

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

INTRODUCCIÓN	3
OBJETIVO GENERAL	4
OBJETIVOS ESPECIFICOS	5
PLANTEAMIENTO DEL PROBLEMA	6
ANÁLISIS DEL PROBLEMA	6
Diseño de estructura Staging	7
Bibliografía.....	10

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 acabo 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.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la base de datos de jardinería cuenta con información crucial que necesitamos trasladar a una nueva base llamada stating, para estos es fundamental que esta migración se lleve a cabo de manera eficiente, garantizando que los datos sean transferidos sin pérdidas, errores o inconsistencias que podrían comprometer su integridad.

Para esto es fundamental desarrollar un ejercicio que nos permita comprender las técnicas y buenas prácticas para realizar una migración de bases de datos exitosa, sin comprometer la integridad de los datos a trasladar a la nueva base de datos

ANÁLISIS DEL PROBLEMA

Es fundamental analizar si ambas bases tienen el mismo esquema o si es necesario realizar ajustes en la estructura de stating para que los datos sean compatibles, ya que de no ser así no podríamos realizar una migración exitosa a la nueva base de datos.

La base de datos original de jardinería no puede ser transferida, es necesario reestructurarla, ya que los datos deben ser auto incrementables para poder ser transferidos

Datos a ser trasladados a la nueva base de datos

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> categoria_producto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> cliente	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	34	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> detalle_pedido	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> empleado	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	31	InnoDB	utf8mb4_general_ci	64.0 KB	-
<input type="checkbox"/> oficina	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> pago	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	24	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> pedido	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	44	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> producto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	11	InnoDB	utf8mb4_general_ci	32.0 KB	-
8 tablas		Número de filas		164 InnoDB utf8mb4_general_ci		240.0 KB
						0 B

Para poder definir cuales son los datos que debemos identificar cuales son aquellos datos cruciales que son requeridos para la migración de los datos, estos se clasifican por:

1. Información esencial
2. Datos relacionados con clientes, pedidos y productos,
3. Información histórica útil
4. Evitar datos redundantes

Tabla	Campos Seleccionados
cliente	ID_cliente, nombre_cliente, telefono, pais, ID_empleado_rep_ventas, limite_credito
producto	ID_producto, nombre_producto, ID_categoria, precio_unidad, cantidad_en_stoc
pedido	ID_pedido, fecha_pedido, ID_cliente, estado
detalle_pedido	ID_pedido, ID_producto, cantidad, precio_unidad
pago	ID_cliente, forma_pago, fecha_pago, total_pago

Cientes: Contiene la información de los compradores, es importante tener encuesta esta información para ser trasladada

Productos: Información relevantes de los productos vendidos por la empresa

Pedido: Solicitud de los productos que los clientes realizan en la plataforma

Detalle pedido: Información de envió de los productos, así como instrucciones de cómo se debe realizar la entrega

Pago: Métodos de pago utilizado por el comprador

Categoría producto: Muestra el tipo de producto del catalogo de ventas

Diseño de estructura Staging

Tablas cliente: Guarda los datos esenciales de los clientes.

```
1 CREATE TABLE stg_cliente (  
2     ID_cliente          INT PRIMARY KEY,  
3     nombre_cliente      VARCHAR(100) NOT NULL,  
4     telefono            VARCHAR(20),  
5     pais                VARCHAR(50),  
6     ID_empleado_rep_ventas INT,  
7     limite_credito      DECIMAL(10,2)  
8 );  
9
```

Tabla stg_productos: Registra los productos y su categoría.

```
1 CREATE TABLE stg_producto (  
2     ID_producto      INT PRIMARY KEY,  
3     nombre_producto  VARCHAR(100) NOT NULL,  
4     ID_categoria     INT,  
5     precio_unidad    DECIMAL(10,2),  
6     cantidad_en_stock INT  
7 );  
8
```

Tablas Stg_pedido: Contiene información general de los pedidos.

```
1 CREATE TABLE stg_pedido (  
2     ID_pedido      INT PRIMARY KEY,  
3     fecha_pedido   DATE NOT NULL,  
4     ID_cliente     INT,  
5     estado         VARCHAR(50),  
6     FOREIGN KEY (ID_cliente) REFERENCES stg_cliente(ID_cliente)  
7 );
```

Tabla detalle_pedido: Registra los productos incluidos en cada pedido.

```
1 CREATE TABLE stg_detalle_pedido (
2     ID_pedido      INT,
3     ID_producto    INT,
4     cantidad       INT,
5     precio_unidad  DECIMAL(10,2),
6     PRIMARY KEY (ID_pedido, ID_producto),
7     FOREIGN KEY (ID_pedido) REFERENCES stg_pedido(ID_pedido),
8     FOREIGN KEY (ID_producto) REFERENCES stg_producto(ID_producto)
9 );
```

