

文章
姚鑫 · 三月 12, 2021 阅读大约需分钟

第九章 SQL查询数据库(一)

第九章 SQL查询数据库

查询类型

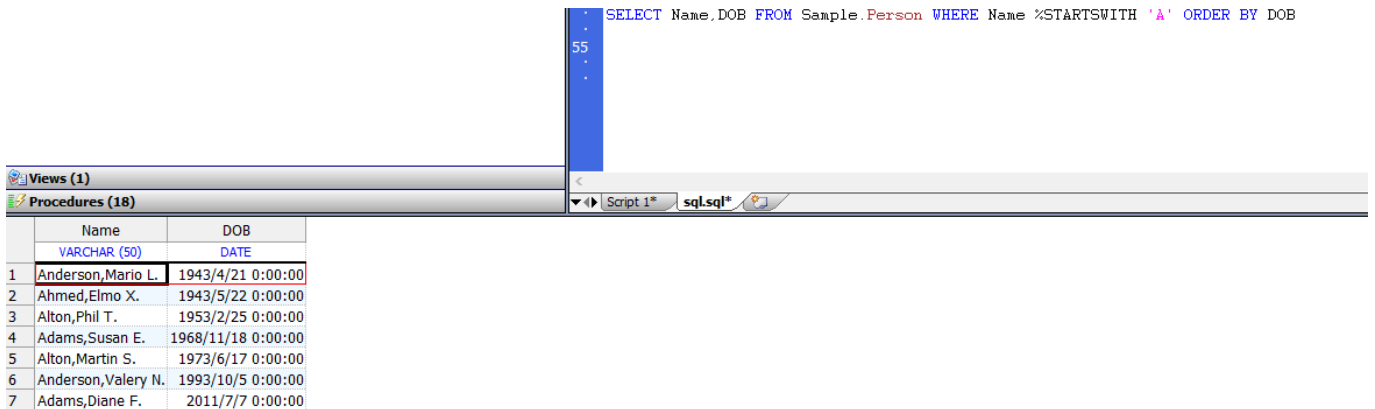
查询是执行数据检索并生成结果集的句子。查询可以包含以下任意项：

- 一个简单的SELECT语句,用于访问指定表或视图中的数据。
- 具有JOIN语法的SELECT语句,用于访问多个表或视图中的数据。
- 合并多个SELECT语句的结果的UNION语句。
- 使用SELECT语句为封闭的SELECT查询提供单个数据项的子查询。
- 在嵌入式SQL中,这是一个使用SQL游标通过FETCH语句访问行数据的SELECT语句。

使用SELECT语句

SELECT语句从一个或多个表或视图中选择一行或行数据。下面的示例显示了一个简单的SELECT：

```
SELECT Name,DOB FROM Sample.Person WHERE Name %STARTSWITH 'A' ORDER BY DOB
```



在此示例Name和DOB是Sample.Person表中的列(数据字段)。

在SELECT语句中必须指定子句的顺序是：SELECT DISTINCT TOP ... 选择项INTO ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY。这是命令语法顺序。所有这些子句都是可选的，但SELECT选择项除外。

SELECT子句的执行顺序

可以通过注意SELECT语句的语义处理顺序(与SELECT语法顺序不同)来理解SELECT语句的操作。

SELECT的子句按以下顺序处理：

1. FROM子句-指定一个表,一个视图,多个表或使用JOIN语法的视图或一个子查询。
2. WHERE子句-限制使用某种条件选择的数据。
3. GROUP BY子句-将所选数据组织为具有匹配值的子集;每个值仅返回一条记录。

4. HAVING子句—限制使用各种条件从组中选择数据。
5. select-item
—从指定的表或视图中选择一个数据字段。选择项也可以引用也可以不引用特定数据字段的表达式。
6. DISTINCT子句—应用于SELECT结果集，它将返回的行限制为包含不同(非重复)值的行。
7. ORDER BY子句—应用于SELECT结果集，它按指定字段对按排序顺序返回行进行排序。

