ORACLE 12c

PL/SQL

Лекция 11

Встроенные функции

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

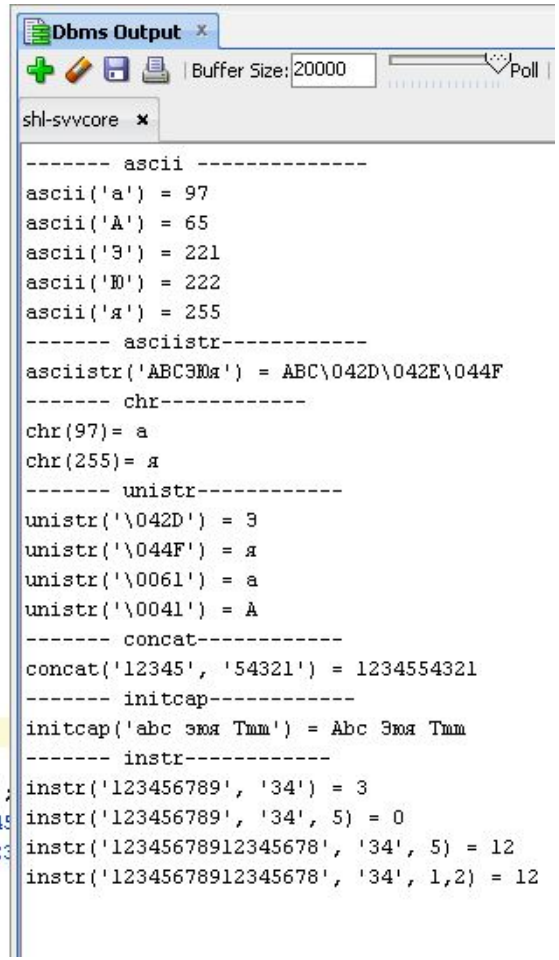
Числовые функции

```
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
```

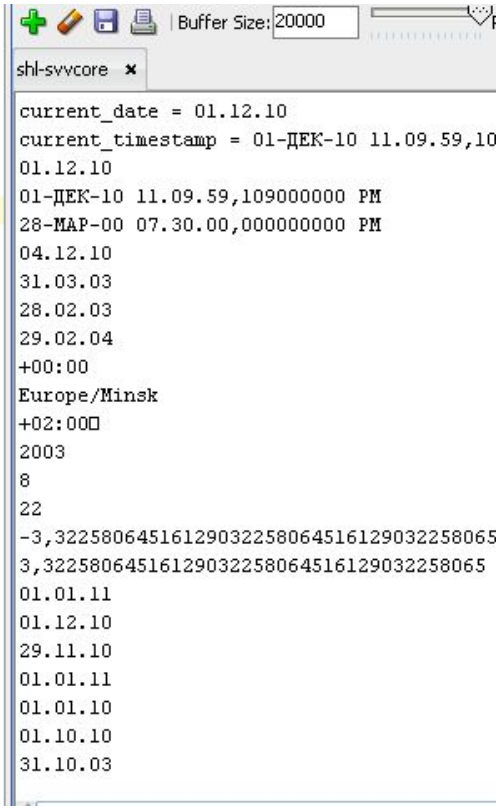
Символьные функции

```
-- 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
```



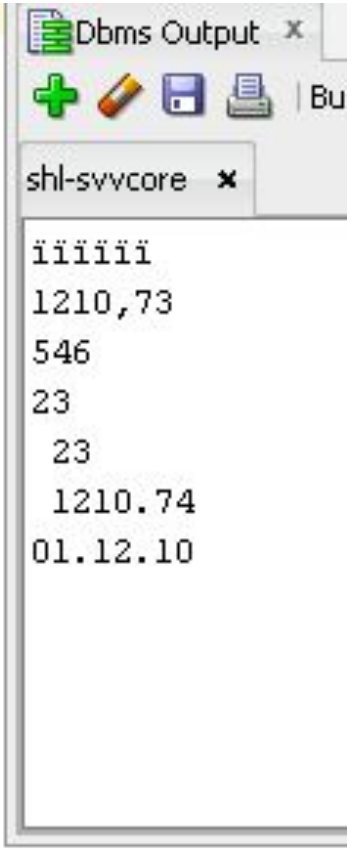
Работа с датами

```
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 из таблицы все строки, соответствующие заданному шаблону регулярного выражения REGEXP
  - select * from table_name
    where REGEXP_LIKE(name,'[0-9]{8}‘);
- REGEXP_INSTR определяет номер первого символа вхождения REGEXP шаблона в строку
  - select REGEXP_INSTR (name,'[0-9]{8}') from table_name;

Функции регулярных выражений

- REGEXP_REPLACE заменяет шаблон регулярного выражения REGEXP в строке на заданный
  - select REGEXP_REPLACE(name,'[0-9]{8}',‘date') from table_name;
- REGEXP_SUBSTR выделяет из строки заданный REGEXP шаблон
  - select REGEXP_SUBSTR(name, '[0 -9] {8}') from table_name;
- REGEXP_COUNT определяет количество вхождений REGEXP шаблона в строку
  - select REGEXP_COUNT(name,'[0-9]{1}') from table_name;

Коллекции

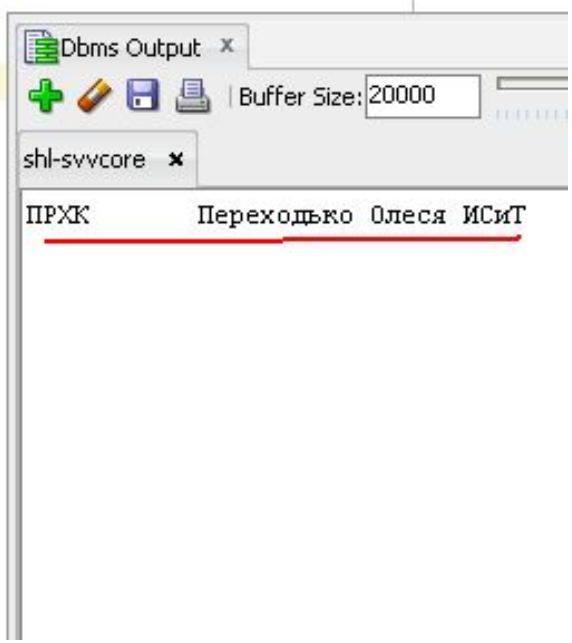
- Коллекция – структура данных, содержащая элементы одного типа
- Элементом коллекции может быть как скалярная величина, так и композитные данные
- Элементы коллекций можно сравнивать между собой на эквивалентность
- Можно передавать параметром в процедуру или функцию
- Можно создавать коллекции коллекций

Записи

- Запись – структура данных, составленная из нескольких частей информации, называемых полями.
- Для объявления записи вначале надо определить как тип, а потом объявить переменную типа «запись»
- Типы записей:
  - Табличные
  - Курсорные
  - Программно-определенные

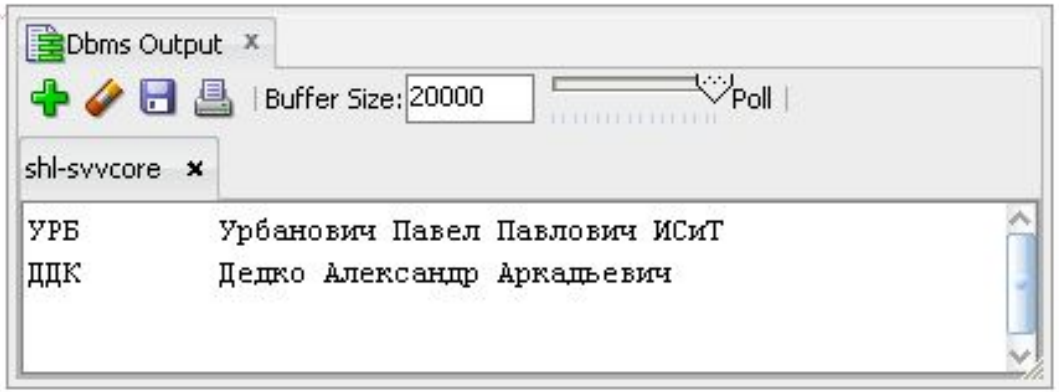
Записи

