

Modelowanie danych z **Power BI** dla ekspertów analityki

Jak w pełni wykorzystać możliwości **Power BI**

Soheil Bakhshi
Przedmowa Christian Wade

Helion 



Tytuł oryginału: Expert Data Modeling with Power BI: Get the best out of Power BI by building optimized data models for reporting and business needs

Tłumaczenie: Anna Mizerska

ISBN: 978-83-283-9451-3

Copyright © Packt Publishing 2021. First published in the English language under the title 'Expert Data Modeling with Power BI' – (9781800205697).

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/modapo>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<https://ftp.helion.pl/przyklady/modapo.zip>

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

Wstęp	13
Część I. Modelowanie danych w Power BI	19
Rozdział 1. Wprowadzenie do modelowania danych i Power BI	21
Warstwy Power BI	22
Warstwa przygotowania danych (Power Query)	23
Warstwa modelu danych	23
Warstwa wizualizacji danych	25
Przepływ danych w Power BI	27
Co oznacza modelowanie danych w Power BI?	28
Model semantyczny	28
Budowanie wydajnych modeli danych w Power BI	29
Schemat gwiazdy (modelowanie wymiarów) i schemat płatka śniegu	31
Licencje Power BI	40
Maksymalny rozmiar pojedynczego zestawu danych	40
Przyrostowe ładowanie danych	41
Grupy obliczeniowe	41
Wspólne zestawy danych	42
Przepływ danych w Power BI	42
Podejście przyrostowego modelowania danych	43
Zbieranie informacji od firmy	44
Przygotowanie danych w oparciu o logikę biznesową	44
Modelowanie danych	44
Testowanie logiki	44
Przedstawienie logiki biznesowej za pomocą prostej wizualizacji danych	45
Sposób myślenia profesjonalnego modelarza danych	45
Podsumowanie	46

Rozdział 2. Język DAX i modelowanie danych	47
Wirtualne tabele	47
Tworzenie tabeli obliczeniowej	48
Użycie tabel wirtualnych w miarach — część 1.	49
Użycie tabel wirtualnych w miarach — część 2.	51
Wyświetlanie wyników wirtualnych tabel	53
Relacje w wirtualnych tabelach	55
Analiza czasowa i modelowanie danych	65
Wykrywanie odpowiednich dat w wymiarze daty	65
Obliczenia okres po okresie	72
Generowanie wymiaru daty za pomocą DAX	80
Tworzenie wymiaru czasu za pomocą DAX	86
Podsumowanie	89
 Część II. Przygotowanie danych w edytorze zapytań	 91
Rozdział 3. Przygotowanie danych w edytorze Power Query	93
Wprowadzenie do języka Power Query M w Power BI	94
W Power Query wleŁkOŚĆ lItEr Ma ZnAcZeNiE	94
Zapytania	95
Wyrażenia	95
Wartości	95
Typy	100
Wprowadzenie do edytora Power Query	102
Panel Zapytania	103
Panel Ustawienia zapytania	105
Panel widoku danych	107
Pasek stanu	109
Edytor zaawansowany	111
Wprowadzenie do funkcji Power Query przydatnych modelarzom danych	112
Jakość kolumn	112
Rozkład kolumn	115
Profil kolumny	119
Parametry zapytania	120
Niestandardowe funkcje	126
Funkcje rekurencyjne	130
Podsumowanie	132
 Rozdział 4. Pobieranie danych z różnych źródeł	 134
Pobieranie danych z popularnych rodzajów źródeł danych	135
Folder	135
Pliki CSV/TXT/TSV	141
Excel	146
Zestawy danych Power BI	152
Przepływy danych Power BI	155
SQL Server	158

Usługi SQL Server Analysis Services i Azure Analysis Services	159
Wielowymiarowy/tabelaryczny SSAS	160
AAS	161
Źródło danych OData	162
Certyfikaty źródeł danych	165
Brązowa	165
Srebrna	166
Złota/Platynowa	166
Tryby połączeń	167
Importowanie danych	167
DirectQuery	168
Połączenie na żywo	169
Tryby przechowywania danych	170
Przechowywanie zestawów danych	171
Podsumowanie	173
Rozdział 5. Najczęściej stosowane kroki przygotowywania danych	174
Zmiana typu danych	176
Podział kolumn w oparciu o separator	182
Scalanie kolumn	185
Dodanie niestandardowej kolumny	186
Dodanie kolumny z przykładów	189
Powielanie kolumny	191
Filtrowanie wierszy	193
Grupowanie według	196
Dołączanie zapytań	198
Scalanie zapytań	202
Duplikowanie zapytania i odwoływanie się do niego	204
Zamiana wartości	206
Wydobywanie liczb z tekstu	208
Praca z datą, datą/godziną oraz datą/godziną/strefą czasową	210
Podsumowanie	214
Rozdział 6. Przygotowanie schematu gwiazdy w edytorze Power Query	215
Określanie wymiarów i faktów	216
Liczba tabel w źródle danych	217
Połączenia między tabelami	217
Znalezienie najmniejszego poziomu rozdrobnienia daty i godziny	218
Definiowanie wymiarów i faktów	220
Tworzenie tabel wymiarów	224
Geografia	224
Zamówienie	227
Produkt	229
Waluta	233
Klient	234
Sprzedaż w ujęciu demograficznym	235
Data	238

Godzina	241
Tworzenie wymiarów daty i godziny — Power Query kontra DAX	244
Tworzenie tabel faktów	244
Podsumowanie	251

Rozdział 7. Sprawdzone metody przygotowania danych **252**

Ważne kwestie przygotowania danych	252
Stopniowe ładowanie danych ze źródła danych OData	253
Uwzględnianie wielkości liter przez Power Query chroni przed problemami podczas modelowania danych	256
Składanie zapytań wpływa na odświeżanie danych	256
Porządkowanie zapytań w edytorze zapytań	262
Konwersja typu danych	264
Zmiana typu danych może wpłynąć na modelowanie	265
Dodaj krok zmiany typu, gdy to tylko możliwe	271
Staraj się mieć tylko jeden krok zmiany typu danych	273
Optymalizacja rozmiaru zapytania	274
Usuwanie zbędnych kolumn i wierszy	274
Grupowanie według	275
Wyłączanie ładowania danych	275
Konwencja nazewnictwa	275
Podsumowanie	276

Część III. Modelowanie danych **277**

Rozdział 8. Komponenty modelowania danych **279**

Modelowanie danych w Power BI Desktop	280
Tabele	280
Właściwości tabeli	280
Tabele polecane	284
Tabele obliczeniowe	285
Pola	288
Typy danych	289
Formatowanie niestandardowe	289
Kolumny	290
Hierarchie	302
Miary	303
Relacje	307
Klucze podstawowe i obce	309
Klucze złożone	310
Kardynalność relacji	314
Filtr propagacji	316
Relacje dwukierunkowe	318
Podsumowanie	321

Rozdział 9. Sprawdzone metody związane ze schematem gwiazdy i z modelowaniem danych	322
Praca z relacjami wiele do wielu	323
Relacje wiele do wielu z użyciem tabeli pomostowej	325
Ukrywanie tabeli pomostowej	332
Ostrożność w pracy z relacjami dwukierunkowymi	332
Praca z nieaktywnymi relacjami	335
Dostęp przez wiele ścieżek filtra	335
Wiele bezpośrednich relacji między dwiema tabelami	337
Użycie tabel konfiguracyjnych	338
Segmentacja	339
Formatowanie warunkowe z użyciem miar	341
Unikanie kolumn obliczeniowych	347
Porządek w modelu danych	349
Ukrywanie nieistotnych obiektów	350
Tworzenie tabel z miarami	352
Stosowanie folderów	356
Zmniejszanie rozmiaru modelu przez wyłączenie automatycznej daty i godziny	359
Podsumowanie	361

Część IV. Zaawansowane modelowanie danych **363**

Rozdział 10. Zaawansowane techniki modelowania danych	365
Stosowanie agregacji	366
Implementacja agregacji dla źródeł danych nieobsługujących trybu DirectQuery	367
Implementacja agregacji na poziomie daty	367
Implementacja agregacji na poziomie roku i miesiąca	372
Używanie funkcji Zarządzanie agregacjami	377
Sprawdzanie agregacji	383
Odświeżanie przyrostowe	389
Konfiguracja odświeżania przyrostowego w Power BI Desktop	390
Testowanie odświeżania przyrostowego	395
Hierarchie nadrzędny-podrzędny	397
Określanie głębokości hierarchii	399
Tworzenie poziomów hierarchii	401
Implementacja wymiarów odgrywających rolę	404
Stosowanie grup obliczeniowych	407
Wymagania	407
Terminologia	408
Implementacja grup obliczeniowych na potrzeby analityki czasowej	409
Testowanie grup obliczeniowych	413
Naprawa błędów z formatem	415
Funkcje DAX dla grup obliczeniowych	416
Podsumowanie	417

Rozdział 11. Zabezpieczenia na poziomie wiersza	418
Co oznaczają zabezpieczenia na poziomie wiersza w modelowaniu danych?	419
Czym RLS nie jest	419
Terminologia związana z RLS	419
Przypisywanie użytkowników do ról w usłudze Power BI	423
Przypisywanie użytkowników do ról na serwerze raportów Power BI	423
Proces implementacji zabezpieczeń na poziomie wiersza	425
Najpopularniejsze podejścia implementacji RLS	426
Implementacja statycznych zabezpieczeń na poziomie wiersza	426
Implementacja dynamicznych zabezpieczeń na poziomie wiersza	433
Podsumowanie	448
Rozdział 12. Dodatkowe możliwości i dostępne funkcje modelowania danych	449
Praca z wolno zmieniającymi się wymiarami	449
SCD typu 0 (SCD 0)	451
SCD typu 1 (SCD 1)	451
SCD typu 2 (SCD 2)	451
Wprowadzenie do zabezpieczeń na poziomie obiektu	454
Wdrażanie OLS	455
Testowanie ról	458
Przypisywanie członków do ról w usłudze Power BI	460
Testowanie ról w usłudze Power BI	461
Wprowadzenie do przepływów danych	462
Zastosowania przepływów danych	463
Terminologia związana z przepływami danych	464
Tworzenie przepływów danych	465
Wprowadzenie do modeli złożonych	477
Nowe pojęcia	478
Podsumowanie	485

Przygotowanie danych w edytorze Power Query

W poprzednich rozdziałach wymieniliśmy różne warstwy Power BI i przeszliśmy przez kilka scenariuszy. Teraz powinno być dla Ciebie oczywiste, że Power BI nie jest tylko do przygotowania raportów. Jest to tak naprawdę zaawansowane narzędzie do analityki biznesowej (BI, ang. *business intelligence*), które obsługuje wiele aspektów BI, między innymi **wyciąganie danych, przekształcanie i wgrywanie** (procesy ETL, ang. *extract, transform, load*), modelowanie danych, analizę danych i wizualizację danych. Co więcej, to potężne narzędzie, jakim jest Power BI, cały czas jest rozwijane, co jest wspaniałe. Odpowiedzialny za to zespół w firmie Microsoft nieustannie wprowadza nowe pomysły do tej technologii, by uczynić ją jeszcze potężniejszą. Jednym z obszarów analityki biznesowej, w którym Power BI sprawdza się świetnie, są zadania związane z procesem ETL w warstwie przygotowania danych, dzięki tak zwanemu **edytorowi Power Query**. To narzędzie służy w Power BI do pisania wyrażeń w języku Power Query i jest dostępne również w Excelu i wielu innych produktach firmy Microsoft. W tym rozdziale przyjrzymy się bliżej językowi **Power Query M**. Zajmiemy się w nim następującymi tematami:

- Wprowadzenie do języka Power Query M w Power BI.
- Wprowadzenie do edytora Power Query.
- Wprowadzenie do funkcji Power Query przydatnych modelarzom danych.
- Parametry zapytania.
- Niestandardowe funkcje.

Aby zobaczyć te koncepcje w działaniu, posłużymy się praktycznymi przykładami.

Wprowadzenie do języka Power Query M w Power BI

Power Query jest technologią przygotowania danych oferowaną przez firmę Microsoft, dającą możliwość łączenia się z wieloma źródłami danych w różnych technologiach, tak by firma mogła integrować, przekształcać, udostępniać dane do analizy i wyciągać z nich cenne wnioski. Power Query nie tylko może łączyć się z różnymi źródłami danych, ale również zapewnia zestaw narzędzi do tworzenia niestandardowych łączników (ang. *connector*), który może być używany przez organizacje spoza Microsoftu do tworzenia własnych łączników. Power Query początkowo był dodatkiem do programu Excel, ale szybko stał się istotną częścią platformy danych Microsoftu, służącą do przygotowania i przekształcania danych.

Obecnie Power Query jest zintegrowane z wieloma produktami firmy Microsoft, takimi jak **Dataverse** (poprzednio Common Data Service — CDS), **SSAS Tabular** (SQL Server Analysis Services Tabular Models) i **AAS** (Azure Analysis Services), jak również Power BI i Excel. Dlatego znajomość Power Query może pomóc osobom zajmującym się zawodowo danymi w ich przygotowaniu we wszystkich wymienionych właśnie technologiach. My skupimy się jednak na Power Query w Power BI.

Power Query M to język formuł łączący dane z wielu różnych źródeł, które można do siebie dopasować w celu stworzenia pojedynczego zestawu danych. W tym podrozdziale zostanie wprowadzony Power Query M.

W Power Query wielkość liter ma znaczenie

Podczas gdy Power Query jest językiem, w którym wielkość liter ma znaczenie, w języku DAX to nie ma znaczenia, co może być mylące dla wielu programistów. Pamiętaj, że Power Query i DAX to dwa różne światy, które łączą się w Power BI i zajmują się różnymi aspektami pracy z danymi. W Power Query wielkość liter ma nie tylko znaczenie, jeśli chodzi o składnię, ale również podczas interakcji z danymi. Na przykład jeżeli wpisujemy następującą funkcję, otrzymamy komunikat błędu:

```
datetime.localnow()
```

A to dlatego, że poprawną składnią jest:

```
DateTime.LocalNow()
```

Nieprzestrzeganie wielkości liter w interakcji z danymi może powodować błędy, które będą trudne do znalezienia. Przykład z życia wzięty — uzyskujemy wartości **identyfikatora globalnie unikatowego (GUID, ang. globally unique identifier)** ze źródła danych, w którym występują małe litery. Z innego źródła danych, z wielkimi literami, możesz również otrzymywać inne wartości GUID. Gdy będziemy dopasowywać te wartości w Power Query przez scalewanie dwóch zapytań, może się okazać, że żadne dane nie pasują. W rzeczywistości zamieniamy wartości GUID zawierające małe litery na wielkie i w ten sposób otrzymujemy dopasowane

wartości. Mimo że nie widzimy żadnych komunikatów o błędzie, gdy porównujemy dwa identyfikatory GUID, uzyskany rezultat jest niepoprawny, jeśli nie mamy dwóch GUID z pasującymi wielkimi i małymi literami.

Na przykład następujące wartości GUID nie są równe dla Power Query, a dla języka DAX już tak:

```
D5E99E0E-0737-45B2-B62A-4170B3FEFC0E
d5e99e0e-0737-45b2-b62a-4170b3fec0e
```

Zapytania

W Power Query zapytanie zawiera **wyrażenia**, **zmienne** i **wartości** wewnątrz instrukcji `let` i `in`. Blok instrukcji `let` i `in` jest zbudowany w taki sposób:

```
let
    NazwaZmiennej = wyrażenie1,
    #"Nazwa zmiennej" = wyrażenie2
in
    #"Nazwa zmiennej"
```

Jak widać w pokazanej właśnie strukturze, w nazwie zmiennej możemy mieć spacje. Jednak musimy zamknąć nazwę zmiennej za pomocą znaku `#` i cudzysłowu, na przykład `#"Nazwa zmiennej"`. Definiując zmienną w zapytaniu, tworzymy w Power Query **krok zapytania**. Kroki zapytania mogą odnosić się do każdego poprzedniego kroku. Wynik zapytania jest dowolną zmienną, która występuje zaraz po instrukcji `in`. Każdy krok, oprócz ostatniego kroku przed instrukcją `in`, musi się kończyć przecinkiem.

