DDL and DML Commands in SQL

Overview

Data Definition Language (DDL) and **Data Manipulation Language (DML)** are two fundamental components of SQL used for managing and manipulating databases.

DDL (Data Definition Language) Commands

DDL commands are used to define and manage the structure of database objects. Here are some key DDL commands:

1. **CREATE**: Used to create database objects such as tables, views, indexes, and more.
    - Example:

```
CREATE TABLE Employees (
    EmployeeID INT,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Department VARCHAR(255)
);
```

2. **ALTER**: Used to modify the structure of an existing database object.
    - Example:

```
ALTER TABLE Employees
ADD Salary INT;
```

3. **DROP**: Used to delete an entire object or part of an object from the database.
    - Example:

```
DROP TABLE Employees;
```

4. **TRUNCATE**: Used to delete all records from a table but does not remove the table structure.
    - Example:

```
TRUNCATE TABLE Employees;
```

5. **RENAME**: Used to rename an existing database object.

- Example:

```
RENAME TABLE Employees TO Staff;
```

DML (Data Manipulation Language) Commands

DML commands are used to manipulate data within a database. Key DML commands include:

1. **INSERT**: Used to add new records to a database table.
   - Example:

```
INSERT INTO Employees (EmployeeID, FirstName, LastName, Department)
VALUES (1, 'John', 'Smith', 'IT');
```

2. **UPDATE**: Used to modify existing records in a database table.
   - Example:

```
UPDATE Employees
SET Salary = 50000
WHERE EmployeeID = 1;
```

3. **DELETE**: Used to delete existing records from a database table.
   - Example:

```
DELETE FROM Employees
WHERE EmployeeID = 1;
```

4. **SELECT**: Used to retrieve data from one or more tables.
   - Example:

```
SELECT * FROM Employees;
```

5. **MERGE**: Used to combine data from two or more tables into one.
   - Example:

```
MERGE INTO TargetTable AS target
USING SourceTable AS source
ON target.ID = source.ID
WHEN MATCHED THEN
```

```
    UPDATE SET target.value = source.value
WHEN NOT MATCHED THEN
    INSERT (ID, value) VALUES (source.ID, source.value);
```

6. **CALL**: Used to call a stored procedure or function.
    - Example:

```
CALL UpdateEmployeeSalary(1, 50000);
```

DDL vs. DML Commands

DDL	DML
Defines database objects like tables, indexes, and views.	Manipulates data within the database.
Examples: `CREATE`, `ALTER`, `DROP`, `TRUNCATE`.	Examples: `SELECT`, `INSERT`, `UPDATE`, `DELETE`.
Changes affect the structure of the database.	Changes affect the data stored in the database.
Not transactional; cannot be rolled back.	Transactional; can be rolled back if necessary.
Typically executed by database administrators.	Executed by application developers or end-users.
Used during database design and setup.	Used during normal operation of a database.

Benefits of DDL and DML Commands

- **DDL Commands**:
    - Define and manage database structures.
    - Ensure data integrity by defining constraints and relationships.
    - Control database access by creating and modifying users and permissions.
- **DML Commands**:
    - Manipulate data efficiently.
    - Ensure accurate and up-to-date data in the database.
    - Support complex queries and data retrieval operations.

Best Practices

- **Use Specific DDL Statements**: Avoid unintended side effects by being specific (e.g., `DROP TABLE` instead of `DROP DATABASE`).
- **Use Transactions for Multiple Changes**: Ensure atomicity and consistency of multiple changes.
- **Avoid DDL in Stored Procedures**: Prevent unintentional side effects and maintain clarity.
- **Validate DML Operations**: Ensure data changes are valid and appropriate for the current database state.
- **Regular Backups**: Protect against data loss and ensure recovery options are available.

Key Takeaways

- **DDL**: Used for defining and modifying database structures.
- **DML**: Used for manipulating and querying data.
- Both DDL and DML are essential for effective database management and operation.

This overview should provide a solid foundation for understanding and using SQL DDL and DML commands effectively. If you have any specific questions or need further examples, feel free to ask!