

Apache OpenOffice

Version 4.1

Getting Started

AOO Documentation Team

Copyright

This document is Copyright © 2023 by The Apache Software Foundation. You may distribute it and/or modify it under the terms of the Creative Commons Attribution License, version 3.0 or later (<https://creativecommons.org/licenses/by/3.0/>).

Apache, Apache OpenOffice, and OpenOffice.org are trademarks of the Apache Software Foundation. Used with permission. .

Acknowledgments

This book is updated from *Taming Apache OpenOffice 3.4* (by Jean Hollis Weber), which was adapted from *Getting Started with OpenOffice.org 3.3* (by the OOoAuthors team) with additional material adapted from *Getting Started with LibreOffice 3.4* (by the ODF Authors Team).

Publication date and software version

Published 27 April 2023. Based on Apache OpenOffice 4.1.

Contents

Chapter 7	
Getting Started with Base.....	4
What Is Base?.....	4
Planning a Database.....	5
Creating a New Database.....	6
Creating Database Tables.....	8
Using the Wizard to Create a Table.....	8
Creating a Table by Copying an Existing Table.....	11
Creating Tables in Design View.....	12
Creating Auxiliary Tables for Data Entry.....	15
Adding Data to the List Table.....	15
Defining Relationships.....	15
Creating a Database Form.....	19
Using the Wizard to Create a Simple Form.....	20
Using the Wizard to Make a Form With a Subform.....	22
Laying Out a Form With Multiple Controls.....	29
Creating Forms and Subforms in Design View.....	38
Accessing Other Data Sources.....	38
Accessing a Spreadsheet as a Data Source.....	38
Registering *.odb Databases.....	39
Using Data Sources in Apache OpenOffice.....	39
Viewing Data Sources.....	39
Editing Data Sources.....	40
Launching Base to Work on Data Sources.....	40
Using Data Sources in Writer and Calc.....	40
Creating Queries.....	45
Using the Wizard to Create a Query.....	46
Using the Design View to Create a Query.....	48
Creating Reports.....	53
Simple Static Report.....	54
Grouped Dynamic Report.....	56

Chapter 7

Getting Started with Base

What Is Base?

Base is the component of Apache OpenOffice that allows the office suite to interact with a wide variety of data sources. For example, a data source could be a list of names and addresses used for producing a mail merge letter. A shop stock list could be a data source managed through AOO.

Note Apache OpenOffice uses the terms “Data Source” and “Database” to refer to the same thing, which could be a database such as MySQL or dBase or a spreadsheet or text document holding data.

This chapter covers creating a database, viewing and entering data, and loading the data into OpenOffice documents. It also covers using the Base component of AOO to register other data sources. A data source can be a database, spreadsheet, or text document.

Note Base is not a database engine. OpenOffice includes the HSQL database engine in the installation files, which is used by default to manipulate data. All the files created by this engine in the default configuration, called an embedded database, are kept within the Base file. It is possible, and sometimes preferable, to connect Base to an external database engine.

Databases bear a superficial similarity to the more widely encountered spreadsheets, but there are important differences. A *database* consists of a number of columns or *fields* that contain the individual pieces of data. Groups of *fields* make up a *table* of the database. When creating a table, you also determine the characteristics of each field in the table. This is one key difference from spreadsheets. Each field or column may contain only one kind of data. Within the broad data types of text and numbers,

there may be further restrictions. For example, the text might have to contain a certain number of characters or numbers might be restricted to be only integers. These restrictions are applied, if they are desired, when the table is defined. *Forms* are used to interact with the data tables, either for data entry or viewing. This is another difference from spreadsheets where you routinely directly view and manipulate the data tables. *Forms* format the data display and enable convenient navigation through the rows or *records* of a table. Database tables do not have the formatting or interactive tools to support convenient direct viewing, but they hold much more data and allow more flexible handling of lists. A *query* creates a new temporary table from the existing tables according to instructions you provide. The query may sort, order, filter, summarize and match data from many tables. A *report* organizes the information of the fields of a query in a document according to your requirements.

Caution

The database engine in AOO requires a Java Runtime Environment (JRE). If you do not have one on your computer, you can download it from www.java.com or adoptopenjdk.net and install it following the instructions on the site. In AOO, use **Tools > Options > OpenOffice > Java** to register Java.

Base creates *relational databases*. This makes it fairly easy to create a database in which the fields of the database have relationships with each other.

For example: Consider a database for a library. It will contain a field for the names of the authors and another field for the names of the books. There is an obvious relationship between the authors and the books they have written. The library may contain more than one book by the same author. This is what is known as a one-to-many relationship: one author and more than one book. Most if not all the relationships in such a database are one-to-many relationships.

Consider an employment database for the same library. One of the fields contains the names of the employees while others contain the social security numbers, and other personal data. The relationship between the names and social security numbers is one-to-one: only one social security number for each name.

If you are acquainted with mathematical sets, a relational database can easily be explained in terms of sets: elements, subsets, unions, and intersections. The fields of a database are the elements. The tables are subsets. Relationships are defined in terms of unions and intersections of the subsets (tables).

To explain how to use a database, we will create one for the sales and expenses of a small group of people. It is not intended to be realistic for any purpose; it will merely supply concrete but highly simplified examples of data. In the process, we will be explaining how a database works.

Planning a Database

Planning a database's structure is a very important step which should not be skipped or minimized. A well planned database can answer all the questions you know you will want to ask and can be adapted easily to changing needs. A poorly planned database will likely be hard to use and may have to be restructured for even a modest change in the data or in the questions to be answered.

The first planning step is to define the purpose and use of the database as carefully as possible. What information do you want to get out of the database and what

information has to go in to achieve that? Be sure to consider not only the immediate questions you want to answer but how those questions might change with time.

The second step is to decide how to store the data in one or more tables to facilitate producing the results you want. Answering this question for a broad range of cases is far beyond the scope of this guide. We will work through a simple case of tracking the expenses and sales of a group of people to illustrate some concepts. It would also be very beneficial to understand *database normalization*, about which there are many tutorials on the internet. A key point is that there may be a clear difference between structuring the data to take advantage of how database engines work and how the data are best presented for humans to read. The former should govern the database table structure, the latter should be handled with Forms.

Creating a New Database

To begin the process of making the example database, we will make a new Base file with the default configuration of embedded HSQL tables. Choose **File > New > Database** from the menu bar, or click the arrow next to the **New** icon on the Standard toolbar and select **Database** from the drop-down menu. Both methods open the Database Wizard.

On the first page of the Database Wizard, select **Create a new database** and then click **Next**.

The second page has two questions. Make sure the choice for the first question is **Yes, register the database for me** and the choice for the second question is **Open the database for editing**. Click **Finish**.

Note

If the database is not registered, it will not be accessible to the other AOO components such as Writer and Calc. If the database is registered, other components can access it. Registering the database only informs your local OpenOffice program that this Base file exists. No information is sent to any other program or outside your computer.

Save the new database with the name *Finance*. This opens the Finance - OpenOffice Base window. Figure 1 shows part of this window.

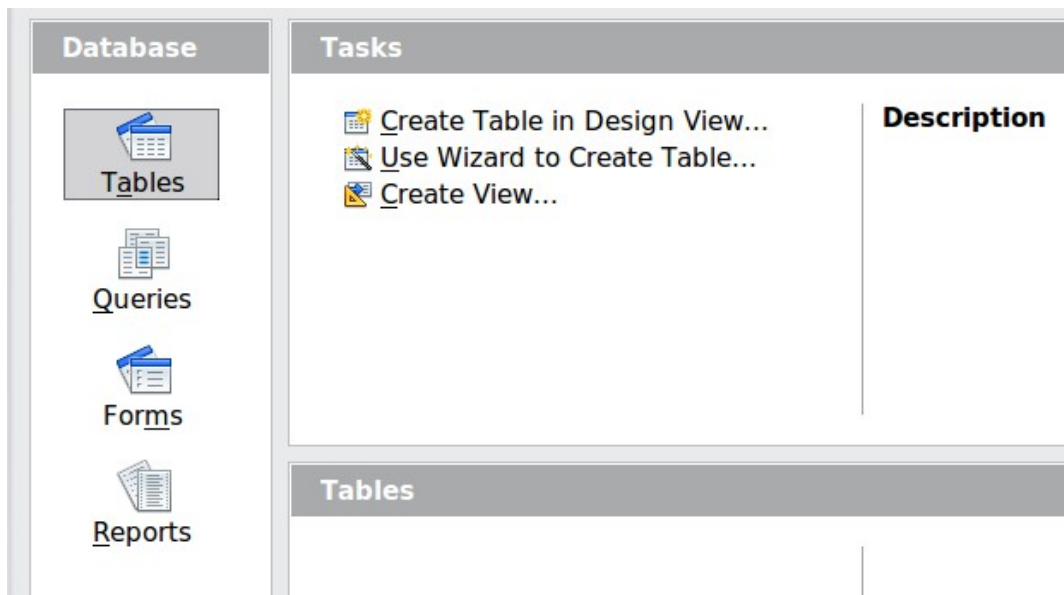


Figure 1: Creating database tables

Tip

Every time the Finance Base file is opened, the Finance - OpenOffice Base window opens. Changes can then be made to the database. The title for this window is always <database name> - OpenOffice Base.

Caution

As you create a database, you should save your work regularly. This means more than just saving what you have just created. You must save the whole database as well. For some common tasks, Base opens a separate document window where you may, for example, design a Form to enter data. Saving the document in that window saves the Form inside the Base document, but you should then also save the Base document from its main window.

Creating Database Tables

In a database, a table stores information for a group of things we call fields or columns. For example, a table might hold an address book, a stock list, a phone book or a price list. A database can have from one to several tables.

To work with tables, click the *Tables* icon in the *Database* list, or press *Alt+A*. The three tasks that you can perform on a table are in the *Tasks* list (see Figure 1).

Using the Wizard to Create a Table

Wizards are designed to do the basic work. Sometimes this is not sufficient for what we want; in those cases we can use a wizard as a starting point and then build upon what it produces.

The Table Wizard in Base contains two categories of suggested tables: business and personal. Each category contains sample tables from which to choose. Each table has a list of available fields. We can delete some of these fields and add other fields.

A field in a table is one piece of information. For example, a price list table might have one field for the item name, one for the description, and a third for the price.

Caution



Every table requires a Primary key field. (What this field does will be explained later.) We will use this field to number our entries and want that number to automatically increase as we add each entry.

Click *Use Wizard to Create Table*. This opens the Table Wizard (Figure 2).

Step 1: Select fields

We will use the *Expenses* sample table in the Business category to select the fields we need for our first table.

- 1) *Category*: Select *Business*. The *Sample tables* drop down list changes to a list of tables appropriate for a business.
- 2) *Sample tables* : Select *Expenses*. The *Available fields* box changes to a list of available fields for this table.
- 3) *Selected fields* : Using the **>** button, move the following fields from the *Available fields* window to the *Selected fields* window in this order: *AmountSpent*, *DatePurchased*, *EmployeeID*, *ExpenseType*. You can select multiple fields by pressing the *Ctrl* key while clicking and then move all the chosen fields at once.
At this point, you could select another sample table and move fields from it to the *Selected fields* list, but that is not necessary in this case.
- 4) If you make a mistake in selecting fields, click on the field name in the *Selected fields* list and use the **<** button to move it from the *Selected fields* list back to the *Available fields* list.
- 5) If you make a mistake in the order of the selected fields, click on the field name that is in the wrong order and use the **Up** or **Down** arrow on the right side of the *Selected fields* list to move the field name to the correct position.
- 6) Click **Next**.

Notice that none of the field names include spaces. It is possible to have spaces in field names, but it is best to avoid them.

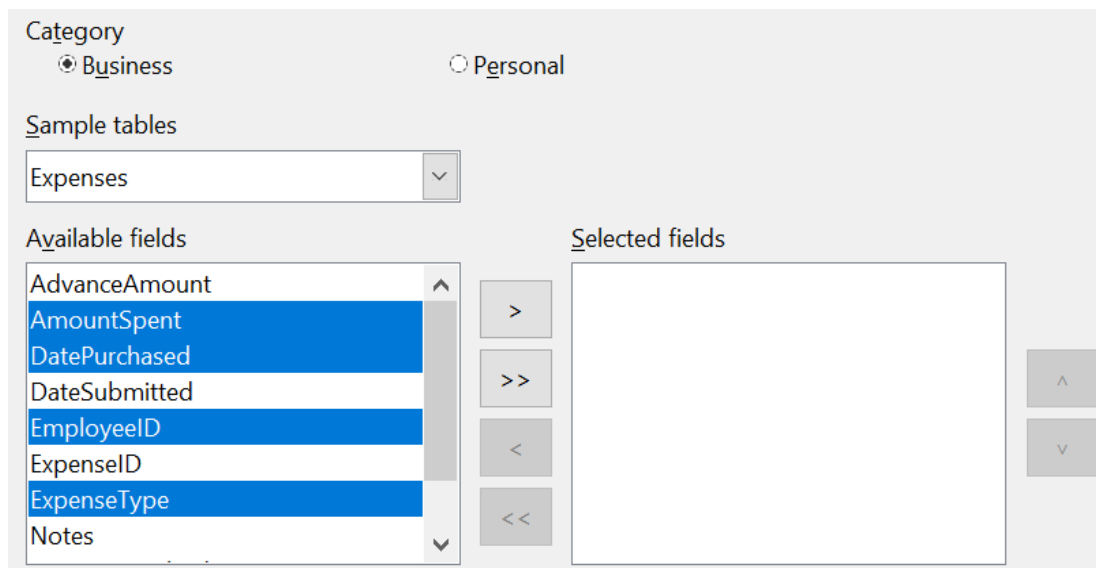


Figure 2: Selecting fields for the table

Step 2: Set field types and formats

In this step you give the fields their properties. When you click a field, the information on the right changes. (See Figure 3) You can then make changes to meet your needs. Click each field, one at a time, and make the changes listed below. Some questions you might consider in a real application are included in the changes.

Note

If any of these fields requires an entry, set *Entry required* to **Yes**. An entry with that field blank will then not be allowed. In general, only set *Entry required* to **Yes** if something must always be put in that field. By default, *Entry required* is set to **No**.

