

Przewodnik po świecie SQL!

Wydanie II

SQL

Leksykon kieszonkowy



O'REILLY®

Jonathan Gennick

HELION



» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

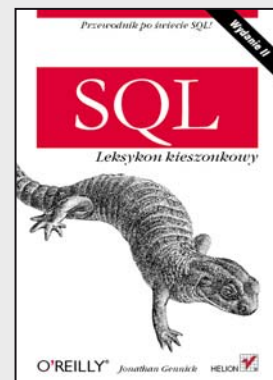
- Fragmenty książek online

» Kontakt

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

SQL. Leksykon kieszonkowy. Wydanie II

Autor: Jonathan Gennick
Tłumaczenie: Marek Pałczyński
ISBN: 978-83-246-3306-7
Tytuł oryginału: [SQL Pocket Guide](#)
Format: 115×170, stron: 220



Przewodnik po świecie SQL!

- Jak wykorzystać podzapytania?
- Jak wykonać złączenie tabel?
- Jak stworzyć tabelę w DB2, MySQL, Oracle, PostgreSQL oraz SQL Server?

SQL to prawdopodobnie jeden z ostatnich wiekowych języków programowania, tak powszechnie używanych w codziennej pracy. Jego możliwości są ogromne. Jeżeli nauczysz się konstruować zapytania SQL, będziesz potrafił wyłowić dowolną informację z każdej bazy danych. Brzmi kusząco?

Dzięki temu przewodnikowi będziesz miał zawsze na wyciągnięcie ręki kompletny zbiór informacji na temat języka SQL. Już nigdy więcej nie zaskoczą Cię niuanse jego składni. Błyskawicznie zapoznasz się z typami danych, sposobami wykonywania operacji na danych oraz dostępnymi funkcjami. Nauczysz się wyciągać odpowiednie dane, złączać tabele, dodawać nowe dane, aktualizować te istniejące oraz usuwać niepotrzebne informacje. Ponadto dowiesz się, jak konstruować podzapytania i konwertować dane, oraz uświadomisz sobie, jakie różnice istnieją pomiędzy różnymi bazami. Te i wiele innych wiadomości znajdziesz w tym niezwykle przydatnym przewodniku po języku SQL!

- Typy danych oraz ich konwersje
- Funkcje operujące na danych i czasie
- Funkcje agregujące
- Indeksy – tworzenie i usuwanie
- Wprowadzanie danych
- Złączenia
- Pobieranie danych z bazy
- Funkcje tekstowe
- Podzapytania
- Tworzenie tabel
- Aktualizacja danych

Zapanuj nad ogromem informacji w bazie danych!

Spis treści

Wprowadzenie	5
Funkcje analityczne	10
Wyrażenia CASE — proste wyrażenia	10
Wyrażenia CASE — przeszukiwane wyrażenia	10
Funkcja CAST	11
Zapytania CONNECT BY	12
Konwersje typów danych	19
Typy danych — liczby całkowite (wartości binarne)	20
Typy danych — ciągi znaków	20
Typy danych — data i czas	21
Typy danych — liczby rzeczywiste	25
Konwersja daty i czasu — DB2	27
Konwersja daty i czasu — MySQL	30
Konwersja daty i czasu — Oracle	36
Konwersja daty i czasu — PostgreSQL	40
Konwersja daty i czasu — SQL Server	43
Funkcje operujące datami i czasem — DB2	47
Funkcje operujące datami i czasem — MySQL	48
Funkcje operujące datami i czasem — Oracle	50
Funkcje operujące datami i czasem — PostgreSQL	54
Funkcje operujące datami i czasem — SQLServer	56
Usuwanie danych	59
Funkcja EXTRACT	64
Funkcja GREATEST	64
Funkcje grupowania i sumowania	65

Zapytania hierarchiczne	76
Indeksy — tworzenie	81
Indeksy — usuwanie	82
Wprowadzanie danych	83
Złączanie tabel	89
Funkcja LEAST	101
Literały	101
Scalanie danych	106
Wartości NULL	108
Konwersja wartości liczbowych — DB2	113
Konwersja wartości liczbowych — MySQL	115
Konwersja wartości liczbowych — Oracle	116
Konwersja wartości liczbowych — PostgreSQL	118
Konwersja wartości liczbowych — SQL Server	120
Funkcje liczbowe i matematyczne	121
Funkcje OLAP	124
Transpozycja tabel	124
Predykaty	133
Zapytania rekurencyjne	137
Wyrażenia regularne	138
Pobieranie danych	150
Funkcje tekstowe	162
Podzapytania	169
Tabele — tworzenie	174
Tabele — usuwanie	179
Tabele — modyfikacja	180
Zarządzanie transakcjami	186
Unie	196
Uaktualnianie danych	203
Funkcje okienkowe	209
Skorowidz	219

Konwersja daty i czasu — DB2

Twórcy platformy DB2 włożyli ostatnio wiele wysiłku w zaimplementowanie emulacji funkcji `TO_CHAR` i `TO_DATE` znanych z bazy danych Oracle. Z tego względu wszyscy projektanci baz danych, którym zależy na zapewnieniu zgodności z systemem Oracle, powinni zapoznać się z zasadami działania wymienionych funkcji. Ich zastosowanie zostało opisane w punkcie „Konwersje daty i czasu — Oracle”.

Jeśli zgodność z platformą Oracle nie jest wymagana, konwersję wartości dat, czasu i znaczników czasowych można przeprowadzić za pomocą opisanych poniżej funkcji. Parametr określany jako *data_czas* może odpowiadać dacie, czasowi lub znacznikowi czasowemu. Argument zapisany jako *data* może być jedynie datą lub znacznikiem czasowym. Z kolei parametr *czas* może oznaczać czas lub znacznik czasowy. Jedynie argument *znacznik_czasowy* określa wyłącznie znacznik czasowy. Analogicznie *zakres_dat*

