

Michael J. Hernandez

Słowo wstępne Michelle Poolet

PROJEKTOWANIE BAZ DANYCH

DLA KAŻDEGO

PRZEWODNIK KROK PO KROKU

WYDANIE IV

Podejście niezależne od oprogramowania!

Niezależnie od oprogramowania, z jakiego korzystasz do tworzenia aplikacji bazodanowych, ta książka pozwoli Ci zaoszczędzić pieniądze i czas oraz uniknąć wielu godzin irytacji — i to zanim jeszcze napiszesz pierwszy wiersz kodu!

EDYCJA
NA

25
LECIE

Tytuł oryginału: Database Design for Mere Mortals: 25th Anniversary Edition, 4th Edition

Tłumaczenie: Ireneusz Jakóbiak, Radosław Meryk, Katarzyna Żarnowska, Tomasz Walczak

ISBN: 978-83-283-8251-0

Authorized translation from the English language edition, entitled Database Design for Mere Mortals: 25th Anniversary Edition, 4th Edition by Michael J Hernandez, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright © 2021 Michael J. Hernandez.

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 © 2022.

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. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/projb4>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

O autorze	13
Podziękowania	14
Słowo wstępne	17
Przedmowa	18
Wprowadzenie	19
Co nowego w czwartym wydaniu?	20
Kto powinien przeczytać tę książkę?	21
Cel niniejszej książki	21
Jak czytać tę książkę?	23
Organizacja książki	23
Część I. Projektowanie relacyjnych baz danych	23
Część II. Proces projektowania	24
Część III. Inne problemy projektowania baz danych	24
Dodatki	25
Słowo na temat przykładów i technik opisywanych w tej książce	25
Nowe podejście do nauki	26

Część I Projektowanie relacyjnych baz danych31

Rozdział 1. Relacyjna baza danych	33
Tematy omówione w tym rozdziale	31
Czym jest baza danych?	31
Model relacyjnych baz danych	32
Pozyskiwanie danych	33
Zalety relacyjnych baz danych	35
Zarządzanie relacyjną bazą danych	36
Co niesie przyszłość?	36
Podsumowanie	38
Pytania kontrolne	38

Rozdział 2. Cele projektowania	39
Tematy omówione w tym rozdziale	39
Dlaczego projektowanie baz danych powinno nas interesować	39
Znaczenie teorii	40
Zalety poznania dobrej metodologii projektowania	42
Cele dobrego projektowania	42
Korzyści wynikające z dobrego projektowania	43
Metody projektowania baz danych	44
Tradycyjne metody projektowania	44
Metoda projektowania prezentowana w tej książce	45
Normalizacja	46
Podsumowanie	47
Pytania kontrolne	48
Rozdział 3. Terminologia	49
Tematy omówione w tym rozdziale	49
Dlaczego terminologia jest ważna	49
Pojęcia związane z wartością	50
Dane	50
Informacje	51
Null	52
Wartość znaczników null	52
Problem ze znacznikami null	53
Pojęcia związane ze strukturą	55
Tabele	55
Pole	56
Rekord	57
Widok (perspektywa)	58
Klucze	59
Indeks	61
Pojęcia związane z relacjami	61
Relacje	61
Typy relacji	61
Typy uczestnictwa	65
Stopień uczestnictwa	65
Pojęcia związane z integralnością	66
Specyfikacja pola	66
Integralność danych	67
Podsumowanie	68
Pytania kontrolne	68

Część II Proces projektowania71**Rozdział 4. Przegląd koncepcyjny73**

Tematy omówione w tym rozdziale	73
Dlaczego ważna jest realizacja całego procesu projektowania	74
Formułowanie definicji celu i założeń wstępnych	75
Analiza istniejącej bazy danych	75
Tworzenie struktur danych	76
Określanie i ustalanie relacji w tabelach	77
Określanie i definiowanie reguł biznesowych	77
Określanie i definiowanie widoków	78
Kontrola integralności danych	78
Podsumowanie	79
Pytania kontrolne	80

Rozdział 5. Rozpoczęcie procesu projektowania81

Tematy omówione w tym rozdziale	81
Przeprowadzanie wywiadów	82
Wytyczne dotyczące rozmówców	83
Wytyczne dotyczące osoby przeprowadzającej wywiad	84
Formułowanie definicji celu	86
Poprawnie sformułowana definicja celu	86
Układanie definicji celu	87
Formułowanie założeń wstępnych	90
Poprawnie sformułowane założenia wstępne	90
Układanie założeń wstępnych	91
Podsumowanie	94
Pytania kontrolne	94

Rozdział 6. Analiza istniejącej bazy danych96

Tematy omówione w tym rozdziale	96
Poznanie istniejącej bazy danych	96
Papierowe bazy danych	98
Odziedziczone bazy danych	99
Przeprowadzenie analizy	99
Spojrzenie na sposób gromadzenia danych	100
Spojrzenie na sposób prezentowania informacji	102
Przeprowadzanie wywiadów	105
Podstawowe techniki przeprowadzania wywiadów	105
Zanim rozpoczniesz przeprowadzanie wywiadów... ..	109

