CADT បណ្ឌិត្យសភាបច្ចេកវិទ្យាឌីជីថលកម្ពុជា
Cambodia Academy of Digital Technology

TOPIC

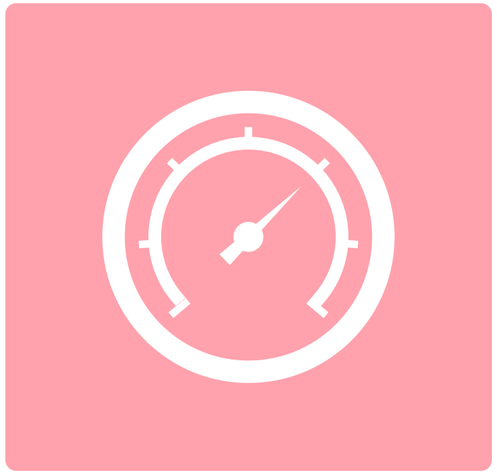
PERFORMANCE CONSIDERATIONS, DEBUGGING AND TOOLS, DAX BEST PRACTICES

Lecturer: Chan Sophal

Group 2: Sok Yongyi, Som Visal, Tet Davann, Siv Sreynoch, Vicheanon Norakpichit, Mak Channa

Contents

Performance Considerations



Factors affecting performance, optimization tips.

Factors affecting performance, optimization tips.

Factors Affecting Performance

1. **Data Source Issues:** Large data sets, complex queries, and slow database performance can all impact report performance
2. **Visualization Issues:** Large and complex visualizations, slow rendering, and slow interactivity can all impact report performance

Factors affecting performance, optimization tips.

Factors Affecting Performance (cont.)

3. Data Modeling Issues: Complex data relationships, inefficient data modeling, and slow data processing can all impact report performance.

4. System Issues: Slow network performance, outdated hardware, and insufficient memory can all impact report performance.

Factors affecting performance, optimization tips.

Optimization tips

- Reduce the size of your data set
- Simplify your queries
- Optimize your database performance
- Simplify visualizations by removing unnecessary fields and reducing the number of data points.
- Optimize rendering by choosing the appropriate visualization type for your data.

Factors affecting performance, optimization tips.

Optimization tips (Cont.)

