

**Verónica González Amor.**

### **UF2176: Ejercicio preguntas SQL enunciado 1.**

A continuación se presenta un conjunto de consultas SQL diseñadas para extraer diversa información de la base de datos de la empresa XYZ, la cual busca gestionar de manera integral su inventario mediante una aplicación web. Las consultas abordan desde consultas básicas sobre los productos registrados, hasta consultas más avanzadas que permiten analizar tendencias, detectar productos próximos a caducar o con baja disponibilidad. En caso de que se considere necesario, se pueden crear vistas para facilitar la reutilización de consultas complejas o mejorar el rendimiento del sistema al gestionar grandes volúmenes de datos. El objetivo de este ejercicio es practicar el diseño y ejecución de consultas eficientes que respondan a las necesidades de un sistema de gestión de inventarios robusto y escalable, teniendo en cuenta los distintos aspectos del negocio, como el control de stock, la relación con proveedores y la gestión de usuarios con distintos roles dentro del sistema.

#### **Preguntas:**

**¿Cómo obtener todos los productos registrados en la base de datos?**

```
SELECT * FROM Productos;
```

**¿Cómo mostrar el nombre y precio unitario de todos los productos?**

```
SELECT nombre, precio_unitario FROM Productos;
```

**¿Cómo obtener el nombre y la cantidad en stock de un producto en particular, filtrando por su nombre?**

```
SELECT NOMBRE, STOCK from productos WHERE nombre = 'Nombre del Producto'
```

**¿Cómo listar todos los productos que están actualmente en stock (cantidad > 0)?**

```
SELECT * FROM Productos WHERE stock > 0;
```

**¿Cómo mostrar los productos que pertenecen a una categoría específica?**

```
SELECT p*  
FROM Productos P  
JOIN Categorías c ON p.categoria_id = c.id  
WHERE c.nombre = 'Electrónica';
```

**¿Cómo obtener los productos con un precio mayor a 100 euros?**

```
SELECT * FROM Productos WHERE precio_unitario > 100;
```

**¿Cómo contar cuántos productos hay en cada categoría?**

```
SELECT c.nombre, COUNT(p.id) AS total_productos  
FROM Productos p  
JOIN Categorías c ON p.categoria_id = c.id
```

```
GROUP BY c.nombre
```

¿Cómo obtener el producto más caro de la base de datos?

```
SELECT * FROM Productos ORDER BY precio_unitario DESC  
LIMIT 1;
```

¿Cómo listar los productos cuyo nombre comienza con la letra "A"?

```
SELECT * FROM Productos WHERE nombre LIKE 'A%';
```

¿Cómo mostrar los productos con una fecha de caducidad en los próximos 30 días?

```
SELECT * FROM Productos  
WHERE fecha_caducidad BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL  
30 DAY;
```

¿Cómo mostrar todos los proveedores que suministran un producto específico?

```
SELECT pr.nombre  
FROM Proveedores pr  
JOIN Productos p ON pr.id = p.proveedor_id  
WHERE p.nombre = 'Producto Específico';
```

¿Cómo obtener los productos que están por debajo de un umbral de cantidad específica (por ejemplo, 5 unidades)?

```
SELECT * FROM Productos WHERE stock < 5;
```

¿Cómo listar todos los productos cuyo precio esté entre 50 y 150 euros?

```
SELECT * FROM Productos WHERE precio_unitario BETWEEN 50 AND 150;
```

¿Cómo obtener el valor total del inventario (sumar el valor de todos los productos en stock)?

```
SELECT SUM(precio_unitario * stock) AS valor_total FROM Productos;
```

¿Cómo obtener los productos cuyo precio ha sido actualizado recientemente (últimos 7 días)?

```
SELECT * FROM Productos  
WHERE fecha_actualización >= CURRENT_DATE - INTERVAL 7 DAY;
```

¿Cómo obtener todos los productos de un proveedor específico, ordenados por precio de mayor a menor?

```
SELECT * FROM Productos  
WHERE proveedor_id = 123  
ORDER BY precio_unitario DESC;
```

**¿Cómo mostrar los productos que no están en stock actualmente (cantidad = 0)?**

```
SELECT * FROM Productos WHERE stock = 0;
```

**¿Cómo obtener un reporte con los productos que están cerca de su fecha de caducidad (por ejemplo, en los próximos 10 días) y su cantidad en stock?**

```
SELECT nombre, stock, fecha_caducidad  
FROM Productos  
WHERE fecha_caducidad BETWEEN CURRENT_DATE and CURRENT_DATE + INTERVAL  
10 DAY;
```

