

## ✓ 1. REQUÊTES AVEC SELECT WHERE / UPDATE (Avec Exemple)

-- Table employe

```
CREATE TABLE employe (  
    id_employe INT PRIMARY KEY,  
    nom VARCHAR(50),  
    age INT,  
    salaire DECIMAL(10,2)  
    id_dept INT,  
    FOREIGN KEY (id_dept) REFERENCES departement (id)  
);
```

-- Table departement

```
CREATE TABLE departement (  
    id INT PRIMARY KEY,  
    nom_dept VARCHAR(50),  
);
```

-- Insertion de plusieurs employés en une seule instruction

```
INSERT INTO employee (id_employe, nom, age, salaire, id_dept)  
VALUES  
    (1, 'Ali Ben Ali', 30, 1500.00 ,2),  
    (2, 'Sana Trabelsi', 28, 1800.50 ,1),  
    (3, 'Mohamed Salah', 40, 2200.75 , 1);
```

-- Augmentation de 150 pour tous les employés

```
UPDATE employee  
SET salaire = salaire + 150;
```

-- Augmentation de 150 uniquement pour les employés de plus de 29 ans

UPDATE employe

SET salaire = salaire + 150

WHERE age > 29;

-- Sélection de toutes les colonnes de la table employe

SELECT \* FROM employe;

-- Sélection des employés dont l'âge est > à 28 ET le nom contient 'a'

SELECT \* FROM employe

WHERE age > 28 AND nom LIKE '%a%';

-- Sélection des employés soit âgés de plus de 30 OU dont le salaire est inférieur à 2200

SELECT \* FROM employe

WHERE age > 30 OR salaire < 2200;

-- Sélection des employés dont le salaire n'est pas défini (NULL)

SELECT \* FROM employe

WHERE salaire IS NULL;

-- Sélection des employés dont le nom est défini (non NULL)

SELECT \* FROM employe

WHERE nom IS NOT NULL;

## ✓ 2. REQUÊTES AVEC Jointure Agrégats

-- Sélection du nom de l'employé et du nom du département associé

```
SELECT e.nom AS NomEmploye, d.nom_dept AS Nom Département
FROM employe e, departement d
WHERE e.id_dept = d.id;
```

-- Compte le nombre total d'employés dans la table

```
SELECT COUNT(*) AS NombreEmployes
FROM employe;
```

-- Calcule la somme totale des salaires de tous les employés

```
SELECT SUM(salaire) AS SommeSalaires
FROM employe;
```

-- Calcule le salaire moyen des employés

```
SELECT AVG(salaire) AS SalaireMoyen
FROM employe;
```

-- Trouve le salaire le plus élevé parmi les employés

```
SELECT MAX(salaire) AS SalaireMax
FROM employe;
```

-- Trouve le salaire le plus bas parmi les employés

```
SELECT MIN(salaire) AS SalaireMin
FROM employe;
```