Wywiady z użytkownikami	109
Przegląd typów danych i sposobów ich wykorzystania	110
Przegląd próbek	111
Przegląd wymagań informacyjnych	113
Wywiady z kierownictwem	119
Przegląd obecnych wymagań informacyjnych	119
Przegląd dodatkowych wymagań informacyjnych	120
Przegląd przyszłych wymagań informacyjnych	120
Przegląd ogólnych wymagań informacyjnych	121
Stworzenie kompletnej listy pól	121
Wstępna lista pól	121
Lista pól obliczeniowych	126
Przegląd obu list wraz z pracownikami i kierownictwem	126
Podsumowanie	131
Pytania kontrolne	132
Rozdział 7. Tworzenie struktur tabel	133
Tematy omówione w tym rozdziale	133
Definiowanie wstępnej listy tabel	134
Identyfikacja domniemanych podmiotów	134
Korzystanie z listy podmiotów	135
Korzystanie z założeń wstępnych	138
Definiowanie ostatecznej listy tabel	140
Dopracowywanie nazw tabel	141
Określanie typów tabel	145
Redagowanie opisów tabel	145
Przypisywanie pól do tabel	149
Dopracowywanie pól	151
Poprawianie nazw pól	151
Korzystanie z idealnego pola do eliminowania anomalii	154
Eliminacja pól wieloczęściowych	157
Eliminacja pól wielowartościowych	159
Dopracowywanie struktur tabel	164
Kilka słów o nadmiarowych danych i duplikatach pól	164
Wykorzystanie warunków idealnej tabeli w celu dopracowania struktur tabel	164
Wyznaczanie tabel-podzbiorów	169
Podsumowanie	179
Pytania kontrolne	180

Rozdział 8. Klucze	182
Tematy omówione w tym rozdziale	182
Dlaczego klucze są ważne	182
Definiowanie kluczy dla tabel	183
Klucze kandydujące	183
Klucze główne	188
Klucze zastępcze	192
Pola niekluczowe	193
Integralność na poziomie tabeli	193
Przegląd wstępnych struktur tabel	194
Podsumowanie	199
Pytania kontrolne	200
Rozdział 9. Specyfikacje pól	201
Tematy omówione w tym rozdziale	201
Dlaczego specyfikacje pól są ważne	202
Integralność na poziomie pól	203
Anatomia specyfikacji pól	203
Elementy ogólne	204
Elementy fizyczne	208
Elementy logiczne	210
Wykorzystywanie unikatowych, ogólnych i kopiowanych specyfikacji pól	213
Definiowanie specyfikacji pól dla każdego pola w bazie danych	216
Podsumowanie	219
Pytania kontrolne	220
Rozdział 10. Relacje między tabelami	221
Tematy omówione w tym rozdziale	221
Dlaczego relacje są ważne	222
Rodzaje relacji	223
Relacja jeden do jednego	223
Relacja jeden do wielu	225
Relacja wiele do wielu	227
Relacja zwrotna	233
Identyfikowanie istniejących relacji	235
Ustanawianie wszystkich relacji	242
Relacje jeden do jednego i jeden do wielu	243
Relacja wiele do wielu	248
Relacje zwrotne	252
Sprawdzanie struktury wszystkich tabel	256
Dokładna analiza wszystkich kluczy obcych	257

Ustanawianie cech relacji	261
Definiowanie reguły usuwania dla każdej relacji	261
Identyfikowanie rodzaju uczestnictwa każdej z tabel	265
Identyfikowanie stopnia uczestnictwa każdej z tabel	267
Weryfikowanie z użytkownikami i kierownictwem relacji istniejących między tabelami	269
Uwaga końcowa	269
Integralność na poziomie relacji	269
Podsumowanie	273
Pytania kontrolne	274
Rozdział 11. Reguły biznesowe	276
Tematy omówione w tym rozdziale	276
Czym są reguły biznesowe?	276
Rodzaje reguł biznesowych	278
Kategorie reguł biznesowych	280
Reguły biznesowe specyficzne dla pól	280
Reguły biznesowe specyficzne dla relacji	281
Definiowanie i ustanawianie reguł biznesowych	282
Praca z użytkownikami oraz kierownictwem	282
Definiowanie i ustanawianie reguł biznesowych specyficznych dla pola	283
Definiowanie i ustanawianie reguł biznesowych specyficznych dla relacji	289
Tabele walidacji	294
Czym są tabele walidacji?	294
Korzystanie z tabel walidacji w celu realizowania reguł biznesowych	295
Sprawdzanie arkuszy specyfikacji reguł biznesowych	297
Podsumowanie	302
Pytania kontrolne	304
Rozdział 12. Widoki	305
Tematy omówione w tym rozdziale	305
Czym są widoki?	305
Anatomia widoku	306
Widok danych	307
Widok zagregowany	311
Widok walidacji	313
Określanie i definiowanie widoków	315
Praca z użytkownikami i kierownictwem	315
Definiowanie widoków	316
Przeglądanie dokumentacji każdego widoku	323

