**Adv DB Winter 2024 – Assignment 1 Report**

**Introduction**

Our code simulates a database system processing transactions and handling potential failures. The system maintains a database and processes transactions sequentially, updating the database accordingly. It also includes features for recovery in case of failure, logging transactions, and maintaining transaction integrity.

**Design Decisions and Data Structures**

**Database Structure**: The database is represented as a list of lists where each inner list represents a record, and the first list serves as the header containing attribute names.

**Log System**: The log system is a list of dictionaries. Each dictionary represents a log entry containing information about a transaction. The structure includes:

**'Transaction'**: Transaction index.

'**Before'**: Database state before the transaction.

**'After'**: Database state after the transaction.

**'Committed**': Indicates whether the transaction was committed successfully.

**'Time'**: Timestamp of the transaction (not implemented in the provided code).

**Functions Overview**

recovery\_script(log): This function is responsible for restoring the database to a stable state after a failure. It processes the database log in reverse order, reverting changes made by transactions up to the point of failure.

transaction\_processing(): Sequentially processes transactions from the global list, updating the database. On failure, it invokes the recovery script to revert changes and halts further processing, while providing transaction status feedback.

newTabl(log, file\_name): Creates a new csv file with the updated table that way if any transactions are successful they will be stored there and won’t overwrite the original.

transactionLog(log, file\_name): creates the journal or log in a csv file that will keep track of the transactions index (in order to know which transaction is updated) timestamp (in order to know who or when made the attempt) and status (to see if the attempt was successful or not).

**Changes and Implementations**

**Transaction Processing:** The provided code implements a basic transaction processing function. We expanded it to update the DB\_Log accordingly with each transaction's status.

**Recovery Mechanism:** The recovery script provided in the original code was enhanced to process the database log and revert changes made by failed transactions.

Error Handling: Implemented error handling to handle potential failures during transaction processing.

**Main Function:** Integrated the transaction processing, recovery mechanism, and logging functionalities into the main function.