

**Zintegrowany Program Rozwoju  
Akademii Górniczo-Hutniczej w Krakowie**  
Nr umowy: **POWR.03.05.00-00-Z307/17**

**Instrukcja do ćwiczeń laboratoryjnych**

<b>Nazwa przedmiotu</b>	<b>Eksploracja danych</b>
<b>Temat ćwiczenia</b>	<b>Hurtownie danych</b>

Poziom studiów	<i>II stopień</i>
Kierunek	Informatyka
Forma i tryb studiów	Studia stacjonarne
Semestr	semestr letni roku akademickiego 2023/2024

mgr inż. Marek Macura



Wydział Informatyki, Elektroniki i Telekomunikacji

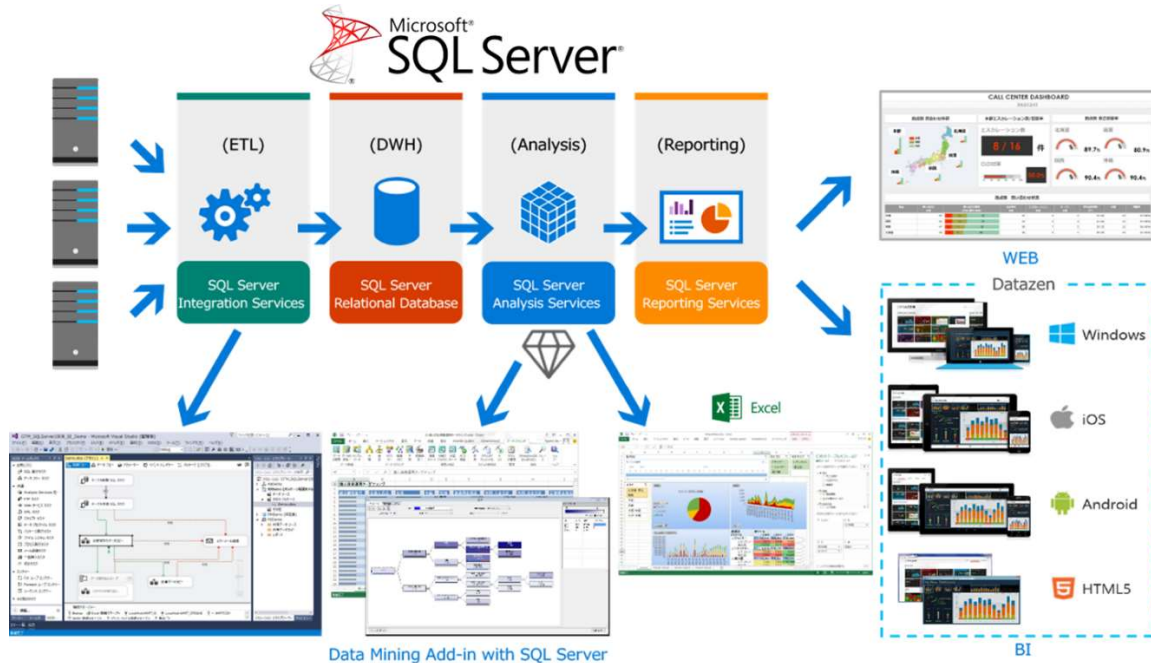
Kraków, 2024

## Cel ćwiczenia

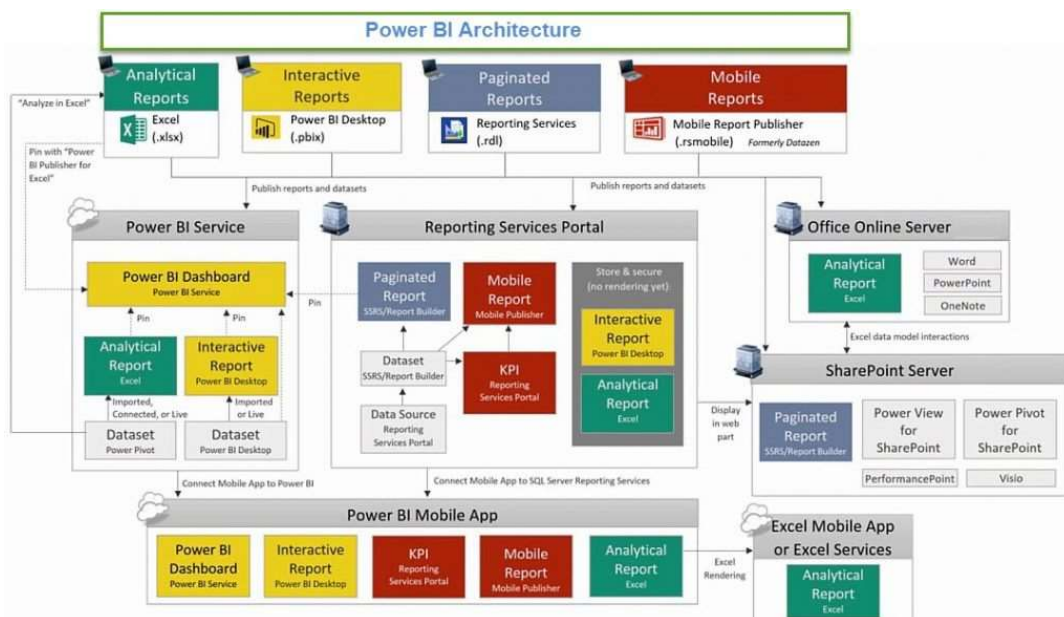
Zapoznanie się z zapytaniami analitycznymi, budową struktur hurtowni danych, analizą danych z wykorzystaniem języka MDX oraz raportowaniem w MS SQL Server 2017 i MS Power BI.

## Wprowadzenie do ćwiczenia

Wymaganymi wiadomościami wstępnymi z wykładu są: rozszerzenia SQL wspierające analizy, podstawowe pojęcia modelu wielowymiarowego (fakty, miary, wymiary i ich hierarchie, kostki), modelowanie ROLAP (schemat gwiazdy, płatka śniegu), język MDX. Dodatkowo wymagana jest znajomość modelowania danych według modelu relacyjnego. Pakiet SQL Server 2017 zawiera pełne zestawienie usług analityki biznesowej (business intelligence) umożliwiając realizację procesu okrywania wiedzy z danych w zakresie/na potrzeby biznesowe.



