

СУБД

# Встроенные функции ORACLE

Лекция 2

# Цифровые функции ORACLE

**Цифровые функции** обеспечивают точность вычислений – 38 десятичных знаков (36 для трансцендентных функций)

## 1. **ABS(n)**

Например: `SELECT ABS(-15) "Absolute" FROM DUAL`

Результат: Absolute

-----

15

## 2. **CEIL(n)** – наименьшее целое, большее или равное n

Например: `SELECT CEIL(15.7) "Ceiling" FROM DUAL`

Результат: Ceiling

-----

16

## 3. **COS (x)** x- в радианах

Например: `SELECT COS( 3.14159265359) "Cos of 180 degrees" FROM DUAL`

Результат: Cos of 180 degrees

-----

-1

# Цифровые функции ORACLE

## 4. **COSH (x)**

Например: SELECT COSH(0) "Hyperbolic cosine of 0" FROM DUAL

Результат: Hyperbolic cosine of 0

-----

1

## 5. **EXP (X)**

Например: SELECT EXP(4) "e to the 4th power" FROM DUAL

Результат: e to the 4th power

-----

54.59815

## 6. **FLOOR (n)**- наименьшее целое, меньше или равно n

Например: SELECT FLOOR(15.7) "Floor" FROM DUAL

Результат: Floor

-----

15

# Цифровые функции ORACLE

## 7. **LN (x)**

Например: `SELECT LN(95) "Natural log of 95" FROM DUAL`

Результат: Natural log of 95

-----  
4.55387689

## 8. **LOG (m,n)** – логарифм по основанию m от n

Например: `SELECT LOG(10,100) "Log base 10 of 100" FROM DUAL`

Результат: Log base 10 of 100

-----  
2

## 9. **MOD (m,n)** – остаток от деления m на n

Например: `SELECT m,n,MOD(m,n), m-n*FLOOR(m/n) "Classical Modulus"FROM test_table`

Результат: M N MOD (M,N) Classical Modulus

-----  
11 4 3

-11 4 -3 1

11 -4 -3 -1

-11 -4 3 -3

# Цифровые функции ORACLE

10. **POWER(m, n)** – m в степени n

Например: SELECT POWER(3,2) "Raised" FROM DUAL

Результат: Raised

-----

9

11. **ROUND(n[,m])** - округление

Например: SELECT ROUND(15.193,1) "Round 1" ROUND(15.193,-1) "Round -1"

FROM DUAL

Результат: Round 1 Round -1

-----

15.2 20

12. **SIGN (n)** – знак n

Например: SELECT SIGN(-15) "Sign" FROM DUAL

Результат: Sign

-----

-1

# Цифровые функции ORACLE

## 13. **SIN(n)**

Например: `SELECT SIN(30 * 3.14159265359/180) "Sin of 30 degrees" FROM DUAL`

Результат: Sin of 30 degrees

-----

.5

## 14. **SINH(n)**

Например: `SELECT SINH(1) "Hyperbolic sine of 1" FROM DUAL`

Результат: Hyperbolic sine of 1

-----

1.17520119

## 15. **SQRT (n)**

Например: `SELECT SQRT(26) "Square root" FROM DUAL`

Результат: Square root

-----

5.09901951

# Цифровые функции ORACLE

## 16. **TAN(n)**

Например: `SELECT TAN(135 * 3.14159265359/180) "Tangent of 135 degrees"`  
`FROM DUAL`

Результат: Tangent of 135 degrees

-----

-1

## 17. **TANH(n)**

Например: `SELECT TANH(.5) "Hyperbolic tangent of .5" FROM DUAL`

Результат: Hyperbolic tangent of .5

-----

.462117157

## 18. **TRUNC(n[,m])** –усечение

Например: `SELECT TRUNC(15.79,1) "Truncate" FROM DUAL`

Результат: Truncate

-----

15.7

# Символьные функции ORACLE

**Символьные функции** – возвращают тип VARCHAR2, ограниченный 2000 байтами или CHAR(255 байт). Для конкатенации строк используется ||

## 1. CHR(n)

Например: SELECT CHR(67)||CHR(65)||CHR(84) "Dog" FROM DUAL

Результат: Dog

-----

CAT

2. **CONCAT(char1, char2)** – конкатенация, эквивалент операции ||

Например: SELECT CONCAT( CONCAT(ename, ' is a '), job) "Job" FROM emp  
WHERE empno = 7900

