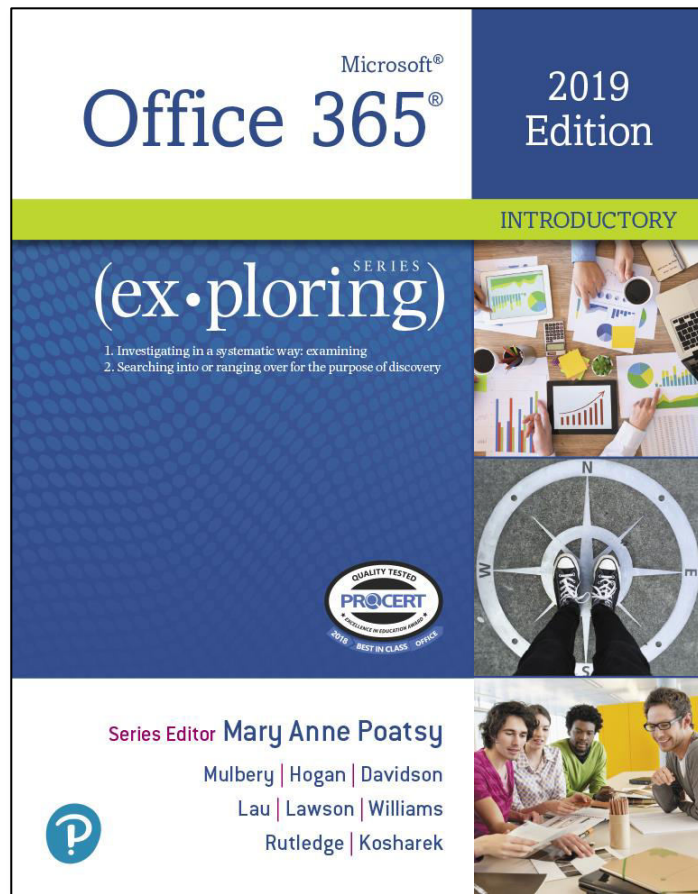


Exploring Microsoft Office 365

Introductory



Chapter 3

Query Calculations and Expressions

Learning Objectives (1 of 2)

- 3.1** Create a Query with a Calculated field
- 3.2** Format Calculated Results
- 3.3** Recover from Common Errors
- 3.4** Verify Calculated Results
- 3.5** Create Expressions using the Expression Builder

Learning Objectives (2 of 2)

3.6 Use Built-in Functions

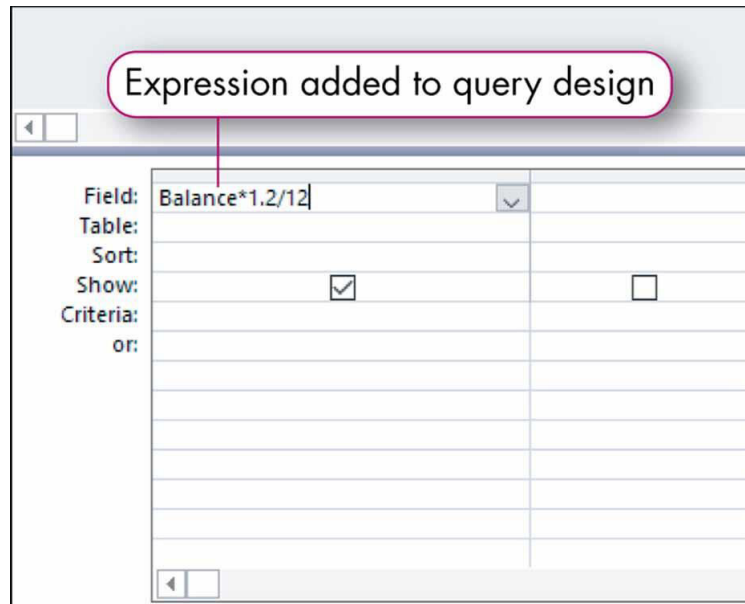
3.7 Add Aggregate Functions to Datasheets

3.8 Create Queries with Aggregate Functions

Creating a Query with a Calculated Field (1 of 4)

- Calculated field—displays the result of a calculation based on existing database field(s)
- Expression—formula used in a calculated field
 - Typed manually
 - Inserted using Access tools

Creating a Query with a Calculated Field (2 of 4)



Pearson Education, Inc. © 2020

The screenshot shows the Microsoft Access interface. On the left is the 'Navigation Pane' with a 'Query1' tab. The main area displays a table named 'MonthlyPayment' with a single column of numerical values. A red bracket on the right side of the table is connected by a line to a callout box containing the text 'Results of calculation'.

