

Architektura Hurtowni Danych

Model wielowymiarowy (Multidimensional)

Model tabularyczny (Tabular)

Rafał Miodowski

1. Wstęp do projektu:

Celem niniejszego projektu jest praktyczne zaprojektowanie i uruchomienie dwóch modeli hurtowni danych: tabelarnego (Tabular) oraz wielowymiarowego (Multidimensional) z wykorzystaniem środowiska Microsoft SQL Server Analysis Services (SSAS). Projekt ma na celu pokazanie różnic w architekturze, sposobie przetwarzania danych, wydajności i możliwościach analitycznych obu podejść w kontekście przetwarzania danych biznesowych.

W ramach ćwiczenia wykonane zostaną następujące elementy:

- utworzenie dwóch niezależnych hurtowni danych (dla obu modeli),
- zaprojektowanie miar, KPI, perspektyw oraz wymiarów,
- opracowanie zapytań analitycznych przy użyciu DAX (dla modelu tabelarnego) i MDX (dla modelu wielowymiarowego),
- wykorzystanie funkcji Time Intelligence w analizach porównujących dane między okresami (np. miesiące, lata 2015–2016).

2. Opis środowiska i infrastruktury technicznej:

Projekt realizowany jest w środowisku wirtualnym (Hyper-V) w oparciu o system Windows Server2022.

W środowisku zostały uruchomione dwie maszyny wirtualne:

1. VM-DC (Domain Controller)

- pełni rolę kontrolera domeny w lokalnym środowisku sieciowym,
- odpowiada za autoryzację użytkowników, zarządzanie kontami i politykami domenowymi,
- zapewnia spójne środowisko logowania i bezpieczeństwa dla pozostałych maszyn.

2. M-SQL (SQL Server)

- zainstalowany system Windows Server 2022,
- zawiera środowiska:
 - Microsoft SQL Server 2022 (Database Engine) – główna instancja bazy danych,
 - SQL Server Analysis Services (Tabular Mode) instancja do tworzenia modelu tabelarycznego
 - SQL Server Analysis Services (Multidimensional Mode) instancja do tworzenia modelu wielowymiarowego
 - SQL Server Management Studio (SSMS) – zarządzanie bazami i serwerami,
 - Visual Studio (z rozszerzeniem Analysis Services Projects) – projektowanie modeli i struktur hurtowni

Obie maszyny zostały połączone w jedną domenę, dzięki czemu możliwa jest komunikacja między serwerem SQL a kontrolerem domeny przy użyciu kont domenowych. Takie rozwiązanie odzwierciedla rzeczywiste środowisko korporacyjne, w którym serwer analityczny funkcjonuje w ramach kontrolowanej infrastruktury domenowej.

3. Cel środowiska i konfiguracji:

Zastosowanie dwóch odrębnych maszyn ma na celu:

- odseparowanie ról (zarządzanie domeną vs. przetwarzanie danych),
- zwiększenie bezpieczeństwa i kontroli dostępu,
- wierne odzwierciedlenie architektury korporacyjnej wykorzystywanej w działach Business Intelligence,
- umożliwienie testowania autoryzacji SSAS w środowisku domenowym (np. użytkownicy domenowi z różnymi uprawnieniami do modeli).

4. Zakres merytoryczny projektu:

W projekcie zostaną opracowane dwie hurtownie danych:

- Model tabelaryczny – oparty o silnik kolumnowy, analizowany przy użyciu języka DAX,
- Model wielowymiarowy – oparty o klasyczny silnik MOLAP, analizowany przy użyciu MDX.

Każda z hurtowni będzie zawierała:

- minimum 3 miary (jedna z wykorzystaniem funkcji CALCULATE),
- 1 KPI oceniający wynik względem celu,
- co najmniej 2 perspektywy wraz z uzasadnieniem ich użycia,
- tłumaczenia (translations) dla wybranych wymiarów (język angielski i polski).

W ramach części analitycznej zostaną przygotowane zapytania porównujące dane z sąsiadujących miesięcy i lat (2015–2016) oraz przykłady wykorzystania funkcji Time Intelligence do analizy dynamiki sprzedaży.

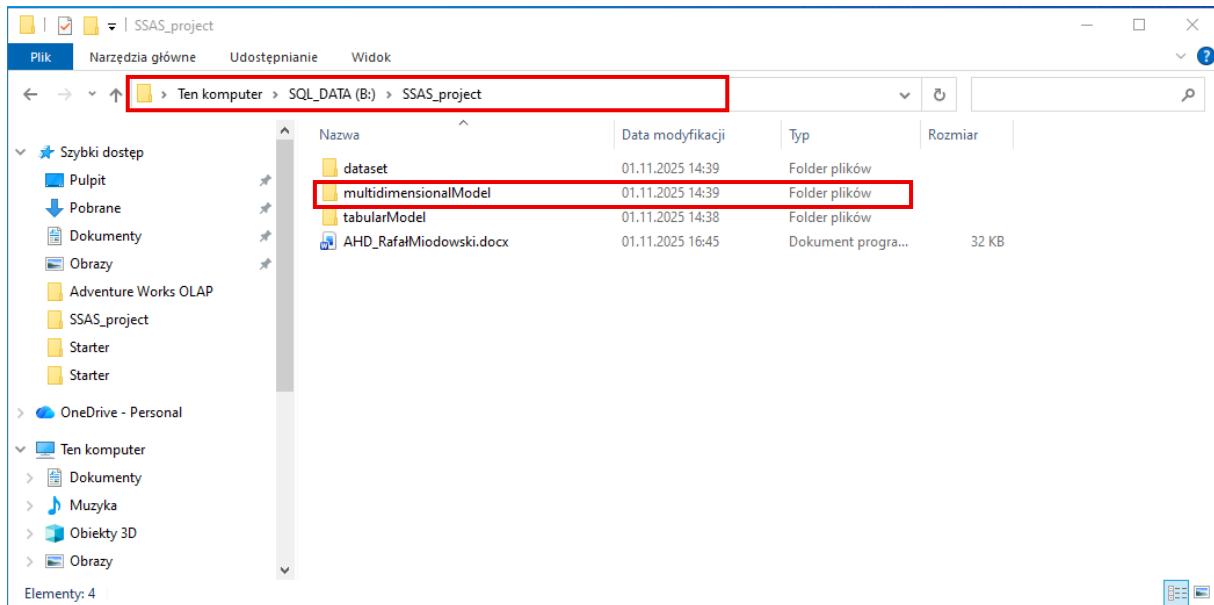
5. Rezultat końcowy:

Efektem końcowym projektu będą:

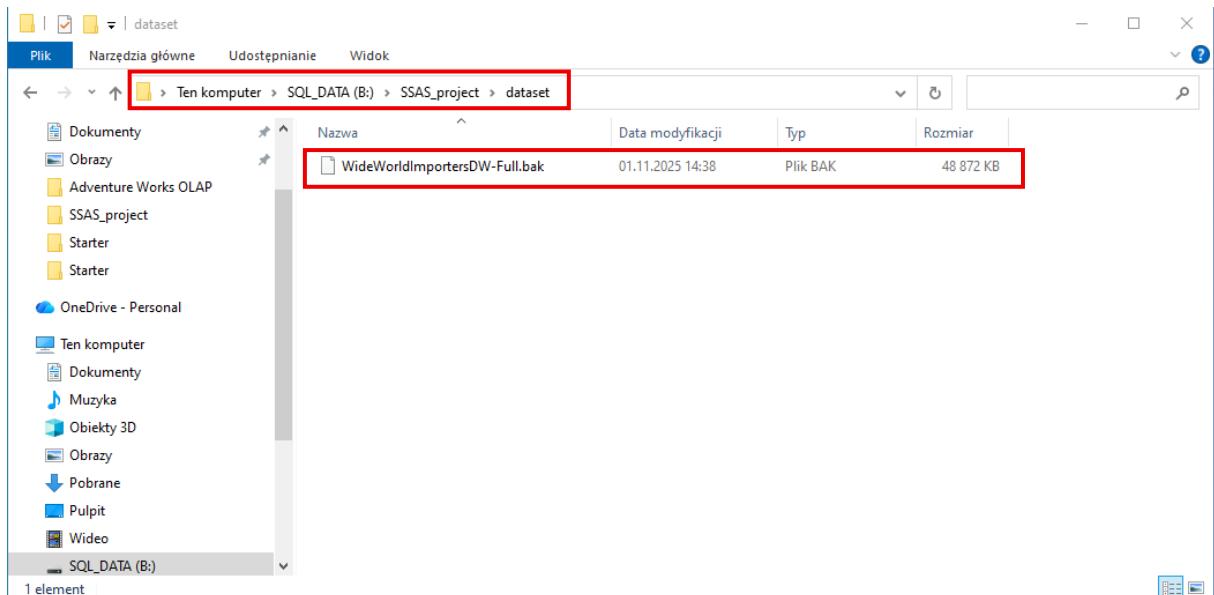
- dwa kompletne modele hurtowni danych (Tabular i Multidimensional),
- sprawozdanie dokumentujące cały proces konfiguracyjny, analityczny i wizualny,
- zrzuty ekranów z VS, SSMS i Excel (Analyze in Excel) przedstawiające działające modele i zapytania,

6. Model wielowymiarowy (Multidimensional):

- Struktura katalogów projektu:

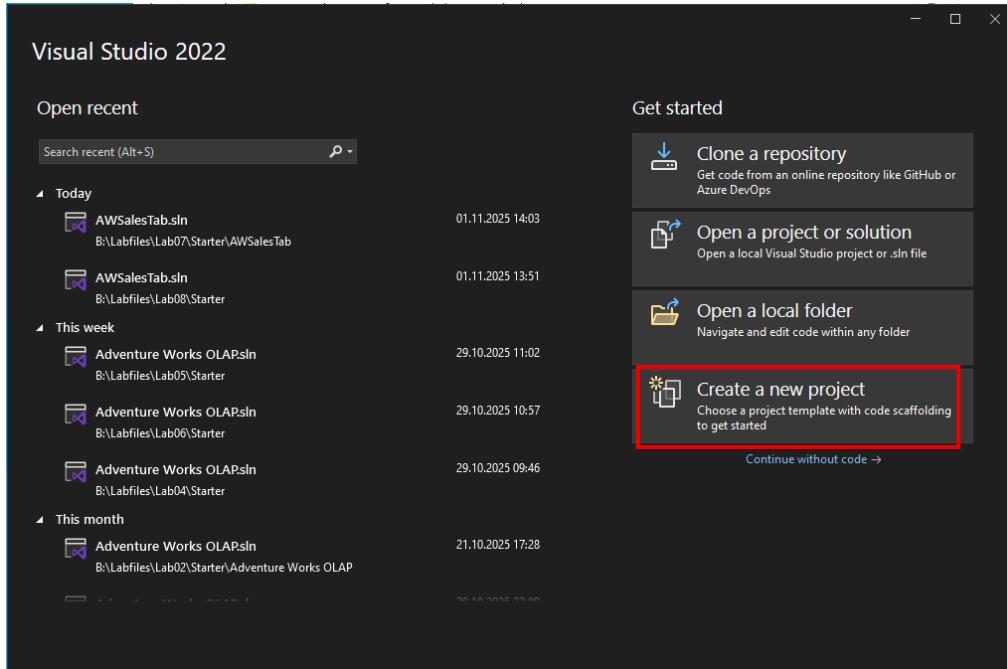


Wszystkie katalogi znajdują się na dysku B, w którym znajduje się katalog „multidimensionalModel” w którym będzie przechowywany projekt dotyczący modelu wielowymiarowego.

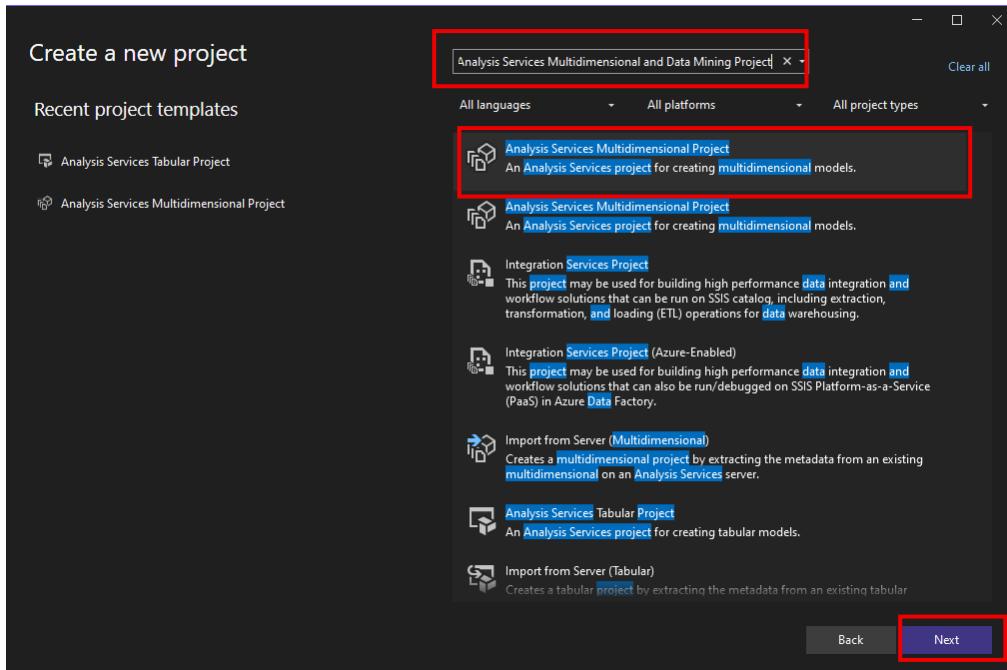


W podkatalogu „dataset” znajduje się backup bazy danych WWI, której będziemy używać.

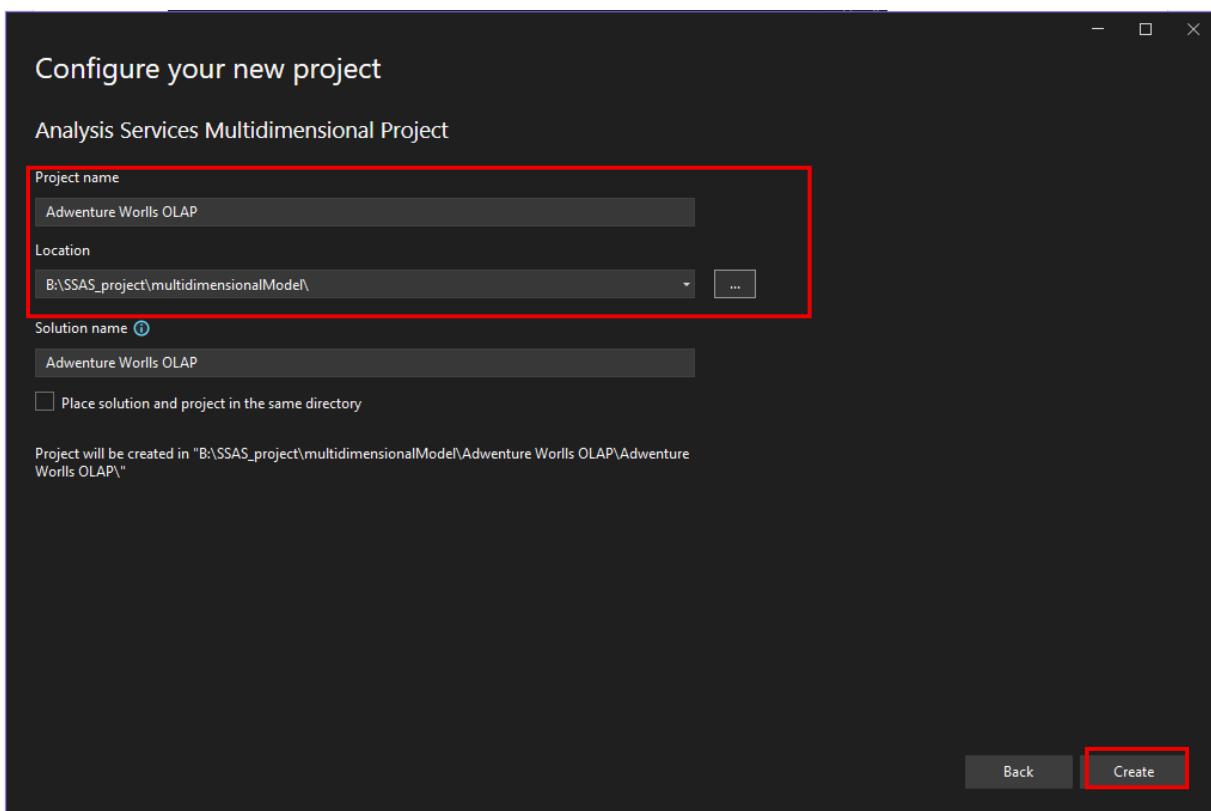
- Utworzenie projektu w Visual Studio:



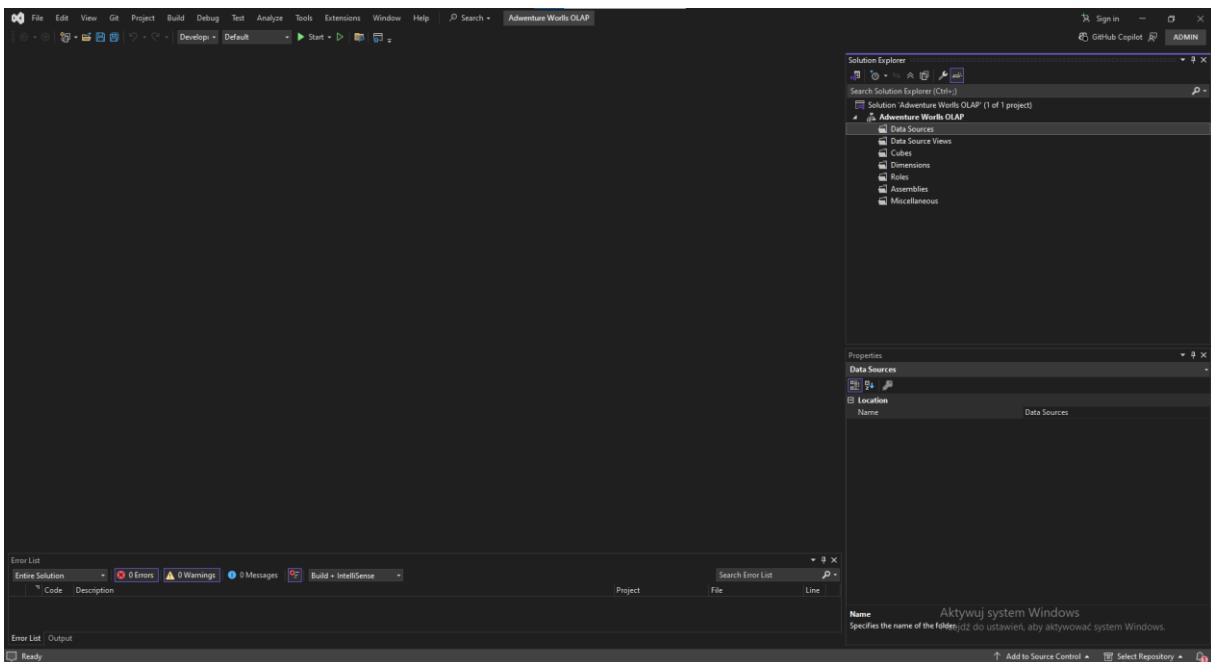
Włączamy Visual Studio i klikamy ikonę utworzenia nowego projektu



W oknie wyszukiwania wpisujemy widoczną na zrzucie ekranu frazę oraz wybieramy projekt dotyczący modelu wielowymiarowego (Multidimensional)

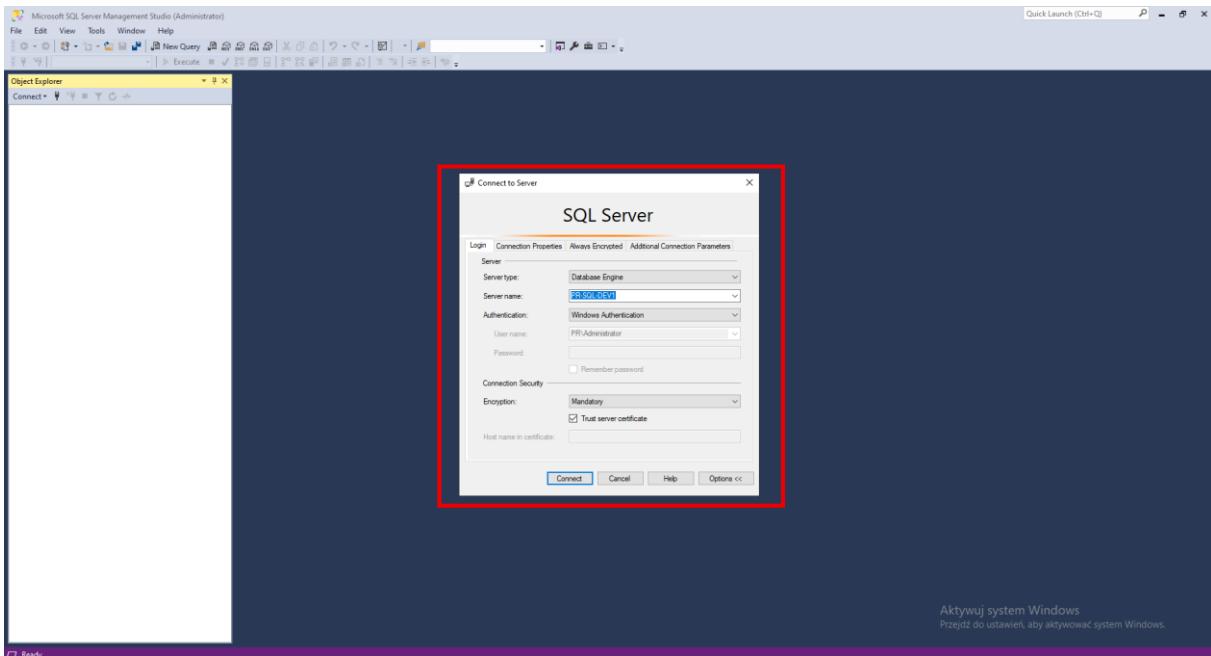


Należy wpisać nazwę projektu oraz wybrać odpowiednią lokalizację

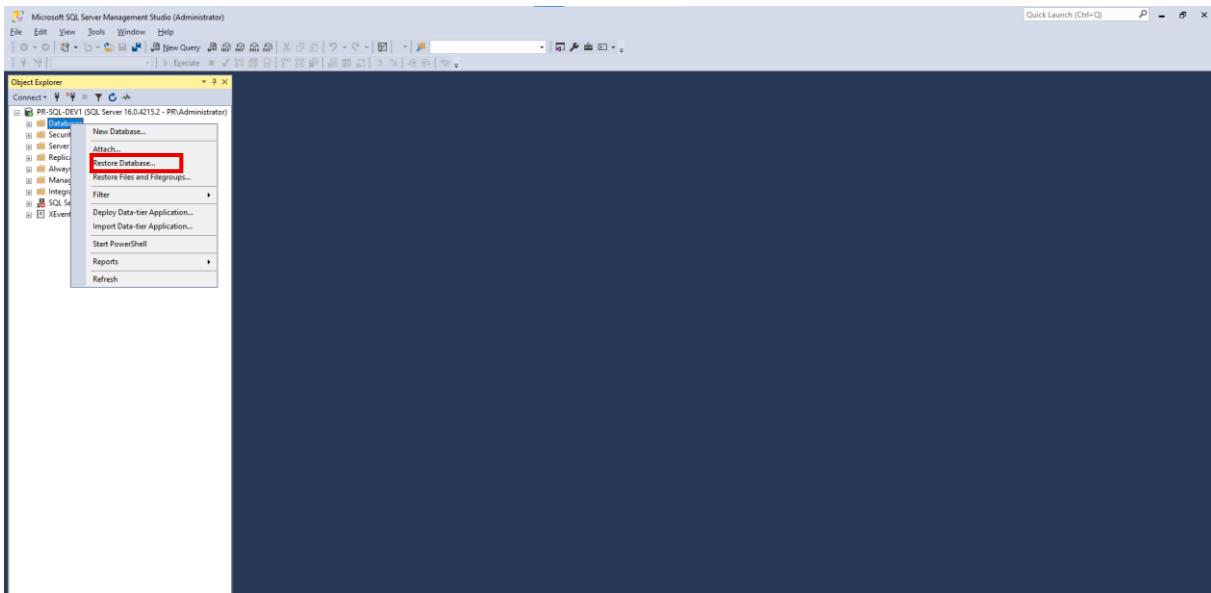


Przygotowany projekt Analysis Services

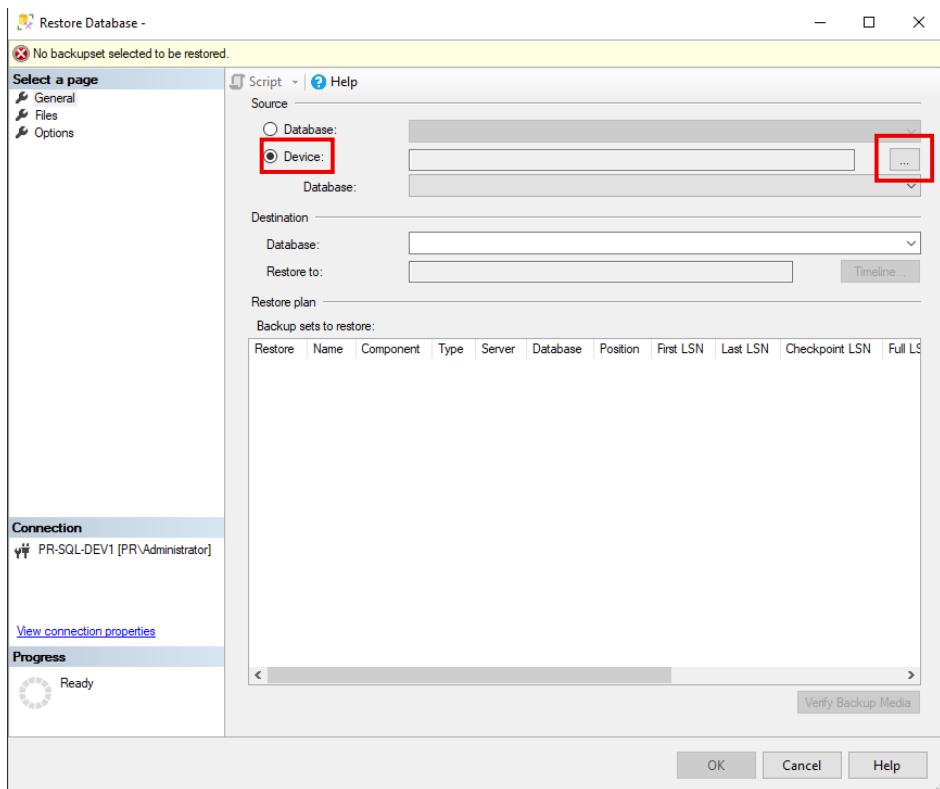
- Dołączenie bazy danych WWI do SSMS:



