Microsoft

Power BI i Power Pivot dla Excela

Analiza danych

Alberto Ferrari, Marco Russo

Helion

Tytuł oryginału: Analyzing Data with Power BI and Power Pivot for Excel (Business Skills)

Tłumaczenie: Zbigniew Waśko

ISBN: 978-83-289-0331-9

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)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/poblpv
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

ROZDZIAŁ 1.

Wprowadzenie do modelowania danych

Rozpoczynasz lekturę książki poświęconej modelowaniu danych. Jest to dobry moment na uświadomienie sobie, dlaczego w ogóle masz się czymś takim interesować. Przecież wiele informacji można uzyskać przez wczytanie wyników kwerendy do Excela i utworzenie na tej podstawie odpowiedniej tabeli przestawnej. Po cóż więc uczyć się czegokolwiek o modelowaniu danych?

Jako konsultanci codziennie jesteśmy proszeni o pomoc przez osoby indywidualne lub firmy, które nie potrafią wyłuskać ze swoich baz danych potrzebnych im informacji. Czują, że liczby, o które im chodzi, gdzieś tam są, ale nie umieją do nich dotrzeć, gdyż przerasta ich złożoność formuł. W 99 przypadkach na 100 przyczyną tych trudności jest jakiś błąd w modelu danych. Po jego usunięciu formuła staje się prosta i dla każdego zrozumiała. A zatem jeśli chcesz poprawić swoje umiejętności analityczne i skupić się na podejmowaniu właściwych decyzji zamiast na wynajdowaniu złożonych formuł DAX, musisz się nauczyć modelowania danych.

Dziedzina ta jest na ogół uważana za trudną i wcale nie zamierzamy twierdzić, że jest inaczej. Modelowanie danych rzeczywiście nie jest łatwe. Jest wymagające i trzeba włożyć trochę wysiłku w takie ukształtowanie swojego umysłu, aby na każde zadanie analityczne patrzeć przez pryzmat modelu danych. Tak, modelowanie danych jest skomplikowane, wymagające i ćwiczy umysł. Krótko mówiąc, niezła zabawa!

W tym rozdziale pokażemy na kilku przykładach, jak właściwy model ułatwia tworzenie odpowiednich formuł. Oczywiście, to są tylko przykłady i niekoniecznie muszą idealnie pasować do przypadków, z którymi masz do czynienia. Mamy jednak nadzieję, że dzięki nim uświadomisz sobie, dlaczego modelowanie danych jest umiejętnością, którą warto posiąść. W zasadzie dobrym modelarzem jest ten, kto potrafi dostosować swój konkretny model do jednego z wielu wzorców, które inni już opracowali i przetestowali. Twój model na pewno nie będzie się aż tak bardzo różnił od innych. Może mieć jakieś cechy szczególne, ale jest raczej mało prawdopodobne, że trafisz na problem, którego (albo bardzo podobnego) nikt wcześniej nie rozwiązał. Znajdowanie podobieństw między swoim modelem danych a modelami opisanymi w przykładach może nie jest łatwe, ale na pewno da Ci dużo satysfakcji. Gdy to opanujesz, rozwiązanie problemu, przed którym stoisz, samo się pojawi.

W większości przykładów będziemy używać bazy danych firmy Contoso. Jest to fikcyjne przedsiębiorstwo handlujące sprzętem elektronicznym na całym świecie i korzystające z różnych kanałów dystrybucji. Jeśli prowadzisz inny biznes, to po prostu dopasuj kwerendy i raporty pozyskiwane przez nas z bazy Contoso do specyfiki swojej branży.

Ponieważ to pierwszy rozdział, zaczniemy od zaprezentowania podstawowych pojęć i reguł. Objaśnimy, czym jest model danych i dlaczego relacje są ważną jego częścią. Wprowadzimy pojęcia normalizacji, denormalizacji i schematu gwiazdy. Z nowymi pojęciami wprowadzanymi przy okazji omawiania konkretnych przykładów spotkasz się w wielu miejscach książki, ale teraz, na początku, jest to najbardziej widoczne.

Zapnij pasy! Nabierz powietrza! Zanurzamy się w ocean sekretów modelowania danych.

Praca z jedną tabelą

Jeśli do analizowania danych używasz Excela i tabel przestawnych, to prawdopodobnie wczytujesz te dane z jakiegoś źródła, na przykład bazy danych, przy użyciu zapytań. Potem tworzysz stosowną tabelę przestawną i rozpoczynasz eksplorację. Oczywiście napotykasz tu typowe dla Excela ograniczenia, z których najpoważniejszym jest to, że tabela nie może liczyć więcej niż milion wierszy, gdyż taka jest pojemność excelowego arkusza. Prawdę mówiąc, gdy dowiedzieliśmy się, że takie ograniczenie istnieje, nie przywiązywaliśmy do tego większej wagi. Dlaczego, u licha, ktoś miałby wczytywać do Excela ponad milion wierszy, zamiast użyć bazy danych? Otóż dlatego, że jak łatwo się domyślić, Excel nie wymaga uciekania się do modelowania danych, a baza — tak.

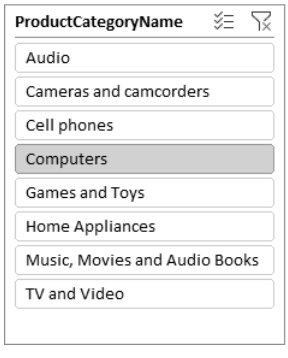
Tak czy inaczej, to pierwsze ograniczenie — jeśli zdecydujesz się na użycie Excela — może się okazać niezwykle istotne. W naszej przykładowej bazie danych firmy Contoso tabela sprzedaży zawiera 12 milionów wierszy. Nie da się więc tak po prostu załadować jej do Excela i zacząć analizować. Problem można jednak dość łatwo rozwiązać, gdyż zamiast bezkrytycznie wczytywać wszystkie wiersze, można je odpowiednio pogrupować i w ten sposób zmniejszyć ich liczbę. Na przykład, gdybyśmy chcieli przeprowadzić analizę sprzedaży według kategorii i podkategorii produktów, to moglibyśmy zrezygnować z wczytywania danych na temat sprzedaży każdego produktu z osobna na rzecz pogrupowania ich wcześniej według poszczególnych kategorii i podkategorii, co pozwoliłoby znacznie obniżyć liczbę wczytywanych wierszy.

Na przykład złożona z 12 milionów wierszy tabela sprzedaży — po pogrupowaniu danych według producenta (`Manufacturer`), marki (`BrandName`), kategorii (`ProductCategoryName`) i podkategorii (`ProductSubcategoryName`), ale z pozostawieniem podziału na poszczególne dni — kurczy się do 63 984 wierszy, które bez problemu mieszczą się w arkuszu Excela. Oczywiście takie grupowanie wymaga napisania stosownego zapytania w języku SQL, a to już jest zadanie dla działu IT lub dobrego twórcy zapytań — chyba że sam masz już jakieś doświadczenie w tej dziedzinie. Jeśli nie umiesz pisać takich zapytań, poproś o to dział informatyczny. Po otrzymaniu odpowiedniego kodu z zapytaniem SQL możesz przystąpić do analizowania uzyskanych liczb. Na rysunku 1.1 pokazano kilka pierwszych wierszy takiej właśnie tabeli po zaimportowaniu do Excela.

FullDateLabel	Manufacturer	BrandName	ProductSubcategoryName	ProductCategoryName	SalesQuantity	SalesAmount	TotalCost
2007-03-31	Adventure Works	Adventure Works	Coffee Machines	Home Appliances	55	14332.268	7651.84
2008-10-22	Contoso, Ltd	Contoso	Cell phones Accessories	Cell phones	2040	23504.88	12648.94
2009-01-31	Adventure Works	Adventure Works	Televisions	TV and Video	194	51593.106	28146.4
2009-01-21	Fabrikam, Inc.	Fabrikam	Camcorders	Cameras and camcorders	282	163007.2	76709.45
2007-12-31	Adventure Works	Adventure Works	Laptops	Computers	29	14008.43	7944.32
2007-06-22	Contoso, Ltd	Contoso	Cell phones Accessories	Cell phones	680	6107.24	3420.44
2007-06-22	Proseware, Inc.	Proseware	Projectors & Screens	Computers	86	71417.6	30786.94
2007-08-23	Adventure Works	Adventure Works	Laptops	Computers	43	22672.2	9954.6
2009-03-30	The Phone Company	The Phone Company	Touch Screen Phones	Cell phones	198	48500.37	24164.56
2008-03-24	Contoso, Ltd	Contoso	Home & Office Phones	Cell phones	306	7353.594	3914.64
2007-09-30	Fabrikam, Inc.	Fabrikam	Microwaves	Home Appliances	44	4805.604	2824.24
2007-11-13	Adventure Works	Adventure Works	Desktops	Computers	153	47357.97	28256.02
2008-12-06	Contoso, Ltd	Contoso	Projectors & Screens	Computers	32	10790.4	6477.2
2007-11-14	Contoso, Ltd	Contoso	Digital SLR Cameras	Cameras and camcorders	146	55397.5	25876
2009-12-30	Adventure Works	Adventure Works	Desktops	Computers	32	15107.75	7952.97
2009-03-13	Wide World Importers	Wide World Importers	Recording Pen	Audio	42	7990.92	3607.26
2009-08-11	Wide World Importers	Wide World Importers	Recording Pen	Audio	9	1466.1	749.16
2009-09-28	Contoso, Ltd	Contoso	Microwaves	Home Appliances	78	9955.268	5189.27
2008-02-18	A. Datum Corporation	A. Datum	Digital Cameras	Cameras and camcorders	345	70989.93	32872.58
2007-08-15	Litware, Inc.	Litware	Washers & Dryers	Home Appliances	69	112603.8	56472.35

RYSUNEK 1.1. *Dane dotyczące sprzedaży po pogrupowaniu tworzą stosunkowo małą i łatwą do analizowania tabelę*

Po wczytaniu tabeli do Excela czujesz się jak ryba w wodzie i z łatwością tworzysz tabelę przestawną, aby analizować pozyskane dane. Na rysunku 1.2 pokazano podział wartości sprzedaży (`SalesAmount`) według marki w ramach danej kategorii uzyskany za pomocą tabeli przestawnej i fragmentatora.



Etykiety wierszy	Suma z SalesAmount
Adventure Works	141178573,9
Contoso	85468758,14
Fabrikam	44940846,17
Proseware	173760754,9
Southridge Video	16092228,97
Wide World Importers	140433368,7
Suma końcowa	**601874530,7**

RYSUNEK 1.2. *Po wczytaniu danych do Excela można łatwo utworzyć tabelę przestawną*

Możesz wierzyć lub nie, ale właśnie w tym momencie zbudowałeś model danych. Tak, to prawda, że składa się on tylko z jednej tabeli, ale jest to model danych. Możesz więc rozpocząć sprawdzanie jego przydatności analitycznej, a nawet szukać sposobów na jego ulepszenie. Oczywiście model ten ma poważne ograniczenia, ponieważ zawiera mniej wierszy niż tabela źródłowa.

