



**Промышленные СУБД**  
**Лабораторная работа №6**

## **Лабораторная работа №6**

Функции.

Полезные ссылки для дополнительного изучения (CREATE FUNCTION, Структуры):

<https://www.postgresql.org/docs/current/sql-createfunction.html>

<https://www.postgresql.org/docs/current/plpgsql-control-structures.html>

Цель: изучить основные операторы и алгоритмические конструкции функций в Postgres.

Теоретический материал: перед выполнением лабораторной работы рекомендуется изучить лекцию №3 «Общие сведения о Transact-SQL», поскольку в ней представлено общее описание и назначение языка SQL, а также рассмотрены все алгоритмические конструкции, доступные в SQL.

Требования к отчету: по результатам работы нужно представить набор SQL-скриптов, решающих задачи из раздела «Самостоятельная работа».

## Задание 1. Объявление переменных

Синтаксис в обозначениях Postgres:

```
name [ CONSTANT ] type[ COLLATE collation_name ] [ NOT NULL ] [
{ DEFAULT | := | = } expression ];
```

**DEFAULT** Предложение, если оно задано, указывает начальное значение, присваиваемое переменной при вводе блока. Если **DEFAULT** предложение не указано, то переменная инициализируется значением SQL null. **CONSTANT** Опция предотвращает присвоение переменной после инициализации, так что ее значение будет оставаться постоянным в течение всего срока действия блока. **COLLATE** Параметр определяет параметры сортировки, которые будут использоваться для переменной

Пример:

```
quantity integer DEFAULT 32;
url varchar := 'http://mysite.com';
transaction_time CONSTANT timestamp with time zone := now();
```

Для инициализации требуется перед объявлением сделать объявление функции!!!

## Задание 2. Присвоение значений переменным и вывод значений на экран

Синтаксис:

```
a(Имя переменной) integer(Тип переменной) := 1(Значение);
```

Пример:

DECLARE

```
x integer := 1;
```

```
y integer := x + 1;
```

Вывод на экран возможен только в составе функции

Функция:

```
CREATE FUNCTION sales_tax(subtotal integer) RETURNS integer AS $$
```

DECLARE

```
x integer := 1;
```

```
y integer := x + 1;
```

BEGIN

```
RETURN x;
```

END;

```
$$ LANGUAGE plpgsql;
```

Синтаксис такой:

```
CREATE FUNCTION Название_функции(Имя_входной_переменной  
Тип_данных_вп) RETURNS тип_данных_функции AS $$
```

```
DECLARE(Здесь объявляем переменные)
```

```
x integer := 1;
```

```
y integer := x + 1;
```

```
BEGIN(Здесь проводим все математические расчеты)
```

```
RETURN x;
```

END;

```
$$ LANGUAGE plpgsql;
```

Для вывода результата используем:

```
SELECT * FROM Имя_функции(входные_данные)
```

Пример:

```
SELECT * FROM sales_tax(5)
```

### Задание 3. Работа с датой и временем

Для работы с датой используется стандартный тип данных `date`.

---

`date + integer → date`

Добавьте количество дней к дате

`date '2001-09-28' + 7 → 2001-10-05`

---

`date + interval → timestamp`

Добавьте интервал к дате

`date '2001-09-28' + interval '1 hour' → 2001-09-28 01:00:00`

---

`date + time → timestamp`

Добавьте время суток к дате

`date '2001-09-28' + time '03:00' → 2001-09-28 03:00:00`

---

Пример:

```
CREATE FUNCTION date_base1(subtotal date) RETURNS date AS $$  
DECLARE  
  x date := '2022-07-22';  
BEGIN  
  RETURN x + 1;  
END;  
$$ LANGUAGE plpgsql;
```

Пример:

```
CREATE FUNCTION date_base1(subtotal date) RETURNS date AS $$  
DECLARE  
  x date := '2022-07-24';  
BEGIN  
  RETURN subtotal + 1;  
END;  
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM date_base1('2022.07.22')
```

**Задание 4. Преобразование типов переменных**

Преобразование данных происходит в самом задании функции. Даже если у вас результат в `int`, вы можете его без всякого приведения вывести в `text`.