Wyrażenia

W Power Query wyrażenie to formuła, której wynikiem są wartości. Na rysunku 3.1 pokazano kilka wyrażeń i ich wyniki.

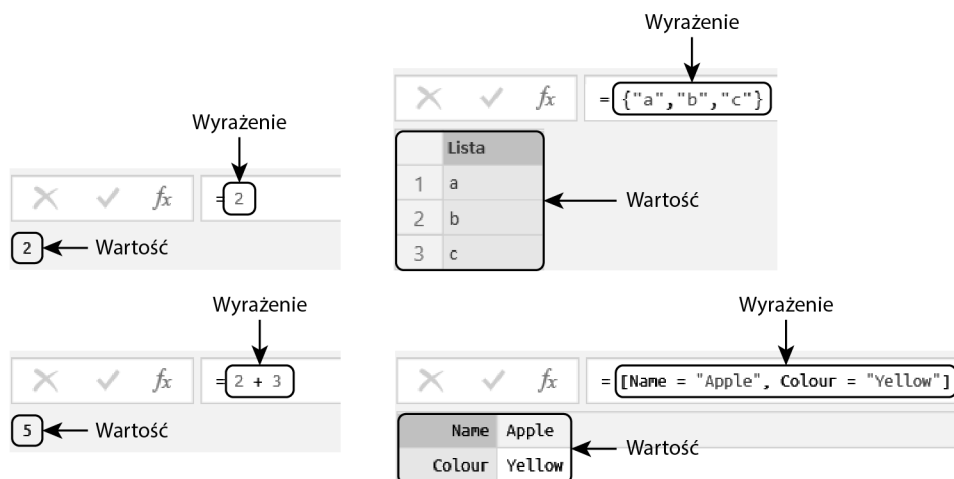
Wartości

Jak już zostało wspomniane wcześniej, wartości są wynikami wyrażeń. Na przykład w lewej górnej części rysunku 3.1 wyrażenie to 2, a jego wynikiem jest stała wartość 2.

W Power Query mamy dwie kategorie wartości: wartości pierwotne i wartości strukturalne.

Wartości pierwotne

Wartość pierwotna to stała wartość, taka jak liczba, tekst czy pusta wartość. Na przykład 123 to prymitywna wartość liczbowa, a "123" (wraz z cudzysłowem) to prymitywna wartość tekstowa.



Rysunek 3.1. Wyrażenia i ich wyniki

Wartości strukturalne

Wartości strukturalne zawierają wartości pierwotne albo inne wartości strukturalne. Istnieją cztery typy wartości strukturalnych: **lista**, **rekord**, **tabela** i **wartości zwrócone przez funkcję**.

- **Lista:** Lista jest sekwencją wartości pokazaną tylko w jednej kolumnie. Możemy ją zdefiniować przy użyciu nawiasów klamrowych `{}`. Na przykład możemy utworzyć listę małych liter podstawowego alfabetu łacińskiego, `{"a".."z"}`, tak jak pokazano na rysunku 3.2.
- **Rekord:** Rekord to zestaw pól składający się na wiersz danych. Aby utworzyć rekord, podajemy nazwę pola, znak równości i wartość pola w nawiasach kwadratowych `[]`. Różne pola i ich wartości oddzielamy przecinkiem w następujący sposób:

```
]
  First Name = "Soheil"
  , Last Name = "Bakhshi"
  , Occupation = "Consultant"
]
```

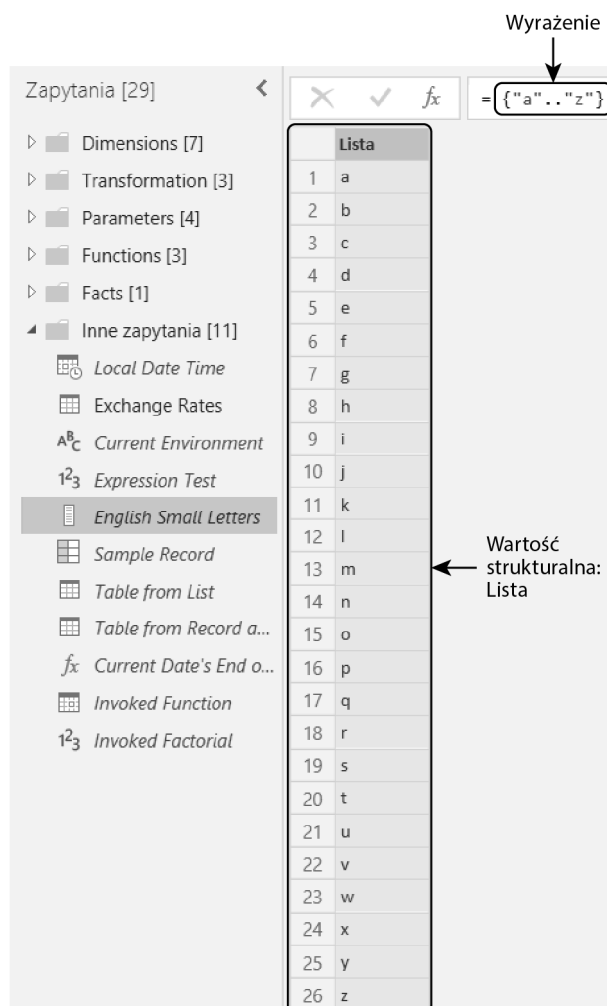
Rysunek 3.3 przedstawia wyrażenie i jego wartości.

Uwaga

Definiując rekord, nie musimy podawać nazw pól w cudzysłowie.

Tak jak widać na rysunku 3.3, w edytorze Power Query rekordy są pokazane pionowo.

Jak już wspomniano wcześniej, wartość strukturalna może zawierać inne wartości strukturalne. Wyrażenie pokazane poniżej tworzy rekord zawierający listę z wartościami pierwotnymi:

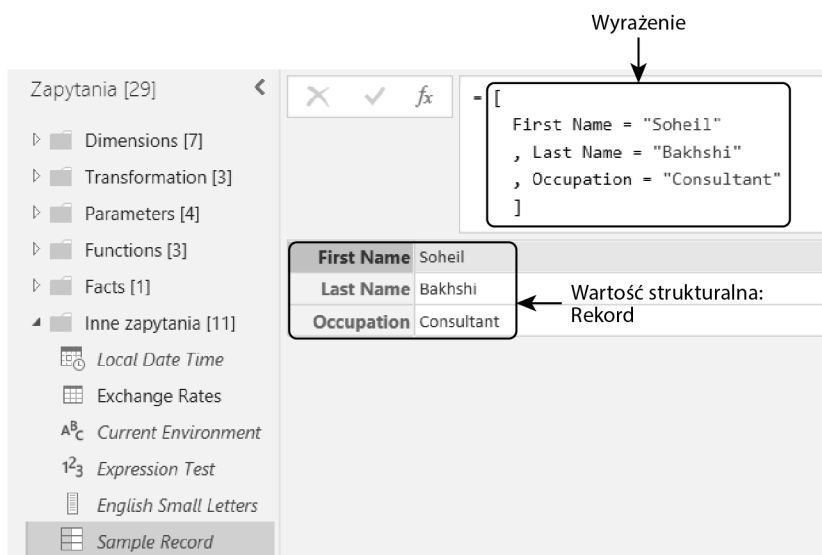


Rysunek 3.2. Definicja listy małych liter podstawowego alfabetu łacińskiego

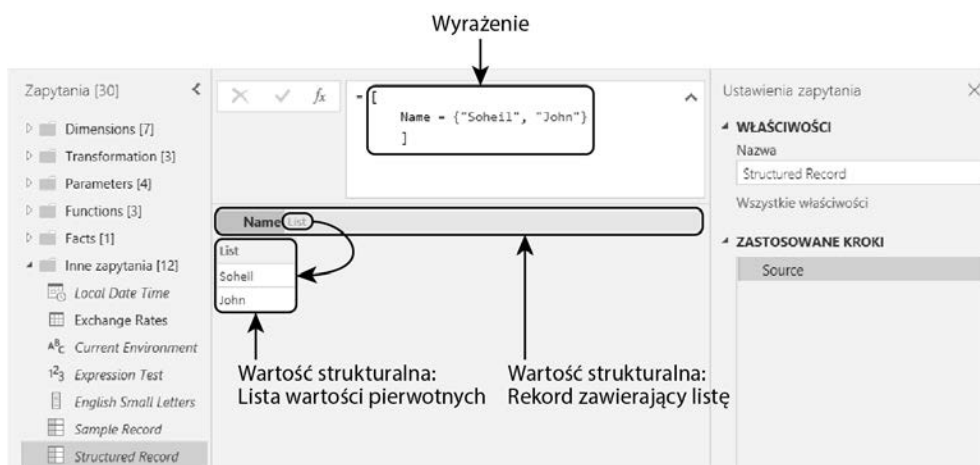
```
[
  Name = {"Soheil", "John"}
]
```

Rysunek 3.4 przedstawia rezultat (rekord zawierający listę wartości).

- **Tabela:** Tabela to zestaw wartości ułożony w kolumny i wiersze. Każda kolumna musi mieć swoją nazwę. Istnieje parę sposobów tworzenia tabeli przy użyciu kilku standardowych funkcji Power Query. Niemniej jednak możemy utworzyć tabelę z listy rekordów. Na rysunku 3.5 pokazano dwa sposoby na utworzenie tabeli przy użyciu słowa kluczowego #table. Sposoby te przedstawiono w dwóch kolejnych listingach.



Rysunek 3.3. Definicja rekordu w Power Query



Rysunek 3.4. Wartość strukturalna zawierająca inną wartość strukturalną

- Oto pierwszy sposób na utworzenie tabeli:

```

#table( {"ID", "Fruit Name"} , {{1,"Apple"}, {2, "Orange"}, {3, "Banana"}})

```

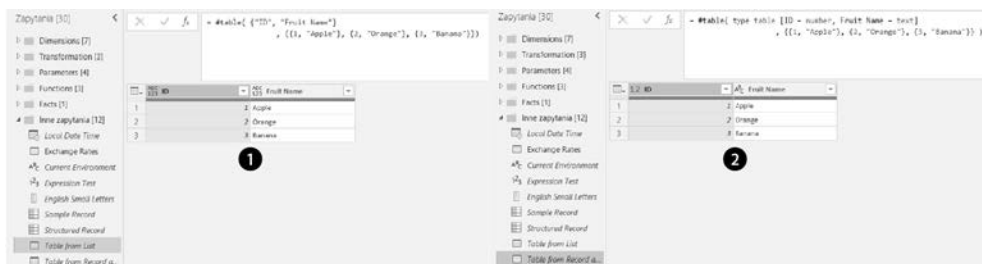
- A oto drugi sposób na uzyskanie tego samego rezultatu:

```

#table( type table [ID = number, Fruit Name = text]
  , {{1, "Apple"}, {2, "Orange"}, {3, "Banana"}} )

```

Jak widać na rysunku 3.5, w drugiej metodzie zdefiniowaliśmy typ danych kolumny, podczas gdy pierwsze wyrażenie zakłada dowolny typ danych.

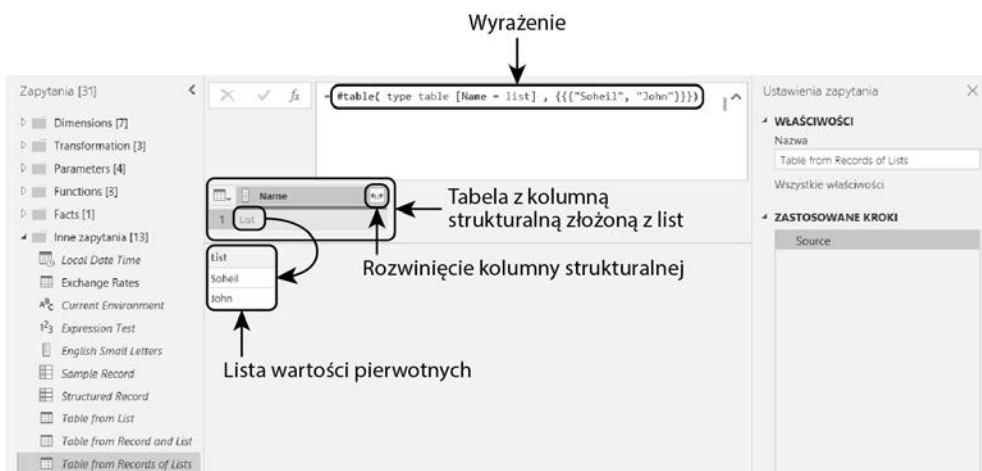


Rysunek 3.5. Tworzenie tabeli za pomocą Power Query

Następne wyrażenie tworzy tabelę złożoną z dwóch list. Każda z tych list zawiera wartości pierwotne.

```
#table( type table [Name = list] , {{{"Soheil", "John"}}})
```

Możemy rozwinąć kolumnę strukturalną, by zobaczyć wartości pierwotne, tak jak pokazano na rysunku 3.6.



Rysunek 3.6. Tabela z kolumną strukturalną

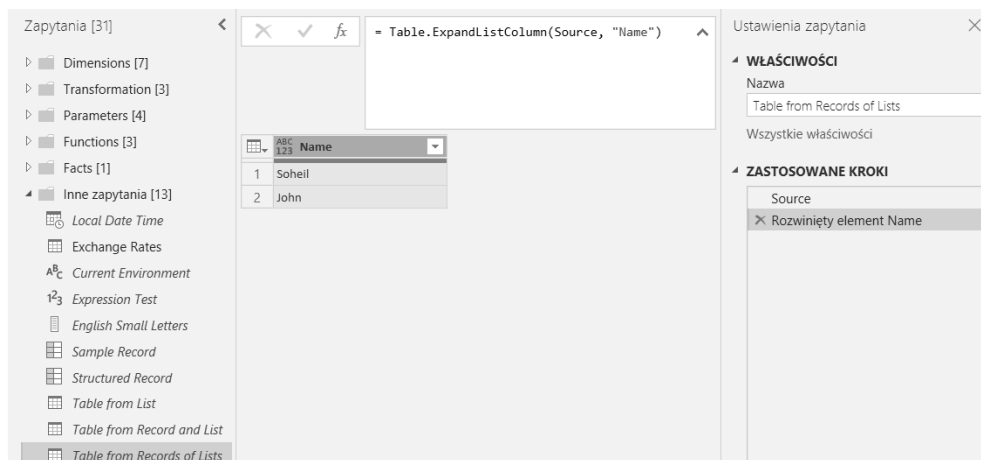
Rysunek 3.7 pokazuje tabelę po rozwinięciu kolumny strukturalnej.

- **Funkcja:** Funkcja przyjmuje parametry wejściowe i zwraca wartość wyjściową. Aby wywołać funkcję, podajemy w nawiasie listę parametrów (jeśli jakieś są), a następnie typ zwracanej wartości. Używamy symbolu =>, po którym umieszczamy definicję funkcji.

Na przykład ta funkcja oblicza datę ostatniego dnia bieżącego miesiąca:

```
() as date => Date.EndOfMonth(Date.From(DateTime.LocalNow()))
```

Pokazana właśnie funkcja nie przyjmuje żadnych parametrów wejściowych, ale zwraca wartość.



