

Database

A *database* is a collection of information or *data*. At its simplest, a database can be a list of names and phone numbers, forming a personal phone book. Ability stores such lists in *tables*.

Tables can be:

- combined to form *relations*
- arranged in various ways to create *queries*

Queries are themselves formed from *query components*:

- *Sort Orders*
- *Column Views*
- *Filters*

In turn, you can *view* a table, relation or query in one of three ways:

- *Grid view*
- *Form view*
- *Report view*

A single database then, is a collection of one or more tables and all the associated queries and relations, which can be viewed in grids, forms, and reports. The Ability database provides an easy environment to manage all your data requirements.

The following sections explain the meaning of basic database terminology in greater depth.

See:

[What is a table?](#)

[What is a relation?](#)

[What is a query component?](#)

[What is a query?](#)

[What is a table view?](#)

[Starting Database](#)

What is a table?

Tables are the fundamental database objects. It is here that your data is stored.

A table is arranged in *rows* and *columns*.

A table may be used, for instance, to store the names and associated phone numbers from your personal phone book. Each row will contain all the data for one entry in your phone book, and is called a *record*. Each column will contain one item of data for all entries in your phone book, and is called a *field*.

Since tables are the fundamental database objects, deciding what tables you need is always the first step in designing a database.

Tables are used to contain information and data that can be structured according to a similar format. Typical examples of tables are:

- mailing lists
- inventory lists
- customer lists
- product lists
- medical records
- league tables

For more information on tables, see [Creating fields in a database](#).

See also:

[What is a table view?](#)

What is a relation?

Tables may be related to one another - an invoice table naturally combines information from a customer table and a product table. In this case, the invoice table is called a *relation*. You can create a relation from selected fields from related tables. The relation then becomes a table in its own right, albeit dependent on the tables from which it is formed.

- For more information on relations, see [Creating a relation](#).

What is a query component?

You can arrange the data from a table in a number of ways:

- *Sort Orders* - sort the table by one or more columns
- *Column Views* - choose the columns to display and their display order
- *Filters* - define conditions to include only specified records

Ability calls these different arrangements of your data *query components*. You can use the query components individually or collectively in *queries*.

For more information on query components, see [Query components](#).

See also:

[What is a query?](#)

What is a query?

A *query* is a table that has been constructed out of one or more query components when these have been applied to an underlying table. Queries enable you to combine various sort orders, column views and filters, which can all be applied at once to a specified table.

- For more information on queries, see [Queries](#).

What is a table view?

Whether you're working with a table or query or relation, you can choose between three different on-screen views:

- *grid view* - displays the table in a grid with rows and columns
- *form view* - each record occupies the entire window for easy editing and allows for custom layout of the screen
- *report view* - the data is arranged in a format suitable for printing, and also allows for a custom layout.

Grid view can be regarded as the natural way to view a table, since we usually think of tables as being arranged into rows and columns. Accordingly the term *table* will often be used to refer to a table when in it is in grid view. Nevertheless the table itself is distinct from any of the ways of viewing it.

For more information on the different ways of viewing your tables, see

[Grid view](#)

[Form view](#)

[Reporting](#)

Starting Database

To start Database, do one of the following:

- Click on the **Start** button, select **Programs**, **Ability Office** and then **Ability Database**.
- Click on the **Database** button on the toolbar of any open Ability application.
- Click on **Database** in the **Ability Launcher**. For further details, see [The Ability Launcher](#).


The Ability Database window is opened.

See:

[Opening a database](#).

Opening a database

To open a database, follow these steps:

1. Click the **Open** button  on the toolbar (or select **Open Database** from the **File** menu).
2. Make sure **Files of type** shows **Ability Database Files (*.adb)**
3. Use the **Look in** box to locate the folder in which the database is contained. Once the folder has appeared in the main box, open it by double clicking (or clicking on **Open**).
4. Open the database itself by double clicking on the database name (or clicking on **Open**).
5. Ability will open the database and display the **Database Manager**.

See:

[Database Manager](#)

Database Manager

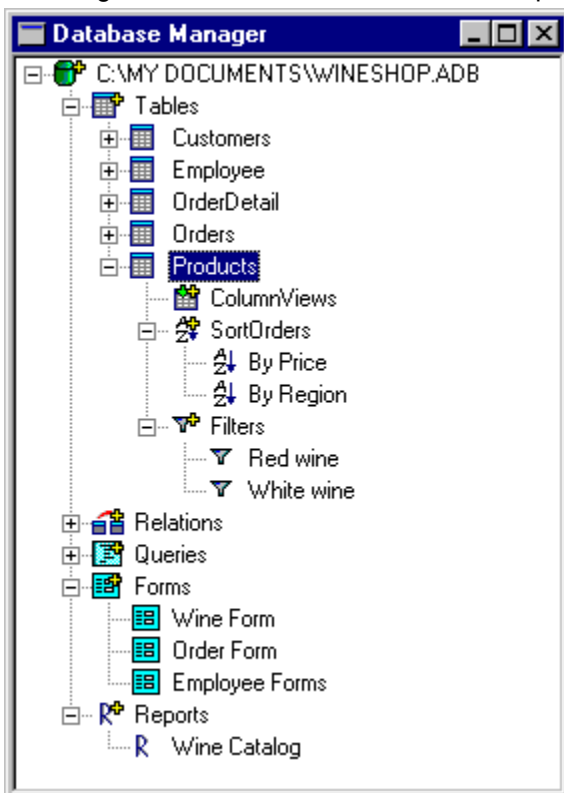
Whenever you open a database, the database is displayed in **Database Manager**. Database Manager is where you manage all the objects in your database, that is to say, all the tables, relations, queries, forms and reports that make up the database. You can use it to create, delete, restructure and open all of these database objects.

In addition, you can import (see [Importing tables](#)) and export tables (see [Exporting tables](#)) while in Database Manager.

To view the Database Manager, should some other window obscure it, select **Database Manager** from the **View** menu.

Database Manager lists all the database objects in a tree structure - individual branches can be opened and closed, revealing more detail as required.

To open a branch, click on the '+' sign before the database object name. '+' will change to '-', and back again with a further click. Here's an expanded view of a sample database:



See:

[Opening a table using database manager](#)

[Closing a database](#)

Closing a database

To close a database you should first close any tables you are working in. This will return you to **Database Manager**. Close the database by selecting **Close Database** from the **File** menu. Once the database is closed the command will be grayed out.

Some commands require you to close a database before they will work, for instance, **Repair Database**.

You can close the **Database Manager** screen by clicking on the button at the top left of the Database Manager and selecting **Close** from the menu. This merely closes the Database Manager, not the database itself. To display the Database Manager again, select **Database Manager** from the **View** menu.

Query components

A query component is a particular arrangement that you apply to a table. The query components are of the following type:

- **sort order**
- **column view**
- **filter**

They are always attached to a table and are listed beneath each table in Database Manager. Using the query toolbar, you can always choose combinations of query components wherever you make use of the table - this could be in a form or report, or in Write and Spreadsheet.

Relations can have associated query components in the same way.

Imagine a *Wineshop* database that contains several tables. Each table can have several named entries underneath - the query components. These are grouped according to whether they are **Column Views**, **Sort Orders** or **Filters**. For example, an *Employee* table might show a Column View - "Name & Address", two Sort Orders - "By Age" and "By Name", and a Filter - "UK Employees". These are all different ways of viewing the data contained in the *Employee* table.

You can use combinations of these query components to view only certain columns from selected records of a table, and in a particular order. For example, you can filter the *Employee* table using "UK Employees", viewing only the columns defined by "Name & Address", in an order determined by "By Name". You can also save combinations of query components as a named **Query**, which can then be applied whenever you like (see [Queries](#)).

See the following sections:

[Rearranging columns using a columns selection](#)

[Hiding and showing columns using a columns selection](#)

[Sorting using a sort order](#)

[Filtering using a named filter](#)

Opening a table


To open a table, do one of the following in Database Manager:

- Double-click the table name to make all rows and columns available
- Double-click a named query component belonging to a table to open the table and apply the selected sort order, column view or filter
- Right click on the table name or named query component belonging to the table and select **Open**.


Moving around a table


There are four ways to move around the table:

- Use the mouse to click into any visible field directly.
- Use the horizontal and vertical scrollbars - with them you can reveal any part of the table, but the cursor won't reposition until you click into a field.
- Use the **Record** toolbar above the top of the table window. There are buttons for moving to the first

, last

, next

 and previous


 records. The same options are available from the **Record** menu.

- Use the **Record** scroll bar at the bottom left of the table window.
- Use the keyboard as follows:

Key	Description
up, down, left & right arrow keys	The arrow keys move the cursor one field at a time in the direction of the arrow
home	Edit the current field and move to the beginning of the field.
end	Edit the current field and move to the end of the field.
ctrl + left	Moves the cursor to the first field in the current record.
ctrl + right	Moves the cursor to the last field in the current record.
ctrl + up	Moves the cursor to the first record in the current field.
ctrl + down	Moves the cursor to the last record in the current field.
ctrl + home	Moves the cursor to the first record and field.
ctrl + end	Moves the cursor to the last record and field.
tab	Moves the cursor one field to the right.
shift + tab	Moves the cursor one field to the left.

Creating a new database

To create a new database:

1. Select **New Database** from the **File** menu or click on the **New** button  on the toolbar.
2. The "Create a new database" dialog is displayed. The **Save as type** box should automatically be set at **Ability Database Files (*.adb)**. In the **File name** box, type the name of the new database. Don't worry about a file extension - this will automatically be added by Ability.
3. Before clicking the **Save** button, select the folder you want to save to in the **Save in** box.

After saving, the dialog is closed and the [Database Manager](#) window is displayed. A database tree is created for the new database and you are now free to create the tables, queries, relations, reports and forms that are to go into it.

See:

[Designing a new table.](#)

Designing a new table

To create a new table, do one of the following:

- Right-click anywhere in the Database Manager window (see [Database Manager](#)) and select **New** and then **Table** from the displayed menu.
- Select **New** and then **Table** from the **File** menu.

The "New Table" dialog is displayed. This contains a **Table Wizard** option and a **Blank Table** option. The Table Wizard enables you to choose pre-designed fields from a selection of tables that come as part of the Ability system (the system database is called *abdata.adb* and is the equivalent of a database template). See [Creating fields in a database using Table Wizard](#) for more information on using Table Wizard.

1. Click on the **Blank Table** button. The "New table" design window is displayed.

The design window is where you create the structure of your table. This is the framework in which you will store your data, so it is worth spending time at this stage planning how you will use the table and deciding what fields you will need (see [Planning field details](#)).

Each table will contain one or more fields. For each field you need to assign

2. a name in **Name** (see [Field names](#)) and a data type in **Type** (see [Field data types](#)), these being compulsory.

In addition, each field can be given

3. an optional summary description in **Description**.

After setting these you can format a field to give it a caption (see [Captions](#)) and a default value, these being optional. The fields can also be formatted for such properties as width, validation, justification, number type, etc. depending on the data type of the field..

For full details of how to create the fields for a new table, see [Creating fields in a database](#).

Planning field details

- It is very important that you plan your table design in accordance with the table's intended purpose. This sounds too obvious – and it is, which is why it is so often overlooked. Think in terms of the information you will want to be able to extract from the table in a week's time, a month's time, and a year's time.
- If it is likely that you will not be the only user of the table, talk to others who may be expected to produce useful, efficient work with your design. They may operate in slightly different ways to you and their experience will be a useful supplement to your own. Basically, a clear idea of the potential functions of your database should guide you in its design.
- It is recommended that you sketch out the design for your table on paper, in row and column layout, with field names at the head of each column. Pencil in a few rows of data to confirm that you have made allowance for all the relevant data items you need. If not, just add some more columns and field names in a second sketch, and repeat the process until you have established fields for all required data items.
- Consider also, at this point, the order of the columns. If you have a large number of fields, it is usually better to keep all the most commonly accessed columns to the left of the layout. In this way you will minimize the need to scroll the table window left and right to enter, edit or view your most important data. You will however be using *Query* files to see many different views of your data, each with only the columns you want, arranged in the order you require for a particular purpose.
- Once you are satisfied with the overall layout of your table, and have allocated meaningful field names to your columns, you need to determine the properties of each field. Typically these will include the type of data that a field is to contain, such as text or numbers; whether the records in your table are to contain any default data; the width of the fields; and so on.

See [Creating fields in a database](#).

Field names

Each item of information or data you wish to store, manipulate or extract using Ability, must be uniquely accessible by the entire Ability system. Ability does this by calling on the address of the data, whose location is identified by the database title, table name, field name and record number.

You specify the field name in the **Name** cell. It is usually helpful if it is easily identifiable by you as well, a year or two after you created the table, so make the name relatively descriptive. For example:

first_name

last_name

The underscore character is often used as a link between words. These examples could be reduced in length without losing any clarity. For example:

FirstName and **LastName**

or

fname and **lname**

This will save typing effort if later on you use these field names in computed fields or in links with other Ability applications.

Note: The following rules should be observed with regard to field names. They can be up to 64 characters long, and contain any mixture of letters, numbers and underscores. They cannot contain spaces, periods, commas, colons, brackets, slashes or other punctuation marks, nor can they contain hyphens, dashes or ASCII characters. The names of pre-existing functions, such as “Sum” or “Title”, are also inadmissible. The following are all legitimate field names:

bank_account

BankAccount

bankaccount

bank_acc

bank_acc_1

BankAcc1

For more details of how to create fields in a database, see [Creating fields in a database](#).

Data types

Ability understands many types of information:

- textual
- numeric
- date
- time

In addition to these basic data types, Ability can also work with:

- Large blocks of text: comments and memorandums
- Automatically incremented numbers: invoice numbering, membership registration.
- Automatically calculated formulas: *cost* equals *unit_price* multiplied by *quantity*.
- Predetermined lists of valid entries: {yes | no | pass}, {Green | Blue | Brown}
- Predetermined logical entries: {yes | no}, {true | false}, {1 | 0}
- OLE objects: pictures, graphics files, other file formats

Ability recognizes that all of these different types of information have unique requirements for data entry, storage and formatting. When you create a field in the design window, you must choose one of the ten standard data types, and appropriate formatting for the type of field (see step 3 of [Creating fields in a database](#)). As the most common type of field, the Character data type is the default.

The following sections explain the various field data types (**Character**, **Numeric**, **Date/Time**, **Memo**, **Incremental**, **Calculated**, **Logical**, **Enumerated**, **OLE**) and their associated formatting properties.

See:

[Character](#)

[Numeric](#)

[Date/Time](#)

[Memo](#)

[Incremental](#)

[Calculated](#)

[Logical](#)

[Enumerated](#)

[OLE](#)

Character data type

A field with Character data type (see [Data types](#)) can contain any characters – letters, numerals or symbols. You can enter up to 255 characters, which includes multiple sentences, but only one paragraph. If, during data entry or when editing, the amount of text you type into a character field exceeds the column width of the display column, Ability scrolls the field box to show the text as you type it. You then need to adjust the width of the field by dragging the right edge of the bar at the top of the column (hold down the left-hand mouse button to do this).

The formatting properties available on the **General** page in the Format dialog are:

- **Width** (this is the number of characters allowed in the field, *not the column width*, up to a maximum of 255 characters - the default is 50) – see [Character field size](#)
- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless another value is entered instead of it - see [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Required** (when checked a value is required in field - see [Required property](#))
- **Allow Zero Length** (allows zero-length string in field - see [Allow zero-length property](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

The formatting properties available on the **Format** page are:

- **Case** (Mixed, Upper & Lower - the default is Mixed)
- **Mask** (various options that control the pattern for certain types of data, such as telephone numbers and zip (post-) codes). See [Masks in database fields](#).

Character field size

If you examine any initial paper sketch of the design of your table, produced in the planning stage, you will notice how some columns have plenty of white space and others are densely packed. Real data rarely conforms to an idealized table with evenly sized columns. Those densely packed columns are probably text fields. It is a good design principle to decide, for each separate text field, the number of characters required to accommodate the largest individual item of information likely to be entered. Write this figure down for each of the text fields. The maximum size of a standard text field is 255 characters. For larger quantities of textual information you may need to use a memo. See [Memo](#).

The other non-character data items in your table design will not require a specific field size to be entered when you create the table. Ability has built-in defaults to make table creation as easy as possible for you.

Masks in database fields

A mask is a field formatting property that determines the pattern of data entered into a field and controls the type of data allowed. A typical example is an invoice number, which is usually a combination of letters and numbers in a set order. An appropriate mask helps data in-putters follow the convention defined by the mask. Masks are used in conjunction with the Character field type (see [Character](#)).

There are three default masks provided with Ability. These are:

- ###-##-#### (Social Security Number) e.g. 456-23-9847
- (###) ###-#### [#####] (Phone) e.g. (447) 756-1252 [4589]
- #####-#### (Zip Code) e.g. 67845-6564

To choose one of these when formatting the field (see step 3 of [Creating fields in a database](#)), click on the arrow at the side of the **Mask** box on the **Format** page of the “Format Field” dialog.

You can, however, create your own masks, using the following codes to determine whether to allow numbers, letters, other characters, or combinations of any of these:

- # (number)
- & (all standard keyboard characters)
- A (numbers and upper and lower case letters)
- ? (lower or upper case letter)
- U (upper case letter)
- L (lower case letter)

All of these can be left blank when filling in the mask. The mask can be formatted to contain, between the codes, any characters that are not in the above list, for example, . , ; : () [] { } / \ < > a b c 1 2 3 and all the other characters on your keyboard except for #, &, A, ?, U, L.

Example

You can create a mask that will take invoice numbers of this form:

(ABC)-[12345 a]

Type the following in the **Mask** box of the “Format Field” dialog when formatting the field (see step 3 of [Creating fields in a database](#)):

(UUU)-[##### L]

Note that the two types of bracket, the hyphen, and the space, constitute the basic structure of the mask. U will accept upper case letters, # numbers, and L lower case letters. If you try to enter data not allowed by the mask you will not be allowed to proceed.

Numeric data type

A field with Numeric data type (see [Data types](#)) can contain a variety of numbers, including ones with decimal places.

The formatting properties available on the **General** page are:

- **Width** (choose from **Byte**, **Integer**, **Long**, **Currency**, **Single**, **Double** - the default is **Long** - see [Width properties in numeric fields](#))
- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless another value is entered instead of it - [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Required** (when checked a value is required in field - see [Required property](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

The formatting properties available on the **Format** page are specifically connected with the various types of number - see [Number formats in database](#).

Number formats in database

Whenever you set a numeric data type for a field in a table, you will need to select which number format to use. Number formats are set on the Format page of the “Format field” dialog (see - [Numeric](#)).

There are five number types: **Number**, **Currency**, **Percent**, **Fraction**, and **Scientific**. Each of these has an associated set of options, which are displayed and can be set when you click on the relevant number type.

For information on the each of the number types and their associated options, see the following:

[Number format](#)

[Currency format](#)

[Percentage format](#)

[Fraction format](#)

[Scientific format](#)

Number format

1. Select **Number** from the list.
2. In the **Decimal Places** box, specify the precision of the number. You can have a minimum of 0 and a maximum of 15.
3. Select the **Thousands with Commas** checkbox for comma formatted numbers i.e. **1,345,622** rather than **1345622**.
4. Select the **Negatives in Brackets** checkbox to enclose negative numbers in brackets i.e. **(123456)** rather than **-123456**.
5. Select the **Negative in Red** checkbox to display negative numbers in red font.
6. In the **Trail/Lead String** option you can choose to have either trailing or leading characters. Enter your text in the text box or choose from the existing group (\$, £, DM, cm, kg, etc.). You can create your own number formats, including alternative currency formats. Here are some examples:
1.2 \$bn
FFr 23.56
DMarks
7. Check that the options you have chosen are correct by glancing at the **Preview** box and then click on **OK**.

See:

[Number formats in database](#)

[Width properties in numeric fields](#)

Currency format

1. Select **Currency** from the list. This will prefix the number with the currency symbol.
2. In the **Decimal Places** box, specify the precision of the number. You can have a minimum of 0 and a maximum of 15.
3. Select the **Thousands with Commas** checkbox for comma formatted numbers i.e. **\$1,345,622** rather than **\$1345622**.
4. Select the **Negatives in Brackets** checkbox to enclose negative numbers in brackets i.e. **(\$123456)** rather than **-\$123456**.
5. Select the **Negative in Red** checkbox to display negative numbers in red font.

Note: The currency symbol used by Ability is taken from the *Regional* settings in the *Windows Control Panel*.

See:

[Number formats in database](#)

[Width properties in numeric fields](#)

Percentage format

1. Select **Percent** from the list to display numbers as percentages e.g. 1 as 100%, 0.3 as 30%.
2. In the **Decimal Places** box, specify the precision of the number. You can have a minimum of 0 and a maximum of 15.
3. Select the **Thousands with Commas** checkbox for comma formatted numbers i.e. **1,345,622%** rather than **1345622%**.
4. Select the **Negatives in Brackets** checkbox to enclose negative numbers in brackets i.e. **(123456)** rather than **-123456**.
5. Select the **Negatives in Red** checkbox to display negative numbers in red font.

