



# 14 edycja konferencji SQLDay

9-11 maja 2022, WROCŁAW + ONLINE



---

partner złoty

---



---

partner srebrny

---



---

partner brązowy

---



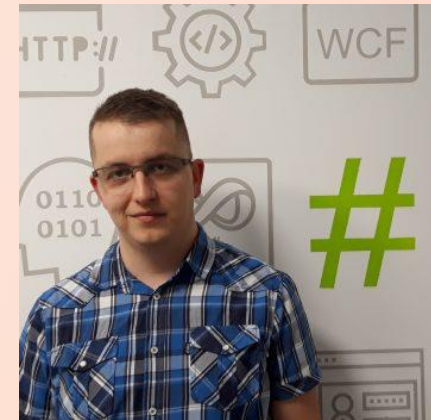
Adrian Chodkowski

# 10 mitów oraz błędów związanych z Power BI

---

# O mnie

- Adrian Chodkowski
- Data Solutions Technical Lead
- Elitmind
- Specjalizacja: Platforma danych Microsoft
- Data Community
- [seequality.net](https://seequality.net)
- [adrian.chodkowski@outlook.com](mailto:adrian.chodkowski@outlook.com)





# AGENDA

---

1. „Na workspace jest najnowsza wersja raportu”
2. „Power Query sam sobie poradzi z ładowaniem danych”
3. „Moje dane są bezpieczne – przecież je pobieram, a nie wysyłam”
4. „Raport zawiera wszystkie wymiary i fakty dzięki czemu nie będzie trzeba go aktualizować”
5. „Płaska struktura jest ok, szkoda czasu na strukturę gwiazdy”
6. „Front-end nie ma wpływu na wydajność”
7. „Direct Query jest wolne i należy go unikać”
8. „Zrób to na kolumnach kalkulowanych”
9. „Power BI nie nadaje się do dużych zestawów danych”
10. „Wystarczy, że nauczę się składni DAX i wszystko działa”



#1

„Na workspace jest  
najnowsza wersja raportu”

# Przechowywanie raportów

- Lokalny komputer ❌
- Współdzielony folder ❌
- Workspace ❌
- OneDrive ✅
- Repozytorium GIT ✅

The screenshot displays a Windows File Explorer window showing a folder named 'MyReports' located on the Desktop. The folder contains seven Power BI report files named 'MyReport (1).pbix' through 'MyReport (7).pbix'. Overlaid on this is a screenshot of the Azure DevOps interface. The top part shows the 'SQLDAY' project dashboard with a table of reports. The bottom part shows the 'Pull requests' section with a list of pending requests.

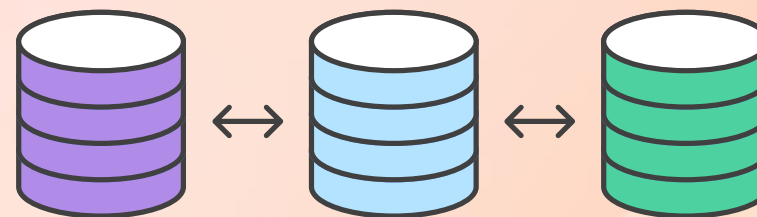
Name	Type	Owner
MyReport (1)	Report	SQLDAY
MyReport (1)	Dataset	SQLDAY
MyReport (1).pbix	Dashboard	SQLDAY
MyReport (2)	Report	SQLDAY
MyReport (2)	Dataset	SQLDAY
MyReport (2)	Dashboard	SQLDAY

**Pull requests**

- Initiate client with client init
- Use latest React version
- Check returned identity for null status
- [WIP] Add tests for reticulating splines
- Add exception mappings for disconnect
- Maintain folder structure when converting isomorphs
- Testing configuration settings
- Hotfix payload to Orion

# Repozytorium

- Pełna historyzacja plików ,
- Brak możliwości porównywania wewnętrznych struktur plików,
- Możliwość implementacji branchy oraz kontrolowanego deploymentu (Azure DevOps),
- W pewnych okolicznościach warto rozważyć projekt Visual Studio – daje to pełną kontrolę wersji!

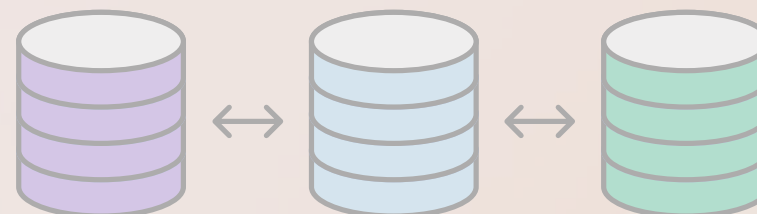


<https://www.atlassian.com/git/tutorials/comparing-workflows>

```
168     "expression": [
169       "let",
170       "  Source = #\"SQL/ ;AdventureWorksDW2017\",",
171       "  dbo_FactInternetSales = Source{[Schema=\"dbo\",Item=\"FactInter",
172       "in",
173       "  dbo_FactInternetSales"
174     ]
175   }
176 },
177 ],
178 "measures": [
179   {
180     "name": "SumOfSalesAmount",
181     "expression": " SUMX('FactInternetSales',[OrderQuantity]*[UnitPrice])",
182     "formatString": "#,0.00"
183   }
184 ],
185 }
186
```

# Repozytorium

- Pełna historyzacja plików ,
- Brak możliwości porównywania wewnętrznych struktur plików,
- Możliwość implementacji branchy oraz kontrolowanego deploymentu (Azure DevOps),



Jeśli to możliwe przechowujemy pbit a nie pbix!

```
170     "dbo_FactInternetSales = Source{[Schema=\"dbo\",Item=\"FactInter
171     \"in\",
172     \"dbo_FactInternetSales\"
173   ]
174 }
175 }
176 ],
177 \"measures\": [
178 {
179   \"name\": \"SumOfSalesAmount\",
180   \"expression\": \"SUMX('FactInternetSales',[OrderQuantity]*[UnitPrice])\",
181   \"formatString\": \"#,0.00\"
182 }
183 ],
184 \"formatString\": \"#,0\"
185 }
186 }
```




# GIT + PBI Desktop!

<https://ideas.powerbi.com/ideas/idea/?ideaid=9daf5a6b-19e1-4b79-9e7c-78d877d54542>

Power BI Ideas > **Microsoft Idea**

2968

Vote 



**Built in Git support in PowerBI Desktop**  
[Matt Smith](#) on 2/28/2019 5:19:38 PM

STATUS DETAILS

PLANNED



Administrator on 3/18/2022 4:12:08 AM

March 2022 update: We are working on this item but no timeline can be shared yet. We appreciate your patience