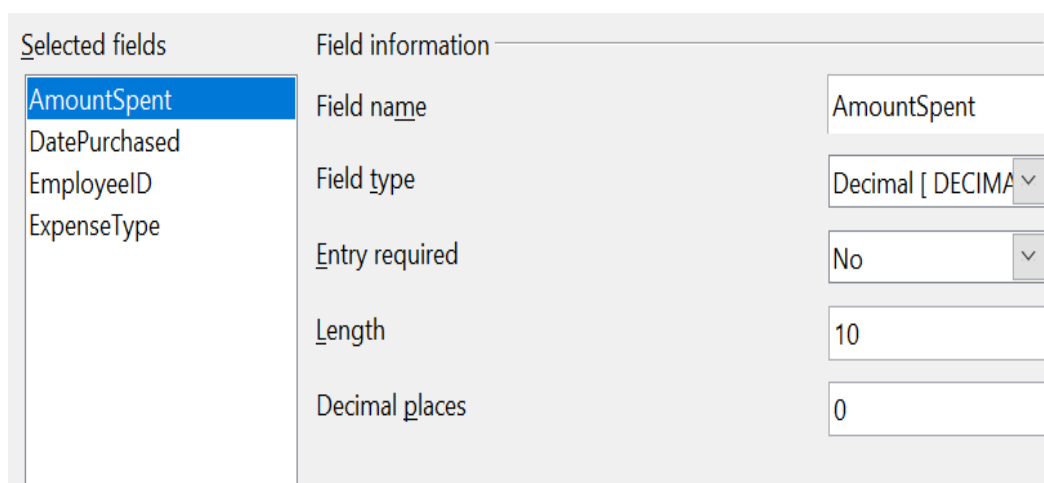


Figure 3: Changing field types

- *AmountSpent*: Change *Decimal places* to 2. You should consider whether you want to allow an expense with no amount recorded. If not, change *Entry required* to Yes.

- *DatePurchased*: Notice the *Field* type is Date, which is correct. If you want to require a date, change *Entry required* to Yes.
- *EmployeeID*: If you want to require that every expense be assigned to an employee, change *Entry required* to Yes.
- *ExpenseType*: The *Field* type is Text [VARCHAR]. With *Length* set to 50, this field must be no more than 50 characters long. Is that enough?

Note

In Base the maximum length of each field must be specified on creation. It is not easy to change this later, so if in doubt specify a greater length. Base uses VARCHAR as the default field format for text fields. This format only uses the actual number of characters in a field up to the limit set. So, a field containing 20 characters will only use space for 20 characters even if the limit is set at 50.

-
- When you have finished, click **Next**.

Note

Each field has a *Field Type*, which must be specified. Types include text, integer, date, and decimal. If the field is going to have general information in it (for example, a name or a description), use text. If the field will always contain a number (for example, a price), the type should be decimal or another numerical field. The wizard picks the right field type, so to get an idea of how this works, see what the wizard has chosen for different fields.

Step 3: Set primary key

- 1) *Create a primary key* should be checked.
- 2) Select the option *Automatically add a primary key* and check *Auto value*.
- 3) Click **Next**.

Note

A primary key uniquely identifies an item (or record) in the table. For example, you might know two people called “Randy Herring” or three people living at the same address and the database needs to distinguish between them.

The simplest method is to assign a unique number to each one: number the first person 1, the second 2, and so on. Each entry has one number and every number is different, so it is easy to say “record ID 172”. This is the option chosen here.

Step 4: Create the table

- 1) If desired, rename the table at this point. If you rename it, make the name meaningful to you. For this example, make no changes.
- 2) Leave the option *Insert data immediately* checked. We will view the table but not actually enter data.
- 3) Click **Finish** to complete the table wizard.

A window will appear titled *Expenses - Finance - OpenOffice Base: Table Data View*. This displays the empty table with the fields (columns) you selected plus a column named ID that is labeled <AutoField> and it was created in Step 3 above. The ID column will be filled with integers automatically when data is added and those

numbers will never repeat even if data is deleted. The ID column is the Primary Key of the table.

Close the window displaying the table, and you will be back to the main window of the database with the listing of the tables, queries, forms, and reports. Notice that a table named Expenses is now listed in the Tables portion of the window. Now is a good time to save the Base file.

Creating a Table by Copying an Existing Table

A situation may arise where you want to have two or more tables with the same structure but different data. Rather than creating each table from the wizard, you can make copies of the original table, naming each differently.

- 1) Click on the **Tables** icon in the Database pane to see the existing tables.
- 2) Right-click on the icon of the table you want to copy. Choose **Copy** from the pop-up menu.
- 3) Move the mouse pointer below this table, right-click, and select **Paste**. The Copy table dialog opens.
- 4) Change the table name and click **Next**.
- 5) Click the **>>** button to move all the fields from the left box to the right box and click **Next**.
- 6) Since all the fields already have the proper Field type, no changes should be needed. However, this is the time and place to make any changes if they are needed. (See Caution below for the reason.) Click **Create**. The new table is created.

Caution



Once tables have been created using the wizard and data has been entered, editing them should be very limited. Fields can be added or deleted, but adding a field requires taking the time to enter the data for that one field for every record having an entry for that field.

Deleting a field deletes **all the data** once contained in that field. Changing the field type of field can lead to data being lost, either partially or completely, if the two data types are not compatible. When creating a new table, it pays to create the fields with the correct names, length, and format before you add any data.

Deleting a table removes all the data contained in every field of the table. Unless you are sure, do not delete a table.

Creating Tables in Design View

Design View is a more flexible method for creating a new table, in which you directly enter information about each field in the table. We will use this method for the rest of the tables of our database.

Note While the Field type and formatting are different in Design View, the concepts are the same as in the Wizard.

The first table to be created is *People*. Its fields are *PeopleID*, *FirstName*, and *LastName*.

- 1) Click Create Table in Design View. A table similar to Figure 4 will appear but with no entries.
- 2) *PeopleID* field:
 - Type *PeopleID* as the first Field Name. Press the *Tab* key to move to the Field Type column.
 - Select *Integer [INTEGER]* as the Field Type from the dropdown list. (The default setting is Text [VARCHAR].)

Tip A shortcut for selecting from the Field Type dropdown list: press the key for the first letter of the choice. You can cycle through the choices for a given letter by repeatedly pressing that key.

- Change the Field Properties in the bottom section. Change *AutoValue* from *No* to *Yes*.
- As a result of choosing *AutoValue*, when you move the cursor to the next row, *PeopleID* will be set as the *Primary key* and a small key icon will appear next to the field name. To set the primary key manually, right-click on the green triangle to the left of the field name (Figure 4) and select *Primary Key* from the menu. This places a key icon in front of the field name. However, if *AutoValue* has not been set, you will need a method to track what values have been used and to create values for new entries.

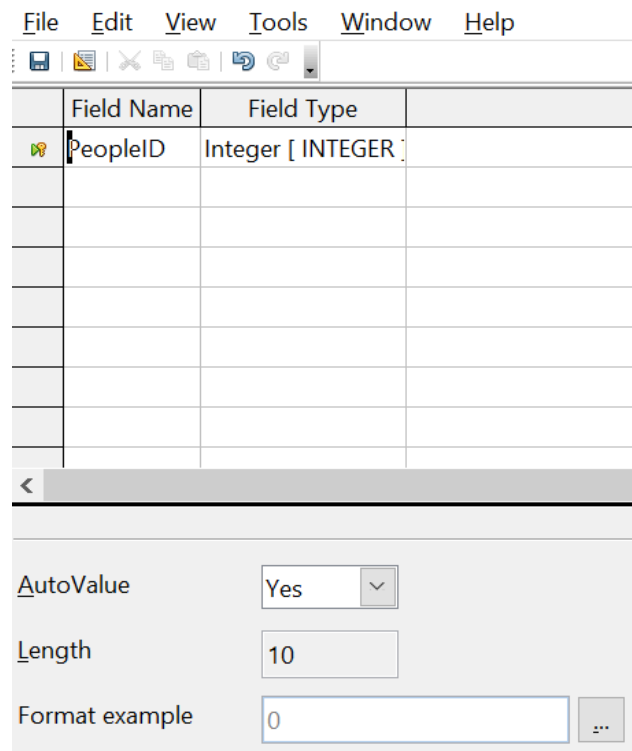


Figure 4: Table definition after the first row is defined

3) FirstName and LastName fields:

- Type the next field name in the Field Name column.
- Select the Field Type *TEXT [VARCHAR]* for each field.

4) To save and close the table, choose **File > Save**. Name the table *People*. Close the table.

Follow the same steps to create a table named *Sales*. The fields and their field types are listed in Figure 5. Make sure you make the *SaleID* field the primary key before closing. This should be true if you define the field with AutoValue = Yes. Be sure to set the *Decimal places* of the *Amount* field to 2 as shown in the figure.

You can set the display format of either the *Date* field or the *Amount* field. To do this, click the button at the right end of the *Format example* box to bring up the *Field Format* dialog. This dialog is shown in Figure 6 with a currency format chosen, which might be appropriate for the *Amount* field.

Once all the fields are defined, save the table and name it *Sales*. Close the *Sales* table and save the Base document.

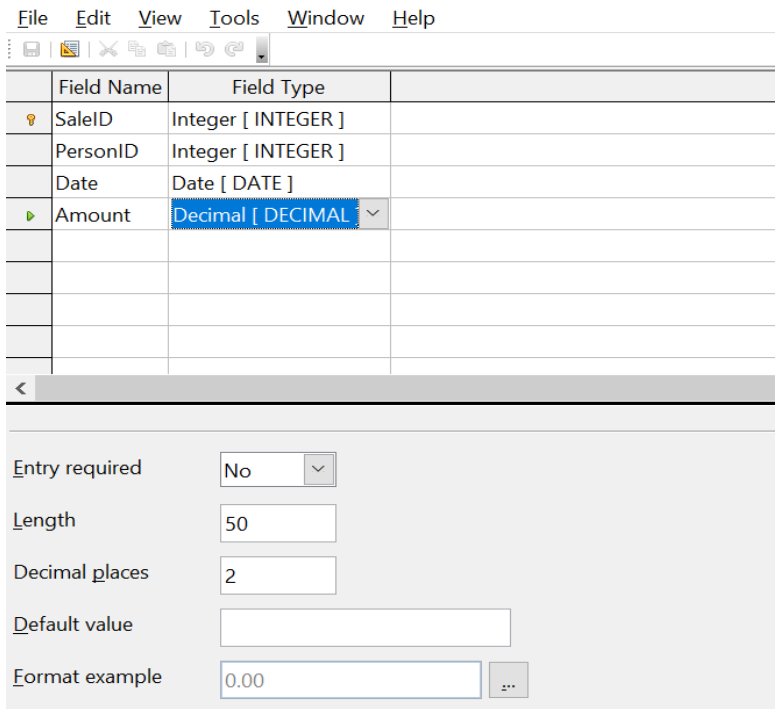


Figure 5: Fields in Sales table

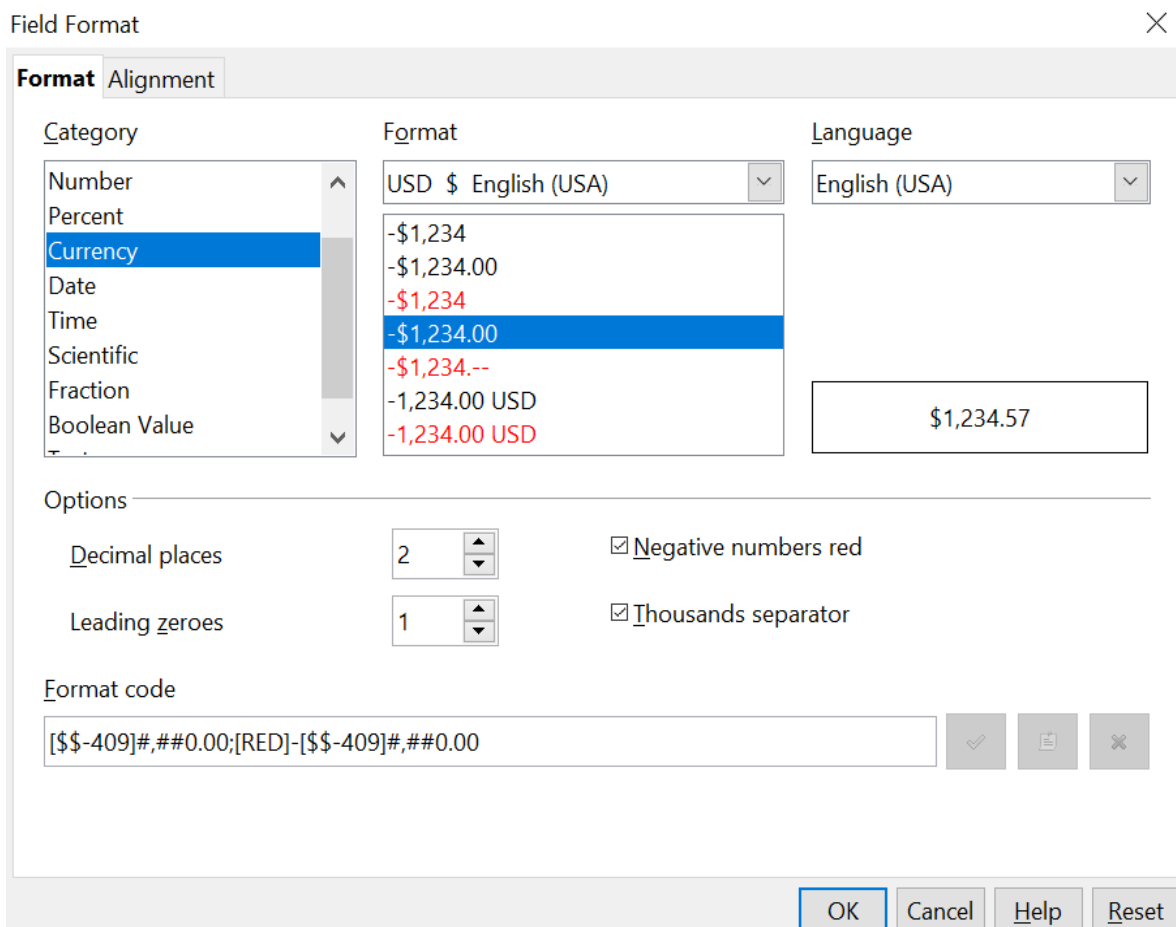


Figure 6: Field Format dialog

Creating Auxiliary Tables for Data Entry

It is often useful to have tables that provide a list of the possible values that a field in another table may have. For example, the Expenses table defined above has a field named ExpenseType. If people are allowed to enter anything in that field, a variety of words may be entered for one item. Buying gasoline might be entered as gasoline, gas, Gas, fuel, or several other possibilities. The entries might also have leading or trailing spaces that are hard to notice. All of that can be dealt with when the table is queried but limiting the entries, even if only you will use the database, leads to a cleaner data set.

For the present case of the ExpenseType field, we will make the simplest possible table with just one column that will serve as both the primary key and the useful data.

Click the Tables icon in the Database region of the main window and choose *Create Table in Design View* in the Tasks pane. Give the field the name *ExpType* and leave the field type as *TEXT [VARCHAR]*. Before leaving that row of the design table, right-click on the small green arrow at the left edge of the row and select *Primary Key*. Save the table using **File > Save** with the name *Expense_Categories* and close it.

