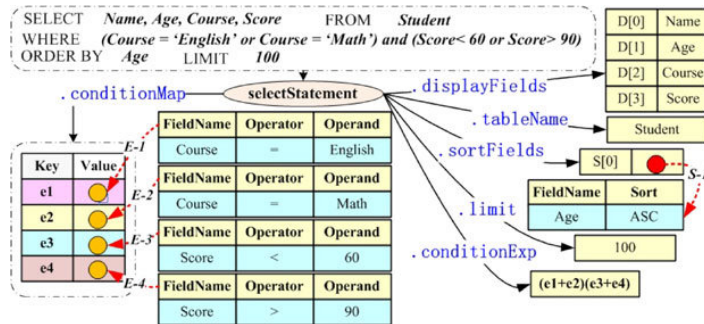
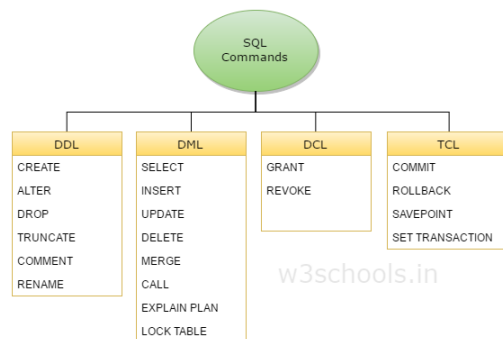


# Le Langage SQL



Le SQL (Structured Query Language) est un langage standardisé utilisé pour interagir avec les bases de données relationnelles. SQL permet de définir, manipuler et contrôler les données dans une base de données. Voici un aperçu détaillé des principales fonctionnalités et commandes SQL :

## 1. Définition de la Structure des Données (DDL - Data Definition Language)



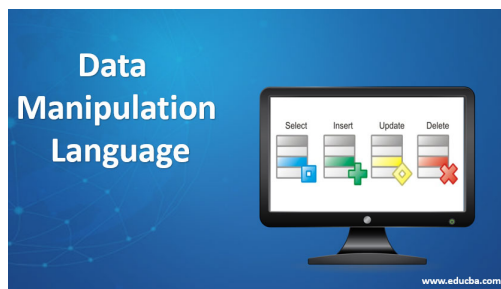
Les commandes DDL sont utilisées pour définir et gérer la structure des objets de base de données comme les tables, les index et les vues.

- **CREATE** : Créer des objets dans la base de données.

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    BirthDate DATE
);
```

- **ALTER** : Modifier la structure d'un objet existant.  
`ALTER TABLE Employees ADD Email VARCHAR(100);`
- **DROP** : Supprimer des objets de la base de données.  
`DROP TABLE Employees;`

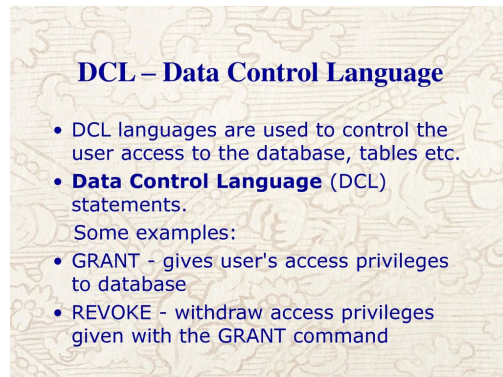
## 2. Manipulation des Données (DML - Data Manipulation Language)



Les commandes DML permettent de manipuler les données à l'intérieur des objets de base de données.

- **SELECT** : Récupérer des données à partir de la base de données.  
`SELECT FirstName, LastName FROM Employees WHERE BirthDate > '1980-01-01';`
- **INSERT** : Insérer des données dans une table.  
`INSERT INTO Employees (EmployeeID, FirstName, LastName, BirthDate)  
VALUES (1, 'John', 'Doe', '1985-05-15');`
- **UPDATE** : Mettre à jour des données existantes dans une table.  
`UPDATE Employees SET Email = 'john.doe@example.com' WHERE EmployeeID = 1;`
- **DELETE** : Supprimer des données d'une table.  
`DELETE FROM Employees WHERE EmployeeID = 1;`

### 3. Contrôle des Données (DCL - Data Control Language)



Les commandes DCL sont utilisées pour contrôler l'accès aux données dans la base de données.

- **GRANT** : Accorder des permissions à un utilisateur.

```
GRANT SELECT, INSERT ON Employees TO 'username';
```

- **REVOKE** : Révoquer des permissions accordées à un utilisateur.

```
REVOKE INSERT ON Employees FROM 'username';
```

### 4. Contrôle des Transactions (TCL - Transaction Control Language)



Les commandes TCL gèrent les transactions dans la base de données, permettant de garantir l'intégrité des données.

- **COMMIT** : Valider les transactions courantes.

