

文章

姚鑫 · 三月 25, 2021 阅读大约需1 分钟

第十三章 使用动态SQL(三)

第十三章 使用动态SQL(三)

执行SQL语句

有两种使用%SQL.Statement类执行SQL语句的方法：

- %Execute(), 它执行以前使用%Prepare()或%PrepareClassQuery()准备的SQL语句。
- %ExecDirect(), 它同时准备和执行一条SQL语句。

也可以通过使用\$SYSTEM.SQL.Execute()方法执行SQL语句而无需创建对象实例。此方法既准备又执行SQL语句。它创建一个缓存的查询。下面的终端示例显示Execute()方法：

```
USER>SET topnum=5
USER>SET rset=$SYSTEM.SQL.Execute("SELECT TOP :topnum Name, Age FROM Sample.Person")
USER>DO rset.%Display()
```

%Execute()

准备查询后，可以通过调用%SQL.Statement类的%Execute()实例方法来执行查询。对于非SELECT语句，%Execute()调用所需操作（例如执行INSERT）。对于SELECT查询，%Execute()生成一个结果集，用于后续遍历和数据检索。例如：

```
SET rset = tStatement.%Execute()
```

%Execute()方法为所有SQL语句设置%SQL.StatementResult类属性SQLCODE和%Message。
%Execute()设置其他%SQL.StatementResult属性如下所示：

- INSERT, UPDATE, INSERT或UPDATE, DELETE和TRUNCATE TABLE语句将%ROWCOUNT设置为受操作影响的行数。TRUNCATE TABLE无法确定删除的实际行数，因此将%ROWCOUNT设置为-1。

INSERT, UPDATE, INSERT OR UPDATE和DELETE将%ROWID设置为最后一行插入，更新或删除的记录的行ID值。如果该行未插入，更新或删除任何记录，则%ROWID是未定义的，或被设置为其先前值。TRUNCATE TABLE没有设置%ROWID。

- SELECT语句在创建结果集时会设置%ROWCOUNT属性为0。当程序遍历结果集的内容（例如，使用%Next()方法）时，%ROWCOUNT会增加。%Next()返回1表示它位于一行上，返回0表示它位于最后一行之后（在结果集的末尾）。如果光标位于最后一行之后，则%ROWCOUNT的值指示结果集中包含的行数。

如果SELECT查询仅返回聚合函数，则每个%Next()都将设置%ROWCOUNT =

1. 即使表中没有数据,第一个%Next()始终设置%SQLCODE =
0. 任后续%Next()都会设置%SQLCODE = 100并设置%ROWCOUNT = 1。

SELECT还设置%CurrentResult和%ResultColumnCount。SELECT未设置%ROWID。

可以使用ZWRITE返回所有%SQL.StatementResult类属性值。

具有输入参数的%Execute()

%Execute()方法可以采用一个或多个与准备的SQL语句中的输入参数(以“?”表示)相对应的参数。%Execute()参数对应于“?”的顺序字符出现在SQL语句中:第一个参数用于第一个“?”,第二个参数用于第二个“?”,依此类推。每个%Execute()参数以逗号分隔。可以通过指定占位符符号来省略参数值。%Execute()参数的数量必须与“?”相对应输入参数。如果%Execute()参数少于或大于相应的“?”输入参数,执行失败,并且%SQLCODE属性设置为SQLCODE -400错误。

可以使用输入参数为SELECT列表和其他查询子句(包括TOP子句和WHERE子句)提供文字值或表达式。不能使用输入参数为SELECT列表或其他查询子句提供列名或列名别名。

当指定为显式%Execute()参数时,最大输入参数数为255。使用可变长度数组%Execute(vals ...)指定时,最大输入参数数为380。

在执行Prepare之后,可以使用Prepare参数元数据来返回?的计数和所需数据类型。输入参数。可以使用%GetImplementationDetails()方法返回?的列表。在准备好的查询中输入参数,并在查询文本中使用?输入参数显示在上文中。

以下ObjectScript示例使用两个输入参数执行查询。它在%Execute()方法中指定输入参数值(21和26)。

```

/// d ##class(PHA.TEST.SQL).PrepareClassQuery7()
ClassMethod PrepareClassQuery7()
{
    SET tStatement = ##class(%SQL.Statement).%New(1)
    SET tStatement.%SchemaPath = "MyTests,Sample,Cinema"
    SET myquery=2
    SET myquery(1)="SELECT Name,DOB,Age FROM Person"
    SET myquery(2)="WHERE Age > ? AND Age < ? ORDER BY Age"
    SET qStatus = tStatement.%Prepare(.myquery)
    IF qStatus'=1 {
        WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT
    }
    SET rset = tStatement.%Execute(21,26)
    WRITE !,"Execute OK: SQLCODE=",rset.%SQLCODE,!!
    DO rset.%Display()
    WRITE !,"End of data: SQLCODE=",rset.%SQLCODE
}