这种语义顺序表明，所有子句都可以识别表别名(在FROM子句中定义)，而列别名(在SELECT选择项中定义)只能由ORDER BY子句识别。

要在其SELECT子句中使用列别名，可以使用子查询，如以下示例所示：

```
SELECT Interns FROM  
    (SELECT Name AS Interns FROM Sample.Employee WHERE Age<21)  
WHERE Interns %STARTSWITH 'A'
```

选择字段

当发出SELECT时，InterSystems SQL会尝试将每个指定的select-item字段名称与对应于指定表的类中定义的属性行匹配。每个类属性都有一个属性称和一个SqlFieldName。如果使用SQL定义表，则在CREATE TABLE命令中指定的字段名称为SqlFieldName，并且InterSystems IRIS从SqlFieldName生成属性称。

字段名称，类属性称和SqlFieldName名称具有不同的命名约定：

- SELECT语句中的字段名称不区分大小写。SqlFieldName名称和属性称区分大小写。
- 遵循标识符命名约定，SELECT语句中的字段名称和SqlFieldName名称可以包含某些非字母数字字符。属性称只能包含字母数字字符。生成属性称时，InterSystems IRIS会去除非字母数字字符。InterSystems IRIS可能必须附加一个字符以创建唯一的属性称。

字段的这三个名称之间的转换确定了查询行为的几个方面。可以使用字母大小写的任意组合来指定选择项目的字段名称，并且InterSystems SQL将标识相应的属性结果集显示中的数据列标题名称是SqlFieldName，而不是select-item中指定的字段名称。这被认为是数据列标题的字母大小写可能与select-item字段名称不同的原因。

可以为选择项字段指定列别名。列别名可以采用字母大小写的任何混合形式，并且可以遵循标识符命名约定包含非字母数字字符。可以使用字母大小写的任意组合来引用列别名(例如，在ORDER BY子句中)，并且InterSystems SQL解析为select-item字段中指定的字母大小写。InterSystems IRIS始终尝试匹配列别名列表，然后再尝试匹配对应于已定义字段的属性表。如果定义了列别名，则结果集显示中的数据列标题名称是指定字母大小写的列别名，而不是SqlFieldName。

SELECT查询完成后，InterSystems SQL会为该查询生成结果集类。结果集类包含一个与每个选定字段相对应的属性。如果SELECT查询包含重复的字段名称，则系统将通过附加字符为查询中字段的每个实例生成唯一的属性称。因此，查询中不能包含36个以上相同字段的实例。

查询生成的结果集类还包含列别名的属性以避免字母大小写解析的歧义，在引用列别名时应使用与在SELECT语句中指定列别名时使用的字母大小写相同的字母大小写。

除了用户指定的列别名外，InterSystems SQL还会自动为每个字段名称生成三个别名，这些别名与该字段名称的常见字母大小写变体对应。这些生成别名对用户不可见。提供它们是出于性能方面的考虑，因为通过别名访问属性通过字母大小写转换解决字母大小写的速度更快。例如，如果SELECT指定FAMILYNAME，并且对应的属性是familyname，则InterSystems SQL使用生成别名(FAMILYNAME AS familyname)来解析字母大小写。但是，如果SELECT指定familyname且对应的属性Familyname，则InterSystems SQL必须使用较慢的字母大小写转换过程来解析字母大小写。

选择项还可以是表达式，聚合函数，子查询，用户定义的函数(如星号)或其他某个值。

JOIN操

JOIN提供了一种将一个表中的数据链接到另一个表中的数据的方法，并且经常用于定义报告和查询。在SQL中，JOI

N是一种操作，它将来自两个表的数据合并以产生第三个表，但要遵守限制条件。结果表的每一行都必须满足限制条件。

InterSystems SQL支持五种类型的联接(有些具有种语法形式): CROSS JOIN, INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN和FULL OUTER JOIN。外部联接通过种条件表达式谓词和逻辑运算符支持ON子句。对NATURAL外部联接和带有USING子句的外部联接有部分支持。

