Exploring Microsoft Office Access 2007



Chapter 3: Customize, Analyze, and Summarize Query Data

Robert Grauer, Keith Mulbery, Maurie Wigman Lockley

Committed to Shaping the Next Generation of IT Experts.



Objectives for the chapter are:

- Understand the order of precedence
- Create a calculated field in a query,
- Create expressions with the Expression Builder
- Create and edit Access functions
- Perform date arithmetic
- Create and work with data aggregates.

Objectives (continued)

- Create and edit Access functions
- Perform date arithmetic
- Create and work with data aggregates

Copyright © 2008 Pearson Prentice Hall. All rights reserved.

3



The Order of precedence is rules for establishing the sequence by which values are calculated in an expression.

Failure to follow these rules will result in faulty calculations, as Access can only perform the calculations as specified by the user.



The order of precedence determines which evaluations occur first in an expression. Parenthetical expressions are evaluated first, then exponentiation, multiplication and division and finally, addition and subtraction.

Access uses the same symbols as Excel: parenthesis for grouping, the caret symbol for exponentiation, the asterisk and forward slash for multiplication and division, and the plus and minus symbol for addition and subtraction



Expressions are formulas based on existing fields and, when calculated, result in new fields. Expressions are used primarily in queries, reports and forms. They may consist of one or more of the following: the names of fields, controls or properties; operators like +, *, or (); functions; and constants or values.



You use an expression to perform a calculation, retrieve the value of a field or control, supply criteria to a query, define rules, create calculated controls and calculated fields, and to define a grouping level for a report.



The parts of an expression may include:

- the identifier that is the name of a field, property, or control; the operator(s) such as +, -, * (multiply), / (divide).
- functions, which are built-in procedures.
- constants that are named items whose values remains constant while Access is running.
- values that consist of literals such as the number 1,75 or the word "Hello".



When creating a calculated field, you must be careful to use correct syntax. A descriptive name should begin the field and should be followed with a colon. All field names used in the expression should be enclosed in square brackets.



In Query Design View, a calculated field is added to a blank column in the design grid. Remember to give your calculated field a descriptive name and follow that name with a colon. Any field names used in the calculation must be enclosed in brackets.



Saving a query that contains a calculated field does not save the data – only the structure or build of the query is saved. This allows new data to be added to a table that is used in a query containing a calculated field. When the query is executed again with the new data, the results will include the new calculations.



The expression builder can be used to help formulate the expressions you use in a calculated field



You can use expressions to add, subtract, multiply, and divide the values in two or more fields or controls.



When using the Expression Builder, all expressions begin with an equal sign. Logic and operand symbols may be either typed or clicked in the area underneath the work area. You may double-click fields to add them to the expression in the work area.



Access functions calculate commonly used expressions such as pmt, present value, and rate. All functions must include arguments , which are the input values to be calculated.



Some functions require arguments. An argument is a value that provides input to the function. If a function requires more than one argument, you separate the arguments with a comma.



The IIF function evaluates a condition and executes one action when the expression is true and an alternate action when the condition is false.



In the following example: **=IIF(Quantity_on_Hand] >= 1**, "**In Stock**", "**Out of Stock**"), the message "In Stock" will display if the value of the Quantity on hand field is 1 or greater, otherwise "Out of Stock" will be displayed. Note that this function has three arguments. One argument is a field in the database, it is surrounded by square brackets. All arguments are separated by quotes.

	Ou		ulto fr	m data calculation	Shipped Date •	Order Date •	Days between order and shipping
					1/22/2006	1/15/2006	
						1/20/2006	
			Invoice Data	1/22/2006	1/20/2006		
				Salesperson 🔺	1/22/2006	1/20/2006	
	Order Date 👝				1/22/2006	1/22/2006	
			Shipped Date			1/22/2006	
					1/31/2006	1/30/2006	
					2/7/2006	2/6/2006	
					2/12/2006	2/10/2006	
<u>a</u>					3/9/2006	3/6/2006	
Field: Table:	If ustomer New	uctomer Nar- Shinned Datel (Order Date) Days between order and chinning (chinned date) londer date				3/10/2006	
	Invoice Data	Invoice Data	Invoice Data	Cays nerveen order and subbuild: Dubbed date-forder date	3/24/2006	3/22/2006	
Sort					3/24/2006	3/24/2006	
Criteria						3/24/2006	
00					4/7/2006	3/24/2006	
	10				4/7/2006	3/24/2006	
	1.1.6000				4/7/2006	3/24/2006	
		_				3/24/2006	
			alcula	ted field using dates			

Access stores dates as serial numbers which allows calculation of dates no matter the format entered.



Variations of the DatePart function:

```
=datepart("q", "01/22/2007")
```

Displays the quarter in which the given date falls

=DatePart("h", Now())

Displays the hour part of the current date

=DatePart("d", Now())

Displays the day part of the current date



Variations of the DateDiff function

=DateDiff("d", [orderdate],[shippeddate])

Displays the number f days between ordering and shipping

=DateDiff("m",#01/06/2006#, #07/23/2007#)

Displays the number of months between the two dates

=DateDiff("d",[dateborn], Now())

Displays the number of days between the dateborn field and the current date

Data Aggregates									
	All Tables 🔍 «	Campus Branch Customers							
	Account 🌣	Expr1000 - LastName - Balance							
	Account : Table	Cordle Collins \$15,490.00							
	Campus Branch Customers	Cordle Collins \$14,250.00							
	Customer	Cordle Collins \$2,100.00							
	Customer : Table	Cordle Collins \$460.00							
Data aggregation	Campus Branch Customers	Bernett Fox \$1,900.00							
allows	Campus Branch Customers	Allison Greene \$18,700.00							
	Branch	Allison Greene \$12,000.00							
summarization and	Branch : Table	Allison Greene \$10,000.00							
consolidation of	Campus Branch Customers	Allison Greene \$3,000.00							
data		Allison Greene \$1,200.00							
uala		Clay Hayes \$4,000.00							
		Clay Hayes \$630.00							
		Clay Hayes \$550.00							
		John Nunn \$16,700.00							
		John Nunn \$1,300.00							
		Laura Peterson \$5,600.00							
		Laura Peterson None							
		Natasha Simpson Sum							
		Eaton Wagner Average							
		Eaton Wagner Count							
		Eaton Wagner Maximum							
		YourFirstName Your Last Minimum							
		YourFirstName Your Last Standard Deviation							
		* Variance							
		Total 🕑							
Copyrigi	nt © 2008 Pearson Prentice Hall. All rights r	reserved. 22							

Data aggregation allows summarization and consolidation of data.



Make sure you are working in a numeric or currency field before beginning an aggregate calculation. Click Totals in the Records group on the Home tab and choose the aggregate function desired from the Totals drop-down list.



Care must be used when working with null fields and aggregate functions. When using the Avg function, null fields ignored.

When using the Count function, null fields are not included unless an asterisk (*), also known as the wildcard characters, is used as the argument for the function. As an example, Count(*), will include null fields in the calculation. The Sum function ignores all null values.

How do Aggregate Functions Handle Null Values? (continued)

- Examples
 - Count(*) will include null fields in the calculation
 - Count(records) will not include null fields
- The Sum function ignores null values

Copyright © 2008 Pearson Prentice Hall. All rights reserved.

25



The total row can be added to the design grid by clicking the Totals Icon.



A totals query organizes query results into groups. This type of query consists of only the field or fields that you want to total and the grouping field.

