

Supprimer des données en SQL (`DELETE`)

Il arrive qu'on doive supprimer des données d'une base : un ancien élève, un produit retiré du stock, ou une réservation annulée. C'est à cela que sert la commande SQL `DELETE`. Elle permet de **retirer des enregistrements existants** dans une table.

Objectifs du cours

À la fin de ce cours, tu seras capable de :

- Comprendre à quoi sert la commande `DELETE`
- Supprimer une ou plusieurs lignes dans une table MySQL
- Utiliser la clause `WHERE` pour cibler précisément les lignes à effacer
- Éviter les erreurs critiques (comme la suppression de toutes les données par accident)
- Appliquer les bonnes pratiques avant de faire un `DELETE`

Introduction

⚠ Attention : une suppression est **définitive**. Une fois les lignes effacées, elles ne peuvent plus être récupérées, sauf si tu as une sauvegarde.

Dans ce cours, tu vas apprendre à utiliser `DELETE` de manière précise et sécurisée.

Syntaxe de base

```
1 | DELETE FROM nom_table
2 | WHERE condition;
```

Exemple simple

Voici une table `Eleves` :

```
1 | CREATE TABLE Eleves (
2 |     id INT AUTO_INCREMENT PRIMARY KEY,
3 |     nom VARCHAR(50),
4 |     age INT,
```

```
5 | classe VARCHAR(10)
6 | );
```

Tu veux supprimer l'élève qui a l'ID 4 :

```
1 | DELETE FROM Eleves
2 | WHERE id = 4;
```

✓ Seule la ligne où `id = 4` sera supprimée.

Supprimer plusieurs lignes

Tu peux aussi supprimer plusieurs enregistrements avec une condition plus large :

```
1 | DELETE FROM Eleves
2 | WHERE classe = '6A';
```

Ici, **tous les élèves de la classe 6A** seront supprimés.

⚠ Attention : sans **WHERE**, tout est supprimé

```
1 | DELETE FROM Eleves;
```

Cette commande supprime **toutes les lignes** de la table `Eleves`.

✗ C'est souvent une erreur. À **ne jamais faire par accident**. Toujours vérifier si `WHERE` est bien présent.

Zoom sur la clause **WHERE**

La clause `WHERE` est indispensable pour **contrôler quelles lignes seront supprimées**. Tu peux cibler une seule ligne (`id = 3`), ou plusieurs (`classe = '5B'`). Tu peux aussi utiliser des conditions plus complexes :

- `age > 17`
- `nom LIKE 'A%'`
- `id IN (1, 2, 3)`

Cela te permet de supprimer exactement ce que tu veux, **ni plus, ni moins**. C'est une protection essentielle : sans `WHERE`, tu perds tout.

Résumé : 5 choses à retenir

1. **DELETE** permet de **supprimer** une ou plusieurs lignes d'une table.
 2. Utilise toujours **WHERE** pour **cibler** les données à effacer.
 3. Sans **WHERE**, toutes les lignes de la table seront **irréremdiablement perdues**.
 4. Tu peux combiner plusieurs conditions avec **AND, OR, IN**, etc.
 5. Fais un **SELECT** avec ta condition avant d'exécuter le **DELETE**, pour vérifier ce que tu vas supprimer.
-

Bonnes pratiques

- Toujours **tester la condition WHERE** avec un **SELECT** avant de supprimer
 - Ne jamais utiliser **DELETE** sans **WHERE**, sauf si tu veux vraiment tout vider
 - Si tu travailles sur une base importante, commence par un **backup** (sauvegarde)
 - Sur des données sensibles ou en production, demande une **confirmation humaine** avant exécution
-

À toi de jouer

Voici la 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. Supprime le livre dont l'auteur est "Anonyme".
 2. Supprime tous les livres publiés avant l'année 1950.
 3. Tente de faire un **DELETE** sans **WHERE** : que se passe-t-il ? Pourquoi faut-il l'éviter ?
-