文章
姚 鑫 · 九月 8, 2021 阅读大约需 7 分钟

第十章 SQL命令 CREATE PROCEDURE（一）

第十章 SQL命令 CREATE PROCEDURE（一）

创建作为SQL存储过程公开的方法或查询。

大纲

```
CREATE PROCEDURE procname(parameter_list)
   [ characteristics ]
   [ LANGUAGE SQL ]
   BEGIN
code_body ;
   END

CREATE PROCEDURE procname(parameter_list)
    [ characteristics ]
    LANGUAGE OBJECTSCRIPT
   { code_body }

CREATE PROCEDURE procname(parameter_list)
    [ characteristics ]
    LANGUAGE { JAVA | PYTHON | DOTNET } EXTERNAL NAME external-stored-procedure

CREATE PROC procname(parameter_list)
   [ characteristics ]
   [ LANGUAGE SQL ]
   BEGIN
code_body ;
   END

CREATE PROC procname(parameter_list)
    [ characteristics ]
    LANGUAGE OBJECTSCRIPT
   { code_body }

CREATE PROC procname(parameter_list)
    [ characteristics ]
    LANGUAGE { JAVA | PYTHON | DOTNET } EXTERNAL NAME external-stored-procedure
```

参数

- procname - 要在存储过程类中创建的过程的名称。
  procname必须是一个有效的标识符。
  过程名可以是限定的(schema.procname)，也可以是非限定的(procname)。
  非限定过程名接受默认模式名。
  procname后面必须跟圆括号，即使没有指定参数。

- parameterlist - 可选——传递给过程的一个包含零个或多个参数的列表。
  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。
  每个参数由(按顺序)组成:一个可选的 IN、OUT或INOUT关键字;
  变量名;
  数据类型;
  和一个可选的DEFAULT子句。
- characteristics - 可选-一个或多个关键字，指定过程的特征。
  在创建方法时，允许的关键字是FINAL、FOR、PRIVATE、RETURNS、SELECTMODE。
  在创建查询时，允许的关键字是CONTAINID、FINAL、FOR、RESULTS、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。
  特性可以以任何顺序指定。
- LANGUAGE OBJECTSCRIPT，LANGUAGE SQL - 可选-
  一个关键字子句，指定用于codebody的编程语言。
  指定语言OBJECTSCRIPT(用于OBJECTSCRIPT)或语言SQL。
  如果省略LANGUAGE子句，则默认为SQL。
- LANGUAGE JAVA，LANGUAGE PYTHON，LANGUAGE DOTNET - 可选-
  关键字子句，指定用于调用指定语言中的现有外部存储过程的编程语言。
  被调用的例程必须是一个静态方法。
- codebody - 该程序的程序代码。SQL程序代码以BEGIN关键字开头，以END关键字结尾。
  codebody中的每个完整SQL语句都以分号(;)结束。ObjectScript程序代码用花括号括起来。
  ObjectScript代码行必须缩进。

描述

CREATE PROCEDURE语句创建一个方法或查询，该方法或查询将自动作为SQL存储过程公开。
存储过程可以由当前名称空间中的所有进程调用。
存储过程由子类继承。
- 如果使用SQL语言，则codebody必须包含SELECT语句，以便生成公开为存储过程的查询。
如果代码不包含SELECT语句，则CREATE PROCEDURE创建一个方法。
- 如果LANGUAGE
OBJECTSCRIPT，则codebody必须调用Execute()和Fetch()方法，以生成公开为存储过程的查询。
它也可以调用Close()， FetchRows()和GetInfo()方法。
如果代码没有调用Execute()和Fetch()，则CREATE PROCEDURE创建一个方法。

默认情况下，CREATE PROCEDURE创建公开为存储过程的方法。

要创建未公开为存储过程的方法，请使用 CREATE METHOD或CREATE FUNCTION语句。
若要创建未公开为存储过程的查询，请使用CREATE QUERY语句。
通过指定procedure特征关键字，这些语句还可以用于创建作为存储过程公开的方法或查询。

为了创建一个过程，必须具有GRANT命令指定的%CREATEPROCEDURE管理权限。
如果为具有已定义所有者的现有类创建过程，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，则不能在类中创建过程。
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'.

