

Larry Rockoff

# JĘZYK SQL

PRZYJAZNY PODRĘCZNIK



SQL dla każdego!

Helion 

Tytuł oryginału: The Language of SQL

Tłumaczenie: Beata Błaszczyk

Projekt okładki: Studio Gravite / Olsztyn; Obarek, Pokoński, Pazdrijowski, Zaprucki  
Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

ISBN: 978-83-246-9354-2

© 2011 Course Technology, a part of Cengage Learning.

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, without the prior written permission of the publisher.

DB2, Informix, and IBM are registered trademarks of IBM Corporation. Oracle is a registered trademark of Oracle Corp. MySQL is a registered trademark of MySQL AB. Sybase and SQL Anywhere are registered trademarks of Sybase Inc. Access, Excel, Microsoft, SQL Server, and Windows are registered trademarks of Microsoft Corporation. Mac OS is a registered trademark of Apple Inc. PostgreSQL is a trademark of PostgreSQL Inc and the Regents of the University of California.

All other trademarks are the property of their respective owners.  
All images © Cengage Learning unless otherwise noted.

Polish edition copyright © 2014 by Helion S.A. All rights reserved.

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 Wydawnictwo HELION 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 Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [helion@helion.pl](mailto: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/jsqlpp>  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzje.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem: <ftp://ftp.helion.pl/przyklady/jsqlpp.zip>

Printed in Poland.

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

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

# SPIS TREŚCI



<b>O autorze .....</b>	<b>11</b>
<b>Wprowadzenie .....</b>	<b>13</b>
<b>Rozdział 1. Relacyjne bazy danych i SQL .....</b>	<b>15</b>
Język i logika .....	15
Definicja SQL .....	17
Microsoft SQL Server, Oracle i MySQL .....	17
Inne bazy danych .....	19
Relacyjne bazy danych .....	20
Klucze główne i obce .....	21
Typy danych .....	22
Wartości NULL .....	24
Znaczenie SQL .....	24
Co dalej? .....	25
<b>Rozdział 2. Podstawy pobierania danych .....</b>	<b>27</b>
Prosta instrukcja SELECT .....	27
Uwagi dotyczące składni .....	28
Wybieranie kolumn .....	29
Nazwy kolumn zawierające spacje .....	30
Co dalej? .....	31
<b>Rozdział 3. Obliczenia i aliasy .....</b>	<b>33</b>
Pola obliczane .....	33
Wartości literału .....	34
Obliczenia arytmetyczne .....	35
Konkatenacja pól .....	36
Aliaszy kolumn .....	37
Aliaszy tabel .....	39
Co dalej? .....	40

<b>Rozdział 4. Korzystanie z funkcji .....</b>	<b>41</b>
Zastosowanie funkcji .....	41
Funkcje znakowe .....	42
Funkcje zagnieżdżone .....	47
Funkcje daty i czasu .....	48
Funkcje liczbowe .....	50
Funkcje konwersji .....	52
Co dalej? .....	54
<b>Rozdział 5. Sortowanie danych .....</b>	<b>55</b>
Dodawanie sortowania .....	55
Sortowanie danych w porządku rosnącym .....	56
Sortowanie danych w porządku malejącym .....	57
Sortowanie względem więcej niż jednej kolumny .....	58
Sortowanie względem pola obliczanego .....	58
Więcej na temat sekwencji sortowania .....	59
Co dalej? .....	62
<b>Rozdział 6. Logika obliczania wartości wyjściowych w kolumnach .....</b>	<b>63</b>
Logika IF-THEN-ELSE .....	64
Format prosty .....	64
Format przeszukujący .....	66
Co dalej? .....	68
<b>Rozdział 7. Logika obliczania wartości wyjściowych w wierszach .....</b>	<b>69</b>
Zastosowanie kryteriów selekcji .....	69
Operatory klauzuli WHERE .....	71
Ograniczanie liczby zwracanych wierszy .....	72
Ograniczanie liczby wierszy za pomocą sortowania .....	73
Co dalej? .....	75
<b>Rozdział 8. Logika Boole'a .....</b>	<b>77</b>
Złożone warunki logiczne .....	77
Operator AND .....	78
Operator OR .....	79
Zastosowanie nawiasów .....	79
Zastosowanie wielu nawiasów .....	81
Operator NOT .....	82
Operator BETWEEN .....	84
Operator IN .....	85
Logika Boole'a a wartości NULL .....	86
Co dalej? .....	88

<b>Rozdział 9. Dopasowywanie nieprecyzyjnie określonych wartości .....</b>	<b>89</b>
Dopasowywanie do wzorca .....	89
Znaki wieloznaczne .....	92
Dopasowanie za pomocą brzmienia .....	94
Co dalej? .....	96
<b>Rozdział 10. Dokonywanie podsumowań .....</b>	<b>99</b>
Usuwanie duplikatów .....	99
Funkcje agregujące .....	101
Funkcja COUNT .....	103
Grupowanie danych .....	104
Grupowanie i sortowanie względem kilku kolumn .....	105
Kryteria selekcji w ramach agregacji .....	107
Co dalej? .....	109
<b>Rozdział 11. Złączenia wewnętrzne tabel .....</b>	<b>111</b>
Łączenie dwóch tabel .....	112
Złączenie wewnętrzne .....	114
Kolejność tabel w złączeniach wewnętrznych .....	115
Alternatywna składnia złączeń wewnętrznych .....	116
Aliasy tabel — ciąg dalszy .....	117
Co dalej? .....	118
<b>Rozdział 12. Złączenia zewnętrzne tabel .....</b>	<b>119</b>
Złączenie zewnętrzne .....	120
Złączenia lewostronne .....	121
Weryfikacja występowania wartości NULL .....	123
Złączenia prawostronne .....	124
Kolejność tabel w złączeniach zewnętrznych .....	125
Złączenia pełne .....	126
Co dalej? .....	128
<b>Rozdział 13. Złączenia zwrotne i widoki .....</b>	<b>129</b>
Złączenia zwrotne .....	129
Tworzenie widoków .....	132
Pobieranie danych z widoków .....	133
Zalety stosowania widoków .....	134
Modyfikowanie i usuwanie widoków .....	136
Co dalej? .....	137
<b>Rozdział 14. Podzapytania .....</b>	<b>139</b>
Rodzaje podzapytań .....	139
Wykorzystanie podzapytania jako źródła danych .....	140
Wykorzystanie podzapytania w kryteriach selekcji .....	143
Podzapytania skorelowane .....	144

