Indiana University
Multi-campus Modular
Computing Curriculum

Introduction to Computing

Module # 9: Database Management Systems

Goals

- Students will recognize components of a database: records, files, and tables
- Students will identify types of problems relevant to database management applications
- Students will create a flat - file database
- Students will understand input tools, processing tools, output
- Students will identify key features of forms, queries, reports, indices.

Prerequisite Skills

- Essence of computation
- Spreadsheet concepts

1 Databases and Database Management Software

One of the most powerful applications of computers is the capability to store, organize, and retrieve large quantities of data. An organized collection of data is referred to as a database. The programs designed to manipulate databases are known as database management systems, commonly abbreviated as DBMS.

The purpose of databases is to make large amounts of information easier for humans to understand. Databases do not necessarily require the use of a computer. You probably already maintain several databases without even knowing it! Any time you keep track of information with some type of organized system, you are managing a database. Your calendar, personal phone book, and 'to do' list are all databases, even if you commonly write them with pencil (or crayon!) The telephone book is a database. Any time you keep a list of any sort, you are forming a sort of database. Databases have been an important information management tool since written language began.

Keeping track of information by hand has some shortcomings. Written lists can get very unwieldy if you have more than a pageful of entries. This problem is compounded if the information is not organized in some fashion like alphabetic order. This process of sorting is very tedious when done by humans. Anyone who files papers in an office knows how easy it is to misplace files when humans are responsible for keeping them in alphabetical order. Every time a new document is created, it must carefully be placed in the right spot. If it is ever 'taken out' and not replaced properly, it may never be found again.

The searching process can also be difficult. Looking up a word in a dictionary or a number in a phone book can be a frustrating experience if the word or name you are looking for is not spelled as you expected. Finally, these manual systems of organization can be quite clumsy when we want to find information in more than one way. For example, if you have a person's name, and you want their phone number, you can go to the phone book to find it. But what if you have the phone number or address, and not the name? A traditional phone book is almost completely useless for this kind of problem. In old detective shows, there were books printed by the phone companies for this type of purpose, but most people did not have access to them. If you think about it, the "find a person from a number" book would contain *exactly* the same information as a traditional phone book, just organized in a different order. It was necessary to recreate the entire database just to allow this amount of flexibility. If we wanted to be able to search for the name of a person with a certain address, we would need yet another entire book of the same size. With paper databases, it is impossible to search by more complex criterion, such as "tell me all the people that live on Poplar Street."

2 The problems databases help you solve

Electronic databases are much more flexible and powerful than their paper ancestors. They have made entirely new kinds of data storage and analysis practical. A database is called for whenever you are interested in keeping track of information. Electronic databases are especially helpful when you want to store large amounts of information, information with a lot of complex characteristics, or when you want the capability for complex types of information retrieval.

You will often come across a database program as a user. Many of you encountered such a program when signing up for this class. The scheduling programs of many universities have the entire course schedule and all student schedules stored in a massive system of databases. You may have examined the class schedule by interacting with such a system. In fact, you may have been able to sign up for a class online. When you did so, your personal scheduling information was added to the university database. As a user, you can usually look at a database, and you can sometimes even add to the pool of information it contains.

You might also interact with a database as a developer. In this role, you are not as concerned with the actual information that the database contains as you are with the general structure of that information; how it is organized, how it can be searched, and how the users will see it. Although you will do this less frequently than you might encounter databases as a user, it is a good idea to understand a little bit about database development, because the principles will make you a much stronger user.

3 How databases are organized

As you have seen, computers can only understand very rudimentary elements. In order to store complicated stuff like fruit, addresses, and whatever else, we have to have some kind of organizational scheme. Computer scientists have come up with some terms that are used to define databases.

3.1 Records

The most critical unit in a database is the record. If you think about it, most databases are arranged around a certain kind of entity. A phone book is a collection of telephone companies. A card catalog is an index to a collection of books in a library. You might also have a database referring to diseases and viruses, giraffes in a zoo, video tapes in your library, or anything else. The important idea here is that you can think of a database as a collection of like things. These things don't even have to be real. You could just as easily have a database of dreams, monsters, or concepts. What really matters is that all the things in the database have to be similar. For example, you might have a database of fruit in a supermarket. You might have entries in this database describing bananas, pears, oranges, and apples, but an entry describing a station wagon might not fit here. It also may not make sense to have an entry for pork chops in this database, even though pork chops might be in the same supermarket.

Each entity in the database is one record. If your database describes a bunch of giraffes, one record in the database describes one giraffe. If your database describes fruit, one record might describe one type of fruit.

3.2 Fields

Each entity in a collection shares a certain number of characteristics. Each type of fruit might have the following relevant characteristics:

- What is the name of the fruit?
- How long can we keep it on the shelf?
- When did it come to the market?
- How much does it cost per pound?
- Is this fruit spoiled?

This list of characteristics is called fields. The fields are the characteristics which describe one record. All of the records in the database must have the same fields, but the values of those fields will be different. For example, all of the fruit in the database will have a name, but the names will be different. One record will have a name field with the value 'Bananas', and another record's name field will have the value 'Apples.' A field can be thought of as a detail of a record.

Fields represent units of the computer's memory. Because the computer stores different kinds of information in different ways, each field must have a specified data type. When you create a database, you need to tell the computer what kind of data each field will be so it knows how to store the information.

Data types in databases get a little more specific than you have seen before, because databases have more specialized ways of storing things in memory. Here's a few of the common ones:

text You are already familiar with text from spreadsheets, but now we also have to specify a length. This has to do with the way that databases find records. All records in a database must be the same length (number of bytes in memory), so you have to specify the length of each text field. You might give a field that will hold state names a length of two characters, for example, and a 'first name' field a length of fifteen characters.

