|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  |  | **FAQ: Detecting and Resolving Locking Conflicts and Ora-00060 errors (Doc ID 15476.1)** | [To Bottom](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\o%20To%20Bottom) |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **In this Document**   |  |  | | --- | --- | |  | [Purpose](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20PURPOSE) | |  | [Troubleshooting Steps](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20TRBLSHOOT) |  |  |  | | --- | --- | |  | [How does Oracle handle locking?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section21) | |  | [How to find the resource definitions?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section22) |  |  |  | | --- | --- | |  | [Which lock modes are required for which table action?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section23) | |  | [How compatibility of locks works](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section24) |  |  |  | | --- | --- | |  | [How to detect locking situations?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section25) | |  | Which views can be used to detect locking problems? |  |  |  | | --- | --- | |  | [Which tools are there to diagnostic locking issues?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section27) | |  | [How to resolve locking situations?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section28) |  |  |  | | --- | --- | |  | [Deadlock Situations](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section29) | |  | [Unusual locking problems](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section210) |  |  |  | | --- | --- | |  | [How to use dbms\_lock?](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section211) | |  | [Community Discussions](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20aref_section212) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=263053791469366&parent=DOCUMENT&sourceId=1443482.1&id=15476.1&_afrWindowMode=0&_adf.ctrl-state=nfgg1wd1h_165%20\l%20REF) |     **Applies to:**  Oracle Database - Enterprise Edition - Version 7.1.1.0 and later  Oracle Database - Personal Edition - Version 7.1.4.0 and later  Oracle Database - Standard Edition - Version 7.0.16.0 and later  Information in this document applies to any platform.  \*\*\*Checked for relevance on 23-JUL-2013\*\*\*  **Purpose**  This document covers frequently asked questions about detecting and resolving locking conflicts.  **Troubleshooting Steps**  **How does Oracle handle locking?**  Oracle use enqueues as locking mechanism for managing access to shared  resources. A shared resource can be a table definition, a transaction or any type of structure that represent something sharable between sessions  Each type of actions performed by Oracle sessions on those shared resources  will require a certain type of lock or lock mode (e.g. a 'select on a table' action will require that the executing session has a shared lock on the resource 'table definition' of the selected table). When conflicting  actions are occuring, Oracle will serialize the processing by putting  a number of sessions in waiting mode until the work of the blocking session has been completed.  Each enqueue represent a sharable resource. Sessions may be acquiring,  converting and releasing locks on resources in function of the work  they need to perform.   * Releasing locks are performed by the sessions  when they issue a commit or a DDL statement (i.e. implicit commit), or  by PMON if the sessions have been killed. (Remember this only pertains to row locks. Only TX, TM, and DX enqueues are held until commit. Other locks such as RT enqueues will be held until the instance is shutdown.) * Conversion is the process of  changing a lock from the mode we currently hold to a different mode.  We are allowed to convert a lock if the mode we require, is a subset of the mode we hold or is compatible with the modes already held by other sessions. Otherwise, we wait on the converters queue of the resource. * Acquiring a lock is the process of getting a lock on a resource on which we  currently do not have a lock. We are allowed to acquire a lock, if there are no  converters or waiters ahead of us and the mode we require is compatible with  the modes already held by others. Otherwise, we wait on the waiters queue  of the resource.  When a session has a lock on a resource, then it stands  in the owner queue of the resource. When a lock is released or converted,  the converters and waiters are re-checked to see if they can be acquired. The converters are processed first, then the waiters.   Row locking in Oracle is based on the TX enqueues and is known as  transactional locking. When two or more sessions are changing data on one  row of a table (DML statements on the same record), the first session will  lock the row by putting his transaction reference in the block containing  the row header. The other sessions will look at this lock information  and will wait on the transaction (i.e. the TX enqueue of the blocking session)  of the first session before proceeding. When the first session performs a commit, the TX resource will be released and the waiters will start their own locking. The waiting sessions are thus waiting on an exclusive TX  resource, but their TM resources they are holding give the objects they are in fact waiting on.  