

### 文章

姚鑫 · 十月 15, 2021 阅读大约需 8 分钟

## 第四十六章 SQL命令 FROM (二)

[toc]

## 第四十六章 SQL命令 FROM (二)

### %PARALLEL

这个可选关键字在查询的FROM子句中指定。

它建议 IRIS使用多个处理器(如果适用)并行处理查询。

这可以显著提高使用一个或多个COUNT、SUM、AVG、MAX或MIN聚合函数和/或GROUP BY子句的某些查询的性能，以及许多其他类型的查询。

这些通常是处理大量数据并返回小结果集的查询。

例如，SELECT AVG(SaleAmt) FROM %PARALLEL User.AllSales GROUP BY Region使用并行处理。

既指定单个字段又指定聚合函数且不包含GROUP BY子句的查询不能执行并行处理。

例如，SELECT Name,AVG(Age) FROM %PARALLEL Sample.

Person不执行并行处理，而是从SELECT Name,AVG(Age) FROM %PARALLEL Sample.Person GROUP BY HomeState执行并行处理。

%PARALLEL用于SELECT查询及其子查询。

INSERT命令子查询不能使用%PARALLEL。

指定%PARALLEL可能会降低某些查询的性能。

在具有多个并发用户的系统上使用%PARALLEL运行查询可能会导致整体性能下降。

注意:指定%PARALLEL的查询必须在读/写而不是只读的数据库中运行。

否则，可能发生<PROTECT>错误。

不管在FROM子句中是否存在%PARALLEL关键字，有些查询可能使用线性处理，而不是并行处理:有些查询不支持并行处理;

一些优化后的查询可能无法从并行处理中获益。

可以使用Show Plan确定 IRIS是否以及如何对查询进行了并行处理分区。

要确定当前系统上的处理器数量，使用 %SYSTEM.Util.NumberOfCPUs()方法。

### %STARTTABLE

这个可选关键字指定查询优化器应该开始对FROM子句中列出的第一个表执行联接。

其余表的连接顺序留给查询优化器。

将此关键字与%INORDER进行比较，后者指定了完整的连接顺序。

%STARTTABLE不能与交叉连接或右外连接一起使用。

不能使用%STARTTABLE(或%FIRSTTABLE)从左OUTER join(或右OUTER join)的左边开始连接顺序。

如果指定的开始表与外部连接的要求不一致，则会生成一个SQLCODE -34错误:“优化器未能找到可用的连接顺序。”

为了避免这种情况，当与外部连接一起使用时，建议%STARTTABLE只与ansi风格的左外部连接或完整外部连接一起使用。

下表显示了在使用%INORDER和%STARTTABLE优化组合超查询父视图和内联视图时的合并行为:

""	没有连接优化器的超查询	具有%STARTTABLE的超级 有%INORDER的超级查询 查询
不带连接优化器的视图	如果可能，合并视图	如果视图是超查询start: don't 合并如果可能的话;视图的底 merge。否则，如果可能， 层表是无序的。 合并视图。
使用%STARTTABLE查看	不合并	如果视图是超级查询start: m 不合并 erge，如果可能的话。视图 的开始表变成了超级查询的 开始表。否则,不合并。
使用%INORDER查看	不合并	不合并 如果视图不是由%INORDER 控制的，则不要合并。否则 ，如果可能，合并视图;视图 的顺序被替换为超级查询连 接顺序。

%FIRSTTABLE提示在功能上与%STARTTABLE相同，但是提供了以任意顺序指定连接表序列的灵活性。

## FROM子句中的表值函数

表值函数是一个类查询，它被投影为一个存储过程，并返回单个结果集。

表值函数是任何具有SqlProc TRUE的类查询。

用作表值函数的类查询必须在LOGICAL或RUNTIME模式下编译。

当作为表值函数使用并在RUNTIME模式下编译时，表值函数查询将在LOGICAL模式下调用。

表值函数遵循与类查询的存储过程名称相同的命名约定。

参数括号是必须的;

