

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: Цой С.А.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы:	3
Практическое задание	3
Выполнение	3
Вывод.....	10

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:

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

Выполнение

Создайте хранимые процедуры:

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

```
CREATE OR REPLACE FUNCTION bank.current_deposit_and_monthly_interest(p_client_id
INTEGER)
```

```
RETURNS TABLE (current_deposit INTEGER, monthly_interest DOUBLE PRECISION) AS
$$
```

```
BEGIN
```

```
RETURN QUERY
```

```
SELECT
```

```
"Initial_deposit_amount",
```

```
("Initial_deposit_amount" * "Interest_rate" / 100 / 12) as monthly_interest
```

```
FROM
```

```
bank."Deposit_contract"
```

```
WHERE
```

```
"Client_ID" = p_client_id;
```

```
END; $$
```

```

bank=# CREATE OR REPLACE FUNCTION bank.current_deposit_and_monthly_interest(p_client_id INTEGER)
bank=# RETURNS TABLE (current_deposit INTEGER, monthly_interest DOUBLE PRECISION) AS $$
bank$# BEGIN
bank$#   RETURN QUERY
bank$#     SELECT
bank$#       "Initial_deposit_amount",
bank$#       ("Initial_deposit_amount" * "Interest_rate" / 100 / 12) as monthly_interest
bank$#     FROM
bank$#       bank."Deposit_contract"
bank$#     WHERE
bank$#       "Client_ID" = p_client_id;
bank$# END; $$
bank=# LANGUAGE plpgsql;
CREATE FUNCTION

```

```

bank=# SELECT * FROM bank.current_deposit_and_monthly_interest(1);
current_deposit | monthly_interest
-----+-----
          100000 | 458.333333333333
          10000000 | 45833.3333333333
(2 строки)

```

```

bank=# SELECT * FROM bank.current_deposit_and_monthly_interest(15);
current_deposit | monthly_interest
-----+-----
          500000 | 1458.33333333333
          100000 | 458.33333333333
(2 строки)

```

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

```
CREATE OR REPLACE FUNCTION bank.GetClientWithMaxCredits()
```

```
RETURNS TABLE(Client_ID integer, Full_name character varying, CreditsCount
integer) AS $$
```

```
BEGIN
```

```
RETURN QUERY
```

```
SELECT
```

```
"Client"."Client_ID",
```

```
"Client"."Full_name",
```

```
CAST(SubQuery."CreditsCount" AS integer)
```

```
FROM
```

```
bank."Client"
```

```

JOIN (

SELECT

    "Credit_contract"."Client_ID",

    COUNT(*) AS "CreditsCount"

FROM

    bank."Credit_contract"

GROUP BY

    "Credit_contract"."Client_ID"

) AS SubQuery ON "Client"."Client_ID" = SubQuery."Client_ID"

WHERE

    SubQuery."CreditsCount" = (

        SELECT MAX("CreditsCount")

        FROM (

            SELECT

                COUNT(*) AS "CreditsCount"

            FROM

                bank."Credit_contract"

            GROUP BY

                "Credit_contract"."Client_ID"

        )

    );

END;

$$ LANGUAGE plpgsql STABLE;

```

```

bank=# CREATE OR REPLACE FUNCTION bank.GetClientWithMaxCredits()
bank=# RETURNS TABLE(Client_ID integer, Full_name character varying, CreditsCount integer) AS $$
bank$# BEGIN
bank$#     RETURN QUERY
bank$#     SELECT
bank$#         "Client"."Client_ID",
bank$#         "Client"."Full_name",
bank$#         CAST(SubQuery."CreditsCount" AS integer)
bank$#     FROM
bank$#         bank."Client"
bank$#     JOIN (
bank$#         SELECT
bank$#             "Credit_contract"."Client_ID",
bank$#             COUNT(*) AS "CreditsCount"
bank$#         FROM
bank$#             bank."Credit_contract"
bank$#         GROUP BY
bank$#             "Credit_contract"."Client_ID"
bank$#     ) AS SubQuery ON "Client"."Client_ID" = SubQuery."Client_ID"
bank$#     WHERE
bank$#         SubQuery."CreditsCount" = (
bank$#             SELECT MAX("CreditsCount")
bank$#             FROM (
bank$#                 SELECT
bank$#                     COUNT(*) AS "CreditsCount"
bank$#                 FROM
bank$#                     bank."Credit_contract"
bank$#                 GROUP BY
bank$#                     "Credit_contract"."Client_ID"
bank$#             )
bank$#         );
bank$# END;
bank$# $$ LANGUAGE plpgsql STABLE;
CREATE FUNCTION