odpowiada okresowi opisanemu za pomocą dat. Parametr *zakres_czasu* wyznacza okres definiowany przez wartości czasu lub znaczników czasowych. Natomiast *zakres_znaczników_czasowych* reprezentuje okres opisany za pomocą znaczników czasowych. Dozwolone są również wszystkie tekstowe odpowiedniki stosowanych wartości.

```
BIGINT(data_czas)
CHAR(data_czas, [ISO|USA|EUR|JIS|LOCAL])
DATE(data)
DATE(liczba_calkowita)
DATE('yyyyddd')
DAY(data)
DAY(zakres_dat)
DAYNAME(data)
DAYOFWEEK(data)
DAYOFWEEK_ISO(data)
DAYOFYEAR(data)
DAYS(data)
DECIMAL(data_czas[,precyzja[,skala]])
GRAPHIC(data_czas, [ISO|USA|EUR|JIS|LOCAL])
HOUR(czas)
HOUR(zakres_czasu)
INTEGER(tylko_data)
INTEGER(tylko_czas)
JULIAN_DAY(data)
MICROSECOND(znacznik_czasu)
MICROSECOND(zakres_znaczników_czasowych)
MIDNIGHT_SECONDS(czas)
MINUTE(czas)
MINUTE(zakres_czasu)
MONTH(data)
MONTH(zakres_dat)
MONTHNAME(data)
QUARTER(data)
SECOND(czas)
SECOND(zakres_czasu)
TIME(czas)
TIMESTAMP(znacznik_czasu)
TIMESTAMP(data, czas)
TIMESTAMP_FORMAT(ciąg_tekstowy, 'YYYY-MM-DD HH24:MI:SS')
TIMESTAMP_ISO(data_czas)
TO_CHAR(znacznik_czasu, 'YYYY-MM-DD HH24:MI:SS')
TO_DATE(ciąg_tekstowy, 'YYYY-MM-DD HH24:MI:SS')
VARCHAR(data_czas)
```

```

VARCHAR_FORMAT(znacznik_czasu, 'YYYY-MM-DD HH24:MI:SS')
VARGRAPHIC(data_czas, [ISO|USA|EUR|JIS|LOCAL])
WEEK(data)
WEEK_ISO(data)
YEAR(data)
YEAR(zakres_dat)

```

W następującym przykładzie wykorzystano niektóre z wymienionych funkcji do utworzenia ciągu tekstowego odpowiadającego wartościom kolumny data_potwierdzenia.

```

SELECT w.id,
       RTRIM(CHAR(DAY(w.data_potwierdzenia))) || ' '
|| MONTHNAME(w.data_potwierdzenia) || ' '
|| RTRIM(CHAR(YEAR(w.data_potwierdzenia))) potwierdzono
FROM wodospady w;

```

ID	POTWIERDZONO
1	8 grudzień 2005
2	8 grudzień 2005
3	8 grudzień 2005
4	8 grudzień 2005

...

Funkcje wymagające określenia parametrów daty, czasu lub znacznika czasowego akceptują również ciągi tekstowe, które mogą być poddane niejawniej konwersji na wartości wymienionych typów. Oto przykład:

```

SELECT DATE('2003-11-7') ,
       TIME('21:25:00'),
       TIMESTAMP('2003-11-7 21:25:00.00')
FROM przestawna WHERE x = 1;

```

Za pomocą funkcji CHAR można formatować daty, czas i znaczniki czasowe w różny sposób, zależnie od wartości drugiego parametru.

```

SELECT CHAR(current_date, ISO),
       CHAR(current_date, LOCAL),
       CHAR(current_date, USA)
FROM przestawna WHERE x=1;

```

```

2003-11-06 2003-11-06 11/06/2003

```

Funkcja `DATE` umożliwia przekształcenie liczby całkowitej na wartość daty. Wartości liczbowe muszą mieścić się w przedziale od 1 do 3 652 059, przy czym liczba 1 odpowiada dacie 01.01.0001. Funkcja `DATE` może również wykonać odwrotną konwersję.

```
SELECT DATE(716194), DAYS('1961-11-15')
FROM przestawna WHERE x=1;
```

```
1961-11-15 716194
```

Zastosowanie funkcji `DECIMAL` i `BIGINT` pozwala na uzyskanie wartości daty, czasu i znacznika czasowego w formie liczby rzeczywistej lub ośmiobajtowej liczby całkowitej zgodnych z formatem `rrrrmmdd`, `ggmmss` bądź `rrrrmmddggmmss.nnnnnnn1`.

```
SELECT DECIMAL(current_date),
       DECIMAL(current_time),
       DECIMAL(current_timestamp)
FROM przestawna
WHERE x=1;
```

```
20110227, 170632, 20110227170632,132511
```

Funkcja `JULIAN_DAY` zwraca liczbę dni, jakie upłynęły od 01.01.4713 p.n.e. (czyli od 1 stycznia -4713 r.). Dacie tej odpowiada zerowa wartość dnia. Odwrotna konwersja nie jest możliwa, gdyż nie istnieje stosowna funkcja.

Konwersja daty i czasu — MySQL

W bazie danych MySQL zostało zaimplementowanych wiele funkcji konwersji daty i czasu, z których pewne obsługują również znaczniki czasu typowe dla systemu Unix. Poszczególne rozwiązania zostały opisane w kolejnych punktach.

¹ W polskiej wersji DB2 po wykonaniu polecenia (które zawiera w swoim kodzie kropkę) w wyniku otrzymuje się przecinek — *przyp. tłum.*

Wyodrębnianie składowych daty i czasu

Następujące funkcje MySQL udostępniają określone składowe daty i czasu:

```
DAYOFWEEK(data)
WEEKDAY(data)
DAYOFMONTH(data)
DAYOFYEAR(data)
MONTH(data)
DAYNAME(data)
MONTHNAME(data)
QUARTER(data)
WEEK(data)
WEEK(data, pierwszy_dzień)
YEAR(data)
YEARWEEK(data)
YEARWEEK(data, pierwszy_dzień)
HOUR(czas)
MINUTE(czas)
SECOND(czas)
```

Aby otrzymać aktualną wartość daty w formacie *dd-miesiąc-rrrr*, należy wykonać następującą instrukcję:

```
SELECT CONCAT(DAYOFMONTH(CURRENT_DATE), '-',
              MONTHNAME(CURRENT_DATE), '-',
              YEAR(CURRENT_DATE));
```

```
27-luty-2011
```

W funkcjach, w których występuje parametr *pierwszy_dzień*, należy wskazać, czy tydzień rozpoczyna się od niedzieli (*pierwszy_dzien* = 0), czy od poniedziałku (*pierwszy_dzien* = 1).

Funkcje TO_DAYS i FROM_DAYS

Zastosowanie funkcji `TO_DAYS` pozwala na przekształcenie wartości daty na liczbę dni, które upłynęły od początku naszej ery (dzień 01.01.0001 jest uznawany za dzień 1).

```
SELECT TO_DAYS(CURRENT_DATE);
```

```
732110
```

Odwrotne przekształcenie zapewnia funkcja `FROM_DAYS`.

```
SELECT FROM_DAYS();
```

2004-06-13

Zastosowanie opisywanych funkcji ogranicza się jedynie do przetwarzania dat kalendarza gregoriańskiego, czyli występujących po 15 października 1582 r. Wartości zwracane przez funkcje `TO_DAYS` i `FROM_DAYS` dla dat wcześniejszych są błędne.