Результат: Job

-----

JAMES is a CLERK

## Символьные функции ORACLE

3. **INITCAP(char)** – первый символ каждого слова делает прописной буквой, а остальные строчными

Например: `SELECT INITCAP('the soap') "Capitals" FROM DUAL`

Результат: Capitals

-----

The Soap

4. **LOWER(char)** – заменяет все буквы на строчные

Например: `SELECT LOWER('MR. SAMUEL') "Lowercase" FROM DUAL`

Результат: Lowercase

-----

mr. samuel

5. **LPAD(char1,n [,char2])** – дополняет строку char1 слева пробелами или последовательностью символов char2 до длины n.

Например: `SELECT LPAD('Page 1',15,'* .') "LPAD пример:"`

`FROM DUAL`

Результат: LPAD пример:

-----

\*.\*.\*.\*.\*Page 1

# Символьные функции ORACLE

6. **LTRIM (char1 [,char2])** – удаляет из строки ведущие пробелы или символы, заданные в char2.

Например: SELECT LTRIM(' LAST WORD') "LTRIM пример:" FROM DUAL

Результат: LTRIM пример:

-----

LAST WORD – удалены все пробелы слева

7. **NLS\_INITCAP** – аналогично INITCAP, но может учитывать лингвистические особенности для разных языков.

Например: SELECT NLS\_INITCAP('ijsland', 'NLS\_SORT = XDutch') "Capital" FROM DUAL

Результат: Capital

-----

IJsland

8. **NLS\_LOWER (char [, 'nlsparams'] )** – то же, что LOWER, но с учетом национальных лингвистических особенностей

Например: SELECT NLS\_LOWER('CITTA"', 'NLS\_SORT = XGerman') "Lower" FROM DUAL

Результат: Lower

-----

citta

# Символьные функции ORACLE

9. **NLS\_UPPER (char [, 'nlsparams'] )** - то же, что UPPER, но с учетом национальных лингвистических особенностей

Например: SELECT NLS\_UPPER('gro?e', 'NLS\_SORT = Xgerman') "Upper" FROM DUAL

Результат: Upper

-----

GROSS

10. **REPLACE (char, search\_string[,replacement\_string])** – замена в строке char одного фрагмента на другой или удаление заданного фрагмента

Например: SELECT REPLACE('JACK and JUE','J','BL') "Changes" FROM DUAL

Результат: Changes

-----

BLACK and BLUE

11. **RPAD (char1, n [,char2])** – дополнение строки char1 пробелами или заданными в char2 символами справа до длины n.

Например: SELECT RPAD(ename,12,'ab') "RPAD пример:" FROM emp

WHERE ename = 'TURNER'

Результат : RPAD пример:

-----

TURNERababab

## Символьные функции ORACLE

12. **RTRIM(char [,set])** – удаляет из строки завершающие пробелы или

Например: SELECT RTRIM('TURNERyxXxy','xy') "RTRIM пример" FROM DUAL

Результат: RTRIM пример

-----

TURNERyxX

13. **SOUNDEX(char)** – заменяет символы в строке в соответствии с фонетикой английского языка

Например: SELECT ename FROM emp

WHERE SOUNDEX(ename) = SOUNDEX('SMYTHE')

Результат : ENAME

-----

SMITH

## Символьные функции ORACLE

14. **SUBSTR (char, m [,n])** – выделение фрагмента строки char длиной n, начиная с позиции m. Если n опущено, то выделяются символы до конца строки, начиная с позиции m.

Например: `SELECT SUBSTR('ABCDEFGF',3,4) "Subs" FROM DUAL`

Результат: Subs

----

CDEF

Тот же результат можно получить с помощью запроса:

`SELECT SUBSTR('ABCDEFGF',-5,4) "Subs" FROM DUAL`

15. **TRANSLATE (char, from, to)** – преобразует каждый символ строки char в соответствии с функцией преобразования, заданной при помощи строк from и to.

