COP6726 – Database System Implementation

Project 2 (Milestone Two) Report

Team members:

1. Akshay Ganapathy (UFID: 3684-6922)

2. Riyaz Shaik (UFID: 4360-0170)

Steps to execute project code:

The code was executed using Windows Subsystem for Linux (Ubuntu 18.04) environment.

- 1. The project source code is in the folder "a2-2test". Its contents include the following:
 - a. Code needed to run test.cc.
 - b. All ".bin" files should be included in the "bin" folder.
 - c. Google test framework has been added in the directory "googletest". This framework is needed to run gtest unit test cases.
 - d. All the gtest unit test cases have been added in "test" folder. DBFileTest.cc & HeapDBFileTest.cc contains the unit test cases for methods in DBFile class and HeapDBFile class, respectively. Make sure to include "nation.bin", "nation.bin.meta", "region.bin", "region.bin.meta" files in the directory, "test/data".
 - e. Makefile to compile test.cc and gtest unit test cases.
- 2. To run test.cc, from project root execute command "make test.out" and then execute "./test.out" and follow the on-screen instructions.

This test driver gives a menu-based interface to three options that allows you to test the code:

- 1. Create a sorted DBfile
- 2. Scan a DBfile
- 3. Run some query
- 3. To run gtest unit test cases, from project root execute command "make gtests" and then execute "./gtests".

The following are the methods with their description:

Method	Description
int MetaDataHandler::Open();	Opens the .meta file and reads its
	contents.
int MetaDataHandler::Close();	Closes the .meta file.
void	Increments number of pages.
MetaDataHandler::IncrementNumberOfPages();	
int MetaDataHandler::GetPages();	Returns number of pages.

