Report Assignment 1:
Implementing a Heap File

Group Members:

1. Dhiraj Mahesh Paryani, UFID: 1692 1261
2. Yogesh Laxman, UFID: 9451 2517

Instructions for Execution/ Code compilation and run tests:

a. Extract the contents of the folder.
b. Open the terminal and navigate to the extracted folder.
c. Execute the following commands. Please note that the tpch files should be generated using the dbgen program before execution. Please change the tpch, dbfile output and catalog directories if necessary in the test.cc file
   I. $ make clean
   II. $ make test.out – Compiles the test driver.
   III. $ ./test.out – Runs the test driver.

d. This would give a menu-based interface the performs the following three options that can test the code:
   1. load (read a tpch file and outputs a binary heap DBFile)
   2. scan (read records from an existing heap DBFile)
   3. scan & filter (read records from an existing heap DBFile and filter using a CNF predicate)

Brief explanation of the methods implemented and how it works:

We have used two separate read and write page buffers to read and write from the DBFile. The index of the pages within the DBFile on which read and write operations are being performed are updated constantly.

a. DBFile::Create:
   - This function uses the File->Open method to create a binary database file. We have given 0 as the length of File->Open function in order to create a .bin file.

b. DBFile::Load:
   - Every record from the given text file is extracted using SuckNextRecord function present within the Record class until end of file is reached. The Load function then uses a custom function addRecordToBuffer which adds every record to a page buffer.

c. DBFile::Open:
   - We have used File->Open method to open the binary database file. We have the assigned the read index to the page to be read and write index to the write buffer page inside the binary database file.

d. DBFile::MoveFirst:
   - This function will fetch the first page of the file and store it in the read buffer so as to point to the first record from the file.

e. DBFile::Close:
   - The close function would return 0 if the DBFile is not open. If write operation was being performed, the contents of the write buffer page are added to the DBFile using File->AddPage before closing the file using File->Close and returning 1.

f. DBFile::Add:
   - This function uses the custom addRecordToBuffer function to add the record provided to the write page buffer.

g. DBFile::GetNext (Record &fetchme):
   - It will fetch the next record.
   - We have used a custom readRecordFromBuffer function as a substitute for GetNext function.

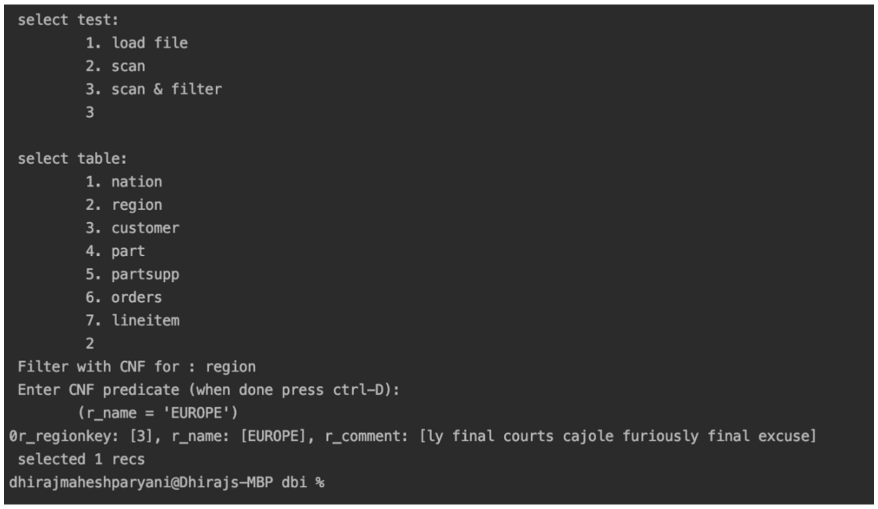
h. DBFile::GetNext (Record &fetchme, CNF &applyMe, Record &literal):
   - Checks if CNF is validated for every record which is read from the memory using the ComparisonEngine class. If the CNF function is validated then, the function uses the custom readRecordFromBuffer function to fetch the records from the buffer.

i. DBFile::addRecordToBuffer(Record &rec):
   - This function adds the given record to the end of the writeBufferPage. If no space is left in the current writeBufferPage, we add a new page to the file and increase the index of the current page being written.
   - If a situation arrives in which the index of writeBufferPage and readBufferPage are the same and write operation is being performed, we would add the record to both read and write buffer.

j. DBFile::readRecordFromBuffer (Record &rec):
   - This function fetches the next record from the readBufferPage. If readBufferPage is empty, then fetches next page from the file and returns next record from that readBufferPage. If all pages are already read it returns zero.
   - It returns a one if and only if there is a valid record return.

