

# Power Query w Excelu i Power BI

Zbieranie  
i przekształcanie  
danych

Gil Raviv

Tytuł oryginału: Collect, Combine, and Transform Data Using Power Query in Excel  
and Power BI (Business Skills)

Tłumaczenie: Andrzej Watrak

ISBN: 978-83-8322-471-8

Authorized translation from the English language edition, entitled COLLECT, COMBINE, AND TRANSFORM DATA USING POWER QUERY IN EXCEL AND POWER BI, 1st Edition by RAVIV, GIL, published by Pearson Education, Inc, publishing as Microsoft Press, Copyright © 2019 by Gil Raviv.

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 Pearson Education, Inc.

Polish language edition published by Helion S.A., Copyright © 2020, 2023.

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” web page are trademarks of the Microsoft group of companies. All other marks are the property of their respective owners.

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.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: [helion@helion.pl](mailto: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/poquqv>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>Przedmowa</b> .....	<b>11</b>
<b>O autorze</b> .....	<b>13</b>
<b>Podziękowania</b> .....	<b>14</b>
<b>Wprowadzenie</b> .....	<b>17</b>
<b>Rozdział 1. Wprowadzenie do Power Query</b> .....	<b>27</b>
Czym jest Power Query? .....	28
Historia Power Query w skrócie .....	29
Gdzie znajduje się Power Query? .....	32
Główne komponenty Power Query .....	33
Pobieranie danych i tworzenie połączeń .....	33
Najważniejsze elementy edytora Power Query .....	34
Ćwiczenie 1.1. Pierwsze kroki z Power Query .....	39
Podsumowanie .....	43
<b>Rozdział 2. Podstawowe operacje przygotowywania danych</b> .....	<b>45</b>
Wyodrębnianie informacji z zakodowanych danych .....	46
Fabryka Przygód .....	46
Ćwiczenie 2.1. Stary sposób: formuły Excela .....	47
Ćwiczenie 2.2. Część I. Nowy sposób .....	48
Ćwiczenie 2.2. Część II. Scalanie tabel wyszukiwania .....	51
Ćwiczenie 2.2. Część III. Tabele faktów i wyszukiwania .....	55
Kolumna z przykładów .....	57
Ćwiczenie 2.3. Część I. Kolumna z przykładów — wprowadzenie .....	57
Kolumna z przykładów w praktyce .....	59
Ćwiczenie 2.3. Część II. Zamiana wielkości na oznaczenie zakresu .....	59
Wyodrębnianie informacji z kolumn tekstowych .....	62
Ćwiczenie 2.4. Wyodrębnianie odnośników z komunikatów tekstowych .....	62

Operacje na danych .....	69
Ćwiczenie 2.5. Operacje na różnych formatach dat .....	69
Ćwiczenie 2.6. Operacje na danych z różnymi ustawieniami regionalnymi .....	71
Wyodrębnianie elementów daty i czasu .....	74
Przygotowanie modelu .....	75
Ćwiczenie 2.7. Dzielanie danych na tabelę faktów i tabelę wyszukiwania .....	75
Ćwiczenie 2.8. Zamiana wartości oddzielonych ogranicznikami na wiersze .....	78
Podsumowanie .....	80
<b>Rozdział 3. Łączenie danych z kilku źródeł .....</b>	<b>81</b>
Łączenie kilku tabel .....	81
Łączenie dwóch tabel .....	82
Ćwiczenie 3.1. Rowery i akcesoria .....	82
Ćwiczenie 3.2. Część I. Przekształcenie „Dołącz zapytania jako nowe” .....	84
Ćwiczenie 3.2. Część II. Zależności między zapytaniami i odwołania .....	84
Łączenie trzech lub więcej tabel .....	87
Ćwiczenie 3.2. Część III. Rowery + Akcesoria + Komponenty .....	87
Ćwiczenie 3.2. Część IV. Rowery + Akcesoria + Komponenty + Ubrania .....	88
Łączenie tabel na szerszą skalę .....	89
Łączenie tabel zapisanych w folderze .....	89
Ćwiczenie 3.3. Łączenie skoroszytów z produktami zapisanymi w folderze .....	90
Uwagi do importowania plików z folderu .....	92
Łączenie arkuszy w skoroszycie .....	92
Ćwiczenie 3.4. Łączenie arkuszy — rozwiązanie .....	93
Podsumowanie .....	97
<b>Rozdział 4. Łączenie niezgodnych tabel .....</b>	<b>99</b>
Problem z niezgodnymi tabelami .....	99
Co to jest niezgodność tabel? .....	100
Symptomy niezgodności tabel i związane z tym ryzyka .....	100
Ćwiczenie 4.1. Uzgadnianie kolumn: podejście reaktywne .....	101
Łączenie niezgodnych tabel zapisanych w folderze .....	102
Ćwiczenie 4.2. Część I. Symptom braku wartości .....	102
Ćwiczenie 4.2. Część II. Założenie takiej samej kolejności kolumn i ujednoczenie nagłówek .....	104
Ćwiczenie 4.3. Proste ujednoczenie nagłówek za pomocą funkcji Table.TransformColumnNames .....	105
Tabela konwersji .....	108
Ćwiczenie 4.4. Transpozycja z użyciem tabeli konwersji .....	109
Ćwiczenie 4.5. Anulowanie przestawienia, scalenie i powtórne przestawienie .....	113
Ćwiczenie 4.6. Transpozycja samych nazw kolumn .....	114
Ćwiczenie 4.7. Ujednoczenie nazw kolumn za pomocą języka M .....	119
Podsumowanie .....	122

<b>Rozdział 5. Pozyskiwanie kontekstu .....</b>	<b>123</b>
Pozyskiwanie kontekstu zawartego w nazwach plików i skoroszytów .....	124
Ćwiczenie 5.1. Część I. Kolumna niestandardowa .....	124
Ćwiczenie 5.1. Część II. Pozyskiwanie kontekstu z nazwy pliku lub arkusza .....	125
Pozyskiwanie kontekstu z tytułów tabel przed ich połączeniem .....	126
Ćwiczenie 5.2. Pozyskiwanie kontekstu z tytułów tabel za pomocą przekształcenia Drill Down .....	127
Ćwiczenie 5.3. Pozyskiwanie kontekstu z tytułów tabel podczas importowania plików z folderu .....	130
Pozyskiwanie kontekstu z tytułów tabel po ich połączeniu .....	133
Ćwiczenie 5.4. Pozyskiwanie kontekstu z tytułów tabel zapisanych w arkuszach jednego skoroszytu .....	133
Wskazówki kontekstowe .....	137
Ćwiczenie 5.5. Kolumna indeksu jako wskazówka .....	137
Ćwiczenie 5.6. Wyszukiwanie informacji kontekstowych w pobliżu określonych komórek .....	140
Podsumowanie .....	143
<b>Rozdział 6. Dekomponowanie tabel .....</b>	<b>145</b>
Rozpoznawanie źle zaprojektowanych tabel .....	146
Wprowadzenie do anulowania przestawienia .....	148
Ćwiczenie 6.1. Przekształcenia Anuluj przestawienie kolumn i Anuluj przestawienie innych kolumn .....	149
Ćwiczenie 6.2. Anulowanie przestawienia tylko zaznaczonych kolumn .....	151
Przetwarzanie sum całkowitych .....	152
Ćwiczenie 6.3. Dekompozycja tabeli zawierającej sumy całkowite .....	153
Dekompozycja tabeli z hierarchią agregacji danych 2×2 .....	155
Ćwiczenie 6.4. Dekompozycja tabeli z hierarchią agregacji danych 2×2 i datami .....	155
Ćwiczenie 6.5. Dekompozycja tabeli z hierarchią agregacji danych 2×2 .....	158
Przetwarzanie sum częściowych .....	160
Ćwiczenie 6.6. Przetwarzanie sum częściowych .....	160
Podsumowanie .....	162
<b>Rozdział 7. Zaawansowane metody dekomponowania i agregowania tabel .....</b>	<b>163</b>
Dekomponowanie tabeli z wielopoziomą hierarchią agregacji danych .....	164
Wirtualna tabela przestawna, pola wierszy i kolumn .....	164
Ćwiczenie 7.1. Dekompozycja tabeli z hierarchią agregacji danych N×M .....	165
Uogólnienie sekwencji przekształceń dekomponujących tabelę .....	167
Ćwiczenie 7.2. Zaczynając od końca .....	168
Ćwiczenie 7.3. Tworzenie funkcji FnDekomponujTabelę .....	169
Przekształcenie Kolumna przestawna .....	178
Ćwiczenie 7.4. Przywrócenie niewłaściwie zdekomponowanej tabeli .....	178
Ćwiczenie 7.5. Agregowanie tabel z wielowierszowymi rekordami .....	180
Podsumowanie .....	184

<b>Rozdział 8. Praca grupowa .....</b>	<b>185</b>
Pliki lokalne, parametry i szablony .....	186
Niewłaściwe korzystanie z lokalnych plików .....	186
Ćwiczenie 8.1. Zdefiniowanie parametru z nazwą ścieżki .....	187
Ćwiczenie 8.2. Tworzenie szablonów w Power BI .....	189
Ćwiczenie 8.3. Definiowanie parametrów w Excelu .....	191
Praca nad współdzielonymi plikami i folderami .....	197
Importowanie danych z plików zapisanych w usługach OneDrive dla Firm i SharePoint .....	197
Ćwiczenie 8.4. Tworzenie zapytań łączących się z usługami OneDrive dla Firm i SharePoint .....	199
Ćwiczenie 8.5. Od lokalnych folderów do usługi SharePoint .....	200
Zagadnienia bezpieczeństwa .....	202
Usuwanie zapytań w drodze inspekcji skoroszytu Excela .....	203
Podsumowanie .....	204
<b>Rozdział 9. Wprowadzenie do języka M .....</b>	<b>205</b>
Nauka języka M .....	206
Etapy opanowania języka M .....	206
Informacje online .....	209
Informacje offline .....	209
Ćwiczenie 9.1. Zmienna #shared i dokumentacja wbudowanych funkcji .....	209
Bloki konstrukcyjne w języku M .....	211
Ćwiczenie 9.2. Witaj, świecie! .....	211
Wyrażenie let .....	213
Zakresy i scalanie wyrażeń z wielu zapytań .....	215
Typy danych, operatory i wbudowane funkcje .....	217
Podstawowe typy danych w języku M .....	218
Typ number .....	219
Typ time .....	220
Typ date .....	220
Typ duration .....	221
Typ text .....	222
Typ null .....	222
Typ logical .....	223
Typy złożone .....	224
Typ list .....	224
Typ record .....	227
Typ table .....	230
Warunki i wyrażenia „if” .....	232
Wyrażenie „if-then-else” .....	233
Wyrażenie if wewnątrz wyrażenia let .....	233

Funkcje niestandardowe .....	235
Wywoływanie funkcji .....	236
Wyrażenie each .....	237
Zaawansowane tematy .....	237
Obsługa błędów .....	238
Leniwe i zachłanne przetwarzanie wyrażeń .....	239
Pętle .....	239
Rekurencja .....	240
Funkcja List.Generate .....	240
Funkcja List.Accumulate .....	241
Podsumowanie .....	242
<b>Rozdział 10. Od złych do dobrych zapytań .....</b>	<b>245</b>
Źródła pułapek i ich skutki .....	246
Świadome działanie .....	247
Dobre praktyki .....	248
Modyfikacje formuł M .....	248
Pułapka 1. Ignorowanie paska formuły .....	249
Ćwiczenie 10.1. Wyszukiwanie w pasku formuły statycznych odwołań do nazw kolumn .....	249
Pułapka 2. Zmiana typów danych .....	251
Pułapka 3. Niebezpieczne filtrowanie .....	253
Ćwiczenie 10.2. Część I. Filtrowanie czarnych produktów .....	254
Działanie warunków w filtrach .....	255
Ćwiczenie 10.2. Część II. Wyszukiwanie wartości w panelu filtru .....	257
Pułapka 4. Zmiana kolejności kolumn .....	257
Ćwiczenie 10.3. Część I. Zmiana kolejności wybranych kolumn .....	258
Ćwiczenie 10.3. Część II. Niestandardowa funkcja FnZmianaKolejnościKolumn ...	260
Pułapka 5. Usuwanie i wybieranie kolumn .....	260
Ćwiczenie 10.4. Usunięcie niepotrzebnych kolumn .....	261
Pułapka 6. Zmiana nazw kolumn .....	263
Ćwiczenie 10.5. Zmiana nazw losowych kolumn .....	263
Pułapka 7. Dzielenie kolumn .....	266
Ćwiczenie 10.6. Niewłaściwy podział kolumny .....	267
Pułapka 8. Scalanie kolumn .....	269
Inne pułapki i techniki zapobiegawcze .....	270
Podsumowanie .....	270
<b>Rozdział 11. Podstawy analizy tekstu .....</b>	<b>273</b>
Wyszukiwanie słów kluczowych w kolumnach tekstowych .....	274
Ćwiczenie 11.1. Proste wyszukiwanie słów kluczowych .....	274
Wyszukiwanie słów kluczowych z użyciem iloczynu kartezjańskiego .....	277
Ćwiczenie 11.2. Iloczyn kartezjański .....	279
Ćwiczenie 11.3. Wyszukiwanie słów kluczowych za pomocą niestandardowej funkcji ...	285

