

IBM DB2 Cube Views



# Podręcznik i skorowidz

*Wersja 8.2*



IBM DB2 Cube Views



# Podręcznik i skorowidz

*Wersja 8.2*

**Uwaga**

Uwaga: Przed użyciem podanych tu informacji i produktu, którego one dotyczą, należy zapoznać się z treścią sekcji “Uwagi” na stronie 291.

**Wydanie drugie (wrzesień 2004)**

Dokument ten zawiera informacje dotyczące praw własności IBM. Informacje te są dostarczane na warunkach umowy licencyjnej i są chronione prawem autorskim. Informacje zawarte w niniejszej publikacji nie dotyczą jakichkolwiek gwarancji udzielanych na produkty, a jakiegokolwiek oświadczenia znajdujące się w tym podręczniku nie mogą być interpretowane jako gwarancje.

Publikacje firmy IBM można zamawiać przez Internet lub za pośrednictwem lokalnego przedstawiciela IBM.

- Aby zamówić publikacje przez Internet, należy przejść do strony IBM Publications Center pod adresem [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)
- Aby odnaleźć lokalnego przedstawiciela firmy IBM, należy przejść do strony IBM Directory of Worldwide Contacts pod adresem [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

Aby zamówić publikacje DB2 w dziale sprzedaży i marketingu DB2 w Stanach Zjednoczonych lub Kanadzie, należy zadzwonić pod numer 1800-IBM-4YOU (425-4968).

Przesyłając informację do IBM, Użytkownik udziela IBM niewyłącznego prawa do używania i rozpowszechniania tych informacji w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

© Copyright International Business Machines Corporation 2004. Wszelkie prawa zastrzeżone.

# Spis treści

## Informacje o podręczniku . . . . . v

Kto powinien przeczytać ten podręcznik . . . . . v

Konwencje składniowe stosowane w tej książce . . . . . v

Informacje w formie elektronicznej . . . . . vi

## Rozdział 1. Instalowanie, migrowanie i konfigurowanie programu DB2 Cube Views . . . . . 1

Rozszerzenia programu DB2 Cube Views . . . . . 1

Wprowadzenie do programu DB2 Cube Views. . . . . 4

Wymagania instalacyjne . . . . . 5

Instalowanie programu DB2 Cube Views . . . . . 6

Przeprowadzanie migracji plików XML programu DB2 Cube Views z wersji 8.1 do 8.2 . . . . . 7

Konfigurowanie bazy danych dla programu DB2 Cube Views . . . . . 8

Konfigurowanie przykładowej bazy danych CVSAMPLE . . . . . 8

Rozwiązywanie problemów z połączeniem z bazą danych . . . . . 9

## Rozdział 2. Obiekty metadanych w programie DB2 Cube Views . . . . . 11

Obiekty metadanych mające odwzorowanie w tabelach relacyjnych . . . . . 11

Przykład modelu kostki mającego odwzorowanie w tabelach relacyjnych . . . . . 14

Wspólne właściwości obiektów metadanych . . . . . 19

Modele kostek . . . . . 20

Obiekty faktów. . . . . 21

Wymiary . . . . . 22

Hierarchie . . . . . 22

Poziomy. . . . . 27

Miary . . . . . 30

Atrybuty . . . . . 32

Związki atrybutów. . . . . 33

Łączenia . . . . . 34

Kostki . . . . . 35

Obiekt faktów kostki . . . . . 36

Wymiary kostki . . . . . 36

Hierarchie kostki . . . . . 37

Poziomy kostki. . . . . 37

Reguły obiektów metadanych . . . . . 38

## Rozdział 3. Projektowanie modeli obiektów metadanych programu DB2 Cube Views. . . . . 43

Uruchamianie i odświeżanie Centrum OLAP. . . . . 43

Kompatybilność Centrum OLAP i wersji interfejsu API . . . . . 44

Uprawnienia do korzystania z programu DB2 Cube Views . . . . . 44

Tworzenie obiektów metadanych programu DB2 Cube Views . . . . . 47

Wymiana metadanych między programem DB2 Cube Views i narzędziami OLAP . . . . . 47

Tworzenie modelu kostki przy użyciu Kreatora szybkiego startu . . . . . 48

Tworzenie kompletnego modelu kostki . . . . . 49

Dodawanie istniejącego wymiaru do modelu kostki . . . . . 53

Tworzenie łączenia . . . . . 53

Tworzenie kostki . . . . . 54

Usuwanie wymiaru z modelu kostki . . . . . 56

Usuwanie obiektu metadanych z bazy danych . . . . . 57

## Rozdział 4. Scenariusze modelowania biznesowego w programie DB2 Cube Views . . . . . 59

Obliczanie ruchu i wartości zapasów w magazynie w czasie . . . . . 59

Korelacja kosztów reklamy ze sprzedażą . . . . . 64

Obliczanie zysku i marży sklepu . . . . . 69

Zliczanie zamówień internetowych. . . . . 73

Ranking wysokości obrotów. . . . . 75

Użycie danych czasowych w tabeli faktów do utworzenia wymiaru czasu . . . . . 78

## Rozdział 5. Optymalizacja modelu kostki programu DB2 Cube Views . . . . . 81

Tabele podsumowań . . . . . 82

Tabele podsumowań z zależnościami funkcjonalnymi i ograniczeniami. . . . . 87

Przegląd procesu optymalizacji . . . . . 90

Projekt metadanych a optymalizacja . . . . . 93

Wycinki optymalizacji dla kostek . . . . . 95

Analizowanie zapytań pod kątem proponowanych wycinków optymalizacji . . . . . 104

Definicje ograniczeń optymalizacyjnych . . . . . 107

Parametry Doradcy optymalizacyjnego . . . . . 111

Optymalizowanie modelu kostki . . . . . 113

Przykład skryptu SQL do tworzenia tabel podsumowań . . . . . 114

Testowanie wyników zapytania . . . . . 117

Rozwiązywanie problemów z tabelami podsumowań . . . . . 118

Aktualizacja tabel podsumowań . . . . . 119

Usuwanie tabeli podsumowań . . . . . 120

## Rozdział 6. Program DB2 Cube Views a stowarzyszone źródła danych. . . . . 123

Przegląd systemów stowarzyszonych . . . . . 123

Optymalizowanie zdalnych źródeł danych za pomocą programu DB2 Cube Views - przegląd . . . . . 124

Umożliwianie obsługi systemu stowarzyszonego przez program DB2 Cube Views . . . . . 126

Definiowanie zdalnych źródeł danych . . . . . 126

Definiowanie pseudonimów tabel zdalnych dla programu DB2 Cube Views . . . . . 127

Definiowanie ograniczeń informacyjnych dla pseudonimów dla programu DB2 Cube Views . . . . . 128

Rozwiązywanie problemów z wydajnością zapytań w przypadku zdalnych źródeł danych . . . . . 128

## **Rozdział 7. Interfejs API programu DB2 Cube Views . . . . . 131**

Przegląd interfejsu API programu DB2 Cube Views . . . . .	131
Interfejs API programu DB2 Cube Views: procedura zapisana w bazie DB2 i analiza składniowa XML . . . . .	132
Procedura zapisana w bazie programu DB2 Cube Views . . . . .	133
Parametry funkcji API programu DB2 Cube Views . . . . .	135
Parametry operacji na metadanych interfejsu API programu DB2 Cube Views . . . . .	135
Parametry wejściowe i wyjściowe . . . . .	135
Operacje na metadanych programu DB2 Cube Views . . . . .	136
Operacja pobierania: Describe . . . . .	136
Operacja doradzania: Advise . . . . .	137
Operacje modyfikujące: Alter, Create, Drop, Import i Rename . . . . .	143
Operacje administracyjne: Validate i Translate . . . . .	147
Przykładowe parametry wejściowe i wyjściowe w operacjach na metadanych . . . . .	150
Parametry operacji . . . . .	152
Operandów operacji . . . . .	155
Struktura komunikatów . . . . .	155
Kolejność kroków operacji . . . . .	156
Protokołowanie i śledzenie . . . . .	157
Śledzenie w czasie wykonywania dla interfejsu API programu DB2 Cube Views . . . . .	157
Pliki protokołów interfejsu API programu DB2 Cube Views . . . . .	157
Protokołowanie błędów . . . . .	158
Scenariusze protokołowania i śledzenia . . . . .	159
Obsługa stron kodowych . . . . .	159
Tabele metadanych i pliki schematu XML programu DB2 Cube Views . . . . .	161

Plik konfiguracyjny programu DB2 Cube Views . . . . .	162
Format obiektów metadanych . . . . .	163

## **Rozdział 8. Pliki przykładowe. . . . . 165**

Przegląd programu narzędziowego db2mdapiclient. . . . .	165
Komenda db2mdapiclient: manipulowanie obiektami metadanych . . . . .	166
Przykładowe pliki bazy danych . . . . .	167
Przykładowe pliki API . . . . .	168

## **Dodatek. Komunikaty . . . . . 173**

SQLSTATE, API i inne komunikaty serwera . . . . .	173
Stany SQL interfejsu API . . . . .	173
Wspólne . . . . .	175
Błędy zewnętrznych funkcji API . . . . .	197
Optymalizacja . . . . .	243
Komunikaty Centrum OLAP . . . . .	253
10000-10600 . . . . .	253
Komunikaty stanu programów DB2 i DB2 Cube Views . . . . .	290

## **Uwagi. . . . . 291**

Znaki towarowe . . . . .	293
--------------------------	-----

## **Glosariusz. . . . . 295**

## **Indeks . . . . . 299**

## **Kontakt z firmą IBM. . . . . 303**

Informacje o produktach . . . . .	303
Komentarz na temat dokumentacji . . . . .	303

---

## Informacje o podręczniku

W tym podręczniku omówiono następujące tematy dotyczące programu DB2 Cube Views:

- Pierwsze kroki w programie DB2 Cube Views.
- Centrum OLAP (graficzny interfejs użytkownika), służące m.in. do importowania i eksportowania metadanych, tworzenia modeli kostki i kostek.
- Optymalizacja, pozwalająca zwiększyć wydajność zapytań OLAP.
- Obiekty metadanych, które można przechowywać w katalogach bazy danych DB2 Universal Database (DB2).
- Aplikacyjny interfejs programowy (API) służący do tworzenia aplikacji umożliwiających dostęp do danych przy użyciu kodu w języku SQL.
- Przykłady budowania wymiarów i złożonych miar na podstawie obiektów metadanych; te wymiary i miary mogą posłużyć do modelowania typowych rozwiązań biznesowych.

---

## Kto powinien przeczytać ten podręcznik

Przy użyciu programu DB2 Cube Views można pobierać wielowymiarowe metadane z narzędzi OLAP i narzędzi baz danych oraz zapisywać te metadane w katalogach bazy danych DB2. Następnie można tych metadanych użyć do utworzenia modeli kostki OLAP (analityczne przetwarzanie na bieżąco) i kostek. (Kostki są podzbiorami modeli kostek).

Program DB2 Cube Views udostępnia również Doradcę optymalizacyjnego dostarczającego skrypty SQL do budowania tabel podsumowań, pomagających zwiększyć wydajność zapytań kierowanych do modeli kostki.

Podręcznik ten powinni przeczytać administratorzy baz danych posługujący się metadanymi OLAP i programem DB2 Universal Database (DB2 UDB). Konieczna jest znajomość:

- katalogów DB2 i tabel podsumowań,
- pojęć z dziedziny przetwarzania OLAP, takich jak kostki, wymiary, hierarchie i miary,
- pojęć z dziedziny interfejsów API i CLI, ODBC, JDBC, XML oraz zapisanych w bazie procedur programu DB2.

---

## Konwencje składniowe stosowane w tej książce

Składnia komend jest przedstawiana w formacie tekstu bieżącego z zastosowaniem następujących konwencji:

- Komendy są przedstawiane czeionką pogrubioną, małymi literami.
- Zmienne są przedstawiane kursywą i są objaśniane na liście tuż po komendzie.
- Jeśli można wprowadzić jeden z co najmniej dwóch alternatywnych parametrów, dostępne parametry są oddzielane pionowymi kreskami, przy czym parametr domyślny jest podkreślany.
- Zmienne i parametry opcjonalne są ujmowane w nawiasy kwadratowe.

Na przykład składnię komendy **copy** można przedstawić w następującej postaci:

**copy** *nazwa\_pliku* [*typ\_pliku*]

*nazwa\_pliku* jest wymaganą nazwą pliku.

[*typ\_pliku*] jest opcjonalnym typem pliku.

---

## Informacje w formie elektronicznej

W tej sekcji podano adresy WWW serwisów związanych z tym produktem.

### **www.ibm.com/redbooks**

Serwis WWW IBM Redbooks

W serwisie WWW Redbooks można wyszukiwać, przeglądać, pobierać lub zamawiać następujące pozycje dokumentacji technicznej (Redbooks) w postaci drukowanej lub na dysku CD:

- *DB2 UDB's High Function Business Intelligence in e-business*, SG24-6546-00
- *Up and Running with DB2 UDB ESE Partitioning for Performance in an e-Business Intelligence World*, SG24-6917-00
- *Database Performance Tuning on AIX*, SG24-5511-01
- *DB2 UDB V7.1 Performance Tuning Guide*, SG24-6012-00

### **www.ibm.com/software/data/db2/db2md**

Serwis WWW programu IBM DB2 Cube Views

### **www.ibm.com/software/data/**

Serwis WWW firmy IBM poświęcony zarządzaniu danymi

### **www.ibm.com/software/data/db2/udb/winos2unix/support/**

Serwis WWW internetowego działu wsparcia dotyczący programów DB2 Universal Database i DB2 Connect

### **www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report**

Konserwacja programu DB2 - pakiety poprawek serwisu WWW programu DB2 UDB

### **www.ibm.com/software/data/developer**

Serwis WWW dla programistów - DB2 Developer Domain

### **www.ibm.com/software/data/db2/library**

Serwis WWW biblioteki technicznej produktów i usług DB2

### **www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8pubs.d2w/en\_main**

Serwis WWW publikacji na temat programu DB2



---

# Rozdział 1. Instalowanie, migrowanie i konfigurowanie programu DB2 Cube Views

W tej sekcji omówiono następujące tematy:

## **Rozszerzenia programu DB2 Cube Views**

W tej wersji programu DB2 Cube Views wprowadzono kilka zmian w obsłudze metadanych w stosunku do wersji 8.1 programu DB2 Cube Views.

## **Wprowadzenie do programu DB2 Cube Views**

DB2 Cube Views jest składnikiem dodatkowym programu DB2 Universal Database, który usprawnia przetwarzanie OLAP programu DB2 UDB. Za pomocą programu DB2 Cube Views można usprawniać wdrażanie rozwiązań OLAP i zarządzać nimi oraz zwiększać wydajność narzędzi i aplikacji OLAP.

## **Wymagania instalacyjne**

Przed zainstalowaniem programu DB2 Cube Views należy się upewnić, że spełnione są wszystkie wymagania.

## **Instalowanie programu DB2 Cube Views**

Program DB2 Cube Views można zainstalować na platformach Windows, AIX, Linux oraz Środowisko Operacyjne Solaris przy użyciu Wyrzutni instalacji.

## **Migrowanie programu DB2 Cube Views**

Migrację bazy danych dostosowanej do programu DB2 Cube Views z wersji 8.1 do wersji 8.2 można przeprowadzić ręcznie uruchamiając skrypt lub posługując się Centrum OLAP.

## **Konfigurowanie bazy danych dla programu DB2 Cube Views**

Nową bazę danych można skonfigurować pod kątem użycia programu DB2 Cube Views.

## **Konfigurowanie przykładowej bazy danych CVSAMPLE**

Z programem DB2 Cube Views dostarczane są przykładowe dane, których można użyć do utworzenia przykładowej bazy danych o nazwie CVSAMPLE.

## **Rozwiązywanie problemów z połączeniem z bazą danych**

Jeśli nie można nawiązać połączenia z bazą danych przy użyciu Centrum OLAP, należy sprawdzić, czy wersja programu DB2 Cube Views jest zgodna z wersją tabel metadanych w katalogu DB2.

---

## Rozszerzenia programu DB2 Cube Views

W tej wersji programu DB2 Cube Views wprowadzono kilka zmian w obsłudze metadanych w stosunku do wersji 8.1 programu DB2 Cube Views.

### **Informacje dotyczące ograniczeń i zależności funkcjonalnych usprawniające tabele podsumowań**

Program DB2 Cube Views, korzystając z informacji o ograniczeniach i zależnościach funkcjonalnych, generuje tabele podsumowań, które są mniejsze i zapewniają lepszą obsługę zapytań.

### **Informacje dotyczące ograniczeń**

W programie DB2 Cube Views ograniczenia modelu kostki definiuje się między tabelami faktów i tabelami wymiarów (czyli między tabelami wymiarów ze

schematu typu płatek śniegu). Jeśli jakaś tabela podsumowań ogranicza klucz podstawowy tabeli wymiarów, program DB2 UDB może przekierować zapytanie do tabeli podsumowań, nawet jeśli zapytanie odwołuje się do innych kolumn tabeli wymiarów.

### **Informacje dotyczące zależności funkcjonalnych**

Program DB2 UDB dysponuje funkcją inteligentnego przekierowywania podczas rozstrzygnięcia zapytań SQL na podstawie informacji o zależnościach funkcjonalnych.

Zależności funkcjonalne umożliwiają określenie, że jedna lub kilka kolumn zależy funkcjonalnie od jednej lub kilku innych kolumn, pod warunkiem że wszystkie te kolumny znajdują się w tej samej tabeli.

Jeśli zapytanie SQL zawiera kolumnę z tabeli podsumowań i kolumnę z tabeli, na podstawie której tabela podsumowań została skonstruowana, *oraz* między tymi kolumnami istnieje zależność funkcjonalna, optymalizator DB2 może rozstrzygnąć zapytanie, łącząc te dwie tabele w celu wygenerowania ostatecznej tabeli wynikowej.

Program DB2 Cube Views może zalecić tabele podsumowań, które będą zawierały tylko kolumny kluczy dla wybranych wymiarów i poziomów, aby powstała tabela podsumowań była jeszcze mniejsza.

### **Wycinki optymalizacji określane przez użytkownika w celu poprawy tabel podsumowań**

W obecnej wersji programu DB2 Cube Views funkcjonuje mechanizm umożliwiający użytkownikowi określanie typu najczęściej używanych zapytań oraz regionu kostki, do którego będą się one odwoływać. Doradca optymalizacyjny wykorzystując te dodatkowe metadane, może skoncentrować optymalizację na odpowiednich regionach kostki. Może to być korzystne dla każdego typu zapytań, jednak największą poprawę obserwuje się w przypadku zapytań raportowych.

### **Wiele wersji schematów XML obsługiwanych przez interfejs API metadanych**

Interfejs API metadanych programu DB2 Cube Views obsługuje w pełni tylko schemat XML w wersji 8.2.0.1.0, w tym również nowe operacje DESCRIBE i TRANSLATE.

Wszystkie dokumenty XML przekazywane do i z interfejsu API metadanych muszą mieć ten numer wersji. Po tym numerze wersji procedura zapisana w bazie identyfikuje schemat XML używany przez klienta.

### **Dopuszczalność wartości pustych w atrybutach i miarach**

Miary i atrybuty interfejsu API metadanych zostały rozszerzone o właściwość tylko do odczytu o nazwie nullable (dopuszczalna wartość pusta), która jest zestawem wartości *yes* (tak), *no* (nie) i *unknown* (nieznana).

#### **Dopuszczalność wartości pustych ustawiona na *yes***

Atrybut lub miara może zawierać wartości puste.

#### **Dopuszczalność wartości pustych ustawiona na *no***

Atrybut lub miara nie może nigdy zawierać wartości puste.

#### **Dopuszczalność wartości pustych ustawiona na *unknown***

Funkcja API nie może stwierdzić, czy atrybut lub miara dopuszcza wartości puste, albo program narzędziowy nie był w stanie tego stwierdzić podczas przeprowadzania migracji z wersji 8.1 do wersji 8.2.

## **Modelowanie hierarchii za pomocą poziomów**

We wcześniejszych wersjach programu DB2 Cube Views hierarchie były modelowane na podstawie uporządkowanej listy atrybutów, która składała się z atrybutów i obiektów zwanych związkami atrybutów. W obecnej wersji programu hierarchie są modelowane na podstawie uporządkowanej listy obiektów poziomu, przy czym każdy poziom odwołuje się do co najmniej jednego atrybutu.

W tej wersji dostępny jest nowy Kreator poziomów i nowe okno właściwości poziomów, czyli narzędzia, za pomocą których można tworzyć i modyfikować nowe obiekty poziomów.

## **Zależności funkcjonalne obiektów poziomów**

Tworząc poziom, można zażądać, aby program DB2 Cube Views spróbował utworzyć obiekt zależności funkcjonalnych.

Zależność funkcjonalna oznacza, że atrybut domyślny i atrybuty pokrewne obiektu poziomu mogą być funkcjonalnie wyznaczane na podstawie atrybutów klucza poziomu. Doradca optymalizacyjny, korzystając z zależności funkcjonalnej, może zminimalizować wielkość zalecanych tabel podsumowań.

Jeśli nie można utworzyć zależności funkcjonalnej, zwracany jest komunikat z ostrzeżeniem. Na przykład zależności funkcjonalne nie mogą wskazywać kolumn obejmujących więcej niż jedną tabelę.

**Ograniczenie:** Wyniki zapytania mogą być niepoprawne, jeśli kolumny bazowe nie będą w rzeczywistości podlegały związkowi zdefiniowanemu w zależności funkcjonalnej dla danego obiektu poziomu. Ponieważ program DB2 UDB nie sprawdza poprawności danych w zakresie zależności funkcjonalnych, należy dbać o to, aby dane w kolumnach tabel były w rzeczywistości zależne funkcjonalnie w sposób, jaki został określony.

Aby uzyskać więcej informacji na temat zależności funkcjonalnych, patrz sekcja “Poziomy” na stronie 27.

## **Zautomatyzowane działanie Doradcy optymalizacyjnego**

W obecnej wersji programu dostępna jest nowa operacja API ADVISE. Operacja ta umożliwia zalecanie tabel podsumowań, które powinny zostać zbudowane w celu poprawienia wydajności zapytania dla danego modelu kostki. W argumentach tej operacji określa się ograniczenie czasu działania Doradcy optymalizacyjnego oraz ilości miejsca na dysku zajmowanego przez tabele podsumowań.

## **Zaktualizowana przykładowa baza danych o nazwie CVSAMPLE**

Dostępna jest nowa, lepsza przykładowa baza danych o nazwie CVSAMPLE. Baza danych CVSAMPLE jest bardziej stabilnym schematem typu płatek śniegu. Dostarczane są także metadane programu DB2 Cube Views odzwierciedlające zmiany w metadanych wprowadzone w tym wydaniu. Aby uzyskać informacje na temat konfigurowania bazy danych CVSAMPLE, patrz sekcja “Konfigurowanie przykładowej bazy danych CVSAMPLE” na stronie 8.

---

## Wprowadzenie do programu DB2 Cube Views

DB2 Cube Views jest składnikiem dodatkowym programu DB2 Universal Database, który usprawnia przetwarzanie OLAP programu DB2 UDB. Za pomocą programu DB2 Cube Views można usprawniać wdrażanie rozwiązań OLAP i zarządzać nimi oraz zwiększać wydajność narzędzi i aplikacji OLAP.

Używając programu DB2 Cube Views, można opisać strukturę wymiarową tabel relacyjnych i tworzyć konstrukcje OLAP. Umożliwia on przechowywanie informacji o strukturach oraz konstrukcjach OLAP jako wielowymiarowych metadanych w bazie danych DB2.

Korzystanie z nowych, wielowymiarowych metadanych w programie DB2 UDB daje dwie zasadnicze korzyści:

### **Poprawia przepływ wielowymiarowych metadanych między inteligentnymi narzędziami analitycznymi a aplikacjami**

Używając Centrum OLAP, stanowiącego interfejs graficzny, użytkownicy narzędzi hurtowni danych oraz inteligentnych narzędzi analitycznych mogą przechowywać wielowymiarowe metadane w bazie danych DB2 i udostępniać je wszystkim narzędziom i aplikacjom.

### **Zwiększa wydajność zapytań typu OLAP**

Na podstawie wielowymiarowych metadanych można tworzyć tabele podsumowań DB2, korzystając z zaleceń Doradcy optymalizacyjnego z Centrum OLAP. Tabele podsumowań zawierają wstępnie obliczone dane stanowiące odwzorowanie struktur OLAP. Zwiększy się wydajność zapytań generowanych z aplikacji hurtowni danych lub inteligentnej analizy danych korzystających z tej samej struktury OLAP.

W programie DB2 Cube Views wykorzystywane są takie opcje programu DB2, jak tabele podsumowań, różne schematy indeksów, operatory typu OLAP i funkcje agregujące. Udostępniane są następujące komponenty:

### **Obiekty wielowymiarowych metadanych**

Możliwe jest utworzenie zbioru obiektów metadanych w celu wymiarowego modelowania danych relacyjnych i struktur OLAP. Wszystkie utworzone obiekty metadanych programu DB2 Cube Views są przechowywane w katalogu DB2.

### **Centrum OLAP**

Przy użyciu Centrum OLAP można tworzyć modele kostek, kostki i inne obiekty metadanych, które mają być używane w narzędziach OLAP, operować nimi, importować je lub eksportować. Centrum OLAP udostępnia łatwych w użyciu kreatorów oraz okna pomagające w pracy z metadanymi. Na przykład Doradca optymalizacyjny analizuje metadane i zaleca sposób budowania tabel podsumowań, w których są przechowywane i indeksowane zagregowane dane dla zapytań SQL typu OLAP. Aby uzyskać informacje na temat uruchamiania Centrum OLAP, patrz sekcja Uruchamianie Centrum OLAP. Aby po uruchomieniu Centrum OLAP użyć kreatora Doradcy optymalizacyjnego, patrz sekcja Optymalizowanie modelu kostki.

### **Usługi wielowymiarowe**

Program DB2 Cube Views udostępnia twórcom aplikacji oparty na językach SQL i XML aplikacyjny interfejs programistyczny (API) do narzędzi OLAP. Korzystając z połączeń CLI, ODBC lub JDBC albo używając wbudowanego języka SQL programu DB2, aplikacje mogą tworzyć, modyfikować i pobierać obiekty metadanych za pomocą tylko jednej procedury zapisanej w bazie.

### **Przykładowe dane**

Aby ułatwić naukę korzystania z produktu, udostępniono przykładową aplikację i przykładową bazę danych.

Ponadto możliwa jest wymiana obiektów metadanych między katalogiem DB2 a narzędziami OLAP. W narzędziach OLAP i narzędziach baz danych są dostępne programy narzędziowe nazywane mostami metadanych, służące do importowania lub eksportowania obiektów metadanych z lub do katalogu DB2. Aby dowiedzieć się, czy w danym narzędziu OLAP lub narzędziu bazy danych jest dostępny most metadanych, należy zapoznać się dokumentacją tego narzędzia.

---

## Wymagania instalacyjne

Przed zainstalowaniem programu DB2 Cube Views należy się upewnić, że spełnione są wszystkie wymagania.

### Wymagania systemowe

Należy zainstalować odpowiednie komponenty na każdym kliencie i serwerze DB2 UDB, z którym ma zostać nawiązane połączenie. Na serwerze i na kliencie musi być zainstalowany program Multidimensional Services. Ponadto na kliencie można zainstalować Centrum OLAP.

**Ograniczenie:** Należy zainstalować tę samą wersję programu DB2 Cube Views dla wszystkich komponentów klienckich i serwerowych. Klient Centrum OLAP w wersji 8.1 może nawiązywać połączenie z serwerem DB2 UDB, wersja 8.2, lecz nie można wykonywać operacji tworzenia (Create), zmieniania (Alter) ani usuwania (Drop).

Muszą być zainstalowane niżej wymienione komponenty serwerowe, klienckie i sprzętowe.

- Komponent serwerowy:

#### Microsoft Windows

Windows NT 4, Windows 2000® (wersja 32-bitowa), Windows XP Professional (wersja 32-bitowa), Windows Server 2003 (wersja 32-bitowa) lub Windows Server 2003 (wersja 64-bitowa)

**AIX** 32-bitowy system AIX, wersja 4.3.3, 32-bitowy system AIX 5L lub 64-bitowy system AIX 5L

#### Linux®:

Linux Red Hat™ 8 (kernel 2.4.18, glibc 2.2.93-5), wersja 32-bitowa, Linux SuSE 8.0 (kernel 2.4.18, glibc 2.2.5), wersja 32-bitowa, Linux SLES 8 SP3 (kernel 2.4.21, glibc 2.2.5), wersja 32-bitowa, lub Linux RHEL 3 Update 2 (kernel 2.4.21-9, glibc 2.3.2.), wersja 32-bitowa.

Najnowsze informacje o poziomach dystrybucji i jądrach obsługiwanych przez program DB2 UDB dostępne są pod adresem: [www.ibm.com/db2/linux/validate](http://www.ibm.com/db2/linux/validate)

#### W systemie operacyjnym Sun Solaris™

Solaris 8 32-bitowy, Solaris 8 64-bitowy, Solaris 9 32-bitowy lub Solaris 9 64-bitowy

#### W systemie HP-UX

HP-UX 11i v2 (wersja 64-bitowa) dla Intel Itanium

- Komponent klienta: Windows NT 4, Windows 2000 (wersja 32-bitowa), Windows XP (wersja 32-bitowa), Windows XP (wersja 64-bitowa), Windows Server 2003 (wersja 32-bitowa) lub Windows Server 2003 (wersja 64-bitowa)
- Komponenty sprzętowe:
  - 500 MB miejsca na dysku
  - 256 MB RAM

## Wymaganie wstępne instalacji programu DB2 Cube Views

Przed zainstalowaniem programu DB2 Cube Views należy zainstalować niżej wymienione komponenty.

### Centrum informacyjne DB2

Aby mieć dostęp do pomocy elektronicznej dla Centrum OLAP oraz do elektronicznej wersji podręcznika *DB2 Cube Views Podręcznik i skorowidz*, należy zainstalować Centrum informacyjne DB2. Centrum informacyjne DB2 instaluje się z dysku CD Centrum informacyjne DB2.

### DB2 Universal Database, wersja 8.2

Przed zainstalowaniem programu DB2 Cube Views należy zainstalować program DB2 UDB, wersja 8.2.

**Uwaga:** Przed zainstalowaniem programu DB2 UDB, wersja 8.2, należy usunąć wszystkie wcześniejsze wersje DB2 UDB.

---

## Instalowanie programu DB2 Cube Views

Program DB2 Cube Views można zainstalować na platformach Windows, AIX, Linux oraz Środowisko Operacyjne Solaris przy użyciu Wyrzutni instalacji.

### Instalowanie programu DB2 Cube Views w systemie Windows

Przed zainstalowaniem programu DB2 Cube Views, należy się upewnić, że:

- Zainstalowano Centrum informacyjne DB2.
- Wykonano czystą instalację programu DB2 Universal Database, wersja 8.2.
- Spełnione są wszystkie wymagania instalacyjne.

Aby zainstalować program DB2 Cube Views w systemie Windows:

1. Włóż dysk CD z programem DB2 Cube Views. Program instalacyjny uruchomi się automatycznie.
2. W Wyrzutni kliknij przycisk **Uwagi do wydania**, aby zapoznać się z najnowszymi wymaganiami w odniesieniu do instalacji, dysku i pamięci. Ponadto w celu uzyskania dodatkowych instrukcji zapoznaj się z plikiem `readme.txt` znajdującym się w głównym katalogu dysku CD z programem DB2 Cube Views.
3. Kliknij przycisk **Instalacja produktów**, aby rozpocząć instalowanie, po czym postępuj według wyświetlanych instrukcji.

### Instalowanie programu DB2 Cube Views w systemie AIX, Linux lub Środowisko Operacyjne Solaris

Przed zainstalowaniem programu DB2 Cube Views, należy się upewnić, że:

- Wykonano czystą instalację programu DB2 Universal Database, wersja 8.2.
- Spełnione są wszystkie wymagania instalacyjne.

Aby zainstalować program DB2 Cube Views w systemie AIX, Linux lub Środowisku Operacyjnym Solaris:

1. Włóż dysk CD z programem DB2 Cube Views.
2. Przejdź do katalogu odpowiedniego dla posiadanego systemu operacyjnego UNIX, a następnie uruchom plik `db2setup`.

3. W Wyrzutni kliknij przycisk **Uwagi do wydania**, aby zapoznać się z najnowszymi wymaganiami w odniesieniu do instalacji, dysku i pamięci. Ponadto w celu uzyskania dodatkowych instrukcji zapoznaj się z plikiem readme.txt znajdującym się w głównym katalogu dysku CD z programem DB2 Cube Views.
4. Kliknij przycisk **Instalacja produktów**, aby rozpocząć instalowanie, po czym postępuj według wyświetlanych instrukcji.

---

## Przeprowadzanie migracji plików XML programu DB2 Cube Views z wersji 8.1 do 8.2

Migrację bazy danych dostosowanej do programu DB2 Cube Views z wersji 8.1 do wersji 8.2 można przeprowadzić ręcznie uruchamiając skrypt lub posługując się Centrum OLAP.

### Przeprowadzanie migracji z zastosowaniem skryptu DB2

**Zalecenie:** Przed uruchomieniem skryptu migracji utwórz kopię zapasową metadanych XML w wersji 8.1 i wyeksportuj je.

Do utworzenia nowych tabel metadanych i dodatkowego kodu SQL niezbędnego do przeprowadzenia migracji metadanych z wersji 8.1 do wersji 8.2 użyj pliku db2mdmigrate.sql znajdującego się w katalogu sqllib/misc. Aby użyć tego pliku:

1. Nawiąż połączenie z bazą danych, która ma być poddana migracji, wprowadzając następującą komendę:  
`db2 connect to nazwa_bazy_danych`
2. Aby uruchomić plik db2mdmigrate.sql, w oknie komend DB2 wprowadź następującą komendę:  
`db2 -tvf db2mdmigrate.sql`

W tym skrypcie nie ma obsługi błędów. Podczas wykonywania tego skryptu realizowany jest zestaw instrukcji DDL i SQL. Jeśli którakolwiek instrukcja nie powiedzie się lub jeśli skrypt zostanie przedwcześnie przerwany, migracja będzie wykonana tylko częściowo i w konsekwencji program DB2 Cube Views może nie działać poprawnie.

3. Jeśli wystąpią błędy, wykonaj następujące zadania:
  - a. Usuń wszystkie tabele DB2INFO.\*.
  - b. Usuń procedurę zapisaną w bazie DB2INFO.MD\_MESSAGE.
  - c. Ponownie utwórz tabele metadanych, używając pliku db2mdapi.sql znajdującego się w katalogu sqllib/misc.

### Przeprowadzanie migracji Centrum OLAP

Kiedy Centrum OLAP nawiązuje połączenie z bazą danych DB2 UDB, automatycznie wykrywa bieżącą wersję programu DB2 UDB i określa, czy tabele metadanych muszą być poddane migracji. Jeśli Centrum OLAP określi, że migracja jest potrzebna, w celu poinformowania o tym użytkownika wyświetli komunikat o błędzie.

Jeśli zalecenie migracji przedstawione przez Centrum OLAP zostanie zaakceptowane, Centrum OLAP połączy się z bazą danych i dokona migracji tabel metadanych do wersji 8.2. W wypadku zrezygnowania z migracji, Centrum OLAP nie połączy się z bazą danych.

Jeśli w trakcie migracji wystąpią jakiegokolwiek błędy, Centrum OLAP wycofa transakcję i baza danych nie będzie poddana migracji.

---

## Konfigurowanie bazy danych dla programu DB2 Cube Views

Nową bazę danych można skonfigurować pod kątem użycia programu DB2 Cube Views.

Konfigurowanie bazy danych obejmuje:

- Rejestrowanie procedury zapisanej w bazie programu DB2 Cube Views z bazą danych
- Tworzenie tabel katalogu metadanych dla programu DB2 Cube Views

Gdy użytkownik po raz pierwszy zaloguje się w bazie danych, która nie została skonfigurowana pod kątem programu DB2 Cube Views przy użyciu Centrum OLAP, Centrum OLAP wstępnie skonfiguruje bazę danych. Bazę danych można także skonfigurować samodzielnie, używając pliku db2mdapi.sql.

**Ważne:** Do pliku db2mdapi.sql nie należy wprowadzać zmian, gdyż skutki tej czynności są nieprzewidywalne.

Aby skonfigurować bazę danych przy użyciu pliku db2mdapi.sql:

1. Otwórz okno komend DB2 i nawiąż połączenie z bazą danych.
2. Przejdź do katalogu SQLLIB\misc i w oknie komend DB2 wprowadź następującą komendę:

```
db2 -tvf db2mdapi.sql
```

Uruchom skrypt db2mdapi.sql tylko raz dla danej bazy danych. Jeśli wystąpią błędy, usuń problem, a następnie wykonaj poniższe zadania.

- a. Usuń wszystkie tabele DB2INFO.\*.
- b. Usuń procedurę zapisaną w bazie DB2INFO.MD\_MESSAGE.
- c. Ponownie utwórz tabele metadanych, używając pliku db2mdapi.sql znajdującego się w katalogu sqllib\misc.

---

## Konfigurowanie przykładowej bazy danych CVSAMPLE

Z programem DB2 Cube Views dostarczane są przykładowe dane, których można użyć do utworzenia przykładowej bazy danych o nazwie CVSAMPLE.

Na te przykładowe dane składa się zestaw tabel zawierających dane fikcyjnej firmy sprzedającej napoje. Dołączono także zestaw obiektów metadanych opisujących przykładowe tabele danych. Dostarczona baza danych CVSAMPLE stanowi ulepszoną i rozwiniętą wersję bazy danych MDSAMPLE dostarczanej w wersji 8.1. Większość przykładów zawarta w publikacji *DB2 Cube Views Podręcznik i skorowidz* jest oparta na bazie danych CVSAMPLE i odpowiadającym jej modelowi kostki.

Przykładową bazę danych CVSAMPLE tworzy się i zapełnia, otwierając okno komend DB2 i wprowadzając w nim następujące komendy:

1. Utwórz przykładową bazę danych o nazwie CVSAMPLE:

```
db2 create db cvsample
```
2. Nawiąż połączenie z bazą danych:

```
db2 connect to cvsample
```
3. Uruchom skrypt db2mdapi.sql, aby skonfigurować bazę danych do pracy z programem DB2 Cube Views. Zmień katalog na SQLLIB\misc, a następnie wprowadź następującą komendę DB2:

```
db2 -tvf db2mdapi.sql
```
4. Zmień katalog na SQLLIB\samples\olap\cvsample, a następnie wprowadź następującą komendę DB2, aby utworzyć tabele bazy danych CVSAMPLE:



```
db2 -tvf CVSampleTables.sql
```

Po utworzeniu bazy danych CVSAMPLE można utworzyć obiekty metadanych programu DB2 Cube Views, importując definicje z pliku XML wyeksportowanego z aplikacji do inteligentnej analizy danych.

Na przykład poniższa procedura powoduje wypełnienie tabel katalogu programu DB2 Cube Views kompletnym opisem bazy danych CVSAMPLE.

Aby zaimportować metadane bazy danych CVSAMPLE:

1. Uruchom Centrum OLAP i nawiąż połączenie z bazą danych CVSAMPLE.
2. Kliknij opcje: **Centrum OLAP** → **Importuj**.
3. Odszukaj plik CVSampleMetadata.xml znajdujący się w katalogu SQLLIB/samples/olap/cvsample. Kliknij przycisk **Dalej**.

Obiekty metadanych można przeglądać w Centrum OLAP. Informacje na temat korzystania z Centrum OLAP można znaleźć w sekcji “Tworzenie obiektów metadanych programu DB2 Cube Views” na stronie 47.

---

## Rozwiązywanie problemów z połączeniem z bazą danych

Jeśli nie można nawiązać połączenia z bazą danych przy użyciu Centrum OLAP, należy sprawdzić, czy wersja programu DB2 Cube Views jest zgodna z wersją tabel metadanych w katalogu DB2.

Poniższa tabela przedstawia, jak zachowuje się Centrum OLAP, gdy wersje programu DB2 Cube Views i tabel metadanych w katalogu DB2 nie są zgodne.

Wersja programu DB2 Cube Views	Wersja tabel metadanych w katalogu DB2	Zachowanie Centrum OLAP
Nie zainstalowano	Nie dotyczy	Połączenie nie powiedzie się i Centrum OLAP wyświetli komunikat o błędzie
Wersja 8.1	Brak	Połączenie nie powiedzie się i Centrum OLAP wyświetli komunikat o błędzie
Wersja 8.1	Wersja 8.1	Połączenie nie powiedzie się i Centrum OLAP wyświetli komunikat o błędzie
Wersja 8.2	Brak	Centrum OLAP może skonfigurować bazę danych w celu dostosowania jej do użytkowania z programem DB2 Cube Views, wersja 8.2
Wersja 8.2	Wersja 8.1	Centrum OLAP może dokonać migracji bazy danych w celu dostosowania jej do użytkowania z programem DB2 Cube Views, wersja 8.2
Wersja 8.2	Wersja 8.2	Połączenie powiedzie się.



---

## Rozdział 2. Obiekty metadanych w programie DB2 Cube Views

W obiektach metadanych programu DB2 Cube Views tabele relacyjne są opisywane jako struktury OLAP, jednak różnią się one od tradycyjnych obiektów OLAP. Obiekty metadanych przechowują metadane o danych w tabelach podstawowych, informują, gdzie znajdują się trwałe dane, oraz opisują związki w danych podstawowych.

Program DB2 Cube Views przechowuje informacje o danych relacyjnych w obiektach metadanych, które umożliwiają ogląd danych z odmiennej perspektywy. Program DB2 Cube Views rozszerza katalog DB2 tak, że zawiera on nie tylko informacje o tabelach i kolumnach, lecz również relacjach między tabelami i kolumnami a obiektami OLAP oraz o związkach między tymi obiektami metadanych.

Niektóre obiekty metadanych zapewniają bezpośredni dostęp do danych relacyjnych w drodze agregacji danych lub jako bezpośrednie odpowiedniki konkretnych kolumn w tabelach relacyjnych. Inne obiekty metadanych opisują związki między podstawowymi obiektami metadanych lub łączą je ze sobą. Wszystkie obiekty metadanych można grupować na podstawie związków między nimi w obiekt metadanych nazywany modelem kostki. Zasadniczo model kostki stanowi reprezentację określonej konfiguracji tabel relacyjnych i zasady ich grupowania.

Program DB2 Cube Views zarządza następującymi obiektami metadanych, i zapisuje je w katalogu DB2:

- modele kostek
- obiekty faktów
- wymiary
- hierarchie
- poziomy
- miary
- atrybuty
- związki atrybutów
- łączenia
- kostki
- obiekty faktów kostki
- wymiary kostki
- hierarchie kostki
- poziomy kostki

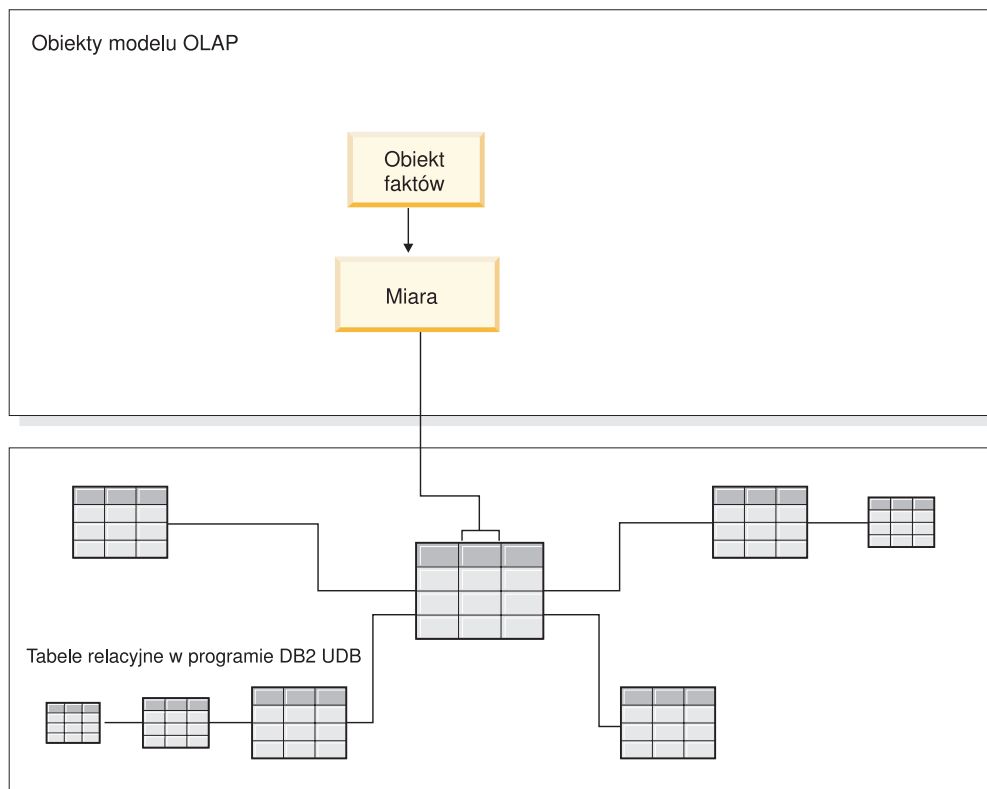
---

### Obiekty metadanych mające odwzorowanie w tabelach relacyjnych

Model kostki może być skonstruowany na wiele sposobów, przy czym często jest oparty na relacyjnym schemacie gwiazdzistym lub schemacie typu płatek śniegu. Model kostki zawiera obiekty metadanych opisujące związki między danymi relacyjnymi.

W schemacie gwiazdzistym tabela faktów znajduje się pośrodku i jest z nią połączona przynajmniej jedna tabela wymiarów. Schemat typu płatek śniegu jest natomiast rozszerzeniem schematu gwiazdzistego polegającym na tym, że co najmniej jeden wymiar jest zdefiniowany przez wiele tabel. Model kostki, który jest oparty na prostym schemacie gwiazdzistym, jest zbudowany wokół centralnego obiektu faktów. Obiekt faktów zawiera

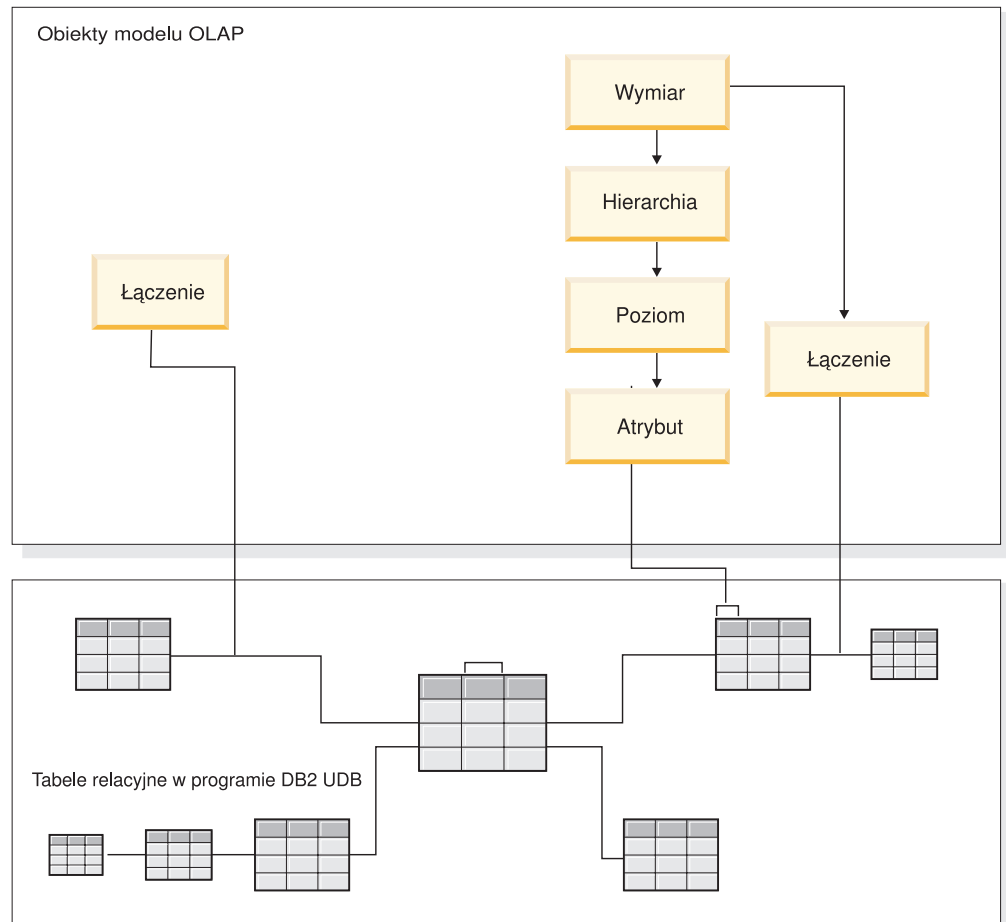
zestaw miar opisujących sposób agregowania danych z tabeli faktów po wymiarach. Miary opisują obliczenia przeprowadzane na danych z kolumn w tabelach relacyjnych i są dołączane w celu utworzenia obiektu faktów. Rys. 1 przedstawia związki między miarami i obiektami faktów a danymi relacyjnymi.



Rysunek 1. Obiekt faktów. Jakie są związki między obiektami faktów i miarami a danymi relacyjnymi

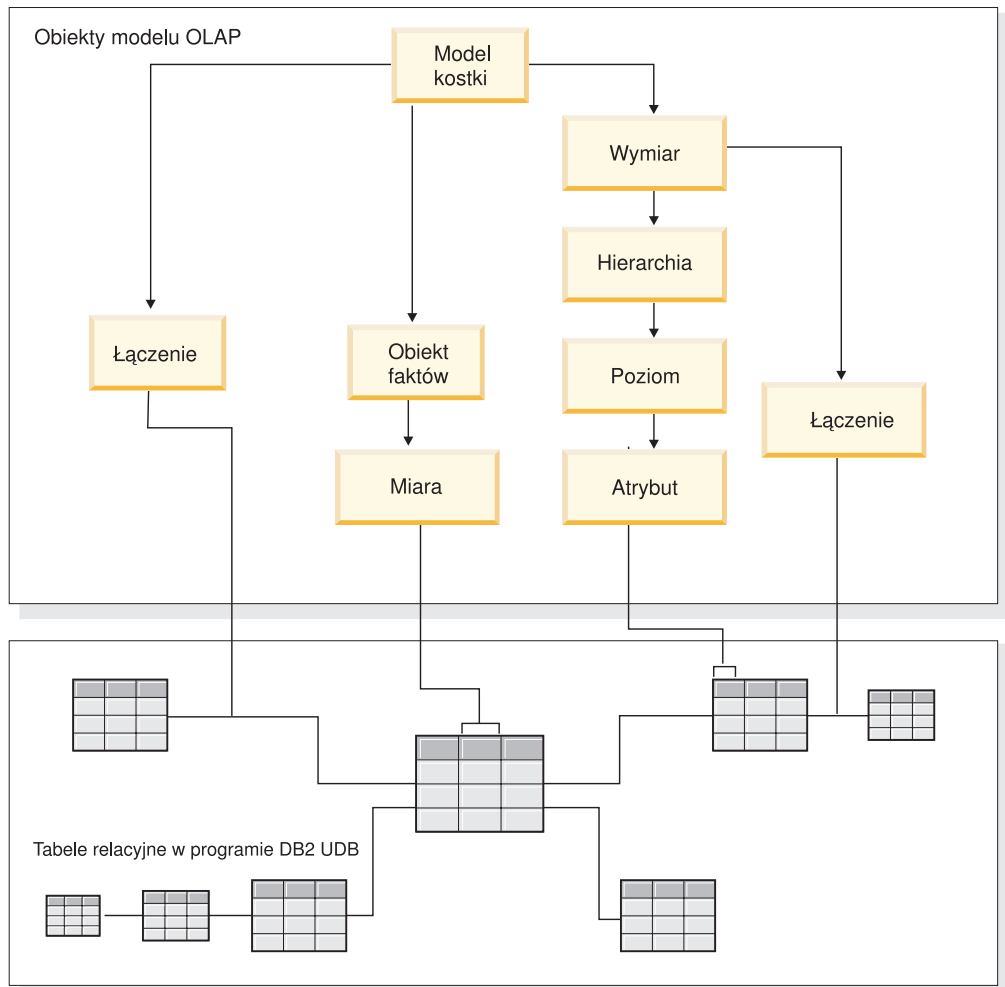
Wymiary są połączone z obiektem faktów w modelu kostki, tak jak tabele wymiarów są połączone z tabelą faktów w schemacie gwiazdzistym. Kolumny danych z tabel relacyjnych są reprezentowane przez atrybuty zorganizowane w celu utworzenia wymiaru.

Rys. 2 na stronie 13 pokazuje, jak wymiary są budowane z tabel relacyjnych. W hierarchiach przechowywane są informacje o powiązaniach między poziomami i ich strukturze w ramach wymiaru. Hierarchia umożliwia przeprowadzanie obliczeń na wymiarze i poruszanie się po nim. Każdemu wymiarowi odpowiada hierarchia zawierająca poziomy i zestawy powiązanych z nimi atrybutów. W modelu kostki każdy wymiar może mieć wiele hierarchii.



Rysunek 2. Wymiar. Jak wymiary są budowane z tabel relacyjnych

Wszystkie wymiary są powiązane z obiektem faktów w modelu kostki, zgodnie ze schematem gwiazdowym lub schematem typu płatek śniegu. Tabele mogą być połączone łączeniami, tworząc obiekt faktów lub wymiar. W modelu kostki łącenia mogą łączyć obiekty faktów z wymiarami. Wymiary odwołują się do odpowiednich hierarchii, poziomów, atrybutów i związanych z nimi łąceń. Obiekty faktów odwołują się do swoich miar, atrybutów i powiązanych z nimi łąceń. Rys. 3 na stronie 14 pokazuje układ obiektów metadanych w modelu kostki i ich odwzorowanie w relacyjnym schemacie typu płatek śniegu.



Rysunek 3. Model kostki. Układ obiektów metadanych w modelu kostki i ich odwzorowanie w relacyjnym schemacie typu płatek śniegu

Komponenty modelu kostki można wykorzystywać wielokrotnie do tworzenia kostek przeznaczonych do konkretnych zastosowań. Kostka jest najbardziej precyzyjnym obiektem metadanych i obiektem najbliższym koncepcji kostki OLAP. Kostka jest określoną instancją lub podzbiorem modelu kostki. Kostka zawiera zestaw podobnych, lecz bardziej restrykcyjnie określonych obiektów metadanych, pochodnych nadrzędnego modelu kostki, takich jak wymiary kostki, hierarchie kostki, poziomy kostki i obiekt faktów kostki. W kostce na jeden wymiar kostki może przypadać tylko jedna zdefiniowana hierarchia kostki, lecz w wymiarze na jeden model kostki może przypadać wiele zdefiniowanych hierarchii. Ze względu na tę strukturalną różnicę między kostką a modelem kostki większość kostek można pobrać przy użyciu pojedynczej instrukcji SQL.

## Przykład modelu kostki mającego odwzorowanie w tabelach relacyjnych

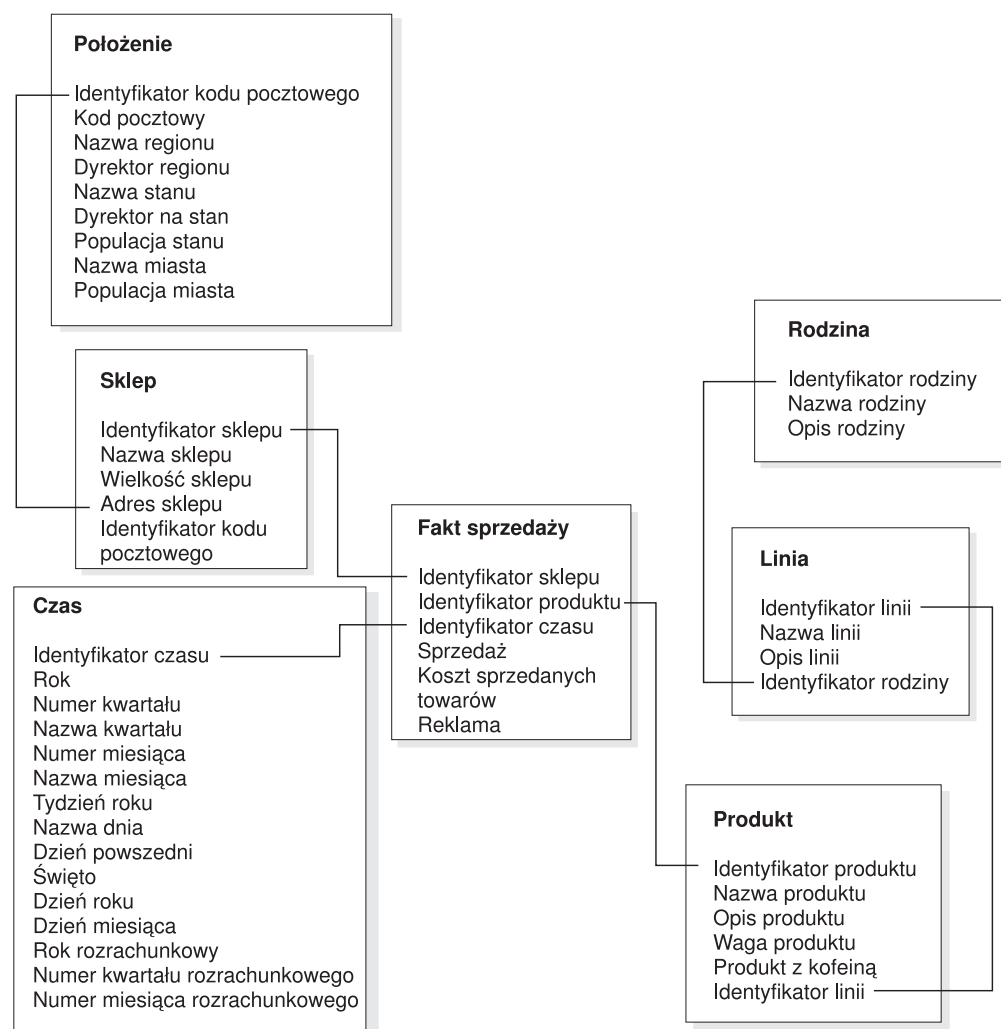
Model kostki i odpowiednie zestawy metadanych są już zdefiniowane dla bazy danych CVSAMPLE. Użyto tu schematu typu płatek śniegu.

Rys. 4 na stronie 15 przedstawia schemat typu płatek śniegu z tabelą faktów Sprzedaż oraz tabelami wymiarów Sklep, Położenie, Czas, Produkt, Linia i Rodzina. Wymiar Rynek ma dwie tabele wymiarów, przy czym Sklep jest podstawową tabelą wymiarów, a Położenie

tabelą pomocniczą. Wymiar Produkcja ma trzy tabele wymiarów, z których Produkt jest tabelą podstawową, a Linia i Rodzina to tabele pomocnicze.

Klucz podstawowy w każdej podstawowej tabeli wymiarów (Sklep, Czas i Produkt) zostaje połączony z odpowiadającym mu kluczem obcym w tabeli faktów Sprzedaż. Na przykład Sklep.IdSklepu = Sprzedaż.IdSklepu, Czas.IdCzasu = Sprzedaż.IdCzasu i Produkt.IdProduktu = Sprzedaż.IdProduktu.

W wymiarach płatka śniegu klucz podstawowy w każdej tabeli wymiarów zostaje połączony z odpowiadającym mu kluczem obcym w innej tabeli wymiarów. Na przykład Położenie.IdKoduPocztowego = Sklep.IdKoduPocztowego, Rodzina.IdRodziny = Linia.IdRodziny i Linia.IdLinii = Produkt.IdLinii.



Rysunek 4. Schemat typu płatek śniegu. Przykład oparty na schemacie typu płatek śniegu CVSAMPLE

Model kostki oparty na schemacie typu płatek śniegu CVSAMPLE jest zbudowany wokół obiektu faktów Sprzedaż, opisującego zagregowane dane relacyjne z tabeli faktów Sprzedaż. Miary opisują sposób obliczania danych z kolumn w tabeli Sprzedaż. Obiekt faktów obejmuje także atrybuty odpowiadające kluczom obcym w tabeli faktów używanej do łączenia wymiarów z obiektem faktów. W tym przykładzie obiekt faktów sprzedaży ma siedem miar: Sprzedaż, Koszt sprzedanych towarów, Reklama, Wydatki łącznie, Korelacja

reklama-sprzedaż, Zysk i Marża. Obiekt faktów sprzedaży ma trzy atrybuty: Identyfikator sklepu (fakt sprzedaży), Identyfikator czasu (fakt sprzedaży) i Identyfikator produktu (fakt sprzedaży).

Wymiary są połączone z obiektem faktów w modelu kostki, tak jak tabele wymiarów są połączone z tabelą faktów w schemacie gwiazdzistym. Kolumny danych z tabel relacyjnych są reprezentowane przez obiekty atrybutów, do których odwołuje się wymiar.

Wymiar Produkt odwołuje się do następujących atrybutów:

- Identyfikator rodziny
- Nazwa rodziny
- Opis rodziny
- Identyfikator linii
- Nazwa linii
- Opis linii
- Identyfikator produktu
- Nazwa produktu
- Opis produktu
- Waga produktu
- Produkt z kofeiną

Wymiar Czas odwołuje się do następujących atrybutów:

- Rok
- Nazwa kwartału
- Numer kwartału
- Nazwa miesiąca
- Numer miesiąca
- Identyfikator czasu
- Dzień miesiąca
- Nazwa dnia
- Dzień tygodnia
- Święto
- Dzień powszedni
- Rok rozrachunkowy
- Nazwa kwartału rozrachunkowego
- Numer kwartału rozrachunkowego
- Miesiąc rozrachunkowy

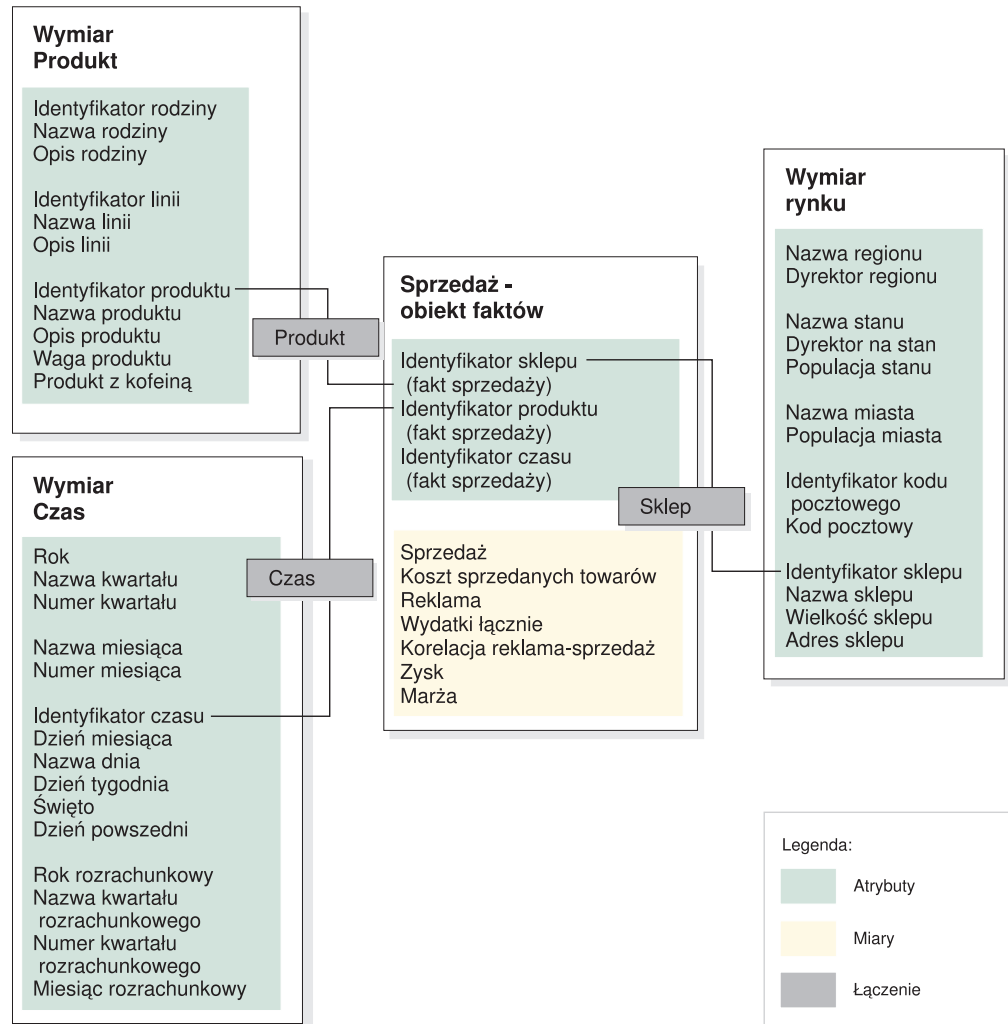
Wymiar Produkt odwołuje się do następujących atrybutów:

- Nazwa regionu
- Dyrektor regionu
- Nazwa stanu
- Dyrektor na stan
- Populacja stanu
- Nazwa miasta
- Populacja miasta
- Identyfikator kodu pocztowego



- Kod pocztowy
- Identyfikator sklepu
- Nazwa sklepu
- Wielkość sklepu
- Adres sklepu.

Łączenie zostaje utworzone tak, aby połączyło każdy wymiar z obiektem faktów. W przykładzie tym występują trzy łączenia: Produkt, Czas i Sklep. Rys. 5 przedstawia opisywany w tym przykładzie model kostki.

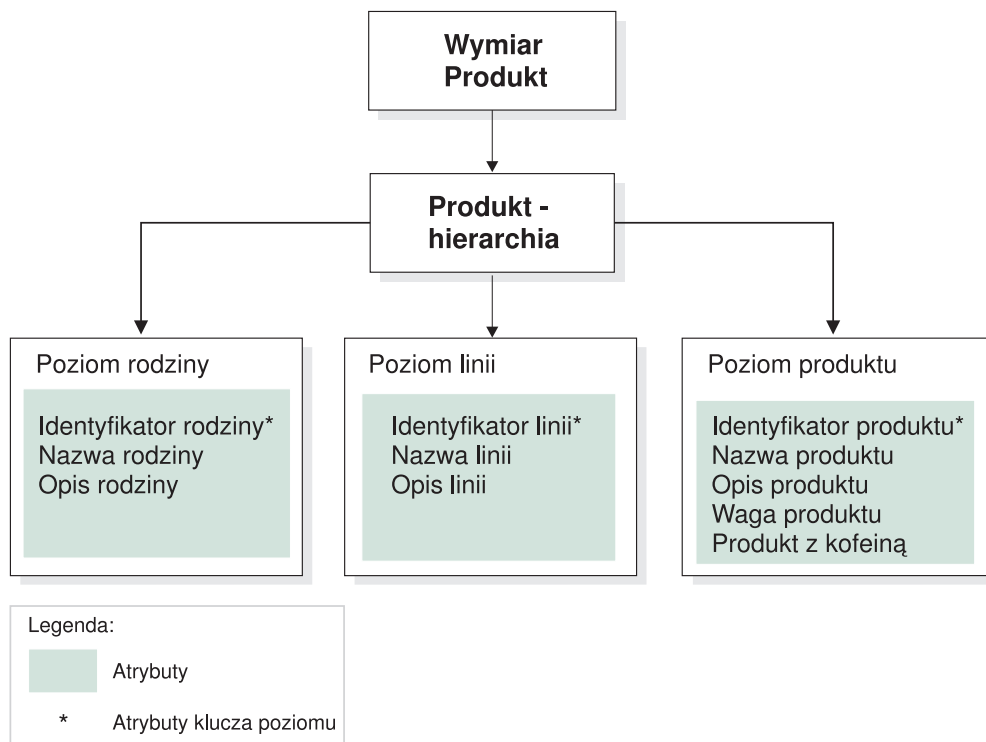


Rysunek 5. Model kostki. Przykładowy model kostki oparty na schemacie gwiazdowym CVSAMPLE

W hierarchiach przechowywane są informacje o tym, w jaki sposób atrybuty - pogrupowane w poziomy w ramach wymiaru - są ze sobą powiązane i jaką tworzą strukturę. Hierarchia, jako obiekt metadanych, umożliwia dokonywanie obliczeń na wymiarze oraz poruszanie się po nim. Każdy wymiar ma odpowiadającą mu hierarchię zawierającą poziomy grupujące powiązane ze sobą atrybuty. W modelu kostki każdy wymiar może mieć wiele hierarchii.

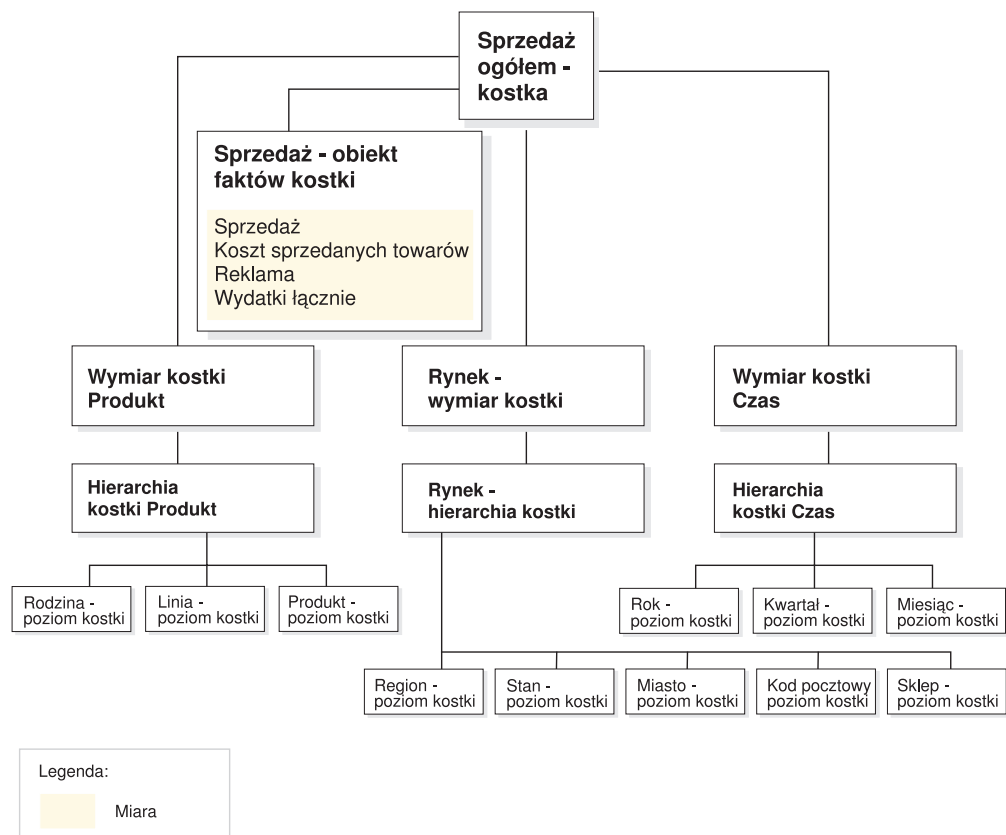
Hierarchia Produkt zawiera wszystkie atrybuty w wymiarze Produkt, jak to przedstawia Rys. 6 na stronie 18. Atrybuty w wymiarze Produkt są pogrupowane w trzech poziomach. Poziom Rodzina jest najwyższym poziomem hierarchii Produkt. Poziom Rodzina zawiera

atrybut klucza poziomemu Identyfikator rodziny, atrybut domyślny Nazwa rodziny oraz atrybut pokrewny Opis rodziny. Drugi poziom - Linia - obejmuje atrybut kluczowy Identyfikator linii, atrybut domyślny Nazwa linii oraz atrybut pokrewny Opis linii. Poziom najniższy - Produkt - zawiera atrybut klucza poziomemu Identyfikator produktu, atrybut domyślny Nazwa produktu oraz atrybuty pokrewne Opis produktu, Waga produktu i Produkt z kofeiną.



Rysunek 6. Wymiar. Przykładowy wymiar oparty na wymiarze Produkt w bazie danych CVSAMPLE

Ponadto według jednego modelu kostki można zbudować jedną lub więcej niż jedną kostkę. Baza danych CVSAMPLE zawiera dwie kostki, lecz tutaj opisujemy wyłącznie kostkę Sprzedaż ogółem. Kostkę Sprzedaż ogółem przedstawia Rys. 7 na stronie 19. Obiekt faktów kostki odwołuje się do podzbioru miar (Sprzedaż, Koszt sprzedanych towarów, Reklama i Wydatki łącznie) z obiektu faktów modelu kostki. Kostka ma trzy wymiary kostki, a każdy z nich odwołuje się do jednego z trzech wymiarów z modelu kostki. Wymiar Produkt zawiera hierarchię kostki Produkt, w której poziomy kostki odwołują się do poziomów Rodzina, Linia i Produkt. Wymiar kostki Rynek zawiera hierarchię kostki Rynek z poziomami kostki odwołującymi się do poziomów Region, Stan, Miasto, Kod pocztowy i Sklep. Wymiar kostki Czas zawiera hierarchię kostki Czas z poziomami kostki odwołującymi się do podzbioru poziomów Rok, Kwartał i Miesiąc. W tej kostce na każdy wymiar kostki przypada tylko jedna zdefiniowana hierarchia. (W kostce na jeden wymiar kostki może przypadać tylko jedna hierarchia kostki).



Rysunek 7. Kostka. Przykładowa kostka oparta na kostce Sprzedaż ogółem w bazie danych CVSAMPLE

## Wspólne właściwości obiektów metadanych

Każdy obiekt metadanych ma zestaw wspólnych właściwości oraz właściwości specyficznych dla danego obiektu metadanych. Wspólne właściwości służą do identyfikowania instancji obiektu metadanych, do opisywania użycia lub roli instancji obiektu metadanych oraz do śledzenia zmian w instancjach obiektu metadanych. Obiektom metadanych nazwy są nadawane z zastosowaniem schematu tak samo, jak w wypadku pozostałych obiektów DB2. Jeśli do obiektu metadanych nie ma być stosowany domyślny schemat nazwy użytkownika, należy podać pełną kwalifikację obiektu metadanych z konkretną nazwą schematu.

W poniższej tabeli opisano wspólne właściwości wszystkich obiektów metadanych.

Tabela 1. Wspólne właściwości obiektów metadanych

Właściwość	Opis
Nazwa	Nazwa obiektu metadanych.
Schemat	Schemat, do którego należy obiekt metadanych.
Nazwa biznesowa	Nazwa przedstawiana użytkownikowi. Tej nazwy można używać jako nazwy więcej mówiącej użytkownikowi o obiekcie.
Komentarze	Opis tekstowy lub komentarz dotyczący rodzaju lub zastosowania obiektu metadanych.
Czas utworzenia	Czas, w którym obiekt metadanych został utworzony.

Tabela 1. Wspólne właściwości obiektów metadanych (kontynuacja)

Właściwość	Opis
Utworzone przez	Użytkownik (schemat), który zdefiniował obiekt metadanych.
Czas modyfikacji	Czas, w którym obiekt metadanych został ostatnio zmodyfikowany.
Zmodyfikowane przez	Użytkownik (schemat), który dokonał modyfikacji.

Oprócz wspólnego zestawu właściwości każdy obiekt metadanych ma zestaw specyficznych właściwości. Te specyficzne właściwości opisują komponenty i cechy jakościowe definiujące obiekt metadanych. Informacje na temat specyficznych właściwości poszczególnych obiektów metadanych zawarte są w tematach poświęconych tym obiektom.

### Konwencje nazewnictwa obiektów metadanych

Program DB2 UDB umożliwia stosowanie dwóch konwencji nazewnictwa obiektów: zwykłej i z ogranicznikami. W odniesieniu do obiektów metadanych konwencji z ogranicznikami używa się do nadawania nazw obiektom i odwoływania się do tabel i kolumn DB2. Konwencja z ogranicznikami umożliwia stosowanie nazw z literami różnej wielkości, z odstępami i znakami specjalnymi, takimi jak litery ze znakami diakrytycznymi. Kompletny zestaw znaków jest określony przez stronę kodową bazy danych, w której obiekty metadanych są przechowywane.

Do obiektów metadanych stosuje się następujące konwencje:

Tabela 2. Konwencje nazewnictwa obiektów metadanych

Obiekt	Konwencja
Schemat	<ul style="list-style-type: none"> <li>Długość: 1-30 bajtów</li> <li>Nazwy zastrzeżone: nazwą schematu nie może być <i>SESSION</i> ani nie może się ona zaczynać ciągiem <i>SYS</i>. Zastrzeżone są tylko nazwy zapisane wielkimi literami.</li> </ul>
Nazwa obiektu metadanych	<ul style="list-style-type: none"> <li>Długość: 1-128 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>
Nazwa biznesowa obiektu metadanych	<ul style="list-style-type: none"> <li>Długość: 1-128 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>
Komentarze do obiektów metadanych	<ul style="list-style-type: none"> <li>Długość: 0-254 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>
Schemat tabeli używany w odwołaniach do kolumn	<ul style="list-style-type: none"> <li>Długość: 1 do 128 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>
Nazwa tabeli używana w odwołaniach do kolumn	<ul style="list-style-type: none"> <li>Długość: 1 do 128 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>
Nazwa kolumny używana w odwołaniach do kolumn	<ul style="list-style-type: none"> <li>Długość: 1-128 bajtów</li> <li>Żadnych innych ograniczeń</li> </ul>

## Modele kostek

Model kostki w programie DB2 Cube Views stanowi reprezentację logicznego schematu gwiazdowego lub schematu typu płatek śniegu, grupując odpowiednie obiekty wymiarów wokół położonego centralnie obiektu faktów.

Każdy wymiar może mieć wiele hierarchii. Model kostki odwołuje się do informacji strukturalnych dotyczących sposobu łączenia tabel używanych przez obiekty faktów i wymiarów. Ponadto w modelu kostki zapisane są informacje wymagane do konstruowania zapytań SQL i pobierania danych OLAP. Z modelu kostki korzystają różne narzędzia raportujące i OLAP, które mogą interpretować jego zawartość i wyświetlać różne widoki poszczególnych wymiarów.

Modele kostki definiują złożony zestaw związków i mogą służyć do selektywnego uwidaczniania odpowiednich obiektów faktów i wymiarów w aplikacji. Każdy obiekt łączenia łączący wymiar z centralnym obiektem faktów jest przechowywany w parze z odpowiednim wymiarem. Podzbiory komponentów modelu kostki mogą być używane w wielu kostkach w celu przeprowadzania różnych analiz.

W Centrum OLAP można utworzyć pusty model kostki, używając Kreatora modeli kostek. Pusty model kostki nie zawiera obiektu faktów ani żadnych wymiarów. Przy użyciu kreatorów w Centrum OLAP można zestawić model kostki, tworząc obiekt faktów i co najmniej jeden wymiar. Kompletny model kostki można także utworzyć przy użyciu Kreatora szybkiego startu. Program DB2 Cube Views sprawdza poprawność modelu kostki podczas otwierania Doradcy optymalizacyjnego. Aby można było zoptymalizować model kostki, musi on zawierać następujące składniki:

- obiekt faktów,
- co najmniej jeden wymiar,
- hierarchia zdefiniowana dla przynajmniej jednego wymiaru,
- łączenia między istniejącymi obiektami faktów i wymiarami,
- atrybuty odwołujące się do istniejących kolumn tabeli.

Opis właściwości specyficznych dla modeli kostek przedstawiono w poniższej tabeli.

*Tabela 3. Właściwości modelu kostki*

Właściwość	Opis
Obiekt faktów	Obiekt faktów używany w modelu kostki
Para (wymiar, łączenie)	Wymiary używane w modelu kostki i odpowiadające im łączenia

## Obiekty faktów

Obiekt faktów jest używany w modelu kostki jako środek schematu gwiazdowego. Grupuje on wokół siebie powiązane miary wykorzystywane przez daną aplikację.

Obiekt faktów odwołuje się do atrybutów używanych w łączeniach fakt-wymiar oraz do atrybutów i łączy używanych do odwzorowywania dodatkowych miar wśród wielu tabel baz danych. Relacyjne tabele faktów można łączyć na podstawie określonych atrybutów w celu odwzorowania dodatkowych miar pokrewnych. Dlatego oprócz zestawu miar obiekt faktów przechowuje zestaw atrybutów oraz zestaw łączy.

Obiekt faktów można utworzyć przy użyciu Kreatora faktów w Centrum OLAP. W Kreatorze faktów należy określić więcej tabel faktów oraz wszelkie niezbędne łączenia, miary i agregacje miar.

Opis specyficznych właściwości obiektu faktów przedstawiono w poniższej tabeli.

*Tabela 4. Właściwości obiektu faktów*

Właściwość	Opis
Zestaw miar	Zbiór wszystkich miar pokrewnych w obiekcie faktów

Tabela 4. Właściwości obiektu faktów (kontynuacja)

Właściwość	Opis
Zestaw atrybutów	Zbiór wszystkich atrybutów używanych w obiekcie faktów
Zestaw łączy	Zbiór wszystkich łączy potrzebnych do łączenia wszystkich określonych miar i atrybutów

## Wymiary

Wymiary umożliwiają kategoryzowanie zbiorów pokrewnych atrybutów, które razem opisują poszczególne aspekty miary. Wymiary służą w modelach kostek do organizowania danych w obiekcie faktów zgodnie z kategoriami logicznymi, takimi jak Region, Produkt lub Czas.

Wymiary odwołują się do dowolnej liczby hierarchii. Hierarchie opisują związki i strukturę atrybutów pogrupowanych w poziomy i stanowią niezbędną ramę do nawigacyjnego i obliczeniowego operowania na wymiarze.

Pokrewne atrybuty i łączenia wymagane do grupowania tych atrybutów są zdefiniowane we właściwościach wymiaru.

Wymiary mają też przypisany typ informujący, czy wymiar jest czasowy. Na przykład wymiar o nazwie Czas, zawierający takie poziomy, jak Rok, Kwartał i Miesiąc, będzie poziomem typu czasowego. Natomiast wymiar o nazwie Rynek, zawierający poziomy takie jak Region Stan i Miasto, będzie poziomem typu zwykłego. Informacje o typie mogą być używane przez aplikacje do inteligentnej i optymalnej realizacji funkcji opartych na parametrach czasowych.

Nowy wymiar w kontekście modelu kostki lub bez odwołania do modelu kostki można utworzyć przy użyciu Kreatora wymiarów w Centrum OLAP. Wymiary mogą być wymieniane między modelami kostki przez dodawanie istniejącego wymiaru do modelu kostki za pomocą kreatora Dodaj wymiar.

Opis specyficznych właściwości wymiarów przedstawiono w poniższej tabeli.

Tabela 5. Właściwości wymiaru

Właściwość	Opis
Zestaw atrybutów	Zbiór wszystkich atrybutów używanych w wymiarze.
Zestaw łączy	Zbiór wszystkich łączy wymaganych do połączenia wszystkich określonych atrybutów. Wymieniono tu tylko te łączenia, które są wymagane do łączenia tabel wymiarów.
Zestaw hierarchii	Zestaw hierarchii mających zastosowanie w wymiarze.
Zestaw poziomów	Zestaw poziomów, do których odwołuje się wymiar.
Typ	Typ wymiaru: zwykły lub czasowy

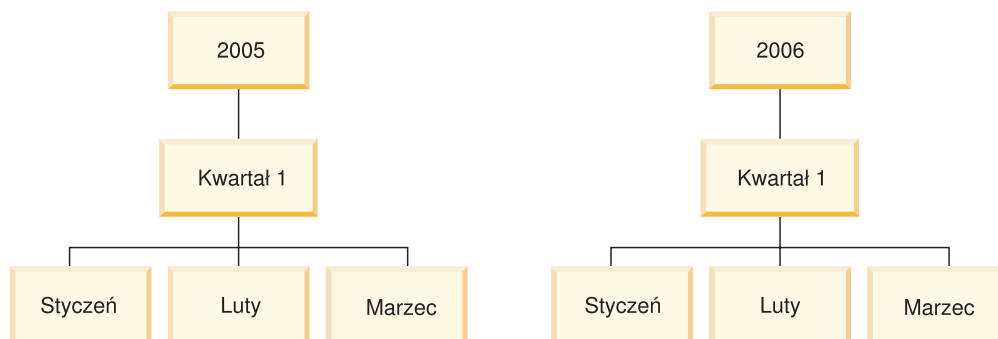
## Hierarchie

Hierarchia definiuje związki w obrębie zestawu atrybutów, grupowanych w poziomy w ramach wymiaru w modelu kostki. Te związki umożliwiają poruszanie się po wymiarach i przeprowadzanie na nich obliczeń. Dla jednego wymiaru modelu kostki można zdefiniować wiele hierarchii.

Typ hierarchii opisuje związek między poziomami wewnątrz hierarchii. Obsługiwane są cztery typy hierarchii:

### Zrównoważona

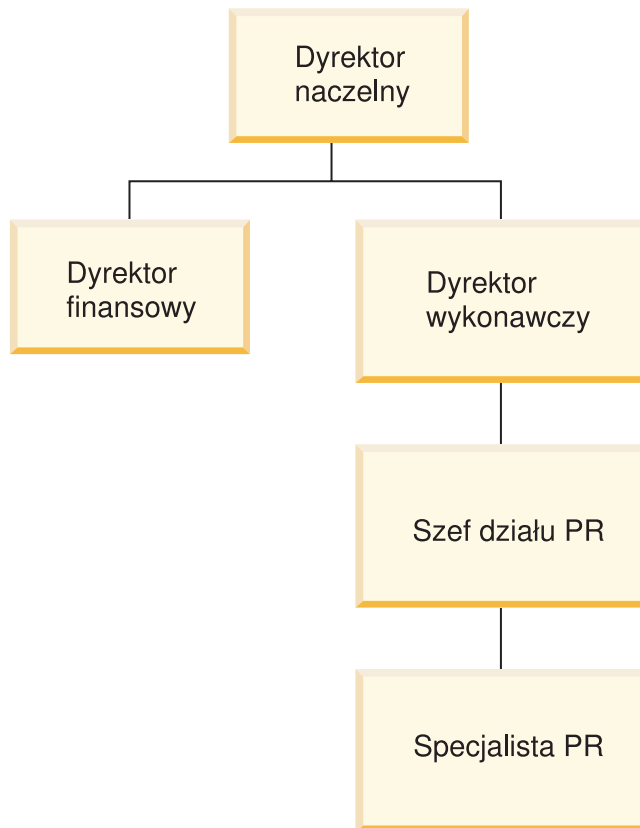
Hierarchia o wyraźnie określonych poziomach i rozgałęzieniach o spójnej głębokości. Logiczny element nadrzędny każdego poziomu znajduje się dokładnie o jeden poziom wyżej od niego. Hierarchia zrównoważona może reprezentować czas, gdy poszczególne poziomy mają spójną głębokość i znaczenie, na przykład Rok, Kwartał i Miesiąc. Poziomy są spójne, ponieważ każdy poziom reprezentuje ten sam typ informacji i każdy poziom jest logicznie równoważny. Rys. 8 pokazuje przykład zrównoważonej hierarchii czasu.



Rysunek 8. Hierarchia zrównoważona. Przykład hierarchii zrównoważonej

### Niezrównoważona

Hierarchia z poziomami wykazującymi spójny związek typu nadrzędny - podrzędny, lecz niespójnymi pod względem logicznym. Gałęzie takiej hierarchii mogą mieć rozbieżne głębokości. Niezrównoważona hierarchia może reprezentować schemat organizacyjny. Na przykład Rys. 9 na stronie 24 przedstawia układ z dyrektorem naczelnym na najwyższym poziomie hierarchii i co najmniej dwiema osobami, które mogą się znaleźć w rozgałęzieniach poniżej, z dyrektorem wykonawczym i dyrektorem finansowym włącznie. Oprócz tego rozgałęzienie od poziomu dyrektora wykonawczego obejmuje kolejne osoby, a rozgałęzienie dyrektora finansowego ich nie obejmuje. Związki typu nadrzędny - podrzędny w obu gałęziach hierarchii są spójne. Niemniej poziomy w obu rozgałęzieniach nie są logicznymi równoważnikami. Na przykład dyrektor finansowy nie jest logicznym równoważnikiem dyrektora wykonawczego.

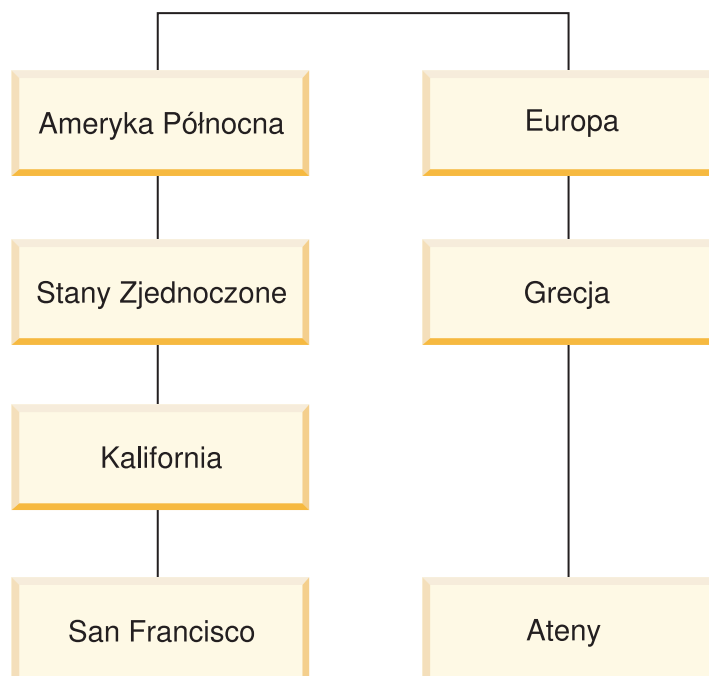


Rysunek 9. Hierarchia nierównoważona. Przykład hierarchii nierównoważonej

### Nierówna

Hierarchia, w której każdy poziom ma spójne znaczenie, lecz rozgałęzienia mają niespójne głębokości, ponieważ co najmniej jeden atrybut składowy na pewnym poziomie gałęzi jest niezapełniony. Hierarchia nierówna może reprezentować hierarchię geograficzną, w której znaczenie każdego poziomu, takiego jak miasto czy kraj, jest stosowane spójnie, lecz głębokość hierarchii jest zmienna. Rys. 10 na stronie 25 przedstawia hierarchię geograficzną, w której zdefiniowano poziomy Kontynent, Kraj, Prowincja/Stan i Miasto. W jednej gałęzi jest Ameryka Północna jako Kontynent, Stany Zjednoczone jako Kraj, Kalifornia jako Prowincja/Stan i San Francisco jako Miasto. Jednak hierarchia staje się nierówna, gdy w jednym elemencie brakuje wpisu na niektórych poziomach. Przykładowa gałąź zawiera Europę (Kontynent), Grecję (Kraj) i Ateny (Miasto), ale nie zawiera wpisu na poziomie Prowincja/Stan, ponieważ poziom ten nie ma zastosowania do Grecji w prezentowanym tu modelu działalności. W tym przykładzie gałęzie Grecji i Stanów Zjednoczonych schodzą na różne głębokości, tworząc hierarchię nierówną.

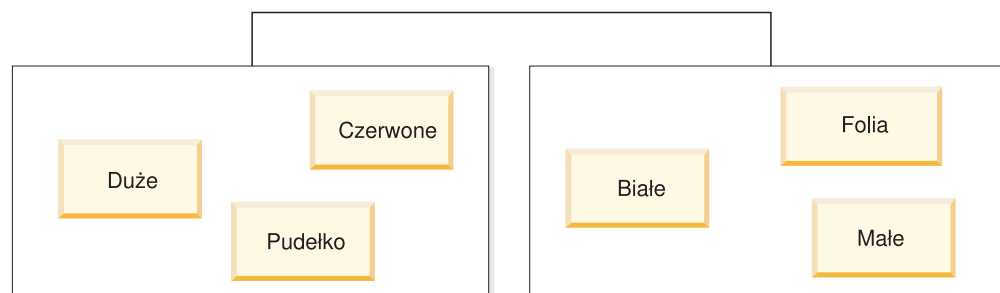




Rysunek 10. Hierarchia nierówna. Przykład hierarchii nierównej

### Sięciowa

Hierarchia, w której porządek poziomów nie jest określony, ale poziomy mają znaczenie semantyczne. Na przykład Rys. 11 przedstawia hierarchię sieciową opisującą atrybuty produktu, takie jak Kolor, Rozmiar i TypOpakowania. Ponieważ poziomy nie mają wewnętrznego związku typu nadrzędny-podrzędny, kolejność poziomów nie ma znaczenia. Firma może mieć takie wpisy elementów, jak biały dla atrybutu Kolor, mały dla atrybutu Rozmiar i folia kurczliwa dla atrybutu TypOpakowania. Drugi wpis elementu może zawierać czerwony jako Kolor, duży jako Rozmiar i pudełko jako TypOpakowania.



Rysunek 11. Hierarchia sieciowa. Przykład hierarchii sieciowej

Hierarchia określa także mechanizmy rozwinięć danej hierarchii. Mechanizm rozwinięcia definiuje sposób interpretowania danych w poziomach hierarchii. Obsługiwane są dwa poniższe mechanizmy rozwinięć:

### Standardowy

Korzysta ze związku definicji poziomów w hierarchii, w którym każdy poziom hierarchii jest używany jako jeden element rozwinięcia. Na przykład zrównoważona hierarchia dla wymiaru Czas byłaby zorganizowana według każdego ze zdefiniowanych poziomów obejmujących Rok, Kwartał i Miesiąc. Rozwinięcie standardowe może być stosowane we wszystkich czterech typach hierarchii. Tabela 6 na stronie 26

na stronie 26 pokazuje, jak niektóre atrybuty hierarchii zrównoważonej dla wymiaru Czas są zorganizowane przy użyciu rozwinięcia standardowego.

*Tabela 6. Rozwinięcie standardowe.* Rozwinięcie standardowe hierarchii zrównoważonej dla wymiaru Czas

Rok	Kwartał	Miesiąc
2003	Kw 1	Sty
2003	Kw 1	Lut
2003	Kw 1	Mar
2004	Kw 1	Sty
2004	Kw 1	Lut
2004	Kw 1	Mar

### Rekurencyjne

Oparte na wewnętrznym związku typu nadrzędny-podrzędny między poziomami hierarchii. Hierarchia niezrównoważona korzystająca z rozwinięcia rekurencyjnego jest reprezentowana przez pary poziomów nadrzędny-podrzędny. Na przykład Tabela 7 przedstawia pary poziomów dla hierarchii niezrównoważonej opisującej schemat organizacyjny pokazany na Rys. 9 na stronie 24. Pary poziomów nadrzędny-podrzędny obejmują stanowiska: dyrektora naczelnego i dyrektora finansowego, dyrektora naczelnego i dyrektora wykonawczego, dyrektora wykonawczego i szefa działu PR, szefa działu PR i specjalistę PR. Rozwinięcie rekurencyjne może być używane tylko w przypadku hierarchii niezrównoważonej.

*Tabela 7. Rozwinięcie rekurencyjne.* Rozwinięcie rekurencyjne hierarchii niezrównoważonej dla wymiaru Organizacja

Atrybut nadrzędny	Atrybut podrzędny
Dyrektor naczelnny	Dyrektor finansowy
Dyrektor naczelnny	Dyrektor wykonawczy
Dyrektor wykonawczy	Szef działu PR
Szef działu PR	Specjalista PR

W Centrum OLAP hierarchię można tworzyć przy użyciu Kreatora hierarchii. Hierarchię dla wymiaru można zdefiniować po utworzeniu wymiaru.

Poniższa tabela zawiera opis właściwości obiektu hierarchii.

*Tabela 8. Właściwości hierarchii*

Właściwość	Opis
Uporządkowany zestaw poziomów	Uporządkowana lista jednego lub kilku poziomów od góry do dołu hierarchii.
Typ	Hierarchia może być zrównoważona, niezrównoważona, nierówna lub sieciowa
Rozwinięcie	Rozwinięcie hierarchii może być standardowe lub rekurencyjne

---

## Poziomy

Poziom składa się z zestawu atrybutów działających wspólnie jako jeden logiczny krok w porządku hierarchii. W skład poziomu wchodzi jeden lub więcej atrybutów powiązanych ze sobą i mogących działać w obrębie poziomu na kilka sposobów. Związek między atrybutami w poziomie jest zwykle zdefiniowany za pomocą zależności funkcjonalnej.

Atrybut może być używany w poziomie na jeden z trzech sposobów:

### **Atrybuty klucza poziomu**

Jeden lub więcej atrybutów, których wartości jednoznacznie identyfikują wszystkie instancje poziomu. Na przykład dla poziomu Miasto najprostszym sposobem zdefiniowania atrybutu klucza poziomu jest użycie kolumny Identyfikator, na przykład Identyfikator miasta, w której wszystkie wartości muszą być unikalne. Atrybut Nazwa miasta nie może być atrybutem klucza poziomu, ponieważ nazwy miast mogą się powtarzać w różnych regionach i krajach. Jednak już połączenie nazwy kraju, nazwy stanu i nazwy miasta można zdefiniować jako atrybuty klucza poziomu, ponieważ zestaw tych trzech atrybutów definiuje miasto w sposób jednoznaczny. Należy zapewnić jednoznaczne definiowanie poziomu przez atrybuty klucza, ponieważ program DB2 Cube Views w oparciu o zdefiniowane poziomy tworzy zależności funkcjonalne używane podczas optymalizacji.

### **Atrybut domyślny**

Atrybut, którego wartość może być wyświetlana przez aplikację raportującą w postaci znaczących nazw wszystkich instancji poziomu w danych. Atrybut domyślny jest właściwością wymaganą i musi być funkcjonalnie określany przez atrybuty klucza poziomu. Na przykład jeśli atrybutem klucza poziomu jest Identyfikator miasta, wartości z tej kolumny nie będą czytelne w raportach. Dlatego jako atrybut domyślny, wyświetlany w raportach, można wyznaczyć nazwę miasta. Atrybutem domyślnym może być atrybut należący do zestawu atrybutów klucza poziomu. Na przykład jeśli kluczem poziomu Miasto jest zestaw nazwy kraju, nazwy stanu i nazwy miasta, domyślnym atrybutem może być także nazwa miasta.

### **Atrybuty pokrewne**

Zestaw dowolnej liczby atrybutów - zawierających dodatkowe informacje o instancjach poziomów - zdefiniowanych jako atrybuty klucza poziomu. Wszystkie atrybuty pokrewne muszą być funkcjonalnie określone przez atrybuty klucza poziomu. Na przykład dla poziomu Miasto atrybutami pokrewnymi mogą być burmistrz, populacja, położenie i opis.

Za pomocą Kreatora poziomów w Centrum OLAP można utworzyć nowy poziom - niezależnie lub w kontekście hierarchii. Poziomy mogą być współużytkowane przez wiele modeli kostek w wymiarach. Podczas tworzenia poziomu program DB2 Cube Views domyślnie tworzy zależność funkcjonalną wiążącą atrybuty w poziomie, tak aby atrybuty domyślne i pokrewne były funkcjonalnie zależne od atrybutów klucza poziomu. Doradca optymalizacyjny korzysta z zależności funkcjonalnych przy generowaniu zaleceń dotyczących tabel podsumowań, a optymalizator DB2 może na ich podstawie prawidłowo kierować zapytania SQL.

Modelowanie hierarchii za pomocą poziomów może się odbywać na wiele sposobów. Bez względu na dobór technik modelowania (idealne lub nieidealne), zawsze należy pamiętać o takim zdefiniowaniu każdego poziomu, aby atrybuty klucza poziomu funkcjonalnie określały atrybut domyślny i atrybuty pokrewne. Atrybuty klucza poziomu powinny w sposób unikalny identyfikować wartości tego poziomu. Należy zdefiniować tworzenie zależności funkcjonalnych między atrybutami każdego poziomu. Zależności funkcjonalne są wykorzystywane przez Doradcę optymalizacyjnego podczas ustalania zaleceń dotyczących

tabel podsumowań oraz przez optymalizator DB2 podczas kierowania zapytań SQL. Dzięki istnieniu zależności funkcjonalnych Doradca optymalizacyjny może wygenerować zalecenia dotyczące mniejszych tabel podsumowań, zapewniając tym samym szybsze przekazywanie wyników zapytań.

