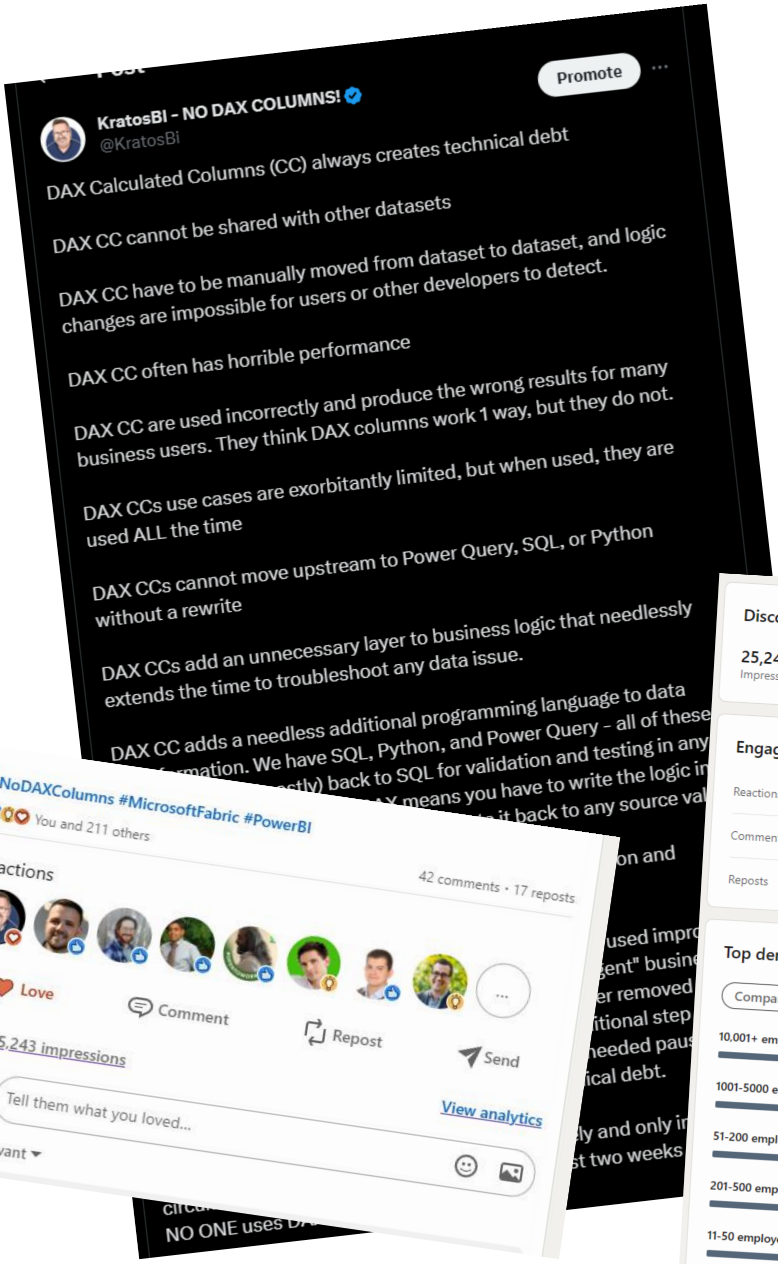
NO DAX Columns

None, Nada, Zero Zilch



Post

Promote

KratosBI - NO DAX COLUMNS!
@KratosBI

DAX Calculated Columns (CC) always creates technical debt

DAX CC cannot be shared with other datasets

DAX CC have to be manually moved from dataset to dataset, and logic changes are impossible for users or other developers to detect.

DAX CC often has horrible performance

DAX CC are used incorrectly and produce the wrong results for many business users. They think DAX columns work 1 way, but they do not.

DAX CCs use cases are exorbitantly limited, but when used, they are used ALL the time

DAX CCs cannot move upstream to Power Query, SQL, or Python without a rewrite

DAX CCs add an unnecessary layer to business logic that needlessly extends the time to troubleshoot any data issue.

