

## **TP3 – Requêtes SQL Objectif**

Créer une base de données et importer les données de gestion de stocks.

### **Liste des Requêtes SQL**

1. **Sélectionner** tous les produits

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0010 seconde(s).)

```
SELECT * FROM products;
```

2. **Afficher** le nom et le prix de tous les produits

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0011 seconde(s).)

```
SELECT name, unit_price FROM products;
```

3. **Lister** toutes les catégories

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0013 seconde(s).)

```
SELECT * FROM categories;
```

4. **Trouver** les produits dont le prix est **supérieur à 20**

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0075 seconde(s).)

```
SELECT * FROM products WHERE unit_price > 20;
```

5. **Afficher** les mouvements de stock de type 'IN' (entrée)

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0017 seconde(s).)

```
SELECT * FROM stock_movements WHERE movement_type = 'IN';
```

6. **Sélectionner** les produits appartenant à la catégorie 'Informatique' (ID 1)

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0013 seconde(s).)

```
SELECT * FROM products WHERE category_id = 1;
```

7. **Lister** les mouvements de stock effectués **après le 2025-02-01**

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0073 seconde(s).)

```
SELECT * FROM stock_movements WHERE movement_date > '2025-02-01';
```

8. **Trier** les produits par prix, par ordre **croissant**

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0010 seconde(s.) [unit\_price: 5.50... - 79.90...])

```
SELECT * FROM products ORDER BY unit_price ASC;
```

9. **Rechercher** les produits dont le nom **contient la lettre 's'**

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0011 seconde(s).)

```
SELECT * FROM products WHERE name LIKE '%s%';
```

10. **Compter** le nombre total de produits

✓ La requête SQL a été exécutée avec succès. (traitement en 0,0024 seconde(s).)

```
SELECT COUNT(*) FROM products;
```

11. **Calculer** le total des mouvements de type 'IN' pour **chaque produit**

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0069 seconde(s).)

```
SELECT product_id, SUM(quantity) AS TotalEntrees FROM stock_movements WHERE movement_type = 'IN' GROUP BY product_id;
```

12. **Déterminer** le **prix moyen** des produits pour **chaque catégorie**.

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0048 seconde(s).)

```
SELECT c.name AS category_name, AVG(p.unit_price) AS AveragePrice FROM products p JOIN categories c ON p.category_id = c.id GROUP BY c.name;
```