

6. sprawozdanie z laboratorium Hurtownie Danych

Mikołaj Kubś, 272662

5 maja 2025

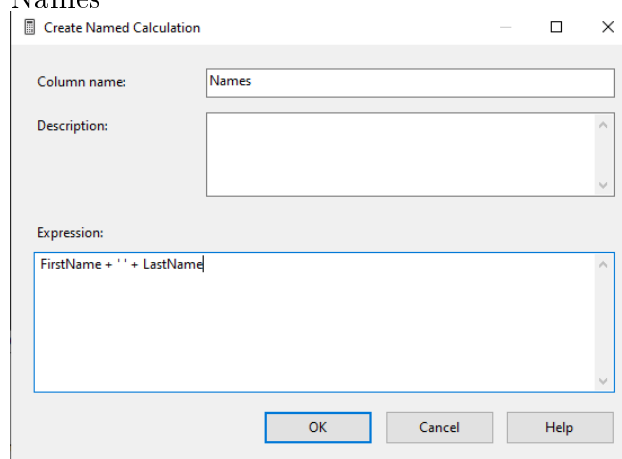
1 Zad. 1. Modyfikacja wymiarów i tabeli faktów

Bazując na kostce utworzonej przy realizacji listy 4, należy:

1.1 Podpunkt a

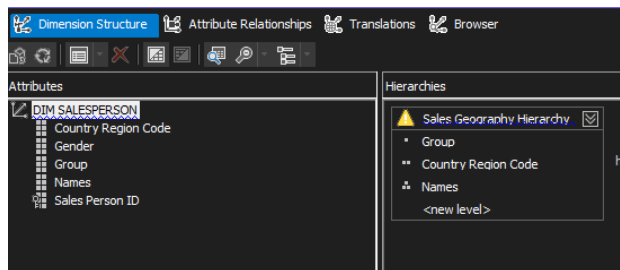
Zmodyfikować definicję wymiarów tak, aby:

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

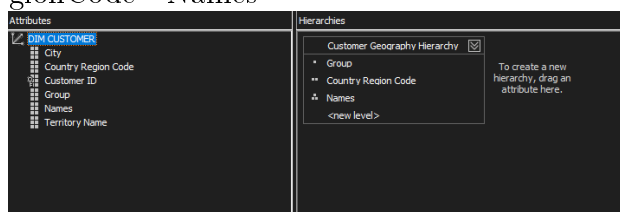


The screenshot shows a 'Create Named Calculation' dialog box. It has three main input fields: 'Column name' with the value 'Names', 'Description' which is empty, and 'Expression' with the formula 'FirstName + '' + LastName'. At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

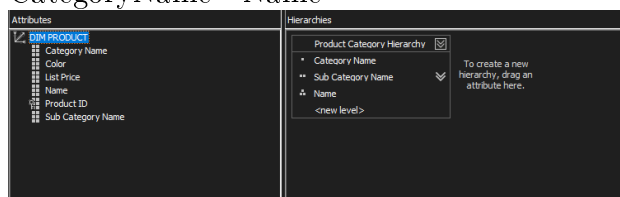
2. W wymiarze SALESPERSON pojawiła się hierarchia Group - CountryRegionCode - Names



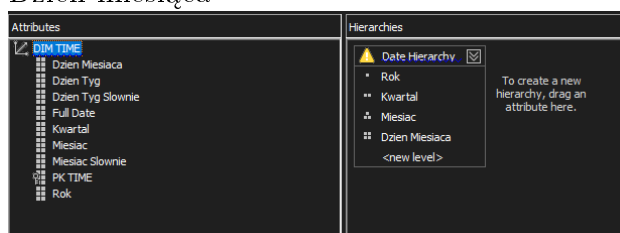
3. W wymiarze CUSTOMER pojawiła się hierarchia Group - CountryRegionCode - Names



