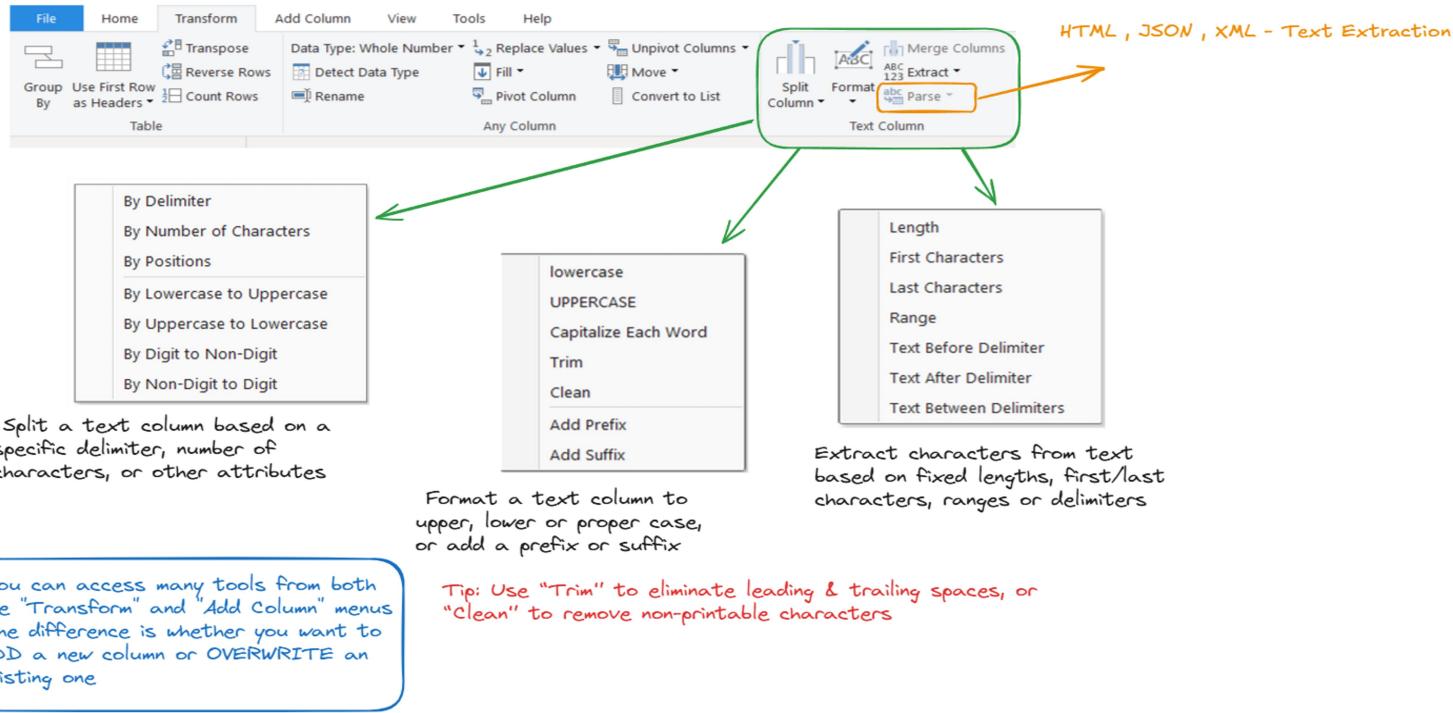


Power BI - Data Transformation - p2 - Lecture 3

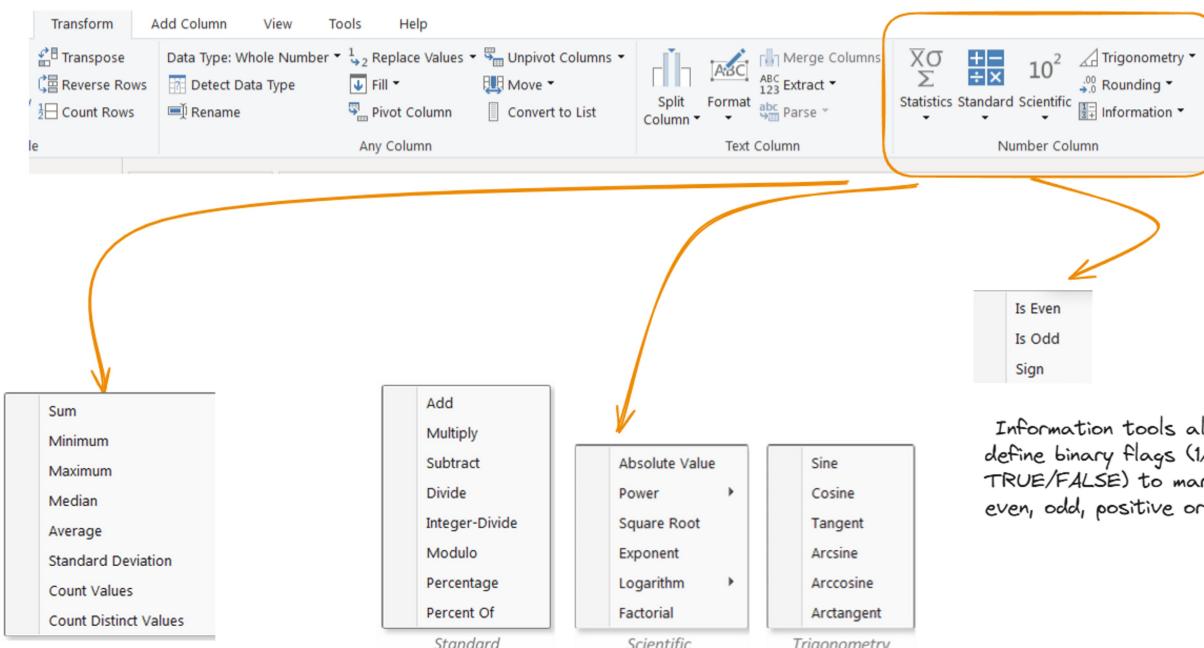
TEXT TOOLS



Text Tool : Assignment

1. Duplicate the email address column and name it "Domain Name"
2. In the new column, remove all text/characters except for the domain name
3. Use transformation steps to clean up and capitalize the domain names (i.e. "Adventure Works")
4. Save & Apply changes

NUMERICAL TOOLS



Statistics functions allow you to evaluate basic stats for a selected column (sum, min/max, average, count, count distinct, etc.)

Note: These tools return a SINGLE value, and are commonly used to explore a table rather than prepare it for loading

Standard, Scientific and Trigonometry tools allow you to apply standard operations (addition, multiplication, division, etc.) or more advanced calculations (power, logarithm, sine, tangent, etc.) to each value in a column

Note: Unlike the Statistics tools, these are applied to each row in the table

Information tools allow you to define binary flags (1/0 or TRUE/FALSE) to mark rows as even, odd, positive or negative

NUMERICAL TOOLS : Assignment

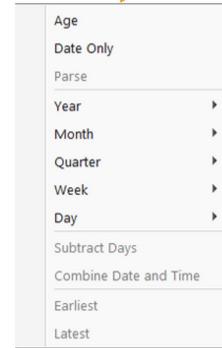
1. What is our average product cost? 413.66
2. How many colors do we sell our products in? 10
3. How many distinct customers do we have? 18018
4. What is the maximum annual customer income? 170000
5. Return the tables to their original state

DATE & TIME TOOLS