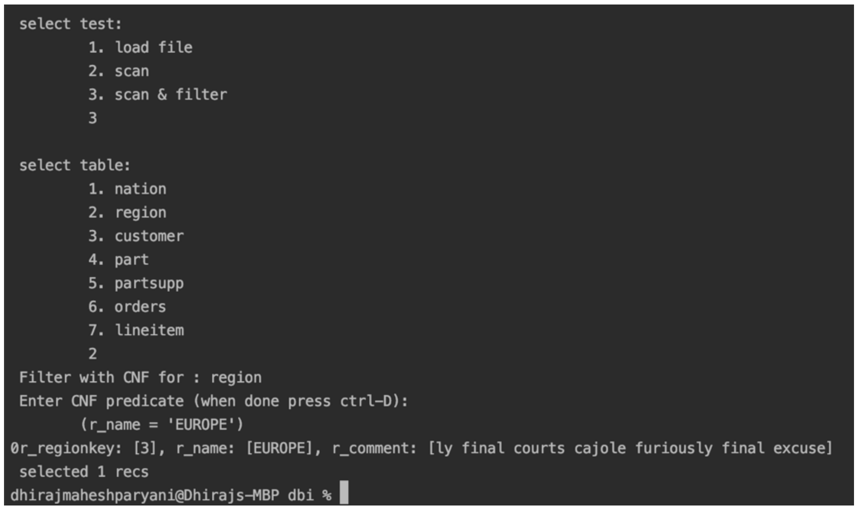
Test Cases and Output Screenshots:

a. q1 (r_name = 'EUROPE')



```
 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')
0r_regionkey: [3], r_name: [EUROPE], r_comment: [ly final courts cajole furiously final excuse]
 selected 1 recs
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

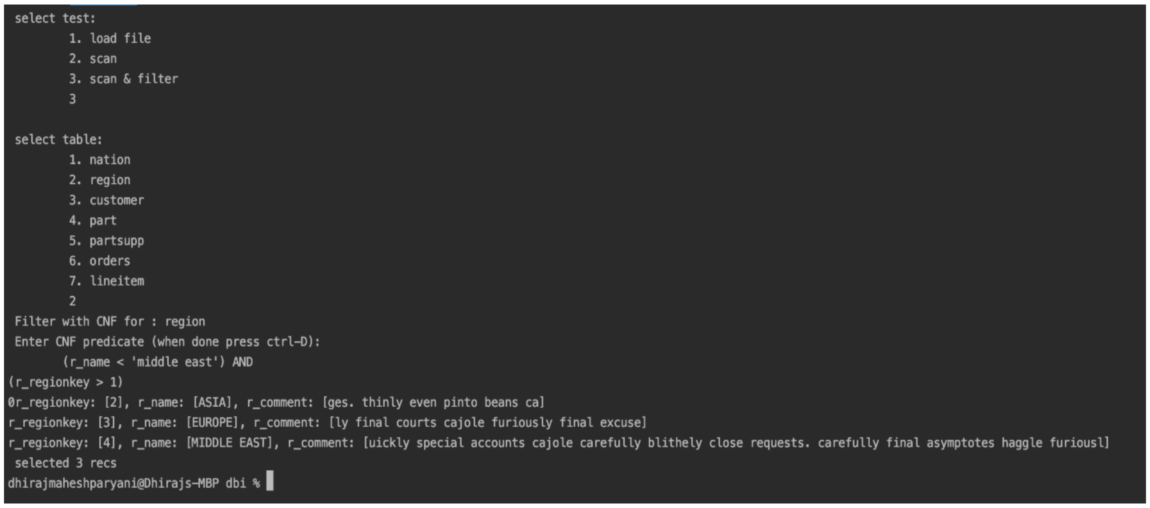
Figure 1: Using 10mb data (q1)



```
 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')
0r_regionkey: [3], r_name: [EUROPE], r_comment: [ly final courts cajole furiously final excuse]
 selected 1 recs
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

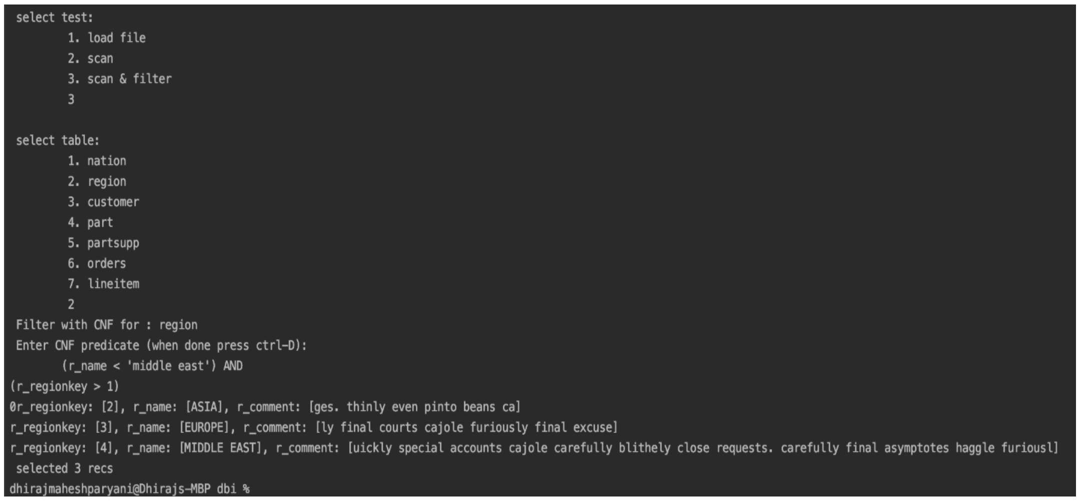
Figure 2: Using 1gb data (q1)