Jakiej metody użyć: zwykłego wyszukiwania, iloczynu kartezjańskiego czy niestandardowej funkcji? .....	288
Dzielenie wpisów na słowa .....	288
Ćwiczenie 11.4. Trywialne dzielenie wpisów na słowa .....	288
Ćwiczenie 11.5. Odrzucenie stop-słów .....	293
Ćwiczenie 11.6. Wyszukiwanie słów kluczowych w wyodrębnionych słowach .....	295
Ćwiczenie 11.7. Chmura słowna w Power BI Desktop .....	302
Podsumowanie .....	304
<b>Rozdział 12. Zaawansowana analiza tekstu: rozpoznawanie znaczenia .....</b>	<b>305</b>
Usługa Microsoft Azure Cognitive Services .....	306
Klucze API i zasoby Azure .....	307
Uwagi dotyczące korzystania z usługi Azure Cognitive Services za pomocą edytora Power Query .....	310
Tłumaczenie tekstu .....	312
Opis interfejsu Translator Text API .....	312
Ćwiczenie 12.1. Proste tłumaczenie .....	313
Ćwiczenie 12.2. Tłumaczenie wielu komunikatów .....	317
Analiza nastroju tekstu .....	321
Czym jest interfejs Sentiment Analytics API .....	322
Ćwiczenie 12.3. Implementacja niestandardowej funkcji FnOkreślenieNastroju .....	324
Ćwiczenie 12.4. Przetwarzanie dużych ilości danych .....	333
Wyodrębnianie kluczowych fraz .....	335
Ćwiczenie 12.5. Przystosowanie kodu do wyodrębniania kluczowych fraz .....	336
Obsługa wielu języków .....	338
Zmiana oznaczenia języka .....	338
Dynamiczne wykrywanie języka .....	339
Ćwiczenie 12.6. Przystosowanie kodu do wykrywania języka .....	339
Podsumowanie .....	340
<b>Rozdział 13. Analiza sieci społecznościowych .....</b>	<b>341</b>
Pierwsze kroki z konektorem do Facebooka .....	342
Ćwiczenie 13.1. Uzyskiwanie informacji o ulubionych stronach .....	342
Analiza sieci znajomych .....	347
Ćwiczenie 13.2. Wyszukiwanie bezpośrednich i pośrednich znajomych za pomocą Power BI .....	347
Ćwiczenie 13.3. Wyszukiwanie stron, które polubili znajomi .....	349
Analiza stron na Facebooku .....	351
Ćwiczenie 13.4. Proste odczytywanie wpisów i komentarzy ze stron Facebooka .....	352
Krótka dygresja: pobieranie danych za wybrany okres .....	355
Ćwiczenie 13.5. Analiza aktywności użytkowników: zliczanie komentarzy i udostępnień .....	356
Ćwiczenie 13.6. Analiza porównawcza stron .....	358
Podsumowanie .....	361



<b>Rozdział 14. Końcowy projekt: wszystko razem .....</b>	<b>363</b>
Ćwiczenie 14.1. Ratowanie sytuacji u Światowych Importerów .....	363
Wskazówki .....	364
Część I. Przygotowanie danych .....	365
Część II. Dekompozycja tabel .....	367
Część III. Agregacja danych za 2018 r. ....	368
Część IV. Scalenie przychodów za lata 2015 – 2017 i za rok 2018 .....	369
Ćwiczenie 14.2. Porównanie tabel i śledzenie hakera .....	369
Wskazówki .....	370
Rozwiązanie .....	370
Szukanie śladów hakera w zmienionej tabeli .....	371
Podsumowanie .....	372



## ROZDZIAŁ 4.

# Łączenie niezgodnych tabel

*Połącz ekstrema, a otrzymasz prawdziwy środek.*

— Friedrich Schlegel

### W TYM ROZDZIALE NAUCZYSZ SIĘ:

- oceniać skutki błędnego połączenia tabel,
- poprawnie łączyć tabele z kolumnami o niezgodnych nazwach,
- poprawnie łączyć niezgodne tabele zapisane w folderze,
- ujednolicać nazwy kolumn za pomocą tabeli konwersji,
- stosować sposoby ujednolicania tabel różniące się złożonością i wydajnością.

W rozdziale 3., „Łączenie danych z kilku źródeł”, dowiedziałeś się, jak łączyć tabele, aby uprościć sobie analizę danych i tworzenie raportów. Często zdarza się, że tabele mają różne formaty lub stopniowo zmieniają się w wyniku modyfikowania ich przez różnych użytkowników nieświadomych potencjalnego zagrożenia, na jakie narażają w ten sposób tworzenie raportów.

Ten rozdział jest poświęcony poważnemu zadaniu: łączeniu niezgodnych tabel. Aby uświadomić sobie skalę problemu, wyobraźmy sobie, że tabele są podobne do skarpetek. Operacje, które wykonywałeś w rozdziale 3., były podobne do zakładania skarpetek. W tym rozdziale dowiesz się, jak sobie radzić ze skarpetkami, które są nie do pary. Na pewno zgodzisz się, że jeżeli skarpetki są takie same, noszenie ich jest przyjemnością. Jednak prawdziwym koszmarem jest zakładanie różnych skarpetek. Czasami znalezienie w stosie ubrań dwóch jednakowych skarpetek okazuje się niemożliwością. Czy można połączyć ekstrema i założyć skarpetki nie do pary? W praktyce, jeżeli mamy na myśli tabele, jest to możliwe. Ten rozdział opisuje, jak za pomocą Power Query radzić sobie z takimi tabelami, choćby były nie wiadomo jak niezgodnie.

## Problem z niezgodnymi tabelami

---

W rozdziale 3. łączyłeś tabele, wykorzystując różne techniki. Ręcznie importowałeś kilka tabel i łączyłeś je, stosując przekształcenia *Dołącz zapytania* i *Dołącz zapytania jako nowe*. Importowałeś również tabele zapisane w folderze oraz w różnych arkuszach skoroszytu. We wszystkich ćwiczeniach przyjęte było milczące założenie, że wszystkie tabele mają ten sam format.

Zrozumiałe jest, że tabele przeznaczone do połączenia muszą mieć wspólny format. Jest to jednak idealistyczne założenie, rzadko sprawdzające się w praktyce. Często będziesz miał do czynienia z różnymi formatami tabel lub — co gorsza — z formatami, które w miarę upływu czasu będą różniły się coraz bardziej. W tym podrozdziale opisane są najczęściej spotykane przypadki niezgodności tabel i sposoby oceniania ich skutków.

## Co to jest niezgodność tabel?

W tym rozdziale niezgodne tabele są to tabele zawierające tego samego rodzaju dane umieszczone w kolumnach o różnych nazwach. Załóżmy, że *Tabela1* składa się z kolumn o nazwach *pracownik*, *wiek* i *pleć*, natomiast *Tabela2* z kolumn *Nazwisko*, *Wiek* i *K/M*. Tabela 4.1 przedstawia różnice pomiędzy kolumnami obu tabel.

TABELA 4.1. Przykład niezgodnych nazw kolumn

	Tabela1	Tabela2	Niezgodność
Kolumna 1	<i>pracownik</i>	<i>Nazwisko</i>	Różne nazwy
Kolumna 2	<i>wiek</i>	<i>Wiek</i>	Różne wielkości liter w nazwach
Kolumna 3	<i>pleć</i>	<i>K/M</i>	Różne nazwy

## Symptomy niezgodności tabel i związane z tym ryzyka

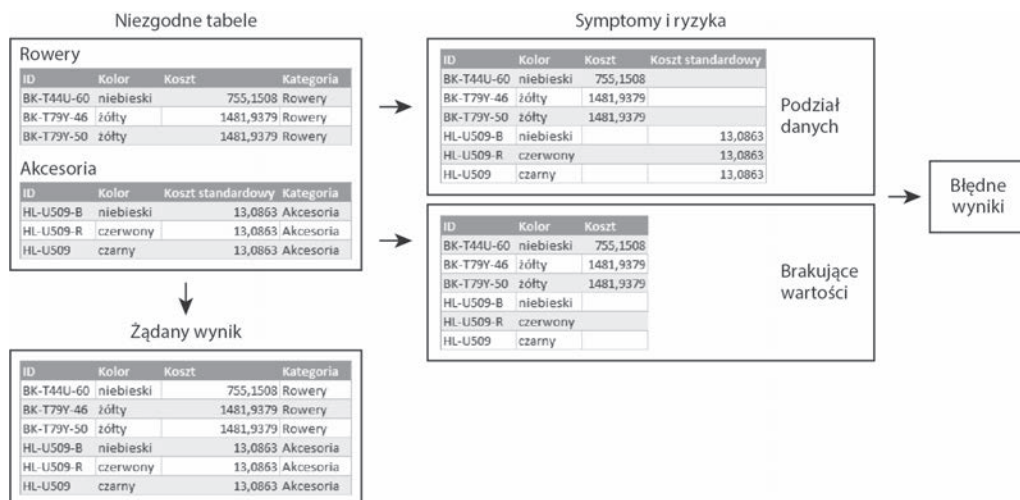
Teraz, kiedy wiesz już, czym są niezgodne tabele, możesz poznać symptomy niezgodności towarzyszące łączeniu tabel lub odświeżaniu raportu opartego na tabelach, które pierwotnie miały takie same formaty, ale które po pewnym czasie zmieniły się.

Podczas łączenia tabel wszystkie kolumny o takich samych nazwach są poprawnie łączone. Do tabeli wynikowej są kopiowane wszystkie wiersze z pierwszej kolumny. Ta sama zasada dotyczy wartości w drugiej kolumnie itd. Wszystkie wiersze kolumny są dołączane do tabeli wynikowej.

Jak jednak Power Query radzi sobie z niezgodnymi kolumnami? W większości przypadków tworzy tabelę zawierającą wszystkie kolumny tabel źródłowych. Każdy wiersz z pierwszej tabeli jest kopiowany do tabeli wynikowej. W takich wierszach wszystkie kolumny należące do drugiej tabeli zawierają puste komórki. W taki sam sposób są kopiowane wiersze drugiej tabeli — wszystkie kolumny właściwe dla pierwszej tabeli pozostają puste. Jest to tzw. symptom *podziału danych*, pokazany na rysunku 4.1. Power Query nie konsoliduje wartości zawartych w różnych kolumnach, tylko zachowuje oryginalne kolumny z nieciąglymi danymi.

Rysunek 4.1 ilustruje również inny symptom niezgodności tabel — brakujące wartości. Polega on na tym, że w tabeli wynikowej umieszczane są wszystkie kolumny pierwszej tabeli, a niezgodne kolumny z drugiej są beztrząsco pomijane. W efekcie wiele wierszy w wynikowej tabeli zawiera puste komórki. Dalej w tym rozdziale poznasz dokładniej powyższe symptomy i sposoby radzenia sobie z nimi.

Jeżeli zlekceważysz opisane symptomy, pojawią się oczywiste ryzyka. Podzielone i brakujące dane mogą być przyczyną błędów i niedokładności w raportach. Na szczęście wiesz, jak objawiają się symptomy niezgodności tabel i rozumiesz ich wagę. Czas więc, abyś dowiedział się, jak można rozwiązywać tego rodzaju problemy i pomyślnie łączyć tabele.



**RYSUNEK 4.1.** Są dwa symptomy niezgodności tabel: podzielone dane i brakujące wartości; oba mogą skutkować błędnymi raportami

## Ćwiczenie 4.1. Uzgadnianie kolumn: podejście reaktywne

W tym ćwiczeniu nauczysz się uzgadniać nazwy kolumn i unikać symptomów podziału i braku danych opisanych wcześniej w tym rozdziale (patrz rysunek 4.1). W tym ćwiczeniu, demonstrującym problem i jego rozwiązanie, wykorzystasz skoroszyt z danymi firmy Fabryka Przygód.

Przeanalizujemy najpierw symptom podziału danych na prostym przykładzie i zastanówmy się, jak można reaktywnie uzgodnić nazwy kolumn. W ćwiczeniu wykorzystasz dwa pliki:

- R04-01 - akcesoria.xlsx,
- R04-01 - rowery.xlsx.

Skoroszyty te zawierają katalogi produktów dwóch kategorii, takich jak *Akcesoria* i *Rowery*. Koszty produktów są zapisane w kolumnie *Koszt* tabeli *Rowery* oraz w kolumnie *Koszt standardowy* w tabeli *Akcesoria*. Spróbujmy teraz połączyć obie tabele.

1. Utwórz nowy skoroszyt w Excelu lub raport w Power BI Desktop i zaimportuj do niego oba powyższe pliki.
2. Zaznacz zapytanie *Akcesoria* i użyj przekształcenia *Dołącz zapytania jako nowe*.
3. Przyjrzyj się w zawartości nowego zapytania *Append1* w panelu podglądu. Zwróć uwagę, że zawiera ono zarówno kolumnę *Koszt*, jak i *Koszt standardowy*. W wierszach z rowerami pierwsza kolumna zawiera wartości liczbowe, a druga wartości *null*, natomiast w wierszach z akcesoriami pierwsza kolumna zawiera wartości *null*, a druga wartości liczbowe.



**Uwaga.** Zapewne zauważyłeś, że połączona tabela nie zawiera informacji o kategorii produktu, zawartej w nazwie pliku i arkusza. W rozdziale 5., „Zachowywanie kontekstu”, dowiesz się, jak zachowywać konteksty tabel.

#### 4. Zaznacz zapytanie *Akcesoria* i zmień nazwę kolumny *Koszt standardowy* na **Koszt**.

Gdy ponownie zaznaczysz zapytanie *Append1*, okaże się, że po uzgodnieniu nazw kolumn zniknął symptom podziału danych.

