

文章

[姚鑫](#) · 三月 13, 2021 阅读大约需分钟

第九章 SQL查询数据库(二)

第九章 SQL查询数据库(二)

调用用户定义函数的查询

InterSystems SQL允许您在SQL查询中调用类方法。这为扩展SQL语法提供了强大的机制。

若要创建用户定义的函数，请在持久性InterSystems IRIS类中定义一个类方法。该方法必须具有数字(非对象)返回值。这必须是一个类方法，因为在SQL查询中将没有对象实例可以在其上调用实例方法。还必须将其定义为SQL存储过程。

例如，我们可以在MyApp.Person类中定义一个Cube()方法：

```
Class MyApp.Person Extends %Persistent [DdlAllowed]
{
  /// Find the Cube of a number
  ClassMethod Cube(val As %Integer) As %Integer [SqlProc]
  {
    RETURN val * val * val
  }
}
```

可以使用CREATE FUNCTION, CREATE METHOD或CREATE PROCEDURE语句创建SQL函数。

要调用SQL函数，请指定SQL过程的名称。可以在可能指定标量表达式的任何地方以SQL代码调用SQL函数。函数名称可以使用其结构名称进行限定，也可以不限定。不合格的函数名称采用用户提供的模式搜索路径或系统范围内的默认模式名称。函数名称可以是定界标识符。

SQL函数必须具有用括号括起来的参数列表。参数列表可以为空，但括号是强制性的所有指定的参数均充当输入参数。不支持输出参数。

SQL函数必须返回一个值。

例如，以SQL查询将用户定义的SQL函数作为方法调用，就像它是内置SQL函数一样：

```
SELECT %ID, Age, Sample.Person_Cube(Age) FROM Sample.Person
```

The screenshot shows the SQL Editor with the following query: `SELECT %ID, Age, Sample.Person.Cube(Age) FROM Sample.Person`. Below the editor, a table displays the results of the query:

	ID	Age	Expression_3
	INTEGER	INTEGER	INTEGER
9	16	43	79507
10	17	88	681472
11	18	47	103823
12	19	93	804357
13	20	73	389017
14	21	7	343
15	22	78	474552
16	23	3	27
17	24	54	157464
18	25	72	373248
19	26	26	17576

对于Age的每个值,此查询将调用Cube()方法并将其返回值放入结果中。

SQL函数可能是嵌套的。

如果找不到指定的功能,则InterSystems IRIS会发出SQLCODE -359错误。如果指定的函数名称不明确,则InterSystems IRIS会发出SQLCODE -358错误。

查询串行对象属性

使用默认存储(%Storage.Persistent)从类中映射为SQL的子表的串行对象属性将在该类映射表中的单个列中映射。该列的值是串行对象属性列值。该单列属性映射为SQL %List字段。

例如,Sample.Person中的Home列定义为Property Home As Sample.Address; 它将映射到类Sample.Address扩展(%SerialObject),其中包含属性Street, City, State和PostalCode。

以下示例从该串行对象列返回值:

```
SELECT TOP 4 Name,Home_Street,Home_City,Home_State,Home_PostalCode
FROM Sample.Person
```

以下示例将所有串行对象列的值(按顺序)作为单个%List格式字符串返回,并将每一列的值作为%List的元素:

```
SELECT TOP 4 Name,$LISTTOSTRING(Home,'^'),$length(Name)
FROM Sample.Person
```

The screenshot shows the SQL Editor with the following query: `SELECT TOP 4 Name,$LISTTOSTRING(Home,'^'),$length(Name) FROM Sample.Person`. Below the editor, a table displays the results of the query:

	Name	Expression_2	Expression_3
	VARCHAR (50)	VARCHAR (255)	SMALLINT
1	yaoxin	889 Clinton Drive^St Louis^WI^78672	6
2	xiaoli	^^^	6
3	姚鑫	^^^	2
4	姚鑫	^^^	2