Пример:

```
CREATE FUNCTION privedenie() RETURNS text(Здесь идет приведение)
AS $$
    DECLARE
        x int := 5;
    BEGIN
        return x+1;
    END;
$$ LANGUAGE plpgsql;
```

Здесь мы получим результат в `text` значении.



**Задание 5. Условная конструкция IF**

Синтаксис:

IF условие THEN

Набор операторов1

ELSE

Набор операторов2

END IF

Пример:

CREATE FUNCTION if\_construction() RETURNS integer AS \$\$

DECLARE

x int := 5;

BEGIN

IF(x = 5) THEN --Задаем к параметру условия x = 5

x = x+1; --Если да, то прибавляем

END IF; --Закрываем условие

return x;

END;

\$\$ LANGUAGE plpgsql;

SELECT \* FROM if\_construction()

**Задание 6. Создание Цикла WHILE**

Синтаксис:

```
WHILE Условие LOOP
    Набор операторов1
END LOOP;
```

Пример:

```
CREATE FUNCTION if_construction2() RETURNS integer AS $$
DECLARE
    x int := 5;
    y int := 0;
BEGIN
    while x < 12 LOOP --Задаем условия цикла
        x = x + 1; --Набор действий
        y = y + 1; --Набор действий
    END LOOP;
    return y;
END;
$$ LANGUAGE plpgsql;


SELECT * FROM if_construction2()
```

**Задание 7. Посчитать количество цифр в числе.**

Указания к выполнению:

Один из возможных вариантов решения данной задачи (см. рис. 4):

```
CREATE FUNCTION chislo_podcheta2() RETURNS integer AS $$  
DECLARE  
  x int := 2145347691;  
  y int := 0;  
BEGIN  
  while x > 10 LOOP  
    x = x / 10;  
    y = y + 1;  
  END LOOP;  
  return y+1;  
END;  
$$ LANGUAGE plpgsql;  
  
SELECT * FROM chislo_podcheta2()
```



chislo_podcheta2 integer	
1	10

Рис. 4. Результат работы кода

**Задание 8. Создайте функцию, принимающую число в качестве делителя и возвращающую количество цифр в нем.**

Один из вариантов решения:

```
CREATE FUNCTION chislo_podcheta1(delitel integer, out z integer) AS $$  
DECLARE  
    y int := 0;  
BEGIN  
    while delitel > 10 LOOP  
        delitel := delitel / 10;  
        y := y + 1;  
    END LOOP;  
    z := y + 1;  
END;  
$$ LANGUAGE plpgsql;  
  
SELECT * FROM chislo_podcheta1(12342342)
```

## Самостоятельная работа

1) Написать программу пересчета веса из фунтов в килограммы (1 фунт равняется 453,6 г). Результат должен быть выведен следующим образом, например:

3.3 фунт(а/ов) – это 1 кг 496 г.

2) Вывести информацию о текущих именах сервера, учетной записи и пользователя базы данных в следующем виде:

Вы вошли на сервер User400-01\SQL2008 как User400-01\User with dbo permissions.

3) Определить количество часов и минут, прошедших со времени запуска служб MS SQL Server.

4) Определить, является ли текущий год високосным.

5) Найти сумму чисел в заданной строке символов.

6) Определить величину оплаты за отправку телеграммы. Признаком завершения телеграммы является точка. Стоимость одного слова 33 коп, результат вывести с указанием количества рублей и копеек. Строка может содержать произвольное число пробелов.

7) Используя шифр Цезаря, зашифруйте заданную строку текста. Идея данного метода шифрования – алфавит размещается как бы по часовой стрелке. Для шифровки буквы текста заменяются буквами, отстоящими на заданное число букв (сдвиг) по часовой стрелке.

8) Известна фамилия, имя и отчество пользователя. Найти число его личности. Правило получения числа личности: каждой букве сопоставлено число – порядковый номер буквы в алфавите. Эти числа складываются, если полученная сумма не является однозначным числом, то цифры числа снова складываются и так до тех пор, пока не будет получено однозначное число.