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

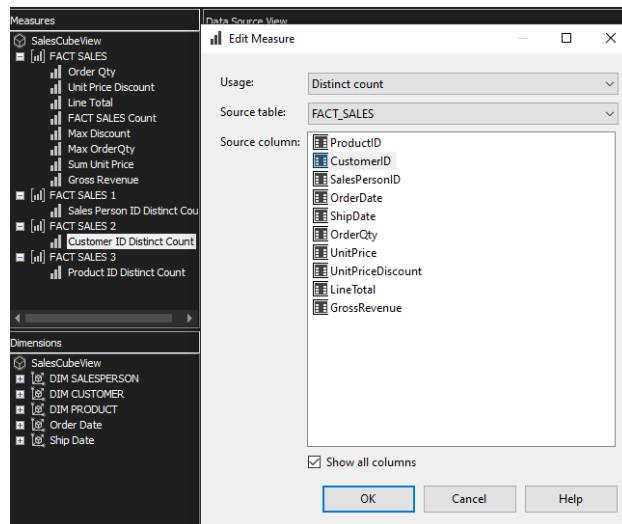
Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	DIM_TIME.PK_TIME (Integer)
NameColumn	DIM_TIME.FullDate (WChar)
ValueColumn	(none)

Rysunek 4: Widok Properties dla DIM_Time

1.3 Podpunkt c

Utworzyć nowe miary, które będą odzwierciedlać:

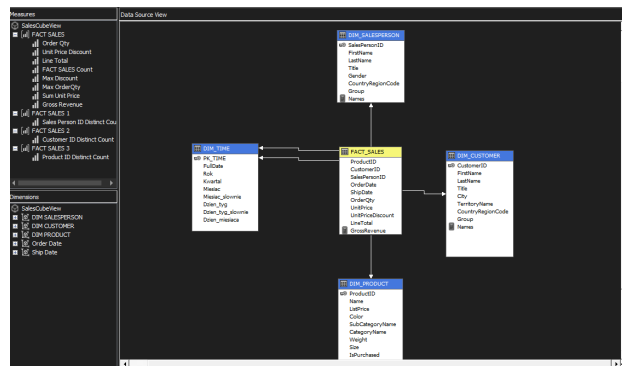
- Liczbę różnych klientów (aggregatedFunction: distinct count)
- Liczbę różnych produktów
- Maksymalną wartość rabatu (aggregatedFunction: max)
- Maksymalną liczbę zamówionych produktów
- Liczbę różnych sprzedawców realizujących zamówienia



Rysunek 5: Miara dotycząca liczby różnych klientów

1.4 Podpunkt d

Wdrożyć i przetworzyć kostkę.



Rysunek 6: Widok przetworzonej kostki

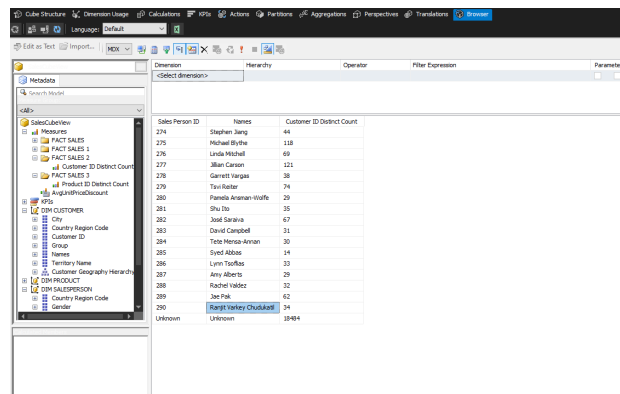
2 Zad. 2. Przegląd danych i tworzenie zestawień

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

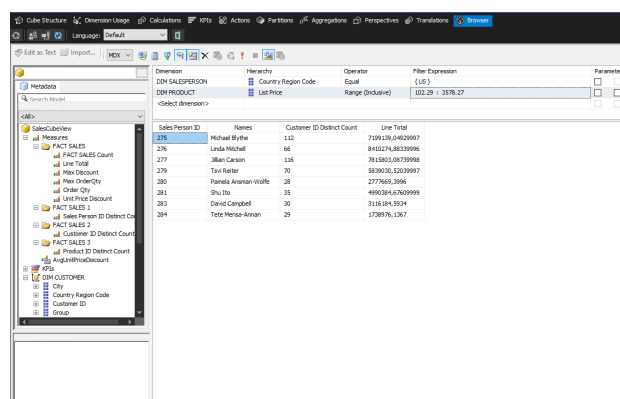


Sales Person ID	Names	Customer ID District Count
274	Stephen Jiang	44
275	Michael Blythe	118
276	Linda Mitchell	69
277	Ellen Cannon	121
278	Garrett Vargas	38
279	Toni Ratter	74
280	Pamela Annan-Violette	29
281	Shu-Ita	35
282	Joel Saravia	67
283	David Campbell	31
284	Tate Henshaw	30
285	Fred Abbas	14
286	Lynn Toffles	33
287	Amy Alberts	29
288	Rachel Valdez	32
289	Joe Pea	42
290	Target Tester Chukwudi	24
Unknown	Unknown	13484

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.