Opisane tu rozwiązanie jest proste i na pewno będziesz je często stosował do uzgadniania nazw kolumn. Czasami jednak nie będziesz mógł robić tego ręcznie. Załóżmy, że masz dwie tabele, każda zawierająca 100 kolumn. Ręczne zmienianie nazw w opisany tu sposób byłoby powtarzalnym i czasochłonnym zadaniem. Co gorsza, w większości przypadków podczas łączenia tabel zapisanych w folderze ręczna zmiana nazwy kolumn nie jest możliwa, o czym przekonasz się w następnym podrozdziale.

## Łączenie niezgodnych tabel zapisanych w folderze

Wcześniej w tym rozdziale nauczyłeś się ręcznie uzgadniać nazwy kolumn przed połączeniem tabel. Jeżeli w folderze zapisanych jest wiele tabel z niezgodnymi kolumnami, zadanie staje się jeszcze trudniejsze. Z jednej strony, ręczne uzgadnianie nie jest dobrą praktyką szczególnie wtedy, kiedy plików jest dużo. Z drugiej, przekształcenie polegające na połączeniu plików zapisanych w folderze nie daje możliwości indywidualnego traktowania każdego pliku.

Aby sprostać temu wyzwaniu, możesz napisać kod, który zmienia nazwy kolumn u źródła. Załóżmy jednak, że w folderze jest 60 skoroszytów i każdy zawiera tabelę składającą się z 20 kolumn. Należałoby sprawdzić 1200 kolumn pod kątem zgodności nazw i ręcznie je zmienić. To nie jest praktyczna metoda zwłaszcza wtedy, gdy nie jesteś autorem plików, tylko otrzymujesz je regularnie z zewnątrz.

W rozdziale 3. nauczyłeś się łączyć tabele zapisane w folderze. Jak już się przekonałeś, wykonując ćwiczenie opisane w tym rozdziale, Power Query wykorzystuje nagłówki pierwszej tabeli do utworzenia kolumn tabeli wynikowej. Podczas łączenia niezgodnych tabel zapisanych w folderze szybko pojawi się symptom braku danych. Następne ćwiczenie demonstruje ten symptom oraz sposób pomyślnego łączenia tabel zapisanych w folderze.

### Ćwiczenie 4.2. Część I. Symptom braku wartości

Wróćmy do naszej fikcyjnej firmy Fabryka Przygód. Menedżer każdej z czterech linii produktów prowadzi w skoroszytcie Excela własny katalog. Ty, jako szef działu analiz, dostałeś zadanie połączenia wszystkich tabel i przygotowania raportu. Jednak po otwarciu skoroszytów stwierdziłeś, że menedżerowie są kreatywni i zmienili wiele nazw kolumn (patrz rysunek 4.2). Przy próbie połączenia tabel zapisanych w folderze okazało się, że jest to niemożliwe. W tym ćwiczeniu dowiesz się, jak sobie radzić z takimi problemami.

Zawartość folderu	Niezgodne nazwy kolumn					
R04-02 - akcesoria.xlsx →	<table border="1"> <tr> <td>Produkt</td> <td>ID</td> <td>Kolor</td> <td>KosztStandardowy</td> <td>CenaListowa</td> </tr> </table>	Produkt	ID	Kolor	KosztStandardowy	CenaListowa
Produkt	ID	Kolor	KosztStandardowy	CenaListowa		
R04-02 - rowery.xlsx →	<table border="1"> <tr> <td>Nazwa</td> <td>Kod_produkту</td> <td>Kolor</td> <td>Koszt_standardowy</td> <td>Cena_listowa</td> </tr> </table>	Nazwa	Kod_produkту	Kolor	Koszt_standardowy	Cena_listowa
Nazwa	Kod_produkту	Kolor	Koszt_standardowy	Cena_listowa		
R04-02 - ubrania.xlsx →	<table border="1"> <tr> <td>Nazwa_produkту</td> <td>Nr_produkту</td> <td>Kolor_produkту</td> <td>Koszt_produkту</td> <td>Cena_produkту</td> </tr> </table>	Nazwa_produkту	Nr_produkту	Kolor_produkту	Koszt_produkту	Cena_produkту
Nazwa_produkту	Nr_produkту	Kolor_produkту	Koszt_produkту	Cena_produkту		
R04-02 - komponenty.xlsx →	<table border="1"> <tr> <td>Nazwa</td> <td>Kod_produkту</td> <td>Kolor</td> <td>Koszt_standardowy</td> <td>Cena_listowa</td> </tr> </table>	Nazwa	Kod_produkту	Kolor	Koszt_standardowy	Cena_listowa
Nazwa	Kod_produkту	Kolor	Koszt_standardowy	Cena_listowa		

**RYСУNEK 4.2.** Skoroszyty z produktami firmy Fabryka Przygód zawierają kolumny o niezgodnych nazwach

- Przejrzyj pliki zapisane w folderze C:\Dane\R04\R04-02 - produkty:
  - R04-02 - akcesoria.xlsx,
  - R04-02 - rowery.xlsx,
  - R04-02 - komponenty.xlsx,
  - R04-02 - ubrania.xlsx.
- Zwróć uwagę, że nazwy kolumn zbyt różnią się od siebie, aby można było poprawiać indywidualnie każdą z nich.
- Zaimportuj cztery niezgodne tabele z folderu, aby stwierdzić symptom brakujących wartości.
 

W **Excelu**: kliknij kartę *Dane*, a następnie polecenia *Pobierz dane/Z pliku/Z folderu*.

W **Power BI Desktop**: kliknij ikonę *Pobierz dane*, a następnie *Folder* i *Połącz*.
- W oknie *Folder* przejdź do folderu C:\Dane\R04\R04-02 - produkty i kliknij *OK*.
- W oknie, które się pojawi (o takiej samej nazwie jak nazwa folderu), kliknij przycisk *Połącz*, a następnie *Połącz i edytuj*.
- W oknie *Połącz pliki* wybierz w liście *Przykładowy plik* dowolny skoroszyt, zaznacz *Arkusz1* i kliknij *OK*.



**Uwaga.** W liście *Przykładowy plik* można wybrać skoroszyt i zastosować jego format do wszystkich plików w folderze. Efektem wyboru są m.in. nazwy kolumn w wynikowej tabeli. Ponieważ każda tabela zawiera kolumny o różnych nazwach, wkrótce zobaczysz symptom brakujących wartości i jego skutki.

- W oknie edytora przewiń zawartość podglądu, aż zawartość kolumny *Source.Name* zmieni się z *R04-02 - akcesoria.xlsx* na *R04-02 - komponenty.xlsx*. Kolumny *Produkt*, *ID*, *KosztStandardowy* i *CenaListowa* będą zawierały wartości *null*, tak jak na rysunku 4.3, które przełożą się na puste wartości w raporcie.

A <sup>B</sup> <sub>C</sub> Source.Name	A <sup>B</sup> <sub>C</sub> Produkt	A <sup>B</sup> <sub>C</sub> ID	A <sup>B</sup> <sub>C</sub> Kolor	1.2 KosztStandardowy	1.2 CenaListowa	1 <sup>2</sup> <sub>3</sub> Wielkość	1.2 Waga
R04-02 - akcesoria.xlsx	Lampa przednia podwojna	LT-H902		14,4334	34,99	null	null
R04-02 - akcesoria.xlsx	Lampa przednia wod				44,99	null	null
R04-02 - akcesoria.xlsx	Kask Sport-100, nieb				34,99	null	null
R04-02 - akcesoria.xlsx	Kask Sport-100, czern				34,99	null	null
R04-02 - akcesoria.xlsx	Kask Sport-100, czarny	HL-U509	czarny	13,0863	34,99	null	null
R04-02 - komponenty.xlsx		null	żółty	null	null	60	1397,06
R04-02 - komponenty.xlsx		null	żółty	null	null	62	1451,49
R04-02 - komponenty.xlsx		null	żółty	null	null	46	1342,63
R04-02 - komponenty.xlsx		null	żółty	null	null	50	1360,77
R04-02 - komponenty.xlsx		null	żółty	null	null	54	1378,91
R04-02 - komponenty.xlsx		null	niebieski	null	null	46	1342,63

**RYSUNEK 4.3.** Symptom brakujących wartości pojawia się podczas importowania z folderu wielu tabel z niezgodnymi nazwami kolumn

Zanim rozwiążesz problem brakujących wartości, otwórz panel *Zapytania* i zaznacz zapytanie *Przekształć przykładowy plik*, które w tej książce jest nazywane zapytaniem przykładowym. Zapytanie to zostało utworzone podczas łączenia plików zapisanych w folderze. Gdy je zaznaczysz, zauważysz, że w panelu podglądu dane akcesoriów są pokazywane poprawnie. Rozwiązywanie problemu brakujących wartości należy zawsze zaczynać od zapytania przykładowego.



**Wskazówka.** Power Query podczas importowania plików z folderu tworzy przykładową funkcję, przykładowe zapytanie, plik zapytania i parametr. Obiekty te są wykorzystywane do łączenia kilku plików na podstawie wybranych danych. Główne przekształcenie każdego pliku jest zaimplementowane w funkcji. Przekształcenia poszczególnych plików dostosowuje się poprzez wprowadzanie zmian w przykładowym zapytaniu. Zmiany te są uwzględniane w funkcji i wprowadzane we wszystkich plikach. Podczas łączenia plików zapisanych w folderze sprawdzaj, jakie zmiany wprowadza zapytanie *Przekształć przykładowy plik* we wszystkich plikach przed ich połączeniem.

## Ćwiczenie 4.2. Część II. Założenie takiej samej kolejności kolumn i ujednolicenie nagłówek

Aby rozwiązać problem brakujących wartości, przyjmijmy założenie, że kolumny — choć różnią się nagłówkami — są we wszystkich tabelach ułożone w takiej samej kolejności. W takim przypadku do połączenia tabel będzie można użyć domyślnych nazw kolumn *Column1*, *Column2* itd. Poniższe kroki należy wykonać po krokach od 1. do 6. z części I ćwiczenia 4.2.

8. Zaznacz zapytanie *Przekształć przykładowy plik* i zmień jego nazwę na *Przykładowe produkty*.
9. W panelu *Zastosowane kroki* usuń ostatni krok *Nagłówki o podwyższonym poziomie*. Kolumny uzyskają nazwy *Column1*, *Column2* itd. Jeżeli wszystkie kolumny we wszystkich tabelach będą miały takie same, ogólne nazwy, wtedy zostaną poprawnie połączone.
10. Kliknij zapytanie *R04-02 - produkty* i zwróć uwagę na komunikat *Expression.Error: Nie można znaleźć kolumny „Produkty” w tabeli*.
11. W panelu *Zastosowane kroki* usuń ostatni krok *Zmieniono typ*. Teraz połączona tabela nie będzie zawierała brakujących wartości. Możesz się o tym przekonać, przewijając jej zawartość. Wszystkie wartości z tabeli *Akcesoria* będą dostępne.
12. W karcie *Przekształć* kliknij ikonę *Użyj pierwszego wiersza jako nagłówek*, aby zmienić nazwy kolumn.

Niestety, połączona tabela zawiera również nagłówki poszczególnych tabel. W rozdziale 3. podobny problem rozwiązałeś za pomocą filtru. Tutaj jednak kolumny mają różne nazwy i trudno określić, które z nich należy zastosować w filtrze. Dlatego przed krokiem użyj indeksu w celu zidentyfikowania wierszy z nagłówkami.

13. W panelu *Zapytania* kliknij zapytanie *Przykładowe produkty*. Następnie w karcie *Dodaj kolumnę* kliknij ikonę *Kolumna indeksu*. Pojawi się kolumna zawierająca indeksy wierszy. Pierwszy wiersz każdej tabeli, zawierający nagłówki kolumn, ma indeks 0.
14. Wróć do zapytania *R04-02 - produkty* i usuń nagłówki za pomocą filtru odrzucającego indeks 0. W tym celu kliknij przycisk filtru w nagłówku ostatniej kolumny o nazwie 0. W panelu, który się pojawi, usuń zaznaczenie przy pozycji 0 i kliknij *OK*.
15. Usuń pierwszą i ostatnią kolumnę (o nazwach odpowiednio *R04-02 - akcesoria.xlsx* i 0). Aby zapytanie było uniwersalne, zmodyfikuj je w sposób opisany w rozdziale 3., „Łączenie danych z kilku źródeł”. W tym przypadku zastąp statyczne odwołanie do kolumny *Source.Name* dynamicznym wyrażeniem `Table.ColumnName(...){0}`.



W pasku formuły znajduje się następujący kod:

```
= Table.RemoveColumns("#Przefiltrowano wiersze",{"R04-02 - akcesoria.xlsx", "0"})
```

Zmień go na następujący:

```
= Table.RemoveColumns("#Przefiltrowano wiersze",{Table.ColumnNames("#Przefiltrowano
↳wiersze"){0}, "0"})
```

Teraz możesz załadować połączoną tabelę i rozpocząć analizę danych.

Gotowe rozwiązanie jest zapisane w plikach *R04-02 - rozwiązanie.xlsx* i *R04-02 - rozwiązanie.pbix*.

W kroku 8. zmieniłeś nagłówki kolumn na ogólne *Column1*, *Column2* itd. Mogłeś to zrobić, ponieważ przyjąłeś założenie, że kolumny we wszystkich tabelach są ułożone w takiej samej kolejności. Zastanówmy się jednak, co by się stało, gdyby w jednej z tabel kolejność kolumn była inna, np. w tabeli *Rowery* zamienione byłyby miejscami kolumny *Nazwa* i *Kod\_produkту*. Po zastosowaniu opisanej wyżej procedury w wynikowej tabeli powstałby bałagan. Nazwy produktów z tabeli *Rowery* zostałyby potraktowane jak kody produktów i odwrotnie. Aby temu zapobiec, mógłbyś się umówić z menedżerem działu rowerów, aby nie zmieniał kolejności kolumn w źródłowej tabeli. Jest jednak inne rozwiązanie, opisane dalej w tym rozdziale, polegające na ujednoczeniu nagłówków kolumn przed połączeniem tabel.

