

Creating a Relational Database in Access

Farrokh Alemi, Ph.D.

The objective of this note is to help you understand how a relational database is organized as a collection of tables, linked to each other. In this note, we show how access software can be used to create tables of data. Then we show how these tables can be linked. A separate note, posted to the class web site, demonstrates the same concepts but uses Microsoft SQL code to do so.

We demonstrate the concepts behind relational databases through showing a simple electronic health record with three tables: a table on patients, another table on providers and a third table on medical encounters/visits that link the patient and the provider.

Entities & Attributes

In database design, everything that has its own table is referred to as an entity. So in our example, patients, providers, and encounters are considered entity. For each of these three entities we need to create separate tables. Figure 1 shows the three entities and their relationship to each other.

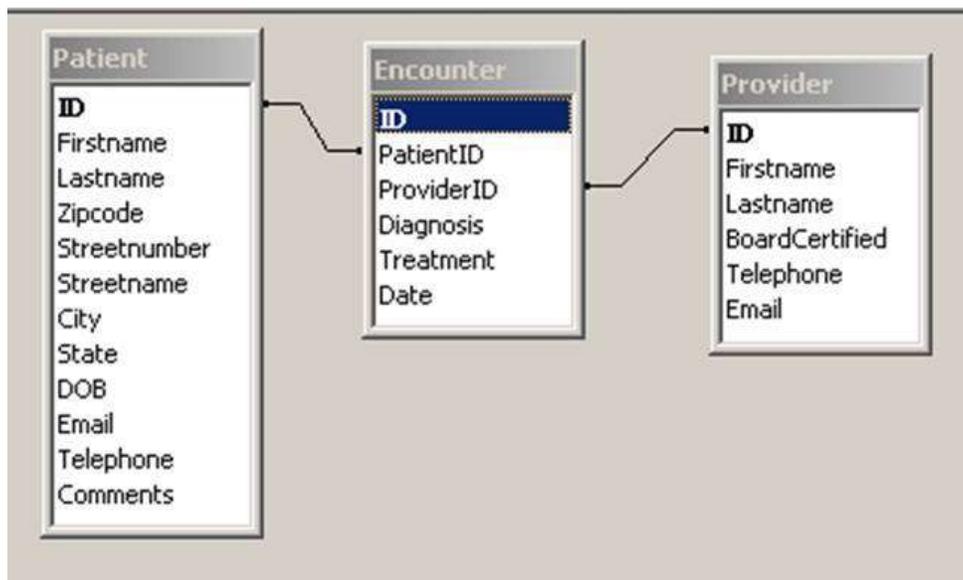


Figure 1: Relationship among Three Entities/Tables

Each table contains a series of fields. Fields are data we collect about various entities. These fields provide the attributes of the entity. The patient attributes are shown in Figure 1 and are assumed to be first name, last name, date of birth, address (street name, street number, city, State and zip code) and email. First name is a string of maximum size 20. Last name is a string of maximum size 50. Street number and zip code are integer numbers with no decimals. Date of birth is a date put in the format DD MMM YY (e.g. 19 Jan 04). The possible values for the State are Maryland, Virginia,

District of Columbia and other. Patient's telephone number could be text or number. A patient ID (auto-number) should be used as the primary key for the table. The provider attributes are assumed to be first name (text of size 20), last name (text of size 50), whether they are board certified (a yes/no value), date of hire (in format DD MMM YY), telephone (text or number) and email (text of size 75). Employer's ID number should be the primary key for the table. The encounter entity is assumed to have the following attributes: patient ID, provider ID, diagnosis (text of size 50), treatment (text of size 50) and date of encounter (date entered in the format DD MMM YY). Each encounter should have its own ID number.