Podsumowanie	326
Pytania kontrolne	328
Rozdział 13. Sprawdzanie integralności danych	329
Tematy omówione w tym rozdziale	329
Dlaczego należy sprawdzać integralność danych?	330
Sprawdzanie i korygowanie integralności danych	330
Integralność na poziomie tabel	330
Integralność na poziomie pól	331
Integralność na poziomie relacji	331
Reguły biznesowe	331
Widoki	332
Kompletowanie dokumentacji bazy danych	332
W końcu zrobione!	333
Podsumowanie	334
 Część III Inne problemy projektowania	
baz danych	335
Rozdział 14. Czego nie należy robić?	337
Tematy omówione w tym rozdziale	337
Płaskie pliki	337
Projekt na bazie arkusza kalkulacyjnego	339
Rozwiązywanie problemów związanych z przyzwyczajeniami	
do widoku arkusza kalkulacyjnego	340
Projekt bazy danych pod kątem konkretnego oprogramowania	341
Wnioski końcowe	342
Podsumowanie	343
Rozdział 15. Naginanie bądź łamanie reguł	344
Tematy omówione w tym rozdziale	344
Kiedy można nagiąć lub złamać reguły?	344
Projektowanie analitycznej bazy danych	344
Poprawianie wydajności obliczeń	345
Dokumentowanie działań	347
Podsumowanie	348
Rozdział 16. Na zakończenie	349

Dodatki	351
Dodatek A Odpowiedzi na pytania kontrolne	353
Dodatek B Diagram procesu projektowania baz danych	368
Dodatek C Wytyczne projektowe	385
Dodatek D Formularze dokumentacyjne	393
Dodatek E Symbole używane w diagramach stosowanych w procesie projektowania baz danych	396
Dodatek F Przykładowe projekty	398
Dodatek G O normalizacji	404
Dodatek H Zalecana lektura	411
 Słowniczek	 413
Literatura	422

1

Relacyjna baza danych

*Ryba musi pływać trzy razy —
w wodzie, w maśle i w winie*
— polskie przysłowie

Tematy omówione w tym rozdziale

Czym jest baza danych?

Relacyjna baza danych

Co niesie przyszłość?

Podsumowanie

Pytania kontrolne

Relacyjne bazy danych istnieją od 50 lat. Naprawdę — od 50 lat! Ten najbardziej rozpowszechniony na świecie typ baz danych niezbędnych w naszym codziennym życiu rozkręcił branżę wartą miliardy dolarów. Jest bardzo prawdopodobne, że korzystasz z relacyjnej bazy danych za każdym razem, kiedy robisz zakupy przez internet lub w lokalnym sklepie, układasz plan podróży, wypożyczasz książki bądź zamawiasz jedzenie przez aplikację na urządzeniu mobilnym.

Czym jest baza danych?

Co to jest baza danych? Jak zapewne wiesz, baza danych to zorganizowana kolekcja danych wykorzystywanych do modelowania niektórych typów organizacji lub ich procesów. Nie ma znaczenia, czy do zbierania i przechowywania danych używasz arkuszy kalkulacyjnych, czy aplikacji komputerowej. Jeśli tylko zbierasz dane w zorganizowany sposób i w konkretnym celu, masz bazę danych. W dalszej części tej książki zakładam, że do zbierania i przechowywania danych wykorzystywane są aplikacje.

Istnieją dwa rodzaje baz danych: *operacyjne* i *analityczne*.

Operacyjne bazy danych są kręgosłupem wielu firm, organizacji oraz instytucji na całym świecie. Ten rodzaj bazy danych jest wykorzystywany głównie do *przetwarzania transakcji*

online (OLTP) w sytuacjach, kiedy istnieje potrzeba zbierania, modyfikacji i utrzymania danych każdego dnia. Dane przechowywane w operacyjnej bazie danych są *dynamiczne*, co znaczy, że wciąż się zmieniają i zawsze odzwierciedlają aktualne informacje. Organizacje takie jak sklepy, firmy produkcyjne, szpitale, kliniki i wydawnictwa korzystają z operacyjnych baz danych, ponieważ ich dane ciągle się zmieniają.

Analityczne bazy danych są głównie wykorzystywane przy *analitycznym przetwarzaniu online* (OLAP) w sytuacjach, kiedy istnieje potrzeba przechowywania i śledzenia danych historycznych i zależnych od czasu. Analityczna baza danych jest cennym zasobem, jeśli trzeba prześledzić trendy, przejrzeć dane statystyczne z długiego zakresu czasu oraz stworzyć taktyczne lub strategiczne projekcje biznesowe. Ten typ bazy danych przechowuje dane *statyczne*, co oznacza, że dane te nie zmieniają się nigdy (lub bardzo rzadko). Informacje zebrane w analitycznej bazie danych pokazują dane dotyczące konkretnego momentu w czasie. Laboratoria chemiczne, firmy geologiczne oraz agencje marketingowe zajmujące się analizą to przykłady firm, które mogą wykorzystywać analityczne bazy danych.

