

HTTP(s)-tunneling: How can SQLyog handle import of DUMPs with tables using Foreign Keys with HTTP tunnel?

This question has been asked sometimes by people having noticed that SQLyog (just like the 'mysqldump' program for instance) generates a SQL-DUMP like

```
SET @old_FOREIGN_KEY_CHECKS = SELECT @@FOREIGN_KEY_CHECKS;
SET FOREIGN_KEY_CHECKS = 0;
CREATE table ..; /* full statement with FOREIGN KEY CONSTRAINT(s) */
INSERT INTO ...;
SET FOREIGN_KEY_CHECKS = @old_FOREIGN_KEY_CHECKS;
```

People asking this question have also noticed that PHP applications (like phpMyAdmin) will generate this for the same table

```
CREATE TABLE ..; /* note : * no* FOREIGN KEY CONSTRAINT here .. */
INSERT INTO ...;
ALTER TABLE ADD FOREIGN KEY CONSTRAINT ..;
```

PHP applications will need to do like in the second example for the simple reason that `php_mysql(i)` connections to MySQL are non-persistent. A new connection will be created for every statement. And local variables (like `FOREIGN_KEY_CHECKS`) are initialized to server defalt when the new connection is established. You can try this yourself with a PHP application

```
SET FOREIGN_KEY_CHECKS = 0;
SHOW VARIABLES LIKE = 'FOREIGN_KEY_CHECKS';
```

.. and you will get "1" returned. So the second example above is a necessary 'workaround' for this used by PHP applications to ensure that data will import and not fail due to violation of FOREIGN KEY CONSTRAINTs. It works of course. But there is one drawback: It is slow because for every `ALTER TABLE` the table will need to be rewritten by the server. With many and large tables having FOREIGN KEYs this results in considerably slowing down the import process.

SQLyog HTTP tunnelling on the opposite processes 'batches of statements' to the server and for every batch the `SET FOREIGN_KEY_CHECKS = 0;` statement is reinstated. Thus SQLyog can create the table **with** FOREIGN KEY CONSTRAINT from the beginning, import data successfully - and thus no need for any `ALTER TABLE` overhead.

SQLyog and SJA handles this fully automatic in Data sync, Import External Data and 'copy to other' functionality. It is controlled by the code of the SQLyog and SJA binaries and the tunneller script in cooperation. For SQL-dump restore we added in 7.12 an option to "force disable FK check throughout HTTP import batch proces". This will ensure same functionality when executing an external script with SQLyog (from tools .. restore from SQLdump). Due to this HTTP batch processing concept SQLyog will also restore DUMPs created with 'mysqldump', MySQL Administrator, backup routines integrated in applications, phpMyAdmin etc. - and even if HTTP tunnel is used by SQLyog connection.

HTTP(s)-tunneling: How can SQLyog handle import of DUMPs with tables using Foreign Keys with HTTP tunnel?

Note however:

- 1) Statements executed from the SQLyog editor will still (as of version 7.12) be processed as individual statements. We have different solutions for this under consideration.
- 2) Accordingly dumps created with SQLyog (also when using HTTP tunnel) will probably fail to restore properly with PHP applications (like phpMyAdmin) if there are tables having FOREIGN KEY CONSTRAINTs.

Unique solution ID: #1158

Author: Peter Laursen

Last update: 2008-10-03 13:47