

Re: Denormalizing for form only

Source: <http://www.tech-archive.net/Archive/Access/microsoft.public.access.forms/2008-05/msg00618.html>

- *From:* "Amy Blankenship" <Amy_nospam@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 10 May 2008 20:26:49 -0500
-

"Bob Quintal" <rquintal@xxxxxxxxxxxxxx> wrote in message
<news:Xns9A9AA7BA9D3B8BQuintal@xxxxxxxxxxxxxxxx>

"Amy Blankenship" <Amy_nospam@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in
<news:ul8gXjqsIHA.552@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>:

"Jeff Boyce" <nonsense@xxxxxxxxxxxxxx> wrote in message
<news:efBDHaisIHA.5832@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Amy

Although the Excel-like Thing1, Thing2, Thing3 approach is familiar to Excel users, as you already know, it isn't necessary (or desirable) to use this in an Access database.

What I've found quite useful for such one-to-many relationship is a main form/subform construction.

So your position is to force the client to adapt to how Access does things, rather than finding a way to force Access to present information in the way your client finds easiest to work with. That's interesting, but it doesn't really answer my question. My preference is to try were possible to do things in a way that my client prefers.

Amy,

you seem a little frustrated, but you are in error when you say

But there is a reason people keep asking questions like this, and it is that Access does a much better job of making it easy to

Re: Denormalizing for form only

make data entry forms with the structure above than with the "correct" structure.

That's not the real reason, the truth is that until people are familiar with the relational model, they tend to think "spreadsheet on steroids", and build the database based on that presumption.

The truth is that spreadsheets are easier for input, and that is why new designers tend to try to use it. Proper table design makes it way easier to get data out and is trivial to populate outside of Access, such as in a web form, but Access really fights you on it.

I'm wondering if there isn't something I've been missing, some wonderfully simple feature of Access, that makes using normalized data less painful from a form-building point of view.

Yes, what you've been missing is using a proper subform with properly defined relations between the tables and queries. Building proper data input and data editing form/subform sets becomes child's play once you learn how. The relational structure becomes transparent to the client.

I love the way that you presume that I don't know what I'm doing, and that I'm asking from a point of complete ignorance. I am asking as someone who has been working in Access for many years, and I was wondering if maybe there wasn't some feature I'd missed that makes data entry into a normalized table structure easy, since clients often don't want to pay for the hoops you have to jump through to do it in Access—they simply don't realize how difficult it is.

As to your preference in doing what your client prefers, I say it is just lack of confidence in asserting that he will prefer it once he's used to it.

OK, so tell me exactly how you'd make this structure easy to do data entry on. This is a real world situation that I find myself up against, trying to solve it in a way that isn't going to use hundreds of dollars worth of my time. The situation is that of a SAT score conversion table. If the subject matter is "writing" all scores must be indexed against the essay score. If the subject matter is "reading" or "math", the score is not indexed. So for any one given score, there can either be one data point, or seven. It's much easier to keep mental track of the scores that index to a

Re: Denormalizing for form only

particular "raw" score if you can enter them all in a row, across. At a minimum, it means you don't have to enter the same raw score seven times. Each practice SAT exam can have its own score conversion table for math, reading, and writing.

The table structure is this:

ScoreSet

=====

ScoreSetId–Autonumber, PK

SubjectName–reading, writing, math

ScoreSetDesc–will allow users to select this set later and associate it with an exam

ScoreSetItems

=====

ScoreID–Autonumber PK

ScoreSetID–FK to scoreset

WritingScore–score to index this on (will be 0 for subjects that don't apply, 0–6 for writing)

RawScore–the actual score on the multiple choice questions

SATScore–the scaled SAT score in the given subject

Your task, should you choose to accept it, is to create a form structure that uses no code and no left joins and is not a royal pain in the butt to enter all 7 data points for scores from –12 to 49 (434 data points PER exam just for writing, plus the 65–90 data points for the ones that are not as complicated) and that will be easy to use and intuitive.

Maybe that will explain to me what I've missed, and how this is so easy ;-).

–Amy

.