M4_L2 Ejercicios de aplicación #2 Optimización de consultas a una rdbms

1. Identificación de Problemas en las Consultas SQL

Posibles causas de lentitud:

- Falta de índices en columnas utilizadas en filtros (WHERE) o ordenamientos (ORDER BY)
- Uso excesivo de SELECT * en lugar de columnas específicas
- Joins innecesarios o mal estructurados
- Filtros poco selectivos que devuelven demasiadas filas
- Funciones en columnas indexadas (ej. WHERE YEAR(fecha_creacion) = 2023)

SELECT * FROM envios WHERE estado = 'pendiente' AND YEAR(fecha_creacion) = 2023;

2. Mejora del Diseño del Esquema e Índices

Recomendaciones:

- Crear índices en columnas utilizadas en filtros frecuentes:

  CREATE INDEX idx_estado ON envios(estado); CREATE INDEX idx_fecha ON envios(fecha_creacion); CREATE INDEX idx_cliente ON envios(id_cliente);

- Evitar funciones sobre columnas indexadas en las consultas
- Normalizar si hay redundancia, o desnormalizar si se requiere velocidad de lectura

Esquema optimizado:

CREATE TABLE envios (id_envio INT PRIMARY KEY, id_cliente INT, fecha_creacion DATE, estado VARCHAR(20), INDEX idx_estado (estado), INDEX idx_fecha (fecha_creacion), INDEX idx_cliente (id_cliente));

3. Optimización de Consultas

Versión optimizada del ejemplo anterior:

SELECT id_envio, id_cliente, fecha_creacion FROM envios WHERE estado = 'pendiente' AND fecha_creacion BETWEEN '2023-01-01' AND '2023-12-31';

Buenas prácticas:

- Seleccionar solo las columnas necesarias
- Usar rangos de fechas en vez de funciones
- Evitar subconsultas innecesarias

- Usar EXPLAIN para entender cómo se ejecuta la consulta

4. Análisis del Plan de Ejecución

Antes de optimizar:

EXPLAIN SELECT * FROM envios WHERE estado = 'pendiente' AND YEAR(fecha_creacion) = 2023;

Después de optimizar:

EXPLAIN SELECT id_envio, id_cliente FROM envios WHERE estado = 'pendiente' AND fecha_creacion BETWEEN '2023-01-01' AND '2023-12-31';

- Resultado esperado: Using index condition