## Ćwiczenie 4.3. Proste ujednoczenie nagłówków za pomocą funkcji `Table.TransformColumnNames`

Automatyczne kojarzenie i zmienianie nazw kolumn w wielu tabelach jest ważną częścią procesu łączenia niezgodnych tabel. Jeżeli takich tabel jest dużo, wtedy ręczne zmienianie nazw wybranych kolumn, tak jak w ćwiczeniu 4.1, jest niepraktycznym rozwiązaniem szczególnie wtedy, gdy plików jest wiele i często się zmieniają. Jeżeli można przyjąć założenie, że kolumny we wszystkich tabelach są ułożone w tej samej kolejności, wtedy sprawdzi się rozwiązanie zaprezentowane w ćwiczeniu 4.2. Jeżeli jednak kolejność kolumn nie jest gwarantowana, wtedy trzeba zastosować bardziej zaawansowany sposób, jakim jest ujednoczenie nazw kolumn.

Ujednoczenie nazw kolumn jest to prosta operacja na tekstach. Zazwyczaj, aby uzyskać spójny format tabel, wystarczy wprowadzić niewielkie zmiany we wszystkich niezgodnych kolumnach, np. zastąpić znaki podkreślenia spacjami lub zmienić wszystkie litery na wielkie.

Prosta operacja na tekstach, np. zmiana wielkości liter, może być bardzo skutecznym sposobem uzgadniania nazw kolumn. Narzędzie Power Query rozróżnia wielkość liter, zatem np. kolumny o nazwach *kolumna1* i *Kolumna1* traktuje jako różne. Jeżeli tabele często się zmieniają, np. małe litery w nazwach kolumn są zamieniane na wielkie lub odwrotnie, wtedy ujednoczenie ich wielkości może znacznie zredukować niezgodności.

Aby w prosty sposób przekształcić nazwę kolumny, użyj funkcji `Table.TransformColumnNames`. Jeżeli przykładowo ostatni krok w panelu *Zastosowane kroki* nazywa się *Poprzedni krok*, wtedy poniższa formuła zamieni pierwsze litery każdego słowa w nazwie kolumny na wielkie:

```
= Table.TransformColumnNames("#Poprzedni krok", Text.Proper)
```

Natomiast poniższa formuła zamieni wszystkie znaki podkreślenia na spacje:

```
= Table.TransformColumnNames("#Poprzedni krok", each Replacer.ReplaceText(_, "_", " "))
```

W tym ćwiczeniu wykorzystasz powyższe formuły do połączenia tabel produktów firmy Fabryka Przygód.



**Uwaga.** Na tym etapie nie musisz w pełni rozumieć składni opisywanych formuł. Ten temat będzie opisany w rozdziale 9., „Wprowadzenie do języka M”. Na razie musisz jedynie wiedzieć, że funkcje `Text.Proper` i `Replacer.ReplaceText` służą do przekształcania tekstu. Obie można stosować wewnątrz funkcji `Table.TransformColumnNames`. W poprzednim przykładzie funkcja `Text.Proper` zamieniała pierwsze litery każdego słowa na wielkie, a `Replacer.ReplaceText` zamieniała wszystkie znaki podane w drugim argumentcie na znaki wskazane w trzecim argumentcie.

1. Przejrzyj pliki zapisane w folderze `C:\Dane\R04\R04-03 - produkty`:

- a. `R04-03 - akcesoria.xlsx`,
- b. `R04-03 - rowery.xlsx`,
- c. `R04-03 - komponenty.xlsx`,
- d. `R04-03 - ubrania.xlsx`.

Pliki te zawierają te same dane, co w ćwiczeniu 4.2, ale zostały w nich wprowadzone dwie zamierzone zmiany. Po pierwsze, inne są nagłówki kolumn. Różnice te można zniwelować, zamieniając pierwsze litery słów na wielkie oraz zastępując znaki podkreślenia spacjami. Po drugie, w tabeli *Rowery* zamienione są miejscami dwie pierwsze kolumny, przez co nie można ujednolicić ich nagłówków w sposób opisany w ćwiczeniu 4.2.

2. Utwórz nowy skoroszyt w Excelu lub raport w Power BI Desktop i wykonaj poniższe kroki.

W **Excelu**: kliknij kartę *Dane*, a następnie polecenia *Pobierz dane/Z pliku/Z folderu*.

W **Power BI Desktop**: kliknij ikonę *Pobierz dane*, a następnie *Folder* i *Połącz*.

3. W oknie *Folder* przejdź do folderu `C:\Dane\R04\R04-03 - produkty` i kliknij *OK*.
4. W oknie, które się pojawi (o takiej samej nazwie jak nazwa folderu), kliknij przycisk *Połącz*, a następnie *Połącz i edytuj*.
5. W oknie *Połącz pliki* wybierz w liście *Przykładowy plik* dowolny skoroszyt, zaznacz *Arkusze1* i kliknij *OK*.

Zgodnie z oczekiwaniami, w edytorze Power Query pojawił się symptom brakujących wartości.

6. Kliknij zapytanie *Przekształć przykładowy plik* i zmień jego nazwę na *Przykładowe produkty*.
7. W pasku formuły kliknij przycisk *fx* (patrz rysunek 4.4).
8. Zwróć uwagę, że w panelu *Zastosowane kroki* pojawił się nowy krok *Niestandardowe1*, a w pasku formuły następujący kod:

```
= #Nagłówki o podwyższonym poziomie"
```

Krok *Nagłówki o podwyższonym poziomie* był ostatni na liście, zanim kliknąłeś przycisk *fx*. Ciąg `#Nagłówki o podwyższonym poziomie` jest zmienną zawierającą wynik wykonania poprzedniego kroku. Ponieważ wynikiem tym jest tabela z niezgodnymi kolumnami, możesz zmienić wszystkie litery na małe, używając tej zmiennej z funkcjami `Table.TransformColumnNames` i `Text.Lower`. W tym celu wpisz następującą formułę:

```
= Table.TransformColumnNames(#Nagłówki o podwyższonym poziomie", Text.Lower)
```

ABC 123	Nazwa	ABC 123	Kod_produktu	ABC 123	Kolor	ABC 123	Koszt_standardowy	ABC 123	Cena_istotowa	ABC 123	Wielkość
1	Butelka na wodę 0,5 l	WB-H096					1,8653		4,99		nał
2	Uchwyt bidonu, górski	BC-M005					3,7983		9,99		nał
3	Uchwyt bidonu, szosowy	BC-R205					3,3623		8,99		nał
4	Zestaw tatek (8)	PK-7098					0,8565		2,29		nał
5	Uchwyt poczwórny	RA-H123					44,88		120		nał
6	Rozdzielacznik	CL-9009					2,9733		7,95		nał
7	Blotniki górskie	FE-6654					8,2205		21,98		nał
8	Stojak uniwersalny	ST-1401					59,466		159		nał
9	Napój izotoniczny	HY-1023-70			srebrny		20,5663		54,99	70	nał
10	Dętka góraska	TT-M928					1,8663		4,99		nał
11	Dętka szosowa	TT-R982					1,4923		3,99		nał
12	Dętka trekkingowa	TT-T092					1,8663		4,99		nał
13	Opona góraska LL	TI-M267					9,3463		24,99		nał
14	Opona góraska ML	TI-M602					11,2163		29,99		nał

**RYSUNEK 4.4.** Za pomocą przycisku *fx* można utworzyć niestandardowy krok i zdefiniować formułę nowego przekształcenia modyfikującego poprzedni krok

- Naciśnij *Enter*. Zwróć uwagę, że teraz nazwy wszystkich kolumn składają się wyłącznie z małych liter. Gdybyś chciał zamienić pierwsze litery wszystkich słów na wielkie, powinieneś użyć w powyższej formule funkcji `Text`. `Proper` zamiast `Text`. `Lower`.



**Uwaga.** Jeśli nawet pracujesz na tabelach, których kolumny mają zgodne nazwy i są ułożone we właściwej kolejności, powinieneś stosować powyższe formuły, aby zapobiec sytuacji, w której autor tabeli postanowi zmienić wielkość liter na małe lub wielkie albo ich kombinację.

- Aby zamienić znaki podkreślenia spacjami, kliknij ponownie w pasku formuły przycisk *fx*.
- Zwróć uwagę, że w panelu *Zastosowane kroki* pojawił się nowy krok *Niestandardowe2*, a w pasku formuły następujący kod:

= Niestandardowe1

Użyj zmiennej *Niestandardowe1* z funkcjami `Table.TransformColumnNames` i `Replacer.ReplaceText`, aby zamienić znaki podkreślenia na spacje. W tym celu wpisz następującą formułę:

= `Table.TransformColumnNames(Niestandardowe1, each Replacer.ReplaceText(_, "_", " "))`

- Naciśnij *Enter*. Zwróć uwagę, że w nazwach kolumn znajdują się spacje, a nie znaki podkreślenia. Teraz nagłówki są bardziej czytelne.
- Zaznacz zapytanie *R04-03 - produkty* i w panelu *Zastosowane kroki* usuń ostatni krok *Zmieniono typ*. Teraz, kiedy wszystkie kolumny zawierają w nagłówkach spacje zamiast znaków podkreślenia, wszystkie tabele zostały połączone poprawnie i zniknął symptom brakujących wartości.

Gotowe rozwiązanie jest zapisane w plikach *R04-03 - rozwiązanie.xlsx* i *R04-03 - rozwiązanie.pbix*.

## Tabela konwersji

W ćwiczeniu 4.3 zastosowałeś proste przekształcenie nazw niezgodnych kolumn i zastąpiłeś znaki podkreślenia spacjami. Niestety, w praktyce proste przekształcenia nie wystarczają.

Aby zademonstrować taki przypadek, wróćmy do naszej firmy Fabryka Przygód i danych użytych w ćwiczeniu 4.2. Tym razem dodatkowo będą zamienione miejscami dwie kolumny, tak jak w ćwiczeniu 4.3, zatem opisane techniki nie będą działać.

W opisanej niżej serii ćwiczeń wykorzystasz dane zapisane w folderze *C:\Dane\R04\R04-04 - produkty*.

Każdy z menedżerów inaczej nazwał kolumnę zawierającą identyfikator produktu: *ID*, *Kod produktu*, *Nr produktu* i *Kod produktu*. Nie możesz więc ujednolicić nagłówków za pomocą prostych przekształceń. Zamiast tego utworzysz tabelę konwersji zawierającą oryginalne i ujednolicone nazwy kolumn. Rysunek 4.5 przedstawia niezgodne kolumny i ich reprezentację w tabeli konwersji. Kolumna *Źródło* zawiera niezgodną nazwę, która będzie zastąpiona ujednoliconą nazwą w kolumnie *Cel*.



**RYСУNEK 4.5.** Tabela konwersji zawiera powiązane niezgodne i ujednolicone nazwy kolumn

Tworzenie tabeli konwersji może być czasochłonnym zajęciem. Tabela nie musi jednak zawierać wszystkich nazw. Przykładowo na rysunku 4.5 nie ma pary *Nazwa* → *Nazwa*, ponieważ nie jest potrzebna. Tabela powinna zawierać tylko pary niezgodnych nazw. Jeżeli są setki kolumn, wtedy w tabeli konwersji należy umieścić tylko różniące się nazwy.



**Wskazówka.** Jeżeli tabela konwersji jest tabelą zewnętrzną zapisaną w skoroszycie Excela, można najpierw umieścić w niej kilka niezgodnych nazw kolumn i stopniowo ją uzupełniać w miarę potrzeb. Po utworzeniu zapytania wykorzystującego tabelę konwersji nie trzeba go modyfikować w celu dalszego ujednolicania kolumn. Zamiast tego wystarczy odpowiednio zmodyfikować tabelę konwersji i odświeżyć raport.

Teraz, gdy wiesz już, jak tworzy się tabelę konwersji, nauczysz się ją wykorzystywać do ujednolicania niezgodnych kolumn. Użyjesz do tego celu przekształcenia *Scal zapytania* dostępnego w karcie *Strona główna*. Jak pamiętasz z rozdziału 2, „Podstawowe operacje przygotowywania danych”, przekształcenie to służy do łączenia kilku tabel na podstawie takich samych wartości zapisanych w określonych kolumnach.

Zanim dowiesz się, jak używać przekształcenia *Scal zapytania* z tabelą konwersji, zatrzymajmy się na chwilę, aby omówić jeden z największych problemów z ujednolicaniem kolumn. Pod koniec tego rozdziału dowiesz się, jak zastępować nazwy kolumn za pomocą funkcji *Table.TransformColumnNames*

opisanej w ćwiczeniu 4.3, funkcji niestandardowej oraz tabeli konwersji. Ponieważ technika ta wymaga głębszej znajomości języka M, wcześniej poznasz trzy sposoby wykorzystujące jedynie interfejs edytora, bez konieczności stosowania zaawansowanych formuł w języku M.

Interfejs edytora Power Query jest wyposażony w wiele przekształceń operujących na kolumnach, ale nie na ich nagłówkach. Ćwiczenia 4.4, 4.5 i 4.6 skupiają się na ujednoceniu nazw kolumn przy użyciu trzech technik wykorzystujących interfejs edytora. Wymagana jest w nich minimalna znajomość języka M. W tych ćwiczeniach nie jest wykorzystywana funkcja `Table.TransformColumnNames`, ponieważ wymagałaby ona utworzenia niestandardowej funkcji. Ćwiczenia te demonstrują, jak zamiast tego użyć funkcjonalności dostępnych w interfejsie edytora.



