

# Czy warto już stosować magazyny danych czyli dlaczego tak wiele analiz OLAP jest „błędnych”

Tadeusz Morzy  
Instytut Informatyki Politechniki Poznańskiej  
60-965 Poznań  
ul. Piotrowo 3A  
morzy@put.poznan.pl

## Abstrakt.

Magazyny danych stanowią obecnie jedną z podstawowych technologii tworzenia zintegrowanego systemu informatycznego przedsiębiorstwa umożliwiając integrację danych przechowywanych w różnych heterogenicznych i autonomicznych systemach informatycznych funkcjonujących w danym przedsiębiorstwie. Zasadniczym celem integracji danych jest umożliwienie wielowymiarowych analiz i porównań danych przechowywanych w magazynie danych. Analiza magazynu danych polega na obliczaniu agregatów (sum, średnich, itp.) dla zadanych przez użytkownika „wymiarów” magazynu takich jak: czas, miejsce, klasyfikacja produktów, itp.). Komercyjnie dostępne magazyny danych zakładają, że wymiary magazynu danych są stałe i wzajemnie ortogonalne. W rzeczywistości, wymiary nie są stałe i również ewoluują: struktura organizacyjna przedsiębiorstwa się zmienia, pewne produkty znikają z rynku, pojawiają się nowe produkty. Zmieniają się również miary i ich relacje do wymiarów. To powoduje, że szereg analiz wykonywanych w dłuższych przedziałach czasowych, w których miały miejsce zmiany w odniesieniu do wymiarów i miar, generują niepoprawne raporty. Celem tego referatu jest przedstawienie problemów związanych z analizą magazynu danych, które pojawiają się w przypadku zmian zachodzących w schemacie lub strukturze magazynu danych.

## 1. Wprowadzenie

Magazyny danych stanowią obecnie jedną z podstawowych technologii tworzenia zintegrowanych systemów informatycznych przedsiębiorstw umożliwiając integrację danych przechowywanych w różnych heterogenicznych i autonomicznych systemach informatycznych funkcjonujących w danym przedsiębiorstwie. Zasadniczym celem integracji danych jest umożliwienie wielowymiarowych analiz i porównań danych dla potrzeb wspomagania procesów podejmowania decyzji w danym przedsiębiorstwie. Dla potrzeb tej analizy opracowano nowy model przetwarzania danych nazwany „przetwarzaniem analitycznym on-line” OLAP (ang. On Line Analytical Processing). OLAP ma za zadanie wspieranie procesów analizy magazynów danych dostarczając narzędzi umożliwiających analizę magazynu w wielu „wymiarach” definiowanych przez użytkowników (czas, miejsce, klasyfikacja produktów, itp.). Analiza magazynu polega na obliczaniu agregatów dla zadanych „wymiarów” magazynu. Typowym przykładem takiej analizy jest zapytanie o sprzedaż produktów w supermarkecie w kolejnych kwartałach, miesiącach, tygodniach, itp., zapytanie o sprzedaż produktów z podziałem na rodzaje produktów (AGD, produkty spożywcze, kosmetyki, itp.), czy wreszcie zapytanie o sprzedaż produktów z podziałem na oddziały supermarketu. Odpowiedzi na powyższe zapytania umożliwiają decydującym określić wąskie gardła sprzedaży, produktów przynoszących deficyt, itp., oraz podjęcie odpowiednich działań poprawiających sytuację. Jednym z podstawowych wymiarów analizy zawartości magazynu danych jest wymiar czasu. Przykładem analizy wykonanej wzdłuż wymiaru czasu jest wspomniane już zapytanie o wielkość

sprzedaży w supermarkecie w kolejnych latach, kwartałach, miesiącach. Wymiar czasu zasadniczo odróżnia systemy OLAP od systemów OLTP. Klasyczna definicja bazy danych mówi, że „baza danych jest abstrakcyjnym [informacyjnym] odzwierciedleniem wybranego fragmentu rzeczywistości nazywanego *miniświatem*”. W przypadku systemów OLTP, stan bazy danych odzwierciedla stan *miniświata* związany z jednym pojedynczym punktem na osi czasu. W przypadku magazynów danych i systemów OLAP, gromadzimy dane związane z kolejnymi punktami na osi czasu. Systemy OLAP są więc z definicji przeznaczone do przetwarzania danych, które zmieniają się w czasie (np. wielkość sprzedaży). Użytkownik magazynu danych zakłada, że magazyn danych i związany z nim system OLAP jest przygotowany na zmiany, które zachodzą w integrowanych systemach informatycznych. Niestety, nie jest to prawda. Okazuje się, że aktualnie komercyjnie dostępne systemy OLAP nie są przygotowane na przypadek ewolucji wymiarów analizy oraz modyfikacji odnoszących się do związków pomiędzy miarami a wymiarami analizy. Wynika to z założenia, które przyjmuje większość systemów magazynów danych, że wymiary magazynu danych są stałe i wzajemnie ortogonalne. W rzeczywistości, wymiary nie są stałe i również ewoluują: struktura organizacyjna przedsiębiorstwa się zmienia, pewne produkty znikają z rynku, pojawiają się nowe produkty. Zmieniają się również miary i ich relacje do wymiarów. To powoduje, że szereg analiz wykonywanych w dłuższych przedziałach czasowych, w których miały miejsce zmiany w odniesieniu do wymiarów i miar, generuje niepoprawne raporty. Przykładowo, jeżeli analizujemy i porównujemy PKB w krajach Europy w ostatnich 20 latach, to należy być świadomym faktu połączenia Niemiec, podziału Czechosłowacji na Czechy i Słowację, pojawienia się nowych państw na mapie Europy, w przeciwnym razie można wyciągnąć błędne wnioski (trudno wytłumaczyć skok PKB w Niemczech w okresie 1989-1990).

