Data Base System Implementation COP-6726

Submitted by:
Naman Arora
UFID: 3979-0439
Nikunj Sarda
UFID: 9360-6581

Introduction:

This document is intended to present the outcomes and the process of the completed work for the Project 4_2. The query optimization for DB file implementation have been successfully materialized and tested. The repository for all the work is present <u>here</u>.

Source File Hierarchy:

The directories are arranged in a flat manner with exception of a tpch-dbgen project directory. The tpch-dbgen is the git repository for the TPCH sub program. Please make sure the *.tbl files are present BEFORE test is run.

Some Points to Note:

 \rightarrow Note 1: flex version equal or higher than 2.6 is needed to run the above implementation and not get below error during compilation phase:

"lexer_func.l:67:1: error: 'yyfunclineno' undeclared (first use in this function); did you mean 'yyfuncleng'?"

while running the make to generate test.out file. If such an error arises, however, please uncomment line 16 in 'lexer func.1'.

 \rightarrow Note 2:

Pipe id for input from database file is is -2 and Pipe id for final output is -1

The compilation process:

The compilation process is accomplished by a recursive call to Makefiles in each subdirectory by the Makefile in the src/ directory.

The provided NamanArora_NikunjSarda_p42.zip has one sub-directory, src/.

\$> unzip NamanArora_NikunjSarda_p42.zip

\$> cd proj42

Now to build:

\$> make

The above command will create a42.out

To run the associated Gtests for the current submission:

\$> make gtest.out

\$> ./gtest.out

To clean the repository, excluding *.bin and *.tbl files,

\$> make clean

To clean the repository, including *.bin and *.tbl files

\$> make distclean

Change Log:

- → Added qp_tree.cc
- → Added qp_tree_helper.cc

qp_tree.cc:

- → query :: query(struct FuncOperator *finalFunction, struct TableList *tables, struct AndList *boolean, struct NameList *groupingAtts, struct NameList *attsToSelect, int distinctAtts, int distinctFunc): constructor to structure query, initializing the variable to initial value
- → query :: ~query(): Destructor to structure query
- operation :: operation(): constructor to structure operation, initializing the variable to initial value
- operation :: operation(int flag): constructor to structure operation, initializing the variable to initial value and type (struct variable) to flag
- → operation :: operation(struct AndList *a_list, Qptree *ref): constructor to structure operation, initializing the variable to provided values
- → operation :: ~operation(): destructor to structure operation
- → void operation :: print(): method to print process information
- → bool sel_op_comp :: operator()(operation *l, operation *r): method to compare cost of left and right operation and return boolean value accordingly
 - → bool join_op_comp :: operator()(operation *l, operation *r): method to compare cost of left and right operation and return boolean value accordingly
- → tableInfo :: tableInfo(): constructor to structure tableInfo
- → tableInfo :: ~tableInfo(): destructor to structure tableInfo
- → struct operation *tableInfo :: dispense_select(): method to evaluate cost for select
- → Qptree :: Qptree(char *stat_fname, char *catalog_file):constructor to class Qptree to initialize class variable
- → Qptree :: ~Qptree(): destructor to class Qptree
- → void Qptree :: process(struct query *q): method to construct the expense tree
- → void print_in_order(struct operation *tree): method to print the expense tree