### Modelowanie idealne

W idealnym przypadku dane relacyjne wszystkich wymiarów są przechowywane w pojedynczej tabeli wymiarów, zawierającej kolumny identyfikatorów wszystkich poziomów w wymiarze. Każda kolumna identyfikatora jednoznacznie określa wartości w poziomie. Przykładem może być pojedyncza tabela wymiarów Region z następującymi kolumnami:

*Tabela 9. Idealne modelowanie tabeli wymiarów*

Kolumny w idealnej tabeli wymiarów Region
Identyfikator miasta (klucz podstawowy)
Nazwa miasta
Burmistrz
Identyfikator stanu
Nazwa stanu
Gubernator stanu
Identyfikator kraju
Nazwa kraju

Można utworzyć hierarchię zawierającą poziomy Kraj, Stan i Miasto. W każdym poziomie można zdefiniować zależność funkcjonalną między kolumną identyfikatora (atrybutem klucza poziomu) a atrybutami pokrewnymi, jak to przedstawia Tabela 10.

*Tabela 10. Idealne modelowanie hierarchii*

Poziom	Atrybut klucza poziomu	Atrybuty pokrewne poziomu
Kraj	Identyfikator kraju	Nazwa kraju
Stan	Identyfikator stanu	Nazwa stanu Gubernator stanu
Miasto	Identyfikator miasta	Nazwa miasta Burmistrz

Zależności funkcjonalne są tworzone między następującymi parami atrybutów:

- Identyfikator kraju → Nazwa kraju
- Identyfikator stanu → Nazwa stanu, Gubernator stanu
- Identyfikator miasta → Nazwa miasta, Burmistrz

Program DB2 Cube Views nie tworzy zależności funkcjonalnej dla atrybutu Identyfikator miasta i atrybutów pokrewnych, ponieważ Identyfikator miasta jest kluczem podstawowym i powinien już mieć zdefiniowane ograniczenie.

### Modelowanie nieidealne

Jeśli nie istnieją kolumny z unikalnymi identyfikatorami dla każdego poziomu w hierarchii, należy zachować większą ostrożność przy definiowaniu atrybutów klucza poziomu, aby uzyskać poprawną zależność funkcjonalną. Przykładem może być pojedyncza tabela wymiarów Region z następującymi kolumnami:

Tabela 11. Nieidealne modelowanie tabeli wymiarów

Kolumny w nieidealnej tabeli wymiarów Region
Identyfikator miasta (klucz podstawowy)
Nazwa miasta
Burmistrz
Nazwa stanu
Gubernator stanu
Nazwa kraju

Można utworzyć hierarchię zawierającą poziomy Kraj, Stan i Miasto, podobnie jak przykładzie z modelowaniem idealnym. Jednak atrybuty klucza poziomu muszą być zdefiniowane w sposób bardziej przemyślany, aby umożliwić jednoznaczną identyfikację każdego wiersza poziomu. Na przykład Nazwa miasta nie może być jednoznacznym identyfikatorem dla poziomu Miasto, ponieważ miasto Leicester znajduje się zarówno w Stanach Zjednoczonych, jak i w Wielkiej Brytanii. Jedynym sposobem jednoznacznego zdefiniowania poziomu Miast jest dobranie kombinacji atrybutów nazwy kraju, nazwy stanu i nazwy miasta, jak to przedstawia Tabela 12.

Tabela 12. Nieidealne modelowanie hierarchii

Poziom	Atrybuty klucza poziomu	Atrybuty pokrewne poziomu
Kraj	Nazwa kraju	
Stan	Nazwa kraju Nazwa stanu	Gubernator stanu
Miasto	Nazwa kraju Nazwa stanu Nazwa miasta	Burmistrz

Zależności funkcjonalne są tworzone między następującymi dwiema kombinacjami atrybutów:

- Nazwa kraju, Nazwa stanu i Nazwa miasta → Burmistrz
- Nazwa kraju i Nazwa stanu → Gubernator stanu

Poniższa tabela zawiera opis właściwości obiektu poziomu.

Tabela 13. Właściwości poziomu

Właściwość	Opis
Zestaw atrybutów klucza poziomu	Zestaw jednego lub większej liczby atrybutów, które łącznie stanowią jednoznaczną definicję poziomu.
Jeden atrybut domyślny	Jeden wymagany atrybut domyślny, funkcjonalnie określany przez atrybuty klucza poziomu i używany przez aplikacje raportujące do wyświetlania danych w czytelnej postaci.
Zestaw atrybutów pokrewnych	Zestaw dowolnej liczby atrybutów zawierających dodatkowe informacje na temat poziomu i określanych funkcjonalnie przez atrybuty klucza poziomu.
Zależność funkcjonalna	Wartość boolowska (Tak lub Nie) informująca, czy poziom ma odpowiednią zależność funkcjonalną w programie DB2 UDB.

Tabela 13. Właściwości poziomu (kontynuacja)

Właściwość	Opis
Nazwa zależności funkcjonalnej	Jeśli opcja zależności funkcjonalnej DB2 UDB ma wartość Tak, ta właściwość zawiera jej nazwę o maksymalnej długości 18 bajtów. Jeśli opcja zależności funkcjonalnej ma wartość Nie, ta właściwość jest ignorowana.

## Miary

Miary definiują jednostkę miary i są używane w obiektach faktów.

Miary mają sens w kontekście zestawu wymiarów w modelu kostki. Na przykład dochód o wartości 300 sam w sobie nie ma znaczenia. Dopiero po umieszczeniu miary przychodu w kontekście wymiarów, takich jak Region i Czas, miara nabiera znaczenia: przychód w styczniu w Nowym Jorku wyniósł 300. Przykładami często stosowanych miar są Dochód, Koszt i Zysk.

Miarę definiuje kombinacja dwóch właściwości: listy wyrażenia SQL i listy agregacji. Kolumny tabeli, atrybuty i miary są odwzorowywane na szablon w celu budowania wyrażen SQL. Wynikowe wyrażenia SQL zostają następnie użyte jako dane wejściowe dla pierwszej funkcji agregującej danej miary. Jeśli miara ma więcej niż jedną agregację, funkcje agregujące są wykonywane w kolejności ich występowania na liście, przy czym każda kolejna agregacja jako swoje dane wejściowe przyjmuje wynik poprzedniej agregacji. Jeśli wyrażenie SQL miary tylko odwołuje się do innych miar, funkcja agregująca jest opcjonalna. Funkcja agregująca jest opcjonalna, ponieważ miary będące obiektem odwołań same zapewniają agregację.

Wyrażenie SQL miary jest tworzone jako kombinacja dwóch właściwości: szablonu oraz listy kolumn, atrybutów i miar. W szablonie jest stosowana notacja leksemowa, w której `{$$n}` jest leksemem, gdzie `n` stanowi odwołanie do określonej kolumny, określonego atrybutu lub określonej miary z listy. Lista kolumn, atrybutów i miar jest uporządkowana, a pozycja kolumny, atrybutu lub miary na liście odpowiada wartości `n` leksemu.

Wyrażenia SQL służą za dane wejściowe pierwszej agregacji. Każda agregacja określa funkcję, która zostaje zastosowana do odpowiedniej listy wymiarów. Funkcją agregującą może być dowolna funkcja agregująca obsługiwana przez podstawową bazę danych. W programie DB2 Cube Views obsługiwane są następujące funkcje agregujące:

- AVG
- CORRELATION
- COUNT
- COUNT\_BIG
- COVARIANCE
- MAX
- MIN
- REGRESSION (wszystkie 9 typów funkcji)
- STTDEV
- SUM
- VARIANCE

Każdy wymiar może być tylko raz zagregowany przez obiekt miary. Jedna miara musi mieć jedną agregację z pustą listą wymiarów, a każda inna agregacja musi mieć jawną listę

wymiarów. Agregacja przeznaczona dla pustej listy wymiarów jest stosowana do wszystkich wymiarów w modelu kostki, które nie są jawnie używane przez inną agregację.

Przykładem prostej miary bezpośrednio odwzorowywanej na kolumnę jest Dochód. Miara Dochód może być utworzona dla modelu kostki z trzema wymiarami: Produkt, Rynek i Czas. Do dochodu jest przypisany szablon wyrażenia SQL `template = "{$$1}"`, reprezentujący proste odwzorowanie kolumny określonej na jednoelementowej liście kolumn, atrybutów i miar, gdzie `list = "Column Fakt.Dochód"`. Lista agregacji to `(SUM, <NULL>)`, gdzie `SUM` jest funkcją agregującą, a `<NULL>` jest pustą listą wymiarów. Wyrażenie SQL stanowi dane wejściowe funkcji agregującej `SUM`, co w wyniku daje wyrażenie SQL: `SUM(Fakt.Dochód)`.

Bardziej skomplikowana miara, Zysk, może mieć szablon wyrażenia SQL postaci `template = "{$$1} - {$$2}"`, gdzie lista atrybutów, kolumn i miar to `list = "Measure Dochód, Column Fakt.Koszt"`. Po zastąpieniu leksemów odpowiednimi odwołaniami wyrażenie SQL przyjmie postać "Dochód - Fakt.Koszt". Poszerzając odwołanie do miary dochodu o jego odwołanie do kolumny, otrzymamy wyrażenie SQL w postaci: "Fakt.Dochód - Fakt.Koszt". Lista agregacji dla miary Zysk wygląda następująco: `(SUM, <NULL>)`. Wyrażenie SQL do obliczania zysku jest argumentem funkcji agregującej `SUM`. Otrzymujemy kod SQL miary Zysk w następującej postaci: `SUM(Fakt.Dochód - Fakt.Koszt)`.

Jeśli miara zawiera funkcję agregującą, na przykład `CORRELATION`, wymagającą co najmniej dwóch parametrów, miara ta będzie mieć co najmniej dwa wyrażenia SQL.

Miary mają także typ danych oparty na typach danych SQL. Program DB2 Cube Views automatycznie ustala typ danych miary. Każda nazwa, gdy jest w pełni kwalifikowana przez schemat, musi być unikalna wśród miar i atrybutów.

Centrum OLAP w znacznym stopniu przesłania złożoność definicji obiektu metadanych. W Centrum OLAP nie ma potrzeby jawnego definiowania dla miar listy wyrażeń SQL lub listy agregacji. Aby utworzyć miarę bezpośrednio odwzorowaną na kolumnę, atrybut lub inną miarę, podczas tworzenia miary w Kreatorze faktów lub w oknie Właściwości faktów należy wybrać źródło. Aby utworzyć miarę obliczaną, do utworzenia wyrażenia źródłowego można użyć okna Kreatora wyrażeń SQL. Kreator wyrażeń SQL zawiera listę dostępnych kolumn, atrybutów, miar, operatorów, funkcji oraz stałych. W oknie Właściwości miary można wyświetlić typ danych źródłowych dla miary oraz typ danych miary po zagregowaniu danych źródłowych.

Opis specyficznych właściwości definiujących miarę przedstawiono w poniższej tabeli. Centrum OLAP definiuje każdą z nich automatycznie podczas tworzenia miary przez użytkownika.

*Tabela 14. Właściwości miary*

<b>Właściwość</b>	<b>Opis</b>
Lista wyrażeń SQL (szablon, [(lista kolumn, atrybutów i miar)])	Lista wyrażeń SQL służących jako dane wejściowe dla pierwszej funkcji agregującej miary. Każde z wyrażeń SQL składa się z szablonu i uporządkowanej listy kolumn, atrybutów i miar.
Lista agregacji (funkcja, lista wymiarów)	Lista agregacji określająca sposób obliczania miary. Każda z agregacji składa się z funkcji agregującej SQL oraz opcjonalnej listy wymiarów, do których funkcja ma być zastosowana.

Tabela 14. Właściwości miary (kontynuacja)

Właściwość	Opis
Typ danych (schemat, nazwa, długość, skala, dopuszczalna wartość pusta)	Określa typ danych miary. Typ danych jest oparty na typach danych SQL i składa się na niego schemat typu danych, jego nazwa, długość, skala oraz dopuszczalność wartości pustej. Centrum OLAP wyświetla schemat tylko wtedy, gdy jest to schemat inny niż SYSIBM.

## Atrybuty

Atrybut jest podstawowym pojęciem abstrakcyjnym używanym podczas opisywania kolumny tabeli w bazie danych. Atrybut zawiera wyrażenie SQL, które może być albo prostym odwzorowaniem na kolumnę tabeli albo wyrażeniem bardziej złożonym. Wyrażenia złożone mogą odwoływać się do wielu kolumn lub atrybutów i zawierać wszystkie funkcje SQL, także te zdefiniowane przez użytkownika.

Złożoność definiowania obiektu atrybutu jest w dużym stopniu maskowana przez Centrum OLAP dla DB2 Cube Views. W Centrum OLAP nie trzeba jawnie definiować szablonu wyrażenia ani listy parametrów atrybutu. Aby utworzyć atrybut bezpośrednio odwzorowany na kolumnę, podczas tworzenia atrybutu w Kreatorze wymiarów lub w oknie Właściwości wymiaru należy wybrać kolumnę źródłową. Aby utworzyć atrybut obliczany, w celu utworzenia wyrażenia źródłowego można użyć okna Kreatora wyrażeń SQL. Kreator wyrażeń SQL udostępnia listy atrybutów, kolumn, operatorów, funkcji i stałych.

Aby utworzyć atrybut nie korzystając z Centrum OLAP, należy utworzyć definicję wyrażenia SQL atrybutu, która jest kombinacją dwóch właściwości: szablonu oraz listy kolumn i atrybutów. W szablonie jest stosowana notacja przy użyciu leksemów, w której `$$$n` jest leksemem, a `n` oznacza określoną kolumnę lub atrybut z listy. Lista kolumn i atrybutów jest uporządkowana, zaś pozycja kolumny lub atrybutu na liście odpowiada wartości `n` leksemu. Na przykład szablonu `template = "{{$1}} || ' ' || {{$2}}` można użyć z odpowiadającą mu listą, taką jak `list = "Column CUSTOMER.FIRSTNAME, Attribute Nazwisko"` w celu połączenia imion z nazwiskami klientów i wstawienia między nimi spacji. Po zastąpieniu leksemów szablonu odpowiednimi odwołaniami do listy wyrażenie SQL ma postać `"Klient.Imię || ' ' || Nazwisko"`. Odwołanie do obiektu zostaje dalej rozwinięte na odwołanie do kolumny, tak że powstaje wyrażenie SQL: `"Klient.Imię || ' ' || Klient.Nazwisko"`.

Jeśli w wyrażeniu SQL definiującym atrybut stosowane są inne atrybuty, nie są dozwolone zapętłone odwołania między atrybutami. Na przykład: jeśli atrybut A odwołuje się do atrybutu B, to atrybut B nie może się odwoływać do atrybutu A.

Każda nazwa kwalifikowana nazwą schematu musi być unikalna w stosunku do nazw wszystkich innych atrybutów i miar w bazie danych.

Opis specyficznych właściwości definiujących atrybut przedstawia poniższa tabela. Centrum OLAP definiuje każdą z nich podczas tworzenia obiektu przez użytkownika.

Tabela 15. Właściwości atrybutu

Właściwość	Opis
Szablon wyrażenia SQL	Wyrażenie SQL definiujące atrybut. W szablonie odwołania do kolumn i atrybutów zapisywane są w notacji <code>\$\$\$n</code> , gdzie <code>n</code> jest liczbą porządkową odpowiadającą pozycji na liście kolumn i atrybutów.



Tabela 15. Właściwości atrybutu (kontynuacja)

Właściwość	Opis
Lista kolumn i atrybutów dla wyrażenia SQL	Uporządkowana lista wszystkich kolumn i atrybutów składających się na atrybut. Te kolumny i atrybuty stosuje się tak jak to określono w szablonie wyrażenia SQL.
Typ danych (schemat, nazwa, długość, skala, dopuszczalna wartość pusta)	Określa typ danych atrybutu. Typ danych jest oparty na typach danych SQL i składa się na niego schemat typu danych, jego nazwa, długość, skala oraz dopuszczalność wartości pustej. Centrum OLAP wyświetla schemat tylko wtedy, gdy jest to schemat inny niż SYSIBM.

## Związki atrybutów

Związek atrybutów opisuje ogólne związki między atrybutami, jednak nie jest elementem modelu kostki.

Związki składają się z następujących właściwości:

- atrybut lewostronny i prawostronny
- typ
- licznosc
- możliwa zależność funkcjonalna

Typ opisuje, jaka jest rola atrybutu prawostronnego względem atrybutu lewostronnego. Możliwe są dwa typy: Opisowy i Skojarzony.

### Opisowy

Określa, że atrybut prawostronny jest deskrytorem atrybutu lewostronnego. Na przykład prawostronny atrybut NazwaProduktu opisuje lewostronny atrybut KodProduktu.

### Skojarzony

Określa, że atrybut prawostronny jest skojarzony z atrybutem lewostronnym, lecz nie jest deskrytorem atrybutu lewostronnego. Na przykład prawostronny atrybut PopulacjaMiasta jest skojarzony z atrybutem IdMiasta, lecz nie jest jego deskrytorem.

Licznosc opisuje, jak są powiązane instancje atrybutów lewostronnych i prawostronnych. Do związków atrybutów można stosować następujące licznosci:

**1:1** Dla każdej instancji atrybutu prawostronnego istnieje nie więcej niż jedna instancja atrybutu lewostronnego i dla każdej instancji atrybutu lewostronnego istnieje nie więcej niż jedna instancja atrybutu prawostronnego.

### 1:wielu

Dla każdej instancji atrybutu prawostronnego istnieje nie więcej niż jedna instancja atrybutu lewostronnego, a dla każdej instancji atrybutu lewostronnego istnieje dowolna liczba instancji atrybutu prawostronnego.

### Wiele:1

Dla każdej instancji atrybutu prawostronnego istnieje dowolna liczba instancji atrybutu lewostronnego, a dla każdej instancji atrybutu lewostronnego istnieje nie więcej niż jedna instancja atrybutu prawostronnego.

### Wiele:wielu

Dla każdej instancji atrybutu prawostronnego istnieje dowolna liczba instancji

atrybutu lewostronnego i dla każdej instancji atrybutu lewostronnego istnieje dowolna liczba instancji atrybutu prawostronnego.

Właściwość zależności funkcjonalnych informuje, czy związek atrybutów definiuje związek funkcjonalny między dwoma atrybutami. Deklaracja, że związek atrybutów jest zależnością funkcjonalną, oznacza gwarancję, że każda instancja atrybutu lewostronnego będzie wyznaczać instancję atrybutu prawostronnego. Program DB2 Cube Views nie tworzy zależności funkcjonalnej między atrybutami opisanymi przez związek atrybutów, bez względu na ustawienie odpowiedniej właściwości.

Można zdefiniować kilka związków atrybutów wskazujących na istnienie zależności funkcjonalnej między identyfikatorem kraju a krajem, identyfikatorem stanu i stanem, identyfikatorem miasta i miastem, a także identyfikatorem miasta a jego populacją.

Obiekt związku atrybutów można utworzyć w sposób jawny w Centrum OLAP. Związek atrybutów można utworzyć jawnie za pomocą Kreatora związków atrybutów. Otwórz Kreatora związków atrybutów z widoku obiektów relacyjnych i określ wszystkie właściwości definicji obiektu.

Opis specyficznych właściwości definiujących obiekt związku atrybutów przedstawia poniższa tabela.

*Tabela 16. Właściwości związku atrybutów*

Właściwość	Opis
Atrybut lewostronny	Atrybut lewostronny używany w związku.
Atrybut prawostronny	Atrybut prawostronny używany w związku.
Typ	Typ związku opisywany przez związek atrybutów. Typ służy do określania roli atrybutu: Opisowy, Skojarzony.
Liczność	Liczność oczekiwana w łączeniu: 1:1, 1:wielu, Wiele:1, Wiele:wielu.
Zależność funkcjonalna	Określa, czy związek atrybutów jest zarazem zależnością funkcjonalną: Yes (tak), No (nie). <b>Uwaga:</b> Program DB2 Cube Views nie tworzy zależności funkcjonalnej w związku atrybutów bez względu na ustawienie tej właściwości na Tak lub Nie.

## Łączenia

Łączenie jest to obiekt metadanych opisujący łącznie między dwiema tabelami relacyjnymi. Łączenie odwołuje się do atrybutów odwołujących się do kolumn w łączonych tabelach.

Najprostsze łącznie odwołuje się do dwóch atrybutów: jednego stanowiącego odwzorowanie kolumny w pierwszej tabeli i jednego stanowiącego odwzorowanie kolumny w drugiej tabeli. Ponadto określa się operator w celu wskazania sposobu porównywania kolumn.

Łączenia można również użyć do modelowania łączy złożonych, w których co najmniej dwie kolumny z pierwszej tabeli są połączone z taką samą liczbą kolumn w drugiej tabeli. W łączeniu złożonym odpowiadające sobie kolumny są kojarzone za pomocą par atrybutów. Każda para atrybutów ma operator wskazujący sposób porównywania kolumn.

Łączenie ma także określony typ i licznosc. Typy łączy mają odwzorowanie w relacyjnych typach łączy. Łączy można używać w wymiarach - do łączy tabel wymiarów, w modelu

kostki - do łączenia wymiarów modelu kostki z jego obiektem faktów lub w obiekcie faktów - do łączenia wielu tabel faktów. Do tworzenia łączeń w Centrum OLAP można używać Kreatora łączeń.

Opis specyficznych właściwości definiujących łączenie przedstawiono w poniższej tabeli.

*Tabela 17. Właściwości łączenia*

<b>Właściwość</b>	<b>Opis</b>
Lista (atrybut lewostronny, atrybut prawostronny, operator)	Atrybut lewostronny: atrybut po lewej stronie łączenia. Atrybut prawostronny: atrybut po prawej stronie łączenia. Operator: operator oczekiwany w łączeniu: =, <, >, <>, >=, <=.
Typ	Oczekiwany typ łączenia: wewnętrzne, pełne, zewnętrzne, lewostronne zewnętrzne, prawostronne zewnętrzne.
Liczność	Liczność oczekiwana w łączeniu: 1:1, 1:wielu, Wiele:1, Wiele:wielu.

## Kostki

Kostka jest dokładną definicją kostki OLAP, którą czasem można sformułować za pomocą pojedynczej instrukcji SQL. Kostka, wyprowadzona z modelu kostki, zawiera podzbiór obiektów metadanych, które są oparte na obiektach metadanych w modelu kostki.

Obiekt faktów kostki oraz lista wymiarów kostki to podzbiory tych elementów zawartych w modelu kostki, do którego dana kostka się odwołuje. Kostki są strukturą odpowiednią dla narzędzi i aplikacji niekorzystających z wielu hierarchii, ponieważ wymiary kostki dopuszczają istnienie tylko jednej hierarchii przypadającej na wymiar.

Kostki mogą być używane przy optymalizowaniu modelu kostki w celu określenia tych jego regionów, które są najbardziej aktywne i najważniejsze. Wyznacza się przy tym wycinki optymalizacji, czyli regiony kostki będące najczęściej celem zapytań.

Do tworzenia kostek można używać Kreatora kostek w Centrum OLAP. Aby można było utworzyć kostkę skojarzoną, musi istnieć kompletny model kostki. Poniższa tabela zawiera opis właściwości kostki.

*Tabela 18. Właściwości kostki*

<b>Właściwość</b>	<b>Opis</b>
Model kostki	Model kostki, od którego pochodzi dana kostka.
Obiekt faktów kostki	Obiekt faktów używany w kostce. Obiekt faktów kostki jest pochodną obiektu faktów w modelu kostki.
Lista wymiarów kostki	Uporządkowana lista wymiarów kostki używanych w kostce. Każdy wymiar kostki jest pochodną odpowiadającego mu wymiaru w modelu kostki. Z każdym wymiarem kostki jest skojarzona jedna hierarchia kostki.

Tabela 18. Właściwości kostki (kontynuacja)

Właściwość	Opis
Lista wycinków optymalizacji	<p>Zestaw dowolnej liczby wycinków optymalizacji. Definicja każdego wycinka optymalizacji obejmuje:</p> <ul style="list-style-type: none"> <li>• Typ: z drażeniem, raport, z wyodrębnianiem MOLAP, z wyodrębnianiem hybrydowym lub z rozszerzaniem.</li> <li>• Zestaw poziomów optymalizacji, po jednym poziomie na każdy wymiar kostki. Poziom optymalizacji odwołuje się do: <ul style="list-style-type: none"> <li>– wymiaru kostki i odpowiadającej mu hierarchii kostki</li> <li>– do wszystkich lub do dowolnego poziomu kostki</li> </ul> </li> </ul>

## Obiekt faktów kostki

Obiekt faktów kostki ma podzbiór miar w formie uporządkowanej listy z określonego obiektu faktów. Obiekt faktów kostki zapewnia kostce elastyczność w doborze zakresów z obiektu faktów modelu kostki. Odwołania do informacji strukturalnych, takich jak łączenia i atrybuty, zawarte są w nadrzędnym obiekcie faktów.

W Centrum OLAP kostkę tworzy się w kontekście modelu kostki, używając Kreatora kostek. Nie ma potrzeby jawnego definiowania obiektu faktów kostki, ponieważ Centrum OLAP automatycznie traktuje obiekt faktów kostki jako wyodrębniony z obiektu faktów w skojarzonym modelu kostki. Użytkownik wybiera, które miary z obiektu faktów modelu kostki mają być użyte w kostce.

Opis właściwości definiujących obiekt faktów kostki przedstawiono w poniższej tabeli.

Tabela 19. Właściwości obiektu faktów kostki

Właściwość	Opis
Obiekt faktów	Obiekt faktów, z którego obiekt faktów kostki został wyodrębniony.
Lista miar	Uporządkowana lista miar używanych w kostce. Wszystkie miary muszą wchodzić w skład obiektu faktów, z którego został wyodrębniony obiekt faktów kostki.

## Wymiary kostki

Obiekt wymiaru kostki służy do określania zakresu wymiaru, jakiego można użyć w kostce. Obiekt wymiaru kostki odwołuje się do wymiaru, którego jest pochodną, oraz do hierarchii kostki właściwej dla danej kostki.

Do wymiaru kostki można zastosować tylko jedną hierarchię kostki. Odwołania do łączeń i atrybutów dotyczących danego wymiaru kostki zawiera definicja wymiaru.

W Centrum OLAP kostkę tworzy się w kontekście modelu kostki, używając Kreatora kostek. Użytkownik wybiera wymiary modelu kostki, które mają się znaleźć w kostce. Dla każdego wymiaru kostki można wybrać poziomy, które mają być uwzględnione w hierarchii kostki.

Opis specyficznych właściwości definiujących obiekt wymiaru kostki przedstawiono w poniższej tabeli.

Tabela 20. Właściwości wymiaru kostki

Właściwość	Opis
Wymiar	Wymiar, od którego pochodzi wymiar kostki.
Hierarchia kostki	Hierarchia kostki właściwa dla wymiaru kostki.

## Hierarchie kostki

Hierarchia kostki stanowi podzbiór hierarchii i jest używana w kostce. Hierarchia kostki odwołuje się do hierarchii, od której pochodzi (nadrzędnej), i może zawierać zestaw poziomów kostki będący podzbiorem poziomów hierarchii nadrzędnej.

Wymiar kostki może mieć tylko jedną hierarchię kostki. Na ogół hierarchia kostki ma ten sam typ hierarchii oraz mechanizm rozwinięcia co hierarchia, od której pochodzi. Jeśli hierarchia jest typu sieciowego, jest zrównoważona, gdy nie brakuje w niej żadnych elementów, i niezrównoważona, jeśli niektórych elementów brakuje.

W Centrum OLAP kostkę tworzy się w kontekście modelu kostki, używając Kreatora kostek. Użytkownik wybiera, który z wymiarów modelu kostki ma być uwzględniony w kostce. Dla każdego wymiaru uwzględnionego jako wymiar kostki można wybrać poziomy kostki, które mają być uwzględnione w hierarchii kostki.

Opis specyficznych właściwości definiujących hierarchię kostki przedstawiono w poniższej tabeli.

Tabela 21. Właściwości hierarchii kostki

Właściwość	Opis
Hierarchia	Hierarchia, z której jest wyodrębniona dana hierarchia kostki.
Uporządkowany zestaw poziomów kostki	Uporządkowany zestaw (jedno- lub wieloelementowy) poziomów kostki, będący podzbiorem poziomów zawartych w hierarchii nadrzędnej. Porządek poziomów kostki powinien być taki sam, jak w hierarchii nadrzędnej.

## Poziomy kostki

Poziom kostki stanowi podzbiór poziomu i jest używany w kostce. Poziom kostki odwołuje się do poziomu, z którego został wyodrębniony (nadrzędnego), i dziedziczy po nim atrybuty klucza oraz atrybut domyślny, zdefiniowane dla poziomu nadrzędnego.

Poziom kostki może zawierać zbiór atrybutów będących podzbiorem pokrewnych atrybutów z poziomem nadrzędnego.

Poniższa tabela zawiera opis właściwości poziomu kostki.

Tabela 22. Właściwości poziomu kostki

Właściwość	Opis
Zestaw atrybutów	Zestaw dowolnej liczby pokrewnych atrybutów, będący podzbiorem atrybutów pokrewnych poziomu nadrzędnego.

---

## Reguły obiektów metadanych

Istnieją trzy typy reguł dotyczących obiektów metadanych: reguły podstawowe, reguły kompletności modelu kostki i reguły optymalizacji. Reguły te służą do sprawdzania, czy każdy z obiektów jest poprawny zarówno w kontekście, jak i poza kontekstem modelu kostki, oraz czy możliwe jest napisanie i zoptymalizowanie efektywnych zapytań SQL.

### Reguły podstawowe

Reguły podstawowe definiują poprawność obiektu metadanych poza kontekstem jego użycia. Każdy obiekt metadanych ma własny zestaw reguł. Obiekt metadanych jest poprawny, jeśli spełnia wszystkie reguły podstawowe.

### Reguły kompletności

Reguły kompletności dotyczą tylko modeli kostki i stanowią rozszerzenie reguł podstawowych umożliwiające sprawdzenie, czy model kostki jest poprawnie połączony z innymi obiektami metadanych i czy możliwe jest napisanie efektywnych zapytań SQL.

### Reguły optymalizacji

Reguły optymalizacji stanowią rozszerzenie reguł podstawowych i reguł kompletności modelu kostki. Reguły te mają na celu zapewnienie możliwości pomyślnego zoptymalizowania zapytań SQL utworzonych dla obiektów metadanych.

### Reguły modeli kostki

Reguły podstawowe:

- W parach wymiar-łączenie musi istnieć zarówno wymiar, jak i łączenie.
- Wszystkie atrybuty po jednej stronie łączenia wymiaru z faktami muszą istnieć na liście atrybutów wymiaru, a wszystkie atrybuty po drugiej stronie łączenia muszą istnieć na liście atrybutów obiektu faktów.
- Model kostki musi się odwoływać do wszystkich jawnych wymiarów, do których odwołują się agregacje miar z obiektu faktów modelu kostki. Jeśli model kostki ma wymiary, agregacja z pustą listą wymiarów musi być zgodna z co najmniej jednym wymiarem z modelu kostki. Należy się upewnić, że do danego wymiaru nie odwołują się inne agregacje o tej samej mierze. Jeśli jednak model kostki nie ma wymiarów, wszystkie miary muszą mieć tylko agregacje z pustą listą wymiarów.

Reguły kompletności:

- Model kostki musi odwoływać się do jednego obiektu faktów.
- Model kostki musi odwoływać się do przynajmniej jednego wymiaru.

Reguły optymalizacji:

- Łączenie użyte do łączenia obiektu faktów i wymiaru musi mieć licznosc 1:1 lub Wiele:1 i musi łączyć tabelę faktów z podstawową tabelą wymiarów.
- Co najmniej jeden wymiar w modelu kostki musi mieć co najmniej jedną hierarchię.

### Reguły obiektu faktów

Reguły podstawowe:

- Obiekt faktów musi odwoływać się do przynajmniej jednej miary.
- Wszystkie atrybuty i miary, do których odwołuje się obiekt faktów, muszą dawać się połączyć w obrębie tego obiektu faktów. Pod uwagę brane są tylko łączenia obiektu faktów.

- Między dowolnymi dwiema tabelami w obiekcie faktów można zdefiniować dokładnie jedno łączenie.
- W obiekcie faktów nie są dozwolone łączenia zapętlone.
- Łączenia, do których odwołuje się obiekt faktów, muszą się odwoływać do atrybutów tego obiektu faktów.

### **Reguły wymiarów**

Reguły podstawowe:

- Wymiar musi odwoływać się do przynajmniej jednego atrybutu.
- Wszystkie atrybuty, do których odwołuje się wymiar, muszą dawać się połączyć w obrębie tego wymiaru. Pod uwagę brane są tylko łączenia wymiaru.
- W obrębie wymiaru nie są dozwolone łączenia zapętlone.
- Między dowolnymi dwiema tabelami w obrębie wymiaru może być zdefiniowane dokładnie jedno łączenie.
- Łączenia, do których odwołuje się wymiar, muszą się odwoływać do atrybutów tego wymiaru.
- Poziomy, do których odwołuje się wymiar, muszą się odwoływać do atrybutów danego wymiaru.
- Hierarchie, do których odwołuje się wymiar, muszą się odwoływać do poziomów wymiaru.

Reguły optymalizacji:

- Wymiar musi mieć jedną tabelę podstawową, z którą łączenia są wiązane z zastosowaniem liczności 1:1 lub wiele:1.

### **Reguły poziomów**

Reguły podstawowe:

- Poziom musi się odwoływać do co najmniej jednego atrybutu klucza poziomu.
- Poziom musi się odwoływać do co najmniej jednego atrybutu domyślnego.

### **Reguły hierarchii**

Reguły podstawowe:

- Hierarchia musi się odwoływać do co najmniej jednego poziomu.
- Muszą istnieć dokładnie dwa poziomy do celów rozwinięcia rekurencyjnego.
- Rozwinięcie standardowe może być stosowane w przypadku wszystkich rodzajów hierarchii, a rozwinięcie rekurencyjne tylko w przypadku hierarchii niezrównoważonych.

### **Reguły miar**

Reguły podstawowe:

- Każdy szablon wyrażenia SQL może mieć zero lub więcej parametrów następujących typów: atrybuty, kolumny i miary.
- Atrybuty i miary stosowane jako parametry w szablonie wyrażenia SQL nie mogą tworzyć pętli zależności.
- Szablon SQL miary nie może być łańcuchem pustym.
- Szablon SQL nie może korzystać z funkcji agregujących.
- Jeśli istnieją odwołania do przynajmniej jednej miary i jeśli obiektami odwołań są tylko miary, definiowanie agregacji jest opcjonalne.

- Liczba szablonów SQL musi być dopasowana do liczby parametrów pierwszej funkcji agregującej, jeśli agregacja istnieje.
- Miara z wieloma szablonami SQL musi definiować przynajmniej jeden krok w skrypcie agregacji.
- Jeśli miara odwołuje się do drugiej miary, która definiuje wiele szablonów SQL, to miara odwołująca nie może mieć skryptu agregacji.
- Wieloparametrowa funkcja agregująca może być używana tylko w pierwszej agregacji.
- Jeśli miara definiuje jedną lub więcej agregacji, jedna z tych agregacji musi określać pustą listę wymiarów.
- Miara może odwoływać się do każdego wymiaru tylko raz, zarówno w obrębie jednej agregacji, jak i w ramach wszystkich agregacji.
- W szablonie SQL numerowanie wskaźników leksemów musi rozpoczynać się od liczby 1 i musi być kontynuowane przy użyciu kolejnych numerów.
- W obrębie wyrażenia SQL każda kolumna, atrybut i miara musi być celem przynajmniej jednego odwołania.

### **Reguły atrybutów**

Reguły podstawowe:

- Każdy szablon SQL może mieć zero lub więcej parametrów typu atrybut lub kolumna.
- Atrybuty stosowane jako parametry w szablonie wyrażenia SQL nie mogą tworzyć pętli zależności.
- Szablon SQL atrybutu nie może być łańcuchem pustym.
- Szablon SQL nie może mieć funkcji agregujących.
- W szablonie SQL numerowanie wskaźników leksemów musi rozpoczynać się od liczby 1 i musi być kontynuowane przy użyciu kolejnych numerów.
- W obrębie wyrażenia SQL każda kolumna i atrybut musi być celem przynajmniej jednego odwołania.

### **Reguły związków atrybutów**

Reguły podstawowe:

- Związek atrybutów musi odwoływać się do dwóch atrybutów.
- Związek atrybutów nie może być zdefiniowany jako zależność funkcjonalna o licznosci wiele:wielu.

### **Reguły łączy**

Reguły podstawowe:

- Łączenie musi odwoływać się do przynajmniej jednego trypletu (atrybut lewostronny, atrybut prawostronny i operator).
- Dla każdego trypletu łączenia musi być zdefiniowana prawidłowa operacja. Typy danych atrybutów lewostronnych i prawostronnych muszą być zgodne ze sobą oraz z wybranym operatorem.
- Wszystkie atrybuty lewostronne muszą umożliwiać rozstrzygnięcie na jedną lub więcej kolumn pojedynczej tabeli.
- Wszystkie atrybuty prawostronne muszą umożliwiać rozstrzygnięcie na jedną lub więcej kolumn pojedynczej tabeli.

Reguły optymalizacji:



- Dla kolumn uczestniczących w łączeniu musi być zdefiniowane ograniczenie. Jeśli po obu stronach relacji równości używany jest ten sam zestaw kolumn, należy zdefiniować klucz podstawowy zgodny z tym zbiorem kolumn. Jeśli po obu stronach relacji równości używane są różne zestawy kolumn, jedna strona musi mieć pasujący klucz podstawowy, a druga strona musi mieć pasujący klucz obcy i odwoływać się do klucza podstawowego.
- Liczność łączenia musi wynosić 1:1, wiele:1 lub 1:wielu. W wypadku łączenia z tym samym zestawem kolumn po obu stronach relacji równości licznosc musi wynosić 1:1. W wypadku wszystkich innych łączeń licznosc musi mieć 1 po stronie ze zdefiniowanym kluczem podstawowym i N po stronie ze zdefiniowanym kluczem obcym. Jeśli strona z kluczem obcym także ma zdefiniowany klucz podstawowy, licznosc po tej stronie musi wynosić 1.
- Wszystkie atrybuty użyte w łączeniu muszą umożliwiać ich rozwiązanie na wyrażenia SQL, w których nie są dopuszczalne wartości puste.
- Typem łączenia musi być łączenie wewnętrzne.

### Reguły kostek

Reguły podstawowe:

- Kostka musi się odwoływać do jednego modelu kostki.
- Kostka musi odwoływać się do co najmniej jednego obiektu faktów kostki.
- Kostka musi odwoływać się do przynajmniej jednego wymiaru kostki.
- Obiekt faktów kostki musi pochodzić od obiektu faktów użytego w odnośnym modelu kostki.
- Wszystkie wymiary kostki muszą pochodzić od wymiarów używanych w odnośnym modelu kostki.
- Kostka musi mieć zero lub więcej wycinków optymalizacji.
- Wycinek optymalizacji musi mieć jedną warstwę optymalizacji na każdy wymiar kostki w kostce.
- Wycinek optymalizacji musi mieć co najmniej jeden poziom optymalizacji.
- Poziom optymalizacji musi mieć jedno odwołanie do wymiaru kostki i jedno odwołanie do hierarchii.
- Poziom optymalizacji musi mieć odwołanie typu **Wszystkie**, **Dowolny** lub odwołanie do poziomu kostki.
- W obiekcie poziomu optymalizacji odnośny wymiar kostki i odnośna hierarchia kostki musi pochodzić od obiektów w obiekcie kostki. Hierarchia kostki musi należeć do wymiaru kostki.
- W obiekcie poziomu optymalizacji: jeśli odwołanie do poziomu kostki nie jest określone jako **Wszystkie** lub **Dowolny**, to poziom kostki musi należeć do hierarchii kostki.
- Odwołanie do wymiaru kostki lub hierarchii kostki w poziomie optymalizacji nie powinno się powtarzać w ramach danego wycinka optymalizacji.
- Wycinek optymalizacji z wyodrębnianiem typu MOLAP nie może istnieć, jeśli istnieje inny obiekt wycinka optymalizacji z wyodrębnianiem hybrydowym, i na odwrót.
- Każda kostka ma maksymalnie jeden wycinek optymalizacji z typem wyodrębniania MOLAP.
- Każda kostka ma maksymalnie jeden wycinek optymalizacji z typem wyodrębniania hybrydowego.
- Wycinek optymalizacji z rozszerzaniem może istnieć, jeśli istnieje inny wycinek optymalizacji z typem wyodrębniania hybrydowego.

### **Reguły obiektów faktów kostki**

Reguły podstawowe:

- Obiekt faktów kostki musi odwoływać się do jednego obiektu faktów.
- Obiekt faktów kostki musi odwoływać się do przynajmniej jednej miary.
- Do wszystkich miar, do których odwołuje się obiekt faktów kostki, muszą również istnieć odwołania w odpowiednim obiekcie faktów.

### **Reguły wymiarów kostki**

Reguły podstawowe:

- Wymiar kostki musi odwoływać się do jednego wymiaru.
- Wymiar kostki musi odwoływać się do jednej hierarchii kostki.
- Odnośna hierarchia kostki musi pochodzić od hierarchii używanej w odnośnym wymiarze.

### **Reguły poziomów kostki**

Reguły podstawowe:

- Poziom kostki musi się odwoływać do jednego poziomu.
- Poziom kostki musi się odwoływać co najmniej do jednego atrybutu pokrewnego.
- Wszystkie odnośne atrybuty muszą być atrybutami pokrewnymi w danym poziomie.

### **Reguły hierarchii kostki**

Reguły podstawowe:

- Hierarchia kostki musi odwoływać się do jednej hierarchii.
- Hierarchia kostki musi się odwoływać co najmniej do jednego poziomu kostki.
- Poziomy kostki, do których odwołuje się hierarchia kostki, muszą pochodzić od poziomów będących częścią odpowiedniej hierarchii.
- Poziomy kostki w hierarchii kostki muszą być wymienione w takiej samej kolejności, jak w odpowiadających im poziomach w odnośnej hierarchii. Kolejność poziomów kostki nie dotyczy hierarchii sieciowych.

---

## Rozdział 3. Projektowanie modeli obiektów metadanych programu DB2 Cube Views

W tej sekcji omówiono następujące tematy:

### Uruchamianie Centrum OLAP

Centrum OLAP umożliwia zarządzanie obiektami metadanych.

### Centrum OLAP a kompatybilność z interfejsami API

Program DB2 Cube Views, wersja 8.2, ma ograniczoną obsługę kompatybilności wstecznej dla Centrum OLAP oraz interfejsu API metadanych.

### Uprawnienia potrzebne do korzystania z programu DB2 Cube Views

Aby móc wykonywać zadania w Centrum OLAP, należy się zwrócić do administratora bazy danych (DBA) po pomoc w uzyskaniu wymaganych uprawnień w danym systemie operacyjnym.

### Tworzenie obiektów metadanych programu DB2 Cube Views

Obiekty metadanych programu DB2 Cube Views można tworzyć przy użyciu Centrum OLAP.

### Usuwanie wymiaru z modelu kostki

Jeśli wymiar nie jest już potrzebny, można go usunąć z modelu kostki. Wymiar można usunąć logicznie nie usuwając go fizycznie, jeśli jest używany w innym modelu kostki.

### Usuwanie obiektu metadanych z bazy danych

Jeśli obiekt metadanych nie jest już używany w modelu kostki w danej bazie danych, to obiekt ten można usunąć.

---

## Uruchamianie i odświeżanie Centrum OLAP

Centrum OLAP umożliwia zarządzanie obiektami metadanych.

Aby uruchomić Centrum OLAP:

1. Kliknij opcje: **Start** → **Programy** → **IBM DB2** → **Inteligentne narzędzia analityczne** → **Centrum OLAP**. Zostanie otwarte okno połączenia z bazą danych.
2. W oknie połączenia z bazą danych zaloguj się w bazie danych, której metadanymi chcesz zarządzać.

Centrum OLAP pokaże obraz stanu obiektów metadanych w bazie danych z określonej chwili. Program DB2 Cube Views zawsze dba o integralność obiektów metadanych znajdujących się pod jego kontrolą, jednak zawartość okna Centrum OLAP nie jest automatycznie aktualizowana, gdy inny użytkownik Centrum OLAP lub inna aplikacja API tworzy obiekty w bazie danych. Jeśli inny użytkownik lub aplikacja API zmodyfikuje obiekty metadanych, można wybrać opcje: **Widok** → **Odśwież**, aby wyświetlić nowy stan bazy danych. W przypadku gdy na tych samych obiektach metadanych pracuje jednocześnie większa liczba użytkowników, mogą zdarzać się błędy wynikające z tego, że użytkownicy nie widzą u siebie aktualnej wersji danych. Na tym samym zestawie obiektów metadanych nie powinno pracować jednocześnie wielu użytkowników.

---

## Kompatybilność Centrum OLAP i wersji interfejsu API

Program DB2 Cube Views, wersja 8.2, ma ograniczoną obsługę kompatybilności wstecznej dla Centrum OLAP oraz interfejsu API metadanych.

Obsługa zapewniana przez interfejs API programu DB2 Cube Views:

- Interfejs API w wersji 8.2 obsługuje żądania Describe z klienta w wersji 8.1. Describe jest jedyną operacją wersji 8.1 obsługiwaną od poprzedniej wersji.

Obsługa zapewniana przez Centrum OLAP programu DB2 Cube Views:

- Centrum OLAP wersji 8.2 nie obsługuje interfejsu API w wersji 8.1 ani tabel metadanych w wersji 8.1.
- Centrum OLAP w wersji 8.1 nie obsługuje interfejsu API w wersji 8.2 ani tabel metadanych w wersji 8.2.
- Centrum OLAP w wersji 8.1 nie może nawiązać połączenia z bazą danych DB2 w wersji 8.2.
- Centrum OLAP obsługuje importowanie plików XML w wersji 8.1. Gdy importowany jest plik XML w wersji 8.1, Centrum OLAP przed zaimportowaniem metadanych dokonuje migracji danego pliku XML do wersji 8.2 przy użyciu operacji Translate.
- Centrum OLAP może eksportować pliki XML zarówno w wersji 8.2, jak i w wersji 8.1.

Obsługa zapewniana przez program db2mdapiclient programu DB2 Cube Views:

- Program db2mdapiclient w wersji 8.1 obsługuje interfejs API w wersji 8.2, jeśli używany jest kod XML w wersji 8.2.
- Program db2mdapiclient w wersji 8.2 obsługuje interfejs API w wersji 8.1, jeśli używany jest kod XML w wersji 8.1.

---

## Uprawnienia do korzystania z programu DB2 Cube Views

Aby móc wykonywać zadania w Centrum OLAP, należy się zwrócić do administratora bazy danych (DBA) po pomoc w uzyskaniu wymaganych uprawnień w danym systemie operacyjnym.

W poniższych tabelach wszystkie odwołania do uprawnień SELECT, INSERT, UPDATE i DELETE, o ile nie podano inaczej, odnoszą się do tabel w schemacie DB2INFO dla bazy danych, w której zainstalowano program DB2 Cube Views.

### Uprawnienia w systemie Windows

W systemach operacyjnych Windows uprawnienia do zadań Centrum OLAP odnoszą się do identyfikatora użytkownika użytego do nawiązania połączenia z bazą danych.

*Tabela 23. Ogólne uprawnienia w systemie Windows*

Zadanie	Uprawnienia
Wszystkie zadania Centrum OLAP	<ul style="list-style-type: none"><li>• SYSADM lub DBADM do bazy danych, w której zainstalowano program DB2 Cube Views</li><li>• EXECUTE do procedury zapisanej w bazie DB2INFO.MD_MESSAGE zawartej w bazie danych, w której zainstalowano program DB2 Cube Views (dotyczy tylko programu DB2 UDB, wersja 8)</li><li>• CONNECT do docelowych baz danych</li></ul>
Tworzenie obiektów metadanych	SELECT, INSERT

Tabela 23. Ogólne uprawnienia w systemie Windows (kontynuacja)

Zadanie	Uprawnienia
Modyfikowanie obiektów metadanych	SELECT, INSERT, UPDATE, DELETE
Usuwanie obiektów metadanych	SELECT, DELETE
Eksportowanie obiektów metadanych do pliku XML	SELECT
Importowanie obiektów metadanych do programu DB2 UDB z pliku XML	SELECT, INSERT, UPDATE
Uruchamianie Doradcy optymalizacyjnego	SELECT do tabel systemowych i tabel podstawowych
Uruchamianie komend DB2 generowanych przez Doradcę optymalizacyjnego	<ul style="list-style-type: none"> <li>• CREATEIN, DROPIN do schematu DB2INFO</li> <li>• SELECT i ALTER (lub CONTROL) do tabel podstawowych</li> </ul>

Niektóre zadania konfiguracyjne oraz instalacyjne można wykonać przy użyciu Centrum OLAP.

Tabela 24. Uprawnienia konfiguracyjne oraz instalacyjne w systemie Windows

Zadanie	Uprawnienia
Wszystkie zadania konfiguracyjne oraz instalacyjne	<ul style="list-style-type: none"> <li>• SYSADM do bazy danych, w której zainstalowano program DB2 Cube Views, do tworzenia schematów dla obiektów metadanych</li> <li>• CONNECT, CREATETAB, IMPLICIT_SCHEMA do bazy danych, w której ma być zainstalowany program DB2 Cube Views</li> <li>• CREATEIN, DROPIN do schematu DB2INFO zawartego w bazie danych, w której zainstalowano program DB2 Cube Views</li> <li>• SELECT, INSERT, UPDATE, DELETE, CONTROL do wszystkich tabel w schemacie DB2INFO dla bazy danych, w której zainstalowano program DB2 Cube Views</li> </ul>

### Uprawnienia w systemie AIX

W systemie AIX są dwa różne typy identyfikatorów użytkownika, przy czym każdy z nich ma odrębny zestaw uprawnień. Identyfikator użytkownika jednego typu należy ustanowić do nawiązywania połączenia z bazą danych i uruchamiania procedury zapisanej w bazie DB2INFO.MD\_MESSAGE. Ten typ identyfikatora użytkownika jest dalej określane mianem połączeniowego identyfikatora użytkownika.

Identyfikator użytkownika drugiego typu należy ustanowić do uruchamiania wszystkich pozostałych zadań Centrum OLAP. Ten typ identyfikatora użytkownika jest dalej określane mianem wykonawczego identyfikatora użytkownika. Ten identyfikator użytkownika jest specjalnym identyfikatorem użytkownika ustanawianym w systemie AIX w celu uruchamiania procedur zapisanych w bazie typu FENCED. Procedury zapisane w bazie typu FENCED działają pod kontrolą identyfikatora użytkownika, który jest właścicielem pliku .fenced w katalogu sqllib/adm. Ten identyfikator użytkownika służy do nadzorowania zasobów systemowych dostępnych dla procedury zapisanej w bazie.

Tabela 25. Ogólne uprawnienia w systemie AIX dla połączeniowego identyfikatora użytkownika

Zadanie	Uprawnienia
Wszystkie zadania Centrum OLAP	<ul style="list-style-type: none"> <li>• SYSADM lub DBADM do bazy danych, w której zainstalowano program DB2 Cube Views</li> <li>• EXECUTE do procedury zapisanej w bazie DB2INFO.MD_MESSAGE zawartej w bazie danych, w której zainstalowano program DB2 Cube Views (dotyczy tylko programu DB2 UDB, wersja 8)</li> <li>• CONNECT do docelowych baz danych</li> </ul>

Tabela 26. Uprawnienia w systemie AIX dla wykonawczego identyfikatora użytkownika

Zadanie	Uprawnienia
Tworzenie obiektów metadanych	SELECT, INSERT
Modyfikowanie obiektów metadanych	SELECT, INSERT, UPDATE, DELETE
Usuwanie obiektów metadanych	SELECT, DELETE
Eksportowanie obiektów metadanych do pliku XML	SELECT
Importowanie obiektów metadanych do programu DB2 UDB z pliku XML	SELECT, INSERT, UPDATE, DELETE
Uruchamianie Doradcy optymalizacyjnego	SELECT do tabel systemowych i tabel podstawowych
Uruchamianie komend DB2 generowanych przez Doradcę optymalizacyjnego	<ul style="list-style-type: none"> <li>• CREATEIN, DROPIN do schematu DB2INFO</li> <li>• SELECT i ALTER (lub CONTROL) do tabel podstawowych</li> </ul>

Niektóre zadania konfiguracyjne oraz instalacyjne można przeprowadzić przy użyciu Centrum OLAP. Zadania te wymagają nadania dodatkowych uprawnień połączeniowemu identyfikatorowi użytkownika.

Tabela 27. Uprawnienia konfiguracyjne oraz instalacyjne w systemie AIX dla połączeniowego identyfikatora użytkownika

Zadanie	Uprawnienia
Wszystkie zadania konfiguracyjne oraz instalacyjne	<ul style="list-style-type: none"> <li>• SYSADM do bazy danych, w której zainstalowano program DB2 Cube Views, do tworzenia schematów dla obiektów metadanych</li> <li>• CONNECT, CREATETAB, IMPLICIT_SCHEMA do bazy danych, w której ma być zainstalowany program DB2 Cube Views</li> <li>• CREATEIN, DROPIN do schematu DB2INFO zawartego w bazie danych, w której zainstalowano program DB2 Cube Views</li> <li>• SELECT, INSERT, UPDATE, DELETE, CONTROL do wszystkich tabel w schemacie DB2INFO dla bazy danych, w której zainstalowano program DB2 Cube Views</li> </ul>

---

## Tworzenie obiektów metadanych programu DB2 Cube Views

Obiekty metadanych programu DB2 Cube Views można tworzyć przy użyciu Centrum OLAP.

### Wymiana metadanych między programem DB2 Cube Views i narzędziami OLAP

Centrum OLAP umożliwia nawiązywanie połączenia z bazą danych DB2 UDB oraz importowanie i eksportowanie obiektów metadanych.

#### Nawiązywanie połączenia z bazą danych DB2

Z bazą danych DB2 można się połączyć lub rozłączyć.

Przed połączeniem się z bazą danych DB2 przy użyciu Centrum OLAP, należy skonfigurować bazę danych do użytkowania z metadanymi programu DB2 Cube Views. Jeśli baza danych, z którą nawiązywane jest połączenie, nie jest skonfigurowana do użytkowania z metadanymi programu DB2 Cube Views, przy próbie połączenia zostanie wyświetlony komunikat. Centrum OLAP umożliwia realizację zadań konfiguracyjnych, jeśli identyfikator użytkownika używany do łączenia ma odpowiednie uprawnienia.

**Uwaga:** Jeśli połączenie z bazą danych już istnieje i nawiązywane będzie połączenie z inną bazą danych, nastąpi rozłączenie z pierwszą bazą danych. Wszystkie metadane w pierwszej bazie danych przestaną być widoczne w drzewie obiektów Centrum OLAP.

Aby połączyć się z bazą danych DB2:

1. Otwórz okno połączenia z bazą danych, klikając opcje: **Centrum OLAP** → **Połącz**, i określ wymagane informacje.
  - W polu **Nazwa bazy danych** wybierz bazę danych, z którą ma zostać nawiązane połączenie.
  - W polu **Nazwa użytkownika** wpisz nazwę użytkownika odpowiednią dla wybranej bazy danych.
  - W polu **Hasło** wpisz hasło odpowiadające podanej nazwie użytkownika.
2. Kliknij przycisk **OK**. Obiekty metadanych w bazie danych DB2, z którą zostało nawiązane połączenie, zostaną wyświetlone w drzewie obiektów Centrum OLAP.

Aby rozłączyć się z bazą danych DB2:

Kliknij opcje: **Centrum OLAP** → **Rozłącz** w głównym oknie Centrum OLAP. Wszystkie obiekty metadanych zostaną usunięte z drzewa obiektów Centrum OLAP.

#### Importowanie metadanych

Do programu DB2 UDB można zaimportować obiekty metadanych i zarządzać nimi przy użyciu programu DB2 Cube Views.

Aby można było importować obiekty metadanych do programu DB2 UDB, musi istnieć plik XML programu DB2 Cube Views. Plik XML programu DB2 Cube Views można utworzyć przy użyciu mostu metadanych do aplikacji dostawcy, eksportując metadane z istniejącej bazy danych DB2 UDB lub używając edytora XML.

Można importować pliki XML w wersji 8.1 i w wersji 8.2 programu DB2 Cube Views. Podczas tworzenia obiektów Kreator importu może przekształcić obiekty metadanych w wersji 8.1 w obiekty metadanych w wersji 8.2.

Aby zaimportować metadane:

1. Otwórz Kreatora importu, klikając opcje: **Centrum OLAP -> Importuj**. Zostanie otwarty Kreator importu.
2. Na stronie Źródło określ plik XML z informacjami o metadanych przewidzianych do zaimportowania.
3. Na stronie Opcje importu przejrzyj listę obiektów metadanych, które mają zostać zaimportowane, i określ opcje importu. Można sprawdzić, czy dany obiekt już istnieje, czy zostanie utworzony po zaimportowaniu obiektów.
4. Na stronie Podsumowanie przejrzyj opcje wybrane dla importu metadanych.
5. Kliknij przycisk **Zakończ**, aby zaimportować obiekty metadanych. Zaimportowane obiekty metadanych zostaną wyświetlone w drzewie obiektów Centrum OLAP.

## Eksportowanie metadanych

Obiekty metadanych programu DB2 Cube Views można eksportować, dzięki czemu mogą być używane w aplikacjach partnerów biznesowych.

Przed wyeksportowaniem obiektów metadanych do pliku XML programu DB2 Cube Views należy utworzyć obiekty metadanych, które mają zostać wyeksportowane.

Aby wyeksportować metadane:

1. Otwórz okno Eksportuj, klikając opcje: **Centrum OLAP -> Eksportuj**.
2. Wybierz wersję metadanych do wyeksportowania. W razie korzystania z interfejsu do aplikacji używanej przez partnera upewnij się, którą wersję metadanych obsługuje aplikacja docelowa.
3. Wybierz opcję **Wszystkie obiekty metadanych**, aby wyeksportować wszystkie obiekty metadanych w bazie danych, lub opcję **Wybrane obiekty metadanych**, aby określić zestaw metadanych do wyeksportowania. Można wybrać jeden model kostki albo dowolną liczbę kostek o jednakowym nadrzędnym modelu kostki. W razie wybrania jednej lub kilku kostek model kostki jest eksportowany automatycznie.
4. Określ nazwę pliku XML, do którego mają być eksportowane metadane. Jeśli plik już istnieje, zostanie zastąpiony.
5. Kliknij przycisk **OK**. Zostanie utworzony plik XML programu DB2 Cube Views zawierający informacje o określonych obiektach metadanych. Plik XML programu DB2 Cube Views może być używany w aplikacjach partnerów biznesowych.

## Tworzenie modelu kostki przy użyciu Kreatora szybkiego startu

Możliwe jest utworzenie modelu kostki oraz odpowiadających mu obiektów faktów, miar, wymiarów, atrybutów i łączy wszystkich jednocześnie na podstawie schematu relacyjnego.

Aby można było utworzyć model kostki i odpowiadające mu metadane, najpierw należy zdefiniować ograniczenia zapewniające spójność referencyjną dla tabel w bazie danych.

Kreator szybkiego startu tworzy obiekty metadanych, które można logicznie wywieść ze schematu. Należy określić tabelę faktów i kolumny miar, a kreator wykryje odpowiednie wymiary, łączenia oraz atrybuty. Gdy Kreator szybkiego startu zakończy działanie, można samodzielnie dodawać, usuwać i modyfikować obiekty metadanych według potrzeb.

Aby utworzyć model kostki i odpowiednie metadane:

1. Otwórz Kreatora szybkiego startu, klikając prawym przyciskiem myszy bazę danych lub folder **Modele kostki** w drzewie obiektów Centrum OLAP i wybierając opcję **Utwórz model kostki - Szybki start**. Zostanie otwarty Kreator szybkiego startu. Jeśli drzewo obiektów nie zawiera folderu **Modele kostki**, upewnij się, że aktywny jest widok obiektów OLAP, klikając opcje: **Widok -> Pokaż obiekty OLAP**.

Na stronie Wprowadzenie znajduje się opis możliwości Kreatora szybkiego startu.



2. Na stronie Tabela faktów wybierz schemat i tabelę faktów dla modelu kostki. Wybrana tabela faktów jest bezpośrednio odwzorowywana na obiekt faktów w modelu kostki. Kolumny używane w kluczach obcych w określonej tabeli faktów będą odwzorowywane na obiekty atrybutów w modelu kostki. Kreator nie jest w stanie wykryć wymiarów niejawnych (wymiary, których kolumny istnieją w tabeli faktów). Wymiary niejawne można dodać do modelu kostki, gdy Kreator szybkiego startu zakończy działanie.
3. Na stronie Miary wybierz z określonej tabeli faktów kolumny, które mają być bezpośrednio odwzorowywane na miary. Klucze podstawowych i klucze obcych nie można odwzorowywać na miary i nie są wymienione na liście. Domyślną funkcją agregacji jest SUM w wypadku miar opartych na kolumnach liczbowych, a COUNT w wypadku miar opartych na kolumnach znakowych. W tym kreatorze nie można tworzyć miar obliczanych; miary obliczane można utworzyć, gdy Kreator szybkiego startu zakończy działanie.
4. Na stronie Podsumowanie przejrzyj utworzone obiekty metadanych. Kliknij przycisk **Zakończ**, aby utworzyć model kostki i odpowiadające mu obiekty metadanych. Gdy zostanie kliknięty przycisk **Zakończ**, Kreator szybkiego startu utworzy w katalogu DB2 następujące obiekty metadanych:
  - Model kostki zawierający wszystkie pozostałe obiekty metadanych.
  - Obiekt faktów odpowiadający określonej tabeli faktów.
  - Miary odpowiadające określonym kolumnom tabeli faktów.
  - Wymiary odpowiadające każdej tabeli wymiarów, dla której istnieje łączenie z tabelą faktów. Tabele pomocnicze to tabele mające łączenie z tabelą wymiarów i uwzględnione w odpowiednim obiekcie wymiarów.
  - Atrybuty odpowiadające każdej kolumnie w tabelach wymiarów i w tabelach pomocniczych oraz wszelkim kluczom obcym w tabeli faktów.
  - Obiekty łączeń służące jako łączenia faktów z wymiarami oraz łączenia wewnątrz obiektu wymiaru łączące tabelę wymiarów i wszelkie odpowiadające jej tabele pomocnicze.

Do modelu kostki można dodawać hierarchie, miary obliczane i kostki.

## Tworzenie kompletnego modelu kostki

Aby utworzyć kompletny model kostki, należy najpierw utworzyć pusty model kostki, a następnie dodać obiekt faktów, wymiary oraz hierarchie i poziomy dla każdego wymiaru w danym modelu kostki. Modele kostki definiują związki między danymi relacyjnymi w schemacie gwiazdzistym lub schemacie typu płatek śniegu, umożliwiające optymalizowanie zapytań kierowanych do danych relacyjnych.

Można utworzyć pusty model kostki lub - wykonując poniższe czynności - utworzyć wszystkie obiekty niezbędne do skompletowania modelu kostki. Po skompletowaniu modelu kostki można utworzyć obiekty kostki, które mogą zawierać wszystkie właściwości obiektów modelu kostki lub ich podzbiór.

Model kostki programu DB2 Cube Views stanowi reprezentację logiczną schematu gwiazdzistego lub schematu typu płatek śniegu. Model kostki powstaje przez zgrupowanie odpowiednich obiektów wymiarów wokół centralnego obiektu faktów. Każdy wymiar może mieć wiele hierarchii, zwiększających elastyczność modelu kostki. W modelu kostki przechowywane są informacje strukturalne o sposobie łączenia tabel używanych w obiektach faktów i wymiaru. W modelu kostki przechowywane są także informacje wystarczające do pobierania danych OLAP. Inne narzędzia raportowania i narzędzia OLAP, które są w stanie zinterpretować model kostki i mogą obsługiwać wiele widoków określonego wymiaru, mogą korzystać z wielu hierarchii zdefiniowanych dla pojedynczego wymiaru.

Aby utworzyć kompletny model kostki:

1. Utwórz model kostki.
2. Utwórz obiekt faktów.
3. Utwórz wymiar.
4. Utwórz jedną hierarchię na jeden wymiar.
5. Utwórz poziomy dla każdej z hierarchii.

## Tworzenie modelu kostki

Model kostki tworzy się w celu zgrupowania obiektów, których wymiary są ze sobą powiązane, wokół centralnego obiektu faktów.

W modelu kostki przechowywane są informacje strukturalne o sposobie łączenia tabel używanych w obiektach faktów i wymiaru. Można tworzyć kostki z podzbiorami obiektów pochodzących z modelu kostki, aby móc selektywnie eksponować wiążące się ze sobą fakty i wymiary dla potrzeb aplikacji.

Kreator modeli kostek prowadzi przez procedurę tworzenia pustego modelu kostki. Po utworzeniu modelu kostki przy użyciu tego kreatora dodaj obiekt faktów oraz wymiary z odpowiadającymi im obiektami łączy. Wymiary mogą być współużytkowane przez różne modele kostek.

Aby utworzyć model kostki:

1. Otwórz Kreatora modeli kostek, prawym przyciskiem myszy klikając bazę danych lub folder **Modele kostki** w drzewie obiektów Centrum OLAP, a następnie klikając opcję **Utwórz model kostki**. Jeśli drzewo obiektów nie zawiera folderu **Modele kostki**, upewnij się, że aktywny jest widok obiektów OLAP, klikając opcje: **Widok → Pokaż obiekty OLAP**.
2. Na stronie **Nazwa** określ nazwę, nazwę biznesową, schemat i komentarz do modelu kostki.
3. Kliknij przycisk **Zakończ**, aby utworzyć model kostki.

## Tworzenie obiektu faktów

Obiekt faktów tworzy się w ramach istniejącego modelu kostki. Obiekt faktów grupuje pokrewne miary, interesujące pod kątem określonego zastosowania. Obiekt faktów stanowi w modelu kostki centrum schematu gwiazdowego. Można utworzyć miary proste i obliczane.

Aby można było utworzyć obiekt faktów, najpierw należy utworzyć model kostki.

Aby utworzyć obiekt faktów:

1. Otwórz Kreatora faktów, prawym przyciskiem myszy klikając model kostki, a następnie klikając opcję **Utwórz fakty**. Jeśli opcja **Utwórz fakty** nie jest dostępna, oznacza to, że obiekt faktów już jest zdefiniowany w wybranym modelu kostki.
2. Na stronie **Nazwa** określ nazwę, nazwę biznesową i schemat obiektu faktów. Można również wpisać komentarze do obiektu faktów. Wszystkie obliczane i nieobliczane miary utworzone w Kreatorze faktów będą mieć taką samą nazwę schematu, jaka została określona dla obiektu faktów.
3. Na stronie **Tabele** wybierz jedną lub więcej tabel faktów z listy dostępnych tabel.
4. Na stronie **Łączenia** określ łączenia między wybranymi tabelami. Można wybrać istniejące łączenie lub kliknąć przycisk **Utwórz łączenie** w celu otwarcia Kreatora łączy. Wszystkie wybrane tabele muszą mieć łączenia bez pętli. W obrębie każdej pary tabel można zdefiniować tylko jedno łączenie. Jeśli na stronie **Tabele** wybrana jest tylko jedna tabela, łączy nie trzeba określać.
5. Na stronie **Miary** utwórz miary proste, miary obliczane lub i takie i takie.

### Miary proste

Miary odwzorowywane bezpośrednio na kolumnę. Przeniesienie kolumny na listę wybranych miar spowoduje utworzenie miary prostej mającej bezpośrednie odwzorowanie na kolumnę.

### Miary obliczane

Miary, które odwołują się do wyrażenia i są tworzone na podstawie przynajmniej jednej kolumny, atrybutu lub innej miary. Aby utworzyć miarę obliczaną, kliknij przycisk **Utwórz miarę obliczaną**.

6. Na stronie Agregacja określ prostą funkcję agregacji dla każdej miary. Domyślną funkcją agregacji jest SUM dla miar o liczbowym typie danych. Dla miar o znakowym typie danych domyślną funkcją agregacji jest COUNT. Aby zmienić agregację, kliknij agregację w tabeli i wybierz inną funkcję lub brak funkcji.
7. Kliknij przycisk **Zakończ**, aby utworzyć obiekt faktów.

## Tworzenie wymiaru

Wymiar można utworzyć jako część modelu kostki lub osobno w celu wykorzystania go w późniejszym czasie.

Aby można było utworzyć wymiar dla modelu kostki, najpierw należy utworzyć obiekt faktów.

Obiekt wymiaru definiuje zestaw pokrewnych atrybutów i połączeń między atrybutami, które razem opisują pewien aspekt miary. Gdy połączenia są niezbędne do grupowania atrybutów, na przykład w schemacie typu płatek śniegu, niezbędne połączenia oraz atrybuty używane w tych połączeniach można znaleźć na liście w definicji wymiaru. Wymiar o nazwie Region może zawierać atrybuty takie, jak Stan, Kraj, Miasto i Populacja. Wymiary odwołują się do hierarchii, które mogą służyć do poruszania się po wymiarze i do wykonywania na jego podstawie obliczeń. Typ wymiaru opisuje jego charakter.

Aby utworzyć wymiar:

1. Otwórz Kreatora wymiarów, prawym przyciskiem myszy klikając model kostki, a następnie klikając opcję **Utwórz wymiar**.
2. Na stronie Nazwa określ nazwę, nazwę biznesową, schemat i komentarze do wymiaru. Wszystkie atrybuty utworzone w Kreatorze wymiarów będą mieć taką samą nazwę schematu, jaka została określona dla wymiaru.
3. Na stronie Tabele wybierz tabele źródłowe, na podstawie których ma być utworzony wymiar. Należy wybrać co najmniej jedną tabelę.
4. Na stronie Łączenia określ połączenia między wybranymi tabelami. Można wybrać istniejące połączenie lub kliknąć przycisk **Utwórz połączenie** w celu otwarcia kreatora umożliwiającego utworzenie połączenia. Wszystkie wybrane tabele muszą mieć połączenia bez pętli. Między każdymi dwiema tabelami można określić tylko jedno połączenie. Jeśli na stronie **Tabele** jest wybrana tylko jedna tabela, połączeń nie trzeba określać.
5. Na stronie Atrybuty określ atrybuty, które mają być uwzględnione w wymiarze. Należy określić co najmniej jeden atrybut. Można wybrać atrybut z listy lub kliknąć przycisk **Utwórz atrybut obliczany**, aby otworzyć okno umożliwiające tworzenie wyrażenia. W drzewie obiektów widoczne są tabele źródłowe oraz ich dostępne kolumny i atrybuty. Atrybut jest widoczny, jeśli już istnieje w bazie danych, która jest odwzorowywana na jedną z kolumn w tabeli. Jeśli kolumna nie jest odwzorowywana na atrybut, widoczna jest sama kolumna. Jeśli kolumna zostanie wybrana i przeniesiona na listę **Wybrane atrybuty**, zostanie utworzony atrybut odwzorowywany na kolumnę. Wybierz wszystkie atrybuty, które mają być używane w hierarchii lub hierarchiach wymiaru. Centrum OLAP automatycznie doda do wymiaru wszelkie identyfikatory atrybutów używane w określonych połączeniach.

6. Na stronie Typ określ typ wymiaru, który ma być utworzony. Wybierz opcję **Czas**, aby określić, że wymiar jest wymiarem czasowym, albo wybierz opcję **Zwykły**, aby określić, że wymiar nie jest wymiarem czasowym.
7. Na stronie łączenia faktów z wymiarem określ obiekt łączenia, aby połączyć wymiar z obiektem faktów. Łączenie można również utworzyć przy użyciu Kreatora łączy. Ta strona jest wyświetlana tylko wtedy, gdy wymiar jest tworzony w ramach modelu kostki.
8. Kliknij przycisk **Zakończ**, aby utworzyć wymiar.

## Tworzenie hierarchii dla wymiaru

Hierarchia definiuje związki między dwoma poziomami lub większą ich liczbą w ramach danego wymiaru modelu kostki. Można również zdefiniować hierarchię, w której używany jest tylko jeden poziom. Związki te umożliwiają poruszanie się po określonym wymiarze i dokonywanie na nim obliczeń. Na przykład hierarchia RokKalendarzowy może zawierać poziomy takie, jak Rok, Kwartał i Miesiąc.

Aby można było utworzyć hierarchię dla wymiaru, najpierw należy utworzyć wymiar.

Dla wymiaru w modelu kostki można zdefiniować wiele hierarchii. O związku między poziomami decyduje typ hierarchii.

Aby utworzyć hierarchię dla wymiaru:

1. Otwórz Kreatora hierarchii. W tym celu rozwiń folder **Wymiary**, aby wyświetlić istniejące wymiary. Prawym przyciskiem myszy kliknij wymiar, a następnie kliknij opcję **Utwórz hierarchię**.
2. Na stronie Nazwa określ nazwę, nazwę biznesową, schemat i dowolne komentarze do hierarchii.
3. Na stronie Poziomy określ poziomy dla hierarchii oraz typ hierarchii:
  - a. Jeśli na liście **Dostępne poziomy** nie ma żadnych poziomów, kliknij opcję **Utwórz poziom** i utwórz poziomy, używając Kreatora tworzenia poziomów.
  - b. Wybierz poziomy. Należy wybrać co najmniej jeden poziom.
  - c. Określ typ i rozwinięcie hierarchii. Jeśli typ hierarchii zostanie określony jako **Rekurencyjny**, można wybrać tylko dwa poziomy.

Po wybraniu przynajmniej jednego poziomu można kliknąć opcję **Pokaż przykład**, aby wyświetlić przykładowe dane w hierarchii.

4. Kliknij przycisk **Zakończ**, aby dodać nową hierarchię do modelu kostki.

## Tworzenie poziomów dla każdej hierarchii

Poziomy definiują związki w zestawie atrybutów pokrewnych. Jeśli jest to możliwe, program DB2 Cube Views tworzy zależności funkcjonalne definiujące związki między atrybutami poziomu.

1. Otwórz Kreatora poziomów, rozwijając wymiar w folderze **Wymiary** dla modelu kostki w drzewie obiektów Centrum OLAP. Prawym przyciskiem myszy kliknij folder **Poziomy**, a następnie kliknij opcję **Utwórz poziom**.
2. Na stronie Nazwa określ nazwę, nazwę biznesową, schemat i komentarz do modelu kostki.
3. Na stronie Atrybuty klucza poziomu wybierz jeden lub więcej atrybutów, które razem w sposób unikalny definiują poziom.
4. Na stronie Atrybut domyślny wybierz jeden atrybut opisujący dane poziomu.
5. Na stronie Atrybuty pokrewne wybierz zero lub więcej atrybutów dostarczających więcej informacji o danym poziomie.
6. Na stronie Optymalizuj za pomocą zależności funkcjonalnej określ, czy mają być utworzone stosowne zależności funkcjonalne między atrybutami klucza poziomu a

atrybutami domyślnym i pokrewnymi. To pole wyboru należy zaznaczyć, jeśli jest pewne, że atrybuty klucza poziomu funkcjonalnie determinują zarówno atrybut domyślny, jak i dowolne atrybuty pokrewne.

7. Kliknij przycisk **Zakończ**, aby utworzyć poziom.

## Dodawanie istniejącego wymiaru do modelu kostki

Możliwe jest dodanie istniejącego wymiaru do modelu kostki. Wymiar może już istnieć, jeśli jest używany w innym modelu kostki albo jeśli był już używany w innym modelu kostki, lecz później został usunięty.

Aby wymiar można było dodać, najpierw należy utworzyć model kostki oraz obiekt faktów. Musi także istnieć wymiar niebędący częścią danego modelu kostki.

Aby dodać istniejący wymiar:

1. Otwórz Kreatora dodawania wymiarów, prawym przyciskiem myszy klikając model kostki, a następnie klikając opcję **Dodaj wymiary**.
2. Na stronie Wymiary wybierz jeden lub więcej istniejących wymiarów w celu dodania do modelu kostki. Należy wybrać co najmniej jeden wymiar.
3. Na stronie Łączenia faktów z wymiarem określ obiekt łączenia w celu ustanowienia łączenia dodanego wymiaru z obiektem faktów modelu kostki.

Jeśli już istnieje jedno odpowiednie łączenie dla wymiaru, łączenie to jest widoczne dla odpowiadającego mu wymiaru. Upewnij się, że to domyślne łączenie jest sensowne, że atrybuty po jednej stronie łączenia odwołują się do kolumn w odpowiedniej tabeli wymiarów oraz że atrybuty po drugiej stronie łączenia odwołują się do kolumn w tabeli faktów.

Jeśli nie istnieją odpowiednie łączenia lub takich łączeń jest więcej niż jedno, łączenie należy określić samodzielnie. Aby określić inne łączenie lub utworzyć łączenie dla konkretnego wymiaru, wybierz wymiar i kliknij przycisk **Określ łączenie**.

4. Kliknij przycisk **Zakończ**, aby dodać wymiar do modelu kostki.

## Określanie łączenia faktów z wymiarem dla istniejącego wymiaru

Możliwe jest określenie łączenia faktów z wymiarem dla istniejącego wymiaru w modelu kostki.

- Aby określić istniejące łączenie:
  1. Wybierz łączenie z listy proponowanych łączeń.
  2. Kliknij przycisk **OK**.
- Aby utworzyć łączenie:
  1. Kliknij przycisk **Utwórz łączenie**. Zostanie otwarty Kreator łączeń.
  2. Utwórz potrzebne łączenie. Nowe łączenie zostanie wyświetlone na liście istniejących proponowanych łączeń.
  3. Z listy proponowanych łączeń wybierz utworzone łączenie.
  4. Kliknij przycisk **OK**.

## Tworzenie łączenia

Łączenie można utworzyć w celu połączenia wymiaru z obiektem faktów, w ramach obiektu faktów lub w ramach wymiaru.

Obiekt łączenia pozwala połączyć dwie tabele relacyjne. Łączenie odwołuje się do atrybutów, które z kolei odwołują się do kolumn w tabelach podlegających łączeniu. Łączenie ma także określony typ i licznosc.

Najprostsze łączenie odwołuje się do dwóch atrybutów: jednego mającego odwzorowanie na kolumnę w pierwszej tabeli i jednego mającego odwzorowanie na kolumnę w drugiej tabeli, przy czym operator wskazuje sposób, w jaki te kolumny powinny być porównywane.

Obiekt łączenia może również służyć do modelowania łączeń złożonych, w których co najmniej dwie kolumny z pierwszej tabeli są łączone z taką samą liczbą kolumn w drugiej tabeli. W łączeniu złożonym odpowiadające sobie kolumny są kojarzone za pomocą par atrybutów. Każdej parze atrybutów towarzyszy operator wskazujący sposób porównywania kolumn.

Obiektów łączenia używa się przede wszystkim w modelu kostki w celu połączenia wymiarów modelu kostki z jej obiektem faktów. Łączenia mogą również służyć do łączenia razem tabel wymiarów w schemacie typu płatek śniegu, a niekiedy są używane w obiekcie faktów do łączenia wielu tabel faktów.

Aby utworzyć łączenie:

1. Otwórz Kreatora łączeń.
2. Na stronie Nazwa określ nazwę, nazwę biznesową, schemat i dowolne komentarze do łączenia.

**Wskazówka:** Nadaj łączeniu nazwę składającą się z nazw obu łączonych tabel. Na przykład jeśli łączeniu podlegają tabela FaktySprzedaży i tabela Produkt, nadaj łączeniu nazwę: FaktySprzedaży-Produkt.

3. Na stronie Łączenie dodaj jedną lub więcej par atrybutów oraz wybierz typ i licznosc łączenia. W celu utworzenia pary atrybutów wybierz atrybut lewostronny i atrybut prawostronny, a następnie kliknij przycisk **Dodaj**. Para pojawi się w tabeli par atrybutów. Domyślnym operatorem łączenia jest =. Operator można zmienić, klikając bieżący operator w tabeli i wybierając nowy operator. Domyślnym typem łączenia jest Wewnętrzne, a domyślną licznoscia jest 1:1.
4. Kliknij przycisk **Zakończ**, aby utworzyć łączenie.

## Tworzenie kostki

Kostkę można utworzyć w celu określenia najistotniejszych regionów modelu kostki. Kostki można również użyć do zdefiniowania podzbioru danych dla aplikacji partnera biznesowego.

Aby można było utworzyć kostkę, najpierw należy utworzyć lub zaimportować model kostki. Model kostki musi mieć obiekt faktów i co najmniej jeden wymiar, a każdy z wymiarów musi mieć jedną hierarchię.

Kostka jest precyzyjną definicją kostki OLAP, którą można przeszukiwać przy użyciu pojedynczej instrukcji SQL. Kostka stanowi pochodną istniejącego modelu kostki. Fakty kostki oraz lista wymiarów kostki są podzbiarami faktów i wymiarów z odnośnego modelu kostki. Kostki są strukturą odpowiednią dla narzędzi i aplikacji niekorzystających z wielu hierarchii, ponieważ wymiary kostki dopuszczają istnienie tylko jednej hierarchii przypadającej na wymiar.

Aby utworzyć kostkę:

1. Otwórz Kreatora kostek, prawym przyciskiem myszy klikając model kostki, a następnie klikając opcję **Utwórz kostkę**.
2. Na stronie Nazwa określ nazwę, nazwę biznesową, schemat i komentarze do kostki.
3. Na stronie Miary wybierz miary, które mają być uwzględnione w kostce. Należy wybrać co najmniej jedną miarę.

4. Na stronie Wymiary wybierz wymiary kostki, które mają być uwzględnione w kostce. Należy wybrać co najmniej jeden wymiar kostki. Wybierz wymiar kostki i kliknij przycisk znajdujący się obok wybranego wymiaru kostki, aby otworzyć okno umożliwiające określenie informacji o hierarchii dla danego wymiaru kostki.
5. Na stronie określania typu zapytań określ oczekiwany sposób korzystania z tej kostki. W razie wybrania opcji **Ustawienia zaawansowane** kliknij przycisk **Określ**, aby określić wycinki optymalizacji, co do których można oczekiwać, że będą przeszukiwane najczęściej.
6. Kliknij przycisk **Zakończ**, aby utworzyć kostkę.

## Określanie hierarchii kostki i poziomów kostki dla wymiaru kostki

Można tworzyć niestandardowe kostki do różnych zastosowań, określając - dla poszczególnych wymiarów kostki - poziomy kostki i atrybuty pokrewne, które mają być uwzględnione hierarchii kostki. Można określić podzbiór poziomów kostki i atrybutów pokrewnych, do których będzie się odwoływać hierarchia kostki.

Aby określić hierarchię kostki dla wymiaru kostki przy użyciu okna Wybierz atrybuty dla hierarchii kostki otwartego z poziomu Kreatora kostek:

1. Wybierz hierarchię kostki z listy **Możliwe hierarchie**.
2. Upewnij się, że na liście **Poziomy i atrybuty** zaznaczone jest pole wyboru obok odpowiedniego poziomu kostki lub atrybutu pokrewnego. Po wybraniu poziomu zostają dodane również atrybuty klucza i atrybuty domyślne poziomu.
3. Kliknij przycisk **OK**, aby dodać poziomy kostki i atrybuty pokrewne do hierarchii kostki i wrócić do Kreatora kostek.

## Określanie wycinków optymalizacji kostki

Określenie wycinków optymalizacji dla kostki jest opcjonalne, niemniej stanowi poważną pomoc dla Doradcy optymalizacyjnego przy udostępnianiu tabel podsumowań skoncentrowanych na najważniejszych regionach modelu kostki.

Wycinki optymalizacji wskazują regiony kostki, co do których można oczekiwać, że będą przeszukiwane najczęściej. Typ wycinka optymalizacji określa sposób realizacji zapytań względem odpowiedniej partii kostki. Na przykład, jeśli 50% zapytań obejmuje poziom Miesiąc, można zdefiniować wycinek optymalizacji jako Dowlolny-Miesiąc-Dowlolny. Doradca optymalizacyjny może na podstawie wycinków optymalizacji określonych przez użytkownika zalecić tabele podsumowań bardzo zbliżone do jego potrzeb.

Aby określić wycinki optymalizacji dla kostki:

1. W oknie Kreatora kostek lub Właściwości kostki otwórz stronę Typ zapytań i wybierz opcję **Ustawienia zaawansowane**, a następnie kliknij przycisk **Określ**.
2. Zdefiniuj jeden lub więcej wycinków optymalizacji dla kostki.

### Dodawanie wycinka

Kliknij przycisk **Nowy**, aby dodać wycinek. Nowy wycinek zostanie wyświetlony w interaktywnym polu grafiki oraz jako wiersz tabeli **Lista wycinków optymalizacji**. Po dodaniu wycinka zmodyfikuj go, określając typ zapytania oraz poziom kostki w każdym z wymiarów kostki, co do których można oczekiwać, że będą przeszukiwane najczęściej.

### Modyfikowanie wycinka

Wycinek można zmodyfikować w interaktywnym polu grafiki lub w tabeli **Lista wycinków optymalizacji**, znajdującej się poniżej grafiki.

Aby zmodyfikować wycinek w tabeli, zaznacz wiersz odpowiadający danemu wycinkowi. Kliknij typ lub wymiar kostki, który ma zostać zmodyfikowany i wybierz opcję.

Aby zmodyfikować wycinek w interaktywnym polu grafiki, zmień poziomy, przesuując węzły w wymiarze kostki na inny poziom, oraz zmień typ, prawym przyciskiem myszy klikając wycinek i wybierając typ zapytania.

Dla każdego wymiaru kostki można wybrać jedną z następujących opcji:

- Wybierz opcję **Dowolne**, jeśli nie jest istotne, na jakim poziomie kostki wycinek będzie się znajdował. Doradca optymalizacyjny automatycznie wybierze poziom kostki do optymalizacji.
- Wybierz **Wszystkie**, jeśli często przeszukiwane są dane na najwyższym poziomie agregacji, na przykład w odniesieniu do wszystkich produktów lub regionów.
- Wybierz poziom kostki zdefiniowany dla wymiaru kostki, jeśli wiesz, do którego poziomu kostki odnosi się duża część zapytań. Na przykład można zdefiniować wycinek optymalizacji na poziomach kostki Dowolne-Miesiąc-Dowolne, przy czym poziom **Dowolne** wybrano w wymiarze kostki Rynek, **Miesiąc** w wymiarze kostki Czas oraz **Dowolne** w wymiarze kostki Produkt.

**Zalecenie:** W celu uzyskania najlepszych wyników należy definiować tylko niewielką liczbę (trzy lub mniej) skoncentrowanych wycinków na kostkę.

#### Usunięcie wycinka

Aby usunąć wycinek, można wybrać wycinek w tabeli lub w interaktywnym polu grafiki, a następnie kliknąć przycisk **Usuń**.

---

## Usuwanie wymiaru z modelu kostki

Jeśli wymiar nie jest już potrzebny, można go usunąć z modelu kostki. Wymiar można usunąć logicznie nie usuwając go fizycznie, jeśli jest używany w innym modelu kostki.

Podczas usuwania wymiaru z modelu kostki wykonywane są następujące działania:

- Wymiar jest usuwany tylko z wybranego modelu kostki, lecz pozostaje we wszystkich pozostałych modelach kostek, które się do niego odwołują.
- Wszystkie wymiary kostki oparte na tym wymiarze w danym modelu kostki zostają usunięte z odpowiednich kostek.
- Wymiar nie jest usuwany z bazy danych.
- Wymiar jest dostępny w folderze **Wszystkie wymiary**.

Aby usunąć wymiar z modelu kostki:

1. W drzewie obiektów Centrum OLAP rozwiń folder **Modele kostki**, aby wyświetlić istniejące modele kostek. Jeśli drzewo obiektów nie zawiera folderu **Modele kostki** upewnij się, że wyświetlany jest widok obiektów OLAP, klikając opcję: **Widok** → **Pokaż obiekty OLAP**.
2. Rozwiń węzeł modelu kostki, aby wyświetlić kategorie obiektów zawarte w danym modelu kostki.
3. Rozwiń folder **Wymiary**, aby wyświetlić istniejące wymiary.
4. Prawym przyciskiem myszy kliknij wymiar, a następnie kliknij opcję **Usuń**.



---

## Usuwanie obiektu metadanych z bazy danych

Jeśli obiekt metadanych nie jest już używany w modelu kostki w danej bazie danych, to obiekt ten można usunąć.

Większość obiektów można usuwać korzystając z widoku obiektów OLAP lub widoku obiektów Relacyjne. Obiekty można usunąć tylko wtedy, gdy nie odwołują się do nich inne obiekty.

Aby usunąć obiekt metadanych z bazy danych:

Wybierz jeden lub więcej obiektów w drzewie obiektów Centrum OLAP i prawym przyciskiem myszy kliknij wybrane obiekty, a następnie kliknij opcję **Usuń**. Jeśli w menu nie ma pozycji **Usuń**, oznacza to, że wybranego obiektu nie można usunąć. Gdy zostaje usunięty obiekt nadrzędny, wraz z nim usuwane są wszystkie jego obiekty podrzędne. Na przykład usunięcie wymiaru spowoduje usunięcie także wszystkich odpowiadających mu hierarchii.



---

## Rozdział 4. Scenariusze modelowania biznesowego w programie DB2 Cube Views

W tej sekcji omówiono następujące tematy:

### **Obliczanie ruchu i wartości zapasów w magazynie w czasie**

Przedsiębiorstwo zajmujące się handlem detalicznym, XYZ Retail, korzysta z magazynu do przechowywania zapasów przed wysłaniem ich do sklepu w celu sprzedania. W bazach danych XYZ Retail rejestrowane są na bieżąco dane na temat stanu zapasów w magazynie. Teraz zachodzi potrzeba poddania tych danych analizie.

### **Korelacja kosztów reklamy ze sprzedażą**

Pewien dealer samochodowy zastanawia się nad zwiększeniem nakładów na reklamę. Aby podjąć trafną decyzję, kierownictwo zamierza najpierw przeanalizować dane historyczne dotyczące związków między wydatkami na reklamę a wysokością sprzedaży. Należy przy tym ustalić, czy zmiana wydatków na reklamę miała jakikolwiek wpływ na sprzedaż, a w szczególności, czy nasilenie reklamy daje się ściśle skorelować ze zwiększeniem sprzedaży.

### **Obliczanie zysku i marży sklepu**

Kierownik sklepu z zabawkami chce mieć możliwość analizowania wpływu różnych czynników, takich jak pora roku i typ towaru, na zysk i marżę.

### **Zliczanie zamówień internetowych**

Pewne przedsiębiorstwo zajmujące się handlem detalicznym kilka lat temu poszerzyło zakres działalności o sprzedaż przez Internet. Obecnie jego kierownictwo pragnie przeanalizować efektywność internetowego kanału sprzedaży. Jednym z pierwszych wskaźników, jakie są potrzebne w takiej analizie, jest liczba zamówień zrealizowanych za pośrednictwem Internetu.

### **Ranking wysokości obrotów**

Pewne przedsiębiorstwo w ciągu ostatnich kilku lat intensywnie rozbudowywało swoją sieć sklepów z artykułami biurowymi. Obecnie kierownictwo rozważa zamknięcie najmniej aktywnych sklepów, aby obniżyć koszty i zwiększyć zyski.

### **Użycie danych czasowych w tabeli faktów do utworzenia wymiaru czasu**

Przedsiębiorstwo zajmujące się handlem detalicznym, XYZ Retail, modeluje dane ze swoich transakcji sprzedaży w programie DB2 Cube Views, aby skorzystać z oferowanych przezeń funkcji analitycznych. Jednak z uwagi na transakcyjny charakter danych, jedyną dostępną informacją czasową to data skojarzona z odpowiednią transakcją.

---

## Obliczanie ruchu i wartości zapasów w magazynie w czasie

Przedsiębiorstwo zajmujące się handlem detalicznym, XYZ Retail, korzysta z magazynu do przechowywania zapasów przed wysłaniem ich do sklepu w celu sprzedania. W bazach danych XYZ Retail rejestrowane są na bieżąco dane na temat stanu zapasów w magazynie. Teraz zachodzi potrzeba poddania tych danych analizie.

Dla kierownictwa szczególnie interesujące są dwa aspekty funkcjonowania magazynu:

- ruch towarów do i z magazynu
- wartość towarów znajdujących się w magazynie w danym momencie

Pierwszy aspekt, czyli ruch towarów, wymaga rozpatrzenia dostępnych danych w wymiarze czasu. Drugi aspekt, czyli wartość towarów, jest statycznym obrazem stanu magazynu w wybranym momencie.

### Szczegóły scenariusza

W bazach danych firmy XYZ Retail znajduje się tabela faktów zawierająca następujące kolumny związane z magazynem: ILOSC\_PRZYJETA, ILOSC\_WYDANA, ILOSC\_BIEZACA, WARTOSC\_PRODUKTU, ID\_PRODUKTU i ID\_CZASU. Dane są wprowadzane do tabeli w odstępach tygodniowych. W bazie danych znajduje się także tabela Produkt i Czas. Przykładowy zestaw danych z tabeli faktów przedstawia Tabela 28.

Tabela 28. Przykładowe dane z tabeli faktów

ID_PRODUKTU	ID_CZASU	ID_ILOSCI	ILOSC_WYDANA	ILOSC_BIEZACA	WARTOSC_PRODUKTU
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

Wartość ID\_PRODUKTU w każdym z trzech przykładowych wpisów jest identyczna, ponieważ produkt tego samego typu może być wielokrotnie przyjmowany i wydawany z magazynu.

Administrator bazy danych XYZ Retail musi utworzyć trzy różne miary:

#### Przyjęcia

Odzwierciedla ruch towarów w kierunku do magazynu.

#### Wydania

Odzwierciedla ruch towarów z magazynu.

#### Wartość bieżąca

Odzwierciedla wartość towarów w magazynie w danym momencie.

Przy tworzeniu pierwszych dwóch miar, Przyjęcia i Wydania, należy utworzyć odwzorowania odpowiednio do kolumn ILOSC\_PRZYJETA i ILOSC\_WYDANA, po czym zsumować dane po wszystkich wymiarach. Jest to przypadek miary w pełni addytywnej, ponieważ agregacja danych odbywa się wyłącznie przy użyciu funkcji SUM we wszystkich wymiarach. Tabela 29 przedstawia przykładowy zestaw danych z kolumn ILOSC\_PRZYJETA i ILOSC\_WYDANA za trzy miesiące dla produktu z identyfikatorem ID\_PRODUKTU 1234. Miary Przyjęcia i Wydania sumują te miesięczne wartości w celu obliczenia ogólnych ilości towaru przyjmowanych i wydawanych z magazynu w skali kwartału.

Tabela 29. Obliczenie przykładowych danych dla miar w pełni addytywnych Przyjęcia i Wydania dla produktu ID\_PRODUKTU 1234

	Styczeń	Luty	Marzec	Kwartał 1
ILOSC_PRZYJETA	5	20	10	35
ILOSC_WYDANA	0	10	20	30

Miary w pełni addytywne to nierzadko najprostsze i najczęściej tworzone miary. Często służą one jako elementy składowe bardziej złożonych miar. W przypadku miar opartych na liczbowych danych źródłowych Centrum OLAP tworzy domyślnie miarę w pełni addytywną.

Aby utworzyć trzecią miarę, Wartość bieżąca, należy obliczyć wynik mnożenia kolumn WARTOSC\_PRODUKTU i ILOSC\_BIEZACA. Na przykład, jeśli dla identyfikatora ID\_PRODUKTU=1234 wartość jednostkowa wynosi 5, wartość miary Wartość bieżąca odpowiadająca danym przykładowym przedstawia Tabela 30.

*Tabela 30. Obliczanie przykładowych wartości miary Wartość bieżąca dla danych przykładowych i ID\_PRODUKTU 1234*

	Styczeń	Luty	Marzec
ILOSC_BIEZACA	5	10	20
Wartość bieżąca	25	50	100

W następnym kroku dane te należy zagregować wzdłuż wymiarów. Ponieważ jednak miara ta oblicza wartość z określonej chwili, sumowanie jej w wymiarze czasu nie miałoby sensu. Dlatego agregacja będzie sumować dane po wymiarze Produkt i obliczać średnią z wyników w czasie. Taką miarę nazywa się póładdytywną, ponieważ tylko część agregacji realizowana jest z użyciem funkcji SUM.

Miary obliczające dane chwilowe, czyli reprezentujące określony moment w czasie - na przykład wynik comiesięcznej inwentaryzacji - często są miarami póładdytywnymi, ponieważ sumowanie miesięcy w kwartały nie daje sensownych wyników. Jeśli dany towar pozostaje w magazynie przez cały miesiąc, jest on uwzględniany w wartości ILOSC\_BIEZACA za kolejne trzy miesiące oddzielnie. Gdyby wartości ILOSC\_BIEZACA zostały zsumowane w czasie, towar przechowywany w magazynie przez trzy miesiące byłby liczony trzykrotnie. Jak przedstawia to Tabela 31, wartość 25 za kwartał 1 nie jest użyteczna z punktu widzenia obsługi magazynu. Jak wynika z tabeli, w magazynie nigdy nie znajdowało się 25 sztuk produktu, toteż wynik w tej postaci jest pozbawiony sensu.

*Tabela 31. Obliczanie przykładowych danych dla kolumny ILOSC\_BIEZACA za pomocą funkcji SUM dla wymiaru czasu i dla ID\_PRODUKTU 1234*

	Styczeń	Luty	Marzec	Kwartał 1
SUM(ILOSC_BIEZACA)	5	15	5	25

Zamiast korzystać z funkcji SUM we wszystkich wymiarach, można użyć innych funkcji agregujących w wymiarze czasu, takich jak AVG, MIN i MAX. Na przykład przy identycznym zestawie danych przykładowych za styczeń, luty i marzec można użyć drugiej funkcji agregującej w wymiarze czasu, jak to przedstawia Tabela 32. Wynikiem będą sensowne wartości dla kwartału. Miara Wartość bieżąca może reprezentować średnią sumaryczną wartość towaru w magazynie za kwartał, ewentualnie minimalną lub maksymalną wartość w dowolnym momencie w ciągu kwartału.

