

文章

[姚鑫](#) · 十月 22, 2021 阅读大约需 6 分钟

第五十三章 SQL命令 INSERT (二)

第五十三章 SQL命令 INSERT (二)

流数据

可以将以下类型的数据值插入到流字段中：

- 对于任何表：字符串文字或包含字符串文字的主机变量，例如：

```
set literal="Technique 1"

//do the insert; use a string
&sql(INSERT INTO MyStreamTable (MyStreamField) VALUES (:literal))
```

- 对于非切片表：对流对象的对象引用(OREF)。IRIS打开此对象并将其内容复制到新的流字段中。例如：

```
set oref=##class(%Stream.GlobalCharacter).%New()
do oref.Write("Technique non-shard 1")

//do the insert; use an actual OREF
&sql(INSERT INTO MyStreamTable (MyStreamField) VALUES (:oref))
```

或流的OREF的字符串版本，例如：

```
set oref=##class(%Stream.GlobalCharacter).%New()
do oref.Write("Technique non-shard 2")

//next line converts OREF to a string OREF
set string=oref_" "

//do the insert
&sql(INSERT INTO MyStreamTable (MyStreamField) VALUES (:string))
```

- 对于切片表：使用存储在^IRIS.Stream.Shard全局中的临时流对象的对象ID(OID)：

```
SET clob=##class(%Stream.GlobalCharacter).%New("Shard")
DO clob.Write("Technique Sharded Table 1")
SET sc=clob.%Save() // Handle $$$ISERR(sc)
    set ClobOid=clob.%Oid()
//do the insert
&sql(INSERT INTO MyStreamTable (MyStreamField) VALUES (:ClobOid))
```

尝试插入定义不正确的流值会导致SQLCODE-412错误：常规流错误。

List 结构化数据

IRIS支持列表结构数据类型%List(数据类型类%Library.List)。这是一种压缩的二进制格式，不会映射到 SQL的相应本机数据类型。它对应于默认MAXLEN为32749的数据类型VARBINARY。因此，动态SQL不能使用INSERT或UPDATE来设置%LIST类型的属性值。

插入计数器值

表可以有选择地将一个字段定义为Identity。默认情况下，每当向表中插入行时，此字段都会从自动递增的表计数器接收整数。默认情况下，插入不能为此字段指定值。但是，此默认值是可配置的。更新操作不能修改身份字段值。此计数器由截断表操作重置。

表可以选择将一个字段定义为数据类型ROWVERSION。如果定义了该字段，插入操作会自动将命名空间范围的Row Version计数器中的整数插入到该字段中。更新操作使用当前命名空间范围的Row Version计数器值自动更新此整数。不能为ROWVERSION字段插入用户指定的值、计算的值或默认值。此计数器无法重置。

表可以有选择地将一个或多个字段定义为数据类型SERIAL(%Library.Counter)。默认情况下，每当向表中插入行时，此字段都会从自动递增的表计数器接收整数。但是，用户可以在插入期间为该字段指定整数值，覆盖表计数默认值。更新操作不能修改序列(%COUNTER)字段值。此计数器由截断表操作重置。

插入序列值SERIAL Values

插入操作可以为具有串行数据类型的字段指定下列值之一，结果如下：

- 无值、0(零)或非数字值：
IRIS忽略指定值，改为将此字段的当前串行计数器值递增1，并将生成的整数插入到该字段中。
- 正整数：IRIS将用户指定的值插入该字段，并将该字段的串行计数器值更改为该整数值。

因此，串行字段包含一系列递增的整数值。这些值不一定是连续的或唯一的。例如，以下是序列字段的一系列有效值：1、2、3、17、18、25、25、26、27。顺序整数是IRIS生成的或用户提供的；非顺序整数是用户提供的。如果希望序列字段值是唯一的，则必须对该字段应用唯一约束。

插入计算值

使用COMPUTECODE定义的字段可以在INSERT操作中插入值，除非对该字段进行了计算。如果为计算字段提供值，或者如果此字段具有默认值，则INSERT将存储此显式值。否则，将计算该字段值，如下所示：

