

## 文章

[Jingwei Wang](#) · 七月 21, 2022 阅读大约需 5 分钟

## InterSystems SQL 的使用 - 第四部分 - 视图

视图为存储查询，提供了物理表的所有灵活性和安全权限。所有的视图都是可更新的或只读的。

注意：不能对**只读**

的数据库中的数据创建视图。不能对存储在通过ODBC或JDBC网关连接的Informix表中的数据创建视图。这是因为InterSystems IRIS查询转换在FROM子句中使用子查询，而Informix不支持FROM子句的子查询。

### 创建视图

视图名称可以是合格的或不合格的。一个没有限定的视图名称是一个简单的标识符。MyView。一个合格的视图名称由两个简单的标识符组成，一个schema名称和一个视图名称，用句号隔开，例如MySchema.MyView。视图名和表名遵循相同的命名规则，并对未限定的名称执行相同的schema名称解析。同一模式中的视图和表不能有相同的名称。

你可以通过几种方式定义视图：

0. 使用SQL CREATE VIEW命令（在DDL脚本中或通过JDBC或ODBC）。

```
CREATE VIEW MySchema.MyView (ViewCol1, ViewCol2, ViewCol3) AS
```

```
SELECT TableCol1, TableCol2, TableCol3
```

```
FROM MyTable
```

2. 使用管理门户的创建视图界面。 系统操作 -> SQL -> 操作 -> 创建视图



3. 这将显示 "创建视图" 窗口，如图所示：

创建视图

☒ 选择现有 schema ☐ 输入新 schema

Schema:

视图名称  必填

带检查选项  ☐ 将此视图上的所有特权授予 \_PUBLIC

查看文本 必填

**schema**：您可以决定将视图包含在现有schema中，或者创建一个新的schema。

**视图名称**：一个有效的视图名称。在同一模式中，你不能对表和视图使用相同的名称。

**带检查选项**：选项有READONLY、LOCAL、CASCADED。

#### 授予视图的所有权限给PUBLIC

：如果选择了这个选项，这个选项会给所有用户执行这个视图的权限。默认是不给所有用户访问该视图的权限。

**查看文本**：你可以通过以下三种方式中的任何一种指定视图文本。在 "查看文本

"区域键入一个SELECT语句。

使用查询生成器创建一个SELECT语句，然后按确定将此查询提供给视图文本区。如果你在管理门户SQL界面左侧选择一个缓存查询名称（例如%sqlcq.USER.cls4），然后调用创建视图，这个缓存查询就会提供给视图文本区。注意，在视图文本区，你必须在保存视图文本之前用实际值替换变量引用（问号）。

#### 4. 使用Objectscript执行DDL

```
ClassMethod CreateTable() As %String
```

```
{
```

```
&sql(CREATE VIEW Sample.VSrStaff
```

```
AS SELECT Name AS Vname, Age AS Vage
```

```
FROM Sample.Person WHERE Age>75)
```

```
IF SQLCODE=0{ WRITE "Created a view"}
```

```
ELSEIF SQLCODE=-201 { WRITE "View already exists" RETURN SQLCODE}
```

```
ELSE { WRITE "Serious SQL Error, returning SQLCODE " RETURN SQLCODE_" " _%msg}
```

```
}
```

**\$SYSTEM.SQL.Schema.ViewExists()**方法可以用来确定一个视图名称是否已经存在。

## 查看视图定义

### 0. 视图信息

```
SELECT * FROM INFORMATION_SCHEMA.VIEWS
```

### 1. 视图的依赖表

#### **INFORMATION\_SCHEMA.VIEWTABLEUSAGE**

持久化类显示当前命名空间中的所有视图和它们所依赖的表。

- 通过SQL脚本显示视图依赖表

```
SELECT View_Schema, View_Name, Table_Schema, Table_Name FROM INFORMATION_SCHEMA.VIEW_TABLE_USAGE
```

- 通过ObjectScript显示依赖表

```
SET statemt = ##class(%SQL.Statement).%New()
```

```
SET cgStatus = statemt.%PrepareClassQuery("%Library.SQLCatalog", "SQLViewDependsOn")
```

```
IF cgStatus '= 1 { WRITE "PrepareClassQuery failed:" DO $System.Status.DisplayError(cgStatus) QUIT }
```

```
SET rset = statemt.%Execute("vschema.vname")
```

```
DO rset.%Display()
```

如果调用者没有视图所依赖的表的权限，该表及其模式将被列为NOT PRIVILEGED。这允许没有表权限的调用者确定视图所依赖的表的数量，但不能确定表的名称。

## View ID: %VID

InterSystems IRIS为视图或FROM子句返回的每条记录分配一个整数的视图ID (%VID)。与表的行ID号一样，这些视图的行ID号也是系统分配的、唯一的、非空的、非零的和不可修改的。这个%VID通常对用户来说是不可见的，只有在明确指定时才会返回。它作为数据类型INTEGER返回。因为%VID值是连续的整数，如果视图返回有序的数据，它们就更有意义；视图只有在与TOP子句配对时才能使用ORDER BY子句。

```
CREATE VIEW Sample.VSrStaff
```

```
AS SELECT TOP ALL Name AS Vname, Age AS Vage
```

```
FROM Sample.Person WHERE Age>75
```

```
ORDER BY Name
```

下面的例子返回VSrStaff视图定义的所有数据（使用SELECT \*），并且还指定要返回每一行的视图ID。与表的行ID不同，视图的行ID在使用星号语法时不会显示；只有在SELECT中明确指定时才会显示。

```
SELECT *,%VID AS ViewID FROM Sample.VSrStaff
```

%VID可以用来进一步限制SELECT从视图返回的行数，如下面的例子所示。

```
SELECT *,%VID AS ViewID FROM Sample.VSrStaff WHERE %VID BETWEEN 5 AND 10
```

因此，%VID可以用来代替TOP（或者作为TOP的补充）来限制查询返回的记录的数量。一般来说，TOP子句用于返回数据记录的一个小子集；%VID用于返回大部分或全部数据记录，将记录返回到小子集的中。这个功能可能很有用，特别是对于Oracle查询（%VID很容易映射到Oracle ROWNUM）。然而，用户应该意识到，与TOP相比，使用%VID有一些性能限制。

- %VID不执行time-to-first-row的优化。TOP的优化是尽可能快地返回第一行数据。%VID的优化是为了尽可能快地返回完整的数据集。
- 如果查询指定了排序的结果，%VID不会执行有限的排序（这是由TOP执行的特殊优化）。查询首先对完整的数据集进行排序，然后使用%VID对返回的数据集进行限制。TOP是在排序前应用的，所以SELECT执行的是有限的排序，只涉及一个受限制的行子集。

为了保持对第一行的优化时间和有限的排序优化，你可以使用一个带有TOP和%VID组合的FROM子句子查询。

```
SELECT *,%VID AS SubQueryID
```

```
FROM (SELECT TOP 10 Name, Age
```

```
FROM Sample.Person
```

```
WHERE Age > 75
```

```
ORDER BY Name)
```

```
WHERE %VID > 4
```

**不能对指定了%VID的查询进行并行执行，即使明确指定了%PARALLEL关键字。**

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

---

### 源

**URL:**<https://cn.community.intersystems.com/post/intersystems-sql-%E7%9A%84%E4%BD%BF%E7%94%A8-%E7%AC%AC%E5%9B%9B%E9%83%A8%E5%88%86-%E8%A7%86%E5%9B%BE>