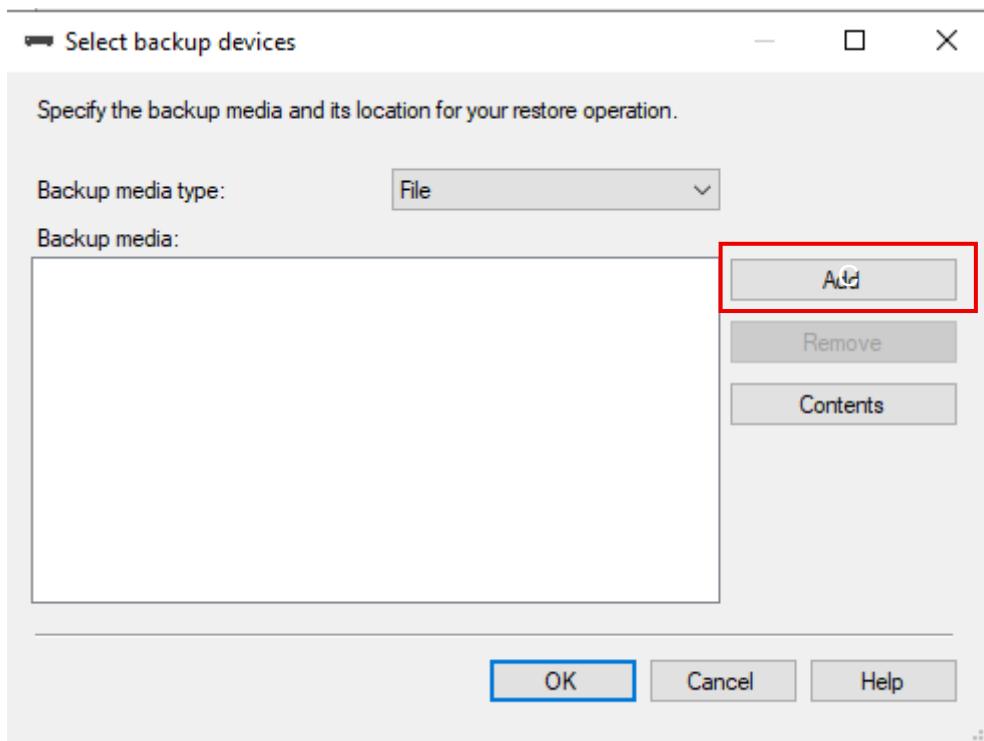
Włączamy SSMS oraz łączymy się z naszą główną instancją PR-SQL-DEV1, ta instancja jest naszym głównym silnikiem (Database Engine)



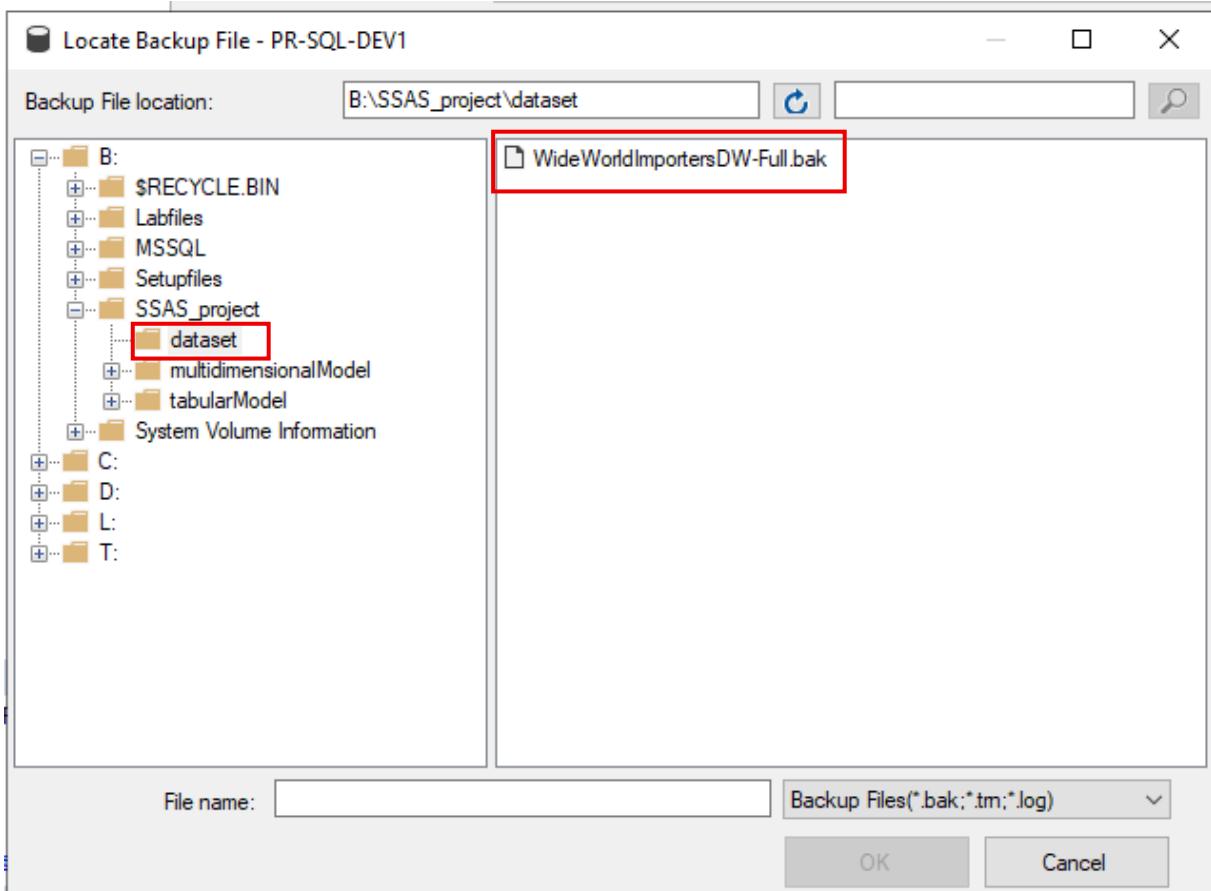
Pravym przyciskiem myszy klikamy na katalog „Databases” a następnie klikamy w „Restore Database...” – w tym momencie wykorzystamy podpięcie naszej bazy danych WWI z pliku .bak z folderu „dataset”



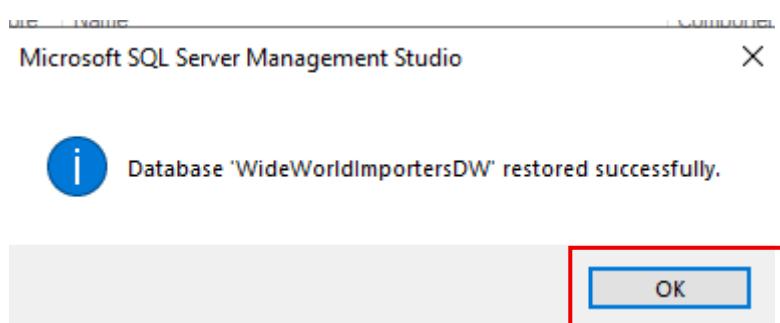
W oknie Device klikamy ... w celu dodania naszej bazy



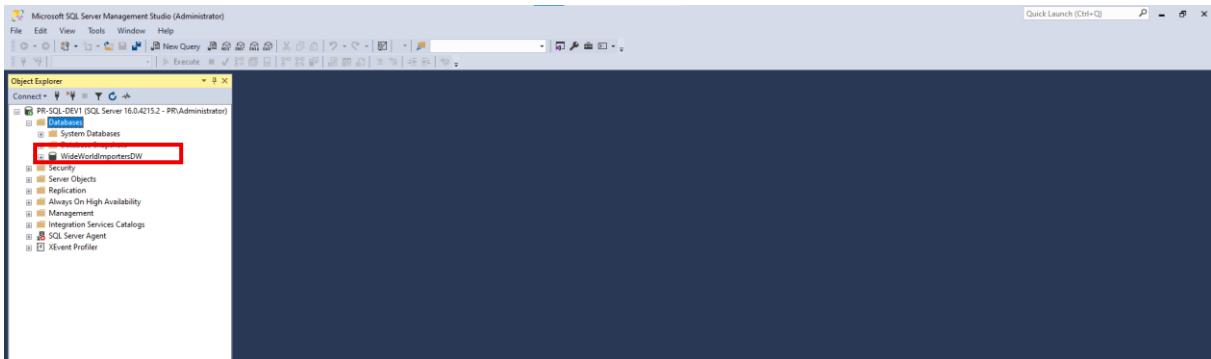
Następnie klikamy przycisk „Add”



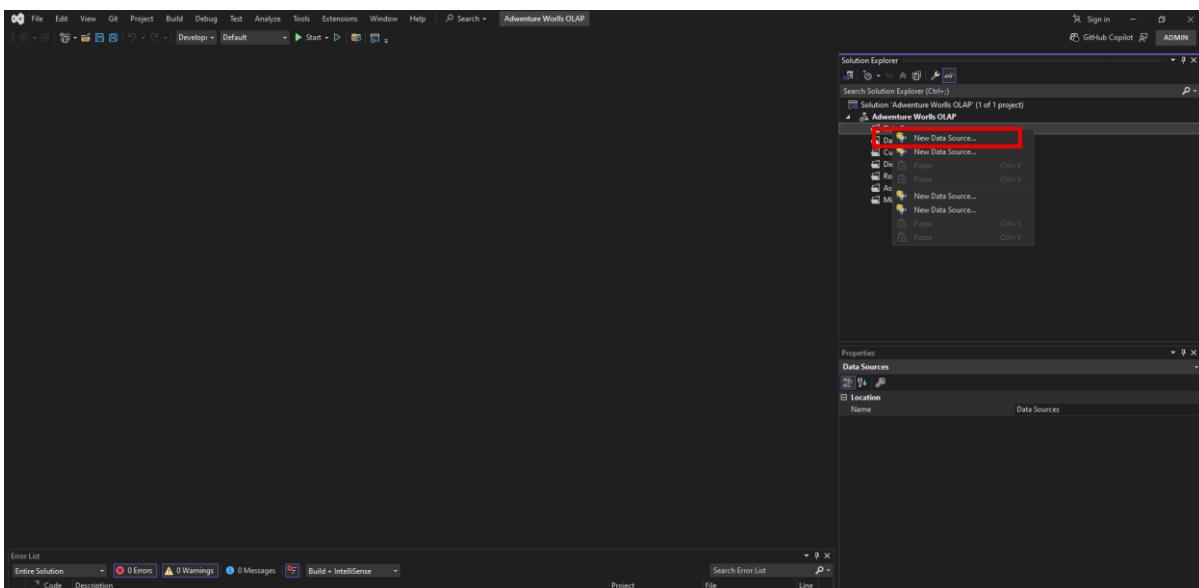
Ukazuje się nam drzewo struktury katalogów, wyszukujemy nasz folder o nazwie „dataset” i mam do wyboru backup naszej bazy – klikamy go a następnie OK



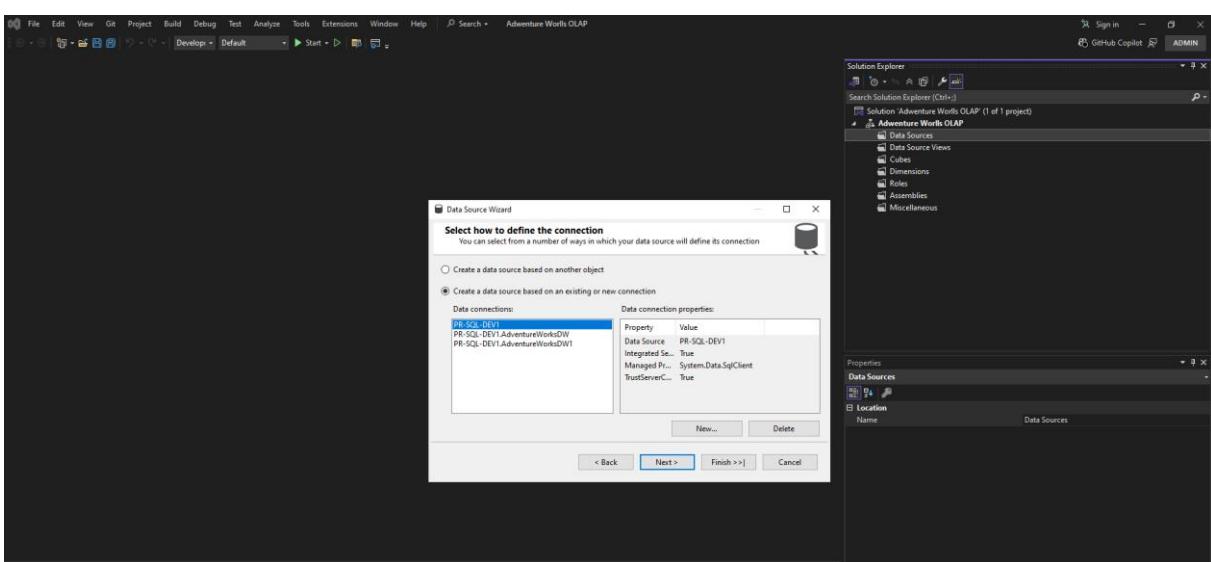
Baza danych została poprawnie załadowana



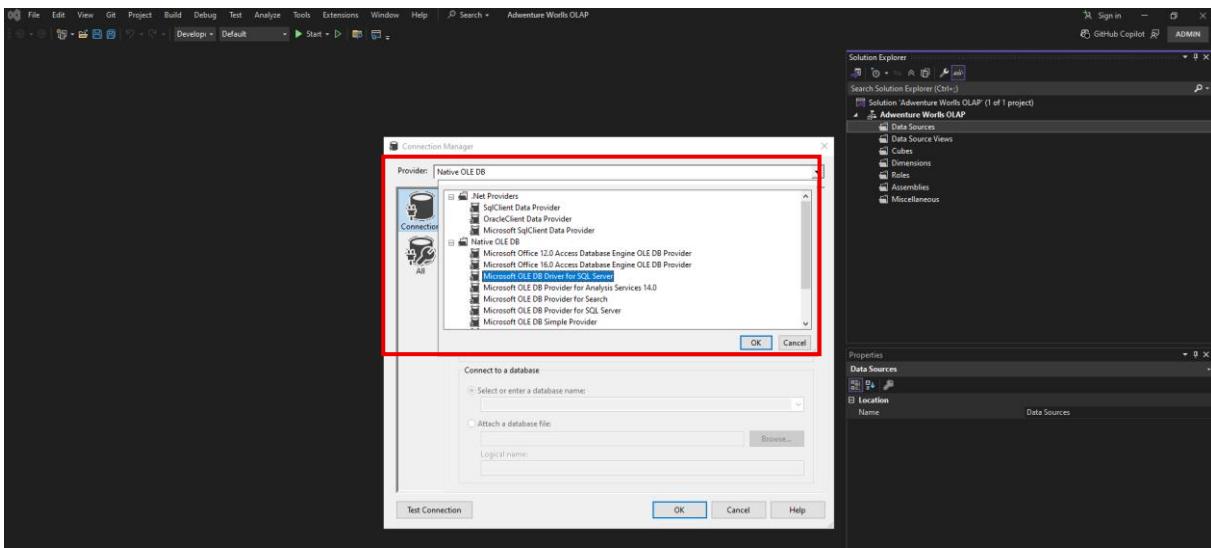
- Połączenie z bazą danych (źródłem danych (Data source)) z pozycji Visual Studio:



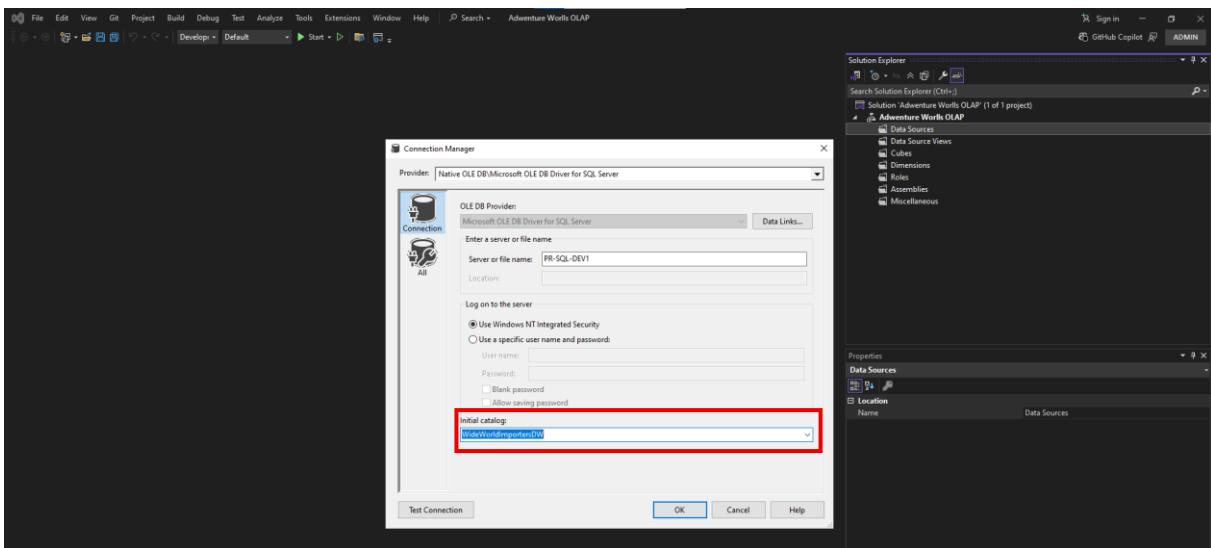
W celu połączenia się ze źródłem danych PPM klikamy na Data source a następnie New Data Source..



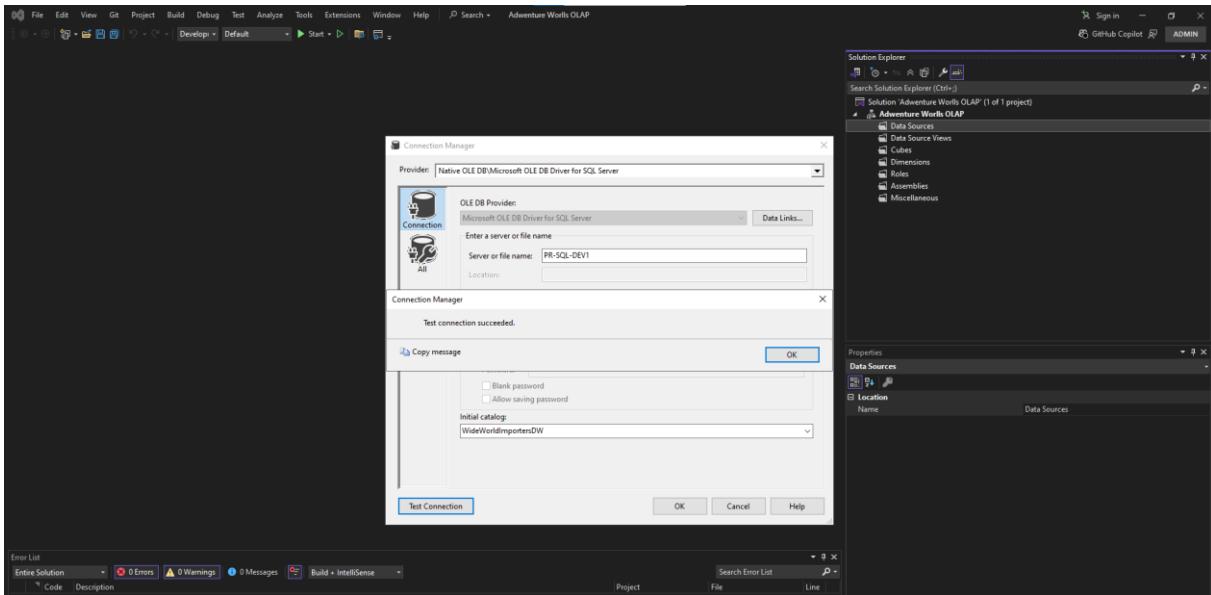
Dalsze kroki wykonujemy zgodnie z załączonym obrazkiem



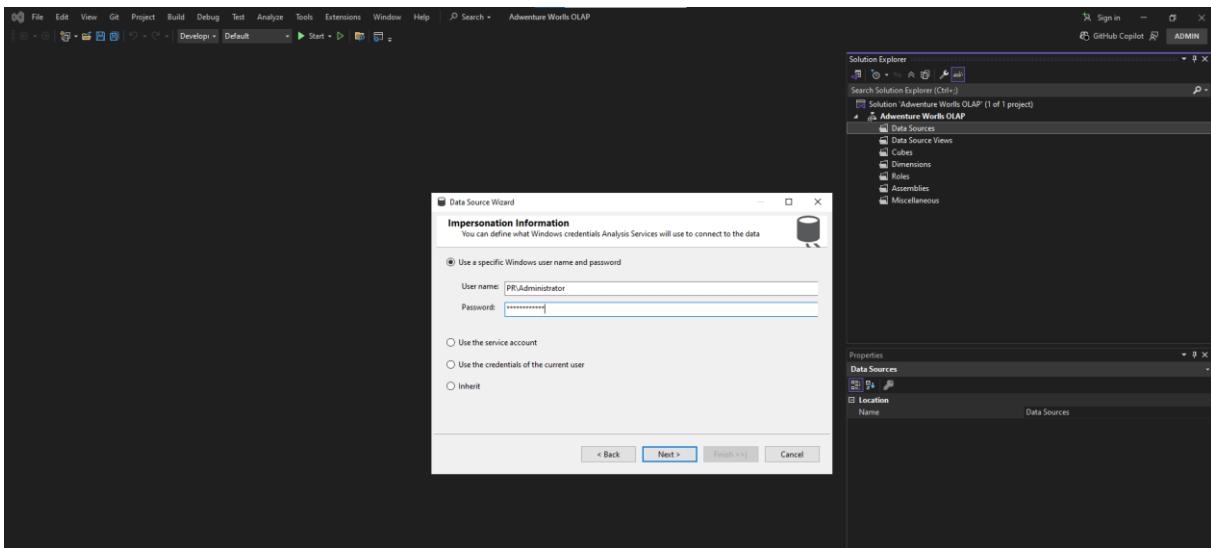
Wybieramy odpowiedniego dostawcę (Provider) obsługujący naszą bazę danych – jest to bardzo ważne, ponieważ jeśli wybierzemy złego providera to nie będziemy mogli się połączyć



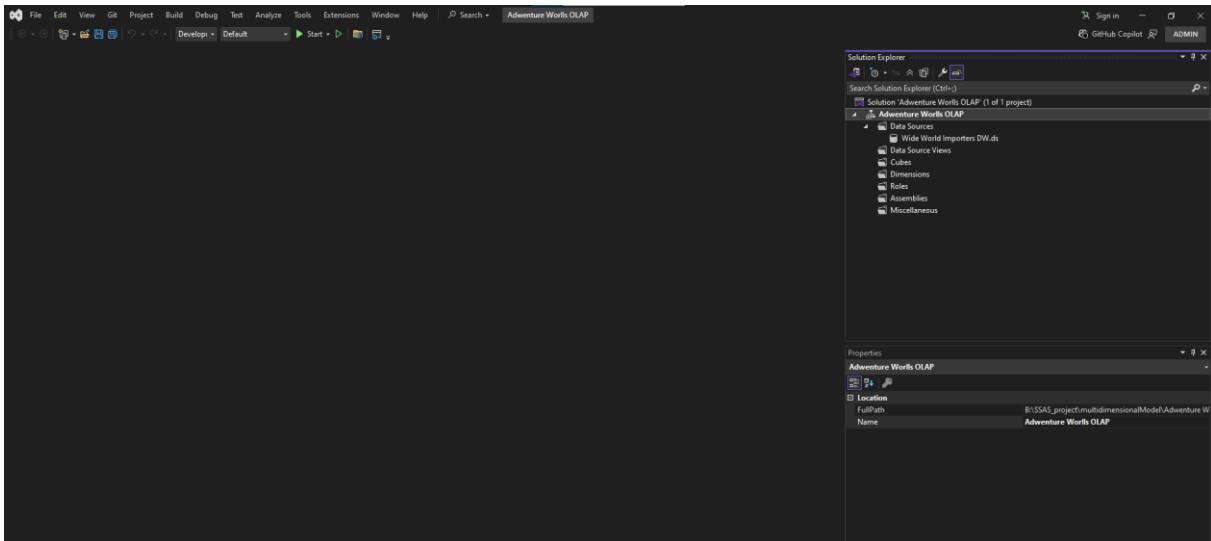
Wpisujemy nazwę naszej głównej instancji oraz nazwę bazy danych z którą chcemy się połączyć (po rozwinięciu listy powinna się pojawić)



W celu sprawdzenia poprawności klikamy Test Connection, jeśli wszystko jest w porządku to wyświetli się nam komunikat ze wszystko przebiegło poprawnie



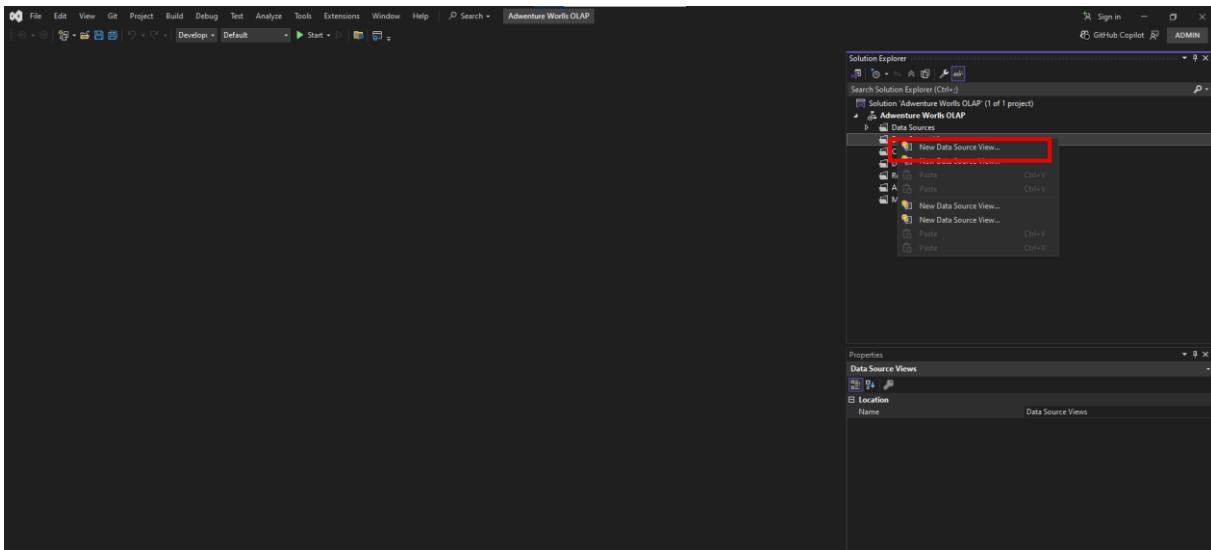
Logujemy się w celu weryfikacji, w tym przypadku zalogowałem się z konta domenowego – na potrzeby projektu tyle wystarczy. W środowiskach korporacyjnych możliwe, logowalibyśmy się z konta dedykowanego, daje to większą kontrolę nad bezpieczeństwem.



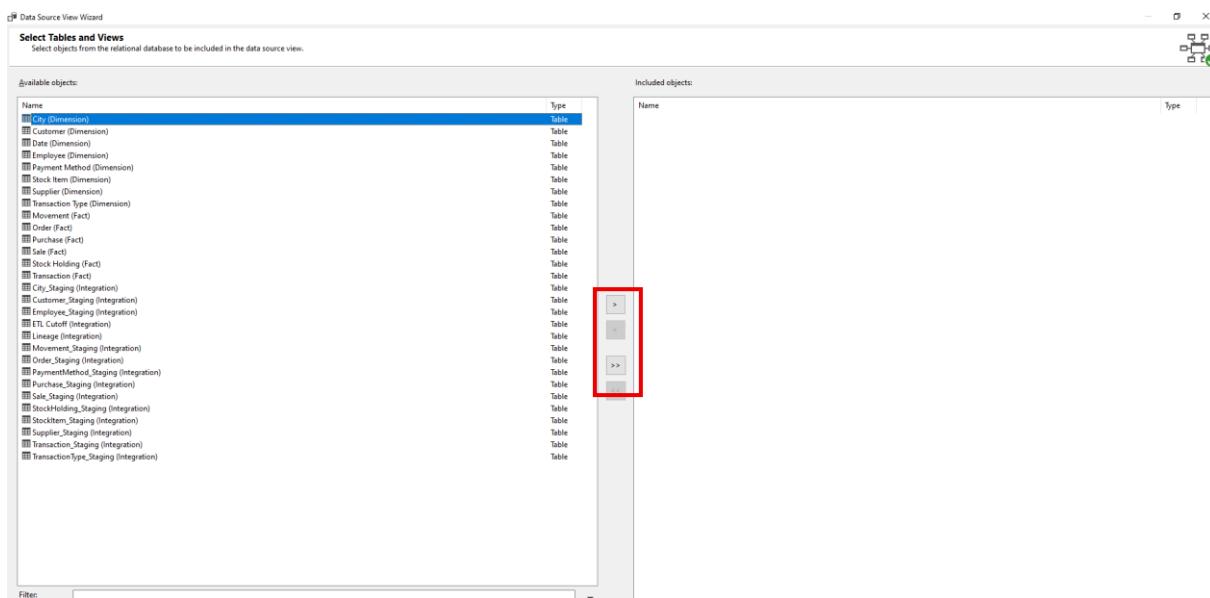
Widzimy, że wszystko przebiegło poprawnie i w Data Sources mamy widoczną naszą bazę danych

- Utworzenie Data Source View (DSV):

Ważny punkt projektu, ponieważ pomoże to nam uprościć i zoptymalizować dostęp do danych. Pozwoli to nam również na zmianę nazw kolumn, łączenie danych oraz zarządzanie strukturą niezależnie od oryginalnych źródeł. Finalnie dostarczy to nam gotową bazę dla kostek (cube) i wymiarów (dimensions).



