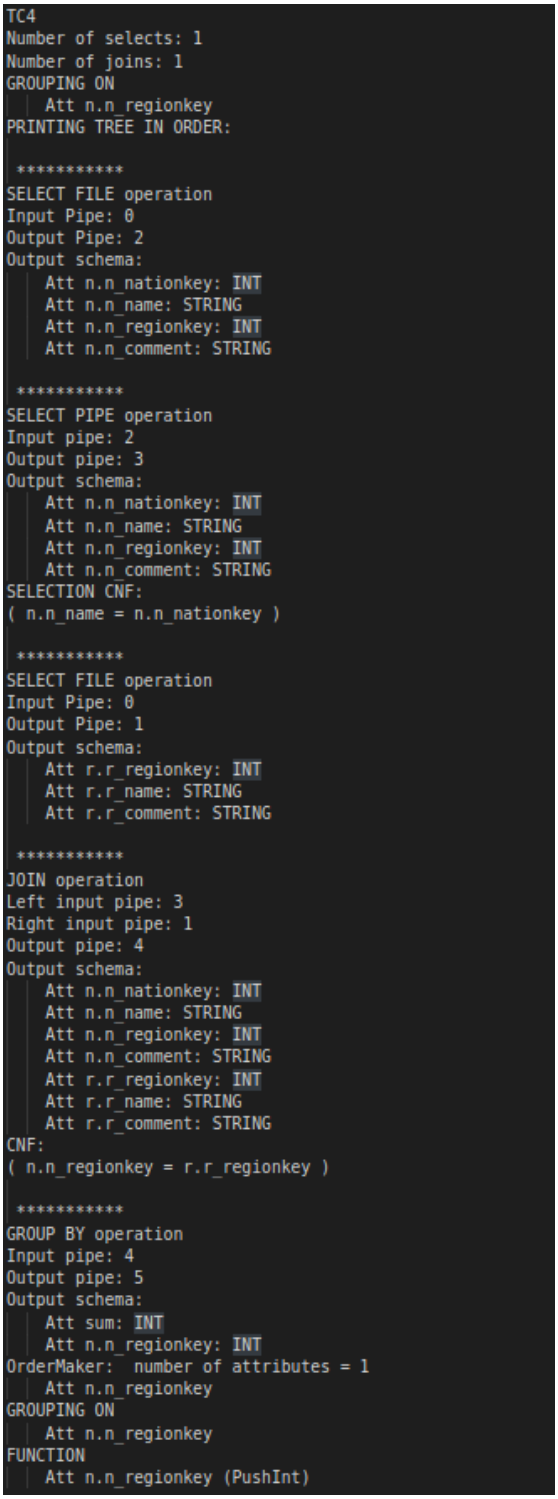
iii. **Query 3 –**

SELECT SUM (n.n_nationkey)
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES')

```
TC3
Number of selects: 1
Number of joins: 1
PRINTING TREE IN ORDER:

 ***********
SELECT FILE operation
Input Pipe: 0
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

 ***********
SELECT PIPE operation
Input pipe: 2
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_name = n.n_nationkey )

 ***********
SELECT FILE operation
Input Pipe: 0
Output Pipe: 1
Output schema:
    Att r.r_regionkey: INT
    Att r.r_name: STRING
    Att r.r_comment: STRING

 ***********
JOIN operation
Left input pipe: 3
Right input pipe: 1
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 )

 ***********
SUM operation
Input pipe: 4
Output pipe: 5
Output schema:
    Att sum: INT
FUNCTION
    Att n.n_nationkey (PushInt)
```