Rysunek 3.7. Tabela z rozwiniętą kolumną strukturalną

Rysunek 3.8 przedstawia wywołanie funkcji bez parametrów, która zwraca datę ostatniego dnia bieżącego miesiąca.

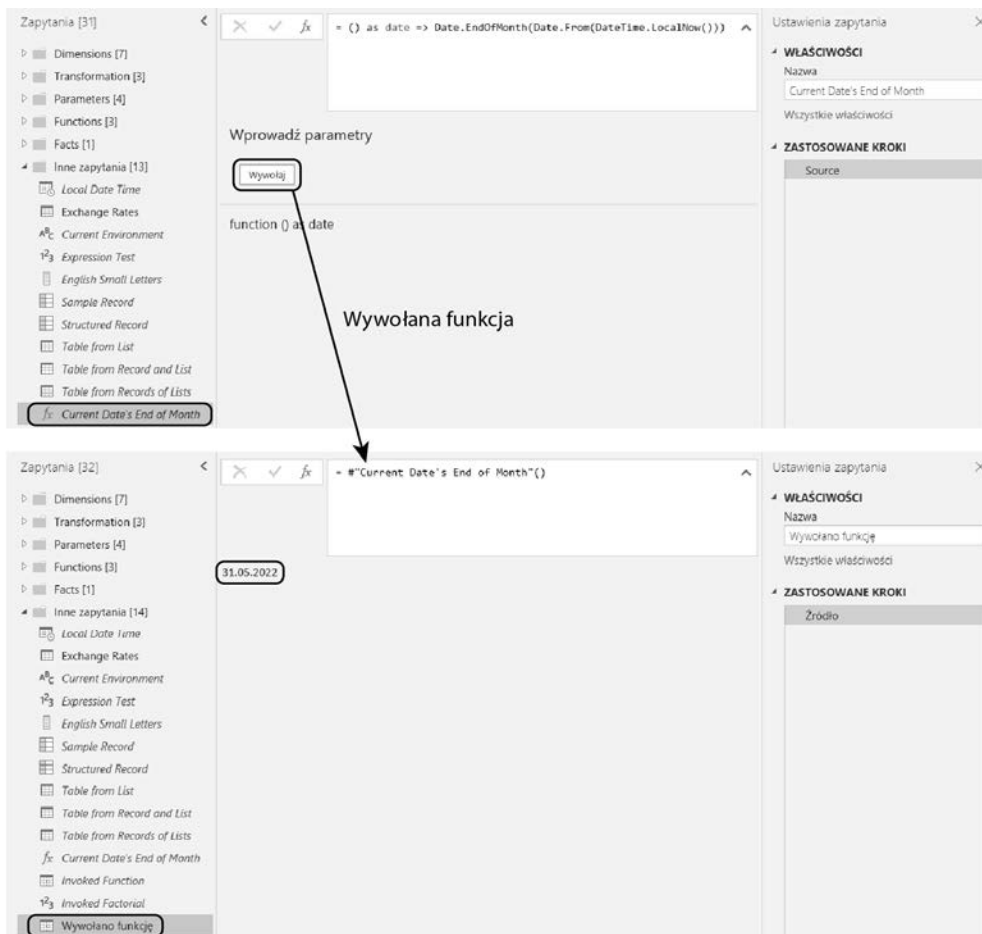
Typy

W Power Query wartości mają swój typ. Są dwie główne kategorie typów: **pierwotne** i **nie-standardowe**.

Typy pierwotne

Wartość może mieć jeden z następujących typów:

- binary (binarny),
- date (data),
- datetime (data i godzina),
- duration (czas trwania),
- list (lista),
- logical (logiczny),
- null (pusty),
- number (liczba),
- record (rekord),
- text (tekst),
- time (czas),
- type (typ),



Rysunek 3.8. Wywołanie niestandardowej funkcji

- function (funkcja),
- table (tabela),
- any (dowolny),
- none (żaden).

Z podanej powyżej listy ciekawym typem jest any. Wszystkie pozostałe typy są z nim zgodne. Jednak nie możemy powiedzieć, że wartość ma typ any (dowolny).

Typy niestandardowe

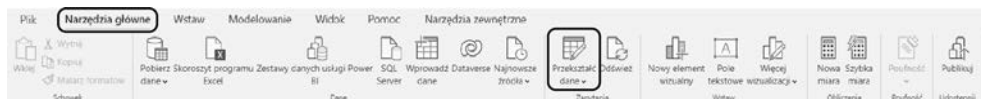
Typy niestandardowe to takie, które sami tworzymy. Na przykład poniższe wyrażenie definiuje niestandardową listę liczb:

```
type { number }
```

Wprowadzenie do edytora Power Query

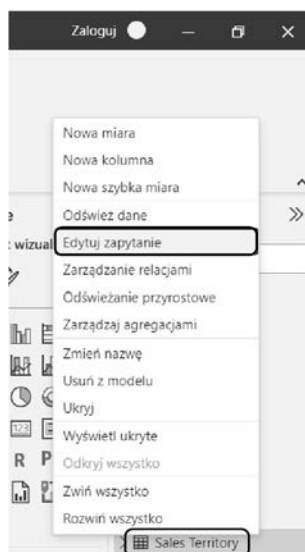
W programie Power BI Desktop Power Query dostępny jest w edytorze Power Query. Jest kilka sposobów, by go otworzyć:

- Naciśnij przycisk *Przekształć dane* w zakładce *Narzędzia główne*, który pokazano na rysunku 3.9.



Rysunek 3.9. Otwarcie edytora Power Query za pomocą przycisku na pasku

- Możemy przejść bezpośrednio do danego zapytania w edytorze Power Query przez kliknięcie prawym przyciskiem myszy wybranej tabeli w panelu *Pola*, a następnie wybranie *Edytuj zapytanie* (patrz rysunek 3.10).

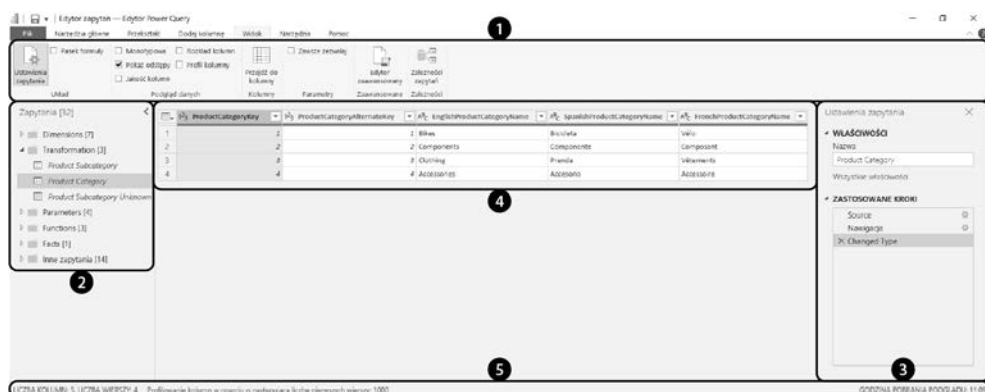


Rysunek 3.10. Przejście bezpośrednio do zapytania wybranej tabeli w edytorze Power Query

Edytor Power Query ma następujące części:

1. Pasek zadań.
2. Panel *Zapytania*.
3. Panel *Ustawienia zapytania*.
4. Panel *Dane*.
5. Pasek statusu.

Rysunek 3.11. przedstawia wymienione właśnie części w edytorze Power Query.



Rysunek 3.11. Części edytora Power Query

W następnych kilku punktach omówimy niektóre funkcjonalności i narzędzia edytora Power Query w odniesieniu do modelowania danych.

Panel Zapytania

Ta część pokazuje wszystkie aktywne i nieaktywne zapytania, w tym tabele, niestandardowe funkcje, parametry zapytania, stałe wartości i grupy. W kilku kolejnych podpunktach omówimy te wszystkie elementy.

Tabele

Znajdziemy tutaj tabele wgrane ze źródła danych, tabele utworzone w Power BI przez opcję *Wprowadź dane* oraz tabele, które odnoszą się do innych zapytań. Tabele oznaczone są ikoną .

Niestandardowe funkcje

To są funkcje, które stworzyliśmy sami w edytorze Power Query. Możemy wywołać takie funkcje i wielokrotnie ich używać w innych zapytaniach. Ikona niestandardowych funkcji wygląda tak: .

Parametry zapytania

Możemy określić parametrami różne części naszych zapytań, które inaczej musiałyby być ustawione na sztywno. Parametry zapytania znajdują się w panelu *Zapytania* i są oznaczone ikoną .

Wartości stałe

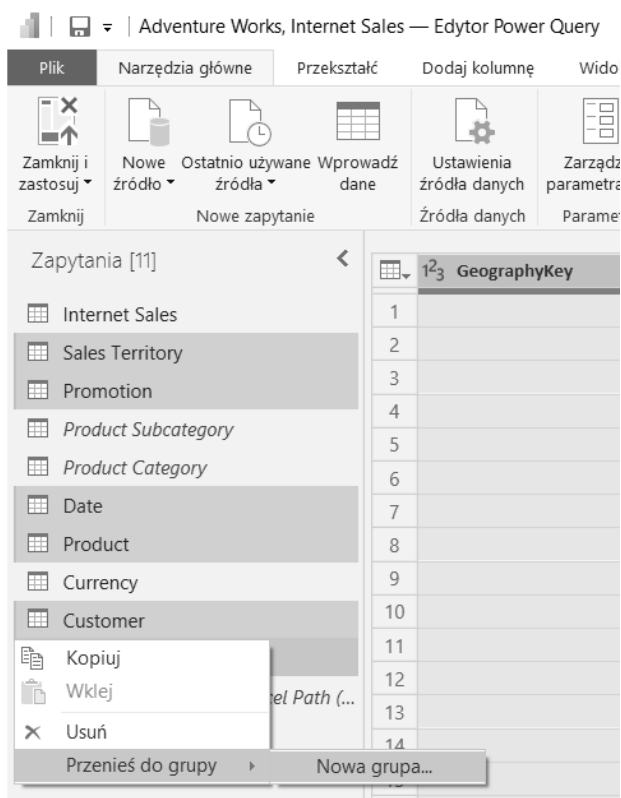
W niektórych przypadkach mamy zapytanie ze stałym wynikiem, który może być łańcuchem znaków, datą i godziną, datą, strefą czasową i tak dalej. Zapytania ze stałą wartością możemy bardzo łatwo rozpoznać po ich ikonie, która zależy od typu zwracanej wartości. Na przykład jeśli zapytanie zwraca dane typu `DateTime`, to ikona będzie wyglądać tak: , a jeśli to będzie łańcuch znaków, to ikona będzie miała taką postać: .

Grupy

W panelu *Zapytania* możemy grupować nasze zapytania w następujący sposób:

1. W panelu *Zapytania* zaznacz tabelę, którą chcesz zgrupować, przez przytrzymanie klawisza **Ctrl** i kliknięcie lewym przyciskiem myszy każdej z wybranych tabel.
2. Kliknij prawym przyciskiem myszy i wybierz *Przenieś do grupy*.
3. Z menu wybierz *Nowa grupa...*

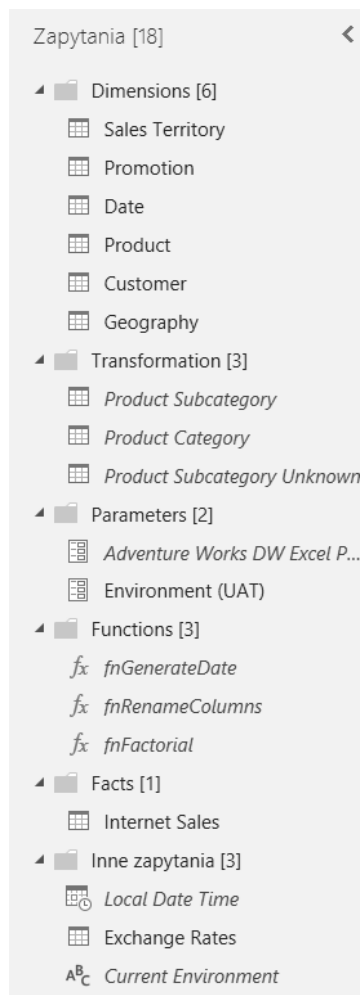
Podane przed chwilą kroki pokazano na rysunku 3.12.



Rysunek 3.12. Grupowanie zapytań w edytorze Power Query

Grupowanie zapytań jest zalecane szczególnie w większym modelach, które mogą mieć wiele zapytań odnoszących się do innych zapytań.

To, jak mogą wyglądać pogrupowane zapytania, przedstawia rysunek 3.13.

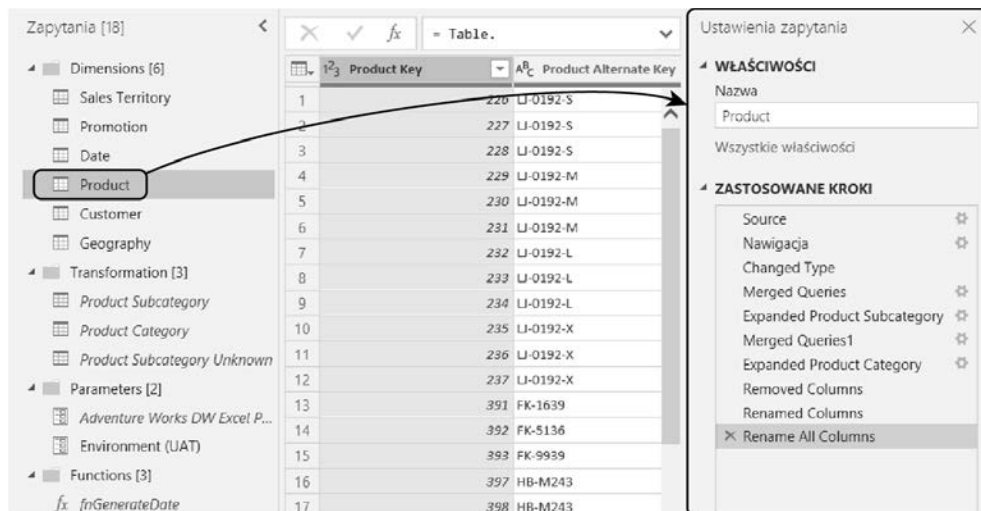


Rysunek 3.13. Zgrupowane zapytania w edytorze Power Query

Panel Ustawienia zapytania

Ten panel, znajdujący się po prawej stronie okna edytora Power Query, zawiera wszystkie właściwości i wszystkie kroki przekształcenia zastosowane do wybranego zapytania (z panelu *Zapytania*). Panel *Ustawienia zapytania* nie pokaże się, jeśli wybierzesz parametr zapytania.

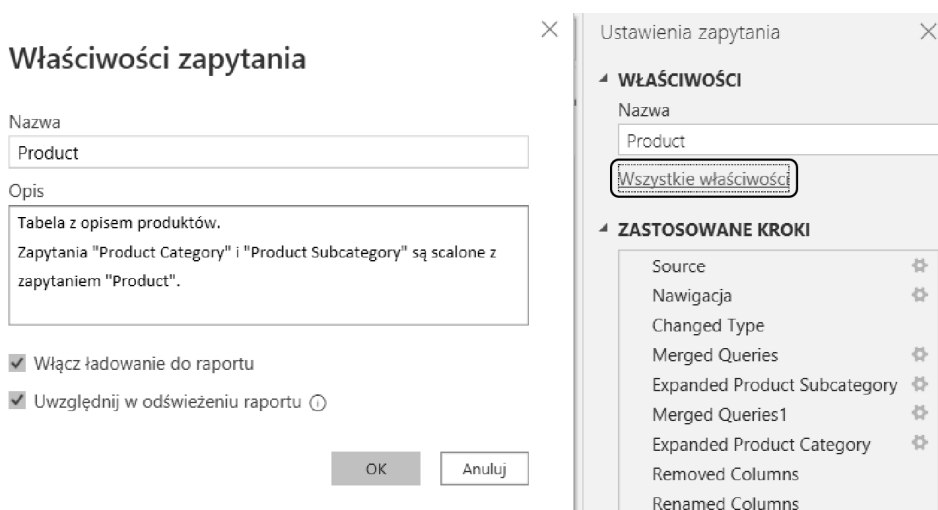
Panel *Ustawienia zapytania* ma dwie części: **WŁAŚCIWOŚCI** i **ZASTOSOWANE KROKI**, co zostało pokazane na rysunku 3.14.