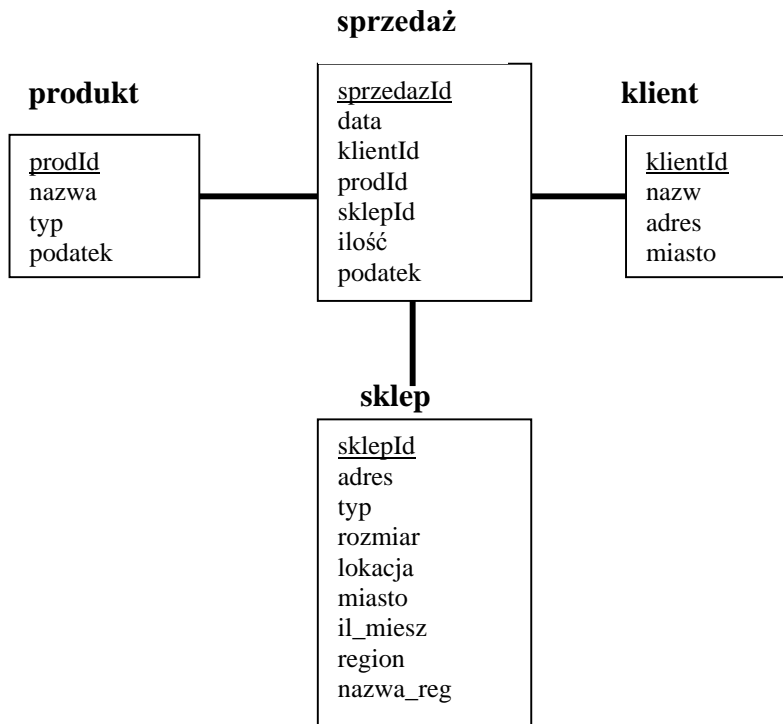
W artykule krótko przedstawiono problemy związane z ewolucją wymiarów i schematów magazynów danych, i ich konsekwencjami w odniesieniu do analiz OLAP.

Struktura artykułu jest następująca.. W rozdziale 2 przypomniano podstawowe pojęcia z zakresu modelu logicznego magazynu danych takie jak miara, wymiary i hierarchia wymiarów. W rozdziale 3 krótko przedstawiono i omówiono problem ewolucji schematów logicznych magazynów danych i konsekwencji takiej ewolucji na kilku wybranych przykładach. W rozdziale 4 krótko przedstawiono koncepcję temporalnych wielowersyjnych magazynów danych będącą odpowiedzią na wady klasycznych jednowersyjnych systemów OLAP. Rozdział 5 zawiera krótkie podsumowanie referatu.

