

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

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.databases/2004-01/0179.html>

From: Joe (joelax_at_dbdirections.com)

Date: 01/23/04

Date: Fri, 23 Jan 2004 17:23:00 GMT

Jim –

Thanks for your response. I agree with your points. I'll try to clarify my question regarding the areas that still bother me.

Before Oracle decided to allow you to write stored procedures in Java, they had this really nice language (as you pointed out) called PL/SQL. It seems to allow you to do just about anything you would want to do with your data and they've continued to enhance it with each version of Oracle. (though I haven't looked at 10G yet myself)

Then along comes Java, a language that attempts to solve a whole different set of problems, and not necessarily the same ones that PL/SQL was designed for. Even without Java existing, let alone being hosted in the database, you can still have all the same arguments about where to put your business logic. However, at least some of the choices would be clearly delineated by what language was being used. If you found a particular problem that had a better solution in PL/SQL, it was going to live in the database. If either because of the nature of the problem, or a particular programmer's facility with something besides PL/SQL, a particular business rule or process was expressed in another language, it would typically live outside the database.

But along comes Oracle and takes Java and hosts it in the database. (Why, I'm not sure. Was it just to grab the Java crowd?) Now, you have a situation where you have the very same piece of code that can live in the database or in your middle tier. What drives the choice then?

In the Microsoft world, the situation is a bit different. SQL Server never had as well developed a language as PL/SQL. I've still managed to support large systems with stored procedures using T-SQL, but it lacks features that Oracle programmers take for granted such as packages, decent error handling (although that will be improved in the next version), the list goes on and on. In the Microsoft world, the database was never treated as the center of the development universe in the way that Oracle seemed to consider their product. But Microsoft with the next version of SQL Server, is taking a set of languages that they developed for the middle tier, and putting it in the

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

database. (As opposed to the Oracle situation, this isn't even a way to reach out to a different developer community like Java programmers, .NET developers are already in the Microsoft camp.). When I look at the literature on the choice Microsoft has made, what I get is a series of examples demonstrating problems that are more easily solved in .NET than in T-SQL. What I don't see then is any reason that I shouldn't simply have this code in the middle tier. In fact, the more I read about it, the more I am curious as to what drove Microsoft to put this feature in.

I've received a lot of valuable feedback from numerous people in this thread. Some of the conversations have been at a strategic level such as "Is it a good or bad development that database vendors seem to be merging the database and middle tier?". What I've gleaned from the conversation so far is

1. Most people agree that you shouldn't forgo declarative referential integrity in favor of rules in the middle tier. In that way lies data corruption. (for those of you that disagree – please note I said "most")
2. When you get to situations that require very high performance, it's worth taking a look at a TPC Monitor or other middleware solution. It's what database vendor's themselves do to get the utmost best performance.
3. You can get into awkward situations by forcing everything into one layer or another – witness some of the stories regarding various vendor packages. One of the ways to judge where to put a particular peice of code and how to write it, should be clarity and maintainability.

I'm still trying to clarify for myself situations that fall in the middle. So, take a situation where the code in question will perform adequately in either tier, and assume we are not talking about enforcing declarative referential rules. Where should this code go? What other factors besides performance come into play?

To me, it's not just a theoretical question to be happily argued over a beer or two. I currently support clients using SQL Server. In around a year, all the .NET developers out there will be faced with this choice. Many of them I suspect won't think much about it, but merrily use either the left mouse click or the right to make their selection (Microsoft makes somethings way too easy to accomplish in opinion). I'm a pragmatist enough to know that I can't just say "Don't use this feature, put all your code in the middle tier" or the opposite answer either. Instead, I'm looking to learn from what has already taken place on the Oracle side of things about the consequences of such choices.

How is code maintenance affected by these choices? Is it easier, harder, or really has no overall effect on projects if Java code lives in two places? Does tuning performance of code become harder?

Are people seeing cases of poorly performing instances because the Java code being hosted is not written will?

Is this causing more situations where the programmer points at the DBA and

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

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

the DBA points at the programmer?

Who actually decides where the code should reside?

Has it caused more security issues?

I'm sure there are also other issues that I haven't thought of, which again is the reason for posting these questions in an Oracle forum, where you've had to deal with these issues on a practical level for several years now.

Thanks

Joe Lax

joelax@dbdirections.com

"Jim Kennedy" <kennedy-downwithspammersfamily@attbi.net> wrote in message news:QX3Pb.103910\$8H.177187@attbi_s03...

>

> "Joe" <joelax@dbdirections.com> wrote in message

> news:Y7IOb.20178\$ko5.19870@nwrddc01.gnilink.net...

> > 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*

> >

> >

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

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

> >

> >

> >

> *Joe,*

> *I want to make a subtle distinction. Just about any database can store code*

> *in a the database. (binary object) That said I think you mean more that*

> *complex business logic can be stored and run in the database or server end*

> *(eg in Oracle pl/sql or Java). Having the business logic (not the GUI*

> *logic) in the database allows one to switch GUIs or have multiple systems*

> *interact with the backend and consistant business rules are followed.*

> *Having it in the middle tier means that every other system has to go through*

> *that middle tier. Which means that other groups will go right to the*

> *database and not through the middle tier.(time constraints, must do it now,*

> *can't wait to use middle tier, middle tier written in a language we don't*

> *like or don't know....)*

>

> *Siebel, Peoplesoft et al hire programmers and not really dbas.*

Programmers

> *drive the projects and dbas are relagated to a lower importance. Thus these*

> *products don't use Referential integrity, stored procedures etc. For an*

> *example, in Siebel you "have to define all database objects through their*

> *tool even indexes". Unfortunately, that means you can't create a Function*

> *based index or an index where one of the elements of the key is descending*

> *instead of the default ascending. Dumb, just dumb.*

>

> *Jim*

>

>