Help for Import data

Load CSV files into MySQL

## Solution 1: Load local data

In the workbench, right-click on the connection you will use, click on edit the connection, on the Connection tab, go to the 'Advanced' sub-tab, and in the 'Others:' box add the line at the end:

OPT\_LOCAL\_INFILE=1

Then click on that connection, copy the following code

Use test;  
SET GLOBAL local\_infile = 'ON';

CREATE TABLE tweet (

Tid BIGINT PRIMARY KEY,

PostDate datetime);  
LOAD DATA LOCAL INFILE 'whatever/directory/tweets.csv'

INTO TABLE tweet  
FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES  
(tid,@col2,@col3,@col4, PostDate,@col6);

You need to change the 'whatever/directory/tweets.csv' to the directory you plan to use. For Windows user, you may need to preplace ‘/’ with ‘\\’ in the directory path ('whatever\\directory\\tweets.csv')

For the columns that do not need to be imported, use @colX to skip. For the columns that do need to be imported, use the attribute name you defined. In this example, only two columns of data will be imported into tweet relation, the 1st column which is tid, and the 5th column which is PostDate.

## Solution 2: Secure load

Use Load data infile statement. This method is the fastest but has a cryptic error message that is difficult to pinpoint where the error is. See https://dev.mysql.com/doc/refman/8.0/en/load-data.html. This requires that the data file you want to import is in a certain directory as indicated by the variable name 'secure\_file\_priv'. To know the directory name, execute the following statement.

show variables like 'secure\_file\_priv';

Put your file in the directory output as the result of the statement above.

Let’s say that the output is "C:\ProgramData\MySQL\MySQL Server 8.0\Uploads".

For mac user, refer to <https://dba.stackexchange.com/questions/168768/mysql-on-macos-sierra-secure-file-priv-setting> to update 'secure\_file\_priv'.

The following example loads the data in tweets.csv in C:\ProgramData\MySQL\MySQL Server 8.0\Uploads into the tweet relation, ignoring the 2nd, 3rd, 4th, and 6th columns. The IGNORE 1 lines is to ignore the header line.

CREATE TABLE tweet (

Tid BIGINT PRIMARY KEY,

PostDate datetime);  
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\tweets.csv'

INTO TABLE tweet  
FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'  
LINES TERMINATED BY '\n'  
IGNORE 1 LINES  
(tid,@col2,@col3,@col4, PostDate,@col6);

Date transformation

After you import the tweet table, the following examples show how to transform a datetime attributes to three separate attributes: day, month, and year.

## Solution 1

You can conduct the transformation of the data before they are entered. In this case, you don’t need to create/import ‘tweet’ relation. The code will create and import a relation called ‘newtweet’.

DROP TABLE IF EXISTS `newtweet`;

CREATE TABLE `newtweet` (

`tid` bigint PRIMARY KEY,

`day\_posted` int DEFAULT NULL,

`month\_posted` int DEFAULT NULL,

`year\_posted` int default NULL

);

Note that the PostDate is transformed into day, month, and year in DDL.

LOAD DATA LOCAL INFILE 'whatever/directory/tweets.csv'

INTO TABLE newtweet

FIELDS TERMINATED BY ';' OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(tid, @col2, @col3, @col4,@col5,@col6)

set day\_posted= day(str\_to\_date(@col5, '%Y-%m-%d %H:%i:%s')),

month\_posted= month(str\_to\_date(@col5, '%Y-%m-%d %H:%i:%s')),

year\_posted= year(str\_to\_date(@col5, '%Y-%m-%d %H:%i:%s'));

## Solution 2

If you do not want to transform the input directly, you can load the data into the tweet table and do the following to do the data conversion and create the newtweet table. Either way is acceptable for the project.

INSERT into newtweet

SELECT tid, day(posted), month(posted), year(posted)

FROM tweet;