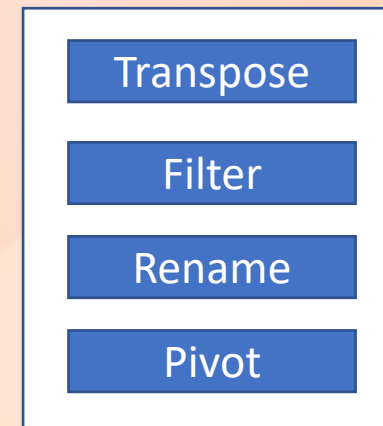
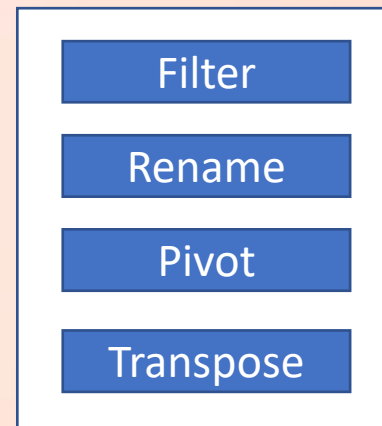
#2

„Power Query sam sobie poradzi  
z ładowaniem danych”



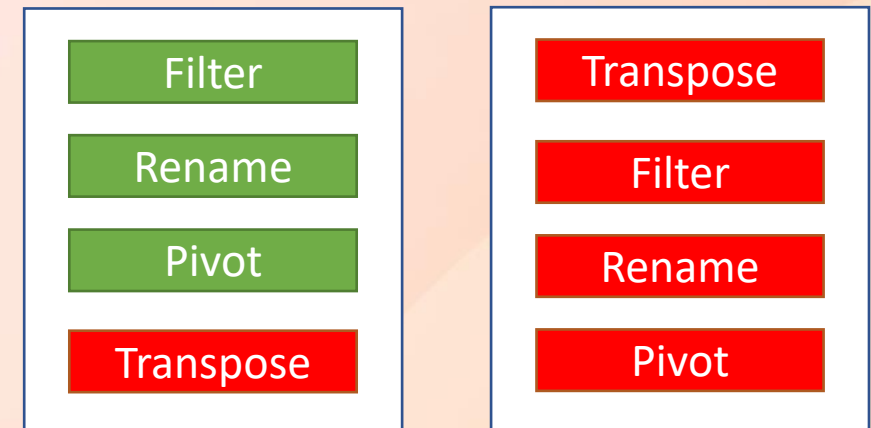
# Query Folding

- Mechanizm transformujący zapytania M na zapytanie źródła
- Jeśli coś nie może zostać przerzucone to jest wykonywane lokalnie przez Power BI
- Zapytanie może zostać przerzucone w całości lub częściowo
- Zależy od:
  - Typu źródła
  - Zastosowanej logiki transformacji
  - Privacy Level
- Pamiętaj, że możesz użyć funkcji **Value.NativeQuery** aby napisać zapytanie do źródła i nie przeszkodzić mechanizmowi Query Folding



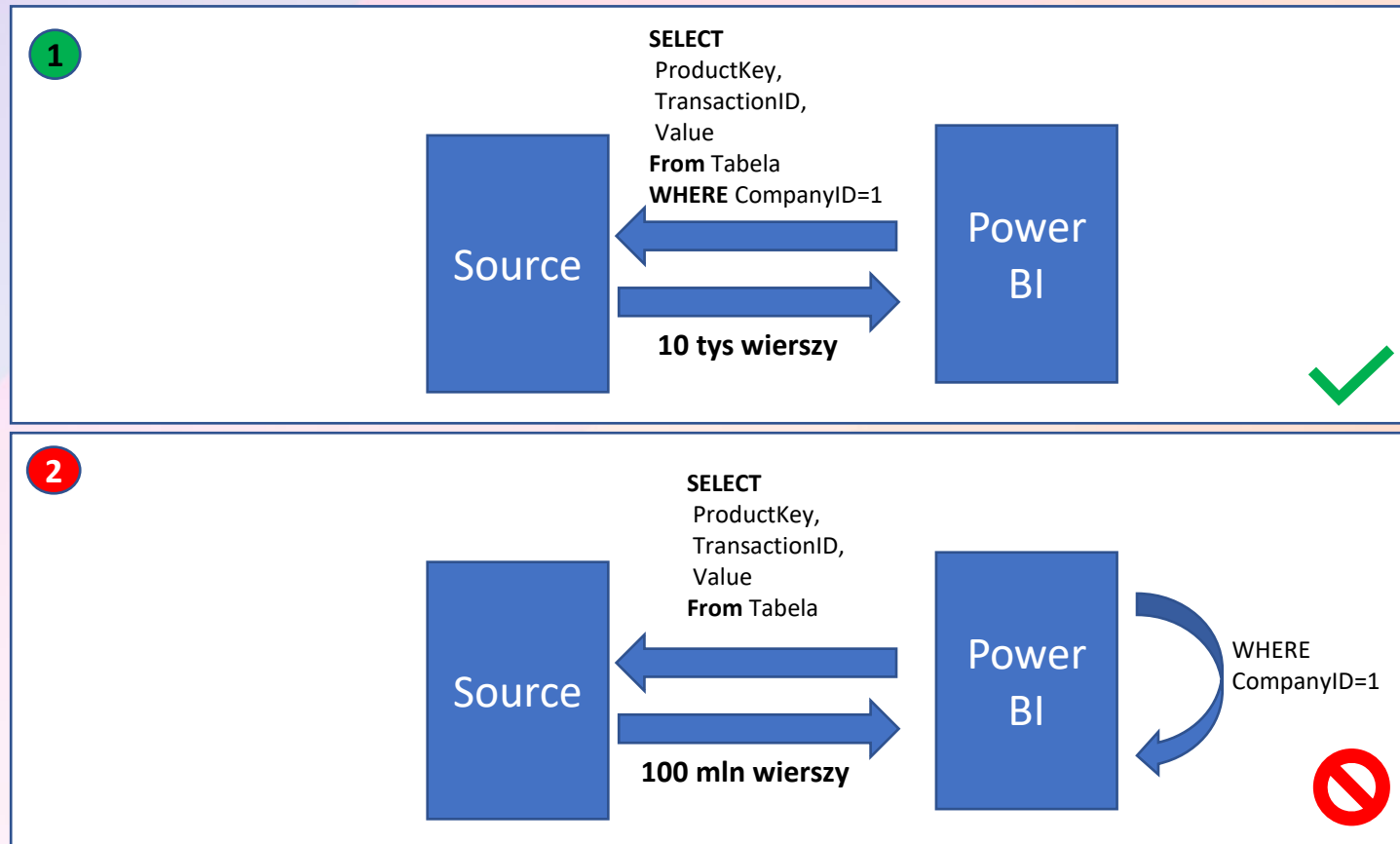
# Query Folding

- Mechanizm transformujący zapytania M na zapytanie źródła
- Jeśli coś nie może zostać przerzucone to jest wykonywane lokalnie przez Power BI
- Zapytanie może zostać przerzucone w całości lub częściowo
- Zależy od:
  - Typu źródła
  - Zastosowanej logiki transformacji
  - Privacy Level
- Pamiętaj, że możesz użyć funkcji `Value.NativeQuery` aby napisać zapytanie do źródła i nie przeszkodzić mechanizmowi Query Folding



**Transformacja która nie może być  
przerzucona powoduje, że  
wszystkie następujące  
po niej transformacje też nie  
zostaną przerzucone!**

