

文章

[姚鑫](#) · 三月 6, 2021 阅读大约需 16 分钟

第五章 SQL定义表（一）

第五章 SQL定义表

表名称和架构名称

可以通过定义表（使用CREATE TABLE）或通过定义投影到表的持久类来创建表：

- DDL：InterSystemsIRIS®数据平台使用CREATE TABLE中指定的表名来生成相应的持久类名，并使用指定的架构名来生成相应的包名。
- 类定义：InterSystemsIRIS®数据平台使用持久类名称来生成对应的表名，并使用包名称来生成对应的模式名。

由于以下原因，这两个名字之间的对应关系可能不相同：

- 持久化类和SQL表遵循不同的命名约定。
适用不同的有效字符和长度要求。
模式和表名不区分大小写；
包名和类名区分大小写。
系统自动将有效提供的名称转换为有效的对应名称，以确保生成的名称是惟一的。
- 持久化类名与对应的SQL表名之间的匹配是默认的。
可以使用SqlTableName类关键字来提供不同的SQL表名。
- **默认模式名可能与默认包名不匹配。**
如果指定一个非限定的SQL表名或持久类名，系统将提供一个默认的模式名或包名。
初始的默认模式名是SQLUser；
初始默认包名为“User”。

模式名称

表、视图或存储过程名称可以是限定的(schema.name)，也可以是限定的(name)。

- 如果指定模式名(限定名)，则指定的表、视图或存储过程将被分配给该模式。

如果模式不存在，则InterSystems SQL创建模式，并将表、视图或存储过程分配给它。

- 如果没有指定模式名(非限定名)，InterSystems SQL将使用默认模式名或模式搜索路径分配模式，如下所述。

模式命名注意事项

模式名遵循标识符约定，需要特别注意非字母数字字符的使用。

模式名不应该指定为带分隔符的标识符。

尝试指定“USER”或任何其他SQL保留字作为模式名会导致SQLCODE -312错误。

INFORMATIONSCHEMA模式名和相应的信息。

模式包名在所有命名空间中保留。

用户不应该在这个模式/包中创建表/类。

当执行一个创建操作(比如create TABLE)，指定一个还不存在的模式时，InterSystems IRIS将创建新的模式。

InterSystems IRIS使用模式名生成相应的包名。

由于模式及其对应包的命名约定不同，用户应该注意非字母数字字符的名称转换注意事项。

这些名称转换的注意事项与表不同:

- 初始字符:
 - % (percent):指定%作为模式名的第一个字符，表示相应的包为系统包，其所有类为系统类。这种用法需要适当的权限；否则，这种用法会发出一个SQLCODE -400错误，%msg表示<PROTECT>错误。
 - (下划线):如果模式名的第一个字符为下划线，则该字符将被对应包名中的小写“u”替换。
例如，模式名MySchema生成名为uMySchema的包。
- 后续的字符:
 - (下划线):如果模式名第一个字符以外的其他字符是下划线，则该字符将被对应包名中的句点(.)替换。
由于句点是类的分隔符，下划线将模式分为包和子包。
因此，MySchema生成包含包模式(My.Schema)的包My。
 - @, #, \$ characters:如果模式名包含任何这些字符，这些字符将从相应的包名中剥离。如果剥离这些字符会产生重复的包名，那么将进一步修改剥离的包名:将剥离的模式名的最后一个字符替换为顺序整数(以0开始)，以产生唯一的包名。
因此，My@#\$Schema生成MySchema包，然后创建My#\$Schema生成MySchema0包。
同样的规则也适用于表名对应的类名。

