

برای مطالعه این بخش لازم است، به Syntax مربوط به Over آشنا باشیم، در [بخش اول](#) بطور کامل به Syntax مربوط به Over پرداختیم.

در این بخش دو فانکشن دیگر از توابع تحلیلی (Analytic functions) به نامهای First_Value و Last_Value را بررسی می‌نماییم.

First_Value

این فانکشن نیز همانند دیگر فانکشنهای تحلیلی در نسخه SQL Server 2012 ارائه گردیده است. و اولین مقدار از یک مجموعه مقادیر را بر می‌گرداند. و Syntax آن بصورت ذیل می‌باشد:

```
FIRST_VALUE ( [scalar_expression ]
OVER ( [ partition_by_clause ] order_by_clause [ rows_range_clause ] )
```

شرح Syntax:

1- Scalar_expression : مقدار آن می‌تواند نام یک فیلد یا Subquery باشد.

2- Over : در [بخش اول](#) بطور مفصل آن را بررسی نمودیم.

قبل از بررسی تابع First_Value، ابتدا Script زیر را اجرا نمایید، که شامل یک جدول و درج چند رکورد در آن است.

```
Create Table Test_First_Last_Value
(SalesOrderID int not null,
SalesOrderDetailID int not null ,
OrderQty smallint not null);
GO
Insert Into Test_First_Last_Value
Values (43662,49,1),(43662,50,3),(43662,51,1),
(43663,52,1),(43664,53,1),(43664,54,1),
(43667,77,3),(43667,78,1),(43667,79,1),
(43667,80,1),(43668,81,3),(43669,110,1),
(43670,111,1),(43670,112,2),(43670,113,2),
(43670,114,1),(43671,115,1),(43671,116,2)
```

مثال: ابتدا Scriptی ایجاد می‌نماییم، بطوریکه جدول Test_Firts_Last_Value را براساس فیلد SalesOrderID گروه بندی نموده و اولین مقدار فیلد SalesOrderDetailID در هرگروه را مشخص نماید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
ORDER BY SalesOrderDetailID) FstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	FstValue
1	43663	52	1	52
2	43667	77	3	77
3	43667	78	1	77
4	43667	79	1	77
5	43667	80	1	77
6	43669	110	1	110
7	43670	111	1	111
8	43670	112	2	111
9	43670	113	2	111
10	43670	114	1	111

مطابق Script چهار گروه در خروجی ایجاد شده است و در فیلد FstValue ، اولین مقدار هر گروه نمایش داده می‌شود. اگر بخش‌های قبلی Window Function ها را مطالعه کرده باشید، تحلیل این تابع کار بسیار ساده ای است.

Last_Value

این تابع نیز در نسخه SQL Server 2012 ارائه گردیده است. و آخرین مقدار از یک مجموعه مقادیر را بر می‌گرداند، به عبارتی فانکشن Last_Value عکس فانکشن First_Value عمل می‌نماید و Syntax آن به شرح ذیل می‌باشد:

```
LAST_VALUE ( [scalar_expression]
            OVER ( [ partition_by_clause ] order_by_clause rows_range_clause )
```

شرح Syntax تابع Last_Value شبیه به تابع First_Value می‌باشد.

مثال: همانند مثال قبل Script یی ایجاد می‌نماییم، بطوریکه جدول Test_Firts_Last_Value را براساس فیلد SalesOrderID گروه بندی نموده و آخرین مقدار فیلد SalesOrderDetailID در هر گروه را مشخص نماید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
       LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
       ORDER BY SalesOrderDetailID RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	LstValue
1	43663	52	1	52
2	43667	77	3	80
3	43667	78	1	80
4	43667	79	1	80
5	43667	80	1	80
6	43669	110	1	110
7	43670	111	1	114
8	43670	112	2	114
9	43670	113	2	114
10	43670	114	1	114

خروجی جدول، به چهار گروه تقسیم، و آخرین مقدار هر گروه، در فیلد LstValue نمایش داده شده است. در این مثال نیز تحلیلی نخواهیم داشت، چون فرض بر آن است که بخش‌های قبلی را مطالعه نموده ایم.

موفق باشید.

نظرات خوانندگان

نویسنده: محمد

تاریخ: ۱۰:۳۹ ۱۳۹۱/۱۰/۲۸

سلام،

مطلب اول: قسمت order by در ماده over در هر دو کوئری به چه جهت آمده است؟

مطلب دوم: first_value چه مزیتی نسبت به min() over دارد، منظورم اینه که میشه خروجی کوئری اولتون رو با این کوئری بدست آورد:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
      MIN(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID) FstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

نویسنده: محمد

تاریخ: ۱۲:۱۱ ۱۳۹۱/۱۰/۲۸

سلام،

من SQL Server 2012 ندارم، ولی تا اونجایی که متوجه شدم بر اساس شواهد دو کوئری زیر باید یک نتیجه رو برگردانند. منظورم اینکه که با first_value همیشه last_value هم شبیه سازی کرد، فقط کافیست که در ماده order by از کلید واژه DESC استفاده بشه. اگه من اشتباه میکنم لطفا راهنمایی بفرمایید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
      LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
      ORDER BY SalesOrderDetailID) LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty

SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
      FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
      ORDER BY SalesOrderDetailID DESC) FstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

نویسنده: فرهاد فرهمندخواه

تاریخ: ۱۱:۵۳ ۱۳۹۱/۱۰/۲۹

سلام

جواب سؤال اول: در Syntax تابع First_value استفاده از Order by اجباری می باشد.

