

文章

姚鑫 · 三月 20, 2021 阅读大约需分钟

## 第十二章 使用嵌入式SQL(四)

## 第十二章 使用嵌入式SQL(四)

### SQL游标

游标是指向数据的指针，该数据允许嵌入式SQL程序对所指向的记录执行操作。通过使用游标，Embedded SQL可以遍历结果集。嵌入式SQL可以使用游标执行查询，该查询从多个记录返回数据。嵌入式SQL还可以使用游标更新或删除多个记录。

必须对SQL游标进行DECLARE，并为其命名。在DECLARE语句中，提供了SELECT语句，该语句标识游标将指向的记录。然后，将此游标名称提供给OPEN游标语句。然后，反复发出FETCH游标语句以遍历SELECT结果集。然后，发出CLOSE游标语句。

- 基于游标的查询使用DECLARE游标名称CURSOR FOR SELECT来选择记录，并(可选)将select列值返回到输出主机变量中。FETCH语句遍历结果集，使用这些变量返回选定的列值。

- 基于游标的DELETE或UPDATE使用DECLARE游标名CURSOR FOR SELECT选择操作的记录。没有指定输出主机变量。FETCH语句遍历结果集。DELETE或UPDATE语句包含WHERE CURRENT OF子句，以标识当前光标位置，以便对所选记录执行操作。

请注意，游标不能跨越方法。因此，必须在同一类方法中声明，打开，获取和关闭游标。在生成和方法的所有代码(例如从.CSP文件生成)中考虑这一点很重要。

下面的示例使用游标执行查询并将结果显示给主设备：

```

/// d ##class(PHA.TEST.SQL).CURSOR()
ClassMethod CURSOR()
{
    &sql(DECLARE C5 CURSOR FOR
        SELECT %ID,Name
        INTO :id, :name
        FROM Sample.Person
        WHERE Name %STARTSWITH 'A'
        ORDER BY Name
    )

    &sql(OPEN C5)
    QUIT:(SQLCODE'=0)
    &sql(FETCH C5)

    While (SQLCODE = 0) {
        Write id, ": ", name,!
        &sql(FETCH C5)
    }

    &sql(CLOSE C5)
}

```

```
DHC-APP>d ##class(PHA.TEST.SQL).CURSOR()  
95:  Adams,Diane F.  
183: Adams,Susan E.  
71:  Ahmed,Elmo X.  
28:  Alton,Martin S.  
175: Alton,Phil T.  
86:  Anderson,Mario L.  
131: Anderson,Valery N.
```

此示例执行步骤:

1. 声明一个游标C1,该游标返回一组按Name排序的Person行。
2. 打开游标。
3. 游标上调用FETCH直到到达数据末尾。每次调用FETCH之后,如果有更多数据要提取,则SQLCODE变量将设置为0。每次调用FETCH后,返回的值都将复制到DECLARE语句的INTO子句指定的主机变量中。
4. 关闭光标。

## DECLARE 游标声明

DECLARE语句同时指定了游标名称和定义游标的SQL SELECT语句。  
DECLARE语句必须在例程中出现在使用游标的任何语句之前。

游标名称区分大小写。

**游标名称在类或例程中必须唯一。因此,递归调用的例程不能包含游标声明。在这种情况下,最好使用动态SQL。**

下面的示例声明一个名为MyCursor的游标:

```
&sql(DECLARE MyCursor CURSOR FOR  
    SELECT Name, DOB  
    FROM Sample.Person  
    WHERE Home_State = :state  
    ORDER BY Name  
)
```

DECLARE语句可以包括一个可选的INTO子句,该子句指定在遍历游标时将接收数据的本地主机变量的名称。例如,我们可以在前面的示例中添加一个INTO子句:

```
&sql(DECLARE MyCursor CURSOR FOR  
    SELECT Name, DOB  
    INTO :name, :dob  
    FROM Sample.Person  
    WHERE Home_State = :state  
    ORDER BY Name  
)
```

INTO子句可以包含逗号分隔的主机变量列表,单个主机变量数组或两者的组合。如果指定为以逗号分隔的列表,则INTO子句宿主变量的数量必须与游标的SELECT列表中的列数完全匹配,否则在编译该语句时会收到“**量不匹配**”错误。

如果DECLARE语句不包含INTO子句,则INTO子句必须出现在FETCH语句中。通过在DECLARE语句而不是FETCH语句中指定INTO子句,可能会导致性能小幅提高。

因为DECLARE是声明,而不是执行的语句,所以它不会设置或终止SQLCODE变量。

如果已经声明了指定的游标,则编译将失败,并显示SQLCODE -52错误,游标名称已声明。

**执行DECLARE语句不会编译SELECT语句。SELECT语句在第一次执行OPEN语句时被编译。嵌入式SQL不在常规编译时进行编译,而是在SQL运行时(运行时)进行编译。**

## OPEN游标声明

OPEN语句为后续执行准备了一个游标:

```
&sql (OPEN MyCursor)
```

执行OPEN语句将编译在DECLARE语句中找到的Embedded SQL代码,创建最优的查询计划,并生成缓存的查询。执行OPEN(在SQL运行时)时,会发出涉及缺少资源(例如未定义的表或字段)的错误。

调用OPEN后,SQLCODE变量将设置为0。

必须先调用OPEN才能从游标中获取数据。

## FETCH游标声明

FETCH语句获取游标一行的数据(由游标查询定义):

```
&sql (FETCH MyCursor)
```

必须先对游标进行DECLARE并打开,然后才能在其上调用FETCH。

FETCH语句可以包含INTO子句,该子句指定在游标游标时将接收数据的本地主机变量的名称。例如,我们可以在前面的示例中添加一个INTO子句:

```
&sql (FETCH MyCursor INTO :a, :b)
```

INTO子句可以包含逗号分隔的主机变量列表,单个主机变量数组或两者的组合。如果指定为以逗号分隔的列表,则INTO子句宿主变量的数量必须与游标的SELECT列表中的列数完全匹配,否则在编译该语句时,将收到SQLCODE -76“不匹配”错误。

通常,INTO子句是在DECLARE语句中指定的,而不是在FETCH语句中指定的。如果DECLARE语句中的SELECT查询和FETCH语句都包含INTO子句,则仅设置由DECLARE语句指定的主机变量。如果仅FETCH语句包含INTO子句,则将设置由FETCH语句指定的主机变量。

如果FETCH检索数据,则将SQLCODE变量设置为0;否则,将SQLCODE变量设置为100(没有更数据)。如果没有数据(或没有更数据)到FETCH,则将SQLCODE设置为100(没有更数据)。主机变量值仅应在SQLCODE = 0时使用。

根据查询,第一次调用FETCH可能会执行其他任务(例如对临时数据结构中的值进行排序)。

## CLOSE游标声明

CLOSE语句终止游标的执行:

```
&sql(CLOSE MyCursor)
```

CLOSE语句清除查询执行所使用的任何临时存储。无法调用CLOSE的程序将遇到资源泄漏(例如,不关闭IRIS TEMP临时数据库增加)。

成功调用CLOSE后,SQLCODE变量将设置为0。因此,在关闭游标之前,应检查最终的FETCH是否将SQLCODE设置为0或100。

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

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%BA%8C%E7%AB%A0-%E4%BD%BF%E7%94%A8%E5%B5%8C%E5%85%A5%E5%BC%8Fsql%E5%BC%88%E5%9B%9B%E5%BC%89>