

## ЛАБОРАТОРНАЯ РАБОТА №20

*Тема:* Анонимные блоки PL/SQL

*Цель:* Пробрести навыки создания и использования анонимных блоков на языке PL/SQL.

### Теоретические сведения

Язык PL/SQL – это разработанное в Oracle процедурное расширение языка SQL. Этот язык является переносимым, высокоэффективным, ориентированным на обработку транзакций языком, который тесно интегрированным с SQL. Базовой конструкцией языка PL/SQL является блок, в которой сгруппированы соответствующие объявления и исполняемые операторы.

Блок PL/SQL определяется ключевыми словами DECLARE, BEGIN, EXCEPTION и END. Эти ключевые слова разделяют блок на декларативную и исполняемую части, а также на секцию обработки ошибок. Только исполняемая часть является обязательной. Остальные используются по мере необходимости.

**begin**

```
dbms_output.put_line('hello world');
```

**end;**

Как указано выше, анонимный блок может иметь собственные секции объявлений переменных и обработки ошибок. По этой причине разработчик имеет возможность использовать анонимный блок с целью обеспечения собственного (ограниченного только этим блоком) контекста видимости используемых в блоке идентификаторов и обработчиков ошибок. Такую внутреннюю область видимости часто называют локальной областью видимости.

```
declare l_today date := sysdate;
```

```
begin
```

```
    dbms_output.put_line( 'today is ' ||  
to_char(l_today, 'Day') );
```

```
exception
```

```
when
```

```
    others then dbms_output.put_line(sqlerrm);
```

```
end;
```

Условная логика

В языке PL/SQL можно использовать условные операторы, такие же как и в большинстве обычных алгоритмических языков. Допускается использование оператора if ... then ... else. Функция TO\_CHAR предназначена для форматирования дат и числовых типов данных (таких как number). Символ D в формате маски соответствует дню недели (число в диапазоне от 1 до 7).

```
declare l_today date := sysdate;
begin
    if to_char(l_today,'D') < 4 then
        dbms_output.put_line( 'Have a wonderful week' );
    else
        dbms_output.put_line( 'Enjoy the rest of the
week' );
    end if;
    dbms_output.put_line('today is '
        || to_char(l_today,'Day')
        || ' day '
        || to_char(l_today,'D')
        || ' of the week. ');
end; /
```

## Статический SQL

В языке PL/SQL сравнительно легко использовать обычные SQL предложения. Ниже показан пример использования оператора **SELECT INTO**, который работает в данном случае со словарем таблиц схемы текущего пользователя Oracle.

```
DECLARE howmany INTEGER;
        num_tables INTEGER;
BEGIN
    -- Begin processing
    SELECT COUNT(*) INTO howmany
        FROM USER_OBJECTS
        WHERE OBJECT_TYPE = 'TABLE';
    -- Check number of tables
    num_tables := howmany;
    -- Compute another value
    dbms_output.put_line
    (to_char(num_tables,'999G999G990')
    || ' tables');
END;
```

Ниже показан пример использования цикла **loop**

```
DECLARE
    l_table_count integer := 0;
```

```

BEGIN
  for c1 in (
    SELECT table_name
      FROM USER_TABLES order by 1)
  loop
    l_table_count := l_table_count + 1;
    dbms_output.put_line(c1.table_name);
  end loop;
  if l_table_count = 0 then
    dbms_output.put_line('You have no tables in your
schema');
  end if;
END;

```

### Динамический SQL

Допускается формированием SQL предложения «на лету». Расширим предыдущий пример, и включим в него динамическое построение SQL предложения. Для того, чтобы получать возвращаемое значение из динамического SQL создадим пакет с названием "PKG".

```

CREATE OR REPLACE PACKAGE pkg AUTHID
DEFINER AS
table_row_count integer;
END;
DECLARE
  l_table_count integer := 0;
  l_sql varchar2(32767);
BEGIN
  for c1 in (
    SELECT table_name FROM USER_TABLES order by 1
  )
  loop
    l_table_count := l_table_count + 1;
    l_sql := 'begin
select count(*) into pkg.table_row_count
from "' || c1.table_name || '"
end;';
    execute immediate l_sql;
    dbms_output.put_line(c1.table_name
|| ' - '
|| to_char(pkg.table_row_count, '999G999G990')
|| ' rows');
  end loop;
  if l_table_count = 0 then

```

```
        dbms_output.put_line('You have no tables in your  
schema');  
    end if;  
END; /
```

### Порядок выполнения работы

1. Выполните фрагменты программ, приведенные в данных методических указаниях.
2. Составьте анонимный блок, который рассчитывает и выводит сумму заработной платы всех сотрудников из таблицы employees БД HR.  
(Указание: при расчете заработной платы сотрудника учитывать размер его премии, если процент премии не указан, принять его равным среднему проценту премии по подразделению)
3. Составьте анонимный блок, который выводит последовательность чисел от 1 до количества дне в текущем месяце. Указание – использовать цикл for ... loop ... end loop;

### Содержание отчета.

1. Название работы
2. Цель работы
3. Листинги запросов
4. Скриншоты результатов выполнения запросов.
5. Выводы

### Контрольные вопросы

1. Укажите назначение и особенности языка PL/SQL.
2. Для чего используются анонимные блоки в PL/SQL?
3. Из каких секций состоит анонимный PL/SQL блок? Какие секция являются обязательными для использования, и какие могут отсутствовать в объявлении PL/SQL блока?
4. Объясните различие между статическим и динамическим SQL в анонимных блоках. Приведите пример использования условного оператора и оператора цикла в анонимных блоках.
5. Приведите пример использования условного оператора анонимных блоках.

6. Приведите пример использования и оператора цикла в анонимных блоках.
7. Для чего используется конструкция INTO в операторе SELECT