

你真的会玩SQL吗？实用函数方法汇总

2015-12-04 来源互联网 DatabaseG

我们往往需要临时存储某些结果集。除了用临时表和表变量，还可以使用公用表表达式的方法。

表表达式

- 期待单个值的地方可以使用**标量子查询**
- 期待多个值的地方可以使用**多值子查询**
- 在期待出现表的地方可用**表值子查询**或**表表达式**

1.派生表

是从查询表达式派生出虚拟结果表的表表达式，派生表的存在范围只是外部查询。

使用形式：from 派生表 as 派生表列名

规则：

- 所有列必须有名称
- 列名必须唯一
- 不允许使用order by（除非指定了top）

不同于标量和多值子查询，派生表不能是相关的，它必须是独立的。

2.公用表表达式（CTE）

非递归公用表表达式（CTE）是查询结果仅仅一次性返回一个结果集用于外部查询调用。

```

1  WITH CTE_Test
2      AS
3      (
4          SELECT * FROM Person_1
5      )
6  SELECT * FROM CTE_Test AS a    --第一次引用
7  INNER JOIN CTE_Test AS b      --第二次引用
8  ON a.Id = b.Id
9  ORDER BY a.Id DESC--SELECT * FROM CTE_Test 再查询一次会报错

```

递归公用表达式

来引用他人的一个示例：

先建一张表栏目表如下，栏目Id，栏目名称，栏目的父栏目

| KISSDODOG-PC.Nx - dbo.Col | | SQLQuery6.sql - KI...G-Pi | |
|---------------------------|------|---------------------------|----------|
| | Id | Name | ParentId |
| ▶ | 1 | 国内新闻 | 0 |
| | 2 | 广东新闻 | 1 |
| | 3 | 广州新闻 | 2 |
| | 4 | 天河新闻 | 3 |
| | 5 | 山东新闻 | 1 |
| | 6 | 青岛新闻 | 5 |
| * | NULL | NULL | NULL |

现在使用CTE查询其每个栏目是第几层栏目的代码如下：

```

1  WITH COL_CTE(Id, Name, ParentId, tLevel )
2  AS

```

```

3  (
4      --基本语句
5      SELECT Id, Name, ParentId, 0 AS tLevel FROM Col
6      WHERE ParentId = 0
7      UNION ALL
8      --递归语句
9      SELECT c. Id, c. Name, c. ParentId, ce. tLevel+1 AS tLevel FROM COL as c
10     INNER JOIN COL_CTE AS ce      --递归调用
11     ON c. ParentId = ce. Id
12 )
13
14 SELECT * FROM COL_CTE

```

结果：



| | Id | Name | ParentId | tLevel |
|---|----|------|----------|--------|
| 1 | 1 | 国内新闻 | 0 | 0 |
| 2 | 2 | 广东新闻 | 1 | 1 |
| 3 | 5 | 山东新闻 | 1 | 1 |
| 4 | 6 | 青岛新闻 | 5 | 2 |
| 5 | 3 | 广州新闻 | 2 | 2 |
| 6 | 4 | 天河新闻 | 3 | 3 |

0表示顶级栏目。1就是1级栏目

排名函数

四个排名函数：

1.*row_number*

2.*rank*

3.*dense_rank*

4.*ntile*

排名函数order by子句是必需的。我们这里不讲定义，直接讲实例用法。

利用row_number生成连续行号

```

1  SELECT empid ,
2         qty ,
3         ROW_NUMBER() OVER ( ORDER BY qty ) AS rownum
4  FROM    sales
5  ORDER BY qty

```

小的分组范围内排序，通过PARTITION BY选项来重新排序，给数据分区或者数据区域唯一的递增序号

如：LastName以‘A’开头的作为第一组，在这个组内进行排序。以‘B’开头的作为第二组，在这个组内排序。以‘C’开头的作为第三组，在这个组内进行排序，如此等等

```

1  select
2  ROW_NUMBER() over(PARTITION by substring(LastName,1,1) order by LastName) as RowN
3  FirstName+' ' + LastName as FullName
4  from HumanResources.vEmployee

```

结果

| Results | | Messages |
|---------|--------------------|----------|
| RowNum | FullName | |
| 1 | Kim Abercrombie | |
| 2 | Jay Adams | |
| 3 | Nancy Anderson | |
| 4 | Bryan Baker | |
| 5 | Ed Dudenhofer | |
| 6 | Maciej Dusza | |
| 7 | Charles Fitzgerald | |
| 8 | Guy Gilbert | |
| 9 | Brandon Heidepriem | |
| 10 | Karan Khanna | |
| 11 | Eugene Kogan | |
| 12 | James Warner | |

