**Imagen que contiene Texto

Descripción generada automáticamente9.2 TALLER – Lenguaje de Consulta Relacional - DML**

Estudiante: Andres Camilo Barco Roa

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.

**Situación**

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

Conocer los productos iniciales.

1. Conocer los productos iniciales.
2. Agregar 20 nuevos productos al inventario.
3. Consultas usando DML:
4. Agregar 5 nuevos productos al inventario: Televisor Samsung 55, valor 2.500.000 y cantidades 15; Teclado Mecánico RGB, valor 2.000.000 y cantidades 30, agregar los faltantes.
5. Actualizar el precio de un producto (Tv Samsung) porque ha recibido un descuento especial (2.200.000).
6. Se requiere ver los 10 productos más baratos primero.
7. Se requiere saber cuántos productos tienen el mismo stock disponible.
8. Mostrar solo los productos con stock mayor a 10 y contar cuántos hay.
9. Se requiere ver solo los primeros 5 productos en la lista.
10. Se requiere saber qué 12 valores de stock hay en la base de datos sin repetirlos.
11. Se dejó de vender el producto de elección (Altavoz Inteligente Alexa Echo Dot) y lo eliminamos de la tabla.
12. Exporte la Base de datos con el nombre de productos.
13. Realice los Joins vistos en clase.
14. Guardar la BD.

**SOLUCIÓN**

1. Importar y Conocer los productos iniciales.



1. Consultas usando DML:
2. Agregar 5 nuevos productos al inventario: Televisor Samsung 55, valor 2.500.000 y cantidades 15; Teclado Mecánico RGB, valor 2.000.000 y cantidades 30, agregar los faltantes.

INSERT INTO productos (nombre, precio, stock)

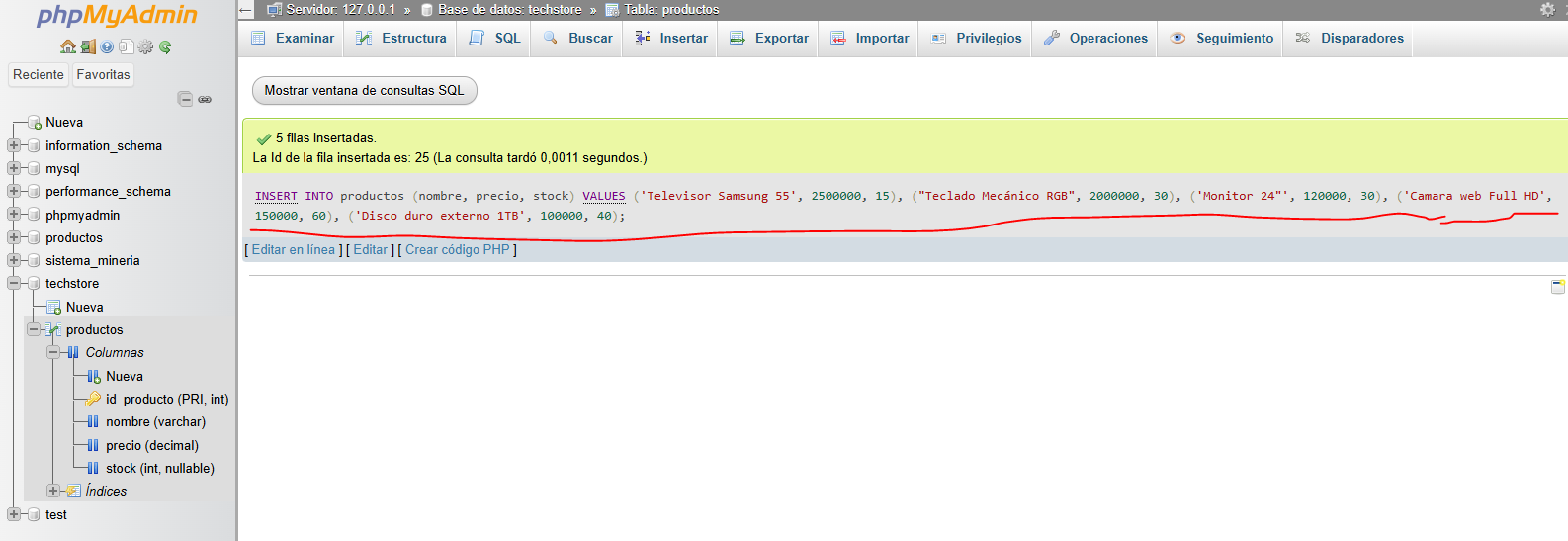
VALUES ('Televisor Samsung 55', 2500000, 15),

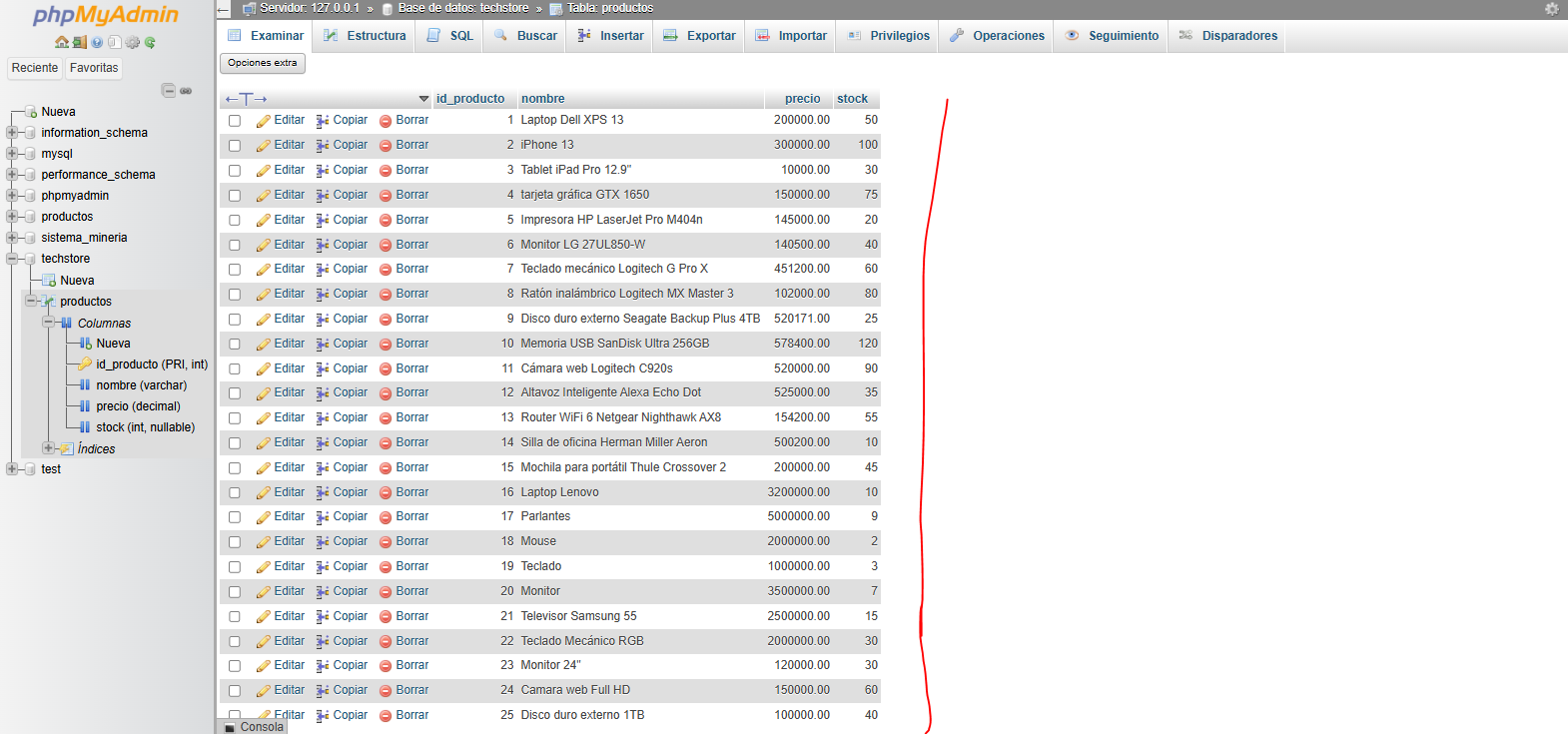
(“Teclado Mecánico RGB”, 2000000, 30),

('Monitor 24"', 120000, 30),

('Camara web Full HD', 150000, 60),

('Disco duro externo 1TB', 100000, 40);