Klikamy PPM na Data Source View potem na stronie powitalnej Next, upewniamy się, że mamy zaznaczone nasze WWI i następnie ponownie klikamy Next



Kolejny kluczowy moment projektu, ponieważ decydujemy jakie tabele będą nam potrzebne

Zakładam, że moja hurtownia będzie analizować sprzedaż w pięciu kluczowych wymiarach biznesowych, dzięki temu projekt będzie wspierać analizę wyników firmy w kontekście różnych aspektów działalności sprzedażowej.

Zakres analizy będzie obejmował odpowiedzi na pytania:

Kto kupuje? – analiza klientów (Customer Dimension), np. według grup zakupowych lub lokalizacji.

Co kupuje? – analiza sprzedawanych produktów (Stock Item Dimension), np. według kategorii lub marki.

Gdzie kupuje? – analiza geograficzna (City Dimension), np. sprzedaż w poszczególnych miastach, regionach i krajach.

Kiedy kupuje? – analiza czasowa (Date Dimension), np. według dni, miesięcy, kwartałów i lat, z wykorzystaniem funkcji Time Intelligence.

Kto sprzedaje? – analiza efektywności pracowników (Employee Dimension), np. sprzedaż według sprzedawców lub działów.

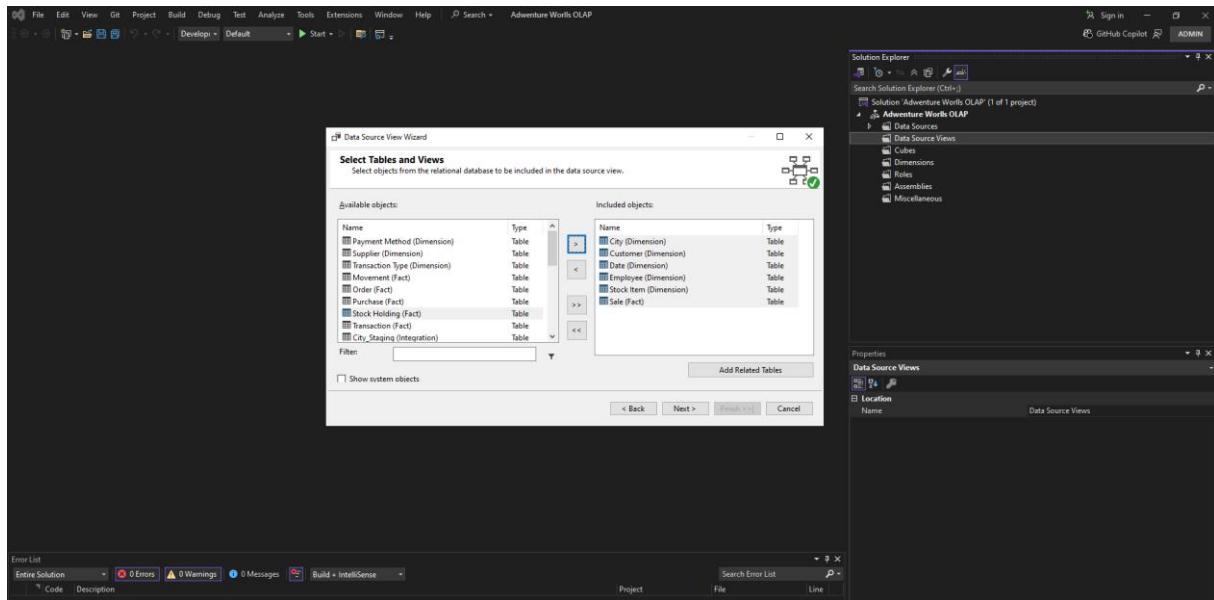
Dodatkowo zostaną opracowane:

KPI – wskaźnik realizacji planu sprzedaży względem celu (np. +10% względem poprzedniego roku),

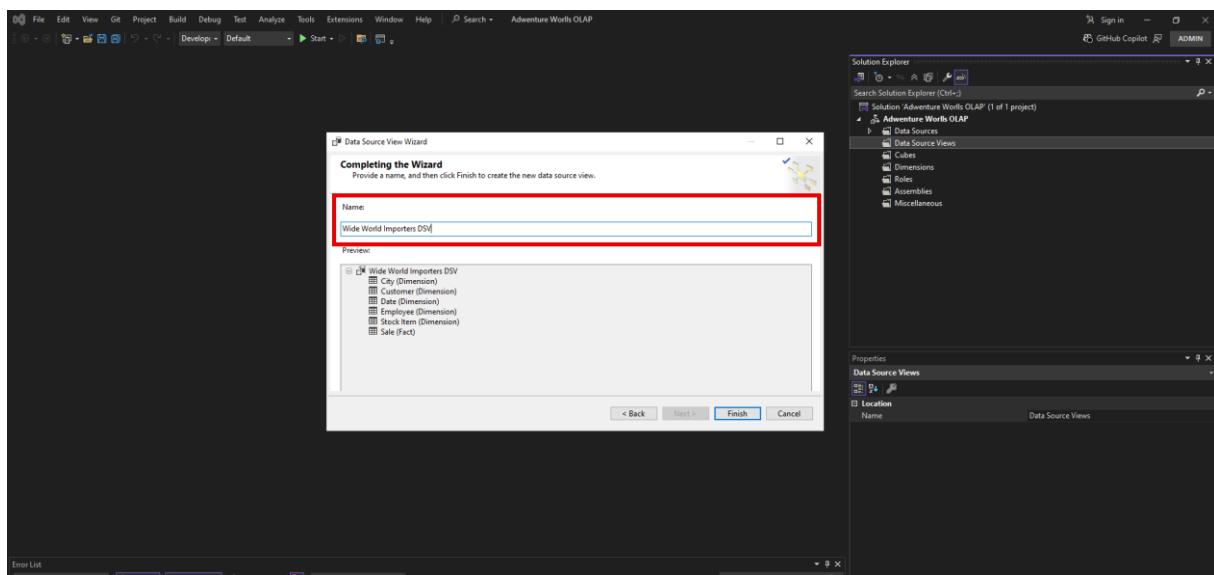
Perspektywy (Sales Overview, Employee Performance) – uproszczone widoki dla działów analitycznych,

Tłumaczenia nazw elementów modelu na język polski,

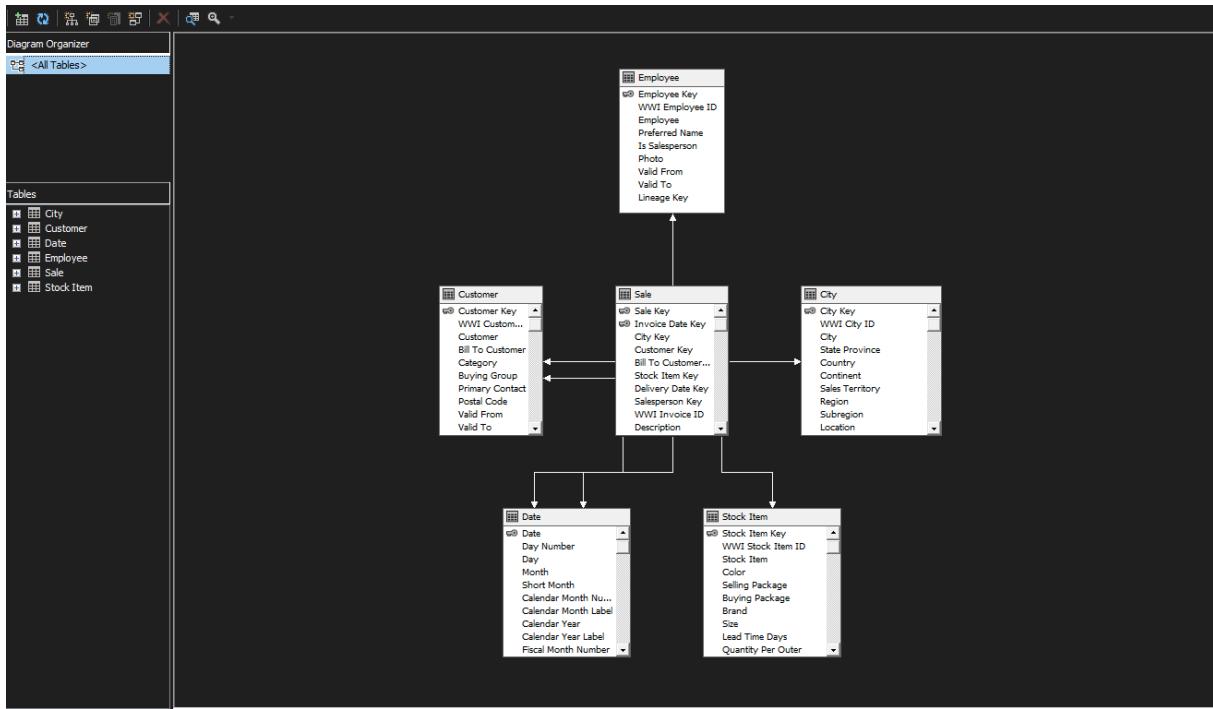
Relacje między tabelami umożliwiające analizę typu drill-down (np. Rok → Miesiąc → Dzień).



Potrzebne tabele do realizacji naszego celu biznesowego



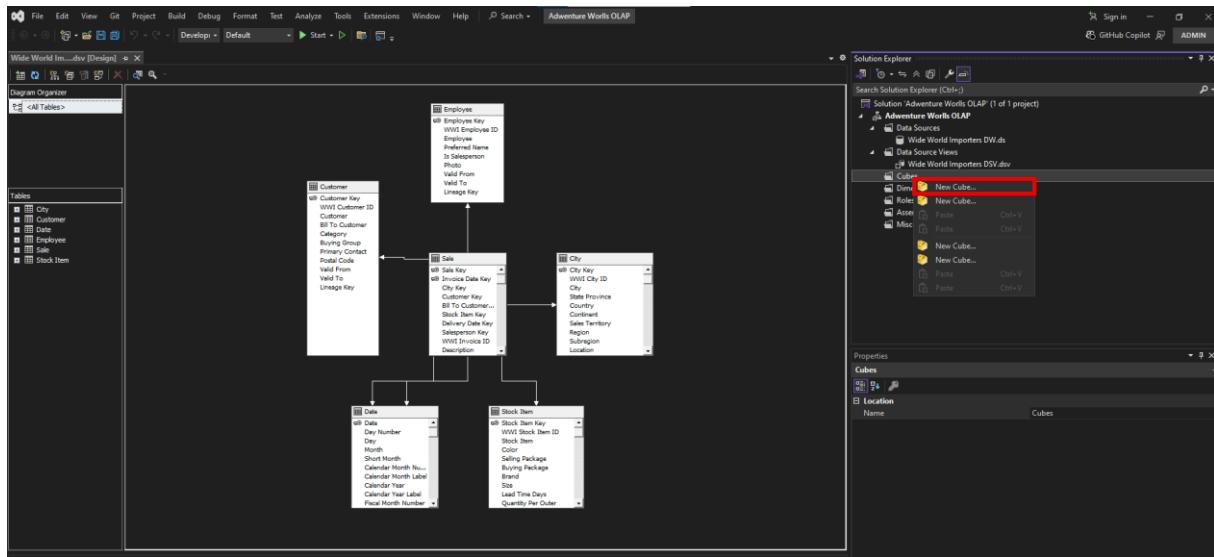
Zmieniam nazwę aby było jasne, że jest to Data Source View a nie coś innego



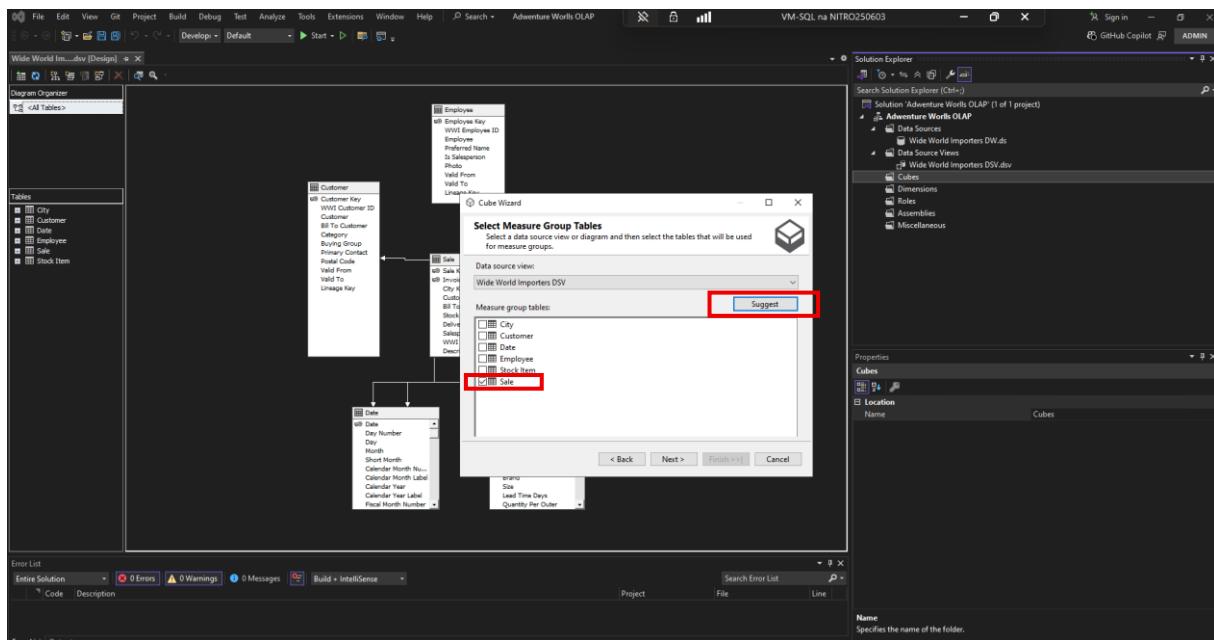
Schemat logiczny relacji

Patrząc na powyższy schemat zauważmy, że powstał nam klasyczny model „gwiazdy” Fact-Dimension. Z tego wynika, iż tabela faktów Sale znajduje się w środku a dookoła mamy pięć wymiarów, które dają różne perspektywy do analizy np. wymiar czasu (dimension date) sprzedaż wg dni, miesięcy, kwartałów, lat. W OLAP będziemy mogli utworzyć hierarchie np. Rok -> Miesiąc -> Dzień. Wymiar klienta (dimension customer) może nam powiedzieć kto kupuje np. kategoria klienta, kraj, grupa zakupowa. Można stworzyć hierarchię np. Kategoria -> Klient. Tego typu analizy dotyczą też pozostałych wymiarów. Te dwie, które są przedstawione na górze stanowią przykład sposobu zrozumienia zadania, które wykonujemy.

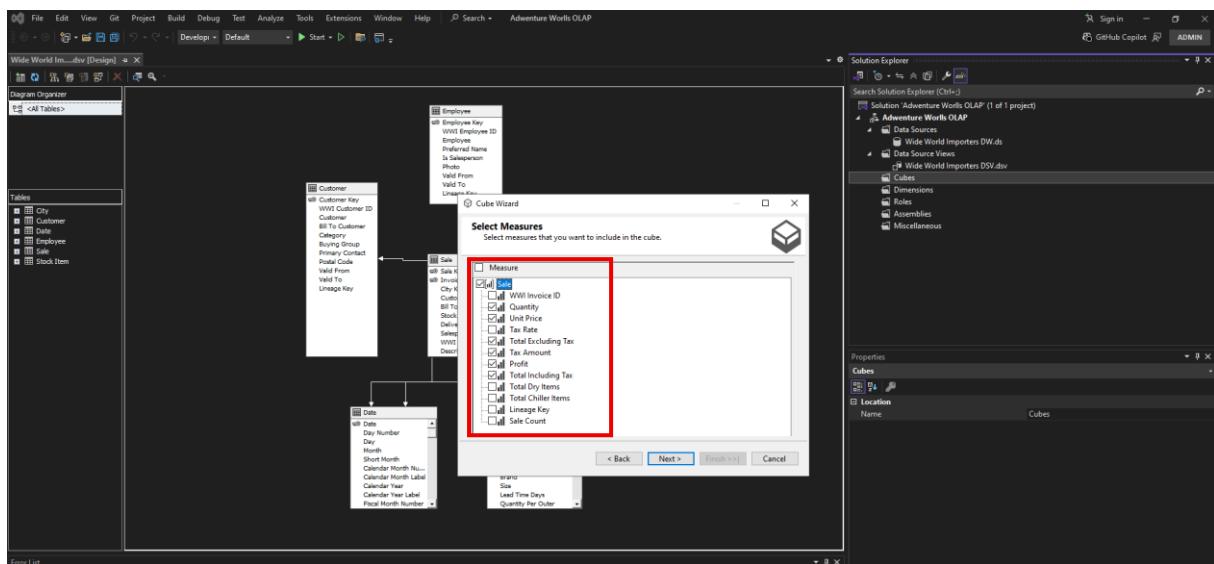
- Tworzenie i konfiguracja kostki w modelu wielowymiarowym (Cube):



W Solution Explorer klikamy PPM na Cube > New Cube... w celu stworzenia nowej kostki



W powyższym oknie musimy wybrać tabelę, która służy jako tabela faktów (Fact Table). W naszym projekcie jest to tabela Sale, aczkolwiek gdybyśmy mieli bardziej rozbudowany model możemy kliknąć ikonę Suggest i wtedy program sam nam zaproponuje tabelę, którą należy wybrać



Wybór odpowiednich miar

Quantity - liczba sprzedanych produktów – podstawowa miara ilościowa

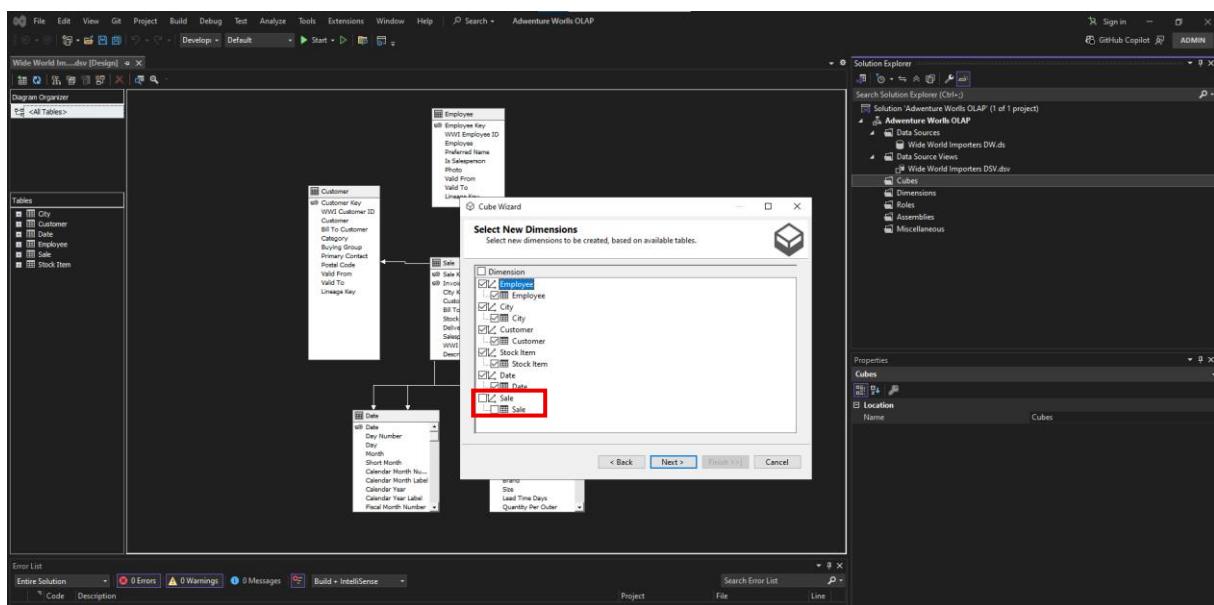
Unit Price - cena jednostkowa – przydatna do analizy marży i średniej ceny

Total Excluding Tax - wartość sprzedaży netto (bez podatku)

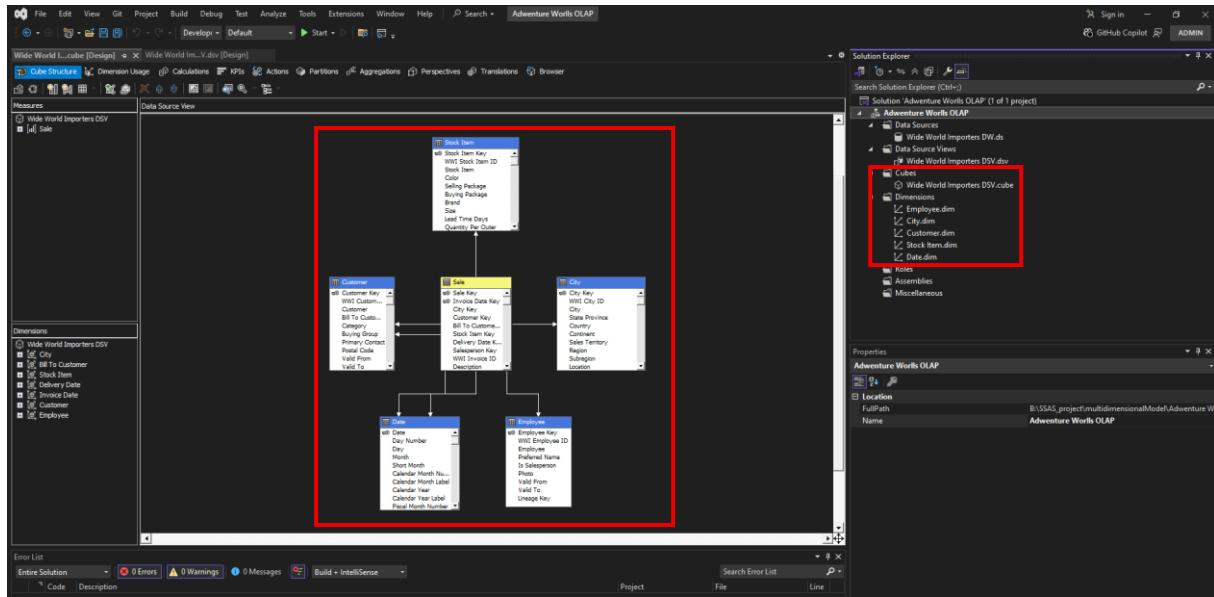
Total Including Tax - wartość sprzedaży brutto – dobra do analizy przychodów

Tax Amount - suma podatku – pomocna przy analizach podatkowych

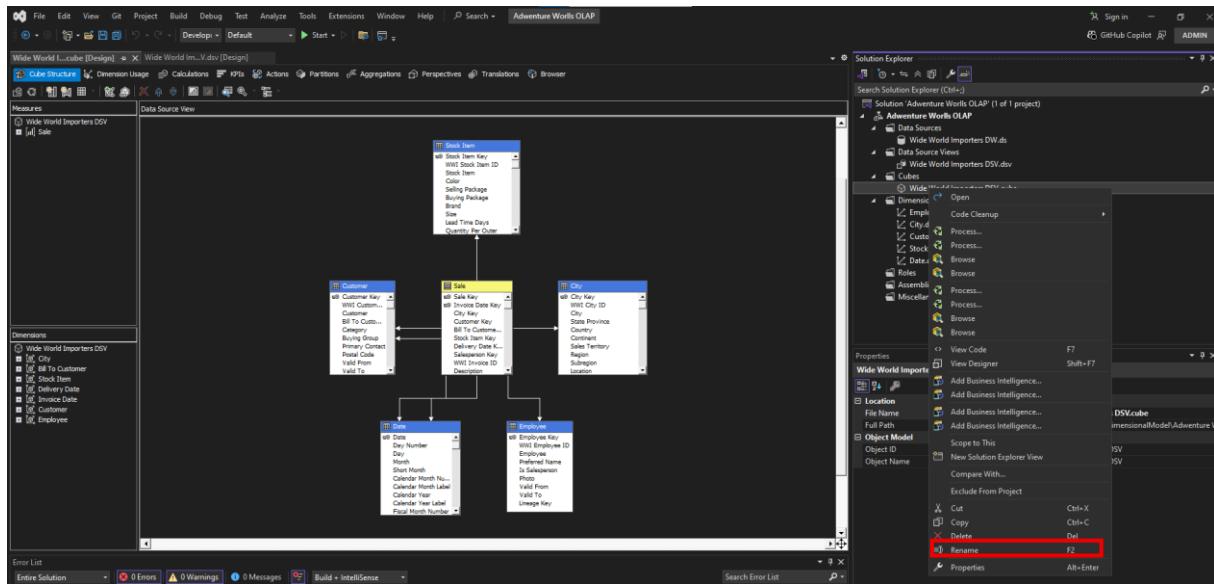
Profit - zysk ze sprzedaży – miara strategiczna do KPI



Odznaczamy Sale, ponieważ jest to tabela faktów a nie wymiar!

