

6. sprawozdanie z laboratorium Hurtownie Danych

Mikołaj Kubś, 272662

4 maja 2025

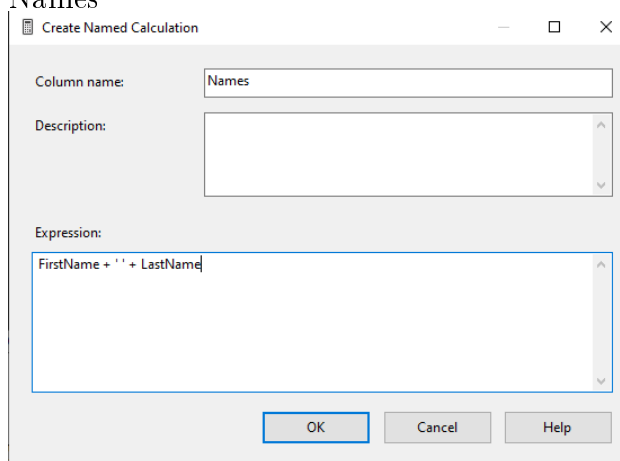
1 Zad. 1. Modyfikacja wymiarów i tabeli faktów

Bazując na kostce utworzonej przy realizacji listy 4, należy:

1.1 Podpunkt a

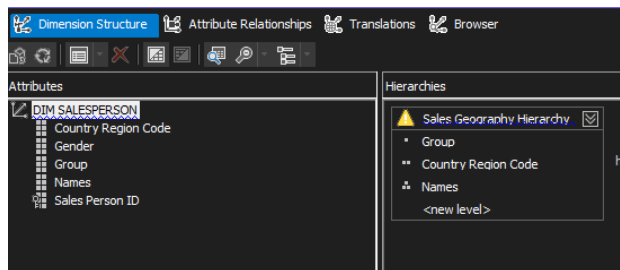
Zmodyfikować definicję wymiarów tak, aby:

1. W wymiarach CUSTOMER i SALESPERSON nie można było korzystać z atrybutów FirstName oraz LastName. W zamian dodać atrybut Names

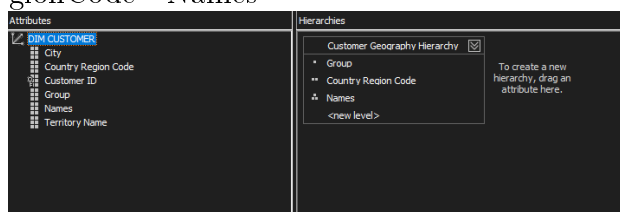


The screenshot shows a 'Create Named Calculation' dialog box. It has three main input fields: 'Column name' with the value 'Names', 'Description' which is empty, and 'Expression' with the formula 'FirstName + '' + LastName'. At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

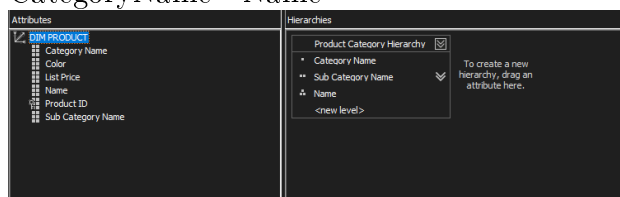
2. W wymiarze SALESPERSON pojawiła się hierarchia Group - CountryRegionCode - Names



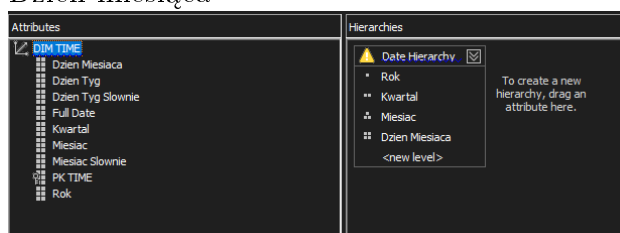
3. W wymiarze CUSTOMER pojawiła się hierarchia Group - CountryRegionCode - Names



4. W wymiarze PRODUCT pojawiła się hierarchia CategoryName - Sub-CategoryName - Name



5. W wymiarze TIME pojawiła się hierarchia Rok - Kwartał - Miesiąc - Dzień miesiąca



1.2 Podpunkt b

Dla każdego atrybutu kluczowego wymiaru, którego wartościami są liczby całkowite, zmodyfikować właściwości (Properties). Zmodyfikować parametr NameColumn, tak aby nazwy kolejnych elementów wymiaru nie były liczbami. (Przykładowo dla wymiaru dotyczącego Produktu można wykorzystać atrybut Name).

Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	DIM_CUSTOMER.CustomerID (Integer)
NameColumn	DIM_CUSTOMER.Names (WChar)
ValueColumn	(none)

Rysunek 1: Widok Properties dla DIM_Salesperson

Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	DIM_CUSTOMER.CustomerID (Integer)
NameColumn	DIM_CUSTOMER.Names (WChar)
ValueColumn	(none)

Rysunek 2: Widok Properties dla DIM_Customer

Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	DIM_PRODUCT.ProductID (Integer)
NameColumn	DIM_PRODUCT.Name (WChar)
ValueColumn	(none)

Rysunek 3: Widok Properties dla DIM_Product

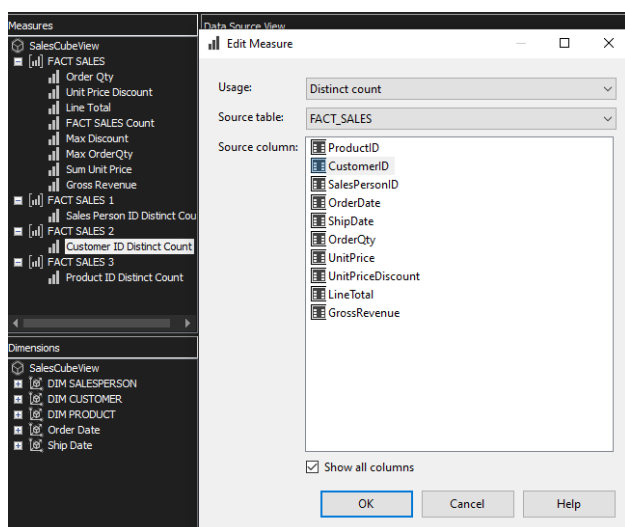
Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	DIM_TIME.PK_TIME (Integer)
NameColumn	DIM_TIME.FullDate (WChar)
ValueColumn	(none)

Rysunek 4: Widok Properties dla DIM_Time

1.3 Podpunkt c

Utworzyć nowe miary, które będą odzwierciedlać:

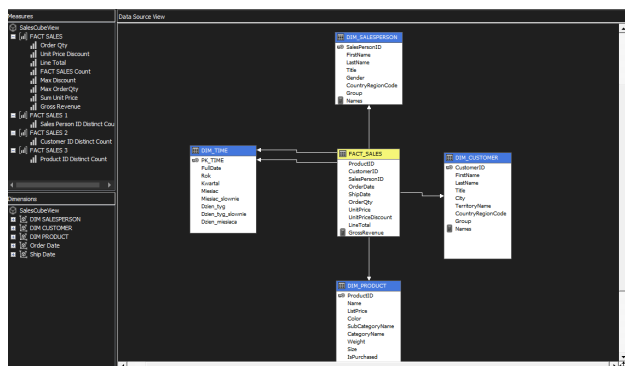
- Liczbę różnych klientów (aggregatedFunction: distinct count)
- Liczbę różnych produktów
- Maksymalną wartość rabatu (aggregatedFunction: max)
- Maksymalną liczbę zamówionych produktów
- Liczbę różnych sprzedawców realizujących zamówienia



Rysunek 5: Miara dotycząca liczby różnych klientów

1.4 Podpunkt d

Wdrożyć i przetworzyć kostkę.



Rysunek 6: Widok przetworzonej kostki

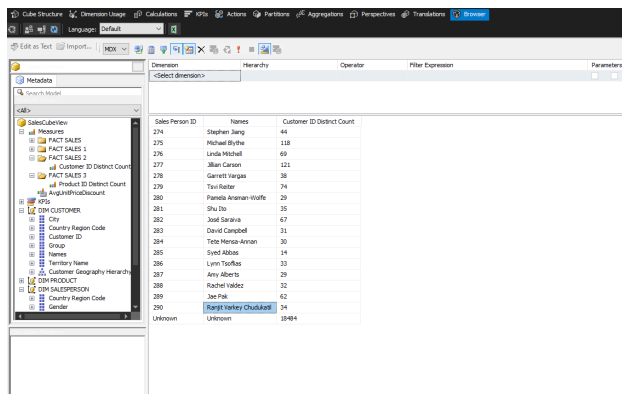
2 Zad. 2. Przegląd danych i tworzenie zestawień

Przy użyciu zakładki Browser:

2.1 Podpunkt a

Sprawdzić, czy dane zapisane w kostce zgadzają się z danymi zapisanymi w tabelach, przeciągając za pomocą myszy:

- atrybuty wymiarów w region wierszy
- miary w część centralną widoku

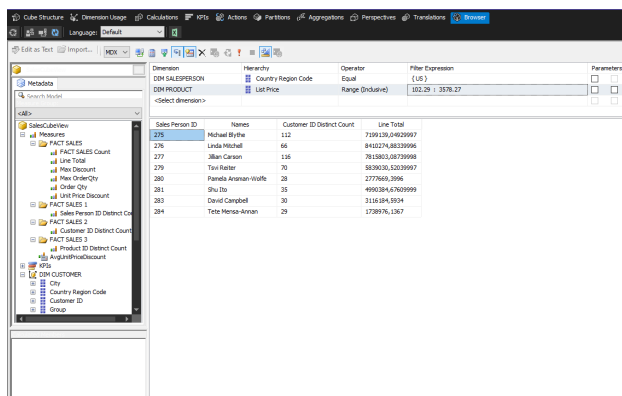


Sales Person ID	Names	Customer ID District Count
274	Stephen Jung	44
275	Michael Blythe	118
276	Linda Mitchell	69
277	Ellen Cannon	121
278	Garnett Vargas	38
279	Toni Kaler	74
280	Pamela Ansman-Wolfe	29
281	Shu-Ita	35
282	Joel Saravia	67
283	David Campbell	71
284	Tate Henshaw	30
285	Syed Abbas	14
286	Lynn Tsofan	33
287	Amy Alberts	29
288	Rachel Valdez	32
289	Jon Pelt	42
290	Rajesh Kulkarni	24
Unknown	Unknown	18484

