**TALLER PRACTICO BASES DE DATOS RELACIONALES SQL**

**PRECONDICIONES:** Para la realización de este ejercicio es necesario tener instalado el servidor XAMPP, de no tenerlo instalado en el equipo colocar el ticket a Xelerica.

**ALCANCE**

Para el desarrollo de este reto es importante estudiar acerca de las consultas básicas, ejecución de **Select, Update, Delete, Insert.**

* **OPERADORES LÓGICOS Y COMPARATIVOS:** Min, Max, And, Not, Or, Between, In, Distinct.
* **JOINS:** Joins, Inner Join, Left Join, Right Join, Full Join, Self Join
* **CLAUSULAS COMO:** Where, Like, Order By, Group By, Asc, Desc.

**Ejercicio:**

1. Realizar las lecturas del material de apoyo entregado para los diferentes temas estipulados en el alcance.

**Links De Apoyo:** [SQL SELECT Statement (w3schools.com)](https://www.w3schools.com/sql/sql_select.asp)

[SQL Tutorial - Dofactory](https://www.dofactory.com/sql/)  
**NOTA: Puedes elegir cualquiera de los dos links anteriores**

1. Después de tener instalado en el servidor XAMPP, ingresar a la URL:

<http://localhost/phpmyadmin/index.php?route=/server/databases>

**Puntos a Evaluar**

1. **\* Proceder a copiar el siguiente Script de Bases de datos:**

CREATE DATABASE tienda CHARACTER SET utf8mb4;

USE tienda;

CREATE TABLE fabricante (

codigo INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL

);

CREATE TABLE producto (

codigo INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

precio DOUBLE NOT NULL,

codigo\_fabricante INT UNSIGNED NOT NULL,

FOREIGN KEY (codigo\_fabricante) REFERENCES fabricante(codigo)

);

INSERT INTO fabricante VALUES(1, 'Asus');

INSERT INTO fabricante VALUES(2, 'Lenovo');

INSERT INTO fabricante VALUES(3, 'Hewlett-Packard');

INSERT INTO fabricante VALUES(4, 'Samsung');

INSERT INTO fabricante VALUES(5, 'Seagate');

INSERT INTO fabricante VALUES(6, 'Crucial');

INSERT INTO fabricante VALUES(7, 'Gigabyte');

INSERT INTO fabricante VALUES(8, 'Huawei');

INSERT INTO fabricante VALUES(9, 'Xiaomi');

INSERT INTO producto VALUES(1, 'Disco duro SATA3 1TB', 86.99, 5);

INSERT INTO producto VALUES(2, 'Memoria RAM DDR4 8GB', 120, 6);

INSERT INTO producto VALUES(3, 'Disco SSD 1 TB', 150.99, 4);

INSERT INTO producto VALUES(4, 'GeForce GTX 1050Ti', 185, 7);

INSERT INTO producto VALUES(5, 'GeForce GTX 1080 Xtreme', 755, 6);

INSERT INTO producto VALUES(6, 'Monitor 24 LED Full HD', 202, 1);

INSERT INTO producto VALUES(7, 'Monitor 27 LED Full HD', 245.99, 1);

INSERT INTO producto VALUES(8, 'Portátil Yoga 520', 559, 2);

INSERT INTO producto VALUES(9, 'Portátil Ideapd 320', 444, 2);

INSERT INTO producto VALUES(10, 'Impresora HP Deskjet 3720', 59.99, 3);

INSERT INTO producto VALUES(11, 'Impresora HP Laserjet Pro M26nw', 180, 3);

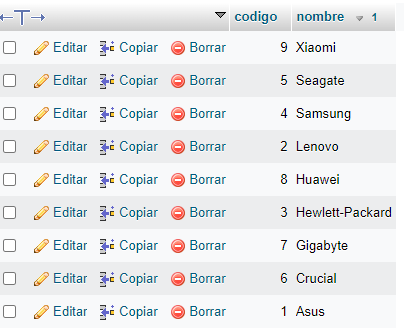
1. De Acuerdo a la Base de Datos creada anteriormente, llamada ‘Tienda’, que contiene dos tablas llamadas “Fabricante” y “Producto”. En ellas se encuentran registrados de los datos generales de cada producto de esta tienda con relación a cada uno de sus fabricantes.