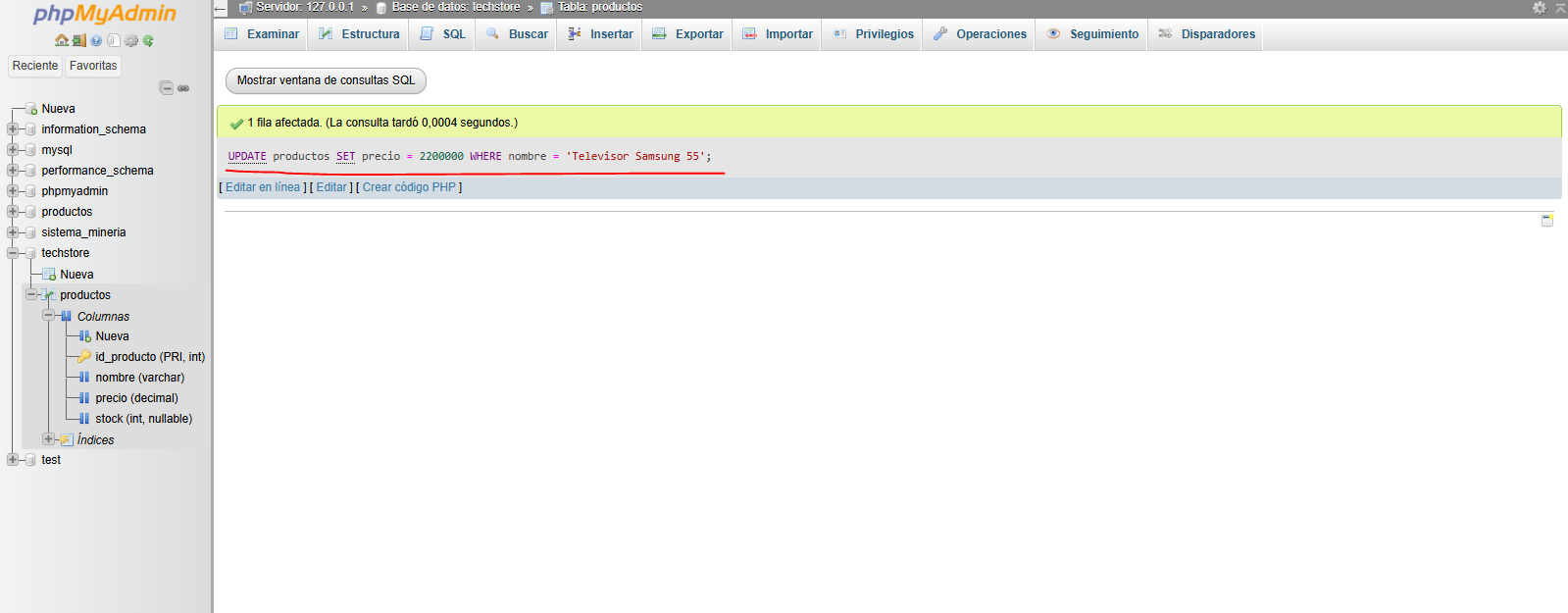


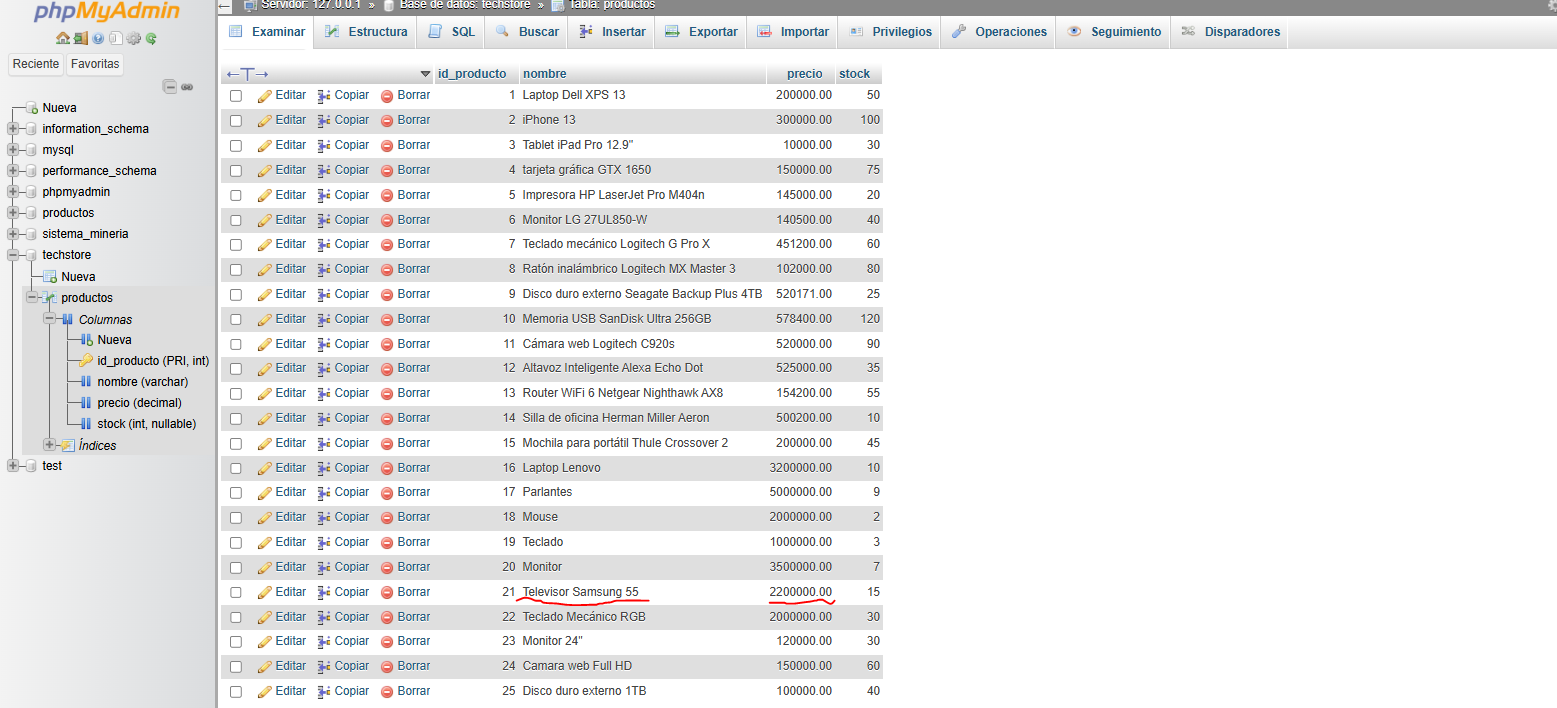
1. Actualizar el precio de un producto (Tv Samsung) porque ha recibido un descuento especial (2.200.000).

UPDATE productos

SET precio = 2200000

WHERE nombre = 'Televisor Samsung 55'



