ETAPA 2 - CARGA DE DATOS MASIVOS

Descripción conceptual del mecanismo elegido

Para la carga masiva de datos del modelo **Pedido–Envío**, optamos por un mecanismo basado en **tablas semilla** y una **generación controlada de valores aleatorios**.

Uso de tablas semilla:
Se crearon tablas temporales (`seed_empresas`, `seed_tipos`, `seed_estados_envio`, `seed_estados_pedido`) con valores fijos que representan los conjuntos válidos para los campos de tipo `ENUM`. Estas tablas funcionaron como tablas auxiliares, evitando inconsistencias o errores de tipeo en los datos generados. Además, al estar en memoria (`TEMPORARY TABLE`), su uso no impacta en el almacenamiento ni interfiere con las tablas reales del esquema.

Generación de identificadores numéricos:
La tabla `seed_numeros` se utilizó como base para crear series de números consecutivos, que sirvieron para generar múltiples bloques de inserciones. Esta estructura permitió dividir la carga en secciones de 50.000 registros, mejorando el rendimiento y reduciendo el riesgo de bloqueos por operaciones muy grandes.

Distribución de valores y proporciones:
Los campos aleatorios (`empresa`, `tipo`, `estado`, `fechaDespacho`, `total`, etc.) se asignaron usando `ORDER BY RAND()` y funciones de distribución uniforme (`RAND() * rango`). Esto asegura una mezcla equilibrada de combinaciones entre las distintas categorías, simulando datos realistas y sin desequilibrios.
En el caso de las fechas, se utilizó una distribución dentro del rango del año 2025 (`DATE_ADD('2025-01-01', INTERVAL RAND()*270 DAY)`), garantizando coherencia cronológica entre `fechaDespacho` y `fechaEstimada`.

Garantía de integridad y cardinalidades:

- La relación **1:1 entre Envío y Pedido** se aseguró utilizando el campo `envio` como clave foránea (`FOREIGN KEY (envio) REFERENCES Envio(id)`), vinculando cada pedido con un único envío existente.
- Para evitar errores por restricciones `NOT NULL` y `UNIQUE` durante la carga masiva, se permitió temporalmente que el campo `numero` en `Pedido` acepte `NULL`. Luego se actualizó con un identificador único (`PED000001`, `PED000002`, …) y se restauró la restricción `NOT NULL UNIQUE`.
- Las verificaciones finales (`SELECT COUNT(*) …` y comparación de claves foráneas) permitieron confirmar la consistencia entre ambas tablas y la ausencia de pedidos sin envío.

Decisiones de eficiencia:
Se concentraron las actualizaciones (`UPDATE`) al final para evitar **procesos innecesarios que ralentizan la ejecución**.Además, se prefirió insertar todos los pedidos en una sola operación, utilizando directamente los `id` existentes de Envío, lo que garantiza una correspondencia directa y evita duplicados.