**Oracle数据库中的锁机制**

数据库是一个多用户使用的共享资源。当多个用户并发地存取数据时，在数据库中就会产生多个事务同时存取同一数据的情况。若对并发操作不加控制就可能会读取和存储不正确的数据，破坏数据库的一致性。

**在数据库中有两种基本的锁类型：排它锁（Exclusive Locks，即X锁）和共享锁（Share Locks，即S锁）。当数据对象被加上排它锁时，其他的事务不能对它读取和修改；加了共享锁的数据对象可以被其他事务读取，但不能修改。**

**根据保护对象的不同，Oracle数据库锁可以分为以下几大类：**

(1) DML lock（data locks，数据锁）：用于保护数据的完整性；

(2) DDL lock（dictionary locks，字典锁）：用于保护数据库对象的结构（例如表、视图、索引的结构定义）；

(3) Internal locks 和latches（内部锁与闩）：保护内部数据库结构；

(4) Distributed locks（分布式锁）：用于OPS（并行服务器）中；

(5) PCM locks（并行高速缓存管理锁）：用于OPS（并行服务器）中。

在Oracle中最主要的锁是DML（也可称为data locks，数据锁）锁。DML锁的目的在于保证并发情况下的数据完整性。在Oracle数据库中，DML锁主要包括TM锁和TX锁，其中TM锁称为表级锁，TX锁称为事务锁或行级锁。

意向锁的含义是如果对一个结点加意向锁，则说明该结点的下层结点正在被加锁；对任一结点加锁时，必须先对它的上层结点加意向锁。如：对表中的任一行加锁时，必须先对它所在的表加意向锁，然后再对该行加锁。这样一来，事务对表加锁时，就不再需要检查表中每行记录的锁标志位了，系统效率得以大大提高。

**TM锁（表级锁）**类型共有5种，分别称为共享锁（S锁）、排它锁（X锁）、行级共享锁（RS锁）、行级排它锁（RX锁）、共享行级排它锁（SRX锁）

当Oracle执行DML语句时，系统自动在所要操作的表上申请TM类型的锁。当TM锁获得后，系统再自动申请TX类型的锁，并将实际锁定的数据行的锁标志位进行置位。这样在事务加锁前检查TX锁相容性时就不用再逐行检查锁标志，而只需检查TM锁模式的相容性即可，大大提高了系统的效率。TM锁包括了SS、SX、S、X等多种模式，在数据库中用0－6来表示。不同的SQL操作产生不同类型的TM锁。如表1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表1 Oracle的TM锁类型 | | | |
| 锁模式 | 锁描述 | 解释 | SQL操作 |
| 0 | none |  |  |
| 1 | NULL | 空 | Select |
| 2 | SS(Row-S) | 行级共享锁，其他对象只能查询这些数据行 | Select for update、Lock for update、Lock row share |
| 3 | SX(Row-X) | 行级排它锁，在提交前不允许做DML操作 | Insert、Update、Delete、Lock row share |
| 4 | S(Share) | 共享锁 | Create index、Lock share |
| 5 | SSX(S/Row-X) | 共享行级排它锁 | Lock share row exclusive |
| 6 | X(Exclusive) | 排它锁 | Alter table、Drop able、Drop index、Truncate table 、Lock exclusive |

1. 共享锁（Share Table Lock，S）：

加锁语法：Lock Table TableName In Share Mode;

允许的操作：一个共享锁由一个事务控制，仅允许其它事务查询被锁定的表。一个有效的共享锁明确地用Select … For update形式锁定行，或执行Lock Table TableName In Share Mode语法锁定整个表，不允许被其它事务更新。

禁止的操作：一个共享锁由一个事务来控制，防止其它事务更新该表或执行下面的语句：

LOCK TABLE TableName IN SHARE ROW EXCLUSIVE MODE;

LOCK TABLE TableName IN ROW EXCLUSIVE MODE;

2. 排它锁（Exclusive Table Lock，X）：

排它锁是在锁机制中限制最多的一种锁类型，允许加排它锁的事务独自控制对表的写权限。

加锁语法：Lock Table TableName In Exclusive Mode;

允许的操作：在一个表中只能有一个事务对该表实行排它锁，排它锁仅允许其它的事务查询该表。

禁止的操作：拥有排外锁的事务禁止其它事务执行其它任何DML类型的语句或在该表上加任何其它类型的锁。

定义排它锁的语法：

LOCK TABLE TableName IN EXCLUSIVE MODE;

3. 行级锁（Row Share Table Lock，RS）：

一个行级锁（有时称为Subshare Table Lock，简称SS，子共享锁）需要该事务在被锁定行的表上用update的形式加锁。当有下面语句被执行的时候行级锁自动加在操作的表上。

SELECT . . . FROM TableName. . . FOR UPDATE OF . . . ;

LOCK TABLE TableName IN ROW SHARE MODE;