Rysunek 7: Widok przykładowej kwerendy w Browser

2.2 Podpunkt b

Przetestować możliwości przeglądarki (Browser) - operator wyboru danych (Operator), wyrażenia filtrujące dane (Filter Expression) itp.



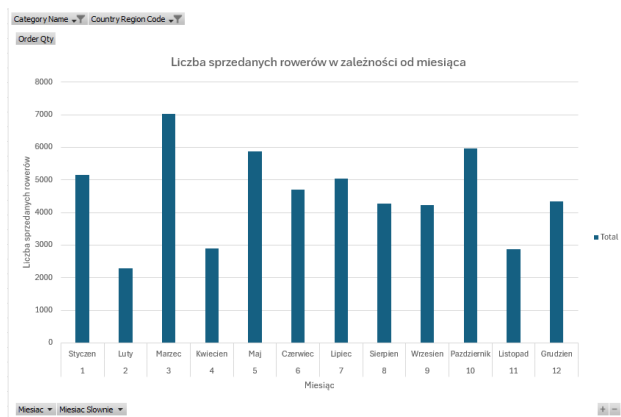
Sales Person ID	Names	Customer ID District Count	Line Total
275	Michael Blythe	112	7199126,04229997
276	Linda Mitchell	66	8402274,38239996
277	Ellen Cannon	125	7618820,08739996
279	Toni Kaler	70	5839026,52039997
280	Pamela Ansman-Wolfe	28	2777669,3996
281	Shu-Ita	35	4990284,67602999
283	David Campbell	30	3116184,9524
284	Tate Henshaw	29	1738976,1367

Rysunek 8: Widok przykładowej kwerendy z dwoma różnymi rodzajami fil-trów (Operator i Filter Expression)

2.3 Podpunkt c

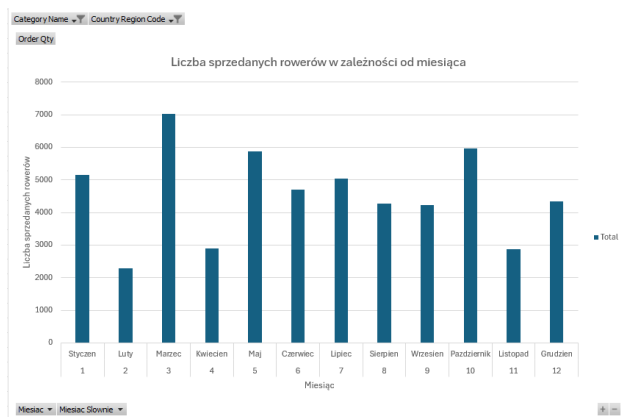
Przygotować przykładowe tabele i wykresy przestawne oraz zinterpretować uzyskane wyniki (proszę zapisać wnioski!)