MonthlyPayment
24.012346
30.012034
109.957755
130.047774
136.723774
150.099997
153.623123
170.022222
180.015554
198.012349
245.027734
270.09
280.012734
342.555554
360.05
560.000009
647.512734
680.047774
900.04

Results of calculation

Record: 1 of 19 No Filter Search

Datasheet View

Pearson Education, Inc. © 2020

Creating a Query with a Calculated Field (3 of 4)

Calculated field elements:

- Arithmetic operators – such as *, /, +, or –
- Constant—a value that does not change
- Function—built-in calculations
- Identifier—names of fields, controls, or properties

Creating a Query with a Calculated Field (4 of 4)

Function	Symbol
Parenthesis	()
Exponentiation	^
Multiplication	*
Division	/
Addition	+
Subtraction	-

Formatting Calculated Results

Property Sheet

Selection type: Field Properties

General Lookup

Description

Format

Input Mask

Caption

Text Format

MonthlyPayment field selected for format updates

Caption property

Format options

Format property arrow

Close the Property Sheet

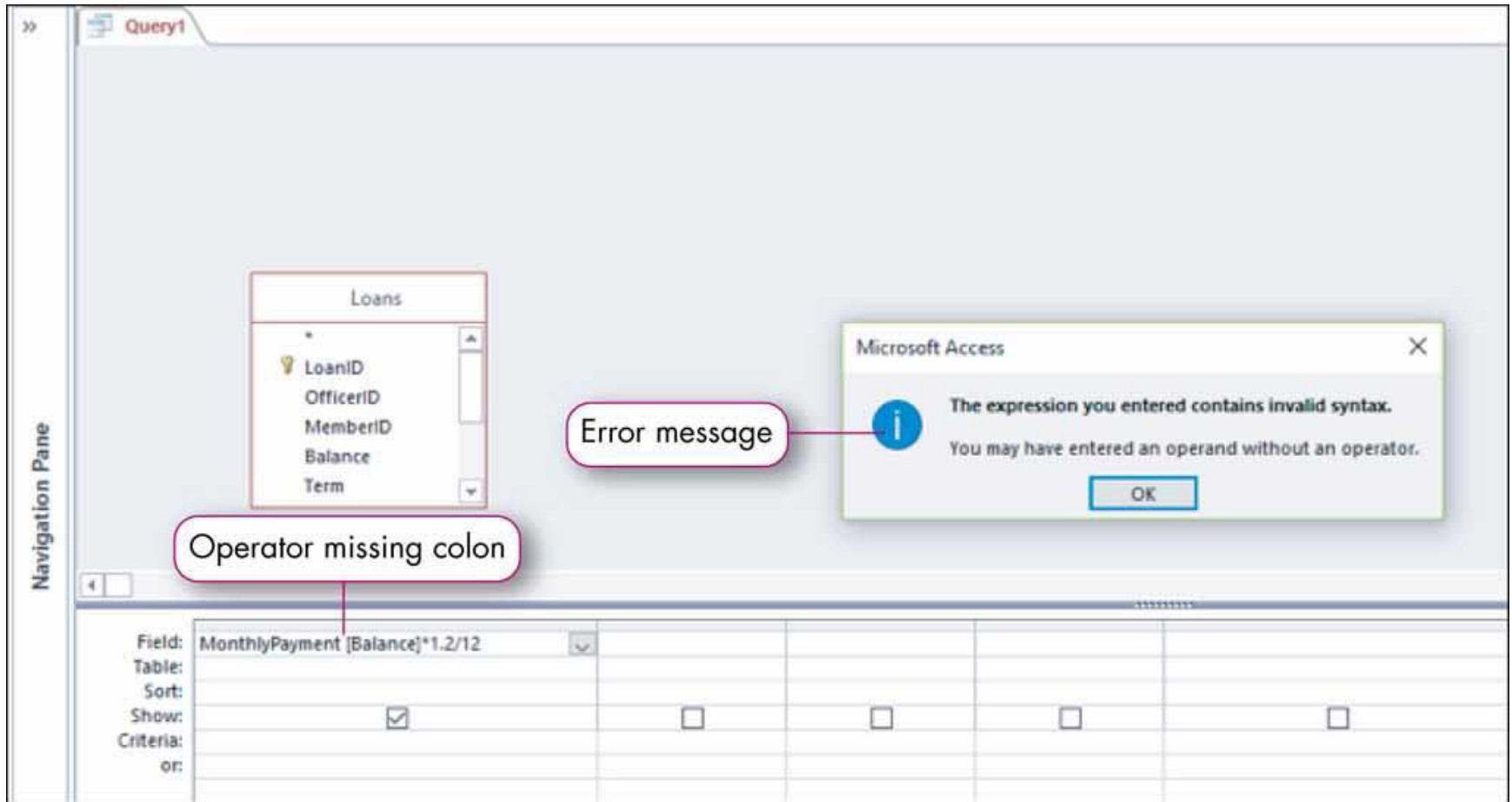
General Date	11/12/2015 5:34:23 PM
Long Date	Thursday, November 12, 2015
Medium Date	12-Nov-15
Short Date	11/12/2015
Long Time	5:34:23 PM
Medium Time	5:34 PM
Short Time	17:34
General Number	3456.789
Currency	\$3,456.79
Euro	€3,456.79
Fixed	3456.79
Standard	3,456.79
Percent	123.00%
Scientific	3.46E+03
True/False	True
Yes/No	No

