

Michael J. Hernandez

Słowo wstępne: Ken Getz



PROJEKTOWANIE BAZ DANYCH

DLA KAŻDEGO

PRZEWODNIK KROK PO KROKU

Praktyczny przewodnik dla projektantów baz danych!



Tytuł oryginału: Database Design for Mere Mortals:
A Hands-On Guide to Relational Database Design (3rd Edition)

Tłumaczenie: Katarzyna Żarnowska (wstęp, rozdz. 1 – 6, 8, 9),
Radosław Meryk (rozdz. 7, 13 – 15, dodatki), Ireneusz Jakóbiak (rozdz. 10 – 12)

ISBN: 978-83-246-7995-9

Authorized translation from the English language edition, entitled: DATABASE DESIGN FOR MERE MORTALS: A HANDS-ON GUIDE TO RELATIONAL DATABASE DESIGN, Third Edition; ISBN 0321884493; by Michael J. Hernandez; published by Pearson Education, Inc, publishing as Addison Wesley. Copyright © 2013 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 © 2014.

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.

Wydawnictwo HELION dołożyło wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

Wydawnictwo HELION

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/projbd>

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	15
Słowo wstępne	17
Do wydania trzeciego	17
Z wydania drugiego...	17
Z wydania pierwszego...	18
Przedmowa	19
Podziękowania	21
Wprowadzenie	23
Co nowego w trzecim wydaniu	25
Kto powinien przeczytać tę książkę	25
Cel niniejszej książki	26
Jak czytać tę książkę	27
Organizacja książki	28
Część I: Projektowanie relacyjnych baz danych	28
Część II: Proces projektowania	28
Część III: Inne problemy projektowania baz danych	29
Część IV: Dodatki	29
Słowo na temat przykładów i technik opisywanych w tej książce	30
Nowe podejście do nauki	30
Część I Projektowanie relacyjnych baz danych	33
Rozdział 1. Relacyjna baza danych	35
Tematy omówione w tym rozdziale	35
Rodzaje baz danych	36
Wczesne modele baz danych	36
Hierarchiczny model bazy danych	37
Sieciowy model baz danych	39

Model relacyjnych baz danych	41
Pozyskiwanie danych	42
Zalety relacyjnych baz danych	44
Zarządzanie relacyjną bazą danych	45
Poza modelem relacyjnym	46
Co niesie przyszłość	47
Ostatnia uwaga	48
Podsumowanie	48
Pytania kontrolne	49
Rozdział 2. Cele projektowania	51
Tematy omówione w tym rozdziale	51
Dlaczego projektowanie baz danych powinno nas interesować?	51
Znaczenie teorii	53
Zalety poznania dobrej metodologii projektowania	54
Cele dobrego projektowania	55
Korzyści wynikające z dobrego projektowania	55
Metody projektowania baz danych	56
Tradycyjne metody projektowania	56
Metoda projektowania zaprezentowana w tej książce	57
Normalizacja	58
Podsumowanie	60
Pytania kontrolne	61
Rozdział 3. Terminologia	63
Tematy omówione w tym rozdziale	63
Dlaczego terminologia jest ważna	64
Pojęcia związane z wartością	64
Dane	64
Informacje	65
Null	66
Wartość znaczników null	67
Problem ze znacznikami null	68
Pojęcia związane ze strukturą	69
Tabele	69
Pole	71
Rekord	72
Widok (perspektywa)	73
Klucze	74
Indeks	76

Pojęcia związane z zależnościami	76
Zależności	76
Typy zależności	77
Rodzaje udziału	80
Stopień udziału	81
Pojęcia związane z integralnością	82
Specyfikacja pola	82
Integralność danych	82
Podsumowanie	83
Pytania kontrolne	84

Część II Proces projektowania 87

Rozdział 4. Przegląd koncepcyjny 89

Tematy omówione w tym rozdziale	89
Dlaczego ważna jest realizacja całego procesu projektowania	90
Formułowanie definicji celu i założeń wstępnych	91
Analiza istniejącej bazy danych	91
Tworzenie struktur danych	92
Określanie i ustalanie relacji w tabelach	93
Określanie reguł biznesowych	93
Definiowanie widoków	94
Kontrola integralności danych	94
Podsumowanie	95
Pytania kontrolne	96

Rozdział 5. Rozpoczęcie procesu projektowania 99

Tematy omówione w tym rozdziale	99
Przeprowadzanie wywiadów	100
Wytyczne dotyczące rozmówców	101
Wytyczne dotyczące osoby przeprowadzającej wywiad	102
Formułowanie definicji celu	106
Poprawnie sformułowana definicja celu	106
Układanie definicji celu	107
Formułowanie założeń wstępnych	109
Poprawnie sformułowane założenia wstępne	109
Układanie założeń wstępnych	111
Podsumowanie	114
Pytania kontrolne	114

Rozdział 6. Analiza istniejącej bazy danych	117
Tematy omówione w tym rozdziale	117
Poznanie istniejącej bazy danych	118
Papierowe bazy danych	120
Spadkowe bazy danych	120
Przeprowadzenie analizy	121
Spojrzenie na sposób gromadzenia danych	121
Spojrzenie na sposób prezentowania informacji	124
Przeprowadzanie wywiadów	127
Podstawowe techniki przeprowadzania wywiadów	127
Zanim rozpoczniesz przeprowadzanie wywiadów... ..	132
Wywiady z użytkownikami	132
Przegląd typów danych i sposobów ich wykorzystania	132
Przegląd próbek	134
Przegląd wymagań informacyjnych	137
Wywiady z kierownictwem	143
Przegląd obecnych wymagań informacyjnych	143
Przegląd dodatkowych wymagań informacyjnych	144
Przegląd przyszłych wymagań informacyjnych	144
Przegląd ogólnych wymagań informacyjnych	145
Stworzenie kompletnej listy pól	145
Wstępna lista pól	145
Lista pól obliczeniowych	150
Przegląd obu list wraz z pracownikami i kierownictwem	151
Podsumowanie	155
Pytania kontrolne	156
Rozdział 7. Tworzenie struktur tabel	159
Tematy omówione w tym rozdziale	159
Definiowanie wstępnej listy tabel	160
Identyfikacja domniemanych podmiotów	160
Korzystanie z listy podmiotów	161
Korzystanie z celów misji	165
Definiowanie ostatecznej listy tabel	167
Dostrajanie nazw tabel	168
Wskazywanie typów tabel	172
Redagowanie opisów tabel	172
Powiązanie pól z każdą z tabel	177

Dostrajanie pól	179
Poprawianie nazw pól	179
Korzystanie z idealnego pola do eliminowania anomalii	182
Eliminacja pól wieloczęściowych	185
Eliminacja pól wielowartościowych	186
Dostrajanie struktur tabel	192
Kilka słów o nadmiarowych danych i duplikatach pól	192
Wykorzystanie warunków idealnej tabeli w celu dostrojenia struktur tabel	193
Wyznaczanie tabel-podzbiorów	198
Podsumowanie	208
Pytania kontrolne	209
Rozdział 8. Klucze	211
Tematy omówione w tym rozdziale	211
Dlaczego klucze są ważne	212
Definiowanie kluczy dla tabel	212
Klucze kandydujące	212
Klucze główne	218
Klucze zastępcze	222
Pola niekluczowe	223
Integralność na poziomie tabeli	223
Przegląd wstępnych struktur tabel	224
Podsumowanie	229
Pytania kontrolne	230
Rozdział 9. Specyfikacje pól	231
Tematy omówione w tym rozdziale	231
Dlaczego specyfikacje pól są ważne	232
Integralność na poziomie pól	233
Anatomia specyfikacji pól	233
Elementy ogólne	234
Elementy fizyczne	239
Elementy logiczne	244
Wykorzystywanie unikatowych, ogólnych i replikowanych specyfikacji pól ...	250
Definiowanie specyfikacji pól dla każdego pola w bazie danych	255
Podsumowanie	256
Pytania kontrolne	259

Rozdział 10. Relacje między tabelami 261

Tematy omówione w tym rozdziale	261
Dlaczego relacje są ważne	262
Rodzaje relacji	263
Relacja jeden-do-jednego	264
Relacja jeden-do-wielu	265
Relacja wiele-do-wielu	267
Relacja zwrotna	273
Identyfikowanie istniejących relacji	276
Ustanawianie wszystkich relacji	284
Relacje jeden-do-jednego i jeden-do-wielu	284
Relacja wiele-do-wielu	290
Relacje zwrotne	294
Sprawdzanie struktury wszystkich tabel	298
Dokładna analiza wszystkich kluczy obcych	299
Ustanawianie charakterystyk relacji	304
Definiowanie reguły usuwania dla każdej relacji	304
Identyfikowanie rodzaju udziału każdej z tabel	308
Identyfikowanie stopnia udziału każdej z tabel	310
Weryfikowanie z użytkownikami i zarządem relacji istniejących między tabelami	312
Uwaga końcowa	312
Integralność na poziomie relacji	313
Podsumowanie	317
Pytania kontrolne	318

Rozdział 11. Reguły biznesowe 321

Tematy omówione w tym rozdziale	321
Czym są reguły biznesowe?	321
Rodzaje reguł biznesowych	324
Kategorie reguł biznesowych	326
Reguły biznesowe specyficzne dla pól	326
Reguły biznesowe specyficzne dla relacji	327
Definiowanie i ustanawianie reguł biznesowych	328
Praca z użytkownikami oraz zarządem	328
Definiowanie i ustanawianie reguł biznesowych specyficznych dla pola	329
Definiowanie i ustanawianie reguł biznesowych specyficznych dla relacji	334