Jeśli dopiero zaczynasz poznawanie modelowania danych, może Ci się wydawać, że ta granica 1 miliona wierszy w arkuszu Excela ma wpływ jedynie na liczbę danych, które można wydobyć w celu przeprowadzenia analizy. Oczywiście jest to prawda, ale warto zauważyć, że wprowadza to również ograniczenie w modelu danych, a to z kolei oznacza ograniczenie możliwości analitycznych tworzonych raportów. W celu zmniejszenia liczby wierszy trzeba pogrupować dane już na poziomie źródłowym i tym samym ograniczyć ilość wydobywanych danych. W naszym przykładzie dokonaliśmy grupowania według kategorii produktu, jego podkategorii i jeszcze kilku innych kolumn.

Takie postępowanie powoduje ostatecznie ograniczenie naszych zdolności analitycznych. Na przykład, gdybyśmy chcieli przeprowadzić analizę sprzedaży produktów z uwzględnieniem ich kolorów, to ta nasza tabela okazałaby się bezużyteczna, gdyż nie zawiera kolumny z kolorami produktów. Dodanie jednej kolumny w zapytaniu kierowanym do bazy danych nie stanowi większego problemu, ale problemem jest to, że im więcej kolumn liczy tabela, tym większe stają się jej wymiary — nie tylko szerokość (liczba kolumn), ale także długość (liczba wierszy). Rzeczywiście, zamiast jednego wiersza z wynikami sprzedaży dla danej kategorii (np. Sprzęt audio) otrzymalibyśmy zestaw wierszy z wynikami sprzedaży produktów z tej samej kategorii, ale o różnych kolorach.

W skrajnym przypadku, gdybyśmy nie chcieli z góry decydować, według których kolumn dane będą filtrowane, musielibyśmy pobrać z bazy pełne 12 milionów wierszy, a takiej ich liczby nie da się „upchać" w żadnej tabeli Excela. Właśnie to mamy na myśli, gdy mówimy, że możliwości modelowania danych w Excelu są ograniczone. Brak możliwości wczytania dowolnej liczby wierszy bezpośrednio przekłada się na niemożność przeprowadzenia zaawansowanej analizy dużego zbioru danych.

W tym miejscu z pomocą przychodzi dodatek Power Pivot, który likwiduje ograniczenie miliona wierszy. Tabele tworzone za jego pomocą mogą przyjmować w zasadzie nieograniczoną ilość danych. Możemy więc wczytać pełną tabelę sprzedaży i przeprowadzić pogłębioną analizę zawartych w niej informacji.

Uwaga. Power Pivot jest dostępny w Excelu począwszy od wersji 2010 — początkowo jako dodatek zewnętrzny, a od wersji 2013 jako stały składnik tej aplikacji. Wraz z wydaniem Excela 2016 firma Microsoft zaczęła używać nowej nazwy na określenie modelu danych Power Pivota, a mianowicie: model danych programu Excel. Nazwa samego dodatku Power Pivot pozostała jednak bez zmian.

Ponieważ tabela zawiera teraz wszystkie informacje na temat sprzedaży, możemy sobie pozwolić na przeprowadzanie bardziej szczegółowych analiz. Jako przykład na rysunku 1.3 pokazujemy tabelę przestawną utworzoną na podstawie modelu danych (z Power Pivota) zawierającego wszystkie kolumny. Teraz można filtrować dane według kategorii, koloru i roku, gdyż wszystkie te informacje są w jednym miejscu. Większa liczba kolumn w tabeli oznacza większe możliwości analityczne.

ProductCategoryName
Audio
Cameras and camcorders
Cell phones
Computers
Games and Toys
Home Appliances
Music, Movies and Audio Bo...
TV and Video

Suma SalesAmount	Etykiety kolumn			
Etykiety wierszy	2007	2008	2009	Suma końcowa
Black	$59 783 936,63	$63 073 257,64	$70 489 997,35	$193 347 191,62
Blue	$5 128 405,12	$6 516 691,17	$7 715 161,42	$19 360 257,71
Brown	$6 301 722,60	$7 183 128,75	$4 904 738,33	$18 389 589,68
Gold		$39 185,40	$122 755,63	$161 941,03
Green	$1 747 237,19	$1 566 822,01	$2 159 190,14	$5 473 249,35
Grey	$6 318 419,68	$5 924 762,67	$6 410 704,05	$18 653 886,41
Orange		$12 800,63	$84 766,80	$97 567,43
Pink	$20 078,44	$43 870,83	$228 526,86	$292 476,13
Red	$6 926 045,96	$7 872 382,02	$11 495 613,50	$26 294 041,47
Silver	$43 375 399,72	$39 082 679,67	$36 471 502,90	$118 929 582,29
White	$64 963 894,39	$64 142 854,42	$70 639 615,97	$199 746 364,78
Yellow	$260 512,33	$330 165,82	$537 704,67	$1 128 382,82
Suma końcowa	**$194 825 652,07**	**$195 788 601,04**	**$211 260 277,62**	**$601 874 530,73**

RYSUNEK 1.3. *Gdy wszystkie kolumny są dostępne, można utworzyć więcej interesujących tabel przestawnych*

Już ten prosty przykład posłuży nam jako ilustracja pierwszego tematu z zakresu modelowania danych: *Rozmiar ma znaczenie, ponieważ wiąże się z ziarnistością*. Ale czymże jest ta *ziarnistość*? Otóż jest to jedno z najważniejszych pojęć, o których piszemy w tej książce, dlatego wprowadzamy je najwcześniej, jak tylko możemy. Jego dogłębne omówienie odkładamy na później, a teraz ograniczymy się do prostego objaśnienia. W pierwszym zestawie danych pogrupowaliśmy informacje na poziomach kategorii i podkategorii produktu, przez co straciliśmy trochę szczegółów na rzecz mniejszego rozmiaru. Używając bardziej technicznego języka, można by powiedzieć, że obniżyliśmy ziarnistość (granulację) danych do poziomu kategorii i podkategorii. Ziarnistość można rozumieć jako stopień szczegółowości tabel. Większa ziarnistość oznacza więcej szczegółowych informacji. Dostęp do większej ilości danych oznacza możliwość przeprowadzania bardziej szczegółowych analiz. W ostatnim zestawie danych — tym załadowanym do Power Pivota — ziarnistość jest na poziomie produktu (a nawet drobniejszym, gdyż są tam informacje o szczegółach sprzedaży poszczególnych produktów), podczas gdy w poprzednim modelu był to poziom kategorii i podkategorii. Możliwości pozyskiwania szczegółowych danych zależą od liczby kolumn w tabeli, a tym samym od jej ziarnistości. Przecież już wiesz, że zwiększanie liczby kolumn prowadzi do zwiększenia liczby wierszy w tabeli.

Wybór odpowiedniego poziomu ziarnistości nigdy nie jest prosty, a ustalenie złej granulacji w zasadzie uniemożliwia napisanie właściwych formuł. Wynika to albo z braku pewnych informacji (jak w powyższym przykładzie, w którym brakowało nam informacji o kolorach produktu), albo z ich rozproszenia po całej tabeli, jeśli ta jest zbudowana niepoprawnie. Prawdą jest też, że zwiększanie ziarnistości nie zawsze jest korzystne. Jej poziom powinien być *odpowiedni*, a odpowiedniość oznacza tu właściwe dopasowanie do potrzeb, jakiekolwiek by one były.

Przypadek utraty informacji już widziałeś, a co należy rozumieć przez jej rozproszenie? To jest trochę trudniejsze do pokazania. Wyobraź sobie, że chcesz obliczyć średni dochód roczny klientów kupujących określoną grupę Twoich produktów. Potrzebne informacje są dostępne, ponieważ w tabeli sprzedaży są zawarte wszystkie informacje o klientach. Widać to na rysunku 1.4, zawierającym część kolumn z naszej przykładowej tabeli (aby zobaczyć zawartość tabeli, należy ją otworzyć w oknie Power Pivota).

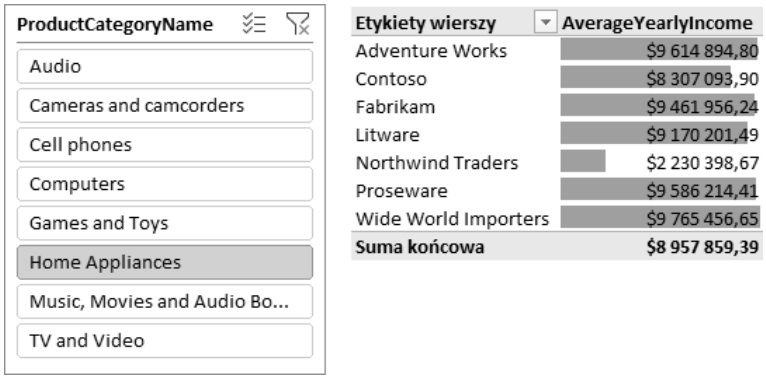
ProductCategoryName	ProductSubcategoryName	ProductName	SalesAmount	FirstName	LastName	YearlyIncome
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Katrina	Xie	€ 20,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Seth	Rodriguez	€ 80,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Evelyn	Arun	€ 10,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Christy	Beck	€ 40,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Alejandro	Nara	€ 40,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Leah	Lu	€ 30,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Robyn	Torres	€ 20,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Jimmy	Moreno	€ 30,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Rafael	Cai	€ 20,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Jenny	Ferrier	€ 110,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Levi	Arun	€ 70,000.00
Cameras and camcorders	Digital SLR Cameras	A. Datum SLR Camera X137 Grey	$627.00	Randall	Torres	€ 40,000.00

RYSUNEK 1.4. *Informacje o produktach i klientach są umieszczone w tej samej tabeli*

W każdym wierszu tej tabeli znajduje się dodatkowa informacja (w kolumnie `YearlyIncome`) o rocznych dochodach klienta, który kupił dany produkt. Prosta próba obliczenia średniego rocznego dochodu wszystkich klientów wymagałaby użycia takiej oto miary DAX:

```
AverageYearlyIncome := AVERAGE ( Sales[YearlyIncome] )
```

Miara działa dobrze i można jej użyć w tabeli przestawnej, na przykład w takiej, która pokazywałaby średnie roczne dochody klientów kupujących sprzęty AGD (`Home Appliances`) poszczególnych marek (patrz rysunek 1.5).



RYSUNEK 1.5. *Analiza średnich rocznych dochodów osiąganych przez klientów kupujących sprzęty AGD*

Raport wygląda nieźle, ale niestety wyliczona wartość jest niepoprawna — jest mocno zawyżona. To, co tutaj zostało obliczone, jest średnią z tabeli sprzedaży, która ma ziarnistość na poziomie pojedynczych aktów sprzedaży. Innymi słowy, tabela ta zawiera odrębne wiersze dla poszczególnych sprzedaży, a to oznacza, że jednemu klientowi może odpowiadać kilka wierszy. Jeśli jakiś klient kupił trzy produkty w trzech różnych dniach, to jego dochód zostanie uwzględniony trzykrotnie przy obliczaniu średniej, a tak być nie powinno.

