**9.1 TALLER – Lenguaje de Consulta Relacional - DML**Imagen que contiene Texto

Descripción generada automáticamente

Apellido y Nombre del estudiante: Andres Camilo Barco Roa

**Nota: cargar el archivo en formato word, recordar la estructura para guardar el archivo mencionada en clase.**

Cada estudiante deberá realizar en el siguiente taller:

1. Los Querys debe ir copiados y resaltados en un color.
2. Los resultados de las consultas(query) debes ser pantallazos resaltando el resultado.
3. Los temas de la clase se aplicarán con el siguiente escenario que se ha venido trabajando:

Una tienda de electrónica llamada "TechStore" gestiona su inventario de productos en una base de datos relacional.

Para ello, utilizan DML (Data Manipulation Language) para insertar, modificar, eliminar y consultar información sobre los productos disponibles con los otros comandos.

**Situación**

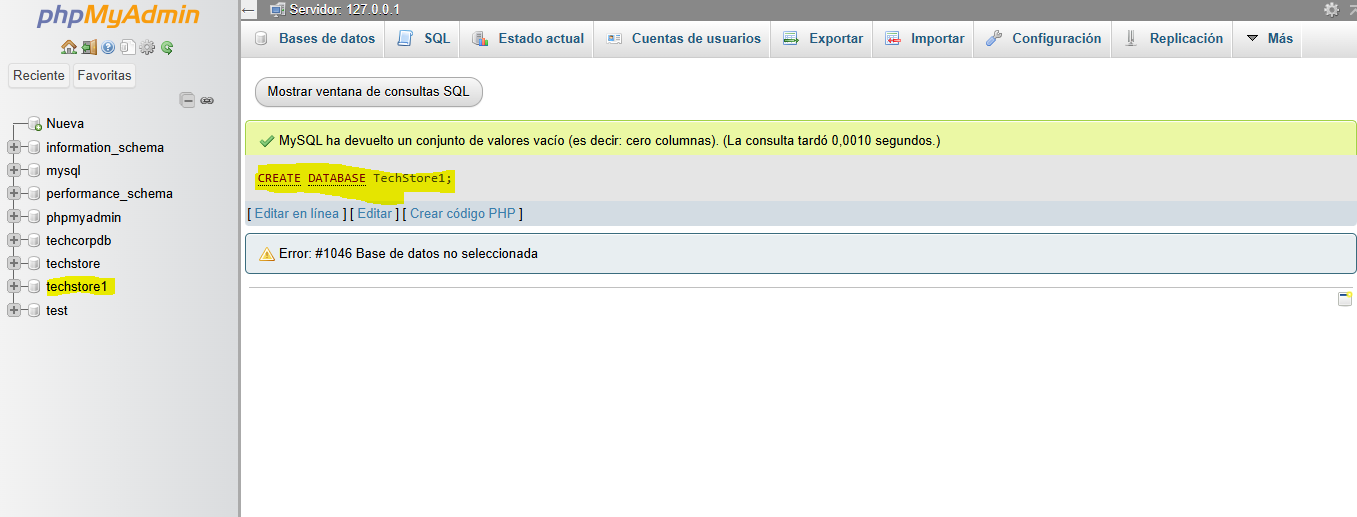
El gerente de la tienda quiere realizar las siguientes acciones en la base de datos:

1. Conocer los productos iniciales.
2. Agregar 20 nuevos productos al inventario.
3. Consultas usando DML:
4. Ajustar el precio del iPhone 13 y el stock de la tarjeta gráfica GTX 1650.
5. Se requiere ver los 10 productos más baratos primero.
6. Se requiere saber cuántos productos tienen el mismo stock disponible.
7. Mostrar solo los productos con stock mayor a 10 y contar cuántos hay.
8. Se requiere ver solo los primeros 5 productos en la lista.
9. Se requiere saber qué 12 valores de stock hay en la base de datos sin repetirlos.
10. Se dejó de vender el producto de elección (Altavoz Inteligente Alexa Echo Dot) y lo eliminamos de la tabla.
11. Exporte la Base de datos con el nombre de productos.
12. Realice los Joins vistos en clase.
13. Borra todos los registros de la tabla, pero conserva la estructura.
14. Borrar la base de datos.

**SOLUCIÓN**

**Aplicación de DML**

1. Crear base de datos TechStore1.

**CREATE DATABASE TechStore1;** 

1. Crear tabla productos.

**CREATE TABLE productos (**

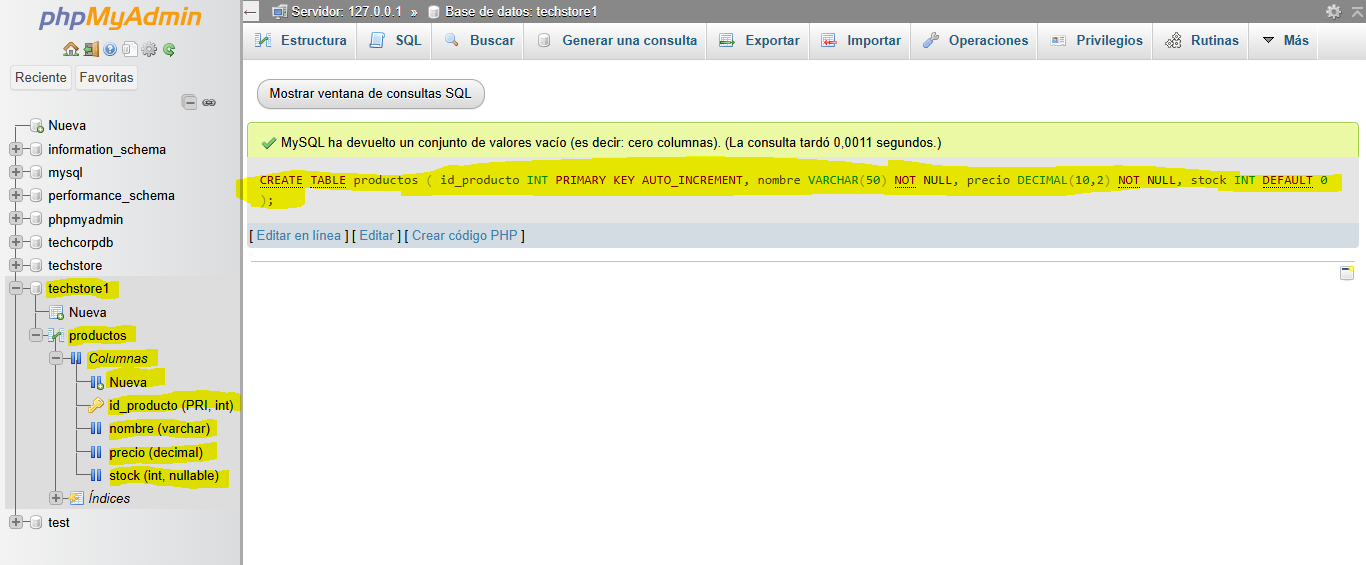
**id\_producto INT PRIMARY KEY AUTO\_INCREMENT,**

