

## Table of Contents

Introduction.....	2
1. Navigation Toolbar & Navigation Bar form control .....	3
Finding records.....	7
Use of Navigation Toolbar with multiple Data Forms.....	8
2. Permanent filters.....	9
Form sub-form relationship.....	9
Indexing example.....	12
Multiple filtering example.....	13
3. Power filtering.....	16
4. The OOO 3.2.? bug.....	20

## Introduction

This document should be read in conjunction with the demonstration database - OpenOffice.org Base file "**FilterExamples.odt**" - that is included in the downloadable zip file. Both files have primarily been prepared for my own use as I put together a series of notes and experiments. Thus, they are offered *as is* with no claims to accuracy or completeness.

Many queries arise in the forums as newcomers tackle the fascinating complexities of Base. Hopefully this document may serve as an introduction to one aspect of Base that frequently challenges newcomers - **filtering and searching** for records in a database.

## Definitions

To avoid a confusion that often results when the word "form" is applied to three quite distinct objects in Base, I try to use more specific descriptions, unless the context makes the meaning obvious:

**form document:** the Writer document that contains one or more data forms;  
**data form:** the distinct logical object that can be a main or a sub data form;  
**form control:** the various controls contained within a data form such as a label, text box, list box, combo box, button or even a grid/table.

## Summary

The first section of the document describes a powerful feature of the Base GUI that is all too often overlooked: navigation bars that can be added to a form document or to any data form. These are not well documented in the Base Help file, but are quite user-friendly and particularly powerful when used in combination with each other.

The second section covers permanent filtering where the selection of filter criteria is easy and its application immediate. It has the limitation that it is not possible to remove totally the filter(s), but in many cases that is not a serious drawback. This section explores in some depth the **relationship between a data form and its sub-form(s)**.

The third section explores the power of SQL to open up comprehensive filtering options that can be both efficient and user-friendly. It is offered as little more than an introduction to a method that is applicable to most filtering requirements.

The final section describes a bug found currently in both of the 3.2 versions of Base. Hopefully this bug, which did not exist in the 3.1.1 version, will be corrected in 3.3. It appears to arise when a new or changed record in a data sub-form is saved, but only when that sub-form has been filtered by more than one criterion.

The document is long enough without discussing the subject of macros. OpenOffice Basic, or rather its API, has a steep learning curve. Its mastery is certainly rewarding and relevant to filtering. It is, however, a subject that would be better covered elsewhere.

# 1. Navigation Toolbar & Navigation Bar form control

Advantages:

- Simple to set up
- Readily available to OOO Base form documents
- Comprehensive filter options
- Fast and suitable for large number of records

Disadvantages:

- Not always user friendly

It is a potentially confusing feature of OOO Base that a form document has access to two distinct “navigation toolbars”. One is the *Form Navigation Toolbar* that can be added to a form document by selecting it from the Toolbars sub-menu of the main View menu. The other is the *Navigation Bar* form control which can be added to a data form in the same way as any other form control .

**The Form Navigation Toolbar selected from the View menu:**

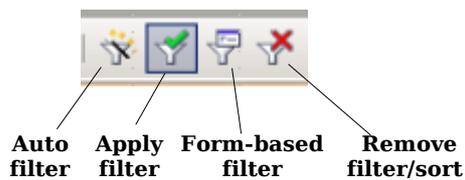


**The form control - Navigation Bar:**

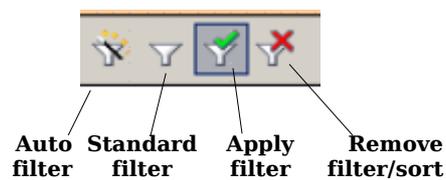


The two navigation bars look largely similar but there is a significant difference in the four icons used for filtering:

**The Form Navigation Toolbar from View menu**



**The form control - Navigation Bar**



Three icons - *Auto filter*, *Apply filter* and *Remove filter/sort* - are the same in each navigation bar, and can be used interchangeably from either. It is not material that the *Apply filter* is located in a different position on each of the bars.

The *Form-based filter*, however, is available only from the Toolbar whereas the *Standard filter* appears only on the Form Control. As it happens these two do perform much the same task, but in different ways: the *Form-based filter* is more powerful but the *Standard filter* is more user friendly.

The following sections describe in greater detail the operation of these icons and the filtering options they provide.

## The Auto Filter



*Auto filter* - The Auto filter icon is available on both the Toolbar and the Form Control and its operation is similar in each case. It is user friendly but has limited scope.

Select with the mouse the content of any particular field in a displayed record, and then click the *Auto filter* icon. It immediately applies a filter to the data form so that only those records with identical content in that field will now be available for display. This operation can be repeated by selecting an entry in another field in the filtered data form and again clicking the *Auto filter* icon. The effect will be to AND the two filters so that the available records will now only be those that match both criteria. This multiple selection and filtering can be repeated, but I have only tested it to a limited number of ANDed criteria because all too soon the stage is reached where, unless there are duplicates in the data source, only a single record complies with all the criteria.

The Auto filter has limitations. It only provides '=' or directly matching comparisons and only permits a Boolean AND operator between criteria. There is no provision for other comparisons such as '<' less than or the Boolean OR. Its strengths are simplicity, speed and a wholly visible operation.

**Note:** a filter created by using the Auto Filter can always be opened with the Standard or Form-based Filter icons (see below) and edited therein to provide greater flexibility.

## The Apply Filter



*Apply filter* - The Apply filter icon is similarly available on both the Toolbar and the Form Control: its operation is the same whichever is used.

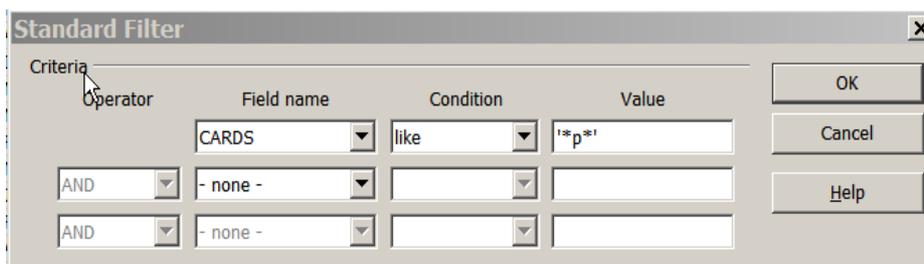
It provides a toggle for applying or temporarily removing whatever pre-defined filter has already been set. Thus, it permits temporary restoration of the unfiltered state, but also re-application of the filter without any need to redefine the filter criteria. It is automatically applied when a filter is initially created by the Auto filter icon.

## The Standard Filter



*Standard Filter* - The Standard filter icon is only available from a form control Navigation Bar that has been added to the data form.

It can be used to define reasonably complex filters. When the icon is clicked it shows the dialog:



The image shows a dialog box titled "Standard Filter" with a close button (X) in the top right corner. The dialog contains a table for defining filter criteria. The table has four columns: "Criteria", "Operator", "Field name", "Condition", and "Value". The "Criteria" column is currently empty. The "Operator" column has a dropdown menu set to "AND". The "Field name" column has a dropdown menu set to "CARDS". The "Condition" column has a dropdown menu set to "like". The "Value" column has a text input field containing "\*p\*". To the right of the table are three buttons: "OK", "Cancel", and "Help".

