PROJECT 1 REPORT: HEAP FILE IMPLEMENTATION OF DATABASE

GROUP MEMBERS:

1. SHREYA DASGUPTA (UFID - 47016900)
2. YASH PATTARKINE (UFID - 28616005)

INSTRUCTIONS TO COMPILE THE CODE:

1. Unzip "shreyaDasgupta_yashPattarkine_p1.zip". Open terminal on the machine and go to the folder where the unzipped files are. Use command -- cd<pathname/foldername>
2. Command -- make clean.
3. Compile main function. Use commands – make main
4. Compile test.cc. Use command -- make test.out
5. Run test.out. Use command -- ./test.out
6. For gtests, complile it using make gtest.out
7. Run gtest using ./gtest.out

BRIEF OVERVIEW OF ALL METHODS OF DBFile Class:

Attributes:

cur_file -> an instance of 'File' Class.

cur_page_w -> an instance of 'Page' Class to keep track while writing outputs to console

cur_page_r -> an instance of 'Page' Class to keep track while writing outputs to console

file_name -> to store the name of the file operated on

page_no -> to keep track of the page currently being read from the database.

Methods:

1. **Create:** This function simply creates a file. We have three parameters here, and we are only using filetype (ftype) of heap for this assignment. At first, we initialize our 'file_name' variable with the path of the file (specified by the fpath parameter). Then, we call the 'Open' -function of 'File' class with parameters 0 and fpath. 0 means that if the file does not exist, the program will create the file and open it, otherwise it will simply open the already existing file.

2. **Load:** This function takes in 'f_schema' and 'loadpath' as its parameters and is used to bulk load the DBFile instance from a file, specified by 'loadpath' parameter. Here, we open the file in ''read"- mode and as long as there is a valid record existing in the opened file, we keep on adding them using "SuckNextRecord" - function.

3. **Open:** For this function, we assume that a file is already created and we open it using "Open" - function of the File-class, with parameters (1, as we assume a file to be opened always exists and the file-path, to specify the file to be opened). This takes in the path of the file to be opened as parameter.

4. **Add:** This function adds a record to the end of the file. This record is taken as a parameter to the function.

5. **GetNext - Version 1:** In this function, we are first checking if the page contains any record or not. If there exists a record, it is directly returning 1. If there is no record in the page, then we are checking if the file contains any pages. If not, it is loading the page to the file and setting the page write pointer (cur_page_w) to point to 1st page and incrementing the "page_no". Now, if there is an existing page, we are checking, if the page has records. If records are available, we are adding that page to the file. Then we are making our write pointer point to the new page, incrementing the "page_no" and returning 1.

6. **GetNext – Version 2:** This function is comparing the entered cnf to the record which is got in rec. (And how to get that record in rec is defined in GetNext – Version (1) If the record matches the cnf, it is returning 1. Otherwise it is returning 0. In this function, we have created an object of "ComparisonEngine" Class and utilized the function Compare of the same, to compare the input cnf and fetch results accordingly.

7. **Close:** This function assumes that a file is already open and is worked on, and it simply closes that file. The logic is to first check if any non-empty page is currently being read and to add it to the file and empty the page-contents out after that. Then, we use

"Close()"- function on our file object to close the file, and return 1 on success or 0 on failure.

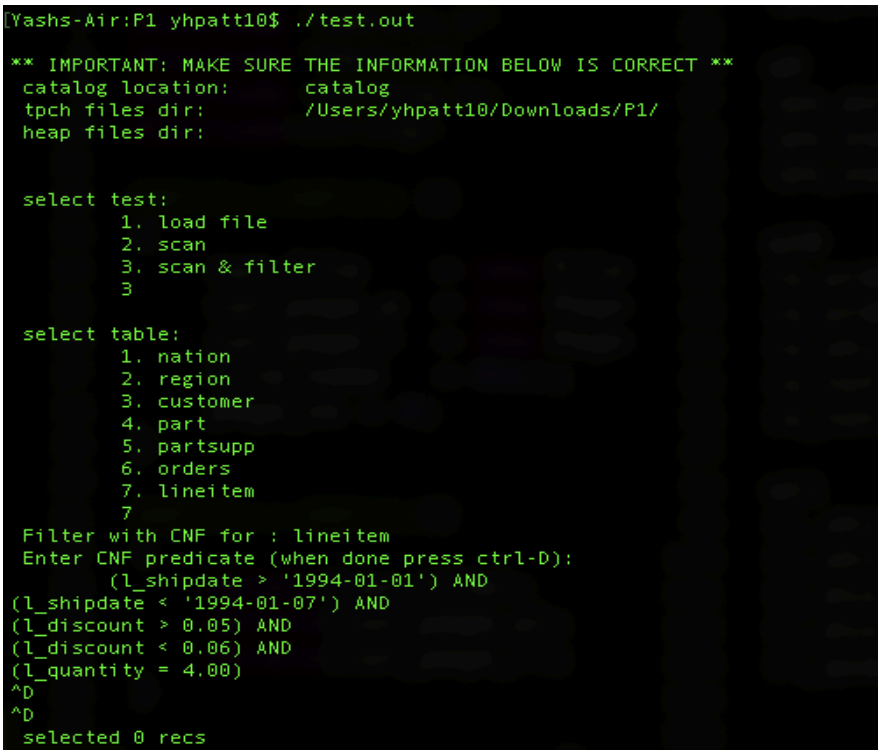
8. **MoveFirst:** In this function, we are forcibly moving the page pointer to point to the first record the file.

