文章
姚 鑫 · 十月 21, 2021 阅读大约需 10 分钟

第五十二章 SQL命令 INSERT（一）

第五十二章 SQL命令 INSERT（一）

向表中添加新行(或多行)。

大纲

```
INSERT [%keyword] [INTO] table
          SET column1 = scalar-expression1
                 {,column2 = scalar-expression2} ...  |
          [ (column1{,column2} ...) ]
                 VALUES (scalar-expression1 {,scalar-expression2} ...)  |
          VALUES :array()  |
          [ (column1{,column2} ...) ] query  |
          DEFAULT VALUES
```

参数

- %keyword - 可选参数:%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。表名(或视图名)可以是限定的(schema.table)，也可以是不限定的(Table)。使用架构搜索路径(如果提供)或默认架构名称将非限定名称与其架构匹配。
- 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：myupdate()、：myupdate(5，)和：myupdate(1，1，)都是有效的规范。
- query - 一种选择查询，其结果集为一个或多个新行的相应列字段提供数据值。

描述

INSERT语句有两种使用方式：

- 单行插入会向表中添加一个新行。它为所有指定的列(字段)插入数据值，并将未指定的列值默认为NULL或定义的默认值。它将%ROWCOUNT变量设置为受影响的行数(始终为1或0)。
- 带有SELECT的INSERT会向表中添加多个新行。它为查询结果集中每一行的所有指定列(字段)插入数据值，并将未指定的列值默认为NULL或定义的默认值。INSERT语句与SELECT查询的结合使用通常用于用从其他表中提取的现有数据填充表，如下面的 “ 插入查询结果 ” 部分所述。

INSERT OR UPDATE

INSERT或UPDATE语句是INSERT语句的变体，它同时执行INSERT和UPDATE操作。首先，它尝试执行插入操作。如果INSERT请求由于唯一键冲突而失败(对于某个唯一键的字段，存在与为INSERT指定的行具有相同值的行)，则它会自动转换为该行的UPDATE请求，并且INSERT或UPDATE使用指定的字段值更新现有行。

INSERT或UPDATE不支持快速插入。

%Keyword字选项

指定%Keyword参数将按如下方式限制处理：

- %NOCHECK-不执行唯一值检查和外键引用完整性检查。也不执行针对数据类型、最大长度、数据约束和其他验证条件的列数据验证。通过视图执行插入时，不执行视图的WITH CHECK选项验证。

注意：由于使用%NOCHECK可能导致无效数据，因此只有在从可靠的数据源执行批量插入或更新时才应使用此%Keyword字参数。

用户必须具有当前命名空间的相应%NOCHECK管理权限才能应用此限制。否则将导致SQLCODE-99错误，因为%msg用户‘ name ’没有%NOCHECK权限。

如果希望在指定%NOCHECK时防止插入非唯一数据值，请在插入之前执行EXISTS检查。

如果只希望禁用外键引用完整性检查，请使用$SYSTEM.SQL.SetFilerRefIntegrity()方法，而不是指定%NOCHECK。或者，可以使用NOCHECK关键字定义外键，这样就永远不会执行外键引用完整性检查。

- %NOFPLAN-
  忽略此操作的冻结计划(如果有)；该操作将生成新的查询计划。冻结的计划将保留，但不会使用。

- %NOINDEX-在插入处理期间未设置索引映射。用户必须具有当前命名空间的相应%noindex管理权限才能应用此限制。否则将导致SQLCODE-99错误，因为%msg用户‘ name ’没有%noindex权限。

- %NOJOURN-在插入操作期间禁止日志记录。任何行中所做的任何更改都不会被记录下来，包括拉出的任何触发器。如果在使用%NOJOURN的语句之后执行ROLLBACK，则不会回滚该语句所做的更改。
- %NOLOCK-插入时该行未锁定。这应该仅在单个用户/进程更新数据库时使用。用户必须具有当前命名空间的相应%NOLOCK管理权限才能应用此限制。否则将导致SQLCODE-99错误，因为%msg用户‘ name ’没有%NOLOCK权限。
- %NOTRIGGER-在插入处理期间不拉取基表插入触发器。用户必须具有当前命名空间的相应%NOTRIGGER管理权限才能应用此限制。否则将导致SQLCODE-99错误，因为%msg用户‘ name ’没有%NOTRIGGER权限。
- %PROFILE或%PROFILEALL-如果指定了其中一个关键字指令，则生成SQLStats收集代码。这与启用PTools时生成的代码相同。不同之处在于，SQLStats收集代码只为该特定语句生成。正在编译的例程/类中的所有其他SQL语句将生成代码，就像PTools已关闭一样。这使用户能够分析/检查应用程序中的特定问题SQL语句，而无需收集未被调查的SQL语句的无关统计信息。