如果查询包含联接,则该查询中的所有字段引用都必须具有附加的表别名。由于InterSystems IRIS在数据列标题名称中不包含表别名,因此可能需为select-item字段提供列别名,以澄清哪个表是数据源。

以下示例使用联接操作将Sample.Person中的“fake”随机分配)邮政编码与Sample.USZipCode中的真实邮政编码和城市名称进行匹配。之所以提供WHERE子句,是因为USZipCode不包括所有可能的5位邮政编码:

```
SELECT P.Home_City,P.Home_Zip AS FakeZip,Z.ZipCode,Z.City AS ZipCity,Z.State
FROM Sample.Person AS P LEFT OUTER JOIN Sample.USZipCode AS Z
ON P.Home_Zip=Z.ZipCode
WHERE Z.ZipCode IS NOT NULL
ORDER BY P.Home_City
```

选择大量字段的查询

一个查询不能选择超过1,000个选择项字段。

选择超过150个选择项字段的查询可能有性能考虑。InterSystems IRIS自动生成结果列别名。这些生成别名是为没有用户定义别名的字段名提供的,以便解决字母大小写的变化。使用别名的字母大小写解析明显快于逐个字母大小写。但是,生成结果列别名的数量限制为500个。因为通常InterSystems IRIS会为每个字段生成其中的三个别名(针对三种最常见的字母大小写变化),所以系统会为查询中大约前150个指定字段生成别名。因此,引用少于150个字段的查询通常比引用更多字段的查询具有更好的结果性能。通过在非常大的查询中为每个字段select-item指定一个精确的列别名(例如,SELECT FamilyName AS FamilyName),然后确保按列别名引用结果项时使用相同的字母大小写,可以避免此性能问题。

定义和执行命名查询

可以按如下方式定义和执行命名查询:

- 使用CREATE QUERY定义查询。该查询被定义为一个存储过程,可以使用CALL执行。
- 定义类查询(在类定义中定义的查询)。类查询被影射为存储过程。可以用CALL执行。也可以使用%SQL准备一个类查询。语句%PrepareClassQuery()方法,然后使用%Execute()方法执行。

创建查询和调用

可以使用CREATE QUERY定义查询,然后使用CALL按名称执行查询。在以下示例中,第一个是定义查询AgeQuery的SQL程序,第二个是执行查询的Dynamic SQL:

```
/// d ##class(PHA.TEST.SQL).Query()
ClassMethod Query()
{
    &sql(
        CREATE QUERY Sample.AgeQuery(IN topnum INT DEFAULT 10,IN minage INT 20)
        PROCEDURE
        BEGIN
            SELECT TOP :topnum Name,Age FROM Sample.Person
            WHERE Age > :minage
            ORDER BY Age ;
```

```

        END
    )
    SET mycall = "CALL Sample.AgeQuery(11,65)"
    SET tStatement = ##class(%SQL.Statement).%New()
    SET qStatus = tStatement.%Prepare(mycall)
    IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
    SET rset = tStatement.%Execute()
    DO rset.%Display()
    &sql(DROP QUERY Sample.AgeQuery)
}

```

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

Dumping result #1

```

Name      Age
Ingrahm, Yan S.    66
Hertz, Uma C.     66
Zweifelhofer, Zelda J.  67
Zampitello, Josephine Q. 67
Xiang, Molly F.   67
Davis, Jane E.    67
Vanzetti, Alexandra O. 67
Solomon, Emily D. 68
Isaacs, Elvis V. 68
Alton, Phil T.   68
Yeats, Debby G. 69

```

11 Rows(s) Affected

类查询

可以在类中定义查询。该类可以是%Persistent类，但不是必须的。该类查询可以引用在同一类或同一命名空间中的另一类中定义的数据。编译包含查询的类时，在类查询中引用的表、字段和其他数据实体必须存在。

