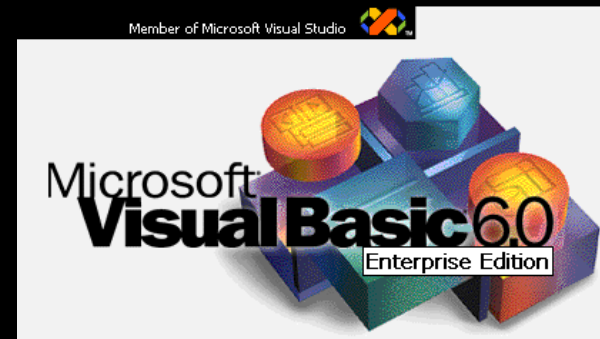
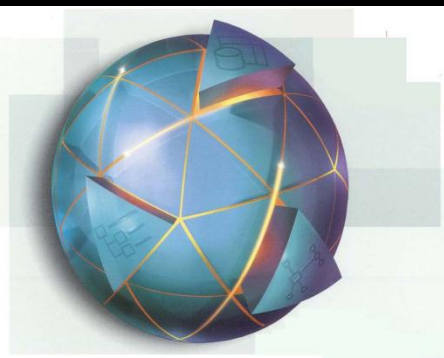


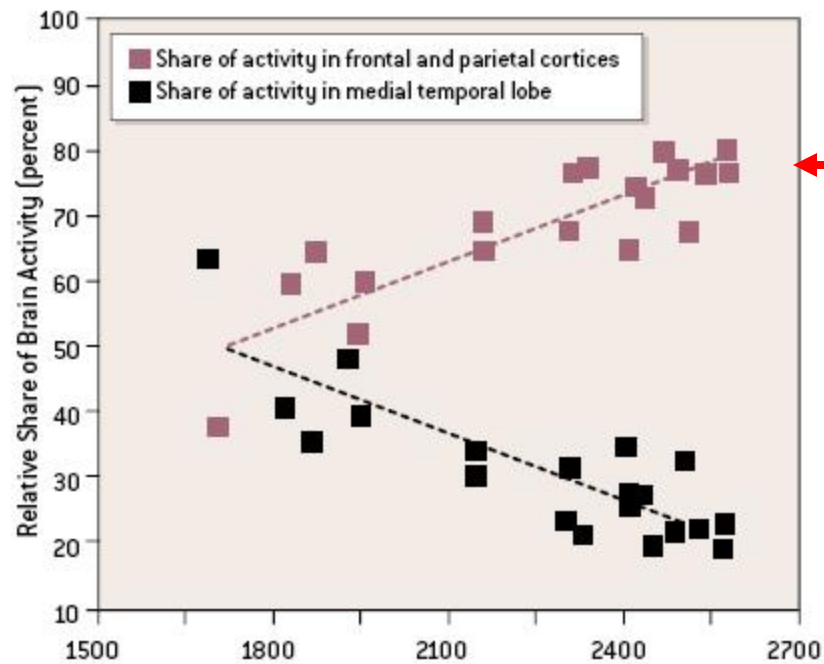
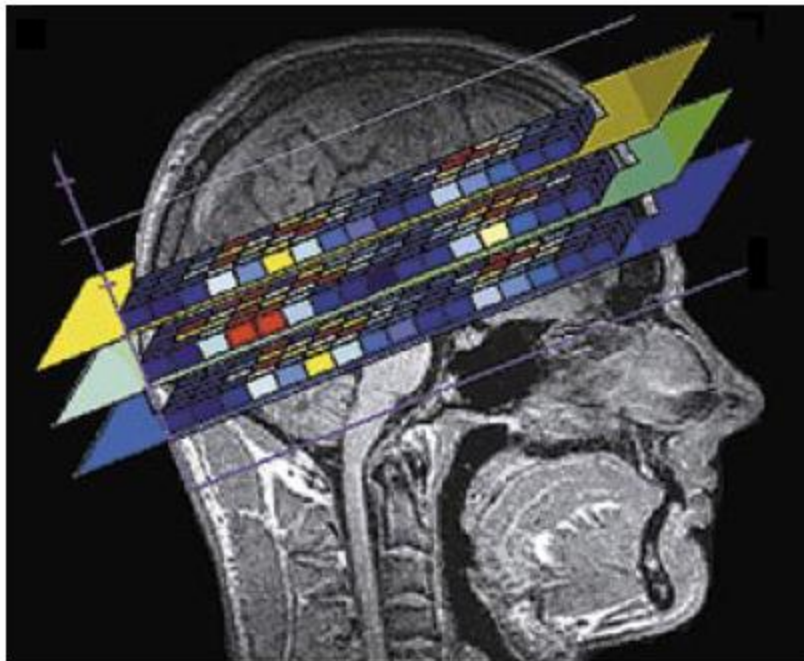
# MS Access Queries

By

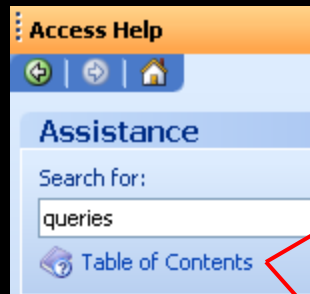
Barry Hynum, Ph.D.

MaxQual Consulting, Inc.