# Query Folding



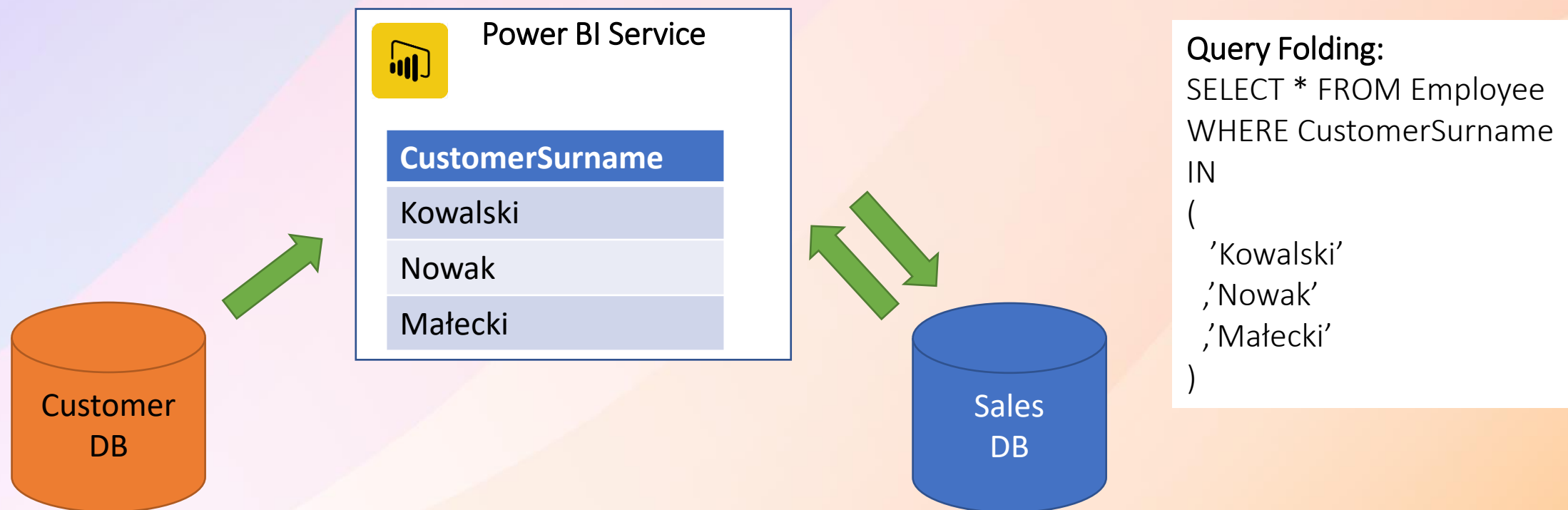


#3

„Moje dane są bezpieczne – przecież je pobieram,  
a nie wysyłam”

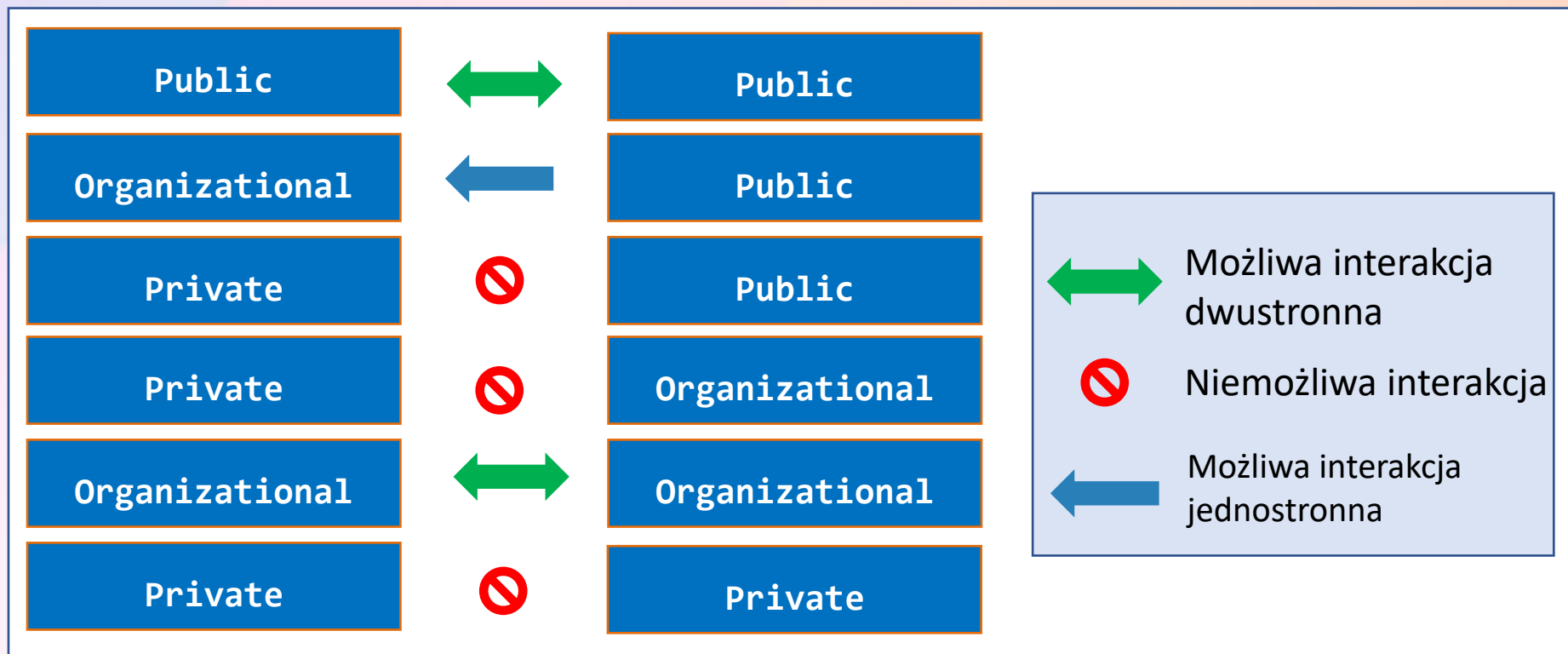
# Privacy Levels

- Interakcja pomiędzy źródłami jest określona poziomami izolacji. Dane nie mogą przepływać ze źródła bardziej do mniej restrykcyjnego. Jeśli korzystamy z jednego źródła można zignorować poziomy izolacji lub ustawić je jako Organizational.



# Privacy Levels

- Interakcja pomiędzy źródłami jest określona poziomami izolacji. Dane nie mogą przepływać ze źródła bardziej do mniej restrykcyjnego. Jeśli korzystamy z jednego źródła można zignorować poziomy izolacji lub ustawić je jako Organizational.