Q. 1

So why not just give each text field a big length, like 500 characters?

(Answer on page 9-18.)

---

integer An integer is a whole number (no fractions or decimals) including zeroes and negative numbers. (...and you told that seventh grade math teacher you would never need this stuff again!!!) Integers are pretty easy for the computer to work with and store. Sometimes you will come across long integers, which are also integers, but stored in a different way that takes more memory but allows for larger (and smaller) values.

real numbers Real numbers are integers and decimal values. As you may remember, the computer has to use a special form of exponential notation to store them in the computer. You might encounter 'super real numbers' (often called double-precision real numbers) These are real numbers that take more memory to store, but allow more decimal places.

logical Logical values (also sometimes called boolean or binary values) are values that can be expressed in terms of 1/0, yes/no, true/false, on/off, and so on. Binary fields take very little memory. (Think about it, you know how much they take!) They can be surprisingly useful. For example, in our fruit example the field that relates to whether the fruit has spoiled might be a binary field. Most of the time you can direct the database to use a yes / no or true / false notation rather than 1 / 0, but of course the 1/0 value is closest to what the computer stores.

date As we discovered with spreadsheets, dates are not exactly text, and not exactly numbers. Spreadsheets usually think of them as numbers, but databases often are a little more sophisticated about dates, and give them their own data type. You often have a great deal of flexibility about how the dates are displayed to the user, but they are stored basically as numbers.

other types There are a number of other data types available. Many database management systems allow you to store other kinds of stuff such as variable-length memos, pictures, sound files, and whatever. Some even let you create your own data types.

3.3 Tables

So far, we have thought of a database being a collection of records. To be more precise, a collection of similar records is a table. All the records in a table must have the same fields, but they can have different values in the fields. A table could make up an entire database, but as we will see later, there are databases with multiple tables.

3.4 Index

We expect tables to be sorted. Think back to the telephone book example. The traditional phone book is sorted by last names. The 'backwards phone book' might be sorted by phone numbers. "Last name" and "phone number" describe fields in a database. The field that you use for sorting the database is often referred to as the *key field*

Q. 2

Most of us still remember libraries with the old card catalogs. Why did they have an entirely separate cabinet for subject, title, and author?

(Answer on page 9-18.)

There are problems with this approach. In order for database programs to be effective, there must be some unique way of dealing with each record. It would clearly not be efficient to use 'first name' as a primary key in a database of people in a city, because there would be far too many people with the same name. To deal with this, most DBMS systems allow you to create an *index*. The index solves this problem of uniqueness with one of two strategies: Each record must have either a truly unique field or a unique combination of fields. The unique field strategy requires each record to have a field whose value cannot possibly be duplicated in the database. These fields are usually either some arbitrary number (like your account number at the bank or insurance company) or the old stand-by in America, the Social Security Number.

Q. 3

Why is your social security number used so frequently in databases?

(Answer on page 9-18.)

The other strategy uses multiple fields to generate an index. The phone book uses the last name as a primary key, but that may not be unique enough if you have a common last name. If the database system encounters a large number of people with the same last name, it then sorts that group by the first name, then any with both the same last name and first name will be sorted by middle initial. Using this combination of fields decreases the likelihood that there will be two records with the same index.

To recap, an index is some way of using one or more fields to uniquely identify each record in a table.

