

Sales & Stock Database

Dales Farm Foods Association

A DBMS designed to bring together individual farmers produce to sell through one central shop.

Work through this example and learn the skills so you can design and build a database system for your client.

See contents list at back of booklet.



Section 9

Action queries

Most of the time you use queries to search, group and sort your data. There will be times when you need to carry out an action such as **making a new table**, **appending (updating) data** to it and **deleting old records**.

- One problem that students new to access have is that they rush in and produce these types of query and realise that they are quite destructive – **they can't be reversed**.
- You should always try them out on a copy of the database first and only do it for real when you have tested it out and found that it works exactly as you want. **Better to be safe than sorry!**



There are many ways you can use action queries. I can't show every single use of them. Here, I will show you what they can do – it is up to you to think about how you can use these tools in your database.

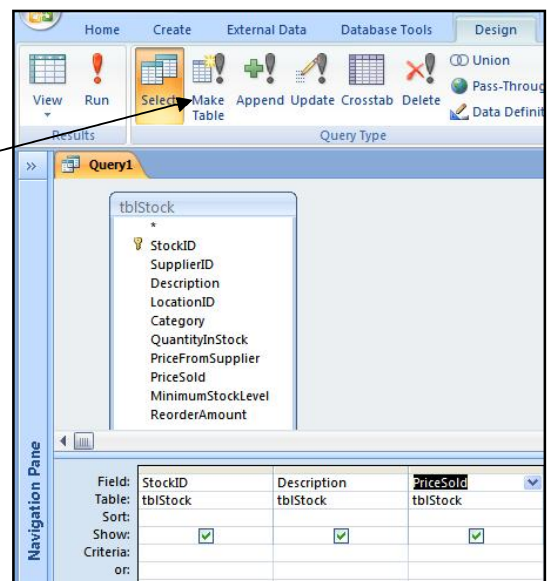
Make table queries

These are used to make a new table from an existing table.

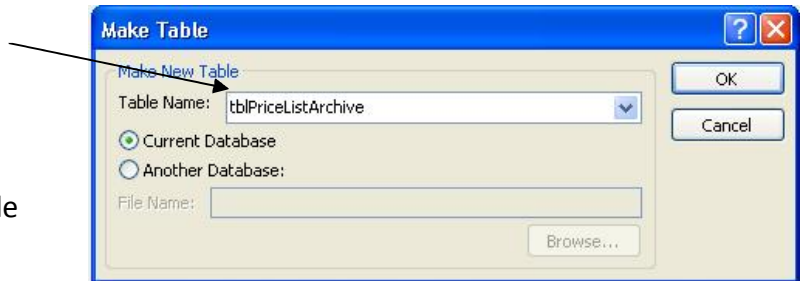
- Suppose you have a stock list that changes regularly. You may want to archive the old stock and prices in a copy table so the main table is always up to date.
- Maybe a customer list is archived – all customers who haven't purchased something for twelve months are moved into the archive table and can be moved back if they start purchasing again.
- Archive tables can be kept for a set time and then deleted. Think of it as an interim table. You don't want the old data to clog up your main table, but you don't want to delete them yet – so archive them!
- This keeps the real (active) table smaller and quicker to search.

Producing a price list archive

95. Open a new query and add the **stock** table.
96. Add the **StockID**, **Description** and **PriceSold** fields.
97. Click on the **Make Table** button