See:

[Number formats in database](#)

[Width properties in numeric fields](#)

Fraction format

1. Select **Fraction** from the list to display numbers as fractions e.g. **0.5** as $\frac{1}{2}$, **2.5** as $2\frac{1}{2}$.
2. In the **Display as** box, specify the precision of the fraction. You can choose from **Auto**, which will display the most accurate fraction possible in Ability, to **4ths**, that is the number will be rounded to the nearest $\frac{1}{4}$ fraction.
3. Select the **Thousands with Commas** checkbox for comma formatted numbers i.e. **1,345,622** rather than **1345622**.
4. Select the **Negatives in Brackets** checkbox to enclose negative numbers in brackets i.e. **(2 $\frac{1}{2}$)** rather than **-2 $\frac{1}{2}$** .
5. Select the **Negatives in Red** checkbox to display negative numbers in red font.

Note that with the Fraction format you should set the width to either **Single**, **Double** or **Currency** to take account of the fractional part of a number i.e. the digits to the right of the decimal point. Fractions cannot be expressed in a field with **Long** width, since this can only contain numbers without a decimal part.

See:

[Number formats in database](#)

[Width properties in numeric fields](#)

Scientific format

1. Select **Scientific** from the list to display numbers in scientific form e.g. **2.5E+003** rather than **2500**.
2. In the **Decimal Places** box specify the precision of the number. You can have a minimum of 0 and a maximum of 15.

See:

[Number formats in database](#)

[Width properties in numeric fields](#)

Width properties in numeric fields

The Width formatting option for numeric fields (see [Numeric data type](#)) determines how large a number the field will accept. The options are as follows:

Byte	The field will accept numbers from 0 to 255
Integer	The field will accept numbers from -32,768 to 32,767
Long	The field will accept numbers from -2,147,483,648 to 2,147,483,647
Currency	The field will accept standard currency formats, including decimal places and currency symbols
Single	The field will accept negative numbers from -3.402823E38 to -1.401298-E45 and positive numbers from 1.401298E-45 to 3.402823E38
Double	The field will accept negative numbers from -1.79769313486231E308 to -4.94065645841247E-324 and positive numbers from 1.79769313486231E308 to 4.94065645841247E-324

Note that **Double** is the default width for all new **Numeric** fields. If you wish to enter numbers that refer to another table's primary key where the primary key is an Incremental field, use the **Long** format.

Date/Time data type

A field with Date/Time data type (see [Data types](#)) can only contain valid dates.

The formatting properties available on the **General** page are:

- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless another value is entered instead of it - [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Required** (when checked a value is required in field - see [Required property](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

The formatting properties available on the **Format** page are:

- **Year** (choose from Long Form e.g. 1999, Short Form e.g. 99, & None)
- **Month** (choose from Long Name e.g. January, Short Name e.g. Jan, & Digits e.g. 01)
- **Order** (choose from Month/Day e.g. Jan/12, & Day/Month e.g. 12/Jan)
- **Separator** (choose from \ - . / or blank e.g. 12-05-99, 12.05.99, 12/05/99, 12 05 99, etc.)
- **Add Time** (click on the check box to add time to the date: select 0..12 to display AM/PM rather than the 24-hour clock e.g. 03:45 PM rather than 15:45, and Seconds to display seconds e.g. 15:45:34)

If you only want to display the time, select **Time** rather than **Date** in the left-hand box and proceed as shown above.

Date/Time format

Fields that contain dates or times can be formatted to control their appearance. Although dates and times are usually displayed in fields that have the Date/Time data type (see [Date/Time data type](#)), they can also be the result of calculations in fields with the Calculated data type (see [Calculated data type](#)). Dates and times are formatted in a similar way in any field that can contain them.

To format the **Date**:

1. Select the **Year** format by choosing from Long Form e.g. 1999, Short Form e.g. 99, & None.
2. Select the **Month** format by choosing from Long Name e.g. January, Short Name e.g. Jan, & Digits e.g. 01.
3. Select the **Order** by choosing from Month/Day e.g. Jan/12, & Day/Month e.g. 12/Jan.
4. Select the **Separator** type by choosing from \ - . / or blank e.g. 12-05-99, 12.05.99, 12/05/99, 12 05 99, etc.

To format the **Time**:

1. Select the **Add Time** checkbox to add time to the date.
2. Select **0..12** to display AM/PM rather than the 24-hour clock e.g. 03:45 PM rather than 15:45.
3. Select **Seconds** to display seconds as well as hours and minutes e.g. 15:45:34.

If you only want to display the time, select **Time** rather than **Date** in the left-hand box and proceed as shown above.

Memo data type

A memo field is similar to a character field, only much larger and can contain multiple paragraphs.

The formatting properties available on the **General** page are:

- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless another value is entered instead of it - [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Allow Zero Length** (allows zero-length string in field - see [Allow zero-length property](#))
- **Required** (when checked a value is required in field - see [Required property](#))

The formatting properties available on the **Format** page are:

- **Memo Type (Plain or Rich Text:** Rich text allows formatting of individual portions of text in bold, italics, etc., whereas Plain text only allows formatting for all the text in a field)
- **Append in New Record:** (click on the check boxes to have the date and/or time appear automatically in the Memo)

Note that when you go to the table view and enter data in a particular record, the date and/or time will not appear in the memo field for that record until you go to the next row.

Incremental data type

An incremental field assigns a unique sequential number for each record added to a table. The *starting* value is 1 and the *step* value is also 1. When you add the first record to the table, the incremental field is set to the start value. When you add additional records, the incremental field will automatically be filled in with the value from the last entered record plus the step value. Even if you subsequently delete records from the table, old values are not re-used. This makes an incremental field ideal for use as a primary key field.

The formatting properties available on the **General** page are:

- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Index** (choose from Primary Key, Indexed, Indexed (Unique), No Index). See [Indexes and primary keys](#)

The formatting properties available on the **Format** page are the same as for Numeric fields (see the Format page in [Numeric](#))

Logical data type

The logical type enables a field to have one of two logical values. Ability has four ways of expressing logical values: True/False, Yes/No, On/Off, 1/0.

The formatting properties available on the **General** page are:

- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless you overwrite it with another logical value - [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Required** (when checked a value is required in field - see [Required property](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

You can choose the type of logical value on the **Format** page:

- In the **Values** box choose from Yes/No, True/False, On/Off, 1/0

Once you go to Table View each new record will display the negative logical value by default i.e. False or No or Off or 0. To select a positive logical value, click on the arrow in the field and select it. Alternatively, at the formatting stage, you can choose the positive value as the default in the **Initial Value** box.

Boolean calculated field: example

A calculated field (see [Calculated](#)) can be formatted to return Boolean values i.e. TRUE or FALSE. An example of the usefulness of such a field is the following.

Imagine you have two sets of data, for which you want to make a direct comparison item by item e.g. yearly income and expenditure over a five-year period. The **Success** column is a calculated field and contains this formula:

=IF(Income>Expenditure, True, False)

Income	Expenditure	Success
30,000	39,000	FALSE
28,000	27,000	TRUE
33,000	33,000	FALSE
100,000	40,000	TRUE

Enumerated data type

A field with the enumerated data type allows you to choose from a list of values that have been chosen by you.

The available formatting properties on the **General** page are:

- **Initial Value** (enter a default value that will automatically appear in this field every time a new record is created, unless you overwrite it with another value from the enumerated list - [Initial value](#))
- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Validation Rule** (set a rule to validate possible data entries - see [Validation rule](#))
- **Validation Text** (write a display message for cases of invalid data - see [Validation text](#))
- **Required** (when checked a value is required in field - see [Required property](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

You can enter your values for the enumerated list on the **Format** page:

- In the **Values** box type the entries one per line (e.g. Brown, Blue, Red), using **Ctrl-Enter** to go to the next line in each case.

See [Adding enumerated field types](#) for more information on enumerated fields and an example.

Adding enumerated field types

You can add your own *enumerated* field types to a database table. An enumerated field is one that limits the values that can be put into the field to a pre-defined set, these being defined by you. This is useful where a field would naturally contain one of a limited set of values. For example, if you create a patient identification table, and it includes fields for height, weight and eye color, you can make the eye color field an enumerated field and permit only the values *blue*, *green*, *brown* and *hazel*.

To create an enumerated field for eye color, do the following:

1. Type a name for the field in **Name**, for example, EYECOLOR; set the field type at **Enumerated in Type**; and click on the **Format** button to display the "Format Field" dialog (See [Creating fields in a database](#) for information on creating table fields).
2. In the "Format Field" dialog click go to the **Format** page. In the **Values** box type the members of the enumerated set: BLUE, GREEN, BROWN, HAZEL. The values should be entered one per line, using **Ctrl-Enter** on the keyboard to proceed to the next line each time.
3. Click on **OK** to save the new information and close the "Format Field" dialog, and click **OK** again to close the "Format Table" dialog.
4. The table grid is displayed. Select the EYECOLOR field to display a drop-down menu arrow. When you click on the arrow all the valid values for that field are displayed. These are the only values allowable in this particular column. Click on one of these to insert it into the table cell.
5. When you are adding records to the table and you *type* a value into the enumerated field and press **Tab** to move on, one of three things will happen:
 - (a) If you have put in one of the acceptable values, Ability will accept it.
 - (b) If the value you enter is not in the enumerated set, Ability will not accept the value and the cell will remain blank.
 - (c) If you type the first letter (or several letters) of a valid value, Ability will display the first alphabetical entry from the list that matches this letter (or letters).

OLE data type

The OLE data type allows you to link to or embed various OLE-compatible objects, such as pictures, graphics, sound, spreadsheets and other files.

The only formatting you need to decide on at this stage is what caption (see [Captions](#)) you will have, if any, and whether or not to make it obligatory that each record contain an OLE object in this field (see [Required property](#)).

Once you have set up the OLE field and gone to the Table Grid View, you need to insert the OLE object.

1. Double-click on an OLE data type cell. This displays a box into which an object can be inserted.
2. Select **File** from the **Insert** menu to insert an existing object, or select **New Object** if you wish to create a new object.

For example, to insert a new Ability Spreadsheet into a Database cell, select **New Object** from the **File** menu and then select Ability Spreadsheet from the **Object Type** box in the "Insert Object" dialog.

Once you have inserted a new or an existing OLE-compatible object, you can resize the object window and edit the object's contents in the usual way. To edit the object, a spreadsheet say, double-click inside the object to put it in edit mode. All the familiar editing and formatting menus become available across the top of the window. To resize the object window, click once outside the window and then use the borders and handles of the window to move and resize the object.

Calculated data type

The calculated data type enables you to resolve simple arithmetic formulae based on single or multiple field values within the current record. The current record may belong to a **Relation**, thereby enabling you to use fields from different tables in a calculation.

The formatting properties available on the **General** page are:

- **Caption** (enter a caption to replace the field name above the column in Table View - see [Captions](#))
- **Index** (choose from Primary Key, Indexed, No Index & Indexed (Unique) - see [Indexes and primary keys](#))

The formatting properties available on the **Format** page are for various types of number, date and time, and the Boolean operator. These determine the format of the answer yielded by the calculated field:

- **General** (the general number format)
- **Number** (see [Number format](#))
- **Currency** (see [Currency format](#))
- **Percent** (see [Percentage format](#))
- **Fraction** (see [Fraction format](#))
- **Scientific** (see [Scientific format](#))
- **Date and Time** (see [Date/Time format](#))
- **Boolean** (use for calculations that return either TRUE or FALSE - see [Boolean calculated field](#))

On the **Formula** page you can either type in a formula yourself or click on the **Function** button to display the “Insert Function” dialog, in which you can choose from the standard Ability functions (see [Using built-in functions](#) and [Using the function dialog box to build formulas](#) for general information on creating functions). Once the formula has been inserted into the **Formula** box, type in the *field names* you want to act as arguments for the formula)

See:

[Using the calculated data type in a table](#)

[Using the calculated data type in a relation](#)

Using the calculated data type in a table

You can use the calculated data type to perform calculations with values from fields in the *same* table. You can also perform calculations using fields from different tables by joining these fields in a *relation* (see [Using the calculated data type in a relation](#)).

The following example should make it clear to you how to perform calculations in a single table.

Imagine you have a table that contains fields for the product ID, the quantity in stock of each product, and the cost of each product:

Product_ID	Quantity	Cost	StockValue
A001	5	27.99	139.95
A002	7	45.50	318.50
A003	4	20.00	80.00
A004	6	8.99	53.94

You want to add the extra column to calculate the total value of each product in stock. Follow these steps:

1. If you are already in your table, select **Table** from the **Format** menu to display the “Format table” dialog.
2. In “Format table” go to the next empty line. Type “StockValue” in the **Name** column and select **Calculated** in the **Type** column.
3. Click on the **Format** button to display the “Format Field” dialog.
4. On the **Format** page select either **Number** or **Currency** and make sure **Decimal Places** is set at 2.
5. On the **Formula** page type the following in the **Formula** line:

=Quantity * Cost

6. Click on **OK** to return to “Format table”.
7. Click on **OK** to return to your table.

The table will be redrawn to show the result of the calculation, for each record, in the StockValue column.

Using the calculated data type in a relation

You can use the calculated data type to perform calculations using fields from different tables by joining these fields in a *relation*. You can also perform calculations with values from fields in the *same* table (see [Using the calculated data type in a table](#)). The following example should make it clear to you how to perform calculations in a relation.

Example

Imagine you have two tables: one called Orders, which records the product and quantity for each order, and one called Products, which records the price of each product and gives a description of the product. Each order and product is given an ID and both tables have the Product ID field in common. It is your aim to calculate the total value for each of the orders.

Orders

Order_ID	Product_ID	Quantity	Value
ORD1	A003	5	
ORD2	A001	7	
ORD3	A003	4	
ORD4	A002	3	
ORD5	A002	2	
ORD6	A004	6	

Products

Product_ID	Description	Cost
A001	File	2.99
A002	Lamp	25.50
A003	Pen	1.20
A004	Fan	18.99

The total value for each order can be calculated from multiplying the values from Quantity (in Orders) by the values from Cost (in Products) for each Product ID. Since Quantity and Cost are in different tables, you need to *join* the tables to create a *relation*. Before doing this you also need to add to one of the tables a field that will contain the total value. In this case I have added the field to Orders and called it Value. This field will be given the **Calculated** data type and formatted to return the result of multiplying the values in the Quantity and Cost fields.

1. Create a table, Orders, with the four fields: Order_ID, Product_ID, Quantity, and Value. Order_ID and Product_ID should be given the **Character** data type; Quantity the **Numeric** data type; and Value the **Calculated** data type (make sure Value is either **Number** or **Currency**). See [Creating fields in a database](#).
2. Use the “Format Field” dialog to format the fields as follows (most of the formats are OK as they stand):
 - (a) Make Order_ID a **Primary Key** (this will prevent duplicates)
 - (b) Make Product_ID a non-unique index i.e. **Indexed** (this will make sort and search operations more efficient when you have many records)
 - (c) Reduce **Decimal Places** in Quantity to 0 (all quantities will be whole numbers)
 - (d) Increase **Decimal Places** in Value to 2 (to display cents or pennies properly)
 - (e) Type the formula **=Quantity * Cost** into the **Formula** line on the **Formula** page for Value (see [Calculated](#) and [Using the calculated data type in a table](#) for an example of how to enter a

function)

For more information on formatting, see [Formatting database fields](#)

3. Next, create a table, Products, with the three fields: Product_ID, Description, and Cost. Product_ID and Description should be given the **Character** data type, and Cost the **Numeric** data type.
4. Use the “Format Field” dialog to format the Product_ID field as a **Primary Key** (to prevent duplicates).
5. Next, create a new relation based on these two tables (see [Creating a relation](#)). The **Join Field** in both tables should be Product_ID and the **Join Type** should be **Inner** (so that only the products which are listed in both tables will be displayed)

The **Relation** table will look like this:

Orders. Order_ID	Orders. Product_ID	Orders. Quantity	Orders. Value	Products. Product_ID	Products. Description	Products. Cost
ORD2	A001	7	20.93	A001	File	2.99
ORD4	A002	3	76.50	A002	Lamp	25.50
ORD5	A002	2	51.00	A002	Lamp	25.50
ORD1	A003	5	6.00	A003	Pen	1.20
ORD3	A003	4	4.80	A003	Pen	1.20
ORD6	A004	6	113.94	A004	Fan	18.99

You would undoubtedly want to sort the first column into ascending order and hide either the second or fifth column. Note that you can rearrange and hide columns, sort and filter records, just as in any other table. See [Query components](#) for more information.

Captions

In addition to the name of the field, you can also give the field a caption. The caption is the name that will appear above the field when the table is open. It is entered in the **Caption** box when you format the table. If it is left empty the name above the field will be the same as that in **Name**. Captions are different from names in that they are usually more descriptive and do not have any of the restrictions on the types of character that can be used in a name.

Note: Although a caption will appear above a column in a table when the table is open, you must refer to the field name, as defined in **Name**, when you are trying to make a link to the database from another Ability (or non-Ability) application. The field name is the official name, so to speak, and is obligatory, unlike the caption.

Validation of data

Ability will automatically check that any data you try to enter into a database field is of the right type (as set when you designed the table). As you would expect, a field set to take numeric values will not be allowed to accept text.

However, there is another level of validation available to you when you are formatting fields. This allows you to exclude data that is of the right data type for the field but is perhaps outside the range of data you want to appear in this field. You can specify a **validation rule** (see [Validation rule](#)) for what is acceptable or unacceptable within the data type you set. Furthermore, you can write your own error message or **validation text** (see [Validation text](#)), which will be displayed whenever you attempt to enter data that breaks the validation rule.

Validation rule

A typical validation rule is a formula that determines what is to be included or excluded in a field. You can use standard Ability expressions to design the validation rule. The rule will then be compared with any data you attempt to enter into the field. If the data is allowed by the validation rule it will be accepted by the field; if not then the data will be refused and a validation text displayed (see [Validation text](#)).

Typical examples of validation rules are:

- **>= "20" AND <= "40"** for a numeric field (allows any numbers from 20 to 40 inclusive)
- **= "Smith" OR "Jones"** for a character or enumerated field (allows only "Smith" or "Jones" but no other names)

Note that the data elements in each of the formulas should be surrounded individually by quotes.

Since a validation rule determines what can be put into a field and will cause an appropriate message to be displayed if the rule is broken, it turns out that a blank field governed by a validation rule will be illegitimate and the validation text displayed as if the rule had been broken. Obviously there are cases where you will want fields to be left empty. To overcome this problem you should do the following:

- Add the expression **Is Null** to the end of the **Validation Rule**.
- Make sure that the **Required** box is unchecked (this indicates that data is not required)

For example, a validation rule might read:

= "Smith" OR "Jones" OR Is Null

which will allow "Smith", "Jones" or a blank field.

See:

[Validation of data](#)

[Required property](#)

Validation text

Validation text always accompanies a validation rule (see [Validation rule](#)) for a database field and is displayed when that rule is broken. You will not be allowed to go to the next line of a table until you have entered data that is allowed by the validation rule. Only then will the validation text not be displayed. If you decide not to create your own validation text, Ability will generate a default message to tell the user what rule has been broken.

Note that in any line there may be several fields that have an attached validation rule and its accompanying validation text. In this case only one validation text at a time will be displayed. Once you have dealt with this by entering legitimate data into the appropriate field, it may still be the case that one or more of the other fields contains illegitimate data. Validation text will continue to be displayed, in turn, for each of the fields still containing illegitimate data, until you have dealt with all these fields.

Typical examples of validation text are:

- **Must be either Smith or Jones** when the validation rule =**“Smith” OR “Jones”** is broken.
- **Must be either Smith or Jones or Blank** when the validation rule =**“Smith” OR “Jones” OR Is Null** is broken
- **Number must be between 20 and 40 inclusive** when the validation rule **>=“20” AND <= “40”** is broken.

See:

[Validation of data](#)

Required property

The Required property should be switched on if you do not want blank fields to be allowable, that is if you *require* that a field contain a value. Any attempt to leave a field blank when it has been formatted with the Required property switched on, will return an error message.

For details of how the Required property relates to Validation Rules and the Zero-length property, see [Validation rule](#) and [Allow Zero-Length Property](#).

Allow zero-length property

The Allow Zero Length property, when switched on, permits a field to contain a zero-length string (“”) as value. This should be distinguished from the **Null** value, which allows fields to be blank and is permitted when the **Required** option is switched off.

The basic difference between a field that has a null value and one that contains a zero-length string, is that the former field contains - and has contained - absolutely nothing, that is no spaces have been entered, no deletions taken place, etc., whereas the latter may be empty to the eye but in fact has at one time contained data, even though it has since been deleted.

Hence, as you would expect, the two types of formatting cause fields to behave in different ways. Likewise, the various format combinations, using both the **Required** property and the **Allow Zero Length** property, also have different effects, as shown below:

- **Allow Zero Length** and **Required** both set to **No**: **Null** value stored when **Enter** or **Spacebar** is pressed, but zero-length string is not allowed (i.e. you will not be allowed to delete a existing value, although you will be able to replace it)
- **Allow Zero Length** and **Required** both set to **Yes**: **Null** value not allowed (i.e. when you press **Enter**), but zero-length string stored when you press **Spacebar** or enter a zero-length string (i.e. as when you delete an existing value)
- **Allow Zero Length** set to **Yes** and **Required** set to **No**: **Null** value stored when **Enter** or **Spacebar** pressed, and a zero-length string stored when it is entered (i.e. as when you delete an existing value)
- **Allow Zero Length** set to **No** and **Required** set to **Yes**: **Null** value not allowed (i.e. when you press **Enter** or **Spacebar**), and a zero-length string not accepted (i.e. as when you try to delete existing value)

See also:

[Required property](#)

Initial value

You can set an initial or default value for a field, depending on its data type. This is set when you are formatting the field (see [Data types](#)). Later on you can reformat the field to change the initial value. To do this just type the new default in the **Initial Value** box in the “Format Table” dialog for the relevant field.