Analityczne bazy danych często wykorzystują dane z baz operacyjnych jako główne źródło, mogą więc istnieć między nimi powiązania. Jednakże operacyjne i analityczne bazy danych spełniają bardzo specyficzne potrzeby w zakresie przetwarzania danych, a tworzenie ich struktur wymaga radykalnie odmiennych metodologii projektowych. Ta książka skupia się na projektowaniu operacyjnych baz danych, ponieważ są one najbardziej rozpowszechnione.

Model relacyjnych baz danych

Relacyjna baza danych powstała w 1969 roku i jest wciąż jednym z najszerzej wykorzystywanych modeli w zarządzaniu danymi. Ojcem modelu relacyjnego jest dr Edgar F. Codd, który w późnych latach 60. pracował jako naukowiec w IBM i szukał nowych sposobów na radzenie sobie z dużymi ilościami danych. Jego niezadowolenie z ówczesnych modeli baz danych doprowadziło go do rozważań na temat wykorzystania dyscyplin i struktur matematycznych do rozwiązania niezliczonych problemów, które napotykał. Jako zawodowy matematyk mocno wierzył, że może wykorzystać określone gałęzie matematyki do rozwiązania problemów takich jak nadmiarowe dane, ich słaba integralność oraz zbyt duża zależność struktur baz danych od ich fizycznej implementacji.

Dr Codd formalnie ujął nowy model relacyjny w swojej książce *A Relational Model of Data for Large Shared Databases*¹ wydanej w czerwcu 1970 roku. Swoją nową model oparł na dwóch gałęziach matematyki — teorii zbiorów i logice predykatów pierwszego rzędu. Sama nazwa modelu pochodzi od terminu *relacja*, który jest częścią teorii zbiorów. Szeroko rozpowszechniona błędna koncepcja głosi, że model relacyjny zapożyczył swoją nazwę od powiązań pomiędzy tabelami relacyjnej bazy danych.

Relacyjna baza danych przechowuje dane w *relacjach*, które są przez użytkowników postrzegane jako tabele. Każda relacja składa się z *krotek*, zwanych rekordami, oraz *atrybutów*, zwanych polami. W dalszej części książki będę używał terminów *tabele*, *rekordy* oraz *pola*. Fizyczny układ rekordów lub pól jest zupełnie nieistotny, a każdy rekord w tabeli jest możliwy do zidentyfikowania poprzez pole zawierające unikatową wartość. To są dwie cechy

¹ Edgar F. Codd, *A Relational Model of Data for Large Shared Databases*, „Communications of the ACM”, czerwiec 1970, s. 377 – 387.

charakterystyczne relacyjnej bazy danych, które pozwalają, by dane istniały niezależnie od sposobu ich fizycznego przechowywania w komputerze. W związku z tym, by wydobyć dane, użytkownik nie musi znać fizycznej lokalizacji rekordu.

Model relacyjny kategoryzuje zależności na *jeden do jednego*, *jeden do wielu* oraz *wiele do wielu*. Te zależności zostaną szczegółowo omówione w rozdziale 10., „Relacje między tabelami”. Zależność pomiędzy parą tabel jest automatycznie ustalana na podstawie dopasowywania wartości we wspólnych polach. Na przykład na rysunku 1.1 tabele KLIENCI i AGENCI są ze sobą powiązane relacją *jeden do wielu* poprzez pole NR AGENTA. Konkretny agent jest powiązany z jednym lub z wieloma klientami poprzez pasujący NR AGENTA. W ten sam sposób tabele ARTYŚCI oraz ANGAŻE powiązane są ze sobą relacją *jeden do wielu* poprzez NR ARTYSTY. Rekord w tabeli ARTYŚCI może być powiązany z jednym lub z wieloma angażami z tabeli ANGAŻE poprzez pasujący NR ARTYSTY.

O ile użytkownik zna zależności występujące pomiędzy tabelami w bazie danych, może uzyskać dostęp do danych na niemal nieskończoną ilość sposobów. Może wydobywać dane z tabel, które są ze sobą powiązane bezpośrednio oraz pośrednio. Przyjrzyj się tabelom na rysunku 1.1. Mimo że tabela KLIENCI łączy się z tabelą ANGAŻE pośrednio, użytkownik jest w stanie uzyskać listy klientów i artystów, którzy dla nich pracowali (to oczywiście zależy od tego, jak skonstruowane są tabele, ale dla naszych potrzeb ten przykład jest wystarczający). Użytkownik zrobi to z łatwością, ponieważ tabela KLIENCI łączy się bezpośrednio z tabelą ANGAŻE, a ta z kolei z tabelą ARTYŚCI.

