ORACLE 12c

PL/SQL – встроенные функции

Лекция 11

Встроенные функции

- Числовые функции
- Символьные функции
- Функции по работе с датами
- Конвертирование
- Функции обработки ошибок

Числовые функции

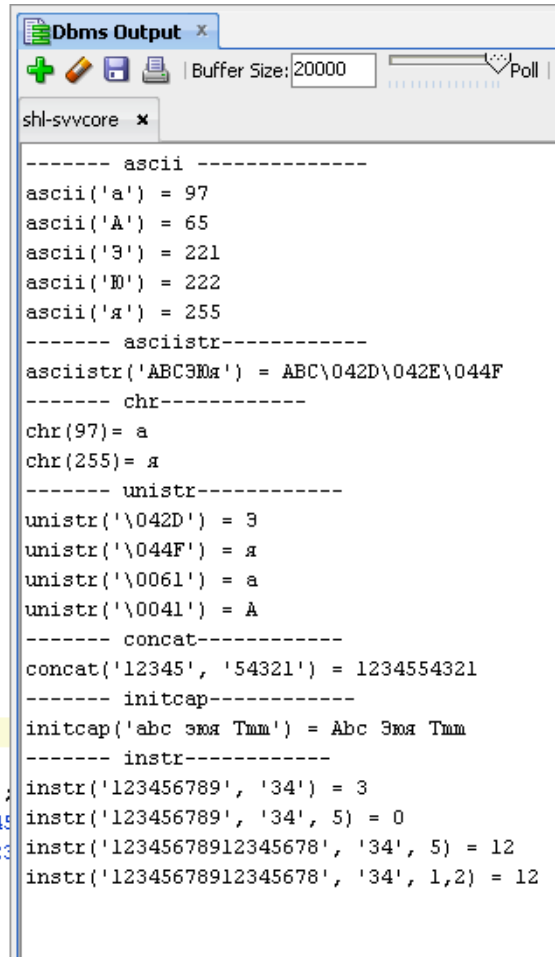
```
begin
  dbms_output.put_line('abs(-10.493) = '||abs(-10.493) );
  dbms_output.put_line('ceil(-10.493) = '||ceil(-10.493) );
  dbms_output.put_line('round(-10.493) = '||round(-10.493));
  dbms_output.put_line('round(-10.493,1) = '||round(-10.493,1));
  dbms_output.put_line('trunc(-10.493,1) = '||trunc(-10.493,1));
  dbms_output.put_line('floor(-10.493) = '||floor(-10.493));
  dbms_output.put_line('floor(10.493) = '||floor(10.493));
  dbms_output.put_line('remainder(10,3) = '||remainder(10,3));  -- 10%3
  dbms_output.put_line('remainder(-10,3) = '||remainder(-10,3));  -- 10%3
  dbms_output.put_line('mod(-10,3) = '||mod(-10,3));
  dbms_output.put_line('bitand(0,1) = '||to_number(bitand(0,1)));
  dbms_output.put_line('bitand(15,7) = '||to_number(bitand(15,7)));
  dbms_output.put_line('bitand(5,3) = '||to_number(bitand(5,3)));
  -- dbms_output.put_line('width_bucket(21, 0,100,10) = '||WIDTH_BUCKET (21,
  dbms_output.put_line('cos(3.14/180*60) = '||cos(3.14/180*60));
  dbms_output.put_line('acos(0.5) = '||acos(0.5)/3.14*180);
  dbms_output.put_line('sin(3.14/180*60) = '||sin(3.14/180*60));
  dbms_output.put_line('asin(0.5) = '||asin(0.5)/3.14*180);
  dbms_output.put_line('tan(3.14/180*60) = '||tan(3.14/180*60));
  dbms_output.put_line('atan(0.5) = '||atan(0.5)/3.14*180);
  dbms_output.put_line('exp(1) = '||exp(1));
  dbms_output.put_line('exp(0) = '||exp(0));
  dbms_output.put_line('power(5,2) = '||power(5,2));
  dbms_output.put_line('power(25,1/2) = '||power(25,1/2));
  dbms_output.put_line('power(5,-2) = '||power(5,-2));
  dbms_output.put_line('sqrt(16) = '||sqrt(16));
  dbms_output.put_line('log(100,10) = '||log(100,10));
  dbms_output.put_line('log(10,100) = '||log(10,100));
  dbms_output.put_line('log(2,16) = '||log(2,16));
  dbms_output.put_line('sign(-25.7) = '||sign(-25.7));
  dbms_output.put_line('sign(25.7) = '||sign(25.7));
  dbms_output.put_line('sign(0) = '||sign(0));
  dbms_output.put_line('greatest(-1,3,45,9,1) = '||greatest(-1,3,45,9,1));
```

```
abs(-10.493) = 10,493
ceil(-10.493) = -10
round(-10.493) = -10
round(-10.493,1) = -10,5
trunc(-10.493,1) = -10,4
floor(-10.493) = -11
floor(10.493) = 10
remainder(10,3) = 1
remainder(-10,3) = -1
mod(-10,3) = -1
bitand(0,1) = 0
bitand(15,7) = 7
bitand(5,3) = 1
cos(3.14/180*60) = ,5004596890082057
acos(0.5) = 60,030432871142545957884
sin(3.14/180*60) = ,8657598394923444
asin(0.5) = 30,015216435571272978942
tan(3.14/180*60) = 1,729929220089790
atan(0.5) = 26,578525356734108572791
exp(1) = 2,718281828459045235360287 4
exp(0) = 1
power(5,2) = 25
power(25,1/2) = 5,000000000000000000
power(5,-2) = ,04
sqrt(16) = 4
log(100,10) = ,5
log(10,100) = 2
log(2,16) = 3,999999999999999999999
sign(-25.7) = -1
sign(25.7) = 1
sign(0) = 0
greatest(-1,3,45,9,1) = 45
```

