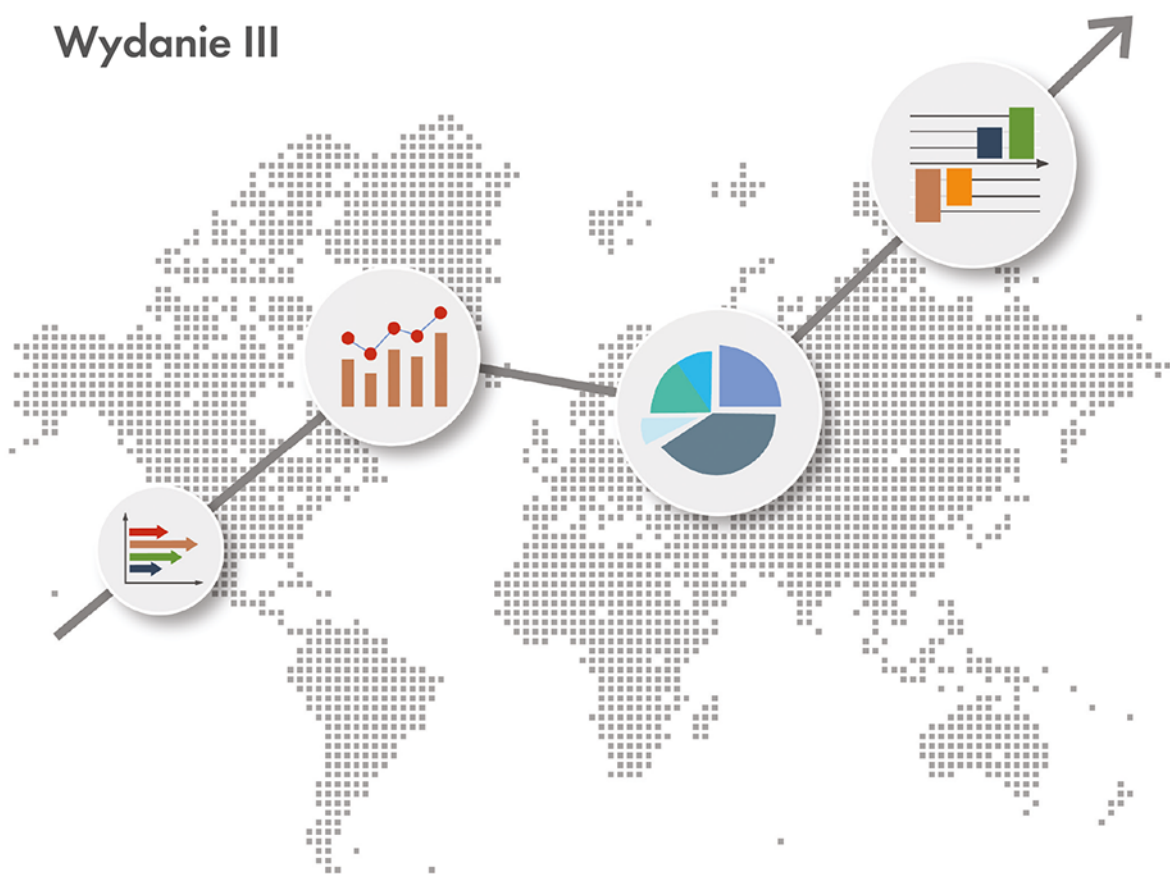


Microsoft® Power BI

Jak modelować i wizualizować dane
oraz budować narracje cyfrowe

Wydanie III



Tytuł oryginału: Microsoft Power BI Quick Start Guide: The ultimate beginner's guide to data modeling, visualization, digital storytelling, and more, 3rd Edition

Tłumaczenie: Anna Mizerska, z wykorzystaniem fragmentów poprzedniego wydania w przekładzie Grzegorza Kowalczyka

ISBN: 978-83-8322-724-5

Copyright © Packt Publishing 2022. First published in the English language under the title 'Microsoft Power BI Quick Start Guide - Third Edition - (9781804613498)'

Polish edition copyright © 2023 by Helion S.A.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/mspbi3>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści |

O autorach	9
O recenzentach	11
Przedmowa	13
ROZDZIAŁ 1.	
Rozwijanie umiejętności korzystania z danych z Power BI	17
Wyzwania organizacji o słabych umiejętnościach korzystania z danych	17
Rozwiązanie problemu słabych umiejętności korzystania z danych za pomocą strategii	18
Dlaczego Power BI	20
Przejęcie z Excela do Power BI	21
Podsumowanie	22
ROZDZIAŁ 2.	
Łączenie się z danymi	23
Pierwsze kroki	24
Importowanie danych	26
Excel jako źródło danych	27
SQL Server jako źródło danych	29
Dataverse jako źródło danych	30
Sieć WWW jako źródło danych	31
DirectQuery	33
Ograniczenia	34
Modele złożone	36
Połączenie na żywo	37
Ograniczenia	38
Wybór metody połączenia się z danymi	39
Podsumowanie	40

ROZDZIAŁ 3.

Strategie przekształcania danych	41
Edytor Power Query	42
Podstawy przekształceń	43
Użyj pierwszego wiersza jako nagłówka	44
Usuwanie kolumn	45
Zmiana typów danych	46
Dodawanie kolumny z przykładów	48
Zaawansowane opcje przekształcania danych	49
Dodawanie kolumn warunkowych	50
Przekształcenie Fill Down (Wypełnij w dół)	52
Przekształcenie Unpivot (Anuluj przestawienie kolumn)	54
Scalanie zapytań	59
Dołączanie zapytań	63
Język programowania R	64
Instalacja i konfiguracja	65
Przekształcenie z wykorzystaniem skryptu R	66
Funkcja AI Insights	69
Ocenianie tonacji za pomocą funkcji Text Analytics (Analiza tekstu)	70
Język formuł M	72
Odkrywanie biblioteki M Query za pomocą funkcji #shared	73
Podsumowanie	75

ROZDZIAŁ 4.

Budowanie modelu danych	76
Organizacja danych za pomocą schematu gwiazdy	77
Tworzenie relacji	78
Edytowanie relacji	81
Tworzenie nowej relacji	84
Praca z relacjami złożonymi	89
Relacje typu wiele-do-wielu	89
Tabele odgrywające role	96
Usprawnienia użyteczności modelu danych	101
Ukrywanie tabel i kolumn	101
Zmiana nazw tabel i kolumn	102
Podsumowanie domyślne	105
Jak wyświetlić jedną kolumnę, a posortować według innej	107
Kategoryzacja danych	109
Tworzenie hierarchii	111

Poprawa wydajności modelu danych	114
Wydajność zapytań	114
Wydajność przetwarzania	118
Podsumowanie	119

ROZDZIAŁ 5.

Wykorzystanie języka DAX	121
Tworzenie kolumn obliczeniowych	121
Tworzenie kolumn obliczeniowych z użyciem funkcji łańcuchowych	124
Zastosowanie funkcji FORMAT w języku DAX	127
Zastosowanie logiki warunkowej za pomocą IF()	127
Zastosowanie logiki warunkowej za pomocą SWITCH()	129
Wykorzystanie istniejących relacji z użyciem funkcji nawigacyjnych	131
Tworzenie miar	133
Tworzenie podstawowych miar	134
Parametry opcjonalne	138
Przypisywanie miar	138
Foldery wyświetlania	139
Kontekst filtra	141
Zmiana kontekstu filtra za pomocą funkcji CALCULATE	142
Obliczanie procentu całości	144
Funkcje analizy czasowej	145
Tworzenie obliczeń sprzedaży YTD	145
Tworzenie obliczenia sprzedaży w poprzednim roku za pomocą funkcji CALCULATE()	147
Tabele odgrywające role w języku DAX	148
Podsumowanie	151

ROZDZIAŁ 6.

Wizualizacja danych	153
Podstawy tworzenia raportów	155
Tworzenie nowych wizualizacji	157
Filtrowanie wizualizacji i danych	159
Filtrowanie krzyżowe i wyróżnianie krzyżowe	160
Edytowanie interakcji	163
Fragmentator	163
Filtry raportu	171
Wizualizacja danych tabelarycznych	173
Tabela	173
Macierz	177

Wizualizacja danych według kategorii	179
Wykresy słupkowe i kolumnowe	180
Wykresy kołowe i pierścieniowe	181
Mapa drzewa	182
Wykres punktowy	185
Wizualizacja trendów	188
Wykresy liniowe i warstwowe	188
Wykresy kombi	189
Wykresy wstążkowe	191
Wykres kaskadowy	192
Wykres lejkowy	194
Wizualizacja kluczowych wskaźników wydajności (KPI)	195
Miernik	195
Kluczowy wskaźnik wydajności	197
Wizualizacja danych za pomocą kart	199
Karta	199
Karta z wieloma wierszami	200
Wizualizacja danych geograficznych	201
Mapa	203
Kartogram	204
Mapa kształtów	206
Mapy ArcGIS	207
Mapy Azure	209
Zaawansowane wizualizacje	210
Język naturalny z funkcją Q&A	210
Wizualizacje analityczne	213
Niestandardowe wizualizacje w Power BI	214
Wskazówki i triki dotyczące wizualizacji danych	215
Zmiana wizualizacji	215
Formatowanie wizualizacji	216
Karta Analiza	217
Filtr Najlepsze N elementów	218
Opcja Pokaż wartości jako	219
Podsumowanie	219

ROZDZIAŁ 7.

Digital storytelling, czyli tworzenie narracji cyfrowych

za pomocą Power BI	221
Konfigurowanie przeglądania szczegółowego	222
Przechwytywanie widoków raportów za pomocą zakładek	229

Połączenie widoczności obiektów z zakładkami	234
Tworzenie zakładek dla różnych widoków tych samych danych	235
Używanie przycisków do wybierania zakładek	237
Używanie nawigatora zakładek	239
Strony raportu jako etykiety narzędzi	242
Używanie nawigatora stron	245
Podsumowanie	247

ROZDZIAŁ 8.

Wdrożenia usługi Power BI w chmurze	248
Omówienie usługi Power BI	249
Tworzenie obszarów roboczych	251
Publikowanie raportów w usłudze Power BI	254
Zestawy danych	258
Skoroszyty	259
Tworzenie pulpitów nawigacyjnych i interakcja z nimi	259
Tworzenie pierwszego pulpitu nawigacyjnego	260
Zadawanie pytań dotyczących danych	264
Udostępnianie raportów pulpitów nawigacyjnych	266
Subskrybowanie raportów	267
Aplikacje Power BI	269
Utrzymywanie pulpitów nawigacyjnych	271
Ustawianie zabezpieczeń na poziomie wiersza	271
Planowanie odświeżania danych	273
Śledzenie postępu za pomocą metryk	278
Podsumowanie	280

ROZDZIAŁ 9.

Oczyszczanie danych w chmurze z wykorzystaniem przepływów danych	281
Rozpoczęcie pracy z przepływami danych	282
Odkrywanie datamart	283
Tworzenie przepływu danych	285
Używanie przepływów danych jako źródła danych w Power BI Desktop	295
Podsumowanie	297

ROZDZIAŁ 10.

Rozwiązania lokalne z wykorzystaniem serwera raportów Power BI	298
Uzyskanie dostępu do Power BI Server	299
Publikowanie na lokalnym serwerze Power BI Report Server	303
Publikowanie raportów na lokalnym serwerze Power BI Report Server	303
Publikowanie raportów z podziałem na strony	305
Zabezpieczanie raportów	307
Tworzenie harmonogramu odświeżania danych	308
Podsumowanie	310

Budowanie modelu danych

W tym rozdziale będziemy się zajmować tworzeniem spójnego i inteligentnego modelu danych. Tworzenie modelu danych to przede wszystkim proces tworzenia niezbędnych relacji między różnymi źródłami danych, które są wykorzystywane w Twoim modelu.

Bez efektywnego modelu danych powstanie rozwiązań BI nie byłoby możliwe. Historycznie rzecz biorąc, tradycyjne projekty BI koncentrujące się na budowaniu modeli danych mogły ciągnąć się miesiącami, a nawet latami, gdy były realizowane w sztywnej strukturze i przy ograniczeniach korporacyjnego środowiska Business Intelligence. Niestety, badania pokazują, że około 50% wszystkich projektów BI w przedsiębiorstwach kończyło się fiaskiem. Takie projekty mogą kosztować od setek tysięcy do milionów dolarów ze względu na koszty związane z infrastrukturą, licencjami i robocizną. Innym powodem niskiego wskaźnika sukcesu jest to, że biznes i użytkownicy końcowi często przez wiele miesięcy nie widzą żadnych konkretnych rezultatów, co może skutkować narastającą frustracją spowodowaną brakiem „widoczności” projektu w kluczowych obszarach biznesowych. Wydłużone terminy realizacji projektów bardzo często wynikają z czasu niezbędnego na zebranie wymagań, zaprojektowanie złożonego modelu danych oraz przygotowanie i oczyszczenie oryginalnych źródeł danych. Wiele projektów BI, które zostały ukończone, często nie jest w stanie zapewnić oczekiwanych rezultatów i nierzadko brakuje w nich komponentów niezbędnych do wykonywania analiz wymaganych przez biznes.

Na szczęście Power BI Desktop usuwa niemal wszystkie bariery, które były przyczyną wysokiego wskaźnika niepowodzeń tradycyjnych projektów BI. Power BI Desktop zapewnia znacznie bardziej elastyczne podejście do budowania modelu danych, dlatego realizacja projektu najczęściej jest mierzona w dniach, a nie miesiącach czy latach, koszty są wykładniczo niższe, a w razie potrzeby można łatwo dodać wszelkie brakujące komponenty. W tym rozdziale omówimy następujące zagadnienia:

- Organizacja danych za pomocą schematu gwiazdy.
- Tworzenie relacji.
- Praca z relacjami złożonymi.

- Poprawa funkcjonalności.
- Poprawa wydajności modelu danych.

Power BI Desktop, jego elastyczność i ogromne możliwości znacznie zwiększają szansę na to, że Twoje projekty zakończą się pełnym sukcesem. Power BI pozwala na łatwe i szybkie tworzenie relacji między różnymi tabelami, które zostały zaimportowane do modelu danych. W tym rozdziale dowiesz się, jak można budować takie relacje w Power BI Desktop.

Organizacja danych za pomocą schematu gwiazdy

Współpracowaliśmy z konsultantami, którzy przez całą swoją karierę zawodową zajmowali się budową modeli danych zaprojektowanych specjalnie na potrzeby raportowania i analityki. Ci sami konsultanci powiedzą, że zawsze uczysz się nowych metod modelowania danych na podstawie wymagań firmy, usprawnień technologicznych i danych. Budowanie modelu danych to bardziej sztuka niż nauka.

Jednak nie musisz mieć 20 lat doświadczenia w modelowaniu danych, by tworzyć z sukcesem rozwiązania w Power BI. Wystarczy, że zrozumiesz podstawy **schematu gwiazdy**.

Schemat gwiazdy to sposób modelowania danych, który szczególnie ułatwia tworzenie raportów i przeprowadzanie analiz. Schemat gwiazdy sprawia, że łatwiej zrozumieć cały model i pracować z nim. Ta metoda modelowania jest częścią dużo szerszego zagadnienia, jakim jest **modelowania wymiarowe**. W modelowaniu wymiarowym uwzględniamy dwa typy tabel: fakty i wymiary.

Fakty to nic innego jak wydarzenia, które chcesz mierzyć. To może być sprzedaż towarów, student uczęszczający na zajęcia, kontakty z członkami i każdy inny rodzaj zdarzenia, który chcesz mierzyć. Typ zdarzenia może różnić się w zależności od jednostki firmy czy branży.

Z kolei wymiary są tym, co opisuje Twoje fakty (wydarzenia). Jeśli firma zarobiła 5 milionów na sprzedaży, możesz zapytać, w którym roku, w jakim kraju, dzięki którym produktom lub sprzedawcom to się stało. Każde z tych niezwykle istotnych pytań dotyczących danych mówi nam dokładnie, jakich tabel wymiarów (opisowych) potrzebujemy w naszym modelu. Tutaj wydzieliłobyśmy cztery osobne tabele wymiarów:

- Data (rok),
- Geografia (kraj),

- Pracownik (sprzedawca),
- Produkty (produkt).

W tej książce korzystamy z modelu, który jest wymiarowy, z jedną tabelą faktów i wieloma tabelami wymiarów. Gdy przejdiesz do widoku modelu, tabele mogą się tam układać w kształt gwiazdy, stąd nazwa „schemat gwiazdy”.

