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 и триггеры в PostgreSQL»

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

Автор: Юркин А.С.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы

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

Практическое задание

1. Создать процедуры/функции согласно индивидуальному заданию
2. Модифицировать триггер на проверку корректности входа и выхода сотрудника с максимальным учётом «узких» мест некорректных данных по входу и выходу
3. Создать авторский триггер по варианту индивидуального задание

Вариант 12. БД «Прокат автомобилей»

Задание 4. Создать хранимые процедуры:

- Выполнить списание автомобилей, выпущенных ранее заданного года.
- Выдачи автомобиля и расчета стоимости с учетом скидки постоянным клиентам.
- Для вычисления количества автомобилей заданной марки.

Процедуры:

1.

```
CREATE OR REPLACE PROCEDURE WriteOffCars(BeforeYear INTEGER)
LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM car
    WHERE car_year < BeforeYear;
    COMMIT;
END;
$$;
```

```
dbalexyrkn=# select * from car;
 id | deposit | registration_number | plate_number | engine_number | body_number | car_year | mileage | car_price | inspection_date |     remarks     | return_receipt | time_in_rent | model
----+---------+---------------------+--------------+---------------+-------------+----------+---------+-----------+-----------------+-----------------+----------------+--------------+-------
  1 |     500 | XYZ1234             | ABC-123      | ENG001        | BODY001     |     2023 |    1200 |     25000 | 2023-10-01      | No remarks      | false          |            0 |     1
  2 |     600 | LMN5678             | DEF-456      | ENG002        | BODY002     |     2023 |    5400 |     30000 | 2023-09-20      | Good condition  | false          |            0 |     2
  3 |     700 | OPQ9101             | GHI-789      | ENG003        | BODY003     |     2022 |    6800 |     35000 | 2023-07-18      | Minor scratches | false          |            0 |     3
  4 |     800 | OPQ9444             | GHI-666      | ENG004        | BODY004     |     2020 |   16800 |     55000 | 2023-07-18      | Minor scratches | false          |            0 |     3
  5 |     800 | OPQ9411             | GHI-777      | ENG005        | BODY005     |     2010 |   36800 |     15000 | 2023-11-09      | Minor scratches | false          |            0 |     1
(5 rows)

dbalexyrkn=# call WriteOffCars(2011);
CALL
dbalexyrkn=# select * from car;
 id | deposit | registration_number | plate_number | engine_number | body_number | car_year | mileage | car_price | inspection_date |     remarks     | return_receipt | time_in_rent | model
----+---------+---------------------+--------------+---------------+-------------+----------+---------+-----------+-----------------+-----------------+----------------+--------------+-------
  1 |     500 | XYZ1234             | ABC-123      | ENG001        | BODY001     |     2023 |    1200 |     25000 | 2023-10-01      | No remarks      | false          |            0 |     1
  2 |     600 | LMN5678             | DEF-456      | ENG002        | BODY002     |     2023 |    5400 |     30000 | 2023-09-20      | Good condition  | false          |            0 |     2
  3 |     700 | OPQ9101             | GHI-789      | ENG003        | BODY003     |     2022 |    6800 |     35000 | 2023-07-18      | Minor scratches | false          |            0 |     3
  4 |     800 | OPQ9444             | GHI-666      | ENG004        | BODY004     |     2020 |   16800 |     55000 | 2023-07-18      | Minor scratches | false          |            0 |     3
(4 rows)
```

