Nicholas Ang

CS457

Project Assignment 4

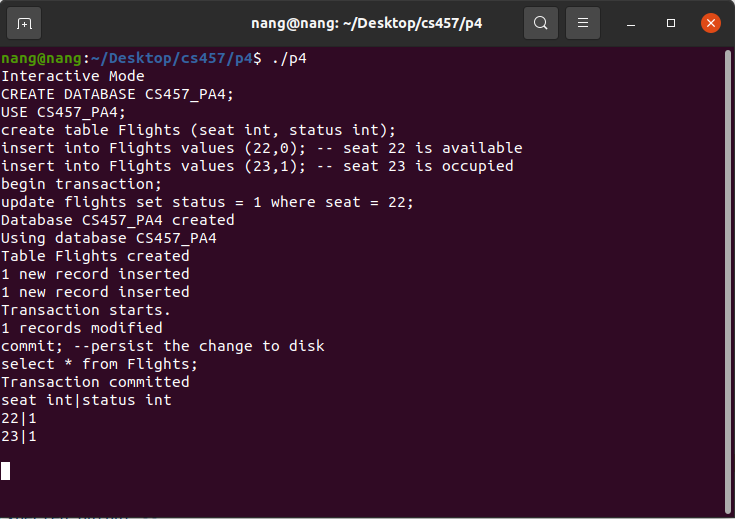
PA4 Design Document

The atomicity of transactions was done using lock files to ensure that transactions are atomic. When a transaction starts and the user tries to modify a table, it will check if there exists a lock file. If the lock file exists, the process will be unable to modify or update the table. The lock file has the format of <table name>\_lock. If the user tries to commit while they do not have access to the table, it will abort the commit and nothing will be changed on the table. The user cannot successfully commit unless they are the one that is modifying the table and has access to the table.

If the lock file does not exist, it will generate a lock file and allow the user to access the table. Only one process or user is able to create/delete a lock for a table at a time. The user is then able to update the table and modify records but the changes will not be written to the disk until a commit occurs. When the user commits, the table and all changes that happened to the table will be saved to the table file. After the commit is successful, the lock file will be deleted. Since the lock file was deleted, any user or process can take the lock, modify the table, and commit.

**Execution Instructions:** Please use the command *g++ p4.cpp -o p4* to compile the program and use *./p4* to run the program. To test the whole script with two “processes”, open two terminals running the program so both are running .*/p4*. One terminal will run the commands for P1 and the other terminal will run the commands for P2. You should be able to simply copy and paste the commands from the script or type the commands manually.

The image below shows the output for process 1 or P1 in the terminal when the commands in the test script file are run with the program.



The image below shows the output for process 2 or P2 in the terminal when the commands in the test script file are run with the program.

