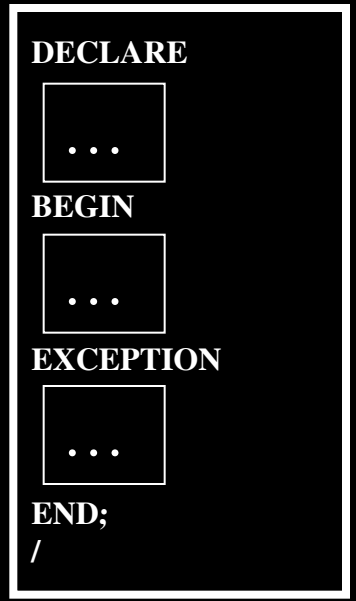
2

Написание исполняемых конструкций

Синтаксис блока PL/SQL

- Программные конструкции блока PL/SQL могут располагаться на нескольких строках.
- Лексические единицы делятся на:
  - Разделители
  - Идентификаторы
  - Литералы
  - Комментарии
- Слэш (/) определяет начало программного выполнения PL/SQL блока



Разделители

Простые символы	Составные символы
+	< >
-	! =
*	||
/	- -
=	/ *
@	* /
;	: =

Разделители. Набор символов PL\SQL

; Завершает объявления и команды

% Индикатор атрибута (атрибут курсора, подобный %ISOPEN, или атрибут неявных объявлений, например %ROWTYPE) также используется в качестве символа подстановки в условии LIKE

_ Обозначение подстановки одного символа в условии LIKE

@ Признак удаленного местоположения

<> или != Оператор сравнения «не равно»

|| Оператор конкатенации, объединение строк

<< и >> Ограничители метки

<= и >= Операторы сравнения «меньше или равно» и «больше или равно»

:= Оператор присваивания

=> Оператор ассоциации

.. Оператор диапазона

-- Признак однострочного комментария

/* и */ Начальный и конечный ограничители многострочного комментария

Комментарии

- Одиночный комментарий обозначается (--).
- Многострочный располагается между /* и */

```
BEGIN
/* Вычислить ежегодное жалованье, основанное
   на ежемесячном вводе жалованья от
   пользователя */
   v_sal := :g_monthly_sal * 12;
END; -- Конец этого блока
```

Идентификаторы

- PL/SQL не учитывает регистр символов
- длина - до 30 символов, может содержать A-Z, 0-9, _, $
- должны начинаться с буквы
- могут включать символы $, _ и #
- не должны содержать пробелов
- не должны (а в большинстве случаев и не сможете) использовать зарезервированные слова : такие, как BEGIN, IF и т.д., есть специальный справочник :V$RESERVED_WORDS ~ 2000 слов.

  PLS-00103: Encountered the symbol "END" when expecting one of the following.
- некоторые из правил именования объектов можно нарушить, заключая идентификатор в кавычки
- переменные могут иметь одинаковые имена в случае, если находятся в разных блоках
- имя переменной не должно совпадать с именем поля таблицы, используемого в данном блоке

Литералы

- Символьные литералы и даты должны заключаться в одиночные кавычки.

```
v_name := 'Henderson';
```

- Числовые литералы могут быть представлены простым или экспоненциальным значением.

Использование SQL функций в PL/SQL

Доступны в процедурных блоках функции:

- однострочные числовые операторы (ROUND, TRUNC, MOD, …)
- однострочные символьные операторы (LOWER, UPPER, INITCAP, CONCAT, SUBSTR, LENGTH, INSTR)
- преобразования типов данных (TO_DATE, TO_CHAR, TO_NUMBER)
-работы с датой (NEXT_DAY, LAST_DAY, ROUND, …)

Не доступны (вне SQL конструкций, в рамках SQL доступны):

-групповые функции (AVG, MIN, MAX, COUNT, SUM, STDDEV, and VARIANCE)
-DECODE

SQL функции в PL/SQL блоках (пример)

Доступны в процедурных блоках функции:

```
v_mailing_address := v_name||CHR(10)||v_state||CHR(10)||v_zip;

v_date := TO_DATE('12-JAN-2001', 'DD-MON-YYYY');

v_ename := LOWER(v_ename);
```

Групповые функции не доступны (вне SQL конструкций, в рамках SQL доступны):

Так нельзя - v_ename := SUM (employees.salary);

Только в рамках SQL конструкции

```
BEGIN
    SELECT SUM(salary) -- group function
        INTO v_sum_sal
        FROM employees WHERE department_id = v_deptno;
END;
```

Преобразование типов данных

- PL\SQL конвертирует типы динамически при необходимости, например при присвоении переменной типа CHAR значения NUMBER и наоборот.
- Смешивание типов данных может приводить к ошибке конвертации и потере производительности.
- Функции преобразования: TO_CHAR, TO_DATE, TO_NUMBER

```
DECLARE
   v_date DATE := TO_DATE('12-JAN-2001', 'DD-MON-YYYY');
BEGIN
. . .
```

Ошибки преобразования и исправления

Эта инструкция приведет к ошибке трансляции если переменная

v_date объявлена как тип данных DATE

```
v_date DATE := 'January 13, 2001';
```

Корректный синтаксис выглядит сл. образом:

```
v_date := TO_DATE ('January 13,2001','Month DD, YYYY');
```

Вложенность блоков и область видимости переменных

- PL/SQL блок может быть вложен везде, где исполняемый раздел это позволяет.
- Вложенный блок сам становится инструкцией.
- Раздел исключения может содержать вложенные блоки.
- Ссылки к идентификатору разрешены согласно области его видимости.

