

文章

[姚鑫](#) · 十一月 9, 2021 阅读大约需分钟

第七章 SQL命令 SELECT(三)

第七章 SQL命令 SELECT(三)

列别名

指定SELECT-ITEM时,可以使用AS关键字指定列名的别名:

```
SELECT Name AS PersonName, DOB AS BirthDate, ...
```

列别名在结果集中显示为列标题。指定列别名是可选的;始终提供默认值。列别名以指定的字母大小写显示;但是,当在ORDER BY子句中引用时,它不区分大小写。C别名必须是有效的标识符。C别名可以是分隔标识符。使用带分隔符的标识符允许列别名包含空格、其他标点符号或作为SQL标识名称。例如,SELECT Name AS "Customer Name" or SELECT Home_State AS "From"。

As关键字不是必需但使查询文本更易于阅读。因此,也是有效的语法:

```
SELECT Name PersonName, DOB BirthDate, ...
```

SQL不执行列别名唯一检查。

字段列和列别名可能具有相同的名称(尽管不可取),或者两个列别名相同。

当ORDER by子句引用此类非唯一列别名时,可能会导致SQLCODE -24"Ambiguous sort column"错误。

列别名与所有SQL标识符一样,不区分大小写。

其他SELECT子句中列别名的使用由查询语义处理顺序控制。

可以通过ORDER by子句中的列别名引用列。

不能在选择列表中的另一个选择项、DISTINCT BY子句、WHERE子句、GROUP BY子句或HAVING子句中引用列别名。

不能在JOIN操作中的ON子句或USING子句中引用列别名。

但是,可以使用子查询使列别名可用来供其他这些其他SELECT子句使用。

字段列别名

选择项字段名不区分大小写。

但是,除非提供列别名,否则结果集中的字段列的名称应遵循与列属性关联的SqlFieldName的字母大小写。

SqlFieldName的大小写对应于表定义中指定的字段名,而不是选择项列表中指定的字段名。

因此,SELECT name FROM Sample.Person返回字段列标签为Name。

使用字段列别名可以指定要显示的字母大小写,示例如:

```
SELECT name,name AS NAME  
FROM Sample.Person
```

字母大小写解析时间。

为了最大性能，您可以指定字段名的确切字母大小写，如表定义中所指定的那样。

但是，在表定义中确定字段的确切字母大小写通常很不方便而且容易出错。

相反，可以使用字段列别名来避免字母大小写问题。

注意，对字段列别名的所有引用必须以字母大小写匹配。

下面的动态SQL示例需要字母大小写解析(SqlFieldNames为" Latitude "和" Longitude ")：

```
ClassMethod Select()
{
  s myquery = "SELECT latitude,longitude FROM Sample.USZipCode"
  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()
  while rset.%Next() {
    w rset.latitude, " ", rset.longitude, !
  }
}
```

下面的动态SQL示例不需要区分大小写，因此执行得更快：

```
ClassMethod Select1()
{
  s myquery = "SELECT latitude AS northsouth,longitude AS eastwest FROM Sample.USZipCode"
  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()
  while rset.%Next() {
    w rset.northsouth, " ", rset.eastwest, !
  }
}
```

列名中不包含alias表别名前缀。

因此，在下面的示例中，两列都被标记为Name：

```
SELECT p.Name,e.Name
FROM Sample.Person AS p LEFT JOIN Sample.Employee AS e ON p.Name=e.Name
```

要区分指定表的查询中的列，您应该指定列别名：

```
SELECT p.Name AS PersonName,e.Name AS EmployeeName
```

```
FROM Sample.Person AS p LEFT JOIN Sample.Employee AS e ON p.Name=e.Name
```

提供列别名以使数据更容易理解。

以表中“Home_State”列为例，将其重命名为“US_State_Abbrev”。

```
SELECT Name,Home_State AS US_State_Abbrev
FROM Sample.Person
```

请注意，%ID引用特定的列，因此返回字段名(默认为ID)或指定的列别名，如面的示例所示：

```
SELECT %ID,%ID AS Ident,Name
FROM Sample.Person
```

Non-Field列别名

非字段列将自动分配一个列名。

如果没有为这些字段提供别名，SQL将提供一个唯一的列名，如“Expression_1”或“Aggregate_3”。

整数后缀指SELECT语句中指定的选择项位置(选择项列号)。

它们不是该类型字段的计数。

下面是自动分配的列名(n是一个整数)。

这些内容的顺序越来越广泛。

例如，在数字上添加号或减号将其从HostVar提升为表达式；

连接HostVar和Literal将其提升为表达式；

在子查询中指定Literal、HostVar、Aggregate或Expression将其提升为子查询：

- Literal_n: 一个伪字段变量，比如%TABLENAME，或NULL说明符。

注意%ID不是Literal_n；

它得到实际RowID字段的列名。

- HostVar_n: 主机变量。

这可能是一个字面量，如' text '，123，或空字符串("")，一个输入变量(:myvar)，或？

由文字转换的输入参数。

请注意，任何对字面量的表达式求值，如在数字后附加符号、字符串连接或算术操作，都使其成为Expression_n，提供给？

参数不受表达式求值影响而返回。

例如，提供5+7将返回字符串'5+7'作为HostVar_n。

- Aggregate_n: 聚合函数，如AVG(Age)、COUNT(*)。

如果最外层的操作是聚合函数，那么列就被命名为Aggregate_n，即使这个聚合包含一个表达式。

例如，COUNT(Name)+COUNT(Spouse)是Expression_n，而MAX(COUNT(Name)+COUNT(Spouse))是Aggregate_n，-AVG(Age)是Expression_n，而AVG(-Age)是Aggregate_n。

