

## 4. Relationships, Referential Integrity, Indexes

### Steps to Document Table Structure using Relationship Diagram

To make it easier to see your tables and relationships

1. Enforce referential integrity on all relationships (unless you have a good reason not to)
2. Put the table for the "1" side of the relationship on the left and the "many" side on the right. Because we read from left-to-right, it is more appropriate to visualize creation of data that way.

... data diagrams should FLOW, just as data does -- in the same direction so it is easier to visualize (very important to get the best ideas)

A trick if you have a field list longer than your screen (to get all fields to show) is to put it on the relationship diagram twice. Place the copy next to the first one (I usually place it to the right and line it up with the bottom). Adjust the height of the copy and the scrollbar position to show the fields at the end -- then take a screen shot.

To print it out, this is a good method:

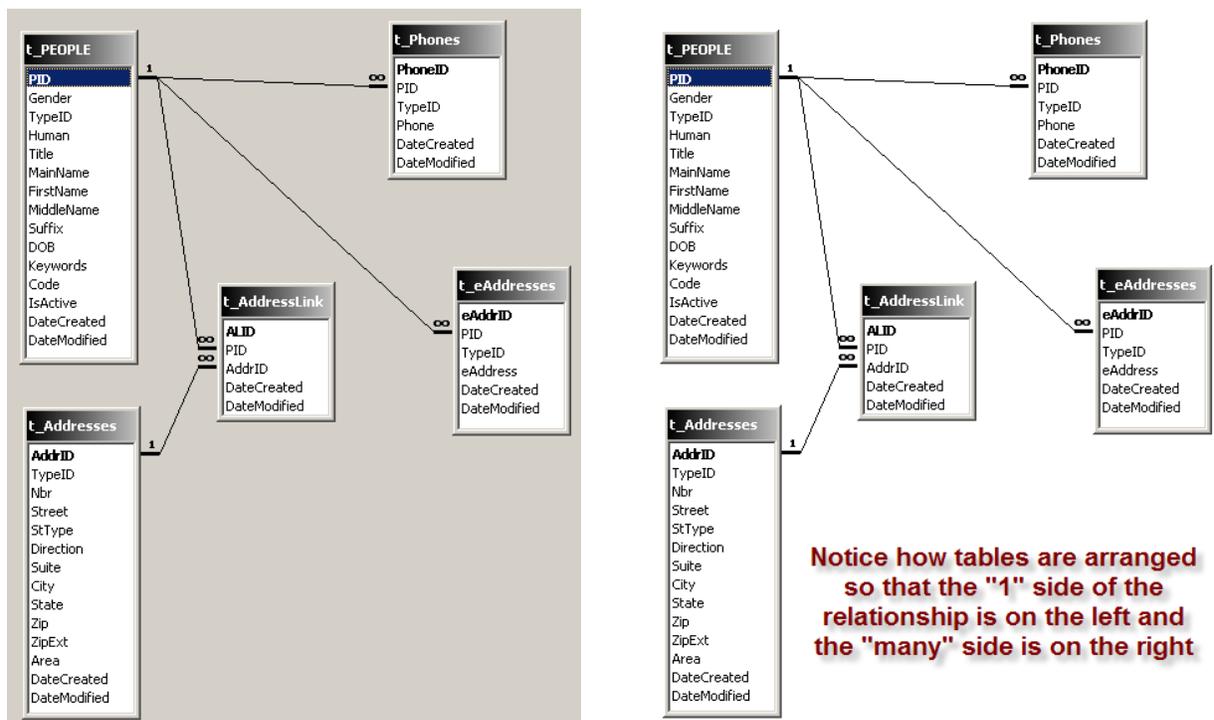
3. Stretch all your field lists to show all fields and put all tables on even if they don't have relationships to anything

4. Press **Print Screen** to copy screen to Windows clipboard

5. Open Paint ([Start, Accessories, Paint](#))

6. Paste **Ctrl V**

(at this point, I like to select part of the image, copy the selection, and make a new Paint document)



Crop the diagram so there is nothing extra in the picture

Change the background to white, it is easier to write on your diagram

## 7. Save As...

- file type = JPG or PNG or GIF
- filename = "Rel\_dbname.jpg" -- just keep overwriting
- location: put it into the directory with your database

A JPG file is about 1/10th the file size with no visible loss of resolution (as long as you don't make it bigger) compared to the default format of bitmap (BMP).