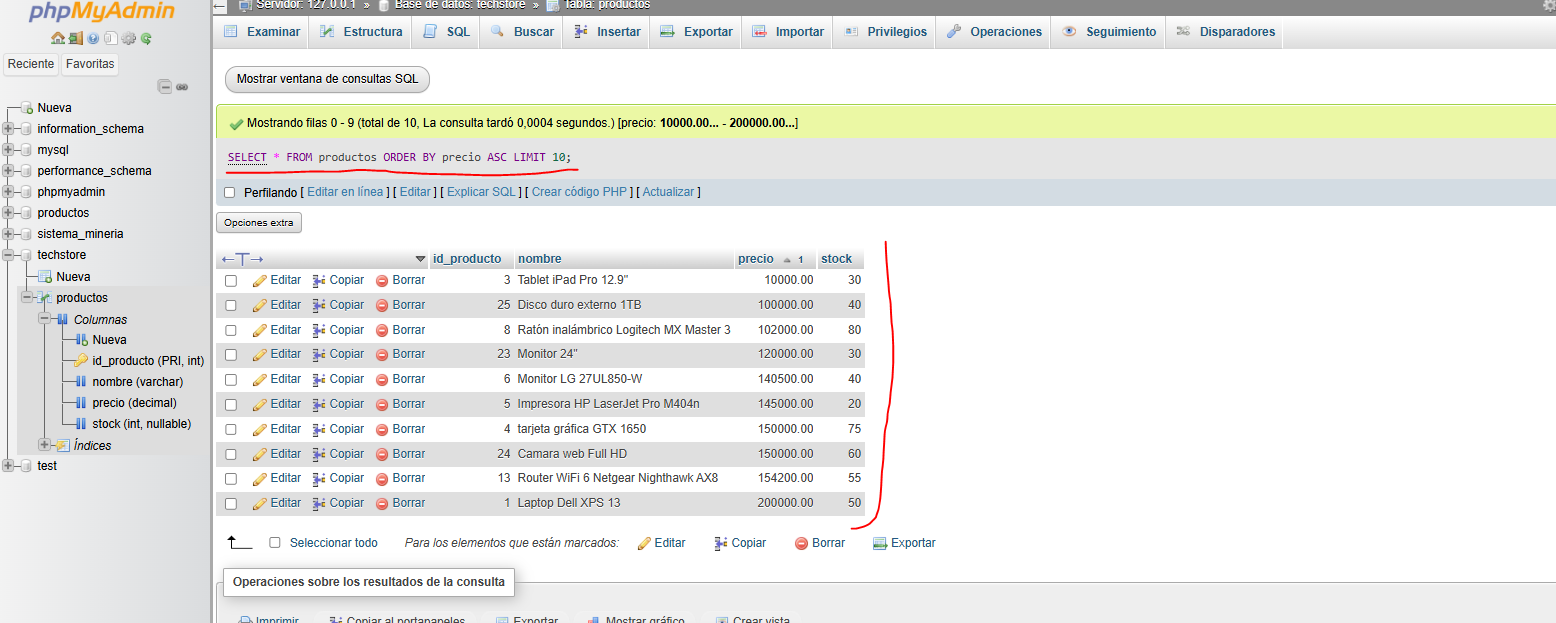


1. Se requiere ver los 10 productos más baratos primero.

SELECT \* FROM productos

ORDER BY precio ASC

LIMIT 10;



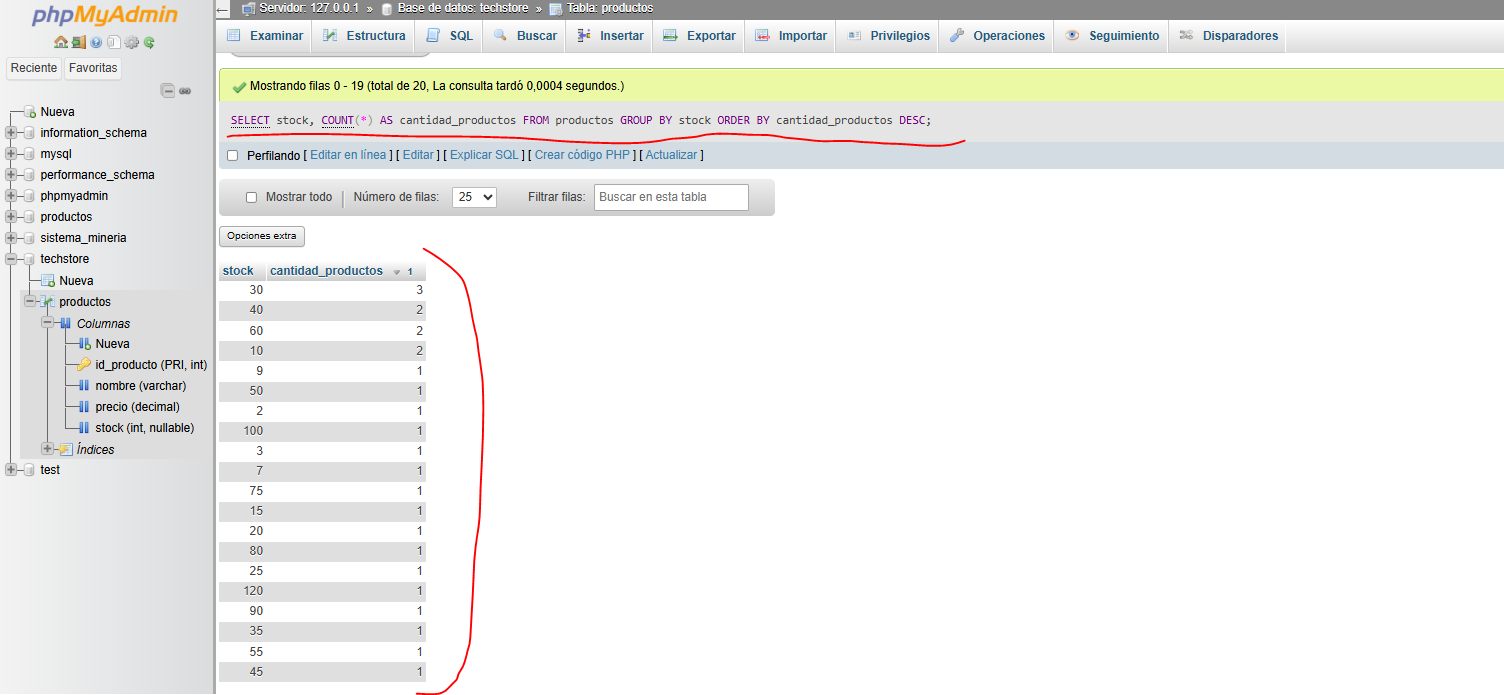
1. Se requiere saber cuántos productos tienen el mismo stock disponible.

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

FROM productos

GROUP BY stock

ORDER BY cantidad\_productos DESC;

