Using the Power Pivot Features of Power BI Desktop

Agenda

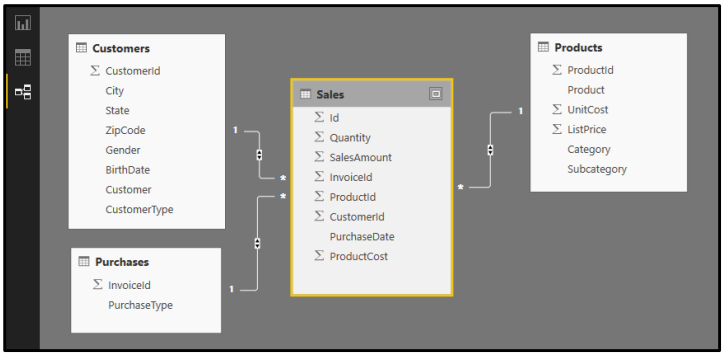
- Data Modeling with PowerPivot and DAX
- Creating Calculated Columns
- Integrating Lookup Tables
- Creating Measures

Building a Data Model with Power Pivot

- Steps to create a data model with Power Pivot
  - Create relationships between tables
  - Modify columns (rename, set formatting, convert type)
  - Create calculated columns
  - Create measures
- Features not yet supported by Power BI Desktop
  - Creating dimensional hierarchies
  - Creating key performance indicators (KPIs)

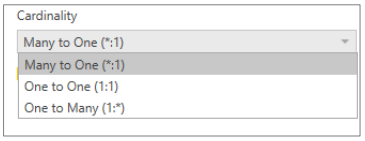
Table Relationships

- Tables in data model associated with relationships
  - Relationships based on single columns
  - Tabular model supports [1-to-1] and [1-to-many] relationships
  - Relationships based on single column in each table

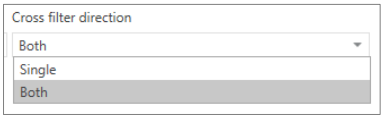


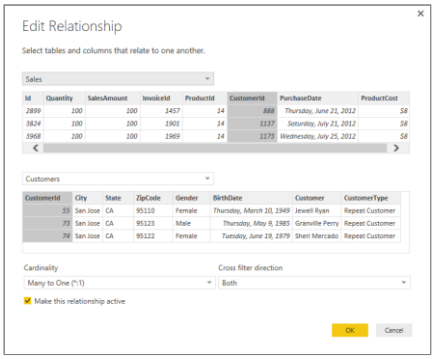
Relationship Properties

- Cardinality



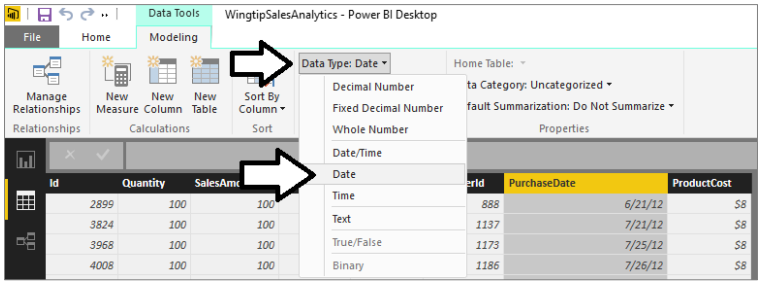
- Cross filter direction





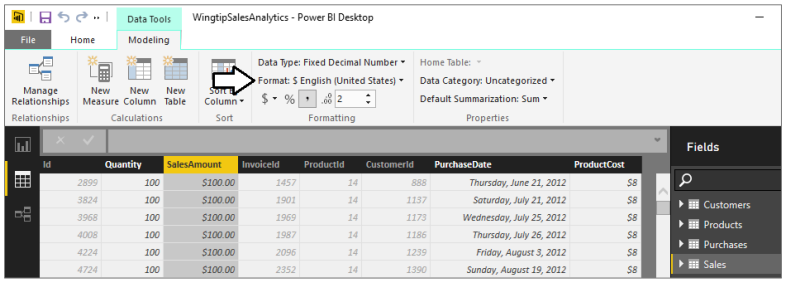
Converting Column Types

- Power Pivot allows you to convert columns
  - Alternative to converting column with Power Query



