II: Further use of Access: relational power, data entry and data coding

Jeremy Boulton (Newcastle University). Email: Jeremy.Boulton@ncl.ac.uk

Database Course for the ESRC Newcastle-Durham DTC

This manual, related databases and other material are available to download at:

http://www.staff.ncl.ac.uk/j.p.boulton/esrcdtcdatabasetraining.htm

Contents:

Introduction

This second part of the course assumes you have completed the first part, and that you are now familiar with some Access basics. You should be able to create tables, construct simple queries, use filters and so on. The examples here relate to Access 2007.

Data entry and how to do it efficiently

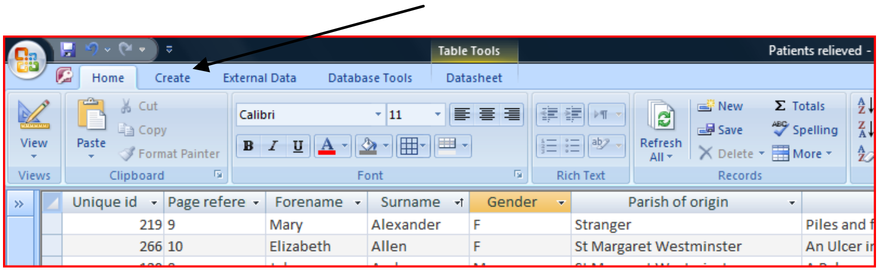
Using and creating forms

You can, and sometimes should, enter data *directly* into tables. However, if the data is complex, with many fields of information, you should use forms. Forms are for *entering data* in Access. Forms can include all the fields in your table, or only a subset.

Creating forms in Access is simple.

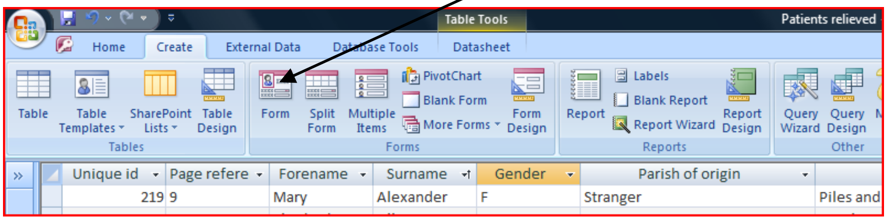
Open the Westminster Infirmary Database. Although there is already a form, you can still create a new one all by yourself, simple by opening the main table of the database, and clicking on the form button.

1. First click on the *Create* tab on the Patients relieved table



Unique id	Page refere	Forename	Surname	Gender	Parish of origin	
219	9	Mary	Alexander	F	Stranger	Piles and f
266	10	Elizabeth	Allen	F	St Margaret Westminster	An Ulcer ir

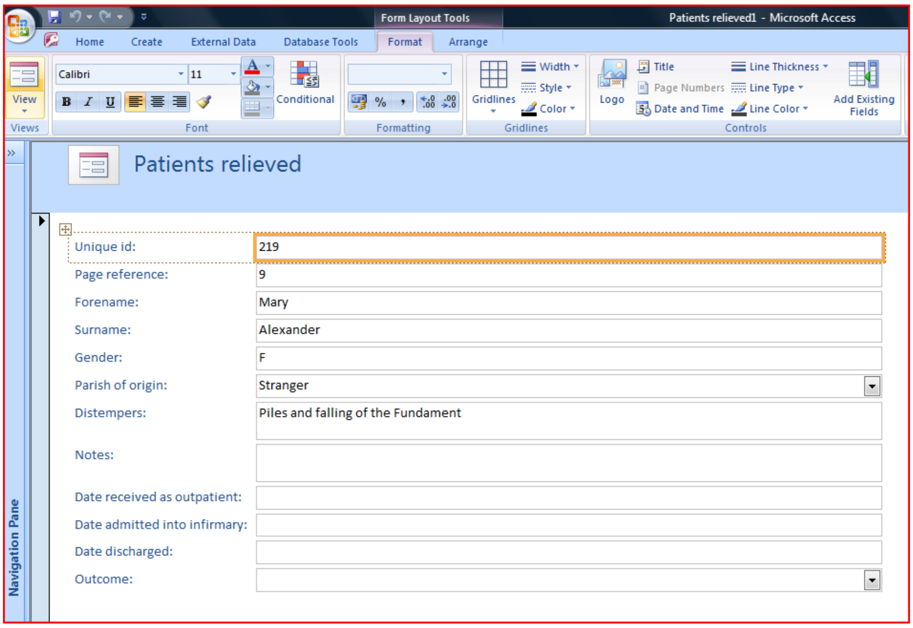
2. Then simply click on the *Form* button.



Unique id	Page refere	Forename	Surname	Gender	Parish of origin	
219	9	Mary	Alexander	F	Stranger	Piles and

3. **This will generate a form** based on the currently open Table. You can save it, if you like. Give it any name you like. You can change the layout of the form by clicking on the *Layout* button. You can completely redesign the form, including changing appearance, size and displayed fields, by clicking on the *Design* button. In general, a form created using the form button is bound to the underlying characteristics of the original table. If you subsequently change the table, such as by adding a look up field, you would either have to edit the form, or, more sensibly simply delete it, and create a new one.

It is often quicker to delete forms and create a replacement rather than redesigning them.



Patients relieved

Unique id:	219
Page reference:	9
Forename:	Mary
Surname:	Alexander
Gender:	F
Parish of origin:	Stranger
Distempers:	Piles and falling of the Fundament
Notes:	
Date received as outpatient:	
Date admitted into infirmary:	
Date discharged:	
Outcome:	

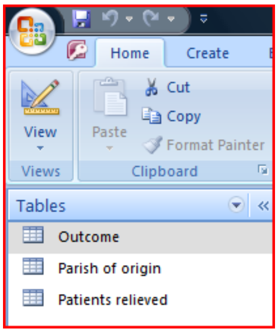
Forms can be complex. You can add sub-forms, based on complex queries and related tables. Throughout, however, remember that forms are really for data entry. You can edit, filter and sort data in forms, just as you can in a table.

Forms are not necessary for data which is relatively simple, with little text and with relatively few fields. They are useless where data is primary numerical.

Don't create forms for the sake of it. Actually, throughout all doctoral research, don't create *anything* for the sake of it.

Return to contents page

Drop down menus to help you enter data consistently



You will notice that the Westminster Infirmary *Patients Relieved* Table (and associated form) has two drop down tables. They are on the *Parish of Origin* field and the *Outcome* field. These are based on two supplementary Tables, which exist in the database, named, with staggering originality: *Parish of Origin* and *Outcome*.

Drop down tables are one way of *standardizing text data*. They are not necessarily ideal, since if you use them to enter historical data (rather than in a field to code existing data) you are not entering the original text string. This is *only* acceptable if you are entering unambiguous text, such as, say, a London parish which you can identify, but which might have multiple spellings or other variations. Clearly, to create such drop down tables you need to have some idea about what data your table should contain.

You can, however, add records later to extend the range of values in the drop down table. The *Outcome* field is particularly suitable for a drop down table since there were only a limited number of outcomes.

Drop down tables help to speed up data entry (since typing is eliminated), give you a standardized result (no spelling differences or mistakes) and should always be used where practical.

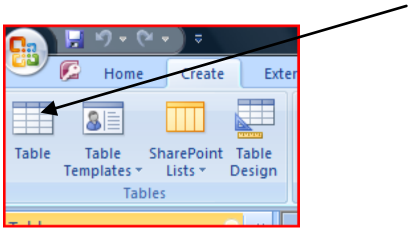
Drop down tables can also be used to standardize text data, *by inserting a new field next to the original text field.* This is useful, particularly if you can decide easily - as you enter data - into which category each text string belongs. It is possible, but more time-consuming, to use drop down tables to codify text fields after all data entry has been carried out. It is probably easier, however, to use *make table queries* to codify data in such cases (see below, **data coding and data standardization**).

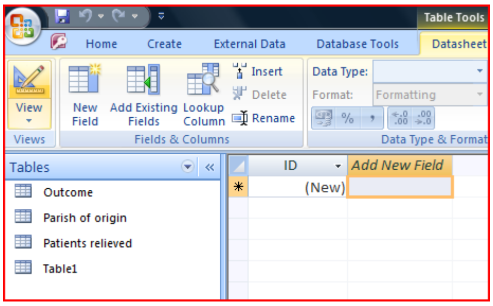
How to create a drop down table

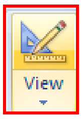
Let's create a new drop down menu in the Westminster Infirmary database. Just for fun, we will create a drop down menu for the gender field. This is simply to illustrate the procedure. There are, after all, only three possibly alternatives i.e. F, M and A (unknown gender). About one in every one hundred thousand entries you might come across hermaphrodites, but let's ignore that remote possibility.

First create a small table *with only one text field* to contain the drop down values.