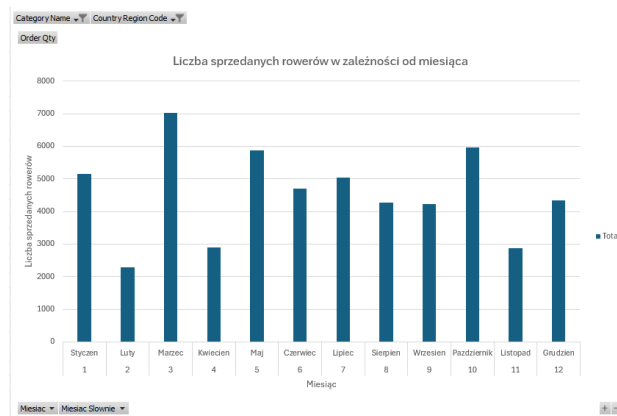
Sales Person ID	Names	Customer ID District Count	Line Total
275	Michael Blythe	112	7109129,04029997
276	Linda Mitchell	66	8402274,38339996
277	Ellen Cannon	125	7910802,08739998
279	Toni Ratter	70	5839026,52039997
280	Pamela Annan-Violette	28	2777029,3996
281	Shu-Ita	35	4990284,67600999
283	David Campbell	30	3116184,9324
284	Tate Henshaw	29	1738976,1367

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

2.3 Podpunkt c

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

2.3.1 Liczba sprzedanych rowerów w zależności od miesiąca w USA

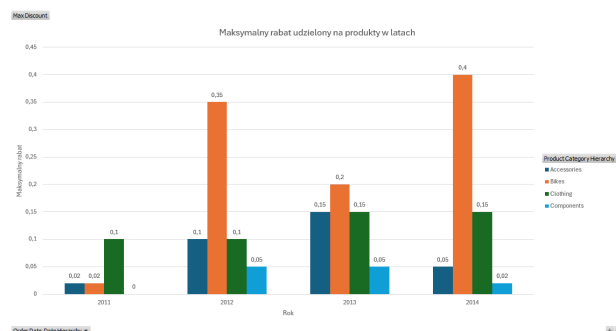


Rysunek 9: Wykres

Na wykres nałożone są 2 filtry - kategoria jest limitowana do rowerów, a kod regionu państwa do USA. Wykres jest posortowany po osi x, czyli związanej z miesiącami, agregując liczbę sprzedanych rowerów dla wszystkich lat naraz.

Generalnie widać trend wzrostu aktywności kupowania w miesiącach letnich. Mimo to, widać również wysokie wartości dla niektórych miesięcy poza sezonem - październik, grudzień, styczeń. Być może np. grudzień jest powiązany ze wzrostem sprzedaży w związku z Bożym Narodzeniem, a październik z potencjalnymi wyprzedażami po sezonie. Marzec osiąga największą sprzedaż, zaczynając większą sprzedaż sklepu do października. Najmniejsza sprzedaż jest w lutym - raczej popularniejsze są wtedy sporty zimowe.

2.3.2 Maksymalny rabat udzielony na produkty w latach od kategorii



Rysunek 10: Wykres

Na wykresie widać, że najwyższe rabaty udzielane są na rowery. Wartość maksymalnego rabatu w latach 2012-2014 była najwyższa właśnie dla rowerów. Zarówno w 2012, jak i w 2014, ta wartość była bardzo wysoka, około 2 razy wyższa, niż 3. najlepszy wynik (należący do rowerów w 2013 roku). W 2014 roku był udzielony najwyższy rabat ogółem - aż 40% zniżki dla szczęśliwego klienta. Ale w 2011, rabaty udzielane dla rowerów były niskie, podobnie jak dla innych kategorii.