Operator EXISTS .....	146
Zastosowanie podzapytania do wyznaczenia wartości kolumny obliczanej .....	147
Co dalej? .....	148
<b>Rozdział 15. Logika zbiorów .....</b>	<b>149</b>
Zastosowanie operatora UNION .....	150
Dołączanie lub eliminowanie duplikatów za pomocą operatora UNION .....	152
Krzyżowanie zapytań .....	154
Co dalej? .....	156
<b>Rozdział 16. Procedury składowane i parametryzacja .....</b>	<b>157</b>
Tworzenie procedur składowanych .....	158
Parametry w procedurze składowanej .....	160
Wykonywanie procedur składowanych .....	161
Modyfikowanie i usuwanie procedur składowanych .....	162
Funkcje — ciąg dalszy .....	163
Co dalej? .....	164
<b>Rozdział 17. Modyfikowanie danych .....</b>	<b>165</b>
Sposoby modyfikacji danych .....	165
Wstawianie danych .....	166
Usuwanie danych .....	170
Aktualizacja danych .....	171
Aktualizacja danych w tabeli za pomocą podzapytań skorelowanych .....	172
Co dalej? .....	174
<b>Rozdział 18. Utrzymanie tabel .....</b>	<b>175</b>
Język definicji danych .....	175
Atrybuty tabel .....	176
Kolumny w tabelach .....	177
Klucze główne i indeksy .....	178
Klucze obce .....	179
Tworzenie tabel .....	180
Tworzenie indeksów .....	182
Co dalej? .....	182
<b>Rozdział 19. Zasady projektowania baz danych .....</b>	<b>183</b>
Cele normalizacji .....	184
W jaki sposób dokonywać normalizacji danych .....	186
Sztuka projektowania bazy danych .....	190
Alternatywy dla normalizacji .....	191
Co dalej? .....	193
<b>Rozdział 20. Sposoby prezentacji danych .....</b>	<b>195</b>
Kilka słów o warstwie prezentacji danych .....	195
Narzędzia do raportowania i raporty krzyżowe .....	196

Arkusze kalkulacyjne i tabele przestawne .....	198
Co dalej? .....	200
<b>Dodatek A. Praca z bazą danych Microsoft SQL Server .....</b>	<b>203</b>
Wstęp .....	203
Instalacja SQL Server Express 2014 i SQL Server Management Studio .....	204
Praca z SQL Server Management Studio .....	204
<b>Dodatek B. Praca z bazą danych MySQL .....</b>	<b>207</b>
Wstęp .....	207
Instalacja MySQL Server oraz MySQL Workbench .....	208
Praca z MySQL Workbench .....	209
<b>Dodatek C. Praca z bazą danych Oracle .....</b>	<b>211</b>
Wstęp .....	211
Instalacja Oracle Database Express Edition .....	211
Praca z Oracle Database Express Edition .....	212
<b>Dodatek D. Lista wszystkich instrukcji SQL .....</b>	<b>215</b>
<b>Skorowidz .....</b>	<b>217</b>







# RELACYJNE BAZY DANYCH I SQL

W pierwszym rozdziale znajdziesz podstawowe informacje, które pozwolą Ci szybko rozpocząć samodzielne pisanie instrukcji SQL w miarę zapoznawania się z zagadnieniami przedstawionymi w kolejnych rozdziałach. Poruszę w nim dwa podstawowe tematy. Po pierwsze, dokonam przeglądu baz danych, do których odniesienia znajdziesz w niniejszej książce. Przedstawię również kilka podstawowych informacji o tym, w jaki sposób w tych bazach danych wykorzystywany jest język SQL. Zwrócę również uwagę na pewne cechy tej książki, które pozwolą Ci łatwo określić składnię SQL wykorzystywaną w przypadku konkretnej bazy danych, której używasz.

Po drugie, omówię kilka kluczowych charakterystyk związanych z projektowaniem relacyjnych baz danych. Będzie więc mowa o tabelach, wierszach, kolumnach, kluczach i typach danych. Gdy posiadasz już podstawową wiedzę, drogi Czytelniku, będziesz mógł w krótkim czasie rozpocząć pracę z SQL. Nie zwlekając więc już dłużej, zaczynamy.

## Język i logika

Na wstępie chciałbym się do czegoś przyznać — tytuł tej książki nie jest do końca odpowiedni. Chociaż brzmi on: *Język SQL*, być może bardziej trafny byłby taki: *Logika SQL*. Twierdzę tak dlatego, że — podobnie jak w przypadku wszystkich języków programowania — język SQL ma znacznie więcej wspólnego z chłodną, twardą logiką niż z językiem angielskim.

Niemniej jednak w SQL mamy do czynienia ze składnią opartą na języku, która odróżnia go od wielu innych języków programowania. W przeciwieństwie do wielu narzędzi programowania, SQL wykorzystuje zwykle angielskie słowa, takie jak WHERE, FROM czy

HAVING, które stanowią słowa kluczowe w jego składni. W rezultacie SQL ma charakter bardziej jednoznaczny niż inne języki programowania, z którymi się spotkałeś.

Po zapoznaniu się z językiem SQL możesz więc odnieść wrażenie, że jego polecenia są analogiczne do zdań w języku angielskim i również mają pewne ekspresyjne znaczenie.

Porównaj na przykład poniższe zdanie:

Poproszę na wynos hamburgera i frytki z promocyjnego menu.

z instrukcją SQL:

```
Select miasto, województwo1
from klienci
order by województwo
```

Nie wchodząc na razie w szczegóły, powyższa instrukcja SQL oznacza, że chcemy wyświetlić pola z informacjami o mieście i województwie z tabeli zawierającej dane o klientach, znajdującej się w naszej bazie danych. Ponadto, chcemy posortować wyniki alfabetycznie według województwa.

W obu przypadkach określamy interesujące nas elementy (hamburger/frytki lub miasto/województwo), wskazujemy, skąd chcemy je pozyskać (promocyjne menu lub tabela z informacjami o klientach), oraz zamieszczamy dodatkowe instrukcje (przygotowanie zamówienia na wynos lub sortowanie wyników według województwa).

Ważnym celem tej książki jest nauka SQL w sposób analogiczny do nauki języka angielskiego, za pomocą prostych i intuicyjnych metod. Będę wprowadzał po jednym nowym pojęciu związanym z SQL, jednocześnie rozbudowując jego kontekst logiczny i znaczenie.

Tytuł tej książki ma także inne, być może mniej oczywiste znaczenie. Należy mieć na uwadze fakt, że często mylone są pojęcia *języka SQL* oraz *baz danych SQL*. Istnieje wiele firm komputerowych sprzedających oprogramowanie dla systemów zarządzania bazami danych (ang. *DBMS — Database Management Systems*). Powszechnie bazy danych w tego rodzaju pakietach oprogramowania są często określane jako *bazy danych SQL* (ang. *SQL databases*), jako że język SQL jest podstawowym narzędziem służącym do zarządzania samymi bazami danych i dostępem do danych w nich przechowywanych. W nazwie baz danych niektórych producentów występuje nawet słowo „SQL”. Na przykład firma Microsoft swój najnowszy system zarządzania bazą danych nazwała *SQL Server 2014*.

Jednak, w gruncie rzeczy, SQL jest raczej językiem. Nie jest to baza danych. Dlatego też w niniejszej książce skupię się na scharakteryzowaniu języka SQL, nie zaś na konkretnej bazie danych.

---

<sup>1</sup> W przykładach zawartych w niniejszej książce zarówno w nazwach tabel, jak i kolumn zastosowano polskie znaki diakrytyczne. Nie jest to jednak zalecane w przypadku realizacji rzeczywistych projektów, m.in. ze względu na możliwe problemy z kodowaniem tego rodzaju znaków w realnie istniejących bazach danych — *przyj. tłum.*

## Definicja SQL

Czym więc jest SQL? W skrócie, SQL jest standardowym językiem programowania wykorzystywanym w celu utrzymywania danych zawartych w relacyjnych bazach danych i korzystania z nich. Mówiąc prościej, SQL to język, który pozwala użytkownikom na interakcję z relacyjnymi bazami danych. Począwszy od 1970 roku, przez wiele lat był rozwijany przez różne organizacje. W 1986 roku *Amerykański Instytut Normalizacyjny* (ang. *ANSI — American National Standards Institute*) opublikował swój pierwszy zestaw norm dotyczących języka SQL i od tego czasu wielokrotnie je aktualizował.

