

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: Залетов А.Д.

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы	3
Практическое задание	3
Вариант 18. БД «ГИБДД»	4
Вывод.....	8

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 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.

Вариант 18. БД «ГИБДД»

Создание функций

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

The screenshot shows the pgAdmin 4 interface. The left pane displays the 'registered_car' table under the 'public' schema. The right pane shows a terminal window with the following SQL commands and results:

```
D:\PostgreSQL\pgAdmin 4\runtime>"D:\PostgreSQL\bin\psql.exe" "host=localhost port=5432 db=postgres user=postgres"
psql (16.0)
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной
страницы Windows (1251).
8-битовые (русские) символы могут отображаться некорректно.
Подробнее об этом смотрите документацию psql, раздел
"Notes for Windows users".
Введите "help", чтобы получить справку.

GIBDD=# \! chcp 1251
Текущая кодовая страница: 1251
GIBDD=# SELECT * FROM get_inspector_violations();
 inspector_name_surname | pts_number | violation_count | total_penalty
-----
Деревсков Денис Климентьевич | 02KP362311 | 2 | 800
(1 строка)

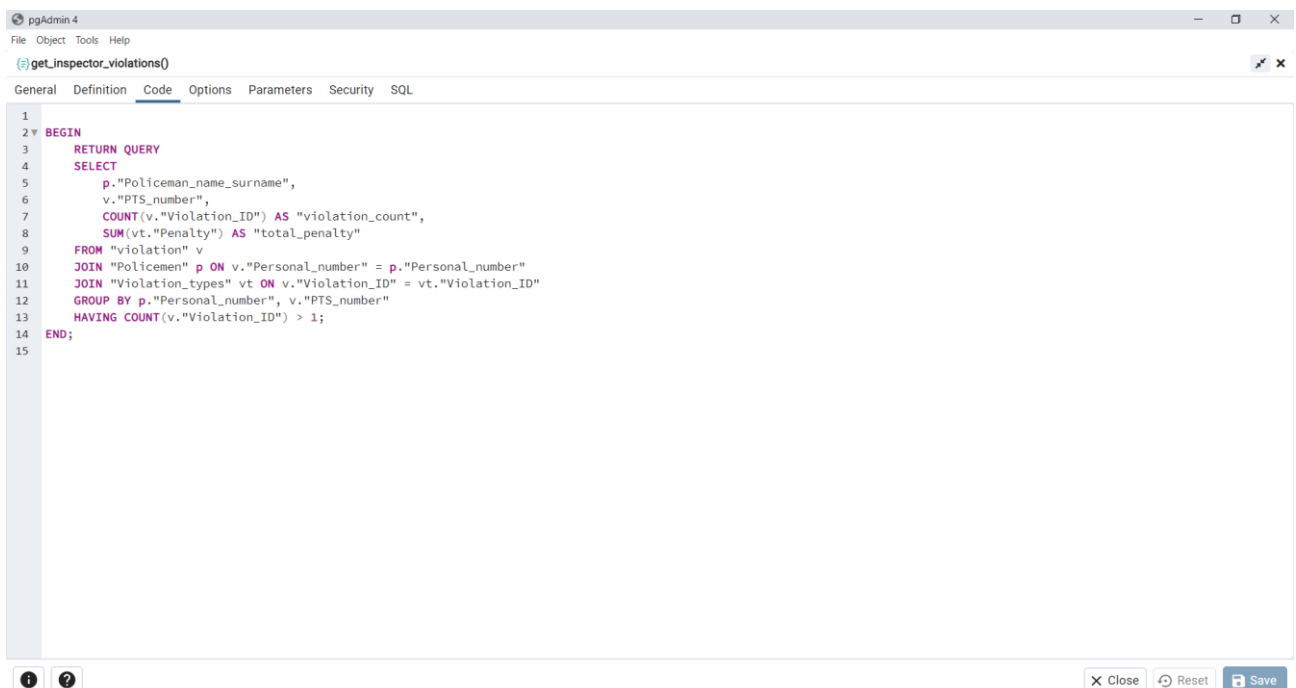
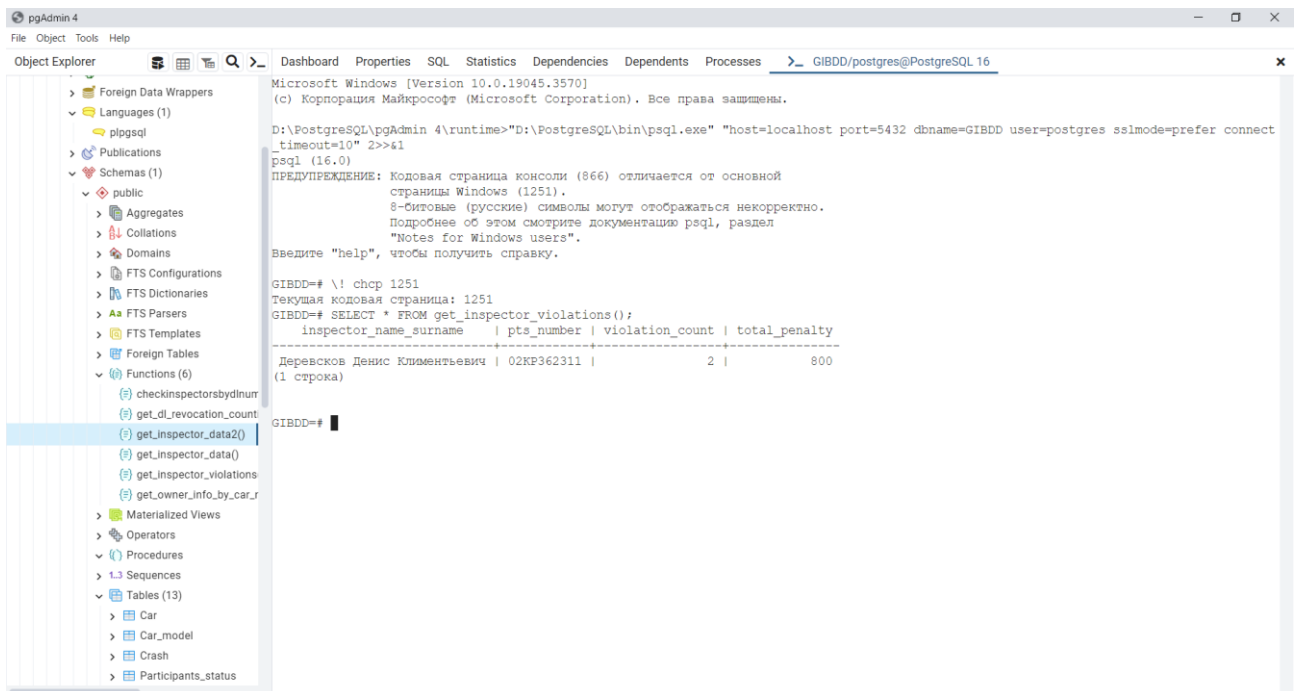
GIBDD=# SELECT * FROM get_owner_info_by_car_number('E100BK37');
Driver_name_surname | Address | Telephone_number | Date_of_birth | Passport
-----
Иванов Иван Иванович | Серебряный Сульвар 12 | +79650897834 | 2001-12-03 | 4018 134534
(1 строка)

GIBDD=#
```

