

Re: Different MSSQL output date format from the same PHP script

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2008-02/msg01400.html>

- *From:* Willem Bogaerts <w.bogaerts@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 20 Feb 2008 10:19:23 +0100
-

Please don't delete attributions; they're important to following the conversation. I've readed them.

Then please read them and don't turn things the other way.

SELECT * is not a good thing to use. You are much better off always specifying the columns.

For one thing, it's very seldom you need all of the columns.

Why put columns in a table if you are not interested in them? It is really rare and a sign of bad datase stucture if a table contains things I do not want to select.

This is a strawman. Jerry suggested no such thing. He said that you don't usually need all of the columns for a particular query. IME that's quite true.

As I indicated, this is not true for my way of programming. It may be very well true for yours. But that does not make "SELECT *" bad in all case. THAT is what Jerry suggested. By the way, this thread was about dates in queries of that type. Just saying it is bad does not help. Especially when "SELECT *" queries just do have their uses.

The only time I want all fields from a particular table is usually when allowing record additions and editing via a user interface. Even in those cases I generally use views because I'm joining against related tables. Most of time when I'm querying, however, is for reporting, which usually requires only a subset of the available columns.

Re: Different MSSQL output date format from the same PHP script

And that is exactly the opposite for most of my programs. My programs are object oriented, and an object usually "maps" to a row in a database. If my application evolves, so do both the table and the object. This is where ORM and "SELECT *" queries are of great value.

So instead of telling that the original poster is wrong, people may actually consider helping him.

But more importantly, what happens if someone later adds a new column to the table (i.e. a 5mb BLOB)?

Well, first of all, that would be me.

Are you the only one who ever works on the databases you create? If so, do you think that your experience is representative?

Yes. As I told above, that is the mere structure of the application. My programs "depersist objects" far more often than they "report data".

If I would add such a column, I would have a need for it.

In every single case that you wanted to retrieve data from that table? You wouldn't want to ever want to, say, run a query returning all unique authors associated with the documents that are in the BLOB field?

No, off course not. I did not say that I NEVER used field names, did I? For searching and reporting, I still use complex JOINS, a subset of field names, etc. It just so happens that those queries are less than 5% of my application. That may very well be the opposite for yours, but I will not tell that to be bad and stop responding to a perfectly normal question.

If that field would only be vaguely connected to the rows in that table, it would be in an only vaguely connected table.

Sorry, but this doesn't make any sense to me.

If the field would not really be part of the entity that that row (and

Re: Different MSSQL output date format from the same PHP script

that object) represents, it would just not be in that table.

An example:

Suppose the table is a table with images. If I want to retrieve the image object, I would need the image ("SELECT *"). If I wanted to search which objects I would want to read, I would use different queries or already have IDs in other objects. If I wanted to do both, I could combine them.

So what happens? I would update the mapping table and a template and I would have an extra field on my web form. Without the need to modify the code. That is why ORM layers exist, is it not?

Also, if someone later deletes or renames a current column, the query will fail, making the problem very obvious.

On the contrary, "SELECT *" will NOT fail.

He meant that if you write your query as "SELECT FirstName, LastName FROM employees" and someone renamed "FirstName" to "GivenName", then the query will fail immediately. SELECT * will not. Whether using SELECT * will cause problems depends on whether your code references "FirstName" by name.

So? The query itself may not fail, but the rest of my code (most probably the ORM section first) will. Just as immediate. What's the difference?

Regards,

—

Willem Bogaerts

Application smith

Kratz B.V.

<http://www.kratz.nl/>

.