Rysunek 3.14. Panel Ustawienia zapytania w edytorze Power Query

Właściwości zapytania

Możemy zmienić nazwę wybranego zapytania przez wpisanie nowej nazwy w polu tekstowym *Nazwa*. Możemy również ustawić inne właściwości przez kliknięcie *Wszystkie właściwości*, tak jak pokazano na rysunku 3.15.



Rysunek 3.15. Właściwości zapytania w panelu Ustawienia zapytania

Oto co widzimy na rysunku 3.15:

- **Nazwa:** Tutaj znowu mamy nazwę zapytania.
- **Opis:** Możemy tutaj wpisać opis zapytania, który może być bardzo pomocny podczas przygotowywania dokumentacji.
- **Włącz ładowanie do raportu:** Gdy to pole jest zaznaczone, dane będą ładowane do modelu z systemu źródłowego (jednego lub więcej). Jak widać na rysunku 3.15, scaliliśmy zapytanie Product z dwoma innymi. Każde zapytanie może pochodzić z innego źródła danych. Gdy ta opcja nie jest zaznaczona, dane nie będą ładowane do modelu. Jeśli jednak inne zapytania odnoszą się do tego zapytania, dane przejdą wszystkie przekształcenia zastosowane w tym zapytaniu.
- **Uwzględnij w odświeżeniu raportu:** W niektórych przypadkach konieczne jest załadowanie danych do modelu tylko raz, więc nie musimy uwzględniać odświeżania raportu. Gdy ta opcja jest zaznaczona, zapytanie odświeża się wtedy, gdy odświeża się model danych. Model danych możemy odświeżyć z poziomu Power BI Desktop, po naciśnięciu przycisku *Odśwież*, albo przez opublikowanie raportu w usłudze Power BI i odświeżenie danych z poziomu usługi. W każdym razie jeżeli taka opcja jest odznaczona dla danego zapytania, wówczas to zapytanie nie będzie uwzględniane przy okazji kolejnych odświeżeń.

Ważna uwaga

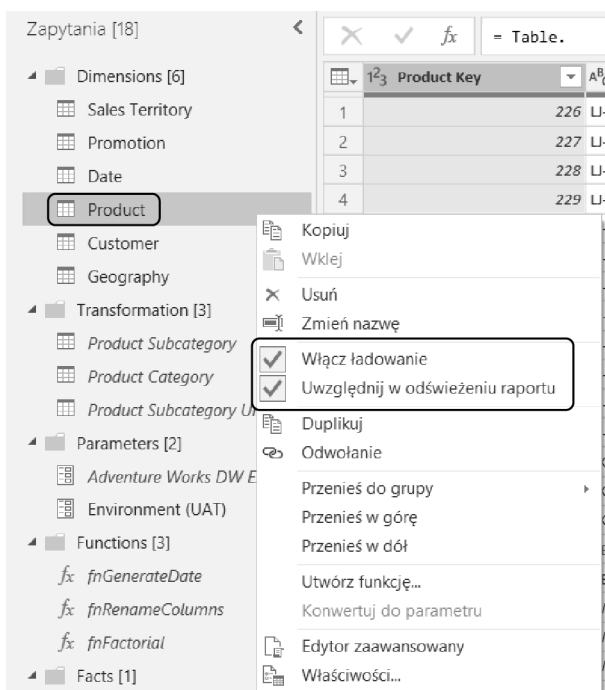
Opcja *Uwzględnij w odświeżeniu raportu* zależy od opcji *Włącz ładowanie do raportu*. Zatem jeśli ładowanie do raportu jest wyłączone, to odświeżanie również będzie wyłączone. To bardzo często stosowana technika w bardziej złożonych scenariuszach, w których wyłącza się ładowanie do raportu dla niektórych zapytań, utworzonych jako zapytania przekształcające. Potem inne zapytania odnoszą się do tych zapytań.

Jak widać na rysunku 3.16, właściwości zapytania, jak również opcje wgrywania i odświeżania możemy ustawić w panelu *Zapytania* przez kliknięcie zapytania prawym przyciskiem myszy.

Panel widoku danych

Panel *Dane* jest umieszczony w centralnej części edytora Power Query. Gdy wybieramy zapytanie w panelu *Zapytania*, w zależności od typu wybranego zapytania widzimy jedną z następujących rzeczy:

- Tabelę z danymi, gdy wybrane zapytanie jest tabelą, co widać na rysunku 3.17.
- **Wprowadź parametry** w celu wywołania funkcji, gdy wybrane zapytanie jest niestandardową funkcją, co widać na rysunku 3.18.
- Wyniki wybranego zapytania. Rysunek 3.19 przedstawia panel *Dane* po wybraniu zapytania, które pobiera lokalną datę i godzinę.



Rysunek 3.16. Opcje Włącz ładowanie i Uwzględnij w odświeżeniu raportu w panelu Zapytania

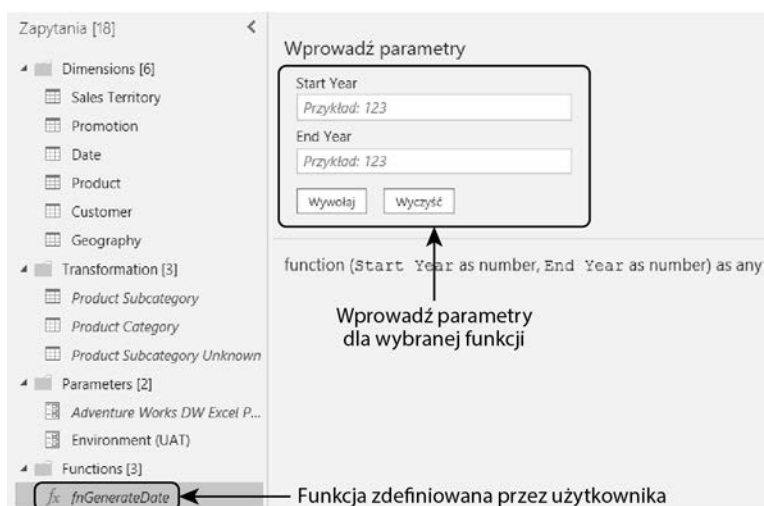


Rysunek 3.17. Panel Dane po wybraniu zapytania, które jest tabelą

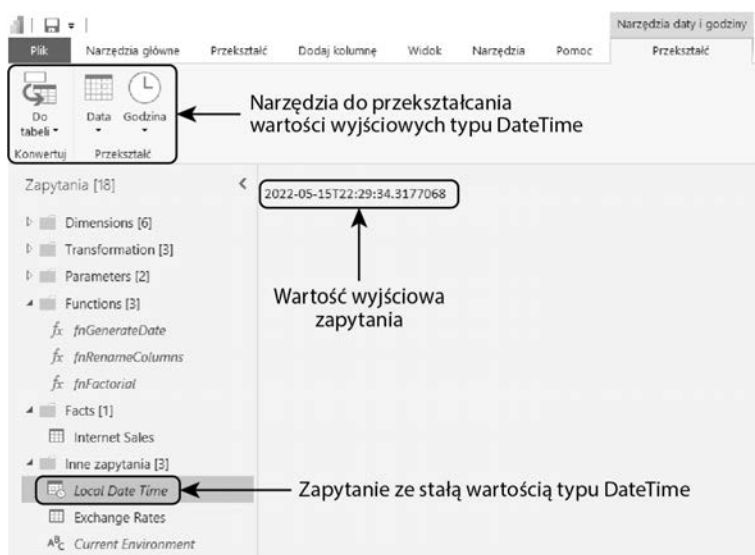
Uwaga

W pasku zadań dostępne są różne narzędzia do przekształcania danych, w zależności od typu danych zwracanych przez wybrane zapytanie. Rysunek 3.20 przedstawia wyniki zapytania, które odwołuje się do parametru Environment. Dlatego wynik zapytania Current Environment zmienia się w zależności od wybranych wartości w parametrze Environment.

Jak widać, narzędzia do przekształcania danych pokazane na rysunkach 3.19 i 3.20 są różne.



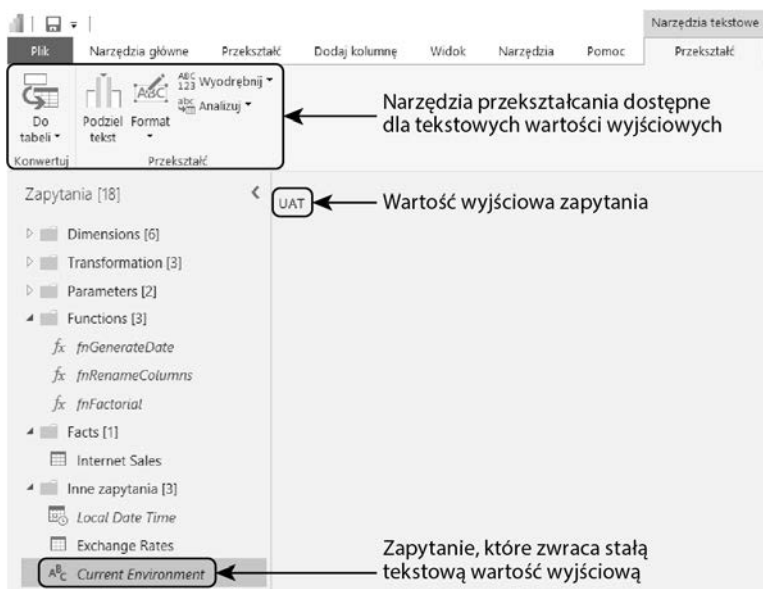
Rysunek 3.18. Panel Dane po wybraniu niestandardowej funkcji w panelu Zapytania



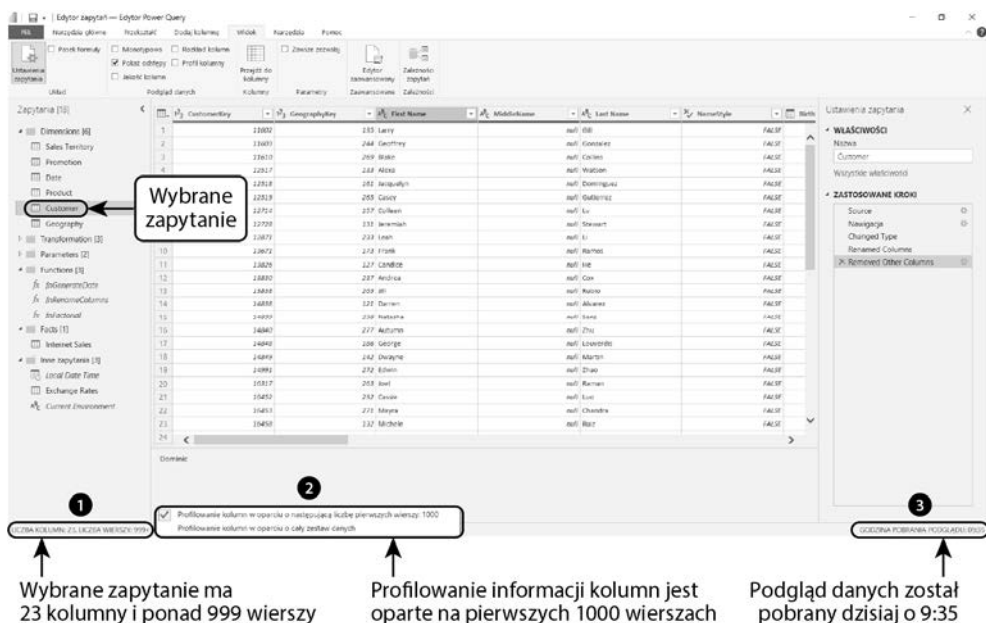
Rysunek 3.19. Panel Dane po wybraniu zapytania, które jest stałą wartością

Pasek stanu

Na dole edytora Power Query znajduje się pasek stanu, który pokazuje informacje na temat zapytania wybranego w panelu *Zapytania*, co widać na rysunku 3.21.



Rysunek 3.20. Narzędzia do przekształceń dostępne dla zapytania, które zwraca wartość tekstową



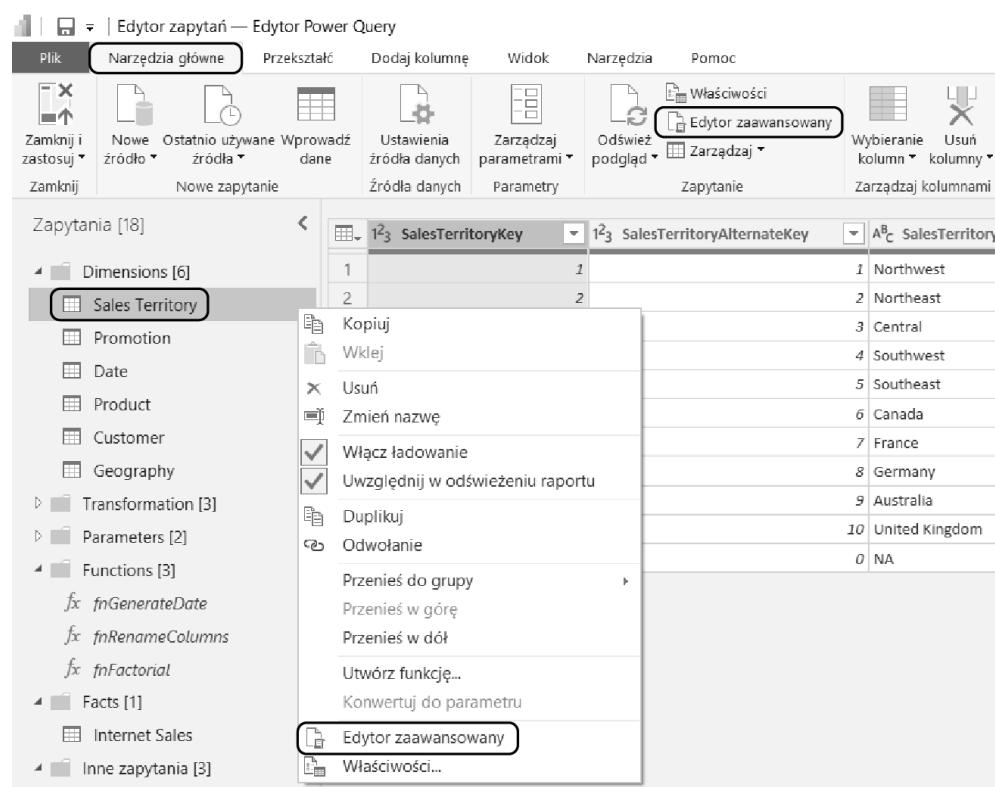
Rysunek 3.21. Pasek stanu w edytorze Power Query

Na rysunku 3.21 widzimy następujące funkcjonalności:

1. Liczba kolumn: Możemy szybko dowiedzieć się, jak szeroka jest tabela.
2. Liczba wierszy będących podstawą profilowania kolumn: To pomaga nam określić, czy profilowanie informacji jest wiarygodne. W niektórych przypadkach *Profilowanie kolumn* pokazuje niepoprawne informacje, gdy jest obliczane na podstawie 1000 wierszy (co jest ustawieniem domyślnym).
3. Informacja, kiedy dane zostały odświeżony po raz ostatni.

