

Ben Forta

SQL

W MGNIENIU OKA

OPANUJ JĘZYK ZAPYTAŃ W



MINUT DZIENNIE

Wydanie IV

Helion 

Tytuł oryginału: Sams Teach Yourself SQL in 10 Minutes, Fourth Edition

Tłumaczenie: Rafał Jońca

ISBN: 978-83-283-1089-6

Authorized translation from the English language edition: SAMS TEACH YOURSELF SQL IN 10 MINUTES, Fourth Edition; ISBN 0672336073; by Ben Forta; published by Pearson Education, Inc; publishing as SAMS Publishing.

Copyright © 2013 by Pearson Education, Inc.

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 © 2015.

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

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

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/sqldko.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

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

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

Wprowadzenie	11
Do kogo kierowana jest ta książka?	12
SZBD opisane w książce	12
Konwencje typograficzne stosowane w książce	12
Rozdział 1. Podstawy języka SQL	15
Podstawy baz danych	15
Język SQL	20
Ćwicz	21
Podsumowanie	21
Rozdział 2. Pobieranie danych	23
Instrukcja SELECT	23
Pobranie konkretnej kolumny	24
Pobranie wielu kolumn	25
Pobranie wszystkich kolumn	27
Pobieranie jedynie unikatowych wierszy	27
Ograniczenie liczby zwracanych wyników	29
Komentarze	31
Podsumowanie	33
Rozdział 3. Sortowanie otrzymywanych danych	35
Sortowanie danych	35
Sortowanie względem wielu kolumn	37
Sortowanie względem położenia kolumny	37
Określenie kierunku sortowania	39
Podsumowanie	40
Rozdział 4. Filtrowanie danych	41
Stosowanie frazy WHERE	41
Operatory frazy WHERE	42
Podsumowanie	47

Rozdział 5.	Zaawansowane filtrowanie danych	49
	Łączenie fraz WHERE	49
	Operator IN	53
	Operator NOT	55
	Podsumowanie	56
Rozdział 6.	Filtrowanie za pomocą znaków wieloznacznych	57
	Korzystanie z operatora LIKE	57
	Wskazówki dotyczące używania znaków wieloznacznych	63
	Podsumowanie	64
Rozdział 7.	Tworzenie pól obliczanych	65
	Pojęcie pól obliczanych	65
	Konkatenacja pól	66
	Przeprowadzanie obliczeń matematycznych	71
	Podsumowanie	72
Rozdział 8.	Modyfikacja danych za pomocą funkcji	73
	Czym są funkcje?	73
	Stosowanie funkcji	74
	Podsumowanie	80
Rozdział 9.	Funkcje agregujące	81
	Funkcje agregujące	81
	Agregacja tylko różnorodnych wartości	87
	Łączenie funkcji agregujących	89
	Podsumowanie	89
Rozdział 10.	Grupowanie danych	91
	Omówienie grupowania danych	91
	Tworzenie grup	92
	Filtrowanie grup	93
	Grupowanie i sortowanie	96
	Kolejność fraz instrukcji SELECT	98
	Podsumowanie	98
Rozdział 11.	Zapytania zagnieżdżone	99
	Zagnieżdżanie zapytań	99
	Filtrowanie na podstawie zapytań zagnieżdżonych	99
	Zapytania zagnieżdżone jako pola obliczane	103
	Podsumowanie	106

Rozdział 12. Łączenie tabel	107
Czym są złączenia?	107
Tworzenie złączeń	109
Podsumowanie	116
Rozdział 13. Tworzenie rozbudowanych złączeń	117
Stosowanie aliasów tabel	117
Typy złączeń	118
Złączenia i funkcje agregujące	123
Złączenia i ich warunki	125
Podsumowanie	125
Rozdział 14. Łączenie zapytań	127
Łączenie zapytań	127
Tworzenie unii	127
Podsumowanie	132
Rozdział 15. Wstawianie danych	133
Wstawianie danych	133
Kopiowanie z jednej tabeli do innej	139
Podsumowanie	140
Rozdział 16. Aktualizacja i usuwanie danych	141
Aktualizacja danych	141
Wskazówki związane z aktualizacją lub usuwaniem danych	145
Podsumowanie	145
Rozdział 17. Tworzenie i modyfikacja tabel	147
Tworzenie tabel	147
Aktualizacja tabel	152
Usuwanie tabel	154
Zmiana nazwy tablicy	154
Podsumowanie	154
Rozdział 18. Stosowanie perspektyw	155
Perspektywy	155
Tworzenie widoków	158
Podsumowanie	164

