**How to load data in MySQL**

There are many ways to load data into a MySQL database. Here I provide detailed, step-by-step guides for some of these options, aimed at beginners using Microsoft Windows. They are by no means the only options. I also included some of my own conventions in this guide. They are not compulsory.

The following options are described:

* [Option 1. Upload data from .csv file using MySQL 8.0 Command line client](#Option1)
* [Option 2: Upload data from .csv file using MySQL Workbench](#Option2)
* [Option 3: Upload data from .txt file using MySQL 8.0 Command line client](#Option3)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Option 1. Upload data from .csv file using MySQL 8.0 Command line client**

*You can either perform the initial data wrangling steps in:*

*(a) Microsoft Excel before uploading the data to mySQL or*

*(b) later in mySQL once the data is uploaded.*

*Here we will assume you are completing these initial steps in Excel.*

1. **Initial steps in Microsoft Excel:**

* Change all headings to lowercase (using the “=LOWER(cell)” formula) and use “\_” instead of spaces between multiple words in a single heading (my convention)
* Replace missing values indicators (such as NA, #DIV/0!, #VALUE etc.) with “\N” (to signify empty cells)
* Delete empty rows
* Replace commas (i.e. “,”) in decimal values with decimal points (i.e. “.”)
* Calculate the minimum and max values for each column. This will be used in determining the data type later and helps you to identify outliers and other issues.
* Use 2025-09-25 format for dates (change using format cells->date…)
* Make sure that you only have a single tab

1. **Save the document as a .csv file**

* Find out which folder mySQL considers the “safe folder” using the following command in mySQL:
  + mysql> show variables like "secure\_file\_priv";
* Save the .csv file you want to upload into that folder.

1. **Prepare a Microsoft Word or Notepad document with the variable names and their data types**

* The document should have the following format: variable1 DATATYPE, variable2 DATATYPE, etc.
* See here for more info about the different data types in mySQL:
* <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
* See <https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html> for more information about storage implications for the different data types.
* Check you file for formatting issues, such as double spaces etc. This will save you a lot of effort later.

1. **Open MySQL 8.0 command line client**

* Enter password

1. **Create a mySQL database (if it hasn’t been created previously) using the following command:**

* mysql> CREATE DATABASE test;

1. **Check whether the database was created correctly (optional)**

* mysql> SHOW DATABASES ;

1. **Tell mySQL to use that database**

* mysql> USE test;

1. **Create a table in the database (This can be a very tedious and lengthy step, depending on the size of the database)**

* mysql> CREATE TABLE tablename (variable1 DATATYPE, variable2 DATATYPE, etc.);
* Use the “variable name and data type” document you prepared earlier to copy the list directly into the statement.

1. **Check whether the table was created**

* mysql> DESCRIBE tablename;

1. **Load the data from the .csv file into the database**

* mysql> LOAD DATA INFILE 'file path.csv’

-> INTO TABLE tablename

-> FIELDS TERMINATED BY ';'

-> LINES TERMINATED BY '\n'

-> IGNORE 1 rows;

* + Remember to use / , not \ in the file path
  + You can check which character is used to terminate your fields by opening the .csv file in Notepad. It was “;” in my case, but can also be “,” etc.
  + If you have empty cells (i.e. “\N” cells) in the last column you might get the error if you use “LINES TERMINATED BY '\n'. Use LINES TERMINATED BY '\r\n' instead.
  + “IGNORE 1 rows;” ignores the header row, since you already added headers when creating the table.

1. **Check whether the data was loaded correctly using**

* mysql> SELECT \* FROM tablename; (this calls up the whole table so only use this command if the table is relatively small)
* Alternatively, you can use: mysql>df.info ();

**Option 2: Upload data from .csv file using MySQL Workbench**

1. **Open MySQL 8.0 workbench**
2. **Double click on the Local instance MySQL80 (grey block)**

* Enter password

1. **Create the database (if it hasn’t been created previously), using the following command in the query tab:**

* CREATE DATABASE test;

1. **Tell MySQL to use that database**

* USE test;

1. **Use the Table Data import wizard:**

* Open the relevant database in the “schemas tab” by clicking on the arrow next to it
* Right click on “Tables” and select the “Table Data Import Wizard”
* Browse for the file location and select the .csv file
* Click “open” and then “next”
* Select “Create new table”, select the correct database (if not already selected) and name the file
* The wizard will then automatically import the column names and suggested data types (termed “field type” here). This is a major time-saving step, compared to option 1 above!
* Check that you are satisfied with the data types and select “next”
* Select “next” to execute the import.

**Option 3: Upload data from .txt file using MySQL 8.0 Command line client**

* *Data is sometimes stored in text files (.txt) instead of .csv files, but the MySQL “Table Data import wizard” does not allow you to select a .txt file.*
* *There are various ways to get past this obstacle.* 
  + *If the .txt file is small: you can copy and paste the content of the .txt file into Microsoft Excel, use “text to columns” to split the text into columns, save the file as a .csv and upload it using option 1 or 2 above.*
  + *But of the .txt file is very large the content will be too much for the Excel grid and some data will be lost.*
  + *There is another way to import the .txt file data (see https://stacktuts.com/how-to-import-data-from-text-file-to-mysql-database), but you need to create the table manually, which will take a long time.*
  + *I, therefore, recommend using MySQL 8.0 Command line.*

1. **Prepare a Microsoft Word or Notepad document with the variable names and their data types**

* The document should have the following format: variable1 DATATYPE, variable2 DATATYPE, etc.
* See here for more info about the different data types in mySQL:
* <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
* See <https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html> for more information about storage implications for the different data types.
* Check you file for formatting issues, such as double spaces etc. This will save you a lot of effort later.

1. **Open MySQL 8.0 command line client**

* Enter password

1. **Create a mySQL database (if it hasn’t been created previously) using the following command:**

* mysql> CREATE DATABASE test;

1. **Check whether the database was created correctly (optional)**

* mysql> SHOW DATABASES ;

1. **Tell mySQL to use that database**

* mysql> USE test;

1. **Create a table in the database (This can be a very tedious and lengthy step, depending on the size of the database)**

* mysql> CREATE TABLE tablename (variable1 DATATYPE, variable2 DATATYPE, etc.);
* Use the “variable name and data type” document you prepared earlier to copy the list directly into the statement.

1. **Check whether the table was created**

* mysql> DESCRIBE tablename;

1. **Load the data from the .txt file into the database**

* mysql> LOAD DATA INFILE 'file path.txt’

-> INTO TABLE tablename

-> FIELDS TERMINATED BY ';'

-> LINES TERMINATED BY '\n'

-> IGNORE 1 rows;

* + Remember to use / , not \ in the file path
  + You can check which character is used to terminate your fields by opening the .txt file in Notepad. It can also be “,” etc.
  + If you have empty cells (i.e. “\N” cells) in the last column you might get the error if you use “LINES TERMINATED BY '\n'. Use LINES TERMINATED BY '\r\n' instead.
  + “IGNORE 1 rows;” ignores the header row, since you already added headers when creating the table.

1. **Check whether the data was loaded correctly using**

* mysql> SELECT \* FROM tablename; (this calls up the whole table so only use this command if the table is relatively small)
* Alternatively, you can use: mysql>df.info ();