Obsługa znacznika czasowego Unix

Przedstawione w tym punkcie funkcje odpowiadają za konwersję znacznika czasowego charakterystycznego dla systemów Unix.

```
UNIX_TIMESTAMP([data])
```

Funkcja ta zwraca znacznik czasowy systemu Unix, który jest wartością całkowitą bez znaku, przechowującą liczbę sekund, które upłynęły od dnia 01.01.1970. Jeśli nie zostanie podany żaden parametr wywołania funkcji, zwracaną wartością jest aktualny znacznik czasowy. Argumentem *data* może być ciąg tekstowy daty, ciąg tekstowy daty i czasu, ciąg tekstowy znacznika czasowego lub ich liczbowe odpowiedniki.

```
FROM_UNIXTIME(znacznik_czasowy_Unix[, format])
```

Funkcja ta przekształca znacznik czasowy systemu Unix w ciąg tekstowy daty i czasu, sformatowany zgodnie z parametrem *format* do postaci pozwalającej na wyświetlenie uzyskanej wartości na ekranie. Lista znaczników formatu została zamieszczona w tabeli 1., przedstawionej w dalszej części tego punktu.

Przykładowo, aby przekształcić datę 04.01.2004 19:18 w liczbę sekund, które upłynęły od 01.01.1970, należy użyć następującej instrukcji:

```
SELECT UNIX_TIMESTAMP(20040104191800);
```

```
1073240280
```

Aby przekształcić znacznik czasowy w ciąg tekstowy czytelny dla człowieka, należy zastosować instrukcję:

```
SELECT FROM_UNIXTIME(1073240280, '%d.%m.%Y godz.:%H:%i:%S');
```

```
04.01.2004 godz.: 19:18:00
```

Parametr *format* ma charakter opcjonalny. Domyślny format daty, dla przedstawionego wcześniej przypadku, umożliwia uzyskanie wartości: 2004-01-04 19:18:00.

Funkcje operujące sekundami dnia

Baza danych MySQL udostępnia dwie funkcje pozwalające na operowanie liczbą sekund, które upłynęły w danym dniu:

`SEC_TO_TIME(sekundy)`

Funkcja ta przekształca liczbę sekund, które upłynęły od północy, na ciąg tekstowy o postaci gg:mm:ss.

`TIME_TO_SEC(czas)`

Funkcja ta przekształca czas w liczbę sekund, które upłynęły od północy.

Oto stosowny przykład:

```
SELECT TIME_TO_SEC('19:18');
```

```
69480
```

```
SELECT SEC_TO_TIME(69480);
```

```
19:18:00
```

Funkcje DATE_FORMAT i TIME_FORMAT

Obydwie te funkcje cechuje niezwykle duża elastyczność w przekształcaniu wartości daty i czasu na postać ciągów tekstowych. Instrukcja DATE_FORMAT odpowiada za operacje na datach, natomiast TIME_FORMAT jest przeznaczona do przetwarzania informacji o czasie.

```
SELECT DATE_FORMAT(CURRENT_DATE, '%W, %d %M %Y');
```

piątek, 11 marzec 2011

Drugi parametr obydwu funkcji jest ciągiem tekstowym formatu. Występujące we wspomnianym ciągu znaczniki formatu są zastępowane przez odpowiednie elementy składowe daty i czasu. Ich opis znajduje się w tabeli 1. Pozostałe znaki występujące w ciągu formatu (takie jak prezentowane w przykładzie przecinek i spacja) są umieszczane w ciągu wyjściowym w miejscach, w których występują w ciągu formatu.

Tabela 1. Znaczniki formatu daty w bazie danych MySQL²

Znacznik formatu	Opis
%a	Skrót nazwy dnia tygodnia: <i>pon, wto, śro, ...</i>
%b	Skrót nazwy miesiąca: <i>sty, lut, mar, ...</i>
%c	Numer miesiąca: <i>1, 2, 3, ...</i>
%D	Dzień miesiąca z sufiksem liczebnika porządkowego (występującym w języku angielskim): <i>1st, 2nd, 3rd, ...</i>
%d	Dzień miesiąca; wartość dwucyfrowa: <i>01, 02, 03, ...</i>
%e	Dzień miesiąca: <i>1, 2, 3, ...</i>
%f	Mikrosekundy: <i>000000 – 999999</i>
%H	Godzina; wartość dwucyfrowa; format 24-godzinny: <i>00...23</i>

² Aby włączyć wyświetlanie polskich nazw miesięcy i dni tygodnia, należy zmienić wartość zmiennej `lc_time_names`, wykonując instrukcję `SET lc_time_names = 'pl_PL'` — *przypr. tłum.*

Tabela 1. Znaczniki formatu daty w bazie danych MySQL — ciąg dalszy

Znacznik formatu	Opis
%h	Godzina; wartość dwucyfrowa; format 12-godzinny: 01..12
%I	Godzina; wartość dwucyfrowa; format 12-godzinny: 01..12
%i	Minuty: 00, 01, ... 59
%j	Numer dnia w roku: 001...366
%k	Godzina; format 24-godzinny: 0, 1, ... 23
%L	Godzina; format 12-godzinny: 1, 2, ... 12
%M	Nazwa miesiąca: styczeń, luty, ...
%m	Numer miesiąca: 01, 02, ... 12
%p	Oznaczenie pory dnia (występujące w języku angielskim): AM, PM
%r	Czas w danym dniu w formacie 12-godzinnym, np. 12:15:05 PM
%S	Sekundy: 00, 01, ... 59
%s	Działanie znacznika jest takie samo jak znacznika %S
%T	Czas w danym dniu w formacie 24-godzinnym, np. 12:15:05
%U	Numer tygodnia, w którym niedziela jest pierwszym dniem: 00, 01, ...53
%u	Numer tygodnia, w którym poniedziałek jest pierwszym dniem: 00, 01, ...53
%V	Numer tygodnia, w którym niedziela jest pierwszym dniem; zliczenie rozpoczyna się od wartości 01 i jest wykonywane analogicznie jak w przypadku znacznika %X: 01, 02, ...53
%v	Numer tygodnia, w którym poniedziałek jest pierwszym dniem; zliczenie rozpoczyna się od wartości 01 i jest wykonywane analogicznie jak w przypadku znacznika %x: 01, 02, ... 53
%W	Nazwa dnia tygodnia: poniedziałek, wtorek, ...
%w	Numer dnia tygodnia: 0 = niedziela, 1 = poniedziałek, ...
%X	Rok danego tygodnia; wartość czterocyfrowa; za pierwszy dzień tygodnia uznawana jest niedziela; przetwarzanie analogiczne jak w przypadku znacznika %V
%x	Rok danego tygodnia; wartość czterocyfrowa; za pierwszy dzień tygodnia uznawany jest poniedziałek; przetwarzanie analogiczne jak w przypadku znacznika %v
%Y	Czterocyfrowa wartość roku: 2003, 2004, ...
%y	Dwucyfrowa wartość roku: 03, 04, ...
%%	Umieszczenie w ciągu wynikowym znaku wartości procentowej (%)

Konwersja daty i czasu — Oracle

Korzystając z bazy danych Oracle do konwersji wartości daty i czasu, można zastosować następujące funkcje:

```
TO_CHAR({data_czas|interwał}, format)
TO_DATE(ciąg_tekstowy, format)
TO_TIMESTAMP(ciąg_tekstowy, format)
TO_TIMESTAMP_TZ(ciąg_tekstowy, format)
TO_DSINTERVAL('D HH:MI:SS')
TO_YMINTERVAL('Y-M')
NUMTODSINTERVAL(liczba, 'jednostka_ds')
NUMTOYMINTERVAL(liczba, 'jednostka_ym')

jednostka_ds ::= {DAY|HOUR|MINUTE|SECOND}
jednostka_ym ::= {YEAR|MONTH}
```

