Project Report: Database System Implementation (COP6726)

Project 1 : Heap Implementation

Name: Vikas Chaubey**, UFID:** 3511 5826**, Email:** vikas.chaubey@ufl.edu

I have done this project alone without any project partner. Hence I am submitting it on my behalf.

1) **Compile and Run Instructions:** This program has been tested on linux machine of UF CISE Linux servers hence all the libraries related to googletests are generated accordingly.

To run main: In order to run main.cc, Please make sure that in main.cc file right path to "tpchgen" table files is updated, the variable that needs to be updated is "FILE *tableFile".

```
> make clean
> make main
> ./main
```

To run test.out: In order to run test.out, Please make sure that in test.cc file dbfile_dir, tpch_dir and catalog_path are updated correctly with right locations.

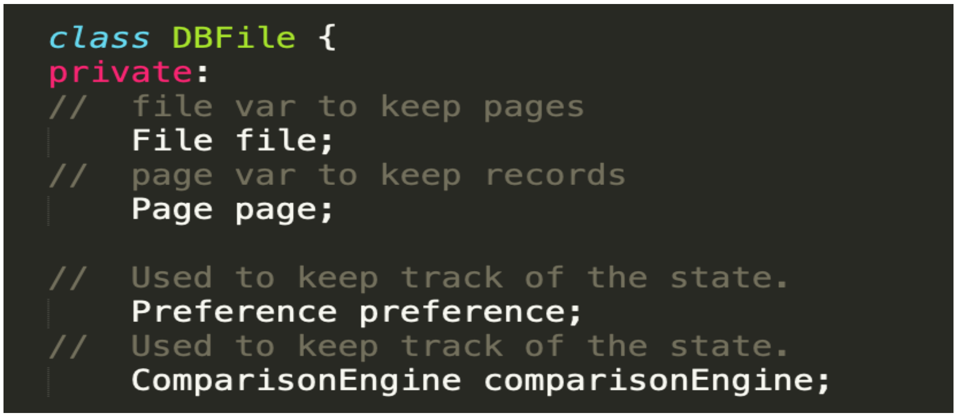
```
> make clean
> make test.out
> ./test.out
```

To run gtest (google tests): : In order to run gtest.out, Please make sure that in gtest.cc file variables like dbBinFilesPath, tpchTblFilesPath and catalogFilePath are updated correctly with right locations. Before running the gtests please delete "region.bin and region.pref" files if they have been created in order to test load and create functionalities properly. The test cases are written around these files.

```
> make clean
> make gtest.out
> ./gtest
```

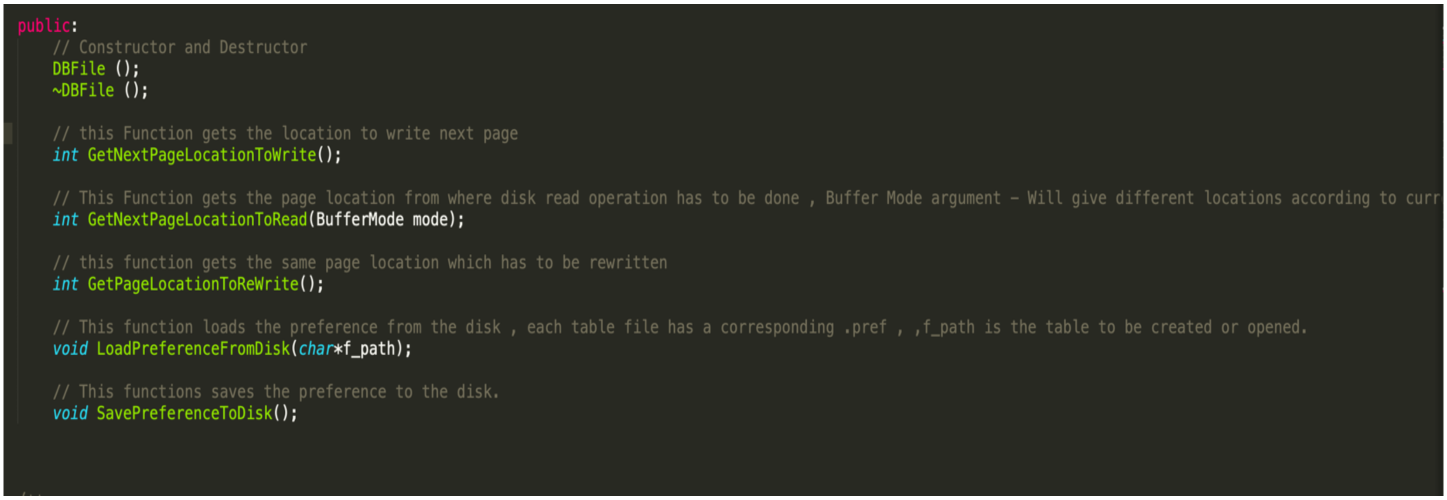
2) **Project files, structure and added functionality:**

The DBFile class has the following private class:



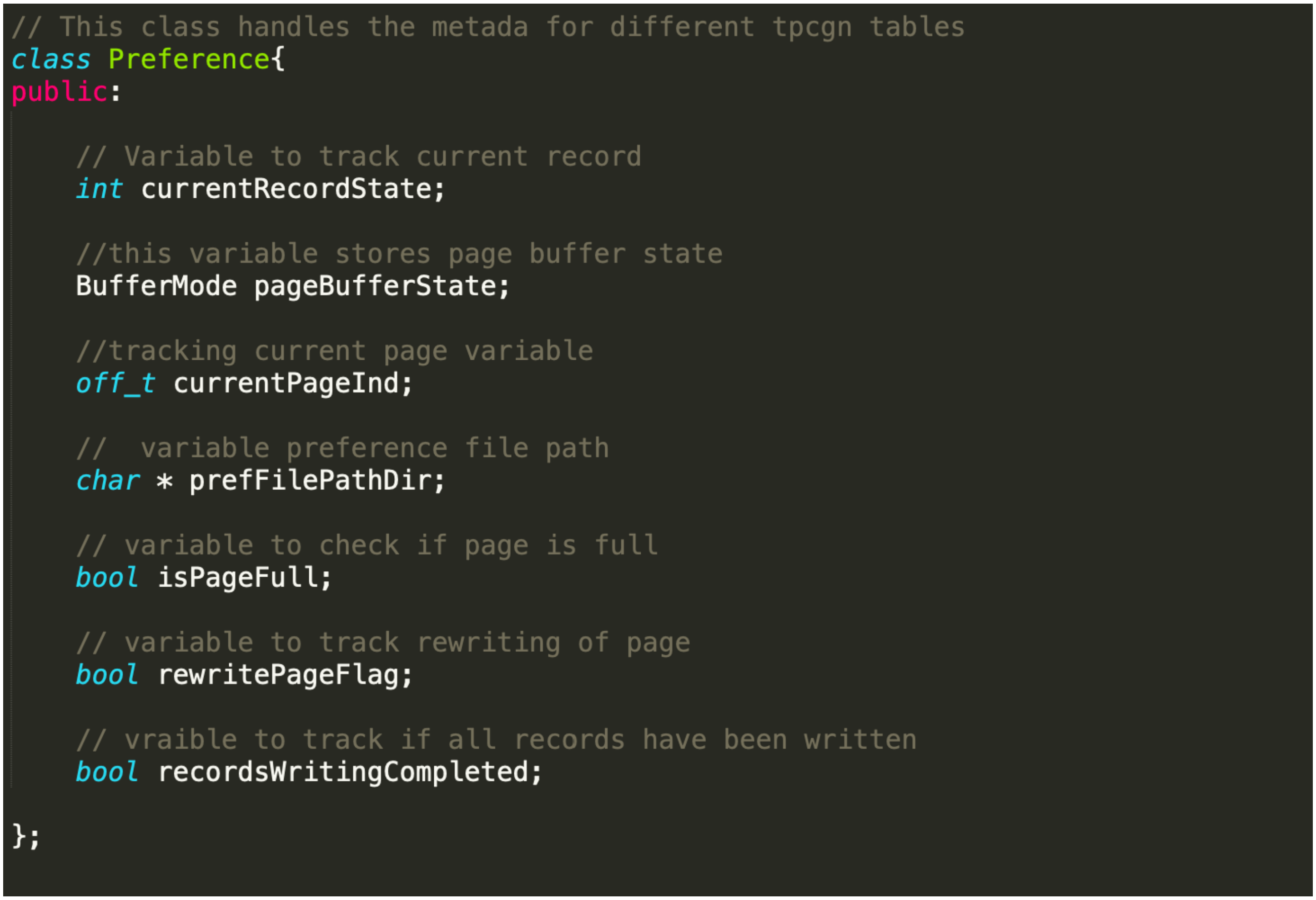
```
class DBFile {
private:
//   file var to keep pages
     File file;
//   page var to keep records
     Page page;

//   Used to keep track of the state.
     Preference preference;
//   Used to keep track of the state.
     ComparisonEngine comparisonEngine;
```

Public Functions added apart from the required one.



```
public:
    // Constructor and Destructor
    DBFile ();
    ~DBFile ();

    // this Function gets the location to write next page
    int GetNextPageLocationToWrite();

    // This Function gets the page location from where disk read operation has to be done , Buffer Mode argument - Will give different locations according to curr
    int GetNextPageLocationToRead(BufferMode mode);

    // this function gets the same page location which has to be rewritten
    int GetPageLocationToReWrite();

    // This function loads the preference from the disk , each table file has a corresponding .pref , ,f_path is the table to be created or opened.
    void LoadPreferenceFromDisk(char*f_path);

    // This functions saves the preference to the disk.
    void SavePreferenceToDisk();
```

The DBFile creates a preference file to store metadat for each .tbl file created using the create function. The preference class looks like below.



```
// This class handles the metada for different tpcgn tables
class Preference{
public:

    // Variable to track current record
    int currentRecordState;

    //this variable stores page buffer state
    BufferMode pageBufferState;

    //tracking current page variable
    off_t currentPageInd;

    //  variable preference file path
    char * prefFilePathDir;

    // variable to check if page is full
    bool isPageFull;

    // variable to track rewriting of page
    bool rewritePageFlag;

    // vraible to track if all records have been written
    bool recordsWritingCompleted;

};
```

Different DBfile Functions:

```
1) int Create (const char *fpath, fType file_type, void *startup);
```