Tabele walidacji	341
Czym są tabele walidacji?	341
Korzystanie z tabel walidacji w celu realizowania reguł biznesowych	342
Sprawdzanie arkuszy specyfikacji reguł biznesowych	346
Podsumowanie	352
Pytania kontrolne	353
Rozdział 12. Widoki	355
Tematy omówione w tym rozdziale	355
Czym są widoki?	355
Anatomia widoku	357
Widok danych	357
Widok zagregowany	361
Widok walidacji	364
Określanie i definiowanie widoków	366
Praca z użytkownikami i zarządem	366
Identyfikowanie widoków	367
Przeglądanie dokumentacji każdego widoku	373
Podsumowanie	378
Pytania kontrolne	380
Rozdział 13. Sprawdzanie integralności danych	383
Tematy omówione w tym rozdziale	383
Dlaczego należy sprawdzać integralność danych?	384
Sprawdzanie i korygowanie integralności danych	384
Integralność na poziomie tabel	385
Integralność na poziomie pól	385
Integralność na poziomie relacji	385
Reguły biznesowe	386
Widoki	386
Kompletowanie dokumentacji bazy danych	387
W końcu zrobione!	388
Podsumowanie	388
Część III Inne problemy projektowania baz danych	389
Rozdział 14. Czego nie należy robić?	391
Tematy omówione w tym rozdziale	391
Płaskie pliki	392

Projekt na bazie arkusza kalkulacyjnego	393
Rozwiązywanie problemów związanych z przyzwyczajeniami do widoku arkusza kalkulacyjnego	394
Projekt bazy danych pod kątem konkretnego oprogramowania	396
Wnioski końcowe	397
Podsumowanie	397
Rozdział 15. Naginanie bądź łamanie reguł	399
Tematy omówione w tym rozdziale	399
Kiedy można nagiąć lub złamać reguły?	399
Projektowanie analitycznej bazy danych	399
Poprawianie wydajności obliczeń	400
Dokumentowanie działań	402
Podsumowanie	403
Na zakończenie	405
Dodatki	407
Dodatek A Odpowiedzi na pytania kontrolne	409
Dodatek B Diagram procesu projektowania baz danych	427
Dodatek C Wytyczne projektowe	445
Definiowanie i wprowadzanie reguł biznesu specyficznych dla pól	445
Definiowanie i wprowadzanie reguł biznesu specyficznych dla relacji	445
Warunki klucza kandydującego	446
Warunki klucza obcego	446
Warunki klucza głównego	446
Reguły tworzenia kluczy głównych	447
Warunki idealnego pola	447
Warunki idealnej tabeli	447
Integralność na poziomie pól	448
Wytyczne tworzenia opisów pól	448
Wytyczne tworzenia opisów tabel	448
Wytyczne tworzenia nazw pól	449
Wytyczne tworzenia nazw tabel	449
Identyfikowanie relacji	450
Identyfikacja wymagań dotyczących perspektyw	450
Wytyczne dotyczące prowadzonych rozmów	451
Wskazówki związane z uczestnikami	451
Wskazówki dotyczące prowadzącego rozmowę	451

Misje	451
Cele misji	452
Integralność na poziomie relacji	452
Eliminowanie pól wielowartościowych	452
Integralność na poziomie tabel	453
Dodatek D Formularze dokumentacyjne	455
Dodatek E Symbole używane w diagramach stosowanych w procesie projektowania baz danych	459
Dodatek F Przykładowe projekty	461
Dodatek G O normalizacji	467
Uwaga...	467
Krótkie przypomnienie	468
W jaki sposób normalizacja jest zintegrowana z moją metodologią projektowania?	471
Projekt logiczny a projekt fizyczny i implementacja	473
Dodatek H Zalecana lektura	475
Słowniczek	477
Literatura	489
Skorowidz	491

1

Relacyjna baza danych

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

Tematy omówione w tym rozdziale

- Rodzaje baz danych
- Wczesne modele baz danych
- Model relacyjnych baz danych
- Zarządzanie relacyjną bazą danych
- Poza modelem relacyjnym
- Co niesie przyszłość
- Podsumowanie
- Pytania kontrolne

Relacyjne bazy danych istnieją od ponad 40 lat. Ten najbardziej rozpowszechniony na świecie typ baz danych niezbędnych w naszym codziennym życiu rozkręcił branżę wartą miliony 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 lub wypożyczasz książki.

Zanim zagłębimy się w proces projektowania, przyjrzyjmy się krótkiej historii relacyjnych baz danych — skąd się wzięły, gdzie są teraz i w jakim kierunku podążają.

Rodzaje baz 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 papieru, czy aplikacji komputerowej. Jeśli tylko zbierasz dane w zorganizowany sposób i w konkretnym celu, masz bazę danych. W dalszej części tej książki przyjmiemy, że do zbierania i przechowywania danych będziemy wykorzystywać 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 internetowych (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, wytwórnie 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 atutem, jeśli potrzebujemy 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.

Wczesne modele baz danych

Zanim narodził się model relacyjnych baz danych, w powszechnym użyciu były dwa inne modele, wykorzystywane do utrzymania danych i wykonywania na nich operacji — *model hierarchiczny* oraz *model sieciowy*.

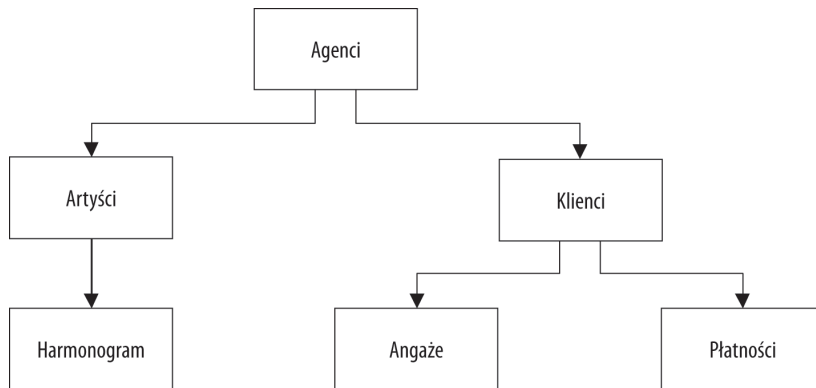
Niektóre pojęcia, z którymi zetkniesz się w tym rozdziale, zostały szczegółowo wyjaśnione w rozdziale 3.

Uwaga Krótki opis każdego z tych modeli przedstawiony jest tu tylko ze względów historycznych. Uważam, że warto wiedzieć, co poprzedziło model relacyjny, aby lepiej zrozumieć, co doprowadziło do jego powstania i ewolucji.

W tym przeglądzie opisuję pokrótce strukturę danych w każdym modelu, dostęp do nich, a także prezentuję zależności pomiędzy dwiema tabelami i kilka wad oraz zalet każdego z modeli.

Hierarchiczny model bazy danych

Dane w tym typie bazy mają strukturę hierarchiczną, a typowy diagram ma kształt odwróconego drzewa. Pojedyncza tabela w bazie danych pełni rolę korzeni odwróconego drzewa, a inne tabele są gałęziami wyrastającymi z tych korzeni. Rysunek 1.1 pokazuje diagram typowej struktury hierarchicznej bazy danych.



Rysunek 1.1. Diagram typowej hierarchicznej bazy danych

Baza danych agentów

W przykładzie pokazanym na rysunku 1.1 agent zajmuje się rezerwacjami kilku artystów estradowych, a każdy artysta ma swój harmonogram. Agent ma również kilku klientów, których potrzebami musi się zająć. Klient rezerwuje występ poprzez agenta i płaci mu za usługę.

Zależność w hierarchicznej bazie danych reprezentowana jest przez pojęcia *rodzic* i *dziecko*. W tym typie zależności tabela-rodzic może być powiązana z jedną lub kilkoma tabelami-dziećmi, ale pojedyncza tabela-dziecko może być powiązana tylko z jedną tabelą-rodzicem. Te tabele są wyraźnie ze sobą połączone poprzez wskaźnik lub fizyczne uporządkowanie rekordów w tabeli. W tym modelu użytkownik dostaje się do danych, zaczynając od głównej tabeli, a kierując się w dół hierarchii, dociera do danych docelowych. Taka metoda dostępu wymaga od użytkownika obeznania ze strukturą bazy danych.

Jedną z zalet korzystania z hierarchicznej bazy danych jest to, że użytkownik może bardzo sprawnie uzyskać dane, ponieważ wewnątrz struktury tabeli istnieją wyraźne połączenia. Kolejną zaletą jest wbudowana integralność odniesień, która jest wykonywana automatycznie. Dzięki temu mamy pewność, że rekord w tabeli-dziecku będzie połączony z istniejącym rekordem w tabeli-rodzicu oraz że usunięty rekord w tabeli-rodzicu spowoduje usunięcie wszystkich powiązanych z nim rekordów w tabeli-dziecku.

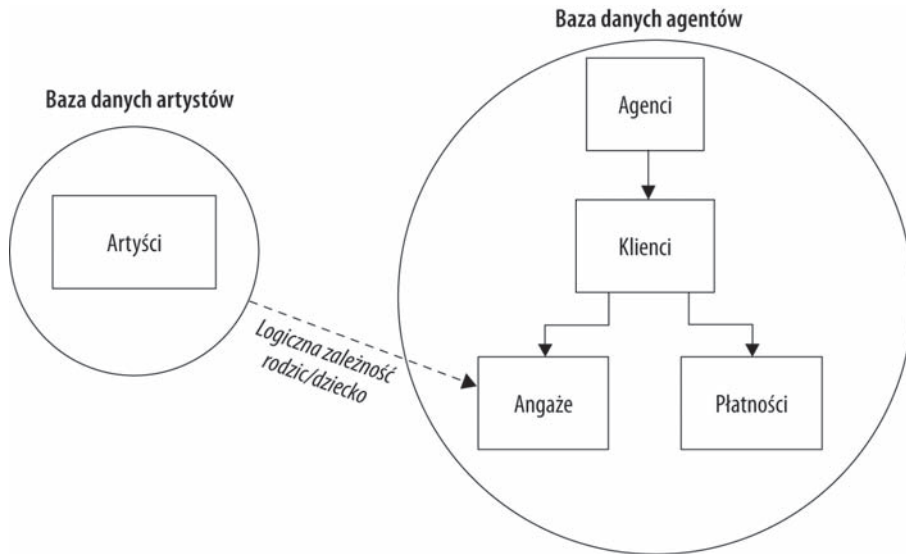
Problem w hierarchicznej bazie danych pojawia się, kiedy użytkownik potrzebuje przechować w tabeli-dziecku rekord, który w danym momencie nie jest połączony z żadnym rekordem z tabeli-rodzica. Przyjrzyj się przykładowi wykorzystania bazy danych agentów z rysunku 1.1. Użytkownik nie może wprowadzić danych nowego artysty do tabeli *Artyści*, dopóki ten artysta nie zostanie przypisany do agenta z tabeli *Agenci*. Przypomnij sobie, że rekord w tabeli-dziecku (w tym przypadku w tabeli *Artyści*) musi być powiązany z rekordem w tabeli-rodzicu (*Agenci*). Jednak w prawdziwym życiu artyści zazwyczaj najpierw zapisują się do agencji, a dopiero później zostaje im przypisany agent. Taki scenariusz jest trudny do wprowadzenia w hierarchicznej bazie danych. Możemy obejść te zasady, zakładając fikcyjny rekord w tabeli *Agenci*. Taka opcja nie jest niestety optymalna.

