

### Funkcje numeryczne

Składnia	Przeznaczenie	Przykład
ABS(n)	Zwraca wartość absolutną liczby n	ABS(-15) Wynik: 15
CEIL(n)	Zwraca najmniejszą liczbę całkowitą większą lub równą n	CEIL(15.7) Wynik: 16
FLOOR(n)	Zwraca największą liczbę całkowitą mniejszą lub równą n	FLOOR(15.7) Wynik: 15
MOD(m, n)	Zwraca resztę z dzielenia liczby m przez n	MOD(7, 5) Wynik: 2
POWER(m, n)	Zwraca liczbę m podniesioną do potęgi n. Liczba n musi być całkowita; w przeciwnym wypadku wystąpi błąd.	POWER(2, 3) Wynik: 8
ROUND(n[, m])	Zwraca liczbę n zaokrągloną do m miejsc po przecinku. Jeśli m jest pominięte, to przyjmuje się 0. Liczba m może być dodatnia lub ujemna (zaokrąglenie do odpowiedniej liczby cyfr przed przecinkiem)	ROUND(16.167, 1) Wynik: 16.2 ROUND(16, 167, -1) Wynik: 20
SIGN(n)	Zwraca 0, jeśli n jest równe 0, -1 jeśli n jest mniejsze od 0, 1 jeśli n jest większe od 0	SIGN(-15) Wynik: -1
SQRT(n)	Zwraca pierwiastek kwadratowy liczby n. Jeśli n<0 to wystąpi błąd	SQRT(25) Wynik: 5
TRUNC(m[, n])	Zwraca m obcięte do n miejsc po przecinku. Jeśli n nie jest podane, to przyjmuje się 0. Jeśli n jest ujemne to obcinane są cyfry przed przecinkiem.	TRUNC(15.79, 1) Wynik: 15.7 TRUNC(15.79, -1) Wynik: 10

### Funkcje znakowe

Składnia	Przeznaczenie	Przykład
CHR(n)	Zwraca znak o podanym kodzie	CHR(65) Wynik: "A"
INITCAP(string)	Zwraca string, w którym każde słowo ma dużą pierwszą literę, a pozostałe są małe.	INITCAP('PAN JAN NOWAK') Wynik: "Pan Jan Nowak"
LOWER(string)	Zamienia wszystkie litery w podanym stringu na małe.	LOWER('PAN JAN NOWAK') Wynik: "pan jan nowak"
LPAD(string1, n [, string2])	Zwraca string 1 uzupełniony do długości n lewostronnie ciągami znaków ze stringu 2. Jeśli string2 nie jest podany to przyjmowana jest spacja. Jeśli n jest mniejsze od długości string1, to zwracane jest n pierwszych znaków z tekstu string1.	LPAD('Ala ma ', kota*, 17) Wynik: "kota*kota*Ala ma "
LTRIM(string [, zbiór])	Usuwa litery z tekstu string od lewej strony aż do napotkania litery nie należącej do tekstu zbiór. Jeśli zbiór nie jest podany to przyjmowany jest ciąg pusty.	LTRIM('xxxXxxOstatnie słowo', 'x') Wynik: "XxxOstatnie słowo"
REPLACE(string, search [, replace])	Zwraca string z zamienionym każdym wystąpieniem tekstu search na tekst replace.	REPLACE('Jack & Jue', 'J', 'Bl') Wynik: "Black & Blue"
RPAD(string1, n [, string2])	Zwraca string 1 uzupełniony prawostronnie do długości n ciągami string2. Jeśli string2 nie jest podany, to przyjmuje się spację. Jeśli n jest mniejsze od długości string1, to zwracane jest n pierwszych znaków z tekstu string1.	RPAD('Ala ma ', 17, 'kota*') Wynik: "Ala ma kota*kota*"
RTRIM(string [, zbiór])	Zwraca string1 z usuniętymi ostatnimi literami, które znajdują się w stringu zbiór. Jeśli zbiór nie jest podany to przyjmowany jest ciąg pusty	RTRIM('Ostatnie słowoxXxxx', 'x') Wynik: "Ostatnie słowoxX"
SOUNDEX(string)	Zwraca ciąg znaków reprezentujący wymowę słów wchodzących w skład string1. Funkcja SOUNDEX może być użyta do porównywania słów zapisywanych w różny sposób, ale wymawianych tak samo.	SELECT nazwisko FROM bibl WHERE SOUNDEX(nazwisko) = SOUNDEX('Mickiewicz');
SUBSTR(string, m [, n])	Zwraca podciąg z ciągu znaków string zaczynający się na znaku m i o długości n. Jeśli n nie jest podane, to zwracany jest podciąg od znaku m do ostatniego w string. Pierwszy znak w ciągu ma numer 1.	SUBSTR('ABCDE', 2, 3) Wynik: "BCD"
TRANSLATE( string, from, to)	Zwraca string powstały po zamianie wszystkich znaków from na znak to.	TRANSLATE('HELLO! THERE!', '!', '-') Wynik: "HELLO- THERE-"
UPPER(string)	Zamienia wszystkie znaki z ciągu string na duże litery.	UPPER('Jan Nowak') Wynik: "JAN NOWAK"
ASCII(string)	Zwraca kod ASCII pierwszej litery w podanym ciągu znaków	ASCII('A') Wynik: 65
INSTR(string1, string2 [, n [, m]])	Zwraca pozycję m-tego wystąpienia string2 w string1, jeśli szukanie rozpoczęto od pozycji n. Jeżeli m jest pominięte, to przyjmowana jest wartość 1. Jeśli n jest pominięte, przyjmowana jest wartość 1.	INSTR('MISSISSIPPI', 'S', 5, 2) Wynik: 7
LENGTH(string)	Zwraca długość podanego ciągu znaków.	LENGTH('Nowak') Wynik: 5