MonthlyPayment: [Balance]*1.2/12

Pearson Education, Inc. © 2020

Recovering from Common Errors

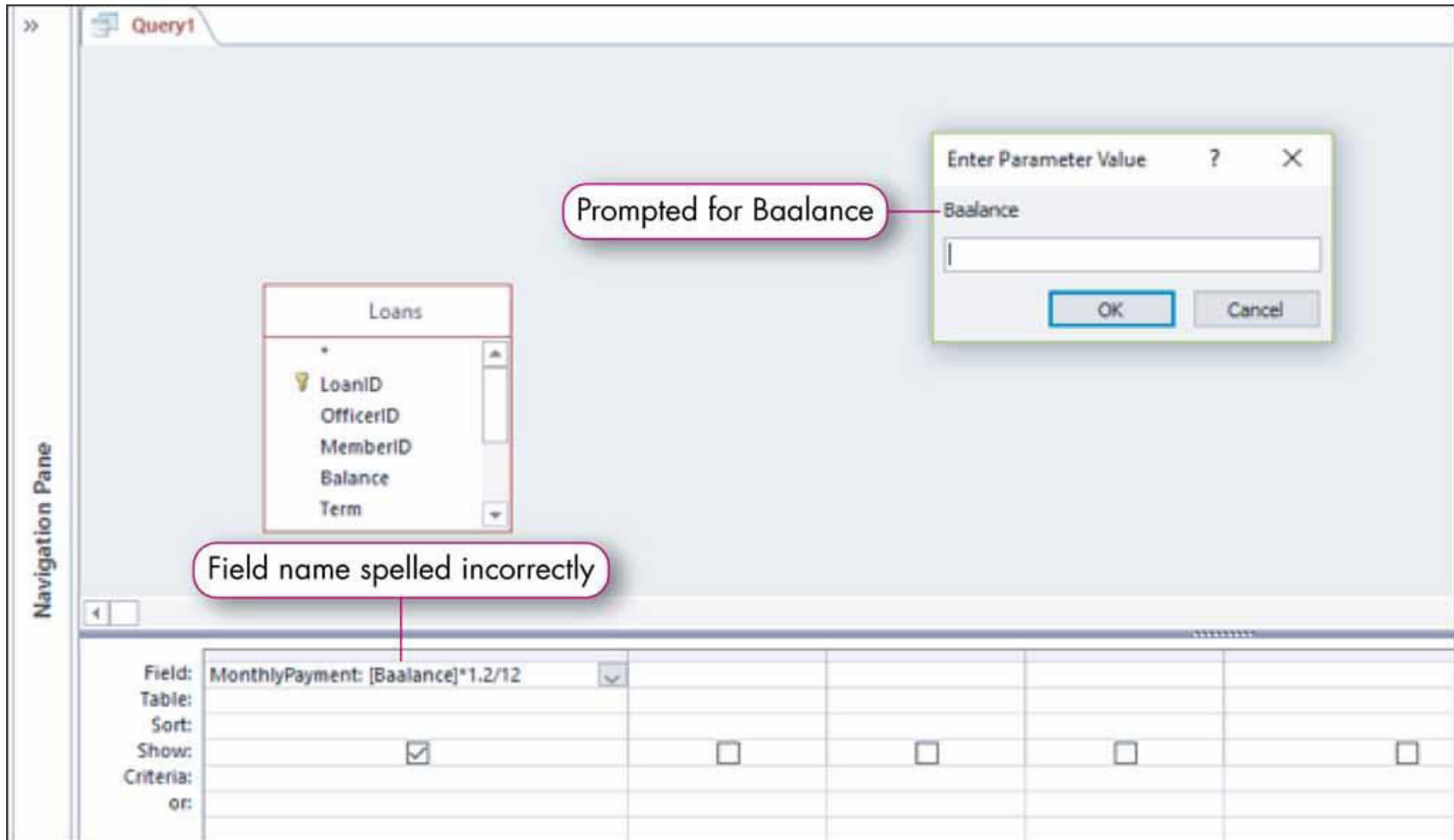
(1 of 2)