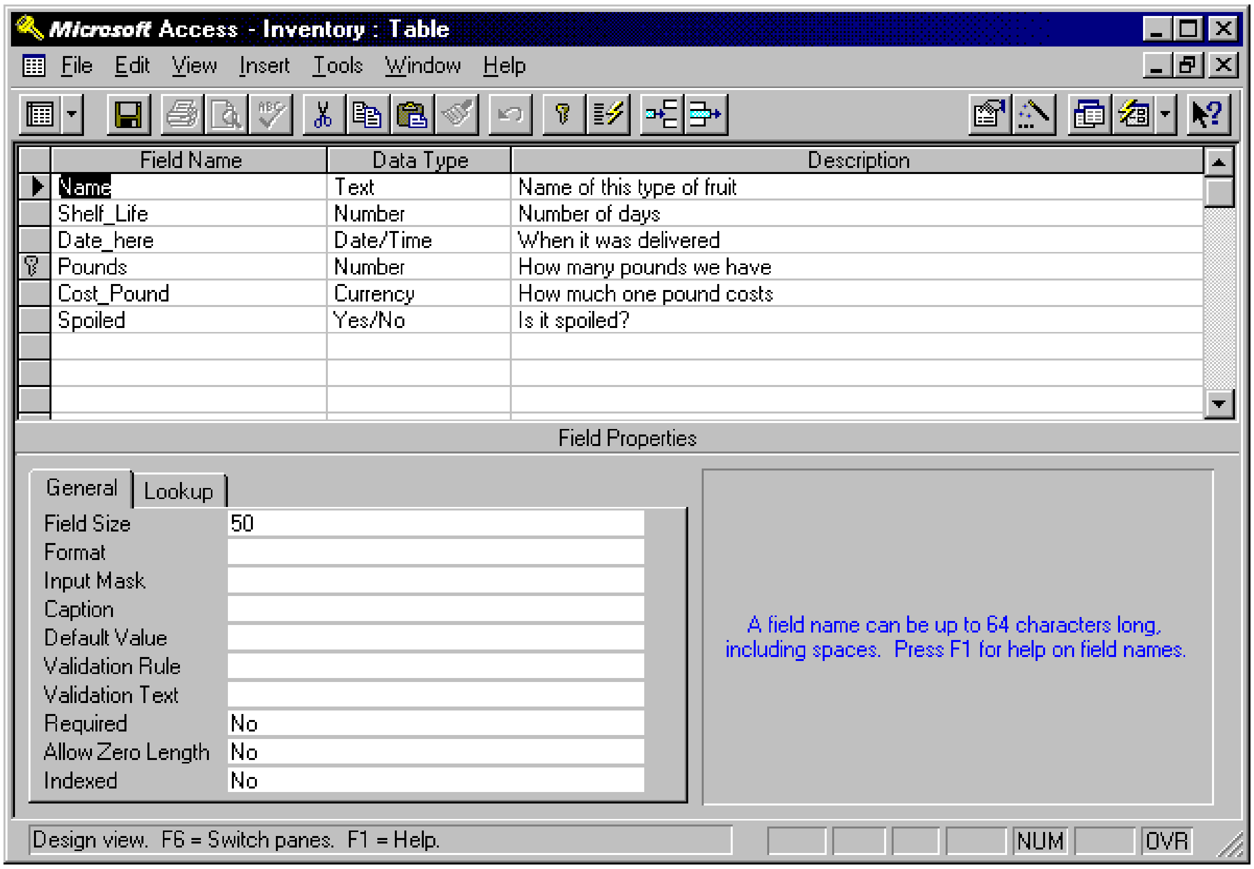


Figure 1: Data Definition Table

Q. 4

So what is the index in a yellow pages phone book?

(Answer on page 9-18.)

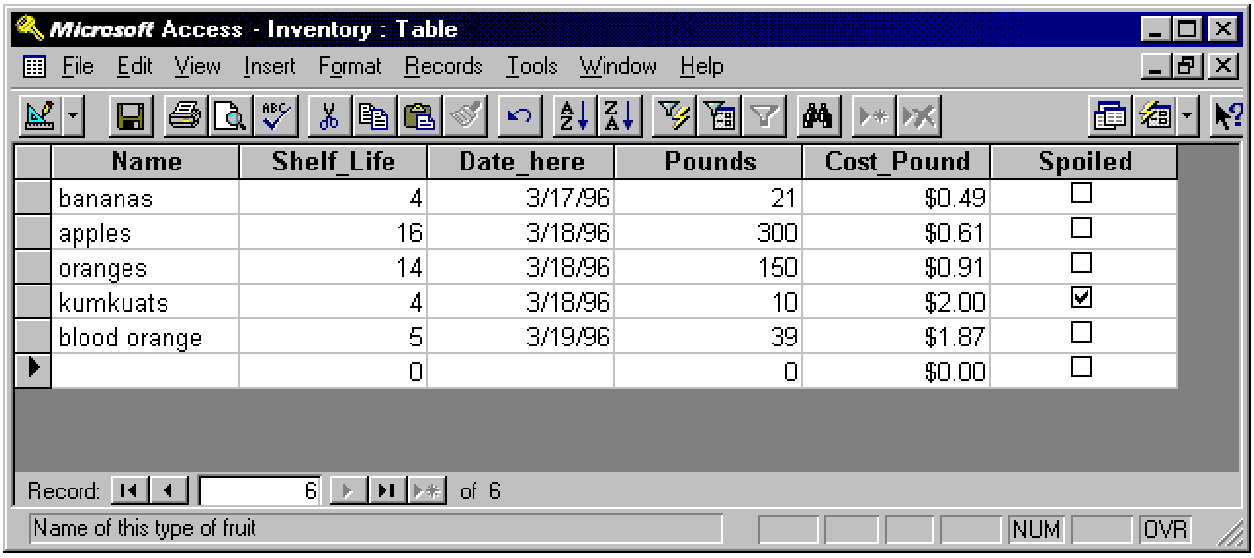
---

4 The main kinds of tools in a DBMS

A database management system is a software package that gives you a number of tools. These tools are related to the main problems one might solve with a DBMS

4.1 Data definition tools

A full-fledged DBMS will have some kind of tools for setting up a brand new database. Such a tool will allow you to define a record, the fields in that record, and the data types of each field in the record. This is a very powerful capability. Most of the time, you will not need it. It is only



Name	Shelf_Life	Date_here	Pounds	Cost_Pound	Spoiled
bananas	4	3/17/96	21	$0.49	☐
apples	16	3/18/96	300	$0.61	☐
oranges	14	3/18/96	150	$0.91	☐
kumkuats	4	3/18/96	10	$2.00	☑
blood orange	5	3/19/96	39	$1.87	☐
	0		0	$0.00	☐

Figure 2: Data Manipulation

used when you need to create an entirely new database. When you are using this tool, you are acting as the developer.

4.2 Data manipulation tools

The data manipulation tools are the part of the database the user interacts with most of the time. Data manipulation consists of a number of activities: The user will have some way of viewing all of the records in the database. She will probably also have a way to add or delete records, and edit the fields of existing records. There will usually also be ways of sorting the records by any field in ascending or descending order.

Often there will be large spreadsheet-looking devices on the screen that allow you to manipulate records. These are quite hand if you want to compare a lot of records at a time, but they can be quite intimidating. Most DBMS systems have a way of using and creating forms for user input.

A form simply displays one record at a time in a less cluttered form on the screen. If you have to do a lot of data entry, you will appreciate well designed forms.

Data manipulation is related to *input* When you use this kind of tool, you are somehow inputting information to the system.