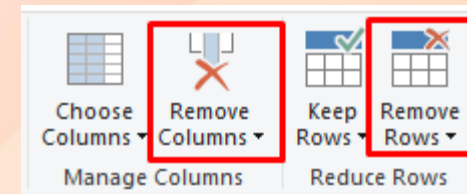
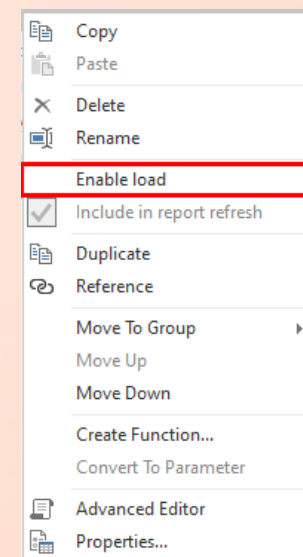


#4

„Raport zawiera wszystkie wymiary i fakty  
dzięki czemu nie będzie trzeba go  
aktualizować”

# Zbędne obiekty w modelu

- Jeśli jakiś obiekt nie jest nam potrzebny na raporcie nie ładujmy go,
- Jeśli obiekt jest nam potrzebny tylko w ramach ETL to nie ładujmy go do modelu końcowego,
- Model tabelaryczny w dużej mierze opiera się na kompresji dlatego:
  - Im mniej tym lepiej,
  - Im lepiej kompresowalne typy danych tym lepiej
- Kilka ogólnych praktyk:
  - Osobne kolumny data, czas > pojedyncza kolumna data czas
  - Liczby całkowite są lepsze niż GUIDy czy HASHe (nie dotyczy Direct Query)
  - Rozważ użycie mniejszego typu (np. z mniejszą liczbą miejsc po przecinku)
  - Twórz wymiary zdegenerowane tylko jeśli mają poziom granulacji faktu (lub zbliżony)



<https://www.sqlbi.com/tools/vertipaq-analyzer/>



# VertiPaq

- VertiPaq (xVelocity) jest silnikiem przechowującym dane kolumnowo
- Taki sposób przechowywania daje możliwość lepszej kompresji oraz agregacji
- Dodatkowo odczyt pojedynczej kolumny nie wymaga odczytu pozostałych kolumn w przeciwieństwie do przechowywania wierszowego

## Przechowywanie wierszowe

Column 1	Column 2	Column 3	Column 4	Column 5

## Przechowywanie kolumnowe

Column 1	Column 2	Column 3	Column 4	Column 5

# Value Encoding

- Metoda działająca wyłącznie dla wartości całkowitoliczbowych nie będących częścią relacji których zakres nie jest zbyt szeroki,
- Znajduje najmniejszą wartość w zakresie i na jej podstawie oblicza różnice dla pozostałych wartości,
- Poszczególne wartości są przechowywane używając mniejszej ilości pamięci.

Value		Value
1562		62
1500		0
1578		78
1544		44
1587		87
1501		1
1509		9

# Dictionary Encoding

- Rzeczywiste wartości są przechowywane w słowniku
- Przypisywany jest im identyfikator używając **funkcji hashującej**
- W określonej kolumnie zamieniane są określone wartości na ich identyfikator
- Używany dla każdego typu danych – na rozmiar ma wpływ liczebność (**cardinality**), a nie sama wartość

Value
Apple
Banana
Apple
Apple
Orange
Apple
Banana



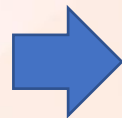
1
2
1
1
3
1
2

ID	Value
1	Apple
2	Banana
3	Orange

# Run-Length Encoding

- Algorytm pozwalający zapisać zakresy od kiedy do kiedy dana wartość występuje
- Bardzo wrażliwy na kardynalność oraz ułożenie danych
- VertiPaq używa heurystyk żeby znaleźć idealne sortowanie – można na nie wpłynąć dostarczając dane w odpowiedniej kolejności dzięki czemu slinik zdecyduje w swojej ocenie.
- Czas na odnalezienie odpowiedniego sortowania jest skończony i nie da się go ustawić w Power BI – w SSAS jest on dostępny pod nazwą **ProcessingTimeboxSecPerMRow** – ustawienie go na 0 spowoduje uzyskanie najlepszego porządku sortowania.

Value
Apple
Apple
Apple
Apple
Banana
Banana
Blueberry



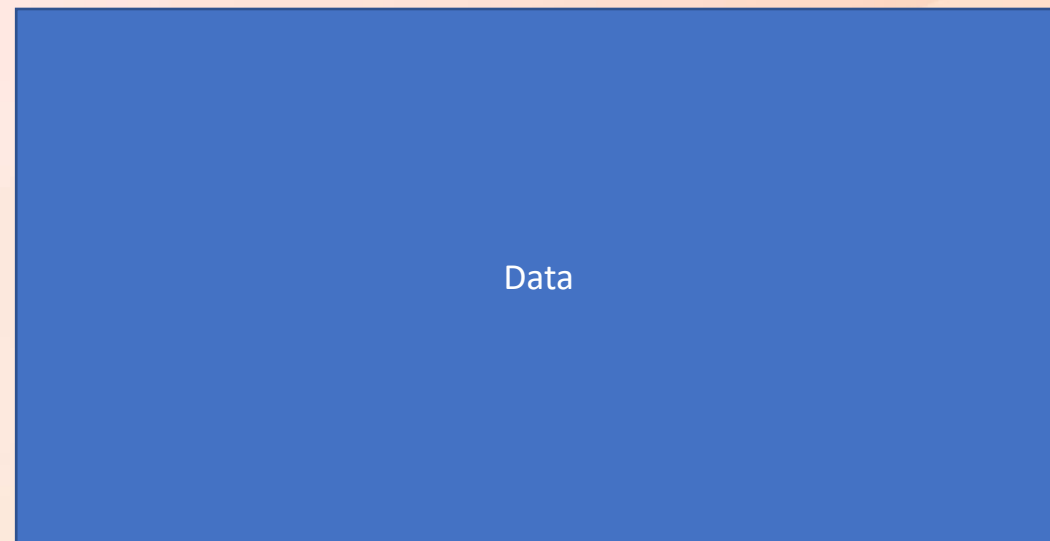
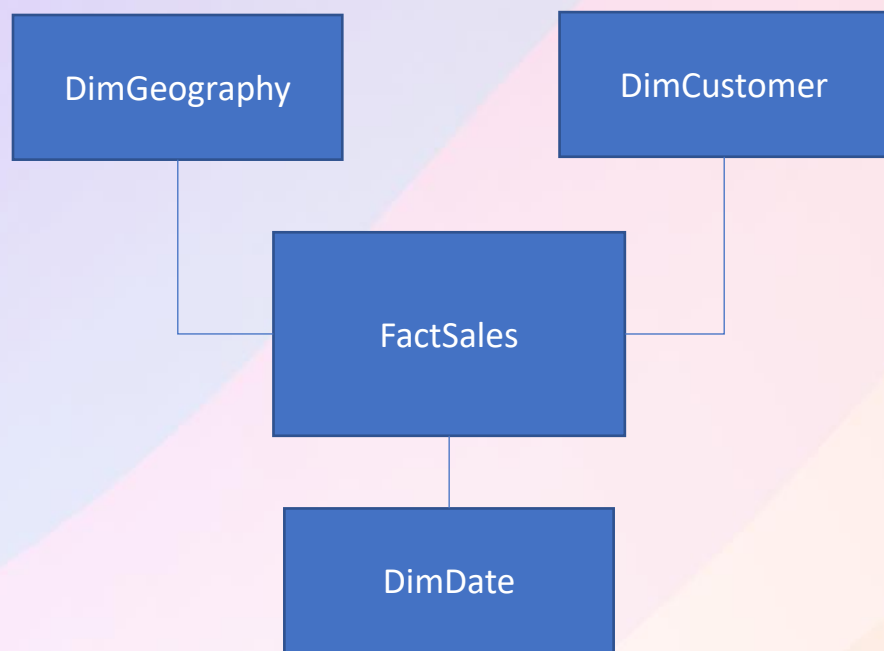
Value	Count
Apple	4
Banana	2
Blueberry	1



