

Script for Elluminate Lecture on DB/TextWorks

Creating a Database Using DB/TextWorks. At this point, we've talked about databases from a general or theoretical point of view, and you have some idea what is meant by terms such as *database management system, record, field* and *index*. The purpose of this session is to help clarify your understanding by looking at databases from a practical standpoint. What do you actually do when you create a database? This will also serve as a tutorial on DB/TextWorks, which is the system you will use for the two major assignments in this class.

DB/TextWorks is a database management system (or DBMS). It is made by a company called Inmagic, and some people prefer to call it by that name. It is actually used in some libraries, museums, and archives – mostly smaller ones – for their information retrieval systems. The version that we use here is fully functional, except for the fact that you will not be able to add more than 50 records – so it's really only useful for learning purposes. By now, you should have installed it on your own computer; if not, here is the link to the instructions for doing so:

<http://slisweb.sjsu.edu/ecommunication/IMDown.htm>

The creation of a database involves two major functions, that is, there are two separate roles to be played in making a database. The first is that of the database designer, who sets up the data structure; the second role is that of the database maintainer, who then populates the database with records for the various entities in the collection. The person or persons in this second role are often called indexers or, in a traditional library setting, catalogers.

The job of the database designer actually begins long before approaching the DBMS. The designer must answer questions such as: What is the purpose of the database? Who will be its users? What attributes of documents will they be interested in, and how are they likely to search for those attributes? I call these *primary* decisions both to reflect the fact that they should come first and to indicate their central importance in the ultimate success of the database. Once those questions are answered, then it's time to create the data structure. This generally means defining a variety of fields, first on paper, then using the DBMS, each field to represent an attribute of the document. For each field, the designer must decide a name for the field – generally something brief but descriptive of what the field is about. The designer must also specify the data type contained by the field, for example, whether it contains text or numeric data, and how the field is to be indexed. The designer may also limit what values are possible for a field, or whether a field entry is required or must be unique, or if multiple entries are allowed. In DB/TextWorks, this is called “validation.”

Not everything about the database will be built into the data structure. The database designer will also need to develop standards and write guidelines or rules as a separate document, which the indexer will later follow in creating records for the database.

Once the data structure is created, the job of the indexer or database maintainer begins. For each document in the collection, a record is created. In that record, appropriate values are assigned for each field, according to the rules for the field. As long as the database is active and growing, the job of the database maintainer will go on, and may involve not only adding new records but also updating or sometimes deleting existing records.

For the purposes of this demonstration, we are going to use this very exciting collection of rocks. Although this is probably not what you will be working with in your professional careers, these rocks do have attributes such as size, color, and texture that will serve our purposes here. I'm going to switch now to DB/TextWorks.

When you first start up the program, you're faced with this largely blank, gray, rather unhelpful screen. The first thing you want to do is click File > New Textbase. ("Textbase," by the way, is simply what Inmagic calls a database; it's not a generally used term.) Choose the folder where you want to store the database. The database will actually comprise 11 different files, so I recommend putting it in its own special folder. In the File Name box, type in a name for the database – preferably something descriptive. I will call this one *RJ-rocks*. Click Save.

When it asks you how you want to create the textbase structure, accept the default of "Define new textbase."

This brings up a dialog box where you will define the fields that make up the database structure.

The first field I'm going to create is a record number. This is not a field that represents some attribute, and it's not something that an end user would likely search on. However, it's a matter of good database design that every record have a field that uniquely and unambiguously identifies it. I choose some descriptive name for the field, say, "RecNo". Now I indicate the type of data that this field will contain, by clicking the Field Type drop down menu.

DB/TextWorks provides a field type called "Automatic Number" that will automatically fill in a number in sequence, so the first record added to the database will be no. 1, the next one no. 2, and so on. That's all I need to do for this field, so now I click the Add button to add the field to the database structure.

