МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение автоматизированных систем»

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине**

**«Базы данных»**

**РФ КГУ 09.03.03КП25. 260091**

**Проект БД для магазина по продаже вычислительной техники**

**ТЕКСТ ПРОГРАММЫ**

**Листов 8**

Курган 2025

**АННОТАЦИЯ**

Данный документ содержит исходный код реализованных на языке PL/pgSQL, функций:

«products.products\_get\_by\_category»,

«storages.inventory\_get\_product\_count»,

«clients.basket\_info\_set».

Оглавление

[ИСХОДНЫЙ КОД ФУНКЦИИ «products.products\_get\_by\_category » 4](#_Toc198110576)

[ИСХОДНЫЙ КОД ФУНКЦИИ «storages.inventory\_get\_product\_count» 6](#_Toc198110577)

[ИСХОДНЫЙ КОД ФУНКЦИИ «clients.basket\_info\_set» 7](#_Toc198110578)

# ИСХОДНЫЙ КОД ФУНКЦИИ «products.products\_get\_by\_category »

CREATE FUNCTION products.products\_get\_by\_category(p\_id\_category integer)

RETURNS TABLE(id integer, c\_name text, c\_description text, c\_price numeric, c\_manufacturer\_name text, c\_model\_name text, c\_count integer)

SECURITY DEFINER

LANGUAGE plpgsql

AS

$$

BEGIN

IF NOT EXISTS(

SELECT 1

FROM categories.t\_category

WHERE id = p\_id\_category

FOR UPDATE) THEN

RAISE EXCEPTION 'Category with id % does not exist.', p\_id\_category;

END IF;

RETURN QUERY

SELECT p.id,

p.c\_name,

p.c\_description,

p.c\_price,

mfr.c\_name AS c\_manufacturer\_name,

m.c\_name AS c\_model\_name,

storages.inventory\_get\_product\_count(p.id) AS c\_count

FROM products.t\_product p

JOIN

products.t\_model m ON p.id\_model = m.id

JOIN

products.t\_manufacturer mfr ON m.id\_manufacturer = mfr.id

JOIN

products.t\_product\_property pp ON pp.id\_product = p.id

JOIN

categories.t\_category\_property cp ON pp.id\_category\_property = cp.id

JOIN

categories.t\_category c ON cp.id\_category = c.id

WHERE c.id = p\_id\_category

GROUP BY p.id, mfr.c\_name, m.c\_name;

IF NOT found THEN

RAISE EXCEPTION 'There are no products in this category with id %', p\_id\_category;

END IF;

END;

$$;

# ИСХОДНЫЙ КОД ФУНКЦИИ «storages.inventory\_get\_product\_count»

CREATE FUNCTION storages.inventory\_get\_product\_count(p\_id integer) RETURNS integer

SECURITY DEFINER

LANGUAGE plpgsql

AS

$$

DECLARE

v\_count integer;

BEGIN

IF NOT EXISTS(SELECT 1 FROM products.t\_product WHERE id = p\_id FOR UPDATE) THEN

RAISE EXCEPTION 'Product with id % does not exist.', p\_id;

END IF;

SELECT COUNT(p.id)

INTO v\_count

FROM products.t\_product\_instance pi

JOIN

products.t\_product p ON p.id = pi.id\_product

JOIN

storages.t\_inventory ti ON pi.id = ti.id\_product\_instance

WHERE ti.c\_event\_type NOT IN ('Sold', 'Scrapped')

AND p.id = p\_id

GROUP BY p.id;

RETURN v\_count;

END;

$$;

# ИСХОДНЫЙ КОД ФУНКЦИИ «clients.basket\_info\_set»

CREATE FUNCTION clients.basket\_info\_set(p\_id\_client integer, p\_id\_product integer, p\_count integer) RETURNS integer

SECURITY DEFINER

LANGUAGE plpgsql

AS

$$

DECLARE

v\_id integer;

v\_product\_cost numeric;

BEGIN

IF NOT EXISTS(SELECT 1

FROM clients.t\_client

WHERE id = p\_id\_client

FOR UPDATE) THEN

RAISE EXCEPTION 'Client with id % does not exist.', p\_id\_client;

END IF;

IF NOT EXISTS(SELECT 1

FROM products.t\_product

WHERE id = p\_id\_product

FOR UPDATE) THEN

RAISE EXCEPTION 'Product with id % does not exist.', p\_id\_product;

END IF;

SELECT c\_price

INTO v\_product\_cost

FROM products.t\_product

WHERE id = p\_id\_product

FOR UPDATE;

IF EXISTS(SELECT 1

FROM clients.t\_basket\_info

WHERE id\_client = p\_id\_client

AND id\_product = p\_id\_product

FOR UPDATE) THEN

UPDATE clients.t\_basket\_info

SET c\_count = c\_count + p\_count,

c\_batch\_cost = c\_batch\_cost + p\_count \* v\_product\_cost

WHERE id\_client = p\_id\_client

AND id\_product = p\_id\_product

RETURNING id INTO v\_id;

ELSE

INSERT INTO clients.t\_basket\_info (id\_client, id\_product, c\_count, c\_batch\_cost)

VALUES (p\_id\_client, p\_id\_product, p\_count, v\_product\_cost \* p\_count)

RETURNING id INTO v\_id;

END IF;

RETURN v\_id;

EXCEPTION

WHEN foreign\_key\_violation THEN

RAISE EXCEPTION 'Client or product with specified id does not exist.';

WHEN check\_violation THEN

RAISE EXCEPTION 'Invalid basket info data.';

WHEN OTHERS THEN

RAISE EXCEPTION 'Error adding basket info: %', sqlerrm;

END;

$$;