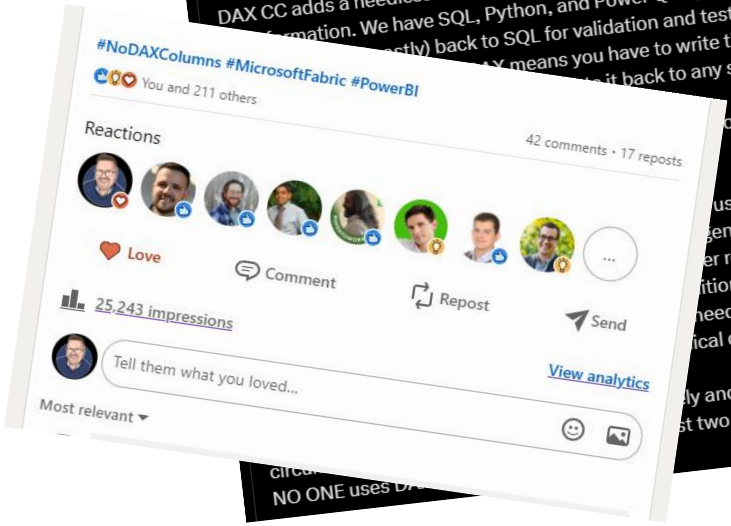
DAX CC adds a needless additional programming language to data rmation. We have SQL, Python, and Power Query - all of these stly) back to SQL for validation and testing in any AX means you have to write the logic in it back to any source val

on and

used impr
gent" busine
er removed
itional step
needed paus
ical debt.

ly and only in
st two weeks

circu
NO ONE uses DA



#NoDAXColumns #MicrosoftFabric #PowerBI

You and 211 others

Reactions

42 comments • 17 reposts

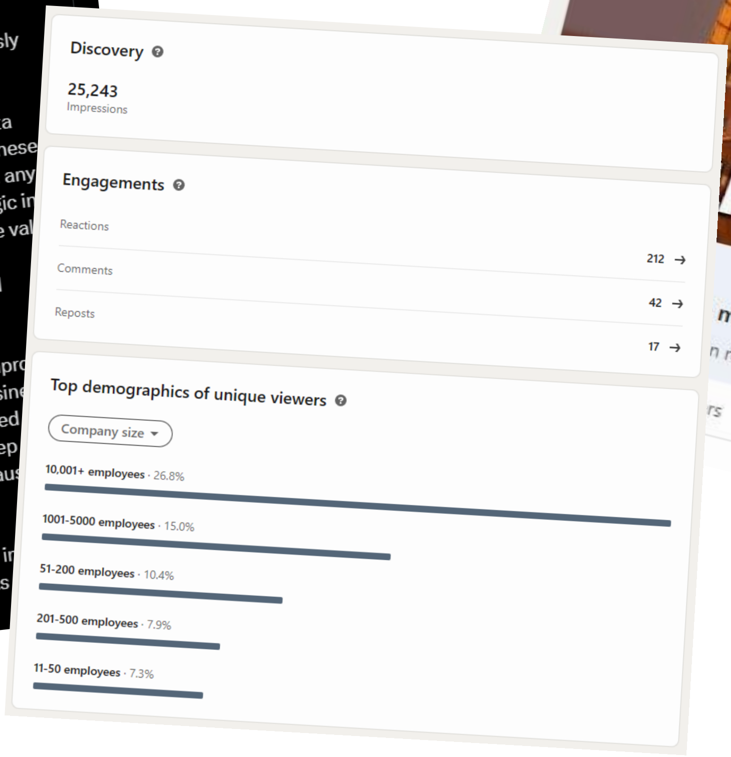
Love Comment Repost Send

25,243 impressions

Tell them what you loved...