Edytor zaawansowany

Aby utworzyć nowe zapytanie lub wprowadzić zmiany w już istniejącym zapytaniu, możemy użyć *Edytora zaawansowanego*, który jest dostępny w różnych miejscach edytora Power Query, co widać na rysunku 3.22.



Rysunek 3.22. Otwarcie Edytora zaawansowanego

Aby użyć *Edytora zaawansowanego*, wykonaj następujące kroki:

1. Wybierz zapytanie w panelu *Zapytania*.
2. Naciśnij *Edytor zaawansowany* w zakładce *Narzędzia główne* w pasku zadań albo kliknij prawym przyciskiem myszy zapytanie i wybierz z menu *Edytor zaawansowany*. Oba sposoby są pokazane na rysunku 3.22.

Wprowadzenie do funkcji Power Query przydatnych modelarzom danych

W tym podrozdziale przyjrzymy się funkcjom, które są obecnie dostępne w edytorze Power Query i pomagają modelarzom danych szybciej znajdować i naprawiać błędy. Modelarze danych mogą ocenić jakość danych, statystyki i rozkład danych w obrębie kolumny (nie w całym zestawie danych). Na przykład modelarz danych może szybko zobaczyć, jak bardzo unikatowe są wartości kolumny, ile jest pustych wartości i tak dalej.

Uwaga

Jak już wspomniano wcześniej, informacje pokazane w polach *Jakość kolumn*, *Rozkład kolumn* i *Profil kolumn* są (domyślnie) oparte na pierwszych 1000 wierszach, co w pewnych przypadkach prowadzi do błędnych informacji. Dobrze jest tak ustawić *Profil kolumny*, by był obliczany w oparciu o cały zestaw danych, gdy ten zestaw nie jest zbyt duży. Jednak to podejście może zabrać więcej czasu niezbędnego na wgranie informacji związanych z profilowaniem kolumny, gdy zestaw składa się z dużej ilości danych. Dlatego bądź ostrożny, wybierając tę opcję, gdy pracujesz z bardzo dużymi tabelami.

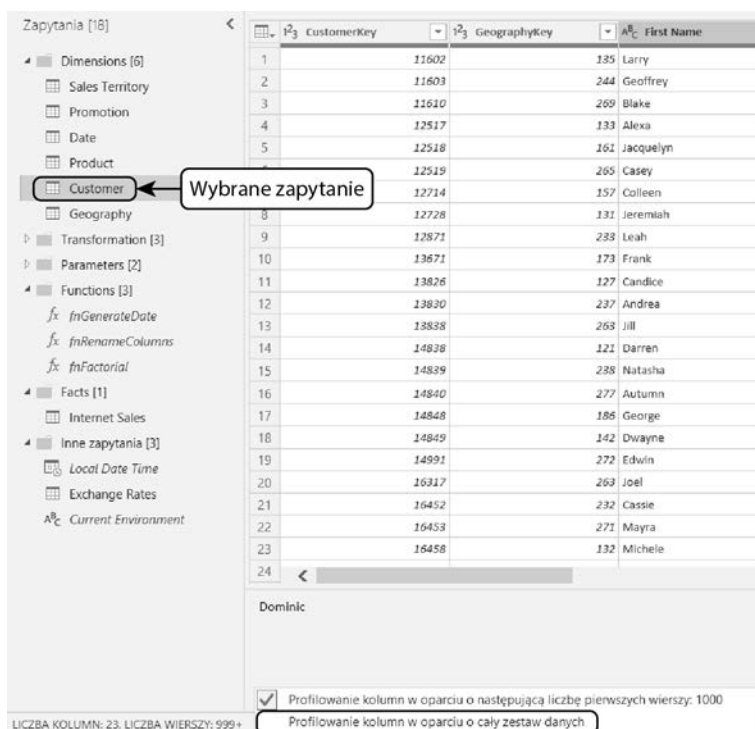
Aby zmienić poprzednie ustawienie w pasku statusu:

1. Naciśnij *Profilowanie kolumn w oparciu o następującą liczbę pierwszych wierszy: 1000*.
2. Wybierz *Profilowanie kolumn w oparciu o cały zestaw danych*.

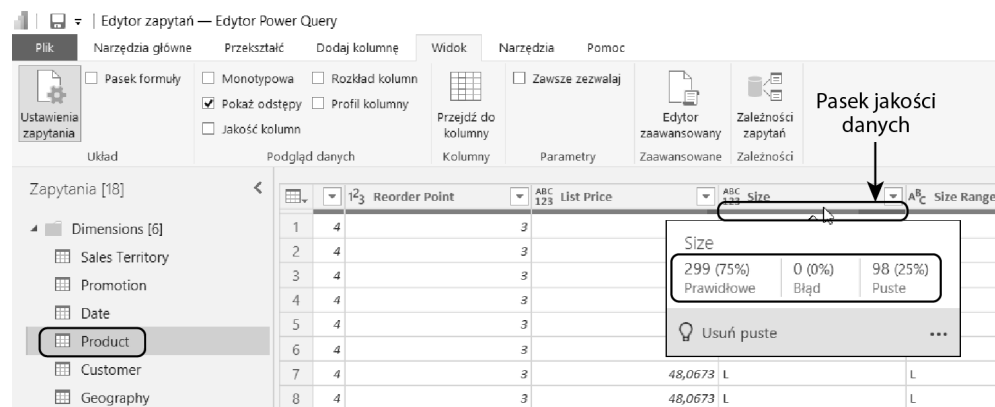
Na rysunku 3.23 pokazano, jak to zrobić.

Jakość kolumn

W edytorze Power Query, pod nagłówkiem każdej kolumny, widzimy zielony pasek, który szybko pokazuje jakość danych w kolumnie. Pasek ten nazywamy *paskiem jakości danych*. Gdy na niego najedziemy kursorem myszy, pokaże się okno, w którym znajdziemy więcej informacji związanych z jakością danych. Rysunek 3.24 pokazuje jakość danych kolumny *Size* w tabeli *Product*.



Rysunek 3.23. Ustawienie profilowania kolumny w oparciu o cały zestaw danych



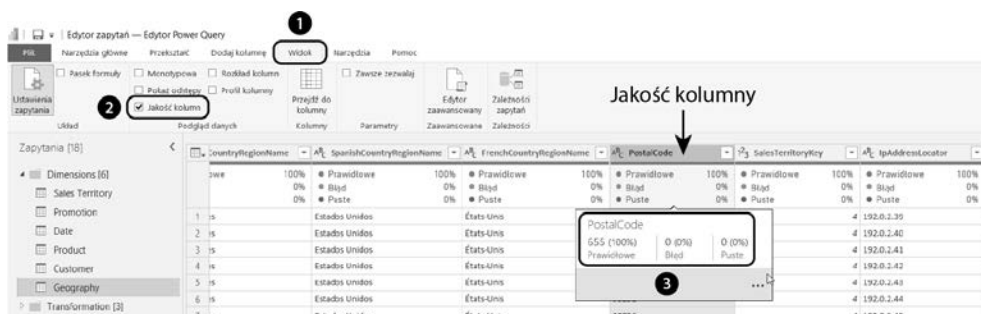
Rysunek 3.24. Pasek jakości danych w edytorze Power Query

Mimo że to świetna funkcjonalność, nadal dość ciężko ocenić jakość danych. W edytorze Power Query mamy jeszcze jedną funkcjonalność — o nazwie *Jakość kolumn*.

1. W edytorze Power Query przejdź do zakładki *Widok*.
2. Zaznacz opcję *Jakość kolumn*.

3. Jeśli najedziesz na pole pokazujące jakość kolumn, wyświetli się okno z dodatkowymi informacjami.

Jak widać na rysunku 3.25, dzięki opcji *Jakość kolumn* możemy szybko ocenić wartości kolumn i zobaczyć, czy są jakieś błędy, zapoznać się z procentem poprawnych i pustych wartości. To bardzo przydatne w szukaniu błędów. Możemy również szybko znaleźć kolumny, które mają dużo pustych wartości, i być może je usunąć.



Rysunek 3.25. Włączanie podglądu jakości kolumn w edytorze Power Query

Możemy również wykonać pewne działania z poziomu pojawiającego się okna. Wystarczy nacisnąć trzy kropki znajdujące się w prawym dolnym rogu wyświetlonego okienka, co widać na rysunku 3.26.

RegionName	FrenchCountryRegionName	PostalCode	SalesTerritoryKey	IpAddressLocator
100%	100%	100%	100%	100%
0%	0%	0%	0%	0%
0%	0%	0%	0%	0%
États-Unis	États-Unis	655 (100%) Prawidłowe	4	192.0.2.39
États-Unis	États-Unis	0 (0%) Błąd	4	192.0.2.40
États-Unis	États-Unis	0 (0%) Puste	4	192.0.2.41
États-Unis	États-Unis		4	192.0.2.42
États-Unis	États-Unis		4	192.0.2.43
États-Unis	États-Unis		4	192.0.2.44
États-Unis	États-Unis		4	192.0.2.45
États-Unis	États-Unis		4	192.0.2.46
États-Unis	États-Unis		4	192.0.2.47
États-Unis	États-Unis		4	192.0.2.48
États-Unis	États-Unis		4	192.0.2.49
États-Unis	États-Unis		4	192.0.2.50
États-Unis	États-Unis		4	192.0.2.51
États-Unis	États-Unis		4	192.0.2.52
États-Unis	États-Unis		4	192.0.2.53
États-Unis	États-Unis		4	192.0.2.54

Rysunek 3.26. Działania dostępne w okienku ze szczegółowymi danymi na temat jakości kolumny

Rysunek 3.26 pokazuje, że możemy skopiować informacje na temat jakości danych, co może być pomocne przy tworzeniu dokumentacji. Z poziomu tego okna możemy również podjąć działania związane z błędami, między innymi usunąć je. Dobrze jest przejrzeć błędy i naprawić je tam, gdzie to tylko możliwe, ale starajmy się usuwać błędy tylko tam, gdy jest to naprawdę konieczne.

Zobaczmy inne przykłady użycia funkcjonalności *Jakość kolumn*, by przekonać się, jak może nam ona pomóc w codziennej pracy. W tym scenariuszu będziemy korzystać z pliku o nazwie *Edytor zapytań.pbix* dla rozdziału 3.

Z tabeli Customer chcemy usunąć wszystkie kolumny, w których procent poprawnych danych wynosi poniżej 10%. Należy to zrobić tak:

1. Otwórz plik *Edytor zapytań.pbix*.
2. Otwórz edytor Power Query.
3. Rzuć okiem na pola jakości kolumn, by zobaczyć, że można usunąć następujące kolumny:
 1. Title
 2. Suffix
 3. AddressLine2.

Na rysunku 3.27 widać, że te kolumny mają dużo pustych wartości.

Możemy usunąć te kolumny w następujących krokach:

1. Naciśnij zakładkę *Narzędzia główne*.
2. Naciśnij przycisk *Wybieranie kolumn*.
3. Odznacz wymienione wyżej kolumny.
4. Kliknij *OK*.

Podane właśnie kroki pokazano na rysunku 3.28.

Rozkład kolumn

Rozkład kolumn to kolejna funkcjonalność, która dostarcza więcej informacji na temat rozkładu danych, odrębnych danych i unikatowych danych. Informacje o rozkładzie kolumn mogą być przydatne dla modelarzy danych do tego, by mogli ocenić kardynalność wybranej kolumny. **Kardynalność kolumn** to kluczowy temat w modelowaniu danych, szczególnie z punktu widzenia zarządzania pamięcią i kompresją danych.

Dzięki opcji *Rozkład kolumn* możemy ocenić, czy ładować daną kolumnę do modelu, czy raczej ją z niego usunąć. Usunięcie niepotrzebnych kolumn może nam potencjalnie pomóc w optymalizacji rozmiaru pliku i poprawie wydajności.

1			2			3		
CustomerKey			Title			Suffix		
	- Prawidłowe 100% - Błąd 0% - Puste 0%			- Prawidłowe 0% - Błąd 0% - Puste 100%			- Prawidłowe 0% - Błąd 0% - Puste 100%	
1	11602			null			null	
2	11603			null			null	
3	11610			null			null	
4	12517			null			null	
5	12518			null			null	
6	12519			null			null	
7	12714			null			null	
8	12728			null			null	
9	12871			null			null	
10	13671			null			null	
11	13826			null			null	
12	13830			null			null	
13	13838			null			null	
14	14838			null			null	
15	14839			null			null	
16	14840			null			null	
17	14848			null			null	
18	14849			null			null	
19	14991			null			null	
20	16317			null			null	
21	16452			null			null	
22	16453			null			null	
23	16458			null			null	
24	16460			null			null	
25	16461			null			null	
26	18038			null			null	

Rysunek 3.27. Szukanie kolumn, w których jest mniej niż 10% poprawnych danych, za pomocą pola jakości kolumn

Uwaga

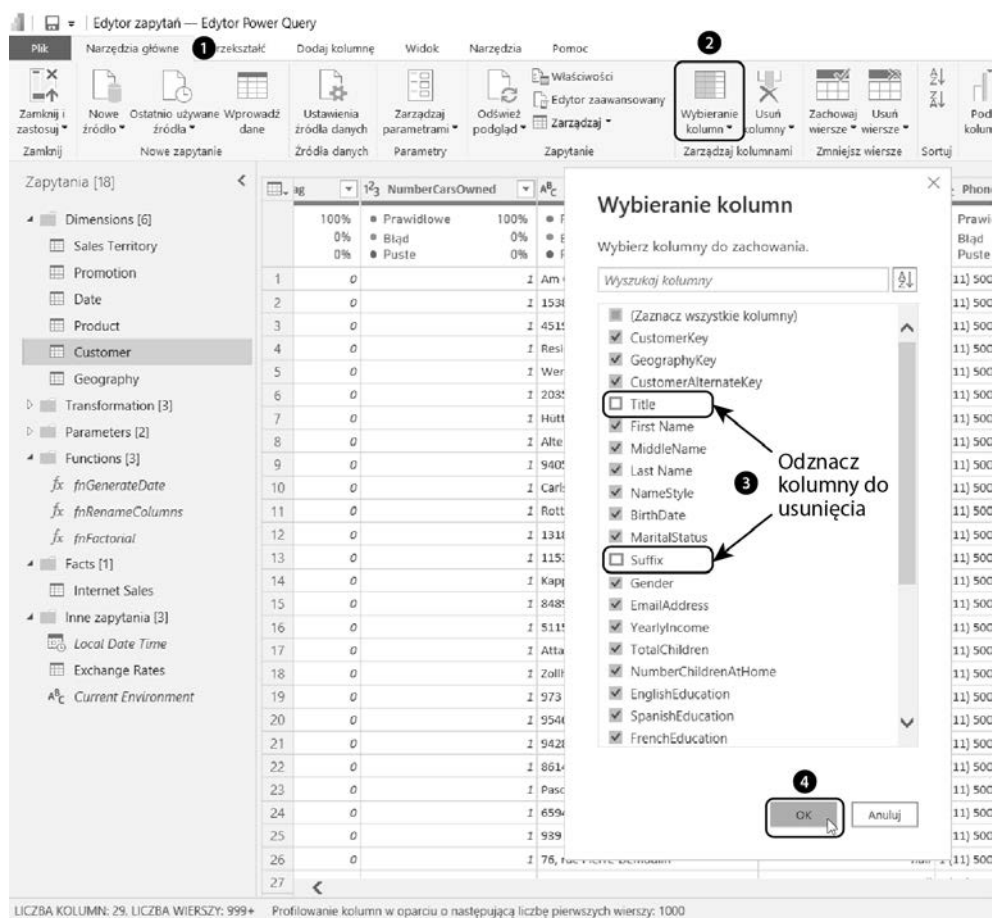
Ogólna zasada mówi, że liczba kardynalna powinna być jak najniższa. Gdy silnik xVelocity wgrywa dane do modelu, lepiej kompresuje dane z niższą liczbą kardynalną. Zatem kolumny z niską kardynalnością mają mniej (lub w ogóle nie mają) unikatowych wartości.