This function creates the binary files. This function takes three arguments the first one is the file path, second one is the file type which tells what type of file it is, three possible values are: heap, sorted, and tree. in this assignment we are dealing with heap files only. This function after validating the file type, proceeds to check whether a .bin file already exists at the same file location with the same name or not and exits the flow it that is the case. After the check is complete, the the function uses the myFile object to call its open function with the params 0 and file path to signal a new file creation. the actual database data using the File class from File.h.The metadat file is saved using .pref extension. The last parameter to Create is a dummy parameter that you won't use for this assignment, but you will use for assignment two. The return value from Create is a 1 on success and a zero on failure.

```
2) int Open (const char *fpath);
```

This function looks for the existence of a file at the provided fpath. The one parameter to this function is simply the physical location of the file. The return value is a 1 on success and a zero on failure.The it proceeds to open the file using myFile object by passing it 1 as param to signify not to recreate the file. The function proceeds to load the last state of the DBFile using the load preference function. If the last state was READ, the it loads the page buffer with last read in page with the help of the function GetPageLocationToRead which determines the page location based on the variable currentPage present in myPreference object. It also removes the unnecessary records which were already read earlier. In case of WRITE mode, the function loads the page buffer with the last page which was written. Return 1 on success, 0 on failure.

```
3) int GetNext (Record &fetchme);
```

This GenNext method version fetches the next record from the file. The nexrt record's position is defined with respect to current position of the file pointer where it is currently pointing.This method will fetch the subsequent Record with respct to the pointer position.After the function call returns, the pointer into the file is incremented, so a subsequent call to GetNext won't return the same record twice. If next record is found and returnned an integer value 1 is returnnedotherwise an integer value 0 is returnned.

```
4) int GetNext (Record &fetchme, CNF &cnf, Record &literal);
```

The next version of GetNext also accepts a selection predicate (this is a conjunctive normal form expression). It returns the record in the file that is accepted by the selection predicate. The literal record is used to check the selection predicate and is created when the parse tree for the CNF is processed.

```
5) void Load (Schema &myschema, const char *loadpath);
```

The Load function bulk loads the DBFile instance from a text file, appending new data to it ,using the SuckNextRecord function from Record.h. The character string passed to Load is the name of the data file to bulk load.

```
6) void Add (Record &addme);
```

This function "Add" adds records to end of the file, there are then two functions that allow for record retrieval from a DBFile instance; all are called GetNext.

```
7) int Close ();
```

Close simply closes the file. The return value is a 1 on success and a zero on failure.it return 1 on a successful close or 0 otherwise.