Rozdział 19.	Korzystanie z zapamiętanych procedur	165
	Zapamiętane procedury	165
	Wykonywanie zapamiętanych procedur	167
	Tworzenie zapamiętanych procedur	169
	Podsumowanie	172
Rozdział 20.	Zarządzanie transakcjami	173
	Działanie transakcji	173
	Sterowanie transakcjami	175
	Podsumowanie	179
Rozdział 21.	Kursory	181
	Działanie kursorów	181
	Praca z kursorami	182
	Podsumowanie	186
Rozdział 22.	Zaawansowane funkcje języka SQL	187
	Ograniczenia	187
	Omówienie indeksów	193
	Wyzwalacze	195
	Bezpieczeństwo baz danych	196
	Podsumowanie	197
Dodatek A	Skrypty przykładowych tabel	199
	Omówienie przykładowych tabel	199
	Pobranie przykładowych tabel i skryptów	203
Dodatek B	Praca z popularnymi aplikacjami	205
	Apache Open Office Base	205
	Adobe ColdFusion	206
	IBM DB2	206
	MariaDB	206
	Microsoft Access	207
	Microsoft ASP	207
	Microsoft ASP.NET	208
	Microsoft Query	208
	Microsoft SQL Server (wraz z Microsoft SQL Server Express)	209
	MySQL	209
	Oracle	210

	Oracle Express	211
	PHP	211
	PostgreSQL	212
	SQLite	213
	Konfiguracja źródeł danych ODBC	213
Dodatek C	Składnia instrukcji SQL	215
	ALTER TABLE	215
	COMMIT	216
	CREATE INDEX	216
	CREATE PROCEDURE	216
	CREATE TABLE	216
	CREATE VIEW	217
	DELETE	217
	DROP	217
	INSERT	218
	INSERT SELECT	218
	ROLLBACK	218
	SELECT	218
	UPDATE	219
Dodatek D	Typy danych języka SQL	221
	Tekstowe typy danych	221
	Numeryczne typy danych	223
	Typy danych daty i czasu	224
	Binarne typy danych	225
Dodatek E	Słowa kluczowe języka SQL	227
	Skorowidz	231

Rozdział 10

Grupowanie danych

W tym rozdziale opisana jest funkcja grupowania danych, która umożliwia podsumowywanie podzbiorów tabeli. Wprowadzone są też dwie nowe frazy instrukcji SELECT: GROUP BY i HAVING.

Omówienie grupowania danych

W poprzednim rozdziale opisywałem funkcje agregujące języka SQL używane do tworzenia podsumowań danych. Umożliwiały one zliczanie wierszy, obliczanie sumy i średniej, a także znajdowanie wartości największej i najmniejszej. Wszystko to odbywało się bez potrzeby pobierania wszystkich danych.

Do tej pory obliczenia przeprowadzane były na wszystkich danych w tabeli lub na danych spełniających warunek określony we frazie WHERE. Oto krótkie przypomnienie — poniższy przykład zwraca liczbę wszystkich produktów oferowanych przez dostawcę DLL01:

Wejście ▼

```
SELECT COUNT(*) AS liczba_prod
FROM Produkty
WHERE dost_id = 'DLL01';
```

Wyjście ▼

```
liczba_prod
-----
4
```

W jaki sposób pobrać liczbę produktów oferowanych przez poszczególnych producentów? Jak uzyskać dane dotyczące tylko tych producentów, którzy oferują jeden produkt lub oferują powyżej 10 produktów?

W takiej sytuacji niezastąpione jest grupowanie. Umożliwia ono podzielenie danych na logiczne zbiory i przeprowadzenie funkcji agregujących na każdej z grup osobno.

Tworzenie grup

Grupy tworzy się za pomocą frazy `GROUP BY` w instrukcji `SELECT`. Najłatwiej zrozumieć to na przykładzie:

Wejście ▼

```
SELECT dost_id, COUNT(*) AS liczba_prod
FROM Produkty
GROUP BY dost_id;
```

Wyjście ▼

dost_id	liczba_prod
BRS01	3
DLL01	4
FNG01	2

Analiza ▼

Powyższa instrukcja `SELECT` określa dwie kolumny, `dost_id` (która zawiera identyfikator dostawcy) i `liczba_prod` (która jest polem obliczanym za pomocą funkcji agregującej `COUNT(*)`). Fraza `GROUP BY` sprawia, że SZBD sortuje wyniki i grupuje je względem `dost_id`. Powoduje to obliczanie `liczba_prod` dla każdego unikalnego `dost_id` zamiast zbiorowo dla całej tabeli. W ten sposób można się dowiedzieć, iż dostawca BRS01 oferuje 3 produkty, dostawca DLL01 4 produkty, a dostawca FNG01 2 produkty.

