

Crosstab Queries

A Crosstab Query is a special kind of query that summarizes data by plotting one field against one or more other fields. Crosstab Queries can handle large amounts of data with ease and are very easy to create.

Suitable Data for a Crosstab Query

A Crosstab Query requires at least three fields to work with, one of which should be either numeric (so that its values can be calculated) or suitable for counting (i.e. to count the number of records). Here's a typical example (Fig. 1):

	CargoID	ShippingDate	Origin	Commodity	Tonnes
▶	1	31/01/2000	USA	Oats	24752
	2	12/02/2000	USA	Corn	15747
	3	05/03/2000	Mexico	Corn	21267
	4	20/03/2000	Canada	Corn	20721
	5	01/04/2000	Mexico	Soybeans	19878
	6	01/04/2000	Argentina	Wheat	16319
	7	10/04/2000	Canada	Rice	18883

Fig. 1 Suitable data for a Crosstab Query.

This table holds details of grain shipments. Each record has, in addition to the primary key field *CargoID*, four fields one of which (*Tonnes*) is numeric. The other fields (*ShippingDate*, *Origin* and *Commodity*) all describe the shipment and can be used to plot the data. A typical Crosstab Query on this data might produce a result like this (Fig. 2):

	Origin	Corn	Oats	Rice	Rye	Soybeans	Wheat
▶	Argentina	272145	63552	128809	80447	328422	209677
	Brazil	168679	36271	128979	39597	196161	152493
	Canada	146778	63863	96935	57682	198587	144297
	Mexico	66761	42446	38964	35400	88953	73392
	USA	231376	87756	148143	81573	299881	262956

Fig. 2 A typical Crosstab Query.

How the Crosstab Query Works

When creating a Crosstab Query you must nominate one field as a *Column Heading* and up to three fields as *Row Headings*. You must also nominate a field whose values will be summarized. When you run the query Access looks through the data in the Column Heading field. For each unique value that it finds it creates a column and uses the value for the column heading. It also looks through the data in the Row Headings field and creates a row for each unique value that it finds, using that value as the row heading. If more than one field were nominated as row headings a row is created for each combination of unique values found.

The column headings are arranged across the top of the resulting datasheet. The row headings are arranged down the left-hand side.

In this example (Fig. 2) the *Commodity* field was set as the Column Heading. Access found six unique values in the *Commodity* field so it created six columns on the datasheet. The *Origin* field was set as the Row Heading. Access found five unique values in the *Origin* field so it created five rows on the datasheet.

The *Tonnes* field was chosen to be summarized so its values appear at the intersections of the columns and rows. The *Sum* option was chosen so the values displayed are the totals for each combination of values.

Crosstab Query Limits

You can nominate up to three fields as row headings but only one field as a column heading. The maximum number of columns that Access can display is 255 (including those used to display the row headings). It cannot execute a Crosstab Query where the number of unique values in the nominated column heading field would require it to exceed this value (Fig. 3).

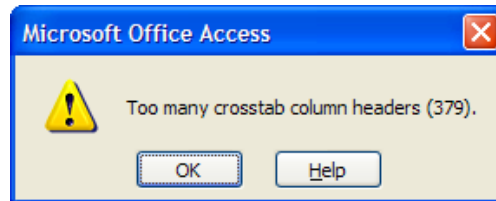


Fig. 3 Access warns that there are too many unique values.

Building a Crosstab Query

Access provides a wizard for creating Crosstab Queries. There is no need to describe it here. To use the wizard go to the **Queries** section of the Database Window click the **New** button, or choose **Insert > Query**, and choose **Crosstab Query Wizard** from the *New Query* dialog.

To create a Crosstab Query manually, go to the **Queries** section of the Database Window and double-click **Create query in Design view** to open the query design window and add the table or tables containing the data you want to summarize. Click the down-arrow next to the **Query Type** button on the query design toolbar and choose **Crosstab Query** from the menu (Fig. 4). Alternatively choose **Crosstab Query** from the **Query** menu.

