

文章

[姚鑫](#) · 十一月 5, 2021 阅读大约需 4 分钟

第六十七章 SQL命令 ROLLBACK

第六十七章 SQL命令 ROLLBACK

回滚事务。

大纲

ROLLBACK [WORK]

ROLLBACK TO SAVEPOINT pointname

参数

- pointname - 作为标识符指定的现有保存点的名称。

描述

ROLLBACK语句将回滚事务，撤消已执行但未提交的工作，减少\$TLEVEL事务级别计数器，并释放锁。ROLLBACK用于将数据库恢复到以前的一致状态。

- ROLLBACK回滚当前事务期间完成的所有工作，将\$TLEVEL事务级别计数器重置为0，并释放所有锁。这会将数据库恢复到事务开始之前的状态。ROLLBACK和ROLLBACK WORK是等价的语句；这两个版本都支持兼容性。
- ROLLBACK TO SAVEPOINT pointname将回滚自指定保存点以来已完成的所有工作，并按未完成的保存点的数量递减\$TLEVEL事务级别计数器。当所有保存点都被回滚或提交，并且事务级别计数器重置为零时，事务就完成了。如果指定的保存点不存在，或者已经回滚，ROLLBACK将发出SQLCODE -375错误并回滚整个当前事务。

回滚到SAVEPOINT必须指定一个点名。

如果不这样做，将导致SQLCODE -301错误。

如果事务操作未能成功完成，则会发出SQLCODE -400错误。

不回滚

ROLLBACK操作不影响以下项目：

- 回滚不会减少默认类的IDKey计数器。

IDKey是由\$INCREMENT(或\$SEQUENCE)自动生成的，它维护一个独立于SQL事务的计数。

- 回滚不会逆转缓存查询的创建、修改或删除。

这些操作不被视为事务的一部分。

- 在事务中发生的DDL操作或调优表操作可以创建并运行临时例程。

这个临时例程与缓存查询一样被处理。

也就是说，临时例程的创建、编译和删除不被视为事务的一部分。

临时例程的执行被认为是事务的一部分。

回滚日志

提示回滚发生的消息和回滚操作中遇到的错误都记录在MGR目录下的Messages.log文件中。您可以通过“管理门户系统操作”、“系统日志”、“消息日志”选项查看“Messages.Log”。

事务挂起

%SYSTEM.Process类的TransactionsSuspending()方法可用于挂起和恢复系统范围内的所有当前事务。挂起事务会挂起更改的日志记录。因此，如果在当前事务期间发生事务挂起，则ROLLBACK不能回滚在事务挂起期间所做的任何更改；但是，回滚将回滚在事务挂起生效之前或之后在当前事务期间所做的任何更改。

ObjectScript事务命令

ObjectScript和SQL TRANSACTION命令完全兼容且可互换，但以下情况除外：

如果没有当前事务，ObjectScript TSTART和SQL START TRANSACTION都会启动事务。但是，START TRANSACTION不支持嵌套事务。因此，如果需要(或可能需要)嵌套事务，最好使用TSTART启动事务。如果需要与SQL标准兼容，请使用START TRANSACTION。

ObjectScript事务处理为嵌套事务提供有限的支持。SQL事务处理为事务内的保存点提供支持。

```
ClassMethod ROLLBACK()  
{  
    &sql(SET TRANSACTION %COMMITMODE EXPLICIT)  
    w !,"Set transaction mode, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
    &sql(START TRANSACTION)  
    w !,"Start transaction, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
    &sql(SAVEPOINT a)  
    w !,"Set Savepoint a, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
    &sql(SAVEPOINT b)  
    w !,"Set Savepoint b, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
    &sql(SAVEPOINT c)  
    w !,"Set Savepoint c, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
    &sql(ROLLBACK)  
    w !,"Rollback transaction, SQLCODE=",SQLCODE  
    w !,"Transaction level=", $TLEVEL  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).ROLLBACK()
```

```
Set transaction mode, SQLCODE=0  
Transaction level=0  
Start transaction, SQLCODE=0  
Transaction level=1  
Set Savepoint a, SQLCODE=0  
Transaction level=2  
Set Savepoint b, SQLCODE=0
```

```
Transaction level=3
Set Savepoint c, SQLCODE=0
Transaction level=4
Rollback transaction, SQLCODE=0
Transaction level=0
```

以下嵌入式SQL示例演示了回滚到保存点名称如何将事务级别(\$TLEVEL)恢复到紧靠指定保存点之前的级别：

```
ClassMethod ROLLBACK1()
{
    &sql(SET TRANSACTION %COMMITMODE EXPLICIT)
    w !,"Set transaction mode, SQLCODE=",SQLCODE
    w !,"Transaction level=", $TLEVEL
    &sql(START TRANSACTION)
    w !,"Start transaction, SQLCODE=",SQLCODE
    w !,"Transaction level=", $TLEVEL
    &sql(SAVEPOINT a)
    w !,"Set Savepoint a, SQLCODE=",SQLCODE
    w !,"Transaction level at a=", $TLEVEL
    &sql(SAVEPOINT b)
    w !,"Set Savepoint b, SQLCODE=",SQLCODE
    w !,"Transaction level at b=", $TLEVEL
    &sql(ROLLBACK TO SAVEPOINT b)
    w !,"Rollback to b, SQLCODE=",SQLCODE
    w !,"Rollback transaction level=", $TLEVEL
    &sql(SAVEPOINT c)
    w !,"Set Savepoint c, SQLCODE=",SQLCODE
    w !,"Transaction level at c=", $TLEVEL
    &sql(SAVEPOINT d)
    w !,"Set Savepoint d, SQLCODE=",SQLCODE
    w !,"Transaction level at d=", $TLEVEL
    &sql(COMMIT)
    w !,"Commit transaction, SQLCODE=",SQLCODE
    w !,"Transaction level=", $TLEVEL
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLCommand).ROLLBACK1()
```

```
Set transaction mode, SQLCODE=0
Transaction level=0
Start transaction, SQLCODE=0
Transaction level=1
Set Savepoint a, SQLCODE=0
Transaction level at a=2
Set Savepoint b, SQLCODE=0
Transaction level at b=3
Rollback to b, SQLCODE=0
Rollback transaction level=2
Set Savepoint c, SQLCODE=0
Transaction level at c=3
Set Savepoint d, SQLCODE=0
Transaction level at d=4
Commit transaction, SQLCODE=0
Transaction level=0
```

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E5%8D%81%E4%B8%83%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-rollback>