– describe how a table is designed and filled.
– describe a form and its use.
– know the appropriate time to use a sort or a query.
– see the value of key fields, common fields, and multiple-field sorts.
– describe the use of *and* and *or* in a query.

Lesson 1

Why Use It?

Terms to Know

database. A collection of related tables and their accompanying queries and reports.

field. A placeholder for a particular piece of data.

record. All the fields of data about one entity.

table. A collection of records.

What Is a Database?

To understand an electronic database, we will compare it with a manual database like the ones kept in a filing cabinet. Let's say the file drawer shown on page 3 has in it information about pupils in a particular school. Inside the drawer are numerous file folders. When you open a folder, you find pieces of information about one student. Another drawer contains information about school equipment; and still another is filled with facts about vendors, the businesses that sell products to the school. The entire collection of information—the whole filing cabinet—is a database.

Now let's compare that collection to an electronic database. Since a database is a collection of related **tables**, we could say the above database is a database of a school. One table contains student records, another contains equipment records, and the third contains vendor records. In the student table, there is a **record** for each student. Each record has several pieces of information, and each piece is stored in a **field**. A field is a placeholder in every record in which information is entered. Therefore, a database is made of tables, tables are made of records, and records are made of fields.

Genesis 11:10-26 is a simple one-table database—a collection of records about Shem and his descendants. Each record gives these details (fields): the descendant's name, his father's name, his age when a son was born, his son's name, and how many years he lived after his son was born.

The computer can manipulate data in an electronic database in meaningful and useful ways. With a computerized database, it takes only a matter of milliseconds or microseconds or, at the most, minutes to sort through hundreds of thousands of records. The importance and value of an electronic database becomes more apparent when the number of records reaches 900 or 9,000 or 900,000 or more. Sifting through such a huge amount of data by hand would be

An old-fashioned manual database

impossible to do in a timely manner. But even with only the nine records in the Genesis database, if you were going to arrange the names alphabetically, it would require rewriting—or reshuffling if you used index cards. Doing calculations based on their ages would require extra math. However, with an electronic database the computer can do any of these operations with ease.

Using an electronic database has four major advantages:

1) changes can be made easily,
2) records can be sorted quickly, easily, and repeatedly in a multitude of ways,
3) records can be cross-referenced and analyzed, and
4) meaningful reports can be produced.

Define these words.

1. database _____

   _____
2. table _____
3. record _____
4. field _____

Study the three records given in this table. Use your Bible to fill in each field for three more records.

GENEALOGY TABLE

	NAME	FATHER	AGE	SON	YEARS AFTER SON
	Shem	Noah	100	Arphaxad	500
	Arphaxad	Shem	35	Salah	403
	Salah	Arphaxad	30	Eber	403
5.	_____	_____	_____	_____	_____
6.	_____	_____	_____	_____	_____
7.	_____	_____	_____	_____	_____

8. How many fields are in the table above? _____
9. How many records are now in the table above? _____

Number these words in order from smallest to largest.

10. _____ field _____ record
_____ database _____ table

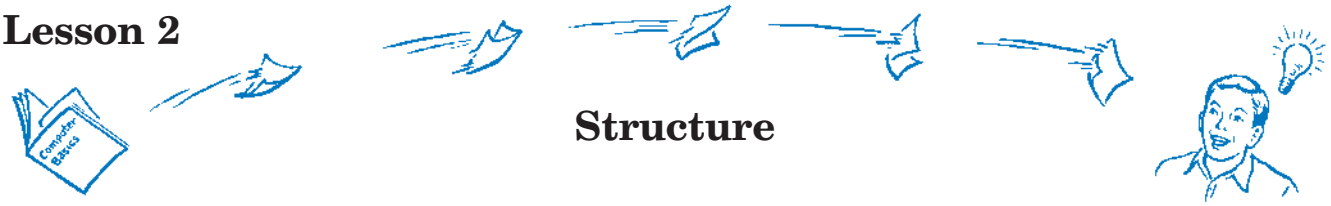
Create another field to represent age at death and give the data for the field.

