

文章

[姚鑫](#) · 三月 13 阅读大约需分钟

第八十章 SQL函数 \$LIST(二)

第八十章 SQL函数 \$LIST(二)

示例

在以嵌入式 SQL 示例中,两个 WRITE 语句都返回“Red”,即列表中的第一个元素。第一个默认写入第一个元素,第二个写入第一个元素,因为位置参数设置为 1:

```
/// d ##class(PHA.TEST.SQLFunction).Li4()
ClassMethod Li4()
{
    s a = $lb("Red", "Blue", "Green")
    &sql(SELECT $LIST(:a), $LIST(:a,1)
        INTO :b,:c )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,"The one-arg sublist is ",b
        w !,"The two-arg sublist is ",c }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li4()
```

```
The one-arg sublist is Red
The two-arg sublist is Red
```

以嵌入式 SQL 示例返回列表中的第二个元素“Blue”:

```
/// d ##class(PHA.TEST.SQLFunction).Li5()
ClassMethod Li5()
{
    s a = $lb("Red", "Blue", "Green")
    &sql(
        SELECT $LIST(:a,2)
        INTO :b
    )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,"The second element is ",b
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li5()
```

The second element is Blue

以嵌入式 SQL 示例返回“Red

Blue”，这是一个双元素列表字符串，从列表中第一个元素开始，到第二个元素结束。使用 ZZDUMP 而不是 WRITE，因为列表字符串包含特殊(非打印)编码字符：

```
/// d ##class(PHA.TEST.SQLFunction).Li6()
ClassMethod Li6()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:a,1,2)
    INTO :b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    w !,"The encoded sublist is"
    ZZDUMP b ; Prints "Red Blue "
  }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li6()
```

The encoded sublist is

```
0000: 05 01 52 65 64 06 01 42 6C 75 65          ..Red..Blue
```

以嵌入式 SQL 示例返回未知长度列表中的最后一个元素。在这里，最后一个元素就作为普通字符串返回，然后作为编码列表字符串返回：

```
/// d ##class(PHA.TEST.SQLFunction).Li7()
ClassMethod Li7()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LISTLENGTH(:a), $LIST(:a, -1)
    INTO :b,:plain
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    &sql(
      SELECT $LIST(:a,:b,-1)
      INTO :encoded
    )
    if SQLCODE '= 0 {
      w !,"Error code ",SQLCODE
    } else {
      w !,"The final element as a string: ",plain
      w !,"The final element as an encoded string: "
      ZZDUMP encoded
    }
  }
}
```

```
    }  
  }  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li7()
```

The final element as a string: Green

The final element as an encoded string:

```
0000: 07 01 47 72 65 65 6E
```

```
..Green
```

注意

无效的参数值

如果列表参数中的表达式未计算为有效列表,则会生成QLCODE -400 致命错误:

```
/// d ##class(PHA.TEST.SQLFunction).Li8()  
ClassMethod Li8()  
{  
  s a = "the quick brown fox"  
  &sql(  
    SELECT $LIST(:a, 1)  
    INTO :b  
  )  
  if SQLCODE '= 0 {  
    w !,"Error code ",SQLCODE  
  } else {  
    w !,"The sublist is"  
    ZZDUMP b ; Variable not set  
  }  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li8()
```

```
Error code -400
```

如果 position 参数或 end 参数的值小于 -1,则会生成QLCODE -400 致命错误:

```
/// d ##class(PHA.TEST.SQLFunction).Li9()  
ClassMethod Li9()  
{  
  s a = $lb("Red", "Blue", "Green")  
  &sql(  
    SELECT $LIST(:a, -2, 3)  
    INTO :b  
  )  
  if SQLCODE '= 0 {  
    w !,"Error code ",SQLCODE  
  } else {  
    w !,"The sublist is"  
    ZZDUMP b ; Variable not set  
  }  
}
```

```
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li9()
```

```
Error code -400
```

如果 position 参数的值引用了一个不存在的列表成员并且没有使用 end 参数,则会生成 QLCODE -400 致命错误:

```
/// d ##class(PHA.TEST.SQLFunction).Li10()
ClassMethod Li10()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:a, 7)
    INTO :b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    w !,"The sublist is"
    ZZDUMP b ; Variable not set
  }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li10()
```

```
Error code -400
```

但是,如果使用 end 参数,则不会发生错误,并且返回 null 字符串。

```
/// d ##class(PHA.TEST.SQLFunction).Li11()
ClassMethod Li11()
{
  s a = $lb("Red", "Blue", "Green")
  &sql(
    SELECT $LIST(:a, 7, -1)
    INTO :b
  )
  if SQLCODE '= 0 {
    w !,"Error code ",SQLCODE
  } else {
    w !,"Error code ",SQLCODE
    w !,"The sublist is"
    ZZDUMP b ; Prints a null string
  }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li11()
```

```
Error code 0
The sublist is
```

如果 position 参数值标识具有未定义值的元素,则会生成SQLCODE -400 致命错误:

```

/// d ##class(PHA.TEST.SQLFunction).Lil2()
ClassMethod Lil2()
{
    s a = $lb("Red", "Blue", "Green")
    &sql(
        SELECT $LIST(:a, 2)
        INTO :b
    )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,"The sublist is"
        ZZDUMP b ; Variable not set
    }
}

```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Lil2()
```

```

The sublist is
0000: 42 6C 75 65                                     Blue

```

两个参数和三个参数 \$LIST

\$LIST(list,1) 不等价于 \$LIST(list,1,1) 因为前者返回一个字符串,而后者返回一个单元素列表字符串。如果没有要返回的元素,则双参数形式不返回值;三参数形式返回一个空字符串。

Unicode

如果一个 Unicode 字符出现在列表元素中,则整个列表元素将表示为 Unicode(宽)字符。列表中的其他元素不影响。

以下嵌入式 SQL 示例显示了两个列表。a 列表由两个仅包含 ASCII 字符的元素组成。列表由两个元素组成一个元素包含一个 Unicode 字符(\$CHAR(960) = pi 符号);第二个元素仅包含 ASCII 字符。

```

/// d ##class(PHA.TEST.SQLFunction).Lil3()
ClassMethod Lil3()
{
    s a = $lb("ABC" _ $CHAR(68), "XYZ")
    s b = $lb("ABC" _ $CHAR(960), "XYZ")
    &sql(
        SELECT $LIST(:a, 1), $LIST(:a, 2), $LIST(:b, 1), $LIST(:b, 2)
        INTO :a1, :a2, :b1, :b2
    )
    if SQLCODE '= 0 {
        w !,"Error code ",SQLCODE
    } else {
        w !,"The ASCII list a elements: "
        ZZDUMP a1
        ZZDUMP a2
        w !,"The Unicode list b elements: "
    }
}

```

```
        ZZDUMP b1
        ZZDUMP b2
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Li13()
```

The ASCII list a elements:

```
0000: 41 42 43 44                                ABCD
```

```
0000: 58 59 5A                                    XYZ
```

The Unicode list b elements:

```
0000: 0041 0042 0043 03C0                        ABC?
```

```
0000: 58 59 5A                                    XYZ
```

请注意, IRIS 将 b 的第一个元素完全编码为宽 Unicode 字符。b 的第二个元素不包含 Unicode 字符,因此 IRIS 使用窄 ASCII 字符对其进行编码。

[#SQL #Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AB%E5%8D%81%E7%AB%A0-sql%E5%87%BD%E6%95%B0-list%EF%BC%88%E4%BA%8C%EF%BC%89>