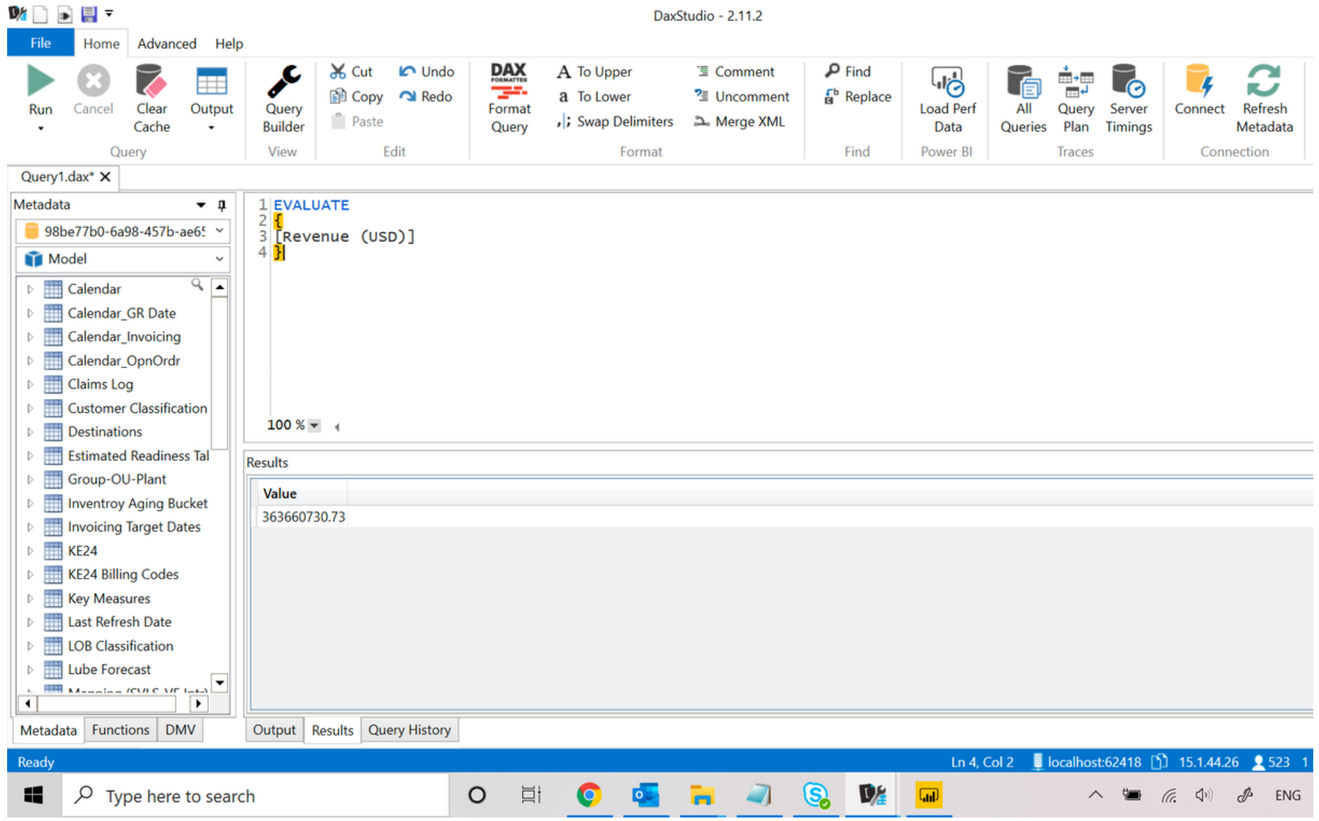
- Simplify data relationships by removing unnecessary joins.
- Optimize data modeling by removing unnecessary columns and reducing the number of tables.
- Improve network performance by optimizing network bandwidth and reducing network congestion.
- Increase memory by adding more RAM to your system.

Debugging and Tools


Define and Fix issue by using tools

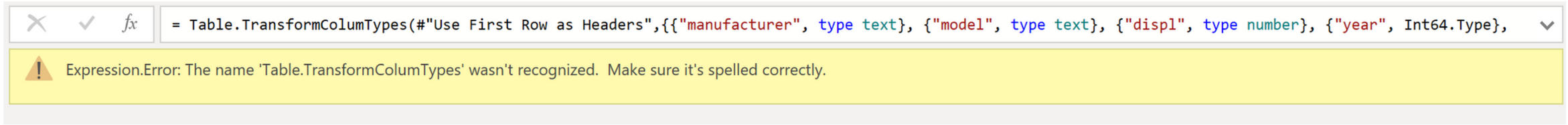
DAX Studio, best practices for debugging.

1. **Use DAX Studio**: DAX Studio is a powerful tool that allows you to connect directly to your data models in Power BI.

