

TP 3: L'ORDRE SELECT

OBJECTIFS :

Apprendre la recherche, restriction et tri des données avec la requête *SELECT*.

Partie I: *SELECT* basique

- La commande *SELECT* permet la recherche d'information dans une base de données

Syntaxe:

```
SELECT * | [DISTINCT] colonne | expression [alias], ...  
FROM table;
```

- Le caractère * après *SELECT* indique que toutes les colonnes de la table doivent être prises en compte.
- L'option *DISTINCT* permet de ne conserver que des lignes distinctes, en éliminant les doublons.
- *colonne* | *expression*: permet de préciser les noms des colonnes.
- *alias*: renomme l'expression (nom valable pendant la durée de la requête).
- La clause *FROM* : permet de spécifier la (ou les) table (s) à interroger.

Application:

1. Afficher le contenu des différentes tables:

```
SELECT * FROM Client;  
....
```

2. Afficher quelques colonnes:

```
SELECT codC, nomC  
FROM Client;
```

3. Opérateurs arithmétique (+,-,*,/):

```
SELECT codP, lib, Pu*Qtes  
FROM Produit;
```

4. Définition des alias:

```
SELECT nomC AS "Nom du client"  
FROM Client;
```

```
SELECT nomC "Nom du client"  
FROM Client;
```

```
SELECT codP, lib, Pu*Qtes AS Total  
FROM Produit;
```

```
SELECT codP, lib, Pu*Qtes Total  
FROM Produit;
```

5. Les lignes dupliqués:

```
SELECT codC FROM Facture;
```

Suppression des doublons:

```
SELECT DISTINCT codC
FROM Facture;
```

6. Opérateur de concaténation ||:

```
SELECT codC || nomC As "code et nom"
FROM Client;
```

7. Chaîne de caractère latérale:

```
SELECT nomC , ' habite A ' ,adresseC
FROM Client;
```

```
SELECT nomC || ' habite A ' || adresseC As "Localisation"
FROM Client;
```

Partie II: Restriction et tri des données

II.1. L'interrogation avec prédicats (conditions):

Limiter les lignes retournées en utilisant la clause WHERE

```
SELECT * | [DISTINCT] colonne | expression [alias], ...
FROM table;
[WHERE condition(s)];
```

- WHERE** : indique qu'une ligne doit valider une condition avant d'être retournée.
 - condition** : peut être composé de nom de colonne, expression, des valeurs constantes et des opérateurs de comparaison.
 - condition** : est composée de colonnes, d'expressions, de constantes liées deux à deux entre des opérateurs :
 - de comparaison (>, =, <, >=, <=, <>);
 - logiques (NOT, AND ou OR);
 - intégrés (BETWEEN, IN, LIKE, IS NULL).

Application:

- Afficher les factures du client numéro 1250:

```
SELECT *
FROM Facture
WHERE codC=1250;
```

- Afficher le numéro et la date des factures du client numéro 1250:

```
SELECT numF, datF
FROM Facture
WHERE codC=1250;
```

3. Afficher les clients qui habitent à 'Tunis':

```
SELECT *
FROM Client
WHERE AdresseC='Tunis';
```

NB: la comparaison des chaînes de caractères est sensible à la casse ('Tunis' est différente de 'TUNIS':

4. Afficher les factures faites le '16/07/99'.

```
SELECT *
FROM Facture
WHERE DatF='16/07/99';
```

II.2. Between, like, in et is null:

1. **Between** : extraire des lignes en se basant sur un intervalle de valeurs.

Afficher les clients qui ont un crédit entre 50 et 250'.

```
SELECT *
FROM Client
WHERE creditC >=50 AND creditC<=200;
```

Ou bien:

```
SELECT *
FROM Client
WHERE creditC BETWEEN 50 AND 200;
```

2. **In**: extraire des lignes en se basant sur une liste de valeurs.