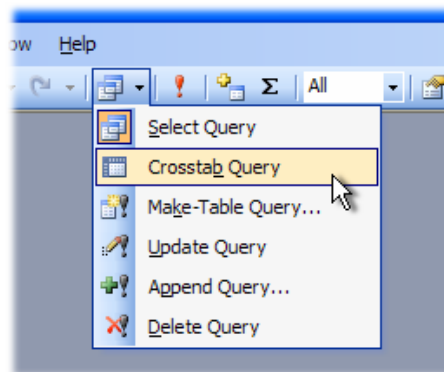


Fig. 4 Select Crosstab Query from the Query Type menu.

When you do this, two new rows appear in the query design grid labelled *Total* and *Crosstab*. Decide which field you want as a row heading and bring it into the design grid (e.g. by double-clicking its name in the field list). By default Access inserts *Group By* in the *Total* row. Click in the **Crosstab** row and choose **Row Heading** from the list (Fig. 5). Now choose a field for the column headings and add it to the grid, entering **Column Heading** in the **Crosstab** row (Fig. 6).

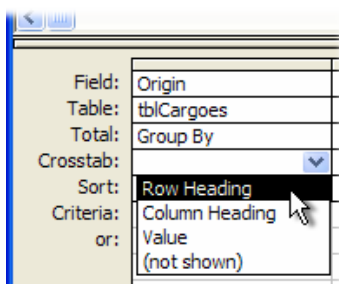


Fig. 5 Select the Row Heading.

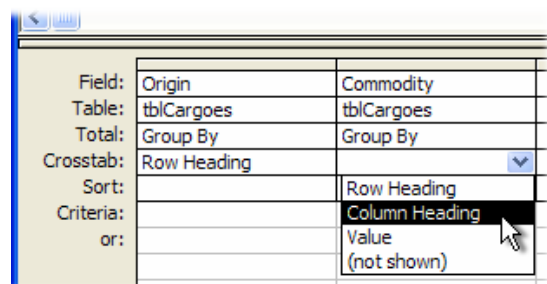


Fig. 6 Select the Column Heading.

Add the field to be calculated and change the value in the *Total* row to the calculation you want to perform (Fig. 7) and the *Crosstab* row to **Value** (Fig. 8).

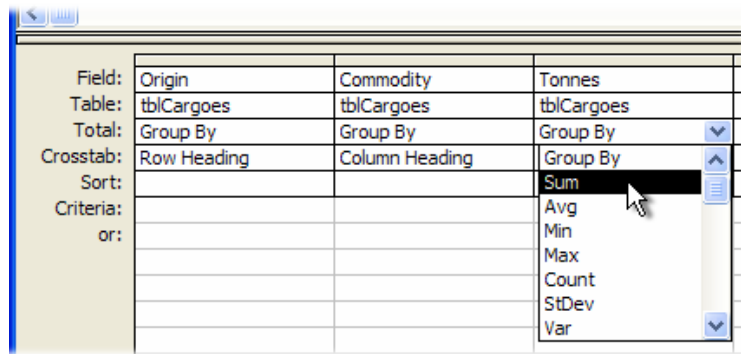


Fig. 7 Choose a type of calculation for the data.

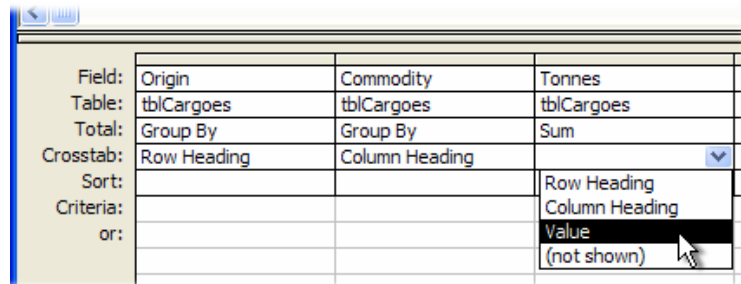


Fig. 8 The calculated field must be designated the Value.