### Funkcje konwersji

Funkcje konwersji służą do zamiany wartości jednego typu na wartość innego typu. Ogólnie nazwy funkcji konwersji tworzone są według następującego schematu: *typTOtyp*. Pierwszy typ jest typem, z którego wykonywana jest konwersja, drugi jest typem wynikowym.

Składnia	Przeznaczenie	Przykład
CHARTOROWID (string)	Wykonuje konwersję ciągu znaków na ROWID	SELECT nazwisko FROM pracownicy WHERE ROWID = CHARTOROWID ('0000000F.0003.0002')
CONVERT(string [,dest_char_set [,source_char_set ] ])	Wykonuje konwersję pomiędzy dwoma różnymi implementacjami zestawu znaków. Zestawem domyślnym jest US7ASCII.	SELECT CONVERT ('New WORD', 'US7ASCII', 'WE8HP') "Conversion" FROM DUAL
HEXTORAW (string)	Konwertuje ciąg znaków zawierający cyfry szesnastkowe na wartość binarną, którą można umieścić w polu typu RAW	INSERT INTO GRAPHICS (RAW_COLUMN) SELECT HEXTORAW ('7D') FROM DUAL
ROWTOHEX(raw)	Przekształca wartość typu raw na tekst zawierający cyfry szesnastkowe odpowiadające podanej liczbie.	SELECT RAWTOHEX (RAW_COLUMN) "Graphics" FROM GRAPHICS
ROWIDTOCHAR	Przekształca identyfikator wiersza na tekst. Wynik konwersji ma zawsze długość 18 znaków.	SELECT ROWID FROM GRAPHICS WHERE ROWIDTOCHAR(ROWID) LIKE '%F38%'
TO_CHAR(n [, fmt]) (konwersja numeryczna)	Konwertuje wartość numeryczną na znakową używając opcjonalnego ciągu formatującego. Jeśli ciąg formatujący nie jest podany, to wartość jest konwertowana tak, by zawrzeć wszystkie cyfry znaczące.	SELECT TO_CHAR(17145, '\$099,999') "Char" FROM DUAL
TO_CHAR(d [, fmt]) (konwersja daty)	Konwertuje datę na tekst, używając podanego formatu.	SELECT TO_CHAR(HIREDATE, 'Month DD, YYYY') "New date format" FROM EMP WHERE ENAME = 'SMITH'
TO_DATE(string [, fmt])	Przekształca ciąg znaków w datę. Używa danych aktualnych, jeśli nie mogą być one odczytane z podanego tekstu. Do konwersji używany jest podany ciąg formatujący lub wartość domyślna postaci "DD-MON-YY"	INSERT INTO BONUS (BONUS_DATE) SELECT TO_DATE ('January 15, 1989', 'Month dd, YYYY') FROM DUAL
TO_NUMBER (string)	Przekształca tekst zawierający zapis liczby na liczbę	UPDATE EMP SET SAL = SAL + TO_NUMBER( SUBSTR('\$100 raise', 2, 3)) WHERE ENAME ='BLAKE'