View analytics

Most relevant



Discovery

25,243
Impressions

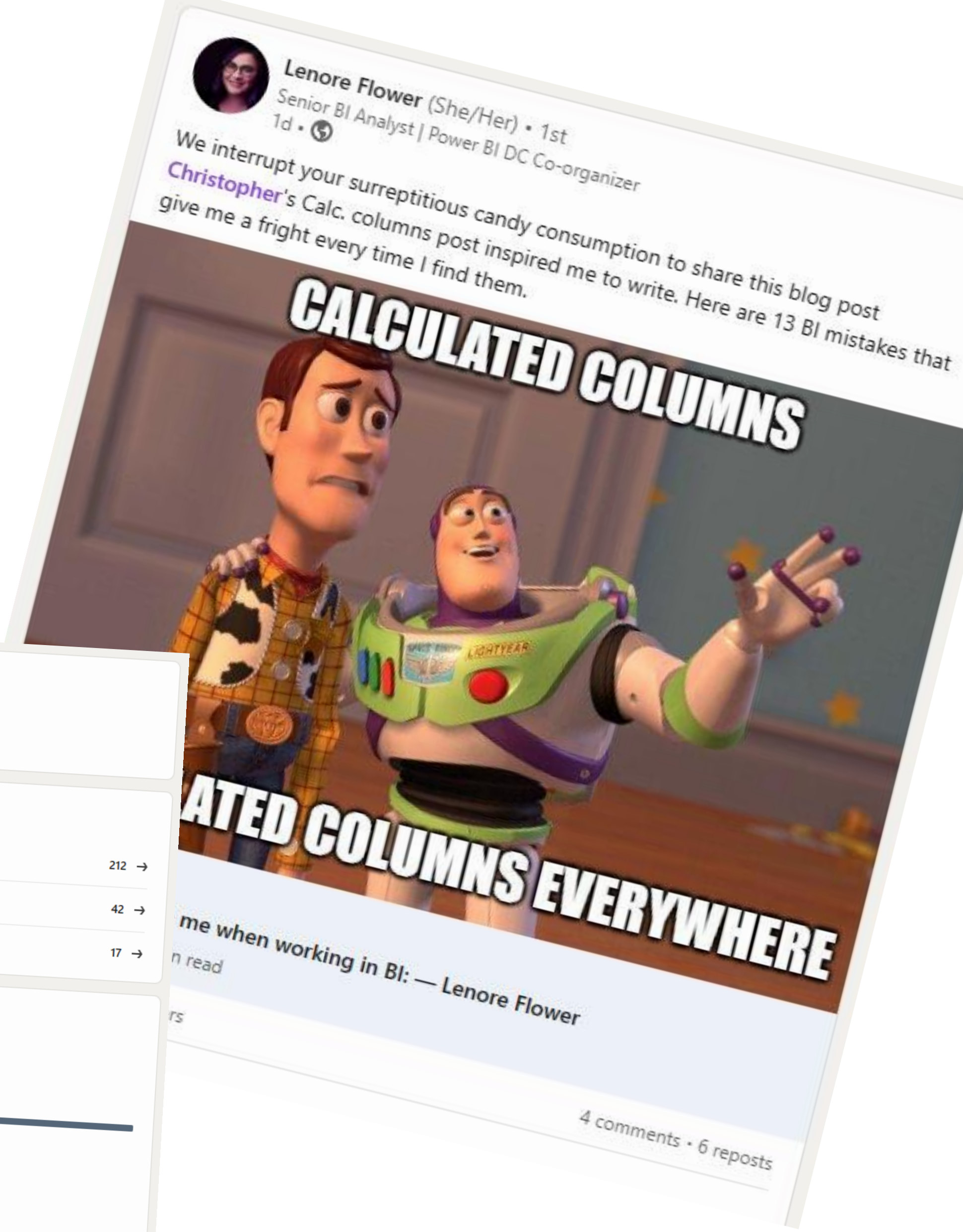
Engagements

Reactions	212 →
Comments	42 →
Reposts	17 →

Top demographics of unique viewers

Company size

10,001+ employees	26.8%
1001-5000 employees	15.0%
51-200 employees	10.4%
201-500 employees	7.9%
11-50 employees	7.3%



Lenore Flower (She/Her) • 1st
Senior BI Analyst | Power BI DC Co-organizer
1d •

We interrupt your surreptitious candy consumption to share this blog post Christopher's Calc. columns post inspired me to write. Here are 13 BI mistakes that give me a fright every time I find them.