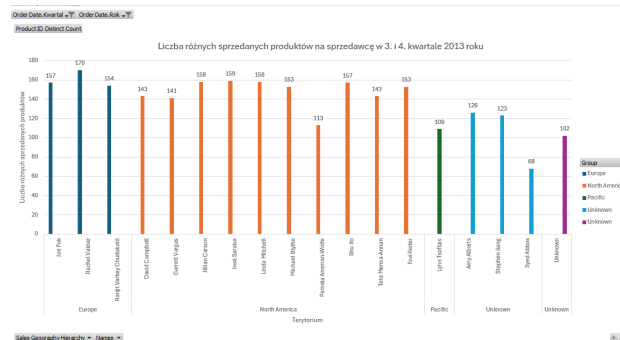
Akcesoria - wartość maksymalnego rabatu powoli rosła, z bardzo niskiego poziomu w 2011 do 4. najwyższego ogółem w 2013. Potem w 2014 znowu spadł do niskiego poziomu.

Ubrania - były najbardziej przeceniane w 2011 roku. Były przeceniane na podobnym poziomie w 2012, ale wtedy to rowery je prześcignęły. W 2013 urosły, utrzymując jeden z najwyższych poziomów również w 2014.

Komponenty - bardzo niskie maksymalne rabaty, zerowe w 2011. W 2012 i 2013 podobne, ale najniższe w zestawieniach, w 2014 znowu spadek do bardzo niskiego poziomu.

Sklep wydaje się mieć dość chaotyczną politykę udzielania rabatów. Należałoby ją lepiej uściślić. Oczywiście, każda kategoria powinna mieć osobną politykę.

2.3.3 Liczba różnych sprzedanych produktów na sprzedawcę w 3. i 4. kwartale 2013 roku



Rysunek 11: Wykres

Wykres został arbitralnie ograniczony do 3. i 4. kwartału 2013 roku - mogło być tak, że menadżer analizuje sytuację w poprzednim roku, czekając na nadejście 3. i 4. kwartału w 2014 roku w bazie danych.

Sprzedawcy utrzymują dość podobny poziom sprzedaży różnych produktów. Bardzo wielu osiąga wysoki poziom 160, zwłaszcza w Europie i w Północnej Ameryce. W rejonie pacyfiku poziom jest niższy, podobny do najniższego poziomu w poprzednich grupach - osiągniętego przez Linde Mitchell.

Aż 3 sprzedawców nie jest przypisanych do regionu - osiągają oni jedne z niższych wyników sprzedaży różnych produktów.

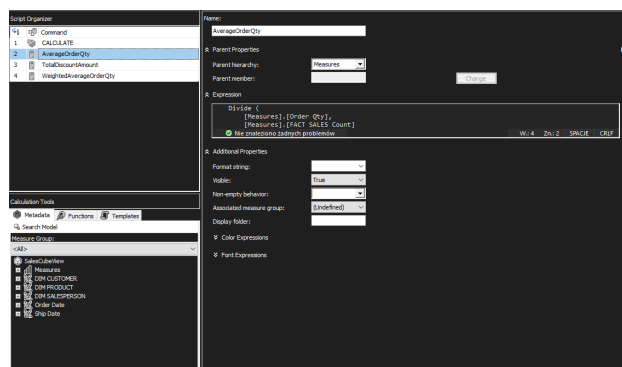
Wszystkim sprzedawcom osiągającym słabsze wyniki należy przyjrzeć się w celu ewentualnych szkoleń. Jeśli z innych analiz, np. analizy rynku, w którym pracują, ogólnej sprzedaży itd. wyszły również słabe wyniki, byłoby to kandydaci do potencjalnych redukcji. Należy również koniecznie przypisać sprzedawców bez regionów do regionów - ułatwi to analizę.

