Radosław Biniek, Adam Pelikant
Wydział Informatyki
Wyższej Szkoły Informatyki w Łodzi

ZASTOSOWANIE ROZSZERZENIA MDX DO TWORZENIA HURTOWNI DANYCH NA PLATFORMIE MICROSOFT SQL SERWER

Streszczenie – Praca zawiera opis rozszerzenia MDX (MultiDimensional Extension) do tworzenia hurtowni danych w oparciu o oprogramowanie Microsoft SQL Server 2005. Tworzenie i zarządzanie hurtownią danych odbywa się w oparciu o środowisko graficzne Microsoft Visual Studio 2005, czysty język MDX przy użyciu usługi Analysis Services oraz przy pomocy aplikacji windowsowej napisanej w języku C#. Aplikacja pozwala na tworzenie dodatkowych miar kalkulowanych oraz przeglądanie danych zawartych w kostkach analitycznych. Użyta w aplikacji biblioteka ADOMD.NET odpowiada za komunikację programu z serwerem analitycznym Microsoft Analysis Services.

1 Wstęp

Wiarygodna informacja staje się podstawą działania i konkurencyjności przedsiębiorstw. Niestety często nadmiar informacji jest bardziej szkodliwy niż jej brak. Dlatego bardzo ważną zdolnością jest umiejętne segregowanie danych, tak, aby można było wyciągnąć z nich przydatną informację. Rozwiązaniem jest hurtownia danych, która jest relacyjną bazą danych wspomagającą zarządzanie. Jednak znajdujące się w niej dane są mało użyteczne, dopóki nie zostaną odpowiednio przetworzone, tak, aby dostarczyć potrzebne informacje osobom podejmującym decyzje.

Oprogramowanie Microsoft SQL Server pozwala tworzyć i zarządzać hurtownią danych. W tym celu wykorzystywana jest usługa Analysis Services, która używa technologii OLAP(ang. OnLine Analytical Processing – Przetwarzanie Analityczne w Trybie Online). Technologia ta wspomaga procesy podejmowania decyzji, gdyż jest systemem bazodanowym zoptymalizowanym właśnie pod tym kątem.

Wykorzystywany w niej język MDX(ang. Multidimensional Expression language – Język Wielowymiarowych Zapytań), który jest częścią specyfikacji technologii OLAP, jest niezwykle rozbudowany w usłudze Analysis Services. Pozwala nie tylko na tworzenie i zarządzanie kostkami danych, kalkulacjami czy wskaźnikami wydajności, ale również

na uzyskiwanie kluczowych, z punktu widzenia przedsiębiorstwa, informacji. Dlatego właśnie celem niniejszej pracy jest pokazanie poszczególnych etapów tworzenia hurtowni danych, zarządzania nią i dostosowywania dla potrzeb danego przedsiębiorstwa, tak, aby możliwa była przekrojowa analiza danych. Daje to ogromne możliwości zwiększenia konkurencyjności firmy i zoptymalizowania jej działalności.

2 Wprowadzenie do technologii Microsoft SQL Server 2005 Analysis Services

Coraz częściej w świecie biznesowym mamy do czynienia, ze względnie nowym terminem, a mianowicie z tzw. logiką biznesową (BI – ang. Business Intelligence). Jej koncepcja opiera się na wykorzystaniu istniejących w firmie informacji tak, aby zoptymalizować działalność przedsiębiorstwa. Innymi słowy należy tak przetworzyć dane, aby ułatwić szybsze i lepsze podejmowanie decyzji biznesowych, które zwiększą konkurencyjność firmy.

BI jest mocno związane z tzw. hurtownią danych, która stanowi repozytorium do przechowywania i analizowania informacji numerycznych [2], [6]. W praktyce hurtowania jest bazą danych, która łączy dane ze wszystkich systemów bazodanowych znajdujących się w firmie. W związku z tym zawiera ona olbrzymią ilość informacji, które zostały zebrane w określonym czasie. Wszelkie zadania związane z hurtownią danych są skierowane przede wszystkim na analizę tych informacji, dlatego jedne z jej głównych założeń to wspomaganie decyzji, analiza efektywności przedsiębiorstwa oraz przetwarzanie analityczne OLAP.

Mówiąc o hurtowni danych nie unikniemy pojęć bezpośrednio z nią związanych, czyli m.in. wielowymiarowa baza danych. Aby wyjaśnić ten termin należy wcześniej zdefiniować pojęcie wymiaru, które oznacza listę etykiet, za pomocą których możemy tabelaryzować wartości. Wielowymiarowa baza składa się zatem z więcej niż jednej etykiety, która tworzy po prostu nowy sposób podziału danej informacji. W ramach wymiaru, wszystkie jego elementy są ze sobą powiązane i nazywane są członkami tego wymiaru. Nazwa członka wymiaru nazywana jest atrybutem [2]. Atrybuty mogą być grupowalne lub nie. Grupowalne atrybuty mogą łączyć się ze sobą tworząc hierarchie, np. jeśli mamy atrybuty Towar i Producent to wiadomo, że dany produkt jest tworzony przez określonego producenta. W związku z tym taka zależność może tworzyć naturalna hierarchię[2]. Warto jednak dodać, że hierarchie wcale nie muszą być stworzone z atrybutów bezpośrednio ze sobą związanych.

Hurtownia danych zawiera także tabele faktów, które charakteryzują się tym, że przechowują szczegółowe wartości, lub inaczej mówiąc

fakty. W każdej kolumnie tabeli faktów znajduje się jedna miara, czyli dająca się podsumować numeryczna wartość służąca do monitorowania aktywności biznesowej [2].

Znając już podstawowe terminy związane z hurtownią danych możemy zapoznać się z technologią, która umożliwia przetworzenie danych w niej zawartych. Technologia OLAP jest jedną z najlepszych technologii przetwarzania danych na informacje [2]. W najprostszym tłumaczeniu jest to technologia bazodanowa zoptymalizowana pod kątem analizy danych, która odbywa się w trybie „online". Kostka danych (ang. Cube) to termin, który jest jednoznacznie kojarzony z tą technologią i oznacza integracje tabeli faktów z tabelami wymiarów. Kostka w geometrii jest kojarzona z trzeba wymiarami, ale w OLAP liczba wymiarów może się wahać od jednego do tylu, ile potrzebujemy. Technologia ta została zaprojektowana w taki sposób, aby zapewnić niezwykle szybie wykonywanie zapytań przy użyciu kostek danych oraz obliczeń wykonywanych w czasie rzeczywistym. Dzięki temu otrzymujemy narzędzie, które szybko, łatwo i skutecznie pozwoli na dogłębne analizowanie danych oraz ułatwi podejmowanie decyzji biznesowych. Analiza trendów sprzedaży,

czy analiza biznesowa nie stanowi już żadnego wyzwania, gdyż szybki dostęp do przetworzonych w kostce danych pozwala na przeprowadzanie scenariuszy „co się stanie, gdy", a co za tym idzie sprawne i właściwe podejmowanie decyzji.

Aby pobrać informacje z kostek danych należy użyć języka opartego na wyrażeniach wielowymiarowych. Język MDX (ang. Multidimensional Expression) jest standardowym językiem zapytań dla OLAP.

Oprogramowanie, które pomoże w tworzeniu kostek danych, zarządzaniu nimi oraz tworzeniu zapytań MDX to produkt firmy Microsoft o nazwie SQL Server 2005. Zawarta w nim usługa Analysis Services zapewni jednolity i zintegrowany wgląd we wszystkie dane biznesowe firmy. Zintegrowane narzędzia projektowe pozwalają projektować oraz tworzyć wymiary oraz kostki danych, a także zapewniają aktualność danych. Narzędzie przeznaczone dla programistów o nazwie Business Intelligence Development Studio (BIDS) dzięki wbudowanym narzędziom projektowym dla logiki biznesowej znacznie ułatwia tworzenie i zarządzanie stworzonymi kostkami danych.