Trype MetaDataHandler::GetType(); OrderMaker *MetaDataHandler::GetOrderMaker(); int MetaDataHandler::GetRunLength(); Returns run length. void GenericDBFile::MoveFirst(); Int GenericDBFile::GetNext(Record &fetchme); int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves first record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. Load the file with respect to the appropriate schema. File GenericDBFile::GetFile(); int GenericDBFile::GetLength(); int SortedDBFile::GetLength(); int SortedDBFile::GetLength(); int SortedDBFile::Close(); int SortedDBFile::Close(); int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record & Gets the next record in the sort order of the file. int SortedDBFile::PerformBinarySearch(Record & Gets the next record in the sort order of the file. int SortedDBFile::Add(Record & Gets the next record in the sort order of the file. GenericDBFile::Add(Record & Gets); Adds a record to the file. GenericDBFile::Add(Record & Gets); Adds a record to the file. GenericDBFile::Add(Record & Gets); Adds a record to the file. GenericDBFile::Add(Record & Gets); Gets the DBFile: Gets the DBFile: Gets the current length of the file.	6	
*MetaDataHandler::GetOrderMaker(); int MetaDataHandler::GetRunLength(); void GenericDBFile::MoveFirst(); Int GenericDBFile::Close(); Int GenericDBFile::GetNext(Record &fetchme); Int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. Int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. Int GenericDBFile::GetFile(); Int GenericDBFile::GetElength(); Int GenericDBFile::GetLength(); Int SortedDBFile::GetLength(); Int SortedDBFile::GetLength(); Int SortedDBFile::SetAttribute(OrderMaker *o, int run); Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. Int GenericDBFile::Add(Record &rec); Adds a record to the file. Gets the DBFile:	fType MetaDataHandler::GetType();	Returns whether type is heap or sorted.
int MetaDataHandler::GetRunLength(); void GenericDBFile::MoveFirst(); Move to the first record on the first page. int GenericDBFile::Close(); Closes the file. int GenericDBFile::GetNext(Record &fetchme); Fetches the next record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); Returns the file. int GenericDBFile::GetLength(); Returns the current length of the file. int SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF &cfs the next record in the sorted file according to cnf. int SortedDBFile::GetNext(Record &fetchme, CNF &cfs the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. Get sthe next record to the file. Get sthe next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. Get sthe DBFile::Add(Record &rec); Adds a record to the file. Get sthe DBFile: Get sthe DBFile.		Returns the sort order of file.
void GenericDBFile::MoveFirst(); int GenericDBFile::Close(); Closes the file. Closes the file. Fetches the next record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. Int GenericDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); Int GenericDBFile::GetLength(); Int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, CNF &cord &literal); int OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile. Move to the file. Closes the file. Fetches the next record on that page. If page is full, retrieves first record from next page. If page is full, retrieves first record in the gance is full, retrieves first record from next page. If page is full, retrieves first record on sectors for successful retrieves first record on the file. Returns the current length of the file. Creates an instance of Bigq class. Creates an instance of Bigq class. Shuts down the input pipe and closes the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the DBFile::GetDBFile:Content of the file. Gets the DBFile.	*MetaDataHandler::GetOrderMaker();	
int GenericDBFile::GetNext(Record &fetchme); fetches the next record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); int GenericDBFile::GetLength(); int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF &cnf &cnf &cnf &cnf &cnf &cnf &cnf &cnf	int MetaDataHandler::GetRunLength();	Returns run length.
int GenericDBFile::GetNext(Record &fetchme); int GenericDBFile::GetNext(Record &fetchme); int GenericDBFile::GetNext(Record &fetchme); int GenericDBFile::GetNext(Record &fetchme); int GenericDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); Int GenericDBFile::GetLength(); Returns the file. int SortedDBFile::SetLength(); Returns the current length of the file. int SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile. Closes the file. Fetches the next record according to the file next record in the sorted file according to the cnf expression and the sort order of the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the DBFile: Gets the DBFile. Gets the DBFile.	void GenericDBFile::MoveFirst();	Move to the first record on the first
int GenericDBFile::GetNext(Record &fetchme); Betches the next record on that page. If page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. Int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. Void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); Int GenericDBFile::GetLength(); Int SortedDBFile::Add(Record &addme); Void SortedDBFile::SetAttribute(OrderMaker *o, int run); Int SortedDBFile::GetNext(Record &fetchme, CNF & Shuts down the input pipe and closes the file. Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record & He input pipe and closes the file. Int SortedDBFile::PerformBinarySearch(Record & Performs a binary search according to the cnf expression and the sort order of the file. Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. Gets the DBFile::Add a record to the file. GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.		page.
page is full, retrieves first record from next page. Returns 1 on successful retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); Int GenericDBFile::GetLength(); Returns the file. int SortedDBFile::GetLength(); Returns the current length of the file. int SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record & Performs a binary search according to the file. file GenericDBFile::Add(Record &rec); Adds a record to the file. Gets the DBFile:	int GenericDBFile::Close();	Closes the file.
next page. Returns 1 on successful retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. void GenericDBFile::Load(Schema &myschema, Load the file with respect to the appropriate schema. File GenericDBFile::GetFile(); Int GenericDBFile::GetLength(); Int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &file. Int SortedDBFile::Add(Record &rec); Adds a record to the file. GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	int GenericDBFile::GetNext(Record &fetchme);	Fetches the next record on that page. If
retrieval, otherwise, returns 0. int GenericDBFile::GetNext(Record &fetchme, CNF & Retrieves next record according to the given CNF expression. void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); int GenericDBFile::GetLength(); int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF & Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &file. file ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); Gets the DBFile. Gets the DBFile. Gets the DBFile.		page is full, retrieves first record from
int GenericDBFile::GetNext(Record &fetchme, CNF & const char *loadpath); File GenericDBFile::GetFile(); int GenericDBFile::GetFile(); int GenericDBFile::GetLength(); int SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF & const char *loadpath); int SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF & const char *loadpath); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. GenericDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.		next page. Returns 1 on successful