```
-- 13/04.sql
declare
  recl teacher%rowtype;
begin
  recl.teacher := 'ПРХК';
  recl.teacher_name := 'Переходько Олеся';
  recl.pulpit := 'ИСиТ';
  dbms_output.put_line(recl.teacher||' '||recl.teacher_name||' '||recl.pulpit);

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



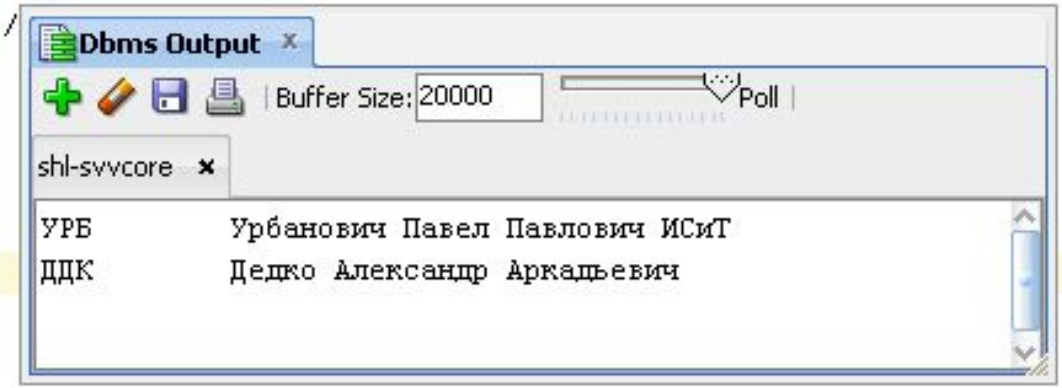
Записи

```
-- 13/05.sql
declare
   rec1 teacher%rowtype;
   type person is record
        (
         code char(10),
         name varchar2(100)
        );
   rec2 person;
begin
   select * into rec1  from teacher where teacher = 'УРБ';
   select teacher, teacher_name  into rec2  from teacher where teacher = 'ДДК';
   dbms_output.put_line(rec1.teacher||' '||rec1.teacher_name||' '||rec1.pulpit);
   dbms_output.put_line(rec2.code||' '||rec2.name);
exception
   when others then dbms_output.put_line(sqlerrm);
end;
/
```



Записи

```
-- 13/05.sql
declare
  rec1 teacher%rowtype;
  type person is record
       (
        code teacher.teacher%type,
        name teacher.teacher_name%type
       );
  rec2 person;
begin
  select * into rec1  from teacher where teacher = 'УРБ';
  select teacher, teacher_name  into rec2  from teacher where teacher = 'ДДК';
  dbms_output.put_line(rec1.teacher||' '||rec1.teacher_name||' '||rec1.pulpit);
  dbms_output.put_line(rec2.code||' '||rec2.name);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



Использование полей записи

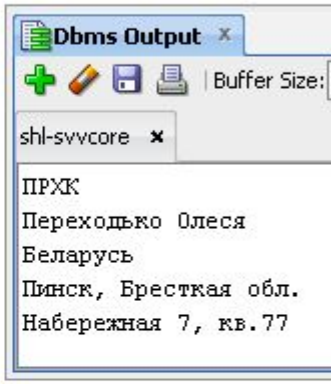
- Сравнение производится по полям записи
- Присвоение:
  - Присвоение для отдельного поля
  - SELECT INTO в запись в целом или в отдельные поля
  - Присвоение значения одной записи другой записи – для одного и того же объявления TYPE

Вложенные записи

```
-- 13/07.sql
declare
  rec1 teacher%rowtype;
  type address is record
      (
       address1 varchar2(100),
       address2 varchar2(100),
       address3 varchar2(100)
      );
    type person is record
       (
        code teacher.teacher%type,
        name teacher.teacher_name%type,
        homeaddress address
       );
  rec2 person;
begin
  rec2.code := 'ПРХК';
  rec2.name := 'Переходько Олеся';
  rec2.homeaddress.address1 :='Беларусь';
  rec2.homeaddress.address2 :='Пинск, Брестская обл.';
  rec2.homeaddress.address3 :='Набережная 7, кв.77';
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```

Присваивание записей

```
-- 13/10.sql
declare
  rec1 teacher%rowtype;
  type address is record
      (
       address1 varchar2(100),
       address2 varchar2(100),
       address3 varchar2(100)
      );
    type person is record
       (
        code teacher.teacher%type,
        name teacher.teacher_name%type,
        homeaddress address
       );
  rec2 person;
  rec3 person;
begin
 rec2.code := 'ПРХК';
 rec2.name := 'Переходько Олеся';
 rec2.homeaddress.address1 :='Беларусь';
 rec2.homeaddress.address2 :='Пинск, Брестская обл.';
 rec2.homeaddress.address3 :='Набережная 7, кв.77';
 rec3 := rec2;
 dbms_output.put_line(rec3.code);
 dbms_output.put_line(rec3.name);
 dbms_output.put_line(rec3.homeaddress.address1);
 dbms_output.put_line(rec3.homeaddress.address2);
 dbms_output.put_line(rec3.homeaddress.address3);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



Локальные программные модули

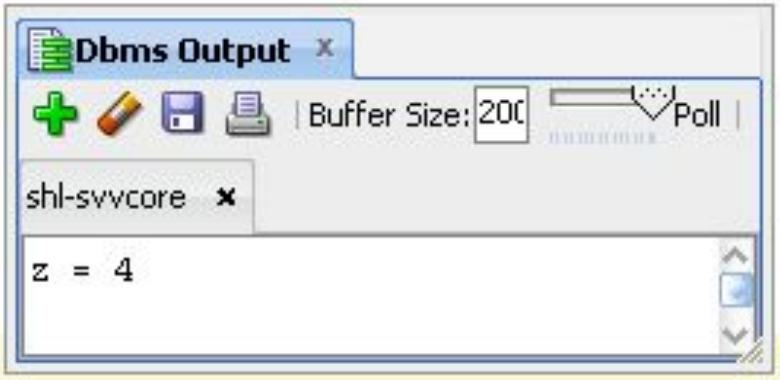
- Локальный программный модуль – это процедура или функция, определенная в секции декларации PL/SQL блока
- Объявление локальных процедур и функций должно размещаться в конце секции декларации после всех типов, записей, курсоров, переменных и исключений
- Локальные процедуры и функции могут быть использованы только в рамках блока, в котором они объявлены
- Локальные процедуры и функции могут быть перегружены

Перегрузка программных модулей

- Параметры должны отличаться семейством (number, character, datetime, boolean)
- Тип программного модуля должен отличаться – можно перегружать процедуру и функцию с одинаковым именем и списком параметров
- Число параметров должно быть разным

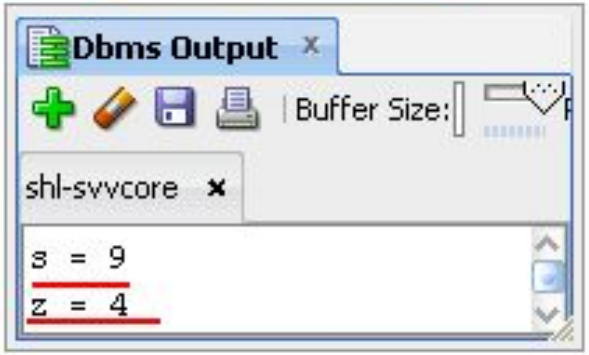
Локальные процедуры