W ramach ćwiczenia pokazane zostanie, jak zbudować struktury hurtowni danych oraz kostki analityczne, które stanowią bazę pod raportowanie i inne metody analityczne. Instrukcje zawierają w tym zakresie także zadania do samodzielnego wykonania. Podobnie przedstawiony został obszar konsumencki odnosząc się do języka dostępowego MDX (Multidimensional Expressions) oraz oprogramowania Power BI typu self-service z następującą architekturą:



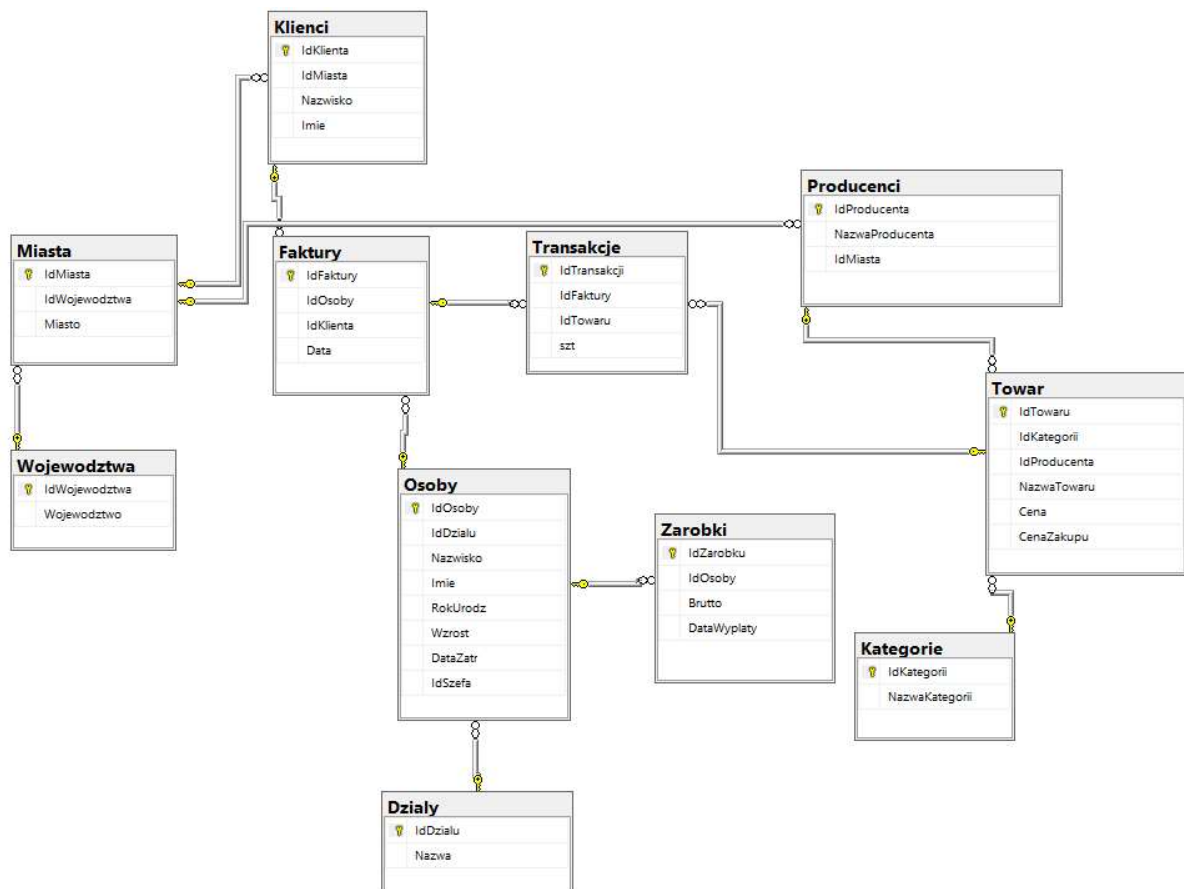
## Plan/ program ćwiczenia

### 1. Zapoznanie się z treningową bazą danych, która opisuje działalność firmy handlowej.

1.1. Uruchom Microsoft SQL Server Management Studio 18 i połącz się z lokalnym silnikiem bazy danych.

1.2. Wybierz bazę „BazaRelacyjna” i zapoznaj się z diagramem *DIAGRAM1* tej bazy.

Firma sprzedaje towary, które są sklasyfikowane według kategorii. Towary kupują klienci mieszkający w miejscowościach. Przy każdym zakupie wystawiana jest faktura, na której w przypadku zakupu pewnej ilości jednego typu towaru definiuje się transakcję. Na fakturze może pojawić się wiele transakcji. Każda faktura jest wystawiana przez pracownika firmy, przypisanego do konkretnego działu. Każdy z pracowników otrzymuje wiele różnego rodzaju wypłat.



### 2. Zapytania analityczne

2.1. Zapytanie wyznaczające sumy wypłat brutto dla każdego z pracowników firmy, wyświetlając obok nazwę działu ma następującą postać:

```
SELECT Nazwa, Nazwisko, SUM(Brutto)
FROM Działy JOIN Osoby
ON Działy.IdDzialu = Osoby.IdDzialu
JOIN Zarobki
ON Osoby.IdOsoby = Zarobki.IdOsoby
GROUP BY Nazwa, Nazwisko, Osoby.IdOsoby;
```

Poziom obliczania sumy jest wyznaczony przez najmniejszy poziom w hierarchii zawarty w klauzuli *GROUP BY*.

