Database Implementation System
Project 2 Part 2

Group Member: Sathya Sai Ram Kumar(34891238) & Mohit Kalra (13906151)

Instruction to Compile and Run:

To run test.out:

To run test.out simply follow the steps as below:

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

The test executable has an inbuilt option to create sorted dbfiles which can be further be queried upon by subsequent options which are thrown for input at the user.

To run gtest:

> make clean
> make gtest
>./gtest < gtest_input.txt

Please note that “gtest_input.txt” contains the CNF statement needed for one of the tests.

Project Flow and Structure:

Generic DBFile Class

Second part of the assignment was mainly about creating a generic dbfile which could be used as both unsorted and sorted heap file. A virtual base class called genericDBFile was created which would be a common parent to both the heap and sorted dbfiles. Functions which were not common to heap, and sorted implementations were implemented as virtual functions which would be implemented in each of the classes uniquely.

```
class GenericDBFile{
protected:
    //  Used to read & write page to the disk.
    File myFile;
    //  Used as a buffer to read and write data.
    Page myPage;
    // Pointer to preference file
    Preference * myPreferencePtr;
    //  Used to keep track of the state.
    ComparisonEngine myCompEng;
public:
    GenericDBFile();
    int GetPageLocationToWrite();
    int GetPageLocationToRead(BufferMode mode);
    int GetPageLocationToReWrite();
    void Create (char * f_path,fType f_type, void *startup);
    int Open (char * f_path);

    //  virtual function
    virtual ~GenericDBFile();
    virtual void MoveFirst ();
    virtual void Add (Record &addme);
    virtual void Load (Schema &myschema, const char *loadpath);
    virtual int GetNext (Record &fetchme);
    virtual int GetNext (Record &fetchme, CNF &cnf, Record &literal);
    virtual int Close();
};
```

HeapDBFile Definition

All the functions defined in assignment 1 for unsorted DBFile were moved in the function definitions which were either common to a generic dbfile mentioned above or in the functions described below.

```
class HeapDBFile: public virtual GenericDBFile{
public:
    HeapDBFile(Preference * preference);
    ~HeapDBFile();
    void MoveFirst ();
    void Add (Record &addme);
    void Load (Schema &myschema, const char *loadpath);
    int GetNext (Record &fetchme);
    int GetNext (Record &fetchme, CNF &cnf, Record &literal);
    int Close ();

};
```

SortedDBFile Definition

All implementation new to this assignment was in the functions listed below. Below is a brief explanation for each of the class attributes and functions in the sortedDBFile Class.

```
class SortedDBFile: public  GenericDBFile{
    Pipe * inputPipePtr;
    Pipe * outputPipePtr;
    BigQ * bigQPtr;
    File newFile;
    Page outputBufferForNewFile;
    OrderMaker * queryOrderMaker;
    bool doBinarySearch;

public:
    SortedDBFile(Preference * preference);
    ~SortedDBFile();
    void MoveFirst ();
    void Add (Record &addme);
    void Load (Schema &myschema, const char *loadpath);
    int GetNext (Record &fetchme);
    int GetNext (Record &fetchme, CNF &cnf, Record &literal);
    int Close ();
    void MergeSortedInputWithFile();
    int BinarySearch(Record &fetchme, Record &literal,off_t low, off_t high);
};
```

class attributes

- Pipe * inputPipePtr
  Input pipe for pushing records into BigQ
- Pipe * outputPipePtr
  Output pipe for retrieving records from BigQ
- BigQ * bigQPtr
  BigQ for sorting records added / loaded into sorted dbfile
- File newFile
  File needed for merging input while switching modes from write to read

- Page outputBufferForNewFile
  Outputbuffer for writing into file being used for merging input. This'll become the new sorted DBFile itself.

- OrderMaker * queryOrderMaker
  QueryOrderMaker is used by GetNext function to construct a sort order with given CNF and sort order of dbfile.

- bool doBinarySearch;

Functions

- void MoveFirst ()

Movefirst implementation in sorted dbfile is like the one in heap file. If the dbfile is currently in write mode and the page buffer is nonempty, we first write the page buffer into dbfile, read the first page and fetch its first record. Otherwise, we just move to the first page and start reading.

- void Add (Record &addme)

This function adds records in the input pipe of the BigQ instance of the sorted dbfiile object. If the buffer was in read mode, then the buffer is first flushed and set in write mode for sorted records which will be coming in from output pipe of BigQ instance.

