

ORACLE 12c

PL/SQL

Лекция 10

Характеристика языка PL/SQL

- ▶ Procedural Language extensions to SQL;
- ▶ Основной язык для программирования хранимых процедур (stored procedures);
- ▶ Интегрирован с базой данных Oracle;
- ▶ Производительность серверных модулей;
- ▶ Приложение может быть проще в реализации при написании бизнес-логики на основе хранимых процедур;
- ▶ Отсутствие накладных расходов на приведение типов;
- ▶ Может выполняться независимо от пользователя;
- ▶ PL/SQL-функции можно вызывать из SELECT запросов

Характеристика языка PL/SQL

- ▶ Взаимодействие с пользователем (user interaction);
- ▶ Внутренний язык (proprietary for Oracle);
- ▶ Содержит элементы объектно-ориентированного программирования;
- ▶ Позволяет использовать объектные типы;
- ▶ Интерпретация (режим по умолчанию);
- ▶ Компиляция (промежуточный код на С и конечный объектный код процессора);
- ▶ Среда выполнения: SQL* Plus, SQL Developer, TOAD.



DBMS_output

- ▶ Модуль DBMS_OUTPUT обеспечивает вывод информации для отладки
- ▶ Владелец – пользователь SYS.
- ▶ Принципы работы модуля DBMS_OUTPUT следующий:
 - ▶ Операция PUT берет свои аргументы и помещает во внутренний буфер для хранения.
 - ▶ Операция GET считывает этот буфер и возвращает его содержимое процедуре в качестве аргумента.
 - ▶ Размер буфера устанавливается с помощью процедуры ENABLE.
- ▶ DBMS_OUTPUT.put_line();



DBMS_output

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL worksheet with a PL/SQL script named `xx.sql`. The script declares 11 variables (`n1` through `n11`) and uses `dbms_output.put_line` to print their values. The script is executed, and the output is displayed in the `Dbms Output` window. The output shows the values of the variables, with `n11` displaying a scientific notation value.

SQL Script (xx.sql):

```
1  -- 0xx.sql
2
3  declare
4      n1 number(20,11) := 123456789.12345678911;
5      n2 number(20,10) := 123456789.12345678911;
6      n3 number(20) := 123456789.12345678911;
7      n4 number(20,-1) := 123456789.12345678911;
8      n5 number(20,-2) := 123456789.12345678911;
9      n6 number(20,-21) := 123456789.12345678911;
10     n7 number(20,-25) := 123456789.12345678911;
11     n8 number(20,-83) := 123456789.12345678911;
12     n9 number(20,-7) := 0.12345678911;
13     n10 number(38,10) := 123456789.12345678911;
14     n11 number(38,10) := 123456789E-12;
15
16 begin
17     dbms_output.put_line('n1 =' || n1);
18     dbms_output.put_line('n2 =' || n2);
19     dbms_output.put_line('n3 =' || n3);
20     dbms_output.put_line('n4 =' || n4);
21     dbms_output.put_line('n5 =' || n5);
22     dbms_output.put_line('n6 =' || n6);
23     dbms_output.put_line('n7 =' || n7);
```

Dbms Output:

```
n1 =123456789,12345678911
n2 =123456789,1234567891
n3 =123456789
n4 =123456790
n5 =123456800
n6 =0
n7 =0
n8 =0
n9 =0
n10 =123456789,1234567891
n11 =,0001234568
```

Script Output:

```
anonymous block completed
anonymous block completed
anonymous block completed
anonymous block completed
anonymous block completed
anonymous block completed
```

Log:

```
Log
```

Connections:

- shl
- shl-svvcore
 - Tables (Filtered)
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Queues
 - Queues Tables
 - Triggers
 - Types
 - Sequences
 - Materialized Views
 - Materialized Views Logs
 - Synonyms
 - Public Synonyms
 - Database Links
 - Public Database Links
 - Directories
 - Java
 - XML Schemas
 - XML DB Repository
 - Recycle Bin
 - Jobs
 - Other Users

Versioning Navigator:

- Subversion

Status Bar: Saved: D:\sql\Oracle\10\xx.sql | Line 14 Column 22 | Insert | Windows: CR/LF Editing

Схема блока PL/ SQL

Заголовок

IS

Раздел объявлений

BEGIN

Исполняемый раздел

EXCEPTION

Раздел исключений

END;



Схема блока PL/ SQL

```
PROCEDURE get_happy (ename_in IN VARCHAR2)
IS
  l_hiredate DATE;
BEGIN
  l_hiredate := SYSDATE - 2;
  INSERT INTO employee
    (emp_name, hiredate)
  VALUES (ename_in, l_hiredate);
EXCEPTION
  WHEN DUP_VAL_IN_INDEX
  THEN
    DBMS_OUTPUT.PUT_LINE
      ('Cannot insert.');
```

Заголовок

Раздел объявлений

Исполняемый раздел

Раздел исключений

END;



Анонимный блок PL/ SQL

- ▶ Не имеет секции заголовка
- ▶ Используется как скрипт для выполнения PL/SQL выражений
- ▶ Не может быть вызван из другого блока
- ▶ Начинается с DECLARE или BEGIN
- ▶ Варианты использования:
 - ▶ Триггер на стороне клиента (Oracle Development Tools)
 - ▶ Триггер базы данных (содержит АБ)
 - ▶ SQL-скрипт (описание процедур, функций и execute)
 - ▶ Откомпилированная программа (блок в execute команде, выполняющейся на сервере)



