Встроенные функции

Замечание

Для выполнения примеров по теме «Встроенные функции» будем использовать таблицу DUAL. Это вспомогательная таблица Oracle. Она состоит ровно из одной колонки с именем DUMMY и одной записи - “X” (попробуйте выполнить “select * from dual”). Владелец этой таблицы – SYS (администратор базы данных), но доступ до нее имеет любой пользователь базы данных. При выполнении select над таблицей dual с использованием констант, мы гарантированно получим ровно одну запись.

Функции для работы со строками в Oracle

Для упрощения работы со строками имеется ряд встроенных функций, что значительно облегчает такие операции как преобразование строк к данным других типов, поиск подстроки в строке, определение длины строки и т. д. Рассмотрим самые распространенные функции для работы со строками:

1) Функция определения длины строки LENGTH(строка), возвращает количество символов в строке, включая концевые пробелы.
   select length('string ') from dual
   вернет значение 7.

2) Функции преобразования регистров символов.

UPPER(строка) - преобразование к верхнему регистру
SELECT UPPER(‘string’) FROM DUAL вернет STRING.

LOWER(строка) - преобразование к нижнему регистру
SELECT LOWER(‘STrinG’) FROM DUAL вернет string

INITCAP(строка) - преобразовывает каждый первый символ слова к верхнему регистру, а все остальные символы к нижнему при условии, что символ-разделитель между словами пробел
SELECT INITCAP(‘string1 string2’) FROM DUAL
вернет строку String1 String2

3) Функции для обрезания начальных и концевых пробелов: LTRIM(строка), RTRIM(строка), TRIM(строка). Соответственно первая функция обрезает все начальные пробелы строки, вторая – все концевые, а третья все начальные и концевые.
SELECT LTRIM(‘ str1’) FROM DUAL вернет строку str1,
SELECT RTRIM(‘str2 ’) FROM DUAL вернет строку str2,
SELECT TRIM(‘ str3 ’) FROM DUAL вернет строку str3.

4) Функция замены части строки другой строкой REPLACE(исходная_строка, заменяемая_подстрока, заменяющая_подстрока). Для большей ясности рассмотрим пример, в некотором текстовом поле таблицы хранится число. Причем символ-разделитель между целой и дробной частью в некоторых полях «.», а нам для дальнейшей обработки данных нужно, чтобы он во всех полях должен быть «,». Для этого воспользуемся функцией REPLACE следующим образом. REPLACE(field1, ’.’, ’,’) и все символы «.» в поле field будут заменены на символ «,». SELECT REPLACE(‘My_string’,’_’,’@’) FROM DUAL вернет строку My@string.

5) Функции преобразования данных к другим типам данных.

TO_CHAR(число) преобразует число в текст.
SELECT TO_CHAR(123) FROM DUAL вернет строку 123,

TO_NUMBER(строка) преобразует текст в число.
SELECT TO_NUMBER('12345') FROM DUAL
вернет число 12345,

TO_DATE(строка, формат_даты) преобразует строку в дату определенного формата.
SELECT TO_DATE('03 JAN 2012','dd.mm.yyyy') FROM DUAL вернет дату 01.03.2012.

6) Функция определения вхождения подстроки в строку INSTR(исходная_строка, подстрока, номер_символа). Даная функция позволяет определять номер символа в исходной строке с которого начинается искомая подстрока, иначе возвращается 0. Например, нам нужно определить все должности в таблице Table1, в наименовании которых встречается подстрока «менеджер». Для этого вполне подойдет следующий оператор

```
select * from table1 where instr(post,'менеджер',1)>0;
```

Запрос выведет только те записи из таблицы TABLE1, где искомая подстрока «менеджер» будет найдена. Причем поиск будет осуществляться с первого символа. Если поиск нужно осуществлять с другой позиции, то номер символа для начала поиска указывается в третьем параметре.

