

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет ИТМО»
Факультет инфокоммуникационных технологий

Лабораторная работа №3

Процедуры, функции, триггеры в PostgreSQL

Выполнил:
студент 2 курса ИКТ группа К32392,
Малышенко А. Р.

Проверила:
Говорова Марина Михайловна

Санкт-Петербург
2022

Цель работы:

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

Практическое задание:**Вариант 2**

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
- 2.1. Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу.
- 2.2. Создать авторский триггер по варианту индивидуального задания.

Выполнение задания

Вариант 12. БД “Прокат автомобилей”

Название: БД «СПБ.Драйв», каршеринг автомобилей.

Описание предметной области: Компания предоставляет прокат автомобилей.

Процедуры и функции

1) Выполнить списание автомобилей, выпущенных ранее заданного года

Команда:

CREATE PROCEDURE old_car(IN date_of_car DATE)

LANGUAGE SQL

AS \$\$

UPDATE "Car" SET "Is the car decommissioned" = 'TRUE' WHERE "Year of release" < date_of_car

AND "A note about the refund" = 'TRUE'

\$\$;

```
CREATE PROCEDURE old_car(IN date_of_car DATE)
LANGUAGE SQL
AS $$
    UPDATE "Car" SET "Is the car decommissioned" = 'TRUE' WHERE "Year of release" < date_of_car
    AND "A note about the refund" = 'TRUE'
$$;
```

До выполнения процедуры:

Car code number	Car model Special marks	Description	Year of release	Engine num	Body number	Rent price	Mileage	Date of last maintenance	A note about the refund	Is the car decommissioned	Registration
1	Aveo	Тёмно-зелёный, быстрый, старый	1989-03-25	17493571	44263998	50,00 ?	20000	2023-03-14	t	f	B679YT32
2	A4 Allroad Quattro	Светло-голубой, медленный, новый	2011-03-25	77870528	70221027	120,00 ?	3000	2023-02-17	t	f	A920BM56
3	M2	Серо-бежевый, быстрый, элитный	2015-03-25	35666883	70854080	150,00 ?	5500	2023-02-24	t	f	T228EK67
4	Nexia	Серебристо-зелёный, скоростной, не новый	2003-03-14	98562424	21718001	67,00 ?	7800	2023-02-13	t	f	E979YC45
5	Emgrand 7	Тёмно-бордовый, быстрый, с автопилотом	2021-03-25	86226869	18754839	125,00 ?	1000	2023-01-24	t	f	E449CX17
6	Lacetti	Сине-зелёный	2011-07-02	14847224	53192890	370,00 ?	21282	2023-02-17	t	f	I601EH47
7	Nexia	Желто-золотой	2000-12-08	90581365	57343634	377,00 ?	32277	2023-01-18	t	f	M648HT91
8	Lacetti	Ярко-красный	2008-07-30	36007059	89002374	279,00 ?	9634	2023-03-11	t	f	U208NP55
9	M2	Серебристо-серо-графитовый	2010-03-28	71518142	43794816	269,00 ?	22908	2023-02-15	t	f	Y315FT44
10	Aveo	Тёмно-бежевый	1990-11-18	83749148	91108688	371,00 ?	8636	2023-02-23	t	f	Z972F011

После выполнения по списанию машин старше 1995 года:

```
lab3_sem4=# CALL old_car('01.01.1995');
CALL
```

Car code number	Car model Special marks	Description	Year of release	Engine num	Body number	Rent price	Mileage	Date of last maintenance	A note about the refund	Is the car decommissioned	Registration
2	A4 Allroad Quattro	Светло-голубой, медленный, новый	2011-03-25	77870528	70221027	120,00 ?	3000	2023-02-17	t	f	A920BM56
3	M2	Серо-бежевый, быстрый, элитный	2015-03-25	35666883	70854080	150,00 ?	5500	2023-02-24	t	f	T228EK67
4	Nexia	Серебристо-зелёный, скоростной, не новый	2003-03-14	98562424	21718001	67,00 ?	7800	2023-02-13	t	f	E979YC45
5	Emgrand 7	Тёмно-бордовый, быстрый, с автопилотом	2021-03-25	86226869	18754839	125,00 ?	1000	2023-01-24	t	f	E449CK17
6	Lacetti	Сине-зелёный	2011-07-02	14847224	53192890	370,00 ?	21282	2023-02-17	t	f	I601EH47
7	Nexia	Желто-золотой	2000-12-08	90581365	57343634	377,00 ?	32277	2023-01-18	t	f	M648HT91
8	Lacetti	Ярко-красный	2008-07-30	36007059	89002374	279,00 ?	9634	2023-03-11	t	f	U208NP55
9	M2	Серебристо-серо-графитовый	2010-03-28	71518142	43794816	269,00 ?	22908	2023-02-15	t	f	Y315FT44
1	Aveo	Тёмно-зелёный, быстрый, старый	1989-03-25	17493571	44263998	50,00 ?	20000	2023-03-14	t	t	B679YT32
10	Aveo	Тёмно-бежевый	1990-11-18	83749148	91108688	371,00 ?	8636	2023-02-23	t	t	Z972F011