```
-- 13/08.sql
declare
  x number(3):= 4;
  y number(3):= 5;
  z number(3);
  procedure summod5 (x1 number, x2 number, x3 out number )
  is
    z number(3):= 5;
    begin
      x3 := mod(x1+ x2,z);
    end  summod5;
begin
   summod5(x,y,z);
   dbms_output.put_line('z = '||z);

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



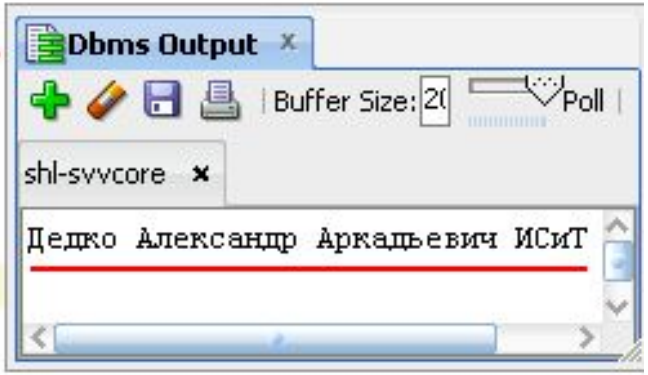
Локальные функции

```
-- 13/09.sql
declare
  x number(3):= 4;
  y number(3):= 5;
  z number(3);
  s number(5);
  function  summod5 (x1 number, x2 number, x3 out number)
    return number  is
        z number(3):= 5;
    begin
      x3 := mod(x1+ x2,z);
      return (x1+x2);
    end  summod5;
begin
    s := summod5(x,y,z);
    dbms_output.put_line('s = '||s);
    dbms_output.put_line('z = '||z);

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



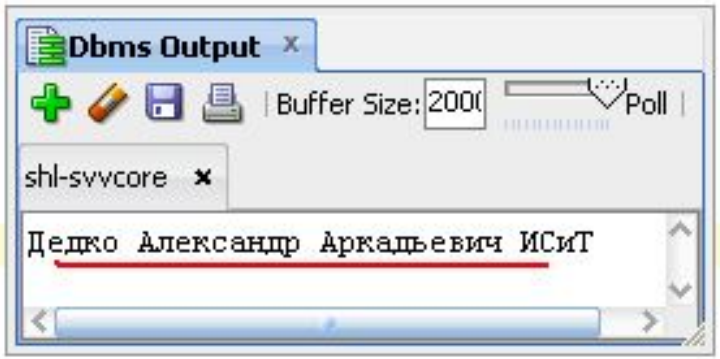
Запись - параметр процедуры и функции

```
-- 13/11.sql
declare
    type tperson is record
        (
         name    teacher.teacher_name%type,
         pulpit teacher.pulpit%type
        );
  rec tperson;
  procedure gettperson (code teacher.teacher%type, r out tperson)
  is
    begin
    select teacher.teacher_name, teacher.pulpit  into r
    from teacher
    where teacher = code;
  end  gettperson;
begin
    gettperson('ДДК', rec);
    dbms_output.put_line(rec.name||' '||rec.pulpit);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



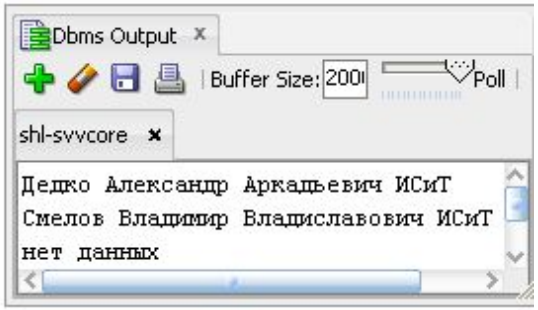
Записи

```
-- 13/12.sql
declare
    type tperson is record
        (
         name    teacher.teacher_name%type,
         pulpit teacher.pulpit%type
        );
  rec tperson;
  function  gettperson (code teacher.teacher%type) return tperson
  is
    r tperson;
    begin
    select teacher.teacher_name, teacher.pulpit  into r
    from teacher
    where teacher = code;
    return r;
  end  gettperson;
begin
    rec := gettperson('ДДК');
    dbms_output.put_line(rec.name||' '||rec.pulpit);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



Записи

```
declare
    type tperson is record
       (
        name    teacher.teacher_name%type,
        pulpit teacher.pulpit%type
       );
  rec tperson;
  function  gettperson (code teacher.teacher%type, r out  tperson) return boolean
  is
    begin
    select teacher.teacher_name, teacher.pulpit  into r
    from teacher
    where teacher = code;
    return true;
  exception
  when others then return false;
  end  gettperson;
begin
   if  gettperson('ДДК', rec)
   then  dbms_output.put_line(rec.name||' '||rec.pulpit);
   else  dbms_output.put_line('нет данных');
   end if;
   if  gettperson('СМЛВ', rec)
   then  dbms_output.put_line(rec.name||' '||rec.pulpit);
   else  dbms_output.put_line('нет данных');
   end if;
   if  gettperson('XXX', rec)
   then  dbms_output.put_line(rec.name||' '||rec.pulpit);
   else  dbms_output.put_line('нет данных');
   end if;
exception
   when others then dbms_output.put_line(sqlerrm);
end;
/
```



Коллекции

- Ассоциативные массивы = индексные таблицы (associative arrays, index-by tables)
- Вложенные таблицы (nested tables)
- Массивы переменной длины VARRAY

Массивы переменной длины

- Массивы переменной длины – одномерные, связанные коллекции однотипных элементов
- Доступны в рамках PL/SQL и в БД
- Являются плотными

Вложенные таблицы

- Вложенные таблицы – одномерные, несвязанные коллекции однотипных элементов
- Доступны в рамках PL/SQL и как поля таблицы в БД
- Изначально являются плотными, но могут впоследствии становиться разреженными

Ассоциативные массивы

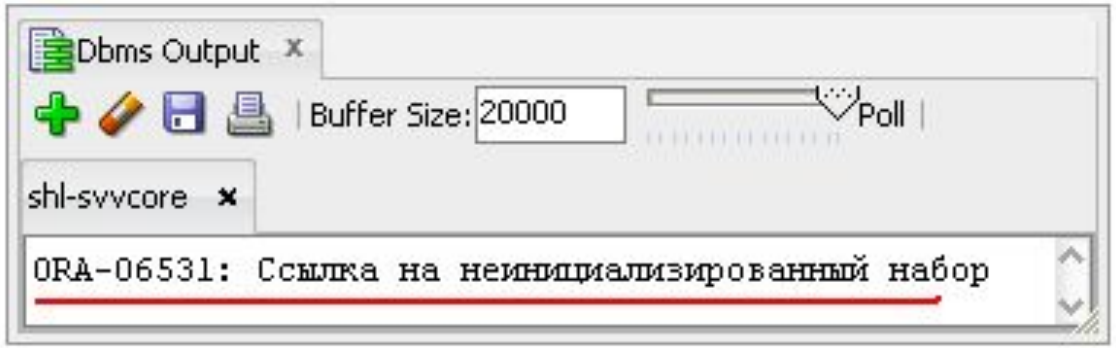
- Ассоциативные массивы – одномерные, неограниченные (по максимальному количеству элементов при создании) коллекции элементов
- Доступны только в рамках PL/SQL
- Изначально являются разреженными, индекс могут принимать непоследовательные значения

Работа с коллекциями

- Объявление коллекций
- Инициализация коллекций
  - Явно с помощью конструктора
  - Неявно при выборке из базы данных
  - Прямым присвоением переменной с другой коллекции такого же типа
- Добавление и удаление элементов
  - Ассоциативный массив – присвоение значения новому элементу
  - Вложенные таблицы и массивы переменной длины – сначала увеличить размер при помощи функции EXTEND, а затем присвоить значения новым элементам

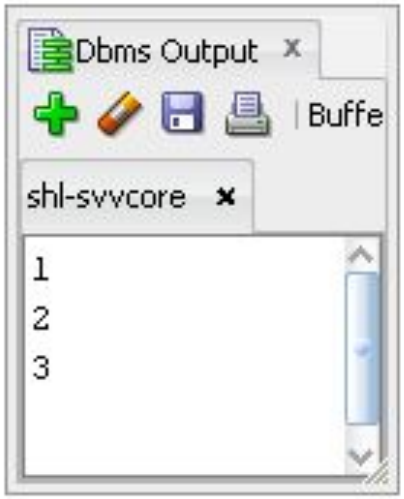
Массивы переменной длины

