

TP 7 : Les regroupement

Rappel du contexte d'étude "magasin".

L'entreprise que nous allons étudier est un commerce de proximité de type "supérette" organisé en rayons contenant des articles de diverses marques. Lors du passage en caisse, on conserve la date, le détail des articles achetés ainsi que le montant total qui a été réglé pour cet achat. On conserve également les informations concernant les fournisseurs et les livraisons. Pour chaque livraison, on mémorise le fournisseur ainsi que le ou les lots qui ont été livrés. Chaque lot concerne un seul article. Un lot est identifié par son numéro de lot et par le code-barres de l'article qu'il concerne.

Un article possède un identifiant unique, le code-barres. C

'est son appartenance à un lot qui permet de retrouver les informations qui le concernent (dates de fabrication, de livraison et de péremption).

Les regroupements (GROUP BY)

Le regroupement est une opération qui nous permet de spécifier un ou plusieurs champs qui déterminent l'appartenance à un groupe.

En SQL, c'est l'instruction "**GROUP BY**" qui effectue des regroupements de toutes les lignes dont la valeur du champ de groupe est identique.

Exemple, dans la table **Animaux** ci-dessous contient deux colonnes.

Nom	Espèce
Berlioz	Félin
Simba	Félin
Louis	Singe
KingKong	Singe
Nala	Félin
Marcel	Singe
Cesar	Singe

Nous souhaitons compter le nombre d'animaux pour chaque espèce. Pour cela nous allons faire la requête suivante.

```
SELECT Espèce, COUNT(*) FROM Animaux  
GROUP BY Espèce
```

Le résultats de cette requête sera :

Espèce	Count(*)
Félin	3
Singe	4

Autre exemple :

Nous souhaitons obtenir le nombre d'articles par rayon d'appartenance

En SQL, sans regroupement

(Attention, cette requête provoque une erreur)

```
SELECT idRayon, COUNT(*) AS Nb_Art  
FROM article
```

Testez la requête

Résultat Théorique :

idRayon	Nb art.
5	1
5	1
5	1
4	1
14	1
14	1
...	...

La requête **provoquant une erreur**, on va maintenant utiliser la fonction **GROUP BY**

```
SELECT idRayon, COUNT(*) AS Nb_Art
FROM article
GROUP BY idRayon
```

Résultat :

idRayon	Nb art.
5	3
4	1
14	2
...	...

Testez la requête

Grâce la fonction « **GROUP BY** », on a pu compter le **nombre d'article par Rayon**, dans le cas d'une requête « `SELECT COUNT(*) FROM article` », celle-ci aurait ramené **une seule ligne avec le nombre d'articles total**.

Nous souhaitons maintenant obtenir le nombre d'articles par rayon d'appartenance avec la référence et le libellé du rayon :

Ce qui nous donne, en SQL :

```
SELECT article.idRayon, rayon.libelle,
       COUNT(*) AS "Nb art."
FROM article JOIN rayon ON
       (article.idRayon = rayon.idRayon)
GROUP BY article.idRayon, rayon.libelle
```



Profitions de cette requête pour faire quelques révisions :

Les champs sont préfixés car il existe un champ « idRayon » dans les 2 tables présentes dans le "FROM". Le moteur d'exécution a donc besoin de savoir avec précision quels sont les champs à utiliser. Même chose pour le champ "libelle" dont le nom existe dans les 2 tables.

La présence d'une fonction dans le "SELECT" contraint à faire un

regroupement sur tous les autres champs à l'aide d'un "GROUP BY"

Exercice

Réaliser les requêtes permettant d'obtenir les informations ci-dessous.

- Le nombre d'articles par achat
- Le nombre d'articles par rayon
- Le nombre de lots livrés par fournisseur
- La liste des articles avec le nombre de fois où ils ont été achetés en affichant les plus "populaires" en premier ;
- Le prix moyen au kilo des articles par rayon.