Ogólnie rzecz biorąc, język SQL składa się z trzech głównych elementów. Pierwszy z nich nosi nazwę *DML* lub *języka manipulowania danymi* (ang. *Data Manipulation Language*). Obejmuje on zestaw instrukcji wykorzystywanych do pobierania, aktualizacji, dodawania i usuwania danych z bazy danych. Drugi element to *DDL* bądź *język definicji danych* (ang. *Data Definition Language*). Umożliwia on tworzenie lub modyfikowanie struktur bazy danych. Na przykład w ramach języka definicji danych występuje instrukcja *ALTER*, która pozwala modyfikować tabele w bazie danych. Wreszcie trzeci komponent języka SQL to *DCL* lub *język kontroli danych* (ang. *Data Control Language*), pozwalający zarządzać bezpieczeństwem dostępu do obiektów bazy danych.

Wielu kluczowych producentów oprogramowania, takich jak Microsoft i Oracle, dostosowało standard SQL do własnych potrzeb i dodało do niego liczne rozszerzenia i modyfikacje. Jednak mimo że każdy dostawca w wyjątkowy sposób interpretuje SQL, podstawy tego języka pozostają niezmienione i wspólne dla wszystkich producentów oprogramowania. Ten właśnie zakres zostanie omówiony w tej książce.

Jako język programowania, SQL różni się od pozostałych języków, takich jak Visual Basic lub C++, które być może znasz. Inne języki mają zazwyczaj charakter proceduralny. Oznacza to, że umożliwiają określenie pewnych procedur w celu osiągnięcia pożądanego wyniku. SQL jest raczej językiem deklaratywnym, w którym cel do osiągnięcia zazwyczaj deklarowany jest za pomocą pojedynczej instrukcji. W SQL możliwe jest zastosowanie prostszej struktury, ponieważ jest on wykorzystywany jedynie w kontekście relacyjnych baz danych, a nie szeroko rozumianych systemów komputerowych.

## Microsoft SQL Server, Oracle i MySQL

Mimo że moim celem jest przedstawienie podstawowych elementów języka SQL, które są wykorzystywane podczas wszystkich wdrożeń systemów opartych na relacyjnej bazie danych, przedstawię także konkretne przykłady składni SQL. Mając jednak na uwadze fakt, że składnia SQL ustanawiana przez poszczególnych dostawców nieco się od siebie różni, zdecydowałem się skupić na składni języka wykorzystywanej przez następujące trzy bazy danych:

- Microsoft SQL Server,
- Oracle Database,
- MySQL.

Gdy między tymi trzema bazami danych wystąpią jakieś różnice, w tekście książki przedstawiona zostanie składnia stosowana w Microsoft SQL Server. Wszelkie różnice w składni występujące w przypadku MySQL oraz Oracle zostaną zaprezentowane w ramce, tak jak poniżej:

### **RÓŻNICE W RAMACH INNYCH BAZ DANYCH**

---

Tego typu ramka pojawi się zawsze, gdy prezentowane będą różnice w składni w przypadku bazy danych Oracle lub MySQL. Składnia dla Microsoft SQL Server zostanie umieszczona w tekście głównym.

W nagłówku ramki znajdzie się informacja, czy opisane w niej różnice występują w składni MySQL, Oracle czy też w obu tych bazach danych.

---

Microsoft SQL Server jest dostępny w kilku wersjach i edycjach. Najnowsza wersja to *Microsoft SQL Server 2014*. Producent oferuje edycje od podstawowej, o nazwie Express, do edycji Enterprise, zawierającej pełny zakres funkcjonalności. Wersja Express jest darmowa, ale zawiera mnóstwo funkcji, które pozwalają rozpocząć przygodę z budowaniem bazy danych. Wersja Enterprise ma wiele wyrafinowanych funkcji służących do zarządzania bazą danych, a także zaawansowane komponenty do analiz business intelligence.

Oracle również jest dostępny w wielu wersjach. Najnowsza wersja to *Oracle Database 12c*. Jak w przypadku baz danych Microsoft, Oracle również oferuje bezpłatną edycję Express swojej bazy danych.

MySQL to baza danych typu open source. Oznacza to, że nie jest własnością żadnej organizacji. Żaden podmiot nie sprawuje również pieczy nad jej rozwojem. Chociaż Sun Microsystems w 2008 roku zakupił MySQL, baza ta pozostaje jednym z najczęściej wybieranych rodzajów oprogramowania typu open source. Sun Microsystems został później zakupiony przez Oracle. Jako baza danych typu open source, MySQL jest dostępna na wielu platformach innych niż Windows, takich jak Mac OS X i Linux. Community Edition to baza danych MySQL, którą można pobrać za darmo.

Zaczynając pracę z bazami danych, czasami warto wcześniej pobrać wybraną bazę danych, aby móc poćwiczyć budowanie instrukcji SQL z wykorzystaniem znajdujących się w niej tabel. Jednak w celu przyswojenia wiedzy zawartej w niniejszej książce nie musisz tego robić. Została ona napisana w taki sposób, abyś nauczył się posługiwania językiem SQL w trakcie jej czytania. W tekście zamieszczę ponadto wystarczającą ilość danych, które umożliwią Ci zrozumienie wyników różnych instrukcji SQL bez konieczności pobierania oprogramowania i samodzielnego wpisywania i wykonywania prezentowanych instrukcji.

Niemniej jednak, jeśli chciałbyś pobrać darmowe wersje którejkolwiek z wymienionych baz danych, pierwsze trzy dodatki na końcu tej książki zawierają kilka przydatnych wskazówek i porad, jak to zrobić. Dodatek A przedstawia kompletne informacje o tym, jak rozpocząć pracę z Microsoft SQL Server. Zawarta w nim instrukcja prezentuje szczegółowe informacje na temat instalowania oprogramowania i wykonywania poleceń SQL. Analogicznie, dodatek B dotyczy bazy danych MySQL, zaś dodatek C objaśnia sposób postępowania w przypadku bazy danych Oracle.

Dodatek D zawiera materiał dodatkowy — listę wszystkich poleceń SQL przedstawionych w tej książce, występujących we wszystkich trzech bazach danych. Jak wspomniano wcześniej, wszystkie zawarte tu instrukcje SQL będą prezentowane z uwzględnieniem składni Microsoft SQL Server. W większości przypadków instrukcje te będą również działać w MySQL i Oracle, ale istnieją pewne odstępstwa.

Większość czytelników uzna za całkowicie zbędne zarówno pobieranie oprogramowania, jak i zapoznawanie się z zawartością dodatku D. Przykłady pokazane w książce są oczywiste i nie wymagają zaglądania do dodatkowych źródeł wiedzy w celu zrozumienia prezentowanego materiału. Jednak jeżeli masz taką potrzebę, zachęcamy Cię do skorzystania z dodatkowych materiałów w dodatku D.