Every time a new record is added to a table, all the fields that are formatted to have initial values will automatically be updated. You can of course replace an initial value in Table Grid View by entering an alternative value.

Note: Initial values must be compatible with the data type of the field e.g. characters for a **Character** field, numbers of the right format for a **Numeric** field, dates of the right format for a **Date** field, and so on. When you change the default value of a field, only new records receive this value; existing records retain their old values, which may be there through the old default or because they were directly entered by you.

Creating fields in a database

You can create fields for a new table or add to and modify them in an existing one.

- Create a new table by right-clicking your mouse anywhere in the **Database Manager** window and selecting **New**. Select either the **Table Wizard** or the **Blank Table** button. The following notes assume you have selected **Blank Table** (see [Creating fields in a database using Table Wizard](#) for more information on using Table Wizard). The "New Table" design dialog is displayed.
- If you want to add fields to an existing table, select the table in **Database Manager** and right-click your mouse. Click on **Format** from the displayed menu to go to the "Format Table" dialog.

"Format table" and "New Table" work in exactly the same way, so follow these steps for each of the fields that is to be created:

1. Click in the **Name** box to enter a name for the field. This must be unique for that particular table. See [Field names](#).

Note: It is good practice to use standard field names in your table designs. Since Ability uses field name references when calling data from other applications, standard field names will give you greater freedom to switch Forms, mailing labels and mail merge documents from one table to another. See [Creating fields in a database using Table Wizard](#) for more information on standard field names.

2. Press **Tab** to move into the **Type** box. Ability automatically displays the **Character** field type. If you need to change the field type, click the arrow next to the **Type** box to display a list of all field types. Select the required type with a single click. For more information, see [Field data types](#).
3. Once you have set the field type, it becomes possible to *format* the field. The format properties differ according to the type selected. It is a good idea to set the format properties immediately after setting the type. To format a field, click once on the **Format** button at the bottom right of the window. The "Format Field" dialog is displayed, in which the various properties can be set, depending on the type. See [Field data types](#) for an explanation of the various data types and their associated formatting properties.
4. Press **Tab** to move to the **Description** box. Here you can type an optional description that will help remind you or others what the field is supposed to contain.
5. Once you have reached the end of the line, that is, defined one field, press **Return** or **Tab** to go to the next line and define another field in the same way.
6. When you have defined all the fields that you will need, you can give the table a general description in the **Table Description** box.
7. Next click on the **Index** button to check (or set up) your indexes. See [Indexes and primary keys](#).

When all is set, click on **OK** to create a table based on your design.

Creating fields in a database using Table Wizard

You can design a database table using Table Wizard. This is a quick and easy way of creating a table based on standard table and field designs. These are meant to reflect the requirements of the average database user and should be adequate for most purposes. To use the Table Wizard follow these steps:

1. Right-click your mouse when in the **Database Manager** window and select **New** followed by **Table**.
2. Click on the **Table Wizard** button in the "New Table" dialog. The "New Table Wizard" dialog is displayed.
3. In the **Available Tables** box the set of standard tables is listed, for example, *Customer*, *Product* and *Employee*. Click on one of these, say *Customer*, to select it and then click on **Next**.
4. The **Available Fields** box displays the standard fields that come with the table you have selected. For the *Customer* table these include name and address fields etc. Select the field that you want to be the first in your table and then click on the **Add** button (the topmost). The selected field is moved from the **Available Fields** box and placed in the **Selected Fields** box. Repeat this operation for the rest of the fields that you require.
5. If you decide that you don't want to use a field after you have moved it to the **Selected Fields** box, then select the field and click on the **Remove** button (the second) to return it to the **Available Fields** box. Likewise, if you want to change the order of fields, you can move selected fields back and forth - including groups of selected fields - by using the **Add** and **Remove** buttons. To move all the fields in the **Available Fields** box over to the **Selected Fields** box, use the **Add All** button (the third). The **Remove All** button (the fourth) moves everything in **Selected Fields** back to **Available Fields**.
6. Once you have chosen all the fields you want, click on **Finish**. This takes you to the "New Table" dialog, where you can format the fields, create captions and descriptions, define initial values, and even add extra fields that are not available through Table Wizard. See [Creating fields in a database](#).

Indexes and primary keys

Indexes are used by Ability to speed up sort and search operations on table fields. They are also used to determine the type of link that can be made from one table to another in a *relation* (See [Joining tables](#)).

Each record in the field is given its own index position, so to speak, by which it can be ordered. You should set an index for a field if you think you may need to search or sort on that field frequently. Indexes can also be used to prevent certain fields holding duplicate data.

The only disadvantage in using indexes is that they may sometimes slow down various table operations, such as query operations, especially when there are many records in the table.

You set an index for a field when you set the data type. The fields that can be indexed are those that are capable of being ordered, either numerically or alphabetically, such as numeric, character, date and time fields (See [Creating indexes and primary keys](#)).

There are three types of index: primary keys, unique indexes, and non-unique indexes. In addition, you can create multiple-field indexes.

See:

[Primary key](#)

[Unique indexes](#)

[Non-unique indexes](#)

[Multiple-field indexes](#)

Primary key

A primary key is an index that *uniquely* identifies each record in a table. It can be set for one and only one field, which is known as the primary key field. This field will not accept duplicate or blank values. If you have entered a duplicate value or left the field blank, you will not be able to move to the next or previous record until you change or delete the duplicate value or replace the blank with a unique value. Primary keys are useful when you use Ability Database to set up relations between tables.

In addition, the Primary Key field in a table automatically sorts the table into ascending order according to the values in the field. If you add extra records, they will be automatically ordered according to the Primary Key field when you close and re-enter the table.

Note: If you haven't set a primary key while designing a table, you can set it *after* records have been entered. However, if the field already contains duplicate values or if it contains blanks, then Ability will display an error message asking you to remove duplicates or to replace blanks with a unique value before it accepts the field as the primary key field. The primary key will not be set for a field with duplicate values or blanks and will format the field so that it has no index. If you can't edit the field to rid it of duplicate values, you can create an extra field and give it the *Incremental* data type. This will number all the records and ensure their uniqueness. Even when you delete records and add new ones, the new records will not be given numbers that have already been used. This is obviously useful as the basis of an invoice table, where invoices are required to be unique.

See:

[Creating indexes and primary keys.](#)

Unique indexes

A unique index is similar to a primary key in that it will not allow duplicate values in the field for which it has been set. However, it is unlike a primary key in that you can set as many unique indexes as you wish for a table, and you are allowed to leave a unique index field blank for any record. This is obviously useful in that you may not want duplicate data to be allowable in a particular field, though you do want to allow certain records to be blank in this field. A unique index field, unlike a primary key field, will not count two or more blanks as duplicates.

See:

[Creating indexes and primary keys.](#)

Non-unique indexes

A non-unique index enables a field to have duplicate values and blanks, thereby enabling you to search and sort faster on fields that will contain repeat information as well as some blanks. You can set as many non-unique indexes as you wish for a table.

See:

[Creating indexes and primary keys.](#)

Creating indexes and primary keys

To create an index or a primary key for a field in a database table, you first need to choose a type for the field in the **Type** box (see [Field data types](#)). At this stage, you may be designing a new table or editing/formatting an existing table design (see [Creating fields in a database](#)).

Once you have set the type for the field, click on the **Format** button in the dialog (either "New Table" or "Format Table").

In the "Format Field" dialog, click in the **Value** box to the right of **Index**. Click on the arrow to display all of the Index formatting options. The full set comprises:

- **No Index** (no index applied to this field)
- **Primary Key** (primary key applied to this field) See [Primary key](#)
- **Indexed** (a non-unique index applied to this field) See [Non-unique indexes](#)
- **Indexed (Unique)** (a unique index applied to this field) See [Unique indexes](#)

Click on one of these to select it. Note that only one field can be formatted as a **Primary Key** field. If you afterwards choose another field as the Primary Key, the first field will automatically become **Indexed (Unique)** instead.

Not all of the available data types can be index formatted. Obviously it makes no sense to index Memo or OLE fields, therefore no indexing options are provided for these. Moreover, though you could set a Primary Key for a Logical field, this would only make sense if you intended your table to contain a maximum of two records. The Incremental data type, if it is set for a field, is automatically the best choice to made the Primary Key.

Here is a list of index options and the data types that can be used with them:

Primary Key	Indexed	Indexed (Unique)
Character	Character	Character
Numeric	Numeric	Numeric
Date	Date	Date
Time	Time	Time
Enumerated	Enumerated	Enumerated
Incremental		Incremental
Logical	Logical	Logical
Calculated	Calculated	Calculated

You can check which fields have indexes set for them at any time. In addition, you can create multiple-field indexes.

See:

[Checking indexes and primary keys](#)

[Multiple-field indexes](#)

Checking indexes and primary keys

Once you have set the indexes and primary key for a table, you can check them by clicking on the **Index** button in the "New Table" or "Format Table" dialog. In the **Table Indexes** box all the indexes set for a particular table are listed:

- Under **Index Name** the name of the index is displayed. By default this is the same as the name of the field, in **Field Name**, for which the index was set (although you can rename the index here, if you wish).
- **Sort Order** is Ascending by default - click in the box to change this to Descending. The sort order determines how certain fields are displayed when you go into Table Grid View. If you have set a Primary Key this will take precedence over any of the other indexes and the records will be sorted into ascending or descending order according to the values in the primary key field. If you have not set a primary key, then the records will be sorted according to the first index listed in the **Table Indexes** box. Every time you go into a table, the sort index will take account of any new records previously added and will make sure that the table is correctly sorted.

To check what kind of index has been set for a field, click anywhere in that field's row in the **Table Indexes** box. The two check boxes at the bottom of the dialog - **Primary Key** and **Unique** - will be activated. Together these enable you to determine the kind of index set for each field:

- If **Primary Key** and **Unique** are both checked, then the index is the **Primary Key** (remember that only one primary key is allowed per table and that it is called *unique* because it does not permit duplicates)
- If **Primary Key** is not checked and **Unique** is checked, then the index is **Indexed (Unique)** (you can have many of these in a table, but duplicate values, except for blanks, are not permitted)
- If **Primary Key** and **Unique** are both not checked, then the index is **Indexed** (you can have many of these in a table and duplicate values are permitted)

For information on how to set indexes in the **Table Indexes** box, see [Editing and setting indexes and primary keys](#).

Editing and setting indexes and primary keys

Although you will usually set the indexes for your table by using the **Format** button when you set the data types for the fields, you can also set or edit them in the **Table Indexes** box of the “Format Table” (or “New Table”) dialog. You first need to click on the **Index** button to access the **Table Indexes** box.

To set a new index:

1. Click in an empty row in the **Field Name** column.
2. Click on the arrow to display all the available fields and then click on a field.
3. Give the index a name in the **Index Name** column (it's best to use the same name as for the field).
4. Once you have given the index a name, click anywhere else in the row to activate the **Primary Key** and **Unique** checkboxes. You can now set the index type by checking the appropriate boxes:
 - Check **Primary Key** to set a Primary Key index (note that the **Unique** box is automatically checked)
 - Check **Unique** to set a unique index i.e. **Indexed (Unique)**
 - Leave **Primary Key** and **Unique** unchecked to set a non-unique index i.e. **Indexed**

You can edit indexes in the **Table Indexes** box. Click in the row that contains the index to be edited and then check or uncheck **Primary Key** and **Unique** checkboxes, as appropriate. For more details on editing indexes, see [Checking indexes and primary keys](#).

Multiple-field indexes

It is possible in Ability Database to create multiple-field indexes. These are combinations of two or more table fields that *together* are unique. This is useful when you want each field in the multiple-field index to be able to hold duplicate data, though you don't want all the fields together to hold duplicate combinations of data.

For example, think of a field that contains orders and one that contains products. You want to be able to duplicate the order number, for there may be several products on any one order. Likewise one product may be attached to more than one order. What you don't want is that a particular product and order number together occur more than once, that is, that more than one record contains a duplicate combination on these fields.

To create a multiple-field index you need to be in the "Format Table" dialog (for an existing table) or the "New table" dialog (when creating a new table). See [Creating fields in a database](#).

To create a multiple-field index for an existing table, follow these steps:

1. Open the table and select the **Table** command from the **Format** menu (alternatively, right-click on the table name in **Database Manager** and select **Format**). The "Format Table" dialog is opened.
2. Click on the **Index** button to go to the index page.
3. Click in the first empty row in the **Table Indexes** box.
4. A multiple-field index for two fields is defined by two rows of the **Table Indexes** box:
 - To set the first row, click in the **Field Name** column for that row. Choose the appropriate field from the drop-down list. Give the index a name in the **Index Name** column. This name will be the name for the multiple-field index as a whole.
 - To set the next row, click in the **Field Name** column for that row. Choose the second field from the drop-down list. *Do not enter a name in Index Name for this row.* Ability recognizes this combination - one row with an Index Name, the next without - as a multiple-field index.
5. Next, click again anywhere in the first row. This will activate the **Primary Key** and **Unique** checkboxes. You can set the multiple-field index as either a primary key in its own right (check **Primary Key**) or as a unique index (check **Unique**). Whichever you choose, you can have duplicate values in each of the fields but not a duplicate combination of values. With **Primary Key** you are not allowed to have blanks in any of the fields of the Multiple-field Index. With **Unique** you are allowed to have duplicate blanks for either field or both in any row.

Although it is unlikely that you will need to create a multiple-field index with more than two fields, it is possible to add more fields as required. Again, with a three-field or higher multiple-field index, it is only the first field that is given a name in **Index** name.

Saving a new table

Once you have designed a new table and entered all the fields you require, click on the **OK** button at the bottom of the "New table" dialog. If this is the first time you have saved the table, the "Save As" dialog will be displayed. Enter a name for the table in the **Table** box.

The new table name will appear in the list of tables, for the current database, in **Database Manager**.

Opening a table using database manager

To open a table, double click on the table name in **Database Manager**. Alternatively, select the table and right click your mouse. Click **Open** in the displayed menu. The table is opened in **Grid View**. There are three types of table view: **Grid**, **Form** and **Report**.

See:

[What is a table view?](#)

Grid view

When you open a table in Grid View (also known as *browse mode*), you can add and delete records and generally edit the table's contents.

In Grid View your data is displayed in a standard table, where every row is a record and every column is a field.

Note: You will notice that the menu bar changes when you enter Grid View. This allows you easy access to the various commands you will need.

See also:

[What is a table view?](#)

Table record indicators

When you are in a table, in Grid View (see [Table views](#)), you will notice several indicators in the left-hand row bar. Here's what they mean:

- the new record is always indicated by a *star*
- the current record is indicated by a *pointer*
- the record that is being edited or into which new data is being entered, but which has not yet been saved (i.e. you haven't yet gone to the next line), is indicated by a *pencil*.

Modifying records

Once you've created your table, you'll probably spend most of your time working with the data in the records. There are various ways of working with records when in Grid or Form View, including adding, editing, deleting, copying and finding records.


See:

- [Adding records](#)
- [Editing records](#)
- [Deleting records](#)
- [Copying records](#)
- [Finding records](#)

Adding records

Whenever you open a table, the last row of the table is always a new record (i.e. blank) and is referred to as the new row. Adding a new record is therefore just a matter of typing data into the new row. The same applies to the forms based on the table.

To add a record, do any one of the following:

- Use the scroll bars to move to the bottom of the table and click on the new row.
- Select **New Record** from the **Record** menu, or click on **Add Record**  in the **Record** toolbar. The active cell will become the first position of the new row.

You can now enter data in the fields of the new record. Whenever you begin to enter data in the **New Record**, another blank **New Record** is automatically created.

Rather than entering fresh data into the new record, you may want to duplicate an existing record.

See:

[Duplicating records](#) for details on how to duplicate existing records.

[Table record indicators](#) for details on what the left-hand table indicators mean.

Duplicating records

You can create a new record, in a table, based on the contents of the current record. This may be useful if, for example, you are adding several contacts from the same company to your database.

Rather than re-type address details, select the **Duplicate Record** command from the **Record** menu (in **Grid View** you must select the whole record first by clicking in the left row bar, but in **Form View** it is sufficient to have the form on screen).

This will create a new record with the details of the current record copied to it. You can then edit the new record as you like.

Entering data and editing records

To fill in a blank field, first click on it or go to it using the **Enter** key. This will make it the active cell. Next, just type in or select the appropriate data for that field, according to the data type you set for the field.

To change the contents of a field, first go to it either by clicking in the field or using the **Enter** key. There are three possible operations you might want to perform on the cell:

- delete the contents
- replace the contents
- edit or add to the contents.

With most types of field you can click at the position in the field where you want to edit, or append new text to, the current contents. To replace or delete the current contents, double-click in the field to select the field contents and then type replacement text or press **Del** (if you entered the field using the **Enter** key, you can delete or replace text without clicking).

The method of editing fields depends to some extent on the data type of the field:

- **Character, Memo and Numeric** fields are straightforward and their contents can be edited, deleted or replaced, as outlined in the previous paragraph.

With other fields the procedure is slightly different but just as obvious:


- **Enumerated:** delete the contents by double-clicking in the field to select the contents and then press **Del**. Enter one of the members of the enumerated list by clicking on the arrow and selecting the appropriate one from the drop-down list, or alternatively type it yourself directly (what you type will only be entered if it is in the enumerated list).
- **Logical:** you can't delete the contents of a logical field but only change the field's value. If you try to delete the contents, they will be restored to the default value (No, False, Off or 0) as soon as you move to another record.
- **Date:** delete the contents by double-clicking in the field to select the contents and then press **Del**. Type the date according to the format you set when creating the field. The field will only accept dates that match the chosen format.
- **Time:** delete the contents by pressing **Del**. Type the time according to the format you set when creating the field. The field will only accept times that match the chosen format.
- **Incremental:** the contents of this field can't be edited or deleted. Incremental values are set when you format the field and are one way of uniquely identifying a record. If you delete an incremental value you won't be able to move out of the current record. To restore the value, press **Esc**. An incremental value can only be deleted by deleting the row in which it occurs.
- **Calculated:** the contents of this field are determined by the formula you set and therefore can't be edited or deleted. The only way to change the contents is to change the contents of the fields that are referred to in the formula. Deleting the formula in Table Format will of course remove all the results that are calculated by the formula.
- **OLE:** delete an OLE object by pressing **Del**. Insert an OLE object by double clicking in the field to display the OLE Table. Select either **File** (to insert an existing file) or **New Object** (to insert a new object) from the **Insert** menu (see [OLE data type](#)).

Usually, when you are editing fields, you can abandon the current edit by pressing the **Esc** key. The field will revert to its previous state.

After editing a field's contents, click on the field you want to edit next (or use the cursor movement keys).

Finding records


If you know the record number of a particular record in a table, it is quite easy to locate the record by using the **Go To** command in the **Edit** menu. However, sometimes you may not know the record number and therefore have to search for the record by finding a particular piece of data, which you know is held in a field of the record. If you are working on a small table it should be fairly easy to find the cell you require (for formatting, deleting, editing, or whatever) by eye. However, if you have a very large table, it can be quicker to ask Ability to find the cell you want. Whether in **Table Grid View** or **Form View** follow these steps:

1. Click on the **Find**  button on the toolbar or select the **Find** command from the **Edit** menu. The “Find” dialog is displayed.
2. In the **Find what** box, type the text you wish to search for.
3. In the **Direction** box choose whether you want to search **Up**, **Down** or in **All** directions from the current position. Ability searches down columns: for example, down one column from the present position, then down the next column, and so on.
4. In the **Scope** box choose whether you want to search in the **Whole document** or only in a **Selection**.
5. In the **Options** box you can choose to **Match case** and/or to **Match whole word only**.
6. Click on the **Find Next** button to begin the search. Ability will find the next occurrence of the search text and the cursor will move to that cell. Click on **Find Next** again to go to the next occurrence, and so on till there are no more to be found.


Deleting records

This section explains how to delete records in a table. In **Grid View** you can delete several records at once, but only one at a time in **Form View**.



You can delete one record, a group of records, or all the records in table **Grid View**. All of these are done through the same "Delete" dialog.

1. If you want to delete just one record it is best to make this record the current one. Do this by clicking anywhere in the record. The current row pointer is displayed in the left row bar. If you want to delete several records at once, you can select (or mark) these records by clicking your mouse on the left row bar and dragging till the required records are all selected.
2. Next, click on the **Delete**  button on the **Record** toolbar (or select **Delete Record** from the **Record** menu). The "Delete" dialog is displayed. Depending on whether you selected a current record, or marked one or more records, one of the following is checked in the dialog for deletion:
 - **Current Record** (deletes the record that has the pointer at its left)
 - **All Marked Records** (deletes all the marked records - note that the current record is not necessarily one of the marked records)
 - **All Records** (if you want to delete all records, then select this yourself - you can't delete the new record, of course)
3. Click on **OK** when you have chosen the records to be deleted. You will be warned that you cannot recover deleted records and asked to click **Yes** to continue with the deletion. The table will be redrawn with the deleted records removed.

The **Delete** command deletes the record *and* the row it occupies. However, if you use the **Cut** command instead, although the record and its row are both deleted, you can use the **Paste** command to paste the cut record in the new row at the end of the table.



