Power BI - Advanced

Grant Shannon

Nov 2024

Housekeeping

- Student check
- Installation and environment setup
  - Download install PBI desktop client
  - Connect to github repo
    - https://github.com/gpsuser/PBI
  - Get data files
    - https://github.com/microsoft/powerbi-desktop-samples/blob/main/DAX/Adventure%20Works%20DW%202020.pbix
- 10 min break on the hour (time permitting)
- Hands on course

Previously

- Data Modelling
  - Dimensional Modelling
  - Star Schema
- DAX
- Measures
- Report Generation

Agenda

- Introduction
- Data Modelling
  - Recap: Business motivation behind dimensional modelling
  - Revisit Star Schema
  - Connecting to a Table as a *Copy* vs. *Reference*
    - Update implications
  - Implement Star Schema from a single raw data table
    - Fact Table
    - Dimension Tables
    - Relationships
  - Types of Joins
  - Enriching DIM table with *Merge* (join)
- Aggregation inside Dimensional Model - without *DAX*
- Introduction to *M-language*
- Conclusion

Introduction

Power BI - recap

- Key Features:
  - **Data Connectivity**: Connects to a wide range of data sources, including Excel, databases, and cloud services
  - **Data Transformation**: Clean and transform data with Power QueryIt helps you connect, prepare, model, visualize, and share data from various sources.
  - **Data Modeling**: Create relationships between data sets and build complex models.
  - **Visualizations**: Create interactive reports and dashboards with a variety of visual tools.
  - **Sharing and Collaboration**: Share insights with your team through Power BI Service and mobile apps.

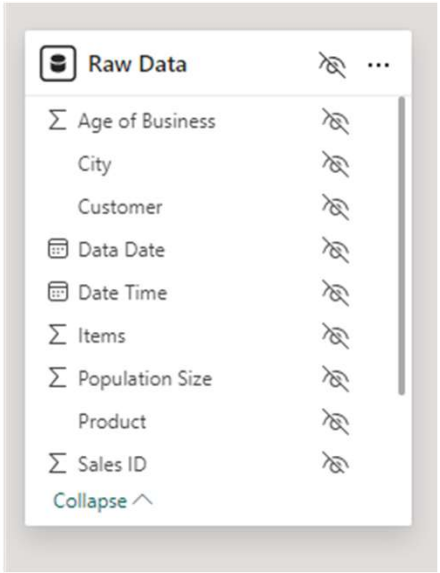
Data Modelling

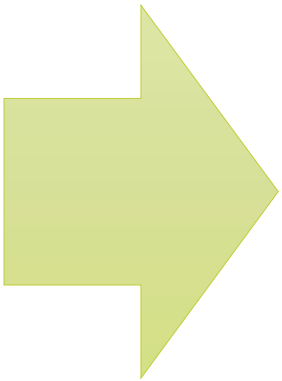
Dimensional Modelling Re-cap

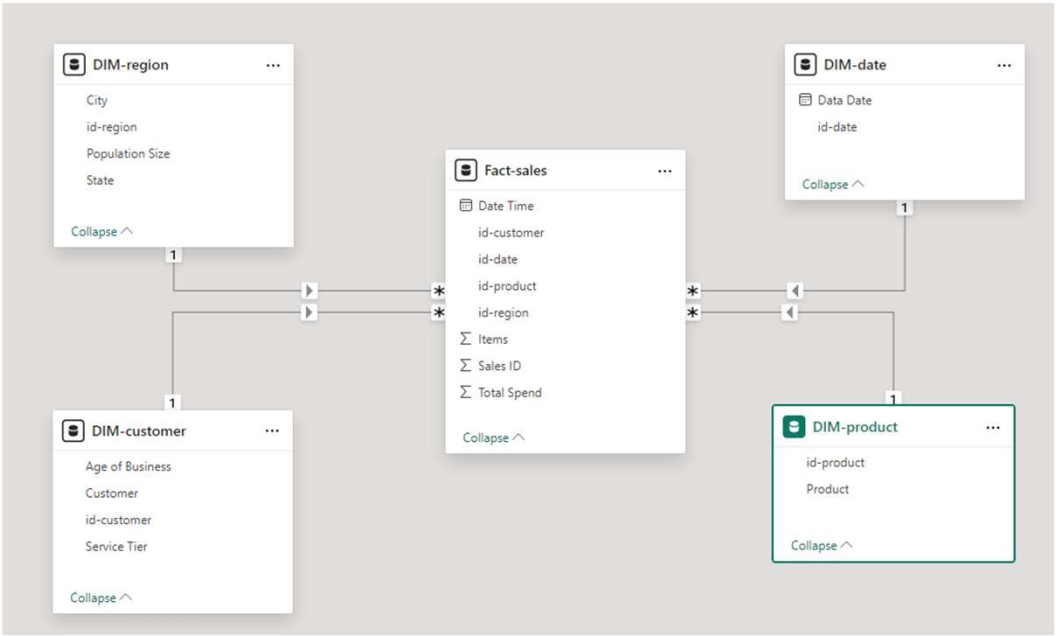
- *Dimensions* hold descriptive data
- *Fact* table holds quantitative / event driven data
  - Contain foreign keys that connect to DIM tables
    - Connections implemented outside of Query Editor – *Model View*

Building a dimensional model

Using a single raw data table







Getting Started

Table as reference

- To get started with our dimensional model – first get the *Raw Data* table as a reference

References vs Duplicates

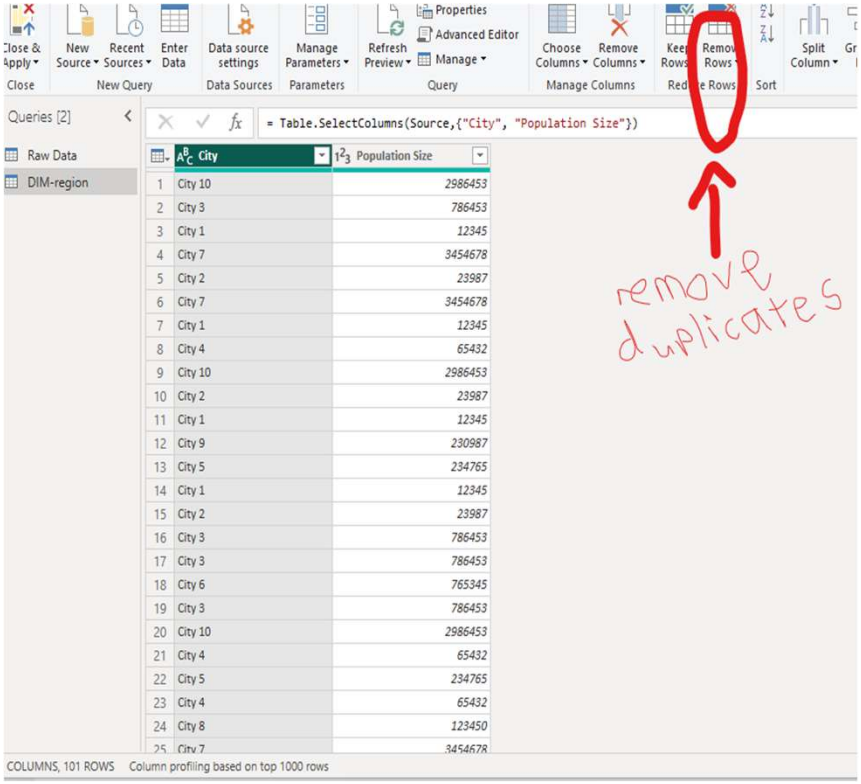
- Table *References* create a link to a table
  - Changes in the original table are recorded in the original table (reference table automatically inherits these changes).
- Table *Duplicates* – creates a copy with no links
  - Copied table includes all the changes (applied steps) in the original table.
  - Changes/updates to the original table do not affect the duplicate table.
- If changes to the original query/table need to reflect in the copied table – then must create a reference table and not a duplicate table.