Ponieważ pojawiła się fraza `GROUP BY`, nie trzeba było określać każdej grupy, by poznać jej wartość. Wszystko zostało wykonane automatycznie. Fraza `GROUP BY` powoduje, że system zarządzania najpierw grupuje dane, a następnie przeprowadza funkcję agregującą osobno dla każdej grupy zamiast dla wszystkich wyników.

Zanim jednak rozpocznie się stosowanie frazy `GROUP BY`, warto dokładniej poznać jej działanie:

- ▶ Fraza `GROUP BY` może zawierać dowolną liczbę kolumn. Umożliwia to tworzenie zagnieżdżonych grup, a tym samym bardziej szczegółowe opracowywanie danych.
- ▶ W przypadku wprowadzenia kilku nazw kolumn we frazie, dane podsumowywane są dla ostatniej określonej kolumny (dla jej grup). Oznacza to, że nie jest możliwe uzyskanie danych dla wszystkich wymienionych kolumn.

- ▶ Wszystkie kolumny wymienione we frazie GROUP BY muszą być kolumnami pobieranymi z bazy lub pełnymi wyrażeniami (ale nie funkcjami agregującymi). Jeśli wyrażenie występuje po SELECT, w takiej samej postaci musi się znaleźć we frazie GROUP BY. Nie można w tym wypadku stosować aliasów.
- ▶ Większość implementacji SQL nie dopuszcza, aby we frazie GROUP BY znalazły się kolumny typów danych o zmiennej długości (na przykład pola tekstowe lub memo).
- ▶ Wszystkie kolumny występujące w instrukcji SELECT muszą się także znaleźć we frazie GROUP BY (nie dotyczy to funkcji agregujących).
- ▶ Jeśli grupowana kolumna zawiera wiersz z wartością NULL, powstanie osobna grupa o nazwie NULL. Jeśli istnieje kilka wierszy o wartości NULL, zostaną one scalone w jedną grupę.
- ▶ W przypadku występowania dodatkowych fraz, funkcja GROUP BY musi się pojawić po frazie WHERE, ale przed ORDER BY.

Fraza ALL

Pewne implementacje SQL (na przykład Microsoft SQL Server) obsługują opcjonalną frazę ALL dla GROUP BY. Fraza ta może posłużyć do zwrócenia wszystkich grup, nawet tych, dla których agregacja spowodowałaby zwrócenie wartości NULL. Szczegółów należy szukać w dokumentacji SZBD.

Wskazówka
Wskazówka

Określanie kolumn za pomocą położenia względnych

Niektóre implementacje SQL umożliwiają podanie we frazie GROUP BY położenia kolumn z listy SELECT. Można wtedy na przykład napisać GROUP BY 2, 1, aby grupowanie najpierw odbyło się względem drugiej kolumny, a następnie pierwszej. Choć ten skrótowy zapis jest bardzo kuszący, nie obsługują go wszystkie implementacje, a dodatkowo niesie ze sobą ryzyko pojawienia się błędów po zmianie kolejności kolumn.

Ostrzeżenie
Ostrzeżenie

Filtrowanie grup

Poza samą możliwością grupowania danych język SQL oferuje także filtrowanie na podstawie danych zebranych dla poszczególnych grup. Na przykład można wyświetlić wszystkich klientów, którzy dokonali przynajmniej dwóch zamówień. Takie filtrowanie musi się odbywać wobec pełnych grup, a nie poszczególnych wierszy.

Nie jest możliwe posłużenie się tutaj frazą WHERE opisaną w rozdziale 4., gdyż powoduje ona filtrowanie wierszy, i to jeszcze przed rozpoczęciem grupowania. Inaczej mówiąc, fraza WHERE nie wie, czym jest grupowanie.

Jaka jest więc alternatywa dla WHERE? Język SQL wprowadza dodatkową frazę HAVING. Jest ona bardzo podobna do WHERE. W zasadzie wszystkie opisane do tej pory techniki filtrowania związane z WHERE mogą zostać także użyte we frazie HAVING. Jedyna różnica polega na tym, iż WHERE filtruje wiersze, a HAVING grupy.

Wskazówka

Fraza HAVING obsługuje wszystkie operatory frazy WHERE

Rozdziały 4. i 5. opisywały proste i zaawansowane filtrowanie danych (włączając w to znaki wieloznaczne i użycie wielu operatorów). Wszystkie opisane tam operatory można z powodzeniem stosować we frazie HAVING. Składnia jest identyczna, zmienia się tylko słowo kluczowe.