7) Функция выделения в исходной строке подстроки SUBSTR(исходная_строка, номер_начального_символа, количество_символов).
SELECT SUBSTR(‘My_string’, 4, 3) FROM DUAL вернет строку str

8) Для определения кода символа используется функция ASCII(строка), которая возвращает код 1 символа строки. Например:
SELECT ASCII(‘W’) FROM DUAL вернет значение 87.

9) Обратная функция преобразования кода символа в символ CHR(число).
SELECT CHR(87) FROM DUAL вернет символ W.

Функции для работы с числами в Oracle

В СУБД Oracle имеется ряд функций для работы с числами. К ним относятся функции возведение числа в степень POWER(), округление ROUND() и т. д.

1) Функция ABS(число) возвращает абсолютное значение аргумента.

SELECT ABS(-3) FROM DUAL вернет значение 3.

2) Функция CEIL(число) возвращает наименьшее целое, большее или равное переданному параметру.

SELECT CEIL(4.5) FROM DUAL вернет значение 5.

3) Функция FLOOR(число) возвращает наибольшее целое, меньшее или равное переданному параметру.

SELECT FLOOR(3.8) FROM DUAL вернет значение 3.

4) MOD(число_1, число_2) - остаток от деления первого параметра на второй.
SELECT MOD(5, 3) FROM DUAL вернет значение 2.
Примечание. Если второй параметр равен 0, то функция возвращает первый параметр.
5) ROUND(число_1, число_2) - округляет первый переданный параметр до количества разрядов, переданного во втором параметре. Если второй параметр не указан, то округление производится до целого значения. Примеры
SELECT ROUND(101.34) FROM DUAL вернет значение 101,
SELECT ROUND(100.1268, 2) FROM DUAL
вернет значение 100.13
SELECT ROUND(1234000.3254, -2) FROM DUAL
вернет значение 1234000,
SELECT ROUND(-100.122, 2) FROM DUAL
вернет значение -100.12.

6) Функция усечения значения TRUNC(число_1, число_2). Возвращает усеченное значение первого параметра до количества десятичных разрядов, указанного во втором параметре.

Примеры:

SELECT TRUNC(150.58) FROM DUAL

вернет значение 150

SELECT TRUNC(235.4587, 2) FROM DUAL

вернет значение 235.45

SELECT TRUNC(101.23, -1) FROM DUAL

вернет значение 100

7) SIN(число), COS(число), TAN(число), ACOS(число), ASIN(число), ATAN(число) - возвращают значение соответствующей названию тригонометрической функции. Для прямых функции параметром является значение угла в радианах, а для обратных – значение функции. Примеры

SELECT COS(0.5) FROM DUAL

вернет значение 0.877582561890373

SELECT SIN(0.5) FROM DUAL

вернет значение 0.479425538604203

SELECT TAN(0.5) FROM DUAL

вернет значение 0.546302489843791

SELECT ACOS(0.5) FROM DUAL

вернет значение 1.0471975511966

SELECT ASIN(0.5) FROM DUAL

вернет значение 0.523598775598299

SELECT ATAN(0.5) FROM DUAL

вернет значение 0.463647609000806

8) Гиперболические функции. SINH(число), COSH(число), TANH(число). SINH() возвращает гиперболический синус переданного параметра, COSH() возвращает гиперболический косинус переданного параметра, TANH() возвращает гиперболический тангенс переданного параметра.

Примеры:

SELECT COSH(0.5) FROM DUAL

вернет значение 1.12762596520638

SELECT SINH(0.5) FROM DUAL

вернет значение 0.521095305493747

SELECT TANH(0.5) FROM DUAL

вернет значение 0.46211715726001

9) Функция возведения в степень POWER(число_1, число_2). Примеры:
SELECT POWER(10, 2) FROM DUAL вернет значение 100
SELECT POWER(100, -2) FROM DUAL
вернет значение 0.0001

10) Функция извлечения квадратного корня SQRT(число). Пример:
SELECT SQRT(4) FROM DUAL вернет значение 2.