Adding Data to the List Table

You should now be back at the main Base window. In the *Tables* pane, right-click on the icon for *Expense_Categories* and select *Open*. Enter the values *Supplies, Equipment, Travel, Meals* as the first four values of the *ExpType* column. You can close the table without saving it, because the values were saved automatically as you entered them. Do be sure to complete the entry of the last value by pressing *Enter* or the down arrow to move to a new row before closing the window.

An alternative to this very simple table structure is to define a primary key column that is separate from the values you want to display in the Form when entering data. This is discussed further in the next section.

Defining Relationships

The structure that we have defined for the database tables assumes some relationships that have not yet been made explicit. The following table shows each database table and its fields.

<i>Expense_Categories</i>	<i>Expenses</i>	<i>People</i>	<i>Sales</i>
ExpType	ID	PeopleID	SaleID
	AmountSpent	FirstName	PersonID
	DatePurchased	LastName	Date
	EmployeeID		Amount
	ExpenseType		

Both the *Expenses* table and the *Sales* table have fields to identify people: *EmployeeID* and *PersonID*. When data is being entered into these tables, the

PeopleID values are imported from the *People* table. This is known as a Foreign Key. The advantage of this relationship is that the *FirstName* and *LastName* values associated with the ID appear in only one place, a single row in the *People* table. If a person's name should change for some reason, only one row in one table needs to be edited. In this very simple example, that is not a very daunting task but in a real situation where there might be tables for salary, medical benefits, vacation, leaves, emergency contacts and much more, updating every entry in all the tables would be prone to error and quickly result in very unreliable data.

A similar relationship exists between the *Expenses_Categories* and the *Expenses* table, though the purpose is different. In that case, the values of *ExpType* serve to control the values that can be entered as an *ExpenseType* in the *Expenses* table. This is another form of a Foreign Key.

Tip As you create your own databases, you need to also determine where tables are related and how.

- 1) To begin defining relationships in the database, choose **Tools > Relationships**. The *Finance - OpenOffice Base: Relation Design* window opens and the *Add Tables* dialog pops up. (You can also open it by clicking the Add Tables icon, highlighted in red in Figure 7, on the Relation Design window.)
- 2) In this database, all the tables have a relationship to another table and must be added to the Relation Design Window. On the Add Tables dialog, use either of these ways to add a table to the Relation Design window:
 - Double-click the name of the table.
 - Or, click the name of the table and then click **Add** for each table.
- 3) Click **Close** to close the Add Tables dialog when you have added the tables you want.

Figure 7 shows the Relation Design window in its final state after the relations have been created. At the current state of the process you will not see the lines connecting the tables. The tables may also be closer together and not display all of their fields. Feel free to drag the tables farther apart and resize them by clicking on their edge and dragging.

You can define the relationship between pairs of tables in two ways:

- Click and drag the primary key field in one table to the foreign key field in the other table. When you release the mouse button, a connecting line forms between the two data fields. For example, drag from *ExpType* in *Expense_Categories* to *ExpenseType* in the *Expenses* table.

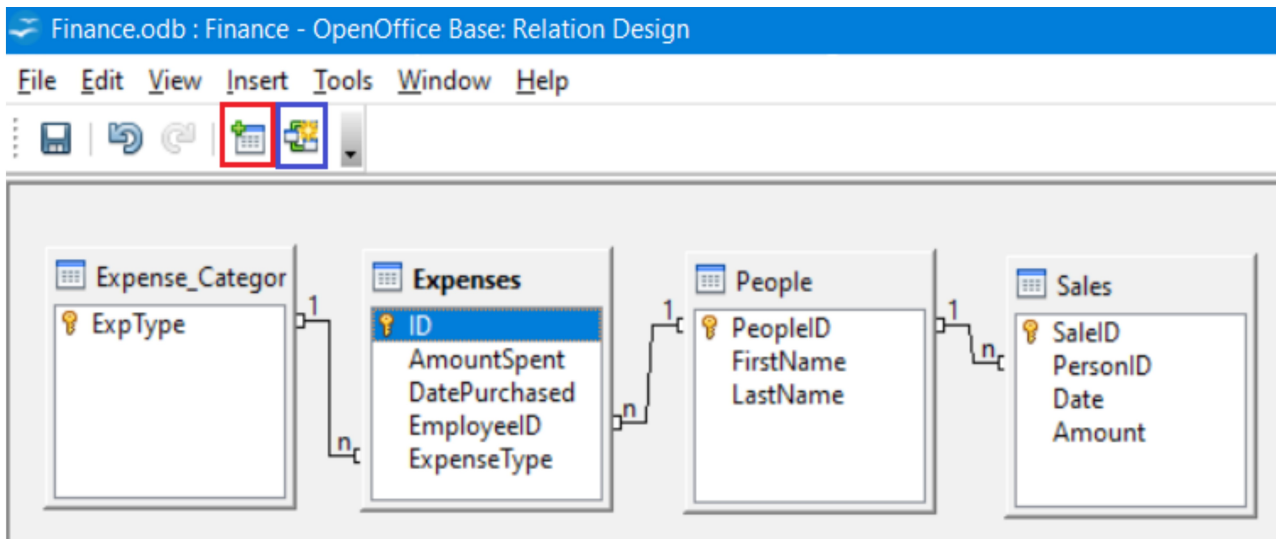


Figure 7: Designation for a 1:n relationships

- Or, click the **New Relation** icon, highlighted in blue in Figure 7. This opens the Relations window (Figure 8). The available tables are listed in the drop-down lists in the *Tables involved* section.
 - Select a table that holds the primary key from the left-hand drop-down list and the corresponding table holding the foreign key from the right-hand drop-down list. Figure 8 shows the tables *Expense_Categories* and *Expenses* selected.
 - In the *Fields involved* section, click the drop-down list under the *Expense_Categories* label.
 - Select *ExpType* from the list.
 - Click in the cell to the right of this drop-down list. This opens a dropdown list for the *Expenses* table.
 - Select *ExpenseType* from the *Expenses* table list. The dialog should now look like the top half of Figure 8.
 - Under *Update options* select **Update cascade** and under *Delete options* select **Delete cascade**.
 - Click **OK**.

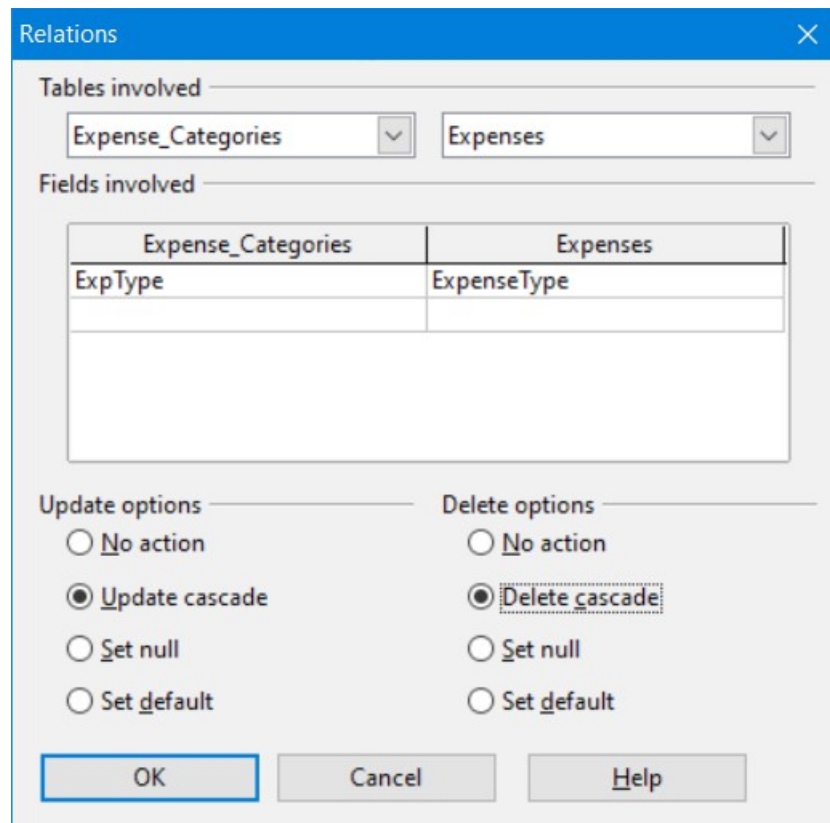


Figure 8: Selected fields in a relationship

The Update cascade and Delete cascade options control what happens when one of the primary key options is updated (changed) or deleted. The cascade option means that any rows in the Expenses table using the modified or deleted value will likewise be modified or deleted. While these options are not strictly necessary, they do help. Having them selected permits you to update a table that has a relationship defined with another table. It also permits you to delete a field from the table.

You can now repeat the process of step 4 to make the relationships between the *People* table and the *Expense* and *Sales* tables. At this point the Relation Design window should look like Figure 7. After closing the Relation Design window, it is a good idea to save the Base document.

Creating a Database Form

Databases are used to store data. But how is the data put into the database? Forms are used to do this. In the language of databases, a form is a front end for data entry and editing.

A simple form consists of the fields from a table. More complex forms can contain much more, including additional text, graphics, selection boxes, and many other elements. We will make three simple forms to demonstrate the basics of tying form controls to database tables, viewing and entering data, and modifying the function and appearance of form controls. The data tables are so simple that they are not realistic, but they serve the purpose of showing how forms work. The examples do not use all the available types of form controls, but the principles are much the same for all controls.

Using the Wizard to Create a Simple Form

Making the Form

We will first use the Form Wizard to create a very simple People form for entering data in the *People* table.

In the main database window (Figure 1), click the **Forms** icon in the left column. In the Tasks list, click **Use Wizard to Create Form** to open the Form Wizard (Figure 9). On the left, you can see the possible steps to create a form. Simple forms require only some of these steps, while more complex forms may use all of them. Towards the center, there is a drop-down list titled *Tables or queries* where you can see that the *People* table has been chosen.

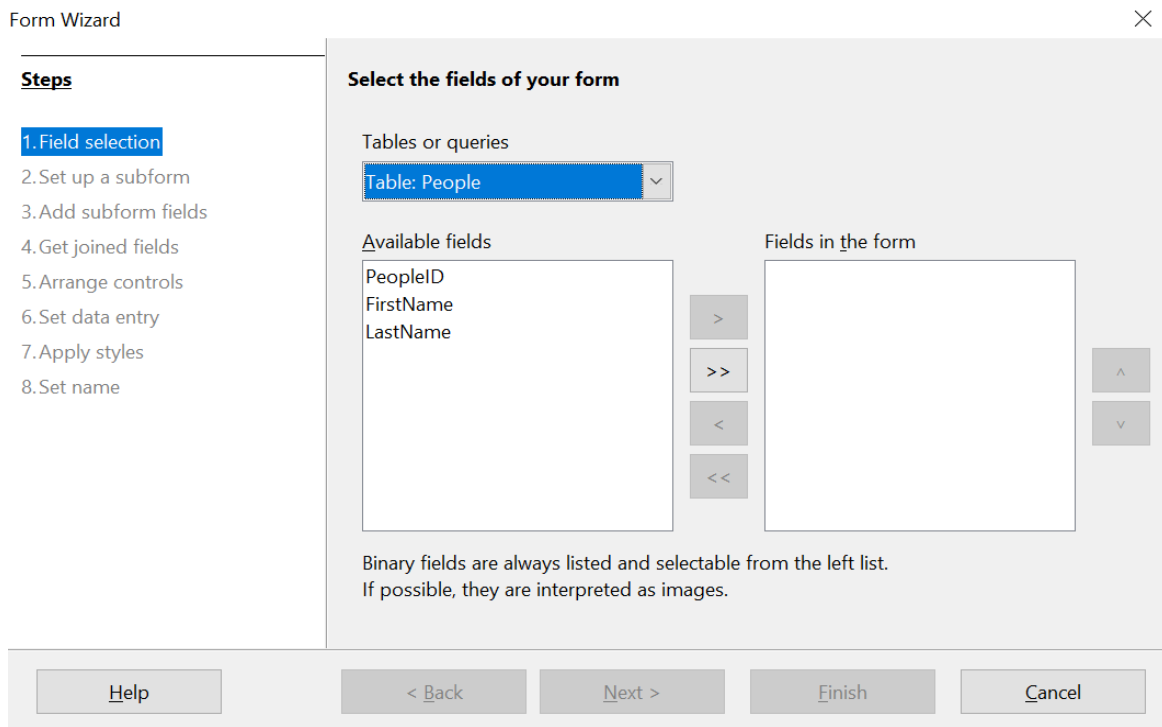


Figure 9: Form Wizard steps

Step 1: Select fields

- 1) Under *Tables or queries*, select *Table: People*. *Available fields* lists the fields for the *People* table.
- 2) Click the field *FirstName* and then on the right single arrow to move it to the *Fields in the form* list. Do the same with *LastName*. Click **Next**. We do not put the *PeopleID* field on the form because it was defined as an *AutoValue* field, and it will be populated automatically.

Step 2: Set up a subform

This form will not have a subform. Click **Next** and the dialog will jump to Step 5.

Step 5: Arrange controls

Each control in a form consists of two parts: a label and a field. This step in creating the form determines where a control's label and field are placed in relationship with

each other. The four choices from left to right are *Columnar - Labels Left*, *Columnar - Labels on Top*, *As Data Sheet*, and *In Blocks - Labels Above*. Click the third icon from the left in the *Arrangement of the main form* section, the *As Data Sheet* icon, then click **Next**.

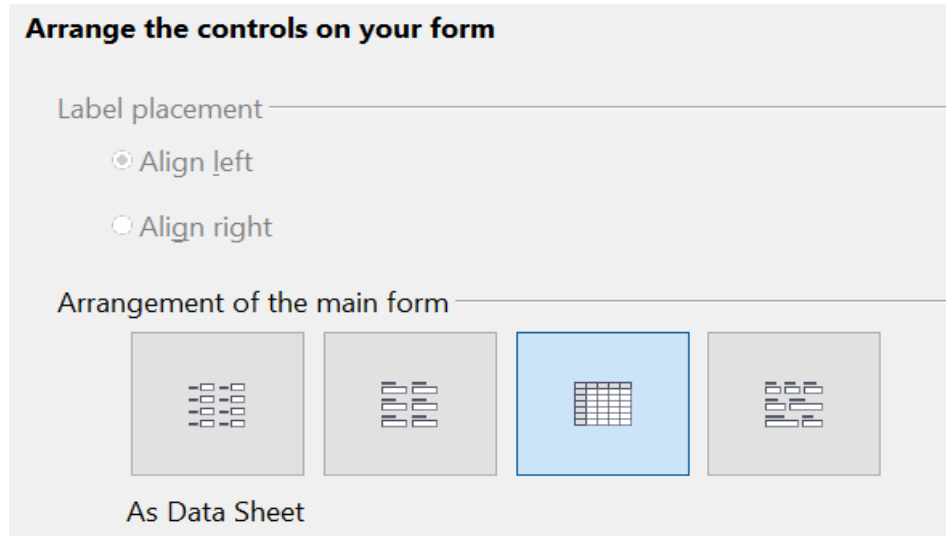


Figure 10: Form Wizard Step 5 - Detail

Step 6: Set data entry

This step allows limiting what can be done with the form, such as preventing data deletion or viewing existing data. We will leave the defaults in place, but this could be an important step in securing data. Click **Next**.