- Expression_n: 在文本、字段或Aggregate_n、HostVar_n、Literal_n或Subquery_n选择项列表中的任何操作都会将其列名更改为Expression_n。

这包括对数字的一元操作(-Age)，算术操作(Age+5)，连接('USA:'||Home_State)，数据类型CAST操作，SQL排序函数(%SQLUPPER(Name)或%SQLUPPER Name)，

SQL标量函数(\$LENGTH(Name))，用户定义的方法，CASE表达式，和特殊变量(如CURRENT_DATE或\$ZPI)。

- Window_n: 窗口函数的结果。

在OVER关键字的右括号之后指定列别名。

- Subquery_n: 指定单个选择项的子查询的结果。

选择项可以是字段、聚合函数、表达式或文字。

在子查询之后而不是在子查询中指定列别名。

在面的例子中，AVG函数创建的聚合字段列的别名是“AvgAge”；

它的默认名称是“Aggregate_3”(一个在SELECT列表中位置3的聚合字段)。

```
SELECT Name, Age, AVG(Age) AS AvgAge FROM Sample.Person
```

下面的示例与上一个示例相同，只是此处省略了AS关键字。
建议使用该关键字，但不是必需

```
SELECT Name, Age, AVG(Age) AvgAge FROM Sample.Person
```

下面的示例演示如何为选择项子查询指定列别名：

```
SELECT Name AS PersonName,  
       (SELECT Name FROM Sample.Employee) AS EmpName,  
       Age AS YearsOld  
FROM Sample.Person
```

FROM子句

FROM table-ref子句指定一个或几个表、视图、表值函数或子查询。

可以将这些table-ref类型的任意组合指定为逗号分隔列表或使用JOIN语法。

如果指定单个table-ref，则从该表或视图检索指定的数据。

如果指定多个表引用，SQL将对这些表执行连接操作，将它们的数据合并到一个结果表中，从这个结果表中检索指定的数据。

如果指定了多个table-ref，可以用逗号或显式连接语法关键字分隔这些表名。

可以使用`$$SYSTEM.SQL.Schema.TableExists("schema.tname")`或`$$SYSTEM.SQL.Schema.ViewExists("schema.viewname")`方法来确定当前名称空间中是否存在表或视图。

可以使用`$$SYSTEM.SQL.Security.CheckPrivilege()`方法来确定是否对该表或视图具有SELECT权限。

表的别名

当指定table-ref时，可以使用AS关键字指定该表名或视图名的别名：

```
FROM Sample.Person AS P
```

AS关键字不是必需但使查询文本更容易阅读。

下面是有效的等价语法：

```
FROM Sample.Person P
```

t-alias名称必须是有效的标识符。

别名可以是分隔的标识符。

t-alias在查询中的表别名之间必须是唯一的。

与所有标识符一样，t-alias不区分大小写。

因此，不能指定两个只有字母大小写不同的t-alias名称。

这将导致SQLCODE -20“名称冲突”错误。

表别名用作字段名的前缀(带句点)，以指示字段所属的表。

例如：

```
SELECT P.Name, E.Name
FROM Sample.Person AS P, Sample.Employee AS E
```

当查询指定多个具有相同字段名的表时，必须使用表引用前缀。
表引用前缀可以是t-alias如上所示)，也可以是全限定表名，如面的等价示例所示：

```
SELECT Sample.Person.Name, Sample.Employee.Name
FROM Sample.Person, Sample.Employee
```

但是，如果已为该表名分配了t-alias，则不能将完整表名作为该选择项的一部分。
尝试这样做会导致SQLCODE -23错误。

当查询仅引用一个表(或视图)时，可选择指定表别名。
当查询引用多个表(和/或视图)且引用的字段名对每个表都是唯一的时，指定表别名是可选的(但推荐)。
当查询引用多个表(和/或视图)，并且在不同的表中引用的字段名相同时，需指定表别名。
没有指定t-alias(或完全限定的表名)前缀将导致SQLCODE -27“字段%1D在适用的表中不明确”错误。

当指定如子查询时，可以使用t-alias，但不是必需

```
SELECT Name, (SELECT Name FROM Sample.Vendor)
FROM Sample.Person
```

t-alias仅唯一标识查询执行的字段；
要唯一地标识用于显示查询结果的字段，还必须使用列别名(c-alias)。
下面的示例使用了表别名(Per和Emp)和列别名(PName和EName)：

```
SELECT Per.Name AS PName, Emp.Name AS EName
FROM Sample.Person AS Per, Sample.Employee AS Emp
WHERE Per.Name %STARTSWITH 'G'
```

可以为字段、列别名和/或表别名使用相同的名称，而不会产生命名冲突。

如果需区分引用的是哪个表，则使用t-alias前缀。
以下是一些例子：

```
SELECT P.%ID As PersonID,
       AVG(P.Age) AS AvgAge,
       Z.%TABLENAME || '=' AS Tablename,
       Z.*
FROM Sample.Person AS P, Sample.USZipCode AS Z
WHERE P.Home_City = Z.City
GROUP BY P.Home_City
ORDER BY Z.City
```

Sharding Transparent to SELECT Queries

分片对SQL查询是透明的；
不需特殊的查询语法。

查询不需要知道FROM子句中指定的表是分片的还是非分片的。
同一个查询可以访问分片表和非分片表。
查询可以包括分片表和非分片表之间的连接。

分片表使用CREATE table命令定义。
它必须在分片主数据服务器上的主命名空间中定义。
这个主命名空间还可以包括非分片表。

[#SQL](#) [#Caché](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%83%E5%8D%81%E4%B8%80%E7%AB%A0-sql%E5%91%BD%E4%BB%A4-select%EF%BC%88%E4%B8%89%EF%BC%89>