**Wskazówka.** Jeżeli poczułeś się pewnie w języku M, spodoba Ci się czwarta technika, czekająca na Ciebie pod koniec rozdziału. Prawdopodobnie uznasz, że ćwiczenie 4.7 jest najlepszym sposobem łączenia niezgodnych tabel i ujednociania kolumn. Warto jednak, abyś wykonał poniższe ćwiczenia, dzięki którym docenisz potęgę dostępnych funkcjonalności, jak również poznasz ich zalety i wady. Ponadto mogą się one przydać w innych sytuacjach.

## Ćwiczenie 4.4. Transpozycja z użyciem tabeli konwersji

Aby ujednoczyć niezgodne kolumny za pomocą tabeli konwersji, musisz najpierw tymczasowo przekształcić nazwy kolumn w jednokolumnową tabelę, scalić z nią tabelę konwersji, zastąpić niezgodne nazwy docelowymi i uzyskać wyniki z powrotem przekształcić w nagłówki kolumn. W tym ćwiczeniu poznasz pierwszą z wymienionych wyżej technik. Najpierw zaimportuj tabelę z folderu, wykonując tę samą sekwencję operacji, jak w ćwiczeniach 4.2 i 4.3.

1. Zaimportuj tabelę konwersji.

**W Excelu:** utwórz nowy skoroszyt, następnie kliknij kartę *Dane* i polecenia *Pobierz dane/ Z pliku/Ze skoroszytu*. Wskaż plik *R04-04 - tabela konwersji.xlsx*. W oknie *Nawigator* kliknij tabelę *Konwersja\_nagłówków*, a następnie przycisk *Przekształć dane*. Na koniec kliknij *Zamknij i załaduj do* oraz zaznacz opcję *Utwórz tylko połączenie*.

**W Power BI Desktop:** utwórz nowy raport, następnie rozwiń menu *Pobierz dane* i wybierz *Excel*. Wskaż plik *R04-04 - tabela konwersji.xlsx*. W oknie *Nawigator* kliknij tabelę *Konwersja\_nagłówków*, a następnie przycisk *Przekształć dane*. W panelu *Zapytania* kliknij prawym przyciskiem myszy nowe zapytanie i usuń zaznaczenie polecenia *Włącz ładowanie*. Na koniec kliknij *Zamknij i zastosuj*.

2. Zaimportuj następujące pliki znajdujące się w folderze *C:\Dane\R04\R04-04 - produkty*:

- a. *R04-04 - akcesoria.xlsx*,
- b. *R04-04 - rowery.xlsx*,
- c. *R04-04 - komponenty.xlsx*,
- d. *R04-04 - ubrania.xlsx*.

**W Excelu:** kliknij kartę *Dane*, a następnie polecenia *Pobierz dane/Z pliku/Z folderu*.

**W Power BI Desktop:** kliknij ikonę *Pobierz dane*, a następnie *Folder* i *Połącz*.

3. W oknie *Folder* przejdź do folderu *C:\Dane\R04\R04-04 - produkty* i kliknij *OK*.

4. W oknie, które się pojawi (o takiej samej nazwie jak nazwa folderu), kliknij przycisk *Połącz*, a następnie *Połącz i edytuj*.
5. W oknie *Połącz pliki* wybierz w liście *Przykładowy plik* dowolny skoroszyt, zaznacz *Arkusze1* i kliknij *OK*.
6. W panelu *Zapytania* kliknij zapytanie *Przekształć przykładowy plik* i zmień jego nazwę na *Przykładowe produkty*. Jak pamiętasz, jest to przykładowe zapytanie, a więc wprowadzone w nim zmiany będą dotyczyły również zapytania *R04-04 - produkty*.
7. Zmień nazwę zapytania *R04-04 - produkty* na *Dołączone produkty*. Zapisz skoroszyt lub raport, a następnie utwórz jego trzy kopie, które wykorzystasz w ćwiczeniach 4.5, 4.6 i 4.7.

## Transpozycja

Pierwsza technika ujednolicania nazw kolumn polega na użyciu przekształcenia *Transponuj* dostępnego w kolumnie *Przekształć*. W jego efekcie każda komórka kolumny X i wiersza Y jest zamieniana na kolumnę Y i wiersz X.

W edytorze Power Query o wiele łatwiej manipulować nazwami kolumn, jeżeli są one ułożone pionowo w jednej kolumnie. W tym momencie przydaje się przekształcenie *Transponuj*.

Jednak przy próbie transpozycji tabeli przekonasz się, że nie są zachowywane oryginalne nazwy kolumn. Aby tego uniknąć, musisz anulować użycie pierwszego wiersza tabeli jako nazw kolumn. W tym celu otwórz skoroszyt lub raport, który utworzyłeś w krokach od 1. do 7., i uruchom edytor Power Query. W panelu *Zapytania* zaznacz zapytanie *Przykładowe produkty* i wykonaj następujące kroki.

8. W panelu *Zastosowane kroki* usuń krok *Nagłówki o podwyższonym poziomie*. Nagłówki kolumn zostaną umieszczone w pierwszym wierszu tabeli.
9. W karcie *Przekształć* kliknij ikonę *Transponuj*. Teraz kolumna *Column1* zawiera nazwy kolumn, które trzeba ujednolicić.

## Sekwencja scaleń

Dotarłeś do punktu, w którym scalisz tabelę konwersji z przetransponowaną tabelą i za jednym razem zamienisz niezgodne nazwy kolumn na docelowe. Wykonaj następujące kroki.

10. Zaznacz zapytanie *Przykładowe produkty*, a następnie w karcie *Strona główna* kliknij ikonę *Scal zapytania*.
11. W oknie *Scalanie*, które się pojawi, kliknij w tabeli *Przykładowe produkty* kolumnę *Column1*, a następnie z rozwijanej listy pod tabelą wybierz *Konwersja\_nagłówków* (patrz rysunek 4.6). W tabeli niżej zaznacz kolumnę *Źródło*.
12. Sprawdź, czy w rozwijanej liście *Rodzaj sprzężenia* jest wybrana opcja *Lewe zewnętrzne (wszystkie z prawej, pasujące z drugiej)*, a następnie kliknij *OK*.
13. W ostatniej kolumnie, *Konwersja\_nagłówków*, zawierającej obiekty *Table*, kliknij przycisk po prawej stronie nagłówka lub zaznacz całą kolumnę i w karcie *Przekształć* kliknij ikonę *Rozwiń*.
14. Aby wyodrębnić kolumnę *Cel* z tabeli *Konwersja\_nagłówków*, usuń w panelu, który się pojawi, zaznaczenie *Źródło* i kliknij *OK*. Pojawi się nowa kolumna o nazwie *Konwersja\_nagłówków.Cel*. Kolumna ta zawiera wartość z kolumny *Cel*, jeżeli wartość w kolumnie *Column1* jest taka sama jak w kolumnie *Źródło* tabeli konwersji, lub jest to wartość *null* w przeciwnym przypadku. W tym drugim przypadku zostaną użyte oryginalne nazwy kolumn.

Scalanie

Wybierz tabelę i pasujące kolumny, aby utworzyć scaloną tabelę.

Przykładowe produkty

Column1	Column2	Column3	Column4	Column5	Column6
Produkt	Butelka na wodę 0,5 l	Uchwyt bidonu, górski	Uchwyt bidonu, szosowy	Zestaw łałek (8)	Uchwyt poczwórny
ID	WB-H098	BC-M005	BC-R205	PK-7098	RA-H123
Kolor					
KosztStandardowy	1,8663	3,7363	3,3623	0,8565	44,8
CenaListowa	4,99	9,99	8,99	2,29	12

Konwersja\_nagłówków

Źródło	Cel
Produkt	Nazwa
Nazwa_produktu	Nazwa
ID	Kod produktu
Kod_produktu	Kod produktu
Nr_produktu	Kod produktu

Rodzaj sprzężenia

Lewe zewnętrzne (wszystkie z pierwszej, pasujące z dr...)

Użyj dopasowywania rozmytego w celu wykonania scalenia

▸ Opcje scalania rozmytego

Zaznaczenie jest zgodne z 4 z 9 wierszy z pierwszej tabeli.

OK Anuluj

**RYSUNEK 4.6.** Tabelę konwersji i przekształcenie „Scal zapytania” można wykorzystać do ujednoczenia niezgodnych nazw kolumn

W następnym kroku wykorzystasz kolumnę warunkową zawierającą docelowe nazwy z kolumny *Konwersja\_nagłówków.Cel* lub oryginalne nazwy, jeżeli nie będą wymienione w tabeli konwersji.

15. W karcie *Dodaj kolumnę* kliknij ikonę *Kolumna warunkowa*. Pojawi się okno *Dodawanie kolumny warunkowej*. Wykonaj poniższe kroki, aby ustawić opcje pokazane na rysunku 4.7.
  - a. W polu *Nazwa nowej kolumny* wpisz *Nowe nazwy kolumn*.
  - b. W rozwijanej liście *Nazwa kolumny* wybierz *Konwersja\_nagłówków.Cel*.
  - c. W polu *Wartość* wpisz *null*.
  - d. W lewej rozwijanej liście pod etykietą *Wartość wyjściowa* wskaż *Wybierz kolumnę*.
  - e. W prawej rozwijanej liście pod etykietą *Wartość wyjściowa* wybierz *Column1*.
  - f. W lewej rozwijanej liście pod etykietą *W przeciwnym razie* wskaż *Wybierz kolumnę*.
  - g. W prawej rozwijanej liście pod etykietą *W przeciwnym razie* wybierz *Konwersja\_nagłówków.Cel* i kliknij *OK*.

Dodawanie kolumny warunkowej

Dodaj kolumnę warunkową obliczaną na podstawie innych kolumn lub wartości.

Nazwa nowej kolumny

Nazwa kolumny	Operator	Wartość	Wartość wyjściowa
Jeśli	Konwersja_nagłó...	równa się	ABC 123
			null
			To
			Column1

W przeciwnym razie

**RYSUNEK 4.7.** W oknie „Dodawanie kolumny warunkowej” można utworzyć nową kolumnę zawierającą wartości pochodzące z różnych kolumn w zależności od spełnienia zadanych warunków. W tym przypadku nowa kolumna będzie zawierała nazwy ujednolicone lub oryginalne, jeżeli nie zostaną znalezione w tabeli konwersji

16. Usuń kolumny *Column1* i *Konwersja\_nagłówków.Cel*.
17. Przenieś kolumnę *Nazwy nowych kolumn* na początek tabeli, przeciągając ją po prostu z prawej strony na lewą lub stosując poniższą wskazówkę.



**Wskazówka.** Jeżeli kolumn jest dużo i musisz jedną z nich przenieść na początek tabeli, wtedy zamiast przeciągać, zaznacz ją, a następnie kliknij w karcie *Przekształć* ikonę *Przenieś i Na początek*. Ewentualnie kliknij prawym przyciskiem myszy nagłówek kolumny i z podręcznego menu wybierz polecenie *Przenieś i Na początek*.



**Uwaga.** Zmianianie kolejności kolumn tabeli jest dość ryzykowną operacją. W rozdziale 10., „Od złych do dobrych zapytań”, dowiesz się, jak unikać związanych z tą operacją problemów. Podczas przenoszenia kolumny formuła odwołująca się do kolumn tworzy podatne za zakłócenia zapytanie, które może uniemożliwić odświeżenie raportu, jeżeli jakiejś kolumny zabraknie lub zostanie zmieniona jej nazwa.

## Powtórna transpozycja

W kroku 9. przetransponowałeś tabelę i umieściłeś nagłówki w kolumnie *Column1*. Przy użyciu transpozycji mogłeś skonsolidować oryginalne nagłówki w kolumnie *Column1* z nazwami w kolumnie *Źródło* w tabeli konwersji. Następnie za pomocą kolumny warunkowej zamieniłeś niezgodne nagłówki na nazwy z kolumny *Cel* tabeli konwersji. Teraz kolumna *Column1* zawiera ujednolicone wartości, których można użyć jako nazw nagłówków. Czas z powrotem przetransponować tabelę i odtworzyć nagłówki z kolumny *Column1*. Wykonaj poniższe kroki.

18. W karcie *Przekształć* kliknij ikonę *Transponuj*. Nazwy kolumn znajdują się w pierwszym wierszu tabeli, dlatego musisz je przenieść z powrotem do nagłówków.
19. W karcie *Przekształć* kliknij ikonę *Użyj pierwszego wiersza jako nagłówków*.
20. W panelu *Zapytania* zaznacz zapytanie *Dołączone produkty*. Jeżeli w panelu podglądu pojawi się błąd, usuń z panelu *Zastosowane kroki* krok *Zmieniono typ*. Zadanie wykonane.



Gotowe rozwiązanie znajduje się w plikach *R04-04 - rozwiązanie - transpozycja.xlsx* i *R04-04 - rozwiązanie - transpozycja.pbix*.

Niestety, opisana technika jest skuteczna tylko w przypadku niewielkich ilości danych. Duże tabele, złożone z bardzo wielu wierszy zajmują mnóstwo pamięci. Power Query obsługuje tabele zawierające maks. 16 384 kolumn. Przy próbie transpozycji tabeli złożonej z więcej niż 16 384 wierszy pojawi się komunikat o błędzie, dane będą importowane przez długi czas i zajmą dużo pamięci. Jednak nawet w przypadku tabeli złożonej np. z 15 000 wierszy należy z powodu zajętości pamięci i długotrwałego odświeżania danych zastosować inną technikę. Potrzebne jest rozwiązanie, które nie zawiedzie nawet przy dużych tabelach.