Creating a region Dimension table

DIM-region

Open Query Editor as follows:

- Home > Transform Data > Transform Data >
- Right click on *Raw Data* Table : reference
- Change name of reference table
  > Right click on the *Raw Data(2)* :
  > rename to *DIM-region*
- Choose columns for the *DIM-region* table
  > Choose Columns : Choose Columns
  > *City, Population Size*
- Remove duplicate rows :
  - Make sure *DIM-region* (Queries) selected
  - Select the *City* Column
    > Remove Rows > Remove Duplicates
- Add index column (make sure *DIM-region* (Queries) still selected)
  - Add Colum menu > Index Column > From 1
  - Click on *Index* column and move it to front
  - Home > Close and Apply: close and apply
- Rename index column:
  - Table view
    > Select *DIM-region*
    > right click on *Index* column
    > rename: *id-region*



Investigating Data Updates

Impact on DIM Tables

- Update the Raw Data table
- Monitor its impact on the *DIM-region* table (as *Reference* Table)

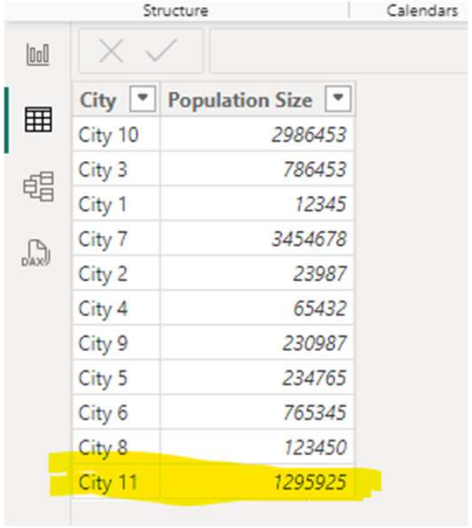
Change Raw Data

- Open *Raw Data.xlsx* and add row 101 from *extra row.xlsx*:
  - Ctrl C , Ctrl V , Ctrl Save
  - Close spreadsheets

	A	B	C	D	E	F	G	H	I	J	K
	Sales ID	Date Time	Items	otal Spen	Product	City	Customer	pulation S	e of Busin	ervice Tie	Data Date
	97	2023-12-11 22:07:31	13	167	Product D	City 6	Customer	765345	4	Platinum	2023-12-11
	98	2023-10-11 07:22:39	12	1775	Product C	City 10	Customer	2986453	8	Premium	2023-10-11
)	99	2023-07-09 11:21:51	15	632	Product C	City 5	Customer	234765	10	Platinum	2023-07-09
L	100	2023-06-28 02:51:11	12	3759	Product A	City 6	Customer	765345	15	Standard	2023-06-28
2	101	2023-06-16 18:20:31	9	6886	Product C	City 11	Customer	1295925	20	Platinum	2023-06-16

Check DIM table update

- Check *Raw Data.xlsx* change caused DIM-region update
  - Home > Refresh
  - Table View > Select DIM-region



- Go to Model View - and delete relationship

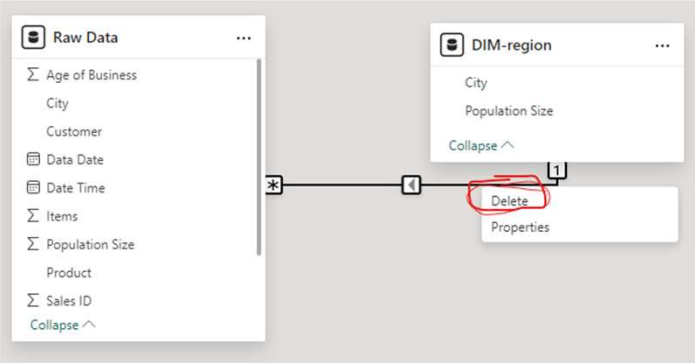


TABLE JOINS

- A table join in database terms is a method of combining rows from two or more tables based on a related column between them.
- This allows for the retrieval of data that is spread across multiple tables, making it easier to analyze and work with related information
- Consider:
  - Outer Joins
  - Anti Joins
  - Inner Joins

Outer Joins

Left

Table

ID	Sales
A	5
B	15
C	24

Right

Table

ID	Region
A	Europe
DD	Asia
E	America

1. Outer Left Join

Table

ID	Sales
A	5
B	15
C	24

2. Outer Full Join

Table

ID	Sales	Region
A	5	Europe
B	15	NULL
C	24	NULL
DD	NULL	Asia
E	NULL	America

3. Outer Right Join

Table

ID	Sales
A	5
DD	NULL
E	NULL

Anti Joins

Left

Table	
ID	**Sales**
A	5
B	15
C	24

Right

Table	
ID	**Region**
A	Europe
DD	Asia
E	America

4. Anti Left Join

Table	
ID	**Sales**
B	15
C	24

5. Anti Right Join

Table	
ID	**Region**
DD	Asia
E	America

Inner Joins

Left

Table	
ID	**Sales**
A	5
B	15
C	24

Right

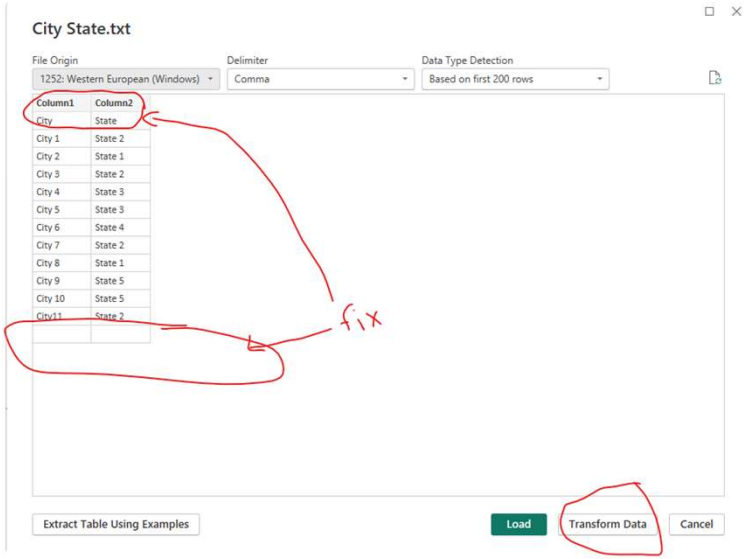
Table	
ID	**Region**
A	Europe
DD	Asia
E	America

6. Inner Join

Table		
ID	**Sales**	**Region**
A	5	Europe

Enriching DIM-region table

- Want to add City- State information to DIM-region from *City State.csv*
- Home > File > Get Data > Text CSV > *City State.csv* > Open > Transform