# TOC



## Table of Contents

- Startup and Settings
- Printing
- Creating and Working with Databases and Objects
  - Working with Access Files
  - Converting Access Databases
- Database Objects
  - Tables
  - Forms
  - Reports
  - Queries
    - Opening, Viewing, and Running Queries
    - Creating Queries
    - Customizing Queries
    - Using Multiple Tables in Queries
    - Using Criteria and Expressions to Retrieve Data
    - Performing Calculations
    - SQL Queries
    - ANSI SQL Query Mode
    - Microsoft Jet SQL Reference
    - Queries I: Get answers with queries
- Data Access Pages
- Working with Database Objects

# About designing a query (MDB)

**Note** The information in this topic applies only to a Microsoft Access database (.mdb).

When you open a query in [Design view](#), or open a form, report, or datasheet and show the [Advanced Filter/Sort window](#), you see the [design grid](#), which you can use to make a variety of changes to get the query results you want.

Field:	LastName	OrderDate	Subtotal
Table:	Employees	Orders	Order Subtotals
Total:	Group By	Where	Sum
Sort:			Descending
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	Between #6/1		

Rename

- Select Query
- Crosstab Query
- Make-Table Query...
- Update Query
- Append Query...
- Delete Query

5  
25  
100  
5%  
25%  
All

Tables Queries Both

- Categories
- Customers
- Employees
- Order Details
- Orders
- Products
- Shippers
- Suppliers
- zdtbl\_MyCommandBarControls
- zdtblSWEJournal

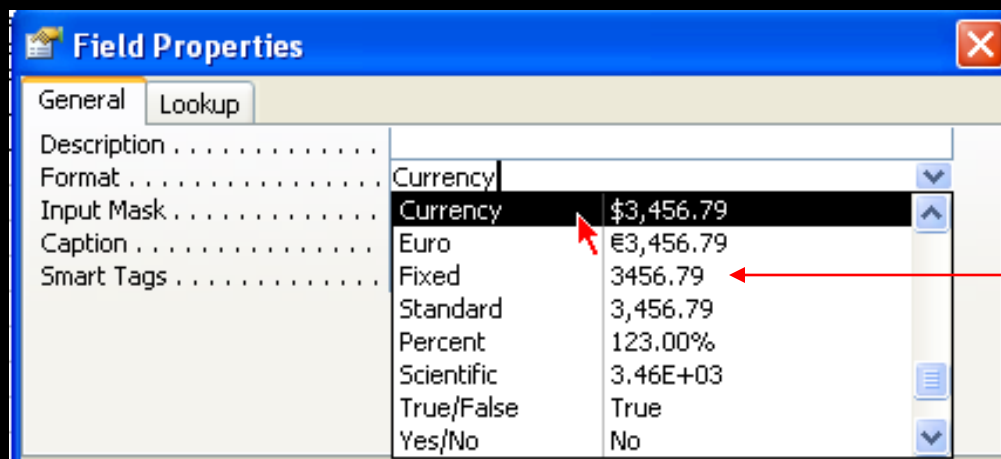
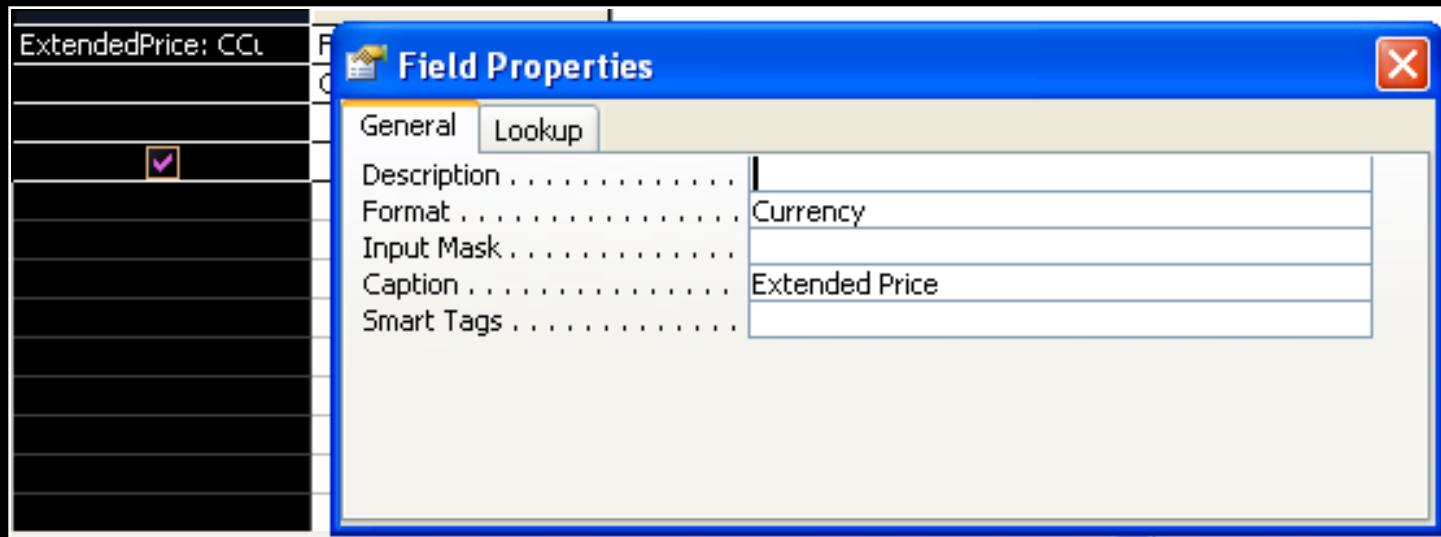
Zoom