行级锁（Row Share Table Lock）在锁类型中是限制最少的，也是在表的并发程度中使用程度最高的。

允许的操作：行级共享锁由一个事务控制，允许其它事务查询、插入、更新、删除或同时在同一张表上锁定行。因此其它事务可以同时在同一张表上得到行级锁、共享行级排它锁、行级排它锁、排它锁。

禁止的操作：拥有行级锁的事务不允许其它事务执行排它锁，即：

Lock Table TableName In Exclusive Mode;

4. 行级排它锁（Row Exclusive Table Lock，RX）：

行级排它锁（亦称为Subexclusive Table Lock，简称SX，子排它锁）通常需要事务拥有的锁在表上被更新一行或多行。当有下面语句被执行的时候行级排它锁被加在操作的表上。

INSERT INTO TableName. . . ;

UPDATE TableName. . . ;

DELETE FROM TableName. . . ;

LOCK TABLE TableName IN ROW EXCLUSIVE MODE;

行级排它锁比行级锁稍微多一些限制。

允许的操作：行级排它锁由一个事务拥有允许其它事务执行查询、修改、插入、删除或同时在同一张表上锁定行。执有行级排它锁的事务允许其它事务在同一张表上同时得到共享锁和行级排它锁。

禁止的操作：行级排它锁由一个事务拥有防止其它事务手动锁定表来排除其它事务的读写权。因此，其它事务不允许在同一张表上使用以下的语句来执行锁事务。

LOCK TABLE table IN SHARE MODE;

LOCK TABLE table IN SHARE EXCLUSIVE MODE;

LOCK TABLE table IN EXCLUSIVE MODE

5. 共享行级排它锁（Share Row Exclusive Table Lock，SRX）：

共享行级排它锁有时也称共享子排它锁（Share Subexclusive Table Lock，SSX），它比共享锁有更多限制。定义共享行级排它锁的语法为：

Lock Table TableName In Share Row Exclusive Mode;

允许的操作：仅允许一个事务在某一时刻得到行级排它锁。拥有行级排它锁事务允许其它事务在被锁定的表上执行查询或使用Select … From TableName For update…来准确在锁定行而不能更新行。

禁止的操作：拥有行级排它锁的事务不允许其它事务有除共享锁外的其它形式的锁加在同一张表上或更新该表。即下面的语句是不被允许的：

LOCK TABLE TableName IN SHARE MODE;

LOCK TABLE TableName IN SHARE ROW EXCLUSIVE MODE;

LOCK TABLE TableName IN ROW EXCLUSIVE MODE;

LOCK TABLE TableName IN EXCLUSIVE MODE;

当Oracle数据库发生TX锁等待时，如果不及时处理常常会引起Oracle数据库挂起，或导致死锁的发生，产生ORA-60的错误。

**TX锁等待的分析**

Oracle数据库中一般使用行级锁。

当Oracle检测到死锁产生时，中断并回滚死锁相关语句的执行，报ORA-00060的错误并记录在数据库的日志文件alertSID.log中。同时在user\_dump\_dest下产生了一个跟踪文件，详细描述死锁的相关信息。

在日常工作中，如果发现在日志文件中记录了ora-00060的错误信息，则表明产生了死锁。这时需要找到对应的跟踪文件，根据跟踪文件的信息定位产生的原因。

|  |  |  |
| --- | --- | --- |
| 表2 数据字典视图说明 | | |
| 视图名 | 描述 | 主要字段说明 |
| v$session | 查询会话的信息和锁的信息。 | sid,serial#：表示会话信息。  program：表示会话的应用程序信息。  row\_wait\_obj#：表示等待的对象，和dba\_objects中的object\_id相对应。 |
| v$session\_wait | 查询等待的会话信息。 | sid：表示持有锁的会话信息。  Seconds\_in\_wait：表示等待持续的时间信息  Event：表示会话等待的事件。 |
| v$lock | 列出系统中的所有的锁。 | Sid：表示持有锁的会话信息。  Type：表示锁的类型。值包括TM和TX等。  ID1：表示锁的对象标识。  lmode,request：表示会话等待的锁模式的信  息。用数字0－6表示，和表1相对应。 |
| dba\_locks | 对v$lock的格式化视图。 | Session\_id：和v$lock中的Sid对应。  Lock\_type：和v$lock中的type对应。  Lock\_ID1： 和v$lock中的ID1对应。  Mode\_held,mode\_requested：和v$lock中  的lmode,request相对应。 |
| v$locked\_object | 只包含DML的锁信息，包括回滚段和会话信息。 | Xidusn,xidslot,xidsqn：表示回滚段信息。和  v$transaction相关联。  Object\_id：表示被锁对象标识。  Session\_id：表示持有锁的会话信息。  Locked\_mode：表示会话等待的锁模式的信  息，和v$lock中的lmode一致。 |

**解锁及Kill Session:**

使用下面的语法查出锁并杀掉Session。

SELECT A.SID,A.SERIAL#,A.USERNAME,B.TYPE FROM V$SESSION A,V$LOCK B WHERE A.SID=B.SID;

ALTER SYSTEM KILL SESSION 'SID,SERIAL#';