

Re: Still Struggling...

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.gettingstarted/2008-07/msg00452.html>

- *From:* "BruceM" <bamoob@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 18 Jul 2008 15:22:51 -0400
-

This will probably be my last posting for a while, as I will be away next week and need to finish up a few things before I go.

"Aria" <Aria@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:ED315AA0-A436-4BA4-9B12-565080680C37@xxxxxxxxxxxxxxxxxxx

Here's the structure, mistakes and all:

tblKeys 1:M tblMasterKeys
KeyID PK (Autonumber)
KeyName
MstrKeyID (FK, number long integer to tblMasterKeys)

I still think the relationship is between keys and locks, not keys and other keys. A key opens one lock. Ignoring master keys for the moment, you have One Lock : Many Keys, so LockID is a FK in tblKeys.

A master key opens several locks. One master key can open several locks, and each lock can be opened by several different keys including the master key. If there is to be a junction table I think it would be between locks and keys. Except for the master key this is a one-to-many relationship, but making it many-to-many even though the first "many" is applied infrequently is OK. There may be other ways to model this relationship, but this is one that occurs to me.

tblMasterKeys
MstrKeyID PK (Autonumber)
MstrKeyName

tblKeysEmployees M:M tblEmployees
KeyID (1/2 PK, FK to tblKeys, number, l.i.)
EmpID (1/2 PK, FK to tblEmployees, #, l.i.)
AllowedtoRetain Yes/No
Approvedby
DateIssued Date/Time
DateLost Date/Time

Re: Still Struggling...

DateRtrnd Date/Time

The question here is how the lock fits into the picture. If you may want to see a listing of who can open a particular lock you need a way to associate locks with employees.

tblKeysRequests 1:M tblKeysEmployees
RequestID PK (Autonumber)
KeyID FK, #, l.i. to tblKeysEmployees
QtyRqstd
DateRqstd Date/Time
DateIssued Date/Time
DateIssued (Hmmm...I just noticed that this is the same as tblKeysEmployees)
Funny how you can gloss right over things.
Comments

tblLocks (I thought this was a junction table but now I don't know)
KeyID PK/FK
LocationID PK/FK

It is not a junction table that I can see. One lock has many keys, not the other way around (except for master keys). In the case of a master key it is one of the many keys that can open a lock.

tblLocations 1:M tblLocks
LocationID PK (Autonumber)
LocationTypeID FK to tblLocationTypes
LocationName
Remarks

tblLocationTypes 1:M tblLocations
LocationTypeID PK (Autonumber)
LocationType

tblCampuses 1:M tblLocations
CampusID PK (Autonumber)
CampusName

tblWings 1:M tblLocations
WingsID PK (Autonumber)
WingName

You may be cutting it a bit fine here. I have to disagree with Sean that all fields need to be filled in. It happens all the time that a Middle Initial field is not filled in, for instance. For another example, Apartment Number does not apply if the person lives in a house, but I see no reason in a database of this scale (i.e. not super large) to separate this datum into its own table. Find a way to identify where the lock is located. A lock will be located on a campus, and maybe in a building (unless it is an outdoors lock to an athletic field gate or

Re: Still Struggling...

Re: Still Struggling...

something). If a building, it may have wings, or maybe not. You can use a lookup table to insert a value into, say, the Wing or Building field in the Locks table. You can even limit the Wing listing to just the Wings in the selected Building. However, you may not need to involve these lookup tables into relationships.

So I think tblKeysEmployees defines the relationship between keys and employees.

Regarding other parts of your previous post:

The complexity here is that one person may be assigned many keys, but a key can be assigned to only one person.

No, this is not quite true. One key can be assigned to one or many people.

One solution here may be to have a LockPerson junction table (one person many locks and vice versa).

...*another* junction table?! Please...is there any way around this?

This is in place of the KeyEmployee junction table, which I maintain is not exactly the relationship you should have.

This would be the source for a subform on the Locks form. A list box could contain >a listing of unassigned keys for that lock. The list box row source would have to be >built as you go, since the available keys are always changing. When somebody >needs to be assigned a key you would go to the Lock form, see the listing of >available keys, and created a new LockPerson record that stores the KeyID, >EmployeeID, and maybe AssignedDate and other details.

It would be nice to have a list of unassigned keys but...it's a little overwhelming right now. This just keeps growing and growing. I'm already onto page 2 for printing the relationship diagram. Let me live with this for awhile. I know yesterday I posted, "Oh well, c'est la vie" as far as junction tables but today I can't deal with another junction table.

