Daniela Castorena
CS 457
Dr. Zhao
15 May 2023

PA4 Design Document

Program's purpose
The goal of this program is to further expand a Python-based database resembling SQL. In the initial version (PA1), fundamental commands were introduced, such as creating or removing databases and tables, as well as querying and modifying the tables. In the second version (PA2), improvements were made to update and query table contents more efficiently. Table joins were implemented in the following stage (PA3). In this final update, transaction functionality was implemented into the program.

At a high level, how transactions were implemented
The only task in this programming assignment was to add a feature for handling transactions, which was a multi-step process. First, the program defaults to no locks being in place made by the user. With each command execution, the program checks if any file locks have been made. If they have, the "isLocked" flag will be set as true. However, if the user created these locks by beginning a transaction, leave it set to false. Once a transaction begins, the program checks for any locks in place. If not, locks are created for every table in the database and the "userMadeLocks" flag is set to true. If this flag is true, then the "isLocked" flag will never be true, and a commit will be successful. When committing, the program checks if there is a lock made by the user, if there is, it will release the locks and run any pending commands. If there was no lock made by the user, the transaction is instead aborted.

Transactions require that tables be modified in-memory without any changes to the database until committed. Table commands, like inserting, altering, or removing tuples, are affected by this. If a transaction is in progress, any changes made to the tables will be stored in a new separate file named "<TableName>.new.txt" and a command will be created to remove the existing table and replace it with the new one. These commands are placed into a list that are executed one by one when the user commits.

The process of removing all changes since the previous commit was not included, however, it would be very straightforward. Simply make a system call to delete any file ending in .new.txt. Dropping of tables was not included either, but would follow a similar process as anything relating to table modifications, wherein system commands would be saved and executed later. With the table modification commands, the user can only make one change per table before committing. To address this limitation, a flag or indicator would be necessary to determine if any changes have been made prior to proceeding.

Execution Commands
Run the command “python3 main.py”
Python 3.7 or newer

*The PA4_test.sql file was modified - “flights” changed to uppercase F “Flights” only since program is case sensitive.

-