

文章

[姚鑫](#) · 三月 18, 2021 阅读大约需分钟

## 第十二章 使用嵌入式SQL(二)

### 第十二章 使用嵌入式SQL(二)

## 嵌入式SQL代码

### 简单的SQL语句

可以将简单的SQL语句(单个嵌入式SQL语句)用于各种操作,包括:

- INSERT, UPDATE, INSERT OR UPDATE和DELETE语句。
- DDL语句。
- GRANT和REVOKE语句。
- 只返回一行的SELECT语句(或者如果只对返回的第一行感兴趣)

简单的SQL语句也被称为非基游标的SQL语句。本章稍后将介绍基游标的嵌入式SQL。

例如,以下语句查找ID为43的(唯一的)患者的姓名:

```
&sql(SELECT Name INTO :name
      FROM Patient
      WHERE %ID = 43)
```

如果对返回行的查询使用简单语句,则只返回第一行:

```
&sql(SELECT Name INTO :name
      FROM Patient
      WHERE Age = 43)
```

根据查询的不同,不能保证哪一行将被返回。此外,如果一个查询包含一个INTO语句,并且该表不包含任何数据(SQLCODE=100),执行该查询将导致未定义(空)的输出主机变量。因此,在访问输出主机变量之前,所有简单嵌入式SQL语句都应该测试SQLCODE=0。

### 架构名称解析

表名,视图名或存储过程名是合格的(指定架构名称)或不合格的(不指定架构名称)。如果名称未指定架构名称,则InterSystems IRIS会按以下方式解析架构名称:

- 数据定义:InterSystems  
IRIS使用系统范围内的默认架构来解析不合格的名称。如果默认模式不存在,则InterSystems IRIS将创建模式和相应的类包。所有数据定义语句都使用系统范围内的默认架构。数据定义语句忽略#Import和#SQLComp

ile Path宏预处理程序指令。

- 数据管理: InterSystems IRIS对包含嵌入式SQL语句的类或例程使用#SQLCompile路径和/或#Import宏预处理程序指令指定数据库搜索路径。#Import和#SQLCompile Path指令是具有不同功能的可能模式名称的相互独立列表。二者之一或两者都可用于为不合格的表、视图或存储过程名称提供数据库名称。如果未指定数据库搜索路径,则InterSystems IRIS将使用系统范围的默认数据库名称。

## 文字值

嵌入式SQL查询可能包含文字值(字符串,数字或日期)。字符串应用单引号(')引起来。(在InterSystems SQL中,双引号指定分隔标识符):

```

/// d ##class(PHA.TEST.SQL).EmbedSQL4()
ClassMethod EmbedSQL4()
{
    &sql(SELECT 'Employee (' || Name || ')' INTO :name
        FROM Sample.Employee)
    IF SQLCODE<0 {
        WRITE "SQLCODE?? ",SQLCODE," ",%msg QUIT
    } ELSEIF SQLCODE=100 {
        WRITE "??????" QUIT
    }
    WRITE name
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).EmbedSQL4()
Employee (Adams,Susan E.)

```

数值可以直接使用。在InterSystems IRIS将这些文字值与字段值进行比较之前,先对文字数字和时间戳值进行"lightly normalized",如以下示例所示,其中+0050.000被格式化为50:

```

/// d ##class(PHA.TEST.SQL).EmbedSQL5()
ClassMethod EmbedSQL5()
{
    &sql(SELECT Name,Age INTO :name,:age
        FROM Sample.Person
        WHERE Age = +0050.000)
    IF SQLCODE<0 {
        WRITE "SQLCODE?? ",SQLCODE," ",%msg QUIT
    } ELSEIF SQLCODE=100 {
        WRITE "??????" QUIT
    }
    WRITE name," age=",age
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).EmbedSQL5()
Russell,Quentin V. age=50

```

可以指定算术、函数和特殊变量表达式:

```

/// d ##class(PHA.TEST.SQL).EmbedSQL6()
ClassMethod EmbedSQL6()

```

```
{
  &sql(DECLARE C3 CURSOR FOR
    SELECT Name, Age - 65, $HOROLOG INTO :name,:retire,:today
    FROM Sample.Person
    WHERE Age > 60
    ORDER BY Age, Name)
  &sql(OPEN C3)
  QUIT:(SQLCODE'=0)
  &sql(FETCH C3)
  WHILE (SQLCODE = 0) {
    WRITE $ZDATE(today)," ",name," has ",retire," eligibility years",!
    &sql(FETCH C3)
  }
  &sql(CLOSE C3)
}
```

```
DHC-APP>d ##class(PHA.TEST.SQL).EmbedSQL6()
03/13/2021 Moon,Rhonda T. has -4 eligibility years
03/13/2021 Olsen,Ashley G. has -4 eligibility years
03/13/2021 Quixote,Terry J. has -4 eligibility years
03/13/2021 Yoders,Liza U. has -4 eligibility years
03/13/2021 Gore,Alfred M. has -3 eligibility years
03/13/2021 Houseman,Alice R. has -2 eligibility years
03/13/2021 Nichols,Heloisa M. has -2 eligibility years
03/13/2021 Houseman,Martin D. has 0 eligibility years
03/13/2021 LaRocca,David X. has 0 eligibility years
03/13/2021 Ng,Liza Z. has 0 eligibility years
03/13/2021 Smith,Elvis Y. has 0 eligibility years
```

