Creación de una base de datos de Staging

Presentado

Jhon Neider Cuervo Quintero

Yuri Marcela Cuervo Quintero

Grupo #5

Formación

ING software y datos

Materia

Base De Datos II - PREICA2502B010064

Docente

Antonio Jesús Valderrama

16/09/2025

Resumen

Este documento describe el análisis de los datos almacenados en la base de datos **Jardinería**, el diseño de una base de datos *staging* (temporal) para soportar procesos ETL, los scripts SQL para crear la base de datos y tablas de *staging*, las consultas ETL para cargar los datos desde jardineria hacia staging_jardineria, la validación de la carga y los pasos para generar copias de seguridad (BK) de ambas bases de datos. Se incluye, además, un anexo con todos los scripts listos para ejecutar en **MySQL Workbench**.

1. Introducción

En entornos de integración de datos, una capa *staging* se usa para alojar copias limpias y homogéneas de los datos fuente antes de la transformación final y la carga en los modelos analíticos o de producción. Este documento guía paso a paso la construcción de una base de datos *staging* para la base de datos jardineria, con instrucciones para ejecutar todo en **MySQL Workbench**.

2. Objetivos

Objetivo general

Diseñar e implementar una base de datos *staging* que permita centralizar y homologar datos provenientes de la base jardineria, facilitando procesos ETL y garantizando trazabilidad.

Objetivos específicos

Analizar las tablas y columnas de jardineria para determinar qué trasladar al *staging*.

Construir la estructura de tablas *staging* (DDL).

Construir las consultas ETL para mover los datos (con transformaciones básicas y cálculo de hash).

Ejecutar las consultas, validar los resultados y generar copias de seguridad de ambas bases.

Documentar el proceso en formato APA para entrega.

3. Planteamiento del problema

La base de datos jardinería contiene datos operacionales (clientes, oficinas, empleados, productos, pedidos, detalle de pedidos, etc.). El reto consiste en obtener una copia controlada de esos datos en una base *staging* que permita:

Cargar datos de forma idempotente (mismos registros no se dupliquen).

Registrar metadatos de carga (fecha, lote, hash) para detectar cambios.

Realizar limpieza básica (trim, normalización) y facilitar validaciones.

4. Análisis del problema / Análisis de datos Criterios

para mover tablas a *staging*:

Tablas transaccionales y de maestro que se usan en reportes o procesos ETL (ej.: oficina, empleado, cliente, producto, pedido, detalle_pedido).

Columnas necesarias: identificadores PK, campos de negocio, y columnas que requieran limpieza (direcciones, teléfonos, formatos de fecha).

No es estrictamente necesario incluir índices/foráneas en *staging* (mejor rendimiento) — dejarlos para la capa destino si se requiere.

Ejemplo (tabla oficina)

Según el script previo que usted tiene en el entorno (CREATE TABLE oficina (...)), movemos las columnas principales: ID_oficina, Descripcion, ciudad, pais, region, codigo_postal, telefono, linea_direccion1, linea_direccion2.

Decisión de diseño: Para cada tabla *staging* mantendremos los campos originales + columnas ETL: stg_batch_id, stg_load_date, stg_source_system, stg_record_hash, stg_is_deleted.

5. Propuesta de solución

Correcciones a la entrega 1

El diseño de nuestra primera entrega fue robusto y no requirió cambios significativos, pero en esta segunda entrega se mejoró mucho más, aunque no se necesitara.

Descripción del análisis realizado a los datos Jardinería y cómo estos se trasladaron a la base de datos Staging.

El proceso de desarrollo se inició con un análisis exhaustivo de la estructura de la base de datos operacional jardineria. Este análisis no solo se centró en la identificación de las tablas y sus columnas, sino también en las relaciones existentes entre ellas (por ejemplo, cómo los pedidos están vinculados a los clientes y los empleados). El modelo de entidad-relación (EER) se utilizó como guía principal durante este proceso.

Una vez que se comprendió la estructura de origen, se procedió a diseñar la base de datos staging. Esta etapa fue crucial para preparar los datos para futuros procesos analíticos, adhiriéndose a las siguientes decisiones de diseño:

Modelo Staging: Se decidió replicar la estructura de la base de datos jardineria en el esquema staging_jardineria, pero con un enfoque en la simplicidad. Se eliminaron las claves foráneas (FK) para optimizar el rendimiento de las cargas masivas.

Convenciones de Nombres: Para diferenciar claramente los datos de origen de los datos staging, se aplicó el prefijo stg_ a todas las tablas del nuevo esquema.

Columnas de Metadatos: Se agregaron campos específicos a cada tabla para registrar metadatos clave del proceso ETL:

stg_batch_id: Identificador del lote de carga. stg_load_date:

Fecha y hora exactas de la carga. stg_source_system: Sistema

de origen de los datos (jardineria). stg_record_hash: Un hash

único generado a partir de la combinación de los valores de las

columnas del registro. Este campo es vital para la

idempotencia de la carga, permitiendo detectar y evitar la

duplicación de registros.

stg_is_deleted: Indicador para marcar registros que han sido eliminados en la fuente.

Finalmente, la transferencia de datos se llevó a cabo utilizando scripts de carga ETL (staging_etl_loads.sql) con sentencias SQL INSERT INTO... SELECT FROM.... Este método permitió extraer los datos directamente de la base de datos jardineria, realizar transformaciones mínimas (como el cálculo del hash) y cargarlos de manera controlada y trazable en la base de datos staging_jardineria. El archivo comparacion_staging.sql se utilizó para validar que el número de registros cargados en el destino coincidiera con el número de registros en el origen, garantizando la integridad de la transferencia de datos.

ANEXOS / LINK DRIVE / MySQL

Sql – Backups - Docs

https://drive.google.com/drive/folders/1tlju1ejfpbISoisudY6oMEUMo0odbcJS?usp=sharing

BIBLIOGRAFIAS

Hernández, R. (2018). *SQL Práctico: Conociendo los Fundamentos del Lenguaje de Consulta Estructurado*. Grupo Editorial Éxodo.

Padron, J. (2019). *Diseño de bases de datos relacionales*. Editorial Ecoe.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.

MySQL. (s. f.). *MySQL Documentation*. Recuperado de https://dev.mysql.com/doc/

Microsoft. (s. f.). *SQL Server Documentation*. Recuperado de https://learn.microsoft.com/es-es/sql/sql-server/sql-server-documentation