Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет Инфокоммуникационных технологий

Лабораторная работа №3 по теме «процедуры, функции, триггеры в PostgreSQL» по дисциплине «Проектирование и реализация баз данных»

Выполнил:

студент 2 курса К32421 группы

Гафаров Данил Альбертович

Преподаватель:

Говорова Марина Михайловна

Санкт-Петербург

2023

Цель работы

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

Практическое задание

Практическое задание по 1 варианту:

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

Выполнение

FUNCTION №1

Вывести все сведения о владельце автомобиля по заданному, как параметр номеру автомобиля.

CREATE FUNCTION driver_by_plate(target_plate varchar) RETURNS TABLE(full_name varchar, license char, phone varchar, adress varchar, passport char) AS

$$

SELECT full_name, license_number, phone, adress, passport FROM

"GIBDD"."CAR_OWNER", "GIBDD"."CITIZEN", "GIBDD"."REG_CAR"

WHERE target_plate = license_plate_reg AND id_citizen = id_citizen_owner AND license_number_reg = license_number

$$ LANGUAGE SQL;

Пример вызова:

SELECT * FROM driver_by_plate('E592TC78')

```
GIBDD=# CREATE FUNCTION driver_by_plate(target_plate varchar) RETURNS TABLE(full_name varcha
r, license char, phone varchar, adress varchar, passport char) AS $$
GIBDD$# SELECT full_name, license_number, phone, adress, passport FROM
GIBDD$# "GIBDD"."CAR_OWNER", "GIBDD"."CITIZEN", "GIBDD"."REG_CAR"
GIBDD$# WHERE target_plate = license_plate_reg AND id_citizen = id_citizen_owner AND license
_number_reg = license_number$$ LANGUAGE SQL;
CREATE FUNCTION
GIBDD=# SELECT * FROM driver_by_plate('E592TC78')
GIBDD-# ;
        full_name        |  license   |    phone     |         adress          |  passport
-------------------------+------------+--------------+-------------------------+------------

 Маратов Марат Маратович | 5008185092 | +79962369879 | ул. Ломоносова 9б, кв.1 | 9214948532
(1 row)
```

FUNCTION №2

Вывести данные инспектора, оштрафовавшего одного и того же водителя более одного раза.

CREATE FUNCTION officers_multiple_fined() RETURNS TABLE(full_name varchar, id_officer integer, role varchar, id_dep integer, phone varchar, adress varchar, passport char) AS $$

SELECT DISTINCT full_name, id_officer, "role", id_dep_officer, phone, adress, passport FROM "GIBDD"."OFFICER"

JOIN (SELECT id_officer_dtp, id_citizen_participant, count(id_dtp) FROM "GIBDD"."DTP" JOIN "GIBDD"."DTP_PARTICIPANT" ON id_dtp = id_dtp_participant WHERE status = 'Виновник' GROUP BY id_officer_dtp, id_citizen_participant HAVING COUNT(id_dtp) > 1) AS prep

ON id_officer = id_officer_dtp JOIN "GIBDD"."CITIZEN" ON id_citizen = id_citizen_officer

$$ LANGUAGE SQL;

```
GIBDD=# CREATE FUNCTION officers_multiple_fined() RETURNS TABLE(full_name varchar, id_officer integer, role varchar, id_
dep integer, phone varchar, adress varchar, passport char) AS $$
GIBDD$# SELECT DISTINCT full_name, id_officer, "role", id_dep_officer, phone, adress, passport FROM "GIBDD"."OFFICER" JO
IN (SELECT id_officer_dtp, id_citizen_participant, count(id_dtp) FROM "GIBDD"."DTP" JOIN "GIBDD"."DTP_PARTICIPANT" ON id
_dtp = id_dtp_participant WHERE status = 'Виновник' GROUP BY id_officer_dtp, id_citizen_participant HAVING COUNT(id_dtp)
 > 1) AS prep ON id_officer = id_officer_dtp JOIN "GIBDD"."CITIZEN" ON id_citizen = id_citizen_officer
GIBDD$# $$ LANGUAGE SQL;
CREATE FUNCTION
GIBDD=# SELECT * FROM officers_multiple_fined()
GIBDD-# ;
  full_name  | id_officer |       role        | id_dep |    phone     |        adress        |  passport
-------------+------------+-------------------+--------+--------------+----------------------+------------
 Джон Гаджет |          1 | Старший инспектор |      4 | +79999364491 | ул. Якорная 6, кв.13 | 8717287992
(1 row)
```

FUNCTION №3

Вывести количество нарушений, повлекших лишение прав в заданном, как параметр районе.