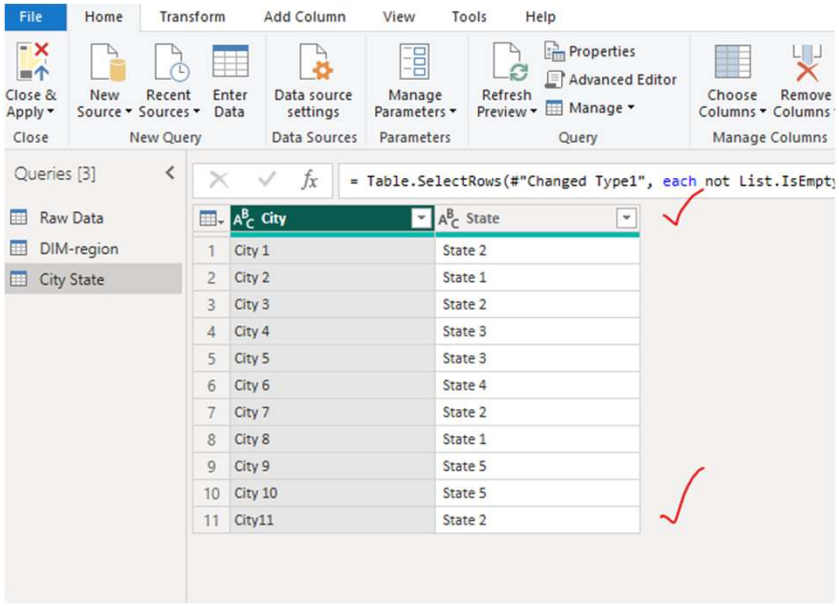
Enriching DIM-region table

- Home > File > Get Data > Text CSV > *City State.csv* > Open > Transform

> Use First Row as Headers: Use first rows as headers

> Remove Rows > Remove Blank Rows

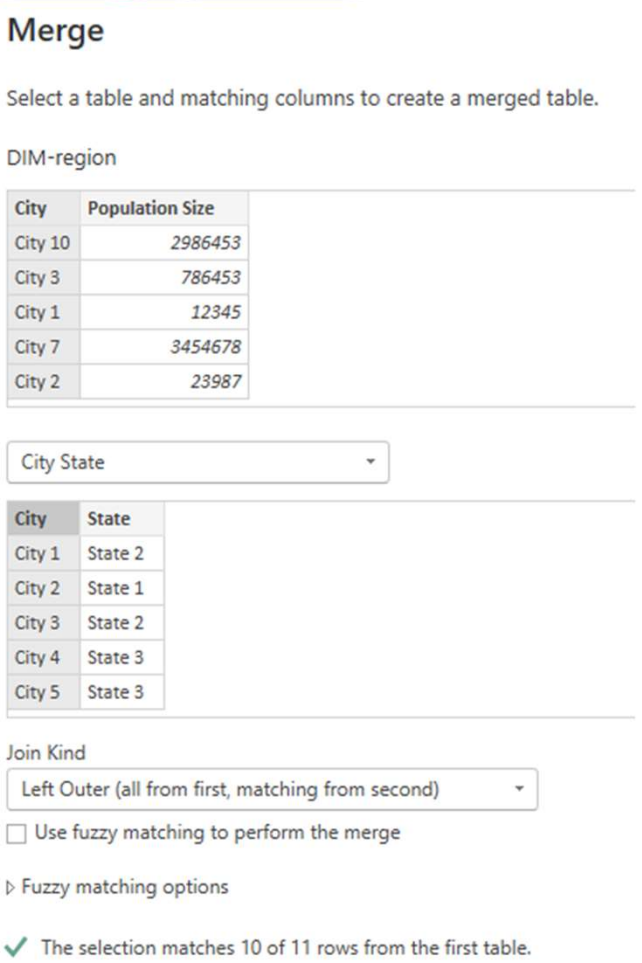
> Close and Apply : close % apply



Enriching DIM-region table

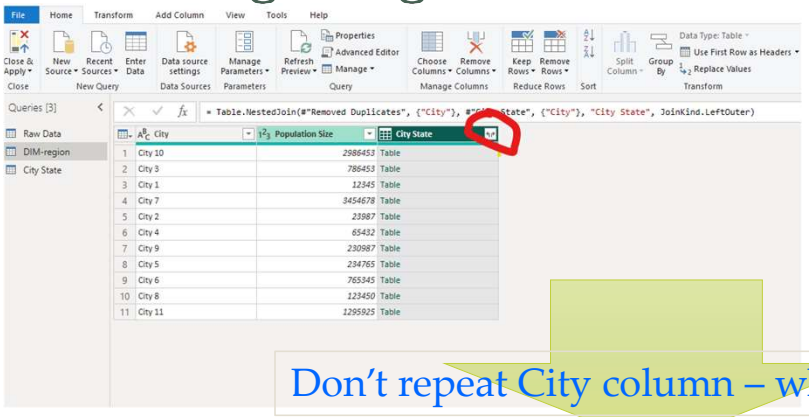
Merging

- Home > Transform Date: Transform data
> Select DIM-region (Query) > Merge Queries:
  - merge queries
- In Merge window
  > For DIM-region :
    > Select Select City column (to join/merge against)
  > For City State:
    > Select City State
    > Select City column
- Join Kind
  - Left outer
    - all from first table, matching from second table
- OK