```
COMMIT;
```

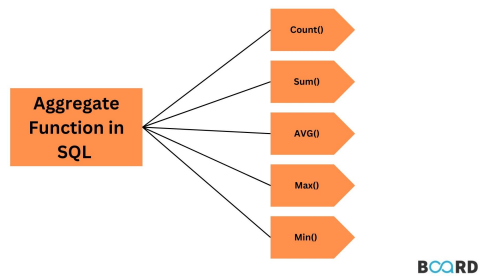
- **ROLLBACK** : Annuler les transactions courantes.

**ROLLBACK;**

- **SAVEPOINT** : Définir un point de sauvegarde dans une transaction pour un retour possible à cet état.

**SAVEPOINT** Savepoint1;

## 5. Fonctions et Opérations



SQL propose de nombreuses fonctions intégrées pour manipuler les données.

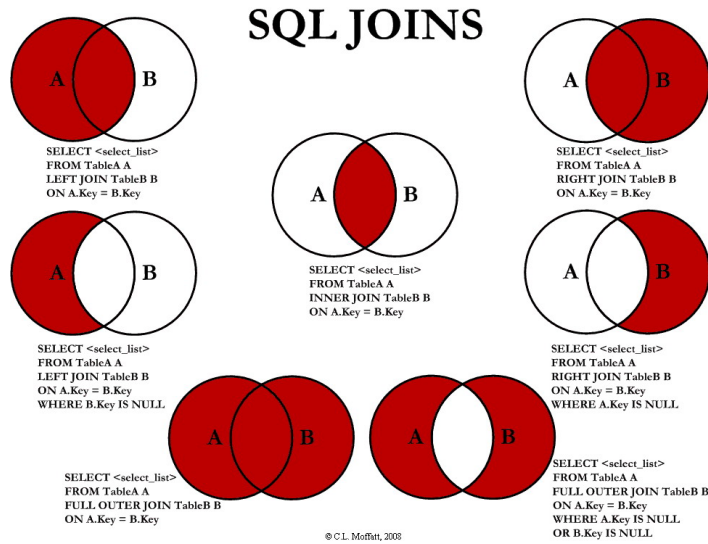
- **Fonctions agrégées** : SUM, COUNT, AVG, MAX, MIN.

**SELECT** COUNT(\*) **FROM** Employees;

- **Fonctions scalaires** : UPPER, LOWER, SUBSTRING, LENGTH.

**SELECT** UPPER(FirstName) **FROM** Employees;

## 6. Joins



Les jointures permettent de combiner des lignes provenant de deux ou plusieurs tables en fonction d'une condition liée à leurs colonnes.

- **INNER JOIN** : Renvoie les enregistrements ayant des valeurs correspondantes dans les deux tables.

```
SELECT Employees.FirstName, Departments.DepartmentName
FROM Employees
INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

- **LEFT JOIN** : Renvoie tous les enregistrements de la table de gauche et les enregistrements correspondants de la table de droite.

```
SELECT Employees.FirstName, Departments.DepartmentName
FROM Employees
LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

- **RIGHT JOIN** : Renvoie tous les enregistrements de la table de droite et les enregistrements correspondants de la table de gauche.

```
SELECT Employees.FirstName, Departments.DepartmentName
FROM Employees
RIGHT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

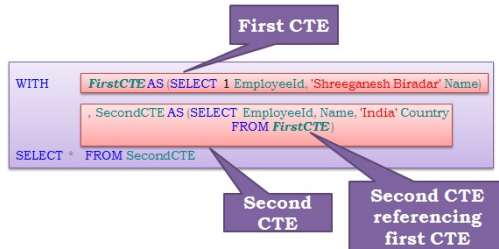
- **FULL JOIN** : Renvoie tous les enregistrements lorsque qu'il y a une correspondance dans l'une des tables.

```

SELECT Employees.FirstName, Departments.DepartmentName
FROM Employees
FULL OUTER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID

```

## 7. Sous-requêtes et CTE (Common Table Expressions)



- **Sous-requêtes** : Une requête imbriquée à l'intérieur d'une autre requête.

```

SELECT FirstName, LastName FROM Employees
WHERE EmployeeID IN (SELECT EmployeeID FROM Projects WHERE ProjectName = 'Alpha')

```

- **CTE** : Une expression de table temporaire utilisée dans une requête.

```

WITH EmployeeCTE AS (
    SELECT FirstName, LastName FROM Employees WHERE BirthDate > '1980-01-01'
)
SELECT * FROM EmployeeCTE;

```

## Conclusion

SQL est un langage puissant et polyvalent pour la gestion des bases de données relationnelles. Il permet aux utilisateurs de définir, manipuler et contrôler les données de manière efficace et flexible, tout en garantissant l'intégrité et la sécurité des données.