



Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

# ПОСТРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

## *Отчёт по лабораторной работе № 2*

**«Постреляционное расширение языка SQL на примере PostgreSQL»**

Выполнил:  
студент группы ИУ5 – 23М

Крутов Т.Ю.

Преподаватель:

Виноградова М. В.

2020г.

### **Цель работы**

- Изучить постреляционные возможности языка SQL.
- Освоить языки и технологии SQL\PSM.
- Получить навыки программирования на стороне сервера.

### **Средства выполнения**

- СУБД PostgreSQL.
- PgAdmin.

## 1.2. Программирование функций с применением SQL\PSM

### 1. Скалярная функция

```
CREATE FUNCTION ages( x integer) RETURNS integer AS $$  
  
    SELECT (current_date - (SELECT birthday FROM usser WHERE  
userid = x ));  
$$ LANGUAGE SQL;  
  
SELECT ages (3) as answer;
```

### 2. Табличная функция

```
CREATE FUNCTION ord (x integer)  
RETURNS TABLE (login varchar , oid integer ,  
                otype ordertype )  
  
AS $$  
  
    SELECT login AS login, orderid AS orid,  ord_type AS  
typ  
    FROM orrder, account  
    WHERE orrder.accountid = x and account.accountid = x;  
$$ LANGUAGE SQL;  
  
SELECT ord(3) as tab;
```

### 3. Хранимая процедура

```
CREATE OR REPLACE FUNCTION add_order (account_id integer,
                                     ammount_items
                                     integer ,
                                     ord_tyoe
                                     ordertype)
RETURNS integer AS
$$
BEGIN
    insert into orrder (accountid, ammount_items, ord_type)
    values ($1 , $2 , $3);
    return 1;
EXCEPTION
    WHEN invalid_parameter_value THEN
        return 0;
    WHEN invalid_text_representation THEN
        return 0;
END
$$
LANGUAGE 'plpgsql'

select add_order ('5', '3', 'Текст' ) as qwerty;
```

### Задание 2. Расширенная часть (хорошо)

#### 2.1. Извлечение части записей и результатов запросов на изменение данных

```
SELECT * FROM orrder
WHERE ord_type = 'Текст'
LIMIT 2
-- limit

UPDATE orrder
```

```
SET sizes = '{(1280,720) (0,0)}'  
WHERE orderid = 3 OR orderid = 5  
RETURNING orderid
```

## 2.2. Выполнение рекурсивных запросов

```
WITH RECURSIVE r AS (  
    SELECT adept, accountid, login  
    FROM account  
    WHERE accountid = 3  
  
    UNION  
  
    SELECT account.adept, account.accountid, account.login  
    FROM account  
    JOIN r  
        ON account.accountid = r.adept  
)  
SELECT * FROM r
```

## 2.3. Создание динамических запросов

```
CREATE OR REPLACE FUNCTION copy_user (namee varchar, X  
integer, Y integer)  
RETURNS integer AS  
$$  
BEGIN  
    IF EXISTS (  
        SELECT 1  
        FROM INFORMATION_SCHEMA.TABLES  
        WHERE table_name = namee)  
    THEN  
        DELETE FROM namee;  
        RETURN 1;  
  
    ELSE
```

```

EXEC SQL BEGIN DECLARE SECTION;

const char *stmt = " CREATE TABLE namee (userid
integer, firstname varchar, lastname varcharr, email
varchar,

telephone varchar, addr address, birthday date, dat
date[], status varchar); "

INSERT INTO namee (userid integer, firstname
varchar, lastname varcharr, email varchar,

telephone varchar, addr address, birthday date, dat
date[], status varchar)

SELECT userid, firstname,lastname,email,
telephone, addr, birthday, dat,status FROM usser WHERE
userid BETWEEN X AND Y;

EXEC SQL END DECLARE SECTION;

EXEC SQL EXECUTE IMMEDIATE :stmt;
RETURN 0;

END IF
END $$

LANGUAGE 'plpgsql';

```

### **Список литературы**

1. Документация PostgreSQL – [postgrespro.ru](http://postgrespro.ru) – Документация – PostgreSQL 9.4 – Серверное программирование.
2. Методические рекомендации.