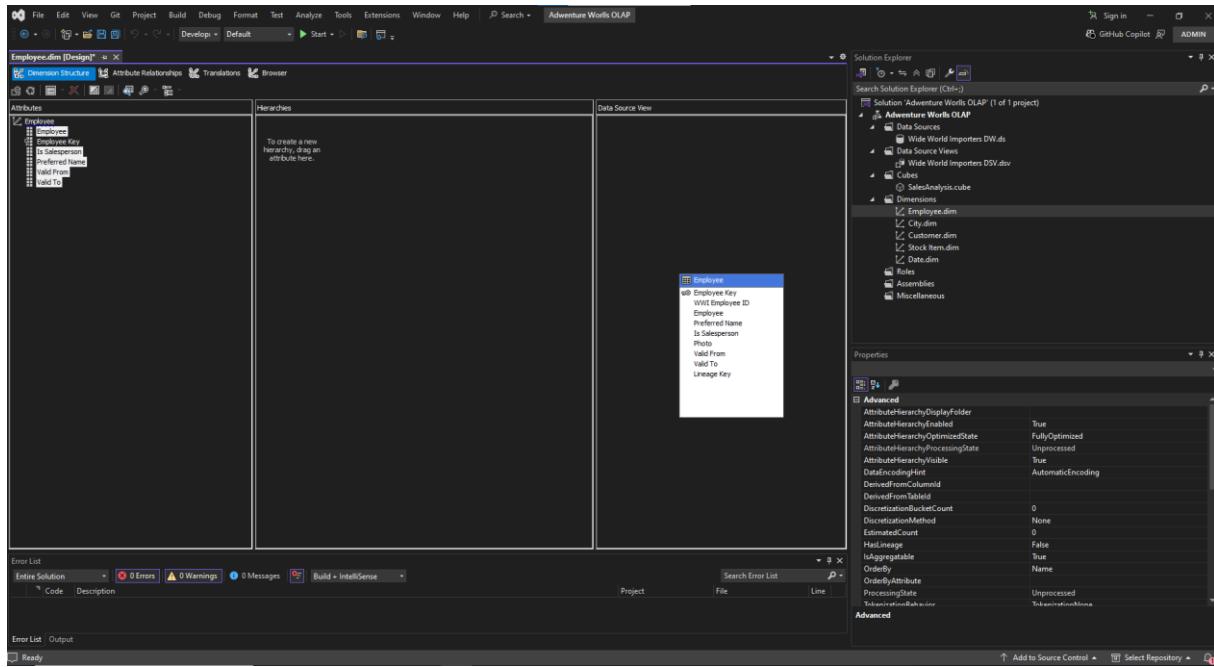


Widok projektu po poprawnym utworzeniu naszej kostki (Cube)



W celach zachowania porządku możemy zmienić nazwę naszej kostki na np. SalesAnalysis

- Konfigurowanie wymiarów:



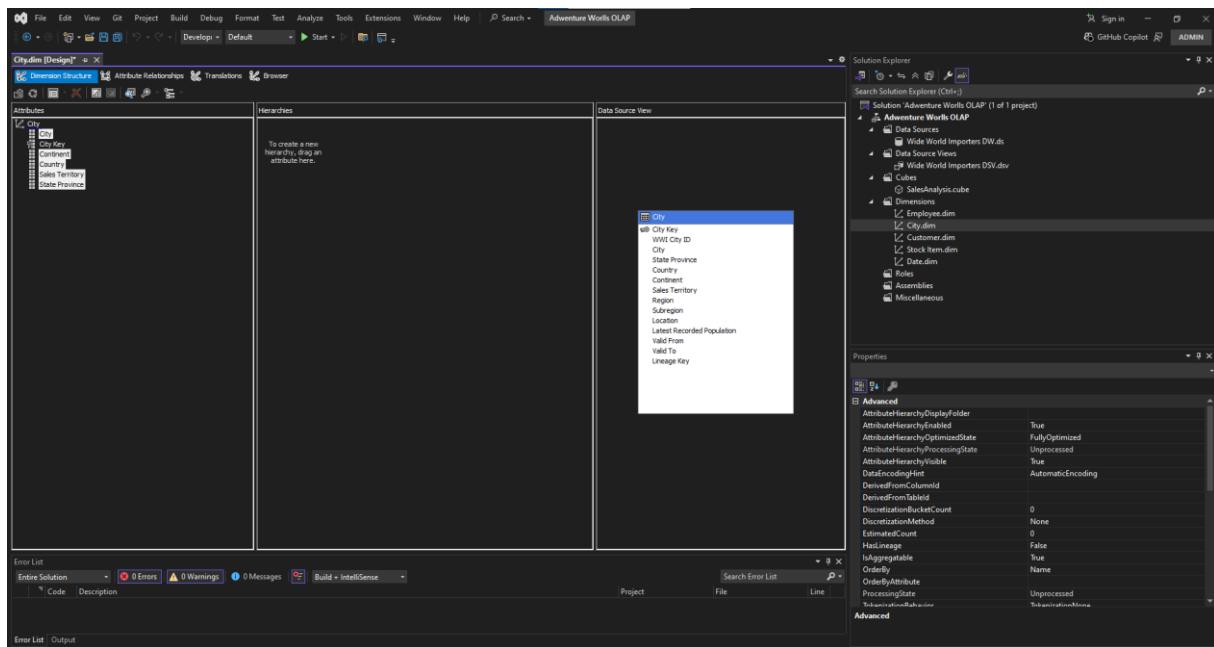
Przeciągamy potrzebne kolumny do sekcji atrybutów

Employee - nazwa pracownika (główna etykieta do analiz).

Preferred Name - imię preferowane, przydatne przy raportach personalnych.

Is Salesperson - flaga sprzedawcy (do filtrowania analizy tylko po sprzedawcach).

Valid From, Valid To - okres zatrudnienia (do filtrów czasowych lub zmian etatów).



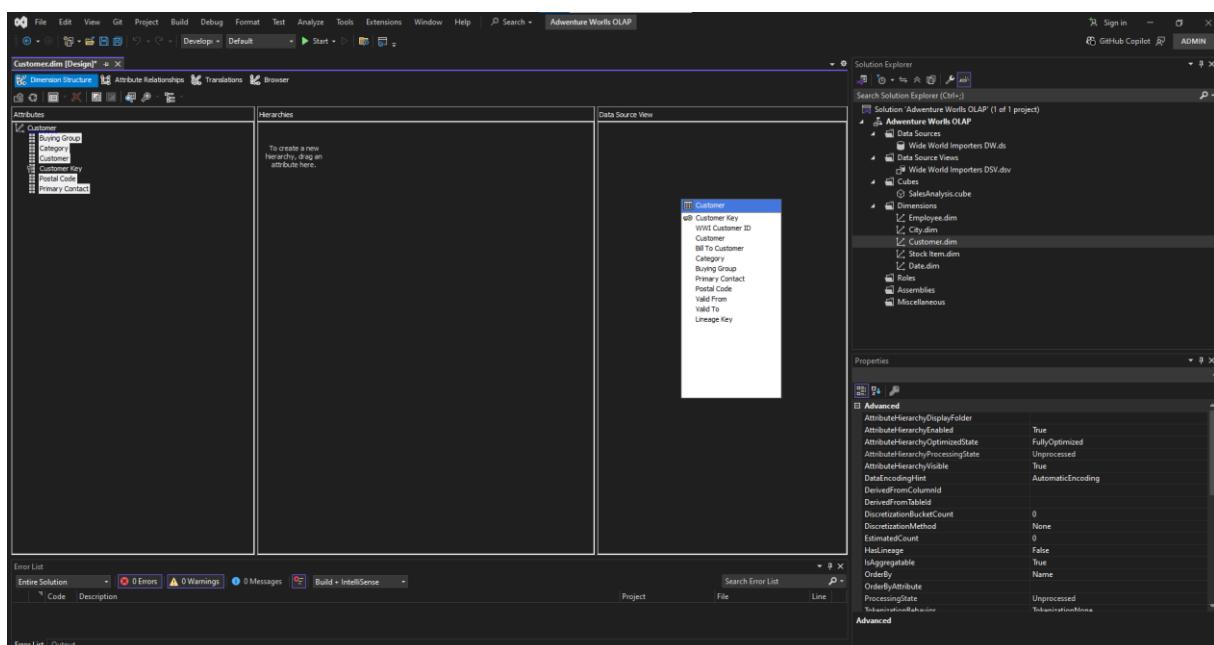
City - nazwa miasta

State Province - województwo

Country - kraj

Continent - kontynent

Sales Territory- strefa sprzedażowa (biznesowa klasyfikacja).



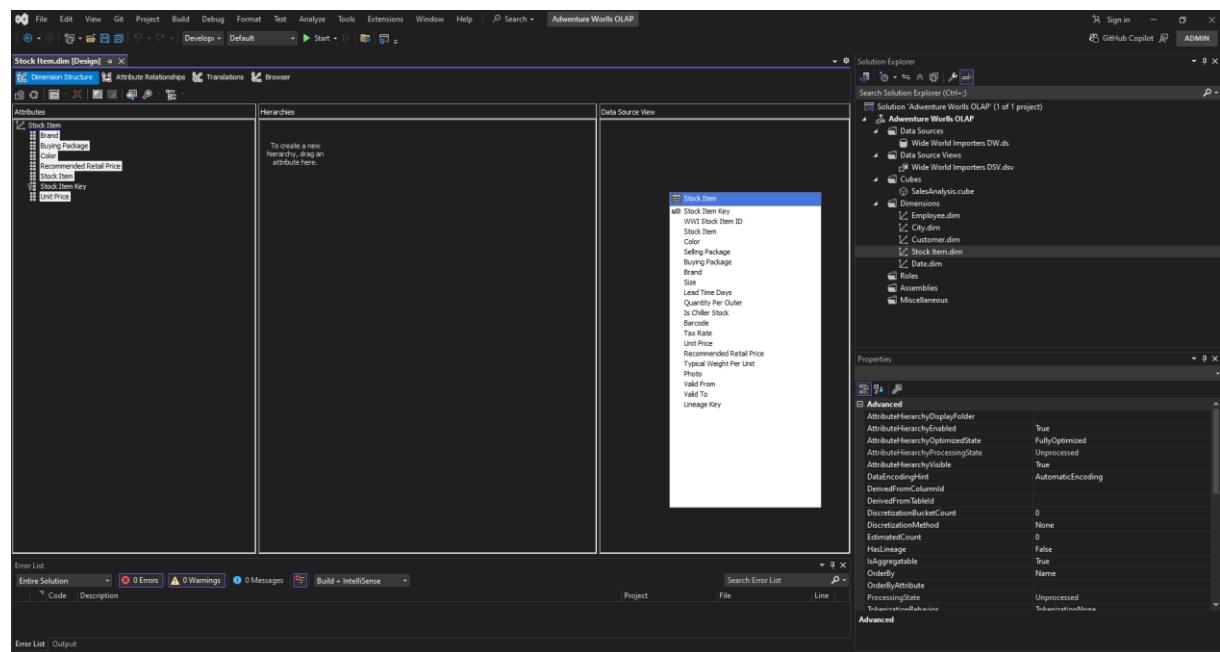
Customer - nazwa klienta (główna etykieta).

Category - segment klienta (np. detaliczny, hurtowy do analizy grup).

Buying Group - grupa zakupowa (analiza wg powiązań biznesowych).

Primary Contact - osoba kontaktowa (dla analiz operacyjnych / raportów sprzedaży).

Postal Code - lokalizacja klienta (może być użyte do łączenia z geolokalizacją lub regionem).



Stock Item - nazwa produktu (podstawowa etykieta).

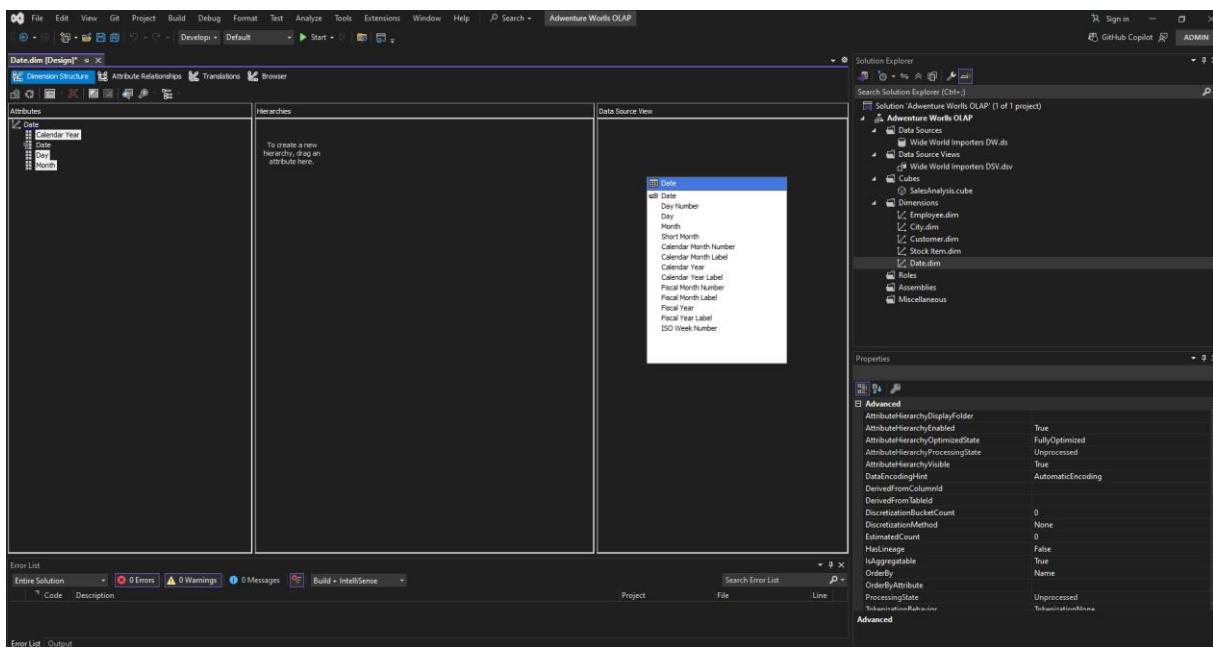
Brand - marka produktu (analiza wg producentów).

Color - kolor (częsty atrybut produktowy).

Buying Package - sposób pakowania (np. karton, sztuka – przydatne przy logistyce).

Recommended Retail Price - sugerowana cena detaliczna (można porównywać z ceną sprzedaży).

Unit Price - aktualna cena jednostkowa (do analiz marży).

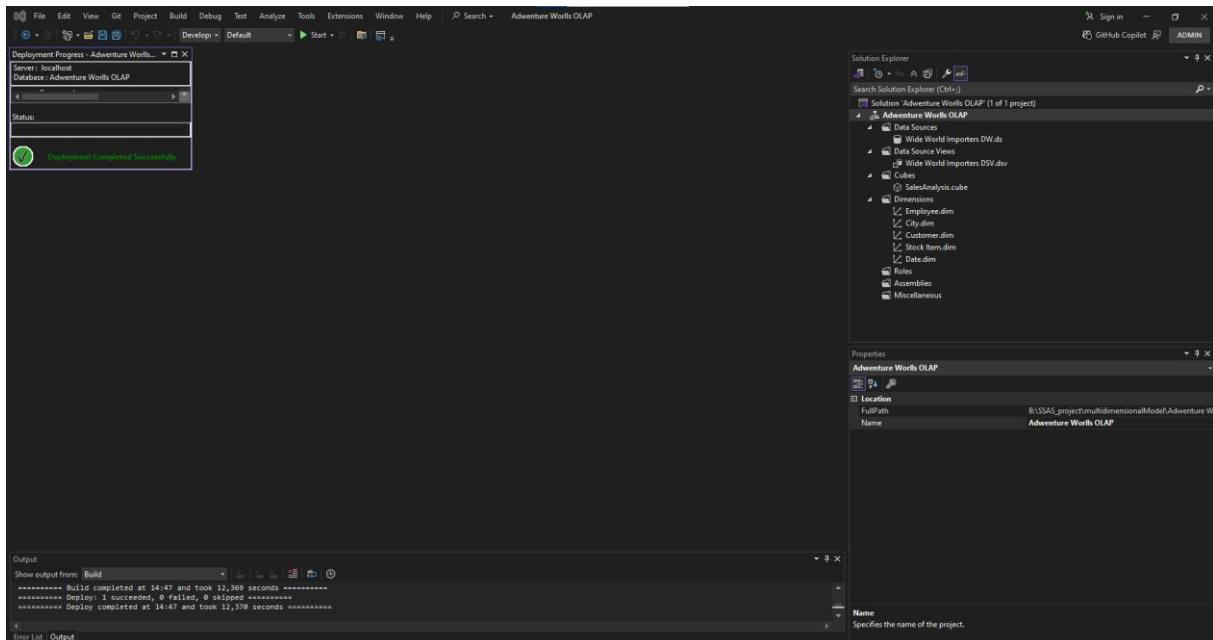


Calendar Year - rok kalendarzowy (główna osz analizy).

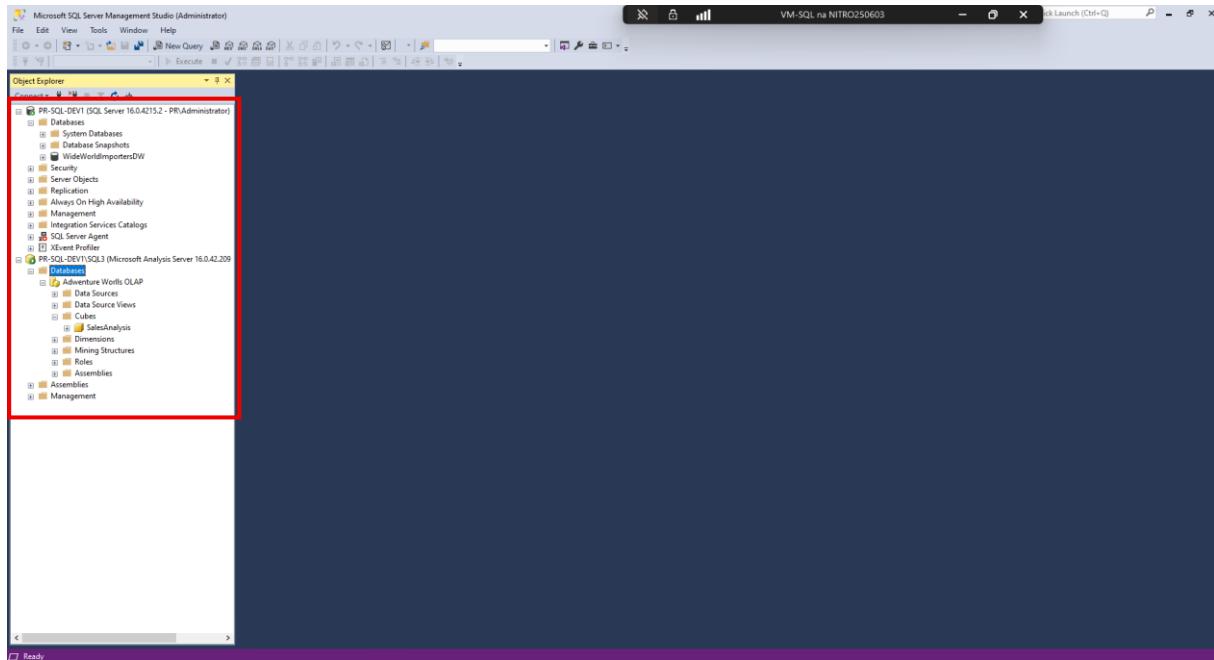
Month - nazwa miesiąca (czytelna dla użytkownika).

Day - dzień miesiąca.

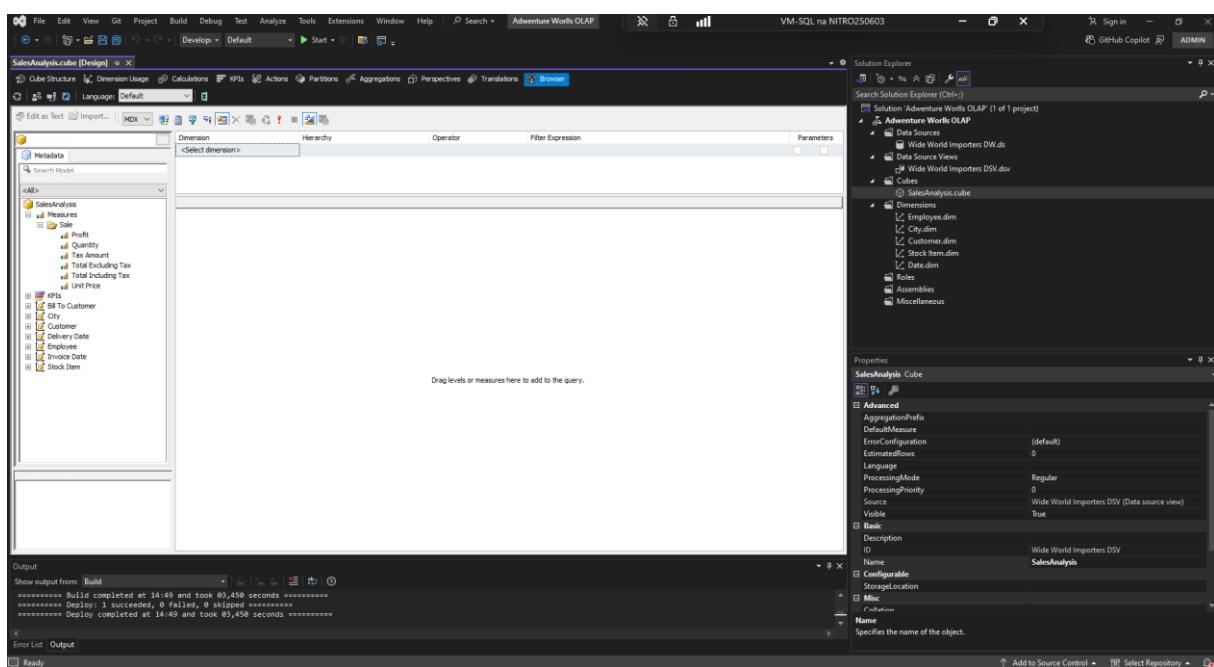
- Podgląd kostki (Browse the Cube):



Widok po poprawnym Deploy naszej kostki



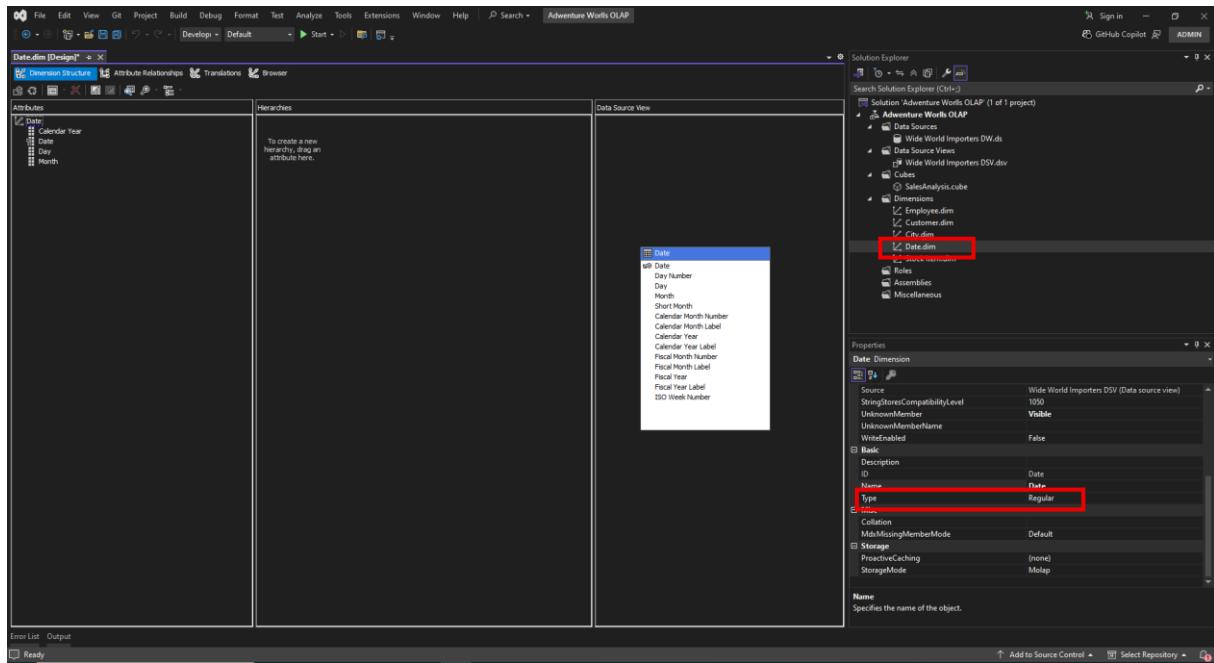
Podgląd struktury w SSMS



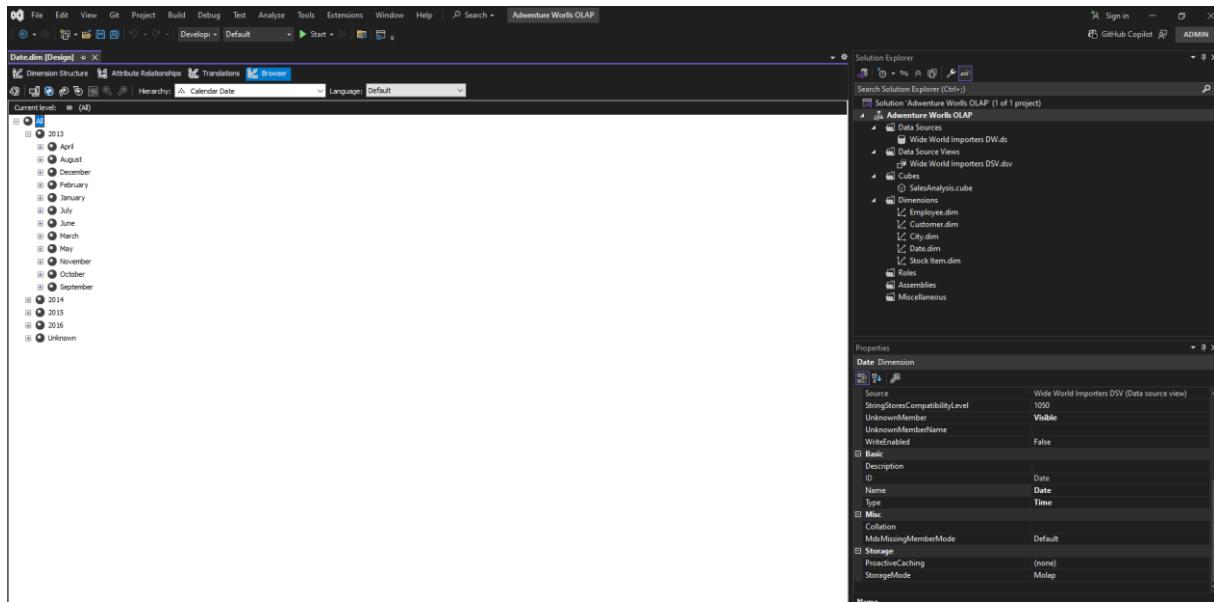
Widok w zakładce Browse naszej kostki

- Utworzenie hierarchii dla Date:

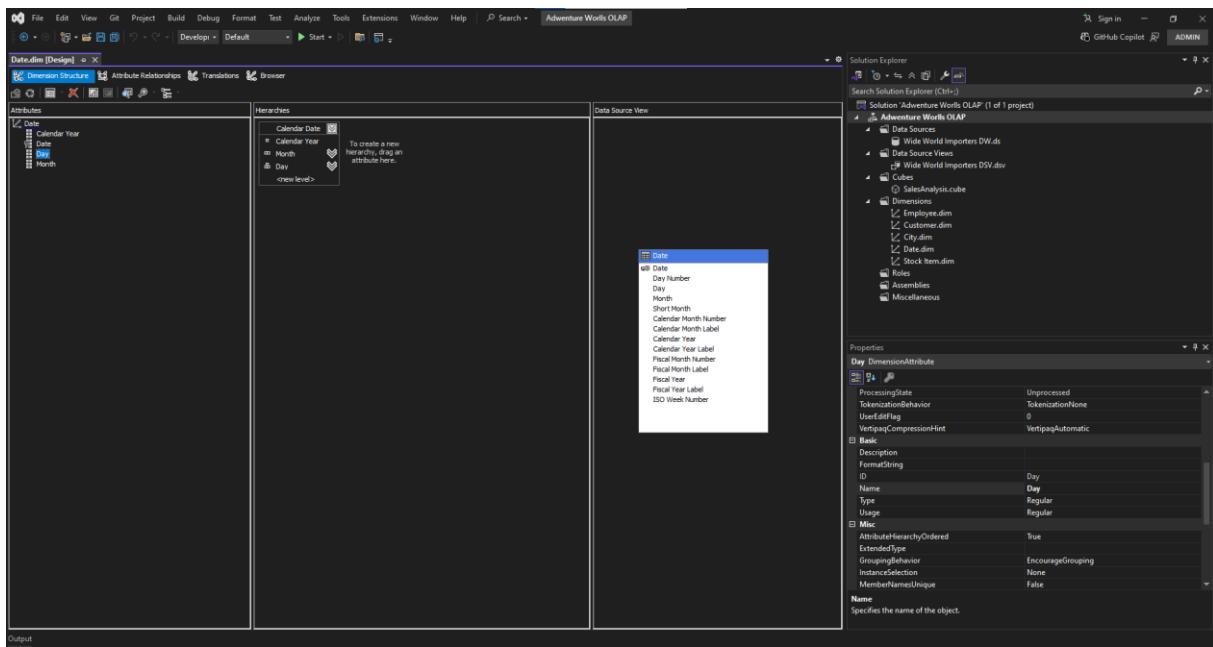
Hierarchii nie trzeba tworzyć dla wszystkich wymiarów. Można je stworzyć tam gdzie istnieje „logiczny” porządek danych czyli np. Year > Month > Day