## Ćwiczenie 4.5. Anulowanie przestawienia, scalenie i powtórne przestawienie

Lepszym sposobem umieszczenia nazw kolumn w nowej kolumnie i przygotowania jej do scalenia z tabelą konwersji jest użycie przekształcenia *Anuluj przestawienie*. W tym ćwiczeniu dowiesz się, jak je stosować i jak przywrócić ujednolicone nazwy za pomocą przekształcenia *Tabela przestawna*.

**Zobacz też.** W rozdziale 6., „Rozdzielanie tabel”, opisano wiele różnych zastosowań przekształcenia *Anuluj przestawienie*, natomiast rozdział 7., „Zaawansowane metody przestawiania tabel i anulowania przestawień”, zawiera praktyczne przykłady użycia przekształcenia *Tabela przestawna*.

W tym ćwiczeniu wykorzystasz plik, który utworzyłeś w ćwiczeniu 4.4, w krokach od 1. do 7.

1. Otwórz kopię skoroszytu lub raportu, którą utworzyłeś w ćwiczeniu 4.4, w kroku 7., a następnie uruchom edytor Power Query.
2. W panelu *Zapytania* zaznacz zapytanie *Przykładowe produkty*, a następnie w karcie *Dodaj kolumnę* kliknij ikonę *Kolumna indeksu*. W następnym kroku dowiesz się, do czego ta kolumna będzie potrzebna.
3. Zaznacz kolumnę *Indeks*, kliknij prawym przyciskiem myszy jej nagłówek i wybierz polecenie *Anuluj przestawienie innych kolumn*. Powstanie nowa tabela zawierająca kolumny *Indeks*, *Atrybut* i *Wartość*. Kolumna *Indeks* zawiera identyfikator wiersza, który w kroku 6. wykorzystasz do przywrócenia tabeli do pierwotnej postaci. Kolumna *Atrybut* zawiera nazwy kolumn, a kolumna *Wartość* oryginalne wartości. Teraz możesz w kolumnie *Atrybut* zastosować funkcje przekształcające tekst.
4. Zmień nazwę kolumny *Atrybut* na **Column1**, by wykonać następane kroki.
5. Aby zmienić nazwy niezgodnych kolumn, wykonaj sekwencję scaleń opisaną w ćwiczeniu 4.4, w krokach od 10. do 17.
6. Aby z powrotem przestawić tabelę, zaznacz kolumnę *Nowe nazwy kolumn* i w karcie *Przekształć* kliknij ikonę *Kolumna przestawna*.

**Zobacz też.** Przekształcenie *Kolumna przestawna* jest bardzo przydatne. W rozdziale 7. dowiesz się, jak można je wykorzystywać do przetwarzania tabel o nieuporządkowanej strukturze.

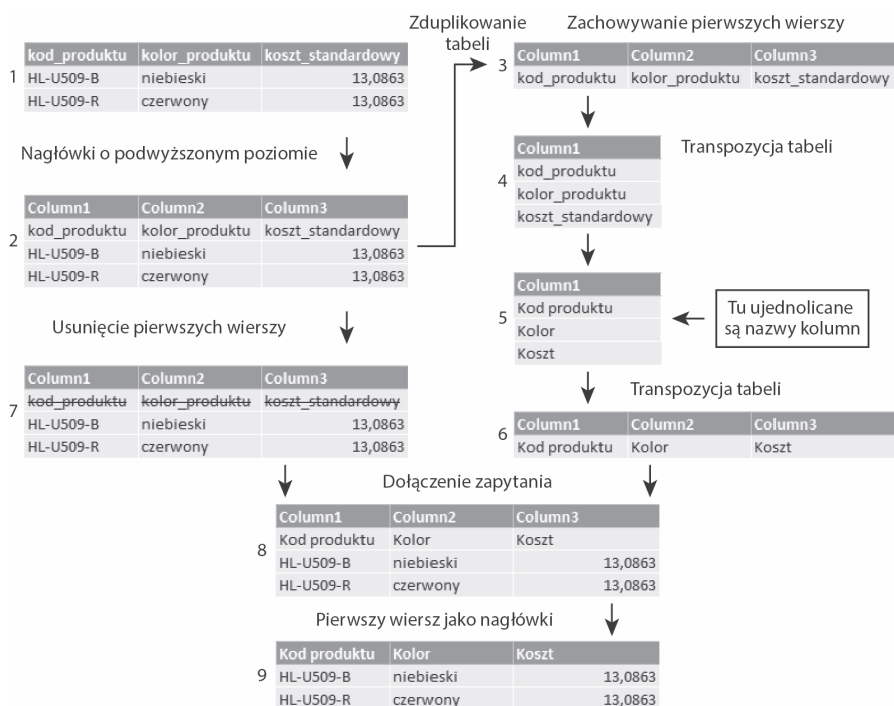
7. W oknie *Kolumna przestawna* wybierz w rozwijanej liście *Kolumna wartości* pozycję *Wartość*, rozwiń sekcję *Opcje zaawansowane* i w rozwijanej liście *Agreguj funkcję wartości* wybierz pozycję *Nie agreguj*. Kliknij OK.
8. Usuń kolumnę *Indeks*.
9. W panelu *Zapytania* zaznacz zapytanie *Dołączone produkty*. Jeżeli pojawi się komunikat o błędzie, usuń w panelu *Zastosowane kroki* ostatni krok *Zmieniono typ*.

Gotowe rozwiązanie znajduje się w plikach *R04-05 - rozwiązanie - anulowanie przestawienia.xlsx* i *R04-05 - rozwiązanie - anulowanie przestawienia.pbix*.

Przekształcenie *Anulowanie przestawienia* można wprawdzie stosować w przypadku dużych ilości danych, ale przekształcenie *Kolumna przestawna*, przywracające tabelę do pierwotnej postaci, może zajmować sporo pamięci i spowalniać odświeżanie raportu. Przyczyną jest konieczność przeglądania przez Power Query całego zbioru danych, wyszukiwanie w kolumnie *Column1* unikatowych wartości, a następnie tworzenie z nich nowej kolumny. Dlatego nie jest to idealna technika, jeżeli danych jest dużo, tabele trzeba łączyć sprawnie, a raporty muszą się odświeżać szybko.

## Ćwiczenie 4.6. Transpozycja samych nazw kolumn

Wydajną metodą przekształcania nazw kolumn dużej tabeli w nową kolumnę i przygotowywania jej do skonsolidowania z tabelą konwersji jest transpozycja samych nazw kolumn. Jest to pośrednie przekształcenie, po którym następuje scalenie zapytań. Rysunek 4.8 przedstawia proces transpozycji nazw kolumn, ujednolicenia nazw i powtórnej transpozycji przywracającej tabeli oryginalną postać.



**RYСУNEK 4.8.** Ujednolicenie nazw kolumn wymaga wykonania dziewięciu przekształceń

Technika pokazana na rysunku 4.8 jest prosta w implementacji, jeżeli trzeba połączyć kilka tabel za pomocą przekształcenia *Dołącz zapytania*. Jednak komplikuje się w przypadku importu plików z folderu, ponieważ wymaga zastosowania niestandardowych funkcji. Tutaj została pokazana jedynie w celach demonstracyjnych. Możesz ją stosować wtedy, gdy z powodów wydajnościowych nie będziesz mógł wykorzystać technik opisanych w ćwiczeniach 4.4 i 4.5 lub nie chcesz używać języka M wykorzystanego w ćwiczeniu 4.7.

1. Otwórz kopię skoroszytu lub raportu, którą utworzyłeś w ćwiczeniu 4.4, w krokach od 1. do 7., a następnie uruchom edytor Power Query.
2. W panelu *Zapytania* zaznacz zapytanie *Przykładowe produkty*, a następnie w panelu *Zastosowane kroki* usuń krok *Nagłówki o podwyższonym poziomie*. W ten sposób nazwy kolumn zostaną umieszczone w pierwszym wierszu tabeli.
3. W panelu *Zapytania* kliknij prawym przyciskiem myszy zapytanie *Przykładowe produkty*, a następnie wybierz polecenie *Duplikuj*. Zaznacz zapytanie *Przykładowe produkty (2)*, w kolumnie *Strona główna* kliknij ikonę *Zachowaj wiersze* i polecenie *Zachowywanie pierwszych wierszy*. W oknie, które się pojawi, wpisz w polu *Liczba wierszy* liczbę 1 i kliknij OK.



**Uwaga.** Power Query podczas łączenia plików zapisanych w folderze tworzy funkcję zapytania, przykładowe zapytanie, zapytanie pliku i parametr. Obiekty te są wykorzystywane do łączenia kilku plików na podstawie wskazanego pliku. Podstawowe przekształcenie każdego pliku jest zaimplementowane w funkcji. W celu dostosowania przekształcenia indywidualnie do każdego pliku wprowadza się zmiany w przykładowym zapytaniu. W tym ćwiczeniu takim zapytaniem jest *Przekształć przykładowy plik*, które przemianowałeś na *Przykładowe produkty*. Zmiany wprowadzane w przykładowym zapytaniu są przenoszone do funkcji i w efekcie stosowane we wszystkich plikach w folderze. Jeżeli użyjesz przekształcenia *Duplikuj* lub *Odwołanie* w przykładowym zapytaniu, wtedy przekształcenia zastosowane w nowym zapytaniu nie będą przenoszone do funkcji, chyba że zastosujesz zaawansowaną technikę wiążącą nowe zapytanie z sekwencją przekształceń w przykładowym zapytaniu. Takie wiązanie wykonasz w krokach 21. i 22.

4. W karcie *Przekształć* kliknij ikonę *Transponuj*, aby przygotować tabelę do ujednoclenia. W tym miejscu mógłbyś wykonać sekwencję scaleń podobną do opisanej w krokach od 10. do 17. ćwiczenia 4.4. Jednak tutaj nie możesz tego zrobić, ponieważ musisz wprowadzić kilka ważnych modyfikacji.
5. Zaznacz zapytanie *Przykładowe produkty (2)* i w karcie *Strona główna* kliknij ikonę *Scal zapytania*.
6. W oknie *Scalanie* zaznacz w tabeli *Przykładowe produkty (2)* kolumnę *Column1*, a w rozwijanej liście niżej wybierz pozycję *Konwersja\_nagłówków*. Zaznacz kolumnę *Źródło*, tak jak w ćwiczeniu 4.4 (patrz rysunek 4.6).
7. Sprawdź, czy w rozwijanej liście *Rodzaj sprzężenia* wybrana jest opcja *Lewe zewnętrzne (wszystkie z prawej, pasujące z drugiej)*, a następnie kliknij OK. Pojawi się nowa kolumna o nazwie *Konwersja\_nagłówków* zawierająca obiekty *Table*.
8. Teraz musisz utworzyć kolumnę indeksu, dzięki której w następnym kroku nie zmieni się kolejność nazw kolumn w kolumnie *Column1*. Kliknij w karcie *Dodaj kolumnę* ikonę *Kolumna indeksu*. W kroku 13. dowiesz się, dlaczego jest to ważne.

9. Zaznacz kolumnę *Konwersja\_nagłówków*, a następnie w karcie *Przekształć* kliknij ikonę *Rozwiń*.
10. W oknie *Rozwiń Konwersja\_nagłówków* usuń zaznaczenie *Źródło* i kliknij *OK*. Nowa kolumna zostanie przekształcona w *Konwersja\_nagłówków.Cel* zawierającą wartość z kolumny *Cel*, jeżeli wartość w kolumnie *Column1* będzie taka sama jak w kolumnie *Źródło* tabeli konwersji, lub *null* w przeciwnym przypadku.
11. W kolumnie *Dodaj kolumnę* kliknij *Kolumna warunkowa*.
12. W oknie *Dodawanie kolumny warunkowej* wykonaj następujące kroki.
  - a. W polu *Nazwa nowej kolumny* wpisz **Nowe nazwy kolumn**.
  - b. W rozwijanej liście *Nazwa kolumny* wybierz *Konwersja\_nagłówków.Cel*.
  - c. W polu *Wartość* wpisz **null**.
  - d. W lewej rozwijanej liście pod etykietą *Wartość wyjściowa* wskaż *Wybierz kolumnę*.
  - e. W prawej rozwijanej liście pod etykietą *Wartość wyjściowa* wybierz *Column1*.
  - f. W lewej rozwijanej liście pod etykietą *W przeciwnym razie* wskaż *Wybierz kolumnę*.
  - g. W prawej rozwijanej liście pod etykietą *W przeciwnym razie* wybierz *Konwersja\_nagłówków.Cel* i kliknij *OK*.
13. Sprawdź, czy wartości w kolumnie *Indeks* są ułożone w kolejności rosnącej. Jeżeli nie, posortuj je, ponieważ kolejność nazw kolumn jest tutaj bardzo ważna. W kroku 9. rozszerzyłeś tabelę, a w tym kroku kolejność wierszy mogła zostać zaburzona. Dlatego do zachowania właściwej kolejności nazw jest wykorzystywana kolumna *Indeks*, którą utworzyłeś w kroku 8.
14. Usuń kolumny *Column1*, *Konwersja\_nagłówków.Cel* i *Indeks*. Zachowaj tylko ostatnią kolumnę, czyli *Nowe nazwy kolumn*.
15. W kolumnie *Przekształć* kliknij ikonę *Transponuj*. W ten sposób uzyskasz ujednoczone nazwy kolumn.  
Zakończyłeś ujednoczanie nazw kolumn w zapytaniu *Przykładowe produkty (2)*, oznaczone numerem 5 na rysunku 4.4. Teraz możesz wykonać pozostałą sekwencję przekształceń.
16. Aby usunąć stare nagłówki, wróć do oryginalnego zapytania *Przykładowe produkty*. Zaznacz je, a następnie w karcie *Strona główna* kliknij ikonę *Usuń wiersze* i polecenie *Usuwanie pierwszych wierszy*. W oknie, które się pojawi, wpisz w polu *Liczba wierszy* liczbę **1** i kliknij *OK*.
17. W karcie *Strona główna* kliknij ikonę *Dołącz zapytania*. W oknie *Dołączanie*, które się pojawi, wybierz w rozwijanej liście *Tabela do dołączania* pozycję *Przykładowe produkty (2)* i kliknij *OK*.
18. Zwróć uwagę, że tym razem w panelu podglądu ujednoczone nazwy kolumn zawarte w tabeli *Przykładowe produkty (2)* pojawiły w ostatnim wierszu. Dlatego musisz table zamienić miejscami. W pasku formuły znajduje się następujący kod:

```
= Table.Combine({#"Usunięto pierwsze wiersze", #"Przykładowe produkty (2)"})
```

Zamień w powyższej formule miejscami zmienne `#"Usunięto pierwsze wiersze"` i `#"Przykładowe produkty (2)"`, tak jak niżej:

```
= Table.Combine({#"Przykładowe produkty (2)", #"Usunięto pierwsze wiersze"})
```

19. Teraz, gdy ujednolicone nazwy kolumn znalazły się w pierwszym wierszu, możesz je przenieść do nagłówek. W tym celu kliknij w karcie *Przekształć* ikonę *Użyj pierwszego wiersza jako nagłówek*, a w panelu *Zastosowane kroki* usuń wiersz *Zmieniono typ*.



