# Trabajo Práctico: Optimización de Consultas y Consultas Multitablas

En este trabajo práctico se trabajará sobre la optimización de consultas SQL.

El objetivo es comprender cómo utilizar EXPLAIN, identificar cuellos de botella, y comparar el rendimiento entre subconsultas y joins múltiples.

Se utilizarán tablas de un sistema de ventas.

## 1. Esquema de la Base de Datos

CREATE TABLE clientes (  
 id\_cliente INT PRIMARY KEY,  
 nombre VARCHAR(100),  
 ciudad VARCHAR(50)  
);  
  
CREATE TABLE productos (  
 id\_producto INT PRIMARY KEY,  
 nombre VARCHAR(100),  
 precio DECIMAL(10,2)  
);  
  
CREATE TABLE ventas (  
 id\_venta INT PRIMARY KEY,  
 id\_cliente INT,  
 id\_producto INT,  
 cantidad INT,  
 fecha DATE,  
 FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente),  
 FOREIGN KEY (id\_producto) REFERENCES productos(id\_producto)  
);

## 2. Scripts para Generar Datos de Prueba

Clientes:

INSERT INTO clientes (id\_cliente, nombre, ciudad) VALUES  
(1, 'Cliente1', 'Ciudad2'),  
(2, 'Cliente2', 'Ciudad3'),  
(3, 'Cliente3', 'Ciudad4'),  
(4, 'Cliente4', 'Ciudad5'),  
(5, 'Cliente5', 'Ciudad1'),  
(6, 'Cliente6', 'Ciudad2'),  
(7, 'Cliente7', 'Ciudad3'),  
(8, 'Cliente8', 'Ciudad4'),  
(9, 'Cliente9', 'Ciudad5'),  
(10, 'Cliente10', 'Ciudad1'),  
(11, 'Cliente11', 'Ciudad2'),  
(12, 'Cliente12', 'Ciudad3'),  
(13, 'Cliente13', 'Ciudad4'),  
(14, 'Cliente14', 'Ciudad5'),  
(15, 'Cliente15', 'Ciudad1'),  
(16, 'Cliente16', 'Ciudad2'),  
(17, 'Cliente17', 'Ciudad3'),  
(18, 'Cliente18', 'Ciudad4'),  
(19, 'Cliente19', 'Ciudad5'),  
(20, 'Cliente20', 'Ciudad1');

Productos:

INSERT INTO productos (id\_producto, nombre, precio) VALUES  
(1, 'Producto1', 217.22),  
(2, 'Producto2', 373.31),  
(3, 'Producto3', 435.88),  
(4, 'Producto4', 145.02),  
(5, 'Producto5', 106.77),  
(6, 'Producto6', 463.87),  
(7, 'Producto7', 154.59),  
(8, 'Producto8', 121.81),  
(9, 'Producto9', 378.37),  
(10, 'Producto10', 114.29),  
(11, 'Producto11', 239.01),  
(12, 'Producto12', 98.35),  
(13, 'Producto13', 347.58),  
(14, 'Producto14', 201.02),  
(15, 'Producto15', 123.84),  
(16, 'Producto16', 445.32),  
(17, 'Producto17', 330.29),  
(18, 'Producto18', 188.11),  
(19, 'Producto19', 115.54),  
(20, 'Producto20', 83.76);

Ventas:

INSERT INTO ventas (id\_venta, id\_cliente, id\_producto, cantidad, fecha) VALUES  
(1, 2, 4, 6, '2024-03-10'),  
(2, 4, 18, 5, '2024-03-07'),  
(3, 1, 18, 4, '2024-01-12'),  
(4, 15, 19, 4, '2024-02-28'),  
(5, 7, 9, 6, '2024-01-31'),  
(6, 11, 1, 10, '2024-02-27'),  
(7, 13, 15, 1, '2024-04-09'),  
(8, 12, 2, 7, '2024-01-07'),  
(9, 16, 10, 2, '2024-04-14'),  
(10, 10, 15, 3, '2024-04-30'),  
(11, 15, 1, 2, '2024-01-06'),  
(12, 8, 16, 6, '2024-02-07'),  
(13, 9, 16, 1, '2024-04-23'),  
(14, 19, 15, 4, '2024-01-30'),  
(15, 17, 10, 5, '2024-02-18'),  
(16, 19, 13, 4, '2024-05-30'),  
(17, 1, 13, 10, '2024-02-05'),  
(18, 16, 8, 1, '2024-04-07'),  
(19, 8, 9, 10, '2024-06-15'),  
(20, 10, 15, 10, '2024-05-14'),  
(21, 12, 6, 4, '2024-06-04'),  
(22, 17, 5, 2, '2024-05-31'),  
(23, 7, 10, 2, '2024-03-14'),  
(24, 5, 8, 9, '2024-01-27'),  
(25, 9, 20, 7, '2024-04-29'),  
(26, 5, 12, 6, '2024-05-16'),  
(27, 12, 11, 2, '2024-05-14'),  
(28, 17, 17, 8, '2024-02-29'),  
(29, 8, 13, 2, '2024-06-07'),  
(30, 20, 19, 3, '2024-03-02'),  
(31, 11, 19, 8, '2024-05-07'),  
(32, 19, 7, 10, '2024-04-06'),  
(33, 10, 15, 8, '2024-02-02'),  
(34, 6, 3, 8, '2024-02-21'),  
(35, 13, 1, 2, '2024-01-17'),  
(36, 20, 19, 5, '2024-01-26'),  
(37, 15, 10, 4, '2024-06-13'),  
(38, 10, 3, 5, '2024-06-24'),  
(39, 18, 13, 1, '2024-03-03'),  
(40, 4, 13, 3, '2024-01-05'),  
(41, 12, 6, 1, '2024-02-01'),  
(42, 13, 5, 6, '2024-05-08'),  
(43, 5, 13, 2, '2024-04-23'),  
(44, 5, 9, 10, '2024-01-27'),  
(45, 8, 14, 2, '2024-03-04'),  
(46, 14, 10, 2, '2024-04-01'),  
(47, 2, 7, 5, '2024-03-13'),  
(48, 11, 2, 7, '2024-06-08'),  
(49, 20, 8, 7, '2024-01-09'),  
(50, 14, 14, 2, '2024-03-29');

## 3. Ejercicios

### EXPLAIN básico

EXPLAIN SELECT \* FROM ventas WHERE id\_cliente = 10;  
  
CREATE INDEX idx\_cliente ON ventas(id\_cliente);  
  
EXPLAIN SELECT \* FROM ventas WHERE id\_cliente = 10;

### Cuello de botella

SELECT \* FROM ventas WHERE fecha BETWEEN '2024-01-01' AND '2024-01-31';  
  
CREATE INDEX idx\_fecha ON ventas(fecha);  
  
EXPLAIN SELECT \* FROM ventas WHERE fecha BETWEEN '2024-01-01' AND '2024-01-31';

### Subconsulta vs JOIN

SELECT nombre, ciudad  
FROM clientes  
WHERE id\_cliente IN (  
 SELECT id\_cliente  
 FROM ventas  
 WHERE cantidad > 10  
);  
  
SELECT DISTINCT c.nombre, c.ciudad  
FROM clientes c  
JOIN ventas v ON c.id\_cliente = v.id\_cliente  
WHERE v.cantidad > 10;

### Join múltiple

SELECT c.nombre, p.nombre AS producto, SUM(v.cantidad) AS total\_vendido  
FROM ventas v  
JOIN clientes c ON v.id\_cliente = c.id\_cliente  
JOIN productos p ON v.id\_producto = p.id\_producto  
GROUP BY c.nombre, p.nombre  
ORDER BY total\_vendido DESC  
LIMIT 10;

## 4. Consignas para Entregar

Cada grupo deberá entregar un informe en el que se incluyan las siguientes consignas:

1. Ejecutar cada consulta con EXPLAIN y pegar las capturas de pantalla de los planes de ejecución.

2. Comparar el rendimiento de las consultas antes y después de aplicar índices, explicando las diferencias observadas.

3. Reescribir las subconsultas propuestas como joins y mostrar los resultados obtenidos.

4. Identificar posibles cuellos de botella en las consultas dadas y proponer soluciones.

5. Resolver la consulta con múltiples joins y explicar qué información aporta al análisis de ventas.

6. Concluir con una reflexión sobre la importancia de optimizar consultas en bases de datos grandes.