```
-- 13/14 sql
declare
    type myarraytype  is varray(3) of number;
    va myarraytype;

begin
   for i in 1..3 loop
   va(i) := 1;
   end loop;

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



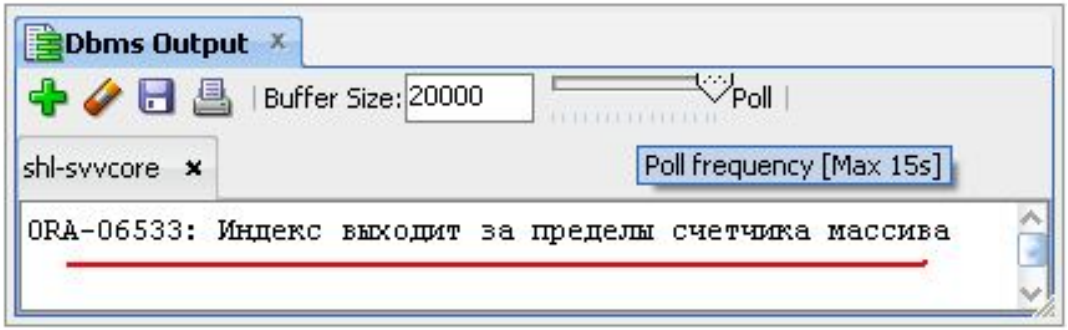
Массивы переменной длины

```
-- 13/14.sql
declare
    type myarraytype  is varray(3) of number;
    va myarraytype := myarraytype(null,null,null);

begin
   for i in 1..3
   loop
     va(i) := i;
   end loop;
   for i in 1..3
   loop
    dbms_output.put_line(va(i));
   end loop;


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



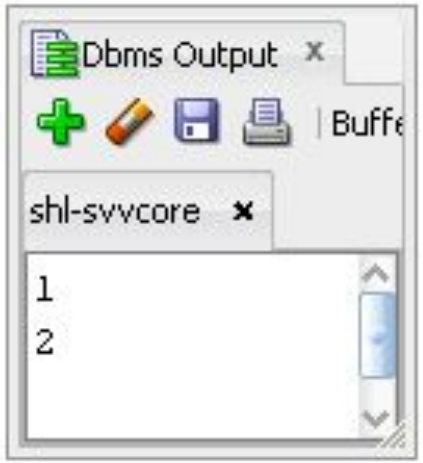
Массивы переменной длины

```
-- 13/16.sql
declare
    type myarraytype  is varray(3) of number;
    va myarraytype := myarraytype(null, null);

begin
   for i in 1..3
   loop
     va(i) := i;
   end loop;
   for i in 1..3
   loop
    dbms_output.put_line(va(i));
   end loop;


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



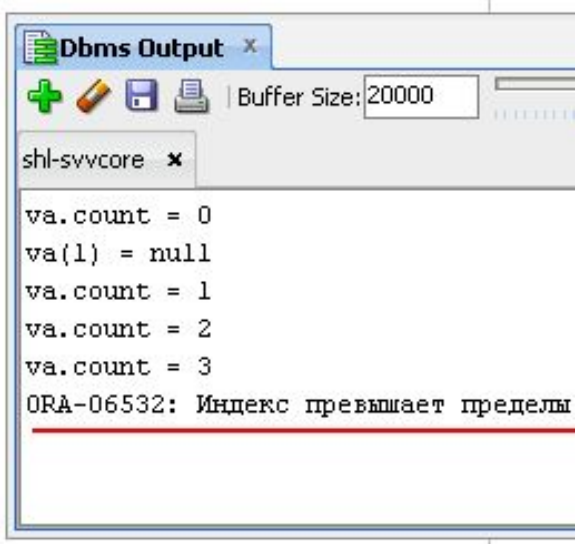
Массивы переменной длины

```
-- 13/17.sql
declare
    type myarraytype  is varray(3) of number;
    va myarraytype := myarraytype(null,null);

begin
   for i in 1..va.count
   loop
     va(i) := i;
   end loop;
   for i in 1..va.count()
   loop
    dbms_output.put_line(va(i));
   end loop;


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



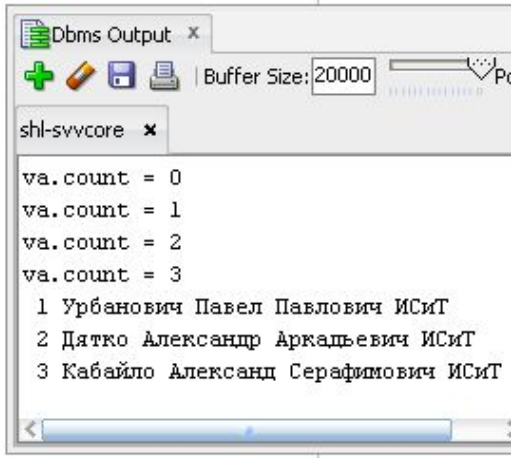
Массивы переменной длины

```
-- l3/l8.sql
declare
    type myarraytype  is varray(3) of number not null;
    va myarraytype := myarraytype();

begin
   dbms_output.put_line('va.count = '||va.count);

   va.extend;

   if va(1) is null
   then  dbms_output.put_line('va(1) = null '||va(1));
   else dbms_output.put_line('va(1) = '||va(1));
   end if;
   dbms_output.put_line('va.count = '||va.count);

   va.extend;
   dbms_output.put_line('va.count = '||va.count);

   va.extend;
   dbms_output.put_line('va.count = '||va.count);

   va.extend;
   dbms_output.put_line('va.count = '||va.count);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