#5

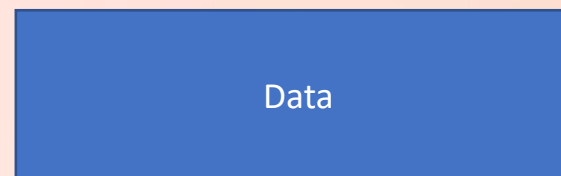
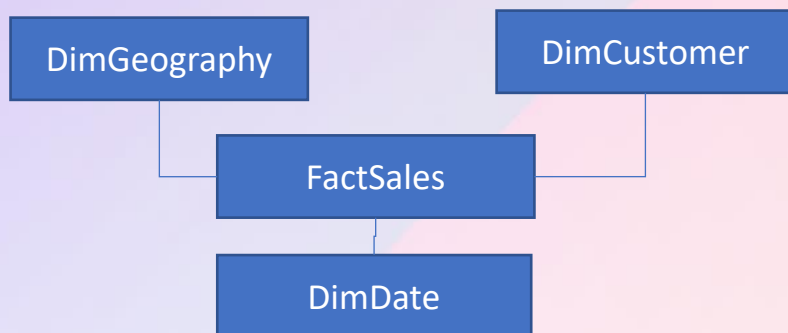
„Płaska struktura jest ok, szkoda czasu na  
strukturę gwiazdy”

# Star schema vs płaska tabela





# Star schema vs płaska tabela



- Możliwość bardziej granularnego odświeżania
- Zoptymalizowany rozmiar modelu
- Operacje wykonywane przez Formula Engine zazwyczaj są szybsze (w stosunku do płaskiej tabeli)
- Operacje opierające się o wymiary i funkcje DISTINCT, VALUES itp. są zoptymalizowane
- Pełny zestaw funkcji języka DAX

- Dłuższe odświeżanie
- Ogromny rozmiar modelu
- Operacje wykonywane przez Storage Engine są często szybsze (w stosunku do Star Schema)
- Operacje opierające się o wymiary i funkcje DISTINCT, VALUES etc potrafią być ekstremalnie wolne
- Bez wymiaru czasu ograniczony Time Intelligence

# Rozszerzanie struktury z tabelą płaską

- Dwie struktury płaskie powodują konieczność tworzenia sztucznych kluczy
- Często wymaga to również relacji wiele do wielu

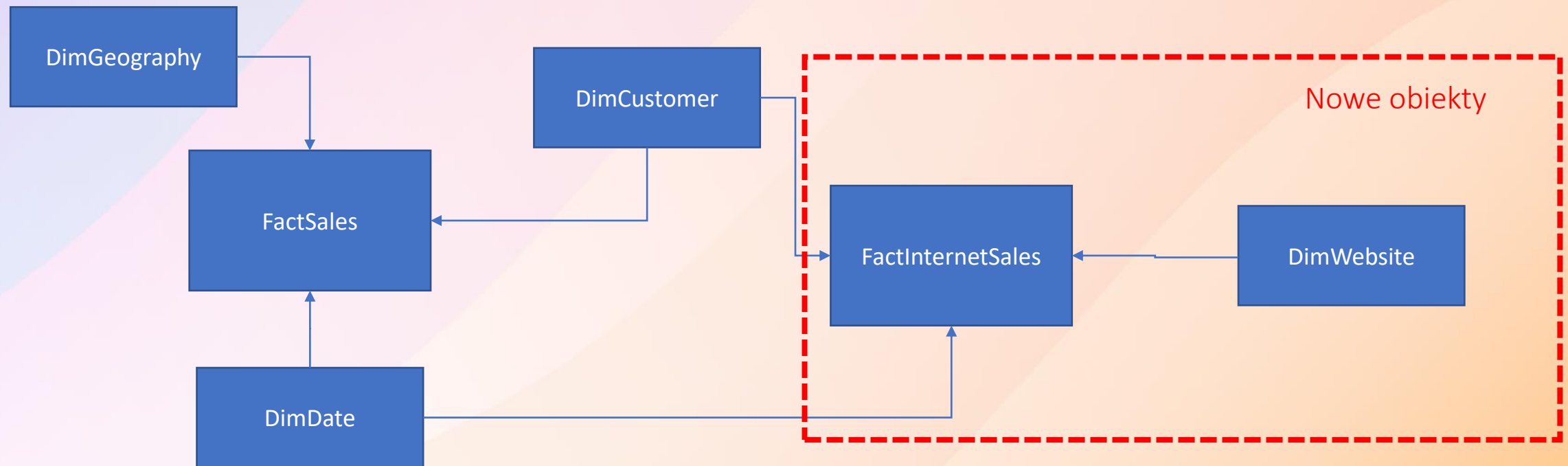
DataId	KlientId	ProduktId	Klucz
20210101	1	1	20210101-1
20210102	1	1	20210102-1
20210102	2	4	20210102-1

Many to  
many(!)

Nowe obiekty			
Klucz	DataId	KlientId	LokalizacjaId
20210101-1	20210101	1	1
20210101-1	20210101	1	1
20210102-2	20210102	2	2
20210102-1	20210102	1	1
20210103-2	20210103	2	2
20210103-2	20210103	2	2

# Rozszerzanie struktury modelem gwiazdy

- Model gwiazdy jest łatwo rozszerzalny
- Nowa tabela faktów podłączona jest do istniejących wymiarów z którymi ma relację oraz dodawane są nowe jeśli jest taka potrzeba
- Takie podejście nie modyfikuje zachowania istniejących miar i nie ma negatywnego wpływu na wydajność



# Rozmiar modelu

Row Labels	Cardinality	Table Size	Columns Total Size	Data Size	Dictionary Size	Columns Hierarchies Size	Bid. Filters	MMR	Encoding
⊕ DimCurrency	105	41,952	41,952	288	39,464	2,200			Many
⊕ DimCustomer	18,484	3,249,686	3,249,686	365,608	2,297,166	586,912			Many
⊕ DimDate	3,652	188,920	188,920	38,080	117,800	33,040			Many
⊕ FactInternetSalesFlat	3,140,696	150,077,065	150,077,065	145,508,040	3,673,801	895,224			Many
⊕ FactInternetSalesStarSchema	3,140,696	31,557,855	31,519,055	29,393,240	1,697,543	428,272			HASH (All)
<b>Grand Total</b>	<b>6,303,633</b>	<b>185,115,478</b>	<b>185,076,678</b>	<b>175,305,256</b>	<b>7,825,774</b>	<b>1,945,648</b>			<b>Many</b>

