

PL/SQL

Процедурное расширение языка
SQL

Темы

- **Простой блок PL/SQL и его компоненты**
- **Роль переменных в PL/SQL**
- **Разница между переменными PL/SQL и прочими переменными**
- **Объявление переменных PL/SQL**
- **Выполнение блока PL/SQL**

Структура блока PL/SQL

- **DECLARE** – необязательно
 - Переменные, курсоры, пользовательские исключения
- **BEGIN** – обязательно
 - Команды SQL
 - Команды PL/SQL
- **EXCEPTION** – необязательно
 - действия при возникновении ошибки
- **END;** – обязательно

```
DECLARE
  ...
BEGIN
  ...
EXCEPTION
  ...
END;
```

Структура блока PL/SQL

```
DECLARE
  v_variable VARCHAR2(5);
BEGIN
  SELECT      column_name
  INTO        v_variable
  FROM        table_name;
EXCEPTION
  WHEN exception_name THEN
  ...
END;
```

```
DECLARE
  ...
BEGIN
  ...
EXCEPTION
  ...
END;
```

Типы блоков

Анонимный блок

```
[DECLARE]

BEGIN
  --команды

[EXCEPTION]

END;
```

Процедура

```
PROCEDURE имя
IS

BEGIN
  -- команды

[EXCEPTION]

END;
```

Функция

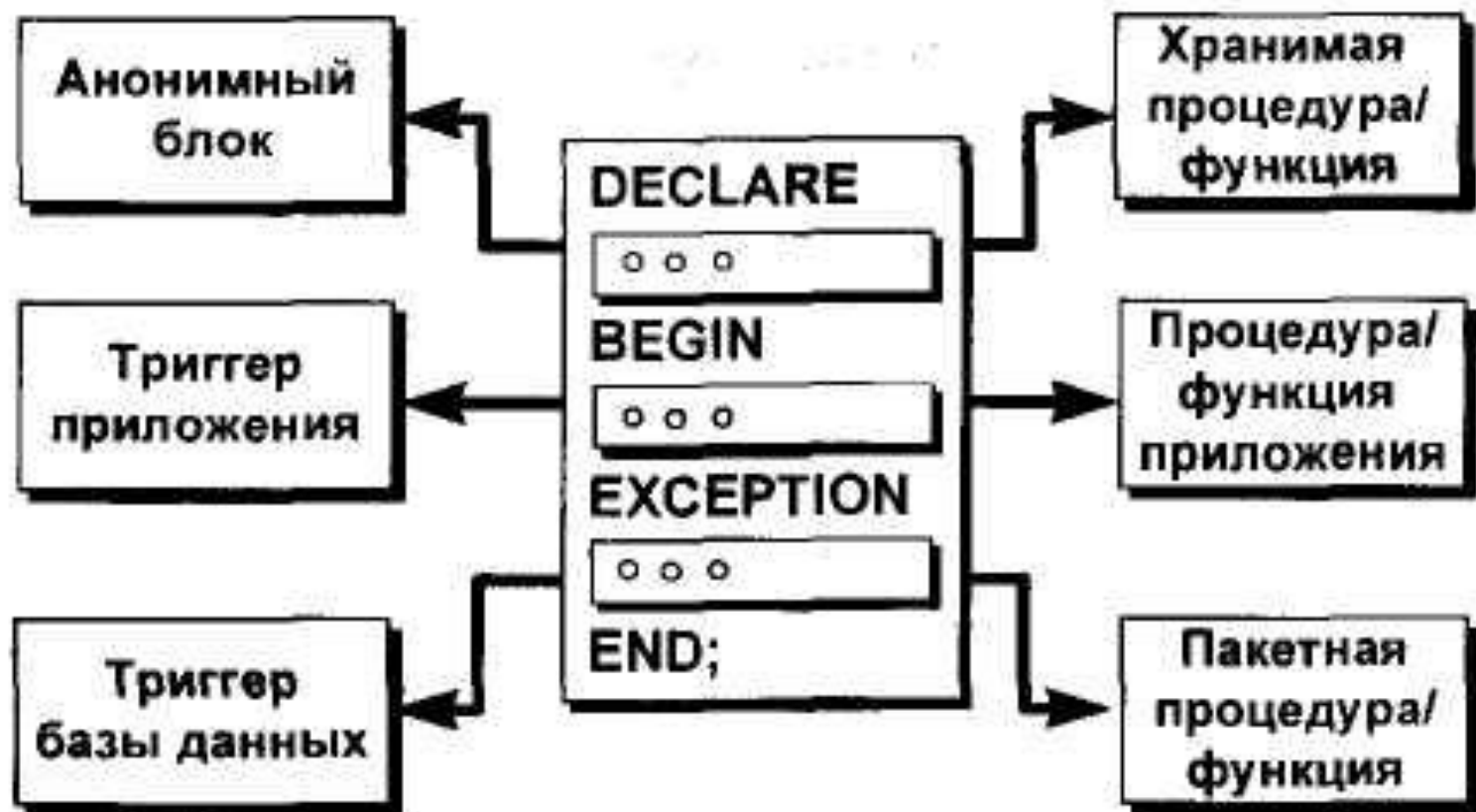
```
FUNCTION имя
RETURN тип данных
IS

BEGIN
  -- команды
  RETURN значение;

[EXCEPTION]

END;
```

