

SOLUCIÓN DE PROBLEMAS CON LOAD DATA DE MYSQL SERVER

Es muy probable que, al intentar ejecutar la instrucción `LOAD DATA` en MySQL surja el siguiente mensaje de error:

```
ERROR 1290 (HY000): The MySQL server is running with the --  
secure-file-priv option so it cannot execute this statement
```

Esto ocurre porque por defecto el servidor impide la carga de datos desde archivos externos para cualquier directorio del sistema, salvo aquel que está habilitado; lo mismo ocurre para las exportaciones de datos. Si esta es la situación, lo primero que debes intentar es ejecutar el siguiente comando en tu consola de MySQL:

```
SHOW VARIABLES LIKE "secure_file_priv";
```

Obtendrás una salida como la siguiente:

```
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| secure_file_priv | C:\ProgramData\MySQL\MySQL Server 8.0\Uploads |  
+-----+-----+  
1 row in set, 1 warning (0.01 sec)
```

Verás en la columna **Value** la ruta del directorio habilitado para hacer importaciones y exportaciones. Lo primero que debes intentar entonces es guardar tu archivo de importación en dicho directorio. Supongamos que dicho archivo se llama `Importar.txt`, así que una vez que lo has guardado en el directorio indicado por la consulta anterior probarás a hacer lo siguiente en tu consola:

```
LOAD DATA LOCAL INFILE "Importar.txt" INTO TABLE NombreTabla;
```

Presta atención a que se ha agregado la palabra **LOCAL** a la instrucción `LOAD DATA`.

Ahora bien, es posible que esta solución funcione para algunos/as sí y para otros no, dependiendo en gran medida del sistema operativo que se esté usando. Si no te funciona lo siguiente a intentar es usar `LOAD DATA` sin incluir **LOCAL** pero usando toda la ruta del archivo:

```
LOAD DATA INFILE "C:\ProgramData\MySQL\MySQL Server 8.0\Uploads"  
INTO TABLE NombreTabla;
```

Esta solución también tenderá a funcionar en ciertos casos y en otro no. Si el problema persiste entonces tendremos que pasar a hacer algunas cosas un poco más avanzadas.

MySQL Server usa un archivo de configuración al iniciar, el cual se llama `my.ini` en Windows, `my.cnf` en Linux y macOS. Tendremos que modificar dicho archivo manualmente para intentar solucionar el problema. En **Windows** suele ubicarse dicho archivo en la misma carpeta donde instalaste MySQL Server; en mi caso está en:

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

En **Linux** suele estar en `/etc/mysql`, pero si no lo encuentras allí ejecuta el siguiente comando en la consola del sistema (no la consola de MySQL):

```
$ /usr/sbin/mysqld --help -verbose
```

Se te mostrará algo por el estilo:

- `/etc/my.cnf`
- `/etc/mysql/my.cnf`
- `/usr/etc/my.cnf`
- `~/my.cnf`

En **macOS** el archivo suele estar ubicado en el directorio `/usr/local/opt/mysql` o `/usr/local/mysql/etc`.

