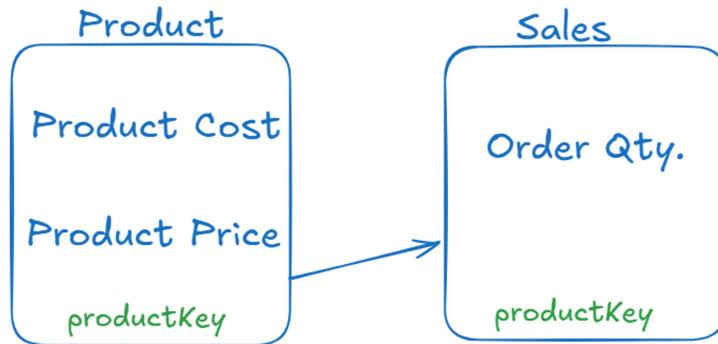


## Power BI - Advance Transformation + Data Modelling - Lecture 4



$\text{Sales} = \text{product Price} * \text{Order Qty.}$

$\text{Expenditure} = \text{Product Cost} * \text{Order Qty.}$

$\text{Profit} = \text{Sales} - \text{Expenditure}$

Matrix Visual - Pivot Table [Product Vs Profit]

$\text{Revenue} = \text{SUM}(\text{Sales})$

## PIVOTING & UNPIVOTING

Pivoting describes the process of turning distinct row values into columns, and unpivoting describes the process of turning distinct columns into rows

Date	A <sup>0</sup> C Product Category	1 <sup>2</sup> 3 North Region	1 <sup>2</sup> 3 Central Region	1 <sup>2</sup> 3 South Region
1	7/1/2022 Bikes	10	19	25
2	7/1/2022 Components	14	31	16
3	7/1/2022 Clothing	35	32	46

Date	A <sup>0</sup> C Product Category	1 <sup>2</sup> 3 Region	1 <sup>2</sup> 3 Quantity Sold
1	7/1/2022 Bikes	North Region	10
2	7/1/2022 Bikes	Central Region	19
3	7/1/2022 Bikes	South Region	25
4	7/1/2022 Components	North Region	14
5	7/1/2022 Components	Central Region	31
6	7/1/2022 Components	South Region	16
7	7/1/2022 Clothing	North Region	35
8	7/1/2022 Clothing	Central Region	32
9	7/1/2022 Clothing	South Region	46

Imagine the table on a hinge; pivoting rotates it from vertical to horizontal, and unpivoting rotates it from horizontal to vertical

NOTE: Transpose works very similarly, but doesn't recognize unique values; instead, the entire table is transformed so that each row becomes a column and vice versa

## MERGING QUERIES

The screenshot shows the 'Merge' dialog box in Power BI. On the left is a navigation pane with options: 'Merge Queries' (selected), 'Append Queries', 'Combine Files', and 'Combine'. The main area has a title 'Merge' and a subtitle 'Select a table and matching columns to create a merged table.' Below this are two tables: 'Sales Data' and 'Product Lookup'. The 'Product Key' column is highlighted in both tables and selected for merging. The 'Join Kind' dropdown is set to 'Left Outer (all from first, matching from second)'. A checkmark indicates 'The selection matches 56046 of 56046 rows from the first table.' At the bottom are 'OK' and 'Cancel' buttons.

Merging queries allows you to join tables based on a common column (like a lookup in Excel)

In this case we're merging the Sales Data table with the Product Lookup table, which share a common Product Key column