- 2.2. Jeżeli po poleceniu *SELECT* ograniczymy się tylko do pola Nazwa to możemy w *GROUP BY* także ograniczyć się do tego samego pola. Podsumowanie będzie wykonywane dla każdego działu – wszystkich jego pracowników. Zmiana opcji grupowania pociąga za sobą zmianę poziomu wyznaczania funkcji agregujących. Nie możliwe jest wyznaczenie podsumowania na dwóch poziomach za pomocą pojedynczego zapytania wybierającego, używającego tej klauzuli. Rozwiązaniem jest zastosowanie podzapytania wyznaczającego podsumowanie na poziomie działów i połączenie go (operator *JOIN*) z zapytaniem głównym, które oblicza sumę dla każdego z pracowników:

```
SELECT Nazwa, Nazwisko, SUM(Brutto) AS Razem, SD
```

```
FROM Działy JOIN Osoby
```

```
ON Działy.IdDziału = Osoby.IdDziału
```

```
JOIN Zarobki
```

```
ON Osoby.IdOsoby = Zarobki.IdOsoby
```

```
JOIN
```

```
(SELECT IdDziału, SUM(Brutto) AS SD
```

```
FROM Osoby JOIN Zarobki
```

```
ON Osoby.IdOsoby = Zarobki.IdOsoby
```

```
GROUP BY IdDziału) AS xxx
```

```
ON Działy.IdDziału=xxx.IdDziału
```

```
GROUP BY Nazwa, Nazwisko, Osoby.IdOsoby, SD;
```

Przez analogię dla trzech poziomów konieczne jest zastosowanie poza zapytaniem głównym dwóch podzapytań, a dla czterech poziomów trzech itd.

- 2.3. Pierwszym wariantem upraszczającym jest operator *ROLLUP*, który wylicza podsumowania na wszystkich poziomach wymienionych, jako argument grupowania i dodaje podsumowanie dla pełnego zestawu rekordów. Przykładem jest wyznaczenie wszystkich podsumowań brutto:

```
SELECT Nazwa, Nazwisko, SUM(Brutto) AS Razem
```

```
FROM Działy JOIN Osoby
```

```
ON Działy.IdDziału = Osoby.IdDziału
```

```
JOIN Zarobki
```

```
ON Osoby.IdOsoby = Zarobki.IdOsoby
```

```
GROUP BY Nazwa, Nazwisko
```

```
WITH ROLLUP;
```

- 2.4. Kolejnym przykładem jest zastosowanie operatora *CUBE*, który stanowi rozszerzenie *ROLLUP*:

```
SELECT Nazwa, Nazwisko, SUM(Brutto) AS Razem
```

```
FROM Działy JOIN Osoby
```

```
ON Działy.IdDziału = Osoby.IdDziału
```

```
JOIN Zarobki
```

```
ON Osoby.IdOsoby = Zarobki.IdOsoby
```

```
GROUP BY Nazwa, Nazwisko
```

```
WITH CUBE
```

```
ORDER BY Nazwa, Nazwisko;
```

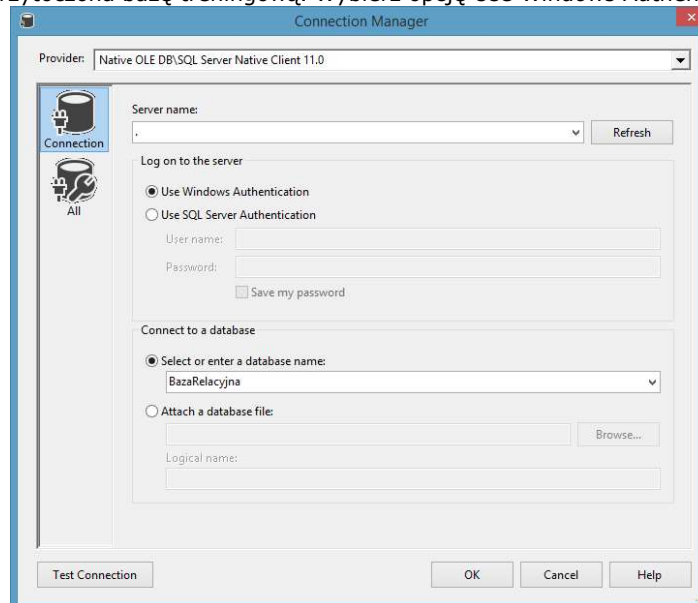
Rezultatem jest rozszerzenie poprzedniego wyniku o podsumowania dla pracowników z odseparowaniem informacji od nazw działów, w których pracują.

- 2.5. Następnym elementem jest operator *GROUPINGS SETS*. Powoduje on, że sumy wyznaczone są na obu poziomach, wskazanych parametrami grupowania. W obu przypadkach informacja jest odseparowana od informacji z drugiego z poziomów. Ciekawą cechą tego operatora jest możliwość łączenia atrybutów:

```
SELECT Wojewodztwo, Miasto, NazwaProducenta, NazwaTowaru, SUM(cena*szt) AS Wartosc
FROM Wojewodztwa
JOIN Miasta ON Wojewodztwa.IdWojewodztwa=Miasta.IdWojewodztwa
JOIN Producenci ON Miasta.IdMiasta=Producenci.IdMiasta
JOIN Towar ON Producenci.IdProducenta=Towar.IdProducenta
JOIN Transakcje ON Towar.IdTowaru=Transakcje.IdTowaru
GROUP BY GROUPING SETS ((Wojewodztwo, Miasto), (NazwaProducenta, NazwaTowaru));
```

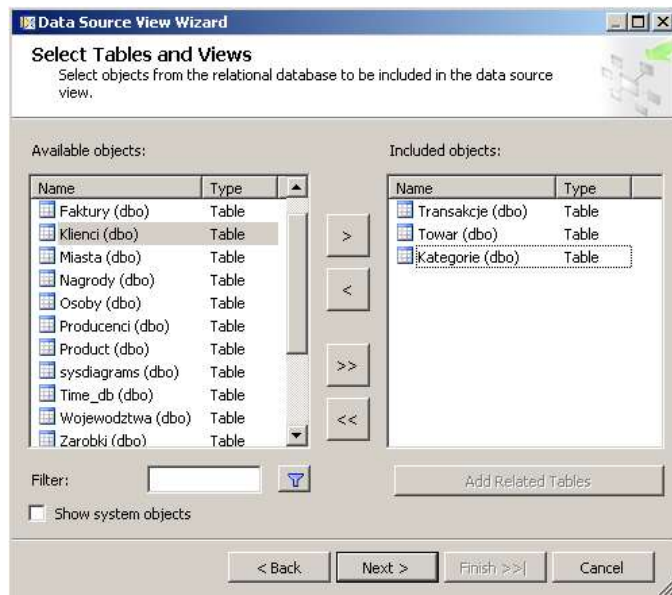
### 3. Tworzenie elementów hurtowni danych

- 3.1. Proszę zapoznać się z widokami: „*Wszystkie\_transakcje*” oraz „*czas*”.
- 3.2. Uruchom Visual Studio 2017 (SSDT) i stwórz nowy projekt typu Analysis Services.
- 3.3. Pierwszym krokiem tworzenia struktur analitycznych OLAP jest definicja źródła danych. Kliknij prawym na Data Sources w Solution Explorer. Dodaj nowe źródło zostawiając domyślny Provider, serwer lokalny i przytoczona bazę treningową. Wybierz opcję Use Windows Authentication.



