Showing Totals in Access Queries

New Access users, especially those who are used to keeping their data on an Excel spreadsheet, often ask how they can summarize their records. In Excel you might enter a formula at the foot of a column to sum its values, or maybe use SUMIF or DSUM to examine the data more closely. But an Access table, whilst it may look like a spreadsheet, is just for storing data. When it comes to analyzing your data you need to work with queries.

When Do You Need Query Totals?

An often-overlooked feature of the *Select Query* is its ability to calculate totals. Consider the data shown in this example (*Fig. 1*). The table holds a large number of records, each containing details of a shipment of grain. Each record includes a date (the data was entered in date order so it is already sorted by this column) a country of origin, the name of the commodity and how many tonnes comprised the shipment.

Fig. 1 Data suitable for totalling.

If you wanted, you could do a sophisticated analysis of this data with a Crosstab Query or a Pivot Table, but to begin with all you want to know is *"What is the total tonnage for each commodity?"* All the data you need is contained in just two fields, *Commodity* and *Tonnes*, and a Crosstab Query needs at least three fields to work with. You can use the Totals tool instead.

Adding Totals to a Query

The first step is to create a query selecting the fields you are interested in. In this case you would choose to see the *Commodity* and *Tonnes* fields. Running the query at this stage simply displays a list of data (*Fig. 2*).
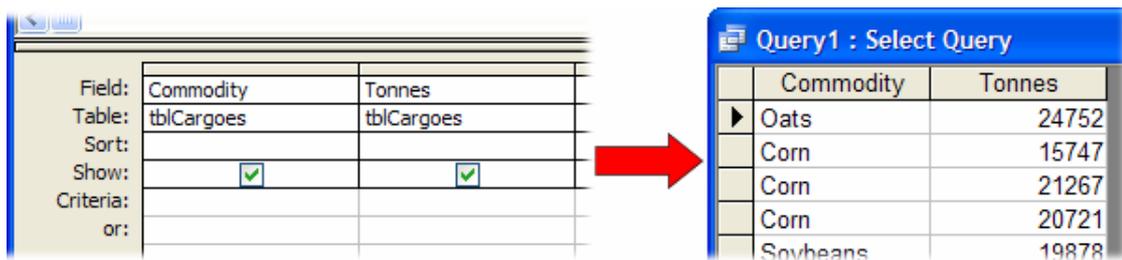

Fig. 2 A simple Select query returns only a list of data.

In the design view of the query activate the *Totals* option by clicking the **Totals** toolbar button (*Fig. 3*), by choosing **View > Totals** or by right-clicking anywhere in the query grid and choosing **Totals** from the shortcut menu.
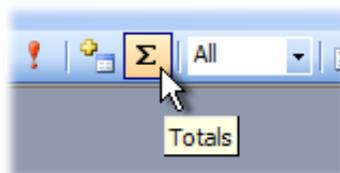

Fig. 3 The Totals tool.

Activating the Totals option creates another row labelled *Total* in the query grid (*Fig. 4*). The default setting for the Total option for each field is *Group By*, but to create a total you need to change one of these by selecting a type of calculation.
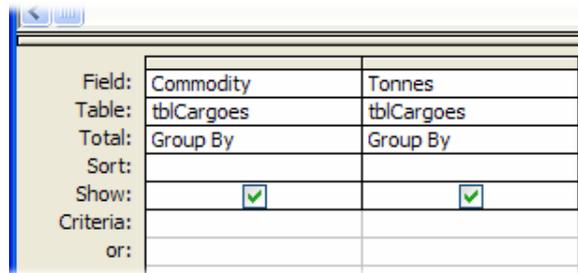

Fig. 4 The Total row is displayed.

In this example you need to ascertain the sum of the *Tonnes* field. Click in the **Total** cell of the column to be calculated (here the *Tonnes* column) then click the down-arrow to reveal a list of choices (*Fig. 5*). Make a choice from the list (here **Sum** is chosen) then run the query.
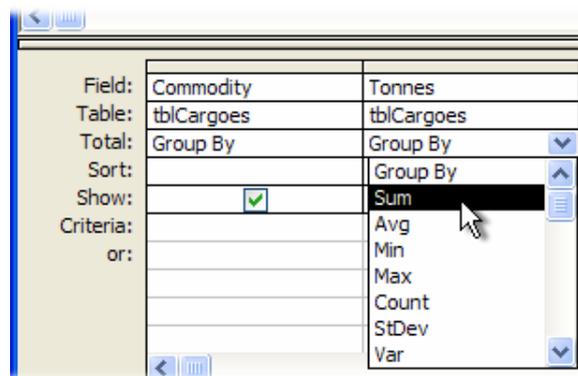

Fig. 5 Calculations offered by the Totals tool.

The result of the query shows a total tonnage for each commodity (*Fig. 6*). The *Commodity* field is automatically sorted into alphabetical order so there's no need to choose a sort option in query design. If you wish you can override the A-Z sort order and choose *Descending* for the grouped field if you want the results sorted the opposite way.

Fig. 6 The total is shown when the query is run.

Formatting the Result

The calculation will inherit the format of its source data. If you want to change that, right-click in the calculated column in the design view of the query and choose **Properties** to display the *Field Properties* dialog. Here you can choose a suitable format from the list or enter your own.

