

امروز در یک تالار سوالی مطرح شد با این عنوان "چگونه می‌توانم گره‌های برگ یک شاخه را بدست بیاورم". خب راه حلی که فوراً به ذهنم رسید استفاده از یک query بازگشتی (recursive) بود. به ساختار سلسله مراتبی زیر توجه بفرمایید:



گره‌هایی 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 گره‌هایی که زیر شاخه گره A هستند را باید بدست آورد سپس توسط یک گزاره گره‌های برگ را فیلتر کنیم.

در واقع گره‌هایی برگ هستند که پدر هیچ گره‌ی دیگری نباشند.

```

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
    
```

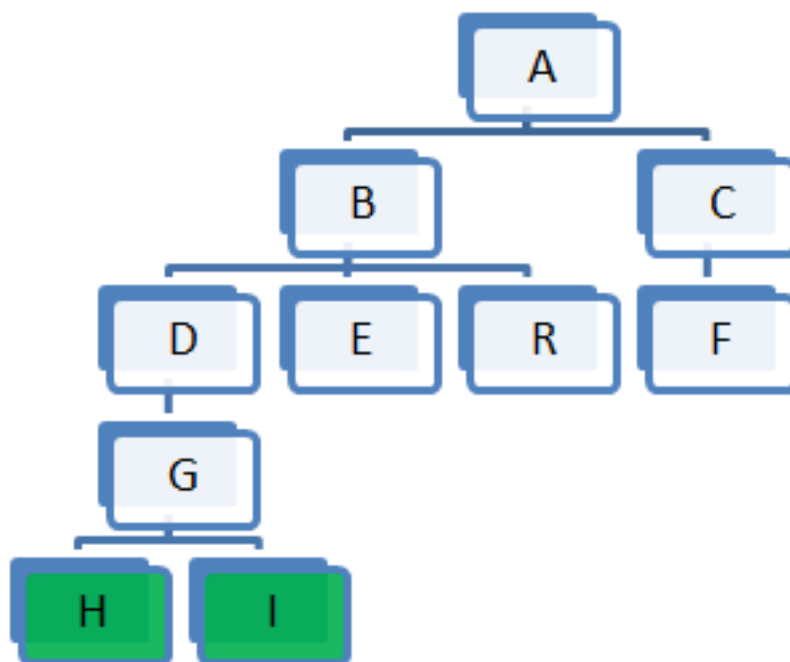
```

;with cte as
(
select id, rnk=0 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\_rank() مورد نظر را بدست میآوریم.

```
;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: ۱۳۹۱/۱۱/۱۷ ۱۴:۴۱

ممنون از مطلب مفیدتون  
اما یک سوال : چگونه میتوان به نتیجه زیر دست پیدا کرد

a , ab , ac , abd , abe , abr , abdg , abdgh , ....

نویسنده: محمد سلم آبادی  
تاریخ: ۱۳۹۱/۱۱/۱۷ ۱۵:۳۰

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,
concat = cast(id as varchar(10))
from @t
where parent is null

union all

select t.id, rnk+1,
cast(cte.concat + t.id as varchar(10))
from cte join @t t
on cte.id = t.parent
)
select * from cte
/*
id    rnk    concat
-----
A      0      A
B      1      AB
C      1      AC
F      2      ACF
D      2      ABD
E      2      ABE
R      2      ABR
G      3      ABDG
H      4      ABDGH
I      4      ABDGI
*/

;with cte as
(
select id, rnk=0,
concat = cast(id as varchar(10))
from @t
where parent is null

union all

select t.id, rnk+1,
cast(cte.concat + t.id as varchar(10))
from cte join @t t
on cte.id = t.parent
)
```

```
select stuff(d.list,1,1,'') as concats
from (select ','+concats
      from cte
      for xml path(''))d(list)
/*
concats
-----
A,AB,AC,ACF,ABD,ABE,ABR,ABDG,ABDGH,ABDGI
*/
```

موفق باشید