Creating Patient Table

Patient table can be created in SQL or in Microsoft Access, which is a graphical approach to SQL. To make the table in Access, we start by opening a new databases and naming it EMR. Next we create the three tables. We begin with the table for the patients. We create the patients table in the design view. Note that we do not put any spaces in field names. Our first field name is Firstname. We select for data type "Text." In the description we type "Legal first name of the patient." In the field's property menu we set the size of this field to 20. We set for caption "First name". Our second field name is Lastname. We select for data type "Text." In the description we type "Legal last name of the patient." In the field's property menu we set the size of this field to 50. We set for caption "Last name". Our next field is zipcode. The data type is number. The description is the "Minimum 5 digits numerical postal zip code." In the field properties we set the size to 10. Our next field is City, set as text type. In the field properties we set the size to 20. Our next field is State. This field has fixed menu items. To set this up we use a look up wizard. We can look up the values from a different table or type them in. We will type in the possible values.

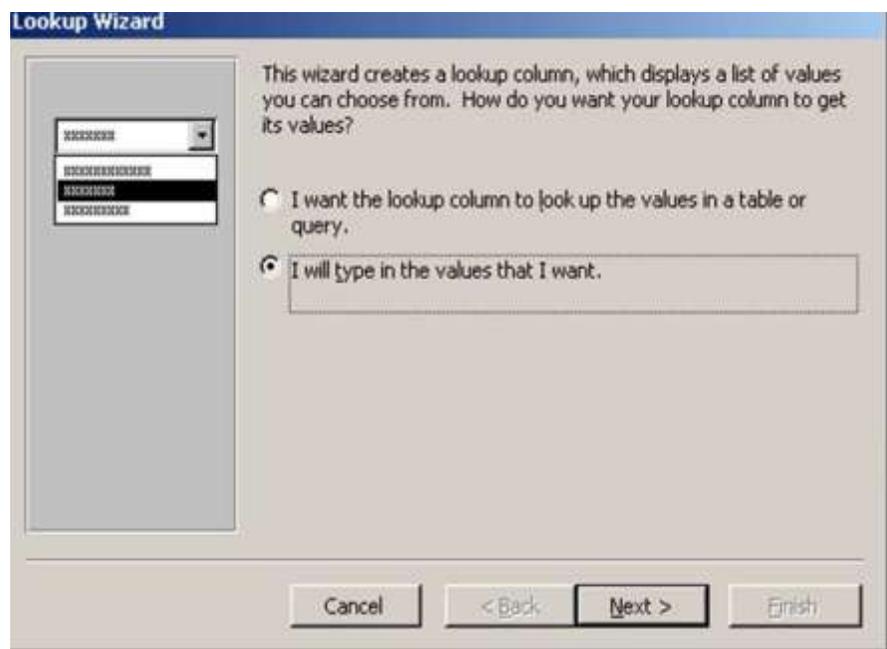


Figure 2: Start of Look Up Wizard

The possible values are Maryland, Virginia, DC, and Other.