1. *Create* then click on the Table button





2. Then click on Design View and  save the table as something like Gender.

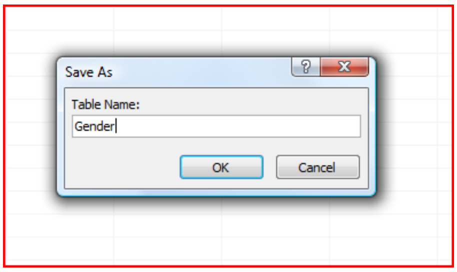
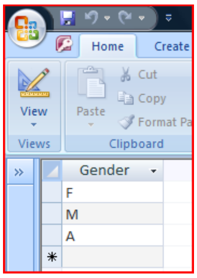


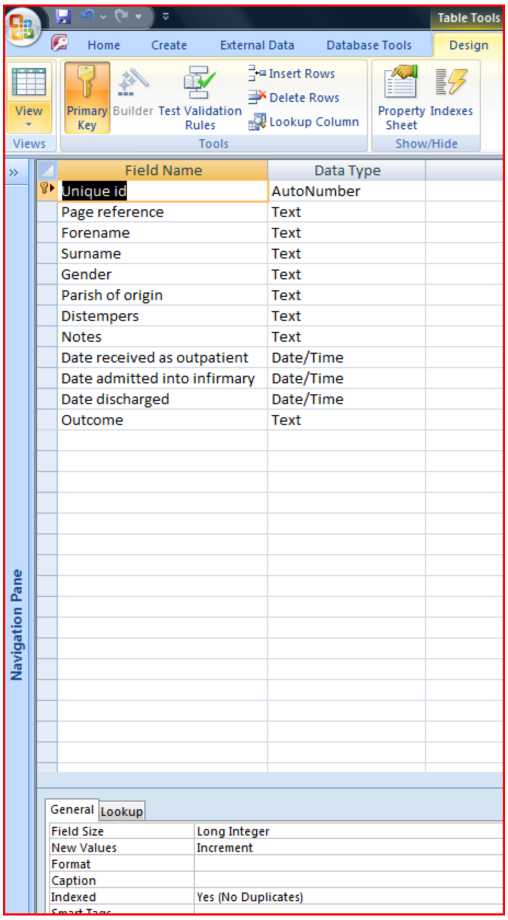
Table Name: Gender

3. Then delete the ID field (you don't need it) and create a text field, called Gender with a field size of just 1. Save the table, and *then enter three rows of data*, F, M, A.



Gender
F
M
A

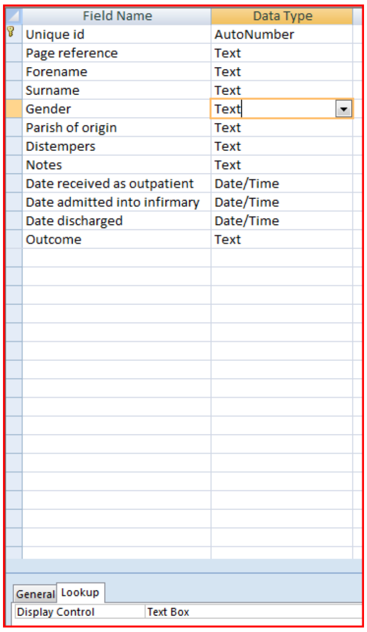
4. Now you have your drop down table. Next you create a look up table in the *Patients relieved* Table.
5. Open the *Patients Relieved* Table, and click on the *Design View* button. You should see something like the following:



Field Name	Data Type
Unique id	AutoNumber
Page reference	Text
Forename	Text
Surname	Text
Gender	Text
Parish of origin	Text
Distempers	Text
Notes	Text
Date received as outpatient	Date/Time
Date admitted into infirmary	Date/Time
Date discharged	Date/Time
Outcome	Text

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

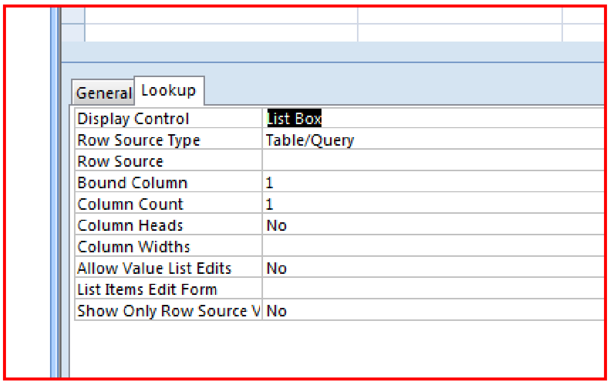
6. Click on the *Gender* field, and then on the *Lookup* tab at the bottom of the screen.



Field Name	Data Type
Unique id	AutoNumber
Page reference	Text
Forename	Text
Surname	Text
Gender	Text
Parish of origin	Text
Distempers	Text
Notes	Text
Date received as outpatient	Date/Time
Date admitted into infirmary	Date/Time
Date discharged	Date/Time
Outcome	Text

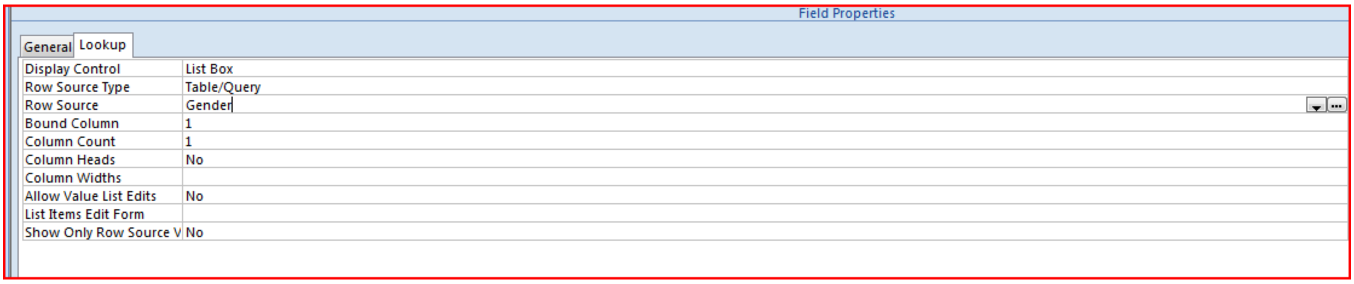
Display Control	Text Box

7. On the Display Control row, choose List Box from the drop down menu (there are only three options, Text Box, List Box or Combi).



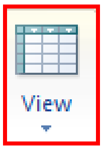
Display Control	List Box
Row Source Type	Table/Query
Row Source	
Bound Column	1
Column Count	1
Column Heads	No
Column Widths	
Allow Value List Edits	No
List Items Edit Form	
Show Only Row Source V	No

8. Leave the *Row Source Type* as Table/Query and select your *Gender* table from the drop down menu in *Row Source*

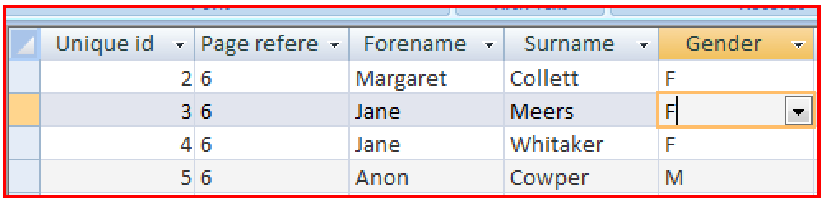


Display Control	List Box
Row Source Type	Table/Query
Row Source	Gender
Bound Column	1
Column Count	1
Column Heads	No
Column Widths	
Allow Value List Edits	No
List Items Edit Form	
Show Only Row Source V	No

9. You have finished! Hit the View button, and save the table when prompted.



10. You should now find a drop down box in the gender field of your *Patients relieved* field



Unique id	Page refere	Forename	Surname	Gender
2	6	Margaret	Collett	F
3	6	Jane	Meers	F
4	6	Jane	Whitaker	F
5	6	Anon	Cowper	M

This is actually a quick procedure, once you get used to it. Obviously you can do much more sophisticated things with drop down boxes. You can use queries, and generate calculated fields, for example. However, since the main aim of this course is to KEEP IT SIMPLE this technique will probably be as far as many of you need to go.

Such drop down menus – or 'look up tables' - would normally be a lot longer than this. Some large databases might contain look up tables containing hundreds of options. This might seem cumbersome, but it is much better than dealing with hundreds of thousands of different spelling variations, typing errors and so on…

Drop down menus also mean that *you can generate meaningful results from your data with only minimal cleaning*. If you *don't* use them to simplify or codify text data, you might spend weeks of precious time cleaning and codifying your laboriously entered text data.

It goes without saying that such drop down menus are particularly vital when more than one person is entering data. It ensures *consistency of data input*, something which is surprisingly difficult to achieve even when only one person is doing it.

When working with databases work on a 'need to know' basis – don't get distracted by the 'lure of the database'. Don't bother changing the colour of forms, for example.

Return to contents page

Why use *relational* databases?