11. Field name: _____
12. Shem: _____
13. Arphaxad: _____
14. Salah: _____
15. Eber: _____
16. Peleg: _____
17. Reu: _____

List four advantages of databases.

18. _____
_____
_____
_____

Lesson 2



Structure

Terms to Know

common field. A field of the same name, length, and type that appears in more than one table, and thus joins the two tables together.

key field. A field that will make every record in a table unique.

one-to-many relationship. A relationship between common fields in two tables in which the value in the common field of one record has many matching values in the common field of the records of the other table.

relational database. A database having its data separated into different tables but related to one another through a common field or fields.

Planning. A good database requires careful planning. You must think about the kinds of things you want to put into the database before you design the database. But even before you can decide what you want to put into it, it is important to think about what you want to get out of it. This will help you to design it properly. In some ways, it seems a little backward—to plan what you will get out of a database before you ever put anything into it. But it's true. Otherwise you could spend a lot of time making a database and entering information and then realize that getting the desired information out of it is very difficult because of its structure. In word processing or spreadsheet software, changing things around "after the fact" is usually a simple process. With a database, that "after-the-fact" change is more complicated.

The starting point of any database is the fields where the information is stored. These fields make up records that must be structured to make tables before any information can be entered.

Think about these questions when setting up fields: Will they contain numbers? words? dates? a combination? What is the largest amount of information you may need to put into one field? How many fields will be needed? What names should the fields have? How will fields in different tables be related to one another? These and other questions are important in making the final database a valuable tool.

Structure. In planning a table, determine how many fields you need—their names, their lengths, and their sizes. What you name a field is not important to the computer, but it is important to you. The computer would be perfectly satisfied if you named a field *TDLMPZ,* but it wouldn't make much sense to you, and you might forget what the name stood for (or how to spell it) when you want to use it in a form, query, or report. Use words or abbreviations that are easily readable and understandable.

How long should each field be? Consider a field for last names. What is the longest last name you expect to enter? If you expect it to be no longer than *Rodeffer,* then the field could be only eight characters long. But if you had the last name of *Rothenberger,* you would need at least a twelve-letter field. It is best to pad the field with extra space for that unexpected, extra-long last name.

For addresses, it is a good idea to have separate fields for street address, city, state, and zip code.

Now you will need to decide what type of information goes into each field. The most commonly used types are text, numbers, date, time, and calculated. Text is a good choice for last name because text fields can have any and all characters in them. Using field types prevents errors by not allowing data to be entered in the wrong field. For example, a number field will not accept regular text—only numbers.

Choosing a field's name, type, and size is known as defining a field. After each field in your table is defined you can start entering information.

Study this sample table structure. Answer the questions about it.

NAME OF FIELD	TYPE	SIZE
TITLE	Text	35
AUTHOR	Text	25
COPYRIGHT	Number	4
PUBLISHER	Text	30
PAGES	Number	3
INTEREST LEVEL	Text	12
SHELF	Text	12
COST	Number	5
PURCHASE DATE	Date	8

1. How many fields does this table have? _____

2. Why is the size of the TITLE field larger than all the rest? _____

_____

3. What should your first consideration be in setting up a database?

_____

4. What three things go into defining a field?

_____

_____

Suppose you wish to keep a record of the people who attend your church. You want to note who they are, where they live, in which family they belong, when they started attending church, their birth dates, and how far they drive to church. Plan the fields for your table of church attendees here.

○ 5.

NAME OF FIELD	TYPE	SIZE

Calculated fields. A calculated field can save you time and work and make the information in your database more accurate. The computer puts its own data into a calculated field based on a calculation you have indicated. For example, in the genealogy table you made from Genesis 11, instead of having to calculate the age as you did in the exercise in the last lesson, make the computer do the work. Make a new field for TOTAL AGE. Then tell the computer to put into that field the sum of AGE and YEARS AFTER SON (TOTAL AGE = AGE + YEARS AFTER SON). The computer will figure the total age of each person for you. Calculated fields can be very complex; but if the formulas are entered correctly, the computer will calculate them perfectly every time.