Например: `SELECT TRANSLATE('2KRW229',  
'0123456789ABCDEFGHIJKLMNORSTUVWXYZ',  
'9999999999XXXXXXXXXXXXXXXXXXXXXXXXXX') "Licence"`

`FROM DUAL`

Результат: Licence

-----

9XX999

## Символьные функции ORACLE

16. **UPPER(char)** – преобразует все буквы в заглавные

Например:

```
SELECT UPPER('Large') "Uppercase" FROM DUAL
```

Результат:

Uppercase

-----

LARGE

# Символьные функции, возвращающие цифровое значение

1. **ASCII(char)** – код первого символа строки char  
Например: SELECT ASCII('Q') FROM DUAL

Результат: ASCII('Q')

-----

81

2. **INSTR(char1,char2[,n[,m]])** – позиция n-ного символа в m-ном вхождении фрагмента char2 в строку char1. Если n и m опущено, то они принимаются равными 1. Если фрагмент не входит в строку, то функция равна 0.

Например: s SELECT INSTR('CORPORATE FLOOR','OR', 3, 2) "Instring" FROM DUAL

Результат : Instring

-----

14

3. **LENGTH(char)** – длина строки

4. **TO\_NUMBER(char [,fmt [, 'nlsparams'] ])**- преобразование строки в числовой тип

Например: UPDATE emp

SET sal = sal + TO\_NUMBER('100.00', '9G999D99')

WHERE ename = 'BLAKE'

# Дата и время в Oracle

## Типы данных

В Oracle имеется единый тип данных **DATE**, хранящей для каждой величины, представленной в этом типе, значение года, месяца, дня, часов, минут и секунд. Дата и время также могут храниться в форматах **CHAR** и **NUMBER**, допускающих преобразование к типу DATE с помощью функции **TO\_DATE**.

Функция без параметров **SYSDATE** возвращает текущее значение даты и времени.

## Константы даты/времени и формат представления даты

Внешнее представление даты и времени - символьная константа, преобразование которой во внутреннее представление определяется форматом, заданным в **NLS\_DATE\_FORMAT**.

**Спецификаторы NLS** (National Language Support) - поддержка национальных языков - позволяет пользователю работать в среде с локализованными средами окружения. Параметры NLS определяют многие значения по умолчанию. От спецификаторов NLS, определяющих соглашения на среду окружения, зависят принимаемые по умолчанию аргументы ряда функций Oracle SQL, в том числе - функций **TO\_CHAR**, **TO\_DATE**, **TO\_NUMBER**.

# Дата и время в Oracle

В имеющийся набор спецификаторов NLS входят, например:

- ***NLS\_TERRITORY*** - название территории (America, France, Japan, Russian,...)
- ***NLS\_CURRENCY*** - название национальной валюты
- ***NLS\_CALENDAR*** - название календаря, по умолчанию используется Григорианский календарь.
- ***NLS\_DATE\_LANGUAGE*** - написание дней недели и месяцев
- ***NLS\_DATE\_FORMAT*** - формат даты по умолчанию (значение данного спецификатора по умолчанию определяется NLS\_TERRITORY).

# Дата и время в Oracle

Значение спецификатора NLS\_DATE\_FORMAT (как и всех других NLS-спецификаторов может задаваться в конфигурационном файле или в операторе ALTER SESSION:

**ALTER SESSION SET NLS\_DATE\_FORMAT** = <маска представления даты>

<маска представления даты> представляет собой строковую константу и некоторые спецификаций преобразования даты в строку или наоборот, которые она может содержать следующие:

- Y, YY, YYY, YYYY - последние 1- 4 цифры года
- MM - номер месяца
- MON - 3-буквенная аббревиатура названия месяца
- MONTH - полное название месяца
- D - номер дня недели
- DAY - название дня недели
- DY - сокращенное название дня недели
- DD - номер дня в месяце
- DDD - номер дня в году
- HH, HH12 - часы (1 - 12)
- HH24 - часы (1 - 24)
- Возможные разделители между спецификациями: - / , . ; :
- Для добавления строковых констант к дате, строковая константа заключается в двойные кавычки, а все выражение - в одиночные кавычки,

# Дата и время в Oracle

Пример:

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = "'Today is' DD/MM/YYYY';
```

Session altered.

