文章
姚 鑫 · 十月 26, 2021 阅读大约需 8 分钟

第五十七章 SQL命令 INTO

第五十七章 SQL命令 INTO

一个SELECT子句，指定在宿主变量中存储选定的值。

大纲

```
INTO :hostvar1 [,:hostvar2]...
```

参数

- :hostvar1 - 在宿主语言中声明的输出宿主变量。
  当在INTO子句中指定时，变量名前面加冒号(:)。
  主机变量可以是局部变量(非下标或下标)或对象属性。
  可以将多个变量指定为逗号分隔的列表、单个下标数组变量或逗号分隔的列表和单个下标数组变量的组合。

描述

INTO子句和主机变量仅在嵌入式SQL中使用。它们不在动态SQL中使用。在动态SQL中，%SQL.Statement类为输出变量提供了类似的功能。在通过ODBC、JDBC或动态SQL处理的SELECT查询中指定INTO子句会导致SQLCODE-422错误。

INTO子句可以在SELECT、DECLARE或FETCH语句中使用。INTO子句对于所有三个语句都是相同的；本页上的示例都引用SELECT语句。

INTO子句使用在SELECT-ITEM列表中检索(或计算)的值来设置相应的输出主机变量，从而使这些返回的数据值可用于ObjectScript。在SELECT中，可选INTO子句出现在SELECT-ITEM列表之后、FROM子句之前。

注意：编译嵌入式SQL时，输出主机变量将初始化为空字符串。这可以防止在执行时出现<UNDECLARED>错误。因此，只有当SQLCODE=0时，主机变量才包含有意义的值。在使用输出主机变量值之前，请始终检查SQLCODE。当SQLCODE=100或SQLCODE为负数时，不要使用这些变量值。

主机变量

主机变量只能包含单个值。因此，嵌入式SQL中的SELECT只检索一行数据。这默认为表格的第一行。当然，可以通过使用WHERE条件限制符合条件的行来从表的其他行检索数据。

在嵌入式SQL中，可以通过声明游标，然后为每一连续行发出FETCH命令，从多行返回数据。INTO子句主机变量可以在DECLARE查询中指定，也可以在FETCH中指定。
- 主机变量列表，由逗号分隔的主机变量列表组成，每个选择项对应一个主机变量列表。
- 主机变量数组，由单个下标主机变量组成。

注意：如果主机语言声明变量的数据类型，则在调用SELECT语句之前，所有主机变量都必须用主机语言声明。检索到的字段值的数据类型必须与主机变量声明匹配。(ObjectScript不声明变量的数据类型。)

使用主机变量列表

在INTO子句中指定主机变量列表时，以下规则适用：
- INTO子句中的主机变量数必须与SELECT-
ITEM列表中指定的字段数匹配。如果所选字段和主机变量的数量不同，SQL将返回 “ 基数不匹配 ” 错误。
- 选定字段和主机变量按相对位置匹配。因此，这两个列表中对应的项必须以相同的顺序出现。
- 列出的主机变量可以是无下标变量或下标变量的任意组合。
- 列出的主机变量可以返回聚合值(如计数、总和或平均值)或函数值。
- 列出的主机变量可以返回%CLASSNAME和%TABLENAME值。
- 列出的主机变量可以从涉及多个表的SELECT返回字段值，也可以从没有FROM子句的SELECT返回值。

下面的示例从包含四个主机变量的列表中选择四个字段。本例中的主机变量带有下标：

```
ClassMethod Into()
{
    &sql(
        SELECT %ID,Home_City,Name,SSN
        INTO :mydata(1),:mydata(2),:mydata(3),:mydata(4)
        FROM Sample.Person
        WHERE Home_State='MA' )
    if SQLCODE = 0 {
        for i = 1 : 1 : 15 {
            if $d(mydata(i)) {
                w "field ",i," = ",mydata(i),!
            }
        }
    } else {
        w "SQLCODE=",SQLCODE,!
    }
}
```