It is perfectly possible to carry out doctoral research using only one table in a database. However, unless the data is very straightforward, requires no significant coding and has a limited number of fields of information, this is unlikely. If your database does have only one table, in fact, you might want to think about how adding more tables could improve the structure, power and meaning of your data.

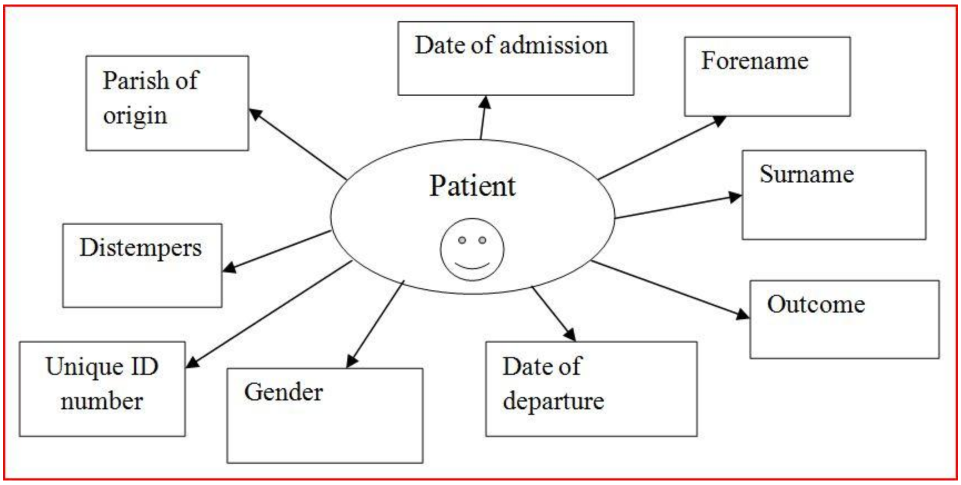
It might also be the case that, although you COLLECT your data in one table, analysis requires you to break your 'data collection table' down into multiple tables.

Database 'theory'

Don't worry, this section is short. Its intention is simply to underline the points made in the first lesson about the nature of all data. Data has structure. When you create a fixed form database (unlike, say, the free form type, where free form text is tagged to identify values) you need to make decisions about how to design the various tables based on that structure.

In general, each table in a database should be based around an *entity*, each with its own *attributes*. Thus, for example, the Westminster Infirmary data is based around the entity patient. Each patient has various attributes (age, date of admission, date of departure, outcome, condition on entry, name, gender, parish of origin). Each record of the table contain *fields* of information (attributes) about the *entity* in question.

This diagram gives a rough idea:



As you can see, however, **a number of the attributes could, themselves, also be entities**. Each parish of origin, for example, could possess its own attributes of acreage, relative wealth, geographical location, population and so on.

The Unique ID number, which each patient *record* is identified by, could also be used in a master table, where other characteristics of that Patient were stored (date of birth, date of death, occupation). To create such a master database of unique individuals, *you would have to create a new numerical field*, and give each occurrence of the patients that unique number (at present the Patients relieved table contains some records that relate to the same individual. At this point you would have to decide which individuals *were the same person*).

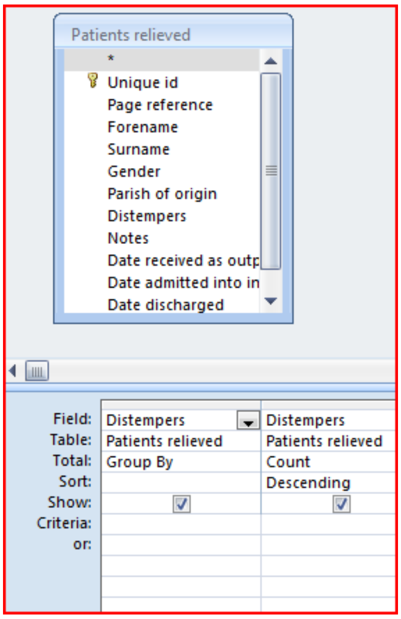
> *The real power of Access lies in its ability to link tables together on common shared fields. Once linked, you can perform analyses on subsets of the linked data*. Note that once you link two tables together on a common shared field, *records that do not have such a common shared field will not be used in the subsequent analysis*.
>
> *Linking tables together on common shared fields is also a good way of coding data*.
>
> *Note that if you link tables together, you must ensure that the common fields are identical to each other (same type, name and size) in both Tables.*
>
> *You must, obviously, also ensure that, where you are linking on unique id numbers, that the numbers really relate to the identical thing.*

You can also use the relational power of Access to codify complex data using queries. The following section provides a step by step guide as to how to do this…

Return to contents page

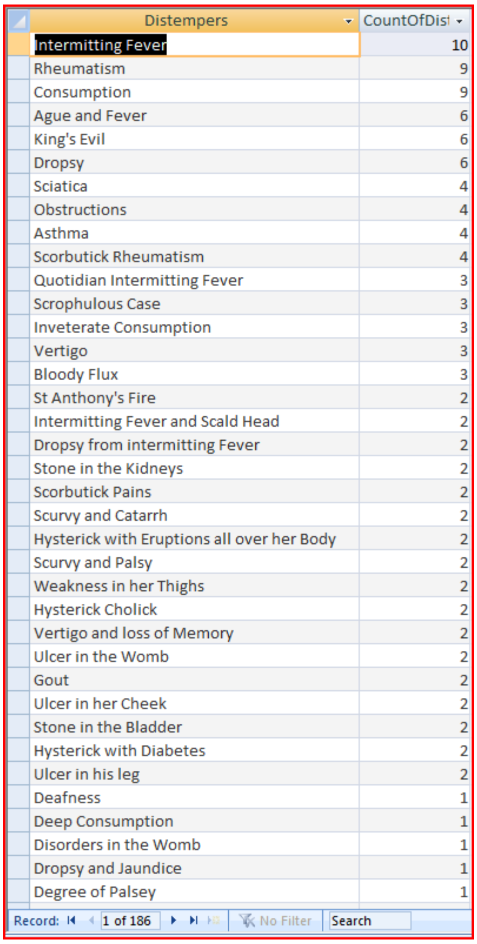
Data coding and data standardization using queries

To carry out this simple task we will use the Westminster Infirmary Database. Start by running a simple query. Count the number of distempers by creating the following simple query (this already exists in the sample database):



Field:	Distempers	Distempers
Table:	Patients relieved	Patients relieved
Total:	Group By	Count
Sort:		Descending
Show:	☑	☑
Criteria:		
or:		

You should get the following table:

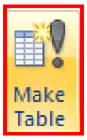


Distempers	CountOfDist
Intermitting Fever	10
Rheumatism	9
Consumption	9
Ague and Fever	6
King's Evil	6
Dropsy	6
Sciatica	4
Obstructions	4
Asthma	4
Scorbutick Rheumatism	4
Quotidian Intermitting Fever	3
Scrophulous Case	3
Inveterate Consumption	3
Vertigo	3
Bloody Flux	3
St Anthony's Fire	2
Intermitting Fever and Scald Head	2
Dropsy from intermitting Fever	2
Stone in the Kidneys	2
Scorbutick Pains	2
Scurvy and Catarrh	2
Hysterick with Eruptions all over her Body	2
Scurvy and Palsy	2
Weakness in her Thighs	2
Hysterick Cholick	2
Vertigo and loss of Memory	2
Ulcer in the Womb	2
Gout	2
Ulcer in her Cheek	2
Stone in the Bladder	2
Hysterick with Diabetes	2
Ulcer in his leg	2
Deafness	1
Deep Consumption	1
Disorders in the Womb	1
Dropsy and Jaundice	1
Degree of Palsey	1

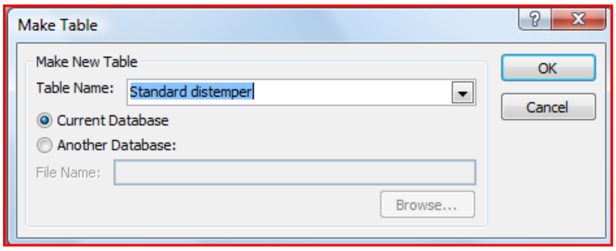
Record: 1 of 186

Even counting the different distempers has still produced 186 different variations from a database of just 265 records. However, we can use Access to recode these text strings.

1. Start by running a Make Table Query. Go back to the *Count Distempers* query design and click on the Make Table Query.



2. This produces a dialogue box. Completing it will enable you to create a new permanent table from your query.



Make Table

Make New Table

Table Name: Standard distemper

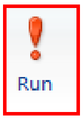
● Current Database

○ Another Database:

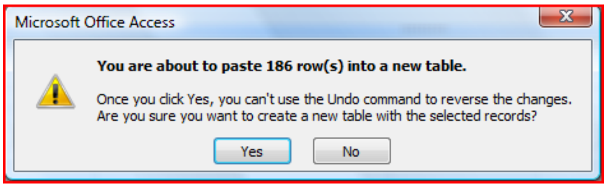
File Name:

3. Give the new Table some sensible name: *Standard distemper*

4. Then click on the Run command (*not* the View command)



This should produce the following warning:



Microsoft Office Access

You are about to paste 186 row(s) into a new table.

Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to create a new table with the selected records?

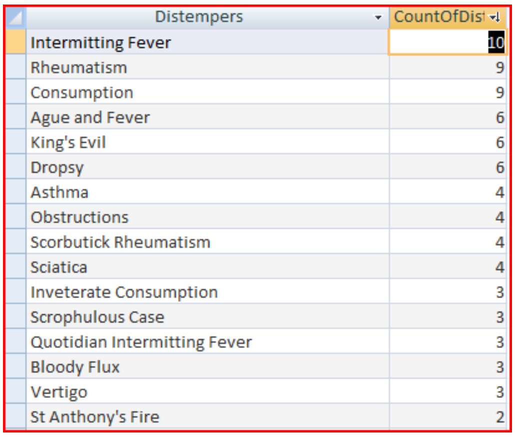
Hint: *If this doesn't work, you might need to enable any blocked content*.

Click on Yes.

Close the query, if you do not save it, you will still have the original Count distemper query.

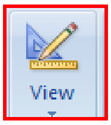
You should now have a Standard Distemper table in your database, with 186 rows.

5. Open it, and it should look like the original query:

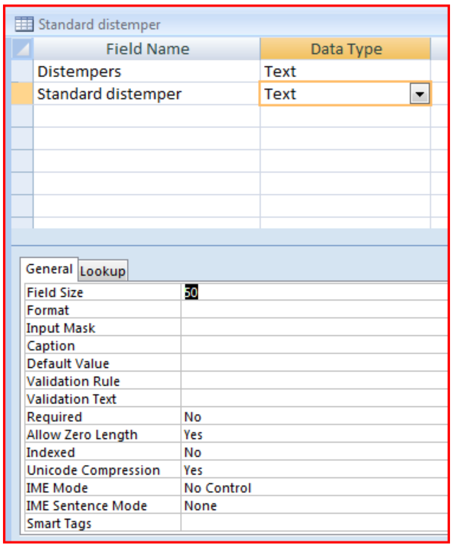


Distempers	CountOfDis
Intermitting Fever	10
Rheumatism	9
Consumption	9
Ague and Fever	6
King's Evil	6
Dropsy	6
Asthma	4
Obstructions	4
Scorbutick Rheumatism	4
Sciatica	4
Inveterate Consumption	3
Scrophulous Case	3
Quotidian Intermitting Fever	3
Bloody Flux	3
Vertigo	3
St Anthony's Fire	2

6. Now go into Design View and delete the *Countofdistemper* field.



And create a **new** *Standard distemper* field, with a field size of 50



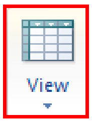
Standard distemper

Field Name	Data Type
Distempers	Text
Standard distemper	Text

General | Lookup

Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

7. Click on the View button, saving the table when prompted;



Now you have created a table which contains all the possible values of the original distemper field, together with a new text field.

The tedious bit is to decide how to categorize each of the various distempers. A historian would turn to a contemporary nosology at this point, to classify each disease.

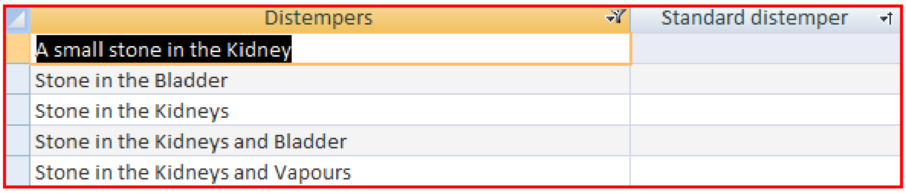
One way of quickening up this process is to use filters, to identify shared words. Obviously this is just an exercise, so what follows is pretty rough and ready.

This coding technique should only be used when all data has been collected, since you don't want to add new data which may not have a matching text code.

What you need to do now is to standardize each distemper to some common headings. The last stage will be to link the Standard distemper table to the Patients relieved table.

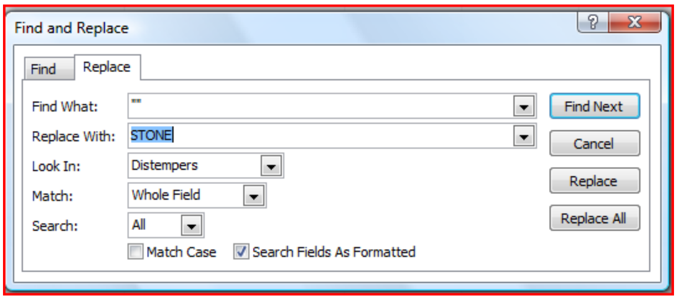
8. Hint: you can *replace* blank records with content. If you apply a filter, this can speed up such coding.

Thus if you filter the table for every record containing 'stone' you get:



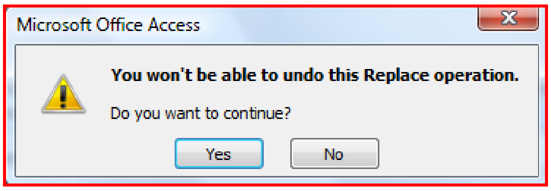
Distempers	Standard distemper
A small stone in the Kidney	
Stone in the Bladder	
Stone in the Kidneys	
Stone in the Kidneys and Bladder	
Stone in the Kidneys and Vapours	

Click in a blank record in Standard distemper field, and click on the Replace button.



Find What: ""
Replace With: STONE
Look In: Distempers
Match: Whole Field
Search: All
□ Match Case ☑ Search Fields As Formatted

Two double quotation marks will find all blank text fields, which you can then replace with text. Click on replace all, and then click on Yes to continue, as follows:



You won't be able to undo this Replace operation.

Do you want to continue?

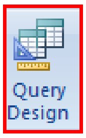
Then clear the filter and move onto the next code.

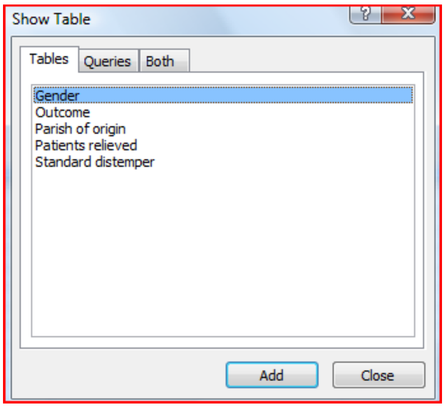
To save time, just code entries which contain the following key words (some might contain more than one term, but just choose one): ULCER, STONE, RHEUMATISM, PALSY, PAIN, HYSTERICK, FEVER, DROPSY, CONSUMPTION, ASTHMA. All the remaining blank field just label as CHRONIC.

Now you have given a text code to *every possible text string in the distemper field*.

9. Lastly we can create a query that links the two tables (*Patients relieved*, and *Standard distemper*)

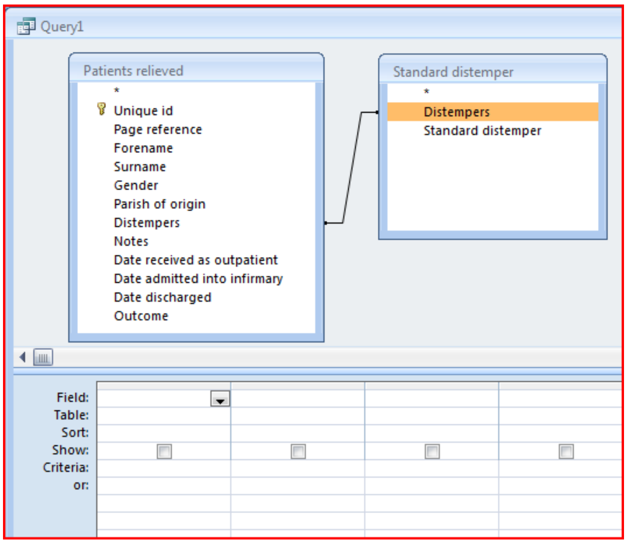
Create, Query Design



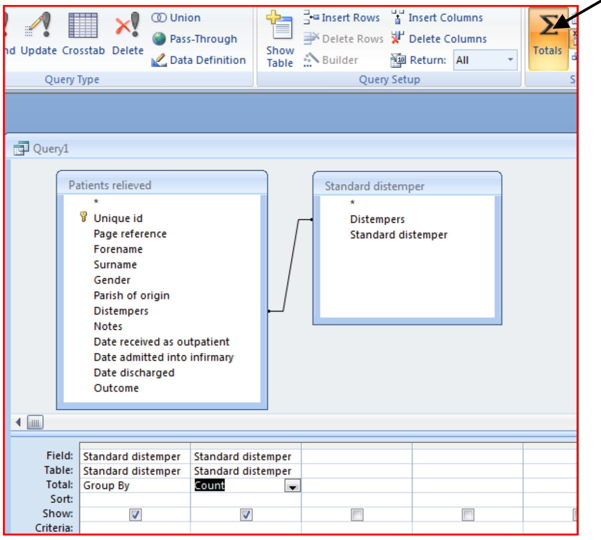


Gender
Outcome
Parish of origin
Patients relieved
Standard distemper

10. Add the *Patients relieved* and *Standard distemper* tables to the query.

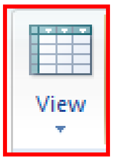


11. Then drag the *Distempers* field from one table to the other, to create the link.
12. Now you have done that, it is possible to count the *Standard distempers*. This will count all those records that link to your *Standard distemper* coding. Click the Totals button, and drag down the *Standard distempers* field <u>twice</u>. Select Group by, and Count.