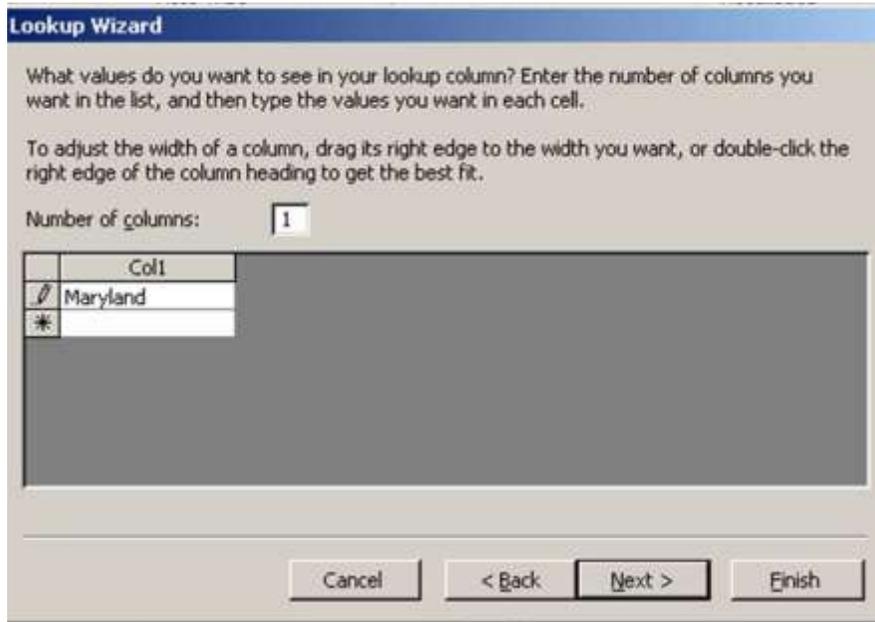


Figure 3: Entering Fixed Menu Items in Lookup Wizard

Our next field is DOB. The data type is set as Date. The description is typed in as "Date of birth of patient." IN the field property we set the format to Medium Date length. We set the input mask, i.e. the additional guides for the database users to input the date, to Medium Date. We need to save the database before the input mask is accepted.

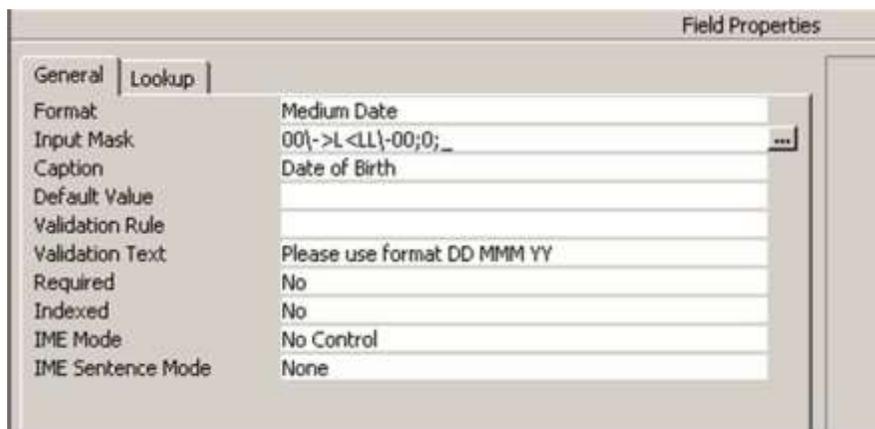


Figure 4: Field Properties for Date

Email field is set to text type. Description includes "Current email of patient." Our next field is Telephone. This is a number type. Because telephones are large numbers, we set the size in the field property to double digit precision. Our last field is a comment field set to text of size 200 and description of "Free form comment about the patient characteristics." We now add a field called ID as an auto-number. This field is selected to be the primary key. A primary key as you know is a unique number that

identifies all contents of a table. We could have used the combination of first and last name as the primary key, but people's first and last name may not be unique. Auto-number solves the problem because now every entry will automatically receive a unique number.

This completes the creation of Patients table. We save the table and name it Patients. Before we proceed, we look at the data entry mode of the table and enter 3 rows of data for our table.

First name	Last name	Zip code	City	State	Date of birth	Email	Telephone of the patient
Farrokh	Alemi	22101	McLean	Virginia	08-Jan-54	Test2@gmu.edu	703 9934226
George	Smith	22102	McLean	Virginia	09-Sep-60	t@tes.com	703 8884545
Jill	Smith	22103	McLean	District of Columbia	01-Aug-89	test@test.com	703 9934226

Table 1: Three Rows of Data for Patient Table

Provider Table

The creation of the Provider table follows a similar procedure. Everything in the table is about providers. Thus a patient's name does not show here. Only facts about the provider show here. We start with creating Firstname as a text of size 20, with the description of "Legal first name of provider." Our second field name is Lastname. We select for data type "Text." In the description we type "Legal last name of the provider." In the field's property menu we set the size of this field to 50. We set for caption "Last name". Note as before that captions can have space but field names cannot. Our next field is BoardCertified. This field is entered as a Yes/No type. The description is "Whether the provider has passed State boards." Our next field is Telephone. This is a number type. Because telephones are large numbers, we set the size in the field property to double digit precision. The description includes "Telephone number of provider." Email field is set to text type. Description includes "Current email of Provider." We now add a field called ID. It contains the employee's unique ID number. This field is selected to be the primary key. This completes the creation of Provider table. We save the table and name it Providers. Before we proceed, we look at the data entry mode of the table and enter 3 rows of data for our provider table.