Włączamy widok wymiaru Date, dodatkowo w Properties zmieniamy Type z „Regular” na „Time”



Utworzona hierarchia w zakładce Browse



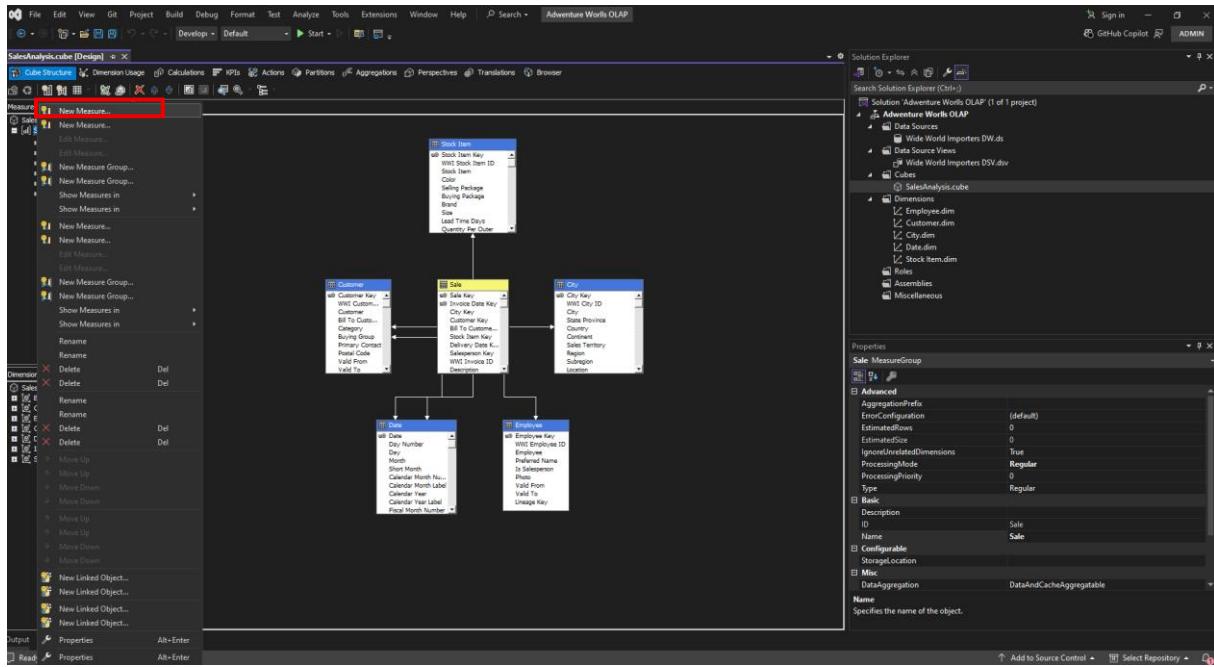
Utworzona hierarchia

Warto zauważyc żółte trójkąty które sygnalizują, że należy zdefiniować te relacje a można to zrobić w zakładce Attribute Relationships

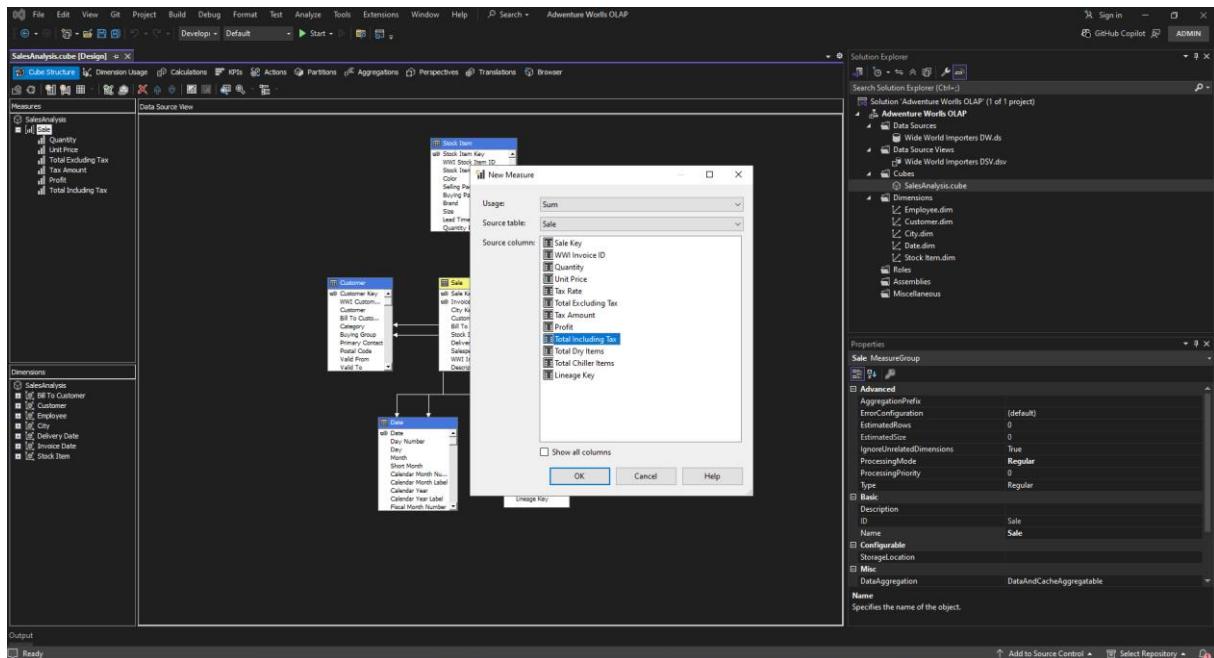
- Miary (Measures):

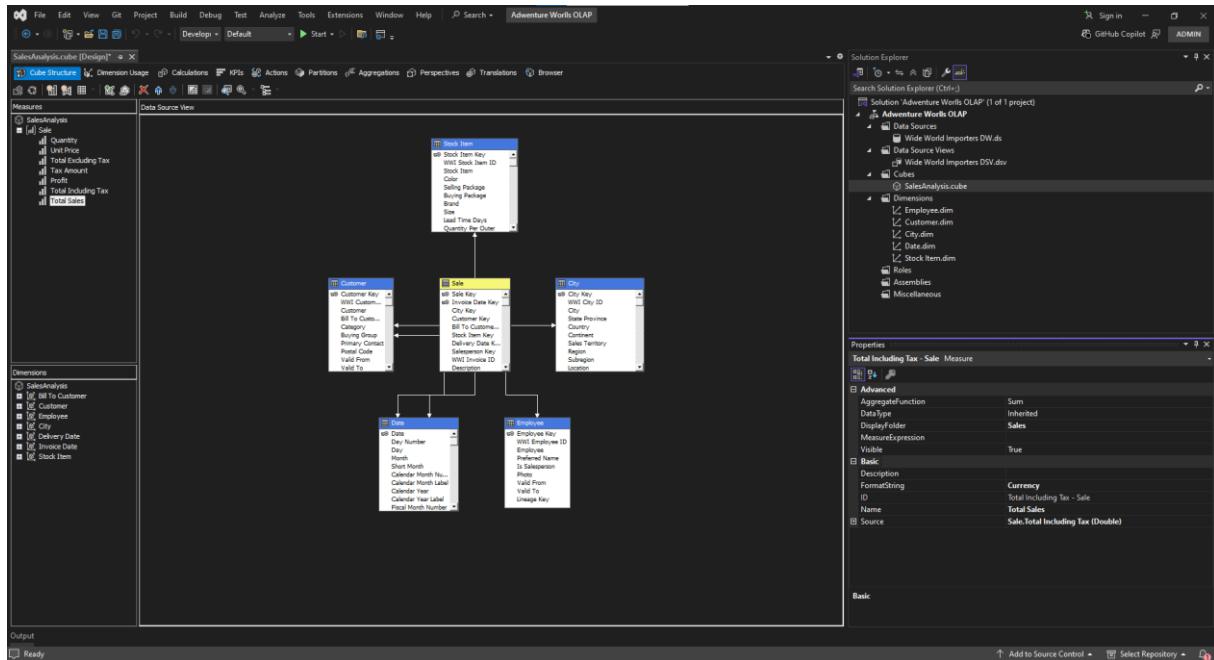
W swoim projekcie proponuje stworzenie trzech miar:

- Total Sales – całkowita wartość sprzedaży, pokazuje ile łącznie zarobiliśmy na sprzedaży razem z podatkiem. Źródłem będzie kolumna Total Including Tax w tabeli Sale. Bazową miarę ustawimy na SUM. Miera będzie pozwalać analizować całkowite przychody ze sprzedaży w ujęciu czasowym (rok, miesiąc, dzień) oraz geograficznym (kraj, miasto). Dzięki niej można łatwo porównywać poziom sprzedaży pomiędzy regionami, klientami lub produktami.
- Quantity Sold- będzie służyć do analizy wolumenu sprzedaży. Pozwala ocenić, które produkty lub kategorie cieszą się największym popytem oraz jak zmienia się sprzedaż ilościowa w czasie.

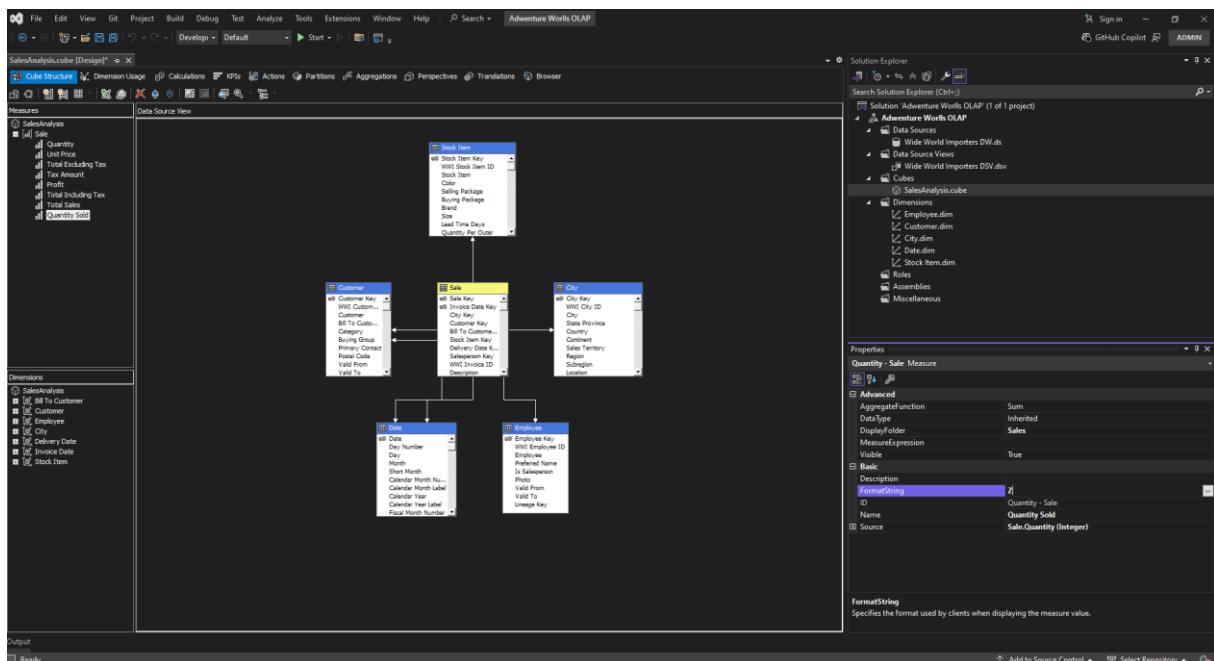


Tworzymy nową miarę, zaczynając od Total Sales

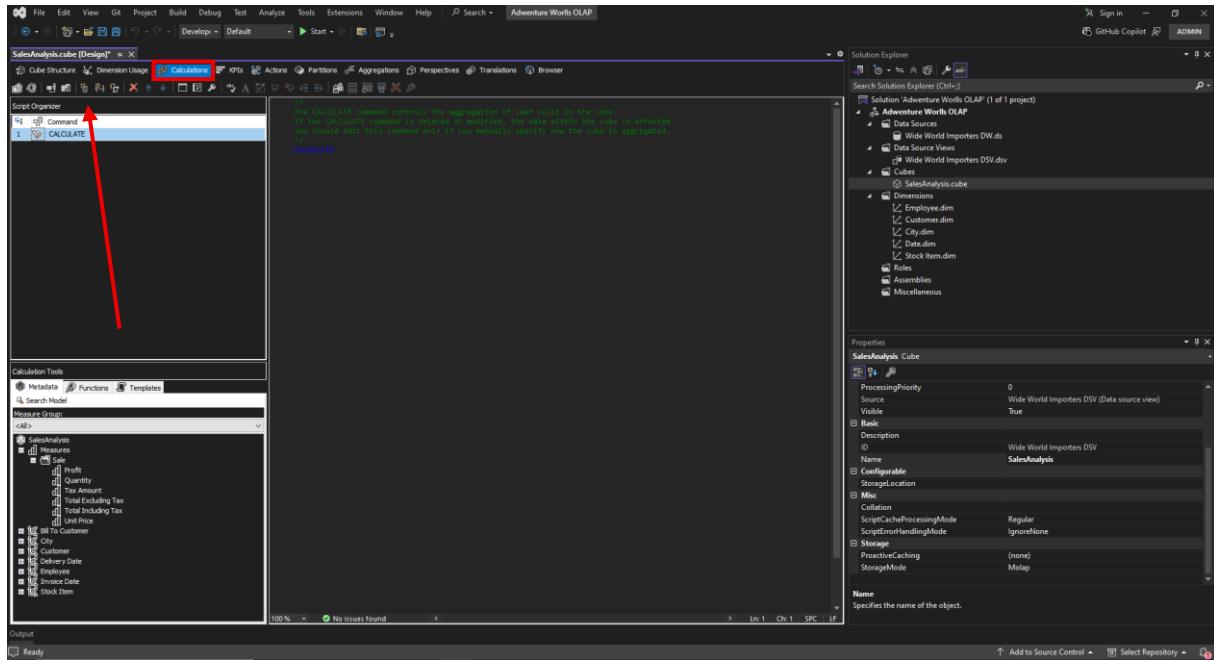




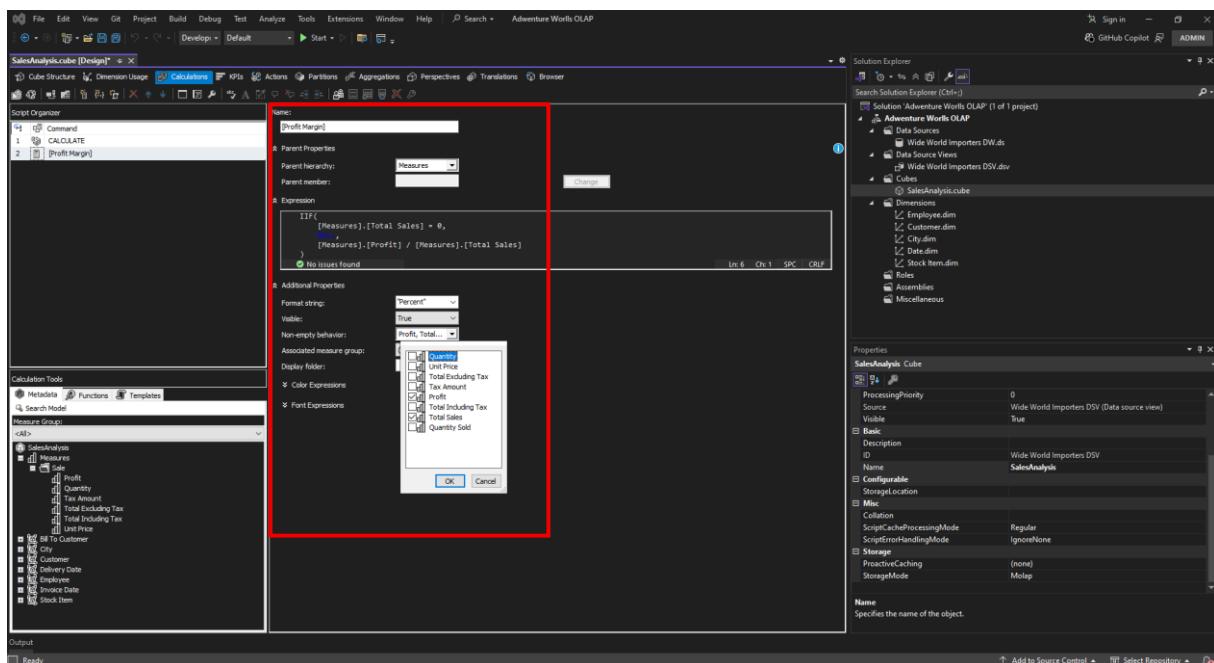
Nowa miara została utworzona

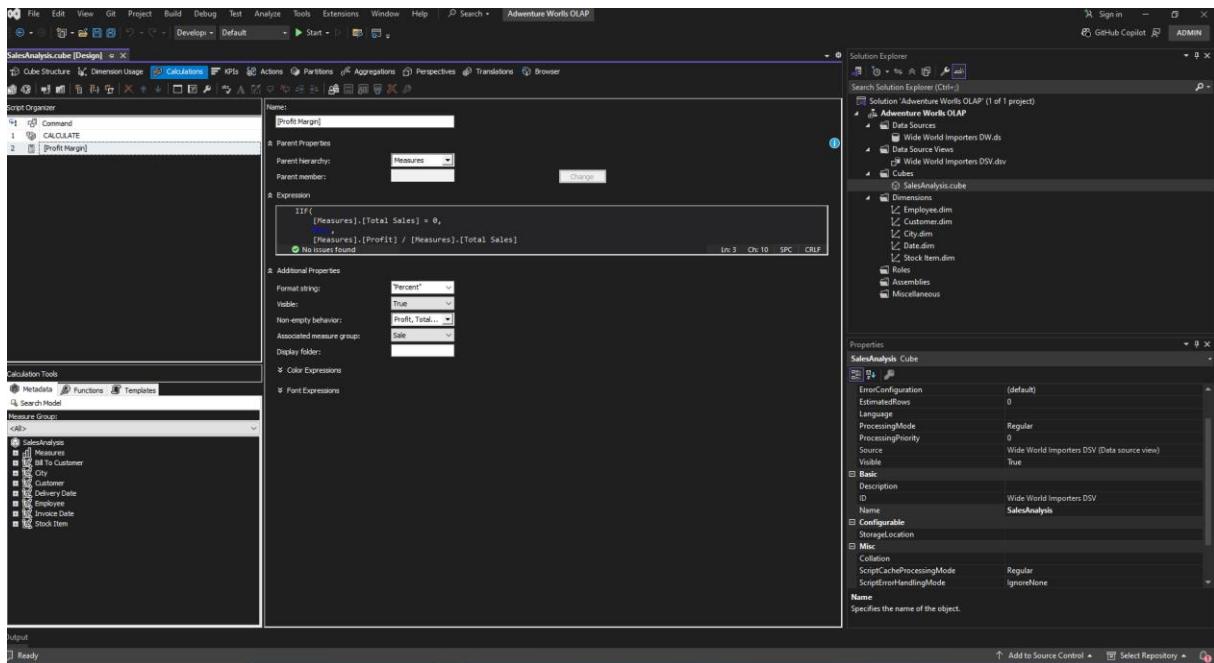


Nowa miara Quantity Sold została utworzona

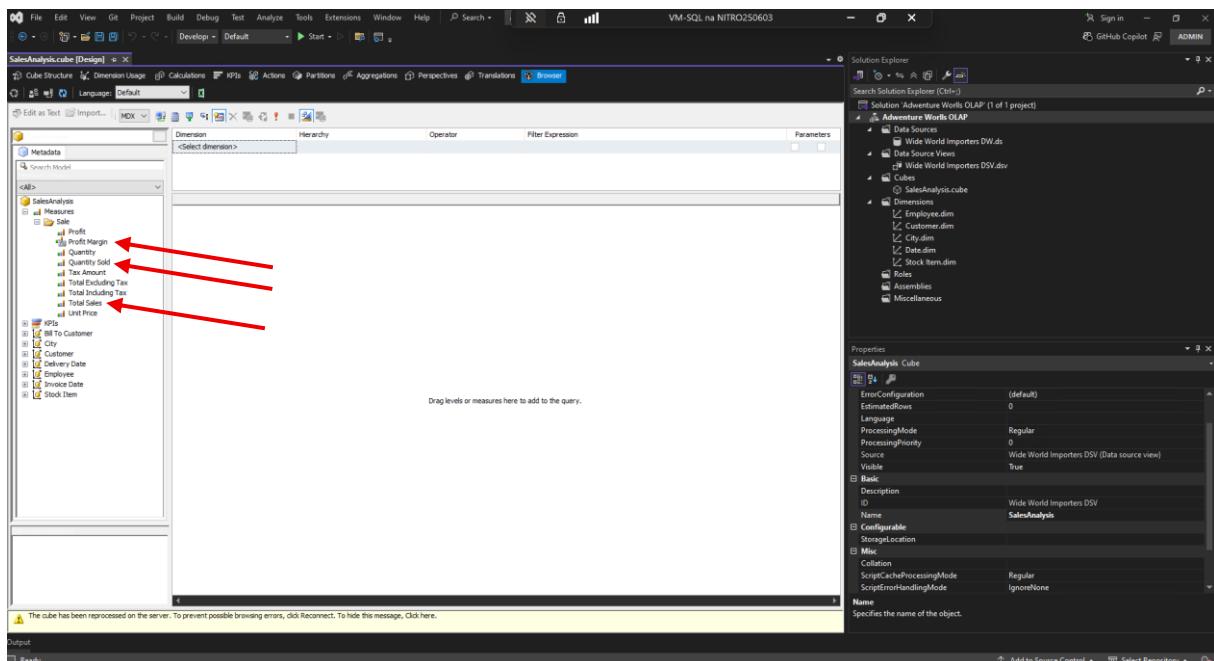


Tworzenie miary obliczeniowej Profit Margin odbywa się bezpośrednio w edytorze kalkulacji kostki





Po tym możemy zrobić Process a następnie Deploy



W zakładce Browse widzimy nowo utworzone miary

Calendar Year	Total Sales	Quantity Sold	Profit	Profit Margin
2013	\$356,327,54,00002	240,657	227,832,25	0,43216085...
2014	\$74,091,68,00001	256,741	240,384,25	0,4320910...
2015	620,922,80,99997	274,026	269,760,65,00001	0,4346822...
2016	299,702,11,00003	124,139	111,747,65,55	0,4327811...

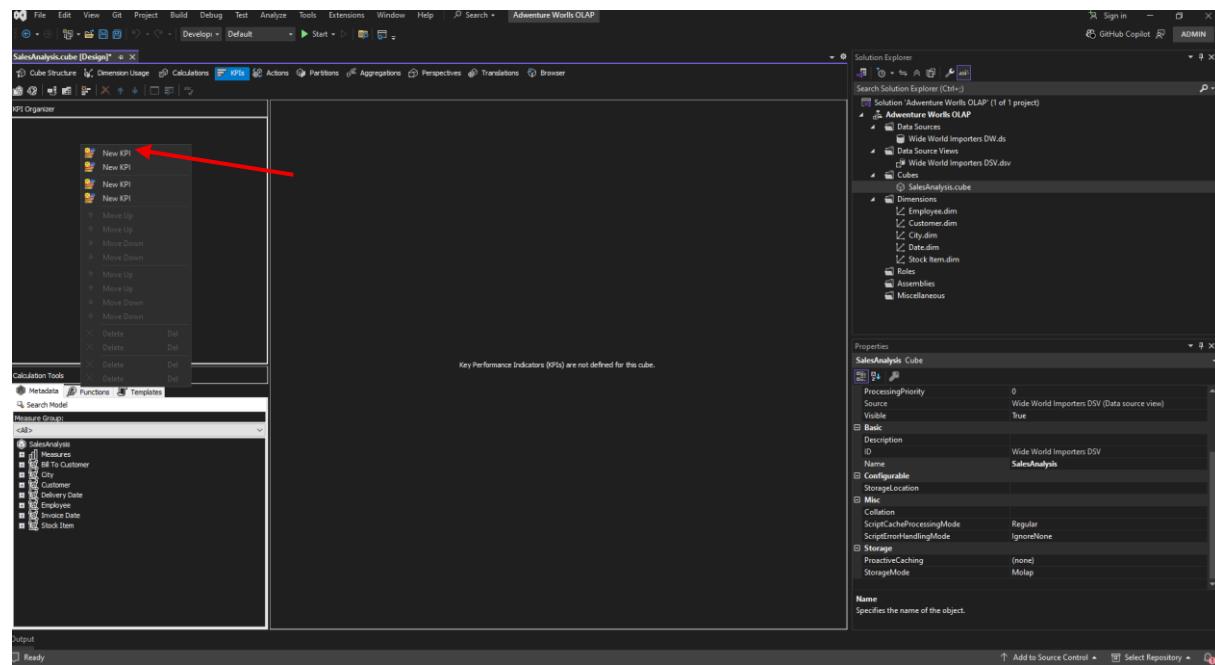
Przykładowy wynik

Przeprowadzono test działania miar w modelu wielowymiarowym. W zakładce Browse zestawiono wartości miar dla kolejnych lat kalendarzowych (Calendar Year) na podstawie wymiaru Date (Invoice Date). Uzyskane wyniki potwierdziły poprawność obliczeń: łączna sprzedaż (Total Sales), liczba sprzedanych produktów (Quantity Sold), zysk (Profit) oraz marża zysku (Profit Margin) generują logiczne i spójne wartości w czasie, wskazując na stabilną rentowność (~43%).