Aby widzieć informację o rozkładzie kolumn, w zakładce *Widok* zaznacz opcję *Rozkład kolumn* (rysunek 3.29).

Po włączeniu tej funkcji pod polem *Jakość kolumn* pokaże się nowe pole z informacjami o rozkładzie kolumn. Jeśli najedziesz kursorem myszy na pole *Jakość kolumn*, pokaże się nowe okno z bardziej szczegółowymi informacjami o rozkładzie kolumny, co pokazano na rysunku 3.30.

Możemy skopiować informacje z tego okna i podjąć odpowiednie działania, które pokazano na rysunku 3.30.

Przyjrzyjmy się *Rozkładowi kolumn* w działaniu, opierając się na praktycznym przykładzie.



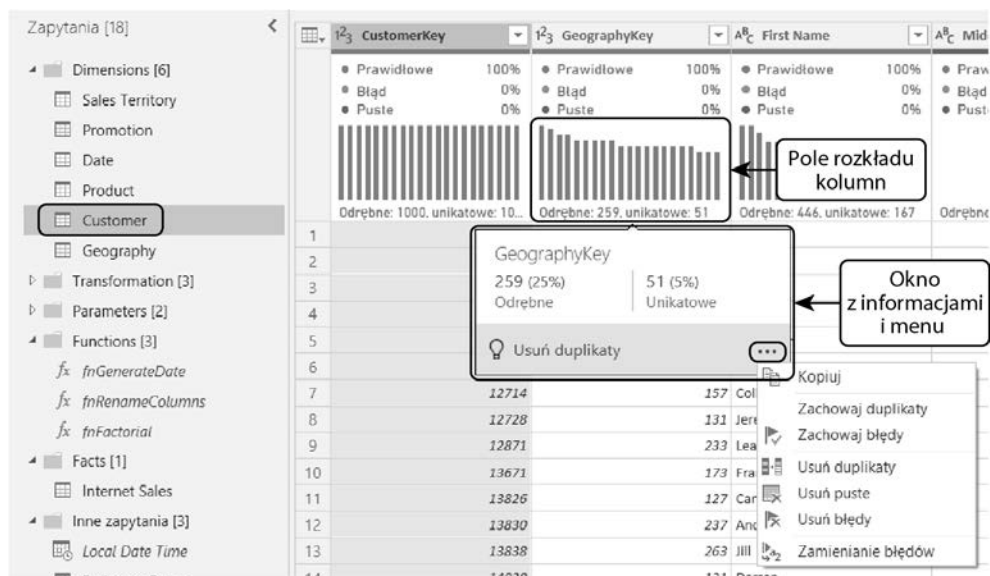
Rysunek 3.28. Usuwanie kolumn



Rysunek 3.29. Włączenie opcji Rozkład kolumn w edytorze Power Query

Spójrz na tabelę Customer (klient), którą znajdziesz w pliku *Edytor zapytań.pbix*. Chcemy znaleźć kolumny, które można usunąć, i te, które należy omówić z firmą, by móc zoptymalizować rozmiar pliku i zużycie pamięci.

1. Wybierz tabelę Customer z pola Zapytania.
2. Ustaw profilowanie kolumn na *Profilowanie kolumn w oparciu o cały zestaw danych*.



Rysunek 3.30. Okno pojawiające się po najechaniu na pole Rozkład kolumn

- Przejrzyj informacje w polu *Rozkład kolumn* każdej kolumny, by znaleźć te z największą liczbą kardynalną.

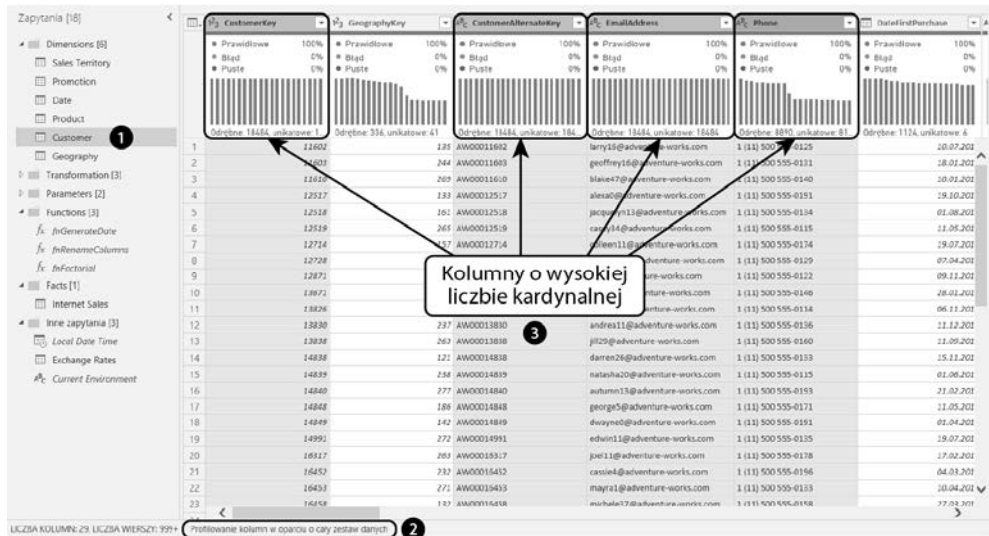
Oto kolumny z wysoką kardynalnością:

- CustomerKey (klucz klienta),
- CustomerAlternateKey (alternatywny klucz klienta),
- EmailAddress (adres e-mailowy),
- Phone (telefon).

Na rysunku 3.31 te kolumny zostały zaznaczone.

Kolumna CustomerKey nie jest kandydatem do usunięcia, gdyż w modelu danych bierze udział w relacji między tabelą Internet Sales a Customer. Możemy usunąć kolumnę CustomerAlternateKey. To nadmiarowa kolumna z bardzo wysoką liczbą kardynalną (100% unikatowych wartości), a ponadto nie wnosi nic do analizy danych. Pozostałe dwie kolumny należy omówić z firmą, by dowiedzieć się, czy można je usunąć z tabeli Customer.

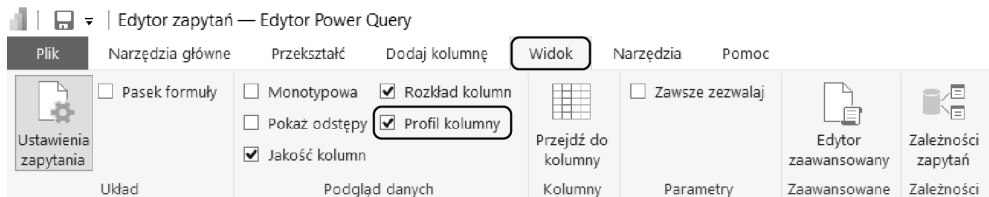
Jeśli usuniemy wszystkie trzy kolumny, możemy zmniejszyć rozmiar pliku z 2485 kilobajtów (kB) do 1975 kB. To znaczna oszczędność pamięci, szczególnie w przypadku bardzo dużych modeli.



Rysunek 3.31. Szukanie kolumn o wysokiej liczbie kardynalnej w tabeli Customer

Profil kolumny

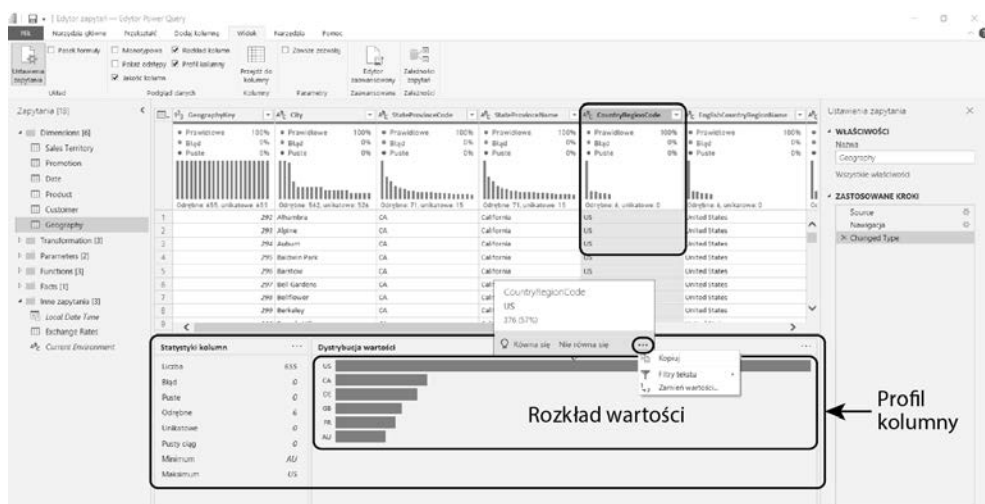
Jak dotąd poznaliśmy funkcje *Jakość kolumn* i *Rozkład kolumn*. Możemy także włączyć opcję *Profil kolumny*, by zobaczyć więcej informacji na temat wartości wybranej kolumny. Aby to zrobić, w zakładce *Widok* zaznacz opcję *Profil kolumny*, co pokazano na rysunku 3.32.



Rysunek 3.32. Włączenie opcji Profil kolumny w edytorze Power Query

Jak widać na rysunku 3.33, po zaznaczeniu opcji *Profil kolumny* możemy zobaczyć *Statystyki kolumny* i *Dystrybucję wartości*. Gdy najedziemy kursorem myszki na daną wartość, zobaczymy, ile razy występuje w kolumnie oraz jaki stanowi procent wszystkich wartości. Na wybranej wartości możemy również wykonać pewne działania dostępne w menu, które pokaże się po naciśnięciu trzech kropek znajdujących się w prawym dolnym rogu, co pokazano na rysunku 3.33.

Do tej pory wyjaśnialiśmy, jak język Power Query działa i jak używać edytora Power Query w Power BI. Przyjrzelśmy się również kilku funkcjom pomocnym podczas modelowania danych.



Rysunek 3.33. Profil kolumny

W kilku kolejnych punktach następnego podrozdziału poruszymy więcej tematów związanych z implementacją, między innymi kwestię parametrów zapytań, i przejdziemy przez kilka praktycznych scenariuszy, w których będziemy mieli okazję użyć tych parametrów.

Parametry zapytania

Jedną z najcenniejszych funkcjonalności jest możliwość definiowania parametrów zapytań, których możemy użyć w różnych sytuacjach. Na przykład możemy utworzyć zapytanie z parametrem, by pobierać dane z różnych źródeł, lub możemy sparazytować filtrowanie wierszy. Parametry możemy zastosować do następujących elementów i działań:

- źródła danych,
- filtrów wierszy,
- zachowania wierszy,
- usunięcia wierszy,
- zastąpienia wierszy.

Co więcej, do modelu danych możemy wgrać wartości parametrów, aby móc się do nich odwoływać z poziomu miar, kolumn obliczeniowych, tabel obliczeniowych i elementów raportu, jeśli zajdzie taka potrzeba.

W edytorze Power Query parametr zapytania możemy bardzo łatwo zdefiniować.

1. Naciśnij *Zarządzaj parametrami*.
2. Wybierz *Nowy parametr*.

3. Podaj nazwę.
4. Wprowadź opis parametru, który wyjaśni jego przeznaczenie.
5. Jeśli zaznaczysz pole *Wymagane*, parametr będzie obowiązkowy.
6. Wybierz typ z listy rozwijalnej.
7. Wybierz wartość z listy rozwijalnej *Sugerowane wartości*.
8. W zależności od sugerowanej wartości wybranej w poprzednim kroku możliwe, że będziesz musiał wpisać kilka wartości (patrz rysunek 3.34). Jeśli z listy rozwijalnej *Sugerowane wartości* wybrałeś *Zapytanie*, będziesz musiał wybrać zapytanie.
9. W zależności od wybranej wcześniej sugerowanej wartości będziesz widział wartość domyślną lub nie będziesz jej widział. Jeśli wybrałeś *Listę wartości*, musisz wybrać wartość domyślną.
10. Wybierz lub wpisz wartość jako *wartość bieżąca*.
11. Kliknij OK.

Podane wyżej kroki zostały zilustrowane na rysunku 3.34.

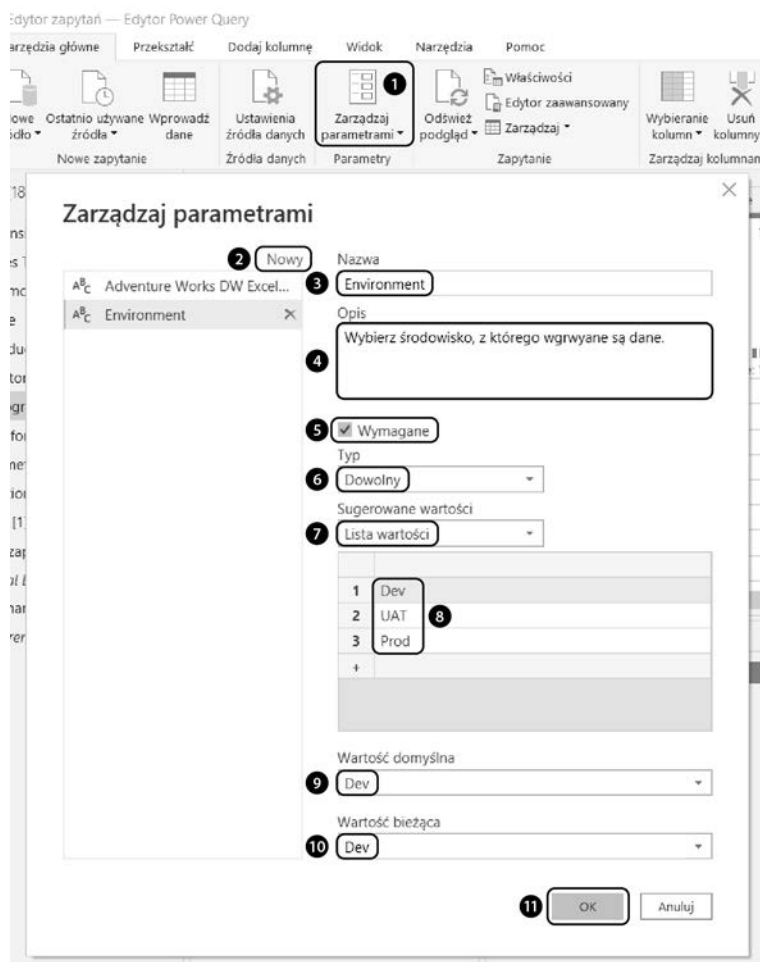
Uwaga

Unikaj podawania źródeł danych na sztywno, a zamiast tego stwórz do tego celu parametr. W niektórych organizacjach nazwy źródeł danych i łańcuchy połączeń są danymi wrażliwymi. Co więcej, pozwalają udostępniać w organizacji lub stosować z zewnętrznymi narzędziami jedynie pliki PBIT, gdyż te pliki nie zawierają danych. Jeśli więc źródła danych nie są sparametryzowane, mogą ujawniać nazwy serwerów, ścieżki folderów, adresy URL SharePointów i tak dalej. Użycie parametru z sugerowaną wartością *Dowolna wartość* zapobiega wyciekowi danych.