Between #6/1/01# And #6/15/01#

- 1 Add or remove tables, queries, and fields
- 2 Calculate amounts
- 3 Limit results using criteria
- 4 Sort records

# Formatting Fields

Rt. Click Column



# Joins

**Options** [?] [X]

View | General | Edit/Find | Keyboard | Datasheet | Forms/Reports | Pages

Advanced | International | Error Checking | Spelling | **Tables/Queries**

**Table design**

**Default field sizes**

Text:

Number:

**Default field type:**

**AutoIndex on Import/Create:**

Show Property Update Options buttons

**Query design**

Show table names

Output all fields

Enable AutoJoin

**Run permissions**

Owner's

User's


**Query design font**

Font:  Size:

**SQL Server Compatible Syntax (ANSI 92)**

This database

Default for new databases



# Joins: Different Tables

Query1 : Select Query

Customers

- \* CustomerID
- CompanyNam
- ContactName
- ContactTitle

Orders

- \* OrderID
- CustomerID
- EmployeeID
- OrderDate

Join Properties

Left Table Name: Customers  
Right Table Name: Orders  
Left Column Name: CustomerID  
Right Column Name: CustomerID

1: Only include rows where the joined fields from both tables are equal.  
 2: Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.  
 3: Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.

OK Cancel New

Query1 : Select Query

Customers

- \* CustomerID
- CompanyNam
- ContactName
- ContactTitle

Orders

- \* OrderID
- CustomerID
- EmployeeID
- OrderDate

Join Properties

Left Table Name: Customers  
Right Table Name: Orders  
Left Column Name: CustomerID  
Right Column Name: CustomerID

1: Only include rows where the joined fields from both tables are equal.  
 2: Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.  
 3: Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.

OK Cancel New

# Joins: Different Tables

Query1 : Select Query

**Customers**

- \* CustomerID
- CompanyNam
- ContactName
- ContactTitle

**Orders**

- \* OrderID
- CustomerID
- EmployeeID
- OrderDate

**Join Properties**

Left Table Name: Customers  
Right Table Name: Orders  
Left Column Name: CustomerID  
Right Column Name: CustomerID

1: Only include rows where the joined fields from both tables are equal.  
 2: Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.  
 3: Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.

OK Cancel New

Query1 : Select Query

**Customers**

- \* CustomerID
- CompanyNam
- ContactName
- ContactTitle

**Orders**

- \* OrderID
- CustomerID
- EmployeeID
- OrderDate

**Join Properties**

Left Table Name: Customers  
Right Table Name: Orders  
Left Column Name: CustomerID  
Right Column Name: CustomerID

1: Only include rows where the joined fields from both tables are equal.  
 2: Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.  
 3: Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.

OK Cancel New



# Joins: Same Table

Query1 : Select Query

Employees\_1

- \* EmployeeID
- LastName
- FirstName
- Title
- TitleOfCourtesy
- BirthDate
- HireDate
- Address
- City
- Region
- PostalCode
- Country
- HomePhone
- Extension
- Photo
- Notes
- ReportsTo

Employees

- \* EmployeeID
- LastName
- FirstName
- Title
- TitleOfCourtesy
- BirthDate
- HireDate
- Address
- City
- Region
- PostalCode
- Country
- HomePhone
- Extension
- Photo
- Notes
- ReportsTo

Field:	EmployeeID	LastName	FirstName	Title	Mgr: [Employees].[LastName] & ", " & [Employees].[FirstName]
Table:	Employees_1	Employees_1	Employees_1	Employees	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

# Criteria

Field:	ContactName	Country	ContactTitle
Table:	Customers	Customer	Customers
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		"France"	"owner"
or:			

This query will find all the customers who live in France **And** own their own store.

Field:	CompanyName	Country	City
Table:	Customers	Customers	Customers
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		"France"	
or:			"London"

This query will find all the customers who work in France **Or** London.

Field:	Country		
Table:	Customers		
Sort:			
Show:	<input checked="" type="checkbox"/>		
Criteria:	"France"		
or:	"Belgium"		
	"Germany"		
	"Denmark"		

This query will find all the customers who live in France, Belgium, Germany, **Or** Denmark.

Field:	Country		
Table:	Customers		
Sort:			
Show:	<input checked="" type="checkbox"/>		
Criteria:	In ("France", "Belgium", "Germany", "Denmark")		
or:			

This query will find customers who work in France, Belgium, Germany, or Denmark.

