
UD7. Seguridad de la información y mecanismos de salvaguarda y transferencia.

ASIR - Gestión de Bases de datos

1. Seguridad de la información y mecanismos de salvaguarda y transferencia.

1.1 Tipos de copias de seguridad

Podemos establecer diferentes clasificaciones:

1. Frío – Caliente.
2. Física – Lógica.
3. Completa – Incremental.
4. Manual – Automática.

1.1.1 Frío - Caliente

- **En frío:** Este tipo de copias de seguridad se realizan parando el servicio de base de datos, para evitar que los usuarios puedan acceder a los datos mientras se realiza la copia. Tiene el inconveniente de que el servicio no estará disponible durante el tiempo que dure el proceso de copia.
- **En caliente:** En este caso no es necesario detener el servicio de base de datos, de modo que los usuarios pueden acceder a la base de datos mientras se realiza la copia de seguridad.

1.1.2 Física o Binaria - Lógica o Textual

- **Física o Binaria:** Este tipo de copias consiste en copiar desde el disco los archivos binarios que utiliza el sistema gestor de bases de datos para almacenar las bases de datos. Este tipo de copias tienen la ventaja de que son muy rápidas y se pueden realizar tanto en frío como en caliente.

La restauración de estas copias de seguridad consiste en copiar los archivos a sus ubicaciones originales. Para realizar copias físicas podemos hacer uso de comandos de copia de archivos.

- **Lógica o Textual:** Una copia lógica consiste en exportar los datos y los diferentes objetos de las bases de datos en archivos de texto. Este tipo de copias tienen el inconveniente de que son más lentas, pero tienen la ventaja de que nos permite importarlas en otros sistemas gestores de bases de datos.

La restauración de estas copias de seguridad consiste en volver a cargar el contenido de los archivos de texto en las bases de datos del servidor. Las técnicas que podemos utilizar para realizar este tipo de copias son la sentencia de SQL `SELECT ... INTO OUTFILE`, `mysqldump`, `MySQL Workbench` o `phpMyAdmin`.

1.1.3 Completa - Incremental

- **Completa:** Una copia completa es una copia de todos los datos de la base de datos (o de varias bases de datos).
- **Incremental:** Es una copia que contiene sólo los aquellos datos que han cambiado respecto a la última copia.

1.1.4 Manual - Automática

- **Manual:** Son las copias que ejecutamos nosotros de forma manual.
- **Automática:** Lo normal es que el proceso de copias de seguridad esté automatizado y que las copias de seguridad se realicen de forma automática.

1.2 ¿Dónde se guardan los datos de una base de datos MySQL?

Los datos de las bases de datos de MySQL se almacenan en el directorio:

En Windows:

`C:\ProgramData\MySQL\MySQL Server 8.0\Data`

En Linux:

`/var/lib/mysql`

Dentro de este directorio se crea un directorio para cada una de las bases de datos.

Por ejemplo, si en hemos creado una base de datos llamada `tendaBD` estará almacenada en:

`C:\ProgramData\MySQL\MySQL Server 8.0\Data\tendaBD` (en Windows)

`/var/lib/mysql/tendaBD` (En Linux)

1.3 Tipos de archivos para almacenar las BD

Los archivos que utiliza MySQL para almacenar las bases de datos **pueden tener** las siguientes extensiones (podemos encontrar otro tipo de archivos):

- `.frm`
- `.ibd`
- `.myd`
- `.myi`

1.3.1 .frm

Los archivos con extensión `.frm` contienen la estructura de las tablas. Hay tantos archivos `.frm` como tablas tenga nuestra base de datos. Estos archivos se usan para las **tablas de tipo InnoDB y MyISAM**.

1.3.2 .ibd

Los archivos con extensión `.ibd` almacenan los datos y los índices de las **tablas de tipo InnoDB**. Hay tantos archivos `.ibd` como tablas tenga nuestra base de datos.

1.3.3 .myd

Los archivos con extensión `.myd` almacenan los datos de las **tablas de tipo MyISAM**. Hay tantos archivos `.myd` como tablas tenga nuestra base de datos.

1.3.4 .myi

Los archivos con extensión `.myi` almacenan los índices de las **tablas de tipo MyISAM**. Hay tantos archivos `.myi` como tablas tenga nuestra base de datos.

1.4 Copias de seguridad lógicas o textuales

mysqldump

La utilidad `mysqldump` permite realizar copias de seguridad lógicas o textuales de una base de datos MySQL.

`mysqldump` es un comando que se encuentra (en Windows) en la siguiente carpeta:

C:\Program Files\MySQL\MySQL Server 8.0\bin (En la carpeta bin donde se haya instalado el servidor.)