保留模式名

INFORMATIONSCHEMA模式名和相应的信息。

模式包名在所有命名空间中保留。

用户不应该在这个模式/包中创建表/类

在所有名称空间中保留IRISShard模式名。

用户不应在此模式中创建表、视图或过程。

存储在IRISShard模式中的项不会通过编目查询或INFORMATIONSCHEMA查询显示。

默认模式名称

- 在执行DDL操作(例如创建或删除表、视图、触发器或存储过程)时，会提供一个非限定名称作为默认的模式名。
架构搜索路径值将被忽略。
- 在执行DML操作时，例如通过选择、调用、插入、更新或删除访问现有表、视图或存储过程，将从模式搜索路径(如果提供了)提供一个不限定的名称。
如果没有架构搜索路径，或者没有使用架构搜索路径定位指定项，则提供默认的架构名称。

初始设置是对所有名称空间(系统范围)使用相同的默认模式名。

可以为所有命名空间设置相同的默认模式名，也可以为当前命名空间设置默认模式名。

如果创建了一个具有非限定名称的表或其他项，InterSystems IRIS将为其分配默认模式名和相应的持久类包名。

如果一个命名的或默认的模式不存在，InterSystems IRIS将创建模式(和包)，并将创建的项分配给该模式。

如果删除模式中的最后一项，InterSystems IRIS将删除该模式(和包)。

下面的模式名解析描述适用于表名、视图名和存储过程名。

系统范围的初始默认模式名是SQLUser。

对应的持久类包名是User。

因此，非限定表名Employee或限定表名SQLUser。

Employee将生成类User.Employee。

因为USER是一个保留字，尝试用USER的模式名(或任何SQL保留字)指定限定名会导致SQLCODE -1错误。

要返回当前默认模式名，请调用\$SYSTEM.SQL.DefaultSchema()方法:

第五章 SQL定义表（一）

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

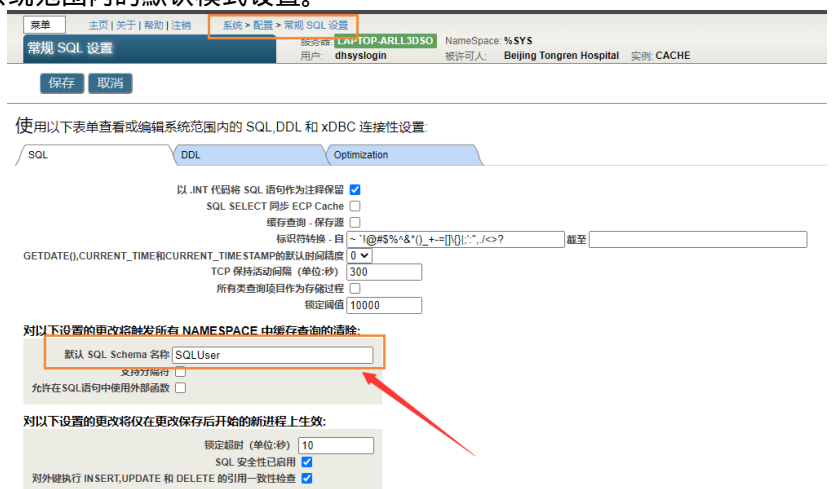
```
DHC-APP>WRITE $SYSTEM.SQL.DefaultSchema()  
SQLUser
```

或者使用以下预处理器宏:

```
#Include %occConstant  
WRITE $$$DefSchema
```

可以使用以下任意一种方式更改默认模式名:

- 进入管理界面。
在系统管理中，选择Configuration，然后选择SQL和对象设置，然后选择SQL。
在这个屏幕上，可以查看和编辑当前系统范围内的默认模式设置。
这个选项设置系统范围的默认模式名。



这个系统范围的设置可以被当前命名空间的SetDefaultSchema()方法值覆盖。

- \$SYSTEM.SQL.SetDefaultSchema()方法。默认情况下，此方法在系统范围内设置默认架构名称。但是，通过将布尔值第3个参数设置为1，可以仅为当前名称空间设置默认架构。当不同的名称空间具有不同的默认架构名称时，DefaultSchema()方法将返回当前名称空间的默认架构名称。

