Нуштаев Никита Петрович ПРИ-О-21/1 1 вариант

Функция с пустыми входными параметрами, результат которой скалярное выражение:

CREATE OR REPLACE FUNCTION get\_total\_orders() RETURNS INTEGER AS $$

DECLARE

total\_orders INTEGER;

BEGIN

SELECT COUNT(\*) INTO total\_orders FROM orders;

RETURN total\_orders;

END;

$$ LANGUAGE plpgsql;

Функция со скалярным аргументом, результат которой соответствует типу существующей таблицы:

CREATE OR REPLACE FUNCTION get\_product\_price(product\_id INTEGER) RETURNS DECIMAL AS $$

DECLARE

product\_price DECIMAL;

BEGIN

SELECT price INTO product\_price FROM products WHERE productid = product\_id;

RETURN product\_price;

END;

$$ LANGUAGE plpgsql;

Функция с выходными аргументами, определенными с помощью OUT:

CREATE OR REPLACE FUNCTION calculate\_order\_total(order\_id INTEGER, OUT total\_cost DECIMAL, OUT product\_count INTEGER) AS $$

BEGIN

SELECT SUM(p.price \* oi.quantity), COUNT(oi.productid)

INTO total\_cost, product\_count

FROM orderitems oi

JOIN products p ON oi.productid = p.productid

WHERE oi.orderid = order\_id;

END;

$$ LANGUAGE plpgsql;

Функция, результат которой определен с помощью RETURNS TABLE:

CREATE OR REPLACE FUNCTION get\_orders\_by\_customer(customer\_id INTEGER) RETURNS TABLE(order\_id INTEGER, order\_date DATE) AS $$

BEGIN

RETURN QUERY SELECT orderid, orderdate FROM orders WHERE customerid = customer\_id;

END;

$$ LANGUAGE plpgsql;

Создание процедуры:

CREATE OR REPLACE PROCEDURE update\_product\_price(product\_id INTEGER, new\_price DECIMAL) AS $$

BEGIN

UPDATE products SET price = new\_price WHERE productid = product\_id;

END;

$$ LANGUAGE plpgsql;

Переименование одной из функций:

ALTER FUNCTION get\_total\_orders() RENAME TO calculate\_total\_orders;

Удаление одной из функций:

DROP FUNCTION IF EXISTS get\_orders\_by\_customer;

Принтскрины:











