

1] L'organisation des données

Organiser les données consiste à regrouper suivant un ordre logique, des lignes de données (dont une ou plusieurs colonnes sont identiques) par rapport (et souvent) à un traitement.

- **La clause GROUP BY**

Cette clause classe les données identiques par groupe. La clause `GROUP BY` suit la clause `WHERE` (s'il y en a une) et précède la clause `ORDER BY` (s'il y en a une). La logique veut que le groupe se trouve aussi après la clause `SELECT`. On peut également regrouper un calcul effectué par des opérateurs et qualifié par un alias de colonne (voir les exemples). En fait la clause `GROUP BY` s'utilise très souvent avec les fonctions d'agrégation.

Remarque : Tous les champs figurant dans le `GROUP BY` doivent figurer dans le `SELECT`

```
... GROUP BY colonne1 [ , colonne2, ... ] ... ;
```

- **La clause HAVING**

Cette clause est à la clause `GROUP BY` ce que `WHERE` est à la clause `SELECT`. Elle donne les conditions sur les groupes créés par `GROUP BY`. Elle se situe évidemment juste après la clause `GROUP BY`.

```
... HAVING ( condition1 [ , condition2, ... ] ... );
```

2] Utilisation des tris et regroupements

Nous allons utiliser une version de la table `legumes` avec une colonne supplémentaire : la colonne `grossiste`.

Table `legumes`

num	nom	prix au kilo (en €)	quantité (kg)	grossiste
1	Pommes de terre	1,5	200	Jean Népasler
2	Carotte	1,6	300	Alain Térieur
3	Oignon	4,1	150	Alain Térieur
4	Céleri	4,8	200	Jean Népasler
5	Poireau	4,6	150	Dino Resto
6	Chou blanc		300	Alex Térieur
7	Chou rouge	3,1	300	Alex Térieur

- **Un regroupement par grossiste**

```
SELECT grossiste, SUM(quantite) FROM legumes GROUP BY grossiste;
```

Ligne	grossiste (varchar)	sum (int8)
1	Alain Térieur	450
2	Dino Resto	150
3	Alex Térieur	600
4	Jean Népasler	400

- **Le regroupement du chiffre d'affaires par un alias**

```
SELECT grossiste, SUM(prix*quantite) as ca FROM legumes GROUP BY grossiste;
```

Ligne	grossiste (varchar)	ca (float8)
1	Alain Térieur	1095
2	Dino Resto	690
3	Alex Térieur	930
4	Jean Népasler	1260

- **Expliquez la requête...**

Cette fois-ci trouvez l'explication de la requête suivante :

```
SELECT grossiste, sum(prix*quantite) AS ca
FROM legumes
GROUP BY grossiste
HAVING ca>=1000
ORDER BY ca;
```

grossiste	Total
Alain T?rieur	1094.99999284744
Jean N?pasler	1260.00003814697

Réponse :

.....

.....

.....

.....

- **Sous-requête renvoyant une table dans le From de la requête principale**

```
SELECT max(ca) AS max_ca
FROM (
    SELECT sum(prix*quantite) AS ca
    FROM legumes GROUP BY grossiste) T
```

ATTENTION : lorsque l'on place une sous requête en tant que table dans la clause FROM d'une requête, il faut pouvoir donner un nom à cette table ne serait-ce que parce qu'elle peut être jointes aux autres. Il convient donc de lui donner systématiquement un surnom. Ici le surnom choisit a été la lettre T, comme Table.