

# GUIDE DE RÉVISION SQL - SÉLECTION ET TRI DES LIGNES

## TABLE DES MATIÈRES

1. Clause WHERE
  2. Opérateurs de comparaison
  3. Opérateurs logiques
  4. Tri avec ORDER BY
- 

## 1 CLAUSE WHERE

### ◇ Syntaxe de Base

```
SELECT [colonnes]  
FROM table  
WHERE condition;
```

### ◇ Règles Importantes

- Se place après la clause FROM
- Permet de filtrer les lignes selon des conditions
- Sensible à la casse pour les chaînes de caractères
- Les chaînes et dates doivent être entre quotes simples ('...')

### Exemple pratique:

```
SELECT ename, job, deptno  
FROM emp  
WHERE job = 'CLERK';
```

---

## 2 OPÉRATEURS DE COMPARAISON

### ◇ Opérateurs de Base

Opérateur	Signification	Exemple
=	Égal à	WHERE sal = 3000
>	Supérieur à	WHERE sal > 2000
>=	Supérieur ou égal	WHERE sal >= 2000
<	Inférieur à	WHERE sal < 1000
<=	Inférieur ou égal	WHERE sal <= 1000
<>	Différent de	WHERE job <> 'MANAGER'

## ♦ Opérateurs Spéciaux

### 1. BETWEEN ... AND ...

```
SELECT ename, sal
FROM emp
WHERE sal BETWEEN 1000 AND 1500;
```

### 2. IN (liste)

```
SELECT ename
FROM emp
WHERE job IN ('CLERK', 'MANAGER');
```

### 3. LIKE (avec caractères génériques)

- % : zéro ou plusieurs caractères
- \_ : un seul caractère

```
SELECT ename
FROM emp
WHERE ename LIKE 'S%';    -- Commence par S
WHERE ename LIKE '_A%';   -- Deuxième lettre = A
```

### 4. IS NULL

```
SELECT ename, comm
FROM emp
WHERE comm IS NULL;
```

---

## 3 OPÉRATEURS LOGIQUES

### ♦ Types d'Opérateurs

Opérateur	Description
AND	Les deux conditions doivent être vraies
OR	Au moins une condition doit être vraie
NOT	Inverse la condition

### ♦ Exemples Pratiques

```
-- AND
SELECT ename, job, sal
FROM emp
WHERE sal >= 1100 AND job = 'CLERK';

-- OR
SELECT ename, job, sal
FROM emp
WHERE job = 'MANAGER' OR job = 'PRESIDENT';

-- NOT
SELECT ename, job
FROM emp
WHERE job NOT IN ('CLERK', 'MANAGER');
```

### ♦ Priorité des Opérateurs

1. Opérateurs de comparaison (=, >, <, etc.)
2. NOT
3. AND
4. OR

### Utilisation des parenthèses:

```
SELECT ename, job, sal
FROM emp
WHERE (job = 'SALESMAN' OR job = 'PRESIDENT')
    AND sal > 1500;
```

---

## 4 TRI AVEC ORDER BY

### ◇ Syntaxe de Base

```
SELECT [colonnes]
FROM table
[WHERE condition]
ORDER BY colonne [ASC|DESC];
```

### ◇ Options de Tri

- **ASC**: Tri ascendant (par défaut)
- **DESC**: Tri descendant

### ◇ Exemples de Tri

```
-- Tri simple
SELECT ename, sal
FROM emp
ORDER BY sal DESC;

-- Tri multiple
SELECT ename, deptno, sal
FROM emp
ORDER BY deptno ASC, sal DESC;

-- Tri sur alias
SELECT ename, sal*12 AS salaire_annuel
FROM emp
ORDER BY salaire_annuel;
```

---

## POINTS CLÉS À RETENIR

1. La clause WHERE filtre les lignes **AVANT** le tri
  2. Les chaînes de caractères sont sensibles à la casse
  3. Utilisez les parenthèses pour contrôler la priorité des opérateurs
  4. ORDER BY est toujours la dernière clause
  5. On peut trier sur des colonnes non sélectionnées
-



# EXERCICES PRATIQUES

## 1. Filtrage simple:

```
-- Trouver les employés du département 10 avec un salaire > 2000
SELECT ename, sal, deptno
FROM emp
WHERE deptno = 10 AND sal > 2000;
```

## 2. Recherche avec LIKE:

```
-- Trouver les employés dont le nom commence par 'S'
SELECT ename
FROM emp
WHERE ename LIKE 'S%'
ORDER BY ename;
```

## 3. Combinaison de conditions:

```
-- Trouver les managers et clerks avec un salaire entre 1000 et 3000
SELECT ename, job, sal
FROM emp
WHERE job IN ('MANAGER', 'CLERK')
      AND sal BETWEEN 1000 AND 3000
ORDER BY sal DESC;
```