2.

```
CREATE OR REPLACE PROCEDURE CreateContract(
    ClientID INTEGER,
    CarID INTEGER,
    RentalDays INTEGER,
    EmployeeID INTEGER,
    InsuranceID INTEGER
)
LANGUAGE plpgsql
AS $$
DECLARE
    CarModelID INTEGER;
    SalePercentage FLOAT;
    RentalRate INTEGER;
    TotalCost INTEGER;
BEGIN
    SELECT model INTO CarModelID
    FROM car
    WHERE id = CarID;

    SELECT sale INTO SalePercentage
    FROM client
    WHERE id = ClientID;

    IF RentalDays <= 2 THEN
        SELECT price_12 INTO RentalRate
        FROM car_price
        WHERE model = CarModelID;
    ELSIF RentalDays <= 7 THEN
        SELECT price_37 INTO RentalRate
        FROM car_price
        WHERE model = CarModelID;
    ELSE
        SELECT price_7 INTO RentalRate
        FROM car_price
```

```
        WHERE model = CarModelID;
    END IF;

    TotalCost := RentalRate * RentalDays * (1 - SalePercentage / 100);

    INSERT INTO contract (client, car, price, startdate, enddate, STATUS,
payment_status, employee, insurance)
    VALUES (ClientID, CarID, TotalCost, CURRENT_DATE, CURRENT_DATE + RentalDays,
'active', 'unpaid', EmployeeID, InsuranceID);

    COMMIT;
END;
$$;
```

```
CREATE PROCEDURE
dbalexyrkn=# call CreateContract(1, 1, 5, 1, 1);
CALL
dbalexyrkn=# select * from contract;
 id | price |       startdate        |        enddate         |  status   | payment_status | client | employee | car | insurance
----+-------+------------------------+------------------------+-----------+----------------+--------+----------+-----+-----------
  1 |   250 | 2023-10-21 00:00:00+03 | 2023-10-26 00:00:00+03 | Active    | Paid           |      1 |        1 |   1 |         1
  2 |   280 | 2023-10-22 00:00:00+03 | 2023-10-27 00:00:00+03 | Active    | Paid           |      2 |        2 |   2 |         2
  3 |   320 | 2023-11-05 00:00:00+03 | 2023-11-10 12:00:00+03 | Completed | Paid           |      3 |        3 |   3 |         3
  7 |  3375 | 2023-12-06 00:00:00+03 | 2023-12-11 00:00:00+03 | Active    | Unpaid         |      1 |        1 |   1 |         1
(4 rows)
```

3.

```
CREATE OR REPLACE PROCEDURE CountCars(BrandName VARCHAR)
LANGUAGE plpgsql
AS $$
DECLARE
    CarCount INTEGER;
BEGIN
    SELECT COUNT(*)
    INTO CarCount
    FROM car
    WHERE model IN (SELECT id FROM model WHERE name = BrandName);

    RAISE NOTICE 'Number of cars for the brand %: %', BrandName, CarCount;
END;
$$;
```

```
dbalexyrkn=# select * from car;
 id | deposit | registration_number | plate_number | engine_number | body_number | car_year | mileage | car_price | inspection_date |     remarks     | return_receipt | time_in_rent | model
----+---------+---------------------+--------------+---------------+-------------+----------+---------+-----------+-----------------+-----------------+----------------+--------------+-------
  1 |     500 | XYZ1234             | ABC-123      | ENG001        | BODY001     |     2023 |    1200 |     25000 | 2023-10-01      | No remarks      | false          |            0 |     1
  2 |     600 | LMN5678             | DEF-456      | ENG002        | BODY002     |     2023 |    5400 |     30000 | 2023-09-20      | Good condition  | false          |            0 |     2
  3 |     700 | OPQ9101             | GHI-789      | ENG003        | BODY003     |     2022 |    6800 |     35000 | 2023-07-18      | Minor scratches | false          |            0 |     3
  4 |     800 | OPQ9444             | GHI-666      | ENG004        | BODY004     |     2020 |   16800 |     55000 | 2023-07-18      | Minor scratches | false          |            0 |     3
(4 rows)

dbalexyrkn=# select * from model;
 id |   name   |  techs   |                  description                  | time
----+----------+----------+-----------------------------------------------+------
  1 | Sedan X  | Hybrid   | A compact sedan with a hybrid powertrain.     |    0
  2 | Coupe Z  | Electric | Two-door sports car with an electric powertrain. |    0
  3 | SUV Y    | Diesel   | Spacious SUV with a diesel engine.            |    0
(3 rows)

dbalexyrkn=# call CountCars('SUV Y');
NOTICE:  Number of cars for the brand SUV Y: 2
CALL
dbalexyrkn=#
```