b. *q2 (r_name < 'MIDDLE EAST') AND (r_regionkey > 1*



```
 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 < 'middle east') AND
(r_regionkey > 1)
0r_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]
r_regionkey: [4], r_name: [MIDDLE EAST], r_comment: [uickly special accounts cajole carefully blithely close requests. carefully final asymptotes haggle furiousl]
 selected 3 recs
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

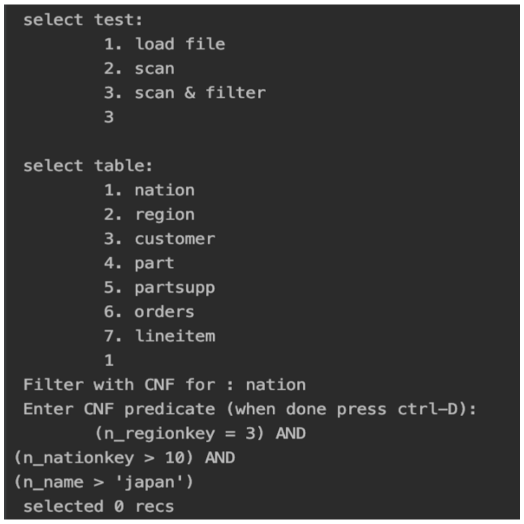
Figure 3: Using 10mb data (q2)



```
 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 < 'middle east') AND
(r_regionkey > 1)
0r_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]
r_regionkey: [4], r_name: [MIDDLE EAST], r_comment: [uickly special accounts cajole carefully blithely close requests. carefully final asymptotes haggle furiousl]
 selected 3 recs
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

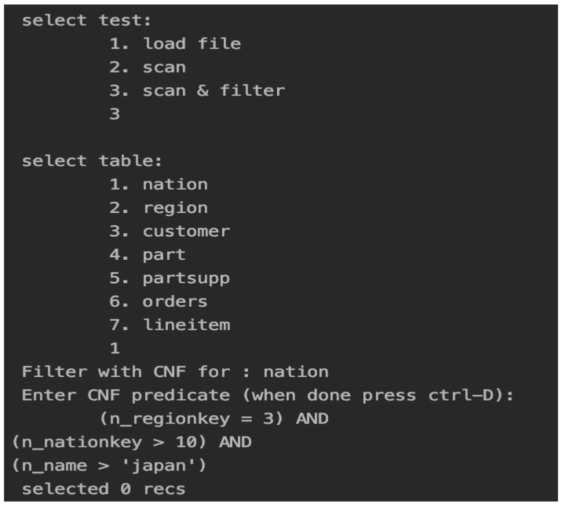
Figure 4: Using 1gb data (q2)

c. *(n_regionkey = 3) AND (n_nationkey > 10) AND (n_name > 'JAPAN')*



```
 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
```

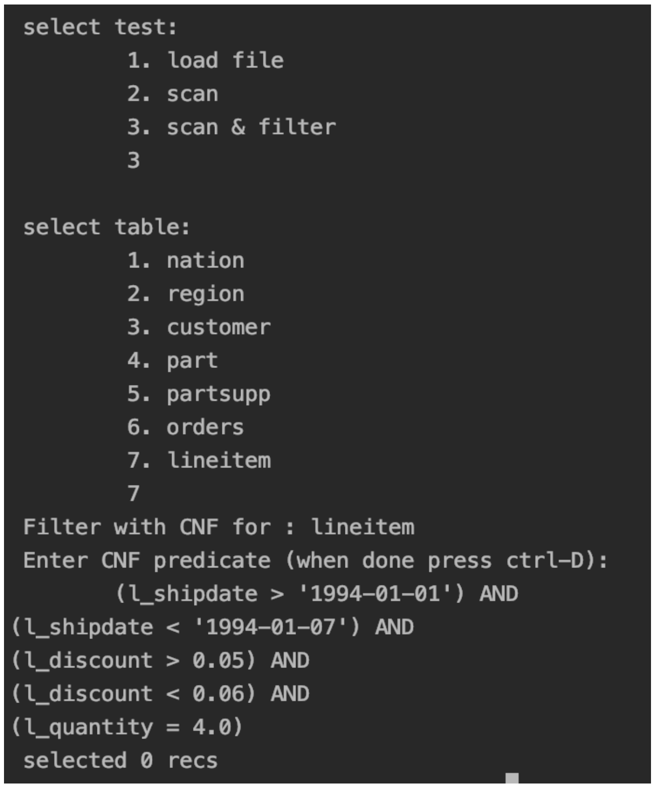
Figure 5: Using 10mb data (q3)



```
 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
```

