```
عنوان: آشنایی با Row_Number،Rank،Dense_Rank،NTILE
```

نویسنده: فرهاد فرهمندخواه تاریخ: ۲۲:۱۴ ۱۳۹۱/۰۵/۰۸ آدرس: www.dotnettips.info

برچسبها: SQL Server

توابعی که در این بررسی عنوان میشود در زمان انتشار نسخه SQL Server2005 ارائه شده است.

قبل از بررسی توابع، Script زیر را اجرا مینماییم، که شامل جدولی به نام Testو درج چند رکورد درون آن میباشد:

```
CREATE TABLE Test (ID INT, Product VARCHAR(100), Price INT, Color VARCHAR(100))
INSERT INTO Test
          'Toy', 100, 'Black'
SELECT 1,
UNION ALL
          'Pen', 100, 'Black'
SELECT 2,
UNION ALL
SELECT 3, 'Pencil', 100, 'Blue'
UNION ALL
          'Pencil', 100, 'Red'
SELECT 4,
UNION ALL
          'Pencil', 200, 'Yellow'
SELECT 5,
UNION ALL
SELECT 6,
          'Cup', 300, 'Orange'
UNION ALL
SELECT 7, 'Cup', 400, 'Brown'
```

اولین تابعی را که بررسی مینماییم، Row Number میباشد و Syntax آن بصورت زیر است:

```
ROW_NUMBER () OVER ([<partition_by_clause>] <order_by_clause>)
```

بوسیله تابع Row_Number میتوان اعداد توالی (ترتیبی) را به رکوردهای یک جدول نسبت داد. برای روشنتر شدن مطلب فوق مثالی را بررسی مینماییم.

در ابتدا Query زیر را اجرا نمایید:

Select *, ROW_NUMBER() OVER (ORDER BY Price DESC) AS RN from Test

خروجی آن بصورت زیر میباشد:

| Results Messages | | | | | |
|------------------|----|---------|-------|--------|----|
| | ID | Product | Price | Color | RN |
| 1 | 7 | Cup | 400 | Brown | 1 |
| 2 | 6 | Cup | 300 | Orange | 2 |
| 3 | 5 | Pencil | 200 | Yellow | 3 |
| 4 | 1 | Toy | 100 | Black | 4 |
| 5 | 2 | Pen | 100 | Black | 5 |
| 6 | 3 | Pencil | 100 | Blue | 6 |
| 7 | 4 | Pencil | 100 | Red | 7 |

همانطور که در شکل مشاهده مینمایید، یک عدد ترتیبی تولید شده و به هریک از رکوردهای جدول نسبت داده شده است، در مطلب بعدی یک مثال کاربردی از Row_Number خواهم زد.

لازم به یادآوری است که استفاده از Order by در Syntax تابع Row_Number الزامی میباشد.

اگر به Syntax تابع Row_Number توجه نماییم، با کلمه Partition مواجه میشویم،که جهت گروه بندی استفاده میشود،به عبارت دیگر ممکن است شما بخواهید، ابتدا جدول خود را براساس فیلد یا فیلدهایی Group by نمایید و سپس روی آنها Row_Number را اعمال کنید، که در این حالت از Partition استفاده میشود.

برای درک بیشتر Query زیر را اجرا نمایید:

Select *,ROW_NUMBER() OVER (PARTITION BY Product ORDER BY Price DESC) AS RN from Test

خروجی بصورت زیر خواهد بود:

| Results Messages | | | | | | |
|------------------|----|---------|-------|--------|----|--|
| | ID | Product | Price | Color | RN | |
| 1 | 7 | Cup | 400 | Brown | 1 | |
| 2 | 6 | Cup | 300 | Orange | 2 | |
| 3 | 2 | Pen | 100 | Black | 1 | |
| 4 | 5 | Pencil | 200 | Yellow | 1 | |
| 5 | 3 | Pencil | 100 | Blue | 2 | |
| 6 | 4 | Pencil | 100 | Red | 3 | |
| 7 | 1 | Toy | 100 | Black | 1 | |

همانطور که در شکل مشاهده مینمایید، در ابتدا، جدول براساس فیلد Product، دسته بندی (Group by) شده است و سپس اعداد ترتیبی روی هر Group by بصورت جداگانه اعمال شده است.

تابع ()RANK

از تابع فوق در جهت رتبه بندی نمودن فیلدهای یک جدول استفاده میشود و Syntax آن بصورت زیر میباشد:

RANK () OVER ([<partition_by_clause>] <order_by_clause>)

برای درک مطلب فوق نیز مثالی میزنیم: ابتدا Query زیر را اجرا مینماییم:

Select *,RANK() over (ORDER BY Price) AS RANK from Test

خروجی بصورت زیر خواهد بود:

| Results Messages | | | | | |
|------------------|----|---------------|-----|--------|------|
| | ID | Product Price | | Color | RANK |
| 1 | 1 | Toy | 100 | Black | 1 |
| 2 | 2 | Pen | 100 | Black | 1 |
| 3 | 3 | Pencil | 100 | Blue | 1 |
| 4 | 4 | Pencil | 100 | Red | 1 |
| 5 | 5 | Pencil | 200 | Yellow | 5 |
| 6 | 6 | Cup | 300 | Orange | 6 |
| 7 | 7 | Cup | 400 | Brown | 7 |

یادآوری: زمانی که دورن Order by ترتیب صعودی یا نزولی بودن را تعیین نکنیم، Order by بصورت پیش فرض صعودی میباشد.