注意:当更改默认的SQL模式名称时，系统将自动清除系统上所有名称空间中的所有缓存查询。
通过更改默认模式名称，可以更改所有包含非限定表、视图或存储过程名称的查询的含义。
强烈建议在安装InterSystems IRIS时建立默认的SQL模式名，以后不要修改。

模式名用于生成相应的类包名。
因为这些名称有不同的命名约定，所以它们可能不相同。

可以通过将其设置为系统范围的默认模式来创建与SQL保留字同名的模式，但是不建议这样做。
名为User的默认模式根据类命名唯一性约定，生成相应的类包名称User0。

CURRENTUSER关键字

- 作为系统范围的默认模式名:如果指定CURRENTUSER作为默认模式名，InterSystems IRIS将指定当前登录进程的用户名作为默认模式名。
CURRENTUSER值是\$USERNAME ObjectScript特殊变量值的第一部分。
如果\$USERNAME包含一个名字和一个系统地址(Deborah@TestSys)，CURRENTUSER只包含名字片段;

这意味着CURRENTUSER可以将相同的默认模式名分配给多个用户。
如果进程没有登录，CURRENTUSER指定SQLUser作为默认的模式名。

如果指定CURRENTUSER/name作为默认模式名，其中name是选择的任意字符串，那么InterSystems IRIS将当前登录进程的用户名分配为默认模式名。

如果进程没有登录，则name将用作默认的模式名。

例如，如果进程没有登录，CURRENTUSER/HMO使用HMO作为默认模式名。

在\$SYSTEM.SQL.SetDefaultSchema()中，指定"CURRENTUSER"作为带引号的字符串。

- DDL命令中的模式名:如果在DDL语句中指定CURRENTUSER作为显式的模式名，InterSystems IRIS将其替换为当前系统范围内的默认模式。
例如，如果系统范围的默认模式是SQLUser，则命令DROP TABLE CURRENTUSER.OldTable SQLUser.OldTable下降。
这是一种方便的方式来限定名称，以显式地指示应该使用系统范围的默认模式。
它在功能上与指定非限定名相同。
此关键字不能在DML语句中使用。

模式搜索路径

当访问一个现有的表(或视图，或存储过程)进行DML操作时，将从模式搜索路径中提供一个非限定的名称。
按照指定的顺序搜索模式，并返回第一个匹配项。

如果在搜索路径中没有找到匹配的模式，或者没有搜索路径，则使用默认的模式名。

(注意，#Import宏指令使用了不同的搜索策略，不会“失败”到默认的模式名。)

- 在嵌入式SQL中，可以使用#SQLCompile Path宏指令或#Import宏指令来提供架构搜索路径，系统间IRIS使用该路径来解析非限定名称。
#SQLCompile Path根据遇到的第一个匹配项解析不限定的名称。
如果搜索路径中列出的所有模式只有一个匹配项，则#Import解析非限定名。
- 下面的示例提供了包含两个模式名的搜索路径:

```
#SQLCompile Path=Customers,Employees
```

- 在动态SQL中，可以使用%SchemaPath属性提供模式搜索路径，系统间IRIS使用该路径解析不限定的表名。
可以直接指定%SchemaPath属性，也可以将其指定为%SQL的第二个参数。
声明%new()方法。
下面的示例提供了包含两个模式名的搜索路径:

```
SET tStatement = ##class(%SQL.Statement).%New(0,"Customers,Employees")
```

- 在SQL Shell中，可以设置PATH SQL Shell配置参数来提供架构搜索路径，系统间IRIS使用该路径解析不限定的名称。

如果非限定名与模式搜索路径中指定的任何模式或默认模式名不匹配，则会发出SQLCODE

-30错误，例如:SQLCODE: -30消息:Table 'PEOPLE' not found in schemas:

CUSTOMERS,EMPLOYEES,SQLUSER.