Step 7: Apply styles

You can adjust the appearance of the form at this step with different colors and borders. We will leave the default in place, but you might want to try different settings. Click **Next**.

Step 8: Set name

You can give the form any name; there is no need to match the name of the table to which it is connected. We will keep the default name of *People* and also accept the option *Work with the form*. Click **Finish** and the form will be ready for data entry. It is a good idea to close the form now and save the main Base document.

Entering Data With the New Form

If you have closed the form, you can open it by clicking on the Forms icon in the main Base window (Figure 1) and then double-clicking on the *People* icon in the list of Forms.

Data can be entered into the *People* table by typing into the *FirstName* and *LastName* fields. You can navigate from field to field by selecting them with the mouse, using the *Tab* key or using the buttons of the navigation bar along the bottom of the form's data sheet. The navigation bar shows the current record number, the

total number of records and buttons which move the selection to the first record, the previous record, the next record, the last record, and a new record. As you enter data, moving out of a record automatically saves it to the database table. Be sure to exit the last row into which you enter data so that it gets written to the database table.

In later sections, we will add controls to forms that include dropdown lists and there are other methods of automating data input such as buttons to increment numeric values. The basic steps are always same. You navigate from field to field by clicking or with the *Tab* key and moving to another record automatically saves the entered data. If a record does not meet a requirement placed on the table, for example, not entering data in a field for which input is required, the form will not move to a new record and an error message may appear.

Modifying the Form

The Form Wizard's choices for the size and layout of controls will probably not be ideal. Adjustments can easily be made by opening the form in edit mode. When the form is closed, right click on its icon and select *Edit*. The form should open ready to be edited, but we will explain how to toggle in and out of Design Mode. If the *Form Controls* toolbar is not visible, open it with **View > Toolbars > Form Controls**. The toolbar is shown in Figure 11 with Design Mode on and off and with the button that does the toggling highlighted with red. When Design Mode is off, most of the toolbar's icons are gray.

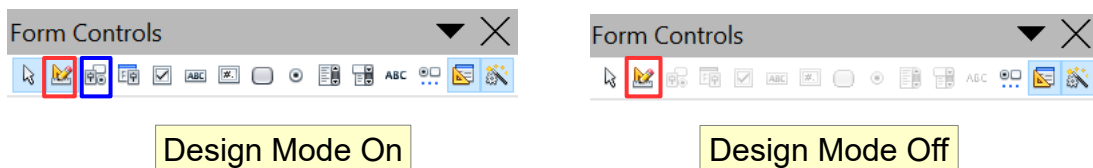


Figure 11: Form Controls toolbar with Design Mode on and off

With Design Mode on, you can click on the form's data sheet to select it. Eight green handles will appear at the edges of the data sheet. You can click and drag with these to resize it. You can also resize the data sheet's columns by dragging the boundary between them. When resizing the sheet, pay attention to the navigation bar at the bottom so that you do not obscure some controls. You can control more details of the appearance of the data sheet using the *Table Control* dialog. To access this, right-click on the data sheet and select *Control* or select it with a left click and then click on the *Control* icon on the *Form Controls* toolbar. The *Control* button is highlighted in blue in Figure 11. In the dialog, you can adjust the size and position, font, background color, border style, and turn off the navigation bar.

Using the Wizard to Make a Form With a Subform

Making the Form

Forms can be arranged in a hierarchy so that the data selected in one form influences the data shown on the second form. The objects referred to as forms here are different from what we have previously called a form. If you click on the *Forms* icon on the main Base page, you will see a pane labeled Forms that displays forms previously saved in the Base file. In the file we have been working with there is now one form called *People*. A more precise descriptor for *People* would be *Form Document*. A Form Document contains one or more groups of Form Controls that

interact with a database table or query. In the *People* Form Document, there is only one control, a data sheet, and it reads data from and writes to the *People* table. We could have had multiple controls, one dealing with each field in the table, but they all would have interacted with one table. Such a group of controls is also called a form, though a more precise name would be a *Data Form*. It is possible, even common, to have a Form Document that contains two or more Data Forms, each Data Form interacting with a different table but also some Data Forms influencing other Data Forms. We will now use the Form Wizard to make a Form Document that contains two Data Forms arranged as Form - Subform. The main Form will use the *People* table and the Subform will use the *Expenses* table. The two Data Forms will allow us to view and record the expenses of a chosen person.

In the main database window (Figure 1), click the **Forms** icon in the left column. In the Tasks list, click **Use Wizard to Create Form** to open the Form Wizard (Figure 9).

Step 1: Select fields

- 1) Under *Tables or queries*, select *Table: People*. The box *Available fields* lists the fields for the *People* table.
- 2) Click the field *FirstName* and then on the right single arrow to move it to the *Fields in the form* list. Do the same with *LastName*. Click **Next**.

Step 2: Set up a subform

Since we have already created a relationship between the *People* and *Expenses* tables, we will use that relationship. If no relationship had been defined, this would be done in step 4.

- 1) Click the box labeled *Add Subform*.
- 2) Click *Subform based upon existing relation*.
- 3) Both the *Expenses* and *Sales* tables have a relation with *People*, so both are listed as possible choices. Click *Expenses* to highlight it, as in Figure 12. Click **Next**. Even if only one table were listed, it would still be necessary to click on it to select it for the subform.

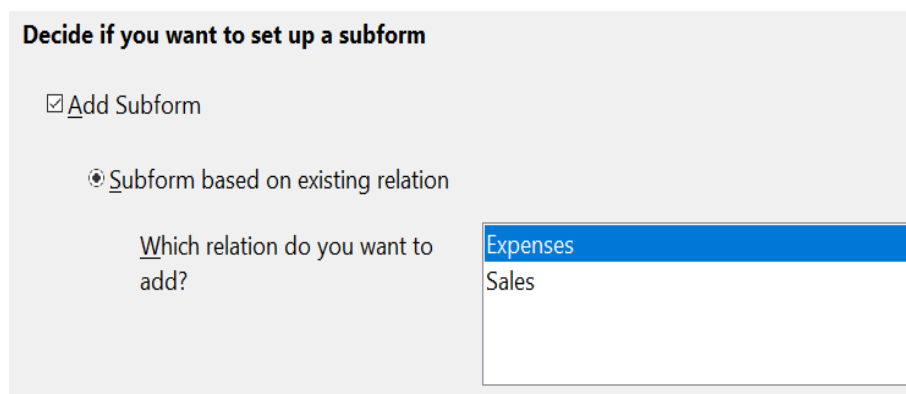


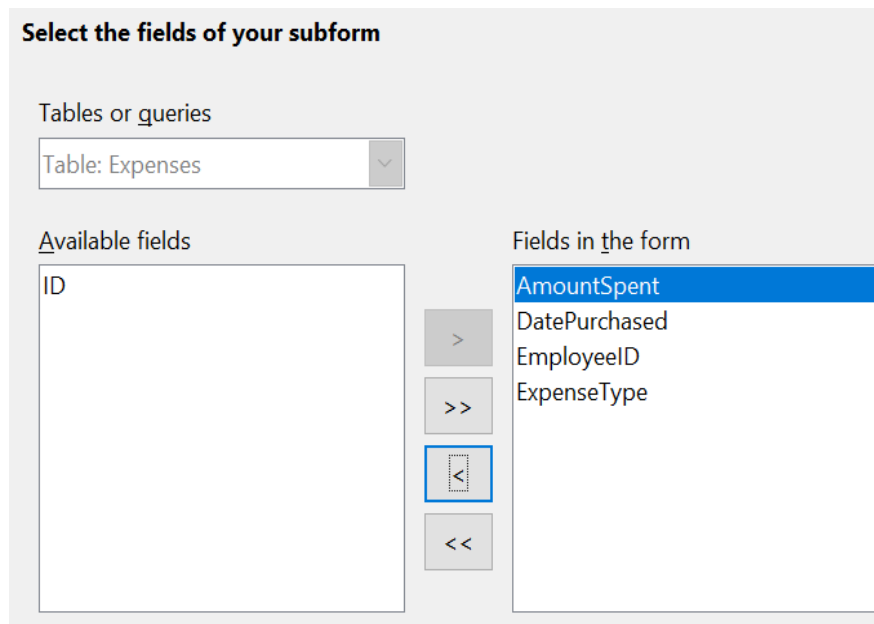
Figure 12: Adding a subform and choose the relation.

Step 3: Add subform fields

This step is much like step 1.

- 1) *Expenses* is preselected under *Tables or queries*.
- 2) Use the **>>** button to move all the fields to the right.
- 3) Click the *ID* field to highlight it.

- 4) Use the < button to move the *ID* to the left (Figure 13). We are excluding the *ID* field because its value is set automatically. There can be circumstances where you would want to be able to view the *ID* and displaying it in the form is a valid choice. Click **Next**.



The screenshot shows a dialog box titled "Select the fields of your subform". At the top, there is a dropdown menu labeled "Tables or queries" with "Table: Expenses" selected. Below this are two lists: "Available fields" on the left and "Fields in the form" on the right. The "Available fields" list contains "ID". The "Fields in the form" list contains "AmountSpent", "DatePurchased", "EmployeeID", and "ExpenseType". Between the two lists are five buttons: ">", ">>", "<", and "<<". The "<" button is highlighted with a blue border, indicating it is the next action to be taken.

Figure 13: Selecting fields of a subform

Step 4: Get Joined Fields

This step is for tables or queries for which no relationship has been defined. Because we have already defined the relationship, the wizard skips this step.

Note It is possible to create a relationship between two tables that is based upon more than one pair of fields.

Caution



When selecting a pair of fields from two tables to use as a relationship, they have to have the same field type. The chosen columns in our example are both INTEGER.

Whether a single pair of fields from two tables are chosen as the relationship or two or more pairs are chosen, certain requirements must be met for the form to work.

- No field from the subform can be the Primary key for its table. (ID cannot be used.)
 - Each pair of joined fields must have the same field type.
 - One of the fields from the main form must be the Primary key for its table.
-

Step 5: Arrange controls

Set the arrangement of both the main form and the subform to be *As Data Sheet*, which is the third icon from the left. Click **Next**.

Step 6: Set data entry

Accept the default settings. Click **Next**.

Step 7: Apply styles

- 1) Select the color you want in the *Apply Styles* list.
- 2) Select the Field border you want.
- 3) Click **Next**.

Step 8 : Set name

- 1) Enter the name for the form. In this case, it is *Expenses*.
- 2) Click *Work with the form*.
- 3) Click **Finish**. The form opens ready for use. Though you can use the form immediately, it is a good time to close the form and save the main Base document.

Entering Data in the Subform

Reopen the *Expenses* form document if you closed it. If you have already entered data in the *People* table as described in *Entering Data With the New Form*, you will see that data in the main form. Figure 14 shows three people entered into the *People* table and the subform ready for data.

	FirstName	LastName	
▶	Ana	Trujillo	
	John	Brown	
	Helen	Wu	
✱			

Record 1 of 3

	AmountSpent	DatePurchased	EmployeeID	ExpenseType
▶			0	

Record 1 of 1

Figure 14: Main and subform of Expenses

Notice that the *EmployeeID* of the subform is already populated. The 0 is the *PeopleID* of Ana Trujillo in the *People* table, which is the record that is currently selected as can be seen by the green triangle at the left edge of the row. Remember that *PeopleID* and *EmployeeID* have a foreign key relationship as seen in Figure 7. If you select another person by clicking on the record or using the navigation bar, the *EmployeeID* in the subform will automatically change.

We could enter data for each employee now. The *AmountSpent* can be any number, the *DatePurchased* should be in the format appropriate for your locale, though YYYY-MM-DD should work anywhere, and the *ExpenseType* must be one of the values in the *Expense_Categories* table. If you try to enter some other value, say, Fish, the database will refuse to accept it. No error message appears, but you cannot move down to the next row with an invalid value in *ExpenseType*. Similarly, you can enter a number in *EmployeeID*, but if it is not a value that exists in the *People* table, it will not be accepted. This is a result of the foreign key relationships set up earlier and is an important tool for maintaining data integrity. Rather than struggling to remember valid *ExpenseTypes*, we can modify the form to improve the ease of data entry.

Modifying the Subform

In addition to making it easier to enter the *ExpenseType*, there are a few other changes that would improve the usability or appearance of the form. The column headers currently show the field names and could be replaced with labels closer to

nontechnical language. Also, the *DatePurchased* column would benefit from a calendar widget to decrease the amount of typing.

Open the form document by right-clicking on its icon and selecting *Edit*. The first step is to put the two Data Sheets into Design Mode. If the *Form Controls* toolbar is not visible, open it with the menu **View > Toolbars > Form Controls**. As shown in Figure 11, use the Design Mode button, which is highlighted in red, to toggle Design Mode to On.

Right-click the *ExpenseType* header and select **Replace with > Combo Box**. The column had been a set of Text Boxes. These are controls which accept text entries. A Combo Box is very similar except that it provides a drop-down list of entries. To set the properties of the Combo Box, right click again on the *ExpenseType* header and select **Column**. A dialog titled *Properties: Combo Box* will appear.

- 1) On the *General* tab, change the *Label* to *Expense Type*, with a space between the words.
- 2) On the *Data* tab, make sure *Type of list contents* is **Sql** and *List content* is `SELECT "ExpType" FROM "Expense_Categories"`.

The code `SELECT "ExpType" FROM "Expense_Categories"` is in the SQL language that is used to interact with databases. We will not cover SQL in any detail but understanding simple statements like this is useful for setting up Combo Boxes and List Boxes that can display values stored in database tables. The `SELECT` keyword means "Show the following columns" and the `FROM` keyword means "Use the following tables to find the data". So, the entire statement means "Show the *ExpType* column from the *Expense_Categories* table". The only other detail is that the column and table names are in double quotes.

To make the other changes in the Data Sheet, click on each column header and select **Column**.

- 1) For the *AmountSpent* field, change the *Label* to *Amount Spent*. You may also adjust the *Formatting* property by clicking the small button with three dots at its right edge and using the dialog to set the *Category* to *Currency* and the *Format* to whichever setting you prefer.
- 2) For the *DatePurchased* field, select the *General* tab and set the *Label* to *Date Purchased* then scroll down to find the Drop-down listing and change it to **Yes**.

It would also make sense to delete the *EmployeeID* column. Its value is set through the foreign key relation. It was displayed only to show the effect of the relation. We will not do that, but a column is deleted by clicking on the header and selecting **Delete Column**.