4.3 Querying tools

Often the entire database is too much information. One main job of databases might be to find out specific information, such as “How many customers do we have in California?” or “Which stamps in my collection are worth more than $500?” You could certainly look at all the data and count out these answers by hand, but the computer ought to automate that kind of tedious work, and it can. A query essentially is a question. It returns an answer as a subset of the table. Only

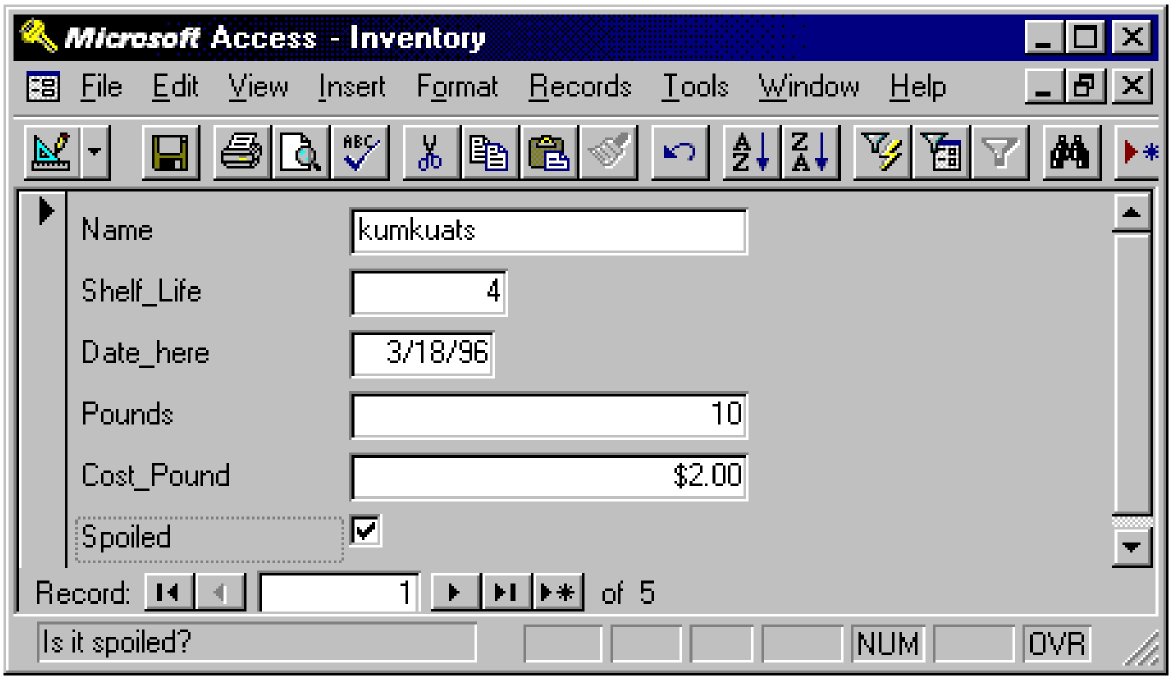


Figure 3: Form

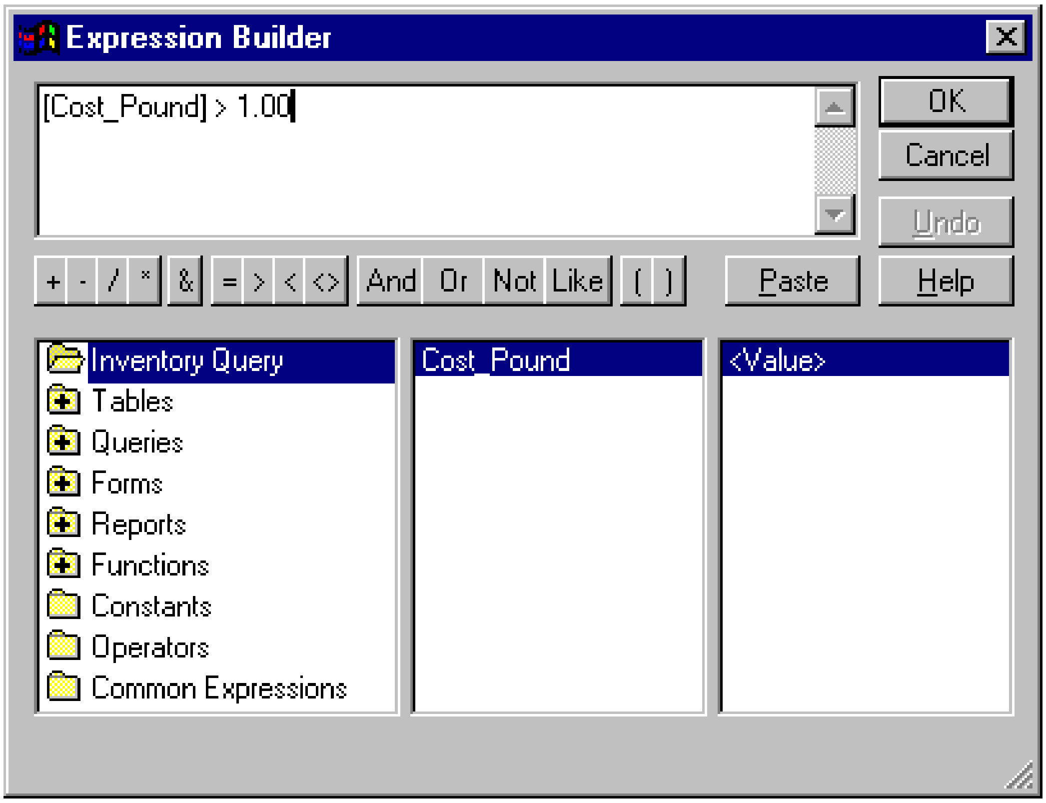


Figure 4: Query

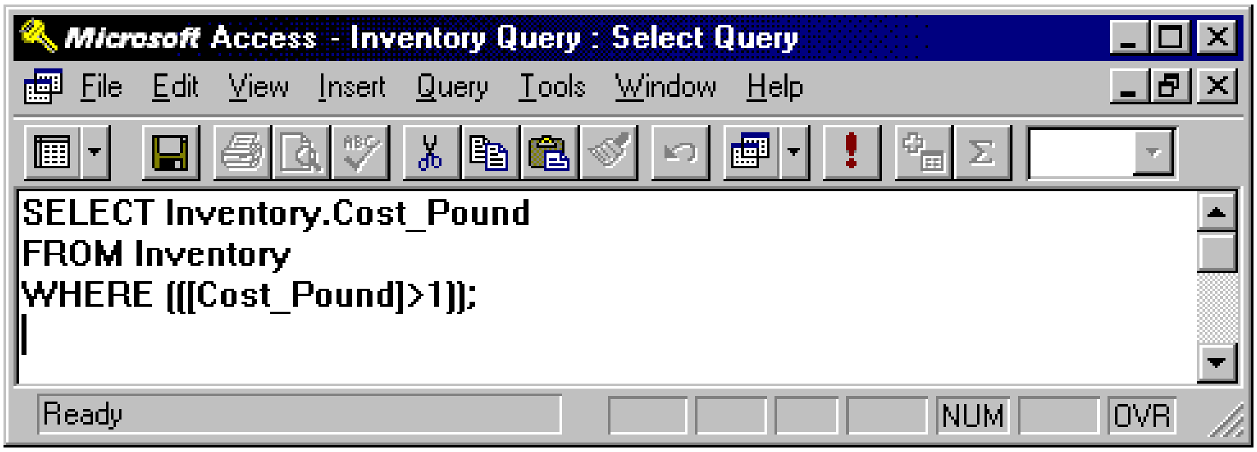


Figure 5: SQL Query

the members of the table that answer the question are returned. Queries are comparisons. They always involve field names and sample values. You tell the computer a field and ask it to search for records that relate to a specific value. In our fruit example, if we wanted to find our record for "apple", our query might be something like

NAME = "apple"

What this means to the database is "Show me all the records where the NAME field is equal to the value 'apple' ".

In fact, we might have a number of kinds of apples in our store, like Jonathan apples, Granny Smith apples, and Macintosh apples. (The fruit, not the computer!) We might be able to make a search for fields that contain a value. If we modify the above query to

NAME contains "apple"

we might find all those other apples as well, (although we might also find pineapple!) The exact syntax of the statements will certainly be different, but the idea will not change. A query consists of comparing a certain field with a certain value.

Other kinds of relationships are also possible. If we wanted to search for all the expensive fruit in our store, we might want to have a query something like:

COST_POUND ¿ $1.00

This query should return to us a list of all the records whose COST_POUND field has a value larger than one dollar.

Querying is an example of *processing* When you do this kind of work to a database, you are somehow massaging the values in the database to impart some kind of meaning from them.

4.4 Report tools

Data manipulation tools and querying tools are very handy for people with some degree of com-

