

ORACLE 12c

PL/SQL

Лекция 11

Записи

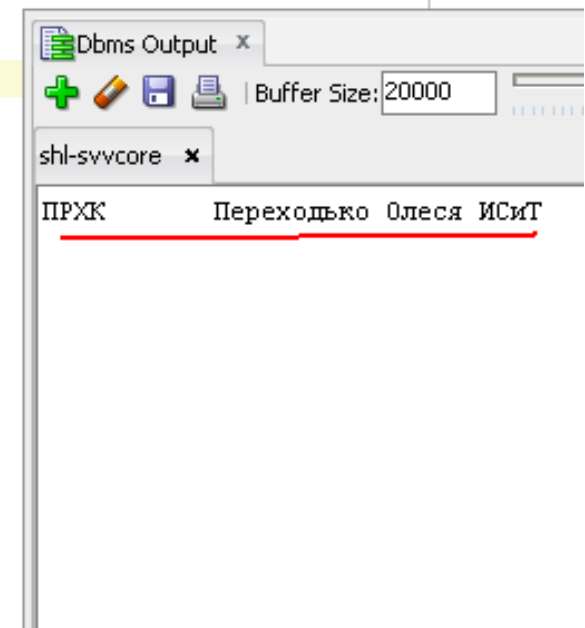
- ▶ Запись – структура данных, составленная из нескольких частей информации, называемых полями
- ▶ Для объявления записи вначале надо определить как тип, а потом объявить переменную типа «запись»
- ▶ Типы записей:
 - ▶ Табличные
 - ▶ Курсорные
 - ▶ Программно-определенные



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 = 'УРЕ';
```

```
select teacher, teacher_name into rec2 from teacher where teacher = 'ДДК';
```

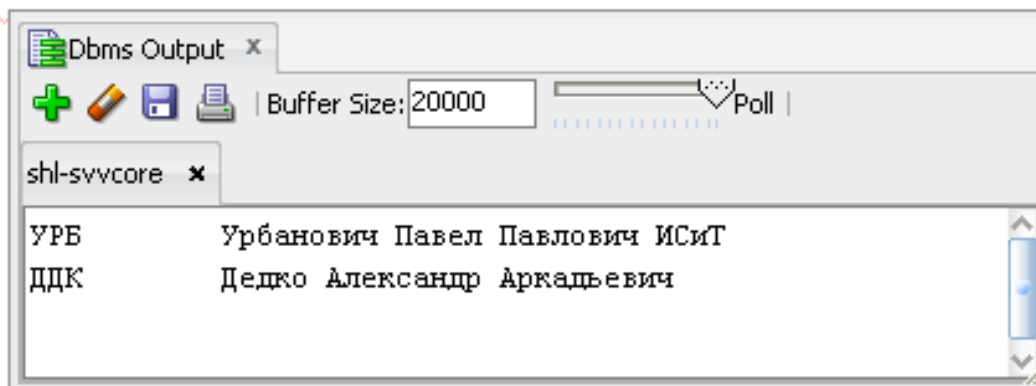
```
dbms_output.put_line(rec1.teacher||' '||rec1.teacher_name||' '||rec1.pulpit);
```

```
dbms_output.put_line(rec2.code||' '||rec2.name);
```

```
exception
```

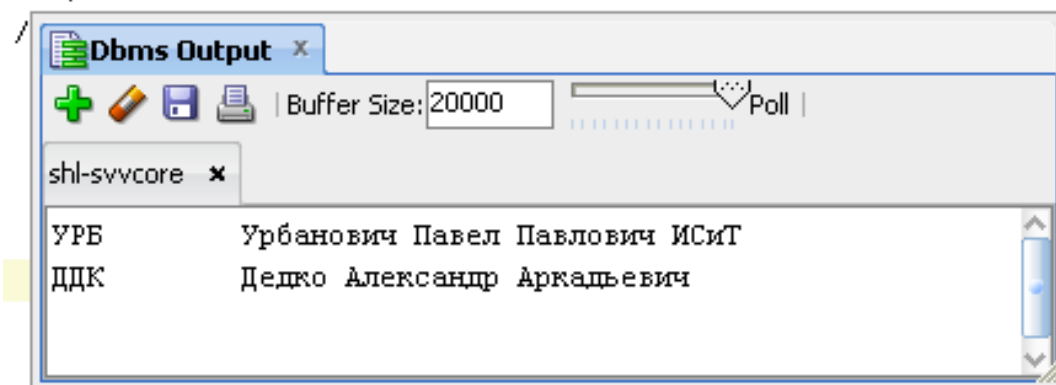
```
when others then dbms_output.put_line(sqlerrm);
```