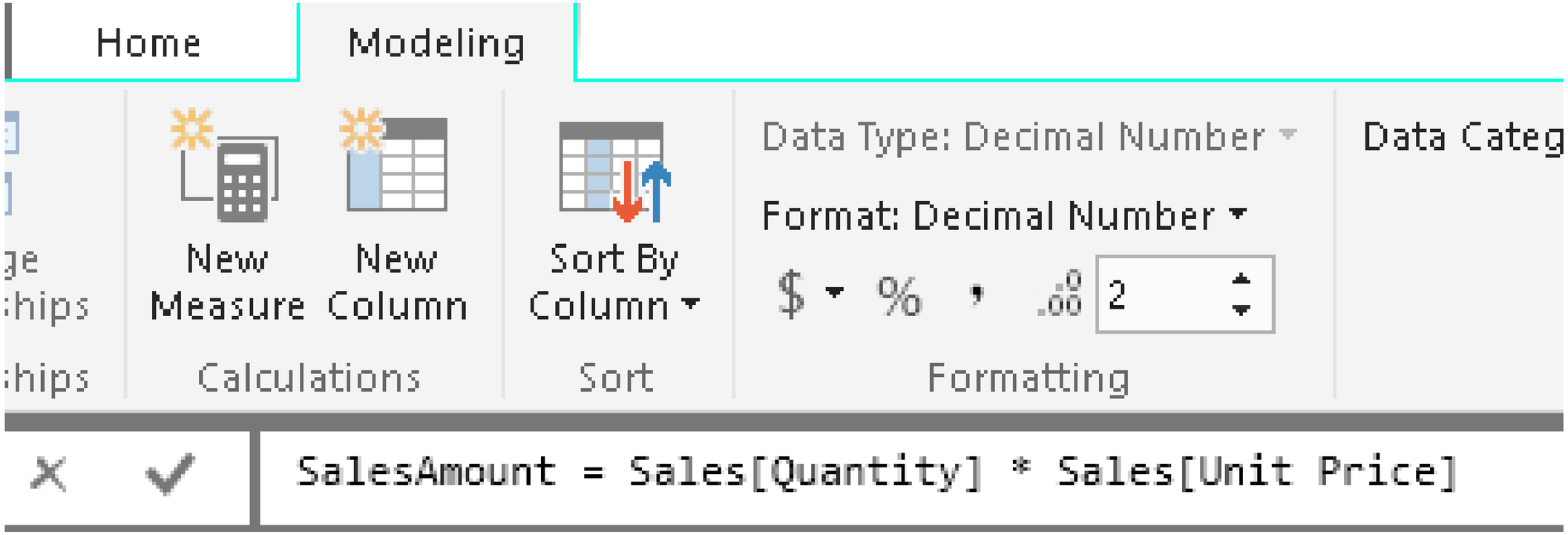


DAX Studio, best practices for debugging.

2. **Break Down Complex Formulas**: If you have complex DAX expressions, break them down into smaller parts to isolate the issue.

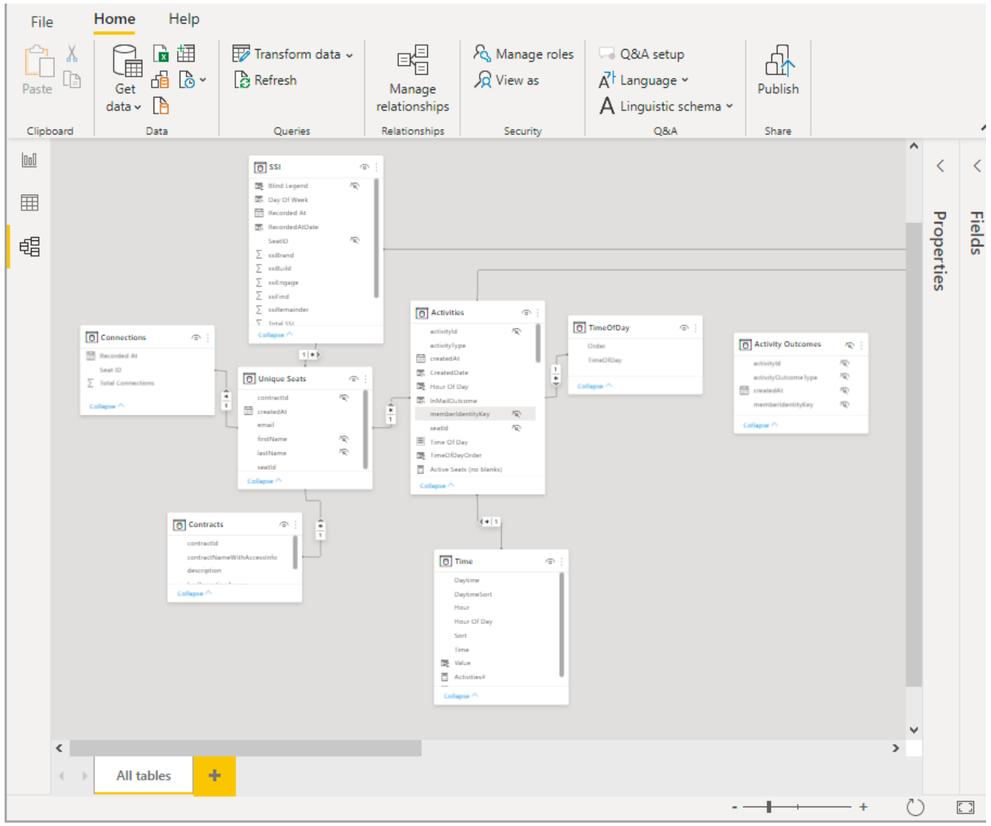


DAX Studio, best practices for debugging.