Criteria	Operator	Field name	Condition	Value
	AND	CARDS	like	*p*
	AND	none -		
	AND	none -		

Thus, it permits selection of more complex filtering options than those provided by the Auto filter. A drop down list assists accurate selections from the *Field name* column. The drop down list in the *Condition* column offers a variety of comparison operators including LIKE for wild card partial matches

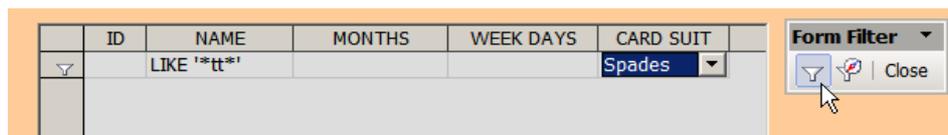
with text strings. Text to be searched for should be entered in the *Value* column enclosed within single quotes, and the LIKE condition uses the standard wild cards: "\*" and "?". Dates, however, should be entered in the #dd/mm/yyyy# format, but the system is fairly tolerant and will usually recognise other formats and often even supplies missing # hash surrounds.

As soon as a further selection is entered on the next line, the left hand *Operator* column becomes active to permit a choice between AND and OR operators. There is, however, no provision for parentheses so care is needed in constructing complex Boolean criteria. Furthermore there is no provision for more than three criteria lines.

Here again the *Apply filter* icon will automatically be set to apply (and consequentially toggle) the specified filter as soon as the OK button is clicked.

### The Form-based Filter

 *Form-based Filter* - The Form-based filter is available only on the Form Navigation Toolbar. In many respects it is less intuitive but more powerful than the *Standard Filter* option. When the icon is clicked, it displays an empty record and a new small Form Filter toolbar. In the case of a grid/table control the display will take the form of a single empty record row together with the header row from the table and any List or Combo Box form controls from the data form to assist selection of filter criteria:

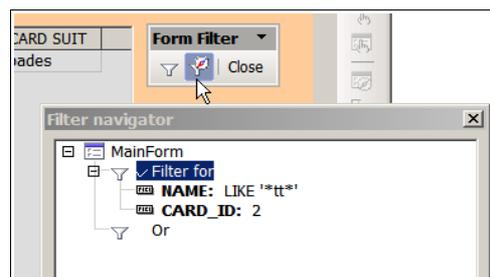


Filtering criteria can be directly entered in the relevant fields. In this example, the filter is set to show only those records that have a double 't' anywhere in the Name field (by using the \* wild card) AND have 'Spades' as the Card suit:

The filter is then applied by clicking the left hand filter funnel icon in the new tiny "Form Filter" toolbar. In this example, application of the filter caused just two matching records to be displayed:

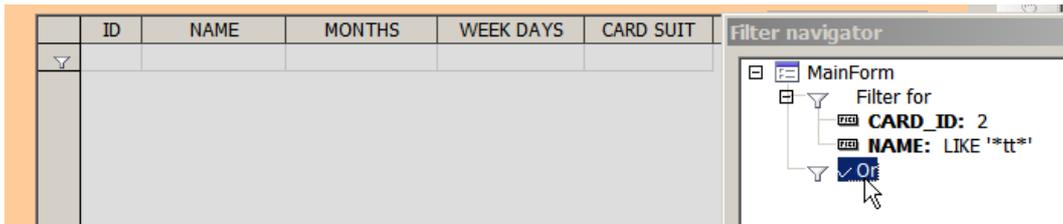
	ID	NAME	MONTHS	WEEK DAYS	CARD SUIT
▶	419	Tttlgdq	April	Friday	Spades
	429	Okttofxv	December	Thursday	Spades
+	Field>				

If the centre icon on this Form Filter toolbar is selected it displays a new "**Filter navigator**" dialog box listing the criteria in a tree control:

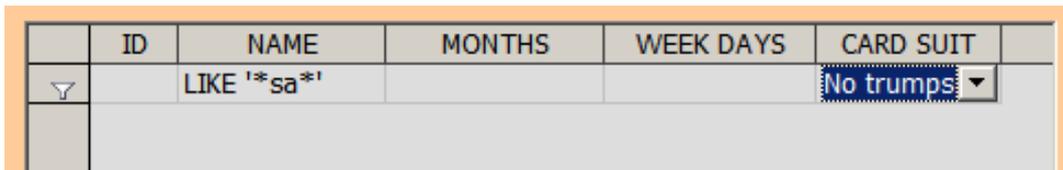


The power of the form-based filter now begins to be revealed. For example click on the empty OR line in the **Filter navigator** and a new empty form is

available:



to permit a further set of filter criteria to be entered that will be OR'd with those previously entered:

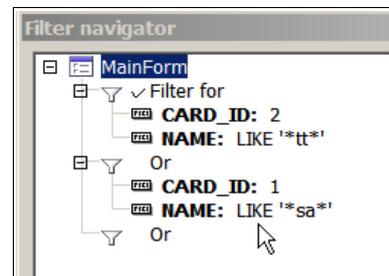


Now when the hour glass icon in the Form filter toolbar is clicked, it produces:

ID	NAME	MONTHS	WEEK DAYS	CARD SUIT
226	Diewszazp	November	Friday	No trumps
419	Tttlgdq	April	Friday	Spades
429	Okttofxv	December	Thursday	Spades
542	Khutsaoa	August	Thursday	No trumps

It has added two further records as matching alternatives to the previous filter.

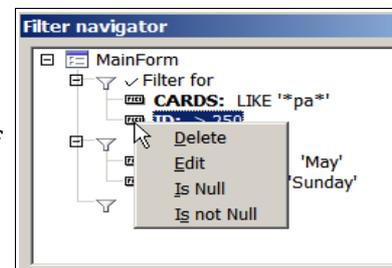
If the Form-based filter icon is now again selected and the **Filter navigator** dialog opened, it will be seen that the criteria list comprises two separate AND selections OR'd together. It is the SQL equivalent to: WHERE CARD\_ID = 2 AND NAME LIKE '\*tt\*' OR CARD\_ID = 1 AND NAME LIKE '\*sa\*'. Further OR'd criteria can be added, but I have yet to test the limit to the complexity that is available.



Thus, the Form-based filter may be less user friendly than the Standard filter but it is not limited to just three criteria. It is, however, all too easy to make errors if the correct order of entering criteria and of selecting the filter icon in this **Filter navigator** dialog is not strictly followed.

It does, however, have a couple of further tricks.

First, it is possible to right click on any entry in the **Filter navigator** and select from the options to delete or edit that criterion directly or to replace it with one of two NULL criteria.



Secondly, it is possible to move the filter criteria lines from one section to another in the **Filter navigator** by dragging and dropping, or by using the keys Ctrl+Alt+Up Arrow or Ctrl+Alt+Down Arrow. It is even possible to copy (and paste) a criterion by dragging it to a new section while holding down the Ctrl key. In short, it is a powerful filtering method, but dangerous in that great care is needed to avoid inadvertent errors.

## The Remove Filter/Sorting icon

 *Remove Filter/Sorting* - The Remove Filtering/Sorting icon is, also, available on both the Toolbar and the Form Control and its operation is similar in either case.

Its effect is to remove and destroy any previously defined filter. Restoration of a filter will therefore require re-definition all over again.

