

## Mettre à jour des données en SQL (`UPDATE`)

Il peut arriver que certaines données dans ta base doivent être corrigées ou mises à jour. Par exemple : un élève a changé de classe, un produit a un nouveau prix, ou une réservation doit être confirmée. C'est dans ce genre de situation que la commande **UPDATE** est utile. Elle permet de **modifier des lignes déjà existantes** dans une table.

6TQ

4TTR

 niveau

## Objectifs du cours

À la fin de cette leçon, tu seras capable de :

- Comprendre à quoi sert la commande **UPDATE**
- Modifier les données existantes dans une base MySQL
- Utiliser la clause **WHERE** pour cibler précisément les lignes à modifier
- Éviter les erreurs classiques (notamment les mises à jour en masse accidentelles)
- Maîtriser la syntaxe correcte avec des exemples concrets

## Syntaxe de base

```
1 UPDATE nom_table
2 SET colonne1 = nouvelle_valeur1,
3     colonne2 = nouvelle_valeur2,
4     ...
5 WHERE condition;
```

## Exemple simple

Imaginons la table `Elevés` :

```
1 CREATE TABLE Elevés (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nom VARCHAR(50),  
4     age INT,  
5     classe VARCHAR(10)  
6 );
```

On veut changer la classe de Jean Dupont (qui a l'ID 1) en "5A" :

```
1 UPDATE Elevés  
2 SET classe = '5A'  
3 WHERE id = 1;
```

✅ Seule la ligne où `id = 1` sera modifiée.

## La clause `where`

Voici un paragraphe explicatif sur la clause `WHERE` :

La clause `WHERE` dans les requêtes `UPDATE` sert à **définir précisément quelles lignes doivent être modifiées**. C'est un peu comme un filtre : seules les lignes qui respectent la condition seront mises à jour. Par exemple, si tu écris `WHERE id = 3`, tu ne modifies qu'un seul enregistrement. Mais tu peux aussi cibler **plusieurs lignes** avec des conditions comme `WHERE classe = '4B'` (tous les élèves de la 4B) ou `WHERE age > 15`. Tu peux même combiner plusieurs conditions avec `AND`, `OR` ou `IN`.

Bien utiliser `WHERE`, c'est **éviter de modifier trop de données par erreur**.

## Mettre à jour plusieurs colonnes en même temps

Tu peux modifier plusieurs champs d'une même ligne :

```
1 UPDATE Elevés  
2 SET age = 16,
```

```
3     classe = '5B'  
4 WHERE nom = 'Jean Dupont';
```

## ⚠ Attention : clause **WHERE** fortement recommandée !

Si tu oublies la clause **WHERE**, toutes les lignes de la table seront modifiées.

```
1 UPDATE Eleves  
2 SET classe = '5A';
```

🚨 Ce code mettra *tous les élèves* en classe "5A". Pas bon.

⚠ Attention : si tu oublies d'indiquer **QUELLE LIGNE MODIFIER**, tu risques de tout changer d'un coup. Ce cours va t'apprendre à éviter ce genre de piège.

## Astuce : tester d'abord avec **SELECT**

Avant d'exécuter un **UPDATE**, tu peux tester la condition avec un **SELECT** :

```
1 SELECT * FROM Eleves WHERE classe = '4B';
```

Si le résultat est correct, tu peux ensuite faire :

```
1 UPDATE Eleves SET classe = '5A' WHERE classe = '4B';
```

## Résumé : 5 notions essentielles

1. `UPDATE` permet de **modifier les données existantes** dans une table.
2. La clause `SET` indique **quelles colonnes** doivent être changées et avec **quelles valeurs**.
3. La clause `WHERE` permet de cibler **les lignes concernées**. Elle est **presque toujours indispensable**.
4. Sans `WHERE`, **toutes** les lignes seront modifiées. À éviter sauf cas très particulier.
5. Tu peux tester ta condition avec `SELECT` avant d'exécuter la mise à jour.

---

## Bonnes pratiques

---

- Toujours tester avec un `SELECT` avant d'exécuter un `UPDATE`
- Utiliser une **clé primaire** (souvent `id`) dans la clause `WHERE` pour éviter d'impacter plusieurs lignes par erreur
- Mettre des guillemets simples autour des **chaînes de caractères**
- Sauvegarder la base ou travailler sur une **copie de test** avant de modifier en production

---

## À toi de jouer

---

Voici une table `Livres` :

```
1 CREATE TABLE Livres (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     titre VARCHAR(100),  
4     auteur VARCHAR(50),  
5     annee INT  
6 );
```

1. Corrige l'année du livre dont le titre est *Le Petit Prince* : elle doit être 1943.
2. Change l'auteur de tous les livres qui ont "Anonyme" comme auteur, pour qu'il soit "Auteur inconnu".

Tu peux faire un `SELECT` avant pour tester les conditions.

---