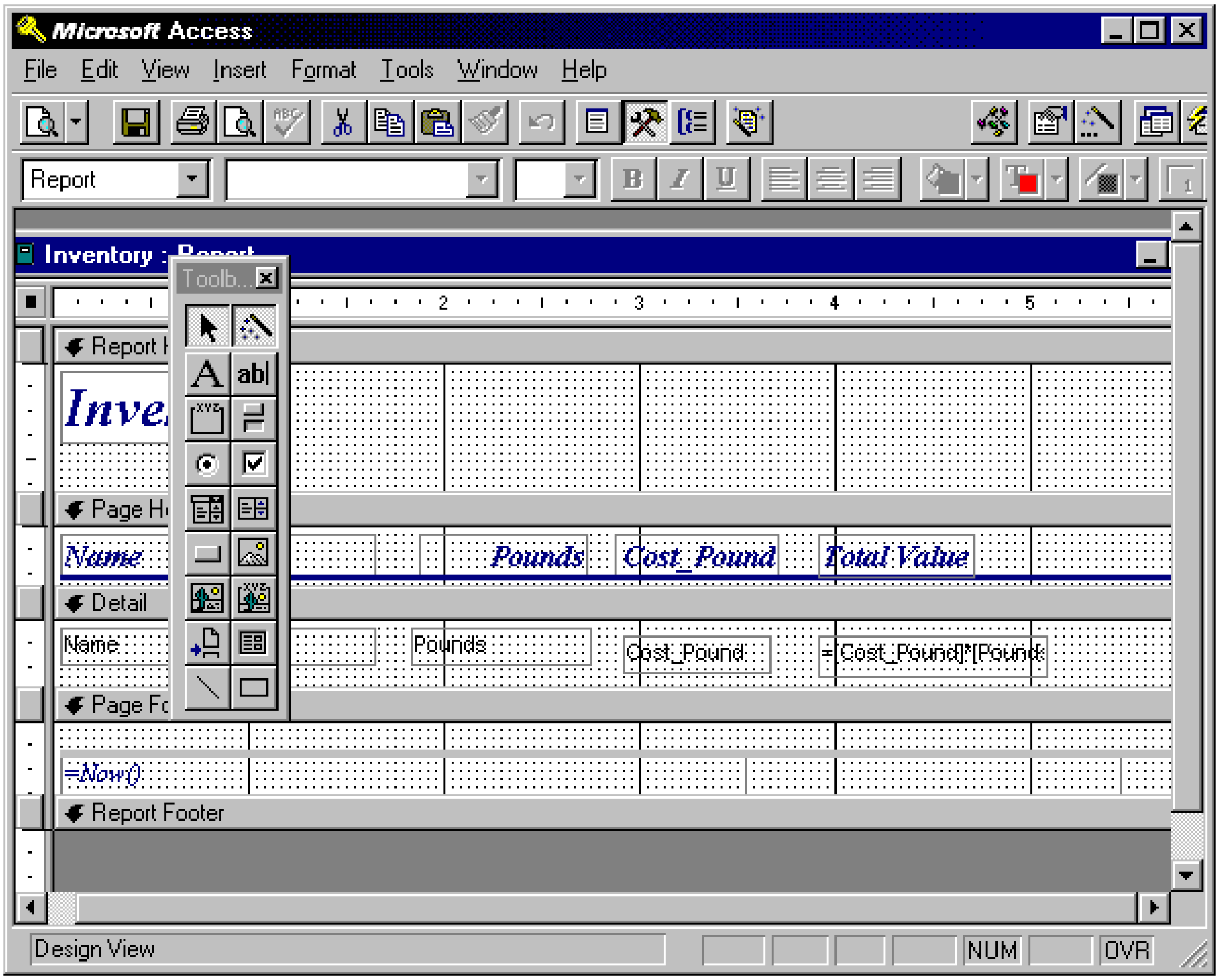


Figure 6: Report Generation

puter skill, but often computer operators are required to produce some sort of report with the data they have been working on. Report tools are a class of tools that focus on the production of attractive output from a database. A report is a specially formatted subset of a table (or tables) designed explicitly for printing or displaying on the screen. Often a report has a list of fields across the top, and each row of the report represents a record.

Many times what you see in a report is not actually the raw data from the table, but some subset of that data that is pertinent to answering a specific question. Sometimes reports are listings only of the results from a specific query, although they could also be the entire table if we wanted. Most reports do not list every field in the table. For example, if we wanted to make a report of the prices of our fruit, we might only include the NAME and PRICE/POUND field. The other fields are still there, but we don't have to include them on our report, because they would distract from the report's main job, listing the prices. (We might have another report to list values of other fields) Sometimes reports contain values that have been calculated from other fields. For example, if we had a POUNDS field for our fruit that showed the number of pounds of each type of fruit on hand, we might want to have a column of our report that contains the calculation COST_POUND * POUNDS. (Wow, that looks a lot like a spreadsheet formula, doesn't it?)

Reports do not have to look like tables. Other common report formats are mailing labels and mail merge letters. (Have you ever gotten mail from Ed McMahon? Are you impressed that he took the time to type a letter with your name and address in it?) The formatting is different in these types of reports, but the process is the same. Pieces of data from the database have been pulled out and integrated into some sort of document for printing and display.

5 Creating a database

Creating databases is an art form. People make their living doing it. You have the skills to build a simple database, and you should know enough now that you can start exploring the more advanced topics on your own.

To build your own database, you follow the same general steps regardless of your specific DBMS software.

1. STAIR! (You KNEW you couldn't get through a chapter without seeing that acronym!) State the problem and identify the tools you might use. Specifically, answer these questions for your particular database:

    - Why do I want this database?
    - What kinds of information do I want to keep track of?
    - What are the entities (will become records) I am keeping track of?
    - What are the characteristics (will become fields) of each of these entities?
    - What kind of input features do I want? (forms, a spreadsheet - thing?)
    - What will my index be?
    - What kinds of questions do I want to ask with this data? (queries)