使用主机变量数组

主机变量数组使用单个下标变量来包含所有选定的字段值。此数组是根据表中字段定义的顺序填充的，而不是根据选择项列表中字段的顺序填充的。

在INTO子句中使用主机变量数组时，适用以下规则：

- 选择项列表中指定的字段被选入单个主机变量的下标。因此，不必将选择项列表中的项数与主机变量COUNT匹配。
- 主机变量下标由表定义中相应的字段位置填充。例如，表定义中定义的第6个字段对应于mydata(6)。与指定选择项不对应的所有下标仍未定义。选择项中项的顺序对如何填充下标没有影响。
- 主机变量数组只能从单个表返回字段值。
- 主机变量数组只能返回字段值。它不能返回聚合值(如COUNT、SUM或Average)、函数值或%CLASSNAME或%TABLENAME值。(可以通过指定将主机变量列表项与主机变量数组相结合的主机变量参数来返回这些参数。)
- 以下示例将四个字段选择到主机变量数组中：

```
ClassMethod Into1()
{
    &sql(
        SELECT %ID,Home_City,Name,SSN
        INTO :mydata()
        FROM Sample.Person
        WHERE Home_State='MA'
    )
```

```
    if SQLCODE = 0 {
        for i = 1 : 1 : 15 {
            if $d(mydata(i)) {
                w "field ",i," = ",mydata(i),!
            }
        }
    } else {
        w "SQLCODE=",SQLCODE,!
    }
}
```

返回字段值的主机变量

下面的嵌入式SQL示例从表的第一条记录中选择三个字段(嵌入式SQL始终检索单个记录)，并使用INTO设置三个相应的无下标主机变量。然后，ObjectScript写入命令使用这些变量。在从嵌入式SQL返回时立即测试SQLCODE变量被认为是很好的编程实践。如果SQLCODE不等于0，则将输出主机变量的值初始化为空字符串。

```
ClassMethod Into2()
{
    &sql(
        SELECT Home_State, Name, Age
        INTO :state, :name, :age
        FROM Sample.Person
    )
    if SQLCODE=0 {
        w !,"  Name=",name
        w !,"  Age=",age
        w !,"  Home State=",state
    } else {
        w !,"SQL error ",SQLCODE
    }
}
```

下面的嵌入式SQL示例返回由两个表联接产生的行中的字段值。从多个表返回字段时，必须使用主机变量列表：

```
ClassMethod Into3()
{
    &sql(
        SELECT P.Name,E.Title,E.Name,P.%TABLENAME,E.%TABLENAME
        INTO :name(1),:title,:name(2),:ptname,:etname
        FROM Sample.Person AS P LEFT JOIN
            Sample.Employee AS E ON E.Name %STARTSWITH 'B'
        WHERE P.Name %STARTSWITH 'A')
    if SQLCODE = 0 {
        w ptname," = ",name(1),!
        w etname," = ",title,!
        w etname," = ",name(2)
    } else {
        w !,"SQL error ",SQLCODE
    }
}
```

返回文字值和聚合值的主机变量

由于输出主机变量仅在SQLCODE=0时有效，因此避免使用发出SQLCODE=100(查询不返回表数据)的查询结果非常重要。SQLCODE=100将所有输出主机变量默认为空字符串，包括返回的文字和计数聚合。

下面的嵌入式SQL示例将一个主机变量(TODAY)传递给SELECT语句，其中的计算结果是INTO子句变量VALUE(：TOWORY)。该主机变量被传递给包含该主机的程序。此查询没有引用表字段，因此没有指定FROM子句。没有FROM子句的嵌入式SQL查询不能发出SQLCODE=100。带有FROM子句的嵌入式SQL查询可以发出SQLCODE=100，这会将所有输出变量定义为缺省的空字符串的值，包括那些不是表字段值的变量，例如：Tomorrow。

```
ClassMethod Into4()
{
    s today = $h
    &sql(
        SELECT :today+1
        INTO :tomorrow
    )
    if SQLCODE=0 {
        w !,"Tomorrow is: ",$ZDATE(tomorrow)
    } else {
        w !,"SQL error ",SQLCODE
    }
}
```

下面的嵌入式SQL示例返回聚合值。它使用COUNT聚合函数对表中的记录进行计数，并使用AVG对工资字段值进行平均。INTO子句将这些值作为两个下标主机变量返回给ObjectScript。