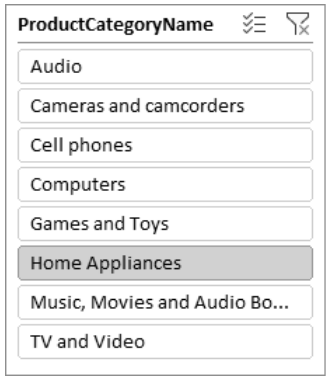
Ktoś mógłby powiedzieć, że w ten sposób wyliczana jest średnia ważona, ale jest to nieprawda. Przecież do obliczenia średniej ważonej potrzebne jest zdefiniowanie wagi, a w tym przypadku nie może nią być liczba zakupów. Mogłaby to być liczba kupionych produktów, łączna kwota wydana na zakupy lub jakaś inna równie sensowna wielkość. W naszym przykładzie chodziło po prostu o wyliczenie zwykłej średniej, ale zastosowana miara niezbyt się do tego nadaje.

Na pierwszy rzut oka może trudno to zauważyć, ale mamy tu jeszcze problem z niewłaściwą ziarnistością. Informacje o dochodach klientów są dostępne, ale są rozrzucone po całej tabeli sprzedaży, zamiast przypisane do poszczególnych klientów, co znacznie utrudnia przeprowadzanie obliczeń. Aby uzyskać właściwą średnią, należałoby ustalić ziarnistość na poziomie klienta, bądź to przez ponowne załadowanie tabeli, bądź przez zastosowanie bardziej rozbudowanej formuły DAX.

Właściwa formuła DAX powinna wyglądać tak:

```
CorrectAverage :=
AVERAGEX (
    SUMMARIZE (
        Sales,
        Sales[CustomerKey],
        Sales[YearlyIncome]
    ),
    Sales[YearlyIncome]
)
```

Jest trochę skomplikowana, ponieważ najpierw trzeba zagregować sprzedaż na poziomie klienta (ziarnistość) i dopiero potem zastosować funkcję `AVERAGE` w odniesieniu do uzyskanej w ten sposób tabeli, gdzie każdy klient występuje tylko raz. Funkcja `SUMMARIZE` przeprowadza wspomnianą agregację na poziomie klienta i tworzy tymczasową tabelę, w której potem jest przeprowadzane uśrednianie kolumny `YearlyIncome`. Jak wynika z rysunku 1.6, poprawne wartości średnich dochodów rocznych (kolumna `CorrectAverage`) różnią się znacznie od tych, które uzyskaliśmy poprzednio.



Etykiety wierszy	AverageYearlyIncome	CorrectAverage
Adventure Works	$9 614 894,80	$535 593,62
Contoso	$8 307 093,90	$262 307,94
Fabrikam	$9 461 956,24	$361 924,73
Litware	$9 170 201,49	$265 677,30
Northwind Traders	$2 230 398,67	$151 583,50
Proseware	$9 586 214,41	$491 908,56
Wide World Importers	$9 765 456,65	$1 035 131,95
Suma końcowa	**$8 957 859,39**	**$260 183,91**

RYSUNEK 1.6. *Zestawienie obok siebie wartości poprawnych z niepoprawnymi uświadamia skalę popełnianego błędu*

Zatrzymajmy się jeszcze przez chwilę przy tym prostym przykładzie, abyś dobrze zrozumiał to, że roczny dochód klienta kupującego dany produkt ma sens tylko przy ziarnistości na poziomie klienckim. Rozpatrywanie tej wartości na poziomie poszczególnych sprzedaży jest niewłaściwe. Mówiąc bardziej ogólnie, nie można używać danej wartości w takim samym znaczeniu przy granulacji na poziomie klienta i przy granulacji na poziomie sprzedaży. Aby uzyskać poprawny rezultat, należy zmienić ziarnistość — przynajmniej w tabeli tymczasowej.

Na podstawie zaprezentowanego przykładu możemy sformułować następujące wnioski:

- Właściwa formuła jest bardziej złożona niż prosta funkcja `AVERAGE`. Należy przeprowadzić tymczasową agregację, aby zmienić ziarnistość tabeli i spowodować uporządkowanie danych.

- Bardzo łatwo można przeoczyć tego typu błędy, jeśli się nie wie, z jakiego rodzaju danymi ma się do czynienia. Patrząc na rysunek 1.5, można łatwo zauważyć, że wyliczone średnie dochody klientów są zbyt wysokie, aby mogły być prawdziwe — jest raczej niemożliwe, by żaden z klientów nie zarabiał rocznie mniej niż 2 miliony dolarów! Jednakże przy bardziej skomplikowanych obliczeniach wychwycenie takiego błędu może być trudniejsze i może skończyć się sporządzeniem raportu zawierającego błędne informacje.

Do sporządzenia raportu o należytej szczegółowości należy zwiększyć ziarnistość danych, ale zbyt duże jej zwiększenie może utrudnić wydobycie pewnych informacji. Jak zatem wybrać właściwą ziarnistość? Odpowiedź na to pytanie nie jest wcale łatwa. Spróbujemy jej udzielić, ale nieco później. Liczymy na to, że uda nam się nauczyć Cię dobierania właściwej ziarnistości danych w swoich modelach, ale pamiętaj, że jest to umiejętność trudna do opanowania nawet dla doświadczonych modelarzy danych. Na razie poprzestańmy na wyjaśnieniu, czym jest ta ziarnistość i dlaczego tak ważne jest jej właściwe ustalenie dla każdej tabeli w określonym modelu danych.

Prawdę mówiąc, model, z którym teraz mamy do czynienia, ma jeszcze większą wadę, w pewnym sensie też powiązaną z ziarnistością danych. Polega ona na tym, że wszystkie informacje są tu zebrane w jednej tabeli. Jeśli model zawiera tylko jedną tabelę, to przy wyborze jej ziarnistości trzeba wziąć pod uwagę wszystkie cięcia i rzuty, jakie ktoś zechce wykonać. To oczywiście oznacza, że mimo usilnych starań nigdy nie uda Ci się wybrać takiej ziarnistości, jaka byłaby idealnie dopasowana do wszystkich przewidywanych pomiarów. W następnych podrozdziałach pokażemy metody posługiwania się wieloma tabelami, pozwalające korzystać z wielu poziomów ziarnistości.

Wprowadzamy model danych

Z poprzedniego podrozdziału wiesz, że model z jedną tabelą utrudnia wybór właściwej ziarnistości danych. Użytkownicy Excela często posługują się modelami jednotabelowymi, a robią to z przyzwyczajenia, ponieważ przed wersją 2013 była to jedyna możliwość tworzenia tabel przestawnych. Dopiero w Excelu 2013 Microsoft wprowadził pojęcie modelu danych i pozwolił wczytywać wiele tabel, by po połączeniu ich relacjami zyskać możliwość analizowania ich na różne sposoby.

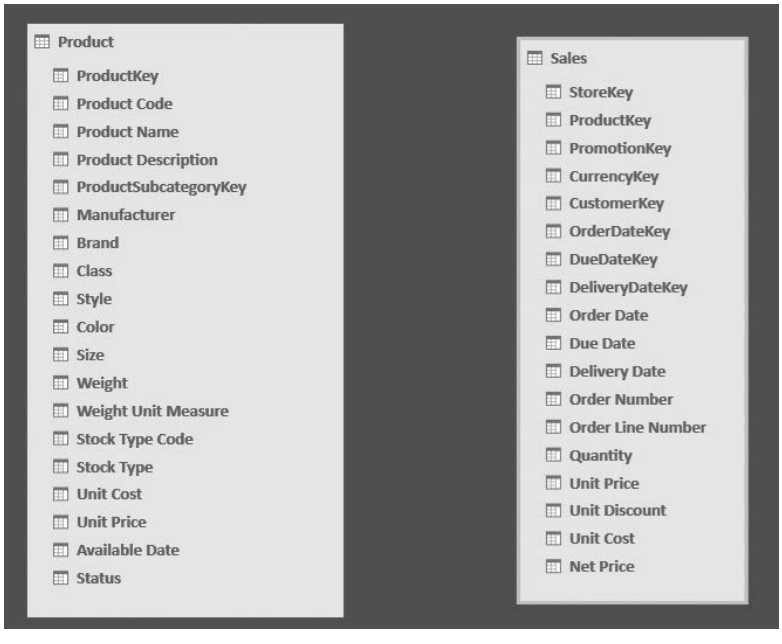
Czym jest *model danych*? Mówiąc najprościej, jest to zbiór tabel powiązanych za pomocą relacji. Jedna tabela też jest swoistym modelem danych, ale niezbyt interesującym. Znacznie ciekawsze pod względem możliwości analitycznych są jednak modele wielotabelowe.

Po wczytaniu więcej niż jednej tabeli naturalną rzeczą staje się budowanie modelu danych. Poza tym dane zawarte w tych tabelach zazwyczaj pochodzą z baz danych zarządzanych przez fachowców, którzy już przygotowali stosowny model. Twój model będzie po prostu odwzorowaniem modelu istniejącego w bazie danych, a to znacznie uprości Twoją pracę.

Niestety, rzadko się zdarza, że model danych źródłowych jest perfekcyjnie dopasowany do analiz, które chcemy wykonać. Zamierzamy na praktycznych przykładach o coraz większym stopniu trudności nauczyć Cię konstruowania własnych modeli z dowolnych źródeł danych. Aby uprościć proces uczenia, w kolejnych rozdziałach będziemy stopniowo omawiać poszczególne techniki, a na razie zaczniemy od zagadnień najbardziej elementarnych.

Żeby zapoznać się z pojęciem modelu danych, wczytaj tabele produktu (`Product`) i sprzedaży (`Sales`) z bazy danych Contoso do excelowego modelu. Po załadowaniu tych tabel ujrzysz diagram pokazany na rysunku 1.7. Diagram ten przedstawia dwie tabele z wyszczególnionymi kolumnami.

Uwaga. Wspomniany diagram jest dostępny w dodatku Power Pivot. Aby go wyświetlić, kliknij na wstążce Excela kartę *Power Pivot*, a następnie wybierz narzędzie *Zarządzaj*. Potem na karcie *Narzędzia główne* wstążki Power Pivota kliknij w grupie *Widok* ikonę *Widok diagramu*.



