**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

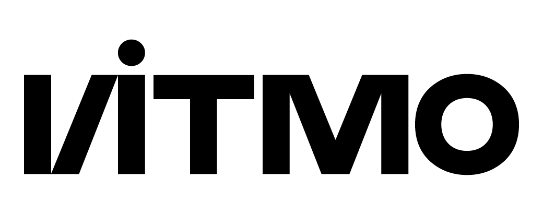
**Отчет**

по Лабораторной Работе № 5 по дисциплине **«Проектирование и реализация баз данных**»

Автор: Белов Г.П

Факультет: ИКТ Группа: К3239

Преподаватель: Говорова Марина Михайловна



Санкт-Петербург, 2023

# Содержание работы

**Цель работы:**

Овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

# Практическое задание:

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

# Выполнение работы:

**Создание хранимых процедур:**

1. О текущей сумме вклада и сумме начисленного за месяц процента для заданного клиента

create or replace procedure

get\_deposit\_payment(agreement\_id integer)

language sql as

$$

update deposit\_agreement da

set amountofpayments = (select coalesce(sum(dps.paymentamount), 0)

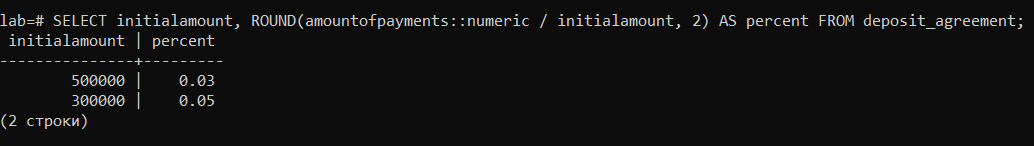
from deposit\_payment\_schedule dps

where dps.actualpaymentdate is not null

and da.id = dps.depositagreementid)

where da.id = agreement\_id;

$$;

 SELECT initialamount, ROUND(amountofpayments::numeric / initialamount, 2) AS percent FROM deposit\_agreement;

2.Найти клиента банка, имеющего максимальное количество кредитов на текущий день.

SELECT c.passportid, name

FROM client c

JOIN (

SELECT ca.clientid

FROM credit\_agreement ca

WHERE dateofclosing IS NULL

GROUP BY clientid

HAVING COUNT (ca.\*) = (

SELECT MAX(cnt)

FROM (

SELECT COUNT(\*) AS cnt

FROM credit\_agreement

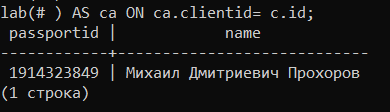
WHERE dateofclosing IS NULL

GROUP BY clientid

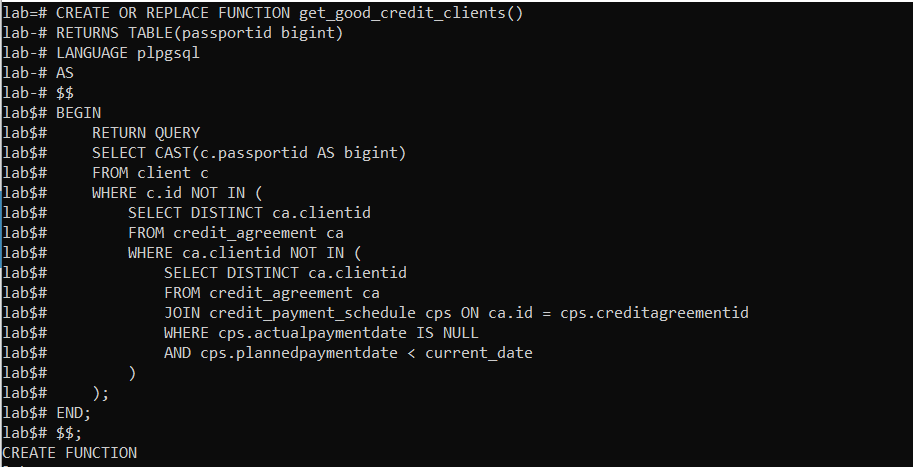
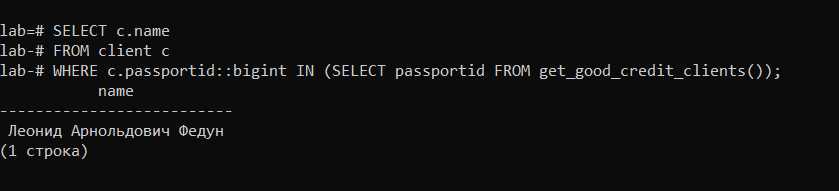
) AS subquery

)

) AS ca ON ca.clientid= c.id;



3.Найти клиентов банка, не имеющих задолженности по кредитам.



# Создание авторского триггера:

Триггер обновляет сумму выплат по вкладу при каждом обновлении таблицы с расписанием платежей:

# create or replace function count\_total\_deposit\_payment()

# returns trigger as $$

# begin

# call get\_deposit\_payment(new.id);

# return new;

# end;

# $$ language plpgsql;

# create or replace trigger change\_total\_deposit\_payment

# after update on deposit\_payment\_schedule

# for each row execute function count\_total\_deposit\_payment();

# 

# Вывод

# В процессе выполнения лабораторной работы были освоены практические навыки создания функций в PostgreSQL с помощью инструментов управления, таких как pgAdmin 4 и PSQL tool. Были разработаны функции, которые позволяют выбирать нужные данные из базы данных, а также были созданы соответствующие триггеры. Эти навыки могут быть полезны при работе с базами данных и помогут автоматизировать процесс получения необходимой информации и выполнения определенных действий при изменении данных в таблице.