6 INTRODUCTION TO THE *BASE* DATABASE
Keeping student records

LEARNING OUTCOMES

In this tutorial you will learn about databases in general and about the *LibreOffice Base* database in particular. The concept of a database is not quite as easy to grasp as that of a word processor or a spreadsheet. So if this is the first time you are learning about databases you might like to read over the lesson first, away from the computer, and perhaps highlight the text to ease understanding.

Specifically this tutorial will introduce you to the following:

- the general idea behind a database
- planning a new database
- creating a database template
- adding records to a new or existing database
- viewing the data in a database
- clearing entries and records from a database
- making a backup copy of a database

A caveat before you begin: You'll find it easiest to use the tutorial if you follow the directions carefully. On computers there are always other ways of doing things, but if you wander off on your own be sure you know your way back!

6.1 AN OVERVIEW OF THE DATABASE CONCEPT

The database component of *LibreOffice* is designed to act as an "intelligent"[1] filing cabinet. Not unlike the manual filing cabinet you may have in your home, it enables you to:

- collect together sets of related data and, if you are careful and conscientious, keep the data organized;

- update the data once you've collected them, adding or deleting records, or changing the contents of existing records;

---

[1] The word "intelligent" is enclosed in quotes so as not to create a false impression. With regard to computers, the word is overused, and strictly speaking inappropriate. It also can be misleading. The only "intelligence" exhibited by these otherwise "dumb" machines has been programmed into them by intelligent human beings.

A database like *LibreOffice*'s *Base*, however, does everything that a manual filing cabinet does, plus it is an *electronic* filing cabinet which makes it so much easier for you to quickly *work with* the data you have gathered—access them, sort them, create reports, merge the data with other documents, and so forth.

How is a database organized?

Base is an object-oriented relational database management system. It is designed to help an organization—such as a company or an institution or a school or school district, and also the individual teacher—to collect, organize, manage, and store data in such a way that they can be easily accessed and made available to users in a variety of useful ways.

The various types of database objects that you can create using *Base* are listed in the *Database* Objects menu in the left hand frame of any *LibreOffice* database that you open (Fig. 6.1).
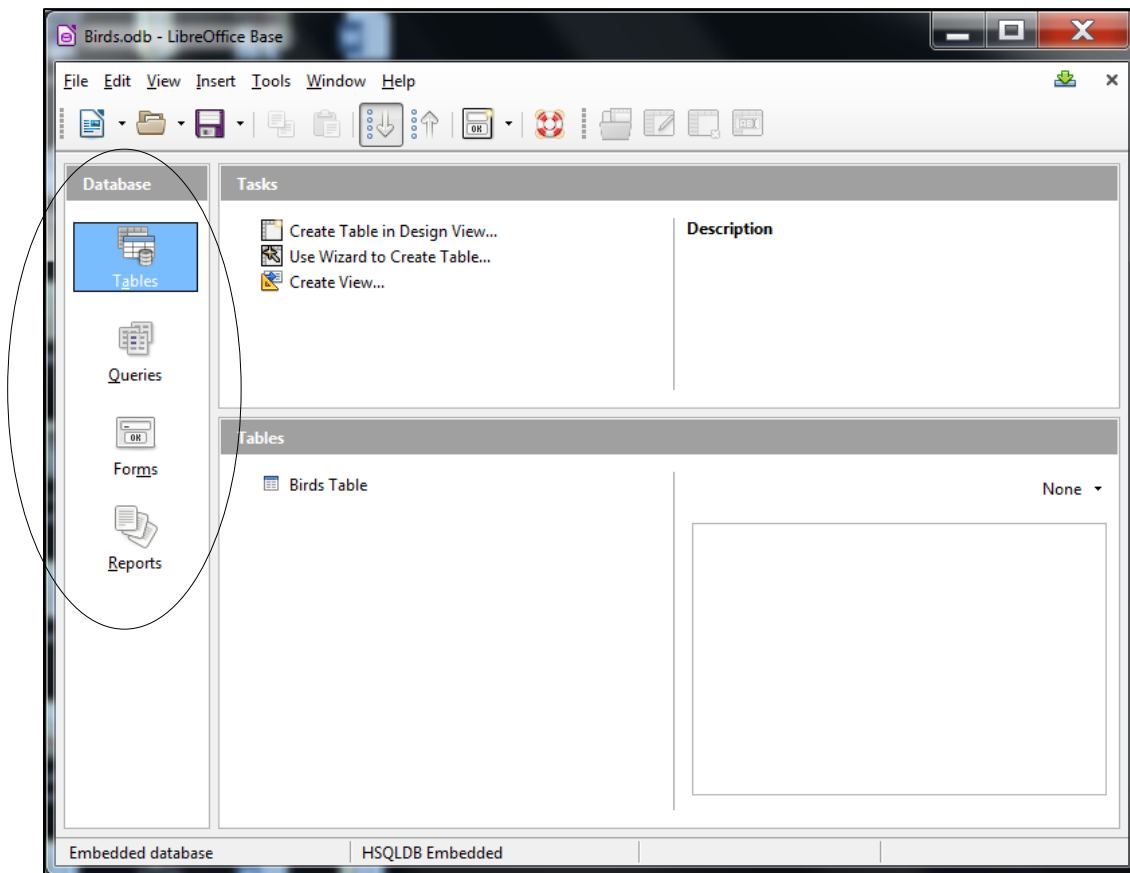

Fig. 6.1 The *Base* database objects

Let's take a look at each of these objects now to see how they can be used to help you organize and access the information collected into a database (in this case a database of information about different birds).

- A **Table** is the fundamental object in a relational database and tables use rows and columns to present the data—rather like a spreadsheet. You will learn about Tables and in this lesson, as well as how to use the *Design View* to create a table, add fields to a table, and define the type of data that is in a specific field.

- A **Query** makes a request to the database, asking it to find some specific data that are stored in the database. You will learn more about queries in Lesson 7.

- A **Form** gives the user another way of looking at the data in the database. Whereas a Table allows the user to view many records at once, a Form displays the contents of just *one record at a time*. Forms are very useful for entering data into new records or updating the data in existing records. You will learn about Forms in this lesson.

- A **Report** is used to present a selected set of information from a database in a format that is neatly laid out and thus easier for the user to mentally digest. You will learn about reports in Lesson 7.

The general terminology used to describe data storage

Words (terminology) are our stepping stones to knowledge. For the most part, personal computers today still use electronic disks and drives of one sort or another to store data. The *Windows* computer operating system, like other operating systems, organizes data hierarchically on the disk using documents or files which are usually stored in directories or subdirectories that are part of the overall directory file management system.

Within documents, different terminology is used to describe the organization of data depending on the type of document in question.

- *Word processed* documents organize the data (mostly text) in paragraphs and sentences;
- *Spreadsheet* documents organize the data (mostly numbers) in rows and columns;
- *Databases* organize the data (of all kinds) into records and fields.

***Base* database terminology**

A *Base* database is able to contain thousands, hundreds of thousands, even millions of records, depending on the purpose for which the database has been designed. A teacher may have dozens of records in his or her class databases; a school district may well have hundreds or thousands of records to collect and maintain; a company like GM will have hundreds of thousands, maybe even millions of records in multiple databases containing information about employees, products and other kinds of data going back decades; a US government office, such as the IRS, will surely have databases with millions of records.

Maintaining masses of data such as this is hard. Actually, maintaining even small amounts of data is hard, so it's not surprising that the computer has been brought to the rescue of those of us who need to manage data.

To avoid a hodge-podge of data organized at random, people have come up with the idea of organizing data into *records*. Records contain *fields*. Each field contains *entries* made up of text and numbers, which are the basic building blocks of data.

For example, let us say you have given your students an assignment to study birds and they must build a database to store the data collected. Each student decides to set up a record for every bird sighted.

Each record will include several field names, such as Common Name of the bird, Type (Family) of the bird, Size of the bird, Habitat, etc. A field name such as Common Name, for example, might contain the entry "Seagull" or "Red Ibis."

A database thus contains records; records have fields; the fields contain entries. These entries are made accessible by the Table, Form, and Report functions of the *Base* database. At the end of a school year, for example, a student would be able to use the word processor (*Writer*) to write a

relatively sophisticated paper, and merge into the paper some of the data collected in the database of Birds information. The student might well also add illustrations of the birds.

6.2 LET'S GET PRACTICAL

It will be useful for you at this stage to see an actual *Base* database.

