文章
姚 鑫 · 三月 16, 2021 阅读大约需 5 分钟

第十一章 SQL隐式联接（箭头语法）

第十一章 SQL隐式联接（箭头语法）

InterSystems SQL提供了一个特殊的->运算符，作为从相关表中获取值的快捷方式，而在某些常见情况下无需指定显式的JOIN即可。可以使用此箭头语法代替显式联接语法，也可以将其与显式联接语法结合使用。箭头语法执行左外部联接。

箭头语法可用于类的属性或父表的关系属性的引用。其他类型的关系和外键不支持箭头语法。不能在ON子句中使用箭头语法（->）。

属性引用

可以使用- >操作符作为从“ 引用表 ”获取值的简写。
例如，假设定义了两个类:Company:

```
Class Sample.Company Extends %Persistent [DdlAllowed]
{
/// The Company name
Property Name As %String;
}
```

Employee:

```
Class Sample.Employee Extends %Persistent [DdlAllowed]
{
/// The Employee name
Property Name As %String;

/// The Company this Employee works for
Property Company As Company;
}
```

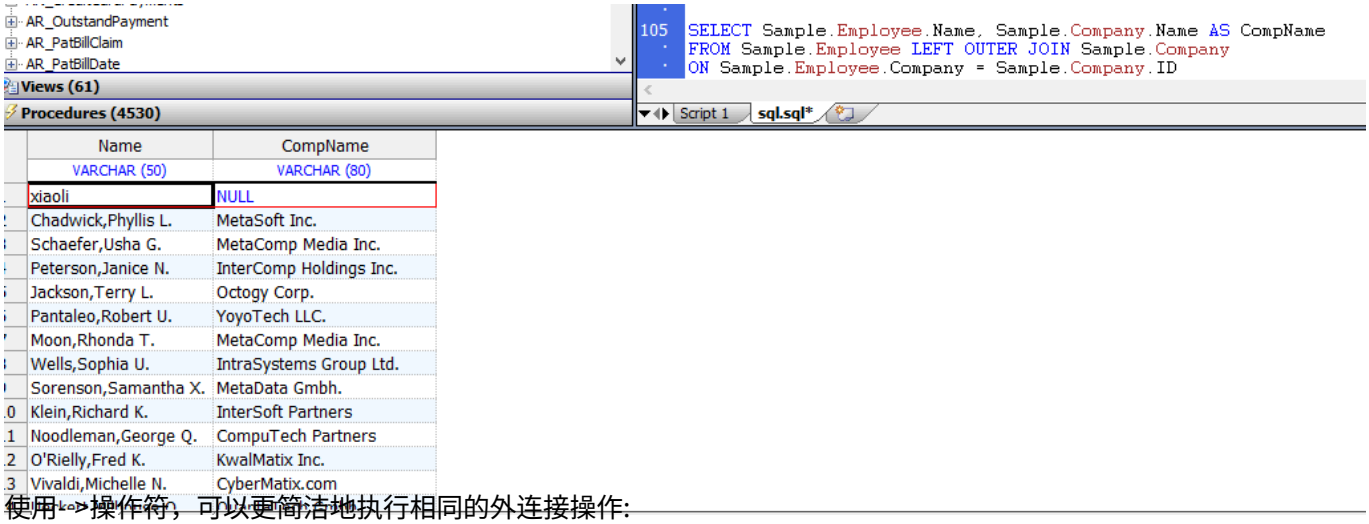
Employee类包含一个属性，该属性是对Company对象的引用。
在基于对象的应用程序中，可以使用点语法遵循此引用。
例如，要查找Employee工作的Company名称:

```
 Set name = employee.Company.Name
```

可以使用使用外部连接来连接Employee和Company表的SQL语句来执行相同的任务:

```
SELECT Sample.Employee.Name, Sample.Company.Name AS CompName
```

```
FROM Sample.Employee LEFT OUTER JOIN Sample.Company
ON Sample.Employee.Company = Sample.Company.ID
```



	Name	CompName
	VARCHAR (50)	VARCHAR (80)
	xiaoli	NULL
	Chadwick,Phyllis L.	MetaSoft Inc.
	Schaefer,Usha G.	MetaComp Media Inc.
	Peterson,Janice N.	InterComp Holdings Inc.
	Jackson,Terry L.	Octogy Corp.
	Pantaleo,Robert U.	YoyoTech LLC.
	Moon,Rhonda T.	MetaComp Media Inc.
	Wells,Sophia U.	IntraSystems Group Ltd.
	Sorenson,Samantha X.	MetaData Gmbh.
0	Klein,Richard K.	InterSoft Partners
1	Noodleman,George Q.	CompuTech Partners
2	O'Rielly,Fred K.	KwalMatix Inc.
3	Vivaldi,Michelle N.	CyberMatix.com

使用->操作符，可以更简洁地执行相同的外连接操作:

```
SELECT Name, Company->Name AS CompName
FROM Sample.Employee
```