Parametr *format* umożliwia określenie tekstowej prezentacji wartości daty i czasu. Oto przykładowa definicja formatu wyświetlania daty:

```
SELECT nazwa,
       TO_CHAR(data_potwierdzenia, 'dd-Mon-yyyy') cdate
FROM wodospady;

Munising Falls    08-Gru-2005
Tannery Falls     08-Gru-2005
Alger Falls       08-Gru-2005
...
```

Możliwa jest również konwersja w przeciwnym kierunku:

```
INSERT INTO wodospady (id, nazwa, data_potwierdzenia)
VALUES (15, 'Tahquamenon',
       TO_TIMESTAMP('29-Dec-2006', 'dd-Mon-yyyy'));
```

Znaczniki składające się na *ciąg formatu* zostały przedstawione w tabeli 2. Wyniki generowane przez większość z nich zależą od języka ustanowionej sesji (tzn. jeśli sesja została ustanowiona w języku francuskim, zwracane są francuskie nazwy miesięcy).

Tabela 2. Znaczniki formatowania daty w bazie danych Oracle

Znacznik formatowania	Opis
AM lub PM A.M. lub P.M.	Wyznacznik pory dnia (AM — przed południem; PM — po południu)
BC lub AD B.C. lub A.D.	Wyznacznik ery (BC — przed naszą erą; AC — nasza era)
CC	Wiek. Dostępny jedynie w odniesieniu do danych wyjściowych
D	Dzień tygodnia. Parametr NLS_TERRITORY określa, któremu dniowi tygodnia jest przypisywana wartość 1
DAY, Day lub day	Nazwa dnia tygodnia
DD	Numer dnia w miesiącu
DDD	Numer dnia w roku
DL	Dłuższy ciąg daty. Dostępny jedynie w danych wyjściowych. Znajduje zastosowanie jedynie w połączeniu z parametrem TS
DS	Krótszy ciąg daty. Dostępny jedynie w danych wyjściowych. Znajduje zastosowanie jedynie w połączeniu z parametrem TS
DY, Dy lub dy	Skrót nazwy dnia. Wielkość znaków skrótu odpowiada wielkości znaków ciągu formatu
E	Skrót nazwy ery dla japońskiego kalendarza imperialnego (ang. <i>Japanese Imperial</i>), oficjalnego kalendarza Republiki Chin (ang. <i>ROC Official</i>) i tajskiego kalendarza buddyjskiego (ang. <i>Thai Buddha</i>). Stosowany jedynie w odniesieniu do danych wejściowych
EE	Pełna nazwa ery
FF, FF1...FF9	Części sekundy. Parametr stosowany jedynie w odniesieniu do wartości typu TIMESTAMP. Zawsze należy używać dwóch wartości F. Parametr FF1...FF9 jest dostępny jedynie w bazie danych Oracle Database 10g i późniejszych
FM	Eliminuje puste znaki z ciągu wyjściowego
FX	Wprowadza wymóg dokładnej zgodności danych wejściowych z określonym formatem
HH lub HH12	Godzina dnia. Wartości od 1 do 12. Parametr HH12 jest stosowany jedynie w odniesieniu do danych wyjściowych

Tabela 2. Znaczniki formatowania daty w bazie danych Oracle — ciąg dalszy

Znacznik formatowania	Opis
HH24	Godzina dnia. Wartości od 0 do 23
IW	Tydzień standardu ISO w danym roku. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
IYY lub IY lub I	Ostatnie trzy, dwie lub jedna cyfra roku zapisanego w standardzie ISO. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
IYYY	Rok standardu ISO. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
J	Data kalendarza juliańskiego. 01.01.4712 p.n.e. jest uznawany za dzień o wartości 1
MI	Minuty
MM	Numer miesiąca
MON, Mon lub mon	Skrót nazwy miesiąca
MONTH, Month lub month	Nazwa miesiąca
Q	Kwartał roku. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
RM lub ɹm	Rzymska cyfra odpowiadająca numerowi miesiąca
RR	Dwie ostatnie cyfry roku. Wykorzystuje przesunięte okno stuleci: 00 – 49 = 20xx, 50 – 99 = 19xx
RRRR	Czterocyfrowa wartość roku (w ciągu wejściowym akceptowana jest również wartość dwucyfrowa). Okno stuleci jest takie samo jak w przypadku parametru RR
SCC	Włączenie ujemnych wartości dat sprzed naszej ery. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
SP	Sufiks przekształcający wartość liczbową na zapis słowny
SPTH	Sufiks przekształcający wartość liczbową na zapis słowny i format porządkowy
SS	Sekundy
SSSS	Liczba sekund, które upłynęły od północy

Tabela 2. Znaczniki formatowania daty w bazie danych Oracle — ciąg dalszy

Znacznik formatowania	Opis
SYEAR, SYear lub syear	Słowna wartość roku. Daty sprzed naszej ery mają wartość ujemną. Wielkość znaków odpowiada wielkości znaków ciągu formatu. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
YYYY	Czterocyfrowa wartość roku. Daty sprzed naszej ery mają wartość ujemną
TH lub th	Sufiks przekształcający liczby na wartości porządkowe
TS	Format skróconego ciągu czasu. Parametr stosowany jedynie w odniesieniu do danych wyjściowych i w połączeniu z parametrami DL i DS
TZD	Skrót nazwy strefy czasowej. Parametr stosowany jedynie w odniesieniu do danych wejściowych
TZH	Godzinowa różnica czasu w odniesieniu do czasu UTC
TZM	Minutowa różnica czasu w odniesieniu do czasu UTC
TZR	Strefa czasowa
W	Numer tygodnia w miesiącu (od 1 do 5). Tydzień o wartości 1 rozpoczyna się pierwszego dnia miesiąca i kończy siódmego dnia miesiąca. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
WW	Numer tygodnia w roku (od 1 do 53). Parametr stosowany jedynie w odniesieniu do danych wyjściowych
X	Znak lokalnego systemu, wykorzystywany do oznaczania separatora dziesiętnego. W języku polskim odpowiada przecinkowi (w amerykańskiej odmianie angielskiego — kropce)
Y, YYYY	Czterocyfrowa wartość roku ze znakiem przecinka
YEAR, Year lub year	Słowna wartość roku. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
YYY lub YY lub Y	Ostatnie trzy, dwie cyfry lub jedna cyfra roku
YYYY	Czterocyfrowa wartość roku

Podczas konwersji *na* tekst wielkość znaków w ciągach wynikowych (np. w skrótach miesięcy) jest uzależniona od wielkości liter znacznika formatu. Zatem zastosowanie znacznika 'Mon' spowoduje zawrócenie skrótów 'Sty' lub 'Lut'. Znacznik 'mon' wygeneruje wartości 'sty' i 'lut'. Natomiast 'MON' zwraca tekst 'STY' lub 'LUT'. Podczas przekształcania wartości z formatu tekstowego wielkość znaków nie jest istotna.

