文章
姚 鑫 · 九月 7, 2021 阅读大约需 4 分钟

第九章 SQL命令 CREATE METHOD（二）

第九章 SQL命令 CREATE METHOD（二）

characteristics

可用的关键字如下:

- FOR className - 指定要在其中创建方法的类的名称。
  如果这个类不存在，它将被创建。
  还可以通过限定方法名来指定类名。
  FOR子句中指定的类名通过限定方法名重写指定的类名。
- FINAL - 指定子类不能重写该方法。
  默认情况下，方法不是final。
  FINAL关键字由子类继承。
- PRIVATE - 指定该方法只能由它自己的类或子类的其他方法调用。
  默认情况下，方法是公共的，可以不受限制地调用。
  这个限制由子类继承。
- PROCEDURE - 指定该方法是一个SQL存储过程。
  存储过程由子类继承。
  (这个关键字可以缩写为PROC。)
- RESULT SETS ,DYNAMIC RESULT SETS [n] - 指定创建的方法将包含ReturnResultsets关键字。
  这一特征短语的所有形式都是同义词。
- RETURNS datatype - 指定调用该方法返回的值的数据类型。
  如果省略RETURNS，则该方法不能返回值。
  这个规范由子类继承，并且可以由子类修改。
  该数据类型可以指定类型参数，如MINVAL、MAXVAL和SCALE。
  例如RETURNS DECIMAL(19,4)。
  注意，当返回一个值时， IRIS会忽略数据类型的长度;
  例如，RETURNS VARCHAR(32)可以接收由调用方法返回的任意长度的字符串。
- SELECTMODE mode - 仅当LANGUAGE为SQL(默认)时使用。
  当指定时， IRIS将#SQLCOMPILE SELECT=mode语句添加到相应的类方法中，从而生成使用指定的SELECTMODE在方法中定义的SQL语句。
  可能的模式值是LOGICAL、ODBC、RUNTIME和DISPLAY。
  默认为LOGICAL。

如果指定对方法无效的查询关键字(如CONTAINSID或RESULTS)，系统将生成SQLCODE -47错误。
如果指定了重复的查询关键字(例如FINAL FINAL)，系统将生成SQLCODE -44错误。

SELECTMODE子句用于SELECT查询操作以及INSERT和UPDATE操作。
它指定编译时选择模式。
为SELECTMODE指定的值添加在ObjectScript类方法代码的开头，如:#SQLCompile Select=mode。

- 在SELECT查询中，SELECTMODE指定返回数据的模式。
  如果模式值为LOGICAL，则返回逻辑(内部存储)值。
  例如，日期以$HOROLOG格式返回。
  如果模式值为ODBC，则应用逻辑到ODBC的转换，并返回ODBC格式值。
  如果模式值为DISPLAY，则应用逻辑到显示的转换，并返回显示格式值。

如果mode值为RUNTIME，则可以在执行时设置显示模式(LOGICAL、ODBC或display)。
- 在INSERT或UPDATE操作中，SELECTMODE RUNTIME选项支持将输入数据值从显示格式(display或ODBC)自动转换为逻辑存储格式。只有当SQL代码执行时的选择模式设置为LOGICAL(这是所有SQL执行接口的默认设置)时，才会应用这个已编译的从显示到逻辑的数据转换代码。

执行SQL代码时，%SQL.Statement类%SelectMode属性指定执行时选择模式。

LANGUAGE

指定CODEBODY使用的语言的关键字子句。允许的子句是Language OBJECTSCRIPT(对于ObjectScript)或Language SQL。如果省略LANGUAGE子句，则默认为SQL。

codebody

要创建的方法的程序代码。可以在SQL或ObjectScript中指定此代码。使用的语言必须与LANGUAGE子句匹配。但是，在ObjectScript中指定的代码可以包含嵌入式SQL。

IRIS使用提供的代码来生成该方法的实际代码。

如果指定的代码是SQL， IRIS会在生成将SQL嵌入到ObjectScript “ 包装器wrapper ” 中的方法时提供额外的代码行，提供过程上下文处理程序(如有必要)，并处理返回值。以下是此IRIS生成的包装代码的示例：

```
NEW SQLCODE,%ROWID,%ROWCOUNT,title
&sql( SELECT col FROM tbl )
QUIT $GET(title)
```

如果指定的代码是OBJECTSCRIPT，则必须用大括号将ObjectScript代码括起来。除标签和宏预处理器指令外，所有代码行都必须从第1列缩进。标签或宏指令必须在第1列中以冒号(：)开头。

对于ObjectScript代码，必须显式定义 “ 包装器 ” (该NEWs变量并使用QUIT退出，并(可选地)在完成时返回一个值)。

通过指定PROCEDURE关键字，可以将该方法公开为存储过程。调用存储过程时，%Library.SQLProcContext类的对象在%sqlcontext变量中实例化。此过程上下文处理程序用于在过程及其调用方(例如，ODBC服务器)之间来回传递过程上下文。

%sqlcontext由几个属性组成，包括错误对象、SQLCODE错误状态、SQL行数和错误消息。下面的示例显示了用于设置其中几个值的值：

```
SET %sqlcontext.%SQLCODE=SQLCODE
SET %sqlcontext.%ROWCOUNT=%ROWCOUNT
SET %sqlcontext.%Message=%msg
```

