Re: Concatenate 2 cells into 1, same record

Source:
http://www.tech−archive.net/Archive/SQL−Server/microsoft.public.sqlserver.programming/2004−08/0950.html

---

From: Hugo Kornelis (*hugo_at_pe_NO_rFact.in_SPAM_fo*)
Date: 08/05/04

Date: Thu, 05 Aug 2004 14:09:27 +0200

On Thu, 5 Aug 2004 12:17:41 +0200, Sjaakie Helderhorst wrote:

>Hugo,
>I probably took the wrong approach and based the uid on the subscribers
>name.
>See sql−script below for my current database (some Dutch terms).
>(INDEXES on telefoonnummer.callerid and vermelding.naam)

Hi Sjaakie,

No problem with the Dutch terms. I'm Dutch myself <g>. For the benefit of others reading this discussion, I'll keep using English in my reply.

I'm sorry if this sounds harsh, and I may even be very wrong (I have to make assumptions about things you know or can ask your employer/client). But this design looks as if it's just slapped together without paying too much attention to important details like normalization.

Your tables have no keys. I hope you just forgot to post these as well; if you really don't have them in your design, you'll soon be posting a "how to remove duplicates" message. (Yes, you did mention two indexes, but you didn't specify whether they are unique or nonunique indexes). Also, there are no foreign key constraints in the code, so the tables look completely unrelated.

Let's first take a look at the relation between the second and third table: Telefoonnummer (phone number) and Vermelding (listing). I can imagine that there are more listings for a given phone number (e.g. several people sharing a phone line) − do you store them all, or do you pick only one of the listings? Do you also want to store phone numbers that are not listed? And what about the other way around − some people have more than one phone number (people with ISDN, companies, etc.) Do you want to store these as one person/company and relate that one occurence to all phone numbers, do you store only one of the phone numbers or do you simply store the same name multiple times without bothering if one "Peter Jansen" is the same as another "Peter Jansen" or not. I can't answer these questions for you, as I don't know what business you're in. But the

answers have a great impact on how the database should look like.

Let's also look at the first and second tables: Adresgegevens (address) and Telefoonnummer (phone number). One might think that each phone number should be related with exactly one address, but I can imagine that some phone numbers are related to more than one address (companies!), or to no address at all (unknown [not listed], or not applicable, e.g. for cell phone numbers). And vice versa − should your application be able to handle multiple phone numbers at the same address? or addresses with no known phone number?

The answers to the above questions should suffice to get a correct logical design for your application's informatiion need. Translating that logical design into a physical table design is the next step. That's the moment when you can choose which of the possible natural keys to use, or you might choose to use an artificial key instead − either IDENTITY, or (only when you really need it) UNIQUEIDENTIFIER.

Best, Hugo

```
--
(Remove _NO_ and _SPAM_ to get my e-mail address)
```