Pearson Education, Inc. © 2020

Recovering from Common Errors

(2 of 2)

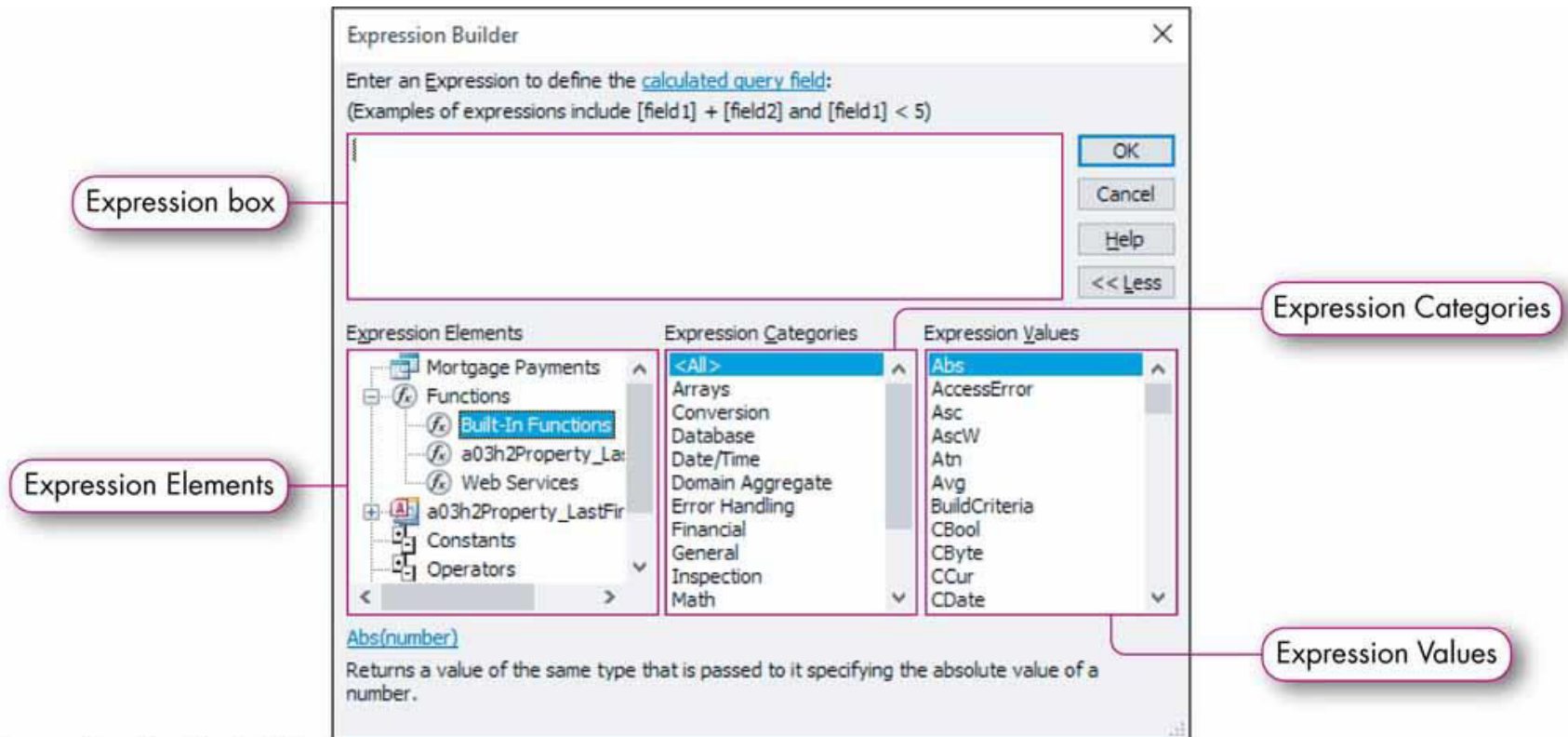


Pearson Education, Inc. © 2020

Verifying Calculated Results

- After running a query, review the field values and calculated values in Datasheet view
- Verify results with a calculator
- Check the results in Excel by:
 - Copying and pasting the data into a worksheet
 - Performing the calculations
 - Comparing the answers to the query results in Access

Create Expressions using the Expression Builder (1 of 1)



Pearson Education, Inc. © 2020

Create Expressions using the Expression Builder (2 of 2)

The screenshot shows the 'Expression Builder' dialog box. The main text area contains the expression: `PricePerBR: [ListPrice]/ [Properties]![Beds]`. Below this, there are three panes: 'Expression Elements', 'Expression Categories', and 'Expression Values'. The 'Expression Elements' pane shows a tree view with 'Tables' selected, and 'Properties' is highlighted. The 'Expression Categories' pane shows a list of fields, with 'Beds' highlighted. The 'Expression Values' pane shows a list of values, with '<Value>' highlighted. Callout boxes point to these elements: 'Expression created' points to the main text area, 'Table selected in Expression Elements' points to the 'Properties' table in the 'Expression Elements' pane, and 'Fields listed in Expression Categories' points to the 'Beds' field in the 'Expression Categories' pane.

Expression created

Table selected in Expression Elements

Fields listed in Expression Categories

Expression Builder

Enter an Expression to define the [calculated query field](#):
(Examples of expressions include [field1] + [field2] and [field1] < 5)

PricePerBR: [ListPrice]/ [Properties]![Beds]