Символьные функции

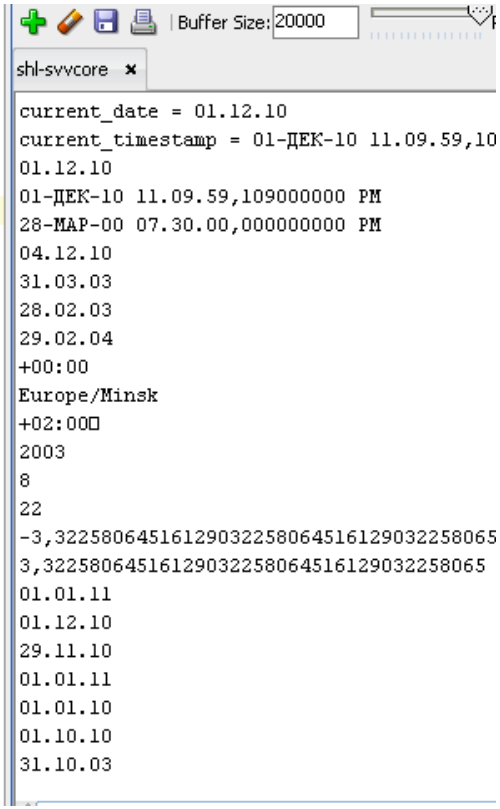
```
-- 12/55.sql
declare
  vvv varchar(200);
begin
dbms_output.put_line('------- ascii --------------');
  dbms_output.put_line('ascii(''a'') = '||ascii('a'));
  dbms_output.put_line('ascii(''A'') = '||ascii('A'));
  dbms_output.put_line('ascii(''Э'') = '||ascii('Э'));
  dbms_output.put_line('ascii(''Ю'') = '||ascii('Ю'));
  dbms_output.put_line('ascii(''я'') = '||ascii('я'));
dbms_output.put_line('------- asciistr------------');
  dbms_output.put_line('asciistr(''ABCЭЮя'') = '||asciistr('ABCЭЮя'));
dbms_output.put_line('------- chr------------');
  dbms_output.put_line( 'chr(97)= '||chr(97));
  dbms_output.put_line( 'chr(255)= '||chr(255));
dbms_output.put_line('------- unistr------------');
  dbms_output.put_line( 'unistr(''\042D'') = '||unistr('\042D'));
  dbms_output.put_line( 'unistr(''\044F'') = '||unistr('\044F'));
  dbms_output.put_line( 'unistr(''\0061'') = '||unistr('\0061'));
  dbms_output.put_line( 'unistr(''\0041'') = '||unistr('\0041'));
dbms_output.put_line('------- concat------------');
  dbms_output.put_line( 'concat(''12345'', ''54321'') = '||concat('12345', '54321'));
dbms_output.put_line('------- initcap------------');
  dbms_output.put_line( 'initcap(''abc эюя Tmm'') = '||initcap('abc эюя Tmm'));
dbms_output.put_line('------- instr------------');
  dbms_output.put_line( 'instr(''123456789'', ''34'') = '||instr('123456789', '34'));
  dbms_output.put_line( 'instr(''123456789'', ''34'', 5) = '||instr('123456789', '34', 5));
  dbms_output.put_line( 'instr(''12345678912345678'', ''34'', 5) = '||instr('1234567891234
  dbms_output.put_line( 'instr(''12345678912345678'', ''34'', 1,2) = '||instr('123456789123


exception
```