&cnf, Record &literal);given CNF expression.void GenericDBFile::Load(Schema &myschema, const char *loadpath);Load the file with respect to the appropriate schema.File GenericDBFile::GetFile();Returns the file.int GenericDBFile::GetLength();Returns the current length of the file.int SortedDBFile::Add(Record &addme);Adds a record to the input pipe.void SortedDBFile::SetAttribute(OrderMaker *o, int run);Creates an instance of Bigq class.int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal);Shuts down the input pipe and closes the file.int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal);Gets the next record in the sorted file according to cnf.int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &cnforder, ComparisonEngine &cmp);Performs a binary search according to the cnf expression and the sort order of the file.GenericDBFile::Add(Record &rec);Adds a record to the file.GenericDBFile *DBFile::GetDBfile();Gets the DBFile.		retrieval, otherwise, returns 0.
void GenericDBFile::Load(Schema &myschema, const char *loadpath); File GenericDBFile::GetFile(); int GenericDBFile::GetLength(); int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &queryorder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); Gets the DBFile. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Load the file with respect to the appropriate schema. Returns the file. Creates an instance of Bigq class. Shuts down the input pipe and closes the file. Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. GenericDBFile::Add(Record &rec); Adds a record to the file. Gets the DBFile.	int GenericDBFile::GetNext(Record &fetchme, CNF	Retrieves next record according to the
const char *loadpath); File GenericDBFile::GetFile(); int GenericDBFile::GetLength(); Returns the file. Returns the current length of the file. Adds a record to the input pipe. Creates an instance of Bigq class. Int run); Int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. Int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record & the file. Returns the file. Adds a record to the input pipe. Creates an instance of Bigq class. Creates an instance of Bigq class. Shuts down the input pipe and closes the file. Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); Gets the DBFile. Gets the DBFile.	&cnf, Record &literal);	given CNF expression.
File GenericDBFile::GetLength(); int GenericDBFile::GetLength(); Returns the file. Returns the current length of the file. Adds a record to the input pipe. Creates an instance of Bigq class. Creates an instance of Bigq class. Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &queryorder, Record & the cnf expression and the sort order of the file. GenericDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile. Returns the file. Returns the file. Returns the file. Returns the file. Returns the current length of the file. Adds a record to the file. Gets the DBFile. Gets the DBFile.	void GenericDBFile::Load(Schema &myschema,	Load the file with respect to the
int GenericDBFile::GetLength(); int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::Close(); Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. GenericDBFile::Add(Record &rec); Gets the DBFile.	const char *loadpath);	appropriate schema.
int SortedDBFile::Add(Record &addme); void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::Close(); Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, CNF & according to cnf. int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record & the cnf expression and the sort order of &literal, OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile. Adds a record to the file. Gets the DBFile.	File GenericDBFile::GetFile();	Returns the file.
void SortedDBFile::SetAttribute(OrderMaker *o, int run); int SortedDBFile::Close(); Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &fetchme, OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Creates an instance of Bigq class. Shuts down the input pipe and closes the file. Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. GenericDBFile::Add(Record &rec); Gets the DBFile.	int GenericDBFile::GetLength();	Returns the current length of the file.
int run); int SortedDBFile::Close(); Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record & the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Shuts down the input pipe and closes the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	int SortedDBFile::Add(Record &addme);	Adds a record to the input pipe.
int SortedDBFile::Close(); Shuts down the input pipe and closes the file. int SortedDBFile::GetNext(Record &fetchme, CNF & Gets the next record in the sorted file according to cnf. int SortedDBFile::PerformBinarySearch(Record & Performs a binary search according to the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. GenericDBFile:Add(Record &rec); Gets the DBFile.	void SortedDBFile::SetAttribute(OrderMaker *o,	Creates an instance of Bigq class.
the file. int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Adds a record to the file. Gets the DBFile: Gets the next record in the sorted file according to the cnf expression and the sort order of the file. GenericDBFile:Add(Record &rec); Gets the DBFile.	int run);	
int SortedDBFile::GetNext(Record &fetchme, CNF &cnf, Record &literal); Gets the next record in the sorted file according to cnf. Performs a binary search according to the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the DBFile: Gets the next record in the sorted file according to the cnf expression and the sort order of the file. Gets the DBFile: Gets the DBFile:	int SortedDBFile::Close();	Shuts down the input pipe and closes
&cnf, Record &literal);according to cnf.int SortedDBFile::PerformBinarySearch(Record &fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &cnforder, ComparisonEngine &cmp);Performs a binary search according to the cnf expression and the sort order of the file.ComparisonEngine &cmp);the file.int HeapDBFile::Add(Record &rec);Adds a record to the file.GenericDBFile *DBFile::GetDBfile();Gets the DBFile.		the file.
int SortedDBFile::PerformBinarySearch(Record & Performs a binary search according to & the cnf expression and the sort order of the file. ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Performs a binary search according to the cnf expression and the sort order of the file. ComparisonEngine &cmp); Gets the DBFile.	int SortedDBFile::GetNext(Record &fetchme, CNF	Gets the next record in the sorted file
&fetchme, OrderMaker &queryorder, Record &literal, OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); the cnf expression and the sort order of the file. Adds a record to the file. Gets the DBFile.	&cnf, Record &literal);	according to cnf.
&literal, OrderMaker &cnforder, ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	int SortedDBFile::PerformBinarySearch(Record	Performs a binary search according to
ComparisonEngine &cmp); int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	&fetchme, OrderMaker &queryorder, Record	the cnf expression and the sort order of
int HeapDBFile::Add(Record &rec); GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	&literal, OrderMaker &cnforder,	the file.
GenericDBFile *DBFile::GetDBfile(); Gets the DBFile.	ComparisonEngine &cmp);	
	int HeapDBFile::Add(Record &rec);	Adds a record to the file.
int DBFile::GetLength(); Gets the current length of the file.	GenericDBFile *DBFile::GetDBfile();	Gets the DBFile.
	int DBFile::GetLength();	Gets the current length of the file.