You would use SQL to insert a list into a list box, or maybe into a continuous form. You do not need another table. The data are already there (or will be when this is all set up).

Re: Still Struggling...

BTW:

tblKeys
KeyID (PK)
LockID (FK to tblLocks)
KeyCode (S2-A, etc., or whatever)
AssignedTo (FK to tblEmployees)
AssignedDate
Retired (Yes/No)

I like the field name KeyCode. I'm not so sure about the Retired (Yes/No) field although that does have to be taken into account. I will probably end up doing it that way. Let me think about it some more.

I meant Retired to refer to the key, in case that is not clear. The point is that if a key is lost or damaged or worn out the Retired field can be used to exclude it from a listing of available keys.

Let me know if this structure resolves **any** of the issues we have.

Aria W.

Perhaps Sean will have some further insight, and perhaps he and I are not at cross purposes. I repeat that I think it will be a good thing to take what you have and start another thread. I would have asked some questions of the group by now, as I am uncertain in some cases how to model this situation. Instead I have been stretching my mind and my capabilities trying to figure this out. It has been good practice, but I am about at the limit of what I can suggest with confidence.

I will check in again, or search for other threads from you, when I return. In the meantime, best of luck. I am confident you will get this figured out, even if you doubt it from time to time.

"BruceM" wrote:

Is the vault key just like any other key for a particular lock, except that it is in the vault? If so, is the vault copy essentially a clean copy to be used as the master for additional copies? If so, you can identify the vault key either by an extra field (maybe a Yes/No Vault field) in tblKeys, or you can "assign" the key to the vault just as you would assign it to a person. To do this you could add (Vault) to the top of the drop-down list of Employees to whom keys are issued, similar to the way some drop-down lists show (All) at the top. I won't go too far down that road until I hear back from you.

Re: Still Struggling...

Regarding the Locks table, I don't think it would be a junction table. A lock is an entity with certain characteristics such as brand, location, date installed, and so forth. One Lock can have many keys. If Master Keys are included in the thinking, one Key can be for many Locks, but in this case the junction table would be tblKeyLock or something like that.

Regarding tblKey, I expect there should be a Lost field or something like that. There is no need to keep an active listing of keys nobody can find. Regarding the question of natural key or surrogate key (any "artificial identifier", including autonumber), it really doesn't matter as long as the "natural" number such as SA-2 is used once for one physical (metal) key, and never used again. If you are not sure this is the case, autonumber would be a simpler choice as the PK. The same idea comes into play with the Lock identifier, I expect. If they change the lock in such a way that the old keys can be used you will need to update the FK of those keys so that they are associated with the new lock.

From an earlier post you wrote:

> Suggestions still on the table:

> 1. Create tblLocations to replace tblRooms

Sounds like a good idea

..

> 2. Create tblLocks to define the relationship between keys and locks. >

There

> should be a 2 field PK involving Key ID and LocationID.

Already discussed. Let me know if something is unclear, or if I am missing something

> 3. Create Master Key table to account for the special attributes of > Master

> Keys.

One possibility is to see the situation as One Lock > Many Keys and One Key

> Many Locks, in which tblKeyLock is needed to resolve the relationship for all locks and all keys. However, I think a separate table for MasterKeys and a junction table tblKeyLock would be simpler to manage. If you are looking at a Lock record you would have a subform listing the ordinary keys and the person to whom they are assigned (some may not be assigned at all, I expect). One Lock > Many Keys, so there is a 1:M between tblLock and tblKey, and the subform is based on tblKey. Another subform based on tblKeyLock could list the master key holders.

One lock could have keys assigned to many people, and each person could be assigned keys. Another consideration is that a Lock record should have a listing of available keys. If it was me I think I would have an AssignedTo field in tblKeys:

tblKeys

KeyID (PK)

LockID (FK to tblLocks)

KeyCode (S2-A, etc., or whatever)

Re: Still Struggling...

AssignedTo (FK to tblEmployees)
AssignedDate
Retired (Yes/No)

If I wanted to list people and the keys they hold I would use a query.

