Experiment No.10
Implementation and demonstration of Transaction and Concurrency control techniques using locks Date of Performance:
Date of Submission:

Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Aim :- Write a query to lock and unlock a table for transaction and concurrency control.

Objective :- To learn locking of tables for transaction processing and concurrency control.

Theory:

A lock is a mechanism associated with a table used to restrict the unauthorized access of the data in a table. MySQL allows a client session to acquire a table lock explicitly to cooperate with other sessions to access the table's data. MySQL also allows table locking to prevent unauthorized modification into the same table during a specific period.

Table Locking in MySQL is mainly used to solve concurrency problems. It will be used while running a transaction, i.e., first read a value from a table (database) and then write it into the table (database).

MySQL provides two types of locks onto the table, which are:

READ LOCK: This lock allows a user to only read the data from a table.

WRITE LOCK: This lock allows a user to do both reading and writing into a table.

The following is the syntax that allows us to acquire a table lock explicitly:

LOCK TABLES table_name [READ | WRITE];

The following is the syntax that allows us to release a lock for a table in MySQL:

UNLOCK TABLES;

Conclusion: Locking and unlocking of tables is achieved and verified using insert command in the same table of a database system.

1. Explain Transaction and Concurrency control techniques using locks.
   Concurrency control techniques using locks are mechanisms employed by the DBMS to manage access to shared resources (e.g., database records) by multiple transactions executing concurrently. Locks help maintain data consistency and prevent conflicts between transactions. Here are some common lock-based concurrency control techniques:

   Locking: Transactions acquire locks on database objects (e.g., rows, tables) to control access to them. Different types of locks include:
   Shared Lock (Read Lock): Allows multiple transactions to read the data but prevents any transaction from writing to it.
   Exclusive Lock (Write Lock): Grants exclusive access to a transaction for both read and write operations, preventing other transactions from reading or writing to the same data concurrently.
   Update Lock: A special type of lock that allows concurrent readers but prevents other transactions from acquiring an exclusive lock until the current transaction completes.
   Two-Phase Locking (2PL): In 2PL, transactions acquire locks in two phases: the growing phase (acquiring locks) and the shrinking phase (releasing locks). Once a transaction releases any lock, it cannot acquire any new locks.
   Deadlock Detection and Prevention: DBMSs implement deadlock detection algorithms to identify and resolve deadlocks (circular dependencies between transactions waiting for resources). Techniques such as timeout mechanisms or deadlock detection graphs are used to detect and break deadlocks.
   Timestamp-based Concurrency Control: Assigning unique timestamps to transactions and data items to determine their relative order of execution. Timestamp ordering or validation protocols ensure serializability by controlling the order of transaction execution based on timestamps.
   Optimistic Concurrency Control (OCC): Instead of locking resources, OCC allows transactions to execute without acquiring locks initially. It detects conflicts during transaction commit and rolls back conflicting transactions if necessary.wh