

## DATA MANIPULATION LANGUAGE (DML)

DML is an acronym for Data Manipulation Language. It is a subset of SQL (Structured Query Language) specifically dedicated to data manipulation operations within a relational database. In other words, DML focuses on how to interact with data already existing in tables, allowing you to:

- Retrieve data (query the database).
- Insert new data into tables.
- Modify existing data in tables.
- Delete data from tables.

The main DML statements in SQL are:

- **SELECT**: Used to retrieve data from one or more tables. It allows you to specify which columns to select, which rows to filter (via **WHERE** clauses), how to order the results, and much more.
- **INSERT**: Used to add new rows (records) to a table. You specify the destination table and the values to insert for each column.
- **UPDATE**: Used to modify the values of existing columns in one or more rows of a table. A **WHERE** clause is usually used to specify which rows should be updated.
- **DELETE**: Used to delete rows from a table. Again, a **WHERE** clause is often used to specify which rows should be deleted.

Main Features of DML:

1. **Manipulating Existing Data**: The fundamental characteristic of DML is its focus on manipulating the data contained in tables. Unlike DDL (Data Definition Language) which deals with the structure of the database (creating tables, columns, etc.), DML operates directly on the data.

2. **Interacting with Data**: DML provides the commands needed to interact fully with data: it is possible to read, write, modify and delete information. This ability to interact is crucial for any application that uses a database.

3. **Oriented towards CRUD (Create, Read, Update, Delete) Operations**: DML statements implement the fundamental operations of CRUD (Create, Read, Update, Delete), which are the four basic operations for managing data in any information system.

- o **SELECT** corresponds to Read
- o **INSERT** corresponds to Create
- o **UPDATE** corresponds to Update
- o **DELETE** corresponds to Delete

4. **Use of WHERE Clauses**: Many DML statements, especially **SELECT**, **UPDATE** and **DELETE**, use the **WHERE** clause to filter data. The **WHERE** clause allows you to specify conditions that must be satisfied for DML operations to be applied to specific rows. This allows you to operate precisely and targeted on the desired data.

5. **Transactionality (in transactional contexts)**: In many relational database systems, DML operations are transactional. This means that a group of DML operations can be treated as a single unit of work. If all operations within the transaction succeed, the changes are committed (made permanent). If any operation fails, the entire transaction is canceled (rolled back), returning the database to its previous state. Transactionality ensures the consistency and integrity of data, even in the event of errors or interruptions.

6. **Part of the Standard SQL Language**: DML is an integral and standardized part of the SQL language. DML statements are generally compatible between different relational database systems (with possible small syntactic variations specific to the SQL dialect).

7. **Essential for Data-Driven Applications**: DML is essential for the development of data-driven applications, that is, applications that base their operation on the management and processing of data. Without DML, applications could not interact dynamically with the information stored in the database.

8. **Power and Flexibility**: DML offers great power and flexibility in data manipulation. Through combinations of statements and clauses, it is possible to perform complex operations of querying, updating and managing data efficiently.

In short, DML is the part of SQL that allows you to actively work with data within a database. Its statements are fundamental for any application that needs to store, retrieve, modify or delete information in a structured and reliable way.