Работа с датами

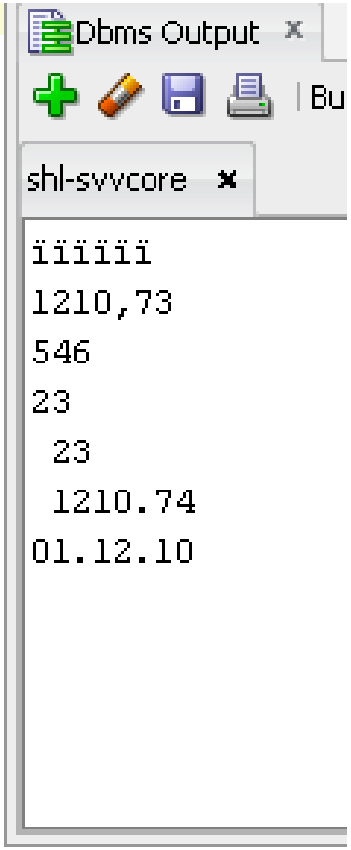
```
declare
    v varchar2(50);
begin
  dbms_output.put_line('current_date = '|| current_date);
  dbms_output.put_line('current_timestamp = '|| current_timestamp);
  dbms_output.put_line(sysdate);
  dbms_output.put_line(localtimestamp);
  dbms_output.put_line(sys_extract_utc(timestamp '2000-03-28 11:30:00.00 -08:00'));
  dbms_output.put_line(next_day('01-12-10', 'суббота'));
  dbms_output.put_line(last_day(to_date('2003/03/15', 'yyyy/mm/dd')));
  dbms_output.put_line(last_day(to_date('2003/02/03', 'yyyy/mm/dd')));
  dbms_output.put_line(last_day(to_date('2004/02/03', 'yyyy/mm/dd')));
  dbms_output.put_line(dbtimezone); -- CREATE/ALTER DATABASE
  dbms_output.put_line(sessiontimezone); -- CREATE/ALTER SESSION
  dbms_output.put_line(tz_offset('Europe/Minsk'));
  dbms_output.put_line(extract(year from date '2003-08-22'));
  dbms_output.put_line(extract(month from date '2003-08-22'));
  dbms_output.put_line(extract(day from date '2003-08-22'));
  dbms_output.put_line(months_between(sysdate, sysdate+100));
  dbms_output.put_line(months_between(sysdate+100, sysdate));
  dbms_output.put_line(round(to_date ('01-12-10'),'YEAR'));
  dbms_output.put_line(round(to_date ('02-12-10'),'MONTH'));
  dbms_output.put_line(round(to_date ('02-12-10'),'DAY'));
  dbms_output.put_line(round(to_date ('02-12-10'),'Q'));
  dbms_output.put_line(trunc(to_date ('01-12-10'),'YEAR'));
  dbms_output.put_line(trunc(to_date ('02-12-10'),'Q'));
  dbms_output.put_line(new_time (to_date ('2003/11/01 01:45', 'yyyy/mm/dd HH24:MI'), 'AST', 'MST'));
```