W tym podrozdziale krótko omówiliśmy modelowanie wymiarowe i schemat gwiazdy. Nie będziemy zagłębiać się w te tematy bardziej, gdyż nie jest to tematem tej książki, pamiętaj jednak o tym, czego się tutaj dowiedziałeś, gdy będziesz czytać pozostałą część niniejszej książki, i zacznij budować własne modele danych!

Tworzenie relacji

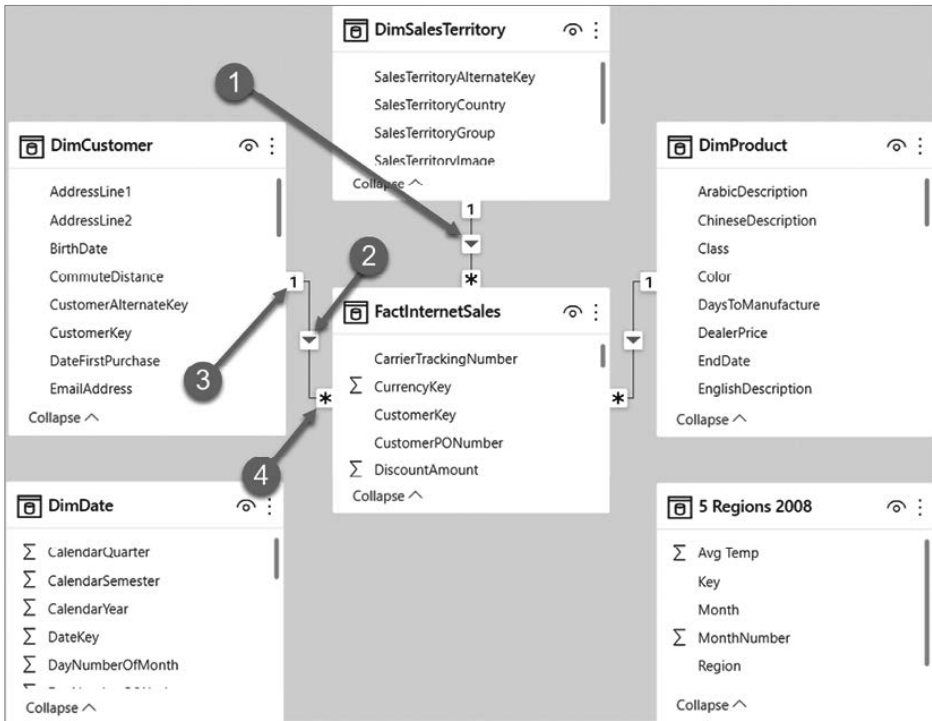
Relacje w Power BI definiują, w jaki sposób powiązane są ze sobą różne tabele. Na przykład tabela Customers może być powiązana z tabelą Sales na podstawie kolumny zawierającej identyfikatory klientów. Można śmiało powiedzieć, że budowanie relacji jest najważniejszym aspektem pracy w Power BI Desktop. To właśnie proces budowania relacji sprawia, że wszystko inne w Power BI działa jak magia. Mechanizmy takie jak automatyczne filtrowanie wizualizacji i raportów, łatwość, z jaką można tworzyć miary i wskaźniki za pomocą **Data Analysis Expressions (DAX)**, oraz szybkie łączenie różnych źródeł danych są możliwe dzięki prawidłowo zbudowanym relacjom w modelu danych.

Czasami Power BI Desktop tworzy relacje automatycznie. Pamiętaj jednak, że powinieneś zawsze zweryfikować takie **automatycznie wykryte relacje** (ang. *auto-detected relationships*), aby upewnić się, że są poprawne.

Istnieje kilka cech relacji, o których należy pamiętać, a które zostaną omówione w tym rozdziale:

- Niektóre relacje mogą zostać zidentyfikowane automatycznie.
- Pomiędzy dwiema tabelami może istnieć tylko jedna aktywna relacja.
- Pomiędzy tabelami może istnieć nieograniczona liczba nieaktywnych relacji.
- Relacje mogą być budowane tylko na pojedynczej kolumnie, nie na wielu kolumnach.
- Relacje automatycznie filtrują się w kierunku od strony *jeden* do strony *wiele*.

Teraz, aby zbadać niektóre z kluczowych elementów relacji w Power BI, otwórz plik *.pbix* o nazwie *Chapter 4 — Building the Data Model.pbix*, znajdujący się w zasobach książki (rysunek 4.1).



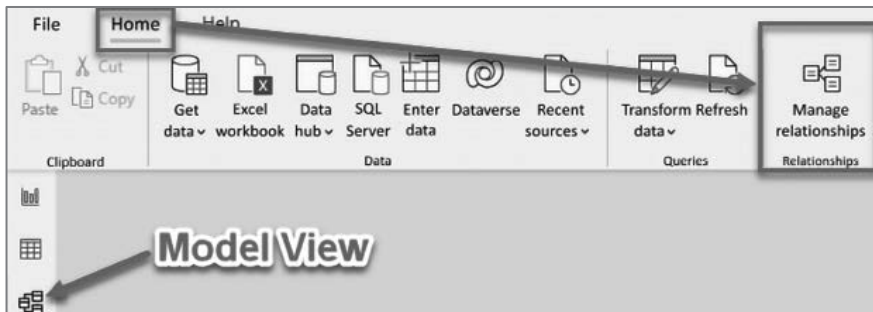
Rysunek 4.1. Przegląd elementów relacji

Przyjrzyjmy się teraz bliżej każdemu z czterech elementów wyróżnionych na rysunku 4.1:

- **Relacja** — linia pomiędzy dwiema tabelami oznacza, że istnieje między nimi relacja.
- **Kierunek** — strzałka wskazuje, w którym kierunku następuje filtrowanie.
- **Strona „jeden”** — cyfra 1 wskazuje, że tabela Customer reprezentuje *stronę jeden* relacji. Oznacza to, że wartości klucza *strony jeden* relacji w tej tabeli są zawsze unikatowe.
- **Strona „wiele”** — znak gwiazdki * wskazuje, że tabela FactInternetSales reprezentuje *stronę wiele* tej relacji. Klucz (identyfikator klienta) zapisywany jest w tabeli sprzedaży dla każdej transakcji, dlatego taka sama wartość klucza może pojawiać się w tabeli sprzedaży wiele razy.

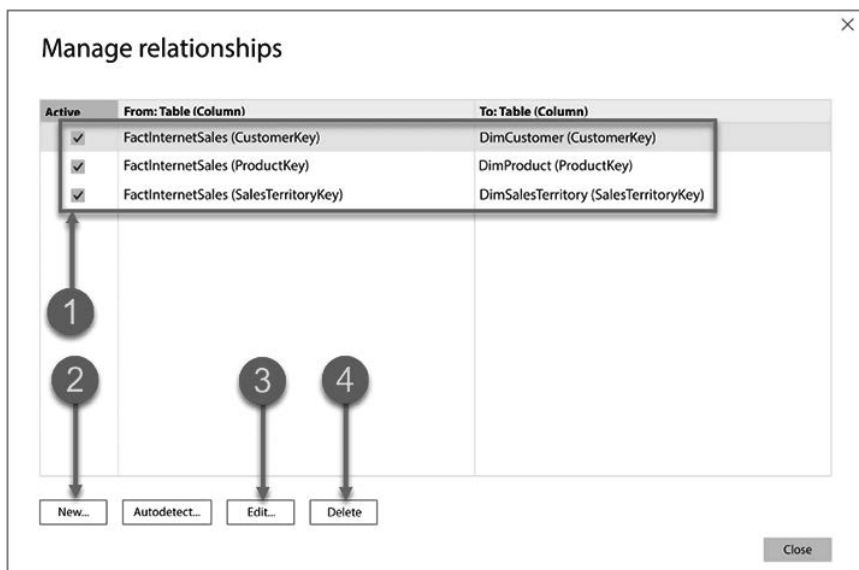
Pierwszą rzeczą, jaką powinieneś zrobić po zaimportowaniu danych, jest sprawdzenie, czy wszystkie automatycznie wykryte relacje zostały utworzone poprawnie. Przejdź na kartę *Home (Narzędzia główne)* i wybierz polecenie *Manage relationships (Zarządzanie relacjami)*, jak pokazano na rysunku 4.2. W widoku *Report (Raport)* polecenie *Manage relationships (Zarządzanie relacjami)* znajduje się na karcie *Modeling (Modelowanie)*. Jeżeli pracujesz w widoku *Data (Dane)*, polecenie *Manage relationships (Zarządzanie*

relacjami) znajdziesz na karcie *Table tools* (*Narzędzia tabel*), a w widoku *Model* polecenie *Manage relationships* (*Zarządzanie relacjami*) jest wyświetlane na karcie *Home* (*Narzędzia główne*).



Rysunek 4.2. Uruchamianie edytora relacji

Na ekranie pojawi się okno edytora *Manage relationships* (*Zarządzanie relacjami*). Edytor relacji jest jednym z dwóch miejsc, w których będziesz mógł tworzyć nowe relacje oraz edytować lub usuwać relacje istniejące. W tym przykładzie edytor relacji wykorzystamy do zweryfikowania relacji, które zostały automatycznie utworzone przez Power BI Desktop. Rysunek 4.3 przedstawia widok edytora *Manage relationships* (*Zarządzanie relacjami*).



Rysunek 4.3. Edytor relacji

Na rysunku oznaczono najważniejsze elementy edytora relacji:

1. Aktualne relacje w modelu danych.
2. Przycisk *New (Nowa)* pozwalający na utworzenie nowej relacji.
3. Przycisk *Edit (Edytuj)* umożliwiający edytowanie istniejących relacji.
4. Przycisk *Delete (Usuń)* pozwalający na usunięcie relacji.

Na początku powinieneś zawsze zweryfikować automatycznie wykryte relacje. W górnej części okna pokazanego na rysunku 4.3 znajdziesz listę pozwalającą szybko i łatwo sprawdzić, które tabele są powiązane relacjami, na których kolumnach zostały one utworzone oraz czy dana relacja jest relacją aktywną. Więcej szczegółowych informacji na temat relacji aktywnych i nieaktywnych znajdziesz w dalszej części tego rozdziału.

Przyjrzyj się fragmentowi okna edytora relacji, pokazanemu na rysunku 4.4; jak łatwo zauważyć, w naszym przykładzie mamy obecnie trzy relacje i wszystkie są aktywne.

Manage relationships		
Active	From: Table (Column)	To: Table (Column)
<input checked="" type="checkbox"/>	FactInternetSales (CustomerKey)	DimCustomer (CustomerKey)
<input checked="" type="checkbox"/>	FactInternetSales (ProductKey)	DimProduct (ProductKey)
<input checked="" type="checkbox"/>	FactInternetSales (SalesTerritoryKey)	DimSalesTerritory (SalesTerritoryKey)

Rysunek 4.4. Weryfikowanie relacji

Pierwszy wiersz na rysunku 4.4 wyświetla relację między tabelą FactInternetSales a tabelą DimCustomer. Relacja między tymi dwiema tabelami została utworzona automatycznie przez Power BI na podstawie kolumny CustomerKey w każdej z tabel. W tym scenariuszu aplikacja Power BI poprawnie wybrała prawidłowe nazwy kolumn. Pamiętaj jednak, że zawsze, jeżeli jakaś relacja została utworzona niepoprawnie, będziesz musiał ją zmodyfikować ręcznie.

Przyjrzyjmy się teraz, jak można edytować istniejącą relację.

Edytowanie relacji

W tym przykładzie spróbujemy zmodyfikować relację między tabelami FactInternetSales i DimCustomer. Aby edytować istniejącą relację, powinieneś zaznaczyć ją na liście, a następnie nacisnąć przycisk *Edit (Edytuj)*. Po naciśnięciu przycisku *Edit (Edytuj)* na ekranie pojawi się okno dialogowe *Edit relationship (Edytowanie relacji)* (rysunek 4.5).

Edit relationship

Select tables and columns that are related.

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
528	20071229	20080110	20080105	11024	1	100	
528	20070910	20070922	20070917	11049	1	100	
528	20080623	20080705	20080630	11086	1	100	

DimCustomer

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	Na
11602	135	AW00011602		Larry		Gill	
11603	244	AW00011603		Geoffrey		Gonzalez	
11610	269	AW00011610		Blake		Collins	

Cardinality: Many to one (*:1)

Cross filter direction: Single

Make this relationship active

Assume referential integrity

Apply security filter in both directions

OK Cancel

Rysunek 4.5. Edytowanie relacji

W tym widoku możesz zmodyfikować istniejącą relację, oznaczyć relację jako aktywną lub nieaktywną oraz zmienić zarówno kardynalność bieżącej relacji (ang. *cardinality*), jak i kierunek filtrowania krzyżowego (ang. *cross filter direction*), tak jak pokazano na rysunku 4.6.

Na rysunku 4.6 pokazano najważniejsze elementy okna edytowania relacji.

1. Elementy identyfikujące tabelę FactInternetSales i kolumnę, na której zbudowana została relacja, CustomerKey.
2. Te dwa elementy identyfikują tabelę DimCustomer i kolumnę, na której zbudowano relację, również CustomerKey.
3. Pole wyboru wskazujące, czy relacja jest aktywna czy nieaktywna.
4. Pole wskazujące kardynalność relacji między tymi dwiema tabelami. W naszym przykładzie możemy zobaczyć, że pomiędzy tabelami FactInternetSales i DimCustomer istnieje relacja typu *Wiele do jednego* (*:1). Power BI doskonale radzi sobie z identyfikowaniem poprawnej kardynalności, ale mimo to zawsze powinieneś sprawdzać, czy sugerowana kardynalność relacji jest poprawna.

Edit relationship ✕

Select tables and columns that are related.

1

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
528	20071229	20080110	20080105	11024	1	100	
528	20070910	20070922	20070917	11049	1	100	
528	20080623	20080705	20080630	11086	1	100	

2

DimCustomer

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	Nan
11602	135	AW00011602		Larry		Gill	
11603	244	AW00011603		Geoffrey		Gonzalez	
11610	269	AW00011610		Blake		Collins	

3

Cardinality

Many to one (*:1)

4

5

Cross filter direction

Single

Make this relationship active

Assume referential integrity

Apply security filter in both directions

OK Cancel

Rysunek 4.6. Edytowanie relacji

5. Kierunek filtrowania krzyżowego może być ustawiony jako *Single (Jeden)* lub *Both (Oba)*. Strona *jeden* relacji zawsze filtruje stronę *wiele* i jest to domyślne zachowanie w Power BI. Opcja wyboru kierunku filtrowania krzyżowego pozwala na zmianę tego zachowania. Filtrowanie krzyżowe zostanie omówione w dalszej części tego rozdziału.

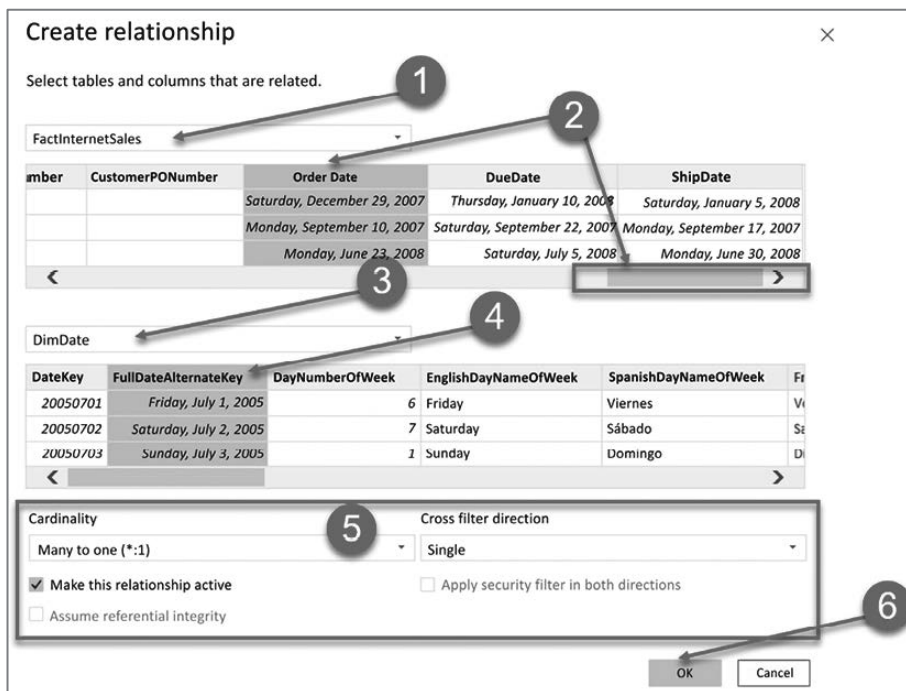
Jeżeli musisz zmienić ustawienia istniejącej relacji, możesz zrobić to w edytorze relacji pokazanym na rysunku 4.6. Aby zmienić kolumnę, dla której została utworzona relacja, wystarczy wybrać inną kolumnę. Pamiętaj, że relacja między dwiema tabelami może być utworzona tylko dla jednej kolumny. Z tego powodu, jeżeli masz relację, która musi być określona na wielu kolumnach (znaną również jako klucz złożony), musisz najpierw, przed utworzeniem relacji, połączyć poszczególne klucze w jedną kolumnę. O tym, jak łączyć kolumny pisaliśmy już w rozdziale 3., „Strategie przekształcania danych”.