SQLCODE和%ROWCOUNT的值由SQL语句的执行自动设置。每次执行前都会重置%sqlcontext对象。

或者，可以通过实例化%SYSTEM.Error对象并将其设置为%sqlcontext.Error来建立错误上下文。

示例

下面的示例使用带有SQL代码的Create方法在Sample.Employee类中生成UpdateSalary方法：

```
CREATE METHOD UpdateSalary ( IN SSN VARCHAR(11), IN Salary INTEGER )
   FOR Sample.Employee
   BEGIN
     UPDATE Sample.Employee SET Salary = :Salary WHERE SSN = :SSN;
   END
```

注意给表添加关键字[DdlAllowed]

下面的示例创建存储为生成随机大写字母的过程的RandomLetter()方法。然后，可以在SELECT语句中将此方法作为函数调用。提供了一个Drop方法来删除RandomLetter()方法。

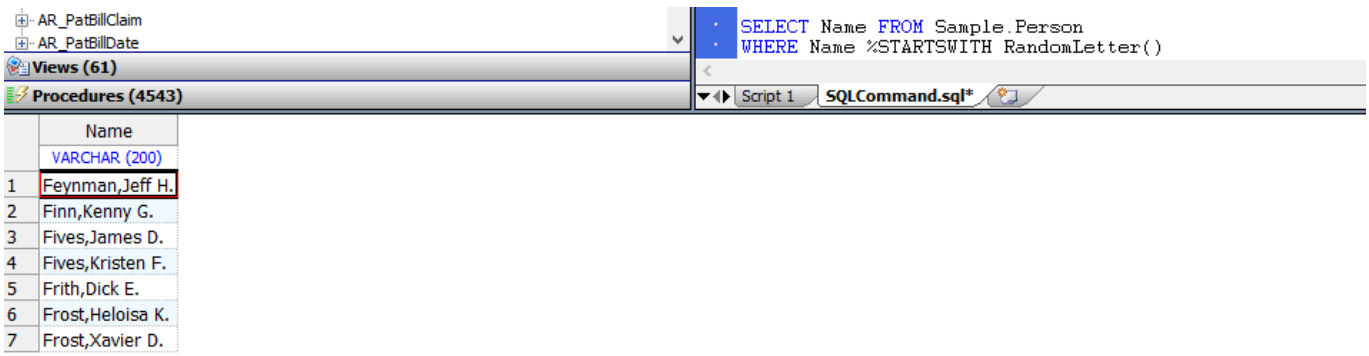
```
CREATE METHOD RandomLetter()
RETURNS INTEGER
PROCEDURE
LANGUAGE OBJECTSCRIPT
{
:Top
 SET x=$RANDOM(91)
 IF x<65 {GOTO Top}
 ELSE {QUIT $CHAR(x)}
}
```

```
Class User.methRandomLetter Extends %Library.RegisteredObject [ ClassType = "", DdlAl
lowed, Owner = {yx}, Not ProcedureBlock ]
{

ClassMethod RandomLetter() As %Library.Integer(MAXVAL=2147483647,MINVAL=-2147483648)
[ SqlName = RandomLetter, SqlProc ]
{
Top
         SET x=$RANDOM(91)
         IF x<65 {GOTO Top}
         ELSE {QUIT $CHAR(x)}
}

}
```

```
SELECT Name FROM Sample.Person
WHERE Name %STARTSWITH RandomLetter()
```




```
DROP METHOD RandomLetter
```

下面的嵌入式SQL示例使用带有ObjectScript代码的Create方法在SQLUser.MyStudents类中生成方法TraineeTitle，并返回一个Title值：

```
ClassMethod CreateMethod()
{
    &sql(
        CREATE METHOD TraineeTitle
        (
            IN SSN VARCHAR(11),
            INOUT Title VARCHAR(50)
        )
        RETURNS VARCHAR(30)
        FOR SQLUser.MyStudents
        LANGUAGE OBJECTSCRIPT
        {
            n SQLCODE,%ROWCOUNT
            &sql(
                SELECT Title INTO :Title FROM Sample.Employee
                WHERE SSN = :SSN
            )
            if $g(%sqlcontext)'= "" {
                s %sqlcontext.%SQLCODE=SQLCODE
                s %sqlcontext.%ROWCOUNT=%ROWCOUNT
            }
            q
        }
    )
    if SQLCODE=0 {
        w !,"????" QUIT
    } elseif SQLCODE=-361 {
        w !,"?????SQLCODE: ",SQLCODE
        &sql(
            DROP METHOD TraineeTitle FROM SQLUser.MyStudents
        )
        if SQLCODE=0 {
            w !,"????" QUIT
        }
    } else {
        w !,"SQL error: ",SQLCODE
    }
}
```

它使用%sqlcontext对象，并使用相应的SQL变量设置它的%SQLCODE和%ROWCOUNT属性。
请注意，在方法的LANGUAGE ObjectScript关键字后面，用花括号括住ObjectScript代码。
在ObjectScript代码中有嵌入式SQL代码，用&sql标记，用括号括起来。

#SQL #Caché

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B9%9D%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-method%EF%BC%88%E4%BA%8C%EF%BC%89