To delete the current record in **Form View**, click the **Delete** button  on the **Record** toolbar or select the **Delete Record** command from the **Record** menu. You will be asked if you are sure that you want to proceed with the deletion. The deleted record will no longer appear as one of the forms and is also deleted from the attached table.

Deleting the contents of fields in a database table

- To delete the contents of the current field in a table, first click in the field to select it and then right-click your mouse and select **Cut** (alternatively press the **Del** key, click on the **Cut**  button on the toolbar or select **Clear** from the **Edit** menu).
- To delete the contents of several fields, select the fields and again select **Cut**, **Clear** or **Del**. You can select several fields in several rows by dragging the mouse.
- If you used **Cut** to delete the contents of the field(s), you can of course use **Paste**  to move the contents to another field(s).

Copying records

To copy records do the following:

1. Select the record by clicking on the left-hand row bar.
2. Click on the **Copy**  button in the toolbar (or right-click your mouse and select **Copy** from the menu).
3. Select the new row at the end of the table and click on the **Paste**  button in the toolbar (or by right-clicking your mouse). The record will be copied to the new row.

To copy several records, follow the above instructions but make sure you first select the relevant records by dragging your mouse before copying and pasting.

In the same way, you can copy records from one table and paste them into another. Take care though, you need to have the same – or almost the same – table structure to be able to do this sensibly.

Records can also be copied to and from Spreadsheet and Write documents.

Managing your tables

You can rename, delete, export and import database tables in **Database Manager**.

- To **Rename** a table (or query, relation, form, or report), click once on the table name to select it, wait a moment and click once again. The table name will be displayed in Rename mode, surrounded by a box. Type the new name or click in the box and edit the current name. Click once outside the Rename box to close it. The new table name is displayed.
- To **Delete** a table (or query, relation, form, or report), right-click your mouse on the table name and select **Delete** from the menu. Ability will ask you if you are sure that you want to delete the table. Click on **Yes**. The table will be deleted from the database. Be careful when deleting tables: DOS and Windows techniques to recover files won't work if you want to recover a deleted table.
- For information on exporting tables, see [Exporting tables](#).
- For information on importing tables, see [Importing tables](#).

Redesigning a table

After you have designed your table and started to use it, that is to say, entered data in the fields, it is possible to change your design without having to start from the beginning. To change the structure of a database table – that is, to add, delete, redefine or reformat fields – do the following:

- If the database tree is on screen in **Database Manager**, click on the table you want to redesign to select it and then right-click your mouse. From the drop-down menu click on **Format**.
- If you are already in a table in **Grid View**, select **Table** from the **Format** menu.

The "Format Table" dialog is opened. This dialog should be familiar to you from when you first designed the table, although it was then called "New Table".

In the **Table Fields** box, in "Format table", you will see all the fields from the table, along with their associated **Name**, **Type**, **Description**. To go to the **Table Indexes** box, where all the indexes for the table are displayed, click on the **Index** button.

There are several things you might want to do when redesigning a table:

1. To delete a field completely (equivalent to deleting a column in the table), click on the left-hand row bar in **Table Fields** to select the field. Next, press **Del** on the keyboard and the selected field will be deleted. Several fields can be deleted at once, by dragging your mouse to select the relevant fields and then pressing **Del**. Be careful here: when a field is deleted the corresponding column in the table is deleted, *along with all its data*. So make sure you really no longer need this data, before deleting the field.
2. To add a field, go to the next empty line and create the field in the normal way (see [Creating fields in a database](#)).
3. To rename a field, just type the new name in **Name**.
4. To change the data type of a field, click on the arrow in **Type** and select a different type from the drop down list.
 - **Note:** When you select a different type, the data in the table column will be lost if the data types are incompatible, which they usually are. You should assume that different data types are incompatible when redesigning a field. However, there are exceptions: for example, if you have text in a column which has **Character** data type and you change the data type of the associated field to **Numeric**, the contents of the column will be deleted. If, however, the text is made up of numeral characters, then the characters will be converted to numbers. Likewise, numeric data will be converted to text when you change the field type from **Numeric** to **Character**.
5. When you change the data type of a field, you will usually want to format the field anew, or you may wish to reformat a field even though you have not changed the data type. The formatting properties include width, validation, initial value, caption, primary key and indexes, etc. depending on the data type of the field. To reformat a field, make sure you are somewhere in the field and then click on the **Format** button (see [Data types](#) for information on the various data types and their respective formatting options).
6. To change or add a description of the field, type the new description in **Description**.
7. If you think the redesigned table merits a different description, then type this in the **Table Description** box.
8. Once you have redesigned the table, click on **OK** to save it and return to Table Grid View. The table will be redrawn to reflect your changes and you can continue to enter data.

Formatting columns in a database table

Once you have designed your table and gone to Table Grid View - where you can begin to enter data - each field corresponds to a column. The formats you have already set for the fields – when designing the table - determine how data is displayed in the columns (see [Formatting database fields](#) for more information).

However, there is another level of formatting available that can only be applied once you are in Table Grid View. Like the field formats, these formats also apply to the *columns* in your table and work alongside the corresponding *field* formats. The column formats are accessed through the **Style** command in the **Format** menu. The format options are:

§ **Font**

§ **Colour**

§ **Borders**

§ **Align**

Since in a table only an entire column can be formatted, all you need to do to format the column is select the whole column by clicking on the bar at the top of the column or position the active cell anywhere in the column by clicking in any cell. You can also select several columns and format them together.

See:

[Formatting fonts in a table](#)

[Formatting a table for color](#)

[Formatting borders in a table](#)

[Alignment in database fields](#)

Formatting fonts in a table

Once you have selected a column (or columns), format the fonts as follows:

1. Select **Style** from the **Format** menu. The "Format Columns" dialog is displayed, turned to the **Font** page.
2. In the **Font** box choose a font for the contents of the column.
3. In the **Outline** box choose a style for the font.
4. In the **Size** box choose a size for the font.
5. In the **Effects** box you can switch on the checkboxes for **Strikeout** and **Underline**.
6. In the **Textcolor** box click on the arrow to display the full list of available colors and then click on the color you require. This option only colors the *contents* of the column, for instance, text or numbers.

The formats you have chosen are displayed in the **Preview** box.

Click on **OK** to apply the new fonts and styles to your table.

See:

[Formatting columns in a database table](#)

Formatting a table for color

Once you have selected a column (or columns), format them for color and 3-D effects, as follows:

1. Select **Style** from the **Format** menu. The "Format Columns" dialog is displayed.
2. Click on the associated tab to turn to the **Color** page.
3. In the **Interior** group box there are two color palettes: one for **Foreground** and one for **Background** color. The box just above these is where you choose the *pattern* that combines the foreground and background color. The pattern and colors you choose will fill the *whole* of the column you are formatting, with text displayed on top of the pattern (in the color you chose, if any, in **Textcolor** on the **Font** page).

Note: The default setting is white on white, that is to say, the white color button is depressed for both foreground and background. Obviously no pattern can be formed from this combination, as you can see from the absence of options if you click on the arrow in the pattern box. Likewise for any other color when it is selected in both palettes.

To form a pattern, proceed as follows:

1. Click on a color in the **Foreground** palette and one in the **Background** palette (note that white is also a valid color option for either or both of the palettes).
2. Click on the pattern box arrow and select one of the displayed patterns. The pattern and its color combination are shown in the pattern box and also in the **Preview** box. You can proceed to combine different colors by experimenting with the palettes and the available patterns.
3. In the **3D-Effect** group box choose whether you want the column to display its pattern in down mode (choose **Inset**), up mode (choose **Raised**), or normal mode (choose **Normal**).
4. The pattern, color and 3D effect are all displayed in the **Preview** box.
5. Click on the **OK** button to save and apply the new color formats.

See:

[Formatting columns in a database table](#)

Formatting borders in a table

Once you have selected a column (or columns), you can format the borders of *all* the cells in the column (or columns) as follows:

1. Select **Style** from the **Format** menu.
2. In the “Format Columns” dialog click on the **Borders** tab to go to the associated page.
3. Click on the **Type** and **Color** that you want to set for the borders. Click on the borders in **Border** for which you want to set the new type and color. As you click on these - **Left, Right, Top, Bottom** - one by one, the new formats will be displayed. If you wish to set border types and colors individually for the borders, click on the relevant border along with the type and color you want it to have, and so on till you have formatted all the borders.

Note that in setting borders in a particular table column all the cells in the column are affected.

See:

[Formatting columns in a database table](#)

[Changing the grid style](#)

Alignment in database fields

Once you have selected a column (or columns), you can determine how the data contained therein will be aligned or justified.

1. Select **Style** from the **Format** menu.
2. Click on the **Align** tab to go the associated page.
3. In the **Horizontal** box choose from **Standard**, **Left aligned**, **Center** and **Right aligned**.
4. In the **Vertical** box choose from **Standard**, **Top**, **Center** and **Bottom**.
5. Below these you can switch on **Wrap Text**, **Allow Enter** and **Auto Size**.

See:

[Formatting columns in a database table](#)

Formatting database fields

When you design a database table and create the fields for the table, you can also format the fields for such properties as:

- § **case** (whether text is to be upper, lower or mixed case)
- § **width** (the width of the field)
- § **precision** (how many decimal places a number can have)
- § **indexes** (primary key, unique index, or non-unique index)

and other format properties, depending on the data type of the field.

This is the *field formatting level*. (See [Field data types](#) for information on data types and their associated formatting properties.)

Once you have designed your table and gone to Table Grid View - where you can begin to enter data - each field corresponds to a column. The formats you have set for fields determine how data is displayed in the columns.

However, there is another level of formatting available that can only be applied once you are in Table Grid View. Like the field formats, these formats also apply to the *columns* in your table and work alongside the corresponding *field* formats. These include font style and size, background and foreground color, text color, alignment, and border styles (see [Formatting columns in a database table](#) for more information).

Changing the grid style

To alter the way the grid (the border around the field contents) is used in the table, do the following:

1. Select the **Options** command from the **Tools** menu and go to the **Grid Display** page in the “Options” dialog.
2. In the **Titles and GridLines** group box most of the Grid display options are switched on by default:
 - **3D-Buttons** - the bars at the top of the columns are 3D rather than flat
 - **Vertical Lines** - sets the vertical lines for the Table Grid
 - **Horizontal Lines** - sets the horizontal lines for the Table Grid
 - **Mark Current Row** - an arrow in the left hand row bar marks the current row
 - **Mark Current Column** - the bar at the top of the column is shown as up when column is not selected, otherwise down
3. In the **Color** group box you can select a color for the Table Grid. Click on the options for which you wish to set a color (**Grid Lines, Fixed Lines, Tracking Line, Dragging Line, Background**) and then select a color from the palette.
4. In the **User Properties** group box set the row and cell properties for the Table Grid:
 - **Row resize mode** (**Automatic** resizes all rows when you resize one row; **Single** allows you to resize single rows independently of other rows; **None** does not allow any row resizing)
 - **Row print mode**
 - **Current Cell** (choose a border thickness for the current cell i.e. the selected cell, from the drop-down list)
5. The changes you make are shown in the **Preview** box. Click on **OK** to redraw your table with the new Grid Style settings.

Note that the new grid style is applied not only to the current table but also to any other tables you may open or create.

Changing the column width

To change the width of a column:

1. Position the mouse pointer over the right hand edge of the column bar.
2. When the pointer turns to a cross, hold down the left mouse button and move the cross to where you want the edge of the column to appear.
3. Release the mouse button and the column will be redrawn with its new width.

The new column width will be saved with the table and so will not be lost when you leave the database.

Performing calculations in tables

You can extract certain types of data about the records and columns in a table, query or relation, as well as perform some calculations.

Select the **Data** option on the menu bar to access options for:

Count

Average

Sum

Calculate

Average, Sum and Calculate can be used with a *selection* of the records rather than the whole table by defining a filter condition on the **Condition** page (see [Performing calculations in a filtered table](#)).

- **Count** has options for **All Records** - counts the total number of records in the table (filters are disregarded); **Current Filter** - counts the number of records in the filtered table (if no filter has been applied the result is the same as for **All Records**); and **Marked Records** - counts the number of currently selected records. Select one of these and select **OK**.
- **Average** allows you to find the average value of one or more numeric columns. In the **Average** page of the "Calculate Totals" dialog, click on the fields (columns) in **Available Fields** for which you want to find the average, and click on **Add** to move them to **Selected Fields**. If you want to move a field into a position *between* fields that have already been moved over, click on the field in **Selected Fields** *above* which you want the field to be inserted and use the **Insert** button rather than **Add**. Click on **OK**. The result is displayed in a "Summary Results" dialog, with an average value for each selected column. You can print this or copy it to the clipboard. Click on **Done** to close the dialog when you have finished.
- **Sum** allows you to find the total value of one or more numeric columns. In the **Sum** page of the "Calculate Totals" dialog, click on the fields (columns) in **Available Fields** for which you want to find the sum, and click on **Add** to move them to **Selected Fields**. If you want to move a field into a position *between* fields that have already been moved over, click on the field in **Selected Fields** *above* which you want the field to be inserted and use the **Insert** button rather than **Add**. Click on **OK**. The result is displayed in a "Summary Results" dialog, with a total value for each selected column. You can print this or copy it to the clipboard. Click on **Done** to close the dialog when you have finished.
- **Calculate** allows you to perform various calculations on values in numeric columns. Select it to display the "Calculate Totals" dialog. In the **Expressions** group box you design the calculation you want to perform on a particular column:
 - i. In the **Field** box click on the arrow to display a full list of the numeric columns in the table. Click on a field (column) to place it in the **Formula** box.
 - ii. Build the formula by clicking on "+", "-", "*", "/", "(", ")" in turn, as required, to insert them into the formula box, and by typing directly into the **Formula** box any numbers needed.
 - iii. Note that you can place as many fields as you want in the **Formula** box. If you make a mistake double-click on the inserted symbol, number or field to select it and then right-click and select **Delete** (or press **Del** on the keyboard).
 - iv. Once you have constructed the formula, click on the **Add** button to insert it into the large box at the bottom of the dialog.
 - v. Click on **Test** to test that the formula is well formed, that is to say, it doesn't break any of the syntax rules for a formula, such as an unmatched bracket or a "+" sign with no expression following it.
 - vi. To clear the formula click on the **Clear** button. To remove selected expressions, first select the expressions to be removed and then click on **Remove**.

- vii. Finally, click on **OK** to apply the formula.
- viii. A "Summary Results" dialog is displayed, with the results of the formula shown in a column. You can copy this to the clipboard or print it. When you have finished, click on **Done** to close the dialog and return to the table.

You can also perform a calculation on a limited number of the records by using **Calculate** in conjunction with filter conditions you set in the **Condition** box.

Example of a formula

If you have two numeric fields, called Income and Outgoings, in a table called Thrift, which records the state of your monthly finances, you can apply the following formula to your table to work out the percentage of your income that you spend month by month:

$(\text{Thrift.Outgoings} / \text{Thrift.Income}) * 100$

The formula will produce a summary column, displaying the monthly ratio of spending to income on a percentage basis.

Performing calculations in a filtered table

You can calculate averages and sums and perform various other calculations on the columns in a table (see [Performing calculations in tables](#)). In addition, you can carry out calculations on a *selection* of the records rather than all of them by defining a filter condition in conjunction with the calculation.

When you select **Average**, **Sum** or **Calculate** from the **Data** menu, the “Calculate Totals” dialog is displayed with a **Condition** tab. This allows you to average, sum or perform calculations on a *subset* of the records in the table. Basically, the **Condition** box enables you to *filter* a table according to conditions you have specified and then extracts data from the filtered table according to which fields you selected in Average, Sum or Calculate. A Condition is always set in conjunction with one of Average, Sum or Calculate.

To set a **Condition** follow these steps:

1. First choose the *fields* you want to perform the calculation on (see [Performing calculations in tables](#))
2. Next, click on the **Condition** tab to go to the page where you design the condition expression.
3. The three boxes along the top line - **Field**, **Operator**, **Value** - are used to define a filter condition for each of the fields that is to be used in the filter. Ability takes the field you select and compares all the values in this field to the condition you specify for the filter. The filter selects only those records where the field value fulfils the specified condition.

- Click on the arrow in the **Field** box to display the full list of fields in your table and then click on one of the fields to select it.
- Click on the arrow in the **Operator** box to display the full list of operators that can be used to construct a condition. These include operators for:

- Equals (=)
- Contains (\$)
- Begins With (<\$)
- Ends With (>\$)
- Less Than (<)
- Greater Than (>)
- Less Than or Equal To (<=)
- Greater Than or Equal To (>=)
- Not Equal To (< >)
- Is Empty ("")
- Not Empty (!!)

- In the **Value** box you type the value that is to be compared, using the operator, to the values in the field you selected in **Field**.

For example, in a table that contains details on customers, you might select, say, Customer_Name in **Field** and < in **Operator**, and type S in **Value**. This would filter the table to show only the records where the Customer_Name begins with any letter less than S, that is, any letter before S in the alphabet.

4. Next, you need to insert the condition into the **Condition** box. Click on the **Insert** button to do this.
5. You can construct more complex conditions by using the **Group Start**, **And**, **Or** and **Group End** buttons. These allow you to use **And** and **Or** to link condition expressions, and to control the order of their evaluation using opening and closing brackets **Group Start (** and **Group End)**. Whenever you insert a condition into the **Condition** box, it is placed *after* the current contents of the box. Therefore, if you have already inserted a condition, you need to insert either an **And** or an **Or** before inserting the next condition. Also, judicious use of brackets will ensure that complex condition expressions are evaluated properly.
6. Once the conditions making up a filter are inserted and the relevant brackets have been placed, you can test the filter to check that it is well-formed, that is to say, it follows the syntax rules or

"grammar" for a condition expression. Click on **Test**. If the condition expression is well-formed a message is displayed: "The Query is Correct!"; if it is not well-formed, an error message is displayed with an indication of what the error is e.g. brackets don't match, syntax error, etc. Note that a well-formed filter expression is merely "grammatically" correct, but doesn't necessarily mean what you want it to mean. For that you need to make sure the filter is well designed according to the purposes you set for it.

7. To delete the contents of the **Condition** box as a whole, click on the **Clear** button. If you only want to delete selected text, first select the text and then right-click your mouse and select **Delete** (or press **Del** on the keyboard).
8. After you have constructed the filter from the constituent conditions and tested it to check that it is well-formed, click on **OK** to apply the condition to your table. The calculation you defined in **Average**, **Sum** or **Calculate** will only be performed on the filtered table.

Note: The conditions set through the **Data** menu in this way are temporary and cannot be saved. If you wish to perform certain calculations regularly on a particular set of selected records, it may be better for you to design a named filter and apply it to your table *before* using the options in the **Data** menu. In this case you would not have to bother setting any conditions in the **Condition** box of the "Calculate Totals" dialog.

See:

[Filtering using a named filter](#)

Rearranging columns using the mouse

You can rearrange columns in a database table. This is useful when you need to have two or more columns visible together on screen. To move a column:

1. Click once on the bar above the table to select the whole column.
2. Click again but hold the mouse button down. A small rectangle is displayed.
3. Continue to hold down the mouse button and move the rectangle to the column immediately to the right of where you want the selected column to be positioned.
4. Release the mouse button and the table will be redrawn with the columns rearranged.

You can rearrange as many columns as you like.

Note: Ability allows you to rearrange columns as an aid to working efficiently with your tables. It does not, however, overwrite the field order you established when you first designed the table. Any rearrangement of the columns will be lost when you close the table and the original order will be reinstated. However, if you want to re-use a particular column rearrangement, you can create a **Columns** selection that can be applied to your table at any time.

See:


[Rearranging columns using a columns selection](#)

Rearranging columns using a columns selection

Ability allows you to rearrange columns in a table. This can be done very easily using your mouse (see [Rearranging columns using the mouse](#)). However, whenever you close a table the original column order, which you set when designing the table, is reinstated.

If particular column rearrangements are to be used frequently, it is a good idea to create a columns selection. The selection can then be applied to the table at any time. The rearrangement defined by the selection will be applied until you close the table or apply another selection.

To create a columns selection, do the following when in Table Grid View:

1. Click on the **Columns**  button in the **Query** toolbar (alternatively, select **Columns** from the **Format** menu). The "Format Query" dialog is opened at the **Columns** page. In the right-hand box, **Selected Fields**, all the fields in the table are listed.
2. Click on **Remove All** to move all the fields to the **Available Fields** box.
3. You can now start moving the fields back to the **Selected Fields** in the order you want. Click on the field you want to be first in the table and then click on the **Add** button. The field is then moved back to the **Selected Fields** box.
4. Click on the field you want to be second in the table and again click on **Add**. The field is moved back to **Selected Fields** and placed after the first field.
5. Repeat the operation till all the fields have been moved and placed in the order you want.
6. You can now give the column rearrangement its own name. Type a name in the **Name** box and then click on **OK**. You will be asked if you want to create a new column definition. Click on **Yes**.

You will be returned to the Table Grid View with the new column arrangement applied.

Note that the name of the columns selection appears in the box to the left of the **Columns** button on the **Query** toolbar. When you next want to apply this columns selection, all you have to do is click on the arrow at the right of the box and select the appropriate columns selection from the drop-down list.