# Criteria: Between

Field:	ProductName	UnitPrice
Table:	Products	Products
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Between 25 And 40

Product Name	Unit Price
Grandma's Boysenberry Spread	\$25.00
Uncle Bob's Organic Dried Pears	\$30.00
Northwoods Cranberry Sauce	\$40.00
Ikura	\$31.00
Queso Manchego La Pastora	\$38.00
Alice Mutton	\$39.00

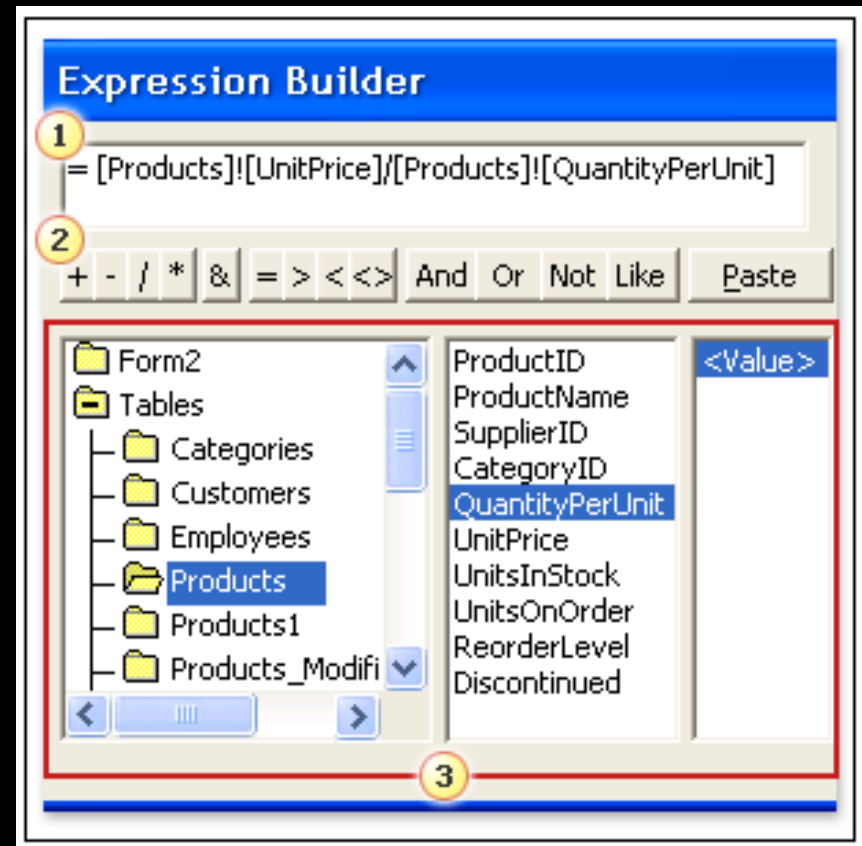
# Criteria: Wildcards

## Wildcards based on the Jet database engine (ANSI-89)

Symbol	Description	Example
*	Matches any number of characters. It can be used as the first or last character in the character string.	wh* finds what, white, and why
?	Matches any single alphabetic character.	B?ll finds ball, bell, and bill
[ ]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill
!	Matches any character not in the brackets.	b[!ae]ll finds bill and bull but not ball or bell
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd
#	Matches any single numeric character.	1#3 finds 103, 113, and 123

# Expressions

Type	Example	Description
Text	"administrator"	Finds all employees whose title is "administrator"
Number	03	Finds all courses whose section number is 3
Date	#04/03/06#	Finds all dates that exactly match 04/03/06
Expression with a comparison operator	<Now()	Uses the date function called <b>Now()</b> to retrieve all dates before today
Expression with a calculation	([UnitsReceived]) < ([UnitsOrdered]-3)	Uses a calculation in criteria



# Expressions: Dates & Nulls

Expression	Purpose
PickUpTime: DateAdd("h", 3, [ArrivalTime])	Displays a time that is three hours after ArrivalTime.
Age: DateDiff("yyyy", [Birthdate], Now())+ Int ( Format(now(), "mmdd") < Format( [Birthdate], "mmdd") )	Calculates someone's age from their birthday, relative to current date.
LagTime: DateDiff("d", [OrderDate], [ShippedDate])	Displays the number of days between the OrderDate and ShippedDate fields.
YearHired: DatePart("yyyy", [HireDate])	Displays the year each employee was hired.
MonthNo: DatePart("M", [OrderDate])	Displays the number of the month, as in 8 for August.
ThisMonth: Format(Now(), "mmm")	Displays the abbreviated month that the current date represents, where mmm is Jan through Dec.

Expression	Purpose
LeadTime: Iif(IsNull ([RequiredDate] - [ShippedDate]), "Check for a missing date", [RequiredDate] - [ShippedDate])	Displays in the LeadTime field the message "Check for a missing date" if the value of either the RequiredDate or ShippedDate fields is null; otherwise, it displays the difference.
CurrentCountry: Iif (IsNull([Country]), "", [Country])	Displays in the CurrentCountry field an empty string if the Country field is null; otherwise, it displays the contents of the Country field.
=Iif(IsNull([Region]), [City]&" "& [PostalCode], [City]&" "&[Region]&" "& [PostalCode])	Displays the values of the City and PostalCode fields if Region is null; otherwise, it displays the values of the City, Region, and PostalCode fields.

