

## Blog Export: LAMPsig, <http://www.lampsig.org/new/>

Monday, June 27, 2005

### LAMPsig MySQL Class :: Class #8 (The last class)

Class #8 (the last class) of the MySQL class is scheduled for this Monday, June 27 at 7:00pm at CalTek and will cover Importing and Exporting Data Statements.

\*Complete syllabus for Class #8 are continued in the further reading section.\*

\_Instructor:\_

Christopher Thompson :: cxthompson\_AT\_charter\_(dot)\_net

\_Class Contact:\_

Sharon Lake :: sharon\_AT\_linuxchixla\_(dot)\_org

\_Lab Help:\_

Steve Glasser :: steve\_AT\_fpig\_(dot)\_net

\*\_CLASS #8 SYLLABUS\_\*

Class 8 &ndash; Importing and Exporting Data (Chapter 9, MySQL MM 1.7.5.2, 2.3.16, 8.8, 8.10, 13.1.5, 13.6.2.3)  
\_5% exam material\_

Review of mysqlimport and mysqldumpmysqlimport Command line interface to LOAD DATA INFILE.mysqldump

LOAD DATA INFILEAlternative to mysql INSERT statementLOAD DATA INFILE 'file\_name' INTO TABLE 'table\_name';file name is a string and must be quotedlocation is defaulted to localhostdefault file format (tab delimited, newline-terminated, a value for each column in table)LOAD DATA INFILE syntaxLOAD DATA [LOCAL] INFILE 'file\_name' [IGNORE | REPLACE] INTO TABLE table\_name format\_specifiers [IGNORE n LINES] [(column\_list)];Specifying the Datafile Location without using LOCALDefault location assumed to be local to the server as MySQL reads the file directly.Absolute path on server: LOAD DATA INFILE '/path/to/file.txt' INTO TABLE table\_name;If using a default database then relative path is relative to the default database: LOAD DATA INFILE 'file.txt' INTO TABLE table\_name;If not using a default database then the relative path is relative the the specific database data directory: LOAD DATA INFILE './data\_directory/file.txt' INTO TABLE table\_name;Using LOCAL means the file is local to the MySQL client host. The MySQL client reads the file and sends to the server.Absolute path on clientRelative path on client is relative to the current directory.Special note on Windows ... the path separator is \, but MySQL treats backslash as escape. You can either use the / (forward) slash, i.e., LOAD DATA INFILE 'C:/path/to/file.txt' INTO TABLE table\_name; or escape the backslash character, i.e., LOAD DATA INFILE 'C:\\path\\to\\file.txt' INTO TABLE table\_name>Loading into Specific ColumnsLOAD DATA INFILE 'file.txt' INTO TABLE table\_name (col1, col2);Note, if order in text file doesn't match order in table columns, you can switch the the insertion text order via specifying columns, i.e., LOAD DATA INFILE 'file.txt' INTO TABLE table\_name (col2, col1);Skipping datafile lines or header row(s) with IGNORELOAD DATA INFILE 'file.txt' INTO TABLE table\_name IGNORE n LINES;Dealing with duplicate records.Default behavior on a duplicate Key violation is to stop loading the file at the point of error. All records previously processed remain.IGNOREkeyword will load the entire file, but discard the records causing duplicate key violations.REPLACE keyword will also load the entire file, but will replace the records with duplicate key violations with the new information.Interpreting LOAD DATA INFILE statement resultRecords: Number of tables read from the file, but not necessarily the number of records input into the table.Deleted: Number of records replaced in the table when using the REPLACE keyword.Skipped: Number of input records ignored in the data file when using the IGNORE keyword.Warnings: Indicates possible problems in the input file, i.e., missing values, data conversions, etc. Can be a number greater than the number of records input as more than 1 error can be generated per record.PrivilegesYou need the INSERT privilege for the table in which you want to LOAD DATA INFILE for LOCAL files.You also need FILE privilege for data files which are located on the server.LOAD DATA INFILE is considered an efficient insert method

SELECT INTO OUTFILEAdding the INTO OUTFILE clause into a SELECT statement creates a file on the server with the SELECT result.Since the file is created on the server, the user must have FILE privileges.The file is created with MySQL server as the owner but world-readable. Also since it is owned by MySQL you might not be able to remove the file without server admin privileges. Also, since the file is world-readable, careful about sensitive data.Default format is

one line per row, delimited by tab characters, and lines terminated with newlines. Since the file is created on the server, to access the file you must have a login account on the server host. Not an issue if you only want to read the file back in via a LOAD DATA INFILE as the server has access to the file ... even if you don't.

**Data Format Specifiers** LOAD DATA INFILE format specifiers are listed after the table name. SELECT ... INTO OUTFILE format specifiers are listed after the output filename. Syntax for format specifiers is the same for both statements and can be used in any order. Default values are supplied if missing. FIELDS TERMINATED BY 'string' ENCLOSED BY 'char' / OPTIONALLY ENCLOSED BY 'char' ESCAPED BY 'char' LINES TERMINATED BY 'string'; Default values TERMINATED BY: tab ('\t') ENCLOSED BY / OPTIONALLY ENCLOSED BY: unquoted LINES TERMINATED BY: newline ('\n') Special Note to platform. Unix line terminators are usually '\n' (newline), MAC OS and OSX line terminators are usually '\r' (carriage return), and Windows line terminators are usually '\r\n' (carriage return and newline) ESCAPE BY: backslash ('\'). Escape sequences as understood by MySQL: \N (NULL) \0 (ASCII NUL byte) \b (backspace) \n (newline) \r (carriage return) \s (space) \' (single quote) \" (double quote) \\ (backslash) Example. To load a file with comma-separated values, with values quoted by double quote, and line terminated by carriage returns. To insert: LOAD DATA INFILE 'file.txt' INTO TABLE table\_name FIELDS TERMINATED BY ',' ENCLOSED BY '&quot;'; LINES TERMINATED BY '\r'; To write: SELECT \* INTO OUTFILE 'file.txt' FIELDS TERMINATED BY ',' ENCLOSED BY '&quot;'; LINES TERMINATED BY '\r' FROM table\_name;

**Importing and Exporting NULL Values** LOAD DATA INFILE a \N appearing unquoted by itself as a column value is interpreted as NULL. MySQL converts empty values to a 0, empty string, or a '&quot;zero&quot;'; temporal value depending on the column type. SELECT ... INTO OUTFILE writes NULL values as \N

Posted by Sharon Lake in Classes: MySQL at 19:00

I just want to once again thank Sharon for the unbelievable job she did in organizing this class, setting the agenda, doing all the syllabi, getting the instructors, and seeing the class through. It was truly a herculean effort, and I've never seen it done better.  
Anonymous on Jul 3 2005, 16:48