## Inne bazy danych

Poza Microsoft SQL Server, Oracle i MySQL istnieje wiele innych implementacji bazy danych SQL. Niektóre z bardziej popularnych pakietów to:

- Microsoft Access firmy Microsoft,
- DB2 firmy IBM,
- Informix, również stworzony przez IBM,
- SQL Anywhere firmy Sybase,
- PostgreSQL, która jest bazą danych typu open source.

Z wymienionych powyżej baz danych dość wyjątkowy jest Microsoft Access. W szczególności przydaje się osobom początkującym w świecie SQL, które chcą nauczyć się tego języka. W istocie, Access jest graficznym interfejsem dla relacyjnych baz danych. Innymi słowy, umożliwia on utworzenie kwerendy do relacyjnej bazy danych przy wykorzystaniu interfejsu graficznego. Użytecznym aspektem Accessa dla początkujących jest możliwość łatwego utworzenia kwerendy poprzez jej wizualizację, a następnie przełączenie się na widok SQL, aby zobaczyć właśnie utworzoną instrukcję SQL. Można w ten sposób tworzyć różne kwerendy, a następnie od razu sprawdzać, jak wygląda ich składnia SQL.

Innym wyróżnikiem Accessa w porównaniu do wymienionych powyżej baz danych jest to, że baza ta jest instalowana lokalnie i jako taka wykazuje dużą elastyczność. Można ją wykorzystać nie tylko do utworzenia bazy danych znajdującej się w całości

w jednym pliku na Twoim komputerze, ale również do podłączenia się do bardziej skomplikowanych baz danych utworzonych za pomocą innych narzędzi, takich jak Microsoft SQL Server.

## Relacyjne bazy danych

Omówmy teraz podstawy relacyjnych baz danych oraz sposób ich działania.

Zasadniczo, relacyjna baza danych jest zbiorem danych przechowywanych w dowolnej liczbie tabel. Termin *relacyjne* (ang. *relational*) jest wykorzystywany w celu wskazania, że tabele są ze sobą powiązane. Weźmy pod uwagę prosty przykład bazy danych składającej się tylko z dwóch tabel: Klienci i Zamówienia. Tabela Klienci zawiera po jednym rekordzie dla każdego klienta, który kiedykolwiek złożył zamówienie. Tabela Zamówienia zawiera jeden rekord dla każdego złożonego zamówienia. Każda tabela może mieć dowolną liczbę pól, które są używane do przechowywania różnych atrybutów związanych z każdym rekordem. Na przykład tabela Klient może zawierać takie pola jak Imię czy Nazwisko.

W tym momencie przydatna może okazać się wizualizacja kilku tabel i zawartych w nich danych. Zwyczajowo tabele przedstawia się jako siatkę składającą się z wierszy i kolumn. Każdy wiersz oznacza rekord, natomiast każda kolumna reprezentuje pole w tabeli. Górny wiersz w nagłówku zazwyczaj zawiera nazwy pól. W pozostałych wierszach znajdują się właściwe dane.

W terminologii SQL rekordy i pola w rzeczywistości noszą nazwę *wierszy* (ang. *rows*) i *kolumn* (ang. *columns*), odpowiadając ich wizualnej reprezentacji. Odtąd zatem w celu opisu budowy tabel w relacyjnych bazach danych będziemy używali terminów *wiersze* i *kolumny* zamiast *rekordy* i *pola*.

Spójrzmy na przykład najprostszej możliwej relacyjnej bazy danych. W tej bazie danych istnieją tylko dwie tabele, Klienci i Zamówienia. Oto, jak mogłyby wyglądać:

Tabela Klienci:

IDKlienta	Imię	Nazwisko
1	Jan	Kowalski
2	Andrzej	Nowak
3	Anna	Kwiatkowska

Tabela Zamówienia:

IDZamówienia	IDKlienta	KwotaZamówienia
1	1	50,00
2	1	60,00
3	2	33,50
4	3	20,00

W powyższym przykładzie tabela *Klienci* zawiera trzy kolumny: *IDKlienta*, *Imię* i *Nazwisko*. Każdy z trzech wierszy tabeli prezentuje dane trzech różnych osób: Jana Kowalskiego, Andrzeja Nowaka i Anny Kwiatkowskiej. Każdy wiersz reprezentuje innego klienta, a każda kolumna zawiera inny fragment informacji o nim. Podobnie, w tabeli *Zamówienia* znajdują się trzy kolumny, ale cztery wiersze. Oznacza to, że w bazie danych istnieją cztery zamówienia i trzy przypisane do nich atrybuty.

Oczywiście, jest to bardzo prosty przykład, mający za zadanie pokazać, jaki typ danych może być przechowywany w rzeczywistej bazie danych. Na przykład, tabela *Klienci* zazwyczaj zawiera wiele dodatkowych kolumn, opisujących inne atrybuty klienta, takie jak miasto, województwo, kod pocztowy i telefon. Podobnie tabela *Zamówienia* zazwyczaj zawiera kolumny opisujące dodatkowe atrybuty zamówienia, takie jak data zamówienia, wartość podatku obrotowego oraz informacje o sprzedawcy, który przyjął zamówienie.

## Klucze główne i obce

Zwróć uwagę na pierwszą kolumnę w każdej tabeli: *IDKlienta* w tabeli *Klienci* i *IDZamówienia* w tabeli *Zamówienia*. Kolumny te są zwykle określane jako *klucze główne* (ang. *primary keys*). Klucze główne są przydatne, a wręcz niezbędne z dwóch powodów. Po pierwsze, pozwalają one jednoznacznie zidentyfikować pojedynczy wiersz w tabeli. Na przykład, gdybyś chciał pobrać wiersz dla Jana Kowalskiego, wystarczy po prostu użyć kolumny *IDKlienta*, aby wyświetlić takie dane. Klucze główne zapewniają również unikalność. Oznaczenie kolumny *IDKlienta* jako klucza głównego gwarantuje, że w tej kolumnie znajdować się będzie unikalna wartość dla każdego wiersza w tabeli. Nawet jeśli w bazie danych znajdują się dwie osoby o takim samym imieniu i nazwisku, na przykład Jan Kowalski, w obu tych wierszach w kolumnie *IDKlienta* będą występowały różne wartości.

W powyższym przykładzie wartości w kolumnach z kluczem głównym nie oznaczają niczego szczególnego. W tabeli *Klienci* kolumna *IDKlienta* zawiera wartości 1, 2 i 3 dla trzech wierszy w tabeli. Często jest tak, że tabele bazy danych są zaprojektowane w taki sposób, aby kolumny z kluczem głównym wraz z dodawaniem kolejnych wierszy wypełniane były automatycznie generowanymi numerami sekwencyjnymi. Ta cecha projektowa zwykle jest określana jako *automatyczny przyrost* (ang. *auto-increment*).

Drugim powodem zastosowania kluczy głównych jest to, że pozwalają w łatwy sposób połączyć jedną tabelę z inną. W naszym przykładzie kolumna *IDKlienta* w tabeli *Zamówienia* wskazuje na odpowiadający jej wiersz w tabeli *Klienci*. Patrząc na czwarty wiersz tabeli *Zamówienia*, można zauważyć, że w kolumnie *IDKlienta* występuje wartość 3. Oznacza to, że to zamówienie odnosi się do klienta z *IDKlienta* o numerze 3, czyli do Anny Kwiatkowskiej. Wykorzystanie wspólnych kolumn między tabelami jest istotnym elementem projektowania w relacyjnych bazach danych.

