

文章

姚鑫 · 十月 25, 2021 阅读大约需 7 分钟

第五十六章 SQL命令 INSERT OR UPDATE

第五十六章 SQL命令 INSERT OR UPDATE

在表中添加新行或更新表中的现有行。

大纲

```
INSERT OR UPDATE [%keyword] [INTO] table
    SET column = scalar-expression {,column2 = scalar-expression2} ... |
    [ (column{,column2} ...) ] VALUES (scalar-expression {,scalar-
expression2} ...) |
    VALUES :array() |
    [ (column{,column2} ...) ] query |
    DEFAULT VALUES
```

参数

- %keyword - 可选-以下一个或多个关键字选项，以空格分隔：%NOCHECK、%NOFPLAN、%NOINDEX、%NOJOURN、%NOLOCK、%NOTRIGGER、%PROFILE、%PROFILEALL。
- table - 要对其执行插入操作的表或视图的名称。此参数可以是子查询。INTO关键字是可选的。
- column - 可选-
与提供的值列表顺序对应的列名或以逗号分隔的列名列表。如果省略，值列表将按列号顺序应用于所有列。
- scalar-expression - 为相应列字段提供数据值的标量表达式或以逗号分隔的标量表达式列表。
- :array() - 仅嵌入式SQL-
指定为主机变量的值的动态本地数组。必须未指定数组的最低下标级别。因此:myupdates(), :myupdates(5,)和:myupdates(1,1,)都是有效的规范。
- query - 一种选择查询，其结果集为一行或多行的相应列字段提供数据值。

描述

INSERT或UPDATE语句是INSERT语句的扩展(它与INSERT语句非常相似)：

- 如果指定的记录不存在，则INSERT或UPDATE执行INSERT。
- 如果指定的记录已存在，则INSERT或UPDATE执行更新。它使用指定的字段值更新记录。即使指定的数据与现有数据相同，也会进行更新。

INSERT或UPDATE通过将唯一关键字字段值与现有数据值匹配来确定记录是否存在。如果发生违反唯一键约束的情况，则INSERT或UPDATE将执行UPDATE操作。请注意，唯一键字段值可能不是在INSERT或UPDATE中显式指定的值；它可能是列默认值或计算值的结果。当对切片表运行INSERT或UPDATE时，如果切片键与UNIQUE KEY约束相同(或是其子集)，则INSERT或UPDATE将执行UPDATE操作。如果INSERT或UPDATE因为找到任何其他唯一键值(不是切片键)而尝试执行更新，则该命令会失败，并由于UNIQUE约束失败而出现SQLCODE-119错误。

注意：由于%NOCHECK关键字禁用唯一值检查，因此INSERT或UPDATE %NOCHECK总是导致INSERT操作。因此，请不要指定%NOCHECK。

单个记录的INSERT或UPDATE始终将%ROWCOUNT变量设置为1，并将已插入或更新的行的%ROWID变量设置为1。

INSERT或UPDATE语句与SELECT语句组合可以插入和/或更新多个表行。

INSERT或UPDATE使用相同的语法，并且通常具有与INSERT语句相同的功能和限制。这里描述了插入或更新的特殊注意事项。除非此处另有说明，否则请参阅插入以了解详细信息。

权限

INSERT或UPDATE同时需要插入和更新权限。必须将这些权限作为表级权限或列级权限拥有。对于表级权限：

- 无论实际执行的是什么操作，用户都必须拥有对指定表的INSERT和UPDATE权限。
- 如果使用SELECT查询插入或更新另一个表中的数据，则用户必须对该表具有SELECT权限。

如果用户是表的所有者(创建者)，则会自动授予该用户对该表的所有权限。否则，必须授予用户对该表的权限。否则将导致SQLCODE-99错误，因为%msg用户 ' name ' 没有该操作的特权。可以通过调用%CHECKPRIV命令来确定当前用户是否具有适当的权限。可以使用GRANT命令为用户分配表权限。

IDKEY字段

可以插入IDKEY字段值，但不能更新IDKEY字段值。如果表具有IDKEY索引和另一个唯一键约束，则INSERT或UPDATE将匹配这些字段以确定是执行INSERT还是UPDATE。如果另一个键约束失败，则强制INSERT或UPDATE执行更新而不是INSERT。但是，如果指定的IDKEY字段值与现有IDKEY字段值不匹配，则此更新将失败并生成SQLCODE E-107错误，因为更新正在尝试修改IDKEY字段。

例如，表MyTest定义了四个字段：A、B、C、D，具有IDKEY(A, B)和Unique(C, D)约束。该表包含以下记录：

Row 1: A=1, B=1, C=2, D=2

Row 2: A=1, B=2, C=3, D=4

调用INSERT或UPDATE ABC(A, B, C, D)值(2, 2, 3, 4)，因为UNIQUE(C, D)约束失败，所以该语句不能执行INSERT。相反，它会尝试更新第2行。第2行的IDKEY为(1, 2)，因此INSERT或UPDATE语句将尝试将字段A的值从1更改为2。但无法更改IDKEY值，因此更新失败，并显示SQLCODE-107错误。

计数器字段

当执行INSERT或UPDATE时，IRIS最初假定操作将是INSERT。因此，它将用于向串行(%Library.Counter)字段提供整数的内部计数器加1。INSERT使用这些递增的计数器值将整数值分配给这些字段。但是，如果IRIS确定该操作需要更新，则INSERT或UPDATE已经递增了内部计数器，但它不会将这些递增的整数值分配给计数器字段。如果下一个操作是INSERT，则会导致这些字段的整数序列出现间隙。下面的示例显示了这一点：

