

Re: Database Design

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.tablesdbdesign/2007-03/msg00082.html>

- *From:* Weste <Weste@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 6 Mar 2007 04:26:47 -0800
-

John,

Thanks for your response. If we use the staging tables to bind the data entry form and then delete the data from this table after it is entered how would edits to data be handled? Would you take the data from the normalized table and place it in the staging tables for editing? Isn't this approach causing more overhead and processing time – ie the data will constantly be going from a normalized to de-normalized state and visa versa?

In addition, the client wants the total budget dollars to be shown on the data entry screen. Not sure how that would be possible since the rates are based on job role, fiscal year, and month and the form is bound to a de-normalized table. A join can no longer be made between the budget table and the rate table to obtain the correct labor rate. I guess we could write some code to extract the monthly rates and store them in variables but again, isn't this adding alot of overhead and processing time to the application?

Thanks.

"John Nurick" wrote:

Hi Weste,

You are certainly right to demand a properly normalised data structure.

But as you know crosstab queries aren't updatable so can't be used behind a data entry form. And it's a lot of work to code an unbound form that works like a continuous form, especially if it also has to denormalise the data.

So if a continuous form is the best design for data entry, I wouldn't hesitate to use a denormalised "staging" table bound to the data entry form to hold the data as it's entered. Once the entry is confirmed, I'd use it to create or update the corresponding records in the "real" normalised tables, and delete the record from the staging table.

Re: Database Design

On Mon, 5 Mar 2007 16:55:08 -0800, Weste
<Weste@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

I am designing a database for a client to track budget data. The labor data is entered by FTEs (Full Time Equivalent Employees) and the dollars are calculated by multiplying the appropriate labor rate by FTEs. The labor rate is based on job role, fiscal year, and month. The client also wants to the ability for 12-month rolling budgets. I recommended the following table structures.

Table: LaborBudget

" LaborBudgetID
" ProjectID
" JobRoleID
" &.

Table: LaborBudgetDetails

" LaborBudgetDetailID
" LaborBudgetID
" FiscalYearID
" FTEs
" &..

Table: FiscalYear

" FiscalYearID
" FiscalYear
" FiscalMonthNumber
" &

Table: LaborRates

" LaborRateID
" JobRoleID
" LaborRate
" &.

The client has just completed 2 Access classes and doesn't agree with my design.

They want the table design as shown below because it will make coding of the data entry screen easier.

Table: LaborBudget

" LaborBudgetID
" ProjectID
" JobRoleID
" FiscalYear
" Jan
" Feb
" Mar
" Apr

Re: Database Design

" May
" Jun
" Jul
" Aug
" Sep
" Oct
" Nov
" Dec

They want the data entry screen to look like the design below.

Project Job Role Jan Feb Mar Apr May Jun Jul Aug Sep
Oct Nov Dec

I explained that this is not a normalized database design and will cause more problems down the road when building queries, reports, etc. Their response was then will have 2 sets of tables 1 for data entry (the non-normalized design) and then copy the data to the normalized tables for reporting. They don t want the normalized design because a crosstab or unbound form will need to be used for the data entry screen.

Can anyone provide anything that I can use to help convince them how to correctly design a database? Thanks for your help.

Weste

--

John Nurick [Microsoft Access MVP]

Please respond in the newsgroup and not by email.