Hiding and showing columns in a database table

While you are working in a table you might find it convenient to hide one or more columns. This is useful if you are currently working with only a few of the columns and find the others a temporary distraction.

- To hide a column right-click your mouse anywhere within the column. Select **Hide Columns** from the displayed menu. The table will be redrawn on screen with the column hidden from view.
- To hide several adjacent columns in one go, click and drag along the column bars to select the appropriate columns. Next, right-click and select **Hide Columns** from the displayed menu. The selected columns will be hidden.

Ability prevents you from hiding *all* the columns by ensuring that at least one column remains visible, but you are free to hide all the rest.

To show hidden columns, do this:

1. Right-click your mouse and select **Show Columns**. The **Columns** page of the "Format Query" is displayed. The hidden columns are shown in the **Available Fields** box.
2. Click on a column that you want to show again and then click on **Add**. The column will be moved to the **Selected Fields** box.
3. Repeat the operation till all the columns you want to show are moved to **Selected Fields**.
4. If you want to show all the hidden columns, click on **Add All**.
5. Click **OK** and the table will be redrawn with the selected columns shown.

Ability will hide columns until you choose to show them again or until you close the table. When you next open the table all previously hidden columns will be displayed once again. However, if you think you might want to hide the same columns frequently, you can create a columns selection, which can be applied at any time to your table.


See:

[Hiding and showing columns using a columns selection.](#)

Hiding and showing columns using a columns selection

Ability allows you to hide columns in a table. This can be done very easily as you work in your table (see [Hiding and showing columns in a database table](#)). However, whenever you close a table all the columns, which you defined when designing the table, are reinstated. If you want to hide the same columns frequently, it is a good idea to create a columns selection. The selection can then be applied to the table at any time. The columns selection will be applied until you close the table or apply another selection.

To create a columns selection to hide select columns, do the following when in Table Grid View:

1. Click on the **Columns** button  in the **Query** toolbar (alternatively, select **Columns** from the **Format** menu). The "Format Query" dialog is opened at the **Columns** page. In the right-hand box, **Selected Fields**, all the fields in the table are listed.
2. Click on a column that you want to hide, followed by **Remove**. The column will be moved to the **Available Fields** box. Repeat the operation till you have moved, from **Selected Fields** to **Available Fields**, all the columns you want hidden. The columns remaining in **Selected Fields** will be shown; those in **Available Fields** will be hidden.
3. You can give this columns selection its own name. Type it in the **Name** box.
4. Click on **OK** and then **Yes**, when you are asked if you want to define a new columns selection.

You will be returned to the Table Grid View with the new columns selection applied. Note that the name of the columns selection appears in the box to the left of the **Columns** button on the **Query** toolbar. When you next want to apply this columns selection, all you have to do is click on the arrow at the right of the box and select the columns selection from the drop-down list. The appropriate columns will then be hidden.

Note that the converse of hiding selected columns in a table is that you show the remainder. This columns selection could just as well be defined as a selection of columns to be shown, rather than as columns to be hidden.

To show columns that you have previously hidden, click on the arrow in the **Columns** box on the **Query** toolbar and select **Columns - All** from the drop-down list. This restores all of the columns in the table.

If you don't want to show all the hidden columns follow these steps:

1. Right-click your mouse and select **Show Columns** from the menu. The **Columns** page of the "Format Query" dialog is displayed.
2. Click on the columns to be shown and click on **Add** to move them to **Selected Fields**.
3. Click on **OK**. You will be asked if you want to save the column definition. Click on **No**. (Only click on **Yes** if you want to change the definition of the original columns selection that was used to hide selected columns).

The table will be redrawn with the selected columns shown.

You can define a columns selection to show particular columns, which can then be applied to your table whenever you like.

Sorting a table

You can sort your table records alphabetically or numerically according to the values of key fields. In Ability sorting works in conjunction with the indexes and primary keys you have set for fields in your table. An indexed field will sort faster than one without an index. Also, when determining which fields to sort on, you need to take into account the types of indexes you have assigned to the fields. For instance, non-unique indexes allow duplicate values, and so you may wish to sort on a second field, in addition to the first, when it is possible for the first to contain duplicates (see [Indexes and primary keys](#)).

You can perform a simple sort on a single field or more complex sorts on several fields at once. In addition, since any sort applied to a table is undone when the table is closed, you can save a sort as a **Sort Order** and re-apply it whenever you choose.

See:

[Sorting on a single field](#)

[Sorting using a sort order](#)

Sorting on a single field

To do a quick sort on a single field, follow these steps:

1. Go into Table Grid View and click on the bar above the column (field) you want to sort on. The column will be selected.
2. Next, right click your mouse and select **Sort Ascending** or **Sort Descending**. The records will be sorted according to the values in the column. (You can also right click your mouse anywhere within the column without having first selected the column, except in the current cell if it is in edit mode).

If you repeat the operation on another column, the records will be sorted according to the values in that column without regard for the first sort.

Note: The **Sort Order** box on the **Query** toolbar indicates that the sort is **Sort Order - Untitled**. This is the name given to any temporary sorts you apply to your table and is redefined for each temporary sort you apply.


To cancel a sort and return to the original order of the records, right click anywhere in the table and select **No Sort** (alternatively, click on the arrow in the **Sort Order** box and select **Sort Order - None** from the drop-down list). Closing the table also cancels the sort. If you want to re-apply a particular sort order, it is a good idea to save it as a named **Sort Order** (see [Sorting using a sort order](#)).

Note: If you have set a Primary Key for your table, canceling a sort returns you to the order imposed by the Primary Key.

Sorting using a sort order

A **Sort Order** allows you to sort on several fields at once. In addition, you can save a sort order and re-apply it at will.

To create a sort order for a table, first go into the table and then follow these steps:

1. Click on the **Sort Order** button  on the **Query** toolbar (or select **Sort Order** from the **Format** menu). The "Format Query" dialog is opened at the **Sort Order** page. All the fields of the table are initially listed in the **Available Fields** box. You need to select the fields you wish to sort on - and in the right order, to determine the priority for each of the sort fields.
2. Select the first field by clicking on it and then click on **Add**. The field will be moved to the **Selected Fields** box.
3. Select the next field and again click on **Add** to position it after the first field.
4. Repeat the operation till all the fields you wish to sort on have been moved to **Selected Fields**. The first field has priority one; the second field priority two; and so on.
5. If you decide that you don't want to sort on a particular field or you have given a field the wrong priority, just click on the field in **Selected Fields** and click on **Remove**.

The sort proceeds in this way:

First sort: the first field is sorted - this is the main sort, with priority one.

Second sort: the second field is sorted, but only if there were duplicate values in the first field i.e. the second field is sorted within the groups determined by the duplicate values in the first field.

Third sort - the third field is sorted, but only if there were duplicate values within the groups in the second; and so on.

Note: This has the consequence that if your table has a primary key or a unique index (or indexes) i.e. no duplicate values are allowed, and this is the first field to which a sort is applied, then sorting on the other fields can have no effect, no matter what type of indexing they have. The basic rule is that it makes sense to sort up to a primary key or unique index field, but not beyond. So if the first two fields, say, in a sort order are not indexed or have a non-unique index, and the third has a primary key or unique index, you would sort up to the third field but not beyond.

6. Once you have decided which fields to use in the sort order and the order of sorting, you can determine whether each of the fields is to be sorted in ascending or descending order. To do this click on each of the fields in turn and click on the **Descending** checkbox to impose descending order. The default is ascending, which is indicated by an unchecked **Descending** checkbox. Ascending order is A...Z, 1...n, descending order Z...A, n...1, as you would expect.
7. Give the sort order a name in the **Name** box and click on **OK**. You will be asked if you want to create a new sort order. Click **Yes** to save the sort order under the name you have given it.

You are returned to the Table Grid View with the sort order applied. The name of the sort order is displayed in the **Sort Order** box on the **Query** toolbar. Whenever you need to apply this sort order in future, just click on the arrow in the **Sort Order** box and select the sort order from the drop-down list.

Note: If you don't give a name to the sort order, the sort order you have defined will still be applied but it won't be saved for future use. In this case the **Sort Order** box on the **Query** bar will say **Sort Order - Untitled**, which is the title given to the last applied unsaved sort.

Filtering a table

Filtering is a way of selecting records from (or creating subsets of) a database table, according to values that you specify for certain fields. The filter can be either a simple filter on a single field or a more complex filter on several fields at once. You can save and name complex filters for future use.

Moreover, you can fine tune a named filter to apply *conditions* before selecting the records. Since any filter applied to a table is undone when the table is closed, you can save a named filter as a **Filter** and re-apply it whenever you choose.

See:

[Filtering on a single field](#)

[Filtering using a named filter](#)

Filtering on a single field

To do a quick filter on a single field, follow these steps:

1. Go into Table Grid View.
2. Choose a value in a particular column as the filter value.
3. Right-click your mouse on the cell that contains the chosen value and select, from the drop-down menu, one of the following:
 - **Filter Using Selection** (choose one of the following operators to make a comparison with the current selected cell e.g. if the cell contains 10 and the operator selected is >, then only the records that have a number greater than 10 in this field will be selected)
 - = Equal
 - > Greater Than
 - < Less Than
 - >= Greater Than or Equal
 - <= Less Than or Equal
 - <> Not Equal
 - **Filter Excluding Selection** (select only the records that *do not include* this value in this field/column).

The table will be redrawn to show only those records allowed by the filter.

Another filter can be applied on top of the first one:

1. Again right-click your mouse on a value in a particular column (this can be the same column as before or different).
2. Select either **Filter Using Selection** or **Filter Excluding Selection**.

You can apply as many filters as you like, each being applied to the records remaining on view after the application of the previous filter.

To clear a filter or succession of filters, right-click anywhere within the table and select **Remove Filter** from the drop-down menu (alternatively, click on the arrow in the **Filter** box and select **Filter - None** from the drop-down list). Closing the table will also clear all filters.

Note: The **Filter** box on the **Query** toolbar indicates that the filter is **Filter - Untitled**. This is the name given to any temporary filters you apply to your table and is redefined for each temporary filter or succession of filters you apply.

If you want to re-apply a filter you can save it as a named filter, which can then be used as many times as you wish.


See:

[Filtering using a named filter](#)

Filtering using a named filter

A *named filter* allows you to filter the records in a table and to save the filter for future use. You may also filter according to the values in several fields at once. In addition, you can specify conditions for the filter in order to fine-tune the record selection. Named filters can be saved and re-applied whenever you like. Note that if you only wish to filter according to the values in a single field, it is easier and quicker to use your mouse (see [Filtering on a single field](#)).

To create a named filter for a table, first go into the table and then follow these steps:

1. Click on the **Filter** button  on the **Query** toolbar (alternatively select **Filter** from the **Format** menu). The "Format Query" dialog is opened at the **Filter** page. The final filter expression will be contained in the **Condition** box but built up using the **Condition Item** box.
2. In the **Condition Item** group box you define a filter condition for each of the fields that is to be used in the filter. Ability takes the field you select and compares all the values in this field to the condition you specify for the filter. The filter selects only those records where the field value fulfils the specified condition.
 - Click on the arrow in the **Field** box to display the full list of fields in your table and then click on one of the fields to select it.
 - Click on the arrow in the **Operator** box to display the full list of operators that can be used to construct a condition. These include operators for:
 - Equals (=)
 - Contains (\$)
 - Begins With (<\$)
 - Ends With (>\$)
 - Less Than (<)
 - Greater Than (>)
 - Less Than or Equal To (<=)
 - Greater Than or Equal To (>=)
 - Not Equal To (< >)
 - Is Empty ("")
 - Not Empty (!!)
 - In the **Value** box you type the value that is to be compared, using the operator, to the values in the field you selected in **Field**. Alternatively you can click on the arrow and choose a field from the drop-down list. In this case, the values of the field in **Field** will be compared to the values of the field in **Value**, for each record in the table.

For example, in a table that contains details on customers, you might select, say, Customer_Name in **Field** and < in **Operator**, and type S in **Value**. This would filter the table to show only the records where the Customer_Name begins with any letter less than S, that is any letter before S in the alphabet.

3. Next, you need to insert the condition you have specified into the **Condition** box. Click on the **Insert** button to do this.
4. You can construct more complex conditions by using the **Start**, **And**, **Or** and **End** buttons. These allow you to use **And** and **Or** to link condition expressions, and to control the order of their evaluation using opening and closing brackets (**Start** and **End**). Whenever you insert a condition into the **Condition** box, it is placed *after* the current contents of the box. Therefore, if you have already inserted a condition, you need to insert either an AND or an OR before inserting the next condition. Also, judicious use of brackets will ensure that complex condition expressions are evaluated properly. If you make a mistake, you can edit the condition expression in the normal way by right-clicking your mouse.
5. Once the conditions making up a filter are inserted and the relevant brackets have been placed, you can test the filter to check that it is well formed, that is to say, it follows the syntax rules or "grammar" for a condition expression. Click on **Test**. If the condition expression is well formed a

message is displayed: "The Query is Correct!"; if it is not well formed, an error message is displayed with an indication of what the error is e.g. brackets don't match, syntax error, etc. Note that a well-formed filter expression is merely "grammatically" correct, but doesn't necessarily mean what you want it to mean. For that you need to make sure the filter is well designed according to the purposes you set for it.

6. You can update or replace a particular condition. Select the condition you want to replace, including its surrounding brackets, in the **Condition** box and create a new condition in the **Condition Item** box. Click on the **Update** button. The selected condition will be replaced by the new condition. Note that everything you select in **Condition** will be replaced by the *one* condition you specify in **Condition Item**, so be careful to select only the text that you want replacing before selecting **Update**.
7. To delete the contents of the **Condition** box as a whole, click on the **Clear** button. If you only want to delete selected text, first select the text and then right-click your mouse and select **Delete** (or press **Del** on the keyboard).
8. After you have constructed the filter from the constituent conditions and tested it to check that it is well formed, you can choose whether you want the filter to replace, act on top of or be added to a currently applied filter. You do this in the **Filter Action** box at the top right corner of the dialog. Click on the arrow and select one of **Replace**, **And** or **Or** from the drop-down list:
 - **Replace** is the default. Any filters currently applied to your table are ignored and the new filter is applied to *all* the records in the table.
 - **And** causes the new filter to be applied only to those records that are already selected by a currently applied filter.
 - **Or** causes the new filter to be applied to all the records in the table but will also leave selected any records that are already selected by a current filter.
9. Finally, you can give the filter a name in **Name**.
10. Click on **OK** to apply the filter. You will be asked if you want to create a new filter. Select **Yes** if you want the filter to be saved for future use under the name you have given it.

You are returned to Table Grid View with the filter applied and the filter name displayed in the **Filter** box on the **Query** toolbar.

When you next want to apply the filter, just click on the arrow in the **Filter** box and select the filter name from the drop-down list.

See:

[Filter example](#)

Filter example

This example applies some of the principles of filter design explained in [Filtering using a named filter](#).

Note: If you are new to filters, don't be put off by the length of the example. Since all of the conditions that go into making the filter are created in a similar manner, most of the example consists of "variations on a theme". Nevertheless, it should make clear to you the importance of thinking about the structure of a filter before actually creating it, the proper use of bracketing, and the difference between AND and OR.

Imagine you have a table, called *Contacts*, with three fields, viz. *Name*, *Age*, and *Occupation*. You want to construct a filter that will display all contacts who are lepidopterists and more than 70 years of age, and whose names are either Smith or Jones (perhaps you wish to invite a select company to a private viewing of your butterfly collection).

The problem is to find all records such that:

(Occupation = Lepidopterist)

AND

(Age > 70)

AND

(Name = Smith OR Name = Jones).

Note that the two ANDs are specified *across* fields, that is we want to select records which contain field values that satisfy the first condition AND the second condition AND the third condition together. The one OR, which occurs within the third condition, is specified *down* records, that is we want to select only those records that have *either* 'Smith' *or* 'Jones' in the Name field.

In the example, I have put AND and OR in capitals to give you an idea of the logical structure of the final condition expression. This requires us to create four conditions in total: three main conditions connected by ANDs, where the third condition is constructed out of two sub-conditions connected by OR.

In the **Condition Item** box, create the first condition:

1. Click on the arrow in **Field** and select *Contacts.Occupation*.
2. Click on the arrow in **Operator** and select = *Equals*.
3. Click in the **Value** box and type *Lepidopterist*.
4. Click on the **Insert** button to insert the condition into the **Condition** box below.

The expression inserted is:

(Contacts.Occupation = 'Lepidopterist').

Note that the expression, as a whole, is surrounded by brackets, the typed value is surrounded by single quotes, and the expression is selected.

5. Next, you need to insert an AND. AND is inserted at the cursor position and will replace any selected text, so click at the end of the expression to deselect it and then click on the **And** button. This gives:

(Contacts.Occupation = 'Lepidopterist') AND.

6. In the **Condition Item** box create the second condition:
7. Select *Contacts.Age* in **Field**.
8. Select > *Greater Than* in **Operator**.
9. Type 70 in **Value**.
10. Again click on **Insert**.

In the **Condition** box this gives:

(Contacts.Occupation = 'Lepidopterist') AND (Contacts.Age > 70).

Note that 70, as a numerical value, is not surrounded by quotes.

11. Click at the end of the compound expression and click on **And** to get:

(Contacts.Occupation = 'Lepidopterist') AND (Contacts.Age > 70) AND.

In the Condition Item box you can now create the third and most complex condition:

12. First, select *Contacts.Name* in **Field** and = *Equals* in **Operator**, and type *Smith* in **Value**.

13. Click on **Insert** to insert the condition.

14. In the **Condition** box click at the end of the compound expression and click on **Or**. This gives:

(Contacts.Occupation = 'Lepidopterist') AND (Contacts.Age > 70) AND (Contacts.Name = 'Smith') OR.

In the **Condition Item** box create the second part of the OR expression:

15. Select *Contacts.Name* in **Field**.

16. Select = *Equals* in **Operator**.

17. Type *Jones* in **Value**.

18. Click on **Insert**.

In the **Condition** box this gives:

(Contacts.Occupation = 'Lepidopterist') AND (Contacts.Age > 70) AND (Contacts.Name = 'Smith') OR (Contacts.Name = 'Jones').

All the relevant conditions have now been inserted to form a compound condition expression. However, you must now insert brackets in the right places to make sure that the expression is evaluated properly. At present a filter using this expression will correctly select all the Smiths who are lepidopterists aged over 70, but will incorrectly select all Joneses, no matter what their occupation or age. This is because the OR has not been restricted to work only within the *scope* of the third - joint - condition. Therefore, you need to surround the two conditions connected by OR by another pair of brackets.

19. Click immediately before *(Contacts.Name = 'Smith')* and then click on **Start (** to insert an opening (left-hand) bracket.

20. Click immediately after *(Contacts.Name = 'Jones')* and then click on **End)** to insert a closing (right-hand) bracket.

This gives you the final compound condition expression for your filter:

(Contacts.Occupation = 'Lepidopterist') AND (Contacts.Age > 70) AND ((Contacts.Name = 'Smith') OR (Contacts.Name = 'Jones')).

Rearranging it on the page should make the structure of the condition expression clearer:

(Contacts.Occupation = 'Lepidopterist')

AND

(Contacts.Age > 70)

AND

((Contacts.Name = 'Smith') OR (Contacts.Name = 'Jones')).

Compare this with the outline of the problem I gave at the start of this example.

Note that using **Test** will indicate a well-formed expression for both the correct version and the previous one with the extra brackets missing. This is because both expressions are perfectly legitimate, though they give different results. In addition, when I put in the extra brackets after inserting all the conditions, I did this to show you the difference brackets make to evaluation. There is nothing to stop you putting in

extra brackets immediately they are needed.

Queries

A query is a way of applying various *combinations* of query components to tables and relations:

§ **sort orders**

§ **filters**

§ **column views**

§ **SQL statements**

Queries are useful in that they allow you to apply their components all in one go, rather than having to individually apply each one. The outcome of a query is a table in its own right. The query table is live and so reflects the latest state of the table or relation on which it is based. Queries can be saved, thereby allowing you to apply them whenever you need to.

To apply a query you first need to have created a table in your database. You can create a query from **Database Manager**, and you can then apply it whenever you need to. Alternatively you can create a query while you are working in a table, and the query can then be applied immediately.

See:

[Creating and applying a query](#)

[Creating a query while working in a table](#)

Creating and applying a query

The following steps show you how to create a query and then apply it to a table (you follow the same steps when you want to apply a query to a relation):

1. Go into **Database Manager**.
2. Right-click your mouse and select **New** and then **Query**. The "Format Query" dialog is opened. This is where you design the new query.
3. In the **Based On** box click on **Table** (or on **Relation** if you want to base the query on a relation) and then click on the arrow in the box just to the right. The drop-down list displays all the tables in the current database. Click on one of these to select it. This is the table that the query will be based on.
4. Next you need to select the query components that you want to apply to the table. It may be that you have already created some of these separately in **Columns**, **Sort Order** or **Filter**. If so, you can click on the arrows in the boxes in the **Select Query Components** group box. In each case the existing components will be displayed in a drop-down list. Click on the required components to select them.
5. However, you may need to design the query from scratch. Notice that the "Format Query" dialog has pages for Columns, Sort Order and Filter, as well as Query. Click on the respective buttons at the top of the dialog to access these pages in turn and to design the query components. For information on designing query components, see these sections:

[Rearranging columns using a columns selection](#)

[Hiding and showing columns using a columns selection](#)

[Sorting using a sort order](#)

[Filtering using a named filter](#)

6. After you have designed and named a query component on a particular page, click on a button for one of the other pages to design another component. You will be asked if you want to create a new filter/column_definition/sort order, depending on the page. Click on **Yes** (alternatively click on **OK** if you want to apply the component immediately and close "Format Query").
7. You can return to a page as many times as you like to design as many components as you think you might need for future queries based on this table.
8. Once you have finished designing the query components, go back to the Query page of "Format Query".
9. In the **Select Query Components** group box choose the components that you want the query to apply from the respective drop-down lists in the boxes. All the query components you have just designed will appear in the lists.
10. Finally, give the query a name in **Query Name** and click on **OK**.

