

Marcin Lis

ĆWICZENIA



PRAKTYCZNE

# SQL

Wydanie II

Błyskawicznie opanuj sztukę  
postępowania się bazami danych opartymi na SQL!

**Podstawy relacyjnych baz danych** – czyli co warto wiedzieć na początek

**Praca z danymi** – czyli umieszczenie ich w bazie, modyfikacja i usuwanie

**Ułatwianie sobie życia** – czyli funkcje agregujące, podzapytania i transakcje



## » Idź do

- Spis treści
- Przykładowy rozdział

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 32 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991–2011

## SQL. Ćwiczenia praktyczne. Wydanie II

Autor: [Marcin Lis](#)  
ISBN: 978-83-246-3051-6  
Format: 140×208, stron: 176



### **Błyskawicznie opanuj sztukę posługiwania się bazami danych opartymi na SQL!**

- Podstawy relacyjnych baz danych – czyli co warto wiedzieć na początek
- Praca z danymi – czyli umieszczanie ich w bazie, modyfikacja i usuwanie
- Ułatwianie sobie życia – czyli funkcje agregujące, podzapytania i transakcje

Relacyjne bazy danych, oparte na języku SQL, to dziś niezwykle popularne i najczęściej używane struktury do przechowywania dużej ilości danych. Nie tylko ułatwiają segregowanie i szybkie uzyskiwanie potrzebnych informacji – umożliwiają też przeprowadzanie na danych rozmaitych złożonych operacji, oszczędzających użytkownikom ogromne ilości czasu. Pozwalają bez trudu dodawać nowe dane, zmieniać i usuwać te znajdujące się już w bazie oraz wyłuskiwać wszelkie wiadomości pod kątem specyficznych, szczegółowych kryteriów wyszukiwania. Wystarczy tylko poznać podstawowe zasady działania języka SQL.

„SQL. Ćwiczenia praktyczne. Wydanie II” pomoże Ci opanować podstawy tego języka, a także wskaże, jak posługiwać się nim w konkretnych przypadkach. Znajdziesz tu wiedzę potrzebną do utworzenia bazy danych w jednym z kilku najpopularniejszych systemów bazodanowych, zaprojektowania tabel optymalnych dla treści, które chcesz w nich przechowywać, wprowadzania danych do bazy i modyfikowania ich. Zobaczysz także, co zrobić, by łatwo, szybko i precyzyjnie wyszukać interesujące Cię informacje, pobrać je i posortować. Dowiesz się, co to są funkcje agregujące, więzy integralności i podzapytania oraz jak korzystać z transakcji. Krótko mówiąc, znajdziesz tu wszystko, co musisz wiedzieć, by sprawnie i wygodnie korzystać ze swojej bazy danych.

- Podstawy relacyjnych baz danych
- Praca z tabelami
- Umieszczanie danych w bazie
- Pobieranie danych z tabel
- Złożone kryteria wyszukiwania
- Modyfikacja i usuwanie danych
- Złączenia
- Funkcje agregujące
- Podzapytania
- Więzy integralności
- Transakcje
- Instalacja i podstawowa konfiguracja baz

**Stwórz własną bazę danych i korzystaj z jej możliwości!**

# Spis treści

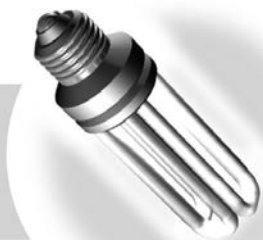
Wstęp	9
<b>Rozdział 1. Podstawy relacyjnych baz danych</b>	<b>13</b>
Tabele	13
Klucze	14
Relacje	15
Podstawowe zasady projektowania tabel	20
Określenie celu	20
Duplikowanie danych	21
Informacje atomowe	22
Puste pola	24
Jednoznaczna identyfikacja rekordów	26
<b>Rozdział 2. Praca z tabelami</b>	<b>27</b>
Typy danych	27
Typy liczbowe	28
Typy daty i czasu	29
Typy łańcuchowe	29
Typ null	30
Tworzenie i wybór bazy danych	31
Tworzenie tabel	32
Atrybuty kolumn	34
Indeksy	39
Modyfikacja tabel	42
Usuwanie tabel	49

---

<b>Rozdział 3. Umieszczanie danych w bazie</b>	<b>51</b>
Instrukcja INSERT INTO	51
Wprowadzanie wielu wierszy	57
Druga postać instrukcji INSERT	59
<b>Rozdział 4. Pobieranie danych z tabel</b>	<b>61</b>
Podstawy instrukcji SELECT	61
Sortowanie wyników zapytań	65
Kryteria pobierania danych	67
Niepowtarzalność wierszy	75
<b>Rozdział 5. Modyfikacja i usuwanie danych</b>	<b>77</b>
Instrukcja UPDATE	77
Modyfikacja danych w tabelach	78
Usuwanie danych	81
<b>Rozdział 6. Złączenia</b>	<b>85</b>
Łączenie wyników zapytań	85
Instrukcja UNION	86
Instrukcja INTERSECT	87
Instrukcja EXCEPT	88
Pobieranie danych z wielu tabel	89
Złączenia	94
Złączenie typu CROSS JOIN	94
Złączenie typu INNER JOIN	95
Złączenie typu LEFT OUTER JOIN	96
Złączenie typu RIGHT OUTER JOIN	97
Złączenie typu FULL OUTER JOIN	99
Złączenia i klauzula WHERE	100
Aliasy tabel i rozróżnianie nazw kolumn	100
<b>Rozdział 7. Funkcje agregujące</b>	<b>105</b>
<b>Rozdział 8. Grupowanie danych</b>	<b>115</b>
<b>Rozdział 9. Podzapytania</b>	<b>123</b>
Podzapytania w klauzuli FROM	124
Podzapytania klauzuli WHERE	126
Podzapytania proste	126
Podzapytania skorelowane	128
Podzapytania w instrukcjach aktualizujących dane	130