Tworzenie nowej relacji

W poprzednim punkcie dowiedziałeś się, jak weryfikować i edytować istniejące relacje. W tym punkcie nauczysz się tworzyć nowe relacje. W naszym przykładowym modelu danych znajduje się sześć tabel, a Power BI automatycznie utworzył relację dla czterech z nich. Zacznijmy od utworzenia relacji między tabelami FactInternetSales i DimDate.

Tabela FactInternetSales przechowuje trzy różne daty: OrderDate, ShipDate i DueDate. W Power BI pomiędzy dwiema tabelami może być tylko jedna aktywna relacja i całe filtrowanie odbywa się poprzez tę relację. Musisz zatem się zastanowić, na podstawie której daty chcesz przeprowadzić obliczenia dotyczące całkowitej sprzedaży, zysku i marży. Jeżeli będzie to OrderDate, wtedy taka relacja będzie utworzona między kolumną OrderDate z tabeli FactInternetSales a kolumną FullDateAlternateKey z tabeli DimDate. Aby utworzyć nową relację, przejdź na kartę *Home* (*Narzędzia główne*) i naciśnij przycisk *Manage relationships* (*Zarządzanie relacjami*).

Teraz utworzymy relację z kolumny OrderDate w tabeli FactInternetSales do kolumny FullDateAlternateKey w tabeli DimDate. Aby to zrobić, w edytorze zarządzania relacjami naciśnij przycisk *New* (*Nowa*). Na ekranie pojawi się okno dialogowe edytora *Create relationship* (*Tworzenie relacji*) (rysunek 4.7).



Rysunek 4.7. Tworzenie nowej relacji

Aby utworzyć nową relację, powinieneś wykonać następujące polecenia:

1. Z pierwszej listy rozwijanej wybierz tabelę `FactInternetSales`.
2. Z listy kolumn wybierz `OrderDate` (użyj paska przewijania, aby przewinąć listę do końca w prawo).
3. Z drugiej listy rozwijanej wybierz tabelę `DimDate`.
4. Z listy kolumn wybierz kolumnę `FullDateAlternateKey`.
5. Opcje *Cardinality (Kardynalność)*, *Cross filter direction (Kierunek filtrowania krzyżowego)* oraz to, czy relacja jest aktywna, czy nieaktywna, zostaną automatycznie ustawione przez Power BI; pamiętaj jednak, aby zawsze zweryfikować te elementy.
6. Naciśnij przycisk *OK*, aby zakończyć tworzenie relacji i zamknąć edytor.

Teraz przyjrzyjmy się tworzeniu relacji na kluczu reprezentującym datę zamiast na kolumnie `Data`.

Tworzenie relacji na kluczu reprezentującym datę

Z pewnością zauważyłeś, że w poprzednim przykładzie w każdej z tabel używaliśmy kolumn zawierających rzeczywiste daty zamiast kluczy reprezentujących daty. Zrobiliśmy tak dlatego, że w większości modeli Power BI po prostu nie będzie klucza reprezentującego daty. Jeżeli jednak pobierasz dane z relacyjnej bazy danych lub hurtowni danych, to klucz daty najprawdopodobniej będzie istniał w obu tabelach i wtedy relacja może zostać oparta na kluczu daty.

Uwaga

W modelowaniu danych daty z zasady są zapisywane w postaci liczby całkowitej. Dokładniej mówiąc, możesz przechowywać datę w postaci **inteligentnego klucza** (ang. *smart key*), na przykład 20200706. Ten typ wartości całkowitej jest tak nazywany, ponieważ data jest zapisywana jako liczba całkowita, ale wciąż możesz z niej wyprowadzić datę. Na przykład cztery pierwsze znaki to rok, dwa kolejne to miesiąc, a dwa ostatnie to dzień miesiąca. Zapisywanie dat w postaci liczb całkowitych jest sprawdzonym sposobem, który pomoże Ci zaoszczędzić miejsce w modelu.

Na rysunku 4.8 pokazano przykład tworzenia relacji na kluczach daty w każdej z tabel.

Tabela daty w modelu danych Power BI jest ważna przy tworzeniu obliczeń z analizą czasową (ang. *time intelligence*), czyli wykorzystujących możliwości filtrowania i obsługiwaną dat w modelu analitycznym lub narzędziu raportowym. W przypadku definiowania relacji na kluczu daty, a nie na kolumnie daty, ważne jest, aby oznaczyć tabelę zawierającą daty jako **tabelę dat** (ang. *date table*)! Jeżeli ten krok nie zostanie wykonany, wbudowane funkcje *time intelligence* nie będą działać zgodnie z założeniem.

FactInternetSales							
ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
528	20071229	20080110	20080105	11024	1	100	
528	20070910	20070922	20070917	11049	1	100	
528	20080623	20080705	20080630	11086	1	100	

DimDate				
DateKey	Date	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek
20050701	Friday, July 1, 2005	6	Friday	Viernes
20050702	Saturday, July 2, 2005	7	Saturday	Sábado
20050703	Sunday, July 3, 2005	1	Sunday	Domingo

Rysunek 4.8. Tworzenie nowej relacji opartej na kluczach daty

Uwaga

Funkcje analizy czasowej zostały omówione bardziej szczegółowo w rozdziale 5., „Wykorzystanie języka DAX”.

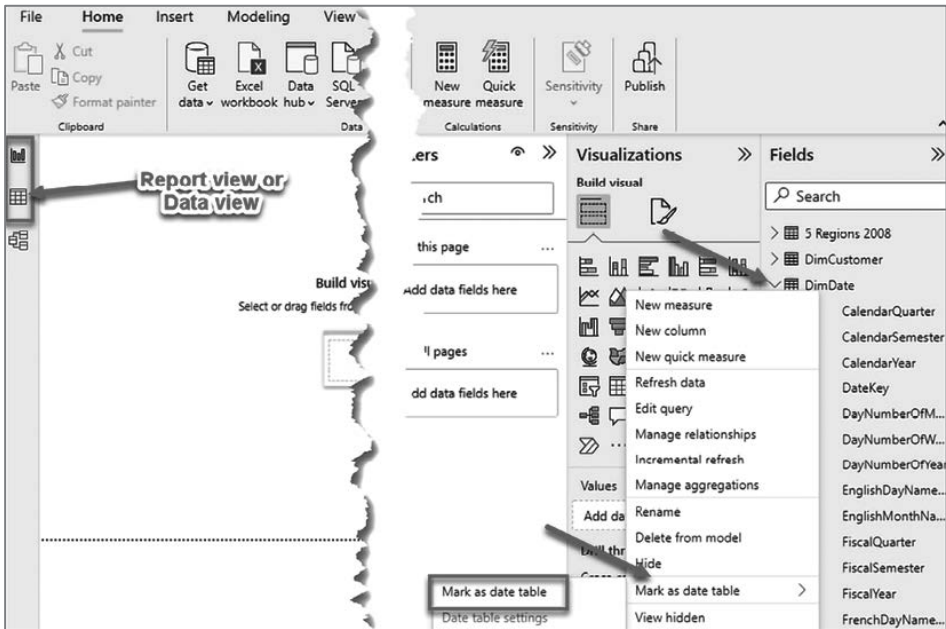
W chwili kiedy powstawała ta książka, oznaczenia tabeli jako tabeli dat można było wykonać tylko w widoku *Report (Raport)* lub w widoku *Data (Dane)*. Istnieją dwa sposoby oznaczenia tabeli jako tabeli dat. Pierwszy sposób polega na kliknięciu prawym przyciskiem myszy tabeli zawierającej daty i wybraniu polecenia *Mark as date table/Mark as date table (Oznacz jako tabelę dat/Oznacz jako tabelę dat)*, tak jak to zostało pokazane na rysunku 4.9.

Zamiast tego możesz również zaznaczyć tabelę, przejść na kartę *Table tools (Narzędzia tabel)* i wybrać polecenie *Mark as date table (Oznacz jako tabelę dat)*, tak jak to zostało pokazane na rysunku 4.10.

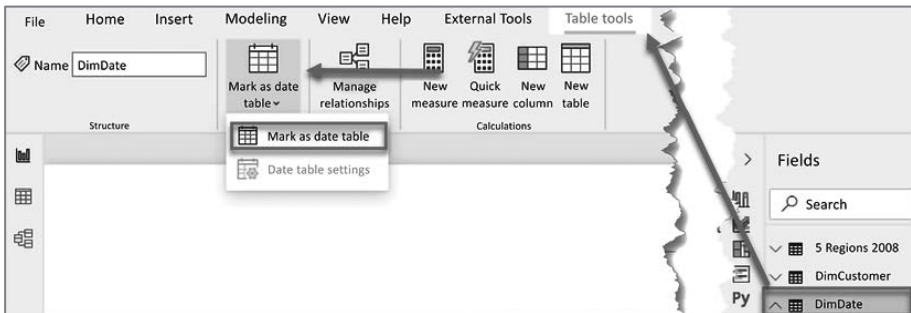
Po wybraniu opcji *Mark as date table (Oznacz jako tabelę dat)* na ekranie pojawi się okno dialogowe, w którym z listy rozwijanej *Date column (Kolumna dat)* powinieneś wybrać kolumnę, w której przechowywane są daty, co zostało pokazane na rysunku 4.11.

Power BI automatycznie ograniczy listę dostępnych opcji do kolumn, które są ustawione na typ daty i zawierają tylko unikatowe wartości.

Jest jeszcze jedna ważna rzecz, którą warto zauważyć: wbudowane tabele dat, które były powiązane z tą tabelą, zostały usunięte. Power BI tworzy w tle ukrytą tabelę dat dla każdego pola typu *date* lub *date/time* (data lub data/czas). W zależności od liczby pól daty w modelu danych może to spowodować utworzenie wielu dodatkowych obiektów

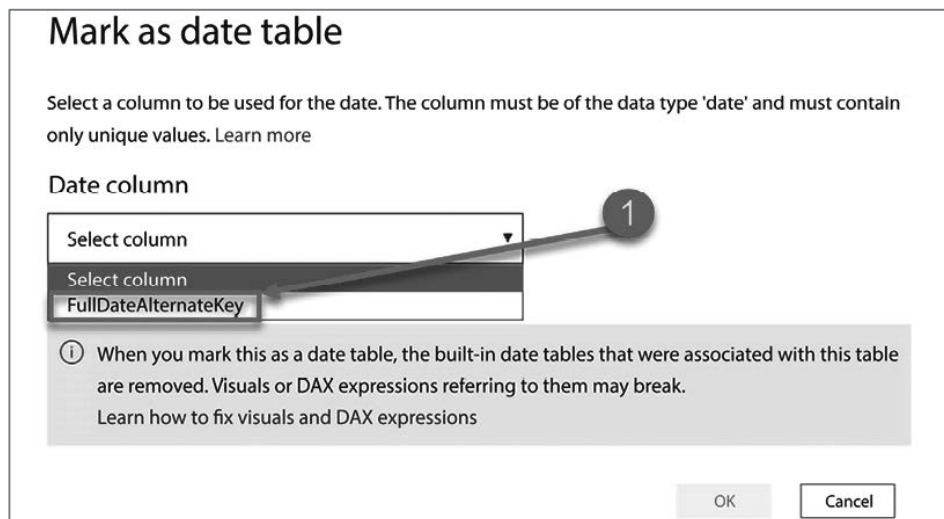


Rysunek 4.9. Oznaczenie tabeli jako tabeli dat



Rysunek 4.10. Alternatywna metoda oznaczenia tabeli jako tabeli dat

i znaczący wzrost zużycia pamięci w modelu danych! Zgodnie z dobrymi praktykami tworzenia modeli danych zalecamy wyłączenie tej funkcjonalności. Omówimy ten proces w następnej sekcji.



Rysunek 4.11. Wybieranie kolumny zawierającej daty

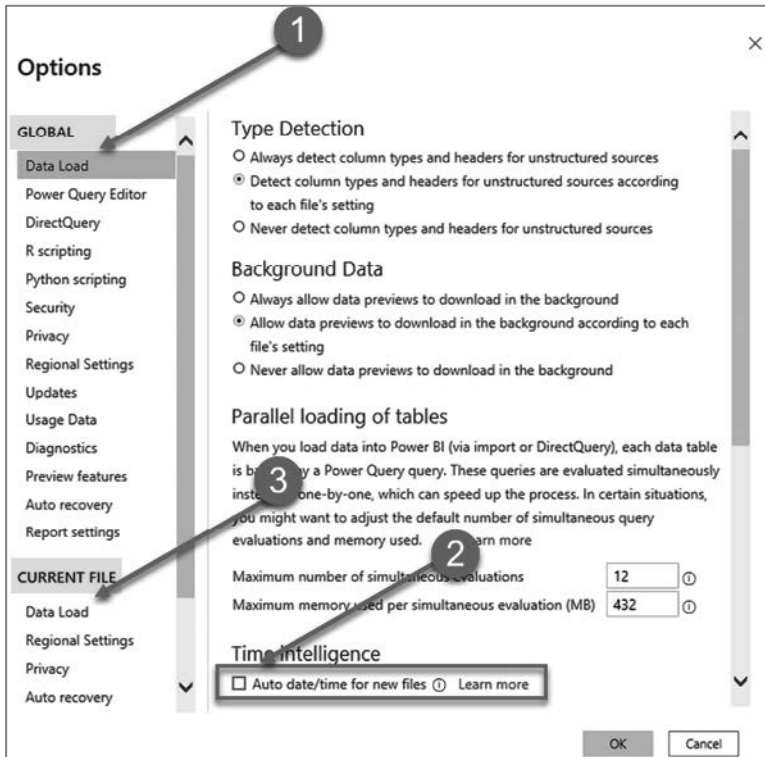
Wyłączanie automatycznie tworzonych tabel dat

Jak wspominaliśmy wcześniej, Power BI automatycznie tworzy ukryte tabele dat dla każdego pola daty lub daty/czasu w modelu danych. Tabele dat tworzone przez Power BI mogą być wyłączone z poziomu menu *Options (Opcje)*. Ta funkcja jest automatycznie włączona w Twoim bieżącym pliku Power BI i będzie włączona również we wszystkich kolejnych. Dlatego musisz wyłączyć automatyczne tworzenie tabel dat w dwóch miejscach w menu *Options (Opcje)*.

Aby otworzyć to menu w Power BI Desktop, powinieneś przejść na kartę *File (Plik)* i wybrać polecenie *Options and Settings/Options (Opcje i ustawienia/Opcje)*. Kiedy na ekranie pojawi się okno dialogowe *Options (Opcje)*, wykonaj polecenia opisane poniżej i pokazane na rysunku 4.12.

1. W sekcji *GLOBAL (GLOBALNE)* zaznacz kategorię *Data Load (Ładowanie danych)*.
2. Usuń zaznaczenie opcji *Auto date/time for new files (Automatyczna data/godzina dla nowych plików)*.
3. W sekcji *CURRENT FILE (BIEŻĄCY PLIK)* wybierz kategorię *Data Load* i odznacz opcję *Auto date/time*.

W tej sekcji dowiedziałeś się, jak weryfikować automatycznie utworzone relacje oraz jak tworzyć nowe relacje w Power BI Desktop. W następnym podrozdziale dowiesz się, jak pracować z relacjami złożonymi.



Rysunek 4.12. Wyłączenie opcji automatycznego tworzenia ukrytej tabeli dat

Praca z relacjami złożonymi

Istnieje wiele skomplikowanych zagadnień, które często należy uwzględnić podczas budowania modelu danych, a Power BI nie odbiega pod tym względem od innych rozwiązań. W tym podrozdziale dowiesz się, jak obsługiwać relacje typu **wiele-do-wielu** (ang. *many-to-many*) oraz tabele odgrywające role w Power BI.