```

```
DHC-APP>d ##class(PHA.TEST.SQL).PrepareClassQuery7()
```

```
Execute OK: SQLCODE=0
```

Name	DOB	Age
Van De Griek,Dick U.	1998-12-21	22
Peterson,Kirsten R.	1997-12-13	23
Van De Griek,Phil S.	1996-09-26	24
Wijnschenk,Lydia G.	1997-01-17	24
Xiang,Kirsten U.	1996-08-06	24
Schaefer,Usha G.	1995-09-16	25

```
Peterson,Sophia A.      1995-12-05      25
Petersburg,Bill O.     1995-10-23      25
```

```
8 Rows(s) Affected
End of data: SQLCODE=100
```

下面的ObjectScript示例执行相同的查询。%Execute()方法形式参数列表使用可变长度数组(dynd...)指定不确定数量的输入参数值,在这种情况下,为dynd数组的标,dynd变量设置为2以指示两个标值。

```
/// d ##class(PHA.TEST.SQL).PrepareClassQuery8()
ClassMethod PrepareClassQuery8()
{
    SET tStatement = ##class(%SQL.Statement).%New(1)
    SET tStatement.%SchemaPath = "MyTests,Sample,Cinema"
    SET myquery=2
    SET myquery(1)="SELECT Name,DOB,Age FROM Person"
    SET myquery(2)="WHERE Age > ? AND Age < ? ORDER BY Age"
    SET dynd=2,dynd(1)=21,dynd(2)=26
    SET qStatus = tStatement.%Prepare(.myquery)
    IF qStatus'=1 {
        WRITE "%Prepare failed:"
        DO $System.Status.DisplayError(qStatus)
        QUIT
    }
    SET rset = tStatement.%Execute(dynd...)
    WRITE !,"Execute OK: SQLCODE=",rset.%SQLCODE,!!
    DO rset.%Display()
    WRITE !,"End of data: SQLCODE=",rset.%SQLCODE
}
```

```
DHC-APP>d ##class(PHA.TEST.SQL).PrepareClassQuery8()
```

```
Execute OK: SQLCODE=0
```

```
Name      DOB      Age
Van De Griek,Dick U.    1998-12-21    22
Peterson,Kirsten R.    1997-12-13    23
Van De Griek,Phil S.   1996-09-26    24
Wijnschenk,Lydia G.    1997-01-17    24
Xiang,Kirsten U.      1996-08-06    24
Schaefer,Usha G.      1995-09-16    25
Peterson,Sophia A.     1995-12-05    25
Petersburg,Bill O.    1995-10-23    25
```

```
8 Rows(s) Affected
End of data: SQLCODE=100
```

可以对准备好的结果执行多个%Execute()操作。这使您可以多次运行查询,并提供不同的输入参数值。不必在%Execute()操作之间关闭结果,如以下示例所示:

```
/// d ##class(PHA.TEST.SQL).PrepareClassQuery9()
ClassMethod PrepareClassQuery9()
{
    SET myquery="SELECT Name,SSN,Age FROM Sample.Person WHERE Name %STARTSWITH ?"
```

```

SET tStatement = ##class(%SQL.Statement).%New()
SET qStatus = tStatement.%Prepare(myquery)
IF qStatus'=1 {
    WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT
}
SET rset = tStatement.%Execute("A")
DO rset.%Display()
WRITE !,"End of A data",!!
SET rset = tStatement.%Execute("B")
DO rset.%Display()
WRITE !,"End of B data"
}

```

```
DHC-APP>d ##class(PHA.TEST.SQL).PrepareClassQuery9()
```

Name	SSN	Age	
Alton,Martin S.	624-25-8488	47	
Ahmed,Elmo X.	950-40-6135	77	
Anderson,Mario L.	604-10-9256	77	
Adams,Diane F.	640-77-5933	9	
Anderson,Valery N.	882-50-4971	27	
Alton,Phil T.	785-37-8519	68	
Adams,Susan E.	947-66-8684	52	

```
7 Rows(s) Affected
End of A data
```

Name	SSN	Age	
Bukowski,Mario V.	683-32-4214	85	
Bachman,Susan O.	102-59-3932	88	
Bush,Jules K.	547-97-7915	13	
Basile,Filomena X.	888-66-1725	86	
Browne,Robert X.	308-58-1444	82	
Burroughs,Barbara H.	627-56-2213	86	
Beatty,Molly Z.	794-64-5615	54	

```
7 Rows(s) Affected
End of B data
```

