عنوان: بدست آوردن برگهای یک درخت توسط Recursive CTE

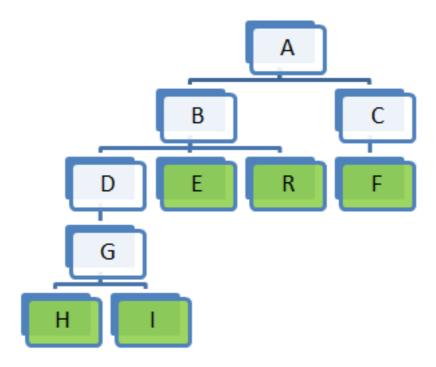
نویسنده: محمد سلیم ابادی تاریخ: ۱۷:۳۵ ۱۳۹۱/۱۱/۱۶

آدرس: www.dotnettips.info

برچسبها: SQL Server, recursive cte, query, not exists, dense_rank

امروز در یک تالار سوالی مطرح شد با این عنوان "چگونه میتوانم گرههای برگ یک شاخه را بدست بیاورم". خب راه حلی که فورا به ذهنم رسید استفاده از یک query بازگشتی (recursive) بود.

به ساختار سلسله مراتبی زیر توجه بفرمایید:



گره هایی که با رنگ سبز علامت گذاری شده اند را گرههای برگ مینامیم چون که آن گرهها بدون زیر شاخه هستند. فرض کنید از ما خواسته شده است با داشتن گره A تمام برگهای این شاخه را بدست بیاوریم.

دو مرحله را باید طی کنیم ابتدا تمام گره هایی که زیر شاخه گره A هستند را باید بدست آورد سپس توسط یک گزاره گرههای برگ را فیلتر کنیم.

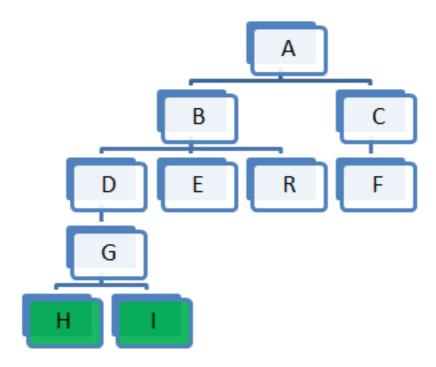
در واقع گره هایی برگ هستند که یدر هیچ گرهی دیگری نباشند.

```
;with cte as
(
select id, rnk=0 from @t
where parent = 'A'
```

```
union all

select t.id, rnk+1
from cte join @t t
on cte.id = t.parent
)
select *
   from cte
where not exists
   (select *
        from @t
        where parent = cte.id);
```

و حالا به درخت زیر توجه بفرمایید:



هدف پیدا کردن برگ هایی از شاخه مورد نظر است که در پایین ترین سطح قرار گرفته باشند. برای این منظور از همان query بازگشتی استفاده کرده و با کمک تابع dense_ranke گرههای مورد نظر را بدست میاوریم.

```
;with cte as
(
select id, rnk=0 from @t
where parent = 'A'
union all
select t.id, rnk+1
from cte join @t t
on cte.id = t.parent
)
select *
from
(
    select *, dense_rank() over(order by rnk desc) rk
    from cte
)t
where rk = 1
```

نظرات خوانندگان

```
نویسنده: بهمن خلفی
تاریخ: ۱۴:۴۱ ۱۳۹۱/۱۱/۱۷
```

ممنون از مطلب مفیدتون

اما یک سوال : چگونه میتوان به نتیجه زیر دست پیدا کرد

a , ab , ac , abd , abe , abr , abdg , abdgh , \dots

```
نویسنده: محمد سلم ابادی
تاریخ: ۱۵:۳۰ ۱۸۹۱/۱۱/۱۷
```

من دقیقا متوجه نشدم نتیجه مورد نظر شما چیست.

آیا نتیجه مورد نظر شما به صورت الحاق یافته (concatenated) هست یا نه؟

در هر صورت باید یکی از دو query زیر نتیجه مورد نظر شما را تولید کند.

```
declare @t table
(id char(1) primary key,
parent char(1));
insert @t values
('A',null), --Level 1
('B', 'A'), ('C', 'A'), --Level 2
('D', 'B'), ('E', 'B'), ('R', 'B'), ('F', 'C'), --Level 3
('G', 'D'), --Level 4
('H', 'G'), ('I', 'G'); --Level 5
;with cte as
select id, rnk=0,
       concats = cast(id as varchar(10))
from @t
where parent is null
union all
from cte join @t t
on cte.id = t.parent
select * from cte
id
     rnk
                   concats
В
     1
                   ΑB
                   AC
C
     1
F
                   ACF
D
                   ABD
Ε
                   ABE
R
                   ABR
                   ABDG
G
     3
Н
     4
                   ABDGH
Ι
     4
                   ABDGI
;with cte as
select id, rnk=0,
       concats = cast(id as varchar(10))
from @t
where parent is null
union all
from cte join @t t on cte.id = t.parent
```

موفق باشيد