If a lock has not been acquired or converted, a deadlock check is made by  the waiting session after a timeout. For example, following situation  generates a deadlock:   * User A gets an S lock on resource 1 * User B gets an S lock on resource 2 * User A request an X lock on resource 2 and waits * User B requests an X lock on resource 1 and waits  A is waiting for B to release resource 2, which is waiting for A to release resource 1 A is indirectly waiting for A. This is a deadlock, generating a tracefile in the user\_dump\_dest and and ORA-60 in the detecting session which results in the session being killed.   The lock and resource information stands within the SGA to allow PMON to recover in the event of process failure. The PMON is responsible for releasing the locks of the crashed/killed process.  **How to find the resource definitions?**  Each resource is represented by an enqueue. An enqueue is identified by a unique name, also known as the resource name. The name has the form: . Type has two characters and represent a resource type (e.g. TM  for the table definition type). ID1 and ID2 are positive numbers and identify the resource fully (e.g. ID1 is the object\_id of the table if the resource type is "TM"). ID1 and ID2 have different meanings depending on the type of enqueue.  The description of most enqueue/resource types can be found in the appendixes of the Oracle Reference Guide.  See also:  [Document 29787.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=29787.1) VIEW: "V$LOCK" Reference Note  The most commonly known resource types are the TM, TX and UL resources:   1. The TM resource, known as the DML enqueue, is acquired during the execution of a statement when referencing a table so that the table is not dropped or altered during the execution of it. 2. The TX resource, known as the transaction enqueue, is acquired exclusive when a transaction initiates its first change and is held until the transaction does a COMMIT or ROLLBACK. Row locking is based on TX enqueues. PMON will mark a transaction status as dead in the undo segment header. At that point the TX enqueue associated with that transaction is also released. PMON will then attempt to rollback some of the changes associated with the "dead" transaction, but will then pass it to SMON to apply the remainder of the associated undo records. 3. The UL resource represent the user-defined locks defined by the DBMS\_LOCK package.     **Which lock modes are required for which table action?**  The following table describes what lock modes on DML enqueues are actually gotten for which table operations in a standard Oracle installation.  Operation                  Lock Mode LMODE Lock Description  ------------------------- --------- ----- ----------------  Select                     NULL      1     null  Select for update          SS        2     sub share  Insert                     SX        3     sub exclusive  Update                     SX        3     sub exclusive  Delete                     SX        3     sub exclusive  Lock For Update            SS        2     sub share  Lock Share                  S        4     share  Lock Exclusive              X        6     exclusive  Lock Row Share             SS        2     sub share  Lock Row Exclusive         SX        3     sub exclusive  Lock Share Row Exclusive   SSX       5     share/sub exclusive  Alter table                 X        6     exclusive  Drop table                  X        6     exclusive  Create Index                S        4     share  Drop Index                  X        6     exclusive  Truncate table              X        6     exclusive  **How compatibility of locks works**  The compatibility of lock modes are normally represented by following matrix:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | LMODE | Description | Name | NULL | SS | SX | S | SSX | X | | 0,1 | No Lock | NULL | YES | YES | YES | YES | YES | YES | | 2 | Row-Share | SS | YES | YES | YES | YES | YES | no | | 3 | Row-Exclusive | SX | YES | YES | YES | no | no | no | | 4 | Share | S | YES | YES | no | YES | no | no | | 5 | Share Row-Exc | SSX | YES | YES | no | no | no | no | | 6 | Exclusive | X | YES | no | no | no | no | no |   **How to detect locking situations?**  Different tools can be used to detect locking issues and the identity of a blocking session or sessions. A methodology using OEM Top Sessions report is explained in:  [Document 164760.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=164760.1) Detecting and Resolving Locking Conflicts using TopSessions  **Which views can be used to detect locking problems?**  A number of Oracle views permits to detect locking problems.  **V$SESSION\_WAIT**  When a session is waiting on a resource, it can be found waiting on the enqueue wait event  Example:  SELECT \* FROM V$SESSION\_WAIT WHERE EVENT = 'enqueue';   * SID identifier of session holding the lock * P1, P2, P3 determine the resource when event = 'enqueue' * SECONDS\_IN\_WAIT gives how long the wait did occurs   **V$SESSION**  Session information and row locking information   * SID, SERIAL# identifier of the session * EVENT event waited on * P1, P2, P3 determine the resource when event = 'enqueue' * # SECONDS\_IN\_WAIT gives how long the wait did occurs * LOCKWAIT address of the lock waiting, otherwise null * ROW\_WAIT\_OBJ# object identified of the object we are waiting on (object\_id of dba\_objects) * ROW\_WAIT\_FILE#, ROW\_WAIT\_BLOCK#, ROW\_WAIT\_ROW# file\_id , block\_id and  row location within block of the locked row   **V$LOCK**  List of all the locks in the system   * SID identifier of session holding the lock * TYPE, ID1 and ID2 determine the resource * LMODE and REQUEST indicate which queue the session is waiting on, as follows: LMODE > 0, REQUEST = 0 owner LMODE = 0, REQUEST > 0 acquirer LMODE > 0, REQUEST > 0 converter * CTIME time since current mode was converted * BLOCK are we blocking another lock BLOCK = 0 non blocking BLOCK = 1 blocking others   **DBA\_LOCK or DBA\_LOCKS**  Formatted view on V$LOCK (created via $ORACLE\_HOME/rdbms/admin/catblock.sql)   * SESSION\_ID == SID in V$LOCK * LOCK\_TYPE, LOCK\_ID1, LOCK\_ID2 formatted value of TYPE, ID1, ID2 from V$LOCK * MODE\_HELD and MODE\_REQUESTED formatted value of LMODE and REQUEST from V$LOCK * LAST\_CONVERT == CTIME of V$LOCK * BLOCKING\_OTHERS formatted value of BLOCK from V$LOCK   **V$TRANSACTION\_ENQUEUE**  Subset of V$LOCK for the blocking TX resources only  (same description as for the V$LOCK view)  **V$ENQUEUE\_LOCK**  Subset of V$LOCK for the system resources only and  blocked TX resources only. (same description as for the V$LOCK view)  **DBA\_DML\_LOCKS**  Subset of the V$LOCK for the DML (TM) locks only  Created via $ORACLE\_HOME/rdbms/admin/catblock.sql  Same description as the DBA\_LOCK view  **V$LOCKED\_OBJECT**  Same info as DBA\_DML\_LOCKS, but linked with the rollback and session information   * XIDUSN, XIDSLOT and XIDSQN rollback information to be linked with V$TRANSACTION * OBJECT\_ID object being locked * SESSION\_ID session id * ORACLE\_USERNAME oracle user name * OS\_USER\_NAME OS user name * PROCESS OS process id * LOCKED\_MODE lock mode   **V$RESOURCE**  List of all the currently locked resources in the system.  Each row can be associated with one or more rows in V$LOCK   * TYPE, ID1 and ID2 determine the resource   **DBA\_DDL\_LOCKS**  Has a row for each DDL lock that is being held, and one row for each outstanding request for a DDL lock.  It is subset of DBA\_LOCKS  Same description as the DBA\_LOCK view  **DBA\_WAITERS**  View that retrieve information for each session waiting on a  lock (created via $ORACLE\_HOME/rdbms/admin/catblock.sql)   * WAITING\_SESSION waiting session * HOLDING\_SESSION holding session * LOCK\_TYPE, LOCK\_ID1, LOCK\_ID2 resource locked * MODE\_HELD lock type held * MODE\_REQUESTED lock type requested   **DBA\_BLOCKERS**  View that gives the blocking sessions (created via  $ORACLE\_HOME/rdbms/admin/catblock.sql)   * HOLDING\_SESSION holding session   **Which tools are there to diagnostic locking issues?**   1. **OEM Top Session OEM Top Session permits to see all the session information. The sessions that are in status ACTIVE can be waiting on locks. By right clicking on those  sessions and choosing the 'detail' screen, you can go to the 'lock' leaflet  and find blocking sessions. For more information see** [**Document 164760.1**](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=164760.1)Detecting and Resolving Locking Conflicts using TopSessions 2. **Lock Charts in Performance Manager of Oracle Enterprise Manager OEM contains a lock graph facility named the Lock Charts in the Performance Manager that is part of the Oracle Diagnostics Pack of OEM. The Performance Manager requires the Oracle Data Gatherer to run. You should select the database from the OEM Console Navigator window. Then launch Performance Manager from the Applications menu bar. When the application comes up, select  the 'Lock Charts'. You can use the Performance Manager to kill the session. You just need to right-mouse click on the session you want to kill. Since the Lock Charts are automatically refreshed, you can watch the charts and instantly see locks being released.** 3. **Scripts for detecting locking A number of sample scripts exist to retrieve locking information.** [**Document 1020012.6**](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1020012.6)TFTS SCRIPT TO RETURN MEDIUM DETAIL LOCKING INFO Document 1020008.6 TFTS FULLY DECODED LOCKING SCRIPT  [Document 1020007.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1020007.6) SCRIPT: DISPLAY LOCKS AND GIVE SID AND SERIAL # TO KILL () [Document 1020010.