Access now has enough information to run your Crosstab Query. The settings described here produced the result shown earlier (Fig. 2). Crosstab Queries can be refined in several ways...

Adding Criteria

As with a regular query you can filter the data by supplying criteria. For the fields designated as row or column headings, do this in the normal way by entering values or expressions in the *Criteria* row of the query design grid (Fig. 9). The *Value* field does not accept criteria.

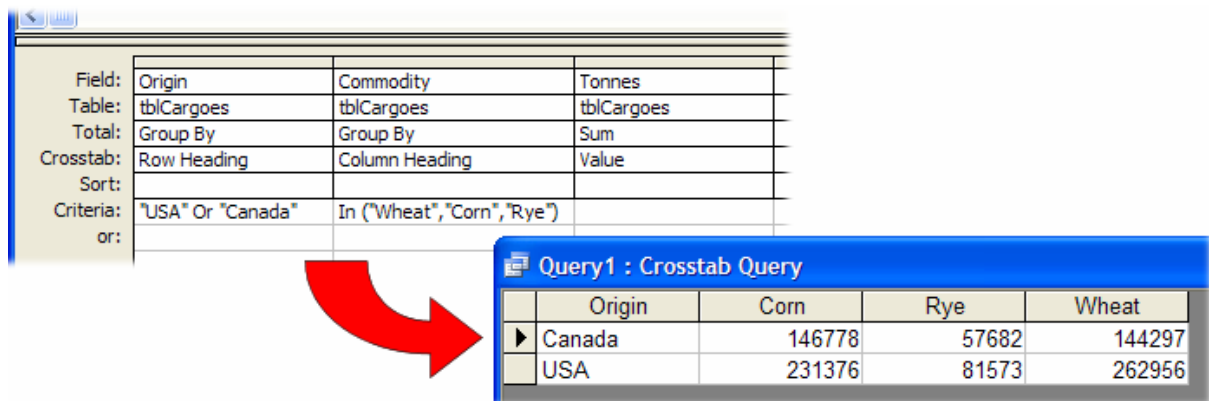


Fig. 9 Adding criteria to the displayed fields.

To filter by a field whose data you don't want to appear in the result, add the field to the query design grid, enter your criteria as usual, and set the *Total* row to **Where**. The *Crosstab* row should be left empty (to clear it you can choose **(not shown)** from the list) (Fig. 10).

tonnes	ShippingDate
Cargoes	tblCargoes
Where	Where
Criteria	
	>=#01/01/2006#

Fig. 10 Adding criteria to an additional field.

Multiple Row Headings

You can nominate up to three fields as row headings. They can be existing data or new, calculated fields as shown in this example (Fig. 11) in which a *Year* field has been created by extracting the year from the *ShippingDate* field.

Field:	Origin	Year: Year([ShippingDate])	Commodity	Tonnes
Table:	tblCargoes		tblCargoes	tblCargoes
Total:	Group By	Group By	Group By	Sum
Crosstab:	Row Heading	Row Heading	Column Heading	Value
Sort:				
Criteria:				
or:				

Fig. 11 A second Row Heading field has been added.

The relative positions left to right of the row heading fields in the query design grid control the arrangement of the data on the resulting datasheet (Fig. 12).

Origin	Year	Corn	Oats	Rice
Argentina	2000			
Argentina	2001	88069	23331	17174
Argentina	2002	63528	40221	20917
Argentina	2003	15358		37748
Argentina	2004			
Argentina	2005			
Argentina	2006			
Brazil	2000			
Brazil	2001			
Brazil	2002			
Brazil	2003			
Brazil	2004			
Brazil	2005			
Brazil	2006			
Canada	2000			
Canada	2001			
Canada	2002			
Canada	2003			
Canada	2004			
Canada	2005			
Canada	2006			

Year	Origin	Corn	Oats	Rice
2000	Argentina			
2000	Brazil	17582		
2000	Canada	20721		18883
2000	Mexico	21267		38964
2000	USA	71274	24752	15817
2001	Argentina	88069	23331	17174
2001	Brazil		15573	24743
2001	Canada	18204		15787
2001	Mexico	20740		
2001	USA		39636	38375
2002	Argentina	63528	40221	20917
2002	Brazil	23453		42996
2002	Canada	24882	23439	20668
2002	Mexico			
2002	USA			20373
2003	Argentina	15358		37748
2003	Brazil	15573		24743

Fig. 12 The data is sorted according to the positions of the column heading fields.