RYSUNEK 1.7. *Posługiwanie się modelem danych umożliwia załadowanie wielu tabel*

Dwie niepołączone ze sobą tabele, jak w powyższym przykładzie, nie stanowią jeszcze modelu danych. Są tylko dwiema tabelami. Aby uczynić z nich sensowny model, trzeba je połączyć relacją. Jak widać, obie tabele zawierają kolumnę o nazwie `ProductKey` (klucz produktu). W tabeli produktu jest to *klucz podstawowy* (lub *główny*), czyli taki, którego wartości w poszczególnych wierszach pozwalają jednoznacznie zidentyfikować każdy produkt. W tabeli sprzedaży kolumna `ProductKey` służy do innych celów, a mianowicie do identyfikacji produktu sprzedanego.

Informacja. Klucz podstawowy tabeli to kolumna, która w każdym wierszu ma inną wartość. A zatem znajomość konkretnej wartości z tej kolumny pozwala jednoznacznie wskazać wiersz, w którym ta wartość się znajduje. W tabeli może być więcej kolumn o niepowtarzających się wartościach; każda z nich jest kluczem. Klucz podstawowy nie jest niczym szczególnym. Z technicznego punktu widzenia jest to po prostu kolumna, którą uznano za przydatną do jednoznacznego identyfikowania wierszy. Na przykład w tabeli klientów kluczem podstawowym jest kod klienta, chociaż mogłoby się zdarzyć, że również w kolumnie z nazwiskami każda wartość byłaby inna.

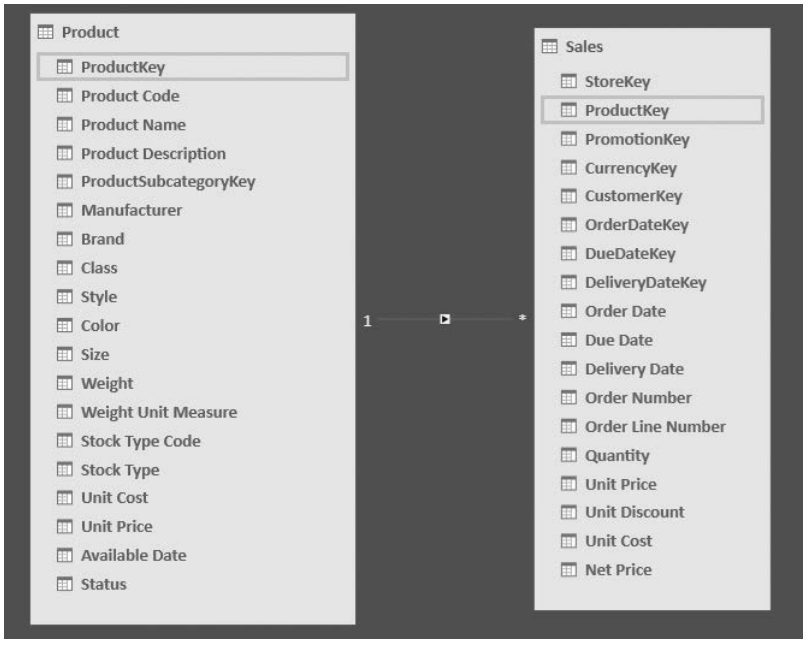
Jeśli w jednej tabeli istnieje unikatowy identyfikator, a w drugiej jest kolumna z wartościami tego identyfikatora, to można utworzyć relację łączącą obie tabele. Oba warunki muszą być spełnione, aby relacja była poprawna. Jeśli model zawiera relację z kluczem niespełniającym warunku unikatowości w żadnej z obu tabel, to trzeba taki model poddać obróbce z użyciem technik, o których będzie mowa w dalszej części książki. Na razie skupmy się na kilku faktach związanych z naszą przykładową relacją.

- **Tabela sprzedaży jest tu tabelą *źródłową*.** Relacja zaczyna się w tabeli sprzedaży. Kierunek jest właśnie taki dlatego, że w celu uzyskania informacji o sprzedanym produkcie pobieramy wartość klucza produktu z tabeli sprzedaży i odszukujemy identyczną wartość w tabeli produktu. Odszukanie jej pozwala nam odczytać wszystkie atrybuty owego produktu.
- **Tabela produktu jest tu *celem* relacji.** Poszukiwanie przeprowadzane w ramach relacji zaczyna się w tabeli sprzedaży i kończy w tabeli produktu. Można więc powiedzieć, że tabela produktu jest celem poszukiwań.
- **Relacja zaczyna się w źródle i zmierza do celu.** Z tego wynika, że relacja ma kierunek. Dlatego często jest symbolizowana przez strzałkę zwróconą od źródła w stronę celu. Różne aplikacje stosują różne sposoby graficznego przedstawiania relacji.
- **W kontekście relacji tabela źródłowa jest często nazywana *stroną wiele* lub *wielowartościową*.** To dlatego, że dla dowolnie wybranego produktu może istnieć wiele transakcji sprzedaży, podczas gdy dla danej sprzedaży istnieje tylko jeden produkt. Dlatego też tabela docelowa jest określana jako *strona jeden* lub *jednowartościowa*. My będziemy używać najczęściej określeń *strona wiele* i *strona jeden*.
- **Kolumna `ProductKey` istnieje w obu tabelach: sprzedaży i produktu.** Kolumna ta jest kluczem w tabeli produktu, ale nie w tabeli sprzedaży. Dlatego w tabeli produktu jest określana mianem klucza podstawowego, a w tabeli sprzedaży — jako klucz obcy. *Klucz obcy* to kolumna odwołująca się do klucza podstawowego w innej tabeli.

Wszystkie wymienione tu określenia są powszechnie używane w świecie modelowania danych, dlatego warto je zapamiętać. Skoro już wiadomo, co oznaczają, my też będziemy się nimi posługiwać. Nie przejmuj się jednak. W pierwszych kilku rozdziałach będziemy te definicje powtarzać wielokrotnie, abyś mógł je dobrze zrozumieć i przyswoić.

Używając Excela lub aplikacji Power BI, możesz tworzyć relacje między tabelami przez zwykłe przeciąganie klucza obcego (w naszym przykładzie `ProductKey` w tabeli `Sales`) na klucz podstawowy (`ProductKey` w tabeli `Product`). Jeśli tak zrobisz, z pewnością zauważysz, że zarówno Excel, jak i Power BI tworzą graficzny symbol relacji w postaci nie strzałki, lecz linii zakończonej po jednej stronie cyfrą 1 (strona *jeden*), a po drugiej gwiazdką (strona *wiele*). Na rysunku 1.8 pokazano, jak to wygląda w oknie Power Pivota przy włączonym widoku diagramu. Zauważ, że tu obecna jest również strzałka (na środku linii łączącej tabele), ale nie reprezentuje ona kierunku relacji, lecz kierunek propagacji filtra i ma zupełnie inny cel, o którym opowiemy w dalszej części książki.

Uwaga. Jeśli ze wstążki Excela znikła karta *Power Pivot*, to prawdopodobnie Excel napotkał jakiś problem i wyłączył wszystkie dodatki. Aby ponownie włączyć Power Pivota, otwórz kartę *Plik* i w lewym panelu kliknij pozycję *Opcje*. W lewej części okna *Opcje programu Excel* wybierz pozycję *Dodatki*. Następnie w dolnej części okna rozwiń listę *Zarządzaj*, wybierz z niej pozycję *Dodatki COM* i kliknij przycisk *Przejdź*. W oknie dialogowym *Dodatki COM* zaznacz pozycję *Microsoft Power Pivot for Excel*, a jeśli jest zaznaczona, to usuń zaznaczenie. Kliknij przycisk *OK*. Jeśli usunąłeś zaznaczenie dodatku Power Pivot, ponownie otwórz okno *Dodatki COM* i zaznacz ten dodatek. Po zatwierdzeniu wprowadzonych zmian karta *Power Pivot* powinna znów być widoczna na wstążce Excela.



RYSUNEK 1.8. *Relacja łącząca tabele produktu i sprzedaży jest reprezentowana graficznie przez linię z oznaczonymi końcami (cyfra 1 po stronie „jeden" i gwiazdka po stronie „wiele")*

Po ustanowieniu relacji można przeprowadzić sumowanie wartości w tabeli sprzedaży z rozbiciem na poszczególne wartości z wybranej kolumny tabeli produktu. Przykład takiego sumowania pokazano na rysunku 1.9. Wykonano tu sumowanie ilości sprzedanych produktów (kolumna `Quantity` w tabeli `Sales` — patrz rysunek 1.8) z rozbiciem na poszczególne kolory produktu (kolumna `Color` w tabeli `Product`).

Etykiety wierszy	Suma Quantity
Azure	60
Black	4307
Blue	985
Brown	453
Gold	155
Green	374
Grey	1551
Orange	179
Pink	600
Purple	10
Red	896
Silver	3604
Silver Grey	143
Transparent	141
White	3746
Yellow	294
Suma końcowa	**17498**

RYSUNEK 1.9. *Ustanowienie relacji umożliwia dzielenie wartości z jednej tabeli według wartości z drugiej*

Na początku tego rozdziału powiedzieliśmy, że bardzo ważne — ale też skomplikowane — jest określenie właściwej ziarnistości w przypadku posługiwania się jedną tabelą. Zły wybór skutkuje znacznymi utrudnieniami w przeprowadzaniu obliczeń. A co z ziarnistością w nowym modelu złożonym z dwóch tabel? Problem ma trochę inny charakter i choć jest trudniejszy pojęciowo, to jednak daje się stosunkowo łatwo rozwiązać.

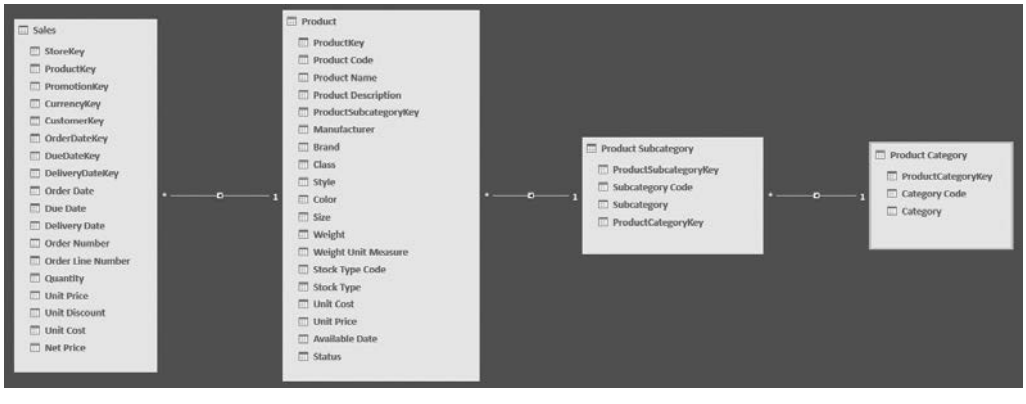
Mamy teraz do czynienia z dwiema tabelami, a więc także z dwiema różnymi ziarnistościami. Tabela `Sales` ma ziarnistość na poziomie pojedynczej sprzedaży, a ziarnistość tabeli `Product` jest na poziomie produktu. Ujmując rzecz ściśle, ziarnistość jest pojęciem odnoszącym się do tabeli, a nie całego modelu. Jeśli model zawiera kilka tabel, to trzeba ustalić ziarnistość każdej z nich. Na pierwszy rzut oka wydaje się to bardziej skomplikowane niż przypadek jednej tabeli, ale w rzeczywistości jest łatwiejsze do opanowania i tak naprawdę ziarnistość przestaje tu być problemem.