## 8. Then, you can open Word or PowerPoint and **Insert, Picture ► From File...**

You can also skip steps 4-8 and print your relationship diagram from the relationship window using **File, Print Relationships...**

If you have a program like *SnagIt*, by TechSmith, you can skip a lot of these steps. *SnagIt* was used to do all the screen captures for this publication.

The reason I like to save the relationship diagram as an image(JPG or PNG or GIF) file is because I prefer to use *PowerPoint* so that I can add comments to the diagram -- also, most of the databases I work on have too many tables for one screen. Allen Browne also has a utility which is quite useful if you do not have a zillion tables in your database <smile>.

## **Relationship Report With Extended Field Information By Allen Browne**

<http://allenbrowne.com/AppRelReport.html>

Keep a relationship diagram in front of you at all times whenever you work on a database, it is an invaluable reference!

Once your relationships are printed, put the printouts where you can spend time studying them ... by your bed, in the bathroom, in your briefcase, etc.

Keep your relationship diagram in front of you whenever you are designing or developing the database. I like to put them in plastic sheet covers with cardboard behind them so they stand up better. *Before* any forms or reports are built, which is like building a first and second story of a house, the foundation has to set. The foundation of every database is the table structure and relationships.

## **Steps to take if you can't Enforce Referential Integrity**

1. Change the **DefaultValue** for numeric foreign key fields to Null instead of 0 (or just delete it)

Access sets the **DefaultValue** of numeric fields to 0 -- this is not good for a foreign key because there won't be a sequential auto number field to match it to -- and if there *is*, it probably won't be right. If you want to ensure that the user fills out the data, use the form **BeforeUpdate** event (used for data validation) so that you can issue an understandable message.

2. Delete the 0's out of the foreign key data in your tables

This should take care of the problem

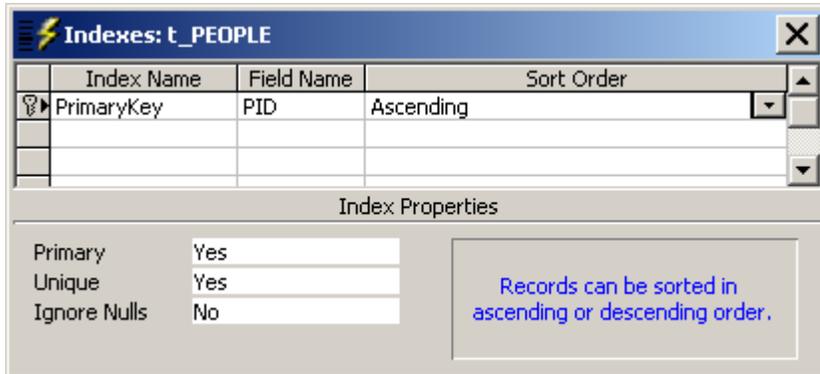
Now that you have the relationships more readable ... in addition to making the diagram easier to read, the purpose for enforcing referential integrity is so that a main record that has children does not get deleted leaving an :( orphan ...

