

***Федеральное агентство по рыболовству***

***Федеральное государственное бюджетное образовательное***

***учреждение высшего образования***

***«Астраханский государственный технический университет»***

**Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована**

**ООО «ДКС РУС» по международному стандарту ISO 9001:2015**

|  |  |
| --- | --- |
| Институт | Информационных технологий и коммуникаций |
| Направление | 09.03.01 Информатика и вычислительная техника |
| Профиль | «Автоматизированные системы обработки информации и управления» |
| Кафедра | «Автоматизированные системы обработки информации и управления» |

**Лабораторная работа № 6**

**«СУБД PostgreSQL. Работа с массивами (часть 1) и JSON-объектами**»

по дисциплине «СУБД PostgreSQL»

|  |  |  |
| --- | --- | --- |
|  |  | Работа выполнена студентом группы ДИНРБ-31  Кузургалиев Р.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Фамилия И.О.) подпись |
|  |  | Проверил работу:  ст. преподаватель Мамлеева А.Р,\_\_\_\_\_\_\_\_\_\_\_\_  (ученая степень, ученое звание, Фамилия И.О.) |

Работа защищена

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**АСТРАХАНЬ – 2024**

**СОДЕРЖАНИЕ ОТЧЕТА**

**Цель работы**: получить практические навыки в использовании массивов и JSON-объектов.

**Литература**: 1. Лекция № 2. Разделы «Массивы» и «JSON-объекты».

2. Моргунов Е.П. PostgreSQL. Основы языка SQL: учеб. пособие. – СПб.:

БХВ-Петербург, 2018.- 336 с.: ил.

**Задание 1**. Объединение (конкатенация) массивов.

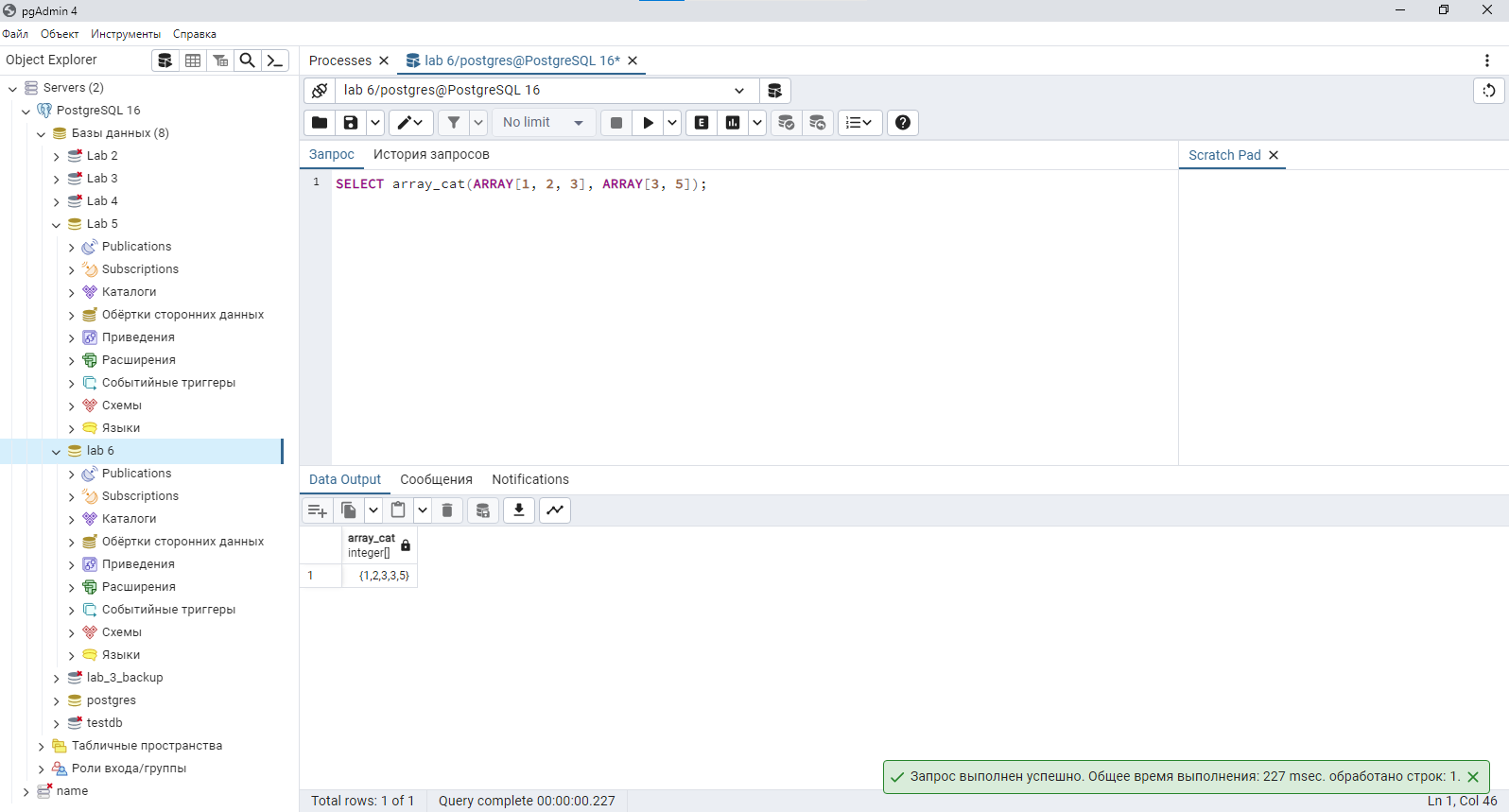
Для объединения (конкатенации) массивов служит функция array\_cat:

SELECT array\_cat(ARRAY[1, 2, 3], ARRAY[3, 5];

array\_cat

-----------------

{1, 2, 3, 3, 5}



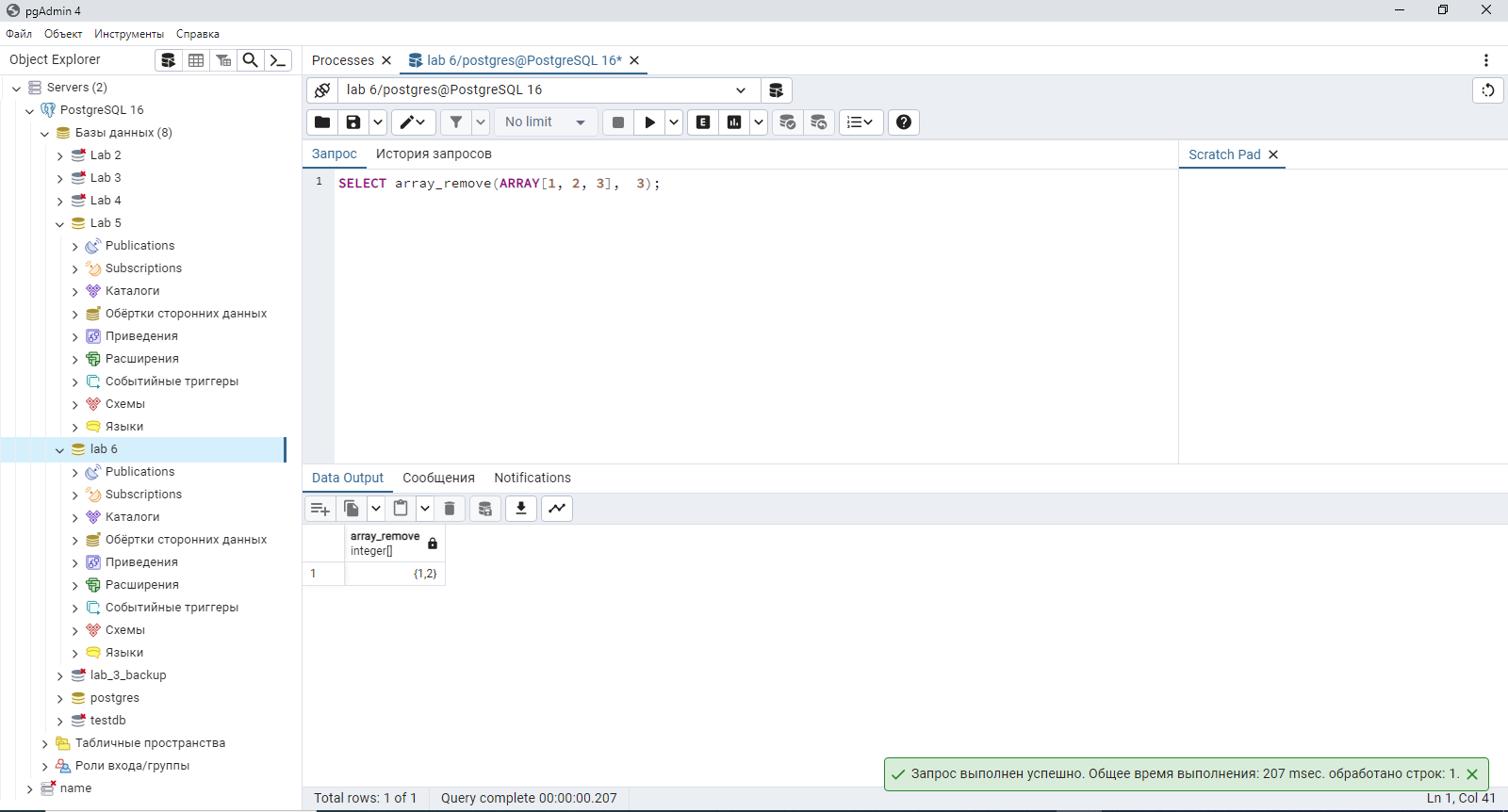
**Задание 2**. Удалить из массива элементы, имеющие указанное значение:

SELECT array\_remove(ARRAY[1, 2, 3], 3);

array\_remove

-----------------

{1, 2}



**Задание 3**. Массивы могут быть многомерными и в них могут содержаться значения любых типов.

Рассмотрим сначала одномерные массивы *текстовых* значений.

Предположим, что пилоты авиакомпании имеют возможность высказывать свои пожелания насчет конкретных блюд, из которых должен состоять их обед во время полета.

Для учета пожеланий пилотов необходимо модифицировать таблицу pilots, с которой Вы работали ранее (но если Вы хотите ее оставить, то создайте новую таблицу, например, pilots2).

DROP TABLE pilots;

CREATE TABLE pilots

( pilots\_name text,

schedule integer[ ],

meal text[ ]

);

Добавьте строки в таблицу:

INSERT INTO pilots VALUES ('Ivan', '{1, 3, 5, 6, 7}'::integer[],

'{"сосиска", "макароны", "кофе"}'::text[]

),

('Petr', '{1, 2, 5, 7}'::integer[],

'{"котлета", "каша", "кофе"}'::text[]

),

('Pavel', '{2, 5}'::integer[],

'{"сосиска", "каша", "кофе"}'::text[]

),

('Boris', '{3, 5, 6}'::integer[],

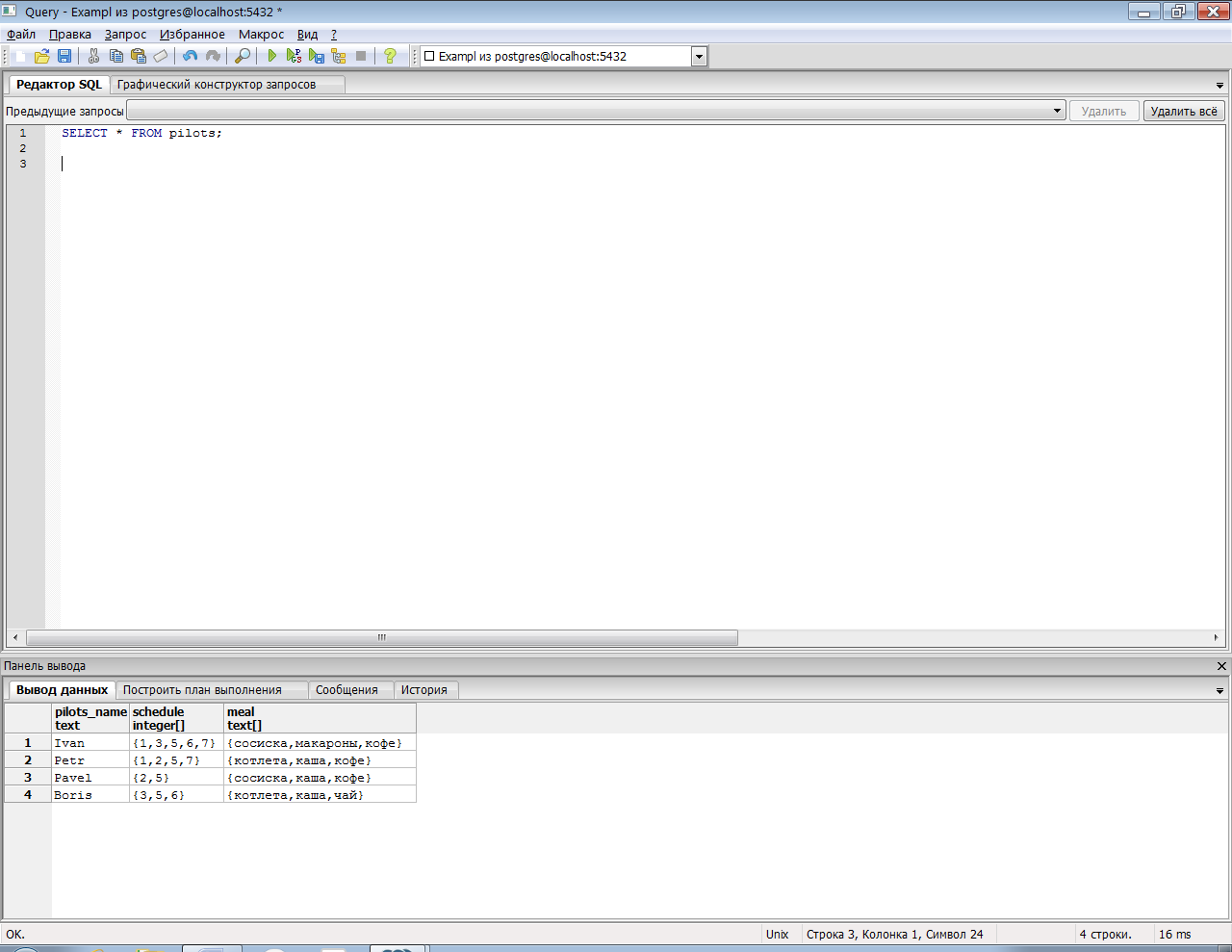
'{"котлета", "каша", "чай"}'::text[]

