Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С.Галущака»

Практическая работа №6

Тема: Изучение PostgreSQL

Учебная дисциплина: МДК.11.01 Технология разработки и защиты баз данных

Работу выполнил:

студент группы ПР-21.102:

Воробьев П.А

Проверил: Еремеева Л.А

2024

# Вариант 8. БД пункта проката автомобилей.

Л. р. №6. Создание функций.

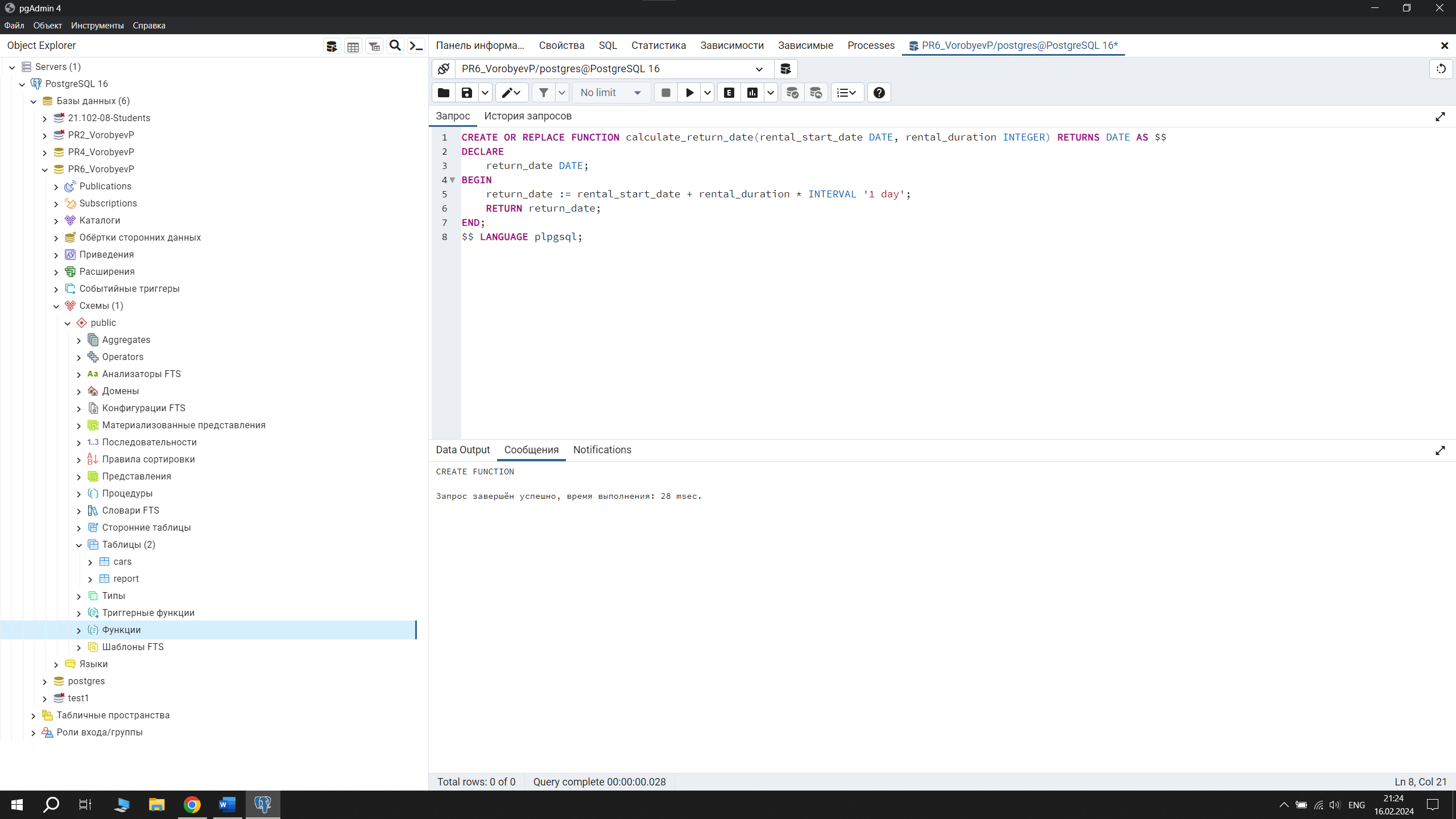
1. Функция, принимающая два параметра – дату и продолжительность аренды. Функция должна возвращать предположительную дату возврата автомобиля.
2. Функция, добавляющая в специальную таблицу «Отчет» сведения о каждом автомобиле определенной марки на текущую дату. Параметр – марка автомобиля. Функция должна перед началом работы очищать таблицу. Для автомобилей, находящихся в прокате, функция должна вносить номер, год выпуска, цвет, пробег, ФИО арендатора, начало и предполагаемую дату возврата автомобиля. Для автомобилей, не находящихся в прокате, функция должна вносить номер, год выпуска, цвет, пробег, дату последнего возврата из аренды и продолжительность аренды за последний год (в днях). Использовать ранее созданную функцию.