OK
Cancel
Help
<< Less

Expression Elements

Functions
a03h3Property_LastFir
Tables
Agents
Lists
Properties
Subdivision
Queries

Expression Categories

Listing
DateListed
DateSold
ListPrice
SalePrice
SqFeet
Beds
Baths
Address
SubDivisionID
County

Expression Values

<Value>

Pearson Education, Inc. © 2020

Using Built-in Functions (1 of 4)

- Function—predefined computation that performs a complex calculation
- Arguments—specific inputs used to complete a calculation
- Each function has a specific syntax that determines the layout and order of the function and its arguments

Using Built-in Functions (2 of 4)

- Pmt function—calculates the loan payment given the rate, number of periods, and the present value of the loan
- Pmt function syntax:

Pmt(<<rate>>, <<num_periods>>, <<present_value>>,
<<future_value>>, <<type>>)

Using Built-in Functions (3 of 4)

Pmt function inserted

Built-In Functions

Expression Builder

Enter an Expression to define the [calculated query field](#):
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Pmt(«rate», «num_periods», «present_value», «future_value», «type»)

OK
Cancel
Help
<< Less

Expression Elements

- List Price Calculations
- Functions
 - Built-In Functions
 - a03h3Property_La:
 - Web Services
- a03h3Property_LastFir
- Constants
- Operators

Expression Categories

- <All>
- Arrays
- Conversion
- Database
- Date/Time
- Domain Aggregate
- Error Handling
- Financial
- General
- Inspection
- Math

Expression Values

- DDB
- FV
- IPmt
- IRR
- MIRR
- NPer
- NPV
- Pmt
- PPmt
- PV
- Rate

[Pmt\(rate, num_periods, present value, \[future value\], \[type\]\)](#)

Returns a Double specifying the payment for an annuity based on periodic fixed payments and a fixed interest rate.