Ten typ baz danych nie wspiera złożonych zależności i często generuje problemy związane ze zbędnymi danymi. Na przykład pomiędzy klientami i artystami istnieje zależność „wiele do wielu” — artysta pracuje dla wielu klientów, a klient zatrudnia wielu artystów. Ten typ zależności nie może być modelowany w hierarchicznej bazie danych, należy więc wprowadzić dodatkowe dane zarówno do tabeli *Harmonogram*, jak i *Angaże*.

1. Tabela *Harmonogram* będzie teraz zawierała dane klienta (takie jak nazwisko, adres i numer telefonu), by pokazać, dla kogo i gdzie będzie występował artysta. Te dane są tutaj zbędne, ponieważ są one przechowywane w tabeli *Klienci*.
2. Tabela *Angaże* będzie teraz zawierała dane artystów (takie jak nazwisko, numer telefonu, rodzaj działalności artysty), aby pokazać, którzy artyści pracują dla których klientów. Te dane również są zbędne, ponieważ są aktualnie przechowywane w tabeli *Artyści*.

Problem z dodatkowymi danymi polega na tym, że możliwy jest brak konsekwencji we wprowadzaniu danych. To z kolei prowadzi do tworzenia niedokładnych informacji.

Użytkownik może obejść ten problem, tworząc jedną hierarchiczną bazę danych specjalnie dla artystów, a drugą specjalnie dla agentów. Nowa baza danych dla artystów będzie się składała tylko z tabeli *Artyści*, a zaktualizowana baza agentów będzie zawierała tabele *Agenci*, *Klienci*, *Płatności* oraz *Angaże*. Tabela *Harmonogram* nie będzie już potrzebna w bazie danych artystów, ponieważ można określić logiczną zależność (tabela-dziecko) pomiędzy tabelą *Angaże* w bazie danych agentów a tabelą *Artyści* w bazie danych artystów. Mając tę zależność ustaloną, jesteś w stanie wyszukać różnorodne informacje, na przykład listę zamówionych przez danego klienta artystów lub harmonogram występów dla danego artysty. Rysunek 1.2 pokazuje diagram nowego modelu.



Rysunek 1.2. Wykorzystywanie hierarchicznej bazy danych do rozwiązywania problemów z zależnością „wiele do wielu”

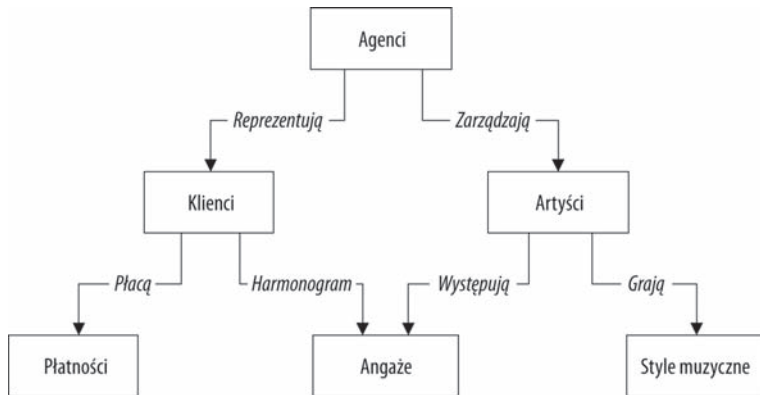
Jak łatwo zauważyć, osoba projektująca hierarchiczną bazę danych musi być zdolna do rozpoznania potrzeby wykorzystania tej techniki dla zależności typu „wiele do wielu”. Tutaj potrzeba ta była w miarę oczywista, ale wiele zależności jest znacznie bardziej niejasnych i mogą zostać zidentyfikowane dopiero na bardzo późnym etapie projektowania lub, co gorsza, już po oddaniu bazy danych do użytku.

Hierarchiczne bazy danych były bardzo popularne i sprawdziły się dobrze w systemach pamięci taśmowej wykorzystywanej przez komputery mainframe w latach 70. Jednak mimo że zapewniały szybki i bezpośredni dostęp do danych i w wielu przypadkach okazały się bardzo przydatne, potrzebny był nowy model baz danych, który rozwiązałby rosnący problem zbędnych danych oraz złożonych zależności pomiędzy nimi.

Sieciowy model baz danych

Sieciowe bazy danych zostały stworzone głównie w celu rozwiązania niektórych problemów istniejących w hierarchicznych bazach danych. Ich struktura reprezentowana jest przez węzły oraz zestawy struktur. Rysunek 1.3 pokazuje diagram typowej sieciowej bazy danych.

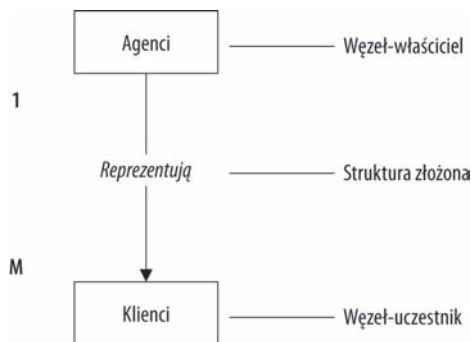
Węzeł przedstawia kolekcję rekordów, a zestaw struktur ustala i prezentuje zależność w sieciowej bazie danych. Wzajemny stosunek pary węzłów prezentuje przejrzysta konstrukcja, pokazując jeden węzeł jako właściciela (ang. *owner*), a drugi jako uczestnika (ang. *member*), co jest znacznym postępowaniem w stosunku do zależności rodzic – dziecko. Zestaw struktur wspiera zależność „jeden do wielu”, co oznacza, że rekord w węźle-właścicielu może być powiązany z jednym lub wieloma rekordami w węźle-uczestniku,



Rysunek 1.3. Diagram typowej sieciowej bazy danych

ale pojedynczy rekord w węźle-uczestniku może być powiązany tylko z jednym rekordem w węźle-właścicielu. Dodatkowo rekord w węźle-uczestniku nie może istnieć bez odwołania do istniejącego rekordu w węźle-właścicielu. Na przykład klient musi być przypisany do agenta, ale agent może wciąż występować w bazie danych bez przypisanych klientów. Rysunek 1.4 pokazuje diagram prostego zestawu struktur.

Jeden lub kilka zestawów (połączeń) może być zdefiniowanych pomiędzy konkretną parą węzłów, a pojedynczy węzeł może być powiązany również z innymi węzłami w bazie danych. Na rysunku 1.3 na przykład węzeł Klienci jest powiązany z węzłem Płatności poprzez strukturę Płacą. Jest on także powiązany z węzłem Angaże poprzez strukturę Występują. Oprócz odnoszenia się do węzła Klienci węzeł Angaże łączy się także z węzłem Artyści za pomocą struktury Występują.



Rysunek 1.4. Podstawowa struktura złożona

W sieciowej bazie danych użytkownik może uzyskać dostęp do danych poprzez odpowiednie struktury złożone. Inaczej jest w hierarchicznej bazie danych, gdzie dostęp można uzyskać poprzez główną tabelę. Tutaj użytkownik ma dostęp do danych z bazy sieciowej, a zacząć może od dowolnego węzła i poruszać się do tyłu lub do przodu

poprzez powiązane zestawy. Przyjrzyjmy się jeszcze raz bazie danych agentów na rysunku 1.3. Przyjmijmy, że użytkownik chce znaleźć agenta, który dokonał rezerwacji określonego występu. Użytkownik zaczyna od zlokalizowania odpowiedniego rekordu występu w węźle Angaże, a następnie poprzez strukturę Występują określa, który agent „zarządza” rekordem tego występu. W końcu użytkownik poprzez strukturę Reprezentują odnajduje odpowiedniego agenta, który „zarządza” rekordem klienta. Użytkownik może znaleźć odpowiedź na wiele pytań, o ile potrafi poprawnie nawigować w odpowiednich strukturach.

Zaletą sieciowej bazy danych jest szybki dostęp do danych. Pozwala ona również użytkownikom na tworzenie bardziej skomplikowanych kwerend, niż byłoby to możliwe w przypadku hierarchicznej bazy danych. Główną wadą sieciowej bazy danych jest to, że aby móc poruszać się w strukturach, użytkownik musi być dobrze zaznajomiony z bazą. Jeszcze raz przyjrzyjmy się bazie danych agentów z rysunku 1.3. To na użytkownika spoczywa obowiązek zaznajomienia się z zestawami struktur, jeśli chce sprawdzić, czy opłata za określony występ została wniesiona. Kolejną wadą jest trudność w takim wprowadzaniu zmian do struktury bazy danych, aby nie wpływało to negatywnie na aplikacje, które z nią współpracują. Przypominam, że w sieciowej bazie danych zależność zdefiniowana jest jako zestaw struktur. Nie możesz zmieniać zestawu struktur bez wpływania na aplikacje, które wykorzystują te struktury do poruszania się pomiędzy danymi. Jeśli zmienisz zestaw struktur, musisz również zmodyfikować wszystkie powiązania tej struktury z zewnętrznymi aplikacjami.