Formatting Columns

- Each column has its own formatting properties
  - Formatting propagated to reports and visuals
  - Makes it easier on data model consumers



Working with DAX

- DAX is the language used to create data model
  - DAX stands for "**D**ata **A**nalysis E**x**pression Language"
- DAX expressions are similar to Excel formulas
  - They always start with an equal sign (=)
  - DAX provides many built-in functions similar to Excel
- DAX Expressions are unlike Excel formulas…
  - DAX expressions cannot reference cells (e.g. A1 or C4)
  - Instead DAX expressions reference columns and tables

```
=SUM('Sales'[SalesAmount])
```

Writing DAX Expressions

- Some DAX expressions are simple

```
Sales Revenue = Sum(Sales[SalesAmount])
```

- Some DAX expressions are far more complex

```
Sales Growth PM = IF(
  ( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() ),
  DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    ),
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    )
  ),
  BLANK()
)
```


Calculated Columns vs Measures

- Calculated Columns
  - Evaluated based on context of a single row
  - Evaluated when data is loaded into memory

```
Column1 = <DAX expression>
```

- Measures
  - Evaluated at query time based on current filter context
  - Commonly used for aggregations (e.g. SUM, AVG, etc.)
  - Used more frequently than calculated columns

```
Measure1 = <DAX expression>
```

Types of DAX Functions

- Date and Time Functions
- Information Functions
- Logical Functions
- Mathematical and Trigonometric Functions
- Statistical Functions
- Filter Functions
- Text Functions
- Time Intelligence Functions

Agenda

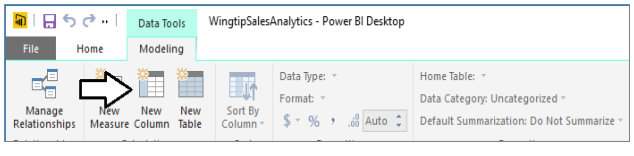
- ✓ Data Modeling with PowerPivot and DAX
- ➢ Creating Calculated Columns
- Integrating Lookup Tables
- Creating Measures

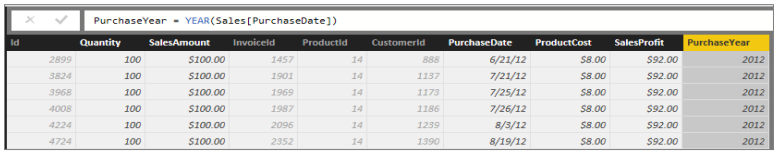
When to Create Calculated Columns

- Measures often better choice than calculate columns
  - Don't create calculated column when you need a measure
  - Prefer to create calculated columns only in specific scenarios
- When should you create calculated columns?
  - To create headers for row labels or column labels
  - To place calculated results in a slicer for filtering
  - Define an expression strictly bound to current row
  - Categories text or numbers (e.g. customer age groups)

Creating Calculated Columns

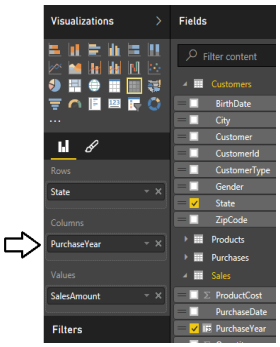
- Edited in formula bar of Power Pivot data view
  - Start with name and then equals (=) sign
  - Enter a valid DAX expression
  - Clicking on column adds it into expression





Calculated Column as a Column Label

- Calculate column can serve as...
  - Row labels
  - Column labels



State	2012	2013	2014	2015	Total
CA	$270,926.32	$550,160.02	$737,878.53	$770,402.11	$2,329,366.98
TX	$212,085.08	$490,643.98	$683,079.11	$919,030.36	$2,304,838.53
FL	$51,730.85	$300,866.87	$535,693.94	$891,344.92	$1,779,636.58
NC	$11,018.02	$164,804.24	$315,139.92	$448,638.72	$939,600.90
NY	$24,207.43	$165,046.23	$256,294.27	$430,971.24	$876,519.17
GA	$40,305.80	$152,807.51	$239,451.05	$417,037.28	$849,601.64