- COMPUTECODE：值在插入时计算并存储，值在更新时不变。
- COMPUTECODE WITH COMPUTEONCHANGE：VALUE在INSERT时计算并存储，在UPDATE时重新计算并存储。
- COMPUTECODE WITH DEFAULT和COMPUTEONCHANGE：默认值在插入时存储，值在更新时计算和存储。
- COMPUTECODE WITH COMPUTECODE WITH COMPUTED或瞬态：不能为此字段插入值，因为没有存储值。查询时会计算该值。但是，作为插入操作的一部分，IRIS确实会对此字段执行验证：
 - 如果尝试在计算字段中插入值，IRIS将对提供的值执行验证，如果值无效则发出错误。如果值有效，IRIS将继续行插入：它不会在此字段中插入值，不会发出SQLCODE错误，并且会递增ROWCOUNT。
 - 如果此类型的字段是外键约束的一部分，则会在插入期间计算此字段的值，以便执行引用完整性检查；不会存储此计算值。

如果计算代码包含编程错误(例如，除以零)，则插入操作将失败，并显示SQLCODE-415错误。

默认值子句

可以将行插入到其所有字段值都设置为默认值的表中。定义了默认值的字段将设置为该值。未定义默认值的字段设置为NULL。这可以使用以下命令来完成：

INSERT INTO Mytable DEFAULT VALUES

使用NOT NULL约束定义的字段和未定义的默认值使用SQLCODE-108使此操作失败。

可以使用此语句插入使用UNIQUE约束定义的字段。如果字段定义了唯一约束且没有默认值，则重复调用会插入多行，并将此唯一字段设置为NULL。如果使用唯一约束和默认值定义字段，则此语句只能使用一次。第二次调用失败，返回SQLCODE-119。

默认值为计数器字段插入具有系统生成的整数值。这些字段包括RowID、可选的标识字段、序列号(%Counter)字段和ROWVERSION字段。

插入查询结果：使用SELECT插入

通过将单个INSERT与SELECT语句结合使用，可以使用单个INSERT将多行插入到表中。可以使用任何有效的SELECT查询。SELECT从一个或多个表中提取列数据，而INSERT在其表中创建包含该列数据的相应新行。对应的字段可以具有不同的列名和列长度，只要插入的数据适合插入表字段即可。相应的字段必须通过数据类型和长度验证；否则将生成SQLCODE-104错误。

可以通过在SELECT语句中指定TOP子句来限制插入的行数。还可以在SELECT语句中使用ORDER BY子句来确定TOP子句将选择哪些行。

可以使用GROUP BY子句仅插入一个(或多个)字段的唯一值。由于默认情况下，GROUP BY会将值转换为大写，以便进行分组，因此可能需要使用%Exact排序规则来保留插入值的字母大小写。下面的示例显示了这一点：

```
INSERT INTO Sample.UniquePeople (Name, Age) SELECT Name, Age FROM Sample.Person WHERE Name IS NOT NULL GROUP BY %EXACT Name
```

INSERT WITH SELECT操作将%ROWCOUNT变量设置为插入的行数(0或正整数)。

以下示例使用具有两个嵌入式SQL语句的例程。Create table创建一个新表SQLUser.MyStudents，然后INSERT用从Sample.Person提取的数据填充该表。(或者，可以使用\$SYSTEM.SQL.Schema.QueryToTable()方法从现有表定义创建新表，并在单个操作中插入现有表中的数据。)

```
ClassMethod Insert()
{
    w !, "?????"
    &sql(
        CREATE TABLE SQLUser.MyStudents
        (
            StudentName VARCHAR(32),
            StudentDOB DATE,
            StudentAge INTEGER COMPUTECODE
            {
                SET {StudentAge} = $PIECE(($PIECE($H, ",", 1) - {StudentDOB}) / 365,
".", 1)
            }
            CALCULATED
        )
    )
    if SQLCODE=0 {
        w !, "?????SQLCODE=", SQLCODE
    }
```

```

} elseif SQLCODE=-201 {
    w !,"?????SQLCODE=",SQLCODE
}
w !,"???????"
n SQLCODE,%ROWCOUNT,%ROWID
&sql(
    INSERT INTO SQLUser.MyStudents
    (
        StudentName,StudentDOB
    )
    SELECT Name,DOB
    FROM Sample.Person WHERE Age <= '21'
)
if SQLCODE=0 {
    w !,"???????",%ROWCOUNT
    w !,"????????????ID=",%ROWID
} else {
    w !,"?????SQLCODE=",SQLCODE
}
}