W jaki sposób filtrować grupy? Oto przykład:

Wejście ▼

```
SELECT k1_id, COUNT(*) AS zamowienia
FROM Zamowienia
GROUP BY k1_id
HAVING COUNT(*) >= 2;
```

Wyjście ▼

k1_id	zamowienia
1000000001	2

Analiza ▼

Pierwsze trzy wiersze są bardzo podobne do wcześniejszego przykładu z tego rozdziału. Ostatni wiersz wprowadza frazę HAVING, która przepuszcza tylko te grupy, które posiadają minimum dwa zamówienia — `COUNT(*) >= 2`.

Można się przekonać, iż fraza WHERE nie przeprowadziłaby poprawnego filtrowania, ponieważ musi ono bazować na wartości z agregacji grupy, a nie wartościach znajdujących się w poszczególnych wierszach.

Uwaga

Różnica między HAVING i WHERE

Można na to zagadnienie spojrzeć inaczej. Fraza WHERE filtruje dane przed grupowaniem, natomiast HAVING po. Jest to ważna różnica — wiersze wyeliminowane przez frazę WHERE w ogóle nie zostaną wzięte pod uwagę przy tworzeniu grup. Powoduje to zmianę wartości pól obliczanych, a tym samym zmianę wyświetlanych grup, gdy przeprowadzane jest filtrowanie.

Czy zachodzi potrzeba jednoczesnego stosowania fraz WHERE i HAVING w jednym zapytaniu? W pewnych sytuacjach jest ona nawet konieczna. Przypuśćmy na przykład, iż poprzednie zapytanie powinno zwrócić dowolnych klientów z więcej niż jednym zamówieniem, ale pod uwagę należy brać tylko ostatnich rok. W takiej sytuacji fraza WHERE spowoduje przeanalizowanie zamówień tylko z ostatnich 12 miesięcy, a fraza HAVING wskaże tylko klientów z co najmniej dwoma zamówieniami.

Oto inny przykład. Zapytanie powoduje wyświetlenie listy dostawców z więcej niż jednym produktem o cenie powyżej 10 złotych.

Wejście ▼

```
SELECT dost_id, COUNT(*) AS liczba_prod
FROM Produkty
WHERE prod_cena >= 10
GROUP BY dost_id
HAVING COUNT(*) >= 2;
```

Wyjście ▼

dost_id	liczba_prod
-----	-----
BRS01	3
FNG01	2

Analiza ▼

Zapytanie to wymaga krótkiego wyjaśnienia. Pierwszy wiersz to prosta instrukcja SELECT z funkcją agregującą — podobnie jak w poprzednich przykładach. Fraza WHERE filtruje wszystkie wiersze, których prod_cena jest mniejsza od 10 złotych. Następnie dane są grupowane pod kątem dost_id, a fraza HAVING przepuszcza tylko grupy zawierające więcej niż jedno zamówienie. Bez frazy WHERE zostałyby pobrane dodatkowy wiersz, ponieważ dostawca DLL01 sprzedaje wszystkie cztery produkty po cenie poniżej 10 złotych.

Wejście ▼

```
SELECT dost_id, COUNT(*) AS liczba_prod
FROM Produkty
GROUP BY dost_id
HAVING COUNT(*) >= 2;
```

Wyjście ▼

dost_id	liczba_prod
BRS01	3
DLL01	4
FNG01	2

Uwaga

Stosowanie HAVING i WHERE

Fraza HAVING jest tak podobna do frazy WHERE, że większość SZBD traktuje je identycznie, jeśli nie określono frazy GROUP BY. Mimo to warto samemu jasno rozdzielać obie frazy, czyli HAVING stosować tylko z GROUP BY, a WHERE używać do filtrowania na niższym poziomie.

Grupowanie i sortowanie

Trzeba zdać sobie sprawę z tego, iż frazy GROUP BY i ORDER BY są bardzo różne, choć wykonują podobne zadanie. Tabela 10.1 wymienia różnice występujące między obiema frazami.

Tabela 10.1. Frazy ORDER BY i GROUP BY

ORDER BY	GROUP BY
Sortuje wygenerowane wyjście.	Grupuje wiersze. Wyjście nie musi być jednak posortowane względem grup.
Można stosować dla dowolnych kolumn (także tych, które nie są zwracane).	Można zastosować tylko wobec zwracanych kolumn lub wyrażeń. Wystąpić muszą wszystkie zwracane kolumny lub wyrażenia.
Stosowanie frazy nie jest zawsze wymagane.	Stosowanie frazy jest konieczne, jeśli używa się kolumn (lub wyrażeń) z funkcjami agregującymi.

