

文章

姚鑫 · 六月 10, 2022 阅读大约需 8 分钟

第五章 数据类型（四）

第五章 数据类型（四）

Strings

%Library.String 数据类型支持的最大字符串长度为 3,641,144 个字符。通常，极长的字符串应分配为 %Stream.GlobalCharacter 数据类型之一。

因为 IRIS 支持 xDBC 协议 50 和更高版本，所以没有强制执行 ODBC 或 JDBC 字符串长度限制。如果 IRIS 实例和 ODBC 驱动程序支持不同的协议，则使用两个协议中较低的一个。实际使用的协议记录在 ODBC 日志中。

请注意，默认情况下 IRIS 建立系统范围的 ODBC VARCHAR 最大长度为 4096；此 ODBC 最大长度是可配置的。

列表结构

IRIS 支持列表结构数据类型 %List（数据类型类 %Library.List）。这是一种压缩的二进制格式，不会映射到 SQL 的相应本机数据类型。在其内部表示中，它对应于数据类型 VARBINARY，默认 MAXLEN 为 32749。IRIS 支持列表结构数据类型 %ListOfBinary（数据类型类 %Library.ListOfBinary）对应于数据类型 VARBINARY，默认 MAXLEN 为 4096。

因此，动态 SQL 不能在 WHERE 子句比较中使用 %List 数据。也不能使用 INSERT 或 UPDATE 来设置 %List 类型的属性值。

动态 SQL 将列表结构化数据的数据类型返回为 VARCHAR。要确定查询中的字段是数据类型 %List 还是 %ListOfBinary，可以使用 select-item columns metadata isList 布尔标志。这些数据类型的 CType（客户端数据类型）整数代码是 6。

如果使用 ODBC 或 JDBC 客户端，则使用 LogicalToOdbc 转换将 %List 数据投影到 VARCHAR 字符串数据。列表被投影为一个字符串，其元素由逗号分隔。这种类型的数据可以用在 WHERE 子句以及 INSERT 和 UPDATE 语句中。请注意，默认情况下，IRIS 建立系统范围的 ODBC VARCHAR 最大长度为 4096；此 ODBC 最大长度是可配置的。

SQL 支持八种列表函数：\$LIST、\$LISTBUILD、\$LISTDATA、\$LISTFIND、\$LISTFROMSTRING、\$LISTGET、\$LISTLENGTH 和 \$LISTTOSTRING。ObjectScript 支持三个额外的列表函数：\$LISTVALID 用于确定表达式是否为列表，\$LISTSAME 用于比较两个列表，以及 \$LISTNEXT 用于从列表中顺序检索元素。

位数据类型

BIT (%Library.Boolean) 数据类型接受 0、1 和 NULL 作为有效值。

- 在逻辑和 ODBC 模式下，唯一接受的值是 0、1 和 NULL。
- 在显示模式下，DisplayToLogical 方法首先将非空输入值转换为 0 或 1，如下所示：
 - 非零数字或数字字符串 = 1，例如 3, '0.1', '-1', '7dwarves'
 - 非数字字符串 = 0。例如，“true”或“false”。
 - 空字符串 = 0。例如”。

流数据类型

Stream 数据类型对应于 IRIS 类属性数据类型 %Stream.GlobalCharacter（用于 CLOB）和 %Stream.GlobalBinary（用于 BLOB）。这些数据类型类可以使用指定的 LOCATION 参数定义流字段，或者省略该参数并默认为系统定义的存储位置。

具有 Stream 数据类型的字段不能用作大多数 SQL 标量、聚合或一元函数的参数。尝试这样做会生成 SQLCODE -37 错误代码。

具有 Stream 数据类型的字段不能用作大多数 SQL 谓词条件的参数。尝试这样做会生成 SQLCODE -313 错误代码。

Stream 数据类型在索引中的使用以及在执行插入和更新时也受到限制。

串行数据类型

具有 SERIAL (%Library.Counter) 数据类型的字段可以采用用户指定的正整数值，或者 IRIS 可以为其分配一个连续的正整数值。%Library.Counter 扩展了 %Library.BigInt。

