

Re: There's got to be a better way

Re: There's got to be a better way

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2007-03/msg01424.html>

- *From:* "shimmyshack" <matt.farey@xxxxxxxxxx>
 - *Date:* 25 Mar 2007 18:01:02 -0700
-

On 26 Mar, 02:33, Jerry Stuckle <jstuck...@xxxxxxxxxxxxxxxxxx> wrote:

shimmyshack wrote:

On 25 Mar, 23:37, Jerry Stuckle <jstuck...@xxxxxxxxxxxxxxxxxx> wrote:

shimmyshack wrote:

On 25 Mar, 21:37, Mary Pegg
<inva...@xxxxxxxxxxxxxxxxxx> wrote:

shimmyshack wrote:

```
if
(isset($c['s']['a5']))
echo
htmlentities($c['s']['a1'])."<br>";
.
.
.
if
(isset($c['s']['a5']))
echo
htmlentities($c['s']['a5'])."<br>";
that's alot of
work for
sake of
sticking
with what
_you_ find
more
readable
```

So what you're really
arguing in favour of is
wrapping it up in a
function, which is what I've
done. But the question is
whether it's

Re: There's got to be a better way

worth creating a for loop to
run through a1 to a5 rather
than simply
calling each by name.

all the
changes
above and
more are so
simple with
Rani's
method. If
you

No, they're simple if it's
wrapped up in a function.
Whether or not
a1 to a5 get generated by a
for loop I've still got b, c, d,
e, f (etc)
to deal with. NB I'm using
these as symbols – in reality
they are the
field names from a database,
so no getting smart and
suggesting that I
can generate b to f
automagically. OTOH I
could stick the field names
in an array and step through
the array. This might be
worth doing.

can't take
good advice
don't ask for
it.

I know this is Usenet but
you don't *have* to be rude
and abrasive.

--
"Checking identity papers is
a complete waste of time. If
anyone can
be counted on to have valid
papers, it will be the
terrorists".

it's not rude or abrasive, to notice you don't
take advice and say so,

Re: There's got to be a better way

perhaps a little brusque though, and I apologise if I made you cross. FYI, I work with annoyingly complex database tables whose structure changes as my client changes their requirements, to stop this kind of hard coded approach which started to cost me too much time, I use DESCRIBE `tablename`, and SHOW FULL COLUMNS FROM `tablename` which can then be used to get the comments, fieldnames etc... then using a regexp on bool enum varchar() int() and so on to get metadata about the table which is then pumped into the application. In this way you can use general methods to print and parse data without ever having to hard code the fieldnames, you can use it to dynamically generate forms etc... Instead of repeatedly writing code to format output from the DB, you just need a valid link, and some form of instructions what your db connection is to "get" and "output" and finally the output format – preventing too much or too little from being drawn from the DB. You only have to code this once, and after that it can be used everywhere you need output. It might seem a little extreme to folks, I don't know, but I find this approach saves time. The class which serves html markup can end up being very complex but you can control things with a couple of calls and an array.

PMJI, but then you aren't much of a programmer.

I've been doing SQL Database work for over 20 years now. I'll bet some of the databases I've designed would make yours look puny – over 100 tables, over 1,500 columns, for instance. And mostly 3rd normal form.

Re: There's got to be a better way

And yes, these databases do change as customer requirements change. But I deal with them.

Your problem is that the user actually *cares* about the contents of the database. They don't. What they do care about is the *data* – including the relationships. Whether data is contained in one table or ten is not important.

You can give all the excuses you want for not taking good advice. But the bottom line is – you haven't given any excuses we haven't heard hundreds of times. And you aren't explaining a situation most of us haven't run into multiple times. And we deal with it properly.

--
=====
Remove the "x" from my email address
Jerry Stuckle
JDS Computer Training Corp.
jstuck...@xxxxxxxxxxxxxx
=====

I'm not sure what to make of that Jerry, but I wonder if my point was clear enough. The idea is to write php classes which auto discover DB structure, as does say phpmysqladmin. The idea is to let the DB structure be independent of the php code, responding to changes within the DB such as character encoding, field types, number of columns etc... so that little or no hard coding and adjusting of the logic is needed, just an adjustment – if any – to the "display" if one can call it that – to config that prescribes the way the code interacts with the DB tables.

