

# Fiche procédure SQL

## Table des matières

- Installation de SQL
- Création d'une base de données
- Utilisation de SQL
- Bonnes pratiques
- Exemple de code SQL

## Installation de SQL

- Télécharge et installe un système de gestion de base de données (SGBD) comme MySQL, PostgreSQL ou SQL Server.
- Pour SQL Server :
  1. Lance le Centre d'installation SQL Server.
  2. Choisis « Nouvelle installation autonome SQL Server ».
  3. Accepte les termes de la licence et configure les options (services, comptes, répertoires de données, etc.).
  4. Clique sur « Installer » pour lancer l'installation.
- Pour MySQL ou PostgreSQL, utilise l'installateur officiel ou les gestionnaires de paquets de ton système d'exploitation.

## Création d'une base de données

1. Connecte-toi à ton SGBD via une interface graphique (ex : MySQL Workbench, pgAdmin) ou en ligne de commande.
2. Crée une base de données :
3. `sql`

```
CREATE DATABASE nom_de_la_base;
```

4.

5. Crée une table :

6. `sql`

```
CREATE TABLE utilisateurs (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nom VARCHAR(50),  
  email VARCHAR(100)  
);
```

7.

## Utilisation de SQL

- SQL (Structured Query Language) permet de créer, modifier, interroger et gérer des bases de données relationnelles.
- Les opérations principales sont :
  - SELECT : lire des données
  - INSERT : ajouter des données
  - UPDATE : modifier des données
  - DELETE : supprimer des données

Exemple de requête SELECT :

```
sql
```

```
SELECT nom, email FROM utilisateurs WHERE nom = 'Dupont';
```

## Bonnes pratiques

- Mets les mots-clés SQL en majuscules pour la lisibilité.
- Indente et aligne les requêtes pour faciliter la lecture et la maintenance.
- Nomme les tables et colonnes de façon logique et cohérente (ex : `utilisateurs, date_creation`).
- Évite d'utiliser `SELECT *` : sélectionne uniquement les colonnes nécessaires.
- Utilise des clauses WHERE précises pour filtrer les données et optimiser les performances.
- Indexe les colonnes souvent utilisées dans les filtres ou les jointures.
- Teste et valide chaque requête avant de l'utiliser en production.
- Sépare les sauvegardes, journaux et données sur des volumes différents pour la performance et la sécurité.

## Exemple de code SQL

```
sql
```

```
-- Création d'une table
```

```
CREATE TABLE produits (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nom VARCHAR(100),  
  prix DECIMAL(10,2)  
);
```

```
-- Insertion de données
INSERT INTO produits (nom, prix) VALUES ('Clavier', 49.99);

-- Lecture de données
SELECT * FROM produits WHERE prix < 100;

-- Mise à jour
UPDATE produits SET prix = 44.99 WHERE nom = 'Clavier';

-- Suppression
DELETE FROM produits WHERE id = 1;
```