Pearson Education, Inc. © 2020

Using Built-in Functions (4 of 4)

The screenshot shows the 'Expression Builder' dialog box with the following components and annotations:

- Expression Builder** (Title Bar)
- Enter an Expression to define the calculated query field:** (Instruction)
- (Examples of expressions include [field1] + [field2] and [field1] < 5)** (Examples)
- Expression Text:** `Pmt(0.05/12, 4*12, 12500, 0, 0)`
 - Negative sign placed in front of PV argument:** Points to the opening parenthesis.
 - Interest rate per period:** Points to `0.05/12`.
 - Number of periods:** Points to `4*12`.
 - Present value of loan:** Points to `12500`.
 - Optional arguments set to zero:** Points to `0, 0`.
- Buttons:** OK, Cancel, Help, << Less
- Expression Elements:**
 - Query1
 - Functions
 - Built-In Functions
 - a03d1Interest
 - Web Services
 - a03d1Interest.accdb
 - Constants
 - Operators
 - Common Expressions
- Expression Categories:**
 - <All>
 - Arrays
 - Conversion
 - Database
 - Date/Time
 - Domain Aggregate
 - Error Handling
 - Financial
 - General
 - Inspection
 - Math
- Expression Values:**
 - NPer
 - NPV
 - Nz
 - Oct
 - Oct\$
 - Partition
 - PlainText
 - Pmt**
 - PPmt
 - PV
 - QBColor
- Function Signature:** `Pmt(rate, num_periods, present_value, [future_value], [type])`
- Description:** Returns a Double specifying the payment for an annuity based on periodic fixed payments and a fixed interest rate.

Pearson Education, Inc. © 2020

Adding Aggregate Functions to Datasheets (1 of 4)

- Aggregate function—performs a calculation on an entire column of data and returns a single value
- Access refers to aggregate functions as Totals and can be added to:
 - Datasheet view of a query
 - Query's Design view

Adding Aggregate Functions to Datasheets (2 of 4)

Two methods for adding aggregate function:

- Total row—use an aggregate function in one or more columns of a data table or a query result without having to change the query design
- Totals query—calculates subtotals for groups of records

Adding Aggregate Functions to Datasheets (3 of 4)

Function	Description
Avg (Average)	Calculates the average value for a column.
Count	Counts the number of values in a column.
Max (Maximum)	Return the item with the highest value.
Min (Minimum)	Returns the item with the lowest value.
Sum	Totals the items in a column

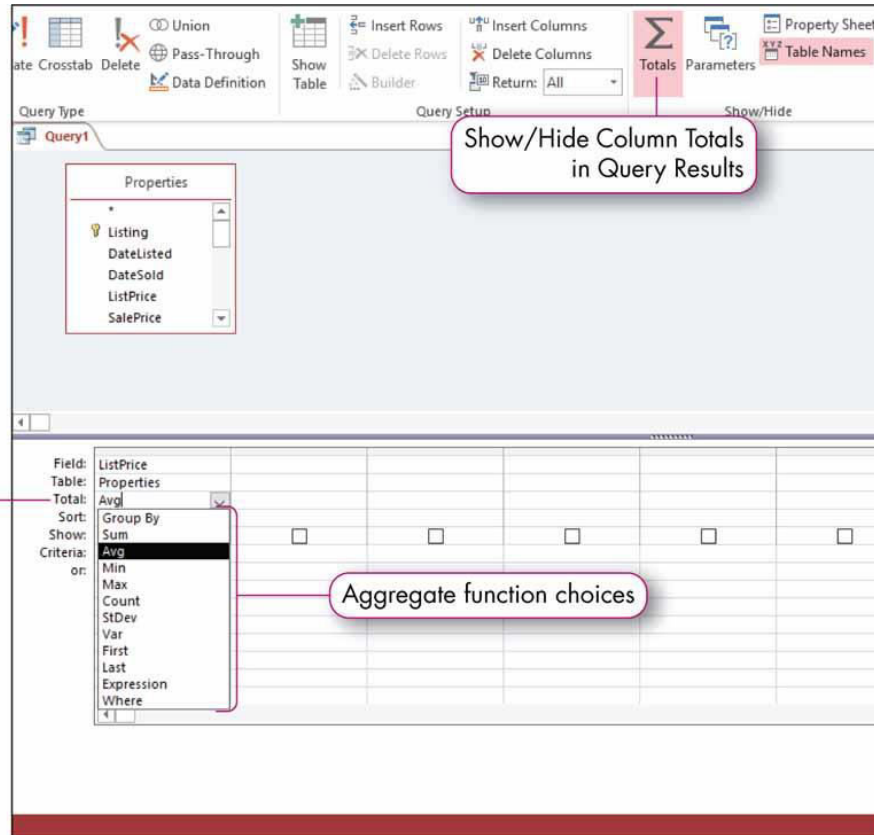
Adding Aggregate Functions to Datasheets (4 of 4)

