



Les fonctions d'agrégation

Les fonctions d'agrégation

- Tous les SGBD disposent des fonctions d'agrégation. Ces fonctions vont nous permettre de faire des regroupements d'enregistrements ou créer des requêtes récapitulatives.
- Il existe 5 fonctions qui sont les suivantes:
SUM, COUNT, MIN, MAX, AVG

La fonction MAX

- **MAX(champ)**: cette fonction va sélectionner la **valeur maximale** du champ entre ()
- Exemple:

Select **MAX(prix)** from produits

Exemple

- Si on a des produits comme suit:

Ref	Nom produit	prix
PR01	produit1	500
PR02	Produit2	489
PR03	Produit3	450
PR04	Produit4	970
PR05	Produit5	360

La requête précédente va sélectionner le 4^{ème} produit

La fonction MIN

- **MIN (champ):** cette fonction va sélectionner la **valeur minimale** du champ
- Exemple:

```
Select MIN(prix) from produits
```

- En se référant au tableau précédant, la valeur à afficher sera : **360**

La fonctions SUM

- **SUM (champ):** permet de faire la **somme** des valeurs d'un **champ** pour tous les enregistrements spécifiées

Exemple 1:

```
Select SUM(prix) from produits
```

Cette requête va afficher la valeur : **2 769**

Exemple 2:

```
Select SUM(prix) from produits where nom like 's%'
```

Cette requête va afficher la somme des prix des produits dont le nom commence par s.

Les fonctions d'agrégation

- **AVG (champ):** cette fonction calcule la **moyenne** de ce champ

- Exemple:

```
Select AVG(prix) from produits
```

- **Va calculer et afficher la moyenne des prix des produits** (soit la somme des prix divisé par le nombre des produits)

La fonction COUNT

- **COUNT(*)**: permet de compter le **nombre de lignes**, c'est-à-dire le nombre d'enregistrements.

- **Exemple:**

```
Select Count(*) from produits
```

- Cette requête va afficher le nombre de produits dans la table produits.
- **Count(champ)**: permet de compter le **nombre de valeurs** dont **champ** est **non null**.
- **C'est-à-dire que** si on met le nom d'un champ, il faut s'assurer qu'il ne prend pas la valeur NULL. Comme la clé primaire par exemple.

Utilisation des aléas

- Un **aléa** est un nom qu'on donne nous même à une valeur calculé.
- Exemple:

```
Select AVG(prix) AS 'Moyenne' from produits
```

Moyenne

144.33181818

Utilisation des aléas

- On peut utiliser les aléas aussi pour les tables. Cette méthode paraît intéressante lorsque le nom de la table est long.
- Exemple:

```
Select * from produits as T where T.prix =500
```

La clause GROUP BY

- Une des utilisations les plus courantes de la clause GROUP BY est son **association avec une fonction d'agrégation**.
- Elle permet de **regrouper** des enregistrements **selon un champ** ou plusieurs champs.

La clause GROUP BY

- Exemple 1:

```
Select COUNT(*) as 'nombreDeProduit' , indisponible  
from produits GROUP BY indisponible
```

On veut afficher le nombre de produits groupé par indisponibilité. C'est-à-dire le nombre de produits disponibles et le nombre de produits indisponible.

La clause GROUP BY

- Exemple 2:

```
Select COUNT(*) as 'nombreDeProduit' , indisponible  
from produits GROUP BY indisponible ORDER BY count(*)  
desc
```

Dans cette requête on veut afficher le nombre de produits disponibles et le nombre de produits indisponible mais triés par nombre de produits dans le sens décroissant.

La clause GROUP BY

- La requête suivante:

```
Select COUNT(*) as 'nombreDeProduit' ,NomDuProduit  
from produits GROUP BY indisponible ORDER BY  
COUNT(*) desc
```

- Génère une erreur dans certains SGBD comme SQL SERVER et autres, mais mysql ne génère pas de problème sauf que le résultat affiché n'est pas cohérent.

Le champ qu'on peut sélectionner doit être un champ de regroupement.

La clause GROUP BY

- **solution:** soit on supprime le nom du produit dans la sélection, ou bien on l'ajoute dans les champs de regroupement.

```
Select COUNT(*) as 'nombreDeProduit' ,NomDuProduit  
from produits GROUP BY indisponible, NomDuProduit  
ORDER BY COUNT(*) desc
```

la clause HAVING

- On peut ajouter une condition sur les valeurs de regroupement en utilisant la clause **HAVING**
- Exemple:

```
Select count(*) from produits GROUP BY  
indisponible HAVING count(*) >10
```


Différence entre WHERE et HAVING

- La clause **where** est faite pour filtrer les lignes c'est-à-dire que la condition porte sur les champs de la table.
- La clause **having** est faite pour réaliser des conditions sur une fonction de regroupement(count, max, min, sum et avg).

having et where

- Exemple contenant les deux clauses:

```
Select  count(*) as 'nombre' , indisponible from  
produits where `prix unitaire` =300 GROUP BY  
indisponible Having count(*) >10
```