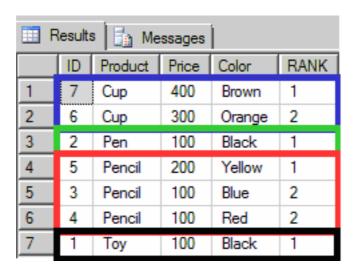
همانطور که در شکل مشاهده مینمایید،رتبه بندی انجام شده به ترتیب نمیباشد، و برای مقادیر تکراری فیلد Price از Rank یکسانی استفاده شده است. نکته دیگر این که بین اعداد مشاهده شده در فیلد Rank نیز gap ایجاد میشود. به عبارت دیگر عمده تفاوت تابع Rank با تابع Row Number همین مواردی است که بیان شده است.

در Syntax تابع Rank نیز کلمه Partition هم وجود دارد، که در جهت Group by فیلد یا فیلدهای خاصی استفاده میشود، و رتبه بندی نیز در این حالت روی Group by انجام می *گر*دد.

برای درک بهتر Query زیر را اجرا نمایی:

Select *,RANK() over (Partition by Product ORDER BY Price Desc) AS RANK from Test

خروجی بصورت زیر خواهد بود:



همانطور که در شکل مشاهده مینمایید، رتبه بندی روی هر Group by بصورت جداگانه اعمال شده است.

تابع Dense_Rank

این تابع نیز همانند تابع Rank عمل میکند، با این تفاوت که هیچ gap ی بین اعداد آن رخ نمیدهد.

با جرای Query زیر خواهیم داشت:

Select *,dense_RANK() over (ORDER BY Price) AS dense_RANK from Test

خروجی بصورت زیر خواهد بود:

| Results Messages | | | | | | |
|------------------|----|---------|-------|--------|------------|--|
| | ID | Product | Price | Color | dense RANK | |
| 1 | 1 | Toy | 100 | Black | 1 | |
| 2 | 2 | Pen | 100 | Black | 1 | |
| 3 | 3 | Pencil | 100 | Blue | 1 | |
| 4 | 4 | Pencil | 100 | Red | 1 | |
| 5 | 5 | Pencil | 200 | Yellow | 2 | |
| 6 | 6 | Cup | 300 | Orange | 3 | |
| 7 | 7 | Cup | 400 | Brown | 4 | |

همانطور که ملاحظه مینماییدهیچ gap ی بین اعداد Rank ایجاد نشده است.

و برای استفاده از Partition، درتابع Dense_Rank همانند تابعهای دیگر میباشد.

تابع NTILE:

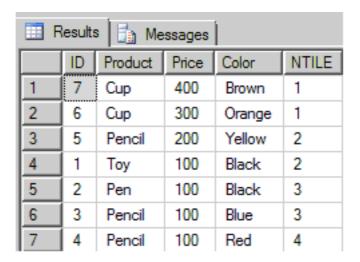
این تابع نیز مانند توابع بالا در جهت رتبه بندی استفاده میشود، و بوسیله تابع فوق شما میتوانید رکوردهای جدول خود را به تعداد گروههای دلخواه تقسیم نمایید.و Syntaxآن بصورت زیر میباشد:

NTILE (integer_expression) OVER ([<partition_by_clause>] <order_by_clause>)

برای درک مطلب فوق مثالی میزنیم:

Select * ,NTILE(4) over (ORDER BY Price desc) from Test

خروجی بصورت زیر خواهد بود:



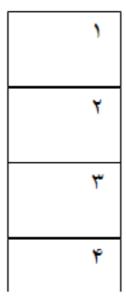
در Syntax تابع فوق اشاره به Integer_Expressionشده است.که یک مقدار عددی دریافت میکند و بیانگر تعداد گروه بندی دلخواه میباشد.

حال سئوال اینجاست که رتبه بندی جدول به چه صورت انجام شده است:

همانطور که مشاهده مینمایید، جدول فوق شامل 7 رکورد میباشد،و ما در مثال خود،تمایل داشتیم که رکوردهای جدول به چهار گروه تقسیم و سپس رتبه بندی شوند، بنابراین 7 تقسیم بر 4 شده است و باقی مانده آن میشود 3

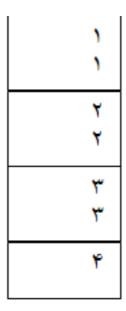
يس خواهيم داشت7=3+1*4

در ابتدا چهار گروه ایجاد میشودو در هر خانه یک رکورد قرار میگیرد



سپس 3 رکورد باقی میماند که از اولین گروه رو به پایین ، برای هر گروه فقط یک رکورد درج میشود، یعنی یک رکورد به گروه

یک،یک رکورد به گروه 2 و هم چنین یک رکورد به گروه 3 بنابراین خواهیم داشت:



نکته مهم: اگر تعداد رکورد باقی مانده بعد از تقسیم بیش از یک عدد باشد، در زمان اختصاص دادن به گروه ها، به هر گروه از بالا به پایین فقط یک رکورد اختصاص داده میشود.

مثالی دیگر:

Select *,NTILE(3) over (ORDER BY Price desc) AS NTILE from Test

خروجی:

| Results Messages | | | | | | |
|------------------|----|---------|-------|--------|-------|--|
| | ID | Product | Price | Color | NTILE | |
| 1 | 7 | Cup | 400 | Brown | 1 | |
| 2 | 6 | Cup | 300 | Orange | 1 | |
| 3 | 5 | Pencil | 200 | Yellow | 1 | |
| 4 | 1 | Toy | 100 | Black | 2 | |
| 5 | 2 | Pen | 100 | Black | 2 | |
| 6 | 3 | Pencil | 100 | Blue | 3 | |
| 7 | 4 | Pencil | 100 | Red | 3 | |

در این حالت 7=1+2*3

امیدوارم مطلب فوق مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: حمیدرضا تاریخ: ۲:۵۹ ۱۳۹۱/۰۵/۰۹

مطلب بسیار مفیدی بود. تشکر میکنم.