Row Labels	Relationships Size	RI Violations #	Table Size %	Database Size %
⊕ DimCurrency				0.02%
⊕ DimCustomer				1.76%
⊕ DimDate				0.10%
⊕ FactInternetSalesFlat				81.07%
⊕ FactInternetSalesStarSchema	38,800			17.05%
<b>Grand Total</b>	<b>38,800</b>			<b>100.00%</b>

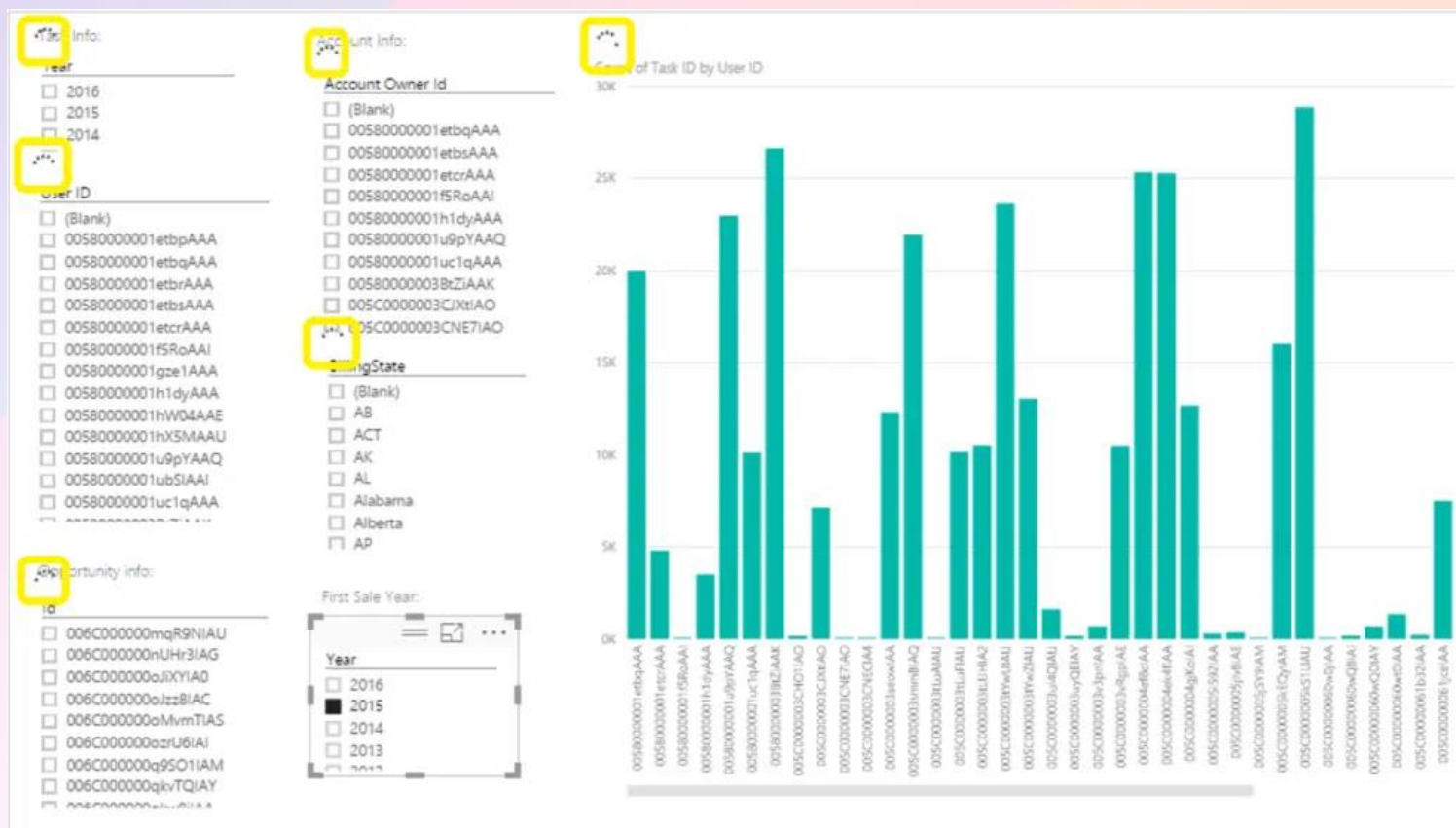


#6

„Front-end nie ma wpływu na wydajność”

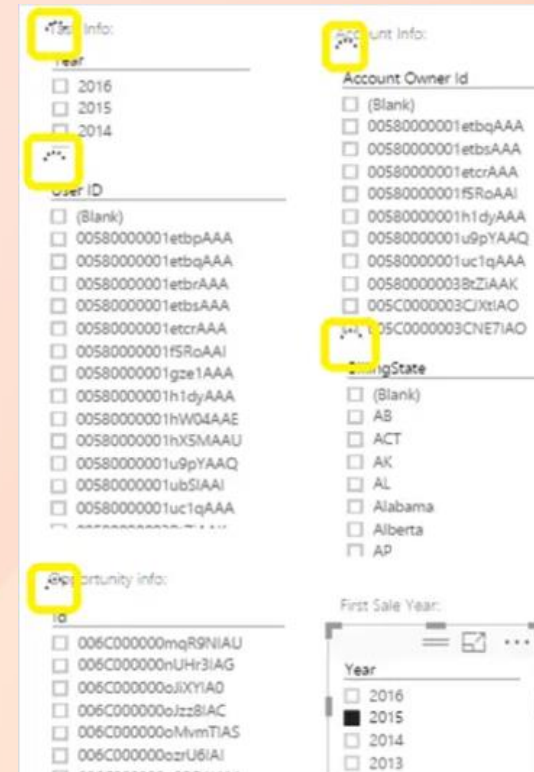


???



# Rzeczy na które należy zwracać uwagę

- Zadbaj o dobrą percepcję raportu w tym:
  - dobierz wizualizację do potrzeb,
  - Opisz wizualizacje tak żeby dane były zrozumiałe,
  - Unikaj dużej liczby slicerów,
  - Unikaj dużej liczby kolorów,
  - Przemyśl nawigację między stronami i zakładkami,
- Każda wizualizacja wysyła jedno lub wiele zapytań,
- Domyślnie każde kliknięcie powoduje ponowne wysłanie zapytania pod wszystkimi wizualizacjami,
- Domyślnie rzecz biorąc wszystkie wizualizacje filtrują siebie nawzajem – warto wyłączyć to zachowanie w **Edit Interactions**
- Synchronizację pomiędzy slicerami można osiągnąć dzięki **Cross-Filter Direction = Both** ale również **dzięki Visual Level Filter na slicerze!**