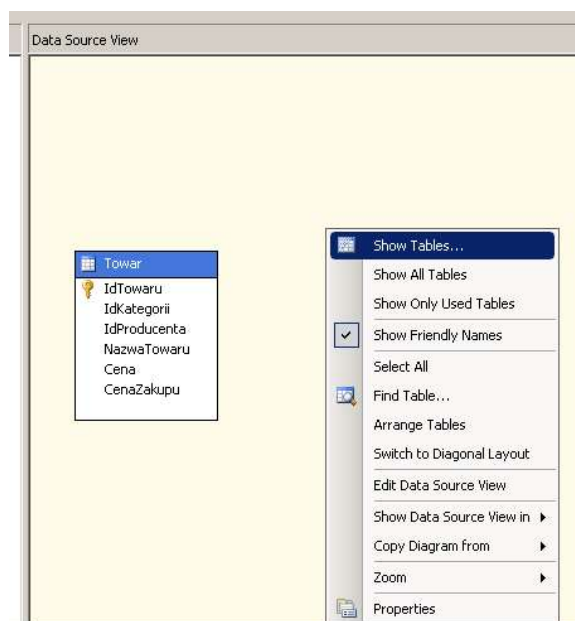
Zaznacz utworzone połączenie a następnie wybierz opcję „*Use the service account*”

- 3.4. Kolejnym krokiem jest utworzenie perspektywy źródła danych (prawym na Data Source Views w Solution Explorer). Wybieramy zdefiniowane już źródło a w kolejnym etapie wybieramy tabele i perspektywy źródłowe do realizacji pozostałych elementów (należy pamiętać, że powinny one zapewniać możliwość zdefiniowania co najmniej jednego wymiaru oraz tabeli faktów). Rozpoczniemy od minimalistycznego podejścia tj. wyboru trzech tabel: Towar i Kategorie jako wymiary oraz Transakcje jako tabela faktów:



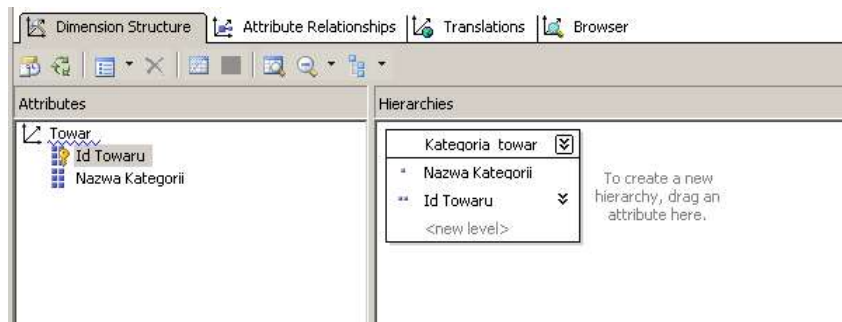
Po zakończeniu kreatora mamy dostępną graficzną prezentację tworzonej struktury. Zwrot strzałki wskazuje tabele będące w relacji „jeden do wielu” po stronie jeden.

- 3.5. Przejdźmy do tworzenia właściwych elementów hurtowni, rozpoczynając od wymiarów (Dimension w Solution Explorer). Zaznaczamy „Use an existing table” następnie zaznaczamy tabelę, która definiuje najniższy poziom w hierarchii, a jednocześnie jest bezpośrednio połączona z tabelą faktów. Będzie to tabela Towar. Wybierz ją oraz IdTowaru jako klucz wymiaru oraz atrybut NazwaTowaru jako nazwę kolumny. Oznacz related tables i zaznacz tylko Id Towaru w available attributes. Zakładka Dimension Structure w oknie edycyjnym reprezentuje strukturę wymiaru. Aby możliwe było obejrzenie rzeczywistych danych wymiar musi zostać najpierw przetworzony (ikona Process, podobna do znaku recyklingu). Takie przetworzenie jest konieczne zarówno bezpośrednio po zdefiniowaniu obiektu, jak i po każdej modyfikacji struktury logicznej. Po przetworzeniu elementu konieczne jest ręczne wymuszenie odświeżania (Reconnect) w zakładce Browser gdzie możliwe jest obejrzenie przetworzonej struktury. Dla każdego nawet najprostszego wymiaru tworzony jest element korzenia All, reprezentujący wszystkie wartości poziomu poniżej – wszystkie towary. Aby dodać kolejne atrybuty czy poziomy hierarchii, musimy przedtem dodać definicje kolejnych tabel (zakładka Dimension Structure). Wybieramy pozycję Show Tables, dostępna w menu podręcznym, gdzie zaznaczamy tabelę Kategorie, która stanowi poziom nadrzędny do już istniejącego:



Po dodaniu tabeli pole NazwaKategorii (definiującej kolejny atrybut) należy przeciągnąć z definicji widoku do panelu Attributes. Po przetworzeniu i odświeżeniu dostępne są struktury dla obu atrybutów (zakładka Browser). Aby zbudować strukturę wielopoziomową należy utworzyć hierarchię w panelu

Hierarchies (metodą „przeciągnij i upuść”). Zaczynamy od atrybutu najogólniejszego do atrybutu najbardziej szczegółowego oraz nadajemy nazwę hierarchii „Kategoria\_towar”:



Po przetworzeniu poza płaskimi strukturami mamy widoczną strukturę o dwóch poziomach.