### Funkcje operacji na datach

Składnia	Przeznaczenie	Przykład
ADD_MONTHS (date, n)	Zwraca padaną datę powiększoną o podaną liczbę miesięcy n. Liczba ta może być ujemna	SELECT ADD_MONTHS (HIREDATE, 12) "Next year" FROM EMP WHERE ENAME = 'SMITH'
LAST_DAY(date)	Zwraca datę będącą ostatnim dniem w miesiącu zawartym w podanej dacie.	SELECT LAST_DAY (SYSDATE) "Last" FROM DUAL
MONTHS_BETWEEN (date1, date2)	Zwraca liczbę miesięcy pomiędzy datami date1 i date2. Wynik może być dodatni lub ujemny. Część ułamkowa jest częścią miesiąca zawierającego 31 dni.	SELECT MONTHS_BETWEEN ('02-feb-86', '01-jan-86') "Months"
NEW_TIME(date, a, b)	Zwraca datę i czas w strefie czasowej b, jeśli data i czas w strefie a są równe date. Parametry a i b są wyrażeniami znakowymi i mogą być jednym z: AST, ADT - Atlantic Standard or Daylight Time BST, BDT - Bering Standard or Daylight Time CST, CDT - Central Standard or Daylight Time EST, EDT - Eastern Standard or Daylight Time GMT - Greenwich Mean Time HST, HDT - Alaska-Hawaii Standard or Daylight Time MST, MDT - Mountain Standard or Daylight Time NST - Newfoundland Standard Time PST, PDT - Pacific Standard or Daylight Time YST, YDT - Yukon Standard or Daylight Time	SELECT TO_CHAR( NEW_TIME(TO_DATE('17:47', 'hh24:mi'), 'PST', 'GMT'), 'hh24:mi') "GREENWICH TIME" FROM DUAL
NEXT_DAY(date, string)	Zwraca datę pierwszego dnia tygodnia podanego w string, który jest późniejszy niż data date. Parametr string musi być poprawną nazwą dnia.	SELECT NEXT_DAY('17-MAR-89', 'TUESDAY') "NEXT DAY" FROM DUAL
ROUND(date [, fmt])	Zwraca datę zaokrągloną do jednostki zaokrąglania podanej w fmt. Domyślnie jest to najbliższy dzień.	SELECT ROUND ( TO_DATE('27-OCT-88'), 'YEAR') "FIRST OF THE YEAR" FROM DUAL
SYSDATE	Zwraca aktualny czas i datę. Nie wymaga podania argumentów.	SELECT SYSDATE FROM DUAL
TRUNC(date [, fmt])	Zwraca datę obciętą do jednostki podanej w fmt. Domyślnie jest to dzień, tzn. usuwana jest informacja o czasie.	SELECT TRUNC( TO_DATE('28-OCT-88', 'YEAR') "First Of The Year" FROM DUAL

W funkcjach ROUND i TRUNC można używać następujących tekstów do identyfikacji jednostki zaokrąglenia lub obcięcia:

CC, SCC	wiek
YYYY, YYYY, YEAR, SYEAR, YYY, YY, Y	rok (zaokrąglenie w zwyż od 1.07)
Q	kwartał (zaokrąglenie w górę od 16-tego drugiego miesiąca)

<b>MONTH, MON, MM</b>	miesiąc (zaokrąglenie w górę od 16)
<b>WW</b>	pierwszy tydzień roku
<b>W</b>	pierwszy tydzień miesiąca
<b>DDD, DD, J</b>	dzień
<b>DAY, DY, D</b>	najbliższa niedziela
<b>HH, HH12, HH24</b>	godzina
<b>MI</b>	minuta

