

Re: Updating the SQL key value

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2007-03/msg01727.html>

- *From:* Erwin Moller <since_humans_read_this_I_am_spammed_too_much@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 30 Mar 2007 15:33:49 +0200
-

Toby A Inkster wrote:

bill wrote:

I will add a autonumbering PK

Auto-numbered synthetic primary keys are the root of all evil. They usually indicate a lazy approach to database design. There is almost always a natural column or combination of columns that can be used as a primary key without the need to add an extra, redundant numerical column which doesn't contain any useful information.

Assuming that the you never plan on having two events that occur concurrently, your "order" column is a perfect natural candidate key. (Though it's a little poorly named, given that ORDER is a SQL keyword.)

Hi Toby,

Despite my high respect for you: I completely disagree.
Why picking 'natural candidates' if you can make it work ALWAYS with a simple autonumbering PK?
What if your database must be upgraded and the logic changes?
Do you want to check all the columns again to be sure the PK still makes sense? (Or watch it fail in a production environment when the UNIQUE constraint is hit you didn't see coming beforehand)

I have been using autonumbering PK my whole programming carrier, and never had any problems with it.

Who seriously cares about the few extra bytes needed?

I am not alone with that thought.
Postgres even makes an OID for each row, something you don't even see but can use if you want.

Re: Updating the SQL key value

It is perfectly natural IMO when designing a database to point to each row in a simple, coherent, and easy to understand fashion by using autonumbering PKs.

It also makes it very easy to use FK.

I don't get it why you think of that as 'the root of all evil'.

What evil comes out of it?

Regards,
Erwin Moller

.