



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

<b>Experiment No.10</b>
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;**

Implementation:

The screenshot shows a MySQL IDE with three tabs: '5\*', 'SQL File 6\*', 'SQL File 7\*', and 'SQL File 12\*'. The active tab 'SQL File 6\*' contains the following SQL code:

```
1 • use studentdb;
2
3 • START TRANSACTION;
4 • LOCK TABLE student WRITE;
5
```

Below the code editor is the 'Output' window, which displays the execution results of the queries. The output is as follows:

#	Time	Action
68	13:57:02	START TRANSACTION
69	13:57:02	LOCK TABLE student WRITE



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

```
5
6 • UPDATE student SET student_id = 15432710 WHERE student_id = 468383271;
7 • SELECT * FROM student;
8
```

student_name	student_id	Address	Contact_no	DOB
Viraj Oza	15432710	MG Road Mira Road	996733234	04/04/2004
Sarvesh Surve	562383271	SV Road Andheri West	996733249	12/09/2004
Rahul Yadav	562387890	LT Road Vasai West	996733678	18/10/2004
*	NULL	NULL	NULL	NULL

```
9 • COMMIT;
10 • UNLOCK TABLES;
11
12
```

#	Time	Action
76	14:17:11	COMMIT
77	14:17:12	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.

Ans.: Transactions ensure that a series of database operations are executed as a single unit of work, either all succeed or none do. Concurrency control techniques using locks manage access to shared resources in a multi-user environment. Locks can be applied at various levels, such as database, table, or row, to prevent conflicts between transactions.

In short, transactions ensure atomicity, consistency, isolation, and durability of database operations, while concurrency control techniques using locks prevent data inconsistency and ensure data integrity by managing concurrent access to shared resources.