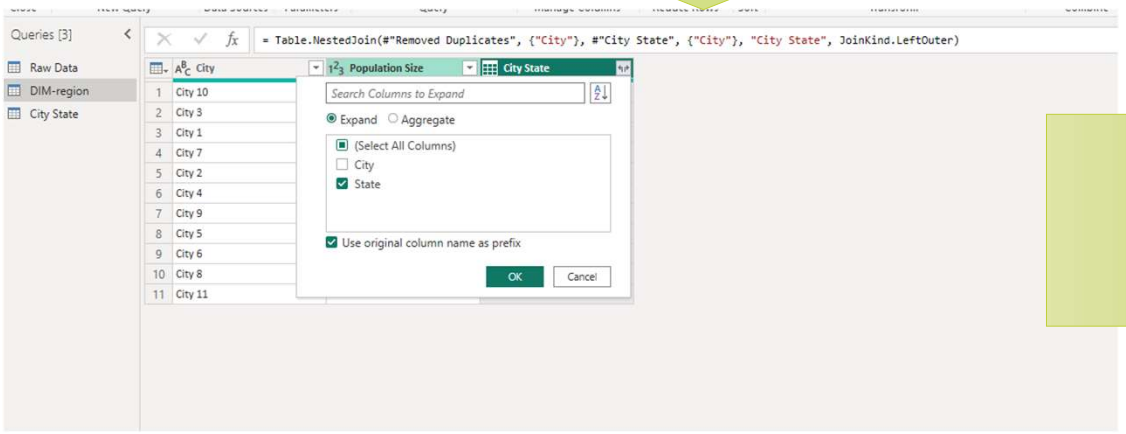


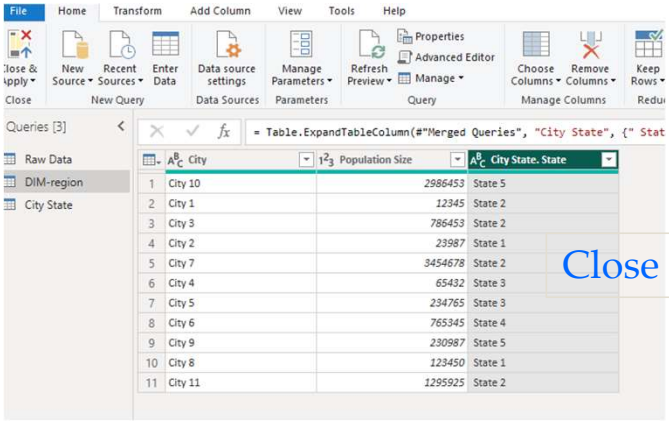
Enriching DIM-region table

Selecting merge columns



Don't repeat City column – when merge



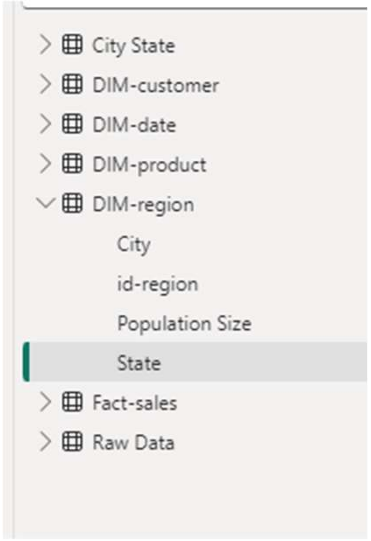


Close and Apply

Rename State Column

- Home > Table View

From *Data* blade > *Dim-region*
> Select *City State.State* + right click
> rename: *State*



Creating the remaining dimensions

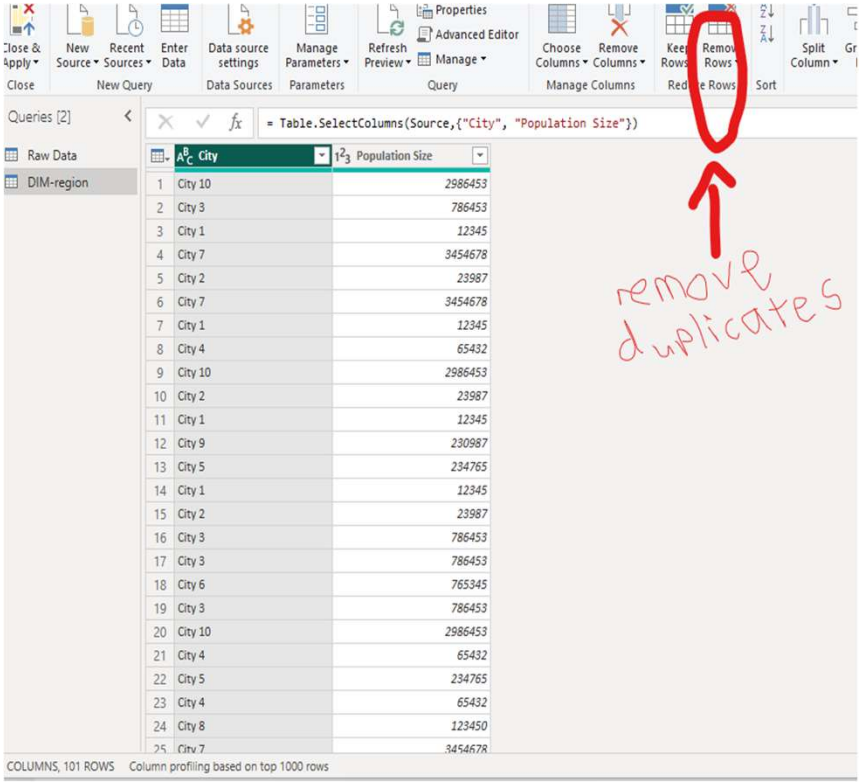
- *DIM-Customer*
- *DIM-Product*
- *DIM-Date*

Creating a customer Dimension table

DIM-customer

Open Query Editor as follows:

- Home > Transform Data > Transform Data >
- Right click on *Raw Data* Table : reference
- Change name of reference table
  > Right click on the *Raw Data(2)* :
  > rename to *DIM-customer*
- Choose columns for the *DIM-customer* table
  > Choose Columns : Choose Columns
  > *Customer, Age of Business, Service Tier*
- Remove duplicate rows :
  - Make sure *DIM-customer* (Queries) selected
  - Select the *Customer* Column
    > Remove Rows > Remove Duplicates
- Add index column (make sure *DIM-customer* (Queries) still selected)
  - Add Colum menu > Index Column > From 1
  - Click on *Index* column and move it to front
  - Home > Close and Apply: close and apply
- Rename index column:
  - Table view
    > Select *DIM-customer*
    > right click on *Index* column
    > rename: *id-customer*

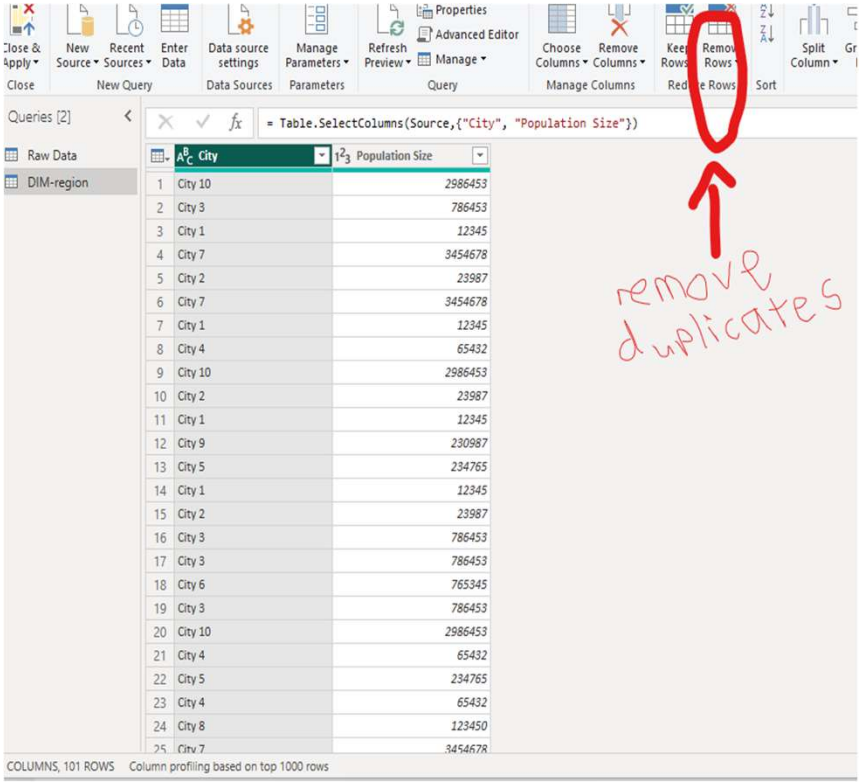


Creating a product Dimension table

DIM-product

Open Query Editor as follows:

- Home > Transform Data > Transform Data >
- Right click on *Raw Data* Table : reference
- Change name of reference table
  > Right click on the *Raw Data(2)* :
  > rename to *DIM-product*
- Choose columns for the *DIM-product* table
  > Choose Columns : Choose Columns
  > *Product*
- Remove duplicate rows :
  - Make sure *DIM-product* (Queries) selected
  - Select the *Product* Column
    > Remove Rows > Remove Duplicates
- Add index column (make sure *DIM-product* (Queries) still selected)
  - Add Colum menu > Index Column > From 1
  - Click on *Index* column and move it to front
  - Home > Close and Apply: close and apply
- Rename index column:
  - Table view
    > Select *DIM-product*
    > right click on *Index* column
    > rename: *id-product*