#### Inne funkcje

Składnia	Przeznaczenie	Przykład
GREATEST(expr [, expr] ...)	Znajduje największą z listy wartości. Wszystkie wyrażenia począwszy od drugiego są konwertowane do typu pierwszego wyrażenia przed wykonaniem porównania.	SELECT GREATEST ('Harry', 'Harriot', 'Harold') "GREATEST" FROM DUAL
LEAST(expr [, expr] ...)	Zwraca najmniejszą z listy wartości. Wszystkie wyrażenia począwszy od drugiego są konwertowane do typu pierwszego wyrażenia przed wykonaniem porównania.	SELECT LEAST ('Harry', 'Harriot', 'Harold') "LEAST" FROM DUAL
NVL (expr1, expr2)	Jeśli expr1 jest równe NULL, to zwraca expr2, w przeciwnym wypadku zwraca expr1.	SELECT ENAME NVL(TO_CHAR(COMM),'NOT APPLICABLE') "COMMISION" FROM EMP WHERE DEPTNO = 30
UID	Zwraca unikalny identyfikator użytkownika wywołującego funkcję.	SELECT USER, UID FROM DUAL
USER	Zwraca nazwę użytkownika	SELECT USER, UID FROM DUAL

#### Formaty zapisu danych

Formaty zapisu danych używane są w dwóch podstawowych celach:

- zmiany sposobu wyświetlania informacji w kolumnie;
- wprowadzanie danej zapisanej inaczej niż domyślnie.

Formaty zapisu używane są w funkcjach TO\_CHAR i TO\_DATE.

- **Formaty numeryczne**

Formaty numeryczne są używane w połączeniu z funkcją TO\_CHAR do przekształcenia wartości numerycznej na wartość znakową.

Użycie formatu numerycznego powoduje zaokrąglenie do podanej w nim liczby cyfr znaczących.

Jeśli wartość numeryczna ma więcej cyfr z lewej strony niż to zostało przewidziane, to wartość ta zastępowana jest gwiazdką '\*'.

Poniższa tabela przedstawia elementy, które może zawierać specyfikacja formatu numerycznego:

Element	Przykład	Opis
<b>9</b>	9999	Liczba '9' określa szerokość wyświetlania
<b>0</b>	0999	Pokazuje wiodące zera
<b>\$</b>	\$9999	Poprzedza wartość znakiem '\$'
<b>B</b>	B9999	Wyświetla zera jako spacje (nie jako zera)
<b>MI</b>	9999MI	Wyświetla '-' po wartości ujemnej
<b>PR</b>	9999PR	Wyświetla wartość ujemną w nawiasach kątowych '<', '>'
<b>, (przecinek)</b>	9,999	Wyświetla przecinek na podanej pozycji
<b>. (kropka)</b>	99.99	Wyświetla kropkę na podanej pozycji
<b>V</b>	999V99	Mnoży wartość przez 10 <sup>n</sup> , gdzie n jest liczbą dziewiątek po 'V'
<b>E</b>	9.999EEEE	Wyświetla liczbę w notacji wykładniczej (format musi zawierać dokładnie cztery litery E)
<b>DATE</b>	DATE	Dla dat przechowywanych w postaci numerycznej. Wyświetla datę w formacie 'MM/DD/YY'

- **Formaty dat**

Formaty dat są używane w funkcji TO\_CHAR w celu wyświetlenia daty. Mogą być również użyte w funkcji TO\_DATE w celu wprowadzenia daty w określonym formacie. Format standardowy, to 'DD-MON-YY'.

Elementy formatu dat przedstawia tabela:

Element	Opis
<b>SCC lub CC</b>	Wiek; 'S' poprzedza daty przed naszą erą znakiem '-'
<b>YYYY lub SYYYY</b>	Czterocyfrowy rok, 'S' poprzedza daty przed naszą erą znakiem '-'
<b>YYY, YY lub Y</b>	Ostatnie 3, 2 lub 1 cyfra roku
<b>Y,YYY</b>	Rok z przecinkiem na podanej pozycji
<b>SYEAR lub YEAR</b>	Rok przeliterowany. 'S' powoduje poprzedzenie daty przed naszą erą znakiem '-'
<b>BC lub AD</b>	Znak BC/AD (przed naszą erą/naszej ery)
<b>B.C. lub A.D</b>	Znak BC/AD z kropkami
<b>Q</b>	Kwartał roku (1, 2, 3 lub 4)
<b>MM</b>	Miesiąc (01-12)
<b>MONTH</b>	Nazwa miesiąca wyrównana do 9 znaków za pomocą spacji
<b>MON</b>	Trzyliterowy skrót nazwy miesiąca
<b>WW</b>	Tydzień roku (1-52) (tydzień zaczyna się w pierwszy dniu roku i trwa 7 dni)