在嵌入式SQL中，字符串文字中不允许使用以##开头的某些字符序列，而必须使##lit指定。这些字符序列是##, ##beginlit, ##expression(), ##function(), ##quote(), ##stripq(), and ##unique()。例如，以下示例失败：

```
ClassMethod EmbedSQL7()
{
  WRITE "Embedded SQL test",!
  &sql(SELECT 'the sequence ##unique( is restricted' INTO :x)
  WRITE x
}
```

以解决方法：

```
/// d ##class(PHA.TEST.SQL).EmbedSQL7()
ClassMethod EmbedSQL7()
{
  WRITE "Embedded SQL test",!
  &sql(SELECT 'the sequence ##lit(##unique() is restricted' INTO :x)
  WRITE x
}
```

```
DHC-APP>d ##class(PHA.TEST.SQL).EmbedSQL7()
Embedded SQL test
the sequence ##unique( is restricted
```

## 数据格式

在嵌入式SQL中,数据值处于“逻辑模式”。也就是说,值采用SQL查询处理器使用的本机格式。对于未定义LogicalToODBC或LogicalToDisplay转换的字符串,整数和其他数据类型,这无效。数据格式会影响%List数据类型以及%Date和%Time数据类型。

%List数据类型在逻辑模式显示为以非打印列表编码字符开头的元素值。WRITE命令将这些值显示为连续元素。例如,Sample.Person的FavoriteColors字段以%List数据类型存储数据,例如:\$LISTBUILD('Red','Black')。在嵌入式SQL中,这在逻辑模式显示为RedBlack,长度为12个字符。在“显示”模式,它显示为“Red,Black”;在ODBC模式,它显示为Red,Black,在下面的示例中显示:

```

/// d ##class(PHA.TEST.SQL).EmbedSQL8()
ClassMethod EmbedSQL8()
{
    &sql(DECLARE C4 CURSOR FOR
        SELECT TOP 10 FavoriteColors INTO :colors
        FROM Sample.Person WHERE FavoriteColors IS NOT NULL)
    &sql(OPEN C4)
    QUIT:(SQLCODE'=0)
    &sql(FETCH C4)
    WHILE (SQLCODE = 0) {
        WRITE $LENGTH(colors)," : ",colors,!
        &sql(FETCH C4)
    }
    &sql(CLOSE C4)
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).EmbedSQL8()
21: ReOrangYellow
28: ReOrangYellowGreen
35: ReOrangYellowGreenGreen
36: ReOrangYellowGreeYellow
7: White
7: Black
14: GreenWhite
8:Purple
8:Yellow
10: RedRed

```

InterSystems IRIS提供的%Date和%Time数据类型使用InterSystems IRIS内部日期表示形式(\$HOROLOG格式)作为其逻辑格式。%Date数据类型在逻辑模式返回INTEGER数据类型值;在“显示”模式为VARCHAR数据类型值,在“ODBC”模式为DATE数据类型值。%TimeStamp数据类型的逻辑,显示和ODBC格式使用ODBC日期-时间格式(YYYY-MM-DD HH:MM:SS)。

例如,考虑以下类定义:

```

Class MyApp.Patient Extends %Persistent
{
    /// Patient name
    Property Name As %String(MAXLEN = 50);

    /// Date of birth

```

```
Property DOB As %Date;  
  
/// Date and time of last visit  
Property LastVisit As %TimeStamp;  
}
```

针对该表的简单嵌入式SQL查询将以逻辑模式返回值。例如，考虑此查询：

```
&sql(SELECT Name, DOB, LastVisit  
      INTO :name, :dob, :visit  
      FROM Patient  
      WHERE %ID = :id)
```

该查询将三个逻辑值返回到主机变量名称，dob和visit中：  
主机变量

name

dob

visit

请注意，dob是\$HOROLOG格式。可以使用\$ZDATETIME函数将其转换为显示格式：

```
SET dob = 44051  
WRITE $ZDT(dob,3),!
```

与WHERE子句中的true相同的考虑因素。例如，要查找具有给定生日的患者，必须在WHERE子句中使用逻辑值：

```
&sql(SELECT Name INTO :name  
      FROM Patient  
      WHERE DOB = 43023)
```

或者，使用主机变量：

```
SET dob = $ZDH("01/02/1999",1)  
  
&sql(SELECT Name INTO :name  
      FROM Patient  
      WHERE DOB = :dob)
```

在这种情况下，我们使用\$ZDATEH函数将显示格式日期转换为其等效的\$HOROLOG逻辑值。

## 权限检查

嵌入式SQL不执行SQL特权检查。可以访问所有表，视图和列，并执行任何操作，而不管特权分配如何。假定使用嵌入式SQL的应用程序将在使用嵌入式SQL语句之前检查特权。

可以在嵌入式SQL中使用InterSystems SQL %CHECKPRIV语句来确定当前权限。

[#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%E4%BA%8C%E5%BC%89>