iv. **Query 4 –**

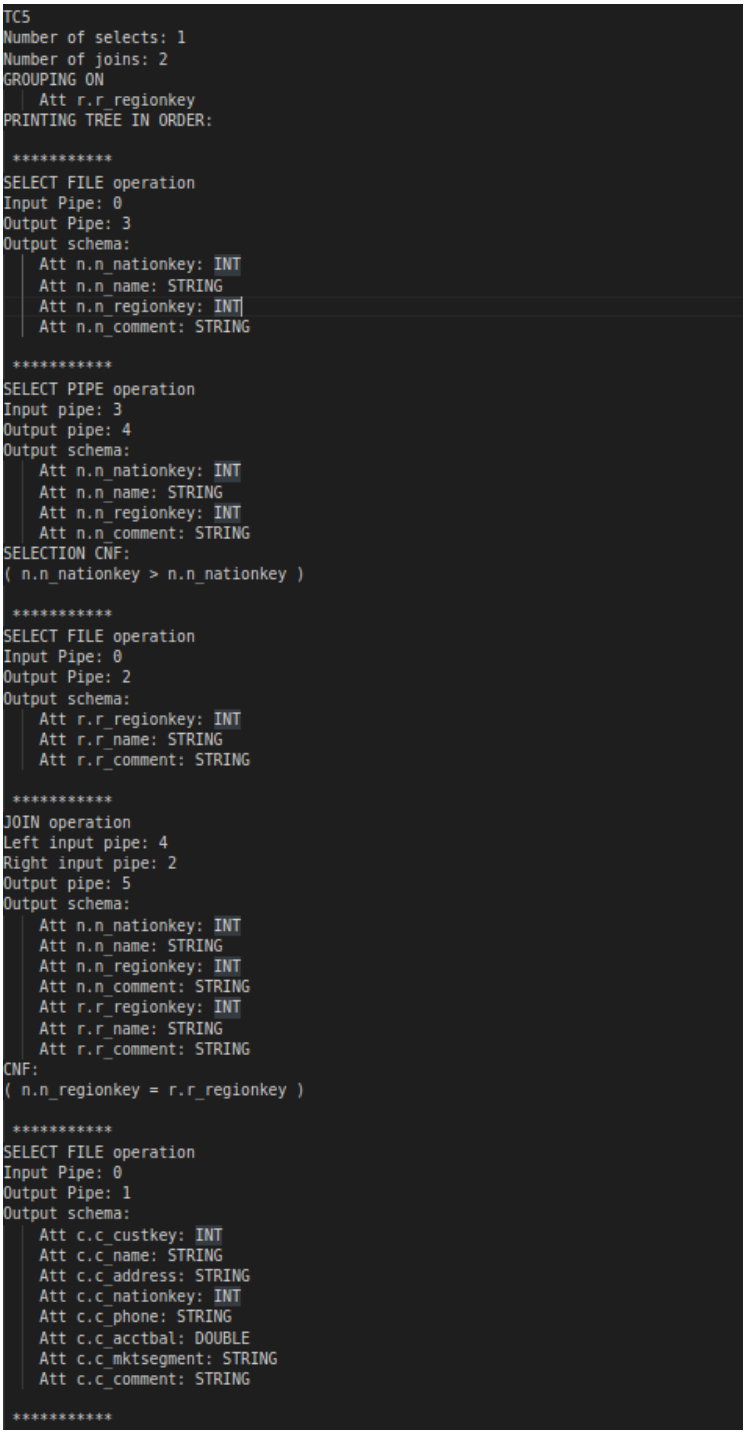
SELECT SUM (n.n_regionkey)
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES')
GROUP BY n.n_regionkey



```
TC4
Number of selects: 1
Number of joins: 1
GROUPING ON
    Att n.n_regionkey
PRINTING TREE IN ORDER:

 ***********
SELECT FILE operation
Input Pipe: 0
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

 ***********
SELECT PIPE operation
Input pipe: 2
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_name = n.n_nationkey )

 ***********
SELECT FILE operation
Input Pipe: 0
Output Pipe: 1
Output schema:
    Att r.r_regionkey: INT
    Att r.r_name: STRING
    Att r.r_comment: STRING

 ***********
JOIN operation
Left input pipe: 3
Right input pipe: 1
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 )

 ***********
GROUP BY operation
Input pipe: 4
Output pipe: 5
Output schema:
    Att sum: INT
    Att n.n_regionkey: INT
OrderMaker:  number of attributes = 1
    Att n.n_regionkey
GROUPING ON
    Att n.n_regionkey
FUNCTION
    Att n.n_regionkey (PushInt)
```

v. **Query 5 –**

SELECT SUM DISTINCT (n.n_nationkey + r.r_regionkey)
FROM nation AS n, region AS r, customer AS c
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey = c.c_nationkey) AND (n.n_nationkey > 10)
GROUP BY r.r_regionkey



```
 ***********
JOIN operation
Left input pipe: 5
Right input pipe: 1
Output pipe: 6
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 )

 ***********
DEDUPLICATION operation
Input pipe: 6
Output pipe: 7
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

 ***********
GROUP BY operation
Input pipe: 7
Output pipe: 8
Output schema:
    Att sum: INT
    Att r.r_regionkey: INT
OrderMaker:  number of attributes = 1
    Att r.r_regionkey
GROUPING ON
    Att r.r_regionkey
FUNCTION
    Att n.n_nationkey (PushInt)
    Att r.r_regionkey (PushInt)
```

3) GTests and results (./runGtestCases.sh)

i. **TEST (OPTIMIZER_TEST, CHECK_JOIN_QUERY_COUNT)** : We are running the query in tc6.sql and checking the results. This test checks that the number of joins calculated is correct.

ii. **TEST (OPTIMIZER_TEST, CHECK_SELECT_QUERY_COUNT) :** This test checks the number of select nodes or leaf nodes created.

```
arpan@arpan-pc:/run/media/arpan/Data/UFL/Sem2/DBI/database-system-implementation/a4-2test$ ./runGtestCases.sh
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from OPTIMIZER_TEST
[ RUN      ] OPTIMIZER_TEST.CHECK_JOIN_QUERY_COUNT
Number of selects: 1
Number of joins: 2
GROUPING ON s.s_suppkey
```

```
[       OK ] OPTIMIZER_TEST.CHECK_JOIN_QUERY_COUNT (22 ms)
[ RUN      ] OPTIMIZER_TEST.CHECK_SELECT_QUERY_COUNT
Number of selects: 1
```

```
[       OK ] OPTIMIZER_TEST.CHECK_SELECT_QUERY_COUNT (10 ms)
[----------] 2 tests from OPTIMIZER_TEST (32 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran. (33 ms total)
[  PASSED  ] 2 tests.
```