Oprócz jedynie wskazania tabeli *Klienci*, kolumna *IDKlienta* w tabeli *Zamówienia* może zostać oznaczona jako *klucz obcy* (ang. *foreign key*). Zagadnieniem kluczy obcych zajmę się szczegółowo w rozdziale 18. Na razie po prostu zapamiętaj, że klucze obce mogą być zdefiniowane w celu zapewnienia, że kolumna ma poprawną wartość. Przykładem może być sytuacja, gdy nie chcesz, aby kolumna *IDKlienta* w tabeli *Zamówienia* miała wartość, dla której nie istnieje odpowiednik w kolumnie *IDKlienta* w tabeli *Klienci*. Takie ograniczenie jest możliwe dzięki oznaczeniu kolumny jako klucza obcego.

## Typy danych

Za pomocą kluczy głównych i obcych tworzona jest struktura tabel bazy danych. Dzięki nim tabele są ze sobą poprawnie powiązane, a także możliwy jest dostęp do wszystkich tabel w bazie danych. Inną ważną cechą każdej kolumny w tabeli jest typ przechowywanych w niej danych.

Typy danych są po prostu sposobem definiowania rodzaju danych zawartych w kolumnie. Typ danych trzeba określić dla każdej kolumny w każdej tabeli. Niestety, w ramach różnych relacyjnych baz danych dozwolone jest użycie zróżnicowanych typów danych, które mają określone znaczenie. Na przykład każda z relacyjnych baz danych — Microsoft SQL Server, MySQL i Oracle — ma ponad 30 różnych dozwolonych typów danych.

Omówienie każdego dostępnego typu danych z uwzględnieniem wszelkich związanych z nim niuansów byłoby niemożliwe, nawet jeśli mowa jest tylko o trzech bazach danych podanych powyżej. Dokonam jednak pewnego streszczenia tego tematu, charakteryzując główne kategorie typów danych, które występują w większości baz danych. Gdy tylko zapoznasz się z istotnymi typami danych w tych kategoriach, nie będziesz miał większych problemów z innymi, z którymi możesz się zetknąć w przyszłości.

Ogólnie rzecz ujmując, istnieją trzy fundamentalne typy danych: liczbowy, znakowy oraz daty i czasu.

*Typy danych liczbowych* (ang. *numeric datatypes*) występują pod różnymi postaciami — w formie bitów, liczb całkowitych, dziesiętnych i rzeczywistych. *Bity* (ang. *bits*) są typami danych liczbowych, które pozwalają na określenie tylko dwóch wartości — 0 i 1. Są one często używane w celu określenia, że dany atrybut ma przyjmować wyłącznie wartości typu prawda lub fałsz. Typ danych określający *liczby całkowite* (ang. *integers*) wskazuje na liczby bez miejsc po przecinku, natomiast typy danych dla *liczb dziesiętnych* (ang. *decimals*) mogą zawierać wartości dziesiętne po przecinku. W odróżnieniu od bitów, liczb całkowitych i dziesiętnych, wartości *liczb rzeczywistych* (ang. *real*) są podawane jedynie w przybliżeniu, według wewnętrznie ustalonych zasad. Jedną wyróżniającą cechą wszystkich typów danych liczbowych jest to, że mogą one być uwzględnione w obliczeniach arytmetycznych. Oto kilka reprezentatywnych przykładów typów danych liczbowych z Microsoft SQL Server, MySQL i Oracle.



Ogólny opis	Typ danych w Microsoft SQL Server	Typ danych w MySQL	Typ danych w Oracle	Przykład
bit (ang. <i>bit</i> )	bit	bit	(brak)	1
liczba całkowita (ang. <i>integer</i> )	int	int	number	43
liczba dziesiętna (ang. <i>decimal</i> )	decimal	decimal	number	58,63
liczba rzeczywista (ang. <i>real</i> )	float	float	number	80,62345

Typy danych *znakowych* (ang. *character*) są czasem określane jako *łańcuchy znaków* (ang. *strings*) lub *ciągły znaków* (ang. *character strings*). W odróżnieniu od typów danych liczbowych, znakowe typy danych nie ograniczają się do liczb. Mogą zawierać jakiegokolwiek litery lub cyfry, a nawet znaki specjalne, takie jak gwiazdki. Gdy za pomocą instrukcji SQL uzupełniana jest wartość w kolumnie o typie znakowym, zawsze musi być podawana w pojedynczym cudzysłowie. W przypadku typów danych liczbowych nigdy nie należy używać cudzysłowu. Poniżej znajduje się kilka przykładów prezentujących typy danych znakowych.

Ogólny opis	Typ danych w Microsoft SQL Server	Typ danych w MySQL	Typ danych w Oracle	Przykład
zmienna długość (ang. <i>variable length</i> )	varchar	varchar	varchar2	'Thomas Edison'
stała długość (ang. <i>fixed length</i> )	char	char	char	'60601'

Wygląda na to, że drugi przykład (60601) może być typem danych liczbowych, ponieważ składa się wyłącznie z cyfr. Nie jest to nic niezwykłego. Mimo że kody pocztowe w Stanach Zjednoczonych składają się jedynie z cyfr, zazwyczaj definiowane są jako znakowe typy danych, ponieważ nigdy nie ma potrzeby wykonywania obliczeń arytmetycznych z ich udziałem.

*Typy danych związanych z datą i czasem* (ang. *date/time*) są wykorzystywane do prezentowania dat i czasu. Podobnie jak typy danych znakowych, muszą być podawane w pojedynczym cudzysłowie. Na tym typie danych możliwe jest wykonywanie specjalnych obliczeń; na przykład można użyć specjalnej funkcji, aby obliczyć liczbę dni pomiędzy dwiema datami zawierającymi zarówno datę, jak i czas. Oto kilka przykładów typów danych związanych z datą i czasem:

Ogólny opis	Typ danych w Microsoft SQL Server	Typ danych w MySQL	Typ danych w Oracle	Przykład
data (ang. <i>date</i> )	date	date	(brak)	'2009-07-15'
data i czas (ang. <i>date and time</i> )	datetime	datetime	date	'2009-07-15 08:48:30'

## Wartości NULL

Inną ważną cechą poszczególnych kolumn w tabeli jest to, czy kolumna może zawierać wartości null. Wartość null oznacza, że nie ma danych dla określonego elementu danych. Dosłownie, pole takie nie zawiera żadnych danych. Wartości null nie są tym samym, co spacje i puste pola. Logicznie rzecz ujmując, wartości null i spacje są traktowane inaczej. Niuanse związane z pobieraniem danych, które zawierają wartości null, zostaną szczegółowo omówione w rozdziale 8.

Podczas wyświetlania danych z wartościami null wiele baz danych SQL wyświetli słowo NULL napisane wielkimi literami. Dzieje się tak po to, aby użytkownik wiedział, że dana kolumna zawiera wartość null, a nie spacje. W całej książce będę trzymał się tej konwencji i używał pisowni NULL, aby podkreślić, że reprezentuje ona unikalny typ wartości.

