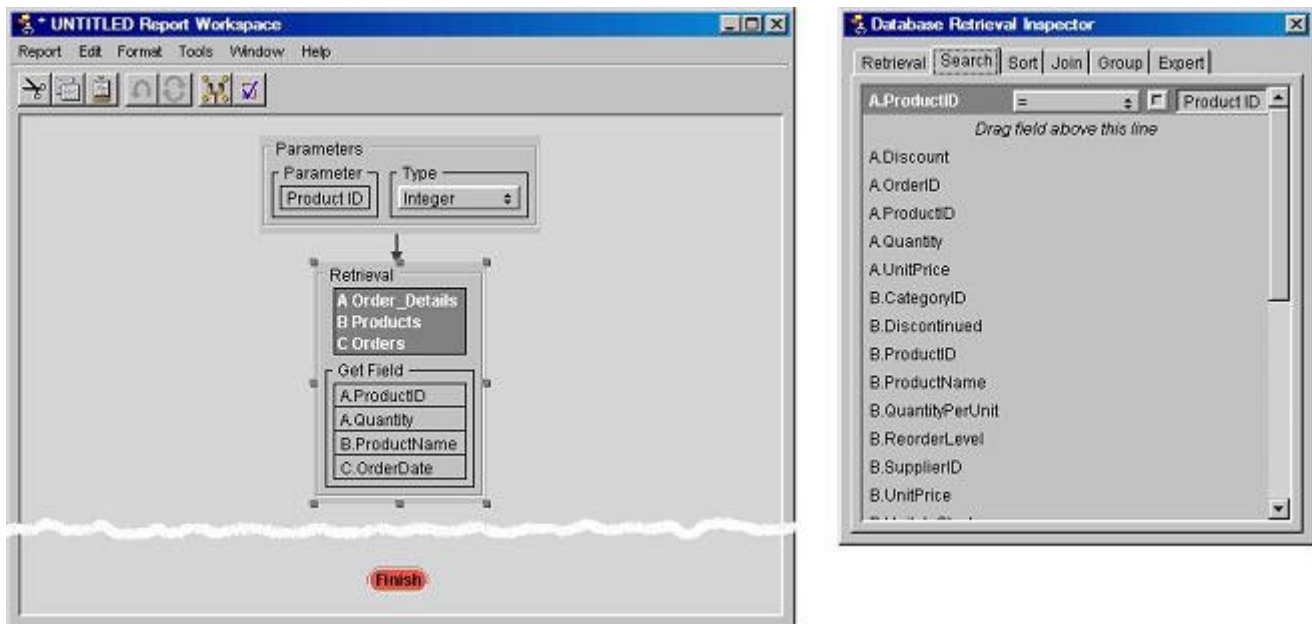


Tutorial 10 - Building the ProductSalesStats Report Procedure - part I

The ProductSalesStats report displays a bar chart showing the quantity of a selected product(s) shipped over the entire sales history of Northwind Traders Company. The main focus of this Tutorial will be on using the Group feature of the Retrieval component, and the Bar Chart executable component.

The Report Procedure takes one input parameter from the user - namely, the product ID number. So, as the first step, select the Parameter component, add a new parameter to it and rename it to "Product ID" (refer to the Tutorial 4 for details). Set the parameter data type to INTEGER.

Next, bring in the Retrieval component, and set it to retrieve data from the following joined tables: Order_Details, Products and Orders. The process of setting up table joins is described in Tutorial 6. The row retrieval is based on the Product ID, as shown in the Inspector:



In this database retrieval we will use the GROUP BY clause of SQL to sum up the Quantities for the given product, ordered on the same date. This element of the SQL SELECT statement is set via a graphical interface implemented in the Group tab of the Database Retrieval Inspector. To set the grouping, select the Group tab in the Inspector, and check the "Group rows by fields" box. The view that is displayed at this point, consists of two parts:

- Grouping fields ("Group by") box
- Aggregate fields ("Aggregate" box)

Grouping fields are used in the GROUP BY clause; they instruct the Retrieval component to generate an SQL statement that would merge all rows having the same content (according to the grouping fields) into one. The remaining (non-grouped) fields in the resulting single row will have their values set according to the aggregate function selected in the "Aggregate" box, e.g. for the SUM aggregate function on field A all values of the field A will be added together, and the sum will be stored in the field A of the returned row.