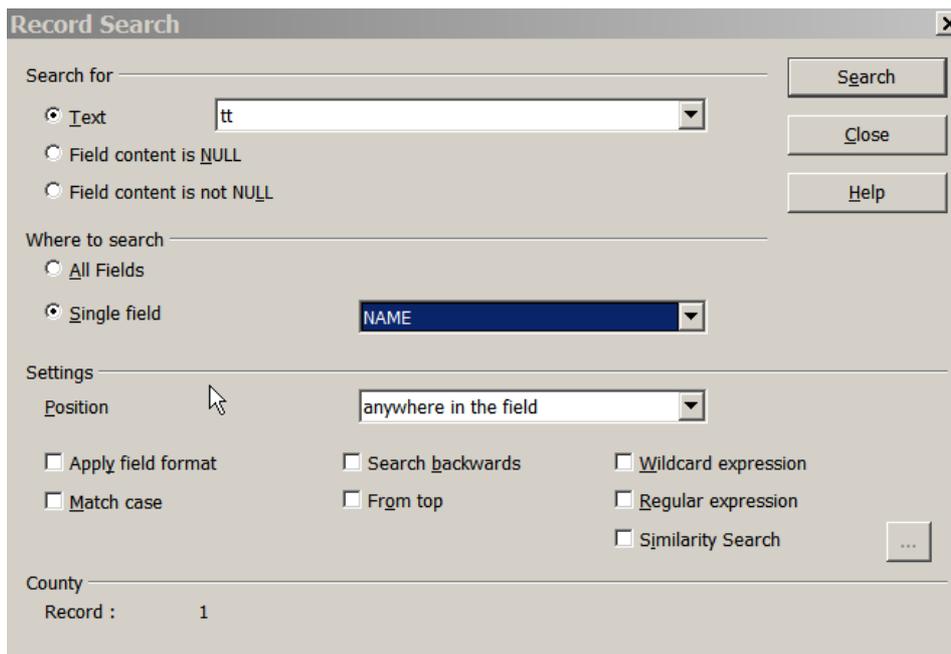
**Note:** it also removes any sorting of the records in the data form that may have been previously applied by use of the sort icons from either bar.

## Finding records

There is a further method by which to select records from a large database. It works in much the same way as finding text in a Writer document.

### The Find Record icon

 *Find Record* icon - is available only from the first, or extreme left hand, icon on the Form Navigation Toolbar. Selecting this icon displays the *Record Search* dialog which offers powerful search options:



The *Record Search* dialog box is shown with the following settings:

- Search for:** A text input field containing "tt".
- Search type:** Radio buttons for "Text" (selected), "Field content is NULL", and "Field content is not NULL".
- Where to search:** Radio buttons for "All Fields" and "Single field" (selected). A dropdown menu shows "NAME".
- Settings:**
  - Position:** A dropdown menu showing "anywhere in the field".
  - Checkboxes for "Apply field format", "Search backwards", "Wildcard expression", "Match case", "From top", "Regular expression", and "Similarity Search".
- Buttons:** "Search", "Close", and "Help".
- Footer:** "County" label and "Record : 1".

The options are comprehensive but also quite complex. Surprisingly useful help, however, will be found by clicking its *Help* button.

Enter the text (or numerals) to be searched for in the *Search for* box. Alternatively the search may be made for where the *Field content is NULL* or *Field content is not NULL*: useful to detect NULLs in a database that are often the source of unexpected errors.

Numerals, even when searched for in numeric fields, are also entered in the box with the *Text* radio button selected. As numbers, however, are often stored in the database in a data type that differs from the format used for the related field/column in the data form, it is advisable to set the *Apply field format* check box in the lower part of the dialog.

The description of this check box is somewhat ambiguous: when set, however, it permits the entry in the *Search for* box to be in the same format as that used by the relevant field in the data form and the search will still be effective even if it differs from the data type used by the source table. Date and time fields are often displayed in formats that differ from the data type used in the database so this option could well prevent errors. If the box is left unchecked searches will be faster - as there is no need to display each record when searching - but the time saving is only justified where the database is large and the searched for *Text* is known to match the data type in the database.

Previous searches are usefully remembered by the text box and can be recovered from its drop down list.

In the *Where to search* section the search can be made through all fields or just in a single field. When selecting a single field a list of searchable fields is presented by the drop down symbol, but is limited only to those fields in the data source that are applicable to the data form. If the form document holds more than one data form or sub-forms, a *Forms Combo Box* will appear so that the search may be restricted to the source data for any particular data form.

The *Settings* section provides further options, such as case insensitive searches when the *Match case* box is not checked. The *Position* of the searched for text can be the entire field, just part of it at the beginning, at the end or anywhere in the field. This choice, however, is disabled if wild cards are to be used by checking the *Wildcard expression* box: '?' and '\*' are used to represent any one or several missing characters respectively. If '\*' is applied both before and after the searched for text, it becomes the equivalent to searching for a match anywhere in the field. Alternatively *Regular expressions* can be selected and these are the same as used in the OOo *Find and Replace* dialogs. The third of the mutually exclusive options, *Similarity Search*, can be selected to search for text that differs from the required text by a defined number of characters. This is a powerful feature and fun to experiment with, though I have yet to find an actual need for it. Again the Base *Help* file is surprisingly helpful in describing how to use it.

## ***Use of Navigation Toolbar with multiple Data Forms***

When a form document contains more than one data form, it is necessary to retain control of how the single Navigation Toolbar is applied to each of them. There is a "*Navigation bar*" property under the "*Data*" tab for each data form: it has three options - Yes, No or Parent. These determine the action of the Toolbar when the cursor moves to, or is in, that particular data form. *Yes* means the Toolbar then applies to that data form. *No* means the Toolbar is not even visible. *Parent* in a sub-form's property means the Toolbar still applies to the main data form even when the cursor is in the sub-form: this helps the form document to appear as one unit controlled largely by its main data form.

A Navigation Bar **form control**, however, always belongs to a specific data form. Thus there can be as many Navigation Bar form controls as there are data forms. Care may still be required when locating them on the form document so that the user clearly sees which one applies to which data form.

## 2. Permanent filters

### Form sub-form relationship

The permanent filter depends on the relationship in OOoBase that underlies the link between a data form and its sub-form(s). A typical example would be a set of 'Customer' records as the data source for the main form and a set of 'Order' records for the sub-form. Thus, whenever a Customer is selected in the main form, the Orders relevant to that customer are shown in the sub-form. As the record selected in the main form is moved to a different customer, Base automatically updates the sub-form to display the new set of relevant orders.

There are two key elements in this process. First, the sub-form is refreshed whenever the main form's record pointer is moved. Secondly, the record set available to the sub-form essentially results from an SQL query that could be interpreted as:

```
SELECT * FROM "SubformDataSource"  
WHERE "LinkSlaveField" = "ValueOfLinkMasterField"
```

It could be said the two link fields are the key properties of the sub-form that dictate the content of the WHERE clause which synchronises the sub-form with the main form.

**Note:** To be more specific the SQL query, on which the sub-form is based, is a *parameter query*. While this has no implication for the current discussion, its relevance for extending filtering options will be explored later.

Typically the data source for a sub-form will differ from that for the main form as in the example of the Customer and Orders tables. When the same data source is used for both forms, it means the record set of the sub-form can be changed by whichever record is pointed to in the main form: in effect the sub-form is filtered by whatever is entered in the *Link Master* and *Slave Fields*.

The "**FilterExamples.odt**" database file was created to serve as an example. It stores its main data in a table named "DATA", which has five fields: the "ID" field is an auto incrementing *Primary Key*:. the "NAME" field comprises illustrative random strings of varying length; the remaining three integer fields serve as *Foreign Keys* for related tables holding the names of months, of days in the week and of playing card suits.

	Field Name	Field Type
	ID	Integer [ INTEGER ]
	NAME	Text [ VARCHAR ]
	MONTH_ID	Integer [ INTEGER ]
	WEEK_ID	Integer [ INTEGER ]
	CARD_ID	Integer [ INTEGER ]

Within the "**DEMO data sub-forms**" folder of this database file a number of form documents demonstrate this filtering principle. In each case the "DATA" table is the data source for both the main form and for the sub-form: for each the data tab *Content type* property holds 'Table' and the *Content* property holds 'DATA'. The content of *Linking fields* properties, however, differ between form documents.