W naszym przykładzie z dwiema tabelami sprzedaży i produktu całkiem naturalne jest ustalenie ziarnistości tych tabel na poziomie, odpowiednio, pojedynczej sprzedaży i produktu. Wróćmy na chwilę do pierwszego przykładu w tym rozdziale. Mieliśmy tam do czynienia z jedną tabelą zawierającą sprzedaż z granulacją na poziomie kategorii i podkategorii produktów. Tak było, ponieważ kategorie i podkategorie produktów były przechowywane w tabeli sprzedaży. Mówiąc inaczej, *podejmowanie decyzji w sprawie ziarnistości było konieczne głównie dlatego, że sposób przechowywania informacji był niewłaściwy*. Jeśli każda porcja informacji trafia we właściwe miejsce, ziarnistość w zasadzie przestaje być problemem.

Przecież kategoria produktu jest atrybutem produktu, a nie pojedynczego aktu sprzedaży. Owszem, w pewnym sensie jest atrybutem również sprzedaży, ale tylko dlatego, że sprzedaż wiąże się z produktem. Gdy wstawimy do tabeli sprzedaży klucz produktu, wszystkie potrzebne nam atrybuty produktu (kategoria, podkategoria, kolor itp.) uzyskamy dzięki relacji łączącej obie tabele. A zatem skoro nie musimy zapisywać kategorii produktu w tabeli sprzedaży, problem ziarnistości właściwie przestaje istnieć. Oczywiście to samo dotyczy pozostałych atrybutów produktu — koloru, ceny jednostkowej, nazwy i wszystkich innych kolumn tabeli `Product`.

Informacja. W prawidłowo zaprojektowanym modelu danych ziarnistość każdej tabeli jest na odpowiednim poziomie, a to oznacza, że jego struktura jest zarazem prosta i wydajna. Na tym polega siła relacji, którą możesz wykorzystać, jeśli tylko zaczniesz myśleć w kategoriach wielu tabel i porzucisz podejście jednotabelowe, jakie dawniej obowiązywało w Excelu.

Jeśli przyjrzysz się uważnie tabeli produktu, to okaże się, że nie ma w niej kolumn z kategorią i podkategorią, ale jest kolumna o nazwie `ProductSubcategoryKey` (klucz podkategorii produktu), sugerującej, że jest to powiązanie (jako klucz obcy) z kluczem w innej tabeli (jako kluczem podstawowym), zawierającej podkategorie produktu. Rzeczywiście, w bazie danych znajdują się jeszcze dwie tabele z kategoriami i podkategoriami produktów. Po ich załadowaniu do modelu i utworzeniu odpowiednich relacji uzyskujemy strukturę pokazaną na rysunku 1.10 (zrzut okna Power Pivota w trybie *Widok diagramu*).



RYSUNEK 1.10. *Kategorie i podkategorie produktu są przechowywane w innych tabelach i dostępne za pośrednictwem relacji*

Jak widać, informacje o produkcie są przechowywane w trzech różnych tabelach: `Product` (produkt), `Product Subcategory` (podkategoria produktu) i `Product Category` (kategoria produktu). Tabele te są połączone łańcuchem relacji zaczynającym się od tabeli `Product` i kończącym na tabeli `Product Category`.

Po co to wszystko? Czy nie próbujemy niepotrzebnie skomplikować przechowywania prostej informacji? Może tego nie widać na pierwszy rzut oka, ale zaprezentowana technika ma naprawdę wiele zalet. Przez umieszczenie kategorii produktu w odrębnej tabeli zyskujemy model danych, w którym nazwa kategorii, choć odnosi się do wielu produktów, jest przechowywana w jednym wierszu tabeli `Product Category`. Taki sposób przechowywania informacji jest korzystny z dwóch powodów. Po pierwsze, zmniejsza ilość miejsca na dysku zajmowanego przez nasz model, ponieważ eliminuje liczne powtórzenia tej samej nazwy. Po drugie, jeśli będzie trzeba wprowadzić zmiany w nazwie kategorii, wystarczy zrobić to w jednym miejscu — każda modyfikacja będzie automatycznie „widoczna" dla wszystkich produktów za pośrednictwem relacji.

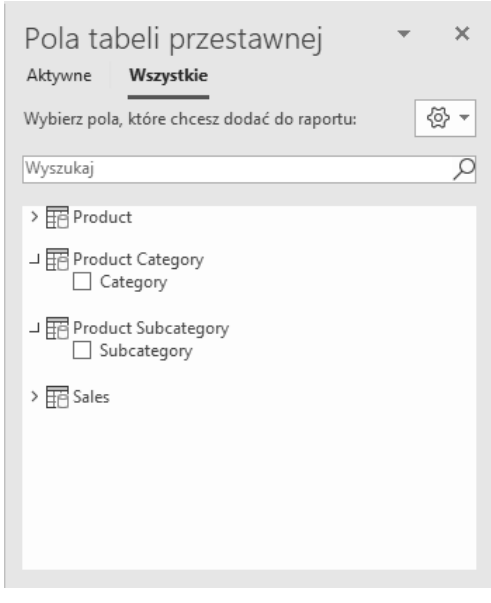
Opisaną technikę zwykło się nazywać *normalizacją*. Atrybut, taki jak kategoria produktu, uważa się za znormalizowany, jeśli jest przechowywany w odrębnej tabeli, a jego miejsce zajmuje klucz skojarzony z tamtą tabelą. Technika ta jest dobrze znana i często stosowana przez projektantów baz danych na etapie opracowywania modelu danych. Technika odwrotna — przechowywanie atrybutów w tabeli, do której należą — jest nazywana *denormalizacją*. W modelu zdenormalizowanym ten sam atrybut występuje wielokrotnie i gdy trzeba coś w nim zmienić, to należy zmodyfikować wszystkie wiersze, w których występuje. Na przykład kolor produktu jest zdenormalizowany, ponieważ łańcuch `Red` (czerwony) występuje we wszystkich wierszach z informacjami o produktach czerwonych.

Być może zastanawiasz się teraz, dlaczego twórca bazy danych Contoso postanowił umieścić kategorię i podkategorię produktu w odrębnych tabelach (przeprowadził normalizację), a kolor, producenta i markę pozostawił w tabeli produktów (zdenormalizował te atrybuty). W tym konkretnym przypadku odpowiedź jest prosta: Contoso jest bazą przykładową i jej struktura została specjalnie tak dobrana, aby można było zilustrować różne techniki budowania modelu danych. W świecie rzeczywistym stosuje się bazy danych wysoce znormalizowane lub mocno zdenormalizowane w zależności od ich przeznaczenia. Zawsze trzeba być przygotowanym na to, że znajdą się jakieś atrybuty znormalizowane i zdenormalizowane. Nie ma w tym nic dziwnego, gdyż w modelowaniu danych istnieje mnóstwo różnych możliwości. Ponadto różne czynniki wpływają na decyzje podejmowane przez projektantów baz danych.

Strukturami mocno znormalizowanymi są zazwyczaj systemy przetwarzania transakcyjnego (OLTP). Są to bazy danych o charakterze operacyjnym zaprojektowane pod kątem wykonywania codziennych zadań, takich jak sporządzanie faktur, składanie zamówień, wysyłanie towarów czy obsługa reklamacji. Stopień normalizacji tych baz jest wysoki, ponieważ mają zajmować jak najmniej przestrzeni dyskowej (co zazwyczaj oznacza również szybsze działanie) i wykonywać dużo operacji wstawiania i aktualizowania danych. Rzeczywiście, podczas codziennej pracy w firmie często przeprowadza się aktualizację informacji — na przykład o kliencie — i oczekuje, że ta aktualizacja zostanie automatycznie uwzględniona we wszystkich miejscach odwołujących się do tej informacji. Jeśli informacja jest należycie znormalizowana, to wszystko przebiega gładko i zgodnie z oczekiwaniami. Nagle wszystkie zamówienia od danego klienta zawierają nowe, zmodyfikowane dane. Gdyby informacje o tym kliencie były zdenormalizowane, aktualizacja jego adresu wymagałaby wykonania setek operacji na serwerze, a to oznaczałoby pogorszenie wydajności systemu.

Systemy OLTP często składają się z setek tabel, ponieważ prawie każdy atrybut jest umieszczany w odrębnej tabeli. Na przykład w przypadku produktów najpewniej jedna tabela będzie przeznaczona dla producenta, jedna dla marki, jedna dla koloru itd. A zatem nawet tak prosta encja jak produkt może zajmować 10 lub 20 tabel połączonych relacjami. To właśnie coś takiego projektanci baz danych z dumą nazywają „dobrze zbudowanym modelem danych" i, choć może się to wydawać dziwne, słusznie widzą w tym powód do dumy. W przypadku baz danych OLTP normalizacja jest niemal zawsze przydatna.

Chodzi o to, że gdy analizujemy dane, to niczego nie wstawiamy ani nie aktualizujemy. Interesuje nas wyłącznie odczytywanie informacji. Gdy w grę wchodzi samo odczytywanie, normalizacja raczej nie jest wskazana. Załóżmy na przykład, że chcemy utworzyć tabelę przestawną w ramach poprzedniego modelu danych. Lista pól będzie wyglądała tak jak na rysunku 1.11.



RYSUNEK 1.11. *Lista pól w modelu znormalizowanym zawiera zbyt wiele tabel*

Informacje o produktach są przechowywane w trzech tabelach, co widać na liście pól (w panelu *Pola tabeli przestawnej*). Co gorsza, tabele `Product Category` i `Product Subcategory` zawierają tylko po jednej kolumnie. Tak więc o ile normalizacja jest pożądana w systemach OLTP, o tyle w systemach analitycznych jest raczej niewskazana. Gdy na potrzeby raportu wycinamy i rzutujemy dane, nie obchodzi nas techniczna reprezentacja produktu — wolimy widzieć kategorię i podkategorię jako kolumny tabeli `Product`, gdyż to umożliwia przeglądanie danych w sposób naturalny.

Uwaga. W prezentowanym przykładzie rozmyślnie ukryliśmy kilka niemających znaczenia kolumn (np. podstawowe klucze tabel) i tak zawsze powinno się robić. Niepotrzebne zwiększanie liczby widocznych kolumn tylko utrudnia orientację w całym modelu. Łatwo sobie wyobrazić listę pól z dziesiątkami tabel — znalezienie w niej kolumn potrzebnych do sporządzenia raportu mogłoby zająć sporo czasu.

