

#\$K+ Queries: Some Common Examples

If you've been using Simple Accounting for any length of time, you have worked with queries a lot. At the very least, you have selected queries already built by Suntower Systems to handle some basic tasks. But at some point, nearly every user wants to do more than just use the pre-built queries. Explaining the basics of queries is easy enough, but usually what is most challenging is determining which fields in the Simple Accounting database one needs to use to ask the questions you wish to ask.



Using Filters With Sales Orders

One of the most common places to use queries is in the Sales Order Browse. Here are the most common fields that users ask about:

FIELD NAME	POSSIBLE VALUES
Order Status	O = Ordered Q = Customer Quote S = Shipped But Not Invoiced I = Invoiced V = Voided Invoice
Payment Status	O = Open C = Credit (Balance < 0) P = Paid In Full <Blank> (No Invoice; not yet active)
Order Type	N = New E = Exact Repeat B = Changed Copy/Spec
Posted To G/L?	Y/N
Employee ID	The sales person to be credited for this order
Employee ID #2	The second sales person or CSR to be credited for this order. When the order is voided, this is filled with the person who voided it. (If Employee Login is activated.)

Some common filter combinations:

Let's get the hang of filters by creating some common queries with sales orders. We'll get started by listing all open orders within the month of November, 1999:

Order Status	=	0	AND
Invoice Date	Month=	11	AND
Invoice Date	Current Year		

When creating Filters, it is useful to say them out loud, such as the above which can be translated as:

'I want to see all Orders which are Open *and* which happened in November *and* which happened in 1999.'

Got it? Now a nother common type of report is basically the same information, but showing all *invoiced* orders within the month of November. We do this by making one small change

Order Status	=	I	AND
Invoice Date	Month=	11	AND
Invoice Date	Current Year		

We're going off on a tangent for a minute here to demonstrate a really big concept. See that the *only* difference between this query and the open order query is the Order Status field which determines whether the sales order has been invoiced or not. We use the same report, but for a very different purpose. This is a key concept in Simple Accounting and it goes like this:

"Many 'different' reports are actually the *same* report, in terms of the columns of information printed, *but* using a different set of records". Which is what the query does: lets *you* control *which* records print. For example, do you know the difference between a Customer Owned List Of Products Report and an Inventoried List Of Products Report?

A Customer Owned List Of Products Report is a query on the Products Browse whose filter looks like this:

Product Owned By	=	C
Customer ID	=	JOHNSON

An Inventoried Customer List Of Products Report is a query on the Products Browse whose filter looks like this:

Product Owned By	=	I
------------------	---	---

Ok, we're back on track now, talking about Sales Orders again. Next, we want to be able to compare sales during November of 1999 and November of 1998. Which is to say, all sales in the month of November, 1999 *or* in the month of November, 1998:

```
Order Status      =          I          AND
Invoice Date      Month=       11          AND
Invoice Date      Year=        1999       OR
Invoice Date      Year=        1998
```

In this example, note the AND and the OR operators. Most often you use the AND operator since most questions you ask involved a combination of conditions being true such as 'Orders which are something *and* which are something else.' But sometimes you use the OR operator as above to show items which are *either* one thing *or* another.

Here's another example, comparing all invoiced orders in 1999 for salesperson JIM *or* for salesperson MARLA:

```
Order Status      =          I          AND
Invoice Date      Current Year  AND
Employee ID       =          JIM        OR
Employee ID       =          MARLA
```

All invoiced orders which were voided by CSR Janice within the month of November, 1999

```
Order Status      =          V          AND
Invoice Date      Month=       11          AND
Invoice Date      Current Year  AND
Employee ID2      =          JANICE
```

OK, one last thing. The operators. Most of these are pretty easy to get if you remember your arithmetic:

```
>      Greater than
<      Less than
```

...and so on. But what about 'Month=' or that dollar sign?

The Dollar Sign Means 'It's In There Somewhere'

Sometimes, you need to find things which, to the human eye look the same, but to the computer are definitely *not*. For example:

```
First Bank Of Boston
Bank Of America
American Bank
```

Now, you know that all these institutions are banks, and would like to see all of them on a list of customers. So how to tell Simple Accounting:

```
Name          $          BANK
```

Which is to say, 'show me all customers who have the characters B A N K in their name, *somewhere*.

Sorts: The Other Half Of The Equation



Sorts seem to be easier to understand than filters (at least at first), but get a bit trickier once beneath the surface. In general, it is easy to get that the sort determines the *order* in which records print. However, it gets more complicated when you add several sort options in order to not only change the order in which records print, but also how they are subtotalled.

Some common sort combinations:

If we want a list of all sales orders, ordered by salesperson, that's easy enough. Go to the Sales Order Browse, press [Print], then Summary Report. Now press [Insert New] on the query window to create a query and in the lower half of the window enter:

```
Sort #1      Employee ID      +
```

Save your query and select it to continue printing the report. Since this particular report allows you to subtotal, you can then opt to subtotal and get a new page starting with each Employee ID. Why? Because the sort (Employee ID) tells the report on what field to subtotal.