> Make sure you have your **Work Files for LibreOffice 5** available on your computer (your USB drive is in one of the USB ports), then open **LibreOffice** > **Base Database**

Base presents you with the Database Wizard dialog box (Fig. 6.2).
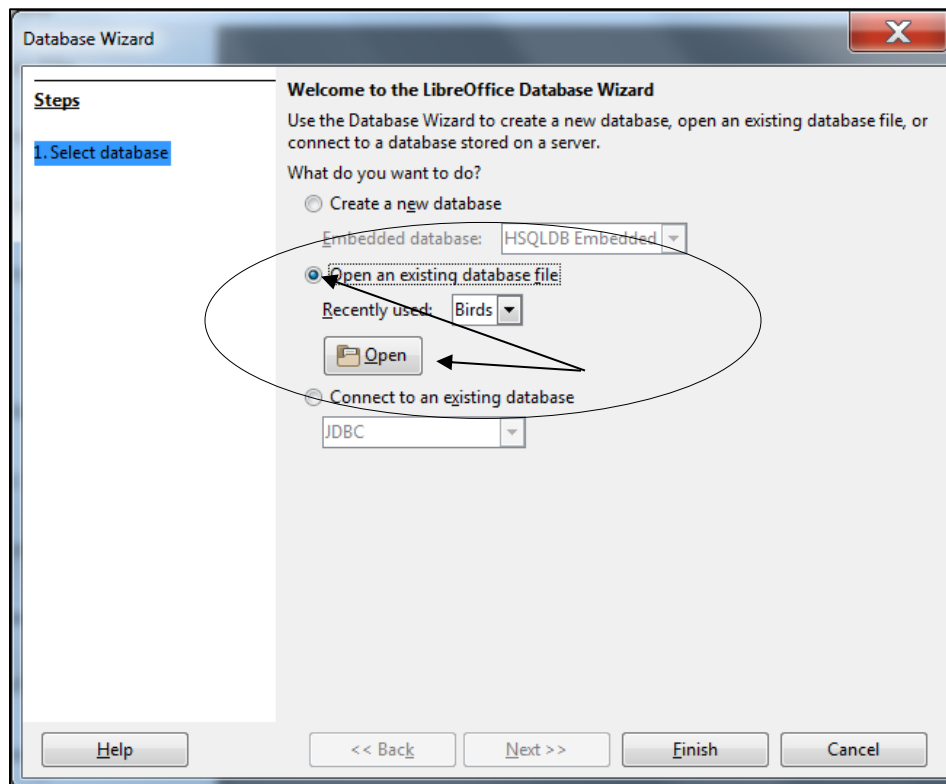

Fig. 6.2 The Database Wizard dialog box

This Wizard will step you through the process of either *creating* a new database or *opening* an existing database or even connecting to an existing database online.

Since you haven't opened any *Base* files recently, you are offered the opportunity to Create a new database. But instead you are going to open a database that has already been created for you to work with in this first section of the tutorial.

> So, click on the **radio button** next to **Open an existing database file**, then click on the **Open** button (Fig. 6.2 above)

This brings up the Open window.

> In the **Open window**, navigate to your **USB drive** > **Work Files for LibreOffice 5** > **Base Files** and **double click** on the **Birds** database to **Open** it

194

In the Birds database dialog box you see the icons for the *LibreOffice Base* database objects (Tables, Queries, Forms, and Reports), which are listed one of top of the other in the *Database* menu on the *left side* of the Birds database window (Fig. 6.3).
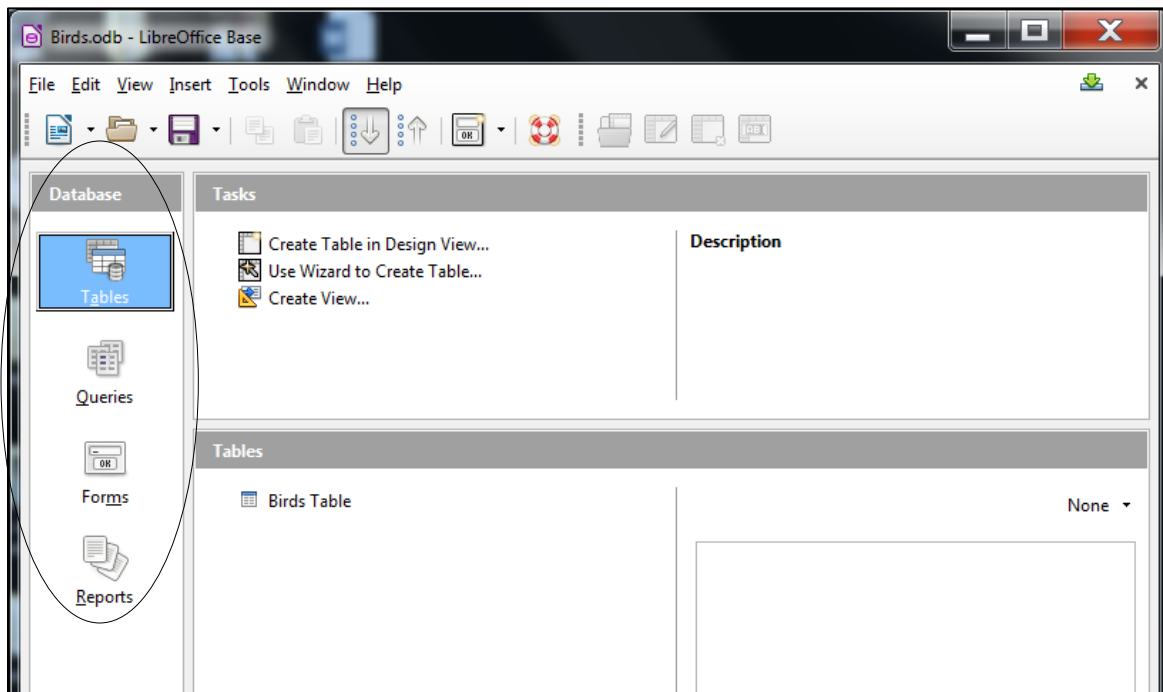

Fig. 6.3 How do you want to view the data in the database?

You can view the data as a *Table*, where the data for all the records in the database are displayed in columns or as a list (Fig. 6.4).

Fig. 6.4 The Table View of the database

In the upper left corner of the Table window, in the window's Title bar, you see the Table name—Bird Sightings Table (Fig. 6.4).

In the lower left of the Table window are tools for you to toggle through, and access, specific records in the database (Fig. 6.5).
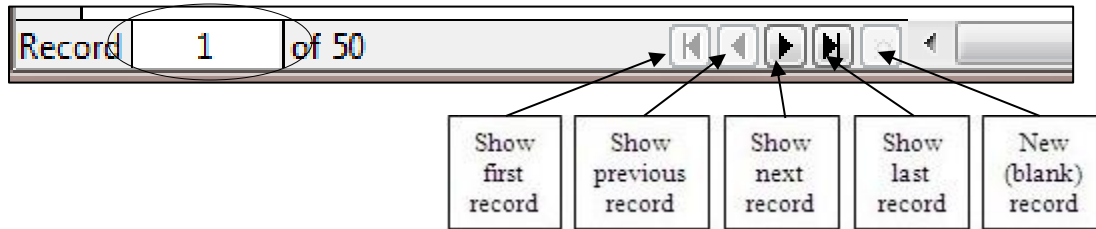

Fig. 6.5 Navigation tools in the *Base* database

There are 50 records in the *Birds* database. The field names (Common Name, Family, Size, etc.) are at the top of each column of data (Fig. 6.5). As you can see, the screen is not big enough to allow you simultaneously to see *all* the records in the database. If you want to scroll through the records or look at any specific record, you have several options.

You can use the scroll bar on the right of the screen to scroll through the records.

> Try this now

To move *one by one* through the records, you can click left or right on the arrows in the navigation tools in the lower left of the window (Fig. 6.5).

> Try this now

To go directly to a specific record, you can select the number of the record you want to view by typing the number of that record in the small data entry box to the left of the arrows (Fig. 6.5).

> Try this now to select record **12** (type the number **12** in the box then hit **Enter**) and you'll see a small green arrow head pointing at the **record #12** in the database table (it's a bird called a **Pintail**)

6.3 PLANNING A NEW DATABASE

So much for the nuts and bolts of databases. You are now ready to get your feet wet planning your own database of student records.

> Click on the Table's **File menu** and select **Close** to close the **Birds database Table**, then in the **Birds Database File menu** select **Close** again

It will be good for you to start from scratch for the next database exercise. Before you develop a new *Base* database, you need to have made up your mind where you are going to save the new database (which disk drive are you going to use?—your hard disk drive or an external drive such as a USB Flash drive or Thumb drive, for example. Make sure you know where you are going to save the new database before you proceed. If you are completing this exercise in a computer lab at your school, your instructor will advise you what to do. For the sake of this tutorial, we'll assume you have a USB drive on which are stored all the files you need.

Points to ponder when planning a database
There are three rules of database design:

Rule 1—Plan;

Rule 2—Plan;

Rule 3—Plan!

The more complex the data, the more you need to plan. But even the simplest database should be thought through on paper before being created in *Base*. Poor planning often results in a database that fails to meet longer term needs. So think about your database ahead of time—and PLAN!!

When planning a database, the rule of thumb that should guide you is this: it becomes increasingly difficult to make changes to anything the further along you go. If you think about it, this is true of anything you create.

Here, then, are some words of wisdom that you should bear in mind when designing a database.

- *If you take your time up front it will save time later on*
  The database you create will have a long, useful life if you take time to plan it carefully. After you have decided on the fields to include with each record, and *before* you create the database, you should still invest time designing layouts for reports. Thinking about reports will cause you to think about what data you plan to put in the database.
- *Teamwork helps*
  During the planning stage, run your ideas by others who are familiar with the kind of database you have in mind. Network among your colleagues and friends. Tell them what you have in mind. Ask them to review your design. You'll be surprised how many valuable ideas they'll come up with that may have escaped you if you had relied on your own resources. Another good idea is to involve your students in the design. This will help them learn skills that will benefit them throughout their lives.
- *Keep fields simple*
  The more "atomic" your fields the more flexible will be your database. Atomic here means "reduced to its simplest form." For example, in a database of names and addresses, you would keep each part of the person's names (first, middle, and last) as a separate field. Lumping the whole name under one field limits your options. The first name should be stored by itself; the same for the middle name and last name. You can print a listing last name first or first name last, with or without the middle name, and so on. By keeping the data *Atomic* you have a wider range of choices when working with the data in the database.