**nombre VARCHAR(50) NOT NULL,**

**precio DECIMAL(10,2) NOT NULL,**

**stock INT DEFAULT 0**

**);**



1. Ingresar 20 productos.

**INSERT INTO productos (nombre, precio, stock)**

**VALUES**

**('Laptop Dell XPS 13', 5000000, 20),**

**('Tablet iPad Pro', 3000000, 10),**

**('iPhone 13', 10000000, 50),**

**('Tarjeta gráfica GTX 1650', 2500000, 15),**

**('Teclado mecánico Logitech Pro X', 350000, 40),**

**('Mouse inalámbrico Logitech G305', 120000, 35),**

**('Memoria USB 64 GB', 60000, 60),**

**('Cámara de seguridad Blink', 500000, 12),**

**('Altavoz Inteligente Alexa Echo Dot', 200000, 22),**

**('Monitor Gaming LG 24"', 700000, 10),**

**('Disco duro externo Seagate Backup Plus', 320000, 15),**

**('Consola de videojuegos PS5', 2200000, 5),**

**('Auriculares inalámbricos Sony WH-1000XM4', 1500000, 8),**

**('iPad Air', 2800000, 6),**

**('Smartwatch Samsung Galaxy Watch', 900000, 9),**

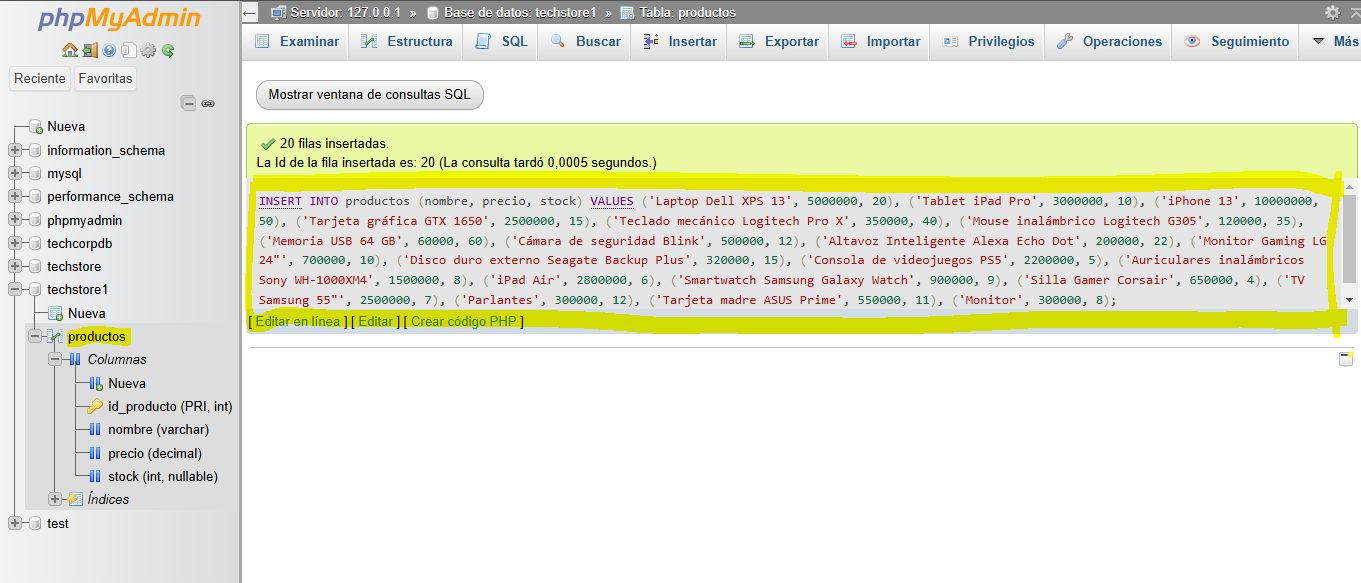
**('Silla Gamer Corsair', 650000, 4),**

**('TV Samsung 55"', 2500000, 7),**

**('Parlantes', 300000, 12),**

**('Tarjeta madre ASUS Prime', 550000, 11),**

**('Monitor', 300000, 8);**



1. Pantallazo de los productos iniciales.

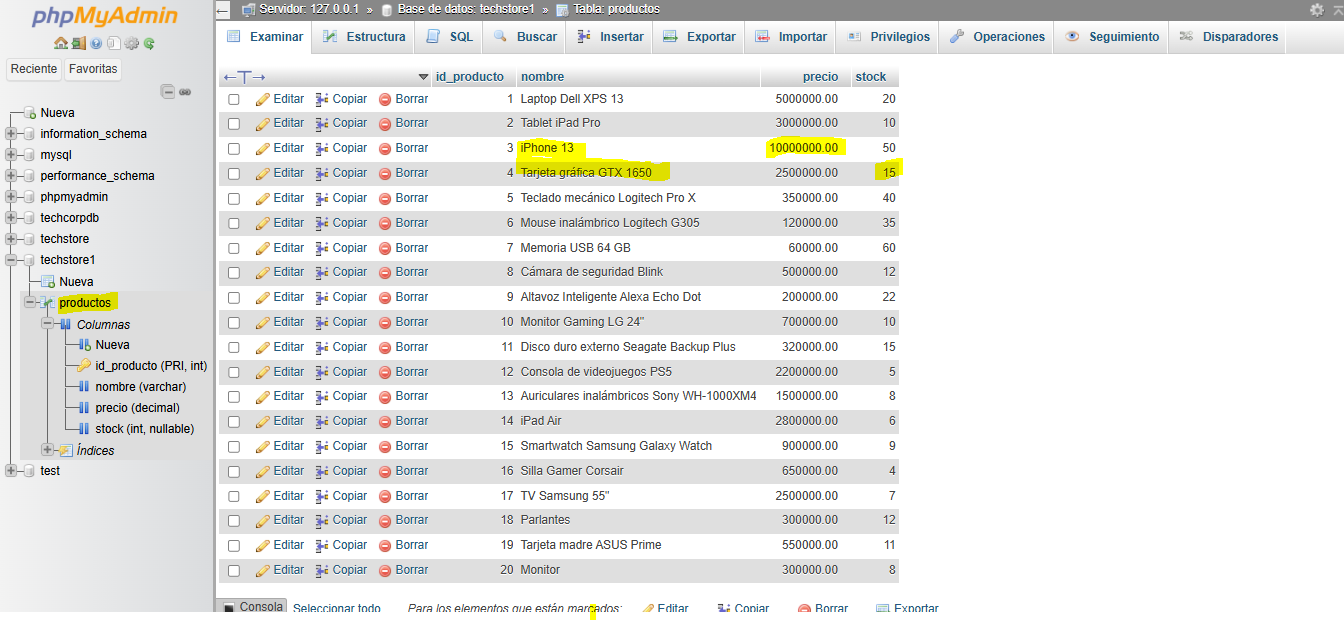
**SELECT \* FROM productos;**



5. Consultas usando DML:

1. Ajustar el precio del iPhone 13 y el stock de la tarjeta gráfica GTX 1650.

Antes:



Despues:

**UPDATE productos**

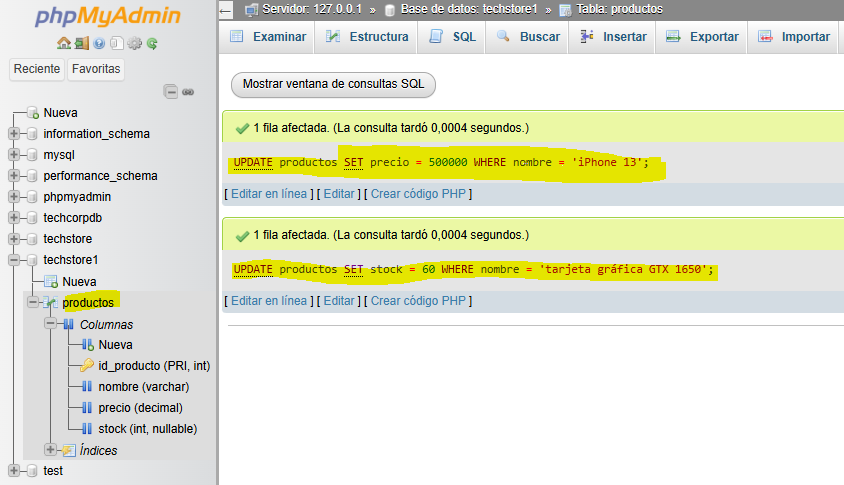
**SET precio = 500000**

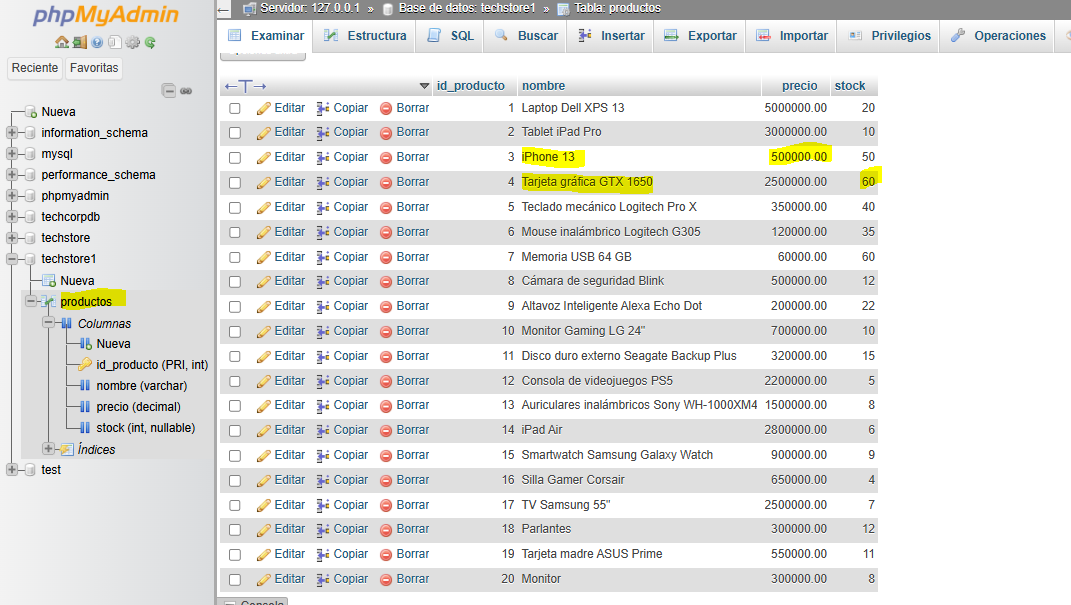
**WHERE nombre = 'iPhone 13';**

**UPDATE productos**

**SET stock = 60**

**WHERE nombre = 'tarjeta gráfica GTX 1650'**





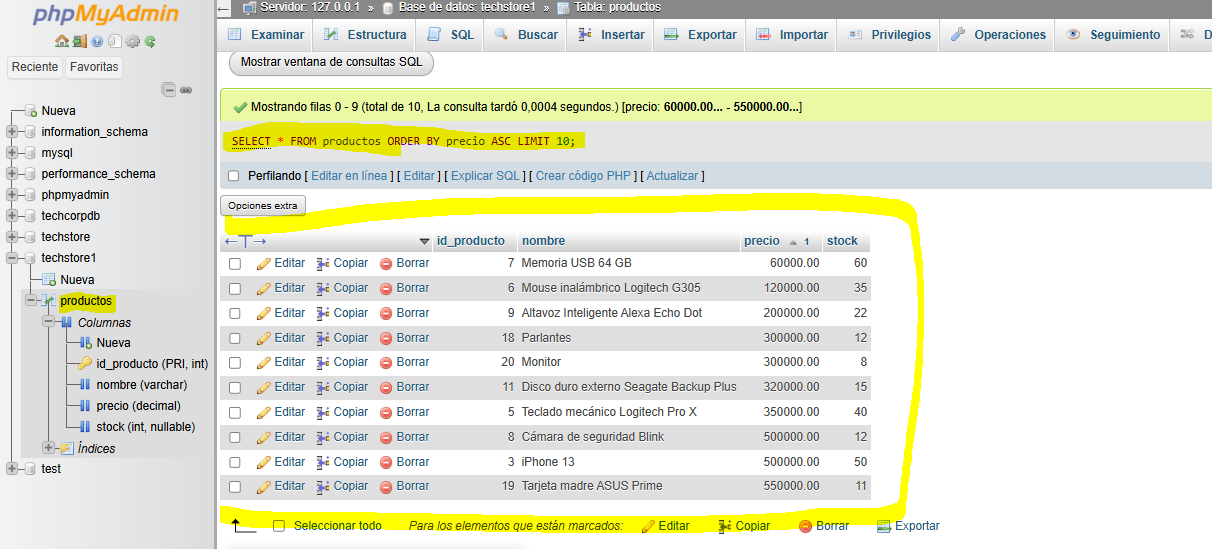
1. Se requiere ver los 10 productos más baratos primero.

**SELECT \***

**FROM productos**

**ORDER BY precio ASC**

**LIMIT 10;**

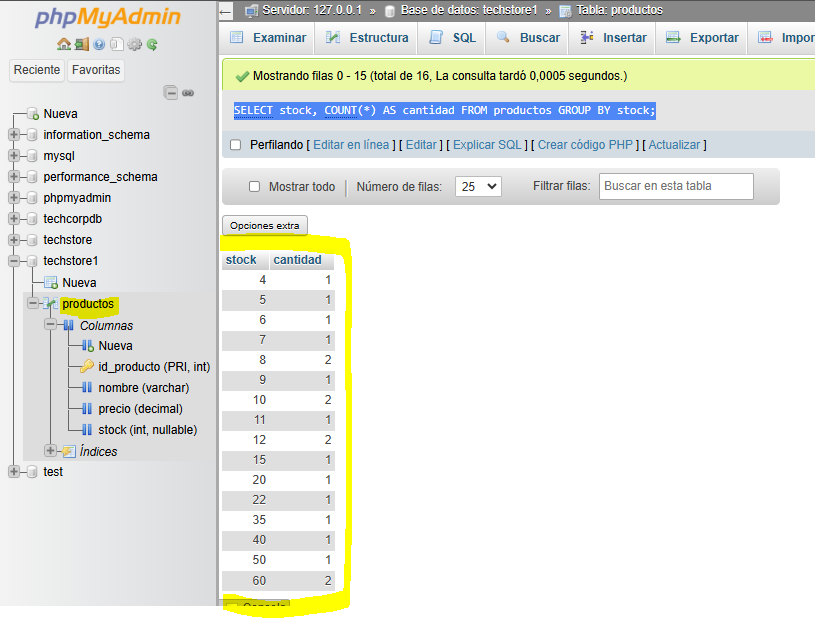


1. Se requiere saber cuántos productos tienen el mismo stock disponible.

**SELECT stock, COUNT(\*) AS cantidad**

**FROM productos**

**GROUP BY stock;**



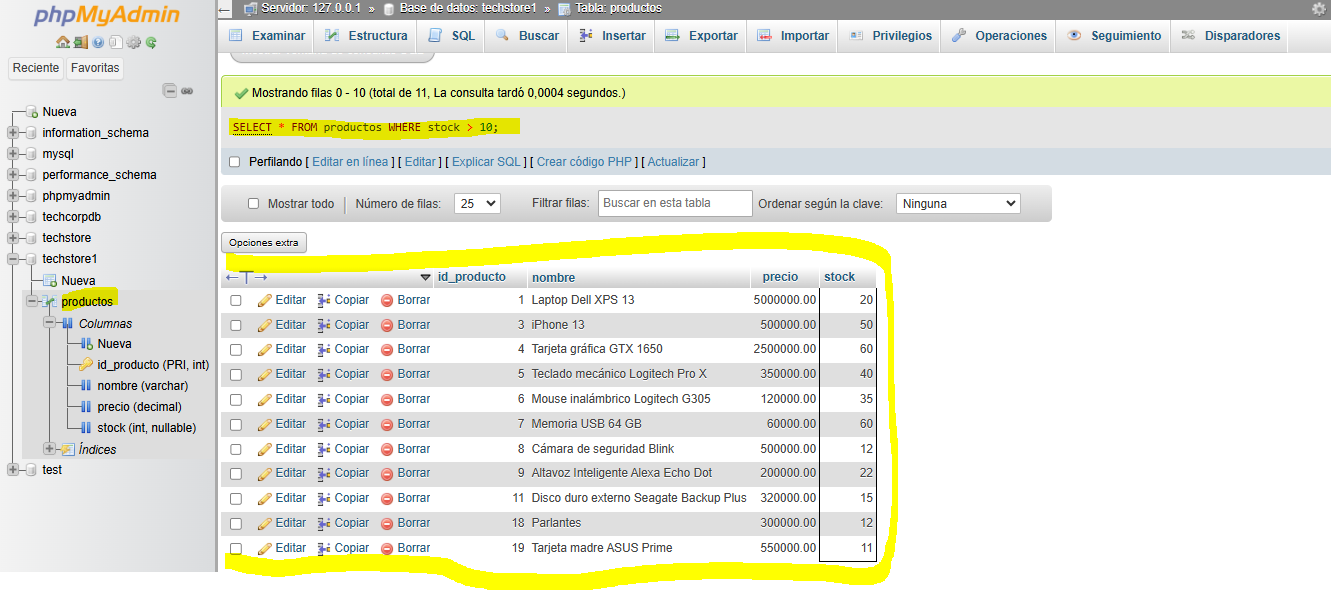
1. Mostrar solo los productos con stock mayor a 10 y contar cuántos hay.