Pierwsza różnica wymieniona w tabeli 10.1 jest niezmiernie ważna. Choć nie jest to wymagane, to najczęściej grupy będą wyświetlane w sposób posortowany. Co więcej, choć dany system może zawsze sortować grupy po zastosowaniu frazy GROUP BY, konieczny może okazać się inny sposób sortowania. Choć grupowanie odbywa się w taki, a nie inny sposób, nie oznacza to jednocześnie, że sortowanie także musi odbywać się w ten sam sposób. Zawsze warto zastosować opcjonalną frazę ORDER BY, aby wymusić odpowiednie sortowanie pogrupowanych danych, nawet jeśli pojawią się dokładnie te same kolumny co we frazie GROUP BY.

Nie zapominaj o frazie ORDER BY

Ogólnie rzecz ujmując, za każdym razem, gdy stosuje się frazę GROUP BY, powinno się także stosować frazę ORDER BY, gdyż jest to jedyny sposób zapewnienia odpowiedniego sortowania danych. Nigdy nie należy polegać na sortowaniu za pomocą GROUP BY.

Wskazówka
Wskazówka

Oto krótki przykład, który zademonstruje użycie fraz GROUP BY i ORDER BY. Przedstawione zapytanie SELECT jest bardzo podobne do poprzednich. Pobiera numery zamówień i liczbę zamawianych elementów dla wszystkich zamówień zawierających minimum trzy elementy.

Wejście ▼

```
SELECT zam_numer, COUNT(*) AS elementy
FROM ElementyZamowienia
GROUP BY zam_numer
HAVING COUNT(*) >=3;
```

Wyjście ▼

zam_numer	elementy
-----	-----
20006	3
20007	5
20008	5
20009	3

Aby posortować wyjście na podstawie liczby zamówionych elementów, wystarczy tylko dodać odpowiednią frazę ORDER BY.

Wejście ▼

```
SELECT zam_numer, COUNT(*) AS elementy
FROM ElementyZamowienia
GROUP BY zam_numer
HAVING COUNT(*) >=3
ORDER BY elementy, zam_numer;
```

Niezgodność z bazą Access

Baza Danych Microsoft Access nie obsługuje sortowania na podstawie aliasów, więc przedstawiony przykład nie zadziała. Rozwiązaniem jest zastąpienie nazwy elementu (z frazy ORDER BY) rzeczywistym wyliczeniem lub pozycją pola, na przykład ORDER BY COUNT(*), zam_numer lub ORDER BY 2, zam_numer.

Uwaga
Uwaga

Wyjście ▼

zam_numer	elementy
20006	3
20009	3
20007	5
20008	5

Analiza ▼

W tym przykładzie fraza `GROUP BY` służy do grupowania danych na podstawie numeru zamówienia (kolumna `zam_numer`), więc funkcja `COUNT(*)` zwraca liczbę elementów w poszczególnych zamówieniach. Fraza `HAVING` filtruje dane, więc zwracane są tylko zamówienia z więcej niż dwoma elementami. Na końcu wyniki są sortowane za pomocą frazy `ORDER BY`.

Kolejność fraz instrukcji SELECT

Nadszedł chyba najlepszy czas na omówienie kolejności występowania fraz w instrukcji `SELECT`. Tabela 10.2 zawiera poprawną kolejność wszystkich omówionych do tej pory fraz.

Tabela 10.2. Frazy instrukcji `SELECT` i ich kolejność

Fraza	Opis	Wymagane
<code>SELECT</code>	zwracanie kolumn lub wyrażeń	tak
<code>FROM</code>	pobranie danych zawartych w tabelach	tylko gdy wymagane są dane z tabel
<code>WHERE</code>	filtrowanie wierszy	nie
<code>GROUP BY</code>	tworzenie grup	tylko do obliczania funkcji agregujących na grupach
<code>HAVING</code>	filtrowanie grup	nie
<code>ORDER BY</code>	sortowanie wyjścia	nie

Podsumowanie

W rozdziale 9. Czytelnik zapoznał się z zastosowaniem funkcji agregujących. W powyższym rozdziale wykorzystał frazę `GROUP BY` do przeprowadzania funkcji agregujących względem określonych grup elementów. Fraza `HAVING` służy do filtrowania grup. Dodatkowo w rozdziale pojawiło się dokładne wyjaśnienie różnic między `GROUP BY` i `ORDER BY` oraz między `WHERE` i `HAVING`.