Podsumowując, gdy budujemy model danych mający służyć do sporządzania raportów, należy zadbać o właściwy poziom denormalizacji bez względu na to, jak pobierane dane są przechowywane. Wiesz już, że zbyt daleko posunięta denormalizacja może oznaczać kłopoty z ziarnistością. Później pokażemy, że model nadmiernie zdenormalizowany ma jeszcze kilka innych wad. Jaki jest zatem właściwy poziom denormalizacji?

Nie ma jednoznacznej reguły mówiącej, jak należy ustalić optymalny poziom denormalizacji. Intuicyjnie wyczuwamy jednak, że należy denormalizować do momentu, w którym tabela staje się zwartą strukturą opisującą w sposób kompletny to, co ma opisywać. W naszym przykładzie należałoby przenieść kolumny `Category` i `Subcategory` do tabeli `Product`, gdyż są to atrybuty produktu i nie powinny się znajdować w odrębnych tabelach. Denormalizacja nie powinna jednak objąć produktu w tabeli `Sales`, ponieważ produkty i sprzedaże to dwa różne zbiory informacji. Sprzedaż wiąże się z produktem, ale nie da się jednoznacznie zidentyfikować sprzedaży za pomocą produktu.

Po tym wszystkim, co powiedzieliśmy, pewnie uważasz, że model z jedną tabelą jest przedenormalizowany. I to jest prawda. W przypadku tabeli sprzedaży musieliśmy zwracać uwagę na jej ziarnistość, a tak nie powinno być. Jeśli model jest dobrze zaprojektowany, z właściwym poziomem denormalizacji, to ziarnistością nie trzeba się zajmować. Problemy z nią pojawiają się w modelach, które zdenormalizowano zbyt mocno.

Schemat gwiazdy

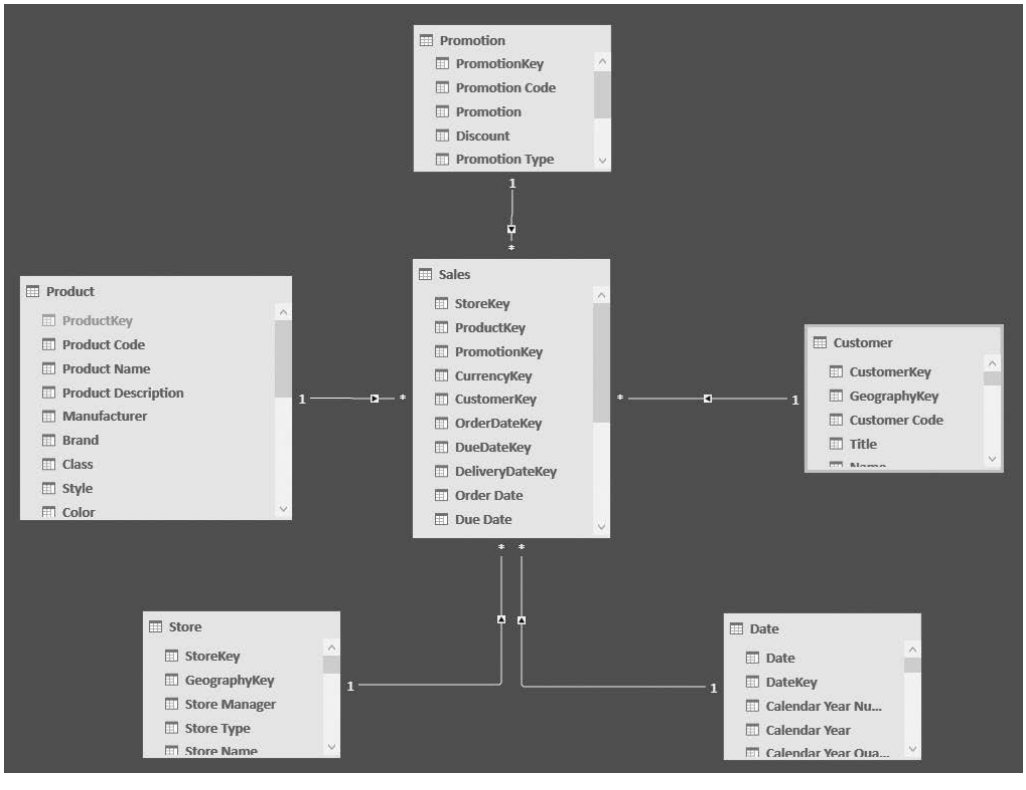
Dotąd zajmowaliśmy się bardzo prostym modelem danych, obejmujących tylko produkty i transakcje sprzedaży. W świecie rzeczywistym raczej rzadko można spotkać modele o tak niskim stopniu złożoności. W typowej firmie, jaką jest Contoso, jest więcej zasobów istotnych z punktu widzenia informacyjnego — produkty, sklepy, pracownicy, klienci i czas. Zasoby te są wzajemnie powiązane i generują rozmaite zdarzenia. Na przykład pracownik określonego sklepu sprzedaje określony produkt określonemu klientowi w określonym czasie.

Oczywiście, różne firmy dysponują różnymi zasobami i generują różne zdarzenia. Jeśli jednak myśli się w kategoriach ogólnych, to niemal zawsze można dostrzec wyraźne rozdzielenie zasobów od zdarzeń. Taka struktura powtarza się w każdym biznesie, nawet jeśli zasoby są zupełnie inne. Na przykład w branży medycznej zasobami mogą być pacjenci, choroby i kuracje, a zdarzeniem może być zdiagnozowanie określonego pacjenta z określoną chorobą i skierowanie go na określoną kurację w celu wyleczenia. Pomyśl przez chwilę o swoim biznesie w tym kontekście. Prawdopodobnie bez trudu zdołasz oddzielić zasoby od zdarzeń.

To rozróżnienie na zasoby i zdarzenia leży u podstaw techniki modelowania danych zwanej *schematem gwiazdy*. Jej istotną cechą jest podział encji (tabel) na dwie kategorie:

- **Wymiary** — będące określonymi zasobami informacyjnymi, takimi jak produkt, klient, pracownik czy pacjent. Wymiary mają swoje atrybuty. Na przykład produkt może mieć takie atrybuty, jak kolor, kategoria, podkategoria, producent i cena. Dla pacjenta atrybutami mogą być nazwisko, adres i data urodzenia.
- **Fakty** — będące zdarzeniami charakteryzującymi się określonymi wymiarami. W modelu Contoso faktem jest sprzedaż produktu. Aby opisać sprzedaż, trzeba określić produkt, klienta, datę i inne wymiary. Fakty mają swoje miary, czyli liczby, które można agregować w celu uzyskania określonych informacji na temat firmy. Miarą sprzedaży może być ilość sprzedanych produktów, ich wartość, wielkość upustu itp.

Po podzieleniu tabel na te dwie kategorie staje się jasne, że fakty są powiązane z wymiarami. Jednemu produktowi odpowiada wiele sprzedaży. Innymi słowy, istnieje relacja łącząca tabele `Sales` i `Product`, w której po stronie *wiele* jest tabela `Sales`, a po stronie *jeden* — tabela `Product`. Jeśli przy projektowaniu tego modelu ułożysz wszystkie wymiary wokół jednej tabeli faktów, otrzymasz układ w kształcie gwiazdy, taki jak na rysunku 1.12, przedstawiającym okno Power Pivota w trybie *Widok diagramu*.



RYSUNEK 1.12. *Schemat modelu przyjmuje kształt gwiazdy, jeśli tabelę faktów ustawi się na środku, a tabele wymiarów wokół niej*

Schemat gwiazdy jest czytelny, łatwy do zrozumienia i wygodny w użyciu. Za pomocą wymiarów można wycinać i rzutować dane, a gdy chcemy je agregować, używamy tabeli faktów. Tego typu schematy generują stosunkowo krótkie listy pól tabeli przestawnych.

Uwaga. Schematy typu gwiazda stały się bardzo popularne wśród twórców hurtowni danych. Obecnie są uważane za standardowy sposób reprezentowania modeli analitycznych.

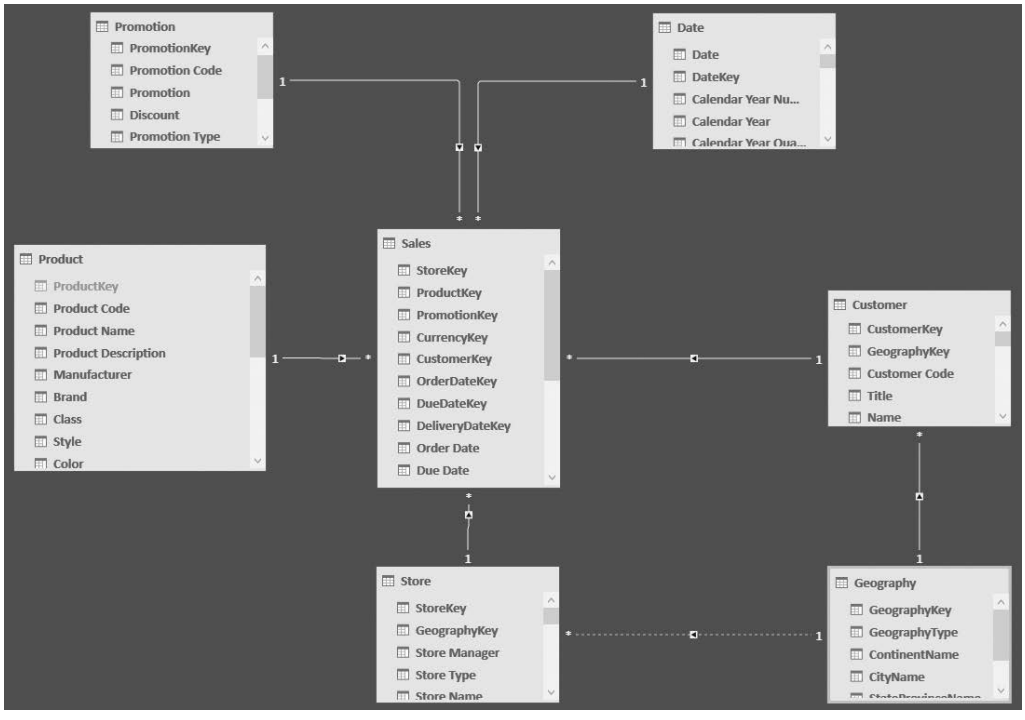
Wymiary z natury są raczej niewielkimi tabelami, mającymi nie więcej niż milion wierszy — na ogół są to liczby rzędu kilkuset lub tysiąca. Znacznie większe są tabele faktów. Te mogą zawierać dziesiątki — jeśli nie setki — milionów wierszy. Schemat gwiazdy jest jednak tak popularny, że opracowano już dla niego rozmaite mechanizmy optymalizujące, które pozwalają usprawnić funkcjonowanie systemu bazodanowego.

