

Oracle (PL/SQL)

Lesson 8: Locks

Lesson Objectives

- Data Concurrency and Consistency
- Locking in Oracle
- Types of Locks
- DDL & DML Locks



8.1: Data Concurrency and Consistency

Introduction

- Data Concurrency: Ensures users can access data at the same time
- Data Consistency: Ensures each user sees a consistent view of data
 - To ensure data concurrency and consistency means
 - Data must be read and modified in a consistent manner



Copyright © Capgemini 2015. All Rights Reserved 3

Data Concurrency and Consistency

In a single user environment, a user can modify data without concern for other users modifying the same data at the same time. On the other hand, in a multiuser database, various users can update the same data. Concurrency is one of the major concerns in a multiuser environment. When multiple sessions write or read data in a shared environment a database might lose its consistency. Transactions executing simultaneously must produce meaningful and consistent results. Therefore every database provides concurrency control mechanisms.

Most of the databases, the concurrency is managed through various locking mechanisms.

In this topic we see how Oracle ensures Data concurrency and consistency.

Locking Mechanism in Oracle

- Oracle manages control of access to Oracle resources and manages consistency by using Locks
- Locks
 - Used to manage access user defined resources such as tables.
 - Help to maintain transaction consistency in database.
 - Prevents one user from overwriting changes to the database made by another user

Locking Mechanism in Oracle

Oracle database provides a matured locking mechanism. It follows the rule that reading and writing processes cannot block each other, even if working on the same set of data. Each session receives a read-consistent image of the data. Thus, even if some other process has begun modifying data in the set but did not commit the changes, every subsequent session will be able to read the data just as it was before; once the changes are committed in the first session, every other session is able to see it. The locks are acquired only when the changes are being committed to the database. Oracle automatically selects the least-restrictive lock. User can choose to manually lock a resource (a table, for example). In this case, other users still might be able to access the data, depending on the type of lock deployed. Locks manage access to user defined resources and also maintain transaction consistency in the database. Locks prevents one user from overwriting changes to the database made by another user. If two user processes are executing a series of procedures to update the database with no transaction consistency, there is no guarantee that the data being updated by each user will remain the same for the life of that users transaction. However, locking mechanism provide the ability to perform transaction processing. It allows users to manipulate data freely during the transaction without worry that someone else will change the data before they are done changing it.

Lock Categories

- DML Locks
 - Row Level Locks
 - Table Level Locks
- DDL Locks
 - Exclusive DDL Locks
 - Shared DDL Locks
 - Breakable Parse Locks



Copyright © Capgemini 2015. All Rights Reserved 5

Lock Categories

Locking in Oracle is actually easy, since you really do not have to use explicit locks. Oracle does implicit locking, it is transparent and is efficient in most of the cases. But when you need to understand why a session is blocked, or to have a efficient validation or resolving deadlocks then you require explicit locking to be applied. Locks in Oracle are used for two basic purposes. The first purpose is for Data definition language (DDL) operations which are used for creating and modifying database objects. Locks are acquired for create and alter operations on database objects. The second purpose locks for DML operations which are acquired by user process to make changes to object data. They allow transactions processing to take place. Locks support transaction level read consistency by preventing two transactions from making change to the same data in a table at the same time.

Let us understand DML and DDL locks.

8.2: DML Locks

Types of DML Locks

- DML locks are applied at two levels : Row level and Table Level
 - Oracle provides six different types of locks:
 - Exclusive – allows queries on the locked table but prevents any other activity
 - Share – allows concurrent queries but prevents updates to the locked table
 - Share Row Exclusive – allows viewing of whole table but prevents others from locking the table in SHARE mode or updating rows



Copyright © Capgemini 2015. All Rights Reserved 6

The locks in Oracle are acquired on table and so all locks are table locks. However there are two different levels of locking viz. row level and table level. Row level locking is when a table lock allows other table locks to be acquired on a resource while preventing those locks from acquiring certain records or rows within table to make data changes. Table level locking is when other locks are restricted from gaining access to change data in an entire table.

Oracle provide with six different types of locks:

Exclusive : When an exclusive lock is achieved, the lock holder has exclusive access to change the table including contents of the table. Other users can only select data. But no other transaction or DML operation can acquire any type of lock until the exclusive lock is released by the holder.

Share: When one transaction has a share lock on a table, other transactions can also acquire a share, share row or share update lock on the same table. But the other transactions will have to wait until the transaction holding the share lock completes in order to complete their own transactions. If a transaction already holds a share lock on a table then no other transaction can acquire exclusive, row exclusive, or share row exclusive lock on that table.

Share Row Exclusive: A share row exclusive lock held by a transaction allows others to query rows. Other transactions can acquire share-row or share update locks on the table while a transaction holds the share row exclusive lock. Any transaction which tries to perform a DML operation will have to wait until the transaction holding the lock completes.

8.2: DML Locks

Types of DML Locks (contd..)

