

---

## **Práctica 8. SQLite**

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2023/2024

# Índice

<b>1</b>	<b>SQLite</b>	<b>1</b>
1.1	Instalación de SQLite en Ubuntu . . . . .	1
1.2	<i>Backup</i> físico . . . . .	1
1.3	mysqldump . . . . .	1
1.4	Exportar una o varias tablas de una base de datos . . . . .	1
1.5	Exportar una o varias bases de datos completas . . . . .	2
1.6	Exportar todas las bases de datos completas . . . . .	2
1.7	Restaurar el <i>backup</i> de una base de datos . . . . .	2
1.8	Automatizar el <i>backup</i> con un <i>script</i> . . . . .	3
1.8.1	Bash <i>script</i> . . . . .	3
1.8.2	crontab . . . . .	3
1.8.3	Telegram bot . . . . .	4
1.8.4	Ejercicios propuestos . . . . .	5
1.9	Referencias . . . . .	5
<b>2</b>	<b>Licencia</b>	<b>6</b>

# Índice de figuras

# Índice de cuadros

# 1 SQLite

## 1.1 Instalación de SQLite en Ubuntu

```
1 sudo apt-get update
2 sudo apt-get install postgresql postgresql-contrib
```

## 1.2 Backup físico

Este tipo de *backup* realiza una copia de los archivos de datos reales de la base de datos. Por ejemplo, podemos usar `mysqlbackup` para bases de datos **InnoDB** y `mysqlhotcopy` para **MyISAM**. Este tipo de *backup* permite restaurar una copia de la base de datos mucho más rápido que el *backup* lógico.

Las utilidades `mysqlbackup` y `mysqlhotcopy` sólo están disponibles en la herramienta [MySQL Enterprise Backup][3] que está incluida en [MySQL Enterprise Edition][4].

## 1.3 mysqldump

La utilidad `mysqldump` permite realizar **backups lógicos** de una base de datos MySQL.

Existen tres formas de usar `mysqldump`:

- para exportar una o varias tablas de una base de datos,
- para exportar una o varias bases de datos completas,
- para exportar todas las bases de datos completas.

```
1 mysqldump [options] db_name [tbl_name ...]
2 mysqldump [options] --databases db_name ...
3 mysqldump [options] --all-databases
```

## 1.4 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:

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

**Ejemplo:**

```
1 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`.

## 1.5 Exportar una o varias bases de datos completas

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

**Ejemplo:**

```
1 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`.

## 1.6 Exportar todas las bases de datos completas

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

**Ejemplo:**

```
1 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`.

## 1.7 Restaurar el *backup* de una base de datos

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:

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

**Ejemplo:**

```
1 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`:

```
1 CREATE DATABASE db_name CHARACTER SET ut8mb4;
```

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

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

**Ejemplo:**

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

## 1.8 Automatizar el *backup* con un *script*

### 1.8.1 Bash *script*

```
1 #!/bin/bash
2
3 # Datos de acceso a MySQL server
4 USER=""
5 PASSWORD=""
6
7 # Ruta donde vamos a guardar los archivos de backup
8 BACKUP_PATH="/path/mysql/backup"
9 DATE=$(date +"%d-%b-%Y")
10
11 # Hacemos la copia de todas las bases de datos que hay en MySQL server
12 mysqldump --user=$USER --password=$PASSWORD --all-databases > $BACKUP_PATH/
    $DATE.sql
13
14 # Comprimos el archivo de backup
15 gzip $BACKUP_PATH/$DATE.sql
16
17 # Eliminamos los archivos de backup creados hace más de 30 días
18 DAYS=30
19 find $BACKUP_PATH/* -mtime +$DAYS -exec rm {} \;
```

### 1.8.2 *crontab*

`crontab` es una utilidad que nos permite ejecutar tareas programas en un sistema operativo GNU/Linux.

Cada usuario tiene su propio `crontab` y para poder editarlo sólo hay que ejecutar el siguiente comando:

```
1 crontab -e
```

### 1.8.3 Telegram bot

Podemos crear un *bot* de Telegram para recibir una notificación en nuestro dispositivo móvil cada vez que se realice un *backup* de las bases de datos.

Para crear un *bot* de Telegram necesitamos:

1. Iniciar una conversación con el *bot* [@BotFather de Telegram](#).
2. Ejecutar el comando `/newbot` para solicitar la creación de un nuevo *bot*.
3. Elegir un **nombre para el bot** y **nombre de usuario**. El nombre de usuario tiene que terminar en **bot**.  
Ejemplo:
  - nombre del *bot*: `Backup`
  - nombre de usuario: `BackupBot`
4. Una vez creado el *bot* Telegram nos devolverá el `API Token` del *bot*.
5. Obtener cuál es nuestro `chat_id` para que el *bot* pueda enviarnos notificaciones a nuestro dispositivo móvil.

En primer lugar hay que iniciar una conversación con el *bot* que acabamos de crear y enviarle algún texto.

Después hay que hacer una petición HTTP GET a la siguiente URL reemplazando `$TOKEN` por el valor de nuestro `API Token` y buscar cuál es nuestro `id`.

```
1 https://api.telegram.org/bot$TOKEN/getUpdates
```

Por ejemplo:

```
1 https://api.telegram.org/bot502192697:AAGnfNmLmXaw8kd0Rh4hMbg6B9sTx0ECzWa/getUpdates
```

6. Una vez que tenemos el `API Token` y nuestro `chat_id` ya podemos hacer que el *bot* nos envíe notificaciones haciendo uso de la API de Telegram.

Podemos hacer una prueba haciendo una petición HTTP GET a la siguiente URL, reemplazando `$TOKEN`, `$ID` y `$TEXT` por nuestros valores.

```
1 https://api.telegram.org/bot$TOKEN/sendMessage?chat_id=$ID&text=$TEXT
```

Una vez que disponemos de toda la información necesaria podemos diseñar un *bash script* sencillo que haga uso de la API de Telegram para enviarnos notificaciones. Por ejemplo:

```
1 #!/bin/bash
2
3 # Credenciales de Telegram
4 TOKEN=""
```



```
5 CHATID=""
6
7 # API de Telegram
8 URL="https://api.telegram.org/bot$TOKEN/sendMessage"
9
10 # Texto de la notificación
11 DATE=$(date +"%d-%b-%Y")
12 TEXT="Backup realizado $DATE"
13
14 # Hacemos una petición HTTP GET a la API de Telegram
15 curl -d "chat_id=$CHATID&disable_web_page_preview=1&text=$TEXT" $URL
```

### 1.8.4 Ejercicios propuestos

1. Diseñe un *script* que realice un *backup* todos los días de la semana a las 00:00, de todas las bases de datos de MySQL Server.
2. Diseñe un *script* que realice un *backup* todos los viernes a las 08:00, de todas las bases de datos de MySQL Server.
3. Diseñe un *script* que realice un *backup* el día 1 de cada mes 05:00, de todas las bases de datos de MySQL Server.

## 1.9 Referencias

- [PostgreSQL](#)
- [Cómo instalar PostgreSQL en Ubuntu 16.04](#)

## **2 Licencia**

Esta página forma parte del curso Bases de Datos de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.