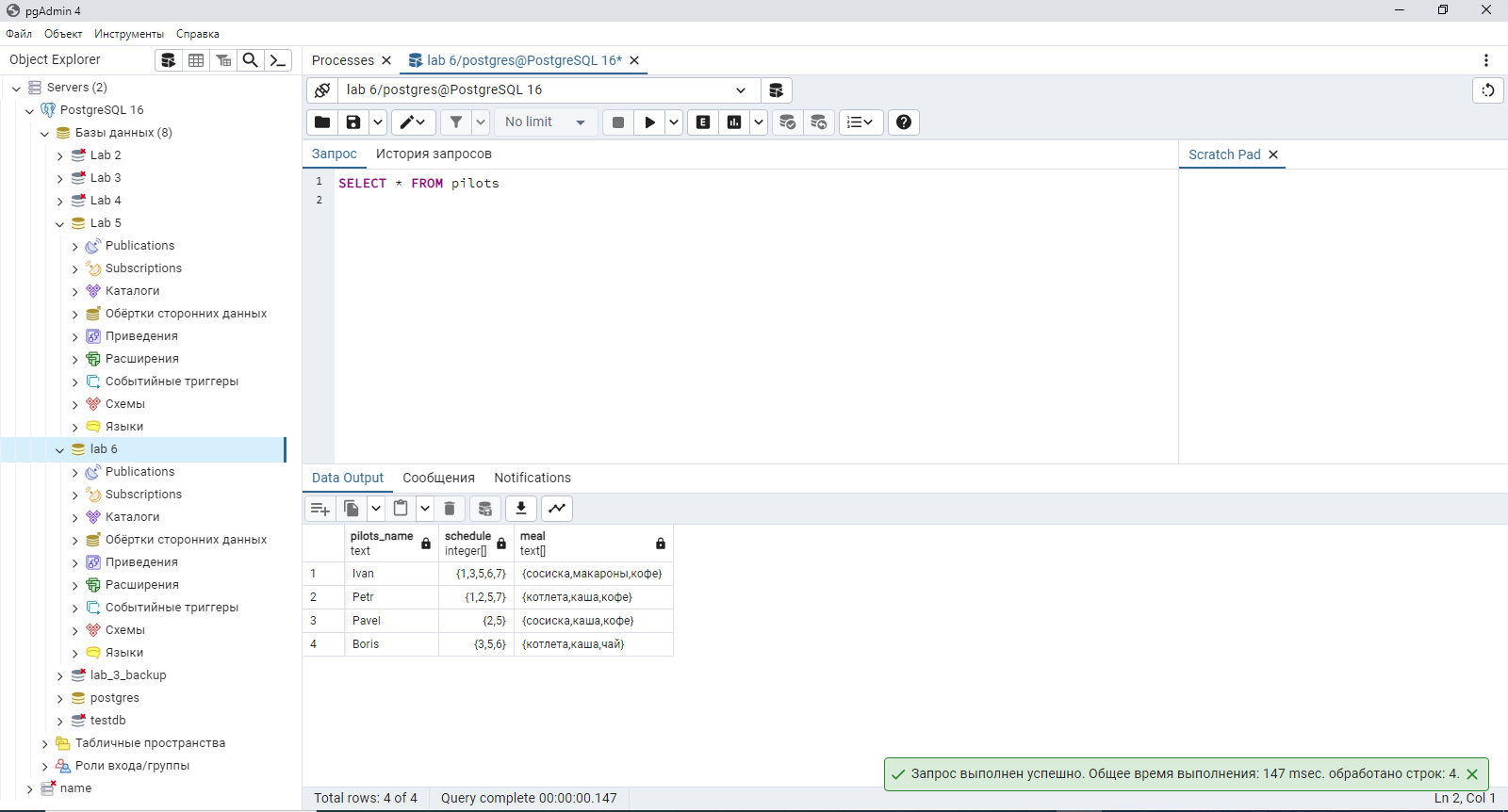
);

*Примечание*: обратите внимание, что каждое из текстовых значений, включаемых в литерал массива, заключается в двойные кавычки, а в качестве типа данных указывается text[ ].

Проверьте, у Вас так получилось, как показано ниже?

