文章
姚 鑫 · 十二月 22, 2021 阅读大约需 分钟

第二章 SQL聚合函数 AVG

第二章 SQL聚合函数 AVG

返回指定列值的平均值的聚合函数。

大纲

```
AVG([ALL | DISTINCT [BY(col-list)]] expression [%FOREACH(col-list)] [%AFTERHAVING])
```

参数

- ALL - 可选-指定AVG返回表达式所有值的平均值。
  如果没有指定关键字，则为默认值。
- DISTINCT - 可选 - DISTINCT子句，指定AVG只计算一个值的唯一实例的平均值。
  DISTINCT可以指定BY(col-list)子句，其中col-list可以是单个字段，也可以是逗号分隔的字段列表。
- expression - 任何有效的表达式。
  通常是包含要取平均值的数据值的列的名称。
- %FOREACH(col-list) - 可选—列名或以逗号分隔的列名列表。
- %AFTERHAVING - 可选 - 应用在HAVING子句中找到的条件。

AVG返回NUMERIC或DOUBLE数据类型。
如果expression是DOUBLE类型，AVG返回DOUBLE;
否则，它返回NUMERIC。

描述

AVG聚合函数返回表达式值的平均值。
通常，表达式是查询返回的行中字段的名称(或包含一个或多个字段名称的表达式)。

AVG可以用于引用表或视图的SELECT查询或子查询。
AVG可以出现在SELECT列表或HAVING子句中，与普通字段值一起出现。

AVG不能在WHERE子句中使用。
AVG不能在JOIN的ON子句中使用，除非SELECT是子查询。

像所有聚合函数一样，AVG可以带有一个可选的DISTINCT子句。
AVG(DISTINCT col1)仅对不同(唯一)的col1字段值进行平均。
AVG(DISTINCT BY(col2) col1)仅对col2值不同(唯一)的记录中的col1字段值进行平均值。
但是请 注 意，不同的col2值可能包含一个单独的NULL值。

数据值

对于非double表达式值，AVG返回双精度浮点数。
AVG返回的值的精度是18。
返回值的比例取决于表达式的精度和比例:AVG返回值的比例等于18减去表达式的精度，加上表达式的比例(as=ap-ep+es)。

对于DOUBLE表达式值，精度为0。

AVG通常应用于具有数值值的字段或表达式，例如数字字段或日期字段。
默认情况下，聚合函数使用逻辑(内部)数据值，而不是Display值。
因为没有执行类型检查，所以可以(尽管很少有意义)对非数字字段调用类型检查;
AVG计算非数值，包括空字符串('')为零(0)。如果expression是数据类型VARCHAR，则返回值为数据类型DOUBLE。

在导出AVG聚合函数值时，数据字段中的NULL值将被忽略。
如果查询没有返回行，或者返回的所有行的数据字段值为NULL，AVG返回NULL。

对单个值求平均值

如果提供给AVG的所有表达式值都是相同的，那么结果的平均值取决于访问表中的行数(除数)。
例如，如果表中的所有行对某个特定列具有相同的值，那么该列的平均值就是一个计算值，它可能与个别列中的值略有不同。
为了避免这种差异，可以使用DISTINCT关键字。

下面的例子展示了计算平均值如何产生轻微的不平等。
第一个查询不引用表行，所以AVG通过除以1进行计算。
第二个查询引用表的行，因此AVG通过除以表中的行数进行计算。
第三个查询引用了表行，但是平均了单个值的DISTINCT值;
在这种情况下，AVG计算除以1。

```
ClassMethod Avg()
{
    s pi = $ZPI
    &sql(SELECT :pi,AVG(:pi) INTO :p,:av FROM Sample.Person)
    w p," pi??",!
    w av," avg of pi/1",!
    &sql(SELECT Name,:pi,AVG(:pi) INTO :n,:p,:av FROM Sample.Person)
    w av," avg calculated using numrows",!
    &sql(SELECT Name,:pi,AVG(DISTINCT :pi) INTO :n,:p,:av FROM Sample.Person)
    w av," avg of pi/1"
}
```

```
3.141592653589793238 pi??
3.141592653589793238 avg of pi/1
3.141592653589793206 avg calculated using numrows
3.141592653589793206 avg of pi/1
```