- 3.6. Przejdźmy do definicji kostki. Podobnie jak poprzednio uruchamiamy kreatora (Cubes w Solution Explorer). Wybieramy „Use existing tables”, wskazujemy tabele faktów (Transakcje), wybieramy miary dla kostki (Szt, Transakcje Count) oraz wskazujemy wymiary, które będą wchodziły do definicji kostki (Towar). Tabela faktów jest zaznaczona na żółto a tabele wymiarów na niebiesko. Po przetworzeniu możliwe jest obejrzenie wartości miar zgodnie z zdefiniowanymi atrybutami wymiarów w zakładce Broszwer. Lewy panel zawiera logiczną definicję kostki – miary i wymiary. Prawy jest formą tabeli przestawnej, do której elementy ją definiujące przeciągamy za pomocą myszy:

Baza Relacyjna

Metadata

Search Model

Measure Group:

<All>

Baza Relacyjna

Measures

Transakcje

Szt

Transakcje Count

KPIs

Towar

Id Towaru

Nazwa Kategorii

Kategoria towar

Members

Nazwa Kategorii

Id Towaru

Calculated Members

Dimension

<Select dimension>

Hierarchy

Operator

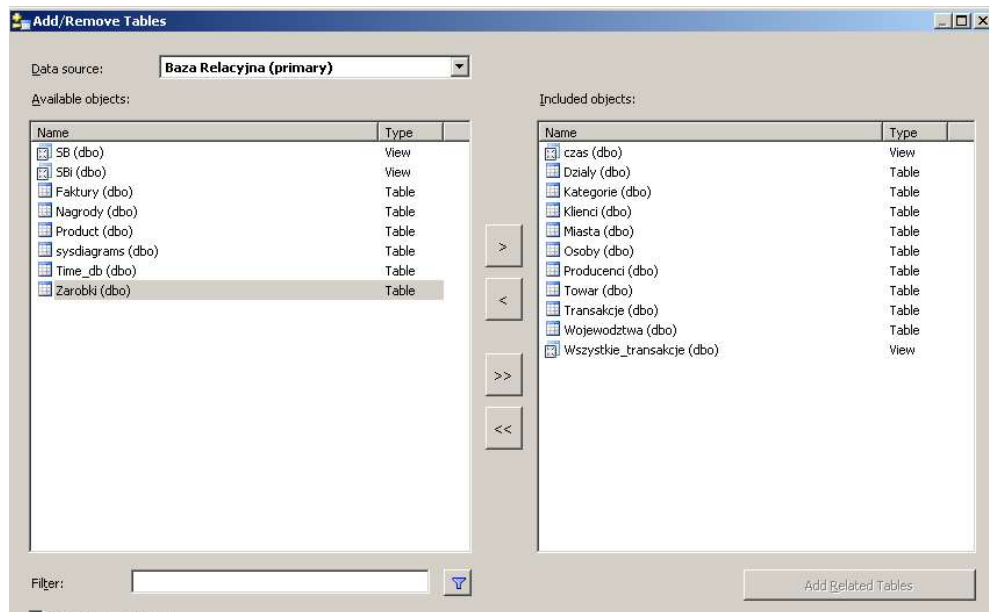
Nazwa Kategorii	Id Towaru	Szt	Transakcje Count
Akcesoria	Pasek	20	6
Akcesoria	Podstawka	9	4
Akcesoria	Pokrowiec	14	6
Akcesoria	Pokrowiec B	2	2
Akcesoria	Statyw	6	2
Akcesoria	Statyw AX	10	4
Akcesoria	Torba	19	5
Akcesoria	Torba Lux	14	4
Akcesoria	Torba St...	19	6
Akcesoria	Uchwyt	27	9
Akcesoria	Zestaw Lux	16	5
GPS	Gio 15	13	6
GPS	Gio 200 S	8	2
GPS	Mov 200	12	4
GPS	Mov 300	23	7
GPS	Mov 350	19	6
GPS	Tom A1	21	5
Muzyka	Kapela	15	6
Muzyka	Koncerty ...	21	6

#### 4. Modyfikacja struktury hurtowni danych

Pomimo pełnej funkcjonalności utworzonych wcześniej struktur spróbujmy utworzyć bardziej złożone rozwiązanie.

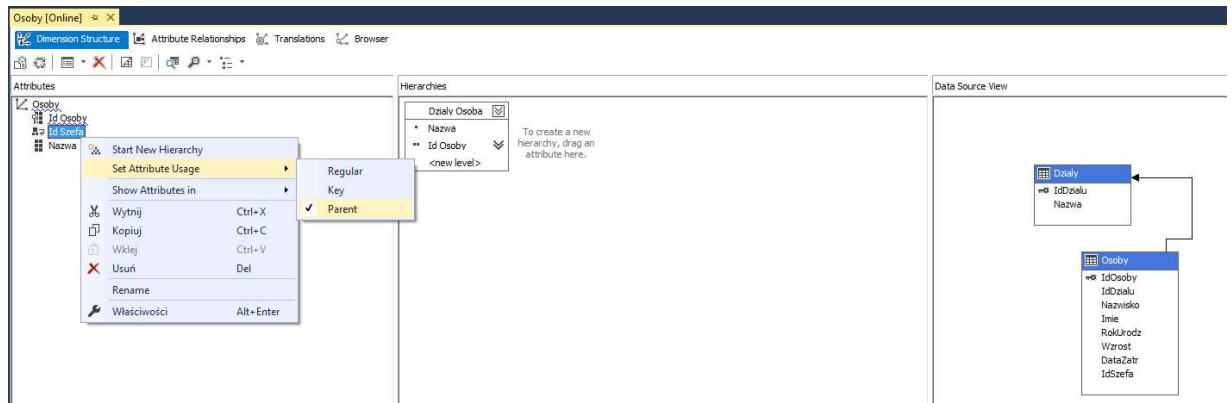
- 4.1. Pierwszym krokiem jest zmiana danych definiujących perspektywę źródła. W edytorze graficznym korzystamy z opcji „Add/Remove Tables” menu kontekstowego. Przeniesiemy pewne obiekty tak jak zostało to pokazane na rysunku:





Ponieważ dla perspektyw nie są definiowane klucze obce, oba zaimportowane obiekty tego typu nie są połączone z żadnym z pozostałych. Dodajmy relacje między tabelą faktów (perspektywa Wszystkie\_transakcje) a pozostałymi elementami wykorzystując pola identyfikatorów (IdTowaru, IdOsoby, IdKlienta). Realizujemy to przeciągając odpowiednie pola tabeli faktów do pól kluczy głównych innych tabel. Perspektywę czas łączymy względem pola DataNum, które zawiera datę przekształconą na liczbę.

