

Re: Three questions.

Re: Three questions.

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.clients/2008-07/msg00044.html>

- *From:* Hugo Kornelis <hugo@xx>
 - *Date:* Sun, 20 Jul 2008 22:01:22 +0200
-

On Sat, 19 Jul 2008 19:07:17 -0400, Noor wrote:

Dear Professional,

I have three questions

First question

(snip)

Hi Noor,

That's the same question you posted in your previous message, and I just replied to that.

Second Question

I have created merge replication and I am little bit worry about Identity columns, do you have guidelines in terms of Identity columns merge to Subscriber?

If you are worried about new identity values created at several servers, there are three standard ways to get around this:

1) Don't use surrogate keys at all, stick to the real keys. There are many advantages to this approach (even without replication), though there are disadvantages as well. It is an often debated issue, since lots of people tend to religiously take position in favor of or against surrogate keys. I won't repeat the arguments here; I'm sure you'll find them if you google for them.

2) Use GUID instead of IDENTITY. This is exactly the kind of scenario that GUIDs are designed for. Though they do have their problems as well (taking more bytes, and hence lowering performance; being non-sequential and hence more prone to causing page splits on inserting).

Re: Three questions.

Re: Three questions.

3) Use IDENTITY with separate ranges at different servers. For instance, if you have three servers where rows are inserted, use IDENTITY(1,3) at the first, IDENTITY(2,3) at the second, and IDENTITY(3,3) at the third server. Thus, the first server will create numbers 1, 4, 7, ...; the second hands out 2, 5, 8...; and the third has 3, 6, 9, ...

Third Question

Currently I am working with Authentication Database and we are using Role Based Security Model for instance, here are the tables.

(snip)

When any module use User Level Permission I have to use UserPermissions table and when any module want to account level permission I have to use UsersAccountsPermission table but looks messy,

Any idea, how I can make it more robust design so down the road if any new scenario comes in I have to use same tables instead of three different tables.

Any thoughts?

I don't consider it messy at all. Too many people are afraid of having "many" tables. Don't be. If the data that has to be stored has many different functional dependencies, you're forced to either have many tables, or disregard the rules of normalization. And in my experience, any database that isn't properly normalized is much, much more messy than any database that is, even if the latter has hundreds of tables.

That being said, I don't know nearly enough of your requirements to verify whether these tables are really all needed. Like I said in my other message, getting the functional dependencies right is the hardest part – and one where newsgroup help often falls short, since they really depend on business rules and practices of the business you are working for. So you'll have to interview domain specialists and end users to get this done.

(Note that if you really want to learn about data modeling, you should read Terry Halpin's book on Object Role Modeling. At first, it may seem like a very roundabout way to get at a model for your database, but it really pays off in the long term).

—

Hugo Kornelis, SQL Server MVP
My SQL Server blog: http://sqlblog.com/blogs/hugo_kornelis

.

Re: Three questions.