Commands to build project

Make sure you have gtest installed on your device and also have Linux subsytem on your device
> make clean
> make
> make a42.out
> ./a42.out

OR
You can also run the following commands
>./runTestCases42.sh
> make gtest
> ./gtest.out

These are the automated unit tests.

Implementation

The functions in the NodeForQuery class are implemented as follows.

1. class JoinNode()
This classes manages Join operation for Nodes

2. class NodeForQuery
This class implement the Query optimization requirements. This class is inherited by the specific node classes.

3. class SelectFileNode()
This class manages the SELECT FILE operation for Nodes.
4. class SumNode()
This class manages the Nodes for SUM operation.

5. class SelectPipeNode()
This class manages the Nodes for SELECT PIPE operation.

6. class GroupByNode()
This class manages the ©

He classes above will have the following three functions

i. Apply ()

This function is a recursive implementation that parses all nodes in the parseTree, calls the appropriate Run () method, depending on the relational operator found, and writes out the output schema. This applies to tree-based functions such as GroupBy, WriteOut, and DuplicateRemovalNode.

ii. printNode ()

Outputs node information such as leftID and output schema. Generated after "applying" the query optimizer.

iii. Generate CNF ()

This function creates a CNF form and creates a new record for each input. Returns the new CNF generated using the GrowFromParseTree.

The format of the output file generated is as per the output6.txt file provided in the starter code.

Testing

Run the following commands to run the test. Also make sure all .bin files have the .bin.meta. We have ensured that we have the a2-test.cc with the P4 code. This is being updates to generate the meta files for all the tables.

>make a42.out

> ./a42.out

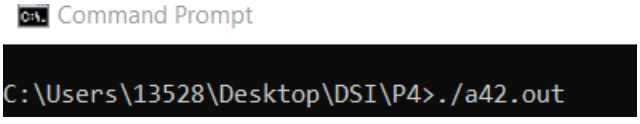
OR run the following commands to run test cases,