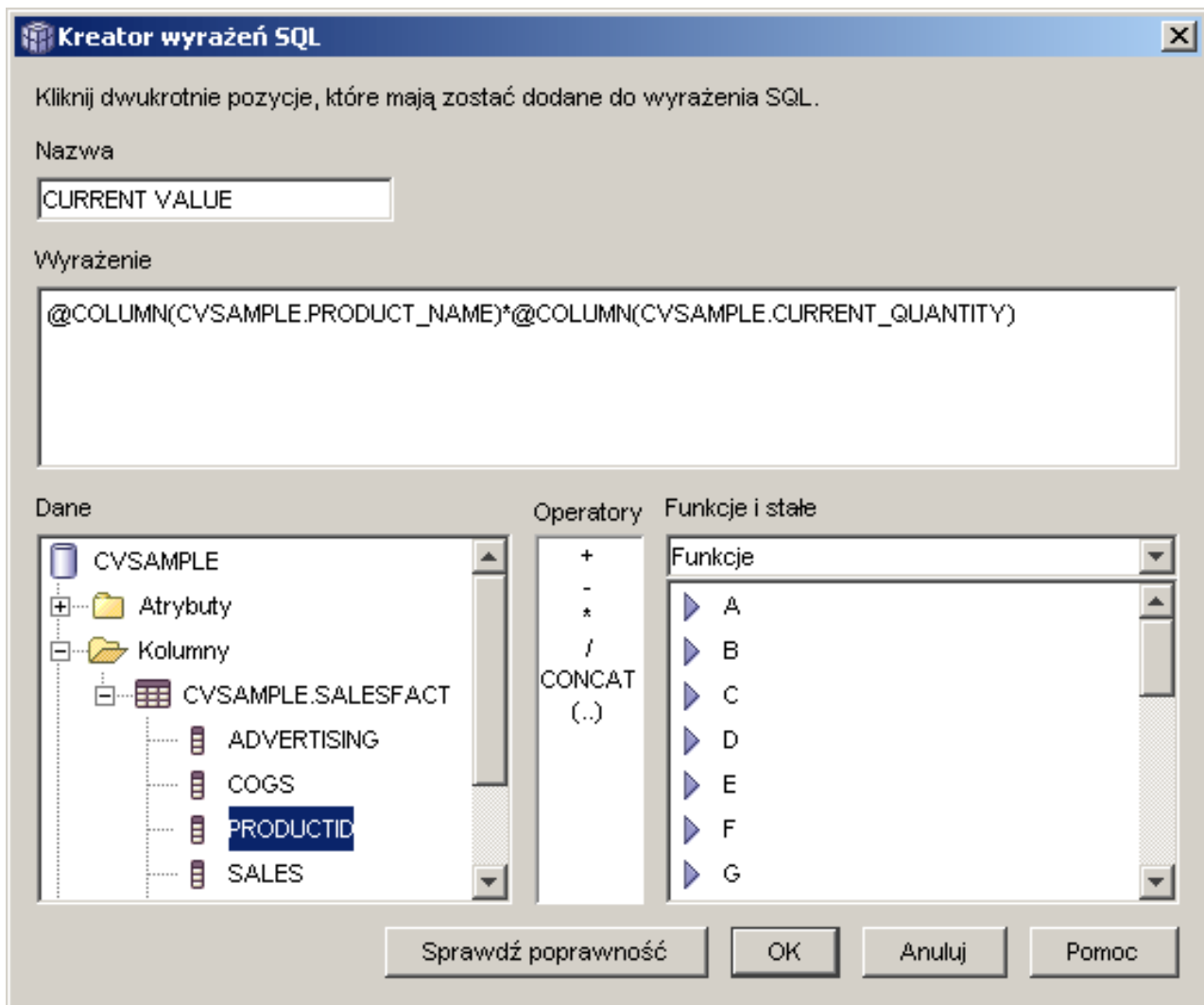
*Tabela 32. Obliczanie przykładowych danych dla kolumny ILOSC\_BIEZACA za pomocą funkcji AVG, MAX i MIN dla wymiaru czasu i dla ID\_PRODUKTU 1234*

	Styczeń	Luty	Marzec	Kwartał 1
AVG(ILOSC_BIEZACA)	5	15	5	8.3
MAX(ILOSC_BIEZACA)	5	15	5	15
MIN(ILOSC_BIEZACA)	5	15	5	5


### **Czynności wymagane do utworzenia miar**

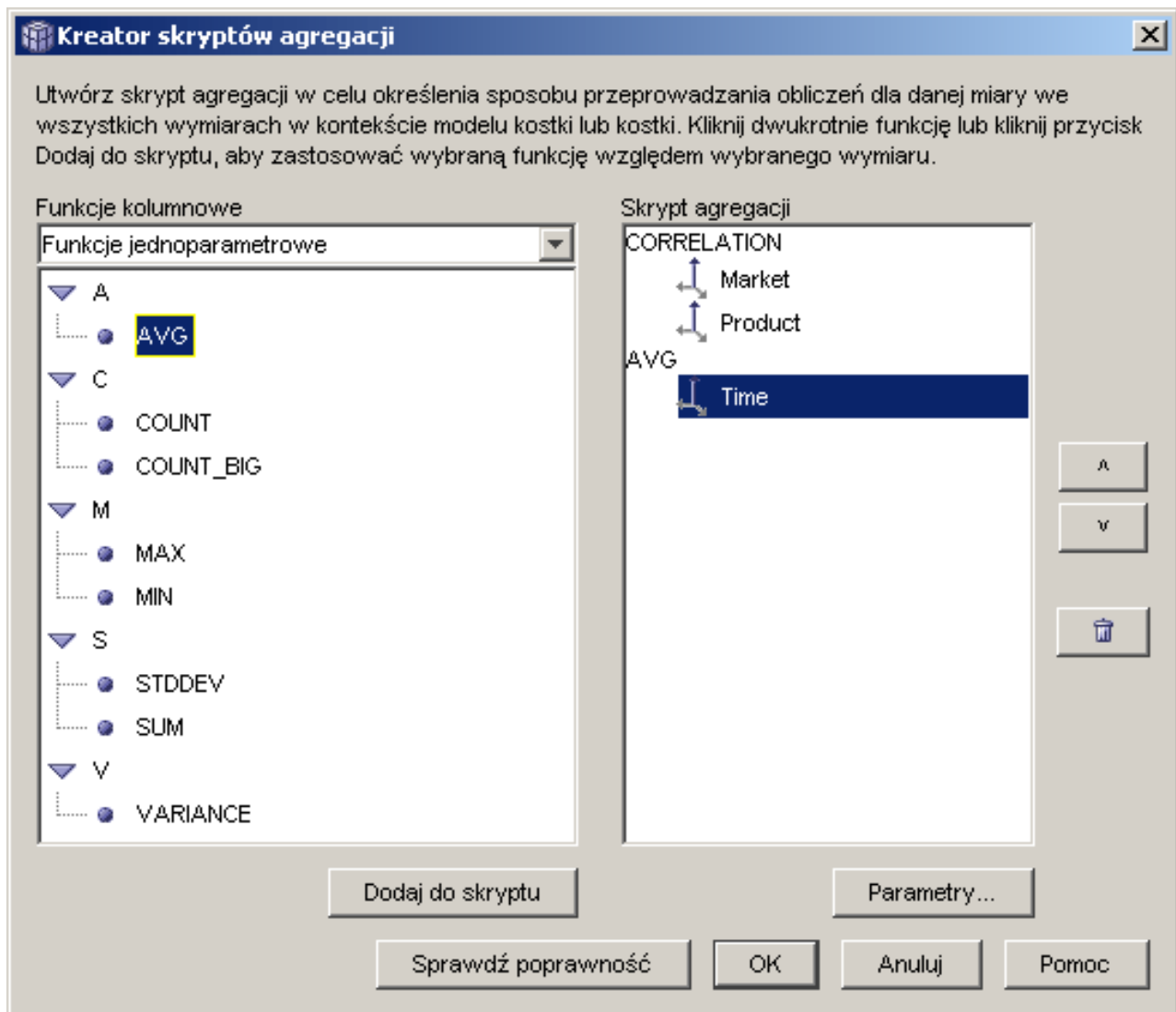
Poniższa procedura opisuje użycie okna Właściwości faktów w Centrum OLAP do utworzenia miar Przyjęcia, Wydania i Wartość bieżąca w istniejącym obiekcie faktów:

1. Aby otworzyć okno Właściwości faktów, kliknij prawym przyciskiem myszy obiekt faktów w drzewie obiektów Centrum OLAP, a następnie kliknij przycisk **Edytuj miary**. Otwarte zostanie okno Właściwości faktów.
  2. Utwórz miarę Przyjęcia:
    - a. Na stronie Miary kliknij opcję **Utwórz miarę obliczaną**, aby utworzyć miarę Przyjęcia. Otwarte zostanie okno Kreatora wyrażeń SQL.
    - b. W oknie Kreatora wyrażeń SQL wpisz **PRZYJĘCIA** w polu **Nazwa**.
    - c. Aby utworzyć wyrażenie obliczające przyjęcia, wykonaj poniższe czynności:
      - Rozwiń folder **Kolumny** i tabelę faktów na liście **Dane**.
      - Kliknij dwukrotnie kolumnę **ILOSC\_PRZYJETA**, aby dodać ją do wyrażenia.
      - Kliknij przycisk **OK**, aby zamknąć okno Kreatora wyrażeń SQL. Na stronie Agregacji nie trzeba zmieniać domyślnej funkcji agregującej, którą jest SUM. Funkcja SUM jest domyślnym wyborem dla miary Przyjęcia, ponieważ dane źródłowe są liczbowe i ponieważ miara odwołuje się do kolumny, a nie do istniejących miar.
  3. Utwórz miarę Wydania:
    - a. Na stronie Miary kliknij opcję **Utwórz miarę obliczaną**, aby utworzyć miarę Wydania. Otwarte zostanie okno Kreatora wyrażeń SQL.
    - b. W oknie Kreatora wyrażeń SQL wpisz **WYDANIA** w polu **Nazwa**.
    - c. Aby utworzyć wyrażenie obliczające wydania, wykonaj poniższe czynności:
      - Rozwiń folder **Kolumny** i tabelę faktów na liście **Dane**.
      - Kliknij dwukrotnie kolumnę **ILOSC\_WYDANA**.
    - d. Kliknij przycisk **OK**, aby zamknąć okno Kreatora wyrażeń SQL. Na stronie Agregacji nie trzeba zmieniać domyślnej funkcji agregującej, którą jest SUM. Funkcja SUM jest domyślnym wyborem dla miary Wydania, ponieważ dane źródłowe są liczbowe, zaś miara odwołuje się do kolumny, a nie tylko do istniejących miar.
  4. Utwórz miarę Wartość bieżąca:
    - a. Na stronie Miary kliknij opcję **Utwórz miarę obliczaną**, aby utworzyć miarę Wartość bieżąca. Otwarte zostanie okno Kreatora wyrażeń SQL.
    - b. W oknie Kreatora wyrażeń SQL wpisz **WARTOŚĆ BIEŻĄCA** w polu **Nazwa**.
    - c. Aby utworzyć wyrażenie miary Wartość bieżąca, wykonaj poniższe czynności:
      - Rozwiń folder **Kolumny** i tabelę faktów na liście **Dane**.
      - Kliknij kolumnę **WARTOSC\_PRODUKTU** na liście **Dane**.
      - Kliknij dwukrotnie operator **\*** na liście **Operatory**.
      - Kliknij kolumnę **ILOSC\_BIEZACA** na liście **Dane**.
- Rys. 12 na stronie 63 przedstawia wyrażenie obliczające wartość bieżącą utworzone w oknie Kreatora wyrażeń SQL.



Rysunek 12. Wpisz wyrażenie Wartość bieżąca w oknie Kreatora wyrażen SQL.

- d. Kliknij przycisk **OK**, aby zamknąć okno Kreatora wyrażen SQL.
- e. Na karcie Agregacje kliknij agregację dla miary Wartość bieżąca i kliknij na liście pozycje **Skrypt agregacji**. Otwarte zostanie okno Kreatora skryptów agregacji. Domyślny skrypt agregacji będzie zawierał funkcję SUM dla wszystkich wymiarów.
- f. W razie potrzeby przenieś wymiar czasu w dół, wybierając pozycję **Czas** i klikając przycisk  przeniesienia w dół, tak aby pozycja ta znalazła się na końcu listy. Następnie, gdy wymiar Czas jest nadal zaznaczony, kliknij dwukrotnie funkcję **AVG** na liście **Funkcje kolumnowe**. Skrypt agregacji, który przedstawia Rys. 13 na stronie 64, sumuje dane po wszystkich wymiarach z wyjątkiem Czas, w którym to wymiarze obliczana jest średnia danych.



Rysunek 13. Skrypt agregacji dla miary Wartość bieżąca

- g. W oknie Kreatora skryptów agregacji kliknij przycisk **Sprawdź poprawność**, aby upewnić się, że skrypt agregacji jest poprawny. Kliknij przycisk **OK**, aby zapisać skrypt agregacji i zamknąć okno.
5. Kliknij przycisk **OK**, aby zapisać zmiany w obiekcie faktów i zamknąć okno Właściwości faktów.

W wyniku wykonanej procedury otrzymaliśmy trzy miary obliczane związane z zapasami w magazynie. Miar tych można używać do analizowania prawidłowości w ruchu towarów do i z magazynu.

## Korelacja kosztów reklamy ze sprzedażą

Pewien dealer samochodowy zastanawia się nad zwiększeniem nakładów na reklamę. Aby podjąć trafną decyzję, kierownictwo zamierza najpierw przeanalizować dane historyczne dotyczące związków między wydatkami na reklamę a wysokością sprzedaży. Należy przy tym ustalić, czy zmiana wydatków na reklamę miała jakikolwiek wpływ na sprzedaż, a w szczególności, czy nasilenie reklamy daje się ściśle skorelować ze zwiększeniem sprzedaży.



## Szczegóły scenariusza

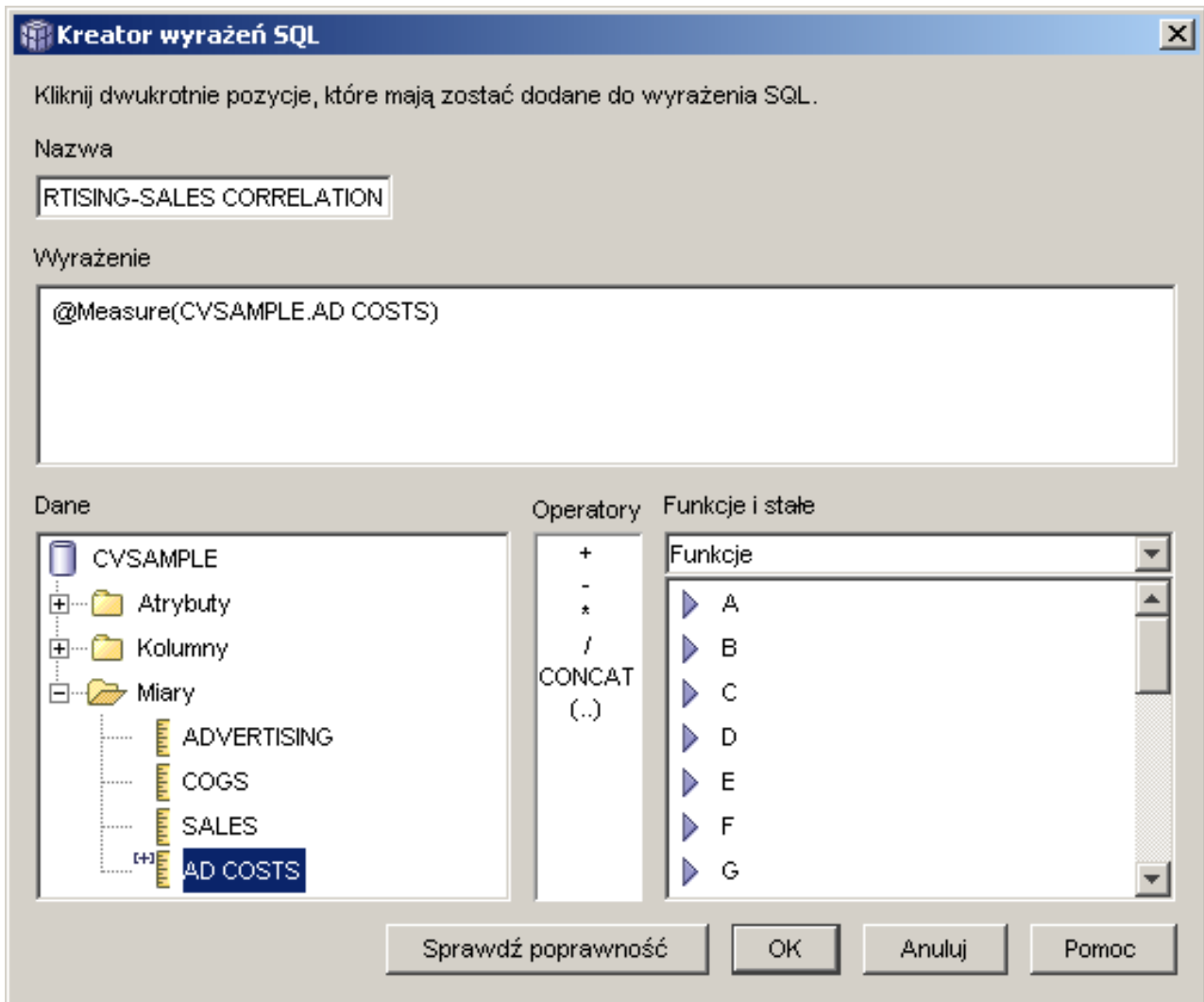
W bazie danych używanej przez dealera istnieje tabela faktów z kolumnami Sprzedaż i Koszty reklamy. Baza danych zawiera także kilka tabel wymiarów. Administrator może utworzyć miarę korzystając z funkcji DB2 CORRELATION do przeprowadzenia obliczeń badających poziom korelacji między wysokością kosztów i sprzedażą. Funkcja CORRELATION wymaga podania dwóch parametrów wejściowych. W tym przypadku jako parametrów administrator użył kolumn Sprzedaż i Koszty reklamy.

Wieloparametrowa funkcja agregująca musi wystąpić w skrypcie agregacji jako pierwsza. Funkcję wieloparametrową można zastosować po wszystkich wymiarach lub najpierw dla wszystkich wymiarów oprócz wymiaru czasu, po czym wymiar czasu zagregować inną funkcją, na przykład MAX. W wyrażeniu SQL odpowiedniej miary będzie istniało bezpośrednie odwzorowanie na kolumnę Koszty reklamy. Wyrażenie SQL będzie pierwszym z dwóch parametrów funkcji wieloparametrowej. Administrator zdefiniuje drugi parametr jako wyrażenie SQL odwzorowane bezpośrednio na kolumnę Sprzedaż. Funkcja CORRELATION jest zdefiniowana jako jedyna funkcja agregacji, przez co miara może obliczyć korelację statystyczną między kosztami reklamy a wynikami sprzedaży we wszystkich wymiarach.

## Czynności wymagane do utworzenia miary

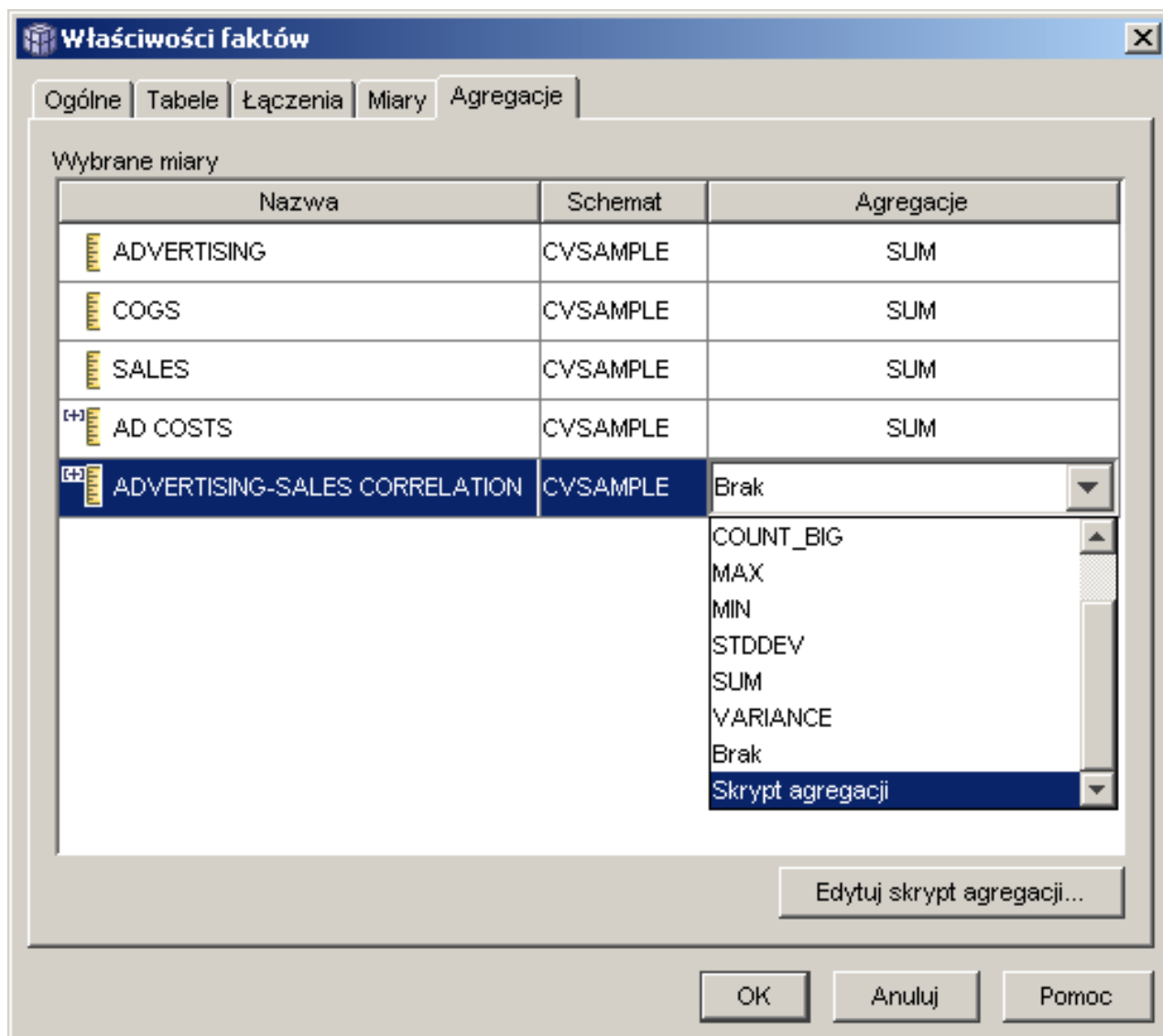
Poniższa procedura opisuje tworzenie miary Korelacja reklama-sprzedaż w istniejącym obiekcie faktów przy użyciu okna Właściwości faktów w Centrum OLAP:

1. Otwórz okno Właściwości faktów, klikając prawym przyciskiem myszy obiekt w drzewie obiektów Centrum OLAP i klikając polecenie **Edytuj miary**.
2. Kliknij przycisk **Utwórz miarę obliczaną**. Otwarte zostanie okno Kreatora wyrażen SQL.
3. W oknie Kreatora wyrażen SQL wpisz KORELACJA REKLAMA-SPRZEDAŻ w polu **Nazwa**.
4. Zdefiniuj wyrażenie miary, które posłuży także jako pierwszy parametr funkcji wieloparametrowej CORRELATION w skrypcie agregacji. Aby zdefiniować wyrażenie, rozwiń folder **Miary** na liście **Dane** i kliknij dwukrotnie miarę **KOSZTY REKLAMY**, aby ją dodać do listy **Wyrażenie**. Rys. 14 na stronie 66 przedstawia wyrażenie tworzone w oknie Kreatora wyrażen SQL.



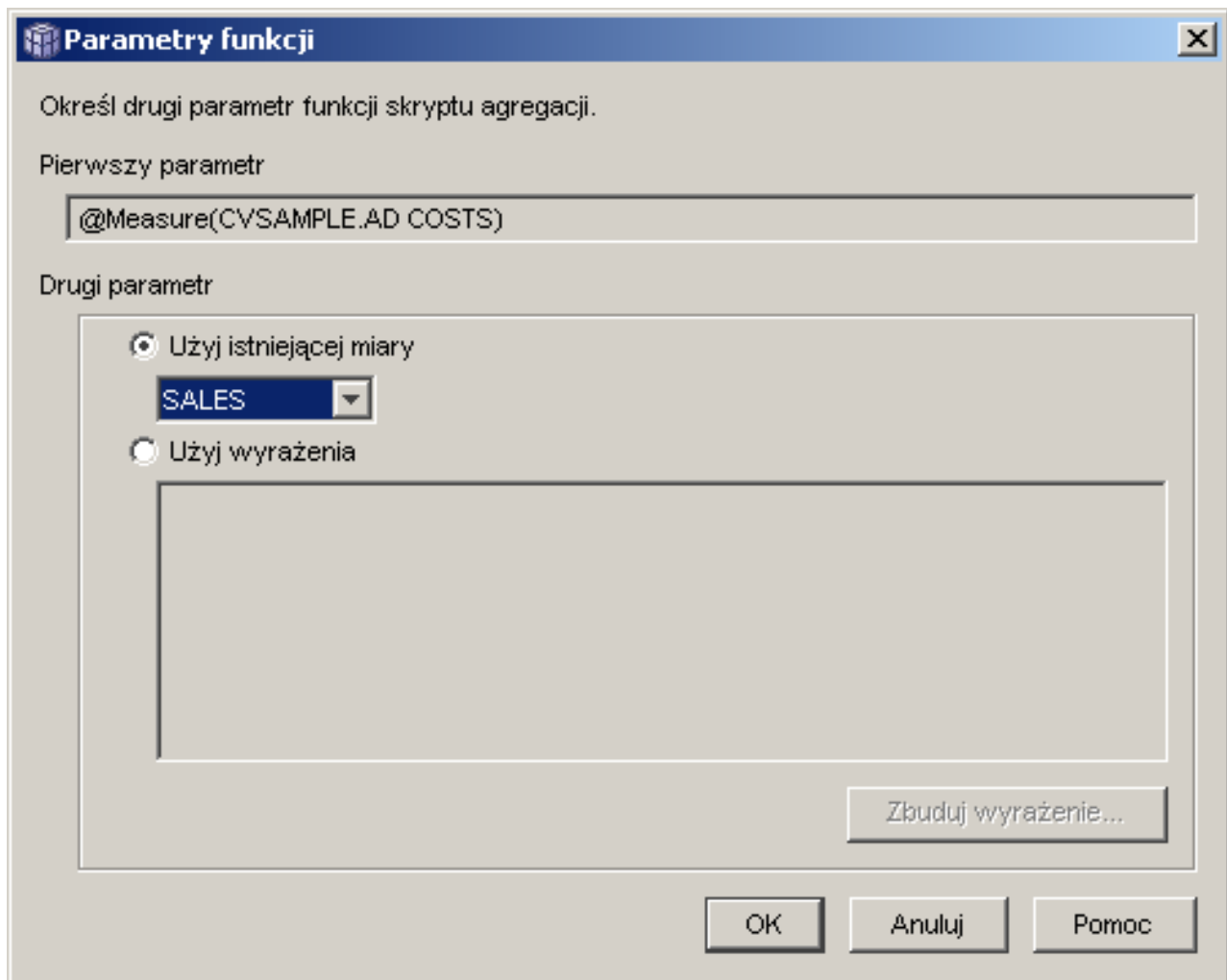
Rysunek 14. Gotowe wyrażenie korelujące koszty reklamy ze sprzedażą w oknie Kreatora wyrażen SQL

5. Na stronie Agregacje kliknij funkcję agregującą dla miary **KORELACJA REKLAMA-SPRZEDAŻ** i wybierz opcję **Skrypt agregacji**, jak to przedstawia Rys. 15 na stronie 67. Otwarte zostanie okno Kreatora skryptów agregacji.



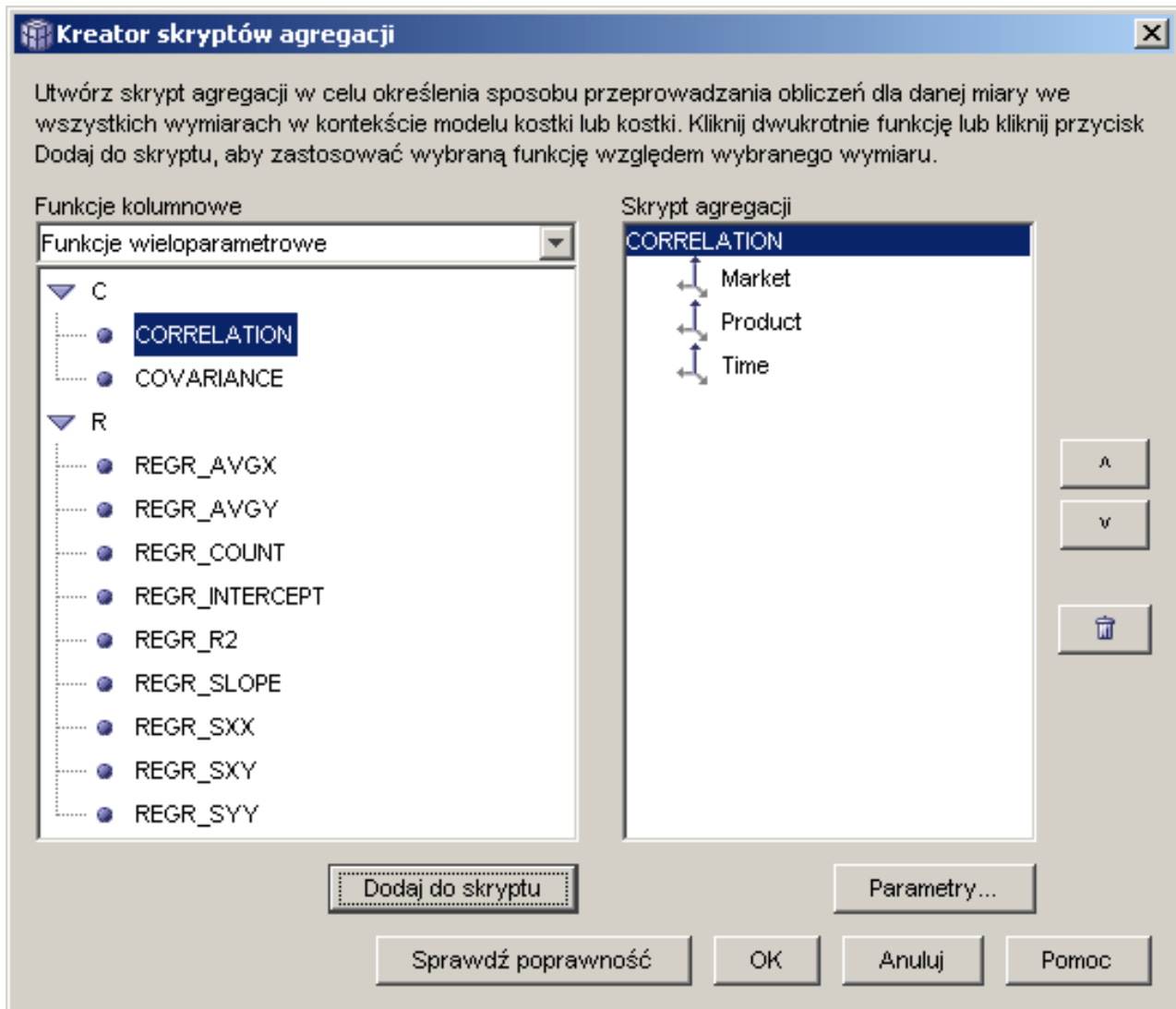
Rysunek 15. Strona Agregacje w oknie Właściwości faktów

6. W polu **Funkcje kolumnowe** wybierz opcję **Funkcje wieloparametrowe**. Na liście funkcji wieloparametrowych wybierz funkcję **CORRELATION** i kliknij przycisk **Dodaj do skryptu**. Otwarte zostanie okno Parametry funkcji.
7. Wybierz opcję **Używając istniejącej miary** i wybierz miarę **SPRZEDAŻ** z listy, jak to przedstawia Rys. 16 na stronie 68.



Rysunek 16. Miara Sprzedaż wybrana jako drugi parametr w oknie Parametry funkcji

8. Kliknij przycisk **OK**, aby zapisać wybrane opcje i zamknąć okno Parametry funkcji.
9. Rys. 17 na stronie 69 przedstawia funkcję CORRELATION na początku listy wymiarów w skrypcie.



Rysunek 17. Skrypt agregacji dla miary Korelacja reklama-sprzedaż

10. W oknie Kreatora skryptów agregacji kliknij przycisk **Sprawdź poprawność**, aby upewnić się, że skrypt agregacji jest poprawny.
11. Kliknij przycisk **OK**, aby zapisać skrypt agregacji i zamknąć okno.
12. Kliknij przycisk **OK**, aby zapisać zmiany w obiekcie faktów i zamknąć okno Właściwości faktów.

Utworzona została miara korelująca dwa rodzaje informacji w bazie danych. Na jej podstawie łatwiej będzie podejmować decyzje o wydatkach na reklamę w przyszłości, analizując tendencje zmian dotychczasowych.

## Obliczanie zysku i marży sklepu

Kierownik sklepu z zabawkami chce mieć możliwość analizowania wpływu różnych czynników, takich jak pora roku i typ towaru, na zysk i marżę.

Zanim można będzie przystąpić do zaawansowanej analizy, administrator bazy danych musi utworzyć miary zysku i marży. W następnej kolejności administrator utworzy dodatkowe miary, które pozwolą skorelować rozmaite czynniki z zyskiem i marżą oraz porównywać je ze sobą.

### Szczegóły scenariusza

Baza danych sklepu zawiera tabelę faktów, w której znajdują się kolumny Sprzedaż, Koszt sprzedanych towarów (KST) i Wydatki, a oprócz tego kolumny klucza obcego dla każdej z siedmiu tabel wymiarów. Administrator utworzył już miary o nazwach Sprzedaż, Koszt sprzedanych towarów i Wydatki, które są odwzorowane odpowiednio do kolumn Sprzedaż, Koszt sprzedanych towarów i Wydatki. Miary Zysk i Marża można zdefiniować wyłącznie w oparciu o te istniejące już miary.

Aby uzyskać miarę Zysk, administrator w oknie Wyrażenie SQL tworzy miarę obliczającą wielkość  $SPRZEDAŻ - (KST + WYDATKI)$ , po czym sumuje wyniki obliczeń po wszystkich wymiarach. Miarę Zysk można utworzyć, odwołując się do istniejących miar, kolumn lub ich kombinacji.

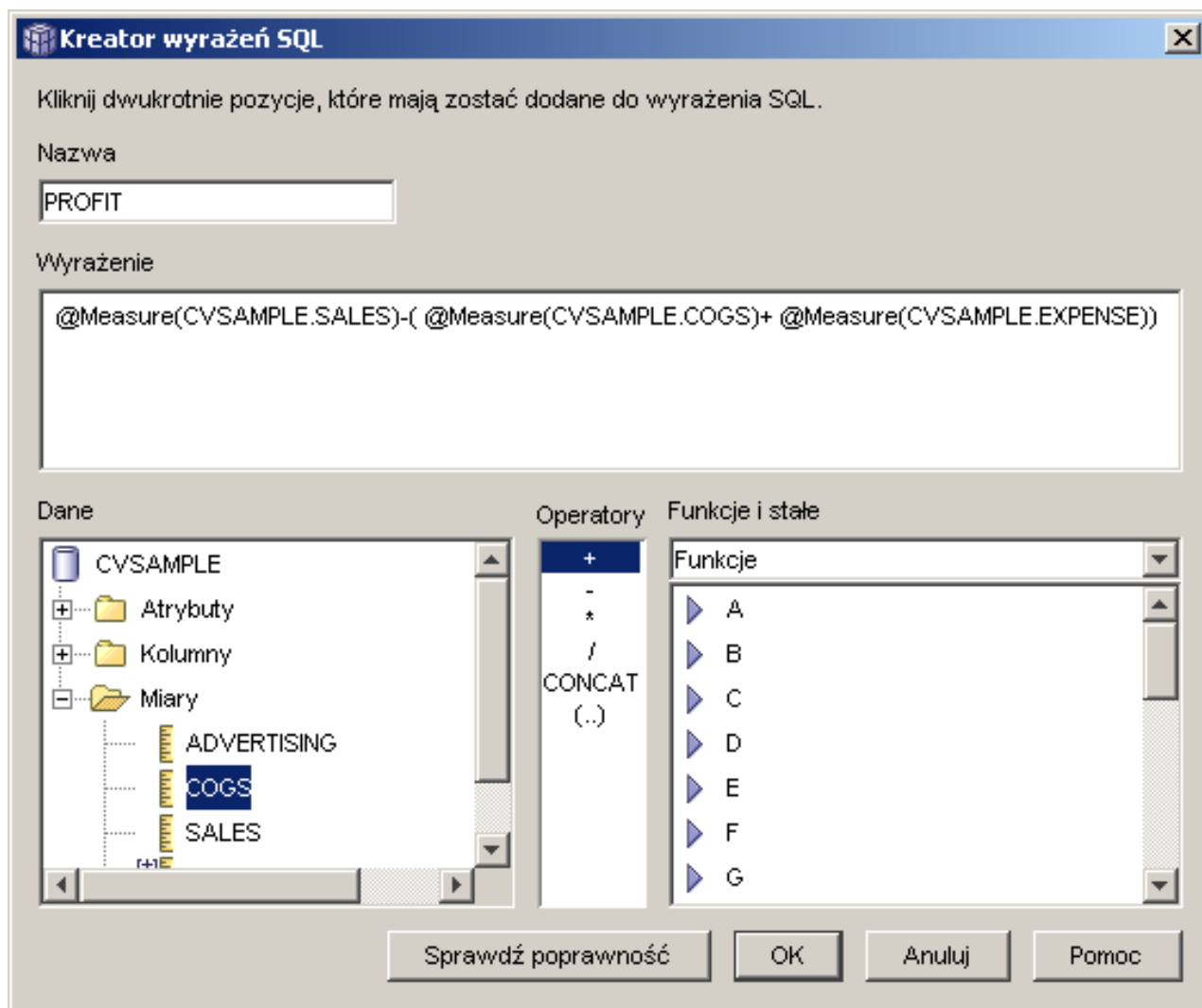
Po utworzeniu miary Zysk administrator może utworzyć miarę Marża. Miara Marża stanowi procentowy stosunek dwóch istniejących miar,  $(Zysk / Sprzedaż) * 100$ , i nie wymaga własnej funkcji agregującej. Funkcja agregująca nie jest wymagana, ponieważ miara odwołuje się tylko do innych miar, których dane są już zagregowane. Gdy administrator korzysta z miary złożonej, czyli takiej, która odwołuje się tylko do innych miar, obliczając ich stosunek, nie ma potrzeby definiowania dodatkowej agregacji. Większość funkcji agregujących, takich jak SUM, nie daje sensownych wyników w odniesieniu do wartości będących ilorazami. Na przykład jeśli sklep z zabawkami osiągał w czterech kolejnych kwartałach marże na poziomie 40%, 32%, 28% i 37%, sumowanie tych wartości w wymiarze czasu dałoby wynik 137% w skali roku, czyli wartość pozbawioną sensu.

### Czynności wymagane do utworzenia miar

Poniższa procedura opisuje sposób użycia okna Właściwości faktów w Centrum OLAP do utworzenia miar Zysk i Marża w istniejącym obiekcie faktów.

1. Aby otworzyć okno Właściwości faktów, kliknij prawym przyciskiem myszy obiekt faktów w drzewie obiektów Centrum OLAP, a następnie kliknij przycisk **Edytuj miary**. Otwarte zostanie okno Właściwości faktów.
2. Utwórz miarę Zysk:
  - a. Na stronie Miary kliknij przycisk **Utwórz miarę obliczaną**. Otwarte zostanie okno Kreatora wyrażen SQL.
  - b. W oknie Kreatora wyrażen SQL wpisz ZYSK w polu **Nazwa**.
  - c. Aby zdefiniować wyrażenie obliczające zysk, rozwiń folder **Miary** na liście **Dane** i wykonaj poniższe czynności:
    - Kliknij dwukrotnie miarę **SPRZEDAŻ** na liście **Dane**, aby dodać ją do wyrażenia.
    - Kliknij dwukrotnie operator - na liście **Operatory**.
    - Kliknij dwukrotnie miarę **KST** na liście **Dane**.
    - Kliknij dwukrotnie operator **+** na liście **Operatory**.
    - Kliknij dwukrotnie miarę **WYDATKI** na liście **Dane**.
    - W polu **Wyrażenie** podświetl część wyrażenia o treści `@Measure(CVSAMPLE.KST)+@Measure(CVSAMPLE.WYDATKI)` i kliknij dwukrotnie operator **(..)** na liście **Operatory**, aby zamknąć zaznaczoną część wyrażenia w cudzysłowach.

Rys. 18 przedstawia wyrażenie obliczające zysk utworzone w oknie Kreatora wyrażen SQL.

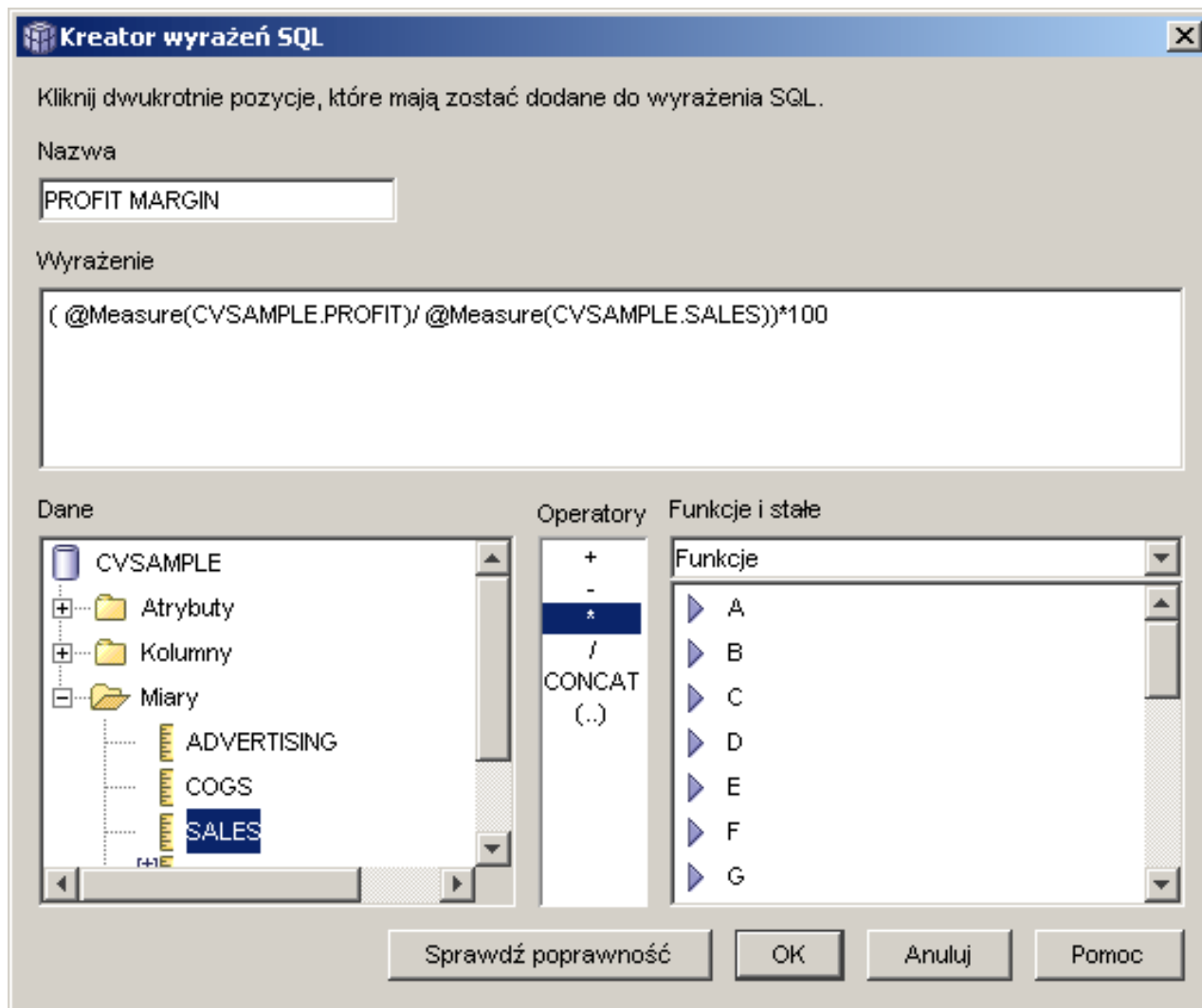


Rysunek 18. Gotowe wyrażenie obliczające zysk w oknie Kreatora wyrażen SQL

- d. Kliknij przycisk **OK**, aby utworzyć miarę Zysk i zamknąć okno Kreatora wyrażen SQL.
  - e. Na stronie Agregacje kliknij agregację dla miary Zysk i wybierz funkcję SUM. Miara Zysk jest gotowa.
3. Utwórz miarę Marża:
- a. Na stronie Miary kliknij opcję **Utwórz miarę obliczaną**. Otwarte zostanie okno Kreatora wyrażen SQL.
  - b. W oknie Kreatora wyrażen SQL wpisz **MARŻA** w polu **Nazwa**.
  - c. Aby zdefiniować wyrażenie obliczające marżę, rozwiń folder **Miary** na liście **Dane** i wykonaj poniższe czynności:
    - Kliknij dwukrotnie miarę **ZYSK** na liście **Dane**, aby dodać ją do wyrażenia.
    - Kliknij dwukrotnie operator **/** na liście **Operatory**.
    - Kliknij dwukrotnie miarę **SPRZEDAŻ** na liście **Dane**.
    - Zamknij całe wyrażenie w nawiasach, wpisując je w polu **Wyrażenie**.

- Ustaw kursor na końcu wyrażenia i kliknij dwukrotnie operator \* na liście **Operatory**.
- Dopisz liczbę 100 na końcu wyrażenia w polu **Wyrażenie**.

Rys. 19 przedstawia wyrażenie obliczające marżę utworzone w oknie Kreatora wyrażen SQL.



Rysunek 19. Gotowe wyrażenie obliczające marżę w oknie Kreatora wyrażen SQL

- d. Kliknij przycisk **OK**, aby utworzyć miarę Marża i zamknąć okno Kreatora wyrażen SQL.

Na stronie Agregacje Centrum OLAP dla miar złożonych domyślnie wybierany jest brak funkcji agregującej, więc nie ma potrzeby wprowadzania tu zmian.

4. Kliknij przycisk **OK**, aby zamknąć okno Właściwości faktów i zapisać obie nowe miary dodane do obiektu faktów.

Gdy obie te miary są już gotowe, można na ich podstawie wykonywać dodatkowe analizy.



## Zliczanie zamówień internetowych

Pewne przedsiębiorstwo zajmujące się handlem detalicznym kilka lat temu poszerzyło zakres działalności o sprzedaż przez Internet. Obecnie jego kierownictwo pragnie przeanalizować efektywność internetowego kanału sprzedaży. Jednym z pierwszych wskaźników, jakie są potrzebne w takiej analizie, jest liczba zamówień zrealizowanych za pośrednictwem Internetu.

### Szczegóły scenariusza

W bazie danych przedsiębiorstwa znajduje się tabela faktów dla zamówień internetowych, zawierająca kolumny ID\_ZAMOWIENIA, ID\_PRODUKTU, ILOSC i ID\_CZASU. W kolumnie ID\_PRODUKTU zapisywany jest każdy produkt sprzedany w ramach danego zamówienia, a w kolumnie ILOSC figurują odpowiednie ilości sprzedanych produktów. Zamówienia zawierające więcej niż jeden produkt zapisywane są w wielu wierszach, których liczba odpowiada liczbie odrębnych produktów w zamówieniu. Na przykład Tabela 33 zawiera informacje o trzech zamówieniach, z których Zamówienie 1 opiewało na trzy produkty A, jeden produkt O i jeden produkt G.

Tabela 33. Zawartość części tabeli faktów

ID_ZAMOWIENIA	ID_PRODUKTU	ILOSC
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

Administrator bazy danych może utworzyć miarę Liczba zamówień, która będzie zliczać każdą niepowtarzalną wartość w kolumnie ID\_ZAMOWIENIA. W kodzie SQL definiującym miarę Liczba zamówień musi wystąpić parametr DISTINCT oraz funkcja COUNT realizująca agregację po wszystkich wymiarach. Wyrażenie SQL tej miary utworzy listę odrębnych zamówień, uzyskaną przez zliczanie w wyniku agregacji. Jako że miara nie realizuje sumowania, nazywana jest miarą nieaddytywną.

Miary nieaddytywne są użyteczne także w sytuacji, gdy zachodzi potrzeba zliczania danych tekstowych, znakowych lub innego typu. Przykładem będzie użycie miary nieaddytywnej do policzenia kodów pocztowych, pod które realizowane były dostawy.

W tym przykładzie administrator postanowił zdefiniować miarę Identyfikator zamówienia, bezpośrednio odwzorowaną do kolumny ID\_ZAMOWIENIA. Można jednak użyć samej kolumny ID\_ZAMOWIENIA w identyczny sposób. Domyślna agregacja jest dobierana w zależności od tego, czy dana kolumna lub miara jest wykorzystywana w wyrażeniu SQL, jednak w obu przypadkach należy zmienić domyślną agregację na COUNT, jak to opisano w punkcie “Czynności wymagane do utworzenia miar”.

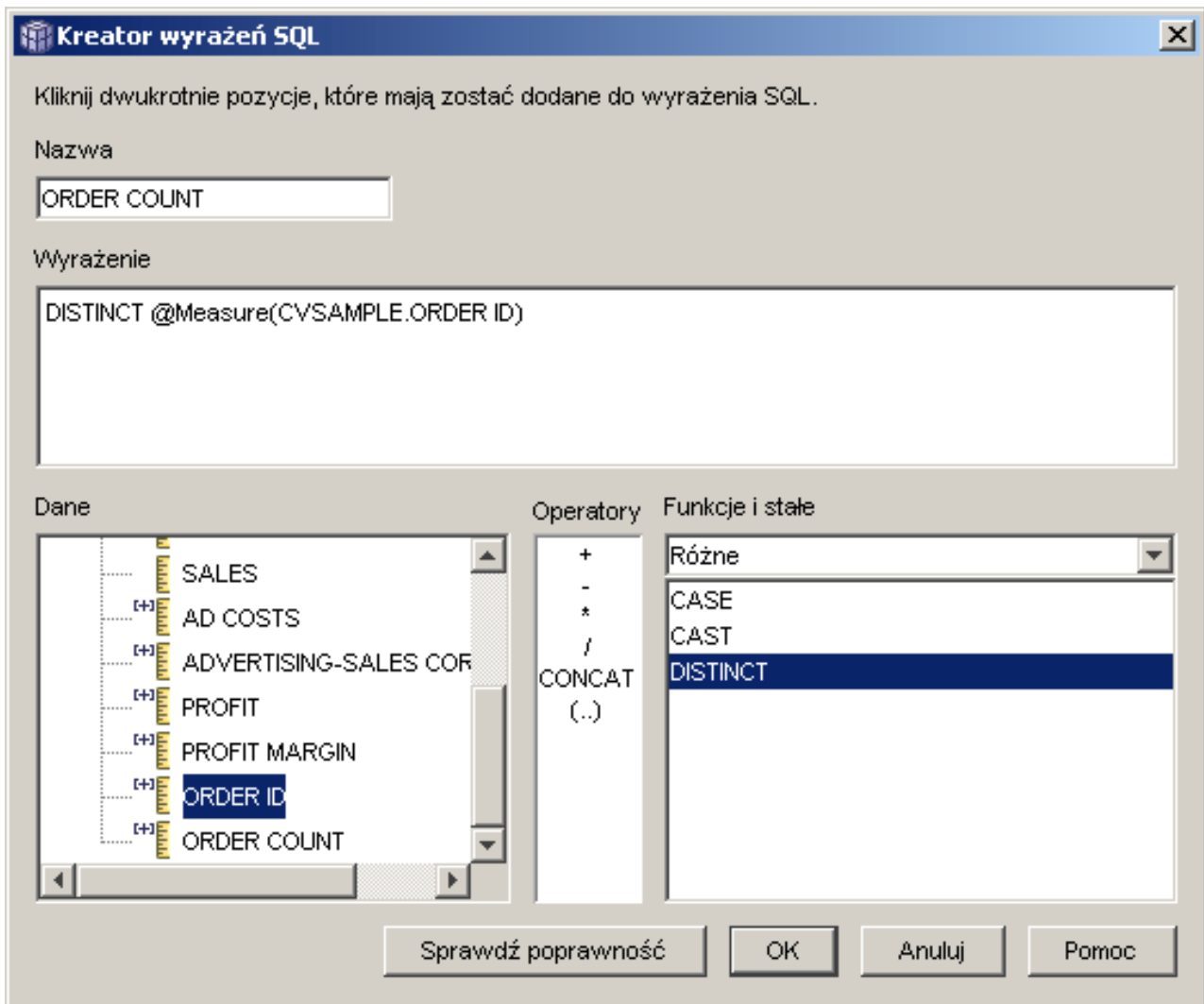
### Czynności wymagane do utworzenia miar

Poniższa procedura opisuje tworzenie miary Liczba zamówień w istniejącym obiekcie faktów przy użyciu okna Właściwości faktów w Centrum OLAP.

1. Aby otworzyć okno Właściwości faktów, kliknij prawym przyciskiem myszy obiekt faktów w drzewie obiektów Centrum OLAP, a następnie kliknij przycisk **Edytuj miary**. Otwarte zostanie okno Właściwości faktów.

2. Na stronie Miary kliknij przycisk **Utwórz miarę obliczaną**. Otwarte zostanie okno Kreatora wyrażeń SQL.
3. W oknie Kreatora wyrażeń SQL wpisz LICZBA ZAMÓWIEŃ w polu **Nazwa**.
4. Aby zdefiniować wyrażenie zliczające zamówienia, rozwin folder **Miary** na liście **Dane** i wykonaj poniższe czynności:
  - W polu **Funkcje i stałe** wybierz pozycję **Różne**. Na liście różnych funkcji i stałych kliknij dwukrotnie parametr **DISTINCT**.
  - Kliknij dwukrotnie miarę **ID ZAMÓWIENIA** na liście **Dane**.

Rys. 20 przedstawia wyrażenie zliczające zamówienia utworzone w oknie Kreatora wyrażeń SQL.



Rysunek 20. Kompletne wyrażenie zliczające zamówienia w oknie Kreatora wyrażeń SQL

5. Kliknij przycisk **OK**, aby zamknąć okno Kreatora wyrażeń SQL.
6. Na stronie Agregacje kliknij agregację dla miary Liczba zamówień i wybierz funkcję **COUNT**.
7. Kliknij przycisk **OK**, aby zapisać zmiany w obiekcie faktów i zamknąć okno Właściwości faktów.

Wynikiem powyższych operacji jest miara zliczająca liczbę odrębnych wartości w wierszu Identyfikator zamówienia. Miary tej można używać w połączeniu z innymi miarami przy prowadzeniu dalszych analiz danych.

---

## Ranking wysokości obrotów

Pewne przedsiębiorstwo w ciągu ostatnich kilku lat intensywnie rozbudowywało swoją sieć sklepów z artykułami biurowymi. Obecnie kierownictwo rozważa zamknięcie najmniej aktywnych sklepów, aby obniżyć koszty i zwiększyć zyski.

Historia realizowanych obrotów jest ważnym czynnikiem przy podejmowaniu decyzji o zamknięciu danego sklepu. Analitycy muszą mieć możliwość uszeregowania placówek pod względem wysokości obrotów i drążenia danych w celu dokonania dokładniejszej analizy.

### Szczegóły scenariusza

W bazie danych sklepów biurowych znajduje się tabela faktów zawierająca między innymi kolumnę Sprzedaż. Baza danych zawiera także kilka tabel wymiarów. Administrator może utworzyć miarę Ranking sprzedaży na bazie funkcji RANK. Jest to specjalna funkcja OLAP dostarczana z programem DB2 Universal Database (DB2 UDB).

Program DB2 Cube Views obsługuje następujące funkcje OLAP udostępniane przez program DB2 UDB:

#### RANK

Porządkuje wiersze i każdemu z nich przypisuje ocenę. Ocena jest definiowana jako 1 plus liczba poprzedzających wierszy o odrębnych pozycjach. Jeśli nie da się ustalić wzajemnej kolejności niektórych wierszy ze względu na występowanie w nich powielonych wartości, wierszom tym przypisywana jest jednakowa wartość oceny. W wynikach mogą występować przerwy w numeracji spowodowane powieleniami wartości w wierszach. Tabela 34 na stronie 76 przedstawia przykład wyników funkcji RANK dla zestawu przykładowych wartości wierszy.

Typowa składnia funkcji RANK wygląda następująco:

```
RANK ( ) OVER (ORDER BY  
wyrażenie-klucza-sortowania  
kolejność-wyrażenia)
```

gdzie *wyrażenie-klucza-sortowania* jest zestawem danych do uporządkowania, a *kolejność-wyrażenia* to parametr **ASC** lub **DESC**, powodujący ułożenie wartości w porządku rosnącym lub malejącym. Program DB2 Cube Views wymaga, by wyrażenie *wyrażenie-klucza-sortowania* było istniejącą miarą, a nie kolumną ani atrybutem. Ponadto program DB2 Cube Views nie obsługuje klauzuli PARTITION BY, której użycie w tej funkcji umożliwia program DB2 UDB. Więcej informacji o funkcji RANK zawiera Centrum informacyjne DB2 UDB.

#### DENSERANK

Porządkuje wiersze i każdemu z nich przypisuje ocenę. Ocena jest obliczana jako 1 plus liczba wszystkich wierszy poprzedzających danych wiersz. Dzięki temu otrzymana zostanie lista uporządkowana i bez przerw w numeracji. Tabela 34 na stronie 76 przedstawia przykład wyników funkcji DENSERANK dla zestawu przykładowych wartości wiersza.

Typowa składnia funkcji DENSERANK wygląda następująco:

```
DENSERANK ( ) OVER (ORDER BY  
wyrażenie-klucza-sortowania  
kolejność-wyrażenia)
```

gdzie *wyrażenie-kłucza-sortowania* jest zestawem danych do uporządkowania, a *kolejność-wyrażenia* to parametr **ASC** lub **DESC**, powodujący ułożenie wartości w porządku rosnącym lub malejącym. Program DB2 Cube Views wymaga, by wyrażenie *wyrażenie-kłucza-sortowania* było istniejącą miarą, a nie kolumną ani atrybutem. Ponadto program DB2 Cube Views nie obsługuje klauzuli **PARTITION BY**, której użycie w tej funkcji umożliwia program DB2 UDB. Więcej informacji o funkcji **DENSERANK** zawiera Centrum informacyjne DB2 UDB.

## ROWNUMBER

Oblicza kolejny numer wiersza w ustalonym porządku, począwszy od numeru 1 dla wiersza pierwszego. Jeśli nie zostanie użyta klauzula **ORDER BY**, numery są przypisywane wierszom w sposób przypadkowy.

Typowa składnia funkcji **ROWNUMBER** wygląda następująco:

```
ROWNUMBER ( ) OVER
([ORDER BY wyrażenie-kłucza-sortowania
kolejność-wyrażenia])
```

gdzie *wyrażenie-kłucza-sortowania* jest zestawem danych do uporządkowania, a *kolejność-wyrażenia* to parametr **ASC** lub **DESC**, powodujący ułożenie wartości w porządku rosnącym lub malejącym. Program DB2 Cube Views wymaga, aby źródłem danych tej funkcji była istniejąca miara, a nie kolumna ani atrybut. Ponadto program DB2 Cube Views nie obsługuje klauzuli **PARTITION BY**, której użycie w tej funkcji umożliwia program DB2 UDB. Więcej informacji o funkcji **ROWNUMBER** zawiera Centrum informacyjne DB2 UDB.

Wymienione tu funkcje OLAP nie figurują na liście funkcji i stałych w Kreatorze wyrażeń SQL.

Tabela 34. Wyniki rankingu uzyskane po zastosowaniu funkcji **RANK** i **DENSERANK** dla przykładowego zestawu wartości

Wartości wiersza	Kolejność	Wyniki rankingu z funkcji <b>RANK</b>	Wyniki rankingu z funkcji <b>DENSERANK</b>
100	1	1	1
35	2	2	2
23	3	3	3
8	4	4	4
8	4	4	5
6	5	6	6

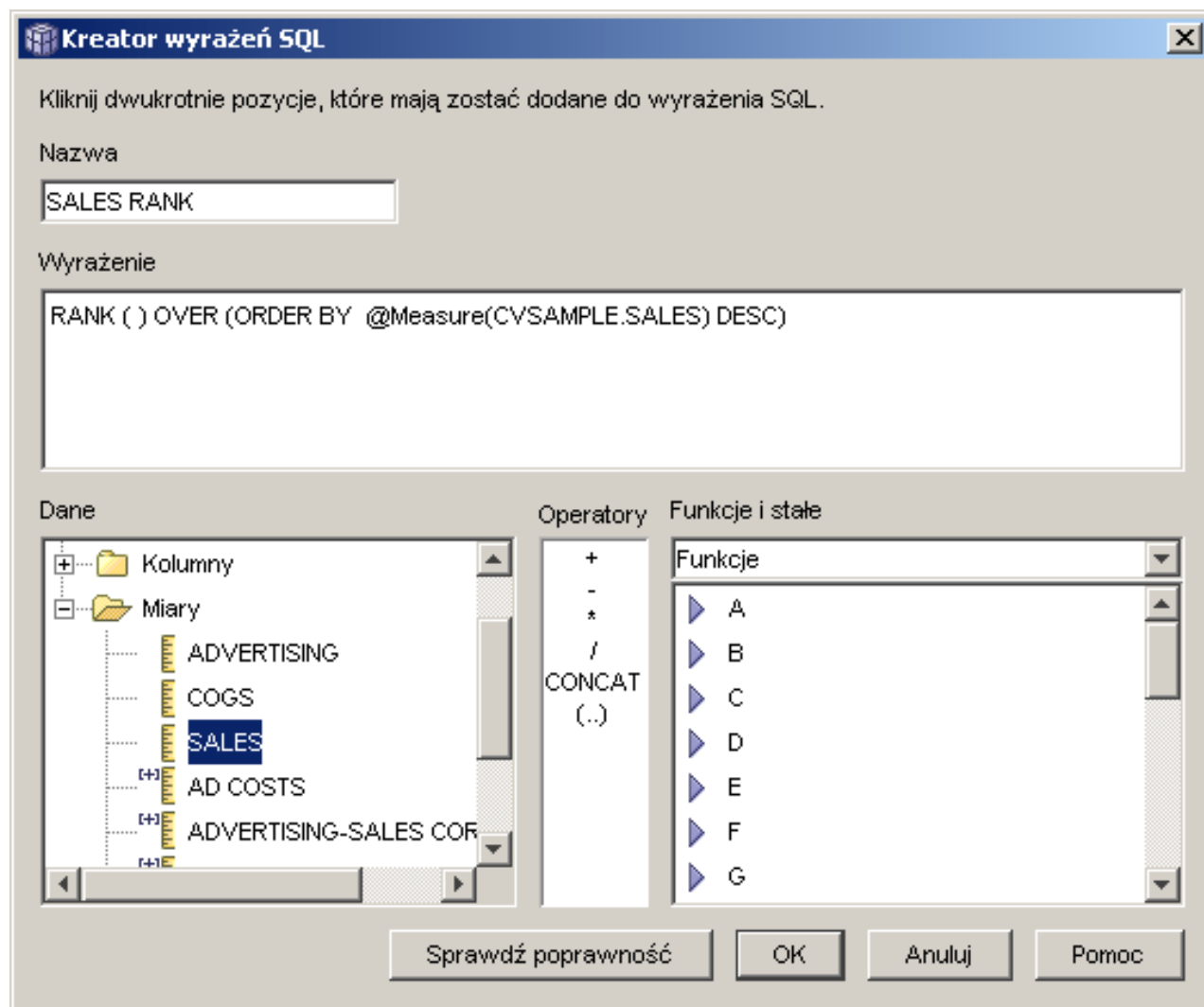
## Czynności wymagane do utworzenia miary

Poniższa procedura opisuje tworzenie miary Ranking sprzedaży w istniejącym obiekcie faktów przy użyciu okna Właściwości faktów w Centrum OLAP:

1. Aby otworzyć okno Właściwości faktów, kliknij prawym przyciskiem myszy obiekt faktów w drzewie obiektów Centrum OLAP, a następnie kliknij przycisk **Edytuj miary**. Otwarte zostanie okno Właściwości faktów.
2. Na stronie Miary kliknij opcję **Utwórz miarę obliczaną**, aby utworzyć miarę Ranking sprzedaży. Otwarte zostanie okno Kreatora wyrażeń SQL.
3. W oknie Kreatora wyrażeń SQL wpisz **RANKING SPRZEDAŻY** w polu **Nazwa**.
4. Wykonaj poniższe czynności w celu utworzenia wyrażenia miary Ranking sprzedaży:
  - W polu Wyrażenie wpisz następujący kod funkcji: **RANK ( ) OVER (ORDER BY measure DESC)**.

- Rozwiń folder **Miary** na liście **Dane**.
- Podświetl wyraz miara w składni funkcji w polu **Wyrażenie** i kliknij dwukrotnie miarę **SPRZEDAŻ**, aby dodać ją do wyrażenia.

Wyrażenie w ostatecznej postaci przedstawia Rys. 21.



Rysunek 21. Gotowe wyrażenie miary Ranking sprzedaży w oknie Kreatora wyrażen SQL

5. Kliknij przycisk **Sprawdź poprawność**, aby zyskać pewność, że wyrażenie jest poprawne. Kliknij przycisk **OK**, aby zamknąć okno Kreatora wyrażen SQL.

Na stronie Agregacje należy zmienić domyślną funkcję agregującą, którą jest Brak. Opcja Brak jest wybierana domyślnie dla miary Ranking sprzedaży, ponieważ źródło danych jest liczbowe i odwołuje się tylko do istniejących miar.

Korzystając z funkcji RANK w wyrażeniu miary do uporządkowania kolumny Sprzedaż w kolejności malejącej, analityk może schodzić do bardziej szczegółowych warstw danych w dowolnych innych wymiarach, na przykład w celu ustalenia placówki o najgorszych wynikach sprzedaży w minionym roku dla wybranej linii produktów.

---

## Użycie danych czasowych w tabeli faktów do utworzenia wymiaru czasu

Przedsiębiorstwo zajmujące się handlem detalicznym, XYZ Retail, modeluje dane ze swoich transakcji sprzedaży w programie DB2 Cube Views, aby skorzystać z oferowanych przezeń funkcji analitycznych. Jednak z uwagi na transakcyjny charakter danych, jedyną dostępną informacją czasowa to data skojarzona z odpowiednią transakcją.

Informacje o czasie modelowane w wymiarze czasu są niezbędne jako kontekst dla wielu typowych obliczeń, takich jak analizy tendencji w sprzedaży według kwartału i obliczanie średniej wartości zapasów w poszczególnych tygodniach.

Wielu administratorów unika przechowywania danych na temat czasu w postaci daty lub datownika transakcji, ponieważ brak transakcji za pewien dzień powoduje powstawanie przerw w ciągłości danych, co stwarza później problemy przy ich agregowaniu i wyświetlaniu. Zazwyczaj lepszym rozwiązaniem jest modelowanie danych czasowych w tabeli czasu. Jednak administrator bazy danych w firmie XYZ Retail jest przekonany, że każdego dnia rejestrowana będzie przynajmniej jedna transakcja i decyduje się pozostawić strukturę danych w obecnej postaci.

### Szczegóły scenariusza

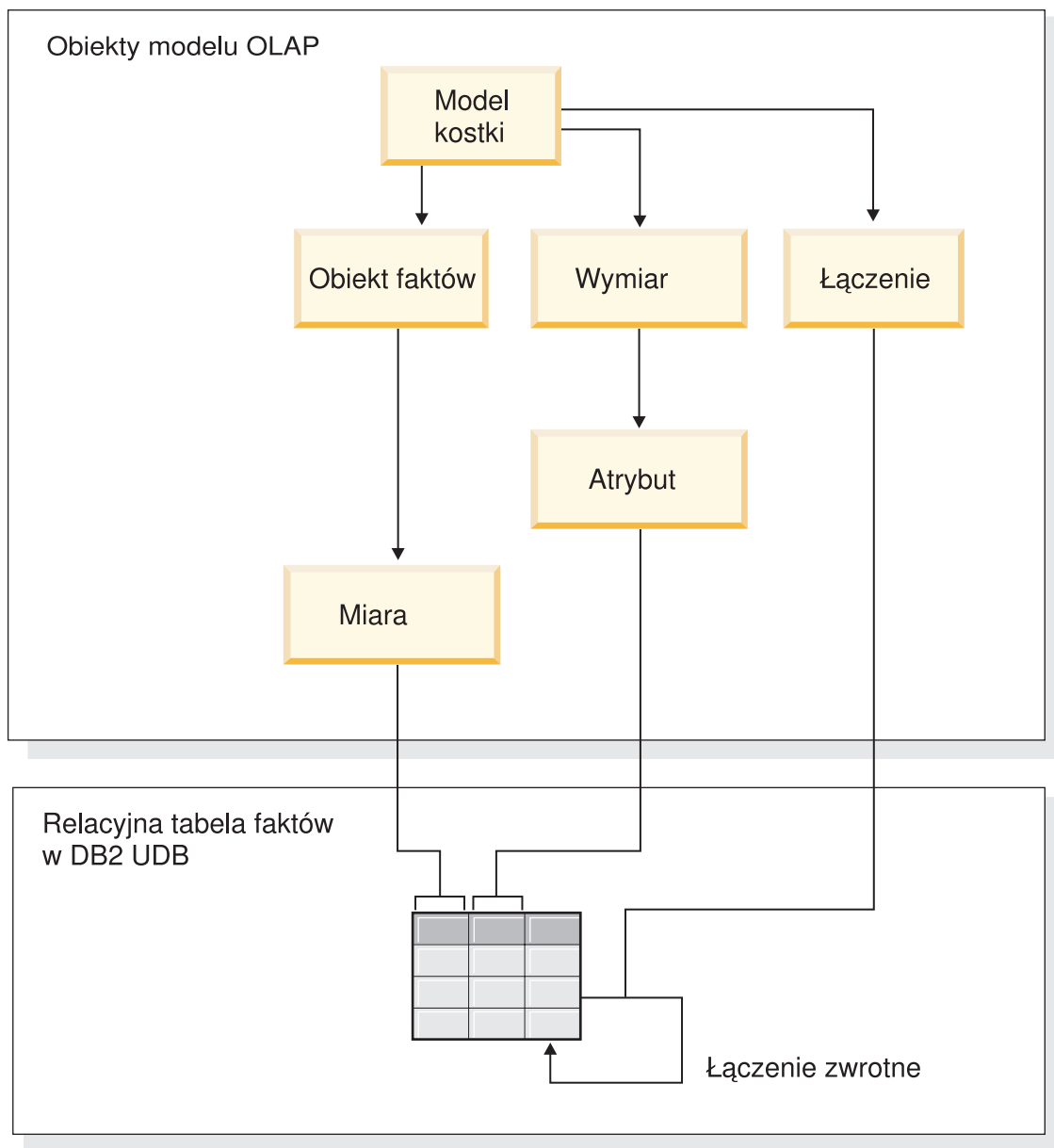
Firma XYZ Retail prowadzi tabelę faktów zawierającą mierzalne parametry każdej transakcji, takie jak Sprzedaż, Koszt, Ilość sprzedana i Data. Ponadto w bazie danych znajdują się tabele wymiarów Region oraz Produkt. Problemem jest to, że dane czasu przechowywane są w tabeli faktów, a nie w osobnej tabeli wymiarów. Administrator musi więc utworzyć obiekt wymiaru bazujący na informacjach o dacie znajdujących się w obiekcie faktów.

Tworząc wymiar czasu w oparciu o pojedynczą kolumnę dat w tabeli faktów, należy spełnić dwa szczególne wymagania:

- Ponieważ wszystkie obiekty wymiarów w poprawnym modelu kostki muszą być połączone z obiektem faktów, a zarówno obiekt wymiaru czasu, jak i obiekt faktów bazują na jednej i tej samej tabeli faktów, obiekt wymiaru czasu musi być połączony z obiektem faktów za pomocą łączenia zwrotnego, które połączy tabelę faktów z nią samą.
- Administrator musi zdefiniować atrybuty obliczane agregujące daty do poziomu przydatnego w analizie, takiego jak tydzień, miesiąc, kwartał lub rok.

Łączenie zwrotne jest to typ łączenia, w którym tabela jest łączona ze sobą. W tym przypadku będzie to tabela faktów. Łączenie zwrotne powinno łączyć jedną lub więcej kolumn, które wspólnie identyfikują jednoznacznie każdy wiersz w tabeli faktów. Najlepszą opcją będzie wybór klucza głównego. Jeżeli jednak klucz podstawowy nie jest zdefiniowany, dobrą alternatywą będzie zestaw kolumn używany do łączenia tabeli faktów z tabelami wymiarów. Aby zoptymalizować model kostki, trzeba zdefiniować klucz podstawowy. Liczność łączenia musi wynosić 1:1, a łączenie musi być typu wewnętrznego.

Rys. 22 na stronie 79 przedstawia sposób odwzorowania między obiektem faktów, wymiarem bazującym na tabeli faktów i łączeniem faktów z wymiarem a jedną i tą samą tabelą faktów.



Rysunek 22. Sposób łączenia tabeli z nią samą przy użyciu łączenia zwrotnego

### Czynności wymagane przy tworzeniu atrybutów i wymiaru

Przedstawiona procedura wyjaśnia sposób użycia Kreatora wymiarów w Centrum OLAP do utworzenia wymiaru Czas i atrybutów obliczanych w oparciu o tabelę faktów:

1. Aby otworzyć Kreatora wymiarów, kliknij dwukrotnie model kostki w drzewie obiektów Centrum OLAP i kliknij polecenie **Utwórz wymiar**. Otwarty zostanie Kreator wymiarów.
2. Na stronie Nazwa wpisz **Czas** w polu **Nazwa**. Opcjonalnie można zmienić nazwę biznesową i wpisać komentarz. Kliknij przycisk **Dalej**.
3. Wybierz tabelę faktów modelu kostki. Kliknij przycisk **Dalej**. Nie ma potrzeby określania łączy wymiarowych, ponieważ wymiar zawiera tylko jedną tabelę. Na stronie Łączenia wymiaru kliknij przycisk **Dalej**.
4. Na stronie Atrybuty wymiaru wybierz kolumnę **Datownik**.

5. Opcjonalnie: Utwórz dodatkowe atrybuty obliczane, które będą agregować datowniki w większe porcje, takie jak miesiąc, kwartał lub rok. Aby utworzyć atrybuty obliczane, kliknij przycisk **Utwórz atrybut obliczany**, aby otworzyć Kreatora wyrażeń SQL, a następnie zdefiniuj dla poszczególnych atrybutów wyrażenia obliczające miesiące, kwartały lub lata na podstawie źródłowych datowników. Po zdefiniowaniu poszczególnych atrybutów obliczanych kliknij przycisk **Sprawdź poprawność**, aby zweryfikować poprawność wyrażenia, a następnie kliknij przycisk **OK**, aby zamknąć Kreatora wyrażeń SQL i powrócić do Kreatora wymiarów. Kliknij przycisk **Dalej** po wybraniu i utworzeniu wszystkich wymaganych atrybutów.
6. Na karcie Typ wymiaru wybierz opcję **Czas**. Kliknij przycisk **Dalej**.
7. Na stronie Łączenie fakt-wymiar kliknij przycisk **Utwórz łączenie**. W otwartym oknie Kreatora łącheń utwórz łączenie zwrotne. Wpisz nazwę i kliknij przycisk **Dalej**. Wybierz kolumnę lub zestaw kolumn jednoznacznie identyfikujący każdy wiersz w tabeli faktów, na przykład klucz podstawowy, dla atrybutu lewostronnego i prawostronnego. Wybieraj po jednej parze i klikaj przycisk **Dodaj**, aby dodać parę atrybutów do łączenia. Jako typ łączenia wybierz łączenie wewnętrzne i licznosc 1:1. Po dodaniu wymaganych par atrybutów kliknij przycisk **Zakończ**. Okno Kreatora łącheń zostanie zamknięte.
8. Na stronie Łączenie fakt-wymiar kliknij przycisk **Zakończ**.

Dzięki zdefiniowaniu wymiaru czasu w modelu kostki firma XYZ Retail może korzystać z całkiem nowej warstwy znaczeniowej w swych analizach danych. Możliwe będzie teraz wykonywanie analiz związanych z czasem, na przykład dotyczących stanu zapasów.



---

## Rozdział 5. Optymalizacja modelu kostki programu DB2 Cube Views

W tej sekcji omówiono następujące tematy:

### **Tabele podsumowań**

W programie DB2 Cube Views tabele podsumowań DB2 służą do poprawiania wydajności zapytań kierowanych do modeli kostek. Tabela podsumowań to zmaterializowana tabela zapytania (MQT) szczególnego typu zawierająca dane podsumowań.

### **Tabele podsumowań z zależnościami funkcjonalnymi i ograniczeniami**

Doradca optymalizacyjny na podstawie informacji o związkach między danymi, na przykład o zależnościach funkcjonalnych i ograniczeniach, zaleca tabele podsumowań zawierające zagregowane miary oraz atrybuty poziomu niezbędne optymalizatorowi DB2 do udzielenia efektywnych odpowiedzi na zapytania.

### **Przegląd procesu optymalizacji**

Zoptymalizowanie schematu gwiazdowego lub schematu typu płatek śniegu za pomocą programu DB2 Cube Views może podnieść wydajność zapytań SQL w stylu OLAP. Proces optymalizacji obejmuje tworzenie, implementowanie i konserwowanie tabel podsumowań zalecanych przez Doradcę optymalizacyjnego.

### **Projekt metadanych a optymalizacja**

Sposób zaprojektowania obiektów metadanych, takich jak poziomy, hierarchie, miary, kostki i wycinki optymalizacji, wpływa na tabele podsumowań zalecane przez Doradcę optymalizacyjnego.

### **Wycinki optymalizacji dla kostek**

Wycinek optymalizacji to opcjonalna, ale przydatna funkcja wykorzystywana przez Doradcę optymalizacyjnego podczas przedstawiania tabel podsumowań koncentrujących się na najważniejszych regionach modelu kostki.

### **Analizowanie zapytań dla wycinków optymalizacji**

Wycinki optymalizacji stanowią silne narzędzie do poprawiania wydajności zapytań, lecz są efektywne tylko wtedy, gdy dokładnie odzwierciedlają zapytania użytkowników.

### **Definicje ograniczeń optymalizacyjnych**

Ograniczenia dostarczają cennych informacji Doradcy optymalizacyjnemu oraz optymalizatorowi DB2. Ograniczenia informacyjne lub ograniczenia wymuszane dla kluczy obcych i kluczy podstawowych muszą być zdefiniowane w schemacie gwiazdowym lub w schemacie typu płatek śniegu.

### **Parametry Doradcy optymalizacyjnego**

Informacje określone dla każdego parametru Doradcy optymalizacyjnego wpływają na tabele podsumowań zalecane przez kreatora oraz na uzyskiwaną poprawę wydajności. Należy podawać dokładne informacje oraz dokonywać rozważnego wyboru między kosztem a wymaganą wydajnością.

### **Optymalizowanie modelu kostki**

Optymalizując zapytania wykonywane na modelu kostki, można zwiększyć wydajność programów zgłaszających zapytania SQL typu OLAP.

### **Przykład skryptu SQL do tworzenia tabel podsumowań**

Doradca optymalizacyjny generuje skrypt SQL do tworzenia zalecanych tabel podsumowań. Skrypt ten zawiera instrukcje SQL niezbędne do zbudowania jednej lub wielu tabel podsumowań.

### **Testowanie wyników zapytania**

Korzystając z narzędzia do testów porównawczych, db2batch, dostępnego w programie DB2 Universal Database, można przy użyciu Doradcy optymalizacyjnego testować wydajność zapytań przed utworzeniem tabel podsumowań oraz po ich utworzeniu.

### **Rozwiązywanie problemów z tabelami podsumowań**

Jeśli wydajność zapytań nie zwiększa się po utworzeniu tabel podsumowań, w celu rozwiązania problemów z kierowaniem zapytań można się posłużyć narzędziem DB2EXPLAIN.

### **Aktualizacja tabel podsumowań**

Gdy dane w tabelach podstawowych ulegną zmianie, należy zaktualizować tabele podsumowań. Tabele podsumowań można aktualizować na dwa różne sposoby: wykonując odświeżanie natychmiastowe albo odświeżanie odroczone.

### **Usuwanie tabel podsumowań**

Gdy zostaje usunięty model kostki, program DB2 Cube Views nie usuwa skojarzonych z nim tabel podsumowań. Jeśli tabele podsumowań nie są już używane do żadnego innego celu, można je usunąć, aby zwolnić miejsce na dysku.

---

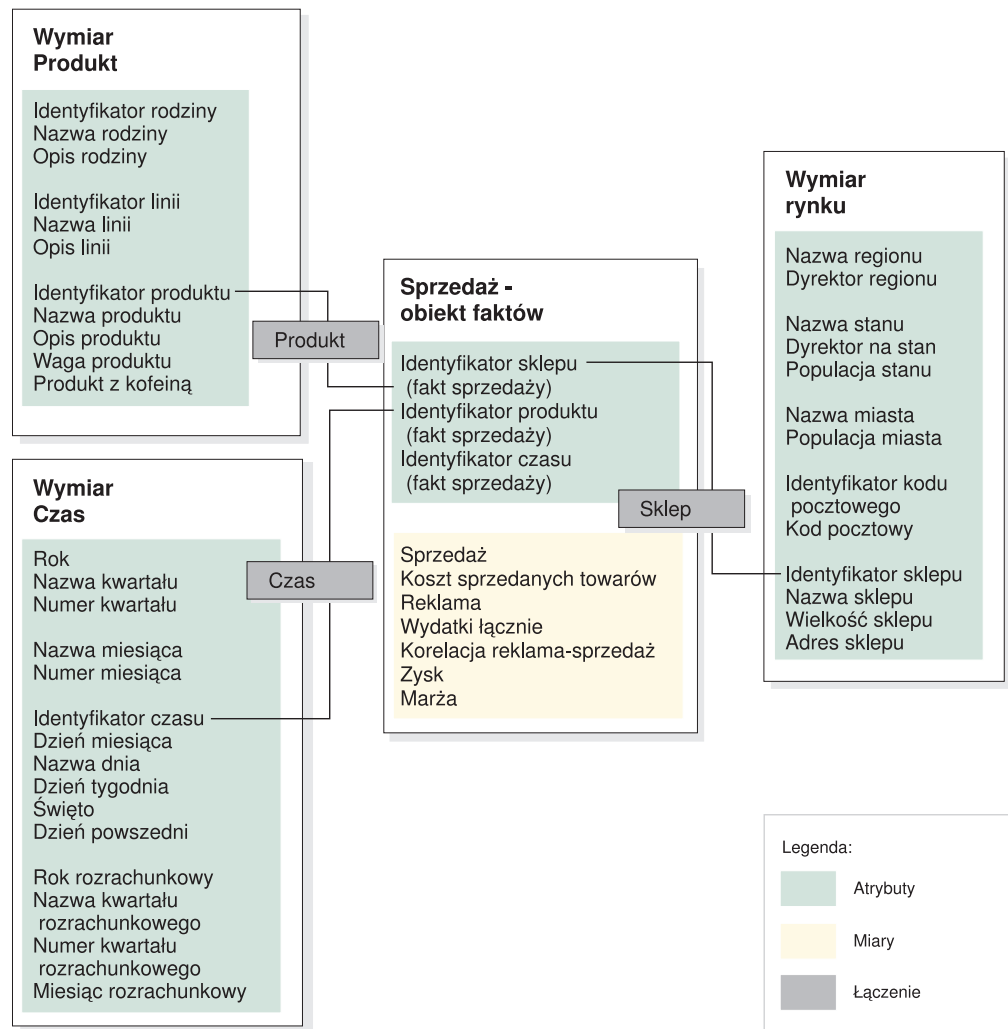
## **Tabele podsumowań**

Tabele podsumowań DB2 w programie DB2 Cube Views służą do poprawiania wydajności zapytań kierowanych do modeli kostek i kostek. Tabela podsumowań to zmaterializowana tabela zapytania (MQT) szczególnego typu zawierająca dane podsumowań.

Ponieważ Doradca optymalizacyjny zawsze zaleca tabele MQT z podsumowanymi danymi, w dokumentacji programu DB2 Cube Views na określenie zalecanych tabel MQT używany jest termin tabela podsumowań.

Duże obliczenia i łączenia dla zapytań można realizować z wyprzedzeniem, a uzyskane dane można przechowywać w tabeli podsumowań. Gdy wykonywane są zapytania, które mogą korzystać ze wstępnie obliczonych danych, program DB2 UDB przekierowuje je do tabeli podsumowań. Zapytanie nie musi być dokładnie zgodne ze wstępnymi obliczeniami. Jeśli zostaje użyta prosta funkcja analityczna, taka jak SUM czy COUNT, program DB2 UDB może dynamicznie zagregować wyniki na podstawie wstępnie obliczonych danych. Jedna tabela podsumowań może wystarczyć do obsługi wielu różnych zapytań. Użycie tabel podsumowań może radykalnie poprawić wydajność zapytań, które sięgają do często używanych danych, lub obejmujących dane zagregowane na podstawie jednego lub więcej niż jednego wymiaru bądź jednej lub więcej niż jednej tabeli.

Rys. 23 na stronie 83 przedstawia model kostki oparty na schemacie typu płatek śniegu z obiektem faktów Sprzedaż oraz wymiarami Czas, Rynek i Produkt. Obiekt faktów ma miary i atrybuty, a każdy wymiar ma zestaw atrybutów i jest połączony z obiektem faktów za pomocą łączenia faktów z wymiarem.



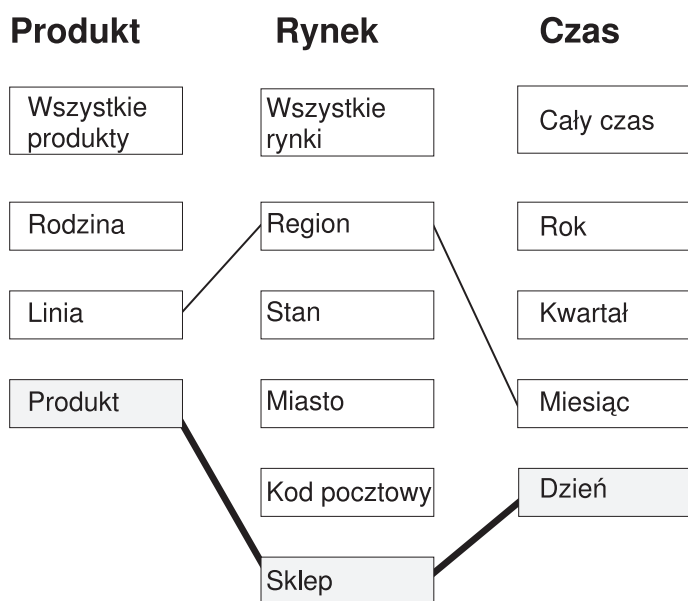
Rysunek 23. Model kostki. Model kostki z obiektem faktów Sprzedaż i wymiarami Czas, Produkt i Rynek

Hierarchie dla poszczególnych wymiarów w modelu kostki przedstawia Rys. 24 na stronie 84. Wyróżnione pola połączone pogrubioną ciemną linią u dołu hierarchii reprezentują dane, które faktycznie znajdują się w tabelach podstawowych. Dane sprzedaży są przechowywane na poziomie Dzień, na poziomie Sklep i na poziomie Produkt. Dane powyżej poziomu podstawowego w hierarchii muszą być agregowane. Jeśli tabela podstawowa jest przeszukiwana pod kątem danych sprzedaży z konkretnego miesiąca, program DB2 UDB musi dynamicznie zsumować dane sprzedaży dziennej, aby zwrócić wyniki sprzedaży miesięcznej. Na przykład za pomocą poniższego zapytania można sprawdzić dane sprzedaży dla każdej linii produktów w każdym regionie w każdym miesiącu 2004 roku:

```
SELECT LINE_ID, REGION_NAME, MONTH_NUMBER, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
  AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
  AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
  AND PRODUCT.LINEID = LINE.LINEID
  AND SALESFACT.TIMEID = TIME.TIMEID
  AND YEAR = '2004'
GROUP BY LINEID, MONTH_NUMBER;
```

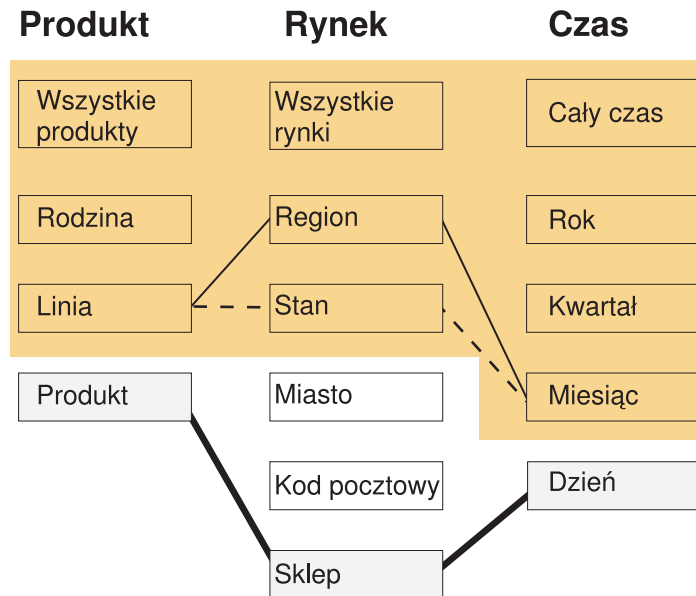
Cienka linia łącząca wycinek Linia-Region-Miesiąc na Rys. 24 na stronie 84 reprezentuje wycinek, do którego uzyskuje dostęp zapytanie. Linia-Region-Miesiąc jest wycinkiem

modelu kostki i zawiera po jednym poziomie z każdej hierarchii. Można zdefiniować tabele podsumowań, które będą zapewniały realizację zapytań na poziomie tego konkretnego wycinka lub powyżej niego. Dla wycinka Linia-Region-Miesiąc używanego przez to zapytanie można zbudować tabelę podsumowań. Do obsługi innych zapytań, które sięgają do danych na poziomie tego wycinka lub wyższym, w tym Cały czas, Rok, Kwartał, Wszystkie rynki, Wszystkie produkty i Rodzina, wystarczy ta sama tabela podsumowań i trochę dodatkowego agregowania. Jeśli jednak będzie trzeba uzyskać dokładniejsze dane poniżej tego wycinka, takie jak Dzień czy Miasto, program DB2 UDB nie będzie mógł skorzystać z tej tabeli podsumowań.



Rysunek 24. Wycinek zapytania. Hierarchie Produkt, Rynek i Czas. Widać wycinek Linia-Region-Miesiąc oraz to, że w wycinku Produkt-Sklep-Dzień znajdują się dane.

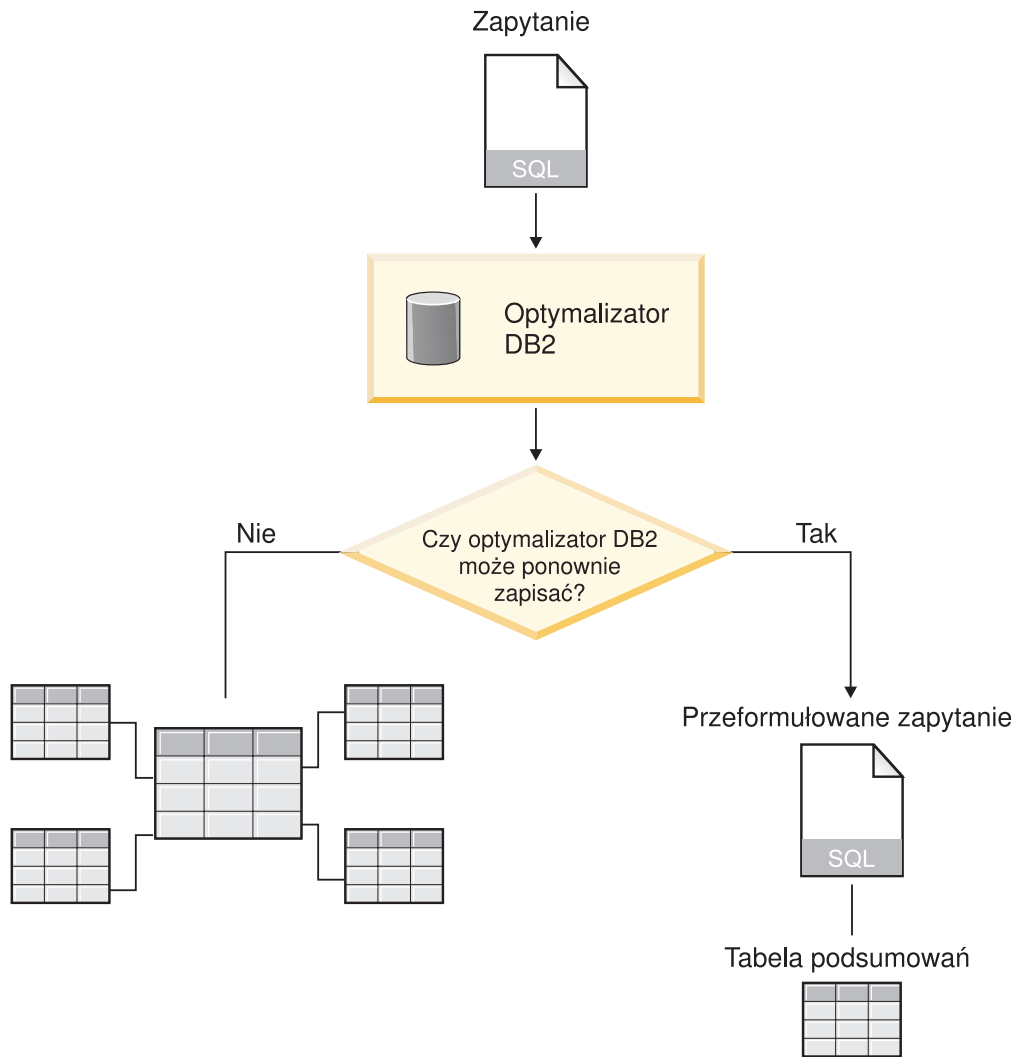
Linia kropkowana na Rys. 25 na stronie 85 definiuje wycinek Linia-Stan-Miesiąc. Tabela podsumowań zbudowana dla wycinka Linia-Stan-Miesiąc zapewnia realizację każdego zapytania sięgającego do danych tego wycinka lub powyżej niego. Wszystkie dane, które można uzyskać z tabeli podsumowań zbudowanej dla wycinka Linia-Stan-Miesiąc, znajdują się w górnym zestawie wyróżnionych pól.



Rysunek 25. Wycinek tabeli podsumowań. Hierarchie Produkt, Rynek i Czas. Wyróżnione dane można uzyskać z tabeli podsumowań zbudowanej na wycinku Linia-Stan-Miesiąc.

Moduł przebudowujący w kompilatorze kodu SQL programu DB2 dysponuje informacjami o istniejących tabelach podsumowań i może automatycznie tak przebudować zapytania, aby odczytywały dane z tabeli podsumowań zamiast z tabel podstawowych. Przeformułowane zapytania zwykle są dużo szybsze, ponieważ tabele podsumowań zwykle są dużo mniejsze niż tabele podstawowe i zawierają wstępnie zagregowane dane. Użytkownicy nadal tworzą zapytania względem tabel podstawowych. Program DB2 UDB decyduje, kiedy do danego zapytania użyć tabeli podsumowań, i przebudowuje zapytanie użytkownika tak, aby korzystało z tabel podsumowań, jak to przedstawia Rys. 26 na stronie 86. Przebudowane zapytanie korzysta z tabeli podsumowań, która zawiera wstępnie zagregowane dane. Tabela podsumowań jest często dużo mniejsza, a więc znacznie szybsza niż tabele podstawowe, a umożliwia uzyskanie tych samych wyników, co za pomocą tabel podstawowych.

Za pomocą narzędzia DB2 EXPLAIN można sprawdzić, czy zapytanie zostało przekierowane i (jeśli ma to zastosowanie) do której tabeli nastąpiło przekierowanie.



Rysunek 26. Przebudowanie zapytania. Proces programu DB2 UDB, który przebudowuje zapytanie

Zapytanie umożliwiające sprawdzenie danych sprzedaży dla każdej linii produktów we wszystkich regionach w poszczególnych miesiącach 2004 roku może zostać przebudowane tak, aby używało tabeli podsumowań zbudowanej dla wycinka Linia-Region-Miesiąc.

Pierwotne zapytanie jest następujące:

```
SELECT LINE_ID, REGION_NAME, MONTH_NUMBER, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
  AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
  AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
  AND PRODUCT.LINEID = LINE.LINEID
  AND SALESFACT.TIMEID = TIME.TIMEID
  AND YEAR = '2004'
GROUP BY LINEID, MONTH_NUMBER;
```

Przebudowane zapytanie ma następującą postać:

```
SELECT LINE_ID, REGION_NAME, MONTH_NUMBER, SUM(SALES)
FROM SUMMARYTABLE1
WHERE YEAR = '2004'
GROUP BY LINE_ID, REGION_NAME, MONTH_NUMBER;
```

Zapytanie przebudowane jest dużo prostsze i szybsze do wykonania w programie DB2 UDB, ponieważ dane są wstępnie zagregowane i wiele łążeń tabel jest wcześniej obliczonych, więc program DB2 UDB używa tylko jednej małej tabeli zamiast sześciu, wśród których jest duża tabela faktów. Tabele podsumowań mogą dawać ogromne oszczędności, szczególnie w przypadku schematów z dużymi tabelami faktów. Na przykład tabela faktów zawierająca 1 miliard wierszy może zostać zagregowana do tabeli podsumowań liczącej zaledwie 1 milion wierszy, a obliczenia związane z tym agregowaniem są wykonywane tylko raz, a nie podczas każdego kierowania zapytania. Uzyskanie dostępu do tabeli podsumowań, która jest 1000 razy mniejsza, jest dużo szybsze niż do dużych tabel podstawowych.

W tym przykładzie Rys. 27 przedstawia tabelę podsumowań dla wycinka Linia-Stan-Miesiąc. Wystarczy obliczyć dane dla wymiaru Region z wyższego poziomu Stan zamiast z niższego poziomu Sklep, więc tabela podsumowań ma mniej wierszy niż tabelę podstawową, ponieważ stanów jest mniej niż magazynów. Program DB2 UDB nie musi wykonywać dodatkowych obliczeń, aby zwrócić dane sprzedaży według atrybutów Miesiąc i Linia, ponieważ dane te są już na tych poziomach zagregowane. Zapytanie jest wykonywane na podstawie danych z tabeli podsumowań, w której tabelę użyte w zapytaniu są łączone wcześniej, a łączenie nie musi być wykonywane w trakcie przetwarzania zapytania. W przypadku bardziej złożonych zapytań wzrost wydajności może być bardzo duży.

	Nazwa regionu	Nazwa Stanu	Identyfikator linii	Rok	Numer kwartału	Nazwa kwartału	Numer miesiąca	Sprzedaż	Koszt towarów	Reklama	Wydatki łącznie	Zysk
Zachód	Idaho	054	2004	1	Kw 1	2	9700	2500	700	3200	6500	
Wschód	Maine	102	2004	2	Kw 2	5	3000	500	200	700	2300	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Rysunek 27. Tabela podsumowań. Przykład tabeli podsumowań tworzonej dla wycinka Linia-Region-Miesiąc

W pewnych przypadkach zapytanie może sięgać do atrybutu związanego z atrybutem, który znajduje się w tabeli podsumowań. Optymalizator DB2, korzystając z zależności funkcjonalnych, może dynamicznie połączyć tabelę podsumowań z odpowiednią tabelą wymiarów.

Gdy Doradca optymalizacyjny zaleca tabelę podsumowań, uwzględniane są wszystkie miary zawarte w modelu kostki. W tym przykładzie obiekt FaktySprzedaży ma tylko pięć miar (Sprzedaż, Koszt towarów, Reklama, Wydatki łącznie i Zysk); wszystkie one znajdują się w tabeli podsumowań. Jeśli w modelu kostki zostanie zdefiniowanych pięćdziesiąt miar, tabela podsumowań będzie zawierać wszystkie miary. Doradca optymalizacyjny nie musi dodawać wszystkich atrybutów pokrewnych zdefiniowanych w tabeli podsumowań, ponieważ w programie DB2 Cube Views zdefiniowane są zależności funkcjonalne między atrybutami danego poziomu.

## Tabele podsumowań z zależnościami funkcjonalnymi i ograniczeniami

Doradca optymalizacyjny na podstawie informacji o związkach między danymi, na przykład o zależnościach funkcjonalnych i ograniczeniach, zaleca tabele podsumowań zawierające zagregowane miary oraz atrybuty poziomu niezbędne optymalizatorowi DB2 do udzielenia efektywnych odpowiedzi na zapytania.

Program DB2 Cube Views, gdy tylko jest to możliwe, definiuje zależności funkcjonalne między atrybutami z danego poziomu. Definiując poziom, definiuje się związek między atrybutami klucza poziomu i pozostałymi atrybutami poziomu (atrybut domyślny i atrybuty pokrewne). Związek ten informuje, że atrybuty klucza poziomu mogą być używane łącznie do wyznaczania pozostałych atrybutów poziomu. Program DB2 Cube Views dokumentuje związek między atrybutami poziomu, definiując zależności funkcjonalne między atrybutami. Programy DB2 UDB i DB2 Cube Views korzystając z tych związków, zdefiniowanych za pomocą zależności funkcjonalnych, mogą przeprowadzać inteligentną optymalizację danych.

Należy zapewnić, aby dane bazowe atrybutów poziomów były funkcjonalnie zależne w sposób opisany przez zależność funkcjonalną. Program DB2 UDB nie sprawdza poprawności zależności funkcjonalnych.

