文章
姚 鑫 · 九月 6, 2021 阅读大约需 分钟

第八章 SQL命令 CREATE METHOD(一)

第八章 SQL命令 CREATE METHOD(一)

在类中创建方法。

大纲

```
CREATE [STATIC] METHOD name (parameter_list)
   [ characteristics ]
   [ LANGUAGE SQL ]
   BEGIN
code_body ;
   END

CREATE [STATIC] METHOD name (parameter_list)
    [ characteristics ]
    LANGUAGE OBJECTSCRIPT
   { code_body }
```

参数

- name - 要在存储过程类中创建的方法的名称。
  名称必须是有效的标识符。
  过程名可以是限定的(schema.procname), 也可以是非限定的(procname)。
  非限定过程名被默认模式名。
  名称后面必须跟括号, 即使没有指定参数。
- parameter_list - 可选——传递给方法的参数列表。
  参数列表用圆括号括起来, 列表中的参数用逗号分隔。
  即使没有指定参数, 括号也是必须的。
- characteristics - 可选—指定方法特征的一个或多个关键字。
  允许的关键字是RETURNS, FOR, FINAL, PRIVATE, PROCEDURE, SELECTMODE。可以指定特征关键字短语RESULT SETS、DYNAMIC RESULT SETS或DYNAMIC RESULT SETS n, 其中n是整数。
  这些短语是同义词;
  DYNAMIC关键字和n整数为no-ops, 提供兼容性。多个特征由空格(一个空格或换行符)分隔。
  特性以任何顺序指定。
- LANGUAGE OBJECTSCRIPT , LANGUAGE SQL - 可选—用于代码体的编程语言。指定语言对象脚本(对于对象脚本)或语言SQL。如果省略了LANGUAGE子句, 则默认为SQL。
- code_body - 方法的程序代码。SQL程序代码以BEGIN关键字开头, 以END关键字结尾。code_body中的每个完整的SQL语句都以分号(;).ObjectScript程序代码用花括号括起来。
  ObjectScript代码行必须缩进。

描述

CREATE METHOD语句创建一个类方法。
这个类方法可能是存储过程，也可能不是。
要在公开为SQL存储过程的类中创建方法，必须指定procedure关键字。
默认情况下，CREATE METHOD不会创建一个同时也是存储程序的方法；
CREATE PROCEDURE语句总是创建一个同时也是存储过程的方法。

提供可选的STATIC关键字是为了说明所创建的方法是一个静态(类)方法，而不是一个实例方法。
该关键字没有提供实际的功能。

为了创建方法，必须具有GRANT命令指定的%CREATE_METHOD管理权限。
如果试图为具有已定义所有者的现有类创建方法，则必须作为该类的所有者登录。
否则，操作将失败，并出现SQLCODE -99错误。

如果类定义是已部署的类，则不能在类中创建方法。
此操作失败，并出现一个带有%msg的SQLCODE -400错误Unable to execute DDL that modifies a deployed class: 'classname'.

下面两个示例都展示了相同类方法的创建。
第一个示例使用CREATE METHOD，第二个示例在类User中定义类方法。
字母:

```
CREATE METHOD RandCaseLetter(IN caps CHAR)
  RETURNS INTEGER
  PROCEDURE
LANGUAGE OBJECTSCRIPT
{
:Top
    if caps = "U" {
        s x = $random(91)
        if x > 64 {
            q $char(x)
        } else {
            g Top
        }
    } elseif caps="L" {
        s x = $random(123)
        if x > 97 {
            q $char(x)
        } else {
            g Top
        }
    } else {
        q "????? 'U' ? 'L'"
    }
}
}
```

自动创建的后台类

```
Class User.methRandCaseLetter Extends %Library.RegisteredObject [ ClassType = "", Ddl
Allowed, Owner = {yx}, Not ProcedureBlock ]
{

ClassMethod RandCaseLetter(caps As %Library.String(MAXLEN=1)) As %Library.Integer(MAX
VAL=2147483647,MINVAL=-2147483648) [ SqlName = RandCaseLetter, SqlProc ]
{
```

```
Top
         IF caps="U" {SET x=$RANDOM(91) IF x>64 {QUIT $CHAR(x)}
           ELSE {GOTO Top}}
         ELSEIF caps="L" {SET x=$RANDOM(123) IF x>97 {QUIT $CHAR(x)}
           ELSE {GOTO Top}}
         ELSE {QUIT "case must be 'U' or 'L'"}
}

}



Class User.Letters Extends %Persistent [ DdlAllowed ]
{
ClassMethod RandCaseLetter(caps) As %String [ SqlName = RandomLetter, SqlProc ]
{
Top
    if caps = "U" {
        s x = $random(91)
        if x > 64 {
            q $char(x)
        } else {
            g Top
        }
    } elseif caps="L" {
        s x = $random(123)
        if x > 97 {
            q $char(x)
        } else {
            g Top
        }
    } else {
        q "????? 'U' ? 'L'"
    }
}
}
```

参数

name

要创建的方法的名称。
此名称可以是非限定的(StoreName)并被系统范围的默认模式名称，也可以通过指定模式名称(Patient.StoreName)
进行限定。
可以使用$SYSTEM.SQL.Schema.Default()方法确定当前系统范围的默认模式名。
系统范围的初始默认模式名是SQLUser，它对应于类包名User。

注 意，FOR特征(将在下面描述)覆盖了name中指定的类名。
如果已经存在具有此名称的方法，则操作将失败，并出现SQLCODE -361错误。

生成类的名称是与模式名对应的包名，后面跟着一个点，然后是“meth”，最后是指定的名称。
例如，如果非限定方法名RandomLetter被初始默认模式SQLUser，则产生的类名将是:User.methRandomLetter。

parameter-list

用于将值传递给方法的参数列表。

形参列表用圆括号括起来，列表中的形参声明用逗号分隔。
即使没有指定参数，括号也是必须的。
列表中的每个参数声明由(按顺序)组成

- 一个可选关键字，指定参数模式是IN(输入值)、OUT(输出值)还是INOUT(传值)。
  如果省略，默认参数模式为IN。
- 参数名称。
  参数名称区分大小写。
- 参数的数据类型。
- 可选:默认值。
  可以指定DEFAULT关键字后跟一个默认值;
  DEFAULT关键字是可选的。
  如果没有指定默认值，则假定默认值为NULL。

方法的输出值自动从Logical格式转换为Display/ODBC格式。

默认情况下，方法的输入值不会从Display/ODBC格式转换为Logical格式。
但是，可以使用$SYSTEM.SQL.Util.SetOption("SQLFunctionArgConversion")方法在系统范围内配置输入显示到逻辑的转换。
可以使用$SYSTEM.SQL.Util.GetOption("SQLFunctionArgConversion")来确定该选项的当前配置。

下面的示例指定两个输入参数，它们都有默认值。
为第一个参数指定可选的DEFAULT关键字，为第二个参数省略:

```
CREATE METHOD RandomLetter(IN firstlet CHAR DEFAULT 'A',IN lastlet CHAR 'Z')
BEGIN
-- SQL program code
END
```

#SQL #Caché

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AB%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-method%EF%BC%88%E4%B8%80%EF%BC%89