The screenshot shows a data table titled "Mortgage Payments" with columns: Listing, List Price, Sold, DownPayment, and MonthlyPayment. A "Totals" button in the top right corner of the table is highlighted with a red circle. A dropdown menu is open from this button, listing various aggregate functions: None, Sum, Average, Count, Maximum, Minimum, Standard Deviation, and Variance. A red line points from the "Average" option in the dropdown to the "Average of List Price" text in a callout box. Another red line points from the "Totals" button to the "Available aggregate functions" text in a callout box. The bottom row of the table shows the calculated total for the "List Price" column as \$167,100.47.

Listing	List Price	Sold	DownPayment	MonthlyPayment
10028	\$129,780.00	No	\$25,956.00	\$495.67
10044	\$180,810.00	No	\$36,162.00	\$690.57
10060	\$187,180.00	No	\$37,436.00	\$714.90
10061	\$140,904.00	No	\$28,180.80	\$538.16
10064	\$189,711.00	No	\$37,942.20	\$724.57
10066	\$177,984.00	No	\$35,596.80	\$679.78
10069	\$140,693.00	No	\$28,138.60	\$537.35
10082	\$166,320.00	No	\$33,264.00	\$635.23
10102	\$179,712.00	No	\$35,942.40	\$686.38
10104	\$174,230.00	No	\$34,846.00	\$665.44
10114	\$184,688.00	No	\$36,937.60	\$705.38
10117	\$164,436.00	No	\$32,887.20	\$628.03
10851	\$166,552.00	No	\$33,310.40	\$636.12
10854	\$184,688.00	No	\$36,937.60	\$705.38
10868	\$175,336.00	No	\$35,067.20	\$669.66
10885	\$168,354.00	No	\$33,670.80	\$643.00
10910	\$163,737.00	No	\$32,747.40	
10921	\$174,720.00	No	\$34,944.00	
11028	\$142,380.00	No	\$28,476.00	
11035	\$174,720.00	No	\$34,944.00	
11036	\$175,560.00	No	\$35,112.00	
11058	\$187,254.00	No	\$37,450.80	
11079	\$179,100.00	No	\$35,820.00	
Total	\$167,100.47			

Pearson Education, Inc. © 2020

Create Queries with Aggregate Functions (1 of 4)



The screenshot shows the 'Overall Results' window. It displays a table with one row and one column. The column header is 'AvgOfSalePrice' and the value is '\$280,229.50'.

AvgOfSalePrice
\$280,229.50

Create Queries with Aggregate Functions (2 of 4)

Overall Results

Properties

County
Sold
AgentID
ListID
Style
Construction

Field: County
Table: Properties
Total: Group By
Sort: Avg
Show: ☒
Criteria: ☒
or: ☒

County set as grouping field

County	SalePrice	SqFeet	Beds	Baths
Properties	Properties	Properties	Properties	Properties
Avg	Avg	Avg	Avg	Avg

Pearson Education, Inc. © 2020

Fields set to Avg

County	AvgOfSalePrice	AvgOfSqFeet	AvgOfBeds	AvgOfBaths
Bergen	\$220,431.12	2218.08163265306	3.12244897959184	1.63265306122449
Essex	\$284,091.38	2844.50574712644	3.90804597701149	1.86206896551724
Hudson	\$243,285.50	2645.23529411765	3.41176470588235	1.64705882352941
Mercer	\$341,085.67	3242.675	4.45	2
Morris	\$322,404.05	3105.6875	4.140625	1.890625
Passaic	\$219,325.20	2049.03846153846	2.80769230769231	1.75
Sussex	\$269,411.17	2968	4.5	2.1