默认情况下,此“主页”列是隐藏的,并且不映射为Sample.Person的列。

查询集合

可以从SQL WHERE子句引用集合,如所示:

```
WHERE FOR SOME %ELEMENT(collectionRef) [AS label] (predicate)
```

FOR SOME %ELEMENT子句可用于指定STORAGEDEFAULT =“list”的列表集合和数组。谓词可以包含对伪列%KEY, % VALUE或两者的引用。一些示例应有助于阐明如何使用FOR SOME %ELEMENT子句。以返回其最喜欢的颜色包括“红色”每个人的名字和最喜欢的颜色的列表。

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FOR SOME %ELEMENT(FavoriteColors) (%Value = 'Red')
```

Name	FavoriteColors
yaoxin	Red, Orange, Yellow
姚鑫	Red, Orange, Yellow, Green
姚鑫	Red, Orange, Yellow, Green, Green
Isaacs, Roberta Z.	Red, Orange, Yellow, Green, Yellow
O'Rielly, Chris H.	Red, Red
Kratzmann, Emily Z.	Red, Green
Eno, George B.	Green, Red
Quixote, Terry J.	Red, Blue
Emerson, Edgar T.	White, Red
Pantaleo, Paul H.	Red
Bush, Jules K.	Black, Red
Ahmed, Elmo X.	Red, White
Jung, Keith W.	White, Red

任何SQL谓词都能出现在%Value(或%Key)之后,因此例如也是合法语法:

```
SELECT Name, FavoriteColors FROM Sample.Person
WHERE FOR SOME %ELEMENT(Sample.Person.FavoriteColors)
(%Value IN ('Red', 'Blue', 'Green'))
```

Name	FavoriteColors
yaoxin	Red, Orange, Yellow
姚鑫	Red, Orange, Yellow, Green
姚鑫	Red, Orange, Yellow, Green, Green
Isaacs, Roberta Z.	Red, Orange, Yellow, Green, Yellow
Vonnegut, Jose P.	Green, White
O'Rielly, Chris H.	Red, Red
Umansky, Susan C.	Blue
Kratzmann, Emily Z.	Red, Green
Eno, George B.	Green, Red
Houseman, Martin D.	Blue, Green
Quixote, Terry J.	Red, Blue
Hanson, George C.	Blue
Xerxes, Angelo P.	Yellow, Green

列表集合被认为是具有连续数字键1..2等数组集合的特例。数组集合具有任意的非空键。

```
FOR SOME (children) (%Key = 'betty' AND %Value > 5)
```

除了内置列表和数组集合类型之外,还可以通过为任何提供BuildValueArray()类方法来创建通用集合。Build ValueArray()类方法将属性转换为本地数组,其中数组的每个元素是一个%KEY,该值是对应的%VALUE。

除了可以在%KEY或%VALUE上进行简单选择之外,还可以在逻辑上连接两个集合,如以下示例所示:

```
FOR SOME %ELEMENT(flavors) AS f
  (f.%VALUE IN ('Chocolate', 'Vanilla') AND
  FOR SOME %ELEMENT(toppings) AS t
    (t.%VALUE = 'Butterscotch' AND
    f.%KEY = t.%KEY))
```

此示例有两个集合:Favors和TOPING,这两个集合通过键在位置上相关。该查询限定了将巧克力或香草指定为口味元素的行,并且还将奶油糖列为相应的配料,其中通过%key建立对应关系。

可以使用\$SYSTEM.SQL配置方法GetCollectionProjection()和SetCollectionProjection()来确定如果将集合映射为子表,则是否将集合映射为列。在编译或重新编译该类时,对该系统范围的设置所做的更改将对每个类生效。

使用说明和限制

- FOR SOME %ELEMENT只能出现在WHERE子句中。
- %KEY和/或%VALUE只能出现在FOR谓词中。
- 任何特定的%KEY或%VALUE只能被引用一次。
- %KEY和%VALUE可能不会出现在外部联接中。
- %KEY和%VALUE可能不会出现在值表达式中(仅在谓词中)。

调用文本搜索的查询

InterSystems IRIS支持所谓的“自由文本搜索”,包括支持:

- 通配符
- 填充物
- 分词搜索(也称为n-gram)
- 自动分类
- 词典管理

此功能使SQL能够支持全文索引,还使SQL能够索引和引用集合的单个元素,而无需集合映射为子表。虽然支持集合索引和全文索引的底层机制密切相关,但文本检索具有许多特殊性因此为文本检索提供了特殊的类和SQL功能。

伪字段

InterSystems SQL查询支持以伪字段值:

- %ID —返回RowID字段值,而不是RowID字段的实际名称是ID。
- %TABLENAME —返回在FROM子句中指定的现有表的限定名称。定义表时,使用限定的字母大小写返回合格的表名,而不是FROM子句中指定的字母大小写。如果FROM子句指定了不合格的表名,则%TABLENAME将返回合格的表名(schema.table),以及从用户提供的模式搜索路径或系统范围内的默认模式名称提供的模式名称。例如,如果FROM子句指定mytable,则%TABLENAME变量可能返回SQLUser.MyTable。
- %CLASSNAME —返回与FROM子句中指定的现有表相对应的合格类名称(package.class)。例如,如果FROM子句指定了SQLUser.mytable,则%CLASSNAME变量可能返回User.MyTable。

注意:请勿将%CLASSNAME伪字段值与%ClassName()实例方法混淆。它们返回不同的值。

伪字段变量只能为包含数据的表返回。

如果在FROM子句中指定了多个表,则必须使用表别名,如以下嵌入式SQL示例所示:

```
/// d ##class(PHA.TEST.SQL).Query3()
```

```

ClassMethod Query3(val As %Integer) As %Integer [ SqlProc ]
{
    &sql(SELECT P.Name,P.%ID,P.%TABLENAME,E.%TABLENAME
        INTO :name,:rid,:ptname,:etname
        FROM Sample.Person AS P,Sample.Employee AS E)
    IF SQLCODE<0 {
        WRITE "SQLCODE error ",SQLCODE," ",%msg QUIT
    } ELSEIF SQLCODE=100 {
        WRITE "Query returns no results" QUIT
    }
    WRITE ptname,"Person table Name is: ",name,!
    WRITE ptname,"Person table RowId is: ",rid,!
    WRITE "P alias TableName is: ",ptname,!
    WRITE "E alias TableName is: ",etname,!
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).Query3()
Sample.PersonPerson table Name is: Adams,Diane F.
Sample.PersonPerson table RowId is: 95
P alias TableName is: Sample.Person
E alias TableName is: Sample.Employee

```

为%TABLE NAME和%CLASS NAME列分配了默认的列名称Literal N，其中n是SELECT语句中伪字段变量的select-item位置。

查询元数据

可以使用Dynamic SQL返回有关查询的元数据，例如查询中指定的列数，查询中指定的列的名称(或别名)以及查询中指定的列的数据类型。

下面的ObjectScript Dynamic SQL示例为Sample.Person中的所有列返回列名和该列的ODBC数据类型的整数代码：

```

/// d ##class(PHA.TEST.SQL).Query4()
ClassMethod Query4()
{
    SET myquery="SELECT * FROM Sample.Person"
    SET rset = ##class(%SQL.Statement).%New()
    SET qStatus = rset.%Prepare(myquery)
    IF qStatus'=1 {
        WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT
    }
    SET x=rset.%Metadata.columns.Count()
    WHILE x>0 {
        SET column=rset.%Metadata.columns.GetAt(x)
        WRITE !,x," ",column.colName," ",column.ODBCType
        SET x=x-1
    }
    WRITE !,"end of columns"
}

```