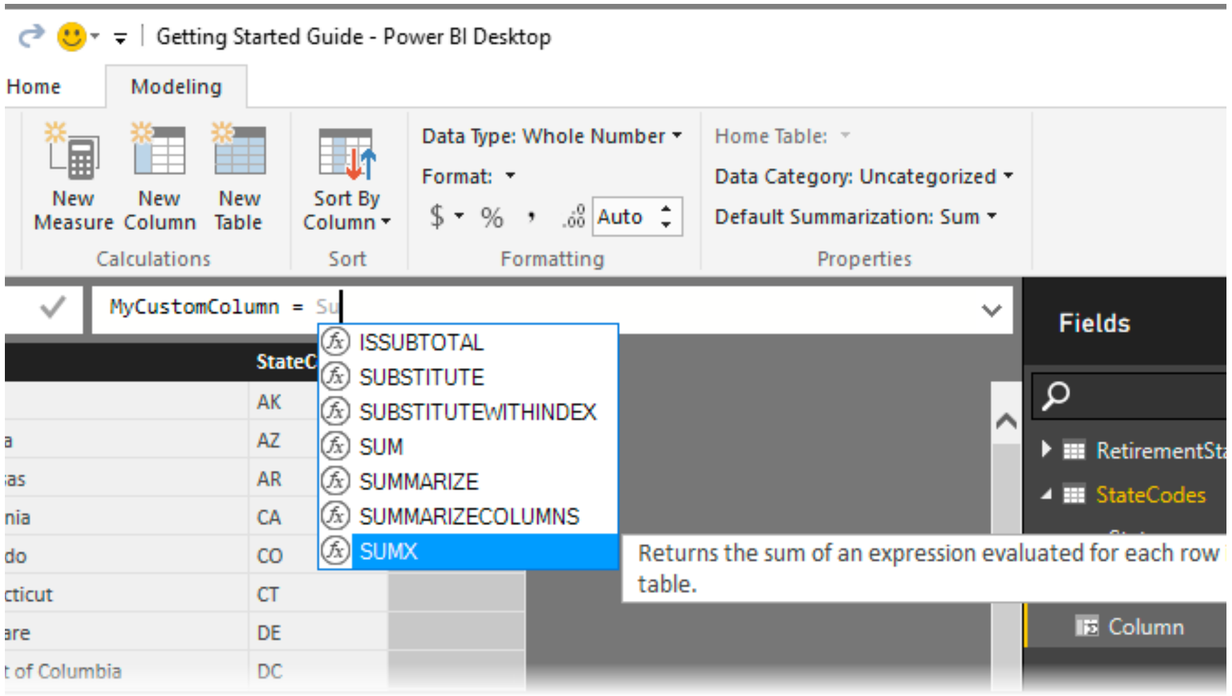
3. Use Measures for Intermediary Calculations: Create intermediary measures to check the intermediate results of your DAX calculations.

DAX Studio, best practices for debugging.



