**Design:**

This project will demonstrate the problem of creating a working ‘transaction’ and ‘commit’ process that makes the lock change. To implement the atomicity property of transactions in my project code let’s start with the main function we declare the variable “lock” to initiate the word “table” inside the array then we set the variable flag for the lock to keep track of the locks for when the flag is true it will print out the lock and flag. We used the test script for this transaction command, “begin transaction” which allows one of the processes to make a change. This shows the transaction process is in motion and that other process, P2 can’t make any changes to it while P1 can make changes since P1 didn’t commit it yet. Therefore, the P2 transaction will have abort since it couldn’t commit due to P1’s action. This demonstrate that the folder is locked when one of the processes can make any changes once P1 has commit it will be unlocked for P2 which they can make change since they couldn’t change due to P1’s action. The lock has been created separate folder that isn’t in the database folder for the processors that want to use the transaction when beginning typing the command “begin transaction”. The flag of the transaction determines if the folder is lock or not. If the flag is true, then they remove the lock from the folder so that other process can use it when two process try to use it the first process that use the transaction can change within the folder while the other processes can’t change anything. If a transaction is successful then commit it, which allows the ‘commit’ command to guarantees all of transaction’s modification are made permanent part of the database and it can free resources like locks used by transaction. I couldn’t figure out how to print “Transaction abort.”, “Error: Table Flights is locked!”, and “1 record modified.” However, I do have the code for “Error: Table Flights is locked!” and “1 record modified.” but it isn’t printing on the P2 for the command, “update flights set status = 1 where seat = 22;” however it did print out when I enter command, “commit;” for some reason when I try to put it in the while loop but then it will print “1 record modified.”, but then it will change P2 result to:

seat int|status int

22|1

23|1

for some reason which I don’t understand how it does that and I try to print the error message, but it wouldn’t let me. For “1 record modified.” I have the code for it when I try to implement it in my transaction function it will give my P2 a different result when I enter the command, “select \* from Flights” from the expected output that I need for this project. However, when I enter the command, “commit;” in P1 it prints out “1 record modified.” instead the command, “update flights set status = 1 where seat = 22;”. On P2 when I enter the command, “commit;” it will give me a error message instead of “Transaction abort.”

**Implementation:**

The program uses the if and elif statement to implement the design for transaction and commit. The command for all projects ends with semicolon (;) to be able to execute except for the line in the sql script has read comment that looks like “- -“. The program doesn’t care if you type uppercase letter or lowercase letter. I used os and re imports for the python script that helps modifying files for each different paths with the benefit of case sensitivity. As well as the space in the .sql command line since my program only allows no space for my design of this project, which is instructed below on how to type the sql commands into the processes.

To compile my program, I go to cse unr server my terminal can’t compile the program at all. Also, on P1 commands make sure no space for create table and insert into commands since it gives me an expected output of the results of the command.

Example: create table Flights(seat int, status int);

insert into Flights values(22,0);

insert into Flights values(23,1);

**Steps:**

You will need to open two terminal and login into cse unr server side by side:

ssh [username@ubuntu.cse.unr.edu](mailto:username@ubuntu.cse.unr.edu)

Go to where my file is located and executed by this command:

python2 jaredlam\_PA4.py

This will start the script which then you type the .sql commands in which they appear for then switch to second terminal for process, P2 to type .sql commands where it says P2 then go back to P1 to type .sql commands and lastly in P2 type .sql commands.