The screenshot shows the Power BI ribbon with the 'Tools' tab selected. In the 'From Date & Time' section, there are several options: 'Date', 'Time', 'Duration', and 'From Date & Time'. The 'From Date & Time' option is highlighted with a yellow box.

Date & Time tools are relatively straight-forward, and include the following options:

- Age: Difference between the current date and the date in each row
- Date Only: Removes the time component from a date/time field
- Year/Month/Quarter/Week/Day: Extracts individual components from a date field (time specific options include Hour, Minute, Second, etc.)
- Earliest/Latest: Evaluates the earliest or latest date from a column as a single value (can only be accessed from the "Transform" menu)



Today() - YEAR("19-01-2025")
2025 - YEAR(BirthDate)
2025 - 1991 = Age

TimeStamp - Instagram :
Create a post - DATE TIME
19-01-2025 11:11:11

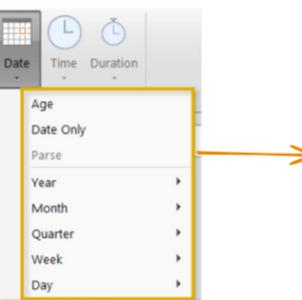
NOW() - Timestamp :
- 19-01-2025 8:58:52

Note: You will almost always want to perform these operations from the "Add Column" menu to build out new fields, rather than transforming an individual date/time column

PRO TIP: Load up a table containing a single date column and use Date tools to build out an entire calendar table

CREATING A CALENDAR TABLE

	Date
1	1/1/2020
2	1/2/2020
3	1/3/2020
4	1/4/2020
5	1/5/2020
6	1/6/2020
7	1/7/2020
8	1/8/2020
9	1/9/2020
10	1/10/2020
11	1/11/2020
12	1/12/2020
13	1/13/2020
14	1/14/2020
15	1/15/2020
16	1/16/2020
17	1/17/2020
18	1/18/2020
19	1/19/2020
20	1/20/2020
21	1/21/2020
22	1/22/2020
23	1/23/2020
24	1/24/2020
25	1/25/2020
26	1/26/2020
27	1/27/2020
28	1/28/2020



