

## Re: 2 tier vs 3 tier

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-12/4436.html>

---

**From:** Murphy (*murphy\_at\_nospam.com*)

**Date:** 12/19/04

Date: Sun, 19 Dec 2004 11:30:21 +1100

Mike,

Thanks and I fully understand exactly what you mean, the reason I asked the question is because of the following item I read on the MS website that has gotten me confused.

<http://lab.msdn.microsoft.com/express/faq/default.aspx#vwd>

You seem to be encouraging folks to do data in a two-tier way (embedded SQL within your page). I want a nice 3-tier databinding model.

We have a rich databinding model in Visual Studio 2005 and ASP.NET 2.0 – one that supports binding ASP.NET controls against a variety of different data sources. One of these data-sources is the "<asp:objectdatasource>" control, which is specifically designed to enable you to bind against middle tier object layers, enabling true n-tier databinding using ASP.NET 2.0 controls.

Does this advocate 2 or 3 tier ? and if 2 how does this reconcile with the points you raised below ?

Thanks

Murphy

"Mike Epprecht (SQL MVP)" <mike@epprecht.net> wrote in message news:eQx7AXN5EHA.4004@tk2msftngp13.phx.gbl...

> *Security and abstraction. Forget about changing the DB engine as that is a massive engineering project.*

>

> *Do you want a web server to talk (directly though a firewall) to your all important data?*

> *With 2-tier, all your data access logic sits on the web server and you end up putting most of your business logic in the DB.*

> *Easy to develop, but terrible to run as once your overload your back-end, to*

> *get better performance, you need to spend massive amount of money of big hardware.*

>

> 3-tier was developed so that you can re-use your business logic, validation  
> and access routines across various display technologies. If only  
> presentation specific logic is on the front end, it means that you can  
> change the front end with no problem as your logic behind it stays the same.  
> Think WinForms, WebServices and WebSites here. In future, stuff like mobile  
> devices will be more popular, but you can't show the same form as on a web  
> site as the device won't display it well, due to the size limitations. You  
> want to write twice the same business logic and data access code because  
you  
> have to support 2 presentation formats?  
>  
> Regards  
> -----  
> Mike Epprecht, Microsoft SQL Server MVP  
> Zurich, Switzerland  
>  
> IM: mike@epprecht.net  
>  
> MVP Program: <http://www.microsoft.com/mvp>  
>  
> Blog: <http://www.msmps.com/epprecht/>  
>  
> "Murphy" <murphy@nospam.com> wrote in message  
> news:u#3D7DM5EHA.3644@tk2msftngp13.phx.gbl...  
> > I am currently in the early stages of developing my 1st web app and have  
> > noticed some discussion about the advantages of a 2 tier approach as  
> > opposed  
> > to 3 tier.  
> >  
> > I see an obvious advantage of 2 tier being that one could change db  
engine  
> > with minimal fuss however you forfeit the advantages of stored  
procedures  
> > etc.  
> >  
> > Comments ?  
> >  
> > Murphy  
> >  
> >  
> >  
> >  
> >