

文章

姚鑫 · 十一月 23, 2021 阅读大约需 3 分钟

第八十五章 SQL命令 UPDATE（四）

第八十五章 SQL命令 UPDATE（四）

示例

本节中的示例更新SQLUser.MyStudents表。

下面的示例创建SQLUser.MyStudents表，并用数据填充它。

因为这个示例的重复执行会积累具有重复数据的记录，所以它使用TRUNCATE TABLE在调用INSERT之前删除旧数据。

在调用UPDATE示例之前执行这个示例：

```
ClassMethod Update1()
{
CreateStudentTable
    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 "DDL %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,!
        w rtn.%Message,!
    }
}

RemoveOldData
    s clearit = "TRUNCATE TABLE SQLUser.MyStudents"
    s qStatus = tStatement.%Prepare(clearit)
    if qStatus'=1 {
        w "Truncate %Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s truncrtn = tStatement.%Execute()
    if truncrtn.%SQLCODE = 0 {
        w !,"?????????",!
```

```

    } elseif truncrtn.%SQLCODE = 100 {
        w !,"????????",!
    } else {
        w !,"truncate failed, SQLCODE=",truncrtn.%SQLCODE," ",truncrtn.%Message,!
    }
}
PopulateStudentTable
s studentpop=2
s studentpop(1)="INSERT INTO SQLUser.MyStudents (StudentName,StudentDOB) "
s studentpop(2)="SELECT Name,DOB FROM Sample.Person WHERE Age <= '21'"
s qStatus = tStatement.%Prepare(.studentpop)
if qStatus '= 1 {
    w "?? %Prepare ??:"
    d $System.Status.DisplayError(qStatus)
    q
}
s poprtn = tStatement.%Execute()
if poprtn.%SQLCODE = 0 {
    w !,"?????",!
    w poprtn.%ROWCOUNT," rows inserted"
} else {
    w !,"?????, SQLCODE=",poprtn.%SQLCODE,!
    w poprtn.%Message
}
}
}

```

可以使用下面的查询来显示这些示例的结果:

```
SELECT %ID,* FROM SQLUser.MyStudents ORDER BY StudentAge,%ID
```

下面的一些UPDATE示例依赖于其他UPDATE示例设置的字段值;
它们应该按照指定的顺序运行。

在下面的动态SQL示例中, SET field=value UPDATE修改选定记录中的指定字段。
在我的学生表中, 7岁以下的孩子是不给分数的:

```

ClassMethod Update2()
{
    s studentupdate=3
    s studentupdate(1)="UPDATE SQLUser.MyStudents "
    s studentupdate(2)="SET FinalGrade='NA' "
    s studentupdate(3)="WHERE StudentAge <= 6"
    s tStatement = ##class(%SQL.Statement).%New(0,"Sample")
    s qStatus = tStatement.%Prepare(.studentupdate)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s uprtn = tStatement.%Execute()
    if uprtn.%SQLCODE=0 {
        w !,"?????"
        w !,"Rows updated=",uprtn.%ROWCOUNT," Final RowID=",uprtn.%ROWID
    } else {
        w !,"Table update failed, SQLCODE=",uprtn.%SQLCODE," ",uprtn.%Message
    }
}

```

```

    }
}

```

在下面的基于游标的嵌入式SQL示例中，SET field1=value1,field2=value2 UPDATE修改选定记录中的几个字段。在MyStudents表中，它用Q1和Q2成绩更新指定的学生记录：

```

ClassMethod Update2()
{
    #SQLCompile Path = Sample
    n %ROWCOUNT,%ROWID
    &sql(
        DECLARE StuCursor CURSOR FOR
            SELECT * FROM MyStudents
            WHERE %ID IN(10,12,14,16,18,20,22,24) AND StudentAge > 6)
    &sql(OPEN StuCursor)
    q:(SQLCODE'=0)
    for {
        &sql(FETCH StuCursor)
        q:SQLCODE
        &sql(
            Update MyStudents SET Q1Grade='A',Q2Grade='A'
            WHERE CURRENT OF StuCursor
        )
        if SQLCODE=0 {
            w !,"?????"
            w !,"Row count=",%ROWCOUNT," RowID=",%ROWID
        } else {
            w !,"?????", SQLCODE=",SQLCODE
        }
    }
    &sql(CLOSE StuCursor)
}