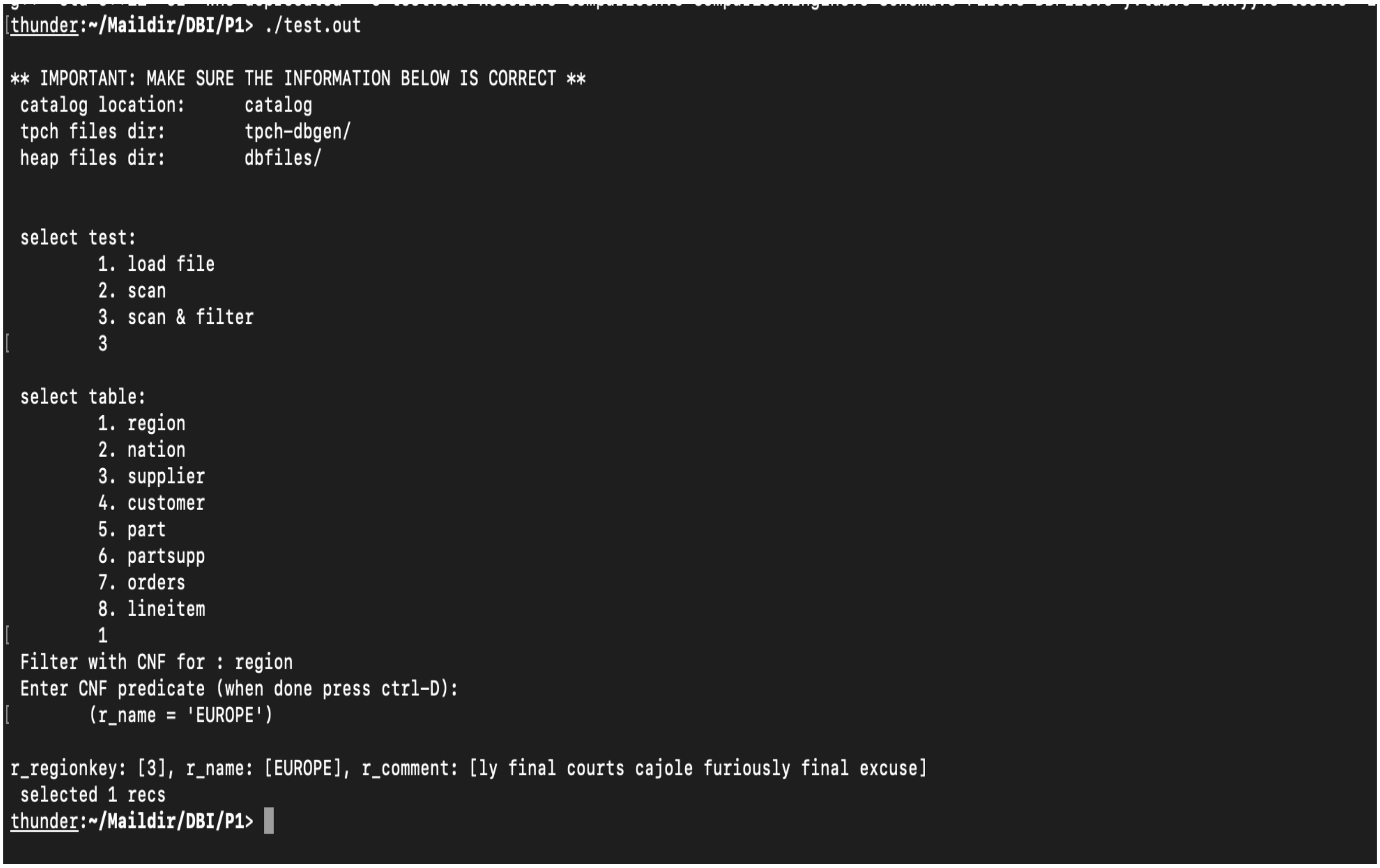
```
8) void MoveFirst ();
```

Each DBfile constitutes a pointer which points to the first record in the file. This pointer can move with respect to retrievals of records from the file. This function Forces the pointer to reset its position to the first record of the file

3) **Program Run Results:** All the queries are runs on 1GB version of tpchgen data. All the screenshots of all program runs like main, test.out (Load, Scan, Scan and filter), googletest run are included in the screenshots folder. This folder contains every single screenshot of all results obtained. In this report I am attaching results for specifically asked query runs.

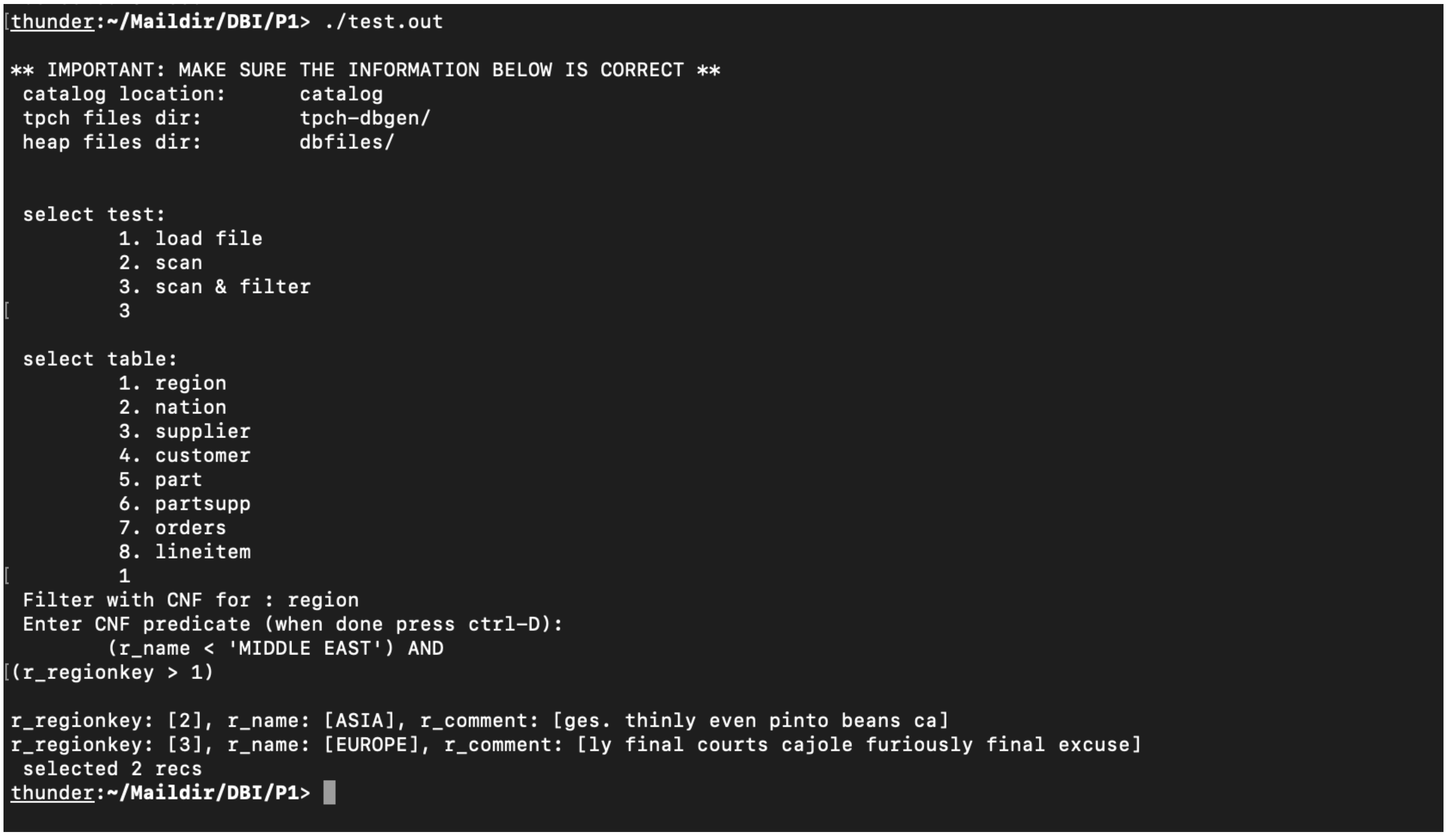
Result Screenshots for Test.out Runs:

1) Q1 Query : run on region table.