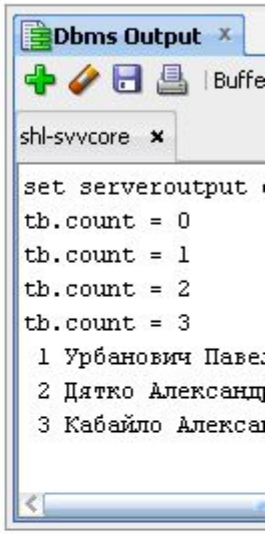
Массивы переменной длины

```
-- 13/19.sql
declare
    type tperson is record
       (
        name    teacher.teacher_name%type,
        pulpit teacher.pulpit%type
       );
    type myarraytype  is varray(3) of tperson;
    va myarraytype := myarraytype();
    rec tperson; --    := tperson('xxx','xxxxxx');
begin
   dbms_output.put_line('va.count = '||va.count);

   va.extend;
   dbms_output.put_line('va.count = '||va.count);
   va(1).name    := 'Урбанович Павел Павлович';
   va(1).pulpit := 'ИСиТ';
   va.extend;
   dbms_output.put_line('va.count = '||va.count);
   va(2).name    := 'Дятко Александр Аркадьевич';
   va(2).pulpit := 'ИСиТ';
   va.extend;
   dbms_output.put_line('va.count = '||va.count);
   va(3).name    := 'Кабайло Александ Серафимович';
   va(3).pulpit := 'ИСиТ';
   for i in 1..va.count
   loop
    dbms_output.put_line( ' '||i||' '||va(i).name||' '||va(i).pulpit);
   end loop;
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



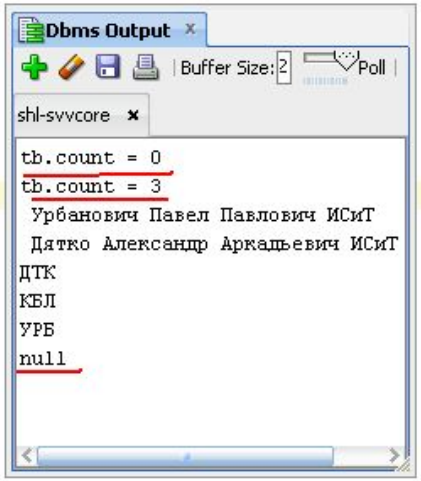
Вложенные таблицы

```
-- 14/01.sql
declare
    type tperson is record
       (
        name    teacher.teacher_name%type,
        pulpit teacher.pulpit%type
       );
    type mytable  is table  of tperson;
    tb mytable := mytable();
    rec tperson; --    := tperson('xxx','xxxxxx');
begin
   dbms_output.put_line('tb.count = '||tb.count);

   tb.extend;
   dbms_output.put_line('tb.count = '||tb.count);
   tb(1).name    := 'Урбанович Павел Павлович';
   tb(1).pulpit := 'ИСиТ';
   tb.extend;
   dbms_output.put_line('tb.count = '||tb.count);
   tb(2).name    := 'Дятко Александр Аркадьевич';
   tb(2).pulpit := 'ИСиТ';
   tb.extend;
   dbms_output.put_line('tb.count = '||tb.count);
   tb(3).name    := 'Кабайло Александ Серафимович';
   tb(3).pulpit := 'ИСиТ';
   for i in 1..tb.count
   loop
    dbms_output.put_line( ' '||i||' '||tb(i).name||' '||tb(i).pulpit);
   end loop;
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```



Ассоциативные массивы

```
-- 14/02.sql
declare
    type tperson is record (name   teacher.teacher_name%type, pulpit teacher.pulpit%type );
    type mytable  is table  of tperson index by teacher.teacher_name%type;
    tb mytable;
    ntx  teacher.teacher_name%type;
    rec tperson;
begin
 dbms_output.put_line('tb.count = '||tb.count);
 tb('УРБ').name   := 'Урбанович Павел Павлович';
 tb('УРБ').pulpit := 'ИСиТ';
 tb('ДТК').name   := 'Дятко Александр Аркадьевич';
 tb('ДТК').pulpit := 'ИСиТ';
 tb('КБЛ').name   := 'Кабайло Александ Серафимович';
 tb('КБЛ').pulpit := 'ИСиТ';
 dbms_output.put_line('tb.count = '||tb.count);
 dbms_output.put_line( ' '||tb('УРБ').name||' '||tb('УРБ').pulpit);
 rec := tb('ДТК');
 dbms_output.put_line( ' '||rec.name||' '||rec.pulpit);
 ntx := tb.first;
 if ntx is not null then  dbms_output.put_line(ntx);
 else  dbms_output.put_line('null');
 end if;
 ntx := tb.next(ntx);
 if ntx is not null then  dbms_output.put_line(ntx);
 else  dbms_output.put_line('null');
 end if;
 ntx := tb.next(ntx);
 if ntx is not null then  dbms_output.put_line(ntx);
 else  dbms_output.put_line('null');
 end if;
 ntx := tb.next(ntx);
 if ntx is not null  then  dbms_output.put_line(ntx);
 else  dbms_output.put_line('null');
 end if;
exception
  when others then dbms_output.put_line(sqlerrm);
end;
```



Методы и исключения коллекций

```
-- 14/03.sql
declare

begin
null;
----------methods -------------------------
-- COUNT                          ALL
-- DELETE(n), DELETE(n,m)         ALL
-- EXISTS(n)                      ALL
-- EXTEND, EXTEND(n)              VARRAY,TABLE
-- FIRST                          ALL
-- LAST                           ALL
-- LIMIT                          VARRAY
-- NEXT(n)                        ALL
-- PRIOR(n)                       ALL
-- TRIM                           ALL
-- TRIM(n)                        ALL
---------- exceptions ----------------------
-- COLLECTION_IS_NULL             ALL
-- NO_DATA_FOUND                  ALL
-- SUBSCRIPT_BEYOND_COUNT         VARRAY, TABLE
-- SUBSCRIPT_OUTSIDE_LIMIT        VARRAY, TABLE
-- VALUE_ERROR                    ALL
end;
/
```

Сравнение характеристик коллекций

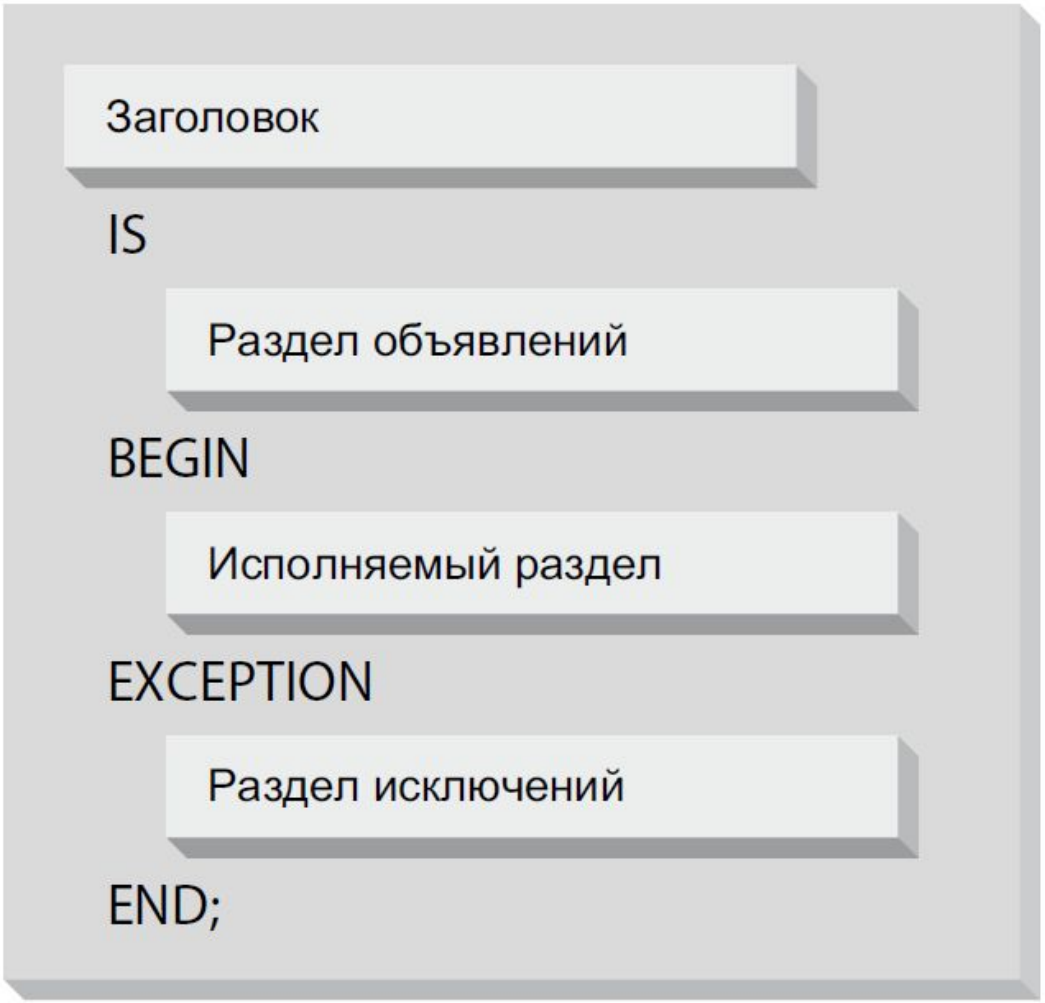
- Размерность?
- Можно ли использовать как поле в таблице?
- Неинициализированное состояние?
- Инициализация?
- Диапазон индексов?
- Разреженность?
- Ограничен по максимальному количеству элементов?
- Можно ли присваивать значение любому элементу?
- Метод расширения и уменьшения?
- Можно ли сравнивать на равенство весь объект целиком?
- Элементы сохраняют позицию при записи или чтении из БД?