The screenshot shows the pgAdmin 4 interface with the SQL code for the `get_owner_info_by_car_number` function. The code is as follows:

```
1
2 BEGIN
3 RETURN QUERY
4 SELECT
5     co."Driver_name_surname",
6     co."Address",
7     co."Telephone_number",
8     co."Date_of_birth",
9     co."Passport"
10 FROM
11     "car_owner" co
12 JOIN
13     "registered_car" rc ON co."DL_number" = rc."DL_number"
14 WHERE
15     rc."Car_number" = carNumber;
16 END
```

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



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

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query Tool' window. The query is as follows:

```
GIBDD=# \! chcp 1251
Текущая кодовая страница: 1251
GIBDD=# SELECT * FROM get_inspector_violations();
      inspector_name_surname | pts_number | violation_count | total_penalty
-----
Деревсков Денис Климентьевич | 02KP362311 | 2 | 800
(1 строка)
```

Below this, another query is shown:

```
GIBDD=# SELECT * FROM get_owner_info_by_car_number('E100BK37');
Driver_name_surname | Address | Telephone_number | Date_of_birth | Passport
-----
Иванов Иван Иванович | Серебряный бульвар 12 | +79650897834 | 2001-12-03 | 4018 134534
(1 строка)
```

And a third query:

```
GIBDD=# SELECT * FROM get_dl_revocation_countin_district('Московский');
get_dl_revocation_countin_district
-----
5
(1 строка)
```

The 'Object Explorer' on the left shows the database structure, including tables like 'Car', 'Car_model', 'Crash', 'Participants_status', 'Police_department', 'Policemen', 'Rights_deprivation', 'Columns', 'Constraints', 'Indexes', 'RLS_Policies', 'Rules', 'Triggers', 'Violation_types', and 'car_owner'.