<b>W</b>	Tydzień miesiąca (1-5) (tydzień zaczyna się w pierwszym dniu miesiąca i trwa 7 dni)
<b>DDD</b>	Dzień roku (1-366)
<b>DD</b>	Dzień miesiąca (1-31)
<b>D</b>	Dzień tygodnia (1-7)
<b>DAY</b>	Nazwa dnia wyrównana do 9 znaków za pomocą spacji
<b>DY</b>	Trzyliterowy skrót nazwy dnia
<b>AM lub PM</b>	Wskaźnik pory dnia
<b>A.M. lub P.M.</b>	Wskaźnik pory dnia z kropkami
<b>HH lub HH12</b>	Godzina (1-12)
<b>HH24</b>	Godzina (1-24)
<b>MI</b>	Minuta (0-59)
<b>SS</b>	Sekunda (0-59)
<b>SSSS</b>	Sekundy po północy (0-86399)
<b>/,.</b>	Znaki przestankowe umieszczane w wyniku
<b>"..."</b>	Ciąg znaków umieszczany w wyniku

Dodatkowo w ciągu znaków określających format można użyć:

- FM - "Fill Mode" przełącznik włączający/wyłączający wypełnianie tekstów spacjami i liczb zerami;
- TH - dodany po kodzie pola powoduje wyświetlanie liczby porządkowej np. 4<sup>TH</sup> dla liczby 4;
- SP - dodany po kodzie pola powoduje, że jest ono literowane
- SPTH lub THSP - połączenie SP i TH.

#### Definicje podstawowe

- Identyfikator (nazwa) - ciąg liter, cyfr i znaków podkreślenia rozpoczynający się literą lub znakiem podkreślenia. Różne systemy baz danych umożliwiają stosowanie innych znaków wewnątrz identyfikatorów (np. znak '\$', lub '!'). Stosowanie tych znaków nie jest jednak zalecane ze względu na późniejsze problemy związane z przenośnością napisanych w ten sposób aplikacji.
- Słowa zarezerwowane - identyfikatory zastrzeżone posiadające specjalne znaczenie w języku SQL. Spis wszystkich słów zarezerwowanych w języku SQL przez twórców ORACLE'a przedstawia tabela:

access	add	all	alter	and	any
as	asc	audit	Between	by	char
check	cluster	column	comment	compress	connect
create	current	Date	dba	decimal	default
delete	desc	distinct	drop	else	exclusive
exists	file	float	for	from	grant
graphic	group	having	identified	if	immediate
in	increment	index	install	initial	insert
integer	intersect	Into	is	level	like
lock	long	maxextents	minus	mode	modify
noaudit	nocompress	Not	nowait	null	number
of	offline	On	online	option	or
order	pctfree	prior	privileges	public	raw
rename	resource	revoke	row	rowid	rownum
rows	select	session	set	share	size
smallint	start	successful	synonym	sysdate	table
then	to	trigger	uid	union	unique
update	user	validate	values	varchar	vargraphic
view	whenever	where	with		

- Liczby - mogą być całkowite lub rzeczywiste. Liczba całkowita nie posiada kropki dziesiętnej. W systemie ORACLE liczby można zapisywać w formacie zwykłym lub wykładniczym. Format wykładniczy składa się z liczby oraz wykładnika liczby 10, przez który należy pomnożyć tę liczbę oddzielonego literą 'e' lub 'E'. Przykłady:

$$7E2 = 7 * 10^2$$

$$25e-03 = 25 * 10^{-3}$$

Dodatkowo w systemie ORACLE liczbę całkowitą można zakończyć literą 'K' lub literą 'M'. Litera 'K' oznacza, że cała liczba ma być pomnożona przez 1024 (1 KB), natomiast litera 'M', że liczbę należy pomnożyć przez 1048576 (1 MB). Przykłady:

$$256K = 256 * 1024$$

$$1M = 1 * 1048576$$

- Rozkazy języka SQL kończą się średnikiem

Zwracana liczba kolumn zależy od parametru rownum

```
select * from emp
```

```
where rownum<=1
```

```
order by 1;
```