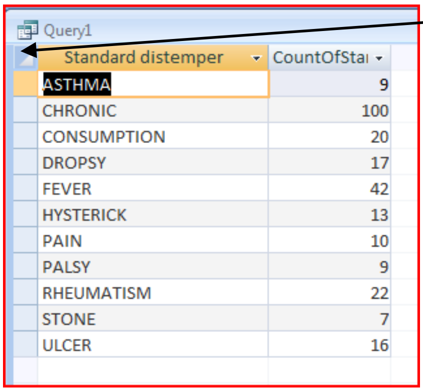


Field:	Standard distemper	Standard distemper
Table:	Standard distemper	Standard distemper
Total:	Group By	Count
Sort:		
Show:	☑	☑
Criteria:		

13. Then run the query with the View button



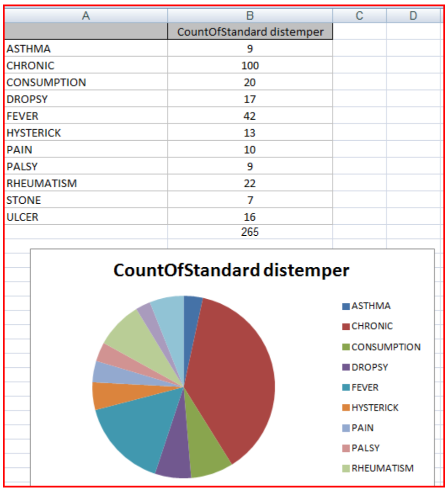
This should produce the following table:



Standard distemper	CountOfSta
ASTHMA	9
CHRONIC	100
CONSUMPTION	20
DROPSY	17
FEVER	42
HYSTERICK	13
PAIN	10
PALSY	9
RHEUMATISM	22
STONE	7
ULCER	16

Click here to select the whole table. Then copy the table into Excel

14. If you copy and paste this table into Excel, you can produce a nice chart of this relatively bogus table (*bogus*, because the grouping contains 100 ‘chronic’ and a number of dual entries)… Note that the table does add up to 265, the number of records in the original *Patients relieved* table.



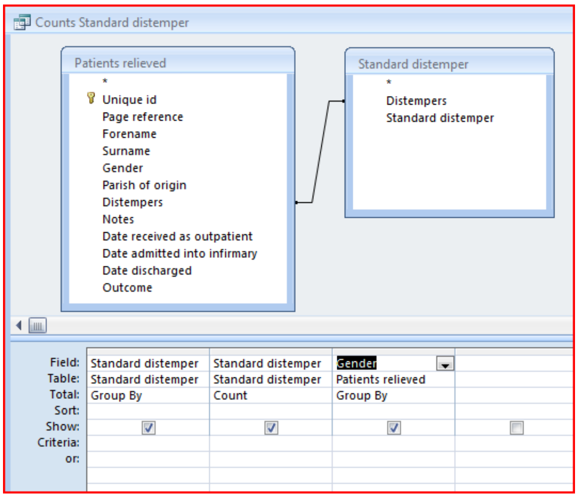
A	B	C	D
	CountOfStandard distemper		
ASTHMA	9		
CHRONIC	100		
CONSUMPTION	20		
DROPSY	17		
FEVER	42		
HYSTERICK	13		
PAIN	10		
PALSY	9		
RHEUMATISM	22		
STONE	7		
ULCER	16		
	265		

This technique, coding by using the *make table* query, and using the relational power of Access, is very useful once your data has been collected. The key part, obviously, lies in the coding. *There are whole articles on coding systems, but the key decisions are, in essence, academic ones that you will have to decide yourself.*

If you decide (as is not uncommon) that the coding system needs revision, the work involved is relatively trivial since the original data is left untouched.

Obviously, once you have coded complex text strings in this way, you can break the data down by other variables in your table.

For example, you could pull down the gender field, and break the above results down by gender.


Counts Standard distemper

Patients relieved

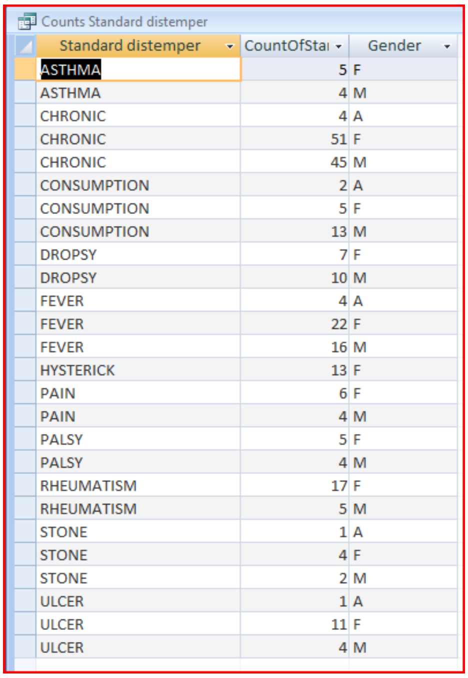
- *
- Unique id
- Page reference
- Forename
- Surname
- Gender
- Parish of origin
- Distempers
- Notes
- Date received as outpatient
- Date admitted into infirmary
- Date discharged
- Outcome

Standard distemper

- *
- Distempers
- Standard distemper

Field:	Standard distemper	Standard distemper	Gender	
Table:	Standard distemper	Standard distemper	Patients relieved	
Total:	Group By	Count	Group By	
Sort:				
Show:	☑	☑	☑	☐
Criteria:				
or:				

This produces the following result:


Counts Standard distemper

Standard distemper	CountOfStai	Gender
ASTHMA	5	F
ASTHMA	4	M
CHRONIC	4	A
CHRONIC	51	F
CHRONIC	45	M
CONSUMPTION	2	A
CONSUMPTION	5	F
CONSUMPTION	13	M
DROPSY	7	F
DROPSY	10	M
FEVER	4	A
FEVER	22	F
FEVER	16	M
HYSTERICK	13	F
PAIN	6	F
PAIN	4	M
PALSY	5	F
PALSY	4	M
RHEUMATISM	17	F
RHEUMATISM	5	M
STONE	1	A
STONE	4	F
STONE	2	M
ULCER	1	A
ULCER	11	F
ULCER	4	M

The numbers are relatively small, but there are clearly some significant features revealed by this modest gender analysis. All 13 ‘Hysterick’ conditions were suffered by females - hysteria being associated with the womb. Rheumatism was commoner amongst females too, as were ulcers. Consumption appears to have been a predominantly male complaint. Further analysis, more variables and greater knowledge about the population at risk would be needed to fully contextualize such findings, but the exercise does reveal the value and power of using a relational database to analyze this data.

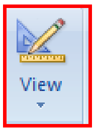
Return to contents page

Building complex databases

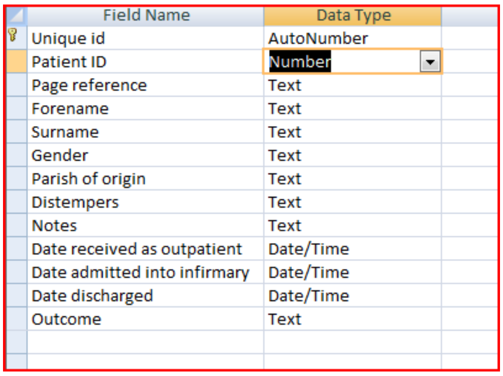
Historians are likely to be most interested in using this sort of information to build up biographies of **people** who used the Infirmary. They would also be interested in using information in *other databases* to increase the amount of information on each individual. Of course others might be interested in looking at the parishes of those who used the Infirmary. Archaeologists might want to structure a database to focus on particular artefacts (each with their own *attributes*) …

Such an analysis can only be done if the Infirmary database is restructured. This will involve creating a 'master table' of all *individuals*. It helps to break this down into stages.

1. Although there are 265 records of admission, some individuals clearly appear more than once. We **need therefore to create a new Patient ID number** to identify each individual in the *Patients relieved* table.
   Open the *Patients relieved* table and click on the View button.

