文章
姚 鑫 · 一月 4, 2022 阅读大约需 6 分钟

第十四章 SQL窗口函数概述（一）

第十四章 SQL窗口函数概述（一）

指定用于计算聚合和排名的每行 “ 窗口框架 ” 的函数。

窗口函数和聚合函数

在应用WHERE、GROUP by和HAVING子句之后，窗口函数对SELECT查询选择的行进行操作。

窗口函数将一组行中的一个(或多个)字段的值组合在一起，并在结果集中为生成的列中的每一行返回一个值。

虽然窗口函数与聚合函数类似，因为它们将多行结果组合在一起，但它们与聚合函数的不同之处在于，它们本身并不组合行。

窗函数的语法

窗口函数被指定为SELECT查询中的选择项。
窗口函数也可以在SELECT查询的ORDER BY子句中指定。

窗口函数执行与由PARTITION by子句、ORDER by子句和ROWS子句指定的逐行窗口相关的任务，并为每一行返回一个值。
这三个子句都是可选的，但是如果指定了，必须按照以下语法中的顺序指定:

```
window-function() OVER (
                       [ PARTITION BY partfield ]
                       [ ORDER BY orderfield ]
                       [ ROWS framestart ] | [ ROWS BETWEEN framestart AND frameend ]
                       )
```

其中framestart和frameend可以是:

```
UNBOUNDED PRECEDING |
offset PRECEDING |
CURRENT ROW |
UNBOUNDED FOLLOWING |
offset FOLLOWING
```

- window-function:支持如下窗口函数:ROWNUMBER()、RANK()、PERCENTRANK()、FIRSTVALUE(字段)和SUM(字段)。
  该字段在显示的地方是必需的，在没有显示的地方是不允许的。
  括号对所有窗口函数都是必需的。
- OVER: OVER关键字后面必须加上括号。

括号中的子句是可选的。

- PARTITION BY partfield:一个可选子句，根据指定的partfield分区行。
  Partfield可以是单个字段，也可以是用逗号分隔的字段列表。
  partfield可以是聚合函数、标量函数(如LENGTH(Name)或ROUND(Salary，-2))，或者表达式(如Salary+Bonus)。
  部分字段不能是流字段;
  尝试这样做会产生一个SQLCODE -37错误。
  如果指定了PARTITION BY，必须在ORDER BY之前指定PARTITION BY。

如果指定了一个PARTITION BY子句，行被分组在指定的窗口中，窗口函数创建一个新的结果集字段并为每一行分配一个值。
例如，PARTITION BY City将共享相同City字段值的所有行分组到同一个窗口中;
窗口函数根据这个分组分配行值。

- ORDER BY orderfield:一个可选子句，根据指定的orderfield对行排序。
  Orderfield可以是单个字段，也可以是用逗号分隔的字段列表。
  订单字段可以是一个聚合函数，一个标量函数(例如LENGTH(Name)或ROUND(Salary，-2))，或者一个表达式(例如Salary+Bonus)。
  订单字段不能是流字段;
  尝试这样做会产生一个SQLCODE -37错误。

ORDER BY按排序规则升序对窗口函数值进行排序。如果指定PARTITION BY和ORDER BY，则行将被分区为组，每个组的orderfield值将被排序，窗口函数将创建一个新的结果集字段并为每行赋值。如果在没有PARTITION BY子句的情况下指定ORDER BY子句，则所有选定的行将在单个窗口中分组、排序，然后赋值。例如，ORDER BY City根据City字段的值对所有行进行排序，然后Window函数按该顺序为每行赋值。

- ROWS：具有两种支持的语法形式的可选子句：ROWS、FRAME START或ROWS介于Frame Start和Frame End之间。ROWS通过指定分区内的起始点和结束点(包括范围点)，对分区内的连续行执行滚动操作。它需要一个ORDER BY子句来建立行序列。它可以选择性地指定PARTITION BY子句。如果未指定ROWS子句，则缺省值为从分区开始处(前面未绑定)到当前行。ROWS子句可以与firstvalue(Field)和sum(Field)窗口函数一起使用。

简单的例子

CityTable包含具有以下值的行:

Name	City
Able	New York
Betty	Boston
Charlie	Paris
Davis	Boston
Eve	Paris
Francis	Paris
George	London
Beatrix	Paris

ROWNUMBER()窗口函数根据指定的窗口为每一行分配一个唯一的连续整数。

```
SELECT Name,City,ROW_NUMBER() OVER (PARTITION BY City) FROM CityTable
```