Вложенные анонимные блоки

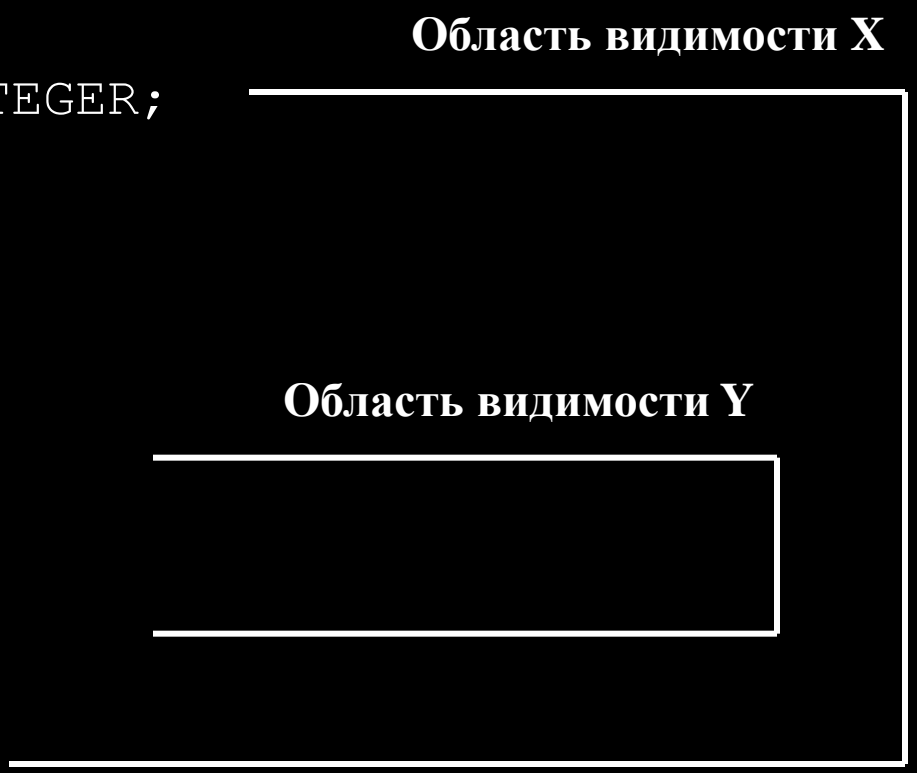
PL/SQL, как и языки Ada и Pascal, относится к категории *языков с блочной структурой*, то есть блоки PL/SQL могут вкладываться в другие блоки.

В PL/SQL переменные, исключения, модули являются локальными для блока, где они объявлены. За переделами блока эти структуры становятся недоступными. Часто вложенный блок создается для обработки исключений, после обработки программа продолжит свое выполнение.

```
DECLARE
   CURSOR emp_cur IS ...;
BEGIN
   DECLARE
      total_sales NUMBER;
   BEGIN
      DECLARE
         l_hiredate DATE;
      BEGIN
         ...
      END;
   END;
END;
```

Вложенность блоков (пример)

```
...
   x BINARY_INTEGER;
...
  BEGIN
   ...
   DECLARE
      y NUMBER;
   BEGIN
      y:= x;
   END;
   ...
   END;
```



Вложенность блоков

```
<<outer>>
DECLARE
   birthdate DATE;
BEGIN
   DECLARE
      birthdate DATE;
   BEGIN
   ...
   outer.birthdate :=
   TO_DATE('03-AUG-1976','DD-MON-YYYY');
   END;
   ....
END;
```

Операторы в PL/SQL

- Логические, Арифметические, Конкатенации
- Порядок выполнения операторов: возведение в степень, унарные операции, умножение и деление, сложение и вычитание, операторы сравнения, логические операторы.
- Оператор возведения в степень (**) - отсутствует в SQL – функция POWER

SQL

```
select (2**32) from dual
        ORA-00936: отсутствует выражение
select power (2, 32) from dual
        4294967296
```

PLSQL

```
 BEGIN
        DBMS_OUTPUT.PUT_LINE(2 ** 32);
        DBMS_OUTPUT.PUT_LINE(POWER(2,32));
    END;
    /
PL/SQL procedure successfully completed.
4294967296
4294967296
```