Klucze główne w bazie danych nie mogą zawierać wartości NULL. Jest tak, ponieważ klucze główne, zgodnie z definicją, muszą zawierać unikalne wartości.

## Znaczenie SQL

Zanim zakończymy omawianie tematu relacyjnych baz danych, chciałbym przedstawić trochę historii, aby uświadomić Ci przydatność relacyjnych baz danych i znaczenie SQL.

W epoce kamienia w dziedzinie informatyki (lata 60. XX wieku) dane zazwyczaj przechowywane były na taśmie magnetycznej lub w plikach na dyskach. Programy komputerowe, napisane w językach takich jak FORTRAN i COBOL, zwykle odczytywały dane za pośrednictwem plików wejściowych i dokonywały przetwarzania w trybie jednego rekordu naraz, na koniec przenosząc dane do plików wyjściowych. Przetwarzanie było zawsze złożone, ponieważ procedury musiały zostać podzielone na wiele pojedynczych etapów, obejmujących tabele tymczasowe, sortowanie i wielokrotne przetwarzanie danych do momentu otrzymania prawidłowego wyniku.

W latach 70., wraz z pojawieniem się hierarchicznych i sieciowych baz danych i rozpoczęciem korzystania z nich, dokonał się postęp. Dzięki skomplikowanemu systemowi wewnętrznych wskaźników bazy danych nowszego typu ułatwiły odczytywanie danych. Na przykład, program mógł odczytać rekord z informacjami o kliencie, automatycznie wskazując wszystkie zamówienia danego klienta, a następnie szczegóły każdego z tych zamówień. Jednak w zasadzie odbywało się to nadal zgodnie z zasadą przetwarzania w danym momencie tylko jednego rekordu.

Przed pojawieniem się relacyjnych baz danych głównym problemem nie było to, w jaki sposób dane były przechowywane, ale jak wyglądał dostęp do nich. Prawdziwy przełom w związku z relacyjnymi bazami danych nadszedł wraz z pojawieniem się języka SQL, ponieważ umożliwił on dostęp do danych w zupełnie inny sposób.

W przeciwieństwie do wcześniej stosowanych metod pobierania danych, SQL umożliwiał użytkownikowi dostęp do dużego zbioru danych w tym samym momencie. Za pomocą jednej instrukcji polecenie SQL mogło pobrać lub zaktualizować tysiące rekordów w wielu tabelach. W ten sposób proces ten stał się dużo mniej złożony. Nie były już potrzebne programy komputerowe do odczytywania jednego rekordu naraz w specjalnej sekwencji przy jednoczesnym podejmowaniu decyzji o tym, co należy zrobić z każdym rekordem. To, co do tej pory wymagało setek linii kodu programowania, mogło być teraz realizowane za pomocą zaledwie kilku linii określonej logiki.

## Co dalej?

W tym rozdziale przedstawiono podstawowe informacje o relacyjnych bazach danych, które stanowią podwaliny pod główne zagadnienie, którym będziemy się zajmować, czyli pobieranie danych z baz danych. Omówiliśmy kilka ważnych cech relacyjnych baz danych, takich jak klucze główne, klucze obce i typy danych. Wspomnieliśmy również o tym, że możliwe jest wystąpienie w danych wartości NULL. Uzupełnimy tę wiedzę w rozdziale 8., natomiast w rozdziale 18. powrócimy do ogólnych zagadnień związanych z utrzymaniem bazy danych. Rozdział 19. poświęcony jest projektowaniu baz danych.

Dlaczego najważniejszy temat projektowania baz danych omówiony zostanie w niniejszej książce dopiero kilkanaście rozdziałów dalej? W rzeczywistym świecie bazy danych są projektowane i tworzone, zanim mają miejsce jakiegokolwiek próby pobierania danych. Dlaczego miałbym zatem nie podążać tą samą drogą w tej książce? Krótko mówiąc, doszedłem do wniosku, że o wiele bardziej produktywnie jest zgłębienie zagadnień związanych z SQL bez zaprzątania sobie głowy szczegółami dotyczącymi projektowania baz danych, które nosi znamiona zarówno sztuki, jak i nauki. Ponadto, zasady projektowania baz danych będą znaczyć o wiele więcej po zapoznaniu się ze szczegółami i niuansami związanymi z pobieraniem danych. Na razie więc zignorujemy zagadnienia związane z projektowaniem baz danych i w następnym rozdziale przejdziemy od razu do kwestii pobierania danych.



# SKOROWIDZ



## A

aggregate function, *Patrz:* funkcja agregująca  
aggregation, *Patrz:* agregacja  
agregacja, 99  
    kryteria selekcji, 107  
arkusz kalkulacyjny, 195, 198  
atrybut, 187  
auto-increment, *Patrz:* przyrost automatyczny

## B

baza danych, 15  
    projektowanie, 183, 184, 190  
    relacyjna, 20, 183  
    schemat gwiazdy, 191  
        kostka, 192  
    SQL, 16  
    struktura, 17  
Boolean logic, *Patrz:* logika Boole'a

## C

calculated field, *Patrz:* pole obliczane  
character string, *Patrz:* ciąg znaków  
child table, *Patrz:* tabela podrzędna  
ciąg znaków, 23  
Cognos, 196  
column-based logic, *Patrz:* logika obliczania  
    wartości wyjściowych w kolumnach  
composite primary key, *Patrz:* klucz główny  
    złożony  
concatenation, *Patrz:* konkatencja  
conditional logic, *Patrz:* logika warunkowa

correlated subquery, *Patrz:* podzapytanie  
    skorelowane  
crosstab report, *Patrz:* raport krzyżowy  
Crystal Reports, 195, 196  
cube, *Patrz:* baza danych schemat gwiazdy  
    kostka

## D

dane  
    agregacja, *Patrz:* agregacja  
    aktualizacja, 165, 166, 171, 172, 186  
    duplikat, *Patrz:* duplikat  
    grupowanie, 104, 105  
    modyfikacja, 165, 170, 179  
    nadmiarowe, 186  
    ograniczenie dostępu, 136  
    pobieranie, 27  
    podzbiór, 135  
    prezentacja, 195  
    redundancja, *Patrz:* redundancja  
    selekcja Top N, 73  
    sortowanie, 55  
        w porządku malejącym, 57, 60  
        w porządku rosnącym, 56, 60  
        względem pola obliczanego, 58  
        względem więcej niż jednej kolumny, 58  
typ, 22, 177  
    daty i czasu, 22, 23  
    konwersja, 52  
    liczbowy, 22, 60  
    znakowy, 22, 23, 42, 44, 60  
źródło, 140

Data Control Language, *Patrz:* DCL  
 Data Definition Language, *Patrz:* DDL  
 Data Manipulation Language, *Patrz:* DML  
 Database Management Systems, *Patrz:* DBMS  
 DB2, 19  
 DBMS, 16  
 DCL, 17, 175  
 DDL, 17, 175  
 diagram związków encji, 113  
 DML, 17, 175  
 dopasowywanie  
   do wzorca, 89  
   za pomocą brzmienia, 94  
 drażnienie, 192  
   wskroś, 199  
 drill down, *Patrz:* drażnienie  
 drill through, *Patrz:* drażnienie wskroś  
 duplikat, 152  
   usuwanie, 99, 152