Обработка исключений

- Исключительная ситуация – событие, возникающее в программе и требующее незамедлительной обработки.
- Два типа исключительных ситуаций:
  - 1) программно-определяемые исключения;
  - 2) предопределенные (стандартные) исключения.

Обработка исключений

- Ошибка, сгенерированная сервером
- Ошибка в результате действий пользователя
- Ошибка, сгенерированная приложением пользователю



Стратегия обработки исключений

- Как и где будут фиксироваться ошибки, чтобы их можно было просмотреть и откорректировать?
- Как выдавать пользователю сообщения об ошибках?
- Нужно ли включать обработку исключений в каждый PL/SQL блок?
- Как управлять транзакцией в случае ошибки?

Термины обработки исключений

- Секция исключений – необязательная секция в PL/SQL блоке, которая содержит один или несколько обработчиков исключений.
- RAISE (RAISE_APPLICATION_ERROR)– команда, которая прерывает выполнение текущего блока.
- Обработка исключений – перехват ошибки в секции исключений.
- Область действия – часть кода, в рамках которого может быть сгенерировано исключение.
- Распространение исключения – процесс передачи исключений от одного блока другому, если исключение не было обработано.

Термины обработки исключений

- Необработанное исключение – исключение становится необработанным, если оно не обработано блоком самого верхнего уровня.
- Неименованное исключение – исключение, которое имеет код ошибки и сообщение, но не имеет наименования, не может быть использовано в команде RAISE или в секции WHEN.
- Именованное исключение – исключение, которому было определено наименование.

Предопределенные исключения

```
-- 14/05.sql
declare
begin
null;
-- DUP_VAL_ON_INDEX                нарушена уникальность
-- TIMEOUT_ON_RESORCE              истекло время ожидания ресурса
-- TRANSACTION_BACKED_OUT
-- INVALID_CURSOR                  запрещенная операция с курсором
-- NOT_LOGGED_ON                   отсутсвует подключение к Oracle
-- NO_DATA_FOUND                   данные не найдены
-- SYS_INVALID_ROWID               не может быть получен ROWID
-- TOO_MANY_ROWS                   не точная выборка в SELECT
-- ZERO_DIVIDE                     деление на нуль
-- USERENV_COMMITSCN_ERROR         ошибка USERENV('COMMITSCN')
-- INVALID_NUMBER                  ошибка  преобразования в NUMBER
-- STORAGE_ERROR                   PL/SQL недостаточо памяти
-- PROGRAMM_ERROR                  внутренняя ошибка PL/SQL
-- VALUE_ERROR                     ошибка преобразование или усечение точности
-- ROWTYPE_MISMATCH                переменнная PL/SQL и курсорная переменна не сместимы
-- CURSOR_AREADY_OPEN              попытка открыть открытый курсор
-- ACCESS_INTO_NULL                попытка присвоить значение атрибуту NULL-объекта
-- CASE_NOT_FOUND                  нет подходящей фразы  WHEN в операторе CASE
-- SELF_IS_NULL                    попытка вызвать метод NULL-объекта
-- INVALID_PATH
-- INVALID_MODE
-- INVALID_FILEHANDLE
-- INVALID_OPERATION
-- READ_ERROR
-- WRITE_ERROR
-- INTERNAL_ERROR
-- INVALID_MAXLINESIZE
-- INALID_FILENAME
-- ACCESS_DENIED
-- INVALID_OFFSET
-- DELETE_FAILED
-- RENAME_FAILED
-- NO_DATA_NEEDED
-- collection exceptions           см. типы исключений для коллекций
end;
/
```

Объявление именованных исключений

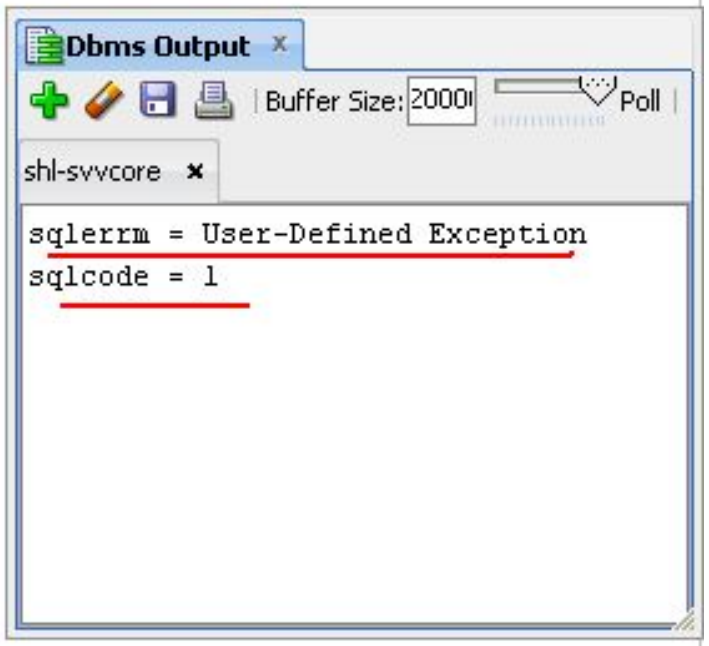
- Чтобы обработать исключение, которое не относится определенным сервером, его необходимо объявить:
  - exception_name EXCEPTION;
- Имена исключений могут быть использованы только для генерации исключения при помощи RAISE и для перехвата исключения в секции WHEN

Связывание исключений с кодом ошибки

- Синтаксис:
  - exception_name EXCEPTION;
  - PRAGMA EXCEPTION_INIT (exception_name, integer);
- Где exception_name – наименование исключения, integer – номер(код) ошибки сервера ORACLE, с которым необходимо связать исключение

