文章
姚 鑫 · 三月 5, 2021 阅读大约需 9 分钟

第四章 SQL标识符

第四章 标识符

标识符

标识符是SQL实体的名称，例如表、视图、列(字段)、模式、表别名、列别名、索引、存储过程、触发器或其他SQL实体。
标识符名称在其上下文中必须是唯一的;
例如，同一模式中的两个表或同一表中的两个字段不能具有相同的名称。
但是，不同模式中的两个表或不同表中的两个字段可以具有相同的名称。
在大多数情况下，相同的标识符名称可以用于不同类型的SQL实体;
例如，一个模式、该模式中的表以及该表中的字段都可以具有相同的名称，而不会产生冲突。
但是，同一个模式中的表和视图不能具有相同的名称。

InterSystems IRIS®数据平台SQL标识符遵循一组命名约定，根据标识符的使用，这可能会受到进一步的限制。
标识符不区分大小写。

标识符可以是简单标识符，也可以是分隔符。
InterSystems SQL默认支持简单标识符和分隔标识符。

简单标识符

简单标识符有以下语法:

```
simple-identifier ::= identifier-start { identifier-part }
      identifier-start ::= letter | % | _
      identifier-part ::=  letter | number | _ | @ | # | $
```

命名约定

标识符start是SQL标识符的第一个字符。
它必须是下列之一:

- 大写或小写字母。
  字母定义为通过ObjectScript $ZNAME函数验证的任何字符;
  默认情况下，这些字母是大写字母A到Z (ASCII 65-90)，小写字母a到z (ASCII 97-122)，以及带有重音标记的字母(ASCII 192-255，不包括ASCII 215和247)。
  InterSystems IRIS可以在SQL标识符中使用任何有效的Unicode(16位)字母字符。
  简单的标识符是不区分大小写的(不过，请参见下面的内容)。
  按照惯例，它们用首字母大写来表示。

日语区域设置不支持标识符中的重音拉丁字母字符。
日语标识符可能包含(除了日语字符之外)拉丁字母字符A-Z和a-z(65-90和97-122)，以及希腊大写字母字符(913-929和931-937)。

- 一个下划线(_)。
- 百分号(%)。InterSystems IRIS以%字符开头的名称(以%Z或%z开头的除外)保留为系统元素，不应用作标识符。

标识符部分是SQL标识符的任何后续字符。这些剩余字符可能由零个或多个字符组成:

- 字母(包括Unicode字符)。
- 数字。数字被定义为数字0到9。
- 下划线(_)。
- At标志(@)。
- 井号(#)。
- 美元符号($)。

一些符号字符也用作运算符。
在SQL中，#符号用作模运算符。在SQL中，下划线字符可以用来连接两个字符串；提供这种用法是为了与ObjectScript兼容，首选的SQL串联运算符是||。 将符号解释为标识符字符总是优先于将其解释为运算符。任何关于符号字符作为运算符的正确解析的歧义都可以通过在运算符前后添加空格来解决。

简单标识符不能包含空格或非字母数字字符(上面指定的符号字符除外)。系统间SQL导入工具从导入的表名中删除空格。

注意:SQL游标名称不遵循标识符命名约定。

InterSystems SQL包含不能用作简单标识符的保留字。
有关这些保留词的列表,
要测试一个单词是否是保留单词，请使用$SYSTEM.SQL.IsReservedWord()方法。
但是，带分隔符的标识符可以与SQL保留字相同。

任何不遵循这些命名约定的标识符都必须在SQL语句中表示为带分隔符的标识符。

字母

默认情况下，InterSystems SQL标识符不区分大小写。
InterSystems SQL通过将标识符转换为所有大写字母后比较它们来实现这一点。
这对名称的实际使用情况没有影响。
(注意，SQL的其他实现可能会以不同的方式处理标识符的大小写敏感性。
因此，建议避免使用基于案例的标识符。)

