**Révision des Concepts SQL : De la Sélection aux Sous-Requêtes**

**1. Requête de base : SELECT ... FROM**

* **Syntaxe minimale :**

SELECT column1, column2

FROM table\_name;

- Cette requête permet de récupérer les colonnes column1 et column2 de la table table\_name.

**2. Ajout de la clause WHERE**

* **Filtrer les résultats :**

SELECT column1, column2

FROM table\_name

WHERE condition;

-La clause WHERE est utilisée pour filtrer les lignes selon une condition.

**3. Tri avec ORDER BY**

* Ordonner les résultats :

SELECT column1, column2

FROM table\_name

WHERE condition

ORDER BY column1 ASC|DESC;ORDER BY column1 ASC|DESC;

-ORDER BY permet de trier les résultats par une colonne spécifique dans un ordre croissant (ASC) ou décroissant (DESC).

**4. Fonctions de groupe et GROUP BY**

* Utiliser les fonctions d’agrégation (comme **COUNT, SUM, AVG, MIN, MAX**) pour regrouper des données :

SELECT column1, COUNT(\*)

FROM table\_name

WHERE condition

GROUP BY column1;

-Les fonctions de groupe sont utilisées pour obtenir des statistiques sur les groupes définis par GROUP BY.

**5. Ajout de la clause HAVING**

* **Filtrer après un GROUP BY :**

SELECT column1, COUNT(\*)

FROM table\_name

GROUP BY column1

HAVING COUNT(\*) > 1;

-La clause HAVING permet de filtrer les résultats après un regroupement, contrairement à WHERE qui s’applique avant.

**6. Les jointures (INNER, LEFT, RIGHT, FULL OUTER)**

* **Jointure interne (INNER JOIN) :**

SELECT t1.column1, t2.column2

FROM table1 t1

INNER JOIN table2 t2 ON t1.id = t2.id;

-Récupère uniquement les lignes avec une correspondance dans les deux tables.

* **Jointure externe gauche (LEFT JOIN) :**

SELECT t1.column1, t2.column2

FROM table1 t1

LEFT JOIN table2 t2 ON t1.id = t2.id;

-Récupère toutes les lignes de table1 et les correspondances de table2 (si elles existent).

* **Jointure externe droite (RIGHT JOIN) :**

SELECT t1.column1, t2.column2

FROM table1 t1

RIGHT JOIN table2 t2 ON t1.id = t2.id;

* Récupère toutes les lignes de table2 et les correspondances de table1 (si elles existent).
* **Jointure externe complète (FULL OUTER JOIN) :**

SELECT t1.column1, t2.column2

FROM table1 t1

FULL OUTER JOIN table2 t2 ON t1.id = t2.id;

-Récupère toutes les lignes des deux tables, même sans correspondance.

**7. Les opérateurs UNION/INTERSECT**

* **UNION :**

-Combine deux requêtes et renvoie des résultats uniques (sans doublons) :

SELECT column1 FROM table1

UNION

SELECT column1 FROM table2;

* **UNION ALL :**

-Comme UNION, mais conserve les doublons :

SELECT column1 FROM table1

UNION ALL

SELECT column1 FROM table2;

* **INTERSECT :**

-Renvoie les lignes communes aux deux requêtes :

SELECT column1 FROM table1

INTERSECT

SELECT column1 FROM table2;

* **MINUS (ou EXCEPT dans certains SGBD) :**

-Renvoie les lignes de la première requête qui ne sont pas présentes dans la seconde :

SELECT column1 FROM table1

MINUS

SELECT column1 FROM table2;

**8. Sous-requêtes (Subqueries)**

* **Sous-requête dans une clause WHERE :**

SELECT column1

FROM table1

WHERE column2 = (SELECT MAX(column2) FROM table2);

-Les sous-requêtes permettent de calculer une valeur ou une condition à l'intérieur d'une requête principale.