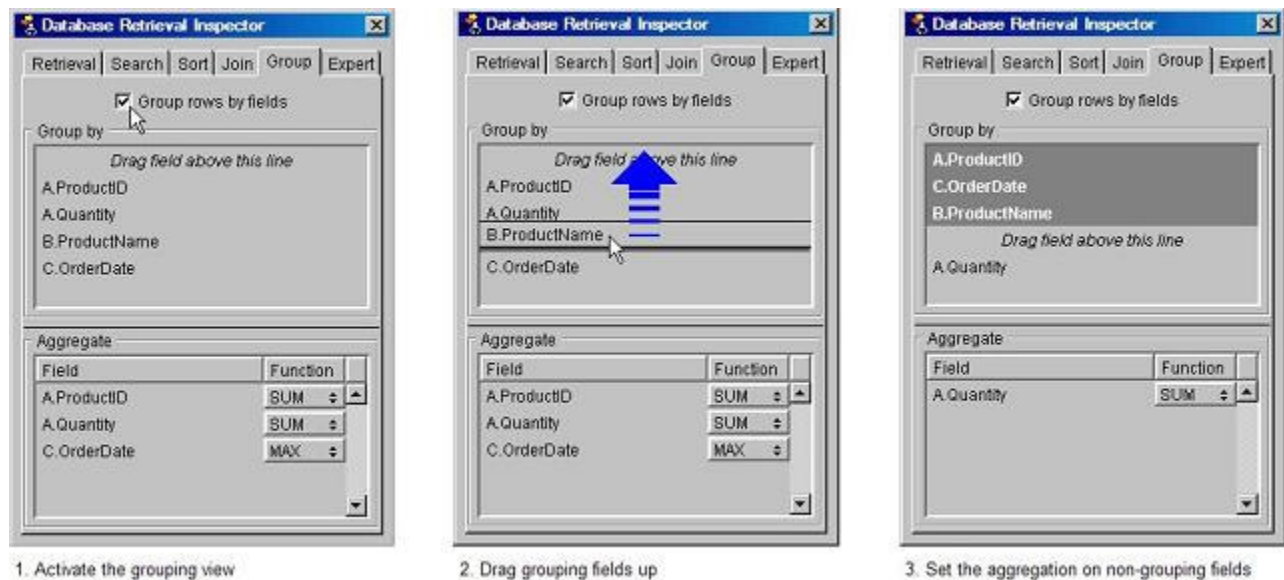
Drag the following fields up to make them grouping fields:

A.ProductID

C.OrderDate

B.ProductName

The steps are shown in the figure below:



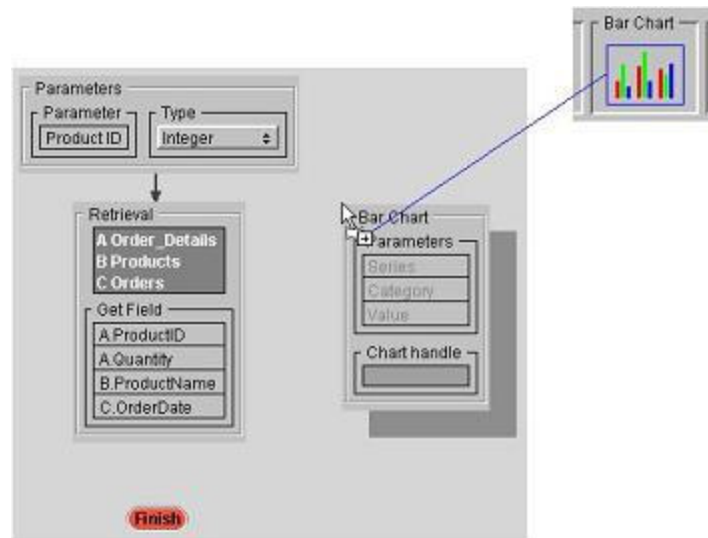
As the fields are dragged up the "Group by" view, they are removed from the "Aggregate" view, because the SQL rules require that only non-grouping fields be aggregated (and, as a flip side of this rule, all retrieved fields not used in grouping, must be aggregatable - i.e. be of the data type that lends itself to SUM, AVG, MAX and MIN aggregate functions. CHAR-based table fields would not be eligible.)

The resulting SQL statement generated by the Retrieval component can be seen in the Expert tab, and is reproduced below:

```
select A.ProductID,C.OrderDate,B.ProductName,SUM(A.Quantity) from Order_Details A
JOIN Products B ON A.ProductID = B.ProductID JOIN Orders C ON A.OrderID = C.OrderID
where A.ProductID = [Product ID] group by A.ProductID,C.OrderDate,B.ProductName
```

(the color scheme, with part of the SQL text shown in gray color, replicates the way the text is displayed in the Expert view. Only the part of the text that is in black color, is directly editable by the user.)