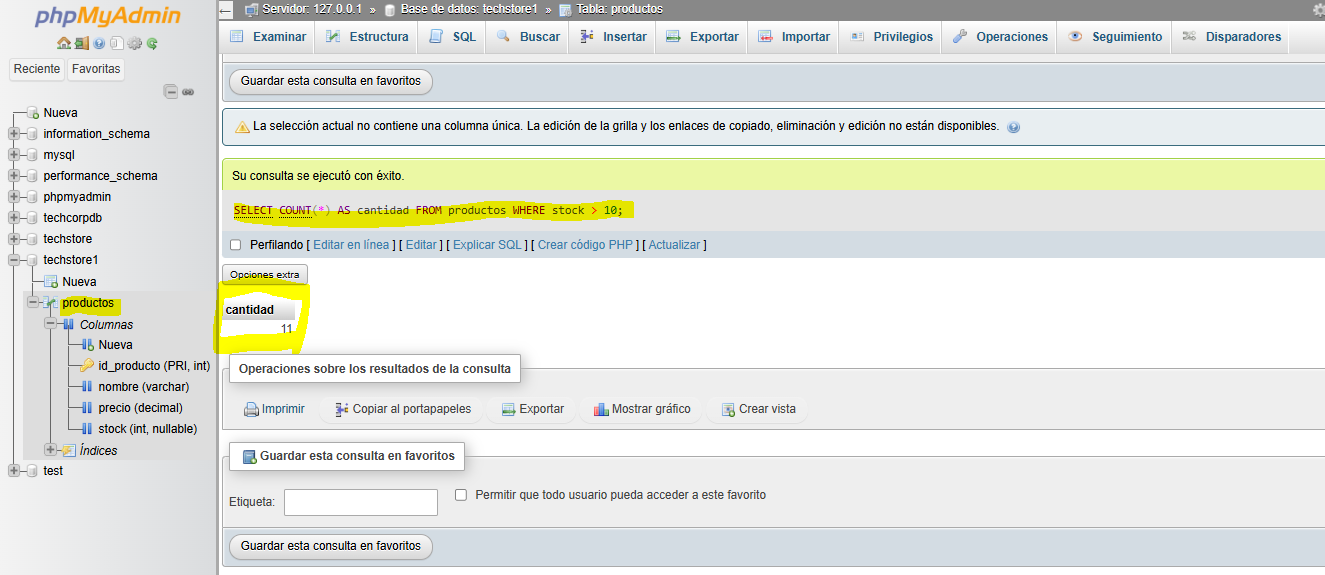
**SELECT \* FROM productos**

**WHERE stock > 10;**

**SELECT COUNT(\*) AS cantidad FROM productos**

**WHERE stock > 10;**

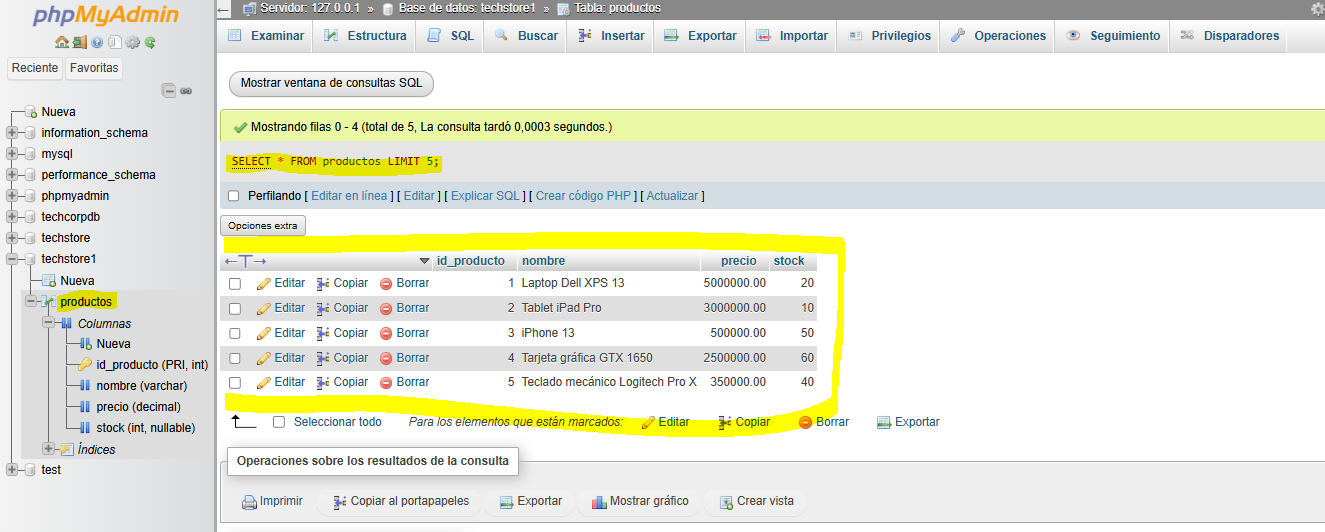




1. Se requiere ver solo los primeros 5 productos en la lista.

**SELECT \* FROM productos**

**LIMIT 5;**

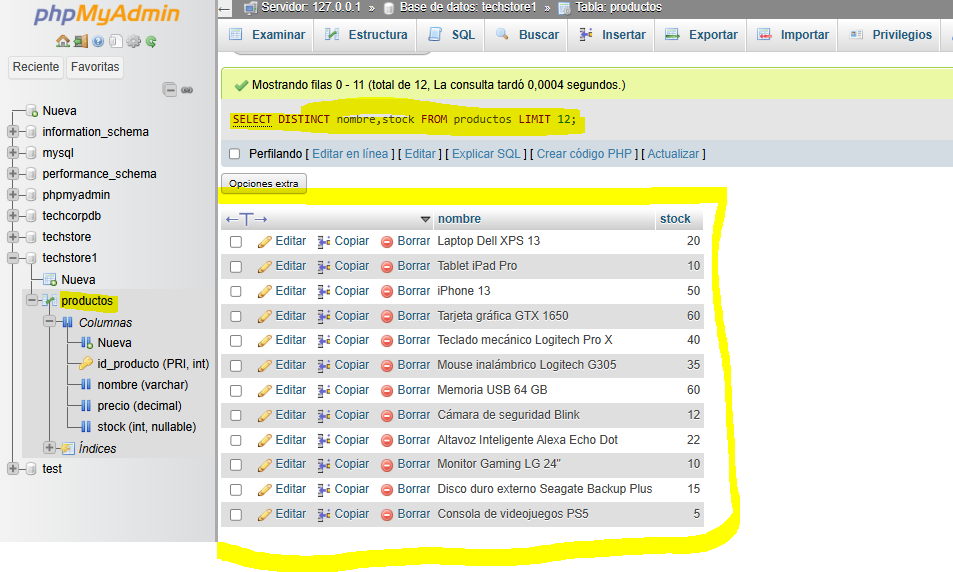


1. Se requiere saber qué 12 valores de stock hay en la base de datos sin repetirlos.

**SELECT DISTINCT nombre,stock**

**FROM productos**

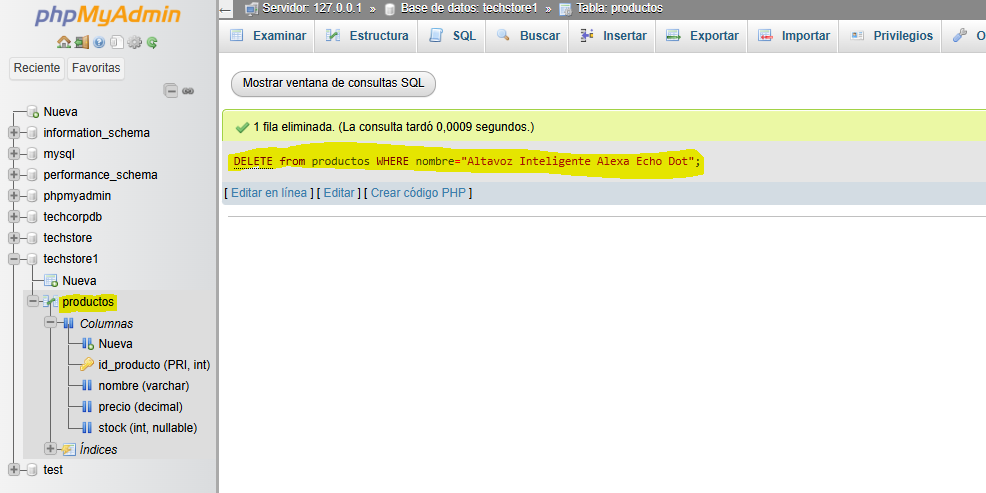
