SQL basics

Data Manipulation Language

SQL basics

Data Manipulation Language (DML)

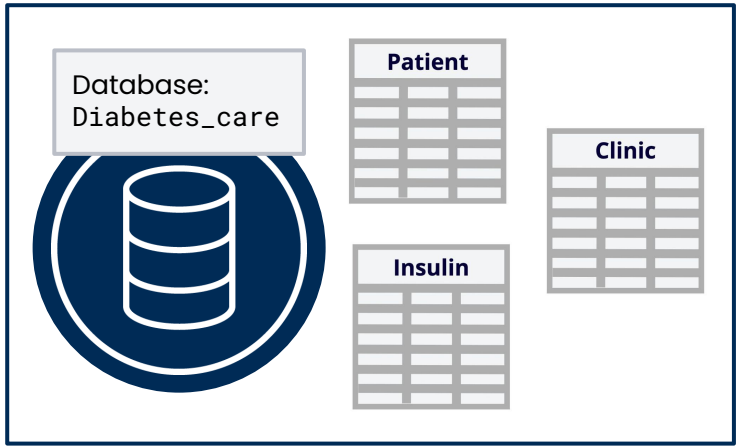
DML is a **sublanguage of SQL** responsible for **manipulating data** in a database.

Most commonly used to **add**, **edit**, or **delete** data from a database.

The main DML statements include **`INSERT`**, **`UPDATE`**, and **`DELETE`**.

DML in practice

To better understand the three basic DML statements, **INSERT**, **UPDATE**, **and DELETE**, let's consider the following **scenario.**



We are data professionals in charge of **managing the database** for an organization that operates a type 1 **diabetes care program**. We will focus on three tables:

- **Patient**: Stores details on the diabetes patients, such as their name and age.
- **Clinic**: Stores details on all the clinics that administer insulin to patients, such as location and insulin vials available.
- **Insulin**: Stores details on the insulin vials administered, such as the type and expiry date.

Now let's explore how we can use DML to manage the `Diabetes_care` database.

SQL basics

INSERT

The `INSERT` statement is used to **add new data** to a database table by inserting new records.

It allows us to **insert** new data into **specific columns** of a table by **providing values** that correspond to the column definitions.

Scenario: If we receive a new patient at a clinic, we will use it to add a new record for the patient to our `Patient` table.

INSERT syntax

The INSERT statement contains **two key parts**, the **`INSERT INTO`** clause and the **`VALUES`** clause.

Syntax

```
INSERT INTO
      Database_name.table_name
      (column1, column2)
VALUES
      (value1, value2),
      (value1, value2);
```

Specify the **table name**, the **columns** to be inserted, and the **corresponding values** for each column. Each list of values (`value1, value2`) becomes a new record in the table.

Example

```
INSERT INTO
    Diabetes_care.Patient
    (First_name, Last_name)
VALUES
    ('Kennedy', 'Ngoma'),
    ('Mulenga', 'Mwamba');
```

1. Specify the **table** to insert the data (**`Patient`**).
2. Specify the **column** names (**`First_name, Last_name`**).
3. Specify the **values** to be inserted.

Note: Make sure the **order** of the **values** is in the **same order** as the **columns** in the table.

SQL basics

UPDATE

The **UPDATE** statement allows us to **modify existing data** within a database table.

It only modifies the data in a table and **does not alter its structure.**

Scenario: If a clinic finishes its insulin supply, we will update the specific clinic's record to reflect this in the `Clinic` table.

UPDATE syntax

The **UPDATE** statement contains **two key parts**, the **UPDATE** and **SET** clauses. We can optionally include the **WHERE** clause to specify a **condition** for the update.

Syntax

```
UPDATE database_name.table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

It specifies the **table name**, the **columns** to be updated, the **new values** for each column, and the **condition** that should be true for the update to be implemented.

Example

```
UPDATE Diabetes_care.Clinic
SET Doses_available = 0
WHERE Clinic_name = 'Kasama';
```

1. Specify the **table name** for the update (**`Clinic`**).
2. Specify **each column** and the **values** the columns should be updated to (**`Doses_available`** updated to **0**).
3. Specify the **condition** for the update (**`Clinic_name`** is `Kasama`).

Note: The **`WHERE`** keyword specifies a condition for updates in a table, ensuring that changes are made only where the condition is true. *WHERE will be covered in detail later on.*

SQL basics

DELETE

The **DELETE** statement is used to **remove specific records** from a database table.

It provides a means to **selectively delete data** based on **specified conditions** allowing for the precise removal of unwanted or outdated records.

Scenario: If an insulin dose expires we can delete the record from the Insulin table. However, this should be done with caution due to potential effects on related records within the database.

DELETE syntax

The **DELETE** statement contains **two key parts**, the **DELETE FROM** and **WHERE** clauses.

Syntax

```
DELETE FROM database_name.table_name
WHERE condition;
```

It specifies the **table name** for the delete operation and the **condition** that must be true for the record to be deleted.

Example

```
DELETE FROM Diabetes_care.Insulin
WHERE Expiry_year = '2022';
```

1. Specify the **table** to delete from (**Insulin**).
2. Specify the **condition** for the delete (**Expiry_year** is equal to '2022').

Note: The **WHERE** keyword specifies a condition for deletes in a database, ensuring that records are only deleted where the condition is true. If **WHERE** is left out, every record in the table will be deleted.

Summary of DML statements

INSERT

The **INSERT** statement is used to **add new records** to a database table.

It specifies the table name, the columns to be inserted, and the corresponding values for each column.

UPDATE

The **UPDATE** statement is used to **modify existing records** in a database table.

It specifies the table name, the columns to be updated, and the new values for each column.

DELETE

The **DELETE** statement is used to **remove records** from a database table.

It specifies the table name for the delete operation and the condition that must be true for the record to be deleted.

Always use the **WHERE** clause for **selective updates** and **deletions.**