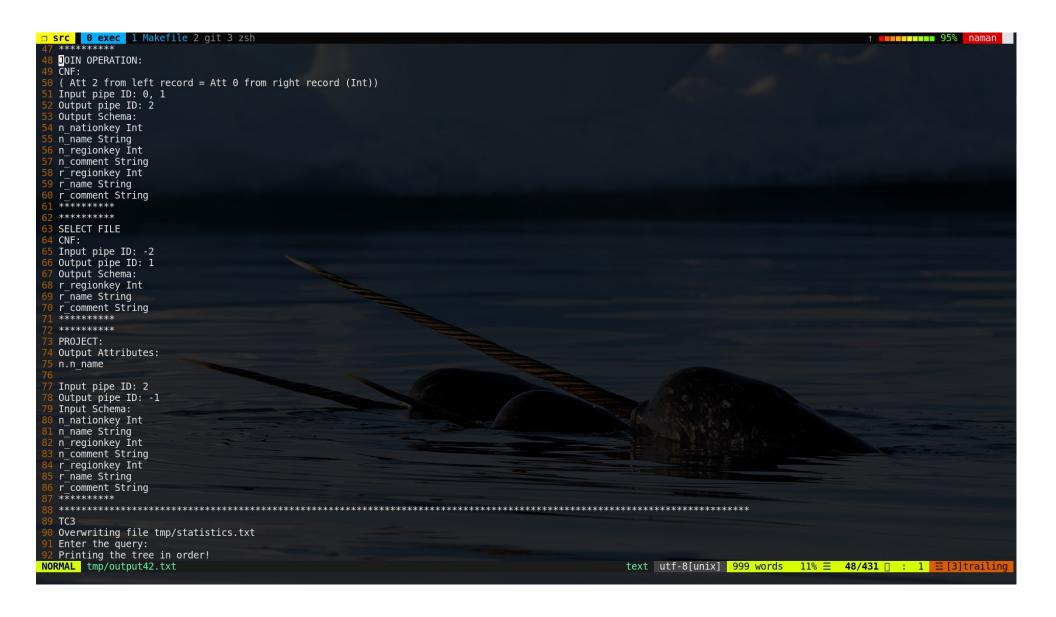
qp_tree_helper.cc:

- → void Qptree :: get_attr(char *att_name, pair<string, unordered_map<string, tableInfo> :: iterator> &p): method to get the attribute via name passed to the function and stored it in pair object
- → void Optree :: process(struct TableList *tables): method to process info for given tables list

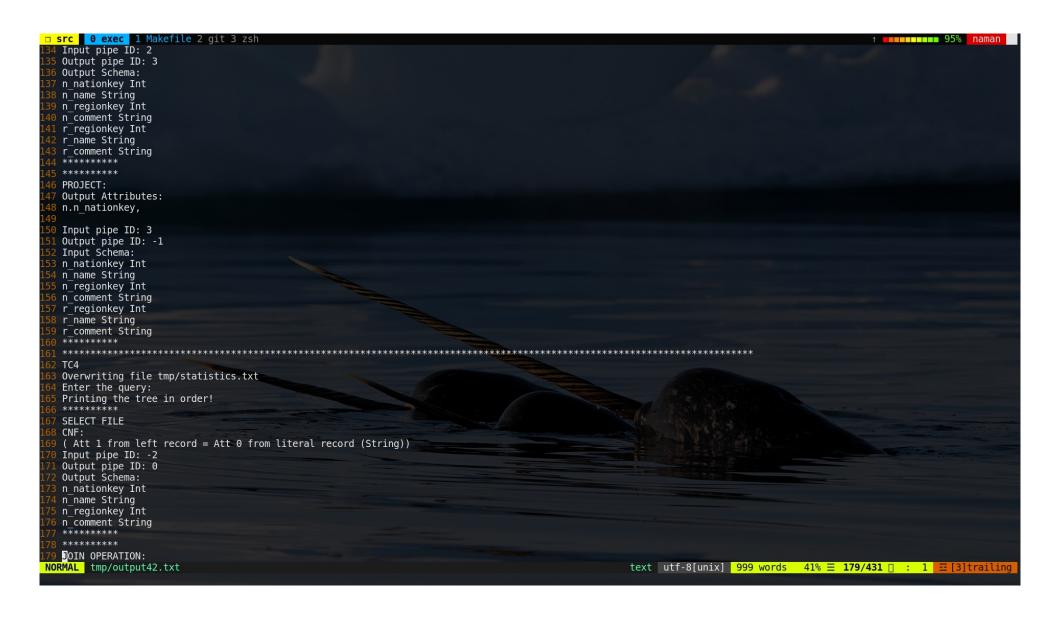
- → void Qptree :: process(struct operation *op, struct AndList *a_list, struct OrList *o_list, char **rels, int *curr_indx): method to process "or" operation
- → void Qptree :: process(struct AndList *a_list): method to process "and" operator starting from right end
- → int Qptree :: dispense_pipe(): method to return pipe count
- ⇒ struct operation *Qptree :: dispense_join(struct operation *j_op, int indx, vector<operation *> &j_vec, stack<operation *> &j_stk): method to get least cost join
- → void Qptree :: process_join(struct operation *j_op, vector<operation *> &j_vec, stack<operation *> &j_stk): method to process joins according to there expense

