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

**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.

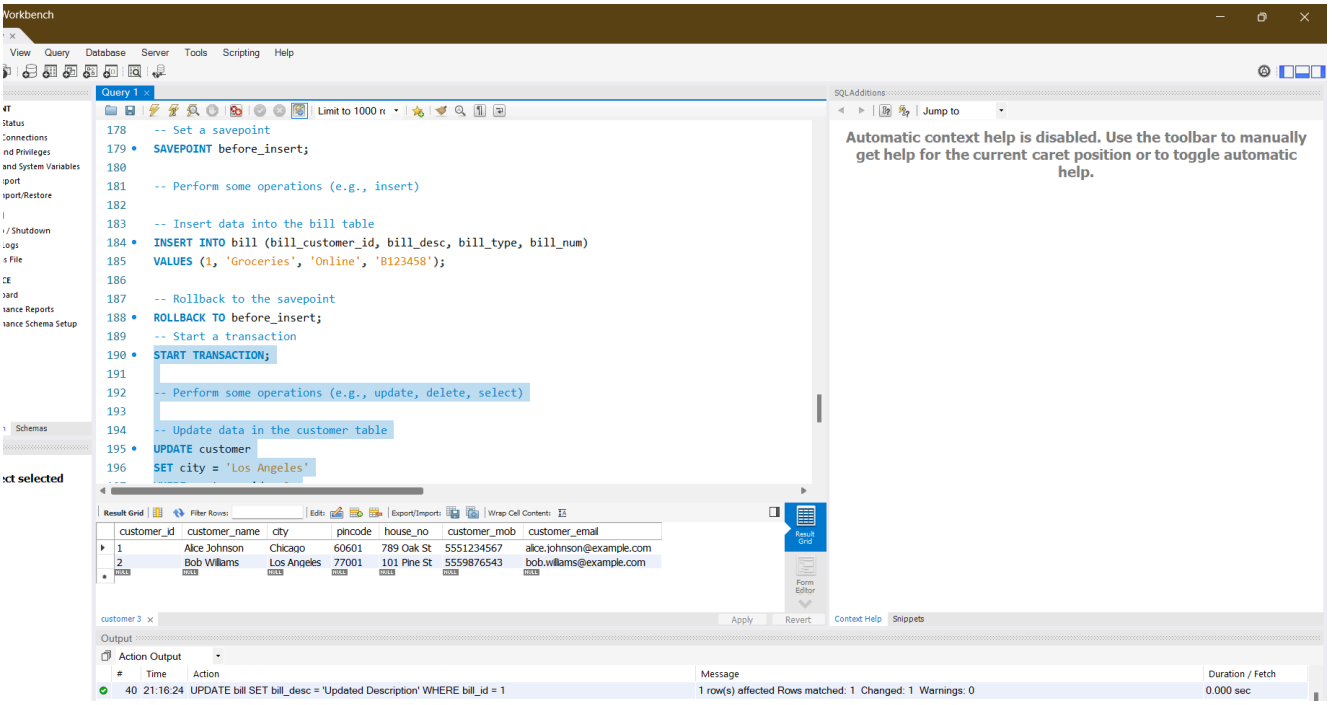
The following is the syntax that allows us to acquire a table lock explicitly:

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;

IMPLEMENTATION:




**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.

In conclusion, transactions and concurrency control techniques using locks are essential aspects of database management, ensuring data consistency, integrity, and isolation in multi-user environments. Here's a summary of their importance and how locks are utilized: 1. Transactions: - Transactions represent a sequence of operations that are executed as a single unit of work, either all successfully completed or all aborted. - ACID properties (Atomicity, Consistency, Isolation, Durability) ensure the reliability and integrity of transactions. - Transactions help maintain data consistency by allowing multiple operations to be treated as a single logical unit, preventing data corruption or partial updates. 2. Concurrency Control Techniques: - Concurrency control ensures that multiple transactions can execute simultaneously without interfering with each other, while still maintaining data consistency. - Lock-based concurrency control is a common technique used to manage access to shared resources. Locks prevent conflicting operations from occurring concurrently, ensuring that only one transaction can modify a resource at a time. - Types of locks include: - Shared Locks (Read Locks): Allow multiple transactions to read a resource simultaneously but prevent any transaction from writing to it. - Exclusive Locks (Write Locks): Restrict access to a resource exclusively to one transaction, preventing other transactions from reading or writing to it. - Locks can be applied at different granularities, such as row-level locks, table-level locks, or database-level locks, depending on the requirements of the application. Concurrency control techniques using locks offer several benefits, including: - Ensuring data integrity by preventing conflicting updates from occurring simultaneously. - Allowing for efficient utilization of system resources by maximizing concurrency while minimizing contention. - Providing mechanisms for deadlock detection and resolution to prevent transactional deadlocks. Overall, transactions and concurrency control techniques using locks play a crucial role in maintaining data consistency and integrity in database systems, ensuring reliable and predictable behavior in multi-user environments