

事务处理与并发控制

单世民



大连理工大学

概述

- 并发性是一个非常重要的概念，是用来解决多个用户对同一数据进行操作的问题。特别是对于分布式数据库来说，这个特点更加突出。提高数据库的处理速度，仅依靠提高计算机的物理速度是不够的，还必须充分考虑数据库的并发性问题，提高数据库并发的效率。Oracle Database 10g系统通过使用事务和锁机制，解决了数据库的并发性问题。

事务

- 事务是数据库区别于文件系统的特性之一。所谓事务，就是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。
- 事务的特征
 - ✧ 原子性 (Atomicity)
 - ✧ 一致性 (Consistency)
 - ✧ 隔离性 (Isolation)
 - ✧ 持久性 (Durability)
- 数据库中引入数据库的主要目的就是：事务会把数据库从一种一致状态转变为另一种一致状态。

事务控制语句

- Oracle中的一个重要概念就是没有“开始事务处理”的语句。用户不能显式地开始一个事务处理。事务处理会隐式地开始与第一条修改数据的语句，或者一些要求事务处理的场合。
- 如果发出COMMIT或ROLLBACK，就会显式地结束一个事务。



*ROLLBACK TO SAVEPOINT*命令不会结束事务！

事务控制语句

- COMMIT

完整形式是COMMIT WORK，二者等价。

COMMIT会结束你的事务，并使得已做的所有修改成为永久性的（持久保存）。

- ROLLBACK

完整形式是ROLLBACK WORK，二者等价。

ROLLBACK会结束你的事务，并撤销正在进行的所有未提交的修改，把数据库块恢复到事务开始之前的状态。

事务控制语句

- 作为开发人员，用户应该使用**commit**或**rollback**显式终止用户的事务处理，否则用户正在使用的工具或环境就将为用户选择其中的一种方式。
- 例如，如果用户没有进行提交或回滚就退出了用户的SQL*Plus会话，那么SQL*Plus就会认为用户希望提交用户工作。无论用户是否想要提交工作，系统都会为用户进行提交。
- 无论事务处理的规模如何，提交都是非常快速的操作。

事务控制语句

- 提交时执行的任务

- ✧ 为用户的事务处理声称SCN(system change number, 系统改变编号)。这是Oracle的内部时钟, 可以称为数据库时间。SCN不是传统意义上的时钟, 因为它不是随着时间推移而递进。相反, 他是在事务处理提交时递进, 由Oracle在内部使用, 以对事务处理排序。
- ✧ 将所有剩余的已经缓冲的重做日志表项写入磁盘, 并且将SCN记录到在线重做日志文件中。这要由数据库后台进程LGWR执行。这一步实际上就是所谓的“提交”, 使事务处理永久地生效。
- ✧ 释放用户会话所占有的所有锁定。

事务控制语句

- **SAVEPOINT**

SAVEPOINT允许你在事务中创建一个标记点。一个事务中可以有多多个SAVEPOINT。

- **ROLLBACK TO <SAVEPOINT>**

此语句和SAVEPOINT命令配合使用，可以把事务回滚到标记点，而不回滚在此标记点之前的任何工作。

- **SET TRANSACTION**

此语句允许你设置不同的事务属性，比如事务的隔离级别以及事务是只读的还是可读写的。

事务的特征-原子性

- Oracle中事务的执行存在两种情况：
 - ✧ 构成事务的每条语句都会提交
 - ✧ 所有语句都回滚
- Oracle的这种保护可以延伸到单个的语句，即一个语句要么完全成功，要么这条语句完全回滚。
- 事务原子性的级别
 - ✧ 语句级原子性
 - ✧ 过程级原子性
 - ✧ 事务级原子性

事务的特征-原子性

- 语句级原子性

语句所导致的任何副作用（例如触发器操作）都会被认为是该语句的一部分。为了得到语句级原子性，Oracle隐式地在每个数据库调用外面包了一个SAVEPOINT。形式化表现如下例：

```
SAVEPOINT statement1;  
    delete from card where cid=2;  
If error then rollback to statement1;  
SAVEPOINT statement2;  
    delete from card where cid=8;  
If error then rollback to statement2;
```