Outputs:

```
□ src 0 exec 1 Makefile 2 git 3 zsh
                                                                                                                             1 95% naman
   C1
   Overwriting file tmp/statistics.txt
   Enter the query:
   Printing the tree in order!
   *******
   SELECT FILE
   ( Att 1 from left record = Att 0 from literal record (String))
   Input pipe ID: -2
   Output pipe ID: 0
   Output Schema:
   n nationkey Int
  n name String
 4 n regionkey Int
 .5 n_comment String
.6 ********
   PROJECT:
   Output Attributes:
   n.n nationkey
   Input pipe ID: 0
   Output pipe ID: -1
   Input Schema:
   n nationkey Int
   n name String
   n regionkey Int
   n comment String
   ********
   Overwriting file tmp/statistics.txt
   Enter the query:
   Printing the tree in order!
   ******
   SELECT FILE
   ( Att 0 from left record > Att 0 from literal record (Int))
   Input pipe ID: -2
   Output pipe ID: 0
   Output Schema:
   n nationkey Int
   n name String
  n regionkey Int
   n comment String
  ********
NORMAL tmp/output42.txt
                                                                                         text utf-8[unix] 999 words 0% ≡ 1/431 [ : 1 ≡
"tmp/output42.txt" 431L, 7451C
```



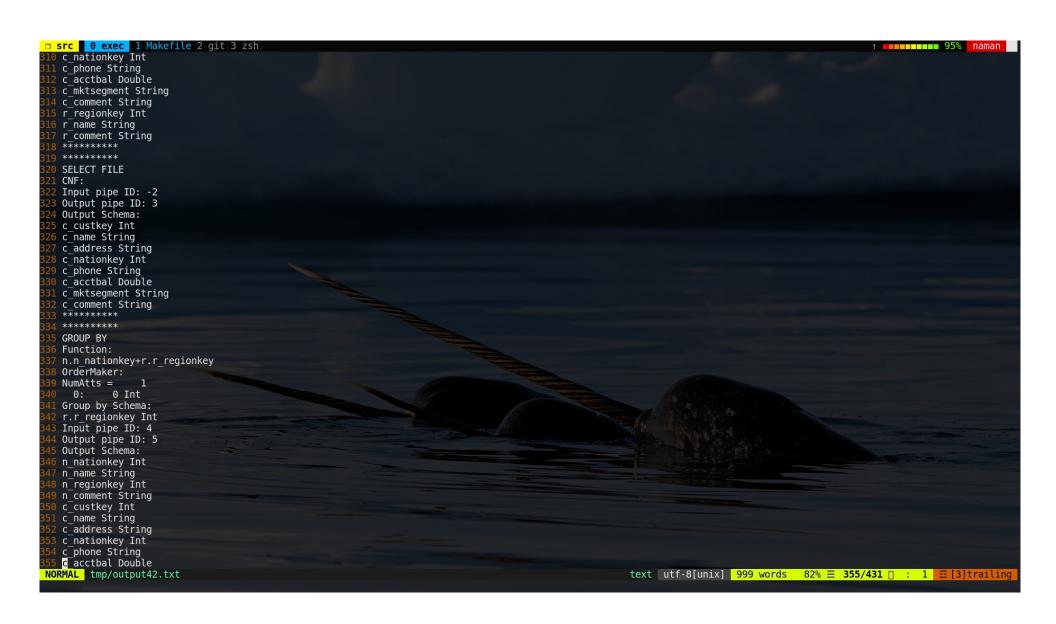
```
□ src 0 exec 1 Makefile 2 git 3 zsh
                                                                                                                                                  ↑ •••••••••• 95% naman
   Overwriting file tmp/statistics.txt
   Enter the query:
   Printing the tree in order!
   ******
   SELECT FILE
   CNF:
   ( Att 1 from left record = Att 0 from literal record (String))
   Input pipe ID: -2
   Output pipe ID: 0
Output Schema:
  n_nationkey Int
   n name String
   n_regionkey Int
   n_comment String
*********
   JOIN OPERATION:
   CNF:
   ( Att 2 from left record = Att 0 from right record (Int))
   Input pipe ID: 0, 1
   Output pipe ID: 2
   Output Schema:
   n nationkey Int
   n name String
   n regionkey Int
   n_comment String
   r_regionkey Int
   r name String
   r comment String
   *******
   SELECT FILE
   CNF:
   Input pipe ID: -2
   Output pipe ID: 1
   Output Schema:
   r regionkey Int
   r name String
   r comment String
   ********
   ******
   SUM
   Function:
   n.n nationkey
   Input pipe ID: 2
   Qutput pipe ID: 3
NORMAL tmp/output42.txt
                                                                                                       text utf-8[unix] 999 words 31% ≡ 135/431 [ : 1 ≡ [3]trailing
```