```
end;
```



Записи

```
-- 10/00.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 = 'УРЕ';  
    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, Бресткая 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, Бресткая 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);

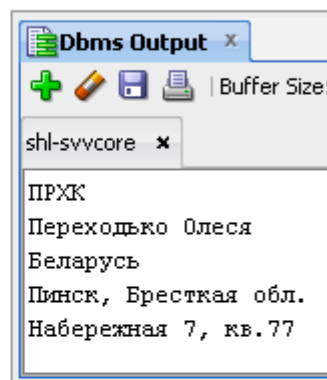
dbms_output.put_line(rec3.homeaddress.address2);

dbms_output.put_line(rec3.homeaddress.address3);

exception

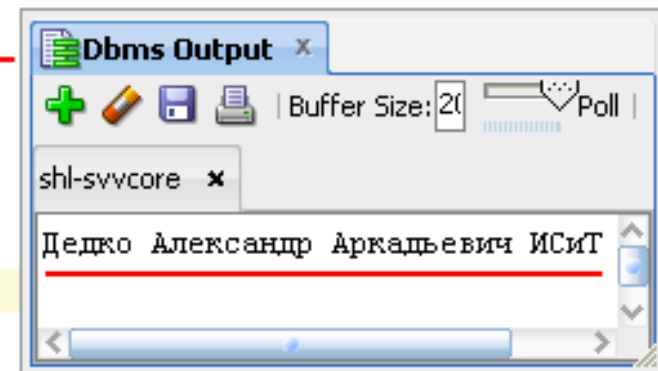
when others **then** dbms_output.put_line(sqlerrm);

end;



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');
```

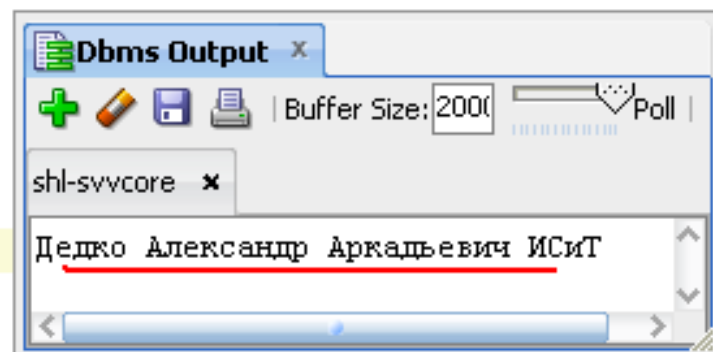
```
    dbms_output.put_line(rec.name || ' ' || rec.pulpit);
```

```
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('СМУБ', 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');
```

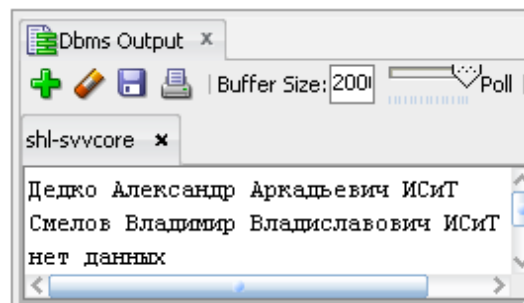
```
end if;
```