3 Struktura bazy danych

Baza danych użyta w projekcie oraz zasilająca hurtownie stanowi bazę relacyjną, która zaprojektowana jest w taki sposób, aby każdy mógł bez problemów zrozumieć jej strukturę. Pomimo swej prostoty umożliwi pokazanie szeregu możliwości analizy zgromadzonych w niej danych.

Baza zawiera fikcyjne dane pochodzące z firmy zajmującej się sprzedażą różnego rodzaju produktów. Sprzedaż towarów odbywa się na terenie kraju w kilku miastach. Transakcje odbywają się w dwojaki sposób: towar zamawiany jest przez Internet bądź w sklepie znajdującym się w danym mieście. Każdy towar należy do określonej kategorii oraz jest produkowany przez danego producenta. Tak zaprojektowana baza umożliwia analizę produktów, klientów, sprzedaży oraz wiele innych aspektów.

4 Projektowanie i tworzenie kostek danych

Aby utworzyć własną kostkę danych potrzebujemy specjalistycznych narzędzi projektowych. Do tego celu możemy użyć platformy programistycznej firmy Microsoft o nazwie Visual Studio 2005. Jednak nie wszyscy potrzebują tak rozbudowanego narzędzia, które dodatkowo jest dość drogie. Dlatego wraz z instalacją Microsoft SQL Server 2005 Analysis Services zostanie także zainstalowana okrojona wersja Visual Studio 2005, która dostarczy niezbędnych składników logiki biznesowej serwera SQL Server, aby sprawnie budować i zarządzać kostkami danych. Należy dodać, że zainstalowana wersja Visual Studio 2005 dostarczona z SQL Server 2005 nazywa się inaczej, a mianowicie: *SQL Server Business Intelligence Development Studio (BIDS)*. Dodatkowym wymaganiem jest zainstalowanie na komputerze .NET Framework w wersji 2.0. Jest to biblioteka programistyczna stworzona przez firmę Microsoft, która zawiera niezbędne komponenty wymagane do poprawnego uruchomienia i działania aplikacji na niej opartych.

Aby rozpocząć tworzenie pierwszej kostki danych należy uruchomić program *SQL Server Business Intelligence Develpoment Studio* znajdujący się w folderze *Microsoft SQL Server 2005* w menu start. Kolejnym krokiem jest stworzenie nowego projektu typu *Analysis Services*. Wybieramy kolejno z głównego menu: File→New→Project. Pojawi się nowe okno, na którym po jego lewej stronie z widoku *Project types* należy wybrać *Business Intelligence Projects*. Po wybraniu tego typu w oknie *Templates* po prawej stronie zaznaczamy *Analysis Services Project*. Nadajemy nazwę projektu w polu *Name* i potwierdzamy swój wybór przyciskiem OK. Kolejne etapy to:

1. Tworzenie i dodawanie źródła danych - jest to pierwszy etap w procesie budowania kostki danych. Źródło danych zawiera wszelkie informacje opisujące połączenie z serwerem, bazą danych oraz wymagane do tego poświadczenia tożsamości.
2. Tworzenie i dodawanie widoku źródła danych - w którym definiujemy zakres danych potrzebnych do zapełnienia wymiarów i kostek danych. Nasza kostka nie musi analizować

zakresu całej hurtowni danych. Możemy wybierać różne wycinki danych do analizy. Jeżeli podczas projektowania widoku źródła danych nie wiemy do końca jakie tabele i widoki będą nam potrzebne do analizy, nie należy się tym przejmować, gdyż zawsze istnieje możliwość modyfikacji tych informacji.

3. Tworzenie kostek danych - Po pomyślnym stworzeniu źródła oraz widoku źródła danych możemy już przejść do stworzenia pierwszej kostki danych. W tym celu użyjemy kreatora, który poprowadzi nas przez cały proces. W związku z tym, że nie mamy zdefiniowanych żadnych wymiarów dla kostki danych, kreator stworzy je dla nas z tabel, które zaznaczymy. Istnieje oczywiście możliwość stworzenia wymiarów i budowy kostki danych na nich opartej.

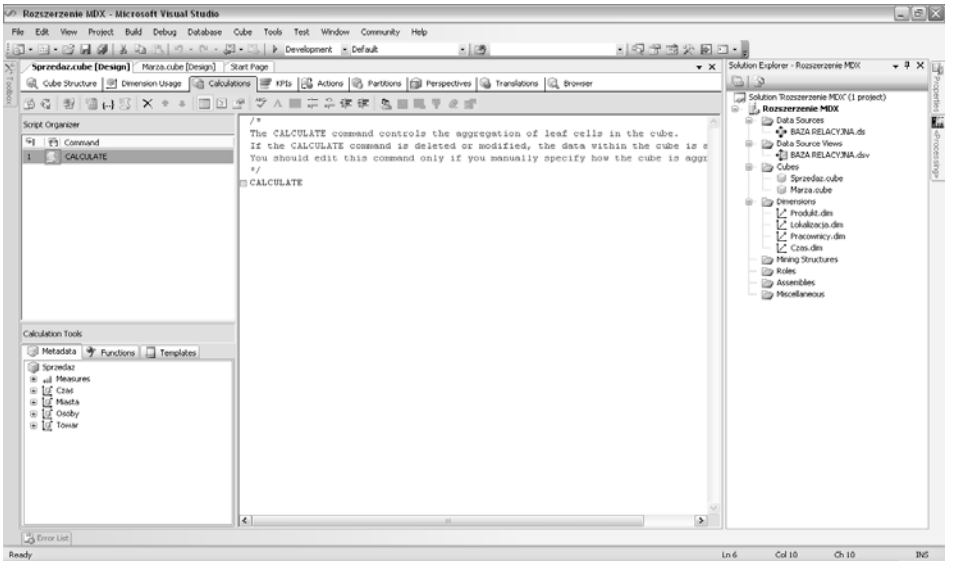
   Po zamknięciu kreatora zostanie utworzona nowa kostka analityczna oraz zostaniemy także automatycznie przeniesieni do specjalnego narzędzia, które pozwoli na dodatkową rozbudowę kostki danych. Po zakończeniu należy jeszcze wdrożyć projekt na serwer *Analysis Services*, aby w pełni przejrzeć wszystkie wymiary i przeglądać kostkę danych. Operacja ta powoduje przetworzenie i zaktualizowanie całego projektu. Jest to wymagane, jeśli dokonamy jakiś zmian w jego strukturze.

5 Zaawansowane projektowanie kostek danych – kalkulacje

Kalkulacje poszerzają analityczne możliwości kostki danych [2]. Dzięki nim tworzone są dodatkowe miary definiowane przez użytkownika, do których później można się odwołać poprzez ich nazwę. Wszelkie kalkulacje lub, inaczej mówiąc, miary kalkulowane, korzystają z kolumn tabel faktów. Tworzenie kalkulacji odbywa się przy użycia specjalnego narzędzia, które dostępne jest po uruchomieniu narzędzia *Cube Designer* i wybraniu zakładki *Calculations - rysunek 1* .

Narzędzie to, przedstawione na rysunku 1, składa się z trzech części. W lewym górnym panelu o nazwie *Script Organizer* znajdują się zdefiniowane kalkulacje, zaraz pod nim w panelu *Calculation Tools*, który podzielony jest na trzy zakładki, znajdują się odpowiednio od lewej, miary i wymiary zdefiniowane w konkretnej kostce(zakładka *Metadata*), następnie dostępne funkcje(zakładka *Functions*), oraz gotowe szablony(zakładka *Templates*). Centralną część narzędzia zajmuje panel, w którym tworzone będą kalkulacje. Składa się on z dwóch typów widoku: widoku formy, który jest domyślnym widokiem oraz widoku

skryptu, przedstawionego na rysunku 18. Nasuwa się pytanie, dlaczego widoczny jest widok skryptu, jeżeli domyślnym i zaznaczonym widokiem jest ten drugi? Dzieje się tak tylko w przypadku, gdy zaznaczona jest domyślna kalkulacja w panelu *Script Organizer*. Kalkulacji tej nie wolno modyfikować, gdyż zależą od niej pozostałe kalkulacje.