Jeśli między atrybutem klucza poziomu a atrybutami pokrewnymi istnieje zależność funkcjonalna, Doradca optymalizacyjny może umieścić w tabeli podsumowań atrybut klucza poziomu bez atrybutów pokrewnych. Zapytania dotyczące atrybutów pokrewnych poziomu mogą być nadal kierowane do tabeli podsumowań, ponieważ podczas generowania zapytania tworzącego końcową tabelę wynikową optymalizator DB2 łączy tabelę podsumowań z tabelą wymiarów.

Na przykład za pomocą zapytania bardzo zbliżonego do opisanego w sekcji “Tabele podsumowań” na stronie 82 można sprawdzać dane sprzedaży dla każdej linii produktów we wszystkich regionach we wszystkich miesiącach w 2004 roku. Przedstawione poniżej zapytanie różni się tym, że grupuje wyniki według nazw linii, a nie identyfikatorów linii, oraz według nazw miesięcy, a nie numerów miesięcy. Tabela wynikowa jest bardziej użyteczna, ale dostęp do danych opiera się na zależnościach funkcjonalnych i ograniczeniach.

```
SELECT LINE_NAME, REGION_NAME, MONTH_NAME, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
  AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
  AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
  AND PRODUCT.LINEID = LINE.LINEID
  AND SALESFACT.TIMEID = TIME.TIMEID
  AND YEAR = '2004'
GROUP BY LINE_NAME, REGION_NAME, MONTH_NAME;
```

Optymalizator DB2 przeformułuje zapytanie tak, aby zawierało łączenie tabeli podsumowań z odpowiednimi tabelami wymiarów, jak pokazano na Rys. 28 na stronie 89.

Poziom Linia zawiera następujące atrybuty:

- kolumna Identyfikator linii jest atrybutem klucza poziomu
- kolumna Nazwa linii jest atrybutem domyślnym
- kolumna Opis linii jest atrybutem pokrewnym

Identyfikator linii jest kluczem podstawowym tabeli Linia, a więc istnieje ograniczenie kolumny Identyfikator linii. Ponieważ istnieje ograniczenie kolumny Identyfikator linii, program DB2 Cube Views nie tworzy zależności funkcjonalnej dla tego poziomu. Optymalizator DB2 za pomocą tego ograniczenia łączy tabelę podsumowań z tabelą Linia i uzyskuje dostęp do danych z pola Nazwa linii na potrzeby tabeli wynikowej zapytania.

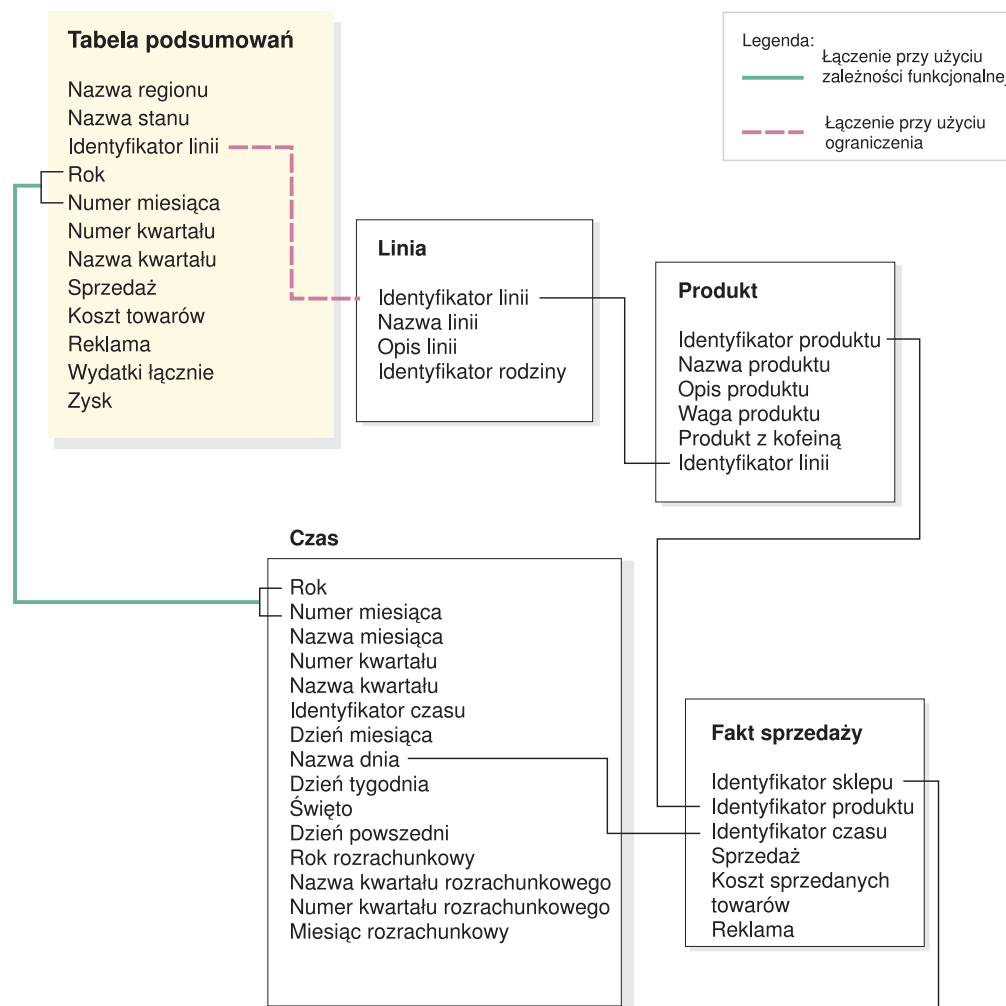
Poziom Miesiąc zawiera następujące atrybuty:

- kolumny Rok i Numer miesiąca są atrybutami klucza poziomu
- kolumna Nazwa miesiąca jest atrybutem domyślnym

Dla poziomu Miesiąc istnieje zależność funkcjonalna, według której kolumna Nazwa miesiąca jest funkcjonalnie zależna od kombinacji kolumn Rok i Numer miesiąca. Kolumny



Rok i Numer miesiąca nie wchodzą w skład klucza podstawowego ani klucza obcego tabeli Czas, nie ma więc ograniczeń dla tych kolumn. Optymalizator DB2, używając zależności funkcjonalnej między kolumną Nazwa miesiąca a kombinacją kolumn Rok i Numer miesiąca, łączy tabelę podsumowań z tabelą Czas i uzyskuje dane z kolumny Nazwa miesiąca do tabeli wynikowej zapytania.



Rysunek 28. Łączenia. Optymalizator DB2 łączy dynamicznie tabelę podsumowań z odpowiednimi tabelami wymiarów, gdy zostanie wystosowane zapytanie

Optymalizator DB2 przebudowuje zapytanie, nadając mu następującą postać:

```
SELECT Q4.LINE_NAME, Q4.REGION_NAME, Q4.MONTH_NAME, SUM(Q4.SALES)
FROM (
  SELECT DISTINCT Q1.LINE_NAME, Q3.REGION_NAME, Q2.MONTH_NAME,
    Q3.SALES, Q2.YEAR, Q2.MONTH_NUMBER
  FROM LINE AS Q1, TIME AS Q2, SUMMARYTABLE1 AS Q3
  WHERE (Q3.YEAR=2004)
    AND Q3.LINEID=Q1.LINEID)
    AND (Q3.MONTH_NUMBER=Q2.MONTH_NUMBER)
    AND (2004=Q2.YEAR)
) AS Q4
GROUP BY Q4.LINE_NAME, Q4.REGION_NAME, Q4.MONTH_NAME
```

W powstałym zapytaniu łączone są tylko trzy table, a nie sześć jak w pierwotnej wersji zapytania, a co najważniejsze, nowe zapytanie nie wymaga dostępu do dużej tabeli faktów.

która jest zazwyczaj pokaźna i wolna. Przebudowane zapytanie jest dużo szybsze, ponieważ tabela podsumowań zawiera już wstępnie zagregowane dane miar.

---

## Przegląd procesu optymalizacji

Zoptymalizowanie schematu gwiazdowego lub schematu typu płatek śniegu za pomocą programu DB2 Cube Views może podnieść wydajność zapytań SQL w stylu OLAP. Proces optymalizacji obejmuje tworzenie, implementowanie i konserwowanie tabel podsumowań zalecanych przez Doradcę optymalizacyjnego.

Doradca optymalizacyjny może pomóc w zoptymalizowaniu modeli kostek, zalecając tabele podsumowań. Tabele podsumowań programu DB2 UDB mogą poprawić wydajność zapytania, ponieważ zawierają wstępnie obliczone wyniki z jednej lub większej liczby tabel, które mogą być użyte w zapytaniu. Kosztowne łączenia tabel i złożone obliczenia mogą być realizowane z wyprzedzeniem i przechowywane w tabeli podsumowań, dzięki czemu późniejsze zapytania korzystające z tych agregacji będą mogły działać szybciej. Aby uzyskać więcej informacji na temat tabel podsumowań, patrz sekcja “Tabele podsumowań” na stronie 82.

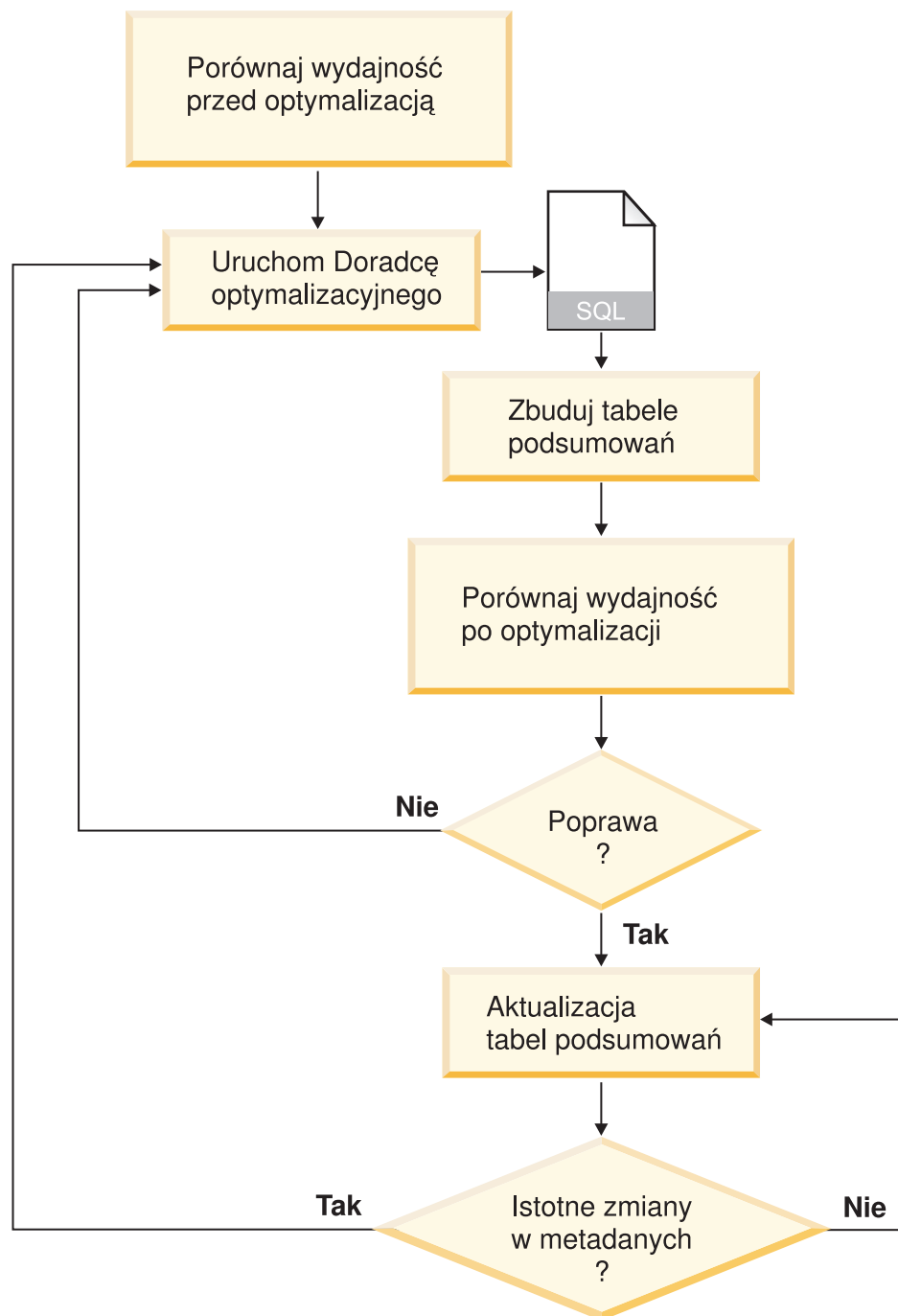
Doradca optymalizacyjny analizuje dostarczone metadane oraz informacje, po czym zaleca odpowiednie tabele podsumowań. W wyniku uruchomienia Doradcy optymalizacyjnego uzyskuje się plik SQL, który może zbudować zestaw zalecanych tabel podsumowań. Ten kod SQL można zmodyfikować przed uruchomieniem, aby utworzyć tabele podsumowań.

Uruchomienie Doradcy optymalizacyjnego to tylko pierwszy krok w procesie optymalizacji. Przed rozpoczęciem optymalizacji należy rozważyć kilka kwestii, a wśród nich następujące:

- Jak efektywnie wykorzystać ograniczenia DB2 nałożone na tabele podstawowe.
- Jak zdefiniować model kostki, aby spełniał reguły sprawdzania poprawności optymalizacji.
- Pod kątem jakich typów zapytań ma być przeprowadzona optymalizacja.
- Ile miejsca ma być udostępnione.
- Jak konserwować tabele podsumowań, aby zawarte w nich dane były aktualne.

Aby przeprowadzić optymalizację, należy najpierw zdefiniować ograniczenia dla tabel podstawowych. Aby uzyskać więcej informacji na temat wymaganych typów ograniczeń, patrz sekcja “Definicje ograniczeń optymalizacyjnych” na stronie 107.

Wiele etapów procesu optymalizacji ma charakter iteracyjny i może wymagać powtarzania w celu dopracowania i utrzymania wzrostu wydajności. Rys. 29 na stronie 91 przedstawia poglądowo główne etapy procesu optymalizacji.



Rysunek 29. Proces optymalizacji. Przegląd głównych etapów procesu optymalizacji

Na proces optymalizacji składają się następujące zasadnicze zadania:

- **Pomiar wydajności**

Przed uruchomieniem Doradcy optymalizacyjnego programu DB2 Cube Views należy zmierzyć bieżącą wydajność dla określonego zestawu typowych zapytań. Pomiar wydajności jest czynnością opcjonalną, której wyniki mogą później posłużyć do przeanalizowania efektywności optymalizacji. Do przeprowadzenia testu porównawczego można wykorzystać narzędzie do testów porównawczych db2batch, dostarczane wraz z programem DB2 UDB. Aby uzyskać więcej informacji na temat korzystania z programu db2batch, patrz sekcja “Testowanie wyników zapytania” na stronie 117. Test wydajności przeprowadza się przy użyciu zapytań przykładowych, lecz Doradca optymalizacyjny nie

wymaga zapytań przykładowych, ponieważ korzysta z metadanych i określa zalecenia bez znajomości konkretnych zapytań, które będą wydawane.

- **Uruchomienie kreatora Doradcy optymalizacyjnego**

Doradca podaje się kilka ważnych parametrów, w tym: typy zapytań, pod kątem których ma być przeprowadzona optymalizacja, ograniczenia pod względem czasu i miejsca na dysku, metodę aktualizacji oraz położenia obszarów tabel. Aby uzyskać więcej informacji na temat tych parametrów, patrz sekcja “Parametry Doradcy optymalizacyjnego” na stronie 111. Doradca optymalizacyjny określa zalecenia na podstawie informacji dostarczonych przez użytkownika, metadanych, statystyki programu DB2 oraz ewentualnych próbkowanych danych, o ile próbkowanie zostało dopuszczone przez użytkownika. Doradca optymalizacyjny analizuje podane parametry i generuje dwa pliki SQL. Jeden plik SQL zawiera komendy SQL, które mają posłużyć do zbudowania zestawu zaleconych tabel podsumowań. Drugi plik SQL zawiera komendy SQL, które mają posłużyć do zaktualizowania zaleconych tabel podsumowań.

- **Tworzenie tabel podsumowań**

Tabele podsumowań można utworzyć natychmiast po zakończeniu pracy kreatora, lecz można też dodać tę operację do harmonogramu normalnych prac konserwacyjnych na bazie danych. Tworzenie tabel podsumowań może zająć znaczną ilość czasu i zasobów procesowych. Po zbudowaniu tabel podsumowań należy sprawdzić, czy poprawiła się wydajność zapytań realizowanych na zoptymalizowanym modelu kostki. Należy uruchomić ten sam zestaw zapytań przykładowych, który był uruchamiany przed optymalizacją, a następnie porównać wyniki opisujące wydajność. Jeśli nie widać znaczącego wzrostu wydajności, potrzebne może być ponowne uruchomienie Doradcy optymalizacyjnego i przydzielenie większej ilości miejsca na dysku, wydłużenie czasu działania kreatora czy zmiana innych ustawień. Aby uzyskać więcej informacji na temat sprawdzania i analizowania wyników testu wydajności, patrz sekcja “Testowanie wyników zapytania” na stronie 117.

- **Konserwacja tabel podsumowań**

Po utworzeniu tabel podsumowań trzeba je regularnie poddawać konserwacji, aby zapewnić ich stałą synchronizację z danymi. Po uruchomieniu Doradcy optymalizacyjnego użytkownik wybiera sposób aktualizacji: odświeżanie natychmiastowe albo odświeżanie odroczone.

**Opcja odświeżania natychmiastowego**

Jeśli zostanie wybrana opcja aktualizacji natychmiastowej, program DB2 UDB będzie zachowywać tabele podstawowe i tabele podsumowań w stanie zsynchronizowanym i przyrostowo aktualizować tabele podsumowań, gdy zmianie ulegną tabele podstawowe. Program DB2 UDB obsługuje konserwację przyrostową w wypadku prostych agregacji, takich jak SUM i COUNT. W przypadku innych agregacji Doradca optymalizacyjny zaleca tabele podsumowań, które używają opcji odświeżania odroczonego niezależnie od wybranej opcji.

**Opcja odświeżania odroczonego**

Jeśli zostanie wybrana opcja odroczenia, tabele podsumowań będą zbudowane na nowo. O tym, kiedy tabela podsumowań ma zostać zaktualizowana, może zdecydować użytkownik. Jeśli w tabelach podstawowych są dokonywane znaczące zmiany, odroczenie aktualizacji może być efektywniejsze niż aktualizacje przyrostowe.

Dokonując wyboru między tymi opcjami, trzeba przyjąć kompromis między zasobami, które można przydzielić na cele konserwacyjne, a dokładnością, z jaką dane muszą być synchronizowane. Więcej informacji na temat opcji aktualizacji natychmiastowej i odroczonej można znaleźć w sekcji “Aktualizacja tabel podsumowań” na stronie 119.

- **Okresowa ponowna ocena**

Aby zapewnić stałą przydatność tabel podsumowań, co pewien czas należy dokonywać ich ponownej oceny. Jeśli metadane ulegną znaczącej zmianie na skutek dodania lub

aktualizacji modelu kostki, konieczne może być ponowne uruchomienie Doradcy optymalizacyjnego i zbudowanie nowego zestawu tabel podsumowań:

- Jeśli zostanie dodany obiekt metadanych taki jak nowy wymiar lub miara, zapytania korzystające z danych w nowym obiekcie nie będą mogły wykorzystać istniejących tabel podsumowań. Jednak z istniejących tabel podsumowań dalej będą korzystały zapytania, które nie będą używać nowego obiektu.
- Jeśli obiekt metadanych zostanie zaktualizowany tak, aby obejmował dane, pod kątem których nie został wcześniej zaktualizowany, zapytania sięgające do tego zaktualizowanego obiektu nie będą mogły wykorzystać tabel podsumowań.
- Jeśli zostanie usunięty jeden lub więcej niż jeden obiekt, efektywność tabel podsumowań nie zmieni się, jednak już nieużywane agregacje będą niepotrzebnie zajmować miejsce na dysku.

Ponowne uruchomienie doradcy może być potrzebne nie tylko w wypadku znacznych zmian w metadanych, lecz również wtedy, gdy zmieni się typ regularnie realizowanych zapytań i nie będą one odpowiadać tym, dla których przeprowadzono optymalizację.

Za każdym razem, gdy uruchamiany jest kreator i budowane są nowe tabele, należy od nowa przeprowadzić cały proces optymalizacji, włącznie z tworzeniem testów porównawczych i analizowaniem wydajności tabel podsumowań.

Jeśli usuwany jest model kostki, można także usunąć skojarzone z nim tabele podsumowań, o ile nie są używane do innych celów. Program DB2 Cube Views nie usuwa tabel podsumowań, gdy usuwany jest skojarzony z nimi model kostki. Aby uzyskać więcej informacji na temat usuwania tabel podsumowań, patrz sekcja “Usuwanie tabeli podsumowań” na stronie 120.

---

## Projekt metadanych a optymalizacja

Sposób zaprojektowania obiektów metadanych, takich jak poziomy, hierarchie, miary, kostki i wycinki optymalizacji, wpływa na tabele podsumowań zalecane przez Doradcę optymalizacyjnego.

Na ogół chcemy zdefiniować podstawową strukturę obiektu faktów, wymiarów i łączów zgodnie ze strukturą danych. Możliwości w zakresie podstawowej struktury tych obiektów wysokiego poziomu w obrębie modelu kostki jest niewiele, więc rzadko udaje się ulepszyć zalecane tabele podsumowań, wprowadzając zmiany do tych obiektów. Więcej swobody mamy w wyborze sposobu definiowania poziomów i hierarchii, miar, kostek i wycinków optymalizacji.

### Poziomy i hierarchie

O ile to możliwe, poziomy należy definiować, korzystając z idealnej metody modelowania opisanej w sekcji “Poziomy” na stronie 27. Posługując się idealną metodą modelowania, można zmniejszyć ilość miejsca na dysku zajmowanego przez zalecane tabele podsumowań oraz ilość tymczasowego miejsca używanego do odświeżania tych tabel.

### Miary

Jeśli ilość miejsca na dysku jest ograniczona, można uwzględnić tylko niewrażliwe miary w kostce i pominąć miary, które nie będą regularnie używane albo które nie mają wartości biznesowej, ponieważ im więcej miar zostanie zdefiniowanych w kostkach, tym większe będą tabele podsumowań.

Na optymalizację może wpływać również typ miar umieszczanych w kostkach - czy są one dystrybuowalne czy niedystrybuowalne:

- Miary dystrybuowalne mogą być zawsze agregowane z poziomu na poziom. Na przykład wyrażenie SUM(Sprzedaż) for Kwartał można obliczyć, sumując miesięczne dane sprzedaży.

- Miary niedystrybuowalne, takie jak odchylenie standardowe, muszą być zawsze obliczane bezpośrednio na podstawie danych bazowych i nie mogą być agregowane z poziomu na poziom.

Na ogół Doradca optymalizacyjny i optymalizator DB2 mają więcej swobody i możliwości, gdy model kostki zawiera tylko miary dystrybuowalne. Model kostki zawierający miary niedystrybuowalne można optymalizować, ale lepsze wyniki osiąga się, pomijając w kostkach miary niedystrybuowalne, jeśli nie są one potrzebne.

## Kostki

Kostki należy projektować tak, aby spełniały potrzeby prowadzonej działalności. Najefektywniejsza jest optymalizacja kostki skoncentrowanej wokół jakiegoś ważnego regionu danych modelu kostki.

Jeśli znane są jakieś cechy zapytań, które będą często uruchamiane przez użytkowników, można podać te informacje Doradcy optymalizacyjnemu w wycinkach optymalizacji. Doradca zaleci tabele podsumowań, które usprawnią zapytania kierowane do określonych regionów kostek. Wycinek optymalizacji definiuje się za pomocą zestawu konkretnych poziomów lub wyznaczników poziomów **Dowolny** określanych dla każdego wymiaru kostki oraz za pomocą oczekiwanego typu zapytania dla wycinka, takiego jak drażenie, raport, wyodrębnianie MOLAP, wyodrębnianie hybrydowe lub rozszerzanie.

W Centrum OLAP można określić, że kostka będzie generalnie używana do jednego typu zapytań, na przykład drażących. Po podaniu dla kostki jednego typ zapytań, Centrum OLAP definiuje wycinek optymalizacji z tym typem oraz z opcjami **Dowolny** dla wszystkich wymiarów kostki. Aby uzyskać więcej informacji na temat wycinków optymalizacji, patrz sekcja “Wycinki optymalizacji dla kostek” na stronie 95.

Na poniższej liście opisano typy zapytań, które można określać dla kostki w Centrum OLAP:

### Zapytania drażące

Zapytania drażące zwykle sięgają do podzbioru danych skupionego na szczycie modelu kostki. Zapytania mogą być kierowane na dowolny poziom w modelu kostki. Gdy użytkownicy głęboko drażą jeden wymiar, zwykle pozostają dużo wyżej w innych wymiarach. Optymalizacja pod kątem zapytań drażących jest najbardziej korzystna dla zapytań pozostających na górnych poziomach modelu kostki. Do realizowania zapytań drażących używane są zwykle aplikacje arkuszy kalkulacyjnych relacyjnego modelu OLAP (ROLAP). Na przykład użytkownik aplikacji arkusza kalkulacyjnego może zacząć od uzyskiwania informacji o dochodzie ze wszystkich regionów i wszystkich produktów w roku 2004. Następnie użytkownik może sięgnąć do danych głębiej, realizując zapytanie o dochód według kwartałów we wszystkich regionach i w poszczególnych krajach.

Zwykle w tego typu zapytaniach istotną rolę odgrywa wydajność, ponieważ są one wydawane w czasie rzeczywistym przez użytkownika, który musi czekać na przetworzenie wyników.

### Zapytania raportowe

Zapytania raportowe mogą z równym prawdopodobieństwem sięgać do każdej części modelu kostki. Zapytania raportowe często są wywoływane w ramach zadań wsadowych. Wydajność zwykle nie jest tutaj tak krytyczna

jak w wypadku zapytań drażących, ponieważ jest mniej prawdopodobne, aby użytkownik oczekiwał natychmiastowej odpowiedzi na każde zapytanie.

### Zapytania wyodrębniania MOLAP

Zapytania wyodrębniania MOLAP sięgają do poziomu podstawowego kostki albo wycinka optymalizacji zdefiniowanego dla kostki. Kostka służy do ładowania danych do składnicy danych MOLAP. Wycinek optymalizacji kostki jest logicznie odwzorowany na wycinek wyodrębniania, który jest ładowany do aplikacji MOLAP w celu dalszego przetwarzania.

Jeśli w Centrum OLAP dla kostki zostanie określony typ wyodrębniania MOLAP albo wycinek optymalizacji z opcją **Dowolny** dla wszystkich wymiarów kostki, Doradca optymalizacyjny utworzy tabele podsumowań zapewniające optymalizację pod kątem danych wyodrębnianych na poziomie podstawowym kostki. Należy zapewnić, aby poziomy podstawowe kostki były odwzorowane na wycinek, dla którego wyodrębniane są dane.

### Ustawienia zaawansowane

Jeśli w Centrum OLAP dla kostki zostanie określony typ Ustawienia zaawansowane, można określić wycinki optymalizacji dla konkretnych regionów kostki, których często dotyczą zapytania.

Jeśli znane są jakieś cechy zapytań, które będą często wystosowywane, można podać te informacje Doradcy optymalizacyjnemu w wycinkach optymalizacji. Doradca optymalizacyjny uwzględni te wycinki podczas zalecania tabel podsumowań. Wycinek optymalizacji definiuje się za pomocą zestawu konkretnych poziomów lub wyznaczników poziomów Dowolny określanych dla każdego wymiaru kostki oraz za pomocą oczekiwanego typu zapytania dla wycinka, takiego jak drażenie, raport, wyodrębnianie MOLAP, wyodrębnianie hybrydowe lub rozszerzanie. Aby uzyskać więcej informacji na temat wycinków optymalizacji, patrz sekcja “Wycinki optymalizacji dla kostek”

---

## Wycinki optymalizacji dla kostek

Wycinek optymalizacji to opcjonalna, ale przydatna funkcja wykorzystywana przez Doradcę optymalizacyjnego podczas przedstawiania tabel podsumowań koncentrujących się na najważniejszych regionach modelu kostki.

Kostki mogą często mieć wiele wymiarów. Określając jeden lub więcej wycinków optymalizacji można określić region kostki o największej aktywności zapytań. Wycinek optymalizacji definiuje się za pomocą zestawu jednego lub więcej poziomów oraz typu zapytania oczekiwanego dla wycinka, na przykład drażenie, raport, wyodrębnienie MOLAP, wyodrębnienie hybrydowe czy rozszerzanie. Zdefiniowanie wycinka optymalizacji może być korzystne dla zapytania każdego typu, ale najbardziej dla zapytań raportowych.

Podczas definiowania wycinka należy określić po jednej opcji na jeden wymiar kostki. Określając opcje dla poszczególnych wymiarów kostki, należy postępować zgodnie z następującymi wskazówkami:

- Jeśli wiadomo, że określony poziom jest ważny lub dotyczą go liczne zapytania, w wymiarze kostki należy określać konkretny poziom, na przykład Miesiąc w wymiarze kostki Czas.
- Jeśli ważny jest najwyższy stopień agregacji lub dotyczą go liczne zapytania, w wymiarze kostki należy użyć słowa kluczowego **Wszystkie**.

- Jeśli żaden z poziomów nie jest ważniejszy od pozostałych w tym wymiarze kostki, zapytania dotyczą wielu poziomów albo nie wiadomo, jak często zapytania dotyczą poszczególnych poziomów w wymiarze kostki, to w wymiarze kostki należy użyć słowa **Dowolne**.

Na przykład w kostce o 10 wymiarach mogą być ważne poziomy tylko w dwóch wymiarach; wtedy w tych dwóch wymiarach można określić konkretne poziomy, a w pozostałych użyć słowa kluczowego **Dowolne**.

Poniżej przedstawiono przykładowe wycinki optymalizacji dla każdego typu zapytań i tabele podsumowań, które może zalecić Doradca optymalizacyjny:

#### **Wycinki optymalizacji drażenia**

Wycinek optymalizacji drażenia oznacza, że użytkownicy często schodzą niżej do poziomów zdefiniowanych w wymiarach kostki. Dlatego Doradca optymalizacyjny powinien uwzględniać te poziomy w co najmniej jednej z zalecanych tabel podsumowań. Ponieważ typem jest zapytanie drażące, Doradca optymalizacyjny może wykonywać optymalizacje pod kątem zarówno płytkich, jak i głębokich zapytań w pewnych wymiarach kostki.

**Zalecane:** Dla danego wymiaru kostki należy określać konkretny poziom tylko wtedy, gdy wiadomo, że poziom ten jest szczególnie ważny. Ogólnie dla większości wymiarów kostki należy stosować opcję **Dowolne**, a konkretny poziom wybierać tylko wtedy, gdy poziom ten jest związany z intensywną aktywnością zapytań.

Rys. 30 na stronie 97 pokazuje przykład wycinka optymalizacji drażenia oraz wycinki, przy których Doradca optymalizacyjny może zalecić tabele podsumowań. Ten wycinek optymalizacji jest zdefiniowany na poziomie Dowolny w wymiarze kostki Produkt i w wymiarze kostki Rynek oraz na poziomie Miesiąc w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że użytkownicy często schodzą niżej do poziomu Miesiąc, ale nie mają konkretnego wzorca drażenia w wymiarach kostki Produkt i Rynek.

W prawdopodobnym zaleceniu tabel podsumowań ujęte są dwa poziomy agregacji. Jeden poziom agregacji jest zdefiniowany w wycinku Linia-Stan-Miesiąc, a drugi w wycinku Rodzina-Region-Miesiąc. Oba poziomy agregacji zawierają agregacje na poziomie Miesiąc wymiaru Czas, ale zapewniają różne poziomy agregacji w dwóch pozostałych wymiarach. Wyższy wycinek zapewnia większą poprawę wydajności w przypadku płytkich zapytań drażących, ponieważ zapytania mogą być realizowane natychmiast i program DB2 UDB nie będzie musiał wykonywać agregacji. Niższy poziom agregacji zapewnia poprawę wydajności dla głębszych zapytań drażących.

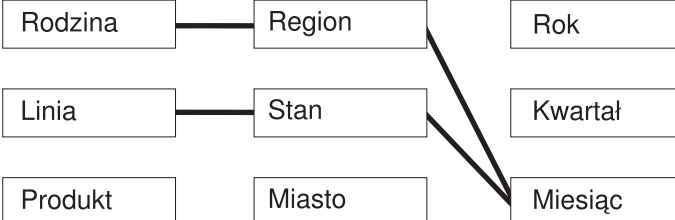


## Wycinek z drążeniem danych

Produkt	Rynek	Czas
Wszystkie produkty	Wszystkie rynki	Cały czas
Rodzina	Region	Rok
Linia	Stan	Kwartał
Produkt	Miasto	<b>Miesiąc</b>

## Możliwe zalecenia

Produkt	Rynek	Czas
Wszystkie produkty	Wszystkie rynki	Cały czas
Rodzina	Region	Rok
Linia	Stan	Kwartał
Produkt	Miasto	Miesiąc



Rysunek 30. Drążenie. Wycinek optymalizacji drążenia i możliwe zalecane tabele podsumowań

## Wycinki optymalizacji raportów

Wycinek optymalizacji raportów oznacza, że użytkownicy często tworzą raporty na poziomach zdefiniowanych w wymiarach kostki. Dlatego Doradca optymalizacyjny powinien uwzględnić te poziomy w co najmniej jednej z zalecanych tabel podsumowań.

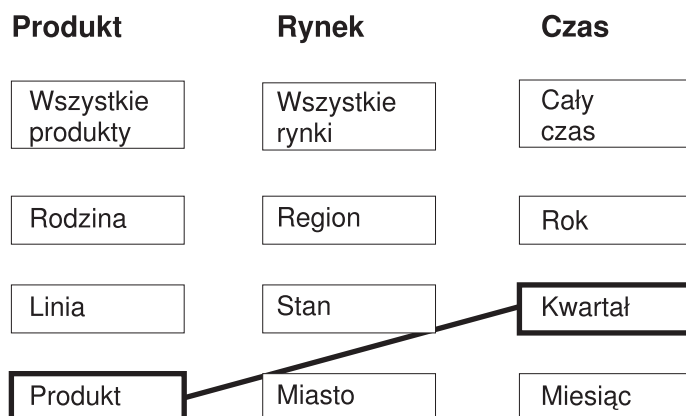
**Zalecane:** Dla danego wymiaru kostki należy określać konkretny poziom tylko wtedy, gdy wiadomo, że poziom ten jest szczególnie ważny. Ogólnie dla większości wymiarów kostki należy stosować opcję **Dowolne**, a konkretny poziom wybierać tylko wtedy, gdy poziom ten jest związany z intensywną aktywnością zapytań.

Rys. 31 na stronie 99 pokazuje przykład wycinka optymalizacji raportów oraz wycinki, przy których Doradca optymalizacyjny może zalecić tabele podsumowań.

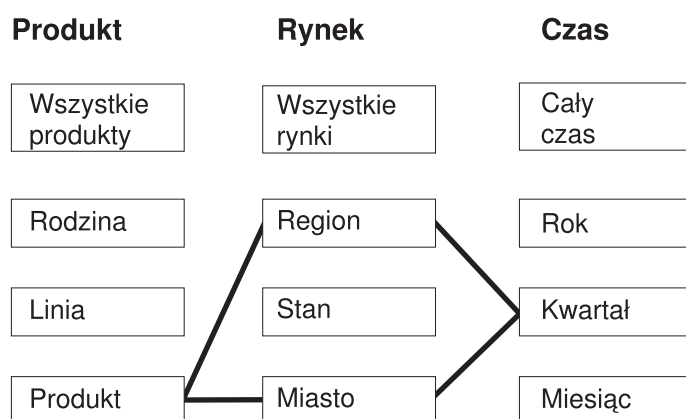
Ten wycinek optymalizacji jest zdefiniowany na poziomie Produkt w wymiarze kostki Produkt, na poziomie Dowolny w wymiarze kostki Rynek i na poziomie Kwartał w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że użytkownicy tworzą raporty, które zawierają poziomy Produkt i Kwartał oraz: nie zawierają wymiaru kostki Rynek, zawierają kilka poziomów w wymiarze kostki Rynek albo nie wiadomo, którego poziomu regionu Rynek używają użytkownicy. Użytkownicy mogą na przykład tworzyć często raporty ukazujące dane sprzedaży dla każdej rodziny produktów w ostatnich czterech kwartałach, ale raporty różniące się tym, że ukazują dane sprzedaży według pól Region, Stan i Miasto.

W prawdopodobnym zaleceniu tabel podsumowań ujęte są dwa poziomy agregacji. Jednym poziomem agregacji jest wycinek Produkt-Miasto-Kwartał, a drugim wycinek Produkt-Region-Kwartał. Oba poziomy agregacji zawierają poziomy Produkt i Kwartał określone w wycinku. Na podstawie próbkowania danych i innych metadanych Doradca optymalizacyjny zaleca, aby utworzyć tabele podsumowań obejmujące dwa poziomy agregacji, jeden zawierający poziom Miasto wymiaru kostki Rynek, a drugi zawierający poziom Region wymiaru kostki Rynek.

## Wycinek z raportem



## Możliwe zalecenia



Rysunek 31. Raport. Wycinek optymalizacji raportów i możliwe zalecane tabele podsumowań

### Wycinki optymalizacji z wyodrębnianiem typu MOLAP

Wycinek optymalizacji z wyodrębnianiem typu MOLAP oznacza, że dane określone w wycinku są często wyodrębniane do kostki MOLAP w produkcie dostawcy. Z tego względu Doradca optymalizacyjny powinien zalecić tabele podsumowań, które zapewnią, że zapytania kierowane do określonego wycinka będą szybkie.

**Zalecane:** Należy określić konkretny poziom dla każdego wymiaru kostki, aby wycinek optymalizacji odpowiadał poziomowi danych wyodrębnianych do kostki MOLAP.

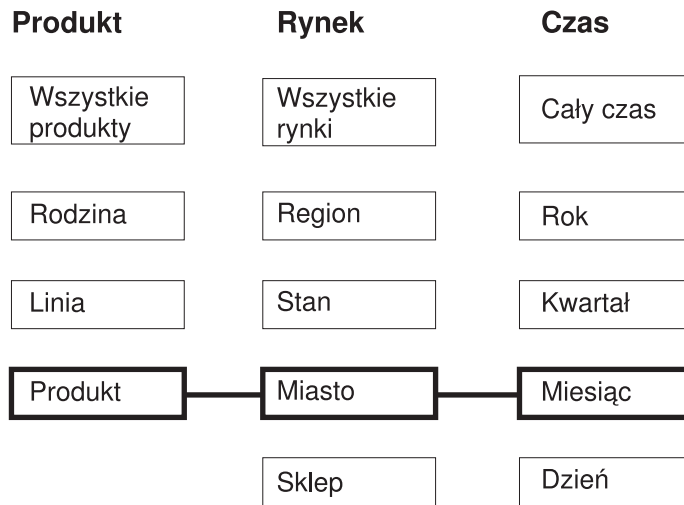
Można zdefiniować tylko jeden wycinek optymalizacji z wyodrębnianiem typu MOLAP na jedną kostkę. W kostce zawierającej wycinek optymalizacji z wyodrębnianiem typu MOLAP nie można definiować wycinka optymalizacji z wyodrębnianiem hybrydowym.

Rys. 32 na stronie 100 pokazuje przykład wycinka optymalizacji z wyodrębnianiem typu MOLAP oraz wycinek, przy którym Doradca optymalizacyjny może zalecić

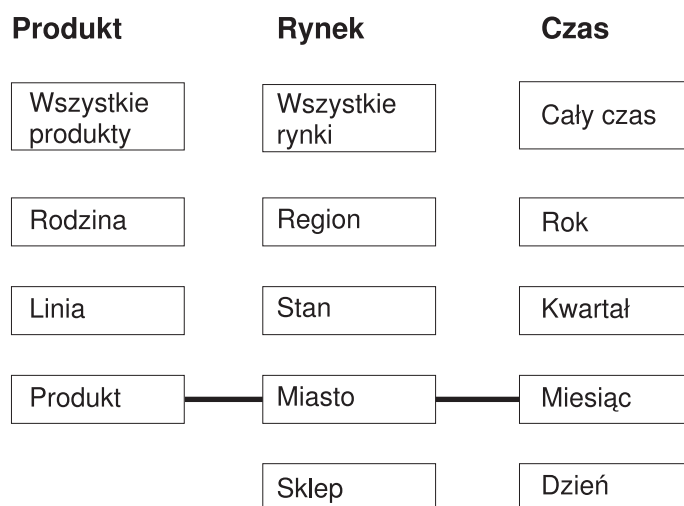
tabele podsumowań. Ten wycinek optymalizacji jest zdefiniowany na poziomie Produkt w wymiarze kostki Produkt, na poziomie Miasto w wymiarze kostki Rynek i na poziomie Miesiąc w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że dane z poziomów Produkt-Miasto-Miesiąc będą wyodrębniane do kostki MOLAP.

Możliwe zalecenie zawiera jedną tabelę podsumowań, która bezpośrednio spełnia zapytanie wyodrębniania MOLAP określone przez wycinek optymalizacji Produkt-Miasto-Miesiąc.

## Wycinek z wyodrębnianiem MOLAP



## Możliwe zalecenia



Rysunek 32. Wyodrębnianie MOLAP. Wycinek optymalizacji z wyodrębnianiem typu MOLAP i możliwe zalecane tabele podsumowań

## Wycinek optymalizacji z wyodrębnianiem hybrydowym

Wycinek optymalizacji z wyodrębnianiem hybrydowym oznacza, że dane określone w wycinku są często wyodrębniane do hybrydowej kostki OLAP (HOLAP) w

produkcje dostawcy. Z tego względu Doradca optymalizacyjny powinien uwzględnić określony wycinek w zalecanej tabeli podsumowań.

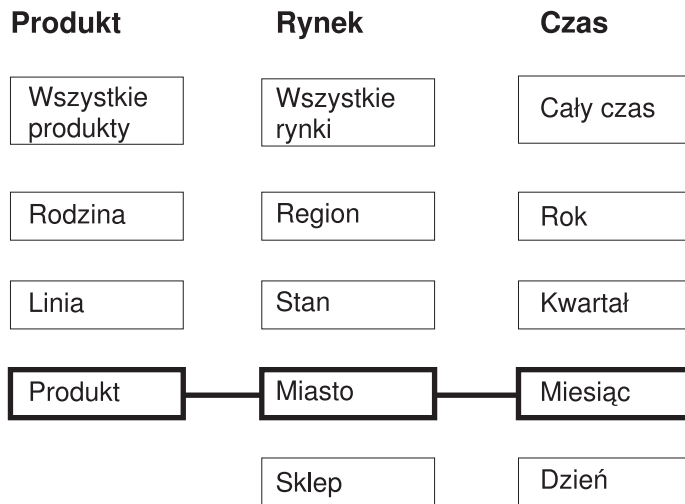
**Zalecane:** Należy określić konkretny poziom dla każdego wymiaru kostki, aby wycinek optymalizacji odpowiadał poziomowi danych wyodrębnianych do kostki HOLAP.

Można zdefiniować tylko jeden wycinek optymalizacji z wyodrębnieniem hybrydowym na jedną kostkę. W kostce zawierającej wycinek optymalizacji z wyodrębnieniem hybrydowym nie można definiować wycinka optymalizacji z wyodrębnieniem typu MOLAP. W kostce zawierającej wycinek optymalizacji z wyodrębnieniem hybrydowym można zdefiniować dowolną liczbę wycinków optymalizacji z rozszerzaniem. Doradca optymalizacyjny zakłada, że zapytania rozszerzeń będą poniżej określonego wycinka optymalizacji z wyodrębnieniem hybrydowym i próbuje wykonywać optymalizację pod kątem zapytań rozszerzeń poniżej określonego wycinka oraz pod kątem zapytań wyodrębniania hybrydowego dla określonego wycinka.

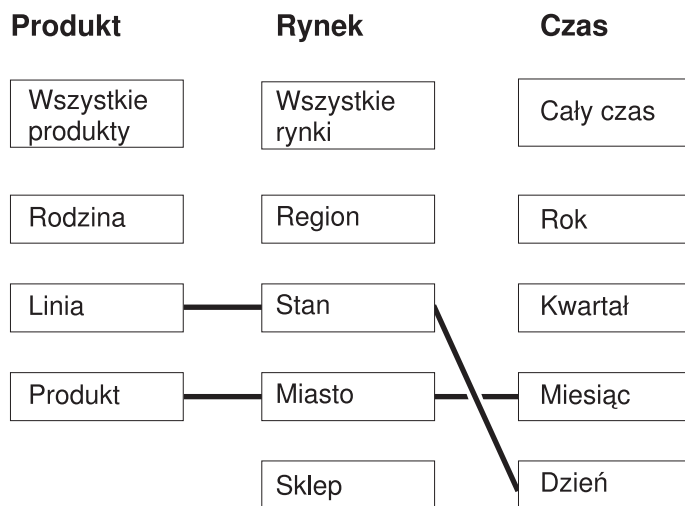
Rys. 33 na stronie 102 pokazuje przykład wycinka optymalizacji z wyodrębnieniem hybrydowym oraz wycinek, przy którym Doradca optymalizacyjny może zalecić tabelę podsumowań. Ten wycinek optymalizacji jest zdefiniowany na poziomie Produkt w wymiarze kostki Produkt, na poziomie Miasto w wymiarze kostki Rynek i na poziomie Miesiąc w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że dane z poziomów Produkt-Miasto-Miesiąc będą wyodrębniane do kostki HOLAP.

W prawdopodobnym zaleceniu tabel podsumowań ujęte są dwa poziomy agregacji. Poziom agregacji dla wycinka Produkt-Miasto-Miesiąc bezpośrednio spełnia zapytanie wyodrębniania HOLAP określone przez wycinek optymalizacji. Poziom agregacji dla wycinka Linia-Stan-Dzień zawiera poziom Dzień, który jest poniżej wycinka wyodrębniania hybrydowego, aby spełniane były ewentualne zapytania rozszerzeń w wymiarze kostki Czas. Doradca optymalizacyjny przeanalizował pozostałe metadane i wykonał próbkowanie danych, aby przygotować tę zalecaną tabelę podsumowań.

## Wycinek z wyodrębnieniem hybrydowym



## Możliwe zalecenia



Rysunek 33. Wyodrębnianie hybrydowe. Wycinek optymalizacji z wyodrębnieniem hybrydowym i możliwe zalecane tabele podsumowań

### Wycinki optymalizacji z rozszerzaniem

Wycinkowi optymalizacji z rozszerzaniem musi odpowiadać zdefiniowany w kostce wycinek optymalizacji z wyodrębnieniem hybrydowym. Wycinek optymalizacji z rozszerzaniem oznacza, że często wykonywane jest rozszerzanie do określonego wycinka z hybrydowej kostki OLAP (HOLAP) w produkcie dostawcy. Z tego względu Doradca optymalizacyjny powinien umieścić wycinek na określonych poziomach w zalecanej tabeli podsumowań lub poniżej ich.

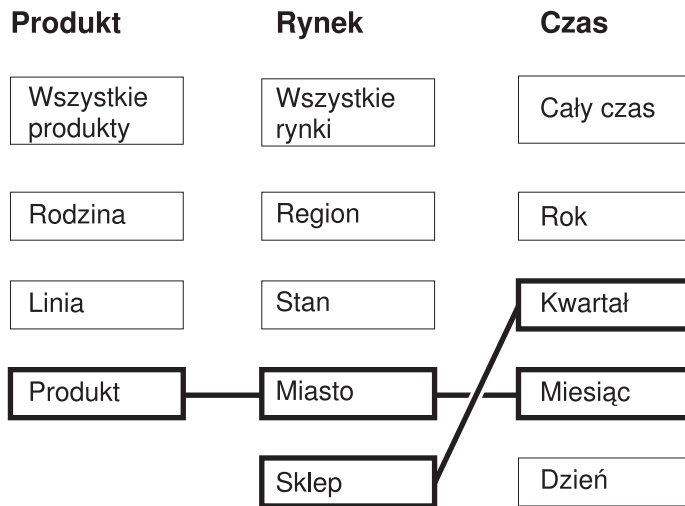
**Zalecane:** Konkretny poziom w wymiarze kostki dla wycinka optymalizacji z rozszerzaniem należy określić tylko wtedy, gdy wiadomo, że poziom ten jest szczególnie ważny. Ogólnie w przypadku wycinków optymalizacji z rozszerzaniem należy stosować opcję **Dowolne** dla większości

wymiarów kostki, a konkretny poziom wybierać tylko wtedy, gdy poziom ten jest związany z intensywną aktywnością zapytań.

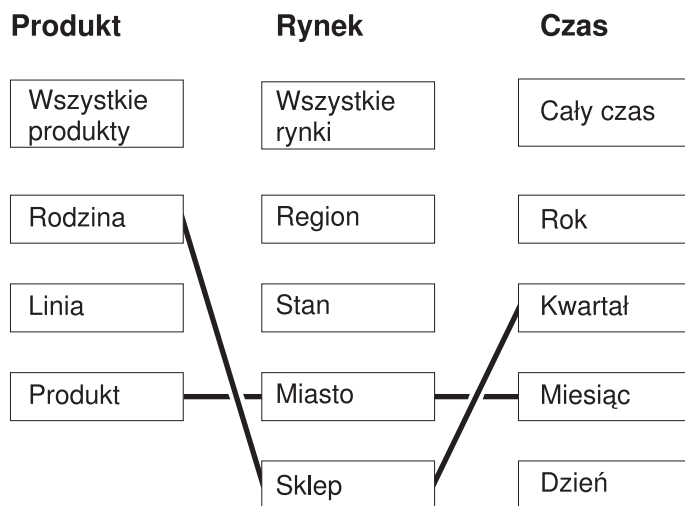
Rys. 34 na stronie 104 pokazuje przykład wycinka optymalizacji z rozszerzaniem i odpowiadający mu wycinek optymalizacji z wyodrębnieniem hybrydowym oraz wycinki, przy których Doradca optymalizacyjny może zalecić tabele podsumowań. Wycinek optymalizacji z wyodrębnieniem hybrydowym jest zdefiniowany na poziomie Produkt w wymiarze kostki Produkt, na poziomie Miasto w wymiarze kostki Rynek i na poziomie Miesiąc w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że dane z poziomów Produkt-Miasto-Miesiąc będą wyodrębniane do kostki HOLAP. Wycinek optymalizacji z rozszerzaniem jest zdefiniowany na poziomie Dowolny wymiaru kostki Produkt, na poziomie Sklep w wymiarze kostki Rynek i na poziomie Kwartał w wymiarze kostki Czas. Ten wycinek optymalizacji oznacza, że zapytania rozszerzające z kostki HOLAP zawierają zwykle poziomy Sklep i Kwartał i że zapytania rozszerzające mogą, ale nie muszą odwoływać się do konkretnych poziomów z wymiaru kostki Produkt.

W prawdopodobnym zaleceniu tabel podsumowań ujęte są dwa poziomy agregacji. Tabela podsumowań dla wycinka Produkt-Miasto-Miesiąc spełnia bezpośrednio zapytanie wyodrębniania HOLAP określone przez ten wycinek optymalizacji. Poziom agregacji dla wycinka Rodzina-Sklep-Kwartał zawiera poziomy Sklep i Kwartał określone przez wycinek optymalizacji z rozszerzaniem. Doradca optymalizacyjny po przeanalizowaniu pozostałych metadanych i wykonaniu próbkowania danych zalecił, aby ta tabela podsumowań zawierała także poziom Rodzina z wymiaru kostki Produkt.

## Wycinek z wyodrębnianiem hybrydowym i drażeniem



## Możliwe zalecenia



Rysunek 34. Rozszerzanie. Wycinek optymalizacji z wyodrębnianiem hybrydowym i wycinek optymalizacji z rozszerzaniem oraz możliwe zalecane tabele podsumowań

## Analizowanie zapytań pod kątem proponowanych wycinków optymalizacji

Wycinki optymalizacji stanowią narzędzie umożliwiające znaczącą poprawę wydajności zapytań, lecz są skuteczne tylko wtedy, gdy w sposób właściwy odzwierciedlają zapytania SQL.

Podczas analizowania wzorców zapytań wydawanych przez użytkowników, należy wziąć pod uwagę następujące informacje:

- Typ zapytań, które najprawdopodobniej będą zgłaszać użytkownicy.
- Poziomy hierarchii, do których zapytania często się odwołują.
- Produkt, z którego użytkownicy zgłaszają zapytania.



Aby określić, dla którego regionu kostki powinny zostać utworzone wycinki optymalizacji, można przejrzeć historię zapytań użytkowników. Warto również dowiedzieć się od samych użytkowników, czy spodziewają się, że w przyszłości zmienią się ich potrzeby co do zapytań. Należy rozpoznać obszary, których optymalizacja ma priorytet najwyższy.

**Zalecenie:** Nie jest wskazane określanie więcej niż trzech wycinków optymalizacji na kostkę.

### Przykłady wycinków optymalizacji odpowiadające konkretnym sytuacjom

Każdy przykład jest oparty na kostce Sprzedaż dzienna w bazie danych CVSAMPLE, w której są trzy wymiary kostki: Produkt, Rynek i Czas, przy czym każdy ma następujące hierarchie kostki:

*Tabela 35. CVSAMPLE. Wymiary kostki i odpowiadające im hierarchie kostki Sprzedaż dzienna w bazie danych CVSAMPLE.*

Wymiar kostki Produkt	Wymiar kostki Rynek	Wymiar kostki Czas
Wszystkie produkty	Wszystkie rynki	Cały czas
Rodzina	Region	Rok
Linia	Stan	Kwartał
Produkt	Miasto	Miesiąc
	Kod pocztowy Sklep	Dzień

Sytuacje opisane w poniższej tabeli stanowią przykłady wycinków optymalizacji, które na podstawie bazy danych CVSAMPLE można definiować pod kątem konkretnego obciążenia zapytaniami.

*Tabela 36. Przykładowe wycinki optymalizacji*

Sytuacja	Scenariusz	Wycinki optymalizacji do zdefiniowania
Użytkownicy używają programu do zapytań zgłaszającego zapytania konkretnego typu	Wiadomo, że użytkownicy najczęściej używają programu, który zgłasza zapytania drażące. Brak jest jakichkolwiek informacji o tym, jakie regiony kostki są przeszukiwane najczęściej.	Ponieważ nie są znane żadne konkretne informacje o tym, jakie regiony kostki są najważniejsze dla użytkowników, konkretny wycinek optymalizacji nie da dużych korzyści. W tej niesprecyzowanej sytuacji całą kostkę można skonfigurować pod kątem zapytań drażących podczas tworzenia kostki w Kreatorze kostek lub zmodyfikować w tym celu właściwości kostki po utworzeniu danej kostki.

Tabela 36. Przykładowe wycinki optymalizacji (kontynuacja)

Sytuacja	Scenariusz	Wycinki optymalizacji do zdefiniowania
Większość zapytań jest jednego typu i koncentruje się na jednym grupowaniu poziomów	Wiadomo, że użytkownicy przede wszystkim zgłaszają zapytania raportowe i że niemal wszystkie zapytania odwołują się do poziomu Stan i wymiaru Rynek.	Znane są konkretne informacje o szczególnie znaczącym regionie kostki, więc utworzenie wycinka optymalizacji będzie bardzo korzystne. Można zdefiniować wycinek Dowolny-Stan-Dowolny typu raportowego. Ten wycinek odwołuje się do poziomu Dowolny w wymiarze kostki Produkt, do poziomu Stan w wymiarze kostki Rynek oraz do poziomu Dowolny w wymiarze kostki Czas.
Większość zapytań jest jednego typu i koncentruje się na kilku grupowaniach poziomów	Wiadomo, że użytkownicy zgłaszają przede wszystkim zapytania raportowe. Około połowy zapytań odwołuje się do poziomu Stan wymiaru kostki Rynek, a druga połowa zapytań jest losowo rozproszona po innych poziomach wymiaru kostki.	Można zdefiniować następujące dwa wycinki optymalizacji, które będą w sposób odpowiedni odzwierciedlać to rozproszenie zapytań: <ul style="list-style-type: none"> <li>• Zdefiniuj raportowy wycinek optymalizacji z poziomami Dowolny-Stan-Dowolny odpowiadającymi zapytaniom odwołującym się do poziomu Stan. Ten wycinek odwołuje się do poziomu Dowolny w wymiarze kostki Produkt, do poziomu Stan w wymiarze kostki Rynek oraz do poziomu Dowolny w wymiarze kostki Czas.</li> <li>• Zdefiniuj raportowy wycinek optymalizacji z poziomami Dowolny-Dowolny-Dowolny odpowiadającymi pozostałym zapytaniom, które mogą odwoływać się do dowolnych poziomów w dowolnym wymiarze kostki. Określenie tego drugiego wycinka stanowi wyraźne wskazanie, że istnieje znacząca liczba zapytań sięgających do pozostałych poziomów wymiaru Wymiar.</li> </ul>

Tabela 36. Przykładowe wycinki optymalizacji (kontynuacja)

Sytuacja	Scenariusz	Wycinki optymalizacji do zdefiniowania
Większość zapytań jest jednego typu i koncentruje się na wielu grupowaniach poziomów	Wiadomo, że użytkownicy zgłaszają przede wszystkim zapytania raportowe. Ukierunkowanie zapytań nie jest przypadkowe, lecz jest rozproszone na 15 regionów kostki.	Celem wycinka optymalizacji jest określenie obszarów intensywnie przeszukiwanych zapytaniami, a w tej sytuacji zapytania są zbyt rozproszone, aby móc określić konkretny zestaw wycinków optymalizacji. Zamiast określać wycinki optymalizacji, wyznacz całą kostkę na zapytania raportowe podczas tworzenia kostki w Kreatorze kostek lub zmodyfikuj w tym celu właściwości kostki po utworzeniu danej kostki.

## Definicje ograniczeń optymalizacyjnych

Ograniczenia dostarczają cennych informacji Doradcy optymalizacyjnemu oraz optymalizatorowi DB2. Ograniczenia informacyjne lub ograniczenia wymuszane dla kluczy obcych i kluczy podstawowych muszą być zdefiniowane w schemacie gwiazdzistym lub w schemacie typu płatek śniegu.

Aby można było użyć Doradcy optymalizacyjnego, należy najpierw zdefiniować ograniczenia dla tabel podstawowych. Aby model kostki nadawał się do optymalizacji, ograniczenia te muszą obsługiwać reguły podstawowe, reguły kompletności modelu kostki oraz reguły optymalizacji opisane w sekcji “Reguły obiektów metadanych” na stronie 38. Reguły przede wszystkim definiują sposób wzajemnego łączenia obiektów metadanych w modelu kostki.

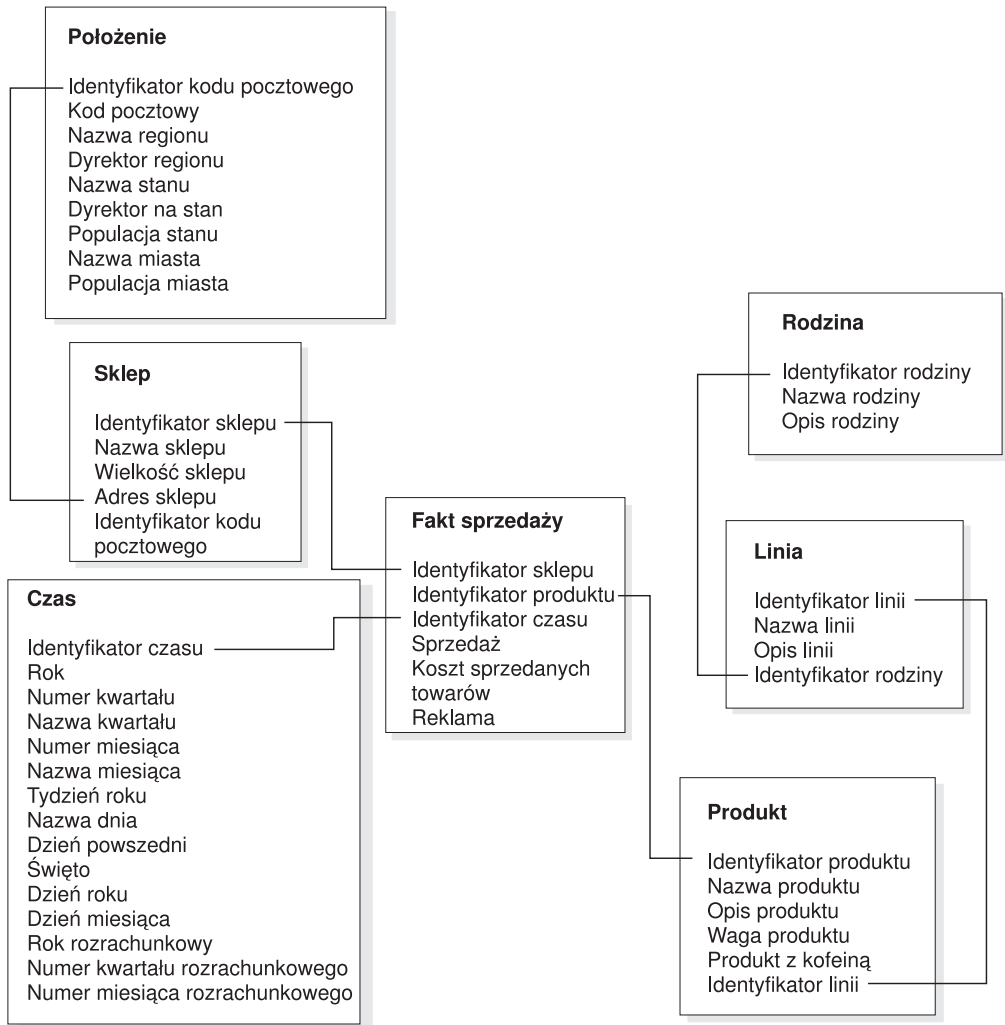
Ograniczenia informacyjne można zastosować do ograniczeń opartych na kluczu obcym, które to ograniczenia trzeba zdefiniować. Ograniczenia informacyjne to nowy typ ograniczeń udostępniany w programie DB2 Universal Database, wersja 8. Ograniczenia informacyjne umożliwiają poprawę wydajności bez zwiększania kosztów konserwacji. Tych ograniczeń może używać kompilator SQL programu DB2, lecz ich stosowanie nie jest wymuszane przez menedżera bazy danych. Ograniczenia tego typu sprawiają, że program DB2 UDB dysponuje informacjami na temat związków między danymi, ale nie wymusza stosowania tych ograniczeń. W przypadku ograniczeń kluczy podstawowych należy używać ograniczeń wymuszanych przez bazę danych, których obsługę zapewnia program DB2 UDB.

Stosowne ograniczenie musi być zdefiniowane dla każdego łączenia. Ograniczeń wymagają na przykład kolumny biorące udział w łączeniach typu fakty-wymiar i wymiar-wymiar, używanych w schemacie typu płatek śniegu.

Aby zoptymalizować model kostki oparty na schemacie typu płatek śniegu pokazanym na Rys. 35 na stronie 108, trzeba zdefiniować ograniczenia dla każdego łączenia fakty-wymiar. Te trzy łączenia fakty-wymiar są określone między następującymi faktami i wymiarami:

- między polami Sklep.IdSklepu i Sprzedaż.IdSklepu
- między polami Czas.IdCzasu i Sprzedaż.IdCzasu
- między polami Produkt.IdProduktu i Sprzedaż.IdProduktu

Do każdego z tych łączy odnosi się kilka reguł. Ograniczeń informacyjnych można używać tylko dla ograniczeń opartych na kluczu obcym.



Rysunek 35. Schemat typu płatek śniegu. Tabele relacyjne używane w schemacie typu płatek śniegu z bazy danych CVSAMPLE.

Dla łączenia między tabelami Sklep i Sprzedaż należy zdefiniować następujące ograniczenia:

- IdSklepu jest kluczem podstawowym w tabeli Sklep.
- W kolumnach Sklep.IdSklepu i Sprzedaż.IdSklepu nie są dopuszczalne wartości puste.
- Kolumna Sprzedaż.IdSklepu jest kluczem obcym odwołującym się do kolumny Sklep.IdSklepu. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.
- Jeśli kolumna Sprzedaż.IdSklepu nie jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:Wielu (Sklep.IdSklepu : Sprzedaż.IdSklepu). Jeśli kolumna Sprzedaż.IdSklepu jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:1.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

Dla łączenia między tabelami Czas i Sprzedaż należy zdefiniować następujące ograniczenia:

- IdCzasu jest kluczem podstawowym w tabeli Czas.
- Czas.IdCzasu i Sprzedaż.IdCzasu są kolumnami niedopuszczającymi wartości pustych.
- Kolumna Sprzedaż.IdCzasu jest kluczem obcym odwołującym się do kolumny Czas.IdCzasu. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.

- Jeśli kolumna Sprzedaż.IdCzasu nie jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:Wielu (Czas.IdCzasu : Sprzedaż.IdCzasu). Jeśli Sprzedaż.IdCzasu jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:1.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

Dla łączenia między tabelami Produkt i Sprzedaż należy zdefiniować następujące ograniczenia:

- IdProduktu jest kluczem podstawowym w tabeli Produkt.
- Produkt.IdProduktu i Sprzedaż.IdProduktu są kolumnami niedopuszczającymi wartości pustych.
- Kolumna Sprzedaż.IdProduktu jest kluczem obcym odwołującym się do kolumny Produkt.IdProduktu. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.
- Jeśli kolumna Sprzedaż.IdProduktu nie jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:Wielu (Produkt.IdProduktu : Sprzedaż.IdProduktu). Jeśli kolumna Sprzedaż.IdProduktu jest kluczem podstawowym tabeli Sprzedaż, łączenie ma licznosc 1:1.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

W schemacie typu płatek śniegu każdy wymiar ma tabelę wymiarów podstawowych, z którą mogą być połączone dodatkowe wymiary. Tabela wymiarów podstawowych jest jedyną tabelą, która może mieć łączenie z tabelą faktów. Każda z tabel oddalonych, która łączy się z tabelą podstawową, musi mieć łączenie o licznosci Wiele:1 (gdzie Wiele odnosi się do strony tabeli podstawowej) lub 1:1. Ze względu na te reguły licznosci łączenia tabela wymiarów podstawowych zwykle zawiera informacje o największej szczegółowości spośród wszystkich tabel wymiarów. Jeśli w zestawie tabel wymiarów używana jest tylko licznosc łączenia 1:1, wszystkie tabele mają ten sam poziom szczegółowości.

Ten model kostki opiera się na schemacie typu płatek śniegu, dlatego trzeba zdefiniować dodatkowe ograniczenia dla łączy między tabelami wymiarów. Ograniczenia dla łączy wymiar-wymiar są następujące:

- między tabelami Sklep i Położenie
- między tabelami Produkt i Linia
- między tabelami Linia i Rodzina

Do każdego z tych łączy odnosi się kilka reguł. Ograniczeń informacyjnych można używać tylko dla ograniczeń opartych na kluczu obcym.

Dla łączenia między tabelami Sklep i Położenie należy zdefiniować następujące ograniczenia:

- IdKoduPocztowego jest kluczem podstawowym w tabeli Położenie.
- Kolumny Położenie.IdKoduPocztowego i Sklep.IdKoduPocztowego nie mogą zawierać wartości pustych.
- Kolumna Sklep.IdKoduPocztowego jest kluczem obcym odwołującym się do kolumny Położenie.IdKoduPocztowego. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.
- Łączenie ma licznosc 1:Wielu (Położenie.IdKoduPocztowego : Sklep.IdKoduPocztowego), ponieważ kolumna Sklep.IdKoduPocztowego nie jest kluczem podstawowym ani unikalnym tabeli Sklep.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

Dla łączenia między tabelami Produkt i Linia należy zdefiniować następujące ograniczenia:

- IdLinii jest kluczem podstawowym tabeli Linia.
- Kolumny Linia.IdLinii i Produkt.IdLinii nie mogą zawierać wartości pustych.

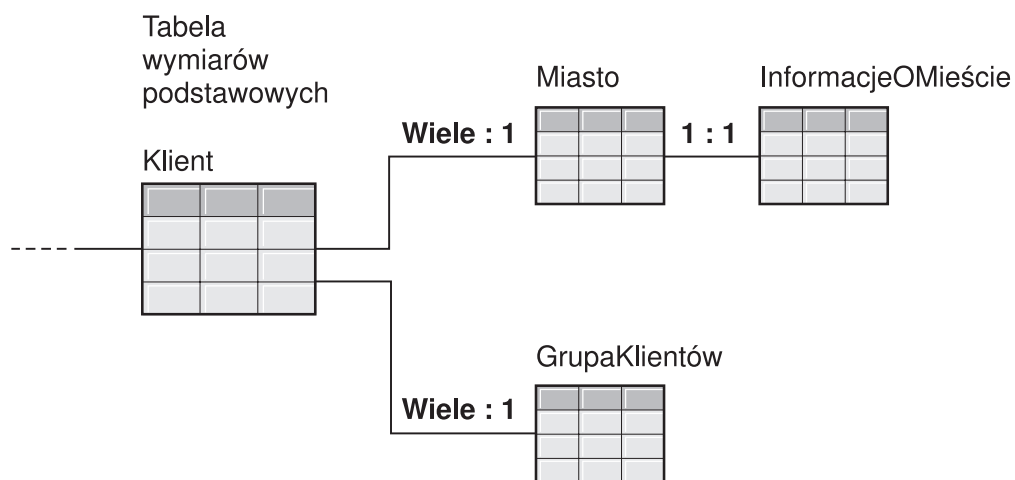
- Kolumna Produkt.IdLinii jest kluczem obcym odwołującym się do kolumny Linia.IdLinii. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.
- Łączenie ma licznosc 1:Wielu (Linia.IdLinii : Produkt.IdLinii), ponieważ kolumna Produkt.IdLinii nie jest kluczem podstawowym ani unikalnym tabeli Produkt.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

Dla łączenia między tabelami Linia i Rodzina należy zdefiniować następujące ograniczenia:

- IdRodziny jest kluczem podstawowym tabeli Rodzina.
- Kolumny Rodzina.IdRodziny i Linia.IdRodziny nie mogą zawierać wartości pustych.
- Kolumna Linia.IdRodziny jest kluczem obcym odwołującym się do kolumny Rodzina.IdRodziny. Ograniczenia oparte na kluczu obcym można definiować jako ograniczenia informacyjne.
- Łączenie ma licznosc 1:Wielu (Rodzina.IdRodziny : Linia.IdRodziny), ponieważ kolumna Linia.IdRodziny nie jest kluczem podstawowym ani unikalnym tabeli Linia.
- Typ łączenia to ŁĄCZENIE WEWNĘTRZNE.

Rys. 36 przedstawia poprawny zestaw tabel wymiarów w wymiarze schematu typu płatek śniegu. Tabelą wymiarów podstawowych jest tabela Klient; trzy dodatkowe tabele oddalone Miasto i GrupaKlientów połączone są bezpośrednio z tabelą Klient, a tabela InformacjeOMieście połączona jest z tabelą Miasto. Licznosci łączeń są semantycznie poprawne, ponieważ w mieście lub grupie klientów może być wielu klientów, a na jedno miasto przypada jeden zestaw informacji o mieście. Jest to wymiar nadający się do optymalizacji, ponieważ są w nim przestrzegane optymalizacyjne reguły sprawdzania poprawności. Wymiar ma tylko jedną tabelę podstawową, a tabele Miasto i GrupaKlientów, które są połączone bezpośrednio z tabelą podstawową, mają łączenie o licznosci wiele:1. Tabela InformacjeOMieście ma łączenie o licznosci 1:1, która również jest poprawna. Tabela Klient zawiera informacje o największej szczegółowości spośród wszystkich czterech tabel wymiarów.

## Wymiar nadający się do optymalizacji



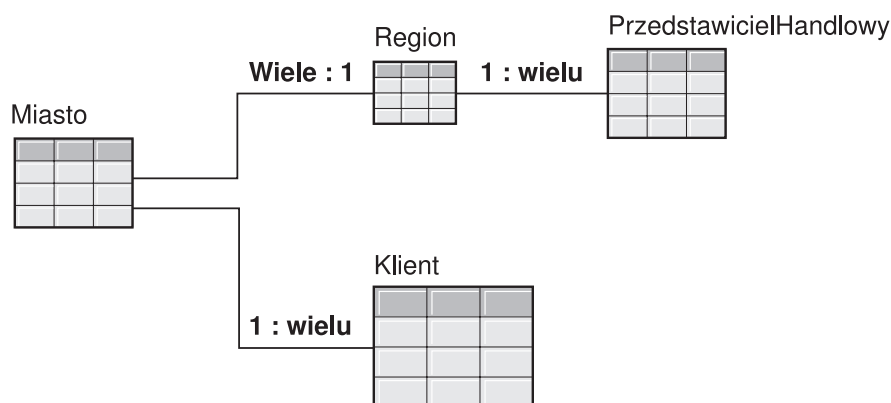
Rysunek 36. Poprawny wymiar. Zestaw tabel wymiarów używanych w jednym wymiarze, który można zoptymalizować

Rys. 37 na stronie 111 przedstawia niepoprawny zestaw tabel wymiarów w wymiarze schematu typu płatek śniegu. Ze względu na zdefiniowane związki licznosci żadna z tych tabel nie może być tabelą wymiarów podstawowych w modelu kostki przeznaczony do optymalizacji. Chociaż licznosci są semantycznie poprawne, to gdyby którakolwiek z tych

tabel była połączona z tabelą faktów jako tabela wymiarów podstawowych, dane w tabeli faktów byłyby zwielokrotnione, co spowodowałoby sytuację zwaną pułapką wachlarzową.

Na przykład, jeśli tabelą wymiarów podstawowych jest tabela Klient, to liczność łączenia 1:wielu między tabelami Region a PrzedstawicielHandlowy tworzy wymiar nienadający się do optymalizacji. Jeśli każdy region ma pięciu przedstawicieli handlowych, a tabele PrzedstawicielHandlowy i Region są połączone, to na każdy region przypada po pięć wpisów. Gdy te tabele są połączone z tabelami Miasto i Klient, a ostatecznie z tabelą faktów, dla każdego istniejącego wiersza z tabeli Miasto, Klient i z tabeli faktów dodawanych jest pięć wierszy. Pięciokrotne powtórzenie tego samego wiersza faktów prowadzi do błędnego obliczenia miar. Podobny problem dotyczy każdej z pozostałych tabel w wymiarze. Tabela Miasto nie może być tabelą wymiarów podstawowych ze względu na łączenia 1:wielu między tabelami Miasto i Klient oraz między tabelami Region i PrzedstawicielHandlowy. Tabela Region nie może się łączyć z tabelą faktów, ponieważ każde łączenie w wymiarze jest łączeniem 1:wielu z tabelą Region. Również tabela PrzedstawicielHandlowy nie może być tabelą wymiarów podstawowych ze względu na łączenia 1:wielu między tabelami Region i Miasto oraz tabelami Miasto i Klient.

### Wymiar nienadający się do optymalizacji



Rysunek 37. Niepoprawny wymiar. Zestaw tabel wymiarów użytych w wymiarze, którego nie można zoptymalizować

## Parametry Doradcy optymalizacyjnego

Informacje określone dla każdego parametru Doradcy optymalizacyjnego wpływają na tabele podsumowań zalecane przez kreatora oraz na uzyskiwaną poprawę wydajności. Należy podawać dokładne informacje oraz dokonywać rozważnego wyboru między kosztem a wymaganą wydajnością.

### Oczekiwane działanie zapytania

Użytkownik określa typ zapytań, pod kątem których ma być zoptymalizowany model kostki. Typy zapytań określają, kiedy i w jaki sposób zazwyczaj uzyskiwany jest dostęp do relacyjnych danych DB2. Informacje te pomagają Doradcy optymalizacyjnemu stwierdzić, których części modelu kostki najczęściej dotyczą zapytania. Dla kostki można także określać wycinki optymalizacji, jeśli wiadomo, że zapytania dotyczą najczęściej kilku konkretnych wycinków.

Dla każdej kostki można podczas jej tworzenia zdefiniować typy zapytań. Za pomocą Doradcy optymalizacyjnego można przejrzeć, co zostało określone dla każdej kostki, i wprowadzić niezbędne zmiany, jeśli są potrzebne.

## Ograniczenia dotyczące miejsca na dysku

Użytkownik określa przybliżoną ilość miejsca na dysku, jaką można przeznaczyć na tabele podsumowań. Doradca optymalizacyjny nie jest w stanie ustalić dokładnej wielkości tabel podsumowań, zanim nie zostaną one zbudowane, więc zaleca tabele podsumowań możliwie najbliższe podanej ilości miejsca na dysku. Już zbudowane tabele podsumowań mogą zajmować mniej lub więcej miejsca, niż określił użytkownik.

Ilość miejsca na dysku, która zostaje określona, jest bezpośrednio powiązana z wynikami optymalizacji. Powiększenie miejsca na dysku może spowodować wzrost zarówno liczby zapytań o poprawioną wydajności, jak i stopnia poprawy. Określając wielkość miejsca na dysku, należy brać pod uwagę następujące czynniki:

- wymagany poziom wydajności zapytań,
- liczbę modeli kostek, dla których jest przeprowadzana optymalizacja,
- istotność poszczególnych modeli kostek,
- częstość korzystania z poszczególnych modeli kostek,
- dostępność i koszt miejsca na dysku.

Zazwyczaj znaczną poprawę można zaobserwować już wtedy, gdy udostępnia się umiarkowaną ilość miejsca na dysku, czyli od 1% do 10% miejsca obecnie zajmowanego przez tabele relacyjne, do których odwołuje się model kostki. Tabela 37 przedstawia związek między ilością miejsca na dysku przeznaczoną na tabele podsumowań a przewidywaną poprawą wydajności zapytań. Decydując o ilości udostępnianego miejsca, każdy model kostki należy rozważyć w kontekście wszystkich metadanych i tabel podstawowych.

*Tabela 37. Miejsce na dysku.* Procent zajętego miejsca na dysku i odpowiadające mu stopnie poprawy wydajności

Procent miejsca na dysku zajmowanego przez tabele podsumowań w stosunku do zajmowanego przez tabele podstawowe	Przewidywany stopień poprawy odpowiednich zapytań
Mniej niż 1%	Niski
5%	Średni
50%	Wysoki
Nieograniczony	Najwyższy

## Ograniczenia czasowe

Czas podany przez użytkownika to maksymalny czas, jaki Doradca optymalizacyjny ma na wyznaczenie zaleceń. Im dłużej będzie mógł działać Doradca optymalizacyjny, tym lepsze będą jego wyniki. W poniższej tabeli zawarto przybliżone wytyczne dotyczące tego, jak długo powinien działać Doradca optymalizacyjny. Uzyskana wydajność bywa różna i może się okazać konieczne określenie większej ilości czasu niż podaje Tabela 38.

*Tabela 38. Ograniczenia czasowe.* Wytyczne dotyczące ilości czasu, jaki należy przeznaczyć dla Doradcy optymalizacyjnego

Przebieg optymalizacji bazy danych	Przybliżone ograniczenie czasowe
Brak próbkowania danych	5 do 30 minut
Próbkowanie danych w małej bazie danych, o wielkości poniżej 10 gigabajtów	1 godzina lub mniej
Próbkowanie danych w dużej bazie danych, o wielkości przekraczającej 10 gigabajtów	Kilka godzin



## Próbkowanie danych

Próbkowanie danych umożliwia Doradcy optymalizacyjnemu badanie danych w modelu kostki. W ten sposób Doradca optymalizacyjny uzyskuje więcej informacji, dzięki czemu może tworzyć efektywniejszy zbiór zaleceń. Zalecenia opracowane z zastosowaniem próbkowania danych będą lepiej dostosowane do określonej ilości miejsca na dysku. Przy braku próbkowania danych Doradca optymalizacyjny w celu określenia zaleceń przeanalizuje tylko metadane i statystykę DB2.

---

## Optymalizowanie modelu kostki

Optymalizując zapytania wykonywane na modelu kostki, można zwiększyć wydajność programów zgłaszających zapytania SQL typu OLAP.

**Wymaganie wstępne:** Tabele podstawowe używane w modelu kostki muszą mieć określone ograniczenia DB2. Ograniczenia muszą być zdefiniowane między każdą tabelą faktów i tabelą wymiarów oraz między wszystkimi tabelami wymiarów w schemacie typu płatek śniegu. Te ograniczenia muszą być określone dla kolumn niedopuszczających wartości pustych. Więcej informacji na temat definiowania ograniczeń zawiera sekcja “Definicje ograniczeń optymalizacyjnych” na stronie 107.

Podczas optymalizowania modelu kostki Doradca optymalizacyjny tworzy kod SQL, którego można użyć do budowania zestawu zalecanych tabel podsumowań dla danego modelu kostki. W tabelach podsumowań agregowane są najczęściej udostępniane dane, co pozwala na przyspieszenie realizacji zapytań.

Aby zoptymalizować model kostki:

1. Otwórz Doradcę optymalizacyjnego, prawym przyciskiem myszy klikając model kostki w drzewie obiektów Centrum OLAP, a następnie klikając opcję **Doradca optymalizacyjny**.
2. Na stronie Typy zapytań przejrzyj typy zapytań, pod kątem których ma być optymalizowana każda z kostek. Można zmodyfikować typ zapytania lub określić wycinki optymalizacji dla danej kostki. Na podstawie typu zapytań kostki zwiększana jest efektywność optymalizacji. Informacje na temat wycinków zapytań zawiera sekcja “Wycinki optymalizacji dla kostek” na stronie 95.
3. Na karcie Tabele podsumowań określ, czy aktualizowanie tabel podsumowań ma być natychmiastowe czy odroczone. Informacje na temat opcji aktualizacji zawiera sekcja “Aktualizacja tabel podsumowań” na stronie 119. Określ, w którym obszarze tabel mają być przechowywane tabele podsumowań oraz indeksy tabel podsumowań.
4. Na karcie Ograniczenia określ, ile miejsca na dysku mogą zająć tabele podsumowań i indeksy, które zostaną zbudowane. Należy też zadeklarować, czy dozwolone będzie próbkowanie danych. Określ także maksymalną ilość czasu, przez jaką Doradca optymalizacyjny może przygotowywać zalecenia. Im więcej miejsca, informacji i czasu będzie miał doradca, tym bardziej znacząco poprawi się wydajność. Więcej informacji na temat określania parametrów dla Doradcy optymalizacyjnego zawiera sekcja “Parametry Doradcy optymalizacyjnego” na stronie 111.

Jest to ostatnia strona parametrów wejściowych Kreatora Doradcy optymalizacyjnego. Kliknij przycisk **Dalej**, aby otworzyć okno postępu, umożliwiające monitorowanie (a w razie potrzeby zatrzymanie) postępu Doradcy optymalizacyjnego w tworzeniu zaleceń dotyczących tworzenia i odświeżania tabel podsumowań. Można zmodyfikować ograniczenia czasowe Doradcy optymalizacyjnego. Kliknięcie przycisku **Zatrzymaj** spowoduje, że Doradca optymalizacyjny zwróci zalecenia dotyczące tabel podsumowań określone w danym czasie.

5. Na stronie Skrypty SQL wpisz unikalną nazwę pliku w polu **Skrypt SQL do tworzenia tabel podsumowań** oraz unikalną nazwę pliku w polu **Skrypt SQL do odświeżania tabel podsumowań**.
6. Kliknij przycisk **Zakończ**, aby zapisać skrypty SQL w plikach o podanych nazwach.
7. Uruchom skrypty SQL. Tworzenie dużych tabel podsumowań może być czasochłonne. Do uruchomienia skryptów SQL można użyć Centrum komend DB2 lub okna komend. Aby skrypty SQL uruchomić z okna komend programu DB2:
  - a. Przejdź do katalogu, w którym zostały zapisane skrypty SQL.
  - b. Nawiąż połączenie z bazą danych zoptymalizowanego modelu kostki. Na przykład wprowadź komendę: `db2 connect to CVSAMPLE`.
  - c. Wprowadź następującą komendę:  
`db2 -tvf nazwa_pliku`

gdzie *nazwa\_pliku* jest nazwą skryptu SQL do tworzenia tabel podsumowań.

---

## Przykład skryptu SQL do tworzenia tabel podsumowań

Doradca optymalizacyjny generuje skrypt SQL do tworzenia zalecanych tabel podsumowań. Skrypt ten zawiera instrukcje SQL niezbędne do zbudowania jednej lub wielu tabel podsumowań.

Rys. 38 na stronie 115 przedstawia fragment przykładowego skryptu SQL tworzącego jedną tabelę podsumowań i pokazuje odwzorowanie obiektów metadanych na SQL. W przykładowym skrypcie SQL tabela podsumowań nosi nazwę DB2INFO.MQT000000021T01, gdzie 21 to identyfikator modelu kostki, a T01 to identyfikator tabeli podsumowań. Identyfikator modelu kostki może się składać maksymalnie z 10 cyfr. Identyfikator tabeli podsumowań identyfikuje tabelę podsumowań w modelu kostki. Identyfikator tabeli podsumowań pozwala na użycie maksymalnie 99 tabel podsumowań w jednym modelu kostki. Nie należy zmieniać nazwy tabeli podsumowań zdefiniowanej przez Doradcę optymalizacyjnego. Zmiana nazw tabel uniemożliwia programowi DB2 Cube Views rozpoznawanie tabel podsumowań tworzonych dla modelu kostki.

```

DROP TABLE DB2INFO.MQT0000000021T01;
UPDATE COMMAND OPTIONS USING c OFF;
CREATE SUMMARY TABLE DB2INFO.MQT0000000021T01 AS
(
  (SELECT
    SUM(T1."SALES") AS "SALES",
    SUM(T1."COGS") AS "COGS",
    SUM(T1."ADVERTISING") AS "ADVERTISING",
    SUM(T1."COGS" + T1."ADVERTISING") AS "Wydatki łącznie",
    SUM(T1."SALES" - (T1."COGS" + T1."ADVERTISING")) AS "Zysk",
    T5."REGION_NAME" AS "REGION_NAME",
    T5."STATE_NAME" AS "STATE_NAME",
    T6."LINEID" AS "LINEID (LINE)",
    T4."YEAR" AS "YEAR",
    T4."QUARTER_NUMBER" AS "QUARTER_NUMBER",
    'Qtr ' || CONCAT (cast (T4."QUARTER_NUMBER" AS CHAR(1))) AS "QUARTER_NAME",
    T4."MONTH_NUMBER" AS "MONTH_NUMBER"
  FROM "CVSAMPLE"."SALESFACT" AS T1,
    "CVSAMPLE"."STORE" AS T2,
    "CVSAMPLE"."PRODUCT" AS T3,
    "CVSAMPLE"."TIME" AS T4,
    "CVSAMPLE"."LOCATION" AS T5,
    "CVSAMPLE"."LINE" AS T6
  WHERE
    T1."STOREID"=T2."STOREID" AND T1."PRODUCTID"=T3."PRODUCTID" AND
    T1."TIMEID"=T4."TIMEID" AND T2."POSTALCODEID"=T5."POSTALCODEID"
    AND T3."LINEID"=T6."LINEID"
  GROUP BY
    T5."REGION_NAME",
    T5."STATE_NAME",
    T6."LINEID",
    T4."YEAR",
    T4."QUARTER_NUMBER",
    'Qtr ' || CONCAT (cast (T4."QUARTER_NUMBER" AS CHAR(1))),
    T4."MONTH_NUMBER")
  DATA INITIALLY DEFERRED
  REFRESH DEFERRED
  ENABLE QUERY OPTIMIZATION
  MAINTAINED BY SYSTEM
  NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000021T01 IS 'AST created for cube model CVSAMPLE.Sales Model';
COMMIT;

ALTER TABLE DB2INFO.MQT0000000021T01 ACTIVATE NOT LOGGED INITIALLY;
REFRESH TABLE DB2INFO.MQT0000000021T01;

CREATE INDEX DB2INFO.IDX0000000021T0101 ON DB2INFO.MQT0000000021T01("STATE_NAME");
CREATE INDEX DB2INFO.IDX0000000021T0102 ON DB2INFO.MQT0000000021T01("MONTH_NUMBER");
CREATE INDEX DB2INFO.IDX0000000021T0103 ON DB2INFO.MQT0000000021T01("LINEID (LINE)") CLUSTER;
COMMIT;

REORG TABLE DB2INFO.MQT0000000021T01;
RUNSTATS ON TABLE DB2INFO.MQT0000000021T01 AND INDEXES ALL;
COMMIT;

```

Odwzorowuje miary

Odwzorowuje atrybuty

Tabele faktów i wymiarów

Łączy tabele faktów i wymiarów.  
W razie potrzeby łączy wymiar z tabelami wymiarów w schemacie typu płatek śniegu

Odwzorowuje jeden wycinek modelu kostki

Tworzy indeksy

### Rysunek 38. Przykładowy skrypt SQL

Jeśli dla danego modelu kostki zalecanych jest kilka tabel podsumowań, skrypt SQL do tworzenia tabel podsumowań będzie zawierał cały zbiór takich instrukcji dla każdej tabeli podsumowań.

W kolejnych sekcjach omawiane są instrukcje zawarte w przykładowym skrypcie SQL do tworzenia tabel podsumowań:

#### Instrukcja DROP TABLE

Każda tabela podsumowań, która ma zostać utworzona, najpierw jest usuwana, aby nie istniała tabela o takiej samej nazwie. Na Rys. 38 usuwana jest tabela DB2INFO.MQT0000000021T01 za pomocą instrukcji: DROP TABLE DB2INFO.MQT0000000021T01;

#### Instrukcja CREATE TABLE

Skrypt tworzy tabelę podsumowań przy użyciu instrukcji CREATE TABLE. Ta instrukcja stanowi największą część skryptu i składa się z instrukcji SELECT z klauzulami SELECT, FROM, WHERE i GROUP BY oraz z definicji metody aktualizacji. Tabela podsumowań jest tworzona z odpowiednimi kolumnami, ale bez danych. Skrypt odświeżający tabele podsumowań agreguje dane z tabel faktów i tabel wymiarów, po czym zapełnia tabelę podsumowań.

Nazwa tabeli jest zdefiniowana w pierwszym wierszu instrukcji CREATE TABLE: CREATE SUMMARY TABLE DB2INFO.MQT000000021T01.

Klauzula SELECT pokazana na Rys. 38 na stronie 115 składa się z pięciu wierszy zaczynających się od słowa SUM. Każdy z tych wierszy odwzorowuje jedną z miar modelu kostki. Na przykład wiersz SUM(T1."SPRZEDAŻ"-(T1."KST"+T1."REKLAMA")) AS "Zysk" jest odwzorowaniem miary obliczanej Zysk na funkcję agregującą SUM. Model kostki, dla której tworzona jest tabela podsumowań, ma następujące miary: Sprzedaż, Koszt sprzedanych towarów, Reklama, Wydatki łącznie, Zysk. Następnymi siedem wierszy, które wybierają kolumnę bez wykonywania jakichkolwiek obliczeń, odwzorowuje atrybuty. Na przykład wiersz T5."NAZWA\_REGIONU" AS "NAZWA\_REGIONU" jest odwzorowaniem atrybutu Nazwa regionu. Ta tabela podsumowań zawiera następujące atrybuty z modelu kostki: Nazwa regionu, Nazwa stanu, Identyfikator linii, Rok, Numer kwartału, Nazwa kwartału i Nazwa miesiąca.

Tabelami w klauzuli FROM są tabele faktów i wymiarów używane w modelu kostki. W tym przykładzie używane są tabele FaktySprzedaży, Sklep, Produkt, Czas, Lokalizacja i Linia.

Klauzula WHERE definiuje łączenia między tabelami faktów i wymiarów, a każde łączenie stanowi odwzorowanie obiektu łączenia w modelu kostki. Optymalizowany model kostki jest oparty na schemacie typu płatek śniegu, zatem w klauzuli WHERE są również uwzględnione łączenia wymiar-wymiar.

Klauzula GROUP BY odwzorowuje wycinki zdefiniowane dla modelu kostki. Rys. 38 na stronie 115 przedstawia jeden zestaw grupowania zapewniający odwzorowanie na konkretny wycinek. Definiując wycinek, w grupowaniach można uwzględnić następujące typy metadanych:

- atrybuty klucza poziomu z hierarchii na poziomie wycinka
- atrybuty klucza poziomu znajdujące się powyżej poziomu wycinka
- atrybuty pokrewne, które nie są funkcjonalnie zależne od atrybutu klucza poziomu

Ta sekcja skryptu SQL może zawierać klauzule GROUPING SETS, aby tabela podsumowań mogła zawierać wiele poziomów. Jeśli model kostki zawiera miary niedystrybuowalne, ta sekcja skryptu SQL może zawierać klauzulę ROLLUP.

W przedstawionym przykładzie optymalizowany model kostki ma następujące hierarchie: Rynek [Region, Stan, Miasto, Kod pocztowy, Sklep], Produkt [Rodzina, Linia, Produkt], Czas [Rok, Kwartał, Miesiąc, Dzień] i Czas fiskalny [Rok fiskalny, Kwartał fiskalny, Miesiąc fiskalny]. Jeśli poziom z hierarchii nie jest uwzględniony w zestawie grupowania, to wycinek jest na najwyższym poziomie, takim jak Cały czas, Wszystkie regiony lub Wszystkie produkty. Wycinek w klauzuli GROUP BY jest wycinkiem Stan–Linia–Miesiąc i obejmuje następujące atrybuty: Nazwa regionu, Rok, Numer kwartału i Nazwa kwartału. Nazwa regionu jest powyżej poziomu Stan, a atrybuty Rok, Numer kwartału i Nazwa kwartału są powyżej poziomu Miesiąc. Każdy z atrybutów poziomu w wycinku to atrybut, który jest odwzorowywany przez klauzulę SELECT.

Ostatnią częścią instrukcji CREATE TABLE jest definicja metody aktualizacji. Na Rys. 38 na stronie 115 trzy ostatnie wiersze instrukcji CREATE TABLE określają tabelę podsumowań jako odświeżaną w sposób odroczoney:

```
DATA INITIALLY DEFERRED
REFRESH DEFERRED
ENABLE QUERY OPTIMIZATION
MAINTAINED BY SYSTEM
NOT LOGGED INITIALLY;
```

Gdyby tabela podsumowań była definiowana jako odświeżana w sposób natychmiastowy, instrukcje miałyby następującą postać:

```
DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
ENABLE QUERY OPTIMIZATION
MAINTAINED BY SYSTEM
NOT LOGGED INITIALLY;
```

### Instrukcje CREATE INDEX

Doradca optymalizacyjny zaleca co najmniej jeden indeks dla tabeli podsumowań, indeksy są tworzone po utworzeniu tabeli podsumowań. W skrypcie na Rys. 38 na stronie 115 tworzone są zarówno indeksy klastrowe, jak i nieklastrowe. Gdy indeksy zostaną utworzone, instrukcja REORG reorganizuje tabelę odpowiednio do indeksu klastrowego. W niektórych wypadkach może to poprawić wydajność czytania tabeli.

### Instrukcja RUNSTATS

Gdy wszystkie zalecane elementy tabeli podsumowań zostaną utworzone, instrukcja RUNSTATS aktualizuje statystykę, na podstawie której optymalizator DB2 ustala tabele podsumowań i indeksy do przekierowywania zapytań.

---

## Testowanie wyników zapytania

Korzystając z narzędzia do testów porównawczych, db2batch, dostępnego w programie DB2 Universal Database, można przy użyciu Doradcy optymalizacyjnego testować wydajność zapytań przed utworzeniem tabel podsumowań oraz po ich utworzeniu.

Aby przetestować wydajność zapytań:

1. Utwórz plik wejściowy zawierający zapytania, które mają zostać przetestowane, oddzielone średnikami.
2. W wierszu komend wprowadź następującą komendę:

```
db2batch -d nazwa_bazy_danych -f nazwa_pliku -cli
```

gdzie *nazwa\_bazy\_danych* określa bazę danych, względem której mają być uruchamiane zapytania, *nazwa\_pliku* określa plik wejściowy z zapytaniami SQL, natomiast *-cli* określa, że proces ma przebiegać w trybie CLI. Narzędzie db2batch opracowuje wynik badania wydajności i przedstawia zarówno średnie arytmetyczne, jak i geometryczne. Aby uzyskać informacje o składni i opcjach, w wierszu komend należy wprowadzić komendę db2batch -h. Więcej informacji na temat narzędzia do testów porównawczych db2batch i tworzenia tego typu testów zawiera Centrum informacyjne DB2.

Jeśli po utworzeniu zalecanych tabel podsumowań wydajność jest satysfakcjonująca, nie trzeba przeprowadzać żadnych dodatkowych analiz.

Jeśli poprawa zapytań jest mniejsza niż oczekiwana, można ponownie uruchomić Doradcę optymalizacyjnego, udostępniając więcej miejsca na dysku i więcej czasu oraz włączając próbkowanie danych, jeśli nie zostało włączone wcześniej. Udostępnienie większej ilości miejsca na dysku jest czynnikiem, który przypuszczalnie będzie mieć największy wpływ na wydajność. Im więcej miejsca dysku zostanie przeznaczony na tabele podsumowań, tym większą poprawę będzie można zaobserwować. Jeśli kreator będzie mógł próbować dane, będzie też mógł podać lepsze zalecenia. Podobnie, im więcej czasu otrzyma kreator na ustalanie zaleceń, tym lepsze powinny być te zalecenia.

Jeśli wydajność nie jest satysfakcjonująca, ponieważ działanie zapytań w ogóle się nie poprawiło lub poprawiło się tylko w niewielkim stopniu, lub jeśli zapytania działają satysfakcjonująco przez pewien czas, a potem ich wydajność spada, zapoznaj się z omówieniem pt. "Rozwiązywanie problemów z tabelami podsumowań".

---

## Rozwiązywanie problemów z tabelami podsumowań

Jeśli wydajność zapytań nie zwiększa się po utworzeniu tabel podsumowań, w celu rozwiązania problemów z kierowaniem zapytań można się posłużyć narzędziem DB2EXPLAIN.

Aby można było użyć programu DB2EXPLAIN do sprawdzenia, czy w programie DB2 UDB są używane tabele podsumowań, najpierw należy:

- Sprawdzić, czy dane statystyczne tabel podstawowych i tabel podsumowań są aktualne.
- Rozpoznać, które zapytania mają niezadowalającą wydajność, jeśli jeszcze ich nie zidentyfikowano. Do wychwytywania wolno działających zapytań można wykorzystać Monitor obrazów stanu SQL DB2.

Aby określić, dlaczego zapytania mają wydajność gorszą od oczekiwanej:

1. Utwórz tabele wyjaśnienia. Aby utworzyć tabele wyjaśnienia dla bazy danych, nawiąż połączenie z bazą danych i z katalogu \SQLLIB\misc uruchom następującą komendę:  
db2 -tvf explain.dd1
2. Przeprowadź działania w programie narzędziowym explain. Gdy tryb wyjaśnienia jest włączony, zapytania SQL nie działają i po wywołaniu komendy explain przetwarzane są tylko żądania informacji. Uruchom poniższe komendy SQL, aby włączyć tryb wyjaśnienia, określić okres odświeżania, tak aby program DB2 uwzględnił tabele podsumowań, jeśli są odświeżane w sposób odroczone, uruchomić zapytanie, wyłączyć tryb wyjaśnienia oraz skierować zapytanie do tabeli wyjaśnienia, aby dowiedzieć się, czy zapytanie zostało przekierowane.

```
set current explain mode explain
```

```
set current refresh age any
```

```
SELECT SUM(SALES) FROM MDSAMPLE.SALESFACT
```

```
set current explain mode no
```

```
SELECT EXPLAIN_TIME, EXPLAIN_LEVEL AS "LEV",  
       QUERYNO, STATEMENT_TEXT  
FROM EXPLAIN_STATEMENT  
WHERE STATEMENT_TEXT LIKE '%SALESFACT%'  
ORDER BY EXPLAIN_TIME
```

3. Przejrzyj informacje wyjaśniające i sprawdź, czy przeformułowane zapytanie jest przekierowywane do tabeli podsumowań. Może zostać wyświetlony raport podobny do następującego:

```
2002-06-30-23.22.12.325002 O 11 SELECT SUM(SALES)  
      FROM MDSAMPLE.SALESFACT  
2002-06-30-23.22.12.325002 P 11 SELECT Q3.$C0  
      FROM (SELECT SUM(Q2.$C0) FROM (SELECT Q1.SALESFACT_SALES  
      FROM DB2INFO.MQT0000000021T01 AS Q1) AS Q2) AS Q3
```

Na jedno wykonanie zapytania przypadają dwa wiersze. Wiersz oznaczony literą O to oryginalne zapytanie wysłane do programu DB2 UDB. Wiersz oznaczony literą P to zapytanie przebudowane przez optymalizator DB2. Na podstawie przebudowanego zapytania w tym przykładzie widać, że optymalizator DB2 wybrał dane z tabeli podsumowań DB2INFO.MQT0000000021T01.