me when working in BI: — Lenore Flower
n read

4 comments • 6 reposts

KRATOS BI

The Roche Rule

Roche Principle

As far upstream as possible

As far downstream as necessary

The new floor

DAX Calculated Columns are ALWAYS unnecessary

Semantic Model

Semantic Model – Power Query

Power Query

KRATOS BI

Semantic Model – DAX Measures

Power Query

DAX Measure

Semantic Model – DAX Calculated Column

Power Query

Calculated Column

DAX Measure

Semantic Model – DAX Calculated Table

Power Query

Calculated Column

Calculated Table

DAX Measure

Semantic Model – Current Best Practice

Use Calculated Columns & Tables in limited situations

Power Query

Calculated Column

Calculated Table

DAX Measure

Semantic Model – NEW Best Practice

NEVER Use DAX Calculated Columns or Tables

NEVER EVER
NADA
ZERO
ZILCH

Power Query

Calculated Column

Calculated Table

DAX Measure

IT DOES <u>NOT</u> DEPEND

NO DAX COLUMNS

Architecture

Architecture

Data Warehouse

Semantic Models

Reports

Kimball Architecture

Data Warehouse

Datamart

Semantic Models

Reports

Adhoc Architecture

Data Warehouse

Datamart

Semantic Models

Reports

Adhoc Architecture

Data Warehouse

Datamart

Semantic Models

Reports

Adhoc Architecture – Rampant Sources

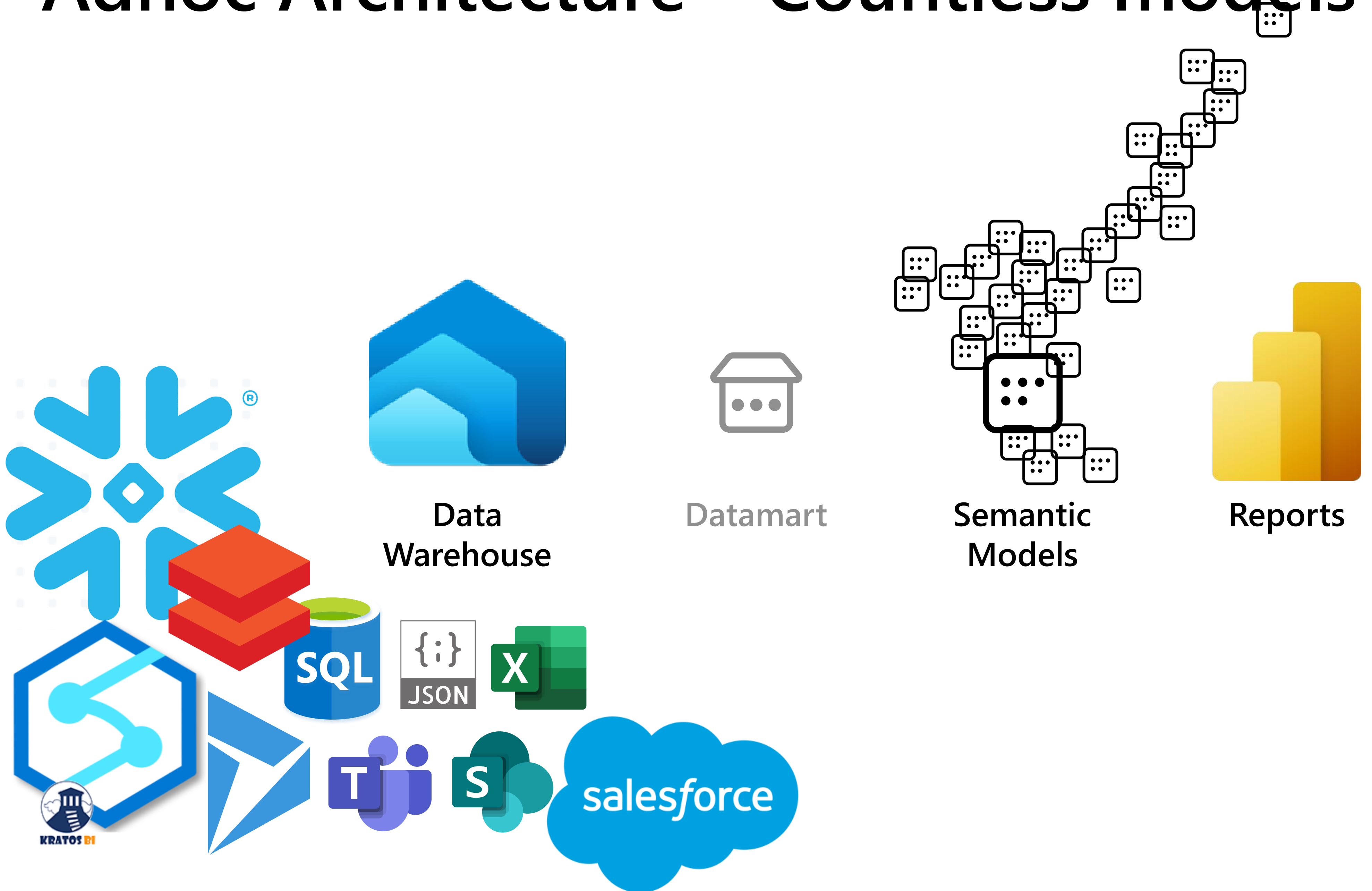
Data Warehouse

Datamart

Semantic Models

Reports

Adhoc Architecture – Countless models



Kimball Architecture – Domain Based Foundation

Data Warehouse

Datamart

Semantic Models

Reports

Datamart

A word on Datamarts

ANY kind of database used to store data for domain needs

Any Database

Anything...

And 12a Issues always impact users

~~19~~ 20

~~17~~ ~~18~~

~~16~~ Reasons WHY NO DAX COLUMNS

And a Prime Reason A – No Limiting Control

And a Prime Reason – No Limiting List

And 2a Management Costs

KRATOS BI

Reasons for NEW Best Practice

- Not in ANY order
- Some may not apply to you
- Some apply in situations
- Some ALWAYS apply
- Too many reasons to ignore
- Always Create Technical Debt
- Too many problems from continuing with current best practice
- No Limiting Factor

Prime Reason – DAX CC No Limiting Factor

No defined list of what TO do or what NOT to do