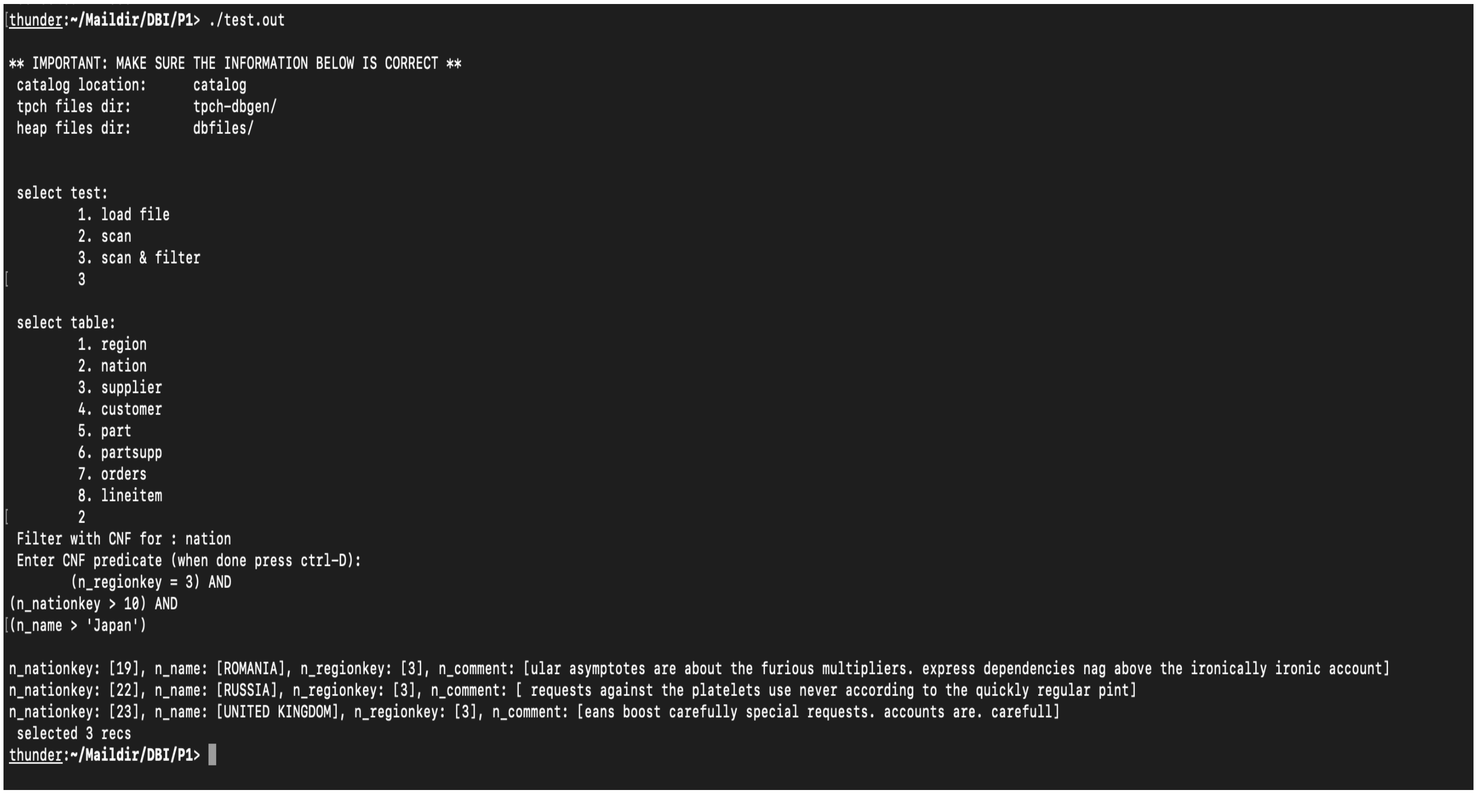
```
thunder:~/Maildir/DBI/P1> ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        tpch-dbgen/
 heap files dir:        dbfiles/


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

 select table:
        1. region
        2. nation
        3. supplier
        4. customer
        5. part
        6. partsupp
        7. orders
        8. lineitem
        1
 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
thunder:~/Maildir/DBI/P1> 
```

2) Q2 Query : run on region table.