## 2. *Miary i wymiary w modelu OLAP*

Podstawowym pojęciem schematu pojęciowego systemu OLAP jest **miara**, która ma charakter liczbowy. Miarą może być ilość sprzedanych produktów, zysk ze sprzedaży produktów, średnia ocen studentów, średnie zarobki, itd. Z każdą miarą związany jest **zbiór wymiarów**, od których zależy wartość danej miary (np. ilość sprzedanych produktów w zależności od produktu, czasu sprzedaży i miejsca sprzedaży - wymiarami są: produkt, lokalizacja i czas). Relację, która wiąże wymiary z miarą (zbiorem miar) nazywamy **tablicą faktów**. Informacja o wymiarach jest reprezentowana przez zbiór tablic nazywanych **tablicami wymiarów**. Z każdym wymiarem jest związany zbiór atrybutów. Najczęściej, atrybuty opisujące pojedynczy wymiar tworzą **hierarchię** nazywaną **hierarchią wymiaru**, która umożliwia definiowanie różnych poziomów agregacji danych – co stanowi zasadniczy cel budowy systemu OLAP. Hierarchia wymiaru może być reprezentowana w magazynie danych *implicite* w pojedynczej tablicy wymiaru lub *explicite* w postaci zbioru powiązanych tablic reprezentujących jeden wymiar. Tablica faktów odzwierciedla w magazynie danych aspekt dynamiczny świata rzeczywistego, podczas gdy tablice wymiarów reprezentują aspekt statyczny. Dla ilustracji wprowadzonych pojęć rozważmy prosty przykład przedstawiony na rysunku 1. Tablica faktów nosi nazwę „*sprzedaż*”. Tablica ta zawiera informacje o dwóch miarach liczbowych: **ilość** i **podatek**, gdzie **ilość** oznacza ilość sprzedanych produktów, natomiast **podatek** oznacza podatek, który należy zapłacić od sprzedaży produktów. Na rysunku przedstawiono 3 tablice wymiarów: sklep, klient oraz produkt. Na rysunku nie przedstawiono czwartej, domyślnej, tablicy wymiaru czasu. Przedstawiony na rysunku 1 schemat pojęciowy posiada strukturę logiczną gwiazdy – schematy o

takiej strukturze logicznej nazywamy schematami gwiazdzistymi lub schematami pojęciowymi typu „gwiazda”. W schemacie typu gwiazda hierarchia wymiarów jest reprezentowana w pojedynczej tabeli wymiaru. Przykładowo, hierarchia wymiaru *produkt* (przedstawiona na rysunku 2) zawiera się całkowicie w tabeli *produkt*.

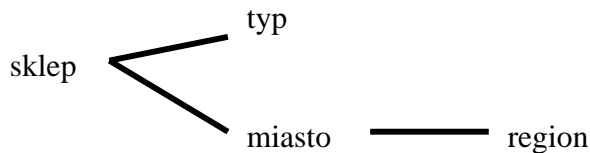


Rysunek 1. Przykładowy schemat logiczny magazynu danych



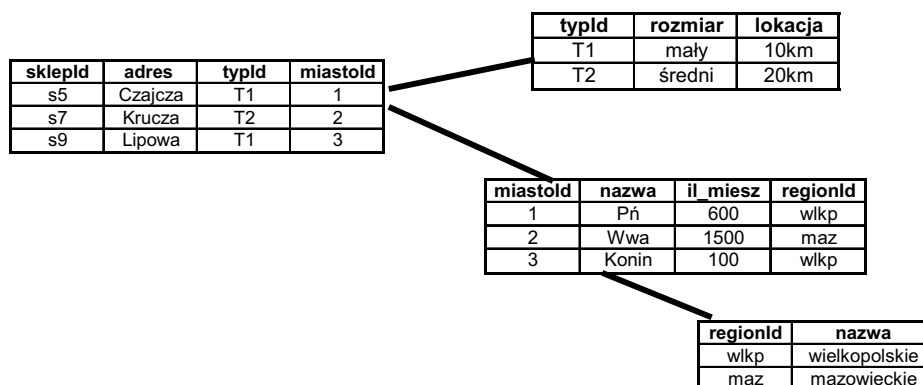
Rysunek 2. Hierarchia wymiaru *produkt*

Hierarchia wymiaru może mieć bardziej złożony charakter. Przykładowo, hierarchia wymiaru *sklep* ma następującą postać (rysunek 3).



Rysunek 3. Hierarchia wymiaru *sklep*

W przypadku, gdy hierarchia wymiaru jest reprezentowana przez zbiór tablic wówczas schemat logiczny przyjmuje strukturę „*płatka śniegu*” (patrz rysunek 4). W przypadku, gdy zbiór miar współdzieli zbiór wymiarów: (1) na różnych poziomach hierarchii wymiarów lub (2) współdzielony zbiór wymiarów nie jest identyczny, wówczas schemat logiczny magazynu danych przyjmuje strukturę „*konstelacji faktów*”. Struktura „konstelacji faktów” jest najogólniejszą strukturą logiczną magazynu danych.



Rysunek 4. Fragment schematu logicznego o strukturze „płatka śniegu”