Mimo iż sieciowe bazy danych były krokiem naprzód w porównaniu z bazami hierarchicznymi, niektórzy wciąż wierzyli, że musi istnieć lepszy sposób na zarządzanie i utrzymywanie dużych ilości danych. W miarę pojawiania się kolejnych modeli danych użytkownicy przekonywali się, że mogą zadawać coraz bardziej złożone pytania, zwiększając tym samym wymagania stawiane przed bazami danych. W ten sposób doszliśmy do modelu relacyjnych baz danych.

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 rozwiązywania problemów takich jak zbędne dane, ich słaba integralność oraz zbytnia zależność struktur baz danych od ich fizycznej implementacji.

Dr Codd oficjalnie przedstawił nowy model relacyjny w swojej książce zatytułowanej *A Relational Model of Data for Large Shared Databanks*¹ w czerwcu 1970 roku. Swój

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

nowy 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 niematerialny, a każdy rekord w tabeli jest możliwy do zidentyfikowania poprzez pole zawierające unikatową wartość. To są dwie cechy 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. Różni się to zupełnie od modeli hierarchicznych oraz sieciowych, w których znajomość struktur była niezbędna do wydobycia danych z bazy.

Model relacyjny kategoryzuje zależności na „jeden do jednego”, „jeden do wielu” oraz „wiele do wielu”. Te zależności zostaną szczegółowo omówione w rozdziale 10. Zależność pomiędzy parą tabel jest bezwzględnie ustalona przez pasujące wartości wspólnego pola. Na rysunku 1.5 na przykład tabele *Klienci* i *Agenci* są ze sobą powiązane poprzez pole *Nr agenta*. Konkretny klient jest powiązany z agentem poprzez pasujący *Nr agenta*. W ten sam sposób tabele *Artyści* oraz *Angaże* powiązane są ze sobą poprzez *Nr artysty*. Rekord w tabeli *Artyści* może być powiązany z rekordem w tabeli *Angaże* poprzez pasujący *Nr artysty*. O ile użytkownik zna się na zależnościach występujących pomiędzy tabelami w bazie danych, może mieć dostęp do danych na niemal nieskończoną ilość sposobów. Może wydobywać dane z tabel, które są ze sobą powiązane bezpośrednio oraz pośrednio. Przyjrzyj się bazie danych agentów na rysunku 1.5. 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. SQL to standardowy język wykorzystywany do tworzenia, modyfikowania, utrzymywania relacyjnej bazy danych oraz tworzenia zapytań. Poniższy przykład pokazuje deklarację SQL, którą możesz wykorzystać do otrzymania listy klientów w mieście Katowice:

```
SELECT Nazwisko klienta, Imię klienta, Telefon klienta
FROM Klienci
WHERE City = "Katowice"
ORDER BY Nazwisko klienta, Imię klienta
```

Agenci

Nr agenta	Imię agenta	Nazwisko agenta	Data zatrudnienia	Numer telefonu agenta
100	Natalia	Wirska	16-05-2011	22 6590051
101	Bartosz	Łucki	15-10-2011	22 6986523
102	Renata	Woj	01-03-2012	32 2958596

Klienci

Nr klienta	Nr agenta	Imię klienta	Nazwisko klienta	Numer telefonu klienta
9001	100	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	3001	01-04-2012	13:00	15:30
9009	3000	13-04-2012	21:00	01:30
9001	3002	02-05-2012	15:00	18:00

Rysunek 1.5. Przykłady tabel powiązanych w relacyjnej bazie danych

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 tabelę, do których należą te pola. Możesz filtrować rekordy, które zwraca kwerenda, poprzez narzucanie kryteriów na jedno lub wiele pól za pomocą warunku WHERE, 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 jest budować kwerendy, wykorzystując narzędzie graficzne. Bez względu na sposób budowy kwerend użytkownik może je zachować do następnego wykorzystania.

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. 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, możesz zacząć od przeczytania innej z moich książek *SQL Queries for Mere Mortals*, a następnie zapoznać się z innymi książkami dotyczącymi SQL, które wymienione są w dodatku H.

Zalety relacyjnych baz danych

Relacyjna baza danych posiada więcej zalet, niż poprzednio omówione. 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 ważna; 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 baz danych.* 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 konsekwencja i dokładność danych.* Dane są podawane konsekwentnie 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 tabel powiązanych w bazie danych przy wykorzystaniu komend 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 zarządzają nimi. 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 zarówno w technologii tworzenia oprogramowania, jak i sprzętu sprawiły, że w ciągu ostatnich 20 lat prędkość przetwarzania przestała mieć znaczenie, a producenci zdołali wygenerować zyski poprzez swe działania służące lepszemu wsparciu modelu.

Więcej na temat modelu relacyjnych baz danych dowiesz się z dalszej części tej książki. Niektóre z poruszonych tematów będą dotyczyły tworzenia tabel, ustalania 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 manipulacji nią. Wiele systemów SZRBD zapewnia także narzędzia niezbędne do tworzenia aplikacji dla użytkownika, które wchodzi w interakcje z danymi przechowywanymi w bazie danych. Oczywiście jakość SZRBD zależy od tego, w jakim stopniu wspiera on model relacyjnych baz danych. Nawet w przypadku „prawdziwych” systemów SZRBD poziom wsparcia dla 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 stają się coraz potężniejsze.

Początkowo systemy SZRBD były przeznaczone do użytku na komputerach mainframe (czy nie wszystkie programy tak zaczynały?). Dwoma najbardziej rozpowszechnionymi systemami SZRBD we wczesnych latach 70. były System R, stworzony przez IBM w San Jose Research Laboratory w Kalifornii, oraz Interactive Graphics Retrieval System (INGRES), stworzony na Uniwersytecie Kalifornijskim w Berkeley. Te dwa programy znacząco przyczyniły się do powszechnego zrozumienia wartości modelu relacyjnych baz danych.

Kiedy z czasem korzyści wynikające z używania relacyjnych baz danych stały się bardziej oczywiste, wiele firm zdecydowało się na ostrożną wymianę hierarchicznych i sieciowych modeli baz danych na relacyjne, tworząc w ten sposób zapotrzebowanie na lepsze systemy SZRBD dla komputerów mainframe. W latach 80. dla tych komputerów komercyjne oprogramowanie zaczęły produkować firmy takie jak Oracle i IBM.

Na początku i w połowie lat 80. nastąpił znaczny wzrost użytkowania komputerów osobistych, a wraz z tym rozwój systemów SZRBD dla tych komputerów. Pierwsze próby

w tej kategorii, podejmowane między innymi przez Ashton-Tate oraz Fox Software, nie różniły się niczym od prostego systemu zarządzania bazą danych opartego na plikach. Prawdziwe systemy SZRBD dla komputerów PC zaczęły dopiero produkować firmy takie jak Microrim oraz Ansa Software. Te firmy rozpowszechniły ideę oraz potencjał zarządzania bazami danych i pomogły im przejść ze zdominowanych przez komputery mainframe działów zarządzania informacjami do komputerów na biurkach zwykłych użytkowników.

Potrzeba dzielenia się danymi stawała się oczywista, w miarę jak coraz większa liczba użytkowników rozpoczynała pracę z bazami danych na przełomie lat 80. i 90. Koncepcja centralnie zlokalizowanej bazy danych, która mogła być dostępna dla wielu użytkowników, wydawała się bardzo obiecującym pomysłem. Mogło to znacznie uprościć zarządzanie danymi oraz zapewnienie bezpieczeństwa całej bazy. Producenci baz danych, tacy jak Microsoft i Oracle, odpowiedzieli na te potrzeby, wytwarzając systemy SZRBD typu klient-serwer.

W środowisku klient-serwer dane przechowywane są w komputerze, który pełni rolę serwera bazy danych, a użytkownicy wchodzą w interakcje z danymi poprzez aplikacje umieszczone w ich własnych komputerach, czyli klientach bazy danych. Programiści baz danych wykorzystują systemy SZRBD do stworzenia i utrzymania bazy danych oraz towarzyszących im aplikacji użytkownika. Integralność danych oraz ich bezpieczeństwo zapewniane są na serwerze, co pozwala na uzyskanie dostępu do tego samego zestawu danych wielu różnym aplikacjom użytkownika, bez wpływu na integralność oraz bezpieczeństwo tych danych.

Poza modelem relacyjnym

Mimo że systemy SZRBD są powszechnie wykorzystywane w typowych aplikacjach biznesowych, takich jak kontrola inwentaryzacyjna, zarządzanie pacjentami, usługi bankowe, przetwarzanie zamówień oraz planowanie wydarzeń, nie sprawdziły się one w przypadku aplikacji typu CAD (ang. *computer-aided design*, projektowanie wspomagane komputerowo), GIS (ang. *geographic information system*, system informacji przestrzennej) oraz systemów przechowywania multimediiów. Odpowiedzią na ten problem było pojawienie się dwóch nowych modeli baz danych: *obiektowych* oraz *obiekto-relacyjnych*.

Model obiektowy zawiera wszystkie cechy charakterystyczne dla obiektowego języka programowania i degraduje relacyjną bazę danych do statusu magazynu danych. Podstawowym założeniem jest tutaj to, że programista bazy danych zajmuje się każdym jej aspektem, także operacjami mającymi na celu manipulację danymi z poziomu oprogramowania obiektowej bazy danych. Nie istnieje już jasny podział na oprogramowanie bazy danych i aplikację. Tak jak w przypadku każdego innego modelu to podejście ma swoje wady i zalety. Versant Corporation i IBM to dwaj producenci wytwarzający oprogramowanie obiektowych baz danych.

Inaczej niż w przypadku modelu relacyjnego, który posiada solidne podstawy teoretyczne w dwóch odrębnych gałęziach matematyki, model obiektowej bazy danych nie posiada żadnych konkretnych podstaw. W związku z tym nie ma jednej spójnej definicji tego modelu. Istnieje jednak wersja modelu zdefiniowana przez konsorcjum Object Management Group (OMG), która pełni rolę standardu dla systemów zarządzania obiektowymi bazami danych.