Pozyskiwanie danych

W bazie relacyjnej dane pozyskuje się, wykorzystując *strukturalny język zapytań* SQL (ang. *Structured Query Language*). SQL to standardowy język wykorzystywany do tworzenia, modyfikowania i utrzymywania relacyjnej bazy danych oraz tworzenia zapytań. Na rysunku 1.2 pokazana jest przykładowa deklaracja SQL, którą możesz wykorzystać do otrzymania listy klientów w mieście Katowice.

Podstawowa kwerenda SQL składa się z trzech komponentów: deklaracji SELECT...FROM, warunku WHERE oraz warunku ORDER BY. Warunek SELECT wykorzystujesz, by wskazać pola, których chcesz użyć w kwerendzie, a warunek FROM, by wskazać tabelę lub tabele, do których należą te pola. Możesz filtrować rekordy, które zwraca kwerenda, poprzez określanie za pomocą warunku WHERE kryteriów dotyczących jednego lub wielu pól, a następnie sortować rezultaty w porządku rosnącym lub malejącym za pomocą warunku ORDER BY.

Większość dzisiejszych programów obsługujących relacyjne bazy danych zawiera w sobie różne formy implementacji SQL, poczynając od okien, w których użytkownicy wpisują „surowe” komendy SQL, aż po narzędzia pozwalające użytkownikom na budowanie kwerend za pomocą elementów graficznych. Na przykład użytkownik pracujący z oprogramowaniem R:BASE firmy R:BASE Technologies może wybrać budowanie i wykonywanie poleceń SQL wprost z okna poleceń, a ktoś korzystający z Microsoft SQL Server może uznać, że łatwiej budować kwerendy, wykorzystując narzędzie graficzne. Bez względu na sposób budowy kwerend użytkownik może je zachować do następnego wykorzystania.

Agenci

Nr agenta	Imię agenta	Nazwisko agenta	Data zatrudnienia	Numer telefonu agenta
100	Natalia	Wirska	16-05-2020	22 6590051
101	Bartosz	Łucki	15-10-2020	22 6986523
102	Renata	Woj	01-03-2021	32 2958596

Klienci

Nr klienta	Nr agenta	Imię klienta	Nazwisko klienta	Numer telefonu klienta	
9001	102	Stanisław	Wojciechowski	504263565	
9002	101	Zuzanna	Bartnicka	781564231	
9003	102	Elwira	Rosińska	606581234	

Artyści

Nr artysty	Nr agenta	Imię artysty	Nazwisko artysty	
3000	100	Jarosław	Śmietana	
3001	101	Anna	Dąbrowska	
3002	102	Mirosław	Czyżykiewicz	

Angaże

Nr klienta	Nr artysty	Data angażu	Początek	Koniec
9003	3002	01-04-2021	13:00	15:30
9009	3000	13-04-2021	21:00	01:30
9001	3002	02-05-2021	15:00	18:00

Rysunek 1.1. Przykłady tabel powiązanych w relacyjnej bazie danych

```
SELECT NazwiskoKlienta, ImięKlienta, TelefonKlienta
FROM Klienci
WHERE Miasto = "Katowice"
ORDER BY NazwiskoKlienta, ImięKlienta
```

Rysunek 1.2. Przykładowa kwerenda w języku SQL

Do pracy z bazami danych nie zawsze konieczna jest znajomość języka SQL. Jeśli oprogramowanie daje możliwość graficznego tworzenia kwerend lub jest zbudowane specjalnie dla Twojej bazy danych, samodzielne wpisywanie komend nie jest konieczne. Dobrze jest jednak poznać podstawy SQL-a. Osobom korzystającym z narzędzi tworzenia kwerend pomoże to w zrozumieniu i poprawieniu ewentualnych błędów w kwerendach, przyda się także w przypadku konieczności skorzystania z oprogramowania wyższej klasy, takiego jak Oracle lub Microsoft SQL Server.

Uwaga Mimo iż szczegółowa analiza języka SQL wykracza poza zakres tej książki, musisz zrozumieć, że SQL to język bezpośrednio powiązany z modelem relacyjnych baz danych. Jeśli masz potrzebę lub ochotę poznać SQL-a, możesz zacząć od przeczytania jednej z moich książek *SQL Queries for Mere Mortals*, wydanie IV², a następnie zapoznać się z innymi książkami dotyczącymi SQL-a, które wymienione są w dodatku H.