```

DHC-APP>d ##class(PHA.TEST.SQL).Query4()

```

```

14 Office_Street 12
13 Office_State 12
12 Office_City 12
11 Home_Zip 12
10 Home_Street 12
9 Home_State 12
8 Home_City 12
7 Spouse 4
6 SSN 12
5 Name 12
4 FavoriteColors 12
3 DOB 9
2 Age 4
1 ID 4
end of columns

```

在此示例中，列以反向列顺序列出。请注意，包含列表结构化数据的FavoriteColors列返回的数据类型为12(VARCHAR)，因为ODBC将InterSystems IRIS列表数据类型值表示为以逗号分隔的值的字符串。

快速查询

InterSystems IRIS支持快速选择，这是一种内部优化，用于通过ODBC和JDBC快速执行查询。此优化将InterSystems全局变量映射到Java对象。它将全局节点(数据记录)的内容作为Java对象传递。收到这些Java对象后，它将从它们中提取所需值并生成结果。InterSystems IRIS会尽可能自动应用此优化。这种优化是自动的，对用户不可见。当准备好查询时，InterSystems IRIS会将查询标记为使用快速选择机制执行还是使用标准查询机制执行。

如果查询仅引用字段，常量或引用字段和/或常量的表达式，则快速选择将应用于%PARALLEL查询和针对分片表的查询。

服务器和客户端都必须支持快速选择。要在客户端中启用或禁用“快速选择”，请在类实例的定义中使用“属性选项”所示：

```

Properties p = new Properties();
p.setProperty("FeatureOption", "3"); // 1 is fast Select, 2 is fast Insert, 3 is both

```

由于性能差异，对于用户来说重要的是要知道具体情况限制了快速选择的应用。

表限制：无法使用快速选择来查询以下类型的表：

- 链接表
- 一个表，其主/数据映射具有多个节点
- 具有映射到同一数据位置的多个字段的表(仅可使用%Storage.SQL来实现)

字段限制：如果选择项列表中包含以下列，则无法使用“快速选择”执行查询。这些类型的列可以在表中定义，但是查询无法选择它们：

- 流字段(数据类型%Stream.GlobalCharacter或%Stream.GlobalBinary)
- 查询时计算的字段(计算的COMPUTECODE或瞬态)
- 是列表集合的字段(具有LogicalToOdbc转换)
- 一个执行LogicalToOdbc转换并且不是数据类型%Date, %Time或%PosixTime的字段
- 覆盖了LogicalToOdbc转换代码的字段
- 执行LogicalToStorage转换的字段
- 地图数据输入使用检索码的字段
- 一个其地图数据条目具有定界符的字段(不是%List存储)
- 映射到一块嵌套存储的字段

索引限制: 如果选择项目列表仅由%ID字段和/或映射到同一索引的字段组成,则不使用快速选择。

如果使用“快速选择”执行查询,则在启用了%System/%SQL / XDBCStatement的情况下,在审计数据库的SELECT审计事件中会标记此事实。

查询和企业缓存协议Enterprise Cache Protocol (ECP)

使用企业缓存协议(ECP)的InterSystems IRIS实现(例如分布式缓存群集)可以同步查询结果。ECP是一种分布式数据缓存结构,用于管理服务器系统的异构网络之间的数据分布和锁定。

如果ECP同步处于活动状态,则每次执行SELECT语句时,InterSystems IRIS都会将所有未决的ECP请求强制发送到数据服务器。完成,这可以确保客户端缓存是同步的。此同步发生在查询的“打开”逻辑中。如果是游标查询,则在OPEN游标执行中。

要激活ECP同步,请使用%SYSTEM.SQL类的SetECPSync()方法。若要确定当前设置,请调用\$SYSTEM.SQL.CurrentSettings()。

[#SQL](#) [#Caché](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B9%9D%E7%AB%A0-sql%E6%9F%A5%E8%AF%A2%E6%95%B0%E6%8D%AE%E5%BA%93%EF%BC%88%E4%BA%8C%EF%BC%89>