11) Функция возведение числа e в степень EXP(число). Пример:
SELECT EXP(2) FROM DUAL вернет значение 7.38905609893065.

12) Логарифмические функции. LN(число) возвращает натуральный логарифм переданного параметра, LOG(число_1, число_2) возвращает логарифм второго переданного параметра по основанию, переданному первом параметре. Причем первый параметр должен быть больше нуля и не равен 1.

Примеры:

SELECT LN(5) FROM DUAL

вернет значение 1.6094379124341

SELECT LOG(10, 3) FROM DUAL

вернет значение 0.477121254719662

Функции для работы с датами в Oracle

1) ADD_MONTHS(дата, количество_месяцев) возвращает дату, отстоящую от даты, переданной в первом параметре на количество месяцев, указанном во втором параметре.
Примеры:
SELECT ADD_MONTHS('01-JAN-2010', 2) FROM DUAL вернет дату '01.03.2010'
SELECT ADD_MONTHS('01-JAN-2010', -3) FROM DUAL вернет дату '01.10.2009'
SELECT ADD_MONTHS('30-JAN-2010', 1) FROM DUAL вернет дату '28.02.2010'

2) Для определения текущей даты и времени применяется функция SYSDATE. Область применения данной функции намного шире чем может показаться на первый взгляд. В первую очередь это контроль за вводом данных в БД. Во многих таблицах выделяется отдельное поля для сохранения даты последнего внесения изменений. Также очень удобно контролировать некие входные параметры для отчетов, особенно если они не должны быть больше чем текущая дата. Помимо даты данная функция возвращает еще и время с точностью до секунд.
Пример:
SELECT SYSDATE FROM DUAL вернет дату ‘22.05.2010 14:51:20’

3) Если необходимо определить последний день месяца, то для этого вполне подойдет функции LAST_DAY(дата).
Пример:
SELECT LAST_DAY(’15-FEB-2010’) FROM DUAL
вернет дату ’28.02.2010’.
SELECT LAST_DAY(SYSDATE) FROM DUAL
вернет последний день текущего месяца.

4) Функция для определения количества месяцев между датами MONTHS_BETWEEN(дата_1, дата_2).
Примеры:
SELECT MONTHS_BETWEEN(’01-JUL-2009’, ’01-JAN-2010’) FROM DUAL вернет значение -6
SELECT MONTHS_BETWEEN(’01-JUL-2009’, ’10-JAN-2010’) FROM DUAL вернет значение -6.29032258064516.
Примечание.
Если дни месяцев совпадают, то функция возвращает целое число, в противном случае результат будет дробным, причем количество дней в месяце будет принято 31.

5) Функция NEXT_DAY(дата, день_недели) позволяет определить следующую дату от даты, переданной в первом параметре, которая соответствует дню недели, переданном во втором параметре.

Пример:

SELECT NEXT_DAY('07.01.2009', 'mon') FROM DUAL

вернет дату '07.06.2009', то есть следующий понедельник после 1 июля 2009 наступил 6 числа.

mon – Понедельник

tue – Вторник

wed – Среда

thu – Четверг

fri – Пятница

sat – Суббота

sun – воскресенье

6) Округление даты ROUND(дата, формат). Второй параметр не обязателен, если его не указывать, то он принимается за ‘DD’, то есть округление будет произведено до ближайшего дня. Примеры:
SELECT ROUND(SYSDATE) FROM DUAL вернет следующий за текущим день
SELECT ROUND(SYSDATE, ‘MONTH’) FROM DUAL округляется до ближайшего первого дня месяца.
CC, SCC – Век; SYYYY, YYYY, YEAR – Год; Q – Квартал MM, MONTH – Месяц; WW – Тот же день недели, что и первый день года; W – Тот же день недели, что и первый день месяца; DD, J – День; Day, DY – Первый день недели; HH, HH12, HH24 – Час; MI – Минута.