- 4.2. Utwórz wymiar opierający się na tabeli Osoby. Atrybut IdSzefa ustala hierarchie podporządkowania. Dlatego należy zmienić sposób użycia tego atrybutu z Regular na Parent.



- 4.3. Utwórz wymiar czasu, gdzie atrybutami wymiaru są wszystkie pola widoku reprezentujące rok, kwartał, miesiąc i datę. Utwórz także hierarchię Rok\_kwartał\_miesiąc.
- 4.4. Po utworzeniu wymiarów zrealizuj kostkę o nazwie **Zyski** zawierającą wymiary: Towar, Osoby oraz Czas oraz wszystkie miary a następnie przedstaw wyniki dla wymiarów Czas i Towar oraz miar Wartość i Ilość. Możliwe jest także utworzenie wymiaru czasowego na bazie szablonu. W kreatorze wymiaru wybieramy opcję „Generate a time table on the server” następnie uzupełniamy odpowiednie wartości tak jak na rysunku:



**Dimension Wizard**

**Define Time Periods**  
Select the time periods to use when generating the hierarchies.

First calendar day: 1 stycznia 2001

Last calendar day: 31 grudnia 2004

First day of the week: Sunday

Time periods:

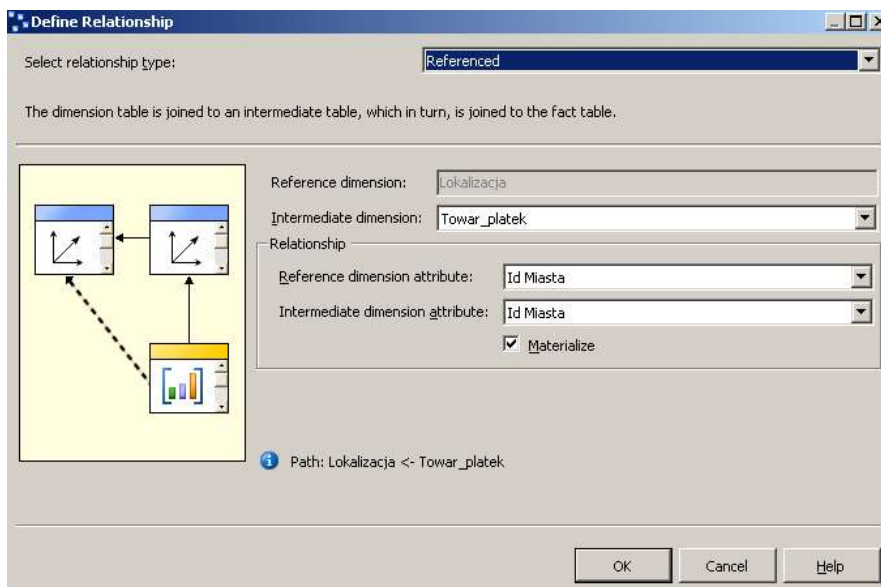
- ☒ Year
- ☐ Half Year
- ☒ Quarter
- ☐ Trimester
- ☒ Month
- ☐ Ten Days
- ☐ Week
- ☒ Date

Language for time member names: English (United States)

< Back   Next >   Finish >>   Cancel

Zaznaczamy opcję „Regular calendar”. Po utworzeniu wymiaru należy go dodać do kostki (opcja „Add Cube Dimension”) oraz połączenie go z tabelą faktów. Połączenie realizujemy w zakładce Dimension Usage. Tylko wymiar Time nie jest połączony. Po uruchomieniu kreatora definicji relacji zaznaczamy typ relacji Regular, Granularity attribute: Date a w tabelce Relationship w kolumnie Measure Group Columns wybieramy pole DataNum. Przedstaw te same wyniki wykorzystując wymiar Time.

- 4.5. Utwórz wymiar Klienci, który pozwala określić lokalizację każdego klienta na dwóch poziomach hierarchii (Województwo, Miasto) oraz dane personalne ograniczone do nazwiska.
- 4.6. Utwórz dwupoziomową hierarchię opisującą przyporządkowanie pracowników firmy (tabela Osoby) do struktury organizacyjnej (tabela Działy)
- 4.7. Rozszerz wymiar Towar poprzez dodanie trzech tabel definiujących producenta towaru oraz lokalizację geograficzną firmy (Miasta, Województwa). Dodaj rozbudowaną hierarchię woj\_miasto\_prod\_towar wykorzystującą atrybuty z dodanych tabel. Zmień nazwę wymiaru na Towar\_calosc.
- 4.8. Przedstaw wyniki nowej kostki dla tych samych miar co wcześniej oraz wymiarów Klient i Towar\_calosc.
- 4.9. Utwórz kostkę o strukturze płatka śniegu. Dwa wymiary (opisujące towary i klientów) mają wspólną część (tabele: Miasta i Województwa). Można ten fragment generować oddzielnie i dołączać do pozostałych atrybutów definiujących wymiar. Utwórz niezależny wymiar Lokalizacja z tabel: Województwa, Miasta oraz hierarchię z atrybutami IdMiasta i Województwo. Następnie utwórz wymiar Towar\_platek zawierający atrybuty opisujące nazwę towaru i nazwę producenta oraz IdMiasta (klucz obcy, który zostanie wykorzystany do dołączenia wymiaru Lokalizacja). Podobnie utwórz wymiar Klienci\_platek zawierający nazwisko klienta i IdMiasta. Kolejnym krokiem jest utworzenie kostki Zysk\_platek z wcześniejszymi wymiarami. Po jej utworzeniu przechodzimy do zakładki dimension usage a tam tworzymy powiązania z tabelą faktów wymiaru Lokalizacja z wymiarem Towar\_platek i jego kopii z wymiarem Klienci\_platek jak na rysunku:

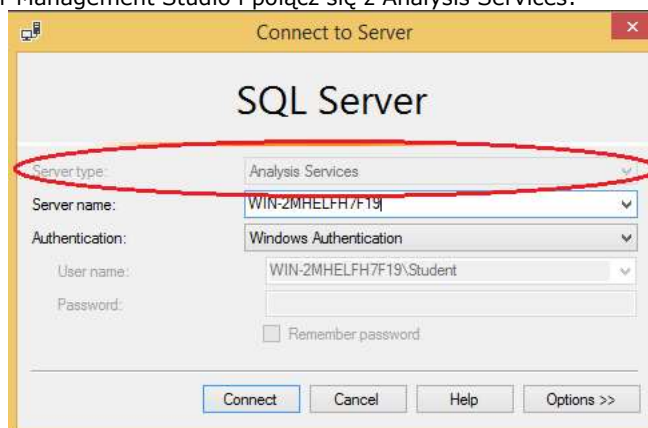


Przedstaw wyniki kostki dla miary Wartosc i wymiarów Klienci\_platek i Towar\_platek z ich połączeniem z wymiarem odpowiadającym lokalizacji. Należy pamiętać, które z hierarchii są ze sobą połączone referencyjnie i tylko te umieszczać w opisie tego samego elementu (kolumny lub wiersza).