9. Besides these, we have also added an additional function "**GetNumRecords**" in File class, to access the private variable "**numRecs**" and use it in our code of DBFile.

Screen Shots:

Query 12:

Query 11:


Yashs-Air:P1 yhpatt10$ ./test.out
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: /Users/yhpatt10/Downloads/P1/
heap files dir:
select test:
1. load file
2. scan
3. scan & filter
3
select table:
1. nation
2. region
3. customer
4. part
5. partsupp
6. orders
7. lineitem
7
Filter with CNF for : lineitem
Enter CNF predicate (when done press ctrl-D):
(l_shipdate > '1994-01-01') AND
(l_shipdate < '1994-01-07') AND
(l_discount > 0.05) AND
(l_discount < 0.06) AND
(l_quantity = 4.00)
^D
^D
selected 0 recs

Query 3:

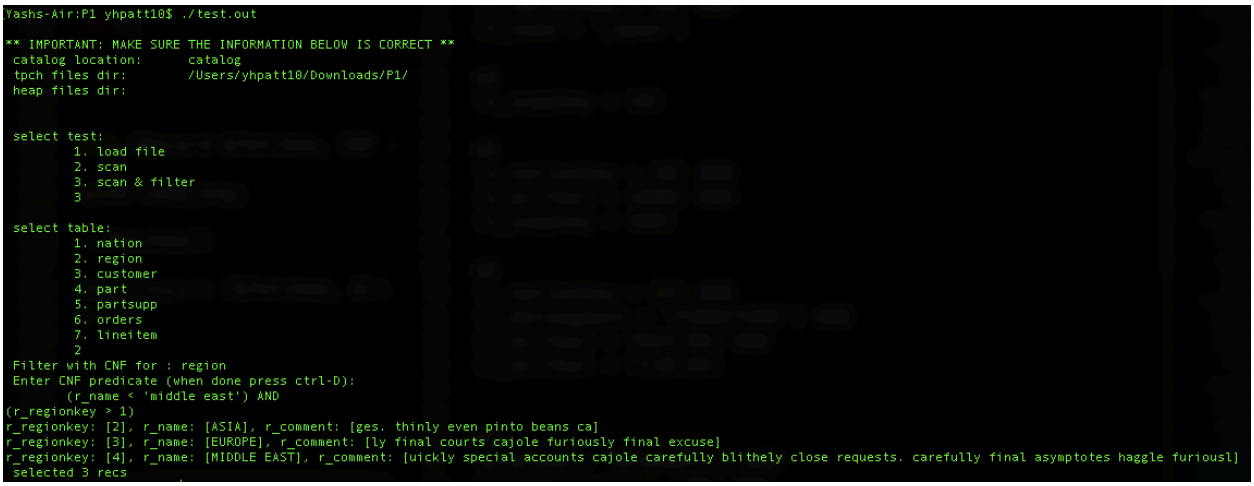
```
Yashs-Air:P1 yhpatt10$ ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /Users/yhpatt10/Downloads/P1/
 heap files dir:

 select test:
         1. load file
         2. scan
         3. scan & filter
         3

 select table:
         1. nation
         2. region
         3. customer
         4. part
         5. partsupp
         6. orders
         7. lineitem
         1
 Filter with CNF for : nation
 Enter CNF predicate (when done press ctrl-D):
        (n_regionkey = 3) AND
(n_nationkey > 10) AND
(n_name > 'japan')
 selected 0 recs
```

Query 2:



Query 1:

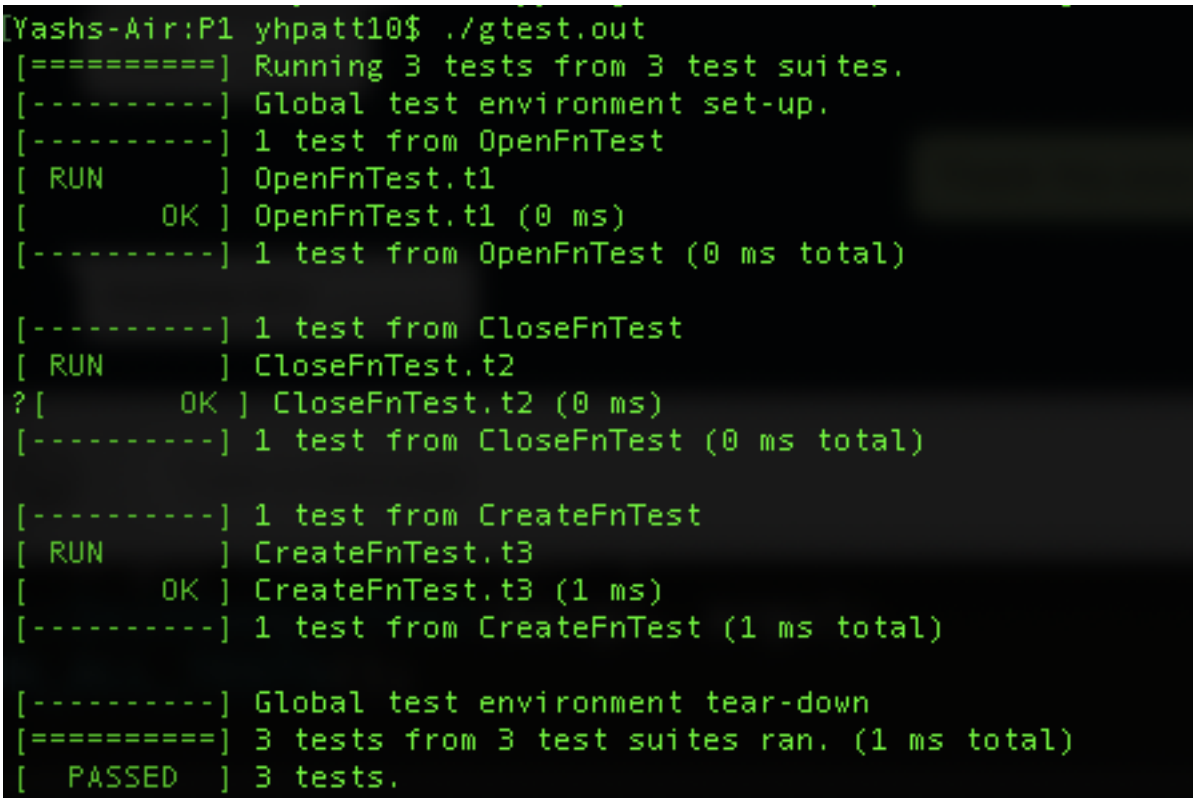
```
Yashs-Air:P1 yhpatt10$ ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /Users/yhpatt10/Downloads/P1/
 heap files dir:


 select test:
        1. load file
        2. scan
        3. scan & filter
        3

 select table:
        1. nation
        2. region
        3. customer
        4. part
        5. partsupp
        6. orders
        7. lineitem
        2
 Filter with CNF for : region
 Enter CNF predicate (when done press ctrl-D):
       (r_name = 'EUROPE')
r_regionkey: [3], r_name: [EUROPE], r_comment: [ly final courts cajole furiously final excuse]
 selected 1 recs
```

Screenshot from Gtest


Yashs-Air:P1 yhpatt10$ ./gtest.out
[==========] Running 3 tests from 3 test suites.
[----------] Global test environment set-up.
[----------] 1 test from OpenFnTest
[RUN   ] OpenFnTest.t1
[OK] OpenFnTest.t1 (0 ms)
[----------] 1 test from OpenFnTest (0 ms total)
[----------] 1 test from CloseFnTest
[RUN   ] CloseFnTest.t2
?[OK] CloseFnTest.t2 (0 ms)
[----------] 1 test from CloseFnTest (0 ms total)
[----------] 1 test from CreateFnTest
[RUN   ] CreateFnTest.t3
[OK] CreateFnTest.t3 (1 ms)
[----------] 1 test from CreateFnTest (1 ms total)
[----------] Global test environment tear-down
[==========] 3 tests from 3 test suites ran. (1 ms total)
[PASSED] 3 tests.