Use the Date options in the Add Column menu to quickly build out an entire calendar table from a list of dates

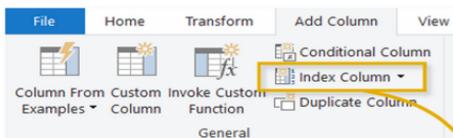
	Date	Day Name	Start of Week	Start of Month	Month Name
1	1/1/2020	Wednesday	12/29/2019	1/1/2020	January
2	1/2/2020	Thursday	12/29/2019	1/1/2020	January
3	1/3/2020	Friday	12/29/2019	1/1/2020	January
4	1/4/2020	Saturday	12/29/2019	1/1/2020	January
5	1/5/2020	Sunday	1/5/2020	1/1/2020	January
6	1/6/2020	Monday	1/5/2020	1/1/2020	January
7	1/7/2020	Tuesday	1/5/2020	1/1/2020	January
8	1/8/2020	Wednesday	1/5/2020	1/1/2020	January
9	1/9/2020	Thursday	1/5/2020	1/1/2020	January
10	1/10/2020	Friday	1/5/2020	1/1/2020	January
11	1/11/2020	Saturday	1/5/2020	1/1/2020	January
12	1/12/2020	Sunday	1/12/2020	1/1/2020	January
13	1/13/2020	Monday	1/12/2020	1/1/2020	January
14	1/14/2020	Tuesday	1/12/2020	1/1/2020	January
15	1/15/2020	Wednesday	1/12/2020	1/1/2020	January
16	1/16/2020	Thursday	1/12/2020	1/1/2020	January
17	1/17/2020	Friday	1/12/2020	1/1/2020	January
18	1/18/2020	Saturday	1/12/2020	1/1/2020	January
19	1/19/2020	Sunday	1/19/2020	1/1/2020	January
20	1/20/2020	Monday	1/19/2020	1/1/2020	January
21	1/21/2020	Tuesday	1/19/2020	1/1/2020	January
22	1/22/2020	Wednesday	1/19/2020	1/1/2020	January
23	1/23/2020	Thursday	1/19/2020	1/1/2020	January
24	1/24/2020	Friday	1/19/2020	1/1/2020	January
25	1/25/2020	Saturday	1/19/2020	1/1/2020	January
26	1/26/2020	Sunday	1/26/2020	1/1/2020	January
27	1/27/2020	Monday	1/26/2020	1/1/2020	January
28	1/28/2020	Tuesday	1/26/2020	1/1/2020	January

ASSIGNMENT: CALENDAR TABLES

Add the following columns to the calendar table:

1. Month Name (e.g. "January")
2. Month Number (e.g. "1")
3. Start of Year (e.g. "1/1/2020")
4. Year (e.g. "2020")

INDEX COLUMNS

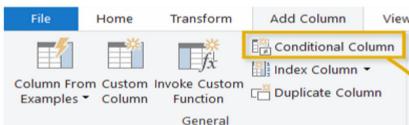


Index Columns contain a list of sequential values that can be used to identify each unique row in a table (typically starting from 0 or 1)

These are often used to create unique IDs that can be used to form relationships between tables (more on that later!)

i ² Index	Order Date	Stock Date	Avg Order Number	i ² Product Key
1	1/1/2020	9/21/2019	S045080	332
2	1/1/2020	12/5/2019	S045079	312
3	1/1/2020	10/29/2019	S045082	350
4	1/1/2020	11/16/2019	S045081	338
5	1/2/2020	12/15/2019	S045083	312
6	1/2/2020	10/12/2019	S045084	310
7	1/2/2020	12/18/2019	S045086	314
8	1/2/2020	10/9/2019	S045085	312
9	1/3/2020	10/3/2019	S045093	312
10	1/3/2020	9/29/2019	S045090	310
11	1/3/2020	12/11/2019	S045088	345
12	1/3/2020	10/24/2019	S045092	312
13	1/3/2020	12/16/2019	S045089	351
14	1/3/2020	10/26/2019	S045091	314
15	1/3/2020	9/11/2019	S045087	350
16	1/3/2020	9/11/2019	S045094	310
17	1/4/2020	10/30/2019	S045096	312
18	1/4/2020	10/30/2019	S045097	313
19	1/4/2020	9/15/2019	S045098	310
20	1/4/2020	12/7/2019	S045095	344

CONDITIONAL COLUMNS



Conditional Columns allow you to define new fields based on logical rules and conditions (IF/THEN statements)

Here we're creating a conditional column named Quantity Type, which is based on Order Quantity:

- If Order Quantity = 1, Quantity Type = "Single Item"
- Else If Order Quantity > 1, Quantity Type = "Multiple Items"
- Else; Quantity Type = "Other"

Add Conditional Column

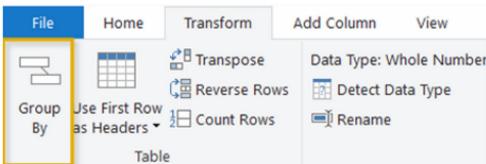
Add a conditional column that is computed from the other columns or values.

