

1

Ejercicios prácticos

**Unidades**: 5

# Guía

* Se plantean diferentes ejercicios para practicar con sentencias DML y DCL.
* Es necesario disponer de un servidor de base de datos (MySQL o MariaDB) y un cliente gráfico.
* Puede existir diferentes soluciones en un ejercicio. Hay que encontrar la solución más efectiva y eficiente.
* La estructura de la base de datos es la siguiente:

CREATE DATABASE pedidos; USE pedidos;

CREATE TABLE CLIENTES (

idCliente INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(255) NOT NULL,

apellido VARCHAR(255) NOT NULL, email VARCHAR(255) UNIQUE NOT NULL,

telefono VARCHAR(20), direccion VARCHAR(255)

);

CREATE TABLE CATEGORIAS (

idCategoria INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(255) NOT NULL,

descripcion TEXT

);

CREATE TABLE PRODUCTOS (

idProducto INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(255) NOT NULL,

descripcion TEXT,

precio DECIMAL(10,2) NOT NULL,

idCategoria INT,

FOREIGN KEY (idCategoria) REFERENCES Categorias(idCategoria)

);

CREATE TABLE PEDIDOS (

idPedido INT AUTO\_INCREMENT PRIMARY KEY,

idCliente INT NOT NULL, fecha DATETIME NOT NULL, total DECIMAL(10,2),

FOREIGN KEY (idCliente) REFERENCES Clientes(idCliente)

);

1

CREATE TABLE LINEAS\_PEDIDOS (

idLineaPedido INT, idPedido INT, idProducto INT, cantidad INT NOT NULL,

precioUnitario DECIMAL(10,2) NOT NULL, descuento DECIMAL(10,2) NOT NULL, total DECIMAL(10,2) NOT NULL,

PRIMARY KEY (idLineaPedido, idPedido, idProducto), FOREIGN KEY (idPedido) REFERENCES Pedidos(idPedido), FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)

);

# Ejemplo

Inserta un registro en la tabla categorías y muestra todas las categorías con una sentencia sql.

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Electrónica', 'Productos electrónicos y dispositivos');

SELECT \* FROM CATEGORIAS;

# Ejercicio 1.

Realiza la siguiente carga de datos.

-- Categorias

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Electrónica', 'Productos electrónicos y dispositivos');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Ropa', 'Ropa y accesorios');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Juguetes', 'Juguetes y juegos para niños y adultos');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Libros', 'Libros de diferentes géneros, incluidos ficción, no ficción, educativos y más.');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Juguetes', 'Juguetes y juegos para niños y adultos');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Hogar', 'Artículos para el hogar'); INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Deportes', 'Productos y accesorios relacionados con deportes y actividades al aire libre.');

INSERT INTO CATEGORIAS (nombre, descripcion) VALUES ('Electrónica', 'Productos electrónicos y dispositivos');

-- Clientes

INSERT INTO CLIENTES (nombre, apellido, email, telefono, direccion) VALUES ('Juan', 'Pérez', 'juan.perez@email.com', '+1234567890', 'Calle 123, Ciudad A');

INSERT INTO CLIENTES (nombre, apellido, email, telefono, direccion) VALUES ('María', 'González', 'maria.gonzalez@email.com', '+2345678901', 'Avenida 456, Ciudad B');

INSERT INTO CLIENTES (nombre, apellido, email, telefono, direccion) VALUES ('Carlos', 'Ramírez', 'carlos.ramirez@email.com', '+3456789012', 'Calle 789, Ciudad C');

INSERT INTO CLIENTES (nombre, apellido, email, telefono, direccion) VALUES ('Ana', 'Rodríguez', 'ana.rodriguez@email.com', '+4567890123', 'Avenida 321, Ciudad D');

INSERT INTO CLIENTES (nombre, apellido, email, telefono, direccion) VALUES ('Pedro', 'López', 'pedro.lopez@email.com', '+5678901234', 'Calle 654, Ciudad E');

-- Productos

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Camiseta Marvel', 'Camiseta con estampado de personajes de Marvel', 19.99, 2);

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Taza de café de Friends', 'Taza de café con el logo de Central Perk de la serie Friends', 12.99, 5);

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Póster Pulp Fiction', 'Póster de la icónica película Pulp Fiction de Quentin Tarantino', 14.99, 3);

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Funko Pop de Star Wars', 'Figura coleccionable Funko Pop de un personaje de Star Wars', 24.99, 3);

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Libro de cocina de películas', 'Libro de cocina con recetas inspiradas en películas famosas', 29.99, 4);

INSERT INTO PRODUCTOS (nombre, descripcion, precio, idCategoria) VALUES ('Peluche de Toy Story', 'Peluche de un personaje de la película animada Toy Story', 34.99, 3);

Responde a las siguientes preguntas:

1. Pon una captura donde se vean todas las categorías. ¿Existen categorías repetidas?

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Hay un total de 3 categorías repetidas.