假设LastName以 'A' 开头的是男子组，这个组共有三个人，Kim Abercrombie是冠军，Jay Adams是亚军，Nancy Anderson是季军。假设LastName以 'B' 开头的是女子组，这个组只有一个人Bryan Baker，无论如何她都是冠军。等等如此类推。这样一眼就能看出他们的小组名次了。

RANK

果有同时撞线的情况发生应该怎么计名次呢？例如A第一个撞线，B和C同时第二个撞线，D第三个撞线，如果我们想把D的名次计为第4名应该怎么处理呢？就是说不计顺序名次，只计人数。这时就可以使用RANK函数了。

在order by子句中定义的列上，如果返回一行数据与另一行具有相同的值，rank函数将给这些行赋予相同的排名数值。在排名的过程中，保持一个内部计数值，当值有所改变时，排名序号将有一个跳跃。

```

1 SELECT ROW_NUMBER() OVER ( ORDER BY Department ) AS RowNum ,
2       RANK() OVER ( ORDER BY Department ) AS Ranking ,
3       FirstName + ' ' + LastName AS FullName ,
4       Department
5 FROM   HumanResources.vEmployeeDepartment
6 ORDER BY RowNum

```

rank()函数右面也要跟上一个over子句。为了看到效果我们以Department作为排序字段，可以看到RowNum作为升序连续排名，Ranking作为计同排名，当Department的值相同时，Ranking中的值保持不变，当Ranking中的值发生变化时，Ranking列中的值将跳跃到正确的排名数值。来看结果：

| | RowNum | Ranking | FullName | Department |
|----|--------|---------|----------------------|------------------|
| 1 | 1 | 1 | Tengiz Kharatishvili | Document Control |
| 2 | 2 | 1 | Zainal Arifin | Document Control |
| 3 | 3 | 1 | Sean Chai | Document Control |
| 4 | 4 | 1 | Karen Berge | Document Control |
| 5 | 5 | 1 | Chris Norred | Document Control |
| 6 | 6 | 6 | Michael Sullivan | Engineering |
| 7 | 7 | 6 | Sharon Salavaria | Engineering |
| 8 | 8 | 6 | Roberto Tamburello | Engineering |
| 9 | 9 | 6 | Gail Erickson | Engineering |
| 10 | 10 | 6 | Jossef Goldberg | Engineering |
| 11 | 11 | 6 | Terri Duffy | Engineering |
| 12 | 12 | 12 | Laura Norman | Executive |
| 13 | 13 | 12 | Ken Sánchez | Executive |

从这个结果中我们可以说这次马拉松赛跑的排名是：Tengiz Kharatishvili，Zainal Arifin，Sean Chai，Karen Berge，Chris Norred并列第1，Michael Sullivan，Sharon Salavaria，Roberto Tamburello，Gail Erickson，Jossef Goldberg并列第6，如此等等。

DENSE_RANK

A第一个撞线，B和C同时第二个撞线，D第三个撞线，如果我们想把B和C的名次计为第2名，D的名次计为第3名应该怎么办呢？就是说考虑并列名次。这里使用DENSE_RANK函数

```

1 SELECT ROW_NUMBER() OVER ( ORDER BY Department ) AS RowNum ,
2       DENSE_RANK() OVER ( ORDER BY Department ) AS Ranking ,
3       FirstName + ' ' + LastName AS FullName ,
4       Department
5 FROM   HumanResources.vEmployeeDepartment
6 ORDER BY RowNum

```

结果

| | RowNum | Ranking | FullName | Department |
|----|--------|---------|----------------------|------------------|
| 1 | 1 | 1 | Tengiz Kharatishvili | Document Control |
| 2 | 2 | 1 | Zainal Arifin | Document Control |
| 3 | 3 | 1 | Sean Chai | Document Control |
| 4 | 4 | 1 | Karen Berge | Document Control |
| 5 | 5 | 1 | Chris Norred | Document Control |
| 6 | 6 | 2 | Michael Sullivan | Engineering |
| 7 | 7 | 2 | Sharon Salavaria | Engineering |
| 8 | 8 | 2 | Roberto Tamburello | Engineering |
| 9 | 9 | 2 | Gail Erickson | Engineering |
| 10 | 10 | 2 | Jossef Goldberg | Engineering |
| 11 | 11 | 2 | Terri Duffy | Engineering |
| 12 | 12 | 3 | Laura Norman | Executive |
| 13 | 13 | 3 | Ken Sánchez | Executive |