因为两个SELECT-Items都是聚合的，所以即使指定的表不包含数据，该程序也总是发出SQLCODE=0。在本例中，count(*)=0，AVG(Salary)是默认的空字符串。

```
ClassMethod Into5()
{
    &sql(
        SELECT COUNT(*),AVG(Salary)
        INTO :agg(1),:agg(2)
        FROM Sample.Employee)
    if SQLCODE = 0 {
        w !,"Total Employee records= ",agg(1)
        w !,"Average Employee salary= ",agg(2)
    } elseif SQLCODE=100 {
        w !,"Total Employee records= ",agg(1)
    } else {
        w !,"SQL error ",SQLCODE
    }
}
```

下面的嵌入式SQL示例与上一个示例相同，只是它还返回一个字段值。因为SELECT-ITEMS包括一个字段值，所以当指定的表不包含数据时，该程序可以发出SQLCODE=100。在此示例中，如果SQLCODE=100，则COUNT(*)是默认的空字符串，而不是0：

```
ClassMethod Into6()
{
    &sql(
        SELECT COUNT(*),AVG(Salary),Salary
        INTO :agg(1),:agg(2),:pay
        FROM Sample.Employee
```

```
    )
    if SQLCODE = 0 {
        w !,"Total Employee records= ",agg(1)
        w !,"Average Employee salary= ",agg(2)
        w !,"Sample Employee salary=",pay
    } else {
        w !,"SQL error ",SQLCODE
    }
}
```

主机变量数组

以下两个嵌入式SQL示例使用主机变量数组从一行返回非隐藏数据字段值。在这些示例中，%ID是在SELECT-Item列表中指定的，因为在默认情况下，SELECT*不返回RowId(尽管它为Sample.Person返回)；RowId始终是字段1。请注意，Sample.Person字段4和9可以为空，字段5不是数据字段(它引用Sample.Address)，字段10是隐藏的。

第一个示例返回指定数量的字段(FirstFld)；此计数中包括隐藏字段和非数据字段，但不显示。当从包含多个字段的表返回行时，使用firstfld将是合适的。请注意，此示例可以返回作为父引用的字段0。Sample.Person不是子表，因此tflds(0)未定义：

```
ClassMethod Into7()
{
    &sql(
        SELECT *,%ID INTO :tflds()
        FROM Sample.Person
    )
    if SQLCODE = 0 {
        s firstflds = 14
        for i = 0 : 1 : firstflds {
            if $d(tflds(i)) {
                w "field ",i," = ",tflds(i),!
            }
        }
    } else {
        WRITE "SQLCODE error=",SQLCODE,!
    }
}
```

第二个示例返回Sample.Person中的所有非隐藏数据字段。请注意，此示例不会尝试返回父引用Field 0，因为在Sample.Person中，tflds(0)是未定义的，因此会生成<UNDEFINED>错误：

```
ClassMethod Into8()
{
    &sql(
        SELECT *,%ID INTO :tflds()
        FROM Sample.Person
    )
    if SQLCODE=0 {
        s x = 1
        while x '="" {
            w "field ",x," = ",tflds(x),!
            s x= $ORDER(tflds(x))
        }
    } else {
```

```
        w "SQLCODE error=",SQLCODE,!
    }
}
```

下面的嵌入式SQL示例将逗号分隔的主机变量列表(用于非字段值)和主机变量数组(用于字段值)组合在一起：

```
ClassMethod Into9()
{
    &sql(
        SELECT %TABLENAME, Name, Age, AVG(Age)
        INTO :tname, :tflds(), :ageavg
        FROM Sample.Person
        WHERE Age > 50
    )
    if SQLCODE = 0 {
        w "Table name is = ",tname,!
        for i = 0 : 1 : 25 {
            if $d(tflds(i)) {
                w "field ",i," = ",tflds(i),!
            }
        }
        w "Average age is = ",ageavg,!
    } else {
        w "SQLCODE=",SQLCODE,!
    }
}
```

#SQL #Caché

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E4%B8%83%E7%AB%A0-sql%E5%91%BD%E4%BB%A4