Л. р. №7. Создание триггеров.

1. Триггер, устанавливающий стоимость проката после установки даты возврата автомобиля как произведение стоимости аренды на ее реальную продолжительность.
2. Проверка значений всех полей отношения "Договоры на аренду", для которых могут быть определены домены: дата начала договора не меньше текущей, дата возврата автомобиля не определена, стоимость договора равна продолжительности, умноженной на стоимость аренды; продолжительность больше 0.

**Функции**

1. Функция, принимающая два параметра – дату и продолжительность аренды.



CREATE OR REPLACE FUNCTION calculate\_return\_date(rental\_start\_date DATE, rental\_duration INTEGER) RETURNS DATE AS $$

DECLARE

return\_date DATE;

BEGIN

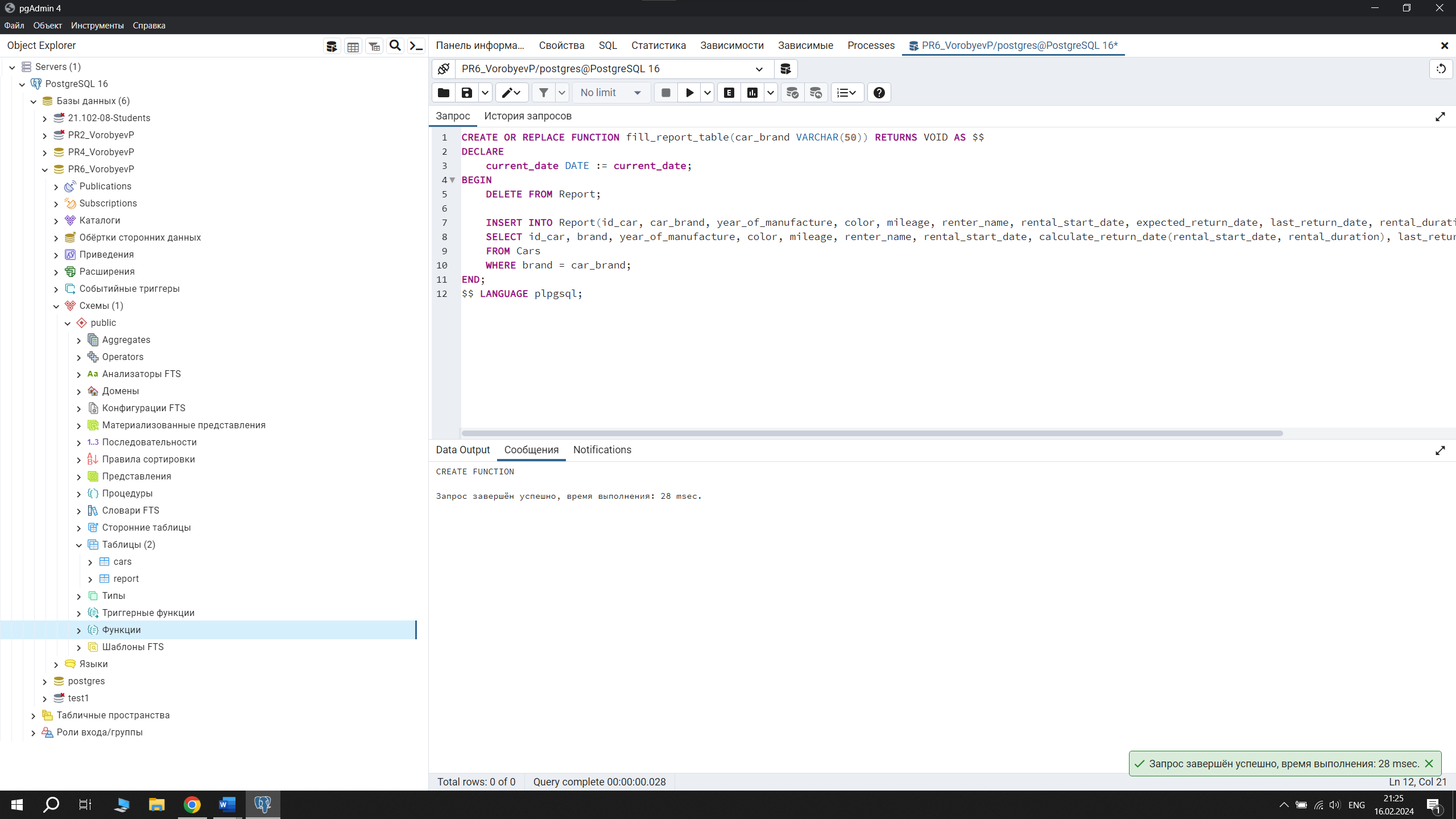
return\_date := rental\_start\_date + rental\_duration \* INTERVAL '1 day';