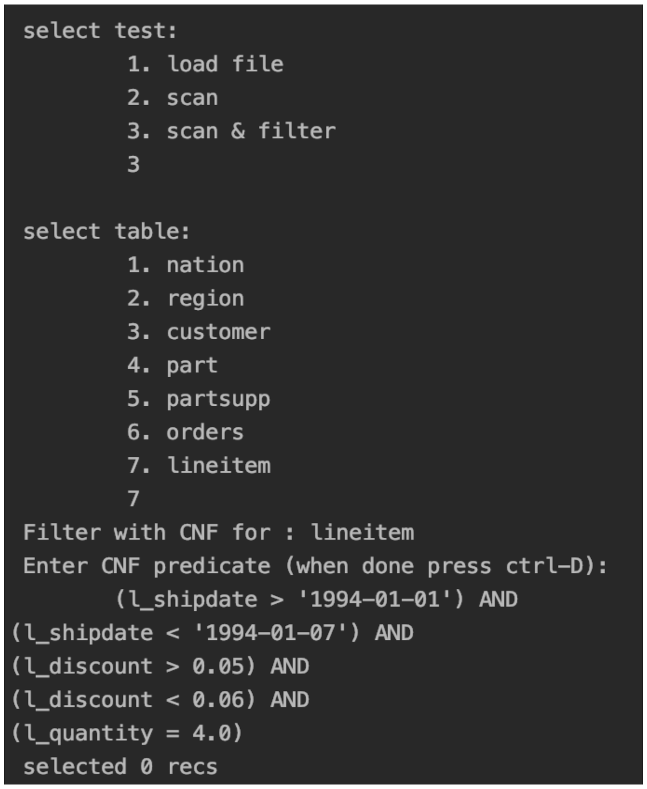
Figure 6: Using 1gb data (q3)

d. *q11 (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)*



```
 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 = 4.0)
 selected 0 recs
```

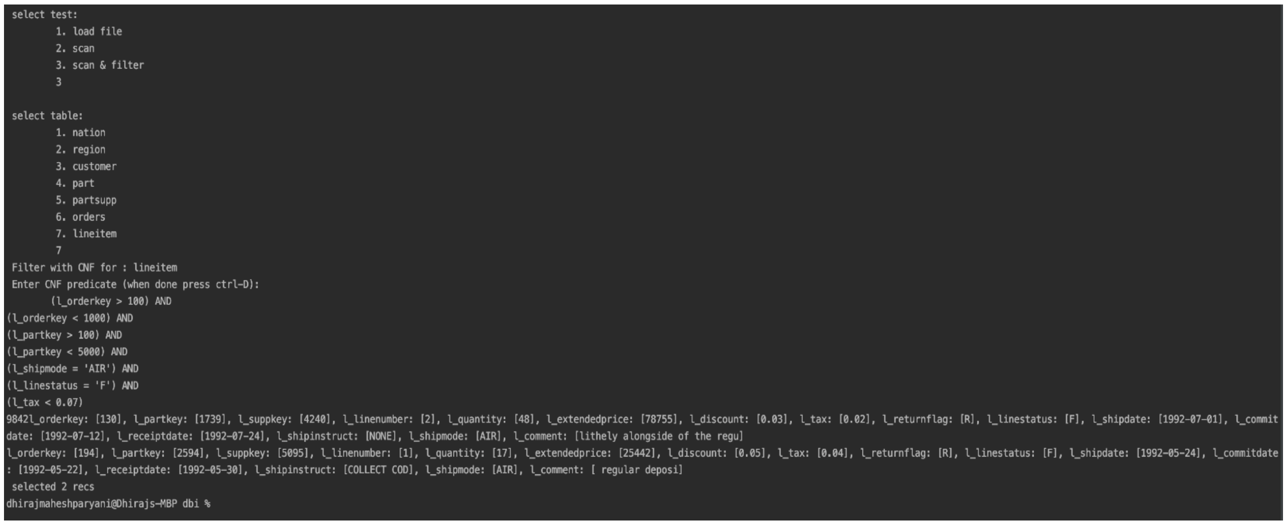
Figure 7: Using 10mb data (q11)



```
 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 = 4.0)
 selected 0 recs
```

