

# Manipulación de datos – DML de SQL

## ¿Qué es DML?

Como ya vimos en la introducción del tema anterior, DML significa Lenguaje de Manipulación de Datos. Dentro de este grupo de sentencias de SQL, se encuentran las sentencias para manipular los datos de nuestra base de datos, esto es consultar datos, añadir, modificar o eliminar datos existentes

## Inserción de datos

Para insertar nuevos datos en las bases de datos relacionales, SQL nos proporciona la instrucción **INSERT INTO**, cuyo esquema es el siguiente:

```
INSERT INTO nombreTabla (columna1, columna2, ...)  
VALUES (valor1, valor2, ...);
```

Así, si tenemos una tabla llamada *clientes*, con los campos *dni*, *nombre*, *apellidos*, *dirección* y *teléfono*, la consulta para introducir un nuevo registro en dicha tabla sería:

```
INSERT INTO clientes (dni, nombre, apellidos, direccion, telefono)  
VALUES ('12345678A', 'Fernando', 'García Rodríguez', 'C/Virgen del Rosal 28', 111222333);
```

## Actualización de datos

En cambio, cuando se desea modificar datos existentes en la base de datos, la sentencia a emplear es **UPDATE**, la cual sigue el esquema:

```
UPDATE nombreTabla  
SET columna1 = nuevoValor1, columna2 = nuevoValor2  
WHERE condición;
```

Siguiendo con el ejemplo del apartado anterior, imaginemos que deseamos modificar el teléfono de *Fernando*, la sentencia a emplear sería:

```
UPDATE clientes  
SET telefono = 961234567  
WHERE dni = '12345678A';
```

Si se desea modificar varios datos simultáneamente, bastaría con indicarlo uno a continuación del otro en la línea del SET, separando mediante comas:

```
UPDATE clientes  
SET telefono = 961234567, direccion = 'Mayor 13'  
WHERE dni = '12345678A';
```

Cabe destacar en este punto la importancia de indicar correctamente la condición mediante la cláusula **WHERE**, ya que sin ella la sentencia afectaría a todos los registros de la tabla, por tanto una sentencia como la siguiente, le asignaría a todos los clientes el teléfono 960000000:

```
UPDATE clientes  
SET telefono = 960000000;
```

## Borrado de datos

Para finalizar, sólo nos queda conocer la sentencia necesaria para eliminar datos de nuestra base de datos, acción que se realiza mediante la cláusula **DELETE FROM**:

```
DELETE FROM nombreTabla  
WHERE condición;
```

Siguiendo con el ejemplo del apartado anterior, si deseamos eliminar el registro anteriormente introducido, podríamos hacerlo mediante la sentencia:

```
DELETE FROM clientes  
WHERE dni = '12345678A';
```

Cabe, de nuevo, destacar la importancia de la cláusula **WHERE** para filtrar y seleccionar los registros a eliminar, así una sentencia como la que sigue, borraría no sólo a éste usuario, sino a todos los llamados *Fernando*:

```
DELETE FROM clientes  
WHERE nombre = 'Fernando';
```

Y si no se indicara ninguna condición, la sentencia borraría por completo el contenido de la tabla *clientes*:

```
DELETE FROM clientes;
```