Tool does not limit users in any way

Prime Reason A – DAX CC Tool does not limit

Tool does not limit users in any way

Reason #1 – DAX CC Do NOT Fold

Cannot Leverage Source Processing

Never Direct Lake / Query

Reason #2 – DAX CC Cost Lots of Money

16-64 x other cloud compute $$$ options

Reason #2a – DAX CC Cost Lots to manage

TONS of $$$ to maintain

KRATOS BI

Reason #3 – DAX CC Always Create Tech Debt

ETL in the wrong layer of your tech stack

Reason #4 – DAX CC Must be Manually Moved

Logic has to be manually moved from model to model

Reason #5 – DAX CC Cannot be shared

Logic has to be reprocessed = potential different results

Reason #6 – DAX CC Have Horrible Performance

Radically increase model load times

Reason #7 – DAX CC Create Incorrect Results

DAX CC are confusing to users and often create incorrect results

DAX CC are used incorrectly and produce the wrong results for many business users. They think DAX columns work 1 way, but they do not.

Reason #8 – DAX CC Increase Load Failures

When used for Joins, model refreshes break when cardinality changes 1 to M becomes M to M

Reason #9 – DAX CC Cannot move without rewrite

ETL is SQL, Python, or Power Query. NOT DAX.

DAX CCs use cases are exorbitantly limited, but when used, they are used ALL the time DAX CCs cannot move upstream to Power Query, SQL, or Python without a rewrite

Reason #10 – DAX CC Additional ETL Language

SQL, Python, or Power Query developer must learn DAX.

DAX CC adds a needless additional programming language to data transformation. We have SQL, Python, and Power Query - all of these can be distilled (mostly) back to SQL for validation and testing in any upstream source. Including DAX means you have to write the logic in both DAX and another language to validate it back to any source value.

Reason #11 – DAX CC increase troubleshooting

Data an issue? DAX CC adds a layer of troubleshooting.

DAX CCs add an unnecessary layer to business logic that needlessly extends the time to troubleshoot any data issue.

Reason #12 – DAX CC Hard to Validate

Hard to manually validate, next to impossible to automate, and ALWAYS impact users.

