Data Types and Storage Engines: How do I define TIMESTAMP properties with SQLyog?

The TIMESTAMP datatype is the MySQL datatype that causes most problems. Basically there are three reasons for this

* The implementation for the TIMESTAMP type was changed between servers 4.0.x and 4.1.x. What used to work with old versions may not work with recent versions.

* The CREATE/ALTER TABLE statement supports (with servers >= 4.1) a parameter for the TIMESTAMP type that is not supported with any other datatype. That parameter is the 'ON UPDATE' clause. As a value for this clause only 'CURRENT_TIMESTAMP' (and the synonyms 'now()' and 'localtimestamp') can be used.

* The MySQL server (also with servers >= 4.1) 'silently' transforms some TIMESTAMP definitions specified by user: 1) 'TIMESTAMP NULL' becomes 'TIMESTAMP NULL DEFAULT NULL' 2) 'TIMESTAMP NOT NULL' becomes 'TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP' provided that the 'CURRENT_TIMESTAMP' keyword is not used for a previous TIMESTAMP declaration in the table definition. The 'CURRENT_TIMESTAMP' keyword (and the synonyms) can only be used once in a table definition. For details refer to:

  * http://dev.mysql.com/doc/refman/5.0/en/datetime.html.

* http://dev.mysql.com/doc/refman/5.0/en/timestamp.html.

The SQLyog facility to define TIMESTAMPs (the CREATE TABLE and ALTER TABLE GUI's) is designed to work with this. You can define TIMESTAMPS like

  a) NULL DEFAULT NULL

b) NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

c) (NOT) NULL DEFAULT 'yyyy-mm-dd hh:mm:ss'

d) (NOT) NULL DEFAULT CURRENT_TIMESTAMP

Details on how to do this:

a) Just define the column type and name/identifier and do nothing else with that row of the CREATE/ALTER GUI (leave the default column empty and do not check the NOT NULL checkbox)

b) Just define the column type and name/identifier and check the NOT NULL checkbox and do nothing else with that row of the CREATE/ALTER GUI (leave the default column empty)

c) Just define the column type and name/identifier and enter the default in 'yyyy-mm-dd hh:mm:ss' format. You may check the NOT NULL checkbox or not and the column will be created accordingly.

d) Just define the column type and name/identifier and enter "CURRENT_TIMESTAMP" (without quotes) in the default column. You may check the NOT NULL checkbox or not and the column will be created accordingly.

Note that only 'CURRENT_TIMESTAMP' (and the synonyms 'now()' and 'localtimestamp') entered in the default column of CREATE/ALTER TABLE GUI's will be treated as keywords. Everything else will be considered literal strings and any such literal string that is not recognized by the MySQL server as a valid DATETIME specification will return an error! Refer to http://webyog.com/faq/content/8/99/en/does-sqlyog-gui-understand-the-meaning-of-special-mysql-keywords.html for more details of KEYWORDS support in SQLyog.

One more detail about the situation described with point b) above is that if a column already exists that uses the CURRENT_TIMESTAMP keyword the server will create as "NOT NULL default '0000-00-00 00:00:00'" if the session sql_mode allows for it. Currently SQLyog always uses the '' (empty) sql_mode for its connection/session, unless changed by user. With this sql_mode ZERO-datetime will insert. For sql_modes where ZERO-dates are not allowed setting an explicit default is required (and also note that ZERO-time - as opposite to ZERO-date - is perfectly legal in every sql_mode - however some specific server versions are buggy in this respect!) if not the server shall return an error.

With the CREATE TABLE and ALTER TABLE GUIs, however, you cannot create TIMESTAMP definitions like

e) NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

f) (NOT) NULL DEFAULT 'yyyy-mm-dd hh:mm:ss' ON UPDATE CURRENT_TIMESTAMP

Those constructions - that are both very rare - can be handled from the editor. Users who want to use those will undoubtedly also understand. We do not find it justified to add yet another column to the CREATE TABLE and ALTER TABLE GUI's that can be used only for a single datatype and can only have one single value when it is not really required.

With MySQL version 4.0 and before it is a little more simpler. The server will create the TIMESTAMP as defined by user - simply. But the first timestamp in the table definition will be handled by the server like described here: http://dev.mysql.com/doc/refman/4.1/en/timestamp-pre-4-1.html: "Automatic updating of the first

Data Types and Storage Engines: How do I define TIMESTAMP properties with SQLyog?

TIMESTAMP column in a table occurs under any of the following conditions: ...The column is not specified explicitly in an UPDATE statement and some other column changes value..." In other words: with those server versions the first timestamp will basically be treated like 4.1+ TIMESTAMPS defined 'NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP'. But with those versions this is not implemented in the CREATE STATEMENT for the table. The server simply 'silently' treats the first TIMESTAMP like this! Also note that with those versions a DATETIME string constant is 'yyyymmddhhmmss' only (no "-"s, no ":"s and no SPACE). Further the parameters 'default CURRENT_TIMESTAMP' and 'on update CURRENT_TIMESTAMP' does not apply to those versions. They will return a server syntax error.

Unique solution ID: #1145
Author: Peter Laursen
Last update: 2008-02-13 15:45