**Uwaga.** Usunięcie kroku *Zmieniono typ* jest bardzo ważne. Ponieważ kolumna *Wielkość* w tabeli *Akcesoria* zawiera wyłącznie liczby, edytor Power Query ustala jej typ na *Liczba całkowita*. Jednak w innych tabelach, np. *Rowery*, w tej kolumnie znajdują się poprawne wartości tekstowe. Jeżeli nie usuniesz kroku zmieniającego typ kolumn, pojawią się komunikaty o błędach, a kolumna *Wielkość* będzie zawierała tylko liczby.

20. W panelu *Zapytania* zaznacz zapytanie *Dołączone produkty*. Błąd, który się pojawi, możesz poprawić, usuwając z panelu *Zastosowane kroki* krok *Zmieniono typ*.

Teraz, gdy przewiniesz tabelę do momentu, aż w kolumnie *Source.Name* pojawi się wartość *R04-04 - rowery.xlsx*, zauważysz, że wartości w kolumnach *Nazwa* i *Kod produktu* są zamienione miejscami (patrz rysunek 4.9). Zatem opisane ujednolicenie nie działa poprawnie.

	Source.Name	Nazwa	Kod produktu	Kol...
156	R04-04 - komponenty.xlsx	Rama szosowa LL - czerw...	FR-R38R-44	czerwony
157	R04-04 - komponenty.xlsx	Rama szosowa LL - czerw...	FR-R38R-48	czerwony
158	R04-04 - komponenty.xlsx	Rama szosowa LL - czerw...	FR-R38R-52	czerwony
159	R04-04 - komponenty.xlsx	Rama Nazwa erw...	FR-R3 Kod	czerwony
160	R04-04 - komponenty.xlsx	Rama produktu erw...	FR-R3 produktu	czerwony
161	R04-04 - komponenty.xlsx	Rama szosowa LL - czerw...	FR-R38R-62	czerwony
162	R04-04 - komponenty.xlsx	Rama szosowa HL - czarna...	FR-R92B-58	czarny
163	R04-04 - komponenty.xlsx	Rama szosowa HL - czerw...	FR-R92R-58	czerwony
164	R04-04 - rowery.xlsx	BK-T44U-60	Rower trekkingowy-2000 ...	niebieski
165	R04-04 - rowery.xlsx	BK-T79Y-46	Rower trekkingowy-1000 ...	żółty
166	R04-04 - rowery.xlsx	BK-T79Y-50	Rower trekkingowy-1000 ...	żółty
167	R04-04 - rowery.xlsx	BK-T7 Kod produktu	Rowe Nazwa 000 ...	żółty
168	R04-04 - rowery.xlsx	BK-T7	Rowe produktu 000 ...	żółty
169	R04-04 - rowery.xlsx	BK-T18U-54	Rower trekkingowy-3000 ...	niebieski
170	R04-04 - rowery.xlsx	BK-T18U-58	Rower trekkingowy-3000 ...	niebieski
171	R04-04 - rowery.xlsx	BK-T18U-62	Rower trekkingowy-3000 ...	niebieski
172	R04-04 - rowery.xlsx	BK-T18Y-44	Rower trekkingowy-3000 ...	żółty

**RYСУNEK 4.9.** W zapytaniu *Dołączone produkty* wartości *Nazwa* i *Kod produktu* pobrane z tabeli *Rowery* są zamienione miejscami



**Wskazówka.** Sprawdzenie końcowego efektu jest ważną czynnością w Power Query. Zawartość panelu podglądu może być bardzo myląca, jak w tym przykładzie, ponieważ stanowi ona tylko część danych. Tutaj wszystko działało jak należy dla tabeli *Akcesoria*, ale po bliższym przyjrzeniu się połączonej tabeli okazało się, że zawartość kolumn *Nazwa* i *Kod produktu* pobrane z tabeli *Rowery* są zamienione miejscami. Dlatego zawsze dokładnie sprawdzaj wynikowe dane.

Zatrzymajmy się na chwilę, aby zrozumieć, co się stało i dlaczego ujednolicenie nie działa poprawnie. Dobra wiadomość jest taka, że zapytanie *Przykładowe produkty (2)* działa poprawnie dla pojedynczego pliku. Gdy je klikniesz, zauważysz, że w ostatnim kroku w panelu *Zastosowane kroki* nagłówki kolumn zostały ujednolicone zgodnie z oczekiwaniami. Zła jest jednak wiadomość, że ujednolicenie obejmuje tylko jeden plik. Gdy przykładowe zapytanie zostało zastosowane w przykładowej funkcji w odniesieniu do wszystkich plików w folderze, bezkrytycznie przekształciło wszystkie nagłówki we wszystkich plikach.

Dlaczego wykorzystywany jest ten sam pojedynczy plik? Spójrz na zapytanie *Przykładowe produkty (2)*. Gdy w panelu *Zastosowane kroki* zaznaczysz krok *Źródło*, wtedy w panelu formuły pojawi się kod ładujący arkusz Excela wskazany w parametrze *Parametr1*. Parametr ten został utworzony przez Power Query podczas łączenia plików zapisanych w folderze. Gdy klikniesz ten parametr w panelu *Zapytania*, zobaczysz, że ma on wartość *Przykładowy plik*. Zapytanie o tej nazwie również znajduje się w panelu *Zapytania* i zwraca zawartość skoroszytu *R04-04 - akcesoria.xlsx*.

Dlaczego więc zostały zamienione miejscami wartości kolumn *Nazwa* i *Kod produktu* z tabeli *Rowery*? Ponieważ w skoroszytcie *R04-04 - akcesoria.xlsx* powyższe kolumny są ułożone w innej kolejności niż w *R04-04 - rowery.xlsx*. Nie można zatem zastosować ujednolicenia przystosowanego do tabeli *Akcesoria* w odniesieniu do tabel, w których kolejność kolumn jest inna.

Aby poprawić ten błąd, musisz się dowiedzieć, jak stosować zapytanie *Przykładowe produkty (2)* indywidualnie dla każdego pliku w folderze. Jak pamiętasz, w kroku 3. tego ćwiczenia zapytanie *Przykładowe produkty (2)* utworzyłeś poprzez zduplikowanie zapytania *Przykładowe produkty*. Jest to przykładowe zapytanie wykonywane dla każdego pliku, wywołujące zapytanie *Przykładowe produkty (2)*. Ponieważ jednak *Przykładowe produkty (2)* nie jest już przykładowym zapytaniem, jest wykonywane tylko dla pliku *R04-04 - akcesoria.xlsx*.

Potrzebny jest więc sposób przekształcenia zapytania *Przykładowe produkty (2)* tak, aby wykorzystywało parametr z nazwą pliku z zapytania *Przykładowe produkty* i ujednolicało nagłówki we wszystkich plikach. W tym celu musisz przekształcić zapytanie *Przykładowe produkty (2)* w funkcję (zwaną również funkcją niestandardową), która będzie wywoływana podczas przetwarzania każdego pliku zapisanego w folderze. Wykonaj poniższe kroki, aby utworzyć taką funkcję na podstawie zapytania *Przykładowe produkty (2)*.

21. W panelu *Zapytania* kliknij prawym przyciskiem myszy zapytanie *Przykładowe produkty (2)* i wybierz polecenie *Utwórz funkcję*. Pojawi się okno o tej samej nazwie. W polu *Nazwa funkcji* wpisz **FnUjednolicenieNazwKolumn** i kliknij OK. Jak pokazuje rysunek 4.10, zostanie utworzona funkcja, która będzie wywoływana dla wszystkich plików.

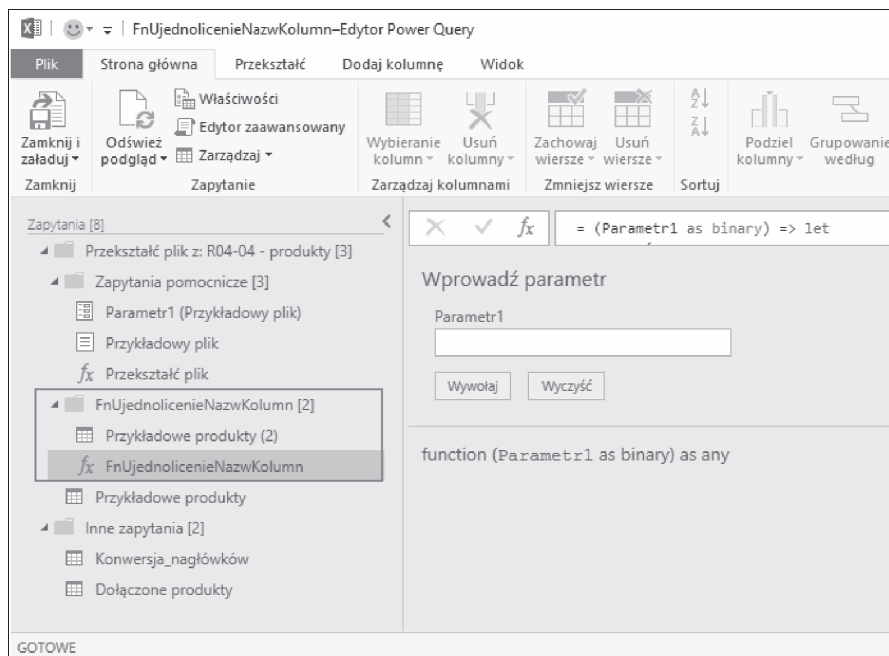
Teraz funkcja **FnUjednolicenieNazwKolumn** będzie wywoływana przez zapytanie *Przykładowe produkty*, a nie przez *Przykładowe produkty (2)*.

22. W panelu *Zapytania* zaznacz zapytanie *Przykładowe produkty*, a w panelu *Zastosowane kroki* zaznacz krok *Dołączone zapytanie*. W pasku formuły pojawi się następujący kod:

```
= Table.Combine({#"Przykładowe produkty (2)", #Usunięto pierwsze wiersze})
```

Zastąp go poniższym kodem:

```
= Table.Combine({FnUjednolicenieNazwKolumn(#"Parametr1"), #Usunięto pierwsze wiersze})
```



**RYSUNEK 4.10.** Funkcja *FnUjednolicenieNazwKolumn* z przykładowym zapytaniem „Przykładowe produkty (2)”

Funkcja *FnUjednolicenieNazwKolumn* odczytuje zawartość skoroszytu Excela podanego w jej argumencie i zwraca ujednolicone nazwy kolumn. Tabela #*Usunięto pierwsze wiersze* jest tworzona w ostatnim kroku zapytania *Przykładowe produkty* przed dołączeniem jej do wyniku. Zawiera ona wiersze danych bez nagłówków.

Dlaczego wykorzystywany jest tutaj parametr #*Parametr1*? Jak pamiętasz, parametr ten w zapytaniu *Przykładowe produkty (2)* zawsze zawiera nazwę pliku z akcesoriami. Jednak teraz, w kontekście funkcji, zawiera nazwę odpowiedniego pliku, określonego przez zapytanie *Przykładowe produkty*.

23. Wróć do zapytania *Dołączone produkty* i przewiń zawartość panelu podglądu do chwili pojawienia się w kolumnie *Source.Name* wartości *R04-04 - rowery.xlsx*. Jak się przekonasz, dane zostały umieszczone we właściwych kolumnach. Zamknij edytor Power Query i załaduj dane do raportu.

Gotowe rozwiązanie znajduje się w plikach *R04-06 - rozwiązanie - transpozycja nagłówków.xlsx* i *R04-06 - rozwiązanie - transpozycja nagłówków.pbix*.

## Ćwiczenie 4.7. Ujednolicenie nazw kolumn za pomocą języka M

W tym ćwiczeniu opisana jest najskuteczniejsza metoda ujednolicania nazw kolumn przy użyciu tabeli konwersji. Polega ona na użyciu języka M i zastosowaniu tabeli konwersji bezpośrednio w odniesieniu do nazw kolumn.

Jak pamiętasz, w ćwiczeniu 4.3 wykonałeś proste przekształcenie nazw kolumn za pomocą funkcji *Table.TransformColumnNames*, której drugim argumentem była właściwa funkcja modyfikująca tekst, np. *Text.Proper*, która zamieniała pierwsze litery każdego słowa na wielkie.