Design guidelines for a Student Roster database
You have to build a database for a Student Roster. Let us say that your planning has helped you decide the following about the database and its use:

- You have decided that *the database will be accessible to, and managed by, your students*. Each of them will enter their own data at the beginning of the year. You will advise them that they are *not* obliged to fill out every field—that it is OK to leave entries blank. Privacy is an important issue to which our students need to be sensitized. We need to take every opportunity to teach them that they should exercise control over data about themselves. They must make decisions about what is, and is not, privileged information. In a world where, inevitably and increasingly, personal data will be available to whoever wants to use them, our students must learn early on in their lives that they have a responsibility to keep tabs on their own personal data so as to ensure, as far as is humanly possible, that the data are correct at all times. Managing their own records on the class database will give them valuable experience in dealing with issues of privacy such as this.
- You have decided on *a list of fields for each record in the database*.
- You have decided that *all the fields will be treated as simple text*, except the Date of Birth field, which will be of *Date* type, and the Brothers and Sisters fields, which will be of *Number*

type. When you declare a field as *Date* type you can later sort the records in the database chronologically, which has its uses. Likewise, if you declare fields as a *Number* type you can later sort them numerically. You can also manipulate numeric data mathematically, creating totals, averages, and so forth.

- You have decided that *in your class you will designate one student per week as the database manager*. A schedule will be drawn up for this purpose at the beginning of the year. As far as possible, any interaction with the database, including the teacher's interaction, will be monitored by that student unless this is inappropriate for reasons of privacy. Other students may use the database only when the student manager is present.

- You have decided that *the database manager (one of the students in your class) will check the database for items of interest (birthdays, etc.) for the week he or she is in charge*, and will interact with the rest of the class to decide how to recognize such items of interest.

- You have decided that *the last activity on the database at the end of each week will be the process of making a backup of the database*. This will be necessary because changes to data can occur at any time (a student has a new brother or sister, for example). This would be done by the student database manager under the teacher's supervision.

You could add other features in an actual working environment. With all this in mind, you are ready to create the database for your student records.

6.4 CREATING THE DATABASE TEMPLATE: PART ONE

You are going to start a new database for Student Records, which initially will be a *template* for a database of student records because it won't contain any actual records right away. This way you'll be able to use this template database for future groups of students you may have.

Open **LibreOffice** > **Base Database** then, in the **Database Wizard** which pops up on the screen, make sure the **radio button** next to **Create a new database** is **selected** (Fig. 6.6) and click on **Next >>**
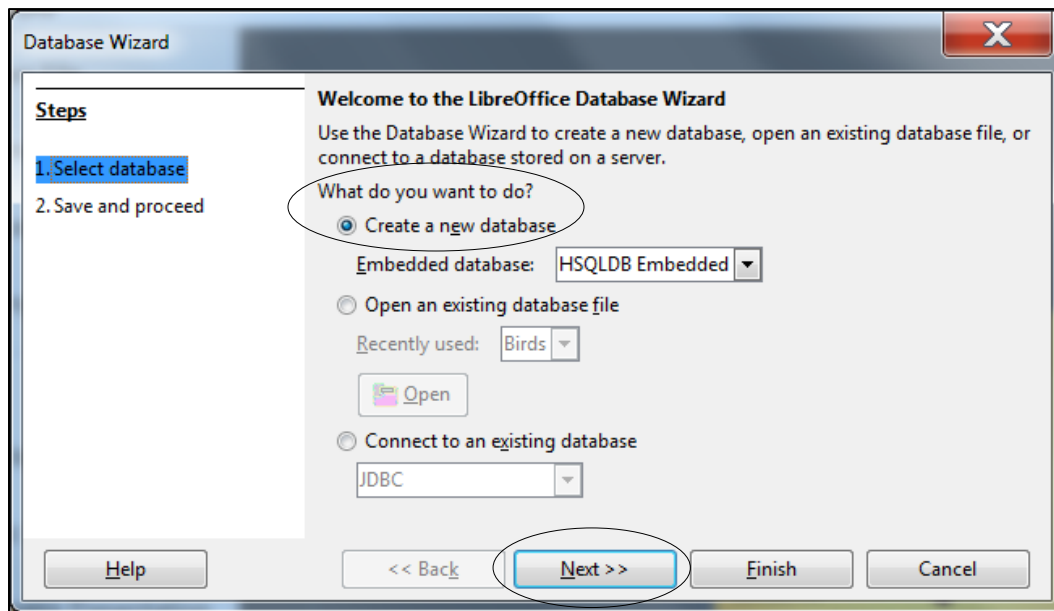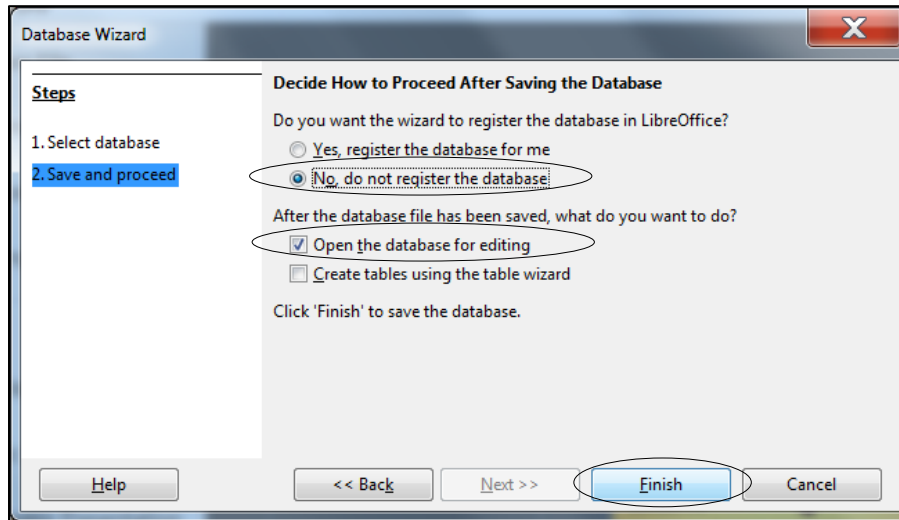

Fig. 6.6 *Base* Wizard to create a new database

There's no need to register the database, so click on the **radio button** next to **No, do not register the database**, leave the **check mark** in the box next to **Open the database for editing**, then click on the **Finish** button to save the new database

How to proceed after saving the database

In the **Save As dialog box** which pops up, navigate to your **USB drive** > **Work Files for LibreOffice 5** > **Data Files** and create a **New Folder** with the name **Base Documents**

Double click on the **Base Documents folder** to **Open** it, then, in the **File Name** data entry box, type the file name **Student Records Template** and click on **Save**

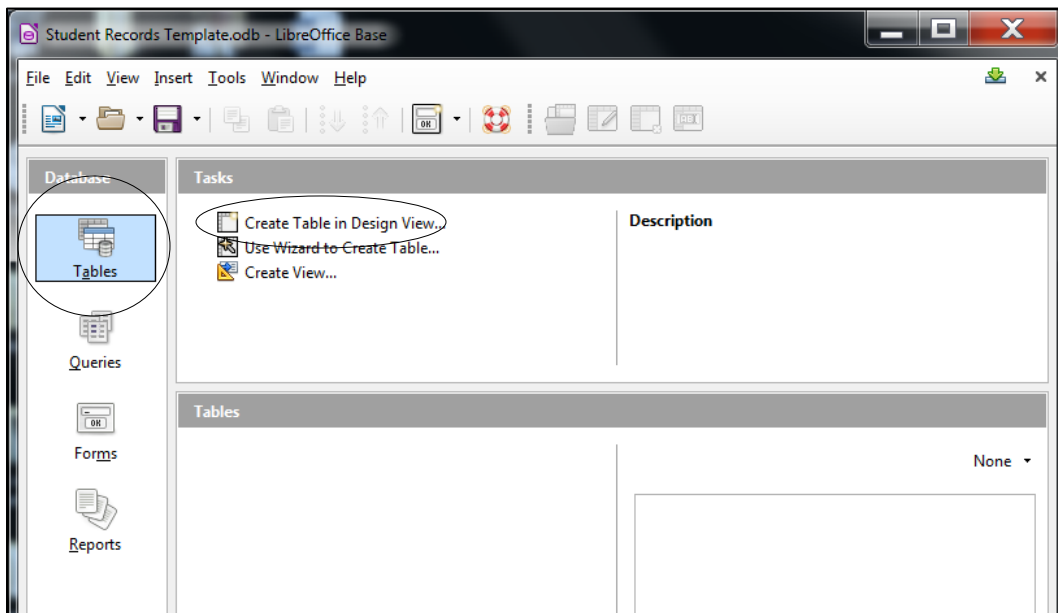*Base* now displays the newly created *Student Records Template* database (Fig. 6.6).

Fig. 6.6 The new Student Records Template database

Notice that the *Tables* object is selected (highlighted in blue) in the *Database* frame on the left side of the *Base* window. A Table is the default object in a new *Base* database. To the *immediate right* of the Database frame, in the *Task frame*, there are *three options*, the first of which is what we want.

> Click once on the **Create Table in Design View** option (Fig. 6.6 above)

Immediately you are presented with the Table Design dialog box (Fig. 6.7)

Fig. 6.7 Table Design dialog box, waiting for you to fill out Field entries

On the screen, *Base* is waiting for you to begin typing in the field names. The cursor is located in the first data entry box in the *Field Name* column, prompting you to type in a name for your first field. *Base* field names can be up to 64 characters in length, by the way, but ideally you want to keep them short and sweet, while still being clearly descriptive of the data that the field contains. Before you enter the field names, however, read what follows.

Editing field names Table 6.1 below provides you the complete list of all the *field names* for this *Student Records Template* database. Remember, you're building a template so that you can use it for other student record databases at a future date. You can easily change, add or delete fields at any time, though the purpose of planning is to try your best to avoid having to make significant changes after the database has been created.

Selecting the data type of a field As illustrated in Fig. 6.7 (previous page), after typing in the field name in the first column, you'll need to enter the *field's data type* in the second column. Table 6.1 below has all this information, too.

6.5 CREATING THE DATABASE TEMPLATE: PART TWO
Entering field names for the database
Table 6.1 contains the complete list of field names for the Student Records Template database.

Field Name	Field Type	Required	Length	Format Example
First Name	Text	Yes	15	
Middle Name	Text	Yes	15	
Last Name	Text	Yes	15	
Home Contact	Text	Yes	30	
Nick Name	Text	No	15	
Gender	Text	Yes	1	
Address Line 1	Text	Yes	50	
City	Text	Yes	25	
State	Text	Yes	2	
Postal Code	Text	Yes	10	
Phone Number	Text	No	12	
Email Address	Text	No	40	
Date of Birth	Text	Yes	-	January 1, 1900
Brothers	Number	Yes	2	
Sisters	Number	Yes	2	
Place in Family	Number	Yes	2	
Clubs	Memo	No	-	
Hobbies	Memo	No	-	
Favorite Sports	Memo	No	-	
Favorite Subjects	Memo	No	-	

Table 6.1 Template Field Name listing

You'll be referring to this Table frequently as you work your way through the next several pages, so you might like to bookmark Table 6.1 in order to find the Table when you need to.

Type only the Field Names that appear in the <u>first column</u> of Table 6.1 (there are 20 field names in all). The second column tells you what Data Type to choose (from a drop down list that pops up when you click on the down arrow at the right side of the entry box.

The other columns in the table (Required, Length, and Format Example columns) are there to help you as you enter each field name and data type and explain whether data is to be "Required" for that field and what size (in text characters) and format the field is allowed to be.

Click in the **first** entry box in the **Field Name** column, then type **First Name** for the field name (Fig. 6.8)
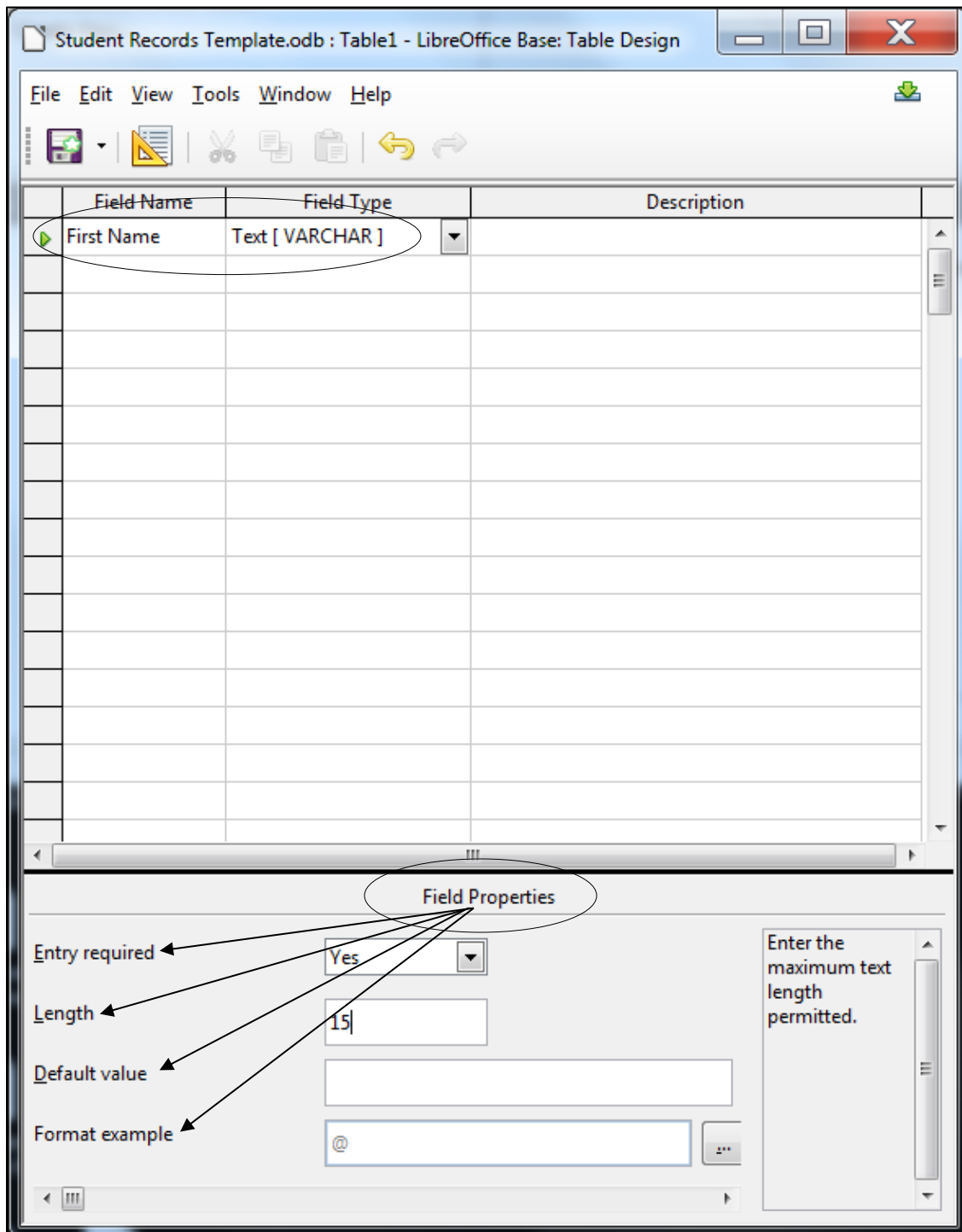

Fig. 6.8 Field Properties

Notice that, in the <u>lower right</u> section of the Student Records Table window, *Base* provides help to guide you as you enter the data for each part of each field. Read this help as you go along. In the <u>lower left</u> you can see the *Field Properties* dialog area (Fig. 6.8 above).

If you take a look at Table 6.1 on page 201, the *Length* column specifies the number of characters required for several of the fields in the database.

Check **Table 6.1** to find out (a) whether an entry is **required** for the **First Name** field (Yes) and (b) the **correct size** for the **First Name** field (15 characters)

Now, look down at the **lower left corner** of the Table Design window (Fig. 6.8 again), where you can see a list of **Field Properties**, along with data entry boxes to specify whether an entry is **required**, the entry's **Length**, **Format example**, and so forth

In the **Entry Required** box select **Yes**, and in the **Length** box type **15** (so as to allow up to 15 characters for the **First Name**)

Click again in the **Field Name** > **First Name** box and hit the **Tab** key to move the cursor over to the **Field Type** column, click on the **down arrow** to the **right** of the box and select **Text [VARCHAR]** as the data type if it is not already selected

Now click in the **second Field Name box** (below the **First Name** box) so you can enter the next database field name

Working down the columns, enter the field name **Middle Name** (**Required** and **15** characters), **Last Name** (**Required** and **15** characters), **Home Contact** (**Required** and **30** characters), in each case making sure you have **Yes** in the **Entry Required** Field Property for each of these fields

Now enter the field name **Nick Name** (**15** characters), but leave **No** in the **Required** Field Property box, since a Nick Name is **optional**

Next enter the following field names, in each case making sure you have **Yes** in the **Entry Required** Field Property for each of the fields:

Gender (M/F) (**1** character)

Address Line 1 (**50** characters)

City (**25** characters)

State (**2** characters)

Postal Code (**10** characters),

Now enter the field name **Phone Number** (Not Required and **12** characters), and **Email Address** (Not Required and **40** characters) leaving **No** in the **Required** Field Property box in case a student's family does not have a phone or email address

Using a Format example to control the format for a field
You want to make sure that certain fields have a consistent format, such as the Date of Birth field, which is next in the listing in Table 6.1 on page 188, you want to select *[DATE]* as the data type for the field, and select a **Format example** for it, too.
Here's how you do this.

Enter the next field name **Date of Birth**, then in the **Field Type column** click on the **arrow** to bring down the **menu of Field Types**

