

文章

[姚鑫](#) · 九月 26, 2021 阅读大约需 4 分钟

第二十六章 SQL命令 DECLARE

第二十六章 SQL命令 DECLARE

声明游标

大纲

DECLARE cursor-name CURSOR FOR query

参数

- cursor-name - 游标的名称，必须以字母开头，并且仅包含字母和数字。(游标名称不遵循SQL标识符约定)。游标名称区分大小写。它们受其他命名限制的约束，如下所述。
- query - 定义游标结果集的标准SELECT语句。此选择可以包括%NOFPLAN关键字，以指定应忽略此查询的冻结计划(如果有)。此SELECT可以包括ORDER BY子句，可以带有或不带有TOP子句。此SELECT可以在FROM子句中指定表值函数。

描述

DECLARE语句声明在基于游标的嵌入式SQL中使用的游标。声明游标后，可以发出OPEN语句来打开游标，然后发出一系列FETCH语句来检索各个记录。游标定义SELECT查询，该查询用于选择要由这些FETCH语句检索的记录。可以发出一条CLOSE语句来关闭(但不是删除)游标。

作为SQL语句，仅嵌入式SQL支持DECLARE。对于动态SQL，可以使用简单的SELECT语句(不带INTO子句)，也可以使用动态SQL和嵌入式SQL的组合。使用ODBC API通过ODBC支持等效操作。

DECLARE声明只进(不可滚动)游标。提取操作从查询结果集中的第一条记录开始，并按顺序遍历结果集记录。一次提取只能提取一次记录。下一次提取将获取结果集中的下一条连续记录。

因为DECLARE是一个声明，而不是执行的语句，所以它不设置或终止SQLCODE变量。

游标名称

游标名称区分大小写。

游标名称在例程和相应类中必须是唯一的。游标名称可以是任意长度，但在前29个字符内必须是唯一的。游标名称区分大小写。如果已声明指定的游标，编译将失败，并显示SQLCODE-52错误，游标名称已声明。

游标名称不是特定于命名空间的。可以在一个命名空间中声明游标，并在另一个命名空间中打开、获取或关闭此游标。在执行OPEN命令时编译嵌入式SQL。SQL表和局部变量是特定于名称空间的，因此必须在查询中指定的表所在的同一名称空间中调用OPEN操作(或者能够访问名称空间中的表)。

游标名称的第一个字符必须是字母。游标名称的第二个和后续字符必须是字母或数字。与SQL标识符不同，游标名称中不允许使用标点符号。

可以使用分隔符字符(双引号)将SQL保留字指定为游标名称。分隔游标名称不是SQL分隔标识符；分隔游标名称仍然区分大小写，不能包含标点符号。在大多数情况下，SQL保留字不应用作游标名称。

通过游标更新

可以使用带有WHERE CURRENT OF子句的UPDATE或DELETE语句，通过声明的游标执行记录更新和删除。SQL中，如果对受影响的表和列具有适当的权限，则游标始终可以用于更新或删除操作。

DECLARE语句可以在查询后指定FOR UPDATE或FOR READ ONLY关键字子句。这些子句是可选的，不执行任何操作。它们是作为在代码中记录发出查询的进程是否具有所需的更新和删除对象权限的一种方式提供的。

示例

下面的嵌入式SQL示例使用DECLARE为指定两个输出主机变量的查询定义游标。然后，光标被打开、重复读取和关闭：

```
ClassMethod Declare()
{
    s name = "John Doe", state = "##"
    &sql(
        DECLARE EmpCursor CURSOR FOR
            SELECT Name, Home_State
            INTO :name,:state FROM Sample.Person
            WHERE Home_State %STARTSWITH 'A'
        FOR READ ONLY
    )
    w !,"BEFORE: Name = ",name," State = ",state
    &sql(
        OPEN EmpCursor
    )
    if SQLCODE < 0 {
        w "SQL?????:",SQLCODE," ",%msg
        q
    }
    n %ROWCOUNT,%ROWID
    for {
        &sql(
            FETCH EmpCursor
        )
        q:SQLCODE
        w !,"DURING: Name = ",name," State = ",state
    }
    w !,"????SQLCODE = ",SQLCODE
    w !,"????? = ",%ROWCOUNT
    &sql(
        CLOSE EmpCursor
    )
    if SQLCODE < 0 {
        w "SQL?????:",SQLCODE," ",%msg
        q
    }
    w !,"AFTER: Name = ",name," State = ",state
}
```

下面的嵌入式SQL示例使用DECLARE为查询定义游标，该查询在INTO子句中指定OUTPUT主机变量，在WHERE子句中指定INPUT主机变量。然后，光标被打开、重复读取和关闭：

```
ClassMethod Declare1()
{
    n SQLCODE,%ROWCOUNT,%ROWID
    s EmpZipLow = "10000"
    s EmpZipHigh = "19999"
    &sql(
        DECLARE EmpCursor1 CURSOR FOR
            SELECT Name,Home_Zip
            INTO :name,:zip
            FROM Sample.Employee
            WHERE Home_Zip BETWEEN :EmpZipLow AND :EmpZipHigh)
    &sql(
        OPEN EmpCursor1
    )
    if SQLCODE < 0 {
        w "SQL??????:",SQLCODE," ",%msg
        q
    }
    for {
        &sql(
            FETCH EmpCursor1
        )
        q:SQLCODE
        w !,name," ",zip
    }
    &sql(
        CLOSE EmpCursor1
    )
    if SQLCODE < 0 {
        w "SQL??????:",SQLCODE," ",%msg
        q
    }
}
```

下面的嵌入式SQL示例使用表值函数作为查询的FROM子句：

```
ClassMethod Declare2()
{
    s $NAMESPACE="Samples"
    &sql(DECLARE EmpCursor2 CURSOR FOR
        SELECT Name INTO :name FROM Sample.SP_Sample_By_Name('A')
        FOR READ ONLY
    )
    &sql(
        OPEN EmpCursor2
    )
    if SQLCODE < 0 {
        w "SQL??????:",SQLCODE," ",%msg
        q
    }
    n %ROWCOUNT,%ROWID
    for {
        &sql(
```

```
        FETCH EmpCursor2
    )
    q:SQLCODE
    w "Name=", name, !
}
w !, "????SQLCODE = ", SQLCODE
w !, "????? = ", %ROWCOUNT
&sql(
    CLOSE EmpCursor2
)
if SQLCODE < 0 {
    w "SQL??????:", SQLCODE, " ", %msg
    q
}
}
```

[#SQL](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E5%8D%81%E5%85%AD%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-declare>