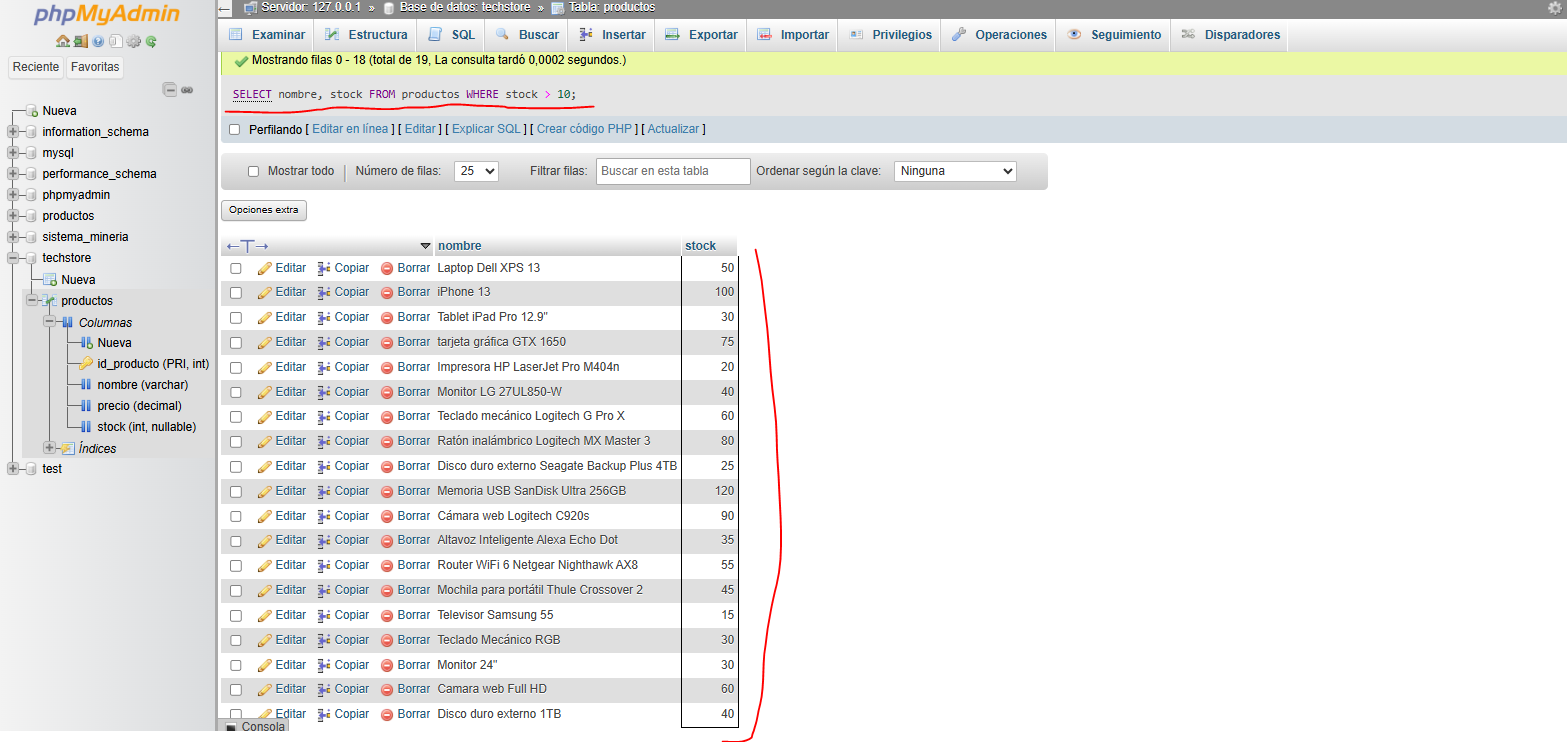


1. Mostrar solo los productos con stock mayor a 10 y contar cuántos hay.

SELECT nombre, stock

FROM productos

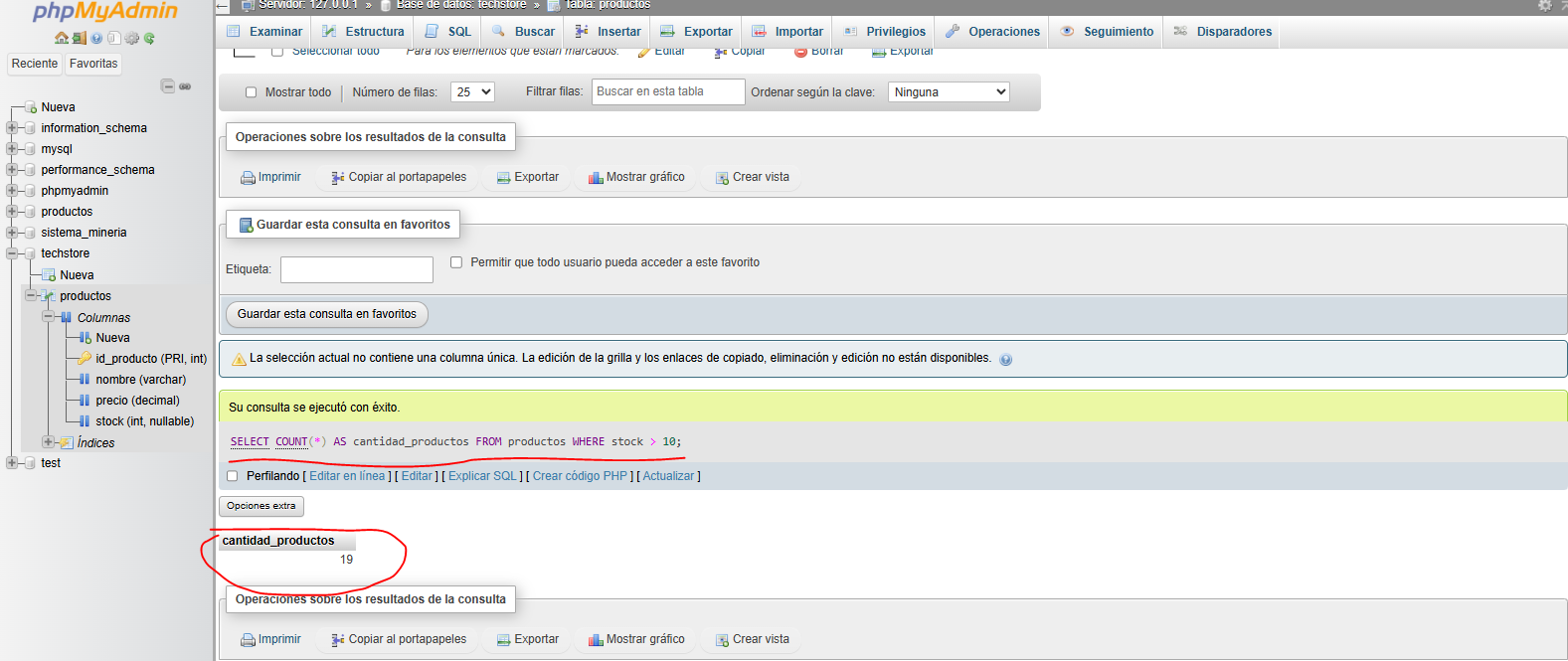
WHERE stock > 10;



SELECT COUNT(\*) AS cantidad\_productos

FROM productos

WHERE stock > 10;

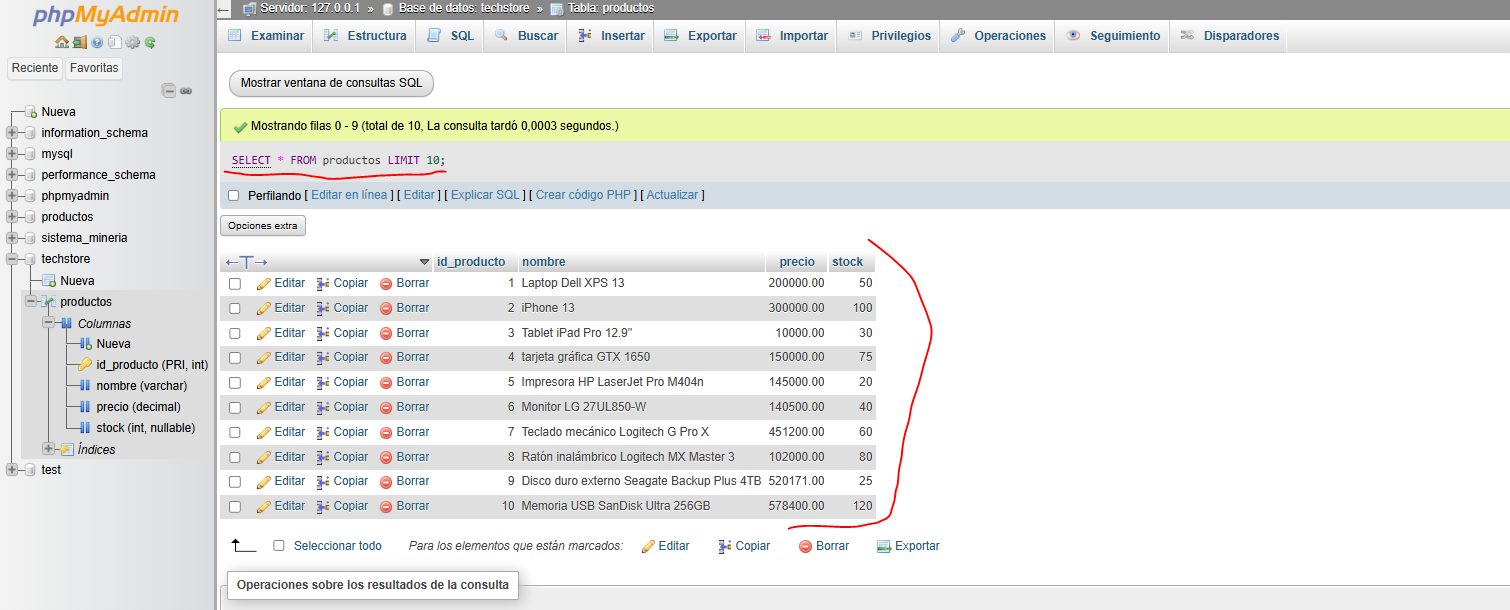


1. Se requiere ver solo los primeros 5 productos en la lista.



SELECT \* FROM productos

LIMIT 10;