1. 内部计数器值为4。INSERT或UPDATE递增内部计数器，然后插入行5：内部计数器=5，串行字段值=5。
2. INSERT或UPDATE递增内部计数器，然后确定它必须对现有行执行更新：INTERNAL COUNTER=6，不更改字段计数器。
3. INSERT或UPDATE递增内部计数器，然后插入一行：内部计数器=7，序列字段值=7。

Identity和RowID字段

INSERT或UPDATE对RowID值分配的影响取决于是否存在标识字段：

- 如果没有为表定义标识字段，则INSERT操作会导致 IRIS自动将下一个连续整数值分配给ID(RowID)字段。更新操作对后续插入没有影响。因此，INSERT或UPDATE执行与INSERT相同的INSERT操作。
- 如果为表定义了标识字段，则INSERT或UPDATE会导致 IRIS在确定操作是INSERT还是UPDATE之前，将用于向标识字段提供整数的内部计数器加1。插入操作将该递增的计数器值分配给标识字段。但是，如果 IRIS确定INSERT或UPDATE操作需要更新，则它已经递增了内部计数器，但不会分配这些递增的整数值。如果下一个INSERT或UPDATE操作是INSERT，则会导致标识字段的整数序列出现间隙。RowID字段值取自Identity字段值，导致ID(RowID)整数值分配存在差距。

示例

以下五个示例：创建一个新表(SQLUser.CaveDwell);使用INSERT或UPDATE用数据填充该表；使用INSERT或UPDATE添加新行并更新现有行；使用SELECT*显示数据；以及删除该表。

以下示例使用CREATE TABLE创建具有唯一字段(NUM)的表：

```
ClassMethod InsertOrUpdate()  
{  
    &sql(  
        CREATE TABLE SQLUser.CaveDwellers (  
            Num          INT UNIQUE,  
            CaveCluster  CHAR(80) NOT NULL,  
            Troglodyte   CHAR(50) NOT NULL,  
            CONSTRAINT CaveDwellerPK PRIMARY KEY (Num)  
        )  
    )  
    if SQLCODE = 0 {  
        w !,"???"  
    } elseif SQLCODE = -201 {  
        w !,"?????"  
    } else {  
        w !,"SQL????????: ",SQLCODE  
        q  
    }  
}
```

下面的示例使用类定义定义同一个表，为num定义唯一键：

```
Class User.CaveDwellers Extends %Persistent [ ClassType = persistent, DdlAllowed, Final, Owner = {yx}, ProcedureBlock, SqlRowIdPrivate, SqlTableName = CaveDwellers ]  
{  
  
    Property Num As %Library.Integer(MAXVAL = 2147483647, MINVAL = -2147483648) [ SqlColumnNameNumber = 2 ];  
  
    Property CaveCluster As %Library.String(MAXLEN = 80) [ Required, SqlColumnNameNumber = 3 ];  
  
    Property Troglodyte As %Library.String(MAXLEN = 50) [ Required, SqlColumnNameNumber = 4 ]  
    ;  
  
    /// DDL Unique Key Specification  
    Index CAVEDWELLERSUNIQUE1 On Num [ SqlName = CAVEDWELLERS_UNIQUE1, Unique ];  
  
    /// DDL Primary Key Specification
```

```
Index CaveDwellerPK On Num [ PrimaryKey, Type = index, Unique ];

/// Bitmap Extent Index auto-
generated by DDL CREATE TABLE statement. Do not edit the SqlName of this index.
Index DDLBEIndex [ Extent, SqlName = "%DDLBEIndex", Type = bitmap ];

}
```

```
SELECT * FROM SQLUser.CaveDwellers ORDER BY Num
```

以任何顺序运行以下两个示例一次或多次。他们将插入记录1到5。如果记录4已经存在，插入或更新将更新它。使用 SELECT *示例显示表格数据:

```
ClassMethod InsertOrUpdate1()
{
    &sql(
        INSERT OR UPDATE INTO SQLUser.CaveDwellers
        (
            Num, CaveCluster, Troglodyte
        )
        VALUES
        (
            3, 'Bedrock', 'Flintstone,Fred'
        )
    )
    if SQLCODE = 0 {
        SET rcount=%ROWCOUNT
    }
    &sql(
        INSERT OR UPDATE INTO SQLUser.CaveDwellers
        (
            Num, CaveCluster, Troglodyte
        )
        VALUES
        (
            4, 'Bedrock1', 'Flintstone,Wilma'
        )
    )
    if SQLCODE = 0 {
        s rcount = rcount + %ROWCOUNT
        w !,rcount," records inserted/updated"
    } else {
        w !,"Insert/Update failed, SQLCODE=",SQLCODE
    }
}
```

```
ClassMethod InsertOrUpdate2()
{
    n SQLCODE,%ROWCOUNT,%ROWID
    &sql(
        INSERT OR UPDATE SQLUser.CaveDwellers
        (
```

```
        Num,CaveCluster,Troglodyte
    )
    SELECT %ID,Home_City,Name
    FROM Sample.Person
    WHERE %ID BETWEEN 2 AND 5)
if SQLCODE=0 {
    w !,"Insert/Update succeeded"
    w !,%ROWCOUNT," records inserted/updated"
    w !,"Row ID=",%ROWID
} else {
    w !,"Insert/Update failed, SQLCODE=",SQLCODE
}
}
```

以下示例删除该表:

```
ClassMethod InsertOrUpdate3()
{
    &sql(DROP TABLE SQLUser.CaveDwellers)
    if SQLCODE = 0 {
        w !,"???"
    } elseif SQLCODE = -30 {
        w !,"???"
    } else {
        w !,"?????. SQLCODE=",SQLCODE
    }
}
```

[#SQL #Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E5%85%AD%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-insert-or-update>