优化

SQL优化AVG计算可以使用位片索引，如果这个索引是为字段定义的。

当前事务期间所做的更改

与所有聚合函数一样，无论当前事务的隔离级别如何，AVG总是返回数据的当前状态，包括未提交的更改。

示例

下面的查询列出了Sample中所有员工的平均工资。
员工的数据库。
因为查询返回的所有行对于这个平均值具有相同的值，所以该查询只返回一行，其中包含平均工资。
为了显示目的，该查询将一个美元符号连接到值(使用||操作符)，并使用AS子句标记列：

```
SELECT '$' || AVG(Salary) AS AverageSalary
     FROM Sample.Employee
```

下面的查询列出了每个州的员工的平均工资：

```
SELECT Home_State,'$' || AVG(Salary) AS AverageSalary
     FROM Sample.Employee
GROUP BY Home_State
```

下面的查询列出了那些工资大于平均工资的员工的姓名和工资。
它还列出了所有员工的平均工资；
这个值对于查询返回的所有行都是相同的：

```
SELECT Name,Salary,
       '$' || AVG(Salary) AS AverageAllSalary
FROM Sample.Employee
HAVING Salary>AVG(Salary)
ORDER BY Salary
```

下面的查询列出了那些工资大于平均工资的员工的姓名和工资。
它还列出了高于平均水平的员工的平均工资；
这个值对于查询返回的所有行都是相同的：

```
SELECT Name,Salary,
       '$' || AVG(Salary %AFTERHAVING) AS AverageHighSalary
FROM Sample.Employee
HAVING Salary>AVG(Salary)
ORDER BY Salary
```

以下查询列出了那些包含三名以上员工的州，这些员工的平均工资为该州员工的平均工资，以及该州收入超过20,000美元的员工的平均工资：

```
SELECT Home_State,
       '$' || AVG(Salary) AS AvgStateSalary,
       '$' || AVG(Salary %AFTERHAVING) AS AvgLargerSalaries
FROM Sample.Employee
GROUP BY Home_State
HAVING COUNT(*) > 3 AND Salary > 20000
ORDER BY Home_State
```

以下查询使用DISTINCT子句的几种形式。AVG(DISTINCT BY COLLIST)示例可以在平均值中包括附加的年龄值，因为如果Home_City包含一个或多个Null，则BY子句可以包括单个NULL作为DISTINCT值：

```
SELECT AVG(Age) AS AveAge,AVG(ALL Age) AS Synonym,
       AVG(DISTINCT Age) AS AveDistAge,
       AVG(DISTINCT BY(Home_City) Age) AS AvgAgeDistCity,
       AVG(DISTINCT BY(Home_City,Home_State) Age) AS AvgAgeDistCityState
     FROM Sample.Person
```

以下查询同时使用%FOREACH和%AFTERHAVING关键字。它为那些包含姓名以A、M或W(HAVING子句和GROUP BY子句)开头的人的州返回一行。每个状态行包含列值：

- LIST(Age %FOREACH(Home_State)):该州所有人的年龄列表。
- AVG(Age %FOREACH(Home_State)):全州所有人的平均年龄。
- AVG(Age %AFTERHAVING):数据库中符合HAVING子句条件的所有人员的平均年龄。(此数字对于所有行都是相同的。)
- LIST(Age %FOREACH(Home_State) %AFTERHAVING):
  该州符合HAVING子句标准的所有人员的年龄列表。
- AVG(Age %FOREACH(Home_State) %AFTERHAVING):该州所有符合HAVING子句标准的人的平均年龄。

```
SELECT Home_State,
       LIST(Age %FOREACH(Home_State)) AS StateAgeList,
       AVG(Age %FOREACH(Home_State)) AS StateAgeAvg,
       AVG(Age %AFTERHAVING ) AS AgeAvgHaving,
       LIST(Age %FOREACH(Home_State)%AFTERHAVING ) AS StateAgeListHaving,
       AVG(Age %FOREACH(Home_State)%AFTERHAVING ) AS StateAgeAvgHaving
FROM Sample.Person
GROUP BY Home_State
HAVING Name LIKE 'A%' OR Name LIKE 'M%' OR Name LIKE 'W%'
ORDER BY Home_State
```

#SQL #Caché

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E7%AB%A0-sql%E8%81%9A%E5%90%88%E5%87%BD%E6%95%B0-avg