CREATE TABLE products (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

quantity INTEGER NOT NULL

);

CREATE TABLE operations\_log (

id SERIAL PRIMARY KEY,

product\_id INTEGER NOT NULL,

operation VARCHAR(10) NOT NULL,

quantity INTEGER NOT NULL,

operation\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (product\_id) REFERENCES products(id)

);

CREATE OR REPLACE PROCEDURE update\_stock(

product\_id INT,

operation VARCHAR,

quantity INT

)

LANGUAGE plpgsql

AS $$

BEGIN

IF operation = 'ADD' THEN

UPDATE products

SET quantity = quantity + quantity

WHERE id = product\_id;

ELSIF operation = 'REMOVE' THEN

IF (SELECT quantity FROM products WHERE id = product\_id) >= quantity THEN

UPDATE products

SET quantity = quantity - quantity

WHERE id = product\_id;

ELSE

RAISE EXCEPTION 'Недостаточно товара на складе';

END IF;

ELSE

RAISE EXCEPTION 'Недопустимый тип операции';

END IF;

INSERT INTO operations\_log (product\_id, operation, quantity)

VALUES (product\_id, operation, quantity);

COMMIT;

END;

$$;

CREATE OR REPLACE FUNCTION log\_product\_operation()

RETURNS TRIGGER AS $$

BEGIN

PERFORM update\_stock(NEW.id, 'ADD', NEW.quantity);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER product\_insert\_trigger

AFTER INSERT ON products

FOR EACH ROW

EXECUTE FUNCTION log\_product\_operation();