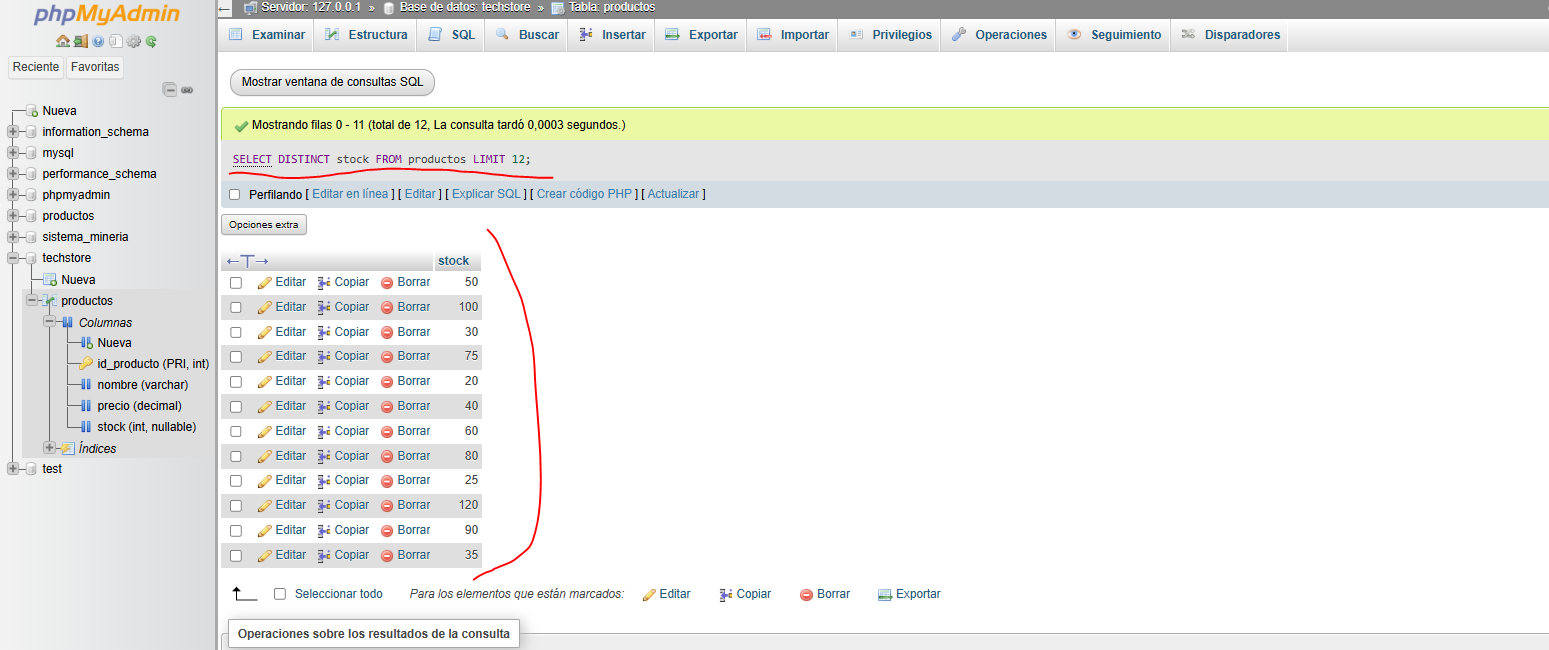


1. Se requiere saber qué 12 valores de stock hay en la base de datos sin repetirlos.

SELECT DISTINCT stock

FROM productos

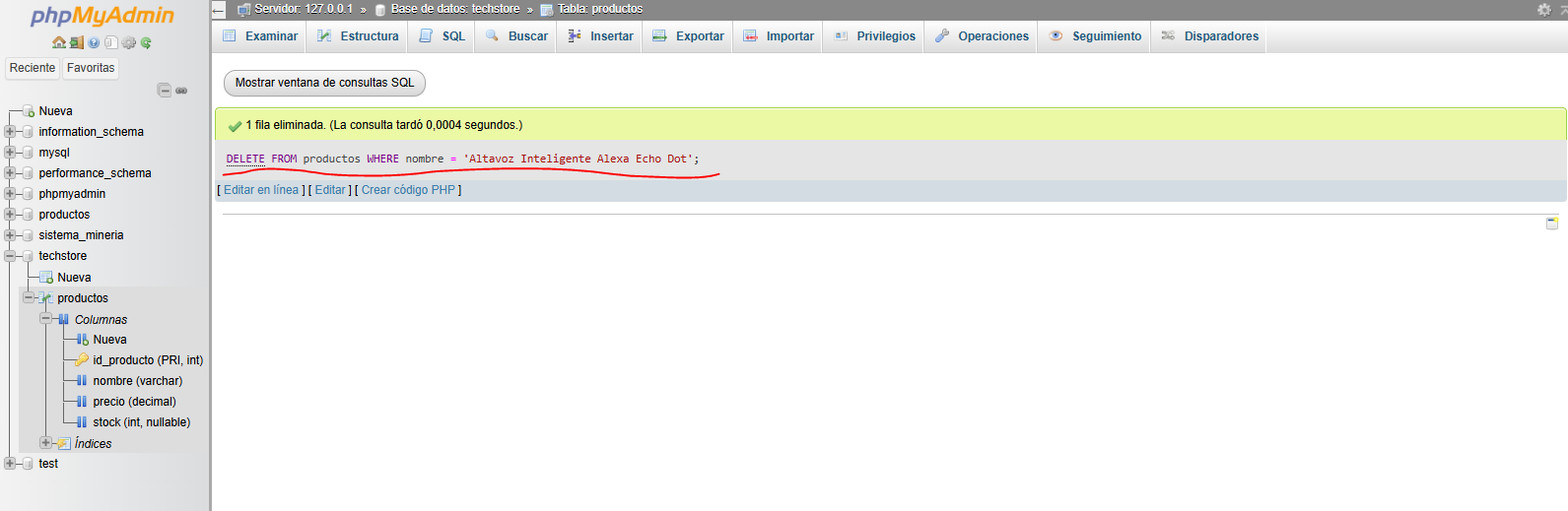
LIMIT 12;