Триггер

Триггер автоматически изменяет дату окончания контракта и пересчитывает итоговую стоимость при добавлении новой записи в таблицу продлений.

```
CREATE OR REPLACE FUNCTION update_contract_duration_and_price()
RETURNS TRIGGER AS $$
DECLARE
    CurrentEndDate TIMESTAMP;
    NewEndDate TIMESTAMP;
```

```
    RentalDays INTEGER;
    BasePrice INTEGER;
    SalePercentage FLOAT;
    NewPrice INTEGER;
BEGIN
    SELECT enddate, (SELECT sale FROM client WHERE id = (SELECT client FROM
contract WHERE id = NEW.contract))
    INTO CurrentEndDate, SalePercentage
    FROM contract
    WHERE id = NEW.contract;

    NewEndDate := CurrentEndDate + INTERVAL '1 hour' * NEW.hours;

    RentalDays := DATE_PART('day', NewEndDate - (SELECT startdate FROM contract
WHERE id = NEW.contract));

    IF RentalDays <= 2 THEN
        SELECT price_12 INTO BasePrice FROM car_price WHERE model = (SELECT model
FROM car WHERE id = (SELECT car FROM contract WHERE id = NEW.contract));
    ELSIF RentalDays <= 7 THEN
        SELECT price_37 INTO BasePrice FROM car_price WHERE model = (SELECT model
FROM car WHERE id = (SELECT car FROM contract WHERE id = NEW.contract));
    ELSE
        SELECT price_7 INTO BasePrice FROM car_price WHERE model = (SELECT model
FROM car WHERE id = (SELECT car FROM contract WHERE id = NEW.contract));
    END IF;

    NewPrice := (BasePrice * RentalDays) * (1 - SalePercentage / 100);

    UPDATE contract
    SET enddate = NewEndDate, price = NewPrice
    WHERE id = NEW.contract;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_update_contract_duration_and_price
AFTER INSERT ON prolongation
FOR EACH ROW
EXECUTE FUNCTION update_contract_duration_and_price();
```

```
dbalexyrkn=# select * from contract;
 id | price |       startdate        |        enddate         |  status   | payment_status | client | employee | car | insurance
----+-------+------------------------+------------------------+-----------+----------------+--------+----------+-----+-----------
  1 |   250 | 2023-10-21 00:00:00+03 | 2023-10-26 00:00:00+03 | Active    | Paid           |      1 |        1 |   1 |         1
  2 |   280 | 2023-10-22 00:00:00+03 | 2023-10-27 00:00:00+03 | Active    | Paid           |      2 |        2 |   2 |         2
  3 |   320 | 2023-11-05 00:00:00+03 | 2023-11-10 12:00:00+03 | Completed | Paid           |      3 |        3 |   3 |         3
  7 |  3375 | 2023-12-06 00:00:00+03 | 2023-12-11 00:00:00+03 | Active    | Unpaid         |      1 |        1 |   1 |         1
(4 rows)

dbalexyrkn=# insert into prolongation (id, contract, date, hours) values (1, 7, '2023-12-06', 48)
;
INSERT 0 1
dbalexyrkn=# select * from contract;
 id | price |       startdate        |        enddate         |  status   | payment_status | client | employee | car | insurance
----+-------+------------------------+------------------------+-----------+----------------+--------+----------+-----+-----------
  1 |   250 | 2023-10-21 00:00:00+03 | 2023-10-26 00:00:00+03 | Active    | Paid           |      1 |        1 |   1 |         1
  2 |   280 | 2023-10-22 00:00:00+03 | 2023-10-27 00:00:00+03 | Active    | Paid           |      2 |        2 |   2 |         2
  3 |   320 | 2023-11-05 00:00:00+03 | 2023-11-10 12:00:00+03 | Completed | Paid           |      3 |        3 |   3 |         3
  7 |  4725 | 2023-12-06 00:00:00+03 | 2023-12-13 00:00:00+03 | Active    | Unpaid         |      1 |        1 |   1 |         1
(4 rows)

dbalexyrkn=#
```

Вывод

В ходе лабораторной работы были изучены и написаны функции и триггеры для базы данных PostgreSQL.