The query will be applied to your table and a **Query table** created. This is based on the original table, with the component conditions of the query applied.

In **Database Manager** the new query will appear under **Queries**. Whenever you are in the database you can select the query directly without having to open the original table first. The query will always reflect any changes you have made to the table. Note that the individual query components - sort orders, column views, filters - are listed under the table from which the query was designed.

Select the **View SQL** button to design more complex queries (see [SQL Guide](#) for more information).

Creating a query while working in a table

If you want to apply a new query while working in a table, follow these steps:

1. Select **Query** from the **Format** menu to open up the "Format Query" dialog. Note that the table name is already selected in the **Based on** box. The table name is also grayed out to show that the query you are about to create must be applied to this table, that is you can't choose another table (or relation).
2. Select the required query components, if there be any, in the **Select Query Components** box, or use the individual component pages to design new components, as explained in [Creating and applying a query](#).
3. If you want to use this particular query again, then give it a name in **Query Name** and click on **OK** (otherwise leave the name blank).

The query will be applied to the table, but you will remain in Table Grid View. If you gave the query a name, it will appear under **Queries** in **Database Manager**. Thereafter it can be used just like any other query; that is to say, if you select it again in future it will open up a **Query table** based on the original table, without you having to go into the table first.

Editing a query

To edit a query right-click on the query name in **Database Manager** and then select **Format**. The "Format Query" dialog is displayed. Alternatively, if you are already in a table, query table or relation, you can select **Query** from the **Format** menu.

You can edit a query by going to the **Query** page in "Format Query". If you change the name of the query, the edited query will be saved under the new name and the original retained. If you then wish to delete the original, you can do so in **Database Manager** (see [Deleting a database object](#)).

To edit a query without changing its name, that is to say, the edited query is saved under its existing name, use the drop-down lists in the **Select Query Components** group box to select the new query components (select **None** if you want to exchange a selected component for no other component).

Note that in editing a query you can go to the other pages in the "Format query" dialog and edit or create new query components as required. See [Editing query components](#).

When you have finished editing the query click on **OK**.

Editing query components

You can edit a query component in the "Format Query" dialog. This has pages for Columns, Sort Orders and Filters, as well as Queries (see [Editing a query](#)).

There are various ways of opening the dialog:

- In **Database Manager** right-click on the query component name (this will be under the relevant table or relation name either in **Column Views**, **Sort Orders**, or **Filters**) and select **Format**.
- If you are already in a table, query table or relation, you can select **Query**, **Sort Order**, **Columns** or **Filter** from the **Format** menu. Alternatively click on the relevant button on the Query toolbar.

You can edit the query component by selecting the component name in the **Name** box and making changes as required in the body of the dialog. When you click on **OK** or move to another page in "Format Query", you will be asked if you want to *save* the edited component. Click on **Yes** and the component will be saved with the changes made.

If you also change the *name* of the component, you will be asked if you want to *create* a new component. In this case the edited component will be saved under its new name and the original component will not be changed.

If you want to delete the original you can do this in **Database Manager** (see [Deleting a database object](#)).

Joining tables

When designing tables in Ability, you will often find that you are about to create a field for information that is already held in another table. For example, you may have a table that contains customer account information and another that contains invoice details. It would be inefficient and unnecessary to have customer names and addresses in both tables. In this case you would create a field, say for a unique customer account number, that both tables could have in common, and put the names and addresses of the customers in the customer accounts table alone.

In order to relate such tables, Ability allows you to use the device of **Relations** to join tables. Joined tables must have at least one field in common, called the *join field*, by which they are related, thereby making it possible to view the contents of two tables together. The resulting table is called a **Relation** and you are allowed to apply **Queries**, **Column Views**, **Sort Orders** and **Filters** to it, just as you can with normal tables.

You can join more tables to the tables you have already joined, with the proviso that each join is between common fields for adjacent tables. For example, a name field in one table may be joined to a name field in a second table, and an age field in the second table may be joined to an age field in a third table, even though the first and third tables do not share a common field.

Note: The term *common field* requires that the joined fields have at least the same *data type*. For a join to make sense the fields should also contain similar data, that is to say, there should be the *possibility* that some of the records in the joined tables contain exactly the same data in the joined fields, otherwise there can be no match.

The data contained in a relation is a *live* reflection of the data contained in the tables that are joined in the relation, that is to say, any change of data in the original tables is reflected in the related table. Likewise, any changes you make in the related table cause similar changes in the original tables. You can therefore *update* several tables while working only in the one related table.

Whenever you join tables, *all* the *fields* from the original tables are shown in the relation. However, what *records* are shown from each of the original tables depends on the *type of join* you use. There are three options:

- **Inner** - only the records that have matching values in their join fields are shown
- **Left outer** - all the records in the left hand table are shown but only the matching ones from the right hand table
- **Right outer** - all the records in the right hand table are shown but only the matching ones from the left hand table

See:

[Creating a relation.](#)

Creating a relation

To create a relation you should have already created at least two tables in the current database. In addition, the tables should have at least one field in common. See [Joining tables](#) for more information.

Note: It is important that the tables to be related have their *join-field*, that is the field that is to be common to the tables, set as either a primary key (in the case of the left-hand table) or as a non-unique index or no index (in the case of the right-hand table). See [Indexes and primary keys](#) for more information. Indexes can be used to control what data is entered into a field: for instance, allowing no duplicates in a primary key field or as many duplicates as you like in an indexed or no index field. Obviously the use of indexes facilitates a relation between, say, a customer table, which wouldn't normally contain more than one record per person, and an invoice table, which will often contain several records for the same customer.

Follow these steps to create a relation between two tables:

1. In **Database Manager** right-click your mouse and then select **New** followed by **Relation**. The "New Relation" dialog is displayed.
2. In the **Relation Structure** box you decide which tables are to be joined, which fields are to be used as the join fields, and the type of join to be applied. Each line represents one join from one table to another.
3. Click in the **Left Table** field and click on the arrow to display a drop-down list of all the tables in the current database. Click on one of these to select it as the left table. Notice that the large left-hand box, just above the **Relation Structure** box, now displays all the fields in the table, with the name of the table appearing just above the box.
4. Next, click in the **Join Field** and click on the arrow to display a drop-down list of all the fields in the left table. Click on the field that you want to use as the join field.
5. Click on the arrow in the **Right Table** field to display a drop-down list of all the tables in the database, excluding the table already selected as the left table. Select one of these as the right table.
6. Click in the **Join Field** and select the field from the right table that is to be joined to the join field from the left table. Notice that a connecting line is drawn between the two join fields, *from* the left table box *to* the right table box.
7. Finally, select the type of join from the **Join Type** box. The default is **Inner**, with **Left outer** and **Right outer** as alternatives.
8. You have now defined a relation between two tables. Give the relation a name in the **Relation Name** box and click on **OK**. You are returned to **Database Manager**, with the new relation listed under **Relations**.
9. Open the relation by double-clicking on the relation name in **Database Manager**.

The relation table is displayed, along with all the fields from the two tables and the records allowed by the join fields and join type you chose. See [Joining tables](#) for more information on types of join.

At this point, if you want to join another table to the new relation, select **Relation** from the **Format** menu. This returns you to the "Format Relation" dialog. Notice that the table you chose as **Right Table** in the first line of the **Relation Structure** box is selected automatically as **Left Table** in the second line. Proceed along the line, selecting the table, join field and join type as appropriate.

You can edit the data in a relation table and thereby change the data in the underlying - joined - tables. Column views, sort orders and filters can be applied in exactly the same manner as for ordinary tables (see [Queries](#) and [Query components](#)).

Relating tables using forms

You can relate tables by using a **Database Field** while in **Form** mode to see data from a field in another table. The following example shows you how to go about this.

Example



You have two tables, Orders and Products, as shown below. It is your aim to design a form that will show, for each order, the product name, the quantity, and the cost per item.

Orders

Order_ID	Product_ID	Quantity
ORD1	A003	5
ORD2	A001	7
ORD3	A003	4
ORD4	A002	3
ORD5	A002	2
ORD6	A004	6

Products

Product_ID	Description	Cost
A001	File	2.99
A002	Lamp	25.50
A003	Pen	1.20
A004	Fan	18.99

1. Open the Orders table.
2. Click on the **View Form**  button on the toolbar to open up an auto-form based on the Orders table.
3. Click on the **Design**  button.
4. You now need to insert two lookup fields from Products, namely, Description and Cost. You will need to insert each field separately, using the “Insert Field Wizard”. To see how to do this, go to [Inserting a lookup field](#).

Note: In this example the following selections should be made in “Insert Field Wizard”. Since the field that connects the two tables is Product_ID, you should select it as both the **Data Field** and the **Lookup Field**. Select the Products table as the **Lookup Source**. Since we want to show the product name rather than just the Product_ID, we will choose the Description field from the Products table as the **Display Field**. The same selections are made when inserting the second lookup field, only this time you should select Cost as the **Display Field**.

Once you have inserted the lookup fields you should return to browse mode to scroll through the records. The two lookup fields will display the name and the cost respectively for each order.

You can hide any field you don't want to see in your form (see [Hiding fields from the form](#)).

Form view

A form is an alternative view of a database table. In a form, the data is displayed one record at a time in whatever layout you want. It is often easier to add and edit records in a form than it is using the table grid view.

There are two processes in forms - creating forms and using forms - and so it has two modes of operation: *design mode* and *browse mode*.

Design mode is used to create the page layout and to create and position fields. The operation of a form in design mode is similar to Draw.

Browse mode is used for working with the data – you can view and edit records, create filters to select subsets of records, hide and show fields, rearrange the order of fields, apply sort orders and print records, just as you can in table grid view.

Any changes you make in a form are reflected in the underlying table.

The form can be created either by opening a standard auto form, or else by creating and designing the form yourself.


See:

[Opening an auto form](#)

[Creating a new form](#)

Opening an auto form

The easiest way to get started with a form is to look at an *auto form*.

1. Select the table (or query or relation) you want to use in **Database Manager**.
2. Click on the **View Form**  button on the toolbar, or select the **Form** command from the **View** menu.

A form is created based on the current table, along with any column views, sort orders and filters that have already been applied in Table Grid View.

The auto form is opened in Browse mode (see [Using browse mode in a form](#)).


You can modify an auto form according to your own design while in Design mode.

See:

[Design mode](#)

Design mode

You may decide you want to make modifications to the layout of an auto form. Auto forms make a good starting point for designing your own customized forms.


Click on the **Design** button  on the toolbar or select the **Design** command from the **View** menu.

Forms switches from browse to design mode and the toolbar changes to the Draw toolbar.

Design mode works in a similar way to the Draw module – you can add and edit text boxes and graphical elements (see [Creating a drawing](#) for information on how to use the Draw module).

Ability allows you to reposition fields in the form and to insert various types of object: database fields, formula fields, summary fields, table grids, page numbers, date and time, pictures, and other objects (see [Creating a new form](#) for information on how to use these).

If you have made any changes to the form while in design mode, you will be asked if you want to save the changes when you try to switch to Grid View or close the form. You will then be asked to give the form a name. This form will now be classed as a **Saved Form**, rather than an auto form, and will be attached to the relevant table. Next time you are in the table and you switch to Form View, a "Change Form Type" dialog will be opened, which will give you the opportunity of choosing a new auto form view or the named (saved) form view. You can create as many named form views, attached to the one table, as you like. These can be opened directly from **Database Manager** (see [Opening an existing form](#)).

To go back to working with your data, switch to browse mode by selecting the **Browse** button  on the toolbar or the **Browse** command from the **View** menu.


See [Editing fields in forms and reports](#) for information on editing and moving fields in form design mode.

Opening an existing form

- To open an existing form in browse mode, select the form in **Database Manager** and double-click to open it (alternatively right-click and select **Browse**). You will go straight into the form view without having to go into the corresponding table first.
- To open the form in design mode, select it in **Database Manager**, right-click and select **Design**.

Using browse mode in a form

You use browse mode to actually work with the data from your database table. In browse mode you can page through your records, edit, add and delete records, create filter queries to select subsets of records, hide and show fields, and apply sort orders. These operations are the same as those you use when working in Table Grid View.

You can switch to browse mode, if you are in design mode, by clicking on the **Browse** button  on the toolbar or selecting the **Browse** command from the **View** menu.

Here's a summary of the things you can do in forms browse mode:

- Page up and down through the database table (see [Moving around in a form](#))
- Edit, add and delete records (see [Adding records](#), [Editing records](#), and [Deleting records](#))
- Print records (current form, all forms or a subset of forms).
- Switch to a table grid view or to form design mode.
- Create filters (see [Filtering a table](#))
- Apply sort orders (see [Sorting a table](#))
- Hide fields (see [Hiding fields from the form](#))
- Show fields (see [Showing hidden fields in the form](#))

Moving around a form

To move around a form, you would usually use the mouse and click to put the cursor where you want it. You can use the scroll bars at the right and bottom of the screen to view more of the form.

However, you can also move the cursor using the keyboard, and there are times when this can be quicker. The following table shows all the ways you can move the cursor, whether within a particular form (the same record) or to another form (another record).

Key	Description
tab	Go to the next field.
shift+tab	Go to the previous field.
home	Go to the start of current field.
ctrl+home	Go to the first record (form)
end	Go to the end of current field.
ctrl+end	Go to the last record.
page up	Scroll one screen upwards, but go to previous record when top of page is reached
ctrl+page up	Go to the previous record.
page down	Scroll one screen downwards, but go to next record when end of page is reached
ctrl+page down	Go to the next record.

You can also use the buttons on the Record toolbar to move through the records in your table.

See:

[Browsing through the database table](#)

Browsing through the database table

You can use the four movement buttons on the browse **Record** toolbar to move forwards and backwards through the records in your table (whether in Grid View or Forms):


-  **Next**
-  **Previous**
-  **First**
-  **Last**

You can also use the **Record** menu to access the same commands.

Editing records

In browse mode all the data is “live” – you can edit the contents of fields by moving the cursor to them and typing in new information.

Any changes you make will automatically be saved as soon as you move to another cell.

You can reverse the changes you’ve made to the current field by clicking on the **Undo**  button on the toolbar (or selecting the **Undo** command from the **Edit** menu).

Hiding fields from the form

You don't need to display every field from the database table on your form. To remove a field from Form View, right-click your mouse and select **Columns**. The **Columns** page of the "Format Query" dialog is displayed. Remove the columns you want to hide from **Selected Fields** to **Available Fields**. This works in the same way as for a table (see [Hiding and showing columns using a columns selection](#)).

Note: If you have already hidden some fields while in Table Grid View, these fields will also be hidden when you go to Form View. This will not be the case the other way round though, unless you use a named columns definition in forms to hide columns.

Showing hidden fields in the form

You can show hidden fields by right clicking your mouse and selecting **Columns**. The **Columns** page of the "Format Query" dialog is displayed. Transfer the columns you want to show from **Available Fields** to **Selected Fields** by using the **Add** buttons. This works in the same way as for a table (see [Hiding and showing columns using a columns selection](#)).

Creating a new form

Forms can be created in three ways:

- an auto form directly from the table (see [Opening an auto form](#))
- an auto form which is redesigned in design mode, given a name and saved as a named form (see [Design mode](#))
- a new form which is designed from scratch and is not based on an auto form (see [Creating a form from scratch](#))

Creating a form from scratch

To create a form from scratch rather than basing it on an auto form, follow these steps:

1. Go to **Database Manager** and right-click your mouse.
2. Select **New** and then **Form** from the displayed menu. A "Create Form Wizard" dialog is displayed.
3. On the first page of the dialog choose the layout of the form in the **Types** box:
 - § **Fields arranged down form**
 - § **Fields arranged across form**Click on **Next** to go to the following page.
4. On this page choose whether you want to base the form on a **Table**, **Query** or **Relation**. Selecting, say, **Table** causes all the tables in your database to be displayed in the **Choose a database object** box. Click on the table (or query or relation) you want the form to be attached to, and then click on **Next**.
5. On this page you choose which fields in the table (query or relation) you want to appear in the form. Select these one by one in the **Available Fields** box and click on the **Add** button each time to move them to the **Selected Fields** box (use the **Add All** button if you want to move all of the fields across). You can determine the order of the fields as you are doing this. If you make a mistake or change your mind, use the **Remove** buttons.
6. Click on **Finish**.

A new form is created and displayed in design mode. The fields chosen from the attached table (or query or relation) are displayed in the order and with the form layout type you set.

It is a good idea at this point to save the form and give it a name. Select **Save Form** from the **File** menu. You will then be asked to give the form a name. Click on **OK** once you have done this.

Note that in form design mode the drawing toolbar is displayed. This has the same functionality as the toolbar in Draw. You can draw lines, shapes and frames, and add text. See [Creating a drawing](#).

Ability allows you to insert database fields (that is, fields from the attached table that you may have forgotten to select in **Available Fields**), formula fields, summary fields, table grids, page numbers, date and time, pictures, and other objects.

You can resize form fields and inserted fields or move them to different locations in the form: click on the field and drag the handles to resize, or move the field by dragging and dropping the whole field in a new position when the crossed arrows appear.

See:

[Inserting fields into a form](#)

[Inserting a page number into a form or report](#)

[Inserting the date and time into a form or report](#)

[Inserting a picture into a form or report](#)

[Inserting an object into a form](#)

Inserting fields into a form

You can insert various types of field into a form when in design mode:

- **Database Field**
- **Formula Field**
- **Lookup Summary Field**
- **Table Grid**

These can be accessed individually by selecting them from the **Insert** menu. All four use the standard "Insert Field Wizard" dialog, with each being designed on its own page.

See:

[Inserting database fields into a form](#)

[Inserting formula fields into a form or report](#)

[Inserting a lookup summary field into a form or report](#)

[Inserting a table grid into a form](#)

Inserting database fields into a form

To insert a database field into a form you first need to make sure you are in form design mode. You can insert any field from those already in the attached table or a field from another table.

If you based the form on an auto form, then you merely need to *show* fields that you have previously *hidden*. However, it could be that you want a field to appear more than once in the form. In this case you need to specifically *insert* the field.

If you created a new form and did not base it on an auto form, then it could be that you forgot to include a particular field or that you want some field to appear more than once. Again, in this case, you need to insert the field.

For information on how to insert a database field, see [Inserting a database field](#).

Inserting a database field

To insert a database field into a form or report, follow these steps:

1. Select **Database Field** from the **Insert** menu.
2. The mouse pointer will turn to a cross. Use your mouse to position the cross at the point where you want the database field to appear (you can move this again later, if you want).
3. Click once. The "Insert Field Wizard" dialog is displayed, turned to the database field page.
4. In the **Database Field** group box you choose the field you want to insert. Note that the **Database Name** line gives the name of the current database and its file path, and the **Data Source** line identifies the table (or query or relation) to which the form or report is attached.
5. Click on the arrow in the **Data Field** box to display a drop-down list of all the fields in the attached table. Select the field you want to insert. The name of the field will also appear in the **Field/Label Name** box. This is the name that will appear as the *label* for the field in the form or report. You can edit the label name if you want, although the field itself will retain the name you gave it when designing the table.
6. Click on **Next** to go to the **Control Type** page. Here you can select a control type for the field (see [Control types](#)). If you want to insert the field you chose, in step 5, from the attached table, leave all the control types blank. (However, see [Inserting a lookup field](#) for information on using the combo box control type with lookup fields).
7. Click on **Next** to go to the next page, where you can determine the font, alignment, gap and position for the field label (see [Editing field labels](#)).
8. Click on **Finish** to close the "Insert Field Wizard" and return to form design mode or report view.

The new database field will be inserted.

Repeat the process for every database field you want to insert.

Inserting a lookup field

A lookup field is a type of database field and can be inserted into a form or report. It is useful when you want to insert a field from another table into a form or report based on the current table. A lookup field can also be used to link to a field in another table while displaying the data from a different field.

To insert a lookup field into a form (or report), you should first make sure you are in form **Design** mode (or **Report View** for a report), and then follow these steps:

1. Select **Database Field** from the **Insert** menu.
2. The mouse pointer will turn to a cross. Use your mouse to position the cross at the point where you want the database field to appear (you can move this again later, if you want).
3. Click once. The "Insert Field Wizard" dialog is displayed, turned to the database field page.
4. In the **Database Field** group box you choose the field you want to insert. Note that the **Database Name** line gives the name of the current database and its file path, and the **Data Source** line identifies the table (or query or relation) to which the form or report is attached.
5. Click on the arrow in the **Data Field** box to display a drop-down list of all the fields in the attached table. Select the field you want to insert. The name of the field will also appear in the **Field/Label Name** box. This is the name that will appear as the *label* for the field in the form or report. You can edit the label name if you want, although the field itself will retain the name you gave it when designing the table.
6. Click on **Next** to go to the **Control Type** page. To insert a lookup field select the **Combobox** option (see [Control types](#) for more information on control types).
7. Click on **Next** to go to the next page, where you can determine the font, alignment, gap and position for the field label (see [Editing field labels](#)).
8. Click on **Next** to go to the design page for the **Combobox** control type.
9. In the **Control Source** group box click on the **Lookup Choices** checkbox to switch it on.
10. Click on the **Lookup Source** button and select a table (or query or relation) from the "Lookup Source" dialog.
11. Click on the **Lookup Field** arrow and select a field from the lookup source table.
12. Click on the **Display Field** arrow and select the field whose contents you want to be displayed in the inserted lookup field.
13. Click on **Finish** to close the "Insert Field Wizard" and return to form design mode or report view.

