

Marcin Lis

ĆWICZENIA



PRAKTYCZNE

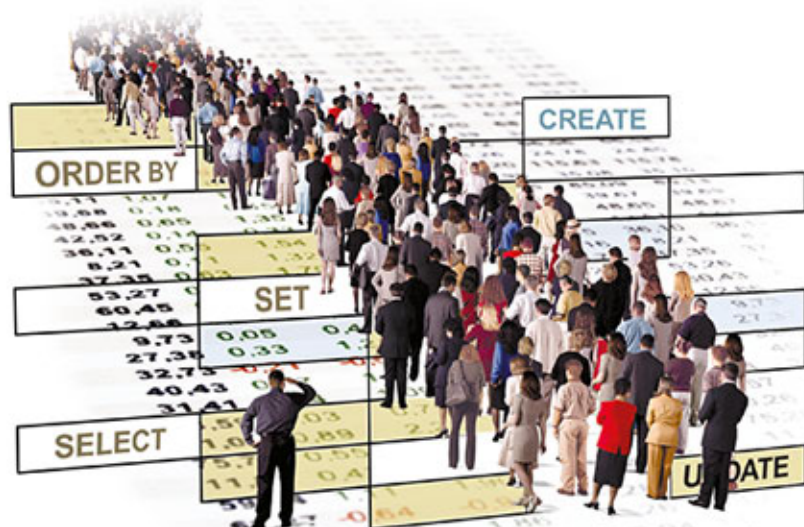
SQL

Naucz się sprawnie używać bazy danych!

Podstawy relacyjnych baz danych, czyli po co tworzy się bazy i czym jest język SQL

Dane w bazie, czyli jak przechowywać informacje, modyfikować je lub usuwać

Przetwarzanie danych, czyli jak szybko wyciągnąć z bazy wszystkie potrzebne dane



Wydanie III

Helion



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek 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.

Redaktor prowadzący: Ewelina Burska

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

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/cwsql3>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-9417-4

Copyright © Helion 2014

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Wstęp	5
Rozdział 1. Podstawy relacyjnych baz danych	9
Tabele	9
Klucze	10
Relacje	11
Podstawowe zasady projektowania tabel	16
Rozdział 2. Praca z tabelami	25
Typy danych	25
Tworzenie i wybór bazy danych	29
Tworzenie tabel	31
Atrybuty kolumn	33
Indeksy	38
Modyfikacja tabel	41
Usuwanie tabel	48
Rozdział 3. Umieszczanie danych w bazie	49
Instrukcja INSERT INTO	49
Wprowadzanie wielu wierszy	55
Druga postać instrukcji INSERT	57
Rozdział 4. Pobieranie danych z tabel	59
Podstawy instrukcji SELECT	59
Sortowanie wyników zapytań	63
Kryteria pobierania danych	65
Niepowtarzalność wierszy	74
Ograniczanie wyników zapytań	75

Rozdział 5. Modyfikacja i usuwanie danych	77
Instrukcja UPDATE	77
Modyfikacja danych w tabelach	78
Usuwanie danych	82
Rozdział 6. Złączenia	85
Łączenie wyników zapytań	85
Pobieranie danych z wielu tabel	90
Złączenia	94
Rodzaje warunków złączenia	104
Rozdział 7. Funkcje agregujące	109
Rozdział 8. Grupowanie danych	119
Rozdział 9. Podzapytania	129
Podzapytania w klauzuli FROM	130
Podzapytania proste	132
Podzapytania skorelowane	136
Podzapytania w instrukcjach aktualizujących dane	141
Rozdział 10. Perspektywy (widoki)	145
Rozdział 11. Transakcje	151
Transakcje w systemach baz danych	151
Objęcie instrukcji transakcją	152
Wycofywanie transakcji	153
Izolacja transakcji	154
Rozdział 12. Więzy integralności	157
Integralność danych	157
Definiowanie klucza obcego	158
Dodawanie i usuwanie więzów	161
Dodatek A Instalacja i podstawowa konfiguracja baz	163
Baza MySQL	163
Baza PostgreSQL	177
Baza SQLite	184
Baza Microsoft SQL Server (MS SQL)	185
Baza Oracle	191