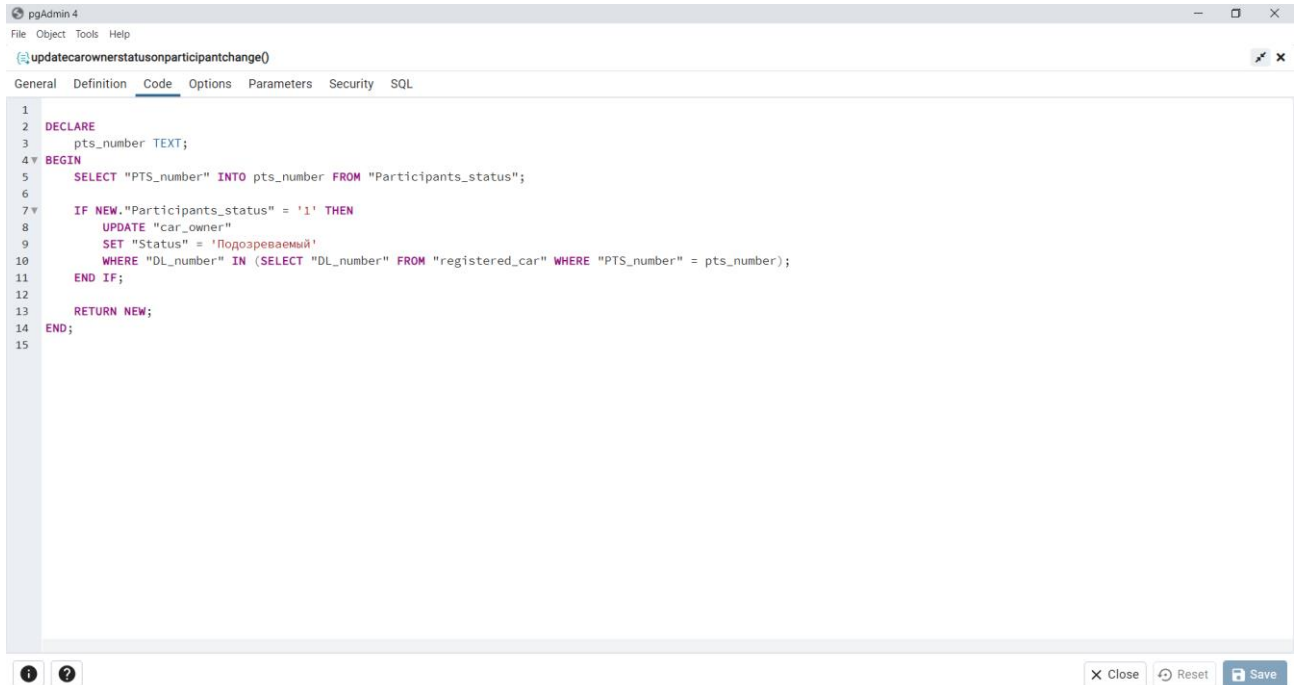
The screenshot shows the pgAdmin 4 interface with a SQL query in the 'Query Tool' window. The query is as follows:

```
1
2 DECLARE
3     revocationCount INTEGER;
4 BEGIN
5     SELECT COUNT(DISTINCT co."DL_number")
6     INTO revocationCount
7     FROM "Violation" v
8     JOIN "Violation_types" vt ON v."Violation_ID" = vt."Violation_ID"
9     JOIN "car_owner" co ON co."DL_number" = co."DL_number"
10    JOIN "registered_car" rc ON v."PTS_number" = rc."PTS_number"
11    WHERE vt."DL_loss_time" > 0 AND v."District" = districtName;
12
13 RETURN revocationCount;
14 END
```