## 5. Analiza danych za pomocą MDX

Możemy manipulować elementami hurtowni danych, wykorzystując język zapytań wielowymiarowych MDX (Multidimensional extension). Do tego celu można użyć narzędzia SQL Server Management Studio.

### 5.1. Uruchom SQL Server Management Studio i połącz się z Analysis Services:



Po połączeniu naciśnij przycisk New Query. Wybierz stworzony wcześniej projekt BI i wybierz kostkę Zyski.

### 5.2. Dość elementarnym zapytaniem MDX jest zapytanie wyświetlające wartości wszystkich miar dla kostki:

```
SELECT [Measures].[Wartosc] ON COLUMNS FROM [Zyski];
```

### 5.3. Wyświetlenie wszystkich członów dla danego atrybutu wymiaru lub miary uzyskujemy dzięki zastosowaniu metody members:

```
SELECT [Towar_calosc].[Nazwa kategorii].members ON ROWS, [Measures].[Wartosc] ON COLUMNS  
FROM [Zyski];
```

### 5.4. Kolejny przykład obrazuje wyświetlanie zestawu atrybutów – zbioru Set. Jego definicja ujęta jest w nawiasy klamrowe i oddzielone przecinkami. Metoda children zwraca wszystkie członów podrzędnego poziomu wskazanego atrybutu hierarchii:

```
SELECT Measures.Members ON COLUMNS,  
{[Towar_calosc].[Kategoria_towar].[Nazwa Kategorii].[GPS].Children,
```

```
[Towar_calosc].[Kategoria_towar].[Nazwa Kategorii].[Muzyka]. Children} ON ROWS
FROM [Zyski];
```

- 5.5. Możliwe jest zastosowanie w definicji zestawu atrybutów wskazanego członka poziomu nadrzędnego oraz wszystkich jego dzieci. W tym celu można użyć wcześniejszej metody children lub funkcji DESCENDANTS():

```
SELECT Measures.Members ON COLUMNS, {[Towar_calosc].[Kategoria_towar].[Nazwa Kategorii].
[GPS],

DESCENDANTS ([Towar_calosc].[Kategoria_towar].[Nazwa Kategorii].[GPS], [Id Towaru])} ON
ROWS FROM [Zyski];
```

Metoda children pozwala tylko na wyznaczenie węzłów leżących bezpośrednio poniżej wybranego węzła, natomiast funkcja DESCENDANTS umożliwia wyświetlenie potomków leżących w dowolnej odległości od węzła macierzystego.

- 5.6. MDX umożliwia tworzenie miar ad hoc. Miary te są realizowane w sekcji WITH. Definicję rozpoczyna słowo kluczowe MEMBER, po którym określa się nazwę kwalifikowaną miary a po słowie kluczowym AS następuje właściwa definicja:

```
WITH MEMBER Measures.ZyskProc AS

([Measures].[Wartosc]-[Measures].[Wartosc Zakupu]) /

[Measures].[Wartosc Zakupu], FORMAT_STRING = '#.00%'

SELECT

NONEMPTY([Towar_calosc].[Nazwa kategorii].members) ON COLUMNS,

NONEMPTY([Klienci].[Id Wojewodztwa].members) ON ROWS

FROM [Zyski]

WHERE Measures.ZyskProc;
```

- 5.7. Miary można wyznaczać jako wyrażenia dla różnych elementów i poziomów hierarchii. Możliwe jest wyznaczenie sumy bieżącej przy użyciu złożenia funkcji PERIODTODATE(), pozwalającej na wskazanie atrybutów na dowolnym poziomie wskazywanym pierwszym parametrem oraz funkcji SUM():

```
WITH MEMBER Measures.Zysk_YTD AS

SUM(PERIODTODATE([Time].[Year - Quarter - Month - Date].[Year],

[Time].[Year - Quarter - Month - Date].CurrentMember), [Wartosc] - [Wartosc Zakupu]),

FORMAT_STRING='#.00'

SELECT

[Towar_calosc].[Nazwa Kategorii].MEMBERS ON COLUMNS,

DESCENDANTS([Time].[Year - Quarter - Month - Date], [Time].[Year - Quarter - Month - Date].
[Month]) ON ROWS

FROM [Zyski]

WHERE Measures.Zysk_YTD;
```

- 5.8. Analizując wyniki poprzedniego zapytania widać, że występują w niej wartości puste. Nie można się ich pozbyć za pomocą funkcji NONEMPTY() – należy zastosować funkcję FILTER z wyrażeniem określającym, że interesują nas tylko te kolumny, dla których zysk jest różny od null (usuwanie pustych kolumny):

```
WITH MEMBER Measures.Zysk_YTD AS

SUM(PERIODTODATE([Time].[Year - Quarter - Month - Date].[Year],
```

[Time].[Year - Quarter - Month - Date].CurrentMember), [Wartosc] - [Wartosc Zakupu]),  
 FORMAT\_STRING='#.00'

SELECT FILTER(

[Towar\_calosc].[Nazwa Kategorii].MEMBERS, ([Wartosc] - [Wartosc Zakupu]) <> NULL) ON  
 COLUMNS,

DESCENDANTS([Time].[Year - Quarter - Month - Date], [Time].[Year - Quarter - Month - Date].  
 [Month]) ON ROWS