Take the forms out of Design Mode and try entering some data. You should be able to use dropdown lists in the *Date Purchased* and *Expense Type* columns and the *Amount Spent* column will be displayed with a currency format if you chose to set that up.

Laying Out a Form With Multiple Controls

A form with multiple controls will likely need some adjustments to their layout and appearance. We will make a form for the Sales table as a tool to demonstrate how to do this. The form could be made with a Form - Subform arrangement as was done for the Expenses form document.

Making the Form

In the main database window (Figure 1), click the **Forms** icon in the left column. In the Tasks list, click **Use Wizard to Create Form** to open the Form Wizard (Figure 9).

Step 1: Select fields

- 1) Under Tables or queries, select *Table: Sales*.
- 2) Move all the fields to the list on the right using the **>>** button then click **Next**. Since *SaleID* is an AutoValue field, you might normally choose to not display it in the form.

Step 2: Set up a subform

This form will not have a subform. Click **Next** and the dialog will jump to Step 5.

Step 5: Arrange controls

In the *Arrangement of the main form* section, choose the second icon from the left which sets the layout to *Columnar - Labels on top*. Click **Next**.

Step 6: Set data entry

Accept the defaults by clicking **Next**.

Step 7: Apply styles

Accept the defaults by clicking **Next**.

Step 8: Set name

Choose the name *Sales* and the option *Modify the form* and then click **Finish**.

You will now have a data form with four controls in a single column as shown in Figure 15. This is a perfectly serviceable arrangement, but we will change it to have the controls horizontally arranged towards the middle of the form. If the Form Controls toolbar is not visible, open it with the menu **View > Toolbars > Form Controls**. Use the Design Mode button, which is highlighted in red, to toggle Design Mode to On as shown in Figure 11.

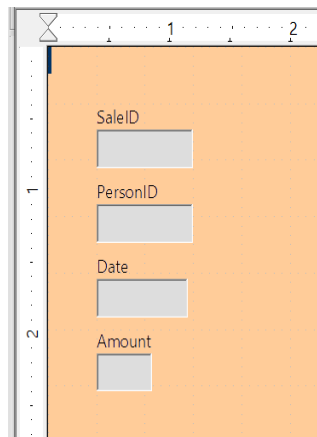


Figure 15: Initial control arrangement of the Sales form

The entire group of controls can be moved by selecting them with click and drag. Make sure the cursor is in select mode by clicking the Select icon on the *Form Controls* toolbar. It is the first icon on the toolbar. Click and hold above and to the left of the SaleID label and drag down and to the right to form a rectangle around all the controls. When you release the mouse button, green handles will define a rectangle encompassing all the controls. Click and hold within this rectangle and drag the group to the right. Click anywhere outside the rectangle to release the grouping of the controls. To arrange the controls horizontally, we will first drag them into their approximate positions and then fine tune them with *Position and Size* dialogs. Each control consists of a label and a field and if you click on one of the controls, both the label and the field are selected. After placing the label/field pairs close to their desired final positions, we can examine and adjust their precise positions. Figure 16 shows the controls after clicking and dragging.

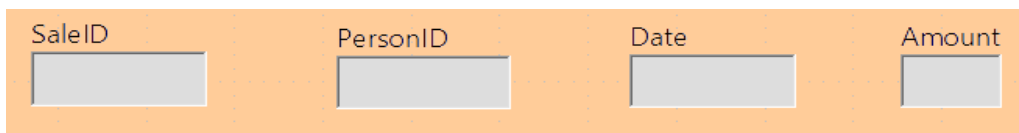


Figure 16: Controls approximately horizontally arranged

Since we are dealing with an artificial set of data, we do not have strong reasons to arrange the controls in any particular way. We will make the rather arbitrary decision to make each field box one inch long and put a half inch between them. We also want them all at the same vertical position.

The label and the field are actually two objects that can be edited independently. To access one of them, use *Ctrl + Click*. Figure 17 shows the difference between clicking on the field of the SaleID control using just the left mouse button or using *Ctrl + Click*.

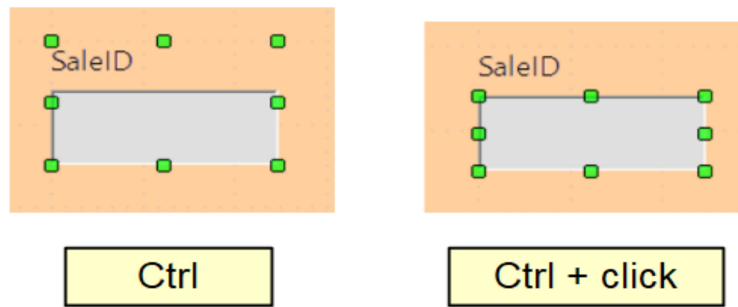


Figure 17: Click vs. Ctrl + click

After using *Ctrl + Click*, you can move from the field to the label and vice versa with the *Tab* key. If you want to move both the label and the field, you should be careful that you have selected both with a plain click, so you do not undo their alignment accidentally.

To examine and adjust a control's position, click on it, then right click and select *Position and Size*. The resulting dialog is shown in Figure 18.

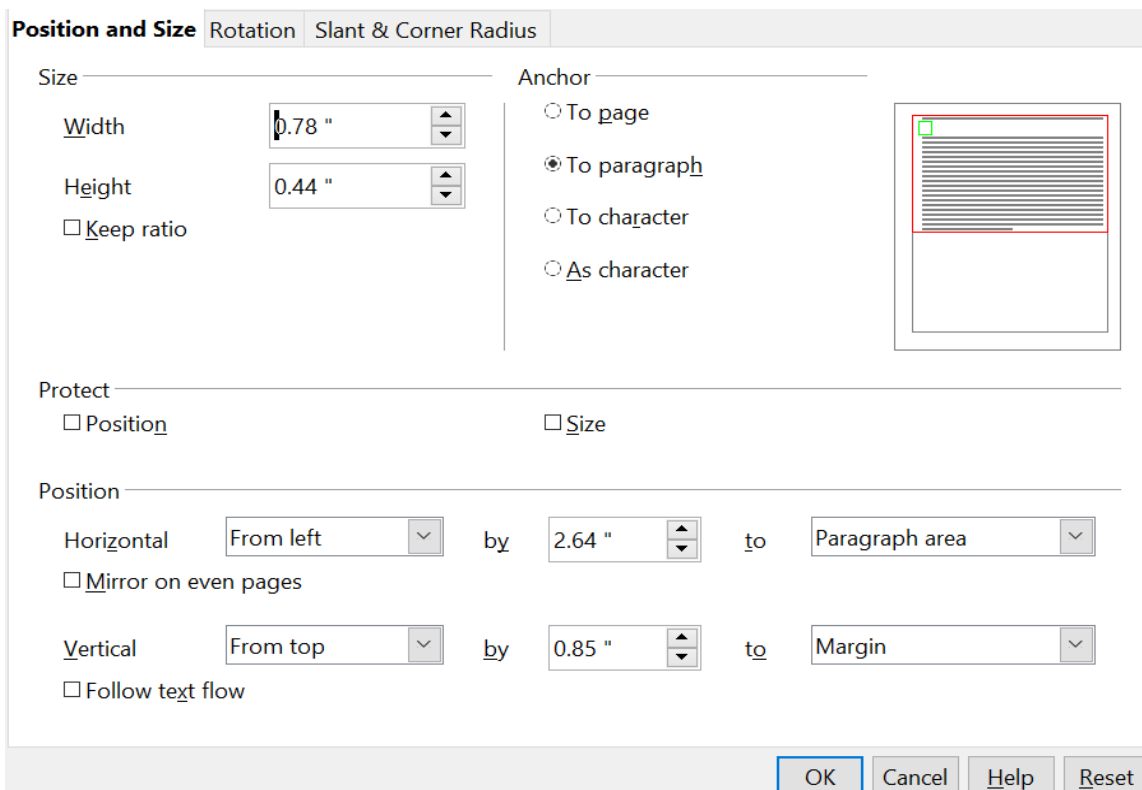


Figure 18: Position and Size dialog of a control

The width and size of the label/field pair is shown at the top left while the position on the page is shown at the bottom. We will set the horizontal position to 2.50 and note that the vertical position is 0.85. We will leave the width unchanged because we want to change the width of only the field and not the label. Likewise, we then exit the dialog and use *Ctrl + Click* to select only the data field of the *SaleID* control. Furthermore, we can as well right-click on the field, choose *Position and Size* and adjust its width to one inch. We can also use *Ctrl + Click* to select the label, right click on it, select *Control*, and change the label text, font color, or any other property. We

can then proceed to the other controls, setting them to be on a 1.5 inch (3.81 cm) pitch with each data field having a width of one inch and placing them all at the 0.85 inch (2.16 cm) vertical position. Furthermore, we should also set the *Date* field to have a dropdown list by using *Ctrl + Click* to select only the date field, then right-clicking and selecting *Control*.

The controls are now evenly positioned, but there are three remaining adjustments to make the form easily usable.

Navigation Bar

The first adjustment is to add a Navigator Bar to move from record to record and add new records. There is a Navigation Bar icon available in the *More Controls* auxiliary toolbar of the *Form Controls* toolbar. Figure 19 shows the location of the *More Controls* icon and the *Navigation Bar* icon.

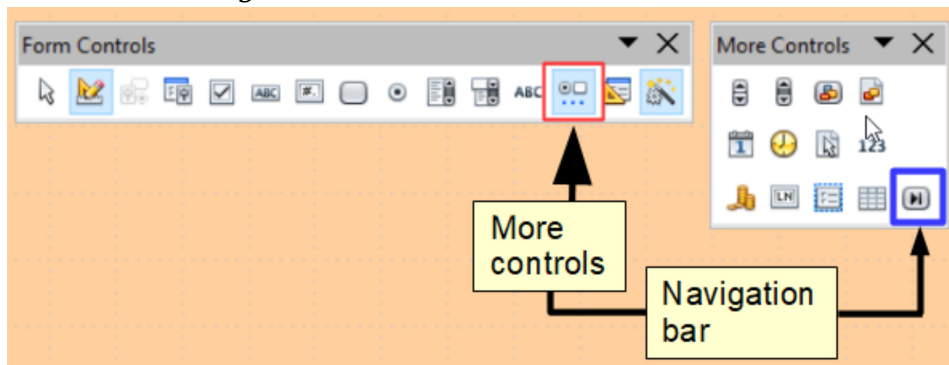


Figure 19: More Controls toolbar and Navigation Bar icon

Click the *Navigation Bar* icon and use click and drag to place the bar. You can select it and lengthen it manually until all of its buttons are visible and use the methods just discussed for positioning it precisely. It has buttons which move the selection to the first record, the previous record, the next record, the last record, and a new record.

Adding a List Box

Another change needed to improve the usability of the form is to change the *PersonID* field into a List Box. The Form Wizard chose a Formatted Field for this control, which is basically a numeric field. You can see this by selecting only the *PersonID* data field, right-clicking and selecting *Control*. The title bar shows that the control is a Formatted Field. The *PersonID* field is indeed numeric, it stores the *PeopleID* value from the *People* table as shown in Figure 7. To input data easily, we would want to know which *FirstName* and *LastName* correspond to a *PeopleID* value. A List Box is like a Combo Box in that it displays values, but it displays one value and stores another. We will use a List Box to display the combined *FirstName* and *LastName* values of a row from the *People* table and store the corresponding *PeopleID* value in the *Sales* table *PersonID* field.

Use *Ctrl + Click* to select the field of the *PersonID* field then right click and choose **Replace with > List Box**. Right click again on the field and select **Control**. On the *Data* tab set *Type of list contents* to **Sql** and set *List content* to
`SELECT "FirstName" || ' ' || "LastName", "PeopleID" FROM "People"`
The *Bound field* should be set to 1 and on the *General* tab, the Drop-down list should be set to **Yes**.

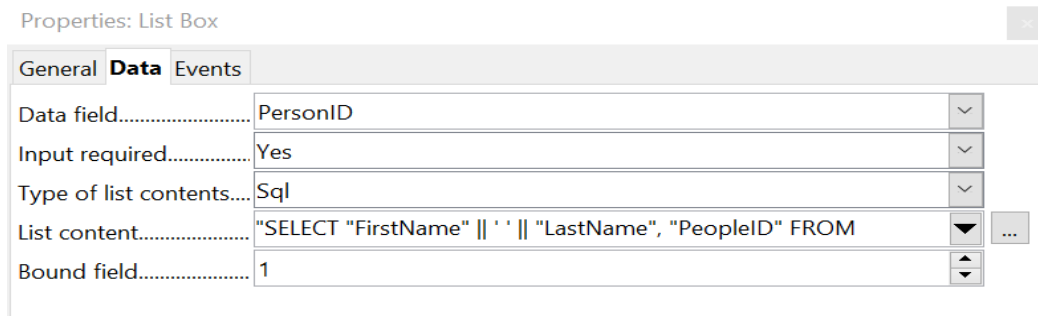


Figure 20: List Box set up

The meaning of a very similar SQL statement was explained in the section Modifying the Subform. The fields named after the SELECT keyword are a little more complicated here. The || symbol acts to concatenate or glue together the text on each side of it. "FirstName" || ' ' || "LastName" means "The value of the FirstName field followed by a space followed by the value of the LastName field". We will call this the Full Name of the record. The full SELECT clause returns the Full Name followed by the *PeopleID* of the record in the *People* table. The *Bound Field* value tells the control to store the second returned item (the control counts from zero) in the target table. This means *PeopleID* will be sent to the *PersonID* field of the *Sales* table. Figure 21 shows the result of setting up the List Box. The *Person ID* dropdown list shows the full name of the person, but the value stored in the *Sales* table is the integer *PeopleID*. This can be confirmed after some data are entered by going to the main window of the Base file, selecting the *Tables* icon and double-clicking on the *Sales* table.

The advantage of this arrangement is that if a person's name should change, the information only needs to be updated in one table. A database might have many tables that need to tie records with a particular person. If the name were input into each of these, a name change would require updating many tables, which is likely to lead to errors. It is a common practice to give people employee or customer identification numbers which look more elaborate than our simple integers but serve the same purpose in the database.