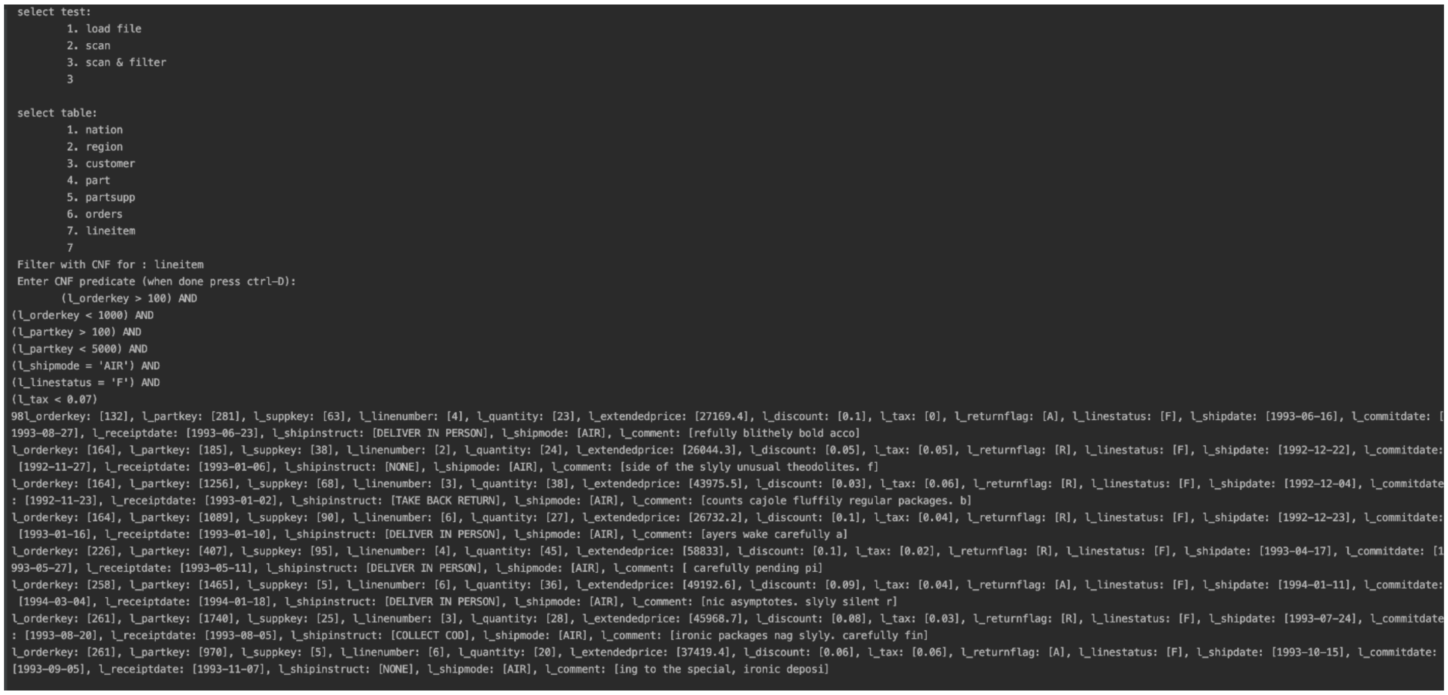
Figure 8: Using 1gb data (q11)

e. q12 (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)

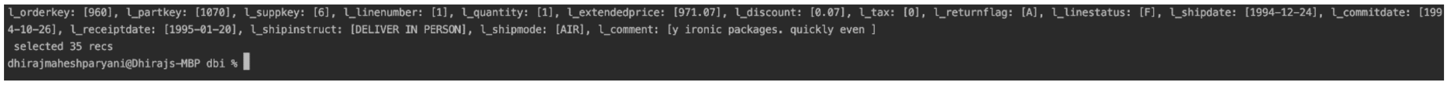


```
 select test:
        1. load file
        2. scan
        3. scan & filter
        3

 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_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)
9842l_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_commit
date: [1992-07-12], l_receiptdate: [1992-07-24], 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: [1992-05-30], l_shipinstruct: [COLLECT COD], l_shipmode: [AIR], l_comment: [ regular deposi]
 selected 2 recs
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

Figure 9: Using 10mb data(q12)



```
 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_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)
98l_orderkey: [132], l_partkey: [281], l_suppkey: [63], l_linenumber: [4], l_quantity: [23], l_extendedprice: [27169.4], l_discount: [0.1], l_tax: [0], l_returnflag: [A], l_linestatus: [F], l_shipdate: [1993-06-16], l_commitdate: [
1993-08-27], l_receiptdate: [1993-06-23], l_shipinstruct: [DELIVER IN PERSON], l_shipmode: [AIR], l_comment: [refully blithely bold acco]
l_orderkey: [164], l_partkey: [185], l_suppkey: [38], l_linenumber: [2], l_quantity: [24], l_extendedprice: [26044.3], l_discount: [0.05], l_tax: [0.05], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1992-12-22], l_commitdate:
 [1992-11-27], l_receiptdate: [1993-01-06], l_shipinstruct: [NONE], l_shipmode: [AIR], l_comment: [side of the slyly unusual theodolites. f]
l_orderkey: [164], l_partkey: [1256], l_suppkey: [68], l_linenumber: [3], l_quantity: [38], l_extendedprice: [43975.5], l_discount: [0.03], l_tax: [0.06], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1992-12-04], l_commitdate
: [1992-11-23], l_receiptdate: [1993-01-02], l_shipinstruct: [TAKE BACK RETURN], l_shipmode: [AIR], l_comment: [counts cajole fluffily regular packages. b]
l_orderkey: [164], l_partkey: [1089], l_suppkey: [90], l_linenumber: [6], l_quantity: [27], l_extendedprice: [26732.2], l_discount: [0.1], l_tax: [0.04], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1992-12-23], l_commitdate:
 [1993-01-16], l_receiptdate: [1993-01-10], l_shipinstruct: [DELIVER IN PERSON], l_shipmode: [AIR], l_comment: [ayers wake carefully a]
l_orderkey: [226], l_partkey: [407], l_suppkey: [95], l_linenumber: [4], l_quantity: [45], l_extendedprice: [58833], l_discount: [0.1], l_tax: [0.02], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1993-04-17], l_commitdate: [1
993-05-27], l_receiptdate: [1993-05-11], l_shipinstruct: [DELIVER IN PERSON], l_shipmode: [AIR], l_comment: [ carefully pending pi]
l_orderkey: [258], l_partkey: [1465], l_suppkey: [5], l_linenumber: [6], l_quantity: [36], l_extendedprice: [49192.6], l_discount: [0.09], l_tax: [0.04], l_returnflag: [A], l_linestatus: [F], l_shipdate: [1994-01-11], l_commitdate:
 [1994-03-04], l_receiptdate: [1994-01-18], l_shipinstruct: [DELIVER IN PERSON], l_shipmode: [AIR], l_comment: [nic asymptotes. slyly silent r]
l_orderkey: [261], l_partkey: [1740], l_suppkey: [25], l_linenumber: [3], l_quantity: [28], l_extendedprice: [45968.7], l_discount: [0.08], l_tax: [0.03], l_returnflag: [R], l_linestatus: [F], l_shipdate: [1993-07-24], l_commitdate
: [1993-08-20], l_receiptdate: [1993-08-05], l_shipinstruct: [COLLECT COD], l_shipmode: [AIR], l_comment: [ironic packages nag slyly. carefully fin]
l_orderkey: [261], l_partkey: [970], l_suppkey: [5], l_linenumber: [6], l_quantity: [20], l_extendedprice: [37419.4], l_discount: [0.06], l_tax: [0.06], l_returnflag: [A], l_linestatus: [F], l_shipdate: [1993-10-15], l_commitdate:
[1993-09-05], l_receiptdate: [1993-11-07], l_shipinstruct: [NONE], l_shipmode: [AIR], l_comment: [ing to the special, ironic deposi]
```