Providers					
Firstname	Lastname	Boardcertified	Email	Telephone	EmployeeID
Jim	Leggit	Yes	jl@w.com	345 6714545	452310
Jill	Smith	No	js@w.com	345 4561234	454545
George	John	Yes	g@w.com	310 4561234	456734

Table 2: Three Rows of Data for Providers Table

Encounter Table

The last table we would like to create is called Encounters. This table keeps the data from each patient visit. We enter as our first field an ID, as an auto-number. This is described as "Unique ID for the encounter." Next field is the PatientID, entered as a number of size 20. This is described as "Unique patient ID number." Next field is ProviderID, entered as number of size 20 and described as "Unique provider employee number." Next field is Date, entered as a date and described as "Date of encounter." The field property is set to Medium Date and the input mask is set to Medium Date. Next field is Diagnosis, set to text type and described as "Main reason for the visit." The size of this field is set to 75 in the field's property window. Next field is Treatment, set to text and described as "Main treatment provided in the visit." The size of this field is set to 200. We save the design of the encounters table and enter following data.

Encounters						
ID	PatientID	ProviderID	Date of encounter	Diagnosis	Treatment	
1	1	452310	10-Jan-04	Hypertension	Assessment	
2	1	452310	17-Jan-04	Followup	Monitoring	
3	2	452310	10-Jan-04	Unknown	Assessment	
4	3	452310	10-Jan-04	Hypertension	Assessment	
5	1	454545	10-Jan-04	Asthma	Education	

Table 3: Five Records for Encounters Table

Table Relationships

We are now ready to set the relationship among the tables. The purpose of relationships is to let the database know how the tables are related to each other so that we can trace information pertaining to one record in a table to another table. To set the relationship among tables, we select from the database menu, tools and then relationships.

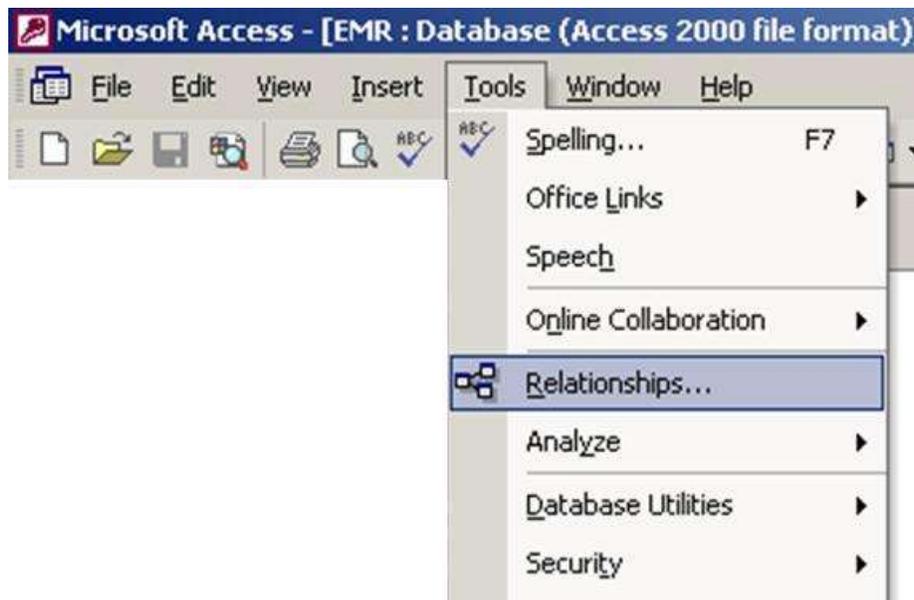


Figure 5: Setting Relationships

In the relationships window, if the window for adding tables is not showing, we right click to show tables.