The new lookup field will be inserted. In forms you should now click on the **Browse** button to return to browse mode, where you can scroll through your records.

Inserting formula fields into a form or report

To insert a formula field into a form (or report), you need to be in form design mode (or report view) first, and then follow these steps:

1. Select **Formula Field** from the **Insert** menu.
2. The mouse pointer turns to a cross. Place this where you want the field to appear and click once on your mouse. The "Insert Field Wizard" dialog is displayed, turned to the formula field page.
3. In the **Formula Field** group box you design the formula you want to insert. You can either type a formula directly into the **Formula** box or create a formula by clicking on the **Functions** button to access the standard Ability functions. If you leave the **Formula** box empty the inserted formula field can be used as a normal text field.
4. If you want to show the formula rather than its result, click in the **Show Formula** checkbox to switch it on.
5. Click on **Next** to go to the control type page, where you can choose what sort of control to use with the formula (see [Control types](#)).
6. Click on **Next** to go to the last page, where you can set field specific attributes, such as **Hide Zeros**, **Hide Errors**, and **Edit Source**. (for setting up two-way links).
7. Click on **Finish** to close the "Insert Field Wizard" and return to form design mode (or report view).

The formula field will be inserted at the place you chose.

Repeat the process for every formula field you want to insert.

Inserting a lookup summary field into a form or report

The lookup summary field establishes a join between a selected field in one table (the table that is attached to your form or report) and a matching field in a second table (the joined table). It then uses this join to summarize the information held in a selected numeric field from the joined table. The matching records from the joined table are grouped together according to the field values from the first table.

To insert a summary field into a form (or report) you need to be in form design mode (or report view) first, and then follow these steps:

1. Select **Lookup Summary Field** from the **Insert** menu.
2. The mouse pointer turns to a cross. Place this where you want the field to appear and click your mouse once. The "Insert Field Wizard" dialog is displayed, turned to the lookup summary field page.
3. In the **Data Field** box you select a field from the attached table. Click on the arrow to display a list of all the fields in this table and click on one of these to select it.
4. In the **Lookup Field Source** group box you select the table (or query or relation) and field you want to join to and the field in the joined table for which a summary is to be calculated:
 - Click on the button to the right of the **Join Source** box. A list of all the tables (or queries or relations) in the current database is displayed. Click on the table you want to join to, and then click on **OK**.
 - In the **Join Field** box, click on the arrow to display a list of all the fields in the table you have just selected in **Join Source**. Click on one of these. (Note that it is your aim to match records in the table to which **Data Field** belongs with records in the **Join Source** table, so make sure you select fields that have the same data type and the *possibility* of some matching values. For example, you might select a *Surname* field in both.)
 - In the **Lookup Field** box, you select the field in the joined table that contains the information which is to be summarized. Since most summary information is calculated using numeric values, this should nearly always be a numeric field.
 - In the **Lookup Formula** box you select the type of formula to apply to **Lookup Field**. Choose from the following list:
 - § **Count** (counts the number of matching records)
 - § **Sum** (sums the values in the matching records)
 - § **Minimum** (finds the minimum value in the matching records)
 - § **Maximum** (finds the maximum value in the matching records)
 - § **Average** (finds the average value of the matching records).

Obviously this normally requires that the field you select in **Lookup Field** be of numeric data type. The exception is **Count**, which can be used to count the number of matching records irrespective of the data type of the **Lookup Field**.
5. Finally, click on **Next** and then on **Finish** to insert the lookup summary field at the place you chose.

See:

[Using lookup summary fields in forms or reports: example](#)

Using lookup summary fields in forms or reports: example

This example shows you how to use lookup summary fields in forms and reports (see [Inserting a lookup summary field into a form or report](#)).

You have two tables in a database, one called *Employees* which contains employee names and ID numbers, and another called *Holidays*, which contains employee ID numbers and information on holidays yet to be taken by each employee by the end of the year, excluding the general Christmas holiday.

<u>Employees</u>		
ID	Surname	First_name
1	SMITH	JACKIE
2	ROACH	PHILIPPA
3	ENGLAND	CHRIS
4	COLEMAN	JOEL
5	SMITH	JAMES
6	HOWLETT	PAULINE

<u>Holidays</u>		
ID	Holidays	Days
6	November 2 - 5	4
5	October 2 - 15	10
4	October 3 - 7	5
6	December 1 - 5	5
1	October 21 - 23	3
4	December 2 - 4	3
5	November 18 - 22	5

Your aim is to make a summary of each employee's holiday entitlement. Note that not all the employees have holiday entitlement and that of those who do some have more than one period, that is to say, they are listed more than once in the *Holidays* table. Note also that there are two employees called Smith, hence the importance of using a unique identifier, in this case an ID number, for each employee.

You can insert a lookup summary field into any form or report that is already attached to the *Employees* table. Make sure you are in form design mode or report view first.

To insert a lookup summary field follow these steps:

1. Select **Lookup Summary Field** from the **Insert** menu and click once at the place where you want the field to appear.
2. In "Insert Field Wizard" do the following:
 - In the **Data Field** box, select the *ID* field from the *Employees* table.
 - In the **Join Source** box, select the *Holidays* table.
 - In the **Join Field** box, select the *ID* field from the *Holidays* table as the common field.
 - In the **Lookup Field** box, select the field from which you want to extract summary information.

Obviously in this case it is the *Days* field in the *Holidays* table.

- In the **Lookup Formula** box, select one of **Count**, **Sum**, **Minimum**, **Maximum** and **Average**. In this case select **Sum**. These work as follows:

Count gives the number of periods of holiday for each employee;

Sum gives the total number of days of holiday for each employee;

Minimum and **Maximum** give the number of days of holiday in the shortest and longest periods respectively for each employee;

Average gives the average number of days of holiday for each employee across all the periods.

3. Click on **Next** and then click on **Finish**.
4. It is a good idea to save the form (or report) at this point. Select **Save Form** (or **Save Report**) from the **File** menu and give the form a name.

The summary field is inserted.

If you have inserted the lookup summary field into a form, click on the **Browse** button to go to browse mode. You can now use the **Record** buttons to move through the forms. Each form corresponds to a record in the original *Employees* table.


Since you have selected **Sum** in the **Lookup Formula** box, the result in the summary field in each form will be the total number of days of holiday for the corresponding employee i.e. the corresponding ID number.

For employees without holiday entitlement, that is those with ID numbers that do not appear in the *Holidays* table, a #REF message is displayed in the summary field, indicating that there is no reference from that ID number in *Employees* to a corresponding ID number in *Holidays*.

Inserting a table grid into a form

Ability allows you to insert a table grid into a form. The inserted table establishes a join between a selected field in the attached table (on which the form is based) and a matching field in a second table (the joined table) from the same database. It then uses this join to select matching records from the two tables according to the values in the join field. The join fields should be of the same data type and have the possibility of containing common data.

To insert a table grid into a form you need to be in form design mode first, and then follow these steps:

1. Select **Table Grid** from the **Insert** menu.
2. The mouse pointer turns to a cross. Move this to where you want the field to appear, hold down the left button and drag the cross to create a box outline for the table. Release the button. The "Insert Field Wizard" dialog is displayed, turned to the table grid page.
3. In the **Data Field** box click on the arrow to display a list of all the fields in the form's attached table. Select the field that is to be the join field.
4. Click on the button to the right of the **Join Source** box to display a list of all the tables (or queries or relations) in the current database. Select the table you want to join to, and then click on **OK**.
5. Click on the arrow in the **Join Field** box to display a list of all the fields in the table you selected in **Join Source**. Select the field that you want to join to the field you selected from the attached table in **Data Field**. Click on **Next** to go to the next page.
6. On this page you select the fields (columns) from the joined table that are to be displayed in the inserted table. By default all the columns are listed in the **Hidden columns** box. Select the columns, in turn, that you want to be visible, and then click on **Show** for each one to move it to the **Visible** columns box. Note that in this manner you can also determine the order of the columns in the inserted table.
7. Finally, click on **Finish** to insert the table into the form.
8. Since the form is in design mode, click on the **Browse** button  to switch to browse mode.

In browse mode you can scroll through the records from the attached table using the **Record** buttons. The inserted table will display the information from the fields you chose to show from the joined table, according to whether the join field matches the corresponding field in the attached table. Where there was no match the inserted table will be empty.

See:

[Inserting a table grid into a form:example](#)

Inserting a table grid into a form: example


You have two tables in a database, one called *Employees* which contains employee names and ID numbers, and another called *Holidays*, which contains employee ID numbers and information on holidays yet to be taken by each employee by the end of the year, excluding the general Christmas holiday.

The aim is to display a table in forms for each of the records in the attached table.

<u>Employees</u>		
ID	Surname	First_name
1	SMITH	JACKIE
2	ROACH	PHILIPPA
3	ENGLAND	CHRIS
4	COLEMAN	JOEL
5	SMITH	JAMES
6	HOWLETT	PAULINE

<u>Holidays</u>		
ID	Holidays	Days
6	November 2 - 5	4
5	October 2 - 15	10
4	October 3 - 7	5
6	December 1 - 5	5
1	October 21 - 23	3
4	December 2 - 4	3
5	November 18 - 22	5

Assuming you have already created a form that is attached to the *Employees* table and that you are in form design mode, follow these steps:


1. Select **Table Grid** from the **Insert** menu.
2. Position the cross at the place you want in the form, depress the left mouse button and release.
3. In "Insert Field Wizard" make the following selections:
 - In **Data Field** select the *ID* field from *Employees*
 - In **Join Source** select the *Holidays* table
 - In **Join Field** select the *ID* field from *Holidays*
 - Click on **Next** to go to the next page
4. Move the *ID* and *Days* fields from **Hidden columns** to **Visible columns**, using the **Show** button.
5. Click on **Finish** to insert the field.
6. Click on the **Browse** button  to switch from design mode to browse mode.

Use the **Record** buttons to move through the forms. Since each record in the attached table has a corresponding form view, there will also be a corresponding table in each form. The table will have fields for *ID* and *Days*.

The first form will show a table with *ID* number 1 and *Days* 3; the second form (for *ID* 2, that is Philippa Roach) will show a blank table (there is no *ID* 2 in the *Holidays* table); the sixth form will show a table with two records (*ID* 6, that is Pauline Howlett, has two holiday periods of 5 and 4 days respectively); and so on for the other forms.


Inserting a page number into a form or report

To insert a page number field into a form (or report) you need to be in form design mode (or report view):

1. Select **Page Number** from the **Insert** menu.
2. The mouse pointer turns to a cross. Place this where you want the field to appear and click once on your mouse.
3. The page number field will be inserted at the chosen place.
 - (a) In forms, click on the **Browse** button  to switch to browse mode. As you use the **Record** buttons to scroll through the forms, each form will display its own page number. If you alter the order of the records by applying a sort order, the new order of forms will still be numbered consecutively from 1 upwards.
 - (b) In reports, click on **Print Preview**. Each page will be numbered in order. Note that with a report it makes sense to put the page number in the **Page Header** or **Page Footer**.

Inserting the date and time into a form or report

To insert the current date and time into a form (or report) you should be in form design mode (or report view):


1. Select **Date and Time** from the **Insert** menu.
2. The mouse pointer turns to a cross. Place this where you want the date and time field to appear and click once on your mouse.
3. The field will be inserted at the chosen place.
 - (a) In forms, click on the **Browse** button  to switch to browse mode. Each form will now contain the current date and time.
 - (b) In reports, click on **Print Preview**. Each page will display the date and time. Note that the date and time field is repeated every time the section where it was inserted is repeated.

Note that the date and time field is live and so will alter as the date and time change.

Inserting a picture into a form or report


To insert a picture into a form (or report) you should be in form design mode (or report view):

Select **Picture** from the **Insert** menu. The "Open" dialog is displayed. You should locate the directory of the required picture file as normal and then double-click on the file (or click on **Open**) to close the dialog and insert the picture. Note that before the picture is inserted the mouse pointer, which will have turned to a cross, should be moved to a suitable position in the form before clicking once. You can always move or resize the picture once it has been inserted.

Once you have inserted the picture field, click on **Browse**  to switch to browse mode in forms (or click on **Print Preview** in reports). The inserted picture will be displayed in each of the forms (or in the report).

Inserting an object into a form

To insert an object into a form you should be in form design mode:

1. Select **Object** from the **Insert** menu.
2. The mouse pointer turns to a cross. Move this to where you want the object field to appear, click and drag your mouse to create a box for the object field, and release.
3. The "Insert Object" dialog is displayed, in which you can locate the object that is to be inserted.
4. Once you have inserted the object field, click on **Browse**  to switch to browse mode.

The inserted object will be displayed in each of the forms.

Editing fields in forms and reports

You can change the position and size of any field in your form (or report) and edit the field's properties. To do this you need to be in form design mode (or report view):

Left-click on the field to resize it using the handles and to reposition it by dragging.

To change the properties of a field, left-click on the field to select it and then right-click and select **Properties** from the drop-down menu (alternatively, left-click on the field to select it and edit individual properties by selecting them from the **Format** menu). The "Format Object Style" dialog is displayed. Note that the properties accessible in the dialog depend on the type of field you are editing.

The following table lists the properties that you can edit:

Field property	Description
General	Data fields, data sources, formulas used in the field, etc.
Color	Background and foreground color of the field
Font	Font styles and sizes
Style	Pen and brush styles
Border	Border types
Align	Align the field's contents left, right or center
Control	Edit control type
Label	Edit placement and alignment of label
Attributes	Choose various field attributes e.g. Hide Zeros, Edit Source
Format	Display format for numbers, dates, and times

Multi-page forms

Ability allows you to use more than one page when designing a form. You may find that you have too many fields to fit onto a single page, and so need to use another.

Before selecting the type of field to insert or the shape that you want to use to create a drawing, go to the required page by using the right hand and bottom scroll bars (to scroll below and to the right of the first page respectively).

The new page will be visible on screen for you to insert fields and design drawings. Note that extra pages are available both below and to the right of the first page.

Reports

Reports provide a powerful method of analyzing database information and allow you to present your data in a more attractive and effective way. The basic data in a report comes from an underlying table (or query or relation), although you can summarize the data and also insert various types of field. The report can be designed using the full resources of the Draw module. You can preview a report on screen and then print it for maximum impact.


See:

[Opening an auto report](#)

[Creating new reports](#)

Opening an auto report

Auto reports are an easy and efficient way of creating reports. When browsing any table, relation or query, you can always create an auto report with a single click, as in step 4 below. To get the most out of this feature though, it is best to select the columns, sort order and filter conditions for the report first, as follows:

1. In table grid view, select the columns you want to see in the report – if you start with all columns, use the right mouse button to hide the columns you don't want (for more details on column selections, see [Hiding and showing columns in a database table](#)).
2. Apply any filters you want. Use the right mouse button to perform a quick filter selection; alternatively use the filter dialog for a more complex filter– see [Filters](#) for more details.
3. Still in table grid view, select a sort order for the report. Sorting has special significance for the auto report: data will be sorted *and* grouped together, with group totals appearing automatically. You can sort on multiple levels – see [Sort orders](#) for more details.
4. Finally, select the **Report** button  or select **Report** from the **View** menu.

The report is initially divided into several sections separated by bars. Each bar heads its own section, in which you can display data or enter text and graphics. Here is a list of the bars and what they stand for:

- **Report Header** - a header for the whole report on the first page
- **Page Header** - a page header for every page (by default, the table field *names*)
- **Detail** - the main body of the report, which displays the table fields
- **Page Footer** - a page footer for every page
- **Report Footer** - a footer for the whole report on the last page, which automatically displays total summary information for some of the fields


Note that the **Draw** toolbar is also displayed.

See:

[Detail of an auto report](#)

Detail of an auto report

- An auto report places the *fields* from the underlying table in the **Detail** section.
- In the **Page Header** section the *names* of these fields are displayed underlined in text boxes. You can see that the page headers will be used as *titles* for the columns in the report.
- The **Report Header** and **Page Footer** sections are left empty for you to insert fields, if you wish.
- The **Report Footer** section automatically displays total summaries for several of the fields displayed in the **Detail** section: the first Character field (Count) and all the Numeric fields (Sum).

The basic detail filled in by the auto report is sufficient to produce a good example of how a report works. Click on the **Print Preview** button  on the toolbar to see how the auto report will look when printed, with all the relevant data displayed (see [Viewing a report in print preview](#) for information on previewing your reports). Return to **Report View** by clicking on **Close**.

You can edit an auto report to change its layout, add or delete fields, insert summaries, format the sections, and so on (see [Editing reports](#)).

Any graphics, fields or text boxes you insert into a section will be repeated whenever the section is repeated in **Print Preview**. For instance, since a page header is repeated at the top of every page, any objects inserted into the page header section will also be repeated at the top of every page. Also, since the fields in the **Detail** section represent *one* record in the underlying table, any objects inserted will be repeated for every record displayed in **Print Preview**.

If you make changes to an auto report, when you try to close it you will be asked if you want to save the report. If you say **Yes**, you will then be asked to give the report a name. In future when you switch to **Report View** from an open table, you will be given the choice of opening another auto report or an already existing named report based on this table.

You can open a named report directly from **Database Manager** (see [Opening an existing report](#)).

Opening an existing report

To open an existing report, select the report name in **Database Manager** and then do one of the following:

- Double-click to open the report in **Print Preview**
- Right-click and select either **Preview** to go straight into **Print Preview**, or **Design** to go into **Report View**.

You will go straight into print preview or report view, without having to open the corresponding table first.

Creating new reports

To create a new report from scratch, follow these steps:

1. You first need to go into **Database Manager**.
2. Right-click your mouse and select **New**, followed by **Report**. The "Create Report Wizard" dialog is opened.
3. On the first page of the dialog you choose the type of the report, from these options:
 - **Fields arranged across report**
 - **Fields arranged down report**
 - **Labels**

(see [Types of report](#) for more information)

The following steps assume you have selected either **Fields arranged across report** or **Fields arranged down report** (since **Labels** works slightly differently, see [Labels in reports](#)).

4. Click on **Next** to go to the next page of the dialog.
5. On this page you choose a database object on which to base the report. Select one of the following:
 - **Tables**
 - **Queries**
 - **Relations**

Selecting **Tables**, say, will cause all the tables in the current database to be displayed in the box. Click on one of the tables (or queries or relations) to select it as the database object, and then click on **Next** to go to the next page.

6. On this page you choose the fields whose data you want to be displayed in the main body of the report, that is in the **Detail** section. Click on these one by one and use the **Add** button to transfer them from the **Available Fields** box to the **Selected Fields** box. Remember that you can determine the order of the fields at this point. If you make a mistake just use the **Remove** buttons and start again. Go to the next page.
7. On this page you choose the way the report is to be grouped. Since you can have groups determined by the values of one field, and groups within these groups determined by the values of another field, it is important to order the fields correctly. Therefore make sure that the field which determines the primary group comes first, the field which determines the secondary group comes second, and so on. To select the fields for the groups, click on these one by one in the **Available Fields** box and use the **Add** button to transfer them to the **Selected Fields** box. If you make a mistake use the **Remove** buttons and start again. You can select up to three group fields. See [Sorting and grouping in a report](#) for more information.

Note: Groups automatically have a sort order applied to them. By default this is ascending for each group field (hence the "ASC" appended to each field in **Available Fields**), although you can change this when you edit a report.

8. On the last page, you choose the fields for which you want to calculate summary values:
 - In the **Field** column all the fields to which summaries can be applied are listed.
 - Next, select the type of summary you want to be applied. The options are:

Count (counts the number of records)

Sum (calculates the total for the values in a numeric field)

Avg (calculates the average of the values in a numeric field)

Min (finds the smallest of the values in a numeric field)

Max (finds the largest of the values in a numeric field).

Note that the type of summary available depends on the data type of the field. So for a character field, for instance, the only summary option is to count the number of records, and for a numeric field all the summary options are applicable. By default, the first *character* field has its **Count** option switched on, and all *numeric* fields have their **Sum** option switched on. You can change these and also select extra summary types for a particular field.

9. Once you have chosen the layout, associated table, report fields, group fields and summary options for your report, click on **Finish** to close the dialog.

The new report, based on the database object you chose, is displayed. The selected fields are displayed in the **Detail** section and the selected group fields are displayed in the **Group Header** sections. The **Group Footers** and the **Report Footer** all display the summary fields and types of summary you chose.

Whether labels and/or page headers are also displayed depends on the type of the report. See the following sections for more details:

[Type of report: vertical fields](#)

[Type of report: horizontal fields](#)

[Labels in reports](#)

At any time you can switch to **Print Preview**  to check what the report will look like.

The new report can be saved and named. Select the **Save Report** command from the **File** menu and enter a name for the report. Alternatively, if you close the report you will be asked whether you want to save it, and if so what name you want it to have.