Jeśli zapytanie jest przekierowywane do tabeli podsumowań, lecz jest mniej wydajne niż oczekiwano, potrzebne może być ponowne uruchomienie Doradcy optymalizacyjnego z innymi opcjami.

Jeśli zapytanie nie jest przekierowywane do tabeli podsumowań, określ tego przyczynę i podejmij odpowiednie działania. Wśród przyczyn, dla których zapytanie może nie być przekierowywane do tabeli podsumowań, można wymienić następujące:

#### **Tabela podsumowań nie istnieje**

Najpierw upewnij się, że tabela podsumowań istnieje. Jeśli nie istnieje, uruchom Doradcę optymalizacyjnego, aby wygenerował skrypt SQL do tworzenia tabel podsumowań. Następnie uruchom ten skrypt, aby table podsumowań utworzyć.

#### **Tabela podsumowań z odświeżaniem odroczonym utraciła ważność**

Jeśli tabela podsumowań istnieje i została określona jako odświeżana w sposób odroczony, może być konieczna aktualizacja okresu odświeżania. Okres odświeżania tabeli można ustawić na największy możliwy i niezależny od sesji, stosując przypisanie (DFT\_REFRESH\_AGE) = 99999999999999 (ANY).

#### **Zapytanie sięga do danych, które nie są uwzględnione w tabeli podsumowań**

Jeśli zapytanie sięga do danych, których nie ma w tabeli podsumowań, optymalizator DB2 nie przekieruje tego zapytania. Jeśli już po utworzeniu tabel podsumowań zostanie dodana nowa miara, nie będzie jej w tabelach podsumowań. Jeśli zapytanie będzie się odwoływać do tej nowej miary, optymalizator DB2 nie będzie mógł przekierować tego zapytania do tabeli podsumowań, ponieważ nie będzie ona zawierać wszystkich danych spełniających warunki zapytania.

Ponadto, jeśli spróbujemy zrealizować zapytanie względem danych znajdujących się poniżej tego wycinka modelu kostki, dla którego tabela podsumowań jest zbudowana, tabeli podsumowań nie będzie można użyć. Na przykład, jeśli w zapytaniu zawarte jest żądanie danych zagregowanych na poziomie Miasto, a tabela podsumowań zawiera dane, które są zagregowane na poziomie Stan, znajdującym się wyżej od poziomu Miasto, zapytanie to nie może skorzystać z tabeli podsumowań.

#### **Zapytanie zawiera konstrukcje, których nie można przekierować**

Optymalizator DB2 nie może przekierowywać zapytań, w których używane są niektóre złożone konstrukcje. Takimi złożonymi konstrukcjami uniemożliwiającymi optymalizatorowi DB2 przekierowywanie zapytań są funkcje rekurencyjne i funkcje odwołujące się do właściwości fizycznych, jak na przykład:

- NODENUMBER,
- łączenia zewnętrzne,
- unie,
- XMLAGG,
- okienkowe funkcje agregujące, czyli funkcje agregujące z klauzulą OVER.

---

## **Aktualizacja tabel podsumowań**

Gdy dane w tabelach podstawowych ulegną zmianie, należy zaktualizować table podsumowań. Tabele podsumowań można aktualizować na dwa różne sposoby: wykonując odświeżanie natychmiastowe albo odświeżanie odroczone.

Po uruchomieniu kreatora Doradcy optymalizacyjnego należy określić, czy tworzone table podsumowań mają być poddawane odświeżaniu natychmiastowemu czy odroczone.

Dokonany wybór wpływa na ustawienie aktualizacji dla tych tabel i skrypt SQL do odświeżania tabel podsumowań. W obydwu przypadkach skrypt do odświeżania tabel podsumowań trzeba uruchamiać w ramach harmonogramu normalnej konserwacji bazy danych. Uruchomienie skryptu do odświeżania może wymagać znacznej ilości czasu i zasobów. W oknie zadań wsadowych konserwacji należy przydzielić dość czasu na zrealizowanie aktualizacji.

### **Odświeżanie natychmiastowe**

Tabele podsumowań odświeżane natychmiastowo pozostają dobrze zsynchronizowane z tabelami podstawowymi. Program DB2 UDB śledzi zmiany w tabelach podstawowych i aktualizuje tabele podsumowań aktualizować przyrostowo, dokonując zmian tylko w tej ich części, która odpowiada zmienionej części tabel podstawowych. Jeśli dla użytkownika jest ważne, aby dane tabeli podsumowań stałe były w pełni zgodne z tabelami podstawowymi, należy użyć opcji odświeżania natychmiastowego. Odświeżanie natychmiastowe może być korzystne, jeśli tabele podstawowe są na przykład aktualizowane na podstawie tygodniowych danych sprzedaży, a użytkownicy sporządzają tygodniowe raporty odzwierciedlające zaktualizowane dane sprzedaży.

Jeśli w tabelach podstawowych często wprowadzanych jest wiele drobnych zmian, to odświeżanie natychmiastowe nie jest najlepszym rozwiązaniem, ponieważ może wymagać znaczącego nakładu pracy ze strony programu DB2 UDB w związku ze śledzeniem zmian i jednostkowym wykonywaniem instrukcji aktualizacji w celu ponownego agregowania zmian.

Jeśli tabele podstawowe są aktualizowane przy użyciu zwykłych instrukcji SQL, takich jak INSERT, UPDATE czy DELETE, program DB2 UDB automatycznie synchronizuje odpowiednie tabele podsumowań po wprowadzeniu zmian do tabel podstawowych. Jeśli jednak tabele podstawowe są aktualizowane przy użyciu komend LOAD lub IMPORT programu DB2, synchronizację trzeba wymusić ręcznie, uruchamiając skrypt odświeżania po wykonaniu aktualizacji.

Nie we wszystkich sytuacjach można korzystać z odświeżania natychmiastowego, a kreator Doradcy optymalizacyjnego może w razie potrzeby zalecić odświeżanie odroczone.

### **Odświeżanie odroczone**

Tabele podsumowań odświeżane w sposób odroczone zwykle są aktualizowane rzadziej niż odświeżane natychmiastowo, ponieważ muszą być ręcznie zsynchronizowane z tabelami podstawowymi. Tabele podsumowań bazują na obrazie danych z chwili, w której powstały. Każdy proces aktualizacji powoduje ponowne utworzenie tabeli podsumowań na podstawie bieżących danych, ale brak informacji o tym, jak dane zmieniły się od czasu ostatniego utworzenia tabeli podsumowań.

Odświeżanie odroczone jest korzystne, gdy dokonywane są liczne zmiany rozproszone po odpowiednich tabelach podstawowych, lub jeśli dane są aktualizowane częściej, niż odczytywane. Na przykład, jeśli dane sprzedaży są aktualizowane tygodniowo, lecz raport musi być sporządzany w odstępach kwartalnych, można użyć opcji odświeżania odroczonego i przebudowywać tabele podsumowań co kwartał, przed wykonaniem raportu.

---

## **Usuwanie tabeli podsumowań**

Gdy zostaje usunięty model kostki, program DB2 Cube Views nie usuwa skojarzonych z nim tabel podsumowań. Jeśli tabele podsumowań nie są już używane do żadnego innego celu, można je usunąć, aby zwolnić miejsce na dysku.



Tabele podsumowań można usuwać przy użyciu normalnych procedur programu DB2, korzystając z Centrum sterowania lub z wiersza komend. Wraz z tabelą podsumowań zostają usunięte wszystkie skojarzone z nią indeksy.

Tabele podsumowań są zdefiniowane w schemacie DB2INFO. W nazwie tabeli podsumowań zawarty jest identyfikator modelu kostki. Tabela podsumowań może mieć na przykład nazwę DB2INFO.MQT000000021T01, gdzie 21 to identyfikator modelu kostki, a T01 służy do jednoznacznej identyfikacji tabeli podsumowań w modelu kostki. Identyfikator modelu kostki może się składać maksymalnie z 10 cyfr.

Aby usunąć tabelę podsumowań za pomocą wiersza komend, wprowadź komendę `DROP TABLE nazwa_tabeli`.



---

## Rozdział 6. Program DB2 Cube Views a stowarzyszone źródła danych

W tej sekcji omówiono następujące tematy:

### **Przegląd systemów stowarzyszonych**

Program IBM DB2 Information Integrator może być wykorzystywany jako system zintegrowanej obsługi danych na poziomie przedsiębiorstwa. Program DB2 Information Integrator to zbiór mechanizmów, scalający w ramach jednej platformy systemy zarządzania danymi, systemy stowarzyszone i kilka innych technologii.

### **Optymalizowanie zdalnych źródeł danych za pomocą programu DB2 Cube Views - przegląd**

Program DB2 Cube Views umożliwia optymalizowanie schematu gwiazdowego lub schematu typu płatek śniegu pod kątem poprawy wydajności zapytań.

### **Umożliwianie obsługi systemu stowarzyszonego przez program DB2 Cube Views**

Aby umożliwić obsługę systemu stowarzyszonego przez program DB2 Cube Views, należy zdefiniować zdalne źródło danych, zdefiniować pseudonimy na serwerze stowarzyszonym oraz zdefiniować ograniczenia informacyjne dla pseudonimów.

### **Rozwiązywanie problemów z przekierowywaniem zapytań dla stowarzyszonych źródeł danych**

Jeśli wydajność zapytań skierowanych do zdalnego źródła danych nie poprawia się po optymalizacji zgodnie z oczekiwaniami, należy się upewnić, że system stowarzyszony jest skonfigurowany poprawnie pod kątem programu DB2 Cube Views.

---

## Przegląd systemów stowarzyszonych

Program IBM DB2 Information Integrator może być wykorzystywany jako system zintegrowanej obsługi danych na poziomie przedsiębiorstwa. Program DB2 Information Integrator to zbiór mechanizmów, scalający w ramach jednej platformy systemy zarządzania danymi, systemy stowarzyszone i kilka innych technologii.

System stowarzyszony DB2 jest specjalnym rodzajem systemu zarządzania rozproszoną bazą danych (DBMS). Jako jedna z technologii integrowania informacji, systemy stowarzyszone mają ułatwiać dostęp do informacji różnego typu, pochodzących z rozmaitych źródeł. System stowarzyszony składa się z następujących elementów:

- instancja DB2 działająca jako serwer stowarzyszony
- baza danych pełniąca rolę stowarzyszonej bazy danych
- przynajmniej jedno źródło danych
- klienci (użytkownicy i aplikacje) korzystający z bazy danych i źródeł danych

W systemie stowarzyszonym istnieje możliwość wysyłania żądań rozproszonych do wielu źródeł danych za pomocą jednej instrukcji SQL. Na przykład jedna instrukcja SQL może odnosić się łącznie do danych w tabeli DB2 Universal Database, w tabeli Oracle oraz w pliku XML.

### **Serwery stowarzyszone**

Serwer DB2 w systemie stowarzyszonym nazywany jest serwerem stowarzyszonym. Do pracy w charakterze serwerów stowarzyszonych można skonfigurować dowolną liczbę instancji DB2. Mogą to być istniejące instancje DB2 lub nowe instancje, utworzone specjalnie w tym celu.

Instancja DB2 zarządzająca systemem stowarzyszonym jest nazywana serwerem, ponieważ odpowiada ona na żądania kierowane przez użytkowników i aplikacje klienckie. Serwer stowarzyszony często przesyła fragmenty otrzymanych zapytań do źródeł danych, zlecając im ich przetwarzanie. Operacja przekazana do źródła jest to operacja wykonywana na zdalnym serwerze. Instancja DB2 zarządzająca systemem stowarzyszonym jest nazywana serwerem stowarzyszonym mimo że delegując żądania do źródeł danych sama pełni rolę klienta.

Serwer stowarzyszony komunikuje się ze źródłami danych za pośrednictwem opakowań. Serwer stowarzyszony korzysta z procedur zapisanych w bibliotece nazywanej modułem opakowującym, która stanowi implementację opakowania. Procedury te umożliwiają serwerowi stowarzyszonemu wykonywanie takich operacji, jak łączenie się ze źródłem danych i iteracyjne odczytywanie z niego danych. Z reguły właściciel instancji serwera stowarzyszonego DB2 używa instrukcji CREATE WRAPPER do zarejestrowania opakowania w stowarzyszonej bazie danych.

### **Stowarzyszone bazy danych**

Z punktu widzenia użytkowników i aplikacji klienckich źródła danych widoczne są jako pojedyncza, zbiorcza baza danych DB2 UDB. Użytkownicy i aplikacje komunikują się ze stowarzyszoną bazą danych, którą zarządza serwer stowarzyszony. W stowarzyszonej bazie danych zawarty jest katalog systemowy. Katalog systemowy stowarzyszonej bazy danych zawiera wpisy identyfikujące źródła danych i opisujące ich cechy. Serwer stowarzyszony korzysta z informacji zapisanych w katalogu systemu stowarzyszonej bazy danych oraz w opakowaniach źródeł danych w celu ustalenia najlepszego planu przetwarzania instrukcji SQL.

### **Źródła danych**

W systemie stowarzyszonym źródłem danych może być instancja relacyjnego systemu zarządzania bazą danych (DB2, Informix, Oracle lub Sybase) lub nierelacyjne źródło danych (jak algorytm wyszukiwania BLAST lub plik XML).

Metoda lub protokół używane podczas dostępu do źródła danych zależą od typu źródła danych. Na przykład dostęp do źródeł z rodziny produktów DB2, takich jak DB2 for z/OS and OS/390 jest realizowany za pośrednictwem mechanizmu DRDA.

### **Klienci**

Klientem może być użytkownik lub aplikacja korzystająca ze stowarzyszonej bazy danych i skonfigurowanych w niej źródeł danych. Przykładem aplikacji odwołującej się jako klient do stowarzyszonej bazy danych i źródła danych może być program DB2 Cube Views.

---

## **Optymalizowanie zdalnych źródeł danych za pomocą programu DB2 Cube Views - przegląd**

Program DB2 Cube Views umożliwia optymalizowanie schematu gwiazdowego lub schematu typu płatek śniegu pod kątem poprawy wydajności zapytań.

Dzięki temu możliwe jest między innymi:

- korzystanie ze zintegrowanej platformy DB2 zapewniającej dostęp do wielu produktów IBM i innych producentów

- poszerzenie zakresu funkcjonalności programu DB2 UDB dla systemów Linux, UNIX i Windows do poziomu DB2 UDB for z/OS and DB2 UDB for iSeries
- zwiększenie wydajności zapytań kierowanych do stowarzyszonych baz danych i źródeł danych

W systemie stowarzyszonym tabele faktów i wymiarów mogą znajdować się na serwerze zdalnym lub na wielu serwerach zdalnych. Niezbędne jest utworzenie lokalnej reprezentacji schematu gwiazdzistego lub schematu typu płatek śniegu na użytek programu DB2 Cube Views. Lokalną reprezentację schematu gwiazdzistego lub schematu typu płatek śniegu można zdefiniować za pośrednictwem pseudonimów odwołujących się do tabel zdalnych, replikowanych kopii tabel zdalnych lub kombinacji pseudonimów i tabel replikowanych. Na serwerze lokalnym musi istnieć lokalna reprezentacja wszystkich tabel faktów i tabel wymiarów w schemacie gwiazdzistym lub schemacie typu płatek śniegu, zdefiniowana w jakikolwiek sposób.

Jeśli schemat gwiazdzisty lub schemat typu płatek śniegu jest już na lokalnym serwerze, można użyć programu DB2 Cube Views do zbudowania modelu kostki w oparciu o ten schemat, a także zoptymalizować model kostki przy użyciu Doradcy optymalizacyjnego. Jeśli niektóre tabele są lokalnie reprezentowane przez pseudonimy, włączenie opcji umożliwiającej Doradcy optymalizacyjnemu próbkowanie danych pozwala uzyskać znacznie lepsze zalecenia odnośnie tabel podsumowań, lecz wtedy generowanie zaleceń wymaga dłuższego czasu. Doradca optymalizacyjny nie w każdej sytuacji ma faktyczną możliwość użycia próbkowania danych, lecz mimo to włączanie tej opcji jest zalecane, jeśli tylko jest to możliwe.

Po utworzeniu zalecanych tabel podsumowań na serwerze lokalnym zapytanie kierowane do tabel w zdalnym schemacie gwiazdzistym lub typu płatek śniegu może być zrealizowane na jeden z trzech poniższych sposobów:

#### **Zapytanie jest kierowane do lokalnej tabeli podsumowań.**

Jeśli zapytanie może być zrealizowane w oparciu o dane w tabeli podsumowań, optymalizator DB2 skieruje zapytanie bezpośrednio do lokalnej tabeli podsumowań, a interakcja z tabelami zdalnymi nie będzie miała miejsca. Taki scenariusz zapewnia największą poprawę wydajności.

Często zrealizowanie zapytania wymaga nie tylko samej tabeli podsumowań, ale również jej połączenia z przynajmniej jedną tabelą wymiarów. W takim przypadku można rozważyć umieszczenie części lub wszystkich tabel wymiarów (ale nie tabel faktów) na serwerze stowarzyszonym, aby optymalizator DB2 miał możliwość uzupełnienia łączy przy użyciu tabel lokalnych.

#### **Zapytanie jest przekazywane niżej do zdalnego źródła danych**

Jeśli zrealizowanie zapytania w oparciu o dane w tabelach podsumowań i dołączonych do nich tabelach wymiarów nie jest możliwe, optymalizator DB2 podejmie próbę skierowania go na serwer zdalny. Zdalny serwer realizuje zapytanie i przekazuje wyniki na serwer lokalny.

Jeśli tabela wynikowa jest niewielka, nadal możliwa jest wyraźna poprawa wydajności. Zastosowanie tej metody jest możliwe, jeśli wszystkie tabele w zdalnym schemacie gwiazdzistym lub typu płatek śniegu rezydują na pojedynczym serwerze zdalnym.

#### **Zapytanie jest wykonywane lokalnie na danych pobranych ze zdalnych źródeł danych**

Jeśli nie jest możliwe wykonanie zapytania w oparciu o dane w tabelach podsumowań i dołączonych do nich tabelach wymiarów ani przekazanie zapytania do wykonania w zdalnym źródle danych, tabele niezbędne do zrealizowania zapytania są pobierane (kopiowane) z serwera zdalnego na serwer lokalny, a zapytanie jest wykonywane lokalnie. Na przykład jeśli schemat gwiazdzisty znajduje

się na zdalnym serwerze zSeries, a zapytanie korzysta z funkcji dostępnej w programie DB2 UDB, lecz nie w instancji DB2 for z/OS, jego zrealizowanie na zdalnym serwerze nie jest możliwe.

W tym scenariuszu poprawa wydajności może być nieosiągalna, jeśli tabele faktów i wymiarów, do których zapytanie się odwołuje, są duże.

---

## Umożliwianie obsługi systemu stowarzyszonego przez program DB2 Cube Views

Aby umożliwić obsługę systemu stowarzyszonego przez program DB2 Cube Views, należy zdefiniować zdalne źródło danych, zdefiniować pseudonimy na serwerze stowarzyszonym oraz zdefiniować ograniczenia informacyjne dla pseudonimów.

**Prerequisites:** Upewnij się, że włączona jest obsługa programu DB2 Information Integrator. Aby obsługę systemu stowarzyszonego włączyć z wiersza komend, wpisz: `db2 update dbm cfg using federated yes`. Po włączeniu obsługi systemu stowarzyszonego zatrzymaj program DB2 UDB i uruchom go ponownie.

Aby włączyć zdalne źródło danych dla programu DB2 Cube Views:

1. Zdefiniuj zdalne źródła danych.
2. Zdefiniuj pseudonimy dla zdalnych tabel.
3. Zdefiniuj ograniczenia informacyjne dla pseudonimów.

### Definiowanie zdalnych źródeł danych

Aby zdefiniować i skonfigurować serwer stowarzyszony pod kątem programu DB2 Cube Views, należy dostarczyć do serwera stowarzyszonego informacje o zdalnych źródłach danych i obiektach, do których ma być możliwy dostęp.

Aby zdefiniować zdalne źródło danych:

1. Wpisz do katalogu zdalną bazę danych i odpowiadający jej węzeł w katalogu bazy danych na serwerze stowarzyszonym, aby poinformować go, z którym zdalnym źródłem danych ma się połączyć. Do wpisania zdalnej bazy danych i odpowiadającego jej węzła do katalogu można użyć Kreatora dodawania bazy danych dostępnego w Asyście podczas konfigurowania.
2. Nawiąż połączenie z lokalną bazą danych na serwerze stowarzyszonym. Jeśli jeszcze nie istnieje lokalna baza danych, utwórz ją, a następnie połącz z tą bazą danych.
3. Zarejestruj opakowanie, wydając komendę `CREATE WRAPPER`. Na przykład jeśli zdalne źródło danych należy do rodziny DB2, na przykład z/OS, wydaj następującą komendę:

```
CREATE WRAPPER drda LIBRARY 'libdb2drda.a'
```

Należy wiedzieć, która biblioteka jest wymagana przez dane źródło.

4. Zarejestruj definicje wszystkich serwerów używanych przez zdalne źródła danych, do których ma być możliwy dostęp. Na przykład utwórz opakowanie `drda` dla serwerów rodziny DB2. Aby zarejestrować zdalny serwer z/OS, wydaj następującą komendę:

```
CREATE SERVER
nazwa_serwera
  TYPE DB2/ZOS
  VERSION 8.1
  WRAPPER DRDA
  AUTHORIZATION "identyfikator_uzytkownika" PASSWORD "haslo"
  OPTIONS (DBNAME 'nazwa_bazy_danych')
```

**Wskazówka:** Komendę tę można uruchomić w opisany sposób z pliku skryptu SQL. Aby wydać komendę bezpośrednio z okna komend DB2, wpisz komendę ujętą w znaki podwójnego cudzysłowu, a podane przez siebie identyfikator użytkownika i hasło opatrz znakami podwójnego cudzysłowu ze znakami zmiany trybu (\"):

```
DB2 "CREATE SERVER
nazwa_serwera
TYPE DB2/ZOS
VERSION 8.1
WRAPPER drda
AUTHORIZATION \"identyfikator_uzytkownika\" PASSWORD \"haslo\"
OPTIONS (DBNAME 'nazwa_bazy_danych')"
```

5. Dla każdego zdefiniowanego serwera ustaw obciążenie zapytaniem tak, aby zawsze było przekazywane w dół z serwera stowarzyszonego na zdalne źródło danych. Dane w zdalnym źródle danych bywają obszerne i nie jest wskazane, aby program DB2 UDB na serwerze stowarzyszonym kopiował dane ze zdalnego źródła danych na serwer stowarzyszony. Gdy serwer będzie ustawiony na maksymalny stopień przekazywania w dół, program DB2 UDB zawsze będzie próbował najpierw przekazać zapytanie w dół do zdalnego źródła danych, tak aby tylko tabela wynikowa zapytania była kopiowana na serwer stowarzyszony. Takie ustawienie jest potrzebne, jeśli zależy nam na uzyskaniu wzrostu wydajności w wyniku optymalizacji wykonanej przez program DB2 Cube Views. Na przykład:

```
CREATE SERVER OPTION DB2_MAXIMAL_PUSHDOWN
FOR SERVER nazwa_serwera
SETTING 'Y'
```

6. Utwórz odwzorowanie użytkownika, aby opakowanie mogło automatycznie połączyć się z serwerem. Na przykład:

```
CREATE USER MAPPING FOR USER
SERVER nazwa_serwera
OPTIONS(REMOTE_AUTHID 'nazwa_uzytkownika', REMOTE_PASSWORD 'haslo')
```

## Definiowanie pseudonimów tabel zdalnych dla programu DB2 Cube Views

Należy zdefiniować pseudonimy dla każdej tabeli zdalnej należącej do schematu gwiazdowego lub schematu typu płatek śniegu tworzonego na serwerze stowarzyszonym. Pseudonimy są potrzebne w programie DB2 Cube Views do współdziałania z tabelami zdalnymi.

Aby zdefiniować pseudonim dla tabeli w zdalnym źródle danych:

Aby zdefiniować pseudonim dla wszystkich tabel zdalnych, do których ma być możliwy dostęp, użyj instrukcji CREATE NICKNAME. Na przykład:

```
CREATE NICKNAME
lokalna_nazwa_schematu.lokalny_pseudonim
FOR
nazwa_serwera_zdalnego.zdalna_nazwa_schematu.zdalna_nazwa_tabeli
```

**Wskazówka:** Dla pseudonimów lokalnych należy użyć tej samej nazwy schematu, która została zdefiniowana na serwerze zdalnym. Na przykład:

```
CREATENICKNAMECVSAMPLE.FAMILY
FOR
nazwa_serwera_zdalnego.CVSAMPLE.FAMILY
```

## Definiowanie ograniczeń informacyjnych dla pseudonimów dla programu DB2 Cube Views

Ograniczenia informacyjne definiuje się w celu udokumentowania związków między danymi, co może poprawić wydajność. Ograniczenia informacyjne należy definiować dla kluczy obcych między tabelami w schemacie gwiazdzistym lub w schemacie typu płatek śniegu na serwerze stowarzyszonym.

Na podstawie ograniczeń informacyjnych Doradca optymalizacyjny programu DB2 Cube Views zaleca zwiększające wydajność tabele podsumowań. Ponadto optymalizator DB2 wykorzystuje te ograniczenia do efektywniejszego przetwarzania zapytań i odpowiedniego kierowania zapytań do istniejących tabel podsumowań.

Program DB2 Information Integrator automatycznie definiuje na serwerze stowarzyszonym ograniczenia przez klucz podstawowy zgodne z istniejącymi ograniczeniami przez klucz podstawowy w zdalnym źródle danych. Należy utworzyć ograniczenia informacyjne dla kluczy obcych używanych do budowania schematu gwiazdzistego lub schematu typu płatek śniegu na serwerze stowarzyszonym.

Aby zdefiniować klucz obcy:

Aby dodać ograniczenia informacyjne do istniejących pseudonimów, użyj instrukcji ALTER NICKNAME. Na przykład:

```
ALTER NICKNAME
  lokalna_nazwa_schematu.lokalny_pseudonim
  ADD FOREIGN KEY(nazwa_kolumny)
  REFERENCES
  lokalna_nazwa_schematu.lokalny_pseudonim
  (nazwa_kolumny)
  ON DELETE RESTRICT
  NOT ENFORCED
  ENABLE QUERY OPTIMIZATION
```

W ten sposób zostało zakończone włączanie zdalnego źródła danych dla programu DB2 Cube Views.

Następnie, posługując się programem DB2 Cube Views, należy zbudować kompletny model kostki dla schematu gwiazdzistego lub schematu typu płatek śniegu na serwerze stowarzyszonym. Po zbudowaniu kompletnego modelu kostki można zoptymalizować dany model kostki przy użyciu Doradcy optymalizacyjnego.

---

## Rozwiązywanie problemów z wydajnością zapytań w przypadku zdalnych źródeł danych

Jeśli wydajność zapytań skierowanych do zdalnego źródła danych nie poprawia się po optymalizacji zgodnie z oczekiwaniami, należy się upewnić, że system stowarzyszony jest skonfigurowany poprawnie pod kątem programu DB2 Cube Views.

Aby można było oczekiwać usprawnienia zapytań zgłaszanych do zdalnych źródeł danych, najpierw należy wykonać następujące czynności:

1. Włącz obsługę systemu stowarzyszonego w programie DB2 Cube Views.
2. Utwórz kompletny model kostki określający reguły podstawowe, reguły kompletności modelu kostki oraz reguły optymalizacji opisane w sekcji “Reguły obiektów metadanych” na stronie 38.
3. Zoptymalizuj model kostki.
4. Jeśli wydajność zapytań nie poprawia się, wykonaj czynności opisane w sekcji “Rozwiązywanie problemów z tabelami podsumowań” na stronie 118.



Jeśli po zrealizowaniu powyższych czynności wydajność zapytań nadal się nie poprawia, wykonaj następujące działania:

- Upewnij się, że zdefiniowane są wszystkie stosowne ograniczenia.
- Upewnij się, że ustawienie `DB2_MAXIMAL_PUSHDOWN` jest ustawione na `yes` (tak), według opisu w sekcji “Definiowanie zdalnych źródeł danych” na stronie 126.
- Rozważ zastosowanie kolokacji tabel wymiarów, które są uwzględniane w zapytaniach, na serwerze stowarzyszonym. Kolokacja wymiarów, w wyniku której zreplikowana kopia tabeli wymiarów znajduje się na serwerze stowarzyszonym, może poprawić wydajność.



---

## Rozdział 7. Interfejs API programu DB2 Cube Views

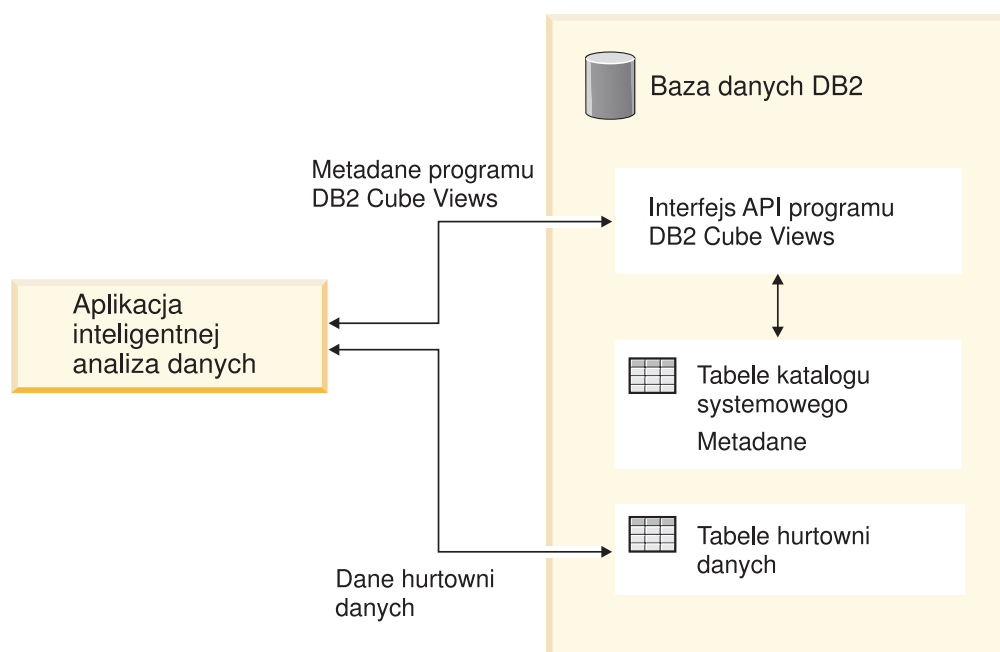
---

### Przegląd interfejsu API programu DB2 Cube Views

Program DB2 Cube Views jest wyposażony w aplikacyjny interfejs programistyczny (API), zapewniający programowy dostęp do metadanych przechowywanych przez program DB2 Cube Views. Za pośrednictwem funkcji API aplikacje mogą korzystać z obiektów metadanych programu DB2 Cube Views i operować na metadanych bez potrzeby odwoływania się do relacyjnych tabel i łączeń.

Interfejs API programu DB2 Cube Views umożliwia dostęp do metadanych przechowywanych w systemowych tabelach katalogu w bazie danych DB2. Aplikacje korzystające z interfejsu API mogą tworzyć i modyfikować obiekty metadanych modelujące struktury wielowymiarowe i struktury OLAP w hurtowni danych.

Rys. 39 pokazuje, jak odbywa się wymiana danych i metadanych za pośrednictwem interfejsu API.



Rysunek 39. Wymiana danych za pomocą interfejsu API programu DB2 Cube Views

Interfejs API jest pojedynczą procedurą zapisaną w bazie, zarejestrowaną w bazie danych DB2. Procedura ta przyjmuje parametry wejściowe i wyjściowe reprezentujące złożone metadane i operacje na metadanych. Format parametrów interfejsu API jest definiowany przez schemat XML.

Interfejs API korzysta z następujących technik wymiany metadanych między programem DB2 Cube Views a aplikacjami do inteligentnej analizy danych:

- SQL z ODBC
- DB2 CLI i JDBC
- XML

---

## Interfejs API programu DB2 Cube Views: procedura zapisana w bazie DB2 i analiza składniowa XML

Interfejs API programu DB2 Cube Views umożliwia wymianę metadanych między aplikacją do inteligentnej analizy danych a bazą danych DB2 przy użyciu procedury zapisanej w bazie programu DB2 Cube Views i analizy składniowej XML.

Informacje na temat programowania procedur zapisanych w bazie DB2 zawiera podręcznik *DB2 Application Development Guide*. Przed rozpoczęciem programowania z wykorzystaniem interfejsu API programu DB2 Cube Views należy zapoznać się z następującymi zagadnieniami:

### Transakcja

Program DB2 UDB działa według modelu transakcyjnego, umożliwiając korzystanie z metadanych wielu użytkownikom jednocześnie (patrz *DB2 Application Development Guide*). Wszystkie działania interfejsu API programu DB2 Cube Views są wykonywane w bazie danych w ramach transakcji aplikacji wywołującej. Dzięki temu aplikacja może wywołać polecenia COMMIT lub ROLLBACK po wywołaniu przez funkcje API procedury md\_message umożliwiającej sterowanie jednostkami pracy w bazie danych.

### Zarządzanie pamięcią

Wymiana parametrów między aplikacjami a procedurą zapisaną w bazie md\_message odbywa się poprzez struktury CLOB. Aplikacje wywołujące procedurę md\_message muszą wstępnie przydzielać dla struktur parametrów CLOB pamięć o takiej wielkości, jaka została podana podczas wpisywania do katalogu procedury zapisanej w bazie. Interfejs API umożliwia stosowanie obiektów CLOB o maksymalnej wielkości dopuszczalnej w programie DB2 UDB, czyli 2 GB. Domyślna wielkość CLOB wynosi 1 MB.

### Konfiguracja systemu

Aby wymiana dużych parametrów była możliwa, konieczna może być zmiana następujących ustawień programu DB2 UDB:

- Aplikacja kliencka bazy danych wywołująca procedurę zapisaną w bazie md\_message może wymagać podania większej sterty i stosu przy konsolidacji.
- Konieczne może być powiększenie sterty zapytań programu DB2 dla bazy danych za pomocą parametru query\_heap\_sz.

### Analiza składniowa kodu XML

Aplikacje korzystające z funkcji API muszą wykonywać analizę składniową parametru wyjściowego zwracanego przez procedurę zapisaną w bazie md\_message. Programiści chcący korzystać z interfejsu API mają do wyboru rozmaite analizatory składni kodu XML.

### Obsługa błędów

Interfejs API generuje informacje o błędach w trojakiem formie:

- kody SQLCODE i SQLSTATE zwracane przez procedurę zapisaną w bazie do aplikacji wywołującej,
- struktury XML przekazywane do aplikacji wywołujących za pomocą parametru API **response**,
- pliki protokołu błędów i czasu wykonania, znajdujące się na serwerze bazy danych, na którym jest uruchomiony interfejs API.

Jeśli błąd wystąpi podczas sprawdzania poprawności kodu XML, jego analizowania lub ze względu na znaczniki, to parametr **response** zostanie zwrócony do aplikacji wywołującej ze znacznikiem <error> zamiast znacznika operacji. Znacznik XML <error> zawierać będzie znacznik <status> z kodem powrotu i komunikatem opisującym problem napotkany przez funkcję API.

Jeśli w funkcji API wystąpi błąd niezwiązany z przetwarzaniem XML, ale związany z wykonaniem operacji na metadanych, zwracana jest zawartość parametru **response**.

Poniższy przykład ilustruje typ informacji zawartych w znaczniku <error>. W tym przykładzie w opisach struktur parametrów wykorzystano ograniczoną liczbę znaczników XML. Większość parametrów zawiera większą liczbę znaczników XML niż tu pokazano, a poprawność zawartości parametrów jest sprawdzana na podstawie schematu XML.

```
<olap:response xmlns:olap="http://www.ibm.com/olap"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="8.1.2.1.0">
<error>
<status id="3100" text="System nie mógł przeanalizować kodu XML
  &quot;PARAMETRU WEJŚCIOWEGO&quot;; (wiersz:&quot;3&quot;;,
  znak:&quot;26&quot;;, komunikat:&quot;Nieznany element
  'dropa'&quot;);. " type="error"/>
</error>
</olap:response>
```

Po wywołaniu procedury zapisanej w bazie programu DB2 Cube Views, bez względu na to, czy procedura została wykonana, czy nie, program DB2 UDB zwraca do aplikacji wywołującej kody SQLCODE i SQLSTATE. Jeśli procedura zapisana w bazie może zostać wykonana, zwraca ona komunikat statusu będący częścią danych XML przekazywanych aplikacji wywołującej.

---

## Procedura zapisana w bazie programu DB2 Cube Views

Procedura nosi nazwę `md_message` i służy do przetwarzania parametrów wyrażonych w formie parametrów programu DB2 Cube Views.

Procedura wyodrębnia z parametrów wejściowych informacje operacyjne oraz informacje o metadanych, a następnie wykonuje żądane operacje na metadanych. Procedura generuje parametry wynikowe zawierające status wykonania (powodzenie lub niepowodzenie) operacji wymienionych w żądaniu oraz informacje o metadanych, zależne od specyfiki operacji.

Procedura zapisana w bazie działa jako użytkownik zdefiniowany w pliku `.fenced` w systemach UNIX. Użytkownicy chronieni muszą mieć uprawnienie do zapisu (WRITE) do plików protokołu podanych w pliku `db2md_config.xml`. W przeciwnym razie procedura zapisana w bazie nie będzie mogła nic protokołować.

Procedura zapisana w bazie programu DB2 Cube Views jest implementowana jako procedura zapisana w bazie programu DB2 UDB. Może z niej korzystać każda aplikacja korzystająca z dowolnego interfejsu programistycznego programu DB2 UDB. W nazwie tej procedury nie jest rozróżniana wielkość liter, natomiast w nazwie i zawartości jej parametrów wielkość liter jest rozróżniana. Składnia procedury `md_message` oraz jej prototyp wyglądają następująco:

```
Składnia:    call md_message (request, metadata, response)
Prototyp:    md_message (request IN CLOB(1M),
                                metadata INOUT CLOB(1M),
                                response OUT CLOB(1M))
```

Parametry **request** (żądanie), **metadata** (metadane) i **response** (odpowiedź) są typu CLOB, który jest typem danych zdefiniowanym w programie DB2 UDB. Aplikacja zapełnia parametr **request** opisem operacji do wykonania i może opcjonalnie zapełnić parametr **metadata**

metadanymi, na których operacja powinna zostać wykonana. Po przetworzeniu parametrów wejściowych procedura md\_message zwraca status operacji w parametrze **response**, a oczekiwane metadane - w parametrze **metadata**. Parametr **metadata** jest przeznaczony na metadane zarówno wejściowe, jak i wyjściowe. Program DB2 UDB przekazuje struktury parametrów między aplikacjami inteligentnej analizy danych a procedurą zapisaną w bazie md\_message na serwerze bazy danych.

Wielkość argumentów CLOB może być różna. Wartością domyślną jest 1 MB. Wielkości argumentów typu CLOB są definiowane w skrypcie sqllib/misc/db2mdapi.sql. Procedurę zapisaną w bazie można ponownie skatalogować z dowolną wielkością parametrów CLOB (maksymalnie 2 gigabajtów). Gdy wielkość parametru zostanie zwiększona, procedura zapisana w bazie będzie zajmować więcej pamięci w czasie wykonywania, ponieważ w momencie uruchomienia procedury pamięć dla buforów parametrów wyjściowych jest wstępnie przydzielana stosownie do wielkości katalogowej. Jeśli wielkość jest zbyt mała, dane przekazywane przez parametry wejściowe i wyjściowe będą obcinane.

Aby zmienić domyślną wielkość obiektów CLOB, należy ponownie zarejestrować funkcję API zapisaną w bazie, podając zwiększone limity wielkości.

1. Zmodyfikuj zawartość pliku sqllib/misc/db2mdapi.sql i zwiększ limity wielkości dla instrukcji CREATE PROCEDURE.

```
CREATE PROCEDURE
  DB2INFO.MD_MESSAGE(IN  request  CLOB(1M),
                    INOUT metadata CLOB(10M),
                    OUT  response CLOB(10M))
```

2. Uruchom plik db2mdapi.sql, aby ponownie zarejestrować funkcję API zapisaną w bazie. Zignoruj błędy sygnalizujące, że skrypt próbuje utworzyć już istniejące tabele.

Więcej informacji na temat wywoływania funkcji API zapisanej w bazie z poziomu kodu C++, można znaleźć w przykładowym kodzie źródłowym w języku C++ w pliku sqllib/samples/olap/client/db2mdapiclient.cpp.

Poniższy przykład ilustruje sposób wywoływania procedury zapisanej w bazie programu DB2 Cube Views z poziomu wbudowanych aplikacji SQL:

```
// Standardowe deklaracje
// ...

// Uwzględnij obszar komunikacyjny, aby uzyskać dostęp do szczegółów błędów
EXEC SQL INCLUDE SQLCA;

// Deklaracje SQL zmiennych języka bazowego, które będą
// używane do wywoływania procedury zapisanej w bazie
// programu DB2 Cube Views
EXEC SQL BEGIN DECLARE SECTION;

// Przydziel obiekt CLOB parametrowi żądania
SQL TYPE is CLOB(1M)      request;

// Przydziel obiekt CLOB parametrowi metadanych
SQL TYPE is CLOB(1M)      metadata;

// Przydziel obiekt CLOB parametrowi odpowiedzi
SQL TYPE is CLOB(1M)      response;

EXEC SQL END DECLARE SECTION;

// Połącz się z bazą danych; pozostałe instrukcje inicjowania aplikacji
// ...

// Zapełnij strukturę parametru żądania opisem operacji
strcpy(request.data, "<request><describe> ... </describe></request>");
```

```

// długość łańcucha ze znakiem końca łańcucha
request.length = strlen(request.data) + 1;

// Zapełnij metadanymi strukturę parametru metadanych
strcpy(metadata.data, "");
// długość łańcucha ze znakiem końca łańcucha
metadata.length = strlen(metadata.data) + 1;

// Wywołaj procedurę zapisaną w bazie programu DB2 Cube Views
EXEC SQL CALL "DB2INFO.MD_MESSAGE"(:request,:metadata,:response);

// Sprawdź, czy procedura zapisana w bazie wróciła bez błędów
if (sqlca.sqlcode)
{
// sprawdzanie błędów przy użyciu sqlaintp()
}

// Przetwórz strukturę parametru odpowiedzi,
// aby określić, czy operacja się powiodła
// ...

// Przetwórz strukturę parametru metadanych,
// aby wyodrębnić żądane metadane
// ...

// Rozłącz się z bazą danych; pozostałe operacje kończące aplikację
// ...

```

---

## Parametry funkcji API programu DB2 Cube Views

### Parametry operacji na metadanych interfejsu API programu DB2 Cube Views

Interfejs API programu DB2 Cube Views oferuje trzy typy funkcji dotyczących metadanych: pobieranie, modyfikowanie i administrowanie. Do każdego typu funkcji należy przynajmniej jedna operacja, a każda operacja ma zestaw parametrów.

Format parametru definiuje standard, zgodnie z którym operacje na metadanych oraz obiekty metadanych są reprezentowane i wymieniane między aplikacjami inteligentnej analizy danych a programem DB2 Cube Views. Format parametru, stosowany do reprezentowania operacji na metadanych oraz obiektów metadanych programu DB2 Cube Views, jest oparty na języku XML. Format parametru jest definiowany przez schemat XML.

### Parametry wejściowe i wyjściowe

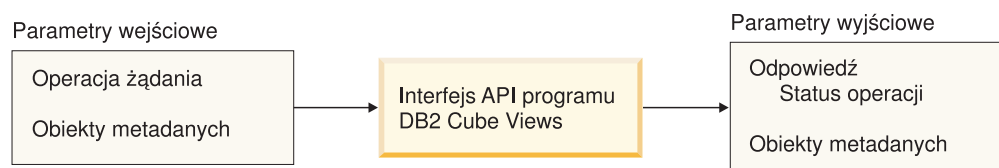
Każda operacja w procedurze zapisanej w bazie md\_message ma dwa parametry wejściowe i dwa parametry wyjściowe.

Parametrami wejściowymi są parametr **request** (żądanie) i parametr **metadata** (metadane). Parametrami wyjściowymi są parametr **response** (odpowiedź) i parametr **metadata** (metadane).

Wejście	Zawartość parametru
Request	Zawiera żadaną operację procedury zapisanej w bazie. W żądaniu mogą być uwzględnione opcje wpływające na działanie i zasięg tego żądania.

W wejście	Zawartość parametru
Metadata	Zawiera obiekty metadanych, które będą używane wraz z operacją opisaną w parametrze <b>request</b> .  W niektórych operacjach żądań, na przykład Describe, metadane wejściowe nie są wymagane.
<b>Wyjście</b>	
Response	Zawiera wszystkie, z wyjątkiem obiektów metadanych, wyniki operacji przeprowadzonej przez procedurę zapisaną w bazie.
Metadata	Zawiera obiekty metadanych, których zażądała operacja opisana w parametrze wejściowym <b>request</b> .  Niektóre operacje, na przykład Create, nie zwracają metadanych wyjściowych.

Rys. 40 pokazuje, jak parametry wejściowe są przekazywane z aplikacji zgłaszającej żądanie przez interfejs API programu DB2 Cube Views do parametrów wyjściowych. Interfejs API tworzy reprezentację obiektów metadanych możliwą do odczytania przez program DB2 UDB.



Rysunek 40. Parametry API

## Operacje na metadanych programu DB2 Cube Views

### Operacja pobierania: Describe

Program DB2 Cube Views udostępnia jedną operację pobierania: Describe.

Operacja Describe obsługuje wersję XML zarówno 8.1.2.1.0, jak i 8.2.0.1.0. Wszystkie pozostałe operacje wymagają XML w wersji 8.2.0.1.0. Więcej informacji na temat wersji plików schematu XML zawiera sekcja “Tabele metadanych i pliki schematu XML programu DB2 Cube Views” na stronie 161.

#### Describe

W ramach tej operacji pobierane są informacje o obiekcie metadanych. W wyniku tej operacji zwracane są informacje odnoszące się do jednego lub kilku obiektów metadanych typu określonego parametrem *objectType* (na przykład pojedynczy obiekt wymiaru, zestaw obiektów wymiarów, zestaw obiektów obejmujący wszystkie typy obiektów). Obiekty metadanych są zwracane w parametrze **metadata**. Format używany do prezentacji pobieranych obiektów metadanych, opisano w sekcji “Format obiektów metadanych” na stronie 163. W trybie rekurencyjnym w wyniku tej operacji zwracane są informacje dla wszystkich pozostałych obiektów metadanych w miarę przetwarzania obiektów według ścieżek powiązań obiektów, zaczynając od zestawu obiektów zgodnych ze specyfikacją *objectType*.

Tabela 39 na stronie 137 zawiera listę komponentów parametru **request** (żądanie) operacji Describe. Tabela 40 na stronie 137 zawiera listę komponentów parametru **response** (odpowiedź) operacji Describe.



Tabela 39. Parametry żądania operacji Describe

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
objectType	argument	Łańcuch XML	all cubeModel cube cubeFacts cubeDimension cubeHierarchy cubeLevel dimension facts hierarchy attributeRelationship level join measure attribute	Typy możliwych do pobrania obiektów metadanych programu DB2 Cube Views. Wartością tego parametru może być jedna z poprawnych wartości.
restriction [opcjonalny]	argument	Łańcuch XML	Patrz: "Parametry operacji" na stronie 152.	Ogranicza zasięg pobierania metadanych. Pełni funkcję analogiczną do predykatu w języku SQL.
recurse	argument	Łańcuch XML	yes no	Włącza lub wyłącza rekurencyjne pobieranie obiektów metadanych.

Tabela 40. Parametry odpowiedzi operacji Describe

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Describe.
object	wyniki pobierania	XML	Patrz: "Operandy operacji" na stronie 155.	Żądane obiekty danych metadanych. Ta wartość jest łańcuchem pustym, jeśli podczas operacji wystąpił błąd.

## Operacja doradzania: Advise

Ta operacja generuje zalecenia dotyczące tabel podsumowań, które należy zbudować w celu poprawy wydajności zapytań w danym modelu kostki. Argumenty tej operacji umożliwiają ograniczenie czasu działania doradcy oraz ilość miejsca na dysku przeznaczoną na tabele podsumowań.

Tabela 41. Parametry operacji Advise

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
cubeModelRef	argument	Element XML		Model kostki, który ma być optymalizowany.
tableName [opcjonalny]	argument	Łańcuch XML		Obszar tabel na tabele podsumowań. Jeśli nie zostanie podany żaden obszar tabel, wygenerowany kod SQL też nie będzie określał obszaru tabel. W takim przypadku program DB2 UDB umieści tabele w domyślnym obszarze tabel.

Tabela 41. Parametry operacji Advise (kontynuacja)

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
indexspaceName [opcjonalny]	argument	Łańcuch XML		Obszar tabel na indeksy tabel podsumowań. Jeśli nie zostanie podany żaden obszar tabel, wygenerowany kod SQL też nie będzie określał obszaru tabel. W takim przypadku program DB2 UDB umieści indeksy w domyślnym obszarze tabel.
diskspaceLimit [opcjonalny]	argument	Nieujemna liczba całkowita XML		Ilość miejsca na dysku (w bajtach) dostępna dla tabel podsumowań i ich indeksów. Wartość 0 oznacza, że doradca samodzielnie decyduje o ilości zajmowanego miejsca na dysku. Większy przydział miejsca na dysku pozwala uzyskać zalecenia wyższej jakości. Wartość 0 oznacza brak ograniczeń. Wartością domyślną jest 0.
timeLimit [opcjonalny]	argument	Nieujemna liczba całkowita XML		Ilość czasu (w sekundach) przeznaczony na generowanie zaleceń. Wydłużenie tego czasu pozwala uzyskać zalecenia wyższej jakości. Wartość 0 oznacza brak ograniczeń; doradca zakończy pracę, gdy stwierdzi, że jej kontynuacja nie przynosi już dalszej poprawy jakości generowanych zaleceń. Wartością domyślną jest 0.
sampling [opcjonalny]	argument	Łańcuch XML	yes lub no	Określa, czy ma być wykonywane próbkowanie danych z tabel podstawowych modelu kostki. Próbkiowanie na ogół poprawia jakość zaleceń, ale wydłuża czas potrzebny na działanie doradcy. Jeśli próbkowanie jest wyłączone, zalecenia doradcy będą oparte jedynie na statystyce bazy danych. Przy włączonym próbkowaniu doradca wykorzystuje zarówno statystykę bazy danych, jak i same dane. Dysponując niewielką repliką oryginalnych danych, można zmienić statystykę bazy DB2 UDB, tak aby tabele sprawiały wrażenie tak dużych, jak oryginały. Jeśli próbkowanie jest wyłączone, doradca wygeneruje takie same zalecenia, jak w przypadku oryginalnych tabel bez próbkowania.

Tabela 41. Parametry operacji Advise (kontynuacja)

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
refresh [opcjonalny]	argument	Łańcuch XML	deferred lub immediate	Określa, czy program DB2 UDB powinien podejmować próby odświeżania tabel podsumowań bezpośrednio po modyfikacji tabel podstawowych, aby zagwarantować ich synchronizację. Parametr deferred (odroczone) oznacza, że należy określić, kiedy tabelle podsumowań mają być odświeżane. Przy ustawieniu immediate (natychmiastowe) program DB2 będzie aktualizował tabelle podsumowań jednocześnie z aktualizacją tabel podstawowych. Jeśli tabelle podstawowe są modyfikowane w wyniku ładowania danych, odświeżanie należy uruchomić ręcznie. Tabelle podsumowań z odświeżaniem natychmiastowym są aktualizowane przyrostowo, natomiast tabelle z odświeżaniem odroczonym są odbudowywane od podstaw. Opcja odświeżania natychmiastowego jest obwarowana wieloma ograniczeniami i doradca może wybrać opcję odświeżania odroczonego, nawet jeśli zostanie podany parametr natychmiastowe.

Poniższy przykład ilustruje operację Advise:

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
<advise
  tablespaceName="TS_MQTTABLE"
  indexspaceName="TS_MQTINDEX"
  diskpaceLimit="75000"
  timeLimit="300"
  sampling="yes"
  refresh="immediate">
<cubeModelRef name="SalesModel" schema="MDSAMPLE"/>
</advise>
</olap:request>
```

Tabela 42. Parametry odpowiedzi doradzania

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views		Komunikat informujący o statusie operacji doradzania

Tabela 42. Parametry odpowiedzi doradzenia (kontynuacja)

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
info	lista komunikatów	Lista struktur komunikatów		Komunikaty ostrzegawcze informujące o napotkanych warunkach niekrytycznych oraz o możliwych zmianach w wynikach operacji doradzenia w stosunku do oczekiwań wynikających z zastosowanych parametrów. Na przykład operacja doradzenia może zignorować parametr REFRESH IMMEDIATE ze względu na obecność miar niedystrybuowalnych w zalecanych tabelach MQT. Komunikaty informacyjne mogą zawiadamiać, dlaczego pewne części modelu nie zostały zoptymalizowane, lub informować o przeprowadzonych zabiegach optymalizacyjnych wraz z wyjaśnieniem, dlaczego zostały wybrane określone agregacje i indeksy.
recommendation		Element XML		Oszacowana łączna ilość miejsca na dysku wymagana na wszystkie zalecane tabele podsumowań i indeksy.
sql		Element XML		Kod SQL, który tworzy i zapełnia tabelę podsumowań oraz tworzy dla niej indeksy. Za uruchomienie kodu SQL odpowiada użytkownik.
refreshSql		Element XML		Kod SQL odświeżający tabele podsumowań w celu ich zsynchronizowania z tabelami podstawowymi, które były modyfikowane. Za uruchomienie kodu SQL odpowiada użytkownik. Tabele podsumowań z odświeżaniem natychmiastowym są czasem aktualizowane przez program DB2 UDB automatycznie, równocześnie z modyfikacją tabel podstawowych. Zależy to od sposobu modyfikacji tabel podstawowych.

Poniższy przykład ilustruje odpowiedź operacji Advise:

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
<advise>
<status id="0" text="Operation completed successfully.
No errors were encountered."type="informational"/>
<info>
<message id="7401" text="The DB2INFO.MQT0000000041T01 summary table
is recommended.
It is estimated to have 100 rows, 55KB table size and 5KB index size."/>
<message id="7401" text="The DB2INFO.MQT0000000041T02 summary table
is recommended.
It is estimated to have 8 rows, 4KB table size and 1KB index size."/>
<message id="7406" text="The PART dimension does not have any
hierarchies that can be optimized by the Optimization Advisor.
The recommendations will not optimize for any attributes
from this dimension."/>
</info>
<recommendation diskspace="65108"/>
```

```

<sql>
<![CDATA[
-- *****
-- * Skrypt tworzący/odświeżający tabele podsumowań.
-- *
-- * Schemat modelu kostki:   MDSAMPLE
-- * Nazwa modelu kostki:     SalesModel
-- * Limit miejsca na dysku:  75000
-- * Limit czasu:             300
-- * Próbkowanie:             Yes
-- * Typ odświeżania:         Refresh immediate
-- * Nazwa obszaru tabel:     TS_MQTTABLE
-- * Nazwa obszaru indeksów:  TS_MQTINDEX
-- *****

DROP TABLE DB2INFO.MQT000000041T01;

DROP TABLE DB2INFO.MQT000000041T02;

UPDATE COMMAND OPTIONS USING c OFF;

CREATE SUMMARY TABLE DB2INFO.MQT000000041T01 AS
(SELECT
SUM(T2."COGS") AS "COGS",
SUM(T2."MARKETING"+T2."PAYROLL") AS "EXPENSE",
SUM(T2."MARKETING") AS "MARKETING",
SUM(T2."PAYROLL") AS "PAYROLL",
SUM(T2."SALES"-(T2."COGS"+(T2."MARKETING"+T2."PAYROLL")))) AS "PROFIT",
SUM(T2."SALES") AS "SALES",

COUNT(*) AS "COUNT",
T5."REGION" AS "REGION",
T5."DIRECTOR" AS "DIRECTOR",
T6."FAMILY" AS "FAMILY",
T6."FAMILYNAME" AS "FAMILYNAME",
T3."SKU" AS "SKU",
T3."CAFFEINATED" AS "CAFFEINATED",
T3."OUNCES" AS "OUNCES",
T3."PKGTYPE" AS "PKGTYPE",
T3."SKUNAME" AS "SKUNAME",
T4."YEAR" AS "YEAR"

FROM
"MDSAMPLE"."MARKET" AS T1,
"MDSAMPLE"."SALESFACT" AS T2,
"MDSAMPLE"."PRODUCT" AS T3,
"MDSAMPLE"."TIME" AS T4,
"MDSAMPLE"."REGION" AS T5,
"MDSAMPLE"."FAMILY" AS T6

WHERE
T1."STATEID"=T2."STATEID" AND
T3."PRODUCTID"=T2."PRODUCTID" AND
T4."TIMEID"=T2."TIMEID" AND
T1."REGIONID"=T5."REGIONID" AND
T3."FAMILYID"=T6."FAMILYID"

GROUP BY
T5."REGION",
T5."DIRECTOR",
T6."FAMILY",
T6."FAMILYNAME",
T3."SKU",
T3."CAFFEINATED",
T3."OUNCES",

```

```

T3."PKGTYPE",
T3."SKUNAME",
T4."YEAR")

DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
IN "TS_MQTTABLE"
INDEX IN "TS_MQTINDEX"
NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000041T01 IS 'AST created for
cube model MDSAMPLE.SalesModel';

REFRESH TABLE DB2INFO.MQT0000000041T01;

CREATE INDEX DB2INFO.IDX0000000041T0101 ON DB2INFO.MQT0000000041T01("FAMILY",
"SKU");
CREATE INDEX DB2INFO.IDX0000000041T0102 ON DB2INFO.MQT0000000041T01("REGION");

RUNSTATS ON TABLE DB2INFO.MQT0000000041T01 AND INDEXES ALL;

CREATE SUMMARY TABLE DB2INFO.MQT0000000041T02 AS
(SELECT
SUM(T2."COGS") AS "COGS",
SUM(T2."MARKETING"+T2."PAYROLL") AS "EXPENSE",
SUM(T2."MARKETING") AS "MARKETING",
SUM(T2."PAYROLL") AS "PAYROLL",
SUM(T2."SALES"-(T2."COGS"+(T2."MARKETING"+T2."PAYROLL"))) AS "PROFIT",
SUM(T2."SALES") AS "SALES",

COUNT(*) AS "COUNT",
T1."YEAR" AS "YEAR",
T1."QUARTER" AS "QUARTER"

FROM
"MDSAMPLE"."TIME" AS T1,
"MDSAMPLE"."SALESFACT" AS T2

WHERE
T1."TIMEID"=T2."TIMEID"

GROUP BY
T1."YEAR",
T1."QUARTER")

DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
IN "TS_MQTTABLE"
INDEX IN "TS_MQTINDEX"
NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000041T02 IS 'AST created for
cube model MDSAMPLE.SalesModel';

REFRESH TABLE DB2INFO.MQT0000000041T02;

CREATE INDEX DB2INFO.IDX0000000041T02C ON DB2INFO.MQT0000000041T02("QUARTER")
CLUSTER;

REORG TABLE DB2INFO.MQT0000000041T02;

RUNSTATS ON TABLE DB2INFO.MQT0000000041T02 AND INDEXES ALL;

COMMIT;

]]>

```

```

</sql>
<refreshSql>
<![CDATA[
-- *****
-- * Skrypt tworzący/odświeżający tabele podsumowań.
-- *
-- * Schemat modelu kostki:   MDSAMPLE
-- * Nazwa modelu kostki:    SalesModel
-- * Limit miejsca na dysku: 75000
-- * Limit czasu:           300
-- * Próbkowanie:           Yes
-- * Typ odświeżania:        Refresh immediate
-- * Nazwa obszaru tabel:    TS_MQTTABLE
-- * Nazwa obszaru indeksów: TS_MQTINDEX
-- *****

UPDATE COMMAND OPTIONS USING c OFF;

REFRESH TABLE DB2INFO.MQT0000000041T01;
REFRESH TABLE DB2INFO.MQT0000000041T02;
COMMIT;

]]>
</refreshSql>
</advise>
</olap:response>

```

## Operacje modyfikujące: Alter, Create, Drop, Import i Rename

Interfejs API programu DB2 Cube Views zawiera pięć operacji modyfikujących:

- Alter
- Create
- Drop
- Import
- Rename

Jeśli operacja modyfikująca zostanie zaimplementowana w procedurze zapisanej w bazie, gwarantuje ona, że obiekty będą kompletne i poprawne referencyjnie.

### Create

Operacja Create służy do tworzenia obiektów metadanych. Akceptuje operand lub operandy obiektów metadanych i tworzy te obiekty w programie DB2 Cube Views w kolejności przekazywania ich do operacji. Sekwencje obiektów przekazywanych do operacji mogą zawierać obiekty różnego typu.

Obiekty przekazywane do tej operacji mogą odwoływać się do innych obiektów. Jeśli między obiektami istnieją odwołania, to muszą być one odzwierciedlone w kolejności obiektów. Jeśli na przykład obiekt **Obiekt1** odwołuje się do obiektu **Obiekt2**, to obiekt **Obiekt2** musi zostać przekazany do operacji przed obiektem **Obiekt1**. Więcej informacji na temat możliwych wzajemnych odwołań obiektów metadanych można znaleźć w sekcji “Format obiektów metadanych” na stronie 163.

W ramach operacji Create sprawdzana jest poprawność każdego obiektu. Błędy są zwracane wtedy, gdy tworzony obiekt już istnieje lub gdy jeszcze nie istnieje obiekt, do którego tworzony obiekt się odwołuje. Jeśli obiekt wejściowy określa nieistniejący schemat, zostaje on w ramach operacji utworzony, o ile użytkownik ma wystarczające uprawnienie w bazie danych.

Tabela 43. Parametry żądania operacji Create

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
object	operand	Element XML	Patrz: "Operandy operacji" na stronie 155.	Obiekty, które można utworzyć.

Tabela 44. Parametry odpowiedzi operacji Create

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Create.

## Alter

Operacja Alter służy do aktualizowania informacji o obiektach metadanych. Przyjmuje operand lub operandy obiektów metadanych i aktualizuje ich obiektowe odpowiedniki w tabelach katalogu metadanych. Obiekty są aktualizowane w kolejności, w jakiej zostały przekazane do operacji. Sekwencje obiektów przekazywanych do tej operacji mogą zawierać obiekty różnego typu.

Obiekty przekazywane do tej operacji mogą odwoływać się do innych obiektów. Jeśli między obiektami istnieją odwołania, to muszą być one odzwierciedlone w kolejności obiektów. Więcej informacji na temat możliwych wzajemnych odwołań obiektów metadanych można znaleźć w sekcji "Format obiektów metadanych" na stronie 163.

Ta operacja nie może aktualizować schematu ani nazwy obiektu. Nazwy obiektów można zmienić za pomocą operacji Rename.

W ramach operacji Alter sprawdzana jest poprawność każdego obiektu. Błędy są zwracane, gdy aktualizowany obiekt nie istnieje.

Tabela 45. Parametry żądania operacji Alter

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
object	operand	Element XML	Patrz: "Operandy operacji" na stronie 155.	Obiekty, które można zaktualizować.

Tabela 46. Parametry odpowiedzi operacji Alter

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Alter.

## Rename

Operacja Rename umożliwia zmianę nazwy jednego obiektu metadanych programu DB2 Cube Views zdefiniowanego przez bieżący schemat i nazwę. Można zmienić tylko nazwę obiektu. Nie można zmienić jego schematu. Operacja Rename może zmieniać nazwy obiektów nawet wtedy, gdy w danej chwili odwołują się do nich inne obiekty metadanych.



Tabela 47. Parametry żądania operacji Rename

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
objectType	argument	Łańcuch XML	cubeModel cube cubeFacts cubeDimension cubeHierachy cubeLevel dimension facts hierarchy attributeRelationship level join measure attribute	Typ obiektów metadanych programu DB2 Cube Views, których nazwy są zmieniane. Wartością tego parametru może być jedna z poprawnych wartości.
currentRef	operand	Odwołanie do obiektu metadanych programu DB2 Cube Views	Patrz: "Operandy operacji" na stronie 155.	Bieżący schemat i nazwa obiektu metadanych, którego nazwa jest zmieniana.
newRef	operand	Odwołanie do obiektu metadanych programu DB2 Cube Views	Patrz: "Operandy operacji" na stronie 155.	Nowy schemat i nowa nazwa obiektu metadanych, którego nazwa jest zmieniana.

Tabela 48. Parametry odpowiedzi operacji Rename

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Rename.

## Drop

Operacja Drop służy do usuwania obiektów metadanych z programu DB2 Cube Views. Usuwa ona jeden lub wiele obiektów metadanych w zależności od określonych komponentów *objectType* i *restriction*. Jeśli do usuwanego obiektu odwołuje się inny obiekt metadanych, zwracany jest błąd.

Tabela 49. Parametry żądania operacji Drop

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
objectType	argument	Łańcuch XML	Patrz: "Parametry operacji" na stronie 152.	Typy usuwanych obiektów metadanych. Wartością tego parametru może być jedna z poprawnych wartości.
restriction [opcjonalny]	argument	Łańcuch XML	Patrz: "Parametry operacji" na stronie 152.	Ogranicza zasięg usuwania metadanych. Pełni funkcję analogiczną do predykatu w języku SQL.

Tabela 50. Parametry odpowiedzi operacji Drop

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Drop.

## Import

Operacja Import tworzy obiekty metadanych lub zgłasza ich istnienie w katalogu metadanych. Operacja ta zachowuje się podobnie do operacji Create z wyjątkiem sytuacji, w której istnieją już obiekty metadanych.

Dla operacji Import można definiować opcjonalne tryby. Tryby te wyznaczają działanie podejmowane w sytuacji, gdy importowane są obiekty o takich samych nazwach, jak nazwy obiektów znajdujących się już w katalogu.

W zależności od trybu błędy zwracane są, gdy tworzony obiekt już istnieje lub gdy jeszcze nie istnieje obiekt, do którego tworzony obiekt się odwołuje. Jeśli obiekt wejściowy określa nieistniejący schemat, jest on tworzony w ramach operacji, o ile użytkownik ma wystarczające uprawnienie w bazie danych.

W ramach operacji Import sprawdzana jest poprawność każdego obiektu.

Szczegółowy opis różnych trybów operacji można znaleźć w sekcji "Parametry operacji" na stronie 152.

Tabela 51. Parametry żądania operacji Import

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
mode	argument	Łańcuch XML	create new - ignore collisions create new - replace collisions create new - abort on collision report new - report collisions	Definiuje działania dla nowych i istniejących importowanych obiektów.  Patrz opis parametru <i>mode</i> w sekcji "Parametry operacji" na stronie 152.
object	operand	Element XML	Patrz: "Operandy operacji" na stronie 155.	Importowane obiekty.

Tabela 52. Parametry odpowiedzi operacji Import

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat wskazujący status operacji Import.
newList	lista odwołań	Element XML	Patrz opis parametru <i>mode</i> w sekcji "Parametry operacji" na stronie 152.	Lista par nazwa-schemat odwołujących się do nowych obiektów.

Tabela 52. Parametry odpowiedzi operacji Import (kontynuacja)

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
collisionList	lista odwołań	Element XML	Patrz opis parametru <i>mode</i> w sekcji “Parametry operacji” na stronie 152.	Lista par nazwa-schemat odwołujących się do obiektów kolidujących z innymi obiektami.

### Jak funkcje API traktują zależności funkcjonalne podczas operacji modyfikujących

W interfejsie API programu DB2 Cube Views zależności funkcjonalne dotyczące obiektu poziomu są traktowane zgodnie z określonymi regułami. Tabela 53 przedstawia działania podejmowane przez funkcję interfejsu API dla każdego żądania modyfikacji poziomu.

Tabela 53. Działania interfejsu API dla zależności funkcjonalnych

Żądanie	Działanie
Tworzenie poziomu	Funkcja API tworzy odpowiednią zależność funkcjonalną programu DB2, o ile jest to możliwe. Jeśli zależności funkcjonalnej nie można utworzyć, funkcja API tworzy poziom bez zależności funkcjonalnej i zwraca komunikat z ostrzeżeniem.
Usunięcie poziomu	Funkcja API usuwa poziom i skojarzoną z nim zależność funkcjonalną programu DB2. Jeśli w funkcji API wystąpi błąd podczas próby usunięcia poziomu lub zależności funkcjonalnej, funkcja zwróci komunikat o błędzie i obiekt poziomu nie zostanie usunięty.
Zmiana poziomu (bez zależności funkcjonalnej)	Jeśli podczas zmiany poziomu nie jest tworzona zależność funkcjonalna, nie jest podejmowane żadne działanie.  Jeśli zależność funkcjonalna jest tworzona podczas zmieniania poziomu, funkcja API tworzy odpowiednią zależność funkcjonalną programu DB2, o ile jest to możliwe. Jeśli zależności funkcjonalnej nie można utworzyć, funkcja API zmienia poziom bez zależności funkcjonalnej i zwraca komunikat z ostrzeżeniem.
Zmiana poziomu (z zależnością funkcjonalną)	Jeśli poziom ma zależność funkcjonalną, która ma zostać usunięta podczas zmieniania poziomu, funkcja API zmieni poziom i usunie skojarzoną z nim zależność funkcjonalną programu DB2. Jeśli w funkcji API wystąpi błąd podczas próby zmiany poziomu, funkcja zwróci komunikat o błędzie i obiekt poziomu nie zostanie zmieniony.  Aby zmienić poziom, zachowując zależność funkcjonalną, funkcja API usuwa zależność funkcjonalną i tworzy ją ponownie. Jeśli nie można usunąć zależności funkcjonalnej, funkcja API zwraca komunikat o błędzie, ale kontynuuje zmienianie poziomu i ponownie tworzy nową zależność funkcjonalną pod inną nazwą.
Importowanie poziomu (tryb Create new: ignore collisions)	Funkcja API tworzy odpowiednią zależność funkcjonalną programu DB2, o ile jest to możliwe. Jeśli zależności funkcjonalnej nie można utworzyć, funkcja API tworzy poziom bez zależności funkcjonalnej i zwraca komunikat z ostrzeżeniem.
Importowanie poziomu (tryb Create new: replace collisions)	Tak samo jak w przypadku zmiany (operacji Alter).

## Operacje administracyjne: Validate i Translate

W programie DB2 Cube Views dostępne są dwie operacje administracyjne: Validate i Translate. Operacja Translate jest używana tylko podczas migracji programu DB2 Cube

Views z wersji 8.1 do wersji 8.2. W ramach operacji Translate metadane XML w wersji 8.1 są odwzorowywane na metadane XML w wersji 8.2.

## Validate

Podczas operacji Validate następuje weryfikacja struktury jednego lub kilku obiektów metadanych. Kryterium poprawności jest zgodność z regułami dotyczącymi obiektów w programie DB2 Cube Views. Dla obiektów, których poprawność jest sprawdzana w tej operacji, należy podać parametry *objectType* oraz *restriction*. Parametr *mode* definiuje zakres sprawdzania poprawności.

Podczas operacji Validate sprawdzane są następujące cechy obiektów:

- kompletność informacji o obiektach metadanych,
- spójność referencyjna między obiektami metadanych,
- istnienie relacyjnych tabel, widoków, aliasów i kolumn pseudonimów, do których obiekt się odwołuje,
- poprawność wyrażenia SQL zapisanego w obiektach metadanych, takich jak atrybuty i miary.

Operacja Validate zostaje zatrzymana po znalezieniu niepoprawnego obiektu metadanych. Po stwierdzeniu naruszenia reguł operacja sprawdzania poprawności przekazuje komunikat z opisem. Inne operacje (Create, Alter i Import) także sprawdzają poprawność obiektów metadanych, chociaż w sposób niejawni. W operacji tłumaczenia nie jest sprawdzana poprawność obiektów metadanych.

Tabela 54. Parametry operacji Validate

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
objectType	argument	Łańcuch XML	Patrz: "Parametry operacji" na stronie 152.	Typy sprawdzanych pod względem poprawności obiektów metadanych programu DB2 Cube Views. Wartością tego parametru może być jedna z poprawnych wartości.
restriction (opcjonalny)	argument [opcjonalny]	Łańcuch XML	Patrz: "Parametry operacji" na stronie 152.	Ogranicza zasięg sprawdzania poprawności metadanych. Pełni funkcję analogiczną do predykatu w języku SQL.
mode	argument	Łańcuch XML	base cubeModel completeness optimization	Definiuje zakres przewidzianych do wykonania działań sprawdzających.

Tabela 55. Sprawdzanie poprawności parametrów odpowiedzi

Nazwa	Typ	Typ danych	Poprawne wartości	Opis
status	komunikat o statusie	Struktura komunikatu programu DB2 Cube Views	Patrz: "Struktura komunikatów" na stronie 155.	Komunikat informujący o statusie operacji Validate.
info	lista komunikatów	Lista struktur komunikatów	Patrz: "Struktura komunikatów" na stronie 155.	Lista komunikatów opisujących ostrzeżenia i błędy generowane w trakcie operacji Validate.

## Translate

Operacja Translate sprawdza poprawność składniową przychodzących danych XML, opierając się na schemacie XML. Nie są jednak sprawdzane odwołania do kolumn ani do

innych obiektów. Za pomocą operacji Translate można przekształcać metadane programu DB2 Cube Views o dowolnej strukturze. Operacja Translate wymaga podania kompletnego dokumentu XML. Odwołania do obiektów, których nie ma w dokumencie XML, mogą powodować błędy.

W ramach operacji Translate obiekty są odwzorowywane w sposób przedstawiony w poniższej tabeli. Każdy obiekt w wersji 8.1 jest odwzorowywany na identyczny obiekt w wersji 8.2, chyba że zostanie zaznaczone inaczej.

Tabela 56. Sposób odwzorowania obiektów w wersji 8.1 na obiekty w wersji 8.2

Obiekty w wersji 8.1	Obiekty w wersji 8.2
Atrybut	<ul style="list-style-type: none"> <li>Atrybuty są rozszerzane o nową właściwość, <i>nullability</i> (dopuszczalne wartości puste).</li> <li>Dla istniejących atrybutów właściwość <i>nullability</i> będzie miała wartość <i>Brak informacji</i>.</li> </ul>
Łączenie	Bez zmian.
Związek atrybutów	<ul style="list-style-type: none"> <li>Nie odwołują się już do niego hierarchie ani hierarchie kostki.</li> <li>Wszystkie związki atrybutów stały się sierotami. (Żadne inne obiekty nie odwołują się już do związków atrybutów).</li> </ul> <p>Po wyeksportowaniu modelu kostki lub kostki z Centrum OLAP wyeksportowany plik XML nie zawiera już związków atrybutów. Aby wyeksportować związki atrybutów, należy wyeksportować z Centrum OLAP wszystkie metadane.</p>
Hierarchie + atrybuty + związki atrybutów	<ul style="list-style-type: none"> <li>Dla każdego atrybutu, do którego odwołuje się dowolna hierarchia w wersji 8.1, tworzony jest dokładnie jeden obiekt poziomy.</li> <li>Obiekt poziomy będzie korzystać z tej samej nazwy (schemat, nazwa itd.), co atrybut źródłowy w wersji 8.1.</li> <li>Domyślnym atrybutem poziomu będzie jego atrybut źródłowy.</li> <li>Klucz poziomu składa się z atrybutu źródłowego oraz atrybutów przodków, leżących powyżej atrybutu źródłowego w jednej z hierarchii w wersji 8.1.</li> <li>Jeśli do atrybutu źródłowego odwoływała się więcej niż jedna hierarchia w wersji 8.1, wybierana jest hierarchia o najmniejszej liczbie atrybutów przodków (jeśli jest ich kilka, wybór jest przypadkowy), a atrybuty przodków są dodawane - w kolejności zstępującej aż do atrybutu źródłowego - jako elementy klucza dla poziomu w wersji 8.2.</li> <li>Wszystkie związki atrybutów z wersji 8.1, do których się odwołują hierarchie i w których atrybut lewostronny jest równy atrybutowi źródłowemu, zostaną wykorzystane do utworzenia atrybutów pokrewnych poziomu. Dla każdego pasującego (lewostronnie) związku atrybutów do poziomu dodawany jest atrybut prawostronny - jako atrybut pokrewny. Po utworzeniu listy atrybutów pokrewnych usuwane są z niej powtórzenia.</li> </ul>
Hierarchie kostki + atrybuty + związki atrybutów	<ul style="list-style-type: none"> <li>Dla każdego atrybutu, do którego odwołuje się hierarchia kostki w wersji 8.1, tworzony jest jeden unikalny obiekt poziomu kostki.</li> <li>Poziom kostki otrzymuje nazwę utworzoną na podstawie nazwy atrybutu i nazwy kostki. Jeśli kostka nosi nazwę Sprzedaż i ma atrybut Region, poziom kostki będzie nazwany Region (Sprzedaż).</li> <li>Każdy poziom kostki odwołuje się do odpowiadającego mu obiektu poziomu. Na przykład obiekt Region (Sprzedaż) wskazuje poziom o nazwie Region.</li> <li>Atrybuty pokrewne poziomu kostki są oparte na obiektach związku atrybutów w danej hierarchii kostki. Dla każdego związku atrybutów, w którym atrybut lewostronny jest równy atrybutowi źródłowemu (w poziomie nadrzędnym), atrybut prawostronny zostanie dodany do poziomu kostki jako atrybut pokrewny.</li> </ul>
Hierarchia	Każda hierarchia w wersji 8.1 jest odwzorowywana na hierarchię w wersji 8.2 z uporządkowanym zestawem poziomów (bez atrybutów i związków atrybutów).
Hierarchia kostki	Każda hierarchia kostki w wersji 8.1 jest odwzorowywana na hierarchię kostki w wersji 8.2 z uporządkowanym zestawem poziomów kostki (bez atrybutów i związków atrybutów).
Wymiar	Dodawane są powiązane poziomy.

Tabela 56. Sposób odwzorowania obiektów w wersji 8.1 na obiekty w wersji 8.2 (kontynuacja)

Obiekty w wersji 8.1	Obiekty w wersji 8.2
Wymiar kostki	Bez zmian.
Miara	Podobnie jak atrybuty, miary zostały rozszerzane o nową właściwość, <i>nullability</i> (dopuszczalne wartości puste). Dla istniejących miar właściwość <i>nullability</i> będzie miała wartość Brak informacji.
Fakty	Bez zmian.
Fakty kostki	Bez zmian.
Model kostki	Bez zmian.
Kostka	Bez zmian podczas tłumaczenia, lecz w wersji 8.2 kostka może zawierać informacje o wykorzystaniu.

## Przykładowe parametry wejściowe i wyjściowe w operacjach na metadanych

Poniższe przykłady ilustrują strukturę parametrów dla trzech typów operacji na metadanych. Z przykładów tych usunięto fragmenty struktur XML, zastępując je wielokropkami (...).

### Operacja pobierania

Poniższe przykłady przedstawiają strukturę operacji pobierania o nazwie describe. Więcej informacji na temat operacji describe zawiera sekcja “Operacja pobierania: Describe” na stronie 136. W poniższym przykładzie parametr **metadata** jest pusty na wejściu, lecz wypełniony na wyjściu.

#### Żądanie i metadane

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <describe objectType="cube" recurse="no">
    <restriction>
      <predicate property="schema" operator="=" value="myschema"/>
    </restriction>
  </describe>
</olap:request>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... />
```

#### Odpowiedź i metadane

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <describe>
    <status id="0" text="Operację zrealizowano
      pomyślnie." type="informational"/>
  </describe>
</olap:response>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
  <cube name="cube1" schema="myschema" ... > ... </cube>
  ...
  <cube name="cubeN" schema="myschema" ... > ... </cube>
</olap:metadata>
```

### Operacje modyfikacji

Poniższe przykłady przedstawiają strukturę operacji modyfikacji o nazwie create. Więcej informacji na temat operacji create i innych operacji modyfikacji zawiera sekcja “Operacje modyfikujące: Alter, Create, Drop, Import i Rename” na stronie 143. Parametr **metadata** jest wypełniany na wejściu, lecz jest pusty na wyjściu.

### Żądanie i metadane

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <create/>
</olap:request>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
  <attribute name="idPołożenia" ... > ... </attribute>
  <attribute name="Kraj" ... > ... </attribute>
  <attribute name="Stan" ... > ... </attribute>
  <attribute name="Miasto" ... > ... </attribute>
  <dimension name="Położenie" ... type="regular">
    <attributeRef name="idPołożenia" ... </attributeRef>
    <attributeRef name="Kraj" ... </attributeRef>
    <attributeRef name="Stan" ... </attributeRef>
    <attributeRef name="Miasto" ... </attributeRef>
    ...
  </dimension>
</olap:metadata>
```

### Odpowiedź i metadane

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <create>
    <status id="0" text="Operację zrealizowano
      pomyślnie." type="informational"/>
  </create>
</olap:response>
```

```
b<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
```

### Operacja administrowania

Poniższe przykłady przedstawiają strukturę operacji administrowania o nazwie validate. Więcej informacji na temat operacji sprawdzania poprawności Validate zawiera sekcja “Operacje administracyjne: Validate i Translate” na stronie 147.

### Żądanie i metadane

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <validate objectType="cube" mode="base">
    <restriction>
      <predicate property="schema" operator="=" value ="myschema"/>
    </restriction>
  </describe>
</olap:request>
```

```
<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... />
```

### Odpowiedź i metadane

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <validate>
    <status id="1" text="...Zwrócono dodatkowe
      informacje." type="informational"/>
    <info><message id="6299" text="Znaleziono co najmniej
      jeden widok bazy danych podczas sprawdzania poprawności."
      type="warning"/></info>
  </validate>
</olap:response>
```

```
<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
```

Dodatkowe przykładowe pliki XML operacji, których można używać z programem narzędziowym db2mdapiclient, znajdują się w katalogu SQLLIB\samples\olap\xml\input.

## Parametry operacji

Do każdej operacji na metadanych dostępne są różne parametry. Parametry te dostosowują działanie operacji do konkretnego zastosowania.

Program DB2 Cube Views udostępnia pięć parametrów do operacji na metadanych:

- objectType
- recurse
- restriction
- mode (dla operacji importowania - Import)
- mode (dla operacji sprawdzania poprawności - Validate)

### Parametr objectType

Ten parametr określa typ obiektów metadanych biorących udział w żądanej operacji. Niżej wymienione typy obiektów mają bezpośrednie przełożenie na model obiektów metadanych programu DB2 Cube Views.

- all
- cubeModel
- cube
- cubeDimension
- cubeFacts
- cubeHierarchy
- cubeLevel
- dimension
- facts
- hierarchy
- attributeRelationship
- level
- join
- measure
- attribute

### Parametr recurse

Ten parametr decyduje o tym, czy operacja ma być przeprowadzana rekurencyjnie. W trybie nierekurencyjnym w ramach danej operacji działania są przeprowadzane tylko na obiektach metadanych bezpośrednio zgodnych z parametrem objectType oraz specyfikacjami parametru **restriction**. Operacja zaczyna od zestawu obiektów metadanych dla trybu nierekurencyjnego i dodatkowo przeprowadza działania na wszystkich pozostałych obiektach metadanych w miarę, jak postępuje według ścieżek powiązań obiektów, zaczynając od zestawu obiektów dla trybu nierekurencyjnego. Parametr **recurse** może mieć wartości *yes* i *no*.

Na przykład operacja nierekurencyjna może zwracać samą listę wymiarów, podczas gdy operacja rekurencyjna może zwracać listę wymiarów, a oprócz niej wszystkie pozostałe obiekty (różnych typów), do których odwołują się te wymiary, oraz obiekty, do których odwołują się te obiekty wymiarów.



## Parametr restriction

Ten parametr określa, że operacja na metadanych będzie podlegać ograniczeniom lub będzie mieć ograniczony zasięg. Ten parametr ma działanie podobne do predykatów w zapytaniu SQL. Ograniczenia wyraża się w języku XML za pomocą znaczników <restriction> i <predicate> zdefiniowanych w schemacie XML programu DB2 Cube Views.

Ograniczenia mogą być oparte na właściwościach obiektów wspólnych dla wszystkich obiektów metadanych oraz na związkach między obiektami metadanych.

Predykaty oparte na właściwościach zawierają następujące atrybuty:

### *property* (właściwość)

Skojarzony jest ze znacznikiem predykatu, musi być określony jako *name* (nazwa), albo *schema* (schemat).

### *operator*

Skojarzony ze znacznikiem predykatu, musi być określony jako znak równości (=).

### *value* (wartość)

Skojarzony ze znacznikiem predykatu, jest reprezentacją łańcuchową wartości, która ma być porównywana z właściwością określoną atrybutem *property* (właściwość).

Aby zapoznać się z opisem parametru **restriction** i jego umiejscowieniem w sekwencji kroków operacji, patrz: "Kolejność kroków operacji" na stronie 156.

W tym przykładzie zasięg operacji zostaje ograniczony do obiektów w schemacie ABC:

```
<restriction>
  <predicate property="schema" operator="=" value="ABC">
</restriction>
```

## Parametr mode (dla operacji import)

Ten parametr powoduje ustawienie trybu dla operacji importowania (Import). Dostępne tryby opisano w poniższej tabeli.

Kolizja zachodzi wtedy, gdy obiekt przekazywany do operacji importowania jako wejściowy już istnieje w katalogu metadanych.

Tabela 57. Tryby importu

Tryb	Opis	Zwracane listy odwołań
Create new: ignore collisions	<ul style="list-style-type: none"><li>• Tworzone są niekolidujące obiekty wejściowe.</li><li>• Obiekty wejściowe, które kolidują, nie są tworzone.</li><li>• Już istniejące obiekty nie są zmieniane.</li><li>• Kolizje nie powodują wygenerowania błędów.</li></ul>	<b>newList</b>  Zawiera pary nazwa-schemat pomyślnie utworzonych obiektów.  <b>collisionList</b>  Zawiera pary nazwa-schemat obiektów kolidujących, które zostały zignorowane i nie zostały utworzone.

Tabela 57. Tryby importu (kontynuacja)

Tryb	Opis	Zwracane listy odwołań
Create new: replace collisions	<ul style="list-style-type: none"> <li>• Tworzone są niekolidujące obiekty wejściowe.</li> <li>• Obiekty wejściowe, które kolidują, zastępują już istniejące obiekty.</li> <li>• Już istniejące obiekty są zastępowane przez obiekty wejściowe.</li> <li>• Kolizje nie powodują wygenerowania błędów.</li> </ul>	<p><b>newList</b></p> <p>Zawiera pary nazwa-schemat pomyślnie utworzonych obiektów.</p> <p><b>collisionList</b></p> <p>Zawiera pary nazwa-schemat obiektów kolidujących, które zostały zastąpione.</p>
Create new: abort on collision	<ul style="list-style-type: none"> <li>• Obiekty wejściowe są tworzone tylko wtedy, gdy w całej operacji nie występują kolizje.</li> <li>• W wypadku kolizji żaden obiekt nie zostaje w ramach operacji utworzony.</li> <li>• Już istniejące obiekty nie są zmieniane.</li> <li>• Kolizje generują błędy.</li> </ul>	<p><b>newList</b></p> <p>Zawiera pary nazwa-schemat obiektów pomyślnie utworzonych lub obiektów niekolidujących, które nie zostały utworzone.</p> <p><b>collisionList</b></p> <p>Zawiera pary nazwa-schemat obiektów kolidujących, które nie zostały utworzone.</p>
Report new: report collisions	<ul style="list-style-type: none"> <li>• Obiekty w ogóle nie są tworzone.</li> <li>• Zgłaszany jest status kolizji obiektów wejściowych.</li> <li>• Już istniejące obiekty nie są zmieniane.</li> <li>• Kolizje nie powodują wygenerowania błędów.</li> </ul>	<p><b>newList</b></p> <p>Zawiera pary nazwa-schemat obiektów niekolidujących, które nie zostały utworzone.</p> <p><b>collisionList</b></p> <p>Zawiera pary nazwa-schemat obiektów kolidujących, które nie zostały utworzone.</p>

Znaczniki list odwołań newList i collisionList muszą występować w predefiniowanym porządku. Na poniższej liście przedstawiono porządek typów referencyjnych dla operacji IMPORT:

1. <attributeRef>
2. <joinRef>
3. <attributeRelationshipRef>
4. <levelRef>
5. <cubeLevelRef>
6. <hierarchyRef>
7. <cubeHierarchyRef>
8. <dimensionRef>
9. <cubeDimensionRef>
10. <measureRef>
11. <factsRef>
12. <cubeFactsRef>
13. <cubeModelRef>
14. <cubeRef>

## Parametr mode (dla sprawdzania poprawności - operacja Validate)

Ten parametr powoduje ustawienie trybu dla operacji sprawdzania poprawności (Validate). Dostępne tryby operacji sprawdzania poprawności (Validate) opisano w poniższej tabeli. Reguły w tej tabeli odwołują się do kategorii reguł obiektów programu DB2 Cube Views.

Tabela 58. Tryby sprawdzania poprawności

Tryb	Opis
base	Służy do sprawdzania zgodności z regułami podstawowymi
cubeModel completeness	<ul style="list-style-type: none"><li>Służy do sprawdzania zgodności z regułami kompletności modelu kostki</li><li>Służy do sprawdzania zgodności z regułami podstawowymi</li></ul>
optimization	<ul style="list-style-type: none"><li>Służy do sprawdzania zgodności z regułami optymalizacji</li><li>Służy do sprawdzania zgodności z regułami kompletności modelu kostki</li><li>Służy do sprawdzania zgodności z regułami podstawowymi</li></ul>

## Operandy operacji

Gdy operacja wymaga, aby żądaniu towarzyszyły obiekty metadanych lub odwołania do nich, obiekty lub odwołania te są określane mianem "operandów" operacji.

Operandy przekazywane do operacji na metadanych za pomocą parametrów **request** (żądanie) lub **metadata** (metadane) są następujące:

**object** Ten operand zawiera obiekty metadanych, na których przeprowadzane są działania. Format obiektów metadanych opisano w sekcji "Format obiektów metadanych" na stronie 163.

### currentRef

Ten operand jest przeznaczony dla operacji zmiany nazwy (Rename) i zawiera schemat i nazwę obiektu metadanych.

### newRef

Ten operand, podobnie jak operand *currentRef*, jest przeznaczony dla operacji zmiany nazwy i zawiera schemat i nazwę obiektu metadanych.

## Struktura komunikatów

Interfejs API programu DB2 Cube Views zwraca komunikaty informacyjne, komunikaty ostrzegawcze oraz komunikaty o błędach mające określoną strukturę.

Poniższa tabela zawiera opis komponentów komunikatu programu DB2 Cube Views:

Tabela 59. Komponent komunikatu

Komponent	Opis
id	Unikalny identyfikator komunikatu.
type	Komunikat może należeć do jednego z trzech typów: <ul style="list-style-type: none"><li>informacyjny</li><li>ostrzegawczy</li><li>błąd</li></ul>
text	Łańcuch znaków zawierający tekst komunikatu.

Tabela 59. Komponent komunikatu (kontynuacja)

Komponent	Opis
tokens	<p>Wartości podstawiane w łańcuchu tekstowym komunikatu. Komunikat może zawierać dowolną liczbę elementów zmiennych. W komunikacie jako elementy zmienne mogą wystąpić następujące elementy XML:</p> <ul style="list-style-type: none"> <li>• attributeRef</li> <li>• joinRef</li> <li>• attributeRelationshipRef</li> <li>• levelRef</li> <li>• cubeLevelRef</li> <li>• hierarchyRef</li> <li>• cubeHierarchyRef</li> <li>• dimensionRef</li> <li>• cubeDimensionRef</li> <li>• measureRef</li> <li>• factsRef</li> <li>• cubeFactsRef</li> <li>• cubeModelRef</li> <li>• cubeRef</li> <li>• column</li> <li>• text</li> </ul>

Przykład komunikatu bez elementów zmiennych:

```
<status id="0" text="Operacja ukończona pomyślnie." type="informational"/>
```

Przykład komunikatu z elementami zmiennymi:

```
<status id="6331" text="Atrybut lewostronny związku atrybutów
&quot;MDOBJ_ID_ATTRIBUTERELATIONSHIP.MDSAMPLE.State_PopGroup&quot;
nie należy do hierarchii
&quot;MDOBJ_HIERARCHY.MDSAMPLE.RegionState&quot;."
type="error">
<tokens>
<attributeRelationshipRef name="State_PopGroup" schema="MDSAMPLE"/>
<text value="MDOBJ_HIERARCHY.MDSAMPLE.RegionState"/>
</tokens>
</status><
```

## Kolejność kroków operacji

Tylko trzy spośród argumentów operacji określają jej zasięg.

Te trzy argumenty wymienione tu są w kolejności, w jakiej wpływają na operację:

1. objectType
2. restriction
3. recurse

Poniższy przykład pokazuje, jak mogą być zwracane niektóre obiekty, które ewidentnie nie spełniają zadanych ograniczeń, lecz są zwracane w ramach rekurencyjnej fazy operacji.

**Przykład:** Rekurencyjny opis kostek należących do schematu myschema:

Argumenty operacji:

```

objectType = "cube"
restriction = <restriction>
    <predicate property="schema" operator="=" value="myschema"/>
</restriction>
recurse = "yes"

```

Na początku zasięg operacji Describe zostaje ograniczony do obiektów typu kostka. Spośród obiektów kostki wybierane są tylko te, które należą do schematu myschema. Oprócz wszystkich wybranych obiektów kostki wybierane są obiekty, do których się one odwołują, przy czym obiekty te mają różne typy i ewentualnie różne schematy. Wszystkie wybrane obiekty są następnie zwracane w ramach odpowiedzi na żądanie kostki.

## Protokołowanie i śledzenie

### Śledzenie w czasie wykonywania dla interfejsu API programu DB2 Cube Views

Interfejs API umożliwia stosowanie trzech priorytetów śledzenia. Korzystając z pliku konfiguracyjnego, administrator może ustawić poziom śledzenia do protokołowania. Śledzenie w czasie wykonywania jest domyślnie wyłączone. Domyślny plik śledzenia nosi nazwę db2mdtrace.log.

Zwykle uruchamianie śledzenia nie jest potrzebne. Śledzenie może być konieczne, jeśli w interfejsie API występuje błąd i dział wsparcia firmy IBM zwróci się o dostarczenie pliku śledzenia.

W poniższej tabeli opisano różne poziomy śledzenia.

Tabela 60. Poziomy śledzenia

Poziom	Opis	Przykłady
Brak	<ul style="list-style-type: none"> <li>Śledzenie jest wyłączone</li> </ul>	Nie dotyczy
Wysoki	<ul style="list-style-type: none"> <li>Śledzone są zewnętrzne i wewnętrzne punkty wejścia i wyjścia interfejsu API</li> <li>Śledzony jest przepływ między komponentami</li> <li>Mogą być uwzględniane argumenty funkcji</li> </ul>	<ul style="list-style-type: none"> <li>Rozpoczęcia i zakończenia analizowania składni</li> <li>Rozpoczęcia i zakończenia operacji Create, Describe, Drop itd.</li> </ul>
Średni	<ul style="list-style-type: none"> <li>Śledzony jest przepływ sterowania między złożonymi funkcjami w wewnętrznym i zewnętrznym interfejsie API</li> <li>Śledzony jest przepływ między komponentami</li> <li>Uwzględniane są punkty śledzenia wysokiego poziomu</li> </ul>	<ul style="list-style-type: none"> <li>Ukazywane są złożone wywołania funkcji ze strony operacji Create</li> </ul>
Niski	<ul style="list-style-type: none"> <li>Śledzone są proste lub atomowe funkcje w wewnętrznym interfejsie API</li> <li>Uwzględniane są punkty śledzenia wysokiego poziomu i średniego poziomu</li> <li>Ten poziom stosuje się do większości punktów śledzenia</li> </ul>	<ul style="list-style-type: none"> <li>Ukazuje wywołania do metod pobierania lub ustawiania dotyczących obiektów metadanych</li> </ul>

Gdy śledzenie jest włączone i jego poziom jest ustawiony na wartość inną niż Brak, błędy występujące w interfejsie API mogą być rejestrowane zarówno w protokole błędów, jak i w protokole śledzenia, zależnie od ustawienia poziomu oraz istotności dla tych protokołów.

### Pliki protokołów interfejsu API programu DB2 Cube Views

Pliki protokołów interfejsu API są generowane na poziomie instancji DB2. Plik protokołu błędów nosi nazwę db2mderror.log, a plik protokołu śledzenia nosi nazwę db2mdtrace.log.

Dla danej instancji DB2, w której działa interfejs API programu DB2 Cube Views, pliki protokołów API będą generowane w katalogu danych diagnostycznych DB2 o nazwie DB2DIAG. Katalog diagnostyczny DB2 znajduje się zwykle w następującej ścieżce:

#### W systemie Windows

Katalog instancji DB2, taki jak `c:\sqllib\<moja_instancja>`

#### W systemie AIX

Katalog instancji DB2/db2dump, taki jak `~moja_instancja/sqllib/db2dump`

Domyślną ścieżkę DB2DIAG można zmienić, korzystając z ustawienia DB2DIAG db2 dbm cfg.

Program narzędziowy db2idrop czyści zawartość plików protokołów skojarzonych z instancją DB2. W przypadku gdy nie jest używana domyślna ścieżka DB2DIAG, program narzędziowy db2idrop nie jest w stanie wyczyścić plików protokołów interfejsu API programu DB2 Cube Views. Pliki protokołów, których wyczyszczenie programem narzędziowym db2idrop nie jest możliwe, muszą być czyszczone ręcznie. Błędy występujące, gdy funkcja API programu DB2 Cube Views ładuje plik konfiguracyjny, są protokołowane w pliku db2mdapi.log. Plik db2mdapi.log znajduje się w katalogu DB2DIAG, razem z innymi plikami protokołów API.

## Protokołowanie błędów

W interfejsie API rozróżniane są trzy poziomy istotności błędów. Domyślnym ustawieniem istotności jest medium (średnia), a domyślną nazwą pliku protokołu błędów jest mderror.log. Gdy podczas odczytywania pliku konfiguracyjnego wystąpi błąd, zostanie on zaprotokołowany w pliku o nazwie db2mdapi.log.

Poniższa tabela zawiera opis poziomów istotności błędów.

Tabela 61. Poziomy istotności błędów

Istotność	Opis	Przykłady
Brak	<ul style="list-style-type: none"> <li>Ignorowanie wszystkich błędów i ostrzeżeń</li> </ul>	Nie dotyczy
Wysoka (taką istotność ma większość błędów)	<ul style="list-style-type: none"> <li>Rejestrowane są tylko błędy krytyczne i nieodwracalne</li> <li>Powoduje zrzut stosu wywołań do protokołu</li> </ul>	<ul style="list-style-type: none"> <li>Wewnętrzny błąd programowy</li> </ul>
Średni	<ul style="list-style-type: none"> <li>Rejestrowane są błędy odwracalne po interwencji użytkownika</li> <li>Protokołowane są także błędy o wysokiej istotności</li> <li>Powoduje zrzut stosu wywołań do protokołu</li> </ul>	<ul style="list-style-type: none"> <li>Pomyłki użytkownika końcowego, takie jak próby utworzenia duplikatów obiektu</li> <li>Błędy przy sprawdzeniu poprawności metadanych</li> <li>Brak pamięci. Można powiększyć pamięć lub zmniejszyć jej użycie.</li> </ul>
Niski	<ul style="list-style-type: none"> <li>Rejestrowane są sytuacje, którym towarzyszą ostrzeżenia</li> <li>Protokołowane są także błędy o wysokiej i średniej istotności</li> <li>Błędy o niskiej istotności nie powodują zrzutu stosu wywołań</li> </ul>	<ul style="list-style-type: none"> <li>Ostrzeżenie o błędzie wewnętrznym</li> <li>Komunikaty informacyjne</li> </ul>

Gdy w konfiguracji interfejsu API protokołowanie błędów będzie ustawione na wysokiej lub średnie i wystąpi błąd o istotności wysokiej lub średniej, interfejs API wygeneruje stos wywołań zaczynający się w punkcie, w którym wystąpił błąd w interfejsie API. Ten stos

wywołań jest podobny do rejestrowanego przy śledzeniu na poziomie średnim, z tą różnicą, że dane są wysyłane do protokołu błędów, a nie do protokołu śledzenia.