INSERT 操作为 SERIAL 字段指定以下值之一：

- 无值、0（零）或非数字值：IRIS 忽略指定值，而是将此字段的当前串行计数器值增加 1，并将结果整数插入该字段。
- 正整数值：IRIS 将用户指定的值插入到字段中，并将该字段的串行计数器值更改为此整数值。

因此，SERIAL 字段包含一系列增量整数值。这些值不一定是连续的或唯一的。例如，以下是 SERIAL 字段的有效值系列：1、2、3、17、18、25、25、26、27。连续整数要么是 IRIS 生成的，要么是用户提供的；非连续整数是用户提供的。如果希望 SERIAL 字段值是唯一的，则必须对该字段应用 UNIQUE 约束。

UPDATE 操作对自动分配的 SERIAL 计数器字段值没有影响。但是，使用 INSERT OR UPDATE 执行的更新会导致对 SERIAL 字段的后续插入操作跳过整数序列。

如果该字段当前没有值（NULL），或者它的值为 0，则 UPDATE 操作只能更改串行字段值。否则，将生成 SQLCODE -105 错误。

IRIS 对表中的 SERIAL 字段的数量没有限制。

ROWVERSION 数据类型

ROWVERSION 数据类型定义了一个只读字段，该字段包含一个唯一的系统分配的正整数，从 1 开始。IRIS 分配顺序整数作为每个插入、更新或 %Save 操作的一部分。这些值不是用户可修改的。

IRIS 在命名空间范围内维护一个单行版本计数器。命名空间中包含 ROWVERSION 字段的所有表共享相同的行版本计数器。因此，ROWVERSION 字段提供行级版本控制，允许确定对命名空间中一个或多个表中的行进行更改的顺序。

每个表只能指定一个 ROWVERSION 数据类型的字段。

ROWVERSION 字段不应包含在唯一键或主键中。ROWVERSION 字段不能是 IDKey 索引的一部分。

ROWVERSION 和 SERIAL 计数器

作为 INSERT 操作的一部分，ROWVERSION 和 SERIAL (%Library.Counter) 数据类型字段都从内部计数器接收顺序整数。但是这两个计数器有很大的不同，并且用于不同的目的：

- ROWVERSION 计数器位于命名空间级别。 SERIAL 计数器位于表级别。这两个计数器完全相互独立，独立于 RowID 计数器。
- ROWVERSION 计数器通过插入、更新或 %Save 操作递增。 SERIAL 计数器仅由插入操作递增。使用 INSERT OR UPDATE 执行的更新可能会导致 SERIAL 计数器序列出现间隙。
- ROWVERSION 字段值不能由用户指定；该值始终由 ROWVERSION 计数器提供。如果没有为该字段指定值，则在插入期间从表的内部计数器提供一个 SERIAL 字段值。如果插入提供了一个 SERIAL 整数，则插入该值而不是当前计数器值：
 - 如果插入提供的 SERIAL 字段值大于当前内部计数器值， IRIS 将该值插入该字段并将内部计数器重置为该值。
 - 如果插入提供的 SERIAL 字段值小于当前计数器值， IRIS 不会重置内部计数器。
 - 插入可以提供 SERIAL 字段值作为负整数或小数。 IRIS 将小数截断为其整数部分。如果提供的 SERIAL 字段值为 0 或 NULL， IRIS 将忽略用户提供的值并插入当前的内部计数器值。
 - 不能更新现有的 SERIAL 字段值。
- ROWVERSION 字段值始终是唯一的。因为可以插入用户指定的 SERIAL 字段值，所以必须指定 UNIQUE 字段约束以保证唯一的 SERIAL 字段值。
- 无法重置 ROWVERSION 计数器。 TRUNCATE TABLE 重置 SERIAL 计数器；对所有行执行 DELETE 不会重置 SERIAL 计数器。
- 每个表只允许一个 ROWVERSION 字段。可以在一个表中指定多个 SERIAL 字段。