### 3. *Ewolucja schematów logicznych magazynów danych*

Jak już wspomnieliśmy, podstawowe założenie przyjmowane w systemach OLAP zakłada, że aspekt dynamiczny świata rzeczywistego jest odzwierciedlany tylko i wyłącznie w tablicach faktów, podczas gdy tablice wymiarów pozostają niezmienione (reprezentują aspekt statyczny). To założenie jest w praktyce bardzo trudne do spełnienia. Przykładowo, biorąc pod uwagę wymiar sklep (rysunek 1 i 4), sklepy są likwidowane, niektóre zmieniają charakter (zmiana typu), zmieniają adresy i regiony. Hierarchia wymiaru może ulec zmianie, np. wprowadzenie nowego podziału organizacyjnego (wprowadzenie powiatów). Wreszcie, ulec zmianie może relacja pomiędzy miarami a wymiarami. Przykładowo, miara podatek, dotychczas związana z wymiarem czasu na poziomie kwartału, po zmianie może być zależna od miesiąca. Reasumując, w praktyce, dynamika świata rzeczywistego jest odzwierciedlana zarówno na poziomie tablic faktów jak i na poziomie tablic wymiarów czy hierarchii wymiarów. W konsekwencji szereg analiz wykonywanych w dłuższych horyzontach czasowych obejmujących zmiany wymiarów jest niepoprawnych lub wymaga zmiany samej aplikacji analitycznej. Dla ilustracji problemów związanych ze zmianami schematu logicznego rozważmy kilka przykładów.

#### Przykład 1.

Dany jest schemat logiczny przedstawiony na rys. 1. Załóżmy, że tablica faktów ma następującą postać. W tablicy pominięto, dla uproszczenia dyskusji, wartość miary podatek.

sprzedazId	data	klientId	prodId	sklepId	ilosc
sp1	d1	c1	p1	s1	100
sp2	d2	c2	p1	s2	100
sp3	d3	c3	p3	s1	100
sp4	d4	c4	p4	s2	100

Założmy również, że klasyfikacja produktów do określonych typów jest następująca:  $\{(p1, t1), (p2, t1), (p3, t2), (p4, t2)\}$ , tzn. produkt p1 jest typu t1, produkt p2 jest typu t1, itd.

Niech dane będzie zapytanie analityczne postaci „podaj łączną sprzedaż różnych typów produktów w różnych sklepach”. Zapytanie to zwróci następujący wynik.

sklepId	typ	ilość
s1	t1	100
s1	t2	100
s2	t1	100
s2	t2	100

Przypuśćmy, że w chwili d5 ( $d5 > d4$ ) produkt p1 został zaklasyfikowany do typu t2. W magazynie danych nastąpi modyfikacja krotki w relacji **produkt** i zastąpienie dotychczasowej krotki (p1, cegła, t1, 7) krotką (p1, cegła, t2, 12). Poprzednia krotka zostanie usunięta z magazynu danych a wraz z nią poprzednia klasyfikacja produktu p1. Powtarzając poprzednie zapytanie, tym razem otrzymamy następujący wynik.

sklepId	typ	ilość
s1	t2	200
s2	t2	200

Okazuje się, że żaden ze sklepów nie sprzedał produktów typu t1. Pojawia się problem z interpretacją otrzymanego wyniku i jego porównanie z poprzednim raportem. Poza problemem związanym z interpretacją otrzymanego wyniku, konsekwencje praktyczne mogą być poważniejsze. Załóżmy, że odprowadziliśmy podatek VAT związany z określonym typem produktu zgodnie z raportem 1. Pod koniec roku okazuje się, że musimy dopłacić podatek, gdyż „błędnie” naliczyliśmy VAT (typ t2 ma wyższą stawkę VAT).

## Przykład 2.

Rozważmy ponownie schemat logiczny przedstawiony na rys. 1. Załóżmy, że tablica faktów ma następującą postać. W tablicy znów pominięto, dla uproszczenia dyskusji, wartość miary podatek.

sprzedazId	data	klientId	prodId	sklepId	ilosc
sp1	d1	c1	p1	s1	100
sp2	d1	c2	p1	s2	100
sp3	d1	c3	p3	s3	100
sp4	d3	c4	p4	s4	200
sp5	d3	c1	p2	s3	100

Przypuśćmy, że w chwili d2 ( $d3 > d2 > d1$ ) nastąpiło połączenie sklepów s1 i s2, a na ich miejsce pojawił się nowy supermarket s4. Usunięto zatem z magazynu danych krotki opisujące sklepy s1 i s2, i wprowadzono krotkę opisującą sklep s4. Nasuwa się kilka pytań. Co w takim przypadku z krotkami opisującymi sprzedaż w sklepach s1 i s2 pamiętanymi dotychczas w tablicy faktów? Jeżeli wraz z krotkami opisującymi sklepy s1 i s2 usuniemy odpowiednie krotki z tablicy faktów (sp1 i sp2) pojawi się „manko” w ilości sprzedanych produktów (wynik raportu dotyczący sumarycznej sprzedaży produktów nie będzie odpowiadał rzeczywistej ilości sprzedanych produktów). Pozostawienie tych krotek w tablicy faktów spowoduje inne problemy. Każdy raport zawierający warunek agregacji zdefiniowany na atrybutach wymiaru sklep będzie błędny (np. zapytanie o sprzedaż produktów w „małych” sklepach). Wynika to z faktu, że w tablicy wymiarów brakuje krotek umożliwiających połączenie krotek sp1 i sp2 z odpowiednimi krotkami tablicy **sklep**. Błędne mogą się również okazać raporty nie wymagające operacji połączenia tablicy faktów z tablicą wymiaru sklep, np. zapytanie o średnią sprzedaż dzienną produktów w poszczególnych sklepach. Wynikiem zapytania (przy założeniu pozostawienia krotek sp1 i sp2) będzie następująca tablica.

sklepId	śr_ilość
s1	50
s2	50
s3	100
s4	100

Łatwo zauważyć, że za wyjątkiem sklepu s3, dla pozostałych sklepów średnia ilość sprzedanych produktów jest niepoprawna.