Анонимный блок PL/SQL

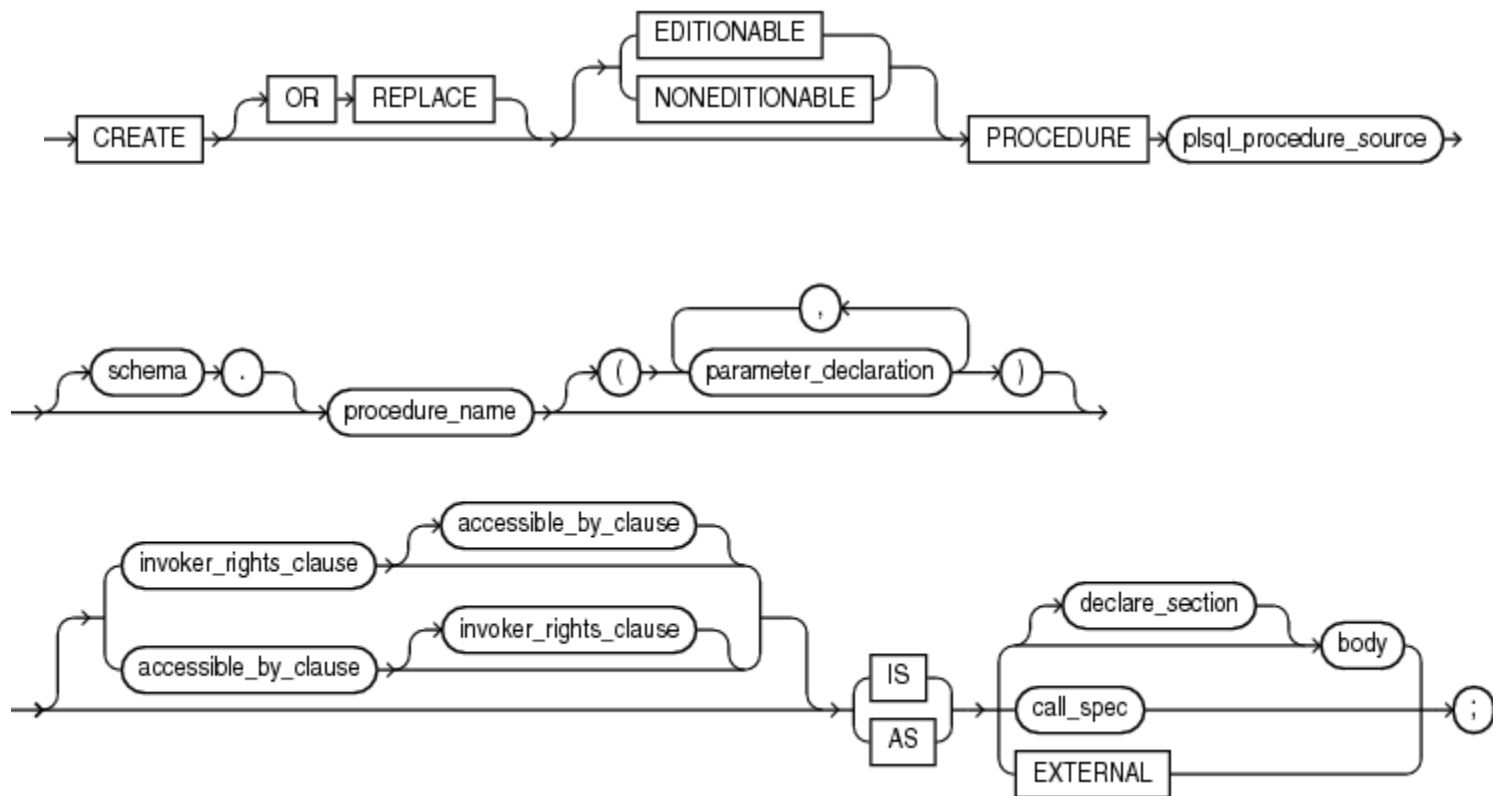
```
-- 01.sql
-- простейший анонимный блок PL/SQL
begin -- начало блока
    -- начало секции выполняемых команд (EXECUTION section)
    null; --
    -- конец секции выполняемых команд
end; -- конец блока
/
```

```
-- 02.sql
-- PL/SQL не чувствителен к регистру
-- простейший анонимный блок PL/SQL
BEGIN -- начало блока
    -- начало секции выполняемых команд (EXECUTION section)
    null; --
    -- конец секции выполняемых команд
eND; -- конец блока
/
```



Именованные блоки PL/ SQL

- ▶ Процедуры
- ▶ Функции



Секция объявления

```
-- 03.sql
declare                                -- необязательная секция объявления
    x number(3) := 3;                    -- объявление переменной (область видимости блок)
begin
    dbms output.put line('x = ' || x);    -- стандартный пакет процедур dbms_output
                                           -- процедура put_line - вывод ( serveroutput on)
                                           -- || конкатенация + неявное преобразование
end;
/
```



Секция исключительных ситуаций

```
-- 04.sql
declare
  x number(3) := 3;
  y number(3) := 0;
  z number(10,2);
begin
  dbms_output.put_line(
    'x = ' || x ||
    ', y = ' || y
  );

  z := x/y; -- деление на 0
  dbms_output.put_line('z = ' || z); -- не выполняется
  exception -- секция исключительных ситуаций
    when others -- когда любая искл. ситуация
    then dbms_output.put_line('error = ' || sqlerrm);
    -- sqlerrm - зарезервированное имя переменной окружения

end;
/
```

```
System.out.println(
  "x=3, y=0
  Error = ORA-01476: делитель равен нулю
  ");
```

Sqlerrm и sqlcode

- ▶ Функция SQLERRM возвращает сообщение об ошибке, связанной с исключительной ситуацией
- ▶ Функция SQLCODE возвращает номер ошибки, связанной с исключительной ситуацией
- ▶ Могут быть использована только в разделе обработки исключений
- ▶ Не имеют параметров или аргументов



Секция исключительных ситуаций

- ▶ Может содержать столько блоков `WHEN`, сколько выделяется обрабатываемых исключений
- ▶ Остальные – в `WHEN OTHERS`
- ▶ Можно определять свои исключения

```
EXCEPTION
  WHEN exception_name
  THEN
    операторы обработки ошибки;
  . . .
  [WHEN OTHERS
  THEN
    обработка исключения по умолчанию;]
```



Вложенные блоки

- ▶ Область действия (scope) – переменные, исключения, модули – локальны в рамках блока;
- ▶ Область видимости – в текущем блоке;

```
-- 05.sql
declare
    x number(3) := 3;
    y number(3) := 0;
    z number(10,2);
begin
    dbms_output.put_line('x = '||x||', y = '||y );
    begin
        z := x/y;
    exception
        when others
        then dbms_output.put_line('error = '||sqlerrm);
    end;
    dbms_output.put_line('z = '||z);
end;
/
```

```
x=3, y=0
Error = ORA-01476: делитель равен нулю
z=
```

Предупреждения компилятора

- ▶ ALL (все);
- ▶ PERFORMANCE (производительность);
- ▶ INFORMATIONAL (информационные);
- ▶ SEVERE (логика программы);
- ▶ Specific error (ошибка);



Предупреждения компилятора

```
show parameter plsql_warnings;
```

Script Output x Query Result x		
Task completed in 0,046 seconds		
NAME	TYPE	VALUE
plsql_warnings	string	DISABLE:ALL

```
select name, value from v$parameter where name = 'plsql_warnings';
```

Script Output x Query Result x	
All Rows Fetched: 1 in 0,016 seconds	
NAME	VALUE
1 plsql_warnings	DISABLE:ALL

Предупреждения компилятора

```
alter system set plsql_warnings = 'ENABLE:INFORMATIONAL';
```

```
select name, value from v$parameter where name = 'plsql_warnings';
```

Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0,016 seconds	
NAME	VALUE
1 plsql_warnings	ENABLE:INFORMATIONAL, DISABLE:PERFORMANCE, DISABLE:SEVERE

```
SELECT DBMS_WARNING.GET_WARNING_SETTING_STRING FROM DUAL;
```

Script Output x Query Result x	
SQL All Rows Fetched: 1 in 0 seconds	
GET_WARNING_SETTING_STRING	
1	ENABLE:INFORMATIONAL, DISABLE:PERFORMANCE, DISABLE:SEVERE

Идентификаторы

- ▶ Идентификатор – наименование объекта PL/SQL
 - ▶ Константы
 - ▶ Скалярные переменные
 - ▶ Составные переменные:
 - ▶ Структуры
 - ▶ Коллекции
 - ▶ Исключения
 - ▶ Пакеты, процедуры и функции
 - ▶ Типы
 - ▶ Курсоры
 - ▶ Зарезервированные слова
 - ▶ Метки



Идентификаторы

- ▶ Не более 30 символов
- ▶ Начинается с буквы
- ▶ Не содержит пробелов
- ▶ Может включать \$ _ #
- ▶ Компилятор приводит идентификаторы к верхнему регистру
- ▶ “идентификатор” регистрозависим



Спецсимволы и зарезервированные слова PL/SQL

```
1 select keyword from v_$reserved words  
2* where length = 1 and keyword != 'A'  
3 /
```

KEYWORD

+
<
^
P
,
M
/
!
:
G
@

KEYWORD

=
.
&
T
U
-
[
|
>
]
K

KEYWORD

*
)
(
E



Спецсимволы и зарезервированные слова PL/SQL

```
1 select keyword from v_$reserved_words
2*  where length > 1 and keyword != 'A' order by keyword
SYS@sh1> /
```

KEYWORD

<<
ABORT
ACCESS
ACCESSED
ACCOUNT
ACTIVATE
ADD
ADMIN
ADMINISTER
ADMINISTRATOR
ADVISE

KEYWORD

ADVISOR
AFTER
ALIAS
ALL
ALLOCATE
ALLOW
ALL_ROWS
ALTER
ALWAYS
ANALYZE
ANCTI I ARY

WRAPPED
WRITE
<_DYN_PRUNE
<ID
<MLATTRIBUTES
<MLCOLATTVAL
<MLELEMENT
<MLFOREST
<MLNAMESPACES

KEYWORD

<MLPARSE
<MLPI
<MLQUERY
<MLROOT
<MLSCHEMA
<MLSERIALIZE
<MLTABLE
<MLTYPE
YEAR
YES
ZONE

1100 строк выбрано.

