

Re: Concatenate 2 cells into 1, same record

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-08/0983.html>

From: Sjaakie Helderhorst (*wont_at_tell.you*)

Date: 08/05/04

Date: Thu, 5 Aug 2004 16:02:15 +0200

"Hugo Kornelis" <hugo@pe_NO_rFact.in_SPAM_fo> schreef in bericht
news:fu54h097tac1ss168vf1h4cso5p7o9lbgt@4ax.com...

> *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 already assumed, by your name, you were Dutch too.

> *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*

I'm just searching for a callerid solution myself not for any employer or client.

I use callerid software, and I was looking for a way to show some more caller information instead of just a plain number.

You're probably also familiar with foondump (<http://www.foondump.nl>) which extracted all caller-data from the CDFoongids (Dutch Digital Phonebook). It has a built-in option for exporting data into a MySQL database, but I want to create a solution for MSSQL myself and, doing so, learn some more about it.

> *Your tables have no keys. I hope you just forgot to post these as well;*

I indeed did not post them. Ignorantly assuming keys were obvious to use.

Telefoonnummer.id and Adresgegevens.id are both related to Vermelding.id

microsoft.public.sqlserver.programming: Re: Concatenate 2 cells into 1, same record

- > *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 occurrence 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.*

<snip>

I'm in the Internet Development business, unfortunately I'm employed by a company which seldom uses SQL Server technology.

Somewhat ashamed, I must admit this database model is pretty crappy and, indeed, doesn't deal with things like subscribers with multiple phonenumber as mentioned in your remarks. Since this is just a private solution I didn't bother. However, you encouraged me to build this into a decent (from a developers point of view: correct) application.

- > *The answers to the above questions should suffice to get a correct logical*
- > *design for your application's information 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.*

I'll review my model and let you know.

Thanks for the support!