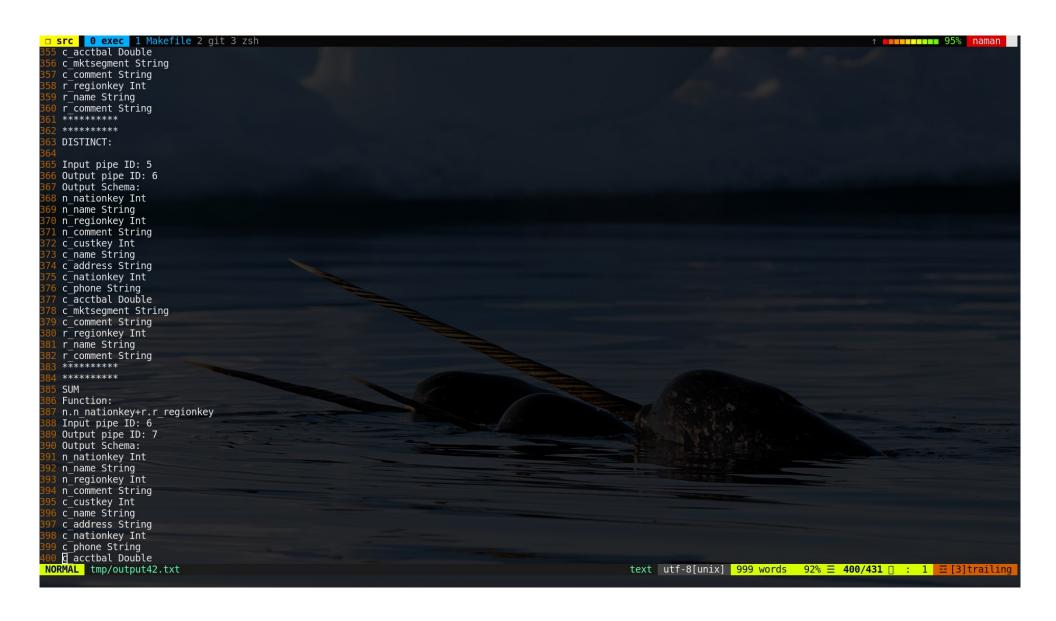


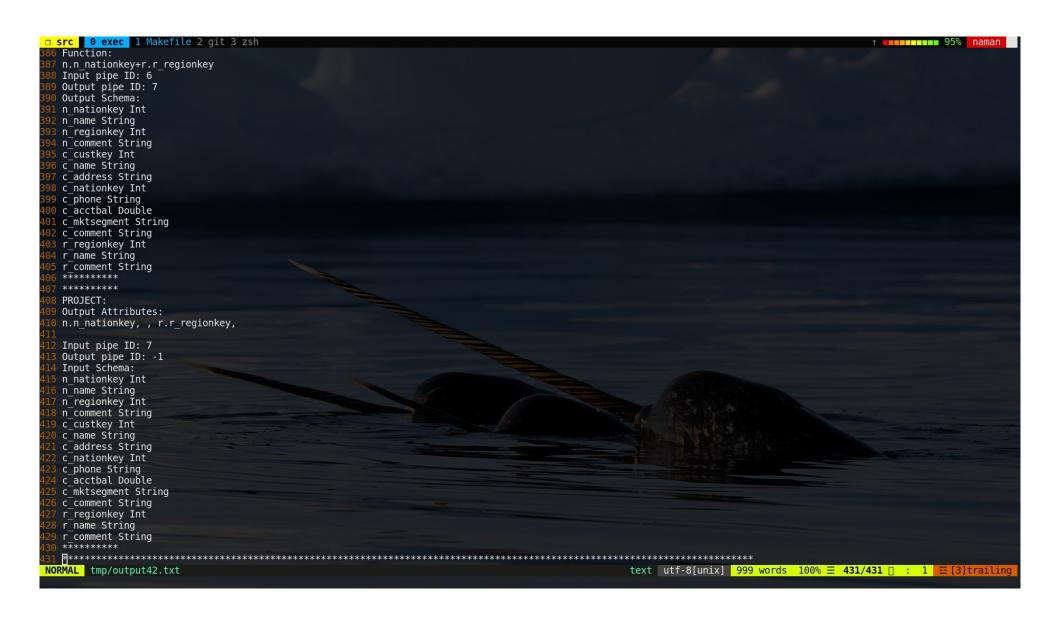
```
1 95% naman
   JOIN OPERATION:
   CNF:
   ( Att 2 from left record = Att 0 from right record (Int))
   Input pipe ID: 0, 1
   Output pipe ID: 2
Output Schema:
   n nationkey Int
   n_name String
   n_regionkey Int
   n_comment String
  r_regionkey Int
  r name String
 1 r_comment String
   SELECT FILE
   CNF:
   Input pipe ID: -2
   Output pipe ID: 1
Output Schema:
   r_regionkey Int
   r name String
   r_comment String
   GROUP BY
   Function:
   n.n regionkey
   OrderMaker:
   NumAtts =
           0 Int
   Group by Schema:
   n.n regionkey Int
   Input pipe ID: 2
   Output pipe ID: 3
Output Schema:
   n_nationkey Int
   n_name String
   n regionkey Int
   n comment String
   r_regionkey Int
   r name String
   r comment String
   ********
   *******
                                                                                                   text utf-8[unix] 999 words 51% ≡ 223/431 [ : 1 ≡ [3]trai
NORMAL tmp/output42.txt
```

```
↑ ••••••••••• 95% <u>naman</u>
□ src 0 exec 1 Makefile 2 git 3 zsh
  r_comment String
  ******
  SUM
  Function:
  n.n regionkey
  Input pipe ID: 3
Output pipe ID: 4
  Output Schema:
  n_nationkey Int
  n name String
  n regionkey Int
  n comment String
 4 r_regionkey Int
  r name String
 6 r_comment String
7 *********
  ******
  PROJECT:
  Output Attributes:
  n.n regionkey,
   Input pipe ID: 4
  Output pipe ID: -1
  Input Schema:
  n nationkey Int
  n name String
  n_regionkey Int
  n comment String
  r_regionkey Int
  r_name String
  r comment String
  Overwriting file tmp/statistics.txt
  Enter the query:
  Printing the tree in order!
  ******
  SELECT FILE
   ( Att 0 from left record > Att 0 from literal record (Int))
  Input pipe ID: -2
  Output pipe ID: 0
  Output Schema:
  n nationkey Int
                                                                                              text utf-8[unix] 999 words 61\% \equiv 266/431 \, \square : 1 \equiv [3] \text{train}
NORMAL tmp/output42.txt
```

```
□ src 0 exec 1 Makefile 2 git 3 zsh
                                                                                                                                                    1 95% naman
 6 n_nationkey Int
   n_name String
   n regionkey Int
   n comment String
   ********
   JOIN OPERATION:
   ( Att 2 from left record = Att 0 from right record (Int))
   Input pipe ID: 0, 1
   Output pipe ID: 2
   Output Schema:
   n nationkey Int
   n name String
  n regionkey Int
 1 n comment String
  2 r_regionkey Int
  3 r name String
  4 r comment String
   ********
   ******
   SELECT FILE
   CNF:
   Input pipe ID: -2
   Output pipe ID: 1
Output Schema:
   r regionkey Int
   r name String
   r comment String
   *******
   JOIN OPERATION:
   ( Att 0 from left record = Att 3 from right record (Int))
   Input pipe ID: 2, 3
   Output pipe ID: 4
Output Schema:
   n nationkey Int
   n_name String
   n regionkey Int
   n comment String
   c_custkey Int
   c name String
   c address String
   c nationkey Int
  1 c phone String
NORMAL tmp/output42.txt
                                                                                                         text utf-8[unix] 999 words 72\% \equiv 311/431 \square : 1 \equiv [3]trai
```