SYS@sh1>

Литералы

- ▶ Литерал – значение идентификатора
 - ▶ Number – 123, 21.6, NULL
 - ▶ String – 'sentence', '01-01-2017' , NULL
 - ▶ Boolean – true, false, NULL
 - ▶ ANSI date – DATE '2016-11-01'



Метки

- ▶ Метка – способ именовать определенную часть программы
- ▶ Синтаксис << label>>
- ▶ Используется для:
 - ▶ Именованная блока на время выполнения
 - ▶ Улучшение читаемости кода
 - ▶ Необходимость ссылаться во вложенном блоке на переменную с таким же именем из внешнего блока
 - ▶ Для перехода по GOTO



Символы специального значения

;	Разделитель выражений
-- /* abcd */	Однострочный и многострочный комментарий
% и _	Множественный и одиночный групповой символ в Like
@	Индикатор удаленного объекта
<>	Не равно
!=	
^=	
~=	
	Конкатенация
<<label>>	Метка
<= и >=	Меньше или равно / Больше или равно
:=	Присваивание



Типы данных ORACLE - СИМВОЛЬНЫЕ

CHAR	Символьное поле фиксированной длины до 2000 байт
NCHAR	Поле фиксированной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 2000 символов или 2000 байт в зависимости от набора символов.
VARCHAR2	Символьное поле переменной длины до 4000 байт
NVARCHAR2	Поле переменной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 4000 символов или 4000 байт в зависимости от набора символов.

Типы данных ORACLE – символные

LONG	Символьный, переменной длины, до 2GB, оставлен для совместимости
RAW(n)	Переменной длины, для бинарных данных $n \leq 2000$ byte оставлен для совместимости
LONG RAW	Бинарные данные до 2GB
CLOB	Символьный тип большой объект до 4GB
NLOB	CLOB для многобайтных символов
BLOB	Большой двоичный объект до 4GB
BFILE	Указатель на двоичный файл операционной системы



Типы данных ORACLE – дата/время

DATE	7 байтовое поле фиксированной длины, используемое для хранения даты и времени
INTERVAL DAY TO SECOND	11 байтовое поле фиксированной длины для интервала времени: Дни, часы, минуты, секунды
INTERVAL YEAR TO MONTH TIMESTAMP	5 байтовое поле фиксированной длины для интервала времени: Годы и месяцы
TIMESTAMP WITH TIME ZONE	13 байтовое поле фиксированной длины Дата, время и настройки, связанные с часовым поясом.
TIMESTAMP WITH LOCAL TIME	7-11 байтовое поле переменной длины Дата и время, приведенные к часовому поясу базы данных



Типы данных ORACLE – числовые

NUMBER(n, s)	Числовой тип переменной длины Точность n \leq 38, общее количество цифр Масштаб s = [-84, 127], количество цифр после запятой
--------------	---



Неявные преобразования типов данных

VARCHAR2 CHAR	DATE
DATE	VARCHAR2
VARCHAR2 CHAR	ROWID
ROWID	VARCHAR2
VARCHAR2 CHAR	NUMBER
NUMBER	VARCHAR2



Character Set

- ▶ NLS - National Language Support, далее Globalization Support
- ▶ Можно хранить данные множества национальных языков, используя Unicode или специальные кодировки – наборы символов (character set)
- ▶ Символы хранятся как коды символов, зависящие от выбранного набора символов
- ▶ В одной БД могут использоваться два набора символов: основной (database character set) и дополнительный (national character set)
- ▶ Устанавливаются при создании БД
- ▶ Изменяются alter database (national) character set



Поддержка национальных языков

- ▶ Основной набор символов используется для:
 - ▶ хранения символьных типов `char`, `varchar2`, `clob` и `long`
 - ▶ описания имен объектов, переменных
 - ▶ Ввода и хранения PL/SQL модулей
- ▶ Дополнительный набор символов используется для:
 - ▶ хранения символьных типов `nchar`, `nvarchar2`, `nclob`
- ▶ Кроме символов алфавита в набор включаются знаки препинания, числа, символы денежных единиц и пр.



Поддержка национальных языков

- ▶ <region> <number of bits used to represent a character>
<standard character set name> [SIC]

WE8ISO8859P1

Western European 8-bit ISO WE (Western Europe)
8859 Part 1

8



Поддержка национальных языков

- ▶ Переменная окружения `NLS_LANG`:
 - ▶ `NLS_LANG = language_territory.charset`
 - ▶ Язык (`LANGUAGE`) – имена месяцев, имена дней, направление текста, сокращения для времени и дат. По умолчанию `AMERICAN`
 - ▶ Территория (`TERRITORY`) – настройки календаря, формат даты, формат денежной единицы. Если не указан, то будет взято значение, соответствующее языку (для `RUSSIAN - CIS`)
 - ▶ Набор символов (`CHARACTER SET`) – отображение символов, отображение и конвертация заглавных букв, порядок замещения символов при преобразовании. Каждому языку поставлен в соответствие набор символов по умолчанию
-



Поддержка национальных языков

- ▶ Представления словаря:
- ▶ NLS_SESSION_PARAMETERS
- ▶ NLS_INSTANCE_PARAMETERS
- ▶ NLS_DATABASE_PARAMETERS



Семантика символов

- ▶ Байтовая семантика рассматривает строки как последовательность байтов
- ▶ Символьная семантика рассматривает строки как последовательность символов
- ▶ Задается параметром `NLS_LENGTH_SEMANTICS`
- ▶ По умолчанию - `BYTE`
- ▶ Можно задавать семантику для столбца:
 - ▶ `VARCHAR2(20 BYTE)`
 - ▶ `VARCHAR2(10 CHAR)`



Семантика символов

```
sh1 - SYSTEM - 09.11.10  
SYSTEM@sh1> show parameter nls
```

NAME_COL_PLUS_SHOW_PARAM	TYPE	VALUE_CO
nl_alendar	string	
nl_comp	string	
nl_currency	string	
nl_date_format	string	
nl_date_language	string	
nl_dual_currency	string	
nl_iso_currency	string	
nl_language	string	AMERICAN
nl_length_semantics	string	BYTE
nl_nchar_conv_excp	string	FALSE
nl_numeric_characters	string	

NAME_COL_PLUS_SHOW_PARAM	TYPE	VALUE_CO
nl_sort	string	
nl_territory	string	AMERICA
nl_time_format	string	
nl_timestamp_format	string	
nl_timestamp_tz_format	string	
nl_time_tz_format	string	

```
SYSTEM@sh1>
```

Типы данных PL/ SQL

- ▶ Скалярные (Scalar)
- ▶ Ссылочные (Reference)
- ▶ Составные (Composite)
- ▶ Большие объекты (LOB)