4. Check Data Model Relationships: Ensure that your data model relationships are correctly defined.

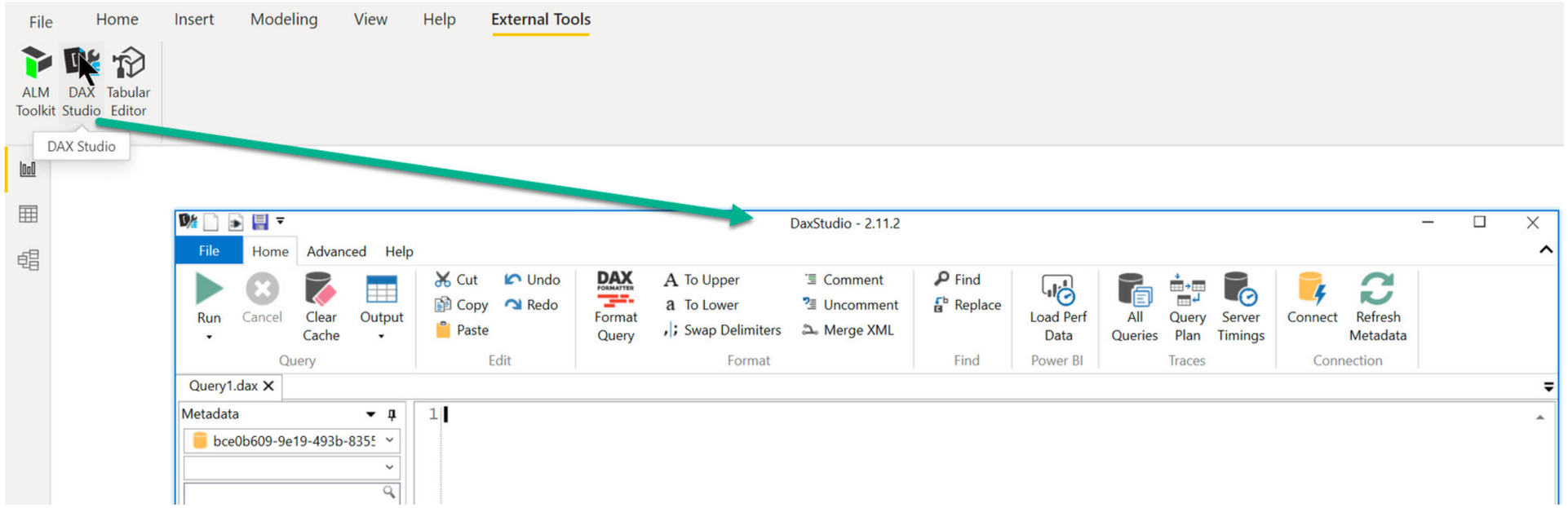
DAX Studio, best practices for debugging.



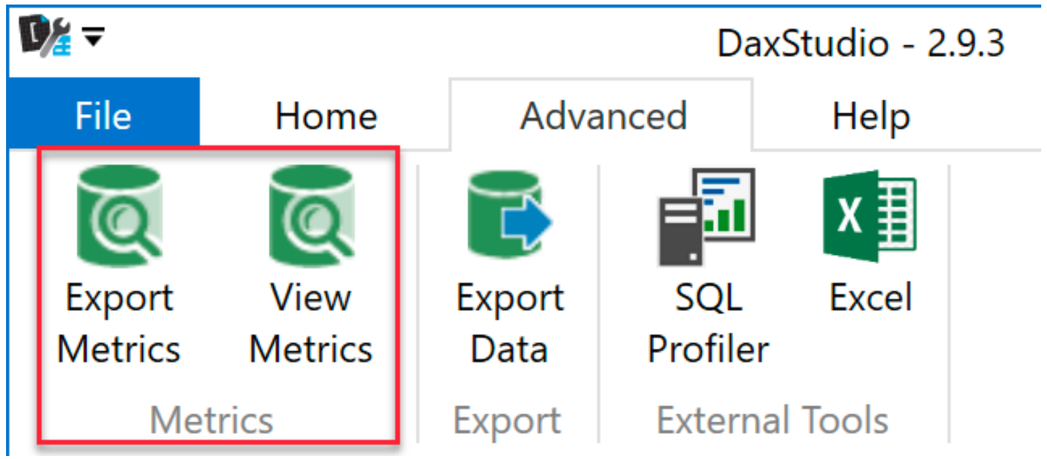
5. Use DAX Functions Step by Step: Test your DAX formulas step by step.

Tools

1. **DAX Studio:** As mentioned earlier, DAX Studio is a widely used external tool for analyzing and debugging DAX expressions.