使用TRY / CATCH处理%Execute错误

可以在TRY块结构内执行Dynamic SQL,将运行时错误传递给关联的CATCH块异常处理程序。对于%Execute()错误,可以使用%Exception.SQL类创建一个异常实例,然后将其扔到CATCH异常处理程序中。

下面的示例在发生%Execute()错误时创建一个SQL异常实例。在这种情况下,错误是数量之间的不匹配。输入参数(1)和%Execute()参数的数量(3)。它将%SQLCODE和%Message属性(作为Code和Data)抛出到CATCH异常处理程序中。异常处理程序使用%IsA()实例方法测试异常类型,然后显示%Execute()错误:

```

/// d ##class(PHA.TEST.SQL).SQLTRY()
ClassMethod SQLTRY()
{
    TRY {
        SET myquery = "SELECT TOP ? Name,DOB FROM Sample.Person"
        SET tStatement = ##class(%SQL.Statement).%New()
        SET qStatus = tStatement.%Prepare(myquery)
        IF qStatus'=1 {

```

```

        WRITE "%Prepare failed:"
        DO $System.Status.DisplayError(qStatus) QUIT
    }
    SET rset = tStatement.%Execute(7,9,4)
    IF rset.%SQLCODE=0 {
        WRITE !,"Executed query",!
    } ELSE {
        SET badSQL=##class(%Exception.SQL).%New(,rset.%SQLCODE,,rset.%Message)
        THROW badSQL
    }
    DO rset.%Display()
    WRITE !,"End of data"
    RETURN
}
CATCH exp {
    WRITE "In the CATCH block",!
    IF 1=exp.%IsA("%Exception.SQL") {
        WRITE "SQLCODE: ",exp.Code,!
        WRITE "Message: ",exp.Data,! }
    ELSE { WRITE "Not an SQL exception",! }
    RETURN
}
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).SQLTRY()
In the CATCH block
SQLCODE: -400
Message: Dynamic SQL Execute, more parameter values passed than are specified in the
dynamic statement

```

%ExecDirect()

%SQL.Statement类提供%ExecDirect()类方法,该方法可以在单个操作中准备和执行查询。它可以准备指定的查询(如%Prepare())或现有的类查询(如%PrepareClassQuery())。

%ExecDirect()准备并执行指定的查询:

```

/// d ##class(PHA.TEST.SQL).ExecDirect()
ClassMethod ExecDirect()
{
    SET myquery=2
    SET myquery(1)="SELECT Name,Age FROM Sample.Person"
    SET myquery(2)="WHERE Age > 21 AND Age < 30 ORDER BY Age"
    SET rset = ##class(%SQL.Statement).%ExecDirect(,myquery)
    IF rset.%SQLCODE=0 {
        WRITE !,"ExecDirect OK",!!
    } ELSE {
        WRITE !,"ExecDirect SQLCODE=",rset.%SQLCODE,! ,rset.%Message QUIT
    }
    DO rset.%Display()
    WRITE !,"End of data: SQLCODE=",rset.%SQLCODE
}

```

```
DHC-APP> d ##class(PHA.TEST.SQL).ExecDirect()
```

ExecDirect OK

Name	Age
Van De Griek,Dick U.	22
Peterson,Kirsten R.	23
Van De Griek,Phil S.	24
Wijnschenk,Lydia G.	24
Xiang,Kirsten U.	24
Schaefer,Usha G.	25
Peterson,Sophia A.	25
Petersburg,Bill O.	25
Ng,Josephine Z.	26
Munt,Valery W.	26
Ingleman,Martin T.	26
Eno,Diane U.	26
Pascal,Kim P.	27
Ipsen,Jane A.	27
Anderson,Valery N.	27
Gomez,Mo Q.	27
Xerxes,Angelo P.	28
Young,Barbara N.	29

18 Rows(s) Affected

End of data: SQLCODE=100

%ExecDirect()准备并执行现有的类查询:

```

/// d ##class(PHA.TEST.SQL).ExecDirect1()
ClassMethod ExecDirect1()
{
    SET mycallq = "?=CALL Sample.PersonSets('A','NH')"
    SET rset = ##class(%SQL.Statement).%ExecDirect(,mycallq)
    IF rset.%SQLCODE=0 {
        WRITE !,"ExecDirect OK",!!
    } ELSE {
        WRITE !,"ExecDirect SQLCODE=",rset.%SQLCODE,!,"rset.%Message QUIT
    }
    DO rset.%Display()
    WRITE !,"End of data: SQLCODE=",rset.%SQLCODE
}