**LIMIT 12;**

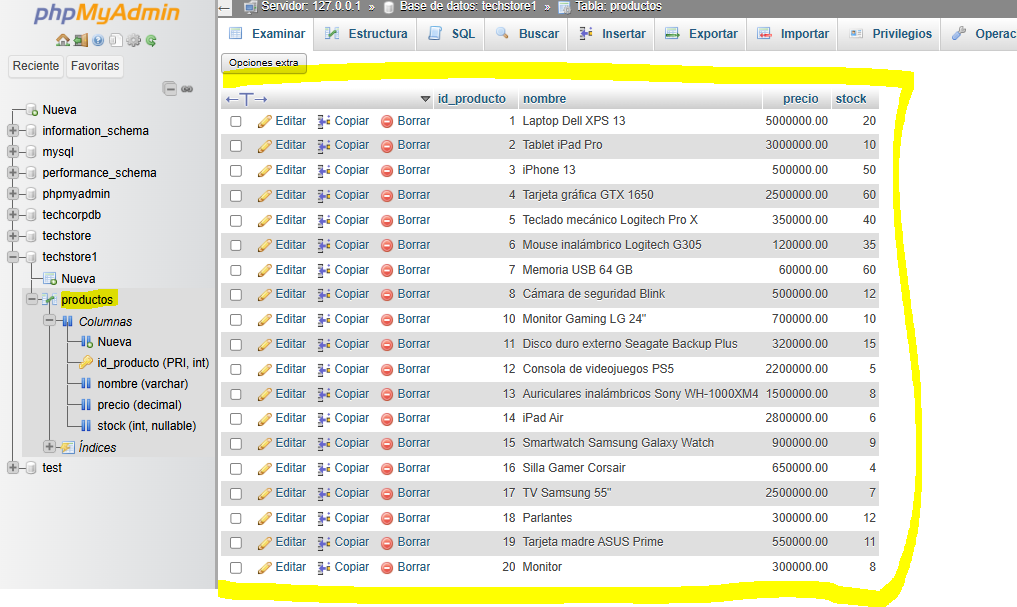


1. Se dejó de vender el producto de elección (Altavoz Inteligente Alexa Echo Dot) y lo eliminamos de la tabla.

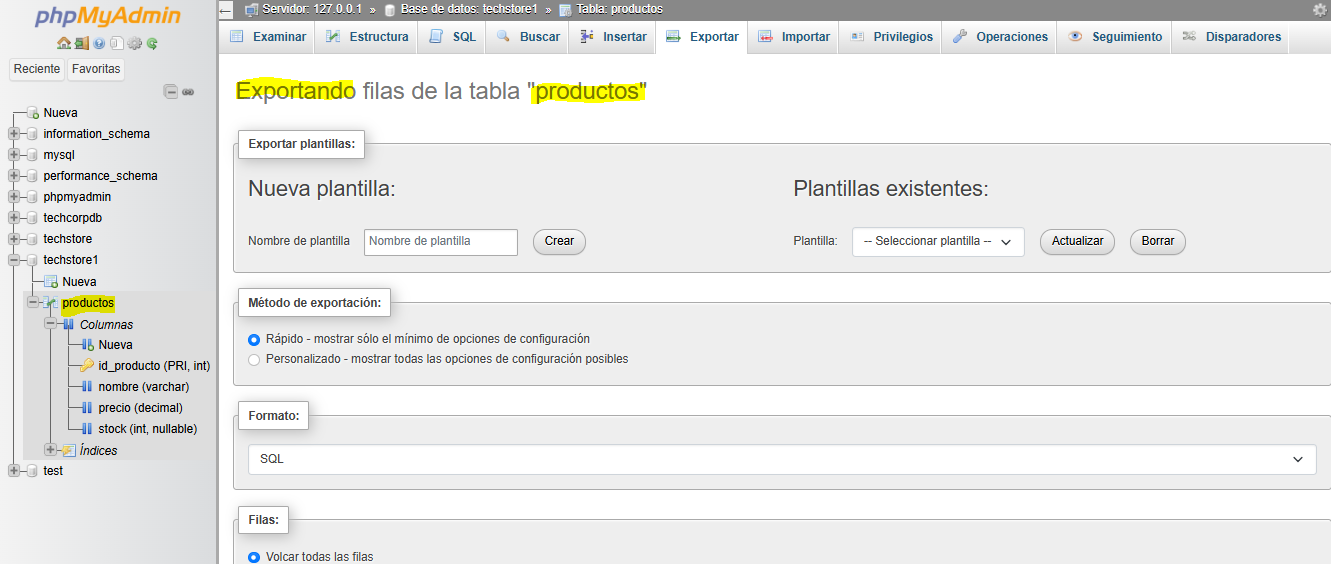
DELETE from productos

WHERE nombre="Altavoz Inteligente Alexa Echo Dot";





1. Exporte la Base de datos con el nombre de productos.



1. Realice los Joins vistos en clase.
2. hacer otra tabla que utilice llaves foráneas para realizar join y ingresar 19 registros.