Calculated Column for Customer Age Group

1. Calculate customer age from birthdate

```
Age = Floor( (TODAY()-Customers[BirthDate])/365, 1)
```

CustomerId	City	State	ZipCode	Gender	BirthDate	Customer	CustomerType	Age
55	San Jose	CA	95110	Female	3/10/49	Jewell Ryan	Repeat Customer	66
73	San Jose	CA	95123	Male	5/9/85	Granville Perry	Repeat Customer	30
74	San Jose	CA	95122	Female	6/19/79	Sheri Mercado	Repeat Customer	36
78	San Jose	CA	95110	Male	6/16/78	Raleigh Olson	Repeat Customer	37
136	San Jose	CA	95124	Female	1/2/45	Carrie Foreman	Repeat Customer	70
150	San Jose	CA	95134	Female	8/11/84	Renee McMillan	Repeat Customer	31

2. Calculate age groups using calculated column

```
Age Group =
  SWITCH( TRUE(),
    [Age] >= 65, "Ages 65 and over",
    [Age] >= 50, "Ages 50 TO 65",
    [Age] >= 40, "Ages 40 TO 49",
    [Age] >= 30, "Ages 30 TO 39",
    [Age] >= 18, "Ages 18 TO 23",
    [Age] < 18, "Ages under 18"
  )
```

CustomerId	City	State	ZipCode	Gender	BirthDate	Customer	CustomerType	Age	Age Group
55	San Jose	CA	95110	Female	3/10/49	Jewell Ryan	Repeat Customer	66	Ages 65 and over
73	San Jose	CA	95123	Male	5/9/85	Granville Perry	Repeat Customer	30	Ages 30 TO 39
74	San Jose	CA	95122	Female	6/19/79	Sheri Mercado	Repeat Customer	36	Ages 30 TO 39
78	San Jose	CA	95110	Male	6/16/78	Raleigh Olson	Repeat Customer	37	Ages 30 TO 39
136	San Jose	CA	95124	Female	1/2/45	Carrie Foreman	Repeat Customer	70	Ages 65 and over
150	San Jose	CA	95134	Female	8/11/84	Renee McMillan	Repeat Customer	31	Ages 30 TO 39
178	San Jose	CA	95123	Male	9/7/89	Wayne Gordon	Repeat Customer	26	Ages 18 TO 23
183	San Jose	CA	95110	Female	5/26/85	Luella Vinson	Repeat Customer	30	Ages 30 TO 39
213	San Jose	CA	95123	Male	8/13/78	Rosario Knight	Repeat Customer	37	Ages 30 TO 39