Rys. 1. Narzędzie tworzenia miar kalkulowanych.

Narzędzie tworzenia nowych kalkulacji przeznaczone jest dla programistów i ma wbudowanych wiele funkcji ułatwiających pisanie kodu. Zaimplementowany został tutaj system *InteliSense*, który znany jest z projektów .NET. Występuje tutaj w zmodyfikowanej formie „przeciągnij i upuść", która pozwala przesuwać miarę lub wymiar w odpowiednie miejsce, zamiast ręcznie go wpisywać, co znacznie ułatwia pracę i pozwala uniknąć wielu potencjalnych błędów, zwłaszcza w połączeniu z wbudowaną funkcją sprawdzania poprawności składni. System ten jest dostępny zarówno w widoku formularza jak i w widoku skryptu.

W widoku skryptu mamy ponadto do dyspozycji kolejne narzędzie znane z projektów .NET, a mianowicie tryb debugowania, dzięki któremu możemy krok po kroku zobaczyć jak liczona jest kalkulacja. Panel *Debug* oferuje pięć zakładek, z których pierwsza od lewej o nazwie *Pivot Table*, umożliwia przeciąganie i upuszczanie obiektów z zakładki *Metadata*, oraz cztery dodatkowe zakładki MDX, w których możemy wykonywać zapytania diagnostyczne [1], [3]. Dzięki swej ogromnej funkcjonalności narzędzie debugowania powoli zaoszczędzić mnóstwo

czasu poświęconego na wyszukiwanie źródła błędu. Po stworzeniu nowej kalkulacji należy oczywiście wdrożyć projekt na serwer.

Tworzenie kalkulacji zaczyna się od nadania jej unikalnej nazwy- rysunek 2. Należy zwrócić uwagę, że domyślna nazwa kalkulacji ujęta jest w nawiasy kwadratowe. Powodem zastosowania nawisów jest dwuczłonowa nazwa miary kalkulowanej.

Pole *Parent hierarchy* służy do ustawienia, w jakiej hierarchii miara kalkulowana ma być przechowywana.

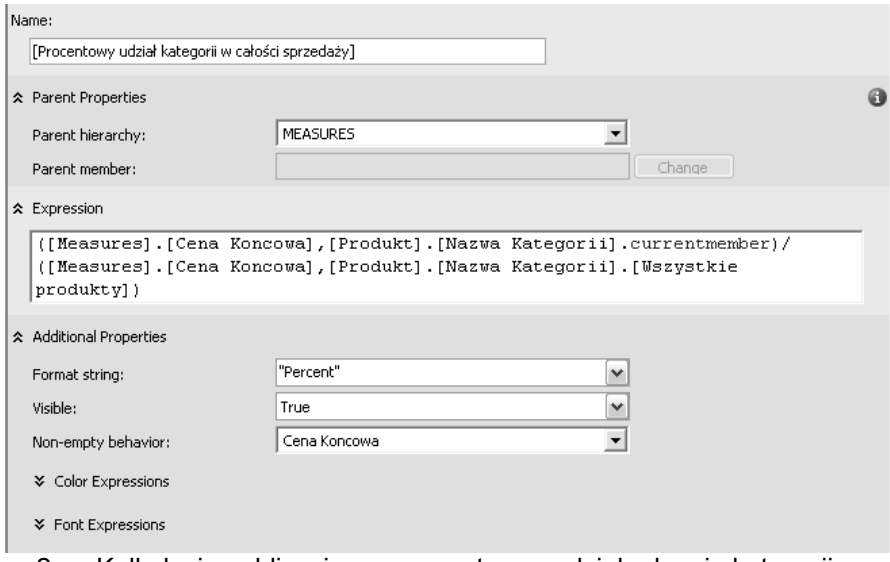
W pole *Expression* należy wpisywać ręcznie bądź przeciągać miary, na podstawie których miara kalkulowana będzie obliczana.

Pole *Format String* określa sposób prezentacji obliczonych danych.

Pole *Visible* określa czy miara kalkulowana ma być widoczna w kostce danych czy nie.

Pole *Non-empty behaviour* służy do obliczenia wartości każdej z wymienionych miar przed wyznaczeniem wartości wyrażenia. Jeżeli miara wymieniona w tym polu będzie pusta to wyrażenie zwróci wartość pustą i nie wygeneruje błędu.

Opcje *Color Expression* oraz *Font Expression* pozwalają na określenie odpowiednio koloru czcionki wyniku oraz tła komórki wyniku dla *Color Expression* oraz ustawienie rodzaju czcionki, jej wielkości i stylu dla *Font Expression*. Opcje te są dostępne po naciśnięciu prawym klawiszem myszy na nazwy obu wyrażeń.



Rys. 2. Kalkulacja obliczająca procentowy udział danej kategorii w sprzedaży.

6 Zaawansowane projektowanie kostek danych – wskaźniki KPI

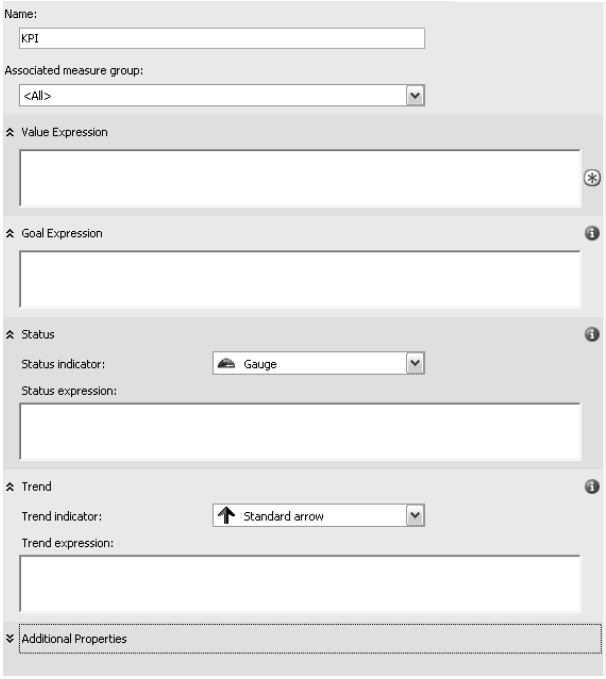
W świecie biznesu wizualizacja postępu przedsiębiorstwa na drodze do wyznaczonego celu staje się coraz istotniejszym elementem. Graficzna prezentacja wyników potrafi powiedzieć o wiele więcej niż suche dane liczbowe. Dlatego właśnie w ostatnich latach uwaga świata biznesowego skupia się na tzw. kluczowych wskaźnikach wydajności (ang. KPI – Key Performance Indicator) [2] [4].

Wskaźniki KPI budowane są przy użyciu języka MDX wykorzystując do tego celu stworzone miary, wymiary, czy miary kalkulowane. W oprogramowaniu *Analysis Services* wskaźniki KPI posiadają kilka wstępnie zdefiniowanych ikon, które w jasny, czytelny i jednoznaczny sposób potrafią poinformować użytkownika o charakterze danych przekazywanych przez nie. Wskaźniki KPI przyczyniają się do wzrostu efektywności przedsiębiorstwa, gdyż pozwalają analizować różne jego obszary oraz szybko reagować na niekorzystne procesy. Dzięki graficznej analizie możemy zwrócić szczególną uwagę na procesy wymagające wyspecjalizowanego sztabu ludzi potrzebnych do ich rozwiązania.

