

文章

[姚鑫](#) · 十一月 8, 2021 阅读大约需 9 分钟

第七十章 SQL命令 SELECT (二)

第七十章 SQL命令 SELECT (二)

select-item

这是所有SELECT语句的必选元素。

通常，选择项指的是FROM子句中指定的表中的一个字段。

选择项由下列一个或多个项组成，多个项之间用逗号分隔：

- 列名(字段名)，带或不带表名别名：

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

字段名不区分大小写。

但是，结果集中与字段关联的标签使用表定义中指定的SqlFieldName的字母大小写，而不是选择项中指定的字母大小写。

包含一个或多个下划线的字段名引用嵌入的串行对象属性。

例如，对于字段名HomeCity，表包含一个引用字段Home，该字段引用定义属性City的嵌入式串行对象。

对于字段名HomePhoneAreaCode，该表包含一个引用字段Home，该字段引用嵌入式串行对象属性Phone，该属性引用定义AreaCode属性的嵌套嵌入式串行对象。

如果选择一个引用字段，如Home或HomePhone，则以%List数据类型格式接收串行对象中所有属性的值。

要显示RowID(记录ID)，可以使用%ID伪字段变量别名，该别名显示RowID，而不管它被分配的名称是什么。

默认情况下，RowID的名称是ID，但如果存在用户定义的名为ID的字段，IRIS可能会重命名它。

默认情况下，RowID是一个隐藏字段。

stream字段上的SELECT返回打开的stream对象的oref(对象引用)：

```
SELECT Name, Picture FROM Sample.Employee WHERE Picture IS NOT NULL
```

当FROM子句指定多个表或视图时，必须使用句点将表名(或表名别名)作为选择项的一部分，如下面的两个示例所示：

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

表名别名：

```
SELECT p.Name, e.Company  
FROM Sample.Person AS p, Sample.Employee AS e
```

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

可以使用排序函数指定选择项字段的排序和显示。

可以提供不带括号(SELECT %SQLUPPER Name)或带括号(SELECT %SQLUPPER(Name))的排序规则函数。如果排序规则函数指定了截断，则括号是必需的(SELECT %SQLUPPER(Name,10))。

当选择项引用嵌入的串行对象属性(嵌入的串行类数据)时，使用下划线语法。

下划线语法由对象属性的名称、下划线和嵌入对象中的属性组成:例如，HomeCity和HomeState。(在其他上下文中，例如索引表，它们使用点语法表示:Home.City。)

```
SELECT Home_City,Home_State FROM Sample.Person
```

可以使用SELECT直接查询引用字段(例如Home)，而不是使用下划线语法。

因为返回的数据是列表格式的，所以可能需要使用\$LISTTOSTRING或\$LISTGET函数来显示数据。例如:

```
SELECT $LISTTOSTRING(Home,'^') AS HomeAddress FROM Sample.Person
```

- 子查询。
子查询返回指定表中的单个列。
这个列可以是单个表字段(SELECT Name)的值，也可以是作为单个列返回的多个表字段的值，可以使用连接(SELECT HomeCity||HomeState)或指定容器字段(SELECT Home)。
子查询可以使用隐式连接(箭头语法)。
子查询不能使用星号语法，即使在子查询中引用的表只有一个数据字段。

子查询的一个常见用法是指定不受GROUP BY子句约束的聚合函数。

在下面的示例中，GROUP BY子句按几十年(例如，25到34)对年龄进行分组。

AVG(Age)选择项给出了由group by子句定义的每个组的平均年龄。

为了获得所有组中所有记录的平均年龄，它使用了一个子查询:

```
SELECT Age AS Decade,  
       COUNT(Age) AS PeopleInDecade,  
       AVG(Age) AS AvgAgeForDecade,  
       (SELECT AVG(Age) FROM Sample.Person) AS AvgAgeAllDecades  
FROM Sample.Person  
GROUP BY ROUND(Age,-1)  
ORDER BY Age
```

- 箭头语法，用于访问from子句表以外的表中的字段。
这被称为隐式连接。
在下面的示例中，示例。
Employee表包含Company字段，其中包含Sample中对应公司名称的RowID。
公司表。
箭头语法从表中检索公司名称:

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

在这种情况下，必须拥有被引用表的SELECT特权:对于被引用表的字段和RowID列，可以是表级的SELECT特权，也可以是列级的SELECT特权。

- 星号语法(*), 按列号顺序选择表中的所有列:

```
SELECT TOP 5 * FROM Sample.Person
```

星号语法选择嵌入的串行对象属性(字段), 包括嵌套在串行对象中的串行对象的属性。

没有选择引用串行对象的字段。

例如, 选择来自嵌入式串行对象的HomeCity属性, 但是使用Home引用字段访问Sample。

没有选择地址嵌入串行类(包含City属性)。

星号语法不选择隐藏字段。

默认情况下, RowID是隐藏的(不通过SELECT *显示)。

但是, 如果表定义为%PUBLICROWID, 则SELECT *返回RowID字段和所有非隐藏字段。

缺省情况下, 该字段的名称为ID, 但如果存在用户自定义的ID字段, IRIS可能会对其进行重命名。

在下面的示例中, select-item由一个非限定星号语法组成, 用于从表中选择所有列。

注意, 你也可以指定重复的列名(在本例中是Name)和非列的select-item元素(在本例中是{fn NOW}):

```
SELECT TOP 5 {fn NOW} AS QueryDate,  
           Name AS Client,  
           *  
FROM Sample.Person
```

在下面的示例中, select-item由限定星号语法组成, 用于从一个表中选择所有列, 以及从另一个表中选择列名列表。

```
SELECT TOP 5 E.Name AS EmpName,  
           C.*,  
           E.Home_State AS EmpState  
FROM Sample.Employee AS E, Sample.Company AS C
```

注意:SELECT *是 SQL完全支持的一部分, 在应用程序开发和调试期间, 它非常方便。

但是, 在生产应用程序中, 首选的编程实践是显式列出所选字段, 而不是使用星号语法形式。

显式列出字段可以使应用程序更清晰、更容易理解、更容易维护, 并更容易按名称搜索字段。

- 包含一个或多个SQL聚合函数的选择项。
聚合函数总是返回单个值。
聚合函数的参数可以是下列任何一种:

- 单个列名计算查询所选行的所有非空值的聚合:

```
SELECT AVG(Age) FROM Sample.Person
```

- 也允许使用标量表达式来计算聚合:

```
SELECT SUM(Age) / COUNT(*) FROM Sample.Person
```

- 星号语法(*)-与COUNT函数一起使用, 用于计算表中的行数:

```
SELECT COUNT(*) FROM Sample.Person
```

- 一个选择不同的函数-通过消除冗余值计算聚合:

```
SELECT COUNT(DISTINCT Home_State) FROM Sample.Person
```

- 虽然ANSI

SQL不允许在单个SELECT语句中组合列名和聚合函数，但SQL扩展了这一标准，允许这样做:

```
SELECT Name, COUNT(DISTINCT Home_State) FROM Sample.Person
```sql
- ??`%FOREACH`??????
?????????????????????:
```sql
SELECT DISTINCT Home_State, AVG(Age %FOREACH(Home_State))
FROM Sample.Person
```

- 使用%AFTERHAVING的聚合函数。

这导致在HAVING子句指定的子population上计算聚合:

```
SELECT Name,AVG(Age %AFTERHAVING)
FROM Sample.Person
HAVING (Age > AVG(Age))
```

将返回年龄大于平均年龄的那些记录，给出年龄高于数据库中所有人平均年龄的那些人的平均年龄。

•

Window function语法，支持基于特定于该行的“窗口框架”为每一行计算聚合、排名和其他函数。支持以下语法

```
window-function() OVER (PARTITION BY partfield ORDER BY orderfield)
```

- window-function:支持如下窗口函数:ROWNUMBER(), RANK(), PERCENTRANK(), FIRSTVALUE(field), SUM(field)。
- OVER:必须在OVER关键字后面加上括号。括号中的子句是可选的。
- PARTITION BY partfield:可选子句，根据指定的partfield对行进行分区。部分字段可以是单个字段，也可以是用逗号分隔的字段列表。如果指定了，PARTITION BY必须在ORDER BY之前指定。
- ORDER BY orderfield:可选子句，根据指定的orderfield对行进行排序。Orderfield可以是单个字段，也可以是逗号分隔的字段列表。