Skorowidz

A

alias, 69, 70, 89, 97, 117
ANSI SQL, 21
Apache Open Office Base, 205
aplikacja internetowa, 182
ASP, 207
ASP.NET, 205, 208

B

baza danych, 15, 16, *Patrz też:* dane
aktualizacja, 141, 143
błąd, *Patrz:* błąd
filtrowanie, 41, 42
integralność referencyjna, 109, 174, 187,
188, 191
klient-serwer, 133, 141, 143
ograniczenie, 188, 189, 191, 196
sprawdzanie, 192
oprogramowanie, 16
relacyjna, 107, 187
schemat, 17
skalowalność, 108
sortowanie, 35, 37, 38
kierunek, 39, 40
SQLite, 213
system zarządzania, *Patrz:* SZBD
uprawnienia, 196
usuwanie obiektów, 191, 217
kaskadowe, 191
wprowadzanie danych, 133, 134, 136,
137
błąd, 174, 178

C

ColdFusion, 182, 205, 206
czas, 74, 77, 151, 224

D

dane, *Patrz też:* baza danych
baza, *Patrz:* baza danych
filtrowanie, 41, 42, 77, 93, 94, 96, 156
formatowanie, 66, 160
grupowanie, 91, 94, 96
nieposortowane, 24
sortowanie, 35, 37, 38, 77, 96, 156, 193
kierunek, 39, 40
logiczne, 193
typ, 18, 193, 221
binarny, 225
BIT, 223
CHAR, 222
CURRENCY, 224
DATE, 224
DATETIME, 224
DECIMAL, 223
FLOAT, 223
INT, 223
INTEGER, 223
konwersja, 74
LONG RAW, 225
MONEY, 224
NCHAR, 222
NUMBER, 223
NUMERIC, 223
numeryczny, 223
NVARCHAR, 222
RAW, 225
REAL, 223
SMALLDATETIME, 224
SMALLINT, 223
tekstowy, 221, 222
TEXT, 222
TIMESTAMP, 224
TINYINT, 223
walutowy, 224
zgodność, 18
typ, 225
usuwanie, 143, 144, 145

data, 74, 77, 151, 224
 aktualna, 151
 ODBC, 224
 systemowa, 151
 Database Management System, *Patrz:* SZBD
 DB2, 206
 DBMS, *Patrz:* SZBD
 słowo kluczowe, 29
 DROP, 158

F

fraza, 36, *Patrz też:* słowo kluczowe
 kolejność, 98
 funkcja, 73, 74, 187
 ABS, 80
 agregująca, 81, 82, 87, 123, 170
 dla unikalnych wartości, 87
 łączenie, 89
 AVG, 81, 82, 83, 88
 CAST, 74
 CONVERT, 74
 COS, 80
 COUNT, 81, 83, 123, 170
 CURDATE, 74
 DATALENGTH, 76
 DATE, 74
 DATE_PART, 78
 DATEPART, 78, 79
 EXP(), 80
 GETDATE, 74
 LCASE, 76
 LEFT, 76
 LEN, 76
 LENGTH, 76
 LOWER, 76
 LTRIM, 69, 76
 MAX, 81, 84, 88
 dla danych nienumerycznych, 85
 MID, 74
 MIN, 81, 85, 88
 dla danych nienumerycznych, 86
 NOW, 74
 numeryczna, 74, 80
 PI, 80
 RIGHT, 76
 RTRIM, 69, 75, 76
 SIN, 80
 SOUNDEX, 76
 SQRT, 80
 SUBSTR, 74
 SUBSTRING, 74

SUM, 81, 86
 systemowa, 74
 TAN, 80
 tekstowa, 74, 75
 TRIM, 69
 UCASE, 76
 UPPER, 75, 76
 YEAR, 79

I

IBM DB2, 206
 iloczyn kartezjański, 111
 indeks, 193, 194, 216
 efektywność, 195
 tworzenie, 194
 instrukcja
 ALTER TABLE, 152, 153, 189, 215
 COMMIT, 177, 178, 216
 CREATE, 175
 CREATE INDEX, 194, 216
 CREATE PROCEDURE, 216
 CREATE TABLE, 147, 148, 153, 189,
 215, 216
 CREATE VIEW, 158, 217
 DECLARE, 183
 DELETE, 144, 145, 195, 217
 uprawnienia, 143
 wycofanie, 175
 DROP, 175, 217
 DROP TABLE, 154
 fraza, *Patrz:* fraza, słowo kluczowe
 GRANT, 197
 INSERT, 133, 134, 137, 138, 195, 218
 bezpieczeństwo, 133, 136
 uprawnienia, 133
 wycofanie, 175
 INSERT SELECT, 138, 139, 218
 niejawne zatwierdzenie, 177
 OPEN CURSOR, 183
 REVOKE, 197
 ROLLBACK, 175, 176, 178, 218
 SELECT, 23, 24, 26, 27, 29, 72, 138, 145,
 156, 161, 175, 181, 218, *Patrz też:*
 zapytanie
 GROUP BY, 92, 94, 96, 97
 ORDER BY, 36, 42, 50, 96, 97, 131, 157
 WHERE, 41, 42, 49, 51, 52, 53, 55
 zagnieżdżanie, 99, 102, 103, 105
 SELECT INTO, 139, 140
 składnia, 215
 TRUNCATE TABLE, 144

