

Exploitation des fonctions Row_Number() et Rank() de SQL Server 2005.

Imaginez que vous ayez une table de ce type, qui donne les volumes de stocks (ou de vente pour ce que j'en sais) par produits.

Categorie	Produit	Volume
Légumes	Choux	7
Légumes	Choux-fleur	12
Légumes	Tomates	15
Légumes	Carottes	9
Légumes	Haricots	4
Légumes	Courgettes	18
Fruits	Bananes	11
Fruits	Pommes	18
Fruits	Poires	14
Fruits	Raisin	23
Fruits	Oranges	19
Confiseries	Malabar	11
Confiseries	Carambar	19
Confiseries	Roudoudou	12
Charcuterie	Saucisson	2
Charcuterie	Paté	3
Charcuterie	Jambon	4

Votre chef (bien aimé) vous demande le classement des plus gros stocks par catégories. Sous SQL 2000 vous auriez été obligé de recourir à des sous requêtes compliquées, tandis que SQL 2005 apporte son lot de fonctions bien pratiques : ROW_NUMBER et RANK

ROW_NUMBER donne le numéro de ligne, selon un ordre de tri, tandis que RANK donne le rang (valeur égale => même rang).

La syntaxe est un peu inhabituelle mais on s'y fait très vite :

ROW_NUMBER() over (ORDER BY champ) renvoie le N° de ligne en triant sur champ.

Ici, en triant sur le volume cela donne :

```
select *, row_number() over(order by volume)
from ventes
```

Categorie	Produit	Volume	N° ligne
Charcuterie	Saucisson	2	1
Charcuterie	Paté	3	2
Charcuterie	Jambon	4	3
Légumes	Haricots	4	4
Légumes	Choux	7	5
Légumes	Carottes	9	6
Confiseries	Malabar	11	7
Fruits	Bananes	11	8
Légumes	Choux-fleur	12	9
Confiseries	Roudoudou	12	10
Fruits	Poires	14	11
Légumes	Tomates	15	12

Légumes	Courgettes	18	13
Fruits	Pommes	18	14
Confiseries	Carambar	19	15
Fruits	Oranges	19	16
Fruits	Raisin	23	17

Le rang, de son côté donne :

Categorie	Produit	Volume	Rang
Charcuterie	Saucisson	2	1
Charcuterie	Paté	3	2
Charcuterie	Jambon	4	3
Légumes	Haricots	4	3
Légumes	Choux	7	5
Légumes	Carottes	9	6
Confiseries	Malabar	11	7
Fruits	Bananes	11	7
Légumes	Choux-fleur	12	9
Confiseries	Roudoudou	12	9
Fruits	Poires	14	11
Légumes	Tomates	15	12
Légumes	Courgettes	18	13
Fruits	Pommes	18	13
Confiseries	Carambar	19	15
Fruits	Oranges	19	15
Fruits	Raisin	23	17

Au passage, je signale que l'on n'est pas obligé de trier le résultat selon la clause de tri de la fonction :

```
select *, row_number() over(order by volume)
from ventes
order by categorie, produit
```

Categorie	Produit	Volume	N° ligne
Charcuterie	Jambon	4	3
Charcuterie	Paté	3	2
Charcuterie	Saucisson	2	1
Confiseries	Carambar	19	15
Confiseries	Malabar	11	7
Confiseries	Roudoudou	12	10
Fruits	Bananes	11	8
Fruits	Oranges	19	16
Fruits	Poires	14	11
Fruits	Pommes	18	14
Fruits	Raisin	23	17
Légumes	Carottes	9	6
Légumes	Choux	7	5
Légumes	Choux-fleur	12	9
Légumes	Courgettes	18	13
Légumes	Haricots	4	4
Légumes	Tomates	15	12

Revenons maintenant à la question du départ : le classement des plus gros stocks par catégories.