Скалярные типы данных

- ▶ символ/строка
- ▶ число
- ▶ булев
- ▶ дата/время



Символы/строки

```
-- 06.sql
-- символы/строки
declare
  c1 char (5);           -- Символьный тип фиксированной длины 5 байт
  c2 char (5 char);     -- Символьный тип фиксированной длины 5 символов
  c3 nchar (5);          -- Национальный символьный тип фиксированной длины 5 байт
  c4 nchar (5 char);     -- Национальный символьный тип фиксированной длины 5 символов

begin
  dbms_output.put_line('c1 = '||c1||', length(c1)=|| length(c1)); -- размер строки в символах
  dbms_output.put_line('c2 = '||c2||', length(c2)=|| length(c2));
  dbms_output.put_line('c3 = '||c3||', length(c3)=|| length(c3));
  dbms_output.put_line('c4 = '||c4||', length(c4)=|| length(c4));
  c1 := 'hhh';
  c2 := 'mmmm';
  c3 := 'hhh';
  c4 := 'mmmm';
  dbms_output.put_line('c1 = '||c1||', length(c1)=|| length(c1));
  dbms_output.put_line('c2 = '||c2||', length(c2)=|| length(c2));
  dbms_output.put_line('c3 = '||c3||', length(c3)=|| length(c3));
  dbms_output.put_line('c4 = '||c4||', length(c4)=|| length(c4));

  c1 =, length(c1)=
  c2 =, length(c2)=
  c3 =, length(c3)=
  c4 =, length(c4)=
  c1 =hhh , length(c1)=5
  c2 =mmmm , length(c2)=5
  c3 =hhh , length(c3)=5
  c4 =mmmm , length(c4)=5

end;
/
```


Символы/строки

```
-- 07.sql
-- СИМВОЛЫ/строки
declare
  c1 varchar (15);           -- симв. SQL-тип (ANSI) переменной длины 15 байт  max 32K (БД:4K)
  c2 varchar (15 char);     -- симв. SQL-тип (ANSI) переменной длины 15  символов
  c3 varchar2 (15);         -- синоним varchar
  c4 nvarchar2 (15 char);  -- национальный симв. тип переменной длины 15  символов

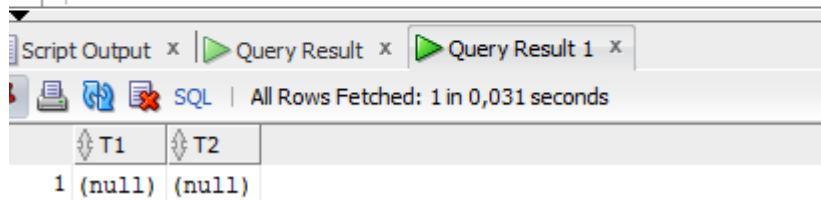
begin
  dbms_output.put_line('c1 = '||c1||', length(c1)= '|| length(c1)); -- размер строки в символах
  dbms_output.put_line('c2 = '||c2||', length(c2)= '|| length(c2));
  dbms_output.put_line('c3 = '||c3||', length(c3)= '|| length(c3));
  dbms_output.put_line('c4 = '||c4||', length(c4)= '|| length(c4));
  c1 := '1234567';
  c2 := '1234567';
  c3 := '1234567';
  c4 := '1234567';
  dbms_output.put_line('c1 = '||c1||', length(c1)= '|| length(c1));
  dbms_output.put_line('c2 = '||c2||', length(c2)= '|| length(c2));
  dbms_output.put_line('c3 = '||c3||', length(c3)= '|| length(c3));
  dbms_output.put_line('c4 = '||c4||', length(c4)= '|| length(c4));

end;
/
```

c1 =, length(c1)=
c2 =, length(c2)=
c3 =, length(c3)=
c4 =, length(c4)=
c1 =1234567, length(c1)=7
c2 =1234567, length(c2)=7
c3 =1234567, length(c3)=7
c4 =1234567, length(c4)=7

Символы/строки

```
CREATE TABLE TEST (T1 CHAR(3), T2 VARCHAR2(3));  
INSERT INTO TEST VALUES ('', '');  
SELECT T1, T2 FROM TEST WHERE T1 IS NULL;  
SELECT T1, T2 FROM TEST WHERE T2 IS NULL;
```



The screenshot shows the SQL Developer interface. At the top, there are tabs for 'Script Output', 'Query Result', and 'Query Result 1'. Below the tabs, a status bar indicates 'All Rows Fetched: 1 in 0,031 seconds'. The main area displays a table with two columns, 'T1' and 'T2'. The first row shows both columns as '(null)'.

T1	T2
1 (null)	(null)

```
DECLARE VCHR VARCHAR2(3) :='';  
BEGIN  
  IF VCHR IS NULL THEN DBMS_OUTPUT.PUT_LINE('VARCHAR2 TRUE');  
  END IF;  
END;  
  
DECLARE CHR CHAR(3) :='';  
BEGIN  
  IF CHR IS NULL  
  THEN DBMS_OUTPUT.PUT_LINE('CHAR TRUE');  
  ELSE DBMS_OUTPUT.PUT_LINE('CHAR FALSE');  
  END IF;  
END;
```

VARCHAR2 TRUE

CHAR FALSE

Числовые типы данных

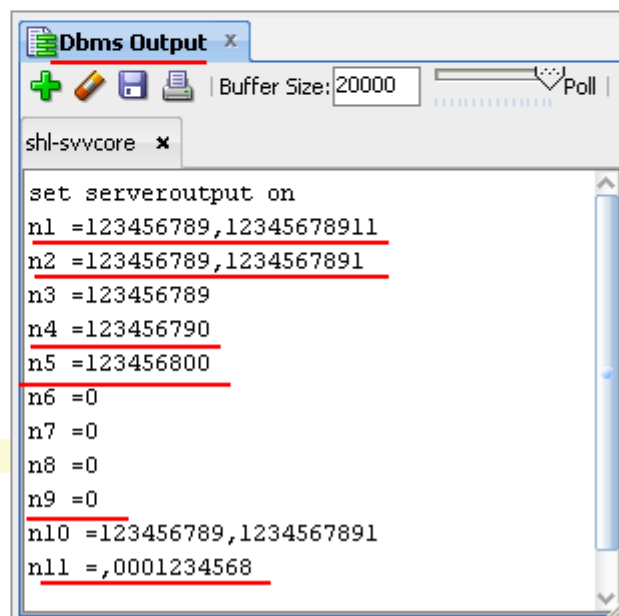
```
-- 11/01.sql
-- number (n,s)          n=[0,38], s=[-84,127]
declare
  n1 number(20,11) := 123456789.12345678911; --
  n2 number(20,10) := 123456789.12345678911;
  n3 number(20) := 123456789.12345678911;
  n4 number(20,-1) := 123456789.12345678911;
  n5 number(20,-2) := 123456789.12345678911;
  n6 number(20,-21) := 123456789.12345678911;
  n7 number(20,-25) := 123456789.12345678911;
  n8 number(20,-83) := 123456789.12345678911;
  n9 number(20,-7) := 0.12345678911;
  n10 number(38,10) := 123456789.12345678911;
  n11 number(38,10) := 123456789E-12;
```

begin

```
dbms_output.put_line('n1 = '||n1);
dbms_output.put_line('n2 = '||n2);
dbms_output.put_line('n3 = '||n3);
dbms_output.put_line('n4 = '||n4);
dbms_output.put_line('n5 = '||n5);
dbms_output.put_line('n6 = '||n6);
dbms_output.put_line('n7 = '||n7);
dbms_output.put_line('n8 = '||n8);
dbms_output.put_line('n9 = '||n9);
dbms_output.put_line('n10 = '||n10);
dbms_output.put_line('n11 = '||n11);
```

end;