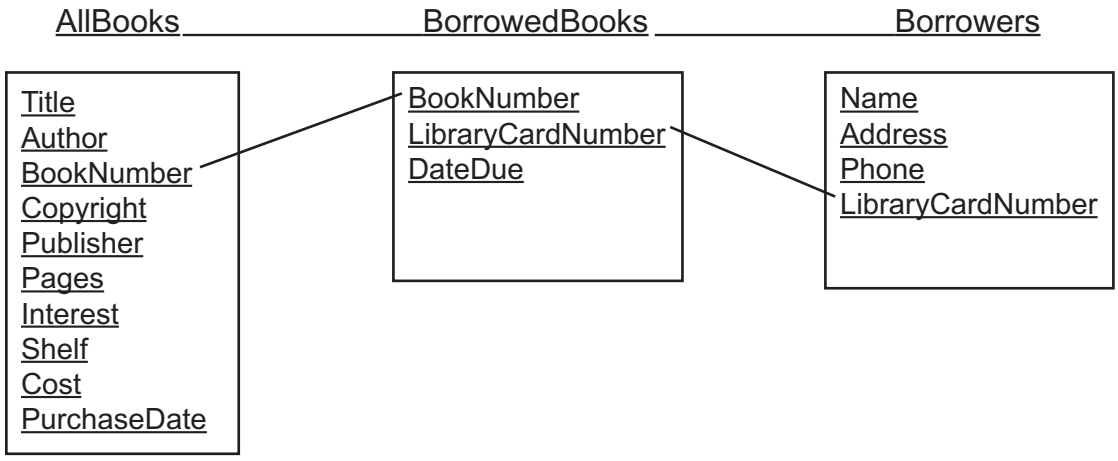
Describe how you could make a computer calculate a person's current age based on his birth date. Why is this better than just typing in a person's age into a record?

6.

More tables. Most databases have more than one table. **Relational databases**, as these multi-table databases are called, have a number of advantages. They allow the computer to deal with the data in a greater variety of useful ways. They require a minimum of data entry because the same information goes into multiple fields. And the fewer times data has to be entered, the fewer the opportunities for error.

A library database is a good example of using multiple tables. The table presented earlier was designed to keep book information. This table could keep track of all a library's books. A library could use another table that listed people who borrowed books. This table would have fields to record names, addresses, phone numbers, and library card numbers. A third table would record the books borrowed and would contain fields such as book number, card number, and due date. The computer can then combine data from all three tables using the link formed by **common fields**.

This ability to combine information from multiple tables eliminates the need for entering the same information many times. The computer can take a book number in the "BorrowedBook" table and look up information in the "Borrowers" table about who has the book out, its due date or give information about the book itself from the "AllBooks" table. The next illustration shows how the relationships look.



Library Database – 3 Tables

Key fields. A **key field** guarantees that each record in that table is unique, even if all the other fields in that record are the same as they are in another record. Look at the "Borrowers" table in the library database above. LibraryCardNumber is the key field for that table. Suppose William Hostetler and his son, William, both have library cards. The name, address, and phone number is the same for both. Their library card numbers, however, are different. Thus the records for William and his son will not be exactly the same. This is important if the computer is to accurately pull related records from different tables. Can you see why BookNumber is a good key field for the "AllBooks" table? Think of the potential for confusion if the Title or the Author field was the key field. It is not absolutely necessary for every table in every database to have a key field, but as your database becomes larger, key fields become more important. Notice that common fields are often key fields.

Relationships. Along with the ideas of common fields and key fields is the concept of relationships between tables. The lines in the library database illustration show that there is a relationship between the common fields, but consider the direction of the relationship.

