

Cours SQL – Requêtes, CRUD, SELECT, LIKE, etc.

♦ INTRODUCTION À SQL

SQL (Structured Query Language) est un langage utilisé pour interagir avec une base de données relationnelle. Il permet :

- d'ajouter des données,
- de les lire,
- de les modifier,
- et de les supprimer.

C'est ce qu'on appelle le CRUD.

♦ CRUD en SQL : Les 4 grandes opérations

✓ 1. CREATE – INSERT INTO

```
INSERT INTO table (colonne1, colonne2)
VALUES ('valeur1', 'valeur2');
```

Exemple :

```
INSERT INTO utilisateurs (nom, age)
VALUES ('Nathan', 30);
```

🔍 2. READ – SELECT (la plus utilisée)

```
SELECT colonne1, colonne2
FROM table
WHERE condition;
```

Exemple simple :

```
SELECT * FROM utilisateurs;
```

Avec condition :

```
SELECT nom FROM utilisateurs WHERE age > 25;
```

Avec LIKE :

```
SELECT * FROM utilisateurs WHERE nom LIKE 'N%'; -- Commence par N
SELECT * FROM utilisateurs WHERE nom LIKE '%n'; -- Finit par n
SELECT * FROM utilisateurs WHERE nom LIKE '%ath%'; -- Contient "ath"
```

Autres filtres utiles :

-- BETWEEN

```
SELECT * FROM produits WHERE prix BETWEEN 10 AND 20;
```

-- IN

```
SELECT * FROM utilisateurs WHERE nom IN ('Nathan', 'Jeremy');
```

-- IS NULL / IS NOT NULL

```
SELECT * FROM utilisateurs WHERE email IS NULL;
```

-- ORDER BY

```
SELECT * FROM utilisateurs ORDER BY age DESC;
```

-- LIMIT (MySQL)

```
SELECT * FROM utilisateurs ORDER BY id LIMIT 5;
```

3. UPDATE – Modifier une donnée

UPDATE table

SET colonne1 = 'valeur'

WHERE condition;

Exemple :

```
UPDATE utilisateurs SET age = 31 WHERE nom = 'Nathan';
```

4. DELETE – Supprimer des données

DELETE FROM table

WHERE condition;

Exemple :

```
DELETE FROM utilisateurs WHERE age < 18;
```

♦ FONCTIONS UTILES EN SELECT

✓ Fonctions d'agrégation :

COUNT(*) - Compte le nombre de lignes
SUM(colonne) - Fait la somme
AVG(colonne) - Moyenne
MAX(colonne) - Valeur max
MIN(colonne) - Valeur min

Exemples :

```
SELECT COUNT(*) FROM utilisateurs;  
SELECT AVG(age) FROM utilisateurs;
```

✓ GROUP BY et HAVING

```
SELECT ville, COUNT(*) AS nb_utilisateurs  
FROM utilisateurs  
GROUP BY ville  
HAVING COUNT(*) > 5;
```

♦ JOINTURES : Combiner des tables

✓ INNER JOIN (jointure classique)

```
SELECT u.nom, c.nom  
FROM utilisateurs u  
INNER JOIN commandes c ON u.id = c.utilisateur_id;
```

✓ LEFT JOIN

```
SELECT u.nom, c.nom  
FROM utilisateurs u  
LEFT JOIN commandes c ON u.id = c.utilisateur_id;
```

♦ CREATE TABLE (bonus)

```
CREATE TABLE utilisateurs (
```

```
id INT AUTO_INCREMENT PRIMARY KEY,  
nom VARCHAR(100),  
age INT,  
email VARCHAR(255)  
);
```

◆ PRATIQUE : Exemple de base de données

Table utilisateurs :

id | nom | age | email

1 | Jeremy | 28 | jeremy@email.fr

2 | Nathan | 30 | nathan@email.fr

Requête pour sélectionner Jeremy :

```
SELECT * FROM utilisateurs WHERE nom = 'Jeremy';
```

◆ À retenir pour SELECT :

SELECT * FROM table	- Tout sélectionner
SELECT colonne1, colonne2	- Colonnes spécifiques
WHERE	- Filtrer les résultats
LIKE	- Recherche avec motif
BETWEEN	- Intervalle
ORDER BY	- Tri
LIMIT	- Nombre max
GROUP BY	- Regroupement
HAVING	- Filtrage après regroupement
JOIN	- Combiner plusieurs tables