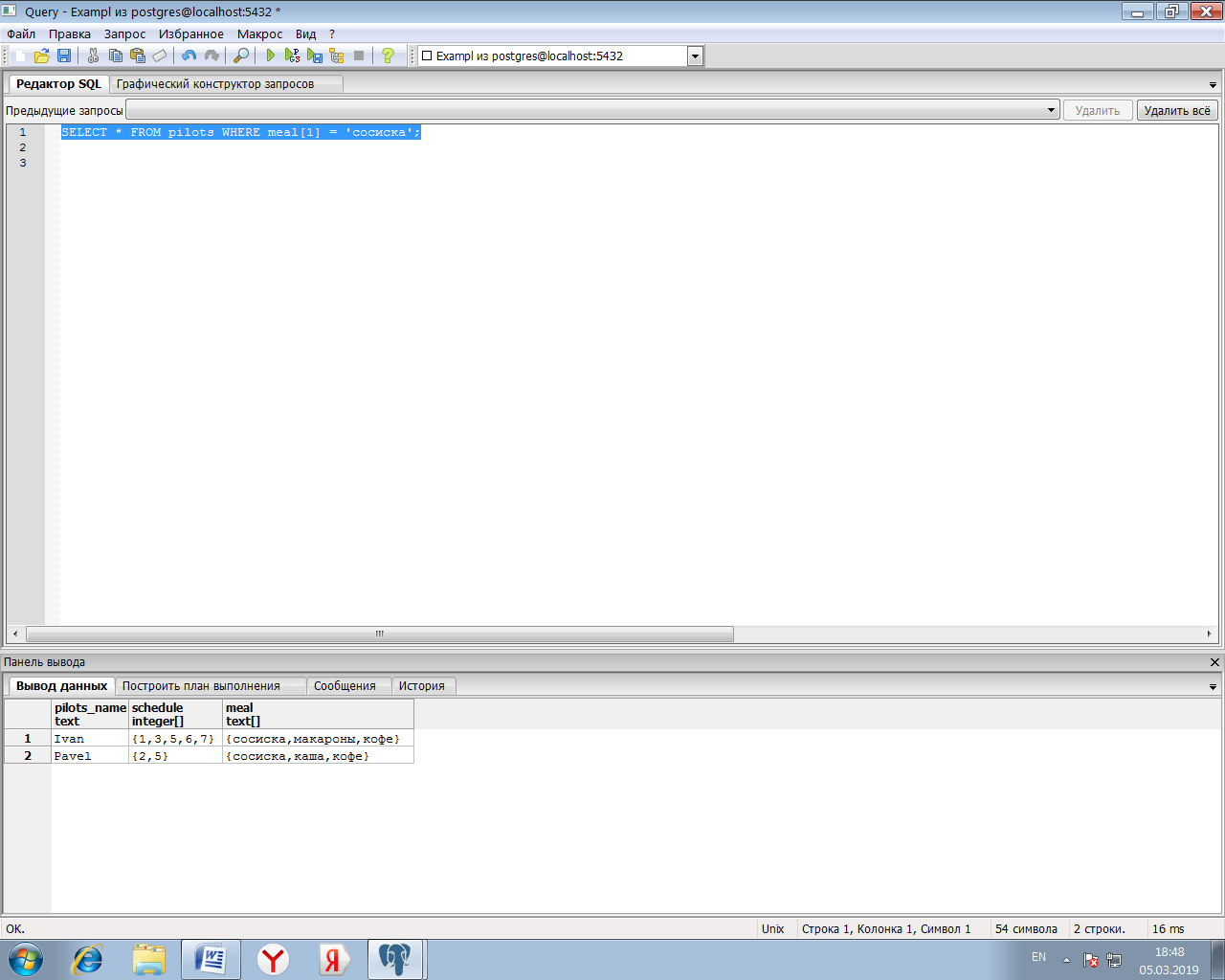
SELECT \* FROM pilots;