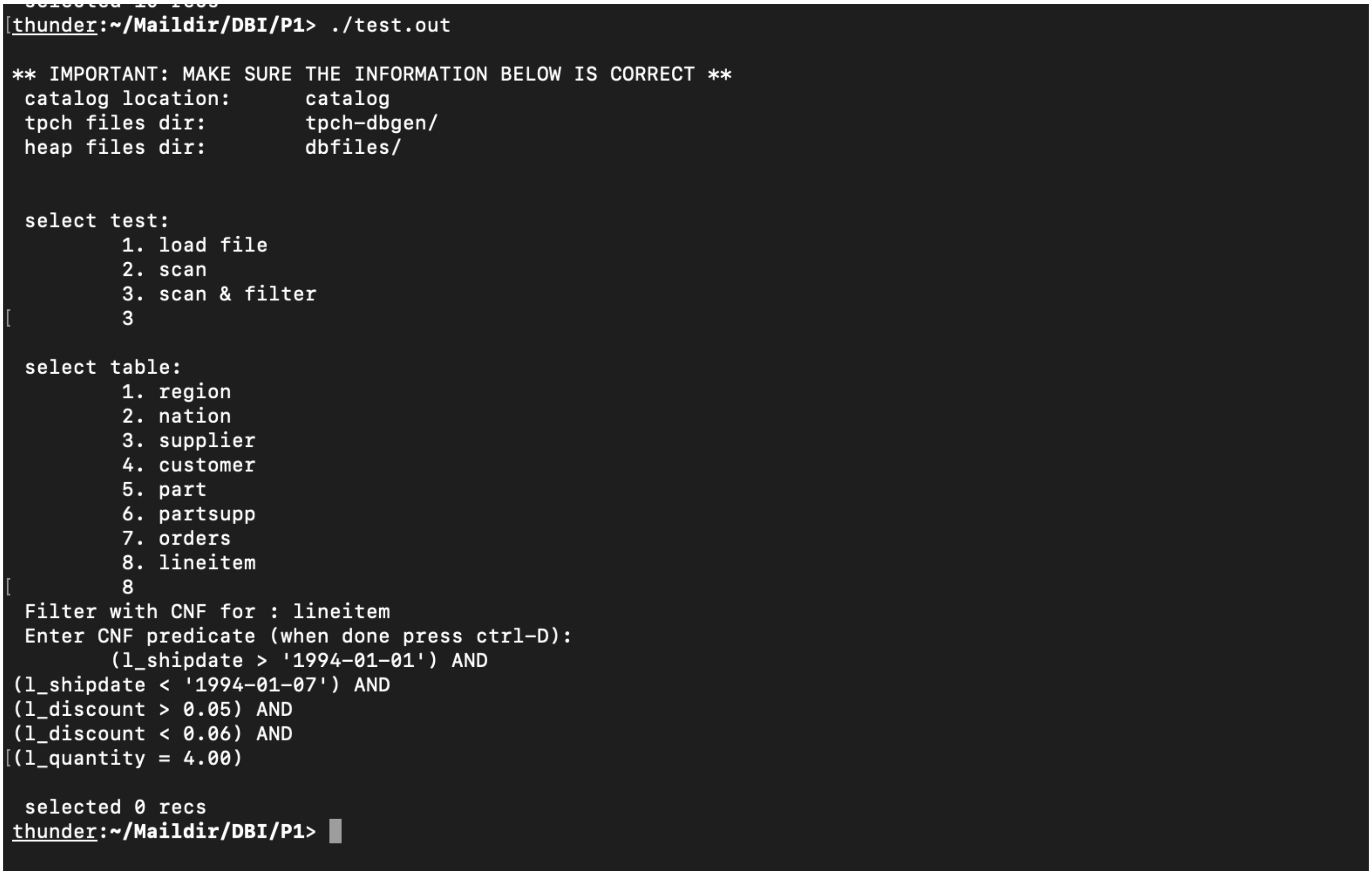
```
[thunder:~/Maildir/DBI/P1> ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        tpch-dbgen/
 heap files dir:        dbfiles/


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

 select table:
         1. region
         2. nation
         3. supplier
         4. customer
         5. part
         6. partsupp
         7. orders
         8. lineitem
[        1
 Filter with CNF for : region
 Enter CNF predicate (when done press ctrl-D):
        (r_name < 'MIDDLE EAST') AND
[(r_regionkey > 1)

r_regionkey: [2], r_name: [ASIA], r_comment: [ges. thinly even pinto beans ca]
r_regionkey: [3], r_name: [EUROPE], r_comment: [ly final courts cajole furiously final excuse]
 selected 2 recs
thunder:~/Maildir/DBI/P1> 
```

3) Q3 Query : Run on Nation table.