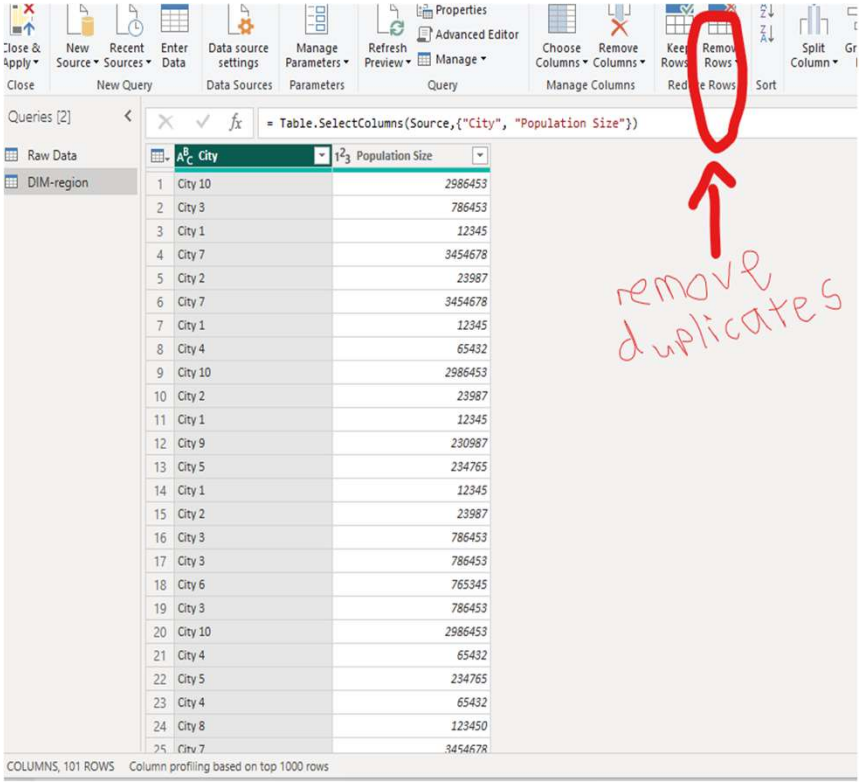


Creating a date Dimension table

DIM-date

Open Query Editor as follows:

- Home > Transform Data > Transform Data >
- Right click on *Raw Data* Table : reference
- Change name of reference table
  > Right click on the *Raw Data(2)* :
  > rename to *DIM-date*
- Choose columns for the *DIM-date* table
  > Choose Columns : Choose Columns
  > *Data Date*
- Remove duplicate rows :
  - Make sure *DIM-date* (Queries) selected
  - Select the *Data Date* Column
    > Remove Rows > Remove Duplicates
- Add index column (make sure *DIM-date* (Queries) still selected)
  - Add Colum menu > Index Column > From 1
  - Click on *Index* column and move it to front
  - Home > Close and Apply: close and apply
- Rename index column:
  - Table view
    > Select *DIM-date*
    > right click on *Index* column
    > rename: *id-date*



Creating a Fact table

Fact-sales

Open Query Editor as follows:

- Home > Transform Data > Transform Data >
- Right click on *Raw Data* Table : reference
- Change name of reference table
  > Right click on the *Raw Data(2)* :
  > rename to *FACT-sales*

- Next - Need to bring in ID columns from DIM tables

Merge product-id into Fact-sales

Using Query Editor - bring in *product-id*, using Merge - make sure *Fact-sales* (Queries) is selected
Merge Queries: Merge Queries

> Select *Product* column in *Fact-Sales*

> From dropdown choose: *Dim-product*
> Select *Product* column in *DIM-product*
> Left Outer Join
> OK

Expand *DIM-product* (click on two diverging arrows icon)
> Select expand
> Select *id-product*
> De-select *use original column name as prefix*
> *OK*
> Delete relevant *Product* columns in *Fact-sales*
> Select *Product* column
> Right-click: remove columns
> Close and Appl: Close and Apply

Merge customer-id into Fact-sales

Using Query Editor - bring in *customer-id*, using Merge - make sure *Fact-sales* (Queries) is selected
Merge Queries: Merge Queries

> Select *Customer* column in *Fact-Sales*

> From dropdown choose: *Dim-customer*
> Select *Customer* column in *DIM-customer*
> Left Outer Join
> OK

Expand *DIM-customer* (click on two diverging arrows icon)
> Select expand
> Select *id-customer*
> De-select *use original column name as prefix*
> *OK*
> Delete relevant *Customer* columns in *Fact-sales*
> Select *Customer + Ctrl +* other columns to delete (*Age of Business, Service Tier*)
> Right-click: remove columns

Merge region-id into Fact-sales

Using Query Editor - bring in *region-id*, using Merge - make sure *Fact-sales* (Queries) is selected
Merge Queries: Merge Queries

> Select *City* column in *Fact-Sales*

> From dropdown choose: *Dim-region*
> Select *City* column in *DIM-region*
> Left Outer Join
> OK

Expand *DIM-region* (click on two diverging arrows icon)
> Select expand
> Select *id-region*
> De-select *use original column name as prefix*
> *OK*
> Delete relevant *Region* columns in *Fact-sales*
> Select *City + Ctrl +* other to delete (*Population Size*)
> Right-click: remove columns

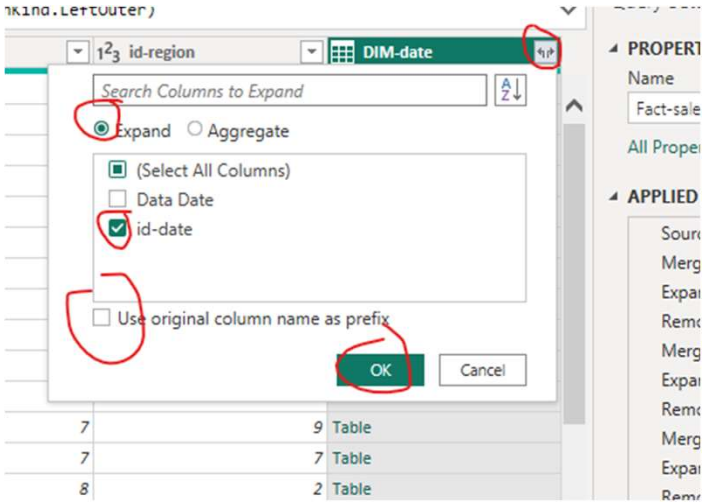
Merge date-id into Fact-sales

Using Query Editor - bring in *date-id*, using Merge - make sure *Fact-sales* (Queries) is selected
Merge Queries: Merge Queries

> Select *Data Date* column in *Fact-Sales*

> From dropdown choose: *Dim-date*
> Select *Data Date* column in *DIM-date*
> Left Outer Join
> OK

Expand *DIM-date* (click on two diverging arrows icon)
> Select expand
> Select *id-date*
> De-select *use original column name as prefix*
> *OK*
> Delete relevant *Date* columns in *Fact-sales*
> Select *Data Date*
> Right-click: remove