2) Выдачи автомобиля и расчета стоимости с учетом скидки постоянным клиентам

Команда:

CREATE OR REPLACE FUNCTION car_rental_and_discount(IN customer_number integer,
IN days_of_rent integer) RETURNS numeric

AS \$\$

DECLARE

discount numeric;

BEGIN

IF((SELECT COUNT(*) FROM "Contract" WHERE "Client code" = customer_number)
> 1) THEN

SELECT "Rent price" * 0.9 * days_of_rent FROM "Car" WHERE "A note about the
refund" = 'TRUE'

AND "Is the car decommissioned" = 'FALSE' LIMIT 1 INTO discount;

ELSE

SELECT "Rent price" * days_of_rent FROM "Car" WHERE "A note about the
refund" = 'TRUE'

AND "Is the car decommissioned" = 'FALSE' LIMIT 1 INTO discount;

END IF;

INSERT INTO "Contract" VALUES(DEFAULT, CURRENT_DATE + days_of_rent,
CURRENT_DATE, NULL, 5753, customer_number,

(SELECT "Car code" FROM "Car" WHERE "A note about the refund" = 'TRUE' AND
"Is the car decommissioned" = 'FALSE' LIMIT 1), 'in progress');

UPDATE "Car" SET "A note about the refund" = 'FALSE' WHERE "Car code" IN

(SELECT "Car code" FROM "Car" WHERE "A note about the refund" = 'TRUE' AND "Is
the car decommissioned" = 'FALSE' LIMIT 1);

RETURN discount;

END;

\$\$ LANGUAGE plpgsql;

```
CREATE OR REPLACE FUNCTION car_rental_and_discount(IN customer_number integer, IN days_of_rent integer) RETURNS numeric
AS $$
DECLARE
    discount numeric;
BEGIN

IF((SELECT COUNT(*) FROM "Contract" WHERE "Client code" = customer_number) > 1) THEN
    SELECT "Rent price" * 0.9 * days_of_rent FROM "Car" WHERE "A note about the refund" = 'TRUE'
    AND "Is the car decommissioned" = 'FALSE' LIMIT 1 INTO discount;
ELSE
    SELECT "Rent price" * days_of_rent FROM "Car" WHERE "A note about the refund" = 'TRUE'
    AND "Is the car decommissioned" = 'FALSE' LIMIT 1 INTO discount;
END IF;

INSERT INTO "Contract" VALUES(DEFAULT, CURRENT_DATE + days_of_rent, CURRENT_DATE, NULL, 5753, customer_number,
(SELECT "Car code" FROM "Car" WHERE "A note about the refund" = 'TRUE' AND "Is the car decommissioned" = 'FALSE' LIMIT 1), 'in progress');

UPDATE "Car" SET "A note about the refund" = 'FALSE' WHERE "Car code" IN
(SELECT "Car code" FROM "Car" WHERE "A note about the refund" = 'TRUE' AND "Is the car decommissioned" = 'FALSE' LIMIT 1);

RETURN discount;
END;
$$ LANGUAGE plpgsql;
```

До выполнения функции:

Contract code	Date and time of the end of the lease	Date and time of issue	Date and time of return	Employee code	Client code	Car code	Contract status
1	2022-09-10 10:00:00	2022-08-10 10:00:00	2022-10-11 10:00:00	5753	6413	1	done
2	2018-06-28 12:00:00	2018-05-12 10:00:00	2018-08-21 11:40:00	5753	2959	3	done
3	2019-01-23 18:00:00	2018-12-22 19:00:00	2019-01-26 20:00:00	3246	8259	4	done
4	2019-08-15 00:00:00	2019-05-08 00:00:00	2019-08-21 00:00:00	3246	7069	1	done
5	2012-10-01 17:42:00	2012-09-02 19:21:00	2012-10-22 13:04:00	5753	8259	5	done
6	2021-08-13 14:10:00	2021-08-02 19:19:00	2021-08-30 23:59:00	8554	7069	3	done
7	2021-10-27 00:00:00	2021-09-22 00:00:00	2021-11-01 00:00:00	5753	8942	10	done
8	2016-06-26 16:45:00	2016-06-10 09:30:00	2016-07-13 05:00:00	8554	8942	7	done
9	2020-06-01 03:00:00	2020-05-01 03:50:00	2020-08-19 01:01:00	5753	2959	8	done
10	2023-05-23 03:50:00	2023-05-10 03:00:00		5753	2959	2	in progress
11	2023-05-08 10:00:00	2023-02-27 10:00:00	2023-07-16 12:00:00	5753	8942	6	done