RETURN return\_date;

END;

$$ LANGUAGE plpgsql;

2. Функция, добавляющая в специальную таблицу «Отчет» сведения о каждом автомобиле определенной марки на текущую дату.



CREATE OR REPLACE FUNCTION fill\_report\_table(car\_brand VARCHAR(50)) RETURNS VOID AS $$

DECLARE

current\_date DATE := current\_date;

BEGIN

DELETE FROM Report;

INSERT INTO Report(id\_car, car\_brand, year\_of\_manufacture, color, mileage, renter\_name, rental\_start\_date, expected\_return\_date, last\_return\_date, rental\_duration\_last\_year)

SELECT id\_car, brand, year\_of\_manufacture, color, mileage, renter\_name, rental\_start\_date, calculate\_return\_date(rental\_start\_date, rental\_duration), last\_return\_date, rental\_duration\_last\_year

FROM Cars

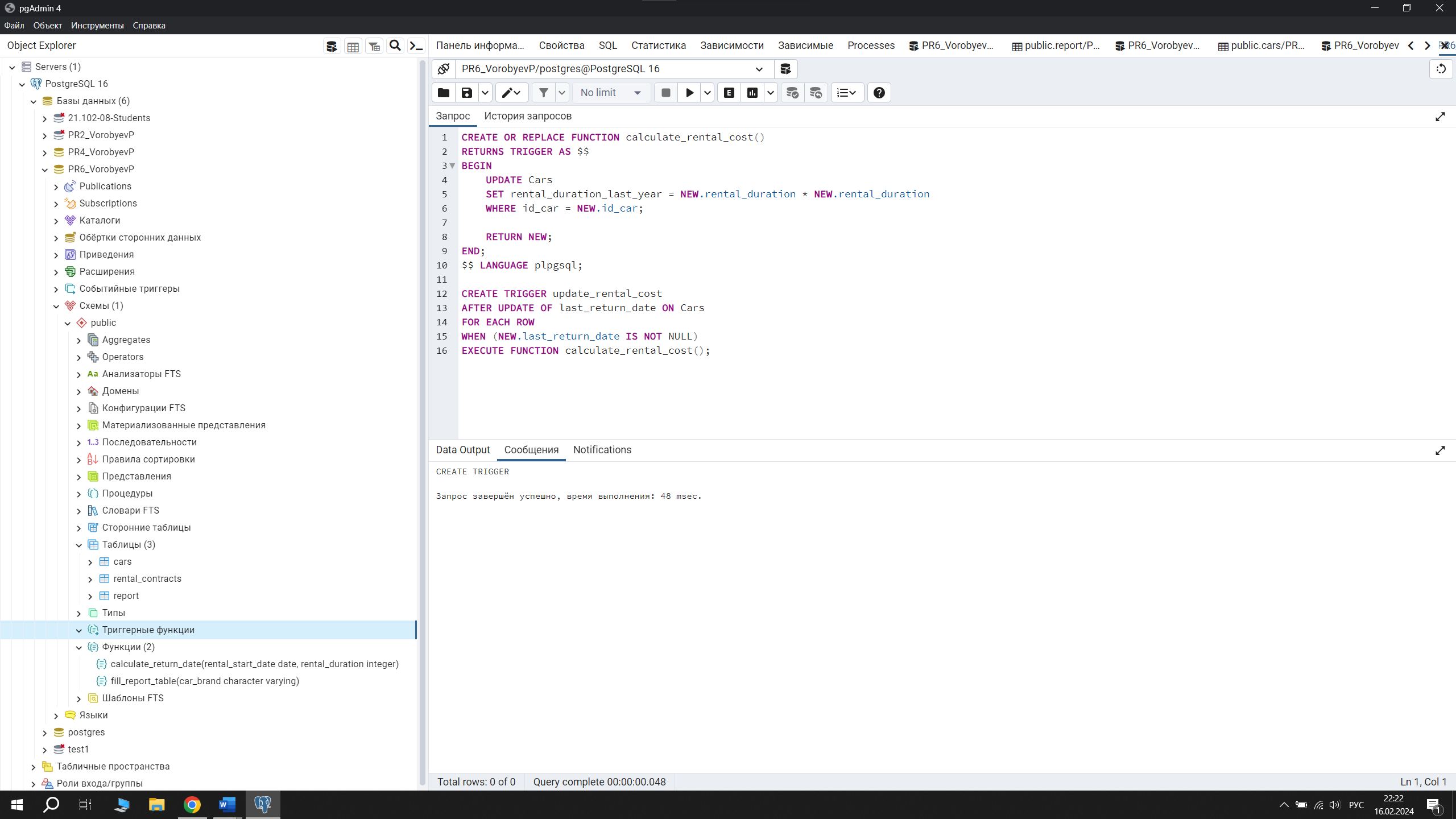
WHERE brand = car\_brand;

END;

$$ LANGUAGE plpgsql;

**Триггеры**

1.



CREATE OR REPLACE FUNCTION calculate\_rental\_cost()

RETURNS TRIGGER AS $$

BEGIN

UPDATE Cars

SET rental\_duration\_last\_year = NEW.rental\_duration \* NEW.rental\_duration

WHERE id\_car = NEW.id\_car;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER update\_rental\_cost