Modyfikacja i usuwanie danych

Instrukcja UPDATE

Do modyfikacji danych zawartych w tabelach służy instrukcja UPDATE. Ma ona ogólną postać:

```
UPDATE nazwa_tabeli  
SET kolumna1=wartość1, kolumna2=wartość2, ..., kolumnaN=wartośćN  
[WHERE warunek]
```

co oznacza: zmień w tabeli *nazwa_tabeli*, w kolumnach spełniających warunek *warunek*, wartość kolumny *kolumna1* na *wartość1*, kolumny *kolumna2* na *wartość2* itd. Warunek występujący po klauzuli WHERE jest przy tym opcjonalny, a jego pominięcie oznacza, że zmiany nastąpią we wszystkich wierszach. W dalszych ćwiczeniach będziemy korzystać z tabeli *pracownicy*, utworzonej w rozdziale 4.

Ć W I C Z E N I E

5.1 Zmiana wszystkich wartości we wskazanej kolumnie

Zmień zawartość kolumny *pesel* w tabeli *pracownicy*, tak aby wszystkie wiersze zawierały ciąg 01234567890.



Wykonanie ćwiczenia zmodyfikuje zawartość WSZYSTKICH wierszy w tabeli pracownicy. Nie pojawi się przy tym żadne ostrzeżenie czy też pytanie o potwierdzenie chęci wykonania instrukcji (chyba że korzystamy z programu klienta oferującego dodatkowe zabezpieczenia). Przywrócenie oryginalnej zawartości tabeli będzie wymagało ponownego wprowadzenia danych.

Zmianę wszystkich wartości w kolumnie pesel zapewni nam instrukcja:

```
UPDATE pracownicy SET pesel='01234567890';
```

Jeśli teraz wykonamy instrukcję SELECT pobierającą wszystkie dane z tabeli pracownicy, zobaczymy, że istotnie wszystkie wiersze kolumny pesel zostały zmienione, tak jak jest to widoczne na rysunku 5.1.

id	imie	nazwisko	placa	stanowisko	pesel
1	Adam	Kowalski	1624.50	magazynier	01234567890
2	Adam	Nowak	3760.00	kierownik	01234567890
3	Andrzej	Kowalski	4200.00	kierownik	01234567890
4	Arkadiusz	Malinowski	1600.00	kierowca	01234567890
5	Andrzej	Malinowski	1450.00	sprzedawca	01234567890
6	Krzysztof	Nowicki	1300.00	sprzedawca	01234567890
7	Kacper	Adamczyk	1610.50	serwisant	01234567890
8	Kamil	Andrzejczak	1200.00	asystent	01234567890
9	Krzysztof	Arkuszewski	1500.00	magazynier	01234567890
10	Kamil	Borowski	1600.00	sprzedawca	01234567890

10 rows in set (0.02 sec)

Rysunek 5.1. Wszystkie dane w kolumnie pesel zostały zmodyfikowane

Modyfikacja danych w tabelach

Ćwiczenie 5.1 pokazało, w jaki sposób zmienić dane we wszystkich wierszach wybranej kolumny. To jednak rzadko spotykany przypadek; najczęściej modyfikacji podlegają tylko rekordy spełniające zadane kryteria. Przykładowo, mogłoby się okazać, że Kacper Adamczyk ma przypisany błędny numer PESEL i należy go zmienić z 92341678903 na 12341678993.

Ć W I C Z E N I E

5.2 Modyfikacja kolumny w wybranym wierszu tabeli

Zmień numer PESEL przypisany Kacprowi Adamczykowi w tabeli `pracownicy`.