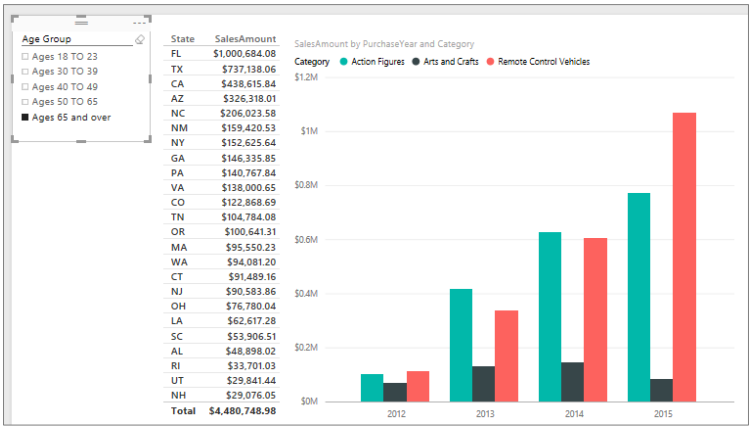
Calculated Column as a Row Label

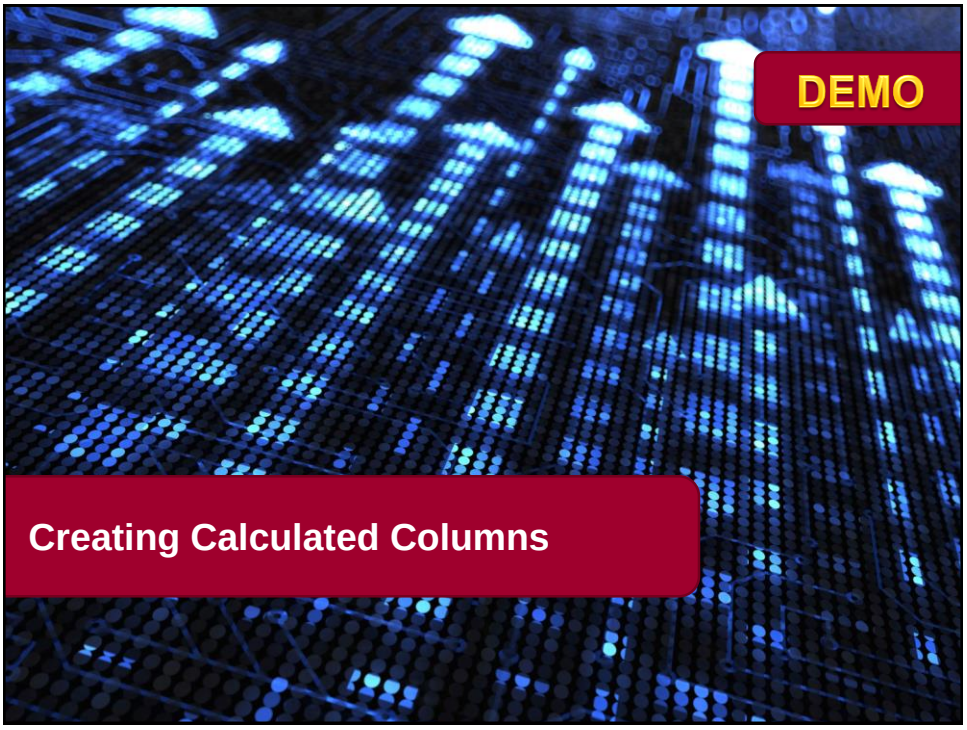
- Age Group can now be used as row label

Age Group	Count of CustomerId	Quantity	SalesAmount
Ages 50 TO 65	7974	1,143,644	$4,904,342.59
Ages 65 and over	7569	1,050,879	$4,480,748.98
Ages 40 TO 49	5103	713,453	$3,083,964.23
Ages 30 TO 39	4547	646,121	$2,798,053.19
Ages 18 TO 23	3278	478,405	$1,985,991.24
Total	**28471**	**4,032,502**	**$17,253,100.23**

Calculated Column used in a Slicer

- Calculated column can populate slicer values



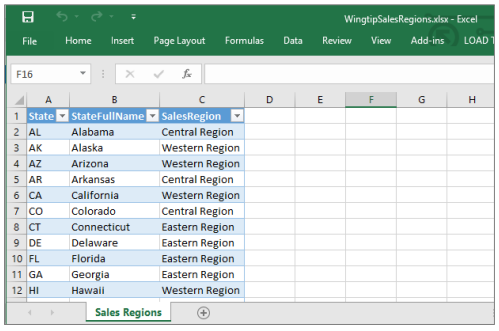


Agenda

- ✓ Data Modeling with PowerPivot and DAX
- ✓ Creating Calculated Columns
- ➢ Integrating Lookup Tables
- • Creating Measures

Adding Lookup Tables to the Data Model

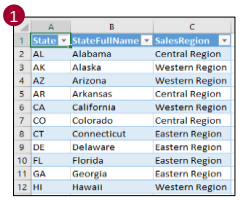
- Data modeling might required adding lookup tables
  - Lookup tables inject extra related data into data model
- Example: Sales Regions table
  - Assign each state to specific sales region
  - Include full state name it required in reporting