The next field will be for the surface texture of the rocks and will contain values such as *smooth*, *rough*, or *jagged*. For the name, I will type "Texture." The field type will be "Text" – this is the most commonly used field type. I also must decide how I want this field to be indexed. The choices are Term Indexed and Word Indexed. I could choose both or, for that matter, neither. I will explain more about indexing in a few minutes; for now I will choose Term Indexed (by unchecking Word Indexed). Again, I click the Add button to add the field.

Next is the Color field, also with "Text" as the field type, and also with term indexing.

Now I will add a Note field to contain free-text notes where the indexer can describe such things as where or when the rock was found, any unusual characteristics, etc. I would probably want to write a rule clarifying the kinds of information to be recorded here. This is also a “Text” field. For the indexing, however, I will select Word Indexed only.

Finally, I will create a Size field. This is one case where I will use the “Number” field type rather than “Text.” You should be aware that not everything that is a number has to use the Number field type. For example, a ZIP Code or a phone number should be considered text, because these do not represent quantities and would not be used in calculation or numerical comparison. But in this case, the Number field type is appropriate. Using it will allow a user to search for all the rocks, for example, larger than 6 cm, or between 3 and 4 cm. For the indexing, I will use Term Indexed.

If you need to make changes, click on the field in the list and make the change. For example, I can click the Note field and click the Down button to change the order of the fields. I can select the Texture field and change the name to “Surface Texture.” Be sure to click the Change button in order to register the change; if you forget, DB/TextWorks will remind you.

Let’s talk about validation. Let’s say that for the “Surface texture” field, I want to limit the possible values that an indexer may enter. The purpose of this is to improve aggregation by eliminating unnecessary variation. First I select the field by clicking on it in this list, then I click the Validation tab. Here we see the various options for defining what is valid for this field. For example, under Entry Validation, if I were to check “Field Entry Required” the indexer would have to enter a value for the field; a blank value would not be valid. Under Content Validation, I will change it from “No Content Validation” to “Use Validation List.” I click the Change button.

Using a validation list mean that only the terms on the list can be entered in the field; the indexer will not be allowed to use other terms. But first I must create that list, which I do now by clicking “Edit List...” In the entry box at the top, I type each term and click the Add button (or carriage return). I will add the terms “smooth,” “rough,” “jagged,” and “other.” I then click OK. In the list of fields I now see ‘Valid,’ meaning some type of validation is now being applied to this field. Although it’s tempting to use Validation lists a lot, they are not always appropriate. They are best used when the possible values for a field are limited, *and always will be* limited. You probably would not use a validation list for the Size field, where the only source of variation would be inaccurate measurement or a misunderstanding of the rules for the field. I would certainly not use it for the Note field, or for any field where the totality of possible values cannot be determined in advance. In this database, in addition to the Surface texture field, the Color field might be a good candidate for using a Validation list.

I have now created my data structure, so I click Finish and then OK.

A message now pops up that tells me that there are no records yet in the database and therefore nothing to search for. The system therefore displays a record entry window. At this point, I switch roles. I am no longer the database designer; I am now a database maintainer, or indexer.

The data entry window shows the Fields that I just created with a fill-in box for each value. Notice that the RecNo field already has a '1' filled in. So ... I fill in values for the first rock:

Surface texture: Smooth

Color: Dark gray

Size: 4.5 cm [although this is a number field, DB/TextWorks is flexible in that it allows you to add extra text such as a unit of measurement]

Note: Found at Montana del Oro; perfect skipping rock.

Now I click the Save button to save the record. To enter a new record, I click the New Record button. Notice the RecNo field has incremented to '2'. I enter values for my second rock:

Surface texture: Polished

Oops. This field is controlled by a validation list, and *Polished* is not on the list. One of my options now is Browse Choices. This displays the validation list for that field, and I can now select a value from the list and click Paste to enter it in the field. I don't have to wait until I make a mistake to use this technique; for any field controlled by a validation list, I can click the Browse Choices button (that's the one that looks like an eye) to see the list of valid terms.