> ./runTestCases42.sh

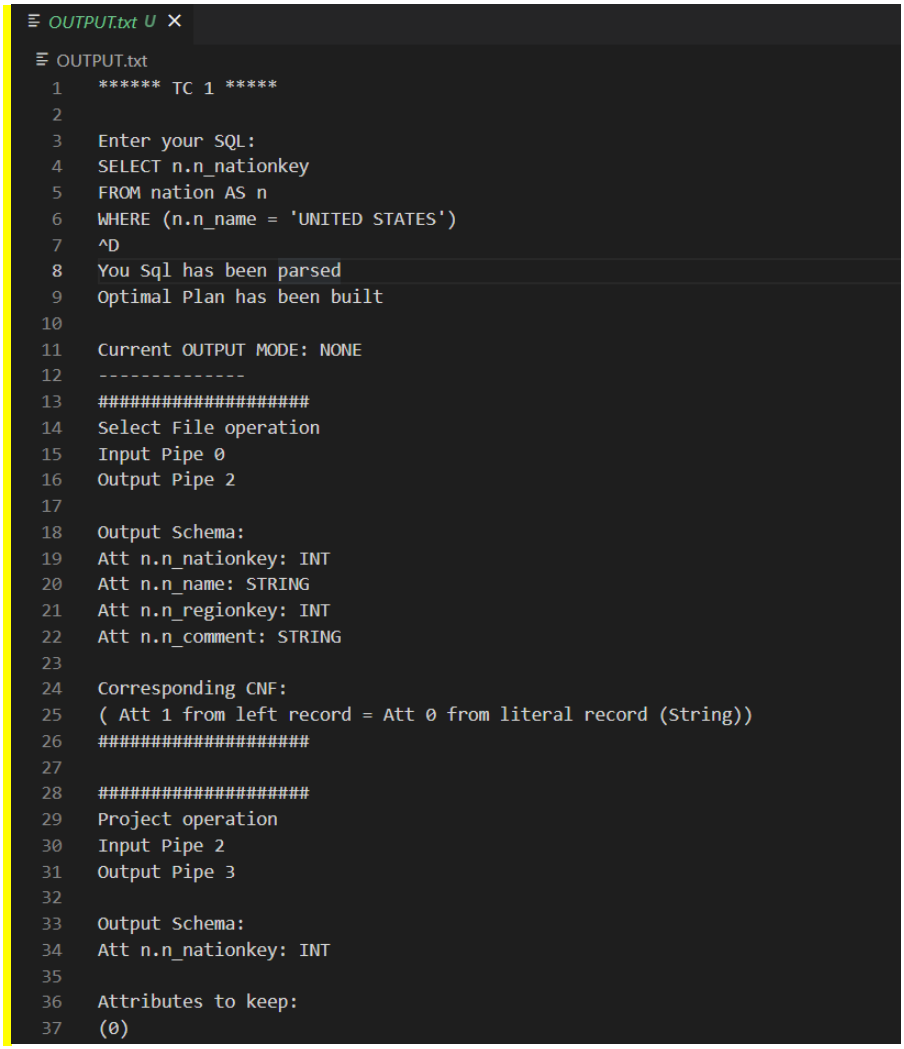
> make gtest

> ./gtest.out

runTestCases42.sh Results



Test Case 1


OUTPUT.txt U
OUTPUT.txt
****** TC 1 *****
Enter your SQL:
SELECT n.n_nationkey
FROM nation AS n
WHERE (n.n_name = 'UNITED STATES')
^D
You Sql has been parsed
Optimal Plan has been built
Current OUTPUT MODE: NONE

####################
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
Corresponding CNF:
(Att 1 from left record = Att 0 from literal record (String))
####################
####################
Project operation
Input Pipe 2
Output Pipe 3
Output Schema:
Att n.n_nationkey: INT
Attributes to keep:
(0)

Test Case 2

```
You Sql has been parsed
Optimal Plan has been built

Current OUTPUT MODE: NONE
--------------
####################
Select File operation
Input Pipe 0
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

Corresponding CNF:
( Att 0 from left record > Att 0 from literal record (Int))
####################

####################
Join operation
Left Input Pipe 4
Right Input Pipe 2
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
```

```
Corresponding CNF:
( Att 2 from left record = Att 0 from right record (Int))
####################

####################
Select File operation
Input Pipe 0
Output Pipe 2

Output Schema:
Att r.r_regionkey: INT
Att r.r_name: STRING
Att r.r_comment: STRING

Corresponding CNF:
####################

####################
Project operation
Input Pipe 5
Output Pipe 6

Output Schema:
Att n.n_name: STRING

Attributes to keep:
(1)
####################
```

Test Case 3

```
***** TC 3 *****

You Sql has been parsed
Optimal Plan has been built

Current OUTPUT MODE: NONE
-------------
####################
Select File operation
Input Pipe 0
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

Corresponding CNF:
( Att 1 from left record = Att 0 from literal record (String))
####################

####################
Join operation
Left Input Pipe 4
Right Input Pipe 2
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
```

OUTPUT.txt U

OUTPUT.txt

```
144   Att r.r_name: STRING
145   Att r.r_comment: STRING
146
147   Corresponding CNF:
148   ( Att 2 from left record = Att 0 from right record (Int))
149   ####################
150
151   ####################
152   Select File operation
153   Input Pipe 0
154   Output Pipe 2
155
156   Output Schema:
157   Att r.r_regionkey: INT
158   Att r.r_name: STRING
159   Att r.r_comment: STRING
160
161   Corresponding CNF:
162   ####################
163
164   ####################
165   Sum operation
166   Left Input Pipe 5
167   Output Pipe 6
168
169   Output Schema:
170   Att SUM: INT
171
172   Corresponding Function:
173   PushInt at index 0 for given record
174   Original litInput is: 0x0
175   returnsInt: 1
176   ####################
177
178   ***** TC 3 ends *****
179
```