**“A. Simple sub-form”** has the “NAME” field set as the link:

The screenshot shows two parts of an Access form. The top part is the 'Main data form' with a table containing 8 records. The bottom part is the 'Data sub-form' which currently displays only one record, '4 Jomqr'. To the right, the 'Form Properties' dialog is open, showing the 'Data' tab. The 'Link master fields' and 'Link slave fields' are both set to 'NAME'. Other settings include 'Content type' as Table, 'Content' as DATA, and various permissions set to Yes or No.

ID	NAME	MONTH_ID	WEEK_ID	CARD_ID
1	Cydqmb	1	1	1
2	Vmbhv	12	6	2
3	Ectfaomq	5	6	5
4	Jomqr	2	7	5
5	Zghbfe	10	3	3
6	Belfvtyq	4	2	2
7	Otyqbksj	4	3	3
8	Gkcsjzn	1	3	2

ID	NAME	MONTH_ID	WEEK_ID	CARD_ID
4	Jomqr	2	7	5

There are few duplicates in the NAME field so when the record pointer is moved to any record in the main form, the sub-form is in effect *filtered* to display the data for just that one record. The pointer in the main form can be moved to any of the 859 records, but data for only the record(s) with a match in the *Link Master* and *Slave fields* will be shown in the sub-form.

**“B. Single filter sub-form”** has the “MONTH\_ID” field as the link between the form and sub-form. In this case several records in the Main data form have the same value in the “MONTH\_ID” field so selection of any one record in the Main data form lists all matching records in the sub-form:

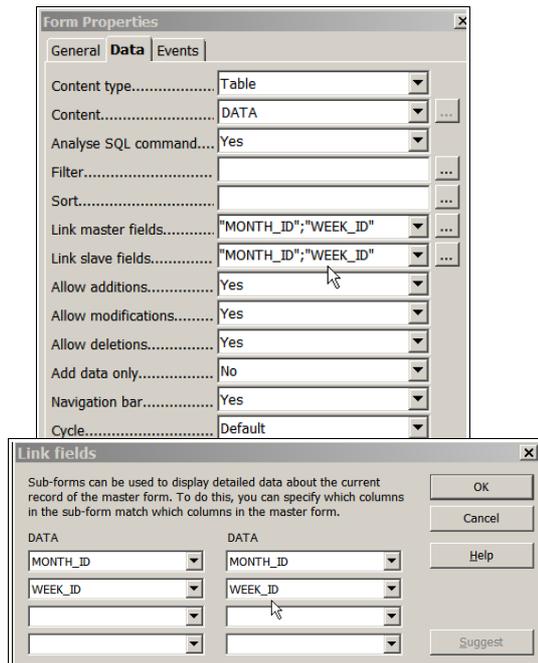
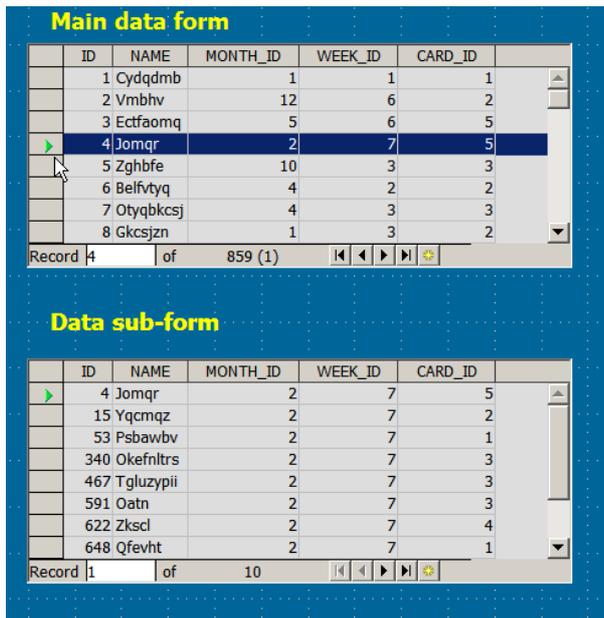
The screenshot shows two parts of an Access form. The top part is the 'Main data form' with a table containing 8 records. The bottom part is the 'Data sub-form' which currently displays 11 records, all with 'MONTH\_ID' equal to 2. To the right, the 'Form Properties' dialog is open, showing the 'Data' tab. The 'Link master fields' and 'Link slave fields' are both set to 'MONTH\_ID'. Other settings include 'Content type' as Table, 'Content' as DATA, and various permissions set to Yes or No.

ID	NAME	MONTH_ID	WEEK_ID	CARD_ID
1	Cydqmb	1	1	1
2	Vmbhv	12	6	2
3	Ectfaomq	5	6	5
4	Jomqr	2	7	5
5	Zghbfe	10	3	3
6	Belfvtyq	4	2	2
7	Otyqbksj	4	3	3
8	Gkcsjzn	1	3	2

ID	NAME	MONTH_ID	WEEK_ID	CARD_ID
4	Jomqr	2	7	5
11	Submmn	2	3	1
15	Yqcmqz	2	7	2
17	Dwgqrrhv	2	5	4
22	Xbihigpp	2	3	2
25	Mmfg	2	6	4
29	Feqhmod	2	6	5
35	Slpsgz	2	4	3

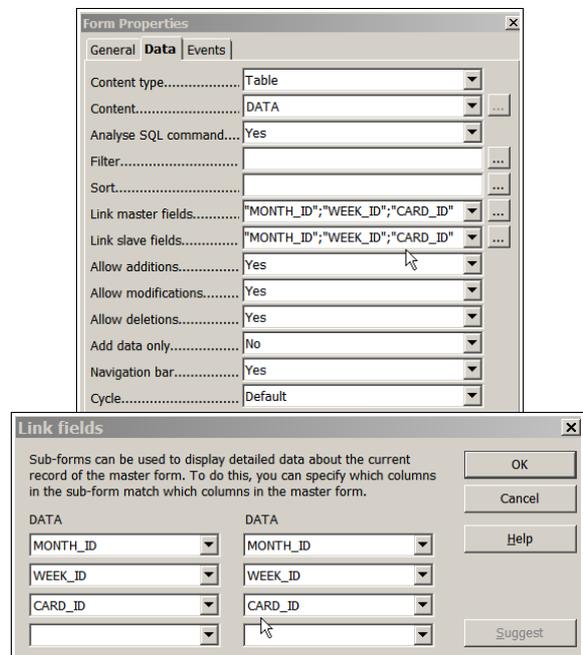
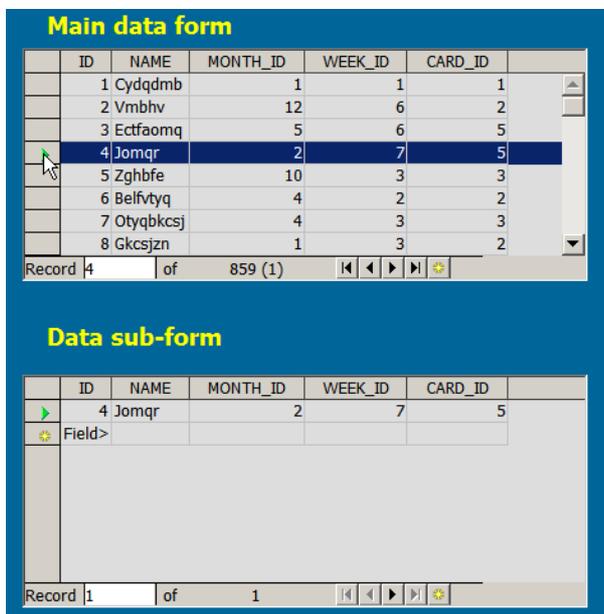
Thus, when the record pointer is moved to the fourth record in the main form, the sub-form is filtered to show all 66 records in the record set that have the same MONTH\_ID as the selected record. Again, if the pointer were to be moved to the 5th record in the main form, the sub-form would be filtered to display the 90 records that had 10 in their “MONTH\_ID” field.