As you would expect, Access creates columns only for the data it finds in the column heading field. It also omits rows for which there is no data. This is more apparent when working with more than one row heading field, especially when filtering the data. In the next example the *Commodity* field has been filtered to display just one of the commodities. Since this commodity has only a small number of records fewer rows are displayed (Fig. 13):

	Year	Origin	Rye
▶	2000	Argentina	23474
	2000	Canada	16427
	2002	Brazil	23393
	2002	Mexico	19599
	2003	Canada	23908
	2003	Mexico	15801
	2004	Argentina	40879
	2004	Brazil	16204
	2004	USA	62836
	2005	Argentina	16094
	2005	USA	18737
	2006	Canada	17347

Fig. 13 Rows for which there is no data are omitted.

Sorting the Columns and Rows

By default Access sorts the row headings in ascending order (A-Z). If you prefer you can have the query sort these fields in descending order in the normal way by entering **Descending** in the *Sort* row of the query design grid. Since the columns are created from the data itself the same option does not apply. Access will sort the columns in ascending order from left to right across the datasheet unless you specify otherwise. This is unlikely to be a problem unless, for example, you are working with items which have a recognised but non-alphanumeric order such as day or month names.

Here the *ShippingDate* field, used as the column heading, has been formatted to show the name of the month (Fig. 14):

Field:	Month: Format([ShippingDate], "mmm")	Origin	Commodity
Table:		tblCargoes	tblCommodities
Total:	Group By	Group By	Group By
Crosstab:	Column Heading	Row Heading	Row Heading
Sort:			
Criteria:			
or:			

Fig. 14 The *ShippingDate* field is formatted to display month names.

The resulting datasheet sorts the months alphabetically which is not what you would normally require (Fig. 15):

	Origin	Commodity	April	August	December	February	January
▶	Argentina	Corn	68904				23176
	Argentina	Oats				20726	19495
	Argentina	Rice	17970	17213	20106		
	Argentina	Rye					
	Argentina	Soybeans	31232	19071	36508		23609

Fig. 15 The column headings are normally sorted alphabetically.

Fortunately Access allows you to specify the order in which you want the column heading items to appear. In design view right-click in the upper part of the query design window and choose **Properties**. In the *Query Properties* dialog you will find a textbox labelled *Column Headings*. Enter a list of column heading items in the order in which you want them to appear. Enclose each item in quotes and separate the items with commas (Fig. 16). When you run the query the columns will be displayed in the order you specified.

Be careful to spell each item correctly and to include all the columns you want to see. Access will only display columns for those items included in your list. If your list includes an item which does not occur in the data the column will still be displayed, but will contain no data (this is useful if you want your query always to display the same set of columns regardless of the data returned).

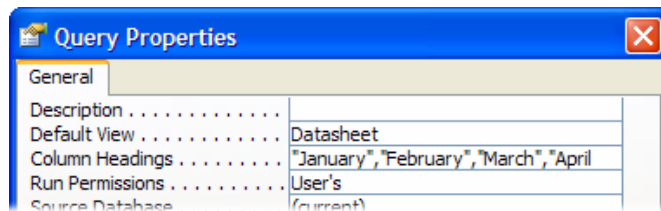


Fig. 16 Define a custom sort order for the column headings.

In this example a list of months in the correct order has been entered into the *Column Headings* property of the *Query Properties* dialog (Fig. 16) ensuring that the resulting datasheet displays the month columns in the required order (Fig. 17):

Origin	Commodity	January	February	March	April	May
Argentina	Corn	23176		38226	68904	37751
Argentina	Oats	19495	20726			
Argentina	Rice				17970	19778
Argentina	Rye			21009		23474
Argentina	Soybeans	23609		46495	31232	22264

Fig. 17 The query displays the month columns in the correct order.