Do tworzenia KPI służy specjalne narzędzie, które znajduje się na zakładce KPI narzędzia *Cube Designer*. Składa się ono z trzech części. W lewym górnym rogu znajduje się panel o nazwie *KPI Organizer*, w którym wyświetlone są stworzone dla danej kostki wskaźniki. Panel ten pozwala na zmianę kolejności wskaźników. Z jego poziomu możemy także stworzyć i usunąć dany wskaźnik. Pod nim znajduje się panel o nazwie *Calculation Tools*, znany już z narzędzia do tworzenia miar kalkulowanych. Centralną część zajmuje okno projektowe, w którym tworzone są wskaźniki. Aby sprawdzić dostępne opcje projektowe należy wcześniej stworzyć nowy wskaźnik KPI. W tym celu przejdźmy na zakładkę KPI w kostce *Sprzedaz – rysunek 3*. Aby utworzyć nowy wskaźnik należy w panelu *KPI Organizer* nacisnąć prawy klawisz myszy i wybrać polecenie *New KPI* lub nacisnąć czwarty przycisk licząc od lewej strony. Okno projektowe powinno wyglądać jak na rysunku poniżej.

Składa się ono z kilku sekcji, na podstawie których budowany jest sam wskaźnik, jak i narzędzia graficzne go reprezentujące. Zaczynając od góry w pole *Name* należy wpisać nazwę nowego wskaźnika. Warto dodać, że nazwy składające się z więcej niż jednego wyrazu nie muszą być ujmowane w nawiasy kwadratowe tak jak miało to miejsce w przypadku miar kalkulowanych. Kolejne pole o nazwie *Assiciated measure group* ustawia domyślną grupę miar dla wskaźnika KPI. W pole *Value Expression* należy wpisać wyrażenie, które pozwoli uzyskać bieżący stan metryki biznesowej. Pole *Goal Expression* reprezentuje cele biznesowe. Pole z listą rozwijaną o nazwie *Status Indicator*

reprezentuje rodzaj graficznej reprezentacji wyników natomiast w pole *Status Expression* należy wpisać wyrażenie, na podstawie którego, graficzna reprezentacja wyników będzie się zmieniać - rysunek 4. Wyrażenie to musi zwracać wyniki z przedziału od -1 do 1, gdzie -1 oznacza niekorzystne na tle postawionych celów wartości, natomiast 1 pozytywne. Tak naprawdę przedział zwracanych wyników jest określony poprzez wybór graficznego wskaźnika. Dla sygnalizatora świetlnego możliwe wartości to -1, 0, 1.



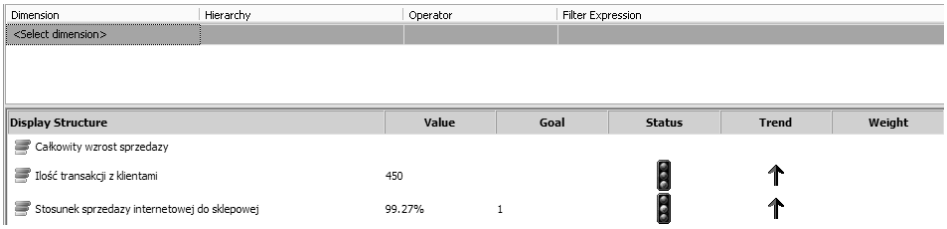
Rys. 3. Narzędzie tworzenia wskaźników KPI.

Pole z listą rozwijaną o nazwie *Trend Indicator* pozwala na określenie graficznej reprezentacji trendu. W pole *Trend Expression* należy wpisać wyrażenie, które porówna bieżącą wartość, zazwyczaj wyrażoną w polu *Value Expression* z tą samą wartością z innego okresu. Wyrażenie to musi zwracać wartość z przedziału od -1 do 1. Sekcja *Additional Properties* zawiera dodatkowe opcje, które nie będą używane w projekcie. Wartość wyrażenia w polach *Status Expression* oraz *Trend Expression* liczona jest poprzez wyrażenie MDX, co zostanie pokazane w dalszej części tego rozdziału.

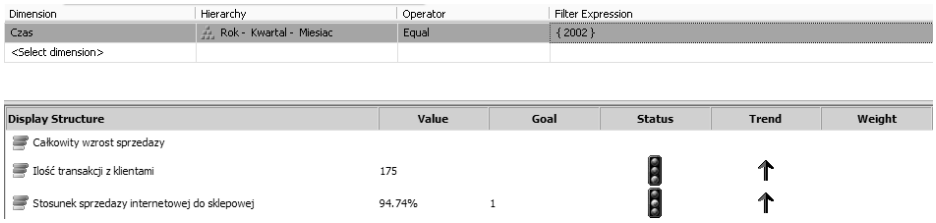
Narzędzie projektowe pozwala także sprawdzić poprawność składni poprzez pierwszy przycisk z prawej strony o nazwie *Check Syntax*.

Stworzone i wdrożone na serwer wskaźniki można testować dzięki wbudowanemu narzędziu. Wystarczy nacisnąć drugi przycisk z prawej strony  , aby przełączyć się do widoku przeglądania. Okno przeglądania składa się z dwóch paneli, z których dolny wyświetla stworzone wskaźniki wraz z ich graficzną reprezentacją oraz panel górny, który służy do ewentualnego zawężania zakresów analizy kostki danych - rysunek 5 i 6.

Rys. 4. Przykładowy wskaźnik KPI.



Rys. 5. Widok przeglądarki stworzonych wskaźników KPI bez zawężania zakresu analiz.



Rys. 6. Widok przeglądarki stworzonych wskaźników KPI z zawężeniem zakresu analiz.

Analiza kodu stworzonych wskaźników pozwala dostrzec pewną prawidłowość. Otóż kod użyty do obliczenia wyrażeń w przypadku wszystkich stworzonych wskaźników jest praktycznie identyczny. Różni się tylko miarą użytą do obliczania danego wskaźnika. Dodatkowo nie ma znaczenia czy w sekcji statusu używamy funkcji czy czystych zapytań MDX. Istotnym szczegółem jest natomiast unikanie mieszania w wyrażeniu obliczonej miary kalkulowanej oraz jej definicji. Może to prowadzić do błędów obliczeń, które w wypadku odpytywania wskaźników KPI z użyciem oprogramowania *BIDS*, uniemożliwiają wyświetlenie, bądź prawidłowe obliczenie wartości wyrażenia. Warto dodać, że w pola celu, statusu oraz trendu można także wpisywać stałe wartości liczbowe, ale dzięki formule MDX zyskuje ogromną swobodę operowania istotnymi z biznesowego punktu widzenia wartościami.

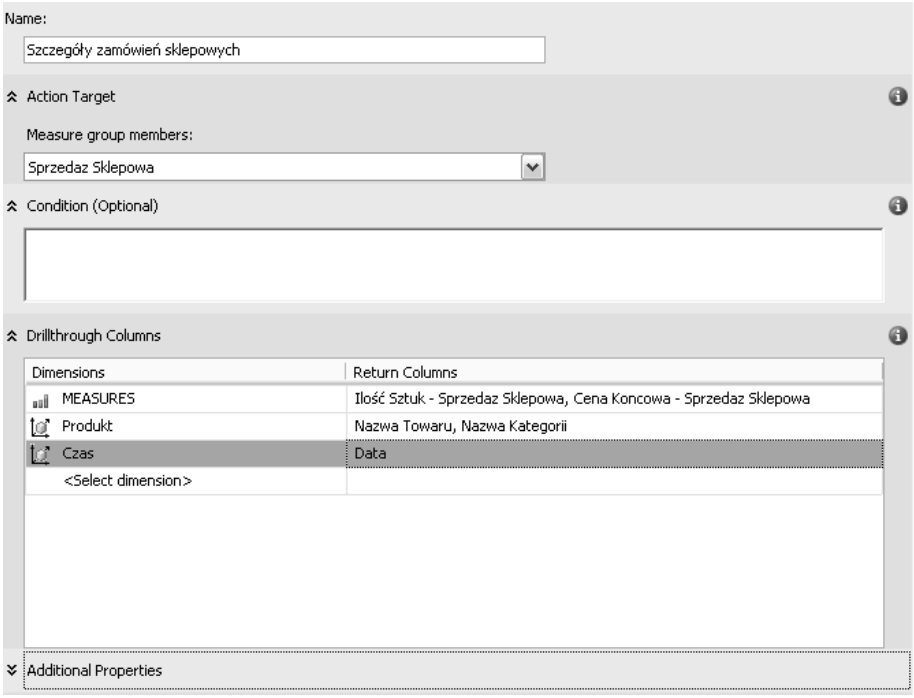
7 Zaawansowane projektowanie kostek danych – akcje zgłębiania danych

