Data Base System Implementation
COP-6726

Submitted by:
Naman Arora
UFID: 3979-0439
Nikunj Sarda
UFID: 9360-6581

Introduction:

This document is intended to present the outcomes and the process of the completed work for the Project 1. The *heap file* implementation has been successfully materialized and tested. The repository for all the work is present here.

Source File Hierarchy:

The Source code for this project, including the boilerplate code, was rearranged for the ground up into logical sub directories, explained as follows:

All the subdirectories contain respective Makefiles that, when called recursively, compile the contained source code into separate object files. These object files are then linked by the root Makefile to produce the output binary of the program as required.

The db/ sub-dir:

This subdirectory contains the schema.cc, schema.h, catalog file and its Makefile. (Fig. 2)

The fs/ sub-dir:

This subdirectory contains the filesystem related source files, *viz*. file.cc, record.cc and dbfile.cc with their respective header files and the Makefile. (Fig. 3)

The mem/ sub-dir:

This subdirectory comprises of the two_way_list.cc and its header files with the required Makefile. (Fig. 4)

The glbl/ sub-dir:

This subdirectory contains the defs.h header along with the timer class source code in timer.cc and timer.h. (Fig. 5)

The lex/ sub-dir:

The subdirectory houses all the flex/yacc related source code, *viz.* comparison.cc, comparison_engine.cc, lexer.l, parser.y and parse_tree.h along with respective headers and Makefile. (Fig. 6)

The out/ and try/ sub-dir:

This directory will house the <relation_name>.bin files after the DBFile class bulk loads the *.tbl files. While the try/ subdirectory can contain any trial code or files that wont be comitted to the git repository. The contents of both these sub-directories are ignored by the git. (Fig. 7)

The tpch-dbgen subdir:

This is a git submodule with origin at https://github.com/electrum/tpch-dbgen.git.

The src/ root level directory:

The src/ directory acts as the root level directory as all the paths in source code are relative to this directory.

All the subdirectories has soft links to some of the subdirectories. This facilitates the linking process when an object file from subdirectory-1 requires to be linked with object file of sub-directory-2.

The compilation process:

The compilation process is accomplished by a recursive call to Makefiles in each subdirectory by the Makefile in the src/ directory.

The provided NamanArora_NikunjSarda_p1.zip has one sub-directory, src/.

```
$> unzip NamanArora_NikunjSarda_p1.zip
$> cd proj1/src
```

Now to build the test.bin:

```
$> make
```

To build the main.bin:

```
$> make main_bin
```

To make the gtest (assuming the gtest framework is installed and available in default linker library path)

```
$> make gtest_bin
```

To generate tpch-dbgen files (tpch-dbgen/ sub-dir is always compiled with all of the above targets)

```
$> cd tpch-dbgen/
$> ./dbgen <opetions> -s <size>
```

All the available relation binary files are available in out/

To clean the repository, excluding the tpch-dbgen/*.tbl and out/*.bin files,

```
$> make clean
```

To clean the repository, everything inclusive

```
$> make distclean
```