2.3.1 Rowery



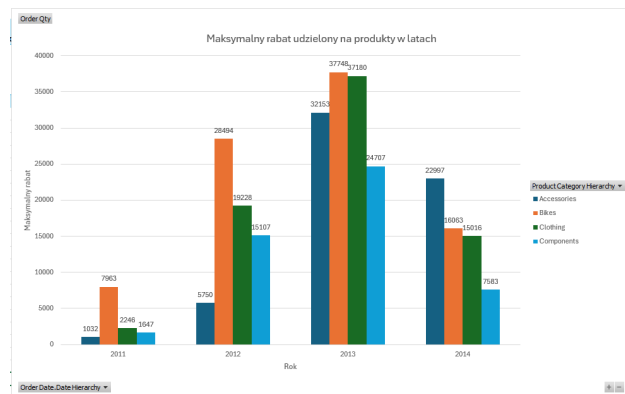
Rysunek 9: Wykres

2.3.2 Rowery



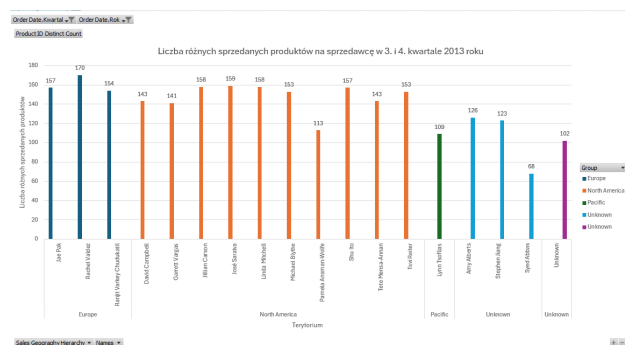
Rysunek 10: Wykres

2.3.3 Zniżka



Rysunek 11: Wykres

2.3.4 Sprzedawca



Rysunek 12: Wykres

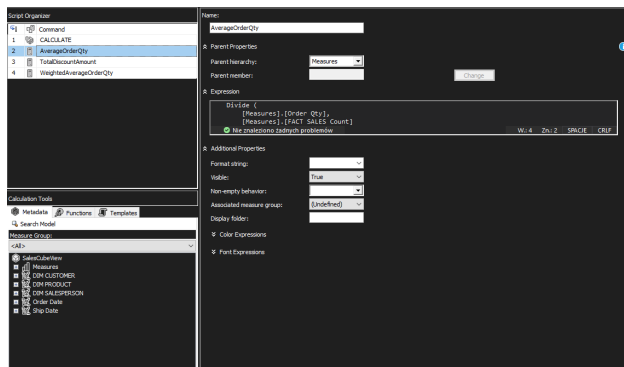
3 Zad. 3. Miary kalkulowane

W zakładce Calculations dodać dwie miary kalkulowane (ang. calculated members):

- średnią liczbę zamówionych towarów na zamówienie
- średnią ważoną liczbę towarów na zamówienie. Jako wagę należy wybrać cenę danego produktu.

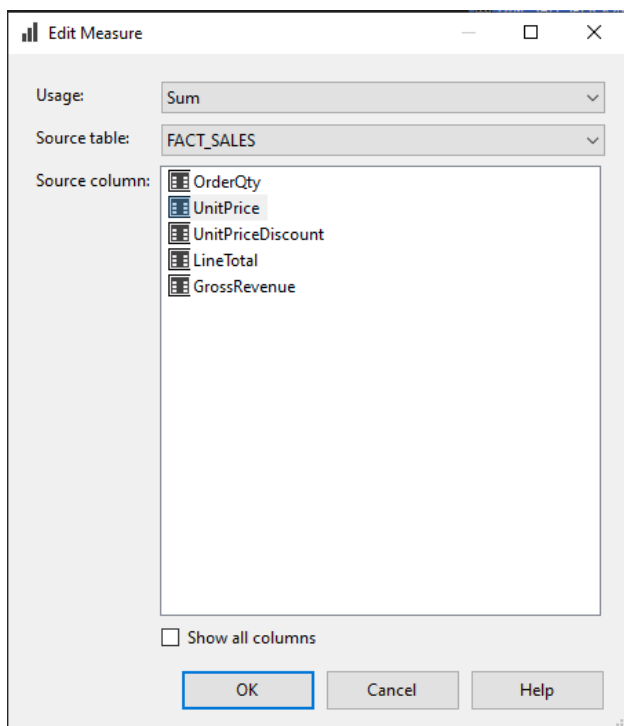
Wskazówka: w celu utworzenia wyżej wymienionej średniej ważonej można posłużyć się nową kolumną zdefiniowaną w widoku źródła danych (lub w

tabeli). Kolumna ta powinna definiować miarę pomocniczą, która pozwoli uzyskać fragment wyrażenia odpowiadającego średniej ważonej.

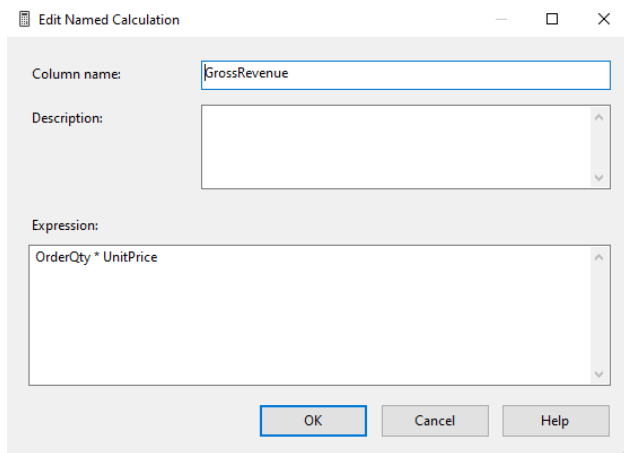


Rysunek 13: Sposób obliczania miary ze zwykłą średnią

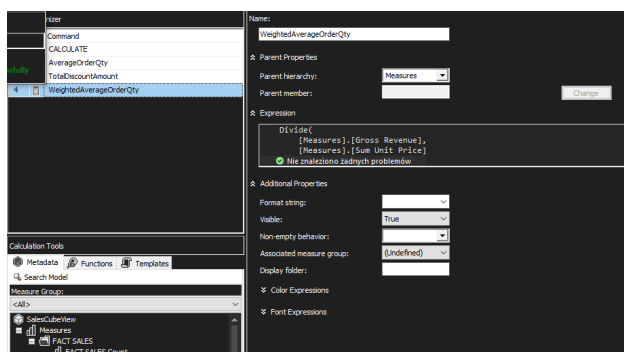
Do obliczenia średniej ważonej należało dodać miarę obliczającą sumę ceny jednostkowej i drugą miarę, będącą iloczynem ceny jednostkowej i liczby zamówionego produktu (LineTotal prawie to spełniał, ale miał w sobie czasem zniżkę).



