



UNIVERSIDAD DE SONSONATE

USO

SCIENTIAE ET BONIS ARTIBUS

INGENIERÍA EN SISTEMAS COMPUTACIONALES
LABORATORIO DE BASES DE DATOS I

Guía 4
DML (Data
Manipulation
Language)



MySQL™

INTRODUCCIÓN

DML

El **Lenguaje de Manipulación de Datos (DML)**, siglas en inglés de **Data Manipulation Language** es el que permite que una base de datos cumpla con su cometido de almacenar y manipular información de una entidad. Este lenguaje se usa luego de diseñar la base de datos (DDL), ya que tenemos que saber dónde estarán los datos para poder manipularlos. El DML nos permite hacer las operaciones básicas de la base de datos: inserción (INSERT), recuperación (SELECT), actualización (UPDATE) y eliminación (DELETE), que en una aplicación real conoceremos como CRUD (Create, Read, Update y Delete).

- Comprender las operaciones básicas de DML.
- Ingresar datos en la base de datos de distintas formas.
- Leer datos en la base de datos.
- Actualizar datos en la base de datos.
- Eliminar datos en la base de datos.

Restricción en Cascada

Se refiere a la propagación automática de cambios realizados en una tabla principal a las tablas relacionadas a través de las restricciones de clave externa. Cuando se crea una relación entre dos tablas en SQL mediante una restricción de clave externa, se establece una dependencia entre las dos tablas. La tabla secundaria tiene una clave externa (FK) que hace referencia a la clave principal (PK) de la tabla principal. Si se realizan acciones en la tabla principal, los cambios se propagan a la tabla secundaria.

Por ejemplo, si tenemos dos tablas: "clientes" y "órdenes". La tabla "órdenes" tiene una clave externa que hace referencia a la clave principal de la tabla "clientes". Si un cliente se elimina de la tabla "clientes", la restricción de clave externa en la tabla "órdenes" se activa y elimina automáticamente todas las órdenes relacionadas con ese cliente.

DESARROLLO

Sentencia INSERT

Como recomendación es mejor iniciar la inserción de datos desde las tablas padres, para que ésta se haga en cascada.

Existen 3 formas de insertar datos en una tabla:

- **Inserción implícita** (no se definen los campos que se quieren insertar):
Esto se hace cuando no se va a insertar en un orden diferente del cual han sido creados los campos.
- **Inserción explícita** (se definen los campos que se quieren insertar):
Esto se hace cuando se va a insertar en un orden diferente del cual han sido creados los campos, sin embargo también puede realizarse si se respeta dicho orden.
- **Varios registros a la vez.**

Ejemplos de sentencia INSERT

- **Inserción explícita** (se definen los campos que se quieren insertar):

```
INSERT INTO estudiante(nie, nombres, apellidos, fecha_nacimiento)
VALUES ('0055874-1', 'Esmeralda', 'Ramos', '1998-09-14');
```

- **Inserción implícita** (no se definen los campos que se quieren insertar):

```
INSERT INTO estudiante
VALUES ('0034578-4', 'Juan', 'Paredes', '1997-08-13' , '7777-8888');
```

- **Inserción múltiple implícita** (se puede ejecutar sin especificar los campos, sin embargo, deben llenarse todos los campos, inclusive si se tiene llaves primarias autoincrementables):

```
INSERT INTO estudiante VALUES  
  ('1', 'Manuel', 'Rodríguez', '1987-12-12', '7666-6666'),  
  ('2', 'Sandra', 'Torres', '1957-05-05', '7999-8888');
```

- **Inserción múltiple explícita** (se puede ejecutar especificando los campos, y si se tiene llaves primarias autoincrementables, estas se omiten en el llenado):

```
INSERT INTO estudiante (nombres, apellidos, fechaNacimiento, teléfono)  
VALUES  
  ('Manuel', 'Rodríguez', '1987-12-12', '7666-6666'),  
  ('Sandra', 'Torres', '1957-05-05', '7999-8888');
```

Sentencia SELECT

- **SELECT con todos los campos de la tabla:**

```
SELECT * FROM estudiante;
```

- **SELECT con campos específicos de la tabla:**

```
SELECT nie, nombres, fecha_nacimiento FROM estudiante;
```

- **SELECT con registros específicos de la tabla:**

```
SELECT nie, nombres, fecha_nacimiento FROM estudiante  
WHERE nie = '12345679';
```

Sentencia UPDATE

- **UPDATE con un campo de la tabla** (todos los registros se actualizan):

```
UPDATE estudiante SET nombres = 'Juan';
```

- **UPDATE con varios campos de la tabla y condición “WHERE”** (sólo los registros que cumplan la condición):

```
UPDATE estudiante  
SET nombres = 'Fernando', apellidos = 'López'  
WHERE nie = '123456789';
```

Sentencia DELETE

- **DELETE con todos los registros de la tabla:**

```
DELETE FROM estudiante;
```

- **DELETE con condición "WHERE"** (sólo los registros que cumplan la condición):

```
DELETE FROM estudiante WHERE apellidos = 'Rodríguez';
```

[El archivo SQL con el DML de la base de datos escolar se encuentra en la carpeta Recursos]

EJERCICIOS

Utilizando el lenguaje DML realizar los siguientes ejercicios, en la base de datos correspondiente a su grupo:

- Realizar **todas las inserciones necesarias para llenar la base de datos**, por lo menos con **5 registros por cada tabla** según sea necesario.
- Realizar 2 consultas, que lleven al menos, un campo definido y WHERE.
- Realizar 2 actualizaciones que lleven al menos, un campo definido y WHERE.
- Realizar 2 eliminaciones que lleven WHERE.
- Se realizará un archivo .sql por cada grupo.
- Cada estudiante debe tener una copia del archivo realizado en su repositorio remoto independiente (Guía 4)