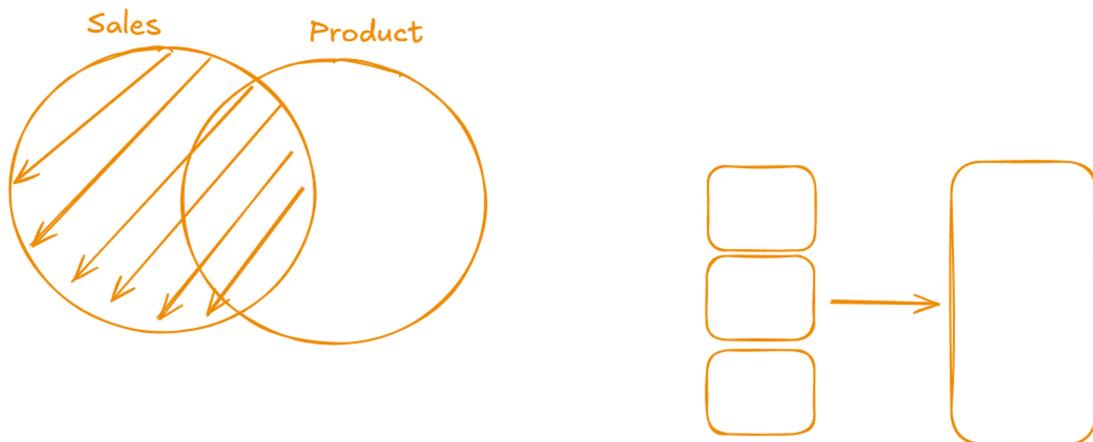
NOTE: Merging adds columns to an existing table/query

Just because you can merge tables, doesn't mean you should!  
In many cases, it's better to keep tables separate and define relationships between them in the data model (more on that soon!)

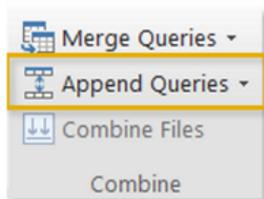
The screenshot shows a Power BI report with a table containing data and validation statistics. The table has three columns: ProductKey, ProductName, and ProductColor. The validation statistics at the top of each column are:

Column	Valid	Error	Empty
ProductKey	100%	0%	0%
ProductName	100%	0%	0%
ProductColor	100%	0%	0%

A blue arrow points from the text 'Table.Column.Name' to the 'ProductColor' column header. The table data includes rows for various products like 'Road Tire Tube' and 'Sport-100 Helmet, Red'.



## APPENDING QUERIES



Appending queries allows you to combine or stack tables sharing the exact same column structure and data types

### Append

Concatenate rows from two tables into a single table.

Two tables  Three or more tables

First table

AdventureWorks Sales Data 2020

Second table

AdventureWorks Sales Data 2021

Here we're appending the Adventure Works Sales 2020 table to the Adventure Works Sales 2021 table, which is valid since they share identical table structures.

**NOTE:** Appending adds rows to an existing table/query

Sales Data 2020
Sales Data 2021
Sales Data 2022
Product Category Sales (...)
<b>Sales Data 2020 - 2022</b>

Let's Bring Territory Lookup Table & Return Data

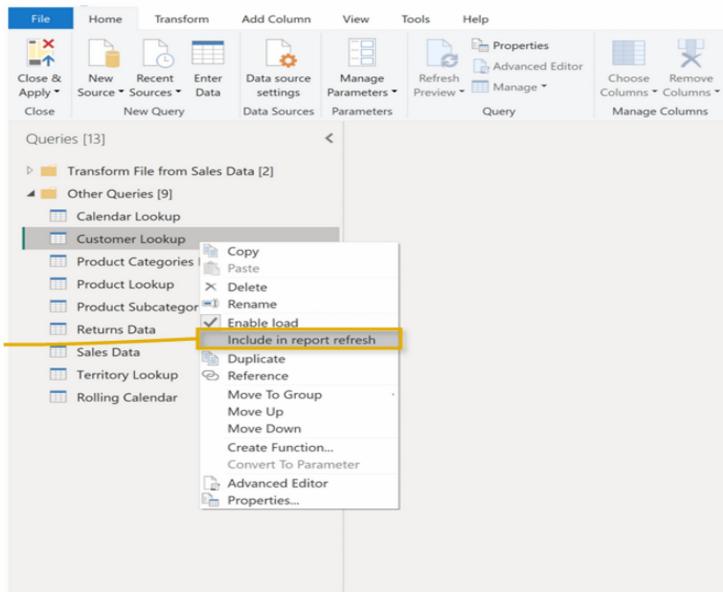
## REFRESHING QUERIES



- By default, all queries will refresh when you use the Refresh command from the Home tab