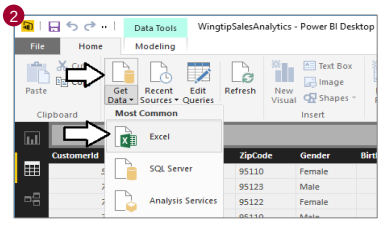
	State	StateFullName	SalesRegion
1	AL	Alabama	Central Region
2	AK	Alaska	Western Region
3	AZ	Arizona	Western Region
4	AR	Arkansas	Central Region
5	CA	California	Western Region
6	CO	Colorado	Central Region
7	CT	Connecticut	Eastern Region
8	DE	Delaware	Eastern Region
9	FL	Florida	Eastern Region
10	GA	Georgia	Eastern Region
11	HI	Hawaii	Western Region

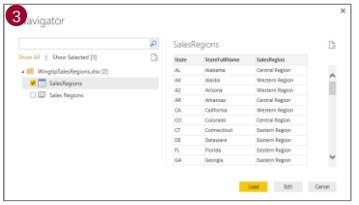
Importing the SalesRegions Table from Excel

- Import table from Excel using Power Query



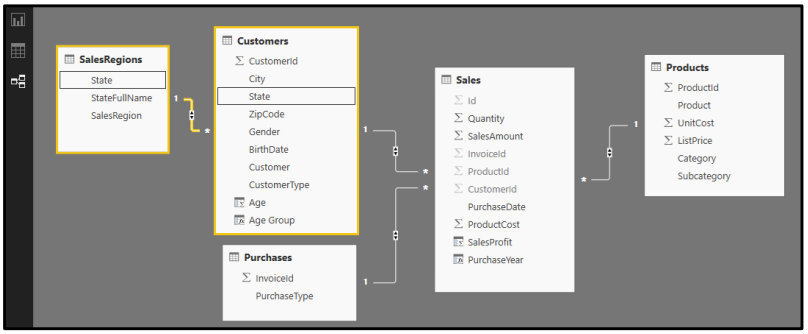
	State	StateFullName	SalesRegion
1	AL	Alabama	Central Region
2	AK	Alaska	Western Region
3	AZ	Arizona	Western Region
4	AR	Arkansas	Central Region
5	CA	California	Western Region
6	CO	Colorado	Central Region
7	CT	Connecticut	Eastern Region
8	DE	Delaware	Eastern Region
9	FL	Florida	Eastern Region
10	GA	Georgia	Eastern Region
11	HI	Hawaii	Western Region





Integrating the Lookup Table into the Data Model

- Lookup table must be integrated into data model
  - Accomplished by creating relationship to one or more tables



The RELATED Function

- RELATED function performs cross-table lookup
  - Effectively replaces older VLOOKUP function
  - Used in many-side table to look up value from one-side
  - Used to pull data from lookup table into primary table

```
Sales Region = RELATED(SalesRegions[SalesRegion])
```

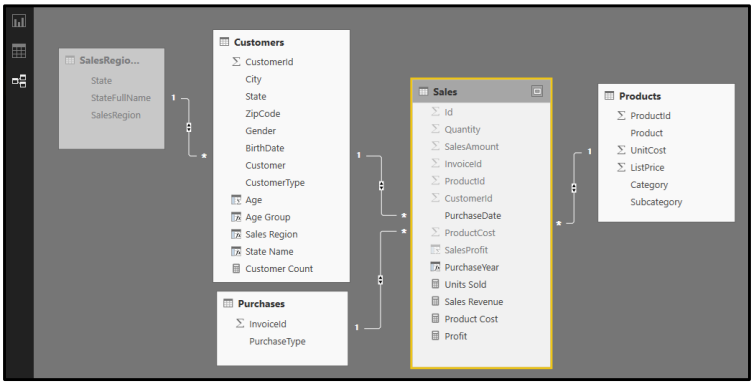
CustomerId	City	State	ZipCode	Gender	BirthDate	Customer	CustomerType	Age	Age Group	Sales Region
55	San Jose	CA	95110	Female	3/10/49	Jewell Ryan	Repeat Customer	66	Ages 65 and over	Western Region
73	San Jose	CA	95123	Male	5/9/85	Granville Perry	Repeat Customer	30	Ages 30 TO 39	Western Region
74	San Jose	CA	95122	Female	6/19/79	Sheri Mercado	Repeat Customer	36	Ages 30 TO 39	Western Region
78	San Jose	CA	95110	Male	6/16/78	Raleigh Olson	Repeat Customer	37	Ages 30 TO 39	Western Region
136	San Jose	CA	95124	Female	1/2/45	Carrie Foreman	Repeat Customer	70	Ages 65 and over	Western Region
150	San Jose	CA	95134	Female	8/11/84	Renee McMillan	Repeat Customer	31	Ages 30 TO 39	Western Region

```
State Name = RELATED(SalesRegions[StateFullName])
```