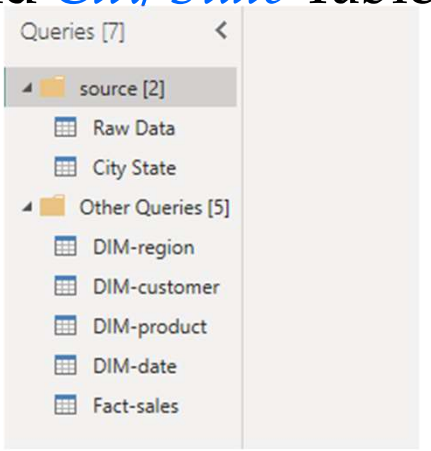


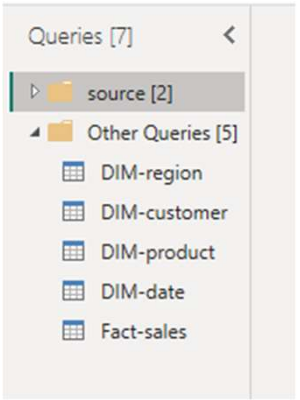
Create folder for source data

Home > Transform data : Transform data (takes you to: *Query Editor*)

- Right click on *Queries* panel > *New Group* > *Name*: *source*
- Click on and drag/Move *Raw Data* and *City State* Tables into *source* folder

