

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Oracle DML Lock Techniques

Sarangkumar S. Dubey¹

Computer Science & Engineering
Sipna C.O.E.T
Amravati, India

Ajay P. Thakare²

Electronics & Telecommunication Engineering
Sipna C.O.E.T
Amravati, India

Abstract: *The purpose of a DML lock, also called a data lock, is to guarantee the integrity of data being accessed concurrently by multiple users. Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource. Locks can occur on any Oracle user objects such as tables and rows. Also System objects that are not visible to users, such as shared data structures in the memory and data dictionary rows. These locks will stay only till the duration of the session or the transaction. . DML locks prevent destructive interference of simultaneous conflicting DML or DDL operations.*

Keywords: *DML Lock; TM Lock; Table Lock; Row Lock.*

I. INTRODUCTION

The purpose of a DML lock, also called a data lock, is to guarantee the integrity of data being accessed concurrently by multiple users. For example, a DML lock can prevent multiple customers from buying the last copy of a book available from an online bookseller. DML locks prevent destructive interference of simultaneous conflicting DML or DDL operations. DML statements automatically acquire locks at both the table level and the row level. In the sections that follow, the acronym in parentheses after each type of lock or lock mode is the abbreviation used in the Locks Monitor of Oracle Enterprise Manager. Enterprise Manager might display "TM" for any table lock, rather than indicate the mode of table lock (such as RS or SRX).

II. TYPES OF LOCK

a) Row Locks (TX)

A row lock, also called a TX lock, is a lock on a single row of a table. A transaction acquires a row lock for each row modified by one of the following statements: INSERT, UPDATE, DELETE, MERGE, and SELECT ... FOR UPDATE. The row lock exists until the transaction commits or rolls back. When a transaction obtains a row lock for a row, the transaction also acquires a table lock for the table in which the row resides. The table lock prevents conflicting DDL operations that would override data changes in a current transaction.

b) Table Locks (TM)

A transaction automatically acquires a table lock (TM lock) when a table is modified with the following statements: INSERT, UPDATE, DELETE, MERGE, and SELECT ... FOR UPDATE. These DML operations require table locks to reserve DML access to the table on behalf of a transaction and to prevent DDL operations that would conflict with the transaction. You can explicitly obtain a table lock using the LOCK TABLE statement

A table lock can be held in any of the following modes:

- » A row share lock (RS), also called a subshare table lock (SS), indicates that the transaction holding the lock on the table has locked rows in the table and intends to update them. An SS lock is the least restrictive mode of table lock, offering the highest degree of concurrency for a table.

- » A row exclusive lock (RX), also called a subexclusive table lock (SX), indicates that the transaction holding the lock has updated table rows or issued SELECT ... FOR UPDATE. An SX lock allows other transactions to query, insert, update, delete, or lock rows concurrently in the same table. Therefore, SX locks allow multiple transactions to obtain simultaneous SX and SS locks for the same table.
- » A share table lock (S) held by one transaction allows other transactions to query the table (without using SELECT ... FOR UPDATE) but allows updates only if a single transaction holds the share table lock. Multiple transactions may hold a share table lock concurrently, so holding this lock is not sufficient to ensure that a transaction can modify the table.
- » A share row exclusive table lock (SRX), also called a share-subexclusive table lock (SSX), is more restrictive than a share table lock. Only one transaction at a time can acquire an SSX lock on a given table. An SSX lock held by a transaction allows other transactions to query the table (except for SELECT ... FOR UPDATE) but not to update the table.
- » An exclusive table lock (X) is the most restrictive mode of table lock, allowing the transaction that holds the lock exclusive write access to the table. Only one transaction can obtain an X lock for a table.

Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource. Locks can occur on any Oracle user objects such as tables and rows. Also System objects that are not visible to users, such as shared data structures in the memory and data dictionary rows. These locks will stay only till the duration of the session or the transaction. Only transactions not waiting for the previously locked resources can acquire locks on now available resources. Waiting transactions continue to wait until after the original transaction commits or completely rolls back. Oracle releases all locks acquired by the statements within a transaction when an explicit or implicit commit or roll back is executed. Oracle also releases locks acquired after a save point when rolling back to the save point.

III. ORACLE LOCKING MODE

1. Share Lock Mode

This mode allows the associated resource to be shared, depending on the operations involved. Multiple users reading data can share the data, holding share locks to prevent concurrent access by a writer (who needs an exclusive lock). Several transactions can acquire share locks on the same resource.

2. Exclusive Lock Mode

This mode prevents the associated resource from being shared. This lock mode is obtained to modify data. The first transaction to lock a resource exclusively is the only transaction that can alter the resource until the exclusive lock is released.