# Aggregate Functions

## ▼ Aggregate functions

The following options in the query design grid's **Total** row are [aggregate functions](#):

Select	To find the	Use with these field data types
<b>Sum</b>	Total of the values in a field.	Number, Date/Time, Currency, and AutoNumber
<b>Avg</b>	Average of the values in a field.	Number, Date/Time, Currency, and AutoNumber
<b>Min</b>	Lowest value in a field.	Text, Number, Date/Time, Currency, and AutoNumber
<b>Max</b>	Highest value in a field.	Text, Number, Date/Time, Currency, and AutoNumber
<b>Count</b>	Number of values in a field, not counting <b>Null</b> (blank) values.	Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, and OLE Object
<b>StDev</b>	Standard deviation of the values in a field.	Number, Date/Time, Currency, and AutoNumber
<b>Var</b>	Variance of the values in a field.	Number, Date/Time, Currency, and AutoNumber

## ▼ First and Last functions

You use the **First** or **Last** functions to return the first or last record in the group you are performing calculations on. These functions return the first or last record as it was entered in chronological order. Sorting the records has no effect on these functions.

## ▼ [Group By, Expression, and Where options](#)

The following table summarizes what each option does:

Select	To
<b>Group By</b>	Define the groups you want to perform the calculations for. For example, to show total sales by category, select <b>Group By</b> for the CategoryName field.
<b>Expression</b>	Create a calculated field that includes an <a href="#">aggregate function</a> in its <a href="#">expression</a> . Usually, you create a calculated field when you want to use multiple functions in an expression.
<b>Where</b>	Specify criteria for a field you aren't using to define groupings. If you select this option for a field, Access will hide the field in the query results by clearing the <b>Show</b> check box.

# Aggregates: Criteria

Before

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum
Sort:			
Show:			
Criteria:	"Canada"		<10000
or:	"UK"		<10000

After

Ship Country	Company Name	SumOf	ExtendedPrice
Canada	Bottom-Dollar Markets		\$28,025.51
Canada	Laughing Bacchus		\$522.50
Canada	Mère Paillarde		\$37,123.65
UK	Around the Horn		\$14,602.15
UK	B's Beverages		\$7,383.90

Ship Country	Company Name	SumOf	ExtendedPrice
Canada	Laughing Bacchus		\$522.50
UK	B's Beverages		\$7,383.90

- 1 This query totals extended prices for companies in Canada and the UK ...
- 2 ... but shows only those that are less than \$10,000.



# Aggregates: Where

Ship Country	Company Name	ExtendedPrice
UK	Seven Seas Imports	\$345.00
Germany	Lehmanns Marktstand	\$525.00
UK	Seven Seas Imports	\$210.00
Canada	Mère Paillarde	\$631.20

1

Field:	ShipCountry	Company Name	ExtendedPrice	ExtendedPrice
Table:				
Total:	Group By	Group By	Sum	Where
Sort:				
Show:				
Criteria:	"Canada"			<500
or:	"UK"			

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$11,938.00
Canada	Mère Paillarde	\$21,631.36
UK	Around the Horn	\$4,907.50
UK	B's Beverages	\$1,380.00
UK	Eastern Connection	\$11,463.00
UK	Seven Seas Imports	\$3788.60

3

Oh, Microsoft!?

1 From these records, the query retrieves only those with extended prices greater than \$500.00 before it groups or totals ...

2 ... and then it sums and shows only those totals for companies in Canada or the UK.

3 This total for Seven Seas Imports does not include the order for \$210.00.

# Union Queries

```
Query1: Union Query
SELECT [CompanyName],[City]
FROM [Suppliers]
WHERE [Country]="Brazil"
UNION
SELECT [CompanyName],[City]
FROM [Customer]
WHERE [Country]="Brazil";
```

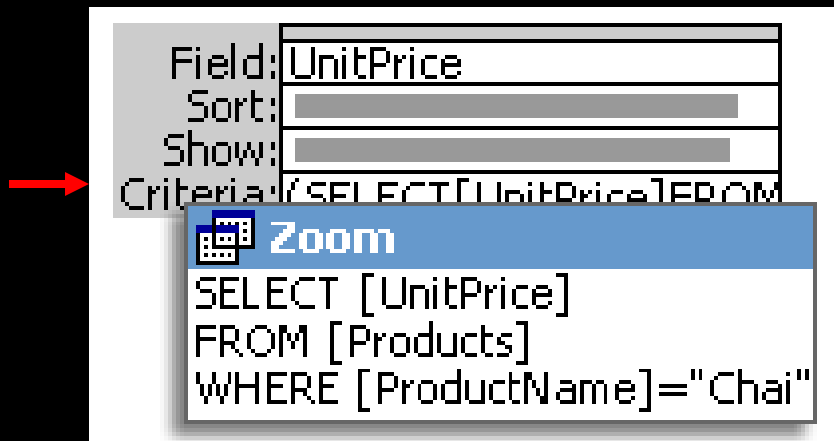
1

1 This union query combines the values in the CompanyName and City fields from the Suppliers and Customers tables if the Country field is "Brazil".

If you want to specify sorting in a union query, add a single ORDER BY clause to the end of the last SELECT statement. In the ORDER BY clause, specify the field name to sort, which must come from the first SELECT statement.

To see the query's results, click **View**  on the toolbar.

