#56daysofSQL

Day 31
Importing CSV File into MySQL Table

Aditya Shenoy

`LOAD DATA INFILE` statement to import CSV file into MySQL table

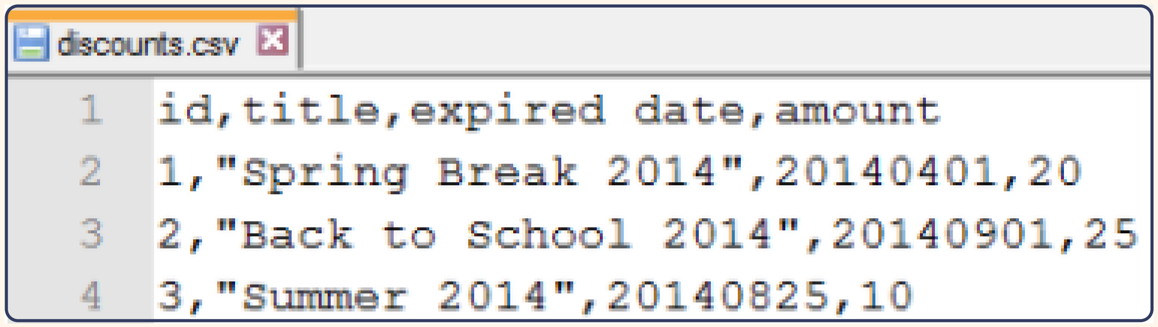
- The **LOAD DATA INFILE** statement allows you to read data from a text file and import the file's data into a database table very fast.

Before importing the file, you need to prepare the following:

- A database table to which the data from the file will be imported.
- A CSV file with data that matches with the number of columns of the table and the type of data in each column.
- The account, which connects to the MySQL database server, has FILE and INSERT privileges.

Use case example:

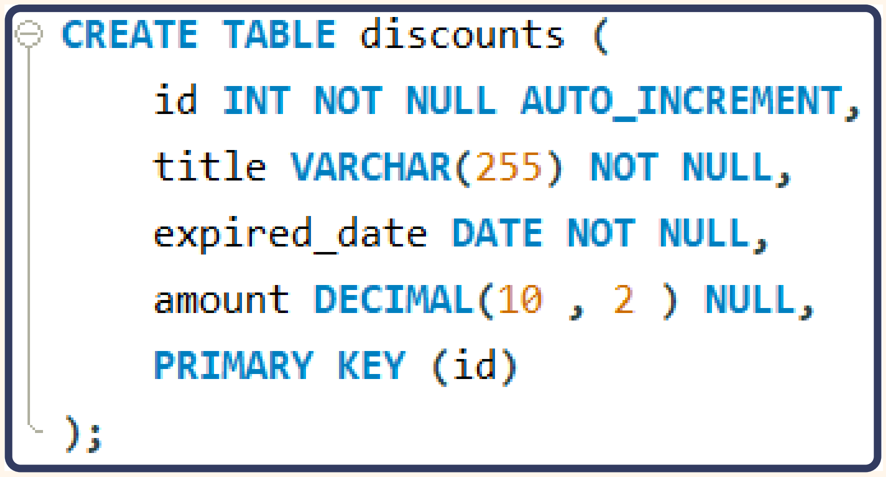
The following **discounts.csv** file contains the first line as column headings and other three lines of data:



```
id,title,expired date,amount
1,"Spring Break 2014",20140401,20
2,"Back to School 2014",20140901,25
3,"Summer 2014",20140825,10
```

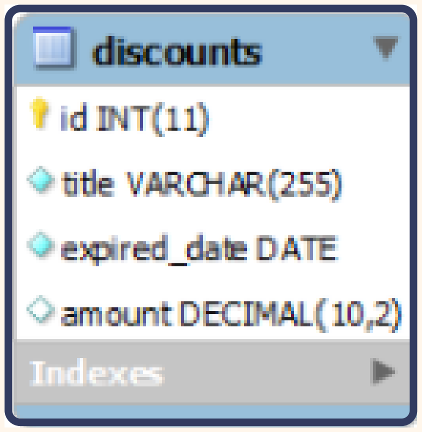
Filename: discounts.csv
File location: C:/Program Files/MySQL/tmp/discounts.csv