Relacje typu wiele-do-wielu

Zanim pokażemy, jak zbudować relację typu wiele-do-wielu między dwiema tabelami, musimy najpierw powiedzieć, czym jest taka relacja. Zasadniczo relacja typu wiele-do-wielu powstaje wtedy, gdy wiele wierszy w jednej tabeli jest powiązanych z wieloma wierszami w innej tabeli. Przykład relacji typu wiele-do-wielu można zaobserwować między produktami i klientami. Każdy produkt może być sprzedany wielu klientom, podobnie każdy klient może zakupić wiele produktów. Taka relacja między produktami i klientami jest relacją typu wiele-do-wielu. Relacja typu **jeden-do-wielu** (ang. *one-to-many*)

może być utworzona bezpośrednio między dwiema tabelami. Jednak w przypadku relacji typu wiele-do-wielu często tworzona jest relacja pośrednia poprzez tabelę pomostową. W tym podrozdziale skupimy się na tym, jak tworzyć relacje typu wiele-do-wielu oraz jak używać takich relacji podczas analizy danych.

Relację wiele-do-wielu spotykamy częściej w modelach z dwoma osobnymi tabelami faktów, które mają podobny klucz. Na przykład jeśli masz tabele FactSale i FactReturns, to mogą zawierać one klucz produktu. Jak możesz z perspektywy analityki utworzyć raport pokazujący sprzedaż i zwroty według lat, według klientów lub według sklepów? Najprostsze rozwiązanie polega na wykorzystaniu obu tabel wymiarów. Bardziej szczegółowe wejście w schemat gwiazdy wykracza poza zakres tej książki, jednak w modelach danych Power BI często spotyka się wiele tabel faktów powiązanych ze sobą relacjami wiele-do-wielu.

Uwaga

Aby dowiedzieć się więcej o relacjach wiele-do-wielu, odwiedź stronę firmy Microsoft <https://learn.microsoft.com/pl-pl/power-bi/guidance/relationships-many-to-many>.

W poprzednim podrozdziale dowiedziałeś się, jak między dwiema tabelami utworzyć relację typu jeden-do-wielu. Natychmiast po utworzeniu takiej relacji w modelu danych rozpoczyna się automatycznie filtrowanie danych, co znakomicie wpływa na efektywność Power BI. Warto jednak pamiętać, że w przypadku relacji typu wiele-do-wielu wzrost wartości analitycznej nie odbywa się automatycznie i wymaga wykonania dodatkowego kroku. Przyjrzyjmy się zatem zagadnieniom związanym z filtrowaniem danych, a następnie tworzeniem efektywnych relacji typu wiele-do-wielu w Power BI.

Działanie filtra

Zanim dowiesz się, jak obsługiwać relacje typu wiele-do-wielu w Power BI, musisz najpierw zrozumieć sposób działania mechanizmu filtrowania. Co prawda filtrowanie zostanie omówione bardziej szczegółowo w następnym rozdziale, ale już teraz zrobimy małą dygresję, aby pokrótce wyjaśnić, jak to działa w tym kontekście. Na rysunku 4.13 pokazano, że suma SalesAmount wszystkich transakcji wynosi 29 358 677,22 dolara. Wizualizacja tabeli, którą widzisz na tym rysunku, jest po prostu sumą wartości kolumny SalesAmount z tabeli FactInternetSales.

SalesAmount
29,358,677.22

Rysunek 4.13. Suma całkowita sprzedaży z niefiltrowanej kolumny SalesAmount

Aby wyświetlić całkowite sumy sprzedaży SalesAmount dla wszystkich transakcji podzielonych na poszczególne kraje, musisz po prostu dodać kolumnę SalesTerritoryCountry z tabeli DimSalesTerritory. To naprawdę niesamowita możliwość Power BI, wynikająca z działania automatycznego mechanizmu filtrowania. Na rysunku 4.14 możesz zobaczyć, jak to wygląda w praktyce.

SalesTerritoryCountry	SalesAmount
Australia	9,061,000.58
Canada	1,977,844.86
France	2,644,017.71
Germany	2,894,312.34
United Kingdom	3,391,712.21
United States	9,389,789.51
Total	29,358,677.22

Rysunek 4.14. Sumy sprzedaży (SalesAmount) według krajów (SalesTerritoryCountry)

Działa to tylko dlatego, że istnieje ważna relacja między tabelami FactInternetSales i DimSalesTerritory. Jeżeli taka relacja nie została wcześniej utworzona lub jeżeli utworzona relacja byłaby nieprawidłowa, wówczas otrzymalibyśmy zupełnie inne, mylne wyniki. Zobaczmy zatem, co by się stało, gdyby wcześniej między tymi dwiema tabelami nie istniała żadna relacja.

Na rysunku 4.15 kolumna SalesTerritoryCountry została usunięta i zastąpiona kolumną Temperature Range z tabeli 5 Regions 2008.

Temperature Range	SalesAmount
Cold	29,358,677.22
Cool	29,358,677.22
Hot	29,358,677.22
Warm	29,358,677.22
Total	29,358,677.22

Rysunek 4.15. Kolumna SalesTerritoryCountry została zastąpiona kolumną Temperature Range

Zauważ, że całkowita kwota sprzedaży została powtórzona dla każdego zakresu temperatur. To zachowanie wskazuje, że tabela 5 Regions 2008 nie jest w stanie filtrować tabeli FactInternetSales. Taka niezdolność do filtrowania może mieć wiele różnych powodów:

- Nie istnieje odpowiednio zdefiniowana relacja między tabelami.
- Istniejąca relacja jest niepoprawna.
- Istniejąca relacja nie pozwala na wykonanie filtrowania przez tabelę pośrednią.

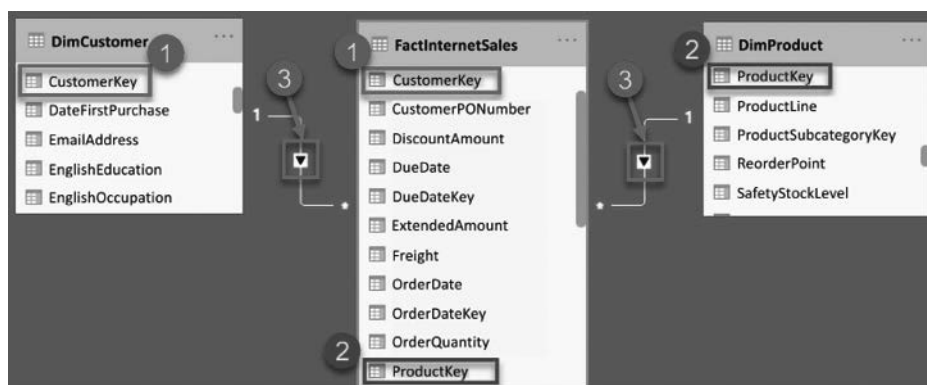
Jeżeli zaobserwujesz zachowanie (powtarzające się wartości) pokazane na rysunku 4.15, powinieneś wrócić do widoku relacji i sprawdzić, czy wszystkie niezbędne relacje zostały utworzone i czy są poprawne.

Kierunek filtrowania krzyżowego

Teraz, gdy rozumiesz już podstawy automatycznego filtrowania w Power BI, przyjrzymy się przykładowi relacji typu wiele-do-wielu. W naszym przykładowym modelu danych tabele `DimProduct` i `DimCustomer` mają relację typu wiele-do-wielu, ponieważ każdy produkt może być sprzedawany wielu klientom (na przykład chleb może być sprzedany Jessice, Kim i Kate), a każdy klient może kupić wiele produktów (na przykład Kim może kupić chleb, mleko i ser).

Tabela pomostowa może być użyta do przechowywania relacji między dwiema tabelami, które mają relację typu wiele-do-wielu, podobnie jak narzędzia, z którymi pracowałeś w przeszłości.

Relacja między tabelami `DimProduct` i `DimCustomer` jest przechowywana w tabeli `FactInternetSales`. Tabela `FactInternetSales` jest dużą tabelą pomostową typu wiele-do-wielu, jak to zostało pokazane na rysunku 4.16.



Rysunek 4.16. Relacje między tabelami `DimCustomer` i `FactInternetSales`

Na rysunku 4.16 przedstawiono relację między tymi dwiema tabelami:

1. Relacja między tabelami `DimCustomer` i `FactInternetSales`.
2. Relacja między tabelami `DimProduct` i `FactInternetSales`.
3. Kierunek filtrowania krzyżowego jest ustawiony na *Jeden*.

Raport przedstawiony na rysunku 4.17 wyświetla całkowitą wartość sprzedaży, całkowitą liczbę transakcji i liczbę klientów dla poszczególnych produktów.

1 EnglishProductName	2 Total Sales	3 Total Transactions	4 Customer Count
Adjustable Race			18484
All-Purpose Bike Stand	\$39,591.00	249	18484
AWC Logo Cap	\$19,688.10	2190	18484
BB Ball Bearing			18484
Bearing Ball			18484
Bike Wash - Dissolver	\$7,218.60	908	18484
Blade			18484

Rysunek 4.17. Liczba klientów dla poszczególnych produktów

Przyjrzyj się bliżej temu rysunkowi i zwróć uwagę na ponumerowane elementy:

1. Kolumna `EnglishProductName` z tabeli `DimProduct`.
2. Kolumna `Total Sales` jest sumą wartości kolumny `SalesAmount` z tabeli `FactInternetSales`.
3. Kolumna `Total Transactions` reprezentuje liczbę powiązanych wierszy z tabeli `FactInternetSales`.
4. Kolumna `Customer Count` reprezentuje liczbę wierszy `CustomerKey` z tabeli `DimCustomer`.

Kolumny `Total Sales` i `Total Transactions` zwracają poprawne wyniki dla każdego produktu, jednak kolumna `Customer Count` zwraca tę samą wartość dla wszystkich produktów (18,484). Jest to spowodowane sposobem, w jaki działa filtrowanie. Obliczenia dla `Total Sales` i `Total Transactions` są oparte na kolumnach lub wierszach z tabeli `FactInternetSales`. Tabela `Product` ma relację typu jeden-do-wielu z tabelą `FactInternetSales`, a zatem filtrowanie odbywa się automatycznie. To wyjaśnia, dlaczego te dwie pierwsze kolumny są filtrowane poprawnie, ale nie wyjaśnia, dlaczego w kolumnie reprezentującej liczbę klientów zwracana jest ta sama powtarzająca się wartość dla wszystkich produktów (a w każdym razie nie wyjaśnia tego całkowicie).

Przyjrzyjmy się jeszcze raz relacji między tabelami `DimProduct` i `DimCustomer`. Na rysunku 4.18 wyraźnie widać, że relacja między tymi dwiema tabelami przepływa przez tabelę `FactInternetSales`. Dzieje się tak, ponieważ mają one relację typu wiele-do-wielu. W tym scenariuszu tabela `FactInternetSales` działa jak duża tabela pomocowa typu wiele-do-wielu. Tabela `DimProduct` filtruje tabelę `FactInternetSales`. Tabela `DimCustomer` również filtruje `FactInternetSales`, a tabela `FactInternetSales` nie jest obecnie w stanie filtrować tabeli klientów (zobacz rysunek 4.18).



Rysunek 4.18. Działanie filtrowania w Power BI

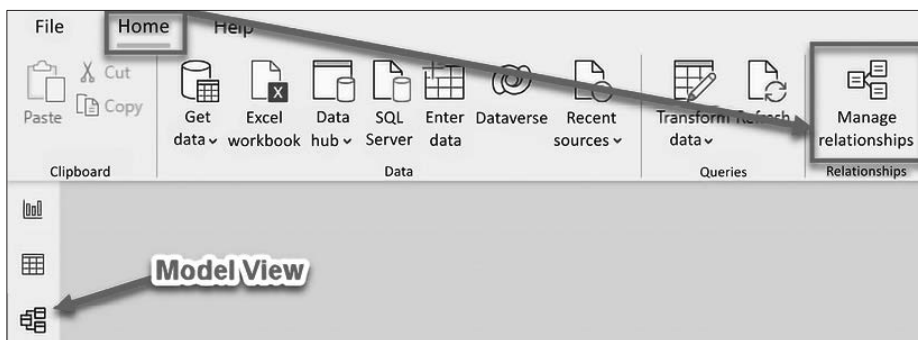
Powtarzająca się wartość liczby klientów pojawia się w każdym wierszu, ponieważ tabela FactInternetSales nie jest w stanie filtrować tabeli DimCustomer. Tabela DimProduct filtruje tabelę FactInternetSales, a lista transakcji jest zwracana dla każdego produktu. Niestety filtrowanie nie przechodzi z FactInternetSales do DimCustomer. Dzieje się tak dlatego, że tabela FactInternetSales jest po stronie *wiele* relacji z tabelą DimCustomer, stąd kiedy nasze obliczenia dokonują zliczania kluczy klienta, tabela nie jest filtrowana i obliczenia „widzą” każdy klucz klienta w tabeli DimCustomer (czyli 18,484 wiersza).

Czy pamiętasz właściwość *Cross filter direction* (*Kierunek filtrowania krzyżowego*), którą krótko omawialiśmy wcześniej w tym rozdziale? Ta niepozorna właściwość służy do zapewnienia obsługi relacji typu *wiele-do-wiele*. Po ustawieniu filtrowania w obu kierunkach, czyli, inaczej mówiąc, po ustawieniu właściwości *Cross filter direction* (*Kierunek filtrowania krzyżowego*) na opcję *Both* (*Oba*), tabela FactInternetSales będzie mogła filtrować tabelę klientów, a zliczanie klientów będzie działało poprawnie.

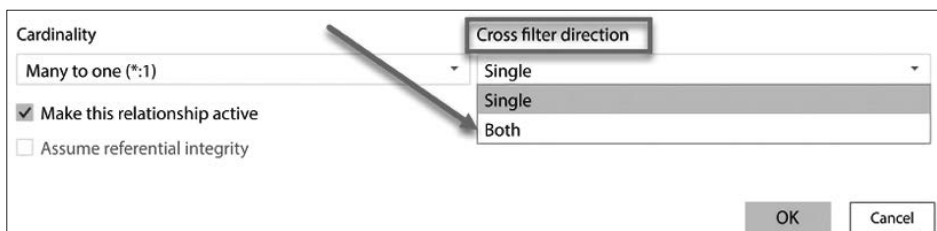
Włączanie filtrowania po stronie wiele

Aby włączyć filtrowanie krzyżowe, powinieneś otworzyć edytor relacji. Pamiętaj, że polecenie *Manage relationships* (*Zarządzanie relacjami*) możesz znaleźć w widoku raportu, widoku danych lub widoku modelu, na różnych kartach menu, tak jak zostało to omówione na początku tego rozdziału. Jeżeli pracujesz w widoku raportu, polecenie *Manage relationships* (*Zarządzanie relacjami*) pojawi się na karcie *Modeling* (*Modelowanie*); jeżeli jesteś w widoku danych, polecenie *Manage relationships* (*Zarządzanie relacjami*) będzie widoczne na karcie *Table tools* (*Narzędzia tabeli*), a jeżeli znajdujesz się w widoku modelu, polecenie zarządzania relacjami będzie dostępne na karcie *Home* (*Narzędzia główne*). Zobacz rysunek 4.19, aby dowiedzieć się, gdzie można je znaleźć w widoku modelu.

Zaznacz relację między tabelami FactInternetSales i DimCustomer, a następnie naciśnij przycisk *Edit* (*Edytuj*). Po uruchomieniu edytora relacji zmień ustawienia właściwości *Cross filter direction* (*Kierunek filtrowania krzyżowego*) z *Single* (*Jeden*) na *Both* (*Oba*), tak jak to pokazane zostało na rysunku 4.20.



Rysunek 4.19. Otwieranie edytora relacji



Rysunek 4.20. Włączanie filtrowania krzyżowego w obu kierunkach

Po powrocie do widoku raportu przekonasz się, że liczby klientów dla poszczególnych produktów (kolumna Customer Count) wyświetlane są prawidłowo, jak pokazano na rysunku 4.21.