Select the Field Type **Date [DATE]** then, in the **Field Properties area**, click on
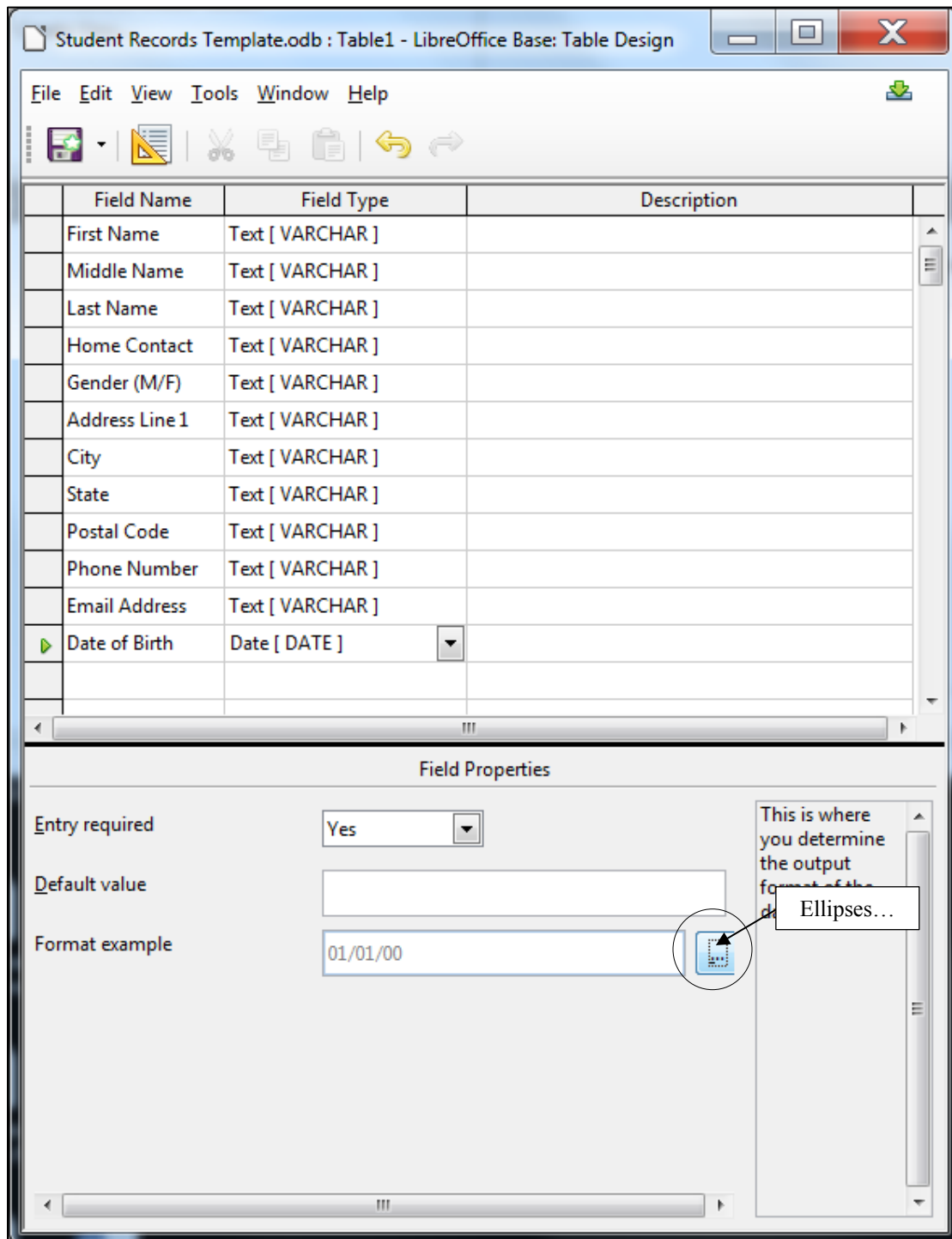the **ellipses** at the end of the **Format example** option (Fig. 6.9)

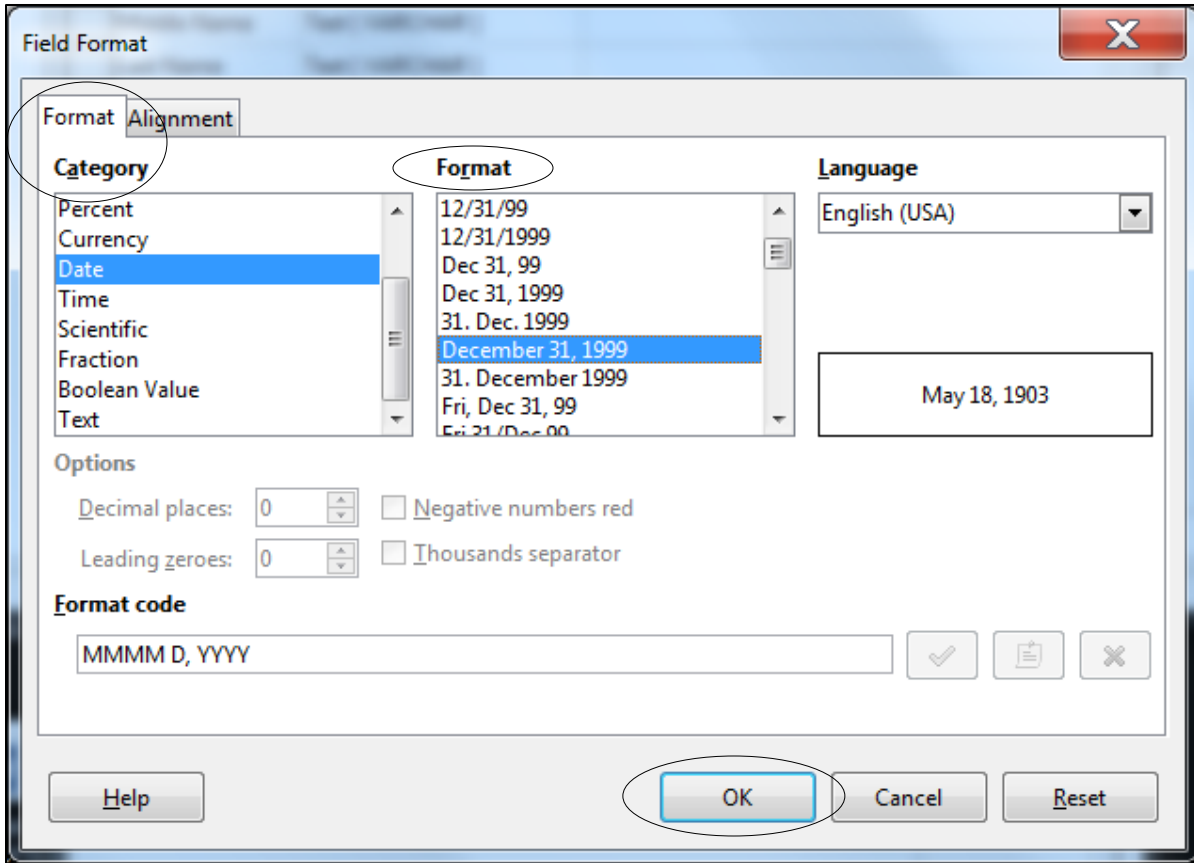Fig. 6.9 Click on the ellipses to specify a Format example for the Date of Birth field

Fig. 6.10 Selecting a Format example in the Field Format dialog box

In the **Field Format** > **Format tab** > **Category** select **Date**, and under **Format** select the option (**December 31, 1999**), then for each of the **next three database field names** (Brothers, Sisters, Place in Family) select the Field type **Number [NUMERIC]** and select **NO** for **Entry required**

The remaining fields in the database (Clubs, Hobbies, Favorite Sports, and Favorite Subjects) will all be *Memo* data type to allow for a longer entry if desired. Notice, in the Table on page 188, that data entries for these memo fields are optional, so *No* will be the answer to *Entry required*.

For each of the **last four database field names** (Clubs, Hobbies, Favorite Sports, and Favorite Subjects) select the Field type **Memo [LONGVARCHAR]** and select **NO** for **Entry required**, then select **File** > **Save As…** (Fig. 6.11), in the **Save As dialog box** type **Student Records Table** and click on **OK**
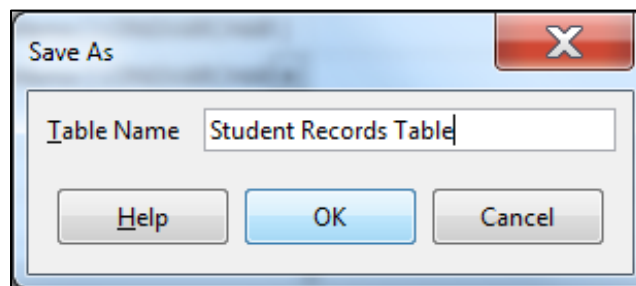

Fig. 6.11 Naming the Table

Whenever you attempt to save a *new LibreOffice Base* table for the first time, you are asked to define a *Primary Key* for the Table (Fig. 6.12).
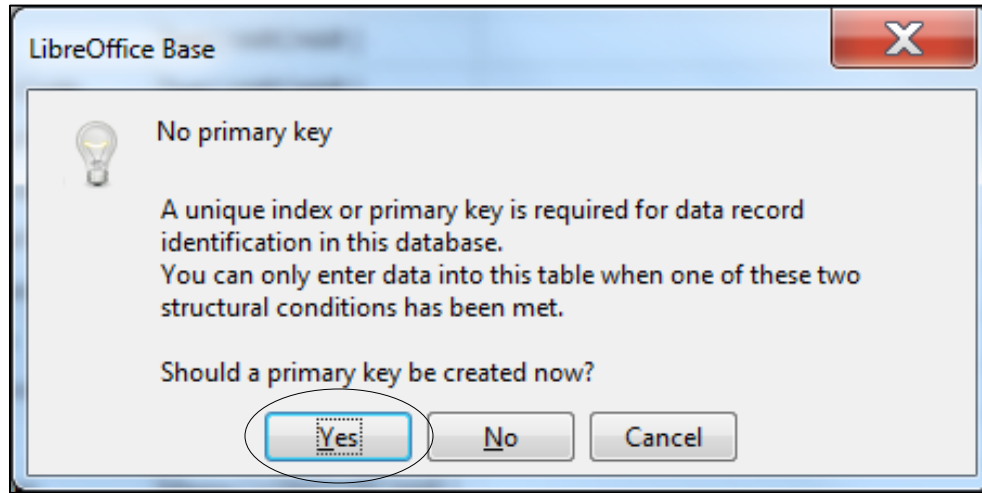

Fig. 6.12 Prompt to define a Primary Key

So what is a Primary Key in a database? The purpose of a Primary Key is to enable easy manipulation of the data across multiple Database Tables.

For example, you were designing a database for an administrative application in the Principal's or Superintendent's office, or for a sizable company, this would be an important consideration, because your database would be large, and there would be many ways of looking at the data in the database (called "views" on the data) represented by different Tables.