/



```
set serveroutput on
n1 =123456789,12345678911
n2 =123456789,1234567891
n3 =123456789
n4 =123456790
n5 =123456800
n6 =0
n7 =0
n8 =0
n9 =0
n10 =123456789,1234567891
n11 =,0001234568
```

Числовые типы данных

```
-- 11/02.sql
```

```
declare
```

```
  n1 binary_float      := 123456789.12345678911;      -- IEEE-754
```

```
  n2 binary_double    := 123456789.12345678911;      -- IEEE-754
```

```
  n3 binary_float      := 123456.12345678911;
```

```
  n4 binary_double     := 123456.12345678911;
```

```
begin
```

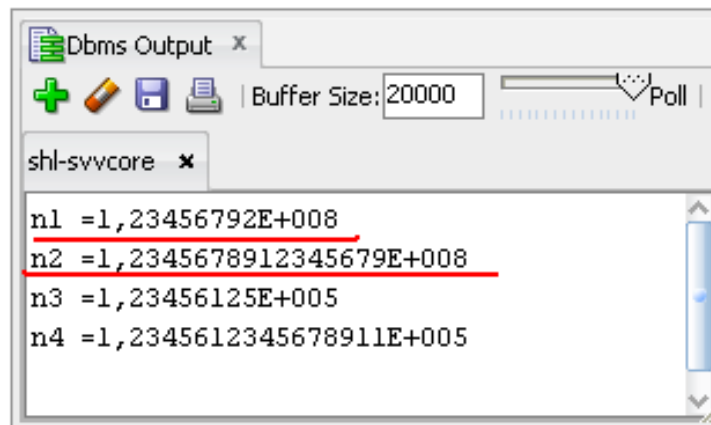
```
  dbms_output.put_line('n1 = ' || n1);
```

```
  dbms_output.put_line('n2 = ' || n2);
```

```
  dbms_output.put_line('n3 = ' || n3);
```

```
  dbms_output.put_line('n4 = ' || n4);
```

```
end;
```



```
shl-svvcare x
n1 =1,23456792E+008
n2 =1,2345678912345679E+008
n3 =1,23456125E+005
n4 =1,2345612345678911E+005
```

Числовые типы данных

-- 11/03.sql

declare

```
n1 binary_integer := 123456.12345678911; -- [-2^31-1, 2^31-1] не БД
n2 pls integer := 123; -- [-2^31-1, 2^31-1] не БД
n3 natural := 0; -- [0, 2^31-1] не БД
n4 naturaln := 0; -- [0, 2^31-1] null не БД
n5 positive := 123; -- [1, 2^31-1] не БД
n6 positiven := 123; -- [1, 2^31-1] null не БД
n7 signtype := - 0.5; -- ]-1,1[ ---> -1,0,+1
```

begin

```
dbms_output.put_line('n1 = '||n1);
dbms_output.put_line('n2 = '||n2);
dbms_output.put_line('n3 = '||n3);
dbms_output.put_line('n4 = '||n4);
dbms_output.put_line('n5 = '||n5);
dbms_output.put_line('n6 = '||n6);
dbms_output.put_line('n7 = '||n7);
n3 := -1;
```

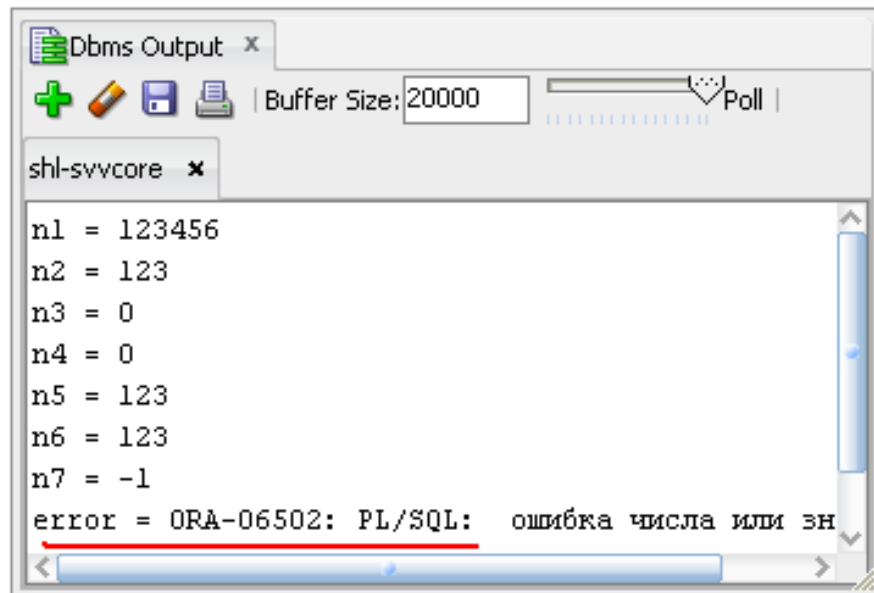
exception

when others

then

```
dbms_output.put_line('error = '||sqlerrm);
```

end;



```
Dbms Output x
+ | Buffer Size: 20000 | Poll |
shl-svcore x
n1 = 123456
n2 = 123
n3 = 0
n4 = 0
n5 = 123
n6 = 123
n7 = -1
error = ORA-06502: PL/SQL: ошибка числа или зн.
```

Булев тип

```
-- 11/04.sql
```

```
declare
```

```
  b1 boolean := true;
```

```
  b2 boolean := false;
```

```
  b3 boolean ;
```

```
begin
```

```
  if b1      then dbms_output.put_line('b1 = '||'true'); end if;
```

```
  if not b1    then dbms_output.put_line('b1 = '||'false'); end if;
```

```
  if b2       then dbms_output.put_line('b2 = '||'true');  end if;
```

```
  if not b2    then dbms_output.put_line('b2 = '||'false'); end if;
```

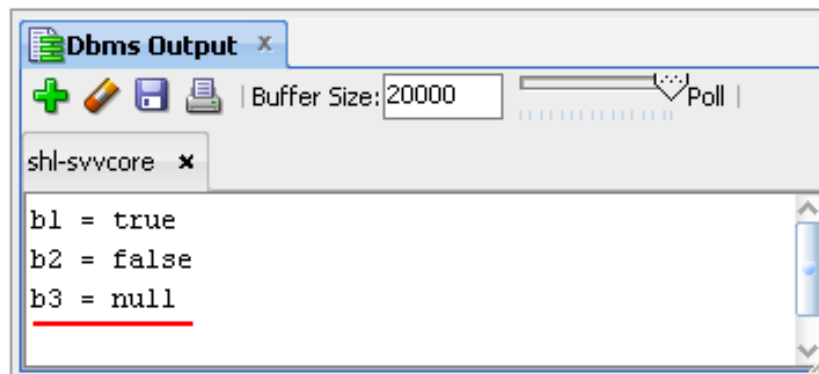
```
  if b3       then dbms_output.put_line('b3 = '||'true');  end if;
```

```
  if not b3    then dbms_output.put_line('b3 = '||'false'); end if;
```

```
  if b3 is null then dbms_output.put_line('b3 = '||'null'); end if;
```

```
end;
```

```
/
```



Дата и время

```
-- 11/05.sql
```

```
declare
```

```
  d1 date := sysdate;
```

```
  d2 date := to_date('02.12.2000', 'DD.MM.YYYY') ;
```

```
  d3 date := to_date('02.12.2000 20:21:17', 'DD.MM.YYYY HH24:MI:SS') ;
```

```
begin
```

```
  dbms_output.put_line('d1 = ' || d1);
```

```
  dbms_output.put_line('d2 = ' || d2);
```

