**Первое задание**

CREATE OR REPLACE FUNCTION stack.select\_orders\_by\_item\_name(item\_name TEXT)

RETURNS TABLE (

order\_id INT,

customer TEXT,

items\_count INT

) AS $$

BEGIN

RETURN QUERY

SELECT

oi.order\_id,

c.name AS customer,

COUNT(\*) AS items\_count

FROM stack.OrderItems oi

JOIN stack.Orders o ON oi.order\_id = o.row\_id

JOIN stack.Customers c ON o.customer\_id = c.row\_id

WHERE oi.name = item\_name

GROUP BY oi.order\_id, c.name;

END;

$$ LANGUAGE plpgsql;

**Второе задание**

CREATE OR REPLACE FUNCTION stack.calculate\_total\_price\_for\_orders\_group(group\_id INT)

RETURNS INT AS $$

DECLARE

total\_price INT;

BEGIN

WITH RECURSIVE order\_tree AS (

SELECT row\_id FROM stack.Orders WHERE row\_id = group\_id

UNION ALL

SELECT o.row\_id

FROM stack.Orders o

JOIN order\_tree ot ON o.parent\_id = ot.row\_id

)

SELECT COALESCE(SUM(oi.price), 0) INTO total\_price

FROM stack.OrderItems oi

JOIN order\_tree ot ON oi.order\_id = ot.row\_id;

RETURN total\_price;

END;

$$ LANGUAGE plpgsql;

**Третье задание**

SELECT DISTINCT c.name

FROM stack.Customers c

JOIN stack.Orders o ON c.row\_id = o.customer\_id

WHERE NOT EXISTS (

SELECT 1 FROM stack.Orders o2

WHERE o2.customer\_id = c.row\_id

AND EXTRACT(YEAR FROM o2.registered\_at) = 2020

AND NOT EXISTS (

SELECT 1 FROM stack.OrderItems oi

WHERE oi.order\_id = o2.row\_id

AND oi.name = 'Кассовый аппарат'

)

);