Uwaga OMG jest międzynarodową organizacją typu non profit, która zajmuje się problemami standardów obiektowych. Została założona w 1989 roku i składa się z ponad 800 organizacji członkowskich. Należy pamiętać, że OMG nie jest organizacją wprowadzającą standardy, jak na przykład American National Standards Institute (ANSI), a jedynie grupą doradczą i certyfikacyjną.

Model obiektowo-relacyjny (wcześniej znany jako rozszerzony relacyjny model danych) rozszerzył z kolei model relacyjnej bazy danych poprzez wprowadzenie elementów obiektowych oraz cech charakterystycznych, takich jak klasy, hermetyzacja oraz dziedziczenie. Chodziło o to, by rozszerzenia pozwoliły relacyjnej bazie danych na zarządzanie i manipulację złożonymi rodzajami danych, takimi jak strumienie audio, wideoklipy czy też rysunki architektoniczne. Wśród producentów tworzących aplikacje bazujące na tym modelu są IBM, Oracle, Microsoft oraz PostgreSQL Global Development Group.

Co niesie przyszłość

Sposób użytkowania baz danych zupełnie się zmienił w ostatnich latach. Wiele organizacji zdało sobie sprawę, że z rozmaitych relacyjnych i nierelacyjnych baz danych można zebrać wiele użytecznych informacji. To popchnęło je do zastanowienia się nad sposobem eksploracji danych do postaci użytecznych informacji analitycznych, które mogłyby być wykorzystane przy podejmowaniu ważnych decyzji biznesowych. Co więcej, organizacje zastanawiały się także nad konsolidacją oraz integracją danych w bazie wiedzy. To rzeczywiście były trudne zagadnienia.

IBM zaproponował stworzenie hurtowni danych, która według oryginalnego założenia umożliwiłaby organizacjom dostęp do danych przechowywanych w wielu nierelacyjnych bazach danych. Pierwsze próby implementacji hurtowni danych nie powiodły się, głównie z powodu złożoności oraz problemów z wydajnością, które takie zadania generują. Dopiero w latach 90. implementacja hurtowni danych stała się wykonalna i praktyczna. Bill Inmon, uznawany za ojca hurtowni danych, jest głośnym orędownikiem technologii, który odegrał rolę w jej ewolucji. Hurtownie danych są obecnie coraz bardziej popularne, w miarę jak firmy starają się spożytkować duże ilości danych przechowywanych latami w bazach.

Na sposób wykorzystywania baz danych przez organizacje znaczący wpływ ma również internet. Wiele firm wykorzystuje sieć do poszerzania bazy konsumentów, a znaczna ilość danych, którymi dzielą się z konsumentami lub które od nich uzyskują, przechowywana jest w bazie danych. Programiści najczęściej do zbierania i konsolidowania danych z wielu relacyjnych i nierelacyjnych systemów wykorzystują język XML (ang. *Extensible Markup Language*). Wielu producentów podjęło znaczny wysiłek, by namówić swoich klientów do tworzenia baz i przechowywania danych w chmurze, to znaczy w miejscu niezwiązanym z siedzibą klienta. Chodzi o to, żeby klient z każdego miejsca i o każdej porze za pomocą internetu mógł mieć dostęp do bazy danych znajdującej się w chmurze. Z uwagi na powszechne w ostatnich latach występowanie i wykorzystywanie urządzeń połączonych zapewne interesujący będzie rozwój systemów zarządzania bazami danych w tego typu środowisku.

Ostatnia uwaga

Systemy SZRBD mają długą historię i wciąż odgrywają ważną rolę w sposobie interakcji ludzi, firm i organizacji z danymi. Rola tych programów nieustannie się poszerza, w miarę jak dane stają się bardziej dostępne przez internet, a firmy narzucają coraz większe tempo, by rozszerzyć swoją obecność w sieci. Liczne organizacje zainwestowały spore sumy w relacyjne bazy danych, które w najbliższym czasie nie znikną.

Podsumowanie

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

Następnie pokrótce omówiłem hierarchiczny model baz danych oraz model sieciowy. Zakres tematyczny objął struktury danych, zależności i sposoby dostępu do danych wykorzystywane w obu modelach, a także ich główne wady. Dowiedziałeś się, że te modele były szeroko wykorzystywane w początkach istnienia zarządzania bazami danych i doprowadziły do stworzenia oraz wprowadzenia modelu relacyjnych baz danych.

Następnie szczegółowo omówiłem temat modelu relacyjnych baz danych, ich historii i charakterystyki. Podkreśliłem, że ten model bazuje na specyficznych gałęziach matematyki oraz że to właśnie matematyczne podstawy sprawiają, iż model ten jest tak zwarty strukturalnie. Następnie przyjrzeliliśmy się strukturom danych oraz zależnościom występującym w tym modelu oraz roli, jaką w dostępności do danych pełni SQL. Bez wątpienia pamiętasz, że SQL to standardowy język wykorzystywany do pracy z relacyjnymi bazami danych. Tę część rozdziału zakończyło podsumowanie zalet modelu relacyjnych baz danych.

Później omówiłem krótką historię systemów zarządzania relacyjnymi bazami danych, zaczynając od systemów mainframe z lat 70., poprzez lata 80. i systemy oparte na PC, aż po systemy klient-serwer z lat 90. Na tym etapie powinniśmy znać okoliczności, które doprowadziły do rozwoju systemów baz danych wykorzystywanych dzisiaj.

Dalsza część rozdziału zawierała krótkie omówienie modeli baz danych obiektowych i obiektowo-relacyjnych. Dowiedziałeś się, że te modele powstały jako środki do radzenia sobie z zaawansowanymi aplikacjami baz danych oraz że każdy z nich zawiera różne obiektowe elementy i cechy.

Rozdział zamknęło krótkie omówienie tematu hurtowni danych oraz dostępu do danych przez internet. Wyjaśniłem, że hurtownie danych stworzone są do konsolidowania i integracji danych z niejednorodnych źródeł oraz że możliwości ich pełnego wykorzystania dopiero niedawno stały się dostępne w praktyce. Napisałem też, że XML jest powszechnym formatem zbierania danych ze źródeł relacyjnych i nierelacyjnych oraz że coraz bardziej popularna staje się metoda przechowywania danych w chmurze. To wszystko sugeruje, że relacyjne bazy danych będą najprawdopodobniej wykorzystywane jeszcze przez jakiś czas bez względu na ogromny wpływ internetu na sposób, w jaki organizacje wykorzystują bazy danych.

W następnym rozdziale zastanowimy się, dlaczego projektowanie baz danych powinno nas interesować oraz dlaczego teoria jest ważna. Omówię wady i zalety dobrego projektu.

Pytania kontrolne

- Wymień dwa typy baz danych wykorzystywanych obecnie.
- Jaki typ danych przechowuje analityczna baza danych?
- Prawda czy fałsz: Operacyjna baza danych jest wykorzystywana głównie do przetwarzania transakcji internetowych (OLTP).
- Jakie dwa modele danych były w powszechnym użytku w czasach przed modelem relacyjnych baz danych?
- Opisz zależność rodzic – dziecko.
- Co to jest zestaw struktur?
- Wymień jedną z gałęzi matematyki, na której opiera się model relacyjny.
- W jaki sposób relacyjna baza danych przechowuje dane?
- Wymień trzy typy zależności w relacyjnej bazie danych.
- W jaki sposób pozyskuje się dane z relacyjnej bazy danych?
- Wymień dwie zalety relacyjnej bazy danych.
- Co to jest system zarządzania relacyjną bazą danych?
- Jaka przesłanka kryje się za modelem obiektowo-relacyjnym?
- Jaki jest cel hurtowni danych?

Skorowidz

A

alias, 237
analiza bazy danych, 91
 cel, 119
 dokumenty papierowe, 121
 etapy procesu, 121
 formy baz, 119
 gromadzenie danych, 121
 identyfikacja i przegląd raportów, 124
 lista pól, 145
 poznanie, 118
 prezentowanie informacji, 124
 programy komputerowe, 121
 przeгляд
 prezentacji, 124
 stron internetowych, 125
Rowery Michała, 151
strony WWW, 122
wywiady, 127
 przykładowe pytania, 118
 ważne informacje, 152
 z kierownictwem, 143
 z użytkownikami, 132
zasada, 119
analiza wymagań, 56
aplikacja, 477
 bazodanowa, 324, 477
 użytkownika końcowego, 477
arkusz kalkulacyjny, 393
 a baza danych, 394
 przykład raportu, 395
 typowy widok danych, 123
arkusz specyfikacji reguł biznesowych, 333
 Deklaracja, 333
 Dotyczy
 charakterystyk relacji, 334

 elementów pola, 334
 struktur, 334
elementy, 333
Kategoria, 334
Ograniczenie, 333
Podjęte działania, 334
Rodzaj, 334
Rowery Michała, 349
sprawdzanie, 346
Test poprzez, 334
arkusz specyfikacji widoku, 372
 filtry, 373
 nazwa, 372
 rodzaj, 372
 tabele bazowe, 373
 wyrażenia pól obliczeniowych, 373
atrybut, 69, 71, 477
autowywiad, 100

B

baza danych, 36
 analityczna, 36, 477
 projektowanie, 399
 definicja celu, 106
 definiowanie tabel
 lista cech podmiotów, 131
 dlaczego projektować, 51
 hierarchiczna, 478
 implementacja fizyczna, 52
 kompletowanie dokumentacji, 387
 kryterium, 371
 lista wartości, 149
 logiczny projekt, 52
 luźno zdefiniowana, 119
 mainframe, 120
 metody projektowania, 56

baza danych
 model
 hierarchiczny, 37
 obiektowo-relacyjny, 47
 obiektywowy, 46
 relacyjny, 41
 sieciowy, 39
 odziedziczona, 91, 480
 oparta na wiedzy ludzkiej, 119
 operacyjna, 36, 480
 papierowa, 91, 118, 481
 analiza, 120
 problemy
 zbędnych danych, 38
 związane z aplikacją, 24
 związane z danymi, 24
 proces tworzenia, 26
 przykład raportu, 395
 redundancja, 190
 relacyjna, 35, 41, 483
 podstawy teoretyczne, 53
 wady, 45
 zalety, 44
 zarządzanie, 45
 reprezentacja aspektów struktury, 56
 rodzaje, 36
 rozwój aplikacji, 52
 sieciowa, 483
 spadkowa, 91, 118
 analiza, 120
 terminologia, 64
 tradycyjna, 485
 typowy widok danych, 123
 w chmurze, 48
 wczesne modele, 36
 BIGINT, 240
 BIT, 240, 241
 brakująca wartość, 67, 477