按照这个结果，我们可以说这次马拉松赛跑的排名是：Tengiz Kharatishvili，Zainal Arifin，Sean Chai，Karen Berge，Chris Norred并列第1，Michael Sullivan，Sharon Salavaria，Roberto Tamburello，Gail Erickson，Jossef Goldberg，Terri Duffy并列第2，等等。

NTILE

梭罗是铅笔的发明者，不过他没有申请专利。据说他天赋异禀，在父亲的铅笔厂里面打包铅笔的时候，从一堆铅笔里面抓取一把，每次都能精确地抓到一打12支。他在森林中目测两颗树之间的

距离，和护林员用卷尺测量的结果相差无几。现在如果我们想从一张表中抓取多比数据，每一笔都是相同的数目，并且标明第几组该怎么办呢？NTILE函数提供了这个功能。

```

1 SELECT NTILE(30) OVER ( ORDER BY Department ) AS NTiles ,
2     FirstName + ' ' + LastName AS FullName ,
3     Department
4 FROM   HumanResources.vEmployeeDepartment

```

| | NTiles | FullName | Department |
|----|--------|----------------------|----------------------------|
| 1 | 1 | Tengiz Kharatishvili | Document Control |
| 2 | 1 | Zainal Arfin | Document Control |
| 3 | 1 | Sean Chai | Document Control |
| 4 | 1 | Karen Berge | Document Control |
| 5 | 1 | Chris Norred | Document Control |
| 6 | 1 | Michael Sullivan | Engineering |
| 7 | 1 | Sharon Salavaria | Engineering |
| 8 | 1 | Roberto Tamburello | Engineering |
| 9 | 1 | Gail Erickson | Engineering |
| 10 | 1 | Jossef Goldberg | Engineering |
| 11 | 2 | Teri Duffy | Engineering |
| 12 | 2 | Laura Norman | Executive |
| 13 | 2 | Ken Sánchez | Executive |
| 14 | 2 | Christian Kleineman | Facilities and Maintenance |
| 15 | 2 | Lori Penor | Facilities and Maintenance |
| 16 | 2 | Stuart Macrae | Facilities and Maintenance |
| 17 | 2 | Jo Berry | Facilities and Maintenance |
| 18 | 2 | Gary Altman | Facilities and Maintenance |
| 19 | 2 | Magnus Hedlund | Facilities and Maintenance |
| 20 | 2 | Pat Coleman | Facilities and Maintenance |
| 21 | 3 | Mike Seamans | Finance |

现在我们要抓取30个组的数据，并保证尽可能的保证每组数目相同

这个视图中共290条数据， $290/30=9.7$ 约等于10，所以每组10条数据，如图每一条数据都有一个组号。这个结果要比索罗精确。

练习

用CTE删除重复数据

创建一个用于测试的表，并在该表里插入几条数据(包括重复的数据)

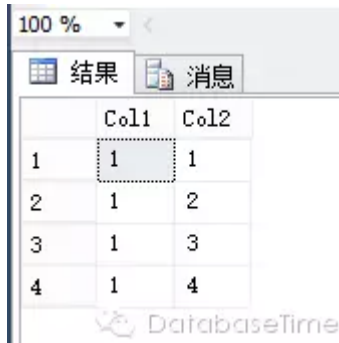
```

1  /* Create Table with 7 records- 3 are duplicate records*/
2  CREATE TABLE DeleteDuplicateTest ( Col1 INT, Col2 INT )
3  INSERT INTO DeleteDuplicateTest
4      SELECT 1 ,
5             1
6      UNION ALL
7      SELECT 1 ,
8             1 --duplicate
9      UNION ALL
10     SELECT 1 ,
11            1 --duplicate
12     UNION ALL
13     SELECT 1 ,
14            2
15     UNION ALL
16     SELECT 1 ,
17            2 --duplicate
18

```

```
19 UNION ALL
20 SELECT 1 ,
21         3
22 UNION ALL
23 SELECT 1 ,
24         4
GO
```

用CTE删除重复数据4条唯一的记录



| | Col1 | Col2 |
|---|------|------|
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |

参考SQL

```
1 WITH CTE ( Col1, Col2, DuplicateCount )
2     AS ( SELECT Col1 ,
3           Col2 ,
4           ROW_NUMBER() OVER ( PARTITION BY Col1, Col2 ORDER BY Col1
5           FROM DELETEDUPLICATETEST
6           )
7     )
8     DELETE FROM CTE
9     WHERE DuplicateCount > 1
10 GO
11 /*用SQL SERVER 的CTE，它将重新生成一个相同的但附加了一行编号的表。在此方案中，
```

以上部分示例总结于：T-SQL中的排名函数

DatabaseTime : 长按识别二维码关注



DatabaseTime

[阅读原文](#)