



Хранимые процедуры и функции

Хранимые процедуры представляют собой группы связанных между собой операторов SQL, применение которых делает работу программиста более легкой и гибкой, поскольку выполнить *хранимую процедуру* часто оказывается гораздо проще, чем последовательность отдельных операторов SQL. Хранимые процедуры представляют собой набор команд, состоящий из одного или нескольких операторов SQL или функций и сохраняемый в базе данных в откомпилированном виде. Хранение процедур в том же месте, где они исполняются, обеспечивает уменьшение объема передаваемых по сети данных и повышает общую производительность системы.

Пример создания хранимой процедуры:

```
CREATE PROCEDURE HelloFit
IS
BEGIN
    DBMS_OUTPUT.enable;
    DBMS_OUTPUT.put_line('HELLO, FIT');
END HelloFit;
```

Пример запуска хранимой процедуры:

```
BEGIN
    HelloFit;
END;
```

Задание. Создайте вышеприведенную процедуру и запустите её.

Пример создания хранимой процедуры с аргументами:

```
CREATE PROCEDURE Proc2 (NUM IN INTEGER)
IS
stoim_put INTEGER;
BEGIN
  SELECT stoim INTO stoim_put
  FROM putevka
  WHERE countmest > NUM;
  DBMS_OUTPUT.enable;
  DBMS_OUTPUT.put_line(stoim_put);
END Proc2;
```

Задание. Создайте вышеприведенную процедуру и запустите её (примечание: процедура должна возвращать единственное значение).

Создание функций аналогично созданию процедуры:

```
CREATE FUNCTION BOOL_TO_CHAR(INBL IN BOOLEAN)
RETURN VARCHAR2
IS
OUT_ST VARCHAR2(5);
BEGIN
    IF INBL THEN
        OUT_ST := 'TRUE';
    ELSE
        IF NOT INBL THEN
            OUT_ST := 'FALSE';
        ELSE
            OUT_ST := 'NULL';
        END IF;
    END IF;
    RETURN(OUT_ST);
END BOOL_TO_CHAR;
```

Для того, чтобы использовать функцию запишем следующий запрос:

```
BEGIN
DBMS_OUTPUT.enable;
    DBMS_OUTPUT.put_line(BOOL_TO_CHAR(true));
    DBMS_OUTPUT.put_line(BOOL_TO_CHAR(false));
    DBMS_OUTPUT.put_line(BOOL_TO_CHAR(null));
END BOOL_TO_CHAR;
```

Пример: функция для расчета факториала числа:

```
CREATE FUNCTION FACTORIAL (NUM IN NUMBER)
RETURN NUMBER
IS
BEGIN
  IF (NUM <=1) THEN
    RETURN (NUM);
  ELSE
    RETURN (NUM * FACTORIAL(NUM-1));
  END IF;
END FACTORIAL;
```

Использование аналогично предыдущему случаю.

Задание. Реализовать вышеприведенные функции и запустить их.

Чтобы просмотреть исходный код созданной функции или процедуры необходимо написать следующий запрос:

```
SELECT OBJECT_NAME, OBJECT_TYPE, STATUS  
FROM USER_OBJECTS  
where OBJECT_NAME='FACTORIAL';
```

Задание. Просмотреть исходный код всех созданных процедур и функций.