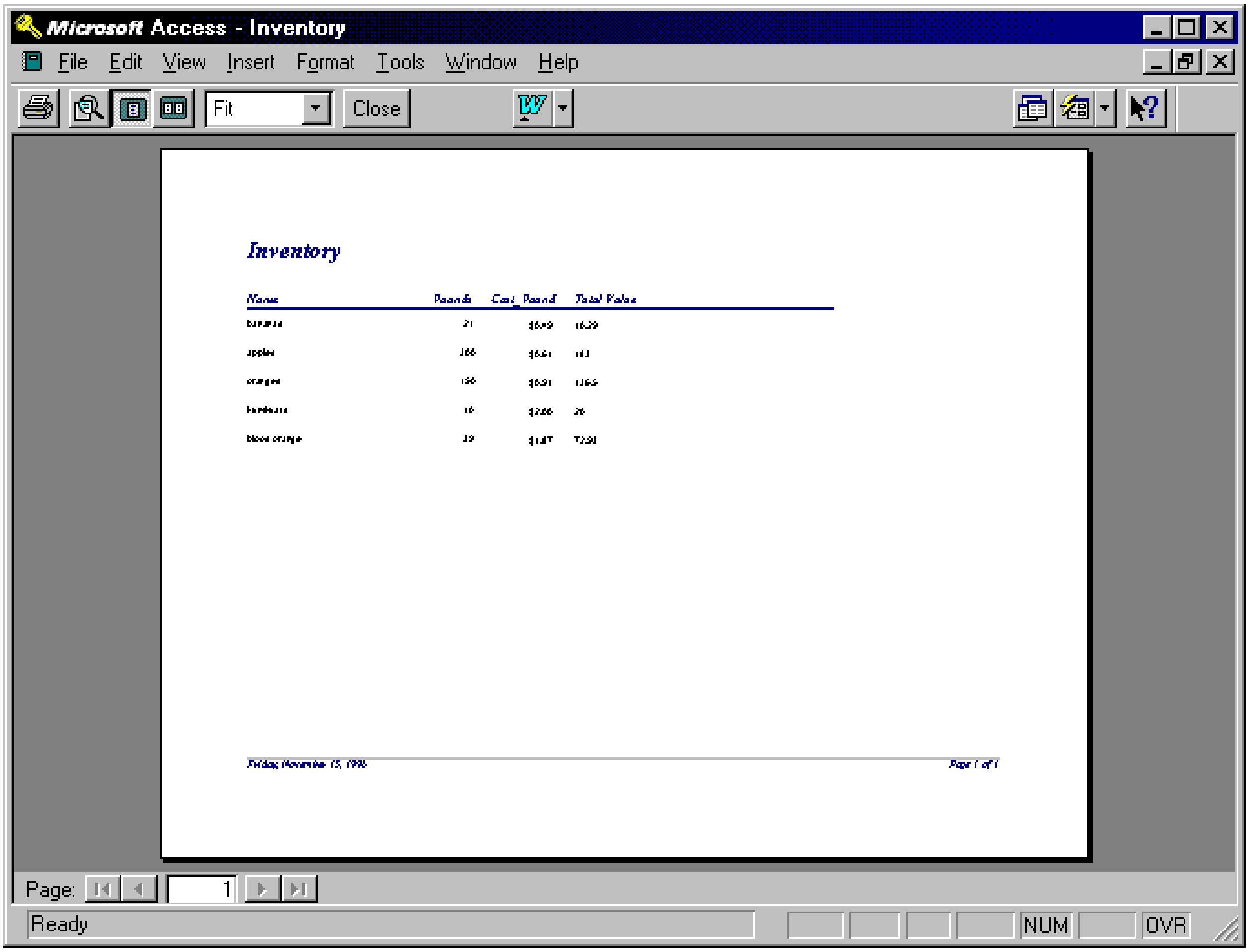


Figure 7: Report

- Do I need any special reports?
- What software will I use to develop this database?
- What tools does it have to solve each of the problems above?

2. Create the table. To do this, you will need to create a template for the records.

   - Create a template for the records

     This means defining all the fields that constitute a record. Think hard about this, because you don't want to change the record definition once you have started adding data to the database.

   - Create the fields For each field, identify the field name and type (and length if necessary)
   - Continue until all the fields of the prototype record are done.

3. Add information to the database. While you are still testing, you shouldn't add too much information, because it may get messed up. You should put in enough so that you can tell it will work. You should also anticipate errors the user might make, like typing in "four" instead of putting a 4 in a numeric field. Try making some of these mistakes on purpose, and see how the database responds.

4. generate any forms to aid user input.

   There will usually be some sort of tools available to help you in this process. If the database is simple, this step may not be necessary.

5. create any reports you need for output

   Again, this may not be necessary, or there may be a number of reports you want to generate from the database. You will probably find some kind of tool to help you with this process as well. This process may require building queries to limit the number of records shown in the report.

6 Types of Database systems

As you have seen, databases can be very complex, and their maintainance can be more involved than some of the other applications we have examined. There is a very wide range of application software for database management.

Read - Only Databases These are databases that have already been established, usually by a professional programmer or team. You interact with these databases only as a user. You can make queries of such a database, and sometimes make reports from them. Some examples: the Internet search engines, on-line card catalogs, airline reservation systems.

Pre-defined databases These databases are often commercially available, and are a shortcut for people who want to manage a database, but don't want to build it themselves. Often these are aimed at a specific topic in the business or home market such as wedding preparation, video library management, invoice handling, inventory, or payroll. In such a database, all the design has been done for you, and usually a number of queries and reports are also predefined, although you may have the capability to create more of your own. The advantage of this type of database is that you don't have to do the hard work of defining data types, so you can concentrate on adding and manipulating data. The down side is that you are stuck with whatever design the programmer gave you. If you want something more specific to your needs, you are out of luck.

Flat - file DBMS These systems generally allow you to do the things we have been talking about, although inexpensive systems may have limited capacity in particular areas. A flat file DBMS is capable of handling only one table at a time. For home use, this is often just fine. You don't need any more to deal with your Christmas card list, or stamp collection inventory. The user-friendliness of DBMS software varies tremendously. Some systems (particularly the database packages bundled with integrated suites) are relatively easy to use and provide point - and - click interfaces. Other DBMSs are nothing more than programming languages - a list of commands that you have to put together on your own. In general, DBMS applications are more involved than other standard applications.

Relational DBMS These are the mainstay of the database world. A relational database is much like a flat file database, except it allows the use of multiple tables. Imagine the following example:

A programmer has been hired by a garage to create a database for their operations. After talking closely with the owners of the garage, she notes three main needs:

1. They want a way to keep track of each part in the inventory. When a part is used for repair, it should be noted, so the owners can order new ones.
2. They want to keep track of the customers who use the garage. When Mrs. Jones, comes in, for example, they would like to know what services she has ordered in the past, and what has already been done to her car. They also want to send her postcards informing her of special deals and tune-up times for her car. This will increase efficiency and customer service.
3. They want to keep track of each transaction. This database will be closely tied to the cash register and should give a report that explains exactly what has been done to her car, and should be a very nice receipt.

At first, it may seem like three entirely different databases are necessary. Although the customer database may share some information (like part numbers) with the inventory database, it appears to be a different database. It will certainly require a different table, because the fields that describe a part in the inventory are very different than the fields that would describe a customer. Likewise, the receipt table may share some fields with other tables in the system (customer name and part number) but the fields that describe a receipt are clearly different than those that describe a part or those that describe a customer, so a third table is necessary.

