

# M164 — Structured Query Language (SQL)

F. Mauron

EPAI

08 décembre 2023

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

# SQL, c'est quoi ?

Le SQL (Structured Query Language) est un langage de programmation standardisé utilisé pour gérer et manipuler des bases de données relationnelles. Il est conçu pour **interroger**, **insérer**, **mettre à jour** et **supprimer** des données stockées dans ces bases de données.

Ses principales composantes et fonctionnalités : - Data Definition Language (DDL)

- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Transaction Control Language (TCL)

# Caractéristiques clés du SQL

- **Langage déclaratif** : SQL permet aux utilisateurs de spécifier ce qu'ils veulent faire avec les données sans décrire comment le faire.
- **Manipulation de données** : SQL est utilisé pour interroger, insérer, mettre à jour et supprimer des données dans les bases de données relationnelles.
- **Gestion de la structure des données** : SQL permet de créer et de modifier la structure des tables de données et d'autres objets associés à une base de données.
- **Gestion des transactions** : SQL permet de gérer les transactions pour assurer l'intégrité des données, en particulier dans les environnements multi-utilisateurs.
- **Interrogation avancée** : SQL offre des fonctions avancées pour des requêtes complexes, incluant le tri, le filtrage, l'agrégation et la jointure de données issues de multiples tables.

# Utilisations typiques du SQL

- **Gestion des bases de données** : SQL est le langage standard pour les systèmes de gestion de base de données relationnelle comme MySQL, PostgreSQL, SQL Server, et Oracle.
- **Analyse de données** : Il est largement utilisé pour l'analyse de données, permettant aux utilisateurs d'extraire et de manipuler des données de manière significative.
- **Développement d'applications** : SQL est couramment utilisé dans le développement d'applications pour interagir avec des bases de données relationnelles.

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

# Définition du langage SQL

Le langage SQL est découpé en quatre principales composantes, à savoir :

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)
- Transaction Control Language (TCL)

Seuls les composantes DDL, DML et une partie DQL du langage vont être abordés dans ce module!



# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

# Data Definition Language (DDL) I/V

Le Data Definition Language (DDL) est un sous-ensemble du langage SQL (Structured Query Language) utilisé pour définir et modifier la structure des bases de données et de leurs objets, tels que les tables, les index, les contraintes, les vues, etc.

# Data Definition Language (DDL) II/V

**CREATE :** Utilisé pour créer de nouveaux objets de base de données, tels que des tables, des vues, des index, etc.

```
CREATE DATABASE SchoolDB;
```

```
CREATE TABLE Students (  
    StudentID int,  
    Name varchar(255),  
    BirthDate datetime,  
    PRIMARY KEY (StudentID)  
);
```

```
CREATE INDEX idx_student_name ON Students (Name);
```

# Data Definition Language (DDL) III/V

**ALTER** : Permet de modifier la structure des objets existants.

- Ajouter une colonne à une table existante :

```
ALTER TABLE Students ADD Email varchar(255);
```

- Modifier le type d'une colonne :

```
ALTER TABLE Students MODIFY COLUMN Name varchar(100);
```

- Supprimer une colonne :

```
ALTER TABLE Students DROP COLUMN Email;
```

- Ajouter une contrainte :

```
ALTER TABLE Students  
  ADD CONSTRAINT fk_department  
  FOREIGN KEY (DepartmentID)  
  REFERENCES Departments(DepartmentID);
```

**DROP** : Supprime des objets existants.

- Supprimer une table :

```
DROP TABLE Students;
```

- Supprimer une base de données :

```
DROP DATABASE SchoolDB;
```

- Supprimer un index :

```
DROP INDEX idx_student_name ON Students;
```

**TRUNCATE** : Supprime rapidement toutes les lignes d'une table, c'est plus rapide que DELETE, car il ne journalise pas chaque ligne supprimée.

```
TRUNCATE TABLE Students;
```

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)**
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

Le langage de manipulation de données SQL est un langage de programmation spécialement conçu pour interagir avec les bases de données relationnelles.

L'une des principales fonctions du SQL est de manipuler les données stockées dans une base de données.



- **INSERT INTO** : Ajoute de nouvelles lignes (enregistrements) à une table.
- **UPDATE** : Modifie les enregistrements existants.
- **DELETE** : Supprime les enregistrements d'une table.