Zalety relacyjnych baz danych

Relacyjna baza danych ma wiele zalet. Należą do nich:

- *Wbudowana, wielopoziomowa integralność.* Integralność danych jest wbudowana w model na poziomie pola, aby zapewnić dokładność danych; na poziomie tabeli, by upewnić się, że rekordy nie są duplikowane, oraz by wykryć brakujące wartości klucza głównego; na poziomie zależności, by upewnić się, że zależność pomiędzy dwiema tabelami jest prawidłowa; na poziomie firmy, by przekonać się, że dane są dokładne w sensie biznesowym. Temat integralności będzie poruszany w miarę omawiania procesu projektowego.
- *Logiczna i fizyczna niezależność danych od aplikacji bazodanowych.* Ani zmiany poczynione przez użytkownika na poziomie logicznego projektu bazy danych, ani też zmiany oprogramowania wprowadzane przez producenta na poziomie fizycznej implementacji nie wpłyną niekorzystnie na aplikacje zbudowane w oparciu o bazę danych.
- *Gwarantowana spójność i dokładność danych.* Dane są podawane spójnie i dokładnie dzięki wielu poziomom integralności, które możesz narzucić bazie danych. Ten temat stanie się jasny w miarę omawiania procesu projektowego.
- *Łatwe pozyskiwanie danych.* Dane mogą być pozyskane z konkretnej tabeli lub z dowolnej liczby powiązanych tabel z bazy danych zgodnie z poleceniami od użytkownika. To pomaga użytkownikowi przeglądać informacje na wiele różnych sposobów.

Te i inne zalety okazały się korzystne dla środowiska biznesowego oraz dla tych wszystkich, którzy zbierają dane i nimi zarządzają. W wielu przypadkach relacyjna baza danych stała się pożądanym wyborem.

Główną wadą relacyjnych baz danych jest to, że bazujące na nich oprogramowanie działa bardzo wolno. Nie jest to wina samego modelu relacyjnego, lecz dostępności technologii pomocniczych w momencie wprowadzania modelu. Szybkość przetwarzania, pamięć oraz pojemność były po prostu niewystarczające, by zapewnić producentom oprogramowania do tworzenia baz danych platformę, na której mogliby zbudować pełną implementację relacyjnej bazy danych. Z tego powodu pierwsze programy do tworzenia relacyjnych baz danych nie pozwalały na rozwinięcie ich pełnego potencjału. Postępy poczynione przez ostatnich 50 lat zarówno w technologii produkcji sprzętu, jak i w inżynierii oprogramowania sprawiły, że szybkość przetwarzania oraz odczytu i zapisu nie stanowi już problemu. Umożliwiło to producentom zapewnienie dużo bardziej kompletnej obsługi modelu relacyjnego.

² Wydanie polskie: *Zapytania w SQL. Przyjazny przewodnik*, Helion, Gliwice 2020 — *przyp. tłum.*

Więcej na temat modelu relacyjnych baz danych dowiesz się z dalszej części tej książki. Niektóre z poruszonych tematów będą dotyczyły tworzenia tabel, zapewniania integralności danych, pracy z zależnościami i ustalania reguł biznesowych.

Zarządzanie relacyjną bazą danych

System zarządzania relacyjną bazą danych (SZRBD) jest aplikacją wykorzystywaną do tworzenia, utrzymywania i modyfikacji relacyjnej bazy danych oraz do operowania nią. Wiele systemów SZRBD zapewnia także narzędzia niezbędne do tworzenia aplikacji dla użytkownika, które wchodzą w interakcje z danymi przechowywanymi w bazie danych. Oczywiście jakość SZRBD zależy od tego, w jakim stopniu obsługuje on model relacyjnych baz danych. Nawet w przypadku „prawdziwych” systemów SZRBD poziom obsługi relacyjnych baz danych różni się w zależności od producenta, a nikt jeszcze nie wykorzystał *pełnego* potencjału tego modelu. Bez względu na to wszystkie systemy SZRBD ewoluują i oferują coraz więcej możliwości. Oto przykładowe SZRBD: IBM DB2, IBM Informix, Microsoft Access, Microsoft SQL Server, MySQL, Oracle RDBMS, PostgreSQL, SAP SQL Anywhere, SAP Sybase ASE i SQLite.

Co niesie przyszłość?

Co niesie przyszłość? To bardzo dobre pytanie. Uważam, że branża technologii, inżynierii oprogramowania i baz danych ewoluowała szybciej, niż ktokolwiek mógł sobie wyobrazić. Możesz stwierdzić: „Ale Mike, to trwało aż 50 lat!”. To prawda — i cóż to były za lata!

Gdy byłem dzieckiem i czytałem komiksy o superbohaterach z dziedziny fantastyki naukowej, samochody, gadzety i różne technologie wydawały się odległą przyszłością. Faktem jest też, że czytywałem osadzone w latach 90. ubiegłego wieku komiksy, w których ludzie podróżowali latającymi samochodami i używali niewielkich urządzeń antygrawitacyjnych! *Wciąż* nie mogę się doczekać na osobisty pas do latania. Gdy byłem nastolatkiem, oglądałem futurystyczny serial *Star Trek*, w którym bohaterowie używali małych przenośnych urządzeń i *odznak* do komunikacji, mówiących komputerów, monitorów z płaskim ekranem, replikatorów żywności itd. Zastanawiałem się: „Kurczę, czy dożyję czasów, w których będę mógł oglądać wszystkie te fajne rzeczy i ich używać?”.