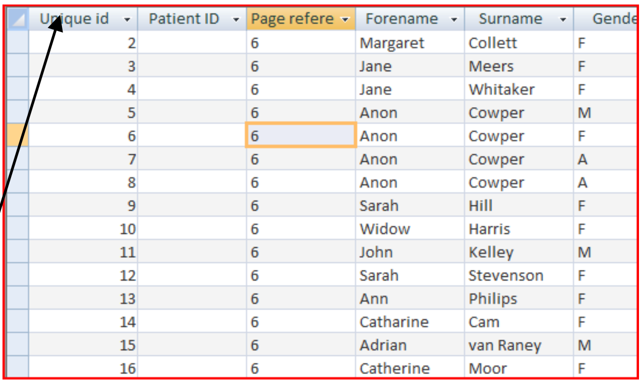


2. Under the Unique ID number insert a new row (*right click, insert row*) and call it Patient ID:



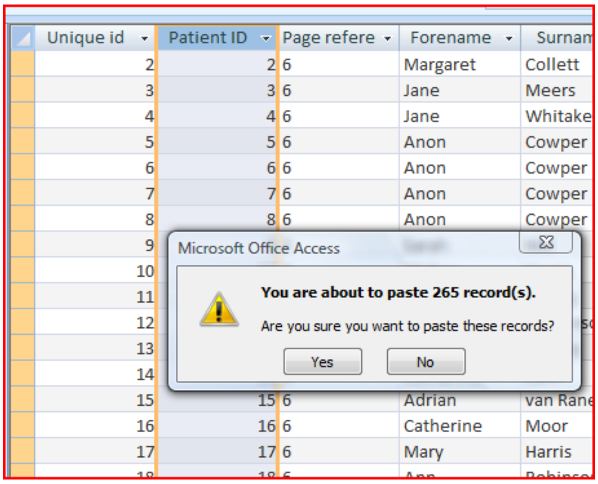
Field Name	Data Type
Unique id	AutoNumber
Patient ID	Number
Page reference	Text
Forename	Text
Surname	Text
Gender	Text
Parish of origin	Text
Distempers	Text
Notes	Text
Date received as outpatient	Date/Time
Date admitted into infirmary	Date/Time
Date discharged	Date/Time
Outcome	Text

3. Save the Table when prompted. You should now have a new field in your *Patients relieved* table.



Unique id	Patient ID	Page refere	Forename	Surname	Gende
2		6	Margaret	Collett	F
3		6	Jane	Meers	F
4		6	Jane	Whitaker	F
5		6	Anon	Cowper	M
6		6	Anon	Cowper	F
7		6	Anon	Cowper	A
8		6	Anon	Cowper	A
9		6	Sarah	Hill	F
10		6	Widow	Harris	F
11		6	John	Kelley	M
12		6	Sarah	Stevenson	F
13		6	Ann	Philips	F
14		6	Catharine	Cam	F
15		6	Adrian	van Raney	M
16		6	Catherine	Moor	F

4. You now need to enter a new *Patient ID* number into that field for each record. Before you laboriously start that, Think. *Hang on*, why not paste the *Unique id* field contents into the *Patient ID* contents? Then you would *only* have to change numbers when an individual appears more than once!
   To copy field contents is easy in Access.
   Just select the *Unique id* field by clicking on the row heading.
   Then click on the row heading of the *Patient ID* field, and select Paste.
   You should get the following:



Unique id	Patient ID	Page refere	Forename	Surnam
2	2	6	Margaret	Collett
3	3	6	Jane	Meers
4	4	6	Jane	Whitake
5	5	6	Anon	Cowper
6	6	6	Anon	Cowper
7	7	6	Anon	Cowper
8	8	6	Anon	Cowper
9				
10				
11				
12				
13				
14				
15	15	6	Adrian	van Rane
16	16	6	Catherine	Moor
17	17	6	Mary	Harris

Microsoft Office Access

You are about to paste 265 record(s).

Are you sure you want to paste these records?

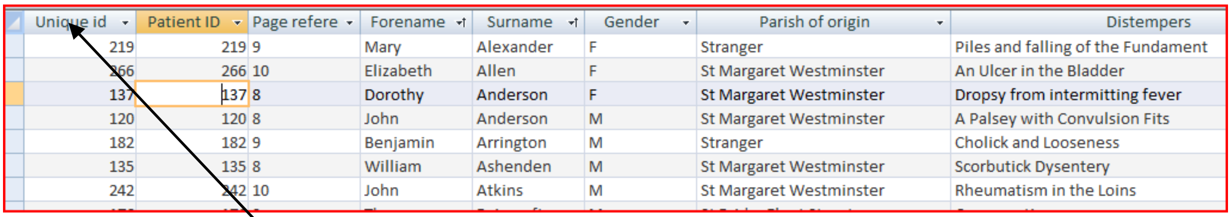
Choose Yes, and you have a full *Patient ID* field. It takes longer to type this than it does to do it.

5. Now sort the database by forename *then* surname. This will make it easy identify the same individuals. *Sorting by forename first, will mean the surnames are sorted by surname, but then by forename*.

Sort by clicking on the *Forename* field and then on the Sort Ascending button:

Repeat for the *Surname* field.

You should now have a table looking like this:



Unique id	Patient ID	Page refere	Forename	Surname	Gender	Parish of origin	Distempers
219	219	9	Mary	Alexander	F	Stranger	Piles and falling of the Fundament
266	266	10	Elizabeth	Allen	F	St Margaret Westminster	An Ulcer in the Bladder
137	137	8	Dorothy	Anderson	F	St Margaret Westminster	Dropsy from intermitting fever
120	120	8	John	Anderson	M	St Margaret Westminster	A Palsey with Convulsion Fits
182	182	9	Benjamin	Arrington	M	Stranger	Cholick and Looseness
135	135	8	William	Ashenden	M	St Margaret Westminster	Scorbutick Dysentery
242	242	10	John	Atkins	M	St Margaret Westminster	Rheumatism in the Loins

6. You are now going to go through all 265 records and decide which records relate to the same individuals. It is helpful when doing this to hide the adjacent numerical fields.

   It is surprising how easy it is to confuse fields when carrying out mind-numbing tasks.

   To hide a column *right click* on the top of the column you want to hide, and select *Hide Columns*. Hide both the Unique ID and Page reference columns.

   Identifying individuals will not take very long, because, luckily, there are not that many individuals who appear more than once. It is worth noting at this point that until the nineteenth century most spelling was phonetic, or at best variable, so the same person can appear differently spelt. In the end, although there are lots of techniques to help out the process of nominal linkage, you have to decide which individuals are the same person. It is usual to err on the side of caution.

   The following individuals appear more than once:

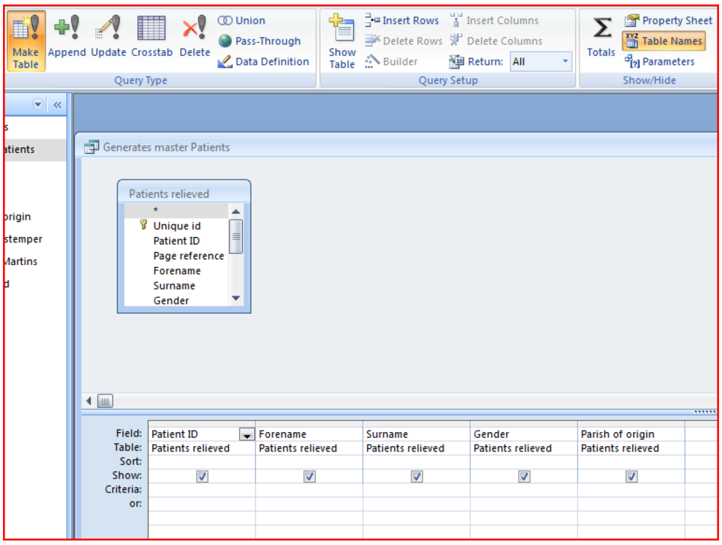
Females (it is worth linking by sex, using a filter): Eleanor Davis (both occurrences assigned the unique Patient ID 23); Elizabeth Hatfield (79); Jane Horne (161); Esther Johnson (111); Margaret/Mary Marlow (112); Ann Robinson (18); Sarah Stallen (22); Rachel Watkins (105). Males: William Coney (88); William Forrester (149); Edward Hawkins (92); John Kelly (11); Nicholas Smith (116); Evan Williams (193);

Of course, without age information, we cannot be 100% sure that all these individuals are the same person. *Thus one Anne Squire appears twice, but is two different people, as is Anne Kellington. Both were admitted with children. Note that in one case an entire family of Cowpers was admitted* ***en bloc*** *(IDs 5-8).* Welcome to the uncertainty that is research.

Get used to it.