```
exception
```

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

```
end;
```

```
/
```



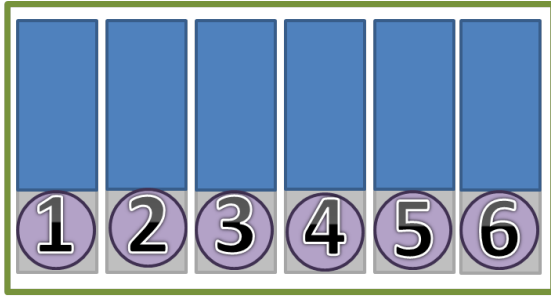
Коллекции

- ▶ 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
 - ▶ Одномерная, но можно создавать коллекции коллекций
 - ▶ Массив переменной длины VARRAY
 - ▶ Вложенная таблица (nested tables)
 - ▶ Ассоциативный массив (associative array)
-

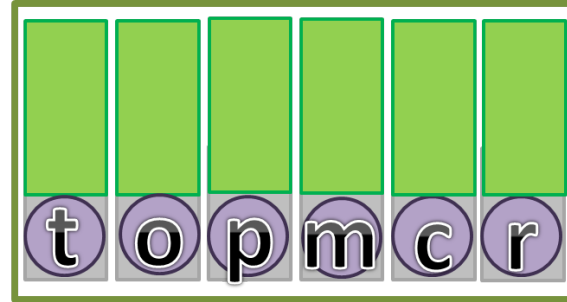


Collection Types

Index by
PLS_INTEGER

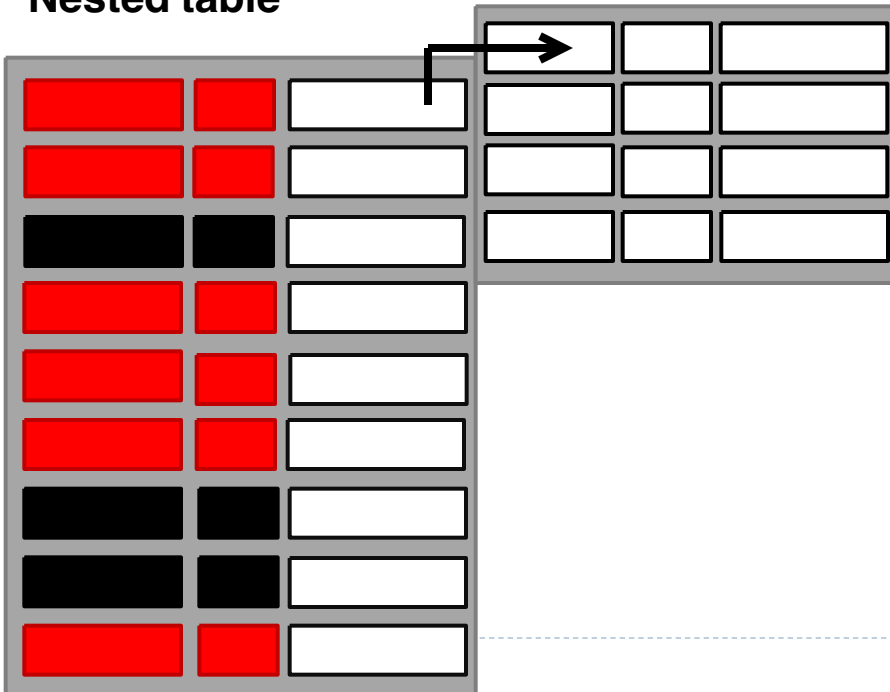


Index by
VARCHAR

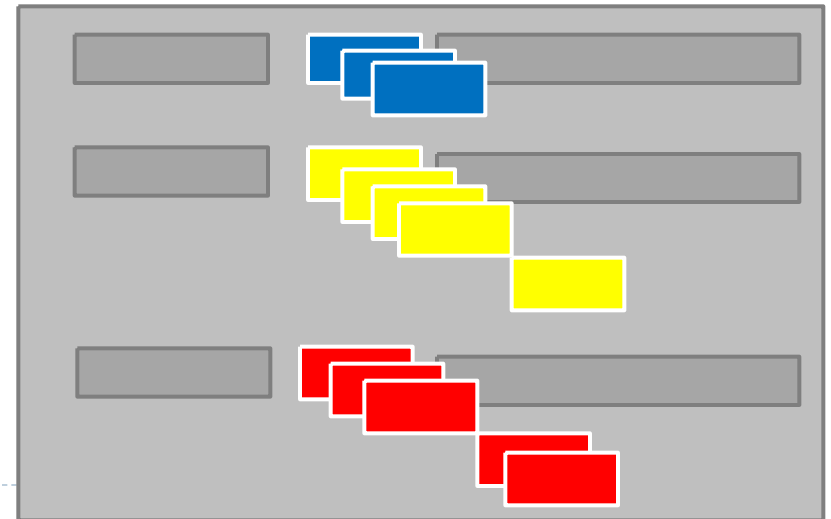


Associative array

Nested table



Varray



Термины

- ▶ Коллекция состоит из **набора** элементов, причем каждый элемент находится в определенной позиции (имеется **индекс** элемента)
- ▶ Необходимо объявить **тип коллекции** – командой TYPE
- ▶ Необходимо объявить **коллекцию** – переменную этого типа для дальнейшего использования



Ограниченная и неограниченная коллекция

- ▶ Коллекция называется **ограниченной**, если заранее определены границы возможных значений индексов ее элементов, иначе **неограниченной**
- ▶ Коллекции типа `VARRAY` всегда ограничены
- ▶ Вложенные таблицы и ассоциативные массивы неограничены



Разреженные и плотные коллекции

- ▶ Коллекция называется плотной, если все ее элементы, определены и каждому из них присвоено некоторое значение (таковым может быть и NULL)
- ▶ Массивы VARRAY всегда являются плотными
- ▶ Вложенные таблицы первоначально всегда плотные, но по мере удаления некоторых элементов становятся разреженными
- ▶ Ассоциативные массивы могут быть как разреженными, так и плотными в зависимости от способа их заполнения



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**
 - ▶ Вложенные таблицы и массивы переменной длины – сначала увеличить размер при помощи функции EXTEND, а затем присвоить значения новым элементам
 - ▶ Ассоциативный массив – присвоение значения новому элементу



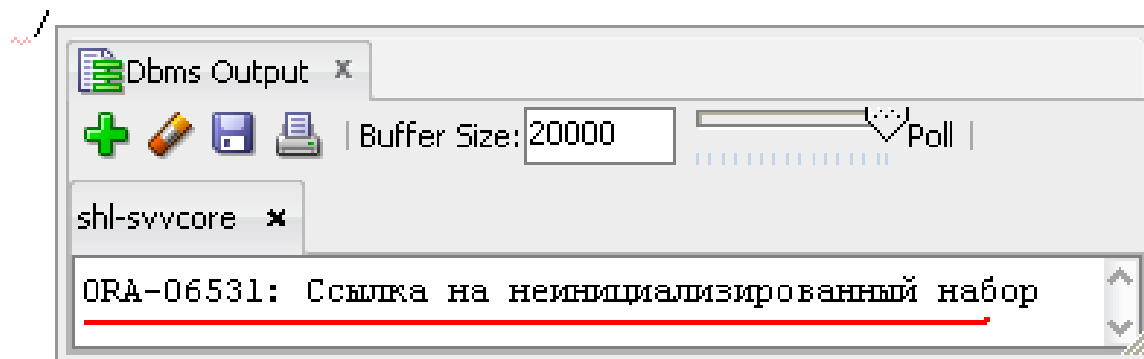
Массивы переменной длины

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



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));
```

