**Организация ветвлений и циклов в PL/SQL**

PL/SQL позволяет организовать ветвления и циклы известными способами.

IF команда имеет формат:

IF <условие> THEN <список\_команд> ELSE <список команд> END IF;

ELSE – не обязательная часть. Для организации вложенных ветвлений используется команда:

IF <условие\_1> THEN ...

ELSIF <условие\_2> THEN ...

... ...

ELSIF <условие\_n> THEN ...

ELSE ...

END IF;

Например:

Изменим предыдущий пример следующим образом: если в отобранной строке второй компонент равен 1, то строка добавляется, как в предыдущем примере, иначе – добавляется строка, каждый компонент которой увеличен на 10.

DECLARE

a NUMBER;

b NUMBER;

BEGIN

SELECT e,f INTO a,b FROM T1 WHERE e>1;

IF b=1 THEN

INSERT INTO T1 VALUES(b,a);

ELSE

INSERT INTO T1 VALUES(b+10,a+10);

END IF;

END;

.

run;

Циклы организуются при помощи команды:

LOOP

<тело\_цикла> /\* Список команд. \*/

END LOOP;

По крайней мере, одна из команд в <тело\_цикла> должна быть командой EXIT формата:

EXIT WHEN <условие>;

Цикл прерывается, если <условие> - истинно.

Например:

В таблицу T1, определенную в примере выше, добавляются строки со значениями от (1, 1) до (100, 100):

DECLARE

i NUMBER := 1;

BEGIN

LOOP

INSERT INTO T1 VALUES(i,i);

i := i+1;

EXIT WHEN i>100;

END LOOP;

END;

.

run;

Дополнительные команды организации циклов:

* Команда EXIT прерывает цикл без условия. Эту команду можно использовать в команде ветвления.
* Команда цикла WHILE:

WHILE <условие> LOOP

<тело\_цикла>

END LOOP;

* Команда цикла FOR:

FOR <переменная> IN <нач.значение>..<кон.значение> LOOP

<тело\_цикла>

END LOOP;

Где <переменная> может быть любой переменной; она является локальной переменной по отношению к команде цикла, поэтому ее не надо объявлять, <нач.значение> и <кон.значение> должны быть константами .