

Artigo

[Danusa Calixto](#) · Ago. 12, 2022 4min de leitura

Cuidado ao combinar OO e SQL

A combinação da sintaxe de objetos com SQL é um dos recursos legais no Object Script. No entanto, em um caso, forneceu resultados estranhos. Portanto, decidi isolar esse caso e descrevê-lo aqui.

Digamos que você precisa escrever um classmethod que atualiza uma única propriedade no disco. Geralmente, eu escreveria isso usando SQL, desta forma:

```
ClassMethod ActivateSQL(customerId) as %Status
{
    &sql(Update Test.Customer Set Active=1 Where ID=:customerId)
    If SQLCODE=0 {
        Set exception = ##class(%Exception.SQL).CreateFromSQLCODE(SQLCODE, $Get(%msg))
        Quit exception.AsStatus()
    } Else {
        Quit $$$OK
    }
}
```

e chamaria esse classmethod sempre que necessário no meu aplicativo.

No entanto, se o código do aplicativo tiver a instância aberta durante a chamada do classmethod e executar %Save em seguida, ele substituirá as atualizações realizadas no classmethod:

```
Set objCust=##class(Test.Customer).%OpenId(id)
Do objCust.ActivateSQL(id)
Set objCust.Name = "something"
Set sc = objCust.%Save()
```

Ao mudar a ordem das linhas, o problema estaria resolvido, mas você precisa tomar cuidado com este tipo de combinação:

```
Do ##class(Test.Customer).ActivateSQL(id)
Set objCust=##class(Test.Customer).%OpenId(id)
Set objCust.Name = "something"
Set sc = objCust.%Save()
```

Quando classmethod fosse escrito usando sintaxe de OO assim:

```
ClassMethod ActivateOO(customerId) as %Status
{
    Set objCust = ##class(Test.Customer).%OpenId(customerId)
    Set objCust.Active = 1
    Quit objCust.%Save()
}
```

não haveria problema, já que a instância aberta no código da chamada e a instância aberta em classmethod apontariam para a mesma instância na memória.

(Além de uma penalidade no desempenho, já que abrir uma instância com várias propriedades para atualizar uma propriedade é mais demorado do que uma atualização do SQL)

Portanto, para concluir: tenha cuidado ao abrir instâncias "muito longas" no seu código se também estiver usando SQL.

Anexei uma classe de teste completa. Se quiser ver por si mesmo, chame `Do ##class(Test.Customer).Test(0)` para ver o código usando somente OO e `.Test(1)`, usando SQL (observe que a atualização do SQL é substituída)

Qualquer comentário é bem-vindo!

```
Class Test.Customer Extends %Persistent
{
    Property Name As %String;
    Property Active As %Boolean;
    ClassMethod ActivateSQL(customerId) As %Status
    {
        #Dim exception

        &sql(Update Test.Customer Set Active=1 Where ID=:customerId)
        If SQLCODE'=0 {
            Set exception = ##class(%Exception.SQL
).CreateFromSQLCODE(SQLCODE, $Get(%msg))
            Quit exception.AsStatus()
        }

        &sql(Select Name, Active Into :name, :active From
Test.Customer Where ID = :customerId)
        Write !,"Result After SQL Update : ",!
        Write "Name    : ",name,!
        Write "Active : ",active,!!
        Quit
    }

    ClassMethod ActivateOO(customerId) As %Status
    {
        #Dim objCust as Test.Customer
        #Dim sc as %Status
        Set objCust = ##class(Test.Customer).%OpenId(customerId)
        Set objCust.Active = 1
        Set sc = objCust.%Save()
        If sc'=$$$OK Quit sc
        &sql(Select Name, Active Into :name, :active From
Test.Customer Where ID = :customerId)
        Write !,"Result After %Save : ",!
        Write "Name    : ",objCust.Name,!
        Write "Active : ",objCust.Active,!!
        Quit
    }
}
```

```
ClassMethod Test(mode = 0)
{
    #Dim objCust as Test.Customer
    #Dim sc as %Status
    #Dim id as %Integer
    ;Create an instance and keep the id in memory
    Set objCust = ##class(Test.Customer).%New()
    Set objCust.Name = "Danny"
    Set sc = objCust.%Save() If sc'=1 Write "Could not save",!
    Set id = objCust.%Id()
    Kill objCust

    ;Open and display the created instance
    Set objCust=##class(Test.Customer).%OpenId(id)
    Write "Name    : ",objCust.Name,!
    Write "Active : ",objCust.Active,!

    ;Chame um classmethod que atualize o id com SQL ou OO
    If mode=0 {
        Do objCust.ActivateOO(id)
    } else {
        Do objCust.ActivateSQL(id)
    }
    ;Mude a instância (que ainda está na memória)
    Set objCust = ##class(Test.Customer).%OpenId(id)
    Set objCust.Name = objCust.Name_ - "edited"
    Set sc = objCust.%Save() If sc'=1 Write "Could not save",!
    Write "Name    : ",objCust.Name,!
    Write "Active : ",objCust.Active,!

;a atualização do SQL em classmethod é substituída pela instância que ainda estava na
memória
    ;Abra e demonstre a instância criada
    Kill objCust
    Set objCust = ##class(Test.Customer).%OpenId(id)
    Write "Name    : ",objCust.Name,!
    Write "Active : ",objCust.Active,!
}
}
```

[#Guia de Codificação](#) [#ObjectScript](#) [#Caché](#)

URL de origem: <https://pt.community.intersystems.com/post/cuidado-ao-combinar-oo-e-sql>