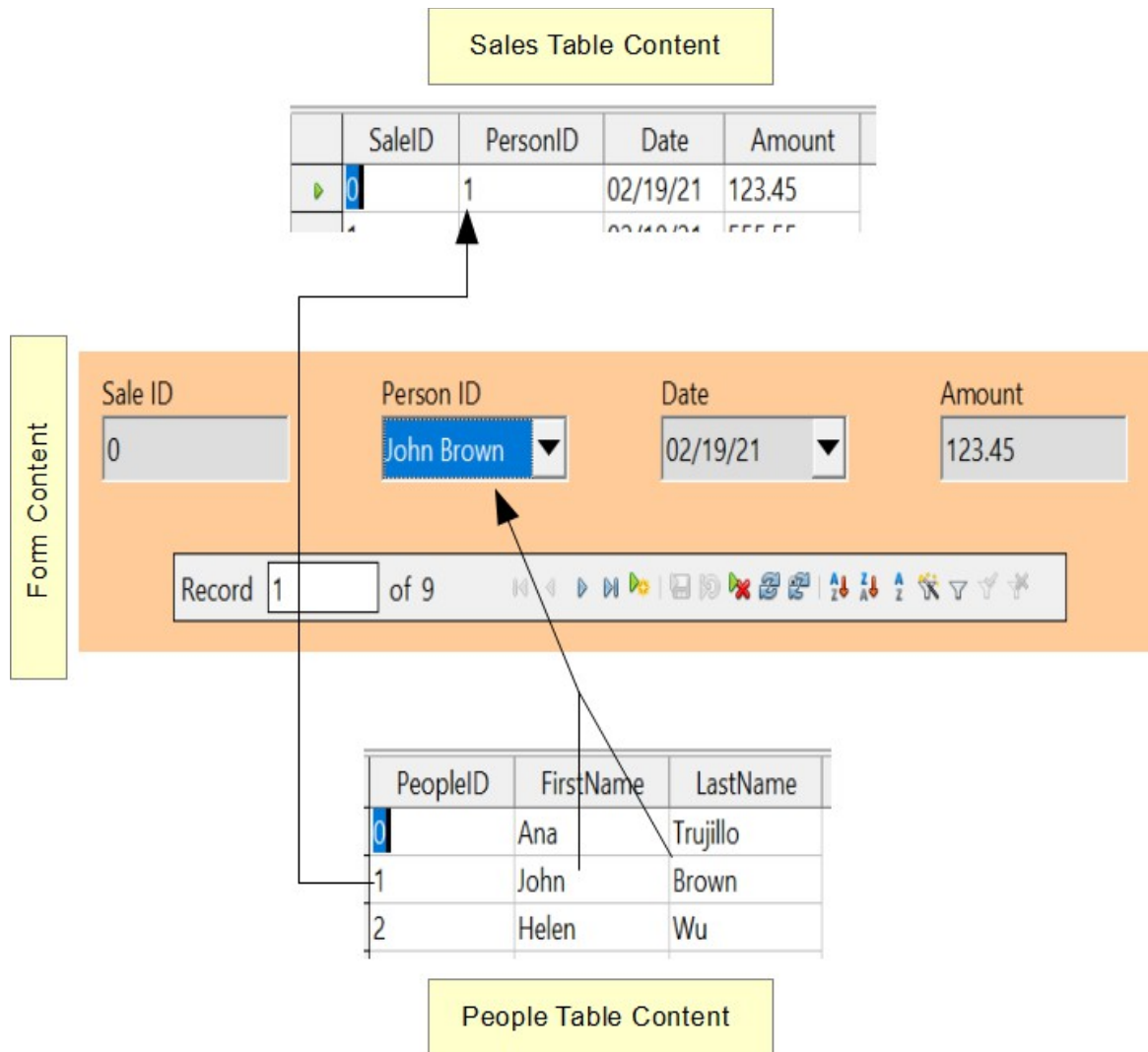


Figure 21: A List Box displays one value from the source table but sends another value to the target table.

Changing the Tab Order

The last change for usability is to adjust the tab order of the controls. In the very simple form we have made, this is not needed but in forms with many controls, it is much more convenient to jump from one control to the next with the tab key than to repeatedly click on the form. To adjust the tab order, open the *Form Design* toolbar with **View > Toolbars > Form Design**. The toolbar is shown in Figure 22 with the *Activation Order* icon highlighted in red. Click that icon to open a dialog showing all the controls in the order in which they will be chosen by the Tab key.



Figure 22: Form Design toolbar with Activation Order highlighted

The dialog, shown in Figure 23, has buttons to move controls up and down the list. The control names are the automatically assigned ones unless you have changed them while editing other control properties. The easiest way to make these names

meaningful is to give the database table fields meaningful names, which is good practice in any case. The list has the controls in a reasonable order but in a form with more fields, it is likely that the order of the fields in the database will not be convenient for data entry.

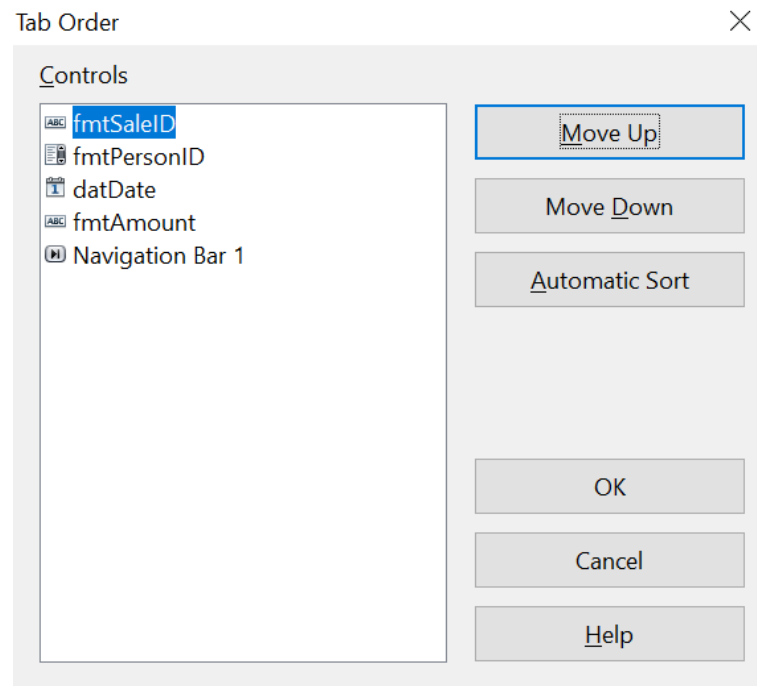


Figure 23: Tab order dialog

Creating Forms and Subforms in Design View

This method requires using the *Form Controls* and *Form Design* toolbars extensively. These techniques are beyond the scope of this document.

Accessing Other Data Sources

OpenOffice allows data sources to be accessed and then linked into AOO documents. For example, a mail merge links an external document containing a list of names and addresses into a letter, with one copy of the letter being generated for each entry.

To access an external data source:

- 1) **File > New > Database** opens the *Database Wizard* window.
- 2) Select *Connect to an existing database*. Click the arrow next to the *Database type* field and select the database type from the drop-down list. Click **Next**.
- 3) Click **Browse** and select the database. Click **Next**.
- 4) Accept the default settings: *Register the database for me*, and *Open the database for editing*. Click **Finish**. Name and save the Base file in the location of your choice.

Accessing a Spreadsheet as a Data Source

Accessing a spreadsheet is similar to accessing other databases:

- 1) Choose **File > New > Database**.

- 2) Select *Connect to an existing database*. Select *Spreadsheet* as the *Database type*. Click **Next**.
- 3) Click **Browse** to locate the spreadsheet you want to access. If the spreadsheet is password protected, check the *Password required* box. Click **Next**.
- 4) You can now choose to register the database, so that it is available to other OpenOffice documents, and to open the it for editing. Click **Next**.
- 5) Name and save the database file as you would any other document. If you have chosen to open it and it requires a password, you will see a prompt for the password.

Note

Using this method of accessing a spreadsheet, you cannot change anything in the spreadsheet. You can only view the contents of the spreadsheet, run queries, and create reports based upon the data already entered into the spreadsheet.

All changes in a spreadsheet must be made in the spreadsheet itself, using Calc. After modifying the spreadsheet and saving it, you will see the changes in the database. If you create and save an additional sheet in your spreadsheet, the database will have a new table the next time you access it.

Registering *.odb Databases

Registering a *.odb database is simple:

- 1) Choose **Tools > Options > OpenOffice Base > Databases**.
- 2) Under *Registered databases*, click **New**.
- 3) Browse to where the database is located.
- 4) Make sure the registered name is correct.
- 5) Click **OK**.

Using Data Sources in Apache OpenOffice

Having registered the data source, whether a spreadsheet, text document, external database or other accepted data source, you can use it in other AOO components including Writer and Calc.

Viewing Data Sources

Open a document in Writer or Calc. To view the data sources available, press *F4* or select **View > Data Sources** from the pull-down menu. This brings up a list of registered databases, which will include Bibliography and any other database registered, such as the database created earlier in this chapter.

To view each database, click on the **+** to the left of the database's name (see Figure 24). This brings up Tables and Queries. Click on the **+** next to Tables to view the individual tables created. Now click on a table to see all the records held in it.

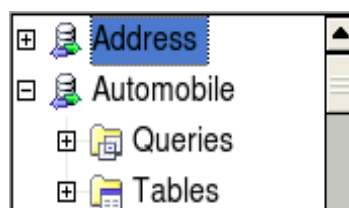


Figure 24: Databases

Editing Data Sources

Some data sources can be edited in the View Data Sources dialog. A spreadsheet can not. In other cases, a record can be edited, added, or deleted.

The data is displayed on the right side of the screen. Click in a field to edit the value.



Figure 25: View Data Sources navigation buttons

Beneath the records are five tiny buttons. The first four move backwards or forwards through the records, or to the beginning or end. The fifth button, with a small star, inserts a new record.

To delete a record, right-click on the gray box to the left of a row to highlight the entire row, and select **Delete Rows** to remove the selected row.

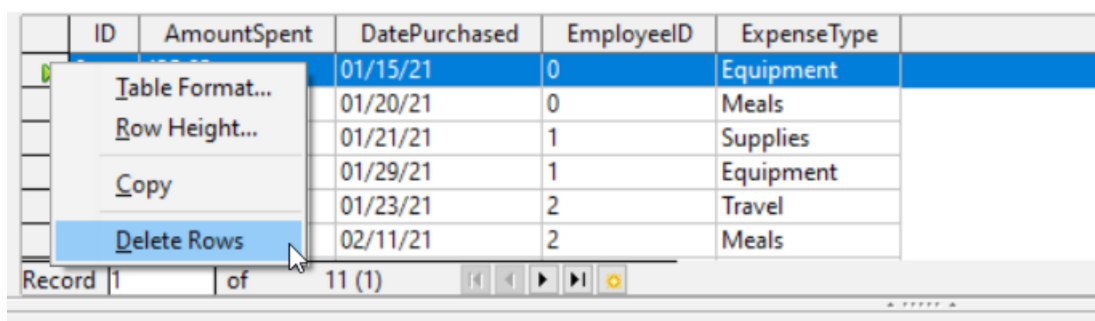


Figure 26: Deleting a row in the Data View window

Launching Base to Work on Data Sources

You can launch Base at any time from the View Data Source pane. Just right-click on a database or the Tables or Queries icons and select **Edit Database File**. Once in Base, you can edit, add, and delete tables, queries, forms, and reports.

Using Data Sources in Writer and Calc

Data can be placed into Writer and Calc documents from the tables in the data source window. In Writer, values from individual fields can be inserted. Or, a complete table can be created in the Writer document. One common way to use a data source is to perform a mail merge.

Tip

Choosing **Tools > Mail Merge Wizard** or clicking on the Mail Merge icon on the View Data Source pane launches the Mail Merge wizard which steps you through creating a mail merge document. See Chapter 11 in the Writer Guide.

Writer Documents

To insert a field from a table opened in the data source window into a Writer document, click on the field name (the gray square at the top of the field list) and, with the left mouse button held down, drag the field onto the document. In a Writer

document, it will appear as <FIELD> (where FIELD is the name of the field you dragged).

For example, to enter the information about an expense you can follow these steps.

- 1) Open the list of data sources by pressing *F4* and select the Finance database and its Expenses table.
- 2) Type the sentence in the Writer document to contain the information from the table, but do not write anything in the locations where database information will go. See Figure 27.
- 3) Click and hold on the column header of one of the desired fields, for example *DatePurchased*, and drag it to the desired location. Do this for each field. The text in the document will now look like
On <DatePurchased>, an expense of <AmountSpent> was submitted for <ExpenseType>.
- 4) Click on the gray box at the left end of the row holding the desired information.
- 5) Click on the *Data to Fields* icon, which is highlighted in red in Figure 27.

The values from the selected row will appear in the document. If **View > Field Shadings** is selected, the values will appear with gray shading. This shading does not print. If the record with ID = 9 was selected, the text in the Writer document will say:

On 02/02/21, an expense of 213.56 was submitted for Travel.

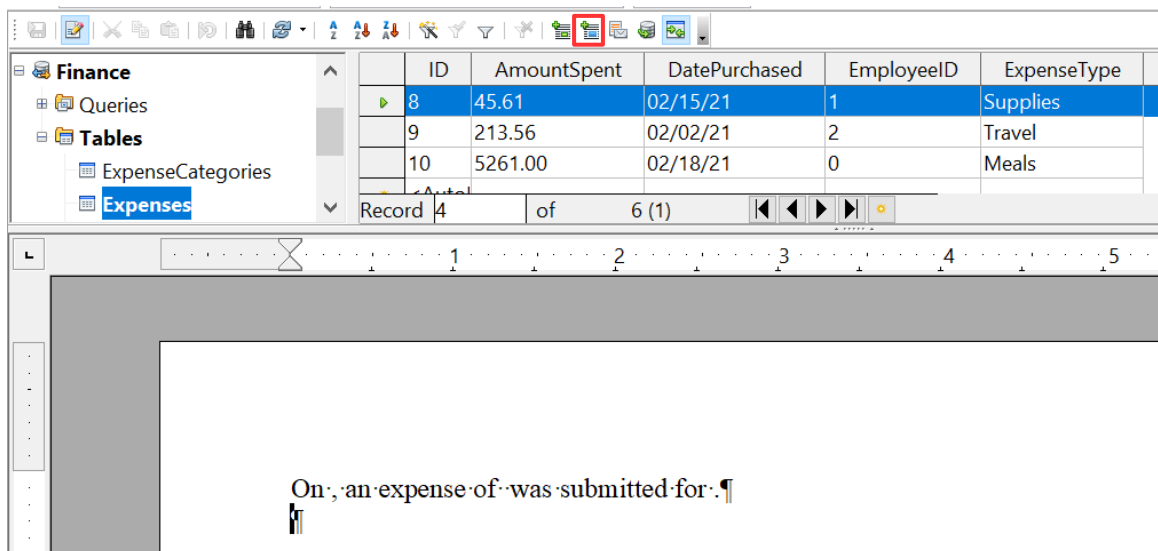


Figure 27: Sending data from Base to Writer with fields

Adding data in table format is a little easier and takes perhaps fewer steps. Some steps will be quite similar.

- 1) Navigate to the place you want to place the table and click the location.
- 2) *Ctrl+Click* the gray box to the left of each row of the data source that you want to be a row in your table when the rows are not consecutive. To select consecutive rows, click the gray box to the left of the top desired row and *Shift+Click* the bottom desired row.
- 3) Click the *Data to Text* icon to open the *Insert Database Columns* dialog. (The *Data to text* icon is to the left of the *Data to Fields* icon in Figure 27.