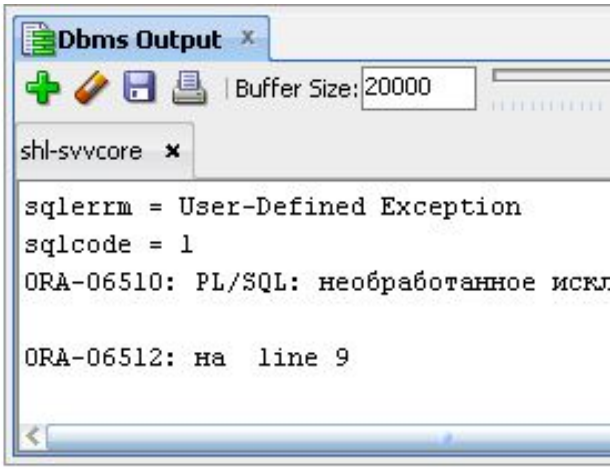
Генерация и обработка исключений

```
-- 14/06.sql
declare
    e_namesake exception;
    n NUMBER(5);
begin
 select count(*) into n
   from teacher t1,  teacher t2
   where t1.teacher_name = t2.teacher_name and
        t1.teacher != t2.teacher;
 if n = 0 then raise e_namesake;
 end if;
 dbms_output.put_line(n);
exception
    when no_data_found
      then  dbms_output.put_line('no_data_found');
    when not_logged_on
      then  dbms_output.put_line('no_logged_on');
    when timeout_on_resource
      then  dbms_output.put_line('timeout_on_resource');
    when others
       then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



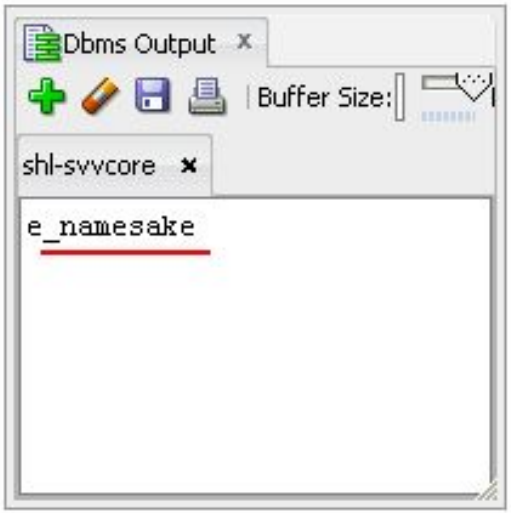
Генерация и обработка исключений

```
-- 14/07.sql
declare
    e_namesake exception;
    n NUMBER(5);
begin
 select count(*) into n
   from teacher t1,  teacher t2
   where t1.teacher_name = t2.teacher_name and
        t1.teacher != t2.teacher;
 if n = 0 then raise e_namesake;
 end if;
 dbms_output.put_line(n);
exception
    when no_data_found
      then  dbms_output.put_line('no_data_found');
    when not_logged_on
      then  dbms_output.put_line('no_logged_on');
    when timeout_on_resource
      then  dbms_output.put_line('timeout_on_resource');
    when others
       then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
            dbms_output.put_line(
                 substr(dbms_utility.format_error_stack, 1,200)
                              );
             dbms_output.put_line(
                 substr(dbms_utility.format_error_backtrace, 1,200)
                              );
end;
```



Генерация и обработка исключений

```
-- 14/08.sql
declare
    e_namesake exception;
    n NUMBER(5);
begin
 select count(*) into n
   from teacher t1,  teacher t2
   where t1.teacher_name = t2.teacher_name and
        t1.teacher != t2.teacher;
 if n = 0 then raise e_namesake;
 end if;
 dbms_output.put_line(n);
exception
    when no_data_found
      then  dbms_output.put_line('no_data_found');
    when not_logged_on
      then  dbms_output.put_line('no_logged_on');
    when timeout_on_resource
      then  dbms_output.put_line('timeout_on_resource');
    when   e_namesake
      then  dbms_output.put_line('e_namesake');
    when others
       then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



Генерация и обработка исключений

SQL Worksheet | History

0,046 seconds

```
1   -- 14/09.sql
2   declare
3       e_namesake exception;
4       pragma exception_init(e_namesake, 100);
5       t teacher.teacher_name%type;
6   begin
7    select tl.teacher into t
8      from teacher tl,  teacher t2
9      where tl.teacher_name = t2.teacher_name and
10           tl.teacher != t2.teacher and
11           rownum < 2;
12   dbms_output.put_line(t);
13  exception
14      when no_data_found
15        then  dbms_output.put_line('no_data_found');
16      when not_logged_on
17        then  dbms_output.put_line('no_logged_on');
18      when timeout_on_resource
19        then  dbms_output.put_line('timeout_on_resource');
20     when   e_namesake
21           then  dbms_output.put_line('e_namesake');
22      when others
23         then dbms_output.put_line('sqlerrm = '||sqlerrm);
24              dbms_output.put_line('sqlcode = '||sqlcode);
25  end;
26
```

Dbms Output

Buffer Size:

shl-svvcore

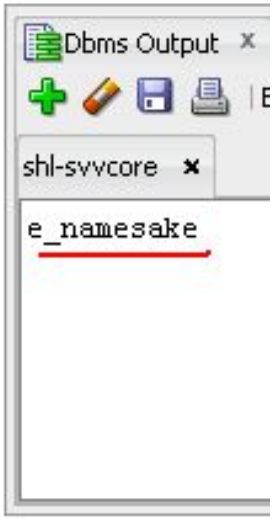
Script Output

Task completed in 0,046 seconds

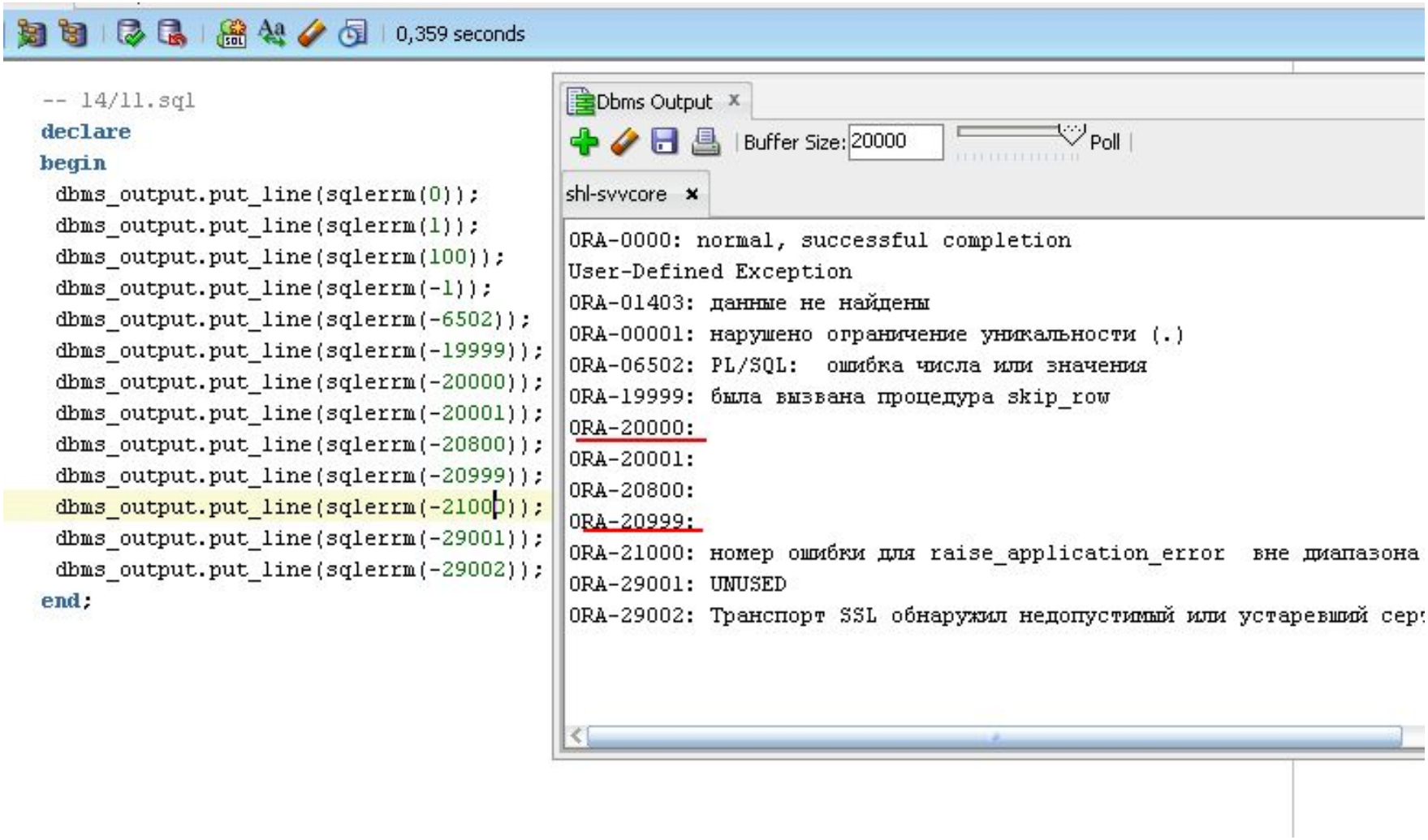
```
Error report:
ORA-06550: Строка 19, столбец 8:
PLS-00484: дополнительные исключения 'NO_DATA_FOUND' и 'E_NAMESAKE' должны появляться в том же обработчике исключений
ORA-06550: Строка 0, столбец 0:
```