使用CALL语句执行存储过程。

参数

procname

要创建为存储过程的方法或查询的名称。即使没有指定参数，procname后面也必须有括号。过程名可以采用以下任何形式:
- Unqualified不限定的:接受默认模式名。例如,MedianAgeProc()。
- Qualified:提供模式名称。例如,Patient.MedianAgeProc()。
- Multilevel多级:限定为一个或多个模式层，以并行相应的类包成员。
在这种情况下，procname可能只包含一个句点字符;
对应类方法名中的其他句点将被下划线字符替换。
在最低级别的类包成员之前指定句点。
例如，%SYSTEM.SQL_GetROWID()或%SYS_PTools.StatsSQL_Export()。

非限定的procname接受默认模式名。
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)覆盖了在procname中指定的类名。
如果已经存在具有此名称的过程，则操作将失败，并出现SQLCODE -361错误。

SQL使用SQL procname生成相应的类名。
该名称由与模式名对应的包名、点、" proc "和指定的过程名组成。
例如，如果非限定过程名RandomLetter()接受默认模式SQLUser，则产生的类名将是:User.procRandomLetter()。

SQL不允许指定只以字母大小写不同的procname。
指定一个只在字母大小写上与现有过程名不同的procname将导致SQLCODE -400错误。

如果指定的procname已经存在于当前命名空间中，系统将生成SQLCODE -361错误。
要确定指定的procname是否已经存在于当前命名空间中，请使用$SYSTEM.SQL.Schema.ProcedureExists()方法。

注意: SQL过程名称和 TSQL过程名称共享同一组名称。
因此，不能在同一命名空间中创建与TSQL过程同名的SQL过程。
尝试这样做会导致SQLCODE -400错误。

parameterlist

用于将值传递给方法或查询的参数列表。
形参列表用圆括号括起来，列表中的形参声明用逗号分隔。
括号是必须的，即使没有指定参数。

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(修改值)。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。

下面的示例创建了一个具有两个输入参数的存储过程，这两个参数都具有默认值。
一个输入参数指定可选的DEFAULT关键字，另一个输入参数忽略该关键字:

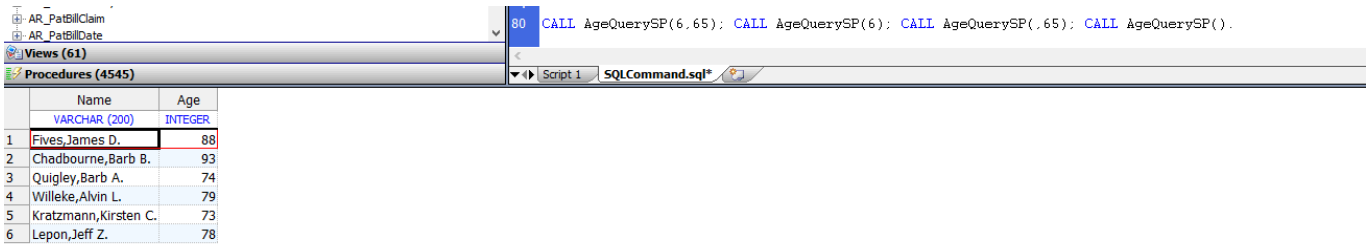
```
CREATE PROCEDURE AgeQuerySP(IN topnum INT DEFAULT 10,IN minage INT 20)
   BEGIN
   SELECT TOP :topnum Name,Age FROM Sample.Person
   WHERE Age > :minage ;
   END
```

下面的示例在功能上与上面的示例相同。可选的DEFAULT关键字省略:

```
CREATE PROCEDURE AgeQuerySP(IN topnum INT 10,IN minage INT 20)
   BEGIN
   SELECT TOP :topnum Name,Age FROM Sample.Person
   WHERE Age > :minage ;
   END
```

下面是这个过程中所有有效的CALL语句:

```
CALL AgeQuerySP(6,65); CALL AgeQuerySP(6); CALL AgeQuerySP(,65); CALL AgeQuerySP().
```



	Name	Age
	VARCHAR (200)	INTEGER
1	Fives,James D.	88
2	Chadbourne,Barb B.	93
3	Quigley,Barb A.	74
4	Willeke,Alvin L.	79
5	Kratzmann,Kirsten C.	73
6	Lepon,Jeff Z.	78

下面的示例创建了一个公开为具有三个参数的存储过程的方法:

```
CREATE PROCEDURE UpdatePaySP
  (IN Salary INTEGER DEFAULT 0,
   IN Name VARCHAR(50),
   INOUT PayBracket VARCHAR(50) DEFAULT 'NULL')
BEGIN
   UPDATE Sample.Employee SET Salary = :Salary
   WHERE Name=:Name ;
END
```

存储过程不执行参数的自动格式转换。
例如，ODBC格式或Display格式的输入参数仍然保持该格式。
调用过程的代码和过程代码本身负责以适合应用程序的格式处理IN/OUT值，并执行任何必要的转换。

因为方法或查询是作为存储过程公开的，所以它使用过程上下文处理程序在过程及其调用方之间来回传递过程上下文。调用存储过程时，%Library.SQLProcContext类的对象在%sqlcontext变量中实例化。这用于在过程及其调用者(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Error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参数的值:

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

SQLCODE和%ROWCOUNT的值是在执行SQL语句时自动设置的。
%sqlcontext对象在每次执行之前都被重置。

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

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-procedure%EF%BC%88%E4%B8%80%EF%BC%89