**CREATE TABLE Ventas (**

**id\_venta INT PRIMARY KEY AUTO\_INCREMENT,**

**id\_producto INT,**

**cantidad INT,**

**fecha DATE,**

**FOREIGN KEY (id\_producto) REFERENCES productos(id\_producto)**

**);**

**INSERT INTO Ventas (id\_producto, cantidad, fecha)**

**VALUES**

**(1, 2, '2025-01-10'),**

**(2, 1, '2025-01-11'),**

**(3, 3, '2025-01-12'),**

**(4, 2, '2025-01-13'),**

**(5, 1, '2025-01-14'),**

**(6, 2, '2025-01-15'),**

**(7, 4, '2025-01-16'),**

**(8, 2, '2025-01-17'),**

**(10, 3, '2025-01-19'),**

**(11, 2, '2025-01-20'),**

**(12, 1, '2025-01-21'),**

**(13, 3, '2025-01-22'),**

**(14, 2, '2025-01-23'),**

**(15, 1, '2025-01-24'),**

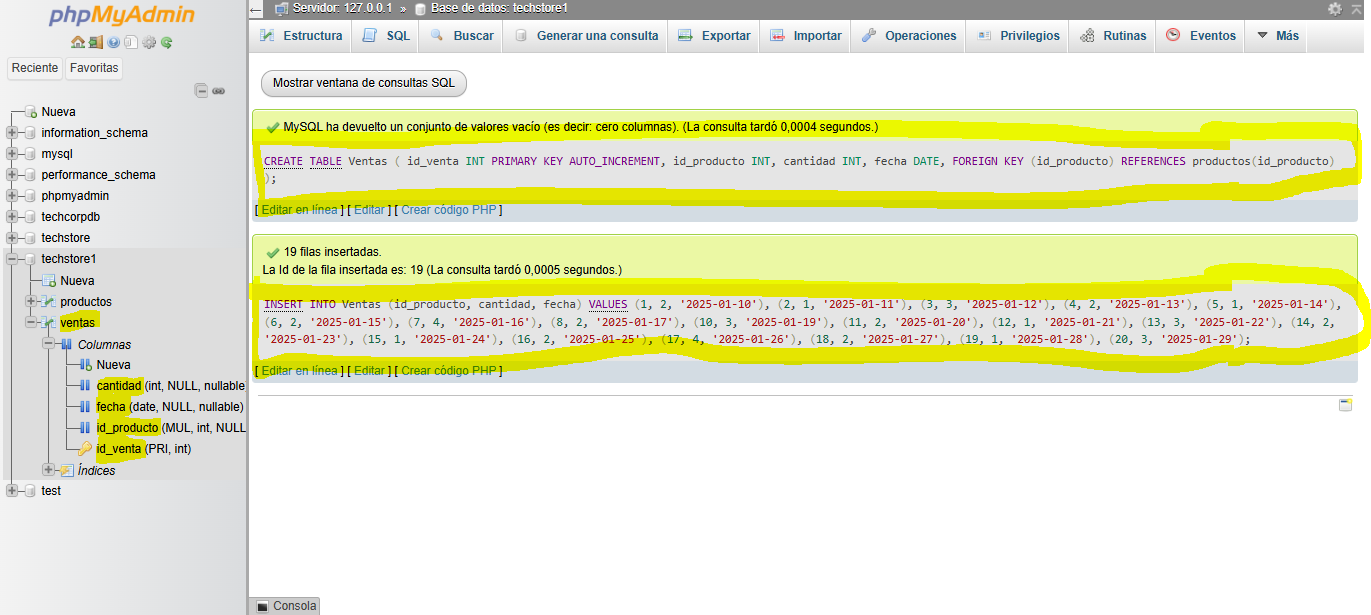
**(16, 2, '2025-01-25'),**

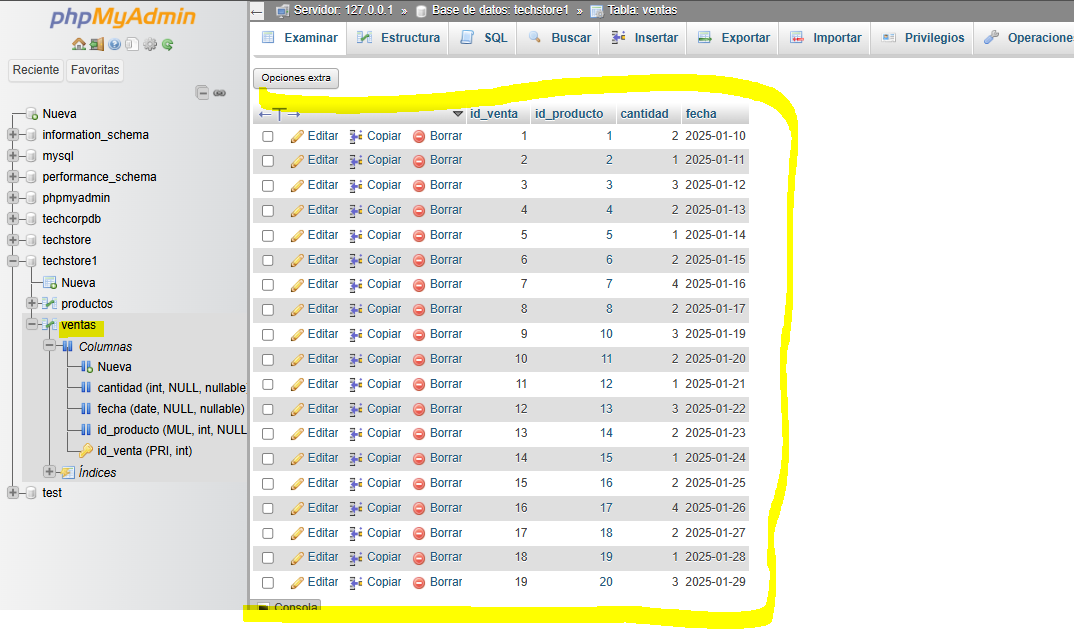
**(17, 4, '2025-01-26'),**

**(18, 2, '2025-01-27'),**

**(19, 1, '2025-01-28'),**

**(20, 3, '2025-01-29');**



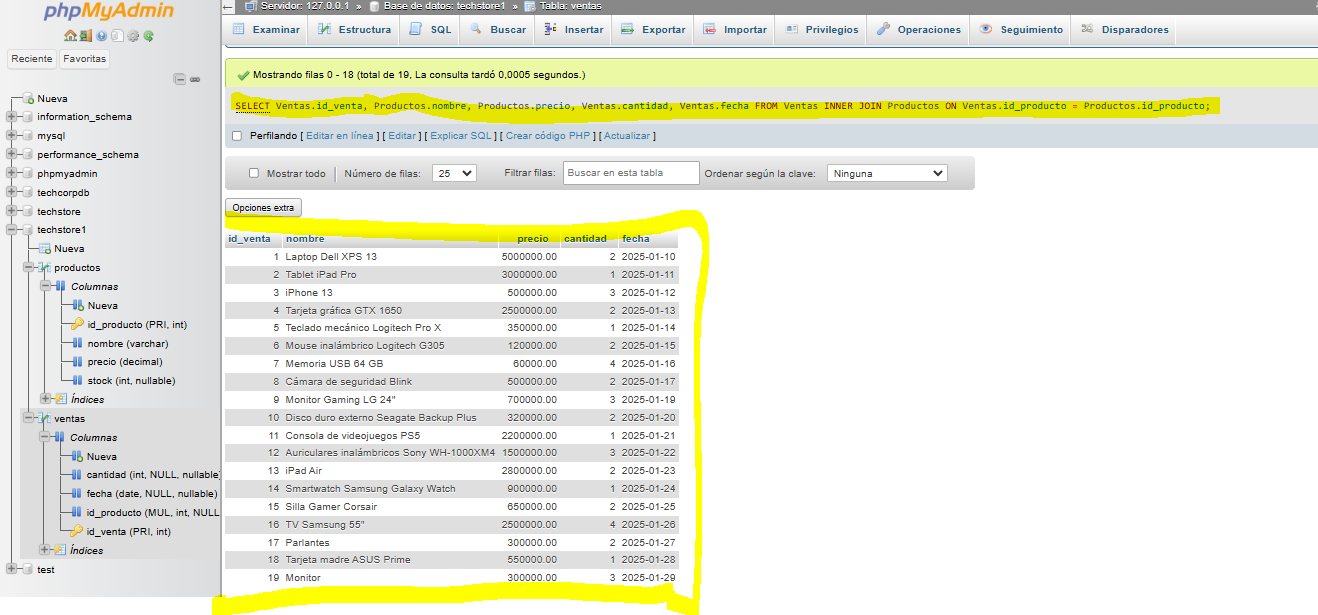