包含特定于平台的模式名

当创建一个基于odbc的查询以通过Mac上的Microsoft query从Microsoft Excel运行时，如果从可用的表列表选择一个表，则生成的查询不包括该表的模式(相当于类的包)。

例如，如果选择从示例模式返回Person表的所有行，则生成的查询为:

```
SELECT * FROM Person
```

因为InterSystems

IRIS将不限定的表名解释为SQLUser模式中的表名，所以该语句要么失败，要么从错误的表返回数据。

要纠正这一点，编辑查询(在SQL View选项卡上)，显式引用所需的模式。

然后查询应该是：

```
SELECT * FROM Sample.Person
```

List模式

INFORMATION.SCHEMA。

SCHEMATA persistent类列出当前名称空间中的所有模式。

下面的示例返回当前命名空间中的所有非系统模式名：

```
SELECT SCHEMA_NAME  
FROM INFORMATION_SCHEMA.SCHEMATA WHERE NOT SCHEMA_NAME %STARTSWITH '%'
```

Management Portal SQL界面的左侧允许查看模式(或匹配筛选器模式的多个模式)的内容。

表名

每个表在其模式中都有一个唯一的名称。

一个表有一个SQL表名和一个对应的持久化类名；

这些名称在允许的字符、区分大小写和最大长度方面有所不同。

如果使用SQL CREATE TABLE命令定义，则指定遵循标识符约定的SQL表名；

系统生成一个对应的持久化类名。

如果定义为持久类定义，则必须指定只包含字母和数字字符的名称；

这个名称既用作区分大小写的持久类名，也用作(默认情况下)对应的不区分大小写的SQL表名。

可选的SqlTableName class关键字允许用户指定不同的SQL表名。

当使用CREATE TABLE命令创建表时，InterSystems IRIS使用表名生成相应的持久化类名。

由于表及其对应类的命名约定不同，用户应该注意非字母数字字符的名称转换：

- 初始字符：

- % (percent): %作为表名的第一个字符是保留的，应该避免(参见标识符)。

如果指定了，%字符将从对应的持久化类名中剥离。

- (下划线):如果表名的第一个字符是下划线，则该字符将从对应的持久化类名中剥离。

例如，表名MyTable生成类名MyTable。

- 数字:表名的第一个字符不能是数字。

如果表名的第一个字符是标点符号，则第二个字符不能是数字。

这将导致一个SQLCODE -400错误，%msg值为"error #5053:类名'schema.name' is invalid "(没有标点字符)。

例如，指定表名7A会生成%msg "ERROR #5053: Class name 'User.7A' is invalid "。

- 后续的字符：

- 字母:表名中至少包含一个字母。

表名的第一个字符或初始标点字符后的第一个字符必须是字母。

如果一个字符通过\$ZNAME测试，它就是一个有效的字母；

\$ZNAME字母验证因不同的地区而不同。

(注意，\$ZNAME不能用于验证SQL标识符，因为标识符可能包含标点字符。)