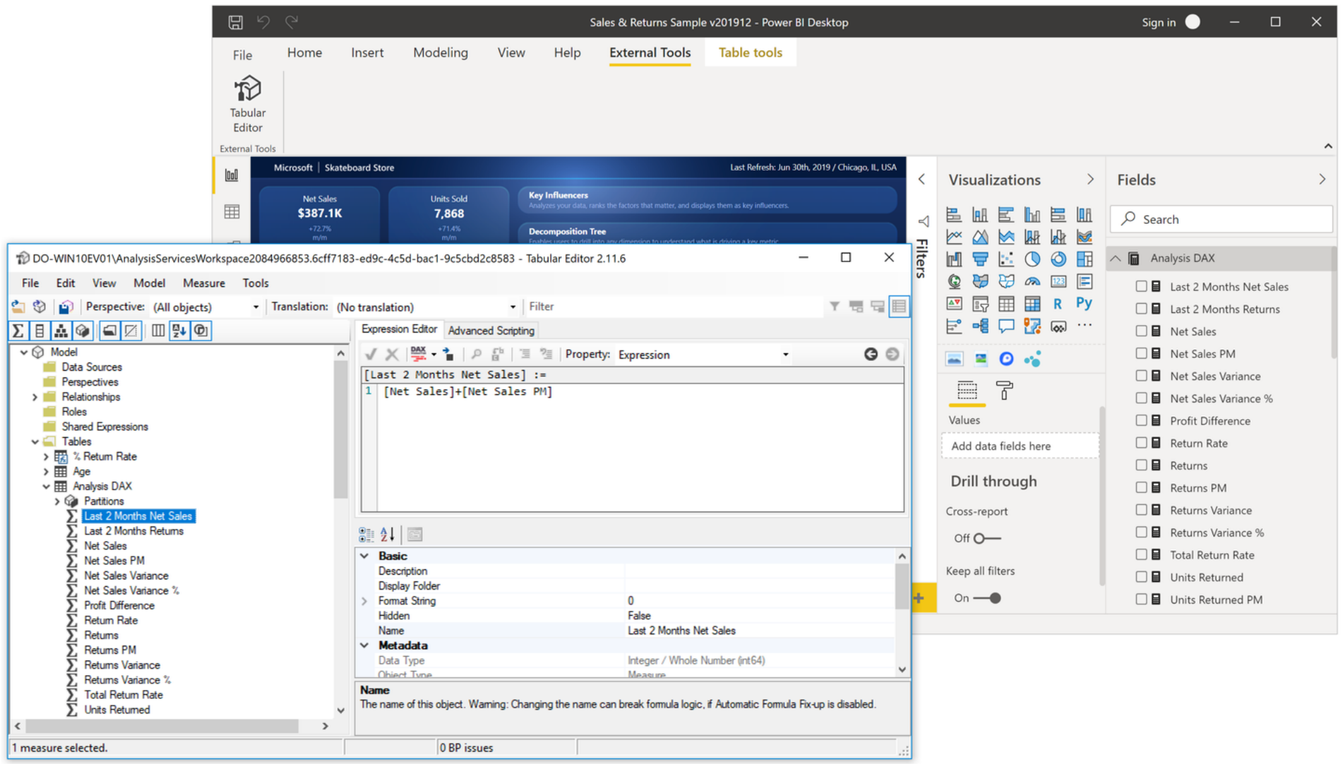
Tools



VertiPaq Analyzer

2. VertiPaq Analyzer: VertiPaq Analyzer is a feature within DAX Studio that helps you analyze the VertiPaq storage engine used in Power BI.

Tools



3. Tabular Editor: Tabular Editor is another external tool that allows you to edit and manage Tabular models in Power BI.

DAX Best Practices

Use SHIFT + CTRL + L to replace words all at once

```
1 Cumulative Total =
2 CALCULATE(
3     [Total Sales (SUMX)],
4     FILTER(
5         ALLSELECTED( Calendar ),
6         Calendar[Date] <= MAX( Calendar[Date] )
7     )
8 )
```

Use DAX Formatter to format the code

```
=SUMX(FILTER(VALUES('Date'[Year]),'Date'[Year]<2005),IF('Date'[Year]>=2000,
[Sales Amount]*100,[Sales Amount]*90)
```