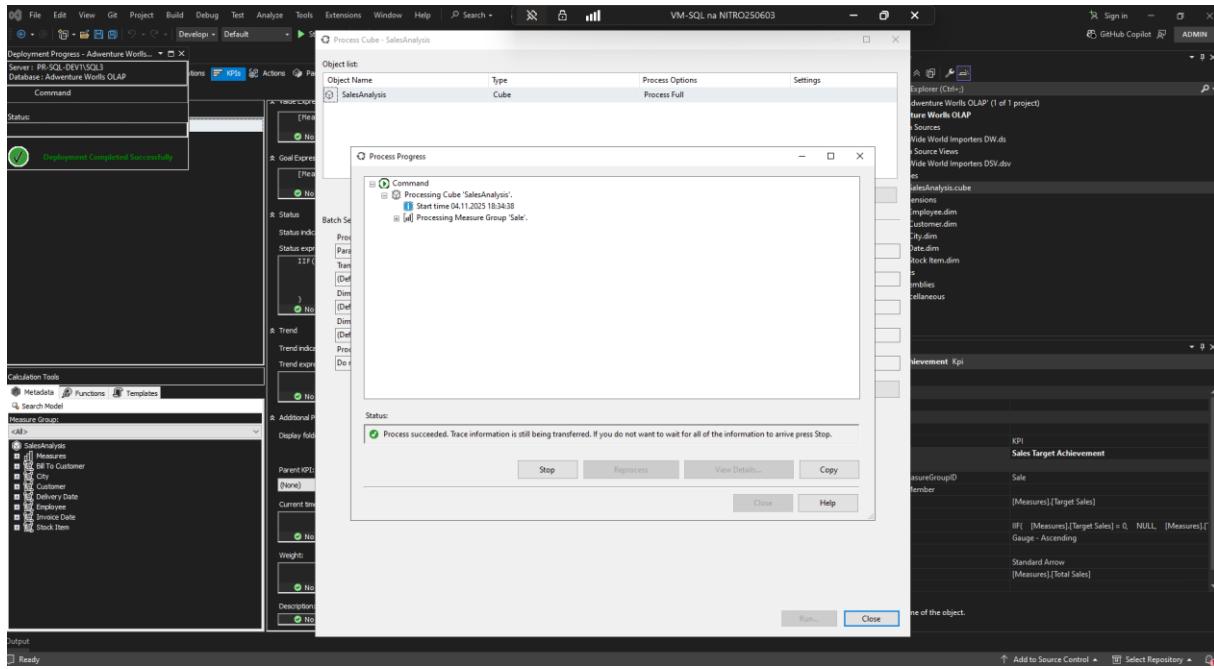
Utworzenie miary typu Calculated, która będzie potrzebna do KPI

- Tworzenie KPI (stopień realizacji celu sprzedaży):

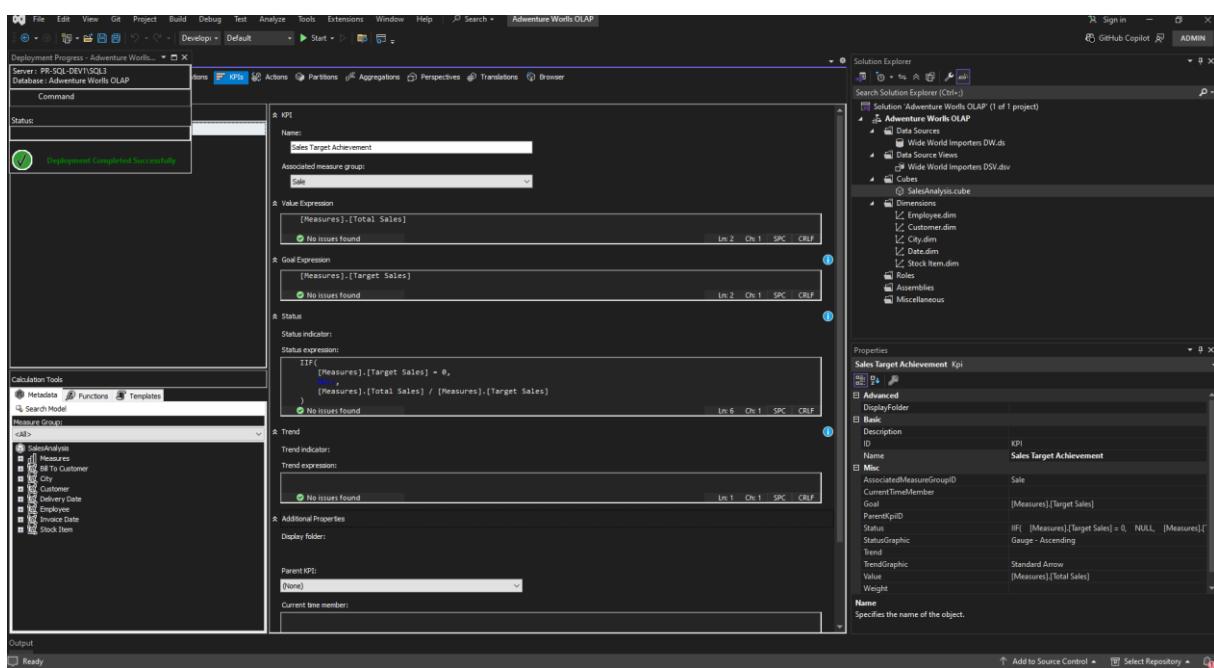
Będę porównywać rzeczywistą sprzedaż (Total Sales) z celem (Target Sales). Dzięki temu KPI pokaże, na ile procent cel sprzedaży został osiągnięty.



Tworzenie nowego KPI



KPI zostało poprawnie utworzone



Parametry, które zostały wpisane

The screenshot shows a Microsoft Excel window with a PivotTable named 'PivotTable1'. The PivotTable Fields pane on the right side lists fields under 'KPIs' such as Sales Target Achievement, Value (Total Sales), Goal (Target Sales), and Status. The main table area displays sales data for various regions, with a 'Grand Total' row at the bottom.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Row Labels		Sales Target Achievement	Sales Target Achievement Goal	Sales Target Achievement Status													
1 Row Labels		\$2 529 291,07	10000000														
2 External		\$22 855 077,65	10000000														
3 Far West		\$23 169 368,53	10000000														
4 Great Lakes		\$29 613 677,16	10000000														
5 Midwest		10000000															
6 N/A		10000000															
7 New England		\$8 847 961,54	10000000														
8 Plains		\$26 796 087,55	10000000														
9 Rocky Mountain		\$12 734 834,76	10000000														
10 Southeast		\$43 992 233,48	10000000														
11 Southwest		\$27 504 907,71	10000000														
12 Grand Total		\$198 643 439,45	10000000														

Przykład poprawnego działania w tabeli przestawnej w Excel

- Perspektywy (dzięki nim użytkownik nie będzie widział całego złożonego modelu a tylko wybrane wymiary, miary i KPI)

Perspektywa: Sales Performance

Cel: analiza wyników sprzedaży i rentowności.

Miary:

Total Sales

Quantity Sold

Profit

Profit Margin

Target Sales

KPI: Sales Target Achievement

Wymiary:

Date

City

Customer

Employee

- Pozwala śledzić trendy, marże i realizację celów KPI w czasie i w podziale na regiony lub handlowców.

Perspektywa: Product & Stock Analysis

Cel: monitorowanie struktury produktów i zapasów.

Miary:

Quantity Sold

Unit Price

Total Excluding Tax

Tax Amount

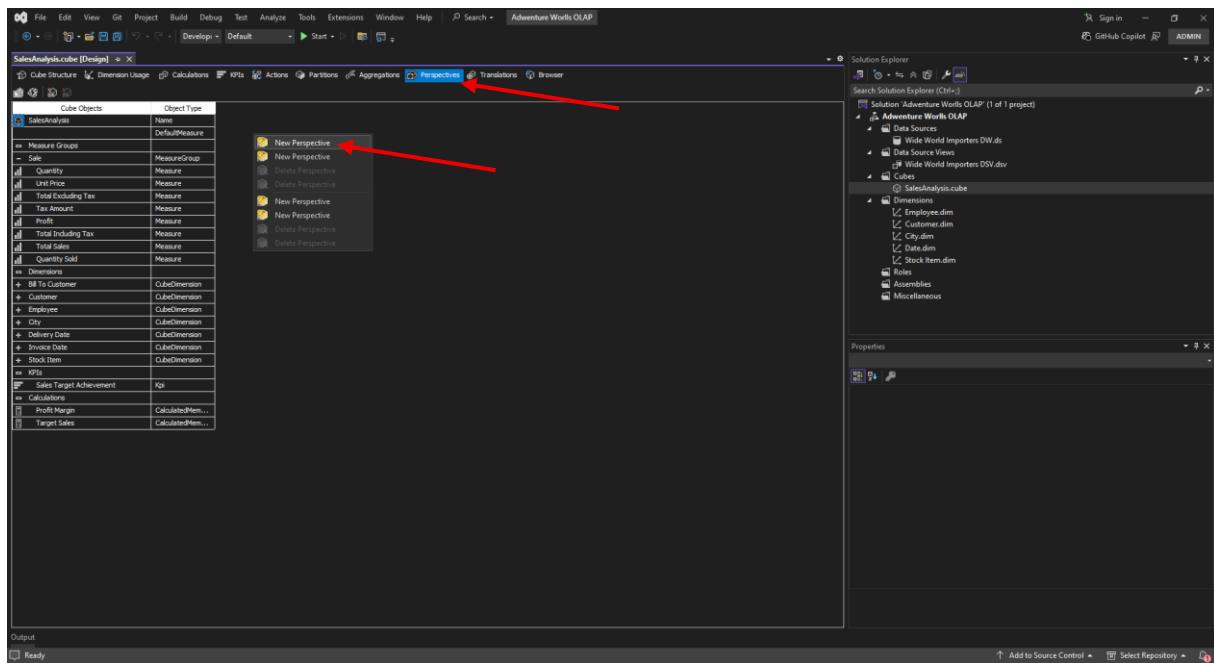
Wymiary:

Stock Item

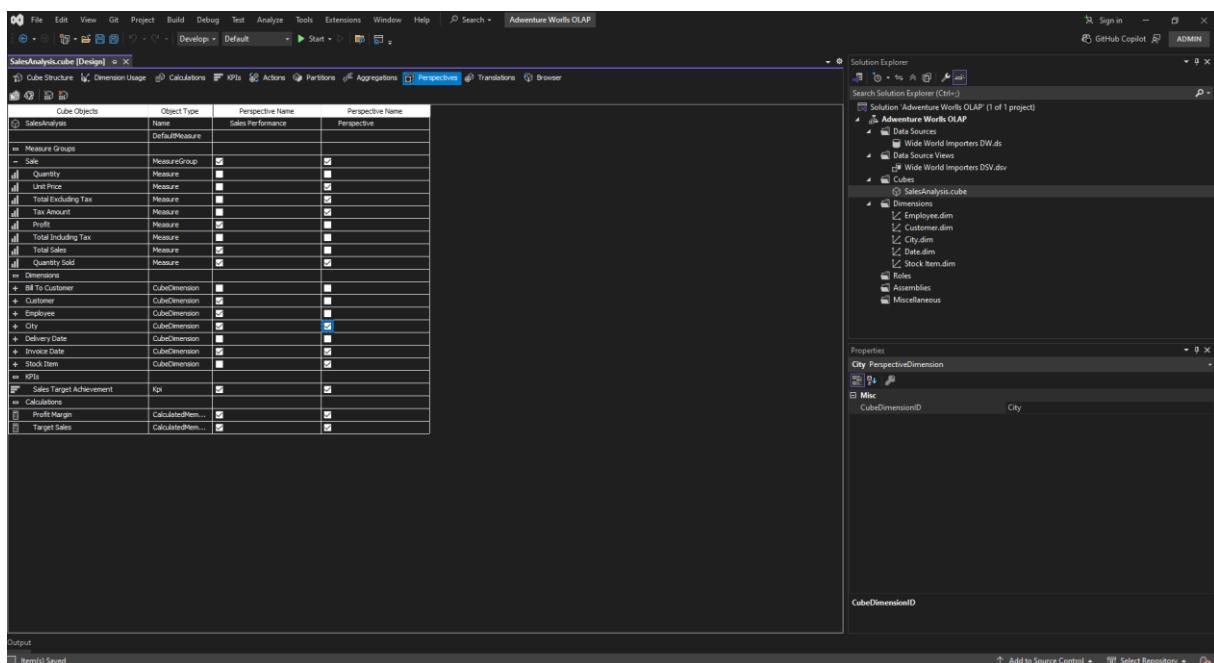
City

Date

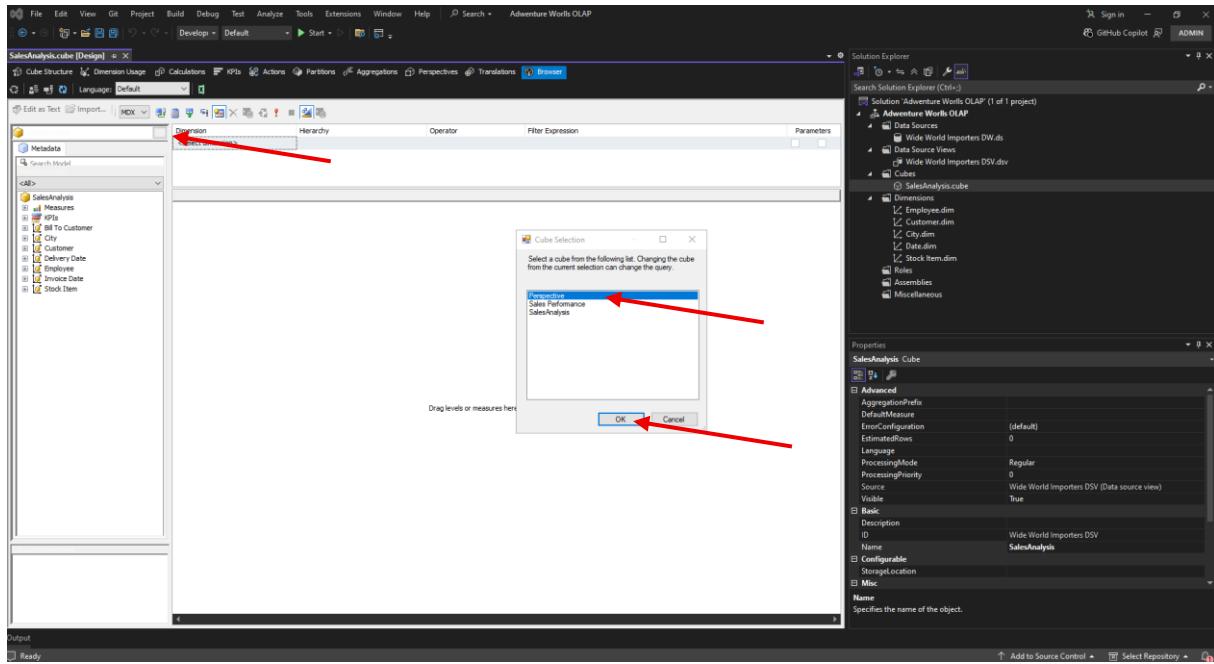
- Pozwala analizować sprzedaż poszczególnych produktów, wpływ podatków i zapasów, a także obserwować sezonowość popytu w czasie.



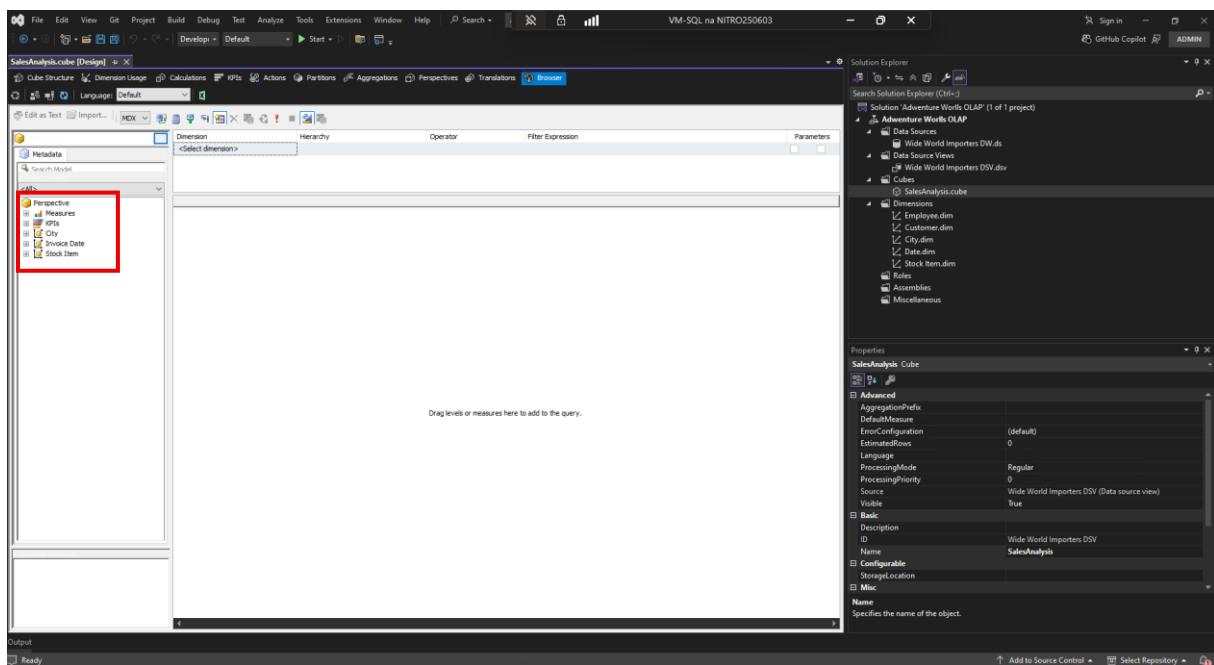
Tworzymy nową perspektywę



Utworzono perspektywy

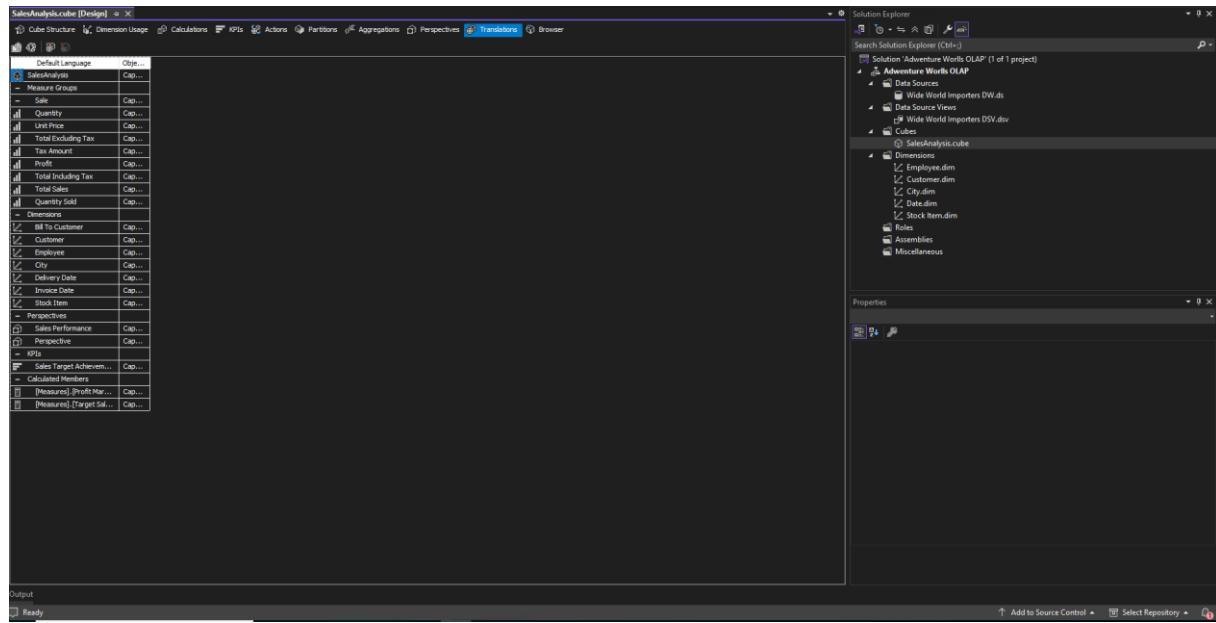


Sprawdzanie perspektyw w zakładce Browser

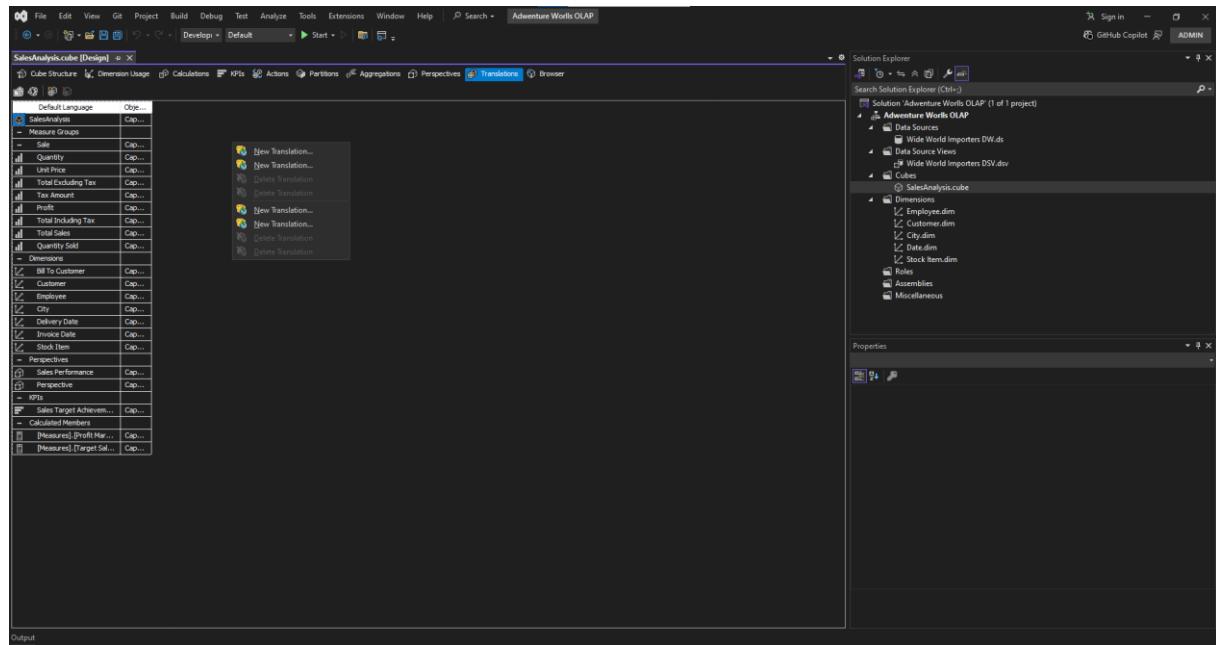


Widzimy, że mamy tylko wybrane miary i wymiary

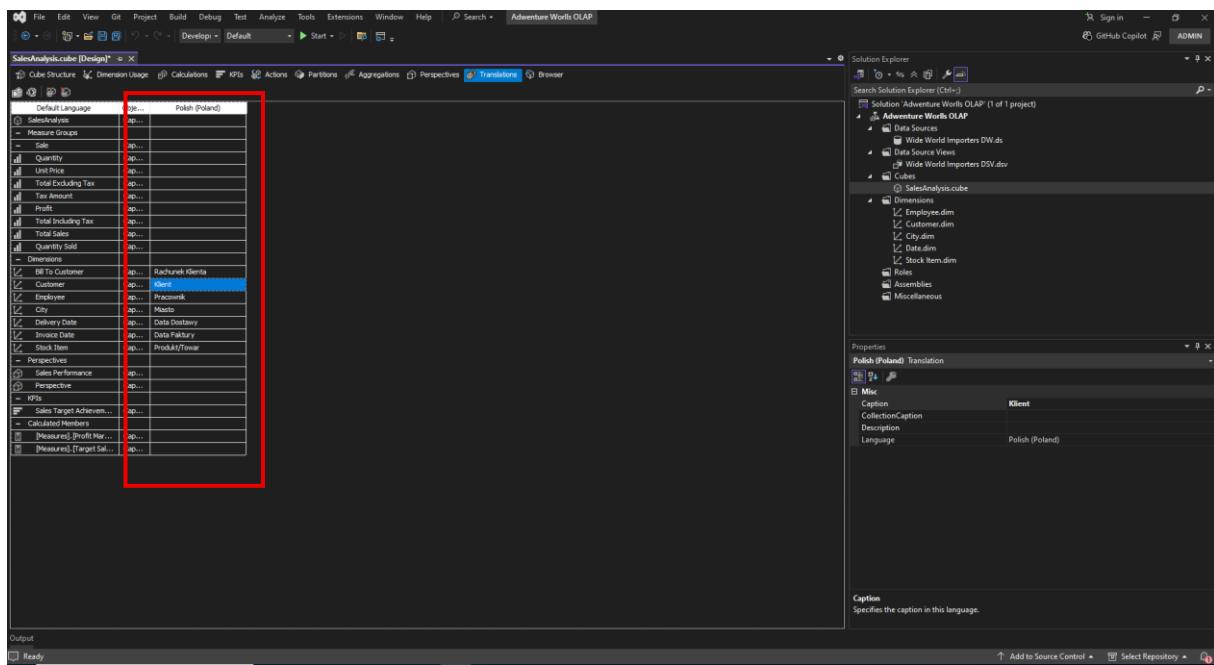
- Tłumaczenie wybranych wymiarów:



Otwieramy zakładkę *Translations*



Następnie *New Translation...*



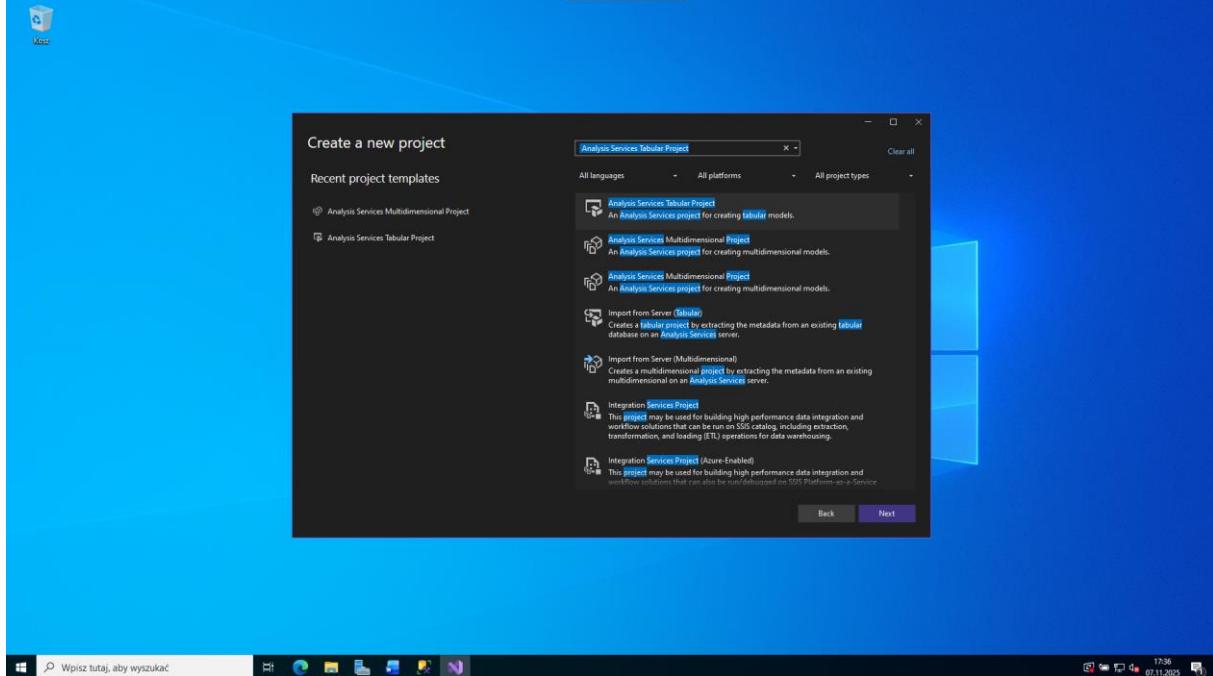
- Przykładowe zapytania MDX:

Lista klientów w której porównam ich zakupy w poszczególnych miesiącach z sąsiadującymi z sobą lat (2015 i 2016) z użyciem funkcji Time Intelligence, który porównuje wartości rok do roku (YoY) za pomocą ParallelPeriod.