GTest

```
□ src 0 exec 1 Makefile 2 git 3 zsh
                                                                                                                                           1 96% naman
[naman@Oxblad3]-(~/git/database-from-scratch/src/submit)-[git://proj4 2 x]-
l<sub>></sub> ls
bigq.cc
                    comparison.h function.cc lexer func.l parse tree.h
                                                                              record.cc
                                                                                           runTestCases42.sh stat helper.cc tournament.cc
bigq.h
                                 function.h
                                              lexer.l
                                                                                                                            tournament.h
                    dbfile.cc
                                                            pipe.cc
                                                                              record.h
                                                                                           schema.cc
                                                                                                             statistics.cc
catalog
                    dbfile.h
                                              Makefile
                                                            pipe.h
                                                                                           schema.h
                                                                                                             statistics.h
                                                                                                                            tpch-dbgen
                                 gtest.cc
                                                                              run gen.cc
                                                                                                                            two_way_list.cc
two_way_list.h
comparison.cc
                    defs.h
                                                            qp tree.cc
                                                                              run gen.h
                                                                                           sorted.cc
                                  gtest.out
                                              parse func.h
                                                                                                             test.cc
comparison engine.cc file.cc
                                  heap.cc
                                              parser func.y qp tree.h
                                                                              run merge.cc sorted.h
                                                                                                             test.h
comparison engine.h file.h
                                                            qp tree helper.cc run merge.h sort helper.cc
                                 heap.h
                                              parser.y
 [naman⊕0xblad3]-(~/git/database-from-scratch/src/submit)-[git://proj4 2 x]-
 > ./gtest.out
 =======] Running 2 tests from 1 test case.
            Global test environment set-up.
           2 tests from QTREETEST
 RUN
           QTREETEST.pipecount
File [
            OK ] QTREETEST.pipecount (0 ms)
 RUN
            QTREETEST.selops
       OK ] QTREETEST.selops (0 ms)
  -----] 2 tests from QTREETEST (0 ms total)
 =======] 2 tests from 1 test case ran. (0 ms total)
  PASSED ] 2 tests.
  naman@0xblad3]-(~/git/database-from-scratch/src/submit)-[git://proj4 2 x]-
```