(11 строк)

После выполнения функции (выдача автомобиля на 10 дней + скидка постоянного клиента):

```
lab3_sem4=# SELECT * FROM car_rental_and_discount(8259, 10);
car_rental_and_discount
-----
1080.00
(1 строка)
```

Contract code	Date and time of the end of the lease	Date and time of issue	Date and time of return	Employee code	Client code	Car code	Contract status
1	2022-09-10 10:00:00	2022-08-10 10:00:00	2022-10-11 10:00:00	5753	6413	1	done
2	2018-06-28 12:00:00	2018-05-12 10:00:00	2018-08-21 11:40:00	5753	2959	3	done
3	2019-01-23 18:00:00	2018-12-22 19:00:00	2019-01-26 20:00:00	3246	8259	4	done
4	2019-08-15 00:00:00	2019-05-08 00:00:00	2019-08-21 00:00:00	3246	7069	1	done
5	2012-10-01 17:42:00	2012-09-02 19:21:00	2012-10-22 13:04:00	5753	8259	5	done
6	2021-08-13 14:10:00	2021-08-02 19:19:00	2021-08-30 23:59:00	8554	7069	3	done
7	2021-10-27 00:00:00	2021-09-22 00:00:00	2021-11-01 00:00:00	5753	8942	10	done
8	2016-06-26 16:45:00	2016-06-10 09:30:00	2016-07-13 05:00:00	8554	8942	7	done
9	2020-06-01 03:00:00	2020-05-01 03:50:00	2020-08-19 01:01:00	5753	2959	8	done
10	2023-05-23 03:50:00	2023-05-10 03:00:00		5753	2959	2	in progress
11	2023-05-08 10:00:00	2023-02-27 10:00:00	2023-07-16 12:00:00	5753	8942	6	done
12	2023-05-27 00:00:00	2023-05-17 00:00:00		5753	8259	1	in progress

(12 строк)

3) Для вычисления количества автомобилей заданной марки

Команда:

```
CREATE FUNCTION count_brand(car_brand text) RETURNS integer
```

```
LANGUAGE SQL
```

```
AS $$
```

```
    SELECT COUNT("Car"."Car code") FROM "Car"
```

```
    LEFT JOIN "Technical specifications" AS TS ON "Car"."Car model" = TS."Car model"
```

```
    WHERE TS."Car brand" = car_brand
```

```
$$;
```

```
CREATE FUNCTION count_brand(car_brand text) RETURNS integer
LANGUAGE SQL
AS $$
    SELECT COUNT("Car"."Car code") FROM "Car"
        LEFT JOIN "Technical specifications" AS TS ON "Car"."Car model" = TS."Car model"
        WHERE TS."Car brand" = car_brand
    $$;
```

Выполнение функции:

```
lab3_sem4=# SELECT * FROM count_brand('BMW');
count_brand
-----
                2
(1 строка)
```

Авторский триггер

При возврате машины автоматически засчитывает контракт, как закрытый

Команды:

```
CREATE FUNCTION status_update() RETURNS TRIGGER AS $my_table$
```

```
    BEGIN
```

```
        UPDATE "Contract" SET "Contract status" = 'done';
```

```
    RETURN NEW;
```

```
    END;
```

```
    $my_table$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER contract_status_update
```

```
    AFTER UPDATE OF "Date and time of return" ON "Contract"
```

```
    FOR EACH ROW
```

```
    EXECUTE FUNCTION status_update();
```

```
-- Функция
CREATE OR REPLACE FUNCTION status_update() RETURNS TRIGGER AS $my_table$
BEGIN
    UPDATE "Contract" SET "Contract status" = 'done';
    RETURN NEW;
END;
$my_table$ LANGUAGE plpgsql;

-- Триггер
CREATE OR REPLACE TRIGGER contract_status_update
AFTER UPDATE OF "Date and time of return" ON "Contract"
FOR EACH ROW
EXECUTE FUNCTION status_update();
```