W przypadku wszystkich funkcji ciąg formatu jest opcjonalny. Może zostać pominięty, jeśli wartości danych wejściowych spełniają kryteria domyślnych formatów daty, określonych przez parametry NLS_DATE_FORMAT w przypadku dat, NLS_TIMESTAMP_FORMAT w przypadku znaczników czasu oraz NLS_TIMESTAMP_TZ_FORMAT w przypadku znaczników czasu zawierających informacje o strefach czasowych. Ustawienia NLS można sprawdzić, kierując zaopiniowanie do tabeli NLS_SESSION_PARAMETERS.

Konwersja daty i czasu — PostgreSQL

Do przekształcania wartości daty i czasu na ciągi tekstowe i odwrotnie służą następujące funkcje:

```
TO_CHAR({timestamp|interval}, format)
TO_DATE(string, format)
TO_TIMESTAMP(string, format)
```

Na przykład, aby uzyskać tekstową reprezentację znacznika czasowego, można zastosować instrukcję:

```
SELECT w.nazwa,
       TO_CHAR(w.data_potwierdzenia, 'dd-Mon-YYYY')
FROM wodospady w;
```

nazwa	to_char
Munising Falls	08-Dec-2005
Tannery Falls	08-Dec-2005
Alger Falls	08-Dec-2005
...	

Aby wykonać operację odwrotną (zamienić ciąg tekstowy na datę), wystarczy wykonać polecenie:

```
SELECT TO_DATE('8-Dec-2005', 'dd-mon-yyyy');
```

Obsługa znaczników formatu w bazach danych PostgreSQL jest bardzo zbliżona do obowiązującej na platformie Oracle. Lista dostępnych specyfikatorów została zamieszczona w tabeli 3. Wielkość znaków wykorzystanych w ciągu formatu wyznacza sposób zapisu ciągu wynikowego. Na przykład zastosowanie znacznika 'MON' spowoduje zwrócenie skrótów 'JAN' lub 'FEB'. Znacznik 'Mon' wygeneruje wartości 'Jan' i 'Feb'. Natomiast 'mon' zwraca tekst 'jan' lub 'feb'. Podczas przekształcania wartości z formatu tekstowego wielkość znaków nie jest istotna.

Tabela 3. Znaczniki formatowania daty w bazie danych PostgreSQL

Znacznik formatu	Opis
AM lub PM A.M. lub P.M.	Wyznacznik pory dnia (AM — przed południem; PM — po południu)
BC lub AD B.C. lub A.D.	Wyznacznik ery (BC — przed naszą erą; AC — nasza era)
CC	Wiek. Dostępny jedynie w odniesieniu do danych wyjściowych
D	Dzień tygodnia
DAY, Day lub day	Nazwa dnia tygodnia
DD	Numer dnia w miesiącu
DDD	Numer dnia w roku
DY, Dy lub dy	Skrót nazwy dnia
FM	Eliminuje puste znaki z ciągu wyjściowego. Dostępny jedynie w odniesieniu do danych wyjściowych
FX	Wprowadza wymóg dokładnej zgodności danych wejściowych z określonym formatem
HH lub HH12	Godzina dnia. Wartości od 1 do 12. Parametr HH12 jest stosowany jedynie w odniesieniu do danych wyjściowych
HH24	Godzina dnia. Wartości od 0 do 23
IW	Tydzień standardu ISO w danym roku. Parametr stosowany jedynie w odniesieniu do danych wyjściowych

Tabela 3. Znaczniki formatowania daty w bazie danych PostgreSQL
— ciąg dalszy

Znacznik formatu	Opis
IYY lub IY lub I	Ostatnie trzy, dwie cyfry lub jedna cyfra roku zapisanego w standardzie ISO. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
IYYY	Rok standardu ISO. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
J	Data kalendarza juliańskiego. 01.01.4712 p.n.e. jest uznawany za dzień o wartości 1
MI	Minuty
MM	Numer miesiąca
MON, Mon lub mon	Skrót nazwy miesiąca
MONTH, Month lub month	Nazwa miesiąca
MS	Liczba milisekund
Q	Kwartał roku. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
RM lub rm	Rzymska cyfra odpowiadająca numerowi miesiąca
SP	Sufiks przekształcający wartość liczbową na zapis słowny (niezaimplementowany)
SS	Sekundy
SSSS	Liczba sekund, które upłynęły od północy
TH lub th	Sufiks przekształcający liczby na wartości porządkowe
TZ lub tz	Nazwa strefy czasowej
US	Liczba mikrosekund
W	Numer tygodnia w miesiącu (od 1 do 5). Tydzień o wartości 1 rozpoczyna się pierwszego dnia miesiąca i kończy siódmego dnia miesiąca. Parametr stosowany jedynie w odniesieniu do danych wyjściowych
WW	Numer tygodnia w roku (od 1 do 53). Parametr stosowany jedynie w odniesieniu do danych wyjściowych
Y, YYY	Czterocyfrowa wartość roku ze znakiem przecinka
YYY lub YY lub Y	Ostatnie trzy, dwie cyfry lub jedna cyfra roku
YYYY	Czterocyfrowa wartość roku

Ostrzeżenie

Do funkcji `TO_CHAR` nie można przekazywać wartości typu `TIME`.

Za przekształcanie znacznika czasowego systemu Unix na wartość znacznika czasowego PostgreSQL odpowiada funkcja `TO_TIMESTAMP`.

```
SELECT TO_TIMESTAMP(0);
```

Dolny zakres dat reprezentowanych przez znaczniki systemu Unix odpowiada dacie 1 stycznia 1970 r. UTC.

Konwersja daty i czasu — SQL Server

Oprogramowanie SQL Server udostępnia cztery opcje konwersji wartości daty i czasu. Zazwyczaj najlepszym rozwiązaniem jest zastosowanie funkcji `CONVERT`, choć funkcje `DATENAME` i `DATEPART` oferują znacznie większą elastyczność w przekształcaniu dat na postać tekstową.

Funkcje `CAST` i `SET DATEFORMAT`

Oprócz obsługi standardowej funkcji `CAST` środowisko SQL Server umożliwia również definiowanie formatu daty i czasu za pomocą polecenia `SET DATEFORMAT`:

```
SET DATEFORMAT dmy
SELECT CAST('1/12/2004' AS datetime)
```

```
2004-12-01 00:00:00.000
```

W przypadku dat zapisanych w formacie niebudzącym wątpliwości nie ma potrzeby modyfikowania ustawień `DATEFORMAT`³.