Liczba przypadków użycia parametrów zapytań jest dość duża. Przyjrzyjmy się z życia wziętemu przykładowi, w którym przekonamy się, jak wygodne jest stosowanie parametrów zapytań.

W tym scenariuszu chcemy sparametryzować źródła danych. Podawanie źródła danych jako parametru jest pomocne w różnych sytuacjach, od łączenia się z różnymi źródłami danych po wgrywanie różnych zestawień kolumn. Jedną z największych zalet parametryzacji źródeł danych jest możliwość uniknięcia podawania na sztywno nazw serwerów, nazw baz danych, plików, ścieżek folderów i tak dalej.

Firma ma określoną platformę zarządzania analityką biznesową, która wymaga oddzielnych środowisk: **programistycznego (Dev)**, **testowego (UAT)** i **produkcyjnego (Prod)**. Firma wymaga od nas przygotowania raportu z analizą sprzedaży na podstawie danych z magazynu dostępnego na serwerze SQL. Mamy trzy różne serwery, każdy dla jednego środowiska, gdzie jest dostępny magazyn. Jako że raport Power BI jest w fazie tworzenia, musi się łączyć z serwerem *Dev* i pobierać dane z bazy *Dev*. Gdy raport już będzie gotowy do przetestowania w środowisku *UAT*, zarówno serwer, jak i baza danych będą musiały być przełączone na to właśnie



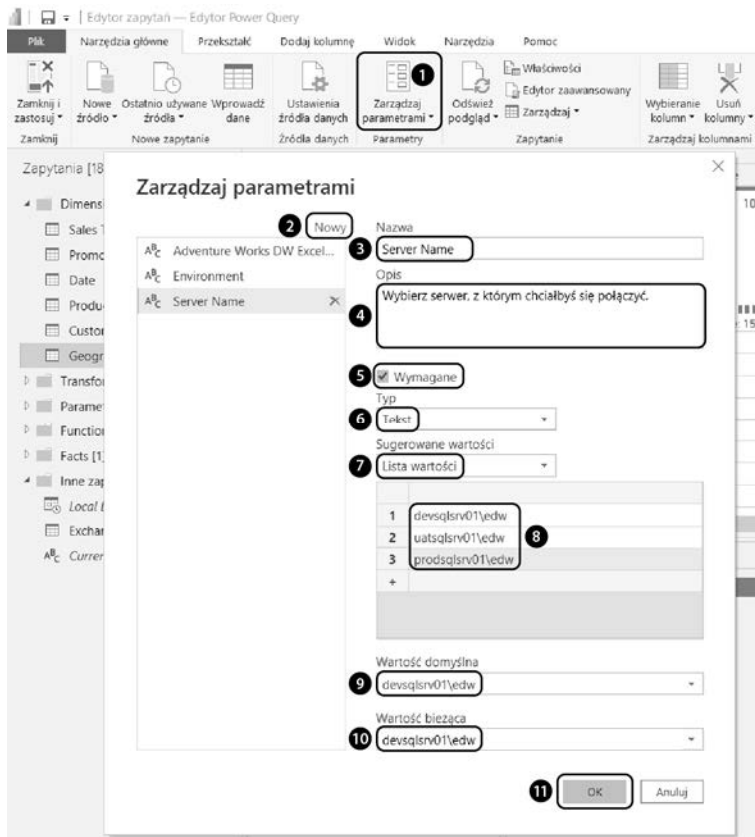
Rysunek 3.34. Definiowanie nowego parametru zapytania

środowisko. Gdy testerzy zakończą swoją pracę i raport będzie gotowy do wdrożenia na produkcję, znowu musimy przełączyć serwer i bazę danych na środowisko *Prod*. Aby zaimplementować ten scenariusz, musimy zdefiniować dwa parametry zapytania. Jeden zapisuje nazwę serwera, a drugi nazwę bazy danych. Następnie ustawimy wszystkie powiązane z tym zapytania, by używały tych parametrów. Będzie nam łatwiej, jeśli od początku projektu będziemy używać parametrów zapytań. Ale nie martw się, jeśli już masz gotowy raport Power BI i chciałbyś w nim sparametryzować źródła danych, bo to bardzo łatwo zrobić. Gdy już to ustawisz, nie musisz w przyszłości zmieniać żadnego kodu, by przełączać się między różnymi środowiskami. Utwórzmy zatem nowy parametr.

1. W edytorze Power Query naciśnij *Zarządzaj parametrami*.
2. Naciśnij *Nowy parametr*.
3. Podaj nazwę *Server Name* (nazwa serwera).

4. Wpisz parę słów opisu.
5. Zaznacz pole *Wymagane*.
6. Z listy wybieralnej *Typ* wybierz *Tekst*.
7. Z listy wybieralnej *Sugerowane wartości* wybierz *Lista wartości*.
8. Wpisz nazwy środowisk.
9. Jako wartość domyślną wybierz *devsqlsrv01\edw*.
10. Ponownie wybierz *devsqlsrv01\edw*, tym razem jako wartość bieżącą.
11. Naciśnij OK.

Podane właśnie kroki widać na rysunku 3.35.



Rysunek 3.35. Utworzenie parametru zapytania dla nazw serwerów dla różnych środowisk

Musimy wykonać te same kroki, by utworzyć kolejny parametr zapytania dla nazw baz danych. Jeśli nazwy baz danych są takie same, możemy pominąć ten krok. Rysunek 3.36 przedstawia drugi parametr dla nazw baz danych.

Zarządzaj parametrami

Nowy

- Adventure Works DW Excel...
- Environment
- Server Name
- Database Name**

Nazwa: Database Name

Opis: Wybierz nazwę bazy danych.

☒ Wymagane

Typ: Tekst

Sugerowane wartości: Lista wartości

1	EDWSalesDev
2	EDWSalesUAT
3	EDWSalesProd
+	

Wartość domyślna: EDWSalesDev

Wartość bieżąca: EDWSalesDev

OK Anuluj

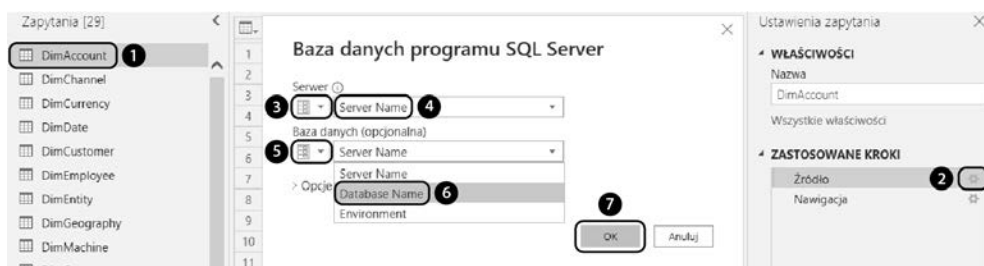
Rysunek 3.36. Tworzenie parametru zapytania dla nazwy bazy danych w różnych środowiskach

Jeśli mamy już zapytania, musimy zmienić źródła danych w następujący sposób:

1. Naciśnij zapytanie, które chcesz sparametryzować.
2. Naciśnij ikonę koła zębatego przy pierwszym kroku *Źródło*.
3. Z listy rozwijalnej *Serwer* wybierz *Parametr*.
4. Z listy rozwijalnej z parametrami serwera wybierz parametr *Server Name*.
5. Z listy rozwijalnej *Baza danych* (opcjonalna) ponownie wybierz *Parametr*.
6. Z listy rozwijalnej z parametrami bazy danych wybierz parametr *Database Name*.
7. Naciśnij *OK*.

Podane właśnie kroki przedstawia rysunek 3.37.

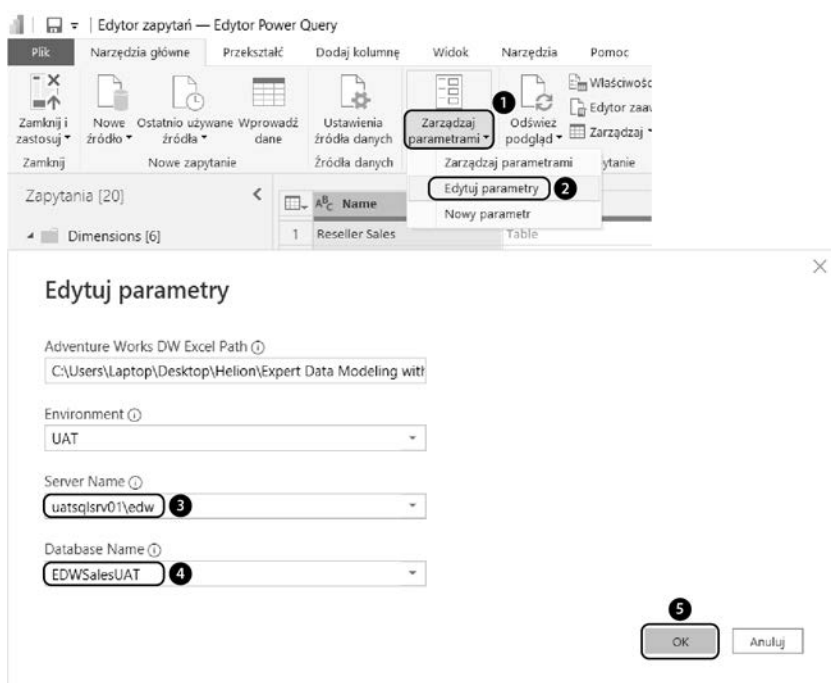
Aby sparametryzować pozostałe zapytania, musimy wykonać podobne kroki. Gdy już to zrobimy, to by zmienić źródło danych, wystarczy, że zmienimy wartość parametru. W tym celu wykonaj następujące kroki w edytorze Power Query:



Rysunek 3.37. Parametryzowanie źródła danych

1. Naciśnij menu rozwijalne *Zarządzaj parametrami*.
2. Wybierz *Edytuj parametry*.
3. Wybierz UAT jako nazwę serwera.
4. Wybierz UAT jako nazwę bazy danych.
5. Naciśnij OK.

Podane wyżej kroki przedstawia rysunek 3.38.



Rysunek 3.38. Zmiana wartości parametrów w edytorze Power Query

Możemy również zmienić wartości parametrów w głównym oknie Power BI.

1. Naciśnij listę rozwijalną *Przekształć dane*.
2. Wybierz *Edytuj parametry*.
3. Wybierz UAT jako nazwę serwera (*Server Name*).
4. Wybierz UAT jako nazwę bazy danych (*Database Name*).
5. Naciśnij OK.

Wymienione powyżej kroki zostały pokazane na rysunku 3.39.



Rysunek 3.39. Zmiana wartości parametrów w głównym oknie Power BI

Niestandardowe funkcje

W licznych przypadkach możemy mieć potrzebę wielokrotnego obliczania pewnych wartości. Wtedy warto stworzyć **niestandardową funkcję**, która obsługuje logikę obliczeń. Po zdefiniowaniu niestandardowej funkcji możemy wywoływać ją wielokrotnie. Jak już wspomniano w podrzdziale „Wprowadzenie do języka Power Query M w Power BI”, w punkcie „Wartości”, niestandardową funkcję możemy stworzyć przez podanie listy parametrów (jeśli jakieś są) w nawiasie, wraz z typem zwracanej wartości i symbolem =>, po których podajemy definicję funkcji.

Oto przykład prostej niestandardowej funkcji, która pobiera datę i dodaje do niej jeden dzień:

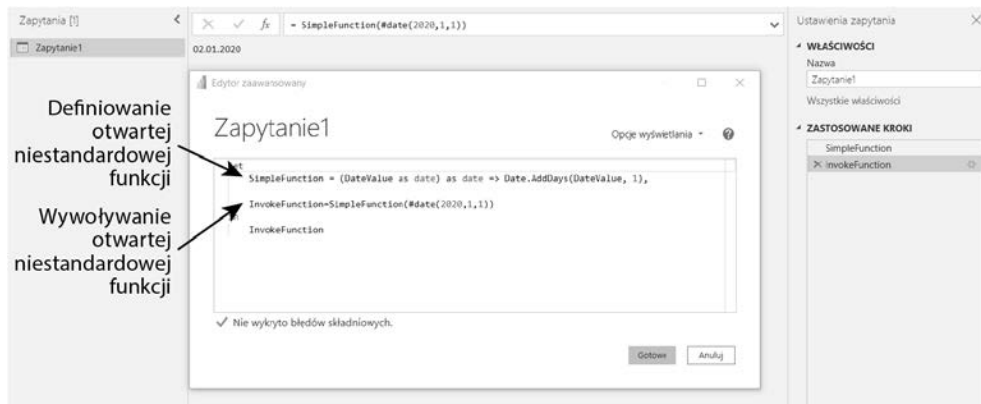
```
SimpleFunction = (DateValue as date) as date => Date.AddDays(DateValue, 1)
```

Tę funkcję możemy wywołać w następujący sposób:

```
SimpleFunction(#date(2020,1,1))
```

Wartością zwracaną przez to wywołanie jest 02.01.2020.

Możemy zdefiniować niestandardową funkcję jako funkcję otwartą (ang. *inline function*) i wywołać ją jako pojedyncze zapytanie w *Edytorze zaawansowanym*. Rysunek 3.40 pokazuje, jak możemy użyć wyżej pokazanego kodu, by zdefiniować SimpleFunction jako funkcję otwartą i wywołać funkcję w obrębie tego samego zapytania.



Rysunek 3.40. Definiowanie i wywołanie otwartej niestandardowej funkcji

Pójdźmy krok dalej i przyjrzyjmy się praktycznemu przykładowi pokazującemu, jak można zaoszczędzić ogromną ilość czasu przez stworzenie niestandardowej funkcji.

Naszym zadaniem jest zbudowanie modelu danych na podstawie 100 tabel. Wstępna analiza pokazuje, że tabele mają od 20 do 150 kolumn. Nazwy kolumn są w postaci połączonych kolejnych wyrazów, a każdy kolejny wyraz zaczyna się wielką literą (ang. *camel case*). Taki zapis nie jest przyjazny dla użytkownika. Mamy dwie możliwości: możemy sami zmienić nazwę każdej kolumny lub znaleźć sposób, by zmienić nazwę wszystkich kolumn naraz. Mimo że musimy to zrobić dla każdej tabeli, możemy zaoszczędzić dużo pracy i czasu przez zmianę nazw kolumn każdej tabeli w jednym kroku. W tym celu możemy stworzyć niestandardową funkcję. Później wywołamy ją dla każdej tabeli.

Spójrzmy na oryginalne nazwy kolumny w jednej z tabel. Na rysunku 3.41 widać pierwotne nazwy kolumn tabeli Product, które nie są zbyt czytelne dla użytkownika. Zauważ, że w pasku statusu widzimy liczbę kolumn wybranej tabeli. W tym przypadku tabela Product ma 37 kolumn. Tak więc zmiana nazwy każdej kolumny ręcznie, przez rozdzielanie wyrazów, byłaby bardzo czasochłonna.

W naszym przykładzie rozdzielanie wyrazów, gdy każdy kolejny zaczyna się wielką literą, jest możliwe. Wykonaj następujące kroki:

1. W edytorze Power Query utwórz puste zapytanie przez kliknięcie listy rozwijalnej *Nowe źródło* i wybranie *Puste zapytanie*, co zostało pokazane na rysunku 3.42.