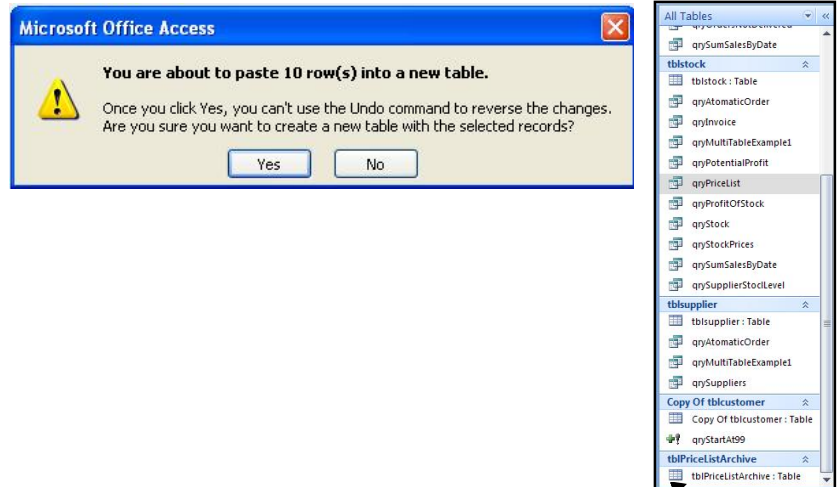


98. Type in the name for the new table **tblPriceListArchive**.



99. Run the query to carry out the make table command.

100. It will give you the warning that you are about to paste some data into the new table – and you can't reverse this.

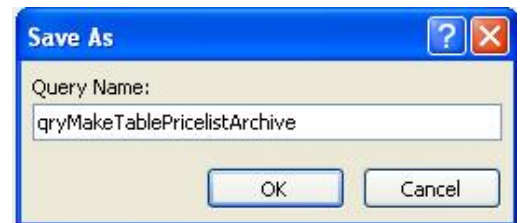


Click yes.

101. It has added the new table at the bottom of the list in the navigation pane.

102. Save this query as **qryMakeTablePriceListArchive**.

➤ When it adds it to the navigation pane – look at the icon it shows to let you know it is a make table query.



Update queries

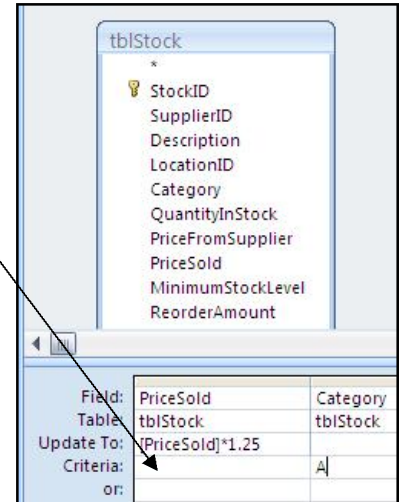
An update can be used to update certain sets of data such as making a change to a price or changing the location of a product.

- You may have customers who have purchased or being given bonus points that can be spent on goods. Their current points need adding to as they purchase more or reducing as they spend them. You can use update queries to manage this field containing their points.
- A shop may need to keep a running total of what a customer owes. Use an update query to keep that total up to date as the customer purchases more goods and pays money off at the end of the month.
- There are many examples of data in fields that needs updating regularly – too many for me to show here – so this is how it works – and you decide how you can use them!

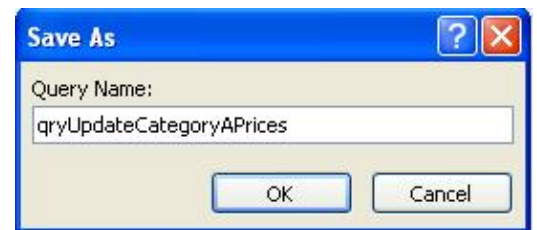
In this task I want to increase the price of **category 'A'** goods by **25%**.

103. Open a new query and add the **stock** table.

- 104. Insert the **PriceSold** and the **Category** fields
- 105. Click the **Update button** and it will add a new row to the grid called **Update to:**
- 106. Type **[PriceSold]*1.25** in the Update to field – under **PriceSold**.
- 107. Type **A** in the criteria field under Category



- 108. Save the query as **qryUpdateCategoryAPrices**. Don't run it yet!



- I have opened up my stock table to check what the original prices are.
- They are £0.25 and £2.00

Category	Price sold	Quant
B	£1.00	
A	£0.25	
B	£0.15	
C	£0.35	
D	£1.35	
D	£3.00	
E	£1.50	
C	£1.75	
B	£1.95	
A	£2.00	

- 109. **Don't run the query yet** – but click on the view button and select **Datasheet View**. Mine has shown me the two records that match the criteria I typed in.

PriceSold
£0.25
£2.00

- It won't work until you run the query – as with all action queries.

Before I ran the query I decided to go and check that the **tblPriceListArchive** has the old prices in it.

I then realised that it did, but it didn't have the category field in it so I could easily check the two fields that are going to be updated.

110. You now need to open up the **qryMakeTablePriceListArchive** in design view and add the category field.

Stock ID	Description	Category	Price sold
1	Bread	B	£1.00
2	Eggs	A	£0.25
3	Scones	B	£0.15
4	Sausage	C	£0.35
5	Pork pie (small)	D	£1.35
6	Pork pie (large)	D	£3.00
7	Honey	E	£1.50
8	Mince Steak	C	£1.75
9	Biscuits	B	£1.95
10	Cream	A	£2.00
*	(New)		

111. **Run this query again.** It will tell you it is deleting the old table you made a few minutes ago. Accept this. It replaces this with the new version.