括号可以是空的，可以包含一个字面值或一个主机变量，也可以包含一个用逗号分隔的字面值 and 主机变量列表。

如果不指定参数(空括号或空字符串)，表值函数将返回所有数据行。

要使用表值函数发出查询，用户必须对定义表值函数的存储过程拥有EXECUTE权限。

用户还必须对表值函数查询访问的表或视图具有SELECT权限。

在下面的示例中，类查询Sample.Person.ByName被投影为一个存储过程，因此可以用作表值函数:

```
SELECT Name,DOB FROM Sample.SP_Sample_By_Name('A')
```

下面的动态SQL示例指定相同的表值函数。它使用%Execute()方法将参数值提供给？入参：

```
ClassMethod From()
{
    s myquery="SELECT Name,DOB FROM Sample.SP_Sample_By_Name(?)"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(myquery)
    if qStatus'=1 {w "%Prepare failed:" d $System.Status.DisplayError(qStatus) q}
    s rset = tStatement.%Execute("A")
    d rset.%Display()
    w !,"End of A data",!!
    s rset = tStatement.%Execute("B")
    d rset.%Display()
    w !,"End of B data"
}
```

表值函数只能在SELECT语句或DECLARE语句的FROM子句中使用。表值函数名可以用模式名限定，也可以用非限定名(没有模式名)限定；非限定名使用默认模式。在SELECT语句FROM子句中，只要可以使用表名，就可以使用表

值函数。它可以在视图或子查询中使用，并且可以使用逗号分隔的列表或显式联接语法与其他表引用项联接。

表值函数不能直接用于INSERT、UPDATE或DELETE语句。但是，可以为这些命令指定子查询，以指定表值函数。

SQL没有为表值函数定义EXTENTSIZE，也没有为表值函数列定义SELECTIVITY。

## FROM子句中的子查询

可以在FROM子句中指定子查询。

这称为流子查询。

子查询被视为与表相同的处理方式，包括它在JOIN语法中的使用以及使用as关键字可选地分配别名。

FROM子句可以以任何组合包含多个表、视图和子查询，但要受JOIN语法的限制，如JOIN中所述。

```
SELECT name,region
FROM (SELECT t1.name,t1.state,t2.region
      FROM Employees AS t1 LEFT OUTER JOIN Regions AS t2
      ON t1.state=t2.state)
GROUP BY region
```

子查询可以指定TOP子句。

当与TOP子句配对时，子查询可以包含ORDER BY子句。

子查询可以使用SELECT \*语法，但有以下限制:因为FROM子句的结果是值表达式，所以包含SELECT \*的子查询只能生成一行。

子查询中的连接不能是NATURAL连接或接受USING子句。

## 从子查询和%VID

当调用FROM子查询时，它为返回的每个子查询行返回一个%VID。

%VID是一个整数计数器字段;

它的值是系统分配的、唯一的、非空的、非零的、不可修改的。

%VID仅在显式指定时返回。

它以数据类型INTEGER返回。

因为%VID值是顺序整数，所以如果子查询返回的是顺序数据，则它们更有意义;

子查询只能在与TOP子句配对时使用ORDER BY子句。

因为%VID是一个顺序整数，所以可以用它来确定带有ORDER BY子句的子查询中项目的排名。

在下面的示例中，10条最新的记录按名称顺序列出，但是使用%VID值可以很容易地看到它们的时间戳排名:

```
SELECT Name,%VID,TimeStamp FROM
  (SELECT TOP 10 * FROM MyTable ORDER BY TimeStamp DESC)
ORDER BY Name
```

%VID的一个常见用途是“window”结果集，将执行划分为符合显示窗口中可用行数的顺序子集。

例如，显示20条记录，然后等待用户按Enter键，然后显示下20条记录。

```
ClassMethod From1()
{
    s myq=4
    s myq(1)="SELECT %VID,* "
```

```

s myq(2)="FROM (SELECT TOP 60 Name, Age FROM Sample.Person "
s myq(3)="WHERE Age > 55 ORDER BY Name) "
s myq(4)="WHERE %VID BETWEEN ? AND ?"
s tStatement = ##class(%SQL.Statement).%New()
s qStatus = tStatement.%Prepare(.myq)
if qStatus'=1 {w "%Prepare failed:" d $System.Status.DisplayError(qStatus) q}
for i=1:10:60 {
    s rset = tStatement.%Execute(i, i+9)
    while rset.%Next() {
        d rset.%Print()
    }
    w !!
}
w "End of data"
}

