

Tabla 1: **clientes**

```
CREATE TABLE clientes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    email VARCHAR(100),  
    fecha_registro DATE,  
    ciudad VARCHAR(50)  
);  
  
INSERT INTO clientes (nombre, email, fecha_registro, ciudad) VALUES  
( 'Lucía Gómez', 'lucia@gmail.com', '2023-01-10', 'Buenos Aires'),  
( 'Pedro López', 'pedro@yahoo.com', '2022-11-05', 'Córdoba'),  
( 'Ana Torres', 'ana@hotmail.com', '2024-03-22', 'Mendoza'),  
( 'Carlos Ruiz', 'cruiz@gmail.com', '2023-06-15', 'Rosario'),  
( 'María Díaz', 'maria.diaz@live.com', '2022-01-01', 'Buenos Aires');
```

Tabla 2: **compras**

```
CREATE TABLE compras (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    cliente_id INT,  
    fecha_compra DATE,  
    monto DECIMAL(10,2),  
    producto VARCHAR(100),  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);  
  
INSERT INTO compras (cliente_id, fecha_compra, monto, producto)  
VALUES  
(1, '2024-01-01', 5000.00, 'Notebook'),  
(2, '2024-02-15', 1200.00, 'Auriculares'),  
(1, '2024-03-03', 1500.00, 'Teclado'),  
(3, '2023-12-10', 2000.00, 'Mouse'),  
(4, '2024-01-22', 2500.00, 'Monitor'),  
(5, '2023-11-20', 800.00, 'Cámara Web');
```

Actividades de Optimización

1. Índice por correo

Crea un índice para mejorar la búsqueda por correo electrónico en la tabla **clientes**.

2. Eliminar SELECT *

Reescribe una consulta que obtenga solo el **nombre** y **email** de los clientes de Buenos Aires.

3. Optimizar JOIN

Consulta todos los productos comprados por clientes de Córdoba. Asegurate de que el **JOIN** use índices.

4. EXPLAIN y mejora

Analiza esta consulta con **EXPLAIN**:

```
SELECT * FROM compras WHERE monto > 1000;
```

5. Normalización de productos

Actualmente los nombres de productos están repetidos en texto. Crea una tabla **productos** y normalizá esa columna en **compras**.

6. Consulta sin funciones en WHERE

Evitá usar funciones como **YEAR()** para obtener compras del año 2024. Reescribí esta consulta:

```
SELECT * FROM compras WHERE YEAR(fecha_compra) = 2024;
```

7. Pagar resultados de compras

Mostrá 3 resultados de la tabla **compras** a partir del tercer registro.

8. Índice para ordenar por fecha

Si necesitás mostrar los clientes por fecha de registro descendente, ¿qué índice podrías crear?

9. Buscar clientes con compras mayores a 2000 sin subconsulta

Reescribí esta consulta:

```
SELECT nombre FROM clientes  
WHERE id IN (SELECT cliente_id FROM compras WHERE monto > 2000);
```

Usá **JOIN**.

10. Medir rendimiento

Ejecutá esta consulta dos veces y compará el tiempo con y sin caché:

```
SELECT * FROM compras WHERE producto = 'Notebook';
```