

文章

[姚鑫](#) · 十月 30, 2021 阅读大约需分钟

## 第六十一章 SQL命令 LOCK

### 第六十一章 SQL命令 LOCK

锁表

#### 大纲

```
LOCK [TABLE] tablename IN EXCLUSIVE MODE [WAIT seconds]
```

```
LOCK [TABLE] tablename IN SHARE MODE [WAIT seconds]
```

#### 参数

- **tablename** - 要锁定的表的名称。  
Tablename必须是已经存在的表。  
表名可以是限定的(schema.table),也可以是非限定的(table)。  
非限定表名默认模式名。  
模式搜索路径被忽略。
- **IN EXCLUSIVE MODE / IN SHARE MODE** - IN EXCLUSIVE MODE关键字短语创建一个常规的IRIS锁。  
IN SHARE MODE关键字短语创建一个共享的IRIS锁。
- **WAIT seconds** - 可选-一个整数,指定在超时前尝试获取锁的秒数。  
如果省略,则应用系统默认超时时间。

#### 描述

LOCK和LOCK TABLE是同义词。

LOCK命令显式锁定SQL表。

此表必须是已存在的表,对其具有必要的特权。

如果tablename是一个不存在的表,LOCK会失败并出现编译错误。

如果tablename是临时表,则命令执行成功,但不执行任何操作。

如果tablename是视图,则命令失败,并出现SQLCODE -400错误。

UNLOCK命令用来反转LOCK操作。

显式LOCK将持续有效,直到针对同一模式发出显式UNLOCK,或者直到进程终止。

可以使用LOCK多次锁定一个表;

必须显式解锁表,解锁次数为表被显式锁定的次数。

每个UNLOCK必须指定与相应LOCK相同的模式。

#### 权限

LOCK命令是一个特权操作。

在使用LOCK IN SHARE MODE之前,进程必须对指定的表拥有SELECT特权。  
在使用LOCK IN EXCLUSIVE MODE之前,进程必须对指定的表拥有INSERT、UPDATE或DELETE特权。  
对于IN EXCLUSIVE MODE, INSERT或UPDATE特权必须在表的至少一个字段上。  
未能持有足够的特权将导致SQLCODE -99错误(特权违反)。  
可以通过调用%CHECKPRIV命令来确定当前用户是否具有必要的特权。  
可以通过调用\$SYSTEM.SQL.Security.CheckPrivilege()方法来确定指定的用户是否具有必要的特权。

这些特权是获取锁所必需

它们没有定义锁的模

IN EXCLUSIVE

MODE锁阻止其他进程执行INSERT、UPDATE或DELETE操,而不管锁持有者是否拥有相应的特权。

## 锁模式

LOCK支持SHARE和EXCLUSIVE两种模式。

这些锁模式是相互独立的。

可以对同一个表应用SHARE锁和EXCLUSIVE锁。

EXCLUSIVE模式的锁只能通过EXCLUSIVE模式的UNLOCK解锁。

“SHARE”模式的锁只能通过“UNLOCK”解锁。

- LOCK mytable IN SHARE

MODE可以防止其他进程对mytable发出EXCLUSIVE锁,或者调用DDL操,比如DROP TABLE。

- LOCK mytable IN EXCLUSIVE MODE可以防止其他进程对mytable发出EXCLUSIVE锁或SHARE锁,也可以防止其他进程对mytable执行插入、更新或删除操,或者调用DDL操(如DROP TABLE)。

LOCK允许对表的读访问。

这两种LOCK模式都不能阻止其他进程在READ

UNCOMMITTED模式(默认的SELECT模式)对表执行SELECT操。

## 锁冲突

- 如果一个表在EXCLUSIVE模式已经被其他用户锁定,那么在任模式都不能锁定它。
- 如果一个表在SHARE模式已经被其他用户锁定,也可以在SHARE模式锁定该表,但不能在EXCLUSIVE模式锁定该表。

这些锁冲突产生SQLCODE -110错误,并生成msg,如所示:

## 锁超时

LOCK尝试获取指定的SQL表锁,直到超时。

当超时发生时,LOCK生成SQLCODE -110错误。

- 如果指定了WAIT秒数,SQL表锁定超时将在该秒数过后发生。

- 否则,当当前进程的SQL超时结束时,SQL表锁定超时发生。

可以使用\$SYSTEM.SQL.Util.SetOption()方法的ProcessLockTimeout选项为当前进程设置锁定超时。

还可以使用带有LOCK\_TIMEOUT选项的SQL命令set OPTION为当前进程设置锁定超时。

(SET OPTION不能从SQL Shell中使用。)

当前进程的SQL锁定超时默认为系统范围的SQL锁定超时。

- 否则,SQL表锁定超时发生时,系统范围的SQL超时。系统范围的缺省值是10秒。设置全系统锁定超时时间有两种方式:

- 使用\$SYSTEM.SQL.Util.SetOption()方法的LockTimeout选项。

这将立即更新进程的系统范围锁定超时默认值,并将当前进程的ProcessLockTimeout重置为这个新的系统范围值。

设置系统范围的锁超时对当前运行的其他进程的ProcessLockTimeout设置没有影响。

- 使用管理门户,选择系统管理、配置、SQL和对象设置、SQL。

查看和编辑当前的锁定超时(秒)设置。

这将更改在配置更改后启动的新进程的系统范围锁定超时默认值。

它对当前运行的进程没有影响。

要返回当前系统范围的锁超时值，调用\$SYSTEM.SQL.Util.GetOption("LockTimeout")方法。

要返回当前进程的锁超时值，请调用\$SYSTEM.SQL.Util.GetOption("ProcessLockTimeout")方法。

## 事务处理

LOCK操作不是事务的一部分。

回滚发出LOCK的事务不会释放锁。

UNLOCK可以定义为在当前事务结束时发生，或者立即发生。

## 其他锁定操作

许多DDL操作，包括ALTER TABLE和DELETE TABLE，都需占表锁。

INSERT、UPDATE和DELETE命令也执行锁定。

默认情况下，它们在当前事务期间锁定在记录级别；

如果其中一个命令锁定了足够的记录(默认设置为1000)，那么锁将自动提升为表锁。

LOCK命令允许显式地设置表级锁，使能够更好地控制数据资源的锁。

INSERT、UPDATE或DELETE可以通过指定%NOLOCK关键字来覆盖LOCK。

带有LOCK\_TIMEOUT选项的SQL SET

OPTION设置当前进程的INSERT、UPDATE、DELETE或SELECT操作的超时时间。

SQL支持\$SYSTEM.SQL.Util.SetOption()方法的CachedQueryLockTimeout选项。

## 示例

下面的嵌入式SQL示例创建一个表，然后锁定它：

```
ClassMethod Lock()
{
    n SQLCODE, %msg
    &sql(
        CREATE TABLE mytest (
            ID NUMBER(12,0) NOT NULL,
            CREATE_DATE DATE DEFAULT CURRENT_TIMESTAMP(2),
            WORK_START DATE DEFAULT SYSDATE
        )
    )
    if SQLCODE = 0 {
        w !, "???"
    } elseif SQLCODE = -201 {
        w !, "?????"
    } else {
        w !, "SQL?????????: ", SQLCODE
        q
    }
}
```

```
ClassMethod Lock1()
{
    n SQLCODE, %msg
```

```
s x = $zh
&sql(
    LOCK mytest IN EXCLUSIVE MODE WAIT 4
)
if SQLCODE = 0 {
    w !,"??"
} elseif SQLCODE = -110 {
    w "?? ", $ZHOROLOG - x, " ?"
    w !,"????????",!,%msg
} else {
    w !,"??: ",SQLCODE,!,%msg
}
}
```

从管理门户运行的SQL程序生成一个进程，该进程在程序执行时立即终止。

因此，锁几乎立即被释放。

因此，要观察锁冲突，请在运行相同名称空间中的SQL Shell的终端中发出lock mytest IN EXCLUSIVE MODE命令。

然后运行上面的嵌入式SQL锁定程序。

在排队模式从终端SQL Shell发出一个UNLOCK mytest。

然后重新运行上面的嵌入式SQL锁定程序。

#### [#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E5%8D%81%E4%B8%80%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-lock>