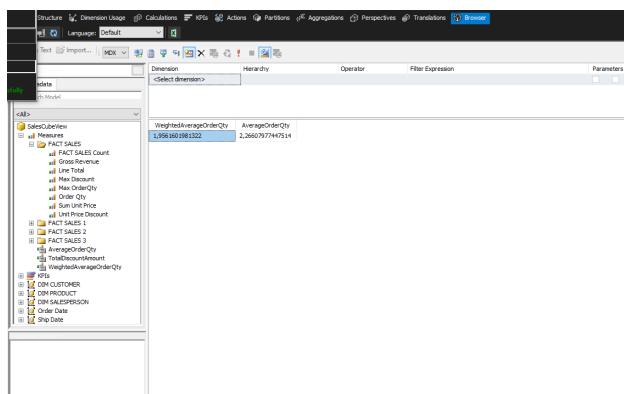
Rysunek 14: Sposób obliczania miary z sumą ceny jednostkowej



Rysunek 15: Sposób obliczania miary zysku brutto



Rysunek 16: Sposób obliczania miary średniej ważonej



Rysunek 17: Wynik średnich dla całego zbioru danych

4 Zad. 4. Partycje

Podzielić zawartość kostki na partycje (zakładka Partitions). Każda partycja powinna odzwierciedlać jeden rok. Istnieją dwa podstawowe sposoby podziału partycjonowania kostek:

- dane do zasilania poszczególnych partycji znajdują się w osobnych tabelach
- dane do zasilania poszczególnych partycji znajdują się w tej samej tabeli, zaś każda z partycji ma przypisanie zapytanie SQL, którego wynik służy do jej zasilenia.

Proszę przygotować partycje na dwa sposoby i znaleźć uzasadnienie dla każdej opcji.

4.1 Sposób pierwszy

Ta metoda wymagała utworzenia w bazie SQL Server osobnych tabel dla każdego roku (np. 'FACT_SALES_2011', 'FACT_SALES_2012', itd.) o strukturze identycznej jak oryginalna tabela faktów, a następnie wypełnienia ich danymi z odpowiednich lat.

```
1 CREATE TABLE Kubs.FACT_SALES_2011 (  
2     ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),  
3     CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(  
4         CustomerID),  
5     SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(  
6         SalesPersonID),  
7     OrderDate INT NOT NULL,  
8     ShipDate INT NULL,  
9     OrderQty SMALLINT NOT NULL,  
10    UnitPrice MONEY NOT NULL,  
11    UnitPriceDiscount DECIMAL(8, 4) NOT NULL,  
12    LineTotal DECIMAL(19, 4) NOT NULL  
13 );  
14  
15 CREATE TABLE Kubs.FACT_SALES_2012 (  
16     ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),  
17     CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(  
18         CustomerID),  
19     SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(  
20         SalesPersonID),  
21     OrderDate INT NOT NULL,
```

```

18     ShipDate INT NULL,
19     OrderQty SMALLINT NOT NULL,
20     UnitPrice MONEY NOT NULL,
21     UnitPriceDiscount DECIMAL(8, 4) NOT NULL,
22     LineTotal DECIMAL(19, 4) NOT NULL
23 );
24
25 CREATE TABLE Kubs.FACT_SALES_2013 (
26     ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),
27     CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(
28     CustomerID),
29     SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(
30     SalesPersonID),
31     OrderDate INT NOT NULL,
32     ShipDate INT NULL,
33     OrderQty SMALLINT NOT NULL,
34     UnitPrice MONEY NOT NULL,
35     UnitPriceDiscount DECIMAL(8, 4) NOT NULL,
36     LineTotal DECIMAL(19, 4) NOT NULL
37 );
38
39 CREATE TABLE Kubs.FACT_SALES_2014 (
40     ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),
41     CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(
42     CustomerID),
43     SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(
44     SalesPersonID),
45     OrderDate INT NOT NULL,
46     ShipDate INT NULL,
47     OrderQty SMALLINT NOT NULL,
48     UnitPrice MONEY NOT NULL,
49     UnitPriceDiscount DECIMAL(8, 4) NOT NULL,
50     LineTotal DECIMAL(19, 4) NOT NULL
51 );
52
53 with Sales1 AS (
54     SELECT
55     Kubs.FACT_SALES.ProductID,
56     Kubs.FACT_SALES.CustomerID,
57     Kubs.FACT_SALES.SalesPersonID,
58     Kubs.FACT_SALES.OrderDate,
59     Kubs.FACT_SALES.ShipDate,
60     Kubs.FACT_SALES.OrderQty,