7) Усечение даты. Функция TRUNC(дата, формат). Также как и рассмотренная выше может не иметь второго параметра. В таком случае усечение будет производиться до ближайшего дня.

Примеры:

SELECT TRUNC(SYSDATE) FROM DUAL

вернет дату ’22.05.2010’

SELECT TRUNC(SYSDATE, ‘WW’) FROM DUAL

вернет дату ’01.05.2010’

SELECT TRUNC(SYSDATE, ‘Day’) FROM DUAL

вернет дату ‘16.05.2010’.

Функции преобразования данных в Oracle

На практике довольно распространены ситуации, когда необходимо строковые величины рассматривать как числа и наоборот. Несмотря на небольшое количество функции их возможностей вполне хватает для решения очень сложных прикладных задач.

1) TO_CHAR(данные, формат). На первый взгляд синтаксис довольно прост, но за счет второго параметра можно очень точно описать в какой формат преобразовать данные. Итак в строку можно преобразовать как дату, так и числовое значение.

D – День недели; DD – День месяца; DDD – День года;
MM – Номер месяца; MON – Сокращенное название месяца; MONTH – Полное название месяца; Q – Квартал;
YY, YYY, YYYY – Год; HH, HH12, HH24 – Час;
MI – Минута; SS – Секунда.

Примеры:

SELECT TO_CHAR(SYSDATE, ‘D-MONTH-YY’) FROM DUAL вернет строку ‘7-MAY -10’

SELECT TO_CHAR(SYSDATE, ‘DDD-MM-YYYY’) FROM DUAL вернет строку ‘142-05-2010’

SELECT TO_CHAR(SYSDATE, ‘Q-D-MM-YYY’) FROM DUAL вернет строку ‘2-7-05-010’

99D9 – число девяток соответствует максимальному количеству цифр; 999G99 – Указание позиции группового разделителя; 99,999 – Возвращает запятую в указанной позиции; 99.999 – Возвращает точку в указанной позиции; 99V9999 – Возвращает значение умноженное на 10 в степени n, где n число девяток после V; 0999 – Возвращает ведущие нули, а не пробелы; 9990 – Возвращает конечные нули, а не пробелы; 9.99EEEE – Возвращает число в экспоненциальной форме; RM – Возвращает число в римской системе исчисления

Примеры

SELECT TO_CHAR(1050, ‘9.99EEEE’) FROM DUAL

вернет строку ‘ 1.050E+03’

SELECT TO_CHAR(1400, ‘9999V999’) FROM DUAL

вернет строку ‘1400000’

SELECT TO_CHAR(48, ‘RM’) FROM DUAL

вернет строку ‘ XLVIII’

2) Функция преобразования строки в дату TO_DATE(строка, формат). Возможные значения форматов уже рассмотрены выше, поэтому приведу несколько примеров использования данной функции.
Примеры:
SELECT TO_DATE(’01.01.2010’, ‘DD.MM.YYYY’) FROM DUAL вернет дату ‘01.01.2010’
SELECT TO_DATE(’01.JAN.2010’, ‘DD.MON.YYYY’) FROM DUAL вернет дату ‘01.01.2009’
SELECT TO_DATE(’15-01-10’, ‘DD-MM-YY’) FROM DUAL вернет дату ‘15.01.2010’.

3) Функция преобразования строки в числовое значение TO_NUMBER(строка, формат). Самые распространенные значения форматов перечислены в таблице, поэтому рассмотрим применение данной функции на примерах.
Примеры:
SELECT TO_NUMBER(‘100’) FROM DUAL
вернет число 100
SELECT TO_NUMBER(‘0010.01’, ’9999D99’) FROM DUAL
вернет число 10.01
SELECT TO_NUMBER('500,000','999G999') FROM DUAL
вернет число 500000.