- (下划线)，@，#，\$ characters:如果表名包含这些字符中的任何一个，这些字符将从对应的类名中剥离出来，并生成一个唯一的持久类名。

由于生成的类名不包括标点字符，因此不建议创建仅在标点字符上不同的表名。

- 表名在其模式中必须是唯一的。

如果试图创建一个名称仅与现有表大小写不同的表，将会产生SQLCODE -201错误。

同一个模式中的视图和表不能具有相同的名称。

尝试这样做会导致SQLCODE -201错误。

可以使用\$SYSTEM.SQL.TableExists()方法确定一个表名是否已经存在。

可以使用\$SYSTEM.SQL.ViewExists()方法确定视图名是否已经存在。

这些方法还返回与表或视图名称对应的类名。

管理门户SQL interface Catalog Details表信息选项显示与所选SQL表名称对应的类名。

试图指定“USER”或任何其他SQL保留字作为表名或模式名会导致SQLCODE -312错误。

要指定SQL保留字作为表名或模式名，可以指定名称作为带分隔符的标识符。

如果使用带分隔符的标识符指定包含非字母数字字符的表或模式名，InterSystems

IRIS将在生成相应的类或包名时删除这些非字母数字字符。

适用以下表名长度限制：

- 唯一性:InterSystems IRIS对持久化类名的前189个字符执行唯一性检查。
对应的SQL表名可能超过189个字符，但是，当去掉非字母数字字符时，它必须在189个字符的限制内是唯一的。
InterSystems IRIS对包名的前189个字符执行唯一性检查。
- 建议最大长度:一般来说，一个表名不应该超过128个字符。
一个表名可能比96个字符长得多，但是在前96个字母数字字符中不同的表名更容易处理。
- 最大组合长度:包名和它的持久类名(加在一起时)不能超过220个字符。
这包括默认的模式(包)名(如果没有指定模式名)和分隔包名和类名的点字符。
当表名转换为对应的持久化类名时，删除超过220个字符时，模式和表名的组合长度可以超过220个字符。

RowID字段

在SQL中，每条记录都由一个唯一的整数值标识，这个整数值称为RowID。

在InterSystems SQL中，不需要指定RowID字段。

当创建表并指定所需的数据字段时，会自动创建RowID字段。

这个RowID在内部使用，但没有映射到类属性。

默认情况下，只有当持久化类被投影到SQL表时，它的存在才可见。

在这个投影表中，将出现一个额外的RowID字段。

默认情况下，这个字段被命名为“ID”，并分配给第1列。

默认情况下，当在表中填充数据时，InterSystems IRIS将从1开始向该字段分配连续的正整数。RowID数据类型为BIGINT（%Library.BigInt）。为RowID生成的值具有以下约束：每个值都是唯一的。不允许使用NULL值。排序规则是精确的。**默认情况下，值不可修改。**

默认情况下，InterSystems IRIS将此字段命名为“

ID”。但是，此字段名称不是保留的。每次编译表时都会重新建立RowID字段名。如果用户定义了一个名为“ID”的字段，则在编译表时，InterSystems IRIS会将RowID命名为“ID1”。例如，如果用户随后使用ALTER TABLE定义了一个名为“ID1”的字段，则表编译会将RowID重命名为“ID2”，依此类推。在持久性类定义中，可以使用SqlRowIdName类关键字直接为此类投影到的表指定RowID字段名。由于这些原因，应避免按名称引用RowID字段。

InterSystems

SQL提供了%ID伪列名称（别名），无论分配给RowID的字段名称如何，该伪列名称始终返回RowID值。

（InterSystems TSQL提供了\$IDENTITY伪列名称，其作用相同。）

ALTER TABLE无法修改或删除RowID字段定义。

将记录插入表中后，InterSystems IRIS将为每个记录分配一个整数ID值。

RowID值始终递增。它们不被重用。因此，如果已插入和删除记录，则RowID值将按升序排列，但可能不连续。

- 默认情况下，使用CREATE TABLE定义的表使用\$SEQUENCE执行ID分配，从而允许多个进程快速同时填充该表。当使用\$SEQUENCE填充表时，会将RowID值序列分配给进程，然后该进程将顺序分配它们。因为并发进程使用它们自己分配的序列分配RowID，所以不能假定多个进程插入的记录按插入顺序排列。

可以通过设置SetDDLUseSequence（）方法，将InterSystems IRIS配置为使用\$INCREMENT执行ID分配。若要确定当前设置，请调用\$SYSTEM.SQL.CurrentSettings（）方法。

- 默认情况下，通过创建持久性类定义的表将使用\$INCREMENT执行ID分配。在持久性类定义中，可以将IdFunction存储关键字设置为序列或增量；否则，可以设置为0。例如，<IdFunction>序列</IdFunction>。

在持久性类定义中，IdLocation存储关键字global（例如，对于持久性类Sample.Person：<IdLocation> ^Sample.PersonD </IdLocation>）包含RowID计数器的最高分配值。（这是分配给记录的最高整数，而不是分配给进程的最高整数。）请注意，此RowID计数器值可能不再与现有记录相对应。要确定是否存在具有特定RowID值的记录，请调用表的%ExistsId（）方法。