Adding a Totals Column

In addition to the columns that the query creates you can add one or more calculated fields summarising the values shown in each row. For example you might want to add a column showing a grand total of the value from all the columns in each row.

To do this enter an expression in the *Field* row of a new column in the query design grid. The expression should include a name for your new field (if you omit this Access will create one) followed by a colon (:) and the name of *Value* field (for example: **Total Tonnes: Tonnes**). In the *Total* row choose a calculation from the list (choose **Sum** to see a total of the values) and in the *Crosstab* row enter **Row Heading**. You are probably thinking that this should be a column heading but, as I mentioned earlier, you are only allowed one of them. Don't worry, Access will understand what you mean.

Field:	Origin	Total Tonnes: Tonnes	Average Tonnage: Tonnes	Commodity	Tonnes
Table:	tblCargoes	tblCargoes	tblCargoes	tblCargoes	tblCargoes
Total:		Sum	Avg	Group By	Sum
Crosstab:	Row Heading	Row Heading	Row Heading	Column Heading	Value
Sort:					
Criteria:					
or:					

Fig. 18 Add a calculated field as a Row Heading to create a summary column.

In this example two summary columns have been created (Fig. 18), one using *Sum* to create a total for each row and another to show averages (Fig. 19).

Origin	Total Tonnes	Average Tonnage	Corn	Oats	Rice
Argentina	1,083,052	20,057	272145	63552	1288
Brazil	722,180	20,061	168679	36271	1289
Canada	708,142	19,671	146778	63863	969
Mexico	345,916	19,218	66761	42446	389
USA	1,111,685	19,852	231376	87756	1481

Fig. 19 Summary columns displayed on the datasheet.

More Ideas for Crosstab Queries

Now that you have built your crosstab query you will probably want to do something with the result. Here are some ideas.

Take a Snapshot of the Data

The Crosstab Query has essentially created an entirely new dataset with new fields that were not present in the original data. You might find it useful to preserve this dataset for other purposes. To do this close and save the Crosstab Query then create a new query based on it. Add all the fields to the query design grid then change your new query to a *Make Table* query (**Query > Make Table Query**). When you run the new query it will create a new table from the data with each column of data in the Crosstab Query becoming a field in the table (Fig. 20).

	Origin	Corn	Oats	Rice	Rye
▶	Argentina	272145	63552	128809	80447
	Brazil	168679	36271	128979	39597
	Canada	146778	63863	96935	57682
	Mexico	66761	42446	38964	35400
	USA	231376	87756	148143	81573
*					

	Field Name	Data Type
▶	Origin	Text
	Corn	Number
	Oats	Number
	Rice	Number
	Rye	Number
	Soybeans	Number
	Wheat	Number

Fig. 20 A table, in datasheet and design view, built from a crosstab query.

Export the Data to Excel

Data is often stored in an Access database because of the large volumes it can handle, and the speed with which it can create a summary with tools like the Crosstab Query. But you might prefer to work with your summarized data in another program such as Excel. Exporting the data is very easy. Open the Crosstab Query and choose **Tools > Office Links > Analyze It with Microsoft Office Excel**. This handy tool exports the recordset as an Excel workbook (Fig. 21).

	A	B	C	D	E	F	G	H
1	Origin	Corn	Oats	Rice	Rye	Soybeans	Wheat	
2	Argentina	272145	63552	128809	80447	328422	209677	
3	Brazil	168679	36271	128979	39597	196161	152493	
4	Canada	146778	63863	96935	57682	198587	144297	
5	Mexico	66761	42446	38964	35400	88953	73392	
6	USA	231376	87756	148143	81573	299881	262956	
7								

Fig. 21 The result of a Crosstab Query exported to Excel.

The workbook will already have been saved, using the query's name for the file and worksheet names, into the *Default database folder*, usually *My Documents*. You can change the default save location from **Tools > Options > General** in Access.