Now, that the extraction of data from the database is taken care of, we can look at the presentation part - namely, the formatting of the Bar Chart. This is done by the Bar Chart executable component:

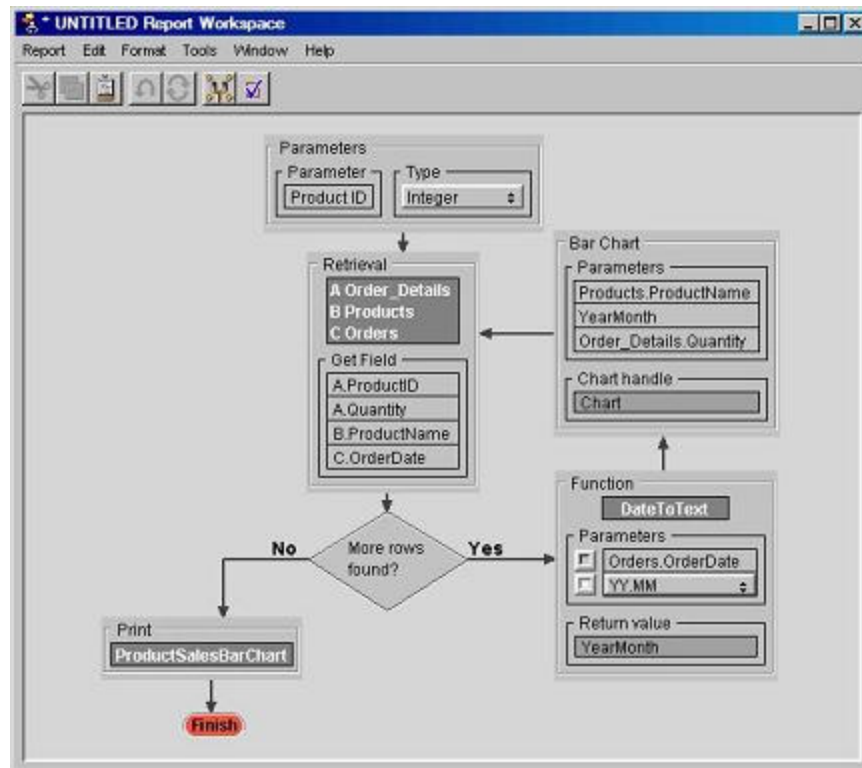


The Bar Chart component takes the following parameters:

- Series name
- Category name
- Numeric value

Category name and numeric value correspond to the marker on the horizontal axis, and the size (height) of the bar drawn at the marker's location. The Bar Chart takes an array of pairs of these parameters; for each Series there is another array. The component is designed to be used in a loop (although this is not a mandatory requirement: multiple instances of the Bar Chart component can be placed into the procedure, and called one after another in a linear sequence).

However, there is a problem: the Category parameter in the Bar Chart must be of TEXT data type, but what we want to pass to it is the order date (DATE type). So, before passing control to the Bar Chart component, we have to convert the order date (C.OrderDate field returned by the Retrieval) into a text string. We do this with the help of the DateToText System function:



Note: before dragging the Function component in, open the [Editor](#), make sure that "YYMMDD" format is listed in it, and add it to the list if necessary.

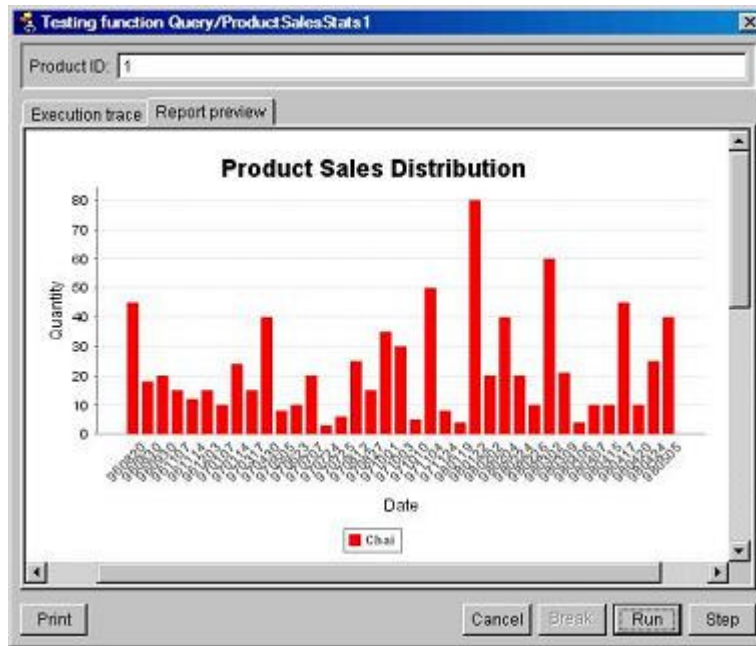
The Bar Chart component formats an internal "image" of the bar chart by collecting all data necessary for producing the chart. The Chart Handle field at the bottom of the Bar Chart component houses a variable that represents this entire bar chart image, and should be passed to the Print Template that formats the actual graphical image of the bar chart.

The Chart Handle field is set to the *New variable* that becomes var1, and is renamed to "Chart". The Print Template component is set to ProductSalesBarChart, and its Inspector is configured as follows:



Here, the Print Template's Bar Chart field ("chart1") is linked to the Report Procedure's Bar Chart handle field ("Chart"), thus providing the flow of the bar chart data into the Print Template.

To test the report, we will run it with the Product ID = 1:



Save the report as Query/ProductSalesStats1.