### Przykład 3.

Dany jest schemat logiczny przedstawiony na rys. 1. W przyjętym schemacie logicznym podatek jest płacony od sprzedaży pojedynczego produktu (stanowi procent ceny produktu). Załóżmy, że nastąpiła zmiana regulacji prawnych i od dnia d5 podatek jest płacony nie od sprzedaży pojedynczego produktu, lecz jest płacony raz w miesiącu od wartości zysku ze sprzedaży produktów określonego typu (materiały budowlane 12%, książki 7%, itd.). Należy pamiętać, że dla różnych produktów, nawet tego samego typu, wartość zysku może być różna. Nie ma zatem prostej funkcji pozwalającej na obliczenie podatku wg. nowych regulacji prawnych dla produktów sprzedanych przed datą d5. Zmiana zasad naliczania podatku pociąga za sobą konieczność zmiany schematu logicznego magazynu danych. Zmiana ta polega na: (1) wprowadzenie do tablicy faktów nowej miary **zysk** (związany ze sprzedażą pojedynczego produktu), (2) usunięciu miary podatek, (3) utworzeniu nowej tablicy określającej wartość podatku będącej wynikiem połączenia tablicy faktów, tablicy wymiaru produkt oraz tablicy wymiaru czas, a następnie wykonania operacji **group by** (typ produktu) i agregacji wartości zysku dla poszczególnych typów produktów w poszczególnych miesiącach. Zasadniczym problemem nie jest zmiana schematu logicznego wynikająca ze zmian w otaczającym świecie, lecz konieczność dokonania zmian w aplikacjach analitycznych np. w raporcie rocznym o zapłaconych podatkach. Pozostaje jeszcze kwestia w jaki sposób należy interpretować wyniki tego raportu w przypadku, gdy raport obejmuje okres funkcjonowania obu regulacji prawnych.

Łatwo zauważyć, że gdyby tablica faktów od początku zawierała informacje o zysku ze sprzedaży poszczególnych produktów, wówczas można by łatwo dokonywać obliczenia wartości podatku zarówno na starych jak i nowych zasadach. W przypadku braku tej informacji, wartość „nowego” podatku można wyliczyć tylko dla okresu, w którym dysponujemy informacją o zysku ze sprzedaży produktów. Wartość „starego” podatku możemy obliczać tak jak poprzednio.

### Przykład 4.

Założmy, że dany jest schemat logiczny z rys. 1, w którym pominięto miarę podatek. Założmy, że dotychczas podatek był płacony nie od sprzedaży, lecz od wielkości sklepu niezależnie od wartości sprzedanych produktów czy też zysku. Podatek był płacony rocznie ryczałtem. Można założyć, że wartość podatku była określona wartością atrybutu podatek w tablicy wymiaru sklep. Założmy następnie, że w późniejszym czasie zmieniono zasady obliczania podatku i przyjęto zasady przedstawione w przykładzie 3. Pytanie: w jaki sposób obliczyć średnią wartość podatku płaconą rocznie od sprzedaży produktów danego typu? Warto zwrócić uwagę, że dotychczasowe aplikacje analityczne (do momentu zmiany zasad naliczania podatku) dotyczyły wyłącznie sklepu. Aplikacje te są bezużyteczne w późniejszym czasie. Z kolei nowe aplikacje mają określony horyzont czasowy.