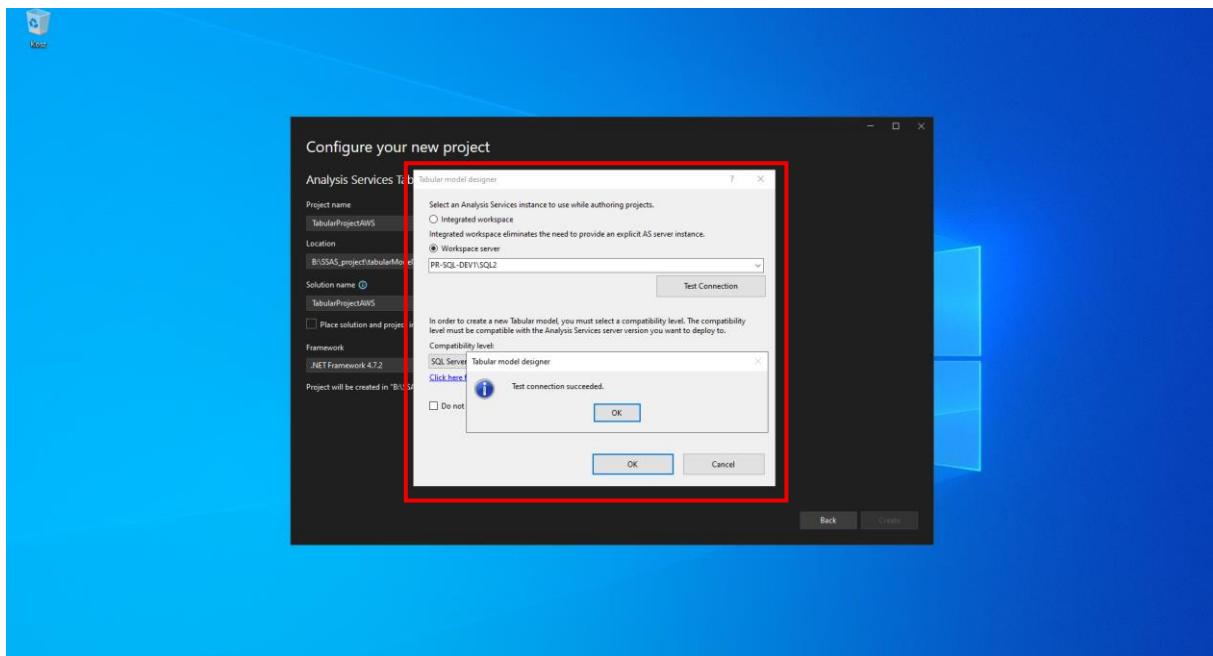
Wynik zapytania w języku MDX

7. Model tabelaryczny (Tabular)

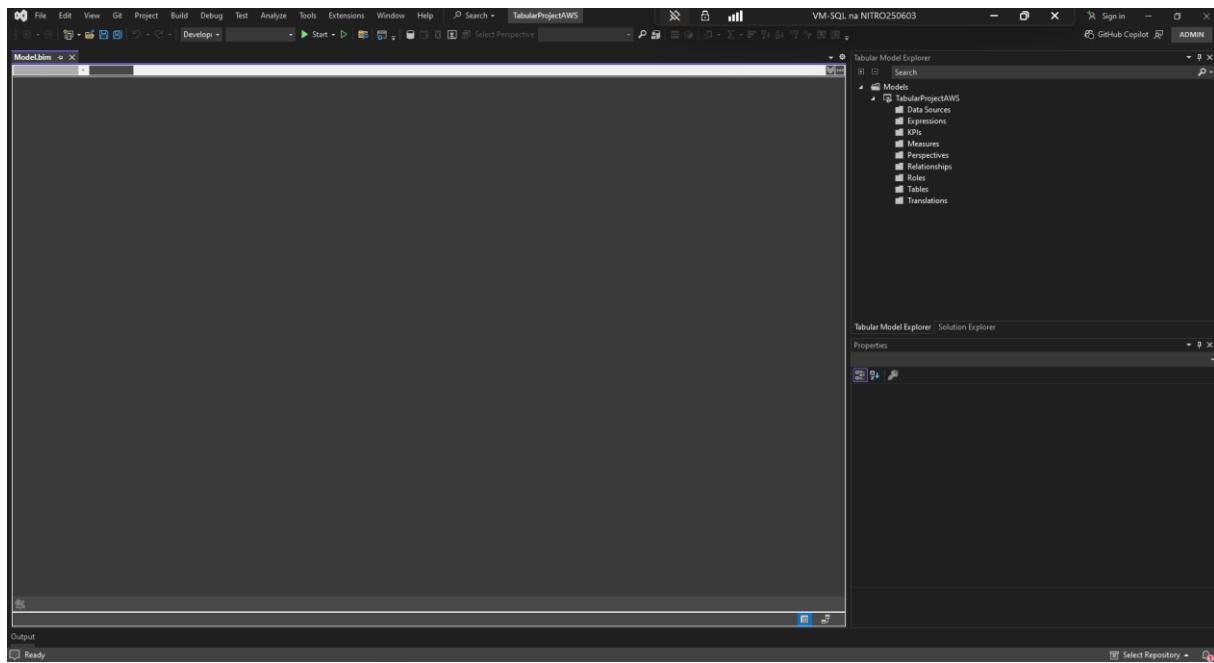
- Utworzenie projektu dla modelu tabelarycznego w Visual Studio:



Wybieramy sobie Analysis Services Tabular Project – oznacza to że będziemy tworzyć projekt opierający się na modelu tabelarycznym

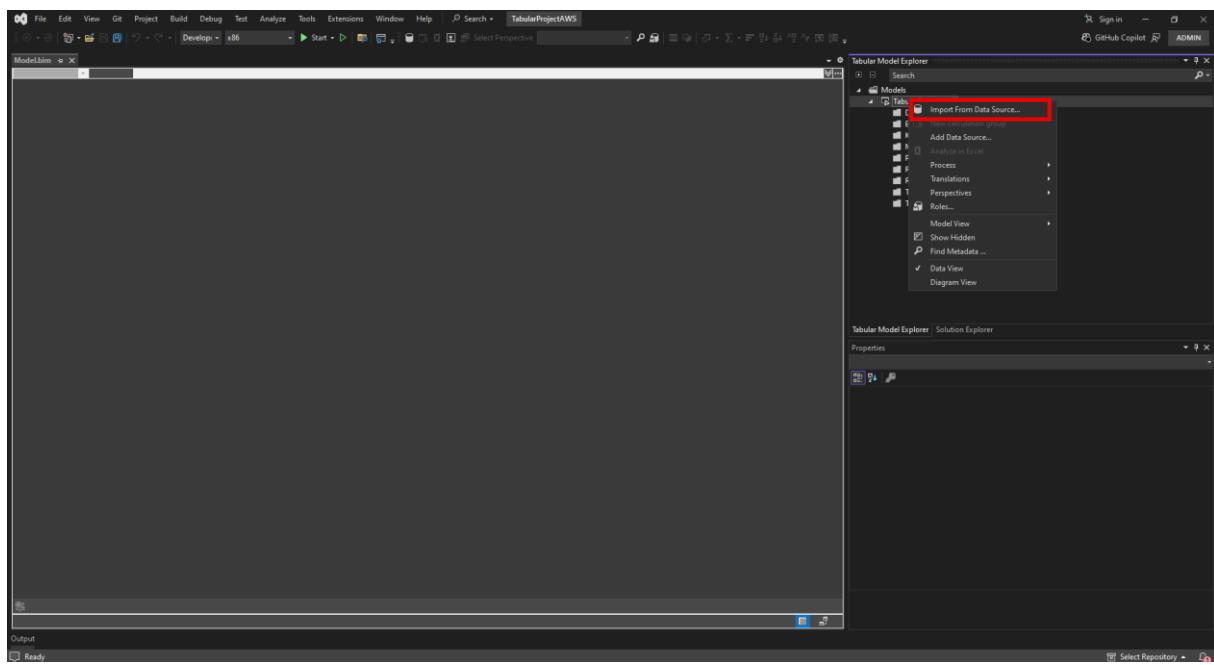


Łączymy się z instancją, na którą będziemy wdrażać nasz model

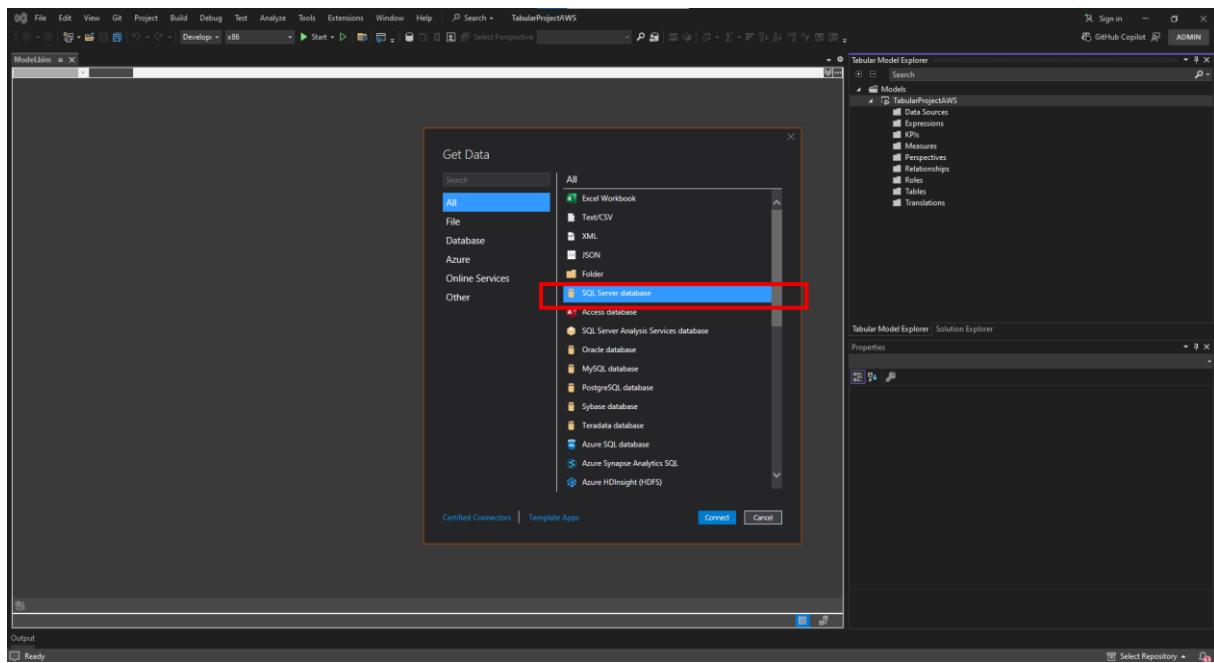


Projekt został utworzony

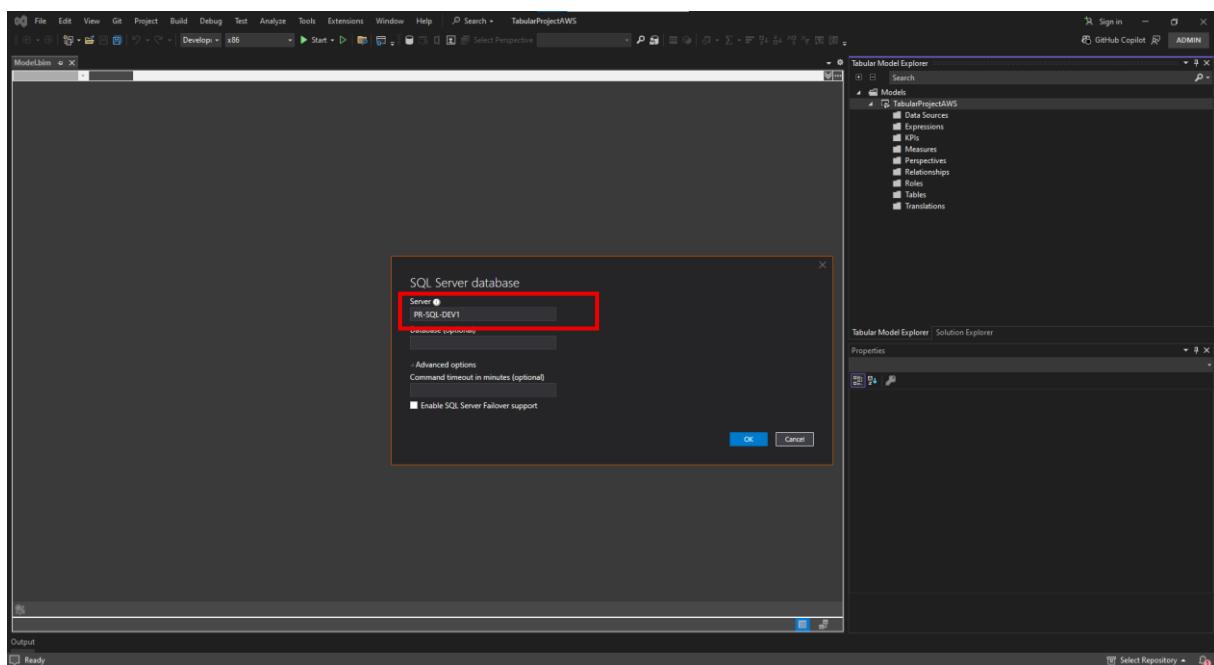
- Importowanie tabeli do modelu:



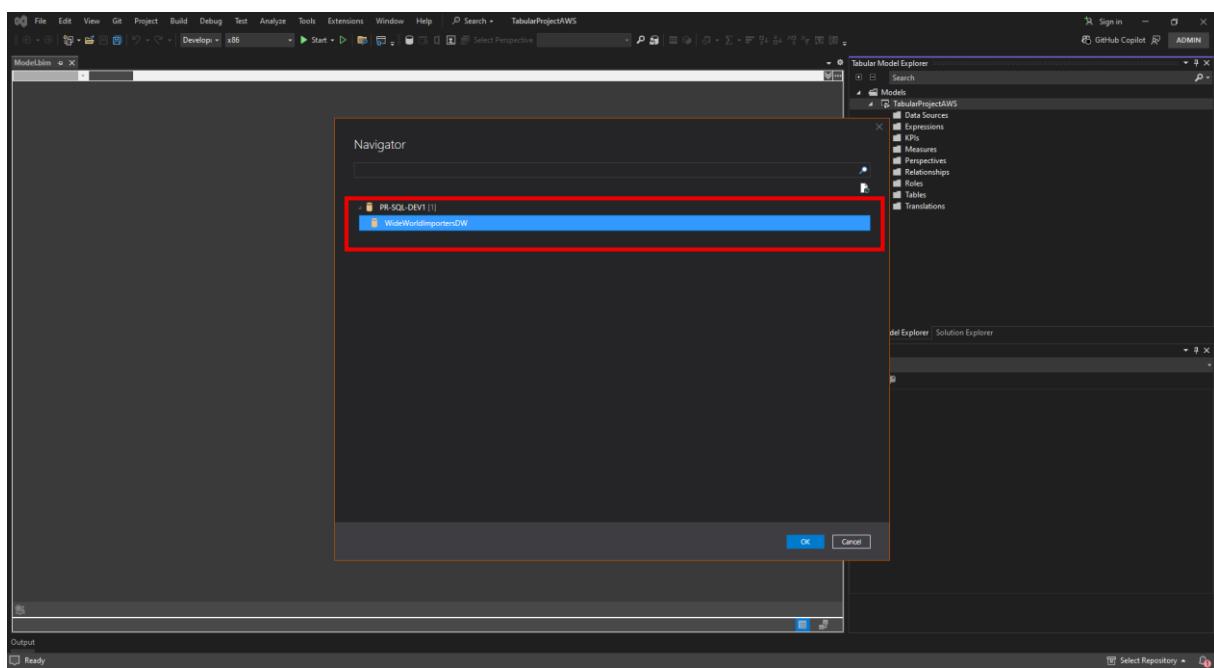
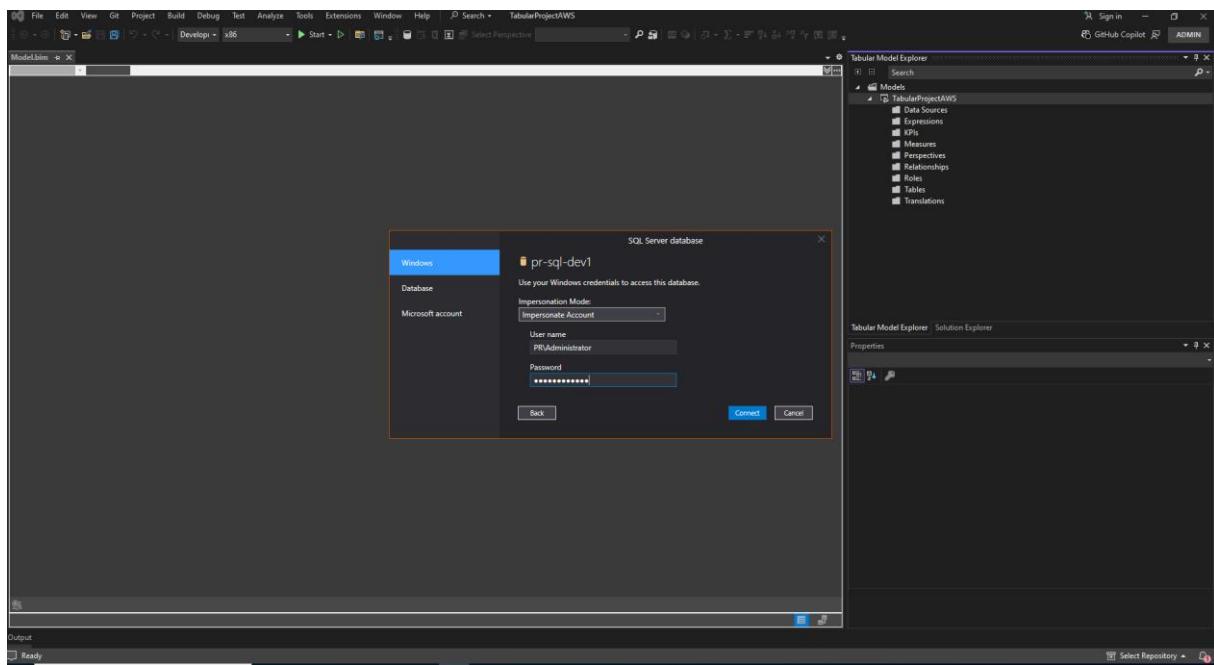
PPM na model i klikamy Import from Data Source



Sposób pobierania danych poprzez SQL Server database



Tabele będziemy pobierać z głównej instancji więc podajemy jej nazwę



Wybieramy źródło danych, z którego będziemy korzystać w projekcie

Navigator

Display Options

PR-SQL-DEV1\WideWorldImportersDW [10]

- Dimension.City
- Dimension.Customer
- Dimension.Date
- Dimension.Employee
- Dimension.Payment Method
- Dimension.Stock Item
- Dimension.Transaction
- Fact.Movement
- Fact.Order
- Fact.Purchase
- Fact.Sale
- Fact.Stock Holding
- Fact.Transaction
- Integration.City_Staging
- Integration.Customer_Staging
- Integration.Employee_Staging
- Integration.ETL_Cutoff
- Integration.Lineage
- Integration.Movement_Staging
- Integration.Order_Staging
- Integration.PaymentMethod_Staging
- Integration.Purchase_Staging
- Integration.Sale_Staging
- Integration.StockHolding_Staging
- Integration.StockItem_Staging
- Integration.Supplier_Staging
- Integration.Transaction_Staging
- Integration.TransactionType_Staging
- Integration.GenerateDateDimensionCal...

Dimensions Date

Date	Day Number	Day	Month	Short Month	Calendar Month Number	Calendar Month Label	Calendar Year	Calendar Year Label	Fiscal Month Number	Fiscal Month Label	Fiscal Year	Fiscal Year Label	ISO Week Number
01.01.2013	1	1	January	Jan	1	CY2013-Jan	2013	CY2013	1	Y2013-Jan	2013	Y2013	
02.01.2013	2	2	January	Jan	2	CY2013-Jan	2013	CY2013	2	Y2013-Jan	2013	Y2013	
03.01.2013	3	3	January	Jan	3	CY2013-Jan	2013	CY2013	3	Y2013-Jan	2013	Y2013	
04.01.2013	4	4	January	Jan	4	CY2013-Jan	2013	CY2013	4	Y2013-Jan	2013	Y2013	
05.01.2013	5	5	January	Jan	5	CY2013-Jan	2013	CY2013	5	Y2013-Jan	2013	Y2013	
06.01.2013	6	6	January	Jan	6	CY2013-Jan	2013	CY2013	6	Y2013-Jan	2013	Y2013	
07.01.2013	7	7	January	Jan	7	CY2013-Jan	2013	CY2013	7	Y2013-Jan	2013	Y2013	
08.01.2013	8	8	January	Jan	8	CY2013-Jan	2013	CY2013	8	Y2013-Jan	2013	Y2013	
09.01.2013	9	9	January	Jan	9	CY2013-Jan	2013	CY2013	9	Y2013-Jan	2013	Y2013	
10.01.2013	10	10	January	Jan	10	CY2013-Jan	2013	CY2013	10	Y2013-Jan	2013	Y2013	
11.01.2013	11	11	January	Jan	11	CY2013-Jan	2013	CY2013	11	Y2013-Jan	2013	Y2013	
12.01.2013	12	12	January	Jan	12	CY2013-Jan	2013	CY2013	12	Y2013-Jan	2013	Y2013	
13.01.2013	13	13	January	Jan	13	CY2013-Jan	2013	CY2013	13	Y2013-Jan	2013	Y2013	
14.01.2013	14	14	January	Jan	14	CY2013-Jan	2013	CY2013	14	Y2013-Jan	2013	Y2013	
15.01.2013	15	15	January	Jan	15	CY2013-Jan	2013	CY2013	15	Y2013-Jan	2013	Y2013	
16.01.2013	16	16	January	Jan	16	CY2013-Jan	2013	CY2013	16	Y2013-Jan	2013	Y2013	
17.01.2013	17	17	January	Jan	17	CY2013-Jan	2013	CY2013	17	Y2013-Jan	2013	Y2013	
18.01.2013	18	18	January	Jan	18	CY2013-Jan	2013	CY2013	18	Y2013-Jan	2013	Y2013	
19.01.2013	19	19	January	Jan	19	CY2013-Jan	2013	CY2013	19	Y2013-Jan	2013	Y2013	
20.01.2013	20	20	January	Jan	20	CY2013-Jan	2013	CY2013	20	Y2013-Jan	2013	Y2013	
21.01.2013	21	21	January	Jan	21	CY2013-Jan	2013	CY2013	21	Y2013-Jan	2013	Y2013	
22.01.2013	22	22	January	Jan	22	CY2013-Jan	2013	CY2013	22	Y2013-Jan	2013	Y2013	
23.01.2013	23	23	January	Jan	23	CY2013-Jan	2013	CY2013	23	Y2013-Jan	2013	Y2013	

The data in the preview has been truncated due to size limits.

Select Related Tables Load Transform Data Cancel

W dalszym ciągu będziemy się zajmować analizą sprzedaży więc wybieram następujące tabele: sprzedaż(tabela faktów), klient(wymiar), miasto(wymiar), czas(wymiar)

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search + TabularProjectAWS

ModelList * x86 Start Search Select Perspective GitHub Copilot ADMIN

Data Processing

Processing Progress

Processing gets updated data from the original data sources.

Success

5 Total 0 Cancelled
5 Success 0 Error

Details:

Work Item	Status	Message
Fact Sale	Success	228,253 rows transferred.
Dimension Customer	Success	403 rows transferred.
Dimension City	Success	16,295 rows transferred.
Dimension Stock Item	Success	6,772 rows transferred.
Dimension Date	Success	1,461 rows transferred.

Stop Processing Close

Output Select Repository

Solution Explorer Properties

Tabular Model Explorer

ModelList * x86

Fact Sale Dimension Customer Dimension City Dimension Stock Item Dimension Date

Record: 0 of 0

Poprawnie zimportowaliśmy tabelę

- W celach porządkowych zmienimy nazwy naszych tabel aby były bardziej przejrzyste (usuniemy przedrostki „dimension”, „fact” – zabieg generalnie nieobowiązkowy, aczkolwiek dla mojego oka potrzebny):

The screenshot shows the Microsoft SQL Server Data Tools (SSDT) interface. The main window displays a table named 'StockItem' with various columns. The 'Tables' section in the Solution Explorer is highlighted with a red box. The 'Properties' pane shows details for the 'Product' table, including its name and ID.

Nazwę tabeli Stock Item zmieniętem na Product, ponieważ jest dla mnie bardziej intuicyjna

- Eksploracja tabel:

Wykonuję przegląd wszystkich tabel, analizuję czy jest możliwość usunięcia zbędnych kolumn co pozwoliłoby nam stworzyć bardziej efektywny model, aczkolwiek jest to projekt dydaktyczny więc po przejrzeniu tych tabel uważam, że mogą zostać one w bazowej formie. Przy innych projektach uważam, że warto zwracać na to uwagę, ponieważ tabele często zawierają kolumny, których i tak nie wykorzystamy, a tylko zaśmiecają nam model.

- Tworzenie miar (jedna będzie wykorzystywała funkcję CALCULATE):

Plan działania przy tworzeniu miar:

1. Total Sales – miara będzie reprezentować łączną wartość sprzedaży brutto
2. Quantity Sold – miara będzie obliczać łączną liczbę sprzedanych jednostek towaru
3. Profit – będzie określać zysk ze sprzedaży
4. Sales 2015/2016 – będą filtrować wyniki dla 2015 i 2016 roku a następnie YoY Growth (Year over Year – rok do roku) czyli dynamikę sprzedaży eok do roku.

The screenshot shows the Microsoft Analysis Services Tabular Model Explorer interface. On the left, the 'Model Explorer' pane displays the model structure with nodes like Models, Measures, and Relationships. A specific measure, 'Total Sales', is highlighted with a red box in the 'Measures' section. In the center, the 'Data View' pane shows a table of sales data with columns: Salesperson Key, Sale ID, Customer, City, Product, Date, Description, Package, Quantity, Unit Price, Tax Rate, Total Excluding Tax, Tax Amount, Profit, Total Including Tax, Total Dry Items, Total Chiller Items, and Lineage Key. The bottom right corner of the Data View pane also has a red box around the 'Total Sales' measure definition. The 'Properties' pane on the right shows the measure's details: Name is 'Total Sales', Type is 'Measure', and Formula is 'SUM(Sale[Total Including Tax])'. The status bar at the bottom indicates '1 Total Sales'.

1 Total Sales

ModelView 4 X

Delivery Date = [Date] >= [Calc(Quantity)]

Quantity Sold Measure

Properties:

- Measure Name: Total Sales
- Format: General
- Formula: SUM([Sale].[Quantity])
- Measure Name: Quantity Sold

Tabular Model Explorer

Search: Quantity Sold

Relationships

Tables

Columns

Properties: Quantity Sold Measure

Detail Row Expression

Diagram Folder

Basic

Description

Format

Formula

Measure Name

Misc

Total Sales: 198043439,450001

Quantity Sold: 8950628

Output

Ready

Add to Source Control

Select Repository

2 Quantity Sold

ModelView 5 X

Delivery Date = [Date] >= [Calc(Profit)]

Total Profit Measure

Properties:

- Measure Name: Total Profit
- Format: General
- Formula: SUM([Sale].[Profit])
- Measure Name: Total Sales

Tabular Model Explorer

Search: Total Profit

Relationships

Tables

Columns

Properties: Total Profit Measure

Detail Row Expression

Diagram Folder

Basic

Description

Format

Formula

Measure Name

Misc

Total Sales: 198043439,450001

Total Profit: 8950628

Output

Ready

Add to Source Control

Select Repository

3 Total Profit

Sales 2015

The screenshot shows the Tabular Model Explorer interface. In the left pane, under the 'Measures' section, there is a node for 'Sales 2015'. A red box highlights this node. In the right pane, the 'Properties' window is open for 'Sales 2015 Measure'. It lists various properties such as Data Type (Measure), Measure Name (Sales 2015), and Description (Total Sales for Year 2015). Below the properties, there is a 'Calculated' section containing the DAX formula:

```
/> Sales 2015 =  
CALCULATE(  
    [Total Sales],  
    Date[Calendar Year] = 2015  
)
```

The main data grid displays sales data for 2015, with columns for Invoice ID, Employee ID, Package, Quantity, Unit Price, Tax Rate, Total Excl., Tax Amount, Profit, Total Including Tax, Total Dry Items, and Total Chiller Items. The total sales for 2015 are shown as 8529180.9.

Sales 2015

Sales 2016

The screenshot shows the Tabular Model Explorer interface. In the left pane, under the 'Measures' section, there is a node for 'Sales 2016'. A red box highlights this node. In the right pane, the 'Properties' window is open for 'Sales 2016 Measure'. It lists various properties such as Data Type (Measure), Measure Name (Sales 2016), and Description (Total Sales for Year 2016). Below the properties, there is a 'Calculated' section containing the DAX formula:

```
/> Sales 2016 =  
CALCULATE(  
    [Total Sales],  
    Date[Calendar Year] = 2016  
)
```

The main data grid displays sales data for 2016, with columns for Invoice ID, Employee ID, Package, Quantity, Unit Price, Tax Rate, Total Excl., Tax Amount, Profit, Total Including Tax, Total Dry Items, and Total Chiller Items. The total sales for 2016 are shown as 8529180.9.