The report will appear under **Reports** in **Database Manager** and can be opened directly from here in future.

See [Editing reports](#) for information on how to edit your report.

Types of report

When you create a new report from scratch, you can choose from one of three basic report types. These are:

- § **Fields arranged across report** (see [Type of report: horizontal fields](#))
- § **Fields arranged down report** (see [Type of report: vertical fields](#))
- § **Labels** (see [Labels in reports](#))

Once the report is created, based on one of the three types, it is likely that you will want to edit it to suit your own purposes. For instance, you can rearrange fields, display some fields and hide others, choose whether to display labels, resize sections, etc. and, of course, insert graphics, other fields and text.

See:

[Creating new reports](#)

[Editing reports](#)

Type of report: vertical fields

- In the **Detail** section of this report type the selected fields are arranged vertically, one above another, with their labels displayed to the right, one for each field.
- Each group field (if there are any) is placed in its own **Group Header** section, with its label placed directly above it. The primary group field is placed at the left of its header; the secondary group field is placed at the next position along in its header; and so on.
- The effect of this in **Print Preview** is to display the values from the fields in a column, each value being identified by its label. If the report is grouped, the groups are clearly sectioned off from one another and each section is headed by a different value from that particular group field, along with the group label, which remains the same throughout the sections defined by a particular group field.

See:

[Types of report](#)

Type of report: horizontal fields

This report type has the same basic format as an auto report.

- In the **Detail** section of this report type the selected fields are arranged horizontally, but no labels are displayed. Instead, the *names* of the fields are displayed as *titles* in text boxes in the **Page Header** section.
- Each group field (if there are any) is placed in its own **Group Header** section, with its label placed directly above it. The primary group field is placed at the left of its header; the secondary group field is placed at the next position along in its header; and so on.
- The effect of this in **Print Preview** is to display the values from the fields in a row, each value being identified by the title in **Page Header** under which it falls. If the report is grouped, the groups are clearly sectioned off from one another and each section displays a different value from that particular group field, with the group label remaining the same throughout the sections.

See:

[Types of report](#)

Labels in reports


You can create mailing and other types of label in a labels report. The report is based on a table (or query or relation). Obviously for mailing labels the table should contain names and addresses. To create a label report from scratch, go to **Database Manager** and do the following:

1. Right-click your mouse and select **New** followed by **Report**.
2. In the "Create Report Wizard" select **Labels**.
3. On the next page:
 - select the type of label from the **Available labels** box (the default is Custom)
 - select the number of columns of labels per page in the **Columns** box (the default is 2)
 - fine tune the label width and height in the **Label** box
 - change the horizontal and vertical gap between labels in the **Gap** box
4. On the next page select the table, query or relation on which you want to base the label report.
5. Next, choose the fields that are to be displayed in the first line of the label report (e.g. these might be title, first name and surname fields). Use the **Add** button to transfer these from the **Available Fields** box to the **Selected Fields** box.
6. On the next page you choose the fields that are to appear on the subsequent lines, making sure that you transfer them in the order you want (just use the **Remove** button if you make a mistake). Label reports display one record per label, using only the fields you have previously selected.
7. Click on **Finish** to close the dialog and go into **Report View**.

The selected fields are displayed in a column in the **Label** section. Note that in the case of a label report you cannot rearrange the fields at the Report View stage (if you have made a mistake in the order of the fields which you haven't corrected at the Wizard stage, then it is best to start again).

You can change the font style, the fore- and background colors, the border style, and so on, of any field. In addition you can exchange one field for another. To do any of these select the field you wish to format by clicking on it. Right-click and select **Properties** from the drop-down menu. In the "Format Object Style" dialog select from the available tabs: **General**, **Color**, **Font**, **Style**, **Border**, **Align**, **Label**.

You can also use the **Draw** toolbar on the left to insert shapes and text boxes.

To see what the labels will look like, click on the **Print Preview** button  on the toolbar.

At this point you can click on the **Print** button to print the labels, otherwise click on **Close** to return to **Report View**.

If the printing of the label contents appears to near to the edge of the physical label, select all the fields and drag them left and/or down.

To save and name the report, select the **Save Report** command from the **File** menu.

See:

[Types of report](#)

Inserting fields in a report

You can insert various types of field into a report. These are accessible through the **Insert** menu:

- **Database Field** (see [Inserting a database field](#))
- **Formula Field** (see [Inserting formula fields into a form or report](#))
- **Lookup Summary Field** (see [Inserting a lookup summary field into a form or report](#))
- **Group Summary Field** (see [Group summary fields](#))

If you want an inserted field to appear just once in every report, you need to insert it into the **Report Header** or **Footer**.

If you want it to appear on every page, insert it into the **Page Header** or **Footer**.

Insert it into the **Detail** section if you want it to appear with every record.

Editing a report

You can edit a report to:

- § Add and remove fields (see [Inserting fields in a report](#))
- § Format sections (see [Format sections in reports](#))
- § Relocate and edit fields (see [Editing fields in forms and reports](#))
- § Edit field labels (see [Editing field labels](#))
- § Add graphics and text (see [Creating a drawing](#))
- § Resize sections (see [Resizing the sections in a report](#))

Resizing the sections in a report

When you first enter a report you will see that the sections of the report are placed in close proximity to one another. To resize the sections so that there is enough space to enter text, create graphics and insert fields, you should do the following:

1. Resize a section by moving your mouse pointer to the border of the bar *above* the following section. The mouse pointer turns to a double vertical arrow.
2. Click and drag the border to the required place, and then release the mouse. The section is resized.

For instance, to resize the **Report Header** section, move the mouse pointer to the top of the **Page Header** bar. When the vertical arrows appear, click and drag and then release to resize.

You can perform this operation on any of the sections, whenever and as many times as you like, till you get a suitable size for all the sections. Appropriate section sizes are important, not just for creating space to design a report but also for display purposes. For instance, if you reduce a section so that certain fields are not visible in Report View, neither will these fields be visible in **Print Preview**, and hence they will not be shown when the report is printed.

You can also resize the height of sections in the “Sections Properties” dialog (see [Format sections in reports](#)).

Headers and footers in a report

There are three types of headers and footers in a report:

- those for the report as a whole
- those for the pages
- those for the groups (see [Sorting and grouping in a report](#) for more information on group headers and footers).

The **Report Header** will appear once at the top of the report, that is, on the first page, and the **Report Footer** will appear at the bottom of the report, that is, on the last page.

The **Page Header** and **Page Footer** will appear at the top and bottom respectively of the individual pages in the report. When you open an auto report directly from its underlying table, the **Page Header** section contains by default the field names, underlined, from the table. These will be displayed at the top of each page, although you can edit them (or delete them if you don't want them to appear at all).

To enter text or graphics into a header or footer, use the buttons on the **Draw** toolbar (see [Creating a drawing](#)). You can also insert page numbers (see [Inserting a page number into a form or report](#)) and the date and time (see [Inserting the date and time into a form or report](#)) into a header or footer.

Note that if you have created a report header and a page header, the report header always occurs *before* the page header on the first page of the report. Similarly, a report footer always occurs *after* the page footer on the last page of the report.

Format sections in reports

You can format the sections in a report in various ways. In **Report View** select the **Sections** command from the **Format** menu. The “Section Properties” dialog is displayed. All the sections in your report are listed in the **Section Name** column, including any group headers and footers you may have created. For each section you can format the following properties:

1. **Background Color**: click in the field to display an arrow and then click on the arrow and select a color from the drop-down list. The default is white.
2. **Height**: type in a new height for the section.
3. **Breaks**: click in the field to display an arrow and select one of the following from the drop-down list:
 - § **Continuous**: sections run on from one another (this is the default)
 - § **New Page**: a new page is started every time the section is printed
4. **Visibility**: click in the field to display an arrow and select one of the following:
 - § **Yes**: the section will be displayed in **Report View** and **Print Preview** (this is the default)
 - § **No**: the section will not be displayed in **Report View** or **Print Preview**
5. Click on **OK** once you have formatted the new section properties.

Report View is redrawn to reflect the new formats.

Editing field labels

You can determine the font, alignment, gap and position for database field labels in forms and reports (and group summary fields in reports). This can be done when you insert a new field (see [Inserting a database field](#) and [Group summary fields](#)). Alternatively, for an already existing field with a label, first select the field by clicking on it once when in Form Design Mode or Report View, then right-click and select **Properties**, followed by **Label** in the “Format Object Style” dialog.

On the **Label** page, follow these steps to edit a label:


1. Choose whether you want the label to be displayed (or not) by clicking the **Show Label** box on (or off).
2. In the **Placement** box you choose whether the label is to be displayed **Over**, **Under**, **Before** or **After** the field.
3. In the **Horizontal Align** box (for **Over** or **Under**) or **Vertical Align** box (for **Before** or **After**), choose whether to align the label **Left**, **Center** or **Right**, or **Top**, **Baseline** or **Bottom** respectively.
4. In the **Gap** box you can edit the default value to set the gap you want between the label and the field.
5. Click on the **Font** button to set a font, style, size and color for the label.

Sorting and grouping in a report

Ability allows you to group records that have common values in a particular field. This is accomplished by sorting on the field. In Reports the sorting command automatically sorts the records in ascending order (according to the values in the chosen field) and then sections off the records with common values.

Within the groups created by this primary grouping, you can create a secondary grouping for records with common values in another of the fields, and so on. The secondary grouping is created by the second sort. In this way Ability provides an effective method of structuring the data in a report. In addition, you can manipulate a report so that the *values* of group fields act as *titles* for each group.

You can choose which fields to sort and group on in both Table Grid View and Report View. This is accomplished through the Sort Order page of the “Format Query” dialog, although grouping itself will only be applied when you are in a Report. To create groups in an auto report or in an already existing report do the following:

1. Click on the **Sort Order**  button on the **Query** toolbar. The “Format Query” dialog is opened at the Sort Order page.
2. The **Available Fields** box shows all the fields in the table (or query or relation) upon which the report is based. To group records by a particular field, click on the field name to select it and then click on the **Add** button. The field name is copied to the **Selected Fields** box. Ability will look at every value under this field in the table and arrange the records into groups, each group of records having a common value in the specified field.
3. You can choose ascending (the default) or descending order for the records. Click on the field in **Selected Fields** to activate the **Descending** checkbox. For a descending order, make sure the **Descending** checkbox is switched on, and for an ascending order, leave it blank.
4. It may be that you want to group the records *within* the primary groups. To do this, click on the field in the **Available Fields** box that you want to group by, and again click on **Add** to copy it to the **Selected Fields** box. It is placed *after* the primary group field. This tells Ability to sort and group by the first field, and then, within the groups so created, sort and group by the second field.
5. Again you can set an order for the secondary group field. Select the field in the **Selected Fields** box and switch the **Descending** checkbox on or off. Note that the secondary grouping can have a different sort order from the primary grouping. You can have as many groups as there are fields in the original table, although of course you would only want to group where it is appropriate.
6. To sort *and* group you need to switch on the **Group on report** checkbox for each of the selected fields that you want to group. If you leave the checkbox blank, the sort order (or orders) will be applied but the sorted records will not be grouped.
7. Once you have defined your groups and their sort order, click on **OK** to close the dialog and return to the report.

The report is redrawn to show a new set of headers and footers for each group. Anything you insert into a group header or footer will be displayed at the head or foot of *each* group in the report (the full effect of this is only apparent when you go in to **Print Preview** - see [Viewing a report in print preview](#)).

Ability automatically gives your report a default layout – see [Layout of a grouped report](#).

Group summary fields

Group summary fields are used to provide summary information of various types for the groups in a report. You can summarize any or all of the groups individually by inserting a Group Summary Field into the relevant **Group Footers** or **Headers**. In addition, you can obtain a total summary for all of the groups combined, by inserting a Group Summary Field into the **Report Footer** or **Header**. To insert a Group Summary Field into a report, follow these steps:

1. Make sure you are in **Report View**.
2. Click in either a Group Header or Footer section (for individual group summaries), or in the Report Footer or Header section (for total report summaries).
3. Select **Group Summary Field** from the **Insert** menu.
4. The mouse pointer will turn to a cross when you move it over the report. Use your mouse to position the cross at the point in the section where you want the Group Summary Field to appear (you can move this again later, if you want) and click once.
5. The “Insert Field Wizard” dialog is displayed, turned to the Group Summary page:
 - (a) In the **Database Field** box, the current database is displayed in **Database Name** and the table upon which the report is based is displayed in **Data Source**. These are fixed and can't be changed by you. Click on the arrow in **Data Field** and choose, from the drop-down list of all the fields in your table, the field for which you want a summary. Note that a label is created for this field in the **Field/Label Name** box, which you can edit if you like.
 - (b) In the **Formula** box, click on the arrow to display the list of available summary options for the chosen field. These are:
 - **Count** (counts the number of records)
 - **Sum** (adds the values if the field is numeric)
 - **Minimum** (finds the minimum value if the field is numeric)
 - **Maximum** (finds the maximum value if the field is numeric)
 - **Average** (finds the average value if the field is numeric)
6. Click on the **Next** button to go to the next page. Here you can set various formats for the label that accompanies the inserted field (see [Editing field labels](#)).
7. Click on **Finish** to close the dialog and return to **Report View**.

The Group Summary Field is inserted into the section.

If you want to insert another field just repeat the above steps.

To see what the report will look like, click on the **Print Preview** button on the toolbar.

Note that a Group Summary Field inserted into a group header or footer provides summary information for *each* of the groups, and one inserted into a report header or footer displays total summary information for *all* the groups combined.

Layout of a grouped report

- Once you have chosen the groups for your report and returned to **Report View** you will see that extra headers and footers have been added for the groups. These group headers and footers are named after the fields they have been based on. The group headers are placed between the page header and the details section, and the group footers are placed between the page footer and the details section. The first or primary group header is placed above the secondary group header, and so on for any other group headers. The primary group footer is placed below the secondary group footer, and so on for any other group footers.
- The group headers cause the report to display the value for each of the groups at the head of each group. The same is true for any secondary groups that are contained (nested) within the primary groups. Each of the values from a group field can be said to act as a title for its own group section. You may wish to display the labels that accompany the group fields. Switch to **Print Preview** to get a clearer idea of the effect of grouping.
- In the **Detail** section the report automatically displays all of the fields from the underlying table. Since for a grouped report the values from one or more of these fields will appear at the head of each group, you may want to delete some of the fields in **Detail**.
- In the group footers and the report footer, there are displayed several summary fields: the first Character field from the underlying table is *counted* and all the Numeric fields are *summed*. Thus each grouped section has its own summary information. This applies to the secondary and lower level groups, as well as the primary ones. You may want to delete some or all of these, or add others. Similarly, the **Report Footer** contains summary information for the whole of the report, and can also be edited.
- You can easily relocate fields within the sections by clicking and dragging.
- The default report layout is designed to contain enough information to make subsequent editing easier for you. For instance, it is much easier to delete the fields you don't want and to relocate the remaining fields, than to have to insert fields that are not displayed.

See [Editing reports](#).

Using queries in reports

When you create a report, it will retain any query formats that were already set in the underlying table. This includes filters, column views, and sort orders. If you then save the report it will retain the query formats. These can be changed in **Report View** by using the options on the **Query** toolbar. You can do this in the same way as for Table Grid View (see [Query components](#)).

Viewing a report in print preview

Whenever you enter a **Report View** you are enabled to view the basic design of the report. Here you can insert objects, create graphics, resize the sections and generally edit the report's design. What you do not see is the report as it will appear when printed, that is, with all its data, summaries, groups, etc.

§ To view the report as a *report* you need to click on the **Print Preview** button  on the **Standard** toolbar (or select **Print Preview** from the **File** menu).


This takes you to the print preview page, where the report as it will appear when printed is displayed. At this stage you can check the report for its overall layout, its group order, the contents of headers and footers, the position of labels, and other features.

§ Use the **Zoom** and **Page** buttons and the scroll bars to see more of the report.

To make changes click on **Close** to close the preview and return to **Report View**.

Printing a report

To print a report, follow these steps:

1. Open the report as a Report View
2. Switch to **Print Preview** (use the button  on the toolbar or the command in the **File** menu)
3. Click on the **Print** button in **Print Preview** to print the report.

Note that to set up a page prior to printing a report, you need to be in **Report View**, not **Print Preview**. Select **Page Setup** from the **File** menu and choose the page margins, paper orientation, paper size, and printer. You can then print from **Print Preview**.

Deleting a database

To delete a database, you should be in the Ability Database window and make sure that the database you want to delete is closed:

1. Select **Open Database** from the **File** menu. The "Open existing database" window is displayed.
2. Locate the database to be deleted and right click on it.
3. Select **Delete** from the displayed menu. You will be asked if you are sure that you want to delete this database. Click on **Yes** and the database will be deleted.

If the database is open, an error message will be displayed and you will not be allowed to go ahead with the deletion (see [Closing a database](#)).

Deleting a database object

To delete a database object (tables, relations, forms, reports, queries, sort orders, filters, column views) follow these steps:

1. Go to **Database Manager**.
2. Right click on the object you wish to delete and select **Delete** from the menu.
3. You will be asked if you are sure that you want to proceed with the deletion. Click on **Yes** and the object will be deleted.

Note that when you delete a query, any query components you created when designing the query are retained under the relevant table name.

Repairing a damaged database

On rare occasions you may find that you are unable to open a database, or, if you can open it, that its behavior is unpredictable. It is more than likely that the database has somehow become damaged. This can be caused, for instance, by a power failure when you are trying to write to the database, or by a general hardware fault. To repair a damaged database, follow these steps:

1. Make sure you are in the Ability Database Window but that the database you want to repair is not open (select **Close Database** from the **File** menu to close the current database).
2. Select **Repair Database** from the **File** menu.
3. In the “Select an existing database” dialog use the **Look in** box to locate the database you want to repair. Once the database name has been placed in the **File** name box, click on the **Open** button.

After a few seconds a message will be displayed telling you that the database was successfully repaired. If the database you tried to repair was open, an error message will be displayed telling you to close the database before trying to repair it (see [Closing a database](#)).

Importing tables

You can import tables into an Ability Database. The imported tables can come from an earlier version of Ability, another database in Ability, or another program altogether. In addition, you can append an imported table to an existing table in Ability (see [Appending imported tables](#)).

To import a table you should first open the Ability database into which the table is to be imported (see [Opening a database](#)). Once you are in **Database Manager** do the following:

1. Select **Import** from the **File** menu. The “Import” dialog is opened.
2. In the **Data Source** box, select the type of database from which you want to import and then click on **OK**. The “Open” dialog is displayed with the selected database type in the **Files of Type** box.
3. Locate the database you want to import from by using the **Look in** box. Once it has been placed in the **File name** box, click on **Open**.
4. The “Import” dialog is opened. All the tables in the source database are listed in the **Objects** box. Click on the table you want to import, to select it, and then click on **OK**.

The table will be imported to the current Ability database and will be listed under **Tables** in **Database Manager**. You can now go ahead and work with the imported table as you would with any table created in Ability.

Exporting tables

You can export tables from an Ability Database. The exported tables can be exported to another database in Ability or another program altogether.

To export a table you should first open the Ability database from which the table is to be exported (see [Opening a database](#)). Once you are in **Database Manager**, do the following:

1. Select **Export** from the **File** menu. The “Export” dialog is opened.
2. In the **Data Destination** box select the type of database to which you want to export and then click on **OK**.
3. In the **Objects** box all the tables in the current Ability database are listed. Select the table you want to export and click on **OK**.
4. The “Save As” dialog is opened. The destination database type is displayed in the **Save as type** box. Locate the destination database in the **Save in** box and select it to place it in the **File name** box.
5. Click on **Save** to export the table to the selected destination database.

Open the destination database and the exported table will be shown in the Database Manager (or equivalent, depending on the program). You can now go ahead and work with the exported table as you would with any table created in the destination database.

Appending imported tables

You can append an imported database table to a table in Ability. This is useful if you want to append, say, an address table to an address table. The two tables should therefore have similar contents and the fields that are to be matched should be of the same data type (for instance, it wouldn't make sense to append a character field to a numeric field).

To append the imported table (the Source table) you should first open the table in Ability that you want to append to (the Destination table), and then follow these steps:

1. Select **Import** from the **File** menu. The "Import" dialog is opened.
2. In the **Data Source** box, select the database type from which you want to import. Note that the **Append to current table** box is switched on. Click on **OK**.
3. In the "Open" dialog, use the **Look in** box to locate the source database and place it in the **File name** box. Click on **Open** to open the "Import" dialog.
4. In the **Objects** box, select the table you want to import and then click on **OK**.
5. The "Field Correspondence" dialog is opened. The **Source table fields** and **Destination table fields** boxes list all the fields in the two tables. The **Current correspondences among the fields** box displays, on the left, all the fields in the source table and, on the right, any matches Ability has been able to make between the tables. You can add and delete matches as follows:
 - § To remove a match, select that line and click on **Delete**.
 - § To make a match select the fields to be matched in the **Source table fields** and **Destination table fields** boxes respectively, and then click on **Add**. The matched fields are linked in the **Current correspondences among the fields** box. If you attempt to match fields with incompatible data types, an error message will be displayed and the match will not go ahead.
 - § Once you have created the appropriate matches, click on **OK** to close the dialog and return to the destination table.

Records will be appended to your table from the imported source table, with values shown in the appropriate match fields.

