

文章

[姚鑫](#) · 九月 23, 2021 阅读大约需 9 分钟

第二十四章 SQL命令 CREATE VIEW (一)

第二十四章 SQL命令 CREATE VIEW (一)

创建视图

大纲

```
CREATE [OR REPLACE] VIEW view-name [(column-commalist)]
AS select-statement
[ WITH READ ONLY | WITH [level] CHECK OPTION ]
```

参数

- view-name - 正在创建的视图的名称。有效的标识符，受与表名相同的附加命名限制。视图名称可以是限定的(schema.viewname)，也可以是不限定的(Viewname)。未限定的视图名称采用默认架构名称。请注意，同一架构中的表和视图不能使用相同的名称。
- column-commalist - 可选-组成视图的列名、一个或多个有效标识符。如果指定，此列表括在圆括号中，列表中的项目用逗号分隔。
- AS select-statement - 定义视图的SELECT语句。
- WITH READ ONLY - 可选-指定不能通过此视图对视图所基于的表执行插入、更新或删除操作。默认情况下，允许通过视图执行这些操作，但要遵守下面描述的约束条件。
- WITH level CHECK OPTION - 可选-指定如何通过此视图对视图所基于的表执行插入、更新或删除操作。级别可以是关键字LOCAL或CASCADED。如果未指定级别，则WITH CHECK选项默认为级联。

描述

CREATE VIEW命令定义视图的内容。定义视图的SELECT语句可以引用多个表，也可以引用其他视图。

权限

CREATE VIEW命令是特权操作。用户必须具有%CREATEVIEW管理权限才能执行CREATE VIEW。否则将导致%msg用户 'name' 没有%CREATEVIEW权限的SQLCODE-99错误。如果拥有适当的授予权限，则可以使用GRANT命令分配%CREATEVIEW权限。

要从正在创建的视图的SELECT子句中引用的对象中进行选择，需要具有适当的权限：

- 使用动态SQL或xDBC创建视图时，必须对从视图引用的基础表(或视图)中选择的所有列具有SELECT权限。如果对指定表(或视图)没有SELECT权限，则不会执行CREATE VIEW命令。

但是，在编译投影已定义视图的类时，不会对从视图引用的基础表(或视图)中选择的列强制执行这些SELECT特权。例如，如果使用特权例程(具有这些SELECT权限)创建视图，则可以在以后编译视图类，因为是视图的所有者，而不管是否对视图引用的表具有SELECT权限。

- 若要获得视图的SELECT特权WITH GRANT OPTION，则必须对该视图引用的每个表(或视图)都具有WITH GRANT OPTION。

- 若要接收视图的INSERT、UPDATE、DELETE或REFERENCES权限，必须对该视图引用的每个表(或视图)具有相同的权限。要接收其中任何权限的WITH GRANT OPTION，必须在基础表上拥有PRIVICATION WITH GRANT OPTION。
- 如果该视图指定为只读，则不会授予该视图INSERT、UPDATE或DELETE权限，无论您对基础表拥有哪些权限。如果稍后将视图重新定义为读/写，则在重新编译投影视图的类时会添加这些权限。

可以通过调用%CHECKPRIV命令来确定当前用户是否拥有这些表级权限。可以通过调用\$SYSTEM.SQL.Security.C heckPrivilege()方法来确定指定用户是否拥有这些表级权限。

编译视图时，视图的创建者(所有者)被授予%ALTER PRIVATION WITH GRANT选项。

在嵌入式SQL中，可以使用\$SYSTEM.Security.Login()方法以具有适当权限的用户身份登录：

```
DO $SYSTEM.Security.Login( "_SYSTEM", "SYS" )
&sql(      )
```

必须具有%ServiceLogin : Use权限才能调用\$SYSTEM.Security.Login方法。

%CREATEVIEW权限是通过GRANT命令分配的，这需要将此权限分配给用户或角色。

默认情况下，强制执行CREATE VIEW安全特权。

这个权限要求是可在系统范围内配置的，使用\$SYSTEM.SQL.Util.SetOption()方法SET status=\$SYSTEM.SQL.Util.SetOption("SQLSecurity", 0, .oldval);