在Window function中指定的字段可以接受表别名前缀。

Window function可以指定列别名。
默认情况下，列被标记为Window_n。

- 作为过程存储的用户定义的方法。
可以是非限定方法名，也可以是限定方法名。
以下是所有有效的类方法名:
Sample.RandLetter()一个限定的类方法名;
和Sample.RandLetter()调用类方法“RandLetter”()。
在下面的例子中，RandCaseLetter()是一个返回随机字母的类方法，可以是大写字母(' U

'), 也可以是小写字母('l'):

```
SELECT RandCaseLetter('U')
```

该方法的返回值将自动从Logical格式转换为Display/ODBC格式。

默认情况下, 该方法的输入值不会从Display/ODBC格式转换为Logical格式。

但是, 可以使用`$SYSTEM.SQL.Util.SetOption("SQLFunctionArgConversion")`方法在系统范围内配置输入显示到逻辑的转换。

可以使用`$SYSTEM.SQL.Util.GetOption("SQLFunctionArgConversion")`来确定该选项的当前配置。

如果指定的方法在当前名称空间中不存在, 系统将生成SQLCODE -359错误。

如果指定的方法不明确(可能引用多个方法), 系统将生成SQLCODE -358错误。

- 对数据库列进行操作的用户提供的ObjectScript函数调用(外部函数):

```
SELECT $$REFORMAT(Name) FROM MyTable
```

如果在系统范围内配置了“允许SQL语句中的外部函数”选项, 则只能在SQL语句中调用用户提供的(外部)函数。

默认为“ No ”;

默认情况下, 尝试调用用户提供的函数会产生SQLCODE -372错误。

不能使用用户提供的函数来调用%routine(以%字符开头的例程)。

尝试这样做会产生SQLCODE -373错误。

- 对字段值应用额外处理的选择项:

算术运算:

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

如果选择项算术运算包括除法, 并且数据库中该字段的任何值都可能产生值为零或NULL的除数, 则不能依赖测试顺序来避免被零除法。

相反, 使用case语句来抑制风险。

SQL函数:

```
SELECT Name, $LENGTH(Name) FROM Sample.Person
```

SQL case转换函数:

SQL case?????:

一个XMLELEMENT、XMLFOREST或XMLCONCAT函数, 它在从指定列检索的数据值周围放置XML(或HTML)标记。

- 为所有记录返回相同值的选择项。

当所有select-items都不引用表数据时，FROM子句是可选的。
如果包含FROM子句，则指定的表必须存在。

- 算术运算:

```
SELECT 7 * 7, 7 * 8 FROM Sample.Person
```

```
SELECT Name, Age, 9 - 6 FROM Sample.Person
```

- 字符串字面值或操作字符串字面值的函数:

```
SELECT UCASE('fred') FROM Sample.Person
```

字符串字面量可以用来产生更可读的输出，如下面的示例所示:

```
SELECT TOP 10 Name, 'was born on', %EXTERNAL(DOB)  
FROM Sample.Person
```

数值字面值的指定方式决定了它的数据类型。

因此，字符串'123'将被报告为数据类型VARCHAR，而数字123将被报告为数据类型INTEGER或numeric。

- %TABLENAME或%CLASSNAME伪字段变量关键字。
%TABLENAME返回当前表名。
%CLASSNAME返回当前表对应的类名。
如果查询引用多个表，可以在关键字前加上表别名。
例如,t1.%TABLENAME。
- 以下ObjectScript特殊变量之一(或其缩写):\$HOROLOG, \$JOB, \$NAMESPACE, \$TLEVEL, \$USERNAME, \$ZHOROLOG, \$ZJOB, \$ZNSPACE, \$ZPI, \$ZTIMESTAMP, \$ZTIMEZONE, \$ZVERSION。

[#SQL #Caché](#)

源

URL:

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