Character Set and Localization Issues: My special characters display as 'noisy' Latin characters in SQLyog!

Most likely the reason for this is that the install script of the application that is used with the data did not explicitly create databases and tables as UTF8. Then the server will create databases and tables with the default character set and in case this default is not UTF8, data will not be stored as real unicode in the database (but only as byte-sequences 'similar to' unicode)

SQLyog will display what is stored in the database! If the database/tables are defined with a latin character set, latin characters will display. And if the databases/tables are defined as a cyrillic character set, cyrillic characters will display. The actual application may decode the byte stream on the client side, however.

If your specific problem is that you have an application that 'uncritically' saves utf8 encoded strings in a non-utf8 database/table it its possible to correct this. Provided that the MySQL version used supports unicode storage. That is, the MySQL version must be version 4.1 or higher.

If the 'client side encoding and decoding' does not use the standard byte patterns of UTF8 but a encoding pattern specific to that application, we cannot do anything about it (we have seen application specific encodings (more or less) based on UTF7 and ISO 8859-1 for instance!)

However you cannot just change the charset for existing databases and tables once they have been created as non-unicode. That will not change existing columns. Also directly changing every column from latin1 (or whatever) to utf8, will not help. The MySQL server will transform the encoding of existing strings and thus preserve the error.

You will have to 'push in' a binary (non-encoded) step using binary data types (binary/varbinary/BLOB) instead of encoded string types (char/varchar/TEXT).

So basically what you need to do is to perform changes in two operations like:

latin1 varchar >> varbinary (or BLOB) >> utf8 varchar (and similar for char and TEXT types).

This you will have to do on every existing string column in every table.
You may want also change the default charset for every table and for the databases(s), so that new tables and columns will be created as utf8 (in case of an application upgrade for instance).

Similar considerations would apply to applications using UCS2 encoding, of course. But we just have not yet encountered any such situation!

(and of course you should always backup your data before doing such change!)

Unique solution ID: #1134
Author: Peter Laursen
Last update: 2007-07-01 10:20