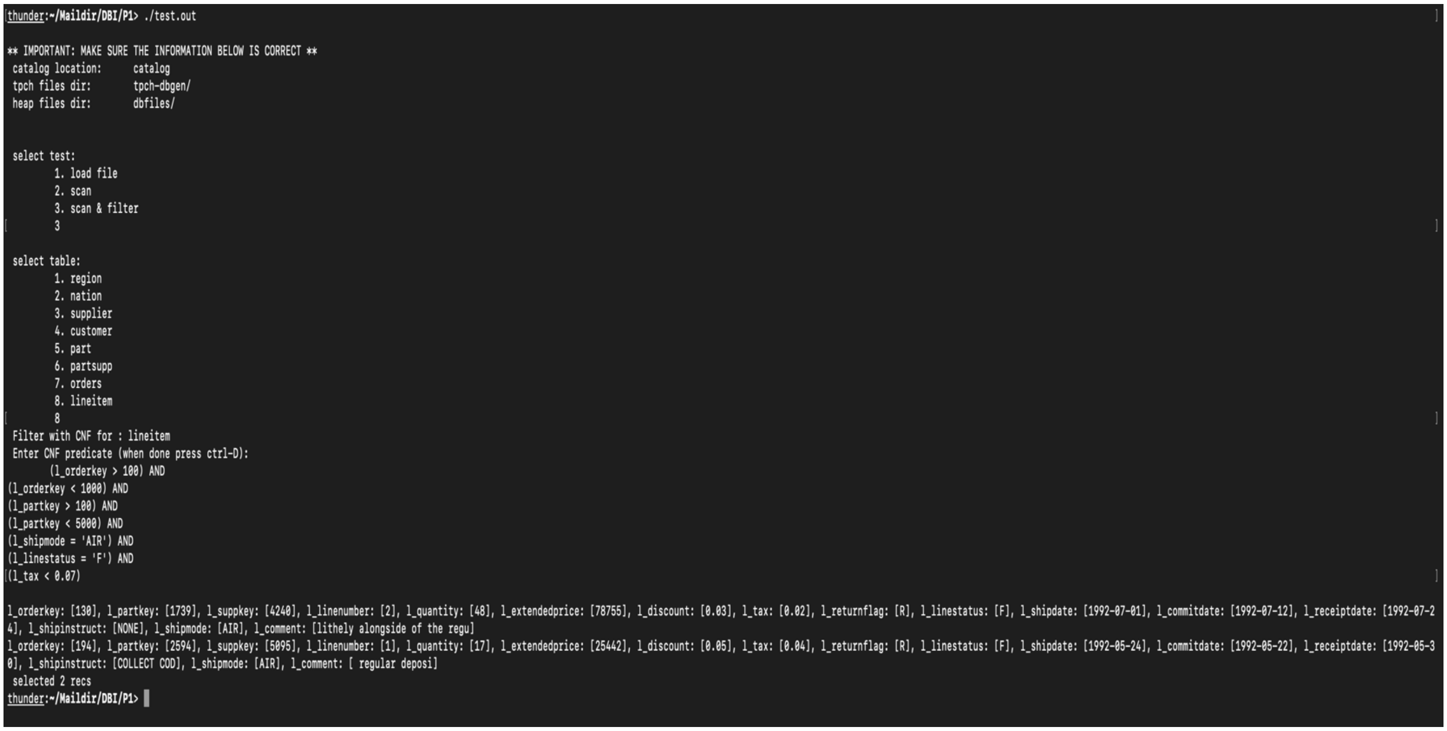
```
[thunder:~/Maildir/DBI/P1> ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        tpch-dbgen/
 heap files dir:        dbfiles/


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

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

n_nationkey: [19], n_name: [ROMANIA], n_regionkey: [3], n_comment: [ular asymptotes are about the furious multipliers. express dependencies nag above the ironically ironic account]
n_nationkey: [22], n_name: [RUSSIA], n_regionkey: [3], n_comment: [ requests against the platelets use never according to the quickly regular pint]
n_nationkey: [23], n_name: [UNITED KINGDOM], n_regionkey: [3], n_comment: [eans boost carefully special requests. accounts are. carefull]
 selected 3 recs
thunder:~/Maildir/DBI/P1> 
```

4) Q11 Query : run on Lineitem table.



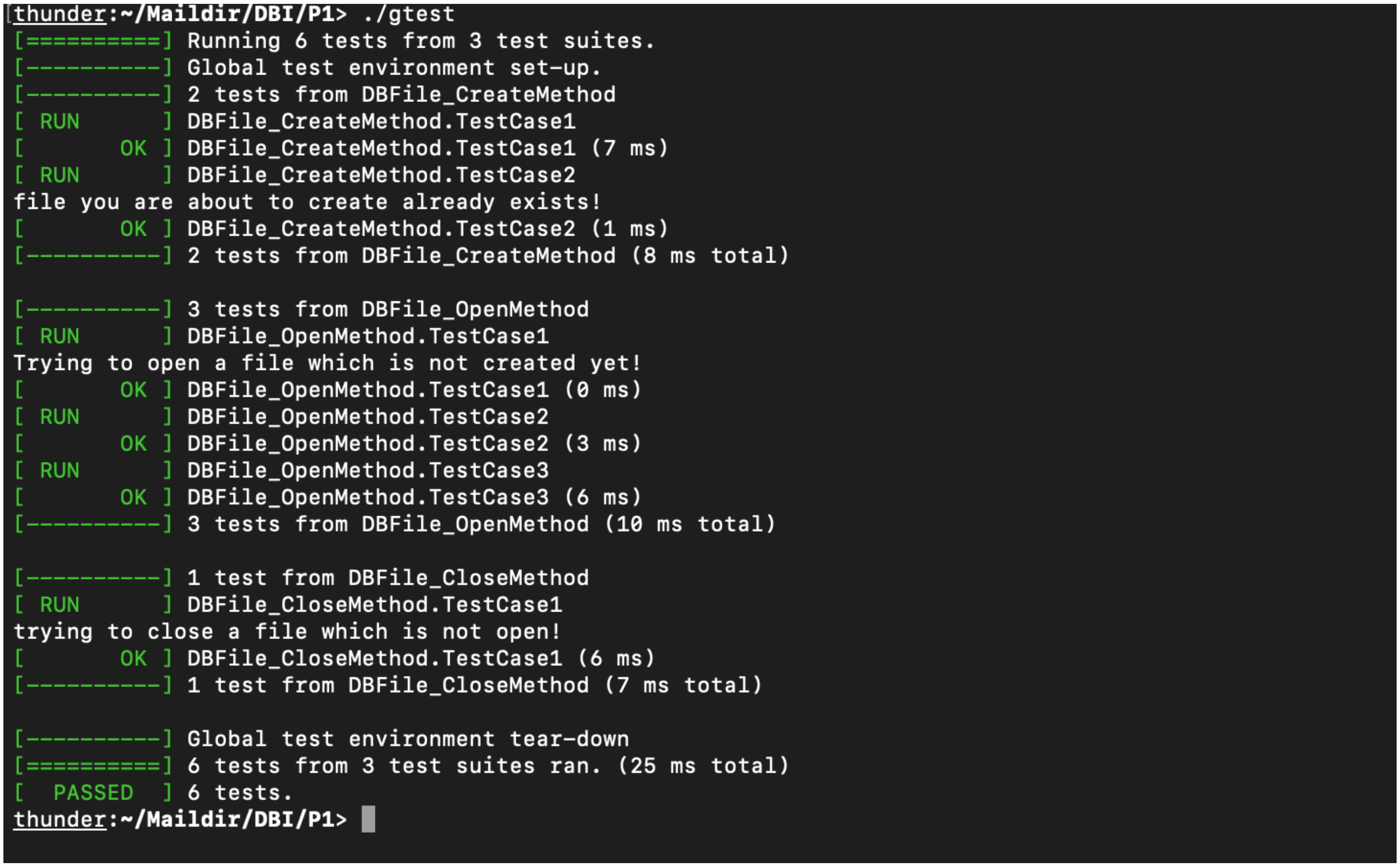
```
 selected 10 recs
[thunder:~/Maildir/DBI/P1> ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        tpch-dbgen/
 heap files dir:        dbfiles/


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

 select table:
         1. region
         2. nation
         3. supplier
         4. customer
         5. part
         6. partsupp
         7. orders
         8. lineitem
[        8
 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)

 selected 0 recs
thunder:~/Maildir/DBI/P1> 
```