Sales 2016

The screenshot shows the Microsoft Power BI Tabular Model Explorer interface. A red box highlights the 'Measures' section in the Properties pane, which contains measures like Sales, Sales 2015, Sales 2016, Total Profit, Total Sales, and YoY Growth. Another red box highlights the 'Sales' table in the Relationships pane. The main area displays a PivotTable with columns for Invoice ID, Description, Pedage, Quantity, Unit Price, Tax Rate, Total Excluding Tax, Tax Amount, Total Including Tax, Total Dry Items, and Total Chiller Items. The PivotTable shows data for various invoices, with totals at the bottom. The Properties pane also shows the 'YoY Growth' measure selected.

YoY – można zauważyc, że zanotowano znaczący spadek (więc nazwa miary YoY Growth w tym wypadku może znacząco drażnić oko)

- Sprawdzenie działania (Analyze in Excel):

The screenshot shows Microsoft Excel with a PivotTable setup. The PivotTable Fields pane on the right shows fields grouped by Region, Sales Territory, State Province, Subregion, Valid From, Valid To, and WWI City ID. It also shows Customer, Bill To Customer, Buying Group, and Category. The PivotTable itself has Row Labels set to 'Customer' and Columns set to 'Sales 2015' and 'Sales 2016'. The Values field is set to 'YoY Growth'. The PivotTable displays sales data for various customers across different regions and years, with the YoY Growth value calculated for each customer.

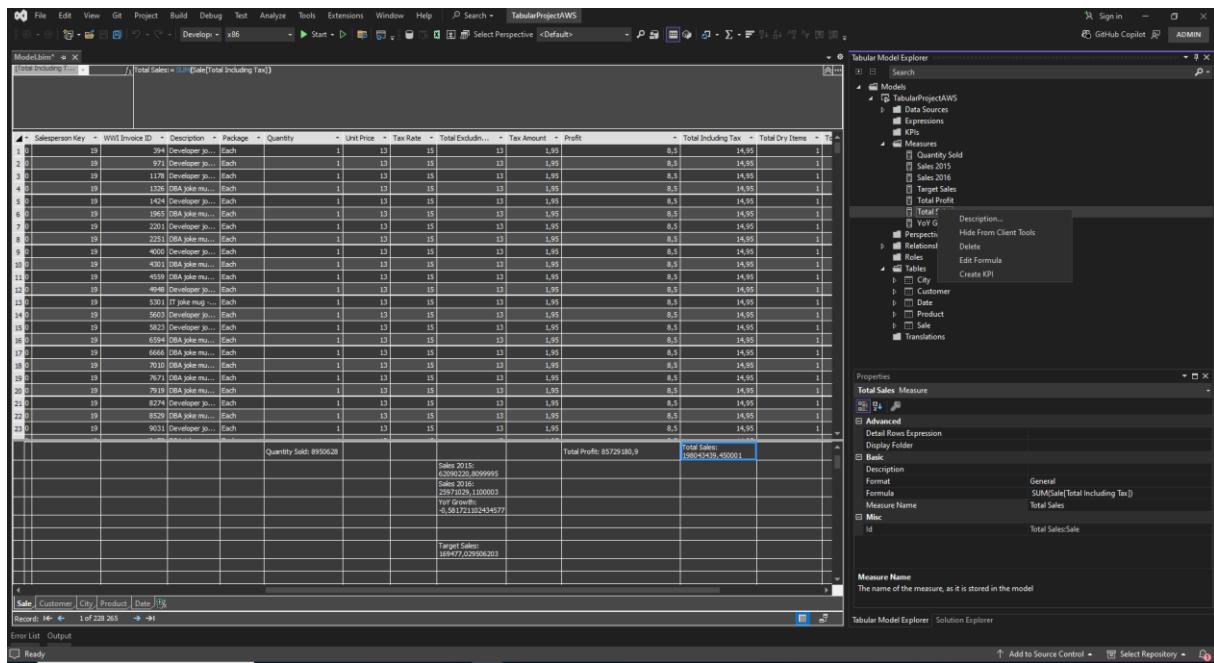
Row Labels	Sales 2014	Sales 2015	YoY Growth
1 Talispin Toys (Absecon, NJ)	76442.69	33500.11	-56.18%
2 Talispin Toys (Acelitas, PR)	75569.12	26451.32	-64.99%
4 Talispin Toys (Airport Drive, MO)	102791.16	44326.00	-56.88%
5 Talispin Toys (Alstead, NH)	65235.35	46753.31	-28.35%
6 Talispin Toys (Anchorage, AK)	39732.97	30201.00	-24.94%
7 Talispin Toys (Andrus, CO)	94015.64	30021.99	-69.37%
8 Talispin Toys (Annamariah, WV)	101240.55	24200.02	-76.10%
9 Talispin Toys (Antares, AZ)	95186.91	4096.9	-57.07%
10 Talispin Toys (Antonito, CO)	104800.09	40105.37	-61.73%
11 Talispin Toys (Arbor Vista, WI)	120603.84	3333.9	-72.36%
12 Talispin Toys (Arietta, NY)	88846.94	100315.11	12.91%
13 Talispin Toys (Armstrong Creek, WI)	72740.44	44726.02	-38.51%
14 Talispin Toys (Arrow Rock, MO)	61229.69	54788.0	10.52%
15 Talispin Toys (Astrabula, OH)	72783.5	36185.79	-50.28%
16 Talispin Toys (Avon Park, FL)	89220.47	35500.0	-59.54%
17 Talispin Toys (Avon Park, FL)	102799.94	24500.0	-76.77%
18 Talispin Toys (Avenal, CA)	95732.14	32134.09	63.72%
19 Talispin Toys (Baraboo, WI)	91109.25	18012.74	-80.23%
20 Talispin Toys (Batsom, TX)	93211.48	1772.0	-80.99%
21 Talispin Toys (Belgreen, AL)	88766.84	59430.01	-33.05%
22 Talispin Toys (Benni, MO)	98025.5	20177.11	-79.42%
23 Talispin Toys (Bethania, NC)	85999.68	50942.4	-40.76%
24 Talispin Toys (Big Moose, NY)	124762.07	56927.4	-54.37%
25 Talispin Toys (Biscay, MN)	88195.92	33190.01	-62.37%
26 Talispin Toys (Bow Mar, CO)	77596.17	47416.0	-38.89%
27 Talispin Toys (Boyertown, PA)	148890.0	40105.37	-74.47%
28 Talispin Toys (Boyle Arbor, SC)	114022.44	46162.07	-59.51%
29 Talispin Toys (Bratenahl, OH)	104641.27	34801.00	-66.74%
30 Talispin Toys (Brown City, MI)	115097.72	42956.04	-62.68%
31 Talispin Toys (Buell, MO)	141313.85	42597.72	-69.86%
32 Talispin Toys (Carbonville, UT)	95508.36	4175.9	-96.28%
33 Talispin Toys (Casperon, NV)	90739.9	56009.7	-38.28%
34 Talispin Toys (Cherry Grove Beach, SC)	87590.27	15997.7	-81.74%
35 Talispin Toys (Cheyenne Wells, CO)	107278.91	4223.1	-60.63%
36 Talispin Toys (Clewlston, FL)	139182.56	5323.0	-95.97%
37 Talispin Toys (Cokato, MN)	86725.28	10502.0	-88.16%

YoY – można ustawić w formacie aby dane były przedstawiane w procentach

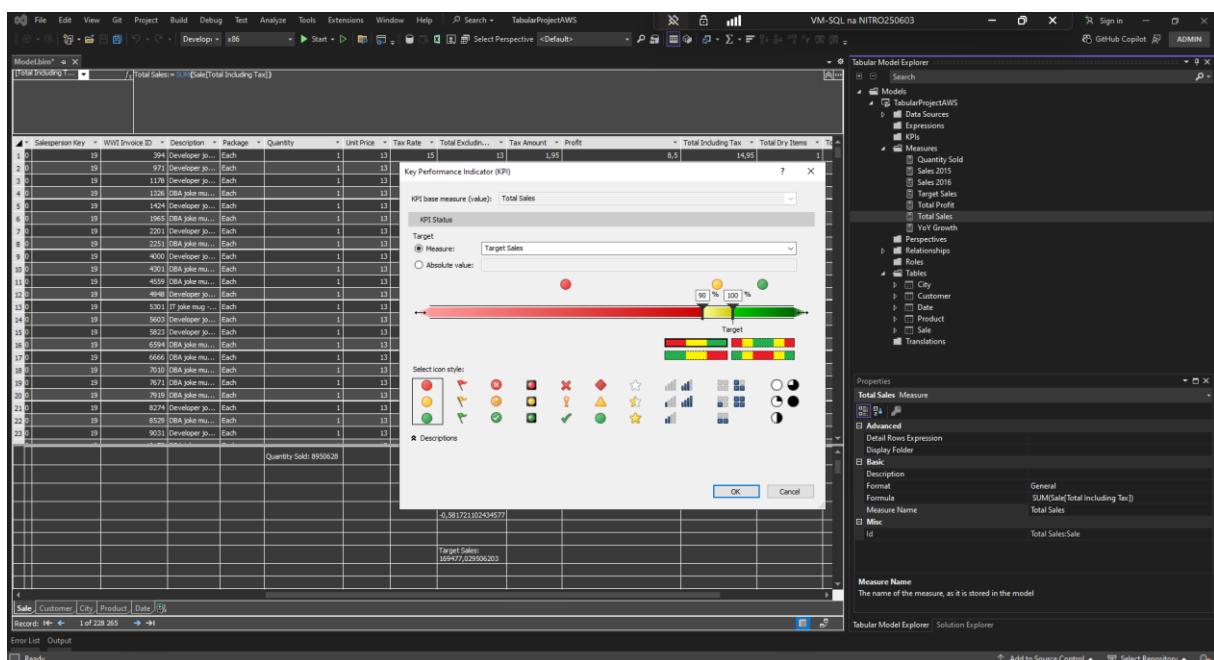
- Utworzenie KPI (w modelu tabelarycznym KPI opiera się na dwóch miarach tj. wartość główna u nas Total Sales oraz na celu u nas będzie to Target sales (musimy jeszcze utworzyć tą miarę) i cel ustawię na 110% średniej sprzedaży z poprzedniego roku)

The screenshot shows the SAP BusinessObjects Data Studio interface. On the left, there's a 'Model Explorer' pane with a tree view of models, tables, and measures. In the center, there's a grid view of sales data with columns like Salesperson Key, AWS Invoice ID, Description, Package, Quantity, Unit Price, Tax Rate, Total Exclu..., Tax Amount, Profit, Total Including Tax, Total Dry Item, and TS. A specific row is selected, showing details for a developer job. On the right, there's a 'Properties' pane where the 'Target Sales' measure is being edited. The 'General' tab shows the formula as 'AVERAGE([Customer].[Customer].[Sales 2015])'. Other tabs include 'Advanced', 'Detail Rows Expression', 'Detail Rows Folder', 'Basic', 'Description', 'Format', 'Formula', 'Measure Name', and 'Misc'. The 'Measure Name' section is expanded, showing the name 'Target Sales' and a note: 'The name of the measure, as it is stored in the model'.

Utworzenie miary Target Sales jako sprzedaż z 2015 razy 110% więc będziemy oczekiwac wzrostu sprzedaży o 10%



Tworzenie KPI



Jako bazę dajemy *Total Sales* czyli sprzedaż rzeczywistą, jako cel ustawimy naszą miarę *Target Sales* i zmieniamy proggi na takie jakie są widoczne u góry(czerwony – poniżej celu, żółty – pomiędzy, zielony – powyżej celu)

Row Labels	Total Sales	Total Sales Goal	Total Sales Status
1 TailsUp Toys (Absecon, NJ)	29969.37	84086.959	Green
2 TailsUp Toys (Acelmanas, PR)	273285.36	83116.132	Green
3 TailsUp Toys (Airport Drive, MO)	374738.45	113970.276	Green
4 TailsUp Toys (Albuquerque, NM)	234899.87	73040.185	Green
5 TailsUp Toys (Albuquerque, NM)	285213.52	107941.297	Green
6 TailsUp Toys (Amundale Park, WA)	302541.62	103745.004	Green
7 TailsUp Toys (Andris, CO)	332798.32	111964.605	Green
8 TailsUp Toys (Ann Arbor, MI)	324134.4	104705.601	Green
9 TailsUp Toys (Antonito, CO)	263486.96	115280.099	Green
10 TailsUp Toys (Arbor Vitae, WI)	306612.06	132664.224	Green
11 TailsUp Toys (Arietta, NY)	331871.87	97731.634	Green
12 TailsUp Toys (Armstrong Creek, WI)	342817.06	80914.484	Green
13 TailsUp Toys (Arrow Rock, MO)	263383.03	67352.659	Green
14 TailsUp Toys (Ashland, OH)	322924.4	102204.138	Green
15 TailsUp Toys (Aspen Park, CO)	301951.74	94846.007	Green
16 TailsUp Toys (Astoria Park, FL)	285575.35	113079.934	Green
17 TailsUp Toys (Avenal, CA)	316480.92	103105.354	Green
18 TailsUp Toys (Baraboo, WI)	268137.35	102210.275	Green
19 TailsUp Toys (Bartons, TX)	340960.65	102332.628	Green
20 TailsUp Toys (Belgreen, AL)	343597.84	97643.524	Green
21 TailsUp Toys (Bennet, MO)	283077.31	107828.05	Green
22 TailsUp Toys (Bethania, NC)	369409.86	94589.748	Green
23 TailsUp Toys (Big Moose, NY)	384621.35	102204.138	Green
24 TailsUp Toys (Biscayne, NY)	309772.97	97915.512	Green
25 TailsUp Toys (Biscayne, NY)	265065.40	85349.197	Green
26 TailsUp Toys (Bowles, MN)	352845.77	154780.264	Green
27 TailsUp Toys (Boyleton Arbor, SC)	326197.71	125424.684	Green
28 TailsUp Toys (Bratenahl, OH)	274933.76	115108.697	Green
29 TailsUp Toys (Brown City, MI)	368897.01	126607.492	Green
30 TailsUp Toys (Buell, MO)	388351.75	155445.235	Green
31 TailsUp Toys (Carbonville, UT)	334355.34	105059.196	Green
32 TailsUp Toys (Castleton, NV)	286124.72	99813.89	Green
33 TailsUp Toys (Cherry Grove Beach, SC)	279690.2	96349.297	Green
34 TailsUp Toys (Cheyenne Wells, CO)	376640.75	118005.301	Green
35 TailsUp Toys (Chestnut Hill, PA)	298101.4	145400.816	Green
36 TailsUp Toys (Cokato, MN)	276540.31	97597.868	Green

Jak widać nasz cel na razie jest osiągany

• Perspektywy:

- Sales Performance – będzie zawierać wyłącznie dane dotyczące klientów, produktów, kalendarza oraz miary sprzedażowe i KPI, dzięki czemu użytkownik widzi tylko dane potrzebne do oceny wyników sprzedaży.
- Product Analytics - zostanie utworzona dla analiz produktów i ich rotacji. Zawiera uproszczony zestaw danych: informacje o produktach, sprzedaży i czasie, co umożliwia analizę popularności produktów i sezonowości bez nadmiaru informacji.

Fields	Sales Performance	Product Analytics
Tables	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ Ctry	<input type="checkbox"/>	<input type="checkbox"/>
- Customer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
- Bill To Customer	<input type="checkbox"/>	<input type="checkbox"/>
Buying Group	<input type="checkbox"/>	<input type="checkbox"/>
Category	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Customer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customer Key	<input type="checkbox"/>	<input type="checkbox"/>
Lineage Key	<input type="checkbox"/>	<input type="checkbox"/>
Postal Code	<input type="checkbox"/>	<input type="checkbox"/>
Primary Contact	<input type="checkbox"/>	<input type="checkbox"/>
Valid From	<input type="checkbox"/>	<input type="checkbox"/>
Valid To	<input type="checkbox"/>	<input type="checkbox"/>
WVVI Customer ID	<input type="checkbox"/>	<input type="checkbox"/>
Date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Calendar Month Label	<input type="checkbox"/>	<input type="checkbox"/>
Calendar Month Number	<input type="checkbox"/>	<input type="checkbox"/>
Calendar Year	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Calendar Year Label	<input type="checkbox"/>	<input type="checkbox"/>
Date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Day	<input type="checkbox"/>	<input type="checkbox"/>
Day Number	<input type="checkbox"/>	<input type="checkbox"/>
Fiscal Month Label	<input type="checkbox"/>	<input type="checkbox"/>
Fiscal Month Number	<input type="checkbox"/>	<input type="checkbox"/>
Fiscal Year	<input type="checkbox"/>	<input type="checkbox"/>
Fiscal Year Label	<input type="checkbox"/>	<input type="checkbox"/>
ISO Week Number	<input type="checkbox"/>	<input type="checkbox"/>
Month	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Short Month	<input type="checkbox"/>	<input type="checkbox"/>

Wybór potrzebnych tabel i kolumn

Wybór potrzebnych tabel i kolumn

Wybór potrzebnych tabel i kolumn

- Podsumowanie:

W projekcie utworzyliśmy dwie perspektywy, ponieważ różne osoby korzystające z modelu potrzebują innych informacji i innego poziomu szczegółowości. Pierwsza perspektywa, Sales Performance, ma być używana głównie przez osoby analizujące wyniki sprzedażowe, więc skupia się przede wszystkim na tym, ile sprzedano, komu i jakie były tego efekty finansowe. Z tego powodu w tej perspektywie znalazła się tabela Customer, z której wybraliśmy kolumny Category i Customer, ponieważ one pozwalają pogrupować i rozpoznać klientów według typu odbiorcy i nazwy. Dodaliśmy także tabelę Date i kolumny Calendar Year, Date oraz Month, ponieważ analiza sprzedaży zawsze odbywa się w czasie i chcemy móc np. porównywać wyniki między miesiącami i latami. W tabeli Product zostawiliśmy tylko kolumnę Stock Item, ponieważ w tym kontekście wystarczy sama informacja, jaki produkt został sprzedany, bez dodatkowych szczegółów. Najważniejsza jest tutaj tabela Sale, w której umieściliśmy kolumny dotyczące miar sprzedażowych: Profit, Quantity Sold, Sales 2015, Sales 2016, YoY Growth oraz Total Sales. Te pola pozwalają ocenić, jak zmienia się sprzedaż, jaki jest wynik finansowy oraz czy występuje wzrost lub spadek pomiędzy latami.

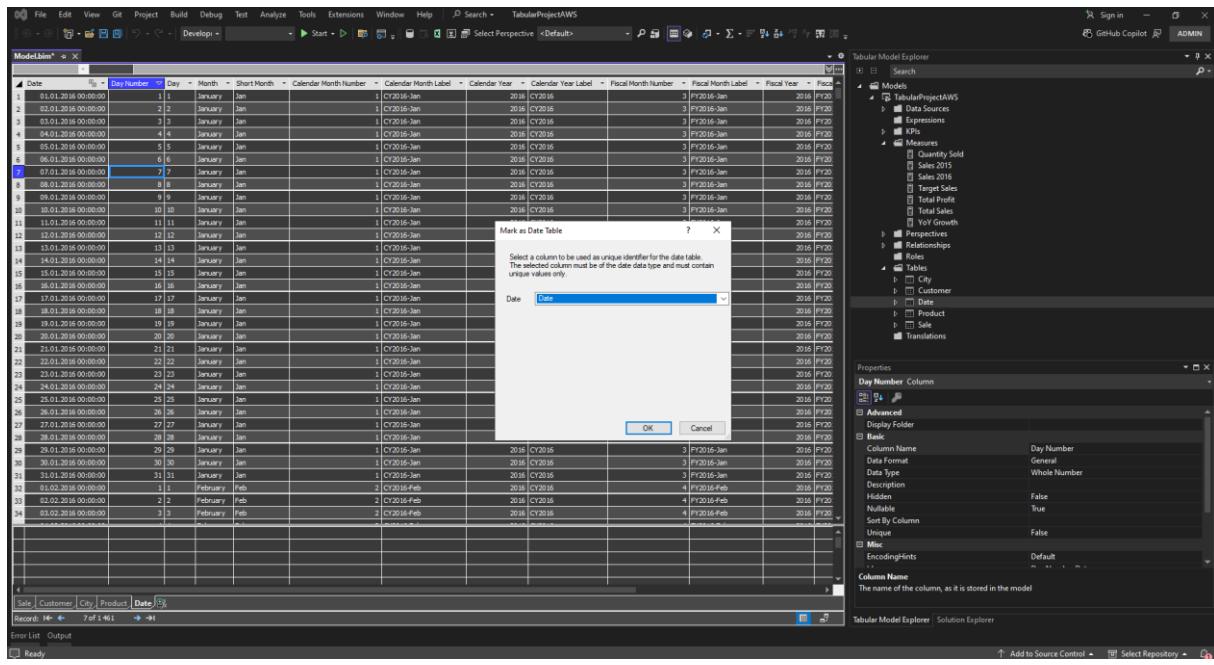
Druga perspektywa, Product Analysis, jest bardziej szczegółowa i stworzona z myślą o analizie asortymentu, czyli patrzeniu na produkty od strony cech, parametrów i zachowania popytu.

Tutaj w tabeli Sale wybraliśmy kolumny Total Sales, Unit Price i Quantity, ponieważ pozwalają one sprawdzić, ile dana pozycja generuje przychodu, w jakiej cenie faktycznie była sprzedawana oraz w jakiej ilości. W tej perspektywie uwzględniliśmy całą tabelę Product, żeby mieć dostęp do wszystkich cech produktu, takich jak marka, kolor, rozmiar czy typ opakowania – to jest kluczowe, jeśli chcemy analizować np. które warianty produktu są bardziej popularne. Podobnie jak wcześniej, tabela Date z kolumnami Calendar Year, Date i Month umożliwia analizę sezonowości oraz trendów sprzedaży w czasie. Dołożyliśmy też tabelę Customer, ale tym razem tylko z kolumną Customer, ponieważ w tej perspektywie nie chodzi o pełną segmentację odbiorców, tylko o możliwość sprawdzenia, kto kupuje określone produkty.

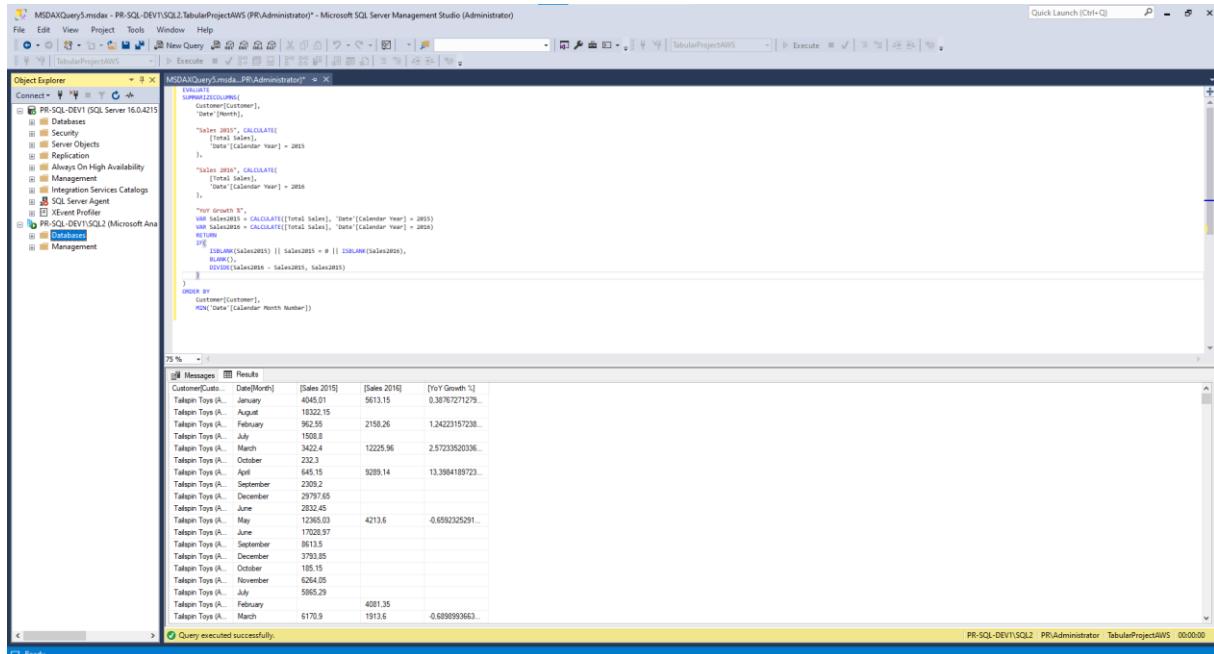
Podsumowując, Sales Performance jest perspektywą bardziej „sprzedażową” i ogólną, skupioną na wynikach finansowych i czasie, natomiast Product Analysis jest bardziej „produktowa” i szczegółowa, pozwalająca analizować asortyment i zachowania rynku na poziomie konkretnych cech produktu. Dzięki temu model jest czytelny i dopasowany do różnych ról użytkowników.

The screenshot shows the Microsoft Tabular Model Explorer interface. On the left, there is a large grid of data for the 'Sales' table, which includes columns for Sale, Customer, City, Product, and Date. The data consists of approximately 100 rows of sales records. On the right side, the 'Properties' pane is open for the 'City' table, showing various configuration options like 'Hidden' (set to False), 'Default Detail Rows Expression' (Click to edit), and 'Table Description'. Above the properties pane, the 'Sales Performance' table is highlighted with a red box in the 'Tables' section of the 'Model' tree. The 'Model' tree also lists other tables like 'Customer', 'Date', 'Product', and 'Table' under the 'Tables' category. The top navigation bar shows 'File', 'Edit', 'View', 'Git', 'Project', 'Build', 'Debug', 'Test', 'Analyze', 'Tools', 'Extensions', 'Window', 'Help', and 'Search'. The status bar at the bottom indicates 'Record: 16 - 1 of 138 295' and 'Ready'.

- Odpytanie modelu tabelarycznego za pomocą DAX (cel zapytania taki sam jak w modelu wielowymiarowym):



W tabeli Date ustawimy **Mark As Date Table** aby dokładnie jedna tabela reprezentowała kalendarz w modelu, dzięki temu wymuszę istnienie unikalnej kolumny dat bez duplikatów



W zapytaniu DAX najpierw pobierana jest lista klientów oraz ich sprzedaż z dwóch kolejnych lat: 2015 i 2016. Odbywa się to dzięki użyciu miar Sales 2015 i Sales 2016, które z kolei filtryują dane tak, aby policzyć sprzedaż tylko dla konkretnego roku. SUMMARIZECOLUMNS tworzy z tego tabelę wynikową, gdzie każdy wiersz odpowiada jednemu klientowi i jednemu miesiącowi.

Następnie tworzona jest miara porównująca sprzedaż rok do roku. Różnicę pomiędzy sprzedażą w 2016 i 2015 dzielimy przez sprzedaż z roku 2015. Do dzielenia wykorzystujemy funkcję DIVIDE, ponieważ jest bezpieczna jeśli sprzedaż z 2015 roku wynosi zero, to nie wyrzuci błędu (tak jak zwykłe dzielenie), tylko zwróci BLANK. Dzięki temu wynik jest czytelny i pozbawiony błędów.

W tym porównaniu używamy logiki Time Intelligence, czyli wbudowanej w DAX obsługi czasu. Działa ona poprawnie dzięki temu, że tabela z datami została oznaczona jako Mark as Date Table, co pozwala DAX zrozumieć kolejność miesięcy i lat. Dzięki temu funkcje mogą automatycznie porównać sprzedaż miesiąc do miesiąca w różnych latach, bez konieczności ręcznego pisania warunków i filtrów.

W efekcie otrzymujemy prostą tabelę, która pokazuje, jak zmieniała się sprzedaż tego samego klienta w tych samych miesiącach między 2015 a 2016 rokiem oraz o ile procent wzrosła lub spadła.