Results shown for each county

Pearson Education, Inc. © 2020

Create Queries with Aggregate Functions (3 of 4)

Overall Results

Properties

- County
- Sold
- AgentID
- ListID
- Style
- Construction

Where condition applied

Field:	County	SalePrice	SqFeet	Beds	Baths	Sold
Table:	Properties	Properties	Properties	Properties	Properties	Properties
Total:	Group By	Avg	Avg	Avg	Avg	Where
Sort:						
Show:	☒	☒	☒	☒	☒	☐
Criteria:						Yes
or:						

Condition added to Sold field

Pearson Education, Inc. © 2020

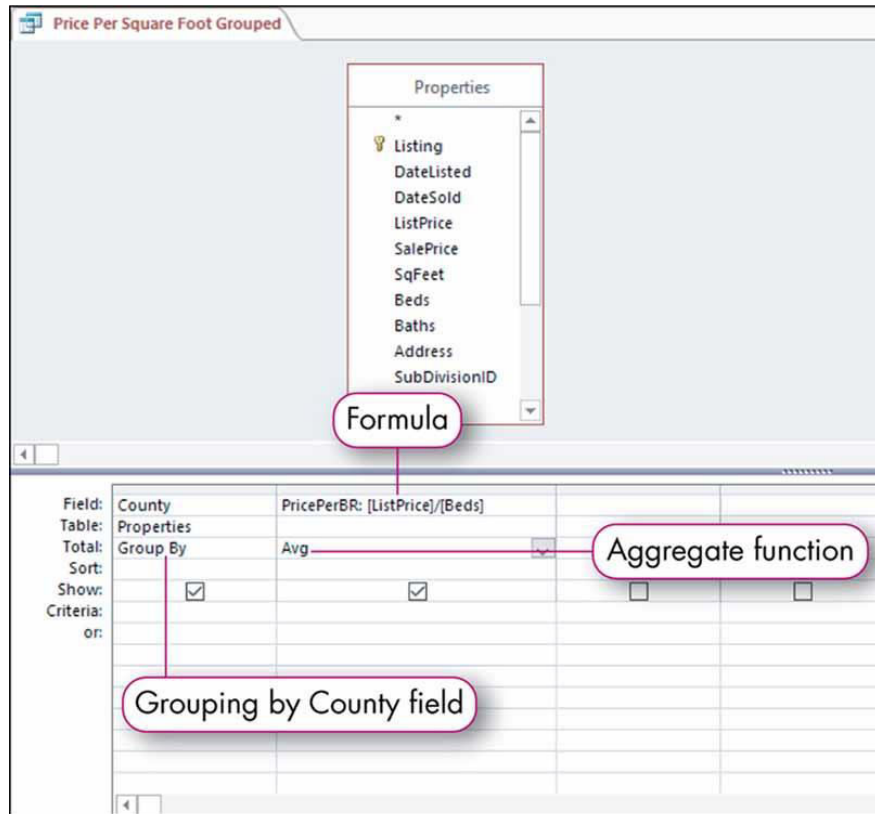
Create Queries with Aggregate Functions (4 of 4)

Only sold homes are included in these averages

Grouped
by County

Overall Results				
County	AvgOfSalePrice	AvgOfSqFeet	AvgOfBeds	AvgOfBaths
Bergen	\$220,431.12	2223.11764705882	3.29411764705882	1.73529411764706
Essex	\$284,091.38	2829.53846153846	3.96153846153846	1.94230769230769
Hudson	\$243,285.50	2432.25	3	1.75
Mercer	\$341,085.67	3440.55555555556	4.55555555555556	2
Morris	\$322,404.05	3233.42857142857	4.33333333333333	1.9047619047619
Passaic	\$219,325.20	2171.4	2.8	1.8
Sussex	\$269,411.17	2610	3.83333333333333	2

Create Queries with Aggregate Functions (5 of 4)



Pearson Education, Inc. © 2020

Results grouped by County name

County	PricePerBR
Bergen	\$73,946.82
Essex	\$76,025.00
Hudson	\$81,793.61
Mercer	\$74,941.35
Morris	\$77,588.72
Passaic	\$75,387.97
Sussex	\$69,097.29

PricePerBR results for each County

Pearson Education, Inc. © 2020

Copyright



This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.