要确定当前设置，调用\$SYSTEM.SQL.CurrentSettings()方法，该方法显示启用SQL安全的设置。

默认值是1(启用)。

当SQL Security被启用时，用户只能对已被授予权限的表或视图执行操作。

这是该选项的推荐设置。

如果此方法设置为0，则更改此设置后启动的任何新进程将禁用SQL Security。

这意味着禁止基于特权的表/视图安全性。

可以在不指定用户的情况下创建表。

在本例中，动态SQL将“_SYSTEM”指定为用户，嵌入式SQL将“ ”(空字符串)指定为用户。

任何用户都可以对表或视图执行操作，即使该用户没有这样做的特权。

视图命名约定

视图名称与表名具有相同的命名约定，并且共享相同的名称集。因此，不能对同一架构中的表和视图使用相同的名称。尝试这样做会导致SQLCODE-201错误。若要确定当前命名空间中是否已存在表，请使用\$SYSTEM.SQL.Schema.TableExists(“ schema.tname ”)方法。投射同名的表定义和视图定义的类也会生成SQLCODE-201错误。

视图名称遵循标识符约定，并受以下限制的约束。默认情况下，视图名称是简单标识符。视图名称不应超过128个字符。视图名称不区分大小写。

IRIS使用视图名称生成相应的类名。类名仅包含字母数字字符(字母和数字)，并且在前96个字符内必须是唯一的。要生成这个类名，IRIS首先从视图名称中去掉标点符号，然后生成一个在前96个字符内唯一的标识符，在需要创建唯一的类名时用一个整数(从0开始)代替最后一个字符。

IRIS从有效的视图名称生成唯一的类名，但此名称生成对视图的命名施加了以下限制：

- 视图名称必须至少包含一个字母。视图名称的第一个字符或首个标点符号字符后的第一个字符必须是字母。
- IRIS支持视图名称使用16位(宽)字符。如果字符通过\$ZNAME测试，则该字符是有效字母。
- 如果视图名称的第一个字符是标点符号，则第二个字符不能是数字。这会导致SQLCODE-400错误，%msg值为Error#5053：类名‘ schema.name ’无效(没有标点符号)。例如，指定视图名称%7A会生成%msg错误#5053：类名‘ User.7 A ’无效。
- 由于生成的类名不包括标点符号，因此不建议(尽管可能)创建仅在标点符号方面与现有视图或表名不同的视图名称。在这种情况下，IRIS用一个整数(从0开始)代替名称的最后一个字符，以创建唯一的类名。
- 视图名称可能比96个字符长得多，但前96个字母数字字符不同的视图名称更易于使用。

视图名称可以是限定的，也可以是非限定的。

限定的视图名称(schema.viewname)可以指定现有架构或新架构。如果指定了新架构，系统将创建该架构。

未限定的视图名称(视图名称)采用默认架构名称。

Existing View

要确定指定的视图是否已经存在于当前命名空间中，请使用\$SYSTEM.SQL.Schema.ViewExists("schema.vname")方法。

当创建与现有视图同名的视图时，会发生什么取决于可选的OR REPLACE关键字和配置设置。

With OR REPLACE

如果指定CREATE OR REPLACE VIEW，则现有视图将被SELECT子句中指定的视图定义和任何指定的WITH READ ONLY或WITH CHECK OPTION替换。

这与执行相应的ALTER VIEW语句相同。

已授予原始视图的任何特权仍然保留。

这个关键字短语没有提供ALTER VIEW不可用的功能。

它是为兼容Oracle SQL代码而提供的。

Without OR REPLACE

默认情况下，如果指定CREATE VIEW，IRIS将拒绝使用现有视图的名称创建视图的尝试，并发出SQLCODE -201错误。

要确定当前设置，调用\$SYSTEM.SQL.CurrentSettings()，它为现有的表或视图设置显示一个Allow DDL CREATE TABLE或CREATE VIEW。

默认值是0 (No)，这是推荐的设置。

如果此选项设置为1 (Yes)，IRIS将删除与视图关联的类定义，然后重新创建它。

这与先执行DROP VIEW，然后执行CREATE VIEW非常相似。

注意，此设置同时影响CREATE VIEW和CREATE TABLE。

在管理门户、系统管理、配置、SQL和对象设置、SQL中，可以通过选择“忽略冗余DDL语句”复选框，在系统范围内设置此选项(以及其他类似的创建、更改和删除选项)。

列名

视图可以有选择地包括用括号括起来的列名的列分隔符列表。

这些列名(如果指定的话)是在使用该视图时用于访问和显示列的数据的名称。

如果省略了列逗号，下面的应用程序:

- 选择源表的列名用于在使用视图时访问和显示数据。
- 如果任何选择源表列名具有列别名，则列别名是使用视图时用于访问和显示数据的名称。
- 如果选择源表列名具有表别名，则在使用视图时用于访问和显示数据的名称中不会使用表别名。

如果省略列名列表，则还必须省略圆括号。

如果指定列COMMANCEL，则以下情况适用：

- 列名列表必须指定外围括号，即使指定单个字段也是如此。多个列名之间必须用逗号分隔。在列注释器中允许有空格和注释。
- 列名的数量必须与SELECT语句中指定的列数相对应。视图列数和查询列数之间的不匹配导致编译时出现SQ

LCODE-142错误。

- 列名的名称必须是有效的标识符。它们可以是与选择列名不同的名称、与选择列名相同的名称或两者的组合。视图列名的指定顺序与选择列名的顺序相对应。由于可以为视图列分配不相关的选择列的名称，因此在分配视图列名称时必须格外小心。
- 列名必须是唯一的。指定重复的列名会导致SQLCODE-97错误。列名通过去除标点符号转换为相应的类属性名；允许使用仅在标点符号方面不同的列名，但不鼓励这样做。

下面的示例显示了一个包含视图列和查询列匹配列表的创建视图：

```
CREATE VIEW MyView (ViewCol1, ViewCol2, ViewCol3) AS
  SELECT TableCol1, TableCol2, TableCol3
  FROM MyTable
```

或者，也可以在查询中使用AS关键字将视图列指定为查询列/视图列对，如下例所示：

```
CREATE VIEW MyView AS
  SELECT TableCol1 AS ViewCol1,
         TableCol2 AS ViewCol2,
         TableCol3 AS ViewCol3
  FROM MyTable
```

选择列和查看列

可以将多个选择列中的数据连接到单个视图列中。例如：

```
CREATE VIEW MyView (fullname) AS SELECT firstname||' '||lastname FROM MyTable
```

多个视图列可以引用同一选择列。例如：

```
CREATE VIEW MyView (lname,surname) AS SELECT lastname,lastname FROM MyTable
```

SELECT子句注意事项

视图不必是一个特定表的行和列的简单子集。

可以使用任意复杂度的SELECT子句创建视图，指定表或视图的任意组合。

然而，对于视图定义的SELECT子句有一些限制：

- 只有当ORDER BY子句与TOP子句成对出现时，才能包括此子句。如果希望包括视图中的所有行，可以使用TOP ALL子句。可以包含不带ORDER BY子句的TOP子句。但是，如果包含没有TOP子句的ORDER BY子句，则会生成SQLCODE-143错误。如果从视图类投影SQL视图，而视图类的查询包含ORDER BY子句，则在视图投影中将忽略ORDER BY子句。
- 不能包含主机变量。如果尝试在SELECT子句中引用主机变量，系统将生成SQLCODE-148错误。
- 不能包含INTO关键字。可以创建使用INTO子句指定SELECT的视图，但该视图的执行失败，并出现SQLCODE-25错误。

CREATE VIEW可以包含UNION语句，以从两个表的联合中选择列。可以指定UNION，如下面的嵌入式SQL示例所示：

```
ClassMethod CreateView()  
{  
    d $SYSTEM.Security.Login("_SYSTEM","SYS")  
    &sql(  
        CREATE VIEW MyView (vname,vstate) AS  
            SELECT t1.name,t1.home_state  
            FROM Sample.Person AS t1  
            UNION  
            SELECT t2.name,t2.office_state  
            FROM Sample.Employee AS t2  
    )  
    if SQLCODE = 0 {  
        w !,"?????"  
    } else {  
        w "?????? SQLCODE=",SQLCODE  
    }  
}
```

请注意，非限定视图名称(如上例中)默认为默认模式名称(例如，初始模式默认SQLUser.MyView)，即使视图引用的表在示例模式中也是如此。因此，通常最好总是限定视图名称，以确保它与其关联表一起存储。

View ID: %vid

通过视图访问数据时，IRIS会为该视图返回的每一行分配一个连续的整数视图ID(%VID)。与表行ID编号一样，这些视图行ID编号是系统分配的、唯一的、非零的、非空的和不可修改的。此%VID通常是不可见的。与表行ID不同，它在使用星号语法时不会显示；只有在SELECT中显式指定时才会显示。%vid可用于进一步限制SELECT访问视图返回的行数

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E5%8D%81%E5%9B%9B%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-view%E5%BC%88%E4%B8%80%E5%BC%89>