3 Zad. 3. Miary kalkulowane

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

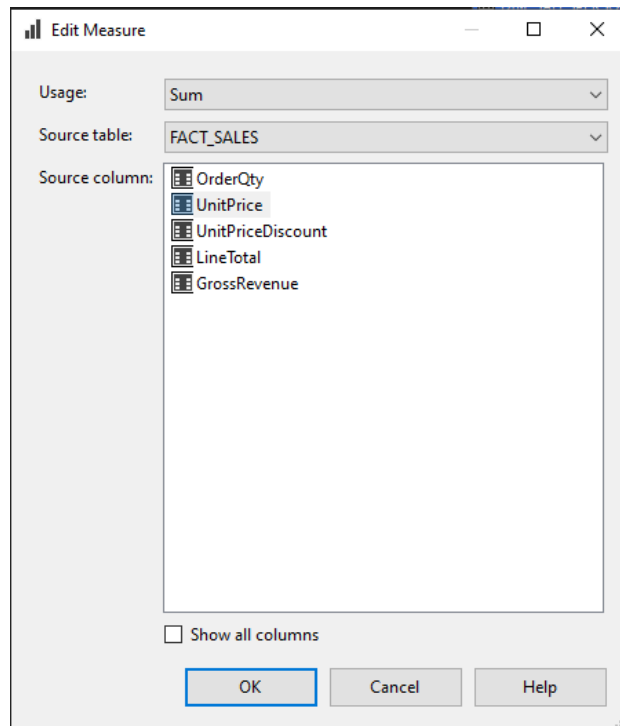
- średnią liczbę zamówionych towarów na zamówienie
- średnią ważoną liczbę towarów na zamówienie. Jako wagę należy wybrać cenę danego produktu.

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.

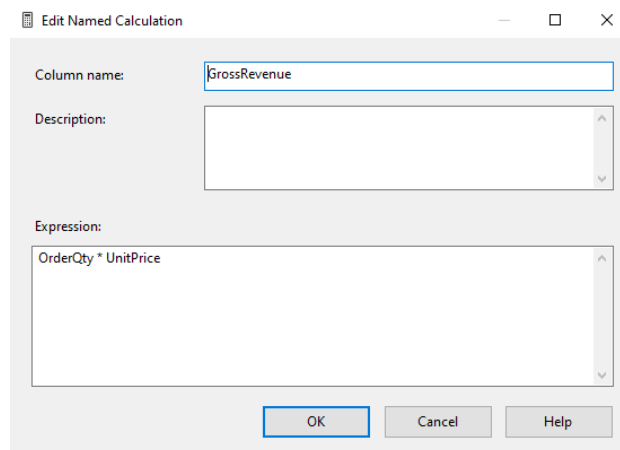


Rysunek 12: Sposób obliczania miary ze zwykłą średnią

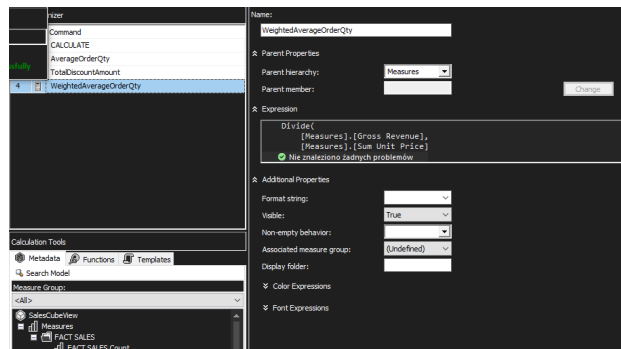
Do obliczenia średniej ważonej należało dodać miarę obliczającą sumę ceny jednostkowej i drugą miarę, będącą iloczynem ceny jednostkowej i liczby zamówionego produktu (LineTotal prawie to spełniał, ale miał w sobie czasem zniżkę).



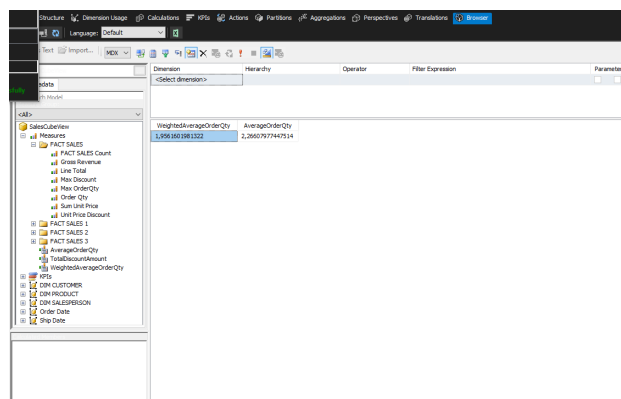
Rysunek 13: Sposób obliczania miary z sumą ceny jednostkowej



Rysunek 14: Sposób obliczania miary zysku brutto



Rysunek 15: Sposób obliczania miary średniej ważonej



Rysunek 16: Wynik średnich dla całego zbioru danych

Średnia ważona wynosi około 1.96, a zwykła średnia około 2.27. Średnia ważona jest niższa prawdopodobnie dlatego, że klienci kupują droższe produkty w statystycznie mniejszej liczbie, niż tańsze.

