Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5 «Процедуры, функции, триггеры в PostgreSQL»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Казарян Т.Г.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql).

Практическое задание:

Вариант 2 (max - 8 баллов)

1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
2. Создать авторский триггер по варианту индивидуального задания.

Выполнение

Создайте хранимые процедуры:

1. **Вывести сведения о заказах заданного официанта на заданную дату.**

```
CREATE OR REPLACE FUNCTION restauran.get_waiter_orders_info(waiter_id INTEGER, date_order date) RETURNS TABLE (
    id_waiter INTEGER,
    fio CHARACTER VARYING(50),
    id_order integer,
    order_cost numeric,
    id_table integer
) AS
$$ BEGIN RETURN QUERY
SELECT employee.id_employee, employee.fio, orders.id_order, orders.cost, orders.id_table FROM restauran.orders
JOIN restauran.employee ON orders.id_waiter = employee.id_employee
WHERE orders.id_waiter = waiter_id AND orders.order_date::date = date_order;
END;
$$
LANGUAGE plpgsql;
```

```
restaurant_db=# CREATE OR REPLACE FUNCTION restauran.get_waiter_orders_info(waiter_id INTEGER, date_order date) RETURNS TABLE
restaurant_db(# id_waiter INTEGER,
restaurant_db(# fio CHARACTER VARYING(50),
restaurant_db(# id_order integer,
restaurant_db(# order_cost numeric,
restaurant_db(# id_table integer
restaurant_db(# ) AS
restaurant_db-# $$ BEGIN RETURN QUERY
restaurant_db$# SELECT employee.id_employee, employee.fio, orders.id_order, orders.cost, orders.id_table FROM restauran.orders
restaurant_db$# JOIN restauran.employee ON orders.id_waiter = employee.id_employee
restaurant_db$# WHERE orders.id_waiter = waiter_id AND orders.order_date::date = date_order;
restaurant_db$# END;
restaurant_db$# $$
restaurant_db-# LANGUAGE plpgsql
restaurant_db-# ;
CREATE FUNCTION
```

```
restaurant_db=# SELECT * FROM restauran.get_waiter_orders_info(1, '2023-12-15');
 id_waiter |         fio          | id_order | order_cost | id_table
-----------+----------------------+----------+------------+----------
         1 | Иванов Иван Иванович |        1 |     506.95 |        1
(1 строка)
```

- **Выполнить расчет стоимости заданного заказа.**

```
CREATE OR REPLACE FUNCTION restauran.get_order_cost(order_id integer) RETURNS TABLE (
        id_order integer,
        id_waiter INTEGER,
        fio CHARACTER VARYING(50),
        order_cost numeric,
        id_table integer
) AS
$$ BEGIN RETURN QUERY
SELECT orders.id_order, employee.id_employee, employee.fio, SUM(cost.price), orders.id_table FROM
restauran.orders
JOIN restauran.employee ON orders.id_waiter = employee.id_employee
JOIN restauran.order_composition oc ON orders.id_order = oc.id_order
JOIN restauran.cost ON oc.id_dish = cost.id_dish
WHERE orders.id_order = order_id
GROUP BY orders.id_order;
END;
$$
LANGUAGE plpgsql
```

```
restaurant_db=# CREATE OR REPLACE FUNCTION restauran.get_order_cost(order_id integer) RETURNS TABLE (
restaurant_db(# id_order integer,
restaurant_db(# id_waiter INTEGER,
restaurant_db(# fio CHARACTER VARYING(50),
restaurant_db(# order_cost bigint,
restaurant_db(# id_table integer
restaurant_db(# ) AS
restaurant_db-# $$ BEGIN RETURN QUERY
restaurant_db$# SELECT orders.id_order, employee.id_employee, employee.fio, SUM(cost.price), orders.id_table FROM restauran.orders
restaurant_db$# JOIN restauran.employee ON orders.id_waiter = employee.id_employee
restaurant_db$# JOIN restauran.order_composition oc ON orders.id_order = oc.id_order
restaurant_db$# JOIN restauran.cost ON oc.id_dish = cost.id_dish
restaurant_db$# WHERE orders.id_order = order_id
restaurant_db$# GROUP BY orders.id_order, employee.id_employee;
restaurant_db$# END;
restaurant_db$# $$
restaurant_db-# LANGUAGE plpgsql;
CREATE FUNCTION
restaurant_db=# SELECT * FROM restauran.get_order_cost(2);
 id_order | id_waiter |          fio           | order_cost | id_table
----------+-----------+------------------------+------------+----------
        2 |         3 | Петров Максим Максимович |        200 |        3
(1 строка)
```

