Database System Implementation

Project 2 - Implementing a Sorted File

Developed by –

Ghodkari Chowdary, Raghunatha Rao - UFID: 6218-1051

Mullapudi, Aseesh - UFID: 9175-1971

Aim of this project is to implement a Sorted file.

We have completed the BigQ class by writing the required functions. The job of the BigQFilewithin the system is to sort the records based on the input predicate, write to console and the store the sorted file with.bigq extension.

Implementation Details:
We have created the following private fields in the BigQ.h

Pipe *inputPipe; // input Pipe to get the records

Pipe *outputPipe; // output Pipe to push records

OrderMaker *sortedOrder; // sorted Order required for sorting

int *runlength; // the run length input from console

File *runsFile; // File pointer for Runs File which stores the Runs

vector <int> runPointers; // list of pointers to all the runs

Method Definitions:
- ***BigQ::BigQ(Pipe &in, Pipe &out, OrderMaker &sortorder, int runlen)***

  Constructor used to initialize the fields of the BigQ class, A new runs File is created to store the runs

- ***static void* invoke_tpmmsAlgo(void* args);***

  Function which is the start of the worker thread which performs the sorting, we are typecasting the void arguments to bigQ and invoking the worker function

- ***void BigQ::worker()***

  worker function which does the TPMMS Algorithm, Algorithm is written in two phases where first phase breaks the File into runs and sorts each run. In second phase all the runs are merged into a single sorted file using priority Queue

- ***void BigQ::sortRun(vector <Record*> &)***

  function which sorts the vector of records using the input predicate

- ***int BigQ::addRuntoFile(vector <Record*> &)***

  function take input of sorted records into vector and writes them to File and return the current size of file which act as pointer to the next run

File **gtest.cc** written to unit test the TPMMS Algorithm of the BigQ on different tables

Instructions to Run the Code:

Compile and run test.cc:

Ensure the directory contains the bin files generated from the Project1 so that code has access to Heap files

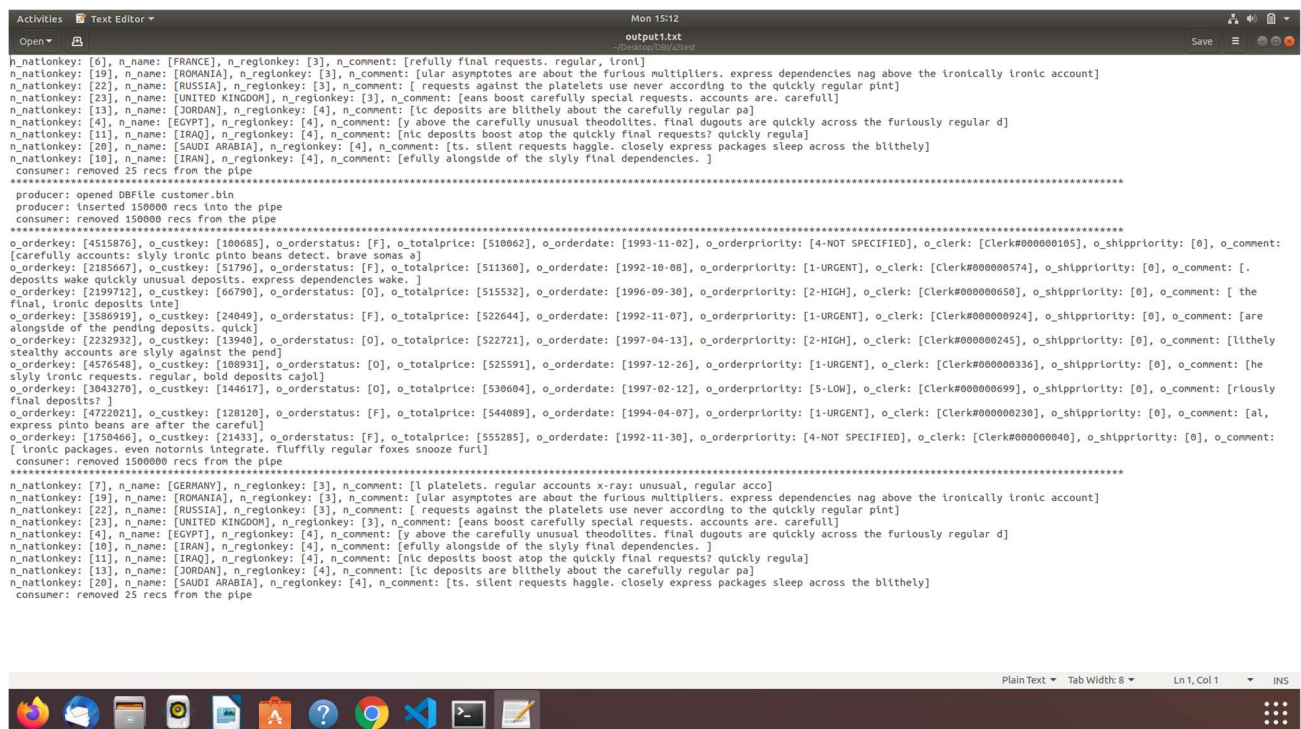Ensure there is Catalog file in the root folder as well