EnglishProductName	Total Sales	Total Transactions	Customer Count
All-Purpose Bike Stand	\$39,591.00	249	243
AWC Logo Cap	\$19,688.10	2190	2132
Bike Wash - Dissolver	\$7,218.60	908	875
Classic Vest, L	\$12,382.50	195	195
Classic Vest, M	\$12,636.50	199	199
Classic Vest, S	\$10,668.00	168	168
Fender Set - Mountain	\$46,619.58	2121	2110

Rysunek 4.21. Poprawne zliczenie liczby klientów dla poszczególnych produktów

Uwaga

Zgodnie z dobrymi praktykami tworzenia modeli danych włączanie filtrowania krzyżowego w modelu danych *nie jest* zalecane. Filtrowanie krzyżowe może spowodować niejednoznaczne wyniki, a niektóre funkcje analizy czasowej mogą nie działać poprawnie. Tabela dat musi mieć ciągły zakres dat i dlatego nie może być filtrowana przez inne tabele. Korzystaj z języka DAX, gdy tylko to możliwe, by uzyskać rezultat widoczny na rysunku 4.21. Język DAX zostanie omówiony w rozdziale 5., „Wykorzystanie języka DAX”.

Dowiedziałeś się już, jak w aplikacji Power BI Desktop można obsługiwać relacje typu wiele-do-wielu, zatem możemy przejść do omawiania innych rodzajów relacji złożonych. W kolejnym punkcie dowiesz się, czym są tabele odgrywające role (ang. *role-playing tables*) i jak je obsługiwać w Power BI Desktop.

Tabele odgrywające role

Tabela odgrywająca role (ang. *role-playing table*) to tabela, która może pełnić wiele ról, co może mieć znaczący wpływ na zmniejszenie redundancji danych. Najczęściej tabelą odgrywającą role jest tabela dat. Na przykład tabela FactInternetSales posiada trzy kolumny daty, które śledzą proces realizacji zamówienia. Jest tam data zamówienia (Order Date), data wysyłki (Ship Date) i termin dostawy (Due Date). Bez tabeli odgrywającej role musiałbyś mieć trzy oddzielne tabele z datami zamiast jednej. Dodatkowe tabele pochłaniają również cenne zasoby, takie jak pamięć, a także wymagają dodatkowej warstwy obsługi administracyjnej, co może być bardzo istotne zwłaszcza w przypadku większych projektów.

Każda z wymienionych wyżej dat jest bardzo ważna dla różnych grup użytkowników w danej firmie czy organizacji. Na przykład dział finansowy z pewnością będzie zainteresowany informacjami o całkowitej sprzedaży i zyskach, podzielonych według daty zakupu produktów i daty zamówienia. Z kolei dla działu spedycji istotne będą dane reprezentujące ilości i rodzaje wysyłanych produktów podzielone według daty wysyłki. Jak zatem możemy uwzględnić żądania z różnych działów w jednym modelu danych?

Jedną z rzeczy, którą uwielbiałem w pracy z SQL Server Analysis Services, była łatwość, z jaką obsługiwał on tabele odgrywające role — być może również pracowałeś kiedyś z podobnymi narzędziami. Niestety tabele odgrywające role nie są natywnie obsługiwane w Power BI, ponieważ wszystkie operacje filtrowania w Power BI odbywają się poprzez aktywną relację, a jak pamiętasz, między dwiema tabelami może istnieć tylko jedna aktywna relacja.

Istnieją dwa sposoby korzystania z tabel odgrywających rolę w Power BI:

- Wielokrotne zaimportowanie danej tabeli i utworzenie aktywnej relacji dla każdej instancji tej tabeli.
- Użycie języka DAX i nieaktywnych relacji do utworzenia kalkulacji, które wykorzystują obliczenia według różnych dat.

Pierwszym sposobem, a zarazem metodą, której tutaj użyjemy, jest wielokrotne zaimportowanie tabeli. Tak, oczywiście oznacza to, że nasze rozwiązanie będzie wykorzystywało więcej zasobów. Model danych będzie posiadał trzy tabele dat, po jednej tabeli do obsługi każdego rodzaju daty w tabeli FactInternetSales. Każda tabela dat będzie miała jedną aktywną relację z tabelą FactInternetSales.

Rozwiązanie wykorzystujące wielokrotne importowanie tabeli posiada całkiem sporo zalet, m.in.:

- Przeszkolenie użytkowników i zapoznanie ich z modelem danych staje się znacznie łatwiejsze. Na przykład jeżeli chcemy zobaczyć informacje o sprzedaży i zyskach według daty wysyłki, do utworzenia raportu używamy po prostu tabeli dat wysyłki.
- Zdecydowana większość, a może nawet wszystkie miary DAX będą działać dla wszystkich tabel dat, więc nie będzie potrzeby tworzenia nowych miar. Tutaj wyjątek stanowią Twoje obliczenia analityki czasowej, gdyż będą musiały być przebudowane dla każdej tabeli dat.
- Możliwość umieszczania różnych dat w macierzy zwiększa wartość analityczną rozwiązania. Przykładem może być raport prezentujący wartości sprzedaży według dat zamówień i dat wysyłki, pokazany na rysunku 4.22.

Year	2005 (ShipDate)	2006 (ShipDate)	2007 (ShipDate)	2008 (ShipDate)	Total
2005 (OrderDate)	\$3,105,587	\$160,786			\$3,266,374
2006 (OrderDate)		\$6,416,193	\$114,151		\$6,530,344
2007 (OrderDate)			\$9,403,398	\$387,663	\$9,791,060
2008 (OrderDate)				\$9,770,900	\$9,770,900
Total	\$3,105,587	\$6,576,979	\$9,517,549	\$10,158,562	\$29,358,677

Rysunek 4.22. Wartości sprzedaży według daty zamówienia (OrderDate) i daty wysyłki (ShipDate)

Na rysunku 4.22 można zaobserwować korzyści analityczne, jakie daje nam posiadanie tabeli z datami wysyłki i tabeli z datami zamówień w tym samym modelu danych. W naszym przykładzie sprzedaż całkowita jest wyświetlana w formie wizualnej macierzy, gdzie poszczególne wiersze reprezentują rok zamówienia z tabeli dat zamówień, a poszczególne kolumny reprezentują rok wysyłki z tabeli dat wysyłki. Dzięki takiemu rozwiązaniu możemy łatwo zauważyć, że na przykład:

1. W 2005 roku całkowita wartość sprzedaży wyniosła 3 266 374 dolary.
2. W 2005 roku wartość sprzedanych produktów, które zostały wysłane w tym samym roku (2005), wyniosła 3 105 587 dolarów.
3. Jeżeli spojrzymy na kolumnę 2006 (ShipDate), zauważymy, że w 2006 roku zostały wysłane produkty o całkowitej wartości 6 576 979 dolarów. Przy bliższej analizie okazuje się jednak, że produkty o wartości 160 786 dolarów wysłane w 2006 roku zostały zamówione jeszcze w 2005 roku, a pozostałe produkty o wartości 6 416 193 dolary zostały zamówione w 2006 roku.

Mimo oczywistych zalet rozwiązania wykorzystujące wielokrotne importowanie tabel mają również pewne wady, na przykład:

- **Zwiększone zużycie zasobów** — wielokrotne importowanie tej samej tabeli zajmuje dodatkową pamięć i miejsce na dysku.
- **Zwiększone zapotrzebowanie na obsługę administracyjną** — wszelkie modyfikacje dokonane w jednej tabeli będą musiały być powtórzone w jej wszystkich zaimportowanych instancjach, ponieważ zaimportowane kopie tabeli nie są ze sobą powiązane. Na przykład, jeżeli utworzysz nową hierarchię w jednej tabeli dat, to będziesz musiał utworzyć taką samą hierarchię w jej wszystkich zaimportowanych kopiach.
- **Analityka czasowa** — obliczenia analityki czasowej będą musiały być dostosowane do każdej tabeli dat.

Raport pokazany na rysunku 4.23 przedstawia całkowitą wartość sprzedaży i całkowitą liczbę transakcji według lat, ale jakich? Czy chodzi o lata, w którym produkty zostały zakupione, czy lata, w których produkty zostały wysłane? Aktywna relacja dla tej tabeli została utworzona dla daty zamówienia (OrderDate), zatem nasz raport wyświetla wyniki obliczone na podstawie daty zamówienia produktu.

CalendarYear	Total Sales	Total Transactions
2005	\$3,266,373.66	1013
2006	\$6,530,343.53	2677
2007	\$9,791,060.30	24443
2008	\$9,770,899.74	32265
Total	\$29,358,677.22	60398

Rysunek 4.23. Całkowita wartość sprzedaży i całkowita liczba transakcji w podziale według lat

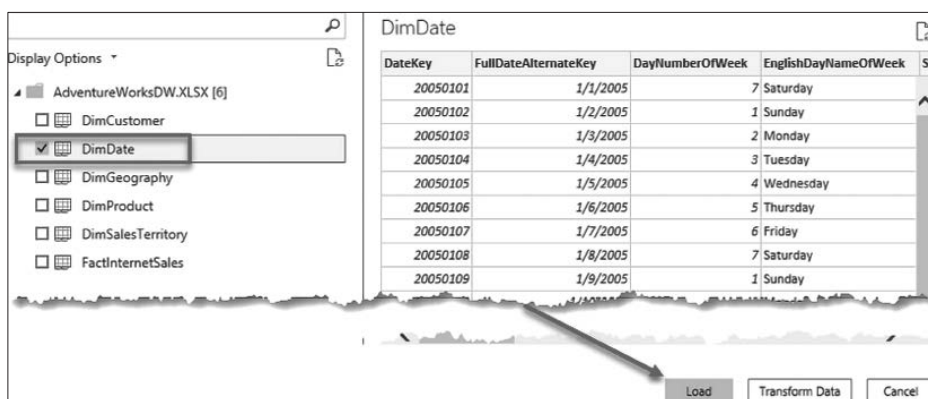
Nietrudno zauważyć, że powyższa wizualizacja, choć całkowicie poprawna, może być dosyć niejednoznaczna. Aby pozbyć się takiej niepewności z naszych raportów, model danych może zostać dodatkowo ulepszony poprzez odpowiednią zmianę nazw

kolumn. W następnej sekcji dowiesz się, jak za pomocą niewielkich zmian w modelu danych spowodować, że wizualizacje będą bardziej konkretne.

Importowanie tabeli dat

W tej sekcji zaimportujemy tabelę dat, aby przeprowadzić analizę danych na podstawie daty wysyłki zamówionych produktów. Aby to zrobić, kliknij polecenie *Get data (Pobierz dane)*, wybierz łącznik dla skoroszytu programu Excel i otwórz plik *AdventureWorksDW.xlsx*, który znajdziesz w zasobach książki.

Po otwarciu pliku zaznacz tabelę *DimDate* i naciśnij przycisk *Load (Załaduj)*, jak pokazano na rysunku 4.24.



Rysunek 4.24. Importowanie tabeli *DimDate* do Power BI

Po zaimportowaniu danych kolejnym krokiem będzie utworzenie odpowiednich relacji. Wybierz polecenie *Manage relationships (Zarządzanie relacjami)*, które znajdziesz na karcie *Home (Narzędzia główne)*, lub *Modeling (Modelowanie)*, w zależności od tego, w którym widoku teraz się znajdujesz. Na ekranie pojawi się okno edytora relacji. Naciśnij przycisk *New (Nowa)*, aby utworzyć nową relację. Jeżeli chcesz sobie przypomnieć, jak utworzyć nową relację, powinieneś wrócić do rysunku 4.7. Teraz wykonaj następujące czynności:

1. Z pierwszej listy rozwijanej wybierz tabelę *FactInternetSales*.
2. Zaznacz kolumnę *ShipDate*; użyj paska przewijania, aby przewinąć ją do końca w prawo.
3. Z drugiej listy rozwijanej wybierz tabelę *DimDate* (2).
4. Zaznacz kolumnę *FullDateAlternateKey*.
5. Naciśnij przycisk *OK*, aby zamknąć okno tworzenia relacji.

Dla większej przejrzystości pozwoliłem sobie tutaj zmienić nazwy tabel i kolumn. W następnym podrozdziale, zatytułowanym „Usprawnienia użyteczności modelu danych”, dowiesz się, jak można zmienić nazwy tabel i kolumn. Nazwa tabeli DimDate została zmieniona na Date (Order), a nazwa tabeli DimDate (2) została zmieniona na Date (Ship).

Nasz model danych posiada teraz dwie tabele dat, każda z aktywną relacją do tabeli FactInternetSales. Jeżeli teraz będziesz chciał zobaczyć dane o sprzedaży według roku zamówienia, powinieneś wprowadzić do raportu kolumnę roku z tabeli Date (Order), a jeżeli chcesz zobaczyć sprzedaż według roku wysyłki, to powinieneś użyć kolumny roku z tabeli Date (Ship), tak jak to zostało pokazane na rysunku 4.25.

Filtrowanie według daty zamówienia			Filtrowanie według daty wysyłki		
Order Year	Total Sales	Total Transactions	Order Year	Total Sales	Total Transactions
2005	\$3,266,373.66	1013	2005	\$3,105,587.33	962
2006	\$6,530,343.53	2677	2006	\$6,576,978.98	2665
2007	\$9,791,060.30	24443	2007	\$9,517,548.53	23313
2008	\$9,770,899.74	32265	2008	\$10,158,562.38	33458
Total	\$29,358,677.22	60398	Total	\$29,358,677.22	60398

Rysunek 4.25. Wyświetlanie kolumny zawierającej rok wysyłki

Wielokrotne importowanie tej samej tabeli do modelu danych jest preferowaną metodą w sytuacji, gdy dwie tabele mają wiele relacji między sobą. Takie rozwiązanie jest łatwe do wytłumaczenia użytkownikom końcowym i pozwala na ponowne wykorzystanie większości, jeśli nie wszystkich, istniejących obliczeń DAX.

Alternatywnym rozwiązaniem jest zdefiniowanie nieaktywnych relacji, a następnie utworzenie nowych kalkulacji (miar) przy użyciu języka DAX. Warto jednak zauważyć, że taka metoda wykorzystania nieaktywnych relacji może stać się bardzo uciążliwa z administracyjnego punktu widzenia. Wyobraź sobie konieczność tworzenia kopii istniejących miar w modelu danych dla każdej relacji między dwiema tabelami. W obecnym modelu danych tabela FactInternetSales przechowuje trzy daty, a to oznaczałoby konieczność utworzenia i utrzymywania trzech kopii każdej miary, po jednej dla każdego rodzaju daty.

W tej sekcji dowiedziałeś się, jak zmienić nazwę kolumny, aby zapewnić przejrzystość modelu danych; jest to bardzo ważny etap. Teraz przyjrzymy się innym usprawnieniom użyteczności, które mogą poprawić funkcjonalność naszego modelu danych.

Usprawnienia użyteczności modelu danych

Usprawnienia użyteczności to takie zmiany, które mogą znacząco poprawić ogólny komfort pracy użytkownika podczas interakcji z modelem danych. Aby zapewnić, żeby użytkownik końcowy był zadowolony z Twojego projektu, powinieneś pamiętać o kilku podstawowych usprawnieniach, które w znaczący sposób mogą wpłynąć na funkcjonalność całego rozwiązania.

W tym rozdziale omówimy następujące usprawnienia:

- Ukrywanie wybranych tabel i kolumn.
- Zmiana nazw tabel i kolumn.
- Zmiana domyślnej właściwości podsumowania.
- Wyświetlanie jednej kolumny, ale sortowanie według innej.
- Definiowanie kategorii danych poszczególnych pól.
- Tworzenie hierarchii.

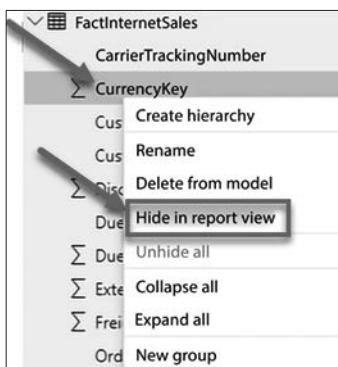
Na początek zastanówmy się nad tym, jak można ukrywać wybrane tabele i kolumny.

Ukrywanie tabel i kolumn

Niektóre tabele dostępne w modelu danych mają tylko charakter pomocniczy i nigdy nie są bezpośrednio używane w raporcie. Na przykład możesz mieć osobną tabelę do obsługi relacji typu wiele-do-wielu czy innych elementów. Niektóre kolumny są po prostu niezbędne do tworzenia relacji w modelu danych, ale nie dodają żadnej wartości analitycznej do raportów. Tabele lub kolumny, które nie będą bezpośrednio wykorzystywane do celów raportowania, powinny być ukryte w widoku raportu, aby zmniejszyć jego złożoność i poprawić komfort pracy użytkownika.