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1020010.6) SCRIPT: DISPLAY SQL TEXT FROM LOCKS [Document 1020047.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1020047.6) SCRIPT: SCRIPT TO DISPLAY USER LOCK INFORMATION [Document 1020088.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1020088.6) SCRIPT: REPORT SESSIONS WAITING FOR LOCKS () 4. Using the logminer (Oracle8i onwards) All locking statements are logged in the redologs.  [Document 198828.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=198828.1) Analyzing them permits to understand locking situations   **How to resolve locking situations?**  Most locking issues are application specifics. To resolve locking contention, one needs to free the resource by:   1. Asking the HOLDER to commit or rollback 2. Killing the session which holds the lock, for example: ALTER SESSION KILL SESSION 'sid, serial#'; 3. Killing the "shadow" process directly. Although this may free the resource it is not recommended as it may prevent proper cleanup of a session. Especially avoid killing shared server or dispatcher processes at the OS level in a shared server environment since this may effect more than one session, not only the deadlocked one.. 4. If the operation is a 2pc pending transaction: ROLLBACK FORCE or COMMIT FORCE   **Deadlock Situations**  If you are encountering a deadlock situation (which may be accompanied by an "ORA-60 Deadlock Detected" error) this should generate a deadlock report trace which should enable you to diagnose the cause. ORA-60 is an application error which usually occurs because a consistent locking strategy has not been followed throughout an application.  Please refer to the following articles for more information:  [Document 18251.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=18251.1) OERR: ORA 60 "deadlock detected while waiting for resource"  Document 62365.1 What to do with "ORA-60 Deadlock Detected" Errors  [Document 62354.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=62354.1) TX Transaction locks - Example wait scenarios  **Unusual locking problems**   1. Some common locking scenarios are explained in:  [Document 62354.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=62354.1)TX Transaction and Enq: Tx - Row Lock Contention - Example wait scenarios 2. When your application has referential integrity and attempts to modify the child/parent table, Oracle will get additional locking on the parent/child table when there is NO index on the foreign key. To bypass this problem,  the most efficient way is to create indexes for all foreign key defined  in the database. See the following notes: [Document 1019527.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=1019527.6) 'TFTS CHECK FOR FOREIGN KEY LOCKING'  [Document 33453.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=33453.1) REFERENTIAL INTEGRITY AND LOCKING [Document 223303.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=223303.1) Correction to 9.2.0 foreign key constraint locking behaviour, per documentation [Bug:2546492](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=15476.1&id=2546492) 3. When your application is using DBMS\_PIPE extensively, your session can  wait for CI locks. You should increase your shared pool. 4. When statements like 'CREATE INDEX' and 'ALTER INDEX REBUILD' are issued,  Oracle behave differently in Oracle7 compared to Oracle8i. To understand  the benefit of the ONLINE option, you can find more information in  [Document 70120.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=70120.1) Locking Behavior During Index Creation or Index Rebuild 5. When a table's INITRANS is set too low , the block is full with data, and there are many concurrent DML's occurring on rows within the block, one may see a Share Lock being requested when doing a DML. To my knowledge, this is only time we grab the SHARED lock. Instead of waiting for a lock, this process is waiting for some extra space or a release of an INITRANS within the transaction layer of the block. See:  [Document 62354.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=62354.1) TX Transaction locks - Example wait scenarios 6. There are other systemwide locks that can be held at any given time. See:  [Document 102925.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=102925.1) Tracing sessions: waiting on an enqueue 7. Distributed transactions also use locks.See:  [Document 118219.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=118219.1) Detecting and Resolving Distributed Locking Conflicts   **How to use dbms\_lock?**  [Document 67680.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=15476.1&id=67680.1) PACKAGE DBMS\_LOCK Specification  **Community Discussions**  Still have questions? Use the communities window below to search for similar discussions or start a new discussion on this subject. (Window is the live community not a screenshot)  Click [here](https://community.oracle.com/message/12341061%20/l%2012341061) to open in main browser window | |