1. make
2. ./test.out

To run the test cases

1. make
2. ./runsTestCases.sh

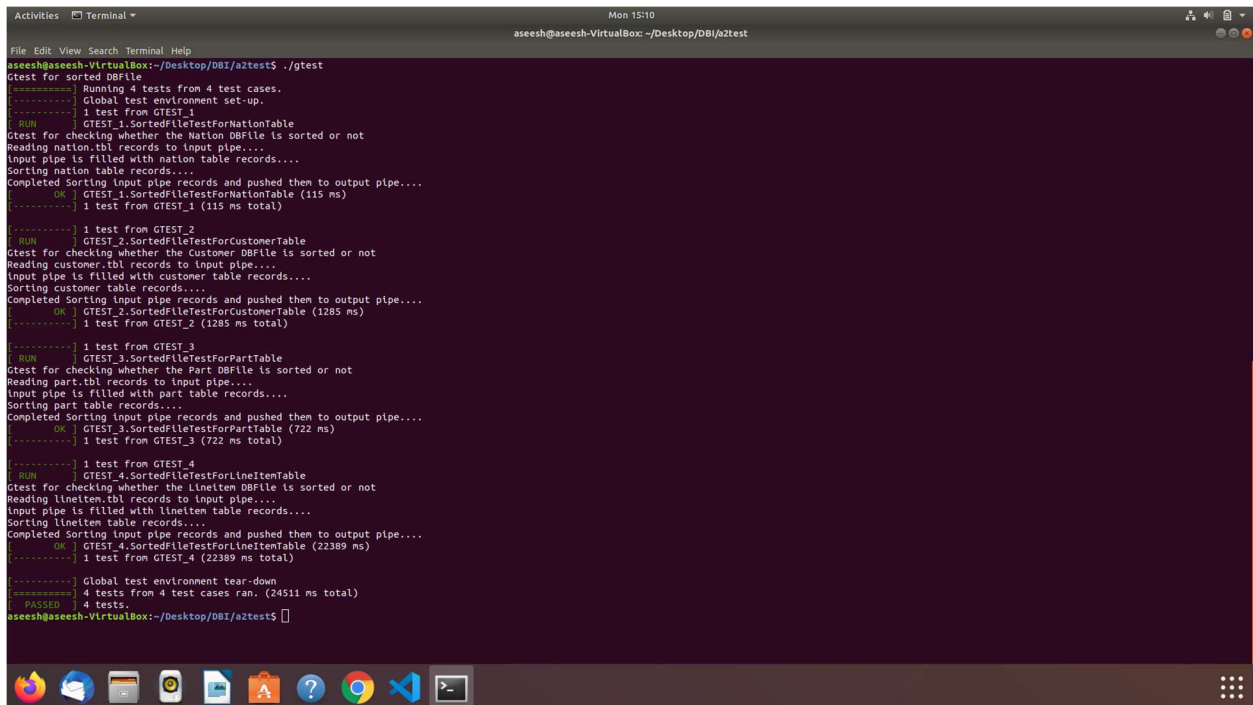File output1.txt is generated

Screen Shot of the output1.txt is below

Instructions to run gtests:

Compile and run code gtest.cc:

Ensure the directory contains the .tbl files generated from the tpch-dbgen so that code has access to datasets

1. make gtest
2. ./gtest

The screenshot of ./gtest is present on the next page