

文章

姚鑫 · 九月 18, 2021 阅读大约需 4 分钟

第十九章 SQL命令 CREATE TABLE（六）

第十九章 SQL命令 CREATE TABLE（六）

WITH子句，%CLASSPARAMETER关键字，STORAGETYPE关键字

可选的WITH子句可以在表格元素逗号结尾的圆括号之后和Shard Key定义(如果存在的话)之后指定。

WITH子句可以包含一个用逗号分隔的列表:

- 一个或多个%CLASSPARAMETER 子句。
- STORAGETYPE子句

%CLASSPARAMETER子句

在WITH关键字之后，可以指定多个%CLASSPARAMETER关键字子句，每个子句定义一个类参数。

多个%CLASSPARAMETER子句子句之间用逗号分隔。

为了向后兼容，支持将%CLASSPARAMETER关键字子句指定为table-element-commalist中的元素。

在两个位置中指定相同的%CLASSPARAMETER关键字子句将产生SQLCODE -327错误。

%CLASSPARAMETER关键字后面跟着类参数名称、一个可选的等号 and 要分配给该类参数的文字值(字符串或数字)

。类参数总是定义为常数值。

因为用户可以用任何名称或值定义额外的类参数，所以只执行语法验证;

既不验证类参数是否存在，也不验证类参数的有效值。

下面的示例定义了两个类参数;

第一个%CLASSPARAMETER子句使用了等号，第二个省略了等号:

```
CREATE TABLE OurEmployees (  
    EMPNUM      INT NOT NULL,  
    NAMELAST    CHAR(30) NOT NULL,  
    NAMEFIRST   CHAR(30) NOT NULL,  
    CONSTRAINT EMPLOYEEPK PRIMARY KEY (EMPNUM)  
)  
WITH %CLASSPARAMETER DEFAULTGLOBAL = '^GL.EMPLOYEE',  
    %CLASSPARAMETER MANAGEDEXTENT 0
```

DEFAULTGLOBAL:默认情况下，CREATE

TABLE用生成的全局名称为创建的表创建IDKEY索引，例如^EPgS.D8T6.1;

其他索引使用生成的具有唯一整数后缀的相同全局名称。这个例子指定%CLASSPARAMETER DEFAULTGLOBAL = '^GL.EMPLOYEE'

作为索引的显式全局名称。可以使用DEFAULTGLOBAL指定扩展的全局引用，或者完整引用(%CLASSPARAMETER DEFAULTGLOBAL = '^|"USER"|GL.EMPLOYEE')或者只是命名空间部分 (%CLASSPARAMETER DEFAULTGLOBAL = '^|"USER"|')。

当前使用的类参数有ALLOWIDENTITYINSERT, DATALOCATIONGLOBAL, DEFAULTGLOBAL, DSINTERVAL, DSTIME, EXTENTQUERYSPEC, EXTENTSIZE, GUIDENABLED, MANAGEDEXTENT, READONLY, ROWLEVELSECURITY, SQLPREVENTFULLSCAN, USEEXTENTSET, VERSIONCLIENTNAME,

VERSIONPROPERTY。

可以使用USEEXTENTSET和DEFAULTGLOBAL类参数定义表数据存储和索引数据存储的全局命名策略。

IDENTIFIEDBY类参数已弃用。

必须将IDENTIFIEDBY关系转换为 IRIS中支持的正确的父/子关系。

定义分片表的CREATE

TABLE不能定义DEFAULTGLOBAL、DSINTERVAL、DSTIME或VERSIONPROPERTY类参数。

STORAGETYPE子句

在WITH关键字之后，可以指定一个STORAGETYPE子句，STORAGETYPE=ROW或STORAGETYPE=COLUMN。该表选项用于设置STORAGEDEFAULT参数。

如果指定ROW，则PARAMETER STORAGEDEFAULT;

将出现在类定义中。将出现在类定义中。

如果指定COLUMN，则PARAMETER STORAGEDEFAULT = "column";

将出现在类定义中。

如果多次指定STORAGETYPE，则生成SQLCODE -327错误。

示例:动态SQL和嵌入式SQL

下面的示例演示了使用动态SQL和嵌入式SQL创建表。

注意，在动态SQL中，可以在同一个程序中创建一个表并将数据插入到表中;

在嵌入式SQL中，必须使用单独的程序来创建表并将数据插入到表中。

最后一个程序示例删除表，以便可以重复运行这些示例。

下面的动态SQL示例创建表SQLUser.MyStudents。

注意，因为COMPUTECODE是ObjectScript代码，而不是SQL代码，ObjectScript \$PIECE函数使用双引号分隔符;因为这行代码本身是一个带引号的字符串，\$PIECE分隔符必须通过加倍的方式转义为字面量，如下所示:

```
ClassMethod CreateTable7()
{
    s stuDDL=5
    s stuDDL(1)="CREATE TABLE SQLUser.MyStudents ("
    s stuDDL(2)="StudentName VARCHAR(32),StudentDOB DATE,"
    s stuDDL(3)="StudentAge INTEGER COMPUTECODE {SET {StudentAge}="
    s stuDDL(4)=" $PIECE(($PIECE($H,"",""),1)-{StudentDOB})/365,""."",1)} CALCULATED,"
    s stuDDL(5)="Q1Grade CHAR,Q2Grade CHAR,Q3Grade CHAR,FinalGrade VARCHAR(2))"
    s tStatement = ##class(%SQL.Statement).%New(0,"Sample")
    s qStatus = tStatement.%Prepare(.stuDDL)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s rtn = tStatement.%Execute()
    if rtn.%SQLCODE = 0 {
        w !,"?????"
    } elseif rtn.%SQLCODE=-201 {
        w "?????SQLCODE=",rtn.%SQLCODE,!
    } else {
        w !,"?????SQLCODE=",rtn.%SQLCODE,!
```

第十九章 SQL命令 CREATE TABLE (六)

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

```
        w rtn.%Message, !
    }
}
```

以下嵌入式 SQL 示例创建表 SQLUser.MyStudents：

```

ClassMethod CreateTable8()
{
    &sql(CREATE TABLE SQLUser.MyStudents
        (
            StudentName VARCHAR(32), StudentDOB DATE,
            StudentAge INTEGER COMPUTECODE {
                SET {StudentAge}=
                $PIECE(($PIECE($H, "", 1) - {StudentDOB})/365, ".", 1)
            } CALCULATED,
            Q1Grade CHAR, Q2Grade CHAR, Q3Grade CHAR, FinalGrade VARCHAR(2)
        )
    )
    if SQLCODE=0 {
        WRITE !, "Created table"
    } ELSEIF SQLCODE=-201 {
        WRITE !, "SQLCODE=", SQLCODE, " ", %msg
    } ELSE {
        WRITE !, "CREATE TABLE failed, SQLCODE=", SQLCODE
    }
}

```

以下示例删除由前面的示例创建的表：

```
ClassMethod CreateTable9()  
{  
    &sql(  
        DROP TABLE SQLUser.MyStudents  
    )  
    if SQLCODE=0 {  
        w !, "?????"  
    } else {  
        w !, "SQLCODE=", SQLCODE, " ", %msg  
    }  
}
```

#SQL #Caché

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%B9%9D%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-create-table%EF%BC%88%E5%85%AD%EF%BC%89>