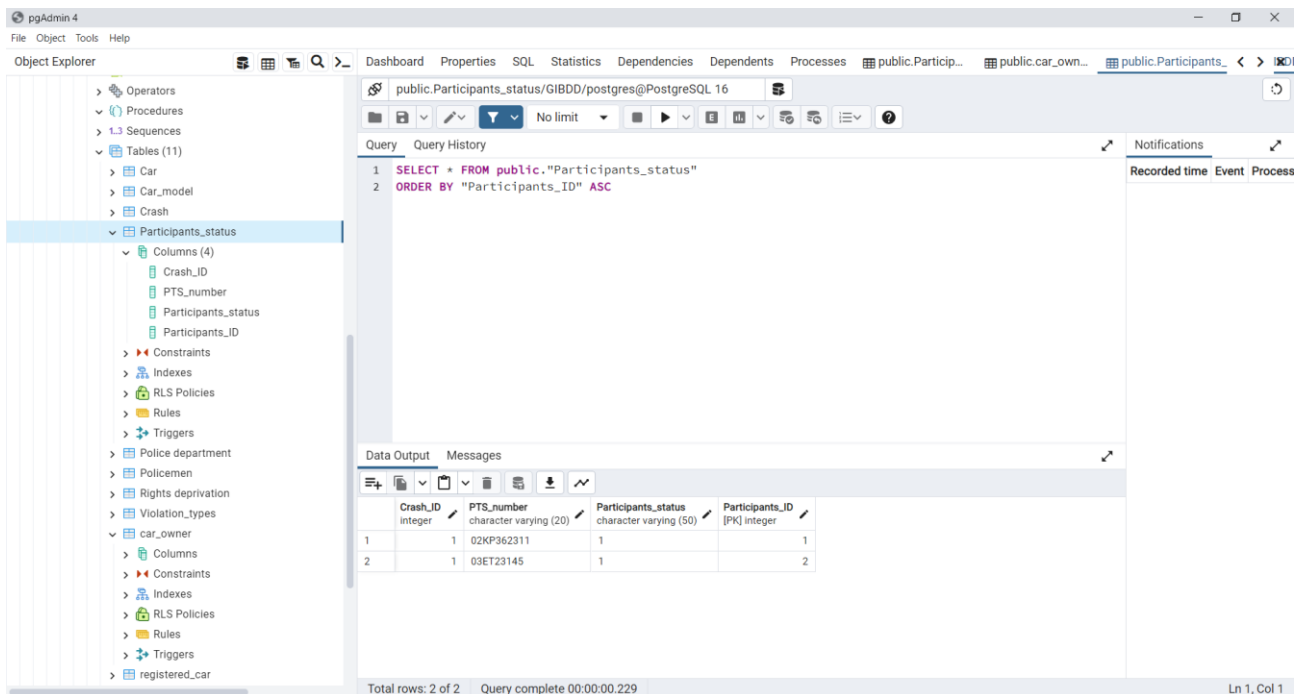
The 'Object Explorer' on the left shows the database structure, including tables like 'Car', 'Car_model', 'Crash', 'Participants_status', 'Police_department', 'Policemen', 'Rights_deprivation', 'Columns', 'Constraints', 'Indexes', 'RLS_Policies', 'Rules', 'Triggers', 'Violation_types', and 'car_owner'.

2)Создание триггера

Создадим триггер который бы при изменении таблицы Participant_status, где устанавливается 1 в поле о том, что водитель виновен в аварии, устанавливал соответствующему водителю статус “Подозреваемый”



```
1
2 DECLARE
3     pts_number TEXT;
4 BEGIN
5     SELECT "PTS_number" INTO pts_number FROM "Participants_status";
6
7 IF NEW."Participants_status" = '1' THEN
8     UPDATE "car_owner"
9     SET "Status" = 'Подозреваемый'
10    WHERE "DL_number" IN (SELECT "DL_number" FROM "registered_car" WHERE "PTS_number" = pts_number);
11 END IF;
12
13 RETURN NEW;
14 END;
```



The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure, with the 'Participants_status' table selected. The main pane shows a query window with the following SQL:

```
1 SELECT * FROM public."Participants_status"
2 ORDER BY "Participants_ID" ASC
```

Below the query window, the Data Output tab displays the results of the query in a table format:

Crash_ID	PTS_number	Participants_status	Participants_ID
1	02KP362311	1	1
2	03ET23145	1	2

At the bottom of the interface, a status bar indicates: Total rows: 2 of 2, Query complete 00:00:00.229, Ln 1, Col 1.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer with a tree view of database objects. The 'car_owner' table is selected under the 'Tables' category. The main pane displays a SQL query:

```
1 SELECT * FROM public.car_owner
2 ORDER BY "DL_number" ASC
```

Below the query, the 'Data Output' tab shows the results of the query in a table format. The table has 6 columns: name_surname, Address, Telephone_number, Date_of_birth, Passport, and Status. There are 5 rows of data.

	name_surname	Address	Telephone_number	Date_of_birth	Passport	Status
1	Илов Иван Иванович	Проспект ветеранов к153	+79656578501	2001-03-21	4618 147216	Подозреваемый
2	ев Иван Иванович	Биржевая Улица 14	+79657005804	1996-05-03	4745 501529	[null]
3	ев Иван Иванович	Серебряный бульвар 12	+79650897834	2001-12-03	4018 134534	Подозреваемый
4	ев Сергей Сергеевич	Улица Марата 13	+79652692357	2000-11-04	4224 941017	[null]
5	ов Артём Дмитриевич	Улица Ленина 15	+79652329401	1998-02-02	4680 471692	[null]

At the bottom, it says 'Total rows: 5 of 5' and 'Query complete 00:00:00.225'.

Вывод

В ходе лабораторной работы были освоены практические навыки по созданию, функций в PostgreSQL с использованием инструмента управления pgAdmin 4 и PSQL tool. Были созданы функции на выборку данных, а также был создан триггер, помогающий нам выявлять подозреваемого, если он был виновен в аварии.