1. Muestra todas las sentencias que son necesarias para que aparezca solo una categorías de cada tipo.

select \* from productos where idCategoria = 1 or idCategoria = 9 or idCategoria = 3 or idCategoria = 5;

update productos set idCategoria = 3 where idCategoria = 5;

select \* from productos where idCategoria = 1 or idCategoria = 9 or idCategoria = 3 or idCategoria = 5;

delete from categorias where idCategoria = 5 and idCategoria = 8;

select \* from categorias;

1. Revisar si cada categoría tiene el identificador como muestra la tabla. Si no, realiza los cambios oportunos y muestra las sentencias que han sido necesarias.

|  |  |
| --- | --- |
| **Identificador** | **Categoría** |
| 1 | Electrónica |
| 2 | Ropa |
| 3 | Juguetes |
| 4 | Libros |
| 5 | Hogar |
| 6 | Deportes |

update categorias set idCategoria = 5 where idCategoria = 6;

update categorias set idCategoria = 6 where idCategoria = 7;

select \* from categorias;

1. Revisar que los productos tienen la categoría correcta. Si no es así, realiza los cambios oportunos y muestra las sentencias que han sido necesarias para corregirlo.

|  |  |
| --- | --- |
| **Producto** | **Categoría** |
| Camiseta Marvel | Ropa |
| Taza de café de Friends | Hogar |
| Póster Pulp Fiction | Juguetes |
| Funko Pop de Star Wars | Juguetes |
| Libro de cocina de películas | Libros |
| Peluche de Toy Story | Juguetes |

update productos set idCategoria = 5 where idCategoria = 3 and idProducto = 2;

select\* from productos;

select productos.nombre, categorias.nombre as nombre\_categoria

from productos

inner join categorias on productos.idCategoria = categorias.idCategoria;

# Ejercicio 2.

Realiza la siguiente carga de datos.

-- Pedidos

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (3, 3, '2022-04-10 20:10:00',

319.98);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (4, 4, '2022-05-01 11:15:00',

99.98);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (5, 4, '2022-12-07 16:00:00',

49.99);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (6, 5, '2023-01-22 18:30:00',

137.94);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (7, 2, '2023-02-09 17:00:00',

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (1, 1, '2022-01-15 10:30:00',

549.98);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (2, 2, '2022-03-24 15:45:00', 19.99)

49.98);

INSERT INTO PEDIDOS (idPedido, idCliente, fecha, total) VALUES (8, 2, '2023-02-19 11:10:00',

19.99); ;

-- Lineas pedidos

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (1, 1, 1, 20, 19.99, 11.17, 355.14);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (2, 1, 2, 15, 12.99, 0.00, 499.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (3, 2, 5, 1, 29.99, 33.34, 19.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (4, 3, 3, 15, 14.99, 22.15, 175.04);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (5, 3, 4, 5, 24.99, 0.00, 124.95);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (6, 3, 5, 1, 29.99, 33.34, 19.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (7, 4, 6, 2, 34.99, 28.55, 50);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (8, 4, 4, 2, 24.99, 0.00, 49.98);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (9, 5, 2, 1, 12.99, 0.00, 29.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (10, 5, 3, 1, 14.99, 0.00, 19.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (11, 5, 6, 1, 34.99, 37.09, 22.01);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (12, 6, 1, 1, 19.99, 0, 19.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (13, 6, 2, 1, 12.99, 0, 12.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (14, 6, 3, 1, 14.99, 0, 14.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (15, 6, 4, 1, 24.99, 0, 24.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (16, 6, 5, 1, 29.99, 0, 29.99);

INSERT INTO LINEAS\_PEDIDOS (idLineaPedido, idPedido, idProducto, cantidad, precioUnitario, descuento, total) VALUES (17, 6, 6, 1, 34.99, 0, 34.99);

Ejecutar la siguiente instrucción

SELECT 'Clientes' AS tabla, COUNT(\*) AS num\_registros FROM CLIENTES UNION ALL

SELECT 'Categorias', COUNT(\*) FROM CATEGORIAS UNION ALL

SELECT 'Productos', COUNT(\*) FROM PRODUCTOS UNION ALL

SELECT 'Pedidos', COUNT(\*) FROM PEDIDOS UNION ALL

SELECT 'Lineas\_Pedidos', COUNT(\*) FROM LINEAS\_PEDIDOS;

La instrucción tiene que devolver el número de registros que tiene cada tabla, revisar que coincide con la siguiente tabla.

|  |  |
| --- | --- |
| **Tabla** | **Número de registros** |
| Clientes | 5 |
| Categorías | 6 |
| Productos | 6 |
| Pedidos | 8 |
| Lineas\_Pedidos | 20 |

Si no coincide el número de registros, hay que ver en que tabla falta o sobran registros. Repasar la carga de datos.