1. **INNER JOIN**

**SELECT Ventas.id\_venta, Productos.nombre, Productos.precio, Ventas.cantidad, Ventas.fecha**

**FROM Ventas**

**INNER JOIN Productos ON Ventas.id\_producto = Productos.id\_producto;**

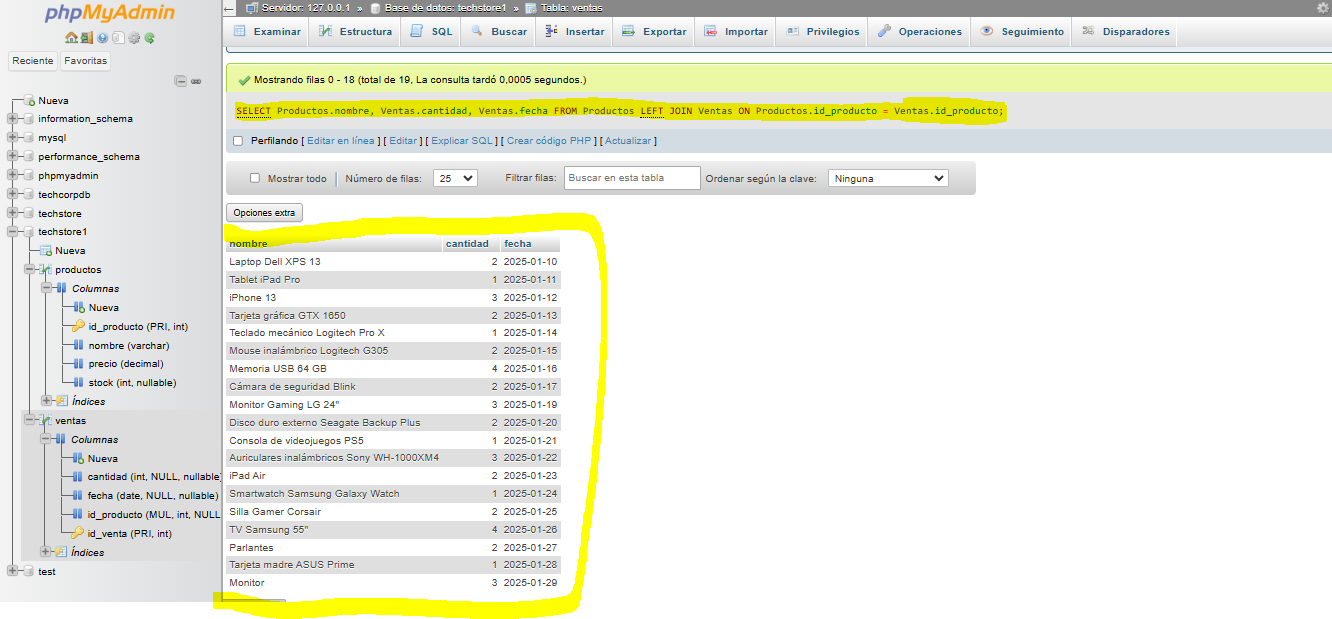


1. **LEFT JOIN**

**SELECT Productos.nombre, Ventas.cantidad, Ventas.fecha**

**FROM Productos**

**LEFT JOIN Ventas ON Productos.id\_producto = Ventas.id\_producto;**

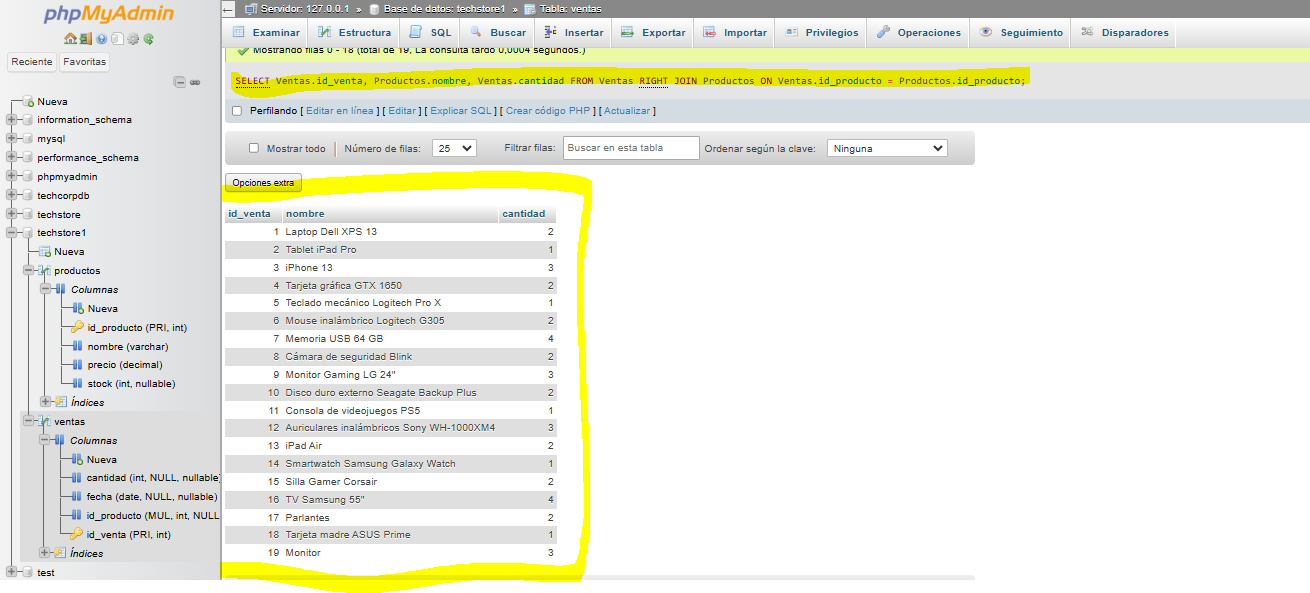


1. **RIGHT JOIN**

**SELECT Ventas.id\_venta, Productos.nombre, Ventas.cantidad**

**FROM Ventas**

**RIGHT JOIN Productos ON Ventas.id\_producto = Productos.id\_producto;**



1. **FULL OUTER JOIN**

**SELECT Productos.nombre, Ventas.cantidad, Ventas.fecha**

**FROM Productos**

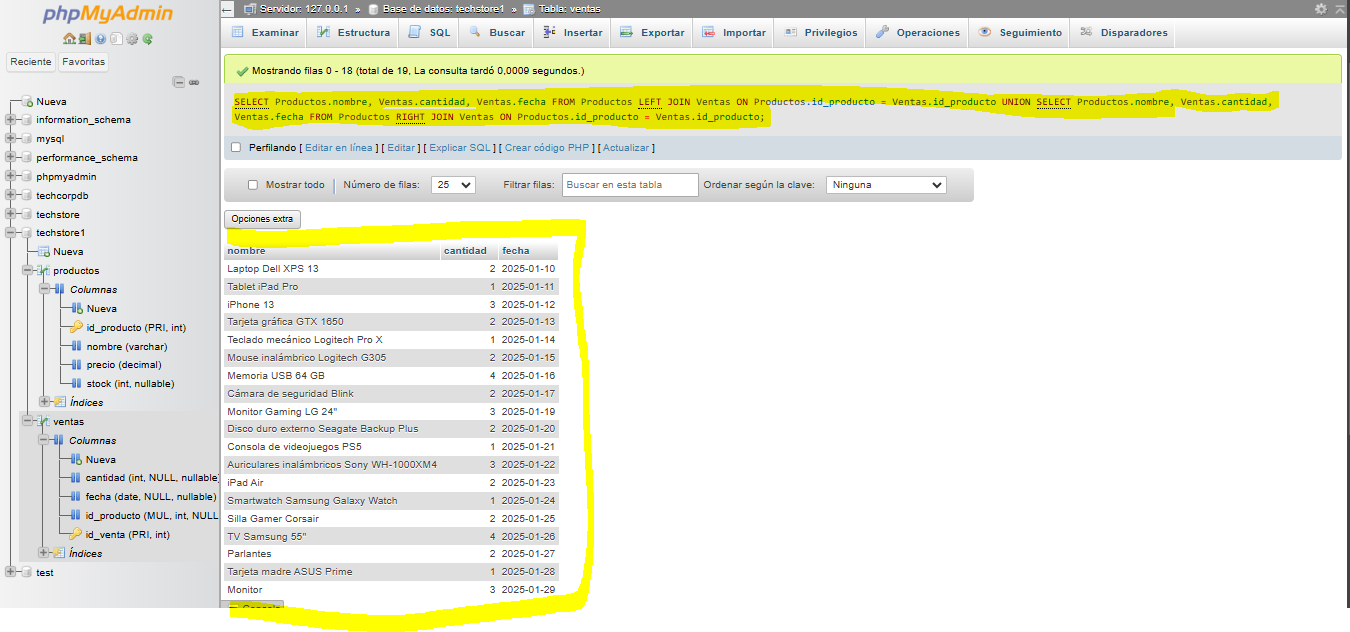
**LEFT JOIN Ventas ON Productos.id\_producto = Ventas.id\_producto**