%PROFILE收集主查询模块的SQLStat。%PROFILEALL收集主查询模块及其所有子查询模块的SQLStat。

可以按任意顺序指定多个%KEYWORD参数。多个参数由空格分隔。

表参数

可以指定要直接插入到表中的表参数、通过视图插入的表参数或通过子查询插入的表参数。如创建视图中所述，通过视图插入受要求和限制的约束。下面是使用子查询代替TABLE参数的INSERT示例：

```
INSERT INTO (SELECT field1 AS ff1 FROM MyTable) (ff1) VALUES ('test')
```

子查询目标必须是可更新的，遵循用于确定视图的查询是否可更新的相同标准。尝试使用不可更新的视图或子查询进行插入会生成SQLCODE-35错误。

不能在表参数中指定表值函数或联接语法。

赋值

本节介绍如何在INSERT操作期间将数据值分配给列(字段)：
- 值赋值语法描述将数据值指定为列(字段)的文字的各种语法选项。
- 显示到逻辑数据的转换
- %SerialObject属性
- 非显示字符
- 特殊变量
- 流数据
- 列出结构化数据
- Identity、ROWVERSION和串行计数器
- 计算字段值
- 默认值子句

如果省略COLUMN LIST参数，则INSERT将假定按列号顺序插入所有列。如果指定列列表，则各个值必须在位置上与列列表中的列名相对应。

值赋值语法

插入记录时，可以通过多种方式为指定列赋值。默认情况下，所有未指定的列必须接受NULL或具有定义的默认值。

- 显式列名使用SET关键字，将一个或多个COLUMN=标量-表达式对指定为逗号分隔的列表。例如：

```
SET StatusDate='05/12/06',Status='Purged'
```

- 使用VALUES关键字显式列名，指定与相应的标量表达式列表相等的列列表。
  例如:

```
(StatusDate,Status) VALUES ('05/12/06','Purged')
```

当将标量表达式值赋给列列表时，每个指定的列必须有一个标量表达式。

- 没有列名。在使用不带列列表的VALUES关键字时，请按顺序指定与表的字段相对应的标量表达式列表。例如：

```
VALUES ('Fred Wang',65342,'22 Main St. Anytown MA','123-45-6789')
```

必须按列号顺序指定值。必须为采用用户提供的值的每个基表列指定值；使用列顺序的插入不能采用定义的字段默认值。如果指定的值少于表列的数量，则会发出SQLCODE-62错误。如果指定的值多于表列的数量，则会发出SQLCODE-116错误。

RowID列不能由用户指定，因此不包括在此语法中。

默认情况下，不能使用此语法填充具有定义的标识字段或RowVersion字段的表。如果定义了其中一个字段，则如果没有为这些字段指定值，此INSERT语法将发出SQLCODE-62错误；如果确实为这些字段指定值，此INSERT语法将发出SQLCODE-138无法插入/更新只读字段的值错误。(身份字段可以配置为允许用户提供的值；请参阅身份字段。)

可以使用此语法填充具有定义的序列(%COUNTER)字段或%AutoIncrement字段的表，但必须为这些计数器字段指定用户提供的值。

- 没有列名。当使用不带列列表的VALUES关键字时，请指定一个标量表达式的动态本地数组，该数组隐式对应于按列顺序的行的列。例如：

```
VALUES :myarray()
```

此值赋值只能使用主机变量从嵌入式SQL执行。与所有其他值赋值不同，这种用法允许将指定要插入哪些列的时间推迟到运行时(通过在运行时填充数组)。所有其他类型的插入都需要指定准备插入时要插入的列。此语法不能与链接表一起使用；尝试这样做会导致SQLCODE-155错误。

必须按列号顺序指定值。必须为采用用户提供的值的每个基表列指定值；使用列顺序的插入不能采用定义的字段默认值。提供的数组值必须以array(2)开头。第1列是RowID字段；不能为RowID字段指定值。

如果指定列名和相应的数据值，则可以省略定义了默认值或接受NULL的列。INSERT可以为大多数字段数据类型插入默认值，包括流字段。

如果未指定列名，则数据值必须在位置上与定义的列列表相对应。必须为每个用户可指定的基表列指定值；不能使用定义的默认值。(当然，可以指定空字符串作为列值。)

显示到逻辑数据的转换

数据以逻辑模式格式存储。例如，日期存储为天数的整数，时间存储为午夜起的秒数，%list存储为编码字符串。大多数其他数据(如字符串和数字)不需要转换；无论当前模式如何，它们都以相同的格式输入和存储。