```

可以将输入参数值指定为%ExecDirect()类方法的三个参数和后续参数,如以下示例所示:

```

/// d ##class(PHA.TEST.SQL).ExecDirect2()
ClassMethod ExecDirect2()
{
    SET myquery=2
    SET myquery(1)="SELECT Name,Age FROM Sample.Person"
    SET myquery(2)="WHERE Age > ? AND Age < ? ORDER BY Age"
    SET rset = ##class(%SQL.Statement).%ExecDirect(,myquery,12,20)
    IF rset.%SQLCODE'=0 {
        WRITE !,"1st ExecDirect SQLCODE=",rset.%SQLCODE,!,"rset.%Message QUIT
    }
    DO rset.%Display()
}

```

```

WRITE !,"End of teen data",!!
SET rset2 = ##class(%SQL.Statement).%ExecDirect(,myquery,19,30)
IF rset2.%SQLCODE'=0 {
    WRITE !,"2nd ExecDirect SQLCODE=",rset2.%SQLCODE,!,"rset2.%Message QUIT
}
DO rset2.%Display()
WRITE !,"End of twenties data"
}

```

```
DHC-APP> d ##class(PHA.TEST.SQL).ExecDirect2()
```

```

Name      Age
Bush,Jules K.    13
...
Eastman,Howard K.    18

```

```

9 Rows(s) Affected
End of teen data

```

```

Name      Age
Ingrahm,Susan N.    20
...
Young,Barbara N.    29

```

```

20 Rows(s) Affected
End of twenties data

```

`%ExecDirect()` 输入参数对应于“?”的顺序字符出现在SQL语句中:第三个参数用于第一个“?”,第四个参数用于第二个“?”,依此类推。可以通过指定占位符逗号来省略参数值。如果`%ExecDirect()` 输入参数少于相应的“?”输入参数,则使用默认值(如存在)。

在面前的示例中,第一个`%ExecDirect()` 指定所有三个“?”输入参数,第二个`%ExecDirect()` 仅指定第二个?输入参数,并省略第一个和第三个。它使用第三个输入参数的默认`Sample.PersonSets()` ('MA'):

```

/// d ##class(PHA.TEST.SQL).ExecDirect3()
ClassMethod ExecDirect3()
{
    SET mycall = "?=CALL Sample.PersonSets(?,?)"
    SET rset = ##class(%SQL.Statement).%ExecDirect(,mycall,"","A","NH")
    IF rset.%SQLCODE'=0 {WRITE !,"1st ExecDirect SQLCODE=",rset.%SQLCODE,!,"rset.%Message QUIT}
    DO rset.%Display()
    WRITE !,"End of A people data",!!
    SET rset2 = ##class(%SQL.Statement).%ExecDirect(,mycall,"B")
    IF rset2.%SQLCODE'=0 {WRITE !,"2nd ExecDirect SQLCODE=",rset2.%SQLCODE,!,"rset2.%Message QUIT}
    DO rset2.%Display()
    WRITE !,"End of B people data"
}

```

```
DHC-APP>d ##class(PHA.TEST.SQL).ExecDirect3()
```

Output Values:

```
0. 1
```

```
Dumping result #1
Name    DOB    Spouse
...
```

```
1 Rows(s) Affected
End of B people data
```

`%ExecDirect()`可以调用`%SQL.Statement%Display()`实例方法或`%GetImplementationDetails()`实例方法以返回当前准备好的语句的详细信息。因为`%ExecDirect()`可以准备并执行指定的查询或现有的类查询，所以可以使用`%GetImplementationDetails()` `pStatementType`参数来确定准备哪种查询：

```
/// d ##class(PHA.TEST.SQL).ExecDirect4()
ClassMethod ExecDirect4()
{
    SET mycall = "?=CALL Sample.PersonSets('A',?)"
    SET rset = ##class(%SQL.Statement).%ExecDirect(tStatement,mycall,,"NH")
    IF rset.%SQLCODE'=0 {
        WRITE !,"ExecDirect SQLCODE=",rset.%SQLCODE,! ,rset.%Message QUIT
    }
    SET bool = tStatement.%GetImplementationDetails(.pclassname,.ptext,.pargs,.pStatementType)
    IF bool=1 {
        IF pStatementType=1 {WRITE "Type= specified query",!}
    } ELSEIF pStatementType=45 {
        WRITE "Type= existing class query",!
    }
    WRITE "Implementation class= ",pclassname,!
    WRITE "Statement text= ",ptext,!
    WRITE "Arguments= ",$LISTTOSTRING(pargs),!! }
    ELSE {WRITE "%GetImplementationDetails() failed"}
    DO rset.%Display()
    WRITE !,"End of data"
}
```

[#SQL #Caché #iKnow #InterSystems IRIS for Health](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%B8%89%E7%AB%A0-%E4%BD%BF%E7%94%A8%E5%8A%A8%E6%80%81sql%EF%BC%88%E4%B8%89%E7%BC%89>