Una vez lo encuentras deberás abrirlo con algún editor básico de texto y buscar la línea que contiene **secure-file-priv**:

```
# ***** Group Replication Related *****
# The host name or IP address of the replica to be reported to the source
# during replica registration. This value appears in the output of SHOW
REPLICAS
# on the source server. Leave the value unset if you do not want the replica
to
# register itself with the source.
# report_host=0.0

# NOTE: Modify this value after Server initialization won't take effect.
lower_case_table_names=1

# Secure File Priv.
secure-file-priv= "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads"

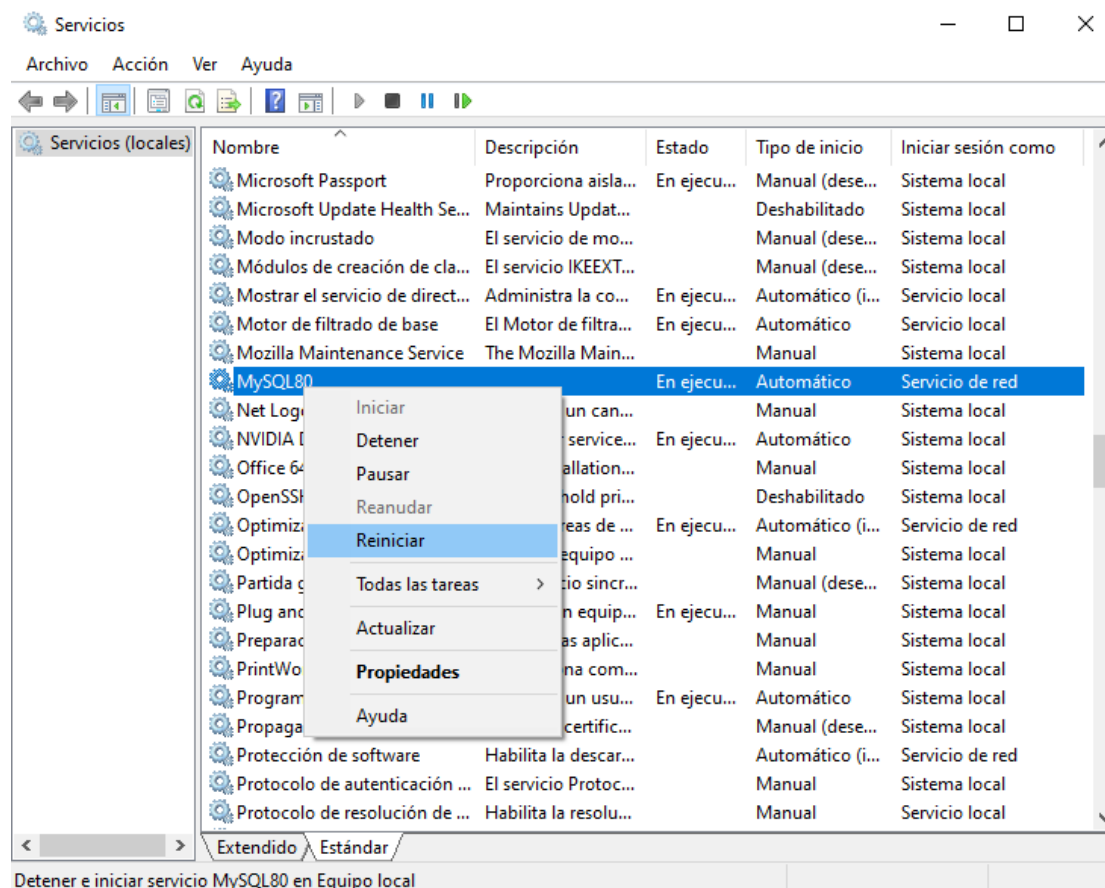
# The maximum amount of concurrent sessions the MySQL server will
# allow. One of these connections will be reserved for a user with
# SUPER privileges to allow the administrator to login even if the
# connection limit has been reached.
max_connections=151
```

La he resaltado en negrita en este ejemplo. Verás que dicha línea contiene entre comillas la ruta del directorio habilitado para importaciones y exportaciones. Pues indicarás allí otro directorio que tú hayas creado. En mi caso pondré “C:\SQLite\Archivos masivos”:

```
secure-file-priv= "C:\\SQLite\\Archivos masivos"
```

Nota que he usado doble barra inclinada en vez una barra común. Hecho el cambio guarda el archivo y ciérralo. Ahora el directorio para importaciones y exportaciones es el indicado, sin embargo hay que reiniciar el servidor para que este cambio tome efecto. Puedes hacerlo reiniciando el sistema directamente, o bien siguiendo los pasos que describo a continuación.

En Windows usarás el cuadro de búsqueda y escribirás **servicios** y abrirás dicha aplicación:



Buscarás MySQL en la lista (está en orden alfabético), harás clic derecho y elegirás **Reiniciar**. Cierra la aplicación.

En Linux abre una consola y ejecuta:

```
service mysql restart
```

Es posible que esto no funcione ya que el nombre del servicio puede llamarse **mysqld**. Si ves un error entonces ejecuta este comando:

```
service mysqld restart
```

En macOS simplemente detén el servidor desde el panel de MySQL y vuelve a iniciarlo.

Reiniciado el servidor es momento abrir una consola en tu sistema y conectarte a **mysql**, luego ejecuta estos comandos:

```
mysql> SET GLOBAL local_infile=1;
Query OK, 0 rows affected (0.00 sec)
mysql> quit
```

Al usar **quit** se cerrará **mysql**. Ahora volverás a conectarte pero de esta manera:

```
mysql --local-infile=1 -u root -p
```

En este ejemplo se usó el usuario **root**, pero bien puedes hacerlo con **admin** o cualquier otro usuario que hayas creado siempre que tenga permisos suficientes. Ahora intentarás

volver a ejecutar **LOAD DATA** indicando la ruta completa del directorio donde está el archivo, el cual deberás haber guardado en el lugar que ingresaste dentro del archivo my.ini o my.cnf. En mi caso:

```
LOAD DATA INFILE "C:\\SQLite\\Archivos masivos\\Importar.txt " INTO  
TABLE NombreTabla;
```

He utilizado dobles barras inclinadas. Prueba tanto con barras dobles como con barras simples.

En siguientes ejecuciones deberías poder entrar normalmente al servidor **mysql** y usar **LOAD DATA** en este directorio.