**¿Cómo obtener el promedio del precio unitario de todos los productos de una categoría específica?**

```
SELECT c.nombre, AVG(p.precio_unitario) AS precio_promedio  
FROM productos p  
JOIN Categorías c ON p.categoria_id = c.id  
GROUP BY c.nombre;
```

**¿Cómo listar los productos más recientes (adquiridos en los últimos 30 días)?**

```
SELECT * FROM Productos  
WHERE fecha_adquisicion >= CURRENT_DATE - INTERVAL 30 DAY;
```

**¿Cómo mostrar todos los productos, junto con la categoría y el proveedor?**

```
SELECT p.*, c.nombre AS categoria, pr.nombre AS proveedor  
FROM Productos p  
JOIN Categorías c ON p.categoria_id = c.id  
JOIN Proveedores pr ON p.proveedor_id = pr.id;
```

**¿Cómo obtener la cantidad total de productos disponibles por categoría, ordenados de mayor a menor cantidad?**

```
SELECT c.nombre, SUM(p.stock) AS stock_total  
FROM Productos p  
JOIN Categorías c ON p.categoria_id = c.id  
GROUP BY c.nombre  
ORDER BY stock_total DESC;
```

**¿Cómo generar un informe que muestre los productos con la menor cantidad en stock y el proveedor de cada uno?**

```
SELECT p.nombre, p.stock, pr.nombre AS proveedor  
FROM Productos p
```

```
JOIN Proveedores pr ON p.proveedor_id = pr.id
WHERE p.stock = (SELECT MIN(stock) FROM Productos);
```

### ¿Cómo obtener el valor total del inventario por proveedor?

```
SELECT pr.nombre, SUM(p.precio_unitario * p.stock) AS valor_total
FROM Productos p
JOIN Proveedores pr ON p.proveedor_id = pr.id
GROUP BY pr.nombre;
```

### ¿Cómo listar todos los usuarios que han generado reportes en un rango de fechas específico, mostrando el ID del reporte y la fecha de generación?

```
SELECT u.nombre, r.id AS reporte_id, r.fecha_generacion
FROM Usuarios u
JOIN Reportes r ON u.id = r.usuario_id
WHERE r.fecha_generacion BETWEEN '2025-01-01' AND '2025-03-30';
```

### Preguntas Extra:

#### ¿Cómo obtener el promedio de días que transcurren entre la fecha de adquisición y la fecha de caducidad de los productos en cada categoría?

```
SELECT c.nombre, AVG(DATEDIFF(fecha_caducidad, fecha_adquisicion))
AS promedio_dias
FROM Productos p
JOIN Categorías c ON p.categoria_id = c.id
GROUP BY c.nombre;
```

#### ¿Cómo obtener una lista de productos que están cerca de su fecha de caducidad y que tienen el stock más bajo dentro de su categoría?

```
WITH Rankeados AS (
SELECT p.*, RANK() OVER (PARTITION BY categoria_id ORDER BY
Stock AS ranking
FROM Productos
WHERE fecha_caducidad BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL
30 DAY)
SELECT * FROM Rankeados WHERE ranking = 1;
```

#### ¿Cómo identificar el top 3 de proveedores que suministran productos cuyo valor total en inventario es mayor al promedio de todos los proveedores?

```
WITH ValorProveedores AS (
SELECT pr.id, pr.nombre, SUM(p.precio_unitario * p.stock)
AS valor_total
FROM Proveedores pr
JOIN Productos p ON pr.id = p.proveedor_id
GROUP BY pr.id, pr.nombre)
```

```
SELECT nombre, valor_total > (SELECT AVG(valor_total) FROM
ValorProveedores)
ORDER BY valor_total DESC
LIMIT 3;
```

**¿Cómo calcular el valor proyectado del inventario en los próximos 6 meses, considerando productos que caducarán en ese período y excluyéndolos del valor total?**

```
SELECT
    (SELECT SUM(precio_uniotario * stock) FROM Productos) -
    (SELECT SUM(precio_unitario * stock) FROM Productos)
    WHERE fecha_caducidad <= CURRENT_DATE + INTERVAL 6 MONTH)
AS valor_proyectado;
```

**¿Cómo obtener los productos cuyo stock ha permanecido sin cambios durante los últimos 90 días, ordenados por el tiempo que llevan sin ser actualizados?**

```
SELECT *
FROM Productos
WHERE fecha_actualizacion_stock <= CURRENTDATE - INTERVAL 90 DAY
ORDER BY fecha_actualizacion_stock;
```