Para poder ejecutar el comando, deberemos posicionarnos en la carpeta o bien agregar la ruta a la **variable path**, en variables del sistema.

Existen tres formas de usar `mysqldump`:

- 1) Exportar una o varias tablas de una base de datos,

```
mysqldump [options] db_name [tbl_name ...]
```

2) Exportar una o varias bases de datos completas,

```
mysqldump [options] --databases db_name ...
```

3) Exportar todas las bases de datos completas.

```
mysqldump [options] --all-databases
```

Referencia:

- [Documentación en la página oficial.](#)

Exportar una o varias tablas de una base de datos

Para exportar una o varias tablas de una base de datos podemos usar este comando:

```
mysqldump -u [username] -p [database_name] [tbl_name ...] > [backup_name].sql
```

Ejemplo:

```
mysqldump -u root -p wordpress > backup.sql
```

En este ejemplo estamos exportando todas las tablas de la base de datos `wordpress` y estamos guardando la salida con las sentencias `SQL` en un archivo llamado `backup.sql`.

Nota importante: En este caso **no se incluye** la sentencia `CREATE DATABASE` en el archivo de *backup*. Sólo se generan sentencias de tipo `CREATE TABLE` y `INSERT`.

Ejemplo:

```
mysqldump -u root -p tendabd clientes > d:\clientes.sql
```

En este ejemplo, estamos exportando la tabla `clientes` de la base de datos `tendabd` al archivo `clientes.sql` que guardará en la unidad `d:\`

Exportar una o varias bases de datos completas

```
mysqldump -u [username] -p --databases db_name [...] > [backup_name].sql
```

Ejemplo:

```
mysqldump -u root -p --databases wordpress mediawiki > backup.sql
```

En este ejemplo estamos exportando dos bases de datos completas llamadas `wordpress` y `mediawiki`, y estamos guardando la salida con las sentencias `SQL` en un archivo llamado `backup.sql`.

Nota importante: En este caso sí se incluye la sentencia `CREATE DATABASE` en el archivo de *backup*.

Exportar todas las bases de datos completas

```
mysqldump -u [username] -p --all-databases > [backup_name].sql
```

Ejemplo:

```
mysqldump -u root -p --all-databases > backup.sql
```

En este ejemplo estamos exportando todas las bases completas que existen en el MySQL Server al que nos estamos conectando. La salida con las sentencias `SQL` se guarda en un archivo llamado `backup.sql`.

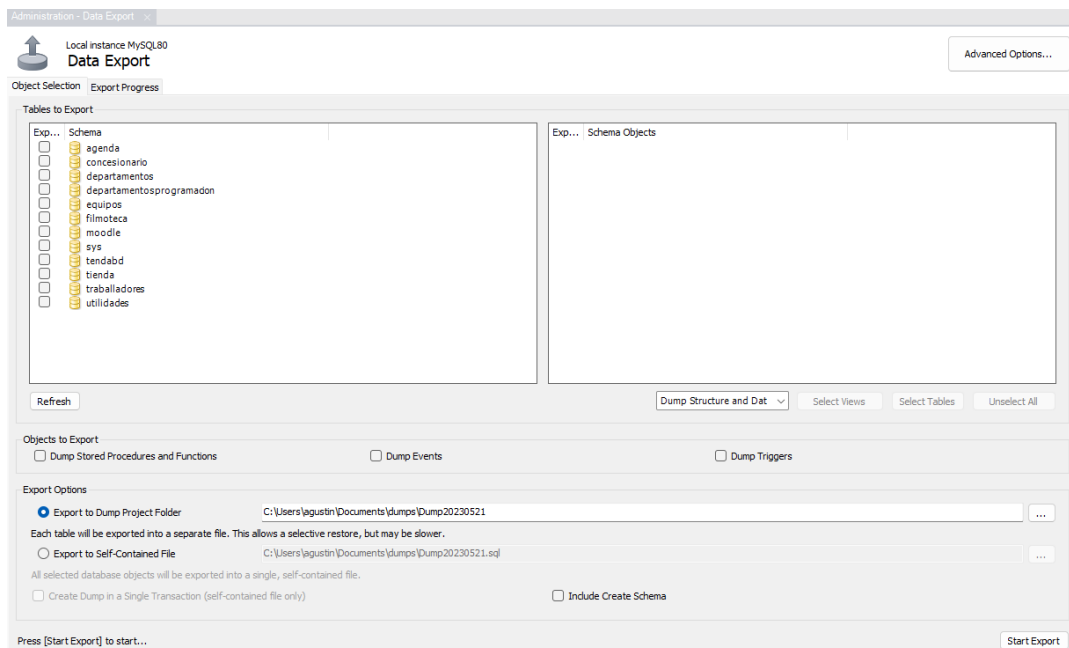
Nota importante: En este caso sí se incluye la sentencia `CREATE DATABASE` en el archivo de *backup*.