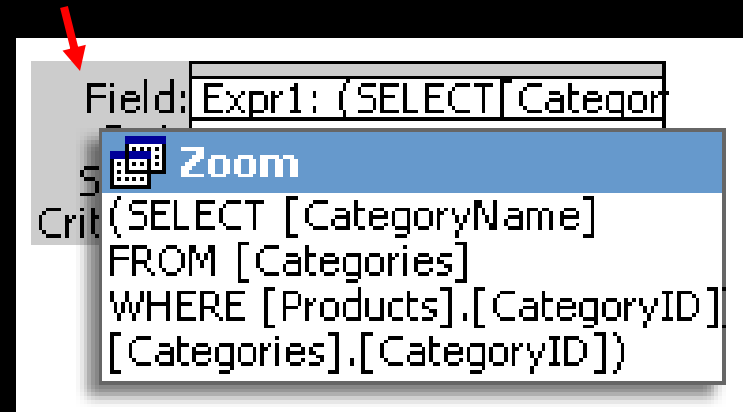
# Subqueries



A screenshot of a query design grid. The 'Field' row contains 'UnitPrice'. The 'Criteria' row contains a subquery: `(SELECT [UnitPrice] FROM [Products] WHERE [ProductName]='Chai')`. A red arrow points to the 'Criteria' row. A 'Zoom' dialog box is open over the subquery, showing the SQL code: `SELECT [UnitPrice]  
FROM [Products]  
WHERE [ProductName]='Chai'`.

Field:	UnitPrice
Sort:	
Show:	
Criteria:	(SELECT [UnitPrice] FROM [Products] WHERE [ProductName]='Chai')

**Zoom**  
SELECT [UnitPrice]  
FROM [Products]  
WHERE [ProductName]='Chai'

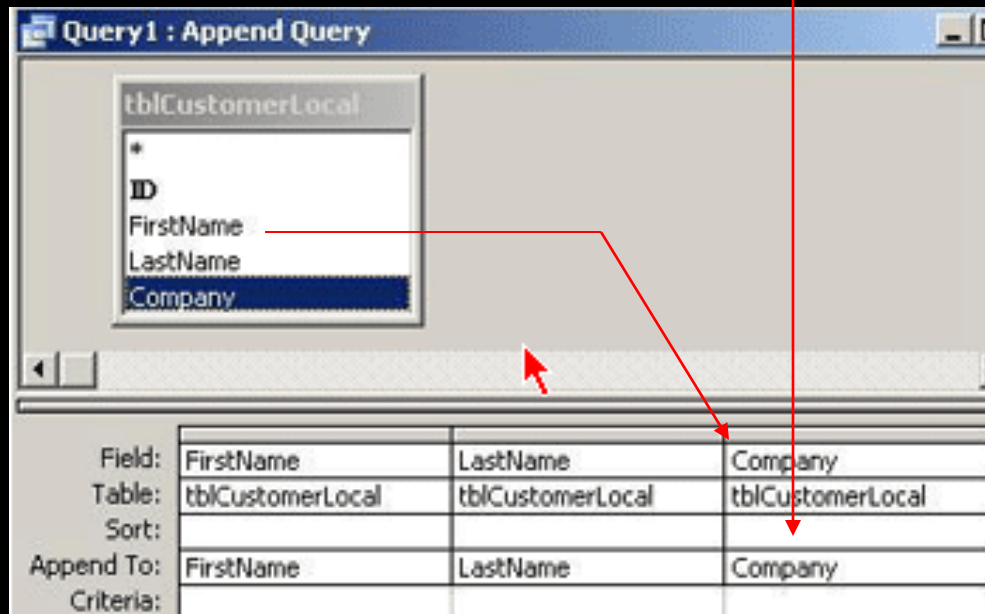
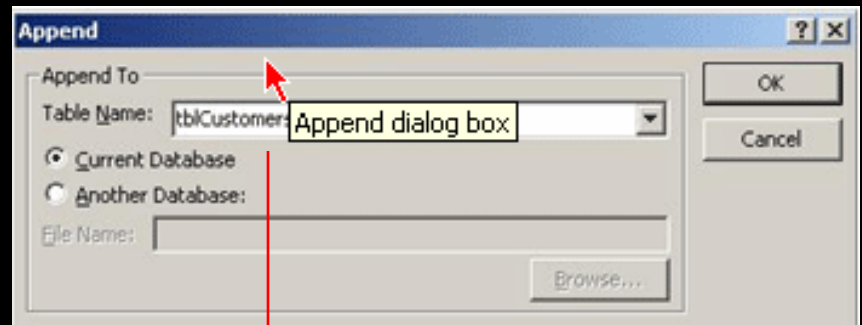
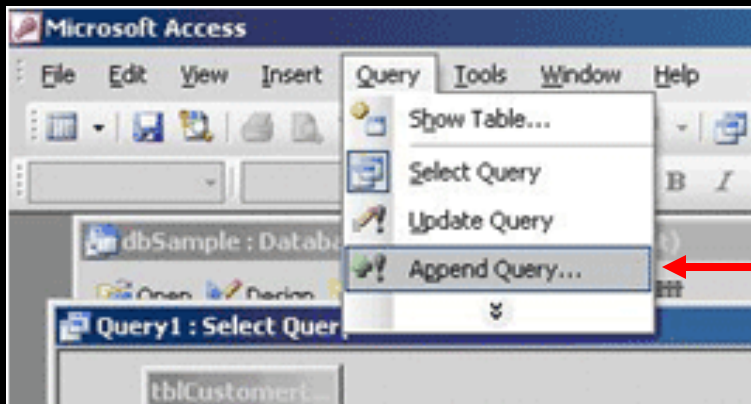


A screenshot of a query design grid. The 'Field' row contains 'Expr1: (SELECT [CategoryName] FROM [Categories] WHERE [Products].[CategoryID]=[Categories].[CategoryID])'. A red arrow points to the 'Field' row. A 'Zoom' dialog box is open over the subquery, showing the SQL code: `SELECT [CategoryName]  
FROM [Categories]  
WHERE [Products].[CategoryID]=[Categories].[CategoryID]`.