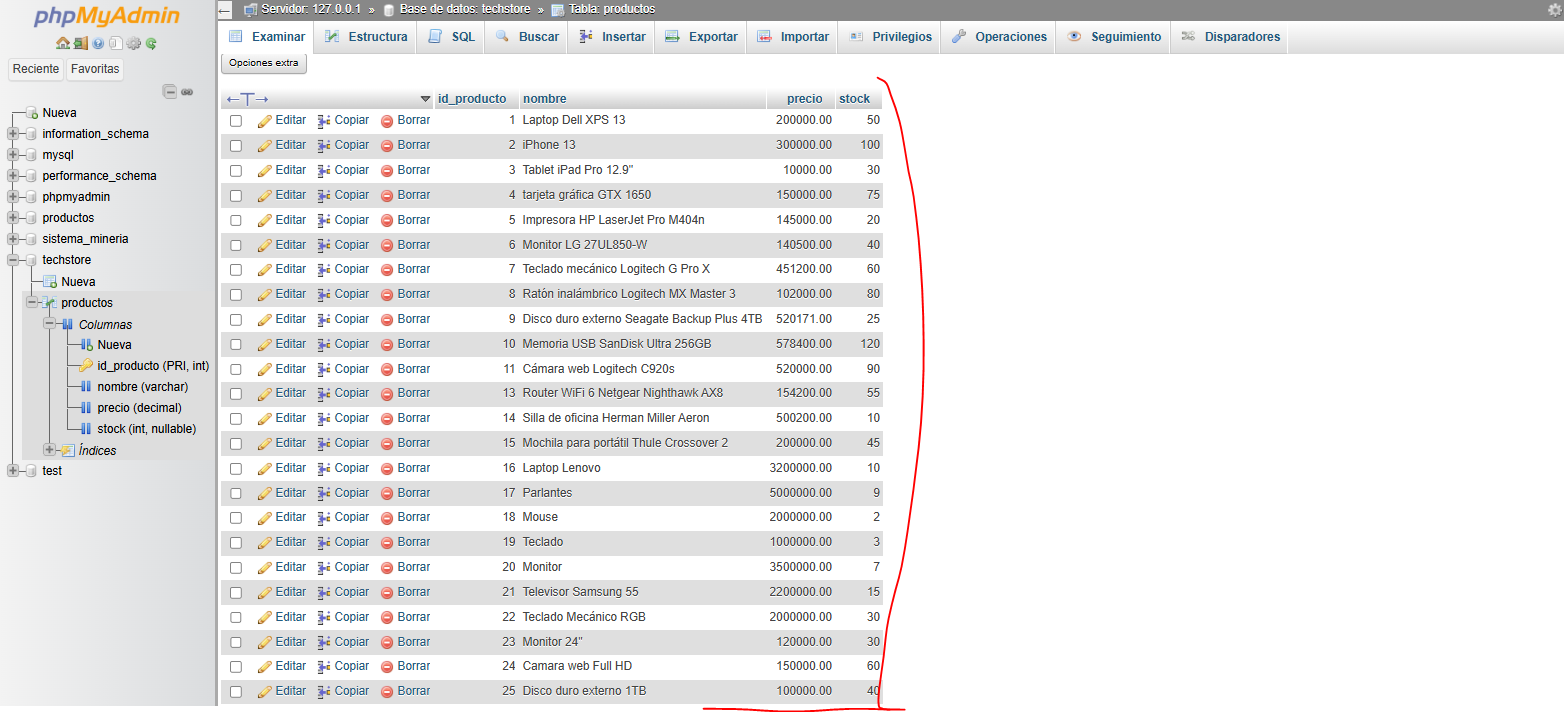


1. Se dejó de vender el producto de elección (Altavoz Inteligente Alexa Echo Dot) y lo eliminamos de la tabla.

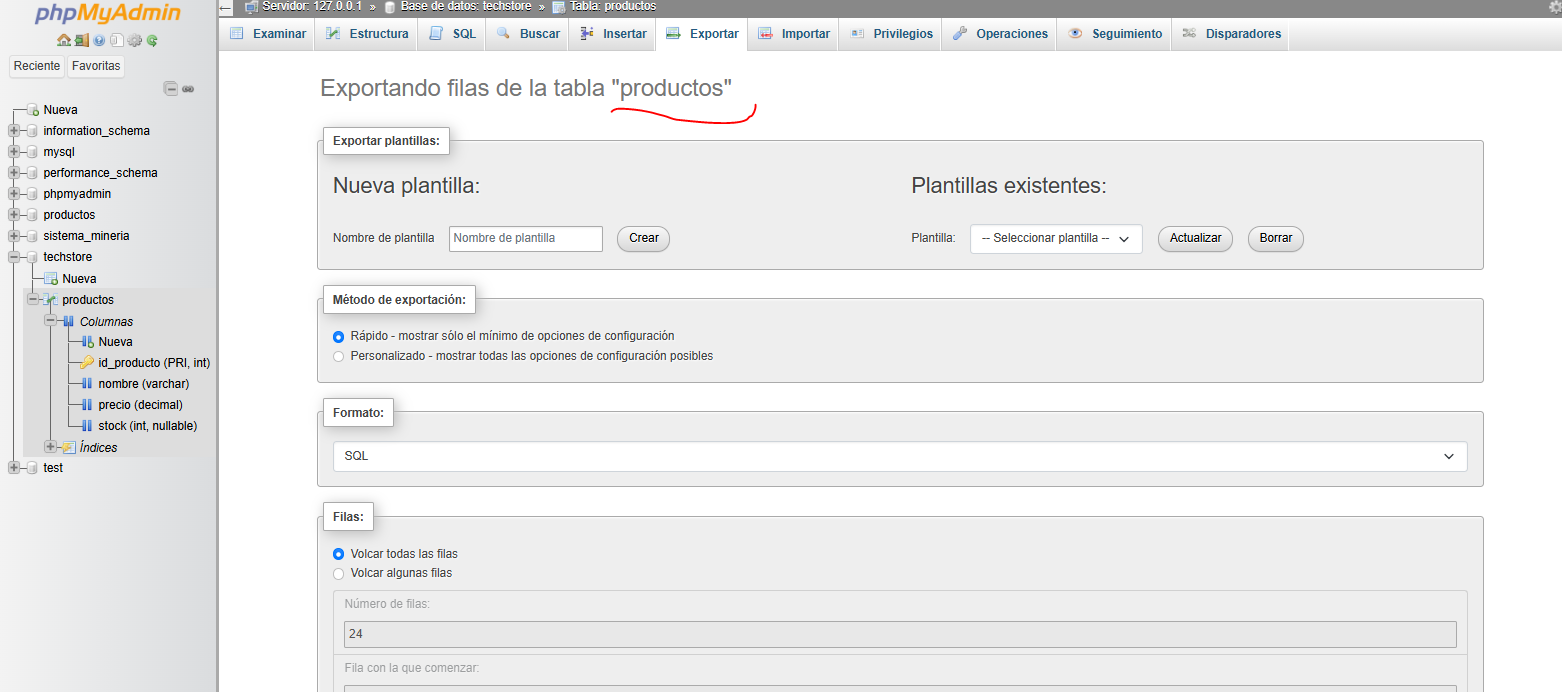
DELETE FROM productos

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.

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\_venta, id\_producto, cantidad, fecha) VALUES

