

Data Truncation Error: Migrating MySQL to MSSQL

Posted At : July 23, 2009 5:41 PM | Posted By : Mark Kruger

Related Categories: MS SQL Server, Coldfusion & Databases

I have one more tip as a follow up to my previous post on [Migrating Between MySQL and MSSQL](#). It has to do with the dreaded "data truncation error". If you have used MSSQL you may have seen this error crop up from time to time. It is a common error and very easy to remedy. The error occurs when you have a character field with a length that is too short for the size of the string you are trying to insert. Check out this example....

Let's say I have a column called "password" in a table called "mytable". I have set it to be a varchar(5) - that is to say a variable length character column with a maximum length of 5 characters. Consider this query:

```
<cfquery ....>
    UPDATE myTable
    SET password = 'longerThan5Char'
    WHERE user_id = 5
</cfquery>
```

What will happen when this query is run? MSSQL will throw an error that says something like (going from memory here), "Data Truncation Error, String or Binary data would be truncated...". The error is saying in effect that if the MSSQL server proceeds to run the insert, only part of the string will be stored in the table and the rest will be discarded (the string will be truncated).

Now if you are used to MySQL there is a good chance you have never run into this error. Why? Because the default behavior of MySQL ignores this type of error. Instead, the above code run against MySQL would result in the string "longe" being inserted in the table with the rest of the data silently discarded. That's why when migrating from MSSQL to MySQL this error tends to crop up.

You may have an insignificant character column that you've been ignoring which is now causing you problems. For example, if you have a comment section and folks tend to write a few hundred words, you are not likely to notice if the occasional verbose user comes along and attempts to insert a novel. His first *n* number of characters will be picked up and none the wiser. But now that same budding Victor Hugo is throwing errors on your site and he is unable to get even his basic intro inserted.

Fixing the Error

To fix the error, located the column in question and increase the length to accommodate the size of the string. This can be done pretty safely in most cases very slight impact. Another possibility is to switch from varchar or nvarchar to text or ntext, but *be careful*. Although changing the column type in this manner is a "permanent" fix, it is a much more significant change. Not all the code that works with varchar will work with text. Make sure you test such a change before you deploy it live.

Suppressing the Error

My take is that it is a *good* thing that this error is thrown. There are cases where silently storing only a portion of the data could come back to bite you - the encrypted

string of a password for example. However, for those of you who just want to bail out and *force* MSSQL to behave like MySQL you can suppress this error by turning off ANSI warning messages like so:

```
<cfquery ....>
SET ANSI_WARNINGS OFF

    UPDATE myTable
    SET password = 'longerThan5Char'
    WHERE user_id = 5

SET ANSI_WARNINGS ON
</cfquery>
```

The thing to note about suppressing the error is that this approach suppresses *all ANSI warnings* including things like divide by zero errors or math overflow type errors. So be careful and make sure you can live with the impact of doing it this way.