> Close and Apply: close and apply

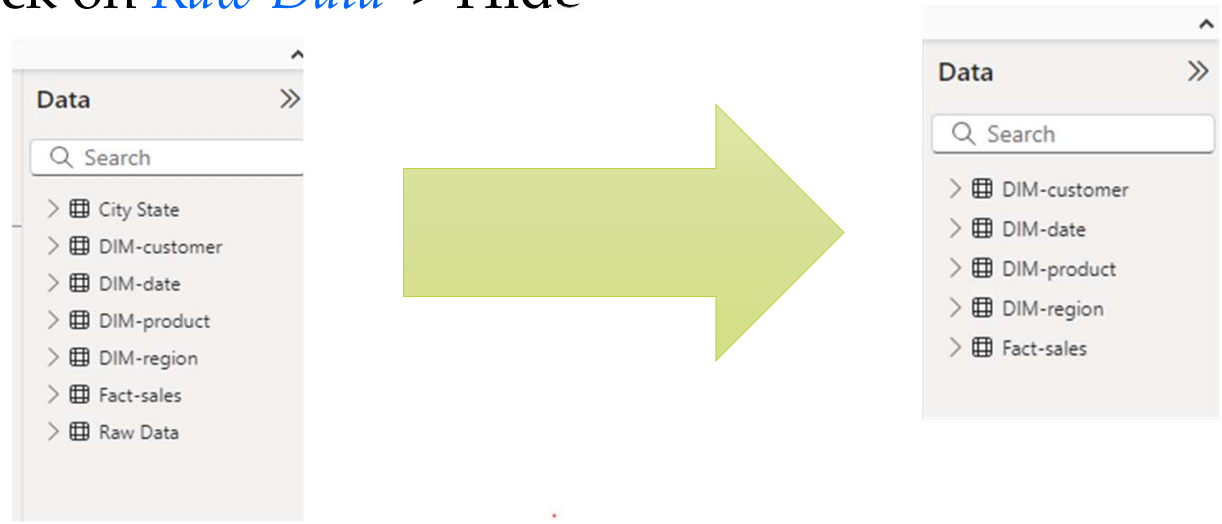




Hide source data

Home > Data blade

- Right click on *City State* > Hide
- Right click on *Raw Data* > Hide



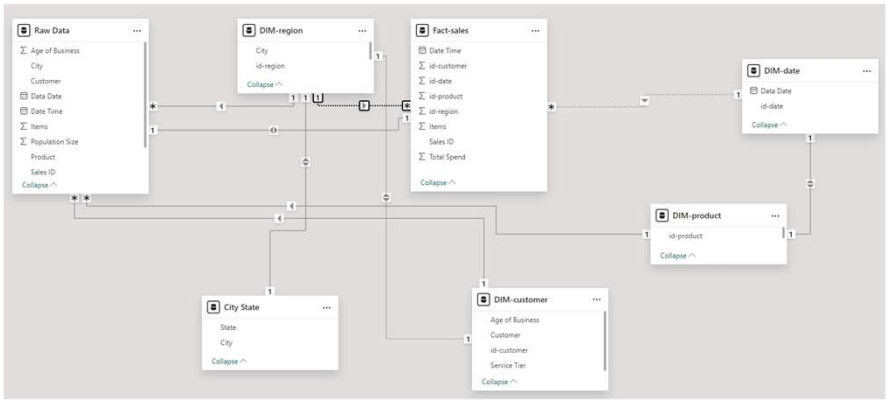
Setting up relationships in Model View

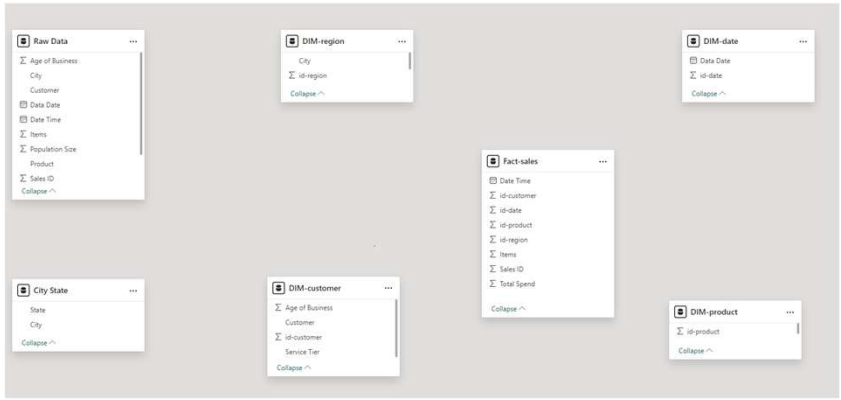
Delete all connections

Home

Open Model View

Delete all connections to all tables



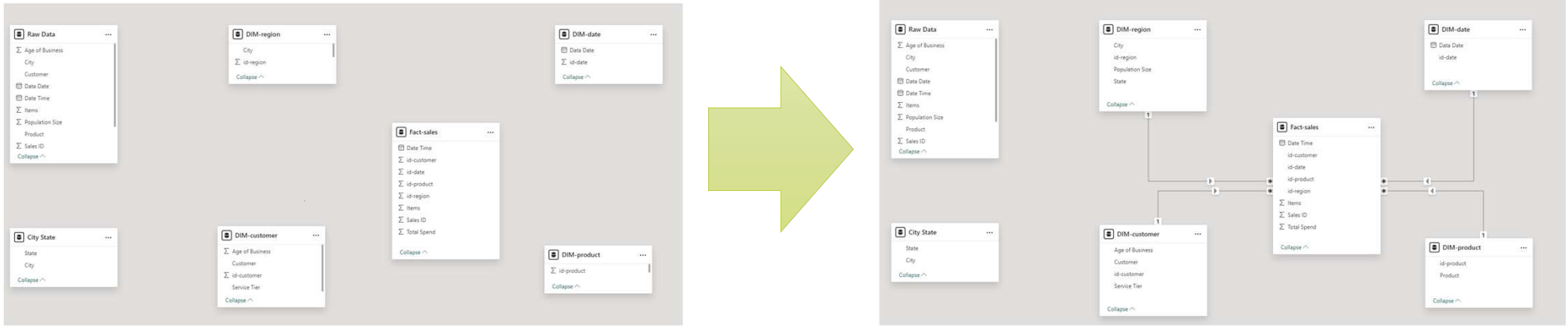


Setting up relationships in Model View

Create Fact and DIM connections

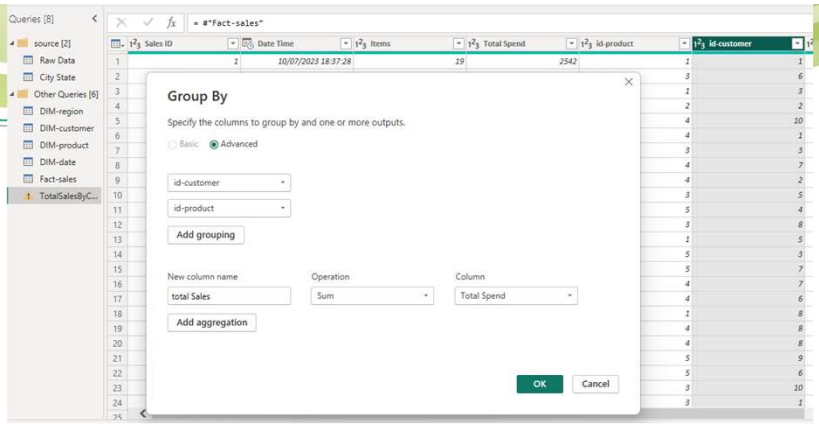
Home > Open Model View

- Create many to one connections by:
  - Connecting (select, drag, drop) *Fact* table *id* columns (FOREIGN KEYS) to respective *DIM* table *id* columns (PRIMARY KEYS)



Aggregation - no DAX

Group By



In Query Editor

- Queries pane > Select Fact-sales > Right click: *Reference*

Rename Reference Table

> Click *Fact-sales (2)* > Right click: *rename: TotSalesByCustomerByProduct*

Apply Aggregation: Group By

In *TotalSalesByCustomerByProduct* select the column to group by

> Select *id-customer*

> Transform Tab > Group By > Advanced > Add Grouping > *id-product*

>New Column name: *Total Sales* > Operation: *Sum* > Column: *Total Spend* > *OK*

Aggregation - no DAX

Joins - for labeling *Customer*

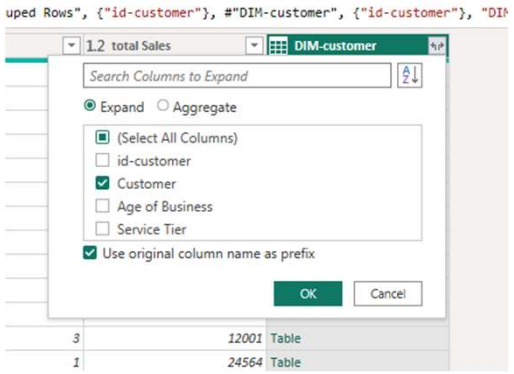
In Query Editor >*Home* tab with *TotalSalesByCustomerByProduct* (Queries) selected > Select *id-customer* column

Join (with Merge)

Merge Queries : merge queries > select *id-customer* for *TotalSalesByCustomerByProduct*

In dropdown > select *DIM-customer* > *id-customer*

> Left outer join



Expand *DIM-customer* > Select *Customer*

> De-select *Use original column name as prefix* > OK

Aggregation - no DAX

Joins - for labeling *Product*

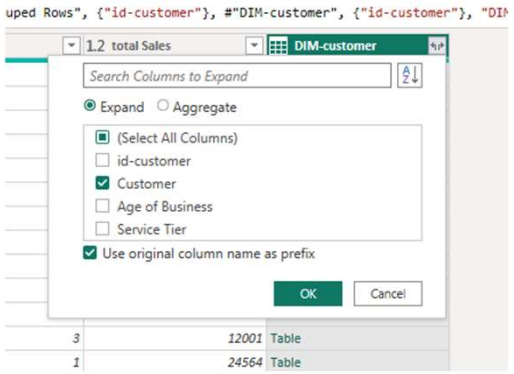
In Query Editor >*Home* tab with *TotalSalesByCustomerByProduct* (Queries) selected > Select *id-product* column

Join (with Merge)

Merge Queries : merge queries > select *id-product* for *TotalSalesByCustomerByProduct*

In dropdown > select *DIM-product* > *id-product*

> Left outer join



Expand *DIM-productr* > Select *Product*

> De-select *Use original column name as prefix* > OK

Delete and Move Columns

- *TotalSalesByCustomerByProduct (*Queries) selected
- Home tab
- Select *id-customer + Ctrl* Select *id-product* > right-click: remove columns
- Select Customer + Ctrl Select Product > Select Transform tab > Move > Left
- Home > Close and Apply: close and apply

Delete *TotalSalesByCustomerByProduct* relationships/connections in model View

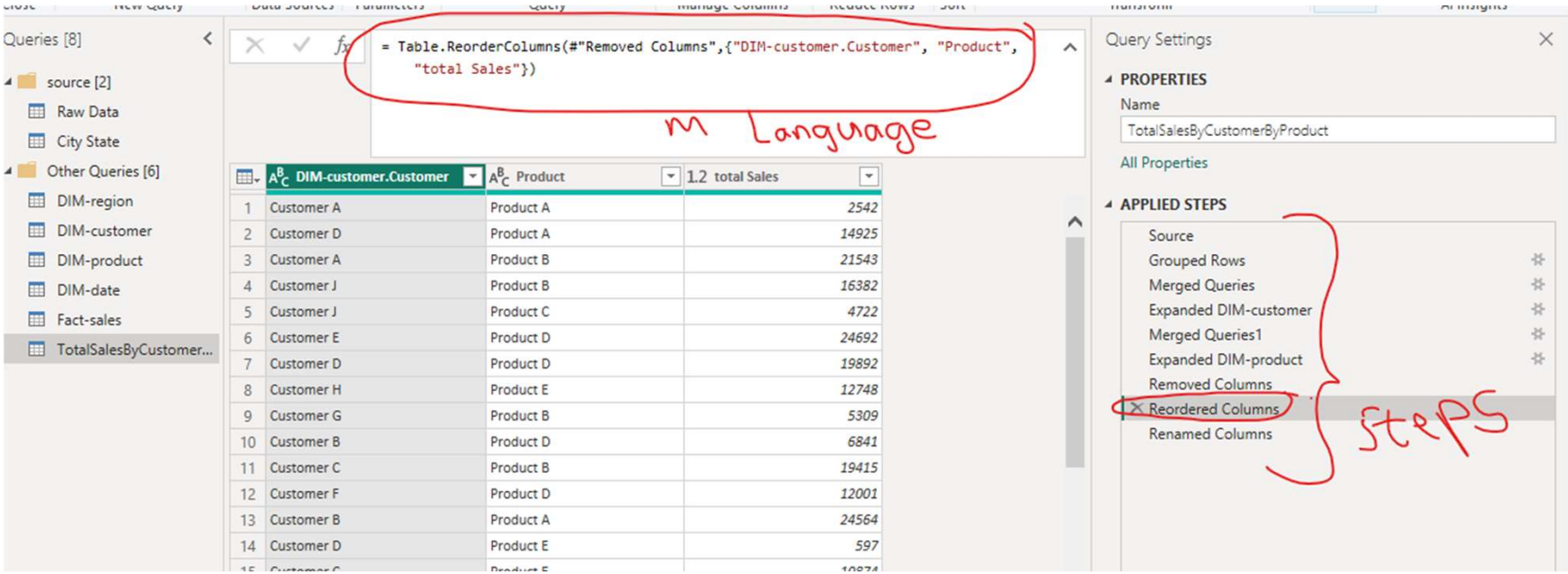
M Language

- M language is used in the Power Query Editor (mostly auto generated code)
  - Power Query formula language (data preparation)
  - Data transformation (before loading data into data model)
- Dax
  - Data Analysis Expression Language (create insights)
  - Analytical Data Calculations (inside data model)
  - Similar to Excel functions