请注意，系统间SQL中的游标名称和密码是区分大小写的。

测试有效标识符

InterSystems IRIS提供了%SYSTEM.SQL的IsValidRegularIdentifier()方法。它测试字符串是否是有效的标识符。它测试字符用法和保留字。它还执行200个字符的最大长度测试(这是用于避免错误输入的任意长度；这不是标识符验证)。以下对象脚本示例显示了此方法的使用:

```
/// d ##class(PHA.TEST.SQL).Identifiers()
ClassMethod Identifiers()
{
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("Fred")
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("%Fred#123")
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("%#$@_Fred")
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("_1Fred")
```

```
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("%#$")

    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("1Fred")
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("Fr ed")
    WRITE !,$SYSTEM.SQL.IsValidRegularIdentifier("%sqlupper")
}


DHC-APP>d ##class(PHA.TEST.SQL).Identifiers()

1
1
1
1
1
0
0
0
```

前三个方法调用返回1，表示有效的标识符。第四个和第五个方法调用也返回1；这些是有效的标识符，尽管它们不能用作表名或字段名。最后三个方法调用返回0，表示标识符无效。其中两个是无效的，因为它们违反了字符规则——在这些情况下是以数字开头或包含空格。最后一次方法调用返回0，因为指定的字符串是保留字。请注意，这些规则测试是最低要求；它们不能证明标识符对所有的SQL使用都有效。

这个方法也可以作为存储过程从ODBC或JDBC调用:%SYSTEM.SQLIsValidRegularIdentifier("nnnn")。

名称空间的名字

命名空间名称(也称为数据库名称)遵循标识符命名约定，并对标点字符和最大长度有额外的限制。

命名空间名称可以作为带分隔符的标识符，并且可以与SQL保留字相同。
但是，相同的命名空间名称标点限制适用于简单标识符和分隔标识符。

标识符和类实体名称

通过去除非字母数字字符，SQL表名、视图名、字段名、索引名、触发器名和过程名用于生成相应的持久类实体。
生成的类实体和全局变量的名称遵循这些规则。

注意:命名空间名称和SQL模式名称以及相应的包名称不遵循这些规则。

- 仅在包含标点字符方面不同的标识符是有效的。
  因为类对象名称不能包含标点字符，InterSystems IRIS通过去掉所有标点字符来生成相应的唯一对象名称。
  如果去掉标识符的标点字符会导致非唯一的类对象名称，InterSystems
  IRIS将最后一个字母数字字符替换为一个递增的字符后缀，从而创建一个唯一的名称。

对于表、视图、字段、触发器和过程类方法名，这是一个以0开头的整数后缀。
例如，myname和myname生成myname和mynam0，添加我的#name生成mynam1。
如果生成的惟一名称的数量大于10 (mynam9)，则通过替换以(mynamA)开头的大写字母后缀生成额外的名称。
因为表和视图共享相同的名称空间，所以表或视图的后缀计数器都是递增的。

对于索引名，这个后缀是一个大写字母，以a开头。例如，myindex和myindex生成myindex和myindeA。

如果定义了一个以后缀字符结束的名称(例如myname0或myindex)， InterSystems
IRIS将通过递增到下一个未使用的后缀来处理惟一名称的生成。

- 第一个字符为标点字符，第二个字符为数字的标识符对于表名、视图名或过程名无效。
  它们对字段名和索引名有效。
  如果SQL字段名或索引名的第一个字符是标点字符(%或)，第二个字符是数字，InterSystems IRIS将追加小写的“ n ”作为相应属性名的第一个字符。
- 完全由标点字符组成的标识符，或以两个下划线字符(name)开头的标识符，或包含两个井号(nn##nn)的标识符作为SQL实体名称通常是无效的，应该在所有上下文中避免使用。

可以将SQL标识符中的特定字符转换为相应对象标识符中的其他字符。
在允许的标识符字符规则不同的环境中，这有助于标识符的使用。
使用 %SYSTEM.SQL的SetDDLIdentifierTranslations()方法。
要确定当前设置，调用$SYSTEM.SQL.CurrentSettings()。