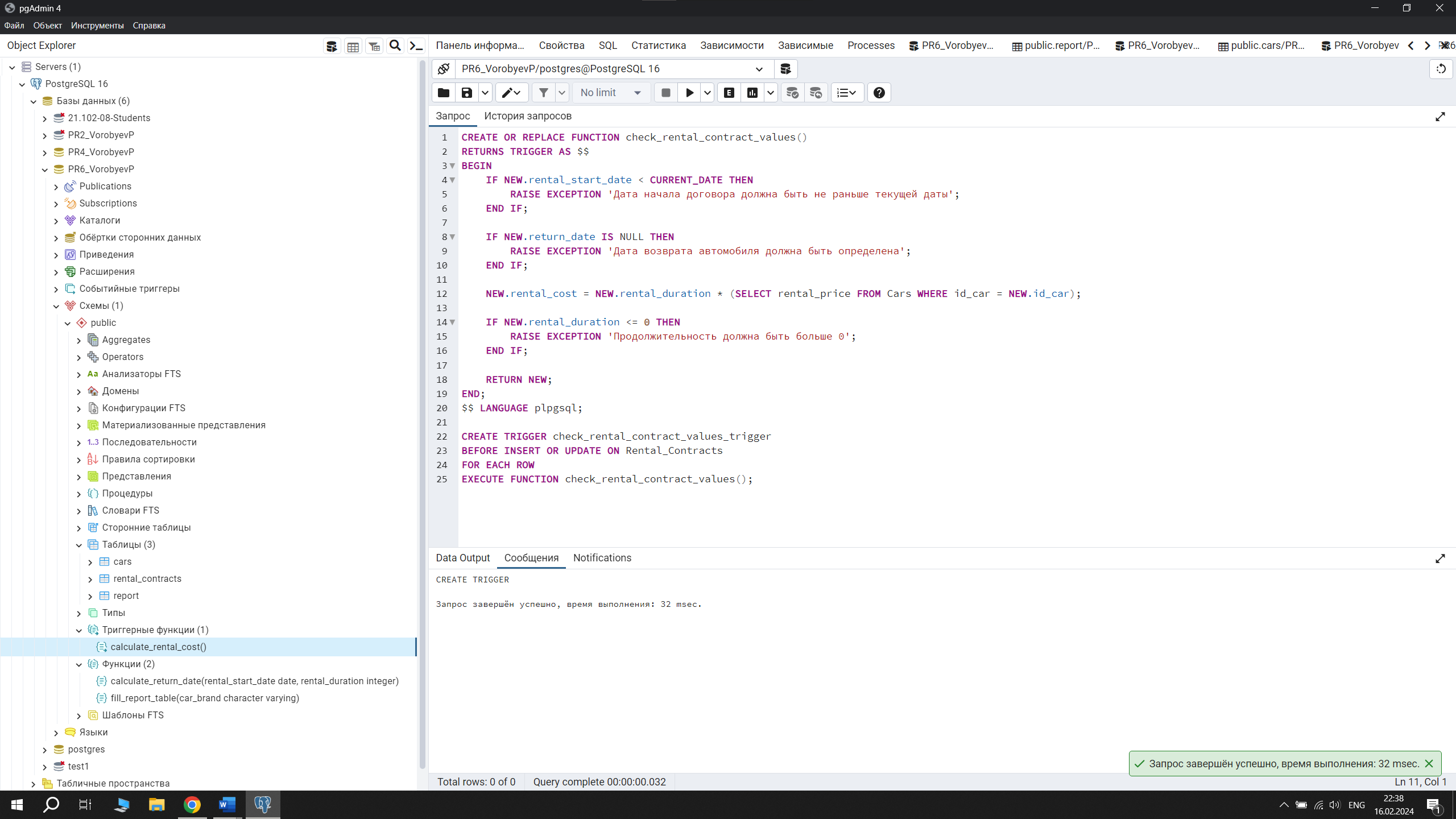
AFTER UPDATE OF last\_return\_date ON Cars

FOR EACH ROW

WHEN (NEW.last\_return\_date IS NOT NULL)

EXECUTE FUNCTION calculate\_rental\_cost();

2.



CREATE OR REPLACE FUNCTION check\_rental\_contract\_values()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.rental\_start\_date < CURRENT\_DATE THEN

RAISE EXCEPTION 'Дата начала договора должна быть не раньше текущей даты';

END IF;

IF NEW.return\_date IS NULL THEN

RAISE EXCEPTION 'Дата возврата автомобиля должна быть определена';

END IF;

NEW.rental\_cost = NEW.rental\_duration \* (SELECT rental\_price FROM Cars WHERE id\_car = NEW.id\_car);

IF NEW.rental\_duration <= 0 THEN

RAISE EXCEPTION 'Продолжительность должна быть больше 0';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER check\_rental\_contract\_values\_trigger

BEFORE INSERT OR UPDATE ON Rental\_Contracts

FOR EACH ROW

EXECUTE FUNCTION check\_rental\_contract\_values();