MySQL Database: How do I import delimited data into MySQL?

If you have data that you need to bring into your MySQL database, there are a few ways to do it. Exporting data out of mysql is another topic, described here.

1. Using the LOAD DATA INFILE SQL statement

For security reasons, no one has the mysql FILE priv, which means you cannot "LOAD DATA INFILE". You can, however, use a "LOAD DATA **LOCAL** INFILE" statement as long as you have a mysql prompt on our system and have uploaded the data file to your account here first.

The "LOAD DATA **LOCAL** INFILE" statement will only work from a MySQL prompt on our local system. It **will not work** from any web-based tool such as phpMyAdmin, and will never pull a file in directly off your own computer.

To import a file this way, first upload your data file to your home directory on our system with FTP or SCP. Then get a shell prompt on our system, and then a MySQL Monitor prompt so that you can issue the SQL that will import your file.

For example, suppose you have a data file named importfile.csv that contains 3 comma separated columns of data on each line. You want to import this textfile into your MySQL table named testtable , which has 3 columns that are named field1, field2 and field3.

To import the datafile, first upload it to your home directory, so that the file is now located at /importfile.csv on our local system. Then you type the following SQL at the mysql prompt:

```
LOAD DATA LOCAL INFILE '/importfile.csv'
INTO TABLE test_table
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
(field1, filed2, field3);
```

The above SQL statement tells the MySQL server to find your INFILE on the LOCAL filesystem, to read each line of the file as a separate row, to treat any comma character as a column delimiter, and to put it into your MySQL testtable as columns field1, field2, and field3 respectively. Many of the above SQL clauses are optional and you should read the MySQL documentation on the proper use of this statement.

2. Using a script to parse and import the file

You can also write a script in any programming language that can connect to MySQL (such as PHP) to open your data file, break it up into an array of lines that each represent a row of data, split each line up by the delimiter character (such as a comma ',', tab '\t', semicolon ';', space ' ', etc.), and then perform invididual MySQL INSERT queries (one INSERT for each line) to insert all your data from the file into the appropriate table fields.

Such scripts are not difficult to write in less than 15 lines and can import data from text files just as effectively as a LOAD DATA LOCAL INFILE command. A working example script written in PHP appears below in the Annotations.

3. Importing a mysqldump

MySQL Database: How do I import delimited data into MySQL?

If your data file actually comes from another MySQL database, and not from Excel or any other source, then the most direct way to export and import your data would be to dump out your table or entire MySQL database on the original database server using the **mysqldump** command, FTP the resulting dump file to your account here, and then import the dump file at a shell prompt.

For instructions on creating the dumpfile using the mysqldump command, see this FAQ. For instructions on how to import a dump made with mysqldump, see this FAQ.

Unique solution ID: #253
Author: FAQ Admin
Last update: 2010-10-03 17:00