Akcje umożliwiają dodanie do kostki danych zadań, zwanych akcjami, które później mogą być wykonywane przez użytkowników. Kostka danych oferuje zatem dodatkowy mechanizm, który pozwala użytkownikom uzyskiwać dodatkowe informacje oraz podejmować właściwe czynności na podstawie uzyskanych informacji. Należy pamiętać, że akcja jest zawsze inicjowana przez użytkownika i związana z konkretnym obiektem kostki danych [2]. Obiekt, który może być konkretną komórka lub członkiem wymiaru, zostaje użyty jako parametr akcji. Warto zaznaczyć, że akcje nie działają ze wszystkimi aplikacjami klienckimi, dlatego przed ich definiowaniem warto sprawdzić, czy będą obsługiwane przez daną aplikację kliencką.

Istnieją trzy rodzaje akcji, które możemy zdefiniować w kostce analitycznej. Akcje typu URL, które pozwalają na przechodzenie do zdefiniowanych witryn sieci Web, np. jeśli chcemy uzyskać dodatkowe informacje na temat jakiegoś produktu lub producenta. Akcje raportujące, które w połączeniu z usługą *Reporting Services*, pozwalają

łączyć raporty z obiektami kostki danych oraz akcje drążenia danych, które pozwalają uzyskać szczegółowe informacje danych, które nie zawsze chcemy przechowywać w kostce danych.

Aby dodać nową akcję należy w narzędziu projektanta kostki przejść na zakładkę *Actions*. Budowa narzędzia jest podobna jak w przypadku KPI – rysunek 7. W górnym lewym panelu znajdują się już zdefiniowane akcje. Panel pozwala na zmianę ich kolejności oraz dodawanie i usuwanie. Pod nim znajduje się panel, który dostarcza niezbędnych miar, wymiarów i funkcji wbudowanych potrzebnych do utworzenia akcji. Centralną część narzędzia stanowi formularz, który jest pusty do momentu utworzenia akcji. Warto zwrócić uwagę, że w odróżnieniu od miar kalkulowanych oraz wskaźników KPI dla akcji jest dostępny tylko jeden tryb widoku, mianowicie tryb formularza.



Rys. 7. Kreator tworzenia akcji drążenia danych.

Powyższy rysunek przedstawia poprawnie stworzoną akcję drążenia danych. Akcja ta dotyczyć będzie sprzedaży sklepowej towarów, co uwzględnione jest wyborem w polu Measure group members grupy miar Sprzedaz Sklepowa. W polu Drillthrough Column ustawiane są kolumny, których dane będą wyświetlone przy wywołaniu akcji. Należy wybrać pole zaznaczone na powyższym rysunku.

Stworzona akcja po uruchomieniu wyświetli szczegółowe dane związane ze sprzedażą sklepową, takie jak ilość sztuk zamawianego towaru, jego cena sprzedaży, nazwa, kategoria, do której należy oraz data zamówienia. Akcje widoczne są w menu podręcznym narzędzia. Przekonamy się, ze zdefiniowana tutaj akcja będzie widoczna w oknie przeglądarki kostek danych tylko wtedy, kiedy będziemy mieli do czynienia z miarami z grupy Sprzedaż sklepowa.

8 Język MDX

Język MDX został stworzony do zapytań w środowisku wielowymiarowych baz danych. Pokazane tutaj konstrukcje nie są wyjątkowo skomplikowane, a oferują spore możliwości, co pokazuje jak ogromny potencjał drzemie w języku MDX. Mamy możliwość tworzenia nie tylko zapytań wybierających, ale także modyfikujących dane. Warto dodać,

że język MDX nie jest charakterystyczny tylko dla produktów firmy Microsoft, lecz jest częścią specyfikacji technologii OLAP. Oznacza to, że do odpytywania kostek danych możemy używać nie tylko produktów firmy Microsoft, lecz także produktów innych dostawców.

Tworzenie zapytań w języku MDX jest znacznie łatwiejsze niż w SQL, dzięki istnieniu metadanych. Metadane to informacje o sposobie przechowywania i strukturze danych oraz ich znaczeniu [2], [5]. Tworzenie kostki danych definiuje nie tylko miary, ale także sposoby ich agregowania. Możemy ustalić czy dane atrybuty w ramach wymiaru są grupowalne oraz czy mogą być łączone w hierarchie. Jeżeli w kostce danych OLAP producent został zdefiniowany jako atrybut nadrzędny towaru, to zapytanie, które ma wyświetlić wszystkie towary produkowane przez producenta X, można określić przy pomocy wyrażenia [X].Children.

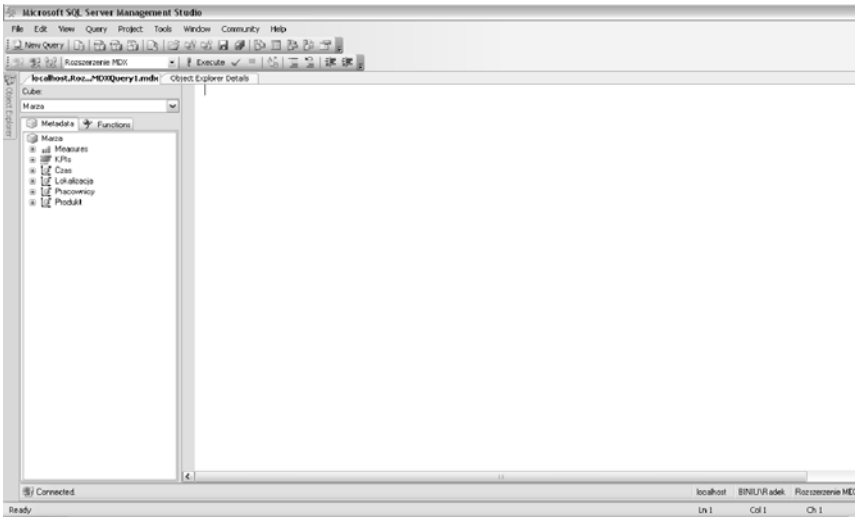
Tworzone w analitycznej kostce danych formuły, podobnie jak formuły znane z arkuszy kalkulacyjnych, pozwalają, podczas procesu ich tworzenia, na odwołanie się do dowolnej komórki kostki, nie interesując się sposobem, w jakim obliczono jej wartość. Język MDX daje możliwość tworzenia formuł, które znacznie przewyższają swoimi możliwościami formuły arkuszy kalkulacyjnych, gdyż:

- Odwołania do formuł mogą mieć nazwy opisowe,
- Kostki danych są wielowymiarowe, w odróżnieniu od arkuszy kalkulacyjnych, które są dwuwymiarowe, co pozwala w łatwy sposób na odwołania tej samej formuły do wielu wymiarów,
- Język MDX upraszcza korzystanie z wielu wymiarów, dzięki zastosowaniu koncepcji „bieżącego członka” [2], tzn. w taki sam sposób, w jaki kopiowana formuła używa wartości z bieżącego

arkusza, tak formuła MDX automatycznie odwołuje się do bieżącego wymiaru.