I don't then mind how often changes are made to the DB structure, which as you say always changes as the project grows and the relationships become clearer, nor do I mind how often the tables are split (pi\$\$ing contest avoided), provided a there exists a data<~>query map.

ALways with programming it's bread and butter, reinventing the wheel for each new app (even with OO)

Re: There's got to be a better way

forming the query, running it, getting the data, persisting it, parsing, filtering, displaying and so on...

I wanted something a bit more like a fluent interface, more readable and "semantic", in that it moves things on from having to write step by step, and makes things a bit more fluid. I would be nice to get eventually to a form where anyone could write in words what they required; perhaps when chicken foot's reg exp parser comes up with this ;)

I personally hate to see the type of hard coded stuff that gets banged out, but it exists in the real world – of course.

You just don't get it, do you. The DB Structure is not important to the client. The DB CONTENTS ARE!

For instance – in the case of an order entry system. Does the user want :

order_number product_number quantity

Or do they want:

Customer name

Order_number product_name quantity Price Total_Price

PhPMyAdmin is for administering a MySQL database. The user isn't interested in the administration of the database. He/she is interested in the contents. And not just one table, like above. Rather, he's interested in the joining of at least three tables – maybe more.

Sure, I rewrite code each time I start a new project with a new database. That's because every database is unique and has unique requirements.

What you think would be nice really isn't realistic, other than for simple database admin like PhPMyAdmin.

Programming is all about reinventing the wheel. For instance, almost every PHP statement has been written at least once. Sure, the variable names change, but that's about it. But it's how you put the statements together that counts.

Same with databases. It's not about the access. And it's not about database design.

It's about how things fit together to provide data in a format a person can use.

--

=====

Remove the "x" from my email address

Jerry Stuckle

Re: There's got to be a better way

JDS Computer Training Corp.
jstuck...@xxxxxxxxxxxxxx
=====

I think you have missed the point again Jerry; although I would say that database design and access can be very important for large projects for optimisation. I am only building on this exact point:

It's about how things fit together to provide data in a format a person can use.

the rest is just a given. You are cross at someone but it isn't me.

It can only be reiterated: although coding is about reinventing the wheel, this is pointless, and will change into the future, becoming more high level and looking and feeling less like C. Finally resulting in code that looks like a sentence, and compiles to be equivalent to today's stuff. Until the compilers get cleverer, the wheel reinvention problem can be solved using clever code.

Your point

The DB Structure is not important to the client. The DB CONTENTS ARE!

can be widened, the client doesn't care about anything except functionality on time to budget.

true but that implies fast writing of code, with less skills required to write it and greater ease of maintainability. One of SQLs great pluses is that it's not that hard to learn and it can read like a sentence at the basic level.

The same can and will be true of a scripting lang like php.

The upshot is writing code independent of DB structure, independent of markup requirements, which takes a config file, rather like Ant builds an application, means less stress for you – the over worked wheel reinventor, and less stress for the client, who likes rapid app dev.

I dunno where we are arguing, unless your 20+yrs has blinded you to the possibility of a world where Joe Bloggs back office boy can program because Joe Blogg's dad the guru made a compiler that understood english.

What got my goat was that printing statements out in googlicate was just the kind of thing that causes problems into the future, while it might compile to something similar to a judiciously chosen while loop, it just feels rather intellectually lazy, no offence intended.

In the same way people argued against calculators I guess you can still disagree. Calculators let anyone leapfrog the grunt work, and start with the interesting Maths. The metaphor is writing code once that then let's the programmer get on with writing the app, rather

Re: There's got to be a better way

Re: There's got to be a better way

than reinventing the wheel and grinding out lines of the same old code. Not everyone loves the destination as much as they do the journey. oo deep

.