DAX CC traps logic in the semantic layer, making validation and distribution painfully difficult

Reason #12a – DAX CC Issues impact Users

Issues ALWAYS impact users.

DAX CC traps logic in the semantic layer, making validation and distribution painfully difficult

Reason #13 – DAX CC Multiplies like Rabbits

1 DAX CC *ALWAYS* leads to 500 DAX CC.

A single DAX CC used properly becomes 500 DAX CC used improperly overnight in favor of immediately delivering some "urgent" business need that comes and goes just as quickly. This is then never removed until the model is not functioning properly. Adding in the additional step of at least putting it into Power Query provides a much-needed pause in the reflexive response that creates much of this technical debt.
At one point, I used to say use DAX columns rarely and only in certain circumstances. My stance has evolved in the last two weeks because NO ONE uses DAX columns properly.

Reason #14 – DAX CC 1 right becomes 1,000 wrong

Each DAX CC increases all issues with DAX CC.

A single DAX CC used properly becomes 500 DAX CC used improperly overnight in favor of immediately delivering some "urgent" business need that comes and goes just as quickly. This is then never removed until the model is not functioning properly. Adding in the additional step of at least putting it into Power Query provides a much-needed pause in the reflexive response that creates much of this technical debt.
At one point, I used to say use DAX columns rarely and only in certain circumstances. My stance has evolved in the last two weeks because NO ONE uses DAX columns properly.

Reason #15 – DAX CC should be Power Query

JUST

MOVE

IT

TO

POWER QUERY

Reason #16 – DAX CC moved to a Datamart

Stop trying to cheat the architecture. Get a Datamart.

IT DOES NOT DEPEND

NO DAX COLUMNS

If you are using DAX Columns

1. Slow down your development process
2. Move them to Power Query
3. Get a Datamart

Benefits of this effort

1. Improves Quality
2. Reduces Costs
3. Lower Development Time
4. Reduces Differences
5. Reduces Failures / Issues
6. Faster troubleshooting
7. Improves Performance
8. Reduces Costs
9. Respect of peers
10. Kimball architecture

KRATOS BI

IT DOES <u>NOT</u> DEPEND

NO DAX COLUMNS

KRATOS BI

Original No DAX Columns Post

DAX Calculated Columns (CC) always creates technical debt

DAX CC cannot be shared with other datasets

DAX CC have to be manually moved from dataset to dataset, and logic changes are impossible for users or other developers to detect.

DAX CC often has horrible performance

DAX CC are used incorrectly and produce the wrong results for many business users. They think DAX columns work 1 way, but they do not.

DAX CCs use cases are exorbitantly limited, but when used, they are used ALL the time DAX CCs cannot move upstream to Power Query, SQL, or Python without a rewrite

DAX CCs add an unnecessary layer to business logic that needlessly extends the time to troubleshoot any data issue.

DAX CC adds a needless additional programming language to data transformation. We have SQL, Python, and Power Query - all of these can be distilled (mostly) back to SQL for validation and testing in any upstream source. Including DAX means you have to write the logic in both DAX and another language to validate it back to any source value.

DAX CC traps logic in the semantic layer, making validation and distribution painfully difficult A single

DAX CC used properly becomes 500 DAX CC used improperly overnight in favor of immediately delivering some "urgent" business need that comes and goes just as quickly. This is then never removed until the model is not functioning properly. Adding in the additional step of at least putting it into Power Query provides a much-needed pause in the reflexive response that creates much of this technical debt.
At one point, I used to say use DAX columns rarely and only in certain circumstances. My stance has evolved in the last two weeks because NO ONE uses DAX columns properly. Using

DAX columns says to me a few things:
1 - You need to slow down the process a little to validate the need and the data
2 - 99.9999% of what people put in DAX CC should be in Power Query. At a minimum, move the business logic there.
3 - You need to be looking at your architecture and add a datamart for anything that cannot be done with Power Query
4 - Data management is complex for us, and doubly complex for people consuming the information. The variability that comes from different datasets with different refresh schedules, measures, and XXXX ensures that the business will make assumptions about the data that WILL be wrong.

Anything that can eliminate 1 of the layers of variables is a good thing in reducing complexity