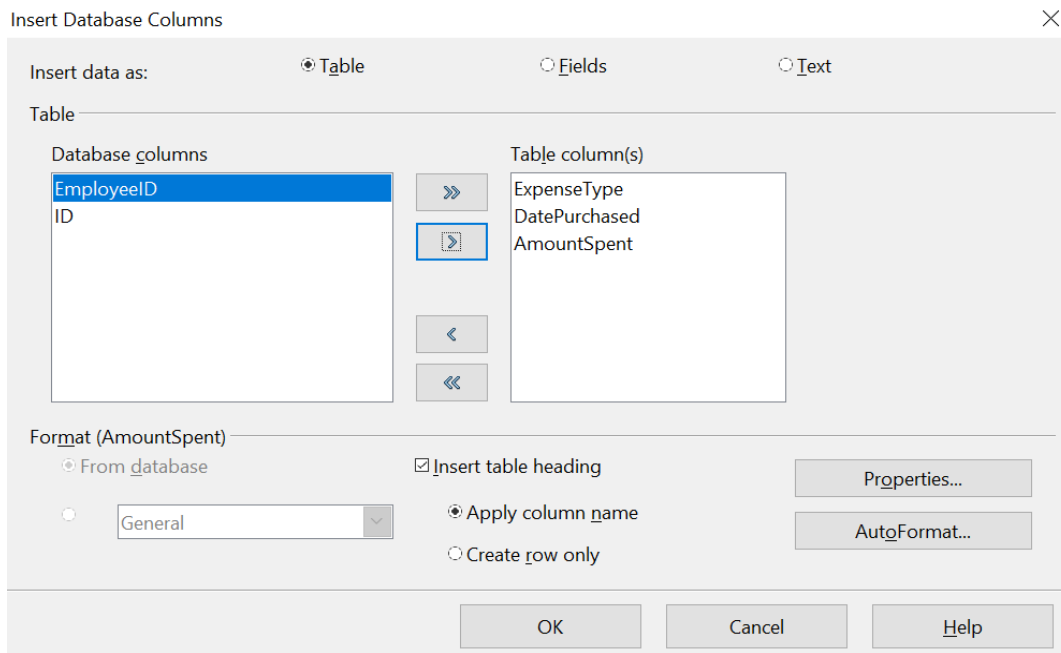


Figure 28: Insert Database Columns dialog

- 4) Move the fields you want in your table from the *Database columns* list to the *Table column(s)* list.
 - To place the fields in the order you select, click the field and use the *single arrow* to move the fields in the order you desire. You can also limit the fields you use to less than all the fields available.
 - If you want to use all the fields, use the *double arrow* pointing to the right to move all of them at one time. The order of the fields in the table you create will be the same as in the data source table.
 - To remove a single field from the *Table Column(s)* list, click the field and use the *single arrow* pointing to the left.
- 5) To start over, click the *double arrow* pointing to the left.
- 6) Select the settings for your table. Use the default settings as in Figure 28.
- 7) Click **OK**. Save the document.

Calc Spreadsheets

There are two ways to enter data in a Calc spreadsheet. One enters the data into the spreadsheet cells. The other creates controls to display the data in the spreadsheet just as is done in creating a form document in a Base file. While you can directly access the data in the spreadsheet cells, you can only see the data in the controls created in the spreadsheet.

Entering data directly to the spreadsheet cells uses the *Data to Text* icon as we did to make a table in a Writer document. But differences exist in these two situations.

The steps are straightforward:

- 1) Click the cell of the spreadsheet which you want to be the top left of your data including the column names.

- 2) Use *F4* to open the database source window and select the table whose data you want to use.
- 3) Select the rows of data you want to add to the spreadsheet:
 - Click the gray box to the left of the row you want to select if only selecting one row. That row is highlighted.
 - To select multiple rows, hold down the *Control* key while clicking the gray box of the rows you need. Those rows are highlighted.
 - To select all the rows, click the gray box in the upper left corner. All rows are highlighted.
- 4) Click the *Data to Text* icon to insert the data into the spreadsheet cells.
- 5) Save the spreadsheet.

Adding records displayed in form controls to a spreadsheet is fairly easy. You need to have the Data Source window open, your spreadsheet open, and the table you want to use selected.

- 1) Click the gray box above a field name.
- 2) Drag and drop the gray box for the field to where you want the record to appear in the spreadsheet.
- 3) Repeat until you have moved all the fields you need to where you want them.
- 4) Name and save the spreadsheet.
- 5) Click a row of the table in the Data Source window.
- 6) Drag the data, not the header, in the first field in the selected row onto its corresponding control in the spreadsheet. The Save icon should activate.
- 7) Click the *Edit File* button to make the spreadsheet read only. Click **Save** when asked if you want to save the file.
- 8) The Data Source window goes blank, the fields in the spreadsheet are populated with data from the row you selected, and the *Form Navigation* toolbar appears at the bottom of the spreadsheet.



Figure 29: Navigation arrows of a form

Click the arrows on the Form Navigation toolbar to view the different records of the table. (The arrows are circled in red.) The number in the box changes when you change the record number by clicking an arrow. The data in the fields changes correspondingly to the data for that particular record number.

Creating Queries

Queries are used to get specific information from a database. Query results are special tables within the database, but the results are not permanently stored; only the code to produce the results is stored.

To demonstrate the use of queries, we will use both the query Wizard and query Design View to look at the Sales data from the Finance database we made in previous sections.

Using the Wizard to Create a Query

Queries created by the wizard provide a list or lists of information based upon what one wants to know. It is possible to obtain all the records meeting the desired conditions or aggregated results such as the sum of data from multiple records.

In the main database window (Figure 1), click the Queries icon in the Databases section, then in the Tasks section, click *Use Wizard to Create Query*. The Query Wizard window opens (Figure 30). We want to display the *PersonID*, the date and the amount of each sale, sorted by the date and including only sales with an Amount greater than or equal to 100.

Note

When working with a query, more than one table can be used. Since different tables may contain the same field names, the format for naming fields in a query is *Table name.field name*, with a period (.) between the table name and the field name. For example, the Date field of the Sales table used in a query has the name *Sales.Date*.

Step 1: Select the fields

- 1) Select the Sales table from the dropdown list of tables.
- 2) Select fields in the *Available fields* list. This step has already been done in the figure.
 - Click *PersonID*, and use the > button to move it to the *Fields in the Query* list.
 - Move the *Date* and *Amount* fields in the same manner.
 - Click **Next**.

Tip

To change the order of the fields, select the field you want to move and click the up or down arrow.

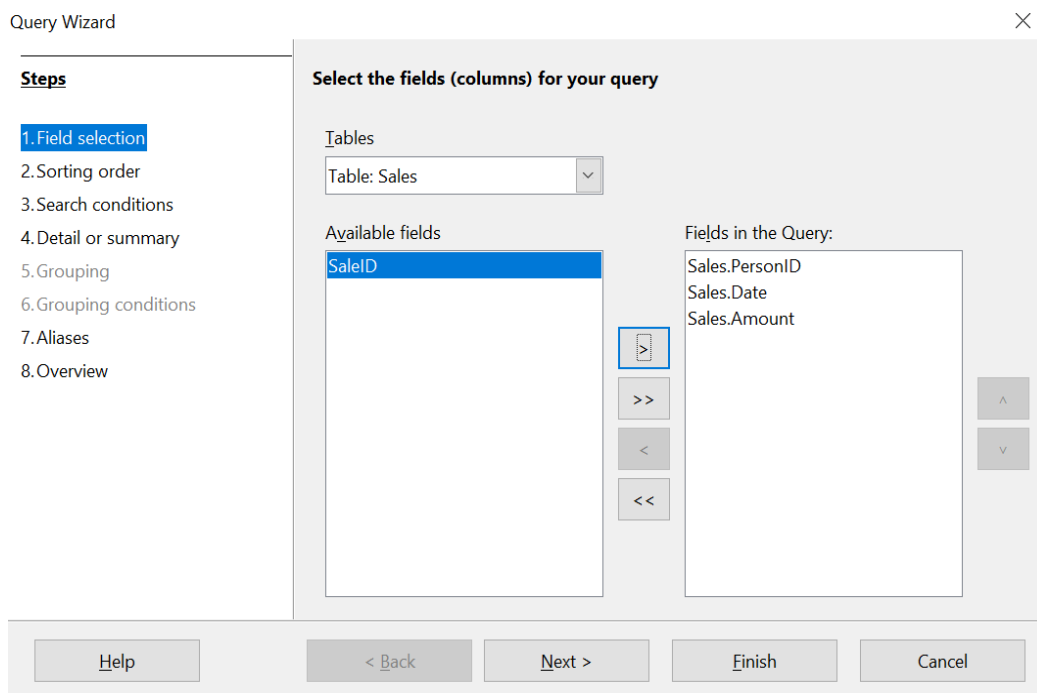


Figure 30: First page of the Query Wizard

Step 2: Select the sorting order

The Wizard allows up to four fields to be used to sort the information of our query. The query language, SQL, does not have a limit on the number of fields that can be used for sorting. In our case, we only want to sort by the date. Set the *Sort by* list to *Sales.Date*, leave the *Ascending* button selected and click **Next**. Figure 31 shows the dialog just before clicking Next.

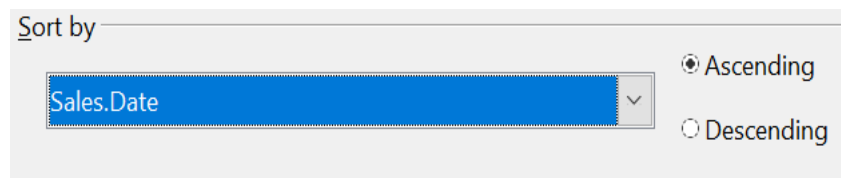


Figure 31: Sorting order page

Step 3: Select the search conditions

The search conditions allow us to compare the field values with desired values and decide whether to include a particular record in our query result or not. Among the choices are:

- *is equal to*
- *is not equal to*
- *is smaller than*
- *is greater than*
- *is equal or less than*
- *is equal or greater than*
- *like* (used for wildcard matching of text)

Note These conditions apply to numbers, letters (using the alphabetical order), and dates.

The conditions can be related to each other with AND, so that all the conditions have to be met, or with OR, so that only one condition need be met. We want to limit the records to amounts greater than or equal to 100.

- 1) Since we are only setting one condition, we will use the default setting of *Match all of the following*.
- 2) We are looking for data beyond a certain value. Select *Sales.Amount* in the *Fields* list and *is equal or greater than* as the *Condition*.
- 3) Set the Value box to 100. Click **Next**.

Step 4: Select type of query

We want all the records that meet the criterion returned, so the default setting: *Detailed query* is what we want. Click **Next** at the bottom of the window.

Note Since we have a simple query, the Grouping and Grouping conditions are not needed. Steps 5 and 6 of the wizard are skipped in our query.

Step 7: Assign aliases if desired

We use the default settings, but it is often useful to give descriptive names at this point. Click **Next**.

Step 8: Overview

Name the query (suggestion: *SalesAbove100*). To the right of this are two choices. Select *Display Query*. Click **Finish**. You will now see a display of the selected rows in the same style that database tables are displayed.

Using the Design View to Create a Query

Make a Basic Query

Creating a query using Design View is not difficult but does require learning the conventions of the tool. We will start with a duplicate of the query from the last section and then make some more complicated queries.

Our first query will display all the sales with an amount greater than or equal to 100, sorted by date.

Step 1: Open the query in Design View

Click **Create Query in Design View** in the Query's pane of the main Base window.

Step 2: Add tables

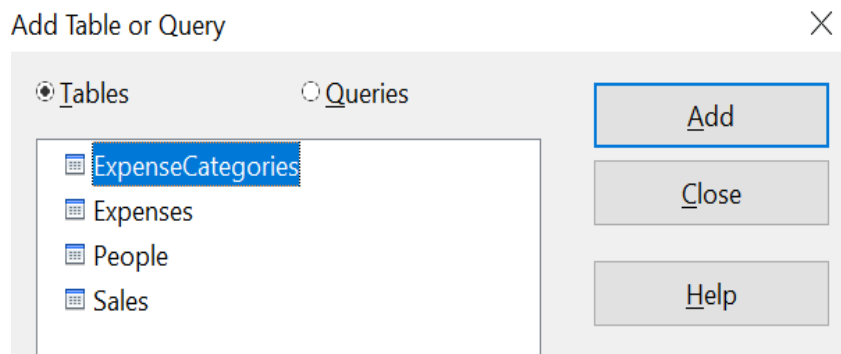


Figure 32: Add Tables or Query window

- 1) Click *Sales* to highlight it.
- 2) Click **Add**. Click **Close**.

Tip

Move the cursor over the bottom edge of the *Sales* table (Figure 33) and drag it to make it longer and easier to see all the fields in the table.

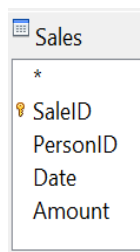


Figure 33: Sales table in query

Step 3: Add fields to the query definition table at the bottom

Double-click the *PersonID*, *Date*, and *Amount* fields to add them to the query definition table.

The table at the bottom of the query window should now have three columns populated. The row labeled *Visible* is used to toggle the visibility of the field in the final query result. The default is to show all the fields.

Field	PersonID	Date	Amount
Alias			
Table	Sales	Sales	Sales
Sort			
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function			
Criterion			

Figure 34: Query definition table

Step 4: Set the criterion for the query

We want the query results to have an *Amount* greater than or equal to 100 and the dates to be sorted in ascending order.

- 1) Type **>= 100** in the Criterion cell under *Amount* in the query definition table.
- 2) Choose *ascending* from the dropdown list in the *Sort* row of the *Date* column.
- 3) Click the *Run Query* icon in the Query Design toolbar, which is highlighted in red in Figure 35.



Figure 35: Query Design toolbar

Step 5: Save and close the query

Save the query and also save the main Base document.

Make a Query With a Calculated Field

Imagine you would like to calculate a commission for each sale that is 7% of the sale amount. The process is just like the last query except that at step 3 an additional column is added manually to the query description, as shown in Figure 36. Notice the fourth column where the *Field* is defined as "Amount" * 0.07. The name of the column upon which the calculation is based should be enclosed in double quotes. Besides the usual arithmetical operators of +, -, *, and /, many other functions such as square root (sqrt) and the trigonometric functions are available. The calculations may also use more than one column, so you could, for example, calculate the difference or ratio of two columns.

Some other details to note in the calculated column are that it has the text *Commission* written in its *Alias* row, and that the *Table* row is blank. These changes should make sense. The column is not stored in a table, and therefore it does not have a good default name. If the alias is not supplied, the text of the calculation is used, which is not often a helpful column header. Note also that the criterion in the *Amount* column has been removed. This was not required; it is perfectly possible to do the calculation only on rows that meet any number of criteria.

