

Циклическое выполнение: команда LOOP

- В циклах исполняемая команд (или несколько команд) повторяются несколько раз.
- Есть несколько типов циклов:
 - Простой цикл `LOOP`
 - Цикл `FOR`
 - Цикл `WHILE`



Простой цикл

Синтаксис:

```
LOOP  
    statement1;  
    . . .  
    EXIT [WHEN condition];  
END LOOP;
```

Простой цикл: пример

```
DECLARE
  v_countryid      locations.country_id%TYPE := 'CA';
  v_loc_id         locations.location_id%TYPE;
  v_counter        NUMBER(2) := 1;
  v_new_city       locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;
  LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 3;
  END LOOP;
END;
/
```

Цикл WHILE

Синтаксис:

```
WHILE condition LOOP  
    statement1;  
    statement2;  
    . . .  
END LOOP;
```

Цикл WHILE повторяется, пока условие равно TRUE.

Цикл WHILE: пример

```
DECLARE
  v_countryid    locations.country_id%TYPE := 'CA';
  v_loc_id       locations.location_id%TYPE;
  v_new_city     locations.city%TYPE := 'Montreal';
  v_counter      NUMBER := 1;
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;

  WHILE v_counter <= 3 LOOP

    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);

    v_counter := v_counter + 1;

  END LOOP;
END;
/
```

Цикл FOR

- Используйте цикл `FOR` для выполнения заданного количества раз.
- Переменную для счетчика объявлять не нужно, она объявляется автоматически.

```
FOR counter IN [REVERSE]
    lower_bound..upper_bound LOOP
    statement1;
    statement2;
    . . .
END LOOP;
```

Цикл FOR: пример

```
DECLARE
  v_countryid    locations.country_id%TYPE := 'CA';
  v_loc_id       locations.location_id%TYPE;
  v_new_city     locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id
    FROM locations
   WHERE country_id = v_countryid;

  FOR i IN 1..3 LOOP

    INSERT INTO locations(location_id, city, country_id)
      VALUES((v_loc_id + i), v_new_city, v_countryid );

  END LOOP;
END;
/
```

Правила цикла FOR

- Счетчик доступен только внутри цикла. Обращаться к переменной счетчика за пределами цикла нельзя.
- Не присваивайте значения переменной счетчика в теле цикла.
- Ни одна из границ цикла не должна быть `NULL`.

Какой тип цикла выбрать

- Используйте простой цикл, если команды в цикле должны исполниться хотя бы один раз.
- Используйте цикл `WHILE`, если перед выполнением обязательно надо проверить условие, и цикл может не исполниться ни разу.
- Используйте цикл `FOR`, если перед началом цикла известно количество повторений.

EXIT и EXIT WHEN

Если вы не хотите, чтобы программа «зациклилась», в теле цикла следует разместить команду EXIT или EXIT WHEN:

```
LOOP
    balance_remaining := account_balance
(account_id);
    IF balance_remaining < 1000
    THEN
        EXIT;
    ELSE
        apply_balance (account_id,
balance_remaining);
    END IF;
END LOOP;
```

EXIT и EXIT WHEN

В сущности, EXIT WHEN сочетает в себе функции IF-THEN и EXIT. Приведенный пример можно переписать с использованием EXIT WHEN

LOOP

```
/* Вычисление баланса */
```

```
    balance_remaining := account_balance  
(account_id);
```

```
/* Условие встраивается в команду EXIT */
```

```
EXIT WHEN balance_remaining < 1000;
```

```
/* Если цикл все еще выполняется, с баланса  
списываются средства */
```

```
apply_balance (account_id, balance_remaining);
```

```
END LOOP;
```

CONTINUE

Он используется для выхода из текущей итерации цикла и немедленного перехода к следующей итерации.

```
BEGIN
  FOR l_index IN 1 .. 10 LOOP
    CONTINUE WHEN MOD(l_index, 2) = 0;

    DBMS_OUTPUT.PUT_LINE('Счетчик = ' || TO_CHAR(l_index));

  END LOOP;
END;
```