

Re: COBOL/DB2 Date edit question

Source: <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2007-08/msg00478.html>

- *From:* "Pete Dashwood" <dashwood@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 14 Aug 2007 20:01:54 +1200
-

"Frank Swarbrick" <Frank.Swarbrick@xxxxxxxxxxxxxx> wrote in message
news:46C045ED.6F0F.0085.0@xxxxxxxxxxxxxxxxxxxx

On 8/12/2007 at 6:17 PM, in message

<5i9m8uF31v17qU1@xxxxxxxxxxxxxxxxxxxx>,
Pete Dashwood<dashwood@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

<snipped conceptual level explanation>.

Stored procedures are becoming a way of life on many sites, and embedded
SQL
is showing a consequent decline.

I still have the issue I mentioned in my other message. Personally, I am
not so much concerned as to *where* the business logic / validations
exist.

If it's in a stored procedure, that's fine (as long as I have access to
look

at it!). My concern is that this appears (from my, granted, limited point
of view) to require a lot of 'back and forth' between the user, the
application, and the database. Is this not the case?

Yes and no.

There is switching between the entities you mention, but if it doesn't incur
heavy overheads, why would you care?

The concepts are as outlined in my previous mail; You can have the validation
processes stored with the actual data and removed from the presentation
layer (user interface) entirely. Some people will like this idea, others
won't. (I do :-))

Re: COBOL/DB2 Date edit question

For me, it isn't just the logical tidiness, it is a step towards the future. (I think this may be influencing several of the places I know where they are moving to this model, too.) Banging away at databases with primitive SQL is not in the same league as running query expressions. This technology hasn't made it's way onto mainframes yet (as far as I know... if anyone is using Query Expressions or Lambda functions for DB access on a mainframe, please post here...), but I am picking it will within the next few years. It is probably fair to say that multiple processors are in the future for most commercial sites and this, combined with new storage technologies that are just around the corner, is going to require different and more powerful techniques of data manipulation if we are to reap the benefits of parallel processing and improved storage hardware.

After the first RDBMS call, the overheads in subsequent calls are virtually nonexistent. Not only that, but the DB subsystem is self monitoring and self optimizing. (Program logic usually is not.) The system will optimize itself to the most frequent access paths. (this is an area where expert software is getting smarter and less fallible than people...)

One piece of logic that we have in many of our user update type programs is the "Do you really mean this" type screen. Here's how it works...

The user enters various fields on various screens. Secondary screens may or may not be present depending on data entered on previous screens. At the end of all of the input the user is presented with a screen detailing all of the input he has entered, with a button to "Submit" or "Cancel". When he submits then the data is actually sent to the database (or to the mainframe!).

Within all of this a lot of validation has been going on. Dates have been checked, fields have been validated to make sure they are allowed in combination with other fields, etc. The final "Submit/Cancel" page will not even be presented to the user until such time as all of the input has been validated as being allowed.

Can this still be done with stored procedures and what have you?

Definitely, and probably even more efficiently than currently. As fields are entered they can be passed to the RDBMS, which triggers a validation. If the field passes, it is already stored and doesn't need to be, later. (it is still available to the program code, but now it is known to be valid). Cross field validations can be done in the same way. Collect the fields required

Re: COBOL/DB2 Date edit question

and submit them together to an UPDATE that triggers the cross validation. If it fails, the user is requested to correct the data. No SUBMIT has been issued so the updates are not applied, and, anyway, triggers can operate immediately BEFORE or AFTER the function they are triggered by. (You always have the option to ROLLBACK if you wanted to approach this differently...)

As you can see, validations can be quite complex and there are facilities to support anything you need to do.

Everything you describe in the penultimate paragraph above can be easily accomplished with stored procedures and triggers. It moves this validation out of program code, and, at least in my opinion, that is a "consummation devoutly to be wished" (Sorry, MacBeth...) . Should you remove any existing fields from the DB, no program changes are required, and if/when new RDBMS technology is implemented, conversion to it is likely to be automatic.

However, you DO need to learn full SQL and write the procedures and triggers.

It's just another language ... :-)

(While you're at it, why not learn Query Expression syntax; (it looks like SQL, but is immensely more powerful?)

I'm not
trying to be argumentative.

A refreshing change for CLC...:-)

I simply see problems and I don't know the
solutions.

Well, this is a good place to gather thoughts and ideas, if not
solutions...:-)

Pete.

—

"I used to write COBOL...now I can do anything."

.