## Scenariusze protokołowania i śledzenia

Przedstawione poniżej scenariusze protokołowania i śledzenia demonstrują, jak może się odbywać przechwytywanie błędów.

**Scenariusz 1** (poziom śledzenia: średni; istotność błędów: wysoka): Gdy wystąpi błąd o wysokiej istotności, zostanie on odnotowany zarówno w protokole błędów, jak i w protokole śledzenia.

```
<log>
  <trace level="medium" logFile="db2mdtrace.log" bufferSize="0" />
  <error level="high" logFile="db2mderror.log" bufferSize="0" />
</log>
```

**Scenariusz 2** (poziom śledzenia: średni; istotność błędów: niska): Gdy wystąpi błąd o niskiej istotności, zostanie odnotowany tylko w protokole błędów, ponieważ w protokole śledzenia możliwe są tylko wpisy na poziomie średnim lub wysokim.

```
<log>
  <trace level="medium" logFile="db2mdtrace.log" bufferSize="0" />
  <error level="low" logFile="db2mderror.log" bufferSize="0" />
</log>
```

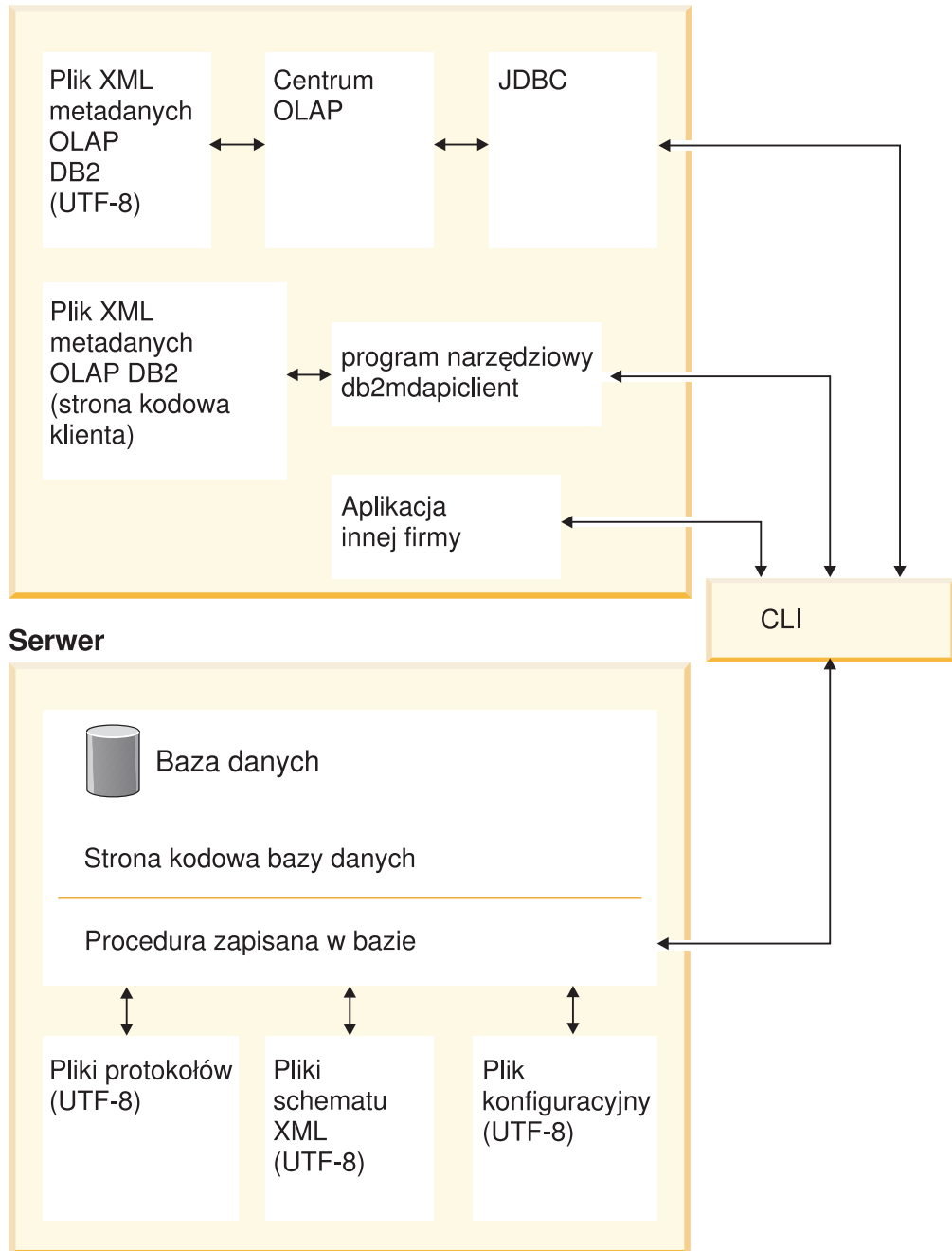
Błędy spowodowane brakiem zmiennych środowiskowych lub niepowodzeniami w dostępie do plików protokołów zwracane są za pośrednictwem parametru SQLSTATE procedury zapisanej w bazie do aplikacji klienckiej bazy danych. Gdy błąd wystąpi podczas przetwarzania pliku konfiguracyjnego, zostanie zaprotokołowany w pliku db2mdapi.log. Jeśli błąd wystąpi podczas otwierania określonych przez użytkownika plików protokołów, nie będą przechwytywane żadne błędy.

## Obsługa stron kodowych

W programie DB2 Cube Views używane są dwie strony kodowe: strona kodowa klienta DB2 (strona kodowa aplikacji) oraz strona kodowa bazy danych DB2. Informacje na temat określania strony kodowej klienta DB2 można znaleźć w podręczniku *Administration Guide: Planning* (Podręcznik administrowania: planowanie), w sekcji "Supported territory codes and code pages" (Obsługiwane kody terytorium i strony kodowe). Funkcja API zapisana w bazie programu DB2 Cube Views działa ze stroną kodową bazy danych DB2. Strona kodowa bazy danych DB2 jest ustawiana podczas tworzenia bazy danych. Strona kodowa klienta DB2 oraz strona kodowa bazy danych DB2 mogą być różne. Interfejs CLI dokonuje konwersji parametrów procedury zapisanej w bazie, przekazywanych jako duże obiekty znakowe (CLOB), na stronę kodową bazy danych tej procedury.

Poniższa ilustracja przedstawia, jak klient komunikuje się z serwerem przez interfejs poziomu wywołań (CLI). Interfejs CLI dokonuje konwersji stron kodowych klienta na stronę kodową bazy danych.

## Klient



Rysunek 41. Przepływ danych z różnych klientów używających stron kodowych lub formatu UTF-8 poprzez interfejs CLI programu DB2 do serwera bazy danych

Interfejs CLI obsługuje konwersję między stroną kodową aplikacji a stroną kodową bazy danych. Dane wysyłane z klienta programu DB2 Cube Views do interfejsu API są traktowane jako dane wejściowe. Dane wysyłane z interfejsu API do klienta programu DB2 Cube Views są traktowane jako dane wyjściowe. Dane wejściowe i wyjściowe są kodowane w stronie kodowej klienta DB2.

Poniżej przedstawiono specyfikację stron kodowych komponentów programu DB2 Cube Views. Centrum OLAP:



- Akceptuje i generuje tylko pliki XML programu DB2 Cube Views kodowane w standardzie UTF-8.
- Zwraca błąd, jeśli w wejściowym pliku XML programu DB2 Cube Views określone jest kodowanie inne niż UTF-8.
- Plik bez określonego kodowania traktuje jako zakodowany w formacie UTF-8.
- Korzystając z funkcji eksportu tworzy pliki XML programu DB2 Cube Views z jawną specyfikacją kodowania w standardzie UTF-8.

Program db2mdapiclient:

- O ile nie określono opcji -l, wejściowe pliki XML programu DB2 Cube Views traktuje jako zakodowane w stronie kodowej klienta DB2, dlatego ignoruje jawną specyfikację kodowania określoną w tych plikach.
- O ile nie określono opcji -l, generuje pliki XML programu DB2 Cube Views zakodowane w standardzie UTF-8 i nie umieszcza w nich jawnej specyfikacji kodowania.

Interfejs API procedury zapisanej w bazie:

- Parametry CLOB traktuje jako zakodowane w stronie kodowej klienta DB2.
- Ignoruje jawną specyfikację kodowania w wejściowych plikach XML programu DB2 Cube Views.
- Generuje pliki XML programu DB2 Cube Views bez jawnej specyfikacji kodowania.
- Wejściowe i wyjściowe pliki XML przetwarza z zastosowaniem strony kodowej bazy danych DB2.
- Generuje pliki protokołów interfejsu API, w tym wszystkie wbudowane komunikaty programu DB2, jako zakodowane w formacie UTF-8. Pliki protokołu interfejsu API nie są w formacie XML.
- Nie tworzy plików protokołów XML z jawnie określonym kodowaniem UTF-8.
- Pliki schematu XML używane przez interfejs API koduje w formacie UTF-8.
- Plik konfiguracyjny interfejsu API w języku XML koduje w formacie UTF-8.

W wypadku aplikacji innych firm: inne aplikacje, które bezpośrednio wywołują funkcje API programu DB2 Cube Views muszą przekazywać i przyjmować jako parametry pliki XML zakodowane w stronie kodowej klienta DB2.

---

## Tabele metadanych i pliki schematu XML programu DB2 Cube Views

Niektóre funkcje w interfejsie API nie są obsługiwane, jeśli numer wersji nie jest poprawny.

### Wersje tabel metadanych

Interfejs API działa tylko wtedy, gdy jest połączony z bazą danych DB2, która zawiera bieżący zestaw tabel metadanych programu DB2 Cube Views. Bieżącym oznaczeniem wersji programu DB2 Cube Views, wersja 8.2, jest 8.2.0.1.0. Program DB2 Cube Views, wersja 8.1, ma numer wersji 8.1.2.0. Numer wersji tabel katalogu metadanych programu DB2 Cube Views jest przechowywany w tabeli SYSINFOVERSION.

Więcej informacji na temat wersji tabel metadanych zawiera sekcja “Przegląd programu narzędziowego db2mdapiclient” na stronie 165.

Pliki schematu XML programu DB2 Cube Views, czyli pliki XSD, stosowane są w interfejsie API programu DB2 Cube Views. Pliki schematu XML wykorzystywane są przez funkcję API zapisaną w bazie o nazwie DB2INFO.MD\_MESSAGE.

## Wersje plików schematu XML

Wszystkie dokumenty XML przekazywane do i z interfejsu API metadanych muszą mieć numer wersji. Numer wersji stanowi dla procedury zapisanej w bazie informację, jakiego schematu XML używa klient. Pliki schematu XSD określają, które numery wersji są akceptowane w konkretnych operacjach.

Bieżącym oznaczeniem wersji programu DB2 Cube Views, wersja 8.2, jest 8.2.0.1.0.

Niżej wymienione pliki schematu zawierają informacje o procedurze zapisanej w bazie md\_message().

### Plik db2md\_parameter.xsd

Ten plik zawiera informacje o operacjach żądań i odpowiedzi. Plik ten współdziała z wersjami 8.1.2.1.0 oraz 8.2.0.1.0.

### Plik db2md\_metadata.xsd

Ten plik zawiera plik db2md\_types.xsd i określa tylko elementy metadanych. Plik ten nie zawiera informacji o wersji.

### Plik db2md\_types.xsd

Ten plik zawiera informacje o wszystkich obiektach metadanych. Plik ten współdziała z wersjami 8.1.2.1.0 oraz 8.2.0.1.0.

Interfejs API jest w stanie obsługiwać wcześniejszą wersję plików schematu, wersję 8.1.2.1.0, niemniej obsługuje tylko operacje Describe i Translate. Interfejs API zwraca błąd po otrzymaniu żądania operacji w wersji 8.1.2.1.0 innej niż Describe lub Translate.

---

## Plik konfiguracyjny programu DB2 Cube Views

Interfejs API może być konfigurowany na poziomie instancji DB2. Można modyfikować parametry w pliku konfiguracyjnym o nazwie db2md\_config.xml .

Każda instalacja programu DB2 Cube Views ma domyślny plik konfiguracyjny w katalogu ścieżka\_instalacyjna\_db2/cfg. Na przykład w systemach Windows domyślny plik konfiguracyjny może się znajdować w katalogu c:\sqllib\cfg, w systemie AIX będzie on umieszczony w katalogu /usr/opt/db2\_08\_01/cfg.

Każda instancja DB2, w której uruchomiony jest program DB2 Cube Views, ma fizyczną kopię pliku db2md\_config.xml w katalogu ścieżka\_instancji\_db2. Na przykład w systemie Windows kopia ta może znajdować się w katalogu c:\sqllib\moja\_instancja, a w systemie AIX w katalogu ~moja\_instancja/sqllib.

Program db2icrt kopiuje domyślny plik konfiguracyjny do katalogu ścieżka\_instancji\_db2 i tworzy nową instancję. W przypadku instancji DB2 utworzonych przed zainstalowaniem programu DB2 Cube Views można ręcznie skopiować plik konfiguracyjny do katalogu instancji, jeśli program instalacyjny nie wykonał tego automatycznie. Jeśli interfejs API nie odnajdzie pliku konfiguracyjnego w katalogu instancji, podejmie próbę skopiowania tam domyślnego pliku konfiguracyjnego.

Plik konfiguracyjny o nazwie db2md\_config.xml służy do określania ustawień protokołowania błędów i śledzenia w czasie wykonania. Modyfikując zawartość pliku konfiguracyjnego, administrator może określać poziom śledzenia, istotność protokołowanych błędów i wielkość buforu (w bajtach) służącego do protokołowania.

Strukturę pliku konfiguracyjnego db2md\_config.xml definiuje plik schematu XML db2md\_config.xsd. Poniżej przedstawiono przykładową zawartość pliku konfiguracyjnego.

```
<olap:config xmlns:olap="http://www.ibm.com/olap"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://www.ibm.com/olap db2md_config.xsd">
  <log>
    <trace level="none" logFile="db2mdtrace.log" bufferSize="0"/>
    <error level="medium" logFile="db2mderror.log" bufferSize="0"/>
  </log>
</olap:config>
```

---

## Format obiektów metadanych

Schemat XML programu DB2 Cube Views definiuje podstawowe elementy XML, które można bezpośrednio odwzorować na obiekty w modelu obiektów metadanych programu DB2 Cube Views. Złożone struktury metadanych mają z kolei reprezentację w postaci sekwencji tych elementów podstawowych. Powiązania między obiektami wewnątrz złożonych struktur metadanych są rozpoznawane przy użyciu odwołań (przez nazwy) między elementami bazowymi.

Jako przykład odwołania przez nazwę można podać sytuację, w której element kostki zawiera odwołanie do elementu wymiaru. Poniżej przedstawiono przykładowy typ danych dla obiektu kostki - zgodnie z definicją ze schematu XML. W tym przykładzie pokazany został tylko opis tekstowy; w aplikacji odpowiednie informacje są przedstawiane w języku XML.

```
cube
->odwołanie do modelu kostki
->odwołania do wymiaru kostki
->odwołanie do faktów kostki
->widok
```

W wypadku obiektu kostki odwołania do innych typów obiektów są w całości zawarte w elemencie podstawowym reprezentującym kostkę. W nierekurencyjnych operacjach pobierania przedstawiane są sekwencje obiektów kostki (i tylko obiektów kostki). W rekurencyjnych operacjach pobierania przedstawiane są informacje o obiektach kostki oraz informacje o innych obiektach (różnego typu), do których odwołują się wskazane kostki.

Kolejność obiektów jest zdefiniowana przez schemat XML programu DB2 Cube Views. Obiekty tego samego typu w zasięgu pojedynczej operacji (na przykład obiekty kostki) są grupowane. Wewnątrz tych grup na kolejność elementów mają wpływ odwołania między obiektami tego samego typu. Obiekty, do których występują odwołania, muszą występować przed obiektami, które się do nich odwołują. Kolejność tych grup jest następująca:

1. attribute
2. join
3. attributeRelationship
4. level
5. cubeLevel
6. hierarchy
7. cubeHierarchy
8. dimension
9. cubeDimension
10. measure
11. facts
12. cubeFacts

13. cubeModel

14. cube

Kolejność grup obiekt-typ jest niezależna od powiązań między obiektami. Fakt, że atrybuty i łączenia odgrywają inną rolę, gdy są powiązane z obiektami innych typów, nie wpływa na ich kolejność w formacie skojarzenia.

Aby uzyskać przykładowy plik XML przedstawiający typ i kolejność metadanych, należy przejrzeć plik metadanych XML dołączony do bazy danych CVSAMPLE.

---

## Rozdział 8. Pliki przykładowe

---

### Przegląd programu narzędziowego db2mdapiclient

Program narzędziowy db2mdapiclient ma służyć jako przykładowy kod źródłowy podczas tworzenia kodu aplikacji dla programu DB2 Cube Views Multidimensional Services. Za pomocą programu narzędziowego można importować, eksportować i optymalizować obiekty metadanych.

**Uwaga:** Program narzędziowy dostarczany z programem DB2 Cube Views, wersja 8.2, został udoskonalony i nie jest dokładnie taki sam, jak przykładowy kod źródłowy w pliku `sqllib\samples\olap\client\db2mdapiclient.cpp`.

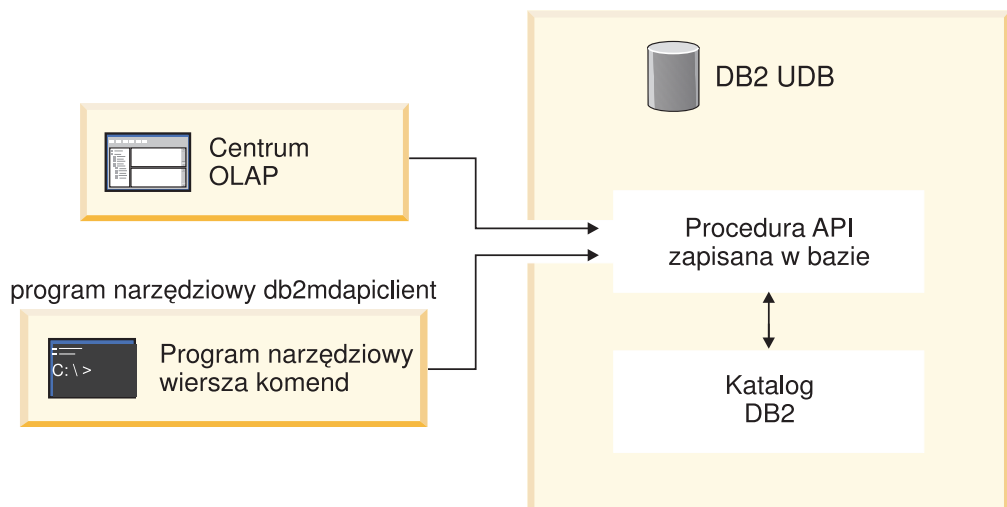
Programu narzędziowego db2mdapiclient można używać do wykonywania dowolnych operacji obsługiwanych przez procedurę zapisaną w bazie DB2 Cube Views - MD\_MESSAGE() - opisanych w następującej tabeli:

*Tabela 62. Operacje.* Operacje na danych wielowymiarowych wykonywane przez program narzędziowy db2mdapiclient

Zadanie	Operacja
Eksportowanie obiektów metadanych do pliku	DESCRIBE
Importowanie obiektów metadanych z pliku	CREATE lub IMPORT
Zmianianie istniejących obiektów metadanych	ALTER lub RENAME
Usuwanie istniejących obiektów metadanych	DROP
Sprawdzenie poprawności istniejących obiektów metadanych	VALIDATE
Optymalizowanie modelu kostki	ADVISE
Migrowanie metadanych XML z wersji 8.1 do wersji 8.2	TRANSLATE

W programie narzędziowym db2mdapiclient używane są pliki, w których przechowywany jest kod XML przekazywany do procedury zapisanej w bazie MD\_MESSAGE() i od niej otrzymywany.

Podczas importu program narzędziowy db2mdapiclient zwykle korzysta z pliku XML wygenerowanego przez most programu DB2 Cube Views lub wyeksportowanego z Centrum OLAP. Podczas eksportowania program narzędziowy db2mdapiclient tworzy plik XML, który może być używany przez program narzędziowy mostu DB2 Cube Views do dodawania metadanych do bazy danych lub narzędzia OLAP. Ważne jest kodowanie znaków używane do wprowadzania i wyprowadzania plików XML. Więcej informacji na temat kodowania znaków zawiera "Obsługa stron kodowych" na stronie 159.



Rysunek 42. Przesyłanie metadanych. Program narzędziowy db2mdapiclient oraz Centrum OLAP przesyłają dane za pośrednictwem programu Multidimensional Services

## Komenda db2mdapiclient: manipulowanie obiektami metadanych

Program narzędziowy db2mdapiclient można uruchomić z wiersza komend.

Aby zapoznać się z listą parametrów komendy db2mdapiclient, można w wierszu komend wpisać db2mdapiclient. Składnia komendy **db2mdapiclient** jest następująca:

```
db2mdapiclient -d nazwa_bazy_danych [-u użytkownik]
                [-p hasło] -i żądanie.xml -o odpowiedź.xml
                [-m metadane_wejściowe.xml] [-n metadane_wyjściowe.xml]
                [-a wielkość_buforu_parametrów]
                [-b wielkość_buforu_metadanych] [-v] [-l] [-h]
```

-d *nazwa\_bazy\_danych* określa nazwę bazy danych.

-u *użytkownik* określa identyfikator użytkownika, który ma być używany do nawiązania połączenia z określoną bazą danych.

-p *hasło* określa hasło, które ma być używane do nawiązania połączenia z określoną bazą danych.

-i *żądanie.xml* określa wymagany plik wejściowy z operacją, która ma zostać wykonana.

-o *odpowieź.xml* określa wymagany plik wyjściowy z kodem XML odpowiedzi uzyskanej z procedury zapisanej w bazie MD\_MESSAGE(). Ten kod XML jest zwracany w trzecim argumencie procedury zapisanej w bazie MD\_MESSAGE().

-m *metadane\_wejściowe.xml* to plik wejściowy z kodem XML obiektu metadanych programu DB2 Cube Views. Ta opcja jest wymagana w wypadku operacji tworzenia (Create) oraz importowania (Import).

-n *metadane\_wyjściowe.xml* to opcjonalny plik wyjściowy z kodem XML obiektu metadanych odpowiedzi (jeśli ma zastosowanie) z drugiego argumentu procedury zapisanej w bazie MD\_MESSAGE().

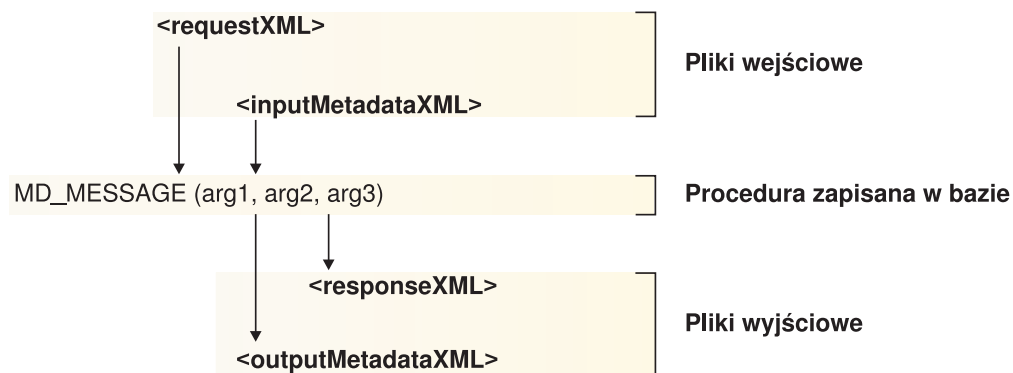
-a *wielkość\_buforu\_parametrów* określa wielkość buforu dla parametrów. Wartość domyślna wynosi 1048576 bajtów.

-b *wielkość\_buforu\_metadanych* określa wielkość buforu dla informacji obiektu metadanych. Wartość domyślna wynosi 1048576 bajtów.

-v określa, że podczas przetwarzania komendy mają być wyświetlane dodatkowe informacje.

- l wyłącza obsługę standardu UTF-8 i określa, że pliki wejściowy i wyjściowy są w lokalnej stronie kodowej.
- h powoduje wyświetlenie informacji o składni.

Poniższy rysunek pokazuje, w jaki sposób procedura zapisana w bazie MD\_MESSAGE() jest powiązana z dwoma plikami wejściowymi i dwoma plikami wyjściowymi:



Rysunek 43. Jak procedura zapisana w bazie obsługuje dwa pliki wejściowe i wyjściowe z programu narzędziowego db2mdapiclient

Na przykład, aby zaimportować obiekty metadanych programu DB2 Cube Views dla bazy danych CVSAMPLE w systemie Windows, zmień katalog na ..\SQLLIB\samples\olap\xml\input i wprowadź następującą komendę:

```
db2mdapiclient -d CVSAMPLE -u db2admin
-p mypasswd -i create.xml
-o myresponse.xml -m ..\..\CVSAMPLE\CVSampleMetadata.xml.xml
```

Opis udostępnianych plików przykładowych zawiera "Przykładowe pliki API" na stronie 168. Więcej informacji na temat operacji Usług wielowymiarowych zawiera sekcja "Przegląd interfejsu API programu DB2 Cube Views" na stronie 131.

## Przykładowe pliki bazy danych

Wszystkie niżej wymienione pliki, powiązane z bazą danych CVSAMPLE, znajdują się w katalogu \SQLLIB\samples\olap\cvsample\.

### CVSampleMetadata.xml

Plik XML zawierający metadane CVSAMPLE. Plik ten służy do importowania metadanych CVSAMPLE przy użyciu Centrum OLAP i programu narzędziowego db2mdapiclient.

### CVSampleTables.sql

Skrypt SQL służący do zapełniania tabel bazy danych CVSAMPLE.

### FAMILY.txt, LINE.txt, LOCATION.txt, PRODUCT.txt, SALESFACT.txt, STORE.txt, TIME.txt

Zestaw plików tekstowych zawierających dane tabel bazy danych CVSAMPLE.

### CVSampleExplain.sql

Skrypt SQL służący do określania, czy program DB2 przekierowuje zapytanie do tabeli podsumowań.

Również katalog \SQLLIB\samples\olap\xml\input zawiera pliki związane z bazą danych CVSAMPLE.

### Create.xml

Plik XML z operacją CREATE. Plik ten służy do załadowywania przykładu przy użyciu programu narzędziowego db2mdapiclient.

---

## Przykładowe pliki API

Przykładowe pliki API dla bazy danych CVSAMPLE są dostarczane wraz z programem DB2 Cube Views. Korzystając z tych plików można zrealizować przykładowe zadania z zastosowaniem programu narzędziowego db2mdapiclient. Program narzędziowy db2mdapiclient stanowi cienkie opakowanie interfejsu zapisanych w bazie procedur Multidimensional Services. Program ten jest udostępniany jako przykładowy kod źródłowy ilustrujący programowanie aplikacji przy użyciu interfejsu API. Kod źródłowy znajduje się w pliku `\SQLLIB\samples\olap\client\db2mdapiclient.cpp`.

Zawartość przykładowych plików wymienionych dla każdego zadania przekazuje się do procedury zapisanej w bazie MD\_MESSAGE() jako jej parametry. Niekiedy parametr metadanych do procedury zapisanej w bazie jest ignorowany na wejściu lub nie jest zwracany na wyjściu, co w niżej przedstawianych zadaniach jest sygnalizowane opisem `<puste>`. Gdy nie są potrzebne metadane wejściowe, w komendzie narzędziowej db2mdapiclient nie określa się opcji `-m`. Gdy nie są potrzebne metadane wyjściowe, w komendzie narzędziowej db2mdapiclient nie określa się opcji `-n`.

Więcej informacji na temat korzystania z procedury zapisanej w bazie MD\_MESSAGE() w programie narzędziowym db2mdapiclient można znaleźć w sekcji “Przegląd programu narzędziowego db2mdapiclient” na stronie 165. Więcej informacji na temat korzystania z samej procedury zapisanej w bazie MD\_MESSAGE() można znaleźć w sekcji “Procedura zapisana w bazie programu DB2 Cube Views” na stronie 133.

Wszystkie przykładowe pliki znajdują się w katalogu `\SQLLIB\samples\olap\xml\`. Korzystając z tych przykładowych plików, można zrealizować niżej przedstawione przykładowe zadania:

### DROP

Te pliki służą do usuwania wszystkich obiektów metadanych znajdujących się w katalogu metadanych. W tym przykładzie przyjęto założenie, że katalog metadanych nie jest pusty. Jeśli katalogi metadanych są puste, zgłoszony zostanie komunikat ostrzegawczy, że nie znaleziono obiektów do tej operacji.

#### Pliki zawierające parametry wejściowe

- **Żądanie:** `input\Drop.xml`
- **Metadane:** `<puste>`

W programie narzędziowym db2mdapiclient nie należy określać opcji `-m`.

#### Pliki zawierające parametry wyjściowe

- **Odpowiedź:** `output\Drop_response.xml`
- **Metadane:** `<puste>`

W programie narzędziowym db2mdapiclient nie należy określać opcji `-n`.

### CREATE

Te pliki służą do tworzenia obiektów metadanych w katalogu metadanych. W tym przykładzie przyjęto założenie, że katalog metadanych jest pusty.

#### Pliki zawierające parametry wejściowe

- **Żądanie:** `input\Create.xml`



- **Metadane:** input\CVSampleTestMetadata.xml

**Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Create\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

**DESCRIBE**

Te pliki służą do opisanie wszystkich obiektów metadanych w katalogu metadanych.

**Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Describe.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

**Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Describe\_response.xml
- **Metadane:** <plik XML metadanych>

**DESCRIBE (z ograniczeniem)**

Te pliki służą do rekurencyjnego opisywania kostki wyników sprzedaży CVSample.Daily Sales. W tym przykładzie przyjęto założenie, że został już zaimportowany plik metadanych CVSampleMetadata.xml dla bazy danych CVSAMPLE.

**Pliki zawierające parametry wejściowe**

- **Żądanie:** input\CVSampleDescribe\_restricted.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

**Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\DescribeRestricted\_response.xml
- **Metadane:** <plik XML metadanych>

**ALTER**

Te pliki służą do dokonywania zmian w obiektach metadanych w katalogu metadanych. W tym przykładzie przyjęto założenie, że został już zaimportowany plik metadanych CVSampleMetadata.xml dla bazy danych CVSAMPLE.

**Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Alter.xml
- **Metadane:** input\CVSampleAlter.xml

**Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Alter\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

**RENAME**

Te pliki służą do zmiany nazwy modelu kostki CVSAMPLE.Sales Model. Nazwa modelu kostki zostaje zmieniona na CVSAMPLE.Sales Model (2004). W tym przykładzie przyjęto założenie, że został już zaimportowany plik metadanych CVSampleMetadata.xml dla bazy danych CVSAMPLE.

#### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\CVSampleRename.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Rename\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

### **VALIDATE**

Te pliki służą do sprawdzania poprawności wszystkich obiektów metadanych w katalogu metadanych z zastosowaniem trybu optymalizacji.

#### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Validate.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Validate\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

### **VALIDATE (z ograniczeniem)**

Te pliki służą do sprawdzania poprawności kostki CVSAMPLE.Daily Sales z zastosowaniem trybu optymalizacji. W tym przykładzie przyjęto założenie, że został już zaimportowany plik metadanych CVSampleMetadata.xml dla bazy danych CVSAMPLE.

#### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\CVSampleValidate\_restricted.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\ValidateRestricted\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

### **IMPORT z zastosowaniem trybu *create new - ignore collisions***

Te pliki służą do importowania obiektów metadanych do katalogu metadanych z zastosowaniem trybu importowania *create new - ignore collisions*. W tym przykładzie przyjęto założenie, że katalog metadanych jest pusty.

#### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Import\_mode1.xml
- **Metadane:** CVSampleTestMetadata.xml

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Import\_mode1\_response.xml

- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

#### **IMPORT z zastosowaniem trybu *create new - replace collisions***

Te pliki służą do importowania obiektów metadanych do katalogu metadanych z zastosowaniem trybu importowania *create new - replace collisions*. W tym przykładzie przyjęto założenie, że już zostało zrealizowane zadanie IMPORT w trybie *create new - ignore collisions*.

##### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Import\_mode2.xml
- **Metadane:** CVSSampleTestMetadata.xml

##### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Import\_mode2\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

#### **IMPORT z zastosowaniem trybu *create new - abort on collision***

Te pliki służą do importowania obiektów metadanych do katalogu metadanych z zastosowaniem trybu importowania *create new - abort on collision*. W tym przykładzie przyjęto założenie, że już zostało zrealizowane zadanie IMPORT w trybie *create new - replace collisions*.

##### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Import\_mode3.xml
- **Metadane:** CVSSampleTestMetadata.xml

##### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Import\_mode3\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

#### **IMPORT z zastosowaniem trybu *create new - report collisions***

Te pliki służą do importowania obiektów metadanych do katalogu metadanych z zastosowaniem trybu importowania *create new - report collisions*. W tym przykładzie przyjęto założenie, że już zostało zrealizowane zadanie IMPORT w trybie *create new - abort on collision*.

##### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Import\_mode4.xml
- **Metadane:** CVSSampleTestMetadata.xml

##### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Import\_mode4\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

#### **TRANSLATE**

Te pliki służą do konwersji metadanych programu Cube Views, wersja 8.1, do metadanych programu Cube Views, wersja 8.2.

##### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\Translate.xml
- **Metadane:** CVSSampleTranslateMetadata.xml

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Translate\_response.xml
- **Metadane:** <plik XML metadanych>

#### **ADVISE**

Te pliki służą do uruchamiania Doradcy optymalizacyjnego dla modelu kostki CVSAMPLE.Sales Model. W tym przykładzie przyjęto założenie, że już został zaimportowany plik metadanych CVSampleMetadata.xml dla bazy danych CVSAMPLE i nie zostało zrealizowane wcześniej wymienione zadanie zmiany nazwy (Rename).

#### **Pliki zawierające parametry wejściowe**

- **Żądanie:** input\CVSampleAdvise.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -m.

#### **Pliki zawierające parametry wyjściowe**

- **Odpowiedź:** output\Advise\_response.xml
- **Metadane:** <puste>

W programie narzędziowym db2mdapiclient nie należy określać opcji -n.

---

## Dodatek. Komunikaty

Niżej wymienione komunikaty mogą być zgłaszane przez serwer, interfejs API oraz Centrum OLAP programu DB2 Cube Views.

**Socket error:** Wielokrotne otwieranie i zamykanie połączenia z bazą danych może powodować błąd gniazda. W sporadycznych przypadkach błąd gniazda może wystąpić po uruchomieniu programu DB2 Cube Views z programem DB2 Universal Database Enterprise Server Edition, wersja 8.1.2, w środowisku partycjonowanej bazy danych w systemie Windows 2000 Advanced Server. Błąd może wystąpić, gdy poniższa procedura zostanie szybko powtórzona ponad 10000 razy w ramach jednej sesji systemu Windows:

1. Otwórz połączenie z bazą danych DB2.
2. W celu wykonania operacji na metadanych wywołaj zapisaną w bazie procedurę programu DB2 Cube Views.
3. Zamknij połączenie z bazą danych.

Obejście tego problemu polega na ponownym uruchomieniu stacji roboczej Windows w celu reaktywacji gniazda.

---

## SQLSTATE, API i inne komunikaty serwera

### Stany SQL interfejsu API

#### **01HQ1: Patrz wynikowy plik XML i protokoły serwera.**

##### **Objaśnienie**

Wywołanie procedury zapisanej w bazie zostało ukończone, lecz stwierdzono błędy podczas wykonywania jednej z operacji na metadanych.

##### **Działanie użytkownika**

W celu uzyskania dodatkowych informacji sprawdź zawartość parametrów wynikowych procedury. Informacje można też znaleźć w odpowiednich wpisach w protokole serwera.

#### **38Q00: Więcej informacji zawierają protokoły serwera.**

##### **Objaśnienie**

Wywołanie procedury zapisanej w bazie zakończyło się niepowodzeniem. Żądane operacje na metadanych nie zostały wykonane. W parametrach wynikowych procedury nie zostały przekazane żadne informacje.

##### **Działanie użytkownika**

Aby uzyskać dodatkowe informacje, należy zapoznać się z wpisami w protokole serwera.

### **38Q01: Ścieżka instalacji nie jest znana.**

#### **Objaśnienie**

Wywołanie procedury zapisanej w bazie zakończyło się niepowodzeniem, ponieważ proces procedury nie może ustalić katalogu instalacyjnego programu DB2. Żądane operacje na metadanych nie zostały wykonane. W parametrach wynikowych procedury nie zostały przekazane żadne informacje.

#### **Działanie użytkownika**

W systemie operacyjnym Windows należy zadbać o to, by zmienna środowiskowa DB2PATH miała prawidłową wartość - domyślnie lub w wyniku ustawienia przez użytkownika. Uruchom ponownie menedżera bazy danych i ponownie wywołaj procedurę. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **38Q02: Nie można otworzyć pliku protokołu serwera.**

#### **Objaśnienie**

Wywołanie procedury zapisanej w bazie zakończyło się niepowodzeniem, ponieważ proces procedury nie mógł otworzyć do zapisu przynajmniej jednego z używanych przez procedurę plików protokołu. Żądane operacje na metadanych nie zostały wykonane. W parametrach wynikowych procedury nie zostały przekazane żadne informacje.

#### **Działanie użytkownika**

Zadbaj o to, by w odpowiednim systemie plików można było tworzyć i otwierać (do odczytu i zapisu) pliki protokołów wskazane w pliku konfiguracyjnym procedury zapisanej w bazie (na przykład olap\_config.xml). Jeśli pliki protokołów jeszcze nie istnieją, procedura zapisana w bazie próbuje je utworzyć. W systemie AIX zadbaj o to, by pliki protokołów były dostępne do odczytu i zapisu przez identyfikator chronionego użytkownika bazy danych.

### **38Q03: Brak parametru wejściowego metadanych.**

#### **Objaśnienie**

Wywołanie procedury zapisanej w bazie zakończyło się niepowodzeniem, ponieważ wywołana operacja na metadanych wymaga przekazania do procedury metadanych w postaci parametrów, tymczasem w parametrach procedury nie przekazano potrzebnych metadanych. W parametrach wynikowych procedury nie zostały przekazane żadne informacje.

#### **Działanie użytkownika**

Przy następnym wywołaniu procedury zapisanej w bazie podaj parametr w postaci metadanych wymaganych przez operację żadaną w procedurze.

### **38Q04: [typ\_błędu] BŁĄD: Za mały bufor wyjściowy odpowiedzi.**

#### **Objaśnienie**

Wywołanie procedury zapisanej w bazie nie powiodło się, ponieważ bufor parametrów wynikowych jest zbyt mały na potrzeby zwracanej struktury CLOB. W parametrach wynikowych procedury nie zostały przekazane żadne informacje.

### Działanie użytkownika

Ponownie skataloguj procedurę zapisaną w bazie, podając większą wartość parametru określającego wielkość danych wyjściowych.

## Wspólne

### Kody powodzenia

#### **0: Operacja zakończona pomyślnie. Nie napotkano żadnych błędów.:**

##### **Objaśnienie**

Żądana operacja na metadanych została ukończona pomyślnie. Podczas wykonywania operacji nie stwierdzono żadnych błędów.

### Działanie użytkownika

Jest to tylko informacja do wiadomości użytkownika. Nie jest wymagane żadne działanie.

#### **1: Operacja zakończona. Zwrócone zostały dodatkowe informacje.:**

##### **Objaśnienie**

Żądana operacja na metadanych została ukończona. W wyniku operacji zwrócone zostały dodatkowe informacje, mogące opisywać ewentualne ostrzeżenia lub błędy.

### Działanie użytkownika

Ewentualne dodatkowe informacje zwrotne zawarte są w elemencie INFO.

#### **2: Operacja zakończona. Nie wprowadzono żadnych zmian w metadanych.:**

##### **Objaśnienie**

Żądana operacja na metadanych została ukończona. W wyniku operacji nie zostały wprowadzone żadne zmiany w metadanych w katalogu bazy danych.

### Działanie użytkownika

Jeśli wprowadzenie zmian metadanych w katalogu bazy danych jest wymagane, należy ponownie wydać żądanie operacji, tym razem w innym trybie.

## **Najczęściej występujące kody zewnętrznych błędów/ostrzeżeń**

#### **100: Nie powiodła się próba przydzielenia pamięci dla operacji *operacja*. Zapewnij dostępność pamięci.:**

##### **Objaśnienie**

Podczas wykonywania żądanej operacji na metadanych w procedurze zapisanej w bazie nie powiodło się przydzielenie wymaganej liczby segmentów pamięci.

### Działanie użytkownika

Zwiększ ilość pamięci dostępnej na użytek chronionego procesu procedury zapisanej w bazie.

#### **101: Podczas przetwarzania obiektu *nazwa obiektu* wystąpił błąd wewnętrzny.:**

### **Objaśnienie**

Podczas wykonywania żądanej operacji na metadanych wystąpił nieoczekiwany błąd wewnętrzny.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z Obsługą klienta firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**102: Bufor wyjściowy o wielkości *wielkość\_buforu* jest zbyt mały. Zmień wielkość buforu tak, aby wynosiła przynajmniej *wielkość*.**

### **Objaśnienie**

Bufor parametrów udostępniony na użytek procedury zapisanej w bazie jest zbyt mały, aby pomieścić strukturę CLOB generowaną przez tę procedurę.

### **Działanie użytkownika**

Jeśli to możliwe, ponownie wpisz do katalogu procedurę zapisaną w bazie, stosując tym razem większe ustawienia parametrów OUT i INOUT. Innym rozwiązaniem jest takie ograniczenie zapytań, aby procedura zwracała mniejszą ilość danych.

**103: Nie istnieje ważna licencja na ten produkt.:**

### **Objaśnienie**

Nie można wykonać żadnych operacji na metadanych ze względu na brak ważnej licencji na tę instalację produktu.

### **Działanie użytkownika**

Zainstaluj w systemie ważną licencję na produkt lub skontaktuj się z działem wsparcia lub z działem sprzedaży oprogramowania firmy IBM w celu nabycia nowej licencji na produkt.

**104: Wystąpił błąd wewnętrzny. Zwrócone zostały następujące elementy zmienne: *element0, element1, element2, element3*.**

### **Objaśnienie**

Podczas wykonywania żądanej operacji na metadanych wystąpił nieoczekiwany błąd wewnętrzny.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**599: Nie wykonano operacji.:**

### **Objaśnienie**

Przed wykonaniem tej operacji wystąpił błąd. W efekcie operacja nie została wykonana.



### Działanie użytkownika

Sprawdź wyniki poprzednich operacji na metadanych wykonanych w trakcie tego samego wywołania procedury zapisanej w bazie. Informacje można też znaleźć w odpowiednich wpisach w protokole serwera. Po usunięciu nieprawidłowości, które uniemożliwiły wykonanie tej operacji, wywołaj procedurę ponownie, żądając w niej wykonania tych samych operacji na metadanych.

### Najczęściej występujące kody wewnętrznych błędów/ostrzeżeń

**600: Parametr wejściowy *nazwa\_parametru* jest niepoprawny. Komunikat: *komunikat*. Sprawdź parametr i spróbuj ponownie.:**

#### Objaśnienie

Jeden z parametrów przekazanych jako parametr wejściowy metody wewnętrznej dla tej procedury jest nieprawidłowy.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**601: Parametr wejściowy *nazwa\_parametru* ma wartość NULL.:**

#### Objaśnienie

Jeden z parametrów przekazanych jako parametr wejściowy metody wewnętrznej dla tej procedury ma niepoprawną wartość NULL.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**602: Parametr *nazwa\_parametru* o wartości *wartość* znajduje się poza dozwolonym zakresem *granica\_zakresu1*, *granica\_zakresu2*.:**

#### Objaśnienie

Jeden z parametrów przekazanych jako parametr wejściowy metody wewnętrznej dla tej procedury ma wartość spoza dozwolonego zakresu.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**603: Łącuch Unicode *łańcuch* jest pozbawiony sensu lub niepoprawny. Może to być objawem problemów z pamięcią.:**

#### Objaśnienie

Łącuch Unicode w procedurze zapisanej w bazie jest niepoprawny. Może to wskazywać na problem z pamięcią procedury lub systemu. Innym możliwym powodem jest ładowanie przez procedurę niewłaściwych wersji bibliotek ICU.

### Działanie użytkownika

Zadbaj o dostępność pamięci w ilości adekwatnej do ilości danych przetwarzanych przez procedurę. Upewnij się, że procedura ładuje właściwą wersję bibliotek ICU. Warto też sprawdzić prawidłowość ustawienia zmiennej środowiskowej określającej ścieżkę wyszukiwania bibliotek czasu wykonywania.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **604: Nie powiodła się próba przekształcenia zawartości łańcucha *łańcuch* z formatu Unicode do aktywnej strony kodowej *strona\_kodowa*.**

##### **Objaśnienie**

Konwersja obiektu łańcucha Unicode do łańcucha z odmiennym kodowaniem zakończyła się niepowodzeniem. Może to wskazywać na problem z pamięcią procedury lub systemu. Inną przyczyną może być problem z konwersją strony kodowej w systemie lub ładowanie niewłaściwej wersji bibliotek ICU przez procedurę.

### Działanie użytkownika

Upewnij się, że w systemie serwera bazy danych zainstalowane są wymagane pliki konwersji stron kodowych ICU. Zadbaj o dostępność pamięci w ilości adekwatnej do ilości danych przetwarzanych przez procedurę. Upewnij się, że procedura ładuje właściwą wersję bibliotek ICU. Może też być konieczne sprawdzenie prawidłowości ustawienia zmiennej środowiskowej określającej ścieżkę wyszukiwania bibliotek czasu wykonywania.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **605: Pamięć przydzielona pod adresem *bufor\_pamięci* musi zostać zwolniona.**

##### **Objaśnienie**

Metoda wewnętrzna procedury zapisanej w bazie zwróciła bufor pamięci, który musi zostać zwolniony przez następną metodę wewnętrzną.

### Działanie użytkownika

Metoda wewnętrzna względem procedury zapisanej w bazie musi zwolnić zwrócony bufor pamięci. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **606: Konwersja typu XMLCh na UChar dla elementu *UChar* nie powiodła się.**

##### **Objaśnienie**

Próba konwersji znaku XMLCh na znak UChar zakończyła się niepowodzeniem.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**607: Bufor wejściowy o wielkości *wielkość* jest zbyt mały. Zmień wielkość buforu tak, aby wynosiła przynajmniej *nowa\_wielkość*.**

**Objaśnienie**

Wewnętrzny bufor pamięci procedury jest zbyt mały, by pomieścić tekst wymaganego komunikatu.

**Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**608: Typ *nazwa\_procedury\_zapisanej\_w\_bazie* nie jest poprawny w bieżącym kontekście.**

**Objaśnienie**

Podczas przetwarzania procedury zapisanej w bazie napotkano nieoczekiwany typ.

**Działanie użytkownika**

Dodatkową pomoc dotyczącą identyfikatora i tekstu statusu dla operacji wykonywanej na metadanych można uzyskać, kontaktując się z działem wsparcia dotyczącego oprogramowania IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**609: Typ danych *typ\_danych* nie jest poprawny w bieżącym kontekście.**

**Objaśnienie**

Podczas przetwarzania procedury zapisanej w bazie napotkano nieoczekiwany typ danych.

**Działanie użytkownika**

Dodatkową pomoc dotyczącą identyfikatora i tekstu statusu dla operacji wykonywanej na metadanych można uzyskać, kontaktując się z działem wsparcia dotyczącego oprogramowania IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury zapisanej w bazie, znajdujące się na serwerze bazy danych.

## **Błędy dotyczące biblioteki MDOBJECT.LIB**

**1000: Nie powiodło się klonowanie obiektu *nazwa\_obiektu*.**

**Objaśnienie**

Wystąpił błąd podczas klonowania wewnętrznego obiektu klasy w procedurze zapisanej w bazie.

**Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1001: Rozwinięcie hierarchii *nazwa\_hierarchii* nie może być zdefiniowane jako rekurencyjne, ponieważ hierarchia ma więcej niż dwa poziomy.**

**Objaśnienie**

Hierarchia ma zbyt wiele poziomów na potrzeby rozwinięcia rekurencyjnego.

### **Działanie użytkownika**

Przed zmianą typu rozwinięcia na rekurencyjne zmodyfikuj hierarchię, tak aby miała nie więcej niż dwa poziomy.

#### **1002: Wywołana funkcja *nazwa\_funkcji* nie jest obsługiwana.:**

##### **Objaśnienie**

Dla jednej z klas procedury zapisanej w bazie nie zaimplementowano wewnętrznej metody wirtualnej.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z Obsługą klienta firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1003: Kontener jest pusty wbrew oczekiwaniom.:**

##### **Objaśnienie**

Wewnętrzna struktura kontenera w procedurze jest wbrew oczekiwaniom pusta.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1004: W kontenerze nie można odnaleźć obiektu *nazwa\_obiektu*.:**

##### **Objaśnienie**

Nie został odnaleziony obiekt poszukiwany w jednej z wewnętrznych struktur kontenera w procedurze.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1005: W kontenerze istnieje już duplikat elementu *nazwa\_elementu*.:**

##### **Objaśnienie**

W jednej z wewnętrznych struktur kontenera procedury zapisanej w bazie istnieje już obiekt, który tam jeszcze istnieć nie powinien.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1006: Podczas operacji na liście wystąpił wyjątek.:**

### **Objaśnienie**

Podczas wykonywania operacji na jednej z wewnętrznych list struktur procedury zapisanej w bazie wystąpił nieoczekiwany wyjątek.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1007: W kontenerze wystąpił błąd wewnętrzny. Kod błędu: *błąd i numer*; komunikat: *komunikat*.**

### **Objaśnienie**

Podczas wykonywania operacji na jednej z wewnętrznych struktur kontenera procedury zapisanej w bazie wystąpił błąd.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1008: W ramach operacji kopiowania nie skopiowano do końca wszystkich właściwości. Operacja kopiowania nie powiodła się dla właściwości *nazwa\_właściwości* o wartości *wartość*.**

### **Objaśnienie**

Podczas wykonywania operacji kopiowania na jednym z wewnętrznych obiektów procedury zapisanej w bazie wystąpił błąd. Skopiowanie jednej z właściwości obiektu wewnętrznego było niemożliwe.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1009: Typ obiektu *typ1* nie jest poprawny. Oczekiwano typu *typ2*.**

### **Objaśnienie**

Podczas przetwarzania procedury zapisanej w bazie napotkano obiekt nieoczekiwanej typu.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1010: Parametr *nazwa\_parametru* nie ma kompletnego identyfikatora.**

### **Objaśnienie**

Jeden z parametrów przekazanych jako parametr wejściowy metody wewnętrznej dla tej procedury jest niekompletnym identyfikatorem obiektu metadanych.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1011: Obiekt *nazwa\_objektu* nie ma kompletnego identyfikatora.:**

##### **Objaśnienie**

Identyfikator obiektu metadanych w procedurze zapisanej w bazie jest wbrew oczekiwaniom niekompletny.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1012: Parametr *nazwa\_parametru* jest taki sam jak obiekt.:**

##### **Objaśnienie**

Jeden z parametrów przekazanych jako argument wejściowy metody obiektu wewnętrznego procedury nieoczekiwanie okazał się tożsamy z obiektem, do którego należy metoda.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1013: Wystąpił nieoczekiwany wskaźnik o wartości NULL.:**

##### **Objaśnienie**

Podczas przetwarzania procedury zapisanej w bazie wystąpił nieoczekiwany wskaźnik NULL.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1014: Kursor osiągnął koniec kontenera.:**

##### **Objaśnienie**

Kursor w jednej z wewnętrznych struktur kontenera procedury nieoczekiwanie osiągnął koniec kontenera.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **1015: Obiekt *nazwa\_objektu* nie jest poprawny. Przyczyna: Identyfikator=*identyfikator*; Komunikat=*komunikat*.:**

## Objaśnienie

Wewnętrzny obiekt metadanych w procedurze jest niepoprawny.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**1016: Z obiektem *nazwa\_obiektu* związana jest lista typu *typ\_listy*, która niespodziewanie okazała się pusta.:**

### Objaśnienie

Struktura kontenera będącego wewnętrznym elementem procedury zapisanej w bazie jest wbrew oczekiwaniom pusta.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

## Błędy dotyczące biblioteki MDOPERATION.LIB

**2001: Wygenerowane zapytanie *zapytanie* nie zawiera wymaganej kolumny *objectType*.:**

### Objaśnienie

W zapytaniu SQL wygenerowanym przez procedurę zapisaną w bazie brakuje wymaganej kolumny.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

## Błędy dotyczące biblioteki MDPARSER.LIB

**3001: W analizatorze składni wystąpił wyjątek XML podczas operacji *operacja*; komunikat: *komunikat*.:**

### Objaśnienie

W procedurze zapisanej w bazie wystąpił nieoczekiwany wyjątek podczas analizy składniowej kodu XML przekazanego do procedury.

### Działanie użytkownika

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3002: Wystąpił nieoczekiwany wyjątek analizatora składni w elemencie *operacja*.:**

### **Objaśnienie**

W procedurze zapisanej w bazie wystąpił nieoczekiwany wyjątek podczas analizy składniowej kodu XML przekazanego do procedury.

### **Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3003: W analizatorze składni wystąpił wyjątek SAXParseException podczas operacji operacja; komunikat: komunikat.:**

### **Objaśnienie**

W procedurze zapisanej w bazie wystąpił nieoczekiwany wyjątek podczas analizy składniowej kodu XML przekazanego do procedury.

### **Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3004: System nie mógł uzyskać komunikatu o błędzie analizatora składni dla elementu operacja.:**

### **Objaśnienie**

W procedurze zapisanej w bazie wystąpił nieoczekiwany błąd podczas analizy składniowej kodu XML przekazanego do procedury. Nie udało się pobrać komunikatu o błędzie z analizatora składni XML.

### **Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3100: Nie powiodła się analiza składni XML dla elementu *typ parametru* (wiersz: wiersz, znak:znak, komunikat: komunikat).:**

### **Objaśnienie**

Procedura zapisana w bazie nie mogła przeanalizować składni wejściowego kodu XML. Wejściowy kod XML może być niewłaściwie zbudowany lub niedopasowany do schematu XML, który opublikowano dla tego produktu.

### **Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu.



**3101: Napotkano nieznaną wartość metadanych. komunikat\_analizatora\_skladni.:****Objaśnienie**

W kodzie XML przekazanym do procedury występuje obiekt metadanych nieznanego typu. Procedura nie może przetworzyć tych danych XML.

**Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu.

**3102: Napotkano nieznaną wartość atrybutu XML. nazwa\_atrybutu, wartosc\_atrybutu.:****Objaśnienie**

W kodzie XML przekazanym do procedury występuje atrybut XML nieznanego typu. Procedura nie może przetworzyć tych danych XML.

**Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu.

**3103: Procedura obsługi napotkała niepoprawną wartość wyliczenia dla atrybutu o nazwie nazwa i wartości wartosc.:****Objaśnienie**

W kodzie XML przekazanym procedurze zapisanej w bazie występuje niepoprawna wartość wyliczenia. Procedura nie może przetworzyć tych danych XML.

**Działanie użytkownika**

Zadbaj o prawidłową budowę kodu XML przekazywanego do procedury oraz o jego dostosowanie do schematu XML opublikowanego dla tego produktu.

**3500: Wymagane są dane dla atrybutu lub elementu o nazwie nazwa.:****Objaśnienie**

Procedura zapisana w bazie nie mogła ustawić wartości wskazanego atrybutu lub elementu XML w kodzie XML, który ma zostać zwrócony przez procedurę.

**Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3501: Wymagane są dane dla atrybutów nazwa\_atrybutu1 i nazwa\_atrybutu2.:****Objaśnienie**

Procedura zapisana w bazie nie mogła ustawić wartości wskazanego atrybutu lub elementu XML w kodzie XML, który ma zostać zwrócony przez procedurę.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**3502: Formater napotkał niepoprawną wartość wyliczenia dla atrybutu o nazwie *nazwa i wartości* *wartość*.**

#### Objaśnienie

Podczas formatowania kodu XML, który ma być zwrócony przez procedurę zapisaną w bazie, napotkano w procedurze nieprawidłową wartość wyliczenia.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

## Błędy dotyczące biblioteki MDDATABASE.LIB

**4000: Połączenie z bazą danych nie powiodło się. Nazwa bazy danych:**

*nazwa\_bazy\_danych*, **nazwa użytkownika:** *nazwa\_użytkownika*.

#### Objaśnienie

Procedura zapisana w bazie nie była w stanie nawiązać własnego połączenia z bazą danych.

### Działanie użytkownika

Upewnij się, że identyfikator użytkownika skojarzony z procedurą ma uprawnienia niezbędne do połączenia się z bazą danych.

**4001: Nie zainicjowano połączenia z bazą danych, ponieważ połączenie już istnieje.**

#### Objaśnienie

Procedura zapisana w bazie nieoczekiwanie napotkała duplikat wewnętrznego połączenia z bazą danych. Nie można przeprowadzić żądanej operacji na metadanych.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4002: Operacja na bazie danych zakończyła się niepowodzeniem.**

#### Objaśnienie

Wystąpił błąd podczas wykonywania instrukcji SQL wydanej w bazie danych przez procedurę zapisaną w bazie.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4003: Wykonanie wywołania interfejsu CLI *nazwa\_wywołania* nie powiodło się.**

### **Objaśnienie**

Podczas wykonywania wskazanego wywołania interfejsu CLI wystąpił błąd.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4004: Zwrócone dane są obcięte.:**

#### **Objaśnienie**

Informacje diagnostyczne zwrócone podczas nieudanej operacji w bazie danych zostały obcięte.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4005: Z bazy danych otrzymano ostrzeżenie. SQLSTATE=*kod*, Komunikat=*komunikat*.:**

#### **Objaśnienie**

W wyniku wywołania interfejsu CLI w ramach procedury zapisanej w bazie zwrócone zostało ostrzeżenie.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera.

### **4008: Napotkano nieznaną typ danych DB2.:**

#### **Objaśnienie**

W trakcie realizacji żądania w bazie danych procedura natrafiła na nieznaną typ danych.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4009: Nie wygenerowano poprawnej nazwy punktu zapisu.:**

#### **Objaśnienie**

Procedura zapisana w bazie nie mogła wygenerować poprawnej nazwy punktu zapisu dla transakcji w bazie danych. Do utworzenia nazwy punktu zapisu procedura używa własnego identyfikatora aplikacji w bazie danych.

### **Działanie użytkownika**

Wywołaj procedurę ponownie. Ponowne wywołanie może spowodować wygenerowanie dla procedury nowego identyfikatora aplikacji w bazie danych, na podstawie którego da się wygenerować poprawną nazwę punktu zapisu. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4010: Nie powiodła się próba ustawienia punktu zapisu programu DB2.:**

##### **Objaśnienie**

Procedura zapisana w bazie nie mogła ustawić punktu zapisu transakcji w bazie danych. W bieżącej transakcji może istnieć już punkt zapisu o nazwie identycznej z używaną przez daną instancję procedury.

### **Działanie użytkownika**

Jeśli to możliwe, zwolnij punkty zapisu dla bieżącej transakcji przed ponowniem wywołania procedury. Można także ponownie wywołać procedurę przy użyciu nowej transakcji.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4011: Punkt zapisu nie został ustawiony przed tym punktem wykonania.:**

##### **Objaśnienie**

W pewnym punkcie procedury zapisanej w bazie nieoczekiwanie brakowało punktu zapisu transakcji. Brakujący punkt zapisu mógł nie zostać ustawiony przez procedurę lub mógł zostać zwolniony przez działania w bazie danych wykonywane niezależnie od procedury.

### **Działanie użytkownika**

Wywołaj procedurę ponownie. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4012: Zapisano niepoprawny łańcuch punktu zapisu.:**

##### **Objaśnienie**

Nazwa punktu zapisu transakcji w bazie danych nie została poprawnie zapisana w wewnętrznej strukturze danych procedury, zapewne ze względu na niewystarczającą ilość pamięci na potrzeby procesu procedury.

### **Działanie użytkownika**

Wywołaj procedurę ponownie. Jeśli problem nie ustąpi, zwiększ ilość pamięci przydzielonej na potrzeby procesu procedury. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4013: Nie powiodło się zerowanie punktu zapisu.:**

### **Objaśnienie**

Procedura zapisana w bazie nie mogła skasować punktu zapisu transakcji w bazie danych. Punkt zapisu mógł nie zostać ustawiony przez procedurę lub mógł zostać zwolniony przez działania w bazie danych wykonywane niezależnie od procedury.

### **Działanie użytkownika**

Wywołaj procedurę ponownie. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4014: Nie powiodła się próba określenia ustawienia DB2 AUTOCOMMIT.:**

#### **Objaśnienie**

Próba ustalenia przez procedurę, czy opcja DB2 AUTOCOMMIT jest włączona, czy wyłączona, zakończyła się niepowodzeniem.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4015: Nie powiodła się próba ustawienia parametru DB2 AUTOCOMMIT na OFF.:**

#### **Objaśnienie**

Próba wyłączenia opcji DB2 AUTOCOMMIT podjęta przez procedurę zakończyła się niepowodzeniem.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4016: Nie zwrócono żadnych danych w wyniku wywołania interfejsu CLI SQLFetch().:**

#### **Objaśnienie**

Funkcja CLI SQLFetch() nie zwróciła do procedury żadnych danych. Taka sytuacja może być dopuszczalna, jednak procedura nie powinna była zezwolić na powielanie tego błędu w procedurze bez zmian.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4017: Obiekt nazwa\_obiektu nie został poprawnie skonstruowany.:**

### **Objaśnienie**

Wewnętrzny obiekt w procedurze zapisanej w bazie nie został prawidłowo zainicjowany.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4018: Próba rozłączenia z bazą danych nie powiodła się.:**

#### **Objaśnienie**

Procedura zapisana w bazie nie mogła zakończyć połączenia z bazą danych.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4019: Błąd SQL programu DB2 - SQLCODE: *kod\_sql*, SQLSTATE: *stan\_sql*, SQLMESG: *komunikat\_sql*.:**

#### **Objaśnienie**

Wystąpił błąd podczas wykonywania instrukcji SQL wydanej w bazie danych przez procedurę zapisaną w bazie.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4020: Błąd SQL programu DB2 - brak szczegółowych informacji.:**

#### **Objaśnienie**

Nie są dostępne żadne informacje diagnostyczne dotyczące błędu, który wystąpił podczas wykonywania instrukcji SQL wydanej przez procedurę w bazie danych.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

### **4021: Błąd SQL programu DB2 - brak szczegółowych informacji.:**

#### **Objaśnienie**

Wystąpił błąd podczas próby uzyskania informacji diagnostycznych dotyczących innego błędu, napotkanego w trakcie wykonywania instrukcji SQL wydanej przez procedurę w bazie danych.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4022: Próba przydziału uchwytu DB2 nazwa\_uchwytu nie powiodła się.:**

##### **Objaśnienie**

Wystąpił błąd podczas próby przydzielania uchwytu DB2 w procedurze zapisanej w bazie.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4023: Próba zwolnienia uchwytu DB2 nazwa\_uchwytu nie powiodła się.:**

##### **Objaśnienie**

Wystąpił błąd podczas próby zwolnienia uchwytu DB2 w procedurze zapisanej w bazie.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4028: Transakcja nie została zatrzymana.:**

##### **Objaśnienie**

Wystąpił błąd podczas próby zakończenia transakcji procedury zapisanej w bazie.

### **Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4029: W głównej tabeli obiektów znaleziono powielone wiersze mające jednakową nazwę i schemat.:**

##### **Objaśnienie**

W jednej z tabel katalogu metadanych nieoczekiwanie znaleziono powielone wiersze mające jednakową nazwę i schemat. Sytuacja taka wskazuje na błąd wewnętrzny w procedurze.

### **Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4030: Nie zainicjowano struktury DBINFO. W bazie danych musi zostać utworzona procedura zapisana w bazie z użyciem opcji DBINFO.:**

**Objaśnienie**

Procedura zapisana w bazie nie otrzymała struktury DBINFO od klienta bazy danych.

**Działanie użytkownika**

Wpisz procedurę do katalogu w odpowiedniej bazie danych, używając opcji DBINFO.

**4031: Nie powiodła się próba ustawienia schematu DB2INFO.:**

**Objaśnienie**

Procedura zapisana w bazie nie mogła ustawić schematu DB2INFO jako schematu bieżącego.

**Działanie użytkownika**

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Wywołaj procedurę ponownie.

**4032: Operacja zakończyła się niepowodzeniem ze względu na konflikt między obiektem obecnym w głównej tabeli obiektów a obiektem wstawianym.:**

**Objaśnienie**

Instrukcja SQL INSERT w procedurze zakończyła się niepowodzeniem, ponieważ jej wynikiem byłoby powielenie wpisu obiektu metadanych w jednej z tabel katalogu metadanych.

**Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4033: Operand funkcji kolumnowej zawiera funkcję kolumnową.:**

**Objaśnienie**

W jednej z instrukcji SQL wydanej przez procedurę wykryta została funkcja kolumnowa zagnieżdżona w innej funkcji kolumnowej. Zagnieżdżanie funkcji kolumnowych w instrukcjach SQL nie jest dozwolone.

**Działanie użytkownika**

Zmień szablon wyrażenia SQL dla atrybutu wejściowego lub obiektu miary tak, aby instrukcje SQL generowane przez procedurę nie zawierały zagnieżdżonych funkcji kolumnowych.

**4034: Nie ustalono ustawienia DB2 ISOLATION LEVEL.:**

**Objaśnienie**

Próba ustalenia przez procedurę poziomu odseparowania transakcji w bazie danych zakończyła się niepowodzeniem. Ustalenie poziomu odseparowania nie było możliwe.



### Działanie użytkownika

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4035: Próba ustawienia parametru DB2 ISOLATION LEVEL na READ STABILITY zakończyła się niepowodzeniem.:**

##### **Objaśnienie**

Próba ustawienia przez procedurę poziomu odseparowania transakcji w bazie danych zakończyła się niepowodzeniem. Procedury zapisane w bazie wymagają poziomu odseparowania READ STABILITY (stabilność odczytu).

### Działanie użytkownika

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4036: Nie udało się ustalić wersji programu DB2 Universal Database, która jest obecnie zainstalowana.:**

##### **Objaśnienie**

Próba ustalenia przez procedurę zapisaną w bazie numeru wersji menedżera bazy danych zakończyła się niepowodzeniem.

### Działanie użytkownika

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

#### **4037: Obecnie zainstalowany jest program DB2 Universal Database w wersji *numer\_wersji* z pakietem poprawek *numer*. Wersja ta nie jest zgodna z tą wersją programu DB2 Cube Views. Dokonaj uaktualnienia wersji programu DB2 Universal Database.:**

##### **Objaśnienie**

Obecnie zainstalowana wersja programu DB2 Universal Database nie jest zgodna z bieżącą wersją programu DB2 Cube Views. Dokonaj aktualizacji posiadanej wersji programu DB2 Universal Database tak, aby był co najmniej na poziomie wersji programu DB2 Cube Views.

### Działanie użytkownika

Należy zapewnić, aby na tym samym serwerze zainstalowane były zgodne wersje programów DB2 Universal Database i DB2 Cube Views. Więcej informacji na ten temat zawiera dokumentacja instalacji i konfiguracji.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i tekst wyświetlany przy próbach operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4038: Przetwarzanie instrukcji SQL nie powiodło się, gdyż instrukcja jest zbyt długa lub zbyt złożona.:**

**Objaśnienie**

Procedura zapisana w bazie wywołała instrukcję, której przetworzenie było niemożliwe ze względu na przekroczenie systemowego limitu długości lub złożoności instrukcji albo ze względu na zbyt dużą liczbę zaangażowanych ograniczeń lub wyzwalaczy.

**Działanie użytkownika**

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i tekst wyświetlony podczas podjętej próby operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4039: Wymagana tabela metadanych SYSINFOVERSION nie istnieje w tabelach katalogu DB2. Przeprowadź migrację metadanych.:**

**Objaśnienie**

Wymagana tabela metadanych SYSINFOVERSION nie istnieje w tabelach katalogu DB2. Przeprowadź migrację metadanych.

**Działanie użytkownika**

Zaktualizuj tabele katalogu DB2, migrując metadane. Informacje na temat przeprowadzania migracji można znaleźć w książce *DB2 Cube Views Podręcznik i skorowidz*. Dodatkową pomoc dotyczącą identyfikatora i tekstu statusu dla operacji wykonywanej na metadanych można uzyskać, kontaktując się z działem wsparcia dotyczącego oprogramowania IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4040: Wersja tabeli metadanych jest niezgodna z aktualną wersją interfejsu API programu DB2 Cube Views. Tabele metadanych są w wersji numer\_wersji1, a aktualna wersja API to numer\_wersji2. Należy przeprowadzić migrację tabel metadanych do aktualnej wersji API.:**

**Objaśnienie**

Wartość VERSION z tabeli SYSINFOVERSION nie odpowiada wersji funkcji API programu DB2 Cube Views, za pomocą której próbujesz wykonać operację. Należy przeprowadzić migrację tabel metadanych i zapewnić, aby w tabeli SYSINFOVERSION znajdowała się poprawna wartość VERSION.

**Działanie użytkownika**

Przeprowadź migrację tabel metadanych w katalogu DB2. Informacje na temat przeprowadzania migracji można znaleźć w książce *DB2 Cube Views Podręcznik i skorowidz*. Dodatkową pomoc dotyczącą identyfikatora i tekstu statusu operacji wykonywanej na metadanych można uzyskać, kontaktując się z działem wsparcia dotyczącego oprogramowania IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4041: Wersja tabel metadanych nie jest określona. Należy przeprowadzić migrację tabel metadanych do aktualnej wersji API.:**

**Objaśnienie**

Wartość VERSION w tabeli SYSINFOVERSION nie istnieje. Należy przeprowadzić migrację tabel metadanych z katalogu DB2, uruchamiając skrypt db2mdmigrate.sql.

### Działanie użytkownika

Przeprowadź migrację istniejących tabel metadanych. Informacje na temat przeprowadzania migracji można znaleźć w książce *DB2 Cube Views Podręcznik i skorowidz*. Dodatkową pomoc dotyczącą identyfikatora i tekstu statusu operacji wykonywanej na metadanych można uzyskać, kontaktując się z działem wsparcia dotyczącego oprogramowania IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**4042: Określono wiele różnych lub powielonych wersji tabel metadanych. Sprawdź tabele metadanych i zapewnij, aby określona była tylko jedna, poprawna wersja.:  
Objaśnienie**

W tabeli katalogu SYSINFOVERSION określono wiele różnych lub powtarzających się wersji tabel metadanych. W tabeli SYSINFOVERSION powinna być określona tylko jedna, poprawna wersja tabeli metadanych.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Jeśli to możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

## Błędy dotyczące biblioteki MDUTILITY.LIB

**5000: Program narzędziowy nie mógł przeprowadzić analizy składniowej łańcucha łańcuch.:  
Objaśnienie**

Metoda wewnętrzna procedury napotkała błąd podczas analizy składniowej wewnętrznej wartości łańcuchowej.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5001: Format liczb programu narzędziowego nie został prawidłowo zainicjowany. Zapewnij dostępność wystarczającej ilości pamięci.:  
Objaśnienie**

Formater liczb ICU nie został prawidłowo zainicjowany w procedurze zapisanej w bazie. Może to wynikać z niedostatecznej ilości pamięci dostępnej na potrzeby procedury. Innym możliwym powodem jest ładowanie przez procedurę niewłaściwych wersji bibliotek ICU.

### Działanie użytkownika

Zwiększ ilość pamięci dostępnej dla chronionego procesu procedury i wywołaj procedurę ponownie. Upewnij się, że procedura ładuje właściwą wersję bibliotek ICU. Warto też sprawdzić prawidłowość ustawienia zmiennej środowiskowej określającej ścieżkę wyszukiwania bibliotek czasu wykonywania.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5002: Pakunek zasobów programu narzędziowego nie został prawidłowo zainicjowany. Kod błędu=kod. Upewnij się, że pakunek istnieje i że znajduje się w ścieżce**

*nazwa\_ścieżki.:*

#### **Objaśnienie**

Pakunek zasobów ICU nie został poprawnie zainicjowany w procedurze zapisanej w bazie. Niewłaściwa inicjacja może wynikać z następujących problemów: proces procedury nie dysponuje wystarczającą ilością pamięci; procedura ładuje niewłaściwe wersje bibliotek ICU; procedura ładuje niewłaściwy pakunek zasobów.

#### **Działanie użytkownika**

Zwiększ ilość pamięci dostępnej dla chronionego procesu procedury i wywołaj procedurę ponownie. Upewnij się, że procedura ładuje właściwą wersję bibliotek ICU. Warto też sprawdzić prawidłowość ustawienia zmiennej środowiskowej określającej ścieżkę wyszukiwania bibliotek czasu wykonywania. Należy też się upewnić, że w systemie serwera bazy danych zainstalowana jest właściwa wersja pakunku zasobów procedury.

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5003: Nie znaleziono ścieżki do danych wskazanej zmienną środowiskową nazwa\_zmiennej. Sprawdź poprawność ustawienia zmiennej środowiskowej.:**

#### **Objaśnienie**

Nie jest ustawiona zmienna środowiskowa DB2 używana przez procedurę zapisaną w bazie.

#### **Działanie użytkownika**

Upewnij się, że program DB2 jest prawidłowo zainstalowany w systemie. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5004: Strumień docelowy jest zamknięty.:**

#### **Objaśnienie**

Strumień danych używany wewnętrznie przez procedurę jest wbrew oczekiwaniom zamknięty. Przymuszczalnie w systemie bazy danych nie jest dostępna wystarczająca liczba uchwytów plików.

#### **Działanie użytkownika**

Zadbaj o dostępność wystarczającej liczby uchwytów plików w ustawieniach systemu operacyjnego. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5005: W bazie docelowej znaki są zapisywane z użyciem kodowania domyślnego.:**

#### **Objaśnienie**

Procedura zapisana w bazie zapisuje dane do plików w systemie plików serwera bazy danych z zastosowaniem domyślnego kodowania przewidzianego dla danej procedury zapisanej w bazie.

### Działanie użytkownika

Aplikacje odczytujące pliki zapisywane przez procedurę muszą być zdolne do interpretacji danych zakodowanych zgodnie z domyślnym kodowaniem procedury zapisanej w bazie.

**5006: Nie zapisano łańcucha protokołu wejścia *łańcuch*. Łańcuch jest na poziomie *poziom\_łańcucha*, a protokół jest na poziomie *poziom\_protokołu*.**

#### Objaśnienie

Bieżący poziom protokołowania nie zezwala na zapis wskazanego komunikatu w jednym z plików protokołów używanych przez procedurę zapisaną w bazie.

### Działanie użytkownika

Zmień poziom protokołowania, jeśli zapisanie wskazanego komunikatu w jednym z plików protokołów procedury jest konieczne.

**5007: Nie znaleziono tekstu komunikatu odpowiadającego kodowi błędu *kod*.**

#### Objaśnienie

Tekst odpowiadający wskazanemu kodowi błędu nie został znaleziony w pliku pakunku zasobów procedury. Wersja używanego pliku pakunku zasobów może być niewłaściwa.

### Działanie użytkownika

Upewnij się, że w systemie serwera bazy danych zainstalowana jest właściwa wersja pliku pakunku zasobów procedury zapisanej w bazie. Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**5008: Wystąpił błąd podczas dostępu do *operacja* w celu pobrania globalnego statycznego obiektu *MsgBase*.**

#### Objaśnienie

Podczas próby dostępu do wewnętrznego obiektu komunikatu w procedurze zapisanej w bazie wystąpił błąd.

### Działanie użytkownika

Dodatkową pomoc można uzyskać, kontaktując się z działem wsparcia firmy IBM. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

## Błędy zewnętrznych funkcji API

### 6000-6199

**6000: Funkcja *OLAPMSG()* zakończyła działanie niepowodzeniem, zwracając kod błędu *kod*.**

#### Objaśnienie

Procedura zapisana w bazie zakończyła działanie niepowodzeniem.

### Działanie użytkownika

W zależności od kodu powrotu, skoryguj problem i ponów wywołanie procedury albo skontaktuj się z działem wsparcia firmy IBM w celu uzyskania dalszej pomocy. Kontaktując się z działem wsparcia firmy IBM, należy podać identyfikator statusu oraz zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**6001: Szablony SQL obiektu *nazwa\_obiektu*, których instancje utworzono, są niepoprawne. Wartość: *wartość*. Identyfikator przyczyny=*Identyfikator*, komunikat: *komunikat*.**

#### Objaśnienie

Szablon SQL, którego instancja została utworzona, jest fragmentem instrukcji SQL, który można utworzyć, łącząc szablony wyrażeń SQL dla wszystkich atrybutów i miar wchodzących w skład atrybutu złożonego lub miary złożonej. Stwierdzono występowanie problemu dotyczącego instancji szablonu SQL dla wskazanego obiektu.

### Działanie użytkownika

Zapoznaj się z identyfikatorem przyczyny i komunikatem. Wywołaj ponownie procedurę po wprowadzeniu zmian sugerowanych w komunikacie przyczyny.

**6002: Obiekt *obiekt1* odwołuje się do obiektu *obiekt2*, lecz obiekt *obiekt2* nie istnieje w bazie danych.**

#### Objaśnienie

Obiekty bazy danych mogą się odwoływać wyłącznie do obiektów istniejących w bazie danych.

### Działanie użytkownika

Utwórz obiekt stanowiący cel odwołania w bazie danych i ponownie wywołaj operację na metadanych. Można także usunąć odwołania do tego obiektu i wtedy ponownie wywołać operację.

**6003: Nie można otworzyć protokołu *nazwa\_protokołu* we wskazanym katalogu. Sprawdź, czy podana ścieżka istnieje i czy plik jest dostępny do zapisu.**

#### Objaśnienie

Nie było możliwe otwarcie przynajmniej jednego z plików protokołu używanych przez procedurę.

### Działanie użytkownika

Upewnij się, że ścieżka wskazana w pliku konfiguracyjnym procedury istnieje. Zadbaj o to, by identyfikator użytkownika, z którym skojarzona jest procedura na serwerze bazy danych, miał uprawnienia do tworzenia, odczytu i zapisywania wymaganych plików protokołów.

**6005: Parametr wejściowy typu metadanych dla tej operacji jest wbrew oczekiwaniom pusty. Brakujący parametr typu metadanych jest dla tej operacji wymagany.**

#### Objaśnienie

Wywołana operacja na metadanych wymagała dostarczenia metadanych jako parametru wejściowego. Parametr procedury służący do wymiany metadanych jest wbrew oczekiwaniom pusty.

### Działanie użytkownika

Ponów wywołanie operacji, przekazując tym razem wymagane metadane.

**6006: Nie znaleziono żadnych obiektów spełniających kryteria wyszukiwania:**  
*kryteria\_wyszukiwania.:*

#### Objaśnienie

Operacja na metadanych nie natrafiła na żadne obiekty metadanych pasujące do podanych kryteriów wyszukiwania. Nie wprowadzono żadnych zmian w zawartości katalogu metadanych.

### Działanie użytkownika

Wywołaj operację na metadanych, używając innego trybu importu, jeśli wymagana jest zmiana zawartości katalogu metadanych.

**6007: Napotkano konflikty między obiektami znajdującymi się w katalogu a obiektami importowanymi. Nie wprowadzono żadnych zmian w metadanych.:**

#### Objaśnienie

Wykryte zostały konflikty między obiektami importowanymi a obiektami, które już znajdują się w katalogu metadanych. Z uwagi na wybrany tryb importu nie wprowadzono żadnych zmian w obiektach katalogu metadanych.

### Działanie użytkownika

Wywołaj operację na metadanych, używając innego trybu importu, jeśli wymagana jest zmiana zawartości katalogu metadanych.

**6008: W obrębie *metadane\_wejsciowe* istnieje duplikat obiekt o tożsamości ID.:**

#### Objaśnienie

W metadanych wejściowych dla tej operacji występują powielone obiekty metadanych. W metadanych wejściowych operacji nie mogą występować duplikaty obiektów.

### Działanie użytkownika

Usuń powielone obiekty z metadanych wejściowych i ponownie wywołaj operację.

**6009: W katalogu metadanych już istnieje obiekt o tej samej tożsamości, co obiekt wprowadzany *nazwa\_obiektu.:***

#### Objaśnienie

Wykonanie operacji na metadanych nie było możliwe, ponieważ w katalogu metadanych istnieje już obiekt o tej samej tożsamości.

### Działanie użytkownika

Przed ponownym wywołaniem nieudanej operacji usuń z katalogu metadanych obiekt o tej samej tożsamości, co obiekt tworzony w wyniku operacji. Alternatywnie można zmodyfikować istniejący obiekt, tak aby pasował do właściwości nowo tworzonego obiektu. Ewentualnie można wyeliminować tworzenie nowego obiektu z wykonywanej operacji.

**6010: Odwołanie do obiektu *nazwa\_obiektu* dla obiektu wprowadzanego *nazwa\_obiektu* już istnieje.:**

### Objaśnienie

W katalogu metadanych jest już zdefiniowane odwołanie między wskazanymi obiektami. Powielanie odwołań nie jest dozwolone.

### Działanie użytkownika

Usuń jedno z powielonych odwołań z wywołania operacji na metadanych.

**6011: Schemat obiektu *nazwa\_obiektu* nie może być zmieniony operacją zmiany nazwy.:**

### Objaśnienie

Operacja zmiany nazwy nie może być użyta do zmiany schematu obiektu metadanych.

### Działanie użytkownika

Upewnij się, że schemat obiektu wybranego do zmiany nazwy pozostaje niezmieniony albo użyj operacji ALTER.

**6013: Wersja *wersja1* schematu XML używanego przez klienta nie jest obsługiwana przez interfejs API na serwerze. Interfejs API na serwerze obsługuje wersję *wersja2* schematu XML.:**

### Objaśnienie

Wersja schematu XML używana przez klienta i osadzona w łańcuchach parametrów wejściowych nie jest zgodna z wersją procedury zapisanej w bazie na serwerze.

### Działanie użytkownika

Upewnij się, że aplikacja kliencka i procedura zapisana w bazie korzystają z tej samej wersji schematu XML opublikowanego dla tego produktu.

**6014: Nie można sformułować szablonu SQL dla obiektu *nazwa\_obiektu*. Powód: ID=ID, komunikat: *komunikat*.:**

### Objaśnienie

Procedura formułuje szablony SQL dla atrybutów i miar, łącząc szablony wyrażeń SQL dla wszystkich atrybutów i miar wchodzących w skład atrybutu złożonego lub miary złożonej. Podczas formułowania szablonu SQL wystąpił problem w procedurze zapisanej w bazie.

### Działanie użytkownika

Zapoznaj się z identyfikatorem przyczyny i komunikatem. Wywołaj ponownie procedurę po wprowadzeniu zmian sugerowanych w komunikacie przyczyny.

**6015: Użytkownik bazy danych o tym identyfikatorze nie ma uprawnień do tworzenia schematu w aktywnej bazie danych.:**

### Objaśnienie

Identyfikator użytkownika będący właścicielem procesu procedury na serwerze bazy danych nie ma uprawnień wymaganych do utworzenia schematu bazy danych w aktywnej bazie danych. Schemat bazy danych jest tworzony dla każdego unikalnego schematu obiektu metadanych.



### Działanie użytkownika

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Nadaj identyfikatorowi będącemu właścicielem procesu procedury uprawnienie do tworzenia schematu w aktywnej bazie danych. Wywołaj procedurę ponownie.

**6016: Użytkownik bazy danych o tym identyfikatorze nie ma uprawnień do wykonania wymaganego działania w aktywnej bazie danych. Z serwera bazy danych zwrócony został następujący komunikat o błędzie: *komunikat.*:**

#### Objaśnienie

Identyfikator użytkownika będący właścicielem procesu procedury na serwerze bazy danych nie ma uprawnień wymaganych do wykonania odpowiedniego działania w aktywnej bazie danych.

### Działanie użytkownika

Sprawdź zawartość protokołów menedżera bazy danych po stronie klienta i serwera. Nadaj wymagane uprawnienia identyfikatorowi użytkownika będącemu właścicielem procesu procedury. Wywołaj procedurę ponownie.

**6017: Obiekt *nazwa\_objektu* nie istnieje w katalogu metadanych.:**

#### Objaśnienie

Żądana operacja wymaga, by w katalogu metadanych istniał wskazany obiekt.

### Działanie użytkownika

Utwórz wskazany obiekt w katalogu metadanych przed ponownym wywołaniem operacji na metadanych.

**6018: Wymagana tabela nie istnieje w bazie danych. Z serwera bazy danych zwrócony został następujący komunikat o błędzie: *komunikat.*:**

#### Objaśnienie

Tabela, której istnienia wymaga wywołana operacja, nie istnieje w bazie danych.

### Działanie użytkownika

Jeśli brakująca tabela jest tabelą użytkownika, utwórz tę tabelę i ponownie wywołaj operację. Jeśli chodzi o tabelę katalogu metadanych lub tabelę systemową bazy danych, skontaktuj się z działem wsparcia firmy IBM w celu uzyskania dalszej pomocy. Należy przy tym podać identyfikator statusu i zapis tekstowy nieudanych operacji na metadanych. Jeśli to tylko możliwe, należy udostępnić pliki protokołów procedury z serwera bazy danych.

**6019: Operacja API korzysta z wersji *numer\_wersji* języka XML, podczas gdy metadane korzystają z wersji *numer\_wersji*. Wersje XML interfejsu API i metadanych muszą być zgodne.:**

#### Objaśnienie

Wersja XML operacji interfejsu API musi być zgodna z wersją XML metadanych.

### Działanie użytkownika

Zapewnij, aby operacja interfejsu API i metadane miały ten sam numer wersji kodu XML.

**6020: W operacji API *nazwa\_operacji* nie jest obsługiwana wersja *numer\_wersji* języka metadanych XML.:**

**Objaśnienie**

W przypadku operacji TRANSLATE kod XML operacji interfejsu API musi być w wersji 8.2.0.1.0, a kod XML metadanych w wersji 8.1.2.1.0.

**Działanie użytkownika**

Dodatkowe informacje na temat reguł metadanych, sprawdzania poprawności metadanych i optymalizacji zapytań zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6021: Języka XML w wersji *numer\_wersji* można korzystać tylko na użytek operacji DESCRIBE.:**

**Objaśnienie**

Zastosowanej wersji kodu XML można używać tylko do operacji DESCRIBE.

**Działanie użytkownika**

Dodatkowe informacje na temat reguł metadanych, sprawdzania poprawności metadanych i optymalizacji zapytań zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

## **Błędy przy sprawdzaniu poprawności**

**6200: Obiekt *nazwa\_obiektu* nie jest kompletny. Upewnij się, że wymagane właściwości są ustawione.:**

**Objaśnienie**

Wskazany obiekt wejściowy nie ma ustawionych wszystkich wymaganych właściwości.

**Działanie użytkownika**

Skonfiguruj wymagane właściwości wskazanego obiektu i wywołaj ponownie operację na metadanych.

**6201: Hierarchia *nazwa\_hierarchii* jest niepoprawna, ponieważ jest typu rekurencyjnego, a nie ma dokładnie dwóch atrybutów.:**

**Objaśnienie**

Hierarchia rekurencyjna musi odwoływać się dokładnie do dwóch atrybutów. Określona hierarchia narusza tę zasadę.

**Działanie użytkownika**

Zmodyfikuj hierarchię tak, aby odwoływała się dokładnie do dwóch atrybutów. Więcej informacji na temat reguł metadanych zawiera *Podręcznik instalacji i użytkownika*.

**6202: Obiekt *nazwa\_obiektu* musi mieć przynajmniej jeden szablon SQL.:**

**Objaśnienie**

Zgodnie z definicjami obiektów metadanych przedstawionymi w dokumentacji produktu wskazana miara musi mieć zdefiniowany przynajmniej jeden szablon SQL.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby miała zdefiniowany przynajmniej jeden szablon SQL. Więcej informacji na temat reguł metadanych zawiera *Podręcznik instalacji i użytkownika*.

#### **6206: Atrybut *nazwa\_atrybutu* powinien mieć tylko jeden szablon SQL.:**

##### **Objaśnienie**

Zgodnie z definicją obiektów metadanych przedstawioną w dokumentacji produktu wskazana miara musi mieć zdefiniowany tylko jeden szablon SQL.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby miała zdefiniowany tylko jeden szablon SQL. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6207: Atrybut *nazwa\_atrybutu* wchodzi w skład łączenia, lecz nie ma odwołania do kolumny.:**

##### **Objaśnienie**

Wskazany obiekt atrybutu musi odwoływać się do kolumny bazy danych, aby mógł być celem prawidłowego odwołania z obiektu łączenia metadanych.

### Działanie użytkownika

Zmodyfikuj wskazany obiekt atrybutu tak, aby odwoływał się do kolumny bazy danych, lub zmodyfikuj odpowiedni obiekt łączenia tak, aby odwoływał się do innego obiektu atrybutu, który będzie odwoływał się do kolumny bazy danych. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6208: Atrybut *nazwa\_atrybutu* wchodzi w skład łączenia i musi wskazywać tę samą tabelę, co *nazwa\_tabeli*.:**

##### **Objaśnienie**

Pierwszy ze wskazanych obiektów atrybutów musi odwoływać się do tej samej tabeli bazy danych, co drugi ze wskazanych obiektów.

### Działanie użytkownika

Zmodyfikuj pierwszy ze wskazanych atrybutów tak, aby odwoływał się do tej samej tabeli bazy danych, co drugi ze wskazanych obiektów, lub zmodyfikuj skojarzony obiekt łączenia tak, aby odwoływał się do innego obiektu atrybutu, który z kolei odwołuje się do tej samej tabeli bazy danych, co ten obiekt. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6209: Długość schematu obiektu *nazwa\_obiektu* przekracza dozwolony limit.:**

##### **Objaśnienie**

Schemat wskazanego obiektu ma długość przekraczającą dopuszczalny limit.

### Działanie użytkownika

Skróć schemat wskazanego obiektu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6210: Długość nazwy obiektu *nazwa\_obiektu* przekracza dozwolony limit.:**

**Objaśnienie**

Nazwa wskazanego obiektu ma długość przekraczającą dopuszczalny limit.

**Działanie użytkownika**

Skróć nazwę wskazanego obiektu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6211: Długość nazwy tabeli obiektu *nazwa\_obiektu* przekracza dozwolony limit.:**

**Objaśnienie**

Nazwa tabeli wskazanego obiektu ma długość przekraczającą dopuszczalny limit.

**Działanie użytkownika**

Skróć nazwę tabeli. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6212: Długość nazwy biznesowej obiektu *nazwa\_obiektu* przekracza dozwolony limit.:**

**Objaśnienie**

Nazwa biznesowa wskazanego obiektu ma długość przekraczającą dopuszczalny limit.

**Działanie użytkownika**

Skróć nazwę biznesową. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6213: Długość komentarza obiektu *nazwa\_obiektu* przekracza dozwolony limit.:**

**Objaśnienie**

Komentarz wskazanego obiektu ma długość przekraczającą dopuszczalny limit.

**Działanie użytkownika**

Skróć komentarz dotyczący wskazanego obiektu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6214: Nazwa schematu obiektu *nazwa\_obiektu* nie może zaczynać się od SYS.:**

**Objaśnienie**

Schemat obiektu metadanych nie może mieć nazwy zaczynającej się od łańcucha SYS.

**Działanie użytkownika**

Dla obiektów metadanych używaj schematu, którego nazwa nie zaczyna się od *SYS*. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6215: Schemat obiektu *nazwa\_obiektu* nie może nosić nazwy SESSION.:**

**Objaśnienie**

Nazwą schematu obiektu metadanych nie może być łańcuch SESSION.

### Działanie użytkownika

Dla obiektów metadanych stosuj schematy o nazwach innych niż SESSION. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6216: Nazwa i schemat obiektu *nazwa\_obiektu* nie są kompletne. Identyfikator przyczyny=*identyfikator*, komunikat=*komunikat*.**

#### Objaśnienie

Stwierdzono, że nazwa wskazanego obiektu lub jego schemat jest nieprawidłowa/-y.

### Działanie użytkownika

Podaj poprawne łańcuchy określające nazwę i schemat wskazanego obiektu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6217: Hierarchia kostki *nazwa\_hierarchii* nie jest poprawna, ponieważ odwołuje się do poziomów, do których nie odwołuje się hierarchia *nazwa\_hierarchii*.**

#### Objaśnienie

Hierarchia kostki odwołuje się do poziomów, do których nie odwołuje się jej hierarchia nadrzędna.

### Działanie użytkownika

Zmodyfikuj hierarchię kostki w taki sposób, aby odwoływała się tylko do tych poziomów, do których odwołuje się także hierarchia nadrzędna. Ewentualnie można zmodyfikować hierarchię nadrzędną, tak aby odwoływała się do tych samych poziomów, co hierarchia kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

## Kody ostrzeżeń

**6250: Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ atrybut klucza poziomu *nazwa\_poziomu* nie jest odwzorowany do jednej kolumny tabeli.**

#### Objaśnienie

Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu, ponieważ atrybut klucza poziomu nie odpowiada pojedynczej kolumnie tabeli.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6251: Funkcja API nie utworzy zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ atrybuty klucza poziomu odpowiadają istniejącemu ograniczeniu przez unikalność, które obowiązuje dla tabeli wymiarów.**

#### Objaśnienie

Funkcja API nie utworzy zależności funkcjonalnej dla obiektu poziomu, ponieważ atrybuty klucza poziomu odpowiadają istniejącemu ograniczeniu przez unikalność zdefiniowanemu dla tabeli wymiarów. Ta zależność funkcjonalna jest nadmiarowa.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6252: Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ dla atrybutu klucza poziomu *nazwa\_poziomu* są dopuszczalne wartości puste.:**

#### Objaśnienie

Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu, jeśli którykolwiek z atrybutów klucza poziomu dopuszcza wartości puste.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6253: Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ atrybuty obejmują więcej niż jedną tabelę.:**

#### Objaśnienie

Funkcja API nie może utworzyć zależności funkcjonalnej, jeśli atrybuty klucza poziomu, atrybuty domyślne lub atrybuty pokrewne obejmują więcej niż jedną tabelę.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6254: Z zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu* wykluczony został atrybut domyślny lub pokrewny *nazwa\_atrybutu*, ponieważ atrybut ten jest już uwzględniony jako atrybut klucza poziomu.:**

#### Objaśnienie

Z zależności funkcjonalnej dla obiektów poziomów wykluczane są atrybuty pokrewne, które są już uwzględnione jako atrybuty klucza poziomu.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6255: Zależność funkcjonalna dla obiektu poziomu *nazwa\_poziomu* wyklucza atrybut domyślny lub pokrewny *nazwa\_atrybutu*, ponieważ ten atrybut domyślny lub pokrewny nie jest odwzorowany do jednej kolumny tabeli.:**

#### Objaśnienie

Zależność funkcjonalna dla obiektu poziomu wyklucza atrybut domyślny lub pokrewny, ponieważ atrybut ten nie jest odwzorowany do jednej kolumny tabeli.

### Działanie użytkownika

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6256: Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ wszystkie atrybuty domyślne i pokrewne zostały wykluczone.:**

**Objaśnienie**

Aby można było utworzyć zależność funkcjonalną odpowiadającą obiektowi poziomu, wymagany jest co najmniej jeden atrybut domyślny lub pokrewny.

**Działanie użytkownika**

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

**6257: Funkcja API nie może utworzyć zależności funkcjonalnej dla obiektu poziomu *nazwa\_poziomu*, ponieważ podczas próby utworzenia zależności funkcjonalnej wystąpił błąd.:**

**Objaśnienie**

Wystąpił błąd, gdy funkcja API próbowała wykonać instrukcję SQL tworzącą zależność funkcjonalną.

**Działanie użytkownika**

Więcej informacji na temat zależności funkcjonalnych zawiera książka *DB2 Cube Views Podręcznik i skorowidz*. Dodatkowe informacje można też znaleźć w odpowiednich wpisach w protokole serwera.

**6258: Funkcja API nie może zmodyfikować lub usunąć obiektu poziomu *nazwa\_poziomu*, ponieważ wystąpił błąd przy próbie usunięcia skojarzonej z nim zależności funkcjonalnej *nazwa\_zależności*.:**

**Objaśnienie**

Program DB2 Cube Views nie może zmodyfikować lub usunąć obiektu poziomu, ponieważ błąd uniemożliwił wykonanie za pośrednictwem funkcji API instrukcji SQL usuwającej zależność funkcjonalną. Zadbaj o posiadanie wszystkich uprawnień wymaganych do tworzenia i usuwania zależności funkcjonalnej.

**Działanie użytkownika**

Więcej informacji na temat zależności funkcjonalnych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*. Aby uzyskać dodatkowe informacje, należy zapoznać się z wpisami w protokole serwera.

**6299: Podczas sprawdzania poprawności znaleziono przynajmniej jeden widok bazy danych. Testy poprawności dotyczące obowiązujących ograniczeń nie zostały przeprowadzone dla łączy, w których uczestniczą kolumny widoków. Wykonano wszystkie inne testy poprawności.:**

**Objaśnienie**

Testy poprawności dotyczące obowiązujących ograniczeń nie zostały przeprowadzone dla łączy, w których uczestniczą kolumny widoków. Testy poprawności dotyczące obowiązujących ograniczeń zostały wykonane dla wszystkich innych żądanych łączy, a ponadto wykonano testy poprawności dotyczące obowiązujących ograniczeń dla wszystkich innych żądanych obiektów.

### Działanie użytkownika

Dodatkowe informacje na temat reguł metadanych, sprawdzania poprawności metadanych i optymalizacji zapytań zawiera książka *DB2 Cube Views Podręcznik i skorowidz*.

### Błędy przy sprawdzaniu poprawności na podstawie reguł

**6300: Model kostki *nazwa\_modelu* nie odwołuje się do przynajmniej jednego obiektu faktów.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Model kostki musi odwoływać się do przynajmniej jednego obiektu faktów.

### Działanie użytkownika

Zmodyfikuj wskazany model kostki tak, aby odwoływał się do przynajmniej jednego obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6301: Model kostki *nazwa\_modelu* nie odwołuje się do zera lub większej liczby wymiarów.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Model kostki musi odwoływać się do zera lub większej liczby wymiarów.

### Działanie użytkownika

Zmień wskazany model kostki tak, aby odwoływał się do zera lub większej liczby wymiarów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6302: W modelu kostki *nazwa\_modelu* brakuje wymiaru i/lub łączenia w jednej z istniejących w nim par wymiar-łączenie.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. W parze wymiar-łączenie w modelu kostki musi znajdować się odwołanie zarówno do wymiaru, jak i do łączenia.

### Działanie użytkownika

Zmodyfikuj wskazany model kostki tak, aby we wszystkich jego parach wymiar-łączenie znajdowały się odwołania zarówno do wymiarów, jak i do łączy. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6303: Łączenie *nazwa\_łączenia*, do którego odwołuje się model kostki *nazwa\_modelu*, nie jest poprawne. Wszystkie atrybuty po jednej jego stronie muszą być celem odwołania z obiektu faktów *nazwa\_faktów*, a wszystkie atrybuty po drugiej stronie muszą być celem odwołania z jednego z wymiarów modelu kostki.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Łączenia w modelu kostki muszą z jednej strony odwoływać się do atrybutów obiektu faktów modelu kostki, a z drugiej strony do atrybutów jednego z wymiarów modelu kostki.



### Działanie użytkownika

Zmodyfikuj niepoprawne łączenie we wskazanym modelu kostki tak, aby wszystkie atrybuty z jednej jego strony pochodziły z obiektu faktów modelu kostki, a wszystkie atrybuty po drugiej stronie łączenia pochodziły z wymiarów modelu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6304: Jedna z agregacji należących do miary *nazwa\_miary* bezpośrednio odwołuje się do wymiaru *nazwa\_wymiaru*, który nie jest celem bezpośredniego odwołania z modelu kostki *nazwa\_modelu*.**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Agregacje w mierze używane przez model kostki muszą odwoływać się tylko do tych wymiarów, które są używane przez ten sam model kostki.