State	ZipCode	Gender	BirthDate	Customer	CustomerType	Age	Age Group	Sales Region	State Name
CA	95110	Female	3/10/49	Jewell Ryan	Repeat Customer	66	Ages 65 and over	Western Region	California
CA	95123	Male	5/9/85	Granville Perry	Repeat Customer	30	Ages 30 TO 39	Western Region	California
CA	95122	Female	6/19/79	Sheri Mercado	Repeat Customer	36	Ages 30 TO 39	Western Region	California
CA	95110	Male	6/16/78	Raleigh Olson	Repeat Customer	37	Ages 30 TO 39	Western Region	California
CA	95124	Female	1/2/45	Carrie Foreman	Repeat Customer	70	Ages 65 and over	Western Region	California
CA	95134	Female	8/11/84	Renee McMillan	Repeat Customer	31	Ages 30 TO 39	Western Region	California

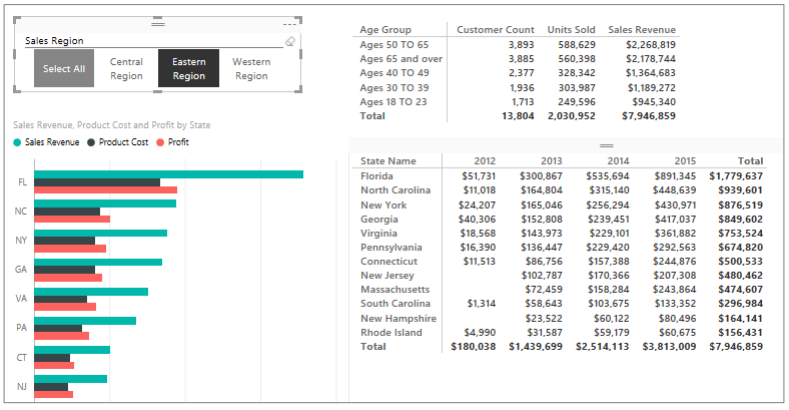
Hiding the Lookup Table

- Lookup table can often be hidden
  - simplifies reporting for data model consumers



Filtering on Sales Region

- Calculated column used to calculate slicer values



Age Group	Customer Count	Units Sold	Sales Revenue
Ages 50 TO 65	3,893	588,629	$2,268,819
Ages 65 and over	3,885	560,398	$2,178,744
Ages 40 TO 49	2,377	328,342	$1,364,683
Ages 30 TO 39	1,936	303,987	$1,189,272
Ages 18 TO 23	1,713	249,596	$945,340
Total	**13,804**	**2,030,952**	**$7,946,859**