CREATE FUNCTION suspensions_by_district(target_district varchar)

RETURNS TABLE(amount bigint) AS

$$

SELECT COUNT(*) FROM "GIBDD"."VIOLATION"

JOIN "GIBDD"."VIOLATIONS_CATALOG"

ON violation_code = violation_id_code

WHERE license_suspension_time IS NOT null AND vio_district = target_district

$$ LANGUAGE SQL;

```
GIBDD=# CREATE FUNCTION suspensions_by_district(target_district varchar)
GIBDD-# RETURNS TABLE(amount bigint) AS
GIBDD-# $$
GIBDD$# SELECT COUNT(*) FROM "GIBDD"."VIOLATION"
GIBDD$# JOIN "GIBDD"."VIOLATIONS_CATALOG"
GIBDD$# ON violation_code = violation_id_code
GIBDD$# WHERE license_suspension_time IS NOT null AND vio_district = target_district
GIBDD$# $$ LANGUAGE SQL;
CREATE FUNCTION
```

```
GIBDD=# SELECT * FROM suspensions_by_district('Московский');
 amount
--------
      2
(1 row)
```

TRIGGER

Создадим триггер добавляющий записи в журнал событий

Создание таблицы с записями:

CREATE TABLE citizen_log

GIBDD-# (action_time timestamp without time zone,action_type varchar,

GIBDD(# name_old varchar, name_new varchar, adress_old varchar, adress_new varchar,

GIBDD(# phone_old varchar, phone_new varchar, passport_old char (10), passport_new char (10));

Создадим триггерную функцию:

CREATE OR REPLACE FUNCTION add_in_citizen_log() RETURNS TRIGGER AS

$$

DECLARE

old_name varchar(50);

new_name varchar(50);

old_adress varchar(50); new_adress varchar(50);

old_phone varchar(12); new_phone varchar(12);

old_passport char(10); new_passport char(10);

```
BEGIN

IF TG_OP='INSERT' THEN

new_name = NEW."full_name";

new_adress = NEW."adress";

new_phone = NEW."phone";

new_passport = NEW."passport";

INSERT INTO citizen_log(action_time, action_type,name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)

VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);

RETURN NEW;

ELSIF TG_OP='UPDATE' THEN

old_name = OLD."full_name";

old_adress = OLD."adress";

old_phone = OLD."phone";

old_passport = OLD."passport";

new_name = NEW."full_name";

new_adress = NEW."adress";

new_phone = NEW."phone";

new_passport = NEW."passport";

INSERT INTO citizen_log(action_time, action_type, name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)

VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);

RETURN NEW;

ELSIF TG_OP='DELETE' THEN

old_name = OLD."full_name";

old_adress = OLD."adress";
```

```
old_phone = OLD."phone";

old_passport = OLD."passport";

INSERT INTO citizen_log(action_time, action_type, name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)

VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);

RETURN OLD;

END IF;

END;

$$ LANGUAGE plpgsql;
```

Создадим триггер:

```
CREATE TRIGGER citizen_log_tg AFTER INSERT OR UPDATE OR DELETE ON

"GIBDD"."CITIZEN" FOR EACH ROW EXECUTE PROCEDURE

add_in_citizen_log();
```

Добавим записи:

```
INSERT INTO "GIBDD"."CITIZEN" ("full_name", "phone", "adress", "passport")

VALUES('Николаев Николай Николаевич', '+79132639280', 'ул. Заречная 16, кв.4', '1862927180');

UPDATE "GIBDD"."CITIZEN" SET "full_name" = 'Петров Петр Петрович'

WHERE "passport" = '1862927180';

DELETE FROM "GIBDD"."CITIZEN" WHERE "passport" = '1862927180';

SELECT action_time, action_type, name_old, name_new FROM citizen_log;
```

