

INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA  
Bases de Datos II

Profesor: Víctor Hugo Mercado  
Alumno: Johan Sebastián Zamudio Cortes

Ingeniería de desarrollo de software y datos  
2025

## Contenido

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>OBJETIVO GENERAL.....</b>	<b>4</b>
<b>OBJETIVOS ESPECIFICOS .....</b>	<b>5</b>
<b>Datos a ser trasladados a la nueva base de datos.....</b>	<b>6</b>
<b>TECNICA UTILIZADA.....</b>	<b>6</b>
<b>TECNICAS DE TRANSFORMACIÓN DE DATOS .....</b>	<b>7</b>
1.1 Limpieza de datos:.....	7
1.2 Normalización de datos: .....	7
1.3 Validación de integridad:.....	7
<b>TECNICAS DE ETL APLICO CON SQL.....</b>	<b>7</b>
<b>WEBGRAFIAS .....</b>	<b>9</b>

## INTRODUCCIÓN

A través de este ejercicio de manipulación de cadenas, comprenderemos cómo llevar a cabo una migración de bases de datos.

Para eso debemos realizar una migración exitosa a una base de datos nueva distinta, específicamente a la base llamada stating, la cual será la nueva base de datos que contendrá los datos de la base anterior llamada jardinería

Por último, deberemos documentar todo el proceso para llevar a cabo este procedimiento, así como generar una copia de las bases de datos.

## **OBJETIVO GENERAL**

Debemos realizar una migración efectiva de la base de datos jardinería a la nueva base llamada stating, asegurando que la integridad, consistencia y optimización de los datos durante el proceso, identificando los pasos a seguir, así como los datos a trasladar a la nueva base de datos

## **OBJETIVOS ESPECIFICOS**

Es fundamental reconocer los pasos esenciales para trasladar datos de una base de datos a otra de manera eficiente y sin riesgo de pérdida de información.

Se debe realizar la transferencia de datos desde la base de jardinería a la nueva base de datos de manera que se garantice la integridad de la misma

Es necesario aplicar las modificaciones pertinentes a los datos, asegurando así su compatibilidad con la nueva estructura.

Se deben realizar pruebas para confirmar que la información migrada sea precisa y completa en la nueva base de datos.

Listas para análisis o carga al Data Warehouse sin errores de integridad ni datos inválidos o incompleto

La base de datos final (Data Mart) tiene datos limpios, normalizados y consistentes.

## Datos a ser trasladados a la nueva base de datos

Se utilizan comandos de extracción de datos para poder incluirlos en la nueva tabla string de sql basándonos en el modelo de estrella



Los datos son extraídos de la base de origen llamada jardinería y se crea una base llamada Staging donde irán los datos del modelo de estrella

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> stg_categoria_producto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> stg_cliente	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	34	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> stg_detalle_pedido	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> stg_pago	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	24	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> stg_pedido	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	44	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> stg_producto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
6 tabla(s)	Número de filas	107	InnoDB	utf8mb4_general_ci	176 KB	0 B

## TECNICA UTILIZADA

Utilizamos consultas **SQL (INSERT INTO ... SELECT ...)** que tomen los datos de la base de datos original en este caso jardinerías para que los copiemos en las tablas equivalentes de la base de datos **Staging**.

Ejemplo

```
INSERT INTO staging.stg_producto (ID_producto, nombre, ID_categoria, dimensiones, proveedor, descripcion, cantidad_en_stock, precio_venta, precio_proveedor)
SELECT ID_producto, nombre, ID_categoria, dimensiones, proveedor, descripcion, cantidad_en_stock, precio_venta, precio_proveedor
FROM jardineria.producto;
```

Según el modelo de estrella que se creó anteriormente extraemos y creamos una nueva base de datos, que en este caso llamamos Staging donde irán los nuevos datos a extraer

Después de validar los datos podemos concluir que

- Los datos no tienen errores (campos vacíos donde no debe haberlos, valores inválidos).
- Se mantenga la integridad referencial (las llaves foráneas apuntan a datos válidos).
- Los datos estén consistentes y normalizados para formar las tablas de Dimensiones Hechos del modelo Estrella (estructura común de Data Warehousing).

## TECNICAS DE TRANSFORMACIÓN DE DATOS

Se utilizar un script que nos permitirá eliminar registros incompletos o inconsistentes para poder llevar acabo una limpieza de los datos.

¿Qué hace el script?

### 1.1 Limpieza de datos:

- Quita espacios sobrantes en textos de clientes y direcciones.
- Elimina clientes sin nombre o teléfono incompletos.
- Completa valores nulos en la región.
- Elimina productos con nombre nulo o categoría no inválida.
- Elimina pagos con valores negativos o no válidos.

### 1.2 Normalización de datos:

- Establece un formato limpio en los números de contacto
- Iguala el estado de los pedidos, dejando solo los valores válidos.

### 1.3 Validación de integridad:

- Elimina pagos cuyos clientes no existan o los datos estén incompletos
- Valida los stock y precios en productos, asegurando valores de forma coherente.

