

Re: choices regarding where to place code – in the database or middle tier

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.databases/2004-01/0136.html>

From: Joe Weinstein (joeNOSPAM_at_bea.com)

Date: 01/19/04

Date: Mon, 19 Jan 2004 13:45:38 -0800

To: Daniel Roy <danielroy10@hotmail.com>

Daniel Roy wrote:

> *Hi Joe,*
> *I am a Siebel configurator/programmer (Siebel is a "Customer*
> *Relationship Management" software, which can be considered analogous*
> *to SAP). My personal experience with the issue which interests you is*
> *that as much as possible should be stored in the database. Siebel, by*
> *some twisted reasoning about compatibility of code on various*
> *databases (it runs on Oracle, SQL Server and DB2), decided to keep*
> *almost all the code (including referential integrity!) in the middle*
> *tier. As a result, on ALL the projects I've been a part of, we have*
> *had data issues. The worst part is for the foreign keys which are not*
> *valid. Other issue are about some code (usually PL/SQL) which is not*
> *in sync with the database, for whatever reason (access rights,*
> *objects/columns which don't exist anymore, ...). Also, performance is*
> *always better from inside the database, from what I've seen so far.*
> *This is logical since there is less network traffic when everything is*
> *done from Oracle.*

Thanks. This conversation will become richer and clearer as folks get into it. Sure, the DBMS is a good place for simple referential integrity constraints, as well as set-based data processing. Stored procedures and triggers are important to do that sort of thing in one place. I knew of a payroll application that took 8 hours to do 40,000 employee run because it sucked raw data, person by person from the DBMS to the fat client to do the real grinding. When this was converted to 4 separately-running-but-pipelined stored procedures, it took 15 minutes to do the same work. What **shouldn't** be in the DBMS is user session control or even most less-volatile online data that is tapped by user applications. I saw a European software company's product (I won't name names but it starts with "Baa" ;-)) that has a set of application queries, from which all useful business functions were built. A typical user would enact a 100 or so business functions per day. Each function had a part where it queried the DBMS for the list of countries in Europe! Every user, hundreds of times a day! I am aware of the political flux in Europe in the 90's but it was never that bad! ;-)

Fundamentally, you don't want every user of your application to cause/require a new

comp.lang.java.databases: Re: choices regarding where to place code – in the database or middle tier

DBMS connection, let alone ask it for the same non-volatile data 10,000 times a day.

To get the best performance, applications are going to have to be DBMS-specific at some level. DBMSes aren't dumb file systems. I knew of another application vendor, who shall remain nameless (but it rhymes with 'beoblesoft' ;)) that 'rolled it's own stored procedures' by storing the SQL for every business query in the DBMS as string data, and to do any given function, would query the DBMS for the SQL needed, and would then send that SQL back to be executed as fresh SQL! However, that dbms-specific level should be as narrow and controllable/switchable as possible. J2EE standards help there.

Joe

> *Just my 2 cents*

>

> *Daniel*

>

>

>>*Hi –*

>>

>>*Over the last several versions of Oracle, developers have been provided with a pretty revolutionary idea for a database product – namely the ability to write code that used to belong in the middle tier and store it in the database. I'm referring here to the ability to write stored procedures in Java.*

>>

>>*Now of course, Microsoft with their SQL Server product is doing the same thing. The next version of SQL Server will allow programmers to write stored procedures in any of the .NET languages.*

>>

>>*I'm interested in looking at the increased choices developers now have because of these new features in more depth, developing some best practices on the subject, and possibly publishing an article on the topic.*

>>

>>*I personally am more experienced with SQL Server than with Oracle. I am therefore looking for someone who has been involved with making these choices in the Oracle environment who would like to collaborate with me on the subject.*

>>

>>*If you are interested, please contact me at joelax@dbdirections.com*

>>

>>*Thank you*

>>

>>*Joe Lax*