**“C. Double filter sub-form”** has more than one field as the links between the Main and Sub data forms. It demonstrates that the filters are ANDed when more than one field is entered in the *Link Field* boxes:



Thus, moving the main form's record pointer to the 4th record means the sub-form is now filtered to display just the 10 records that match both the “MONTH\_ID” field AND the “WEEK\_ID” field.

**“D. Triple filter sub-form”** adds a further field to the *Link Field* boxes which further restricts the filter effectively applied to the sub form:



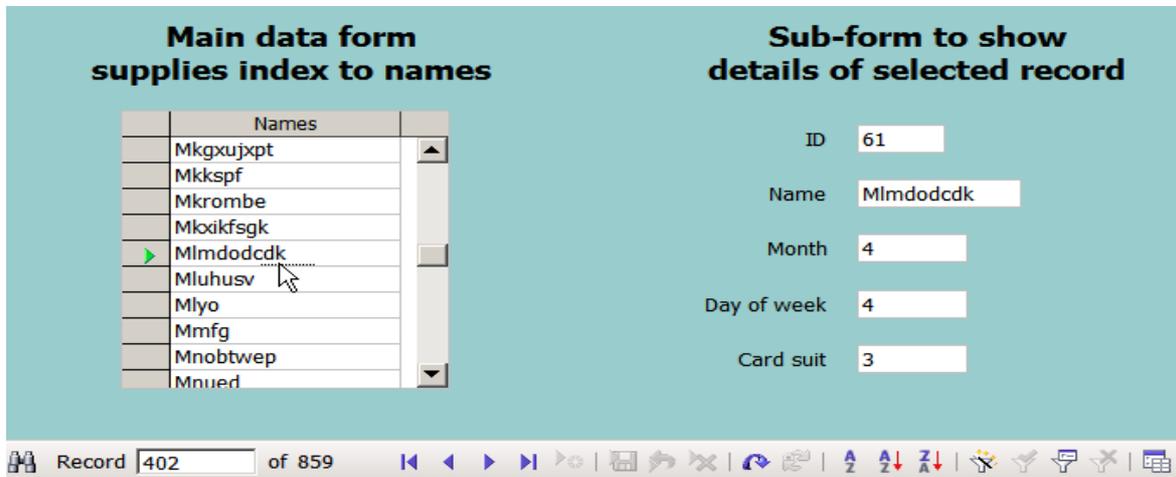
As above, two points should be noted.

First, whenever the record pointer in the main form is moved the sub-form is refreshed. Secondly, the sub-form record set is implicitly filtered by the equivalent SQL query:

```
SELECT * FROM "DATA"
WHERE "MONTH_ID" = 2 AND "WEEK_ID" = 7 AND "CARD_ID" = 5
```

## Indexing example

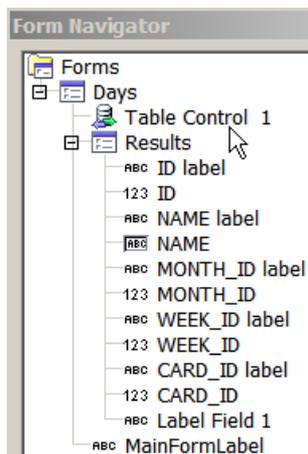
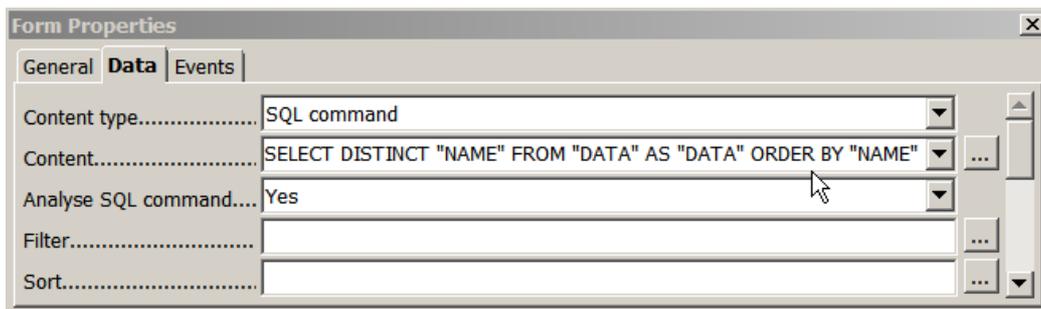
Some more practical examples will be found in the “2. Permanent filters” form documents folder of the example database. In particular, the “A. Index” form document demonstrates how the main data form can serve as an index to a sub-form:



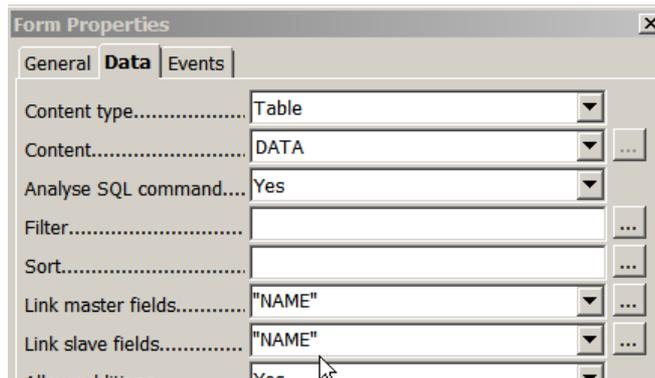
Such an index is better suited with fewer names to choose from, but it does serve to illustrate how a selection change or move of the record pointer in the main form automatically refreshes the sub-form to show the selected record's full content.

List and Combo Box form controls in OOo Base are seldom suitable for this purpose. Changing the selection in a List or Combo box does not move the record pointer: at most, it merely updates the selection's bound field. In place of a List or Combo box, the Main data form comprises a *Table/grid* form control with a single *Text* column control:

### Main form properties



### Sub-form properties

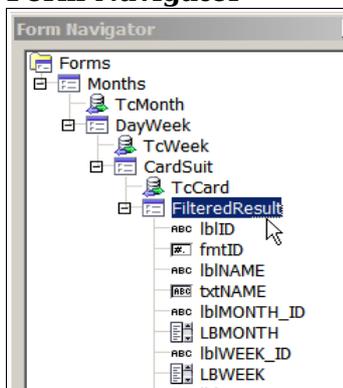


## Multiple filtering example

The same principle can be applied to create multiple filters. The “**D. Triple filter**” form document in the “**2. Permanent filters**” folder demonstrates how selections from the list of months, from the list of days and from the list of Cards are ANDed to filter the sub-sub-sub-form.