Tabla stg_pago: Guarda los pagos de los clientes.

```
1 CREATE TABLE stg_pago (
2     ID_pago        INT PRIMARY KEY AUTO_INCREMENT,
3     ID_cliente     INT,
4     forma_pago     VARCHAR(50),
5     fecha_pago     DATE,
6     total_pago     DECIMAL(10,2),
7     FOREIGN KEY (ID_cliente) REFERENCES stg_cliente(ID_cliente)
8 );
```

CONSTRUCCIÓN DE DATOS DE MIGRACIÓN

1. Clientes: Trae todos los datos de la base de clientes a la nueva base de datos

```
1 INSERT INTO staging.stg_cliente (
2     ID_cliente, nombre_cliente, nombre_contacto, apellido_contacto, telefono, fax,
3     linea_direccion1, linea_direccion2, ciudad, region, pais, codigo_postal,
4     ID_empleado_rep_ventas, limite_credito
5 )
6 SELECT
7     ID_cliente, nombre_cliente, nombre_contacto, apellido_contacto, telefono, fax,
8     linea_direccion1, linea_direccion2, ciudad, region, pais, codigo_postal,
9     ID_empleado_rep_ventas, limite_credito
10 FROM jardineria.cliente;
```

2. Productos: Trae todos los elementos de productos a la nueva base de datos

```
1 INSERT INTO staging.stg_producto (
2     ID_producto, nombre, ID_categoria, cantidad_por_unidad,
3     precio_unitario, unidades_en_stock, unidades_en_pedido,
4     nivel_reposicion, suspendido
5 )
6 SELECT
7     ID_producto, nombre, ID_categoria, cantidad_por_unidad,
8     precio_unitario, unidades_en_stock, unidades_en_pedido,
9     nivel_reposicion, suspendido
10 FROM jardineria.producto;
```


3. Pedido: Trae todos los elementos de pedidos a la nueva base de datos

```
1 INSERT INTO Staging.stg_pedido (ID_pedido, fecha_pedido, fecha_esperada, fecha_entrega, estado, comentarios, ID_cliente)
2 SELECT ID_pedido, fecha_pedido, fecha_esperada, fecha_entrega, estado, comentarios, ID_cliente
3 FROM Jardineria.pedido;
```

4. Detalle pedido: Trae todos los datos de los pedidos a la nueva base de datos

```
1 INSERT INTO Staging.detalle_pedido (ID_pedido, ID_producto, cantidad, precio_unidad)
2 SELECT ID_pedido, ID_producto, cantidad, precio_unidad
3 FROM Jardineria.detalle_pedido;
```

5. Pagos: Incluye los datos de los pagos de la antigua base de datos y los trae a la nueva base de datos

```
1 INSERT INTO Staging.stg_pago (ID_cliente, forma_pago, id_transaccion, fecha_pago, total)
2 SELECT ID_cliente, forma_pago, id_transaccion, fecha_pago, total
3 FROM Jardineria.pago;
```

6. Categoría producto:

```
1 INSERT INTO staging.stg_categoria_producto (ID_categoria, desc_categoria, descripcion_texto, descripcion_html, imagen)
2 SELECT ID_categoria, desc_categoria, descripcion_texto, descripcion_html, imagen
3 FROM Jardineria.categoria_producto;
```

Bibliografía.

- Calzada, J. M. (2020, 16 noviembre). *Seis razones por las que usar staging*. Consultoria Certia. Formación | Consultoria | Desarrollo.
<https://www.certia.net/seis-razones-por-las-que-usar-staging/#:~:text=Una%20base%20de%20datos%20de,que%20contenga%20metadatos%20m%C3%A1s%20detallados>.
- *Creación de una base de datos de Staging - Google Search*. (s. f.).
https://www.google.com/search?sca_esv=e86fb0d398013f57&sxsrf=AHTn8z0j-6oanmv8RJgPICCECGoTXq4QFA:1741727618498&q=Creaci%C3%B3n+de+una+base+de+datos+de+Staging&udm=7&fbs=ABzOT_BnMAgCWdhr5zilP5f1cnRvK9uZj3HA_MTJAA6IXR8yQIHhBi298nC38CQZOY2HEJbzdXfJmMbf5S48k7ONt_PRS00VGqm6lbkjj7xM3Lv-u4CDP4OaWf9g_5ET3ktUPLY3PO3mFe136kqXnT8OVvjj-T38A8xdaD2lk7jGC1IIC8wl3dLa4ailaZZ-2DhixW5PkB6GFCYWjPM0Bmp23X9zsHaNQ&sa=X&ved=2ahUKEwjPy_v_-IKMAxV1SzABHcG7BB4QtKgLegQIExAB&biw=1920&bih=951&dpr=1#fpstate=ive&vld=cid:02124b4c,vid:783w-4f_iek,st:0