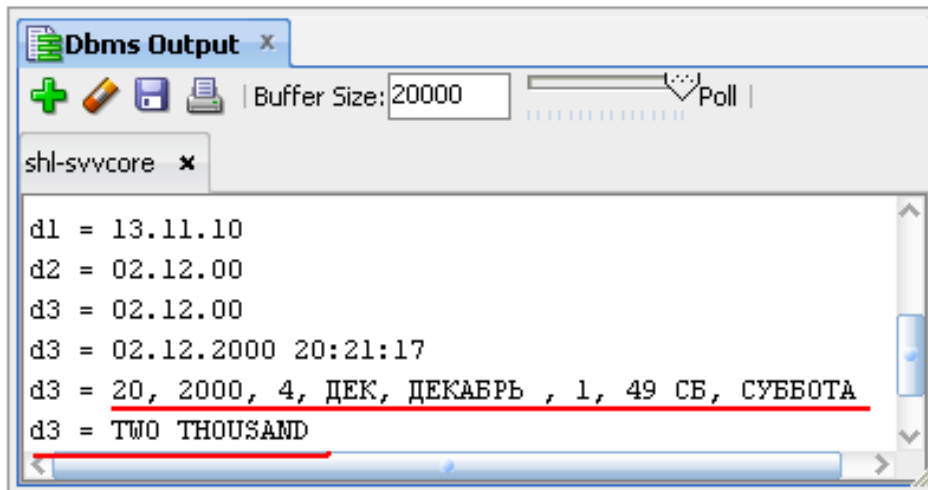
```
  dbms_output.put_line('d3 = ' || d3);
```

```
  dbms_output.put_line('d3 = ' || to_char(d3, 'DD.MM.YYYY HH24:MI:SS'));
```

```
  dbms_output.put_line('d3 = ' || to_char(d3, 'CC, YYYY, Q, MON, MONTH, W, WW DY, DAY'));
```

```
  dbms_output.put_line('d3 = ' || to_char(d3, 'YEAR'));
```

```
end;
```



The screenshot shows a window titled "Dbms Output" with a tab labeled "shl-svvcore". The window contains a list of output lines from the SQL script execution. The first three lines are "d1 = 13.11.10", "d2 = 02.12.00", and "d3 = 02.12.00". The fourth line is "d3 = 02.12.2000 20:21:17". The fifth line is "d3 = 20, 2000, 4, ДЕК, ДЕКАБРЬ, 1, 49 СБ, СУББОТА". The sixth line is "d3 = TWO THOUSAND". The text in the window is monospaced and black on a white background. There are icons for adding, editing, saving, and printing at the top left, and a "Buffer Size: 20000" and "Poll" button at the top right.

```
Dbms Output x
+  Buffer Size: 20000 Poll
shl-svvcore x
d1 = 13.11.10
d2 = 02.12.00
d3 = 02.12.00
d3 = 02.12.2000 20:21:17
d3 = 20, 2000, 4, ДЕК, ДЕКАБРЬ, 1, 49 СБ, СУББОТА
d3 = TWO THOUSAND
```

Дата и время

declare

```
d1 timestamp(3) := systimestamp; -- [0,9] дробная часть секунд
d2 timestamp(3) with time zone := systimestamp;
d3 timestamp(5) with local time zone := systimestamp;
d4 timestamp(5) with local time zone := to_timestamp_tz ('02.12.2000 19:21:31 +02:00', 'DD.MM.YYYY HH24:MI:SS TZH:TZM');
d5 timestamp(3) with time zone := to_timestamp_tz ('02.12.2000 19:21:31 EUROPE/MOSCOW MSK', 'DD.MM.YYYY HH24:MI:SS TZR TZD');
```

begin

```
dbms_output.put_line('d1 = '||d1);
dbms_output.put_line('d2 = '||d2);
dbms_output.put_line('d3 = '||d3);
dbms_output.put_line('d3 = '||to_char(d3, 'DD.MM.YYYY HH24:MI:SS'));
dbms_output.put_line('d3 = '||to_char(d3, 'CC, YYYY, Q, MON, MONTH, W, WW DY, DAY, TZR, TZD'));
dbms_output.put_line('d4 = '||to_char(d4, 'DD.MM.YYY HH24:MI:SS'));
dbms_output.put_line('d5 = '||to_char(d5, 'DD.MM.YYYY HH24:MI:SS TZD'));
dbms_output.put_line(to_char(sys_extract_utc(d5), 'DD.MM.YYY HH24:MI:SS'));
dbms_output.put_line(to_char(sys_extract_utc(d2), 'DD.MM.YYY HH24:MI:SS'));
```

end;

/

```
Dbms Output x
+ Buffer Size: 20000 Poll
shl-svvcore x
d1 = 13-НОЯ-10 08.54.15,375 PM
d2 = 13-НОЯ-10 08.54.15,375 PM +02:00
d3 = 13-НОЯ-10 09.54.15,37500 PM
d3 = 13.11.2010 21:54:15
d3 = 21, 2010, 4, НОЯ, НОЯБРЬ, 2, 46 СБ, СУББОТА, EUROPE/MOSCOW, MSK
d4 = 02.12.000 20:21:31
d5 = 02.12.2000 19:21:31 MSK
02.12.000 16:21:31
13.11.010 18:54:15
```


Дата и время

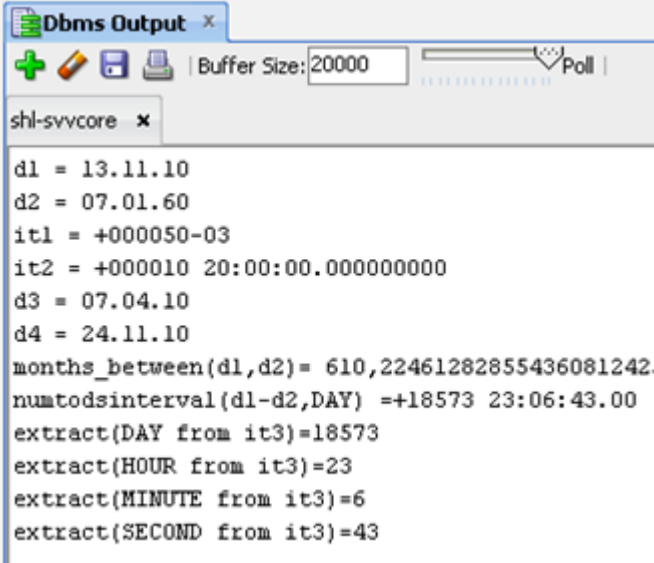
```
SYSTEM@ssh1> select * from sys.v $timezone_names  
2 where upper(tzname) = 'EUROPE/MINSK'  
3 /
```

TZNAME	TZABBREV
Europe/Minsk	LMT
Europe/Minsk	MMT
Europe/Minsk	EET
Europe/Minsk	MSK
Europe/Minsk	CEST
Europe/Minsk	CET
Europe/Minsk	MSD
Europe/Minsk	EEST