The following are the screenshots of the output after executing the 3 testcases.

The following results are obtained by using 1 GB TPCH data generated from tpch-dbgen.

1)

```
kshayg1996@LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$ ./test.out
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:
                     catalog
tpch files dir:
                       tpch-dbgen/
heap files dir:
                       bin/
select test option:
        1. create sorted dbfile
        2. scan a dbfile
        3. run some query
select table:
        1. nation
        region
        customer
        4. part
        5. partsupp
        6. supplier
        orders
        8. lineitem
specify sort ordering (when done press ctrl-D):
        (c_phone)
specify runlength:
output to dbfile : bin/customer.bin
input from file : tpch-dbgen/customer.tbl
select option for : bin/customer.bin
        1. add a few (1 to 1k recs)
        2. add a lot (1k to 1e+06 recs)
        3. run some query
        4. finish
File present at path: tempFile.bin is now opened.
        added 150000 recs..so far 150000
create finished.. 150000 recs inserted
akshayg1996@LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$
```

```
.c_custkey: [3174], c_name: [Customer#000003174], c_address: [9XGPFKLWZMU], c_nationkey: [24], c_phone: [34-997-244-3200], c_acctbal: [6117.29],
 courts are furio
c_custkey: [86364], c_name: [Customer#000086364], c_address: [vMKeNm EzZa75U3vkmI2Rnp0jC9], c_nationkey: [24], c_phone: [34-997-251-4190], c_acc
regular pinto beans. slyly regular ideas sleep slyly along the f]
.c_custkey: [35286], c_name: [Customer#000035286], c_address: [P HN6QXTmlBpJ5Xps], c_nationkey: [24], c_phone: [34-997-517-4660], c_acctbal: [12
ording to the special, even requests. express foxes ]
c_custkey: [48253], c_name: [Customer#000048253], c_address: [gAVusA09FVBERzgQoz S1Qc9wYkO7gYX1TLRkG], c_nationkey: [24], c_phone: [34-997-697-
the carefully ironic ideas. slyly express foxes sleep quickly. carefully special]
.c_custkey: [31537], c_name: [Customer#000031537], c_address: [82Fr5r7gWxt2a5], c_nationkey: [24], c_phone: [34-998-190-8709], c_acctbal: [-252.:
.c_custkey: [117513], c_name: [Customer#000117513], c_address: [cuBZ TE0EONnPFmoOdv6EIHxKU43AV], c_nationkey: [24], c_phone: [34-998-315-4817], or
ains]
.c_custkey: [53730], c_name: [Customer#000053730], c_address: [6fv7IW9Pr91QntCZzwsi1e2GXOxJj9A ], c_nationkey: [24], c_phone: [34-998-382-6518],
 regular, regular packages. enticing, even deposits wake slyly among the]
.c_custkey: [12022], c_name: [Customer#000012022], c_address: [PwVDMfqZtKWOo], c_nationkey: [24], c_phone: [34-998-408-4382], c_acctbal: [2877.36
lar ideas cajole blithely]
.c_custkey: [125210], c_name: [Customer#000125210], c_address: [BJiJAg79TvSTdxgPeb63sUqKaOVv], c_nationkey: [24], c_phone: [34-998-711-2597], c_
lites among the slyly even patterns nag slyly fluffi]
.c_custkey: [50667], c_name: [Customer#00050667], c_address: [9aGxwCYLcG6y2eA], c_nationkey: [24], c_phone: [34-999-115-2195], c_acctbal: [8764
.c_custkey: [7288], c_name: [Customer#000007288], c_address: [,dZCqJe94SqXQjxAd7WxRMYyFVKGd4pFIrAshk], c_nationkey: [24], c_phone: [34-999-121-77
lly 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: [34-999-195-7029], c_acctbal: [6912.63
p blithely after the blithely even deposits. carefully regula]
.c_custkey: [148611], c_name: [Customer#000148611], c_address: [HONNKAELd2vMNyGWpw3ZZgSNaDs7fOdNIH64], c_nationkey: [24], c_phone: [34-999-283-64
pending asymptotes sleep carefully about the slyly even excu]
.c_custkey: [102689], c_name: [Customer#000102689], c_address: [uPFxngvSOaDQIL], c_nationkey: [24], c_phone: [34-999-363-7145], c_acctbal: [6256.
to the furiously bold courts. daring ]
.c_custkey: [35478], c_name: [Customer#000035478], c_address: [Ot5zcyiiLhzXXivX4Y2FeToQIh,iovs6zyA], c_nationkey: [24], c_phone: [34-999-379-6016
ackages. daring, even de]
.c_custkey: [103505], c_name: [Customer#000103505], c_address: [XPNXpdCD7HRqcV8ABpjF3ixDc9A6X ZB2 ], c_nationkey: [24], c_phone: [34-999-404-9770
ickly 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_acctbal: [7429.32],
.c_custkey: [102414], c_name: [Customer#000102414], c_address: [Xzg9eiWbsxiQW,LUCyUAcPoOENdLUQt], c_nationkey: [24], c_phone: [34-999-618-6881],
even packages by the permanently special gifts ne]
scanned 150000 recs
 kshayg1996@LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$
```

```
cshayg1996@LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$ ./test.out
  ** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:
tpch files dir:
                                             catalog
                                             tpch-dbgen/
 heap files dir:
                                             bin/
  select test option:
                 1. create sorted dbfile
                 2. scan a dbfile
                 3. run some query
 select table:
                 1. nation
                region
                customer
                 4. part
                5. partsupp
                6. supplier
                 orders
                 8. lineitem
enter CNF predicate (when done press ctrl-D):

(c_phone > '34-999-195-7029') AND (c_mktsegment = 'FURNITURE')File present at path: tempFile.bin is now opened.

c_custkey: [103505], c_name: [Customer#000103505], c_address: [XPNXpdCD7HRqcV8ABpjF3ixDc9A6X ZB2], c_nationkey: [24], c_phone: [34-99 ost. 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_acctbal: [7429.32] ina]
c_custkey: [41774], c_name: [Customer#000041774], c_address: [DpHfJFKVkr4LQeif9t5SSKzDUa4G3asKeya5M7dz], c_nationkey: [24], c_phone: [34-999-5
y 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: [34-999-618-6653],
 query over bin/customer.bin returned 4 recs
```