³ Aby włączyć obsługę polskich nazw miesięcy i dni, należy zmienić domyślne ustawienia językowe, np. za pomocą instrukcji `SET LANGUAGE polski` — *przykład tłum.*

```
SET DATEFORMAT dmy
SELECT CAST('12 styczeń 2004' AS datetime)
```

```
2004-01-12 00:00:00.000
```

W poleceniu SET DATEFORMAT można umieszczać następujące argumenty: *mdy*, *dmy*, *ymd*, *myd* oraz *dym*.

Funkcja CONVERT

Ogólna konwersja wartości daty i czasu jest realizowana przez funkcję CONVERT.

```
CONVERT(wartość_daty[(długość)], wyrażenie[, style])
```

Drugi (opcjonalny) parametr — *style* — pozwala na określenie pierwotnego i docelowego formatu wartości daty i czasu, zależnie od tego, czy konwersja jest przeprowadzana z ciągu tekstowego, czy na ciąg tekstowy. Lista obsługiwanych znaczników stylu została przedstawiona w tabeli 4.

W następującym przykładzie zademonstrowano jeden ze sposobów konwersji daty i czasu na ciąg tekstowy i odwrotnie.

```
SELECT CONVERT(
    VARCHAR,
    CONVERT(DATETIME, '15 styczeń 1961', 106), 104)
```

```
15.01.1961
```

Aby wyznaczyć długość wynikowego ciągu tekstowego, należy określić właściwą wartość parametru *długość*. Natomiast w celu uzyskania dwucyfrowej wartości roku można odjąć wartość 100 od liczby wyznaczającej styl formatowania.

```
SELECT CONVERT(DATETIME, '1/1/50', 1)
1950-01-01 00:00:00.000
```

```
SELECT CONVERT(DATETIME, '49.1.1', 2)
2049-01-01 00:00:00.000
```

Tabela 4. Znaczniki stylu prezentacji daty i czasu w bazie danych SQL Server

Znacznik stylu	Opis
0, 100	Ustawienie domyślne: mies dd rrrr gg:miAM (lub PM)
101 ^a	Format USA: mm/dd/yyyy
102 ^a	Format ANSI: rrrr.mm.dd
103 ^a	Format brytyjski i francuski: dd/mm/yyyy
104 ^a	Format niemiecki: dd.mm.rrrr
105 ^a	Format włoski: dd-mm-rrrr
106 ^a	dd mies rrrr
107 ^a	mies dd, rrrr
108 ^a	gg:mm:ss
9, 109	Ustawienie domyślne z uwzględnieniem milisekund: mies dd rrrr gg:mi:ss:mmAM (lub PM)
110 ^a	Format USA: mm-dd-rrrr
111 ^a	Format japoński: rrrr/mm/dd
112 ^a	Format ISO: rrrrmmdd
13, 113	Domyślne ustawienie europejskie z uwzględnieniem milisekund: dd mies rrrr gg:mm:ss:mmm
114 ^a	gg:mi:ss:mmm z 24-godzinnym formatem czasu
20, 120	Podstawowy format ODBC z 24-godzinnym formatem czasu: rrrr-mm-dd gg:mi:ss
21, 121	Podstawowy format ODBC z 24-godzinnym formatem czasu i z uwzględnieniem milisekund: rrrr-mm-dd gg:mi:ss:mmm
126	Format ISO8601, bez znaków spacji: rrrr-mm-ddTgg:mm:ss:mmm
130	Format kuwejcki: dd mies rrrr gg:mi:ss:mmAM
131	Format kuwejcki: dd/mm/yyyy gg:mi:ss:mmAM

^a Aby uzyskać dwucyfrowy format roku, należy odjąć 100 od podanej wartości.

W przypadku stosowania dwucyfrowych oznaczeń roku baza danych SQL Server uznaje wartość 2049 za wartość maksymalną. Lata od 50 do 99 są interpretowane jako 1950 – 1999. Z kolei wartości od 00 do 49 są uznawane za lata 2000 – 2049. Taki sposób interpretacji wartości lat został pokazany w prezentowanych

przykładach. Należy jednak pamiętać, że wartość progowa roku może zostać zmieniona przez administratora bazy danych (za pomocą opcji *two digit year cutoff*).

Funkcje DATENAME i DATEPART

Funkcje DATENAME i DATEPART służą do wyodrębniania określonych elementów daty i czasu.

```
DATENAME(element_daty, data_czas)  
DATEPART(element_daty, data_czas)
```

Różnica między wymienionymi funkcjami polega na tym, że funkcja DATENAME zwraca tekstową reprezentację wybranego elementu daty, natomiast funkcja DATEPART udostępnia liczbową wartość wskazanego elementu daty. Oto przykład:

```
SELECT DATENAME(month, GETDATE()),  
       DATEPART(month, GETDATE())
```

```
marzec          3
```

Niektóre elementy, takie jak rok (*year*) i dzień (*day*), są reprezentowane wartościami liczbowymi niezależnie od zastosowanej funkcji. Niemniej przeznaczeniem wspomnianych funkcji jest zagwarantowanie dostarczenia wartości tekstowych bądź liczbowych. W przedstawionym przykładzie obydwie funkcje zwracają wartość roku, ale funkcja DATENAME udostępnia ciąg tekstowy '2004', natomiast funkcja DATEPART zwraca liczbę 2004:

```
SELECT DATENAME(year, GETDATE()),  
       DATEPART(year, GETDATE())
```

Jako słowa kluczowe oznaczające poszczególne składowe daty można stosować następujące ciągi tekstowe: *year*, *yy*, *yyyy*, *quarter*, *qq*, *q*, *month*, *mm*, *m*, *dayofyear*, *dy*, *y*, *day*, *dd*, *d*, *week*, *wk*, *ww*, *weekday*, *dw*, *hour*, *hh*, *minute*, *mi*, *n*, *second*, *ss*, *s*, *millisecond*, *ms*, *microsecond*, *mcs*, *nanoseconds*, *ns*, *TZoffset*, *tz*, *ISO_Week*, *isowk*, *isoww*.