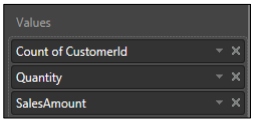
State Name	2012	2013	2014	2015	Total
Florida	$51,731	$300,867	$535,694	$891,345	$1,779,637
North Carolina	$11,018	$164,804	$315,140	$448,639	$939,601
New York	$24,207	$165,046	$256,294	$430,971	$876,519
Georgia	$40,306	$152,808	$239,451	$417,037	$849,602
Virginia	$18,568	$143,973	$229,101	$361,882	$753,524
Pennsylvania	$16,390	$136,447	$229,420	$292,563	$674,820
Connecticut	$11,513	$86,756	$157,388	$244,876	$500,533
New Jersey		$102,787	$170,366	$207,308	$480,462
Massachusetts		$72,459	$158,284	$243,864	$474,607
South Carolina	$1,314	$58,643	$103,675	$133,352	$296,984
New Hampshire		$23,522	$60,122	$80,496	$164,141
Rhode Island	$4,990	$31,587	$59,179	$60,675	$156,431
Total	$180,038	$1,439,699	$2,514,113	$3,813,009	$7,946,859

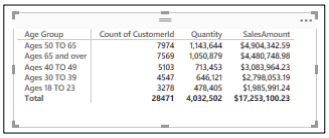
Agenda

- ✓ Data Modeling with PowerPivot and DAX
- ✓ Creating Calculated Columns
- ✓ Integrating Lookup Tables
- ➢ Creating Measures

Benefits of Measures over Calculated Columns

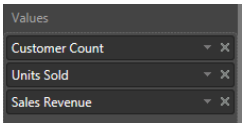
- Calculated columns can be aggregated in visual
  - However, aggregation details are stored in visual
  - Visual doesn't offer control over name and formatting

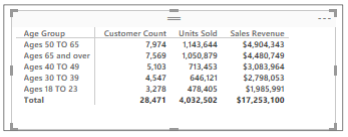




Age Group	Count of CustomerId	Quantity	SalesAmount
Ages 50 TO 65	7974	1,143,644	$4,904,342.59
Ages 65 and over	7569	1,050,879	$4,480,748.98
Ages 40 TO 49	5103	713,453	$3,083,964.23
Ages 30 TO 39	4547	646,121	$2,798,053.19
Ages 18 TO 23	3278	478,405	$1,985,991.24
Total	28471	4,032,502	$17,253,100.23

- Measure defines name, aggregation and formatting
  - Work is done once and reused across many visuals
  - Makes data model more fool-proof for report designers

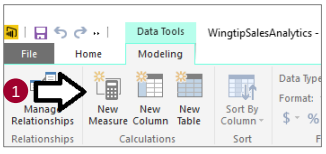


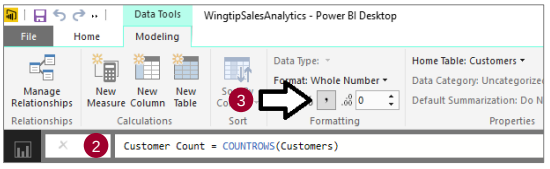


Age Group	Customer Count	Units Sold	Sales Revenue
Ages 50 TO 65	7,974	1,143,644	$4,904,343
Ages 65 and over	7,569	1,050,879	$4,480,749
Ages 40 TO 49	5,103	713,453	$3,083,964
Ages 30 TO 39	4,547	646,121	$2,798,053
Ages 18 TO 23	3,278	478,405	$1,985,991
Total	28,471	4,032,502	$17,253,100

Creating Measures

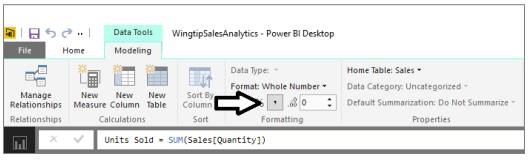
- Measures have advantage over calculated columns
  - They are evaluated based on the current evaluation context
- Creating a measure with Power BI Desktop
  1. Click New Measure button
  2. Give measure a name and write DAX expressions
  3. Configure formatting



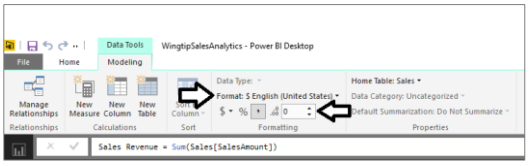


Formatting Measures

- Format as whole number



- Format as currency



Summary

- ✓ Data Modeling with PowerPivot and DAX
- ✓ Creating Calculated Columns
- ✓ Integrating Lookup Tables
- ✓ Creating Measures