The following are the results of gtest unit test cases:

Result: All the unit test cases passed

DBFileTest.cc & HeapDBFileTest.cc contains the unit test cases for methods in DBFile class and HeapDBFile class, respectively.

To run gtest unit test cases, from project root execute command "make gtests" and then execute "./gtests".

Make sure to include "nation.bin", "nation.bin.meta", "region.bin", "region.bin.meta" files in the directory, "test/data".

```
LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$ ./gtests
Running main() from gmock_main.cc
            Running 7 tests from 2 test suites.
            Global test environment set-up.
            6 tests from DBFileTest
           DBFileTest.testCreateHeapFile
       OK ] DBFileTest.testCreateHeapFile (2 ms)
            DBFileTest.testCreateSortedFile
           DBFileTest.testCreateSortedFile (5 ms)
            DBFileTest.testLoadFile
            DBFileTest.testLoadFile (8 ms)
            DBFileTest.testOpenHeapFile
           DBFileTest.testOpenHeapFile (5 ms)
           DBFileTest.testOpenSortedFile
ile present at path: tempFile.bin is now opened.
       OK ] DBFileTest.testOpenSortedFile (7 ms)
           DBFileTest.testClose
       OK ] DBFileTest.testClose (9 ms)
        ---] 6 tests from DBFileTest (36 ms total)
       ----] 1 test from HeapDBFileTest
            HeapDBFileTest.testGetNext
            HeapDBFileTest.testGetNext (11 ms)
            1 test from HeapDBFileTest (11 ms total)
        ---] Global test environment tear-down
            7 tests from 2 test suites ran. (47 ms total)
  PASSED ] 7 tests.
kshayg1996@LAPTOP-ACV240Q1:/mnt/d/Gitlab/dbi/Project 2.2$
```