(1, 2, 10, '2024-01-10'),

(2, 1, 5, '2024-02-15'),

(3, 5, 8, '2024-03-20'),

(4, 1, 15, '2024-04-25'),

(5, 3, 12, '2024-05-05'),

(6, 4, 3, '2024-06-10'),

(7, 2, 7, '2024-06-15'),

(8, 5, 10, '2024-07-01'),

(9, 3, 6, '2024-07-10'),

(10, 1, 20, '2024-08-05'),

(11, 2, 4, '2024-08-10'),

(12, 5, 18, '2024-09-01'),

(13, 1, 8, '2024-09-15'),

(14, 3, 5, '2024-09-25'),

(15, 4, 12, '2024-10-10'),

(16, 2, 9, '2024-10-15'),

(17, 5, 11, '2024-11-01'),

(18, 1, 10, '2024-11-15'),

(19, 3, 13, '2024-12-01'),

(20, 4, 5, '2024-12-10'),

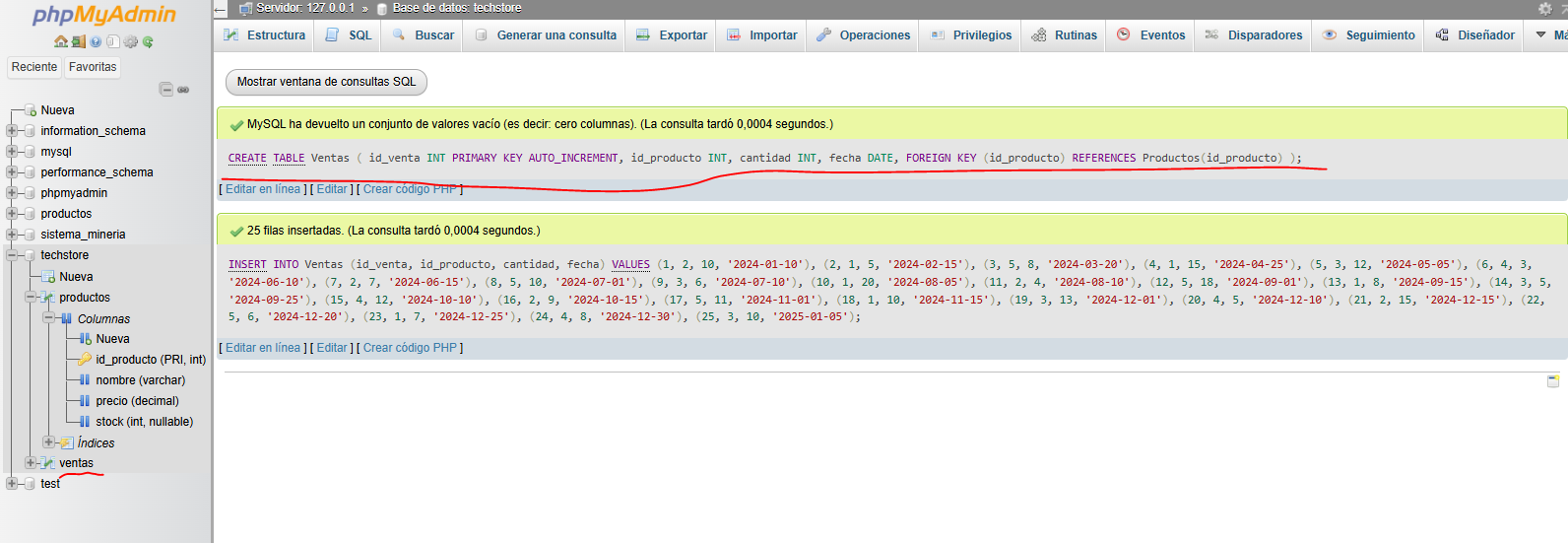
(21, 2, 15, '2024-12-15'),

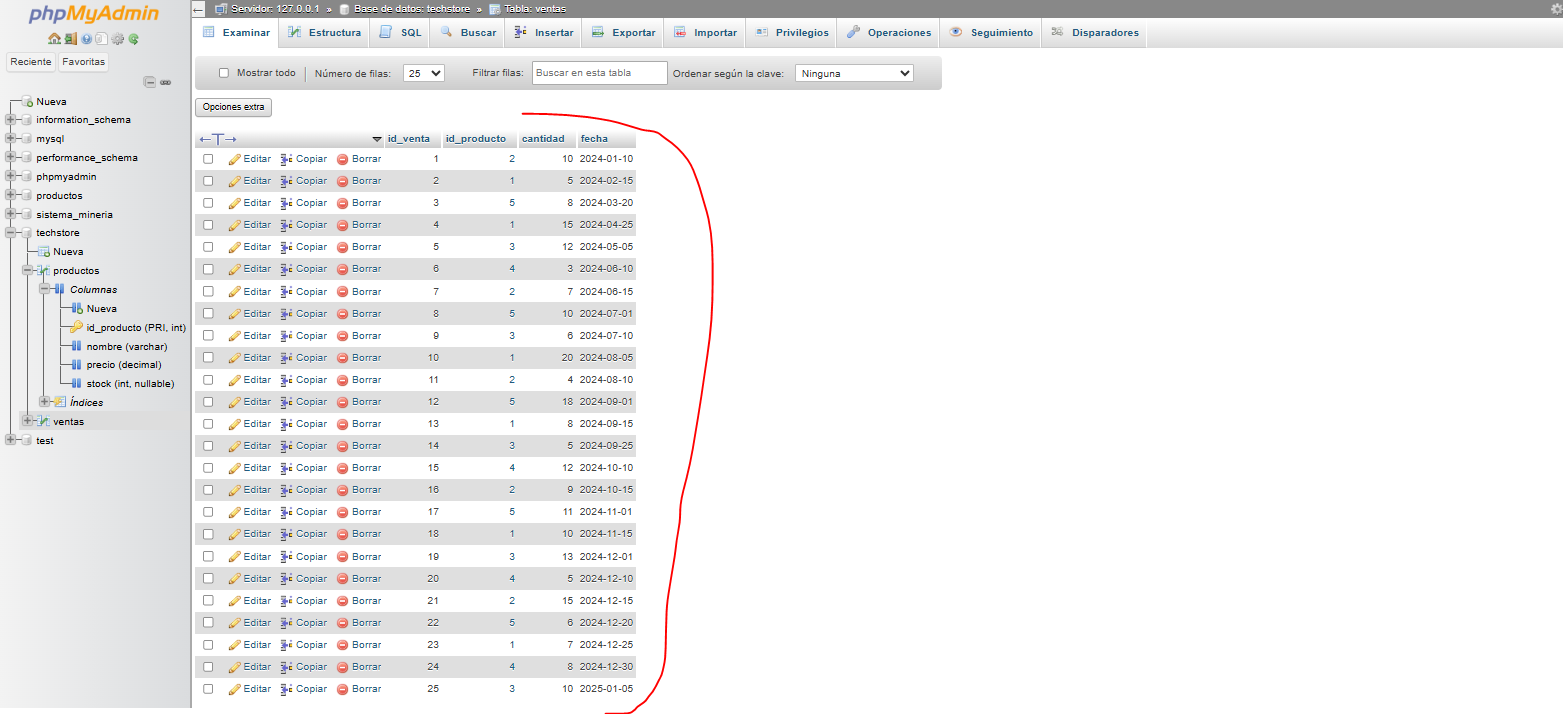
(22, 5, 6, '2024-12-20'),

(23, 1, 7, '2024-12-25'),

(24, 4, 8, '2024-12-30'),

(25, 3, 10, '2025-01-05');



