

Base de données Zoo requêtes imbriquées

Examinons comment réaliser des requêtes avancées avec un principe simple nous allons imbriquer les résultats.

1.1 Regrouper des résultats :

<https://learnsql.fr/blog/5-exemples-de-group-by/>

<https://learnsql.fr/blog/clause-group-by-la-connaissiez-vous-bien/>



Nous avons ici le tableau des médailles avec les médaillés de la Coupe du monde de saut à ski pour les quatre dernières saisons.

country	person	season	place
Norway	Anders Bardal	2011-12	1
Austria	Gregor Schlierenzauer	2011-12	2
Austria	Andreas Kofler	2011-12	3
Austria	Gregor Schlierenzauer	2012-13	1
Norway	Anders Bardal	2012-13	2
Poland	Kamil Stoch	2012-13	3
Poland	Kamil Stoch	2013-14	1
Slovenia	Peter Prevc	2013-14	2
Germany	Severin Freund	2013-14	3
Germany	Severin Freund	2014-15	1
Slovenia	Peter Prevc	2014-15	2
Austria	Stefan Kraft	2014-15	3

Q1. Que fait cette requête ?

```
SELECT count(*)  
FROM medals  
WHERE country = 'Poland';
```

Q2. Et celle-ci ?

```
SELECT count(*)  
FROM medals  
WHERE country = 'Germany';
```

Alors comment faire un bilan et compter automatiquement le résultat en médaille de chacun des pays ?

C'est le rôle de la clause `group by`

1.2 La clause `group by`

```
SELECT country, count(*)  
FROM medals  
GROUP BY country;
```

country	count
Poland	2
Germany	2
Austria	4
Norway	2
Slovenia	2

Avec la requête `GROUP BY`, la base de données divise les données en **groupes**. Les lignes ayant la même colonne `GROUP BY` (le pays dans l'exemple) sont placées dans un seul groupe. Donc, en utilisant notre exemple, les médaillés de Pologne sont placés dans un groupe, les médaillés d'Allemagne sont placés dans un autre groupe et ainsi de suite. Voici le **regroupement** que nous obtenons pour cette requête :

country	person	season	place
Poland	Kamil Stoch	2012-13	3
	Kamil Stoch	2013-14	1
Germany	Severin Freund	2013-14	3
	Severin Freund	2014-15	1
Austria	Gregor Schlierenzauer	2012-13	1
	Stefan Kraft	2014-15	3
	Gregor Schlierenzauer	2011-12	2
	Andreas Kofler	2011-12	3
Norway	Anders Bardal	2012-13	2
	Anders Bardal	2011-12	1
Slovenia	Peter Prevc	2013-14	2
	Peter Prevc	2014-15	2

Avec `GROUP BY` les agrégats (`count`, `sum`, `avg`, `min`, `max`, et autres) sont calculés pour chacun séparément. Dans l'exemple, la base de données compte le nombre de lignes dans chaque groupe.

1.3 Requêtes imbriquées

Etudions une interrogation du type :

Donner les années de naissance et les noms des animaux qui ont 2 ou plus de maladies ?

Première étape

```
SELECT nomA FROM LesMaladies;
```

On observe bien que certains animaux ont plusieurs maladies. ils sont tous listés dans les résultats.

	nomA
1	Charlotte
2	Charly
3	Charly
4	Milou
5	Chloé
6	Huan
7	Huan
8	Huan

Deuxième étape on fait un décompte

```
SELECT nomA, COUNT(*) FROM LesMaladies GROUP BY nomA;
```

L'agrégat count(*) décompte les maladies regroupées par nomA. Voir l'exemple ci-dessus.

	nomA	COUNT(*)
1	Charlotte	1
2	Charly	2
3	Chloé	1
4	Huan	3
5	Milou	1

Troisième étape on donne un nom à la colonne créée count(*)

```
SELECT nomA, COUNT(*) AS nbpre  
FROM LesMaladies GROUP BY nomA;
```

	nomA	nbpre
1	Charlotte	1
2	Charly	2
3	Chloé	1
4	Huan	3
5	Milou	1



Première imbrication

```
SELECT nomA
FROM
( SELECT nomA, COUNT(*) AS nbpre
  FROM LesMaladies GROUP BY nomA
)
WHERE nbpre >= 2;
```



	nomA	nbpre
1	Charlotte	1
2	Charly	2
3	Chloé	1
4	Huan	3
5	Milou	1

	nomA
1	Charly
2	Huan

Deuxième imbrication

```
SELECT anNais, nomA
FROM LesAnimaux
WHERE nomA IN
```

```
( SELECT nomA
  FROM
    ( SELECT nomA, COUNT(*) AS nbpre
      FROM LesMaladies GROUP BY nomA
    )
  WHERE nbpre >= 2
);
```



	nomA
1	Charly
2	Huan

D'où le résultat :

Donner les années de naissance et les noms des animaux qui ont 2 ou plus de maladies ?

	anNais	nomA
1	1999	Charly
2	2005	Huan