You can also use the *Caption* property to change the heading displayed at the top of the column from the one Access supplies (e.g. *SumOfTonnes*) to something more suitable (*Fig. 7*).
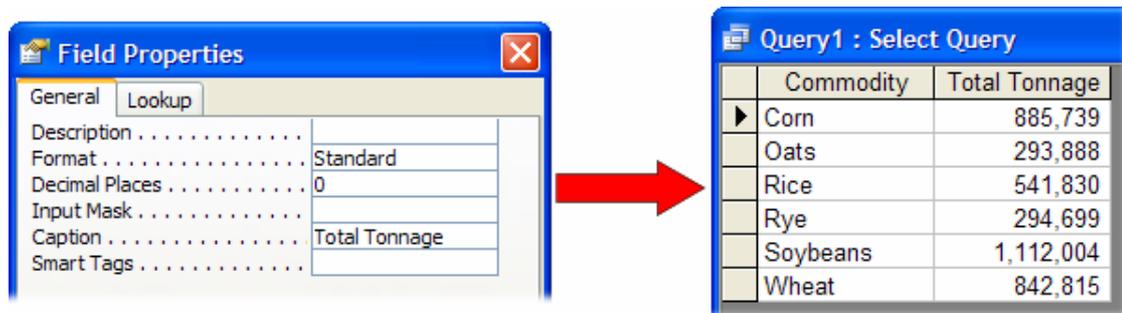
Fig. 7 Use the Field Properties to format the result.

Refining the Query

Since the Totals tool is simply an addition to the Select Query you can still add criteria to the query definition if you wish. Here, criteria have been added to the *Commodity* field to select records for specific commodities (*Fig. 8*).
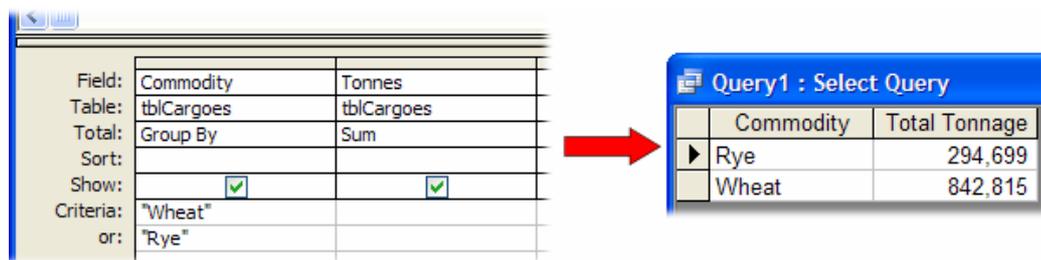

Fig. 8 Adding criteria to filter the results.

You may want to filter by a certain field but not group by that field. To do this, add the field to the grid and set the value in the *Total* row to **Where**. You can now add your criteria to this column (*Fig. 9*).
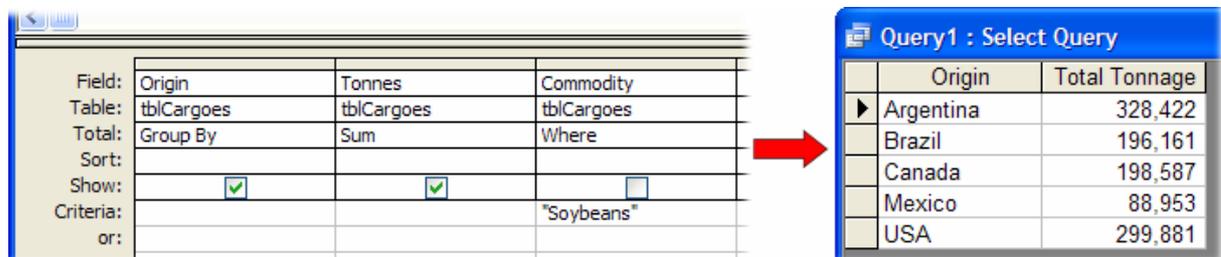

Fig. 9 Adding criteria without grouping by using Where.

When you use the *Where* option the *Show* checkbox for that column is automatically unchecked so that it does not appear in the query result.

Grouping by Several Fields

You can group my as many fields as you wish, providing the chosen fields contain suitable data.
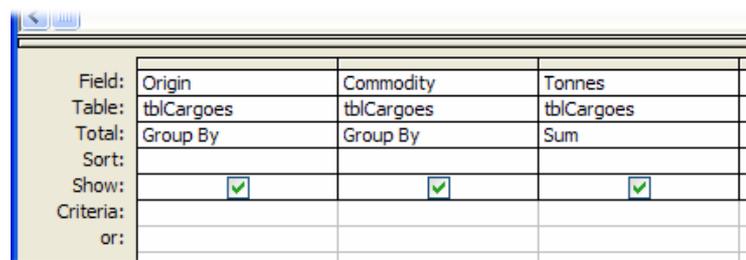

Fig. 10 Grouping by an additional field.

Here, the data has been grouped by both *Origin* and *Commodity* (*Fig. 10*). Again the data is sorted automatically (*Fig. 11*).

Fig. 11 The resulting query grouped by two fields.

Fig. 12 You can change grouping order.

When grouping by more than one field you can dictate the order in which the data is grouped by arranging the columns appropriately in design view (*Fig. 12*).

Grouping can also be applied to calculated fields. For example, when dates are included in the data you might want to group the results by month or year. Here the data is grouped by year by creating a new calculated field which uses the *Year()* function to generate the year from the *ShippingDate* field (*Fig. 13*).

Fig. 13 A calculated field creates a new Year field.

The data can now be grouped additionally by the new calculated field (*Fig. 14*).

Fig. 14 A calculated field (Year) is included in the grouping.

NOTE: If you create a calculated field that contains one or more *aggregate* functions (e.g. Sum, Count, Avg, Var etc.) the Total row for that field must be set to *Expression*.