5) Q12 Query: Run on lineitem table.



```
[thunder:~/Maildir/DBI/P1> ./test.out

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        tpch-dbgen/
 heap files dir:        dbfiles/


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

 select table:
         1. region
         2. nation
         3. supplier
         4. customer
         5. part
         6. partsupp
         7. orders
         8. lineitem
[        8
 Filter with CNF for : lineitem
 Enter CNF predicate (when done press ctrl-D):
        (l_orderkey > 100) AND
(l_orderkey < 1000) AND
(l_partkey > 100) AND
(l_partkey < 5000) AND
(l_shipmode = 'AIR') AND
(l_linestatus = 'F') AND
[(l_tax < 0.07)

l_orderkey: [130], l_partkey: [1739], l_suppkey: [4240], l_linenumber: [2], l_quantity: [48], l_extendedprice: [78755], l_discount: [0.03], l_tax: [0.02], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1992-07-01], l_commitdate: [1992-07-12], l_receiptdate: [1992-07-2
4], l_shipinstruct: [NONE], l_shipmode: [AIR], l_comment: [lithely alongside of the regu]
l_orderkey: [194], l_partkey: [2594], l_suppkey: [5095], l_linenumber: [1], l_quantity: [17], l_extendedprice: [25442], l_discount: [0.05], l_tax: [0.04], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1992-05-24], l_commitdate: [1992-05-22], l_receiptdate: [1992-05-3
0], l_shipinstruct: [COLLECT COD], l_shipmode: [AIR], l_comment: [ regular deposi]
 selected 2 recs
thunder:~/Maildir/DBI/P1> 
```

4) **ScreenShots of Gtest Results :** All the google tests are written and tested for region.bin and region.pref files.Before running the gtest file please delete these files in the heap file directory if these files are already created.



```
[thunder:~/Maildir/DBI/P1> ./gtest
[==========] Running 6 tests from 3 test suites.
[----------] Global test environment set-up.
[----------] 2 tests from DBFile_CreateMethod
[ RUN      ] DBFile_CreateMethod.TestCase1
[       OK ] DBFile_CreateMethod.TestCase1 (7 ms)
[ RUN      ] DBFile_CreateMethod.TestCase2
file you are about to create already exists!
[       OK ] DBFile_CreateMethod.TestCase2 (1 ms)
[----------] 2 tests from DBFile_CreateMethod (8 ms total)

[----------] 3 tests from DBFile_OpenMethod
[ RUN      ] DBFile_OpenMethod.TestCase1
Trying to open a file which is not created yet!
[       OK ] DBFile_OpenMethod.TestCase1 (0 ms)
[ RUN      ] DBFile_OpenMethod.TestCase2
[       OK ] DBFile_OpenMethod.TestCase2 (3 ms)
[ RUN      ] DBFile_OpenMethod.TestCase3
[       OK ] DBFile_OpenMethod.TestCase3 (6 ms)
[----------] 3 tests from DBFile_OpenMethod (10 ms total)

[----------] 1 test from DBFile_CloseMethod
[ RUN      ] DBFile_CloseMethod.TestCase1
trying to close a file which is not open!
[       OK ] DBFile_CloseMethod.TestCase1 (6 ms)
[----------] 1 test from DBFile_CloseMethod (7 ms total)

[----------] Global test environment tear-down
[==========] 6 tests from 3 test suites ran. (25 ms total)
[  PASSED  ] 6 tests.
thunder:~/Maildir/DBI/P1>
```