SUBCONSULTAS

Consultar el producto más caro en cada categoría.

```sql
SELECT p.nombre AS productos, p.precio AS Precio, s.categoria_id
FROM Productos AS p
JOIN Subcategorias AS s ON p.subcategoria_id = s.id
WHERE (s.categoria_id, p.precio) IN (
    SELECT s2.categoria_id, MAX(p2.precio)
    FROM Productos AS p2
    JOIN Subcategorias AS s2 ON p2.subcategoria_id = s2.id
    GROUP BY s2.categoria_id
);
```

Encontrar el cliente con mayor total en pedidos.

```sql
SELECT nombre,tipo_documento_id
FROM Clientes
WHERE id = ( SELECT cliente_id FROM Pedidos GROU
P BY cliente_id ORDER BY SUM(total) DESC LIMIT 1 );
```

Listar empleados que ganan más que el salario promedio

```sql
SELECT e.nombre
FROM DatosEmpleados AS e
WHERE salario > (
SELECT AVG(salario)
FROM DatosEmpleados
);
```

Pedidos cuyo total es mayor al promedio de todos los pedidos

```sql
SELECT p.id AS Pedido_id
FROM Pedidos AS p
WHERE p.total > (
SELECT AVG(p.total)
FROM Pedidos
);
```

Seleccionar los 3 proveedores con más productos

```
SELECT nombre,
       (SELECT COUNT(*) FROM Productos p WHERE p.proveedor_id = pr.id) AS
total_productos
FROM Proveedores pr
ORDER BY total_productos DESC
LIMIT 3;
```

Consultar productos con precio superior al promedio en su tipo.

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

Mostrar clientes que han realizado más pedidos que la media.

```
SELECT nombre
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

```sql
SELECT nombre,precio
FROM Productos
WHERE precio > (
    SELECT AVG(precio)
    FROM Productos
);
```

Mostrar empleados cuyo salario es menor al promedio del departamento

```sql
SELECT nombre,salario
FROM DatosEmpleados e
WHERE salario < (
SELECT AVG(e2.salario)
FROM DatosEmpleados e2
WHERE e2.id_puesto = e.id_puesto
);
```

PROCEDIMIENTOS ALMACENADOS

Actualizar el precio de todos los productos de un proveedor

```sql
DELIMITER //
CREATE PROCEDURE actualizar_precios_proveedor(
IN p_proveedor_id INT,
IN p_porcentaje DECIMAL(5,2)
)
BEGIN
UPDATE Productos
SET precio = precio * (1 + p_porcentaje / 100)
WHERE proveedor_id = p_proveedor_id;
END;
//
DELIMITER ;
```

Un procedimiento que devuelva la dirección de un cliente por ID.

```
DELIMITER //
CREATE PROCEDURE obtener_direccion_cliente(
IN p_cliente_id INT
)
BEGIN
SELECT uc.direccion, uc.codigo_postal
FROM UbicacionCliente uc
WHERE uc.cliente_id = p_cliente_id;
END;
//
DELIMITER ;
```

Registrar un pedido nuevo y sus detalles

```
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 ;
```

Calcular el total de ventas de un cliente

```
DELIMITER //
CREATE PROCEDURE total_ventas_cliente(
IN p_cliente_id INT
)
BEGIN
SELECT SUM(total) AS total_ventas
FROM Pedidos
WHERE cliente_id = p_cliente_id;
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 ;
```