文章
姚 鑫 · 五月 3, 2021 阅读大约需 6 分钟

第二章 全局变量结构（一）

第二章 全局变量结构（一）

本章描述全局变量的逻辑视图，并概述全局变量是如何在磁盘上物理存储的。

全局变量的逻辑结构

全局变量是存储在物理InterSystems IRIS®数据库中的命名多维数组。
在应用程序中，全局变量到物理数据库的映射基于当前名称空间——名称空间提供一个或多个物理数据库的逻辑统一视图。

全局命名约定和限制

全局名称指定其目标和用途。有两种类型的全局变量和一组单独的变量，称为“ 进程私有全局变量 ”：
- 全局变量 -
这就是所谓的标准全局变量；通常，这些变量被简称为全局变量。它是驻留在当前命名空间中的永久性多维数组。
- **扩展全局引用-这是位于当前命名空间以外的命名空间中的全局引用。**
- **进程私有全局变量-这是一个数组变量，只有创建它的进程才能访问。**

全局变量的命名约定如下：

- 全局变量名称以脱字符(^)前缀开头。这个插入符号区分全局变量和局部变量。
- 全局变量名称中脱字符(^)前缀后的第一个字符可以是：
  - 字母或百分号字符(%)-仅适用于标准全局变量。对于全局变量名称，字母被定义为ASCII 65到ASCII 255范围内的字母字符。如果全局名称以“ % ”开头(但不是“ %Z ”或“ %z ”)，则此全局名称供InterSystems IRIS系统使用。%GLOBAL通常存储在IRISSYS或IRISLIB数据库中。
  - 竖线(|)或左方括号([)-表示扩展全局引用或进程专用全局变量。使用取决于后续字符。
- 全局变量名称的其他字符可以是字母、数字或句号(.)字符。
  百分比(%)字符不能使用，除非作为全局名称的第一个字符。
  “ . ”字符不能作为全局名称的最后一个字符。
- **全局名称最长可达31个字符(不包括脱字符前缀)。可以指定更长的全局名称，但InterSystems IRIS只将前31个字符视为重要字符。**
- 全局名称区分大小写。
- InterSystems IRIS对全局引用的总长度施加限制，而该限制又对任何下标值的长度施加限制。

在IRISSYS数据库中，InterSystems将除以“ z ”、“ Z ”、“ %z ”和“ %Z ”开头的所有全局变量名称保留给自己。在所有其他数据库中，InterSystems保留所有以“ ISC ”开头的全局名称。和“ %isc. ”。

示例全局名称及其用法

以下是各种全局名称的示例以及每种名称的用法：
- ^globalname - 标准全局变量
- ^|"environment"|globalname - 扩展全局变量引用的环境语法
- ^||globalname - 进程私有全局变量
- ^|"^"| - 进程私有全局变量

- ^[namespace]globalname - 扩展全局变量引用中显式命名空间的括号语法
- ^[directory,system]globalname - 扩展全局变量引用中隐含命名空间的括号语法
- ^["^"]globalname - 进程私有全局变量
- ^["^",""]globalname - 进程私有全局变量

注意：全局名称只能包含有效的标识符字符；默认情况下，这些字符如上所述。但是，NLS(国家语言支持)定义了一组不同的有效标识符字符集。全局名称不能包含Unicode字符。

因此，以下都是有效的全局名称：

```
SET ^a="The quick "
SET ^A="brown fox "
SET ^A7="jumped over "
SET ^A.7="the lazy "
SET ^A1B2C3="dog's back."
WRITE ^a,^A,^A7,!,^A.7,^A1B2C3
KILL ^a,^A,^A7,^A.7,^A1B2C3 // keeps the database clean
```

全局节点和下标简介

全局通常有多个节点，通常由一个下标或一组下标标识。下面是一个基本示例：

```
set ^Demo(1)="Cleopatra"
```

此语句引用全局节点^Demo(1)，它是^Demo全局节点中的一个节点。此节点由一个下标标识。

再举一个例子：

```
set ^Demo("subscript1","subscript2","subscript3")=12
```

该语句指的是全局节点^Demo("subscript1","subscript2","subscript3")，它是同一全局中的另一个节点。此节点由三个下标标识。

再举一个例子：

```
set ^Demo="hello world"
```

该语句引用不使用任何下标的全局节点^Demo。

全局的节点形成分层结构。ObjectScript提供了利用此结构的命令。例如，可以删除节点或删除节点及其所有子节点。

全局变量下标

下标有以下规则：

- 下标数值区分大小写。
- **下标值可以是任何ObjectScript表达式，前提是该表达式的计算结果不是空字符串("")。**

该值可以包括所有类型的字符，包括空格、非打印字符和Unicode字符。(请注意，非打印字符在下标数值中不太实用。)

- 在解析全局引用之前，InterSystems IRIS计算每个下标的方式与计算任何其他表达式的方式相同。在下面的示例中，我们设置了^Demo全局的一个节点，然后以几种等效的方式引用该节点：

```
DHC-APP>s ^Demo(1+2+3)="a value"

DHC-APP>w ^Demo(3+3)
a value
DHC-APP>w ^Demo(03+03)
a value
DHC-APP>w ^Demo(03.0+03.0)
a value
DHC-APP>set x=6

DHC-APP>w ^Demo(x)
a value
```

- InterSystems IRIS对全局引用的总长度施加限制，而该限制又对任何下标值的长度施加限制。

注意：上述规则适用于IRIS支持的所有排序规则。对于出于兼容性原因仍在使用的旧归类，如“ pre-ISM-6.1 ”，下标的规则有更多限制。例如，字符下标不能以控制字符作为其初始字符；整数下标中可以使用的位数也有限制。

全局变量节点

在应用程序中，节点通常包含以下类型的结构：
1. 字符串或数字数据，包括本机Unicode字符。
2. 具有由特殊字符分隔的多个字段的字符串：

```
 SET ^Data(10) = "Smith^John^Boston"
```

可以使用ObjectScript $PIECE 函数来拆分这些数据。

3. InterSystems IRIS $LIST
   结构中包含多个字段。$LIST结构是包含多个长度编码值的字符串。它不需要特殊的分隔符。
4. 空字符串 ("")。在下标本身用作数据的情况下，实际节点中不存储任何数据。
5. 一个位串。如果全局变量用于存储位图索引的一部分，那么存储在节点中的值就是位字符串。位串是包含1和0值的逻辑压缩集的字符串。可以使用$BIT函数构造位串。
6. 更大的数据集的一部分。例如，对象和SQL引擎将流(BLOB)存储为全局中连续的32K节点系列。通过流接口，流的用户不知道流是以这种方式存储的。

请注意，任何全局节点都不能包含长度超过字符串长度限制的字符串，字符串长度限制非常长。

全局变量排序规则

在全局中，节点按排序(排序)顺序存储。

应用程序通常通过将转换应用于用作下标的值来控制节点的排序顺序。例如，SQL引擎在为字符串值创建索引时，会将所有字符串值转换为大写字母，并在前面加上一个空格字符，以确保索引不区分大小写并且以文本形式排序(即使数值存储为字符串)。

全局变量引用的最大长度

全局变量引用(即对特定全局节点或子树的引用)的总长度限制为511个编码字符(少于511个键入字符)。

要保守地确定给定全局变量引用的大小，请使用以下准则：

1. 全局变量名称：每个字符加1。
2. 对于纯数字下标：每个数字、符号或小数点加1。
3. 对于包含非数字字符的下标：为每个字符添加3。

如果下标不是纯数字的，则根据用于编码字符串的字符集的不同，下标的实际长度会有所不同。一个多字节字符最多可以占用3个字节。

请注意，ASCII字符可能占用1或2字节。
如果排序规则进行大小写折叠，那么ASCII字符可以使用1个字节表示字符，1个字节表示消除歧义字节。
如果排序不执行大小写折叠，ASCII字符占用1字节。

4. 每个下标加1。

如果这些数字的总和大于511，则引用太长。

由于确定限制的方式，如果必须使用长下标或全局名称，这有助于避免使用大量下标级别。
相反，如果使用多个下标级别，则应避免长全局名称和长下标。
因为无法控制正在使用的字符集，所以保持全局名称和下标更短是很有用的。

当对特定引用有疑问时，创建与最长预期全局变量引用长度相等(甚至稍长一点)的全局变量引用的测试版本是有用的。
这些测试的数据为构建应用程序之前可能修订的命名约定提供了指导。

#SQL #Caché #InterSystems IRIS #InterSystems IRIS for Health

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E7%AB%A0-%E5%85%A8%E5%B1%80%E5%8F%98%E9%87%8F%E7%BB%93%E6%9E%84%EF%BC%88%E4%B8%80%EF%BC%89