An important point in database design is that calculations based on field values should be done in queries and not stored in the regular database tables. The main reason for this is that if the underlying values are updated, the calculated values in a regular table will not be automatically updated. You have to explicitly recalculate them. In a query, on the other hand, only the code to run the query is stored, so any calculated values are updated every time the query is run.

Field	PersonID	Date	Amount	"Amount" * 0.07
Alias				Commission
Table	Sales	Sales	Sales	
Sort				
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function				
Criterion				

Figure 36: A query with a calculated field

Query with Two Related Tables

An obvious deficiency in the previous queries is that the people associated with an expense or a sale are only identified by the value of the *PeopleID* field in the *People* table. Adding their names to the query output is simple because we have established relations between the *People* table and the *Expenses* and *Sales* tables. We will make a new query that displays sales information along with the name of the associated person.

Step 1: Start the query in Design View

From the main Base window, click *Queries* in the left pane and then *Create Query in Design View* from the Tasks pane.

Step 2: Add tables to the query

In the *Add Query or Table* dialog, click on *People* and then the *Add* button and do the same for the *Sales* table. Click **Close**. You will now have small representations of the two tables with a line running from the *PeopleID* field of one to the *PersonID* field of the other, as shown in Figure 37. You may want to resize the tables by dragging down their lower edges.

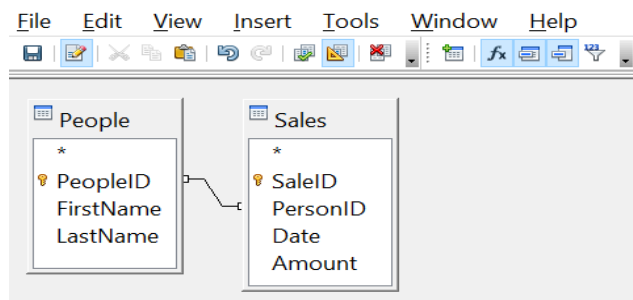


Figure 37: The *People* and *Sales* tables in a query

Step 3: Add fields to the query

Double-click on the *FirstName* and *LastName* from the *People* table and *Date* and *Amount* from the *Sales* table to add them to the query definition table.

We now have a query that will display the names along with the information from the *Sales* table. Save both the query and the main Base document. The query can be enhanced with criteria in any column or with calculated columns as shown in the previous sections.

Query That Summarizes Data

All the queries we have written so far display all the rows that meet the query criteria. It is very common to want a summary of the data that displays quantities like the sum or the average of a field calculated over some group of rows. For example, we might want the sum of sales for each person. Writing such a query can be done using the *Function* row of the query definition table in Design View.

Step 1: Start the query in Design View

From the main Base window, click *Queries* in the left pane and then *Create Query in Design View* from the Tasks pane.

Step 2: Add tables to the query

In the *Add Query or Table* dialog, click on *People* and then the *Add* button and do the same for the *Sales* table. Click **Close**.

Step 3: Add fields to the query

Double-click on the *FirstName* and *LastName* from the *People* table and *Amount* from the *Sales* table to add them to the query definition table.

Step 4: Set the grouping and the sum function

In the *Function* row of the query definition table, use the drop-down list to set *FirstName* and *LastName* to *Group* and set *Amount* to *Sum*. Also set the *Alias* of the *Amount* column to *Sales_Sum*. The final settings of the query definition table are shown in Figure 38.

Field	FirstName	LastName	Amount
Alias			Sales_Sum
Table	People	People	Sales
Sort			
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function	Group	Group	Sum

Figure 38: Query summing Amount for each person

Note

In a query that uses one or more summarizing functions, also known as aggregating functions, every column must have an entry in the *Function* row, either as *Group* or one of the other functions.

Query Summarizing Over a Calculated Field

As a final example of a query, we will calculate the total sales of each person for each month and year. We will need to use the *Date* column from the *Sales* table twice, one to calculate the year and again to calculate the month, and we will also use the fields *FirstName*, *LastName* from the *People* table and *Amount* from the *Sales* table.

Step 1: Start the query in Design View

From the main Base window, click *Queries* in the left pane and then *Create Query in Design View* from the Tasks pane.

Step 2: Add tables to the query

In the *Add Query or Table* dialog, click on *People* and then the *Add* button and do the same for the *Sales* table. Click **Close**.

Step 3: Add fields to the query

Double-click on the *Date* field from the *Sales* table twice, then double-click on *FirstName* and *LastName* from the *People* table and *Amount* from the *Sales* table to add them to the query definition table.

Step 4: Set the calculation of the year and the month

Change the *Field* row of the first *Date* column to read *YEAR("Date")* and change the *Field* row of the second *Date* column to read *MONTH("Date")*. Set the *Alias* row to read *Sale_Year* and *Sale_Month* for these two columns. The *YEAR* and *MONTH* functions are among the many functions provided by the HSQL database engine. A complete list is available in the documentation at <https://www.hsqldb.org/doc/1.8/guide/guide.html#N1251E>.

Step 5: Set the grouping and the sum function

Set the *Function* row of every field except *Amount* to read *Group*. Set the *Function* row of the *Amount* column to read *Sum* and set its alias to *Sales_Sum*.

Step 6: Set the sorting

For a more logical presentation of the data, set the *Sort* column of the year, month and *LastName* columns to read *ascending*.

Be sure to save the query and the main Base document. We saved the query with the name *GrpCalcColumn*, and we will use it in the reports in the next section.

The final configuration of the query definition table and the query results are shown in Figure 39.

Field	YEAR("Date")	MONTH("Date")	FirstName	LastName	Amount
Alias	Sale_Year	Sale_Month			Sales_Sum
Table			People	People	Sales
Sort	ascending	ascending		ascending	
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function	Group	Group	Group	Group	Sum

Sale_Year	Sale_Month	FirstName	LastName	Sales_Sum
2021	1	John	Brown	402.65
2021	1	Ana	Trujillo	542.12
2021	2	John	Brown	123.45
2021	2	Ana	Trujillo	799.22
2021	2	Helen	Wu	1557.68

Figure 39: Definition and result of a query that groups on a calculated column

Creating Reports

Reports provide information found in the database in a useful way. In this, they are similar to queries. Reports are generated from the database's tables or queries. They can contain all the fields of the table or query, or just a selected group of fields. Reports can be static or dynamic. Static reports contain the data in the selected fields at the time the report was created. Dynamic reports can be updated to show the latest data.

For example, a report on expenses in the past should probably be a static report because it is based upon specific data that does not change. However, a report on year-to-date expenses should probably be a dynamic report because this report depends upon data that does change.

All reports are based upon a single table or query, and you need first to decide what fields you want to use in the report. If you intend to use fields from different tables, you must first combine these fields in a single query. Then you can create a report on this query.

The most flexible method for making a report is to write a query that returns the desired columns and sorts the rows, and then save this query as a View in the database. A View is simply a query whose code has been stored alongside the database tables, and it can be treated just like a table. To make a *View* from a query, click on the *Queries* icon in the main Base window, then right-click on the name of the specific query and select *Create as View*. You will be prompted for a name to give the View. A good practice is to give the View the same name as the query, with View as either a prefix or suffix. For the View made from the *GrpCalcColumn* query that we will use in the reports, we used the name *GrpCalcColumn_View*. This makes it easy to see the code saved in the View. Keep in mind that editing the query after saving the View does not change the View. You would have to make a new View to embody the new query. Once the View is saved, it will be listed along with the tables in the database.

Once a View is ready for the report, making the report is very simple.

Simple Static Report

Step 1: Start the Report Wizard

On the main Base window, click the *Reports* icon and then click *Use Wizard to Create Report*.

Step 2: Choose the fields from the View

Choose the View you want to use from the *Tables or queries* dropdown list. The available fields will be listed. Use the >> button to move all of them to the *Fields in report* list. You can just move a subset of the fields by selecting them individually and using the > button. The fields can also be reordered with the up and down arrow buttons. Click **Next**.

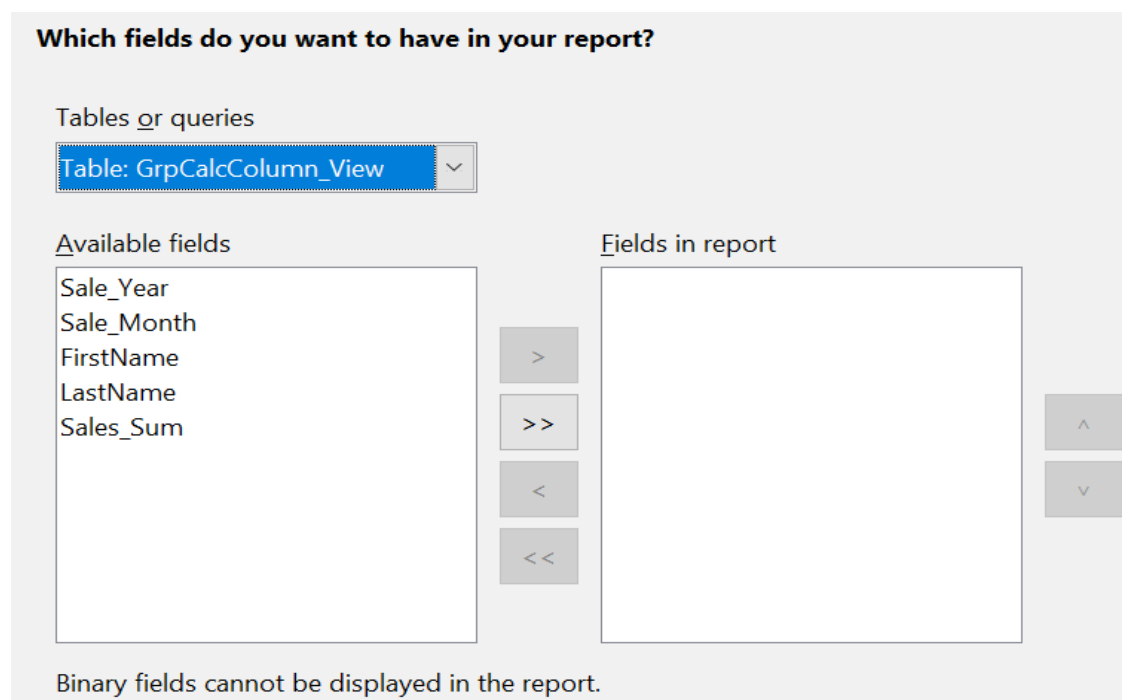


Figure 40: Dialog for choose the fields in a report.

Step 3: Labeling the fields

Here you can modify the field labels from their default values. We will remove the underscores from the field names and add spaces so that Sale_Year becomes Sale Year and LastName becomes Last Name. Click **Next**.

Step 4: Grouping levels

In this first report we will not add grouping levels. In the next iteration of the report, we will do that. Click **Next**.

Step 5: Sort options

It is better to sort in the View rather than using the Report Wizard's sort. Click **Next**.

Step 6: Choose layout

There are many layout options. We will leave the defaults in place, but you can explore alternatives. A mock version of the report is visible behind the Report Wizard dialog and if you drag the dialog low in the screen, you can see the effect of different layouts. Click **Next**.

Step 7: Create report

In this step, you give the report file a name, rather misleadingly called *Title of report* and choose whether the report is static or dynamic. We will select static for this version. Click **Finish**.

We now have a Writer file that looks like Figure 41.

Title:

Author: f c

Date: 2/24/21

<i>Sale Year</i>	<i>Sale Month</i>	<i>First Name</i>	<i>Last Name</i>	<i>Sales Sum</i>
2021	1	John	Brown	402.65
2021	1	Ana	Trujillo	542.12
2021	2	John	Brown	123.45
2021	2	Ana	Trujillo	799.22
2021	2	Helen	Wu	2079.35

Figure 41: Static report

The Author field is taken from **Tools > Options > OpenOffice > User Data > First/Last name**. All the information is locked and not easily edited. It can be copied to a new document and edited there. If you do that, the Title field can be filled out in the new document by selecting the menu **File > Properties** and filling out the *Title* on the *Description* tab.

Grouped Dynamic Report

We will now change the report in two ways. We will make it dynamic, and we will add some Grouping Levels. Keep in mind that these two are independent. The Grouping Levels could have been added to the static report in the last section.

Step 1: Start the Report Wizard

On the main Base window, click the *Reports* icon and then click *Use Wizard to Create Report*.

Step 2: Choose the fields from the View

Choose the View you want to use from the *Tables or queries* dropdown list. The available fields will be listed. Use the >> button to move all of them to the *Fields in report* list. Click **Next**.

Step 3: Labeling the fields

Here you can modify the field labels from their default values. We will remove the underscores from the field names and add spaces so that *Sale_Year* becomes *Sale Year* and *LastName* becomes *Last Name*. Click **Next**.

Step 4: Grouping levels

Use the > button to move the *Sale_Year* and *Sale_Month*, in that order, from the *Fields* to the *Groupings* column. Click **Next**.

Step 5: Sort options

Note that *Sale_Year* and *Sale_Month* are now listed as sort fields. Click **Next**.

Step 6: Choose layout

We will keep the defaults but, as mentioned in the previous section, you can try many alternatives. Click **Next**.

Step 7: Create report

Set the name of the report file, choose *Dynamic report* and *Create report now*. Click **Finish**.

The top part of the report looks much like Figure 41, but the data table now has a hierarchical structure with the year being the outermost category, then the month and then the data for each person's monthly sales in a table.

The report layout can be edited by closing the report then right-clicking on its icon and selecting Edit. The report will appear filled with dummy data. The Title field can be populated by selecting the menu **File > Properties** and filling out the *Title* on the *Description* tab. The Author field is taken from **Tools > Options > OpenOffice > User Data > First/Last name**. The date format in the header area can be adjusted by right-clicking on it and selecting Fields. A dialog with a long list of formats will appear. Choose one then click *OK*. In the data table, the column widths can be adjusted by hovering the cursor over a border until it becomes a double-headed arrow then clicking and dragging it left or right.

When you are done changing the format, save the document and also save the main Base document. If you double-click on the report now, you will see the format changes along with the actual data from the View. Compare Figure 42 to Figure 41 to see the effect of the groupings by the *Sale Year* and *Sale Month*. Remember that the grouping could have been done in the static report. Since this report is dynamic, if data is added or deleted from the People and Sales tables, the report will reflect the changes.

Title: Sales Report by Month
 Author: f c
 Date: February 24, 2021

Sale Year		2021		
Sale Month		1		
	<i>First Name</i>	<i>Last Name</i>	<i>Sales Sum</i>	
	John	Brown	402.65	
	Ana	Trujillo	542.12	
Sale Month		2		
	<i>First Name</i>	<i>Last Name</i>	<i>Sales Sum</i>	
	John	Brown	123.45	
	Ana	Trujillo	799.22	
	Helen	Wu	2079.35	

Figure 42: Dynamic report with grouped data