Surface texture: Other

Color: Black

Size: 7 cm

Note: Obsidian.

I save the record and bring up a new record form for rock #3.

Surface texture: ??

Let's say the rock is partly rough, partly smooth. DB/TextWorks does allow an indexer to enter multiple values for a field, unless the database designer has specified single entries only. Such fields are said to be repeatable. However, there's a bit of a trick to this. You can't just type *rough* and then carriage return and then *smooth*. You have to tell the system you want to enter an additional value. To do that, click the Edit menu, then New Entry. (I usually use the keyboard shortcut, which is F7.) You can see how each term is on its own line with a kind of bullet point in front of it to indicate that these are two, separately indexed terms. If I need to add a third term, I would type F7 again.

Surface texture: Rough; Smooth

Color: Pink

Size: 3.5 cm

Note: Found in northern New Mexico, 1967. Rose quartz with a piece of mica embedded in it.

OK, that's enough rocks.

I do want to show you briefly how searching works, although there isn't time to go into much detail. I am now in a third role – that of the end user or searcher. To enter a search, I click the Query Screen button. That's the one with the little Q in the corner. Notice that the query form looks very much like the record entry form. This is called Query by Example, or QBE. In essence, I create an example of what I want my retrieved records to look like, by filling in those field values that I want to see. I don't have to fill in every field; I just put in those attributes I'm interested in. For example, I could type *smooth* in the Surface Texture field. I click the Go button (the green light) to execute the query. It tells me that it found 2 records, which I can display in a report window. Notice that the word I searched is highlighted. Notice also that the record that had both *rough* and *smooth* as values was retrieved.

New query: This time, I will look for rocks that are Dark gray in Color. A trick I can use is to click in any field I want to search and click the Browse Choices button, the one that looks like an eye. When you're in query mode, Browse Choices has a slightly different meaning than when you're adding records. Instead of showing you the validation list (remember there isn't one for the Color field), it takes you into the inverted file (or the index) for that field. There I can see all the values that have been entered so far. I can also see at the left how many records in the database contain each value. Now I can simply select "Dark gray" and paste it into my query. I called this a trick, but it's really a very useful and important searching technique. Instead of searching blindly for values, I can browse within a list of values that are actually being used.

In addition to the color, I'm going to specify that the rocks must be >4 cm in size. Notice the buttons to the left of each field. These are Boolean operators, and they can be switched by clicking on them from AND to OR to NOT and back to AND. I will leave it at AND, which means both these conditions must be true for the record to be retrieved. I run the query. Only one record was found that is both dark gray in color and larger than 4 cm.

If you just want to see all your records at once, click the Find All Records button; that's the one that looks like a globe. If you need to, you can print these records by clicking File > Print. If instead of printing the records on paper you want to write the records to a file, use a different command. After you display the records, click File > Write Report to File... Choose the format you want to save the report in – I usually use plain text or rich text format – and save it under whatever filename you want. You can later insert that file into a Word document, such as for an assignment. There's more information about this in the 202 Online Resource.

If you need to change a record, you need to retrieve it first. If you have retrieved multiple records, select the one you want to change, then click the Edit Record button – that’s the one that looks like a record with a pencil over it – and you’re back in the Edit Record form.

What if you need to change the data structure itself? For example, you’ve decided you want to add a field to indicate the type of rock, or you need to change the Color field so that it’s both term indexed and word indexed. Let me say that making changes like this after you have added many records can be problematic and should be avoided if possible. In fact, in earlier versions of DB/TextWorks, if you wanted to change the structure, you would have to delete all your existing records and start over. That no longer true, fortunately, because often the need to make changes simply cannot be avoided. To make changes to the data structure, click Maintain In the top menu, then Edit Textbase structure. Click the Edit fields button, and you’re back in the data structure. Let’s select the Surface Texture field, change the name back to Texture, click the Change button. Then click the Validation tab, click Edit List, and then add, change, or delete terms as needed. I will add a new term: *Polished* – OK. Now I would have some database maintenance to do; I should go back and see if any of the records that have the value *Other* should really be *Polished*. Note that the buttons that would allow me to move fields up or down are no longer functional. Once I have added records to the database, it is no longer possible to change the order of the fields. Close this by clicking Finish, then OK. Now I’m going to switch back to the slides.