```

```

57     Kubs.FACT_SALES.UnitPrice,
58     Kubs.FACT_SALES.UnitPriceDiscount,
59     Kubs.FACT_SALES.LineTotal
60 FROM Kubs.FACT_SALES
61 WHERE OrderDate >= 20110101 AND OrderDate < 20120000
62 )
63 INSERT INTO Kubs.FACT_SALES_2011
64
65 SELECT * FROM Sales1;
66
67 with Sales2 AS (
68     SELECT
69         Kubs.FACT_SALES.ProductID,
70         Kubs.FACT_SALES.CustomerID,
71         Kubs.FACT_SALES.SalesPersonID,
72         Kubs.FACT_SALES.OrderDate,
73         Kubs.FACT_SALES.ShipDate,
74         Kubs.FACT_SALES.OrderQty,
75         Kubs.FACT_SALES.UnitPrice,
76         Kubs.FACT_SALES.UnitPriceDiscount,
77         Kubs.FACT_SALES.LineTotal
78     FROM Kubs.FACT_SALES
79     WHERE OrderDate >= 20120101 AND OrderDate < 20130000
80 )
81 INSERT INTO Kubs.FACT_SALES_2012
82
83 SELECT * FROM Sales2;
84
85 with Sales3 AS (
86     SELECT
87         Kubs.FACT_SALES.ProductID,
88         Kubs.FACT_SALES.CustomerID,
89         Kubs.FACT_SALES.SalesPersonID,
90         Kubs.FACT_SALES.OrderDate,
91         Kubs.FACT_SALES.ShipDate,
92         Kubs.FACT_SALES.OrderQty,
93         Kubs.FACT_SALES.UnitPrice,
94         Kubs.FACT_SALES.UnitPriceDiscount,
95         Kubs.FACT_SALES.LineTotal
96     FROM Kubs.FACT_SALES
97     WHERE OrderDate >= 20130101 AND OrderDate < 20140000
98 )
99 INSERT INTO Kubs.FACT_SALES_2013

```

```

100
101 SELECT * FROM Sales3;
102
103 with Sales4 AS (
104     SELECT
105         Kubs.FACT_SALES.ProductID,
106         Kubs.FACT_SALES.CustomerID,
107         Kubs.FACT_SALES.SalesPersonID,
108         Kubs.FACT_SALES.OrderDate,
109         Kubs.FACT_SALES.ShipDate,
110         Kubs.FACT_SALES.OrderQty,
111         Kubs.FACT_SALES.UnitPrice,
112         Kubs.FACT_SALES.UnitPriceDiscount,
113         Kubs.FACT_SALES.LineTotal
114     FROM Kubs.FACT_SALES
115     WHERE OrderDate >= 20140101
116 )
117 INSERT INTO Kubs.FACT_SALES_2014

```

Listing 1: Tworzenie i wypełnianie tabeli DIM_TIME.

Następnie należało dodać każdą z tabel do projektu, a potem do partycji w kostce.

Item	Partition Name	Source	Estimated Rows	Storage Mode	Aggregation Design
1	FACT_SALES_2011	FACT_SALES_2011	0	HOLAP	
2	FACT_SALES_2012	FACT_SALES_2012	2569	HOLAP	Aggregation Design
3	FACT_SALES_2013	FACT_SALES_2013	0	HOLAP	
4	FACT_SALES_2014	FACT_SALES_2014	0	HOLAP	

Rysunek 18: Dodane partycje

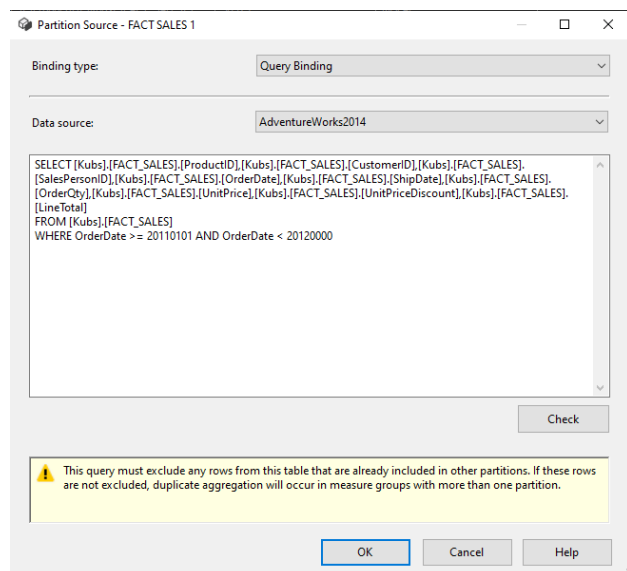
Year	Count	Fact_Sales_Count
2011	1718	
2012	2189	
2013	5573	
2014	3739	

Rysunek 19: Wynik

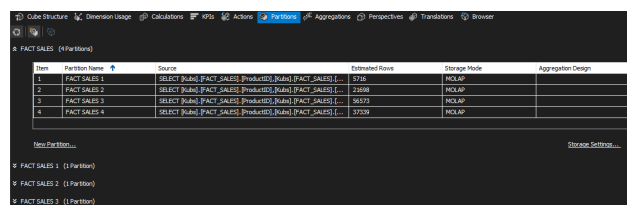
Uzasadnienie: Lepsza izolacja danych, potencjalnie szybsze przetwarzanie partycji (odczyt z mniejszych tabel), zgodność z procesami ETL ładującymi dane rocznie, większa możliwość automatyzacji procesu, zwłaszcza dla zmieniających się danych - tak jak tutaj, dla lat.