Creating the table:



```
CREATE TABLE discounts (
    id INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    expired_date DATE NOT NULL,
    amount DECIMAL(10 , 2 ) NULL,
    PRIMARY KEY (id)
);
```

discounts table structure:



discounts

id INT(11)

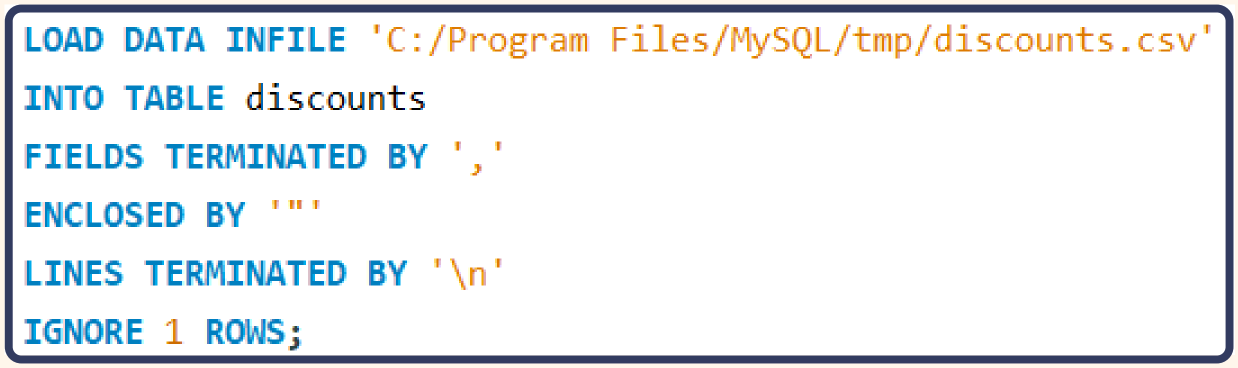
title VARCHAR(255)

expired_date DATE

amount DECIMAL(10,2)

Indexes

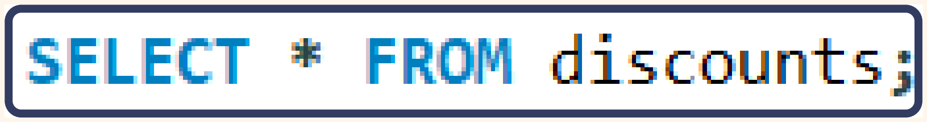
Importing data from **C:/Program Files/MySQL/tmp/discounts.csv** file into the **discounts** table:



```
LOAD DATA INFILE 'C:/Program Files/MySQL/tmp/discounts.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

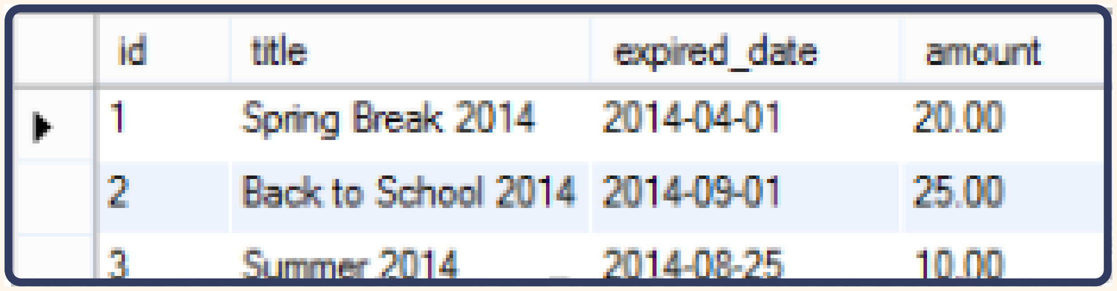
- The field of the file is terminated by a comma indicated by **FIELD TERMINATED BY ','** and enclosed by double quotation marks specified by **ENCLOSED BY '"** ‘.
- Each line of the CSV file is terminated by a newline character indicated by **LINES TERMINATED BY '\n'** .
- Because the file has the first line that contains the column headings, which should not be imported into the table, therefore we ignore it by specifying **IGNORE 1 ROWS** option.

Now, we can check the **discounts** table to see whether the data is imported:



```sql
SELECT * FROM discounts;
```

Output:



	id	title	expired_date	amount
▶	1	Spring Break 2014	2014-04-01	20.00
	2	Back to School 2014	2014-09-01	25.00
	3	Summer 2014	2014-08-25	10.00