```
l_orderkey: [960], l_partkey: [1070], l_suppkey: [6], l_linenumber: [1], l_quantity: [1], l_extendedprice: [971.07], l_discount: [0.07], l_tax: [0], l_returnflag: [A], l_linestatus: [F], l_shipdate: [1994-12-24], l_commitdate: [199
4-10-26], l_receiptdate: [1995-01-20], l_shipinstruct: [DELIVER IN PERSON], l_shipmode: [AIR], l_comment: [y ironic packages. quickly even ]
 selected 35 recs
dhirajmaheshparyani@Dhirajs-MBP dbi % 
```

Figure 10: Using 1gb data (q12)

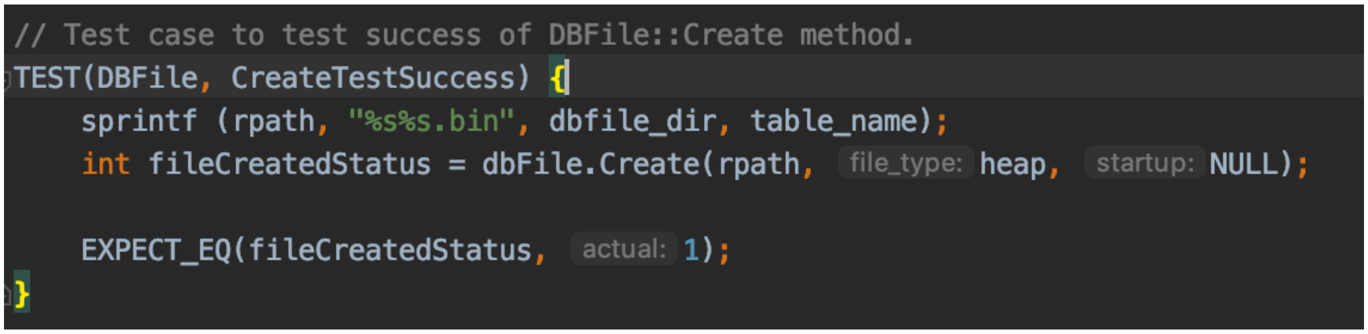
Instructions to run GTest:

I. $ make clean
II. $ make gtest.out
III. $ ./gtest.out

Screenshots of GTest results that match results generated by code:

a. CreateTestSuccess function:

This function tests the correctness of DBFile::Create function. This is a positive testing in which DBFile::Create function should return 1. This should match the actual status within the EXPECT_EQ method.

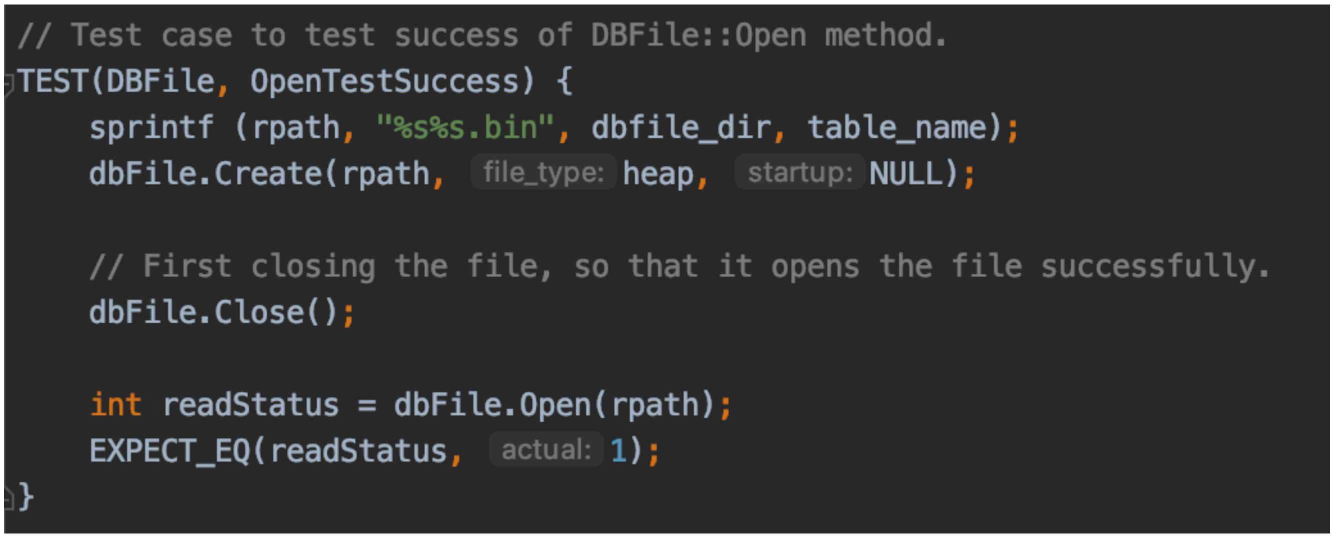