- void Load (Schema &myschema, const char *loadpath)

Load implements the exact same functionality as add except for doing it in bulk. Infact, Load implements a while loop over Add function while we are getting records from the raw .tbl file.

- int GetNext (Record &fetchme)

GetNext iteratively loops over all the records of the dbfile. Data is fetched page by page from the persistent file and read in record by record. This is run until all the records are read in which case GetNext returns a 0.

- int GetNext (Record &fetchme, CNF &cnf, Record &literal)

This version of GetNext is somewhat involved. We need to compare the given CNF with the existing sort order of the dbfile. We convert the CNF to an OrderMarker object using a function like GetSortOrders of CNF class. Once we have the OrderMaker object corresponding to the CNF, we start comparing the attributes of both the OrderMakers to see if we can accommodate the CNF with the sort order of the dbfile. If yes, we use the BinarySearch function (see below for details) to see if there's a record satisfying the CNF. We keep getting such records for each of the subsequent calls to this function.

- int Close ()

Close shuts the dbfile off after writing the state of the file in a .pref file (.pref stands for preference). In case of sorted dbfile we also write the ordermaker object into the state so that we know the order of sort of the file when we are reading it back.

- void MergeSortedInputWithFile()

This function comes in handy while we are switching from write to read mode. The records which are present in the BigQ instance need to be merged with the already sorted file. This function merges the 2 sorted sources together which is exactly like a merge step. A temporary file is used for writing the merge stream which is renamed to be the original dbfile we are working for in this case.

- int BinarySearch(Record &fetchme, Record &literal,off_t low, off_t high)

This function is an auxiliary function to the GetNext function which accepts a CNF. We apply binary search on the pages of the dbfile. We compare the first record for the given CNF. If the record matches, we scan through previous

pages until the CNF is satisfied and return the first record of the order. BinarySearch returns a 0 if no record satisfying the CNF is found.

Results:

Test case 1

```
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /home/mk/Documents/uf docs/sem 2/Database Implementation/10_git/tpch-dbgen/
 heap files dir:        dbfiles/


 select test option:
        1. create sorted dbfile
        2. scan a dbfile
        3. run some query
        1

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

 specify sort ordering (when done press ctrl-D):
        (c_phone)
NumAtts =     0
NumAtts =     0
NumAtts =     1
  0:     4 String

 specify runlength:
        8

 output to dbfile : dbfiles/customer.bin
 input from file : /home/mk/Documents/uf docs/sem 2/Database Implementation/10_git/tpch-dbgen/customer.tbl

 select option for : dbfiles/customer.bin
        1. add a few (1 to 1k recs)
        2. add a lot (1k to 1e+06 recs)
        3. run some query
        2
        .......
        added 150000 recs..so far 150000

 create finished.. 150000 recs inserted
```

Test case 2

```
.c_custkey: [12022], c_name: [Customer#000012022], c_address: [PwVDMfqZtKWOo], c_nationkey: [24], c_phone: [34-998
uickly regular ideas cajole blithely]
.c_custkey: [125210], c_name: [Customer#000125210], c_address: [BJiJAg79TvSTdxgPeb63sUqKaOVv], c_nationkey: [24],
nts. theodolites among the slyly even patterns nag slyly fluffi]
.c_custkey: [50667], c_name: [Customer#000050667], c_address: [9aGxwCYLcG6y2eA], c_nationkey: [24], c_phone: [34-9
tions boost furi]
.c_custkey: [7288], c_name: [Customer#000007288], c_address: [,dZCqJe94SqXQjxAd7WxRMYyFVKGd4pFIrAshk], c_nationkey
sly. carefully ironic platelets grow furiously. ironic, bold theodolites hang above th]
.c_custkey: [84642], c_name: [Customer#000084642], c_address: [MJerprOnUYRK], c_nationkey: [24], c_phone: [34-999-
layers sleep blithely after the blithely even deposits. carefully regula]
.c_custkey: [148611], c_name: [Customer#000148611], c_address: [HoNNKAELd2vMNyGWpw3ZZgSNaDs7fOdNIH64], c_nationkey
e blithely pending asymptotes sleep carefully about the slyly even excu]
.c_custkey: [102689], c_name: [Customer#000102689], c_address: [uPFxngvSOaDQIL], c_nationkey: [24], c_phone: [34-9
y according to the furiously bold courts. daring ]
.c_custkey: [35478], c_name: [Customer#000035478], c_address: [Ot5zcyiiLhzXXivX4Y2FeToQIh,iovs6zyA], c_nationkey:
ly ironic packages. daring, even de]
.c_custkey: [103505], c_name: [Customer#000103505], c_address: [XPNXpdCD7HRqcV8ABpjF3ixDc9A6X ZB2 ], c_nationkey:
s boost. quickly ironic requests could have to are according to the final accounts. final]
.c_custkey: [62707], c_name: [Customer#000062707], c_address: [bRbtSTIXhh], c_nationkey: [24], c_phone: [34-999-47
ans. bold, fina]
.c_custkey: [41774], c_name: [Customer#000041774], c_address: [DpHfJFKVkr4LQeif9t5SSKzDUa4G3asKeya5M7dz], c_nation
leep. quickly silent requests wake fluffily. furiously close packages nag furiously across the caref]
.c_custkey: [94308], c_name: [Customer#000094308], c_address: [5aE9PLelngeVg8v6KCgOyR,NhDT5uC4], c_nationkey: [24]
efully unus]
.c_custkey: [102414], c_name: [Customer#000102414], c_address: [Xzg9eiWbsxiQW,LUCyUAcPoOENdLUQt], c_nationkey: [24
 carefully even packages by the permanently special gifts ne]

 scanned 150000 recs
```