UPDATE, 141, 145, 195, 219
 uprawnienia, 141
 wycofanie, 175
 interfejs użytkownika graficzny, 109

J

język

Java, 205
 JSP, 182, 205
 PHP, 182, 205, 211
 Python, 182, 205
 Ruby on Rails, 182, 205
 SQL, 20, 23
 biała spacja, 25
 implementacja, 31
 rozszerzenie, 21
 wielkość liter, 25

K

klucz

główny, 19, 20, 108, 150, 153, 173, 188,
 191
 sortowanie, 193
 tworzenie, 188, 189
 obcy, 20, 144, 153, 189, 190, 191
 tworzenie, 191

kod

kolorowanie, 102
 przenośny, 73

kolumna, *Patrz:* tabela kolumna

komentarz, 32, 33, 171
 jednowierszowy, 32
 wielowierszowy, 32

konkatenacja, 66, 67

krok, 181

kursor, 181, 182
 otwieranie, 183
 tworzenie, 183
 zamykanie, 186
 zastosowania, 182

M

MariaDB, 56, 206
 Microsoft Access, 207
 Microsoft ASP, 207
 Microsoft ASP.NET, 208
 Microsoft Query, 208
 Microsoft SQL Server, 209

MS-Query, 208
 MySQL, 209

O

obliczenia matematyczne, 71

ODBC, 213

operator, 42

!<, 43
 !=, 43, 45
 !>, 43
 ||, 67
 +, 67
 <, 43
 <=, 43
 <>, 43, 45
 =, 43
 >, 43
 >=, 43
 AND, 49, 50
 BETWEEN, 43, 45, 56
 EXISTS, 56
 IN, 53, 54, 56
 IS NULL, 43, 45, 46
 LIKE, 58
 logiczny, *Patrz:* operator
 NOT, 55, 56
 OR, 50
 UNION, 127, 128, 129
 UNION ALL, 130
 Oracle Express, 211
 Oracle SQL Developer, 210

P

perspektywa, 155, 156, 157, 197

filtrowanie niechcianych danych, 162

formatowanie danych, 160

nazwa, 157

odczytująca, 155

pole obliczane, 162

tworzenie, 158, 159, 160, 217

uprawnienia, 157

usuwanie, 158

wielokrotnego użytku, 159

zagnieżdżanie, 157

złożona, 157

PL-SQL, 21

pole obliczane, 65, 66, 103, 162

obliczenia matematyczne, 71

polecenie, *Patrz:* instrukcja

PostgreSQL, 212
 predykat, 58
 primary key, *Patrz:* klucz główny
 procedura
 tworzenie, 216
 zapamiętana, 153, 165, 166, 195, 216
 tworzenie, 169
 wykonywanie, 167

Q

Query, 208

R

rekord, 19

S

serwer uprawnień, 197
 skorowidz, 193
 słowo kluczowe, 23, 227
 ALL, 87, 89, 93
 AND, 49, 50
 AS, 70, 118
 ASC, 40
 BETWEEN, 43, 56
 Commit, 175
 CREATE TABLE, 148
 DEFAULT, 151
 DESC, 39
 DISTINCT, 28, 88, 89
 EXCEPT, 132
 EXISTS, 56
 FROM, 98, 142, 143, 144
 GROUP BY, 92, 96, 97, 98
 HAVING, 94, 96, 98
 IN, 53, 54, 56
 INNER JOIN, 113
 INSERT, 134
 INTERSECT, 132
 LIKE, 58
 LIMIT, 30
 NOT, 55, 56
 NULL, 43, 46, 60, 83, 84, 137, 149, 150
 OR, 50
 ORDER BY, 96, 97, 98, 131, 157
 OUTER JOIN, 122
 Rollback, 175
 Savepoint, 175
 SELECT, 98

TOP, 29, 89
 TOP PERCENT, 89
 Transaction, 175
 UNION, 127, 128, 129
 UNION ALL, 130
 VALUES, 136
 WHERE, 94, 96, 98, 111, 113, 145

SQL Server, 209
 SQLite, 213
 Structured Query Language, *Patrz:* język SQL
 system zarządzania bazą danych, *Patrz:* SZBD
 SZBD, 12, 16, 205

T

tabela, 16, 199
 aktualizacja, 141, 143, 144, 215
 definicji, 152
 kolumna, 17, 18, 19, 20, 25, 27, 37, 66, 108
 alias, 69
 nazwa w pełni kwalifikowana, 105
 położenie względne, 93
 wartość domyślna, 151, 152
 kopiowanie, 140
 łączenie, 107, 114, 115
 nazwa, 17
 alias, 117
 zmiana, 154
 relacyjna, 99
 tworzenie, 147, 148, 216
 usuwanie, 144, 154
 wiersz, 19, 181
 dodawanie, 218
 usuwanie, 217
 wirtualna, *Patrz:* perspektywa
 Transact-SQL, 21
 transakcja, 173, 174, 216
 punkt kontrolny, 178, 179
 sterowanie, 175
 wycofanie, 178
 zatwierdzenie, 178

