**Циклы в базах данных средствами PL/SQL**

|  |
| --- |
|  |

*В наши дни базы данных применяются везде: в бухгалтерии, в библиотеке,на складе. А наиболее большим поставщиком программного обеспечения для обработки информации является фирма Oracle. Стандартный язык структурированных запросов SQL не поддерживает циклы, однако весьма нередко, для образования отчетов, к примеру, без циклов не обойтись. Потому корпорация Oracle сделала расширение этого языка - PL/SQL.*

Циклы предназначены для многократного исполнения конкретных операций. ВPL/SQL есть 3 типа циклов:

1. Безусловные циклы (бесконечные) - выход из них не предусмотрен. Потому как такие циклы специфичны и применяются редко, то организуются они при помощи условных циклов. Условие возможно вообще не указывать или вместо него применять константу. Примером подобного цикла является машина LOOP ... END LOOP.
2. Условные циклы - команды выполняются прежде тех пор, пока истинно условие.Пример, WHILE условие LOOP ... END LOOP.
3. Циклы со счетчиком - цикл, в коем счетчик меняет свое значение с начального прежде конечного с определенным этапом, и для любого значения счетчика выполняется последовательность команд, находящихся в цикле. Представителем данного типа циклов является машина FOR счетчик IN стартовое значение..конечное значение LOOP ... END LOOP.

Самый простой цикл, написанный на PL/SQL, выглядит так:

LOOP

NULL;

END LOOP

/

Такой цикл совершенно бессмысленен и небезопасен для для экземпляра БД, так как не имеет выхода. Из таких циклов есть три способа выхода:

1. EXIT - это безусловный выход. Обычно он используется вместе с операторомIF, в котором проверяется условие выхода.
2. EXIT WHEN - в этом случае выход осуществляется пр при выполнении условия,описанного после WHEN.
3. GOTO - в этом случае выход из цикла осуществляется с переходом во внешний контекст, заданный меткой.

Давайте рассмотрим пример применения цикла вида LOOP EXIT WHEN. Вот код:

DECLARE

i NUMBER := 0;

BEGIN

LOOP -- start loop 1

i := i + 1;

IF (i >= 100) THEN

i := 0;

EXIT; -- exit when i >= 0

END IF;

END LOOP; -- end loop 1

LOOP -- start loop 2

i := i + 1;

EXIT WHEN (i >= 100); -- exit when i == 100

END LOOP; -- end loop 2

END;

/

Видимых действий этот код не производит, но и ошибок тоже! Первый цикл заканчивается когда i становится равной 100. При этом она обнуляется и осуществляется выход из цикла. Следующий цикл применяет вложенную конструкциюEXIT WHEN, использование которой является более синтаксически правильным.Впрочем, применение условных операторов позволяет проделать некоторые действия перед тем как выйти из цикла. В цикл вида LOOP EXIT WHEN END LOOP является самым часто используемым циклом при построении курсоров.

Вот еще одна разновидность цикла:

DECLARE

k NUMBER := 0;

BEGIN

WHILE (k < 10) LOOP

k := k + 1;

END LOOP;

END;

/

В отличие от ранее написанного кода, действия здесь выполняются только пока истинно в круглых скобках. Если условие принимает ложное значение, то выполнение цикла прекращается.

В конструкциях циклов в PL/SQL нет оператора CONTINUE, который иногда является довольно полезным. Это имеет место потому, что выражение CONTINUEявляется зарезервированым в языке PL/SQL для других целей. Впрочем,конструкцию CONTINUE можно искусственно эмулировать, с помощью циклаLOOP EXIT WHEN END LOOP а также полезного, хотя и непопулярного, оператораGOTO.

Для примера выведем нечетные числа вполоть до 20 с помощью цикла и GOTO:

SET SERVEROUTPUT ON

DECLARE

s NUMBER := 0;

BEGIN

DBMS\_OUTPUT.enable;

LOOP

IF(MOD(s, 2) = 1) THEN

GOTO LESS;

END IF;

DBMS\_OUTPUT.put\_line(TO\_CHAR(s)||' is even!');

<<LESS>>

EXIT WHEN (s = 20);

s := s + 1;

END LOOP;

END;

/

Надеюсь, теперь для вас не составит труда применять циклы PL/SQL для построении серверных приложений баз данных! :)

Смотрите также близкую по теме статью:["Оператор IF в стандартеSQL:2003"](http://xbb.uz/db/Operator_IF_v_standarte_SQL_2003), с описанием синтаксиса и работы условного оператора согласно стандарту и SQL и примером его использования в PL/pgSQL.