```
// Test case to test success of DBFile::Create method.
TEST(DBFile, CreateTestSuccess) {
    sprintf (rpath, "%s%s.bin", dbfile_dir, table_name);
    int fileCreatedStatus = dbFile.Create(rpath, file_type: heap, startup: NULL);

    EXPECT_EQ(fileCreatedStatus, actual: 1);
}
```

Figure 11: CreateTestSuccess (GTest)

b. OpenTestSuccess:

This function checks the functionality of DBFile::Open function. The DBFile::Open should return 1 if the function successfully opens the binary database file.

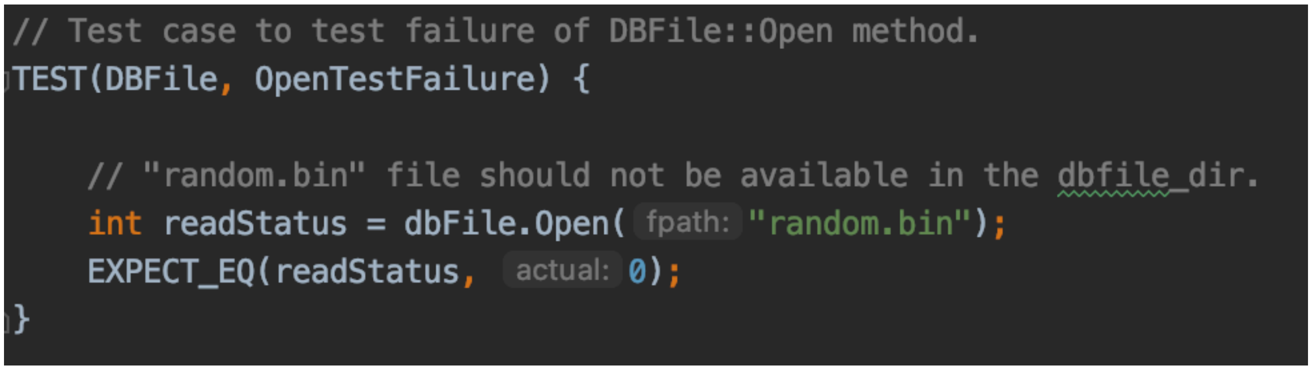


```
// Test case to test success of DBFile::Open method.
TEST(DBFile, OpenTestSuccess) {
    sprintf (rpath, "%s%s.bin", dbfile_dir, table_name);
    dbFile.Create(rpath,  file_type: heap,  startup: NULL);

    // First closing the file, so that it opens the file successfully.
    dbFile.Close();

    int readStatus = dbFile.Open(rpath);
    EXPECT_EQ(readStatus,  actual: 1);
}
```

Figure 12: OpenTestSuccess (GTest)

c. OpenTestFailure:

This function checks the functionality of DBFile::Open function. The DBFile::Open should return 0 if the function is not successful in opening the binary database file. We have passed the file path “random.bin” and our directory doesn’t contain that file. Hence it should give read status as 0 as file is not present.

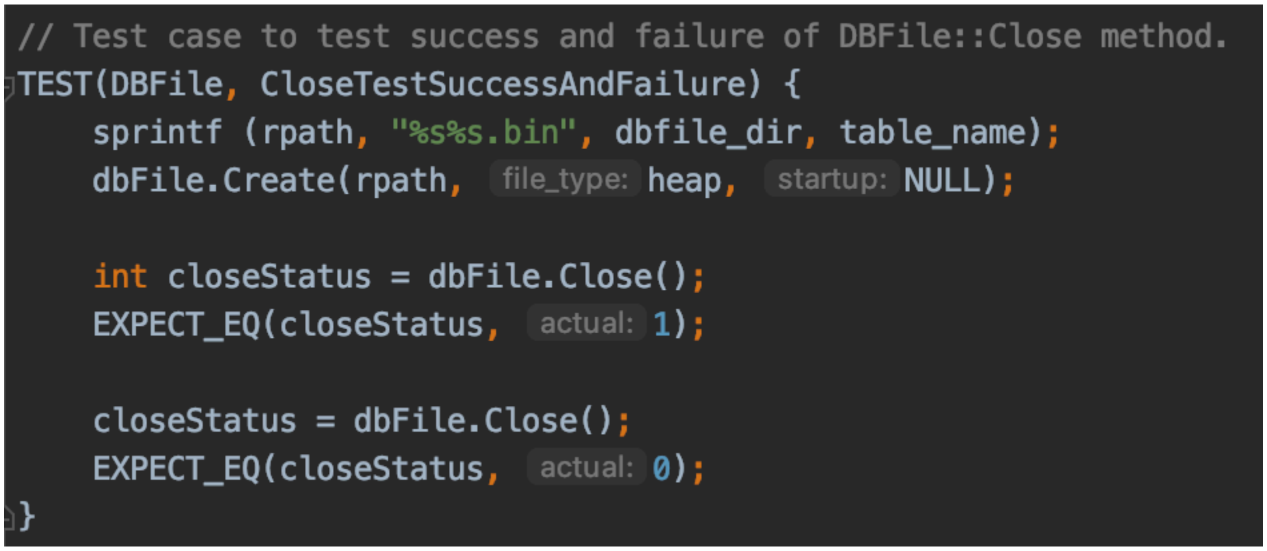