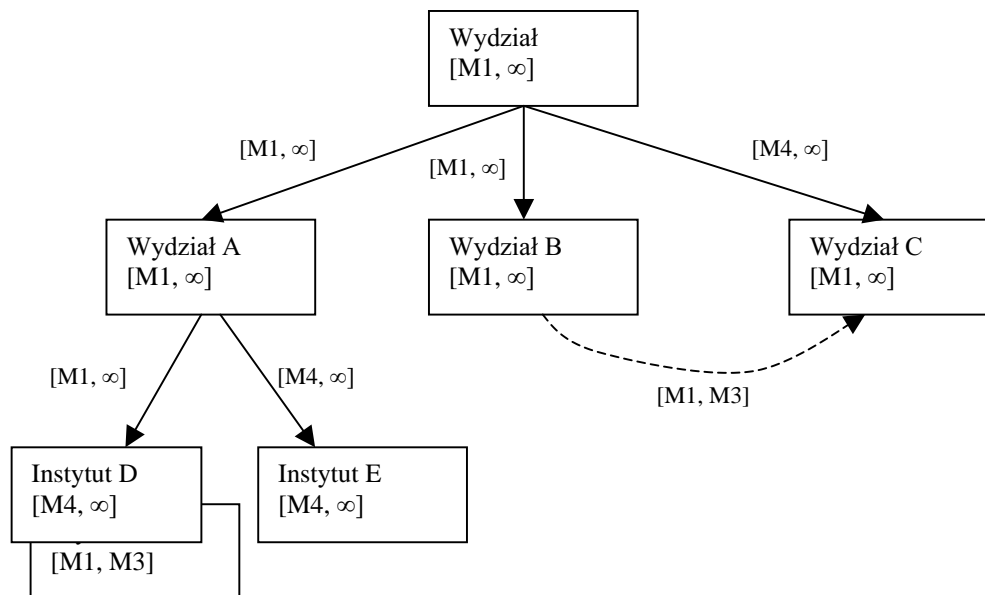
Wszystkie przedstawione powyżej przykłady ilustrują problemy jakie wiążą się z ewolucją schematów logicznych magazynów danych i z interpretacją wyników generowanych przez raporty analityczne w przypadku takiej ewolucji. Jak wynikało z przedstawionych przykładów, część raportów analitycznych generuje poprawne wyniki mimo modyfikacji schematu logicznego magazynu danych. Część aplikacji wymaga aktualizacji po to, aby poprawnie generować wyniki. W przypadku części raportów mamy do czynienia z problemem poprawnej interpretacji wyników. Wreszcie, część raportów jest niepoprawnych i nie może być uruchamiana ze względu na brak danych.

Jest oczywistym, że modyfikacje schematu logicznego magazynu danych są nieuniknione w praktyce. W związku z tym, aby zagwarantować poprawność zapytań OLAP-owych i zapewnić tym samym szerszą stosowalność magazynów danych i systemów OLAP, konieczne i niezbędne jest rozwiązanie problemów związanych z ewolucją schematów magazynów danych, w szczególności, problemów związanych z modyfikacjami tablic wymiarów i hierarchii wymiarów. W przeciwnym razie, błędne

raporty analityczne będą prowadziły do błędnych decyzji. Przykładów na to mamy pod dostatkiem w naszym życiu publicznym.

#### 4. Temporalne systemy OLAP

Problem poprawności ewolucji tablic i hierarchii wymiarów w magazynach danych jest znany od wielu lat. Niestety, nie doczekał się on jeszcze rozwiązania praktycznego. Zdecydowana większość proponowanych rozwiązań opiera się na propozycji wprowadzeniu do magazynów danych elementów systemów temporalnych lub wielwersyjnych. W obu przypadkach, podstawowym mechanizmem, którego zadaniem jest umożliwienie śledzenia zmian zachodzących w schemacie magazynu danych, jest mechanizm **etykiet czasowych** (ang. timestamp). Zgodnie z proponowanymi rozwiązaniami, każdy obiekt w magazynie danych otrzymuje etykietę czasową. Etykieta czasowa obiektu określa przedział czasu, w którym obiekt jest ważny (aktualna). Przez obiekt rozumiemy rekordy tablic wymiarów, rekordy tablic faktów, przypisanie elementów tablic wymiarów do hierarchii wymiarów, zależności pomiędzy miarami a wymiarami. Przydzielenie etykiet czasowych do obiektów w magazynie danych pozwala na wyspecyfikowanie różnych wersji schematu logicznego magazynu danych. Wersje te reprezentują spójny stan magazynu danych w określonym przedziale czasowym. Dla ilustracji proponowanych rozwiązań ograniczmy się wyłącznie do problemu aktualizacji wymiarów analizy. Każdy wymiar składa się z dwóch elementów: (1) składników wymiaru (rekordy tablic wymiarów) oraz (2) hierarchicznej relacji opisującej zależność pomiędzy składnikami danego wymiaru (graf hierarchii wymiaru). Etykietowaniu muszą zatem podlegać oba elementy każdego wymiaru. Wymiar można interpretować jako graf skierowany, którego wierzchołkami są składniki wymiaru, natomiast łuki reprezentują relację zależności pomiędzy składnikami wymiaru. Rysunek 5 przedstawia etykietowany graf wymiaru **Wydział** ilustrujący wprowadzone wyżej pojęcia.