Wskazówka. Przerwij na chwilę czytanie tej książki i spróbuj ustalić, jak powinien wyglądać schemat gwiazdy dla Twojego modelu biznesowego. Nie musisz od razu budować perfekcyjnego schematu ze wszystkimi szczegółami, ale sama próba wykonania takiego ćwiczenia z pewnością skłoni Cię do szukania lepszych sposobów konstruowania tabel faktów i wymiarów.

Ważne jest, by oswoić się ze schematem gwiazdy, gdyż jest to naprawdę wygodny sposób reprezentowania danych. Poza tym warto się zapoznać z pojęciami związanymi z tego typu schematami, ponieważ w świecie analityki biznesowej (BI) są one powszechnie stosowane (podobnie jak w naszej książce). Często piszemy o tabelach faktów i wymiarów, aby podkreślić różnice między tymi tabelami (dużymi i małymi). W następnym rozdziale przejdziemy do omawiania tabel nagłówek/treść, w związku z czym skupimy się na tworzeniu relacji między tabelami faktów, a do tego będzie potrzebne dobre rozumienie wspomnianych różnic.

Oto kilka istotnych szczegółów na temat schematu gwiazdy, o których warto pamiętać. Jednym z nich jest to, że tabele faktów są związane z wymiarami, ale wymiary nie powinny mieć żadnych powiązań między sobą. Abyś się przekonał, do czego może prowadzić nieprzestrzeganie tej reguły, dodajmy jeszcze jeden wymiar, o nazwie `Geography` (geografia), z informacjami o położeniu geograficznym, takimi jak nazwa miasta, województwa, kraju itp. Wymiary `Store` (sklep) i `Customer` (klient) można w oczywisty sposób powiązać z tym nowym wymiarem. Moglibyśmy więc zbudować taki model danych zgodny z diagramem pokazanym na rysunku 1.13.

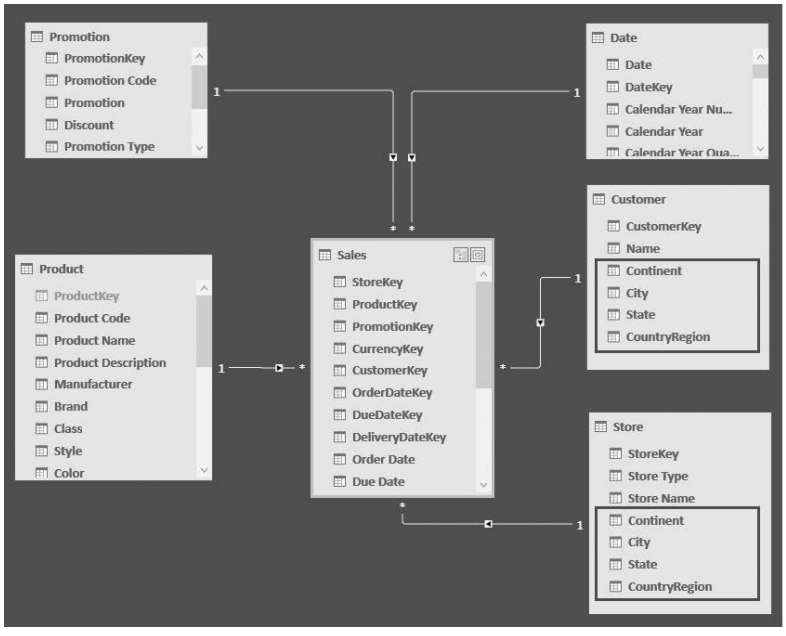
Taki model łamie zasadę mówiącą, że nie mogą istnieć relacje pomiędzy wymiarami. Trzy tabele, `Customer`, `Store` i `Geography`, pełnią funkcję wymiarów i zarazem są ze sobą powiązane. Dlaczego taki model jest zły? Ponieważ wprowadza *niejednoznaczność*.



RYSUNEK 1.13. *Nowy wymiar, Geography, jest powiązany relacjami z wymiarami Customer i Store*

Wyobraź sobie podział według miast z obliczaniem ilości sprzedanych produktów. System może się pokierować relacją między wymiarami `Geography` i `Customer` i zwrócić wielkość sprzedaży z podziałem na miasta, z których pochodzą klienci. Ale może też podążyć za relacją łączącą wymiary `Geography` i `Store`, a wtedy zwróci wielkość sprzedaży z podziałem na miasta, w których są zlokalizowane sklepy. Możliwa jest też trzecia opcja — system wybierze obie relacje i zwróci wielkości sprzedaży dokonanych w sklepach zlokalizowanych w poszczególnych miastach na rzecz klientów mieszkających w tych miastach. Model jest niejednoznaczny i nie ma prostego sposobu, by ustalić, co uzyskane liczby oznaczają. Problem ma charakter nie tylko techniczny, ale również logiczny. Rzeczywiście, użytkownik takiego modelu będzie zdezorientowany i nie będzie wiedział, jak ma interpretować wyniki obliczeń. Ze względu na tę niejednoznaczność zarówno Excel, jak i Power BI nie pozwalają na budowanie modelu z powiązaniami między wymiarami. O niejednoznacznościach będziemy jeszcze więcej mówić w następnych rozdziałach, a na razie dodamy tylko, że Excel (narzędzie, za pomocą którego zbudowaliśmy model przykładowy) wyłączył relację pomiędzy wymiarami `Store` i `Geography`, aby zlikwidować opisaną niejednoznaczność.

Jako modelarz danych musisz robić wszystko, aby unikać wszelkich niejednoznaczności. Jak można by ten problem rozwiązać w przytoczonym przez nas przykładzie? Odpowiedź jest bardzo prosta. Trzeba zdenormalizować odpowiednie kolumny tabeli `Geography` przez umieszczenie ich odpowiedników w tabelach `Store` i `Customer`. To pozwoli na usunięcie tabeli `Geography`. Poprawiony w ten sposób model jest pokazany na rysunku 1.14.

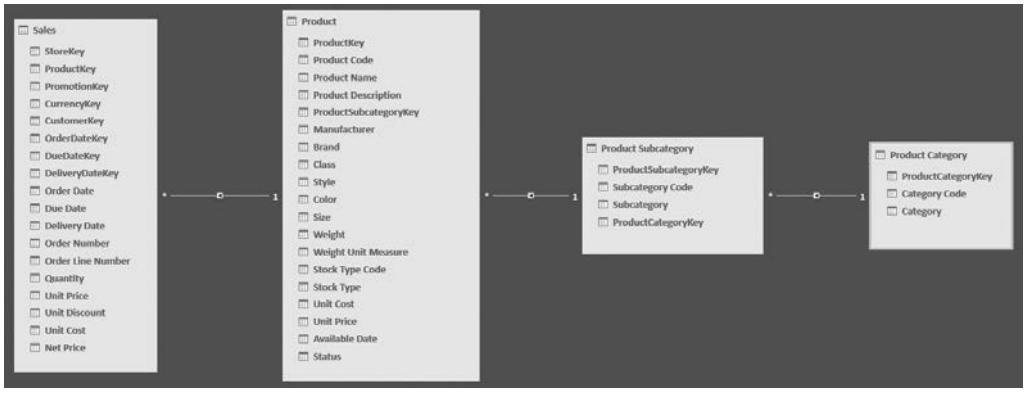


W ramach denormalizacji tabeli Geography jej kolumny zostały przeniesione do tabel Customer i Store, a sama tabela została usunięta z modelu

RYSUNEK 1.14. *Po denormalizacji kolumn tabeli Geography model powrócił do schematu gwiazdy*

Poprawna denormalizacja usuwa niejednoznaczność. Teraz każdy użytkownik będzie mógł filtrować dane według kolumn tabeli `Geography` z wykorzystaniem wyłącznie tabel `Customer` i `Store`. Dane geograficzne są tu wymiarem, ale żeby model mógł być zgodny z prawidłowym schematem gwiazdy, należało je zdenormalizować.

Zanim przejdziemy do innego tematu, poznaj inny, również często używany schemat modelu danych, a mianowicie *płatek śniegu*. Jest on odmianą schematu gwiazdy, w której wymiary nie są powiązane bezpośrednio z tabelą faktów. W połączeniach tych biorą udział wymiary pośrednie. Przykład takiego schematu jest pokazany na rysunku 1.15.



RYSUNEK 1.15. *Tabele Product Category, Product Subcategory i Product są powiązane łańcuchem relacji i tworzą schemat płatka śniegu*

Czy schemat płatka śniegu łamie zasadę niełączenia się wymiarów? W pewnym sensie tak, ponieważ relacja łącząca tabele `Product Subcategory` i `Product` jest przecież relacją między dwoma wymiarami. Różnica między tym przykładem a poprzednim polega na tym, że ta relacja jest jedyną pomiędzy `Product Subcategory` a innymi wymiarami połączonymi z tabelą faktów, czyli w tym przypadku z wymiarem `Product`. A zatem `Product Subcategory` można traktować jako wymiar grupujący rozmaite produkty, który jednak nie grupuje żadnych innych wymiarów lub faktów. To samo dotyczy oczywiście wymiaru `Product Category`. Tak więc płatek śniegu, choć łamie wspomnianą zasadę, nie wprowadza żadnej niejednoznaczności, a model danych zbudowany zgodnie z takim schematem działa bardzo dobrze.

Uwaga. Aby uniknąć stosowania schematu płatka śniegu, należy przeprowadzić denormalizację polegającą na przeniesieniu kolumn z najodleglejszych tabel do tej, która jest najbliżej tabeli faktów. Nie zawsze jednak jest to wskazane, ponieważ płatki śniegu okazują się niekiedy przydatnym sposobem przedstawiania danych i — pomijając niewielki spadek wydajności — nie ma w nich nic złego.

Jak się przekonasz w trakcie dalszej lektury, niemal zawsze najlepszym schematem modelu danych okazuje się gwiazda. Oczywiście są sytuacje, w których nie jest rozwiązaniem *doskonałym*. Jest to jednak schemat, na którym zawsze można polegać. Może nie być doskonale, ale na pewno nie będzie źle.

Uwaga. W miarę pogłębiania swojej wiedzy na temat modelowania danych z pewnością napotkasz sytuacje, w których pomyślisz, że może warto by porzucić schemat gwiazdy. Nie rób tego. Jest kilka powodów, dla których schemat gwiazdy jest niemal zawsze najlepszą opcją. Niestety, powody te dostrzega się i docenia dopiero po nabyciu pewnego doświadczenia w modelowaniu danych. Jeśli jeszcze nie masz dużego doświadczenia w tej materii, po prostu zaufaj dziesiątkom tysięcy profesjonalistów z całego świata, którzy twierdzą, że niemal zawsze gwiazda jest najlepsza — bez względu na wszystko.

Dlaczego nazywanie obiektów jest istotne?

Gdy budujemy model danych, zazwyczaj pobieramy dane z bazy SQL Server lub innego źródła. Najprawdopodobniej twórca tego źródła przyjął już jakąś konwencję nazewniczą. Takich konwencji jest mnóstwo — na tyle dużo, by można było śmiało powiedzieć, że każdy ma swoją własną konwencję nazewniczą.