**Realizar las siguientes consultas de una sola tabla:**

1. Lista los nombres de los fabricantes ordenados de forma descendente.

**R/:** Se usa la siguiente estructura para realizar la consulta de la tabla fabricante para mostrar los registros en forma descendente:

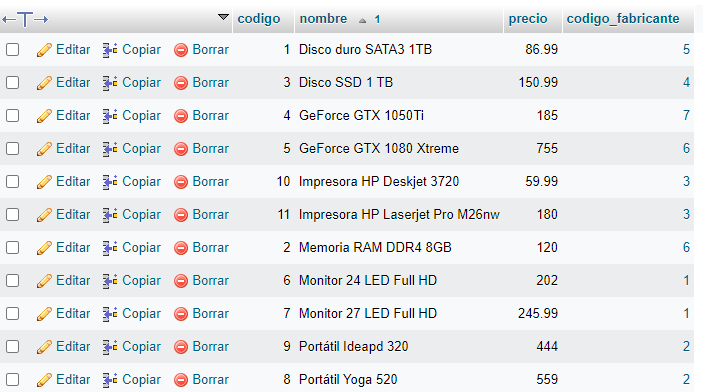
**SELECT \* FROM fabricante ORDER BY nombre DESC;**



1. Lista los nombres de los productos ordenados de manera Ascendente.

**R/:** Se usa la siguiente estructura para realizar la consulta en la tabla producto mostrando los registros de forma ascendente

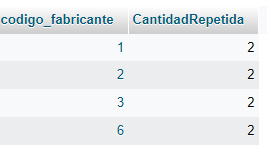
**SELECT \* FROM producto ORDER BY nombre ASC;**



1. Lista el código de los fabricantes que tienen productos en la tabla producto, eliminando los códigos que aparecen repetidos.

**R/:** Para saber cuáles están repetidos se usa la siguiente estructura

**SELECT codigo\_fabricante,COUNT(\*) AS CantidadRepetida FROM**  **producto GROUP BY codigo\_fabricante HAVING COUNT(\*) > 1;**



Con la siguiente estructura se muestran los códigos de los fabricantes omitiendo el registro repetido más reciente ya que con el distinct se muestra el más antiguo

**SELECT DISTINCT codigo\_fabricante FROM producto;**



1. Traer los productos que tienen menor precio.

**R/:** Se usa la siguiente estructura la cual nos trae el precio mínimo de la tabla

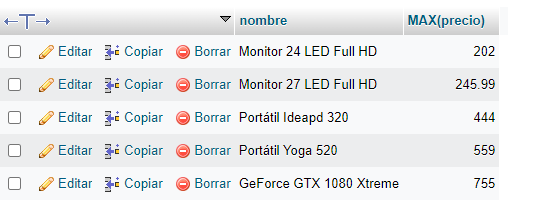
**SELECT MIN(precio) AS MenorPrecio FROM producto;**



1. Traer los cinco productos de mayor precio.

**R/:** Con la siguiente estructura logramos obtener los 5 productos con mayor precio

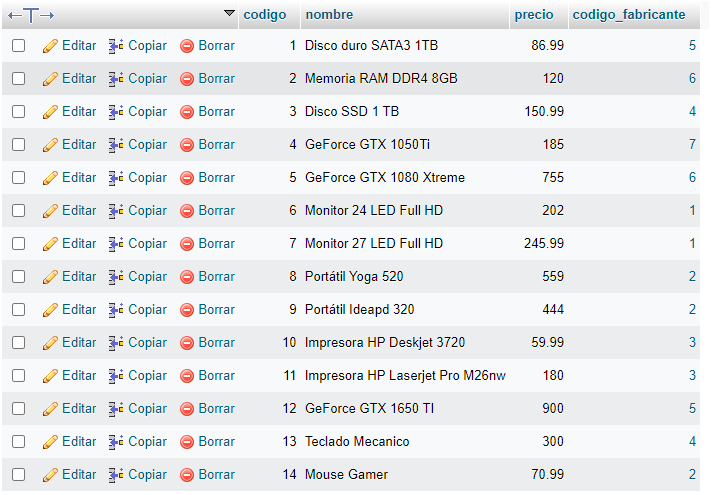
**SELECT nombre, MAX(precio) FROM producto GROUP BY precio HAVING**  **MAX(precio) >200 LIMIT 5**