Creating a column with M - language

- There is a visual representation of the transformation steps/instructions you apply in the Power Query Editor
  - You can reverse/undo these steps
- M language is the script behind these transformation steps/instructions
  - If you select a step - you notice the formula bar shows the relevant M language script for that step
    - (expand the formula bar with the down arrow)
  - Manually add/change code in the M language formula bar
- The advanced editoexposes all the M language code behind a Power Query table

M language - behind each step



Use M to create column

Power Query Editor

Select *TotalSalesByCustomerByProduct* (Queries)

> Add Column tab

> Select last Step in *Applied Steps*

> Custom Column

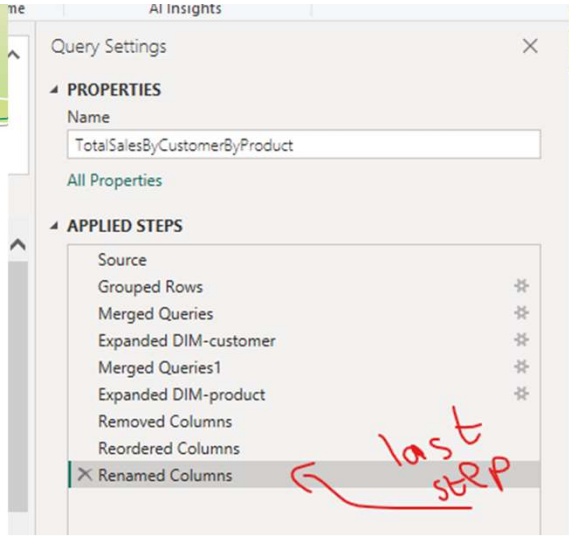
> New Column Name:

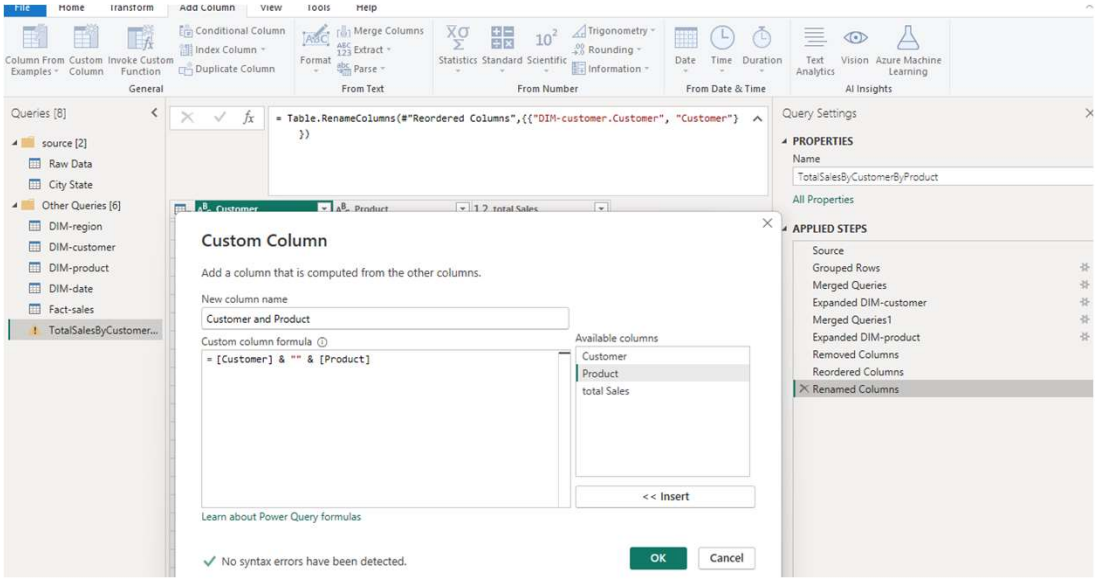
Customer and Product

> Custom column formula

= [Customer] & "" & [Product]

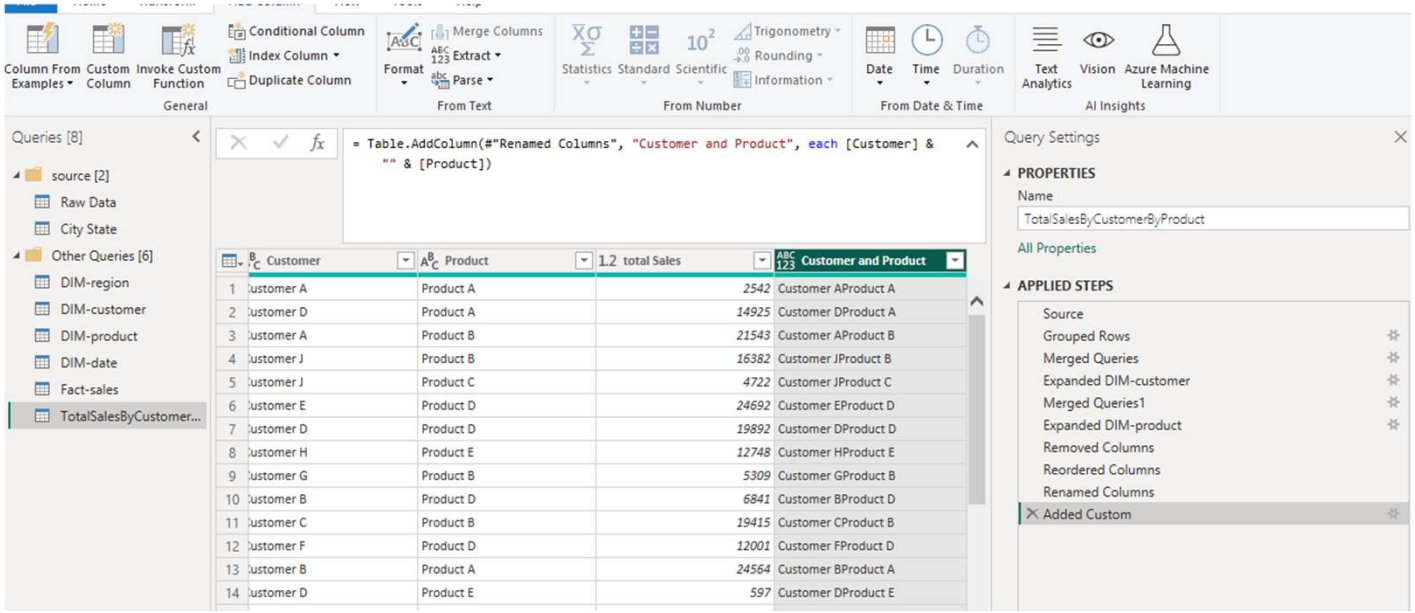
> OK





M language script – to create column

```
= Table.AddColumn(#"Renamed Columns", "Customer and Product", each [Customer] & "" & [Product])
```



Conclusion

Conclusion

Single most important rule:

- *Practice*

Introduction
Data Modelling
- Recap: Business motivation behind dimensional modelling
- Revisit Star Schema
- Connecting to a Table as a *Copy* vs. *Reference*
  - Update implications
- Implement Star Schema from a single raw data table
  - Fact Table
  - Dimension Tables
  - Relationships
- Types of Joins
- Enriching DIM table with *Merge* (join)

Aggregation inside Dimensional Model - without *DAX*
Introduction to *M-language*
Conclusion