```

```

DHC-APP>d ##class(PHA.TEST.SQLCommand).From1()
1 "Ahmed,Elmo X." 78
2 "Alton,Phil T." 68
3 "Anderson,Mario L." 78
4 "Bachman,Susan O." 88
5 "Basile,Filomena X." 87
6 "Browne,Robert X." 83
7 "Bukowski,Mario V." 86
8 "Burroughs,Barbara H." 86
9 "Cerri,Stavros Q." 96
10 "Chadbourne,Barb B." 94

```

## 可选FROM子句

如果SELECT项列表(直接或间接)没有引用表数据，则FROM子句是可选的。  
 这种SELECT可以用于从函数、运算符表达式、常量或宿主变量返回数据。  
 对于不引用表数据的查询：

- 如果省略FROM子句，则不管TOP关键字值如何，最多返回一行数据；  
 TOP 0不返回任何数据。  
 DISTINCT子句被忽略。  
 不需要特权。
- 如果指定了FROM子句，则必须指定当前命名空间中的现有表。  
 必须对该表具有SELECT权限，即使该表没有被引用。  
 除非指定了TOP或DISTINCT子句，或者用WHERE或HAVING子句限制它，否则返回的相同数据行数等于指定表中的行数。  
 指定DISTINCT子句将输出限制为单行数据。  
 TOP关键字将输出限制为TOP值指定的行数；  
 TOP 0不返回任何数据。

无论是否有FROM子句，都可以指定后续子句(如GROUP BY、HAVING或ORDER BY)。  
 WHERE或HAVING子句可用于确定是否返回结果，或返回多少相同的结果行。  
 即使没有指定FROM子句，这些子句也可以引用表。  
 可以指定GROUP BY或ORDER BY子句，但这些子句没有意义。

下面是不引用表数据的SELECT语句示例。  
 两个示例都返回一行信息。

下面的例子省略了FROM子句。  
DISTINCT关键字不是必需的，但是可以指定。  
不允许使用SELECT子句。

```
SELECT 3+4 AS Arith,  
       {fn NOW} AS NowDateTime,  
       {fn DAYNAME({fn NOW})} AS NowDayName,  
       UPPER('MixEd cASe EXPreSSioN') AS UpCase,  
       {fn PI} AS PiConstant
```

下面的示例包含一个FROM子句。  
DISTINCT关键字用于返回单行数据。  
FROM子句表引用必须是一个有效的表。  
这里允许使用ORDER BY子句，但没有意义。  
注意，ORDER BY子句必须指定一个有效的选择项别名：

```
SELECT DISTINCT 3+4 AS Arith,  
       {fn NOW} AS NowDateTime,  
       {fn DAYNAME({fn NOW})} AS NowDayName,  
       UPPER('MixEd cASe EXPreSSioN') AS UpCase,  
       {fn PI} AS PiConstant  
FROM Sample.Person  
ORDER BY NowDateTime
```

下面的例子都使用了WHERE子句来决定是否返回结果。  
第一个包含FROM子句，并使用DISTINCT关键字返回单行数据。  
第二个省略了FROM子句，因此最多返回一行数据。  
在这两种情况下，WHERE子句表引用必须具有SELECT权限的有效表：

```
SELECT DISTINCT  
       {fn NOW} AS DataOKDate  
FROM Sample.Person  
WHERE FOR SOME (Sample.Person)(Name %STARTSWITH 'A')
```

```
SELECT {fn NOW} AS DataOKDate  
WHERE FOR SOME (Sample.Person)(Name %STARTSWITH 'A')
```

[#SQL #Caché](#)

---

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E5%8D%81%E5%85%AD%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-%EF%BC%88%E4%BA%8C%EF%BC%89>