3. Difference between Exclusive and shared lock

Table.1

Shared Lock	Exclusive Lock
Provided only to Readers of the data	Provided to writers of the data
Enables all the users to read the concurrent data at the same time, but they are not allowed to change/ write the data or obtain exclusive lock on the object	only writer, who has set the lock can change the data, and other users cannot access the locked object
Lock is released or unlocked at the end of transaction	Lock is released at the end of change in transaction

IV. ORACLE DML LOCK TYPE

- » Row Level Locks
- » Table Level Locks

Row Level Locks [TX]

All DML locks Oracle acquires automatically are row-level locks. There is No limit to the number of row locks held by a transaction. Row locking provides the lowest level of locking possible provides the best possible transaction concurrency. Readers of data do not wait for writers of the same data rows. A modified row is always locked exclusively so that other users cannot modify the row until the transaction holding the lock is committed or rolled back. If a transaction obtains a row lock for a row, the transaction also acquires a table lock for the corresponding table. The table lock prevents conflicting DDL operations that would override data changes in a current transaction. A transaction gets an exclusive DML lock for each row modified by any of the following statements: INSERT, UPDATE, DELETE, and SELECT with the FOR UPDATE clause.

Table Level Locks [TM]

A transaction acquires a table lock for DML statements such as INSERT/UPDATE/DELETE, SELECT with the FOR UPDATE, and LOCK TABLE. Reasons are to reserve DML access to the table on behalf of a transaction and prevent DDL operations. Table locks prevent an exclusive DDL lock on the same table which prevents DDL operations. Example, a table cannot be altered or dropped if any uncommitted transaction holds a table lock for it.

- » A table lock can be held in several modes: row share (RS), row exclusive (RX), share (S), share row exclusive (SRX), and exclusive (X).
- » The restrictiveness of a table lock's mode determines the modes in which other table locks on the same table can be obtained, below are the allowed Table Lock modes:

Table.2

Lock Modes						
SQL	Mode	RS	RX	S	SRX	X
SELECT		Y	Y	Y	Y	
INSERT		Y	Y	N	N	N
UPDATE	RX	Y*	Y*	N	N	N
DELETE	RX	Y*	Y*	N	N	N
SELECT FOR UPDATE OF	RX	Y*	Y*	Y*	Y*	N*
LOCK TABLE <table_name>						
IN ROW SHARE MODE	RS	Y	Y	Y	Y	N
IN ROW EXCLUSIVE MODE	RX	Y	Y	N	N	N
IN SHARE MODE	S	Y	N	Y	N	N
IN SHARE ROW EXCLUSIVE MODE	SRX	Y	N	N	N	N
IN EXCLUSIVE MODE	X	N	N	N	N	N

* Waits if another transaction has a lock

Row Share Table Lock (RS)

- » Indicates a transaction holding the lock on the table has locked rows in the table and intends to update them.
- » *Permitted Operations:* Allows other transactions to query, insert, update, delete, or lock rows concurrently in the same table. Therefore, other transactions can obtain simultaneous row share, row exclusive, share, and share row exclusive table locks for the same table.

- » *Prohibited Operations:* Lock Table in Exclusive Mode.

Row Exclusive Table Lock (RX)

- » Indicates that a transaction holding the lock has made one or more updates to rows in the table. A row exclusive table lock is acquired automatically by: INSERT, UPDATE, DELETE, LOCK TABLE.. IN ROW EXCLUSIVE MODE; A row exclusive table lock is slightly more restrictive than a row share table lock.
- » *Permitted Operations:* Allows other transactions to query, insert, update, delete, or lock rows in the same table. The row exclusive table locks allow multiple transactions to obtain simultaneous row exclusive and row share table locks in the same table.
- » *Prohibited Operations:* Prevents locking the table for exclusive reading or writing. Therefore, other transactions cannot concurrently lock the table: IN SHARE MODE, IN SHARE EXCLUSIVE MODE, or IN EXCLUSIVE MODE.

Share Table Lock (S)

- » Acquired automatically for the *table* specified in the following statement: LOCK TABLE <table> IN SHARE MODE;
- » *Permitted Operations:* Allows other transactions only to query the table, to lock specific rows with SELECT . . . FOR UPDATE, or to execute LOCK TABLE . . . IN SHARE MODE; no updates are allowed by other transactions. Multiple transactions can hold share table locks for the same table concurrently. No transaction can update the table (with SELECT.. FOR UPDATE). Therefore, a transaction that has a share table lock can update the table only if no other transaction has a share table lock on the same table.
- » *Prohibited Operations:* Prevents other transactions from modifying the same table or lock table: IN SHARE ROW EXCLUSIVE MODE, IN EXCLUSIVE MODE, or IN ROW EXCLUSIVE MODE.