Afficher les produits dont la quantité en stock est égale à 20, 40 ou bien 50.

```
SELECT *
FROM Produit
WHERE QteS=20 OR QteS=40 OR QteS=50;
```

Ou bien:

```
SELECT *
FROM Produit
WHERE QteS IN (20,40,50);
```

3. **LIKE**: rechercher les valeurs qui vérifie une condition textuelle. La condition de recherche peut contenir des caractères ou des entiers :

% indique zéro ou plusieurs caractères

_ indique un seul caractère

Afficher les clients dont la deuxième lettre des noms est 'a'.

```
SELECT *
FROM Client
WHERE nomC LIKE ('_a%');
```

4. **IS [NOT] NULL**: rechercher les valeurs dont certains champs contiennent des valeurs nulles.

Afficher les clients qui n'ont pas une adresse.

```
SELECT *
FROM Client
WHERE AdresseC IS NULL;
```

II.3. tri des données: Order by

Trier les lignes retournées avec ORDER BY

Syntaxe:

```
SELECT expr FROM table
[WHERE condition(s)]
[ORDER BY {column, expr} [ASC | DESC]];
```

On peut spécifier une expression, un alias, un nom ou une position de colonne dans la clause ORDER BY

Par défaut l'ordre de tri est ascendant.

Pour renverser l'ordre de tri spécifier le mot clé **DESC** après le nom de colonne dans la clause ORDER BY

Exemple :

Rechercher la liste des produits en triant le résultat avec le plus chère :

```
Select *
from Produit
Order by PU DESC;
```

Plus Grand -> plus Petit

Rechercher le prix total des produits en triant le résultat par le montant le plus faible

```
Select codP, Lib, PU*QteS AS "total"
from Produit
Order by total;
```

Plus Petit -> plus Grand

Application :

Formuler les requêtes suivantes en SQL :

1. Afficher toutes les informations sur toutes les commandes,
2. Afficher toutes les informations sur les clients de 'Tunis' et dont le code est un multiple de 1000,
3. Quel est le code du client qui a lancé la commande numéro 1,
4. Quels sont les produits dont la quantité en stock est comprise entre 500 et 700 et possédant un seuil?
5. Quels sont les clients dont le nom se termine par "ed" ?
6. Quels sont les Produits dont la troisième lettre du libellé est "p" ?
7. Quels sont les factures faites en 1999 ?
8. Donner les codes des clients qui ont commandés des commandes ?
9. Quels sont les clients qui ont lancé une commande après le 20/03/1998 ?
10. Trier les produits sur leurs libellés et ce dans l'ordre croissant.
11. Afficher tous les produits avec un ordre croissant pour les prix unitaires et décroissant pour la quantité en stock.
12. Afficher les facture suivant la manière suivante:

La Facture "NumF" du client "codC", total = "MontF"

Partie III: Exercice

1. Créer un nouveau utilisateur **Cabinet/Medecin**.
2. On se propose l'exemple suivant, relatif à la gestion des consultations des patients pour un médecin:

Patient

	Nom du champ	Type de données	
🔑	Mat_Pat	Texte	Matricule du patient
	Nom	Texte	Nom du Patient
	Prenom	Texte	Prénom du Patient
	Date_Naiss	Date/Heure	Date de Naissance du patient
	Groupe_Sanguin	Texte	Groupe Sanguin du patient
	Poids	Numérique	Poids du patient
	Taille	Numérique	Taille du patient
	Sexe	Texte	Sexe du patient

Maladie

	Nom du champ	Type de données	
🔑	Code_Mal	Texte	Code Maladie
	Designation	Texte	Désignation de la Maladie

Consultation

	Nom du champ	Type de données	
🔑	N°	NuméroAuto	
	Mat_Pat	Texte	Matricule du patient
	Date_Cons	Date/Heure	Date consultation
	Heure_Cons	Date/Heure	heure consultation
	Type_Cons	Texte	Type consultation

Suivi

	Nom du champ	Type de données	
🔑	Mat_Pat	Texte	Matricule du Patient
🔑	Code Mal	Texte	Code Maladie

Créer les tables **Patient**, **Maladies**, **Consultation** et **Suivi** sans les contraintes de clé primaire et étrangère (la taille des variables chaines de caractères est à préciser).