New column name:

Column Name	Operator	Value	Output
If	Order Quantity	equals	ABC
Else If	Order Quantity	is greater than	ABC
		1	Single Item
		1	Multiple Items
... Add Clause			
Else	ABC	Other	

OK Cancel

GROUPING & AGGREGATING



Group By allows you to aggregate data at a different level (i.e. group daily records into monthly, aggregate transactions by store, etc.)

	Order Date	Product Key	Customer Key	Order Quantity
1	6/25/2022	214	14719	1
2	10/8/2021	214	21990	1
3	12/30/2021	214	22098	1
4	6/29/2022	214	22748	1
5	8/16/2021	214	27821	1
6	10/9/2021	214	15685	1
7	8/9/2021	214	14951	1
8	1/19/2022	214	23101	1
9	9/23/2021	214	17158	1
10	1/19/2022	214	24198	1
11	6/29/2022	214	12963	1
12	9/13/2021	214	12715	1
13	10/2/2021	214	14846	1
14	7/31/2021	214	11290	1
15	11/24/2021	214	22103	1
16	8/1/2021	214	16982	1
17	10/12/2021	214	20410	1
18	9/10/2021	214	14217	1
19	10/22/2021	214	19642	1
20	8/11/2021	214	11666	1

Group By

Specify the column to group by and the desired output.

Basic Advanced

Product Key	New column name	Operation	Column
TotalQuantity	Sum	Order Quantity	

OK Cancel

	Product Key	TotalQuantity
1	214	2099
2	215	1940
3	220	1995
4	223	4151
5	226	392
6	229	408
7	232	424
8	235	381
9	310	169
10	311	139
11	312	179
12	313	168
13	314	157
14	320	65
15	322	39
16	324	72
17	326	65

Here we're transforming a daily, transaction-level table into a summary of Total Quantity by Product Key

NOTE: Any fields not specified in the Group By settings are lost

The screenshot shows a data transformation interface. On the left is a transaction-level table with columns: Order Date, Product Key, Customer Key, and Order Quantity. On the right is a 'Group By' dialog box. The dialog has sections for 'Basic' and 'Advanced' grouping, with 'Advanced' selected. It shows 'Product Key' and 'Customer Key' as grouped columns, and 'Order Quantity' as the column being summarized. A green arrow points from the bottom-left of the summary table towards the 'Group By' dialog.

	Order Date	Product Key	Customer Key	Order Quantity
1	6/25/2022	214	14719	1
2	10/8/2021	214	21990	1
3	12/30/2021	214	22098	1
4	6/29/2022	214	22748	1
5	8/16/2021	214	27821	1
6	10/9/2021	214	15685	1
7	8/9/2021	214	14951	1
8	1/19/2022	214	23101	1
9	9/23/2021	214	17158	1
10	1/19/2022	214	24196	1
11	6/29/2022	214	12963	1
12	9/13/2021	214	12715	1
13	10/2/2021	214	14846	1
14	7/31/2021	214	11290	1
15	11/24/2021	214	22103	1
16	8/1/2021	214	16982	1
17	10/12/2021	214	20410	1
18	9/10/2021	214	14217	1
19	10/22/2021	214	19642	1
20	8/11/2021	214	11666	1

The screenshot shows a summary table with columns: Product Key, Customer Key, and TotalQuantity. The data is identical to the transaction-level table above, grouped by Product Key and Customer Key. A green arrow points from the bottom-left of the summary table towards the 'Group By' dialog.

	Product Key	Customer Key	TotalQuantity
1	214	19356	1
2	214	15101	1
3	214	12473	1
4	214	12963	1
5	214	26986	1
6	214	13202	1
7	214	14951	1
8	214	11201	1
9	214	19538	1
10	214	22749	1
11	214	15815	1
12	214	19252	1
13	214	14849	1
14	214	11290	1
15	214	27851	1
16	214	16982	1
17	214	21863	1
18	214	19725	1
19	214	15684	1
20	214	11666	1
21	214	26941	1

This time we're transforming the daily, transaction-level table into a summary of Total Quantity grouped by both Product Key and Customer Key (using the "Advanced" option)

NOTE: This is like creating a PivotTable in Excel and pulling in Sum of Order Quantity with Product Key and Customer Key as row labels