What we see here is three distinct tables that share some fields with each other. These tables are distinct but *related*, thus the database is a *relational* database. (Just when you think you can count on those computer scientists to come up with a completely meaningless term, they give you one that makes sense!)

The big DBMSs like Oracle, Paradox, Access, and so on are examples of relational databases. Like flat - file database management systems, these DBMS packages come in all flavors, from fairly straightforward point - and click to cryptic but very powerful programming languages.

7 Databases and Ethical Concerns

Any tool with as much power as databases can give must have some caveats, and databases certainly do. Databases are easy to build and maintain, and there is no way to keep unscrupulous people from using them as well as those that intend only good. Even when they are not deliberately used for foul purposes, databases can have unsuspected side effects.

Privacy is a major concern of the present and very near future. Although there have been databases since written language began, they were often very inefficient. There was usually little danger of the information in a database getting into the wrong hands, because computers tended to not be connected, and there was relatively little information being kept. Now the increased ease of database development along with increases in computer connectivity, processing and storage capacity have made the amount of information stored on databases increase exponentially. Now nearly every organization or corporation has some sort of database. When you give your name and address to a company by entering a contest or filling out a questionnaire, your information is almost certainly going to be entered on a database. That corporation is not always required to keep the information about you to themselves. Have you ever wondered why you get the combination of junk mail you do? Every one of those things in the mailbox has a mailing label on it generated by a database. You may have given them the address directly, or it may have been sold to the company by another company who you gave your information to.

Security of information is another concern. As more and more computers are linked via the Internet, there are more opportunities for data to be taken by those who should not have it. Be very careful using your credit card number on the Internet. Where will it go on the way to its destination? Will it be carefully guarded in the destination database, or can somebody just break in and assume your identity? Take the same care with other personal information. Who will get this information? Do you want them to have it? The legal system is wrestling with issues of privacy over the Internet. Technology is changing more quickly than the legal system can keep up.

8 Laboratory Assignment

1. Create a simple flat file database for mail merge in your word processor. Most word processors have a tool that allows you to create a simple database and mail merge without having to go to a separate document. Your table should have at least 3 fields and five records. Make a letter that can read the fields.

2. Use a full-fledged DBMS to create a telephone directory. Your database should have the following characteristics:

   - One table

- At least three fields per record
- At least five records
- An index field

3. Add more advanced features to your telephone directory:

   - A form for input
   - A report for output
   - A query for looking at specific records

   Don't forget to utilize any shortcuts or tools the DBMS gives you as shortcuts!

9 Summary

The collection of information into organized bundles has been with us for a long time, but computing technology has made the management and manipulation of databases far more practical than it has ever been in the past.

The software used to create and maintain databases is called database management systems, and these systems come in a number of levels of complexity ranging from read - only databases and pre-defined databases to complex programming languages.

Databases are organized into tables, which are collections of records. One record describes one discrete entity. The properties which describe a record are known as fields. A field is defined using a specific data type. Records are sorted by indexes, which are related to one or more key fields.

The main types of tools one might find in a DBMS relate to the principle jobs of such a system. Data definition is the creation of new tables, records, fields, and indices. Data manipulation is the act of entering information into an existing database. Querying is using special tools or languages to ask for sub sets of the data in the database, and reports are output for the printer or screen which format the information nicely.

10 Vocabulary

Database: an organized collection of information

DBMS: Database Management System. A software application designed to help users create, maintain, and manipulate databases.

User: The person who uses a database. The user generally does not worry about the design of the database, ot its components.

Developer: The person who designs a database and its complement of forms, reports, and other support materials.

Record: The central entity in a database. One record describes one discrete element.

Field: A detail of a record. A field is usually assigned a data type, and sometimes a length. A number of fields make up a record.

Table: A collection of like records. All the records in the table will have the same field definitions, but different field values.

Data types: The type of organization used to translate binary values into a more useful type such as real numbers, dates, and text.

Integers: The whole numbers including zero and negative numbers. (No fractions or decimal values)

Real Numbers: Integers and decimal vaules. Often DBMSs will have single and double precision real numbers available.

Logical Fields: A datatype defined by a yes / no, true/false value. Also referred to as binary or boolean fields.

Index: A sorting and searching tool organized by one or more key fields.

Key Field: A field used as part of an index. The first one used is often called the *primary key*

Data Definition: The process of creating a new database, especially the record definitions. Only the developer does this.

Data Manipulation: The process of adding new information to the database. This may be done by the developer and / or the user.

Querying: Requesting a subset of the database in terms of the values of fields. Queries involve comparing some field or fields to some value or values.

Reporting: The process of creating specially designed output of the database. Often tied to a specific query.

Flat - File: A type of database characterized by one table only. These are sufficient for much home use, but are not powerful enough for many business applications.

Relational: A type of database characterized by multiple tables. The tables are different, but related. They may have common fields, or other types of relationships.

Answers to Questions

1. The database will reserve the specified amount of memory for each record regardless of the values in that record. If you set up a state field with a length of 500 characters, and the user always puts in two letter abbreviations, the DBMS will 'pad' the field with 498 spaces. All of those spaces will be held in memory and on the disk each time the record is accessed, and waste a lot of valuable real estate on the disk and memory. Don't make fields any bigger than you have too, but don't make them too small.

2. The card catalog was actually 3 copies of the same database, sorted on different fields! The author catalog is sorted on the author field. (I don't have to tell you the others, do I?)

3. Because it makes an *ideal* key field. Every citizen in the United States has one, and they all are different. They also all have the same format. Social Security Numbers were never intended for this general use beyond government databases, but it has become very widespread.

4. The type of business, followed by the name of the business