```
// Test case to test failure of DBFile::Open method.
TEST(DBFile, OpenTestFailure) {

    // "random.bin" file should not be available in the dbfile_dir.
    int readStatus = dbFile.Open( fpath: "random.bin");
    EXPECT_EQ(readStatus,  actual: 0);
}
```

Figure 13: OpenTestFailure (GTest)

d. CloseTestSuccessAndFailure:

This function checks the DBFile::Close method. It expects status as 1 if the file is open and 0 if the file is not open.

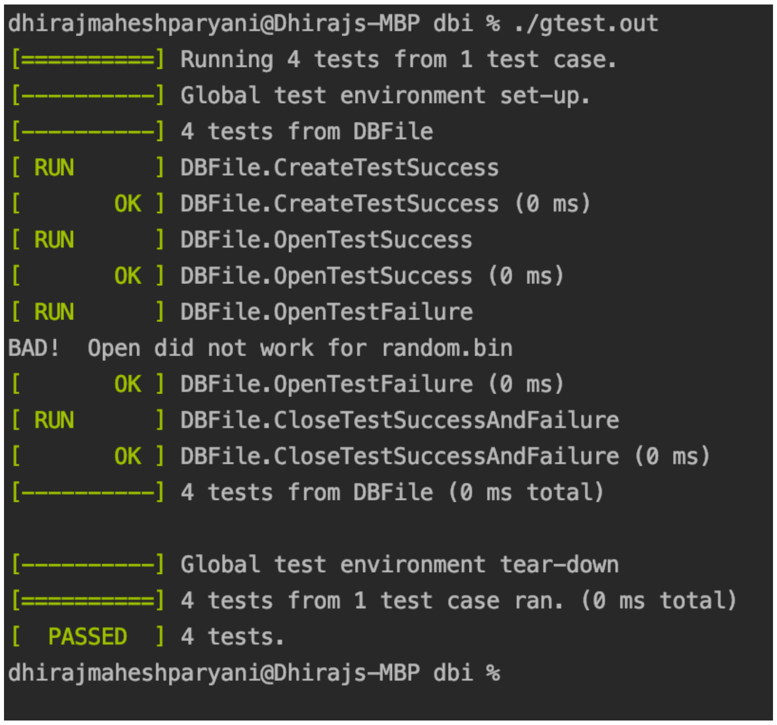


```
// Test case to test success and failure of DBFile::Close method.
TEST(DBFile, CloseTestSuccessAndFailure) {
    sprintf (rpath, "%s%s.bin", dbfile_dir, table_name);
    dbFile.Create(rpath,  file_type: heap,  startup: NULL);

    int closeStatus = dbFile.Close();
    EXPECT_EQ(closeStatus,  actual: 1);

    closeStatus = dbFile.Close();
    EXPECT_EQ(closeStatus,  actual: 0);
}
```

Figure 14: CloseTestSuccessAndFailure (GTest)

GTest Output:

The following screenshot shows that all the functions that were implemented within the DBFile are working as expected.



```
dhirajmaheshparyani@Dhirajs-MBP dbi % ./gtest.out
[==========] Running 4 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 4 tests from DBFile
[ RUN      ] DBFile.CreateTestSuccess
[       OK ] DBFile.CreateTestSuccess (0 ms)
[ RUN      ] DBFile.OpenTestSuccess
[       OK ] DBFile.OpenTestSuccess (0 ms)
[ RUN      ] DBFile.OpenTestFailure
BAD!  Open did not work for random.bin
[       OK ] DBFile.OpenTestFailure (0 ms)
[ RUN      ] DBFile.CloseTestSuccessAndFailure
[       OK ] DBFile.CloseTestSuccessAndFailure (0 ms)
[----------] 4 tests from DBFile (0 ms total)

[----------] Global test environment tear-down
[==========] 4 tests from 1 test case ran. (0 ms total)
[  PASSED  ] 4 tests.
dhirajmaheshparyani@Dhirajs-MBP dbi %
```

Figure 15: GTest Output