

#####

ЛАБОРАТОРНАЯ РАБОТА N3

ТЕМА: Язык PL/SQL.

Процедуры и функции. Пакеты

Используемые объекты:

1. Учебная база данных - таблицы SAL, CUST, ORD
2. Описание учебной базы данных - таблица OPDB2

Для манипуляций с данными можно создавать хранимые процедуры. Процедуры создаются командой CREATE OR REPLACE PROCEDURE и сохраняются в базе данных. Процедура имеет имя и представляет собой программный блок на языке PL/SQL. Данные передаются в процедуру и из нее при помощи входных и выходных параметров. Ниже показан пример создания процедуры, которая удаляет строки. Для определения строк, которые должны быть удалены, в процедуру передается название города как входной параметр.

```
CREATE OR REPLACE PROCEDURE del_dep
(in_city IN VARCHAR2)
AS
BEGIN
DELETE FROM dep WHERE loc = in_city;
END;
/
```

При создании процедуры могут возникать ошибки компиляции. Для анализа ошибок можно воспользоваться системной таблицей USER_ERRORS.

Вызов процедуры в блоке PL/SQL - например

```
BEGIN
del_dep('Boston');
END;
```

ЗАДАНИЕ 1.

Создайте процедуру, которая изменяет указанный номер отдела в таблице DEP (создайте сами) на другой. Текущий номер отдела и новый номер должны задаваться как входные параметры процедуры.

Подобно тому, как создаются хранимые процедуры, можно создавать хранимые функции. Так как функция всегда должна возвращать какое-то значение, в ее описании необходимо указать тип возвращаемых данных и включить в тело функции команду возврата RETURN:

```
CREATE OR REPLACE FUNCTION show_bool
(p_Bool IN BOOLEAN DEFAULT NULL) RETURN VARCHAR2 IS
str VARCHAR2(10);
BEGIN
```

```

    IF (p_Bool is null) THEN
        str := 'NULL';
    ELSIF (p_Bool) THEN
        str := 'TRUE';
    ELSE
        str := 'FALSE';
    END IF;
    RETURN(str);
END;
/

```

Вызвать функцию можно в блоке PL/SQL

```

BEGIN
    DBMS_OUTPUT.put_line(show_boll());
    DBMS_OUTPUT.put_line(show_boll(TRUE));
END;
/

```

или в команде SQL

```
SELECT show_boll(FALSE) FROM dual;
```

ЗАДАНИЕ 2.

Разработайте функцию, которая принимает в качестве параметра значение поля АМТ заказа. Функция должна возвращать строку в формате NNN рублей NN копеек. Слова «рубли» и «копейки» должны склоняться по правилам русского языка. Используйте функцию в команде SELECT следующим образом:

```
SELECT onum, имя_функции(AMT) FROM ord;.
```

Пакеты - это набор программных объектов, использующих общие данные и другие элементы PL/SQL. Пакет состоит из двух частей - спецификации (или заголовка) и тела. Каждая часть пакета создается отдельной командой CREATE. В спецификации нет исполняемого кода, там находятся декларации языковых конструкций, которые доступны для всех. Для создания спецификации необходимо выполнить команду

```
CREATE OR REPLACE PACKAGE имя_пакета
```

В теле пакета содержится программный код всех упоминаемых в декларации программ. Кроме того, в тело пакета можно поместить любые программные конструкции, не декларированные в заголовке - они будут доступны из программ пакета, но не будут доступны из внешних процедур и пакетов. Наконец, здесь может также содержаться программа инициализации, которая выполняется один раз во время первого обращения к пакету. Тело пакета создается командой

```
CREATE OR REPLACE PACKAGE BODY имя_пакета
```

В следующем примере показано создание заголовка и тела пакета Pack_1, содержащего две процедуры.

```
CREATE OR REPLACE PACKAGE Pack_1 IS
  PROCEDURE Sal_cnt;
  PROCEDURE Cust_cnt;
END Pack_1;
/

CREATE OR REPLACE PACKAGE BODY Pack_1 IS
  cnt NUMBER(2);
  PROCEDURE Sal_cnt IS
  BEGIN
    select count(*) into cnt from sal;
    DBMS_OUTPUT.put_line('Sal count = '||cnt);
  END Sal_cnt;
  PROCEDURE Cust_cnt IS
  BEGIN
    select count(*) into cnt from cust;
    DBMS_OUTPUT.put_line('Cust count = '||cnt);
  END Cust_cnt;
END Pack_1;
/
```

Обратите внимание, что в теле пакета используется глобальная переменная cnt. Чтобы вызвать процедуру из созданного пакета, нужно, используя точечную нотацию, задать имя пакета и имя процедуры. Например:

```
BEGIN
  Pack_1.Sal_cnt;
END;
/
```

ЗАДАНИЕ 3.

Создайте свой пакет по аналогии с Pack_1. В теле пакета в каждой процедуре определите курсор с параметром для указания города. Аналогичный параметр задайте для обеих процедур. В процедурах вместо команды SELECT используйте курсор для получения количества продавцов (покупателей) в городе, указанном в параметре. Название города укажите при вызове процедуры.

При разработке приложений обычно создается несколько пакетов, один из которых является библиотечным. В таком пакете находятся процедуры и прочие языковые конструкции, используемые в других пакетах. Для обращения к этим конструкциям используется все та же точечная нотация, т.е. имя пакета и, через точку, имя конструкции (например, процедуры). Параметры для пакетов, функций и курсоров задаются по тем же правилам, что и для обычных хранимых процедур, функций и т.д.

ЗАДАНИЕ 4.

Создайте два пакета, один из которых является библиотечным и содержит две функции. Во втором пакете выполняются вызовы этих функций. Библиотечные функции должны иметь параметры для задания фамилий членов бригады, причем первая функция должна возвращать самую длинную из фамилий, а вторая - самую короткую. Для определения длины строки используйте стандартную функцию LENGTH().

КОНТРОЛЬНЫЕ ВОПРОСЫ

=====

Контрольные вопросы выдает преподаватель после выполнения всех заданий.