II.2) Recherche de données : La recherche peut consister à effectuer :

- ✓ La …….. sur certaines (une ou plusieurs ou la totalité) des colonnes d'une table.
- ✓ Une …….. sur certaines lignes d'une table.
- ✓ Une …….. sur deux ou plusieurs tables.
- ✓ Toutes …….. de projection, sélection et jointure.

La commande permettant d'effectuer une recherche dans une base de données est la commande **SELECT**.

a) Recherche de colonnes à partir d'une table : « projection »

La forme générale de cette commande est

Activité 12:

Afficher la liste des « employé ».

Activité 13 :

Afficher la liste des « fonction» tout en donnant des nouveaux noms significatifs aux différentes colonnes.

Activité 14 :

Afficher la liste des noms des employés. (sans doublons)

b) Recherche de lignes à partir d'une table : « sélection »

La forme générale de cette commande est

Activité 15 :

Afficher les détails (nom, prénom, salaire) des employés ayant des salaires compris entre 800 et 2000 tout en donnant des nouveaux noms significatifs aux différentes colonnes.

Activité 16 :

Afficher la liste des employés ayant des salaires compris entre 800 et 2000, dont le code appartient à l'ensemble (E0012, E0014, E0016, E0018, E0020), dont l'adresse ***commence*** *par Rue et* ***ne sont pas nés durant*** *l'année 1970.*

2ème solution

Remarque :

** Les opéateurs relationels en SQL sont :* ***égal*** *(=),* ***différent*** *(<>, !=),* ***supérieur ou égal*** *(>=),* ***supérieur*** *(>),****inférieur ou égal*** *(<=),* ***inférieur*** *(<),* ***appartient à une liste*** *(in),* ***recherche dans une chaine de caractères*** *(like),* ***sélectionner un intervalle de données*** *(between).*

**Les opérateurs logiques en SQL sont : Et (And), ou (Or)*

Les fonctions prédéfinies sur les dates :* ***YEAR* *pour récupérer l'année d'une date,* ***MONTH*** *pour récupérer le mois d'une date,* ***DAY*** *pour récupérer le jour d'une date.*

c) Recherche de données à partir de plusieurs tables : « jointure »

La forme générale de cette commande est ;

Activité 17 : *Afficher* ***les détails des enfants*** *de l'employé 'MAHJOUB FARID'*

Activité 18 : *Afficher les intitulés des fonctions affectées à l'employé 'E0011' dans les années 2000…2008*

*Utiliser des alias (**alias**=nom synonyme abrégé à une table) pour alléger l'écriture de la commande SELECT.*

2ème solution

Activité 19 :*Afficher les codes, nom et prénoms des enfants accompagnés du code de leur père pour les gérants*

Remarque : *trois tables ont été utilisées pour répondre à ce besoin. ….. Opérations de jointures seront donc exécutées ;*
**La 1ère : entre ……. * La 2ème : entre …… La 3ème : entre …….*

Activité 20 : *Afficher les codes, les salaires des employés ayant le même salaire que l'employé ayant pour code 'E0031'.*

Remarque **:** *dans cette requête nous avons utilisé deux fois la même table. Il s'agit d'une opération* ***d'auto- jointure.***
Dans ce cas l'utilisation d'un ***alias*** *dans la commande SELECT devient indispensable.*

Activité 20': *Afficher les détails des frères de l'enfant ABID MOUNA.*

d) **Recherche de données avec Tri :** La forme générale de cette commande est ;

Dans la clause ***ORDER BY****, on peut remplacer les noms des colonnes par* ***leur rang*** *dans le paramètre Liste_nom_colonne*

Activité 21 : *Afficher le nom, le prénom et la date de naissance des employés commençant par le plus âgé.*

Activité 22 :

Afficher le nom complet des employés dans une nouvelle colonne (Nom & Prénom) *par ordre décroissant des noms.*

2ème solution

Remarque :

Dans le langage SQL la fonction **CONCAT() permet de concaténer les valeur de plusieurs colonnes pour ne former qu'une seule chaîne de caractère.*

Dans certain SGBD tel que **Microsoft Access la fonction **CONCAT()** n'est pas reconnue, elle est remplacée par l'opérateur de concaténation **+**.*

Activité 23 :

Afficher le code, le nom, le prénom et le 'salaire net' (salaire+prime) des employés triés par ordre croissant des codes.

Remarque IS **NULL / IS NOT NULL**:

*L'opérateur **IS** permet de filtrer les résultats qui contiennent la valeur **NULL**. Cet opérateur est indispensable car la valeur NULL est une valeur inconnue et ne peut par conséquent pas être filtrée par les opérateurs de comparaison*

(O.dfaff **= Null**)

e) Utilisation des fonctions de calculs dans les opérations de recherche (fonctions Agrégat) :

*Les **fonctions Agrégat** ne peuvent être utilisées qu'avec la commande SELECT et en dehors de la clause WHERE.*
*Les fonctions offertes par SQL sont les suivantes : **COUNT, SUM, MIN, MAX, AVG***

Activité 24 : *Afficher le nombre des employés de la base.*

Activité 25 : *Afficher le montant total des salaires versé aux employés à la fin du mois occupant encore une fonction.*

Activité 26 : *Afficher la date de naissance de l'employé le plus âgé.*

Activité 27 : *Afficher le nombre d'enfant d'un employé pour code père e0011*

Activité 28 : *Afficher le plus haut, le plus bas ainsi que la moyenne des salaires des employés dont la date de fin d'affectation est Null. (Càd travaillent encore jusqu'à ce jour ci). Tout en modifiant les noms des colonnes affichées.*

f) Utilisation des regroupements GROUP BY

Elle est utilisée pour arranger des données identiques dans des groupes.
Il faut introduire une clause de groupage dans la requête. Pour cela SQL fournit la clause GROUP BY

Activité 30 : *Afficher le nombre d'enfant pour chaque employé*

SELECT *expression 1, expression 2, ... expression_n, fonction_agrégat (expression_agregat)*
FROM *nom_table*
GROUP BY *expression1***;**

Dans le cas où on ne sélectionne aucune colonne

SELECT *fonction_agrégat (expression_agregat)*
FROM *nom_table;*

Remarque :

- *Le mot-clé GROUP BY s'utilise lorsque diverses colonnes d'une ou de plusieurs tables sont sélectionnées et qu'une fonction d'agrégat au moins apparaît dans l'instruction SELECT*
- *Regrouper toutes les autres colonnes sélectionnées, c'est-à-dire, toutes les colonnes doivent figurer dans la clause GROUP BY sauf celle(s) utilisée(s) par les fonctions d'agrégat.*

g) Utilisation d'une condition sur le groupage HAVING

Sert à préciser quels groupes doivent être sélectionnés.
Elle se place après la clause GROUP BY. Elle suit la même syntaxe que celui de la clause WHERE.
Cependant, il ne peut porter que sur des caractéristiques de groupe : fonction de groupe ou expression figurant dans la clause GROUP BY.

SELECT *expression 1, expression 2, ... expression_n, fonction_agrégat (expression_agregat)*
FROM *nom_table*
[WHERE *conditions***]**
GROUP BY *expression1, expression2, ... expression_n***;**
HAVING *conditions ;*

Activité 31 : *Afficher les employés ayant un nombre d'enfant supérieur ou égale à 2*

Activité 32 :*Afficher les fonctions qui ont dépensé un montant supérieur à 3500 pour la rémunération de leur employé*