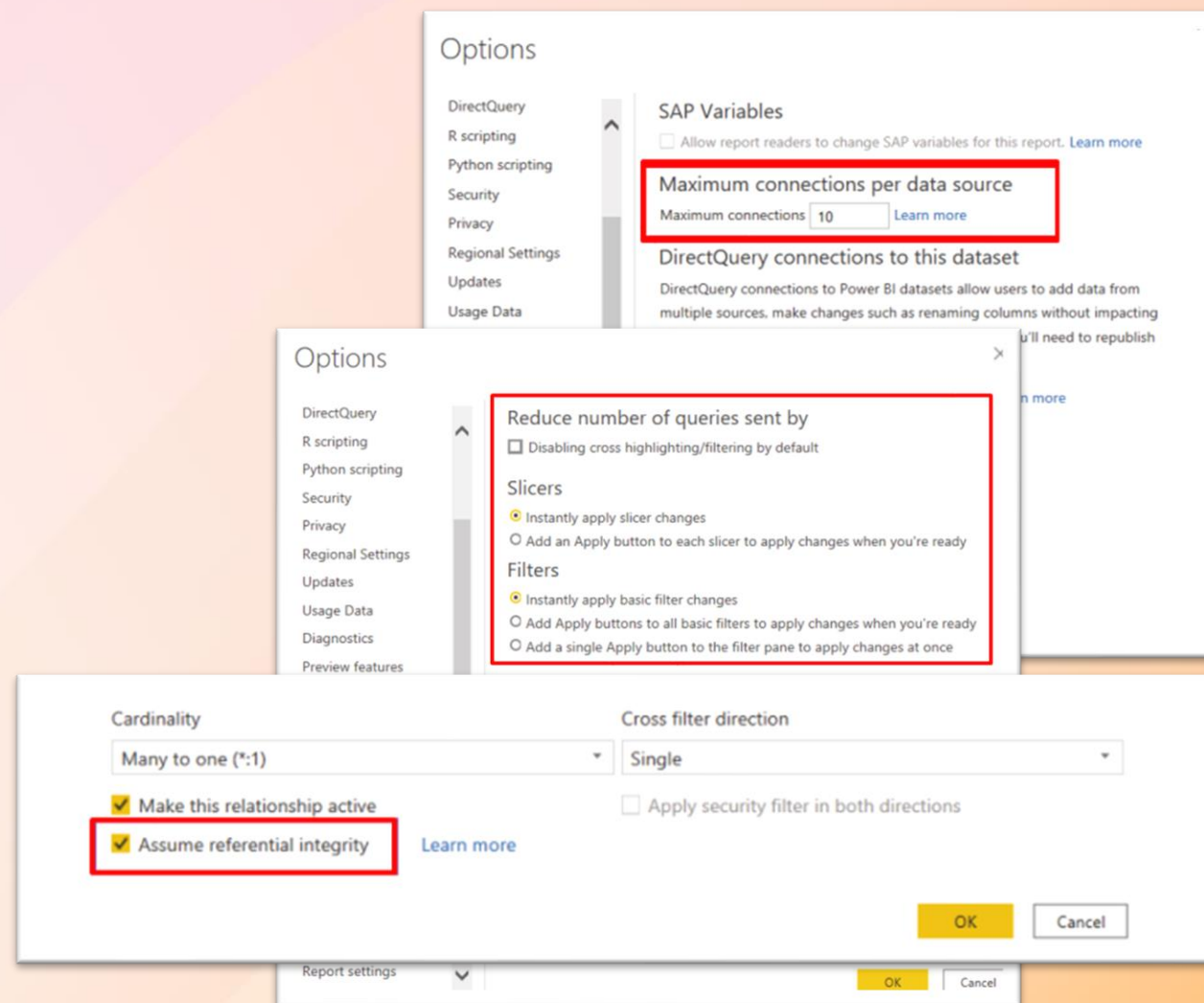
#7

„Direct Query jest wolne  
i należy go unikać”



# Optymalizacja Direct Query

- Twoje rozwiązanie jest tak szybkie jak twoje źródło,
- Optymalizacja relacji w Direct Query opiera się o:
  - Indeksację źródła (Foreign Key)
  - Brak użycia kolumn kalkulowanych jako klucza relacji,
  - Jeśli to możliwe użycia **Assume referential integrity**,
  - Nie używaniu Many-to-Many relationship,
  - Ograniczeniu użycia **Cross filter direction = both**,
- Pamiętaj o **Maximum connections per data source** oraz o tym, że workspace ma limit jeśli chodzi o maksymalną liczbę połączeń które są nieudokumentowane (jedyna możliwość informacji na ten temat to Microsoft Support),
- Warto rozważyć użycie ustawień **Query Reduction**
- **Materializuj** jeśli to tylko możliwe!





#8

„Zrób to na kolumnach kalkulowanych”

# Kolumny kalkulowane

- Materializowane podczas ładowania i/lub odświeżania tabeli
- Kolumny obliczane zazwyczaj nie zapewniają szybszego wykonania zapytań,
- Bardzo często mają niższy stopień kompresji od standardowych kolumn (nie uwzględniane przez VertiPaq w momencie wyszukiwania odpowiedniego sortowania)
- Odświeżenie dowolnego elementu tabeli powoduje konieczność odświeżenia wszystkich kolumn obliczanych w tej tabeli oraz wszystkich tych, które się do niej odwołują,
- Obiekty kalkulowane obliczane są sekwencyjnie przy użyciu pojedynczego wątku – brak skalowalności!

Date	Year	Month	Day
2022-01-01	2022	1	1
2022-01-02	2022	1	2
2022-01-03	2022	1	3
2022-01-04	2022	1	4
2022-01-05	2022	1	5

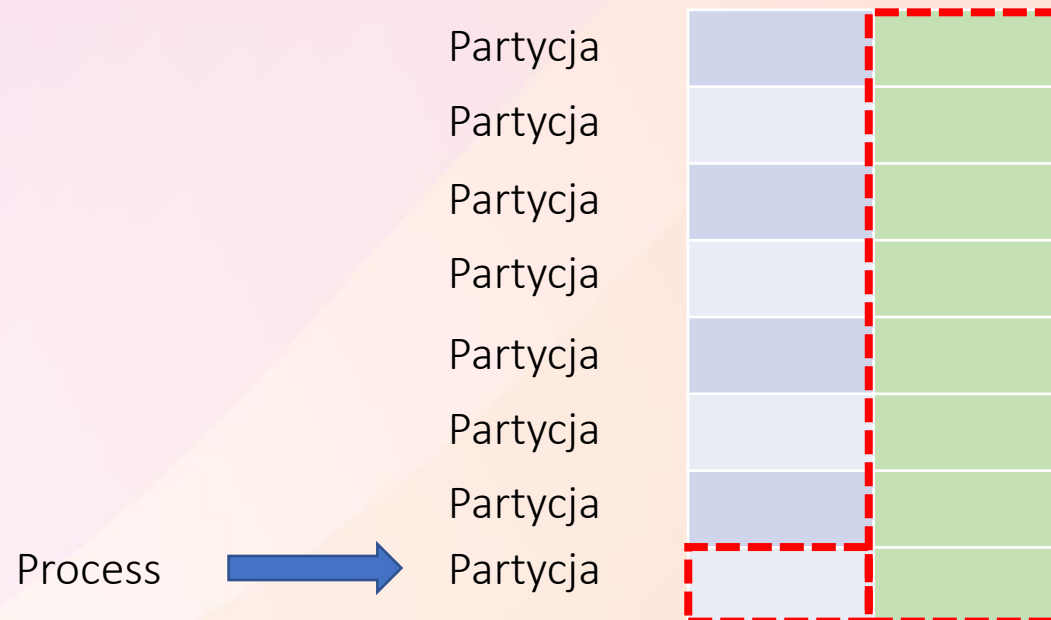


Sekwencja!  
1 wątek!