Test Case 4

```
You Sql has been parsed
Optimal Plan has been built

Current OUTPUT MODE: NONE
--------------
####################
Select File operation
Input Pipe 0
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

Corresponding CNF:
( Att 1 from left record = Att 0 from literal record (String))
####################

####################
Join operation
Left Input Pipe 4
Right Input Pipe 2
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
```

OUTPUT.txt U

OUTPUT.txt

```
217   Corresponding CNF:
218   ( Att 2 from left record = Att 0 from right record (Int))
219   ####################
220
221   ####################
222   Select File operation
223   Input Pipe 0
224   Output Pipe 2
225
226   Output Schema:
227   Att r.r_regionkey: INT
228   Att r.r_name: STRING
229   Att r.r_comment: STRING
230
231   Corresponding CNF:
232   ####################
233
234   ####################
235   GroupBy operation
236   Left Input Pipe 5
237   Output Pipe 6
238
239   Output Schema:
240   Att SUM: INT
241
242   Corresponding OrderMaker:
243   NumAtts =     1
244     0:     2 Int
245
246   Corresponding Function:
247   PushInt at index 2 for given record
248   Original litInput is: 0x0
249   returnsInt: 1
250   ####################
251
```

Test Case 5

OUTPUT.txt U

OUTPUT.txt

```
253
254   ***** TC 5 *****
255
256   You Sql has been parsed
257   Optimal Plan has been built
258
259   Current OUTPUT MODE: NONE
260   --------------
261   ####################
262   Select File operation
263   Input Pipe 0
264   Output Pipe 6
265
266   Output Schema:
267   Att n.n_nationkey: INT
268   Att n.n_name: STRING
269   Att n.n_regionkey: INT
270   Att n.n_comment: STRING
271
272   Corresponding CNF:
273   ( Att 0 from left record > Att 0 from literal record (Int))
274   ####################
275
276   ####################
277   Join operation
278   Left Input Pipe 6
279   Right Input Pipe 4
280   Output Pipe 7
281
282   Output Schema:
283   Att n.n_nationkey: INT
284   Att n.n_name: STRING
285   Att n.n_regionkey: INT
286   Att n.n_comment: STRING
287   Att r.r_regionkey: INT
288   Att r.r_name: STRING
289   Att r.r_comment: STRING
```