---

<b>Rozdział 10. Transakcje</b>	<b>133</b>
Transakcje w systemach baz danych	133
Objęcie instrukcji transakcją	134
Wycofywanie transakcji	135
Izolacja transakcji	136
<b>Rozdział 11. Więzy integralności</b>	<b>139</b>
Integralność danych	139
Definiowanie klucza obcego	140
Dodawanie i usuwanie więzów	143
<b>Dodatek A Instalacja i podstawowa konfiguracja baz</b>	<b>145</b>
Baza MySQL	145
Instalacja w systemie Linux	145
Instalacja w systemie Windows	150
Baza PostgreSQL	154
Instalacja w systemie Linux	154
Instalacja w systemie Windows	158
Baza SQLite	161
Praca z bazą	162
Baza Microsoft SQL Server (MS SQL)	162
Instalacja	163
Tworzenie bazy testowej	165
Praca z bazą	166
Baza Oracle	167
Instalacja w systemie Linux	167
Instalacja w systemie Windows	171



# 5

## 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 będą dokonane we wszystkich wierszach. W dalszych ćwiczeniach będziemy korzystać z tabeli pracownicy powstałej 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. 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 faktycznie 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 polecany. 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 ta kolumna nie gwarantuje unikalności każdego wpisu 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 na przykład zmienić od razu imię, 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 wykonać instrukcję:

```
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 faktycznie 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 tylko 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 faktycznie została dokonana (rysunek 5.3).

Naraz można także modyfikować wiele kolumn w wielu wierszach. Jeśli więc firma przechowująca dane w tabeli pracownicy przejdzie kolejną reorganizację i doradcy klientów ponownie staną się sprzedawcami, i jednocześnie ich płace zostaną zrównane do 1400 zł, wszelkich niezbędnych zmian dokonamy również dzięki jednemu tylko zapytaniu.

id	imie	nazwisko	płaca	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”*

## 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 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 konstrukcją należy więc stosować z rozwagą, gdyż serwer nie wygeneruje żadnego ostrzeżenia czy dodatkowego pytania. Wpisanie powyższej konstrukcji i zatwierdzenie jej klawiszem *Enter* spowoduje natychmiastowe skasowanie danych!

Selektywne usuwanie danych zapewnia użycie 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* 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 Usuwanie 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';
```

# SQL. ĆWICZENIA PRAKTYCZNE

Stwórz własną bazę danych i korzystaj z jej możliwości!



Relacyjne bazy danych, oparte na języku SQL, to dziś niezwykle popularne i najczęściej używane struktury do przechowywania dużej ilości danych. Nie tylko ułatwiają segregowanie i szybkie uzyskiwanie potrzebnych informacji – umożliwiają też przeprowadzanie na danych rozmaitych złożonych operacji, oszczędzających użytkownikom ogromne ilości czasu. Pozwalają bez trudu dodawać nowe dane, zmieniać i usuwać te znajdujące się już w bazie oraz wyłuskiwać wszelkie wiadomości pod kątem specyficznych, szczegółowych kryteriów wyszukiwania. Wystarczy tylko poznać podstawowe zasady działania języka SQL.

„SQL. Ćwiczenia praktyczne. Wydanie III” pomoże Ci opanować podstawy tego języka, a także wskaże, jak posługiwać się nim w konkretnych przypadkach. Znajdziesz tu wiedzę potrzebną do utworzenia bazy danych w jednym z kilku najpopularniejszych systemów bazodanowych, zaprojektowania tabel optymalnych dla treści, które chcesz w nich przechowywać, wprowadzania danych do bazy i modyfikowania ich. Zobaczysz także, co zrobić, by łatwo, szybko i precyzyjnie wyszukać interesujące Cię informacje, pobrać je i posortować. Dowiesz się, co to są funkcje agregujące, więzy integralności i podzapytania oraz jak korzystać z transakcji. Krótko mówiąc, znajdziesz tu wszystko, co musisz wiedzieć, by sprawnie i wygodnie korzystać ze swojej bazy danych.

- Podstawy relacyjnych baz danych
- Praca z tabelami
- Umieszczanie danych w bazie
- Pobieranie danych z tabel
- Złożone kryteria wyszukiwania
- Modyfikacja i usuwanie danych
- Złączenia
- Funkcje agregujące
- Podzapytania
- Więzy integralności
- Transakcje
- Instalacja i podstawowa konfiguracja baz

Nr katalogowy: 6134



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>

**helion.pl**  
księgarnia  
internetowa

**Cena 29,00 zł**

ISBN 978-83-246-3051-6



Informatyka w najlepszym wydaniu