- W odróżnieniu od arkuszy kalkulacyjnych, których formuły mogą odwoływać się do wartości znajdujących się w tym samym obszarze roboczym, formuły MDX mogą korzystać z każdej wartości znajdującej się w dowolnym miejscu przestrzeni kostki danych [2].

Pokazanie możliwości języka MDX zostanie zaprezentowane przy użyciu narzędzia należącego do pakietu SQL Server 2005. Należy uruchomić program *SQL Server Management Studio*, w oknie logowania należy wybrać usługę *Analysis Services* oraz w pole *Server name* wpisać localhost. Po naciśnięciu przycisku *Connect* zostaniemy podłączeni do usługi analitycznej naszego komputera. Następnie należy rozwinąć menu górne *File→New* i wybrać polecenie *Analysis Services MDX Query* – rysunek 8. Przed rozpoczęciem tworzenia zapytań, bardzo istotne jest, aby wybrać odpowiednią bazę danych, na której chcemy pracować.



Rys. 8. Widok okna tworzącego zapytania MDX.

Panel ten umożliwia wybranie kostki danych oraz wyświetlenie obiektów z nią związanych w zakładce *Metadata*. Zakładka *Functions* jest stała, niezależnie od wybranej kostki, i zawiera funkcje wbudowane, które pomagają w tworzeniu zapytań MDX. Wszelkie obiekty oraz funkcje znajdujące się w tym panelu można przeciągać do panelu centralnego,

w którym pisany jest kod zapytania.

Wykonywanie zapytań odbywa się poprzez naciśnięcie przycisku Execute. Istnieje możliwość sprawdzenia poprawności składni zapytania, co umożliwia przycisk, znajdujący się po prawej stronie od przycisku *Execute,* o nazwie *Parse* .

Po wykonaniu zapytania panel główny zostanie podzielony na pół i w jego dolnej części, na zakładce *Result*, zobaczymy wynik zapytania.

Składania zapytania wybierającego w języku MDX jest bardzo podobna do składni zapytania w języku SQL, z tą różnicą, że mając do czynienia w bazami wielowymiarowymi, jesteśmy zmuszeni do umieszczania konkretnych wymiarów na osiach. Do dyspozycji są 63 osie, z których kilka pierwszych ma określoną nazwę. Klauzula FROM określa w tym wypadku nazwę kostki danych, z której dane mają być pobierane.

Najprostsze możliwe zapytanie w języku MDX przedstawione jest poniżej:

```
SELECT
FROM Sprzedaz
```

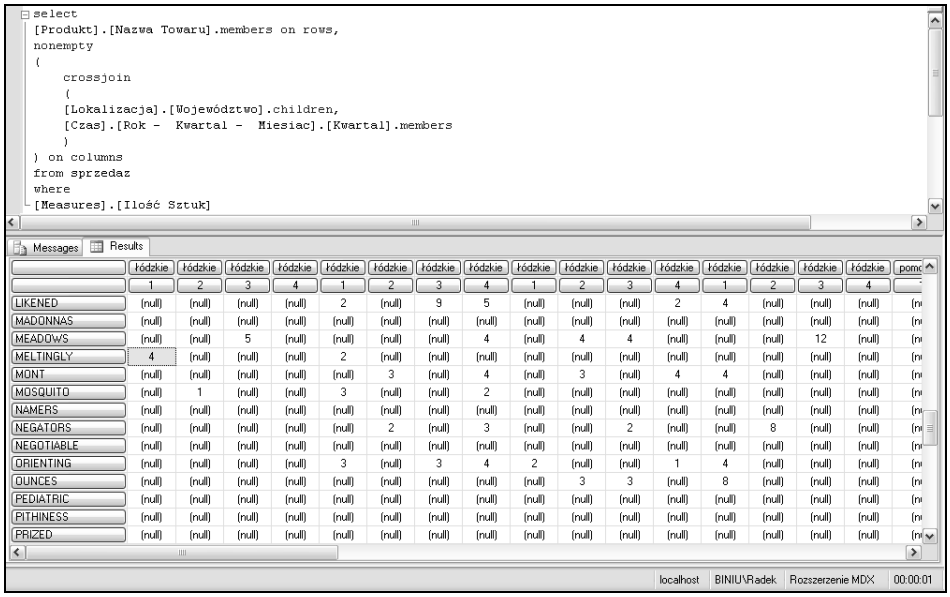
Zapytanie to zwróci pojedynczą wartość: 1686. Jest to suma całkowitej ilość sztuk sprzedanych towarów. Z uwagi na fakt, że domyślna miara nie została zdefiniowana, język MDX pobierze pierwszą miarę zdefiniowaną w oknie *Cube Structure* dla kostki *Sprzedaz* – rysunek 9.

W zapytaniu dodatkowo nie zdefiniowano żadnego członka wymiaru, dlatego został użyty domyślny członek każdej hierarchii atrybutu, czyli element *All*.

Jak widać na powyższym zapytanie to zawiera kilka funkcji, które należy wyjaśnić. Zaczniemy od funkcji *members.* Funkcja ta zwraca wszystkich członków należących do wymiaru, poziomu lub hierarchii łącznie z domyślnym członkiem *All*. Warto zaznaczyć, że funkcja ta zastosowana do wymiaru zwróci wszystkich jego członków. W przypadku hierarchii funkcja ta zwróci wszystkich członków danej hierarchii przechodząc przez wszystkie jej poziomy, natomiast zastosowana do określonego poziomu, zwróci członków znajdujących się na tym poziomie. Funkcja *children* zwraca wszystkie „dzieci" dla określonego członka. W przypadku, gdy dany element nie ma żadnych elementów podrzędnych, funkcja zwróci pustą wartość.

W przypadku pokazanym na powyższym rysunku, zostaną wypisane wszystkie „dzieci" członka *Województwo,* czyli innymi słowy wszystkie nazwy województw. Funkcja *children* nie może być zastosowana do poziomów hierarchii.

Funkcja *crossjoin* łączy ze sobą dwa zbiory podane jako jej argumenty w jeden zbiór. Łączenie elementów jest krzyżowe, tzn. każda pozycja z pierwszego argumentu jest łączona z każdą pozycją z drugiego. Funkcja ta nie może być zastosowana dwa razy do tego samego wymiaru.



Rys. 9. Zapytanie MDX pokazujące ilość sprzedanych sztuk towarów w danym województwie na przestrzeni lat.

Ostatnia zastosowana funkcja to funkcja *nonempty*. Jej zadaniem jest wyeliminowanie niepowiązanych ze sobą rekordów. Przykład bardziej zaawansowanego zapytania MDX przedstawiony jest poniżej.

```
    SELECT
{
      [Produkt].[Nazwa Kategorii].&[EXPLOITATIONS],
      [Produkt].[Nazwa Kategorii].&[THAN]
} ON ROWS,
CROSSJOIN
(
      CROSSJOIN
      (
         FILTER
         (
            [Czas].[Rok].CHILDREN,
            [Czas].[Rok].CURRENTMEMBER.Name="2003"
         ),
         FILTER
         (
```

```
                [Lokalizacja].[Miasto].CHILDREN,

      [Lokalizacja].[Miasto].CURRENTMEMBER.Name="ŁÓDŹ"
OR

      [Lokalizacja].[Miasto].CURRENTMEMBER.Name="ZGIERZ"
            )
         ),
         {
            [Measures].[Marza - Sprzedaz Internetowa],
            [Measures].[Marza - Sprzedaz Sklepowa],
            [Measures].[Stosunek marzy internetowej do
sklepowej]
         }
) ON COLUMNS
FROM marza
```

Za pomocą języka MDX możemy także odpytywać wskaźniki KPI. Schemat zapytania dla wskaźników KPI jest taki sam, zmienia się tylko nazwa oraz ewentualnie warunki filtrowania danych. Poniżej znajduje się przykład zapytania, które odpyta stworzony wskaźnik KPI.

```
      SELECT
{
      KPIValue("Całkowity wzrost sprzedazy"),
      KPIGoal("Całkowity wzrost sprzedazy"),
      KPIStatus("Całkowity wzrost sprzedazy"),
      KPITrend("Całkowity wzrost sprzedazy")
}
on COLUMNS
FROM Sprzedaz
```

Tak zbudowane zapytanie zwróci poprawnie wszystkie wartości ze wskazanego wskaźnika co pokazane jest na rysunku 10.