## E

encja, 113  
 entity, *Patrz:* encja  
 entity-relationship diagram, *Patrz:* diagram  
   związków encji  
 Excel, 195, 198  
   tabela przestawna, 198

## F

first normal form, *Patrz:* postać normalna  
   pierwsza  
 foreign key, *Patrz:* klucz obcy  
 funkcja, 41, 132  
   agregująca, 42, 101, 104  
   argument, 43  
   AVG, 101, 102  
   CAST, 52, 53  
   CONCAT, 37, 46  
   COUNT, 101, 103  
   CURRENT\_DATE, 48  
   DATE\_FORMAT, 49  
   DATEDIFF, 48, 49, 50  
   DATEPART, 48, 49  
   daty i czasu, 42, 48  
   DIFFERENCE, 94, 95  
   GETDATE, 48

IFNULL, 54  
 ISNULL, 53, 54, 61, 87, 141  
 konwersji, 52  
 LEFT, 42, 43, 44  
 liczbowa, 42  
 LOWER, 47  
 LTRIM, 42, 45  
 łańcuchowa, 42  
 matematyczna, 50  
 MAX, 101, 102  
 MIN, 101, 102  
 NOW, 48  
 NVL, 54  
 PI, 50, 51  
 RAND, 50, 51  
 RIGHT, 43, 44, 48  
 ROUND, 50  
 RTRIM, 44, 45  
 skalarna, 42, 101  
 SOUNDEX, 94, 95  
 SUBSTR, 44, 45, 48  
 SUBSTRING, 41, 44, 45  
 SUM, 42, 101  
 UPPER, 47  
 zabezpieczeń bazy danych, 136  
 zagnieżdżona, 47, 51  
 znakowa, 42

## G

generator liczb losowych, 51  
 Graphical User Interface, *Patrz:* GUI  
 GUI, 176

## I

identity column, *Patrz:* kolumna identyfikująca  
 indeks, 182  
 Informix, 19  
 inner join, *Patrz:* złączenie wewnętrzne  
 instrukcja, 215  
   ALTER FUNCTION, 163  
   ALTER PROCEDURE, 162, 163  
   ALTER TABLE, 182  
   ALTER VIEW, 136  
   CREATE, 176  
   CREATE FUNCTION, 163  
   CREATE INDEX, 182  
   CREATE PROCEDURE, 162, 175, 176

CREATE TABLE, 180, 181  
 CREATE VIEW, 133, 175, 176  
 DDL, 176  
 DELETE, 139, 170, 175  
 DML, 175  
 DROP FUNCTION, 163  
 DROP INDEX, 182  
 DROP PROCEDURE, 162, 163  
 INSERT, 139, 175  
 INSERT INTO, 168, 169  
 SELECT, 27, 39, 42, 64, 139, 175, 198  
   łączenie, 149  
     ORDER BY, 55, 56, 57, 58  
 TRUNCATE TABLE, 170  
 UPDATE, 139, 171, 172, 175  
 interfejs użytkownika graficzny, *Patrz:* GUI

## J

język

definicji danych, 17  
 deklaratywny, 17  
 kontroli danych, *Patrz:* DCL  
 manipulowania danymi, 17  
 proceduralny, 17  
 SQL, *Patrz:* SQL

join, *Patrz:* łączenie

## K

keyword, *Patrz:* słowo kluczowe

klauzula, *Patrz też:* słowo kluczowe

CREATE PROCEDURE, 159  
 CREATE VIEW, 136  
 DROP VIEW, 136  
 FROM, 43, 45, 56, 140  
 GROUP BY, 145  
 HAVING, 145  
 LEFT JOIN, 121  
 ON, 131  
 ORDER BY, 55, 56, 58, 59, 73, 152  
 WHERE, 67, 69, 71, 74, 75, 103, 107, 123

klucz

główny, 21, 22, 24, 28, 111, 176, 177, 178  
 złożony, 178, 187  
 obcy, 22, 111, 113, 176, 179

kolumna, 20, 177

alias, 35, 37, 38, 43, 59  
 atrybut, 177, 178

identyfikująca, 177  
 logika obliczania wartości wyjściowych,  
   *Patrz:* logika obliczania wartości  
     wyjściowych w kolumnach  
 nagłówkek, 34  
 nazwa, 30, 177  
   zmiana, 135  
 obliczana, 135  
 wartość  
   domyślna, 178  
   przyrostowa, 177, 179  
 wybieranie, 29  
 konkatencja, 36  
 Kreator raportów, 196

## L

liczba

losowa, 51

pi, 51

literal value, *Patrz:* literał wartość

literał, 34, 35, 42

logika

Boole'a, 77

NULL, 86

IF-THEN-ELSE, 64

obliczania wartości wyjściowych  
 w kolumnach, 63

warunkowa, 63

zbiorów, 149

## Ł

łańcuch znaków, 23, 42

## M

matrix report, *Patrz:* raport macierzowy

Microsoft Access, 19

Microsoft Reporting Services, 195, 196

Microsoft SQL Server, 18

instalacja, 203, 204

wersja, 18

MicroStrategy, 196

model relacyjny, 184

MySQL, 18

instalacja, 207, 208

MySQL Server, 207

MySQL Workbench, 207, 209

**N**

normalizacja, 184, 185, 186, 187, 188, 191  
zasady, 187  
NULL, 24, 52, 53, 60, 177  
numeric datatype, *Patrz:* dane typ liczbowy

**O**

obiekt, 132  
  biznesowy, 111  
  pakiet, *Patrz:* pakiet  
operator  
  AND, 78, 80  
  arytmetyczny, 36  
  BETWEEN, 84  
  EXCEPT, 155  
  EXISTS, 146, 173  
  IN, 84, 85, 143  
  INTERSECT, 154, 155  
  kolejność, 80  
  LIKE, 89  
  MINUS, 155  
  NOT, 78, 82, 84  
  OR, 78, 79, 80, 154  
  UNION, 150, 151, 154  
  UNION ALL, 152, 154  
  NOT, 84  
Oracle, 211  
Oracle Database, 18  
Oracle Database Express Edition, 212  
  instalacja, 211  
outer join, *Patrz:* złączenie zewnętrzne

**P**

pakiet, 159  
parametr, 160  
parent table, *Patrz:* tabela nadrzędna  
pattern matching, *Patrz:* dopasowywanie  
  do wzorca  
pivot table, *Patrz:* Excel tabela przestawna  
podzapytanie, 39, 75, 86, 139, 140  
  nieskorelowane, 144  
  skorelowane, 144, 145, 172, 173  
  w miejscu  
    listy\_kolumn, 140  
    listy\_tabel, 140  
    warunku, 140, 143  
wartość w kolumnie obliczanej, 147

pole obliczane, 33, 34, 35, 58  
  w klauzuli ORDER BY, 59  
postać normalna  
  druga, 186  
  pierwsza, 186  
  trzecia, 186  
PostgreSQL, 19  
primary key, *Patrz:* klucz główny  
procedura  
  składowana, 157, 162, 176  
  modyfikowanie, 162  
  parametry, 160, 163  
  tworzenie, 158, 159  
  wykonywanie, 161, 163  
przyrost automatyczny, 21, 177