Aby wykonać ćwiczenie, należy zastosować instrukcję:

```
UPDATE pracownicy SET pesel='12341678993' WHERE id=7;
```

Warunek `id=7` został zastosowany, gdyż pole `id` jest kluczem głównym jednoznacznie identyfikującym każdy rekord. Co prawda można by wykonać również instrukcję:

```
UPDATE pracownicy SET pesel='12341678993' WHERE pesel='92341678903';
```

ale nie jest to sposób zalecany. W pierwszym bowiem przypadku zawsze mamy pewność, który rekord zostanie zmodyfikowany, w drugim — niestety, nie. Nie możemy mieć bowiem gwarancji, że w bazie nie znajduje się już PESEL 92341678903, gdyż kolumna `pesel` nie gwarantuje unikatowości każdego wpisu (brak atrybutu `UNIQUE`) i nie powinna być stosowana jako wyróżnik modyfikowanego wiersza (jeden PESEL mógł być np. błędnie przypisany kilku osobom).

Nic nie stoi również na przeszkodzie, aby jednocześnie zmodyfikować kilka pól w danym wierszu. Moglibyśmy np. zmienić od razu nazwisko, stanowisko i płacę danej osoby.

Ć W I C Z E N I E

5.3 Modyfikacja kilku kolumn w jednym wierszu

Zmodyfikuj dane wybranej osoby tak, aby jednocześnie zostały zmienione: nazwisko, stanowisko oraz płaca.

Aby wykonać to zadanie, możemy użyć instrukcji:

```
UPDATE pracownicy SET nazwisko='Andrzejewski', placa=3440.00,  
↳stanowisko='kierownik' WHERE id=8;
```

Tym samym Kamil Andrzejczak, pracujący na stanowisku asystenta, z płacą 1200 zł, stanie się Kamilem Andrzejewskim, pracującym na stanowisku kierowniczym, z płacą 3440 zł. O tym, że taka zmiana

rzeczywiście nastąpiła, możemy się przekonać, wykonując instrukcję SELECT w postaci:

```
SELECT * FROM pracownicy WHERE id=8;
```

co zostało również zobrazowane na rysunku 5.2.

```
mysql> SELECT * FROM pracownicy WHERE id=8;
+----+-----+-----+-----+-----+-----+
| id | imie  | nazwisko | placa | stanowisko | pesel |
+----+-----+-----+-----+-----+-----+
| 8  | Kamil | Andrzejczak | 1200.00 | asystent    | NULL  |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> UPDATE pracownicy SET nazwisko='Andrzejewski', placa=3440.00, stanowisko=
'kierownik' WHERE id=8;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM pracownicy WHERE id=8;
+----+-----+-----+-----+-----+-----+
| id | imie  | nazwisko | placa | stanowisko | pesel |
+----+-----+-----+-----+-----+-----+
| 8  | Kamil | Andrzejewski | 3440.00 | kierownik   | NULL  |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Rysunek 5.2. Zmiana kilku danych w wybranym wierszu

Możliwości instrukcji UPDATE nie ograniczają się do modyfikacji danych w jednym wierszu. To, które rekordy zostaną zmodyfikowane, zależy tylko od warunku klauzuli WHERE. Możemy więc np. zmienić nazwę stanowiska „sprzedawca” na „doradca klienta” we wszystkich wierszach kolumny stanowisko.

Ć W I C Z E N I E

5.4 Modyfikacja kilku rekordów

W kolumnie stanowisko zmień wpisy o treści sprzedawca na doradca klienta.

Aby wykonać tak przedstawione zadanie, należy posłużyć się instrukcją:

```
UPDATE pracownicy SET stanowisko='doradca klienta' WHERE
↪stanowisko='sprzedawca';
```

Pobranie danych z tabeli pokaże, że zmiana rzeczywiście została dokonana (rysunek 5.3).

id	imie	nazwisko	placa	stanowisko	pesel
1	Adam	Kowalski	1624.50	magazynier	12345678901
2	Adam	Nowak	3760.00	kierownik	92345678901
3	Andrzej	Kowalski	4200.00	kierownik	72345678901
4	Arkadiusz	Malinowski	1600.00	kierowca	92345678909
5	Andrzej	Malinowski	1450.00	doradca klienta	NULL
6	Krzysztof	Nowicki	1300.00	doradca klienta	NULL
7	Kacper	Adamczyk	1610.50	serwisant	92341678903
8	Kamil	Andrzejewski	3440.00	kierownik	NULL
9	Krzysztof	Arkuszewski	1500.00	magazynier	02343678913
10	Kamil	Borowski	1600.00	doradca klienta	32349678913

10 rows in set (0.00 sec)

Rysunek 5.3. Nazwa stanowiska „sprzedawca” została zmieniona na „doradca klienta”

Naraz można także modyfikować kilka kolumn w wielu wierszach. Jeśli więc firma przechowująca dane w tabeli pracownicy przejdzie kolejną reorganizację, a doradcy klientów ponownie staną się sprzedawcami i jednocześnie ich płace zostaną zrównane do 1400 zł, wszelkie niezbędne zmiany wprowadzimy również dzięki jednemu tylko zapytaniu.

Ć W I C Z E N I E

5.5 Modyfikacja kilku kolumn w wielu wierszach

Użyj pojedynczego zapytania do zmiany nazwy stanowiska doradca klienta na sprzedawca oraz do zmiany płacy osób na tym stanowisku na 1400 zł.

Niezbędne zapytanie ma postać:

```
UPDATE pracownicy SET stanowisko='sprzedawca', placa=1400.00 WHERE
↳ stanowisko='doradca klienta';
```

Ć W I C Z E N I E

5.6 Modyfikacja kolumn z wartością NULL

Zmodyfikuj zawartość kolumny pesel w taki sposób, aby wiersze mające w niej wartość NULL miały przypisany ciąg znaków 'nieznany'.

```
UPDATE pracownicy SET pesel='nieznany' WHERE pesel IS NULL;
```

Usuwanie danych

Do usuwania danych służy instrukcja DELETE o ogólnej postaci:

```
DELETE FROM tabela  
[WHERE warunek]
```

Oznacza ona: usuń z tabeli *tabela* wszystkie wiersze spełniające warunek *warunek*. Jeśli warunek zostanie pominięty, zostaną usunięte wszystkie dane (podobnie jak w przypadku instrukcji UPDATE, gdzie pominięcie warunku powodowało modyfikację wszystkich wierszy tabeli).

Ć W I C Z E N I E

5.7 Usunięcie wszystkich danych z tabeli

Usuń wszystkie dane z tabeli *pracownicy*.

Aby usunąć wszystkie dane z tabeli *pracownicy*, należy wykonać instrukcję:

```
DELETE FROM pracownicy;
```

Po jej wykonaniu tabela *pracownicy* nie będzie zawierała żadnych danych. Taką konstrukcję należy więc stosować z rozważą, gdyż serwer nie wygeneruje żadnego ostrzeżenia czy dodatkowego pytania (niektóre programy klienckie zawierają jednak zabezpieczenia przed wydawaniem zapytań tego typu). Wpisanie powyższej konstrukcji i zatwierdzenie jej klawiszem *Enter* spowoduje natychmiastowe skasowanie danych.

Selektywne usuwanie danych jest możliwe dzięki zastosowaniu klauzuli *WHERE* z odpowiednim wyrażeniem warunkowym, które konstruuje się na takich samych zasadach jak w przypadku instrukcji *SELECT* czy *UPDATE*. Zostanie to pokazane w kilku kolejnych ćwiczeniach.

Ć W I C Z E N I E

5.8 Usunięcie wybranego wiersza

Usuń z tabeli *pracownicy* dowolnie wybrany wiersz.

Aby usunąć z tabeli pracownicy np. dane osoby o identyfikatorze 5, zastosujemy instrukcję:

```
DELETE FROM pracownicy WHERE id=5;
```

Ć W I C Z E N I E

5.9 Jednoczesne usunięcie kilku wierszy

Usuń z tabeli pracownicy wiersze o identyfikatorach (wartości kolumny id): 1, 3, 7.

Aby usunąć dane osób o identyfikatorach 1, 3 i 7, najprościej wykonać instrukcję:

```
DELETE FROM pracownicy WHERE id IN (1, 3, 7);
```

Można również zastosować serię warunków połączonych operatorem OR:

```
DELETE FROM pracownicy WHERE id=1 OR id=3 OR id=7;
```

Ć W I C Z E N I E

5.10 Instrukcja DELETE i operator BETWEEN

Usuń z tabeli pracownicy wiersze o identyfikatorach z przedziału 4 – 8.

Usunięcie z tabeli pracownicy wierszy, które mają w kolumnie id wartości z przedziału 4 – 8, uzyskamy, wykonując instrukcję:

```
DELETE FROM pracownicy WHERE id BETWEEN 4 AND 8;
```

Ć W I C Z E N I E

5.11 Usunięcie rekordów ze względu na ciąg znaków

Usuń z tabeli pracownicy dane wszystkich osób o nazwisku Kowalski.

Aby usunąć z bazy dane pracowników o nazwisku Kowalski, zastosujemy instrukcję:

```
DELETE FROM pracownicy WHERE nazwisko='Kowalski';
```


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>

GRUPA WYDAWNICZA

 **Helion SA**

SQL. ĆWICZENIA PRAKTYCZNE

Wydanie III

Sprawdź możliwości
Twojej bazy danych!



Relacyjnych baz danych używa się w większości instytucji. Nie sposób dziś prowadzić jakiejkolwiek działalności komercyjnej bez bazy klientów, dostawców czy wykazu podatków. Często jednak efektywna obsługa takiej bazy wykracza poza kwalifikacje przeciętnego pracownika, który wykorzystuje zaledwie ułamek jej mocy. Ta książka może to zmienić! Znajdziesz w niej informacje, jak grupować określone dane, co ułatwi Ci wykonanie rzetelnej analizy działalności Twojej firmy. Dzięki temu przewodnikowi wykorzystasz w swojej pracy potencjał bazy danych i nauczysz się z nią współpracować.

Książka ta, wydana w cenionej serii „Ćwiczenia Praktyczne”, pozwoli Ci szybko zapoznać się zarówno z podstawami działania relacyjnych baz danych, jak i z bardziej zaawansowanymi zagadnieniami, takimi jak funkcje agregujące czy transakcje. Znajdziesz tu wszystkie najważniejsze kwestie związane z językiem zapytań SQL i operowaniem danymi. Odkryjesz, jak grupować dane, do czego służą podzapytania i co to są więzy integralności. Podręcznik ten, przygotowany w formie bogatego zestawu praktycznych przykładów i konkretnych ćwiczeń, pomoże Ci opanować umiejętność poruszania się po bazie danych.

- Praca z tabelami i umieszczanie danych w bazie
- Pobieranie danych, ich modyfikacja i usuwanie
- Złączenia i funkcje agregujące
- Uzyskiwanie informacji statystycznych
- Grupowanie danych i podzapytania
- Perspektywy (widoki), transakcje i więzy integralności
- Optymalizacja podzapytań
- Instalacja i podstawowa konfiguracja baz

helion.pl
księgarnia
internetowa



Helion

Nr katalogowy: **21033**



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



Zamówienia telefoniczne:
0 801 339900



0 601 339900

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>

sięgnij po **WIECEJ**



KOD KORZYŚCI

ISBN 978-83-246-9417-4



Cena 32,90 zł

Informatyka w najlepszym wydaniu

9 788324 694174