The problem is that with only one sort option, you leave a lot to chance. You're telling Simple Accounting to organize the records by Employee ID, but nothing more. So if each sales person has hundreds of orders, finding a particular one on the report would require you to find the Employee ID in question, then *randomly* search for the order you need. Not good. So try the above example again, but this time, when you get to the sort half of the Query Window, type:

```
Sort #1      Employee ID      +
Sort #2      Order ID        +
```

Much better. Now, our report will subtotal by Employee ID, but within each salesperson's records, the orders will be printed in Order ID order.

But our salespeople want an open order report that lists all the items ever ordered for each customer. No problem. Go to the Sales Order Browse, press [Print], then Detail Report. Now press [Insert New] on the query window to create a query and in the lower half of the window enter:

```
Sort #1      Employee ID      +
Sort #2      Customer ID      +
```

Again, save your query and select it to continue printing the report. Since this particular report also allows you to subtotal, select both the first subtotal *and* the Second Subtotal so that you get a break after each salesperson's items, and then further broken down by Customer ID. You see: the first sort option determines the first subtotal and the second sort option determines that second subtotal.

Of course, this is *still* too vague! This gives us the items, but there isn't anything to tell Simple Accounting how to list the history within each customer. Fortunately, we have two other sort options to play with. So try the above example again, but this time, when you get to the sort half of the Query Window, type:

```
Sort #1      Employee ID      +
Sort #2      Customer ID      +
Sort #3      Product ID       +
Sort #4      Ship Date        -
```

This gives us the results we want: products are organized within each customer by product. But look at Sort #4; see the minus (-) sign? That means sort in *descending* order. Which means that if the customer has bought a particular form several times, the most *recent* order will be printed first and the oldest last.

Descending order is helpful in many situations. For example, lets say we want a list of customers organized by YTD sales. This lets us know who are our best customers. Go to the customer browse, press the [Print] button and select the Totals report. Once again, press [Insert New] on the query window to create a query (see how it works the same way for all reports!) and in the lower half of the window enter:

```
Sort #1      Employee ID      +
Sort #2      YTD Sales        -
```

This will create a list of customers, subtotaled by each Employee ID. And this list will show each salesperson's best customers (those with highest YTD sales) at the top.

So the important things to remember about the plus and minus signs (+/-) are:

1. In ascending order records print from low to high, but in descending order they print from high to low.
2. Ascending order means 0-9 in the case of numbers and A to Z in the case of letters. (Descending order means just the opposite: 9-0 and Z-A.)



Key Concepts:

1. Sort Criteria #1 determines the First Subtotal for reports
2. Sort Criteria #2 determines the Second Subtotal for reports
3. The + or - sign determines if the sort is ascending or descending.

Other Common Fields/Files And Their Uses In Queries

Now that you've learned some important concepts, we should apply these to the other common files you will be using with the Query Window.



A/P (Bills Browse)

FIELD NAME POSSIBLE VALUES

Payment Status O = Open, C = Credit, P = Paid In Full (Closed)
A blank Payment Status means that you have not keyed an invoice number.

Payment Type M = Manual, C = Check

Check Number

Invoice

Some samples using various filter options:

- **A List Of Open Invoices Which Are Partially Paid**
Payment Status <> P AND
Payment Date <>
- **A List of A/P Transactions With No Invoice Number**
Invoice <>
- **A List of A/P Transactions With No Invoice Number**

Invoice <>

- **A List of All Open Credits**
Payment Status = C
- **A List of All Open Invoices & Credits Due In The Current Month**
Payment Status <> C AND
Invoice <> AND
Due Date Current Month AND
Due Date Current Year
- **A List of All Open Invoices With Discounts To Be Taken By 05/01/00**
Payment Status <> O AND
Discount Date >Today AND
Entry Date <= 05/01/00
- **A List of All Invoices & Credits Entered In The Current Month**
Invoice <> AND
Entry Date Current Month AND
Entry Date Current Year
- **A List of Invoices Paid Today**
Payment Date Today
- **A List of Invoices Paid Today From One Bank Account**
Payment Date Today AND
Bank Account = 00590221233



Products

FIELD NAME POSSIBLE VALUES

Spec 1 = Yes, 0 = No
Owned By I Distributor (Bill As Released)
C Customer
V Vendor

Some samples using various filter options:

- **A List Of All Distributor Owned Products**
Owned By = I
- **A List Of All Customer Owned Products**
Owned By = C
- **A List Of All Managed Forms**
FMS = 1
- **A List Of All Distributor Owned Envelopes**
Owned By = I AND
Product Type = ENVELOPE
- **A List Of All Items Tagged For Reordering**
Tagged = Y
- **A List Of Items With No Sales In The Last Year**
YTD Sales Exactly Equal 0.00
- **A List Of Items With On Hand Quantities**
On Hand Quantity > 0.00

G/L

FIELD NAME POSSIBLE VALUES

Posted Y/N Has the record been posted (locked so that it cannot be changed)?
Source Module SOH Sales Orders
APH Vendor Bills
CKR Check Register

Some samples using various filter options:

- **A List Of Entries Generated From A/P During The Period 12 of 1999**
Period = 12 AND
Year = 1999 AND
Source = APH

- **A List Of Entries Generated Today (No Manual Entries)**
Source <> GL
Entry Date Today

- **A List Of Entries Generated For One Employee ID Within The Current Month**
Employee ID = JIM_H
Entry Date Current Month