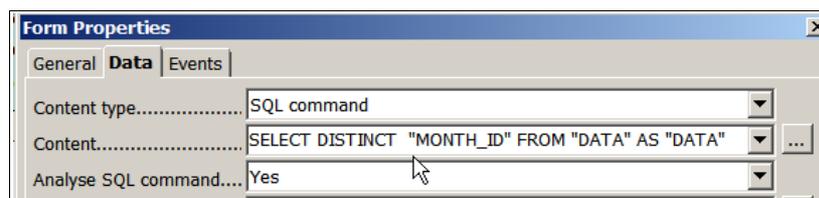
In this example, *Table/grid* form controls are again used for each of the top three data forms to ensure progressive refresh of each lower form whenever the record pointer in the superior form is moved. The single column controls in these *Table/grid* form controls, however, are *List boxes* to assist selection of the required filter criteria by display of user friendly names obtained from the respective tables to each *Foreign key*:

### Form Navigator

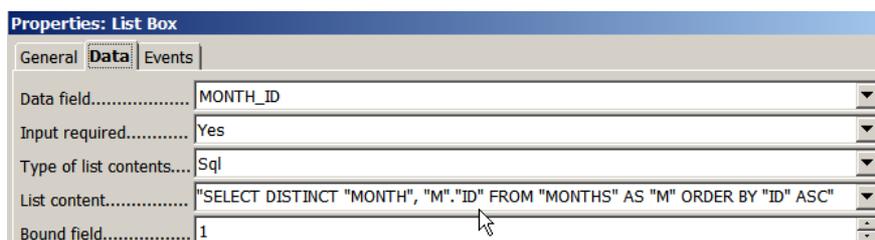


The Form Navigator on the left shows the hierarchical relationship between these data forms, while the data source properties for the Main data form are:

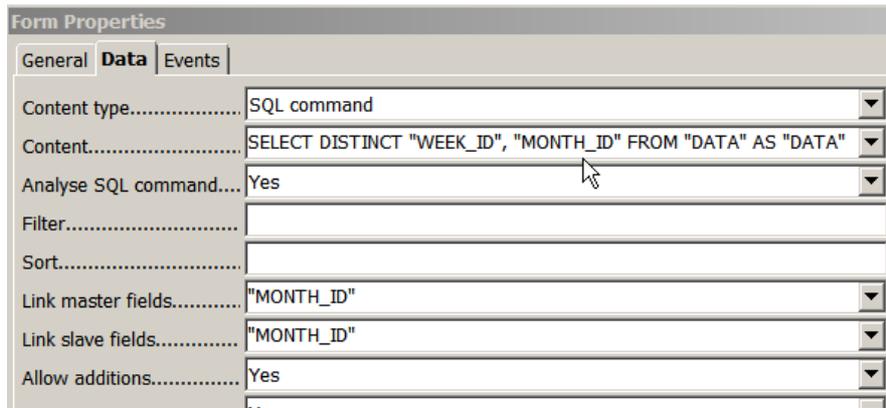
### Main data form properties



The effect is that the record-set for the Main data form comprises the 12 “MONTH\_ID”s from the “DATA” table. The properties of the List Box column control in its *Table/grid* form control, however, ensures display of the names from the “MONTHS” table while the bound field remains the “MONTH\_ID” field from the “DATA” table:



Hence, any change in selected month name moves the record pointer in the Main data form and triggers the automatic refresh and filtering of its immediate sub-form (“**DayWeek**”). The key data properties of this sub-form are more complicated:

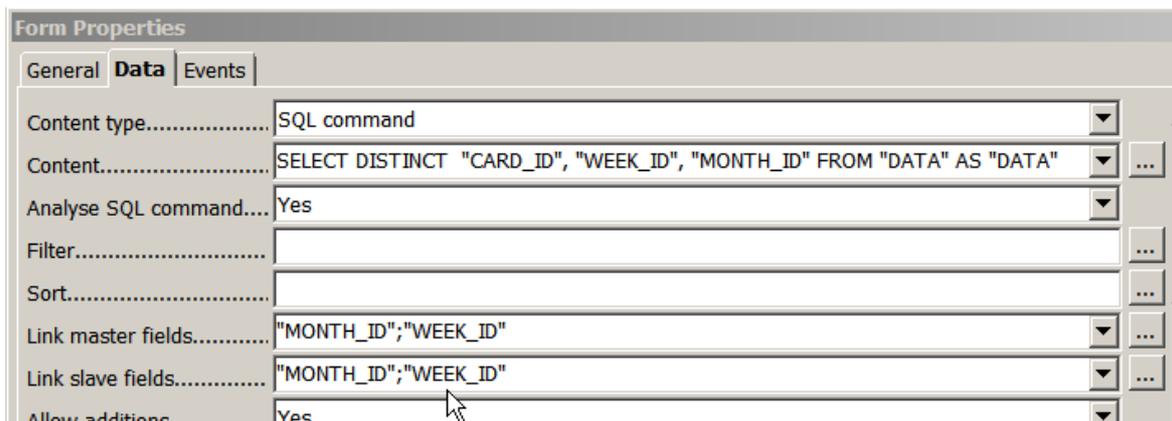


The record-set is again an SQL command but both the “WEEK\_ID” and “MONTH\_ID” fields from the “DATA” table must be included to ensure continuity through the hierarchy of sub-forms. This sub-form is similar to the main form in that it comprises a *Table/grid* form control with a single List box column control, but the latter now bound to the the “WEEK\_ID” field in the “DATA” table and displays the names of days from the “DAYS” table. Thus the *List content* properties of this List box are:

```
SELECT "DAYS", "ID"
FROM "DAYS" AS "DAYS" ORDER BY "ID" ASC
```

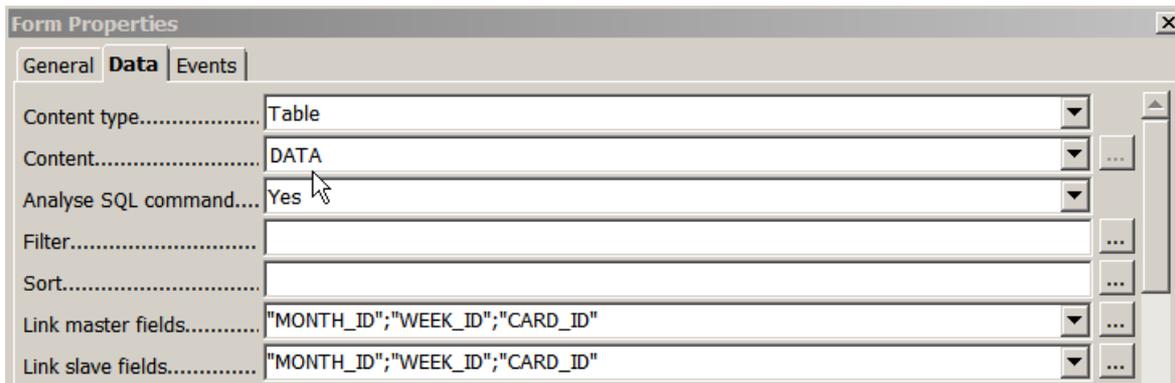
Similarly any selection change in the List box moves the sub form's record pointer and cascades a refresh down through the hierarchy of sub-forms.

The next level is the sub-sub-form (“**CardSuit**”) that similarly maintains the links through the cascade as its record set includes all three fields. To ensure continuity with the senior forms two fields must now be listed in the *Link master* and *Link slave fields*

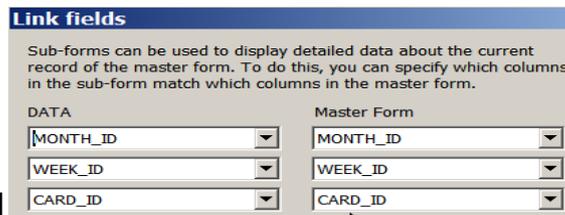


The final sub-sub-sub-form (“**FilteredResult**”) in this example uses a normal form layout that shows each record individually. A grid/table form control would be equally acceptable and circumstances should dictate the choice.