对于转换的数据，可以以逻辑模式(默认)输入数据，或者通过指定选择模式以更易于阅读的格式(显示模式或ODBC模式)输入数据。例如，通过指定选择模式，可以输入显示格式(例如2018年2/22/2018)、odbc格式(例如2018-02-22)或逻辑格式(例如64701)的日期。对于某些数据类型，还可以在ODBC或显示选择模式下以逻辑格式指定数据。可以在SQL执行环境中显式设置SELECT模式，如下所示：

- 在ObjectScript程序中或从Terminal 接口：调用SetOption()方法，如下所示：设置SET status=$SYSTEM.SQL.Util.SetOption("SelectMode",n,.oldval)，其中整数n为0=逻辑(默认值)，1=odbc，2=显示。
- 在动态SQL中，指定%SelectMode=n属性，其中整数n为0=逻辑(默认值)，1=ODBC，2=显示。
- 从SQL Shell中，指定SET SELECTMODE关键字，其中关键字=逻辑、ODBC或DISPLAY。
- 从管理门户中选择系统资源管理器、SQL，然后使用显示模式下拉列表指定逻辑模式、ODBC模式或显示模式。

非逻辑模式格式的输入数据必须转换为逻辑模式格式进行存储。编译后的SQL支持将输入值从显示或ODBC格式自动转换为逻辑格式。无法转换的输入值会导致SQLCODE错误，例如SQLCODE-146和SQLCODE-147。输入数据的自动转换需要两个因素：编译时，SQL必须指定运行时模式；执行时，SQL必须在逻辑模式环境中执行。

- 在嵌入式SQL中，如果指定#SQLCompile Select=Runtime，
  IRIS将使用将输入值从显示格式转换为逻辑模式存储格式的代码编译SQL语句。
  IRIS对单个值和值数组执行此模式转换。
- 在SQL CREATE Function、CREATE METHOD或CREATE PROCEDURE语句中，如果指定SELECTMODE运行时，
  IRIS将使用将输入值从显示格式转换为逻辑模式存储格式的代码编译SQL语句。

如果SQL执行环境处于逻辑模式，则数据以逻辑格式存储。这是所有 SQL执行环境的默认模式。

%SerialObject属性

将数据插入%SerialObject时，必须插入引用嵌入的%SerialObject的表(持久化类)；不能直接插入%SerialObject。从引用表中，可以执行以下任一操作：
- 使用引用字段将多个%SerialObject属性的值作为%List结构插入。例如，如果持久类具有引用包含特性Street、City和Country(按顺序)的序列对象的特性PAddress，则插入set PAddress=$LISTBUILD(‘ 123 Main St. ’，‘ Newtown ’，‘ USA ’)或(PAddress)值($LISTBUILD(‘ 123 Main St. ’，‘ Newtown ’，‘ USA ’))或(PAddress)值)。%List必须包含串行对象属性(或占位符逗号)的值，其顺序与这些属性在串行对象中指定的顺序相同。

此类型的插入可能不会执行%SerialObject属性值的验证。因此，强烈建议在使用%List结构插入%SerialObject属性值之后，使用$SYSTEM.SQL.Schema.ValidateTable()方法执行表数据验证。
- 使用下划线语法可以按任意顺序为各个%SerialObject属性插入值。例如，如果持久类具有引用包含特性Street、City和Country的序列对象的特性PAddress，则插入set PAddressCity=‘ Newtown ’，PAddressStreet=‘ 123 Main St. ’，PAddressCountry=‘ USA ’。未指定的串行对象属性默认为NULL。

此类型的插入执行%SerialObject属性值的验证。

非显示字符

可以使用CHAR函数和串联运算符插入非显示字符。例如，下面的示例插入一个由字母“ A ”、换行符和字母“ B ”组成的字符串：

```
INSERT INTO MyTable (Text) VALUES ('A'||CHAR(10)||'B')
```

请注意，要连接函数的结果，必须使用||连接运算符，而不是连接运算符。

查询可以使用LENGTH或$LENGTH函数确定是否存在非显示字符。

特殊变量

可以在列中插入以下特殊变量的值：

%TABLENAME或%CLASSNAME伪字段变量关键字。%TABLENAME返回当前表名。%CLASSNAME返回与当前表对应的类名。

以下一个或多个ObjectScript特殊变量(或其缩写)：$HOROLOG、$JOB、$NAMESPACE、$TLEVEL、$USERNAME、$ZHOROLOG、$ZJOB、$ZNSPACE、$ZPI、$ZTIMESTAMP、$ZTIMEZONE、$ZVERSION。

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E5%8D%81%E4%BA%8C%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-insert%EF%BC%88%E4%B8%80%EF%BC%89