Re: chooses not to generate code at all

Source: http://coding.derkeiler.com/Archive/General/comp.object/2005−08/msg00655.html

- *From*: JXStern <JXSternChangeX2R@xxxxxxx>
- *Date*: Wed, 24 Aug 2005 02:52:05 GMT

On Tue, 23 Aug 2005 15:03:03 GMT, "H. S. Lahman"
<h.lahman@xxxxxxxxxxx> wrote:
>>>[At least in the conventional sense where the procedure is triggered by
>>>specific database updates.
>>
>> That would be a "trigger", not a "stored procedure" as such.
>
>The whole point of having a stored procedure as an integral part of the
>DBMS is so that the DBMS engine can execute it when triggered by some
>update activity on the stored data. If there is no trigger, then...

My usage of the terms is standard, yours is not. Boot up your
favorite database and see what the tools call things.

>>>In theory one could store code, parametric
>>>data, and whatnot in the DBMS that is selectively invoked by external
>>>applications. In that case those applications define the scope and
>>>context of execution rather than the DBMS' data. So in that case the
>>>DBMS is just a storage medium for the procedures and doesn't decide when
>>>they should be executed.]
>>
>> That is exactly what a "stored procedure" is all about.
>
>This is just an arbitrary blob of data and the execution dynamics are
>external to the DBMS. IOW, it is not an integral part of the DBMS itself.

Well, fine, that's how ANSI felt about it up until recently, too.
Unfortunately, best practice for twenty years has been to use the
non−standard stored procedures to encapsulate functionality and
decouple applications from the data model.

If you're not into it, btw, you would be AMAZED at the issues of
efficient execution that underlies what seems such a simple thing as a
SQL statement. Makes your average C compile−link−go seem like a
kiddie script.

>>>An RDB is not "an MDA". It is simply a model instance that can be
>>>either an Input Model or an Output model under MDA, depending upon
>>>context. It happens to instantiate a meta−model, which is reflected in

>>>schemas, etc.. That meta−model instantiates a general data model
>>>especially tailored to persistence (Table, Tuple, Field).
>>
>> I'm not sure that these abstractions are "especially tailored to
>> persistence". Certainly "field" is a general concept.
>
>The point here is that SQL is not the only way to access an RDB; it is
>just one access paradigm. Nor is a conventional RDB the only way to
>implement Codd's RDM (e.g., OODB's and UML Class Models also honor that
>model.)

Oh, I hardly think so. The point of RDM is normal form, which is
barely even known to most practitioners of OO.

OTOH, it's barely known to half the DBA's I run into, either.

>[Note, BTW, that a UML Class Model is normalized to 3NF just like an RDB
>schema.

Cite me an authority on that, please. The difference in style and
practice between relational and object is long−standing, if things
have changed I didn't get the memo.

> But the abstractions may be quite different even though both
>the application and the RDM are abstracting from the same customer
>problem space.

Impossible.

> (They won't be so different that one cannot map
>unambiguously between them because they /are/ derived from the same
>problem space.)

Nonsense.

> In addition, the application developer should not care
>if the data is persisted in an RDB, OODB, flat files, or clay tablets;
>that sort of thing should be completely transparent to the problem
>solution.]

Easy to say, but the "impedance mismatch" between technologies is an
every−day problem.

J.

.

- ***Follow−Ups***:
  - ***Re: chooses not to generate code at all***

- *From:* H. S. Lahman

- ***References***:
  - ***Re: chooses not to generate code at all***
    - *From:* H. S. Lahman
  - ***Re: chooses not to generate code at all***
    - *From:* JXStern
  - ***Re: chooses not to generate code at all***
    - *From:* H. S. Lahman
  - ***Re: chooses not to generate code at all***
    - *From:* JXStern
  - ***Re: chooses not to generate code at all***
    - *From:* H. S. Lahman
  - ***Re: chooses not to generate code at all***
    - *From:* JXStern
  - ***Re: chooses not to generate code at all***
    - *From:* H. S. Lahman

- Prev by Date: ***Re: XP and Pair Programming***
- Next by Date: ***Re: XP and Pair Programming***
- Previous by thread: ***Re: chooses not to generate code at all***
- Next by thread: ***Re: chooses not to generate code at all***
- Index(es):
  - ***Date***
  - ***Thread***