Share Row Exclusive Table Lock (SRX)

- » More restrictive than a share table lock. A share row exclusive table lock is acquired for a table as follows: LOCK TABLE <table> IN SHARE ROW EXCLUSIVE MODE;
- » *Permitted Operations:* Only one transaction at a time can acquire a share row exclusive table lock on a given table. A share row exclusive table lock held by a transaction allows other transactions to query or lock specific rows using SELECT with the FOR UPDATE clause, but not to update the table.
- » *Prohibited Operations:* Prevents other transactions from obtaining row exclusive table locks and modifying the same table. A share row exclusive table lock also prohibits other transactions from obtaining share, share row exclusive, and exclusive table locks.

Exclusive Table Lock (X)

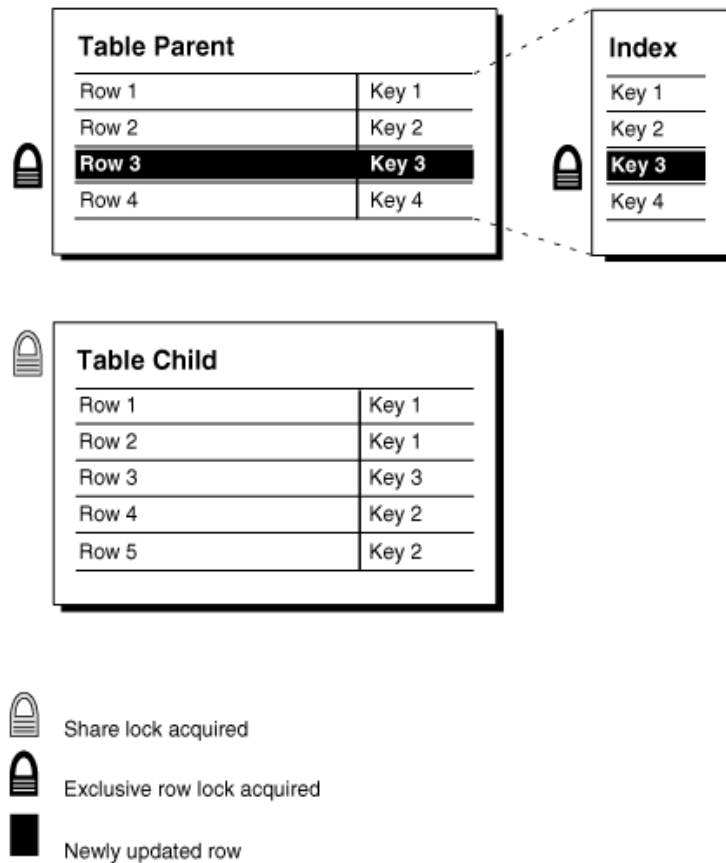
- » Most restrictive mode of table lock, allowing the transaction that holds the lock exclusive write access to the table. An exclusive table lock is acquired by: LOCK TABLE <table> IN EXCLUSIVE MODE;
- » *Permitted Operations:* Only one transaction can obtain an exclusive table lock for a table. An exclusive table lock permits other transactions only to query the table.
- » *Prohibited Operations:* Prohibits other transactions from performing any type of DML statement or placing any type of lock on the table.
- »

V. TM LOCK EXAMPLE

The Major contributor for the TM Locks or the Table Locks is the Missing Indexes on the Foreign Keys.

Locking with No Index on a Foreign Key

Non-indexed foreign keys cause DML on the primary key to get a share row exclusive table lock (also sometimes called a share-sub exclusive table lock, SSX) on the foreign key table. This prevents DML on the table by other transactions. The SSX lock is released immediately after it is obtained. If multiple primary keys are updated or deleted, the lock is obtained and released once for each row. The below diagram illustrates the locking mechanism on table with no index on the foreign key column.



Rows 1 through 4 of the parent table are indexed on keys 1 through 4, respectively. The child table is not foreign-key indexed to the parent table. Row 3 in the parent table is updated and acquires an exclusive row lock. At the same time, the child table acquires a share lock on the whole table.

Locking with Index on a Foreign Key

Indexed foreign keys cause a row share table lock (also sometimes called a subshare table lock, SS). This prevents other transactions from exclusive locking the whole table, but it does not block DML on the parent or the child table. This situation is preferable if there is any update or delete activity on the parent table while update activity is taking place on the child table. Inserts, updates, and deletes on the parent table do not require any locks on the child table, although updates and deletes will wait for row-level locks on the indexes of the child table to clear.