Дата и время

```
-- 11/09.sql
declare
  d1 date := sysdate;
  d2 date := to_date('07.01.1960','DD.MM.YYYY');
  it1 interval year(2) to month := interval '50-3' year to month;
  it2 interval day(5) to second(2) := interval '10 20:00:00' day to second ;
  d3 date := d2 + it1;
  d4 date := d1 + it2;
  it3 interval day(5) to second(2);
begin
  dbms_output.put_line('d1 = '||d1);
  dbms_output.put_line('d2 = '||d2);
  dbms_output.put_line('it1 = '||to_char(it1,'YY,MM'));
  dbms_output.put_line('it2 = '||to_char(it2,'YY,MM'));
  dbms_output.put_line('d3 = '||d3);
  dbms_output.put_line('d4 = '||d4);
  dbms_output.put_line('months_between(d1,d2)= '||months_between(d1,d2));
  it3 := numtodsinterval(d1-d2,'DAY');
  dbms_output.put_line('numtodsinterval(d1-d2,DAY) = ' || it3 );
  dbms_output.put_line('extract(DAY from it3)= '||extract(DAY from it3));
  dbms_output.put_line('extract(HOUR from it3)= '||extract(HOUR from it3));
  dbms_output.put_line('extract(MINUTE from it3)= '||extract(MINUTE from it3));
  dbms_output.put_line('extract(SECOND from it3)= '||extract(SECOND from it3));
end;
```



The screenshot shows a 'Dbms Output' window with a tab labeled 'shl-svvcare'. The window contains the following output:

```
d1 = 13.11.10
d2 = 07.01.60
it1 = +000050-03
it2 = +000010 20:00:00.000000000
d3 = 07.04.10
d4 = 24.11.10
months_between(d1,d2)= 610,22461282855436081242.
numtodsinterval(d1-d2,DAY) =+18573 23:06:43.00
extract(DAY from it3)=18573
extract(HOUR from it3)=23
extract(MINUTE from it3)=6
extract(SECOND from it3)=43
```

Константы

```
1  -- 11/10.sql
2  declare
3      n1 constant number (5):=5;
4      d1 constant date := sysdate;
5      n2 pls_integer not null := 25;
6      v1 varchar(25) not null default 'Hello World';
7  begin
8      n1 := 10;
9  exception
10     when others
11     then dbms_output.put_line('error = '|| n1);
12 end;
13 /
14
```

Error starting at line 2 in command:

```
declare
    n1 constant number (5):=5;
    d1 constant date := sysdate;
    n2 pls_integer not null := 25;
    v1 varchar(25) not null default 'Hello World';
begin
    n1 := 10;
exception
    when others
        then dbms_output.put_line('error = '|| n1);
end;
```

Error report:

ORA-06550: Строка 7, столбец 5:

PLS-00363: выражение 'N1' не м.б. использовано как адресат назначения

ORA-06550: Строка 7, столбец 5:

PL/SQL: Statement ignored

06550. 00000 - "line %s, column %s:\n%s"

*Cause: Usually a PL/SQL compilation error.

*Action:

Связанные объявления переменных

- ▶ Тип переменной основан на известной структуре данных
- ▶ Скалярная ссылка %TYPE для определения переменной на основе другой переменной или поля в таблице
- ▶ Ссылка на запись %ROWTYPE для определения структуры записи на основе таблицы или курсора



%TYPE и %ROWTYPE

```
-- 11/11.sql
```

```
declare
```

```
    subject      svvcore.subject.subject%type;
```

```
    pulpit       svvcore.pulpit.pulpit%type;
```

```
    faculty_rec  svvcore.faculty%rowtype;
```

```
begin
```

```
    subject := 'ПИС';
```

```
    pulpit  := 'ИСИТ';
```

```
    faculty_rec.faculty := 'ИДПИ';
```

```
    faculty_rec.faculty_name := 'Факультет издательского дела и полиграфии';
```

```
    dbms_output.put_line(subject);
```

```
    dbms_output.put_line(pulpit);
```

```
    dbms_output.put_line(rtrim(faculty_rec.faculty)||':'||faculty_rec.faculty_name);
```

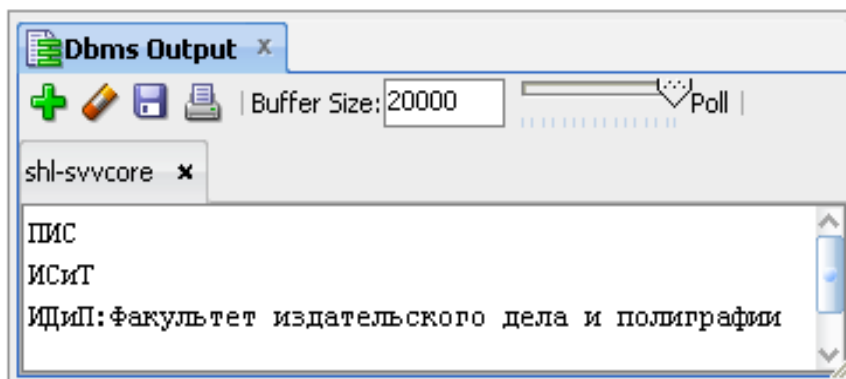
```
exception
```

```
    when others
```

```
        then dbms_output.put_line('error = '|| sqlerrm);
```

```
end;
```

```
/
```



Constraints

```
----- not null assignment
```

```
create table t11 (x date not null);
```

```
-- pl/sql var
```

```
⊞ DECLARE
```

```
    max_available_date DATE NOT NULL :=
```

```
    LAST_DAY(ADD_MONTHS(SYSDATE, 3));
```

```
    last_ship_date max_available_date%TYPE; -- not null variable
```

```
begin
```

```
    null;
```

```
end;
```

```
-- from table
```

```
⊞ DECLARE
```

```
    last_ship_date t11.x%TYPE; -- not null variable
```

```
begin
```

```
    last_ship_date:= to_date('2019-11-11','YYYY-MM-DD');
```

```
    dbms_output.put_line('last_ship_date = '||to_char(last_ship_date, 'YYYY-MM-DD'));
```

```
end;
```



Constraints

Error starting at line : 5 in command -

```
DECLARE
```

```
    max_available_date DATE NOT NULL :=
```

```
    LAST_DAY(ADD_MONTHS(SYSDATE, 3));
```

```
    last_ship_date max_available_date%TYPE; -- not null variable
```

```
begin
```

```
    null;
```

```
end;
```

Error report -

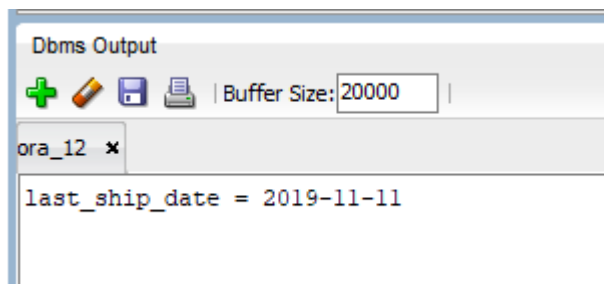
ORA-06550: Строка 4, столбец 20:

PLS-00218: a variable declared NOT NULL must have an initialization assignment

06550. 00000 - "line %s, column %s:\n%s"

*Cause: Usually a PL/SQL compilation error.