ODBC / JDBC 公开的 DDL 数据类型

ODBC 公开了 DDL

数据类型的子集，并将其他数据类型映射到该数据类型的子集。这些映射是不可逆的。例如，语句 CREATE TABLE mytable (f1 BINARY) 创建一个 IRIS 类，该类作为 mytable (f1 VARBINARY) 投影到 ODBC。 IRIS 列表数据类型作为 VARCHAR 字符串投影到 ODBC。

ODBC 公开以下数据类型：BIGINT、BIT、DATE、DOUBLE、GUID、INTEGER、LONGVARBINARY、LONGVARCHAR、NUMERIC、OREF、POSIXTIME、SMALLINT、TIME、TIMESTAMP、TINYINT、VARBINARY、VARCHAR。请注意，默认情况下 IRIS 建立系统范围的 ODBC VARCHAR 最大长度为 4096；此 ODBC 最大长度是可配置的。

当这些 ODBC/JDBC 数据类型值之一映射到 SQL 时，会发生以下操作：使用 \$DOUBLE 强制转换 DOUBLE 数据。 NUMERIC 数据使用 \$DECIMAL 进行转换。

GUID 数据类型对应于 SQL UNIQUEIDENTIFIER 数据类型。未能为 GUID / UNIQUEIDENTIFIER 字段指定有效值会生成 #7212 一般错误。要生成 GUID 值，请使用 %SYSTEM.Util.CreateGUID() 方法。

查询元数据返回数据类型

可以使用动态 SQL 返回有关查询的元数据，包括查询中指定列的数据类型。

以下动态 SQL 示例为 Sample.Person 和 Sample.Employee 中的每个列返回列名和 ODBC 数据类型的整数代码：

```
/// d ##class(PHA.TEST.SQLFunction).QueryMetadataReturnsDataType()  
ClassMethod QueryMetadataReturnsDataType()  
{  
    s myquery = "SELECT * FROM Sample.Person"  
    s tStatement = ##class(%SQL.Statement).%New()  
    s tStatus = tStatement.%Prepare(myquery)  
    s x = tStatement.%Metadata.columnCount  
    while x > 0 {  
        s column = tStatement.%Metadata.columns.GetAt(x)  
        w !,x," ",column.colName," ",column.ODBCType  
        s x = x-1  
    }  
    w !,"end of columns"
```

```
}
```

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

```
16 Office_Zip 12
15 Office_Street 12
14 Office_State 12
13 Office_City 12
12 Home_Zip 12
11 Home_Street 12
10 Home_State 12
9 Home_City 12
8 Spouse 4
7 SSN 12
6 Name 12
5 FavoriteColors 12
4 DOB 9
3 Age 4
2 AddDateTime 11
1 ID 4
end of columns
```

```
/// d ##class(PHA.TEST.SQLFunction).QueryMetadataReturnsDataType1()
ClassMethod QueryMetadataReturnsDataType1()
{
    s myquery = "SELECT * FROM Sample.Employee"
    s tStatement = ##class(%SQL.Statement).%New()
    s tStatus = tStatement.%Prepare(myquery)
    s x = tStatement.%Metadata.columnCount
    while x > 0 {
        s column = tStatement.%Metadata.columns.GetAt(x)
        w !,x," ",column.colName," ",column.ODBCType
        s x = x - 1
    }
    w !,"end of columns"
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).QueryMetadataReturnsDataType1()
```

```
20 Office_Zip 12
19 Office_Street 12
18 Office_State 12
17 Office_City 12
16 Home_Zip 12
15 Home_Street 12
14 Home_State 12
13 Home_City 12
12 Title 12
11 Spouse 4
10 Salary 4
9 SSN 12
8 Picture -4
7 Notes -1
6 Name 12
5 FavoriteColors 12
```

Published on InterSystems Developer Community (<https://community.intersystems.com>)

#SQL #Caché

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E7%AB%A0-%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B%E5%BC%88%E5%9B%9B%E5%BC%89>