The record set for this ultimate sub-form, however, can include any number of fields and the example uses the “DATA” table as its data source:



Three fields are now required to maintain the links with the ancestor forms. The illustration to the right shows how these can be easily entered by clicking on the ... ellipsis beside the *Link master fields* box and selecting options from the drop down boxes in the dialog that opens.



List boxes are, also, used again in this final “FilteredResult” form to permit display of user friendly names rather than IDs. Additions, deletions or other data changes in this form may, however, give rise to the bug in OOo 3.2.

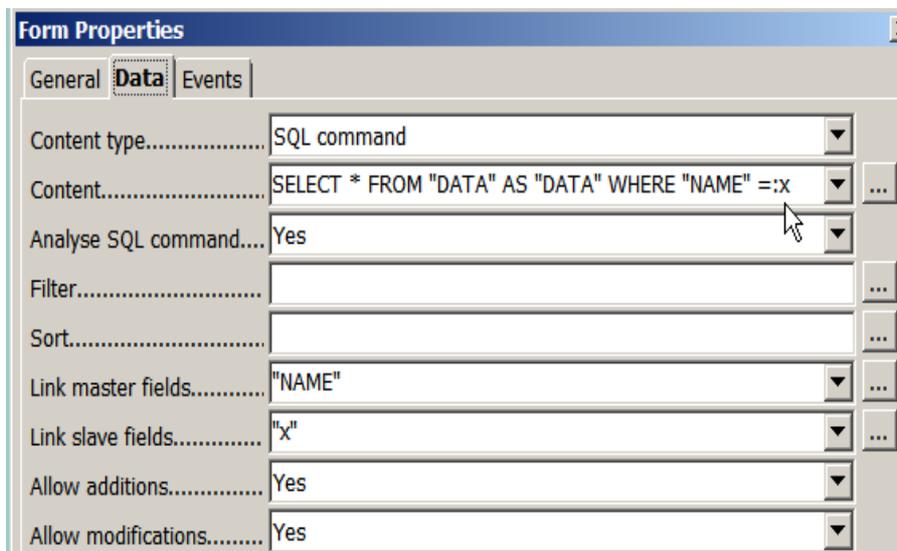
### 3. Power filtering

I am indebted to Villeroy who has promoted and described this method. Before describing it in detail, I refer again to this statement in the OOo Base Help file:

*“The sub-form is based on an SQL query; more specifically, on a Parameter Query”*

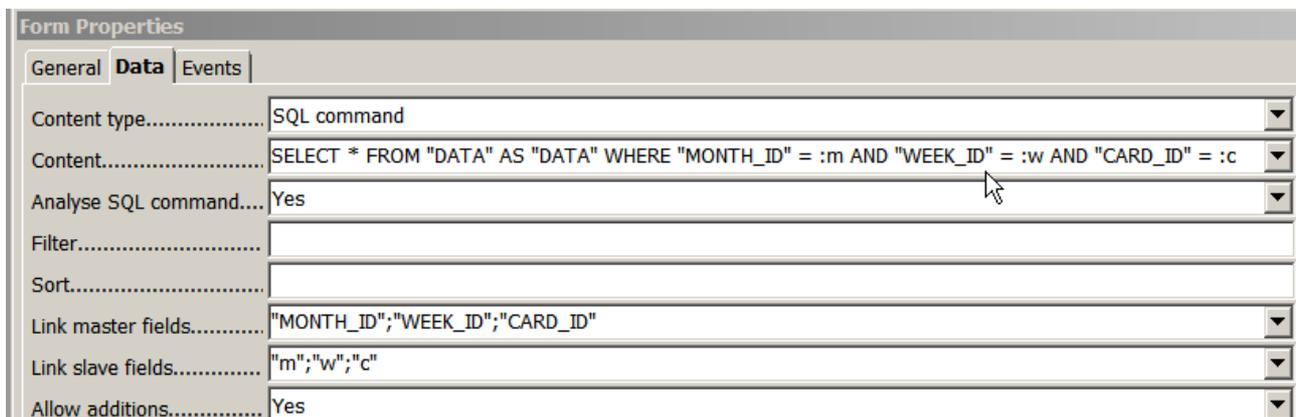
This somewhat cryptic definition means that it is possible to use an explicit parameter query as the data source for a sub-query.

The sub-form examples described so far have used field name(s) in the *Link slave fields* property. The **“Aa. Index (using parameter)”** form document in the example database, however, demonstrates that a parameter variable “x” could equally be used in the *Link slave field* property of the sub-form and an explicit parameter SQL query then used as the sub-form's data source:



The “NAME” field of the main form is still the content of the *Link master fields* box, but the effect now is to pass the value held in that NAME field to the variable “x” and thus automatically to supply that value to the WHERE clause in the parameter query.

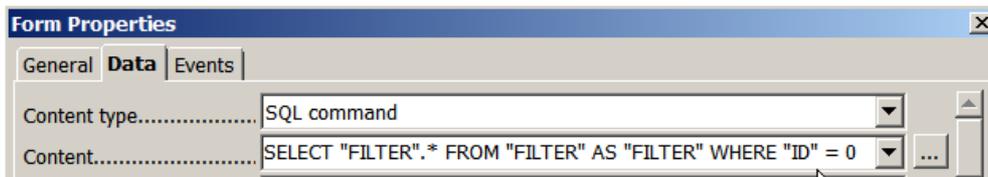
Similarly if there were to be multiple links between form and sub-form, multiple parameter variables can be used. The **“Da. Triple filter (using parameters)”** form document demonstrates how its “FilteredResult” sub-sub-form is changed from the one previously described on page 18 above to make use of a multiple parameter query:



Practical application of this feature will permit yet more flexible and complex filtering. The “**3. Power filtering**” form document in the example database demonstrates how parameter queries can be exploited.

The method requires an extra (“FILTER”) table which holds a single record with as many fields as are needed for the various filtering criteria. This table is used to store the values against which the filtering criteria will be matched. Those values are set in a row of boxes in the main data form whose data source is restricted to the first record in the FILTER table. The key data properties for the main form comprise this SQL command:

Field Name	Field Type
ID	Integer [ INTEGER ]
MONTH	Text [ VARCHAR ]
WEEK	Text [ VARCHAR ]
SUIT	Text [ VARCHAR ]
NAME	Text [ VARCHAR ]



Thus in the form document any data entered in or selected from the upper row of boxes stores those values in the FILTER table. Furthermore, if the contents are cleared from any box a NULL will be stored in the bound field in the FILTER table. The effect is to remove that that filter element. The following illustration shows that if all boxes are empty no filter is applied so a complete records set is available to the sub-form:

ID	Name	Day of Week	Month	Card suit
1	Cydaqmb	Tuesday	January	No trumps
2	Vmbhv	Saturday	December	Spades
3	Ectfaomq	Saturday	May	Clubs
4	Jomqr	Sunday	February	Clubs
5	Zghbfe	Wednesday	October	Hearts
6	Belfvtyq	Tuesday	April	Spades
7	Otyqbksj	Wednesday	April	Hearts
8	Gkcsjzn	Wednesday	January	Spades
9	Hznw	Friday	June	Diamonds
10	Nkpcpubm	Tuesday	November	Spades
11	Submmn	Wednesday	February	No trumps
12	Wnujggeq	Friday	June	Hearts
13	Gneanzcnc	Wednesday	September	Spades

As selections are made or data entered in the top row of boxes those values are stored in the FILTER table. This action, however, does not refresh the sub-form automatically because there has been no change yet to the record pointer in the sub-form's data source. It is therefore necessary to add an “Update form” button control to the sub-form. The button's *Action* property is set as 'Refresh form' so that when it is clicked the sub-form is updated and its record set filtered by the new criteria:

Name: y Day of week: Tuesday Month: January Card suit: No trumps

Update form

ID	Name	Day of Week	Month	Card suit
1	Cydqdm	Tuesday	January	No trumps
477	Kkyxe	Tuesday	January	No trumps
:AutoField>				

Record 1 of 2

The filtering of the sub-form is implemented by the SQL parameter query that serves as its data source:

Form Properties

General Data Events

Content type: SQL command

Content: SELECT \* FROM "DATA" AS "d" WHERE (UPPER ( "d"."NAME" ) LIKE ( '% ' || UPPER ( :n ) || ' % ' ) OR :n IS NULL ) AND ( "MONTH\_ID" = :m OR :m IS NULL ) AND ( "WEEK\_ID" = :w OR :w IS NULL ) AND ( "CARD\_ID" = :c OR :c IS NULL )

Analyse SQL command: Yes

Filter:

Sort:

Link master fields: "MONTH";"WEEK";"SUIT";"NAME"

Link slave fields: "m";"w";"c";"n"

The full text of this parameter query uses four variables (n,m,w,c):

```
SELECT * FROM "DATA" AS "d"
WHERE (UPPER ( "d"."NAME" ) LIKE ( '% ' || UPPER ( :n ) || ' % ' ) OR :n IS NULL)
AND ( "MONTH_ID" = :m OR :m IS NULL )
AND ( "WEEK_ID" = :w OR :w IS NULL )
AND ( "CARD_ID" = :c OR :c IS NULL )
```

This filtering method has many advantages. It imposes no limit on the number of criteria, and permits for each a variety of comparators. Thus, this example makes use of =, LIKE, IS NULL, OR as well as AND but any other comparator understood by SQL can equally be used.

For example, changing the first AND to OR and adding parenthesis gives:

```
SELECT * FROM "DATA" AS "d"
WHERE (UPPER ( "d"."NAME" ) LIKE ( '% ' || UPPER ( :n ) || ' % ' ) OR :n IS NULL)
OR ( ( "MONTH_ID" = :m OR :m IS NULL )
AND ( "WEEK_ID" = :w OR :w IS NULL )
AND ( "CARD_ID" = :c OR :c IS NULL ) )
```

Name: y OR Day of week: Tuesday Month: January Card suit: No trumps

Update form

ID	Name	Day of Week	Month	Card suit
1	Cydqdm	Tuesday	January	No trumps
6	Belftyq	Tuesday	April	Spades
7	Otyqbksj	Wednesday	April	Hearts
15	Yqcmqz	Sunday	February	Spades
21	Frzy	Wednesday	September	Spades
27	Lkmjogyt	Tuesday	May	Hearts
28	Dgyxteq	Sunday	March	Diamonds
30	Endvrv	Monday	September	Spades

Record 1 of 228

It now shows that many more records had a 'y' in the NAME as well as those that matched the Day/Month/Card specification:

## Removing filter criteria

The important “OR :parameter IS NULL” phrase is included in the SQL query for each of the criteria in order to preserve the integrity of the WHERE clause whenever a criterion is to be removed from the filtering process. In the case of the Text box form control it is easy to delete the contents to set its bound field to a NULL. In the case of List box controls it is less easy to delete entries. It can be done by first pressing the HOME key and then the DELETE key, but that is not immediately intuitive. A better work around is to include an <All> option as one of the selections to be provided in the List box.

It requires a further small table - “ALL” - which has just two fields and one fixed record as illustrated on the right. The ID field in the single record is left as a NULL, and for that reason the **NAME field must be set as the primary key**.

	Field Name	Field Type
	ID	Integer [ INTEGER ]
	NAME	Text [ VARCHAR ]

A UNION query is then used to add the <All> option to the list contents of each List box control. It requires the list box control, for example (“lboMonth”) in the main form, to have its *Type of list contents* property set as '**SQL [native]**' to run the SQL command directly because OOo Base cannot yet interpret a UNION query. Then, its *List content* property would be:

	ID	NAME
		<All>

```
SELECT "MONTH", "ID" FROM "MONTHS" AS "MONTHS"
UNION
SELECT "NAME", "ID" FROM "ALL"
ORDER BY "ID" ASC
```

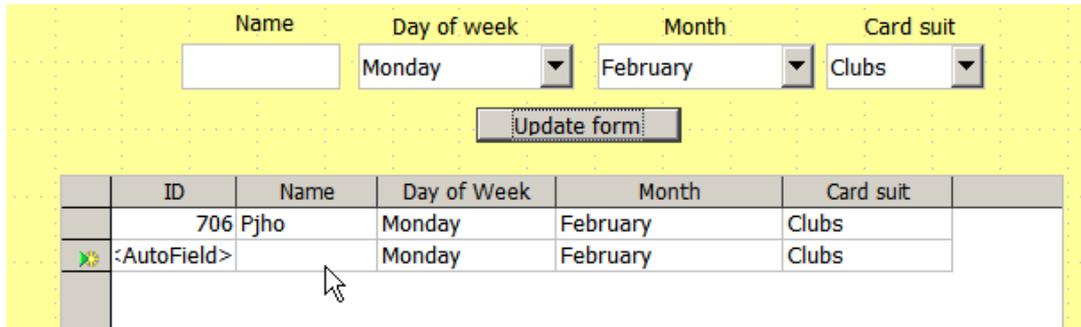
MONTH
<All>
January
February
March
April
May
June
July
August
September
October
November
December

Thus, the UNION query adds <All> to the top of the selection options. When that item is selected, its related ID field in the ALL table is a NULL and this NULL is stored in its bound field in the FILTER table. Thus, selecting <All> removes that particular criterion from the filtering process.

## 4. The OOo 3.2.? bug

It should be possible to edit, change or delete any record displayed in a sub-form provided its data source is derived solely from a single table in the database. This should still apply even when a query is used as the data source and should not be affected if the WHERE clause is used to apply a filter.

A further benefit offered by such filtering should be that any field used in the filter will be pre-filled by the criteria data when the record pointer is moved to a new empty row in order to create a new record. This feature can be useful when entering several records with the same data in certain fields as shown in this example where only new data for the *Name* field need be entered:



The screenshot shows a sub-form interface with a yellow background. At the top, there are four input fields: 'Name' (empty), 'Day of week' (dropdown menu showing 'Monday'), 'Month' (dropdown menu showing 'February'), and 'Card suit' (dropdown menu showing 'Clubs'). Below these fields is a button labeled 'Update form'. Below the button is a table with the following columns: ID, Name, Day of Week, Month, and Card suit. The table contains two rows: the first row has ID '706', Name 'Pjho', Day of Week 'Monday', Month 'February', and Card suit 'Clubs'; the second row has ID ':AutoField>', Name (empty), Day of Week 'Monday', Month 'February', and Card suit 'Clubs'. A mouse cursor is pointing to the second row.

There is, however, a bug in the current 3.2.1 version of OOo Base. When the record set in a sub-form has been filtered by more than one criterion, an attempt to save an added or changed record in the sub-form often throws an error such as “No data is available”. The claim tends to be false, because if OK is clicked the added or changed record will still be correctly saved in the table. The problem is compounded as any repeated effort to save the record leads to a repeat of the error message and to multiple saved records.

While it is possible to work around the problem, it is neither intuitive nor wholly reliable. For serious work with sub-forms filtered by multi criteria, therefore, it would be wiser to wait for Base 3.3 or revert to version 3.1.1.

**[WARNING: This draft is regularly updated, so comments or suggestions for improvements will be welcomed.]**