

# Taller mysql 2

---

## SUBCONSULTAS

**\*\*Producto más caro en cada categoría**

```
SELECT *
FROM Productos p
WHERE precio = (
    SELECT MAX(p2.precio)
    FROM Productos p2
    JOIN Subcategorias sc ON p2.subcategoria_id = sc.id
    WHERE sc.categoria_id = (
        SELECT sc2.categoria_id FROM Subcategorias sc2 WHERE sc2.id =
p.subcategoria_id
    )
);
```

**\*\*Cliente con mayor total en pedidos**

```
SELECT *
FROM Clientes
WHERE id = (
    SELECT cliente_id
    FROM Pedidos
    GROUP BY cliente_id
    ORDER BY SUM(total) DESC
    LIMIT 1
);
```

**\*\*Empleados que ganan más que el salario promedio**

```
SELECT *
FROM Empleados
WHERE salario > (
    SELECT AVG(salario)
    FROM Empleados
);
```

**\*\*Encuentra el pedido con el total de ventas más alto.**

```
SELECT *
FROM Pedidos
WHERE pedido_id = (
    SELECT pedido_id
    FROM DetallesPedido
    GROUP BY pedido_id
    ORDER BY SUM(precio_unitario * cantidad) DESC
    LIMIT 1
);
```

**\*\*Pedidos cuyo total es mayor al promedio de todos los pedidos**

```
SELECT *
FROM Pedidos
WHERE total > (
    SELECT AVG(total)
    FROM Pedidos
);
```

**\*\*3 proveedores con más productos**

```
SELECT proveedor_id, COUNT(*) AS total_productos
FROM Productos
GROUP BY proveedor_id
ORDER BY total_productos DESC
LIMIT 3;
```

**\*\*Productos con precio superior al promedio de su tipo (subcategoría)**

```
SELECT *
FROM Productos p
WHERE precio > (
    SELECT AVG(p2.precio)
    FROM Productos p2
    WHERE p2.subcategoria_id = p.subcategoria_id
);
```

**\*\*Clientes que han realizado más pedidos que la media**

```
SELECT *
FROM Clientes
```

```

WHERE id IN (
    SELECT cliente_id
    FROM Pedidos
    GROUP BY cliente_id
    HAVING COUNT(*) >= (
        SELECT AVG(cnt)
        FROM (
            SELECT COUNT(*) AS cnt
            FROM Pedidos
            GROUP BY cliente_id
        ) AS sub
    )
);

```

**\*\*Encontrar productos cuyo precio es mayor que el promedio de todos los productos**

```

SELECT *
FROM Productos
WHERE precio > (
    SELECT AVG(precio)
    FROM Productos
);

```

**\*\*Mostrar empleados cuyo salario es menor al promedio del departamento**

```

SELECT *
FROM Empleados e
WHERE salario < (
    SELECT AVG(e2.salario)
    FROM Empleados e2
    WHERE e2.puesto = e.puesto
);

```

## PROCEDIMIENTOS ALMACENADOS

**\*\*Actualizar el precio de todos los productos de un proveedor**

```

DELIMITER $$

CREATE PROCEDURE actualizar_precio(
    IN precio DECIMAL(10,2)
)

```

```

BEGIN
    INSERT INTO Productos (precio)
    VALUES (precio);
END$$

DELIMITER ;

```

**\*\*Un procedimiento que devuelva la dirección de un cliente por ID**

```

DELIMITER $$

CREATE PROCEDURE obtener_direccion(
    IN cliente_id INT,
    OUT direccion VARCHAR(200)
)

BEGIN
    SELECT direccion INTO direccion
    FROM UbicacionCliente
    WHERE cliente_id = cliente_id;
END$$

DELIMITER ;

```

**\*\*Registrar un pedido nuevo y sus detalles**

```

DELIMITER $$

CREATE PROCEDURE nuevo_pedido(
    IN id INT,
    IN cantidad INT,
    IN producto_id INT,
    IN pedido_id INT,
    IN precio_unidad DECIMAL (10,2)
)

BEGIN
    INSERT TO DetallesPedido (cantidad, producto_id, precio_unidad)
    VALUES (cantidad, producto_id, precio_unidad);
END$$

DELIMITER ;

```

**\*\*Calcular el total de ventas de un cliente**

```
DELIMITER $$
CREATE PROCEDURE registrar_pedido(
IN p_cliente_id INT,
IN p_empleado_id INT,
IN p_fecha DATE,
IN p_total DECIMAL(10,2),
IN p_producto_id INT,
IN p_cantidad INT,
IN p_precio DECIMAL(10,2)
)
BEGIN
DECLARE v_pedido_id INT;
INSERT INTO Pedidos (cliente_id, fecha, total, empleado_id)
VALUES (p_cliente_id, p_fecha, p_total, p_empleado_id);
SET v_pedido_id = LAST_INSERT_ID();
INSERT INTO DetallesPedido (pedido_id, producto_id, cantidad, precio_unidad)
VALUES (v_pedido_id, p_producto_id, p_cantidad, p_precio);
END $$

DELIMITER ;
```

**\*\*Obtener los empleados por puesto**

```
DELIMITER $$
CREATE PROCEDURE empleados_por_puesto(
IN p_puesto VARCHAR(50)
)
BEGIN
SELECT * FROM Empleados
WHERE puesto = p_puesto;
END $$

DELIMITER ;
```

**\*\*Actualizar el salario de empleados por puesto**

```
DELIMITER $$
CREATE PROCEDURE actualizar_salario_por_puesto(
IN p_puesto VARCHAR(50),
IN p_incremento DECIMAL(5,2)
)
```

```

)
BEGIN
UPDATE Empleados
SET salario = salario * (1 + p_incremento / 100)
WHERE puesto = p_puesto;
END $$

DELIMITER ;

```

**\*\*Listar pedidos entre dos fechas**

```

DELIMITER $$
CREATE PROCEDURE pedidos_entre_fechas(
IN p_inicio DATE,
IN p_fin DATE
)
BEGIN
SELECT * FROM Pedidos
WHERE fecha BETWEEN p_inicio AND p_fin;
END $$

DELIMITER ;

```

**\*\*Aplicar descuento a productos de una categoría**

```

DELIMITER $$
CREATE PROCEDURE aplicar_descuento_categoria(
IN p_categoria_id INT,
IN p_descuento DECIMAL(5,2)
)
BEGIN
UPDATE Productos p
JOIN Subcategorias sc ON p.subcategoria_id = sc.id
SET p.precio = p.precio * (1 - p_descuento / 100)
WHERE sc.categoria_id = p_categoria_id;
END $$

DELIMITER ;

```

**\*\*Listar todos los proveedores de un tipo de producto (subcategoría)**

```
DELIMITER $$
CREATE PROCEDURE proveedores_por_tipo(
IN p_subcategoria_id INT
)
BEGIN
SELECT DISTINCT pr.id, pr.nombre
FROM Proveedores pr
JOIN Productos p ON p.proveedor_id = pr.id
WHERE p.subcategoria_id = p_subcategoria_id;
END $$

DELIMITER ;
```

**\*\*Devolver el pedido de mayor valor**

```
DELIMITER $$
CREATE PROCEDURE pedido_mayor_valor()
BEGIN
SELECT *
FROM Pedidos
ORDER BY total DESC
LIMIT 1;
END $$

DELIMITER ;
```