C

CAD, 46
 cele misji, 452, 477
 CHARACTER, 240
 ciąg zerowej długości, 477
 Codd Edgar F., 41
 computer-aided design, 46
 CURRENCY, 241

D

dane, 64, 477
 a informacje, 65
 ciągi
 binarne, 240
 znaków, 240
 data i godzina, 241
 dostęp przez internet, 48
 dynamiczne, 36, 65, 70, 477
 eliminacja duplikatów, 78
 gromadzenie
 analiza widoków baz danych, 122
 sposoby, 121
 integralność, 44, 82
 klucz obcy, 75
 korygowanie, 384
 ogólna, 346
 relacje, 266
 sprawdzanie, 384
 interwał, 241
 konsekwencja i dokładność, 44
 liczbowe
 dokładne, 240
 logiczne, 241
 przybliżone, 241
 nadmiarowe, 192
 niezależność od aplikacji, 44
 pozyskiwanie, 42, 44
 redundantne, 192, 477
 schemat widoku, 361
 statyczne, 36, 64, 70, 477
 typy, 240
 alfanumeryczny, 241
 data i czas, 241
 numeryczny, 241
 rozszerzony, 241
 w środowisku klient-serwer, 46
 DATE, 241
 DECIMAL, 240
 definicja celu, 91
 formułowanie, 106
 przykładowe pytania wywiadu, 108
 Rowery Michała, 108
 układanie, 107
 zadania specyficzne, 106
 definiowanie
 elementów fizycznych pola, 239

diagramy
 relacji, 477
 związków encji, 56
 długość pola, 242
 dokumentacja
 formularze, 455
 projektu, 387
 łamanie zasad projektowania, 402
 domena, 82
 DOUBLE PRECISION, 241
 dozwolone
 operacje, 248
 porównania, 247
 znaki, 242
 duplikat
 danych, 478
 pola, 193
 dziecko, 37
 dziedzina, 478
 dzielone z, 236

E

elementy
 fizyczne, 233
 logiczne, 234
 specyfikacja pola klucza obcego, 301
 ogólne, 233
 pola
 fizyczne, 82
 logiczne, 82
 ogólne, 82
 ERD, 56
 etykieta, 234
 Extensible Markup Language, 48

F

filtr, 478
 filtrowanie rekordów, 43
 fizyczna niezależność danych, 478
 FLOAT, 241
 format wyświetlania, 243
 formularze
 dokumentacyjne, 455
 wprowadzania danych, 478

funkcja
 agregacji, 361, 478
 znacznik null, 69
 COUNT, 69
 Count(Nazwa Produktu), 376
 MAX(), 370

G

geographic information system, GIS, 46

H

hurtownia danych, 47, 478

I

identyfikacja
 bazy danych, 118
 cech podmiotów, 130
 domniemanych podmiotów, 160
 duplikatów
 lista podmiotów i tabel, 162
 elementów tabeli, 212
 metod prezentacji danych, 124
 niezdefiniowanych tabel-podzbiorów, 202
 przyszłych struktur danych, 141
 relacji, 276, 450
 klucze główne, 283
 rodzaju
 udziału tabel, 308
 widoków, 366
 wykorzystywanych danych, 132
 sposobu gromadzenia danych, 121
 stopnia udziału tabel, 310
 tematów, 129
 wymagań informacyjnych
 dodatkowych, 139, 144
 dotyczących perspektyw, 450
 istniejących, 137
 obecnych, 143
 ogólnych, 145
 przyszłych, 141, 144
 implementacja fizyczna, 26
 indeks, 76, 478
 informacje, 65, 478
 INGRES, 45

INT, 241
 INTEGER, 240
 integralność
 danych, 82, 478
 zasady biznesowe, 83
 domeny, 83
 dozwolone znaki, 242
 dziedziny, 478
 encji, 83, 478
 klucze, 212
 korygowanie, 384
 łamanie zasad projektowania, 400
 na poziomie
 pól, 83, 233, 385, 448, 472, 479
 relacji, 262, 313, 385, 452, 472, 478
 tabel, 83, 223, 385, 453, 472, 478
 zależności, 83
 odniesień, 83
 ograniczenia, 346
 referencyjna, 478
 reguły biznesowe, 386
 Rowery Michała, 388
 specyfikacja wszystkich pól, 232
 sprawdzanie, 384
 strukturalna, 478
 widok walidacji, 364
 widoki, 386
 wielopoziomowa, 479
 Interactive Graphics Retrieval System, 45

J

jawne informacje, 479
 język programowania, 479

K

kardynalność, 479
 kartoteka, 118
 kierownictwo, 143
 klucz, 74, 479
 funkcja, 212
 główny, 194, 218, 479
 oznaczenie, 221
 wartość, 218
 warunki, 219, 446
 wymagania, 218
 wytyczne wyboru, 218

zasady definiowania, 222
 złożony, 487
 kandydujący, 212
 kombinacja pól, 213
 oznaczenie pola, 217
 sztuczny, 216, 484
 warunki, 213, 446, 472
 wybór pola, 214
 obcy, 75
 dokładna analiza, 299
 warunki, 299, 446, 472
 podstawowy, 69, 74
 złożony, 79
 Rowery Michała, 225
 ustanawianie
 relacji, 285
 relacji zwrotnych, 294
 zastępczy, 222, 479
 oznaczenie, 223
 komputer mainframe, 479
 kreator zapytań, 479
 krotka, 69, 72, 479

L

LAN, 479
 lista
 cech, 131, 479
 elementy ogólne, 146
 listy wartości, 149
 przegląd i czyszczenie, 146
 celów
 Rowery Michała, 204
 oznaczanie datami, 155
 podmiotów, 129, 479
 a podmiot na liście tabel, 163
 cechy podmiotu, 130
 definiowanie wstępnej listy tabel, 161
 duplikaty na liście tabel, 162
 połączenie elementów z listą tabel, 164
 pól
 definiowanie wstępnej listy tabel, 160
 identyfikacja podmiotów, 160
 nowe cechy w próbkach, 148
 przegląd z uczestnikami rozmów, 151
 Rowery Michała, 203
 weryfikacja, 151
 wstępna, 145

pól obliczeniowych, 150, 479
 wykorzystanie w widokach, 370
 tabel
 ostateczna, 167
 Rowery Michała, 203
 wskazywanie typów, 172
 wstępna, 160, 164
 wartości, 149
 wyliczeniowa, 149
 logiczna
 niezależność danych, 480
 relacja potomna, 480
 logika predykatów pierwszego rzędu, 480
 lokalna sieć komputerowa, 480

M

macierz tabeli, 276
 maska wprowadzania, 243
 mechanizm
 pobierania danych z wielu tabel, 262
 member, 39
 metodologia projektowania, 472
 miejsca dziesiętne, 242
 misja, 451, 480
 model relacyjny, 480
 modelowanie
 danych, 56
 roli obiektów, 56
 semantyczno-obiektowe, 56
 MONEY, 241

N

NATIONAL CHARACTER, 240
 nazwa pola, 234
 nazywanie
 pól
 akronimy, 180
 etykiety, 234
 identyfikacja cech, 180, 181
 liczba mnoga, 186
 liczba pojedyncza, 181
 minimalizm, 180
 unikatowość nazw, 179
 wytyczne, 449
 zaciemnianie sensu, 181

tabel
 akronimy i skróty, 170
 dwuznaczność nazw, 171
 identyfikacja podmiotu, 169
 kolekcje, 171
 liczba mnoga, 171, 172
 nazwy własne, 170
 unikanie cech fizycznych, 169
 unikatowość nazw, 168
 wytyczne, 449
 zwięzłość, 169

NCHAR, 240
 niejawne informacje, 480
 nieznaną wartość, 67
 normalizacja, 57, 58, 467, 480
 cel, 469
 rozwiązywanie tradycyjnych problemów, 472
 uwagi, 467
 wizualizacja procesu, 469
 null, 66
 NUMERIC, 240

O

obiekt, 480
 ograniczenia reguły biznesowej, 480
 OLAP, 36, 480
 OLTP, 36, 480
 OMG, 47
 Online Analytical Processing, 480
 Online Transaction Processing, 480
 opis
 pola, 238
 a nazwa pola, 238
 akronimy i skróty, 239
 identyfikacja pola, 238
 informacje zależne od implementacji, 239
 uzależnienie od innych opisów, 239
 używanie przykładów, 239
 wytyczne, 448
 zwięzłość, 238
 tabeli, 480
 definicja, 173
 implementacja, 174
 istotność dla instytucji, 173
 jasność i zwięzłość, 174
 redagowanie, 173

opis
 tabeli
 stosowanie przykładów, 175
 wytyczne, 448
 wywiady z użytkownikami, 175
 zależność od innego opisu, 174
 ostateczna lista tabel, 481
 owner, 39