*Action:

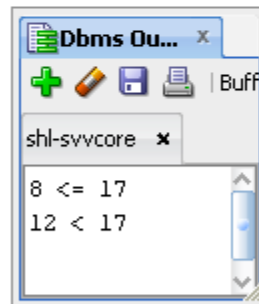


Оператор IF

```
11/10.sql
declare
    x pls_integer := 17;
begin
    if 8 > x
    then
        dbms_output.put_line('8 > '||x);
    end if;
```

```
-----
if 8 > x
then
    dbms_output.put_line('8 > '||x);
else
    dbms_output.put_line('8 <= '||x);
end if;
```

```
-----
if 8 > x
then
    dbms_output.put_line('8 > '||x);
elsif 8 = x
then
    dbms_output.put_line('8 = '||x);
elsif 12 > x
then
    dbms_output.put_line('12 > '||x);
elsif 12 = x
then
    dbms_output.put_line('12 = '||x);
else
    dbms_output.put_line('12 < '||x);
end if;
end;
/
```



Оператор CASE

-- 11/19.sql

declare

 x pls_integer := 17;

begin

case x

when 1 **then** dbms_output.put_line('1');

when 2 **then** dbms_output.put_line('2');

when 3 **then** dbms_output.put_line('3');

else dbms_output.put_line('else');

end case;

case

when 8 > x **then** dbms_output.put_line('8 > ' || x);

when 8 = x **then** dbms_output.put_line('8 = ' || x);

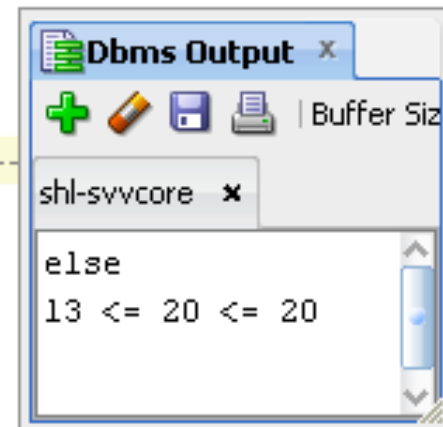
when 12 = x **then** dbms_output.put_line('12 = ' || x);

when x **between** 13 **and** 20 **then** dbms_output.put_line('13 <= ' || x || ' <= 20');

else dbms_output.put_line('else');

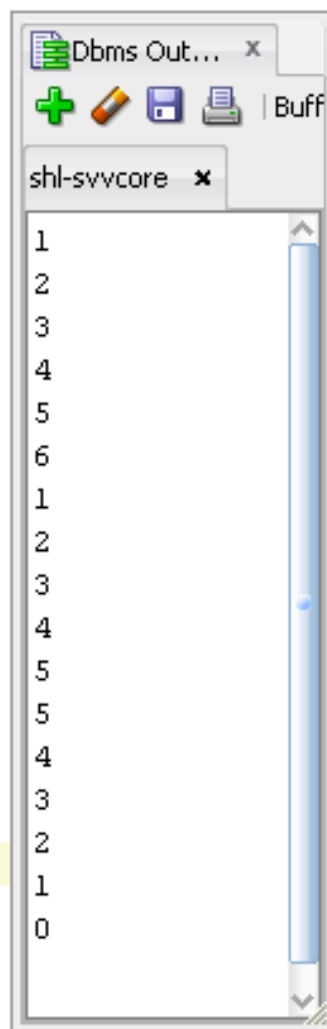
end case;

end;



Циклы loop, for, while

```
-- 11/14.sql
declare
    x pls_integer := 0;
begin
    -----
    loop
        x := x + 1;
        dbms_output.put_line(x);
    exit when x > 5;
    end loop;
    -----
    for k in 1..5
    loop
        dbms_output.put_line(k);
    end loop;
    -----
    while (x > 0)
    loop
        x := x - 1;
        dbms_output.put_line(x);
    end loop;
    -----
end;
/
```



TO_NUMBER()

```
declare
a number;
begin
a := TO_NUMBER('F123.456,78',
               'L999G999D99',
               'NLS_NUMERIC_CHARACTERS=''',''')
|| ' NLS_CURRENCY=''F''')
|| ' NLS_ISO_CURRENCY=FRANCE');

dbms_output.put_line('a = '||a);
end;
```

a = 123456,78

```
SELECT TO_NUMBER('-AusDollars100','L9G999D99',
               ' NLS_NUMERIC_CHARACTERS = ''',''')
       NLS_CURRENCY          = ''AusDollars''
       ')"Amount"
FROM DUAL;
```

	Amount
1	-100

TO_CHAR()

----- S МИНУС И ПЛЮС

```
select to_char(-1234567890, '9999999999S')
  from dual;
select to_char(1234567890, '9999999999S')
  from dual;
```

TO_CHAR(-1234567890,'9999999999S')	
1	1234567890-

TO_CHAR(1234567890,'9999999999S')	
1	1234567890+

----- 99.99 десятичная точка

```
select to_char(0, '99.99')
  from dual;
select to_char(0.1, '99.99')
  from dual;
select to_char(-0.1, '99.99')
  from dual;
```

TO_CHAR(0,'99.99')	
1	.00

TO_CHAR(0.1,'99.99')	
1	.10

```
select to_char(-100, '090.990')
  from dual;
```

----- 090.990 - начальные и конечные нули

```
select to_char(0, '090.990')
  from dual;
select to_char(0.1, '090.990')
  from dual;
select to_char(-0.1, '090.990')
  from dual;
select to_char(-100, '090.990')
  from dual;
```

TO_CHAR(0,'090.990')	
1	000.000

TO_CHAR(0.1,'090.990')	
1	000.100

TO_CHAR(-0.1,'090.990')	
1	-000.100

TO_CHAR(-100,'090.990')	
1	-100.000

TO_CHAR()

```
----- 999,999.99 - разделитель  
select to_char(-1000, '999,999.99')  
from dual;
```

TO_CHAR(10.10,'C999,999.99')	
1	RUB10.10

```
----- B - пробелы по формату  
select to_char(-0.1000, 'B999,999.99')  
from dual;
```

TO_CHAR(-0.1000,'B999,999.99')	
1	-.10

```
----- C - знак валюты  
----- NLS_ISO_CURRENCY  
select to_char(10.10, 'C999,999.99')  
from dual;
```

TO_CHAR(10.10,'C999,999.99')	
1	RUB10.10

TO_CHAR()

----- D - знак разделителя целой и дробной части

----- (NLS_NUMERIC_CHARACTER)

```
select to_char(10.10, 'C999D999')  
from dual;
```

TO_CHAR(10.10,'C999D999')	
1	RUB10,100

----- EEEE - знак научной записи

```
select to_char(10.10, '999EEEE')  
from dual;
```

TO_CHAR(10.10,'999EEEE')	
1	1E+01

----- G - знак разделителя

----- (NLS_NUMERIC_CHARACTER)

```
select to_char(100.100, '9G99')  
from dual;
```

TO_CHAR(100.100,'...')	
1	1 00

----- L - знак валюты

----- (NLS_CURRENCY)

```
select to_char(100.100, 'L999.99')  
from dual;
```

TO_CHAR(100.100,'L999.99')	
1	₽100.10

TO_CHAR()

```
----- MI - знак числа
select to_char(-100.100, '999.99MI')
from dual;
```

TO_CHAR(-100.100,'999.99MI')	
1	100.10-

```
----- PR - разное отображение
-- положительных (с пробелом)
-- отрицательных / (в угловых скобках)
select to_char(100.100, '999.99PR')
from dual;
```

TO_CHAR(100.100,'999....')	
1	100.10

```
----- TM - минимально возможное число символов
select to_char(100.100000, 'TM')
from dual;
```

TO_CHAR(100.100000...)	
1	100,1

```
----- U - знак валюты
----- (NLS_DUAL_CURRENCY)
select to_char(100.100, 'U999.99')
from dual;
```

TO_CHAR(100.100,'U99...')	
1	p.100.10

```
----- V - *10 - умножить
select to_char(100.100, '999V99')
from dual;
```

TO_CHAR(100.100,'99...')	
1	10010

```
----- XXXX - шестнадцатеричное значение (целое)
-- только положительные
select to_char(100.100, 'XXXX')
from dual;
```

TO_CHAR(100.100,'XXXX')	
1	64

Вопросы?