```

要显示数据，请转到管理门户，选择所需命名空间的全局选项。滚动到“SQLUser.MyStudentsD”并单击Data选项。

以下程序显示MyStudents表数据，然后删除该表：

```
SELECT * FROM SQLUser.MyStudents ORDER BY StudentAge
```

```

&sql(DROP TABLE SQLUser.MyStudents)
IF SQLCODE=0 {WRITE !,"Table deleted" }
ELSE {WRITE !,"SQLCODE=",SQLCODE," ",%msg }

```

默认情况下，插入查询结果操作是原子操作。指定的所有行都插入到表中，或者没有插入任何行。例如，如果插入指定行中的一行会违反外键引用完整性，则插入将失败，并且不会插入任何行。此默认值是可修改的，如下所述。

将数据复制到复制表中

只要列顺序匹配且数据类型兼容，就可以使用INSERT WITH SELECT*将数据从表复制到复制表。列名不必匹配。

```
INSERT INTO Sample.DupTable SELECT * FROM Sample.SrcTable
```

- 与数据值兼容的数据类型：例如，可以将整数字段中的整数数据插入到VARCHAR字段中。如果任何数据值与目标数据类型不兼容，插入将失败，并显示SQLCODE-104。
- 与数据值兼容的数据类型长度：定义的列数据长度不必彼此匹配，只需与实际数据匹配即可。例如，SrcTable可以具有列FullName VARCHAR(60)，而DupTable可以具有对应的PersonName VARCHAR(40)。只要现有的FullName值都不超过40个字符，插入就会成功。如果任何FullName值超过40个字符，插入将失败，并显示SQLCODE-104。
- 兼容的列顺序：这两个表必须具有相同的列顺序。否则将生成SQLCODE-64错误。DDL CREATE TABLE操作按定义的顺序列出列。定义表的持久化类按字母顺序列出列。
- 兼容列计数：目标表可以具有复制列之外的其他列。例如，SrcTable可以具有列FullName VARCHAR(60)、Age INTEGER和DupTable可以具有PersonName VARCHAR(60)、Year INTEGER和ShoeSize INTEGER。但是，请注意，定义表的持久化类按字母顺序列出列。
- 私有行ID：定义表时，RowID字段被定义为公共或私有(隐藏)。默认情况下，DDL CREATE TABLE操作将R

owID定义为私有。默认情况下，定义表的持久化类将RowID定义为公共的；要使其成为私有的，必须在定义持久化类时指定SqlRowIdPrivate类关键字。

复制表的最简单方法是使用私有的RowID定义源表和目标表。但是，外键只能引用具有公共RowID的表。表复制操作的行为如下：

- 如果源私有，目标私有：可以使用INSERT、SELECT和SELECT*将数据复制到复制表。
- 如果源为PUBLIC，目标为PUBLIC：不能使用INSERT
SELECT将数据复制到重复表。将生成SQLCODE-111错误。
- 如果源为私有，目标为公共：不能使用INSERT SELECT将数据复制到重复表。将生成SQLCODE-111错误。
- 如果Source是公共的而Destination是私有的:不能使用带有SELECT *的INSERT SELECT将数据复制到重复表中。会生成一个SQLCODE -64错误，因为RowID出现在一个选择列表中使该选择列表不兼容。
可以使用包含所有字段名(不包括RowID)的列表的INSERT SELECT将数据复制到重复表中。
但是，如果Source有一个外键public RowID，则不会为目标表保留外键关系。
目的地将拥有新的系统生成的RowIDs。

如果Source具有外键public RowID，并且希望Destination具有相同的外键关系，则必须使用ALLOWIDENTITYINSERT=1参数定义Destination。将目标定义为持久类时，请指定参数ALLOWIDENTITYINSERT=1；。使用CREATE TABLE定义目标时，请指定%CLASSPARAMETER ALLOWIDENTITYINSERT=1。如果将表定义为ALLOWIDENTITYINSERT=1，则不能使用SetOption(“ IdentityInsert ”)方法更改此设置。

- 定义这些表的持久化类是否为Final对将数据复制到复制表中没有任何影响。
- 此操作可用于将现有数据复制到重新定义的表中，该表将接受在原始表中无效的未来列数据值。

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E4%B8%89%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-insert%E5%BC%88%E4%BA%8C%E5%BC%89>