Całkowity wzrost sprzedaży	Całkowity wzrost sprzedazy Goal	Całkowity wzrost sprzedazy Status	Całkowity wzrost sprzedazy Trend
(null)	0.5	-1	1

Rys. 10. Wyniki odpytania wskaźnika KPI Całkowity wzrost sprzedazy.

Jak widać na powyższym rysunku zwrócone zostały bieżące wartości wskaźnika, założony cel biznesowy, status oraz trend. Bieżąca wartość jest pusta, gdyż kalkulacja użyta do jej obliczenia porównuje sprzedaż roku bieżącego z poprzednim. Ponieważ nie zdefiniowano, dla jakiego roku wskaźnik jest obliczany, został wzięty pod uwagę pierwszy rok, który nie ma swojego poprzednika. W związku z tym została zwrócona pusta wartość. Pozostałe wartości są jednak policzone zgodnie z zaprojektowanym schematem.

```
SELECT
```

```
{
     KPIValue("Całkowity wzrost sprzedazy"),
     KPIGoal("Całkowity wzrost sprzedazy"),
     KPIStatus("Całkowity wzrost sprzedazy"),
     KPITrend("Całkowity wzrost sprzedazy")
}
ON COLUMNS
FROM Sprzedaz
WHERE
[Czas].[Rok - Kwartal - Miesiac].[Rok].&[2002]
```

Wynik przedstawionego i omówionego zapytania przedstawia rysunek 11.

Całkowity wzrost sprzedaży	Całkowity wzrost sprzedazy Goal	Całkowity wzrost sprzedazy Status	Całkowity wzrost sprzedazy Trend
114.13%	0.5	1	-1

Rys. 11. Wynik odpytania wskaźnika Całkowity wzrost sprzedaży dla roku 2002.

Zastosowane filtrowanie pozwoliło na uzyskanie bieżącej wartości całkowitego wzrostu sprzedaży.

Język MDX pozwala także na tworzenie oraz usuwanie miar kalkulowanych. Tworzenie miar może odbywać się w dwojaki sposób. Miara kalkulowana może być stworzona w zasięgu zapytania lub w zasięgu sesji. Kalkulacja stworzona w zasięgu zapytania jest automatycznie usuwana po jego zakończeniu, natomiast w zasięgu sesji istnieje do momentu jej zakończenia. Jeśli podczas tworzenia miary kalkulowanej nie określimy z góry jej zasięgu, zostanie ona domyślnie stworzona w zasięgu sesji. Aby utworzyć nową miarę kalkulowaną należy wydać następujące polecenie:

```
   CREATE MEMBER Marza.[nowa miara kalkulowana]
AS
[Measures].[Marza]-[Measures].[Marza - Sprzedaz
Internetowa]
```

Aby usunąć, stworzona przed chwilą, miarę kalkulowaną należy wydać w oknie jej bieżącej sesji polecenie:

```
DROP MEMBER Marza.[nowa miara kalkulowana]
```

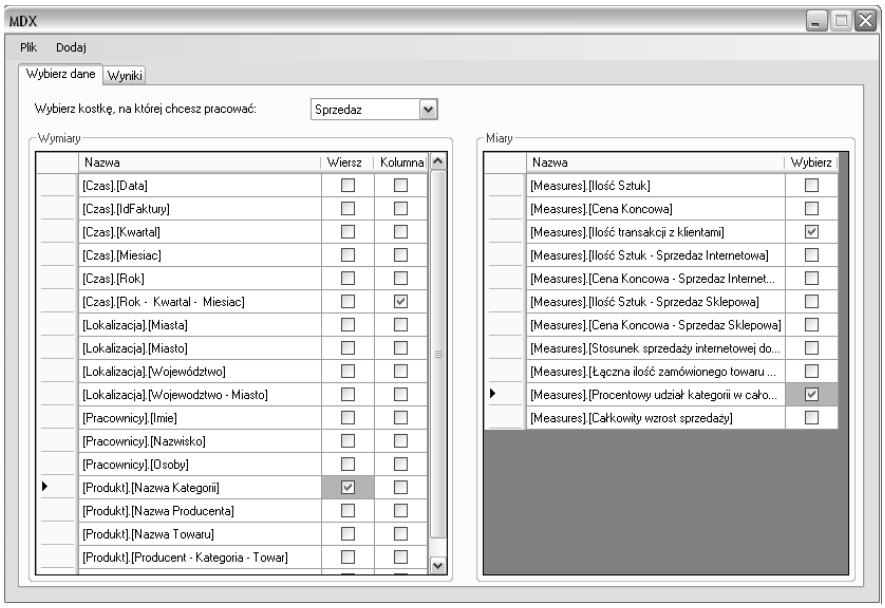
Polecenie to usunie miarę kalkulowaną o nazwie *nowa miara kalkulowana* z kostki danych *Marza* przed zakończeniem danej sesji.

9 Aplikacja w technologii OLAP

Nie ulega wątpliwości, że kostka analityczna może być najbardziej użyteczna, kiedy jej możliwości zostaną zintegrowane z używaną aplikacją [1]. Daje to ogromne możliwości dostosowywania aplikacji dla potrzeb konkretnego klienta.

Program został napisany w *Visual Studio 2005* i jego zadaniem było odpytywanie wybranej przez użytkownika kostki danych oraz tworzenie miar kalkulowanych o zasięgu sesyjnym.

Przy użyciu biblioteki o nazwie *Microsoft.Analysis Services.AdomdClient* do aplikacji pobierane są kostki danych, wszystkie związane z nimi wymiary oraz miary, a także wykonywane zapytania. Biblioteka ta jest częścią ADOMD.NET, która została zaprojektowana, aby ułatwić tworzenie aplikacji klienckich zawierających rozwiązania analityczne. Biblioteka ta jest instalowana jako jeden z komponentów SQL Server 2005, ale równie dobrze można ją pobrać z Internetu. Aby móc korzystać z metod w niej zawartych, należy do swojej aplikacji dodać odpowiednią referencję oraz dodać do przestrzeni nazw odpowiednią komendę.



Rys. 12. Główne okno aplikacji.

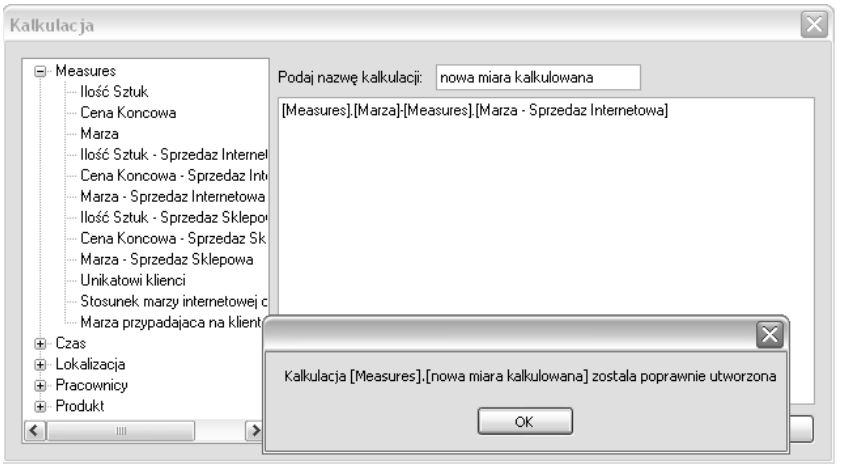
Aplikacja składa się z dwóch zakładek. Na pierwszej z nich, widocznej na rysunku 12, po wyborze kostki danych dostępnej w ramach stworzonych projektów, mamy możliwość wyboru parametrów zapytania. W dwóch panelach o nazwach odpowiednio *Wymiary* oraz *Miary*,

pojawiać się będą zdefiniowane w ramach danej kostki analitycznej wszelkie wymiary oraz miary, łącznie z miarami kalkulowanymi.