Każdy wierzchołek i każdy łuk w grafie posiada przedział czasowy  $[T_s, T_e]$ , określający ważność danego elementu. W przedstawionym grafie, Instytut D został zmodyfikowany w chwili  $M_4$ . W chwili  $M_4$  został utworzony nowy Instytut E (stąd konieczność modyfikacji Instytutu D). Oba instytuty należą do Wydziału A. W chwili  $M_4$  z Wydziału B został wydzielony Instytut C i na jego bazie został utworzony nowy wydział – Wydział C ( w czasie  $[M_1, M_3]$  Wydział C stanowił część Wydziału B). Analizując powyższy graf można zidentyfikować dwie wersje schematu logicznego magazynu danych, które obowiązywały w dwóch różnych okresach czasu:

- Wersja SV1: ważna w okresie od M1 do M3 -  $\langle \text{SV1}, [M1, M3], \{\{\text{Wydział}, \text{Wydział A}, \text{Wydział B}, \text{Wydział C}, \text{Instytut D}\}, \{\text{miara}\}, \{\text{Wydział A} \rightarrow \text{Wydział}, \text{Wydział B} \rightarrow \text{Wydział}, \text{Instytut D} \rightarrow \text{Wydział A}, \dots\}\rangle$
- Wersja SV2: ważna w okresie od M4 do  $\infty$  -  $\langle \text{SV2}, [M1, \infty], \{\{\text{Wydział}, \text{Wydział A}, \text{Wydział B}, \text{Wydział C}, \text{Instytut D}, \text{Instytut E}\}, \{\text{miara}\}, \{\text{Wydział A} \rightarrow \text{Wydział}, \text{Wydział B} \rightarrow \text{Wydział}, \text{Wydział C} \rightarrow \text{Wydział}, \text{Instytut D} \rightarrow \text{Wydział A}, \text{Instytut E} \rightarrow \text{Wydział A}, \dots\}\rangle$

Koncepcja temporalnego wielowersyjnego magazynu danych odnosi się zarówno do struktury logicznej jak i zbioru operatorów służących do pielęgnacji takiego temporalnego wielowersyjnego magazynu danych. Wprowadza się trzy podstawowe operatory zmian struktury logicznej magazynu danych (insert, delete, update), a następnie, korzystając z tych operatorów, można zdefiniować złożone operacje na schematach logicznych: *split* (podziel pojedynczy składnik wymiaru na n składników), *merge* (połącz n składników wymiaru w jeden składnik), *change* (zmień wartość atrybutu składnika wymiaru), *move* (zmień pozycje składnika w hierarchii wymiaru), *new-member* (wprowadź nowy składnik), *delete-member* (usuń składnik wymiaru). W odniesieniu do klasycznej koncepcji magazynów danych, przedstawiony zbiór operacji na wymiarach stanowi istotne rozszerzenie funkcjonalności magazynów danych. Należy podkreślić, że przedstawiony zbiór operatorów odnosi się tylko i wyłącznie do modyfikacji wymiarów analizy. Uwzględniając inne możliwe modyfikacje schematu logicznego magazynu danych (np. zmiana schematu płatka śniegu na schemat konstelacji faktów), zbiór operatorów musi być, dodatkowo, rozszerzony o operacje modyfikacji tablic faktów. Przykładowo, konieczne jest wprowadzenie nowych operatorów umożliwiających połączenie dwóch tablic faktów czy też ich podział.

Pojedynczą wersję schematu logicznego magazynu danych można interpretować jako perspektywę (lub migawkę) zdefiniowaną na schemacie logicznym magazynu danych i ważną w określonym przedziale czasowym  $[T_s, T_e]$ . Z każdą wersją schematu logicznego magazynu danych jest związany zbiór ograniczeń integralnościowych określających spójny (poprawny) stan magazynu danych. Jednakże, podobnie jak w przypadku wielowersyjnych czy też rozproszonych replikowanych baz danych, poza ograniczeniami integralnościowymi związanymi z pojedynczą wersją schematu logicznego magazynu danych mogą występować ograniczenia integralnościowe pomiędzy wersjami schematu logicznego. Oznacza to również konieczność rozszerzenia języka definiowania danych o możliwość definiowania ograniczeń integralnościowych „intra i inter wersyjnych”. Takie prace są sygnalizowane w literaturze, ale jak dotąd nie zaproponowano żadnych rozwiązań w tym zakresie.

Podstawowym problemem wymagającym rozwiązania w przypadku temporalnych wielowersyjnych magazynów danych jest problem języka zapytań. Rozważmy temporalny wielowersyjny magazyn danych zawierający cztery wersje ważne w określonych przedziałach czasowych.