事务的特征-原子性

- 过程级原子性

Oracle把客户提交的代码块认为是“语句”。特别的，PL/SQL匿名块也会被当作是语句。

- 事务级原子性

事务的隔离级别

- 事务的隔离级别（transaction isolation level）
一个事务对数据库的修改与并行的另外一个事务的隔离程度。
- 输出处理可能出现的3种问题：
 - ✧ 幻像读取（phantom read）
 - ✧ 不可重复读取（nonrepeatable read）
 - ✧ 脏读（dirty read）

事务的隔离级别

- SQL标准定义的隔离级别

隔离级别	脏读	不可重复读取	幻像读取
READ UNCOMMITTED	允许	允许	允许
READ COMMITTED		允许	允许
REPEATABLE READ			允许
SERIALIZABLE			



Oracle支持 READ COMMITTED和 SERIALIZABLE两种事务隔离性级别

事务的隔离级别

- `set transaction`语句可以用来设置事务处理的各种属性，例如，它的隔离层次，它是只读的还是可以进行读写的；是否需要使用特定的回滚段等。

```
set transaction isolation level serializable
```

```
set transaction isolation level read committed
```

事务的隔离级别

```
set transaction isolation level serializable
```

- 当使用了该命令之后，无论是否出现改变，数据库都会被冻结，可以完全隔离其他事务处理的影响。
- 与此命令相似的命令是：

```
set transaction read only
```

- 但两个命令存在区别：
 - ❏ 第一，**read only**会确保用户无法执行修改数据的**DML**操作，例如，**insert**、**update**或**delete**。如果执行了这样的操作，则系统就会发出错误消息，如图15-1所示的操作示意了这种过程。
 - ❏ 第二，用**read only**可以有效地将其数据库视图冻结到某个时间点。也就是说，无论数据库中的其它会话如何工作，数据库在用户的前面都会是使用**set transaction**语句时的样子。

事务的隔离级别

```
set transaction isolation level read committed
```

- 此模式是设置隔离层次时Oracle的默认操作模式，很少使用。如果用户在会话前面使用alter session命令，将用户会话的事务处理的默认隔离层次从read committed改变为serializable，那么就可能会用到这个命令。
- 与此命令相当的命令是：

```
set transaction read write
```

完整性约束与事务

- 在Oracle中，约束可以在DML语句执行之后立即生效，也可以延迟到事务处理提交时才生效。
`set constraint`语句可以允许用户在事务处理中设置延迟约束的强制模式。可以使用下面的语法格式延迟单独的约束：
- `set constraint constraint_name deferred`
- 也可以使用如下的语法格式延迟所有的约束：
- `set constraint all deferred`

完整性约束与事务

- IMMEDIATE约束模式
一般情况下，约束都是IMMEDIATE模式。在这种情况下，完整性约束会在整个*SQL语句*得到处理之后立即检查
- DEFERRABLE约束模式和级联更新
采用DEFERRABLE约束模式我们能够延迟约束的检查到引用请求或发出COMMIT时。

锁升级

- 可以在许多层次上加锁。用户可以把锁放在一行上，也可以把锁放在一个表上，甚至还可以把锁放在整个数据库上。在有些数据库系统中，锁是稀缺的资源，拥有许多锁定可以负面地影响系统的性能。在这些数据库中，用户可以发现，为了保存这些资源，用户的100个行级锁定就会转换为一个表级锁定。这种过程就是锁升级。换句话说，用户以前可能拥有很多行级锁定，现在可以拥有一个更高级别的单独锁定，可以使用一个锁来锁定以前需要锁定的所有资源。
- 但是，Oracle不会升级锁。锁升级不是数据库想要的属性。事实上，数据库支持锁升级暗示着其锁定机制存在固有的系统开销，管理上百个锁是需要处理的重要工作。在Oracle中，拥有一个锁或上百万个锁的系统开销都是相同的，并没有任何区别。

悲观锁定

- 采用悲观锁定，就是在选择数据之前，对其进行锁定。

乐观锁定

- 当更新数据时，用户可以使用一个更新将列设置为其新值，与此同时，验证数据库中的数据行是否具有与用户读取它时相同的值。

小结

- 事务
- 事务控制语句
- 事务特征
- 事务的隔离级别
- 完整性约束与事务
- 锁升级
- 悲观锁定与乐观锁定

The End