

SQL

Introduction

SQL (Structured Query Language) est le langage standard pour interagir avec les bases de données relationnelles.

Il permet d'effectuer :

- **CRUD** → Create, Read, Update, Delete.
- **Administration** → Créer/modifier/supprimer des bases, tables, utilisateurs.
- **Optimisation** → Indices, vues, procédures stockées.

Objectif

- Comprendre la logique relationnelle.
- Être capable d'écrire des requêtes SQL efficaces.
- Suivre les bonnes pratiques de conception de bases de données.

Terminologie essentielle

Terme	Définition
Base de données	Ensemble de tables organisées.
Table	Structure en lignes et colonnes.
Ligne (Row)	Enregistrement d'une table.
Colonne (Field)	Type de donnée d'un champ.
Requête (Query)	Commande SQL envoyée à la base.

Catégories de Commandes SQL

Catégorie	But	Exemples
DDL (Data Definition Language)	Définir la structure	CREATE, ALTER, DROP
DML (Data Manipulation Language)	Manipuler les données	INSERT, UPDATE, DELETE
DQL (Data Query Language)	Interroger la base	SELECT
DCL (Data Control Language)	Gérer les permissions	GRANT, REVOKE

SQL

Principales Commandes SQL

Lire les données — SELECT

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

```
SELECT * FROM table WHERE condition;
```

Insérer des données — INSERT

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

Mettre à jour des données — UPDATE

```
UPDATE table SET colonne1 = 'nouvelle_valeur' WHERE condition;
```

Supprimer des données — DELETE

```
DELETE FROM table WHERE condition;
```

Clauses Utiles

Clause	Utilité
WHERE	Filtrer les résultats
ORDER BY	Trier les résultats
GROUP BY	Grouper des résultats
HAVING	Filtrer après regroupement
LIMIT	Limiter le nombre de résultats

Création et modification de structures (DDL)

Créer une base de données

```
CREATE DATABASE nom_de_la_base;
```

Créer une table

```
CREATE TABLE utilisateurs (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nom VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE,  
  age INT,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

SQL

Modifier une table

```
ALTER TABLE utilisateurs ADD COLUMN adresse VARCHAR(255);
```

Supprimer une table

```
DROP TABLE utilisateurs;
```

Contraintes de tables

Contrainte	Description
PRIMARY KEY	Identifiant unique.
FOREIGN KEY	Référence à une autre table.
UNIQUE	Valeur unique obligatoire.
NOT NULL	Champ obligatoire.
DEFAULT	Valeur par défaut.

Relations entre tables (JOINS)

Type de JOIN	But	Exemple
INNER JOIN	Associer les correspondances	A INNER JOIN B ON A.id = B.a_id
LEFT JOIN	Toutes les lignes de gauche, correspondances ou non	
RIGHT JOIN	Toutes les lignes de droite, correspondances ou non	
FULL JOIN	Toutes les lignes des deux côtés (si supporté)	

Fonctions Utiles

COUNT()	Compter le nombre de lignes
SUM()	Additionner
AVG()	Moyenne
MAX(), MIN()	Valeurs extrêmes

SQL

Exemples :

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

Triggers (Déclencheurs)

Automatiser des actions lors d'INSERT, UPDATE ou DELETE.

Exemple :

```
CREATE TRIGGER trigger_nom  
BEFORE INSERT ON table  
FOR EACH ROW  
BEGIN  
    SET NEW.created_at = NOW();  
END;
```

Type	Quand ?
BEFORE	Avant l'événement
AFTER	Après l'événement
INSTEAD OF	Remplace l'événement (principalement pour les vues)

Utiliser NEW.colonne et OLD.colonne pour accéder aux données.

Transactions

Pour grouper des opérations et garantir leur cohérence :

```
BEGIN;  
UPDATE comptes SET solde = solde - 100 WHERE id = 1;  
UPDATE comptes SET solde = solde + 100 WHERE id = 2;  
COMMIT;  
-- ou ROLLBACK si erreur
```

Bonnes Pratiques SQL

- ✓ Toujours utiliser WHERE dans UPDATE et DELETE.
- ✓ Utiliser des INDEX sur les colonnes filtrées/très recherchées.
- ✓ Normaliser la base (3FN) pour éviter la redondance.
- ✓ Protéger contre l'injection SQL (requêtes préparées).
- ✓ Utiliser des noms clairs (snake_case recommandé).
- ✓ Limiter l'usage de SELECT * en production.

SQL

Types de Données Standards

Type	Usage
INT, BIGINT	Entiers
DECIMAL, FLOAT	Décimaux
VARCHAR(n), TEXT	Textes
DATE, TIME, DATETIME, TIMESTAMP	Dates et heures
BOOLEAN	Vrai/Faux

Outils et Interfaces Recommandés

- DBeaver : Multibase, open-source.
- DataGrip : Puissant IDE SQL.
- phpMyAdmin : Interface Web MySQL.
- pgAdmin : Pour PostgreSQL.
- Azure Data Studio : Pour SQL Server.

Concepts avancés

Concept	Utilité
VUES (CREATE VIEW)	Tables virtuelles pour simplifier les requêtes complexe
PROCÉDURES STOCKÉES (CREATE PROCEDURE)	Scripts SQL précompilés.
INDEXES (CREATE INDEX)	Accélérer la recherche.
CTE (WITH)	Rendre les requêtes lisibles et récursives.
Partitionnement	Diviser de très grandes tables.