Wersja	Ts	Te
SV1	styczeń 1970	marzec 1989
SV2	kwiecień 1989	styczeń 1995
SV3	luty 1995	grudzień 1999
SV4	styczeń 2000	

Użytkownik definiując zapytanie do temporalnego wielowersyjnego magazynu danych musi określić wersję schematu, do której realizuje swoje zapytanie. Wersja ta określa semantykę zapytania. Przykładowo, jeżeli miarą jest produkt krajowy brutto i zapytanie jest adresowane do wersji magazynu z roku 1979 (istniały wówczas RFN, NRD, Czechosłowacja), to wówczas zapytanie postaci „podaj produkt krajowy brutto w okresie od 1970 – 1994 odnosi się do składników wymiarów istniejących w schemacie magazynu danych w roku 1979. Podobnie, zapytanie o zysk ze sprzedaży sformułowane w odniesieniu do schematu z roku np. 1995 odnosi się do interpretacji zysku z tego roku (definicja zysku może zmieniać się w czasie i może lub nie uwzględniać, przykładowo, amortyzację czy podatek VAT). Dane zwracane przez zapytanie mogą pochodzić z kilku wersji temporalnego magazynu danych. W pierwszym z podanych wyżej przykładów zapytanie zostało sformułowane w odniesieniu

do wersji SV1 schematu magazynu danych, jednakże dane są pobierane z wersji SV1 i SV2. W takim przypadku potrzebna jest funkcja transformująca dane wersji SV2 do postaci wersji schematu SV1. Oznacza to, przykładowo, że użytkownik w raporcie odnoszącym się do np. produktu krajowego brutto zobaczy „hipotetyczny” produkt krajowy brutto nieistniejącego państwa NRD w roku 1991, 1992, 1993, 1994. Pojawia się pytanie o sens takiej analizy. Z drugiej strony, można interpretować taki raport jako symulację sytuacji i analizować „hipotetyczny” wpływ NRD na produkt krajowy Niemiec.

Generalnie, można wyróżnić następujące przypadki formułowania zapytań do wielowersyjnego temporalnego magazynu danych:

- 1) zapytanie jest formułowane w odniesieniu do danych należących do pojedynczej wersji schematu. Ten przypadek odpowiada zapytaniu do klasycznego magazynu danych.
- 2) zapytanie jest formułowane w odniesieniu do danych należących do dwóch kolejnych wersji schematu magazynu danych. Jeżeli istnieje funkcja transformująca dane pomiędzy tymi wersjami danych, to po transformacji otrzymujemy przypadek (1). Jeżeli taka funkcja jest niezdefiniowana, to wynik zapytania jest „niezdefiniowany”.
- 3) zapytanie jest formułowane w odniesieniu do danych należących do dwóch, ale nie kolejnych wersji schematu magazynu danych. Jeżeli istnieje funkcja transformująca dane pomiędzy tymi wersjami danych, to po transformacji otrzymujemy przypadek (1). Jeżeli taka funkcja jest niezdefiniowana, to wynik zapytania jest „niezdefiniowany”.
- 4) zapytanie jest formułowane w odniesieniu do danych należących do trzech i więcej wersji schematu magazynu danych. Jeżeli istnieją funkcje transformujące dane pomiędzy tymi wersjami danych do jednej wybranej wersji schematu, to po transformacji otrzymujemy przypadek (1). Jeżeli takie funkcje są niezdefiniowane, to wynik zapytania jest również „niezdefiniowany”.

## **5. Podsumowanie**

W przeciwieństwie do systemów temporalnych i wielowersyjnych baz danych intensywnie rozwijanych i analizowanych w literaturze, problematyka temporalnych i wielowersyjnych magazynów danych nie doczekała się jak dotąd poważniejszej analizy. Podobnie rzecz się ma z zagadnieniem ewolucji schematów baz danych i magazynów danych. Problem ewolucji i wersjonowania schematów magazynów danych był rozważany w pracach Blaschki. Zagadnienie nowych operatorów dla zmiennych w czasie wymiarów magazynów danych rozważał A. Mendelzon. Spośród producentów oprogramowania komercyjnego problem ten podjęły, jak wynika z dostępnych autorowi materiałów opublikowanych w tzw. „white papers” SAP Amerika i Essbase. Proponują rozszerzenie schematów logicznych o etykiety czasowe w celu umożliwienia użytkownikom analizy różnych scenariuszy rozwoju magazynu danych. Zaproponowane podejście jest ograniczone do kilku podstawowych operatorów (wstaw/usuń składnik wymiaru, zmień strukturę hierarchii wymiaru). Jak wynika z krótkiego przedstawienia problemu ewolucji schematów logicznych magazynów danych badania nad rozwiązaniem tych problemów znajdują się na etapie wstępnym. Rozwiązania wymagają podstawowe zagadnienia związane z modelem wielowersyjnego magazynu danych, z określeniem zbioru operatorów umożliwiających modyfikację schematu logicznego magazynu danych, z możliwością definiowania ograniczeń integralnościowych, czy wreszcie opracowaniem języka dostępu do temporalnych wielowersyjnych magazynów danych.