

# TP 5: FONCTIONS DE GROUPE

## OBJECTIFS :

Comprendre le principe de regroupement et les fonctions de groupe d'oracle.

## Partie I: présentation

### I.1. Les fonctions de groupe:

Les fonctions de groupe sont utilisées pour afficher des informations sur un ensemble d'enregistrement.

Les fonctions de groupe : AVG, COUNT, MAX, MIN, SUM

```
SELECT fonctionGroupe(colonne), ...
FROM table
[WHERE condition]
```

Le tableau suivant présente les principales fonctions. L'option DISTINCT évite les duplicatas alors que ALL les prend en compte (par défaut). À l'exception de COUNT, toutes les fonctions ignorent les valeurs NULL (il faudra utiliser NVL pour contrer cet effet).

Fonction	Objectif
AVG ([DISTINCT   ALL] <i>expr</i> )	Moyenne de <i>expr</i> (nombre).
COUNT ({*   [DISTINCT   ALL] <i>expr</i> })	Nombre de lignes (* toutes les lignes, <i>expr</i> pour les colonnes non nulles).
MAX ([DISTINCT   ALL] <i>expr</i> )	Maximum de <i>expr</i> (nombre, date, chaîne).
MIN ([DISTINCT   ALL] <i>expr</i> )	Minimum de <i>expr</i> (nombre, date, chaîne).
SUM ([DISTINCT   ALL] <i>expr</i> )	Somme de <i>expr</i> (nombre).

### Application: base de données "Vente"

- Afficher le nombre des clients:

```
.....
```

- Quelle est la somme des chiffres d'affaires de tous les clients,

```
.....
```

- Quelle est la moyenne des prix unitaire de tous les produits

```
.....
```

4. Quelle est la quantité maximale et minimale des produits en stock

.....  
 .....

5. Quelle est la date de création de la dernière commande du client numéro 1250

.....  
 .....

6. Quelle est le nombre de clients qui ont commandés des commandes.

.....  
 .....

**I.2. Utilisation de GROUPE BY:**

La clause groupe by permet de diviser les enregistrement d'une table en groupes.

```
SELECT [colonne,] fonctionGroupe(colonne), ...
FROM table
[WHERE condition]
[GROUP BY column]
[ORDER BY column];
```

- Toutes les colonnes dans la liste SELECT qui ne sont pas des paramètres de la fonction de groupe doivent figurer dans la clause GROUP BY
- Les alias de colonnes ne peuvent pas être utilisés dans la clause GROUP BY.

**Application:**

1. Quelle est la somme des quantités commandées par produit,

.....  
 .....

2. Quel est le produit (code et libellé) le plus cher,

.....  
 .....

3. Combien de fois chaque article a-t-il été commandé ?

.....  
 .....

4. Afficher les numéros des commandes regroupées par clients et triées suivant la date de création (on commence par la dernière commande)

.....  
 .....

5. Afficher la somme des factures de chaque client

.....  
 .....

6. Quel est le nombre de produit par commande ?

.....  
 .....

### I.3. Utilisation de HAVING:

On ne peut pas utiliser des fonctions de groupes dans la clause WHERE.

☞ La clause HAVING agit comme le filtre WHERE, mais permet de filtrer non plus les données, mais les opérations résultant des regroupements

```

SELECT [colonne,] fonctionGroupe(colonne), ...
FROM table
[WHERE condition]
[GROUP BY column]
[HAVING groupe_condition]
[ORDER BY column];
    
```

**Application:**

1. Quels sont les produits (code) commandés plus de 10 fois ?  
.....
2. Quels sont les produits (code) dont la moyenne des quantités commandées est supérieure à 50 ?  
.....
3. Quels sont les clients (code) qui ont lancé plus que 2 commandes,  
.....
4. Afficher le nombre des produits qui ont un prix unitaire supérieur à la moyenne.  
.....
5. Quels sont les clients qui ont des factures en 2014 dont le total dépasse 1000.  
.....
6. Afficher combien de fois le produit numéro 2 a été commandé dans des commandes dont le nombre de ligne dépasse 10.  
.....
7. Regrouper les produits qui ont le même prix unitaire (commençant par le plus élevé) et un seuil supérieur à 10 et la quantité en stock inferieur à la moyenne des quantités.  
.....

## Partie II: *Exercice*

Soit le schéma de base de données relationnel suivant :

**AGENCE** (NumAgence, Nom, Ville)

**CLIENT** (NumClient, Nom, Ville, dateNaiss)

**COMPTE** (NumCompte, #NumAgence, #NumClient, Solde)

Écrire les requêtes SQL suivantes :

1. Créer la table Compte avec les différentes contraintes sachant que le solde doit être obligatoirement  $\geq 0$ .
2. Liste les numéro des agences ayant des comptes-clients.
3. Afficher le solde moyen des comptes-clients des agences dont le solde moyen est  $> 10\,000$ .
4. Afficher le nombre de clients habitant à "Tunis".
5. Afficher le nombre de clients n'ayant pas d'adresse.
6. Insérer le Client dont le numéro est 100 et de nom="Tunssi" avec son compte 2000 créé a l'agence numéro 5014 et de solde initial = 0.
7. Augmenter le solde de tous les comptes de l'agence numéro 5014 de "5%".
8. Afficher le total des comptes de chaque agence en commençant par le plus grand dont le nombre de client dépasse 10,
9. Supprimer de la relation Compte toutes les comptes vides.
10. Quel est le client le plus vieux.
11. Quel est le total des comptes du client numéro 100 à l'agence 5014.
12. Afficher les clients nés en mois d'octobre et qui habitent à 'nabeul' (ou bien 'NABEUL' et faite attention aux espaces) e dont le nom se termine par 'ed'.