The Primary Key field is designed to allow relationships between any two Tables. While this does not concern us in these "*Essential*" LibreOffice Tutorials, we'll keep the Primary Key in case you decide at some future date to expand the Student Records database for your own purposes.

> Click on **Yes** to tell *Base* you *do* **need a Primary Key field** in the database,
> then go to **File** > **Close** to close the **Table Design window**

6.6 COMPLETING PREPARATION OF THE DATABASE TEMPLATE

Creating a Form for the new database

The following exercises will help you learn how to prepare a database Form.

A Form is a database object which allows you to view the data one record at a time, unlike a Table where you can view the data as a continuous list of records—recall how the Table looked in the Birds database you checked out at the beginning of this tutorial.

A Form is a convenient tool for data entry purposes because it allows the user to see all the Field entries *in a single record* at one time, unlike the Table view, which will often be unable to show all the data fields of a single record across the screen.

Let's create a Form for our Student Records Template database now.

> In the **Database Objects pane** on the **left** of the *Base* window, click on the
> **Forms** object, then click on **Use Wizard to Create Form…**

206

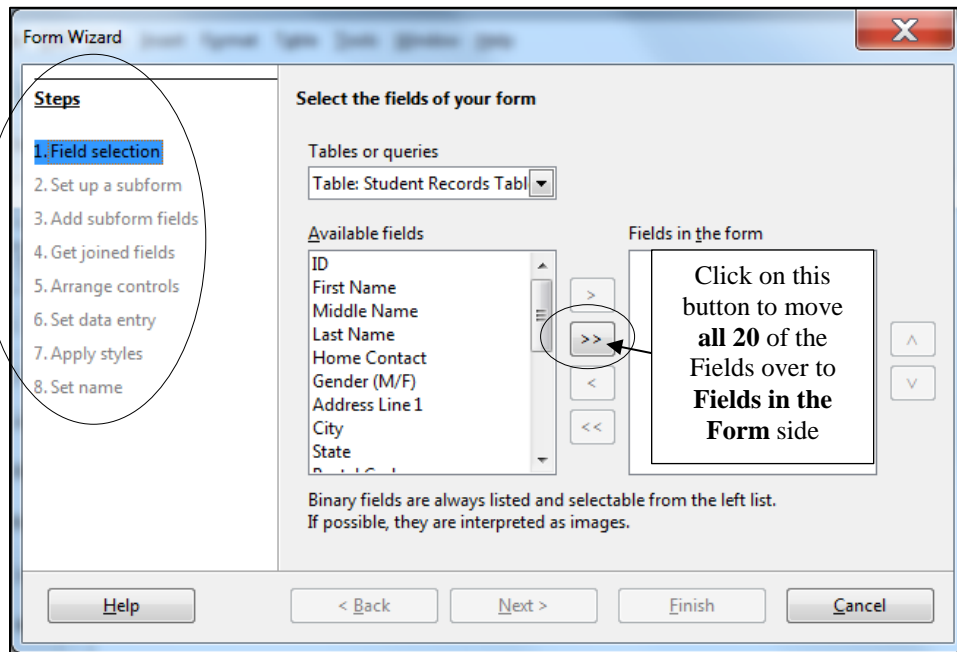You should now see the *Form Wizard* dialog box (Fig. 6.13)

Fig. 6.13 Step 1 in the Form Wizard dialog box

As you see in Fig. 6.13 above, the *Form Wizard* steps you through the 8-step process of creating a Form for the *Student Records Template* database. Not all of the 8 Steps are relevant for our purposes, but most of them are.

In **Step 1. Field Selection**, in the "**Select the fields of your form**" frame (Fig. 6.14), click on the **double arrow (>>)** symbol to move **all 20** of the fields in the **Student Records Template** over to the **Fields in the Form** box (Fig. 6.14)
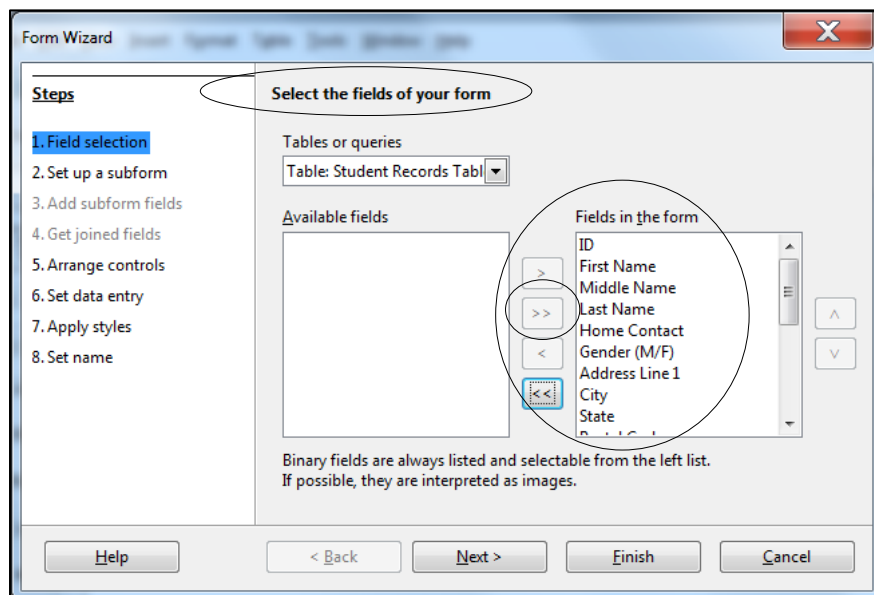

Fig. 6.14 The result of Step 1 in the Form Wizard

Step 2—**Set up a Subform**—is irrelevant to our purposes in this tutorial, as are also Steps 3 and 4, so just click on **Next** and **Next** to move on to **Step 5**, where you select the **design controls** for your database **Form** (Fig. 6.15)
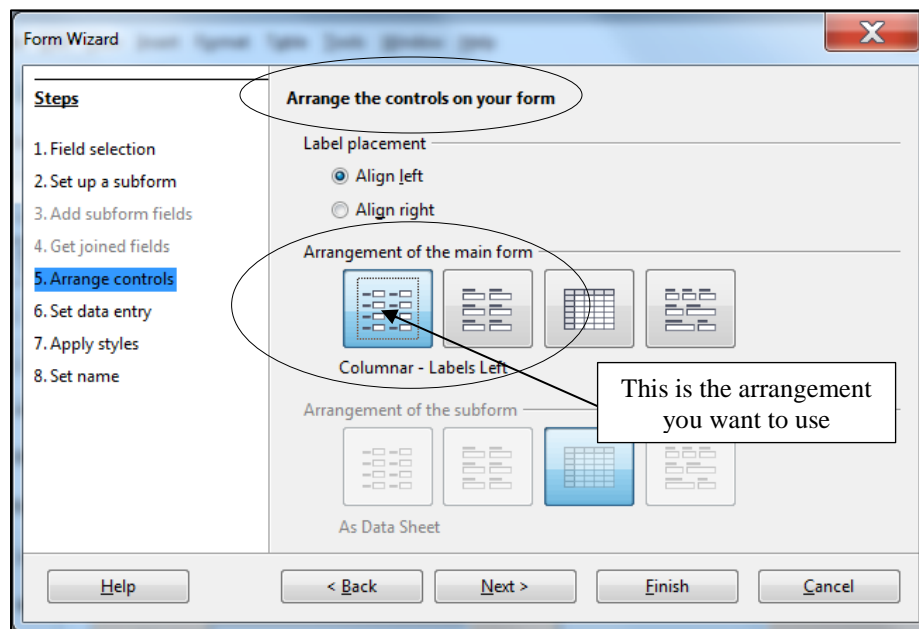

Fig. 6.15 Selecting the Arrangement of the main form

In **Step 5**—**Arrange controls**, select the **first** of the four options for **Arrangement of the main form** (i.e. **Columnar – Labels Left**—see Fig. 6.15 previous page), then click on **Next**

In **Step 6**—**Set data entry**, accept the **default data entry**—**The Form is to Display all data**—and click on **Next** to go to **Step 7**—**Apply styles**, choose any **color and Field border** you like for the **Form Style (**Fig. 6.16), then click on **Next**
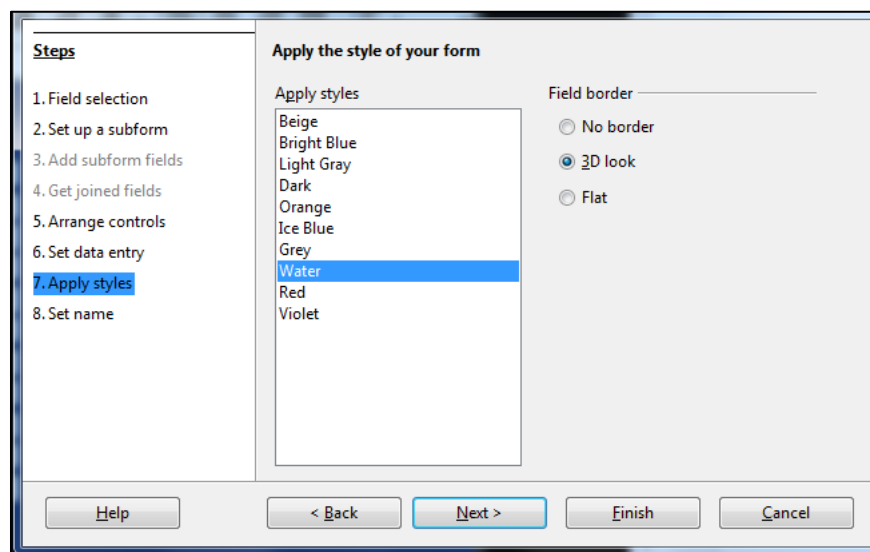

Fig. 6.16 Selecting the Style for the Form background

All that remains—**Step 8**—is to **name** the Form (**Student Records Form**) and make sure the **radio button** next to **Work with the form** is selected, then click on **Finish** (Fig. 6.17)

Fig. 6.17 Naming the database Form

Base finally presents the Student Records Database window, ready for you to start filling out the individual records with student data (Fig. 6.18)

Fig. 6.18 The final look of the Student Records Database form

Go to **File** > **Close** to close the **Student Records Form**

If you wanted to reorganize the form's design, as illustrated above (Fig. 6.18), you would *right click*, at the top level of the Fall Roster 2016 database, on Student Record Form (see Fig. 10.19 next page) and, in the context menu, select the option to *Edit*. It's tricky, but the Edit mode allows you to move fields around on the form and resize them. This is left as an optional exercise.