Test case 3

```
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /home/mk/Documents/uf docs/sem 2/Database Implementation/10_git/tpch-dbgen/
 heap files dir:        dbfiles/


 select test option:
        1. create sorted dbfile
        2. scan a dbfile
        3. run some query
        3

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

 enter CNF predicate (when done press ctrl-D):
        (c_phone > '34-999-195-7029') AND (c_mktsegment = 'FURNITURE')
       c_custkey: [103505], c_name: [Customer#000103505], c_address: [XPNXpdCD7HRqcV8ABpjF3ixDc9A6X ZB2 ], c_nationkey: [24],
deposits boost. quickly ironic requests could have to are according to the final accounts. final]
c_custkey: [62707], c_name: [Customer#000062707], c_address: [bRbtSTIXhh], c_nationkey: [24], c_phone: [34-999-473-9448], c_ac
ns. bold, fina]
c_custkey: [41774], c_name: [Customer#000041774], c_address: [DpHfJFKVkr4LQeif9t5SSKzDUa4G3asKeya5M7dz], c_nationkey: [24], c_
eep. quickly silent requests wake fluffily. furiously close packages nag furiously across the caref]
c_custkey: [94308], c_name: [Customer#000094308], c_address: [5aE9PLelngeVg8v6KCgOyR,NhDT5uC4], c_nationkey: [24], c_phone: [3
fully unus]

 query over dbfiles/customer.bin returned 4 recs
```


```
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /home/mk/Documents/uf docs/sem 2/Database Implementation/10_git/tpch-dbgen/
 heap files dir:        dbfiles/


 select test option:
        1. create sorted dbfile
        2. scan a dbfile
        3. run some query
        3

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

 enter CNF predicate (when done press ctrl-D):
        (c_phone > '34-999-195-7029') AND (c_mktsegment = 'FURNITURE')
       c_custkey: [103505], c_name: [Customer#000103505], c_address: [XPNXpdCD7HRqcV8ABpjF3ixDc9A6X ZB2 ], c_nationkey: [24],
deposits boost. quickly ironic requests could have to are according to the final accounts. final]
c_custkey: [62707], c_name: [Customer#000062707], c_address: [bRbtSTIXhh], c_nationkey: [24], c_phone: [34-999-473-9448], c_ac
ns. bold, fina]
c_custkey: [41774], c_name: [Customer#000041774], c_address: [DpHfJFKVkr4LQeif9t5SSKzDUa4G3asKeya5M7dz], c_nationkey: [24], c_
eep. quickly silent requests wake fluffily. furiously close packages nag furiously across the caref]
c_custkey: [94308], c_name: [Customer#000094308], c_address: [5aE9PLelngeVg8v6KCgOyR,NhDT5uC4], c_nationkey: [24], c_phone: [3
fully unus]

 query over dbfiles/customer.bin returned 4 recs
```

GTest Results

```
(base) mk@mk:~/Documents/uf docs/sem 2/Database Implementation/workspace/DBI/P2_P2$ ./gtest
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from testingLoadSortedFile
[ RUN      ] testingLoadSortedFile.testingLoadSortedFile
[       OK ] testingLoadSortedFile.testingLoadSortedFile (0 ms)
[----------] 1 test from testingLoadSortedFile (0 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (0 ms total)
[  PASSED  ] 1 test.
(base) mk@mk:~/Documents/uf docs/sem 2/Database Implementation/workspace/DBI/P2_P2$
```