P

parsowanie, 481
 perspektywa, 73
 agregacji, 481
 danych, 481
 walidacji, 481
 pojęcia
 związane z
 integralnością, 82
 strukturą, 69
 wartością, 64
 zależnościami, 76
 pokaz slajdów, 124
 pole, 42, 69, 71, 146, 481
 anomalia, 192
 eliminacja, 182
 grupujące, 364
 idealne
 cechy, 233
 warunki, 183, 447, 472
 integralność, 233
 klucz kandydujący, 212
 kryteria, 371
 nazywanie, 179
 niekluczowe, 223, 481
 obliczeniowe, 72, 363, 481
 stosowanie w widoku, 369
 odniesienia, 194
 opis, 238
 referencyjne, 481
 reguły biznesowe
 definiowanie, 331
 Rowery Michała, 205
 specyfikacja, 82, 231
 struktura, 239
 testowanie, 184
 udział w powiązaniu tabel, 192

wartość
 klucz główny, 220
 null, 67
 obliczona lub scalona, 183
 opcje wprowadzania, 246
 wieloczęściowe, 183, 481
 eliminacja, 185, 472
 wielowartościowe, 72, 183, 481
 eliminacja, 186, 196, 452, 472
 relacje, 288
 spłaszczenie, 187
 spłaszczone, 272
 wystąpienie w wielu tabelach, 184
 zdublowane, 183, 193, 481
 złożone, 72, 183
 poprawianie wydajności obliczeń, 400
 dostrojenie systemu operacyjnego, 401
 implementacja bazy danych, 402
 przegląd aplikacji, 402
 struktura bazy danych, 402
 usprawnienie sprzętu, 401
 porządkowanie listy cech
 cechy podmiotów, 147
 elementy
 o tej samej nazwie, 146
 o tych samych cechach, 146
 postać normalna, 57, 469, 481
 a problemy projektowania, 471
 Boyce'a-Codda, 471
 czwarta, 471
 druga, 471
 dziedzina-klucz, 471
 piąta, 471
 pierwsza, 471
 szósta, 471
 trzecia, 58, 471
 pozyskiwanie danych, 42
 prefiks, 180
 prezentacja, 124
 informacji metody, 124
 ekranowa, 481
 procedura
 eliminowania pól wielowartościowych, 189
 identyfikowania relacji między tabelami, 281
 łączenia listy tabel i podmiotów, 203
 opracowywania pól wielowartościowych, 288
 tworzenia ostatecznej listy tabel, 204

- udoskonalania
 - pól, 205
 - struktur tabel, 206
 - ustanawiania relacji wiele-do-wielu, 290
 - proces
 - implementacji, 481
 - logiczny, 26
 - projektowania bazy danych, 481
 - projekt
 - a implementacja, 473
 - aplikacji bazodanowej
 - reguły biznesowe, 324
 - logiczny
 - a fizyczny, 473
 - reguły biznesowe, 324
 - na bazie arkusza kalkulacyjnego, 393
 - plaskie pliki, 392
 - pod kątem oprogramowania, 396
 - przykłady, 461
 - projektowanie, 54
 - analitycznej bazy danych, 399
 - analiza istniejącej bazy danych, 91, 117
 - cele, 55
 - definiowanie
 - celu, 91
 - tabel, 92
 - widoków, 94
 - założeń wstępnych, 91
 - diagram procesu, 427
 - użyte symbole, 459
 - dostrajanie struktur tabel, 192
 - etapy, 468
 - fazy, 56, 427
 - integralność danych, 82
 - kontrola, 94, 383
 - wprowadzanie, 94
 - korzystanie z papieru, 177
 - listy
 - pól, 92
 - struktur tabel, 178
 - metodologia, 54, 57, 470
 - korzyści, 54
 - normalizacja, 471
 - na podstawie systemu SZRBD, 473
 - nauka, 405
 - określanie reguł biznesowych, 93, 321
 - popelniane błędy, 391
 - poznanie organizacji, 107
 - proces
 - rozpoczęcie, 99
 - ważność, 90
 - relacje między tabelami, 93, 261
 - specyficzne ograniczenia, 93, 324
 - struktur danych, 92, 159
 - tabeli idealnej, 59
 - tradycyjne metody, 56
 - ustalanie zależności, 93
 - zalety dobrego projektu, 55
 - próbki, 121
 - danych, 134
 - identyfikacja widoków, 366
 - opis, 135
 - przechowywanie, 124
 - raporty, 124, 137
 - rola, 135
 - szkice nowych raportów, 145
 - złożone, 135
 - znajdowanie nowych cech, 148
 - zrzut ekranu, 122
 - przeprowadzanie wywiadów, 100
 - pytania kontrolne
 - odpowiedzi, 409
 - pytanie
 - asocjacyjne, 277
 - kontekstowe, 278
 - otwarte, 481
 - skojarzeniowe, 277
 - zamknięte, 482
 - zorientowane
 - na działania, 278
 - na własność, 278
- R**
- R:BASE, 43
 - raporty, 124, 482
 - szkice, 142, 145
 - widok zagregowany, 364
 - RDBMS
 - pole wielowartościowe, 187
 - REAL, 241
 - reguła
 - kaskadowa, 304
 - odmowy, 304
 - restrykcyjna, 304

- reguła
 - usuwania relacji, 304
 - oznaczenia, 306
 - wybór, 305
 - wartości
 - domyślnej, 305
 - null, 305
 - wprowadzania, 246
- reguły biznesowe, 321
 - arkusz specyfikacji, 333, 340
 - czynności weryfikujące, 332, 338, 344
 - definiowanie i ustanawianie, 328
 - egzekwowanie
 - klucz główny, 342
 - identyfikacja widoków, 367
 - kategorie, 326
 - modyfikacja
 - charakterystyk relacji, 337
 - specyfikacji pola, 331
 - ograniczenia
 - dopuszczalne wartości pola, 341
 - rodzaje, 324
 - Rowery Michała, 346
 - specyficzne
 - dla pól, 326, 329, 445, 482
 - dla relacji, 327, 334, 445, 482
 - specyfikacje, 472
 - sprawdzanie, 346
 - stopień uczestnictwa tabeli, 323
 - tabele walidacji, 342
 - ustalanie charakterystyki relacji, 343
 - ustanawianie ograniczeń, 323
 - zorientowane
 - na aplikację, 324
 - na bazę danych, 324, 482
- reguły projektowania
 - naginanie, 399
 - dokumentowanie działań, 402
 - integralność danych, 401
 - nadmiarowe dane, 401
 - niespójne dane, 401
 - nieściśle informacje, 401
- rekord, 42, 69, 72, 212, 482
 - osierocony, 304, 482
 - relacje zwrotne, 273
- relacje, 42, 69, 261, 482
 - analiza kluczy obcych, 299
 - bezpośrednie, 277
 - cechy, 472
 - identyfikowanie, 276
 - pomoc, 283
 - widoków, 367
 - integralność, 262, 313
 - jeden-do-jednego, 264, 281, 482
 - tworzenie schematu, 265
 - ustanawianie, 284
 - z odwołaniem do samej siebie, 482
 - jeden-do-wielu, 265, 281, 482
 - tworzenie schematu, 267
 - ustanawianie, 287
 - z odwołaniem do samej siebie, 482
 - pośrednie, 277
 - reguły biznesowe
 - definiowanie, 336
 - wymóg ograniczenia, 336
 - rekurencyjne, 483
 - rekursywne, 273
 - rodzaje, 263, 483
 - rodzic-dziecko, 483
 - Rowery Michała, 313
 - schemat
 - reguły biznesowe, 328
 - rodzaj udziału tabeli, 309
 - stopień udziału tabeli, 310
 - sprawdzenie struktur tabel, 298
 - symbole, 279
 - udział tabel, 263
 - ustanawianie, 284
 - charakterystyk, 304
 - schematy, 287
 - usuwanie
 - definiowanie reguł, 304
 - ważność, 262
 - weryfikacja
 - lista kontrolna, 312
 - wiele-do-wielu, 267, 272, 281
 - nadmiarowe dane, 292
 - problemy, 268
 - rozwiązywanie, 290
 - tworzenie schematu, 269
 - ulepszanie struktur tabel, 292
 - ustanawianie, 290
 - z odwołaniem do samej siebie, 483
 - właściciel-składowa, 483
 - wzory, 281
 - z odwołaniem do samej siebie, 483

- relacje zwrotne, 273
 - eliminowanie, 296
 - identyfikowanie, 280
 - jeden-do-jednego, 273
 - tworzenie schematu, 274
 - ustanawianie, 294
 - jeden-do-wielu, 274
 - tworzenie schematu, 275
 - ustanawianie, 294
 - nazwa klucza obcego, 301
 - stopień udziału tabel, 311
 - ustanawianie, 294
 - wiele-do-wielu, 275
 - tworzenie schematu, 275
 - ustanawianie, 297
- rodzaj udziału tabeli, 308
 - obowiązkowy, 308
 - opcjonalny, 308
- rodzic, 37
- Rowery Michała, 105
- ROWID, 241
- rozległa sieć komputerowa, 483
- rozszerzone typy danych, 483
- rozwój aplikacji, 26

S

- SERIAL, 241
- składowa, 483
- słowniczek, 477
- SMALLINT, 240
- specyfikacja
 - pola, 483
 - reguły biznesowej, 483
 - widoku, 484
 - źródłowa, 236
- specyfikacje pól, 232, 472
 - definiowanie
 - dla każdego pola w tabeli, 255
 - elementy
 - fizyczne, 239
 - logiczne, 244
 - ogólne, 234
 - formularz, 235
 - kategorie elementów, 233
 - konsultacje, 255
 - ogólna, 252

- reguły biznesowe, 329
- replikowana, 253
- Rowery Michała, 256
- typy, 236
 - wykorzystanie, 250
- unikatowa, 250
- spójność danych, 484
- SQL, 42, 484
 - budowa kwerendy, 43
 - formy implementacji, 43
- stopień
 - uczestnictwa, 484
 - udziału, 81
 - tabeli, 310
- strona
 - internetowa, 124
 - WWW, 484
- struktura
 - bazy danych
 - logiczna, 184
 - danych, 484
 - klucza, 244
 - zbioru, 484
 - złożona, 40
- strukturalny język zapytań, 484
- struktury tabel
 - anomalia
 - puste pola, 198
 - dostrajanie, 192
 - duplikaty pól, 194
 - eliminowanie, 194
 - klucz główny, 194
 - nadmiarowość danych, 194
 - płaskie pliki, 392
 - pola
 - obliczeniowe, 194
 - wieloczęściowe, 194
 - wielowartościowe, 194
- relacje, 262
- Rowery Michała, 227
- sprawdzenie, 298
- tabela idealna, 193
- tabela-podzbiór, 201
 - wyznaczanie, 198
- wstępne
 - przegląd, 224

system
 operacyjny, 484
 zarządzania relacyjną bazą danych, 45, 484
 klient-serwer, 484