Szybko prześledźmy lata 70., 80. i 90. W tym czasie pojawiły się komputery mieszczące się na biurku, przenośne telefony, smartfony, monitory i telewizory z płaskim ekranem oraz internet. Komputery mają ułatwiać nam życie, umożliwiać szybsze i skuteczniejsze wykonywanie zadań oraz eliminować wszelkie papierowe formularze i raporty.

Teraz przejdźmy do pierwszej dekady naszego wieku. Mamy olbrzymie telewizory z płaskim ekranem. Wiele użytkowników zamiast aparatów i kamer wideo używa smartfonów. Moim zdaniem najbardziej fantastyczną rzeczą jest moc obliczeniowa, jaką mamy dostępną dosłownie na wyciągnięcie ręki. W przeszłości megabajty pamięci znajdowały się w komputerach, które zajmowały całe klimatyzowane pomieszczenia. Wydawało się, że pamięć na poziomie gigabajtów stanie się dostępna dopiero za kilkadziesiąt lat. Obecnie użytkowników nie dziwi pamięć na poziomie *terabajtów*. Można kupić smartfony o takiej pamięci. Terabajty. Tak, pokonaliśmy długą drogę. To zdumiewające.

Dlaczego wspominam o rozwoju technologii? Ponieważ wpłynęło to na systemy SZRBD i zapewniło podstawy do rozwoju oraz stosowania systemów nierelacyjnych.

W przeszłości producenci oprogramowania bazodanowego mieli problemy z implementowaniem baz relacyjnych, co wynikało z określonych aspektów tego modelu. Początkowo trudno było na przykład zaimplementować kwerendy dotyczące wielu tabel. Przetworzenie i wyświetlenie wyników kwerendy mogło wymagać długiej pracy, jeśli dane pochodziły z wielu tabel — zwłaszcza gdy te tabele zawierały dużo rekordów. Ponadto z tych samych powodów czasochłonne było drukowanie raportów bazujących na takich kwerendach. Jednak postęp technologiczny w obszarach przetwarzania pamięci oraz szybkości odczytu i zapisu danych na dyskach znacznie zredukował wspomniane problemy. Te postępy pozwoliły też zwiększyć solidność i skalowalność systemów SZRBD oraz zwiększyć integralność przechowywanych w nich danych.

W ostatnich latach pojawiła się potrzeba przechowywania danych i obiektów, które nie wpasowują się dobrze w strukturę tabela – pole – rekord z modelu relacyjnego. Na przykład zdjęcia, dane tylko do odczytu z aplikacji sieciowych, dane grafowe, dane geoprzestrzenne i dane analityczne trudno przechowywać za pomocą tego modelu. Rozwój technologii zapewnił użytkownikom podstawowe narzędzia, które umożliwiają tworzenie nowych typów baz danych i systemów zarządzania nimi do przechowywania wspomnianych rodzajów danych. W przeszłości opracowanie takich rozwiązań było prawie niemożliwe. Przykładowe systemy bazodanowe nowego typu to między innymi MongoDB, Couchbase, HBase, Cassandra i Redis.

Wcześniej wspomniałem, że model relacyjny ma 50 lat. Oczekuję, że za drugie 50 lat wciąż będzie używany. Dlaczego mam tak optymistyczne nastawienie? Ponieważ bazy relacyjne są używane wszędzie — są wszechobecne w naszym życiu. Występują w rozmaitych miejscach — od systemów w małych firmach, przez wewnętrzne systemy współpracy, po systemy na poziomie korporacji. Są używane w komputerach osobistych, sieciach firmowych, a nawet w urządzeniach przenośnych. Relacyjne bazy danych są łatwe w użyciu i konserwacji, świetnie nadają się do zapewniania integralności danych, mają solidne struktury (o ile są poprawnie zbudowane), dobrze się skalują i gdy potrzebna jest baza danych, zwykle można użyć właśnie modelu relacyjnego. Należy jednak pamiętać, że relacyjne bazy danych nie są uniwersalnym rozwiązaniem.

Innym źródłem mojego optymizmu jest ten cytat z artykułu „Best Relational Database” z 14 sierpnia 2019 roku z witryny *Database Trends and Applications*: „Według Craiga S. Mullinsa, prezesa i głównego konsultanta w Mullins Consulting, Inc., bazy relacyjne nadal dominują na rynku. IDC prognozuje, że w 2022 roku relacyjne bazy danych wciąż będą stanowić ponad 80% używanych baz. Z kolei Gartner przewiduje, że w 2020 roku technologia relacyjna będzie używana w przynajmniej 70% nowych aplikacji i projektów”. Nawet jeśli te prognozy w kolejnych 10 latach się nie zmienią, bazy relacyjne będą zajmować dominującą pozycję na rynku.

Wszystkiego najlepszego, bazy relacyjne!

Podsumowanie

Rozpocząłem ten rozdział od zdefiniowania pojęcia *baza danych* i opisanie dwóch typów baz danych wykorzystywanych obecnie w zarządzaniu bazami: operacyjnych i analitycznych.