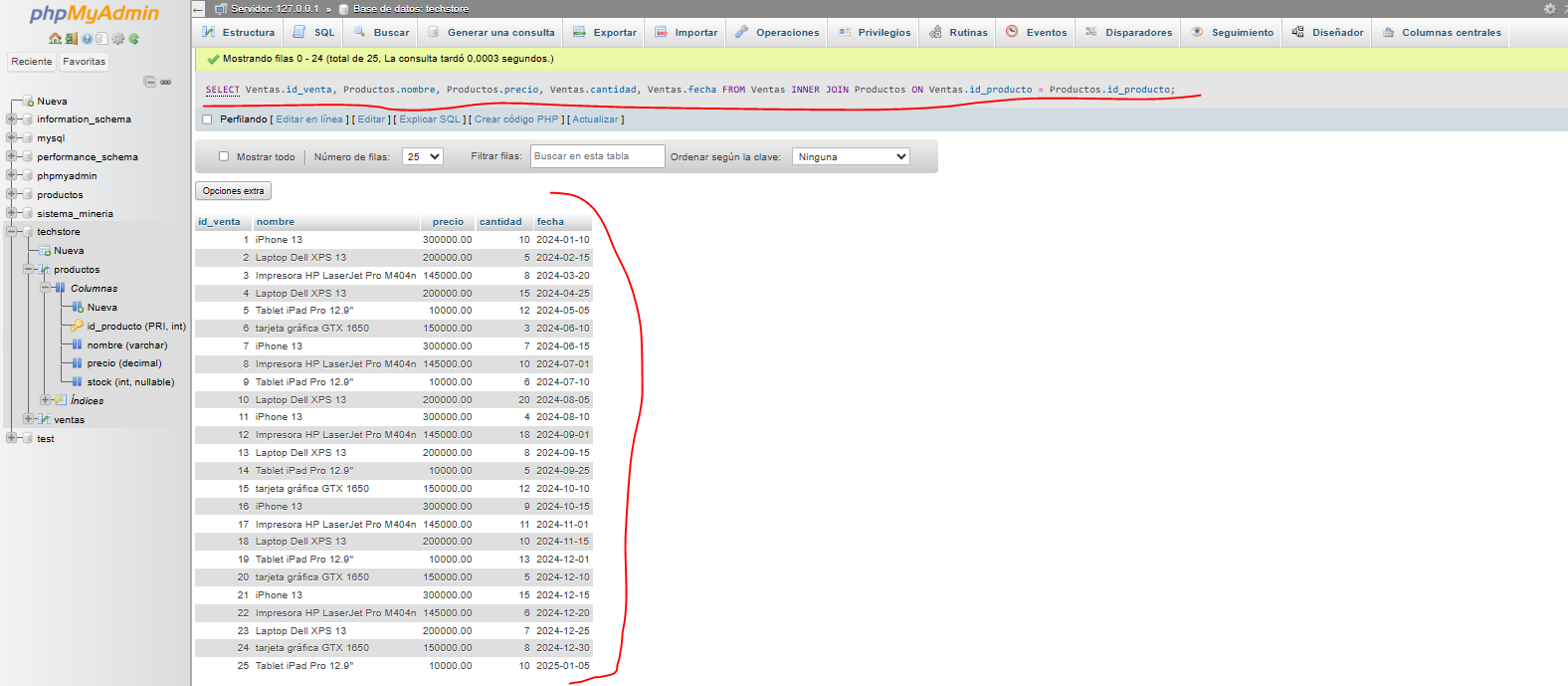


INNER JOIN

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

FROM Ventas

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

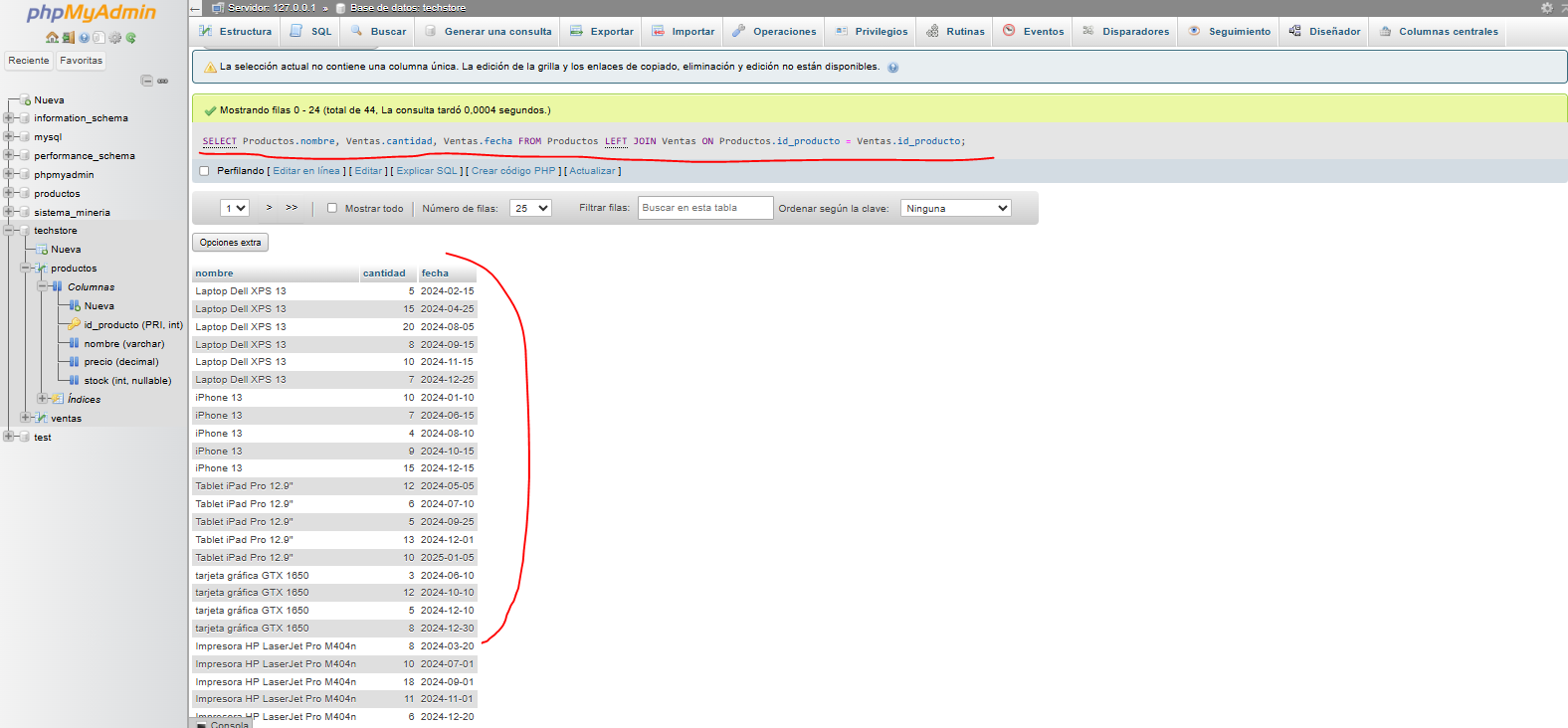


LEFT JOIN

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

FROM Productos

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

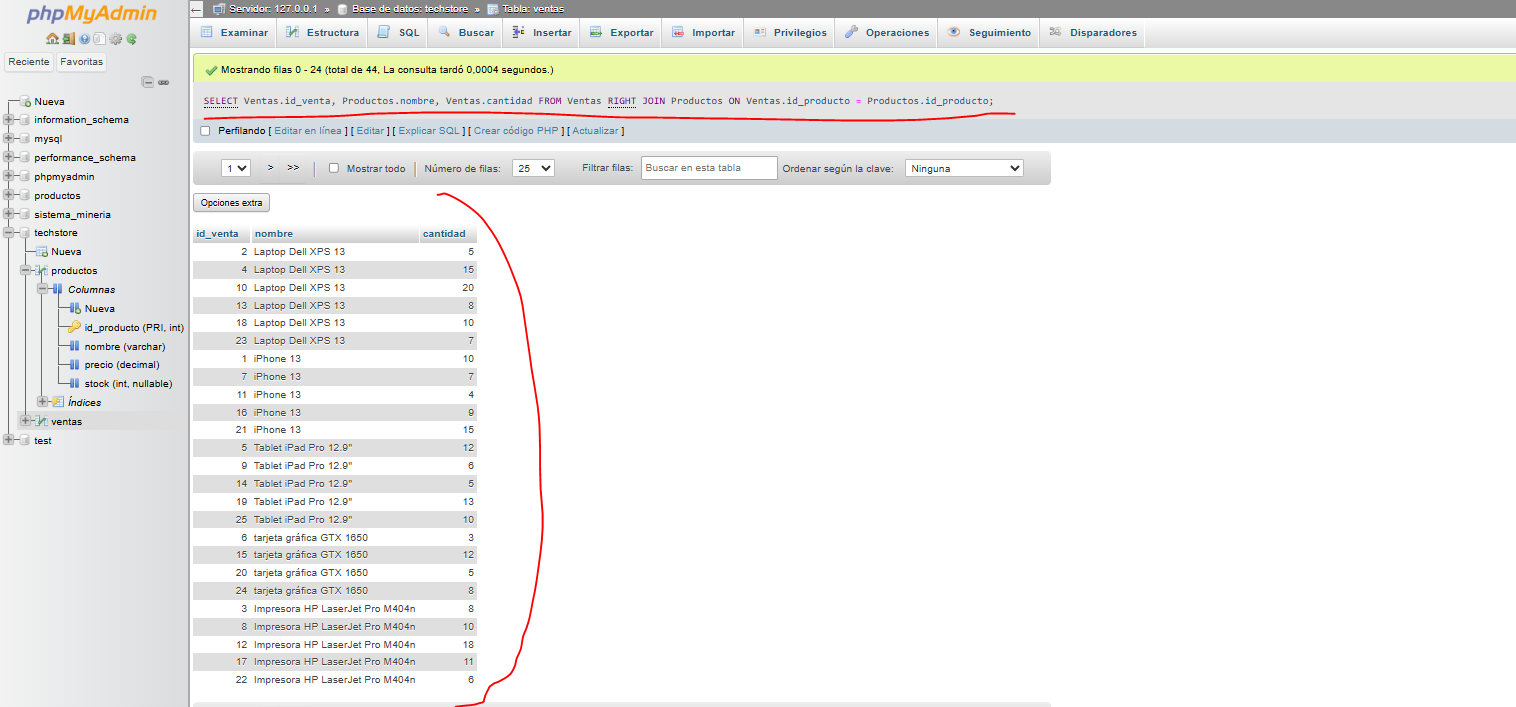


RIGHT JOIN

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

FROM Ventas

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



FULL OUTER JOIN

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

FROM Productos

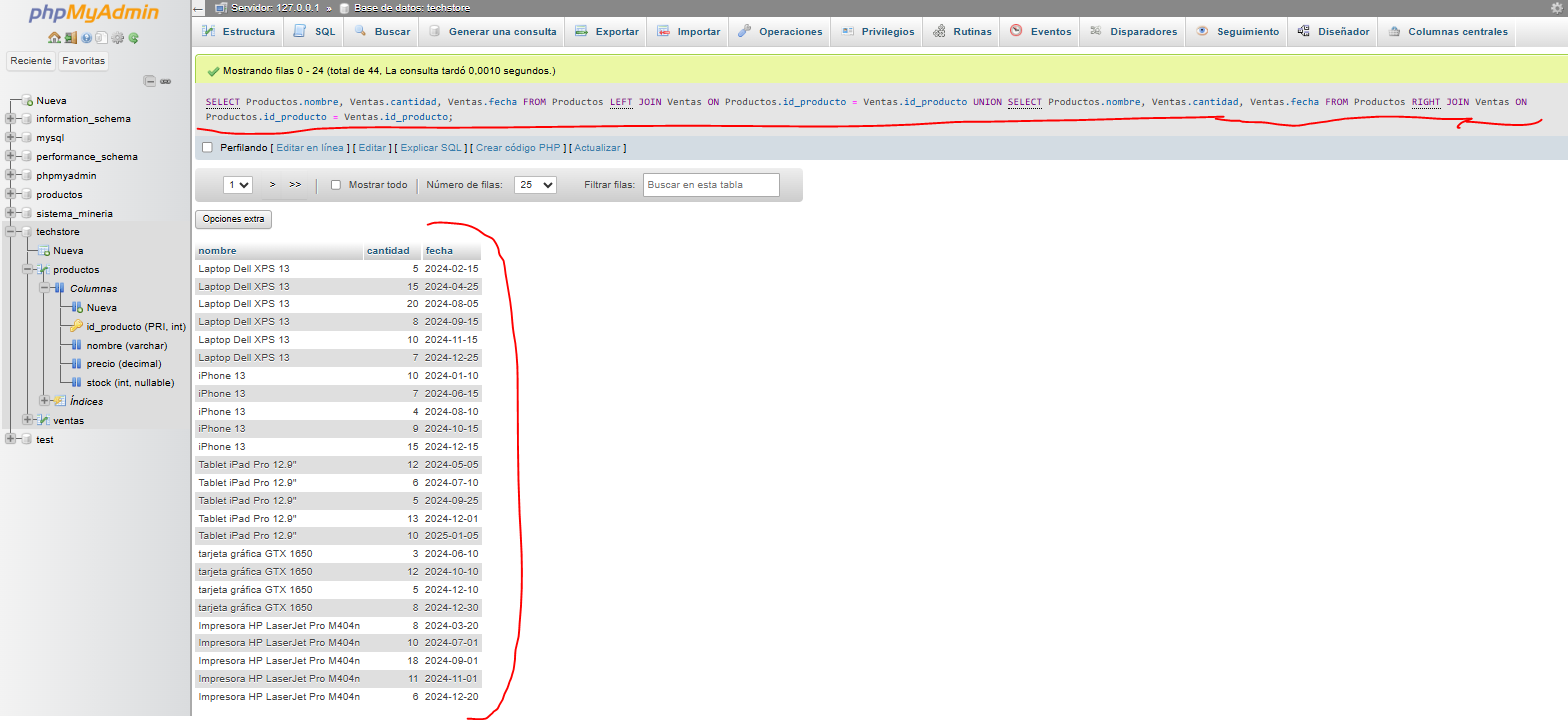
LEFT JOIN Ventas ON Productos.id\_producto = Ventas.id\_producto

UNION

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

FROM Productos

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



CROSS JOIN

SELECT Productos.nombre, Ventas.fecha

FROM Productos

CROSS JOIN Ventas;

