

How to load CSV to MySQL by Examples.

Loading csv to MySQL or any delimited data files to MySQL database is a very common task frequently questioned about and almost every-time LOAD DATA INFILE come into rescue.

In this article I will explain you about all different scenarios and explain you how to get the data loaded directly from csv to mysql using single command.

The Load Data Syntax:

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[CHARACTER SET charset_name]
[{FIELDS | COLUMNS}
[TERMINATED BY 'string']
[[OPTIONALLY] ENCLOSED BY 'char']
[ESCAPED BY 'char']
]
[LINES
[STARTING BY 'string']
[TERMINATED BY 'string']
]
[IGNORE number LINES]
[(col_name_or_user_var,...)]
[SET col_name = expr,...]
```

Consider we have to load file with following contents:

```
#File-name: example.csv
```

```
col-1,col-2,col-3
```

```
a,2,3
```

```
b,4,5
```

```
c,6,7
```

** Excel file can be easily exported as comma separated / delimited file (csv) by File-Save As option to load data.

How to load CSV to MySQL by Examples.

1. A simple comma-separated file with column header:

#table structure:

col-1 col-2 col-3

Considering our MySQL table having same column sequence we can issue:

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES  
TERMINATED BY '\n' IGNORE 1 LINES ;
```

This is a very common and simple scenario.

Quick updates:

- Ofcourse, if we don't have column headers (col-1,col-2,col-3) in example.csv, IGNORE 1 LINES is not required.
- Note the file path. Here you should make sure your slashes are proper.

You may give path as: C:\\path\\file.csv or C:/path/file.csv.

- If we have datafile to be loaded stored on client (Not on server), we will add LOCAL keyword as given in Syntax.

So, the command will become:

```
LOAD DATA LOCAL INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n' IGNORE 1 LINES ;
```

- If we want to replace existing data by data being loaded from file, we will add REPLACE keyword before INTO TABLE.

Similarly if we want input rows that duplicate an existing row on a unique key value to be skipped, we will use IGNORE keyword before INTO TABLE.

2. Column sequence in file and table are different.

#table structure: example

col-2 col-1 col-3

In this case we need to specify column-name sequence of csv file in order to get data loaded in to proper columns.

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES  
TERMINATED BY '\n' IGNORE 1 LINES (col-1,col-2,col-3);
```

How to load CSV to MySQL by Examples.

3. csv / load data file have lesser number of columns than targeted table

#table structure: example

col-1 col-2 col-3 col-4

Consider, col-1 is auto-increment and not provided in csv.

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (col-2,col-3,col-4) set col-1=null;
```

Passing null value will make col-1 to take an auto-increment value.

Using SET you can assign values to those columns which were not available in csv and are not-null.

You may also use a function for doing some particular task and set a value.

eg. SET col-x=rand();

4. Filling the extra date columns:

This is very similar to 3. Here, we are required col-4 to be filled with present timestamp value, a very simple way to do is altering table

```
ALTER TABLE example CHANGE COLUMN col-4 col-4 TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
```

And then,

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (col-1,col-2,col-2=3) set col-4=null;
```

It should automatically fill the current_timestamp values for us.

5. Loading data with calculated columns:

#table: example

col-1 col-2 col-3 col-4

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (col-1,col-2,col-3, @var1) SET col-4 = @var1/100;
```

Similarly we can alter a string variable as well by altering the variable as follows:

```
SET col-4 = replace(@var1,"find","replace")
```

How to load CSV to MySQL by Examples.

6. Loading csv with table value lookup:

Consider you've got a csv with col1 and col2 data and the 3rd column data is available in another table. You can load the referenced data using a sub-query as follows. You have to make sure you get single row in return may be by using distinct or limit clause.

```
LOAD DATA INFILE 'path/to/example.csv' INTO TABLE example FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' IGNORE 1 LINES (col-1,col-2) SET col-3 = (SELECT field-name FROM linkedTable where keyfield=col-1);
```

I suggest you to visit the source article at <http://kedar.nitty-witty.com/blog/load-delimited-data-csv-excel-into-mysql-server> and go through the user comments, where you'll find few more scenarios.

Thanks for reading.

Kedar.