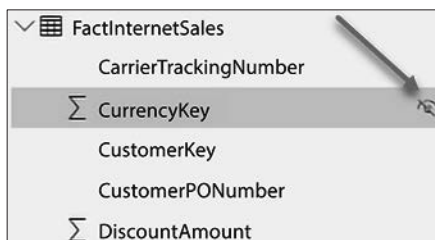
Aby ukryć wybraną kolumnę lub tabelę, kliknij prawym przyciskiem myszy na obiekcie, który chcesz ukryć, a następnie wybierz polecenie *Hide in report view* (*Ukryj w widoku raportu*). Jeżeli już pracujesz w widoku raportu, opcja ta będzie po prostu nosiła nazwę *Hide* (*Ukryj*).

Przejdź do widoku modelu, odszukaj tabelę *FactInternetSales*, kliknij prawym przyciskiem myszy na kolumnie *CurrencyKey*, a następnie z menu podręcznego wybierz opcję *Hide in report view* (*Ukryj w widoku raportu*), jak pokazano na rysunku 4.26.



Rysunek 4.26. Opcja Hide in report view (Ukryj w widoku raportu)

Kolumny, które zostały ukryte w widoku raportu, są nadal widoczne w widoku danych i relacji. Kolumny ukryte można rozpoznać na dwa sposoby. Kolumny te będą miały ikonę widoczności z prawej strony nazwy kolumny, tak jak to zostało pokazane na rysunku 4.27.



Rysunek 4.27. Ukryta kolumna

Teraz przejdź po kolei do każdej tabeli i ukryj wszystkie pozostałe kolumny kluczy, z wyjątkiem kolumny FullDateAlternateKey.

Uwaga

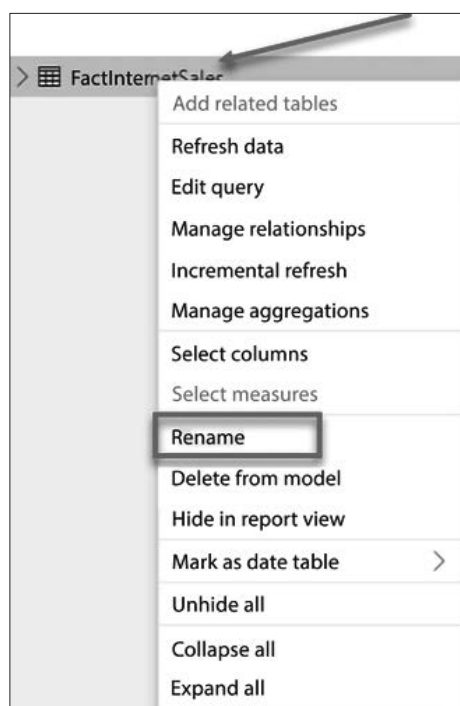
Kiedy pracujesz w widoku modelu, możesz zaznaczać wiele kolumn, przytrzymując wciśnięty klawisz *Ctrl* podczas zaznaczania kolumn. Dzięki temu możesz najpierw zaznaczyć wszystkie kolumny, które mają zostać ukryte, a następnie ukryć je za pomocą jednego polecenia.

Zmiana nazw tabel i kolumn

Zmiana nazw wybranych tabel i kolumn jest ważnym krokiem w tworzeniu efektywnego i wygodnego w użyciu modelu danych. Różne grupy użytkowników często używają różnych określeń dla tego samego obiektu, dlatego ważne jest, aby podczas modyfikowania

nazw obiektów wziąć pod uwagę wymagania czy nawet przyzwyczajenia poszczególnych grup użytkowników. Na przykład założmy, że w swoim raporcie masz kolumnę z listą klientów i zdecydowałeś się nadać jej nazwę *Customer*. Jednak może się okazać, że w dziale sprzedaży używa się dla takiej kolumny nazwy *Prospect*, *Client* lub nawet jeszcze innej. Pamiętaj, aby podczas modyfikowania nazw tabel i kolumn mieć na względzie użytkowników końcowych i odbiorców raportów.

Nazwę tabeli lub kolumny możesz zmienić w widoku raportu, danych lub modelu. Na przykład przejdź do widoku modelu i kliknij prawym przyciskiem myszy tabelę *FactInternetSales*, a następnie z menu podręcznego wybierz polecenie *Rename* (*Zmień nazwę*), jak pokazano na rysunku 4.28. Zmień nazwę tej tabeli na *Internet Sales*.

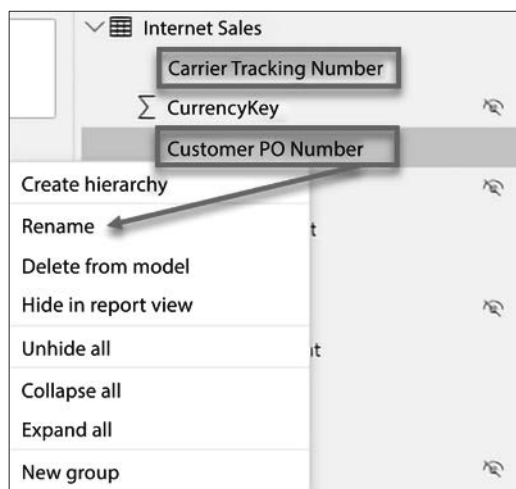


Rysunek 4.28. Zmiana nazwy tabeli *FactInternetSales*

Następnie zmień nazwy pozostałych tabel, usuwając prefiks *Dim* i dodając spacje tam, gdzie to jest niezbędne. Dla ułatwienia zamieszczamy poniżej listę proponowanych zmian.

Pierwotne nazwy	Nowe nazwy
FactInternetSales	Internet Sales
DimDate	Date (Order)
DimDate (2)	Date (Ship)
DimProduct	Product
DimCustomer	Customer
DimSalesTerritory	Sales Territory
5 Regions 2008	Temperature

Wykonanie następnego kroku jest konieczne, ale może to być żmudny proces. Jeżeli jesteś programistą, to z pewnością przyzwyczyłeś się do unikania spacji w nazwach tabel i kolumn. Użytkownicy końcowi i odbiorcy raportów zazwyczaj jednak oczekują takich spacji, stąd raczej zaleca się ich używanie tam, gdzie jest to przydatne. Spacje w nazwach kolumn powinny być używane w kolumnach, które są widoczne, ale niekoniecznie w kolumnach ukrytych w widoku raportu. Aby zmienić nazwę wybranej kolumny, kliknij ją prawym przyciskiem myszy, a następnie z menu podręcznego wybierz polecenie *Rename (Zmień nazwę)*. Na rysunku 4.29 spacje zostały dodane w nazwach kolumn CarrierTrackingNumber i CustomerPONumber.



Rysunek 4.29. Zmiana nazw kolumn

Aby zmienić nazwy pozostałych kolumn, powinieneś wykonać polecenia opisane poniżej:

1. Powtórz proces dodawania spacji dla pozostałych kolumn w każdej tabeli.
2. Zmień nazwę kolumny FullDateAlternateKey na Date.

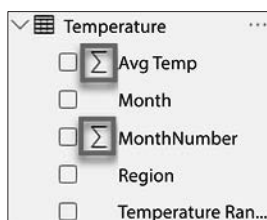
Pamiętaj, odpowiednia zmiana nazw kolumn jest prostym, ale skutecznym krokiem do poprawy komfortu użytkownika modelu danych! Teraz przyjrzyjmy się kolejnemu ważnemu usprawnieniu użyteczności modelu danych, czyli zmianie domyślnego podsumowania kolumn numerycznych.

Uwaga

Możesz zautomatyzować proces zmiany nazwy kolumn w Power BI za pomocą edytora Power Query. W serwisie YouTube znajdziesz film (w języku angielskim) pokazujący, jak to zrobić, <https://www.youtube.com/watch?v=GF5S2ktPTB0&feature=youtu.be>.

Podsumowanie domyślne

Domyślnie Power BI tworzy podsumowanie dla kolumn numerycznych, które nie posiadają relacji lub występują po stronie *jeden* relacji. Domyślnym podsumowaniem jest zazwyczaj suma wartości. Kolumny, którym przypisano domyślne podsumowania, są oznaczone przez Power BI symbolem (Σ), który jest widoczny w widoku raportu. Takie domyślne podsumowania można zaobserwować na przykład w tabeli Temperature. Na rysunku 4.30 możesz zauważyć, że dla kolumn Avg Temp i Month Number zostały utworzone domyślne podsumowania.



Rysunek 4.30. Domyślne podsumowania przypisane do kolumn numerycznych

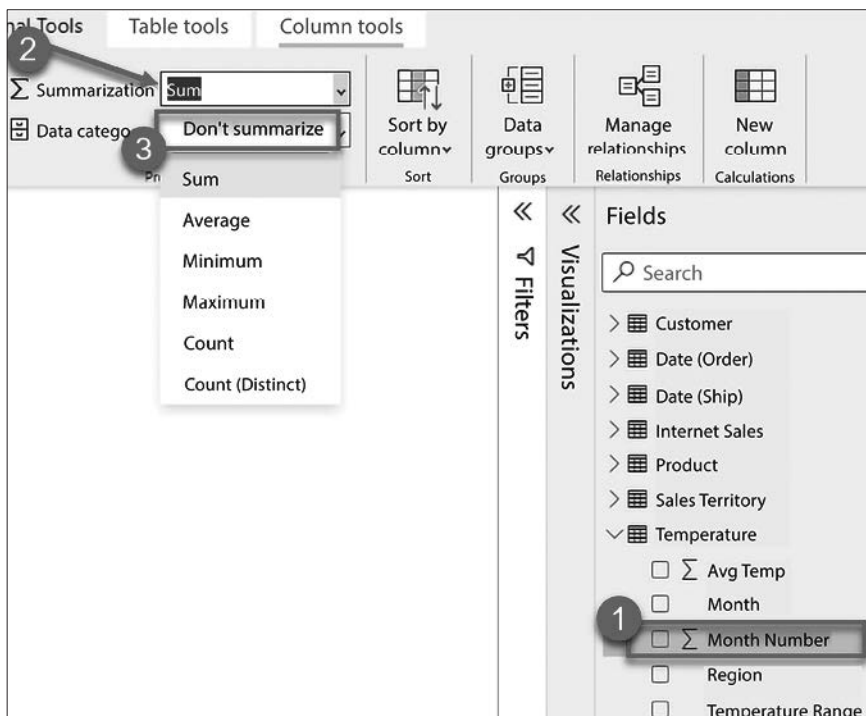
Takie automatyczne przypisywanie domyślnych podsumowań ma swoje plusy i minusy. Zaletą jest to, że pola takie jak Sales Amount czy Total Cost będą automatycznie agregowane po dodaniu do raportu, co znakomicie ułatwia proces jego tworzenia. Z kolei wady takiego rozwiązania ujawniają się w przypadku modeli danych, które zawierają kolumny numeryczne mające charakter opisowy — automatyczne agregowanie takich kolumn po dodaniu ich do raportu może powodować pewne zamieszanie. Kolumny oznaczone na

rysunku 4.30 spełniają rolę atrybutów opisowych, które nadają analizowanym danym odpowiedni kontekst, stąd kolumny te nie powinny być agregowane. Przyjrzyj się sytuacji przedstawionej na rysunku 4.31.

Month Number
780

Rysunek 4.31. Suma numerów miesięcy w tabeli Temperature

Na rysunku 4.31 kolumna Month Number z tabeli dat została dodana do wizualizacji tabeli, a jej oczekiwanym zachowaniem jest wyświetlenie listy numerów miesięcy (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12). Jednak zamiast zwracać odrębną listę, raport zwraca sumę wszystkich wartości reprezentujących numery miesięcy z tabeli Temperature, dając w efekcie zupełnie nieprzydatną wartość 780. Na szczęście domyślny sposób agregacji można zmienić. Zobacz rysunek 4.32 i towarzyszące mu kroki.



Rysunek 4.32. Dostosowywanie domyślnego sposobu podsumowania

Aby dowiedzieć się, jak zmienić domyślny sposób podsumowania, powinieneś kolejno wykonać polecenia pokazane na rysunku 4.32.

1. Rozwiń tabelę *Temperature* i wybierz kolumnę *Month Number*. Pamiętaj, aby kliknąć nazwę kolumny, a nie pole wyboru. Po wybraniu kolumny w menu programu *Power BI Desktop* pojawi się karta *Column tools* (*Narzędzia kolumny*).
2. Przejdź na kartę *Column tools* (*Narzędzia kolumny*).
3. Z listy rozwijanej *Summarization* (*Podsumowanie*) wybierz wartość *Don't summarize* (*Nie sumuj*).

Powtórz powyższy proces dla każdej kolumny w modelu danych, która nie powinna być automatycznie sumowana.

Uwaga

Kolumny, które znajdują się w tabelach po stronie *jeden* relacji, będą miały automatycznie ustawioną opcję *Don't summarize* (*Nie sumuj*). Jeżeli przyjrzyj się uważnie tabelom dat, z pewnością zauważysz, że inne kolumny numeryczne, takie jak *Calendar Year* czy *Month Number*, mają opcję domyślnego podsumowania automatycznie ustawioną na wartość *Don't summarize* (*Nie sumuj*). Jest to kolejna korzyść płynąca z poprawnego definiowania relacji w *Power BI*.

Teraz, gdy wiesz już, jak zmienić rodzaj domyślnego podsumowania wybranych kolumn, przyjrzymy się kolejnemu ważnemu usprawnieniu użyteczności modelu danych. W następnym punkcie dowiesz się, jak skonfigurować kolumny w modelu danych tak, aby dane były odpowiednio posortowane w wizualizacjach.

Jak wyświetlić jedną kolumnę, a posortować według innej

Często zdarza się, że chcesz wyświetlić dane z jednej kolumny, ale posortować je według innej. Na przykład kolumna zawierająca nazwy miesięcy po dodaniu do wizualizacji raportu jest sortowana alfabetycznie; zobacz rysunek 4.33.

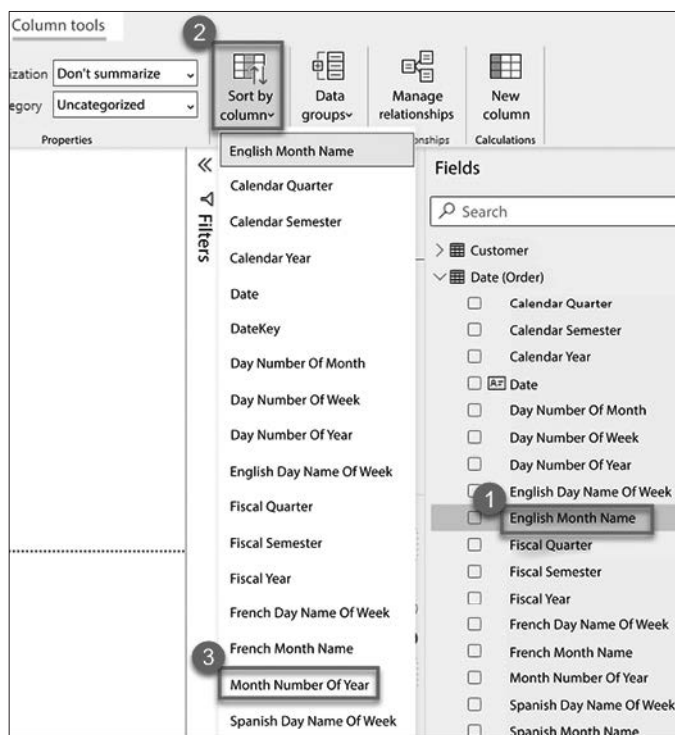
Pożądanym zachowaniem jest oczywiście w takim wypadku sortowanie chronologiczne. W związku z tym w raporcie powinny być wyświetlane nazwy miesięcy, ale sortowane według numerów miesięcy w roku. Aby to zrobić, powinieneś wykonać polecenia pokazane na rysunku 4.34.

Aby zmienić porządek sortowania kolumny, wykonaj następujące czynności:

1. Rozwiń tabelę *Date* i wybierz kolumnę *English Month Name*, jak pokazano na rysunku 4.34.
2. Przejdź na kartę *Column tools* (*Narzędzia kolumny*).
3. Rozwiń listę *Sort by column* (*Sortuj według kolumny*) i wybierz kolumnę *Month Number of Year*.