The below diagram illustrates the locking mechanism on table with an index on the foreign key column.

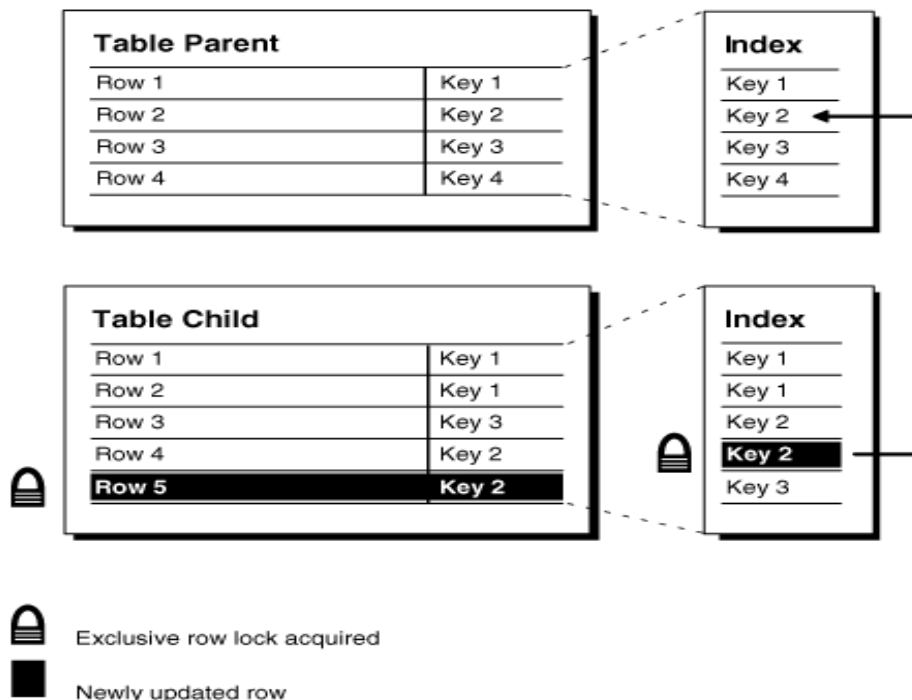


Fig.02

Rows 1 through 4 of the parent table are indexed on keys 1 through 4, respectively. Row 5 in the child table is updated and acquires an exclusive row lock (key 2).

VI. EFFECT OF LOCK

When the table gets locked out and there are other transactions that are waiting to acquire lock, turns out to a dead lock and all the waiting transactions will be indefinitely waiting on this table/resource. This would cause the DB job that is waiting for this table or the application that is waiting to Hang or wait until the deadlock is released.

a) Table Dead Lock Recovery

The Only way of recovering from the Table deadlock is to identify the transactions/sessions and kill them. If the table is used by any Applications and there are active sessions trying to lock this table, would keep the deadlocks growing on this particular table. So this table has to be taken out of access (Revoking the privileges to Application or users other than the owner) and clearing off the existing deadlocks and granting them the privilege after creating the required foreign key index.

b) Preventing the Deadlocks

The TM Locks or the Deadlocks can be prevented only by following set of design or coding standards and vigorous code/design reviews. During the design phase, if there is any new table designed with foreign key columns. The index creation on the foreign key column must be defined and the same has to be followed by the developers while developing the Table creation scripts. Also this can be caught during any of the lower environment testing or validation.

c) Identifying and fixing the cause for locks

One way of fixing the TM lock is to identify all the foreign key columns that are not having an appropriate index created on them and create the index. Also there is a possibility of having multiple update/delete queries in the same transaction. This can only be identified or fixed with the help of a code walkthrough.

VII. CONCLUSION

Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource either user objects such as tables and rows or system objects not visible to users, such as shared data structures in memory and data

dictionary rows. In all cases, Oracle automatically obtains necessary locks when executing SQL statements, so users need not be concerned with such details. Oracle automatically uses the lowest applicable level of restrictiveness to provide the highest degree of data concurrency yet also provide fail-safe data integrity. Oracle also allows the user to lock data manually

References

1. http://www.oraFAQ.com/wiki/Category:Frequently_Asked_Questions
2. http://www.dba-oracle.com/m_tm.htm
3. http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/data_int.htm#i8605
4. http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/data_int.htm#i8589

AUTHOR(S) PROFILE



Sarangkumar S. Dubey, Received Bachelor's Degree in Computer Science & Engineering and Pursuing Masters Degree in Computer Science & Engineering



Ajay P. Thakare, Received Master's Degree in Electronics Engineering and Pursuing Ph. D. His area of interest is communication system, wireless communication, Antenna system & Design.