```
SQL> SELECT SYSDATE FROM dual;
```

```
SYSDATE
```

```
-----
```

```
Today is 10/09/2001
```

# Преобразование типов

**Преобразование типа DATE в тип CHAR** выполняется функцией **TO\_CHAR**. Эта функция символьное представление даты и времени и может иметь от одного до трех параметров:

- первый (обязательный) параметр - преобразуемое значение типа DATE;
- второй (необязательный) параметр - маска представления даты, построенная так, как описано в предыдущем разделе; если этот параметр не задан, то принимается значение маски по умолчанию, установленное в спецификаторе NLS\_DATE\_FORMAT;
- третий (необязательный) параметр - значение других спецификаторов NLS, если третий параметр не задан, то принимаются их значения по умолчанию

**TO\_CHAR(d [, fmt [, 'nlsparams' ] ])** – преобразование даты или числа d в VARCHAR2

Например: `SELECT TO_CHAR(HIREDATE, 'Month DD, YYYY') "New date format"`

`FROM emp WHERE ename = 'SMITH'`

Результат: New date format

December 17, 2010

Например: `SELECT TO_CHAR(-10000, 'L99G999D99MI') "Amount" FROM DUAL`

Результат: Amount

\$10,000.00-

# Преобразование типов

**Преобразование символьной строки к типу DATE** выполняется функцией **TO\_DATE**. Эта функция возвращает значение даты и времени и может иметь от одного до трех параметров:

- первый (обязательный) параметр - значение типа CHAR или VARCHAR2 - строковое представление даты;
- второй (необязательный) параметр - маска представления даты, построенная так, как описано в предыдущем разделе; если этот параметр не задан, то принимается значение маски по умолчанию, установленное в спецификаторе NLS\_DATE\_FORMAT;
- третий (необязательный) параметр - значение других спецификаторов NLS, если третий параметр не задан, то принимаются их значения по умолчанию

**TO\_DATE (char [, fmt [, 'nlsparams' ]])** – преобразование строки в дату

Например: INSERT INTO bonus (bonus\_date)

