**Exporting data from MySQL command line (on Mac)**

**How do you proceed when you get the following error?**

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

I received the above error when I tried to convert a query result into a csv file. I spent hours on solving the issue, so thought would write about what it means and how to solve it. My mac's specification/server setting:

* ​macOS High Sierra – version 10.13.5
* mysql Ver 8.0.11 for osx10.13 on x86\_64 (Homebrew) – no workbench setting

**What does the error mean?**

It means that the privileges to import and export the data are limited. This is because, usually, the privileges are granted only to certain users. As soon as I received the above error, I tried to check the value assigned to the variable: secure\_file\_priv (Please Note: it’s underscore used in the variable not hyphen) by using the following command in the mysql console and I have displayed the outcome immediately below the command:

**Select @@global.secure\_file\_priv;**

+---------------------------+

| @@global.secure\_file\_priv |

+---------------------------+

| Null

+---------------------------+

If secure\_file\_priv = Null, it means that the server disables import and export operations. So in order to export query results into files the variable has to be set to an appropriate value.

**The values for the variable can be set in the following ways:**

* If **secure\_file\_priv** is set as **empty** then the variable has no effect on who should load data into a table or export data from a table or export from query result into a file in mysql. Although it’s not a secure setting, I went for this option as the setting was in my personal laptop.
* If **secure\_file\_priv** is set to the **name of a directory,** the server limits import and export operations to work only with the files in that directory. P.S.: we have to create the directory as the server wouldn’t create it.

**Solution:**

One can set or change the file privileges in order to import data from a file to a table in mysql or to export query results from a table into a csv file format.

**How?**

By locating the configuration file under mysqlid. But, boy, locating the file was a daunting task. What operating system and what version of mysql one’s using are significant factors in locating the file. For **mac OS**, the configuration file will be in the name: **my.cnf** and not in the name: **my.ini**

I used the following command on bash to list all my.cnf file paths:

**locate my.cnf**

There were two files that looked like appropriate ones as the rest of them were of macports that started off like:

**/Users/user\_name/macports/var/macports/sources/**

Of the two appropriate ones that started off like:

* /usr/local/Cellar/mysql
* /usr/local/etc/my.cnf (Note: this is the entire path address)

**/usr/local/etc/my.cnf** is the configuration file. The next step is to figure out a way to open this file from bash (as I do not use mysql workbench). I used the following command to open the file in Text Editor:

**open -a TextEdit /usr/local/etc/my.cnf**

The following details can be seen in the my.cnf file:

# Default Homebrew MySQL server config

[mysqld]

# Only allow connections from localhost

bind-address = some value (the value doesn’t have to be same as one might see in many blogs/posts)

After bind-address, I set the variable to an empty string:

**secure\_file\_priv=""**

After altering, the file looked like:

# Default Homebrew MySQL server config

[mysqld]

# Only allow connections from localhost

bind-address = some value

**secure\_file\_priv=""**

I saved and restarted mysql from bash for the change to take effect using the command:

**mysql.server restart**

While restarting one will see:

Mysql shutdown

…

Success!

Mysql started

…

Success!

After that step, I logged-in to my mysql console and checked the value of the variable again by using the command:

Mysql > **Select @@global.secure\_file\_priv;**

And the value of the variable was set empty as intended.

+---------------------------+

| @@global.secure\_file\_priv |

+---------------------------+

| |

+---------------------------+

**Result:**

I exported the result of a query into a csv file using the command:

SELECT \* FROM table\_name INTO OUTFILE '/Users/user\_name/whichever\_folder or even Desktop/file\_name.csv'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n';

The usual mysql message, when a query gets executed, popped up and I found the exported file in the folder I exported it to.

**Tips:**

To locate the my.cnf path, I first used the following command:

**brew list mysql**

It listed the file path, one of the two file paths mentioned above, that started off as:

**usr/local/Cellar/mysql**

I opened the file in Text Editor and set the variable: secure\_file\_priv=”” and then restarted mysql. But nothing changed. How did I know? When I checked using the command in the mysql console:

Select @@global.secure\_file\_priv it listed Null as its value and that was the initial value that was set to the variable.

+---------------------------+

| @@global.secure\_file\_priv |

+---------------------------+

| Null

+---------------------------+

**Learnings:**

* **brew list mysql** doesn’t list all my.cnf paths
* But **locate my.cnf** does list all my.cnf paths