جواب سؤال دوم:

First_Value اولین مقدار یا اولین Row در یک گروه را مشخص می کند و به مفهوم کوچکترین مقدار نمی باشد، شاید، مثالی که در مقاله زدم شما را به اشتباه انداخت، در زیر با یک مثال، First_value و Min را مقایسه می کنیم.

ابتدا یک جدول و چند رکورد، در آن درج می کنیم:

```
CREATE TABLE Employees (
  EmployeeId INT IDENTITY PRIMARY KEY,
  Name VARCHAR(50),
  HireDate DATE NOT NULL,
  Salary INT NOT NULL
)
```

GO

```
INSERT INTO Employees (Name, HireDate, Salary)
VALUES
    ('Alice', '2011-01-01', 20000),
    ('Brent', '2011-01-15', 19000),
    ('Carlos', '2011-02-01', 22000),
    ('Donna', '2011-03-01', 25000),
    ('Evan', '2011-04-01', 18500)
GO
```

در ادامه Script زیر را اجرا می‌کنیم:

```
Select EmployeeId, Name, Salary, HireDate,
    First_VALUE(HireDate) OVER(ORDER BY Salary RANGE BETWEEN UNBOUNDED PRECEDING
                                AND UNBOUNDED FOLLOWING) AS First,
    Min(HireDate) OVER(ORDER BY Salary
                        RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Min
FROM Employees
ORDER BY EmployeeId
GO
```

خروجی بصورت زیر می‌شود:

	EmployeeId	Name	Salary	HireDate	First	Min
1	1	Alice	20000	2011-01-01	2011-04-01	2011-01-01
2	2	Brent	19000	2011-01-15	2011-04-01	2011-01-01
3	3	Carlos	22000	2011-02-01	2011-04-01	2011-01-01
4	4	Donna	25000	2011-03-01	2011-04-01	2011-01-01
5	5	Evan	18500	2011-04-01	2011-04-01	2011-01-01

در شکل بالا تفاوت Min و First_Value بطور کامل مشخص است، اگر به Query دقت نمایید، Sort براساس Salary انجام شده است، برای حالت First_value مقدار فیلد HireDate در اولین رکورد، برابر است با 2011-04-01، بنابراین سورت روی نمایش First_value تاثیر گذار است، بطوریکه Sort برای حالت Min، تاثیر گذار نمی‌باشد، و تابع Min، کوچکترین مقدار، از مقادیر ستون HireDate را بدست می‌آورد، به بیان ساده‌تر در حالت استفاده از Min، عملیات Sort بیهوده می‌باشد. چون تابع MIN روی کل مقادیر یک گروه یا ستون تاثیر می‌گذارد.

نویسنده: فرهاد فرهمندخواه

تاریخ: ۱۳۹۱/۱۰/۲۹ ۱۲:۵۱

سلام

شما می‌توانید، با دستکاری Query ها خروجی‌های یکسانی را ایجاد نمایید، دو Query که ایجاد نمودید، خروجی یکسانی ندارند. Query دوم شما با خروجی Last_Value، مقاله یکسان است، اما باید بگویم که مفهوم Last_Value این است که آخرین سطر در یک گروه را بر می‌گرداند. بهتر است [بخش اول](#) را مطالعه نمایید.

علت یکسان نبودن نتیجه دو Query شما در نحوه Sort و مفهوم First_value و Last_Value می‌باشد:

نکته: اگر در Over Clause شرط Order by اعمال نماییم، اما از Row یا Range استفاده نکنیم، SQL Server بصورت پیش فرض از قالب زیر استفاده می‌نماید:

RANGE UNBOUNDED PRECEDING AND CURRENT ROW

نویسنده: محمد

تاریخ: ۱۳:۲۳ ۱۳۹۱/۱۰/۲۹

ممون از پاسختون، الان متوجه تفاوتشون شدم.

ستون older_method باید مقادیرش دقیقا مشابه با first_value شما باشد:

```
Select EmployeeId,Name,Salary,HireDate,
      First_VALUE(HireDate) OVER(ORDER BY Salary RANGE BETWEEN UNBOUNDED PRECEDING
                                AND UNBOUNDED FOLLOWING) AS First,
      D.HireDate AS older_method
FROM Employees
CROSS APPLY (SELECT TOP 1 HireDate
              FROM Employees
              --WHERE E1.EmployeeId = E2.EmployeeId
              ORDER BY Salary ASC) AS D
ORDER BY EmployeeId;
```

نویسنده: محمد

تاریخ: ۱۳:۳۱ ۱۳۹۱/۱۰/۲۹

ممون از شما، من مطالب بخش اول رو مطالعه کردم.

عبارت RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING به معنای تمام سطرهاى جدول هست دیگه درسته. یعنی تمام سطرهاى جدول از اولین گرفته، جاری گرفته و آخرین رو پوشش میده.

با این توضیحات باید دو کوئری زیر اینبار جواب یکسانی بدهند:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
      FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
      ORDER BY SalesOrderDetailID RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
      LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty

SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
      LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
      ORDER BY SalesOrderDetailID DESC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
      LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

دو کوئری کاملا یکسان هستند به غیر از اینکه در کوئری دوم یک DESC اضافه شده و نام تابع از first به last تغییر کرده است.