W ostatnim ćwiczeniu utworzysz niestandardową funkcję, która ujednolici nazwy kolumn według reguł zdefiniowanych w tabeli konwersji. Wcześniej jednak musisz poznać jeszcze jedno przekształcenie, przy użyciu którego kod będzie prostszy i będzie szybciej przeszukiwał tabelę konwersji. Jak pamiętasz, tabela ta zawiera pary wartości *Źródło* i *Cel*. Teraz przetransponujesz tę tabelę i użyjesz pierwszego wiersza jako nagłówków. W nowej tabeli wartości źródłowe będą nagłówkami kolumn, a docelowe będą umieszczone w pierwszym wierszu, tak jak na rysunku 4.11.

Źródło	Cel
Produkt	Nazwa
Nazwa_produkту	Nazwa
ID	Kod produktu
Kod_produkту	Kod produktu
Nr_produkту	Kod produktu
Kolor_produkту	Kolor

Produkt	Nazwa_produkту	ID	Kod_produkту	Nr_produkту	Kolor_produkту
Nazwa	Nazwa	Kod produ	Kod produktu	Kod produktu	Kolor

**RYSUNEK 4.11.** *Transpozycja oryginalnej tabeli konwersji skutkuje utworzeniem jednowierszowej tabeli zawierającej wartości źródłowe i docelowe, dzięki czemu metoda ujednolicania nazw kolumn jest bardziej wydajna*

Po takim przekształceniu funkcja niestandardowa będzie wyszukiwała wartości źródłowe w nagłówkach tabeli i odczytywała wartości docelowe z wiersza. Aby przetransponować tabelę konwersji, wykonaj poniższe kroki.

1. Otwórz kopię skoroszytu lub raportu, który utworzyłeś w ćwiczeniu 4.4, w krokach od 1. do 7.
2. W panelu *Zapytania* zaznacz zapytanie *Konwersja\_nagłówków*.
3. W karcie *Przekształć* kliknij ikonę *Transponuj*, a następnie *Użyj pierwszego wiersza jako nagłówków*.
4. W panelu *Zastosowane kroki* usuń automatycznie utworzony krok *Zmieniono typ1*. Nie jest on potrzebny i jeżeli go nie usuniesz, zmiany, które będziesz wprowadzał w przyszłości w kolumnie *Źródło*, mogą uniemożliwiać odświeżanie raportu.
5. W panelu *Zapytania* zaznacz zapytanie *Przykładowe produkty*, a następnie w pasku formuły kliknij przycisk *fx*, tak jak w ćwiczeniu 4.3, w kroku 7.
6. W pasku formuły pojawi się poniższy kod:  

```
= #Nagłówki o podwyższonym poziomie"
```

 Zmień go na następujący:  

```
= Table.TransformColumnNames(#Nagłówki o podwyższonym poziomie",  
  each try Table.Column(Konwersja_nagłówków, _) {0} otherwise _)
```
7. Naciśnij *Enter* i sprawdź nagłówki kolumn w panelu podglądu. Jak się przekonasz, zostały ujednolicone. Teraz kliknij zapytanie *Dodane produkty* i sprawdź, czy niezgodne tabele zostały poprawnie połączone.

Przeanalizujemy krótko powyższe wyrażenie. W ćwiczeniu 4.3 wykorzystałeś funkcję `Table.TransformColumnNames`. Tym razem skupimy się na jej drugim argumentcie:

```
each try Table.Column(Konwersja_nagłówków, _) {0} otherwise _
```

W rozdziale 9. dokładniej poznasz tego rodzaju wyrażenia. Kombinacja słowa kluczowego `each` i znaku podkreślenia jest skróconą definicją funkcji. W tym przypadku argumentem funkcji `Table.TransformColumnNames` jest inna funkcja, która za pomocą słowa `each` i znaku podkreślenia jest wywoływana dla każdej nazwy kolumny.



Wróćmy teraz do podstaw składni i przeanalizujemy powyższe wyrażenie krok po kroku. Do odwołania się do pierwszej komórki wiersza w tabeli konwersji użyłeś zapisu {0}. Jak pamiętasz, wartości indeksu w języku M zaczynają się od zera, zatem w celu odwołania się do pierwszej komórki należy użyć zera, a nie jedynki. Nawiasy klamrowe definiują listę, do której elementów można odwoływać się za pomocą indeksu.

Załóżmy, że tabela *Konwersja\_nagłówków* zawiera kolumnę *Kolumna1*. Aby odwołać się do listy wartości w kolumnie *Kolumna1*, należy użyć następującego wyrażenia:

```
Konwersja_nagłówków[Kolumna1]
```

W naszej formule nazwę kolumny reprezentuje znak podkreślenia umieszczony w argumencie funkcji. Aby odwoływać się do poszczególnych kolumn tabeli *Konwersja\_nagłówków* za pomocą ich nazw, należy użyć następującej funkcji:

```
each Table.Column(Konwersja_nagłówków, _)
```

Zwróć uwagę, że użyta została funkcja `Table.Column`. Poniższe wyrażenie nie jest poprawne:

```
each Konwersja_nagłówków[_]
```

Wyrażenie to odwołuje się do kolumny o nazwie `_` i w efekcie pojawi się następujący komunikat:

*Expression.Error: Nie można znaleźć kolumny „\_” w tabeli.*

Aby uniknąć tego problemu, należy użyć funkcji `Table.Column`, której argumentami są nazwa tabeli i nazwa funkcji, a zwracanym wynikiem lista kolumn. Łącząc oba opisane wyrażenia, można odwoływać się do wartości docelowej za pomocą wartości źródłowej będącej nazwą kolumny w przetransponowanej tabeli konwersji utworzonej w kroku 3.

```
each Table.Column(Konwersja_nagłówków, _){0}
```

Dlaczego w kroku 6. zostały dodatkowo użyte słowa `try/otherwise`? Jak wiesz, tabela konwersji nie zawiera wszystkich nazw kolumn, a jedynie te niezgodne. Dlatego wyrażenie z kroku 6. zawiera słowa `try/otherwise` wyróżnione pogrubioną czcionką:

```
each try Table.Column(Konwersja_nagłówków, _){0} otherwise _
```

Jeżeli wartość źródłowa nie zostanie znaleziona w tabeli *Konwersja\_nagłówków*, wtedy wyrażenie po słowie `try` nie zostanie wykonane. Zamiast niego zostanie wykonane wyrażenie po słowie `otherwise` zwracające wartość źródłową.

Poniżej ponownie jest pokazana cała funkcja:

```
= Table.TransformColumnNames("#Nagłówki o podwyższonym poziomie",
  each try Table.Column(Konwersja_nagłówków, _){0} otherwise _)
```

Funkcję tę można przełożyć na naturalny język w następujący sposób: „weź tabelę utworzoną w kroku *Nagłówki o podwyższonym poziomie* i zmień nazwy jej kolumn według następującej reguły: spróbuj znaleźć w tabeli konwersji kolumnę o nazwie takiej jak kolumna w powyższej tabeli. Jeżeli się uda, zwróć znajdującą się w tej kolumnie nazwę docelową. W przeciwnym przypadku zwróć nazwę kolumny z tabeli *Nagłówki o podwyższonym poziomie*”.

Jeżeli nie rozumiesz wszystkich opisanych tu zawiłości, nie przejmuj się. Do łączenia niezgodnych tabel przy użyciu tabeli konwersji możesz stosować metody opisane w ćwiczeniach 4.4, 4.5 i 4.6. Celem ćwiczenia 4.7 było pokazanie, jak szybko można łączyć niezgodne tabele za pomocą języka M. Warto, abyś wrócił do tego ćwiczenia, gdy po przeczytaniu rozdziału 9. dowiesz się więcej o tym języku.

Gotowe rozwiązanie znajduje się w plikach *R04-07 - rozwiązanie - język M.xlsx* i *R04-07 - rozwiązanie - język M.pbix*.

## Podsumowanie

W tym rozdziale poznałeś skutki łączenia niezgodnych tabel oraz różne techniki umożliwiające wiązanie nazw kolumn i pomyślne łączenie tabel.

Dowiedziałeś się, jak rozpoznawać symptomy niezgodności tabel, takie jak podział danych i brakujące wartości. Nauczyłeś się również ręcznie zmieniać nazwy kolumn przed połączeniem tabel.

Jeżeli we wszystkich tabelach zapisanych w folderze kolumny są ułożone w takiej samej kolejności, wtedy możesz stosować technikę uzgadniania nagłówków. Poznałeś funkcję `Table.TransformColumnNames` i użyłeś jej do wykonania prostych przekształceń nazw kolumn, takich jak zamiana pierwszych liter słów z małych na wielkie oraz zastąpienie znaków podkreślenia spacjami.

Za pomocą tabeli konwersji można ujednociliac nazwy kolumn w sytuacjach, gdy nie jest możliwe użycie funkcji przekształcających tekst. Poznałeś trzy techniki wykorzystujące tabelę konwersji do zmiany niezgodnych nagłówków: transpozycję, anulowanie przestawienia i transpozycję samych nazw kolumn. Na koniec zastosowałeś najskuteczniejszą metodę ujednociania nagłówków wykorzystującą kod w języku M.

Poniższa tabela zawiera podsumowanie opisanych technik wraz z ich zaletami i wadami.

Technika	Zalety	Wady
1. Uzgodnienie nagłówków	Prosta i wydajna implementacja.	Jeżeli tabele różnią się kolejnością kolumn, wtedy wartości są zamienione miejscami.
2. Transpozycja, scalenie i ponowna transpozycja	Prosta implementacja uwzględniająca różnice w kolejności kolumn.	Duże zapotrzebowanie na pamięć. Technika nieskuteczna, jeżeli połączona tabela składa się z więcej niż 16 384 wierszy.
3. Anulowanie przestawienia, scalenie i przestawienie	Prosta implementacja przystosowana do dużych ilości danych, uwzględniająca różnice w kolejności kolumn.	Technika znacznie wolniejsza od 1., 4. i 5.
4. Transpozycja samych nazw kolumn	Technika uwzględniająca różnice w kolejności kolumn. Odświeżanie raportu odbywa się znacznie szybciej niż w technikach 2 i 3.	Najdłuższy czas implementacji.
5. Ujednoczenie nazw kolumn przy użyciu języka M	Najskuteczniejsza technika pod względem wydajności. Szybka implementacja.	Wymagana znajomość języka M.

Możliwość łączenia niezgodnych tabel oferuje wiele korzyści. Wyobraź sobie, że pracujesz w firmie, która co jakiś czas przejmuje inne firmy. Techniki opisane w tym rozdziale możesz wykorzystywać do szybkiego konsolidowania danych pochodzących z przejętych firm i tworzenia raportów, zanim dział IT zintegruje bazy danych. Opisane metody analityki biznesowej robią wielkie wrażenie podczas łączenia dużych ilości tego rodzaju danych.

# PROGRAM PARTNERSKI

— GRUPY HELION —

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

Zmień swoją stronę WWW w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

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

GRUPA  
**Helion** 

# Power Query i Power BI: sprostasz wyzwaniu przetwarzania danych!

Czy praca na danych kojarzy Ci się z frustrującym ręcznym wklejaniem i oczyszczaniem danych w Excelu? A może tracisz mnóstwo energii na importowanie nieuporządkowanych danych pochodzących z różnych źródeł? Być może podejrzewasz, że Twoje arkusze kryją mnóstwo wartościowych informacji, ale nie wiesz, w jaki sposób je wydobyć i ile godzin musisz na to poświęcić? A może spędzasz sporo czasu na tworzeniu skomplikowanych raportów w Power BI lub programowaniu baz danych SQL Server? Dzięki wiedzy zawartej w tej książce wykonasz te zadania prościej i szybciej, a efekty będą o wiele bardziej niezawodne!

Ten praktyczny przewodnik po narzędziu Power Query nauczy Cię efektywnie przetwarzać dane w Excelu: importować, oczyszczać, przekształcać i wydobywać potrzebne informacje. Dowiesz się, jak sprawnie wykonywać typowe zadania i prace analityczne, jak zwiększyć swoją skuteczność dzięki opanowaniu podstaw języka M oraz jak zautomatyzować proces przygotowywania danych do przetwarzania. Zapoznasz się z możliwościami Microsoft Cognitive Services oraz konektora Power Query Web, a także z wykorzystaniem sztucznej inteligencji do rozpoznawania treści tekstu. Nabyte umiejętności będziesz mógł przetestować w realistycznym projekcie, któremu poświęcono ostatni rozdział!

## W książce między innymi:

- solidne podstawy pracy z narzędziem Power Query
- automatyzacja operacji na danych
- proste i zaawansowane techniki dekompozycji tabel
- tworzenie procedur w języku M
- pozyskiwanie informacji z danych tekstowych i sieci społecznościowych

## Gil Raviv

jest programistą od ponad dwudziestu lat, zdobył cztery patenty w dziedzinie bezpieczeństwa cyfrowego, analizy danych i sieci społecznościowych. Jest też certyfikowanym specjalistą Microsoft — kierował tworzeniem i integracją dodatku Power Query w Excelu. Opracował wiele różnorodnych programów, od zaawansowanych systemów wykrywania zagrożeń w firmach po ochronę dzieci w serwisie Facebook. Obecnie jest menedżerem zespołu w firmie Avanade. Zajmuje się rozwiązaniami opartymi na narzędziu Power BI i chmurze Azure.

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

KOD KORZYŚCI  
Sięgnij po więcej! ▶



ISBN 978-83-8322-471-8



Cena: 89,00 zł

Microsoft Press