

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

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

---

*Source:* <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2008-02/msg01434.html>

---

- *From:* Charles Calvert <[cbciv@xxxxxxxx](mailto:cbciv@xxxxxxxx)>
  - *Date:* Wed, 20 Feb 2008 21:48:58 -0500
- 

[attributions readded... again]

On Wed, 20 Feb 2008 10:19:23 +0100, Willem Bogaerts  
<[w.bogaerts@xx](mailto:w.bogaerts@xx)> wrote in  
<[47bbf09e\\$0\\$14343\\$e4fe514c@xxxxxxxxxxxxxxxx](mailto:47bbf09e$0$14343$e4fe514c@xxxxxxxxxxxxxxxx)>:

On Tue, 19 Feb 2008 17:54:38 -0500, Charles Calvert <[cbciv@xxxxxxxx](mailto:cbciv@xxxxxxxx)>  
wrote in <[3nmmr3hvf2qtk9rggrr6u99gcir1kpias@xxxxxxxx](mailto:3nmmr3hvf2qtk9rggrr6u99gcir1kpias@xxxxxxxx)>:

On Tue, 19 Feb 2008 13:03:43 +0100, Willem Bogaerts  
<[w.bogaerts@xx](mailto:w.bogaerts@xx)> wrote in  
<[47bac59f\\$0\\$14355\\$e4fe514c@xxxxxxxxxxxxxxxx](mailto:47bac59f$0$14355$e4fe514c@xxxxxxxxxxxxxxxx)>:

On Tue, 19 Feb 2008 06:40:01 -0500, Jerry  
Stuckle <[jstucklex@xxxxxxxxxxxxxxxx](mailto:jstucklex@xxxxxxxxxxxxxxxx)> wrote in  
<[25idnRKzfsPjXSfanZ2dnUVZ\\_t3inZ2d@xxxxxxxxxxxxxxxx](mailto:25idnRKzfsPjXSfanZ2dnUVZ_t3inZ2d@xxxxxxxxxxxxxxxx)>:

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

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

I don't understand this sentence. Removing attributions is rude.  
Please don't do it. For your reference:

<[http://en.wikipedia.org/wiki/Top-posting#Attribution\\_lines](http://en.wikipedia.org/wiki/Top-posting#Attribution_lines)>  
<<http://www.i-hate-computers.demon.co.uk/quote.html>>  
<[http://www.cs.indiana.edu/docproject/zen/zen-1.0\\_6.html](http://www.cs.indiana.edu/docproject/zen/zen-1.0_6.html)>  
<<http://coherentdog.org/netquote.htm>>  
<<http://linux.sgms-centre.com/misc/netiquette.php#quoting>>  
<<http://www.newsreaders.com/guide/netiquette.html>>

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

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 database structure 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 cases. THAT is what Jerry suggested.

Jerry suggested that SELECT \* is generally bad practice, and he suggested it to someone who, based on the question they were asking, seemed to be less experienced with databases than Jerry or a lot of the other regular posters in this group. Was he making an assumption? Yes, but it was a fairly good one, given the available evidence.

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.

Jerry was pointing out what he perceived to be a second issue that arose as a result of the discussion of the first. This happens all the time.

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.

And that is exactly the opposite for most of my programs. My programs are object oriented,

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

So are mine.

and an object usually "maps" to a row in a database.

Ah. So you're using the database as a persistence layer for your objects. Fine. There's nothing inherently wrong with that and I can see why you'd want to map all available columns to an object.

If my application evolves, so do both the table and the object. This is where ORM and "SELECT \*" queries are of great value.

Okay.

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

If you go back and reread the responses, you'll notice that he was helped. AnrDaemon posted the reason behind the difference the O.P. was seeing and several people suggested solutions. The O.P. raised an objection to one of the solutions because he was using SELECT \* and didn't want to change that. At this point, Jerry chimed in. It was a perfectly reasonable thing to do at that point in the conversation.

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".

I take it that your "yes" was to my first question, not my second. I can tell you that your model is not the most common one by far. Your applications are an exception to the rule. Let's assume for the moment that if I examined your applications that I would agree that SELECT \* is the best way to handle it. Even so, I would still apply

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

the rule of thumb that SELECT \* is not a good way to select data from a database, generally speaking. There are always corner cases. That they exist doesn't invalidate good rules of thumb, especially when those rules of thumb can stop the inexperienced from causing themselves problems.

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?

No, but I wasn't sure that I grasped what you were saying, so I asked questions to help refine my understanding.

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,

Part of my point is that it is the opposite of most database applications.

but I will not tell that to be bad and stop responding to a perfectly normal question.

I didn't suggest that you stop responding to anything. I'm merely arguing with you. It's not a personal attack. This is Usenet.

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

An example:

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

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.

I understand your example, but not your point. Your original paragraph still doesn't make sense to me. It's quite common to have columns in a table that are infrequently used, but still correctly part of a normalized table.

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?

As Jerry pointed out, it's easier to troubleshoot when the error is manifested closer to the source.

There's another issue that has been implied, but not stated explicitly. SELECT \* can be a performance nightmare when the table has many records, when the connection is slow, when the server has to handle many requests or any combination of the above.

Like any programming construct, SELECT \* has its uses. Unfortunately, it is often used incorrectly by the inexperienced, hence the general rule to not use it in production code. That's what Jerry was getting

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

at. If I were to codify the rule, it would be something like this.

Rule 1: Don't use SELECT \*

Rule 2: (For the experienced only) Don't use SELECT \* in production code

Rule 3: (For experts only) Don't use SELECT \* unless nothing else will do

The third rule may apply in your situation. It's unlikely that it does in the O.P.'s.

—

Charles Calvert | Software Design/Development

Celtic Wolf, Inc. | Project Management

<http://www.celticwolf.com/> | Technical Writing

(703) 580-0210 | Research

.