
文章

[姚鑫](#) · 六月 16, 2022 阅读大约需 4 分钟

第二章 锁定和并发控制（二）

第二章 锁定和并发控制（二）

关于零超时的说明

如上所述，如果您将 timeout 指定为 0，会添加锁。但是，如果使用零超时锁定父节点，并且已经在子节点上锁定，则忽略零超时并使用内部 1 秒超时。

删除锁

要删除默认类型的锁，请使用 LOCK 命令，如下所示：

```
LOCK - lockname
```

如果执行此命令的进程拥有具有给定名称的锁（默认类型），则此命令将删除该锁。或者，如果进程拥有多个锁（默认类型），此命令将删除其中一个。

或者删除另一种类型的锁：

```
LOCK - lockname#locktype
```

其中 locktype 是一串锁类型代码。

LOCK 命令的其他基本变体

为了完整起见，本节讨论 LOCK 命令的其他基本变体：使用它来创建简单的锁并使用它来删除所有锁。这些变化在实践中并不常见。

创建简单的锁

对于 LOCK 命令，如果省略 + 运算符，LOCK 命令首先会删除该进程持有的所有现有锁，然后尝试添加新锁。在这种情况下，锁称为简单锁而不是增量锁。一个进程可以拥有多个简单的锁，如果该进程使用如下语法同时创建它们：

```
LOCK (^MyVar1, ^MyVar2, ^MyVar3)
```

简单的锁在实践中并不常见，因为通常需要持有多个锁并在代码的不同步骤中获取它们。因此使用增量锁更实用。

但是，如果简单锁适合，请注意，可以在创建简单锁时指定 locktype 和 timeout 参数。此外，要删除一个简单的锁，可以使用带有减号 (-) 的 LOCK 命令。

移除所有锁

要删除当前进程持有的所有锁，请使用不带参数的 LOCK 命令。在实践中，以这种方式使用命令并不常见，原因有两个：

- 最好尽快释放特定的锁。
- 当进程结束时，它的所有锁都会自动释放。

锁类型

locktype 参数指定要添加或删除的锁的类型。添加锁时，请包含此参数，如下所示：

```
LOCK +lockname#locktype
```

或者在移除锁时：

```
LOCK -lockname#locktype
```

在任何一种情况下，locktype 都是用双引号括起来的一个或多个锁类型代码（以任何顺序）。请注意，如果指定 locktype 参数，则必须包含井号 (#) 以将锁名称与锁类型分开。

有四种锁类型代码，如下所示。请注意，这些不区分大小写。

- S — 添加共享锁。
- E — 添加升级锁。
- I - 添加立即解锁的锁。
- D — 添加延迟解锁的锁。

锁类型代码 D 和 I 在事务中有特殊行为。对于同一个锁名称，不能同时使用这两个锁类型代码。

独占锁和共享锁

任何锁要么是独占的（默认），要么是共享的。这些类型具有以下意义：

- 虽然一个进程拥有一个独占锁（具有给定的锁名称），但没有其他进程可以获取具有该锁名称的任何锁。
- 当一个进程拥有一个共享锁（具有给定的锁名称）时，其他进程可以获取具有该锁名称的共享锁，但没有其他进程可以获取具有该锁名称的独占锁。

排他锁的典型目的是表明打算修改一个值，并且其他进程不应尝试读取或修改该值。共享锁的典型目的是表明打算读取一个值并且其他进程不应尝试修改该值；但是，他们可以读取该值。

非升级和升级锁

任何锁也是非升级（默认）或升级。升级锁的目的是为了更容易管理大量锁，这会消耗内存并增加锁表被填满的机会。

当锁定同一阵列的多个节点时，使用升级锁。对于升级锁，如果给定进程在给定阵列的并行节点上创建了超过特定数量（默认为 1000）的锁，将替换各个锁名称并用包含锁计数的新锁替换它们。（相比之下，从未对非升级锁执行此操作。）

注意：只能为包含下标的锁名称创建升级锁。如果尝试使用没有下标的锁名称创建升级锁，会发出 <COMMAND>

错误。

锁类型总结

下表列出了所有可能的锁类型及其描述：

非升级锁	排他锁	共享锁 ("S" locks)
	locktype 省略 - 默认锁定类型	"S" — 共享锁
升级锁 ("E" locks)	"I" — 立即解锁的独占锁	"SI" — 立即解锁的共享锁
	"D" — 具有延迟解锁的排他锁	"SD" — 具有延迟解锁的共享锁
	"E" — 独占升级锁	"SE" — 共享升级锁
	"EI" — 立即解锁的独占升级锁	"SEI" — 立即解锁的共享升级锁
	"ED" — 具有延迟解锁的独占升级锁	"SED" — 具有延迟解锁的共享升级锁

对于使用多个锁的任何锁类型，锁可以是任何顺序。例如，锁类型"SI" 等价于"IS"。

[#SQL #Caché](#)

源
URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E7%AB%A0-%E9%94%81%E5%AE%9A%E5%92%8C%E5%B9%B6%E5%8F%91%E6%8E%A7%E5%88%B6%E5%BC%88%E4%BA%8C%E5%BC%89>