OUTPUT.txt U

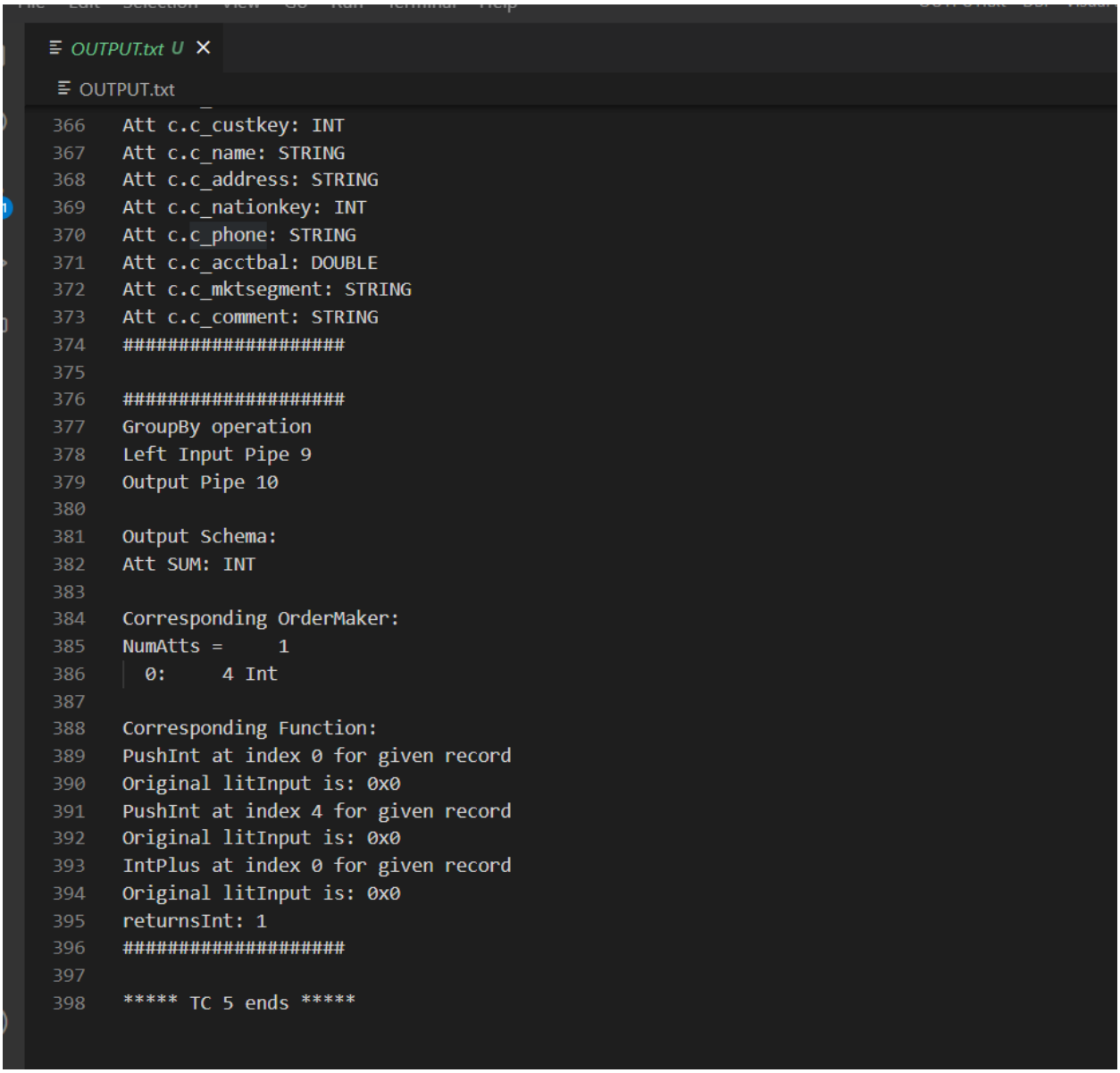
OUTPUT.txt

```
289   Att r.r_comment: STRING
290
291   Corresponding CNF:
292   ( Att 2 from left record = Att 0 from right record (Int))
293   ####################
294
295   ####################
296   Select File operation
297   Input Pipe 0
298   Output Pipe 4
299
300   Output Schema:
301   Att r.r_regionkey: INT
302   Att r.r_name: STRING
303   Att r.r_comment: STRING
304
305   Corresponding CNF:
306   ####################
307
308   ####################
309   Join operation
310   Left Input Pipe 7
311   Right Input Pipe 2
312   Output Pipe 8
313
314   Output Schema:
315   Att n.n_nationkey: INT
316   Att n.n_name: STRING
317   Att n.n_regionkey: INT
318   Att n.n_comment: STRING
319   Att r.r_regionkey: INT
320   Att r.r_name: STRING
321   Att r.r_comment: STRING
322   Att c.c_custkey: INT
323   Att c.c_name: STRING
324   Att c.c_address: STRING
325   Att c.c_nationkey: INT
326   Att c.c_phone: STRING
```

OUTPUT.txt U

OUTPUT.txt

```
Att c.c_comment: STRING

Corresponding CNF:
( Att 0 from left record = Att 3 from right record (Int))
####################

####################
Select File operation
Input Pipe 0
Output Pipe 2

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

Corresponding CNF:
####################

####################
Duplicate remove operation
Left Input Pipe 8
Output Pipe 9

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
```



GTest Results

Make sure you have gtest installed on your device. Then the Makefile is updated to build the teste written in Gtest. Run the following commands

>make clean && make gtest
> ./gtest.out

The gtest have been written to test the constructor and the destructor for the below mentioned classes.
JoinNode()
SelectFileNode()