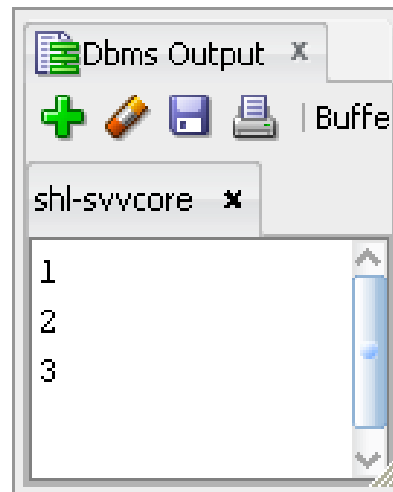
```
    end loop;
```

```
exception
```

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

```
end;
```

```
/
```



Массивы переменной длины

```
-- 13/1b.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));
```

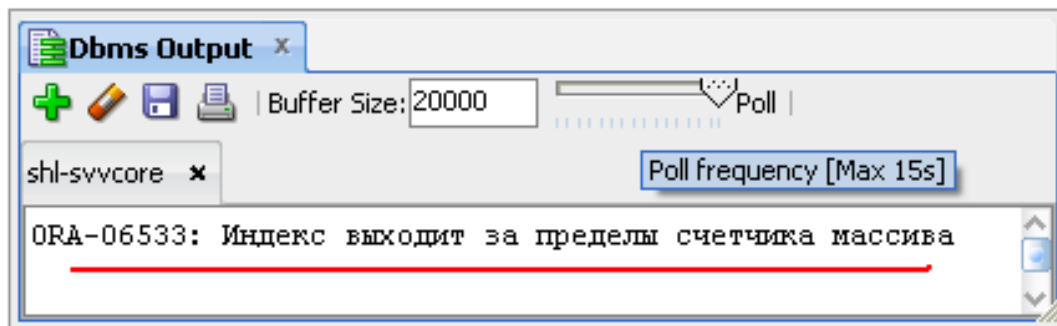
```
    end loop;
```

```
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;
```

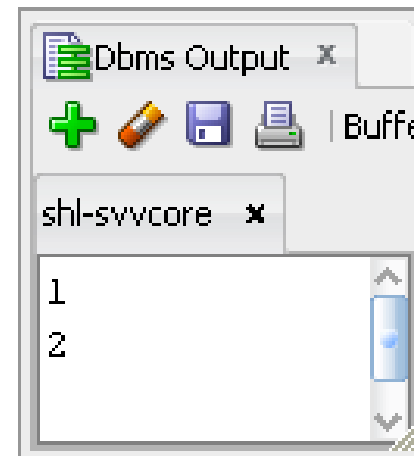
```
    end loop;
```

```
    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

