

SESIÓN 11: EJERCICIO BACKUP

Practicar backup, restauración completa y parcial y recuperación de datos usando phpMyAdmin y un poco de SQL.

Paso 1: Crear la base de datos de trabajo

```
CREATE DATABASE tienda_online;
```

Paso 2: Crear tablas y datos de ejemplo

```
-- Usar la base de datos
```

```
USE tienda_online;
```

```
-- Tabla de usuarios
```

```
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    fecha_registro DATE NOT NULL
);
```

```
-- Tabla de productos
```

```
CREATE TABLE productos (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    precio DECIMAL(10,2) NOT NULL,  
    stock INT NOT NULL  
);
```

-- Tabla de pedidos

```
CREATE TABLE pedidos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    fecha_pedido DATE NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
    FOREIGN KEY (id_producto) REFERENCES productos(id)
);
```

-- Insertar usuarios

```
INSERT INTO usuarios (nombre, email, fecha_registro) VALUES
('Ana Pérez', 'ana@example.com', '2024-01-10'),
('Luis Gómez', 'luis@example.com', '2024-02-05'),
```

```
('Marta Ruiz', 'marta@example.com', '2024-03-20');
```

```
-- Insertar productos
```

```
INSERT INTO productos (nombre, precio, stock) VALUES
('Teclado mecánico', 59.99, 50),
('Ratón inalámbrico', 24.50, 100),
('Monitor 24 pulgadas', 149.00, 30);
```

```
-- Insertar pedidos
```

```
INSERT INTO pedidos (id_usuario, id_producto, cantidad, fecha_pedido)
VALUES
(1, 1, 1, '2024-04-01'), -- Ana compra un teclado
(2, 2, 2, '2024-04-02'), -- Luis compra 2 ratones
(3, 3, 1, '2024-04-03'); -- Marta compra un monitor
```

Paso 3 – Hacer un BACKUP COMPLETO con phpMyAdmin

Vamos a simular que somos responsables de continuidad del servicio y hacemos una copia antes de “tocar nada”.

3.1. Exportación completa (modo recomendado)

1. En phpMyAdmin, en la columna de la izquierda, haz clic sobre la BD tienda_online.
2. Arriba, pulsa la pestaña “Exportar”.

3. Selecciona:
 - Método de exportación: Personalizado
 - Formato: SQL
4. En la sección Tablas, comprueba que todas las tablas están seleccionadas.
5. En Salida:
 - Marca “Compresión: gzip” (o zip), si está disponible.
6. En Opciones de creación de objeto:
 - Marca “Aregar instrucción DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT / TRIGGER”.
 - Marca “Aregar la instrucción IF NOT EXISTS” (si está).
7. Pulsa “Continuar” y guarda el archivo, por ejemplo:
backup_tienda_online_completo.sql
(o .sql.gz si está comprimido).

Paso 4: Simular un DESASTRE

Ahora viene la parte divertida, vamos a romper la base de datos como si alguien hubiera metido la pata.

4.1. Borrar una tabla completa (desastre grave)

1. Ve a la BD tienda_online.
2. Haz clic en la tabla pedidos.
3. Pulsa en la pestaña “Operaciones”.
4. Busca la opción “Eliminar la tabla (DROP)”.

5. Confirma.

Has borrado todos los pedidos.

4.2. Borrar filas de otra tabla (error humano típico)

1. Ve a la tabla usuarios.
2. En la pestaña “Examinar”, marca el usuario Luis Gómez.
3. Abajo, en el menú desplegable de acciones, elige “Eliminar”.
4. Confirma.

Hemos perdido los datos de Luis.

Paso 5: Recuperar la tabla borrada desde el BACKUP (Restauración parcial)

Situación:

- Tenemos un backup completo.
 - Está borrada la tabla pedidos.
 - No queremos tocar usuarios y productos (porque han podido cambiar desde el backup).
- Solo queremos recuperar pedidos.

5.1. Extraer solo la tabla pedidos del backup

- Abrir el archivo .sql con un editor de texto.
- Buscar la parte que crea e inserta datos en pedidos.
- Copiar solo ese fragmento en otro archivo pedidos_backup.sql.

5.2. Restaurar la tabla pedidos desde el backup entero en una BD temporal

5.2.1. Crear una base de datos temporal

1. Ve a “Bases de datos”.
2. Crea una nueva BD llamada:
tienda_online_backup
3. Cotejamiento: utf8mb4_general_ci.

5.2.2. Importar el backup completo en la BD temporal

1. Selecciona la BD tienda_online_backup.
2. Ve a la pestaña “Importar”.
3. Elige el archivo: backup_tienda_online_completo.sql (o .sql.gz).
4. Pulsa “Continuar”.

Ahora tienes:

- tienda_online: rota (sin pedidos y sin Luis).
- tienda_online_backup: copia intacta de antes del desastre.

5.3. Copiar la tabla pedidos desde la BD de backup

Ahora vamos a “resucitar” la tabla pedidos.

Opción A: Usar phpMyAdmin (modo clics)

1. En la columna izquierda, selecciona la BD tienda_online_backup.
2. Haz clic en la tabla pedidos.
3. Pestaña “Operaciones”.

4. Busca la sección “Copiar tabla a (base_de_datos.tabla)”.
5. En “Tabla destino” escribe:
tienda_online_pedidos
6. Marca “Estructura y datos”.
7. Pulsa “Continuar”.

Con esto has copiado la tabla pedidos (estructura + datos) desde la BD de backup a la BD rota.

Opción B: Usar una consulta SQL (modo más técnico)

En la BD tienda_online ejecuta:

```
CREATE TABLE pedidos LIKE tienda_online_backup.pedidos;
```

```
INSERT INTO pedidos
```

```
SELECT * FROM tienda_online_backup.pedidos;
```

Comprobación:

- Ve a tienda_online → tabla pedidos → pestaña “Examinar”.
- Deberías ver de nuevo los 3 pedidos iniciales.

Paso 6: Recuperar el usuario borrado usando BD temporal (recuperación de filas)

Situación:

- En tienda_online, falta el usuario "Luis Gómez".
- Pero en tienda_online_backup, todavía existe.

6.1. Comparar usuarios entre versión rota y versión de backup

En la pestaña SQL (seleccionando la BD tienda_online) ejecuta:

```
SELECT *  
  
FROM tienda_online_backup.usuarios  
  
WHERE id NOT IN (SELECT id FROM tienda_online.usuarios);
```

Esto te mostrará los usuarios que están en el backup y no en la BD actual.

Debería salir el usuario borrado (Luis).

6.2. Insertar solo el usuario que falta

Ahora ejecuta:

```
INSERT INTO tienda_online.usuarios (id, nombre, email, fecha_registro)  
  
SELECT id, nombre, email, fecha_registro  
  
FROM tienda_online_backup.usuarios  
  
WHERE id NOT IN (SELECT id FROM tienda_online.usuarios);
```

Comprueba que Luis Gómez ha vuelto a aparecer.