Функции конвертирования

```
-- 13/03.sql
declare
    v varchar2(3):='A';
begin
   dbms_output.put_line(convert('АБВГДЕ','WE8ISO8859P1'));
   dbms_output.put_line(to_number('1210.73', '9999.99'));
   dbms_output.put_line(to_number('546', '999'));
   dbms_output.put_line(to_number('23', '99'));
   dbms_output.put_line(to_char(23, '99'));
   dbms_output.put_line(to_char(1210.73777, '9999.99'));
   dbms_output.put_line(to_date('01.12.2010', 'DD.MM.YYYY'));

exception
   when others then dbms_output.put_line(sqlerrm);
end;
/
```



Sqlerrm и sqlcode

- Функция SQLERRM возвращает сообщение об ошибке, связанной с исключительной ситуацией
- Функция SQLCODE возвращает номер ошибки, связанной с исключительной ситуацией

Функции регулярных выражений

- Регулярные выражения - формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов

Функции регулярных выражений

- REGEXP_LIKE выбирает все строки, соответствующие заданному шаблону
  - REGEXP_LIKE (source_char, pattern [, match_parameter])
- REGEXP_INSTR определяет местоположение вхождения шаблона в строку
  - REGEXP_INSTR(source_char, pattern [, start_position [, nth_appearance [, return_option [, match_parameter [, sub_expression]]]]])

Функции регулярных выражений

- REGEXP_REPLACE заменяет шаблон выражения на заданный
  - REGEXP_REPLACE(source_char, pattern [, replacement_string [, start_position [, nth_appearance [, match_parameter]]]])
- REGEXP_SUBSTR выделяет из строки шаблон
  - REGEXP_SUBSTR(source_char, pattern [, start_position [, nth_appearance [, match_parameter [, sub_expression]]]])
- REGEXP_COUNT определяет количество вхождений
  - REGEXP_COUNT (source_char, pattern [, position [, match_param]])

Метасимволы привязки

Мета-символ	Описание	Пример
^	Привязать выражение к началу строки	REGEXP_LIKE(str,'^t') Результат: test11 => true 11123345 => false
$	Привязать выражение к концу строки	REGEXP_LIKE(str,'$5') Результат: test11 => false 11123345 => true

Квантификаторы и операторы повтора

Квантификатор	Описание	Пример
*	Встречается 0 и более раз	REGEXP_REPLACE(str, ’11*’, ‘1’) Результат: test11 => test1 11123345 => 123345
?	Встречается 0 или 1 раз	
+	Встречается 1 и более раз	REGEXP_LIKE(str,’5+’) Результат: test11 => false 11123345 => true
{m}	Встречается ровно m раз	REGEXP_LIKE(str,’3{2}’) Результат: test11 => false 11123345 => true
{m,}	Встречается по крайней мере m раз	
{m, n}	Встречается по крайней мере m раз, но не более n раз	

Предопределенные символьные классы POSIX

Класс символов	Описание
.	Любой символ
[:alpha:]	Буквы
[:lower:]	Буквы в нижнем регистре
[:upper:]	Буквы в верхнем регистре
[:digit:]	Цифры
[:alnum:]	Буквы и цифры
[:space:]	Пробелы (не печатаемые символы), такие как перевод каретки, новая строка, вертикальная табуляция и подача страницы
[:punct:]	Знаки препинания
[:cntrl:]	Управляющие символы (не печатаемые)
[:print:]	Печатаемые символы

Альтернативное сопоставление и группировка

Метасимвол		Описание	
|	Альтернатива	Разделяет альтернативные варианты, часто используется с оператором группировки ()	SELECT first_name, last_name FROM employees WHERE REGEXP_LIKE (first_name, '^Ste(vlph)en$'); Steven King Steven Markle Stephen Stiles
( )	Группа	Группирует выражения для альтернативы	
[char]	Список символов	Список символов, которые должны присутствовать в строке	SELECT last_name FROM employees WHERE REGEXP_LIKE (last_name, '([aeiou])\1', 'i'); De Haan Greenberg Khoo Gee Greene Lee
[^char]	Список символов	Список символов, которые не должны присутствовать в строке	

Метасимвол ссылки

Метасимвол		Описание
\digit	Обратная косая черта	Предыдущее сопоставление с соответствующим номером выражения в скобках Обратная косая черта может иметь другое значение

Вопросы?