Скриншот работы в psql:

```
GIBDD=# INSERT INTO "GIBDD"."CITIZEN" ("full_name", "phone", "adress", "passport")
GIBDD-# VALUES('Николаев Николай Николаевич', '+79132639280', 'ул. Заречная 16, кв.4', '1862927180');
INSERT 0 1
GIBDD=# UPDATE "GIBDD"."CITIZEN" SET "full_name" = 'Петров Петр Петрович'
GIBDD-# WHERE "passport" = '1862927180';
UPDATE 1
GIBDD=# DELETE FROM "GIBDD"."CITIZEN" WHERE "passport" = '1862927180';
```

```
GIBDD=# SELECT action_time, action_type, name_old, name_new FROM citizen_log;
        action_time         | action_type |          name_old           |          name_new
----------------------------+-------------+-----------------------------+----------------------------
 2023-08-13 15:08:03.268836 | INSERT      |                             | Николаев Николай Николаевич
 2023-08-13 15:10:00.813617 | UPDATE      | Николаев Николай Николаевич | Петров Петр Петрович
 2023-08-13 15:10:56.810409 | DELETE      | Петров Петр Петрович        |
(3 rows)
```

Скриншоты создания таблицы, функции и триггера:

```
GIBDD=# CREATE TABLE citizen_log
GIBDD-# (action_time timestamp without time zone,action_type varchar,
GIBDD(# name_old varchar, name_new varchar, adress_old varchar, adress_new varchar,
GIBDD(# phone_old varchar, phone_new varchar, passport_old char (10), passport_new char (10));
CREATE TABLE
```

```
GIBDD=# CREATE OR REPLACE FUNCTION add_in_citizen_log() RETURNS TRIGGER AS
GIBDD-# $$
GIBDD$# DECLARE
GIBDD$# old_name varchar(50);
GIBDD$# new_name varchar(50);
GIBDD$# old_adress varchar(50); new_adress varchar(50);
GIBDD$# old_phone varchar(12); new_phone varchar(12);
GIBDD$# old_passport char(10); new_passport char(10);
GIBDD$# BEGIN
GIBDD$# IF TG_OP='INSERT' THEN
GIBDD$# new_name = NEW."full_name";
GIBDD$# new_adress = NEW."adress";
GIBDD$# new_phone = NEW."phone";
GIBDD$# new_passport = NEW."passport";
GIBDD$# INSERT INTO citizen_log(action_time, action_type,name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)
GIBDD$# VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);
GIBDD$# RETURN NEW;
GIBDD$# ELSIF TG_OP='UPDATE' THEN
GIBDD$# old_name = OLD."full_name";
GIBDD$# old_adress = OLD."adress";
GIBDD$# old_phone = OLD."phone";
GIBDD$# old_passport = OLD."passport";
GIBDD$# new_name = NEW."full_name";
GIBDD$# new_adress = NEW."adress";
GIBDD$# new_phone = NEW."phone";
GIBDD$# new_passport = NEW."passport";
GIBDD$# INSERT INTO citizen_log(action_time, action_type, name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)
GIBDD$# VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);
GIBDD$# RETURN NEW;
GIBDD$# ELSIF TG_OP='DELETE' THEN
GIBDD$# old_name = OLD."full_name";
GIBDD$# old_adress = OLD."adress";
GIBDD$# old_phone = OLD."phone";
GIBDD$# old_passport = OLD."passport";
GIBDD$# INSERT INTO citizen_log(action_time, action_type, name_old, name_new, adress_old, adress_new, phone_old, phone_new, passport_old, passport_new)
GIBDD$# VALUES(now(), TG_OP, old_name, new_name, old_adress, new_adress, old_phone, new_phone, old_passport, new_passport);
GIBDD$# RETURN OLD;
GIBDD$# END IF;
GIBDD$# END;
GIBDD$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

```
GIBDD=# CREATE TRIGGER citizen_log_tg AFTER INSERT OR UPDATE OR DELETE ON
GIBDD-# "GIBDD"."CITIZEN" FOR EACH ROW EXECUTE PROCEDURE
GIBDD-# add_in_citizen_log();
CREATE TRIGGER
```

Выводы по проделанной работе

В ходе проделанной лабораторной работы мы овладели практическими навыками создания, использования процедур, функций и триггеров в консольном клиенте SQL SHELL. Было создано 3 функции и 1 триггер согласно варианту, проверена их работоспособность. Полученные навыки пригодятся нам в дальнейшей работе и реальных проектах.