* **Sous-requête dans une clause FROM :**

SELECT t1.column1, t2.avg\_value

FROM table1 t1,

(SELECT AVG(column2) AS avg\_value FROM table2) t2;

-Utilisée pour générer des tables temporaires à partir d'une requête.

**Exercice d’application :**

En utilisant la base de données **TestJoinsDB** que nous avons créée :

1. **Concept : SELECT et FROM**
2. **Sélectionnez le nom et l'âge de tous les étudiants.**
3. **Sélectionnez toutes les colonnes de la table employees.**
4. **Sélectionnez l'identifiant et le département de chaque étudiant.**
5. **Sélectionnez le nom et le salaire de chaque employé.**
6. **Sélectionnez uniquement les prénoms des étudiants.**
7. **Concept : Clause WHERE**
8. **Sélectionnez les étudiants dont l'âge est supérieur à 20.**
9. **Sélectionnez les employés qui travaillent dans le département 2.**
10. **Sélectionnez les étudiants dont le nom commence par 'A'.**
11. **Sélectionnez les employés ayant un salaire supérieur à 3000.**
12. **Sélectionnez les étudiants du département 'Informatique'.**
13. **Concept : ORDER BY**
14. **Sélectionnez les étudiants et ordonnez-les par âge croissant.**
15. **Sélectionnez les employés et ordonnez-les par salaire décroissant.**
16. **Sélectionnez les étudiants et ordonnez-les par leur nom en ordre alphabétique.**
17. **Sélectionnez les employés et ordonnez-les par leur département.**
18. **Sélectionnez les salaires des employés et ordonnez-les par salaire puis par nom.**
19. **Concept : Fonctions de groupe et GROUP BY**
20. **Comptez le nombre d'étudiants par département.**
21. **Calculez le salaire moyen par département pour les employés.**
22. **Affichez le nombre total d'employés par département.**
23. **Affichez la somme des salaires pour chaque département.**
24. **Comptez le nombre d'employés dans chaque grade de salaire.**
25. **Concept : Clause HAVING**
26. **Comptez le nombre d'étudiants par département où il y a plus de 5 étudiants.**
27. **Affichez les départements où le salaire moyen est supérieur à 4000.**
28. **Comptez le nombre d'employés par département, mais seulement pour les départements avec plus de 10 employés.**
29. **Affichez les grades de salaires où le total des salaires dépasse 20 000.**
30. **Affichez les départements ayant plus de 3 employés avec un salaire moyen supérieur à 5000.**
31. **Concept : Les Jointures (INNER, LEFT, RIGHT, FULL OUTER)**
32. **Récupérez le nom des employés et le nom de leur département (INNER JOIN).**
33. **Récupérez tous les employés et leurs départements, même ceux sans département (LEFT JOIN).**
34. **Récupérez tous les départements et leurs employés, même ceux sans employés (RIGHT JOIN).**
35. **Récupérez tous les employés et tous les départements, même ceux sans correspondance (FULL OUTER JOIN).**
36. **Récupérez le nom des employés et leur grade de salaire.**
37. **Concept : UNION, UNION ALL, INTERSECT, et MINUS**
38. **Sélectionnez tous les noms des étudiants et des employés (sans doublons).**
39. **Sélectionnez tous les noms des étudiants et des employés (avec doublons).**
40. **Sélectionnez les noms présents à la fois dans les étudiants et les employés.**
41. **Sélectionnez les noms des étudiants qui ne sont pas dans la liste des employés.**
42. **Sélectionnez les noms des employés qui ne sont pas étudiants.**
43. **Concept : Sous-requêtes (Subqueries)**
44. **Sélectionnez les étudiants dont l'âge est supérieur à l'âge moyen de tous les étudiants.**
45. **Sélectionnez les employés ayant un salaire supérieur à la moyenne de leur département.**
46. **Sélectionnez les employés travaillant dans des départements qui ont plus de 5 employés.**
47. **Sélectionnez les employés dont le salaire est supérieur au salaire maximum des étudiants.**
48. **Sélectionnez les départements qui n'ont pas d'employés.**