This is not necessarily the best design in that AssignedTo and AssignedDate are not really attributes of keys. The complexity here is that one person may be assigned many keys, but a key can be assigned to only one person. Similarly, one lock may have many keys, but other than master keys a key may open only one lock. On the other hand, one person may open many locks, and a lock may be opened by many people. By the way, it just occurred to me that if there may be several locks keyed to accept the same key you may need to add a LockLocation table related to tblLocks to take care of this detail. One solution here may be to have a LockPerson junction table (one person >>

many locks and vice versa). This would be the source for a subform on the Locks form. A list box could contain a listing of unassigned keys for that lock. The list box row source would have to be built as you go, since the available keys are always changing. When somebody needs to be assigned a key you would go to the Lock form, see the listing of available keys, and created a new LockPerson record that stores the KeyID, EmployeeID, and maybe

AssignedDate and other details.

It may be worth your while to start a new thread on this specific topic. All you would need to say is that you have:

A Locks table containing the LockId, LockLocation, etc.

A Keys table containing KeyID (PK), LockID (FK), KeyCode, etc.

An Employee table

Explain that you understand a Lock may have many keys, and that a person may

be assigned many keys. However, you are unsure how to store the AssignedTo

information for Keys. Should it be in the Keys table? If not, how is that relationship modeled.

I suggest this because frankly I am unsure how best to proceed on this point, and in a new thread you would attract the attention of very experienced designers.

4. Create additional look-up tables: tblCampuses, tblWings and tblLocationTypes.

Sounds good.

"Aria" <Aria@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:110BDA30-102F-4449-9041-7766EAFAC9C1@xxxxxxxxxxxxxxxxxxxx

Re: Still Struggling...

>I think I may have a problem that I need to work through. We can discuss
>this
> when you have time. I just wanted to get this down before I lose track >
>and
> start dealing with other issues. I think we are going to need to > re-visit
> an
> issue from the very first post: vault keys. Originally, we decided that
> the
> situation was unclear and dropped it. In looking through some of the > data
> that I copied to help map my direction, I just noticed something that I
> probably saw before but did not grab my attention as it has now. > During
> the
> re-key, among the many replacement keys made, one key was important >
> enough
> to
> be "assigned" to the vault. There are probably others, but that > happened
> before my time so I am not aware of the previous history. This does >
> change
> things a bit. I'm going to have to account for not only that key, but > all
> of
> the vault keys.
>
> I have two trains of thought right now:
>
> The first is that the vault key is a sub type key that will need its > own
> table. I don't know about this because it doesn't seem to fit the model > of
> a
> sub type (at least not as far as I can see). It's not like the masters
> which
> are actually a different type of key. Their attributes are > significantly
> different. Not so with the vault key; there's nothing special or > different
> about them.
>
> The second thought is that this is really a location type and will need > to
> be included with tbllocations or is it tbllocationtypes. I'm still > working
> on
> this because I'm unsure right now. It does seem to fit the "where" > model
> though.
>
> Just throwing it out there...
>
> -- > Aria W.
>
>
> "Aria" wrote:
>
>> > I don't have much time to reply today, but I have noted a few things
>> > inline.
>>
>> Please, don't give it a second thought. I appreciate all of the >> support
>> you

Re: Still Struggling...

>> have both given me. I understand.
>>
>> > If you select the combo box, open the Property Sheet (with the tabs >>
> for
>> > Format, Data, etc.), select an event, click the three dots, click >> >
Code
>> > Builder, and click OK the name of the control will be filled in for >> >
you
>> > in
>> > the code window. If you type it yourself you need the underscore. >> >
I
>> > think
>> > that is so there are no spaces in the procedure name.
>>
>> I'll check on that.
>>
>> > There's no definitive answer for that, but anything that involves
>> > junction
>> > tables is a lot to get your mind around if you are just starting.
>> > Before
>> > long, if it hasn't already happened, the need for junction tables, >> >
and
>> > how
>> > to use them, will be clear in your mind. In your case there were a
>> > number
>> > of junction tables, and a situation that involved keys, locks, >> > people
>> > to
>> > whom the keys were issued, locations of the locks, campuses, and a
>> > number of
>> > other entities, with a variety of relationships between the >> > entities.
>>
>> Yeah, no kidding. I started thinking about that because I was trying >> to
>> avoid adding yet another junction table. I recall a previous >> discussion
>> where
>> we decided that the room phone could go into tblLocations. That would
>> leave
>> district cell phone orphaned and I could not figure out where to put >> it.
>> It
>> doesn't belong in tblLocations and doesn't belong in tblEmployees so I
>> figured why am I drawing the line at this junction table when there >> are
5
>> others just like it. Oh well...c'est la vie! I keep looking at the
>> relationship diagram and I 'm concerned how I will manage all of this. >>
In
>> particular, what kind of form to design and whether I should have
>> another
>> subform for tbKleysRequests on frmEmployees. I'm trying not to borrow
>> trouble; one thing at a time.
>>
>> > What became of tblLocks? If one lock may have several keys, locks >>
are

Re: Still Struggling...