7. Now we have to create a new table, consisting only of *unique individuals*.

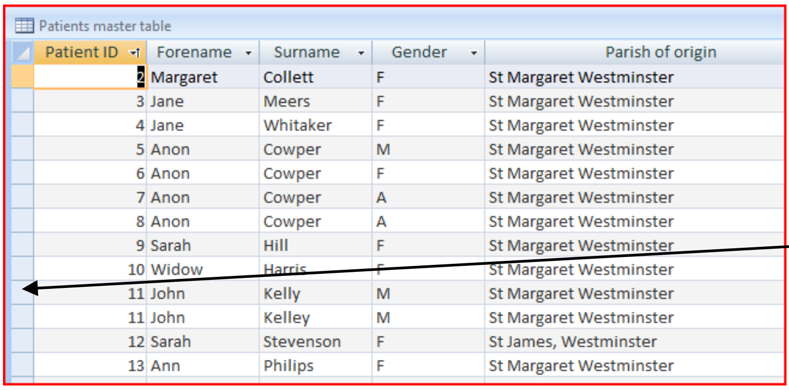
   Construct a *Make Table* query, to select out the unique Pauper ID's together with *forename*, *surname*, *gender* and *Parish of origin* (the key entities).



Field:	Patient ID	Forename	Surname	Gender	Parish of origin
Table:	Patients relieved	Patients relieved	Patients relieved	Patients relieved	Patients relieved
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

   Call the new table something sensible Pauper master table

8. Then go into that table, and DELETE each duplicated Pauper ID record.


Patients master table

Patient ID	Forename	Surname	Gender	Parish of origin
2	Margaret	Collett	F	St Margaret Westminster
3	Jane	Meers	F	St Margaret Westminster
4	Jane	Whitaker	F	St Margaret Westminster
5	Anon	Cowper	M	St Margaret Westminster
6	Anon	Cowper	F	St Margaret Westminster
7	Anon	Cowper	A	St Margaret Westminster
8	Anon	Cowper	A	St Margaret Westminster
9	Sarah	Hill	F	St Margaret Westminster
10	Widow	Harris	F	St Margaret Westminster
11	John	Kelly	M	St Margaret Westminster
11	John	Kelley	M	St Margaret Westminster
12	Sarah	Stevenson	F	St James, Westminster
13	Ann	Philips	F	St Margaret Westminster

To delete a record, click on the left hand row and right click, select *Delete record*

You should need to delete 14 records. There are therefore (hopefully) 251 individuals who appear 265 times in the Infirmary records.

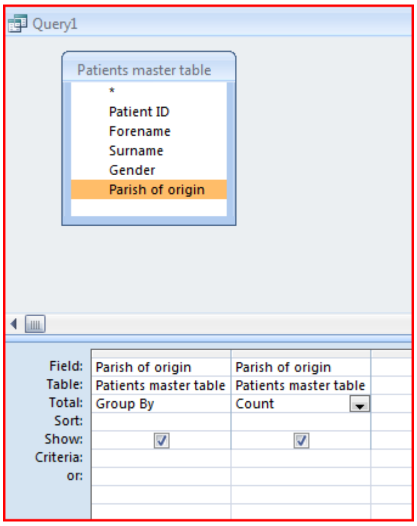
Now you have done that, you have also effectively *standardized* the *Patients relieved* table. That is, you have a table in which each individual has got one forename, one surname, one gender and one address. If you had a table with 85,000 records, you can imagine how many variations in spelling would be standardized by this approach.

If you have done this correctly, and deleted each duplicated Patient ID, you should be able to make the Patient ID field the primary key. You cannot have duplicated primary key fields by the default setting in the relevant indexed field *(Yes (No Duplicates))*

You have also got a clear idea of the individuals who appeared at the Infirmary. You could count the unique addresses, for example, which would give the number of parishes of origin controlled for individuals who appear more than once.

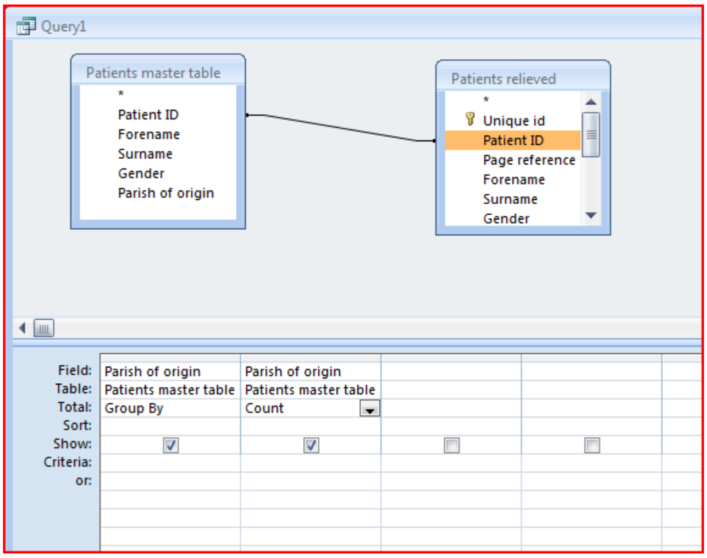
9. To see the difference, run two queries.

    i. One counting the Parish of origin in the *Patient master table*:



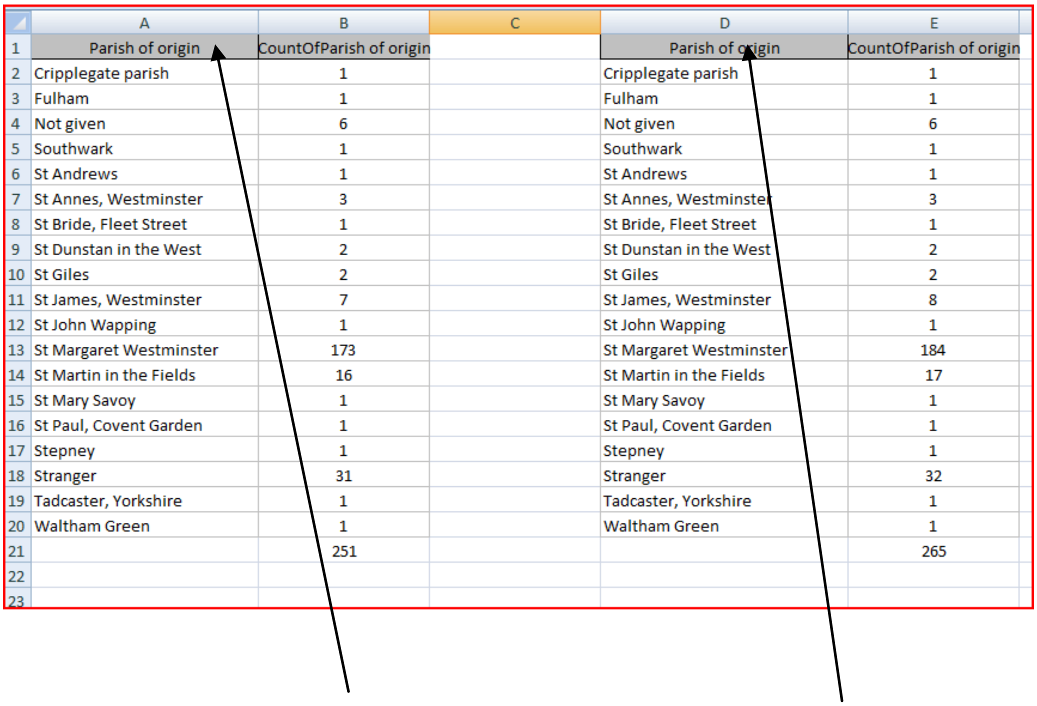
Field:	Parish of origin	Parish of origin
Table:	Patients master table	Patients master table
Total:	Group By	Count
Sort:		
Show:	☑	☑
Criteria:		
or:		

    ii. One counting the Parish of origin in the *Patient relieved* linked to the *Patient master table on the common shared Patient ID field*



Field:	Parish of origin	Parish of origin		
Table:	Patients master table	Patients master table		
Total:	Group By	Count		
Sort:				
Show:	☑	☑	□	□
Criteria:				
or:				

Paste both results into the same worksheet into Excel:



	A	B	C	D	E
1	Parish of origin	CountOfParish of origin		Parish of origin	CountOfParish of origin
2	Cripplegate parish	1		Cripplegate parish	1
3	Fulham	1		Fulham	1
4	Not given	6		Not given	6
5	Southwark	1		Southwark	1
6	St Andrews	1		St Andrews	1
7	St Annes, Westminster	3		St Annes, Westminster	3
8	St Bride, Fleet Street	1		St Bride, Fleet Street	1
9	St Dunstan in the West	2		St Dunstan in the West	2
10	St Giles	2		St Giles	2
11	St James, Westminster	7		St James, Westminster	8
12	St John Wapping	1		St John Wapping	1
13	St Margaret Westminster	173		St Margaret Westminster	184
14	St Martin in the Fields	16		St Martin in the Fields	17
15	St Mary Savoy	1		St Mary Savoy	1
16	St Paul, Covent Garden	1		St Paul, Covent Garden	1
17	Stepney	1		Stepney	1
18	Stranger	31		Stranger	32
19	Tadcaster, Yorkshire	1		Tadcaster, Yorkshire	1
20	Waltham Green	1		Waltham Green	1
21		251			265
22					
23					

Results from unlinked Patient master table Results from linked tables

This shows that most of the individuals who appeared more than once were from St Margaret Westminster.