Операторы в PL/SQL(пример)

Инкремент счетчика для цикла

```
v_count := v_count + 1;
```

Установка булева значения (:= vs =):

```
v_equal := (v_n1=v_n2);
```

Проверка условия:

```
v_valid := (v_emp IS NOT NULL);
```

DBMS_OUTPUT.PUT_LINE

- Стандартный пакет СУБД ORACLE
- Вывод информации на экран из блока PL/SQL
- Необходимо задать в SQL Plus , SQLDeveloper :

```
SET SERVEROUTPUT ON
```

DBMS_OUTPUT.PUT_LINE (пример)

```
set serveroutput on

DECLARE
        v_sal EMPLOYEES.SALARY%TYPE;
BEGIN
        SELECT SALARY
                INTO v_sal
                FROM EMPLOYEES
                WHERE EMPLOYEE_ID = 189;
        dbms_output.put_line(v_sal);
        dbms_output.put_line('зарплата – ' || v_sal);

END;

set serveroutput off
```

DBMS_OUTPUT.PUT_LINE

```
set serveroutput on
DECLARE
        start_date TIMESTAMP;
        end_date TIMESTAMP;
        service_interval INTERVAL YEAR TO MONTH;
        years_of_service NUMBER;
        months_of_service NUMBER;
BEGIN
        start_date := TO_TIMESTAMP('29-01-1988','dd-mm-yyyy');
        end_date := TO_TIMESTAMP ('26-11-1995','dd-mm-yyyy');
        service_interval := (end_date - start_date) YEAR TO MONTH;
        DBMS_OUTPUT.PUT_LINE(service_interval);
        years_of_service := EXTRACT(YEAR FROM service_interval);
        months_of_service := EXTRACT(MONTH FROM service_interval);
                DBMS_OUTPUT.PUT_LINE(years_of_service || ' years and '|| months_of_service || '
months');
END;
```

Использование связанных переменных BIND(связывать) переменные

Связанная переменная обозначается символом (:) перед идентификатором (пример в SQL Developer и SQL Plus) Присваивание значения связанной переменной аналогично переменной PL/SQL.

```
-- SQL -- DEFINE  g_employee =189;
-- SQL -- VARIABLE g_salary NUMBER;

DECLARE
  v_sal EMPLOYEES.SALARY%TYPE;
BEGIN
  SELECT SALARY
          INTO v_sal
          FROM EMPLOYEES
          WHERE EMPLOYEE_ID = &g_employee;
  :g_salary := v_sal;
END;

print g_salary
```

Рекомендации по разработке кода

- Использовать комментарии в коде
- Использование различных регистров символов для различных частей кода (сл. слайд)
- Именование объектов согласно рекомендациям

  Правила создания PLSQL кода и обьектов баз данных

  \\Fileserver.office.bercut.ru\СКС\RBT\DB monitoring_Обучение\PLSQL курс
- Использование выравнивания текста (через слайд)

  mnSMSC\3.9.0\Sun_Solaris\mnSMSC\3.9.0.14\mnSMSC Common Databases\mnSMSCMessage\Scripts

Рекомендации по регистру кода

Категория	Регистр	Пример
SQL инструкции	UPPER	INSERT, UPDATE
Ключевые слова PL/SQL	UPPER	DECLARE, BEGIN, IF
Описание типа данных	UPPER	VARCHAR2, NUMBER, и пр.
Переменные, константы	lower	v_salary, c_setl, и пр.
БД, таблицы, столбцы	lower	employees, dbs и пр.

Выравнивание кода

Для удобства чтения кода рекомендуется выравнивание каждой строки:

```
IF x>y THEN v_max:=x;ELSE v_max:=y; END IF;
```

```
IF x>y THEN
v_max:=x;
ELSE
    v_max:=y;
END IF;
```

```
DECLARE
   v_deptno NUMBER(4);
                  v_location_id
               NUMBER(4);
               BEGIN
   SELECT deptno, loc
   INTO v_deptno, v_location_id
   FROM dept
   WHERE dname = 'SALES';
   ...
END;
```

Итоги

- Синтаксис PL/SQL блока
- Работа с идентификаторами
- Область видимости и действия идентификаторов и вложенных блоков
- Программирование PL/SQL
  - Функции
  - Преобразование типов
  - Операторы
  - Рекомендации к коду

Практика №2!

30 минут

Самостоятельное изучение

Распространенные механизмы ввода/вывода в PL/SQL

- DBMS_OUTPUT — вывод информации на экран;
- UTL_FILE — чтение и запись файлов операционной системы;
- UTL_MAIL и UTL_SMTP — отправка электронной почты из PL/SQL;
- UTL_HTTP — получение данных с веб-страниц.

* Глава 22 **Ввод/вывод в PL/SQL.**