通过TRUNCATE TABLE命令重置RowID计数器。即使使用DELETE命令删除表中的所有行，也不会通过DELETE命令将其重置。如果没有数据插入表中，或者已使用TRUNCATE TABLE删除所有表数据，则IdLocation存储关键字全局值未定义。

默认情况下，RowID值不可用户修改。尝试修改RowID值会产生SQLCODE -107错误。覆盖此默认值以允许修改RowID值可能会导致严重的后果，只有在非常特殊的情况下并应格外谨慎。Config.SQL.AllowRowIDUpdate属性允许RowID值是用户可修改的。

基于字段的RowID

通过定义一个用于投影表的持久类，可以定义RowID以具有字段或字段组合中的值。为此，请使用IdKey index关键字指定一个索引。例如，一个表可以具有一个RowID，其RowID通过在PatientName [IdKey]上指定索引定义IdxId来与PatientName字段的值相同；或者可以通过指定索引定义IdxId来将PatientName和SSN字段的组合值在（PatientName，SSN）[IdKey]上。

- 基于字段的RowID效率比采用系统分配的连续正整数的RowID效率低。
- 在INSERT上：为构成RowID的字段或字段组合指定的值必须唯一。指定非唯一值将生成SQLCODE -119 “在插入时唯一性或主键约束唯一性检查失败”。
- 在UPDATE上：默认情况下，组成RowID的每个字段的值都是不可修改的。尝试修改这些字段之一的值会生成SQLCODE -107 “无法基于字段更新RowID或RowID”。

当RowID基于多个字段时，RowID值是由||连接的每个组成字段的值。操作员。例如，Ross，Betsy || 123-45-6789。InterSystems

IRIS尝试确定基于多个字段的RowID的最大长度。如果无法确定最大长度，则RowID长度默认为512。

隐藏的RowID？

- 使用CREATE TABLE创建表时，默认情况下隐藏RowID。SELECT *不会显示隐藏字段，而是PRIVATE。创建表时，可以指定%PUBLICROWID关键字以使RowID不隐藏和公开。可以在CREATE TABLE逗号分隔的表元素列表中的任何位置指定此可选的%PUBLICROWID关键字。不能在ALTER TABLE中指定。
- 创建作为表投影的持久类时，默认情况下不会隐藏RowID。它由SELECT *显示，并且是PUBLIC。可以通过指定类关键字SqlRowIdPrivate来定义具有隐藏且为PRIVATE的RowID的持久类。

用作外键引用的RowID必须是公共的。

默认情况下，不能将具有公共RowID的表用作源表或目标表，以使用INSERT INTO Sample.DupTable SELECT * FROM Sample.SrcTable将数据复制到重复表中。

可以使用Management Portal SQL界面“目录详细信息字段”列出“隐藏”列来显示RowID是否被隐藏。

可以使用以下程序返回指定字段（在此示例中为ID）是否被隐藏：

```
/// d ##class(PHA.TEST.SQL).RowID()  
ClassMethod RowID()  
{  
    SET myquery = "SELECT FIELD_NAME,HIDDEN FROM %Library.SQLCatalog_SQLFields(?) WHE  
RE FIELD_NAME='ID'"  
    SET tStatement = ##class(%SQL.Statement).%New()  
    SET qStatus = tStatement.%Prepare(myquery)  
    IF qStatus'=1 {  
        WRITE "%Prepare failed:"  
        DO $System.Status.DisplayError(qStatus)  
        QUIT  
    }  
    SET rset = tStatement.%Execute()  
    DO rset.%Display()  
    WRITE !,"End of data"  
}
```

[#Caché #InterSystems IRIS #InterSystems IRIS for Health](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E7%AB%A0-sql%E5%AE%9A%E4%B9%89%E8%A1%A8%EF%BC%88%E4%B8%80%EF%BC%89>