3. Ajouter à la table Patient le champ **adresse** chaine de caractères.
4. Modifier le champ Type_Cons de telle sorte qu'il sera de type varchar(30).
5. Ajouter les contraintes suivantes :
 - Ajouter aux tables les contraintes de clé primaire et clé étrangère.
 - Le groupe sanguin doit être dans l'ensemble {'O-', 'O+', 'A-', 'A+', 'B-', 'B+', 'AB-', 'AB+'}
 - Le nom et le prénom doit être non vide.
 - Le poids doit être compris entre 3 et 200.
 - Le sexe doit être égale à 'M' ou bien 'F'.
 - Le Type de la consultation doit être égale à 'Normale', 'Contrôle' ou bien 'Domicile'
6. Saisie des données:

a. Table Patient

Mat_Pat	Nom	Prénom	Date_Naiss	Groupe_Sanguin	Poids	Taille	Sexe
120001	ABDI	Sami	01/07/1960	A+	60	1,6	M
120002	BEN KHALED	Ahlem	24/04/1980	B+	50	1,55	F
120003	BEN SLIMANE	Abdelkerim	16/05/1980	O-	68	1,7	M
120004	GHARBI	Isam	03/06/1975	O+	88	1,75	M
120005	MELKI	Raoudha	15/12/1968	A+	54	1,45	F
120006	SELMi	Ali	13/11/1952	A-	60	1,55	M

b. Table **Maladie**

Code_Mal	Designation
150001	Fièvre
150002	Grippe
150003	Asthme
150004	Diabète

c. Table **Consultation**

num	Mat_Pat	Date_Cons	Heure_Cons	Type_Cons
1	120003	12/04/2004	08 :30	Normale
2	120003	12/05/2004	15 :00	Contrôle
3	120002	12/04/2004	20 :20	Domicile
4	120001	20/04/2004	15 :30	Normale

d. Table **Suivi**

Mat_Pat	Code_Mal
120003	150002
120002	150003
120005	150003
120001	150004

7. Création des Requêtes:

- Afficher les noms des patients.
- Afficher la liste des patients qui ont comme groupe sanguin A+.
- Afficher les femmes qui ont un groupe sanguin O+ ou bien O- et dont le poids ne dépasse pas 70Kg.
- Afficher les patients ayant une taille supérieure à 1,6 et un poids supérieur à 62.
- Afficher les noms des patients nés entre 1975 et 1985.
- Afficher la liste des patients dont leurs noms contiennent "S".
- Afficher la liste des patients dont le 2ème caractère de leurs noms est "E".
- Afficher la liste des patients dont leurs prénoms ne commencent ni par "S" ni par "A".
- Ordonner les patients en fonction de leurs poids.
- Afficher les patients qui ont un corps parfait c à d le poids est égale à la partie décimale de son longueur (Exp: 60kg | 1.6m ou bien 75kg | 1.75m)
- Afficher la liste des consultations faites à domicile après midi.

Avec jointure:

- Afficher le nom, prénom, Sexe et la Désignation de la maladie des patients.
- Afficher le nom, prénom des patients et leurs consultations enlevez les patients qui ne sont pas nés en 1980.
- Affichez les consultations avec les noms des patients.
- Afficher le nombre des patients par groupe sanguin.
- Affiche le poids moyen par sexe
- Afficher le nombre de patients asthmatiques ayant subi une consultation à domicile.
- Afficher le nombre de consultations normales pour chaque patient.
- Afficher le nom et Prénom des diabétiques triés par ordre alphabétique croissant suivant leurs noms.