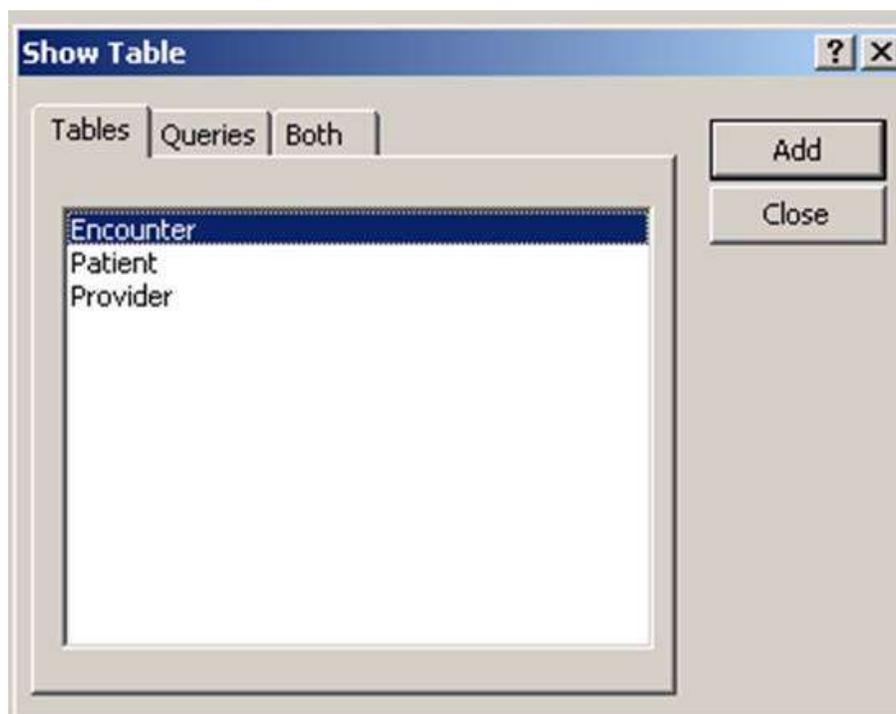


Figure 6: Adding tables to relationships

We add the three tables Patients, Providers and Encounters. We then drag the patient ID in the encounter table to the ID field in the Patient table. We also drag the field ProviderID to the field ID in the provider table. You should see something that looks like the very first figure on this page, Figure 1. In a normal join of two tables, the values of the field must be present in both tables. It is possible to join two tables and require that only one field should always be present and if the value of the field has no matching value in the other table, then it will have a null value. If you carefully click on the join line so that it becomes bold, you can right click and see the option to edit the join properties. In here you can select join type and set the different types of joins.

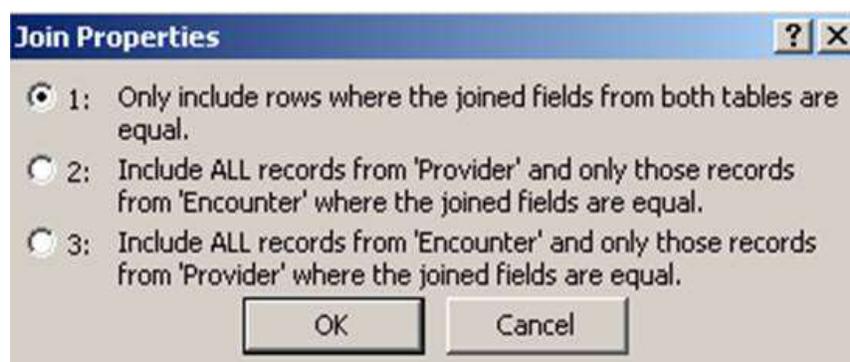


Figure 7: Three types of Joins

For the time being we select to join the encounter and patient table so that the value of patient ID is exactly the same in the two tables. One consequences of this join is if there is an encounter with a patient whose ID has not been entered, it will not show in the joined tables. We will review more of this later when we talk about queries.