## TECNICAS DE ETL APLICADO CON SQL

Se utiliza SQL como técnica ETL

Staging Layer lista para cargar en un Data Warehouse o análisis avanzado

ETL	Técnicas	Scripts
Extract	Usas datos de las tablas stg_cliente, stg_producto, stg_pedido, entre otros	FROM stg_cliente, FROM stg_producto
Transform	<ul style="list-style-type: none"> <li>- Limpieza (TRIM, eliminación de datos nulos)</li> <li>- Normalización (mayúsculas, formateo)</li> <li>- Enriquecimiento (crear columnas nuevas con cálculos o concatenaciones)</li> </ul>	TRIM(nombre_cliente) UPPER(nombre_cliente) margen_ganancia = precio_venta precio_proveedor
Load	Actualizas y carga los datos limpios en las tablas destino (Staging)	UPDATE stg_cliente SET DELETE FROM stg_pago WHERE total <= 0

**Cargar los datos de la tabla producto a Staging:** Copia los datos de la tabla producto de la base de datos jardinería a la tabla de staging stg\_producto.( esto solo se puede hacer una vez, ya que las llaves primarias no lo permiten)

```
INSERT INTO staging.stg_producto (ID_producto, nombre, ID_categoria, dimensiones, proveedor, descripcion, cantidad_en_stock, precio_venta, precio_proveedor)  
SELECT ID_producto, nombre, ID_categoria, dimensiones, proveedor, descripcion, cantidad_en_stock, precio_venta, precio_proveedor  
FROM jardineria.producto;
```

**Normalización del nombre del cliente:** Convierte a mayúsculas el campo nombre\_cliente en la tabla stg\_cliente, esto ayuda a estandarizar los datos

```
UPDATE staging.stg_cliente  
SET nombre_cliente = UPPER(nombre_cliente)  
WHERE nombre_cliente IS NOT NULL;
```

**Eliminación de registros inválidos:** Elimina los registros de la tabla stg\_cliente donde no exista un ID\_cliente, esto garantiza la integridad de los datos, evitando registros incompletos.

```
DELETE FROM staging.stg_cliente  
WHERE ID_cliente IS NULL;
```

**Agregar una nueva columna para el margen:** Se utilizará para calcular la ganancia de cada producto

```
ALTER TABLE staging.stg_producto ADD COLUMN margen DECIMAL(10,2);
```

**Calcular el margen de cada producto:** Esto permite conocer cuánto se gana por cada producto.

```
UPDATE staging.stg_producto  
SET margen = precio_venta - precio_proveedor;
```

**Cargar los datos transformados al Data Mart:** Se cargan solo los campos necesarios para el análisis, incluyendo el margen calculado.

```
INSERT INTO datamart.dm_producto (ID_producto, nombre, ID_categoria, precio_venta, margen)  
SELECT ID_producto, nombre, ID_categoria, precio_venta, margen  
FROM staging.stg_producto;
```

## CARGA DE REGISTROS EN EL DATA MART FINAL

En esta etapa, se elaboran las consultas o scripts necesarios para extraer los datos que han sido debidamente limpiados y transformados de la base de datos Staging, con el objetivo de cargarlos en la base final

En este momento, los datos en Staging han sido sometidos a procesos de limpieza,



normalización y validación, lo que los hace aptos para alimentar la nueva base de datos

¿Qué se hace en este paso?

Se preparan consultas INSERT INTO o scripts ETL.

- Estas consultas toman los datos de las tablas y los cargan en las tablas de Data Mart.
- Aquí ya no solo se copian los datos, sino que se insertan solo los datos que cumplen las reglas del modelo de análisis, por ejemplo: fechas completas, sin nulos, claves válidas
- Es el último paso de la carga de datos antes de que la información quede lista para los reportes y análisis.

Prueba de carga de datos

```
Mostrar ventana de consultas SQL

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

-- Crear base de datos Staging CREATE DATABASE IF NOT EXISTS staging;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Note: #1007 No puedo crear base de datos 'staging', la base de datos ya existe

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

USE staging;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Error: #1046 Base de datos no seleccionada

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

-- Crear tabla stg_cliente CREATE TABLE IF NOT EXISTS stg_cliente ( ID_cliente INT PRIMARY KEY, nombre_cliente VARCHAR(100), nombre_contacto VARCHAR(100), apellido_contacto VARCHAR(100), telefono VARCHAR(20), fax VARCHAR(20), linea_direccion1 VARCHAR(150), linea_direccion2 VARCHAR(150), ciudad VARCHAR(50), region VARCHAR(50), pais VARCHAR(50), codigo_postal VARCHAR(20), ID_empleado_rep_ventas INT, limite_credito DECIMAL(10,2), nombre_completo VARCHAR(200), categoria_cliente VARCHAR(20) );

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Note: #1050 La tabla 'stg_cliente' ya existe

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

-- Crear tabla stg_producto CREATE TABLE IF NOT EXISTS stg_producto ( ID_producto INT PRIMARY KEY, nombre VARCHAR(100), ID_categoria INT, dimensiones VARCHAR(50), proveedor VARCHAR(100), descripcion TEXT, cantidad_en_stock INT, precio_venta DECIMAL(10,2), precio_proveedor DECIMAL(10,2) );

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Note: #1050 La tabla 'stg_producto' ya existe

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

-- Crear tabla stg_pedido CREATE TABLE IF NOT EXISTS stg_pedido ( ID_pedido INT PRIMARY KEY, fecha_pedido DATE, fecha_esperada DATE, fecha_entrega DATE, estado VARCHAR(50), comentarios TEXT, ID_cliente INT );
```

## WEBGRAFÍAS

- Jhon Goyeneche. (2022, 4 mayo). Data warehouse con Mysql [Vídeo]. YouTube. Evidencia de aprendizaje 3

<https://www.youtube.com/watch?v=Jt5QieHNE7o>

- ¿Qué es ETL? - Explicación de extracción, transformación y carga (ETL) - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/etl/>