Name	City	Window3
Able	New York	1
Betty	Boston	1
Charlie	Paris	1
Davis	Boston	2
Eve	Paris	2

Name	City	Window3
Francis	Paris	3
George	London	1
Beatrix	Paris	4

```
SELECT Name,City,ROW_NUMBER() OVER (ORDER BY City) FROM CityTable
```

本例将所有行视为单个分区。
它根据City值对行排序，并返回以下结果:

Name	City	Window3
Able	New York	4
Betty	Boston	1
Charlie	Paris	5
Davis	Boston	2
Eve	Paris	6
Francis	Paris	7
George	London	3
Beatrix	Paris	8

```
SELECT Name,City,ROW_NUMBER() OVER (Partition BY City ORDER BY Name) FROM CityTable
```

这个例子根据City值对行进行分区，根据Name值对每个City分区排序，并返回以下结果:

Name	City	Window3
Able	New York	1
Betty	Boston	1
Charlie	Paris	2
Davis	Boston	2
Eve	Paris	3
Francis	Paris	4
George	London	1
Beatrix	Paris	1

NULL

PARTITION BY子句将字段为NULL(没有分配值)的行作为分区组处理。
例如，ROWNUMBER() OVER (Partition BY City)会将没有City值的行分配为顺序整数，就像它将顺序整数分配给City值为'Paris'的行一样。

ORDER BY子句将字段为NULL(没有分配值)的行按照在任何分配值(具有最低的排序值)之前的顺序处理。
例如，ROWNUMBER() OVER (ORDER BY City)首先将顺序整数分配给没有City值的行，然后将顺序整数分配给排序顺序中具有City值的行。

ROWS子句将NULL(没有赋值)的字段视为值为零。
例如，SUM(Scores) OVER (ORDER BY Scores ROWS 1 above)/2将分配0.00给所有没有分数值的行((0 + 0)/2)，并通过将0加到它然后除以2来处理第一个分数值。

支持的窗口函数

支持以下窗口函数:

- FIRSTVALUE(field)——将指定窗口中第一行(ROWNUMBER()=1)的字段列的值赋给该窗口中的所有行。
  例如:FIRSTVALUE(Country) OVER (PARTITION BY City)。
  FIRSTVALUE()支持ROWS子句。
  注意，NULL排序在所有值之前，所以如果第一行中的字段值是NULL，那么窗口中的所有行都将是NULL。

- PERCENTRANK()——将排名百分比作为0到1(包括1)之间的小数分配给同一窗口中的每一行。
  如果窗口函数字段的多个行包含相同的值，那么排名百分比可能包含重复的值。
- RANK()——给同一窗口中的每一行分配一个排序整数，从1开始。
  如果窗口函数字段的多个行包含相同的值，那么对整数的排序可以包含重复的值。
- ROWNUMBER()——为同一窗口中的每一行分配一个唯一的连续整数，从1开始。
  如果多行窗口函数字段包含相同的值，则为每一行分配一个唯一的连续整数。
- SUM(field)——将指定窗口中字段列值的和赋给该窗口中的所有行。

SUM既可以用作聚合函数，也可以用作窗口函数。
SUM()支持ROWS子句。

下面的例子比较了这些窗口函数中ORDER by子句返回的值:

```
SELECT Name,City,ROW_NUMBER() OVER (ORDER BY City) AS RowNum,
  RANK() OVER (ORDER BY City) AS RankNum,
  PERCENT_RANK() OVER (ORDER BY City) AS RankPct
  FROM CityTable ORDER BY City
```

本例将所有行视为单个分区。
它根据City值对行排序，并返回以下结果:

Name	City	RowNum	RankNum	RankPct
Harriet		1	1	0
Betty	Boston	2	2	.111111111111111111
Davis	Boston	3	2	.111111111111111111
George	London	4	4	.333333333333333333
Able	New York	5	5	.444444444444444444
Charlie	Paris	6	6	.555555555555555555
Eve	Paris	7	6	.555555555555555555
Francis	Paris	8	6	.555555555555555555
Beatrix	Paris	9	6	.555555555555555555
Jackson	Rome	10	10	1

#SQL #Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E5%9B%9B%E7%AB%A0-sql%E7%AA%97%E5%8F%A3%E5%87%BD%E6%95%B0%E6%A6%82%E8%BF%B0%EF%BC%88%E4%B8%80%EF%BC%89