4.2 Sposób drugi

W tej metodzie wszystkie dane pozostają w oryginalnej tabeli 'FACT_SALES'. Partycje tworzone są przez zdefiniowanie zapytania SQL (query binding) dla każdej z nich, które wybiera dane tylko dla konkretnego roku za pomocą klauzuli 'WHERE'.



Rysunek 20: Zmieniony kod SQL w partycji - dodana klauzula WHERE ograniczająca daty



Rysunek 21: Dodane partycje

Rok	FACT SALES Count
2011	5716
2012	12089
2013	95573
2014	37329

Rysunek 22: Wynik

Uzasadnienie: Brak konieczności tworzenia dodatkowych tabel w bazie SQL, prostsze zarządzanie bazą danych, łatwiejsza zmiana kryteriów partycjonowania (modyfikacja zapytań).

5 Zad. 5. * Definiowanie KPI

5.1 Prosty wskaźnik KPI

Przygotować wskaźnik KPI (zakładka *KPI*), która umożliwi podział klientów na dobrych i lepszych w zależności od liczby sztuk zamówionych produktów.

Tworząc nowy wskaźnik należy podać jego nazwę, wybrać (przeciągnąć) miarę, na podstawie której będzie dokonany podział zbioru, wybrać odpowiedni status (np. *Shapes*) i podać warunek:

$$\text{iif}([\text{Measures}].[OrderQty] < \eta, -1 \text{ /*czerwony*/}, 1 \text{ /*zielony*/})$$

Należy uzasadnić wybór wartości progowej η .

Po przetworzeniu kostki, należy zobrazować działanie wskaźnika dla wybranych atrybutów w raporcie w Excelu.

Postanowiono przyjąć KPI dzielące klientów na 2 grupy - około top 20% "elitarnych" klientów i całą resztę.

```

1 SELECT TOP 20 PERCENT OrderQty
2 FROM Kubs.FACT_SALES
3 ORDER BY OrderQty DESC;
```

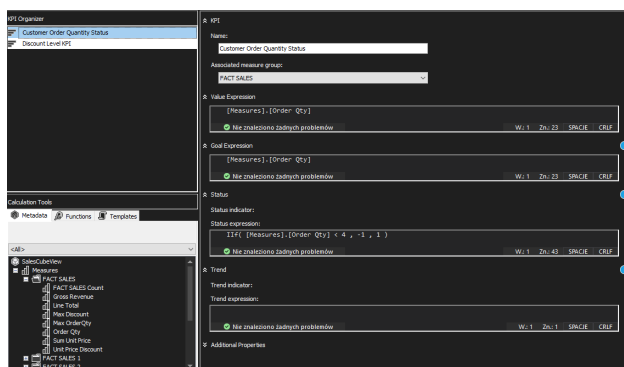
Listing 2: Kwerenda znajdująca dane do wyliczenia KPI

Po krótkiej analizie wyników okazało się, że wartość progową η można wyznaczyć jako 4. Tak więc kliencie, którzy kupili co najmniej 4 przedmioty, są elitarni.

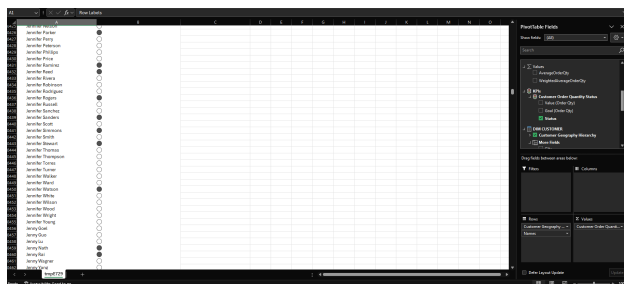
Wybór wartości granicznej równej 20% wynika z konceptu „krzywej wieloryba” (ang. *whale curve*), gdzie:

- Top 20% klientów generuje ponad 100% zysku,
- Środkowe 60% klientów przynosi niewielki zysk lub bilansuje się,
- Najsłabsze 20% klientów generuje straty, przez co łączny zysk netto wraca do poziomu poniżej 100%.