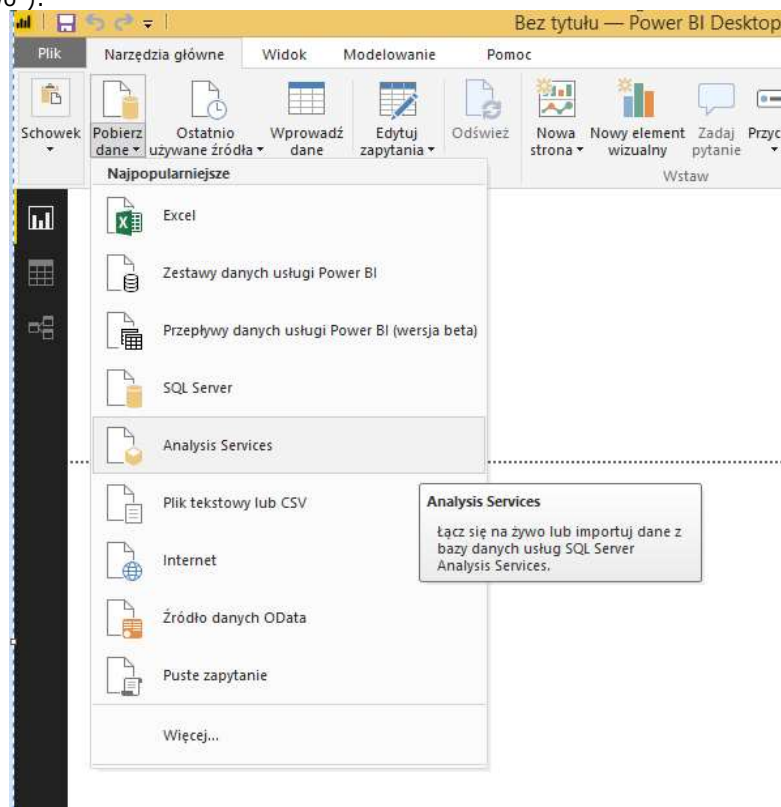
FROM [Zyski]

WHERE Measures.Zysk\_YTD;

## 6. Raportowanie przy użyciu MS Power BI

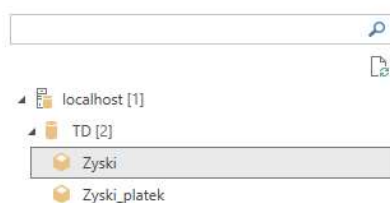
6.1. Możemy manipulować elementami hurtowni danych, aby utworzyć raporty, dashboard'y i interaktywne wizualizacje używając różnych aplikacji klienckich takich jak: Reporting Services, Excel lub Power BI. Ostatnia z nich jest przykładem nowego podejścia do raportowania. Składa się z kilku komponentów, które umożliwiają tworzenie i zarządzanie interaktywnymi wizualizacjami/dashboard'ami/raportami danych.

6.2. Uruchom Power BI Desktop i połącz się z lokalną instancją serwera Analysis Services (wybierz opcję „Połącz na żywo”):



Następnie wybierz kostkę Zyski w oknie nawigatora:

### Nawigator



Zyski

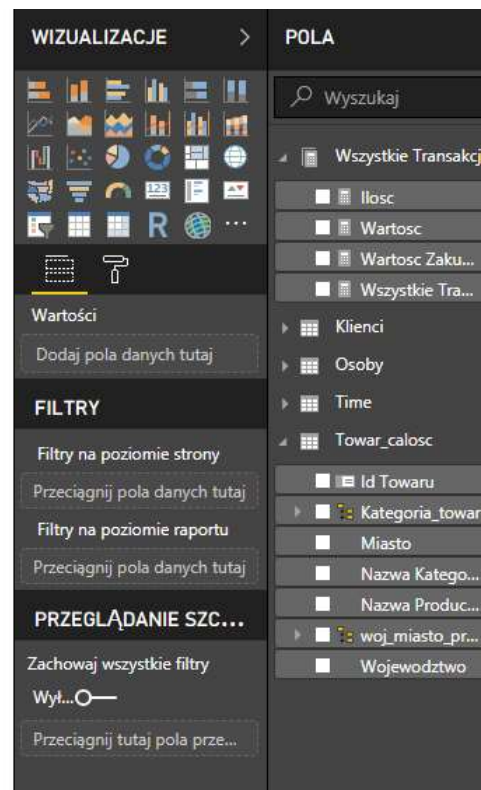
Ostatnia modyfikacja: 12/01/2018 17:52:55

Ta perspektywa zawiera następujące wymiary i miary

Klienci; Osoby; Time; Towar\_calosc; Ilosc; Wartosc; Wartosc Zakupu; Wszystkie Transakcje Count

6.3. Utwórz prostą wizualizację tzn. Tabelę przestawną z hierarchią "Kategoria\_towar" oraz dwoma miarami Ilość i Wartość jak na rysunku:

Nazwa Kategorii	Ilość	Wartość
Oprogramowanie	192	13 515,05
Akcesoria	156	7 765,48
Podręczniki	126	8 597,75
Telefony	126	10 928,25
Ozdoby	97	6 233,98
GPS	96	9 794,00
Muzyka	82	8 167,92
RTV	78	10 164,43
Optyka	74	7 057,26
Zegary	71	5 856,49
Pomoce	42	3 189,13
Opakowania	39	1 714,70
<b>Suma</b>	<b>1181</b>	<b>92 984,44</b>



6.4. Proszę poeksperymentować z interfejsem tego narzędzia. Spróbować stworzyć inne wizualizacje na bazie innych danych oraz sprawdzić, jak działa przechodzenie względem hierarchii itp.

## Sposób oceny/ uzyskania zaliczenia

Ocenie podlega wiedza z zakresu wykładu w formie aktywności na laboratoriach oraz zadań dodatkowych do wykonania z wydłużonym terminem według skali ocen zgodnej z regulaminem studiów AGH (3.0 – 5.0).

## Literatura

- W. H. Inmon, *Building the Data Warehouse*, QED Tech. Pub. Group, 1992
- R. Kimball, *The Data Warehouse ETL Toolkit*, Wiley Publishing, Inc., 2004
- <https://docs.microsoft.com/pl-pl/sql/sql-server/?view=sql-server-2017>
- <https://docs.microsoft.com/en-us/power-bi/>
- <https://docs.microsoft.com/en-us/sql/mdx/multidimensional-expressions-mdx-reference?view=sql-server-2017>