Let’s talk about indexing. In order for a field to be searchable, it must be indexed. If it’s not, then the user will not be able to search for that information. The field values in a non-indexed field are still part of the record, and may provide useful descriptive information after the record has been retrieved by other means, but the user would not be able to search for those values directly. For your work in this class, every field will probably be indexed. When a field is indexed, the database designer must decide whether it is to be *term indexed* or *word Indexed*. It’s important to understand the difference, because this has a profound effect on how records can be retrieved.

With Term indexing, the entire value you enter in a field becomes a single entry in that field’s index. Even if the value contains multiple words, it is indexed as a single, complete whole. For example, the value “Dark gray” in the Color field is indexed as Dark gray (that is, alphabetized under “d”). Someone searching for “gray” will not find records with the term “dark gray”; only record #3 in this illustration, which contains the term “gray” alone, will be retrieved. This might seem like a disadvantage, and sometimes it is, but it’s also an advantage for the user who wishes to discriminate between “gray” and “dark gray.” Term indexing is best used when the values for a field are clearly defined and perhaps limited to a set number of terms, such as those you would give in a validation list.

With Word indexing, the system takes the value of a field and parses it into words. Each word is then separately indexed, which allows a searcher to find any words *contained* within a field. This makes more sense for any field where there are likely to be multiple words, but where the exact text cannot be anticipated by the searcher. For example, in our Note fields, there's no way a user could guess the exact wording of a complete note, but a user could search for words that might be found within a note. With a query on the word "quartz," for example, the system will locate the word in the inverted file and retrieve any records where the Note contains that word. You might notice that a few of the words in the Notes don't appear in the index. With Word indexing, DB/TextWorks uses a "stop list" of words that are not indexed. These are common words like *a*, *an*, *in*, *of*, and *the* that are generally insignificant.

Before we close, I'm going to switch to Windows Explorer. I want to say something about how your DB/TextWorks database is stored on your computer, and how you can send a copy of it to other members of the class as you work in groups, or to me when you turn in an assignment. As you can see, the database we just created is made up of 11 files; they all have the name I chose but with different file extensions. You don't need to understand what these files do, but you should understand that all of them are a necessary part of your database. To send a copy of your database to someone else, you need to send all 11 files. Rather than attaching them separately to an e-mail message, you should "zip" them into a single file. First, select all the files. You can do that by clicking on the first one, and then holding the Shift key down and clicking the last one. Next, right-click anywhere in the area of the selected files, go to Send To, then Compressed (zipped) folder, and click that. A file folder with a zipper through it now appears; it contains all 11 files zipped up into one file and has the file extension .zip. You can now attach that one file to an e-mail message or upload it to Angel.

Let's say I'm now the recipient of this file and I've saved it to my computer. In order to use the database contained within it, I have to "unzip", or extract, the files from this zipped folder. Right click on the folder and then click "Extract all..." Follow the steps in the file extraction wizard. This creates a new subfolder with the 11 extracted files within it. To open the database, start DB/TextWorks, click File > Open, then browse to the folder where the files are.

By the way, the operating system on my computer is Windows XP. If you're using a different operating system, these instructions might need to be altered somewhat. [back to PowerPoint]

There is a lot more to DB/TextWorks than I've been able to cover here. You may want to review the 202 Online Resource, which has several pages devoted to DB/TextWorks. Also, for questions about particular functions in DB/TextWorks, the online help can actually be pretty helpful. Just type F1 anywhere in DB/TextWorks, and click the Index tab to browse or search for topics of interest. And as always, please post any questions you have about any of this on Angel. Thank you. This concludes the session.