**UNION**

**SELECT Productos.nombre, Ventas.cantidad, Ventas.fecha**

**FROM Productos**

**RIGHT JOIN Ventas ON Productos.id\_producto = Ventas.id\_producto;**

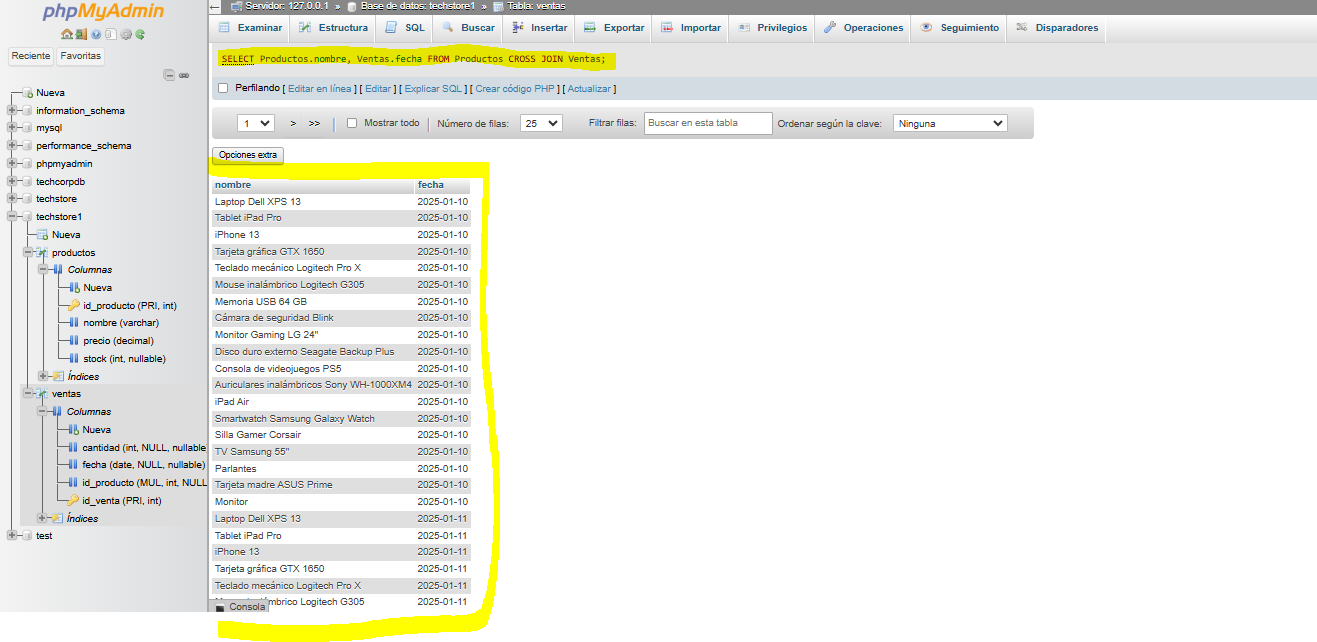


1. **CROSS JOIN**

**SELECT Productos.nombre, Ventas.fecha**

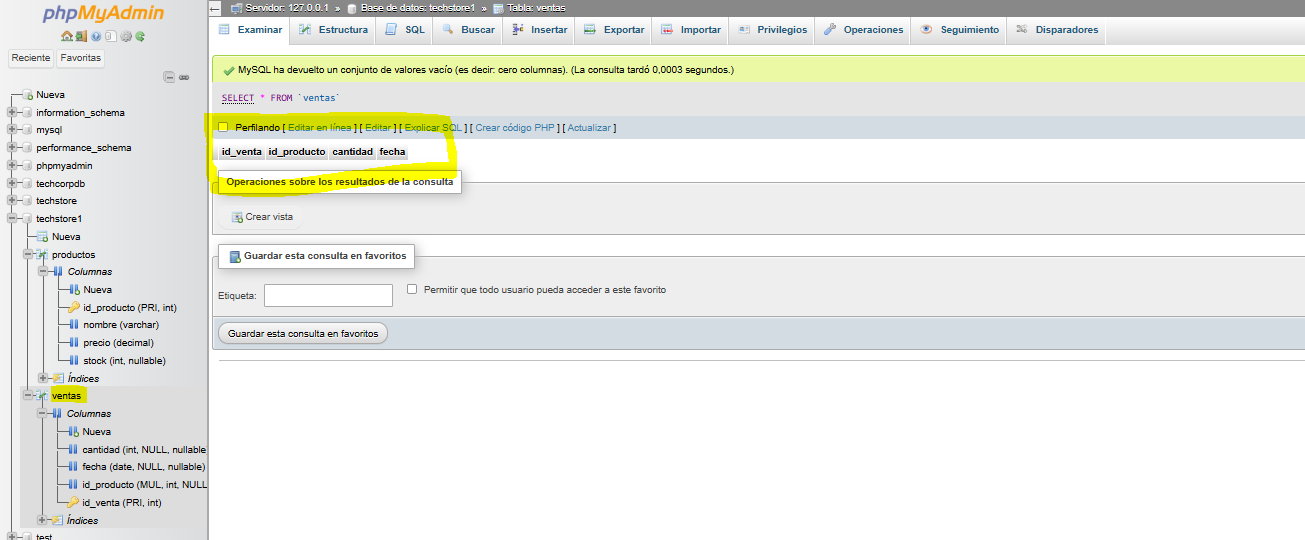
**FROM Productos**

**CROSS JOIN Ventas;**

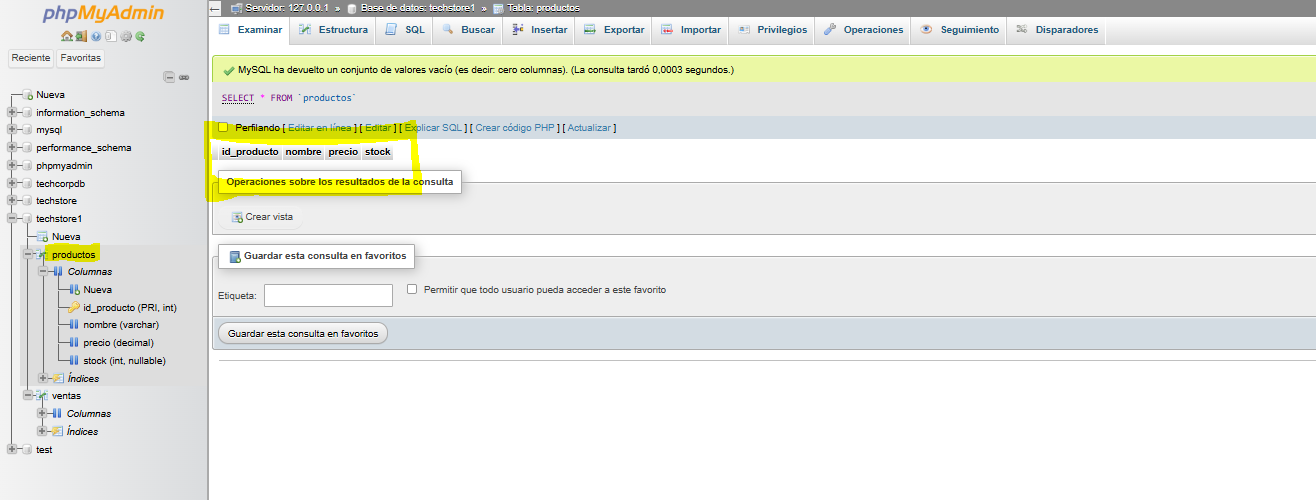


1. Borra todos los registros de la tabla, pero conserva la estructura.

**TRUNCATE TABLE ventas;**



**TRUNCATE TABLE productos;**



1. Borrar la base de datos.

**drop DATABASE techstore1;**