### Działanie użytkownika

Zmień agregację wskazanej miary tak, aby odwoływała się tylko do tych wymiarów, które są używane we wskazanym modelu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6305: Agregacja pustej listy wymiarów w mierze *nazwa\_miary* nie jest dopasowana do żadnego z uprzednio niedopasowanych wymiarów w modelu kostki *nazwa\_modelu*.**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Agregacje pustej listy wymiarów w miarach używanych przez modele kostek muszą być dopasowane do przynajmniej jednego z niedopasowanych w inny sposób wymiarów w każdym modelu kostki.

### Działanie użytkownika

Zmodyfikuj agregację wskazanej miary tak, aby jej pusta lista wymiarów była dopasowana do przynajmniej jednego z poprzednio niedopasowanych wymiarów we wskazanym modelu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6306: Miara *nazwa\_miary* może zawierać wyłącznie agregację pustej listy wymiarów, ponieważ model kostki *nazwa\_modelu* nie zawiera odwołania do żadnego obiektu wymiaru.**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Jeśli model kostki nie odwołuje się do żadnego wymiaru, miara modelu kostki może zawierać wyłącznie agregację pustej listy wymiarów.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby zawierała wyłącznie agregację pustej listy wymiarów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6307: Kostka *nazwa\_kostki* nie odwołuje się do jednego obiektu faktów kostki.**

## Objaśnienie

Wskazany obiekt kostki naruszył regułę obowiązującą dla obiektów metadanych. Kostka musi odwoływać się do co najmniej jednego obiektu faktów kostki.

## Działanie użytkownika

Zmień wskazaną kostkę tak, aby odwoływała się do dokładnie jednego obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6308: Kostka *nazwa\_kostki* nie odwołuje się do przynajmniej jednego obiektu wymiaru kostki.:**

## Objaśnienie

Wskazany obiekt kostki naruszył regułę obowiązującą dla obiektów metadanych. Kostka musi odwoływać się do przynajmniej jednego obiektu wymiaru kostki.

## Działanie użytkownika

Zmień wskazaną kostkę tak, aby odwoływała się do przynajmniej jednego wymiaru kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6309: Obiekt faktów kostki *nazwa\_obiektu\_faktów\_kostki*, do którego odwołuje się kostka *nazwa\_kostki*, nie wywodzi się z obiektu faktów, do którego odwołuje się model kostki *nazwa\_modelu*.:**

## Objaśnienie

Wskazany obiekt kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt faktów kostki używany przez wskazaną kostkę musi się wywodzić z obiektu faktów używanego przez wskazany model kostki.

## Działanie użytkownika

Zmodyfikuj co najmniej jeden ze wskazanych obiektów tak, aby ta reguła nie była naruszana. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6310: Wymiar kostki *nazwa\_wymiaru\_kostki*, do którego odwołuje się kostka *nazwa\_kostki* nie wywodzi się z żadnego z wymiarów, do których odwołuje się model kostki *nazwa\_modelu*.:**

## Objaśnienie

Wskazany obiekt kostki naruszył regułę obowiązującą dla obiektów metadanych. Wymiar kostki używany przez wskazaną kostkę musi się wywodzić z jednego z wymiarów używanych przez wskazany model kostki.

## Działanie użytkownika

Zmodyfikuj co najmniej jeden ze wskazanych obiektów tak, aby ta reguła nie była naruszana. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6311: Obiekt faktów *nazwa\_obiektu\_faktów* nie odwołuje się do żadnej miary.:**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt faktów. Obiekt faktów musi odwoływać się do przynajmniej jednej miary.

## Działanie użytkownika

Zmień wskazany obiekt faktów tak, aby odwoływał się do przynajmniej jednej miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6312: Niektóre z atrybutów i miar, do których odwołuje się obiekt faktów nazwa\_obiektu\_faktów, nie mogą być łączone przy użyciu łączeń obiektów faktów.:**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt faktów. Wszystkie atrybuty i miary obiektu faktów muszą dawać się połączyć przy użyciu łączeń obiektu faktów.

## Działanie użytkownika

Zmodyfikuj wszystkie atrybuty i miary, do których odwołuje się obiekt faktów tak, aby dawały się połączyć przez odwołania do innych obiektów łączeń w obiekcie faktów. Ewentualnie usuń z obiektu faktów lub miary te atrybuty, które nie dają się połączyć przy użyciu istniejących łączeń obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6313: Obiekt faktów nazwa\_obiektu\_faktów zawiera łączenia wielokrotne między dwiema tabelami.:**

## Objaśnienie

Dla wskazanego obiektu faktów została naruszona reguła obowiązująca dla obiektów metadanych. Obiekt faktów nie może zawierać łączeń wielokrotnych między dwiema tabelami.

## Działanie użytkownika

Zmodyfikuj wskazany obiekt faktów tak, aby między każdymi dwiema tabelami istniało tylko jedno łączenie. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6314: Obiekt faktów nazwa\_obiektu\_faktów zawiera pętlę łączenia.:**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt faktów. Łączenia we wskazanym obiekcie faktów tworzą pętlę. Taka sytuacja nie jest dozwolona.

## Działanie użytkownika

Ze wskazanego obiektu faktów usuń jedno z łączeń, które tworzą pętlę, albo zmodyfikuj jedno z łączeń tak, aby wyeliminować pętlę. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6315: Łączenie nazwa\_łączenia nie odwołuje się wyłącznie do atrybutów w obiekcie faktów nazwa\_obiektu\_faktów.:**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt faktów. Łączenia obiektu faktów muszą odwoływać się tylko do atrybutów tego obiektu faktów.

### Działanie użytkownika

Zmodyfikuj wskazane łączenie tak, aby odwoływało się wyłącznie do atrybutów wskazanego obiektu faktów, lub dodaj do obiektu faktów brakujące atrybuty, do których łączenie się odwołuje. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6316: Obiekt faktów kostki *nazwa\_obiektu\_faktów\_kostki* nie odwołuje się do obiektu faktów lub odwołuje się do więcej niż jednego obiektu faktów.:**

### Objaśnienie

Wskazany obiekt faktów kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt faktów kostki musi odwoływać się do jednego obiektu faktów.

### Działanie użytkownika

Zmień wskazany obiekt faktów kostki tak, aby odwoływał się do dokładnie jednego obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6317: Obiekt faktów kostki *nazwa\_obiektu\_faktów\_kostki* nie odwołuje się do żadnych miar.:**

### Objaśnienie

Wskazany obiekt faktów kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt faktów kostki musi odwoływać się do przynajmniej jednej miary.

### Działanie użytkownika

Zmień wskazany obiekt faktów kostki tak, aby odwoływał się do przynajmniej jednej miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6318: Miara *nazwa\_miary*, do której odwołuje się obiekt faktów kostki *nazwa\_obiektu\_faktów\_kostki*, nie jest częścią obiektu faktów *nazwa\_obiektu\_faktów*.:**

### Objaśnienie

Wskazany obiekt faktów kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt faktów kostki musi odwoływać się do miar, do których odwołuje się obiekt faktów, z którego wywodzi się obiekt faktów kostki.

### Działanie użytkownika

Dodaj wskazaną miarę do wskazanego obiektu faktów lub usuń wskazaną miarę ze wskazanego obiektu faktów kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6319: Wymiar *nazwa\_wymiaru* nie odwołuje się do żadnych atrybutów. Wymiar musi odwoływać się do przynajmniej jednego atrybutu.:**

### **Objaśnienie**

Wskazany obiekt wymiaru naruszył regułę obowiązującą dla obiektów metadanych. Obiekt wymiaru musi odwoływać się do przynajmniej jednego atrybutu.

### **Działanie użytkownika**

Zmień wskazany obiekt wymiaru tak, aby odwoływał się do przynajmniej jednego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6320: Niektóre z atrybutów, do których odwołuje się wymiar *nazwa\_wymiaru*, nie mogą być łączone przy użyciu łączy wymiarów.:**

### **Objaśnienie**

Wskazany obiekt wymiaru naruszył regułę obowiązującą dla obiektów metadanych. Atrybuty obiektu wymiaru muszą dawać się połączyć przy użyciu łączy wymiaru.

### **Działanie użytkownika**

Zmodyfikuj wszystkie atrybuty związane ze wskazanym obiektem wymiaru tak, aby dawały się połączyć przez odwołania do innych obiektów łączy w obiekcie wymiaru. Ewentualnie usuń z obiektu wymiaru te atrybuty, które nie dają się połączyć przy użyciu istniejących łączy wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6321: Wymiar *nazwa\_wymiaru* zawiera pętlę łączenia.:**

### **Objaśnienie**

Wskazany obiekt wymiaru naruszył regułę obowiązującą dla obiektów metadanych. Łączenia we wskazanym obiekcie wymiaru tworzą pętlę. Taka sytuacja nie jest dozwolona.

### **Działanie użytkownika**

Ze wskazanego obiektu wymiaru usuń jedno z łączy, które tworzą pętlę, albo zmodyfikuj jedno z łączy tak, aby wyeliminować pętlę. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6322: Wymiar *nazwa\_wymiaru* zawiera łączenia wielokrotne między dwiema tabelami.:**

### **Objaśnienie**

We wskazanym obiekcie wymiaru nastąpiło naruszenie reguły obowiązującej dla obiektów metadanych. Obiekt wymiaru nie może zawierać łączy wielokrotnych między dwiema tabelami.

### **Działanie użytkownika**

Zmodyfikuj wskazany obiekt wymiaru tak, aby między każdymi dwiema tabelami istniało tylko jedno łączenie. Więcej informacji na temat reguł metadanych zawiera dokumentacja produktu.

**6323: Hierarchia *nazwa\_hierarchii* odwołuje się do poziomów, do których nie odwołuje się nadrzędny wymiar hierarchii *nazwa\_wymiaru*.:**

## Objaśnienie

Hierarchie wymiaru mogą się odwoływać wyłącznie do poziomów należących do nadrzędnego obiektu wymiaru. Ta hierarchia odwołuje się do poziomów, do których nie odwołuje się wymiar nadrzędny.

### Działanie użytkownika

Zmodyfikuj hierarchię w taki sposób, by odwoływała się tylko do poziomów obiektu nadrzędnego wymiaru lub uzupełnij obiekt wymiaru o poziomy, do których się odwołuje hierarchia. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6325: Łączenie w wymiarze może się odwoływać tylko do atrybutów tego wymiaru.**

**Łączenie** *nazwa\_łączenia* odwołuje się do atrybutów nie należących do wymiaru *nazwa\_wymiaru*.

### Objaśnienie

Wymiar i łączenie naruszają regułę obowiązującą dla obiektów metadanych, zgodnie z którą łączenie wymiaru może odwoływać się wyłącznie do atrybutów tego wymiaru.

### Działanie użytkownika

Zmodyfikuj łączenie, tak aby odwoływało się tylko do atrybutów wymiaru nadrzędnego lub uzupełnij wymiar o atrybuty, do których odwołuje się łączenie. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6326: Wymiar kostki** *nazwa\_wymiaru\_kostki* **nie odwołuje się do żadnego wymiaru.**

### Objaśnienie

Wskazany obiekt wymiaru kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt wymiaru kostki musi odwoływać się do wymiaru.

### Działanie użytkownika

Zmodyfikuj wskazany obiekt wymiaru kostki tak, aby odwoływał się do wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6327: Wymiar kostki** *nazwa\_wymiaru\_kostki* **nie odwołuje się do żadnej hierarchii kostki.**

### Objaśnienie

Wskazany obiekt wymiaru kostki naruszył regułę obowiązującą dla obiektów metadanych. Obiekt wymiaru kostki musi odwoływać się do hierarchii kostki.

### Działanie użytkownika

Zmodyfikuj wskazany obiekt wymiaru kostki tak, aby odwoływał się do hierarchii kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6328: Hierarchia kostki** *nazwa\_hierarchii\_kostki*, **do której odwołuje się wymiar kostki** *nazwa\_wymiaru\_kostki*, **nie wywodzi się z żadnej z hierarchii, do których odwołuje się wymiar** *nazwa\_wymiaru*.

## Objaśnienie

Wskazany obiekt wymiaru kostki naruszył regułę obowiązującą dla obiektów metadanych. Hierarchia kostki używana przez wskazany wymiar kostki musi wywodzić się z jednej z hierarchii używanych przez wskazany wymiar.

## Działanie użytkownika

Zmodyfikuj co najmniej jeden ze wskazanych obiektów tak, aby ta reguła nie była naruszana. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6329: Hierarchia nazwa\_hierarchii nie odwołuje się do żadnych poziomów.:**

#### Objaśnienie

Obiekt hierarchii musi odwoływać się do przynajmniej jednego poziomu.

## Działanie użytkownika

Zmień wskazany obiekt hierarchii, tak aby odwoływał się do przynajmniej jednego poziomu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6330: Hierarchia nazwa\_hierarchii, w której używane jest rozwinięcie rekurencyjne, nie odwołuje się do dokładnie dwóch poziomów.:**

#### Objaśnienie

Wskazany obiekt hierarchii spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Obiekt hierarchii używający rozwinięcia rekurencyjnego musi odwoływać się do dwóch poziomów.

## Działanie użytkownika

Zmodyfikuj wskazany obiekt hierarchii, tak aby odwoływał się do dokładnie dwóch poziomów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6332: Typ hierarchii nazwa\_hierarchii nie jest zgodny z jej rozwinięciem.:**

#### Objaśnienie

Wskazany obiekt hierarchii spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Zasady zgodności typu hierarchii i rozwinięcia opisano w dokumentacji produktu.

## Działanie użytkownika

Zmodyfikuj wskazaną hierarchię tak, aby jej typ był zgodny z rozwinięciem. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6334: Hierarchia kostki nazwa\_hierarchii musi odwoływać się do dokładnie jednej hierarchii.:**

#### Objaśnienie

Hierarchia kostki narusza regułę obowiązującą dla obiektów metadanych, zgodnie z którą hierarchia kostki musi odwoływać się do dokładnie jednej hierarchii.

### Działanie użytkownika

Zmień wskazany obiekt hierarchii kostki, tak aby odwoływał się do jednej hierarchii. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6335: Hierarchia kostki *nazwa\_hierarchii* musi odwoływać się przynajmniej do jednego poziomu kostki.:**

#### Objaśnienie

Wskazany obiekt hierarchii kostki spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Obiekt hierarchii kostki musi odwoływać się do przynajmniej jednego poziomu kostki.

### Działanie użytkownika

Zmień wskazany obiekt hierarchii, tak aby odwoływał się do przynajmniej jednego poziomu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6336: Hierarchia kostki *nazwa\_hierarchii1* odwołuje się do poziomu kostki, lecz odpowiadająca jej hierarchia *nazwa\_hierarchii2* nie odwołuje się do odpowiedniego poziomu *nazwa\_poziomu*. Należy dodać poziom *nazwa\_poziomu* do hierarchii *nazwa\_hierarchii2* albo usunąć odpowiedni poziom kostki z hierarchii kostki *nazwa\_hierarchii1*.:**

#### Objaśnienie

Hierarchia kostki powoduje naruszenie reguły obowiązującej dla obiektów metadanych, zgodnie z którą hierarchia kostki musi odwoływać się do poziomów kostki wyodrębnionych z poziomów, do których odwołuje się odpowiednia hierarchia.

### Działanie użytkownika

Dodaj poziom do hierarchii lub usuń poziom z hierarchii kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6337: Kolejność poziomów kostki w hierarchii kostki *nazwa\_hierarchii* nie jest zgodna z kolejnością odpowiednich poziomów w hierarchii *nazwa\_hierarchii*.:**

#### Objaśnienie

Wskazany obiekt hierarchii kostki spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Względna kolejność poziomów w hierarchii kostki musi być taka sama, jak względna kolejność tych samych poziomów w hierarchii, z której została wyodrębniona hierarchia kostki.

### Działanie użytkownika

Zmodyfikuj jeden ze wskazanych obiektów, tak aby względna kolejność atrybutów w obu obiektach była jednakowa. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6340: W jednym z szablonów wyrażeń SQL dla miary *nazwa\_miary* używany jest parametr, który nie jest atrybutem, miarą ani kolumną.:**



### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Szablony wyrażeń SQL w obiektach miar muszą mieć parametry, które są atrybutami, miarami lub kolumnami.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby jej szablony wyrażeń SQL korzystały z parametrów będących atrybutami, miarami lub kolumnami. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6341: Między atrybutami lub miarami używanymi jako parametry w szablonie wyrażenia SQL dla miary *nazwa\_miary* istnieje pętla zależności.:**

#### Objaśnienie

Wskazany obiekt miary spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Atrybuty i miary używane jako parametry szablonu wyrażenia SQL miary nie mogą tworzyć pętli zależności.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby jej szablony wyrażenia SQL nie zawierały pętli zależności z udziałem ich parametrów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6342: Miara *nazwa\_miary* zawiera pusty łańcuch w definicji jednego z szablonów wyrażeń SQL.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Szablon wyrażenia SQL dla miary nie może być łańcuchem pustym.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby jej szablon wyrażenia SQL nie był łańcuchem pustym. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6343: Szablon wyrażenia SQL dla miary *nazwa\_miary* zawiera funkcję agregującą.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Szablon wyrażenia SQL dla miary nie może zawierać funkcji agregującej.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby jej szablon wyrażenia SQL nie zawierał funkcji agregującej. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6344: W mierze *nazwa\_miary* nie ma agregacji lub miara ta w nieprawidłowy sposób odwołuje się do obiektów innych niż miary.:**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Agregacja dla miary nie jest wymagana, jeśli miara ta odwołuje się do przynajmniej jednej innej miary i jeśli odwołuje się wyłącznie do miar.

## Działanie użytkownika

Zmień wskazaną miarę, dodając agregację lub wprowadzając do niej odwołanie do przynajmniej jednej innej miary i usuwając odwołania do obiektów innych niż miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6345: Liczba szablonów wyrażeń SQL w mierze *nazwa\_miary* nie jest zgodna z liczbą parametrów pierwszej funkcji agregującej.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Liczba szablonów SQL miary musi być zgodna z liczbą parametrów pierwszej funkcji agregującej tej miary (jeśli agregacja występuje).

## Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby liczba parametrów jej pierwszej funkcji agregującej była zgodna z liczbą szablonów wyrażeń SQL miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6346: Miara *nazwa\_miary*, zawierająca więcej niż jeden szablon wyrażenia SQL, nie definiuje przynajmniej jednego kroku w skrypcie agregacji.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Miara o wielu szablonach wyrażeń SQL musi zawierać przynajmniej jeden zdefiniowany krok w swoim skrypcie agregacji.

## Działanie użytkownika

Zmień wskazaną miarę tak, aby jej skrypt agregacji zawierał przynajmniej jeden krok. Można także usunąć jeden z szablonów SQL miary, pod warunkiem, że pozostały szablon wyrażenia SQL odwołuje się tylko do innych miar. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6347: Miara *nazwa\_miary1* ma zdefiniowany skrypt agregacji. Tymczasem nie powinna ona zawierać definicji żadnych skryptów agregacji, ponieważ miara *nazwa\_miary2*, będąca celem odwołania, definiuje więcej niż jeden szablon wyrażeń SQL.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Jeśli miara A odwołuje się do miary B, która definiuje wiele szablonów SQL, to miara A nie może zawierać skryptu agregacji. Reguła ta obowiązuje na wszystkich poziomach w drzewie odwołań do miar.

### Działanie użytkownika

Usuń z miary skrypt agregacji, który jest przyczyną problemu, lub usuń jeden z szablonów wyrażenia SQL z miary będącej celem odwołania. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6348: Miara *nazwa\_miary* zawiera wieloparametrową funkcję agregującą, która nie jest używana jako pierwsza agregacja.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Wieloparametrowa funkcja agregująca może być użyta wyłącznie jako pierwsza agregacja miary.

### Działanie użytkownika

Skonfiguruj wieloparametrową funkcję agregującą jako pierwszą agregację wskazanej miary lub usuń ją z miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6349: Miara *nazwa\_miary* nie ma dokładnie jednej agregacji z pustą listą wymiarów.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. Jeśli miara definiuje jedną lub więcej agregacji, jedna z agregacji musi mieć pustą listę wymiarów.

### Działanie użytkownika

Zmień wskazaną miarę tak, aby miała jedną pustą listę wymiarów, lub zmień wskazaną miarę tak, aby nie definiowała żadnych agregacji. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6350: Wymiar *nazwa\_wymiaru* jest celem więcej niż jednego odwołania w obrębie miary *nazwa\_miary*.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. W obrębie miary wymiar nie może być celem więcej niż jednego odwołania ani w jednej agregacji, ani w różnych agregacjach.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby odwoływała się do wskazanego wymiaru tylko jeden raz. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6351: Szablon wyrażenia SQL dla obiektu *nazwa\_obiektu* jest pozbawiony wskaźnika leksemu o numerze *numer*. Wskaźniki leksemów muszą być oznaczone numerami kolejnymi począwszy od 1.:**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. W szablonie wyrażenia SQL dla miary wskaźniki leksemów muszą być numerowane kolejno, począwszy od 1.

### Działanie użytkownika

Zmodyfikuj wskazaną miarę tak, aby wskaźniki leksemów w szablonach wyrażeń SQL były numerowane kolejno poczynając od 1. Więcej informacji na temat reguł obowiązujących dla metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6352: Miara *nazwa\_miary* zawiera szablon wyrażenia SQL, w którym nie jest używane przekazane odwołanie, *odwołanie*.**

#### Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt miary. W szablonie wyrażenia SQL dla miary musi być używane przekazane odwołanie do kolumny, atrybutu i miary. Każde odwołanie może być użyte więcej niż jeden raz.

### Działanie użytkownika

Zmodyfikuj szablon wyrażenia SQL dla wskazanej miary tak, aby było w nim używane każde przekazane odwołanie do kolumny, atrybutu i miary. Ewentualnie usuń te odwołania do kolumn, atrybutów i miar, które nie są używane we wskazanym szablonie wyrażenia SQL miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6353: W jednym z szablonów wyrażenia SQL dla atrybutu *nazwa\_atrybutu* używany jest parametr niebędący atrybutem ani kolumną.**

#### Objaśnienie

Wskazany obiekt atrybutu naruszył regułę obowiązującą dla obiektów metadanych. W szablonach wyrażeń SQL w obiektach atrybutów muszą być używane parametry, które są atrybutami lub kolumnami.

### Działanie użytkownika

Zmodyfikuj wskazany atrybut tak, aby w jego szablonach wyrażeń SQL były używane parametry będące atrybutami lub kolumnami. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6354: Między atrybutami używanymi jako parametry w szablonie wyrażenia SQL dla atrybutu *nazwa\_atrybutu* istnieje pętla zależności.**

#### Objaśnienie

Wskazany obiekt atrybutu spowodował naruszenie reguły obowiązującej dla obiektów metadanych. Atrybuty używane jako parametry szablonu wyrażenia SQL atrybutu nie mogą tworzyć pętli zależności.

### Działanie użytkownika

Zmodyfikuj wskazany atrybut tak, aby jego szablony wyrażenia SQL nie zawierały pętli zależności z udziałem ich parametrów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6355: Definicja jednego z szablonów wyrażeń SQL atrybutu *nazwa\_atrybutu* zawiera pusty łańcuch.**

#### Objaśnienie

Wskazany obiekt atrybutu naruszył regułę obowiązującą dla obiektów metadanych. Szablon wyrażenia SQL dla atrybutu nie może być łańcuchem pustym.

### Działanie użytkownika

Zmodyfikuj wskazany atrybut tak, aby jego szablon wyrażenia SQL nie był łańcuchem pustym. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6356: Szablon wyrażenia SQL dla atrybutu *nazwa\_atrybutu* zawiera funkcję agregującą.:**

#### Objaśnienie

Wskazany obiekt atrybutu naruszył regułę obowiązującą dla obiektów metadanych. Szablon wyrażenia SQL dla atrybutu nie może zawierać funkcji agregującej.

### Działanie użytkownika

Zmodyfikuj wskazany atrybut tak, aby jego szablon wyrażenia SQL nie zawierał funkcji agregującej. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6358: Atrybut *nazwa\_atrybutu* zawiera szablon wyrażenia SQL, w którym nie jest używane przekazane odwołanie, odwołanie.:**

#### Objaśnienie

Wskazany obiekt atrybutu naruszył regułę obowiązującą dla obiektów metadanych. W szablonie wyrażenia SQL dla atrybutu muszą być używane wszystkie odwołania do kolumn i atrybutów. Każde odwołanie może być użyte więcej niż jeden raz.

### Działanie użytkownika

Zmodyfikuj szablon wyrażenia SQL dla wskazanego atrybutu tak, aby były w nim używane wszystkie przekazane odwołania do kolumn i atrybutów. Ewentualnie usuń te odwołania do kolumn i atrybutów, które nie są używane we wskazanym szablonie wyrażenia SQL atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6359: Związek atrybutów *nazwa\_związku* nie odwołuje się do dwóch różnych atrybutów.:**

#### Objaśnienie

Wskazany obiekt związku atrybutów naruszył regułę obowiązującą dla obiektów metadanych. Obiekt związku atrybutów musi odwoływać się do dwóch różnych atrybutów.

### Działanie użytkownika

Zmień wskazany związek atrybutów tak, aby odwoływał się do dwóch różnych atrybutów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6360: Związek atrybutów *nazwa\_związku* jest nieprawidłowo zdefiniowany. Właściwość licznosci jest ustawiona na N:N, podczas gdy właściwość zależności funkcjonalnej jest ustawiona na YES.:**

#### Objaśnienie

Wskazany obiekt związku atrybutów naruszył regułę obowiązującą dla obiektów metadanych. Jeśli właściwość zależności funkcjonalnej związku atrybutów jest ustawiona na YES (TAK), właściwość licznosci tego związku nie może być ustawiona na N:N.

### Działanie użytkownika

Zmodyfikuj wskazany związek atrybutów tak, aby jego licznosc była inna niż N:N, albo jego właściwość zależności funkcjonalnej miała ustawienie NO (NIE). Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6361: Łączenie *nazwa\_łączenia* nie odwołuje się do przynajmniej jednego trypletu. Tryplet złożony jest z atrybutu lewostronnego, atrybutu prawostronnego i operatora.:  
Objaśnienie**

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt łączenia. Łączenie musi odwoływać się do przynajmniej jednego trypletu złożonego z obiektu lewostronnego, obiektu prawostronnego i operatora.

### Działanie użytkownika

Zmień wskazany obiekt łączenia tak, aby odwoływał się do przynajmniej jednego trypletu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6362: Nie wszystkie atrybuty lewostronne w łączeniu *nazwa\_łączenia* dają się rozstrzygnąć z uzyskaniem kolumny lub kolumn pojedynczej tabeli.:  
Objaśnienie**

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt łączenia. Wszystkie atrybuty lewostronne w łączeniu muszą w rozstrzygnięciu dawać kolumnę lub kolumny jednej tabeli bazy danych.

### Działanie użytkownika

Zmodyfikuj wskazany obiekt łączenia tak, aby rozstrzygnięcia wszystkich jego atrybutów lewostronnych dawały kolumnę lub kolumny z jednej tabeli. Ewentualnie zmień atrybuty lewostronne obiektu łączenia na inne, które spełniają tę regułę. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6363: Nie wszystkie atrybuty prawostronne w łączeniu *nazwa\_łączenia* dają się rozstrzygnąć z uzyskaniem kolumny lub kolumn pojedynczej tabeli.:  
Objaśnienie**

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt łączenia. Wszystkie atrybuty prawostronne w łączeniu muszą w rozstrzygnięciu dawać kolumnę lub kolumny jednej tabeli bazy danych.

### Działanie użytkownika

Zmodyfikuj wskazany obiekt łączenia tak, aby rozstrzygnięcia wszystkich jego atrybutów prawostronnych dawały w wyniku kolumnę lub kolumny z jednej tabeli. Ewentualnie zmień atrybuty prawostronne obiektu łączenia na inne, które spełniają cytowaną regułę. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6364: Co najmniej jeden z trypletów dla łączenia *nazwa\_łączenia* nie definiuje poprawnej operacji. Przyczyną może być niezgodność typów danych między atrybutami lewo- i prawostronnymi lub ich niezgodność z operatorem.:  
Objaśnienie**

## Objaśnienie

Nastąpiło naruszenie reguły obiektu metadanych przez wskazany obiekt łączenia. Każdy tryplet obiektu łączenia musi definiować poprawną operację. Typy danych atrybutów lewo- i prawostronnych muszą być zgodne w kontekście użytego operatora.

## Działanie użytkownika

Zmodyfikuj wskazany obiekt łączenia tak, aby każdy z jego trypletów definiował poprawną operację. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6365: Model kostki nazwa\_modelu nie odwołuje się do jednego i tylko jednego obiektu faktów.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Pełny model kostki musi odwoływać się do jednego obiektu faktów.

## Działanie użytkownika

Zmień wskazany obiekt modelu kostki tak, aby odwoływał się do dokładnie jednego obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6366: Model kostki nazwa\_modelu nie odwołuje się do przynajmniej jednego wymiaru.:**

#### Objaśnienie

Wskazany obiekt modelu kostki naruszył regułę obowiązującą dla obiektów metadanych. Pełny model kostki musi odwoływać się przynajmniej do jednego obiektu wymiaru.

## Działanie użytkownika

Zmień wskazany model kostki tak, aby odwoływał się do przynajmniej jednego obiektu wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6367: Liczność łączenia faktów z wymiarem nazwa\_łączenia jest inna niż 1:1 lub N:1.:**

#### Objaśnienie

Model kostki nie skorzysta z zaleceń Doradcy optymalizacyjnego, ponieważ liczność łączenia faktów z wymiarem jest inna niż 1:1 lub N:1. Optymalizacja nie zostanie wykonana.

## Działanie użytkownika

Aby model kostki mógł skorzystać z zaleceń Doradcy optymalizacyjnego, liczność wszystkich łączy skierowanych z obiektu faktów do obiektu wymiaru musi wynosić 1:1 albo N:1. Liczność łączenia po stronie atrybutów obiektu faktów musi wynosić 1 lub N, a liczność po stronie atrybutów wymiaru musi wynosić 1. Więcej informacji na temat reguł optymalizacyjnych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6368: Łączenie faktów z wymiarem nazwa\_łączenia nie łączy tabeli dla obiektu faktów nazwa\_obiektu\_faktów z podstawową tabelą wymiarów nazwa\_wymiaru.:**

## Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. W sieci połączeń jednego wymiaru musi istnieć przynajmniej jedna tabela (tabela podstawowa), w której wszystkie połączenia wychodzące mają licznosc N:1 lub 1:1. W modelu kostki łączenia z obiektu faktów do obiektu wymiarów muszą być tworzone z udziałem tej tabeli podstawowej wymiaru.

## Działanie użytkownika

Zadbaj o to, aby wszystkie połączenia faktów z wymiarem w modelu kostki były skierowane z obiektu faktów do tabeli podstawowej odpowiednich wymiarów. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6369: Wymiar *nazwa\_wymiaru* nie ma tabeli podstawowej wskazanej w sieci połączeń tego wymiaru.:**

#### Objaśnienie

Wskazany obiekt wymiaru spowodował naruszenie reguły optymalizacji. W sieci połączeń jednego wymiaru musi istnieć przynajmniej jedna tabela, w której wszystkie połączenia wychodzące mają licznosc N:1 lub 1:1. Wobec braku takiej tabeli podstawowej wymiaru optymalizacja nie zostanie wykonana.

## Działanie użytkownika

Sprawdź licznosci obiektów łączenia używanych w wymiarze. Aby możliwe było wykonanie optymalizacji, wymiar musi mieć tabelę podstawową odpowiadającą opisowi w regułach optymalizacji. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6370: W połączeniu *nazwa\_łączenia* uczestniczą kolumny, dla których nie zdefiniowano ograniczeń referencyjnych.:**

#### Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy zdefiniować ograniczenia dotyczące kolumn uczestniczących w połączeniu. W przypadku połączenia zwrotnego, w którym po obu stronach równości znajduje się ten sam zestaw kolumn, musi być zdefiniowany klucz podstawowy dopasowany do zestawu kolumn. We wszystkich innych przypadkach, gdy zestaw kolumn z jednej strony różni się od zestawu z drugiej strony połączenia, do kolumn z jednej strony musi pasować klucz podstawowy, a do kolumn z drugiej strony musi pasować klucz obcy odwołujący się do klucza podstawowego. Z uwagi na brak ograniczenia optymalizacja nie zostanie wykonana.

## Działanie użytkownika

Utwórz ograniczenia dotyczące kolumn uczestniczących w połączeniu. Jeśli ze względów wydajnościowych nie jest wskazane tworzenie standardowego ograniczenia, utwórz ograniczenia informacyjne z włączoną optymalizacją zapytań. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6371: Klucz podstawowy nie jest zdefiniowany przy użyciu kolumn uczestniczących w połączeniu zwrotnym *nazwa\_łączenia*.:**

#### Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy zdefiniować ograniczenia dotyczące kolumn uczestniczących w połączeniu. Jeśli połączenie jest połączeniem



zwrotnym, co oznacza, że po obu stronach znaku równości figuruje ten sam zestaw kolumn, wymagane jest zdefiniowanie klucza podstawowego dopasowanego do tego zestawu kolumn. Z uwagi na brak ograniczenia optymalizacja nie zostanie wykonana.

#### **Działanie użytkownika**

Jeśli w tabeli jest zdefiniowany klucz podstawowy, ustaw atrybuty łączenia zwrotnego zgodnie z atrybutami odpowiadającymi kolumnom klucza podstawowego w tabeli. W przeciwnym razie utwórz klucz podstawowy z kolumn uczestniczących w łączeniu zwrotnym. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6372: Klucz podstawowy nie jest zdefiniowany przy użyciu kolumn z jednej strony łączenia nazwa\_łączenia.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy zdefiniować ograniczenia dotyczące kolumn uczestniczących w łączeniu. Gdy zestaw kolumn z jednej strony łączenia różni się od zestawu z drugiej strony łączenia, do kolumn z jednej strony musi pasować klucz podstawowy, a do kolumn z drugiej strony musi pasować klucz obcy odwołujący się do klucza podstawowego. Z uwagi na brak ograniczenia optymalizacja nie zostanie wykonana.

#### **Działanie użytkownika**

Utwórz klucz podstawowy z kolumn jednej strony łączenia. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6373: Klucz obcy nie jest zdefiniowany przy użyciu kolumn z jednej strony łączenia nazwa\_łączenia.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy zdefiniować ograniczenia dotyczące kolumn uczestniczących w łączeniu. Gdy zestaw kolumn z jednej strony łączenia różni się od zestawu z drugiej strony łączenia, do kolumn z jednej strony musi pasować klucz podstawowy, a do kolumn z drugiej strony musi pasować klucz obcy odwołujący się do klucza podstawowego. Z uwagi na brak ograniczenia optymalizacja nie zostanie wykonana.

#### **Działanie użytkownika**

Utwórz ograniczenie klucza obcego między kolumnami klucza podstawowego łączenia a kolumnami po drugiej stronie łączenia. Jeśli ze względów wydajnościowych nie jest wskazane tworzenie standardowego ograniczenia, utwórz ograniczenia informacyjne z włączoną optymalizacją zapytań. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6374: Klucz obcy zdefiniowany przy użyciu kolumn z jednej strony łączenia nazwa\_łączenia nie odwołuje się do klucza podstawowego przy użyciu kolumn z drugiej strony łączenia.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy zdefiniować ograniczenia dotyczące kolumn uczestniczących w łączeniu. Gdy zestaw kolumn z jednej strony łączenia różni się od zestawu z drugiej strony łączenia, do kolumn z jednej strony musi

pasować klucz podstawowy, a do kolumn z drugiej strony łączenia musi pasować klucz obcy odwołujący się do klucza podstawowego. Z uwagi na brak ograniczenia optymalizacja nie zostanie wykonana.

#### **Działanie użytkownika**

Utwórz ograniczenie klucza obcego między kolumnami klucza podstawowego łączenia a kolumnami po drugiej stronie łączenia. Jeśli ze względów wydajnościowych nie jest wskazane tworzenie standardowego ograniczenia, utwórz ograniczenia informacyjne z włączoną optymalizacją zapytań. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6375: Liczność łączenia nazwa\_łączenia nie jest ustawiona na 1:1, N:1, ani 1:N.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Przeprowadzenie optymalizacji nie jest możliwe, jeśli liczność łączenia wynosi M:N.

#### **Działanie użytkownika**

Ustaw licznosc łączenia na 1:1, 1:N lub N:1, zależnie od ograniczeń, na których dane łączenie bazuje. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6376: Liczność łączenia zwrotnego nazwa\_łączenia nie jest ustawiona na 1:1.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Przeprowadzenie optymalizacji nie jest możliwe, jeśli liczność łączenia zwrotnego nie wynosi 1:1.

#### **Działanie użytkownika**

Ustaw licznosc łączenia zwrotnego na 1:1. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6377: Liczność łączenia nazwa\_łączenia nie jest ustawiona na 1 dla strony, po której zdefiniowany jest klucz podstawowy.:**

##### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Licznosc łączenia musi wynosić 1 po stronie, po której zdefiniowany jest klucz podstawowy, oraz N po stronie klucza obcego. Jeśli po stronie klucza obcego jest także zdefiniowany klucz podstawowy, również należy użyć licznosci 1. Gdy te warunki nie są spełnione, wykonanie optymalizacji nie jest możliwe.

#### **Działanie użytkownika**

Licznosc łączenia należy ustawić na 1 po stronie, po której zdefiniowany jest klucz podstawowy. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6378: Liczność łączenia nazwa\_łączenia nie jest ustawiona na N dla strony, po której zdefiniowany jest klucz obcy.:**

## Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Liczność łączenia musi wynosić 1 po stronie, po której zdefiniowany jest klucz podstawowy, oraz N po stronie klucza obcego. Jeśli po stronie klucza obcego jest także zdefiniowany klucz podstawowy, również należy użyć liczności 1. Gdy te warunki nie są spełnione, wykonanie optymalizacji nie jest możliwe.

## Działanie użytkownika

Liczność łączenia należy ustawić na N po stronie, po której zdefiniowany jest klucz obcy. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6379: Liczność łączenia *nazwa\_łączenia* nie jest ustawiona na 1 dla strony, po której zdefiniowany jest zarówno klucz podstawowy, jak i obcy.:**

## Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Liczność łączenia musi wynosić 1 po stronie, po której zdefiniowany jest klucz podstawowy, oraz 1 po stronie, gdzie zdefiniowany jest jednocześnie klucz podstawowy i klucz obcy. Gdy te warunki nie są spełnione, wykonanie optymalizacji nie jest możliwe.

## Działanie użytkownika

Należy ustawić licznosc łączenia na 1:1. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6380: Atrybut *nazwa\_atrybutu*, do którego odwołuje się łączenie *nazwa\_łączenia*, nie daje się rozstrzygnąć z uzyskaniem wyrażenia SQL niedopuszczającego wartości pustych.:**

## Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Wszystkie atrybuty używane w łączeniu muszą być rozstrzygane z uzyskaniem wyrażen SQL niedopuszczających wartości NULL. Przeprowadzenie optymalizacji nie jest możliwe, gdy łączenie odwołuje się do atrybutu rozstrzyganego z uzyskaniem wyrażenia SQL dopuszczającego wartość NULL.

## Działanie użytkownika

Usuń z łączenia odwołanie do atrybutu z dozwoloną wartością NULL. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6381: Łączenie *nazwa\_łączenia* jest typu innego niż ŁĄCZENIE WEWNĘTRZNE.:**

## Objaśnienie

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Należy ustawić typ łączenia ŁĄCZENIE WEWNĘTRZNE. Optymalizacja nie jest możliwa.

## Działanie użytkownika

Zmodyfikuj łączenie tak, aby odwoływało się wyłącznie do atrybutów dających się rozstrzygnąć z uzyskaniem pojedynczej kolumny. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6382: Odwołanie do atrybutu *nazwa\_atrybutu* w łączeniu *nazwa\_łączenia* nie daje się rozstrzygnąć z uzyskaniem pojedynczego wyrażenia kolumnowego, co jest warunkiem**

### **koniecznym uczestnictwa w ograniczeniu.:**

#### **Objaśnienie**

Wskazany obiekt łączenia spowodował naruszenie reguły optymalizacji. Wymagane jest stosowanie ograniczeń DB2 względem atrybutów, do których odwołuje się łączenie. Ograniczenia można stosować wyłącznie względem kolumn, dlatego atrybut, do którego odwołuje się łączenie, musi dawać się rozstrzygnąć z uzyskaniem pojedynczej kolumny w tabeli. Gdy te warunki nie są spełnione, wykonanie optymalizacji nie jest możliwe.

#### **Działanie użytkownika**

Zmodyfikuj łączenie tak, aby odwoływało się wyłącznie do atrybutów dających się rozstrzygnąć z uzyskaniem pojedynczej kolumny. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6385: Model kostki *nazwa\_modelu* musi odwoływać się do co najmniej jednego wymiaru mającego hierarchię.:**

#### **Objaśnienie**

Wskazany model kostki narusza regułę optymalizacji, zgodnie z którą model kostki musi odwoływać się do co najmniej jednego wymiaru posiadającego hierarchię.

#### **Działanie użytkownika**

Zmień wskazany wymiar modelu kostki, tak aby wymiar ten odwoływał się do co najmniej jednej hierarchii. Więcej informacji na temat reguł optymalizacji zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6386: Każdy wycinek optymalizacji musi mieć zdefiniowany dokładnie jeden poziom optymalizacji na każdy wymiar kostki *nazwa\_kostki*.:**

#### **Objaśnienie**

Kostka nie spełnia reguły mówiącej, że wycinek optymalizacji musi mieć dokładnie jeden poziom optymalizacji na każdy wymiar kostki.

#### **Działanie użytkownika**

Zmień wycinek optymalizacji tak, aby odwoływał się on do jednego poziomu optymalizacji na każdy wymiar kostki z obiektu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6387: Poziom optymalizacji musi odwoływać się do jednego wymiaru kostki *nazwa\_kostki*.:**

#### **Objaśnienie**

Kostka nie spełnia reguły obiektów metadanych, zgodnie z którą poziom optymalizacji musi odwoływać się do dokładnie jednego wymiaru kostki.

#### **Działanie użytkownika**

Zmień poziom optymalizacji tak, aby odwoływał się on do jednego wymiaru kostki w obiekcie kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6388: Poziom optymalizacji musi odwoływać się do jednej hierarchii kostki *nazwa\_kostki*.:**

### Objaśnienie

Kostka nie spełnia reguły obiektów metadanych, zgodnie z którą poziom optymalizacji musi odwoływać się do dokładnie jednej hierarchii należącej do kostki.

### Działanie użytkownika

Zmień poziom optymalizacji tak, aby odwoływał się on do jednej hierarchii kostki w obiekcie kostki.

**6389: Poziom optymalizacji należy ustawić na allLevel, anyLevel lub na odwołanie do poziomu kostki nazwa\_kostki.:**

### Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą poziom optymalizacji musi być ustawiony na allLevel, anyLevel lub odwołanie do poziomu kostki.

### Działanie użytkownika

Zmień wskazany poziom optymalizacji tak, aby zawierał wartość allLevel, anyLevel albo odwołanie do poziomu kostki z obiektu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6390: Wymiar kostki i hierarchia kostki, do których odwołuje się poziom optymalizacji, muszą wywodzić się z obiektów kostki nazwa\_kostki. Hierarchia kostki musi należeć do wymiaru kostki.:**

### Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą odnośny wymiar kostki i hierarchia kostki muszą być pochodnymi obiektów z kostki dla poziomu optymalizacji. Hierarchia kostki musi należeć do wymiaru kostki.

### Działanie użytkownika

Zmień poziom optymalizacji tak, aby odnośny wymiar kostki i odnośna hierarchia kostki były pochodnymi obiektów z kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6391: Jeśli na poziomie objętym optymalizacją odwołanie do poziomu kostki ma wartość inną niż anyLevel lub allLevel, to poziom kostki nazwa\_poziomu musi należeć do hierarchii kostki nazwa\_hierarchii w kostce nazwa\_kostki.:**

### Objaśnienie

Kostka nie spełnia reguły dotyczącej obiektów metadanych w poziomie optymalizacji, zgodnie z którą, jeśli odwołanie do poziomu kostki ma postać inną niż anyLevel lub allLevel, poziom kostki musi należeć do hierarchii kostki.

### Działanie użytkownika

Zmień poziom optymalizacji tak, aby odnośny poziom kostki należał do hierarchii kostki. Więcej informacji na temat reguł metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6392: Kostka nazwa\_kostki nie może zawierać jednocześnie wycinków optymalizacji z wyodrębnianiem typu MOLAP i z wyodrębnianiem hybrydowym.:**

## Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą kostka nie może używać jednocześnie wycinka optymalizacji z wyodrębnieniem typu MOLAP i wycinka optymalizacji z wyodrębnieniem hybrydowym.

## Działanie użytkownika

Zmień jeden z wycinków optymalizacji tak, aby nie używać jednocześnie wycinka optymalizacji z wyodrębnieniem typu MOLAP i wycinka optymalizacji z wyodrębnieniem hybrydowym w jednej kostce. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6393: Kostka *nazwa\_kostki* nie może używać więcej niż jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP.:**

#### Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą kostka może używać maksymalnie jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP.

## Działanie użytkownika

Zmień kostkę tak, aby miała jeden wycinek optymalizacji z wyodrębnieniem typu MOLAP albo aby nie miała żadnych wycinków tego typu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6394: Kostka *nazwa\_kostki* nie może używać więcej niż jednego wycinka optymalizacji z wyodrębnieniem hybrydowym.:**

#### Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą kostka może używać maksymalnie jednego wycinka optymalizacji z wyodrębnieniem hybrydowym.

## Działanie użytkownika

Zmień kostkę tak, aby miała jeden wycinek optymalizacji z wyodrębnieniem hybrydowym albo aby nie miała żadnych wycinków tego typu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6395: Wycinek optymalizacji z rozszerzaniem może istnieć tylko pod warunkiem, że kostka *nazwa\_kostki* zawiera wycinek optymalizacji z wyodrębnieniem hybrydowym.:**

#### Objaśnienie

Kostka nie spełnia reguły obiektu metadanych, zgodnie z którą wycinek optymalizacji z rozszerzaniem może istnieć tylko wtedy, gdy w kostce znajduje się wycinek optymalizacji z wyodrębnieniem hybrydowym.

## Działanie użytkownika

Jeśli kostka nie zawiera wycinka optymalizacji z wyodrębnieniem hybrydowym, zmień typ wycinka optymalizacji z rozszerzaniem na inny. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6396: Atrybut *nazwa\_atrybutu*, do którego odwołuje się poziom *nazwa\_poziomu* w wymiarze *nazwa\_wymiaru*, musi znajdować się na liście atrybutów wymiaru.:**

### **Objaśnienie**

Wymiar nie spełnia reguły obiektu metadanych, zgodnie z którą wszystkie atrybuty, do których odwołują się poziomy w wymiarze, muszą znajdować się na liście atrybutów tego wymiaru.

### **Działanie użytkownika**

Zmień poziom tak, aby nie odwoływał się do wskazanego atrybutu, albo dodaj wskazany atrybut do wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6397: Poziom *nazwa\_poziomu* musi odwoływać się do przynajmniej jednego atrybutu klucza poziomu.:**

#### **Objaśnienie**

Poziom nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą poziom musi odwoływać się przynajmniej do jednego atrybutu klucza poziomu.

### **Działanie użytkownika**

Zmodyfikuj poziom w taki sposób, aby odwoływał się do przynajmniej jednego atrybutu klucza poziomu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6398: Poziom *nazwa\_poziomu* nie może zawierać atrybutu *nazwa\_atrybutu* więcej niż jeden raz w ramach zestawu atrybutów klucza poziomu.:**

#### **Objaśnienie**

Poziom nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą zestaw atrybutów klucza poziomu nie może zawierać powielonych odwołań do atrybutu.

### **Działanie użytkownika**

Zmodyfikuj poziom w taki sposób, aby nie zawierał powielonych atrybutów klucza poziomu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

### **6399: Poziom *nazwa\_poziomu* musi odwoływać się dokładnie do jednego atrybutu domyślnego.:**

#### **Objaśnienie**

Poziom nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą poziom musi mieć dokładnie jeden atrybut domyślny.

### **Działanie użytkownika**

Zmodyfikuj poziom w taki sposób, by zawierał dokładnie jeden atrybut domyślny. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

## **6400-6499**

### **6401: Nie można wykorzystać jednego atrybutu jednocześnie jako atrybutu domyślnego i atrybutu pokrewnego w poziomie *nazwa\_poziomu*.:**

## Objaśnienie

Poziom nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą jeden atrybut nie może pełnić jednocześnie roli atrybutu domyślnego i atrybutu pokrewnego.

## Działanie użytkownika

Zmodyfikuj poziom, tak aby atrybut domyślny nie był używany jako atrybut pokrewny. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6402: Nie można wykorzystać jednego atrybutu jednocześnie jako atrybutu domyślnego i atrybutu pokrewnego w poziomie *nazwa\_poziomu*.**

## Objaśnienie

Poziom nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą zestaw atrybutów pokrewnych nie może zawierać powielonych odwołań do atrybutu.

## Działanie użytkownika

Zmodyfikuj poziom w taki sposób, aby nie zawierał powielonych atrybutów pokrewnych. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6403: Poziom *nazwa\_poziomu* nie może zawierać atrybutu *nazwa\_atrybutu* więcej niż jeden raz w ramach zestawu atrybutów pokrewnych.**

## Objaśnienie

Poziom kostki nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą poziom kostki musi odwoływać się do dokładnie jednego poziomu.

## Działanie użytkownika

Zmodyfikuj poziom kostki, tak aby odwoływał się dokładnie do jednego poziomu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6404: Poziom kostki *nazwa\_poziomu* musi odwoływać się do dokładnie jednego poziomu.**

## Objaśnienie

Poziom kostki nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą wszystkie atrybuty pokrewne poziomowi kostki muszą być także atrybutami pokrewnymi odpowiedniego poziomu.

## Działanie użytkownika

Zmodyfikuj poziom kostki, tak aby odwoływał się do atrybutów, do których jednocześnie odwołuje się poziom nadrzędny. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6405: Poziom kostki *nazwa\_poziomu* nie może zawierać atrybutu *nazwa\_atrybutu* więcej niż jeden raz w ramach zestawu atrybutów pokrewnych.**



## Objaśnienie

Poziom kostki nie spełnia reguły obowiązującej dla obiektów metadanych, zgodnie z którą zestaw atrybutów pokrewnych nie może zawierać powielonych odwołań do atrybutu.

## Działanie użytkownika

Zmodyfikuj poziomy kostki w taki sposób, aby nie zawierał powielonych atrybutów pokrewnych. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

## Błędy związane z ograniczeniami spójności referencyjnej

**6500: Nie można wykonać tej operacji, ponieważ szablon SQL dla atrybutu lub miary *nazwa\_atrybutu* nadal zawiera odwołania do innych atrybutów, miar lub kolumn. Odwołania te muszą być skasowane przed wykonaniem tej operacji.:**

### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Szablon wyrażenia SQL dla wskazanego obiektu zawiera odwołania do innych atrybutów, miar lub kolumn, które należy usunąć ze wskazanego obiektu przed wykonaniem tej operacji.

## Działanie użytkownika

Przed usunięciem wskazanego obiektu zmodyfikuj go tak, aby jego szablon wyrażenia SQL nie odwoływał się do atrybutów, miar lub kolumn. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6501: Nie można wykonać operacji, ponieważ do atrybutu lub miary *nazwa\_atrybutu* odwołuje się inny atrybut lub miara.:**

### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu lub miary odwołuje się obecnie inny atrybut lub miara, dlatego usunięcie tego atrybutu lub miary nie jest możliwe.

## Działanie użytkownika

Przed usunięciem wskazanego atrybutu lub miary zmodyfikuj odwołujące się do niej obiekty, usuwając odwołanie do wskazanego atrybutu lub miary. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6502: Nie można wykonać operacji, ponieważ do wymiaru *nazwa\_wymiaru* odwołuje się agregacja zdefiniowana w obrębie miary.:**

### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Wskazany wymiar jest obecnie celem odwołania agregacji miary, dlatego jego usunięcie nie jest teraz możliwe.

### Działanie użytkownika

Przed usunięciem wskazanego wymiaru zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6503: Nie można wykonać tej operacji dla obiektu *nazwa\_obiektu*. Hierarchia kostki musi się odwoływać do atrybutów, które są już celem odwołania hierarchii, od której hierarchia kostki się wywodzi.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłyby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z hierarchii atrybutu, który jest nadal używany przez pokrewną hierarchię kostki.
- Podjęto próbę dodania do hierarchii kostki atrybutu, który nie jest używany przez pokrewną hierarchię.

### Działanie użytkownika

Wykonaj jedno z następujących działań:

- Usuń atrybuty z hierarchii kostki przed usunięciem tych samych atrybutów z pokrewnych hierarchii.
- Dodaj atrybuty do hierarchii przed dodaniem tych samych atrybutów do pokrewnych hierarchii kostki.

Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6504: Nie można wykonać tej operacji dla obiektu *nazwa\_obiektu*. Hierarchia kostki musi się odwoływać do związków atrybutów, które są już celem odwołania hierarchii, od której hierarchia kostki się wywodzi.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłyby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z hierarchii związku atrybutów, który jest nadal używany przez pokrewną hierarchię kostki.
- Podjęto próbę dodania do hierarchii kostki związku atrybutów, który nie jest używany przez pokrewną hierarchię.

### Działanie użytkownika

Wykonaj jedno z następujących działań:

- Usuń związki atrybutów z hierarchii kostki przed usunięciem tych samych związków atrybutów z pokrewnych hierarchii.
- Dodaj związki atrybutów do hierarchii przed dodaniem tych samych związków atrybutów do pokrewnych hierarchii kostki.

Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6505: Nie można wykonać operacji, ponieważ do hierarchii *nazwa\_hierarchii* odwołuje się hierarchia kostki.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanej hierarchii odwołuje się obecnie hierarchia kostki, dlatego nie można jej usunąć.

**Działanie użytkownika**

Przed usunięciem wskazanej hierarchii zmodyfikuj odwołujące się do niej obiekty, usuwając odwołania do wskazanej hierarchii. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6506: Nie można wykonać tej operacji dla obiektu *nazwa\_objektu*. Obiekt faktów kostki musi się odwoływać do miar, które są już celem odwołania obiektu faktów, od którego wywodzi się dany obiekt faktów.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z obiektu faktów miary, która jest nadal używana przez pokrewny obiekt faktów kostki.
- Podjęto próbę dodania do obiektu faktów kostki miary, która nie jest używana przez pokrewny obiekt faktów.

**Działanie użytkownika**

Wykonaj jedno z następujących działań:

- Usuń miary z obiektu faktów kostki przed usunięciem ich z pokrewnego obiektu faktów.
- Dodaj odpowiednie miary do obiektu faktów przed dodaniem ich do pokrewnego obiektu faktów kostki.

Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6507: Nie można wykonać operacji, ponieważ do obiektu faktów *nazwa\_objektu\_faktów* odwołuje się obiekt faktów kostki.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego obiektu faktów odwołuje się obecnie obiekt faktów kostki, dlatego nie można go usunąć.

**Działanie użytkownika**

Przed usunięciem wskazanego obiektu faktów zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego obiektu faktów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6508: Nie można wykonać operacji, ponieważ do hierarchii *nazwa\_hierarchii* odwołuje się wymiar.:**

## Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanej hierarchii odwołuje się obecnie wymiar, dlatego nie można jej usunąć.

## Działanie użytkownika

Przed usunięciem wskazanej hierarchii zmodyfikuj odwołujące się do niej obiekty, usuwając odwołania do wskazanej hierarchii. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6509: Nie można wykonać operacji, ponieważ do łączenia nazwa\_łączenia odwołuje się obiekt faktów.:**

## Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego łączenia odwołuje się obecnie obiekt faktów, dlatego nie można usunąć.

## Działanie użytkownika

Przed usunięciem wskazanego łączenia zmodyfikuj odwołujące się do niego obiekty, usuwając odwołania do wskazanego łączenia. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6510: Nie można wykonać operacji, ponieważ do wymiaru kostki nazwa\_wymiaru\_kostki odwołuje się kostka.:**

## Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego wymiaru kostki odwołuje się obecnie kostka, dlatego nie można go usunąć.

## Działanie użytkownika

Przed usunięciem wskazanego wymiaru kostki zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego wymiaru kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6511: Nie można wykonać tej operacji dla obiektu nazwa\_obiektu. Wymiary kostki muszą się wywodzić od wymiarów, do których odwołuje się model kostki, od którego wywodzi się dana kostka.:**

## Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z modelu kostki wymiaru, który jest nadal używany przez pokrewny wymiar kostki.
- Podjęto próbę dodania do kostki wymiaru, który nie jest używana przez pokrewny model kostki.

## Działanie użytkownika

Wykonaj jedno z następujących działań:

- Usun' wymiary kostki z kostek przed usuni'ciem odpowiednich wymiarów z pokrewnych modeli kostki.
- Dodaj wymiary do modeli kostki przed dodaniem tych wymiarów do odpowiednich kostek.

Więcej informacji na temat regu' dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6512: Nie można wykonać operacji, ponieważ do wymiaru *nazwa\_wymiaru* odwołuje się wymiar kostki.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego wymiaru odwołuje się obecnie wymiar kostki, dlatego nie można go usunąć.

**Działanie użytkownika**

Przed usuni'ciem wskazanego wymiaru zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego wymiaru. Więcej informacji na temat regu' dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6513: Nie można wykonać tej operacji dla obiektu *nazwa\_obiektu*. Hierarchia kostki, do której należy wymiar kostki, musi wywodzić się od hierarchii, do której odwołuje się ten sam wymiar, z którego wywodzi się dany wymiar kostki.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z wymiaru hierarchii, która jest nadal używana przez hierarchię pokrewnego wymiaru kostki.
- Podjęto próbę dodania do wymiaru kostki hierarchii kostki, która nie jest używana przez pokrewny wymiar.

**Działanie użytkownika**

Wykonaj jedno z następujących działań:

- Usun' hierarchie kostki z wymiarów kostki przed usuni'ciem pokrewnych hierarchii z wymiarów.
- Dodaj hierarchie do wymiarów przed dodaniem pokrewnych hierarchii kostki do odpowiednich wymiarów kostki.

Więcej informacji na temat regu' dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6514: Nie można wykonać operacji, ponieważ do hierarchii kostki *nazwa\_hierarchii\_kostki* odwołuje się wymiar kostki.:**

**Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanej hierarchii kostki obecnie odwołuje się wymiar kostki, dlatego jej usunięcie nie jest możliwe.

### Działanie użytkownika

Przed usunięciem wskazanej hierarchii kostki zmodyfikuj odwołujące się do niej obiekty, usuwając odwołanie do wskazanej hierarchii kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6515: Nie można wykonać tej operacji dla obiektu *nazwa\_obiektu*. Hierarchia kostki, do której należy wymiar kostki, musi wywodzić się od hierarchii, do której odwołuje się ten sam wymiar, z którego wywodzi się dany wymiar kostki.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Problem może mieć jedną z następujących przyczyn:

- Podjęto próbę usunięcia z wymiaru hierarchii, która jest nadal używana przez hierarchię pokrewnego wymiaru kostki.
- Podjęto próbę dodania do wymiaru kostki hierarchii kostki, która nie jest używana przez pokrewny wymiar.

### Działanie użytkownika

Wykonaj jedno z następujących działań:

- Usuń hierarchie kostki z wymiarów kostki przed usunięciem pokrewnych hierarchii z wymiarów.
- Dodaj hierarchie do wymiarów przed dodaniem pokrewnych hierarchii kostki do odpowiednich wymiarów kostki.

Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6516: Nie można wykonać operacji, ponieważ do łączenia *nazwa\_łączenia* odwołuje się wymiar.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego łączenia odwołuje się obecnie wymiar, dlatego łączenia nie można usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego łączenia zmodyfikuj odwołujące się do niego obiekty, usuwając odwołania do wskazanego łączenia. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6517: Nie można wykonać operacji, ponieważ do atrybutu *nazwa\_atrybutu* odwołuje się wymiar.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie wymiar, dlatego atrybutu nie można usunąć.

### **Działanie użytkownika**

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6518: Nie można wykonać operacji, ponieważ do atrybutu nazwa\_atrybutu odwołuje się hierarchia.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie hierarchia, dlatego nie można go usunąć.

### **Działanie użytkownika**

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6519: Nie można wykonać operacji, ponieważ do związku atrybutów nazwa\_związku odwołuje się hierarchia.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego związku atrybutów odwołuje się obecnie hierarchia, dlatego nie można go usunąć.

### **Działanie użytkownika**

Przed usunięciem wskazanego związku atrybutów zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego związku atrybutów. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6520: Nie można wykonać operacji, ponieważ do wymiaru nazwa\_wymiaru odwołuje się model kostki.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego wymiaru odwołuje się obecnie model kostki, dlatego związku wymiaru nie można usunąć.

### **Działanie użytkownika**

Przed usunięciem wskazanego wymiaru zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego wymiaru. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6521: Nie można wykonać operacji, ponieważ do łączenia nazwa\_łączenia odwołuje się model kostki.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego łączenia odwołuje się obecnie model kostki, dlatego łączenia nie można usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego łączenia zmodyfikuj odwołujące się do niego obiekty, usuwając odwołania do wskazanego łączenia. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6522: Nie można wykonać operacji, ponieważ do elementu *nazwa\_obiektu* odwołuje się obiekt faktów.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego obiektu odwołuje się obecnie obiekt faktów, dlatego nie można usunąć wskazanego obszaru.

### Działanie użytkownika

Przed usunięciem wskazanego obiektu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego obiektu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6523: Nie można wykonać operacji, ponieważ do atrybutu lewostronnego *nazwa\_atrybutu* odwołuje się związek atrybutów.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie związek atrybutów, dlatego atrybutu nie można usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6524: Nie można wykonać operacji, ponieważ do atrybutu prawostronnego *nazwa\_atrybutu* odwołuje się związek atrybutów.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie związek atrybutów, dlatego atrybutu nie można usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6525: Nie można wykonać operacji, ponieważ do atrybutu prawostronnego *nazwa\_atrybutu* odwołuje się łączenie.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie łączenie, dlatego atrybutu nie można usunąć.



### Działanie użytkownika

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6526: Nie można wykonać operacji, ponieważ do atrybutu lewostronnego**

*nazwa\_atrybutu* odwołuje się łączenie.:

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego atrybutu odwołuje się obecnie łączenie, dlatego atrybutu nie można usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego atrybutu zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego atrybutu. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6527: Nie można wykonać operacji, ponieważ do modelu kostki *nazwa\_modelu* odwołuje się kostka.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego modelu kostki odwołuje się obecnie kostka, dlatego nie można go usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego modelu kostki zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego modelu kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6528: Nie można wykonać operacji, ponieważ do obiektu faktów kostki**

*nazwa\_obiektu\_faktów\_kostki* odwołuje się kostka.:

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego obiektu faktów kostki odwołuje się obecnie kostka, dlatego nie można go usunąć.

### Działanie użytkownika

Przed usunięciem wskazanego obiektu faktów kostki zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego obiektu faktów kostki. Więcej informacji na temat reguł dotyczących metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

#### **6529: Nie można wykonać operacji, ponieważ do obiektu faktów *nazwa\_obiektu\_faktów* odwołuje się model kostki.:**

#### **Objaśnienie**

Nie można wykonać żądanej operacji, ponieważ naruszyłaby ona istniejące ograniczenie referencyjne między obiektami w katalogu metadanych. Do wskazanego obiektu faktów odwołuje się obecnie model kostki, dlatego nie można usunąć wybranego obiektu faktów.

### Działanie użytkownika

Przed usunięciem wskazanego obiektu faktów zmodyfikuj odwołujące się do niego obiekty, usuwając odwołanie do wskazanego obiektu faktów. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6530: Nie można wykonać operacji, ponieważ do poziomu *nazwa\_poziomu* odwołuje się wymiar.:**

#### Objaśnienie

Nie można wykonać żądanej operacji, ponieważ narusza ona ograniczenie referencyjne wiążące obiekty metadanych w katalogu metadanych. Do tego poziomu odwołuje się aktualnie wymiar, więc nie można go usunąć.

### Działanie użytkownika

Przed usunięciem poziomu zmodyfikuj odwołujące się do niego obiekty, tak aby poziom przestał być celem odwołań. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6531: Nie można wykonać operacji, ponieważ do poziomu *nazwa\_poziomu* odwołuje się hierarchia.:**

#### Objaśnienie

Nie można wykonać danej operacji, ponieważ narusza ona ograniczenie referencyjne wiążące obiekty metadanych w katalogu metadanych. Do tego poziomu odwołuje się aktualnie hierarchia, więc nie można go usunąć.

### Działanie użytkownika

Przed usunięciem poziomu zmodyfikuj odwołujące się do niego obiekty, tak aby poziom przestał być celem odwołań. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6532: Nie można wykonać operacji, ponieważ do poziomu kostki *nazwa\_poziomu* odwołuje się hierarchia kostki.:**

#### Objaśnienie

Nie można wykonać danej operacji, ponieważ narusza ona ograniczenie referencyjne wiążące obiekty metadanych w katalogu metadanych. Do tego poziomu kostki odwołuje się aktualnie hierarchia kostki, więc nie można go usunąć.

### Działanie użytkownika

Przed usunięciem poziomu zmodyfikuj odwołujące się do niego obiekty, tak aby poziom kostki przestał być celem odwołań. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6533: Nie można wykonać tej operacji dla kostki *nazwa\_kostki*. Wymiar kostki i hierarchia kostki muszą się wywodzić z tej samej kostki. Hierarchia kostki musi należeć do wymiaru kostki.:**

## Objaśnienie

Nie można wykonać danej operacji, ponieważ narusza ona ograniczenie referencyjne wiążące obiekty metadanych w katalogu metadanych. Wymiar kostki jest aktualnie celem odwołania wymiaru kostki i hierarchii kostki z wycinka optymalizacji, toteż usunięcie wycinka optymalizacji kostki nie jest teraz możliwe.

## Działanie użytkownika

Przed usunięciem wycinka optymalizacji kostki zmodyfikuj obiekty odwołujące się, tak aby wycinek optymalizacji kostki przestał być celem odwołań. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

**6534: Nie można wykonać tej operacji na poziomie *nazwa\_poziomu*. Nie można usunąć atrybutu pokrewnego z określonego poziomu, ponieważ na odpowiednim poziomie kostki istnieją odwołania do tego atrybutu.:**

## Objaśnienie

Nie można wykonać danej operacji, ponieważ narusza ona ograniczenie referencyjne wiążące obiekty metadanych w katalogu metadanych. Nie można usunąć atrybutu pokrewnego z poziomu, ponieważ do atrybutu pokrewnego odwołuje się odpowiedni poziom kostki.

## Działanie użytkownika

Aby usunąć atrybut pokrewny, zmodyfikuj odwołujący się do niego obiekt poziomu kostki, tak aby zlikwidować odwołanie do atrybutu pokrewnego przeznaczonego do usunięcia. Więcej informacji na temat reguł obowiązujących dla metadanych oraz ograniczeń referencyjnych między obiektami metadanych zawiera podręcznik *DB2 Cube Views Podręcznik i skorowidz*.

# Optymalizacja

## Kody błędów dotyczących optymalizacji

**7001: Nie ma zdefiniowanych kostek dla modelu kostki *nazwa\_modelu*.:**

## Objaśnienie

Dla wskazanego modelu kostki nie ma zdefiniowanych kostek.

## Działanie użytkownika

Jeśli ma być przeprowadzana optymalizacja na potrzeby zapytań wyodrębniających, które odczytują dane z modelu kostki do kostki MOLAP, najpierw należy zdefiniować odpowiednie kostki MOLAP. Optymalizacja na potrzeby zapytań wyodrębniających nie jest możliwa, jeśli nie zostanie zdefiniowana przynajmniej jedna kostka.

**7002: Model kostki *nazwa\_modelu* nie istnieje.:**

## Objaśnienie

Model kostki o podanej nazwie nie jest zdefiniowany.

### Działanie użytkownika

Sprawdź, czy podane zostały poprawne nazwy modelu kostki i schematu. W nazwach i schematach rozróżniana jest wielkość liter. Aby zapoznać się z listą istniejących modeli kostki, użyj Centrum OLAP.

**7003: Nie znaleziono obszaru tabel *nazwa\_obszaru\_tabel*.**

#### Objaśnienie

Obszar tabel o tej nazwie nie jest zdefiniowany.

### Działanie użytkownika

Sprawdź, czy podana została prawidłowa nazwa obszaru tabel.

**7004: Doradca optymalizacyjny nie może ustalić zaleceń.**

#### Objaśnienie

Określone zostały limity ograniczające miejsce na dysku przydzielane na potrzeby optymalizacji tego modelu kostki. Doradca nie mógł wygenerować zaleceń, które pozwoliłyby zmieścić się w narzuconym ograniczeniu.

### Działanie użytkownika

Zwiększ limit miejsca na dysku i uruchom ponownie Doradcę optymalizacyjnego.

**7005: Obszar tabel *nazwa\_obszaru\_tabel* nie może służyć do przechowywania tabel podsumowań.**

#### Objaśnienie

Obszar tabel nie dysponuje typem pamięci masowej wymagany do przechowywania danych tabeli. Obszar tabel musi być obszarem typu REGULAR. Obszary tabel typu LONG, USER TEMPORARY i SYSTEM TEMPORARY nie mogą służyć do przechowywania tabel podsumowań.

### Działanie użytkownika

Określ obszar tabel typu REGULAR przeznaczony do przechowywania tabel podsumowań.

**7006: Obszar tabel *nazwa\_obszaru\_tabel* nie może służyć do przechowywania indeksów.**

#### Objaśnienie

Obszar tabel nie dysponuje typem pamięci masowej wymagany do przechowywania danych indeksu. Obszar tabel musi być obszarem typu REGULAR lub LONG. Obszary tabel typu USER TEMPORARY i SYSTEM TEMPORARY nie mogą służyć do przechowywania indeksów.

### Działanie użytkownika

Określ obszar tabel typu REGULAR lub LONG przeznaczony do przechowywania indeksów.

**7007: Sprawdzenie poprawności optymalizacji modelu kostki *nazwa\_modelu* zakończyło się niepowodzeniem.**

## Objaśnienie

Model kostki i skojarzone z nim obiekty metadanych naruszają przynajmniej jedną z reguł obiektów metadanych, których spełnienie jest niezbędne dla celów optymalizacji. Optymalizacja nie zostanie przeprowadzona.

## Działanie użytkownika

Optymalizacja może zostać przeprowadzona tylko wtedy, gdy model kostki i skojarzone z nim obiekty metadanych spełniają określone reguły na potrzeby optymalizacji. Więcej informacji na temat reguł optymalizacji zawiera *Podręcznik instalacji i użytkownika*.

### **7008: Model kostki nie ma żadnych wymiarów, które miałyby hierarchie zdadne do optymalizacji.:**

#### Objaśnienie

Optymalizacja nie zostanie przeprowadzona, ponieważ Doradca optymalizacyjny nie może znaleźć wymiarów mających hierarchie podlegające optymalizacji.

## Działanie użytkownika

Model kostki musi mieć przynajmniej jeden wymiar z hierarchią nierekurencyjną.

### **7009: W ramach jednej kostki nie można określić więcej niż jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP ani z wyodrębnieniem hybrydowym, ani też jednocześnie wycinków z wyodrębnieniem typu MOLAP i hybrydowym. W kostce *nazwa\_kostki* określono więcej niż jeden typ wyodrębniania dla wycinków optymalizacji.:**

#### Objaśnienie

Podana kostka nie jest optymalizowana, ponieważ określono w niej więcej niż jeden typ wyodrębniania dla wycinka optymalizacji. W każdej kostce dla wycinka optymalizacji z wyodrębnieniem typu MOLAP albo hybrydowym można wybrać tylko raz.

## Działanie użytkownika

Zadbaj o to, by dla każdej kostki wycinek optymalizacji miał tylko jeden typ wyodrębniania, MOLAP albo hybrydowy.

### **7010: Wycinek optymalizacji z rozszerzaniem musi być definiowany na poziomie kostki, który znajduje się poniżej wycinka optymalizacji z wyodrębnieniem hybrydowym w przynajmniej jednym wymiarze kostki.:**

#### Objaśnienie

Wycinek optymalizacji z rozszerzaniem musi być definiowany na poziomie kostki poniżej wycinka optymalizacji z wyodrębnieniem hybrydowym przynajmniej w jednym wymiarze kostki.

## Działanie użytkownika

Zadbaj o to, by wycinek optymalizacji z rozszerzaniem był zdefiniowany na poziomie kostki, który znajduje się poniżej wycinka optymalizacji z wyodrębnieniem hybrydowym w przynajmniej jednym wymiarze.

### **7011: Doradca optymalizacyjny został zatrzymany i nie mógł ustalić żadnych zaleceń w przydzielonym czasie.:**

## Objaśnienie

Doradca optymalizacyjny został zatrzymany operacją stop advise jeszcze przed ustaleniem zaleceń.

## Działanie użytkownika

Zezwól na dłuższe działanie Doradcy optymalizacyjnego, aby umożliwić mu opracowanie zaleceń.

### 7012: Niepoprawny identyfikator.:

#### Objaśnienie

Podany identyfikator nie był poprawny.

### 7013: Podjęto próbę próbkowania tabeli DB2 względem obiektu faktów bazującego na widoku, aliasie, pseudonimie lub innym obiekcie bazy danych, który nie obsługuje próbkowania.:

#### Objaśnienie

Program DB2 Cube Views podjął próbę próbkowania danych z obiektu faktów, który opiera się na widoku, aliasie lub pseudonimie. Próbkowanie tabel DB2 nie jest obsługiwane w przypadku widoków, aliasów i pseudonimów.

## Działanie użytkownika

Program DB2 Cube Views może wykonywać próbkowanie danych w obiektach faktów, które są zdefiniowane wyłącznie na podstawie tabel, a nie na widokach, aliasach lub pseudonimach. Wyłącz opcję próbkowania danych w ustawieniach Doradcy optymalizacyjnego.

### 7014: Doradca optymalizacyjny nie może określić zaleceń.:

#### Objaśnienie

Doradca optymalizacyjny nie może ustalić żadnych zaleceń dla modelu kostki przy użyciu podanych parametrów.

## Działanie użytkownika

Sprawdź treść komunikatów informacyjnych i ostrzegawczych, aby się dowiedzieć, dlaczego Doradca optymalizacyjny nie może zalecić żadnych tabel podsumowań.

## Ostrzeżenia dotyczące optymalizacji

### 7200: Zalecane tabele podsumowań będą korzystały z odświeżania odroczonego, ponieważ model kostki zawiera przynajmniej jedną miarę niedystrybuowalną.:

#### Objaśnienie

Dla tabel podsumowań włączona została opcja natychmiastowego odświeżania. Jednak tabele podsumowań nie mogą być odświeżane natychmiastowo, jeśli w modelu kostki są zdefiniowane miary niedystrybuowalne. Miary dystrybuowalne korzystają z prostych funkcji agregujących, jak SUM i COUNT, które można stosować dla dowolnych wartości pośrednich. Miary niedystrybuowalne korzystają z bardziej złożonych funkcji agregujących, jak STDDEV, i muszą być stosowane dla tabel podstawowych.

### Działanie użytkownika

Jeśli nie jest istotne utrzymanie synchronizacji między tabelami podsumowań a tabelami podstawowymi, nie jest wymagane żadne działanie. Jeśli tabele podsumowań muszą być zsynchronizowane z tabelami podstawowymi, należy zmodyfikować metadane tak, aby zdefiniowane były jedynie miary dystrybuowalne.

**7201: Zalecana tabela podsumowań *nazwa\_tabeli* będzie korzystać z odświeżania odroczonego, ponieważ w instrukcji pełnej selekcji tej tabeli jako kolumna występuje przynajmniej jeden atrybut dopuszczający wartość pustą.:**

#### Objaśnienie

Dla tabel podsumowań włączona została opcja natychmiastowego odświeżania. Jednak zalecana tabela podsumowań ma przynajmniej jeden atrybut, który jest używany jako kolumna dopuszczająca wartość pustą w wyrażeniu pełnej selekcji tabeli podsumowań. Stosowanie kolumn dopuszczających wartość pustą w wyrażeniu pełnej selekcji tabeli podsumowań może obniżać wydajność odświeżania natychmiastowego. Aby uniknąć tego problemu z wydajnością dla tej tabeli podsumowań ustawiona została opcja *odświeżania odroczonego*.

### Działanie użytkownika

Aby zmienić dopuszczalność wartości pustej atrybutu, należy zmodyfikować wyrażenie SQL atrybutu lub zmienić dopuszczalność wartości pustej w kolumnach tabeli DB2 używanych przez atrybut. Zmiany te nie są zwykle zalecane, ponieważ mogą być trudne do wykonania.

**7202: Tabela *nazwa\_tabeli* nie ma statystyki.:**

#### Objaśnienie

Doradca optymalizacyjny nie może znaleźć poprawnych wartości statystyki tabeli dla wskazanej tabeli.

### Działanie użytkownika

Użyj komendy RUNSTATS, aby utworzyć statystykę dla wskazanej tabeli. Następnie ponownie uruchom Doradcę optymalizacyjnego.

**7203: W przypadku wycinka optymalizacji z wyodrębnianiem hybrydowym *nazwa\_wycinka* w kostce *nazwa\_kostki* należy określić poziom kostki dla każdego wymiaru kostki.:**

#### Objaśnienie

Doradca optymalizacyjny nie może przeprowadzić optymalizacji dla wycinka z wyodrębnianiem hybrydowym, ponieważ wycinek ten nie jest należycie zdefiniowany. Wycinek optymalizacji z wyodrębnianiem hybrydowym musi mieć zdefiniowany poziom kostki dla każdego wymiaru kostki. Nie można wybrać opcji Nieokreślony.

### Działanie użytkownika

Zadbaj o to, by wycinek optymalizacji miał zdefiniowany poziom kostki dla każdego wymiaru kostki.

**7203: W przypadku wycinka optymalizacji z wyodrębnianiem hybrydowym *nazwa\_wycinka* w kostce *nazwa\_kostki* należy określić poziom kostki dla każdego wymiaru kostki.:**

## Objaśnienie

Doradca optymalizacyjny nie może przeprowadzić optymalizacji dla wycinka z wyodrębnianiem hybrydowym, ponieważ wycinek ten nie jest należycie zdefiniowany. Wycinek optymalizacji z wyodrębnianiem hybrydowym musi mieć zdefiniowany poziom kostki dla każdego wymiaru kostki. Nie można wybrać opcji Nieokreślony.

### Działanie użytkownika

Zadbaj o to, by wycinek optymalizacji miał zdefiniowany poziom kostki dla każdego wymiaru kostki.

**7205: W zalecanych tabelach podsumowań musi zostać użyta metoda aktualizacji z odświeżaniem odroczonym, ponieważ co najmniej jedna bazowa tabela podstawowa w modelu kostki jest określona przy użyciu pseudonimu.:**

### Objaśnienie

Dla tabel podsumowań wybrano opcję odświeżania natychmiastowego; tabele podsumowań nie mogą korzystać z metody odświeżania natychmiastowego, jeśli chociaż jedna tabela bazowa jest pseudonimem.

### Działanie użytkownika

Nie jest wymagane żadne działanie.

**7206: Doradca optymalizacyjny nie może użyć próbkowania danych do określenia zaleconych tabel podsumowań.:**

### Objaśnienie

Włączono opcję, zgodnie z którą Doradca optymalizacyjny powinien korzystać z próbkowania danych przy ustalaniu optymalnych zaleceń odnośnie tabeli podsumowań. Doradca optymalizacyjny nie może wykonać próbkowania danych ze względu na sposób zdefiniowania obiektu faktów. Jeśli obiekt faktów jest zdefiniowany jako pseudonim, Doradca optymalizacyjny podejmuje próbę próbkowania danych, lecz próba ta może się powieść tylko pod warunkiem, że wymiary modelu kostki mają umiarkowaną licznosc, a łączenia wymiarów z obiektem faktów są zbudowane na bazie pojedynczego atrybutu odwzorowanego do pojedynczej kolumny. Indeksy zdefiniowane dla tych kolumn tabeli faktów, które pełnią rolę kluczy wymiarów, mogą także wpływać na możliwość realizacji próbkowania danych przez Doradcę optymalizacyjnego, w przypadku obiektu faktów zdefiniowanego jako pseudonim. Jeśli obiekt faktów jest zdefiniowany jako widok, próbkowanie danych nie jest obsługiwane. Jeśli obiekt faktów jest zdefiniowany jako alias, próbkowanie danych może nie być możliwe, zależnie od tego, do czego alias jest odwzorowany. Skutkiem wygenerowania zaleceń bez próbkowania danych jest uzyskanie tabel podsumowań, które nie są optymalne. Jeśli obiekt faktów jest tabelą, Doradca optymalizacyjny zawsze może wykonać próbkowanie danych.

### Działanie użytkownika

Zalecenia można wykorzystać w otrzymanej postaci lub też próbować je ulepszyć, podając obiekt faktów w postaci tabeli zamiast widoku lub aliasu i uruchamiając Doradcę optymalizacyjnego ponownie.

## Komunikaty informacyjne dotyczące optymalizacji

**7400: Tabele podsumowań zostały zdefiniowane za pomocą operatora ROLLUP, ponieważ model kostki zawiera przynajmniej jedną miarę niedystrybuwalną.:**



### Objaśnienie

Miary mogą być dystrybuowalne lub niedystrybuowalne. Miary dystrybuowalne korzystają z prostych funkcji agregujących, jak SUM i COUNT, które można stosować dla dowolnych wartości pośrednich. Miary niedystrybuowalne korzystają z bardziej złożonych funkcji agregujących, jak STDDEV, i muszą być stosowane dla tabel podstawowych. Aby wyeliminować koszt agregacji miar niedystrybuowalnych z tabel podstawowych, tabele podsumowań są definiowane z użyciem operatora ROLLUP, którego zadaniem jest wstępna agregacja miar niedystrybuowalnych.

### Działanie użytkownika

Nie jest wymagane żadne działanie.

**7401: Zalecana jest tabela podsumowań *nazwa\_tabeli*. Ma ona szacunkowo *wiersze* wierszy, wielkość *n* MB i wielkość indeksu *n* MB.:**

### Objaśnienie

Ten komunikat stanowi opis zalecanej tabeli podsumowań, obejmujący zalecaną liczbę wierszy, szacowaną zajętość dysku i szacowany rozmiar indeksu.

### Działanie użytkownika

Nie jest wymagane żadne działanie.

**7402: Istniejące tabele podsumowań w liczbie *n* nie mieszczą się w określonym limicie miejsca na dysku. Ich łączna szacowana wielkość wynosi *n* MB.:**

### Objaśnienie

Ten komunikat zawiera informacje na temat zalecanych tabel podsumowań, które nie mieszczą się w przydzielonym limicie miejsca na dysku.

### Działanie użytkownika

Aby wyświetlić te tabele podsumowań w zaleceniach, uruchom ponownie Doradcę optymalizacyjnego, zwiększając limit miejsca na dysku.

**7403: Zalecenia obejmują optymalizacje dla kostki *nazwa\_kostki*.:**

### Objaśnienie

Dla wskazanej kostki zalecane są tabele podsumowań. Niektóre z zapytań dla tej kostki zostaną zoptymalizowane.

### Działanie użytkownika

Nie jest wymagane żadne działanie.

**7404: Zalecenia nie obejmują optymalizacji dla kostki *nazwa\_kostki*.:**

### Objaśnienie

Brak jest zalecanych tabel podsumowań dla tej kostki. W zapytaniach dotyczących tej kostki nie należy się spodziewać zauważalnego wzrostu wydajności.

### Działanie użytkownika

Jeśli tabele podsumowań nie zostały zawarte w zaleceniach ze względu na ograniczoną dostępność miejsca na dysku, uruchom Doradcę optymalizacyjnego ponownie, przydzielając wyższy limit miejsca. Zalecenia mogą obejmować jedną lub kilka tabel podsumowań optymalizujących zapytania dla tej kostki.

#### **7405: W trakcie ustalania zaleceń przez Doradcę optymalizacyjnego upłynął wyznaczony limit czasu.:**

##### **Objaśnienie**

Doradca optymalizacyjny sformułował zalecenie. W przypadku przyznania większego limitu czasu Doradca optymalizacyjny mógłby zaproponować lepsze zalecenia, gdyż miałby wtedy więcej czasu na dodatkową analizę. Jednak wydłużanie czasu działania Doradcy optymalizacyjnego nie stanowi gwarancji uzyskania lepszych wyników.

### Działanie użytkownika

Można uruchomić Doradcę optymalizacyjnego ponownie, wydłużając przyznany mu limit czasu, lub utworzyć zalecane tabele podsumowań i sprawdzić, czy uzyskany wzrost wydajności jest zadowalający.

#### **7406: Wymiar *nazwa\_wymiaru* nie zawiera żadnych hierarchii zdalnych do optymalizacji przez Doradcę optymalizacyjnego.:**

##### **Objaśnienie**

Doradca optymalizacyjny nie może prowadzić optymalizacji dla hierarchii rekurencyjnych. Wskazany wymiar nie zawiera żadnych hierarchii, które byłyby zdalne do optymalizacji, dlatego Doradca optymalizacyjny zignoruje ten wymiar. Zapytania odwołujące się do atrybutów tego wymiaru nie będą zoptymalizowane.

### Działanie użytkownika

Nie jest wymagane żadne działanie. Zapytania odwołujące się do atrybutów tego wymiaru nie wykażą żadnego wzrostu wydajności.

#### **7407: Zalecane tabele podsumowań zapewniają optymalizację dla *n* procent wycinków w modelu kostki. Zapytania uruchamiane względem zoptymalizowanych wycinków powinny charakteryzować się wyższą wydajnością.:**

##### **Objaśnienie**

Zapytania SQL odwołują się do określonych wycinków w modelu kostki. Jednym ze sposobów analizy wzrostu wydajności jest rozważanie, jaka część wycinków będących przedmiotem zapytania zostanie usprawniona. Jeśli w modelu kostki używane są miary dystrybuowalne, zapytania odwołujące się do wycinków umieszczonych logicznie powyżej wycinka tabeli podsumowań wykażą poprawę wydajności.

Na przykład w modelu kostki mającym wymiar Czas z hierarchią [Wszystko-Rok-Kwartał-Miesiąc-Dzień] oraz wymiar Region z hierarchią [Wszystko-Kraj-Region-Stan-Miast-Sklep], istnieje 30 możliwych wycinków. Liczbę możliwych wycinków można obliczyć, mnożąc przez siebie liczby poziomów w hierarchiach wymiarów. Jeśli zalecana tabela podsumowań zapewnia optymalizację w wycinku Miesiąc-Miasto, optymalizacja ta obejmie wszystkie wycinki na tym samym poziomie i na wyższych poziomach. W tym przykładzie optymalizacją objętych zostanie 20 spośród 30 możliwych wycinków, czyli 67% (20/30). Wartość ta nigdy nie osiągnie 100%, ponieważ wymagałoby to powielenia tabel

podstawowych w tabelach podsumowań. Zazwyczaj optymalizacja najniżej położonych wycinków jest najmniej korzystna, gdyż wycinki te nie odbiegają zbyt od tabel podstawowych.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie. Jeśli uzyskana wartość jest niska, można ponownie uruchomić Doradcę optymalizacyjnego, podając większy limit miejsca na dysku.

#### **7408: Odczytywanie metadanych modelu kostki z bazy danych.:**

##### **Objaśnienie**

Doradca optymalizacyjny odczytuje metadane opisujące model kostki. Metadane zawierają informacje istotne dla zaleceń optymalizacyjnych.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

#### **7409: Wybieranie, które agregacje mają być uwzględniane w tabelach podsumowań.:**

##### **Objaśnienie**

Doradca optymalizacyjny testuje potencjalne konfiguracje tabel podsumowań w celu ustalenia, która z nich jest optymalna w kontekście podanych kryteriów.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

#### **7410: Próbkowanie danych z modelu kostki.:**

##### **Objaśnienie**

Doradca optymalizacyjny odczytuje podzbiór danych z tabel faktów i wymiarów, dążąc do ustalenia wielkości tabeli podsumowań. Próbkowanie może następować wiele razy, gdyż Doradca optymalizacyjny analizuje duże tabele podsumowań.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

#### **7411: Definiowanie indeksów dla zalecanych tabel podsumowań.:**

##### **Objaśnienie**

Doradca optymalizacyjny ustalił zalecane tabele podsumowań i dobiera indeksy, które zostaną zbudowane na ich potrzeby.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

#### **7413: Dla kostki *nazwa\_kostki* nie zdefiniowano żadnych wycinków optymalizacji, toteż Doradca optymalizacyjny zoptymalizuje tę kostkę pod kątem zapytań z drażeniem.:**

##### **Objaśnienie**

Jeśli dla kostki nie zdefiniowano wycinków optymalizacji, Doradca optymalizacyjny zoptymalizuje kostkę pod kątem zapytań z drażeniem.

**7414: Dla optymalizowanego modelu kostki nie zdefiniowano żadnej kostki, toteż Doradca optymalizacyjny przeprowadzi optymalizację modelu kostki, tak jakby zawierał on kostkę dla zapytań typu raportowego.:**

**Objaśnienie**

Jeśli na podstawie modelu kostki nie są zdefiniowane żadne kostki, Doradca optymalizacyjny może zoptymalizować ten model kostki, zakładając istnienie hipotetycznej kostki używanej dla zapytań drażących. Jeśli nie zdefiniowano żadnych kostek dla optymalizowanego modelu kostki, Doradca optymalizacyjny utworzy zalecenia, zakładając istnienie hipotetycznej kostki ze zdefiniowanym wycinkiem optymalizacji na potrzeby zapytań drażących na poziomie Wszystkie All w każdym wymiarze kostki.

**7415: Doradca optymalizacyjny nie zalecił żadnej tabeli podsumowań dla wycinka optymalizacji *nazwa\_wycinka* w kostce *nazwa\_kostki*.:**

**Objaśnienie**

Dla wskazanego wycinka optymalizacji nie jest zalecana żadna tabela podsumowań.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**7416: Operacja została zatrzymana. Zalecenia mogą nie być optymalne.:**

**Objaśnienie**

Doradca optymalizacyjny został zatrzymany w wyniku jawnej operacji zatrzymania (stop advise). Zalecenia mogą nie być optymalne.

**Działanie użytkownika**

Aby uzyskać zalecenia optymalne, należy Doradcę optymalizacyjnego uruchomić ponownie, nie wprowadzając ograniczeń czasowych.

**7417: Doradca optymalizacyjny wstrzymał generowanie zaleceń ze względu na osiągnięcie ustawionego limitu czasu. Doradca optymalizacyjny może zaproponować lepsze tabele podsumowań, jeśli otrzyma na to więcej czasu.:**

**Objaśnienie**

Czas wyznaczony na działanie Doradcy optymalizacyjnego może być zbyt krótki. Zaleca się uruchomienie Doradcy optymalizacyjnego z dłuższym limitem czasu.

**Działanie użytkownika**

Uruchom Doradcę optymalizacyjnego z wydłużonym limitem czasu. Wówczas Doradca optymalizacyjny będzie mógł sformułować lepsze zalecenia odnośnie tabel podsumowań.

**7418: Doradca optymalizacyjny nie może zalecić tabeli podsumowań dla kostki *nazwa\_kostki*, ponieważ tabela podsumowań ma zbyt wiele wierszy w porównaniu z tabelą faktów.:**

**Objaśnienie**

Doradca optymalizacyjny nie może zalecić tabeli podsumowań dla tej kostki. Liczba wierszy w tabeli podsumowań stanowi zbyt wysoki odsetek liczby wierszy w tabeli faktów.

### Działanie użytkownika

Jeśli zdefiniowano wycinki optymalizacji, przynajmniej jeden z nich może mieć zbyt wiele poziomów lub zawierać poziomy o stopniu szczególności za mało odbiegającym od wartości w tabeli faktów. W tej sytuacji można wybrać opcję Dowolny poziom dla niektórych wymiarów kostki lub wybrać poziomy znajdujące się bliżej szczytu hierarchii. Oprócz tego warto rozważyć przydzielenie dłuższego czasu na działanie Doradcy optymalizacyjnego, aby uzyskać zalecenia wyższej jakości.

---

## Komunikaty Centrum OLAP

### 10000-10600

#### **10000: Centrum OLAP nie może pobrać żadnych nazw baz danych.**

##### Objaśnienie

Podczas pobierania listy nazw baz danych z programu DB2 wystąpił błąd.

##### Działanie użytkownika

Sprawdź, czy Centrum OLAP zostało poprawnie zainstalowane. Jeśli problem nie ustąpi, skontaktuj się z działem wsparcia firmy IBM.

#### **10001: Wpisz nazwę użytkownika.**

##### Objaśnienie

Pole **Nazwa użytkownika** jest puste.

##### Działanie użytkownika

Wpisz nazwę użytkownika w polu **Nazwa użytkownika**.

#### **10002: Wpisz hasło.**

##### Objaśnienie

Pole **Hasło** jest puste.

##### Działanie użytkownika

Wpisz hasło w polu **Hasło**.

#### **10004: Nie można przeprowadzić analizy składniowej atrybutu w wyrażeniu SQL.**

##### Objaśnienie

Wprowadzone wyrażenie SQL odwołuje się do atrybutu, który ani nie jest poprawny w danym kontekście, ani nie istnieje w bazie danych.

##### Działanie użytkownika

Zadbaj o to, by wyrażenie SQL odwoływało się tylko do atrybutów widocznych w sekcji **Dane** Kreatora wyrażeń SQL.

**10005: Obiekty metadanych zostały pomyślnie wyeksportowane do pliku *nazwa\_pliku*.**

**Objaśnienie**

Eksport został przeprowadzony pomyślnie.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10006: Wprowadź nazwę pliku, do którego wyeksportowane zostaną obiekty metadanych.**

**Objaśnienie**

Moduł eksportujący zapisuje eksportowane obiekty metadanych w pliku o nazwie wpisanej przez użytkownika.

**Działanie użytkownika**

Wpisz odpowiednią nazwę w polu **Nazwa pliku**.

**10007: Wybierz kostkę lub model kostki do wyeksportowania.**

**Objaśnienie**

Okno eksportu pozwala wyeksportować kostkę lub model kostki.

**Działanie użytkownika**

Wybierz obiekt przeznaczony do eksportu.

**10008: Wprowadź unikalną nazwę tworzonego obiektu.**

**Objaśnienie**

W Kreatorze wyrażeń SQL puste jest pole **Nazwa** dotyczące tworzonego atrybutu lub miary.

**Działanie użytkownika**

Wpisz nazwę obiektu w polu **Nazwa**. Nazwa obiektu musi być unikalna w obrębie przestrzeni nazw atrybutów i miar.

**10009: Wprowadź wyrażenie SQL dla obiektu.**

**Objaśnienie**

Pole wyrażenia SQL obiektu jest puste.

**Działanie użytkownika**

Wprowadź wyrażenie SQL dla obiektu.

**10010: Kolumna *nazwa\_kolumny* nie ma kwalifikatora w postaci nazwy tabeli.**

**Objaśnienie**

Nazwa kolumny wprowadzona w wyrażeniu SQL nie jest kwalifikowana nazwą tabeli.

#### Działanie użytkownika

Upewnij się, że nazwa kolumny w wyrażeniu SQL ma kwalifikator w postaci nazwy tabeli oddzielony separatorem w postaci kropki '.'.

#### **10011: Kolumna *nazwa\_kolumny* nie ma kwalifikatora w postaci nazwy schematu.**

##### Objaśnienie

Odwołania do kolumn w wyrażeniu SQL muszą być kwalifikowane nazwą tabeli i nazwą schematu, rozdzielonymi kropką '.'.

#### Działanie użytkownika

Zadbaj o to, by odwołania do kolumn w wyrażeniu SQL były kwalifikowane nazwą tabeli i nazwą schematu, rozdzielonymi kropką '.'.

#### **10012: Pierwszy element w skrypcie agregacji nie może być wymiarem.**

##### Objaśnienie

Utworzono skrypt agregacji, w którym pierwszym elementem jest wymiar.

#### Działanie użytkownika

Użyj funkcji agregującej jako pierwszego elementu w skrypcie agregacji.

#### **10013: Wybierz istniejącą miarę lub wprowadź wyrażenie SQL jako drugi parametr wieloparametrowej funkcji *nazwa\_funkcji* w agregacji.**

##### Objaśnienie

W funkcji wieloparametrowej w skrypcie agregacji pierwszy parametr jest zdefiniowany jako miara, z którą agregacja jest skojarzona. Jako drugi parametr należy wybrać istniejącą miarę lub wprowadzić wyrażenie SQL.

#### Działanie użytkownika

Wprowadź miarę lub wyrażenie SQL jako drugi parametr.

#### **10014: Brak zgodnych wymiarów dla funkcji agregującej *nazwa\_funkcji*.**

##### Objaśnienie

Każda funkcja agregująca w skrypcie agregacji musi zostać zastosowana przynajmniej do jednego wymiaru.

#### Działanie użytkownika

Zadbaj o to, by każda funkcja agregująca w skrypcie agregacji została zastosowana przynajmniej do jednego wymiaru.

**10015: Brak nawiasu zamykającego w obiekcie *nazwa\_objektu*.**

**Objaśnienie**

Atrybuty, miary i kolumny użyte w wyrażeniu w Centrum OLAP muszą być ujęte w odpowiednie operatory: @Attribute(), @Measure() lub @Column().

**Działanie użytkownika**

Wprowadź nawias zamykający dla obiektu.

**10016: Odwołanie do obiektu *nazwa\_objektu* w wyrażeniu tworzy niedopuszczalną pętlę odwołań.**

**Objaśnienie**

Obiekt w swoim wyrażeniu SQL odwołuje się sam do siebie.

**Działanie użytkownika**

Zadbaj o to, by obiekty w wyrażeniu SQL nie tworzyły pętli odwołań.

**10017: Nie napotkano żadnych błędów. Wyrażenie SQL jest poprawne.**

**Objaśnienie**

Wyrażenie SQL jest poprawne.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10018: Nie napotkano żadnych błędów. Skrypt agregacji jest poprawny.**

**Objaśnienie**

Zestaw agregacji w skrypcie agregacji jest poprawny.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10020: Wpisz nazwę.**

**Objaśnienie**

Pole **Nazwa** dla obiektu jest puste.

**Działanie użytkownika**

Wpisz nazwę obiektu w polu **Nazwa**.

**10021: Wpisz nazwę schematu.**

**Objaśnienie**

Pole **Schemat** jest puste.



#### **Działanie użytkownika**

Wpisz nazwę schematu w polu **Schemat**.

#### **10022: Wpisz nazwę biznesową.**

##### **Objaśnienie**

Pole **Nazwa biznesowa** jest puste.

#### **Działanie użytkownika**

Nazwa biznesowa może być wyświetlana w aplikacjach do inteligentnej analizy danych, stanowiąc identyfikator obiektu dla użytkownika końcowego. Wpisz odpowiednią nazwę w polu **Nazwa biznesowa**.

#### **10023: Wybierz przynajmniej jeden poziom do uwzględnienia w hierarchii kostki.**

##### **Objaśnienie**

W hierarchii kostki nie są określone żadne poziomy.

#### **Działanie użytkownika**

Wybierz przynajmniej jeden poziom do uwzględnienia w hierarchii kostki.

#### **10024: Wybierz przynajmniej jedną miarę dla faktów kostki.**

##### **Objaśnienie**

Dla obiektu faktów kostki nie określono żadnych miar.

#### **Działanie użytkownika**

Wybierz przynajmniej jedną miarę dla faktów kostki.

#### **10025: Wybierz przynajmniej jeden poziom do uwzględnienia w hierarchii kostki.**

##### **Objaśnienie**

W hierarchii kostki nie są określone żadne atrybuty.

#### **Działanie użytkownika**

Wybierz przynajmniej jeden atrybut do uwzględnienia w hierarchii kostki.

#### **10026: Wybierz przynajmniej jeden wymiar kostki.**

##### **Objaśnienie**

W kostce nie określono żadnych wymiarów.

#### **Działanie użytkownika**

Wybierz przynajmniej jeden wymiar, a następnie kliknij przycisk [...], aby określić szczegóły wymiaru kostki.