112. Run the new update query - **qryUpdateCategoryAPrices**. It will give you a warning about updating rows and you not being able to reverse it.

➤ Open **tblStock** to see if it has added 25% on to category A prices.

Location ID	Category	Price sold
B	B	£1.00
A	A	£0.31
B	B	£0.15
C	C	£0.35
D	D	£1.35
D	D	£3.00
E	E	£1.50
C	C	£1.75
B	B	£1.95
A	A	£2.50



Be aware – every time you run this query it adds that increase on – if you run the query again because you can't see the change straight away and you think nothing happened - it adds another 25% on and when you eventually check **tblStock** – you are annoyed with yourself for doing this – **and you can't undo!**

Section 10

Introducing SQL

So far, we have set up queries using the design grid. There is another way. It is called SQL which stands for **Structured Query Language** – pronounced sequel.

With this method you can type in the commands you wish to carry out in an English written way.

It is awkward to understand when you come across it for the first time – but it is something that you can learn with a little effort and you can produce some complex query criteria quite easily. It is worth trying to learn.

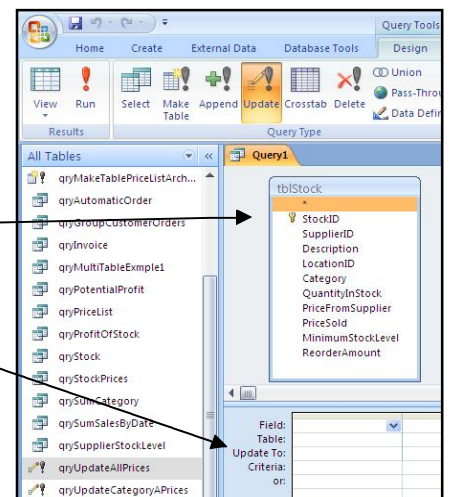
Updating all prices using SQL

As with all businesses there are times when the owner needs to update all of their prices at one go.

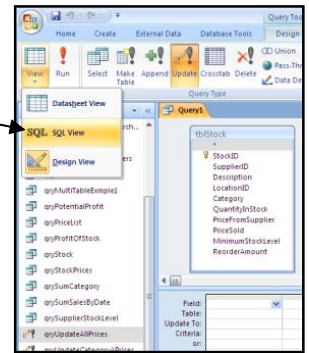
- ✓ We want to set up an update query that does this – but asks the user how much.
- ✓ It will have a parameter box built in that asks for the percentage amount.

113. Open a new query and add **tblStock**.

114. Click the **Update button** – and it will add a new row into the grid.



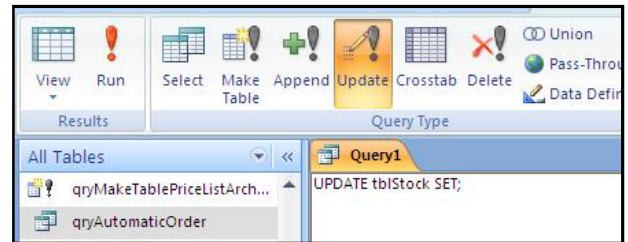
115. Click on the View button and select **SQL**.



116. It automatically starts the SQL statement.

117. You now need to complete this statement.

- It tells it to update **tblStock**.
- It tells it to **set** tblStock.PriceSold field to the **following**;
- PriceSold multiplied **by 1 and a percentage** you type in.
- The percentage needs to be typed in as 0.25 for 25%.
- If you typed in 25 it would be 1 x 25 – so the price goes up 25 fold.



UPDATE tblStock SET tblStock.PriceSold = [PriceSold]*(1+[How much?]);

- Make sure you get the square and round brackets right!



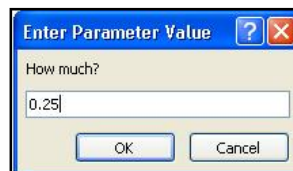
118. Let's have a quick check of the prices before we run this query.

119. **Run the query** and it will run the parameter.

120. Type in **0.25**.

121. **Click OK.**

122. It warns you that it is about to update a table with a number of records. **Click OK.**



PriceFromSupplier	PriceSold
£0.50	£1.00
£0.15	£0.31
£0.10	£0.15
£0.25	£0.35
£1.00	£1.35
£2.00	£3.00
£1.00	£1.50
£1.25	£1.75
£1.40	£1.95
£1.30	£2.50