Funkcje DAY, MONTH, YEAR

Baza danych SQL Server udostępnia również kilka funkcji umożliwiających wyodrębnianie z dat wartości określonych składowych.

```
DAY(data_czas)  
MONTH(data_czas)  
YEAR(data_czas)
```

Oto przykład:

```
SELECT DAY(CURRENT_TIMESTAMP),  
       MONTH(CURRENT_TIMESTAMP),  
       YEAR(CURRENT_TIMESTAMP)
```

```
11           3           2011
```

Skorowidz

A

ABS(), 121
ADD, 182
ADD_MONTHS, 48, 52
ADDDATE, 49
ALL, 89, 156
ALTER SESSION, 50
ANSI_NULLS, 113
ARCHIVE, 176
AT TIME ZONE, 56
AVG(), 66

B

BIGINT, 23, 28, 30
BIN(), 116
BLACKHOLE, 176
BOTH, 167
BULK COLLECT, 62

C

CASE, 10, 60, 109
CAST, 11, 43
CEIL(), 121
CEILING(), 121
CENTURY, 55, 64
CHAR, 21, 28, 29, 114

COALESCE, 109
CONNECT BY, 12, 13, 14, 15
 funkcje, 18
 operatory klauzuli, 18
 pętle, 17
 sortowanie, 16
 złączenia, 15
CONNECT BY NOCYCLE, 18
CONNECT_BY_ISCYCLE, 17,
 18
CONNECT_BY_ISLEAF, 18
CONNECT_BY_ROOT, 18, 19
CONSTRAINT, 176
CONV(), 115
CONVERT, 43, 44, 120
COUNT(), 66
CREATE INDEX, 81
CREATE TABLE, 174, 175, 177,
 178, 179
CSV, 176
CUBE, 73, 75
CURDATE(), 48
CURRENT_DATE, 47, 48, 50,
 54
CURRENT_TIME, 47, 48, 54
CURRENT_TIMESTAMP, 47,
 49, 51, 54, 57
CURRENT_TIMEZONE, 47
CURTIME(), 48

D

dane, uaktualnianie, 203, 206
data i czas, 21
 konwersja, 27, 30, 36, 40, 43
 obcinanie wartości, 51, 52, 55
 pobieranie bieżącej
 wartości, 50
 rejestrowanie, 23
 wyodrębnianie
 składowych, 31
 zaokrąglanie wartości, 51,
 52, 55
DATE, 21, 22, 23, 24, 28, 30, 50,
 53
DATE_ADD, 49
DATE_FORMAT, 34
DATE_PART, 56
DATE_SUB, 49
DATE_TRUNC, 55
DATEADD, 57
DATEDIFF, 58
DATEFORMAT, 43
DATENAME, 43, 46
DATEPART, 43, 46
DATETIME, 22, 24
DATETIME2, 24, 58
DATETIMEOFFSET, 24, 58
DAY, 28, 47, 48, 49, 55, 64
DAY_HOUR, 50
DAY_MINUTE, 50
DAY_SECOND, 50
DAYNAME, 28, 31
DAYOFMONTH, 31
DAYOFWEEK, 28, 31
DAYOFYEAR, 28, 31
DAYS, 28, 48
DBTIMEZONE, 51

DECADE, 55, 64
DECFLOAT, 25
DECIMAL, 26, 27, 28, 30, 114
DEFAULT, 83
definiowanie nazw kolumn, 151
DELETE, 59, 60, 61
DENSE_RANK(), 213
DISTINCT, 65, 156
DOUBLE PRECISION, 23
DOW, 64
DOY, 64

E

ENGINE, 176
EPOCH, 64
EXAMPLE, 176
EXCEPT, 196, 200
EXCEPT ALL, 200, 201
EXP(), 121
EXTRACT, 56, 64

F

FEDERATED, 176
FIRST, 89
FLOAT, 120
FLOOR(), 121
FORMAT(), 115
FROM, 63, 157, 158, 206
FROM_DAYS, 31, 32
FROM_UNIXTIME(), 32
funkcja
 agregująca, 65
 analityczna, 10
 CAST, 11
 liczbowa, 121

funkcja
matematyczna, 121
okienkowa, 10, 209
operująca sekundami dnia,
33
tekstowa, 162

G

GETDATE(), 57
GETUTCDATE(), 57
GRAPHIC, 28
GREATEST, 64
GROUP BY, 67, 69, 75
GROUP_ID(), 75
GROUPING SETS, 74, 75
GROUPING(), 75
GROUPING_ID(), 75
grupowanie wierszy, 65

H

HAVING, 71
HEX(), 116
HOUR, 28, 31, 48, 49, 55, 64
HOUR_MINUTE, 50
HOUR_SECOND, 50
HOURS, 48

I

IGNORE NULLS, 215
indeks, 82
tworzenie, 81
usuwanie, 82
INNODB, 176
INTEGER, 28
INTERSECT, 196, 202

INTERSECT ALL, 202
INTERVAL, 55
IS NULL, 10
ISOLATION LEVEL READ
COMMITTED, 190
ISOLATION LEVEL
SERIALIZABLE, 190

J

JULIAN_DAY, 28, 30

K

kolejność przetwarzania
instrukcji, 199
konwersja typów danych, 19
konwersja wartości
liczbowych, 113, 115, 116,
118, 120
konwersja dat, 22

L

LAG, 214
LAST_DAY, 48, 53
LATERAL, 92
LEAD, 214
LEADING, 167
LEAST, 101
LEVEL, 19
LISTAGG, 68
literały, 101
daty i czasu, 104
interwałów, 105
liczbowe, 104
tekstowe, 102

LN(), 122
LOCAL, 54
LOCALTIME, 54
LOCALTIMESTAMP, 51, 54
LOG(), 122
LOG10(), 122
LTRIM, 167, 168

M

MAX(), 66
MEDIAN(), 66
MEMORY, 176
MERGE, 176
MICROSECOND, 28, 48
MICROSECONDS, 48, 55, 64
MIDNIGHT_SECONDS, 28
MILLENNIUM, 55, 64
MILLISECONDS, 55, 64
MIN(), 66
MINUS, 200
MINUTE, 28, 31, 48, 49, 55, 64
MINUTE_SECOND, 50
MINUTES, 48
MOD(), 122
MODIFY, 182
MONEY, 120
MONTH, 28, 31, 47, 48, 49, 55, 64
MONTHNAME, 28, 31
MONTHS, 48
MONTHS_BETWEEN, 48, 53
MYISAM, 176

N

NAME, 189
NANVL(), 122
nazwy w uniach, 199

NEXT_DAY, 48, 53
NLS_DATE_FORMAT, 40
NLS_SESSION_PARAMETERS,
40, 53
NLS_TIMESTAMP_FORMAT,
40
NLS_TIMESTAMP_TZ_
→FORMAT, 40
NOCYCLE, 17
NOW(), 49, 54
NTH_VALUE, 215
NULL, 108, 109, 110, 111, 112
NUMBER, 26
NUMTODSINTERVAL, 36
NUMTOYMINTERVAL, 36

O

OCT(), 116
OLAP, 124
ORDER BY, 59, 60, 160, 198,
213, 215
OUTPUT, 87

P

PARTITION BY, 215
PIVOT, 128
pobieranie danych, 150
podzapytania, 169, 170
porównywanie wartości, 213
predykaty, 133
 BETWEEN, 136
 EXISTS, 134
 IN, 135
 LIKE, 136
 NOT LIKE, 136
PRIOR, 13, 18, 19

Q

QUARTER, 28, 31, 64

R

RANK(), 212
READ ONLY, 189
READ WRITE, 190
REAL, 120
REMAINDER(), 122
ROLLUP, 71, 75
ROUND, 48, 123
ROUND_CEILING, 25
ROUND_DOWN, 26
ROUND_FLOOR, 26
ROUND_HALF_EVEN, 26
ROUND_HALF_UP, 26
ROW_NUMBER(), 212
RTRIM, 167