```

```

bank=# SELECT * FROM bank.GetClientWithMaxCredits();
 client_id |          full_name          | creditscount
-----+-----+-----
       3 | Сидоров Сидор Сидорович   |           3
       1 | Иванов Иван Иванович       |           3
       5 | Васильев Василий Васильевич |           3
       2 | Петров Петр Петрович       |           3
(4 строки)

```

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

```

CREATE OR REPLACE FUNCTION bank.get_clients_without_debt()
RETURNS TABLE(
    Client_ID integer,
    Full_name character varying
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        "Client"."Client_ID",
        "Client"."Full_name"
    FROM

```

```

        bank."Client"
    INNER JOIN bank."Credit_contract" ON "Client"."Client_ID" =
"Credit_contract"."Client_ID"
    WHERE
        "Credit_contract"."Current_debt" = 0
    GROUP BY
        "Client"."Client_ID";
END; $$
LANGUAGE plpgsql;

```

```

bank=# CREATE OR REPLACE FUNCTION bank.get_clients_without_debt()
bank=# RETURNS TABLE(
bank(#    Client_ID integer,
bank(#    Full_name character varying
bank(# ) AS $$
bank$# BEGIN
bank$#     RETURN QUERY
bank$#     SELECT
bank$#         "Client"."Client_ID",
bank$#         "Client"."Full_name"
bank$#     FROM
bank$#         bank."Client"
bank$#     INNER JOIN bank."Credit_contract" ON "Client"."Client_ID" = "Credit_contract"."Client_ID"
bank$#     WHERE
bank$#         "Credit_contract"."Current_debt" = 0
bank$#     GROUP BY
bank$#         "Client"."Client_ID";
bank$# END; $$
bank=# LANGUAGE plpgsql;
CREATE FUNCTION

```

```

bank=# SELECT * FROM bank."Client";

```

Client_ID	Address	Phone	Full_name	E_mail
1	Россия, Москва, ул. Ленина, д.1, кв.1	+71234567890	Иванов Иван Иванович	ivanov@mail.ru
10	Россия, Саратов, ул. Достоевского, д.10, кв.10	+70123456789	Андреев Андрей Андреевич	andreev@mail.ru
11	Россия, Калининград, ул. Чайковского, д.11, кв.11	+71123456780	Борисов Борис Борисович	borisov@mail.ru
12	Россия, Иркутск, ул. Римского-Корсакова, д.12, кв.12	+72234567891	Григорьев Григорий Григорьевич	grigoriev@mail.ru
13	Россия, Ярославль, ул. Прокофьева, д.13, кв.13	+73345678902	Дмитриев Дмитрий Дмитриевич	dmitriev@mail.ru
14	Россия, Владивосток, ул. Шостаковича, д.14, кв.14	+74456789013	Евгеньев Евгений Евгеньевич	evgeniev@mail.ru
15	Россия, Мурманск, ул. Стравинского, д.15, кв.15	+75567890124	Жуков Сергей Сергеевич	zhukov@mail.ru
16	Россия, Красноярск, ул. Мусоргского, д.16, кв.16	+76678901235	Зайцев Виталий Иванович	zaitsev@mail.ru
17	Россия, Пермь, ул. Чайковского, д.17, кв.17	+77789012346	Ильин Илья Ильинич	ilin@mail.ru
18	Россия, Вологда, ул. Рахманинова, д.18, кв.18	+78890123457	Козлов Петр Алексеевич	kozlov@mail.ru
19	Россия, Чебоксары, ул. Скрябина, д.19, кв.19	+79901234568	Лебедев Артемий Иванович	lebedev@mail.ru
2	Россия, Санкт-Петербург, ул. Сталина, д.2, кв.2	+72345678901	Петров Петр Петрович	petrov@mail.ru
20	Россия, Тверь, ул. Щедрина, д.20, кв.20	+70012345679	Медведев Дмитрий Анатольевич	medvedev@mail.ru
3	Россия, Новосибирск, ул. Кирова, д.3, кв.3	+73456789012	Сидоров Сидор Сидорович	sidorov@mail.ru
4	Россия, Екатеринбург, ул. Дзержинского, д.4, кв.4	+74567890123	Алексеев Алексей Алексеевич	alekseev@mail.ru
5	Россия, Казань, ул. Гагарина, д.5, кв.5	+75678901234	Васильев Василий Васильевич	vasiliev@mail.ru
6	Россия, Ростов-на-Дону, ул. Чехова, д.6, кв.6	+76789012345	Михайлов Михаил Михайлович	mikhailov@mail.ru
7	Россия, Уфа, ул. Пушкина, д.7, кв.7	+77890123456	Николаев Николай Николаевич	nikolaev@mail.ru
8	Россия, Волгоград, ул. Гоголя, д.8, кв.8	+78901234567	Александров Александр Александрович	alexandrov@mail.ru
9	Россия, Краснодар, ул. Толстого, д.9, кв.9	+79012345678	Сергеев Сергей Сергеевич	sergeev@mail.ru