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,  
22     ShipDate INT NULL,  
23     OrderQty SMALLINT NOT NULL,  
24     UnitPrice MONEY NOT NULL,  
25     UnitPriceDiscount DECIMAL(8, 4) NOT NULL,  
26     LineTotal DECIMAL(19, 4) NOT NULL  
27 );  
28  
29 CREATE TABLE Kubs.FACT_SALES_2013 (  
30     ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),  
31     CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(  
32         CustomerID),  
33     SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(  
34         SalesPersonID),  
35     OrderDate INT NOT NULL,  
36     ShipDate INT NULL,  
37     OrderQty SMALLINT NOT NULL,  
38     UnitPrice MONEY NOT NULL,  
39     UnitPriceDiscount DECIMAL(8, 4) NOT NULL,  
40     LineTotal DECIMAL(19, 4) NOT NULL  
41 );
```

```

SalesPersonID),
29   OrderDate INT NOT NULL,
30   ShipDate INT NULL,
31   OrderQty SMALLINT NOT NULL,
32   UnitPrice MONEY NOT NULL,
33   UnitPriceDiscount DECIMAL(8, 4) NOT NULL,
34   LineTotal DECIMAL(19, 4) NOT NULL
35 );
36
37 CREATE TABLE Kubs.FACT_SALES_2014 (
38   ProductID INT FOREIGN KEY REFERENCES Kubs.DIM_PRODUCT(ProductID),
39   CustomerID INT FOREIGN KEY REFERENCES Kubs.DIM_CUSTOMER(
CustomerID),
40   SalesPersonID INT FOREIGN KEY REFERENCES Kubs.DIM_SALESPERSON(
SalesPersonID),
41   OrderDate INT NOT NULL,
42   ShipDate INT NULL,
43   OrderQty SMALLINT NOT NULL,
44   UnitPrice MONEY NOT NULL,
45   UnitPriceDiscount DECIMAL(8, 4) NOT NULL,
46   LineTotal DECIMAL(19, 4) NOT NULL
47 );
48
49 with Sales1 AS (
50   SELECT
51     Kubs.FACT_SALES.ProductID,
52     Kubs.FACT_SALES.CustomerID,
53     Kubs.FACT_SALES.SalesPersonID,
54     Kubs.FACT_SALES.OrderDate,
55     Kubs.FACT_SALES.ShipDate,
56     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 with Sales2 AS (
66   SELECT
67     Kubs.FACT_SALES.ProductID,
68     Kubs.FACT_SALES.CustomerID,

```

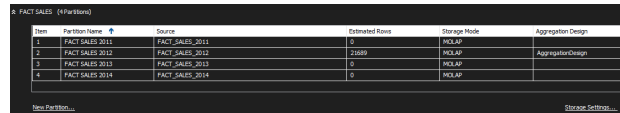
```