在DDL运行时将SQL标识符转换为对象标识符时，“ From ”字符串中的字符被转换为“ to ”字符串中的字符。

在类定义中指定SQL名称

定义投射SQL实体的持久化类时，每个SQL实体的名称与其对应的持久化类定义元素的名称相同。
要使SQL表、字段或索引名称不同，可以使用SqlTableName、SqlFieldName或SqlName(对于索引)关键字在类定义中指定SQL名称。
例如:

```
Property LName As %String [SqlFieldName = "Family#Name"];


Index NameIdx As %String [SqlName = "FullNameIndex"];
```

标识符长度注意事项

SQL标识符的最大长度为128个字符。当InterSystems IRIS将SQL标识符映射到相应的对象实体时，它会创建最多96个字符的相应属性、方法、查询或索引名称。如果前96个字符的两个SQL标识符相同，InterSystems IRIS会将相应对象名称的第96个字符替换为整数(从0开始)以创建唯一名称。

分隔标识符

分隔标识符的语法如下：

```
delimited-identifier ::= " delimited-identifier-part { delimited-identifier-part } "
    delimited-identifier-part ::= non-double-quote-character | double-quote-symbol
    double-quote-symbol ::= ""
```

带分隔符的标识符是由分隔符字符括起来的唯一标识符。InterSystems SQL支持双引号(“)作为分隔符，分隔符一般用于避免简单标识符的命名限制。

请注意，InterSystems SQL使用单引号字符(‘)来分隔文字。因此，必须使用双引号字符(“)指定分隔标识符，必须使用单引号字符(‘)指定文字。例如，’ 7 ‘ 是数字文字7，但 ” 7 “ 是分隔标识符。当SQL语句用双引号括起来时(例如，在动态SQL中)，该字符串中的双引号字符必须是双引号。

SQL空字符串应始终指定为一对单引号字符‘ ’。启用分隔标识符支持时，一对双引号字符“ ”将被解析为无效的分隔标识符，并生成SQLCODE-1错误。

分隔标识符有效名称

分隔的标识符必须是唯一的名称。带分隔符的标识符不区分大小写；按照惯例，标识符用首字母大写表示。

分隔标识符可以与SQL保留字相同。分隔标识符通常用于避免与SQL保留字的命名冲突。

分隔标识符几乎可以包含任何可打印字符，包括空格。大多数分隔的标识符名称不能包含以下字符：逗号(，)、句点(.)、插入符号(^)和两个字符的箭头序列(->)；但是分隔的标识符角色名称和用户名可以包含这些字符。分隔的标识符类名可以包含句点(.)。任何分隔的标识符都不能以星号(*)开头。以下术语不能用作分隔标识符：%vid。违反这些命名约定会导致SQLCODE-1错误。

用作表、架构、列或索引名的分隔标识符必须能够转换为有效的类实体名称。因此，它必须至少包含一个字母数字字符。以数字(或标点符号后跟数字)开头的分隔标识符会生成带有字母“ n ”前缀的相应类实体名称。

以下示例显示了对列名和表名使用分隔标识符的查询：

```
SELECT "My Field" FROM "My Table" WHERE "My Field" LIKE 'A%'
```

为表名指定分隔标识符时，必须分别分隔表名和架构名。因此，“ schema ”。“ tablename ”或schema。“ tablename ”是有效的标识符，但是“ schema.tablename ”不是有效的标识符。

禁用分隔标识符支持

默认情况下，启用对定界标识符的支持。

禁用分隔标识符支持时，双引号内的字符将被视为字符串文字。

可以使用带有SUPPORTDELIMITEDIDENTIFIERS关键字的SET OPTION命令在系统范围内设置分隔标识符支持。

可以使用%SYSTEM.SQL类的SetDelimitedIdentifiers（）方法在系统范围内设置分隔标识符支持。

若要确定当前设置，请调用$SYSTEM.SQL.CurrentSettings（）。

SQL保留字

SQL包含一长串不能用作标识符的保留字。

#Caché #InterSystems IRIS #InterSystems IRIS for Health

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E7%AB%A0-sql%E6%A0%87%E8%AF%86%E7%AC%A6