## **Indexes**

Another consideration is how many Indexes exist -- and what Access creates automatically if you don't stop it. Each time you name a field with **ID** or **Code**, **Num**, or **Key** in it, Access creates an

**Index.** This is like the index in the back of a book -- makes things easier and faster to find. But each time a record is changed, all the indexes have to be updated. Whenever you are designing your tables, keep the indexes window open and delete extra indexes that you don't have a specific purpose for. Click the lightning bolt icon  or from the menu → [View, Indexes](#)

**Figure 4-1 Index Window showing Primary Key**

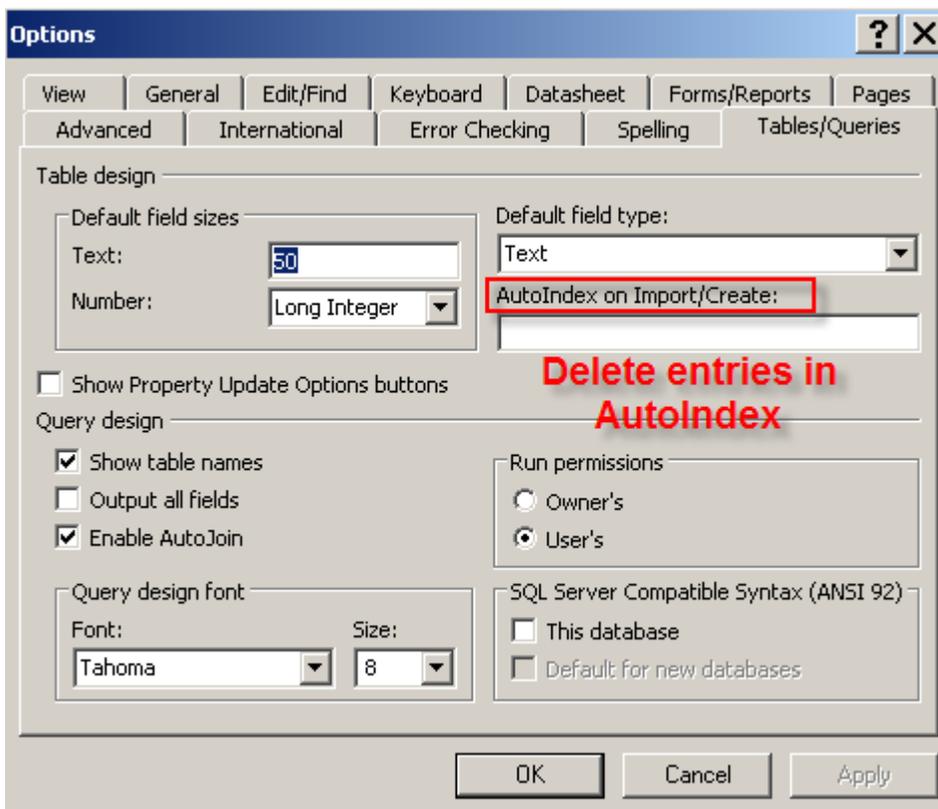


Change the default so that Access does not create so many indexes while you are creating your tables (there IS a limit to how many a database can have).

From the menu in the database window

--> [Tools, Options, Tables/Queries](#) --> [AutoIndex on Import/Create](#)

**Figure 4-2 Tools, Options - Delete AutoIndex Entries**



... Personally, I delete everything there, so Access doesn't create any indexes for me. If I want them, I will make them myself.

If you are using Access 2007, or want more information on this, See this gem tip:

### Prevent Automatic Indexes -- By Tom Wickerath

[http://www.access.qbuilt.com/html/gem\\_tips.html#AutoIndex](http://www.access.qbuilt.com/html/gem_tips.html#AutoIndex)

I generally delete any index (or just make sure it is never created!) that is not needed for the Primary Key or a Unique Index. A table can have 32 indexes. It best NOT to create indexes on foreign keys that will be used in a relationship that enforces referential integrity (RI) because they will be redundant. Each time you create a RI relationship, Access creates a hidden index – and these hidden indexes count toward your limit!