1. Realizar la inserción de tres nuevos productos.

**R/:** Para realizar la inserción de nuevos datos de usa la siguiente estructura

**INSERT INTO producto (codigo, nombre, precio, codigo\_fabricante)**  **VALUES ('12', 'GeForce GTX 1650 TI', '900', '5'),('13', 'Teclado Mecanico',**  **'300', '4'),('14', 'Mouse Gamer', '70.99', '2');**



1. Consultar los productos que sea impresora HP.

**R/:** Para realizar esta consulta hacemos lo siguiente

**SELECT nombre FROM producto WHERE nombre LIKE '%Impresora Hp%';**



1. Eliminar uno de los nuevos productos creados.

**R/:** Con la siguiente estructura eliminamos el producto Mouse Gamer, como se

puede observar ha desaparecido de la tabla

**DELETE FROM producto WHERE nombre = 'Mouse Gamer'**



1. Actualizar el precio de uno de los productos creados.

**R/:** Con la siguiente estructura hacemos la actualización del precio, como se observa en la imagen se observa que el precio de 900 incrementa a 1000

**UPDATE producto SET precio = '1000' WHERE codigo = 12;**

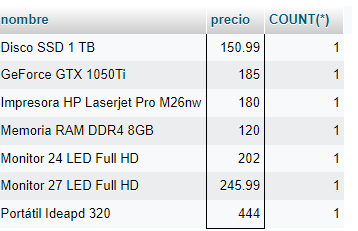




1. Traer los productos que estén dentro del rango de precios 120 y 500.

**R/:** Con la siguiente estructura buscamos los productos que están en el rango solicitado

**SELECT nombre,precio,COUNT(\*) FROM producto WHERE precio BETWEE** **N 120 AND 500 GROUP BY nombre, precio;**

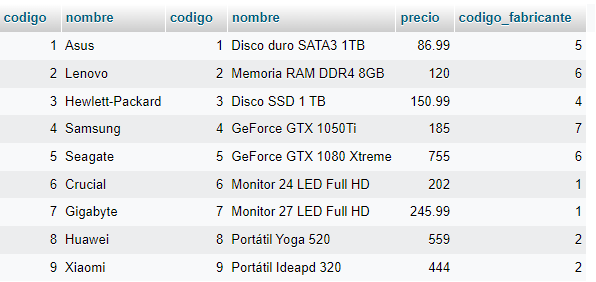


**Realiza las siguientes consultas mutitablas:**

1. Consultar el nombre de cada fabricante con los productos asociados a ellos.

**R/:** Para hacer esta consulta usamos la siguiente estructura

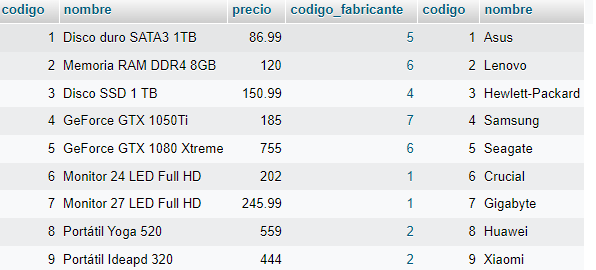
**SELECT\*FROM fabricante LEFT JOIN producto ON fabricante.codigo= prod** **ucto.codigo;**



1. Consultar cada producto con el nombre de cada fabricante

**R/:** Usamos la siguiente estructura

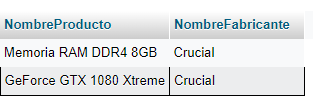
**SELECT\*FROM producto INNER JOIN fabricante ON producto.codigo= fabri** **cante.codigo;**



1. Traer los productos asociados cuyo fabricante es “Crucial” con su respectivo nombre.

**R/:** Con la siguiente estructura hacemos la consulta

**SELECT producto.nombre AS NombreProducto, fabricante.nombre AS**  **NombreFabricante FROM producto JOIN fabricante ON producto.codigo\_fabri** **cante = fabricante.codigo WHERE fabricante.nombre LIKE '%Crucial%';**



**Recuerda, que se realizara la respectiva sustentación de este reto.  
Mucha Suerte!!**

**Control de Cambios**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Cambios u Observaciones** | **Clasificación** | **Responsable** |
| 1.1 | 15/07/2022 | Se agrega el campo de clasificación al documento y se ajusta su píe de página. | Se asigna según la información que contenga.  (Confidencial, Restringido, Interno, Público) | Analista Creativo |
|  |  |  |  |  |