Facile, il faut ordonner par volume au sein de la catégorie. Essayons comme ceci :

```
select *, row_number() over(order by categorie, volume)
from ventes
```

Categorie	Produit	Volume	N°
Charcuterie	Saucisson	2	1
Charcuterie	Paté	3	2
Charcuterie	Jambon	4	3
Confiseries	Malabar	11	4
Confiseries	Roudoudou	12	5
Confiseries	Carambar	19	6
Fruits	Bananes	11	7
Fruits	Poires	14	8
Fruits	Pommes	18	9
Fruits	Oranges	19	10
Fruits	Raisin	23	11
Légumes	Haricots	4	12
Légumes	Choux	7	13
Légumes	Carottes	9	14
Légumes	Choux-fleur	12	15
Légumes	Tomates	15	16
Légumes	Courgettes	18	17

Pas si facile, finalement. C'est pas mal pour la charcuterie mais il faudrait recommencer à chaque catégorie. Essayons le rang par catégorie :

```
select *, rank() over(order by categorie)
from ventes
```

Categorie	Produit	Volume	Rang
Charcuterie	Saucisson	2	1
Charcuterie	Paté	3	1
Charcuterie	Jambon	4	1
Confiseries	Malabar	11	4
Confiseries	Carambar	19	4
Confiseries	Roudoudou	12	4
Fruits	Bananes	11	7
Fruits	Pommes	18	7
Fruits	Poires	14	7
Fruits	Raisin	23	7
Fruits	Oranges	19	7
Légumes	Choux	7	12
Légumes	Choux-fleur	12	12
Légumes	Tomates	15	12
Légumes	Carottes	9	12
Légumes	Haricots	4	12
Légumes	Courgettes	18	12

Domage, j'obtiens les mêmes valeurs pour chaque item de la catégorie... quoique si je rapproche les deux infos :

```
select *, rank() over(order by categorie), row_number() over(order by
categorie, volume)
from ventes
```

Categorie	Produit	Volume	Rang	N° Ligne
Charcuterie	Jambon	4	1	1
Charcuterie	Paté	3	1	2
Charcuterie	Saucisson	2	1	3
Confiseries	Carambar	19	4	4
Confiseries	Roudoudou	12	4	5
Confiseries	Malabar	11	4	6
Fruits	Raisin	23	7	7
Fruits	Oranges	19	7	8
Fruits	Pommes	18	7	9
Fruits	Poires	14	7	10
Fruits	Bananes	11	7	11
Légumes	Courgettes	18	12	12
Légumes	Tomates	15	12	13
Légumes	Choux-fleur	12	12	14
Légumes	Carottes	9	12	15
Légumes	Choux	7	12	16
Légumes	Haricots	4	12	17

Eureka ! Il ne reste plus qu'à faire la différence entre les deux :

```
select *, (row_number() over(order by categorie, volume desc) ) - (rank()
over(order by categorie))
from ventes
```

Categorie	Produit	Volume	Rang
Charcuterie	Jambon	4	0
Charcuterie	Paté	3	1
Charcuterie	Saucisson	2	2
Confiseries	Carambar	19	0
Confiseries	Roudoudou	12	1
Confiseries	Malabar	11	2
Fruits	Raisin	23	0
Fruits	Oranges	19	1
Fruits	Pommes	18	2
Fruits	Poires	14	3
Fruits	Bananes	11	4
Légumes	Courgettes	18	0
Légumes	Tomates	15	1
Légumes	Choux-fleur	12	2
Légumes	Carottes	9	3
Légumes	Choux	7	4
Légumes	Haricots	4	5

Et si on ne veut que les vainqueurs, un peu de ménage suffit :

```
select categorie, produit, volume
from (
select *, (row_number() over(order by categorie, volume desc) ) -
(rank() over(order by categorie)) as Rank_in_categorie
```

```
        from ventes
    ) ventes_triee
where
    Rank_in_categorie = 0
```

Et voila : les meilleurs ventes (ou les plus gros stocks) par catégorie

Categorie	Produit	Volume
Charcuterie	Jambon	4
Confiseries	Carambar	19
Fruits	Raisin	23
Légumes	Courgettes	18

Merci SQL Server 2005 !