```

在下面的动态SQL示例中，字段列表VALUES值列表UPDATE修改选定记录中几个字段的值。在我的学生表中，没有拿到期末成绩的孩子也没有拿到季度成绩：

```

ClassMethod Update4()
{
    s studentupdate=3
    s studentupdate(1)="UPDATE SQLUser.MyStudents "
    s studentupdate(2)="(Q1Grade,Q2Grade,Q3Grade) VALUES ('x','x','x') "
    s studentupdate(3)="WHERE FinalGrade='NA' "
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(.studentupdate)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s uprtn = tStatement.%Execute()
    if uprtn.%SQLCODE=0 {
        w !,"Table Update successful"
        w !,"Rows updated=",uprtn.%ROWCOUNT," Final RowID=",uprtn.%ROWID
    } else {
        w !,"Table Update failed, SQLCODE=",uprtn.%SQLCODE," ",uprtn.%Message,!
    }
}

```

```

    }
}

```

在下面的动态SQL示例中，VALUES值列表UPDATE修改所选记录中的所有字段值。

注意，此语法要求为记录中的每个字段指定一个值。

在“user.student”表中，有几个孩子已经从学校退学。

保留它们的记录id和名称，并在名称后附加单词withdraw;

所有其他数据被删除，DOB字段用于提取日期:

```

ClassMethod Update5()
{
    s studentupdate=4
    s studentupdate(1)="UPDATE SQLUser.MyStudents "
    s studentupdate(2)="VALUES (StudentName||' WITHDRAWN', "
    s studentupdate(3)="$PIECE($HOROLOG,' ',1),00,'-','-','-','XX') "
    s studentupdate(4)="WHERE %ID IN(7,10,22)"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(.studentupdate)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s uprtn = tStatement.%Execute()
    if uprtn.%SQLCODE=0 {
        w !,"Table Update successful"
        w !,"Rows updated=",uprtn.%ROWCOUNT," Final RowID=",uprtn.%ROWID
    } else {
        w !,"Table Update failed, SQLCODE=",uprtn.%SQLCODE," ",uprtn.%Message,!
    }
}

```

在下面的动态SQL示例中，子查询UPDATE使用子查询来选择记录。

然后使用SET field=value语法修改这些记录。

因为在SQLUser中，StudentAge是从出生日期开始计算的。

我的学生们，任何不满一年的人的计算年龄都是<Null>，任何出生日期已被取消的人的计算年龄都非常高。

这里标记了StudentName字段，以便将来确认出生日期:

```

ClassMethod Update6()
{
    s studentupdate=3
    s studentupdate(1)="UPDATE (SELECT StudentName FROM SQLUser.MyStudents "
    s studentupdate(2)="WHERE StudentAge IS NULL OR StudentAge > 21) "
    s studentupdate(3)="SET StudentName = StudentName||' *** CHECK DOB' "
    s tStatement = ##class(%SQL.Statement).%New(0,"Sample")
    s qStatus = tStatement.%Prepare(.studentupdate)
    if qStatus'=1 {
        w "%Prepare failed:"
        d $System.Status.DisplayError(qStatus)
        q
    }
    s uprtn = tStatement.%Execute()
    if uprtn.%SQLCODE=0 {
        w !,"Table Update successful"
    }
}

```

第八十五章 SQL命令 UPDATE (四)

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

```

        w !,"Rows updated=",uprtn.%ROWCOUNT," Final RowID=",uprtn.%ROWID
    } else {
        w !,"Table Update failed, SQLCODE=",uprtn.%SQLCODE," ",uprtn.%Message,!
    }
}

```

在下面的嵌入式SQL示例中，VALUES:array() UPDATE修改选定记录中的数组中列号指定的字段值。

只能在嵌入式SQL中进行VALUES:array()更新。

注意，此语法要求您通过DDL列号指定每个值(包括在列计数中RowID列(列1)，但不为这个不可修改的字段提供值)。

在MyStudents表中，在Q1Grade(第5列)和Q2Grade(第6列)字段中，4到6岁(包括4到6岁)的孩子会得到一个“P”(代表“Present”)。

所有其他记录数据保持不变:

```

ClassMethod Update7()
{
    s array(5)="P"
    s array(6)="P"
    &sql(
        update SQLUser.MyStudents VALUES :array()
        where FinalGrade='NA' AND StudentAge > 3)
    if SQLCODE=0 {
        w "Table Update successful",!
        w "Rows updated=",%ROWCOUNT," Final RowID=",%ROWID
    } else {
        w "Table Update failed, SQLCODE=",SQLCODE,!
    }
}
}

```

#SQL #Caché

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%B%E5%8D%81%E4%BA%94%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-update%EF%BC%88%E5%9B%9B%E4%BC%89>