6.7 ADDING RECORDS TO A NEW OR EXISTING DATABASE

Creating a new database based on a Template database

The Student Records Template is so named because you want to use this same template document (with its table *Student Records Table*) and data entry form (*Student Records Form*) over and over for class records in the years ahead.

Before you start adding data about this year's students, you need to create a new database using this template as the model. You will use the copy of the template to store actual student data for a hypothetical class (or a real class, if you are a practicing teacher). The template will thus remain unchanged for future use as the basis for other class databases.

How you name documents is important. For the sake of this exercise, you're going to call the new database *Fall Roster 2016* (or some other appropriate name if you prefer). The name indicates that the document will contain the database of student records for a Fall 2016 class. If you are currently teaching, you can use a different name if you want to use this tutorial to build an actual database for your own class. Substitute the name you choose wherever we refer to *Fall Roster 2016* as you go along.

The Template database should still be displayed on your screen. All you need to do is *Save* the template with the new name Fall Roster 2016 and you're ready to add the data for the new class of students.

> In the **File menu** select **Save As…** and, in the **Save As dialog box**, type the
> file name **Fall Roster 2016** and click on the **Save** button

Entering the data for the records in the database document

> The **Fall Roster 2016** database should be open on your screen, so **click** in the
> **left hand frame** on **Forms** to make sure the **Forms object** is selected then, in
> the **Forms frame**, **double click** on **Student Records Form** to open it (Fig.
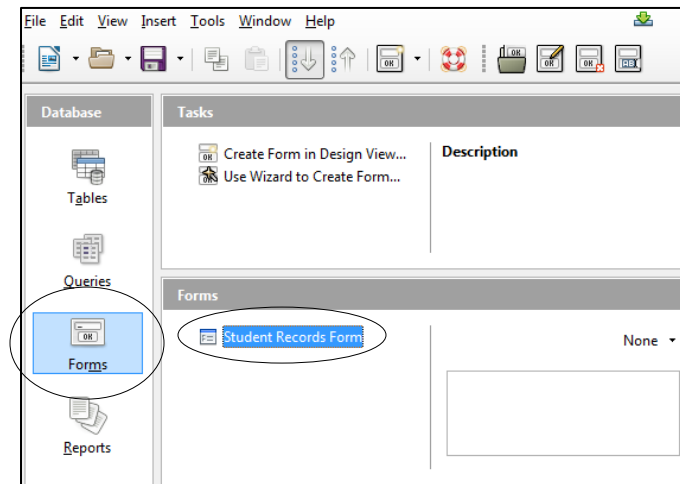> 10.19)

Fig. 10.19 Fall Roster 2016 database

Each record in the Fall Roster 2016 database is made up of the data for a full set of 20 entries for each student. If you are a practicing teacher, you can use the data from one of your classes. If you are a student teacher, you may or may not have the opportunity to complete this exercise with a real class. Whatever the case, you will need to enter a few records at this stage (*five* will do nicely for now) so that you can complete the rest of this tutorial.

> Begin by selecting the **ID field** for **record #1** and enter the number **1** as the
> **Primary Key** for the record (the next record will have the Primary Key of **2**, and
> so on—the **Primary Key field must be unique**)

> Hit the **Tab key** to move to the next field's data entry box—the **First Name** field of **record #1** and type in the **First Name** of one of the students in your roster

The names don't have to be in any particular order since you can quickly sort them later on. Check the name you just typed. Is it correct?

> If so, hit the **Tab** key again to proceed to the next field (you'd press **Shift-Tab** if you wanted to move to a previous field)

Notice that the data are now stored in the data entry box for the First Name field in Record #1. If you wanted to correct the data, you would do so by clicking in the data entry box and then make any changes as if you were in the word processor.

> Continue entering data for **Record #1** (the data for the first student in the roster); make up the data as you go along

Adding a New Record to the database

> After you type the **last field entry** for each record, press **Tab**, and *Base* will automatically present a **new record Form**

Base will also automatically save each completed record into the database on your disk, so you don't have to worry about losing your work; you also won't need to save the database form.

> Type the entries into each field one by one, starting with the **ID field** which will contain the next number in sequence—**2**, if it's the **second record** and so on; be sure to press the **Tab** key after each entry to get to the next field (eventually you should have **at least 5 records** in the **Fall Roster 2016** database)

Adding a record to an existing database is just the same. New records are always added at the end of the database. If you want the new record to appear in a specific place in the Datasheet view you would use the Sort function. You'll learn how to do this in Lesson 7.

The new record will always have the same layout and format characteristics as the others in the database. Also, remember that *Base* automatically saves a new record in the database after you have filled in the data for every entry.

6.8 VIEWING THE DATA IN THE DATABASE

Here are some of the ways to move about in a database. *Base* makes it easy for you to get around. Take a few moments to move from field to field and from record to record. Try these exercises.

> Click anywhere in the **data box** next to a **field name**

This allows you to edit or add data if needed.

> Hit the **Tab** key to move to the **next field** in the **same record**

> To move to the **same field** in the **next** or **previous** *record*, click on the **Navigation tools** at the bottom of the Form window (Fig. 6.19)
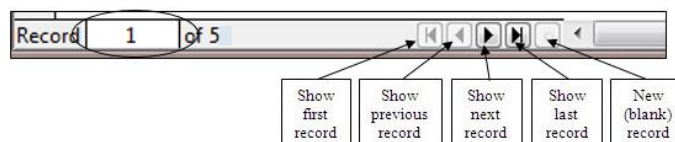

Fig. 6.19 The *Base* database navigation tools

In Lesson 7 you will learn other ways of searching through the database, along with other ways of selecting sets of records from the database. As a final exercise in Lesson 7, let's learn how to update the data in a Base database.

6.9 UPDATING ENTRIES AND RECORDS IN THE DATABASE

Updating the contents of a field (one item of data in a student record)

> You should still be looking at the **Student Records Form**

To delete, or otherwise change, the contents of a field in a record, click on the field data entry box, then erase characters or words in the same way as you would if you were in the word processor.

Deleting (clearing) a record

In order to delete a single record from a database, you need the Form Navigation toolbar, which should already be open at the bottom of the database Form, along with the other toolbars relevant to a *Base* database Form (Fig. 6.20).
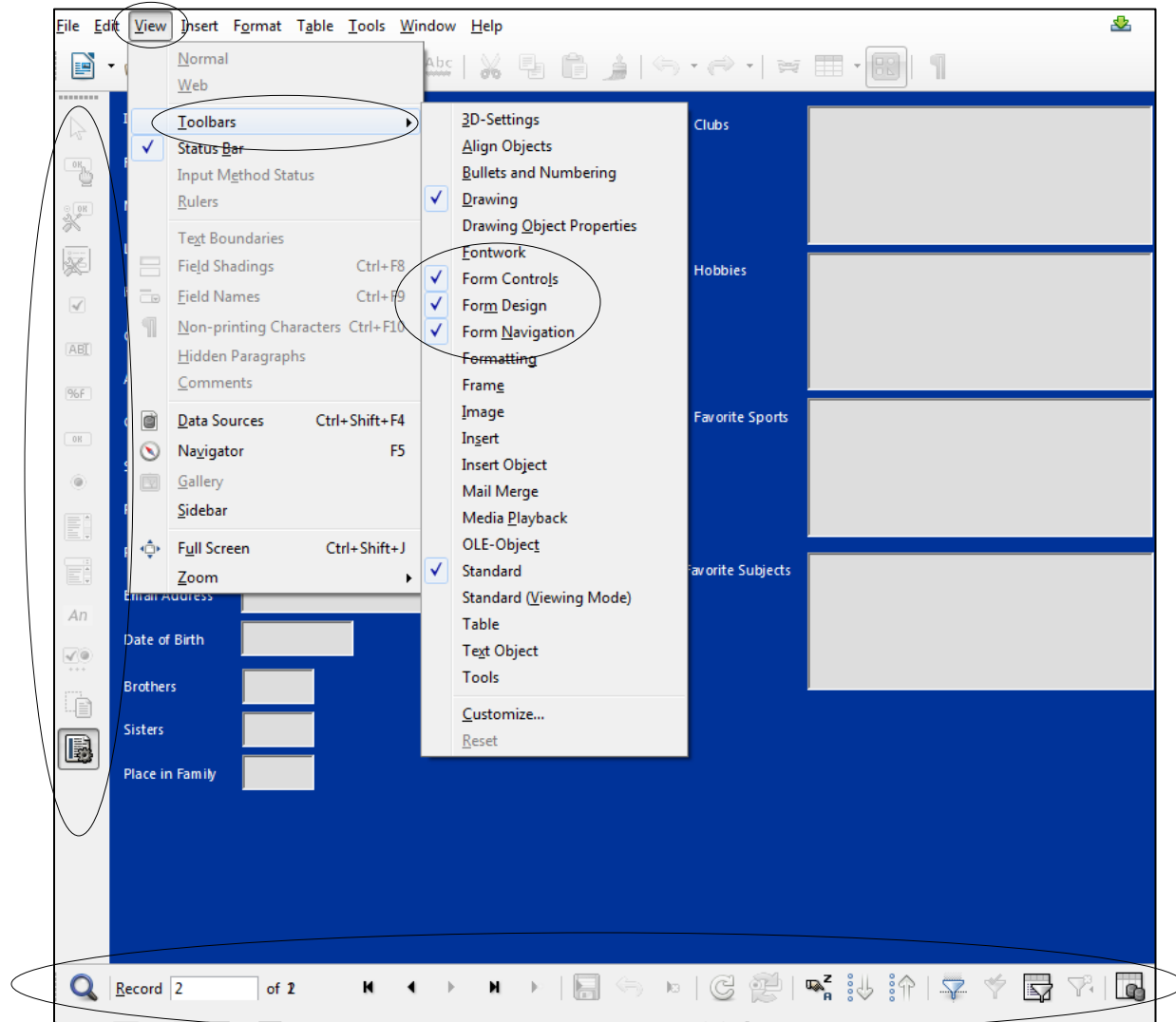