## Unique Indexes

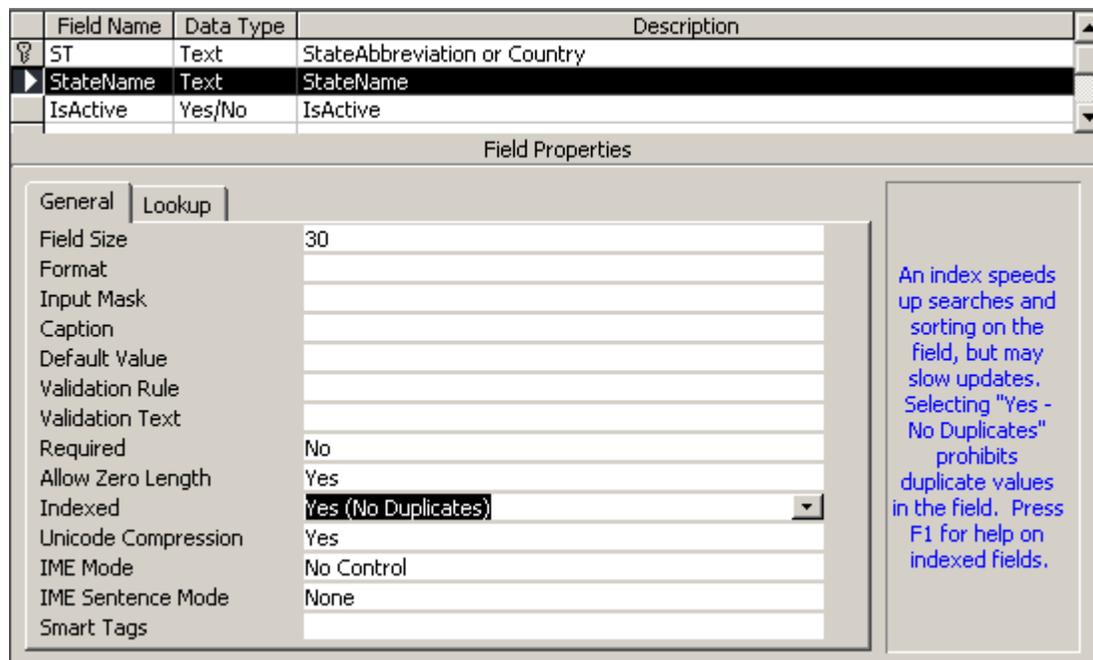
Sometimes, you want to make sure certain fields are not repeated, such as the user value for CustomerCode or a ChartNumber for a patient. In these cases, make a UNIQUE index on that field by setting the Index property.

Figure 4-3 shows a unique index set on StateName in the States table.

Indexes --> Yes (No duplicates)

Primary Key indexes are always unique.

**Figure 4-3 Unique Index on StateName**



## Steps to make a Multi-Field, Unique Index

For resolution of the problem allowing a duplicate combination into the table that must be unique:, make a unique index in the table on a multi-field combination

1. From the table design, turn on the Indexes window (from the menu: [View, Indexes](#))
2. [Click](#) on the first field in the table design and, in the lower pane, set the [Index](#) property to → [Yes \(No Duplicates\)](#)

that will add a line to the [Indexes](#) window and the [Unique](#) property will be Yes.

3. In the row just below the index you just made, in the 2nd column of the indexes window, **Click** in the fieldname column, drop the list, and choose the 2nd fieldname
4. If you have another field to add, **Click** on the fieldname column in the next row down and choose the 3rd fieldname

A Unique Index will disallow duplicate combinations into the table

Using a Unique index is very helpful when you use Append queries to create data such as:

1. creating test question records for each student so you can quickly fill in the data
2. adding oil & gas production for a well, product, and a day
3. creating survey detail records for each person

This ensures that, even if the query to add records runs twice, duplicate records will not go in.

Figure 4-4 shows a multi-field unique index on the combination of three fields: PID, TypeID, and Phone.

**Figure 4-4 Multi-Field Unique Index**