69     Kubs.FACT_SALES.SalesPersonID,
70     Kubs.FACT_SALES.OrderDate,
71     Kubs.FACT_SALES.ShipDate,
72     Kubs.FACT_SALES.OrderQty,
73     Kubs.FACT_SALES.UnitPrice,
74     Kubs.FACT_SALES.UnitPriceDiscount,
75     Kubs.FACT_SALES.LineTotal
76 FROM Kubs.FACT_SALES
77 WHERE OrderDate >= 20120101 AND OrderDate < 20130000
78 )
79 INSERT INTO Kubs.FACT_SALES_2012
80
81 with Sales3 AS (
82     SELECT
83         Kubs.FACT_SALES.ProductID,
84         Kubs.FACT_SALES.CustomerID,
85         Kubs.FACT_SALES.SalesPersonID,
86         Kubs.FACT_SALES.OrderDate,
87         Kubs.FACT_SALES.ShipDate,
88         Kubs.FACT_SALES.OrderQty,
89         Kubs.FACT_SALES.UnitPrice,
90         Kubs.FACT_SALES.UnitPriceDiscount,
91         Kubs.FACT_SALES.LineTotal
92 FROM Kubs.FACT_SALES
93 WHERE OrderDate >= 20130101 AND OrderDate < 20140000
94 )
95 INSERT INTO Kubs.FACT_SALES_2013
96
97 with Sales4 AS (
98     SELECT
99         Kubs.FACT_SALES.ProductID,
100        Kubs.FACT_SALES.CustomerID,
101        Kubs.FACT_SALES.SalesPersonID,
102        Kubs.FACT_SALES.OrderDate,
103        Kubs.FACT_SALES.ShipDate,
104        Kubs.FACT_SALES.OrderQty,
105        Kubs.FACT_SALES.UnitPrice,
106        Kubs.FACT_SALES.UnitPriceDiscount,
107        Kubs.FACT_SALES.LineTotal
108 FROM Kubs.FACT_SALES
109 WHERE OrderDate >= 20140101
110 )
111 INSERT INTO Kubs.FACT_SALES_2014

```

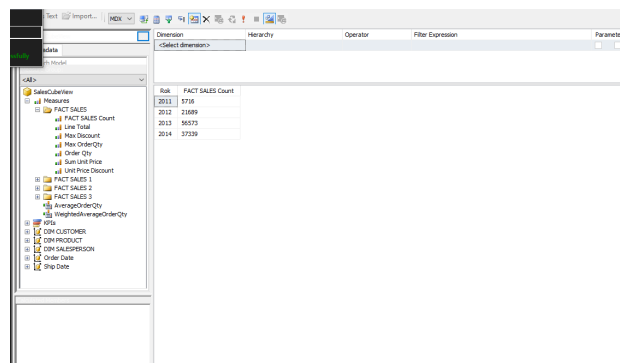
Listing 1: Tworzenie i wypełnianie tabeli DIM_TIME.

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	AggregationDesign
3	FACT_SALES_2013	FACT_SALES_2013	0	HOLAP	
4	FACT_SALES_2014	FACT_SALES_2014	0	HOLAP	

Rysunek 17: Dodane partycje



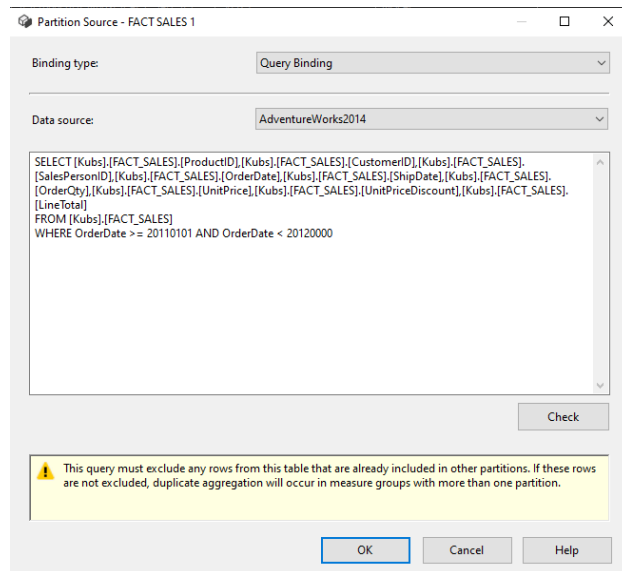
Year	FACT_SALES Count
2011	1718
2012	2569
2013	5575
2014	5739

Rysunek 18: 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 19: Zmieniony kod SQL w partycji - dodana klauzula WHERE ograniczająca daty

Name	Source	Estimated Rows	Storage Mode	Aggregation Design
FACT SALES 1	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20111010 AND OrderDate < 20120000	1716	HOLAP	
FACT SALES 2	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20120000 AND OrderDate < 20130000	2169	HOLAP	
FACT SALES 3	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20130000 AND OrderDate < 20140000	9573	HOLAP	
FACT SALES 4	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20140000 AND OrderDate < 20150000	2739	HOLAP	

New Partition...

Storage Settings...

FACT SALES 1 (1 Partition)

FACT SALES 2 (1 Partition)

FACT SALES 3 (1 Partition)

Rysunek 20: Dodane partycje

Name	Source	Estimated Rows	Storage Mode	Aggregation Design
FACT SALES 1	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20111010 AND OrderDate < 20120000	1716	HOLAP	
FACT SALES 2	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20120000 AND OrderDate < 20130000	2169	HOLAP	
FACT SALES 3	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20130000 AND OrderDate < 20140000	9573	HOLAP	
FACT SALES 4	SELECT [Kubs].[FACT_SALES].[ProductID],[Kubs].[FACT_SALES].[CustomerID],[Kubs].[FACT_SALES].[SalesPersonID],[Kubs].[FACT_SALES].[OrderDate],[Kubs].[FACT_SALES].[ShipDate],[Kubs].[FACT_SALES].[OrderQty],[Kubs].[FACT_SALES].[UnitPrice],[Kubs].[FACT_SALES].[UnitPriceDiscount],[Kubs].[FACT_SALES].[LineTotal] FROM [Kubs].[FACT_SALES] WHERE OrderDate >= 20140000 AND OrderDate < 20150000	2739	HOLAP	

New Partition...

Storage Settings...

FACT SALES 1 (1 Partition)

FACT SALES 2 (1 Partition)

FACT SALES 3 (1 Partition)

Rysunek 21: 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:

`iif([Measures].[OrderQty] < η , -1 /*czerwony*/, 1 /*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;
```