System R, 45

SZBD, 484

SZRBD, 45, 484

błędne projektowanie, 396
 dodatkowe typy danych, 241
 klucze zastępcze, 222
 projektowanie baz danych, 52
 reguły usuwania relacji, 304
 składnia funkcji, 370
 usprawnianie przetwarzania danych, 76
 widok, 355
 wykorzystanie, 46

Ś

środowisko programistyczne, 485

T

tabela, 42, 69, 159, 485

asocjacyjna, 76
 danych, 70, 168, 485
 definicja, 173
 definiowanie
 cele misji, 165
 kluczy, 212
 lista podmiotów, 161
 lista pól, 160
 ostatecznej listy tabel, 167
 widoku, 359
 wstępnej listy, 160

dostrajanie
 nazw, 168
 pól, 179
 struktur, 192

duplikaty
 pól, 192
 struktur, 170

dziecko, 77

idealna
 warunki, 193, 299, 447, 472

identyfikowanie
 rodzaju udziału, 308
 stopnia udziału, 310
 widoków, 367

łącząca, 76, 78, 168, 485
 redundancja danych, 291
 ustanawianie relacji, 290
 złożony klucz główny, 291

nadmiarowe dane, 192

nazywanie, 168

opis, 168, 172

podglądu, 70

podmioty podrzędne, 200
 podrzędna

usuwanie relacji, 339

podstawowa, 73

pojedynczy podmiot, 193

poła odniesienia, 195

potomna, 485

powiązanie logiczne, 262

prefiks, 180

prezentacja

obiektu, 70

wydarzenia, 70

projektowanie, 92

przełączkowa, 341, 485

przypisywanie pól, 177

reguły biznesowe, 329

relacje, 262

zwrotne, 273

rodzic, 77

rola, 168

stopień udziału, 81

testowanie struktury

postać normalna, 57

typ, 168

typowa struktura, 69

typu podzbiór, 168

udział w zależności, 80

walidacji, 70, 168, 341, 485

a widok walidacji, 364

reguły biznesowe, 342

ustanawianie relacji, 342

wirtualna, 196, 355

wskazywanie typów na liście tabel, 172

zestawy pól, 200

zestawy pól-duplikatów, 197

tabela-korzeń, 485

tabela-podzbiór, 200, 485

procedura identyfikacji, 202

relacja jeden-do-jednego, 264

Rowery Michała, 207

tabela-rodzic, 234, 485
 tabele bazowe, 485
 tablica asocjacyjna, 485
 technika

- identyfikacji podmiotów
 - cele misji a lista tabel, 165
- identyfikująca cechy, 131
 - dotatkowe wymagania informacyjne, 140
 - nowe tematy, 142
- identyfikująca podmioty, 129
 - nowe tematy, 142
- identyfikująca temat
 - dotatkowe wymagania informacyjne, 140
- kierowania na wybrany temat, 130

 teoria, 53

- zbiorów, 485

 TIME, 241
 TIMESTAMP, 241
 TINYINT, 241
 tworzenie aplikacji, 485

- prawidłowe struktury pól, 183

 typ

- danych, 240
- klucza, 244
- specyfikacji, 236
 - ogólna, 236, 250
 - replikowana, 236, 250
 - unikatowa, 236, 250
- uczestnictwa, 485

U

udział tabeli

- dowolny, 81
- obowiązkowy, 81

 UML, 56
 Unicode, 240
 unikatowość pola, 244
 URL, 485
 użytkownik końcowy, 485

V

VARCHAR, 240

W

WAN, 485
 wartość

- domyślna, 246
- nieznana, 486
- null, 486
- opcjonalna, 213
- wprowadzona przez, 245
- wymagana, 245
- zerowa, 245

 warunek

- FROM, 43
- ORDER BY, 43
- SELECT, 43

 warunki

- idealnego pola, 486
- idealnej tabeli, 486
- klucza
 - głównego, 486
 - kandydującego, 486
 - obcego, 486

 wbudowana integralność odniesień, 38
 weryfikacja podmiotów na liście, 160
 węzeł, 39, 486
 widok, 73, 355, 486

- a potrzeby użytkowników, 356
- anatomia, 357
- arkusz specyfikacji, 372
- danych, 357
 - bezpieczeństwo i poufność, 357
 - filtrowanie, 371
 - jednotabelowy, 357
 - modyfikacja, 358, 361
 - wielotabelowy, 359
- identyfikacja, 367
- implementacja, 359
- indeksowany, 74, 356
- kryteria pól, 371
- najświeższe zmiany, 356
- nazywanie, 372
- określanie i definiowanie, 366
- opisywanie, 372
- przeglądanie dokumentacji, 373
- Rowery Michała, 375
- schemat, 369
- stosowanie pól obliczeniowych, 369

- widok
 - tabela bazowa, 355, 365
 - ustalanie struktury, 367
 - walidacji, 356, 364
 - walidacyjny, 74
 - wirtualny, 355
 - wywiad z użytkownikami, 366
 - zagregowany, 361
 - pola danych, 364
 - pole grupujące, 364
 - pole obliczeniowe, 363
 - zalety, 356
 - zmaterializowany, 74, 356
 - właściciel, 486
 - wskaźnik, 486
 - wspieranie wartości zerowych, 245
 - wstępna lista
 - pól, 486
 - tabel, 486
 - wymagania informacyjne, 486
 - wyszukiwanie
 - informacji ad hoc, 486
 - wytyczne projektowe, 445
 - wywiad, 100
 - analiza bazy danych, 127
 - asystent, 103
 - atmosfera, 102
 - definiowanie
 - reguł biznesowych, 328
 - wstępnych struktur pól i tabeli, 127
 - założeń wstępnych, 111
 - grupy uczestników, 102
 - identyfikacja
 - relacji między tabelami, 276
 - rodzajów widoków, 366
 - informowanie o intencjach, 101
 - kierunek rozwoju firmy, 141, 144
 - kontrola przebiegu, 104
 - nieporozumienia, 101
 - ograniczenie liczby uczestników, 102
 - określanie
 - cech podmiotów, 130
 - cech próbek, 127
 - tematów, 129
 - opisywanie tabel, 175
 - pełna uwaga, 104
 - po zdefiniowaniu tabel, 224
 - podstawowe techniki, 127
 - proces, 128
 - przeгляд
 - próbek danych, 134
 - wymagań informacyjnych, 137
 - przydatność w fazie analitycznej, 127
 - przyszłe wymagania informacyjne, 127
 - pytania
 - istota pytań, 128
 - lista pytań, 103
 - otwarte, 103, 107
 - zamknięte, 103
 - relacje zwrotne, 280
 - rodzaj wykorzystywanych danych, 132
 - specyfikacje pól, 255
 - sposób wykorzystania danych, 127
 - szukanie ukrytych informacji, 112
 - tempo rozmowy, 104
 - ustalanie
 - definicji celu, 107
 - lidera grupy, 103
 - wkład uczestników, 101
 - wytyczne, 451
 - osoba przeprowadzająca wywiad, 102
 - rozmówcy, 101
 - z kierownictwem
 - ważne kwestie, 143
 - z użytkownikami
 - zagadnienia, 132
- X**
- XML, 48
- Z**
- zakres wartości, 246
 - zalecana lektura, 475
 - zależność, 76
 - jeden do jednego, 77, 191
 - jeden do wielu, 39, 78
 - klucze, 212
 - między tabelami, 42
 - nierozwiązana, 79
 - rodzaje udziału, 80
 - stopień udziału, 81
 - typy, 77
 - wiele do wielu, 38, 78

- założenia wstępne, 91
 - błędy, 110
 - definiowanie, 111
 - formułowanie, 109
 - przykłady, 110, 113
 - pytania wywiadu, 111
 - Rowery Michała, 113
 - wyciąganie wniosków, 112
- zapytanie, 486
 - ad hoc, 486
- zasada usuwania, 487
- zbiór, 212
- zdarzenie, 487
- zestaw struktur, 39
- znacznik null, 66
 - wady, 68
 - wartość, 67

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ 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>


Wiedza dotycząca baz danych w pigułce!

Dzisiejszy świat opiera się na bazach danych. Są one sercem każdego przedsięwzięcia, począwszy od działalności banku, a skończywszy na zakupach internetowych. Projektowanie baz danych wymaga nie lada kunsztu, a drobny błąd może doprowadzić do nieoczekiwanych konsekwencji. Dlatego od projektantów baz danych wymaga się ogromnej wiedzy i dokładności, a doświadczenie w tej dziedzinie zdobywa się latami.

Dzięki tej książce będziesz w stanie zgłębić tajniki budowy baz danych, podane w przejrzysty, przejrzysty i przystępny sposób. W trakcie lektury poznasz rodzaje baz, ich dostępne modele oraz cel ich projektowania. Kolejne rozdziały dotyczą procesu projektowania nowej bazy oraz analizowania baz istniejących. Ponadto dowiesz się, jak istotne jest właściwe określenie kluczy i relacji oraz nałożenie więzów integralności. Szczególną uwagę powinieneś zwrócić na rozdział poświęcony najczęściej popełnianym błędom — jego dokładna lektura pozwoli Ci uniknąć wielu problemów. Książka ta jest obowiązkową lekturą dla wszystkich osób mających styczność z bazami danych w codziennej pracy.

Dzięki tej książce:

- ▶ poznasz rodzaje baz danych
- ▶ zorganizujesz proces projektowania bazy
- ▶ nauczysz się analizować strukturę istniejącej bazy
- ▶ unikniesz pułapek
- ▶ poznasz tajniki baz danych

 Addison-Wesley
Pearson Education

helion.pl
księgarnia
internetowa

Nr katalogowy: 16983

 Księgarnia internetowa:
<http://helion.pl>

 Zamówienia telefoniczne:
0 801 339900
 **0 601 339900**

 **Helion**

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/nowości>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

ślęgnij po **WIECEJ**



KOD KORZYŚCI

ISBN 978-83-246-7995-9



9 788324 679959

cena: 79,00 zł

Informatyka w najlepszym wydaniu