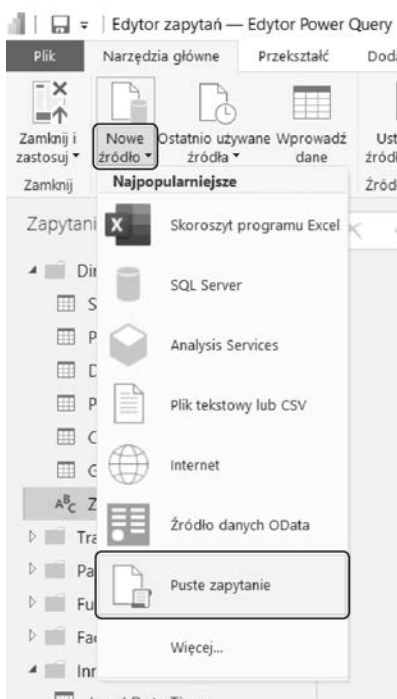
Zapytania [20]

- Dimensions [6]
 - Sales Territory
 - Promotion
 - Date
 - Product**
 - Customer
 - Geography
- Transformation [3]
- Parameters [2]
- Functions [3]
- Fact11

LICZBA KOLUMN: 37 LICZBA WIERZYSZ: 397 Profilowanie kolumn w oparciu o następującą liczbę pierwszych wierszy: 1000

	ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMeasureCode	SizeUnit
1	226	LI-0192-S		21	null
2	227	LI-0192-S		21	null
3	228	LI-0192-S		21	null
4	229	LI-0192-M		21	null
5	230	LI-0192-M		21	null
6	231	LI-0192-M		21	null
7	232	LI-0192-L		21	null
8	233	LI-0192-L		21	null
9	234	LI-0192-L		21	null
10	235	LI-0192-X		21	null
11	236	LI-0192-X		21	null
12					

Rysunek 3.41. Oryginalne nazwy kolumn w tabeli Product



Rysunek 3.42. Utworzenie pustego zapytania w edytorze Power Query

- Otwórz *Edytor zaawansowany*.
- Skopiuj i w *Edytorze zaawansowanym* wklej skrypt pokazany w kroku numer 4, a potem naciśnij **OK**.
- Zmień nazwę zapytania na `fnRenameColumns`, zgodnie z następującym skryptem:

```
let
    fnRename = (ColumnName as text) as text =>
let
    SplitColumnName = Splitter.SplitTextByCharacterTransition({"a"."z"},
        ↪ {"A"."Z"})(ColumnName)
```

```

in
    Text.Combine(SplitColumnName, " ")
in
    fnRename

```

Na rysunku 3.43 pokazano, jak wygląda utworzona właśnie funkcja w edytorze Power Query.



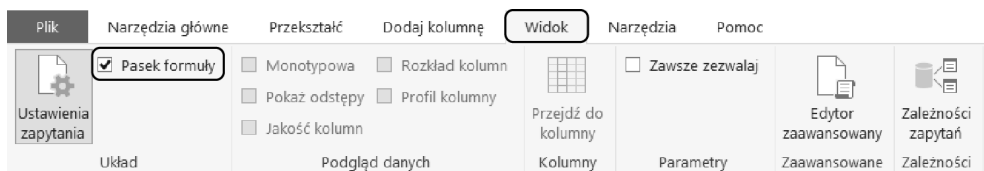
Rysunek 3.43. Tworzenie funkcji niestandardowej

Poprzednia funkcja przyjmuje wartości tekstowe, a następnie rozdziela wartość tekstową na listę wartości tekstowych, jeśli w tekście występuje przejście z małej na wielką literę. Następnie łączymy rozdzielony tekst, umieszczając między kolejnymi słowami spację. Aby zrozumieć, jak działa ta niestandardowa funkcja, musimy przeczytać dokumentację funkcji `Splitter.SplitTextByCharacterTransition()` na stronie Microsoft Docs.

Zauważ, że funkcja `Splitter.SplitTextByCharacterTransition()` zwraca inną funkcję, zatem część `SplitTextByCharacterTransition({"a".."z"}, {"A".."Z"})(ColumnName)` poprzedniego skryptu stosuje funkcję `SplitTextByCharacterTransition()` do parametru wejściowego, którym jest `ColumnName`, i w rezultacie otrzymujemy listę wartości tekstowych.

Teraz wywołajmy funkcję `fnRenameColumns` na tabeli `Product`:

1. Włącz *Pasek formuły* (jeśli go nie widać) w zakładce *Widok*, co pokazano na rysunku 3.44.



Rysunek 3.44. Włączenie paska formuły w edytorze Power Query

2. W panelu *Zapytania* wybierz tabelę `Product`.
3. W *Pasku formuły* naciśnij przycisk *Dodaj krok* ($\mathcal{F}$). To dość wygodne, gdyż można tu zobaczyć nazwę ostatniego kroku, którego teraz użyjemy. Rysunek 3.45 przedstawia, jak wygląda nowy krok.

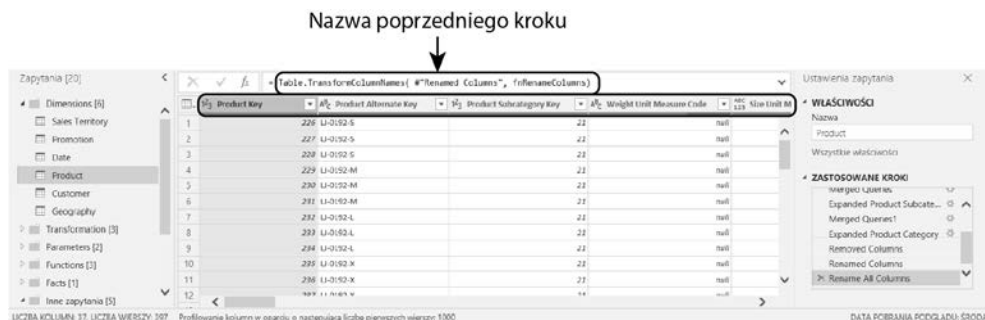


Rysunek 3.45. Dodawanie nowego kroku z poziomu paska formuły

- Teraz użyjemy funkcji `Table.TransformColumnNames()`, która zmienia nazwy kolumn podanej tabeli poprzez podaną funkcję generującą nazwę. Ta tabela pochodzi z poprzedniego kroku, a funkcja generująca nazwy to funkcja `fnRenameColumns`, którą utworzyliśmy wcześniej. Zatem funkcja będzie wyglądać tak:

```
Table.TransformColumnNames("#Renamed Columns", fnRenameColumns)
```

- Po potwierdzeniu uruchomienia kroku wszystkie kolumny tabeli `Product` zostaną natychmiast zmienione tak jak pokazano na rysunku 3.46.



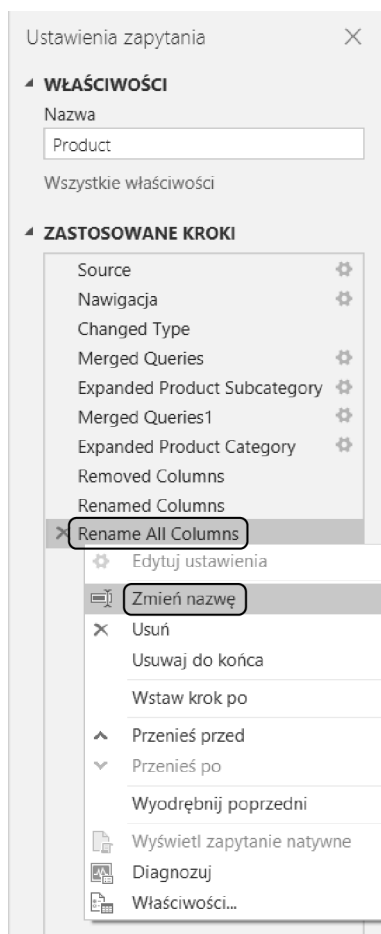
Rysunek 3.46. Jednoczesna zmiana nazw kolumn

- Ostatni krok polega na zmianie nazwy nowego kroku na taką, która będzie nam mówić więcej. W tym celu kliknij prawym przyciskiem myszy wybrany krok i wybierz `Zmień nazwę`, po czym wpisz nową nazwę (rysunek 3.47).

Funkcje rekurencyjne

Możemy odnieść się do funkcji z niej samej, co sprawia, że ta funkcja stanie się rekurencyjna. Silnia jest obliczeniem matematycznym, w którym mnożone są wszystkie kolejne liczby całkowite od dowolnie wybranej liczby, w dół, aż do 1. W matematyce silnię oznacza się wykrzyknikiem (na przykład $n!$). Oto wzór na silnię:

$$n! = n \cdot (n - 1)!$$



Rysunek 3.47. Zmiana nazwy kroku zapytania

Jak widać, silnia jest obliczeniem rekurencyjnym. Silnię możemy obliczyć dla dodatniej liczby całkowitej (liczby całkowitej większej od 0). 0 jest wyjątkiem, jego silnia wynosi 1. Te przykłady pomogą Ci lepiej zrozumieć silnię:

$$10! = 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 3\,628\,800$$

$$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

$$1! = 1$$

$$0! = 1$$

Aby napisać funkcję rekurencyjną i odnieść się do niej w niej samej, musimy użyć operatora @. Na przykład ta funkcja oblicza silnię z wejściowej wartości liczbowej:

```
let
    Factorial =
        (ValidNumber as number) as number =>
```

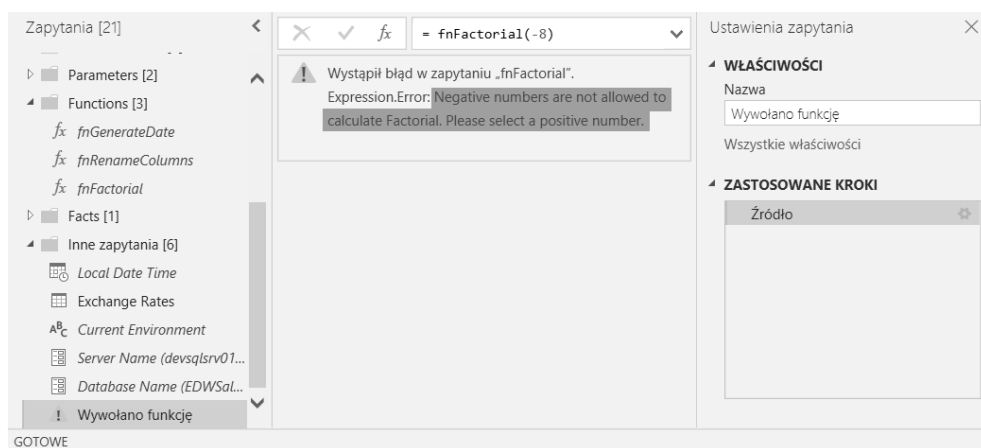
```

if ValidNumber < 0
then error "Negative numbers are not allowed to calculate Factorial.
Please select a positive number."
else
    if ValidNumber = 0
    then 1
    else ValidNumber * @Factorial(ValidNumber - 1)
in
    Factorial

```

Jak możesz zauważyć w podanym właśnie kodzie, zwracamy komunikat błędu, jeśli wartość wejściowa nie jest poprawna. Zwracamy 1, gdy wartość wejściowa jest równa 0, w przeciwnym razie obliczamy silnię rekurencyjnie.

Rysunek 3.48 przedstawia wynik wywołania funkcji Factorial z podaniem niepoprawnej liczby.



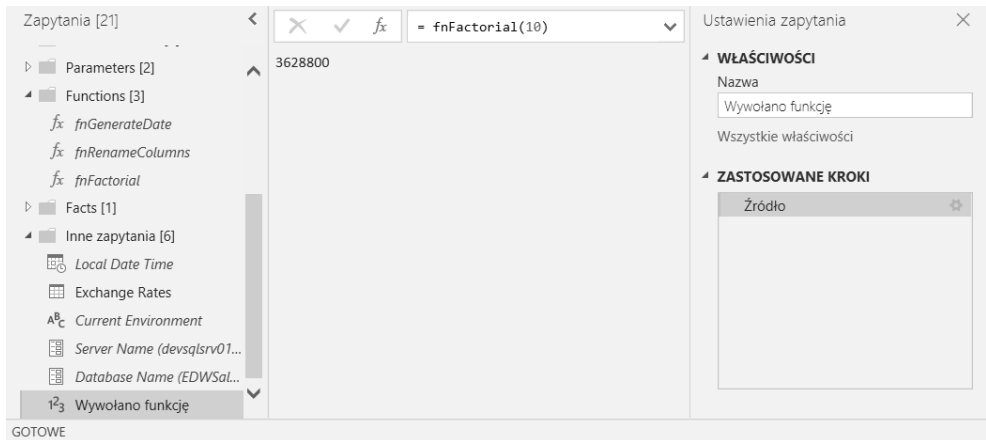
Rysunek 3.48. Komunikat błędu po wywołaniu funkcji Factorial z niepoprawną liczbą

Na rysunku 3.49 pokazano wyniki wywołania funkcji Factorial dla liczby 10.

Podsumowanie

W tym rozdziale wprowadziliśmy różne aspekty języka Power Query M i zobaczyliśmy, do czego możemy użyć edytora Power Query. Przyjrzelśmy się również przykładom i rzeczywistym wyzwaniom, które mogą bezpośrednio wpłynąć na naszą produktywność. Ponadto dowiedzieliśmy się, jak efektywniej zarządzać przygotowaniem danych.

W następnym rozdziale omówimy pobieranie danych z różnych źródeł i kilku trybów połączeń oraz to, jak mogą one wpłynąć na modelowanie danych.



Rysunek 3.49. Rezultat wywołania funkcji Factorial dla liczby 10

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

Optymalny model danych — oto prawdziwa inteligencja biznesowa!

Microsoft Power BI zdobył uznanie jako idealne narzędzie do analizy, modelowania i przetwarzania złożonych zbiorów danych. Dzięki niemu można bez trudu tworzyć wyrafinowane modele danych, łączyć dane z różnych źródeł, definiować relacje między nimi, a także je kształtować i nimi zarządzać. W ten sposób uzyskuje się świetną podstawę do przygotowywania raportów oraz zestawów danych na potrzeby analityki biznesowej — a to przekłada się na bardzo konkretne korzyści.

Ta książka jest znakomitą wprowadzeniem do Power BI. Dzięki niej nauczysz się modelowania danych, technik definiowania relacji oraz tworzenia modeli danych. Dowiesz się też, jak prowadzić obliczenia za pomocą funkcji modelowania. Poznasz także podstawy pisania kodu w języku DAX i korzystania z nowych funkcji modelowania danych. Stopniowo przejdziesz do bardziej zaawansowanych rozwiązań, w efekcie Twoje modele danych sprawdzą się nawet przy bardzo złożonych zadaniach. Poszczególne zagadnienia zilustrowano praktycznymi przykładami, które pozwolą Ci zrozumieć, jak bardzo przydatne w pokonywaniu wyzwań biznesowych są zoptymalizowane modele danych.

W książce między innymi:

- korzystanie z wirtualnych tabel i funkcji analizy czasowej języka DAX
- tabele wymiarów i tabele faktów oraz ich implementacja w edytorze Power Query
- przygotowywanie danych do budowy schematu gwiazdy
- najlepsze metody przygotowywania i modelowania danych
- różne koncepcje modelowania danych i zmniejszania poziomu złożoności modelu

Soheil Bakhshi jest założycielem firmy Data Vizioner i wziętym konsultantem w branży analityki biznesowej. Specjalizuje się w zagadnieniach Microsoft BI i Power BI oraz w stosowaniu hurtowni danych. Uzyskał certyfikaty MCSE i MCSA oraz tytuł Microsoft MVP. Często występuje jako prelegent na prestiżowych konferencjach dotyczących Power BI. Jest zwolennikiem prostoty i wydajności.

	KOD KORZYŚCI Sięgnij po więcej! 
 helion.pl  HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	ISBN 978-83-283-9451-3  9 788328 394513
Cena: 119,00 zł	

Packt