- Row share – allows concurrent access to the locked table but prevents users from locking the entire table for exclusive access
- Share update – Similar to ROW SHARE, and is included for backward compatibility
- Row exclusive – Similar to ROW SHARE but also prevents locking in SHARE mode. These locks are automatically when updating, inserting or deleting



Copyright © Capgemini 2015. All Rights Reserved 7

Row Share: A row share lock held by a transaction allows others to query any rows or insert new rows. Also users can update or delete rows other than the ones on which lock is being held. As a result two transactions can make changes to different rows in the same table at the same time, using row share, row exclusive, share update and share row exclusive lock. A transaction cannot acquire an exclusive lock on a table if another transaction has already acquired the row share lock on that table.

Share Update: A share update lock is acquired for making changes to data in table rows. When a transaction holds this lock, any other transaction can acquire any other type of lock on a table except for a exclusive lock. This lock is similar in behavior like the Row share lock

Row Exclusive: A row-exclusive lock held by a transaction allows other transactions to query any rows or insert new rows on the table while the row-exclusive lock is being held. In addition, transactions can concurrently process update or delete statements on rows other than those held under the row-exclusive lock in the same table. Therefore, row-exclusive locks allow multiple transactions to obtain simultaneous row-exclusive, share-row, or share-update locks for different rows in the same table. However, while one transaction holds a row-exclusive lock, no other transaction can make changes to rows that the first transaction has changed until the first transaction completes. Additionally, no transaction may obtain an exclusive, share, or share-row-exclusive lock on a table while another transaction holds a row-exclusive lock on that same table.

Acquiring Locks in Oracle

- **Syntax**

```
LOCK TABLE tablename IN lockmode MODE NOWAIT;
```

- **Example**

```
LOCK TABLE staff_master IN SHARE UPDATE mode nowait;
```

```
LOCK TABLE student_master IN EXCLUSIVE mode;
```

Add the notes here.

Deadlock

- Deadlock occurs when two or more sessions are waiting for data locked by each other, resulting in all the sessions being blocked.
- Oracle automatically deals with deadlocks by raising an ORA-00060 exception in one of the sessions.

A deadlock is the situation where you have two, or more, Oracle "sessions" competing for mutually locked resources. In other words, it is a condition where two or more users are waiting for data locked by each other. Oracle deals with deadlocks automatically raising an exception in one of the sessions.

There are a few reasons why your application may experience deadlocks, most of which are about application design or by poorly implemented locking in application code. However, there are a few situations when, due to certain architectural design decisions, you may experience deadlocks simply due to the internal mechanisms of Oracle itself.

Deadlock

- Example

- If session 1 is locking row 1, session 2 locks row 2, then session 1 attempts to lock row 2 which will block since session 2 has the lock on that row, and then session 2 attempts to lock row 1 which will also block since session 1 has the lock on that row, then session 1 is waiting for session 2, and session 2 is waiting on session 1, which of course will never be resolved



Copyright © Capgemini 2015. All Rights Reserved 10

Considering the example given on slide it looks like an infinite loop. This is where Oracle steps in automatically detecting the situation and resolves it by raising an exception for one of the sessions. But what happens to other session. The session continues to wait. The client getting the exception should either commit or rollback to release the locks and only then the operations for both the sessions can go ahead.

8.2: DDL Locks

Types of DDL Locks

- Oracle provides three types of DDL locks
 - Exclusive
 - Shared
 - Breakable Parse



Copyright © Capgemini 2015. All Rights Reserved 11

A DDL Lock protects the definition of a schema object while the object is referenced in a DDL statement. Oracle automatically acquires a DDL lock to prevent other DDL operation from referencing or altering the same object. In case a DDL lock is requested on an object that already has a DDL lock on it, the next lock request will wait. Users cannot explicitly request DDL locks. Oracle provides three types of DDL locks.

Exclusive: This lock prevents other session from obtaining a DDL or DML lock. Most DDL operation require exclusive DDL lock to prevent destructive interference with other DDL operations that might modify or reference the same schema object. For example, a DROP TABLE operation will not be possible while an ALTER TABLE operation is modifying the structure of the same table. A query against the table is not blocked. These locks last for the duration of DDL statement execution.

Shared: This lock prevents destructive interference with conflicting DDL operations, but allows data concurrency for similar DDL operations. For example, when a CREATE PROCEDURE statement is run, the containing transaction acquires share DDL locks for all referenced tables. You could concurrently create procedures that reference the same tables and acquire shared DDL locks. But no transaction can acquire an exclusive DDL lock on any referenced table. This lock lasts for the duration of DDL statement execution.

Breakable Parse Locks: This lock is held by a SQL statement or PL/SQL program unit for each schema object that it references. Parse locks are acquired so that the associated shared SQL area can be invalidated if a referenced object is altered or dropped. This lock is called as breakable parse lock because it does allow any DDL operation and can be broken to allow conflicting DDL operation. This lock is acquired in the shared pool during the parse phase of a SQL statement.

Summary

- Data Concurrency and Consistency
- Locking in Oracle
- Types of Locks
- DDL & DML Locks



Review Question

- Question 1: Which of the following is a DML lock?
 - Option 1: Share Update
 - Option 2: Row update
 - Option 3: Row Exclusive

- Question 2 : DDL locks are acquired automatically and cannot be acquired explicitly.
 - True/ False