English Month Name
April
August
December
February
January
July
June
March
May
November
October
September

Rysunek 4.33. Nazwy miesięcy po dodaniu do wizualizacji raportu są sortowane alfabetycznie



Rysunek 4.34. Zmiana porządku sortowania wybranej kolumny

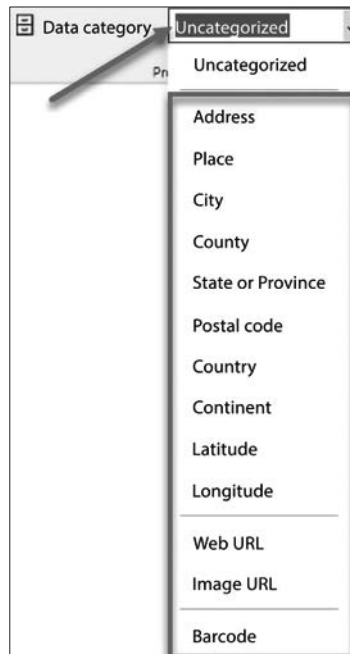
W kolejnym punkcie dowiesz się, jak można kategoryzować kolumny, co jeszcze bardziej może usprawnić pracę z raportem.

Kategoryzacja danych

Power BI automatycznie przyjmuje pewne założenia dotyczące kolumn na podstawie typów danych, nazw kolumn oraz relacji w modelu danych. Założenia te są wykorzystywane w widoku raportu podczas tworzenia wizualizacji, aby poprawić interakcję użytkownika z narzędziem. Po rozpoczęciu budowania wizualizacji możesz zauważyć, że Power BI dla różnych kolumn wybiera różne typy wizualizacji; jest to oczywiście zamierzone działanie. Power BI automatycznie decyduje również o rozmieszczeniu kolumn w sekcji *pól* wizualizacji — więcej na temat tworzenia wizualizacji dowiesz się w rozdziale 6, „Wizualizacja danych”. Jak już wspominaliśmy wcześniej w tym rozdziale, gdy Power BI wykryje kolumnę, która ma wartości liczbowe po stronie *wiele* relacji, domyślnie tworzy dla niej podsumowanie. Power BI zakłada, że użytkownik będzie chciał agregować takie dane i dlatego automatycznie umieszcza takie kolumny numeryczne w obszarze *Values* (*Wartości*) w wizualizacji raportu.

Kategoryzacja danych pozwala zarówno na usprawnienie pracy użytkownika, jak również zwiększenie jej dokładności. W chwili kiedy powstawała ta książka, dostępnych było trzynaście różnych opcji kategoryzacji danych.

Na rysunku 4.35 przedstawiono pełną listę kategorii danych dostępnych w Power BI.



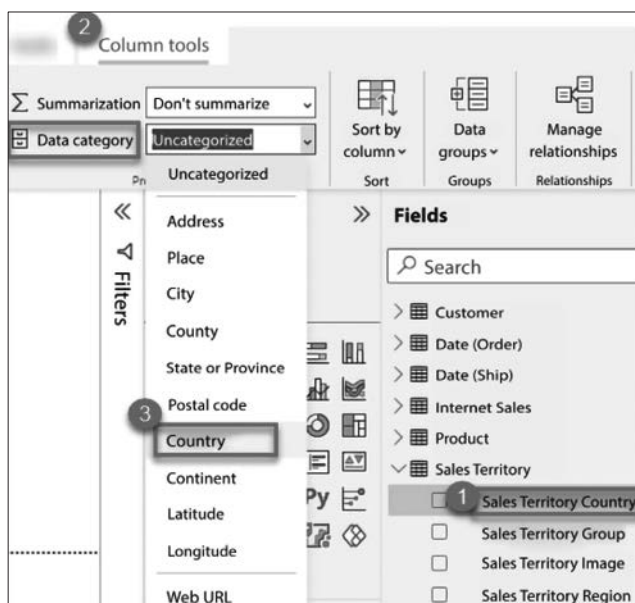
Rysunek 4.35. Opcje kategoryzacji danych

Najczęstszym zastosowaniem tych opcji jest kategoryzacja danych geograficznych. Przykładowo, kiedy takie dane dodawane są do mapy, usługa Mapy Bing musi przyjąć pewne założenia dotyczące sposobu ich przetwarzania, co czasami może skutkować niedokładnymi wynikami. Takie problemy można jednak znacznie ograniczyć, a nawet całkowicie wyeliminować dzięki przeprowadzeniu odpowiedniej kategoryzacji danych.

Uwaga

Jedną z metod, którą uznałem za niezwykle przydatną, jest połączenie wielu kolumn adresowych (na przykład City, State) w jedną kolumnę i przypisanie nowej kolumnie kategorii *Place (Miejsce)*. Używam tej metody od lat z wielkim powodzeniem. Więcej wskazówek na temat mapowania danych geograficznych znajdziesz w artykule zamieszczonym na blogu MitchellSQL, dostępnym pod skróconym adresem <https://tinyurl.com/pbiqs-categoryplace>.

Aby zaktualizować kategorię danych z wybranej kolumny, powinieneś wykonać polecenia przedstawione na rysunku 4.36.



Rysunek 4.36. Modyfikowanie kategorii danych

1. Rozwiń tabelę Sales Territory i wybierz kolumnę Sales Territory Country.
2. Przejdź na kartę *Column tools (Narzędzia kolumn)*.
3. Rozwiń listę *Data category (Kategoria danych)* i wybierz opcję *Country (Kraj)*.

Odpowiednia klasyfikacja kolumn zawierających dane geograficzne w modelu danych pomoże usłudze Bing Maps prawidłowo odwzorować Twoje dane. Kiedy dane geograficzne nie posiadają przypisanych kategorii, jest o wiele bardziej prawdopodobne, że zostaną nieprawidłowo odwzorowane. Wspomniany wcześniej post z bloga MitchellSQL przedstawia przykłady danych, które zostały nieprawidłowo odwzorowane przez Mapy Bing, i pokazuje, jak właściwa klasyfikacja danych rozwiązała problem mapowania.

Tworzenie hierarchii

Predefiniowane hierarchie danych mogą przynieść kilka kluczowych korzyści. Niektóre z tych zalet wymieniono poniżej:

1. Hierarchie porządkują atrybuty i pokazują relacje między danymi.
2. Hierarchie umożliwiają interaktywne wykorzystanie metody „przeciągnij i upuść”.
3. Hierarchie dodają znaczącą wartość analityczną do warstwy wizualizacji poprzez rozwijanie i zwiżanie grup danych, w zależności od potrzeb.

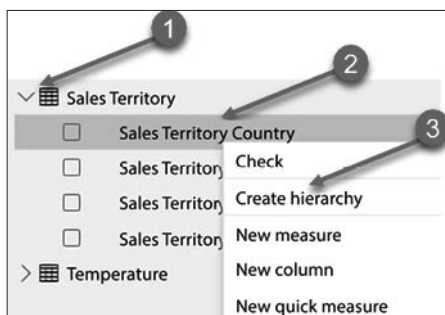
Hierarchie przechowują informacje o relacjach między danymi, o których w innym przypadku użytkownicy mogliby się nie dowiedzieć. Pamiętam, jak pracowałem kiedyś dla klienta z branży telekomunikacyjnej, który posiadał rozbudowaną sieć bazowych stacji nadawczych (ang. *Base Transceiver Station* — BTS) i sektorów działania i bez spojrzenia na swoje notatki, nigdy nie mogłem zapamiętać poprawnej kolejności. Czy BTS-y posiadały wiele sektorów, czy też każdy sektor zawierał wiele BTS-ów? Kiedy w modelu danych została utworzona odpowiednia hierarchia, nie musiałem się już tym martwić, ponieważ relacje były przechowywane w hierarchii. Poniżej zamieszczamy listę przykładowych hierarchii danych:

- *Kategoria* → *Podkategoria* → *Produkt*.
- *Kraj* → *Region* → *Miasto*.
- *Rok* → *Kwartał* → *Miesiąc* → *Dzień*.

Aby utworzyć nową hierarchię, powinieneś wykonać polecenia pokazane na rysunku 4.37.

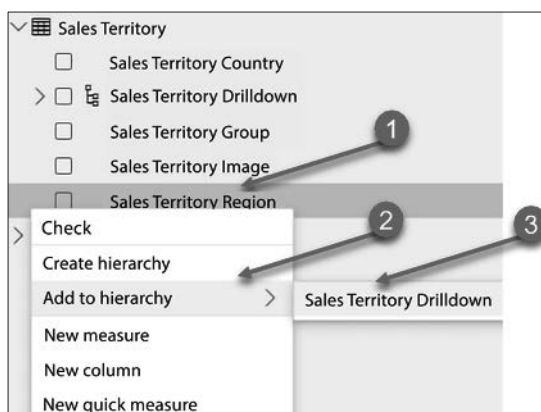
1. Rozwiń tabelę `Sales Territory`.
2. Kliknij prawym przyciskiem myszy kolumnę `Sales Territory Country`.
3. Z menu podręcznego wybierz polecenie *Create hierarchy (Utwórz hierarchię)*.

Wykonanie powyższych operacji spowodowało utworzenie nowej hierarchii z jedną kolumną, której nadana została domyślna nazwa `Sales Territory Country Hierarchy`. Kliknij prawym przyciskiem myszy nowo utworzoną hierarchię i zmień jej nazwę na `Sales Territory Drilldown`. Kolejnym etapem będzie dodanie do tej hierarchii dodatkowych kolumn bądź atrybutów.



Rysunek 4.37. Tworzenie nowej hierarchii

Aby to zrobić, wykonaj polecenia pokazane na rysunku 4.38.



Rysunek 4.38. Dodawanie kolumn lub atrybutów do hierarchii

1. Przejdź do tabeli Sales Territory i kliknij prawym przyciskiem myszy kolumnę Sales Territory Region.
2. Z menu podręcznego wybierz polecenie *Add to hierarchy* (*Dodaj do hierarchii*).
3. Z listy wybierz hierarchię Sales Territory Drilldown.
4. Powtórz kroki 1 – 3 dla kolumny Sales Territory Group.

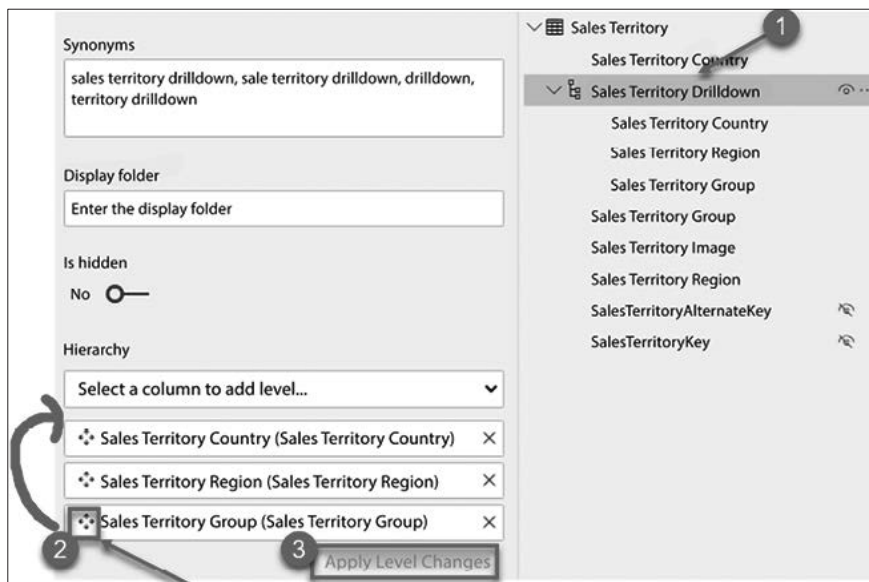
Gotową hierarchię możesz zobaczyć na rysunku 4.39. Zauważ, że kolejność atrybutów jest jednak nieprawidłowa; powinna ona wyglądać tak: Sales Territory Group → Sales Territory Country → Sales Territory Region.

Aby zmienić kolejność atrybutów hierarchii, przejdź do widoku modelu i wykonaj następujące polecenia (patrz rysunek 4.40):

1. W widoku modelu wybierz Sales Territory Drilldown z tabeli Sales Territory.



Rysunek 4.39. Gotowa hierarchia



Rysunek 4.40. Zmiana kolejności kolumn w hierarchii

2. Z panelu *Properties* (Właściwości) przeciągnij Sales Territory Group na górę hierarchii.
3. Naciśnij *Apply Level Changes* (Zastosuj zmiany na poziomie).

Na rysunku 4.41 pokazano prawidłowo uporządkowaną hierarchię.



Rysunek 4.41. Prawidłowo uporządkowana hierarchia

W tym podrozdziale dowiedziałeś się, jak wykorzystać drobne, ale bardzo skuteczne usprawnienia do poprawy czytelności i efektywności modelu danych. Takie modyfikacje wpływają na zwiększenie komfortu pracy użytkownika i ułatwiają zapoznanie się z modelem danych i jego zrozumienie.

W kolejnym podrozdziale przejdziemy do omawiania wybranych zagadnień związanych z wydajnością modelu danych.

Poprawa wydajności modelu danych

Wydajność modelu danych w Power BI może być mierzona na dwa sposoby: poprzez wydajność zapytań oraz poprzez wydajność przetwarzania. Wydajność zapytań określa, jak szybko wyniki są zwracane przez wizualizacje i raporty. Wydajność przetwarzania jest miarą czasu potrzebnego na wykonanie odświeżenia danych w bazowym zbiorze danych. Wydajność modelu danych jako całości jest bardzo ważna i każdy programista Power BI powinien zawsze zdawać sobie sprawę z tego, jak decyzje projektowe mogą wpłynąć na wydajność modelu dziś lub w przyszłości. Dogłębna analiza zagadnień związanych z wydajnością modelu danych wykracza daleko poza zakres tej książki, stąd w tym podrozdziale przedstawimy tylko jej ogólny zarys.

Wydajność zapytań

Jak dowiedziałeś się w rozdziale 2., „Łączenie się z danymi”, w Power BI istnieje wiele sposobów łączenia się ze źródłami danych. Na przykład możemy zaimportować dane, użyć zapytania DirectQuery, skorzystać z połączenia na żywo lub zastosować kombinację importu i zapytań bezpośrednich w modelu złożonym.

Metodologie projektowania modeli danych

Modele danych w Power BI są projektowane specjalnie w celu wydobycia z danych wartości analitycznej i podejmowania na tej podstawie świadomych decyzji biznesowych. Z tego względu dane powinny być modelowane w taki sposób, aby można było skutecznie tworzyć poprawne i efektywne raporty. W Power BI możemy wyróżnić trzy typy modeli danych: modele płaskie (ang. *flat model*), schemat gwiazdy (ang. *star schema*) oraz schemat płatków śniegu (ang. *snowflake schema*).