Panel *Wymiary* umożliwia wybór atrybutów, które mają być wyświetlane w wierszach oraz kolumnach danego zapytania, poprzez zaznaczenie checkboxów znajdujących się w odpowiednich kolumnach. Panel *Miary*, podobnie jak poprzedni, umożliwia wybór dostępnych miar, na podstawie których zbudowane będzie zapytanie.

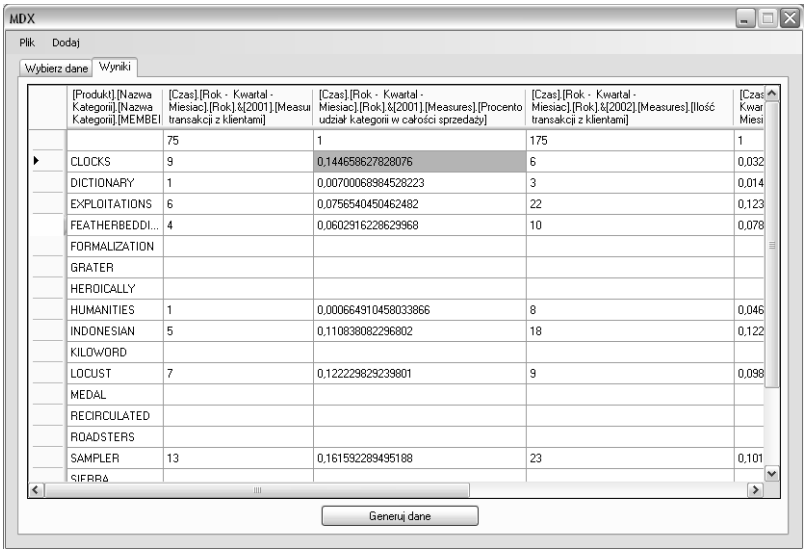
Na zakładce *Wyniki* będą znajdować się rezultaty stworzonych zapytań, które pojawią się po naciśnięciu przycisku *Generuj dane*. Należy zaznaczyć, iż z uwagi na użytą kontrolkę, będziemy musieli ograniczyć się w budowie zapytań do łącznie trzech członków wymiarów, oraz do dwóch miar. Jest to niezbędne z uwagi na ograniczenia użytej kontrolki.

Po wyborze kostki danych mamy możliwość utworzenia w niej dodatkowej miary kalkulowanej, której istnienie ograniczone jest do czasu działania aplikacji. Panel tworzenia nowej kalkulacji dostępnych jest poprzez wybranie z menu górnego *Dodaj* polecenia *Kalkulacja*. Poniżej przedstawione jest okno tworzenia nowej miary kalkulowanej.



Rys. 13. Tworzenie nowej miary kalkulowanej z poziomu aplikacji.

Po naciśnięciu przycisku OK, nowa kalkulacja jest dodawana do istniejącej kostki danych i można z niej korzystać przy dalszych zapytaniach - rysunki 13 i 14. Jej czas istnienia jest jednak ograniczony do bieżącego połączenia, więc przy ponownym uruchomieniu aplikacji, miara kalkulowana o nazwie *nowa miara kalkulowana* nie będzie istniała w danej kostce analitycznej.



[Produkt].[Nazwa Kategorii].[Nazwa Kategorii].[MEMBEI	[Czas].[Rok - Kwartał - Miesiąc].[Rok].&[2001].[Measu transakcji z klientami]	[Czas].[Rok - Kwartał - Miesiąc].[Rok].&[2001].[Measures].[Procento udział kategorii w całości sprzedaży]	[Czas].[Rok - Kwartał - Miesiąc].[Rok].&[2002].[Measures].[Ilość transakcji z klientami]	[Czas Kwar Miesi
	75	1	175	1
CLOCKS	9	0,144658627828076	6	0,032
DICTIONARY	1	0,00700068984528223	3	0,014
EXPLOITATIONS	6	0,0756540450462482	22	0,123
FEATHERBEDDI...	4	0,0602916228629968	10	0,078
FORMALIZATION				
GRATER				
HEROICALLY				
HUMANITIES	1	0,000664910458033866	8	0,046
INDONESIAN	5	0,110838082296802	18	0,122
KILOWORD				
LOCUST	7	0,122229829239801	9	0,098
MEDAL				
RECIRCULATED				
ROADSTERS				
SAMPLER	13	0,161592289495188	23	0,101
SIERRA				

Rys. 14. Wyniki odpytywania kostki danych przy użyciu aplikacji.

Podsumowanie i wnioski

Celem pracy było pokazanie poszczególnych etapów tworzenia hurtowni danych, zarządzania nią i dostosowywania jej do indywidualnych potrzeb użytkownika. Dlatego w pracy zbudowano struktury wielowymiarowe oparte na dwóch kostkach analitycznych, pozwalające na szeroką analizę danego przedsiębiorstwa. Stworzone dodatkowo miary kalkulowane pozwoliły na rozszerzenie możliwości analizy danych. Zbudowane, na podstawie dostarczonych informacji, wskaźniki KPI, dzięki wizualnej reprezentacji wyników, pozwoliły mierzyć postęp przedsiębiorstwa na drodze do wyznaczonego celu. Zaprojektowana akcja drążenia danych pozwoliła uzyskać jeszcze bardziej szczegółowe dane. Pokazane w artykule przykładowe sposoby odpytywania stworzonych kostek danych, pozwoliły na wyłuskanie najważniejszych, z biznesowego punktu widzenia, informacji. Wreszcie opracowana końcówka klienta, dzięki zastosowaniu powszechnie dostępnej biblioteki ADOMD.NET, pozwoliła na stworzenie aplikacji, która dostosowana do potrzeb konkretnego klienta, umożliwiła dostarczenie najważniejszych informacji o przedsiębiorstwie.

Literatura

[1] Andrew J. Burst, Stephen Forte. *Programowanie Microsoft SQL Server 2005*. Microsoft Press, 2006

[2] Reed Jacobson, Stacia Misner. Microsoft *SQL Server 2005 Analysis Services Krok po kroku.* Microsoft Press, 2006

[3] Eric L. Brown. SQL *Server 2005 Wyciśnij wszystko.* Helion, 2007

[4] Tomasz Skurniak. Język *MDX i obliczenia w kostkach analitycznych*. http://www.microsoft.com/poland/technet/article/art0064_01.mspx

[5] William Pearson. MDX *Member Functions: The "Family" Functions*. http://www.databasejournal.com/features/mssql/article.php/10894_2168911_3/MDX-Member-Functions--The-Family-Functions.htm

[6] Wojciech Kowasz. Rozwiązania Business *Intelligence i hurtownie danych w Microsoft SQL Server 2005.* http://wss.pl/Articles/6371.aspx

IMPLEMENTATION OF MDX EXTENSION FOR WAREHOUSING ON MS SQL SERVER PLATFORM

Summary – The paper widely described multidimensional extension of SQL query language MDX SQL. It can be used for structures creation and data processing for data warehouse at Microsoft SQL Server platform. First part of this work concentrated on methods of creation using visual Analysis Services built in tools. Special attention was dedicated to Key Performance Indicators and Actions creation methods. Both structures was described in very precise way. The second part is dedicated to examples of MDX queries with special attention for presentation of KPI values. The last part described functionality of created .NET client application, which give opportunity to create, testify and process MDX queries. The ADOMD library use in client program was shortly described. Some exemplary results was presented.