DS PHASE 3

Part 2

Analysis

- The compatibility matrix is as follows:

	Read	Write
Read	Yes	No
Write	No	No

- Here, WRITE operation is UPDATE and READ operation is PRINT.

Learnings

- We use locking as a means to concurrency control. When a transaction requests a lock on a data item, no other transaction can perform certain operations on the data item(as specified by the lock).
- A very simple lock is the binary lock, which prohibits other transactions from performing any operation on the data item.
- A less stricter form of lock allows for more concurrency. We can write out a compatibility matrix as mentioned in the Analysis section to decide how conflicting transactions must be dealt with.
- In the case when only the two operations(READ and WRITE) are considered, the lock for every data item can have three values: no lock, read lock and write lock.
- When a transaction is initiated, we check if the data items it wants to operate on is locked. If it is locked for that operation, the transaction is queued. Else, the transaction locks the data items with a relevant value and continues with its execution.

Implementation

- The data item in our case is at the table level. For every table, we can have three values: no lock(0), read lock(1) and write lock(2). We maintain a lock file for this purpose. Each line in a lock file consists of the table name and its lock status.
- When we execute a READ operation, we read the lock file to check the status of the relevant table. If it is 2, we keep reading the lock file until it becomes 0 or 1. We, then, change the lock status to 1, execute the command and change the status back to 0.
- When we execute a WRITE operation, we read the lock file to check the status of the relevant table. If it is 1 or 2, we keep reading the lock file until it becomes 0. We, then, change the lock status to 1, execute the command and change the status back to 0.
- We define a new class called the `LockManager`, that handles the lock files created for each table. It contains functions to create lock file when server is instantiated, read lock file, check current status of a given table from the lock file and update the status of a given table.

Video link

https://drive.google.com/file/d/1x2FlXJs0iN8RLP3aNu8FXxqUDfKoM4UQ/view?usp=sharing