Model płaski lub inaczej mówiąc, całkowicie zdenormalizowany, to taki model, który składa się z pojedynczej tabeli danych bez żadnych tabel pomocniczych. W takim modelu wszystkie mierzalne elementy i wszystkie atrybuty opisowe znajdują się w tej samej tabeli. Ten model jest bardzo powszechny, co wynika z tego, że wielu użytkowników Excela importuje dane do Power BI bezpośrednio ze swoich arkuszy Excela. Taka metoda jest jednak wysoce nieefektywna i ma kilka wad:

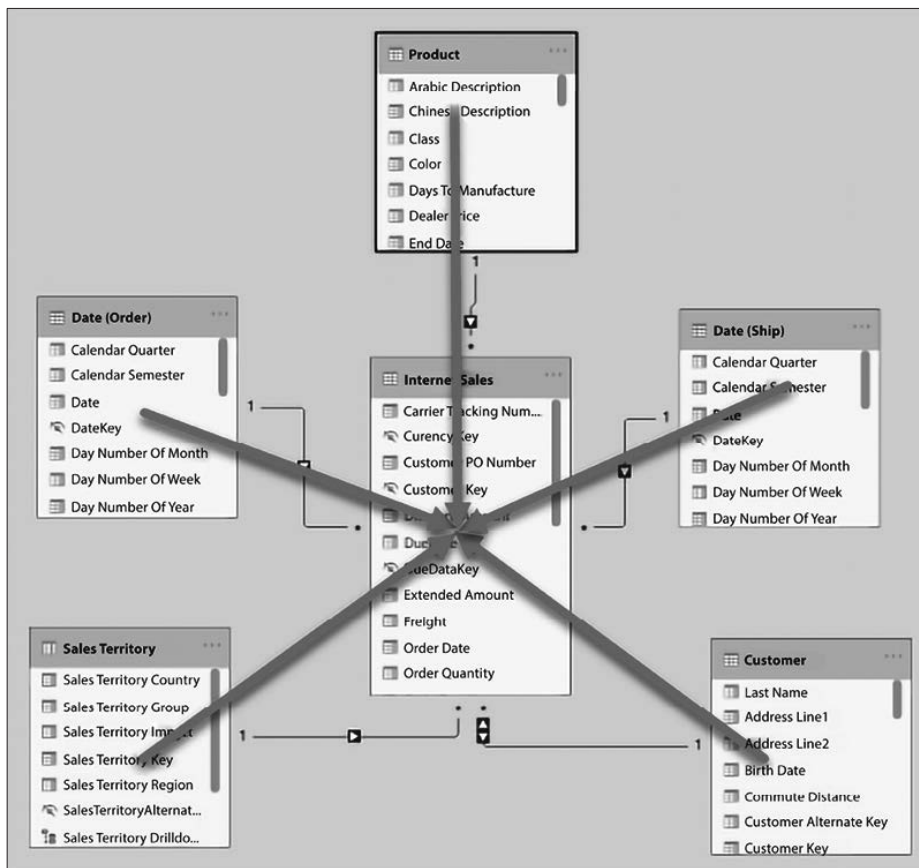
- Model płaski nie skaluje się zbyt dobrze; wraz ze wzrostem liczby rekordów w tabeli taki model danych będzie zużywał znacznie więcej zasobów ze względu na redundancję danych i rosnącą liczbę kolumn.
- Model płaski nie jest elastyczny i nie dostosowuje się dobrze do przyszłych zmian.
- Model płaski nie jest „inteligentny”, co oznacza po prostu, że czasami bywa zagmatwany i trudny do zrozumienia.
- Obliczenia wykorzystujące analizę szeregów czasowych, takie jak od początku roku do daty (ang. *YTD — year to date*) czy rok do roku (ang. *year over year*) są znacznie trudniejsze do wykonania.

Kolejnym typem modelu danych jest schemat gwiazdy, który jest preferowanym sposobem modelowania danych w Power BI. Termin „schemat gwiazdy” wywodzi się z modelowania wymiarowego. Modelowanie wymiarowe jest sposobem, w jaki przedsiębiorstwa i organizacje projektują swoje hurtownie danych analitycznych od ponad 30 lat. Schemat gwiazdy posiada dwa rodzaje tabel: tabele faktów i tabele wymiarów. Powodem, dla którego takie modele danych nazywane są schematami gwiazdy, jest to, że tabele wymiarów otaczają tabelę faktów, co przy odrobinie wyobraźni przypomina nieco kształt gwiazdy (patrz rysunek 4.42).

Tabele faktów przechowują w modelu danych mierzalne elementy, takie jak sprzedaż, kwota podatku, czas trwania rozmowy telefonicznej. Tabele wymiarów przechowują atrybuty opisowe, które pomagają wyjaśnić modelowane elementy. Atrybuty powiązane z poszczególnymi elementami przechowywane są zazwyczaj w osobnych tabelach. Na przykład atrybuty produktu, takie jak nazwa, kolor, rozmiar, waga i inne, będą przechowywane w tabeli produktu.

Zaprojektowanie modelu danych opartego na schemacie gwiazdy w Power BI Desktop może co prawda wymagać nieco więcej czasu, ale za to przynosi kilka niezaprzeczalnych korzyści — warto zauważyć, że poniższa lista z całą pewnością nie obejmuje wszystkich zalet schematu gwiazdy:

- Model danych jest skalowalny, co oznacza, że będzie płynnie powiększał się w miarę dodawania kolejnych danych.
- Model danych jest elastyczny, dzięki czemu dodatkowe tabele mogą być łatwiej zintegrowane z istniejącym rozwiązaniem i wspierać nowe wymagania analityczne, które mogą pojawić się w przyszłości.
- Modele danych oparte na schemacie gwiazdy są przejrzyste i łatwe do zrozumienia.
- Modele danych oparte na schemacie gwiazdy ułatwiają implementację obliczeń związanych z analizą czasową.



Rysunek 4.42. Schemat gwiazdy

Ostatnim typem modelu danych, o którym chcielibyśmy tutaj wspomnieć, jest schemat płątka śniegu; jak sama nazwa wskazuje, wizualne wyobrażenie tego modelu bardziej przypomina płatek śniegu niż gwiazdę. W modelowaniu wymiarowym większość modeli danych zaczyna jako modele oparte na schemacie gwiazdy, ale z czasem mogą one ewoluować w kierunku schematów płątka śniegu, które pozwalają na wspieranie bardziej zaawansowanych wymagań analitycznych. Prostym przykładem schematu płątka śniegu byłoby rozbitcie wymiaru produktu na wiele osobnych tabel, takich jak kategoria produktu, podkategoria produktu i sam produkt.

Importowanie danych

Importowanie danych jest najczęściej stosowaną metodą łączenia się ze źródłami danych dla modeli Power BI, gdzie zaimportowane dane przechowywane są w pamięci. Metoda ta jest bardzo skuteczna pod względem wydajności zapytań, ponieważ wszystkie zapytania są wykonywane bezpośrednio w pamięci podręcznej, co zapewnia niezrównaną

wydajność analityczną! Modele korzystające z importowanych danych historycznych zazwyczaj nie wymagają żadnego strojenia wydajności, szczególnie w przypadku mniejszych rozwiązań.

DirectQuery

Jak wspominaliśmy w rozdziale 2., „Łączenie się z danymi”, jedną z metod łączenia się ze źródłami danych jest połączenie bezpośrednio DirectQuery. Niestety, historycznie rzecz biorąc, takie połączenia nie cechują się zazwyczaj zbyt dużą wydajnością zapytań. Połączenia DirectQuery są najczęściej wykorzystywane, gdy zbiór danych, który ma być analizowany, jest zbyt duży, aby go w całości zaimportować do Power BI.

Agregacje

Tworzenie agregacji w Power BI dostarcza potężnego mechanizmu poprawiającego wydajność zapytań. Agregacje mogą być używane zarówno z modelami importującymi dane, jak i połączeniami DirectQuery, co może przekładać się na ogromny wzrost wydajności.

Efektywne wdrożenie agregacji wymaga jednak zrozumienia specyfiki danych oraz sposobu, w jaki użytkownicy końcowi będą wykonywać zapytania. Dzięki takiej wiedzy tabela agregowana może zostać odpowiednio zaprojektowana i użyta do odpowiadania na dużą liczbę zapytań użytkownika, zamiast oryginalnej tabeli przechowującej o wiele więcej wierszy.

Przyjrzyjmy się hipotetycznemu przykładowi. Wyobraź sobie tabelę transakcji, która ma 100 milionów wierszy danych z ostatniego roku. Jeżeli większość wizualizacji będzie wykonywała zliczenia i sumy według daty, to można zbudować agregację, która zwróci informacje o całkowitej sprzedaży i transakcjach pogrupowane według daty. W takim scenariuszu zapytanie do tabeli agregującej zwróciłoby 365 wierszy danych, 1 wiersz na każdy dzień w roku, zamiast przetwarzać 100 milionów wierszy z oryginalnej tabeli. W większości modeli danych jest mało prawdopodobne, aby sama data wystarczyła do odpowiedzi na większość zapytań od użytkowników końcowych. Dlatego do tabeli agregującej mogą być dodawane dodatkowe atrybuty, na przykład informacje geograficzne. Nasza przykładowa tabela agregująca zawierałaby w takim scenariuszu na przykład informacje o całkowitej sprzedaży i transakcjach pogrupowane według dat i lokalizacji punktów sprzedaży. Oczywiście znacznie zwiększyłoby to jej rozmiar, w zależności od liczby unikatowych rekordów w tabeli geograficznej, ale nadal taka tabela agregująca byłaby dużo mniejsza niż oryginalna tabela transakcyjna i prawdopodobnie wystarczająco mała, aby zaimportować ją w całości do Power BI.

W kolejnym punkcie omówimy kilka zagadnień związanych z odpowiednim modelowaniem danych w celu poprawy wydajności przetwarzania.

Uwaga

Szczegółowe omówienie zagadnień związanych ze skutecznym wdrażaniem agregacji w modelach danych Power BI zdecydowanie wykracza poza zakres tej książki. Jeżeli jednak chciałbyś dowiedzieć się czegoś więcej na ten temat, powinieneś zajrzeć na blog Shabnam Watson: <https://shabnamwatson.wordpress.com/2019/11/21/aggregations-in-power-bi/>.

Wydajność przetwarzania

Jeżeli pracujemy na importowanych zbiorach danych, to od czasu do czasu musimy je odświeżyć, aby załadować najnowsze dane do modelu. W dzisiejszej kulturze działania opartego na danych (ang. *data-driven*) firmy i organizacje chcą gromadzić coraz więcej danych i przetwarzać je szybciej niż kiedykolwiek wcześniej. Jednak jeżeli odświeżenie zbioru danych zajmuje długie godziny, to z oczywistych powodów częstotliwość odświeżania takiego zbioru danych będzie ograniczona. Z drugiej strony, jeżeli odświeżanie zbioru danych trwa tylko kilka minut, można go odświeżać znacznie częściej, zapewniając bardziej aktualne informacje i szybszy dostęp do danych bazowych.

Składanie zapytań

Jak wspominaliśmy już wcześniej, składanie zapytań (ang. *query folding*) jest mechanizmem pozwalającym na przekazywanie procesu przetwarzania z powrotem do bazowego źródła danych. Składanie zapytań jest bardzo istotne, gdy bazowym źródłem danych jest relacyjna baza danych, taka jak Microsoft SQL Server. Przekazywanie procesu przetwarzania z powrotem do SQL Server może znacząco poprawić wydajność przetwarzania modelu danych.

Odświeżanie przyrostowe

Historycznie rzecz biorąc, odświeżanie modelu danych w Power BI zazwyczaj wymagało odświeżenia wszystkich danych. Z tego powodu, jeżeli model danych zawiera na przykład dane z pięciu lat, to przy każdej aktualizacji odświeżane są wszystkie dane z całych pięciu lat i powiązane z nimi wiersze. Nietrudno sobie wyobrazić, że może to wymagać dużego nakładu pracy. Co by jednak było, gdyby operacja odświeżania danych trwała kilkadziesiąt sekund zamiast kilku godzin?

Widziałem rozwiązania, w których wprowadzenie procesu odświeżania przyrostowego (ang. *incremental refresh*) redukowało czas trwania operacji odświeżania danych z wielu godzin do zaledwie kilku minut. Dzieje się tak, ponieważ odświeżanie przyrostowe nie przetwarza ponownie wszystkich danych w modelu — zamiast tego ładowane są tylko najnowsze dane, bez modyfikowania przy tym danych historycznych, które tego nie wymagają. Proces odświeżania przyrostowego obejmuje nowe rekordy lub rekordy, które zostały zaktualizowane.

Uwaga

Szczegółowe omówienie zagadnień związanych ze skutecznym wdrażaniem mechanizmu odświeżania przyrostowego w modelach danych Power BI zdecydowanie wykracza poza zakres tej książki. Jeżeli jednak chciałbyś dowiedzieć się czegoś więcej na ten temat, powinieneś zajrzeć na stronę <https://docs.microsoft.com/pl-pl/power-bi/connect-data/incremental-refresh-overview>.

Najlepsze praktyki

Istnieje kilka zalecanych najlepszych praktyk, które mogą pomóc w przyspieszeniu operacji odświeżania danych. Nie jest to w żadnym wypadku wyczerpująca lista, ale stanowi dobry punkt wyjścia do budowania efektywnych modeli danych:

- Importuj tylko kolumny, które są niezbędne do raportowania; usuń wszystkie inne, niepotrzebne kolumny z modelu danych.
- Analogicznie, importuj tylko niezbędne wiersze danych; odfiltruj dane, które nie są potrzebne.
- Staraj się minimalizować liczbę kolumn przechowujących unikatowe wartości; takie dane zwykle słabo się kompresują i zajmują cenne zasoby.
- Wyłącz opcję *Auto date/time for new files* (*Automatyczna data/godzina dla nowych plików*); patrz rysunek 4.12.
- Jeżeli to możliwe, powinieneś tworzyć nowe kolumny z poziomu edytora Power Query, a nie języka DAX.

Podsumowanie

W tym rozdziale dowiedziałeś się, że modele danych w Power BI Desktop powinny być przejrzyste, zrozumiałe i zaprojektowane z myślą o skalowalności i elastyczności. Dowiedziałeś się, jak tworzyć nowe relacje i modyfikować relacje już istniejące. Zobaczyłeś również, jak obsługiwać i modelować relacje złożone, takie jak wiele-do-wielu, oraz do czego służą tabele odgrywające role i jak z nich korzystać. W tym rozdziale omówiliśmy także znaczenie różnych usprawnień użyteczności, takich jak sortowanie kolumn, dostosowywanie podsumowania domyślnego, kategoryzacja danych oraz ukrywanie kolumn i tabel oraz zmiana ich nazw. Wreszcie, rozdział zakończyliśmy krótką dyskusją na temat wydajności zapytań i przetwarzania modelu danych. Jesteś teraz już przygotowany do rozpoczęcia budowania modeli danych w Power BI Desktop!

Odpowiednio zdefiniowane relacje między danymi, w połączeniu z prostymi, ale bardzo efektywnymi usprawnieniami użyteczności, pozwalają na zbudowanie modelu danych, który jest zarówno spójny, jak i elastyczny. W przeszłości tworzenie projektów BI

wymagało znacznych ilości czasu i pieniędzy. Dzięki Power BI Desktop i samoobsługowemu podejściu do analizy biznesowej użytkownik ma teraz narzędzia niezbędne do zbudowania własnego projektu BI w ciągu kilku godzin i szybkiego wydobywania z modelu danych informacji niezbędnych do podejmowania świadomych decyzji biznesowych.

W następnym rozdziale dowiesz się, jak wykorzystać wyrażenia analizy danych, aby jeszcze bardziej zwiększyć możliwości analityczne modelu danych.

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Power BI: poznaj się profesjonalnej analizy danych!

Pakiet samoobsługowych narzędzi analitycznych Microsoft Power BI zdobył uznanie najlepszych analityków danych. Ułatwia on nie tylko pozyskiwanie ze zbioru danych informacji niezbędnych do podejmowania decyzji biznesowych, ale także tworzenie przekonujących wizualizacji danych i narracji cyfrowych. Zapewnia też pełne wdrożenie zasad bezpieczeństwa danych.

Ta książka jest trzecim, zmienionym i zaktualizowanym wydaniem znakomitego przewodnika dla analityków biznesowych, dzięki któremu bezproblemowo rozpoczniesz pracę z Power BI i efektywnie skorzystasz z jego możliwości. Po lekturze będziesz lepiej rozumieć, w jaki sposób w pełni wykorzystać potencjał Power BI do tworzenia efektywnych strategii business intelligence. Dowiesz się, jak pozyskiwać i oczyszczać dane oraz jak projektować modele danych. Zapoznasz się ze stylami wizualizacji danych i z technikami tworzenia narracji cyfrowych. Nauczysz się analizować relacje w modelu danych i tworzyć reguły w języku DAX. W tym wydaniu znajdziesz też nowy rozdział poświęcony przepływowi danych; oprócz podstaw omówiono w nim takie zagadnienia jak projektowanie efektywnych modeli danych i budowanie podstawowych pulpitów nawigacyjnych.

Najważniejsze zagadnienia:

- sposoby łączenia się z różnymi źródłami danych
- edytor Power Query i skrypty w językach M i R
- niestandardowe wizualizacje w raportach
- bezpieczeństwo pulpitów nawigacyjnych
- korzystanie z AI w przekształcaniu danych
- serwer raportów Power BI Report Server

Devin Knight jest Microsoft Data Platform MVP i autorem kilkunastu cenionych książek na temat SQL Server i BI.

Erin Ostrowsky to przedsiębiorczyni i konsultantka z zakresu Microsoft Power Platform.

Mitchell Pearson jest trenerem w zakresie platform danych i autorem książek technicznych. Zdołał doświadczenie we wdrażaniu rozwiązań BI.

Bradley Schacht pełni funkcję głównego kierownika programu w zespole Microsoft Azure Synapse Analytics.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-8322-724-5



9 788383 227245

Cena: 79,00 zł

<packt>
www.packt.com