**10027: Anulowano zaznaczenie przynajmniej jednego z wymiarów istniejących obecnie w kostce. Kliknij przycisk Tak, aby usunąć wymiary kostki. Kliknij przycisk Nie, aby powrócić do okna, a następnie przycisk Anuluj, aby zamknąć okno bez zapisywania zmian.**

**Objaśnienie**

Anulowano przynajmniej jedno z zaznaczeń wymiarów. Odpowiednie wymiary kostki zostaną usunięte z kostki.

**Działanie użytkownika**

Kliknij przycisk **Tak** w oknie, aby usunąć wymiary kostki z kostki. Kliknij przycisk **Nie**, aby zachować wymiary kostki, a następnie kliknij przycisk **Anuluj**, aby zamknąć okno bez zapisywania.

**10028: Związek atrybutów nie może mieć licznosci Wiele:Wielu, jeśli zaznaczone jest pole wyboru zależności funkcjonalnej.**

**Objaśnienie**

Nie można definiować związku atrybutów z wybraną licznością *wiele:wielu* i z zależnością funkcjonalną.

**Działanie użytkownika**

Wybierz inną liczność związku atrybutów lub usuń zaznaczenie opcji zależności funkcjonalnej.

**10029: Wybierz lewostronny i prawostronny atrybut w związku atrybutów.**

**Objaśnienie**

Nie można zdefiniować związku atrybutów, jeśli nie jest wybrany jednocześnie atrybut lewo- i prawostronny.

**Działanie użytkownika**

Wybierz atrybut lewo- i prawostronny.

**10030: W związku atrybutów ten sam atrybut nie może być jednocześnie atrybutem lewostronnym i prawostronnym.**

**Objaśnienie**

Nie można zdefiniować związku atrybutów, w którym atrybut lewostronny jest tożsamy z atrybutem prawostronnym.

**Działanie użytkownika**

Wybierz różne atrybuty lewostronne i prawostronne.

**10031: W bazie danych istnieje już obiekt o podanej nazwie i schemacie. Wpisz inną nazwę.**

**Objaśnienie**

W bazie danych istnieje już obiekt tego samego typu i o takiej samej nazwie i schemacie, jak obiekt tworzony lub modyfikowany.

#### **Działanie użytkownika**

Wprowadź unikalną nazwę obiektu.

#### **10032: Określ przynajmniej jedną parę atrybutów.**

##### **Objaśnienie**

Łączenie musi mieć przynajmniej jedną parę atrybutów.

#### **Działanie użytkownika**

Określ przynajmniej jedną parę atrybutów.

#### **10033: Nie można tworzyć powielonych par atrybutów.**

##### **Objaśnienie**

Istnieje już para atrybutów odpowiadająca nowym ustawieniom.

#### **Działanie użytkownika**

Wybierz różne atrybuty lewostronne i prawostronne.

#### **10034: Wybierz przynajmniej jedną tabelę.**

##### **Objaśnienie**

Nie wybrano żadnych tabel.

#### **Działanie użytkownika**

Wybierz przynajmniej jedną tabelę, aby kontynuować.

#### **10035: Wybierz lub utwórz nowe łączenia, aby połączyć wszystkie wybrane tabele.**

##### **Objaśnienie**

Nie wybrano żadnych łączeń.

#### **Działanie użytkownika**

Wybierz lub utwórz nowe łączenia, które połączą wszystkie wybrane tabele.

#### **10036: Wybierz przynajmniej jeden atrybut.**

##### **Objaśnienie**

Nie wybrano żadnych atrybutów.

#### **Działanie użytkownika**

Wybierz przynajmniej jeden atrybut.

#### **10037: Wybierz łączenie w celu połączenia wymiaru z obiektem faktów.**

##### **Objaśnienie**

Nie wybrano żadnych łączeń.

#### Działanie użytkownika

Wybierz łączenie, które połączy wymiar z obiektem faktów.

**10038: Określ dokładnie jedno łączenie między dwiema danymi tabelami. Łączenie *nazwa\_łączenia1* i łączenie *nazwa\_łączenia2* łączą ze sobą te same tabele.**

**Objaśnienie**

Dla jednej pary wybrano więcej niż jedno łączenie.

#### Działanie użytkownika

Wybierz dokładnie jedno łączenie dla każdej pary tabel.

**10039: Wszystkie wybrane tabele muszą być połączone. Wybierz łączenie dla tabeli *nazwa\_tabeli*.**

**Objaśnienie**

Wszystkie wybrane tabele muszą być połączone.

#### Działanie użytkownika

Wybierz łączenie dla podanej tabeli.

**10040: Liczba wybranych tabel nie jest zgodna z liczbą wybranych łączeń. Sprawdź, czy nie ma łączeń zapętlnionych, oraz czy każda tabela ma swoje łączenie.**

**Objaśnienie**

Wszystkie wybrane tabele muszą być połączone.

#### Działanie użytkownika

Zadbaj o to, by nie było łączeń zapętlnionych, oraz aby wszystkie tabele były połączone.

**10042: Wybierz przynajmniej jedną miarę.**

**Objaśnienie**

Nie określono żadnych miar.

#### Działanie użytkownika

Wybierz przynajmniej jedną miarę.

**10043: Wybierz kolumnę tabeli.**

**Objaśnienie**

Nie określono kolumny tabeli.

#### Działanie użytkownika

Wybierz kolumnę.

### **10044: Wybierz wyrażenie SQL.**

#### **Objaśnienie**

Nie określono wyrażenia SQL.

#### **Działanie użytkownika**

Kliknij przycisk **Buduj wyrażenie**, aby zbudować wyrażenie.

### **10045: Nie określono skryptu agregacji.**

#### **Objaśnienie**

Nie określono skryptu agregacji.

#### **Działanie użytkownika**

Kliknij przycisk **Buduj skrypt**, aby zbudować skrypt agregacji.

### **10046: Wybierz miarę przed otwarciem kreatora wyrażeń.**

#### **Objaśnienie**

Nie wybrano żadnej miary.

#### **Działanie użytkownika**

Wybierz miarę z tabeli.

### **10047: Wybierz miarę przed otwarciem kreatora skryptów agregacji.**

#### **Objaśnienie**

Nie wybrano żadnej miary.

#### **Działanie użytkownika**

Wybierz miarę.

### **10048: Nie można uruchomić kreatora skryptów agregacji dla miary *nazwa\_miary*, ponieważ model kostki *nazwa\_modelu* nie zawiera żadnego wymiaru.**

#### **Objaśnienie**

Nie można określić skryptu agregacji, jeśli model kostki nie ma przynajmniej jednego wymiaru.

#### **Działanie użytkownika**

Przed określeniem skryptu agregacji dodaj wymiary do modelu kostki.

### **10049: Aby edytować wyrażenie, określ atrybut.**

#### **Objaśnienie**

Nie wybrano atrybutu.

### Działanie użytkownika

Wybierz atrybut.

**10050: Metadane zostaną odświeżone na podstawie bazy danych. Wszystkie zmiany wprowadzane w chwili wystąpienia błędu zostaną utracone.**

#### Objaśnienie

Wystąpił błąd podczas wywoływania procedury zapisanej w bazie DB2.

### Działanie użytkownika

Kliknij przycisk **OK**, aby odświeżyć metadane wyświetlane w Centrum OLAP. Wszystkie zmiany wprowadzane w chwili wystąpienia błędu zostaną utracone. Obiekty wyświetlane w Centrum OLAP zostaną odświeżone na podstawie odpowiednich obiektów w bazie danych i będzie możliwe kontynuowanie pracy.

**10051: Sprawdzenie poprawności modelu kostki *nazwa\_modelu* w celu przeprowadzenia optymalizacji nie jest możliwe. Program DB2 zwrócił następujący komunikat: *komunikat*.**

#### Objaśnienie

Centrum OLAP nie może uruchomić Doradcy optymalizacyjnego dla wybranego modelu kostki, ponieważ wybrany model kostki nie przeszedł pomyślnie procedury sprawdzenia poprawności wykonanej przez procedurę API zapisaną w bazie.

### Działanie użytkownika

Odszukaj w dokumentacji API procedury informacje na temat sprawdzania poprawności modelu kostki. Wykonaj instrukcje zawarte w komunikacie przekazany przez program DB2.

**10052: Niektóre z załadowanych atrybutów lub miar są odwzorowane do kolumn *nazwy\_kolumn*, które już nie istnieją w bazie danych. Wyeliminuj ten problem, odtwarzając tabele, do których należą brakujące kolumny, lub usuwając niepoprawne atrybuty i/lub miary.**

#### Objaśnienie

Ten komunikat jest wyświetlany podczas uruchamiania Centrum OLAP lub po wybraniu opcji: **Widok -> Odśwież**. Przyczyną jego wyświetlenia jest brak tabeli, do której odwzorowane są ładowane atrybuty lub miary, a która została usunięta lub której zmieniono nazwę.

### Działanie użytkownika

Wyeliminuj problem w jeden z następujących sposobów:

- Odtwórz usuniętą tabelę lub przywróć poprzednią nazwę tabeli.
- Odwzoruj atrybuty/miary do tabeli, która istnieje w bazie danych.
- Usuń atrybuty/miary odwzorowane do nieistniejących kolumn.

**10053: W wyniku sprawdzenia optymalizacji modelu kostki *nazwa\_modelu* zwrócone zostało ostrzeżenie. Program DB2 zwrócił następujący komunikat: *komunikat***

**Objaśnienie**

W Centrum OLAP model kostki został poddany sprawdzeniu poprawności przed uruchomieniem Doradcy optymalizacyjnego i program DB2 zwrócił ostrzeżenie. Ostrzeżenie to może oznaczać, że danego modelu kostki nie da się zoptymalizować. Na przykład model kostki może zawierać widoki odwołujące się do tabel, między którymi nie są zdefiniowane żadne ograniczenia.

**Działanie użytkownika**

Sprawdź komunikat zwrócony przez program DB2 i zdecyduj, czy nadal chcesz uruchamiać Doradcę optymalizacyjnego.

**10060: Model kostki nie jest kompletny. Aby utworzenie kostki było możliwe, model kostki musi zawierać obiekt faktów oraz przynajmniej jeden wymiar i przynajmniej jedną hierarchię dla każdego z wymiarów.**

**Objaśnienie**

Aktualny stan modelu kostki nie umożliwia utworzenia kostki.

**Działanie użytkownika**

Zmodyfikuj model kostki tak, aby zawierał on obiekt faktów i przynajmniej jeden wymiar. Zadbaj o to, by każdy wymiar miał przynajmniej jedną hierarchię.

**10061: Skasowanie modelu kostki powoduje także usunięcie jego wymiarów oraz skasowanie jego faktów. Usunięte wymiary nadal będą dostępne w folderze Wszystkie wymiary. Czy na pewno chcesz skasować model kostki *nazwa\_modelu*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10062: Skasowanie wymiaru powoduje także skasowanie jego hierarchii i odpowiadających mu wymiarów kostki. Czy na pewno chcesz skasować wymiar *nazwa\_wymiaru*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10063: Skasowanie kostki powoduje także skasowanie jej wymiarów kostki, hierarchii kostki i faktów kostki. Czy na pewno chcesz skasować kostkę *nazwa\_kostki*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10064: Skasowanie wymiaru kostki powoduje także skasowanie jego hierarchii kostki. Czy na pewno chcesz skasować wymiar kostki *nazwa\_wymiaru\_kostki*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10065: Czy na pewno chcesz skasować obiekt *nazwa\_obiektu*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10066: Usunięcie wymiaru powoduje także usunięcie wszystkich odpowiadających mu wymiarów kostki z ich kostek. Czy na pewno chcesz usunąć wymiar *nazwa\_wymiaru* z *nazwa\_obiektu*?**

**Objaśnienie**

Komunikat potwierdzający usunięcie wymiaru.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10067: Plik o nazwie *nazwa\_pliku* już istnieje. Czy chcesz zastąpić jego zawartość?**

**Objaśnienie**

Komunikat potwierdzający zastąpienie pliku.

**Działanie użytkownika**

Upewnij się, że chcesz zastąpić zawartość wskazanego pliku.



**10068: Nie można ustalić typu danych dla obiektu o nazwie *nazwa\_obiektu* i schemacie *nazwa\_schematu*. Baza danych zwróciła następujące informacje: *komunikat*.**

**Objaśnienie**

Centrum OLAP nie jest w stanie ustalić źródłowego typu danych ani zagregowanego typu danych dla wskazanego obiektu.

**Działanie użytkownika**

Upewnij się, czy wyrażenie SQL wskazanego obiektu jest poprawne. W razie trudności z rozwiązaniem problemu skontaktuj się z działem wsparcia firmy IBM.

**10069: Nie można ustalić źródłowego typu danych dla miary o nazwie *nazwa\_miary* i schemacie *nazwa\_schematu*.**

**Objaśnienie**

Centrum OLAP nie może ustalić źródłowego typu danych dla wskazanej miary, ponieważ miara ta ma niepoprawne wyrażenie źródłowe. Niepoprawność wyrażenia źródłowego miary może wynikać z zastosowania dla niej ustawienia agregacji Brak, ponieważ miara podlega sprawdzeniu poprawności za pomocą agregacji miar, do których wyrażenie się odwołuje, a nie jako wyrażenie autonomiczne.

**Działanie użytkownika**

Wykonaj jedno z następujących działań:

- Zmodyfikuj wyrażenie źródłowe wskazanej miary tak, aby przy ustawieniu agregacji Brak dawało pomyślny wynik sprawdzenia poprawności.
- Nie używaj tej miary w wyrażeniu.

**10070: Skasowanie obiektu faktów spowoduje także skasowanie jego miar. Czy na pewno chcesz skasować obiekt faktów *nazwa\_obiektu\_faktów*?**

**Objaśnienie**

Komunikat potwierdzający zamiar skasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

**10071: Wszystkie zaznaczone obiekty zostaną skasowane z bazy danych. Czy chcesz skasować te obiekty?**

**Objaśnienie**

Zaznaczono więcej niż jeden obiekt i wybrano operację kasowania.

**Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać skasowany, i kliknij przycisk **Tak**. Jeśli wybrane obiekty nie powinny być kasowane, kliknij przycisk **Nie**.

### **10072: Niektóre z wybranych obiektów nie mogą zostać skasowane. Obiekty te pozostaną w bazie danych.**

#### **Objaśnienie**

Centrum OLAP nie może usunąć wszystkich zaznaczonych obiektów. Prawdopodobną przyczyną jest to, że niektóre z wybranych obiektów stanowią cel odwołań innych obiektów z bazy danych, a ich usunięcie sprawiłoby, że obiekty odwołujące się stałyby się niepoprawne.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

### **10073: Żaden z wybranych obiektów nie może zostać skasowany.**

#### **Objaśnienie**

Centrum OLAP nie może usunąć wszystkich wybranych obiektów. Prawdopodobnie przyczyną jest to, że do wybranych obiektów odwołują się inne obiekty z bazy danych, a usunięcie ich sprawiłoby, że obiekty odwołujące się stałyby się niepoprawne.

#### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

### **10074: Czy na pewno chcesz usunąć obiekt *nazwa\_objektu* z katalogu *nazwa\_katalogu*?**

#### **Objaśnienie**

Komunikat potwierdzający usunięcie.

#### **Działanie użytkownika**

Upewnij się, że wybrany obiekt jest tym, który ma zostać usunięty, i kliknij przycisk **Tak**. Jeśli wybrany obiekt nie powinien być kasowany, kliknij przycisk **Nie**.

### **10075: Wszystkie wybrane obiekty zostaną usunięte z katalogu *nazwa\_katalogu*. Czy chcesz usunąć te obiekty?**

#### **Objaśnienie**

Zaznaczono więcej niż jeden obiekt i wybrano operację usuwania.

#### **Działanie użytkownika**

Upewnij się, że zaznaczone obiekty są tymi, które mają zostać usunięte, i kliknij przycisk **Tak**. Jeśli zaznaczone obiekty nie powinny zostać usunięte, kliknij przycisk **Nie**.

### **10076: Niektórych z wybranych obiektów nie można usunąć. Obiekty te pozostaną w bazie danych.**

#### **Objaśnienie**

Centrum OLAP nie może usunąć wszystkich wybranych obiektów. Prawdopodobną przyczyną jest to, że do wybranych obiektów odwołują się inne obiekty z bazy danych, a usunięcie ich sprawiłoby, że obiekty odwołujące się stałyby się niepoprawne. Przyczyną może być również to, że obiekt nadrzędny musi mieć co najmniej jeden obiekt podrzędny (jest tak na przykład w przypadku hierarchii kostki).

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

#### **10077: Żadnego z wybranych obiektów nie można usunąć.**

##### Objaśnienie

Centrum OLAP nie może usunąć jednego lub kilku wybranych obiektów. Prawdopodobną przyczyną jest to, że do wybranych obiektów odwołują się inne obiekty z bazy danych, a usunięcie ich sprawiłoby, że obiekty odwołujące się stałyby się niepoprawne.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

#### **10078: Operacja zakończyła się pomyślnie. Z bazy danych zwrócone zostały następujące komunikaty informacyjne: nazwa\_bazy\_danych.**

##### Objaśnienie

Operacja bazy danych powiodła się, ale zostały zwrócone jakieś komunikaty informacyjne.

#### Działanie użytkownika

Przejrzyj te komunikaty informacyjne i zdecyduj, czy konieczne jest podjęcie jakichś dalszych działań.

#### **10080: W trakcie drugiego przebiegu analizy XML nie znaleziono obiektu typu typ.**

##### Objaśnienie

Nie znaleziono obiektu, do którego odwołuje się odczytywany kod XML.

#### Działanie użytkownika

Sprawdź poprawność strukturalną importowanego pliku XML. Jeśli błąd występuje w trakcie uruchamiania Centrum OLAP, skontaktuj się z działem wsparcia firmy IBM.

#### **10081: System nie mógł przeanalizować składni XML pliku nazwa\_pliku. Błąd wystąpił w wierszu numer\_wiersza na znaku numer\_znaku. Analizator składni zwrócił następujące informacje: informacja.**

##### Objaśnienie

Podczas próby zaimportowania pliku XML wystąpił błąd analizy składni.

#### Działanie użytkownika

Zapewnij, aby importowany plik XML był poprawnie utworzony i był prawidłowym plikiem metadanych XML programu Cube Views. Sprawdź wskazany wiersz i znak, aby znaleźć błąd.

**10082: Wystąpił nieoczekiwany wyjątek analizatora składni w pliku *nazwa\_pliku*. Zwrócone zostały następujące informacje: informacje.**

**Objaśnienie**

Podczas próby zaimportowania pliku XML wystąpił nieoczekiwany błąd analizy składni.

**Działanie użytkownika**

Zapewnij, aby importowany plik XML był poprawnie utworzony i był prawidłowym plikiem metadanych XML programu Cube Views. Sprawdź wskazany wiersz i znak, aby znaleźć błąd.

**10084: Już istnieje obiekt o nazwie *nazwa\_objektu* w schemacie *nazwa\_schematu*. Utworzenie obiektu nie jest możliwe. Wpisz unikalną nazwę i/lub schemat nowego obiektu.**

**Objaśnienie**

W Centrum OLAP podjęta została próba utworzenia nowego obiektu, lecz istnieje już obiekt tego samego typu o takiej samej nazwie i schemacie.

**Działanie użytkownika**

Wprowadź inną nazwę lub schemat dla tworzonego obiektu.

**10085: Już istnieje obiekt o nazwie *nazwa\_objektu* w schemacie *nazwa\_schematu*. Zmiana nazwy obiektu nie jest możliwa. Wpisz unikalną nazwę i/lub schemat obiektu, którego nazwa jest zmieniana.**

**Objaśnienie**

W Centrum OLAP podjęta została próba zmiany nazwy obiektu, lecz istnieje już obiekt tego samego typu o takiej samej nazwie i schemacie.

**Działanie użytkownika**

Wprowadź inną nazwę lub schemat dla obiektu, którego nazwa jest zmieniana.

**10086: Nawiązanie połączenia z bazą danych nie było możliwe. Dane zwrócone przez program DB2: *komunikat*.**

**Objaśnienie**

W Centrum OLAP nie udało się nawiązać połączenia z bazą danych. Pewne informacje o błędzie przekazane przez program DB2 są zawarte w komunikacie.

**Działanie użytkownika**

Zapoznaj się z informacjami przekazanymi przez program DB2 i wyeliminuj problem.

**10087: Obiekt metadanych *nazwa\_objektu1* nie może zostać skasowany, ponieważ odwołuje się do niego obiekt *nazwa\_objektu2* typu *typ*.**

**Objaśnienie**

Nie można skasować zaznaczonego obiektu metadanych, ponieważ jest on używany przez co najmniej jeden inny obiekt metadanych.

### Działanie użytkownika

Usuń odwołania do obiektu ze wszystkich innych obiektów metadanych, po czym podejmij ponowną próbę skasowania obiektu.

**10088: Wystąpił błąd podczas rejestrowania sterownika DB2 w Menedżerze sterowników JDBC. Nawiązanie połączenia z bazą danych nie było możliwe. Zwrócone zostały następujące informacje:** *komunikat*.

#### Objaśnienie

Przed nawiązaniem połączenia z bazą danych DB2 Centrum OLAP musi zarejestrować sterownik JDBC, który będzie używany przez Menedżera sterowników. Podczas rejestracji sterownika JDBC wystąpił błąd.

### Działanie użytkownika

Sprawdź poprawność instalacji DB2, aby upewnić się, że pliki db2java.zip i db2jcc.jar zostały zainstalowane. Upewnij się co do prawidłowości instalacji środowiska Java i komponentów JDBC. Zapoznaj się z informacjami zawartymi w komunikacie, aby wyeliminować problem.

**10089: Wystąpił błąd podczas dostępu do bazy danych. Baza danych zwróciła następujące informacje:** `\n Stan SQL: komunikat\n Kod błędu SQL: kod\n Komunikat SQL: komunikat_SQL`.

#### Objaśnienie

Aplikacja Centrum OLAP wywołała program DB2 za pośrednictwem funkcji API zapisanej w bazie. Komenda uruchomienia wywołała wyjątek SQLException, który nie został obsłużony w Centrum OLAP.

### Działanie użytkownika

Posłuż się dodatkowymi informacjami w komunikacie, aby wyeliminować problem. W razie trudności z rozwiązaniem problemu skontaktuj się z działem wsparcia firmy IBM.

**10090: Wykonanie procedury zapisanej w bazie DB2 spowodowało przekazanie fałszywego kodu powrotu. W zwróconym dokumencie XML nie odnaleziono żadnych informacji o błędzie. Skontaktuj się z działem wsparcia firmy IBM.**

#### Objaśnienie

Aplikacja Centrum OLAP wywołała program DB2 za pośrednictwem funkcji API zapisanej w bazie. Komenda wykonania zwróciła wartość *fałsz*, lecz w dokumencie XML zwróconym przez procedurę zapisaną w bazie nie było żadnych informacji o błędach.

### Działanie użytkownika

Nie jest wykluczone, że operacja została zakończona pomyślnie, jednak wskazane jest zgłoszenie problemu obsłudze technicznej firmy IBM.

**10091: Wystąpił błąd podczas przetwarzania wywołania funkcji API bazy danych. Zwrócone zostały następujące informacje: \n Stan SQL: komunikat\n Kod błędu SQL: kod\n Operacja: operacja\n Identyfikator statusu: identyfikator\n Tekst statusu: tekst.**

**Objaśnienie**

Wywołanie procedury zapisanej w bazie korzystającej z API w Centrum OLAP spowodowało błąd przy realizacji pewnych zmian w Centrum OLAP.

**Działanie użytkownika**

Zapoznaj się z informacjami zawartymi w komunikacie. W razie trudności z rozwiązaniem problemu skontaktuj się z działem wsparcia firmy IBM.

**10092: Wystąpił błąd podczas analizy składniowej dokumentu XML zwróconego w wyniku wywołania funkcji API bazy danych. Zwrócone zostały następujące informacje: komunikat.**

**Objaśnienie**

W wyniku wywołania przez Centrum OLAP procedury API zapisanej w bazie zwrócony został niepełny lub nieprawidłowo zbudowany kod XML. Odczytanie zwróconego kodu XML w Centrum OLAP nie było możliwe.

**Działanie użytkownika**

Użyj informacji zawartych w komunikacie, aby rozwiązać problem. W razie trudności z rozwiązaniem problemu skontaktuj się z działem wsparcia firmy IBM.

**10093: Plik *nazwa\_pliku* nie istnieje.**

**Objaśnienie**

Wskazany plik nie istnieje.

**Działanie użytkownika**

Podaj nazwę istniejącego pliku.

**10094: Podczas odczytu pliku *nazwa\_pliku* wystąpił błąd we/wy. Zwrócone zostały następujące informacje systemowe: komunikat.**

**Objaśnienie**

Podczas odczytu pliku wystąpił błąd we/wy.

**Działanie użytkownika**

Zapoznaj się z informacjami zwróconymi przez system, aby spróbować rozwiązać problem, lub wskaż inny plik.

**10095: Podczas zapisu pliku *nazwa\_pliku* wystąpił błąd we/wy. Zwrócone zostały następujące informacje systemowe: komunikat.**

**Objaśnienie**

Podczas operacji zapisu w pliku wystąpił błąd we/wy.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez system, aby spróbować rozwiązać problem, lub wskaż inny plik.

**10096: Przetwarzanie zapytania pobierającego schemat bazy danych nie powiodło się. Baza danych zwróciła następujące informacje:** *komunikat.*

#### Objaśnienie

Przetwarzanie zapytania pobierającego schemat bazy danych nie powiodło się.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez bazę danych, aby rozwiązać problem.

**10097: Przetwarzanie zapytania pobierającego tabele schematu nie powiodło się. Baza danych zwróciła następujące informacje:** *komunikat.*

#### Objaśnienie

Przetwarzanie zapytania pobierającego tabele schematu nie powiodło się.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez bazę danych, aby rozwiązać problem.

**10098: Przetwarzanie zapytania pobierającego kolumny tabeli nie powiodło się. Baza danych zwróciła następujące informacje:** *komunikat.*

#### Objaśnienie

Przetwarzanie zapytania pobierającego kolumny tabeli nie powiodło się.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez bazę danych, aby rozwiązać problem.

**10099: Zatwierdzenie połączenia z bazą DB2 nie powiodło się. Baza danych zwróciła następujące informacje:** *komunikat.*

#### Objaśnienie

Zatwierdzenie połączenia z bazą DB2 nie powiodło się.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez bazę danych, aby rozwiązać problem.

**10100: Wycofanie zmian w ramach połączenia z bazą DB2 nie powiodło się. Baza danych zwróciła następujące informacje:** *komunikat.*

#### Objaśnienie

Wycofanie zmian w ramach połączenia z bazą DB2 nie powiodło się.

#### Działanie użytkownika

Zapoznaj się z informacjami zwróconymi przez bazę danych, aby rozwiązać problem.

**10101: Nie można skasować wymiaru *nazwa\_objektu*, ponieważ jest to ostatni wymiar kostki w kostce *nazwa\_kostki*. Poprawna kostka musi zawierać przynajmniej jeden wymiar kostki.**

#### Objaśnienie

W Centrum OLAP podjęto próbę usunięcia ostatniego wymiaru kostki w kostce.

#### Działanie użytkownika

Poprawna kostka musi zawierać przynajmniej jeden wymiar kostki. Nie należy kasować ostatniego wymiaru kostki w kostce.

**10102: Obiekt *nazwa\_objektu1* typu *typ1* odwołuje się do obiektu *nazwa\_objektu2* lub typu *typ2*, którego nie można odnaleźć.**

#### Objaśnienie

Obiekt w odczytywanym pliku XML odwołuje się do obiektu, którego odnalezienie jest niemożliwe. Jeśli błąd występuje podczas importu, obiekt będący celem odwołania może być nieobecny w importowanym pliku.

#### Działanie użytkownika

Jeśli błąd wystąpił podczas importu, upewnij się, że plik zawiera wszystkie obiekty wymagane do powodzenia importu. Jeśli błąd ten występuje w trakcie uruchamiania Centrum OLAP, skontaktuj się z działem wsparcia firmy IBM.

**10103: Nie można skasować miary, ponieważ obiekt faktów musi zawierać przynajmniej jedną miarę.**

#### Objaśnienie

Nie można skasować miary, ponieważ obiekt faktów musi zawierać przynajmniej jedną miarę.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10104: Nie można usunąć miary, ponieważ obiekt faktów kostki musi zawierać przynajmniej jedną miarę.**

#### Objaśnienie

Nie można usunąć miary, ponieważ obiekt faktów kostki musi zawierać przynajmniej jedną miarę.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.



**10105: Nie można usunąć poziomu kostki, ponieważ hierarchia kostki musi zawierać przynajmniej jeden poziom kostki.**

**Objaśnienie**

Nie można usunąć poziomu kostki, ponieważ obiekt hierarchii kostki musi zawierać przynajmniej jeden poziom kostki.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10106: Nie można usunąć poziomu, ponieważ odwołuje się do niego poziom kostki w hierarchii kostki *nazwa\_hierarchii*, która jest skojarzona z hierarchią *nazwa\_hierarchii*.**

**Objaśnienie**

Nie można usunąć poziomu, ponieważ odwołuje się do niego poziom kostki w hierarchii kostki skojarzony z hierarchią, z której usuwany jest atrybut.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10107: Nie można usunąć atrybutu, ponieważ odwołuje się do niego poziom kostki *nazwa\_poziomu*, skojarzony z poziomem *nazwa\_poziomu*.**

**Objaśnienie**

Nie można usunąć atrybutu, ponieważ odwołuje się do niego poziom kostki z hierarchii skojarzony z poziomem, z którego usuwany jest atrybut.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10108: Nie można usunąć atrybutu, ponieważ odwołuje się do niego hierarchia *nazwa\_hierarchii*, skojarzona z wymiarem *nazwa\_wymiaru*.**

**Objaśnienie**

Nie można usunąć atrybutu, ponieważ odwołuje się do niego hierarchia skojarzona z wymiarem, z którego usuwany jest atrybut.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10109: Nie można usunąć atrybutu, ponieważ odwołuje się do niego łączenie *nazwa\_łączenia*, skojarzone z wymiarem *nazwa\_wymiaru*.**

**Objaśnienie**

Nie można usunąć atrybutu, ponieważ odwołuje się do niego łączenie skojarzone z wymiarem, z którego usuwany jest atrybut.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10110: Nie można przenieść poziomu, ponieważ odwołuje się do niego poziom kostki w hierarchii kostki *nazwa\_hierarchii*, która jest skojarzona z hierarchią *nazwa\_hierarchii*.**

#### Objaśnienie

Nie można przenieść poziomu, ponieważ odwołuje się do niego jakiś poziom kostki z hierarchii kostki, który jest skojarzony z edytowaną hierarchią.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10111: Nie można usunąć atrybutu, ponieważ jest on jedynym atrybutem klucza poziomu *nazwa\_poziomu*.**

#### Objaśnienie

Nie można usunąć atrybutu, ponieważ jest to jedyny atrybut klucza poziomu, z którego jest on usuwany, a poziom ten musi mieć co najmniej jeden atrybut klucza poziomu.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10112: Nie można usunąć atrybutu, ponieważ jest on atrybutem domyślnym poziomu *nazwa\_poziomu*.**

#### Objaśnienie

Nie można usunąć atrybutu, ponieważ jest on atrybutem domyślnym poziomu, z którego usuwany jest atrybut.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10113: Nie można usunąć atrybutu, ponieważ odwołuje się do niego poziom *nazwa\_poziomu*, skojarzony z wymiarem *nazwa\_wymiaru*.**

#### Objaśnienie

Nie można usunąć atrybutu, ponieważ odwołuje się do niego poziom skojarzony z wymiarem, z którego usuwany jest atrybut.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10114: Nie można usunąć poziomu, ponieważ jest on jedynym poziomem w hierarchii *nazwa\_hierarchii*.**

#### Objaśnienie

Nie można usunąć poziomu, ponieważ jest to jedyny poziom w hierarchii, a hierarchia ta musi zawierać co najmniej jeden poziom.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

#### **10200: Importowany plik nie jest kodowany w standardzie UTF-8. Wybierz plik kodowany w standardzie UTF-8.**

##### Objaśnienie

Centrum OLAP może importować tylko pliki kodowane w standardzie UTF-8.

#### Działanie użytkownika

Zaimportuj plik z obsługiwanym typem kodowania.

#### **10201: Wprowadź nazwę pliku ze skrypcem SQL używanym do odświeżania tabel podsumowań.**

##### Objaśnienie

Doradca optymalizacyjny tworzy skrypt SQL służący do odświeżania tabel podsumowań, gdy wybrana jest opcja aktualizacji odroczonej. Skrypt ten powinien zostać zapisany w pliku i uruchamiany w celu odświeżania tabel podsumowań.

#### Działanie użytkownika

Wprowadź nazwę pliku, w którym ma zostać zapisany skrypt SQL.

#### **10202: Wprowadź nazwę pliku ze skrypcem SQL używanym do tworzenia tabel podsumowań.**

##### Objaśnienie

Doradca optymalizacyjny generuje skrypt SQL tworzący tabele podsumowań. Skrypt ten powinien zostać zapisany w pliku i uruchamiany w celu utworzenia tabel podsumowań.

#### Działanie użytkownika

Wprowadź nazwę pliku, w którym ma zostać zapisany skrypt SQL.

#### **10203: Dla wybranej miary nie można określić braku agregacji. Takie ustawienie agregacji mogą mieć wyłącznie miary obliczane, które odwołują się tylko do innych miar obecnych w ich wyrażeniu.**

##### Objaśnienie

Ustawienie agregacji Brak może być wybrane jedynie w przypadku miar, które korzystają wyłącznie z wyrażeń odwołujących się tylko do innych miar.

#### Działanie użytkownika

Wybierz inne ustawienie agregacji.

#### **10204: Nie istnieją żadne wymiary. Utwórz nowy wymiar, aby go dodać do modelu kostki.**

##### Objaśnienie

Nie istnieją żadne wymiary. Utwórz nowy wymiar, aby go dodać do modelu kostki.

#### Działanie użytkownika

Zamiast dodawać wymiar, utwórz nowy.

**10205: Nie można dodać żadnych wymiarów, ponieważ wszystkie istniejące wymiary zostały już uwzględnione w modelu kostki.**

#### Objaśnienie

Wszystkie istniejące wymiary zostały już dodane do modelu kostki.

#### Działanie użytkownika

Nie jest wymagane żadne działanie.

**10206: Zmieniono wybór opcji. Aby wyświetlić nowe zalecenia Doradcy optymalizacyjnego, należy ponownie uruchomić proces Doradcy. Bez ponownego uruchomienia Doradcy optymalizacyjnego wyświetlane zalecenia dotyczyć będą poprzedniego zestawu opcji. Czy chcesz ponownie uruchomić proces Doradcy optymalizacyjnego?**

#### Objaśnienie

Zmieniono dobrane opcje po wykonaniu procesu Doradcy optymalizacyjnego. Aby wyświetlić zaktualizowane zalecenia dotyczące tabel podsumowań, uruchom ponownie proces Doradcy optymalizacyjnego. Bez ponownego uruchomienia Doradcy optymalizacyjnego wyświetlane zalecenia dotyczyć będą poprzedniego zestawu opcji.

#### Działanie użytkownika

Kliknij przycisk **Tak**, aby ponownie uruchomić proces Doradcy optymalizacyjnego. Kliknij przycisk **Nie**, jeśli nie chcesz ponownie uruchamiać Doradcy optymalizacyjnego.

**10207: Nie wykryto żadnej tabeli wymiarów.**

#### Objaśnienie

Nie wykryto żadnej tabeli wymiarów.

#### Działanie użytkownika

Upewnij się co do prawidłowego ustawienia ograniczeń spójności referencyjnej.

**10208: W bazie danych znajdują się obiekty *nazwa\_objektu*, których Centrum OLAP nie może wyświetlić bezpośrednio. Obiekty te mogą być w przyszłości przyczyną problemów w działaniu Centrum OLAP. Kliknij przycisk **Tak**, aby usunąć te obiekty, lub przycisk **Nie**, aby je zachować w bazie danych.**

#### Objaśnienie

W Centrum OLAP wykryta została pewna liczba obiektów (na przykład hierarchii lub obiektów faktów) w bazie danych, które nie mogą być poprawnie wyświetlone. Mogą to być już istniejące obiekty lub obiekty utworzone po imporcie metadanych. Obiekty te mogą powodować w przyszłości konflikty nazw i problemy z odwołaniami w Centrum OLAP. Jeśli obiekty te nie są absolutnie niezbędne, zaleca się ich usunięcie.

#### Działanie użytkownika

Kliknij przycisk **Tak**, aby usunąć te obiekty, lub przycisk **Nie**, aby je zachować w bazie danych.

#### **10209: Podczas importu wystąpił nieoczekiwany błąd. Sprawdź poprawność wejściowego pliku XML.**

##### Objaśnienie

Podczas importu interfejs API procedury zapisanej w bazie zwrócił ostrzeżenie, którego wynikowe dane XML nie zawierały żadnej treści.

#### Działanie użytkownika

Sprawdź, czy wejściowe metadane XML mają format zgodny z formatem zdefiniowanym w schemacie metadanych OLAP, oraz czy plik XML zawiera definicje wszystkich obiektów metadanych, do których się odwołuje.

#### **10210: Operacja importu nie powiodła się. Funkcja API zapisana w bazie zwróciła następujący komunikat: *komunikat*.**

##### Objaśnienie

Podczas procesu importu funkcja API zapisana w bazie zwróciła komunikat o błędzie.

#### Działanie użytkownika

Wyeliminuj problem, korzystając z informacji zawartych w komunikacie. W razie trudności z rozwiązaniem problemu skontaktuj się z działem wsparcia firmy IBM.

#### **10211: Miara nieliczbowa *nazwa\_miary* nie może korzystać z funkcji agregującej *nazwa\_funkcji*, ponieważ funkcja ta wymaga parametru liczbowego.**

##### Objaśnienie

Miary o typach danych innych niż liczbowe nie mogą mieć liczbowych funkcji agregujących. W przypadku danych nieliczbowych jedyne dostępne funkcje agregujące, to **MIN**, **MAX** i **COUNT**.

#### Działanie użytkownika

Wybierz inną funkcję agregującą.

#### **10212: Nie można odczytać obiektów z wejściowego pliku XML. Sprawdź poprawność wejściowego pliku XML.**

##### Objaśnienie

Centrum OLAP nie zdołało odczytać obiektów z wejściowego pliku XML.

#### Działanie użytkownika

Sprawdź, czy wejściowe metadane XML mają format zgodny z formatem zdefiniowanym w schemacie metadanych OLAP, oraz czy plik XML zawiera definicje wszystkich obiektów metadanych, do których się odwołuje.

**10213: Wejściowy plik XML *nazwa\_pliku* nie istnieje we wskazanym katalogu.**

**Objaśnienie**

Wejściowy plik XML nie istnieje we wskazanym katalogu.

**Działanie użytkownika**

Upewnij się, że wejściowy plik XML istnieje we wskazanym katalogu.

**10214: Obiekt *nazwa\_obiektu* zawarty w pliku importu odwołuje się do kolumny *nazwa\_kolumny*, która nie istnieje w bazie danych. Przed przystąpieniem do importu pliku należy się upewnić, że istnieją tabele i kolumny, do których odwołują się obiekty metadanych w tym pliku.**

**Objaśnienie**

Importowany plik XML zawiera obiekty odwołujące się do tabel i kolumn, które nie istnieją w bazie danych.

**Działanie użytkownika**

Przed przystąpieniem do importu pliku należy się upewnić, że istnieją tabele, do których odwołują się obiekty w tym pliku.

**10215: Centrum OLAP nie może uruchomić skryptu SQL zalecanego przez Doradcę optymalizacyjnego. Baza danych zwróciła następujące informacje: *komunikat*.**

**Objaśnienie**

Centrum OLAP nie może wykonać skryptu SQL zalecanego przez Doradcę optymalizacyjnego. Być może nie masz uprawnień wymaganych do wykonania tego skryptu SQL.

**Działanie użytkownika**

Uzyskaj uprawnienia niezbędne do wykonania zaleceń Doradcy optymalizacyjnego. Uprawnienia te są opisane w temacie "Uprawnienia" w pomocy elektronicznej Centrum OLAP. Więcej informacji na temat optymalizowania modelu kostki zawiera podręcznik *DB2 Cube Views Podręcznik instalacji i użytkowania*.

**10216: Zalecenia Doradcy optymalizacyjnego zostały pomyślnie zapisane we wskazanym pliku (plikach).**

**Objaśnienie**

Zgodny z zaleceniami skrypt SQL tworzący tabele podsumowań i ewentualny skrypt SQL odświeżający tabele podsumowań zostały zapisane we wskazanych plikach.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

## **10217: Pomyślnie utworzono tabele podsumowań i ich indeksy.**

### **Objaśnienie**

Tabele podsumowań i indeksy zalecane przez Doradcę optymalizacyjnego zostały pomyślnie utworzone w bazie danych.

### **Działanie użytkownika**

Nie jest wymagane żadne działanie.

## **10218: Wybrany został widok. Doradca optymalizacyjny nie może sprawdzić, czy dla tabel, do których odwołuje się wybrany widok, istnieją ograniczenia referencyjne.**

### **Objaśnienie**

Optymalizacja może być mało skuteczna, jeśli dla modeli kostek tworzone są tabele podsumowań korzystające z widoków odwołujących się do tabel bez zdefiniowanych ograniczeń. Doradca optymalizacyjny nie ma możliwości stwierdzenia, czy w tabeli podstawowej widoku są zdefiniowane ograniczenia.

### **Działanie użytkownika**

Jeśli tabele, do których odwołuje się widok, są pozbawione ograniczeń, a uruchomienie Doradcy optymalizacyjnego jest wymagane, istnieją dwa wyjścia: 1. Zrezygnować z tego widoku w modelu kostki. 2. Utworzyć ograniczenia dla tabel przed uruchomieniem Doradcy optymalizacyjnego.

## **10219: Anulowanie działania Doradcy optymalizacyjnego spowoduje odrzucenie zalecanych skryptów SQL bez zapisywania. Kliknij przycisk Tak, aby zamknąć Doradcę optymalizacyjnego bez zapisywania zalecanych skryptów SQL. Kliknij przycisk Nie, aby powrócić do Doradcy optymalizacyjnego, co umożliwi zapisanie zalecanych skryptów SQL.**

### **Objaśnienie**

Kliknięcie przycisku **Anuluj** w kreatorze Doradcy optymalizacyjnego powoduje odrzucenie zaleconych skryptów SQL bez zapisania ich.

### **Działanie użytkownika**

Kliknij przycisk **Tak**, aby zamknąć kreatora Doradcy optymalizacyjnego, nie zapisując zaleconych skryptów SQL, albo kliknij przycisk **Nie**, aby powrócić do kreatora, kontynuować pracę i zapisać skrypty SQL.

## **10220: Wskazany plik jest zapisany w formacie starszej wersji programu DB2 Cube Views. Kreator importu może przekształcić obiekty metadanych opisane we wskazanym pliku do formatu aktualnej wersji programu DB2 Cube Views. Kliknij przycisk Tak, aby przekształcić obiekty metadanych opisane we wskazanym pliku i kontynuować import. Kliknij przycisk Nie, aby zatrzymać import i wskazać inny plik lub zamknąć kreatora importu.**

### **Objaśnienie**

Określony plik źródłowy metadanych zawiera kod XML w formacie starszej wersji programu DB2 Cube Views. Kreator importu może odczytać plik, a następnie przekształcić jego zawartość do formatu, który jest zrozumiały dla bieżącej wersji programu DB2 Cube Views.

### Działanie użytkownika

Kliknij przycisk **Tak**, aby przekształcić obiekty metadanych opisane przez wskazany plik źródłowy do bieżącego formatu programu DB2 Cube Views. Na stronie Opcje importu dostępne są obiekty z określonego pliku przekształconego do nowej wersji programu DB2 Cube Views. Kliknij przycisk **Nie**, aby powstrzymać Kreatora importu od przekształcania obiektów metadanych opisanych przez określony plik. Możesz określić inny plik źródłowy metadanych albo zamknąć Kreatora importu.

### **10221: Zalecane tabele podsumowań powinny zgodnie z oczekiwaniem zająć miejsce na dysku w ilości wielkość\_miejsca\_na\_dysku.**

#### Objaśnienie

Tabele podsumowań zalecone przez Doradcę optymalizacyjnego powinny zajmować określoną wielkość miejsca na dysku.

### Działanie użytkownika

Zapewnij co najmniej wskazaną ilość wolnego miejsca na dysku, zanim uruchomisz zalecone skrypty.

### **10222: Czy chcesz zatrzymać Doradcę optymalizacyjnego? Kliknij przycisk Tak, aby przerwać generowanie zaleceń przez Doradcę optymalizacyjnego i powrócić do zaleceń wygenerowanych dotychczas. Kliknij przycisk Nie, aby umożliwić Doradcy optymalizacyjnemu dalsze generowanie zaleceń.**

#### Objaśnienie

Gdy Doradca optymalizacyjny tworzył zalecenia optymalizacyjne, został kliknięty przycisk **Zatrzymaj**.

### Działanie użytkownika

Kliknij przycisk **Tak**, aby powstrzymać Doradcę optymalizacyjnego od tworzenia zaleceń i powrócić do zaleceń uzyskanych dotychczas. Kliknij przycisk **Nie**, aby umożliwić Doradcy optymalizacyjnemu dalsze generowanie zaleceń. Kliknięcie przycisku **Tak** spowoduje wyświetlenie w kreatorze zaleceń uzyskanych dotychczas.

### **10300: Nie powiodła się analiza składniowa miary nazwa\_miary wprowadzonej w wyrażeniu SQL.**

#### Objaśnienie

Podane wyrażenie SQL odwołuje się do miary, która jest niepoprawna w danym kontekście lub nieobecna w bazie danych.

### Działanie użytkownika

Upewnij się, że wyrażenie SQL odwołuje się wyłącznie do miar występujących na liście Dane w Kreatorze wyrażen SQL.



### **10301: Nie powiodła się analiza składniowa kolumny *nazwa\_kolumny* wprowadzonej w wyrażeniu SQL.**

#### **Objaśnienie**

Podane wyrażenie SQL odwołuje się do kolumny, która jest niepoprawna w danym kontekście lub nieobecna w bazie danych.

#### **Działanie użytkownika**

Upewnij się, że wyrażenie SQL odwołuje się wyłącznie do kolumn występujących na liście Dane w Kreatorze wyrażen SQL.

### **10302: Atrybut *nazwa\_atrybutu* nie ma kwalifikatora w postaci nazwy schematu.**

#### **Objaśnienie**

Odwołania do atrybutów w wyrażeniu SQL muszą mieć kwalifikator w postaci nazwy schematu oddzielonej kropką ('.').

#### **Działanie użytkownika**

Zadbaj, aby wszystkie odwołania do atrybutów w wyrażeniu SQL miały kwalifikator w postaci nazwy schematu oddzielonej kropką ('.').

### **10303: Miara *nazwa\_miary* nie ma kwalifikatora w postaci nazwy schematu.**

#### **Objaśnienie**

Odwołania do miar w wyrażeniu SQL muszą mieć kwalifikator w postaci nazwy schematu oddzielonej kropką ('.').

#### **Działanie użytkownika**

Zadbaj, aby wszystkie odwołania do miar w wyrażeniu SQL miały kwalifikator w postaci nazwy schematu oddzielonej kropką ('.').

### **10304: Brak nazwy obiektu wewnątrz znacznika obiektu *nazwa\_obiektu*.**

#### **Objaśnienie**

Wskazane wyrażenie SQL ma pusty znacznik kolumny @Column, pusty znacznik atrybutu @Attribute lub pusty znacznik miary @Measure.

#### **Działanie użytkownika**

Zadbaj o to, by znaczniki typu obiektów @Column, @Measure i @Attribute zawierały w nawiasie nazwy obiektów.

### **10305: Podane wyrażenie nie jest poprawne. Baza danych zwróciła następujące informacje: *komunikat*.**

#### **Objaśnienie**

Wyrażenie SQL zawiera błąd składniowy. Błąd ten jest sygnalizowany także w sytuacji, gdy wyrażenie SQL odwołuje się do kolumn, atrybutów lub miar nieujętych w znaczniki typu. Odwołania do kolumn, atrybutów i miar muszą być umieszczone odpowiednio w znacznikach @Column(), @Attribute() lub @Measure().

#### Działanie użytkownika

Popraw błąd składniowy. Zadbaj o to, by każde odwołanie do kolumny, atrybutu i miary było ujęte w odpowiednim znaczniku.

**10306: Typ danych wprowadzonego wyrażenia nie jest typem liczbowym. Wprowadź wyrażenie typu liczbowego jako drugi parametr.**

#### Objaśnienie

Drugi parametr musi należeć do liczbowego typu danych.

#### Działanie użytkownika

Upewnij się, że wprowadzone wyrażenie daje wynik typu liczbowego.

**10307: Wyrażenie miary *nazwa\_miary* daje w wyniku wartość typu innego niż liczbowy. Wybierz miarę, której wyrażenie daje w wyniku wartość typu liczbowego.**

#### Objaśnienie

Drugi parametr musi należeć do liczbowego typu danych.

#### Działanie użytkownika

Upewnij się, że wybrane wyrażenie miary ma liczbowy typ danych.

**10308: Centrum OLAP nie może nawiązać połączenia ze wskazaną bazą danych. Powodem może być niewłaściwe skonfigurowanie bazy danych do współpracy z programem DB2 Cube Views. Konfigurowanie bazy danych może być czasochłonne. Kliknij przycisk Tak, aby skonfigurować wskazaną bazę danych. Kliknij przycisk Nie, jeśli nie chcesz teraz skonfigurować tej bazy danych.**

#### Objaśnienie

Za pomocą podanego identyfikatora i hasła użytkownika Centrum OLAP może nawiązać połączenie z bazą danych, jednak nie może skomunikować się z funkcją API zapisaną w bazie.

Przyczyny mogą być następujące:

- Funkcja API zapisana w bazie programu DB2 Cube Views nie jest zarejestrowana dla określonej bazy danych.
- Nie istnieją tabele katalogu programu DB2 Cube Views dla określonej bazy danych.

#### Działanie użytkownika

Aby skonfigurować bazę danych pod kątem programu DB2 Cube Views, kliknij przycisk **Tak**. W przeciwnym razie kliknij przycisk **Nie**.

**10309: Centrum OLAP nie może nawiązać połączenia ze wskazaną bazą danych, ponieważ baza danych ma konfigurację odpowiednią dla starszej wersji programu DB2 Cube Views. Należy przeprowadzić migrację bazy danych do aktualnej wersji programu DB2 Cube Views. Kliknij przycisk Tak, aby wykonać migrację wskazanej bazy danych w Centrum OLAP. Kliknij przycisk Nie, jeśli Centrum OLAP nie powinno wykonywać migracji wskazanej bazy danych.**

**Objaśnienie**

Centrum OLAP może połączyć się z bazą danych przy użyciu podanej nazwy użytkownika i hasła, ale nie może pobrać metadanych z katalogu. Może to być spowodowane tym, że tabele katalogu programu DB2 Cube Views są skonfigurowane dla starszej wersji programu DB2 Cube Views.

**Działanie użytkownika**

Kliknij przycisk **Tak**, jeśli chcesz przeprowadzić migrację katalogu programu DB2 Cube Views do właściwej wersji; w przeciwnym razie kliknij przycisk **Nie**.

**10310: Baza danych *nazwa\_bazy\_danych* została pomyślnie skonfigurowana.**

**Objaśnienie**

W Centrum OLAP pomyślnie utworzono tabele katalogu DB2 Cube Views i zarejestrowano funkcje API zapisane w bazie dla określonej bazy danych.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10311: Centrum OLAP nie może skonfigurować bazy danych na potrzeby programu DB2 Cube Views. Baza danych zwróciła następujące informacje: *informacje*.**

**Objaśnienie**

Centrum OLAP nie może skonfigurować określonej bazy danych pod kątem programu DB2 Cube Views.

Przyczyny mogą być następujące:

- Centrum OLAP nie może zarejestrować funkcji API zapisanych w bazie programu DB2 Cube Views.
- Centrum OLAP nie może utworzyć co najmniej jednej tabeli katalogu programu DB2 Cube Views.

**Działanie użytkownika**

Upewnij się, że dysponujesz odpowiednimi uprawnieniami do konfiguracji i instalacji, zgodnie z treścią tematu "Uprawnienia" w pomocy ekranowej Centrum OLAP. Więcej informacji na temat konfigurowania bazy danych zawiera podręcznik DB2 Cube Views Podręcznik instalacji i użytkownika.

**10312: Sprawdzenie poprawności agregacji zakończyło się niepowodzeniem. Przynajmniej jedna z podanych funkcji agregujących nie jest zgodna ze źródłowym wyrażeniem SQL.**

**Objaśnienie**

Przynajmniej jedna z podanych funkcji agregujących nie jest zgodna ze źródłowym wyrażeniem SQL. Powodem może być to, że wskazana funkcja agregująca oczekuje parametru o innym typie danych niż typ źródłowego wyrażenia SQL.

**Działanie użytkownika**

Upewnij się, że funkcja agregująca jest poprawna dla źródłowego typu danych określonej miary.

**10313: Wyrażenie źródłowe miary jest poprawne składniowo tylko przy ustawieniu opcji agregacji na Brak. Opcja agregacji dla tej miary musi mieć ustawienie Brak.**

**Objaśnienie**

Miara wymaga ustawienia agregacji Brak w następujących sytuacjach:

- Wyrażenie SQL jest składniowo niepoprawne, gdy funkcje agregujące nie są stosowane względem miar będących celem odwołań, a jest składniowo poprawne, gdy funkcje te są stosowane. Na przykład `char + int` jest wyrażeniem składniowo niepoprawnym, natomiast `COUNT(char) + SUM(int)` jest wyrażeniem poprawnym.
- Wyrażenie SQL korzysta z funkcji OLAP takich jak `RANK()`, `DENSE_RANK()` i `ROW_NUMBER()`.

**Działanie użytkownika**

Upewnij się, że do miary zastosowane zostało ustawienie agregacji Brak.

**10401: Wyrażenie nie może zawierać funkcji kolumnowej, selekcji skalarnej ani podzapytania.**

**Objaśnienie**

Wyrażenie SQL nie może zawierać funkcji kolumnowej, pełnej selekcji skalarnej ani podzapytania.

**Działanie użytkownika**

Skoryguj użycie funkcji kolumnowej, aby wyeliminować niepoprawne wyrażenie.

**10501: Nazwa schematu nie może zaczynać się od przedrostek.**

**Objaśnienie**

Nazwa schematu nie może zaczynać się przedrostkiem 'SYS' ani 'SESSION'.

**Działanie użytkownika**

Wpisz inną nazwę schematu.

**10502: Właściwości łączenia nie są poprawne na potrzeby optymalizacji wydajności modelu kostki. Wyeliminuj ten problem i uruchom ponownie Doradcę optymalizacyjnego. Baza danych zwróciła następujące informacje: *komunikat*.**

**Objaśnienie**

Właściwości łączenia uniemożliwiają optymalizację wydajności modelu kostki.

**Działanie użytkownika**

Określ poprawne ustawienia dla łączenia, stosując reguły sprawdzenia poprawności optymalizacji.

**10503: Nie można zmodyfikować hierarchii, ponieważ jest z nią skojarzona hierarchia kostki.**

**Objaśnienie**

Jeśli dla danej hierarchii istnieje hierarchia kostki, zmodyfikowanie jej nie jest możliwe.

**Działanie użytkownika**

Przed wprowadzeniem zmian w hierarchii upewnij się, że nie odwołuje się do niej żadna hierarchia kostki. Można też utworzyć inną hierarchię uwzględniającą wymagane modyfikacje.

**10504: Opcja agregacji tej miary musi mieć ustawienie Brak, ponieważ odwołuje się ona do miary, która korzysta z wieloparametrowej funkcji agregującej.**

**Objaśnienie**

Do miar korzystających z funkcji wieloparametrowych mogą odwoływać się wyłącznie miary z ustawieniem agregacji Brak. Nie można zmienić ustawienia agregacji z Brak na inną funkcję.

**Działanie użytkownika**

Wykonaj jedno z następujących działań:

- Nie modyfikuj ustawienia agregacji funkcji.
- Zmodyfikuj wskazaną miarę tak, aby nie korzystała z funkcji wieloparametrowej.

**10505: Ta miara nie może korzystać z funkcji wieloparametrowej, ponieważ do tej miary odwołuje się miara *nazwa\_miary*, dla której opcja agregacji ma ustawienie inne niż Brak.**

**Objaśnienie**

Do miar korzystających z funkcji wieloparametrowych mogą odwoływać się wyłącznie miary z ustawieniem agregacji Brak. Do skryptu agregacji modyfikowanej miary nie można wprowadzić funkcji wieloparametrowej, ponieważ modyfikowana miara stanowi cel odwołania miary mającej ustawienie agregacji inne niż Brak.

**Działanie użytkownika**

Wykonaj jedno z następujących działań:

- Nie modyfikuj skryptu agregacji funkcji.
- Zmień wskazaną miarę tak, aby nie odwoływała się już do miary modyfikowanej.

**10506: Aktualne ustawienie agregacji nie jest zgodne z podanym wyrażeniem SQL. Centrum OLAP przywróci ustawienie agregacji do wartości *ustawienie*.**

**Objaśnienie**

Obecne ustawienie agregacji jest niepoprawne w kontekście nowego wyrażenia SQL, toteż w jego miejsce zostało wprowadzone domyślne ustawienie agregacji. Przyczyny mogą być następujące:

- Zmienił się typ danych źródłowego wyrażenia SQL.
- Aktualnie oczekiwanym ustawieniem agregacji jest Brak. Ustawienie Brak jest wymagane, gdy:
  - Wyrażenie SQL jest składniowo niepoprawne, gdy funkcje agregujące nie są stosowane względem miar będących celem odwołań, a jest składniowo poprawne, gdy funkcje te są stosowane. Na przykład `char + int` jest wyrażeniem składniowo niepoprawnym, natomiast `COUNT(char) + SUM(int)` jest wyrażeniem poprawnym.
  - Wyrażenie SQL korzysta z funkcji OLAP takich jak `RANK()`, `DENSE_RANK()` i `ROW_NUMBER()`.

**Działanie użytkownika**

Nie jest wymagane żadne działanie.

**10507: Przynajmniej jeden z wymiarów w modelu kostki jest pozbawiony hierarchii. Ich włączenie do kostki nie będzie możliwe.**

**Objaśnienie**

Wymiar kostki można utworzyć tylko na podstawie wymiaru mającego przynajmniej jedną hierarchię. Tymczasem podjęto próbę utworzenia lub zmodyfikowania kostki mającej co najmniej jeden wymiar bez hierarchii. Wymiary te zostaną pominięte na liście wyboru używanej do definiowania wymiarów kostki.

**Działanie użytkownika**

Utwórz lub zmodyfikuj kostkę bez odwołań do pominiętych wymiarów lub zadбай o to, by każdy wymiar w modelu kostki miał hierarchię.

**10508: Opcja rozwinięcia rekurencyjnego jest poprawna tylko w przypadku, gdy dla hierarchii wybrano dwa poziomy.**

**Objaśnienie**

Dla rozwinięcia rekurencyjnego muszą istnieć dokładnie dwa poziomy.

**Działanie użytkownika**

Wybierz dokładnie dwa poziomy dla rozwinięcia rekurencyjnego albo wybierz inną opcję rozwinięcia.

**10509: Nie określono wszystkich wymaganych właściwości wycinków optymalizacji. Dla każdego wycinka optymalizacji określ typ zapytania i opcje dla poszczególnych wymiarów kostki.**

**Objaśnienie**

Nie określono wszystkich wymaganych właściwości wycinków optymalizacji.

#### **Działanie użytkownika**

Dla każdego wycinka optymalizacji określ typ zapytania i opcje dla poszczególnych wymiarów kostki.

#### **10510: Wybierz przynajmniej jeden atrybut klucza poziomemu.**

##### **Objaśnienie**

Dla poziomu nie określono żadnych atrybutów klucza poziomemu.

#### **Działanie użytkownika**

Wybierz przynajmniej jeden atrybut klucza poziomemu.

#### **10511: Nie można określać jednocześnie wycinka z typem wyodrębniania MOLAP i wycinka z hybrydowym typem wyodrębniania dla tej samej kostki. Należy usunąć jeden z wycinków z kostki.**

##### **Objaśnienie**

Jedna kostka nie może mieć wycinka optymalizacji z wyodrębnieniem typu MOLAP i z wyodrębnieniem hybrydowym jednocześnie. Dla jednej kostki można zdefiniować tylko jeden typ wyodrębniania w wycinku optymalizacji.

#### **Działanie użytkownika**

Usuń z kostki wycinek z wyodrębnieniem MOLAP lub z wyodrębnieniem hybrydowym.

#### **10512: W ramach jednej kostki dopuszczalne jest istnienie tylko jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP.**

##### **Objaśnienie**

W ramach jednej kostki dopuszczalne jest istnienie tylko jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP.

#### **Działanie użytkownika**

Upewnij się, że nie masz więcej niż jednego wycinka optymalizacji z wyodrębnieniem typu MOLAP.

#### **10513: W ramach jednej kostki dopuszczalne jest istnienie tylko jednego wycinka optymalizacji z wyodrębnieniem hybrydowym.**

##### **Objaśnienie**

W ramach jednej kostki dopuszczalne jest istnienie tylko jednego wycinka optymalizacji z wyodrębnieniem hybrydowym.

#### **Działanie użytkownika**

Upewnij się, że nie masz więcej niż jednego wycinka optymalizacji z wyodrębnieniem hybrydowym.

**10514: Wycinek optymalizacji z rozszerzaniem można utworzyć tylko pod warunkiem, że w danej kostce zdefiniowany już jest wycinek optymalizacji z wyodrębnianiem hybrydowym.**

**Objaśnienie**

Do określenia wycinka optymalizacji z rozszerzaniem wymagany jest wycinek optymalizacji z wyodrębnianiem hybrydowym.

**Działanie użytkownika**

Utwórz typ wyodrębniania hybrydowego dla wycinka optymalizacji. Dopiero wtedy możliwe będzie utworzenie wycinka optymalizacji z rozszerzaniem.

**10515: Program DB2 Cube Views określił parametr ANY jako specyfikację poziomów optymalizacji nowych wymiarów kostki dla każdego z istniejących wycinków optymalizacji.**

**Objaśnienie**

Jeśli dodasz wymiary kostki do kostki po zdefiniowaniu wycinków dla kostki, program DB2 Cube Views rozszerzy istniejące wycinki optymalizacji, określając słowo kluczowe Dowolny w poziomach optymalizacji dla nowych wymiarów kostki.

**Działanie użytkownika**

Poziomy optymalizacji w wycinkach można zmodyfikować, klikając przycisk Określ na stronie Typy zapytań okna Właściwości kostki.

**10516: Należy określić przynajmniej jeden wycinek optymalizacji w oknie Wycinki optymalizacji, ponieważ dla typu zapytań do kostki wybrano ustawienia zaawansowane.**

**Objaśnienie**

W oknie Wycinki optymalizacji należy określić co najmniej jeden wycinek optymalizacji, ponieważ jako typ zapytań do kostki wybrano ustawienia zaawansowane.

**Działanie użytkownika**

Kliknij przycisk **Określ** i utwórz w oknie Wycinki optymalizacji co najmniej jeden wycinek optymalizacji.

**10517: Aktualne zmiany ustawień kostki i wymiarów kostki zostaną zapisane po zamknięciu strony Wymiary. Kliknij przycisk Tak, aby zapisać zmiany w bazie danych. Kliknij przycisk Nie, aby pozostać na stronie Wymiary i wprowadzić dalsze zmiany lub anulować.**

**Objaśnienie**

Po zmodyfikowaniu wymiarów kostki w oknie Właściwości kostki i opuszczeniu strony Wymiary wszystkie zmiany wprowadzone w wymiarach kostki zostaną zapisane w bazie danych i nie będzie można ich cofnąć klikając przycisk **Anuluj**.

**Działanie użytkownika**

Kliknij przycisk **Tak**, aby zapisać bieżące zmiany. Kliknij przycisk **Nie**, aby anulować transakcję i pozostać na stronie Wymiary.



**10518: Nie można zmodyfikować hierarchii kostki, ponieważ dla kostki nazwa\_kostki zostały już zdefiniowane zaawansowane wycinki optymalizacji. Należy usunąć wszystkie zaawansowane wycinki optymalizacji i dopiero wtedy przystąpić do modyfikacji hierarchii kostki.**

**Objaśnienie**

Jeśli dla kostki zostaną zdefiniowane zaawansowane wycinki optymalizacji, nie można modyfikować hierarchii kostki.

**Działanie użytkownika**

Należy usunąć wszystkie zaawansowane wycinki optymalizacji i dopiero wtedy przystąpić do modyfikowania hierarchii kostki.

**10519: Nie można zmodyfikować zestawu wymiarów kostki, ponieważ dla kostki nazwa\_kostki zdefiniowano już zaawansowane wycinki optymalizacji. Należy usunąć wszystkie zaawansowane wycinki optymalizacji i dopiero wtedy dodawać lub usuwać wymiary kostki.**

**Objaśnienie**

Jeśli dla kostki zostaną zdefiniowane zaawansowane wycinki optymalizacji, nie można modyfikować zestawu wymiarów kostki.

**Działanie użytkownika**

Należy usunąć wszystkie zaawansowane wycinki optymalizacji i dopiero wtedy dodawać lub usuwać wymiary kostki.

**10520: Zmodyfikowanie tych właściwości spowoduje także zmodyfikowanie istniejących wycinków optymalizacji kostki. Użytkownik może przejrzeć wycinki optymalizacji w celu wyświetlenia zmian i dokonania modyfikacji lub kliknąć znajdujący się w oknie właściwości przycisk Anuluj, aby zamknąć to okno i anulować dokonane zmiany.**

**Objaśnienie**

Zmodyfikowanie zestawu wymiarów kostki lub hierarchii kostki spowoduje, że istniejące zaawansowane wycinki optymalizacji kostki zostaną zmodyfikowane w jeden z następujących sposobów:

- Jeśli zostanie dodany nowy wymiar kostki, wszystkie istniejące wycinki optymalizacji zostaną rozszerzone o nowy wymiar kostki na poziomie Dowolny.
- Jeśli jakiś wymiar kostki zostanie usunięty, z istniejących wycinków optymalizacji zostanie usunięty odpowiadający mu poziom.
- Jeśli hierarchia kostki zostanie zmieniona tak, że nie będzie istniał poziom zdefiniowany w wycinkach optymalizacji, brakujący poziom hierarchii zostanie zastąpiony poziomem Dowolny we wszystkich stosownych wycinkach optymalizacji.
- Jeśli istniejąca hierarchia kostki zostanie zastąpiona nową, poziomy w wycinkach optymalizacji dla odpowiedniego wymiaru kostki zostaną ustawione na Dowolny.

**Działanie użytkownika**

Otwórz okno Wycinki optymalizacji i zweryfikuj modyfikacje wykonane przez Centrum OLAP. Wprowadź ewentualne potrzebne zmiany.

## Komunikaty stanu programów DB2 i DB2 Cube Views

Po wywołaniu procedury zapisanej w bazie programu DB2 Cube Views, bez względu na to, czy procedura została wykonana, czy nie, program DB2 zwraca do aplikacji wywołującej kody SQLCODE i SQLSTATE. Jeśli procedura zapisana w bazie może zostać wykonana, zwraca ona komunikat stanu będący częścią zwrótnych danych XML przekazywanych aplikacji wywołującej.

Poniższa tabela przedstawia związki między komunikatami stanu, które są zwracane w wyniku operacji na metadanych, a kodami SQLSTATE zwracanymi przez program DB2 jako efekt wywołania procedury.

*Tabela 63. Identyfikatory operacji na metadanych i kody SQLSTATE*

SQL CODE	SQL STATE	Identyfikator komunikatu stanu operacji na metadanych	Typ komunikatu stanu operacji na metadanych	Zwracany komunikat stanu operacji na metadanych
0	0	0 2	Informacyjny	Nie
0	0	1	Informacyjny	Tak
0	0	599 6006 6299 7200 7201 7202	Ostrzegawczy	Nie
462	01HQ1	0 – 7999 (z wyjątkiem identyfikatorów wymienionych w pozostałych wierszach)	Błąd	Nie
443	38Q00	Nie dotyczy	Nie dotyczy	Nie dotyczy
443	38Q01	Nie dotyczy	Nie dotyczy	Nie dotyczy
443	38Q02	Nie dotyczy	Nie dotyczy	Nie dotyczy
443	38Q03	Nie dotyczy	Nie dotyczy	Nie dotyczy

---

## Uwagi

Niniejsza publikacja została przygotowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju/regionie można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi, pochodzących od producenta innego niż IBM, spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie tego dokumentu nie daje żadnych uprawnień licencyjnych do tychże patentów. Zapytania dotyczące licencji można wysłać na piśmie na adres:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Zapytania w sprawie licencji na informacje dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub zgłaszać na piśmie pod adresem:

IBM World Trade  
Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego:**

INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS"), BEZ UDZIELANIA JAKIKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych i domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszym dokumencie mogą zawierać nieścisłości techniczne lub błędy typograficzne. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

IBM ma prawo do używania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Informacje na temat możliwości stosowania tego programu, takie jak: (i) wymiana informacji między niezależnie tworzonymi programami a innymi programami (włącznie z tym programem) czy (ii) wspólne używanie wymienianych informacji, można uzyskać pod adresem:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, uiszczenie odpowiedniej opłaty.

Licencjonowany program opisany w tej publikacji i wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Informacje dotyczące produktów firm innych niż IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące produktów firm innych niż IBM należy kierować do dostawców tych produktów.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennych operacjach działalności gospodarczej. W celu kompleksowego ich zilustrowania, podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary te wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów firm innych niż IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące produktów firm innych niż IBM należy kierować do dostawców tych produktów.

Przy przeglądaniu tych informacji w formie elektronicznej, fotografie i ilustracje kolorowe mogą się nie pojawić.

---

## Znaki towarowe

Następujące nazwy są znakami towarowymi International Business Machines Corporation w Stanach Zjednoczonych i/lub w innych krajach:

AIX  
DB2  
DB2 Connect  
DB2 Universal Database  
IBM  
Office Connect  
Dokumentacja techniczna

Następujące nazwy są znakami towarowymi lub zastrzeżonymi znakami towarowymi innych podmiotów:

Microsoft, Windows, Windows NT, Windows 2000, Windows XP i Microsoft Excel są znakami towarowymi lub zastrzeżonymi znakami towarowymi Microsoft Corporation.

Java oraz wszystkie znaki towarowe i logo dotyczące Java i Solaris są znakami towarowymi Sun Microsystems, Inc. w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym w Stanach Zjednoczonych i/lub w innych krajach i jest licencjonowany na zasadzie wyłączności przez X/Open Company Limited.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa. Red Hat oraz wszystkie znaki towarowe i logo dotyczące Red Hat są znakami towarowymi lub zastrzeżonymi znakami towarowymi Red Hat, Inc. w Stanach Zjednoczonych i/lub w innych krajach.

Nazwy innych firm, produktów i usług mogą być znakami towarowymi lub znakami usług innych podmiotów.



---

## Glosariusz

Ten glosariusz zawiera definicje terminów używanych w podręczniku.

### A

**atrybut.** Obiekt programu DB2 będący odwzorowaniem pojedynczej kolumny w tabeli lub wyrażenie będące kombinacją kolumn lub innych atrybutów, lub jednych i drugich. Atrybut może pełnić kilka ról. Na przykład może stanowić odwołanie do danych znajdujących się w kostce lub odwołanie do kolumny używanej przez łączenie lub inny związek atrybutów.

### F

**fakty kostki.** Obiekt DB2 będący częścią kostki i pochodny względem wymiaru w modelu kostki odpowiadającym danej kostce. Obiekt faktów kostki odwołuje się do podzbioru miar, od których pochodzi.

**funkcja agregująca.** Jedna z funkcji agregujących SQL programu DB2, na przykład SUM, AVG, MIN lub MAX. Funkcja agregująca służy do sterowania operacjami zwiżania miar.

### H

**hierarchia.** Obiekt DB2 definiujący związki w zbiorze złożonym z jednego lub więcej niż jednego atrybutu wewnątrz konkretnego wymiaru modelu kostki. DB2 Cube Views obsługuje cztery typy hierarchii: zrównoważony, nierównoważony, nierówny i sieciowy. Hierarchie można rozwijać jako standardowe lub rekurencyjne.

**hierarchia kostki.** Obiekt DB2 będący częścią wymiaru kostki i pochodny względem hierarchii w wymiarze odpowiadającym wymiarowi kostki. Hierarchia kostki odwołuje się do podzbioru atrybutów hierarchii, od której pochodzi, przy czym kolejność atrybutów musi być taka sama, jak w danej hierarchii.

**hierarchia nierówna.** Hierarchia, w której każdy poziom ma spójne znaczenie, lecz rozgałęzienia mają niespójne głębokości, ponieważ co najmniej jeden atrybut elementu na poziomie rozgałęzienia jest niezapełniony. Patrz: *hierarchia zrównoważona*, *hierarchia sieciowa* i *hierarchia nierównoważona*.

**hierarchia nierównoważona.** Hierarchia o poziomach mających spójny związek typu nadrzędny–podrzędny, lecz mających niespójne znaczenie semantyczne dla wszystkich elementów określonego poziomu. Ponadto rozgałęzienia takiej hierarchii mają niespójne głębokości. Patrz: *hierarchia zrównoważona*, *hierarchia sieciowa* i *hierarchia nierówna*.

**hierarchia sieciowa.** Hierarchia, w której porządek poziomów nie jest określony, ale poziomy mają znaczenie semantyczne. Ponieważ poziomy atrybutów nie mają wewnętrznego związku typu nadrzędny–podrzędny, porządek poziomów nie ma znaczenia. Patrz: *hierarchia zrównoważona*, *hierarchia nierówna* oraz *hierarchia nierównoważona*.

**hierarchia zrównoważona.** Hierarchia o wyraźnie określonych poziomach i rozgałęzieniach o spójnej głębokości. Logiczny element nadrzędny każdego atrybutu znajduje się w hierarchii bezpośrednio nad danym atrybutem. Patrz: *hierarchia sieciowa*, *hierarchia nierówna* oraz *hierarchia nierównoważona*.

### K

**kostka.** Obiekt DB2 pochodny względem modelu kostki. Fakty kostki i wymiary kostki są podzbiorem tych, do których odwołuje się model kostki. Kostki są strukturą odpowiednią dla narzędzi i aplikacji niekorzystających z wielu hierarchii, ponieważ wymiary kostki dopuszczają istnienie tylko jednej hierarchii przypadającej na wymiar.

**kostka hybrydowa.** Zawiera wielowymiarowe dane i odwołuje się do danych relacyjnych, tak że zapytania umożliwiają uzyskanie dostępu do danych niższego poziomu w tabelach podstawowych.

### Ł

**łączenie.** Łączy dwie tabele relacyjne. Łączenie odwołuje się do atrybutów, które z kolei odwołują się do kolumn w tabelach podlegających łączeniu. Najprostsze łączenie odwołuje się do dwóch atrybutów: jednego stanowiącego odwzorowanie kolumny w pierwszej tabeli i jednego stanowiącego odwzorowanie kolumny w drugiej tabeli. Na łączenie składa się także operator wskazujący sposób porównywania kolumn. Obiekt łączenia może również służyć do modelowania łączy złożonych, w których co najmniej dwie kolumny z pierwszej tabeli są łączone z taką samą liczbą kolumn w drugiej tabeli. W łączeniu złożonym pary atrybutów służą do odwzorowywania odpowiadających sobie kolumn. Każda para atrybutów ma przypisany operator wskazujący sposób porównywania kolumn. Łączenie ma także określony typ i licznosc. Łączy można używać w wymiarach do łączenia tabel wymiarów lub w modelu kostki do łączenia wymiarów modelu kostki z obiektem faktów, lub wewnątrz obiektu faktów do łączenia wielu tabel faktów.

**łączenie gwiazdziste.** Łączenie między tabelą faktów (zwykle dużą tabelą faktów) i co najmniej dwiema tabelami wymiarów. Tabela faktów ma łączenie z każdą z tabel wymiarów według klucza wymiaru.

## M

**metadane.** Informacje o właściwościach danych, na przykład o typie danych w kolumnie (liczbowy, tekstowy itd.) lub długości kolumny. Mogą to być również informacje o strukturze danych lub informacje opisujące budowę obiektów takich jak kostki lub wymiary.

**miara.** Obiekt DB2 definiujący jednostkę miary i używany w obiektach faktów. Miary nabierają znaczenia w kontekście wymiaru. Typowymi przykładami obiektów miar są Dochód, Koszt i Zysk.

**miara obliczana.** Zawiera wbudowane obliczenia tworzone przez użytkownika za pomocą Konstruktora wyrażeń w Centrum OLAP lub w języku SQL. Miara obliczana ma przypisane wyrażenie SQL, które wykonuje obliczenia, i nie stanowi odwzorowania pojedynczej kolumny ani atrybutu.

**model kostki.** Obiekt DB2 opisujący wszystkie dane powiązane z kolekcją miar. Zwykle model kostki bazuje na schemacie gwiazdowym lub schemacie typu płatek śniegu w bazie danych. Model kostki odwołuje się do pojedynczego obiektu faktów i co najmniej jednego wymiaru. Modele kostki można optymalizować w celu zwiększenia wydajności zapytań SQL kierowanych do schematu gwiazdowego lub schematu typu płatek śniegu modelu kostki.

**MQT.** Patrz *zmaterializowana tabela zapytania*.

## O

**obiekt faktów.** Obiekt DB2 grupujący pokrewne miary istotne w konkretnej aplikacji. Obiekt faktów przechowuje informacje o atrybutach używanych w łączeniach fakt-wymiar, przy czym atrybuty i łączenia służą do odwzorowywania dodatkowych miar między wieloma tabelami bazy danych. Dlatego oprócz zestawu miar obiekt faktów przechowuje zestaw atrybutów oraz zestaw łączeń. Obiekt faktów stanowi w modelu kostki centrum schematu gwiazdowego.

**ograniczenie.** Reguła, której przestrzeganie jest wymuszane przez menedżera bazy danych. Istnieją cztery typy ograniczeń: przez unikalność, referencyjne, sprawdzające tabeli oraz informacyjne.

## P

**poziom.** Obiekt DB2 zawierający jeden lub więcej atrybutów, które są ze sobą powiązane i działają razem jako jeden logiczny krok w porządkowaniu hierarchii.

**poziom kostki.** Obiekt DB2, który jest podzbiorem poziomu i jest używany w kostce. Poziom kostki odwołuje się do poziomu, z którego został wyodrębniony (nadrzędnego), i dziedziczy po nim atrybuty klucza oraz atrybut domyślny, zdefiniowane dla poziomu nadrzędnego.

## R

**rozwińnięcie rekurencyjne.** Bazuje na wewnętrznym związku typu nadrzędny–podrzędny między atrybutami hierarchii. Hierarchia nierównoważona, w której zastosowano rozwińnięcie rekurencyjne, ma reprezentację w postaci par atrybutów typu nadrzędny–podrzędny.

**rozwińnięcie standardowe.** Bazuje na definicji poziomów hierarchii, przy czym każdy atrybut w hierarchii definiuje jeden poziom. Na przykład hierarchię zrównoważoną dla wymiaru Czas zwykle tworzą poszczególne zdefiniowane poziomy, w tym Rok, Kwartał i Miesiąc. Rozwińnięcie standardowe można stosować we wszystkich czterech typach hierarchii.

## S

**schemat.** W standardzie SQL-92 jest to kolekcja obiektów bazy danych, których właścicielem jest pojedynczy użytkownik, i które tworzą pojedynczą przestrzeń nazw. Przestrzeń nazw to zbiór obiektów, których nazwy nie mogą się w ramach tego zbioru powtarzać. Na przykład dwie tabele mogą mieć tę samą nazwę tylko wtedy, gdy znajdują się w osobnych schematach, zaś żadne dwie tabele w tym samym schemacie nie mogą mieć tej samej nazwy.

**schemat gwiazdowy.** Struktura relacyjnej bazy danych, w której dane są przechowywane tylko w jednej tabeli faktów zajmującej centralne miejsce w schemacie, a dane dodatkowych wymiarów są przechowywane w tabelach wymiarów. Każda tabela wymiarów jest bezpośrednio związana i zwykle połączona z tabelą faktów przez kolumnę klucza. Schematów gwiazdowych używa się w hurtowniach danych.

**schemat typu płatek śniegu.** Rozszerzenie schematu gwiazdowego polegające na tym, że co najmniej jeden wymiar jest zdefiniowany przez wiele tabel. W schemacie typu płatek śniegu tylko tabele wymiarów podstawowych mają łączenia z tabelą faktów. Dodatkowe tabele wymiarów mają łączenia z tabelą wymiarów podstawowych.

## T

**tabela faktów.** Centralna tabela w schemacie hurtowni danych zawierająca miary liczbowe i klucze wiążące fakty z tabelami wymiarów. Tabela faktów zawiera dane opisujące konkretne zdarzenia biznesowe, takie jak transakcje bankowe lub wyniki sprzedaży produktu.

**tabela oddalona.** Dowolna tabela wymiarów w schemacie typu płatek śniegu niebędąca tabelą wymiarów podstawowych w wymiarze.

**tabela podsumowań.** Zawiera zagregowane dane tabel podstawowych używanych w modelu kostki. Moduł DB2 Cube Views wykorzystuje tabele podsumowań DB2 do zwiększania wydajności zapytań kierowanych do modeli kostki. Tabela podsumowań to zmaterializowana tabela



zapytania (MQT) szczególnego typu zawierająca dane podsumowań. Ponieważ w programie DB2 Cube Views zawsze zalecane jest stosowanie tabel MQT mających podsumowane dane, w dokumentacji modułu DB2 Cube Views termin tabela podsumowań jest używany na określenie zalecanych tabel MQT. Patrz *zmaterializowana tabela zapytania*.

**tabela wymiarów.** Tabela w hurtowni danych, której pozycje opisują dane w tabeli faktów. Tabele wymiarów zawierają dane, na podstawie których wymiary zostały utworzone.

**tabela wymiarów podstawowych.** W schemacie typu płatek śniegu tabela wymiarów łącząca się z tabelą faktów.

## W

**wycinek.** Rejon wielowymiarowej bazy danych lub kostki.

**wymiar.** Obiekt DB2 odwołujący się do kolekcji pokrewnych atrybutów opisujących pewien aspekt zestawu miar. Wymiar może odwoływać się do atrybutów z jednej lub więcej niż jednej tabeli wymiarów. Jeśli jednak są używane atrybuty z wielu tabel wymiarów, między tymi tabelami muszą istnieć łączenia, a wymiar musi się do tych połączeń odwoływać. Wymiar ponadto odwołuje się do co najmniej jednej hierarchii i może się odwoływać do związków między swoimi atrybutami.

**wymiar kostki.** Obiekt DB2 będący częścią kostki i pochodny względem wymiaru w modelu kostki odpowiadającym danej kostce. Wymiar kostki odwołuje się do podzbioru atrybutów wymiaru, od którego pochodzi. Odwołuje się również do pojedynczej hierarchii kostki.

## Z

**zależność funkcjonalna.** Obiekt DB2 wskazujący, że atrybut domyślny obiektu poziomym oraz atrybuty pokrewne są funkcjonalnie zdeterminowane przez atrybuty kluczy poziomym. Zależności funkcjonalne umożliwiają określenie, że jedna lub więcej niż jedna kolumna jest funkcjonalnie zależna od jednej lub więcej niż jednej spośród pozostałych kolumn, przy czym wszystkie te kolumny muszą istnieć w tej samej tabeli.

**zmaterializowana tabela zapytania.** Tabela, której definicja bazuje na wyniku zapytania, i której dane mają postać wstępnie obliczonych wyników pobranych z jednej lub więcej niż jednej tabeli będącej podstawą definicji zmaterializowanej tabeli zapytania.

**związek atrybutów.** Ogólnie opisuje związki obiektów atrybutów. Związki są opisywane przez atrybut lewostronny i prawostronny, typ, licznosc i to, czy określa zależność funkcjonalną. Typ opisuje, jaka jest rola atrybutu prawostronnego względem atrybutu lewostronnego. Możliwe są dwa typy: Opisowy i Skojarzony. Typ Opisowy określa, że atrybut prawostronny jest deskryptorem atrybutu lewostronnego.



# Indeks

## A

- Alter, operacja 144
- analiza składniowa kodu XML 132
- API (aplikacyjny interfejs programistyczny)
  - parametry 135
  - procedura zapisana w bazie 133
  - przegląd 131
  - śledzenie 157
  - wymiana danych 131
- atrybuty
  - domyślne 27
  - klucz poziomu 27
  - odzworowanie relacyjne 12, 15
  - opis 32
  - pokrewne 27
  - reguły podstawowe 40
  - w tabelach podsumowań 87
  - właściwości 32
- atrybuty domyślne
  - opis 27
- atrybuty klucza poziomu
  - opis 27
- atrybuty pokrewne
  - dodawanie do hierarchii kostki 55
  - opis 27

## B

- baza danych DB2
  - nawiązywanie połączenia 47
  - rozłączanie 47
  - usuwanie obiektów metadanych 57
- bazy danych
  - zdalne 124
- błędy, protokołowanie 158

## C

- Create, operacja 143
- currentRef, operand 155
- CVSAMPLE, baza danych 167, 168

## D

- DB2 EXPLAIN, narzędzie 85
- db2batch, narzędzie do testów porównawczych 117
- Describe, operacja 136
- drażące, zapytania 94
- Drop, operacja 145

## E

- eksportowanie
  - obiekty metadanych 48
- Eksportuj, okno 48

## F

- fakty-wymiar, łączenia 53, 107
- tworzenie 53

## G

- gwiazdziste, schematy 11

## H

- hierarchie
  - modelowanie pod kątem optymalizacji 93
  - odzworowanie relacyjne 12, 17
  - opis 22
  - reguły podstawowe 39
  - rozwnięcia 25
  - tworzenie 52
  - tworzenie poziomów 52
  - typy 22
  - właściwości 26
- hierarchie kostki
  - dodawanie do wymiaru kostki 55
  - hierarchie kostki
    - określanie 55
  - odzworowanie relacyjne 18
  - określanie 55
  - opis 37
  - reguły podstawowe 42
  - właściwości 37

## I

- Import, operacja 146
- importowanie
  - obiekty metadanych 47
- instalowanie
  - AIX 6
  - Linux 6
  - Środowisko Operacyjne Solaris 6
  - Windows 6

## K

- konfiguracja systemu 132
- Korelacja
  - reklama 65
  - sprzedaż 65
- kostki 55
  - modelowanie pod kątem optymalizacji 94
  - odzworowanie relacyjne 18
  - opis 35
  - reguły podstawowe 41
  - tworzenie 54
  - właściwości 35
- Kreator dodawania wymiaru 53
- Kreator Doradca optymalizacyjny 82, 92, 93, 111, 113, 119

- Kreator faktów 50
- Kreator hierarchii 52
- Kreator importu 47
- Kreator kostek 54
- Kreator łączeń 53
- Kreator modeli kostek 50
- Kreator poziomów 52
- Kreator szybkiego startu 48
- Kreator wymiarów 51

## L

- liczności 33

## Ł

- łączenia
  - fakty-wymiar 107
  - liczności 107
  - odzworowanie relacyjne 13, 15
  - opis 34
  - reguły optymalizacji 40
  - reguły podstawowe 40
  - tworzenie 53
  - typy 107
  - właściwości 35
- łączenia faktów z wymiarem, tworzenie 51

## M

- md\_message
  - procedury zapisane w bazie 133
- miary
  - modelowanie pod kątem optymalizacji 93
  - obliczane 51
  - odzworowanie relacyjne 11
  - opis 30
  - proste 51
  - reguły podstawowe 39
  - w tabelach podsumowań 87
  - właściwości 31
- mode dla operacji import 153
- mode dla sprawdzania poprawności (operacja Validate) 155
- modele kostek
  - hierarchie, tworzenie 52
  - obiekty faktów, tworzenie 50
  - odzworowanie relacyjne 11
  - opis 21
  - optymalizowanie 113
  - reguły kompletności 38, 107
  - reguły optymalizacji 38
  - reguły podstawowe 38
  - tworzenie 49, 50
  - tworzenie przy użyciu Kreatora szybkiego startu 48
  - usuwanie wymiarów 56
  - właściwości 21
  - wymiary, dodawanie 53

- modele kostek (*kontynuacja*)
  - wymiary, tworzenie 51
- modelowanie
  - idealne 28
  - nieidealne 28
- modelowanie idealne 28
- modelowanie nieidealne 28
- Monitor obrazów stanu SQL DB2 118
- mosty 4
- mosty metadanych 4

## N

- newRef, operand 155
- nierówne, hierarchie 24
- niezrównoważone, hierarchie 23

## O

- obiekty faktów
  - odzworowanie relacyjne 11
  - opis 21
  - reguły podstawowe 38
  - tworzenie 50
  - właściwości 21
- obiekty faktów kostki
  - odzworowanie relacyjne 18
  - opis 36
  - reguły podstawowe 42
- obiekty metadanych 4
  - atrybuty 32
  - eksportowanie 48
  - format 163
  - hierarchie 22
  - hierarchie kostki 37
  - hierarchie, tworzenie 52
  - importowanie 47
  - konwencje nazewnictwa 20
  - kostki 35
  - kostki, tworzenie 54
  - łączenia 34
  - łączenia, tworzenie 53
  - miary 30
  - modele kostek 21
  - modele kostek, tworzenie 50
  - modele kostek, tworzenie przy użyciu Kreatora szybkiego startu 48
  - modelowanie pod kątem optymalizacji 93
  - obiekty faktów 21
  - obiekty faktów kostki 36
  - obiekty faktów, tworzenie 50
  - ogólne właściwości 19
  - poziomy 27
  - poziomy kostki 37
  - poziomy, tworzenie 52
  - przegląd 11
  - reguły podstawowe 38
  - wymiana 47
  - wymiary 22
  - wymiary kostki 36
  - wymiary, dodawanie 53
  - wymiary, tworzenie 51
  - związki atrybutów 33
- object, operand 155
- objectType, parametr 152

- obliczane, miary 51
- obliczanie
  - marża 70
  - przepływ towarów 59
  - wartość 59
  - zysk 70
- obsługa błędów 132
- odświeżanie natychmiastowe tabel podsumowań 120
- odświeżanie odroczone tabel podsumowań 120
- utrata ważności 119
- ograniczenia 82, 113, 128
  - informacyjne 107
  - klucz obcy 107
  - tabele podsumowań 88
- ograniczenia informacyjne 82, 107, 113
- operacja Translate 148
- operacja Validate 148
- operacje administrowania 151
- operacje modyfikacji 150
- operacje na metadanych
  - Alter 144
  - Create 143
  - Describe 136
  - Drop 145
  - Import 146
  - Rename 144
  - Translate 148
  - Validate 148
- operacje pobierania 150
- operandy 155
- operandy operacji 155
- optymalizacja 88
  - ograniczenia czasowe 112
  - ograniczenia dotyczące miejsca na dysku 112
  - proces 90
  - próbkowanie danych 113
  - przy użyciu tabel podsumowań 82
  - reguły 107
  - skrypty SQL 114
  - testowanie wydajności 91
  - testy porównawcze 117
  - typy zapytań 111
  - wycinki 111
- optymalizator DB2 82, 88
- optymalizowanie
  - modele kostek 113

## P

- parametry operacji 152
  - mode dla operacji import 153
  - mode dla sprawdzania poprawności (operacja Validate) 155
  - objectType 152
  - recurse 152
  - restriction 153
- plik konfiguracyjny 159, 162
- plątek śniegu, schematy 11, 109
- Połączenie z bazą danych, okno 47
- poziomy
  - atrybuty domyślne 27
  - atrybuty klucza poziomu 27
  - atrybuty pokrewne 27
  - dodawanie do hierarchii kostki 55

- poziomy (*kontynuacja*)
  - modelowanie idealne 28
  - modelowanie nieidealne 28
  - modelowanie pod kątem optymalizacji 93
  - odzworowanie relacyjne 12, 15
  - opis 27
  - reguły podstawowe 39
  - tworzenie 52
  - właściwości 29
- poziomy kostki
  - opis 37
  - reguły podstawowe 42
  - właściwości 37
- procedury zapisane w bazie md\_message 133
- proste, miary 51
- protokołowanie
  - śledzenie 157
- protokołowanie błędów 158
- przegląd
  - obiekty metadanych 11
- przekierowywanie zapytań 82, 88
- przykład
  - pliki aplikacji 168
  - pliki bazy danych 167
- przykłady
  - parametry API
    - operacje administrowania 151
    - operacje modyfikacji 150
    - operacje pobierania 150
- pseudonimy 127

## R

- Ranking wysokości obrotów
  - DENSERANK 75
  - RANK 75
  - ROWNUMBER 75
- raportowe, zapytania 94
- recurse, parametr 152
- reguły
  - kompletność modelu kostki 107
  - optymalizacja 107
  - podstawowe 107
- reguły kompletności
  - modele kostek 38
- reguły optymalizacji
  - łączenia 40
  - modele kostek 38
  - wymiary 39
- reguły podstawowe 107
  - atrybuty 40
  - hierarchie 39
  - hierarchie kostki 42
  - kostki 41
  - łączenia 40
  - miary 39
  - modele kostek 38
  - obiekty faktów 38
  - obiekty faktów kostki 42
  - poziomy 39
  - poziomy kostki 42
  - wymiary 39
  - wymiary kostki 42
  - związki atrybutów 40
- Rename, operacja 144

restriction, parametr 153  
rozwińnięcie rekurencyjne 26  
rozwińnięcie standardowe 25

## S

schemat typu płatek śniegu 14  
schematy  
gwiazdzisty 11  
o strukturze płatka śniegu 11, 109  
serwery  
stowarzyszone 124  
sieciowe, hierarchie 25  
skrypty SQL 113, 114  
stowarzyszone  
bazy danych 124  
serwery 124  
strony kodowe 159  
struktura komunikatów w interfejsie API 155  
systemy stowarzyszone 124  
klienci 123  
ograniczenia 128  
przegląd 123  
pseudonimy 127  
serwer stowarzyszony 123  
stowarzyszone bazy danych 123  
umożliwianie obsługi przez DB2 Cube Views 126  
zdalne źródła danych 126, 127  
źródła danych 123

## Ś

śledzenie 157  
protokołowanie 157  
śledzenie w czasie wykonywania 157

## T

tabele oddalone 109  
tabele podsumowań 82, 90, 113  
konserwacja 92, 119  
odświeżanie natychmiastowe 120  
odświeżanie odroczone 119, 120  
ograniczenia 88  
skrypty SQL 114  
tworzenie 92  
usuwanie 93, 121  
zależności funkcjonalne 88  
tabele relacyjne 11  
transakcja 132

## U

uprawnienia 44  
usuwanie obiektów metadanych 57

## W

właściwości  
atrybuty 32  
fakty kostki 36  
właściwości 36  
hierarchie 26  
hierarchie kostki 37

właściwości (*kontynuacja*)

kostki 35  
łączenia 35  
miary 31  
modele kostek 21  
obiekty faktów 21  
poziomy 29  
poziomy kostki 37  
wspólne 19  
wymiary 22  
wymiary kostki 37  
związki atrybutów 34  
wycinek 83  
wycinki  
optymalizacja 95  
wycinki optymalizacji 94, 95, 104  
Dowolne 95  
drażnienie 96  
określanie 55  
poziomy 95  
raportowe 97  
rozszerzanie 102  
Wszystkie 95  
wyodrębnianie hybrydowe 100  
wyodrębnianie MOLAP 99  
wycinki optymalizacji, określanie 55  
wymagania  
oprogramowanie 5  
sprzęt 5  
wymagania instalacyjne 5  
wymagania systemowe 5  
wymagania wstępne 6  
Wymiar Czas 78  
wymiary  
dodawanie 53  
hierarchie, tworzenie 52  
odzworowanie relacyjne 12, 15, 17  
opis 22  
reguły optymalizacji 39  
reguły podstawowe 39  
tworzenie 51  
usuwanie z modelu kostki 56  
właściwości 22  
wymiary kostki  
dodawanie hierarchii kostki 55  
odzworowanie relacyjne 18  
opis 36  
reguły podstawowe 42  
właściwości 37  
wyniki pobierania 136

## Z

z DB2 Cube Views 124  
zależności funkcjonalne  
określanie 52  
tabele podsumowań 88  
zapytania  
DB2EXPLAIN 118  
drażące 94, 96  
przechwytywanie 118  
przekierowywanie 82, 85, 88  
raportowe 94, 97  
rozszerzanie 102  
testy porównawcze 117  
typy 94  
wyodrębnianie hybrydowe 100

zapytania (*kontynuacja*)

wyodrębnianie MOLAP 95, 99  
zapytania wyodrębniania MOLAP 95  
zarządzanie pamięcią 132  
zdalne  
źródła danych 124  
zdalne źródła danych  
definiowanie 126  
zliczanie 73  
zmaterializowane tabele zapytań 82  
zrównoważone, hierarchie 23  
związki atrybutów  
opis 33  
reguły podstawowe 40  
typy  
opisowy 33  
skojarzony 33  
w tabelach podsumowań 87  
właściwości 34

## Ź

źródła danych  
zdalne 124, 126



---

## Kontakt z firmą IBM

Jeśli wystąpi problem techniczny, przed skontaktowaniem się z Obsługą klienta w sprawie programu DB2 Cube Views należy zapoznać się z działaniami proponowanymi w dokumentacji produktu i je wykonać. Podręcznik ten zawiera sugestie co do informacji, które można zebrać, aby ułatwić pracownikom Obsługi klienta udzielenie skutecznej pomocy.

W celu uzyskania informacji na temat produktów DB2 Cube Views lub w celu ich zamówienia należy się skontaktować z przedstawicielem firmy IBM w najbliższym jej oddziale lub z autoryzowanym remarketerem oprogramowania firmy IBM.

Mieszkańcy USA mogą zatelefonować pod jeden z następujących numerów:

- 1-800-237-5511 (dział obsługi klienta)
- 1-888-426-4343 (informacje o dostępnych opcjach serwisowych)

---

## Informacje o produktach

Mieszkańcy USA mogą zatelefonować pod jeden z następujących numerów:

- 1-800-IBM-CALL (1-800-426-2255) lub 1-800-3IBM-OS2 (1-800-342-6672), aby zamówić produkty lub uzyskać ogólne informacje.
- 1-800-879-2755, aby zamówić publikacje.

**<http://www.ibm.com/software/data/db2/db2md/>**

Zawiera łącza do stron z informacjami o programie DB2 Cube Views.

**<http://www.ibm.com/software/data/db2/udb>**

Na stronach WWW programu DB2 Universal Database udostępniane są m.in. bieżące informacje o nowościach, opisy produktów i terminy szkoleń.

**<http://www.elink.ibm.com/>**

Kliknięcie odsyłacza Publications powoduje otwarcie serwisu WWW zamówień International Publications, udostępniającego informacje o zamawianiu podręczników.

**<http://www.ibm.com/education/certify/>**

Przedsięwzięcie Professional Certification Program prowadzone w ramach serwisu WWW firmy IBM udostępnia informacje o testach certyfikacyjnych ze znajomości rozmaitych produktów firmy IBM.

**Uwaga:** W niektórych krajach autoryzowani dealerzy IBM powinni się kontaktować ze swoją placówką obsługi dealerów, a nie z Centrum wsparcia IBM.

---

## Komentarz na temat dokumentacji

Opinie użytkowników pomogą firmie IBM w opracowywaniu najwyższej jakości informacji. Prosimy o przysyłanie wszelkich komentarzy na temat tego podręcznika lub pozostałej dokumentacji programu DB2 Cube Views. Swoje komentarze można dostarczać w następujący sposób:

- Wysyłając je przy użyciu internetowego formularza komentarzy czytelników, znajdującego się pod adresem [www.ibm.com/software/data/rcf](http://www.ibm.com/software/data/rcf).
- Wysyłając je pocztą elektroniczną (w wiadomości e-mail) na adres [comments@us.ibm.com](mailto:comments@us.ibm.com). Należy pamiętać o podaniu nazwy produktu, numeru wersji

produktu oraz tytułu podręcznika i jego numeru PN (jeśli występuje). Jeśli komentarz dotyczy konkretnego tekstu, prosimy wskazać położenie tekstu (na przykład tytuł, numer tabeli lub numer strony).







Numer Programu: 5724-E15

SC85-0102-01