Następnie szczegółowo omówiłem model relacyjnych baz danych, ich historię i cechy. Podkreśliłem, że ten model bazuje na gałęziach matematyki oraz że to właśnie matematyczne podstawy sprawiają, iż model ten jest tak solidny strukturalnie. Następnie opisałem strukturę danych oraz zależności występujące w tym modelu oraz rolę języka SQL w dostępie do danych. Dowiedziałeś się, że SQL to standardowy język wykorzystywany do pracy z relacyjnymi bazami danych. Tę część rozdziału zakończyło podsumowanie zalet modelu relacyjnych baz danych i omówienie, czym jest SZRBD. Dowiedziałeś się również, jak postępy w technologiach wpłynęły na nasze codzienne życie i jak wszechobecne są nowe rozwiązania. Opisałem też, dlaczego relacyjne bazy danych są używane już od 50 lat i że systemy SZRBD wykorzystuje się w różnego rodzaju scenariuszach biznesowych.

W następnym rozdziale wyjaśniam, dlaczego projektowanie baz danych powinno nas interesować oraz dlaczego teoria jest ważna. Omawiam też wady i zalety dobrego projektu.

Pytania kontrolne

1. Wymień dwa typy wykorzystywanych obecnie baz danych.
2. Jakiego typu dane przechowuje analityczna baza danych?
3. Prawda czy fałsz: Operacyjna baza danych jest wykorzystywana głównie do przetwarzania transakcji online (OLTP).
4. Wymień jedną z gałęzi matematyki, na której opiera się model relacyjny.
5. W jaki sposób relacyjna baza danych przechowuje dane?
6. Wymień trzy typy zależności w relacyjnej bazie danych.
7. W jaki sposób pozyskuje się dane z relacyjnej bazy danych?
8. Wymień dwie zalety relacyjnej bazy danych.
9. Czym jest system zarządzania relacyjną bazą danych?
10. Prawda czy fałsz: Pamięć w urządzeniach przenośnych jest ograniczona do poziomu gigabajtów.
11. Wyjaśnij, dlaczego producenci oprogramowania bazodanowego mieli trudności z implementowaniem relacyjnych baz 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 

Nie ma dobrej bazy danych bez dobrego projektu!

Mimo upływu lat relacyjne bazy danych wciąż mają się świetnie! Z każdym rokiem są coraz doskonalsze i radzą sobie z coraz większymi zbiorami danych. Wciąż jednak podstawą dobrej aplikacji bazodanowej jest dobry projekt samej bazy. Wielu osobom projektowanie poprawnych struktur bazodanowych wydaje się czymś z pogranicza wyższej matematyki i czarnej magii. Tymczasem zdobycie tej umiejętności jest możliwe bez lat studiowania skomplikowanych teorii matematycznych. Wystarczy przyswoić kluczowe podstawy i nauczyć się korzystać z kilku zdroworozsądkowych koncepcji i teorii.

Ta książka jest rocznicowym, przejrzanym i zaktualizowanym wydaniem kultowego podręcznika do samodzielnej nauki projektowania relacyjnych baz danych. Zawarte w nim informacje można wykorzystywać niezależnie od zastosowanego oprogramowania. Przedstawiono tu, jak projektować nowoczesne bazy danych, które mają poprawną strukturę, są niezawodne i ułatwiają wprowadzanie zmian. Opisano wszystkie etapy projektowania: od planowania po definiowanie tabel, pól, kluczy, relacji między tabelami, reguł biznesowych i widoków. Dodatkowo znajdziemy tu praktyczne techniki zwiększania integralności danych, omówienie często popełnianych błędów i wskazówki, kiedy warto łamać zasady. Treść przewodnika wzbogacają pytania kontrolne i rysunki, które bardzo pomagają w jej skutecznym opanowaniu.

W KSIĄŻCE:

- ▶ typy baz danych, modele i cele projektowe
- ▶ tworzenie tabel i relacji, specyfikacje pól i widoki
- ▶ poziomy integralności danych
- ▶ tworzenie reguł biznesowych
- ▶ perspektywy relacyjnych baz danych

MICHAEL J. HERNANDEZ

to człowiek renesansu. Jest niezależnym konsultantem, uznanym szkoleniowcem, prelegentem na konferencjach i autorem licznych publikacji. Pracował dla poważnych korporacji, rządu, wojska i wielu przedsiębiorstw. Nigdy się nie nudzi i cały czas rozwija swoje rozliczne pasje: wróży z tarota, gra na gitarze, a nawet para się aktorstwem.

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

Sprawdź nasze szkolenia!
SZKOLENIA
AKADEMIA IT & BUSINESS
HELIONSZKOLENIA.PL

KOD KORZYŚCI
Śięgnij po więcej! ▶



ISBN 978-83-283-8251-0
9 788328 382510
Cena: 99,00 zł

Pearson
Addison-Wesley