S

scalanie danych, 106
SEARCH BREADTH FIRST, 79
SEARCH DEPTH FIRST, 79
SEC_TO_TIME(), 33
SECOND, 28, 31, 48, 49, 55, 64
SECONDS, 48
SELECT, 151, 155
SESSIONTIMEZONE, 51
SET DATEFORMAT, 43, 44
SET TRANSACTION, 188
SIGN(), 123
słowo kluczowe
 NOCYCLE, 17
 PRIOR, 13

SMALLDATETIME, 24
SMALLMONEY, 120
sortowanie
 rozszerzające, 78
 w głąb, 79
 wszerz, 79
 zagłębiające, 78
START TRANSACTION, 188
START WITH, 12
STDDEV(), 66
SUBDATE, 49
SUM(), 66
sumowanie wierszy, 65
SWITCHOFFSET, 58
symbol skrótu, 151
SYS_CONNECT_BY_PATH, 19
SYSDATE, 49, 50, 51
SYSDATETIME(), 57
SYSDATETIMEOFFSET(), 57
SYSTIMESTAMP, 51
SYSUTCDATETIME, 57

T

tabela
 modyfikacja, 180, 182, 183,
 184, 185
 tworzenie, 174, 175, 177,
 178, 179
 usuwanie, 179
tekst
 wyodrębnianie fragmentu,
 164
 wyszukiwanie, 162
 zamiana, 164
test IS NULL, 10
TEXT, 20
TIME, 21, 22, 23, 24, 28, 43

TIME [WITH[OUT] TIME
ZONE], 23
TIME_FORMAT, 34
TIME_TO_SEC(), 33
TIMEOFDAY(), 55
TIMESTAMP, 21, 22, 23, 24, 28,
51, 53, 55
TIMESTAMP WITH LOCAL
TIME_ZONE, 22
TIMESTAMP WITH TIME
ZONE, 22, 51
TIMESTAMP_FORMAT, 28
TIMESTAMP_ISO, 28
TIMEZONE, 56, 64
TIMEZONE_ABBR, 64
TIMEZONE_HOUR, 64
TIMEZONE_MINUTE, 64
TIMEZONE_REGION, 64
TO_CHAR, 27, 28, 36, 40, 43
TO_DATE, 27, 28, 36, 40
TO_DAYS, 31, 32
TO_DSINTERVAL, 36
TO_TIMESTAMP, 36, 40
TO_TIMESTAMP_TZ, 36
TO_YMINTERVAL, 36
TODATETIMEOFFSET, 58
TRAILING, 167
transakcja, 186
 kończenie, 192
 przerwanie, 194, 195
 rozpoczęcie, 188, 189,
 190, 191
 zarządzanie, 186
transpozycja
 odwrotna, 129, 131, 132
 tabel, 124, 125, 128
TRIM, 167
TRUNC, 48, 123

tryb automatycznego
zatwierdzenia, 187
typ danych
 ciąg znaków, 20
 data i czas, 21
 konwersja, 19
 liczby całkowite, 20
 liczby rzeczywiste, 25
 w uniach, 199

U

uaktualnianie
 proste, 203
 danych, 203, 206
unie, 196
UNION, 196, 197
UNION ALL, 196, 197
UNIX_TIMESTAMP, 49
UNIX_TIMESTAMP(), 32
UPDATE, 206
USE ROLLBACK SEGMENT,
190
USING, 93
usuwanie
 danych, 59, 61
 wierszy, 60

V

VALUES, 159
VARCHAR, 20, 28
VARCHAR_FORMAT, 29
VARCHAR2, 21
VARGRAPHIC, 29
VARIABLE, 87
VARIANCE(), 66

W

warunki TRUE-FALSE, 10
warunkowe wprowadzanie
 danych, 88
WEEK, 29, 31, 55, 64
WEEK_ISO, 29
WEEKDAY, 31
WHERE, 15, 159
WITH, 12
WITH CUBE, 73
WITH LOCAL TIME ZONE, 22
WITH TIME ZONE, 22
wprowadzanie
 danych, 83, 87
 pojedynczych wierszy, 83
 wielu wierszy, 84
wyrażenia CASE, 10
wyrażenia regularne, 138, 142,
 150
wyszukiwanie tekstu, 162

Y

YEAR, 29, 31, 47, 48, 49, 55, 64
YEAR_MONTH, 50
YEARS, 48
YEARWEEK, 31

Z

zamiana tekstu, 164
zapisywanie wyrażań, 152
zapytania
 hierarchiczne, 76
 rekurencyjne, 77, 137
złączanie
 krzyżowe, 91, 92
 naturalne, 95
 tabel, 89
 wewnętrzne, 92
 wyznaczane za pomocą
 nierówności, 96
 zewnętrzne lewostronne, 97
 zewnętrzne prawostronne,
 99
zmniejszanie liczby kolumn, 69
znacznik czasowy Unix, 32
znak gwiazdki, 151
zwracanie
 uaktualnionych danych,
 207, 208
 usuwanych wierszy, 62, 63
 wprowadzanych wierszy, 87
 wprowadzonych wartości,
 86

SQL. Leksykon kieszonkowy



SQL to prawdopodobnie jeden z ostatnich wiekowych języków programowania, tak powszechnie używanych w codziennej pracy. Jego możliwości są ogromne. Jeżeli nauczysz się konstruować zapytania SQL, będziesz potrafił wyłowić dowolną informację z każdej bazy danych. Brzmi kusząco?

Dzięki temu przewodnikowi będziesz miał zawsze na wyciągnięcie ręki kompletny zbiór informacji na temat języka SQL. Już nigdy więcej nie zaskoczą Cię niuanse jego składni. Błyskawicznie zapoznasz się z typami danych, sposobami wykonywania operacji na danych oraz dostępnymi funkcjami. Nauczysz się wyciągać odpowiednie dane, złączać tabele, dodawać nowe dane, aktualizować te istniejące oraz usuwać niepotrzebne informacje. Ponadto dowiesz się, jak konstruować podzapytania i konwertować dane, oraz uświadomisz sobie, jakie różnice istnieją pomiędzy różnymi bazami. Te i wiele innych wiadomości znajdziesz w tym niezwykle przydatnym przewodniku po języku SQL!

- Typy danych oraz ich konwersje
- Funkcje operujące na danych i czasie
- Funkcje agregujące
- Indeksy – tworzenie i usuwanie
- Wprowadzanie danych
- Złączenia
- Pobieranie danych z bazy
- Funkcje tekstowe
- Podzapytania
- Tworzenie tabel
- Aktualizacja danych

Zapanuj nad ogromem informacji w bazie danych!

helion.pl
księgarnia internetowa



Helion

Nr katalogowy: 6624

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

📖 Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>



KOD KORZYŚCI



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



Zamówienia telefoniczne:
0 801 339900



0 601 339900

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

ISBN 978-83-246-3306-7



9 788324 633067

Cena 29,90 zł

Informatyka w najlepszym wydaniu