Koncepcja ta pomaga zrozumieć, że niewielka część klientów odpowiada za większość rentowności firmy, co uzasadnia przyjęcie progu 20% w konstrukcji KPI. [1]



Rysunek 23: Sposób obliczenia KPI



Rysunek 24: Tabela przestawna z widokiem na wartości KPI

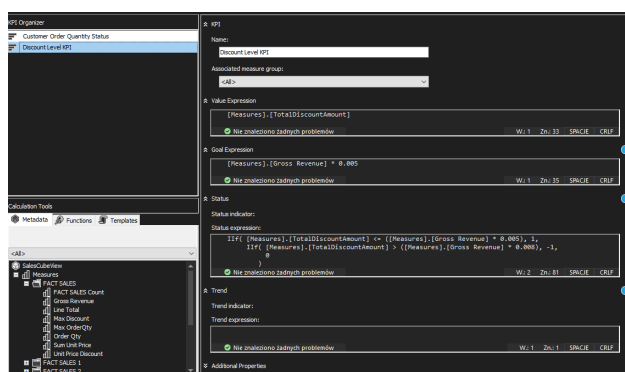
5.2 KPI na podstawie kalkulowanej miary

Zaproponować własną miarę w zakładce *Calculation* → *New Calculated Member*, (np. zysk z uwzględnieniem rabatu i frachtu), na podstawie której zostanie zdefiniowany odpowiedni wskaźnik KPI. Należy przeanalizować status

opracowanego wskaźnika oraz jego trend. Wynik należy zaprezentować w wybranym kontekście.

Zdefiniowano własną miarę kalkulowaną ‘Total Discount Amount’ (Całkowita Wartość Rabatu) jako różnicę między przychodem brutto (wymagało użycia dodanej wcześniej miary ‘Gross Revenue’) a przychodem netto (‘Line Total’).

Na podstawie tej miary stworzono ‘Discount Level KPI’, który ocenia poziom rabatu względem przychodu brutto (cel: $\leq 0.5\%$, źle: $> 0.8\%$) oraz pokazuje trend porównując wartość rabatu do roku poprzedniego.



Rysunek 25: Sposób obliczenia KPI

Row Labels	Gross Revenue	TotalDiscountAmount	Discount Level KPI Goal	Discount Level KPI Status
Accessories	1278760,912	6688,0528	6393,804562	●
Bikes	95145813,35	494640,6947	475729,0668	●
Clothing	2141507,024	20964,5468	10707,53512	○
Components	11807808,02	5214,742	59039,04012	●
Unknown				●
Grand Total	110373889,3	527508,0363	551869,4466	●

Rysunek 26: Tabela przestawna z widokiem na wartości KPI

Column Labels													Total Discount Value
Row Labels	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
Row Labels	Discount Level 40% (1)	Discount Level 40% (2)	Discount Level 40% (3)	Discount Level 40% (4)	Discount Level 40% (5)	Discount Level 40% (6)	Discount Level 40% (7)	Discount Level 40% (8)	Discount Level 40% (9)	Discount Level 40% (10)	Discount Level 40% (11)	Discount Level 40% (12)	Discount Level 40% (13)
4 - Domestic	264,233,938	0	127,088,622	0	339,322,561	0	0	0	0	237,122,618	0	0	0
4 - Foreign	309,943,865	0	145,807,348	0	455,751,213	0	0	0	0	279,943,369	0	0	0
4 - Shipping	289,610,171	0	139,435,972	0	429,046,143	0	0	0	0	259,599,463	0	0	0
4 - Domestic	339,882,260	0	168,898,968	0	498,781,228	0	0	0	0	338,681,236	0	0	0
4 - Foreign	0	0	0	0	0	0	0	0	0	0	0	0	0

Modyfikacja wymiarów i tworzenie hierarchii znacząco poprawia czytelność i możliwości analizy danych w kostce OLAP, umożliwiając drążenie danych.

Miary kalkulowane pozwalają na definiowanie bardziej złożonych wskaźników biznesowych bezpośrednio w kostce przy użyciu MDX, bez konieczności modyfikacji źródła danych.

Różnica między średnią arytmetyczną a ważoną może być znacząca w zależności od kontekstu biznesowego. W tym biznesie raczej większe znaczenie będzie miała dobra średnia ważona, gdyż przekazuje ona więcej informacji o znaczeniu biznesowym.

Partycjonowanie tabel faktów jest kluczową techniką optymalizacyjną, szczególnie dla dużych hurtowni danych, przyspieszając przetwarzanie i potencjalnie zapytania. Wybór metody partycjonowania (osobne tabele vs zapytania) zależy od specyfiki systemu źródłowego i procesów ETL.

Definiowanie KPI umożliwia szybką wizualną ocenę kluczowych wskaźników biznesowych oraz ich trendów, co jest niezwykle przydatne w raportowaniu i analizie menedżerskiej. Z drugiej strony, trzeba uważać na złe KPI - stworzona w drugim podpunkcie KPI może wprowadzić w błąd trendem ujemnym. W tym przypadku po głębszej analizie okazuje się, że ponieważ zyski sklepu i jego popularność rosną, to zwiększyła się liczba zniżek. Można by ulepszyć KPI, by trend brał pod uwagę zmiany w sprzedaży ogólnie - aktualnie tego nie robi.

Bibliografia

- [1] Baker Tilly. *Visualizing Customer Profitability with the Whale Curve*. Accessed: 2025-05-04. 2021. URL: <https://www.bakertilly.com/insights/visualizing-customer-profitability-with-the-whale-curve>.