Nota importante: Si no se especifica la ruta donde se quiere guardar el archivo que contiene la exportación, se hará en el directorio donde estemos ubicados y ejecutemos el comando `mysqldump`. Si no tenemos `mysqldump` incluido en el `path`, y lo ejecutamos posicionándonos en la carpeta `bin` de la instalación del servidor `MySQL`, y pretendemos guardar en dicha carpeta el archivo generado, es posible que no nos deje, por denegarnos el acceso a esa carpeta. Es mejor poner siempre una ruta donde queramos exportar el archivo.

1.5 Realizar copias de seguridad con MySQL Workbench

También es posible realizar copias de seguridad haciendo uso de herramientas gráficas como `MySQL Workbench`.

Desde la opción "Server" de la barra de menú, tenemos la opción "Data Export". También podemos acceder a esta opción desde el panel de administración



1.6 Restaurar una copia de seguridad lógica o textual

Para restaurar una copia de seguridad lógica o textual existen varias opciones. Podemos hacerlo **haciendo uso de aplicaciones gráficas** como MySQL Workbench o phpMyAdmin, o **desde línea de comandos**.

Para restaurar una copia de seguridad lógica o textual desde la línea de comandos tenemos las siguientes posibilidades:

mysql < backup.sql

Con el cliente de línea de comandos de MySQL (*mysql*) podemos hacer uso del operador de redirección de entrada < para ejecutar sentencias SQL desde un script e importar una copia de seguridad lógica o textual.

Dependiendo de la opción que hayamos elegido para generar el *backup*, será necesario indicar previamente el nombre de la base de datos donde vamos a restaurar la copia.

Recuerda que la sentencia `CREATE DATABASE` sólo se incluye en el *backup* cuando usamos las opciones `--databases` y `--all-databases`. En estos casos podemos restaurar el *backup* con el siguiente comando:

```
mysql -u [username] -p < [backup_name].sql
```

Ejemplo:

```
mysql -u root -p < backup.sql
```

En este caso nos estamos conectando con el usuario `root` y estamos restaurando todas las sentencias SQL que están incluidas en el archivo `backup.sql`.

Si hemos realizado el *backup* sin usar las opciones `--databases` y `--all-databases` entonces la base de datos sobre la que vamos a restaurar los datos debe existir. Si no estuviese creada la podemos crear con la siguiente sentencia SQL:

```
CREATE DATABASE db_name CHARACTER SET utf8mb4;
```

Una vez que tengamos creada la base de datos podemos restaurar el *backup* con el siguiente comando:

```
mysql -u [username] -p [db_name] < [backup_name].sql
```

Ejemplo:

```
mysql -u root -p wordpress < backup.sql
```

1.6.1 `source backup.sql`

Desde el cliente de línea de comandos de MySQL (`mysql`) podemos hacer uso del comando `source` para ejecutar sentencias SQL e importar una copia de seguridad lógica o textual.

Ejemplo:

```
# Desde la línea de comandos del sistema operativo
mysqldump -u root -p --databases tienda > tienda.sql
mysql -u root -p

# Desde la línea de comandos de MySQL
source tienda.sql
```

También es posible utilizar el comando `\.` que es equivalente al comando `source`.

Ejemplo:

```
# Desde la línea de comandos del sistema operativo
mysqldump -u root -p --databases tienda > tienda.sql
mysql -u root -p

# Desde la línea de comandos de MySQL
\. tienda.sql
```

En estos ejemplos, primero realizamos un backup de la base de datos tienda. Después entramos en la consola `mysql`, indicando que el usuario será `root` y `-p` para que nos pida la password.

Después, dentro de la consola mysql, ejecutamos source o .\ para restaurar la copia de seguridad. Tener en cuenta, que, si la copia de seguridad lleva la sentencia de creación de BD, se creará la BD y se restaurará la copia de seguridad en la BD.

Si la copia de seguridad no contiene la sentencia de creación de BD y solo contiene sentencias de creación de tablas, deberemos mediante el comando *use* poner en uso la BD sobre la que queremos restaurar las tablas.

Ejemplo:

```
# Desde la línea de comandos del sistema operativo
mysqldump -u root -p tienda > tienda.sql

mysql -u root -p
use tienda

# Desde la línea de comandos de MySQL
source tienda.sql
```