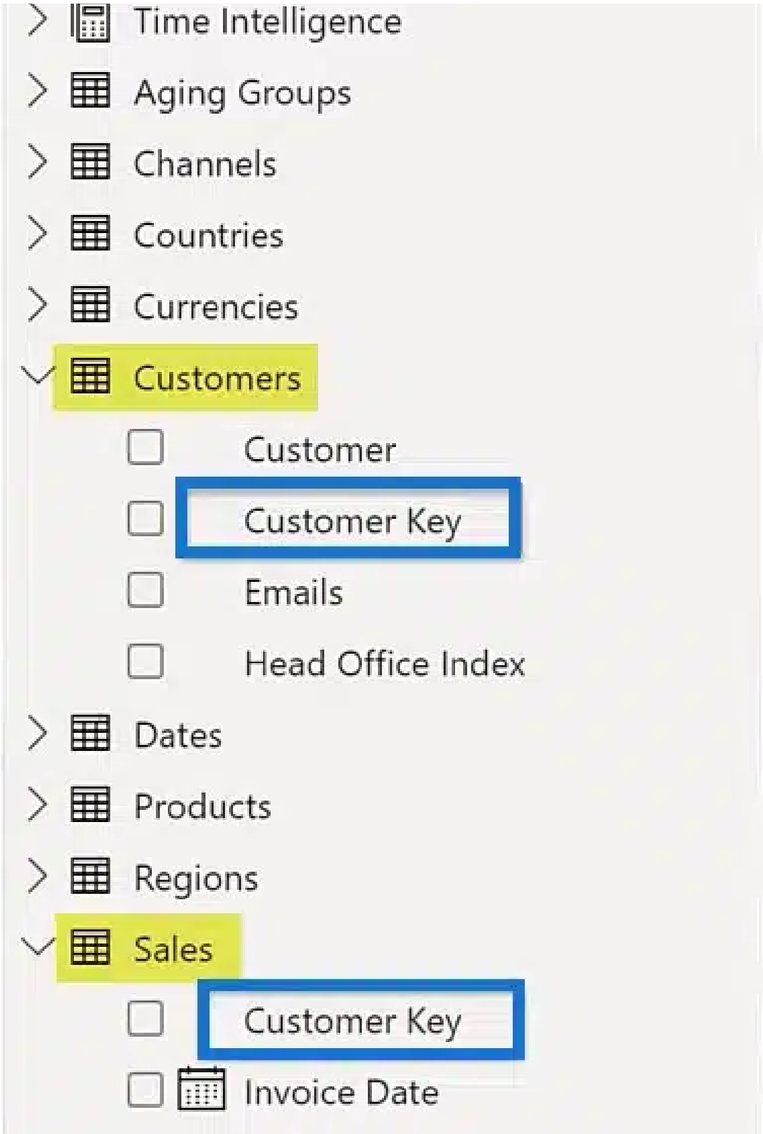
```
=SUMX (
    FILTER (
        VALUES ( 'Date'[Year] ),
        'Date'[Year] < 2005
    ),
    IF (
        'Date'[Year] >= 2000,
        [Sales Amount] * 100,
        [Sales Amount] *  90

)
```

DAX FORMATTER

www.daxformatter.com

Specify Column or Measure Name



Use underscore (_) for variables name

```
Top Product =
VAR _CurrentChannel = SELECTEDVALUE( Channels|
VAR _TopProduct = TOPN( 1, VALUES( Products[Pr
VAR _Result = _
                _CurrentChannel
                _TopProduct
RETURN
_Result
```

Use @ as a prefix for added columns

```
1 Top Channel =
2 VAR _vTable = ADDCOLUMNS(
3     SUMMARIZE( ALL( Channels ), Channels[Channel] ),
4     "@ChannelSales", [Total Sales (SUMX)],
5     "@ChannelSalesRank", RANKX( Channels, [Total Sales (SUMX)],, DESC )
6 )
7 VAR _Result = CALCULATE(
8     MAX( Channels[Channel] ),
9     FILTER( _vTable, [@ChannelSalesRank] = 1 )
10 )
11
12 RETURN
13 _Result
```

THANK YOU

FOR YOUR ATTENTION