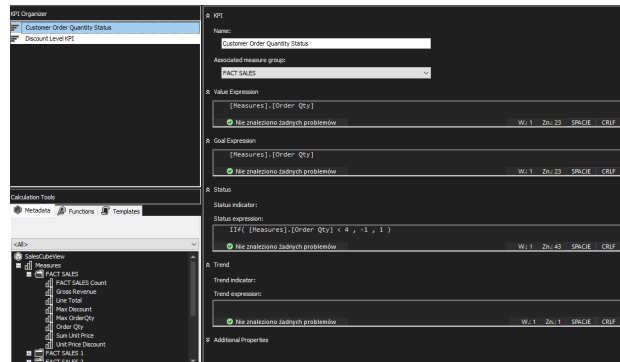
Listing 2: Kwerenda znajdująca dane do wyliczenia KPI

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.

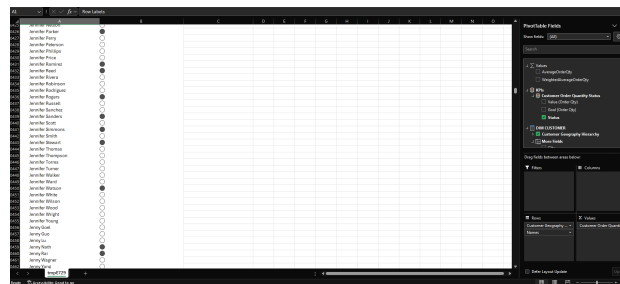
Wybór wartości granicznej równej 20% wynika z konceptu „*krzywej wieloryba*” (ang. *whale curve*), gdzie:

- 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 22: Sposób obliczenia KPI



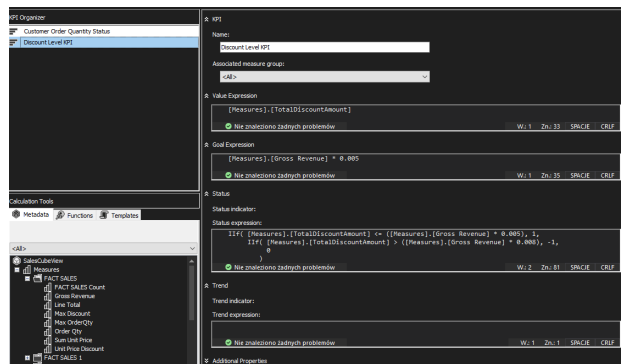
Rysunek 23: Tabela przestawna z widokiem na wartości KPI

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: $\leq .5\%$, źle: $> .8\%$) oraz pokazuje trend porównując wartość rabatu do roku poprzedniego.



Rysunek 24: 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 25: Tabela przestawna z widokiem na wartości 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 z trendami

6 Wnioski

Realizacja laboratorium pozwoliła na pogłębienie wiedzy na temat zaawansowanych funkcji SSAS.

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