Projektanci hurtowni danych zwykli poprzedzać nazwy wymiarów przedrostkiem `Dim` (od *dimension* — wymiar), a nazwy tabel faktów przedrostkiem `Fact`. Dlatego często można spotkać takie nazwy, jak `DimCustomer` (wymiar klient) czy `FactSales` (fakty sprzedaż). Inni lubią rozróżniać między widokami a tabelami fizycznymi i stosują przedrostki `Tbl` dla tabeli (*table*) oraz `Vw` dla widoku (*view*). Jeszcze inni uważają, że nazwy są niejednoznaczne, i wolą używać liczb, na przykład `Tbl_190_Sales`. Moglibyśmy długo wymieniać rozmaite konwencje, ale ograniczymy się do stwierdzenia, że istnieje dużo standardów, a każdy z nich ma plusy i minusy.

Uwaga. Można by dyskutować nad sensem tych standardów w przypadku baz danych, ale wykraczałoby to poza tematykę tej książki. Skupimy się raczej na dyskusji o sposobach wdrażania tych standardów w modelach danych opracowywanych za pomocą Power BI i Excela.

Nie ma potrzeby przestrzegania jakichkolwiek standardów technicznych, wystarczy zdrowy rozsądek i łatwość użycia. Frustrująca byłaby na przykład praca z modelem, w którym tabele miałyby jakieś bezsensowne nazwy, typu `VwDimCstmr` czy `Tbl_190_FactShpmt` — dziwne i zupełnie nieintuicyjne. Mimo to takie nazwy spotykamy dość często. A na razie mówimy tylko o nazwach tabel. Gdy chodzi o nazwy kolumn, brak właściwych pomysłów staje się jeszcze bardziej widoczny. Jedyne, co możemy doradzić, to pozbycie się tego typu nazw i wprowadzenie zamiast nich czegoś sensownego, co pozwoliłoby jednoznacznie zidentyfikować wymiar bądź tabelę faktów.

Budową systemów analitycznych zajmujemy się od lat. Z czasem udało nam się wypracować prosty, ale efektywny system nazewnictwa tabel i ich kolumn:

- **Nazwy wymiarów powinny się składać wyłącznie z nazwy zasobu biznesowego, w liczbie pojedynczej lub mnogiej.** A zatem dane o klientach umieszczamy w tabeli `Klient` lub `Klienci`. Informacje o produktach — w tabeli `Produkt` lub `Produkty`. (Naszym zdaniem liczba pojedyncza jest odpowiedniejsza, ponieważ lepiej pasuje do naturalnego języka zapytań w Power BI).
- **Jeśli nazwa zasobu biznesowego składa się z kilku wyrazów, każdy z tych wyrazów rozpoczynamy wielką literą.** Kategorie produktów umieszczamy więc w kolumnie `KategoriaProduktu`, a kraj dostawy — w kolumnie `KrajDostawy` lub `KrajDostarczenia`. Alternatywną opcją może być stosowanie przerw między wyrazami, na przykład `Kategoria Produktu`. Wygląda ładnie, ale nieco utrudnia pisanie kodu w języku DAX. W gruncie rzeczy jest to kwestia wyłącznie upodobań.
- **Nazwy tabel faktów powinny odzwierciedlać biznesowe nazwy faktów i zawsze mieć liczbę mnogą.** Fakty sprzedażowe trafiają więc do tabeli o nazwie `Sprzedaże`, a zakupowe, jak łatwo się domyślić, do tabeli `Zakupy`. Dzięki zastosowaniu liczby mnogiej w sposób naturalny myślimy, patrząc na nazwy tabel, o jednym kliencie (tabela `Klient`) i wielu transakcjach sprzedaży (tabela `Sprzedaże`), czyli o relacji jeden-do-wielu.
- **Należy unikać stosowania zbyt długich nazw.** Nazwy takie jak `KrajDostawyTowaru` ↪`SprzedanegoPrzezPośrednika` są dezorientujące. Nikt nie chce się wczytywać w takie tasiemce. Nazwy powinny być skrótowe i pozbawione zbędnych wyrazów.
- **Nazwy nie powinny być zbyt krótkie.** Wiemy, że teraz panuje moda na mówienie i pisanie skrótami, ale stosowanie ich w raportach nie jest wskazane, gdyż nie wszystkie skróty są dla wszystkich zrozumiałe. Wyobraź sobie, że na co dzień używasz w pracy skrótu *KDP* zamiast określenia *kraj dostawy dla pośrednika*, ale przecież nikt spoza grona Twoich współpracowników nie będzie wiedział, co ten skrót oznacza. Pamiętaj: raporty trafiają do wielu ludzi i nie wszyscy z nich rozumieją Twoje skróty.

- **Nazwa klucza wymiaru powinna być nazwą wymiaru z przyrostkiem *Key* (klucz).** Tak więc niech klucz podstawowy wymiaru `Klient` ma nazwę `KlientKey`. To samo dotyczy kluczy obcych. Jeśli jakiś klucz będzie miał inną nazwę niż tabela, w której się znajduje, od razu będzie wiadomo, że jest kluczem obcym. Klucz `KlientKey` w tabeli `Sprzedaże` będzie kluczem obcym wskazującym na tabelę `Klient`, ale w tabeli `Klient` będzie kluczem podstawowym.

Lista reguł jest bardzo krótka, Cała reszta zależy od Ciebie. Konkretne nazwy ustalasz samodzielnie, ale pamiętaj, żeby były sensowne i czytelne dla innych użytkowników. Współdzielenie modelu danych jest łatwiejsze, jeśli zastosowane w nim nazwy są dobrze przemyślane. Poza tym stosowanie powyższych reguł może się okazać pomocne w wyszukiwaniu ewentualnych błędów.

Wskazówka. Jeśli masz wątpliwości, czy jakaś nazwa będzie dobra, zadaj sobie pytanie: „Czy ktoś inny będzie wiedział, o co chodzi?”. Pamiętaj, że raczej rzadko będziesz jedynym czytelnikiem swoich raportów. Wcześniej czy później zdarzy się, że Twój raport trafi do rąk kogoś innego, kto będzie miał inną wiedzę niż Ty. Jeśli ta osoba będzie w stanie zrozumieć sens użytych przez Ciebie nazw, to łatwiej się porozumiecie. Jeśli nie będzie wiedziała, co się pod tymi nazwami kryje, powinieneś na nowo przemyśleć nazewnictwo swojego modelu.

Podsumowanie

W tym rozdziale omówiliśmy podstawy modelowania danych i teraz powinieneś już wiedzieć, że:

- Pojedyncza tabela też stanowi model danych, tyle że jest to forma najprostsza.
- W modelu jednotabelowym trzeba określić ziarnistość danych. Właściwy jej wybór znacznie ułatwia późniejsze przeprowadzanie analiz i obliczeń.
- Różnica między modelem jednotabelowym a wielotabelowym polega głównie na tym że w tym drugim tabele są połączone relacjami.
- Relacja może mieć strony *jeden* i *wiele* w zależności od tego, czy podążając za relacją, napotykamy jeden czy więcej wierszy. Ponieważ jednemu produktowi może odpowiadać wiele sprzedaży, tabela `Product` będzie po stronie *jeden* a tabela `Sales` — po stronie *wiele*.
- Jeśli tabela ma być celem relacji, musi zawierać klucz podstawowy, czyli kolumnę z wartościami unikatowymi, które mogą służyć do jednoznacznej identyfikacji wszystkich wierszy. Jeśli taki klucz nie istnieje, relacji nie da się ustanowić.
- W modelu znormalizowanym dane są przechowywane w sposób kompaktowy, czyli bez powtarzania tych samych wartości w różnych wierszach. Taki model zazwyczaj wymaga utworzenia większej liczby tabel.

- Model zdenormalizowany zawiera dużo powtórzeń (np. słowo `Czerwony` może się wielokrotnie powtórzyć w tabeli produktów), ale za to ma mniej tabel.
- Modele znormalizowane są stosowane w systemach OLTP, a zdenormalizowane — w systemach analitycznych.
- W typowym modelu analitycznym wyróżnia się zasoby informacyjne (wymiary) i zdarzenia (fakty). Jeśli każda encja danego modelu zostanie zaklasyfikowana jako fakt lub do wymiar, to o takim modelu mówi się, że jest zbudowany zgodnie ze schematem gwiazdy. Schemat gwiazdy jest najczęściej stosowany w modelach analitycznych i niemal zawsze sprawdza się bardzo dobrze.

Profesjonalne modelowanie danych — pewny sukces w biznesie!

Możliwości Excela są imponujące, a odkąd Microsoft udostępnił narzędzia w rodzaju Analysis Services, Power BI czy Power Pivot, arkusz ten stał się znakomitym narzędziem do analizy, modelowania oraz innych form przetwarzania dużych i złożonych zbiorów danych. Są to umiejętności, które przydadzą się w wielu dziedzinach życia, nie tylko w biznesie. Ich opanowanie nie jest zbyt trudne, a może stać się źródłem wielkiej radości i prawdziwej satysfakcji zwłaszcza dla kogoś, kto lubi pracować z liczbami. Oczywiście osoby, które osiągną wysoki poziom umiejętności w tym zakresie, będą mogły liczyć na bardzo konkretne profity!

Ta książka jest świetnym wprowadzeniem do modelowania danych w Excelu za pomocą narzędzi Power BI i Power Pivot. Dowiesz się z niej, jak optymalnie analizować zgromadzone dane i skutecznie wydobyć z nich potrzebne informacje. Zapoznasz się z ważnymi pojęciami i przyswoisz podstawowe techniki kształtowania modeli danych w Excelu i Power BI. Dzięki licznym praktycznym i przydatnym przykładom uzyskasz nową perspektywę — spojrzysz na zgromadzone dane okiem wytrawnego modelarza. Co więcej, szybko się przekonasz, że należyte zbudowanie modelu wcale nie jest trudne, a w efekcie przynosi prawidłowe odpowiedzi na wiele ważnych pytań!

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 978-83-289-0331-9

Cena: 69,00 zł

W tej książce między innymi:

- zasady i popularne techniki modelowania danych
- tabele faktów w złożonym modelu danych
- metody śledzenia atrybutów historycznych
- migawki i ich zastosowania
- analiza zdarzeń o określonym czasie trwania
- dobieranie rodzaju modelu do konkretnych pytań biznesowych

Alberto Ferrari i Marco Russo

od dwóch dekad zajmują się procesami business intelligence i Analysis Services. Obaj posiadają tytuły Microsoft MVP i SSAS Maestro. Często występują na prestiżowych konferencjach. Ferrari jest autorytetem w dziedzinie modelowania danych oraz usług analitycznych w dużych i złożonych hurtowniach danych, Russo specjalizuje się w analizie danych dla potrzeb wywiadu gospodarczego.

Microsoft Press