First, think of the relationship between BookNumber in the "AllBooks" table and BookNumber in the "BorrowedBooks" table. For each occurrence of a particular book number in the "AllBooks" table, there could be many matching entries in the BookNumber field in the "Borrowed Books" table because the same book gets borrowed many times. That is sometimes called a **one-to-many relationship.**

The same could be said of the LibraryCardNumber field in the "Borrowers" table as it relates to the same field in the "BorrowedBooks" table. For each occurrence of a particular library card number in the "Borrowers" table, there could be many matching entries in the LibraryCardNumber field in the "BorrowedBooks" table. Why? Because one person can borrow many books, and each time he borrows a book, another record is made in the "BorrowedBooks" table.

Sometimes, having only one field in a table designated as the key is not sufficient for making every record unique. Look at the "BorrowedBooks" table in the library database as an example. Since the same book could be borrowed many times and thus appear in the table many times, BookNumber could not serve as a key field. Neither could LibraryCardNumber, because any one person could borrow many books, and his library card number would be a part of many records in the "BorrowedBooks" table. In such a case, a combination of fields serves as the key, meaning that any particular combination of values in those fields is not found more than once in the entire table. What fields would be the key in the "BorrowedBooks" table? Why would the software require the user to fill every field designated as a key field?

Write *true* **or** *false*.

7. _____ A field is defined by its name, type, and size.
8. _____ Different tables can have the same fields in them.
9. _____ A database can only have one table in it.
10. _____ A one-to-many relationship joins tables on a key field.
11. _____ The information in a key field will make that record unique.
12. _____ Calculated fields cause more trouble than they are worth.
13. _____ A database with multiple related tables reduces the amount of data entry needed.
14. _____ A common field is one that appears in more than one table.

Tell why you should use a key field.

15. _____

_____

Give three advantages of relational databases.

16. _____

_____

_____

LOOKING BACK . . .

Define these terms.

17. database _____
18. table _____
19. record _____
20. field _____

List four reasons for using electronic databases.

21. _____

_____

_____

_____

Complete the sentences.

22. Databases are made of _____ , which are made of _____ , which are made of _____ .
23. Databases have been maintained for a long time, even by people in the _____ .
24. A filing cabinet is an example of a manual _____ .
25. Each individual record contains _____ for different information.

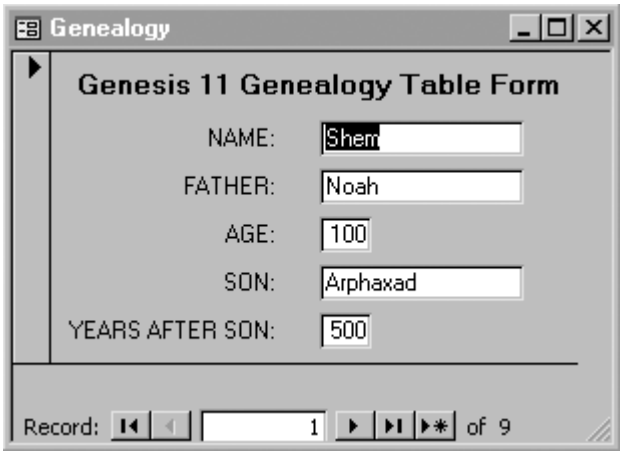
Lesson 3

Adding and Editing Records

Term to Know

form. A cover-up for a table; it will often mirror a paper form and make data entry easier.

Adding the records. Once the database structure has been defined, the records can be added. You can add records directly into the table or you can use a **form** like this one. The user designs the form to fit his liking.



A form makes it easier to enter records into a table when there are many fields to fill or when the person putting the data into the computer is typing it from a paper form that has a layout similar to the one on the computer screen. Notice that this form shows all the fields of one record. Compare it to the table view of the Genealogy Table on page 4. Using a form allows you to see one record at a time, eliminating the possibility of typing something into a field while on the wrong record. The form is simply a cover-up for the table and can be used for adding records, editing records, deleting records, or just looking through them.

Editing the records. Any change that needs to be made to a record is easily done. If a church member's address needs changing, type in the new address over the old one, and the new address replaces the old one. If an item in inventory has a price change, type the new price over the old one, and the change has been made. There's no erasing, no crossing out, and no correction fluid. If a minor change needs to be made to a field, it can be edited without having to type the whole entry over again. The whole process is much easier than a pen and paper system.