Программные конструкции



Обработка переменных в PL/SQL

- **Объявление и инициализация переменных в декларативной секции.**
- **Присвоение новых значений переменным в исполняемой секции.**
- **Передача значений в блоки PL/SQL с помощью параметров.**
- **Просмотр результатов с помощью выходных переменных.**

Типы переменных

- • **Переменные PL/SQL**
- **Скалярные**
- **Составные**
- **Ссылочные**
- **LOB (большие объекты)**
- • **Прочие переменные (не PL/SQL)**
- - **Связанные переменные и хост-переменные**

Типы переменных

25-OCT-99

TRUE

"Four score and seven years ago
our fathers brought forth upon
this continent, a new nation,
conceived in LIBERTY, and dedicated
to the proposition that all men
are created equal."

256120.08

Atlanta

Объявление переменных PL/SQL и КОНСТАНТ

Синтаксис

```
identifier [CONSTANT] datatype [NOT NULL]  
    [:= | DEFAULT expr];
```


Примеры

```
Declare  
    v_hiredate      DATE;  
    v_deptno        NUMBER(2) NOT NULL := 10;  
    v_location      VARCHAR2(13) := 'Atlanta';  
    c_comm          CONSTANT NUMBER := 1400;
```

Правила присвоения имен

- **Две переменные могут иметь одинаковые имена, если находятся в разных блоках.**
- **Имена (идентификаторы) переменных не должны совпадать с именами столбцов таблицы, используемыми в этом блоке.**

```
DECLARE
  empno NUMBER(4);
BEGIN
  SELECT empno
  INTO empno
  FROM emp
  WHERE empname = 'SMITH';
END;
```



Присвоение значений переменным

Синтаксис

```
identifier := expr;
```

Примеры

Установка определенной даты найма для новых служащих.

```
v_hiredate := '31-DEC-98';
```

Установка имени служащего "Maduro."

```
v_ename := 'Maduro';
```

Второй способ

- Еще один способ присвоения значений переменным - это выборка значений базы данных в эти переменные.

```
SELECT sal * 0.10 INTO bonus
```

- FROM emp

- WHERE empno = 7369;

Пример

- DECLARE
- Product NUMBER;
- BEGIN
- SELECT quantity_on_hand
- INTO product
- FROM pl.product
- WHERE product_name = 'Small Widget';
- dbms_output.put_line ('Small Widget ' || product);
- END;

Оператор IF

- Оператор IF имеет следующий синтаксис:
- IF *условие_1* THEN
- *действие_1*;
- [ELSIF *условие_2* THEN
- *действие_2*,
- [ELSE
- *альтернативное_действие*]
- END IF;

Пример

- CREATE FUNCTION compute_discounts (order_amt NUMBER)
- RETURN NUMBER IS
- small_order_amt NUMBER := 400;
- large_order_amt NUMBER := 1000;
- small_disct NUMBER := 1;
- large_disct NUMBER := 5;
- BEGIN
- IF (order_amt < large_order_arat AND order_amt >= small_order_amt)
- THEN
- RETURN (order_amt * small_disct / 100) ;
- ELSIF (order_amt >= large_order_amt) THEN
- RETURN (order_amt * large_disct / 100);
- ELSE
- RETURN(0);
- END IF;
- END compute_discounts;

Циклы

- <<имя_цикла>>
- LOOP
- операторы;
- EXIT имя_цикла [WHEN условие_выхода];
- Операторы;
- END LOOP;
- Если условие WHEN отсутствует, операторы между LOOP и EXIT выполняются только один раз

Цикл WHILE

- WHILE *условие_выхода*
- LOOP
- *операторы;*
- END LOOP;

Цикл FOR

- FOR *счетчик* IN [REVERSE]
нижняя_граница .. Верхняя_граница ;
- LOOP
- *операторы*;
- END LOOP;

Пример