Field:	Expr1: (SELECT [CategoryName] FROM [Categories] WHERE [Products].[CategoryID]=[Categories].[CategoryID])
Sort:	
Show:	
Criteria:	

**Zoom**  
SELECT [CategoryName]  
FROM [Categories]  
WHERE [Products].[CategoryID]=[Categories].[CategoryID]

# Append



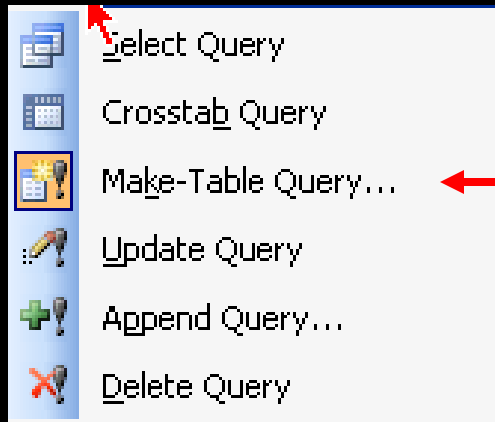
# Update

Field:	UnitPrice
Table:	Products
Update To:	[UnitPrice]*1.1

The unit price will be increased by 10%.

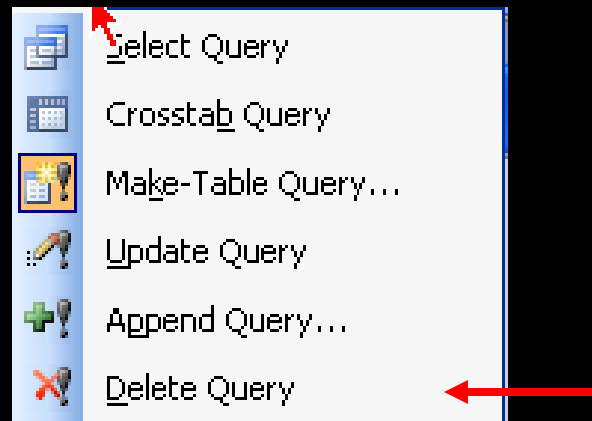
```
UPDATE Products SET Products.UnitPrice = [UnitPrice]*1.1;
```

# Make Table

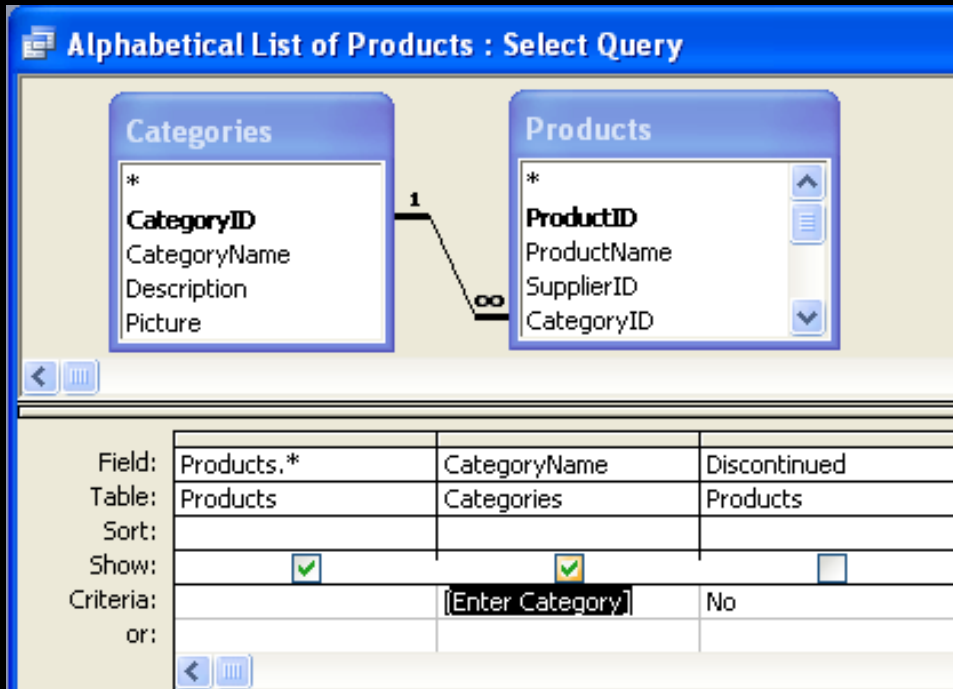


```
SELECT Customers.* INTO tbl_New  
FROM Customers;
```