# Kolumny kalkulowane

- Kolumna kalkulowana zawsze jest obliczana w całości bez względu na partycjonowanie!
- Przy ogromnych modelach może to stanowić wąskie gardło!



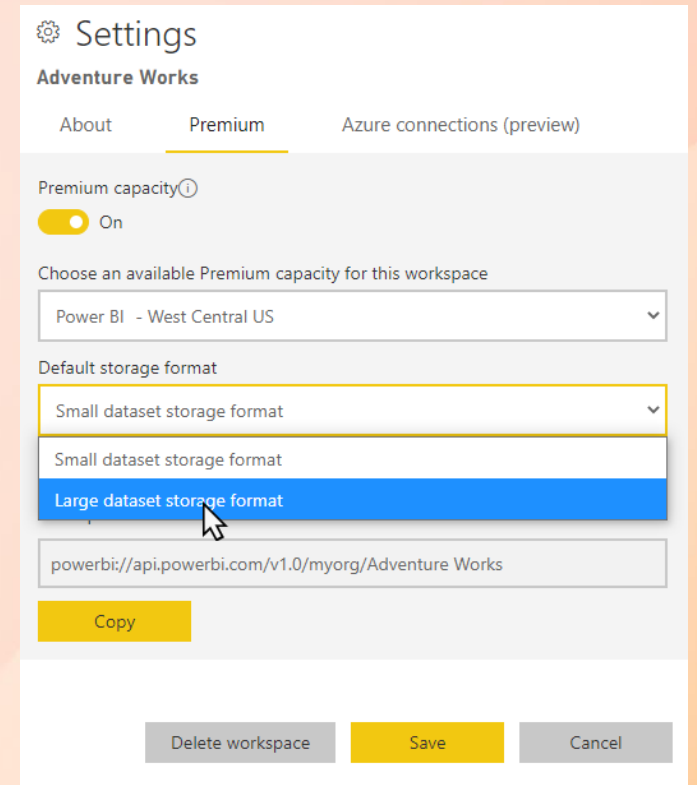


#9

„Power BI nie nadaje się do  
dużych zestawów danych”

# Large models

- Workspace premium pozwala skorzystać z **Large dataset storage format** pozwalający zwiększyć dostępny rozmiar na powyżej 10GB. Dane przechowywane są wtedy na **Azure Premium Storage**.
- Jeśli dataset jest większy niż połowa dostępnych zasobów na workspace – rozważ **Incremental Processing** poprzez **XMLA Endpoint**,
- Power BI zawiera cały szereg mechanizmów, które wspierają przetwarzanie dużych zbiorów danych:
  - Direct Query
  - Composite models
  - Hybrid tables
  - Aggregations
- Power Query nie powinno być wykorzystywane do budowania ciężkich przepływów i transformacji - Power BI integruje się z innymi narzędziami do przetwarzania w skali jak Azure Synapse czy Databricks, które są w stanie te dane przygotować na potrzeby datasetu.



The screenshot shows the 'Settings' page for a workspace named 'Adventure Works'. The 'Premium' tab is selected. Under 'Premium capacity', the toggle is 'On'. A dropdown menu for 'Default storage format' is open, showing 'Small dataset storage format' and 'Large dataset storage format', with the latter selected. Below this is a text field with a URL and a 'Copy' button. At the bottom are 'Delete workspace', 'Save', and 'Cancel' buttons.

Settings  
Adventure Works

About Premium Azure connections (preview)

Premium capacity ⓘ  
On

Choose an available Premium capacity for this workspace  
Power BI - West Central US

Default storage format  
Small dataset storage format  
Large dataset storage format

powerbi://api.powerbi.com/v1.0/myorg/Adventure Works  
Copy

Delete workspace Save Cancel

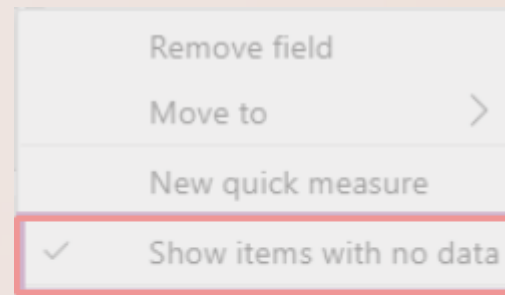


#10

„Wystarczy, że nauczę się  
składni DAX i wszystko działa”

# DAX – dobre praktyki

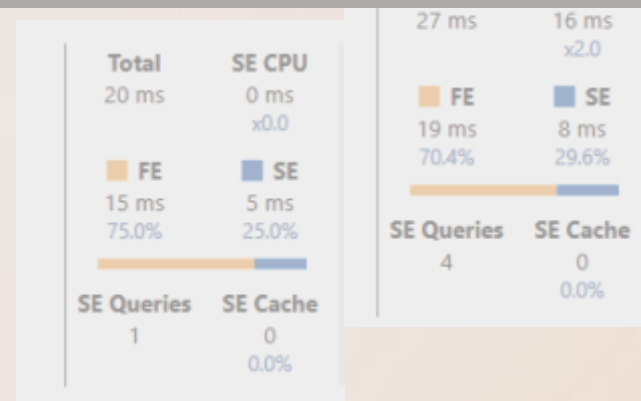
- Język DAX zawiera wiele dobrych praktyk, które są proste w implementacji, a mają duży wpływ na wydajność np...:
- Używanie zmiennych zamiast powtarzania tego samego kodu



Więcej na: [seequality.net](http://seequality.net)

HASONEVALUE oraz VALUES

- Unikanie obsługi błędów (IFERROR oraz ISERROR) na poziomie miary
- Używanie COUNTROWS zamiast COUNT tam gdzie to możliwe (uwaga na wartości BLANK)
- TREATAS > CONTAINS, FILTER, INTERSECT...







# Źródła

---

- <https://pl.seequality.net/>
- <https://sqlbi.com>
- <https://zebrabi.com/power-bi-dashboard-design/>
- <https://blog.enterprisedna.co/dax-query-structure-in-power-bi-two-engines-dax-studio/>
- <https://sqlserverbi.blog/2021/07/19/a-developers-guide-to-creating-bad-power-bi-projects-part-1/>
- <https://maqsoftware.com/insights/power-bi-best-practices>



# 14 edycja konferencji SQLDay

9-11 maja 2022, WROCŁAW + ONLINE



---

partner złoty

---



---

partner srebrny

---



---

partner brązowy

---