(20 строк)

```
bank=# SELECT * FROM bank.get_clients_without_debt();
 client_id |          full_name
-----+-----
      4 | Алексей Алексей Алексеевич
     10 | Андреев Андрей Андреевич
      9 | Сергеев Сергей Сергеевич
      7 | Николаев Николай Николаевич
     15 | Жуков Сергей Сергеевич
      6 | Михайлов Михаил Михайлович
     12 | Григорьев Григорий Григорьевич
     14 | Евгеньев Евгений Евгеньевич
      3 | Сидоров Сидор Сидорович
      1 | Иванов Иван Иванович
      5 | Васильев Василий Васильевич
      2 | Петров Петр Петрович
      8 | Александров Александр Александрович
(13 строк)
```

Создайте необходимые триггеры:

Триггер для обновления процентной ставки вклада или суммы выплаты по вкладу.

```
CREATE OR REPLACE FUNCTION bank.calculate_total_payout()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    NEW."Total_payout" := NEW."Initial_deposit_amount" + (NEW."Initial_deposit_amount"
* NEW."Interest_rate" / 100);
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
bank=# CREATE OR REPLACE FUNCTION bank.calculate_total_payout()
bank=# RETURNS TRIGGER AS $$
bank=# BEGIN
bank$#   NEW."Total_payout" := NEW."Initial_deposit_amount" + (NEW."Initial_deposit_amount" * NEW."Interest_rate" / 100);
bank$#   RETURN NEW;
bank$# END;
bank$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

```
CREATE TRIGGER tr_auto_update_total_payout
```

```
BEFORE UPDATE OF "Interest_rate", "Initial_deposit_amount"
```

```
ON bank."Deposit_contract"
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION bank.calculate_total_payout();
```



```

bank=# CREATE TRIGGER tr_auto_update_total_payout
bank=# BEFORE UPDATE OF "Interest_rate", "Initial_deposit_amount"
bank=# ON bank."Deposit_contract"
bank=# FOR EACH ROW
bank=# EXECUTE FUNCTION bank.calculate_total_payout();
ОШИБКА: триггер "tr_auto_update_total_payout" для отношения "Deposit_contract" уже существует

```

```

bank=# SELECT "Agreement_number", "Interest_rate", "Initial_deposit_amount", "Total_payout" FROM bank."Deposit_contract" WHERE "Agreement_number" = 1;

```

Agreement_number	Interest_rate	Initial_deposit_amount	Total_payout
1	10	100000	110000

(1 строка)

```

bank=# UPDATE bank."Deposit_contract" SET "Interest_rate" = 20 WHERE "Agreement_number" = 1;
UPDATE 1

```

```

UPDATE 1
bank=# SELECT "Agreement_number", "Interest_rate", "Initial_deposit_amount", "Total_payout" FROM bank."Deposit_contract" WHERE "Agreement_number" = 1;

```

Agreement_number	Interest_rate	Initial_deposit_amount	Total_payout
1	20	100000	120000

(1 строка)

Триггер для удаления строк с данными договора кредита, если кредит погашен:

```
CREATE OR REPLACE FUNCTION bank.auto_delete_credit_contract()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    IF NEW."Current_debt" = 0 THEN
```

```
        DELETE FROM bank."Credit_contract" WHERE "Contract_number" =
NEW."Contract_number";
```

```
        RETURN NULL;
```

```
    ELSE
```

```
        RETURN NEW;
```

```
    END IF;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```

CREATE TRIGGER
bank=# CREATE OR REPLACE FUNCTION bank.auto_delete_credit_contract()
bank=# RETURNS TRIGGER AS $$
bank$# BEGIN
bank$#     IF NEW."Current_debt" = 0 THEN
bank$#         DELETE FROM bank."Credit_contract" WHERE "Contract_number" = NEW."Contract_number";
bank$#         RETURN NULL;
bank$#     ELSE
bank$#         RETURN NEW;
bank$#     END IF;
bank$# END;
bank$# $$ LANGUAGE plpgsql;
CREATE FUNCTION

```

```
CREATE TRIGGER tr_auto_delete_credit_contract  
AFTER UPDATE OF "Current_debt"  
ON bank."Credit_contract"  
FOR EACH ROW  
WHEN (NEW."Current_debt" = 0)  
EXECUTE FUNCTION bank.auto_delete_credit_contract();
```

```
bank=# CREATE TRIGGER tr_auto_delete_credit_contract  
bank=# AFTER UPDATE OF "Current_debt"  
bank=# ON bank."Credit_contract"  
bank=# FOR EACH ROW  
bank=# WHEN (NEW."Current_debt" = 0)  
bank=# EXECUTE FUNCTION bank.auto_delete_credit_contract();  
CREATE TRIGGER
```

Вывод

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