- **Повышения оклада заданного сотрудника на 30 % при повышении его категории.**

```
CREATE OR REPLACE PROCEDURE up_oklad(employee_id integer)
LANGUAGE plpgsql as
$$
BEGIN
UPDATE restauran.employee
SET category = category + 1
WHERE employee.id_employee = employee_id;

UPDATE restauran.employee
SET rate = rate + 0.3
WHERE employee.id_employee = employee_id;

END;
$$;
```

```
restaurant_db=# CREATE OR REPLACE PROCEDURE up_oklad(employee_id integer)
restaurant_db-# LANGUAGE plpgsql as
restaurant_db-# $$
restaurant_db$# BEGIN
restaurant_db$# UPDATE restauran.employee
restaurant_db$# SET category = category + 1
restaurant_db$# WHERE employee.id_employee = employee_id;
restaurant_db$#
restaurant_db$# UPDATE restauran.employee
restaurant_db$# SET rate = rate + 0.3
restaurant_db$# WHERE employee.id_employee = employee_id;
restaurant_db$#
restaurant_db$# END;
restaurant_db$# $$;
CREATE PROCEDURE
```

```
 id_employee | passport_data |            fio            | id_post | category | rate
-------------+---------------+---------------------------+---------+----------+------
           1 | 4018 998445   | Иванов Иван Иванович      |       3 |        1 |    1
           2 | 5334 584036   | Лысенко Виктория Юрьевна  |       1 |        2 |    1
           3 | 3804 638490   | Петров Максим Максимович  |       3 |        2 |    1
           4 | 4567 377988   | Меньшин Александр Юрьевич |       1 |        1 |    1
(4 строки)


restaurant_db=# CALL up_oklad(1);
CALL
restaurant_db=# SELECT * FROM restauran.employee;
 id_employee | passport_data |            fio            | id_post | category | rate
-------------+---------------+---------------------------+---------+----------+------
           2 | 5334 584036   | Лысенко Виктория Юрьевна  |       1 |        2 |    1
           3 | 3804 638490   | Петров Максим Максимович  |       3 |        2 |    1
           4 | 4567 377988   | Меньшин Александр Юрьевич |       1 |        1 |    1
           1 | 4018 998445   | Иванов Иван Иванович      |       3 |        2 |  1.3
(4 строки)
```

Создайте необходимые триггеры:

- Триггер для автоматического обновления стоимости заказа

```
CREATE OR REPLACE FUNCTION update_order_cost()

RETURNS TRIGGER AS $$

BEGIN

	UPDATE restauran.orders

	SET orders.cost = (

	  SELECT SUM(cost.price), orders.id_table FROM restauran.orders

	  JOIN restauran.order_composition oc ON orders.id_order =
oc.id_order

	  JOIN restauran.cost ON oc.id_dish = cost.id_dish

	  WHERE orders.id_order = NEW.id_order

	  GROUP BY orders.id_order

	)

	WHERE orders.id_order = NEW.id_order;

  RETURN NEW;

END;

$$ LANGUAGE plpgsql;


CREATE OR REPLACE TRIGGER process_orders

AFTER INSERT ON restauran.order_composition

FOR EACH ROW

EXECUTE FUNCTION update_order_cost();
```

```
restaurant_db=# CREATE OR REPLACE FUNCTION update_order_cost()
restaurant_db-# RETURNS TRIGGER AS $$
restaurant_db$# BEGIN
restaurant_db$# UPDATE restauran.orders
restaurant_db$# SET orders.cost = (
restaurant_db$# SELECT SUM(cost.price), orders.id_table FROM restauran.orders
restaurant_db$# JOIN restauran.order_composition oc ON orders.id_order = oc.id_order
restaurant_db$# JOIN restauran.cost ON oc.id_dish = cost.id_dish
restaurant_db$# WHERE orders.id_order = NEW.id_order
restaurant_db$# GROUP BY orders.id_order
restaurant_db$# )
restaurant_db$# WHERE orders.id_order = NEW.id_order;
restaurant_db$#   RETURN NEW;
restaurant_db$# END;
restaurant_db$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
restaurant_db=#
restaurant_db=# CREATE OR REPLACE TRIGGER process_orders
restaurant_db-# AFTER INSERT ON restauran.order_composition
restaurant_db-# FOR EACH ROW
restaurant_db-# EXECUTE FUNCTION update_order_cost();
CREATE TRIGGER
```

Data Output | Messages | Notifications

	id_order [PK] integer	order_date timestamp without time zone	book_date timestamp without time zone	cost numeric	id_waiter integer	id_table integer
1	3	2024-01-18 18:00:00	[null]	1	2	2
2	1	2023-12-15 18:00:00	[null]	506.95	1	1
3	2	2023-12-18 18:36:00	2023-12-07 18:00:00	1263	3	3

3	2	1	2	Взято в работу	1
4	1	2	1	Готово	2
5	3	1	1	В ожидании	3

	id_dish integer	id_order integer	number_of_servings integer	ready_status character varying (14)	order_composition_id [PK] integer
1	2	3	1	Взято в работу	5
2	1	3	1	Взято в работу	4
3	2	1	2	Взято в работу	1
4	1	2	1	Готово	2
5	3	1	1	В ожидании	3

Query | Query History

```
1  SELECT * FROM restauran.orders
2  ORDER BY id_order ASC
```

Data Output | Messages | Notifications

	id_order [PK] integer	order_date timestamp without time zone	book_date timestamp without time zone	cost numeric	id_waiter integer	id_table integer
1	1	2023-12-15 18:00:00	[null]	506.95	1	1
2	2	2023-12-18 18:36:00	2023-12-07 18:00:00	1263	3	3
3	3	2024-01-18 18:00:00	[null]	700	2	2

Вывод

В ходе лабораторной работе была освоена работа с процедурами и триггерами.