

文章

[姚鑫](#) · 三月 16, 2021 阅读大约需分钟

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, 而在某些常见情况无需定义显式的 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姓父表的关系属性。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

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;

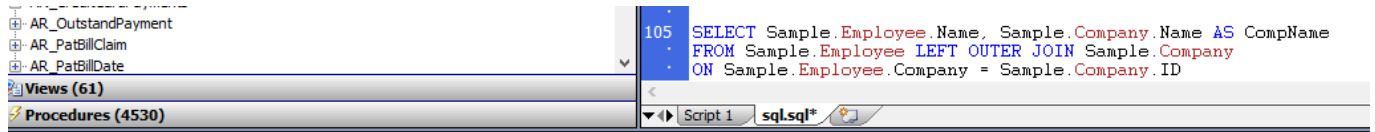
  /// The Company this Employee works for
  Property Company As Company;
}
```

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
1	Chadwick,Phyllis L.	MetaSoft Inc.
2	Schaefer,Usha G.	MetaComp Media Inc.
3	Peterson,Janice N.	InterComp Holdings Inc.
4	Jackson,Terry L.	Octogy Corp.
5	Pantaleo,Robert U.	YoyoTech LLC.
6	Moon,Rhonda T.	MetaComp Media Inc.
7	Wells,Sophia U.	IntraSystems Group Ltd.
8	Sorenson,Samantha X.	MetaData Gmbh.
9	Klein,Richard K.	InterSoft Partners
10	Noodleman,George Q.	CompuTech Partners
11	O'Rielly,Fred K.	KwalMatix Inc.
12	Vivaldi,Michelle N.	CyberMatix.com

使用->操作符,可以更简洁地执行相同连接操作:

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



	Name	CompName
	VARCHAR (50)	VARCHAR (80)
1	xiaoli	NULL
2	Chadwick,Phyllis L.	MetaSoft Inc.
3	Schaefer,Usha G.	MetaComp Media Inc.
4	Peterson,Janice N.	InterComp Holdings Inc.
5	Jackson,Terry L.	Octogy Corp.
6	Pantaleo,Robert U.	YoyoTech LLC.
7	Moon,Rhonda T.	MetaComp Media Inc.
8	Wells,Sophia U.	IntraSystems Group Ltd.
9	Sorenson,Samantha X.	MetaData Gmbh.
10	Klein,Richard K.	InterSoft Partners
11	Noodleman,George Q.	CompuTech Partners

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'
```

The screenshot shows the SQL Editor interface. The query window contains the following SQL code:

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

The Results window displays the following data:

	Name	CompID	CompName
	VARCHAR (50)	INTEGER	VARCHAR (80)
1	Kovalev, Wilma T.	15	GigaTel LLC.
2	Alton, Phil T.	89	GlobaMatix Gmbh.
3	Kovalev, Edward Z.	60	GigaCalc Media Inc.
4	Orwell, Wolfgang G.	60	GigaCalc Media Inc.
5	Underman, Maria P.	18	GlobaDynamics Gmbh.

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 Home_State 相同的 Home_State 或 Office_State 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
```

The screenshot shows the SQL Editor interface. The query window contains the following SQL code:

```
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
```

The Results window displays the following data:

	Name	Spouse	Home_State	Office_State	SpouseState
	VARCHAR (50)	INTEGER	VARCHAR (2)	VARCHAR (2)	VARCHAR (2)
1	Wijnschenk, Lydia G.	52	NV	MT	MT
2	Kovalev, Wilma T.	25	WY	NH	WY
3	Frith, Dick E.	32	OH	PA	PA

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
```

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

	Name	CompName
	VARCHAR (50)	VARCHAR (80)
1	xiaoli	NULL
2	Chadwick,Phyllis L.	METASOFT INC.
3	Schaefer,Usha G.	METACOMP MEDIA INC.
4	Peterson,Janice N.	INTERCOMP HOLDINGS INC.
5	Jackson,Terry L.	OCTOGY CORP.
6	Pantaleo,Robert U.	YOYOTECH LLC.
7	Wells,Sophia U.	INTRASYSTEMS GROUP LTD.
8	Sorenson,Samantha X.	METADATA GMBH.
9	Klein,Richard K.	INTERSOFT PARTNERS
10	Noodleman,George Q.	COMPUTECH PARTNERS
11	O'Rielly,Fred K.	KWALMATIX INC.

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
```

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

	Name	CompName
	VARCHAR (50)	VARCHAR (80)
1	xiaoli	NULL
2	Feynman,Jeff H.	AccuCalc Gmbh.
3	Fives,Kristen F.	AccuCalc Gmbh.
4	Ingrahm,Susan N.	AccuLateral Holdings Inc.
5	Murray,Zelda X.	AccuLateral Holdings Inc.
6	Xerxes,Jose S.	BioData Group Ltd.
7	Browne,Robert X.	Biogy Holdings Inc.
8	Joyce,Elmo R.	BioWare LLC.
9	Nelson,Paul O.	CompuDynamics Gmbh.

或在ORDER BY子句中为->运算符引用列别名:

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

支持复合表头语法, 如以示例所示。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访问以三个表, 以返回具有ReviewScore的每部电影的电影的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行生成一行。它等效于:

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

其中custorder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。必须在被引用列上具有表级SELECT权限或列级SELECT权限。使用列级权限,需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>