Note that linking tables could sometimes reduce the number of records you are working when only a subset of the records in one database appear in the other.

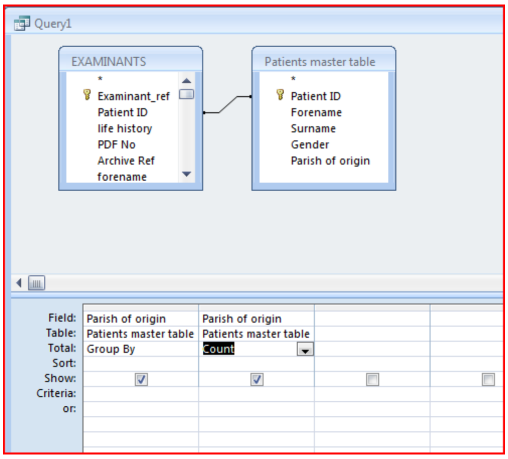
If you had other databases, and you believed some of these individuals also appear in the Infirmary database, by giving them the same Patient Id number, you could link the databases together. You would link to the *Patient master table*, your table of unique patients.

Return to contents page

As it happens, there IS a small *Examinants* table which contains three individuals whose settlement examinations have been recorded. These might be the same individuals as appear in the Westminster Infirmary Database. I have added a Patient ID field to this *Examinants* table and matched the relevant numbers from the *Patients Master* Table

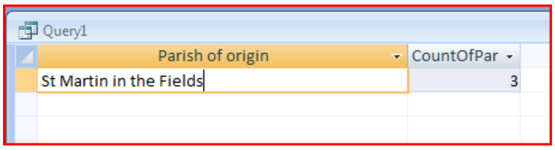
10. *For this exercise you need to download (from http://www.staff.ncl.ac.uk/j.p.boulton/esrcdtcdatabasetraining.htm and copy the Examinants table from the small Examinants Database into the Westminster Infirmary Database*. Copy the table and simply paste it into the Patients relieved database.

11. Now, if you link the two tables together and repeat the same *Parish of origin* query, you will get the answer 3. This is because only three records share the same Patient ID.



Field:	Parish of origin	Parish of origin		
Table:	Patients master table	Patients master table		
Total:	Group By	Count		
Sort:				
Show:	☑	☑	□	□
Criteria:				
or:				

This gives this result:



Parish of origin	CountOfPar
St Martin in the Fields	3

See the Appendix for a discussion as to whether these individuals are actually the same person. It is more difficult to link across different classes of record without supporting information of good quality. It seems likely that at least one of the links (Elizabeth Hyfield) is a valid one.

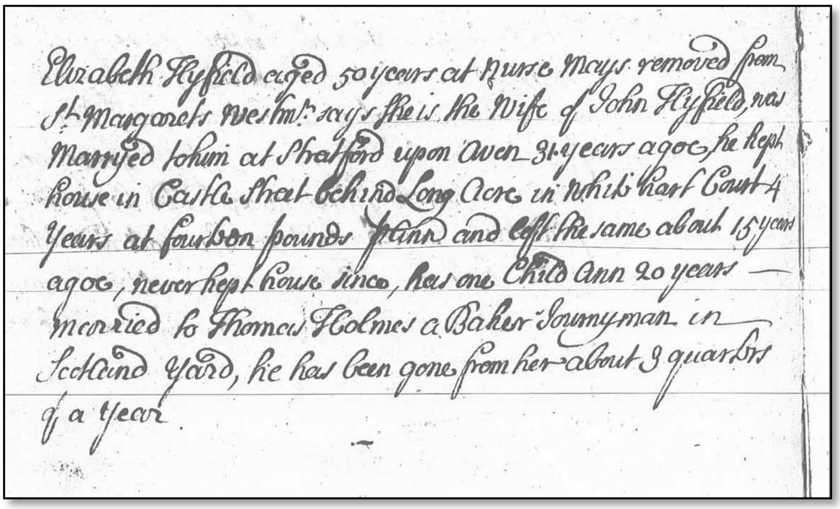
This linkage exercise is designed to show you that:

1) With a Master table you can build up lots of information from different sources about the person/thing you are interested in.
2) Linkage of tables by queries can sometimes *generate small subsets of results*. This must be borne in mind when carrying out more complex queries.

Appendix: the possible linked Settlement records

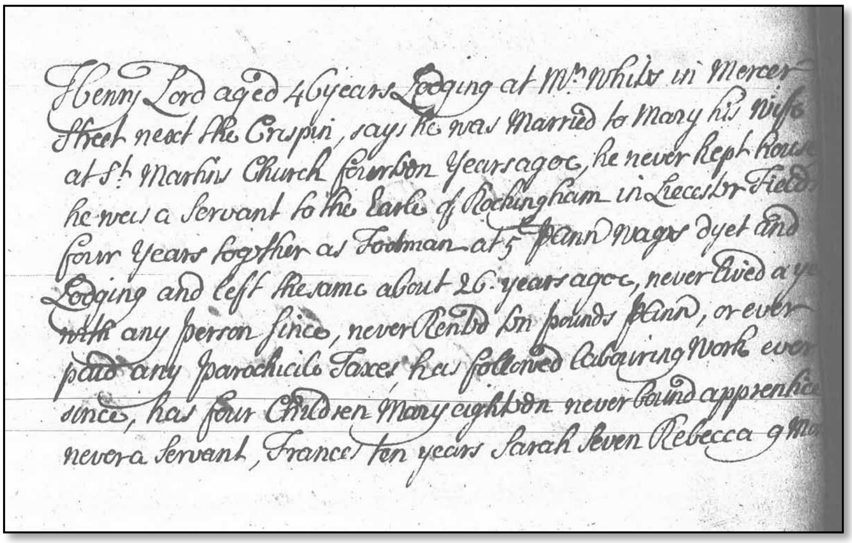
For those interested, here are the settlement records in question:

Elizabeth Hyfield was at Nurse Mays removed from St Margarets Westminster, 25 November 1720. It seems very likely that this is the same person as is listed in the Infirmary.



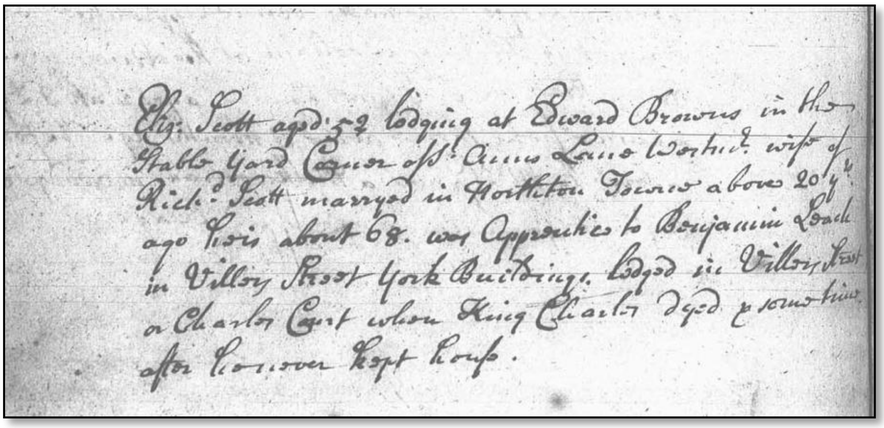
Elizabeth Hyfield aged 50 years at Nurse Mays removed from
St Margarets Westmr says she is the Wife of John Hyfield, was
Marryed to him at Stratford upon Aven 31 years agoe, he kept
house in Castle Street behind Long Acre in White hart Court 4
Years at fourteen pounds p ann and left the same about 15 years
agoe, never kept house since, has one Child Ann 20 years —
married to Thomas Holmes a Baker Journyman in
Scotland Yard, he has been gone from her about 3 quarters
of a year.

Henry Lord’s examination (c. 25 November 1720) provides no link to the Infirmary or St Margaret Westminster:



Henry Lord aged 46 years Lodging at Mr Whites in Mercer
Street next the Crispin, says he was Married to Mary his Wife
at St Martins Church fourteen Years agoe, he never kept house
he was a Servant to the Earle of Rockingham in Leicester Fields
four Years together as Footman at 5 p ann wages dyet and
Lodging and left the same about 26 years agoe, never lived a ye
with any person since, never Rented ten pounds p ann, or ever
paid any parochiall Taxes has followed Labouring Work ever
since, has four Children Mary eighteen never bound apprentice
never a Servant, Frances ten years Sarah seven Rebecca 9 mo

Elizabeth Scott’s examination (08 June 1719) does suggest a prior link to Westminster, but has no reference to any period of sickness.



Eliz: Scott aged 52 lodging at Edward Browns in the
Stable Yard Corner of St. Anns Lane Westmr. wife of
Richd. Scott marryed in Northton Towne above 20 yrs.
ago hee is about 68. was Apprentice to Benjamin Leach
in Villers Street York Buildings. lodged in Villers Street
or Charles Court when King Charles dyed & some time
after he never kept house.

Return to contents page