```
-- 13/18.sql
declare
    type myarraytype is varray(3) of number not null;
    va myarraytype := myarraytype();

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

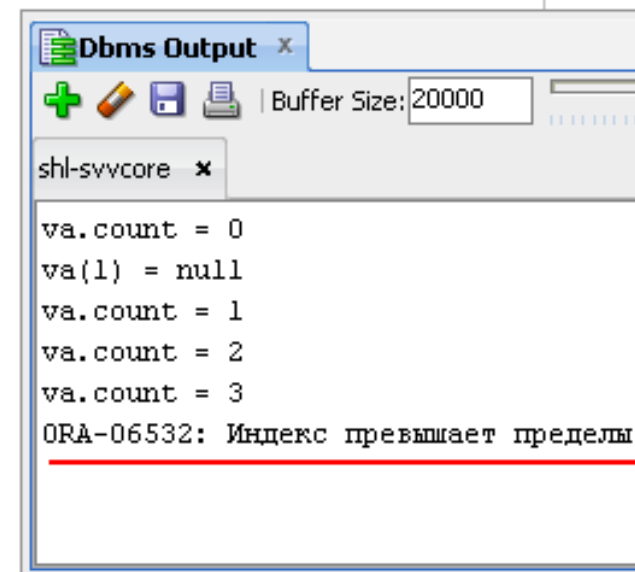
    va.extend;

    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);

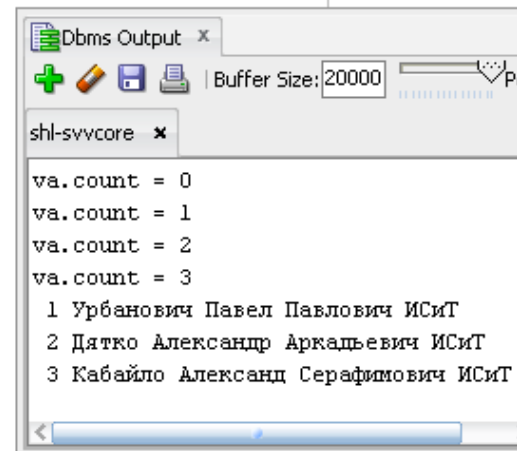
    va.extend;
    dbms_output.put_line('va.count = ' || va.count);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



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 Александр 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;
```



```
Dbms Output x
+  Buffer Size: 20000
shl-svvcare x
va.count = 0
va.count = 1
va.count = 2
va.count = 3
1 Урбанович Павел Павлович ИСиТ
2 Дятко Александр Аркадьевич ИСиТ
3 Кабайло Александр Серафимович ИСиТ
```

Вложенные таблицы

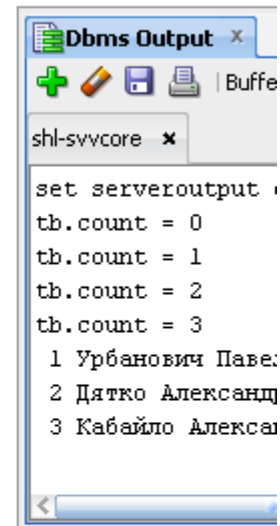
- ▶ Вложенные таблицы — одномерные, несвязанные коллекции однотипных элементов
- ▶ Доступны в рамках PL/SQL и как поля таблицы в БД
- ▶ Изначально являются плотными, но могут впоследствии становиться разреженными



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 Александр 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;
```



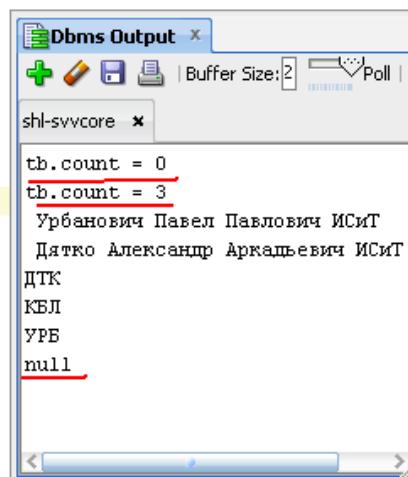
Ассоциативные массивы

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



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 Алексанц 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;
```



```
Dbms Output x
+ + + + + Buffer Size: 2 Poll
shl-svcore x
tb.count = 0
tb.count = 3
Урбанович Павел Павлович ИСИТ
Дятко Александр Аркадьевич ИСИТ
ДТК
КБЛ
УРБ
null
```

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?



Вопросы?