U

unia, 127, 128, 130
 EXCEPT, 132
 INTERSECT, 132
 użytkownika autoryzacja, 197

W

widok, *Patrz:* perspektywa
wyzwalacz, 195, 196
wzorzec wyszukiwania, 57, 58

Z

zapytanie, *Patrz też:* instrukcja SELECT
 łączenie, 127
 zagnieżdżone, 99, 102, 103, 105, 120,
 142
 złożone, *Patrz:* unia
zbiór wyników, 181
złączenie, 109, 111, 123, 144, 158
 INNER JOIN, 113
 naturalne, 120
 nazwa kolumn, 110
 rozbudowane, 117, 118
 tworzenie, 109
 warunek, 111
 wewnętrzne, 113, 125
 własne, 118, 120
 zewewnętrzne, 121, 125
 lewe, 122
 pełne, 123
 prawe, 122
zmienna, 170
znak
 !, 63
 !<, 43
 !=, 43, 45
 !>, 43
 %, 58
 *, 27, 57, 58, 63
 */, 33
 /*, 33
 ?, 60
 @, 170
 [], 62
 ^, 62, 63
 _ 60
 ||, 67
 +, 67
 <, 43
 <=, 43
 <>, 43, 45
 =, 43
 >, 43
 >=, 43
apostrof, 44, 222
biała spacja, 25
gwiazdka, 27, 58
łącznik, 32
podkreślenia, 60
procent, 58
przecinek, 26
średnik, 25
wieloznaczności, 27, 57, 58, 63
zapytania, 60

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>

SQL to język, który pozwala manipulować danymi znajdującymi się w bazie danych. Jego znajomość pozwoli Ci błyskawicznie pobrać z bazy interesujące Cię informacje. To pierwszy krok, by poddać je zaawansowanej obróbce lub wyciągnąć ciekawe wnioski. SQL został opracowany w latach 70. i wciąż jest powszechnie używany. Nic nie wskazuje na to, żeby ten stan rzeczy uległ zmianie. Dlatego zainwestowanie czasu w jego naukę jest doskonałym ruchem!

Jeżeli chcesz błyskawicznie opanować możliwości SQL-a, sięgnij po tę niezwykłą książkę. Jeśli poświęcisz tylko 10 minut dziennie*, już za kilka tygodni zaczniesz swobodnie tworzyć nawet bardzo zaawansowane zapytania. Złączenia, indeksy, sortowanie wyników, funkcje agregujące to tylko niektóre z poruszanych tu tematów. W trakcie lektury kolejnych rozdziałów nauczysz się tworzyć zaawansowane warunki wyszukiwania danych, skorzystasz z podzapytań oraz przekonasz się, w czym może pomóc stosowanie widoków. Ponadto zobaczysz, w jaki sposób modyfikować dane w bazie oraz dlaczego powinieneś być szczególnie ostrożny, jeżeli chcesz coś z bazy usunąć. Najwyższa pora opanować język SQL!

Dzięki tej książce:

- nauczysz się wyciągać z bazy interesujące dane
- zaktualizujesz zawarte w niej informacje
- skorzystasz z zaawansowanych złączeń
- opanujesz język SQL

Zainwestuj 10 minut dziennie w naukę SQL!

* Możesz poświęcić więcej niż 10 minut dziennie i opanować SQL w jeszcze krótszym czasie!

Ben Forta — ma ponad 20-letnie doświadczenie w tworzeniu oprogramowania, udzielaniu wsparcia technicznego, prowadzeniu szkoleń oraz marketingu. Jest specjalistą ColdFusion oraz autorem licznych książek poświęconych bazom danych i wyrażeniom regularnym. Często występuje jako prelegent podczas wydarzeń związanych z branżą IT w różnych zakątkach świata.

34121	numer katalogowy
księgarnia internetowa	
	http://helion.pl
zamówienia telefoniczne	
	0 801 339900
	0 601 339900
Informatyka w najlepszym wydaniu	
Sprawdź najnowsze promocje: 📍 http://helion.pl/promocje Książki najchętniej czytane: 📍 http://helion.pl/bestsellery Zamów informacje o nowościach: 📍 http://helion.pl/nowosci	
KOD KORZYŚCI	
ISBN 978-83-283-1089-6	
9 788328 310896	
cena: 39,90 zł	