

Re: Help with Old Database – new requests

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.tablesdbdesign/2004-05/1040.html>

From: John Vinson (jvinson_at_STOP_SPAM.WysardOfInfo.com)

Date: 05/25/04

Date: Mon, 24 May 2004 23:18:25 -0600

On Mon, 24 May 2004 15:51:43 -0700, "Linda"
<anonymous@discussions.microsoft.com> wrote:

>I have an old database which they want to add some complex
>relationships to now:
>
>They want to put in: up-to ten Fee Codes (Fee Code Charged-
>1 to 10) by Vendor Contract Num by City – then lookup by
>Fee Code-X the fee amount (Fee-1 thru Fee-124) then sum
>those codes used and print or show in a report the
>specific code and \$-amount plus total \$ amount.....

WHOA!!!! Stop RIGHT there.

"Fields are expensive, records are cheap". If you have up to ten Fee Codes, someday you will have ELEVEN fee codes, and need to restructure all your tables, and all your forms, and all your reports, and all your queries. If you have a one to many relationship, *model it as a one to many relationship*, not as redundant fields.

>The specific one-relationship is: Vendor Contract Num +
>Vendor Name + City; everything else can be multipules:
> More than one Vendor Name per City.
> More than one City per Vendor Name.
> More than one Vendor Contract Num per Vendor Name
> More than one Vendor Name per Vendor Contract Num
>
>Vendor Table
> Autonumber PK
> Vendor Contract Num text
> Vendor Name text
> Vendor City text
> Fee-1 \$
> Fee-2 \$
> Fee-3 \$ to Fee-124 \$

That's where you went wrong (one of the places). Instead of ANY of these Fee fields, have a VendorFee table:

VendorID <Long Integer, link to the Autonumber>
FeeNum <Integer 1 to 124, or maybe to 125 or 131 tomorrow>
Fee <Currency>

VendorID and FeeNum would be a joint two-field Primary Key, and the table would have up to 124 rows for each vendor.

> *Fee Codes Table (New table I just setup)*
> *Fee Code text PK*
> *Fee Code-1 text*
> *Fee Code-2 text*
> *Fee Code-3 to Fee Code 124*

Same logic. A single table with FeeNum (1 to 124) as its Primary Key, and a single text field. Join it to the VendorFee table to link the fee currency value to the fee code text.

> *Transaction Table*
> *Autonumber PK*
> *Vendor Contract Num text*
> *Vendor Name text*
> *Vendor City text*
> *Fee Code Charged-1*
> *Fee Code Charged-2*
> *Fee Code Charged-3*
> *Fee Code Charged-4 to Fee Code Charged-10*

and again – you have a Many to Many link from Vendor Contracts to Fee Codes.

> *I may not have enough time to change all the tables to*
> *what they should be setup as – unless it can't be done any*
> *other way.*

Well, they should certainly NOT be set up the wide-flat way you are proposing!!

> *I am on my second year with MS Access, but just came to*
> *this company*
>
> *I would be grateful for constructive help (I understand*
> *what this messy looks like!!!!)*

Good luck introducing your coworkers to the wonderful world of normalized data structures!

John W. Vinson[MVP]
Come for live chats every Tuesday and Thursday

microsoft.public.access.tablesdbdesign: Re: Help with Old Database – new requests

<http://go.compuserve.com/msdevapps?loc=us&access=public>