```
SELECT TO_DATE('January 15, 1989, 11:00 A.M.', 'Month dd, YYYY, HH:MI A.M.',  
'NLS_DATE_LANGUAGE = American') FROM DUAL
```

# Элементы форматирования масок дат

Маска	Значение
SCC или CC	Век; префикс 'S' задает выдачу дат до н.э. со знаком '-'
YYYY или SYYYYY	Год; префикс 'S' задает выдачу дат до н.э. со знаком '-'
YYY или YY или Y	Последние 3, 2 или 1 разряд года
Y,YYY	Год с запятой после третьего разряда
SYEAR или YEAR	Год, задаваемый словами на языке нац. поддержки. Префикс 'S' задает '-' для дат до н.э.
BC или AD	BC/AD индикатор эры (может быть задан любой из двух, что не влияет на результат)
B.C. или A.D.	BC/AD индикатор с точками
Q	Квартал года (число от 1 до 4)
MM	Месяц по порядку от 1 до 12
MONTH	Название месяца, дополняемое пробелами до максимального размера из домена значений
MON	Трехсимвольная аббревиатура названия месяца
WW или W	Неделя года или месяца
DDD или DD или D	Порядковый день года, месяца или недели
DAY	Название дня недели, дополняемое до максимального размера из домена значений
DY	Двухсимвольная аббревиатура названия дня недели
J	День по юлианскому календарю, количество дней от 31 декабря 4713 г. до н.э.
AM или PM	Индикатор времени до или после полудня (может быть задана любая из двух масок)
A.M. или P.M.	Индикатор времени до или после полудня с точками
HH или HH12	Часы от 1 до 12
HH24	Часы от 0 до 23
MI	Минуты
SS	Секунды

# Арифметика даты и времени

Данные типа DATE могут быть операндами операций увеличения, уменьшения и вычитания.

В операциях увеличения и уменьшения тип первого операнда - DATE, тип второго операнда - NUMBER. Тип результата совпадает с типом первого операнда.

В операции вычитания оба операнда должны иметь тип DATE, результат имеет тип NUMBER.

Число типа NUMBER в арифметике даты и времени интерпретируется как количество дней. Дробная часть числа - дробные части дня, т.е., часы, минуты, секунды. Некоторые функции Oracle обеспечивают дополнительную поддержку арифметики даты и времени:

**ADD\_MONTHS(d,n)** - возвращает дату d плюс n месяцев (n может быть только целым)

**MONTHS\_BETWEEN(d1,d2)** - возвращает, количество месяцев между датами d1 и d2 (возможно, с дробной частью)

**LAST\_DAY(d)** - возвращает дату последнего дня того же месяца, к которому относится дата d

**NEXT\_DAY(d,c)** - возвращает дату, соответствующую следующему появлению дня недели, заданного текстовым значением c, после даты d

**ROUND(d,fmt)** - возвращает дату, округленную до формата, заданного вторым (необязательным) параметром

**TRUNC(d,fmt)** - возвращает дату, усеченную до формата, заданного вторым (необязательным) параметром

# Функции для работы с датами

1. **ADD\_MONTHS (d,n)** – прибавляет к дате d n месяцев.

Например: SELECT TO\_CHAR(ADD\_MONTHS(hiredate,1),'DD-MON-YYYY') "Next month"

FROM emp

WHERE ename = 'SMITH'

Результат : Next Month

-----  
17-JAN-1981

2. **LAST\_DAY (d)** – дата последнего дня месяца d

Например: SELECT SYSDATE, LAST\_DAY(SYSDATE) "Last",

LAST\_DAY(SYSDATE) - SYSDATE "Days Left" FROM DUAL

Результат: SYSDATE Last Days Left

-----  
10-APR-95 30-APR-95 20

3. **MONTHS\_BETWEEN (d1, d2)** – число месяцев между датами d1 и d2

Например: SELECT MONTHS\_BETWEEN(TO\_DATE('02-02-1995','MM-DD-YYYY'),  
TO\_DATE('01-01-1995','MM-DD-YYYY')) "Months" FROM DUAL

Результат: Months

-----  
1.03225806

# Функции для работы с датами

4. **NEW\_TIME(d, z1, z2)** – дата и время d, соответствующие временному поясу z1, преобразуются к временному поясу z2. Обозначения временных поясов:

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 Time 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

5. **NEXT\_DAY(d, char)** – дата первого дня недели char, следующего за датой d.

Например: `SELECT NEXT_DAY('15-MAR-92','TUESDAY') "NEXT DAY"`  
`FROM DUAL`

Результат: NEXT DAY

-----

17-MAR-92 – первый вторник после 15 марта

# Функции для работы с датами

6. **ROUND (d[,fmt])** – округление даты в соответствии с заданным форматом fmt

Например: `SELECT ROUND(TO_DATE('27-OCT-92'),'YEAR')`

`"FIRST OF THE YEAR" FROM DUAL`

Результат : FIRST OF THE YEAR

-----

01-JAN-93

7. **SYSDATE** – текущая дата

Например: `SELECT TO_CHAR(SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "NOW"`

`FROM DUAL`

Результат : NOW

-----

10-29-1993 20:27:11.

# Функции для работы с датами

8. **TRUNC(d,[fmt])** – усечение даты

Например: SELECT TRUNC(TO\_DATE('27-OCT-92', 'DD-MON-YY'), 'YEAR')

"First Of The Year" FROM DUAL

Результат : FIRST OF THE YEAR

-----

01-JAN-92

Ниже приводятся форматы для округления или усечения даты

Format Model Rounding or Truncating Unit

CC, SCC Century

SYYYY, YYYYY, YEAR, SYEAR, YYY, YY, Y Year (rounds up on July 1)

IYYYY, IY, IY, I ISO Year

Q Quarter (rounds up on the sixteenth day of the second month of the quarter)

MONTH, MON, MM, RM Month (rounds up on the sixteenth day)

WW Same day of the week as the first day of the year.

IW Same day of the week as the first day of the ISO year.

W Same day of the week as the first day of the month.

DDD, DD, J Day

DAY, DY, D Starting day of the week

HH, HH12, HH24 Hour

MI Minute

# Другие функции

1. **GREATEST (expr [,expr] ...)** – наибольшее из списка

Например: SELECT GREATEST('HARRY','HARRIOT','HAROLD') "GREATEST"  
FROM DUAL

Результат: GREATEST

-----

HARRY

2. **LEAST (expr [,expr] ...)** – наименьшее из списка

Например: SELECT LEAST('HARRY','HARRIOT','HAROLD') "LEAST"  
FROM DUAL

Результат: LEAST

-----

HAROLD

# Другие функции

3. **NVL(expr1, expr2)**- возвращает expr1, если оно не пустое, и expr2, если пустое

Например: SELECT ename, NVL(TO\_CHAR(COMM),'NOT APPLICABLE') "COMMISSION"  
FROM emp WHERE deptno = 30

Результат: ENAME COMMISSION

-----

ALLEN 300

WARD 500

MARTIN 1400

BLAKE NOT APPLICABLE

4. **UID** – возвращает идентификационный номер текущего пользователя

Например: SELECT USER, UID FROM DUAL

USER UID

-----

OPS\$BQUIGLEY 46

5. **USERENV (option)** – данные о текущей сессии