**Q**

Query Builder, 196

**R**

raport  
  krzyżowy, 196  
  macierzowy, 196, 198  
  tabelaryczny, 196  
  tworzenie, 196  
raportowanie, 195  
redundancja, 186, 191  
relationship, *Patrz:* związek  
Report Wizard, *Patrz:* Kreator raportów

**S**

samogłoska, 94  
second normal form, *Patrz:* postać normalna  
  druga  
sekwencja, 177  
selekcja, 140, 143  
self join, *Patrz:* złączenie zwrotne  
set logic, *Patrz:* logika zbiorów  
słowo kluczowe, 27, 28, *Patrz też:* klauzula  
  ALTER FUNCTION, 163  
  ALTER PROCEDURE, 162, 163  
  AND, 78  
  AS, 39, 117  
  ASC, 57  
  BEGIN, 159  
  BETWEEN, 84  
  CALL, 162



CASE, 63  
 CREATE FUNCTION, 163  
 CREATE PROCEDURE, 158, 159, 162  
 CREATE VIEW, 132, 136  
 DESC, 57  
 DISTINCT, 99, 152  
 DROP FUNCTION, 163  
 DROP PROCEDURE, 162, 163  
 DROP VIEW, 136  
 ELSE, 64  
 END, 64, 159  
 EXEC, 161  
 EXISTS, 146  
 FROM, 27, 43, 56, 116, 117, 140  
 GROUP BY, 104, 105  
 HAVING, 107  
 IN, 84  
 INNER JOIN, 114, 115, 116, 117  
 INSERT, 166  
 IS NOT NULL, 87  
 IS NULL, 86  
 LIMIT, 73  
 NOT, 78, 82, 86  
 NULLS FIRST, 60  
 ON, 114, 116  
 OR, 78, 79  
 ROWNUM, 73, 75  
 SELECT, 27  
 THEN, 64  
 TOP, 72, 73  
 VALUES, 167, 168  
 WHEN, 64  
 WHERE, 69, 70, 116  
     operator, 71  
 SQL, 15, 16  
     baza danych, 16  
     definicja, 17  
     język, 16  
     składnia, 17, 29, 30  
 SQL Anywhere, 19  
 SQL Server Express 2014, 203  
 SQL Server Management Studio, 203, 204  
 stored procedure, *Patrz:* procedura składowana  
 string, *Patrz:* łańcuch znaków  
 string function, *Patrz:* funkcja łańcuchowa  
 subquery, *Patrz:* podzapytanie  
 system zarządzania bazami danych, *Patrz:* DBMS

## T

tabela, 111, 132  
     alias, 39, 117  
     atrybut, 176, 177  
     DUAL, 45  
     faktów, 192  
     indeks, 178  
     łączenie, 112, 149  
     nadrzędna, 179  
     podrzędna, 179  
     tworzenie, 175, 177, 180, 181  
     widok, *Patrz:* widok  
     złączenie, 152, *Patrz:* złączenie  
 tabular report, *Patrz:* raport tabelaryczny  
 third normal form, *Patrz:* postać normalna  
     trzecia

## U

uncorrelated subquery, *Patrz:* podzapytanie  
     nieskorelowane

## V

variable, *Patrz:* zmienna

## W

wartość NULL, *Patrz:* NULL  
 widok, 132, 134, 176  
     dane, 133  
     modyfikowanie, 136  
     usuwanie, 137  
 wiersz, 20  
     sortowanie, 55  
     usuwanie, 165, 170, 179  
         miękkie, 165  
     wstawianie, 165, 166, 167  
     zliczanie, 103  
 wildcard, *Patrz:* znak wieloznaczny  
 wyrażenie  
     CASE, 63, 64, 67  
     proste, 64  
     przeszukujące, 64, 66  
 wyzwalacz, 132, 177

## Z

- zapytanie, 196
- złączenie, 112
  - pełne, 126
  - prawostronne, 124
  - wewnętrzne, 114, 115, 116
    - kolejność tabel, 115
  - zewewnętrzne, 119, 120
    - FULL JOIN, 120, 127
    - FULL OUTER JOIN, 120
    - INNER JOIN, 120
    - kolejność tabel, 125
    - LEFT JOIN, 120, 121
    - LEFT OUTER JOIN, 120
    - RIGHT JOIN, 120, 124
    - RIGHT OUTER JOIN, 120
  - zwrotne, 129
- zmienna, 160
- znak
  - \$\$, 159
  - %, *Patrz:* znak procentu
  - \*, *Patrz:* znak gwiazdki
  - /, 36
  - ;, 29, 30
  - [ ], *Patrz:* znak nawias kwadratowy
  - ^, *Patrz:* znak daszka
  - \_, *Patrz:* znak podkreślenia
  - `, *Patrz:* znak akcentu słabego
  - ||, 37
  - +, *Patrz:* znak plus
  - <, 84
  - <=, 84
  - =, *Patrz:* znak równości
  - >, 84
  - >=, 84
  - akcentu słabego, 134
  - cudzysłów, 134
  - daszka, 92
  - gwiazdki, 36, 103, 114, 117
  - nawias kwadratowy, 92
  - plus, 37
  - podkreślenia, 92
  - procentu, 90, 92
  - równości, 71
  - średnika, 159
  - wieloznaczny, 89, 92
  - związek, 113

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



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

Zmień swoją stronę WWW  
w działający bankomat!

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

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

GRUPA WYDAWNICZA

 **Helion SA**

# JĘZYK SQL

## PRZYJAZNY PODRĘCZNIK

**Bazy danych SQL** to fundament większości współczesnych systemów informatycznych. Przechowują niewyobrażalne ilości danych i pozwalają na błyskawiczny dostęp do nich. Zastanawiasz się, jak skutecznie wybrać z Twojej bazy potrzebne dane? Chcesz połączyć dane z wielu tabel i wykonać na nich obliczenia? Na to wszystko i mnóstwo innych rzeczy pozwoli Ci język SQL!

**Na rynku dostępnych jest wiele książek poświęconych SQL**, zazwyczaj jednak skupiają się one na składni. Ten podręcznik jest zupełnie inny! Podczas lektury nie musisz instalować żadnego oprogramowania ani używać komputera. Zastanawiasz się, jak to możliwe? W tej książce znajdziesz przykłady SQL, które da się zrozumieć, po prostu je czytając. Sięgnij po nią i naucz się korzystać z funkcji, aliasów, złączeń oraz warunków logicznych. Ponadto zobacz, jak stosować procedury składowane i podzapytania. Z ostatnich rozdziałów dowiesz się, jak modyfikować dane, utrzymywać tabele oraz projektować bazy danych. To doskonała książka prezentująca język SQL od nowej strony.

### Dzięki tej książce:

- poznasz składnię języka SQL
- nauczysz się pobierać dane spełniające Twoje kryteria
- wykorzystasz procedury składowane
- opanujesz najlepsze praktyki w projektowaniu baz danych
- będziesz swobodnie używać SQL w codziennej pracy

### Twój przewodnik po języku SQL!

**helion.pl**  
księgarnia  
internetowa

Nr katalogowy: 25826



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/nowosci>

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

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-246-9354-2



9 788324 693542

Cena: 39,90 zł

Informatyka w najlepszym wydaniu