只要在表中有引用列，就可以使用->运算符；也就是说，其列的值是被引用表的ID（本质上是外键的特殊情况）。在这种情况下，Sample.Employee的Company字段包含Sample.Company表中记录的ID。可以在可以在查询中使用列表达式的任何地方使用->运算符。例如，在WHERE子句中：

```
SELECT Name,Company AS CompID,Company->Name AS CompName
FROM Sample.Employee
WHERE Company->Name %STARTSWITH 'G'
```

使用->运算符，可以更简洁地执行相同的OUTER JOIN操作：

这等效于：

```
SELECT E.Name,E.Company AS CompID,C.Name AS CompName
FROM Sample.Employee AS E, Sample.Company AS C
WHERE E.Company = C.ID AND C.Name %STARTSWITH 'G'
```

请注意，在这种情况下，此等效查询使用INNER JOIN。

以下示例使用箭头语法访问Sample.Person中的“ Spouse ”字段。如示例所示，Sample.Employee中的Spouse字段包含Sample.Person中记录的ID。本示例返回Employee与其Spouse的HomeState相同的HomeState或OfficeState的那些记录：

```
SELECT Name,Spouse,Home_State,Office_State,Spouse->Home_State AS SpouseState
FROM Sample.Employee
WHERE Home_State=Spouse->Home_State OR Office_State=Spouse->Home_State
```

可以在GROUP BY子句中使用->运算符：

```
SELECT Name,Company->Name AS CompName
FROM Sample.Employee
GROUP BY Company->Name
```

可以在ORDER BY子句中使用–>运算符：

```
SELECT Name,Company->Name AS CompName
FROM Sample.Employee
ORDER BY Company->Name
```

或在ORDER BY子句中为–>运算符列引用列别名：

```
SELECT Name,Company->Name AS CompName
FROM Sample.Employee
ORDER BY CompName
```

支持复合箭头语法，如以下示例所示。在此示例中，Cinema.Review表包含 “ Film ” 字段，其中包含Cinema.Film表的行ID。 Cinema.Film表包含Category字段，其中包含Cinema.Category表的行ID。因此，Film-> Category-> CategoryName访问以下三个表，以返回具有ReviewScore的每部电影的CategoryName：

```
SELECT ReviewScore,Film,Film->Title,Film->Category,Film->Category->CategoryName
FROM Cinema.Review
ORDER BY ReviewScore
```

子表引用

可以使用–>运算符来引用子表。例如，如果LineItems是Orders表的子表，则可以指定：

```
SELECT LineItems->amount
FROM Orders
```

请注意，在Orders中没有称为LineItems的属性。
LineItems是包含数量字段的子表的名称。该查询在结果集中为每个Order行生成多个行。它等效于：

```
SELECT L.amount
FROM Orders O LEFT JOIN LineItems L ON O.id=L.custorder
```

其中ustust是LineItems表的父引用字段。

箭头语法权限

使用箭头语法时，必须对两个表中的引用数据都具有SELECT权限。必须在被引用的列上具有表级SELECT权限或列级SELECT权限。使用列级权限，需要对被引用表以及被引用列的ID具有SELECT权限。

以下示例演示了所需的列级权限：

```
SELECT Name,Company->Name AS CompanyName
FROM Sample.Employee
GROUP BY Company->Name
ORDER BY Company->Name
```

在上面的示例中，必须对Sample.Employee.Name，Sample.Company.Name和Sample.Company.ID具有列级SELECT权限：

```
// d ##class(PHA.TEST.SQL).arrow()
ClassMethod arrow()
{
    SET tStatement = ##class(%SQL.Statement).%New()
    SET privchk1="%CHECKPRIV SELECT (Name,ID) ON Sample.Company"
    SET privchk2="%CHECKPRIV SELECT (Name) ON Sample.Employee"
CompanyPrivTest
    SET qStatus = tStatement.%Prepare(privchk1)
    IF qStatus'=1 {
        WRITE "%Prepare ??:"
        DO $System.Status.DisplayError(qStatus)
        QUIT
    }
    SET rset = tStatement.%Execute()
    IF rset.%SQLCODE=0 {
        WRITE !,"??Company??",!
    } ELSE {
        WRITE !,"???: SQLCODE=",rset.%SQLCODE,!
    }
EmployeePrivTest
    SET qStatus = tStatement.%Prepare(privchk2)
    IF qStatus'=1 {
        WRITE "%Prepare ??:"
        DO $System.Status.DisplayError(qStatus)
        QUIT
    }
    SET rset = tStatement.%Execute()
    IF rset.%SQLCODE=0 {
        WRITE !,"??Employee??",!
    } ELSE {
        WRITE !,"???: SQLCODE=",rset.%SQLCODE
    }
}
```

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

??Company??

??Employee??
```

#SQL #Caché #InterSystems IRIS #InterSystems IRIS for Health

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E4%B8%80%E7%AB%A0-sql%E9%9A%90%E5%BC%8F%E8%81%94%E6%8E%A5%EF%BC%88%E7%AE%AD%E5%A4%B4%E8%AF%AD%E6%B3%95%EF%BC%89