Fig. 6.20 The Form toolbars

If these toolbars are not available in the Fall Roster 2016 database Form, go to the **View menu** > **Toolbars** and make sure you select each of the **Form toolbars** in turn—the **Form Controls** toolbar, the **Form Design** toolbar, and the **Form Navigation** toolbar (click to put a **check mark** next to each of them, as illustrated in Fig. 6.20 on the previous page)

Now try this to practice deleting a record from the database.

In the **Navigation toolbar** (the upper of the two toolbars at the bottom of the Form window), click on the **record control tools** to select **Record #2** in the **Fall Roster 2016** database (Fig. 6.21)
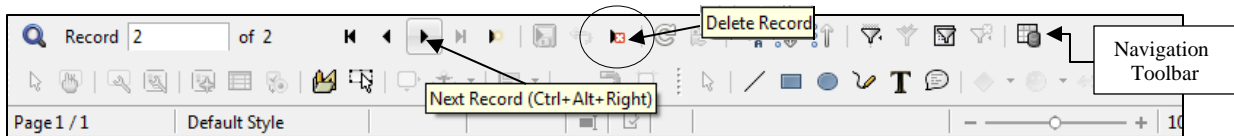

Fig. 6.21 Selecting a specific record in the database

In the **Navigation toolbar**, click on the **Delete Record** tool

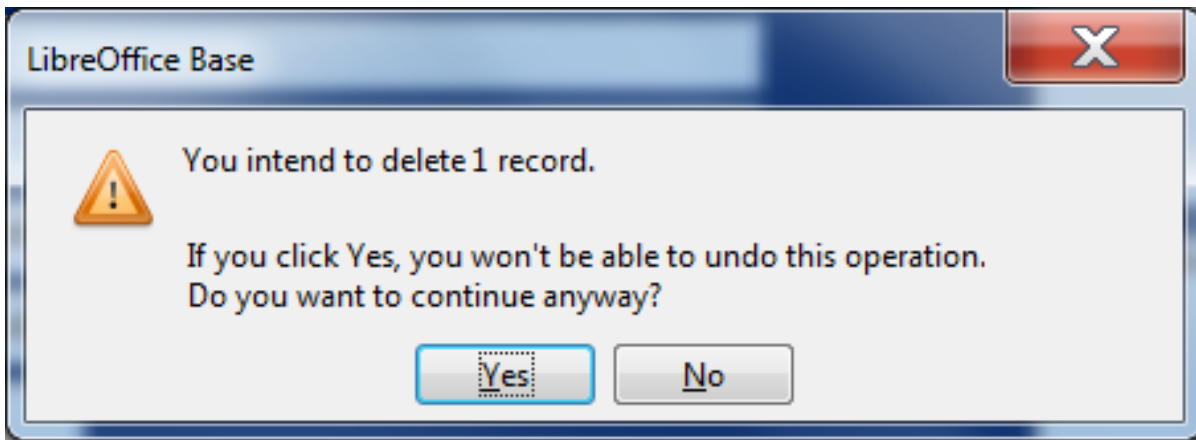*Base* will warn you that you are about to delete a record (Fig. 6.22).

Fig. 6.22 Alert to inform you that deleting a record is undoable

This gives you the opportunity to change your mind, which you need to do in this case. *Base* will otherwise immediately permanently remove the record if you click on *Yes*.
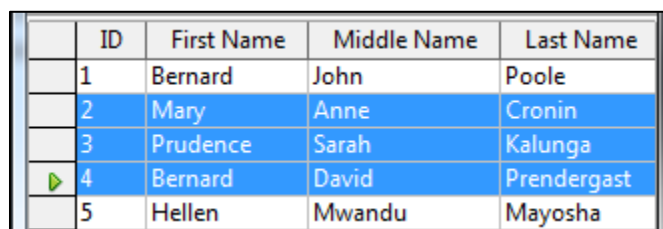
Click on **No** in the dialog box

Deleting several records at once
It's easiest to use the Table View to clear several records that are listed next to each other. To delete a block of records at one time here's what you do.

In the **File menu** select **Close** to close the Fall Roster 2016 database Form, then, in the **Database frame** on the **left hand side** of the Fall Roster 2016 window, click on the **Tables icon**, and then **double click** in the Tables section on **Student Records Table**

Now, in the **left hand margin** of the Table, click to select **Record #2**, then hold down the **Shift key** while you click in the left hand margin to select **Record #4**

Records 2, 3, and 4 should now be highlighted (Fig. 6.23)

	ID	First Name	Middle Name	Last Name
	1	Bernard	John	Poole
	2	Mary	Anne	Cronin
	3	Prudence	Sarah	Kalunga
▷	4	Bernard	David	Prendergast
	5	Hellen	Mwandu	Mayosha

Fig. 6.23 Selecting several records for deletion

> In the **Navigation toolbar**, click on the **Delete Record** tool

Again you will get a warning (see Fig. 6.22 on previous page) that you are about to delete however many records you have selected, and will ask you if you are sure you want to do this.

> You don't, so click on **No** in the dialog box

Bear in mind that the Delete Record command completely removes a record, such that it is irretrievable once deleted.

6.10 SAVING A BACKUP COPY OF YOUR WORK

Your last task before completing this session at the computer is to make a backup of your Student Records Template and Fall Roster 2016 databases. The Fall Roster 2016 database is still on the Desktop. Both databases are also saved in the *Work Files for LibreOffice 5* on your disk, which should still be in the disk drive.

> Go to **File** > **Save As…**, then, on your **USB drive**, navigate to your **Work Files for LibreOffice 5** folder, **double click** to open the **Data Files** folder, then **double click** to open the **Database Documents** folder

> Now, in the **Save As** dialog box, **rename** the **Fall Roster 2016** file by **adding** the word **Backup** to the file name (so the backup copy of the database will be named **Fall Roster 2016 Backup**)

> **Close** the **Fall Roster 2016 Backup** database, then **Open** the **Student Records Template** database, again go to **File** > **Save As…**, then, in the **Save As** dialog box, **rename** the **Student Records Template** by **adding** the word **Backup** to the file name (so the backup copy of the database will be named **Student Records Template Backup**)

It is always advisable to make backup copies of all the documents that you create.

LOOKING BACK

You have learned how to build a Base database. The concepts that have been covered in this tutorial apply in general to all computer-based databases. You have to plan first. Then you have to define, or set up, the database, telling the system what fields you need in each record. This done, you enter the data for each record and save everything on the disk.

You practiced scanning the database in multiple (Datasheet or columnar) or single (Form) record layout. You also learned how to clear records and change entries.

LOOKING FORWARD

There is a great deal more to learn about the *Base* database, especially with regard to searching for specific sets of data, sorting (arranging) data, and producing various types of reports. Lesson 7 will provide you with the opportunity to practice these skills. In the meantime, in Lesson 7 you will learn how to do Mail Merge. This is an especially useful skill for teachers, since you will often want to "mass mail" correspondence to the parents of your students. Mail merge is also useful for other types of projects, such as forms of various kinds which need to be tailored for individual needs. Mail merge involves the use of the *Base* database, too; so now that you are familiar with *Base*, you are ready to tackle Mail merge.

SKILL CONSOLIDATION

Complete these exercises to reinforce what you have learned in Lesson 6.

1. Design and create a database for the names, addresses, and telephone numbers you have in your address book. Enter as many names and addresses as you like (as long as you have at least 6). Don't forget to save the database records as you go along.

2. Add a couple of records to the database you created in exercise 1. Clear a record.

3. Add a field to the address book database—say Association (how are the people associated to you—friend, colleague, etc.). Then enter the data for each record for the new field.

4. Imagine you are teaching a class and you want your students to do research on a topic. Design and create a database around that topic. You could discuss the fields with your students. The students would then go off, do the research, and enter the data into the database as a basis for a paper they must prepare. This would be a nice group (team) exercise.

5. Design and create a database for a Birthday List of your family and friends. Include fields such as Name, Birthday, Greek sign, Favorite color, Clothes sizes, Hobby, etc. Have at least 6 records in the database, with each record containing at least 5 fields. Create a Form for the database as well as a Table.

6. Open the *Base* database Fall Roster 2016 and enter a complete new record for a student named Jane Doe.

7. Open the *Base* database named Template. Change the name to Grade X.2016 so you won't lose the template, and add three (3) records. Then save the updated database.