编译包含查询的类时，不会编译该类查询，而是在第一次执行SQL代码(运行时)时进行类查询的编译。当使用%PrepareClassQuery()方法在Dynamic SQL中准备查询时，会发生这种情况。第一次执行定义了一个可执行的缓存查询。

以下类定义示例定义了一个类查询：

```

/// DO ##class(%ResultSet).RunQuery("PHA.TEST.SQL", "MyQ", "MO")
Query MyQ(Myval As %String) As %SQLQuery(CONTAINID = 1, ROWSPEC = "Name,Home_State")
[ SqlProc ]
{
    SELECT Name,Home_State FROM Sample.Person
    WHERE Home_State = :Myval ORDER BY Name
}

```

```
DHC-APP>DO ##class(%ResultSet).RunQuery("PHA.TEST.SQL", "MyQ", "MO")
```

```

Name:Home_State:
Burroughs,Barbara H.:MO:
Emerson,Edgar T.:MO:

```

Frost,Xavier D.:MO:
Joyce,Elmo R.:MO:
King,Dmitry G.:MO:
Klingman,Rhonda G.:MO:
Kratzmann,Emily Z.:MO:
Martinez,Emilio G.:MO:
Schaefer,Usha G.:MO:

下面的示例执行上一示例中的Sample.QClass中定义的MyQ查询:

```
/// d ##class(PHA.TEST.SQL).Query1()  
ClassMethod Query1()  
{  
    SET Myval="NY"  
    SET stmt=##class(%SQL.Statement).%New()  
    SET status = stmt.%PrepareClassQuery("PHA.TEST.SQL","MyQ")  
    IF status'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(status) QUIT}  
    SET rset = stmt.%Execute(Myval)  
    DO rset.%Display()  
    WRITE !,"End of data"  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQL).Query1()
```

```
Dumping result #1  
Name      Home_State  
Chadbourne,Danielle G.  NY  
Eastman,Clint G.        NY  
Pape,Linda M.           NY  
Peterson,Janice N.      NY  
Schaefer,Jocelyn V.     NY
```

```
5 Rows(s) Affected  
End of data
```

以动态SQL示例使用%SQL.Statement执行在Sample.Person类中定义的ByName查询,并传递一个字符串以将返回的名称限制为以该字符串值开头的名称:

```
/// d ##class(PHA.TEST.SQL).Query2()  
ClassMethod Query2()  
{  
    SET statemt=##class(%SQL.Statement).%New()  
    SET cqStatus=statemt.%PrepareClassQuery("Sample.Person","ByName")  
    IF cqStatus'=1 {WRITE "%PrepareClassQuery failed:" DO $System.Status.DisplayError(cqStatus) QUIT}  
    SET rs=statemt.%Execute("L")  
    DO rs.%Display()  
}
```

```
DHC-APP>d ##class(PHA.TEST.SQL).Query2()
```

Dumping result #1

ID	Name	DOB	SSN		
146	LaRocca,David X.		42013	603-23-8852	
54	Larson,Nataliya Z.		52896	277-65-7763	
65	Lee,Zoe Z.	62253	548-60-1784		
105	Leiberman,Nataliya F.		46706	624-13-9765	
56	Lennon,Chelsea T.		54537	190-51-5484	
106	Lennon,Imelda Z.		57911	594-60-9044	
137	Lennon,Maureen M.		38392	746-77-6520	
178	Lepon,Janice T.	45675	188-86-7267		
29	Lepon,Jeff Z.	37144	212-43-4979		
112	Lepon,Kevin N.	31575	929-85-8355		
154	Lopez,Ralph W.	45541	391-39-9235		
77	Love,Janice E.	33050	515-29-7228		

12 Rows(s) Affected

[#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%B8%80%EF%BC%89>