>> the
>> > top level.
>>
>> "A lock can have several keys". There have been a number of turning
>> points
>> in this journey. You both have dropped statements that on the surface
>> seemed
>> inconsequential ,but in reality had a major impact. That is one thing >> I
>> am
>> *never* going to forget. You made 3 on the mark statements in that >>
>> post.
>> tblLocks is still here. It*is* a junction table, correct? It sure >> looks
>> like
>> one.
>> > > Suggestions still on the table:
>> > > 1. Create tblLocations to replace tblRooms.
>> > > 2. Create tblLocks to define the relationship between keys and >> >
>> locks.
>> > > There
>> > > should be a 2 field PK involving Key ID and LocationID.
>> > > 3. Create Master Key table to account for the special attributes >> > >
>> of
>> > > Master
>> > > Keys.
>> > > 4. Create additional look-up tables: tblCampuses, tblWings and
>> > > tblLocationTypes.
>>
>> > > Questions/Comments:
>> > > 1.Decision--Unbeknownst to you and Beetle, I was going to use a
>> > > natural key
>> > > for KeyID in tblKeys. Your statement about a possible re-key and >>
>> > > our
>> > > history
>> > > of such finally made me realize that this was a *lousy* idea from >>
>> > > the
>> > > beginning. I'm switching to autonumber.
>> >
>> > There's nothing wrong with a natural key, depending on what it is.
>> > With a
>> > surrogate key such as autonumber the main thing is that the record
>> > needs to
>> > be unique for reasons apart from the autonumber field. If two >> >
>> records
>> > are
>> > identical except for an artificial identifier then they are not >> > really
>> > unique. The autonumber is a convenience, but cannot by itself >> >
>> enforce
>> > what
>> > is known as a unique constraint.
>>
>> Won't this cause a problem? With the re-key, they not only changed the

Re: Still Struggling...

>> lock,
>> they changed the lock identifier. So if the key was previously, let's
>> say,
>> SA-2 , it may have been changed to XJ-3A after the re-key. If I used >>
this
>> as
>> my primary key, I would have thought this would be a nightmarish
>> situation.
>> No? If it happened once, there's always the possibility that it >> *could*
>> happen again. Keys are lost pretty much on a weekly basis. What do you
>> think?
>> Do I need an inactive button for keys? I don't know exactly *how* this
>> would
>> work.
>>
>> Let me know what you think about that and the master keys. Don't >>
concern
>> yourself if you don't have time. I have problems in the db that I need >>
to
>> straighten out. Thanks so much for your time Bruce.
>>
>> -- >> Aria W.
>>
>>
>> "BruceM" wrote:
>>
>> > I don't have much time to reply today, but I have noted a few things
>> > inline.
>> >
>> > "Aria" <Aria@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
>> >
>> > news:BC0C0F83-14A9-4830-88EC-BB2EBF73DA6C@xxxxxxxxxxxxxxxxxxxx
>> > > I found the compile button. It was under commands so I added it to >>
>> > the
>> > > toolbar. Thanks.
>> >>
>> >>> I will take a look at your questions, but you will need to sort >> >
>> > of
>> >>> start
>> >>> over with your descriptions, as I have not been following very
>> >>> closely of
>> >>> late. When you do so, list only the essential fields.
>> >>
>> >> Not quite ready to list the table structures for this part
>> >> yet...still
>> >> stinging from the last time when they were all wrong. I just
>> >> basically
>> >> want
>> >> to think on post. Ask a few questions and have you weigh in on >> >
>> > where
>> >> I'm

Re: Still Struggling...

>>>> going wrong. Are you Ok with that? I completely understand what >>>>
>> you
>>>> mean
>>>> as
>>>> far as having lost track. I have all my notes and posts and I >>>>
still
>>>> have
>>>> to
>>>> read up on what's going on and what transpired.
>>>>
>>>> Questions unrelated to 2nd half of db:
>>>> 1. When I input the programmng code for the AfterUpdate event, I
>>>> input
>>>> cboStaff_Lookup. The line highlighted in yellow said, "Private Sub
>>>> cboStaff_Lookup_ AfterUpdate () Was I supposed to input that final
>>>> underscore
>>>> after Lookup?