Генерация и обработка исключений

```
-- 14/10.sql
declare
    e_namesake exception;
    pragma exception_init(e_namesake, 100);
    t teacher.teacher_name%type;
begin
 select t1.teacher into t
   from teacher t1,  teacher t2
   where t1.teacher_name = t2.teacher_name and
         t1.teacher != t2.teacher and
         rownum < 2;
 dbms_output.put_line(t);
exception
  --  when no_data_found
  --    then  dbms_output.put_line('no_data_found');
    when not_logged_on
      then  dbms_output.put_line('no_logged_on');
    when timeout_on_resource
      then  dbms_output.put_line('timeout_on_resource');
   when   e_namesake
         then  dbms_output.put_line('e_namesake');
    when others
       then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



Sqlerrm и sqlcode



RAISE_APPLICATION_ERROR

```
-- 14/12.sql
declare
    e_namesake exception;
    pragma exception_init(e_namesake, 100);
    t teacher.teacher_name%type;
begin
 select tl.teacher into t
   from teacher tl,  teacher t2
   where tl.teacher_name = t2.teacher_name and
         tl.teacher != t2.teacher and
         rownum < 2;
  dbms_output.put_line(t);
exception
  --  when no_data_found
    when not_logged_on
      then  dbms_output.put_line('no_logged_on');
    when timeout_on_resource
      then  dbms_output.put_line('timeout_on_resource');
   when   e_namesake
          then  dbms_output.put_line('e_namesake');
                raise_application_error(-20001,'e_namesake');
    when others
       then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
end;
```

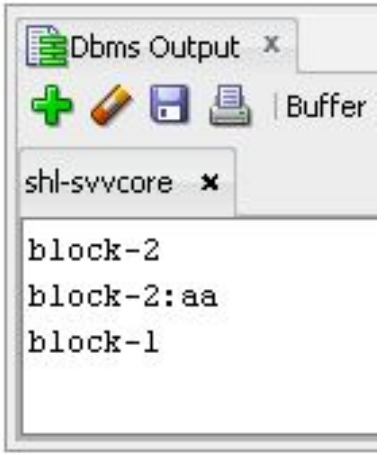

```
Error report:
ORA-20001: e namesake
ORA-06512: на  line 20
```

RAISE_APPLICATION_ERROR

- Определена в пакете DBMS_ STANDARD
- Можно присвоить сообщение об ошибке
- При выполнении процедуры:
  - Выполнение блока прерывается
  - Любые изменения в аргументах IN OUT и OUT откатываются
  - Изменения в глобальных структурах (пакетные переменные, объекты базы данных) не откатываются – для отката надо явно выполнить ROLLBACK

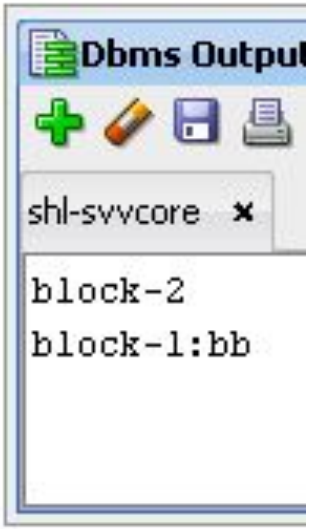
Распространение исключения

```
-- 14/13.sql
declare
    aa exception;
begin       -- 1
  begin     -- 2
    dbms_output.put_line('block-2');
    raise aa;
  exception
    when aa  then  dbms_output.put_line('block-2:aa');
  end;      -- 2
  dbms_output.put_line('block-1');
exception
    when others then dbms_output.put_line('block-1:others');
end;        -- 1
```



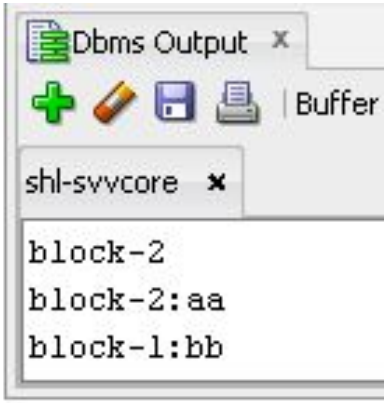
Распространение исключения

```
-- 14/14.sql
declare
    aa exception;
    bb exception;
begin       -- 1
  begin     -- 2
    dbms_output.put_line('block-2');
    raise bb;
  exception
    when aa  then  dbms_output.put_line('block-2:aa');
  end;      -- 2
  dbms_output.put_line('block-1');
exception
    when bb     then dbms_output.put_line('block-1:bb');
    when others then dbms_output.put_line('block-1:others');
end;        -- 1
```



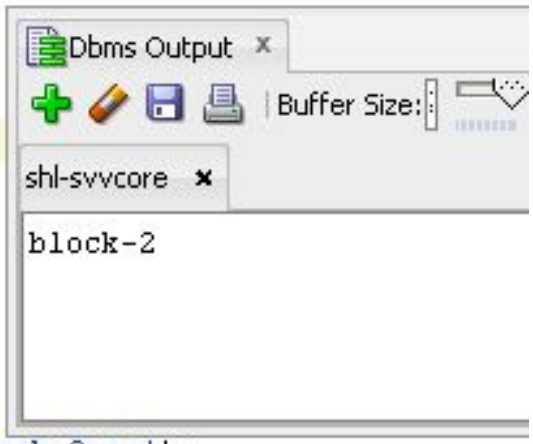
Распространение исключения

```
declare
    aa exception;
    bb exception;
    cc exception;
begin       -- 1
  begin     -- 2
    dbms_output.put_line('block-2');
    raise aa;
  exception
    when aa  then  dbms_output.put_line('block-2:aa');
                   raise bb;
    when others  then
       dbms_output.put_line('block-2:others');
  end;      -- 2
  dbms_output.put_line('block-1');
exception
    when bb  then
       dbms_output.put_line('block-1:bb');

end;        -- 1
```



Распространение исключения

```
-- 14/16.sql
declare
    aa exception;
    bb exception;
    cc exception;
begin       -- 1
  begin     -- 2
    dbms_output.put_line('block-2');
    raise cc;
  exception
    when aa  then  dbms_output.put_line('block-2:aa');
  end;      -- 2
  dbms_output.put_line('block-1');
exception
    when bb     then dbms_output.put_line('block-1:bb');
    -- when others then dbms_output.put_line('block-1:others');
end;        -- 1
```




```
Error report:
ORA-06510: PL/SQL: необработанное исключение, определенное пользователем
ORA-06512: на  line 8
```

Вопросы?