До создания триггера и его активации:

Contract code	Date and time of the end of the lease	Date and time of issue	Date and time of return	Employee code	Client code	Car code	Contract status
1	2022-09-10 10:00:00	2022-08-10 10:00:00	2022-10-11 10:00:00	5753	6413	1	done
2	2018-06-28 12:00:00	2018-05-12 10:00:00	2018-08-21 11:40:00	5753	2959	3	done
3	2019-01-23 18:00:00	2018-12-22 19:00:00	2019-01-26 20:00:00	3246	8259	4	done
4	2019-08-15 00:00:00	2019-05-08 00:00:00	2019-08-21 00:00:00	3246	7069	1	done
5	2012-10-01 17:42:00	2012-09-02 19:21:00	2012-10-22 13:04:00	5753	8259	5	done
6	2021-08-13 14:10:00	2021-08-02 19:19:00	2021-08-30 23:59:00	8554	7069	3	done
7	2021-10-27 00:00:00	2021-09-22 00:00:00	2021-11-01 00:00:00	5753	8942	10	done
8	2016-06-26 16:45:00	2016-06-10 09:30:00	2016-07-13 05:00:00	8554	8942	7	done
9	2020-06-01 03:00:00	2020-05-01 03:50:00	2020-08-19 01:01:00	5753	2959	8	done
10	2023-05-23 03:50:00	2023-05-10 03:00:00		5753	2959	2	in progress
11	2023-05-08 10:00:00	2023-02-27 10:00:00	2023-07-16 12:00:00	5753	8942	6	done
12	2023-05-27 00:00:00	2023-05-17 00:00:00		5753	8259	1	in progress

(12 строк)

После добавления триггера и искусственного возврата машины:

```
lab3_sem4=# UPDATE "Contract" SET "Date and time of return" = CURRENT_DATE WHERE "Contract code" = 10;
UPDATE 1
```

Contract code	Date and time of the end of the lease	Date and time of issue	Date and time of return	Employee code	Client code	Car code	Contract status
1	2022-09-10 10:00:00	2022-08-10 10:00:00	2022-10-11 10:00:00	5753	6413	1	done
2	2018-06-28 12:00:00	2018-05-12 10:00:00	2018-08-21 11:40:00	5753	2959	3	done
3	2019-01-23 18:00:00	2018-12-22 19:00:00	2019-01-26 20:00:00	3246	8259	4	done
4	2019-08-15 00:00:00	2019-05-08 00:00:00	2019-08-21 00:00:00	3246	7069	1	done
5	2012-10-01 17:42:00	2012-09-02 19:21:00	2012-10-22 13:04:00	5753	8259	5	done
6	2021-08-13 14:10:00	2021-08-02 19:19:00	2021-08-30 23:59:00	8554	7069	3	done
7	2021-10-27 00:00:00	2021-09-22 00:00:00	2021-11-01 00:00:00	5753	8942	10	done
8	2016-06-26 16:45:00	2016-06-10 09:30:00	2016-07-13 05:00:00	8554	8942	7	done
9	2020-06-01 03:00:00	2020-05-01 03:50:00	2020-08-19 01:01:00	5753	2959	8	done
10	2023-05-23 03:50:00	2023-05-10 03:00:00	2023-05-17 00:00:00	5753	2959	2	done
11	2023-05-08 10:00:00	2023-02-27 10:00:00	2023-07-16 12:00:00	5753	8942	6	done
12	2023-05-27 00:00:00	2023-05-17 00:00:00		5753	8259	1	in progress

(12 строк)

Модифицированный триггер

```
create or replace function fn_check_time_punch() returns
trigger as $psql$
    declare
        time_and_out RECORD;

    begin
        SELECT punch_time, is_out_punch INTO time_and_out
            FROM time_punch
            WHERE employee_id = new.employee_id
            ORDER BY punch_time DESC LIMIT 1;

        IF(time_and_out is NULL AND new.is_out_punch = 'TRUE') THEN
            RETURN NULL;
        END IF;

        IF(new.pucnh_time <= time_and_out.punch_time) THEN
            RETURN NULL;
        END IF;

        IF(new.punch_time > NOW()) THEN
            RETURN NULL;
        END IF;

        IF(new.is_out_punch = time_and_out.is_out_punch) THEN
            RETURN NULL;
        END IF;

        RETURN new;
    end;
$psql$ language plpgsql;
```

Выводы:

Я овладел навыками по созданию и использованию хранимых процедур, функций и триггеров, а также их модифицирование в базе данных PostgreSQL с использованием консоли SQL Shell.