A continuación, se presentan diferentes consultas que se tienen que realizar sobre los datos insertados, por cada una de ellas pon la sentencia SQL y una captura que se pueda apreciar los datos obtenidos.

1. Obtén una lista de todos los clientes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Encuentra todos los productos con un precio mayor a 30.

Texto

Descripción generada automáticamente

1. Encuentra todos los productos con un precio entre 10 a 20.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Muestra todos los pedidos realizados por el cliente con el id 2.

Texto, Tabla

Descripción generada automáticamente

1. Obtén una lista de todas las categorías ordenadas alfabéticamente.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. Obtén los nombres de los clientes ordenado alfabéticamente de forma descendente [Z-A].

Tabla

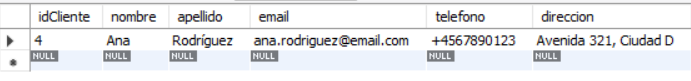
Descripción generada automáticamente con confianza media

1. Encuentra todos los productos cuya descripción contenga la palabra 'Star Wars'.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Encuentra el cliente con el email 'ana.rodriguez@email.com'.



1. Muestra todos los productos que pertenecen a la categoría con el nombre 'Juguetes'.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Obtén el total de pedidos por cada cliente.

Tabla

Descripción generada automáticamente

1. Encuentra el precio promedio de los productos por categoría.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Encuentra el cliente que ha realizado más pedidos.



1. Encuentra el producto más caro en cada categoría.

Texto

Descripción generada automáticamente

1. Obtén el total de ventas por mes y año.

Tabla

Descripción generada automáticamente

1. Muestra el número de productos por categoría y su precio promedio, pero solo para las categorías con 2 o más productos.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. Encuentra el cliente con el mayor gasto total en todos sus pedidos, incluyendo sus datos personales y el total gastado.



1. Encuentra los tres productos más vendidos (asumiendo la cantidad total de cada línea de pedidos).

Interfaz de usuario gráfica, Texto, Aplicación

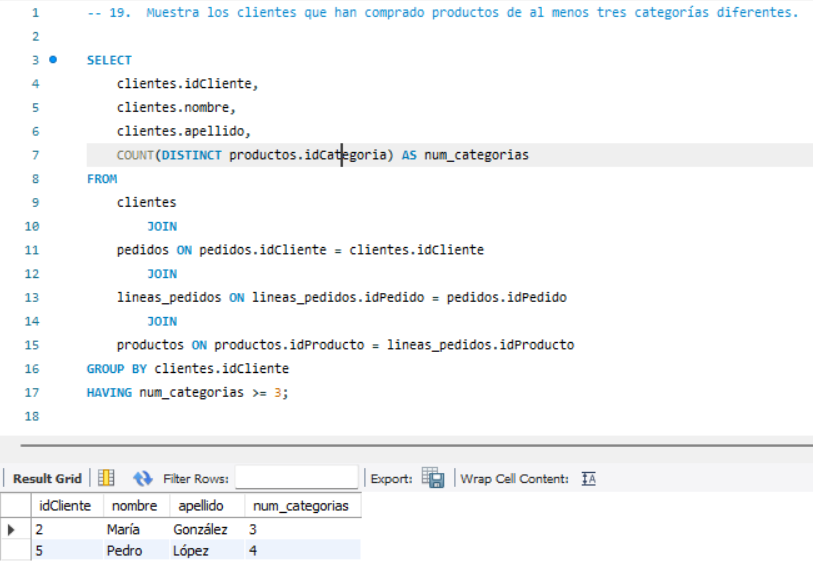
Descripción generada automáticamente

1. Calcula el total gastado en cada categoría de productos por todos los clientes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Muestra los clientes que han comprado productos de al menos tres categorías diferentes.



1. Encuentra el mes con las mayores ventas totales. Y el mes con las menores ventas totales. Tiene que aparecer cada información en una fila diferente.

(SELECT

MONTH(fecha) AS month,

YEAR(fecha) AS year,

SUM(total) AS total

FROM

pedidos

WHERE

total = (SELECT

MAX(total)

FROM

pedidos)

GROUP BY month , year , total

LIMIT 1) UNION (SELECT

MONTH(fecha) AS month,

YEAR(fecha) AS year,

SUM(total) AS total

FROM

pedidos

WHERE

total = (SELECT

MIN(total)

FROM

pedidos)

GROUP BY month , year , total

LIMIT 1);

Tabla

Descripción generada automáticamente