# Delete



# Parameters

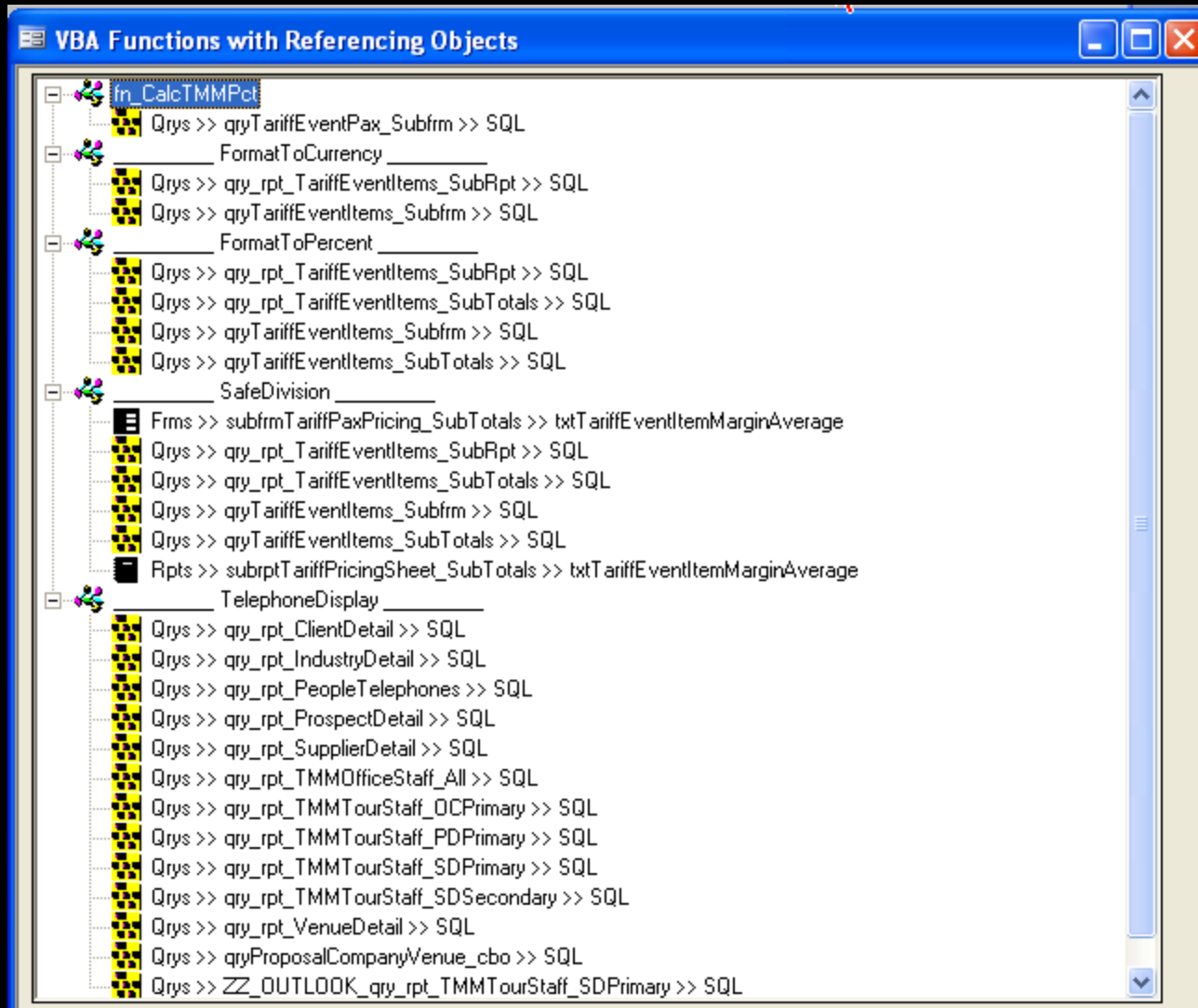


Naming Conventions?

```
SELECT Products.*, Categories.CategoryName
FROM Categories INNER JOIN Products ON Categories.CategoryID =
Products.CategoryID
WHERE (((Categories.CategoryName)=[Enter Category]) AND
((Products.Discontinued)=No));
```



# VBA Functions



# Functions

Telephone:

```
If(IsNull([CompanyMainTel]),"",TelephoneDisplay([CompanyMainTel]))
```

```
Public Function TelephoneDisplay(strIn As String) As String
```

```
    Select Case Len(strIn & vbNullString)
```

```
        Case 0
```

```
            TelephoneDisplay = ""
```

```
        Case 1 To 6, 8, 9
```

```
            TelephoneDisplay = "Invalid Number"
```

```
        Case Is = 7
```

```
            TelephoneDisplay = "Missing Area Code"
```

```
        Case 10
```

```
            TelephoneDisplay = "(" & Left(strIn, 3) & ")" _  
                                & " " & Mid(strIn, 4, 3) _  
                                & "-" & Right(strIn, 4)
```

```
    End Select
```

```
End Function
```

# Performance

## UltraEdit

# Wrap Up

- Huge Topic!
- Get globally familiar w/ Query Topics
- Make sure u have experience w/ each
  - Some understanding of usage
  - Practice, practice, practice ...
- Use samples
- Ask the experts → @AUGSD