## INSERT INTO

Cette instruction est utilisée pour ajouter de nouvelles lignes de données à une table. Elle permet d'insérer des données dans une table spécifique en spécifiant les valeurs pour chaque colonne ou en les sélectionnant à partir d'une autre table.

```
-- Insérer un nouvel élève  
INSERT INTO Eleve (nom, prenom, adresse)  
VALUES ('Doe', 'John', '123 Main Street');
```

## UPDATE

Cette instruction est utilisée pour mettre à jour des enregistrements existants dans une table. Vous pouvez spécifier les colonnes à mettre à jour et les nouvelles valeurs correspondantes, ainsi que des conditions pour filtrer les enregistrements à mettre à jour.

*-- Mettre à jour l'adresse de l'élève avec ID 1*

**UPDATE** Eleve

**SET** adresse = '456 Elm Street'

**WHERE** ID = 1;

## DELETE

Cette instruction est utilisée pour supprimer des enregistrements d'une table. Vous pouvez spécifier des conditions pour filtrer les enregistrements à supprimer, ou vous pouvez supprimer tous les enregistrements d'une table.

*-- Supprimer tous les élèves de la classe 'A'*

```
DELETE FROM Eleve WHERE classe = 'A';
```

**Attention aux conditions de suppression :** Lors de l'utilisation de l'instruction DELETE, soyez prudent avec les conditions que vous spécifiez. Une mauvaise condition peut entraîner la suppression de données importantes.

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)**
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

Le langage de requête de données SQL (DQL) est la partie du SQL qui permet d'interroger et de récupérer des données à partir d'une base de données relationnelle. Les opérations DQL sont essentielles pour extraire des informations pertinentes de la base de données.

La découverte des possibilités du DQL sera couvert dans le module 106!

## SELECT

Cette instruction est utilisée pour récupérer des données à partir d'une ou plusieurs tables. Elle permet de spécifier les colonnes à récupérer, la table source et les conditions de filtrage.

```
SELECT colonne1, colonne2, ...  
FROM nom_de_la_table  
WHERE condition;
```

- colonne1, colonne2, ... : Les colonnes que vous souhaitez récupérer.
- nom\_de\_la\_table : La table à partir de laquelle vous récupérez les données.
- condition : (Facultatif) Les critères pour filtrer les données.

## Distinct

L'opérateur DISTINCT est utilisé pour éliminer les doublons des résultats de la requête. Il retourne uniquement des valeurs uniques dans les colonnes spécifiées.

*-- Sélectionner les noms uniques des classes*

**SELECT DISTINCT** Classe

**FROM** Eleve;



## ORDER BY

L'opérateur ORDER BY est utilisé pour trier les résultats de la requête selon une ou plusieurs colonnes, dans un ordre ascendant (ASC) ou descendant (DESC).

*-- Sélectionner les noms des élèves de la classe 'A' triés par ordre alphabétique*

```
SELECT Nom  
FROM Eleve  
WHERE Classe = 'A'  
ORDER BY Nom ASC;
```

## Conseils

- Utilisation des wildcards : Les wildcards (jokers) comme % et \_ sont utiles pour effectuer des recherches partielles ou génériques dans les données.
- Requêtes complexes : Vous pouvez combiner plusieurs clauses SELECT pour créer des requêtes complexes, et utiliser les opérateurs logiques (AND, OR, NOT) pour spécifier des conditions multiples.
- Utilisation de fonctions : SQL propose de nombreuses fonctions intégrées (telles que SUM, COUNT, AVG, etc.) pour effectuer des calculs sur les données récupérées.

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)**
- 7 Transaction Control Language (TCL)

# Data Control Language (DCL)

- **GRANT** : Donne des droits d'accès aux utilisateurs.
- **REVOKE** : Supprime des droits d'accès aux utilisateurs.

# Plan

- 1 Introduction
- 2 Définition du langage SQL
- 3 Data Definition Language (DDL)
- 4 Data Manipulation Language (DML)
- 5 Data Query Language (DQL)
- 6 Data Control Language (DCL)
- 7 Transaction Control Language (TCL)

# Transaction Control Language (TCL)

- **COMMIT** : Valide une transaction et rend toutes les modifications permanentes.
- **ROLLBACK** : Annule les modifications depuis le dernier COMMIT.
- **SAVEPOINT** : Crée des points dans une transaction pour lesquels un ROLLBACK peut être effectué.