From the Query Editor, uncheck Include in report refresh to exclude individual queries from the refresh

**PRO TIP:** Exclude queries from refresh that don't change often (like lookups or static data tables)



## POWER QUERY BEST PRACTICES

Get organized before connecting and loading data

- Define clear and intuitive table/query names from the start, and establish an organized file/folder structure if you are working with local flat files to avoid changes to file names or paths.

Disable report refresh for any static data sources

- There's no need to constantly refresh data sources that don't change, like lookups or static data tables

When working with large tables, only load the data you need

- Don't include hourly data when you only need daily, or transaction-level data when only need a product-level summary (extra data will only slow your report down!)

## CREATING A DATA MODEL

In this section we'll cover foundational data modeling topics like normalization, fact and dimension tables, primary and foreign keys, relationship cardinality and filter flow

### TOPICS WE'LL COVER:

1. Data Modeling 101
2. Facts & Dimensions
3. Cardinality
4. Common Schemas
5. Normalization
6. Primary & Foreign Keys
7. Filter Flow
8. Hierarchies

### GOALS FOR THIS SECTION:

- Understand the basic principles of data modeling, including normalization, fact & dimension tables and common schemas.
- Create table relationships using primary and foreign keys, and discuss different types of relationship cardinality
- Configure report filters and trace filter context as it flows between related tables in the model
- Explore data modeling options like hierarchies, data categories and hidden fields

## WHAT IS A DATA MODEL?

The screenshot shows a data visualization interface with three separate tables:

- Product Lookup**: Fields: Model Name, Product Color, Product Cost, Product Description, Product Key, Product Name.
- Sales Data**: Fields: Customer Key, Index, Order Date, Order Line Item, Order Number, Order Quantity, Product Key, Stock Date, Territory Key.
- Returns Data**: Fields: Product Key, Return Date, Return Quantity, Territory Key.

This IS NOT a data model 😞

- This is a collection of independent tables, which share no connections or relationships
- If you tried to visualize Orders and Returns by Product, this is what you'd get

ProductName	OrderQuantity	ReturnQuantity
All-Purpose Bike Stand	84,174	1,828
AWC Logo Cap	84,174	1,828
Bike Wash - Dissolver	84,174	1,828
Cable Lock	84,174	1,828
Chain	84,174	1,828
Classic Vest, L	84,174	1,828
Classic Vest, M	84,174	1,828
Classic Vest, S	84,174	1,828
Fender Set - Mountain	84,174	1,828
<b>Total</b>	<b>84,174</b>	<b>1,828</b>

The screenshot shows the same three tables as before, but with a connection established between them:

- Product Lookup**: Fields: Product Key, Product Name, Product Price, Product S K U.
- Sales Data**: Fields: Customer Key, Index, Order Date, Order Line Item, Order Number, Order Quantity, **Product Key**.
- Returns Data**: Fields: Product Key, Return Date, Return Quantity, Territory Key.

A relationship diagram is shown between Sales Data and Returns Data, both connected to Product Lookup via their Product Key fields.

This IS a data model! 😊

- The tables are connected via relationships, based on a common field (Product Key)
- Now Sales and Returns data can be filtered using fields from the Product Lookup table!

ProductName	OrderQuantity	ReturnQuantity
All-Purpose Bike Stand	234	8
AWC Logo Cap	4,151	46
Bike Wash - Dissolver	1,706	25
Classic Vest, L	182	4
Classic Vest, M	182	7
Classic Vest, S	157	8
Fender Set - Mountain	3,960	54
Half-Finger Gloves, L	840	18
Half-Finger Gloves, M	918	16
<b>Total</b>	<b>84,174</b>	<b>1,828</b>