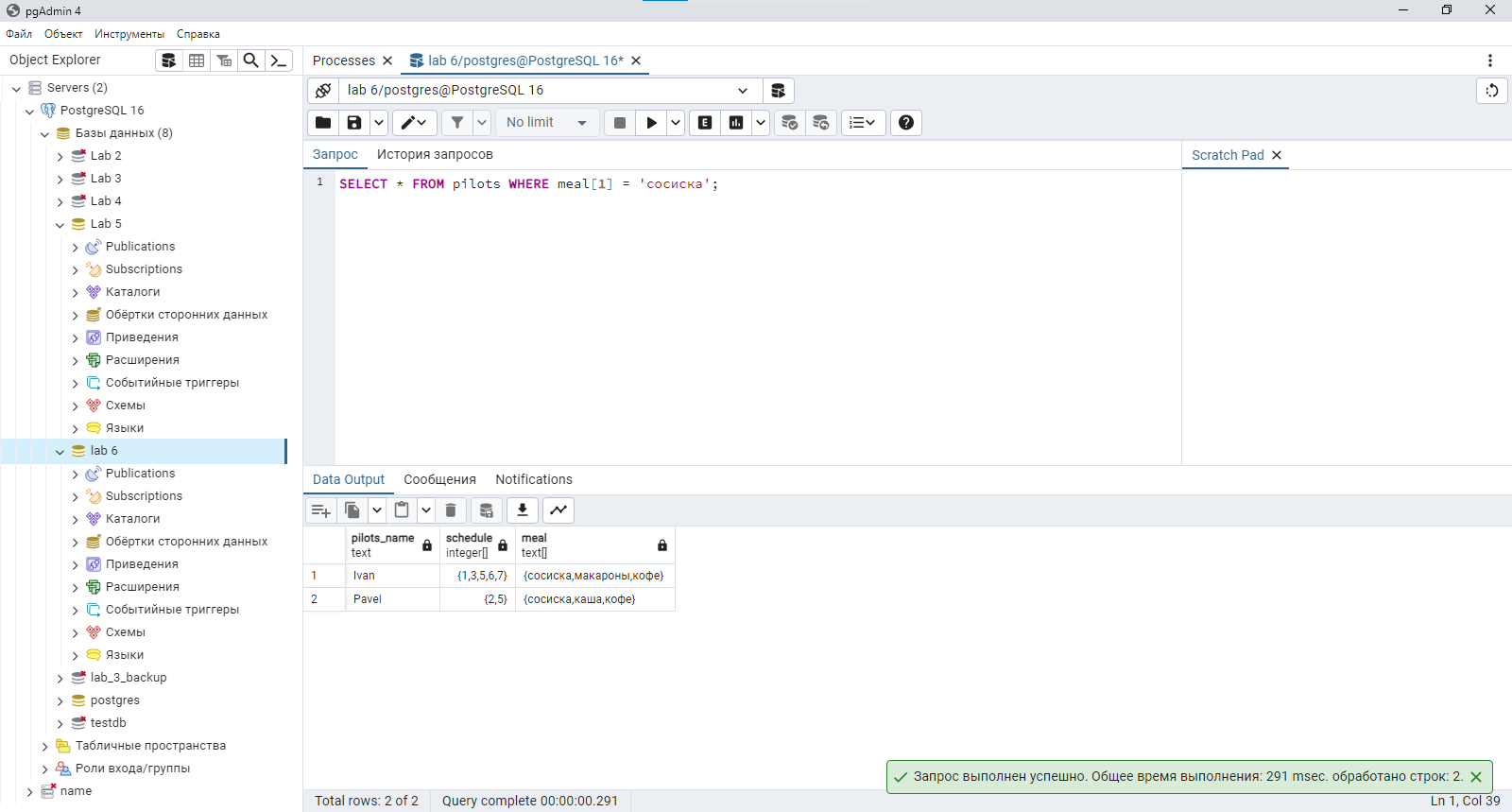




**Задание 4**. Получите список пилотов, предпочитающих на обед сосиски:

SELECT \* FROM pilots WHERE meal[1] = 'сосиска';





**Задание 5**. Предположим, что руководство авиакомпании решило, что пища пилотов должна быть разнообразной. Оно позволило им выбрать свой рацион на каждый из четырех дней недели, в которые пилоты совершают полеты.

Для нас это решение руководства выливается в необходимость модифицировать таблицу, а именно: столбец meal теперь будет содержать двумерные массивы. Определение этого столбца станет таким: meal text [ ] [ ].

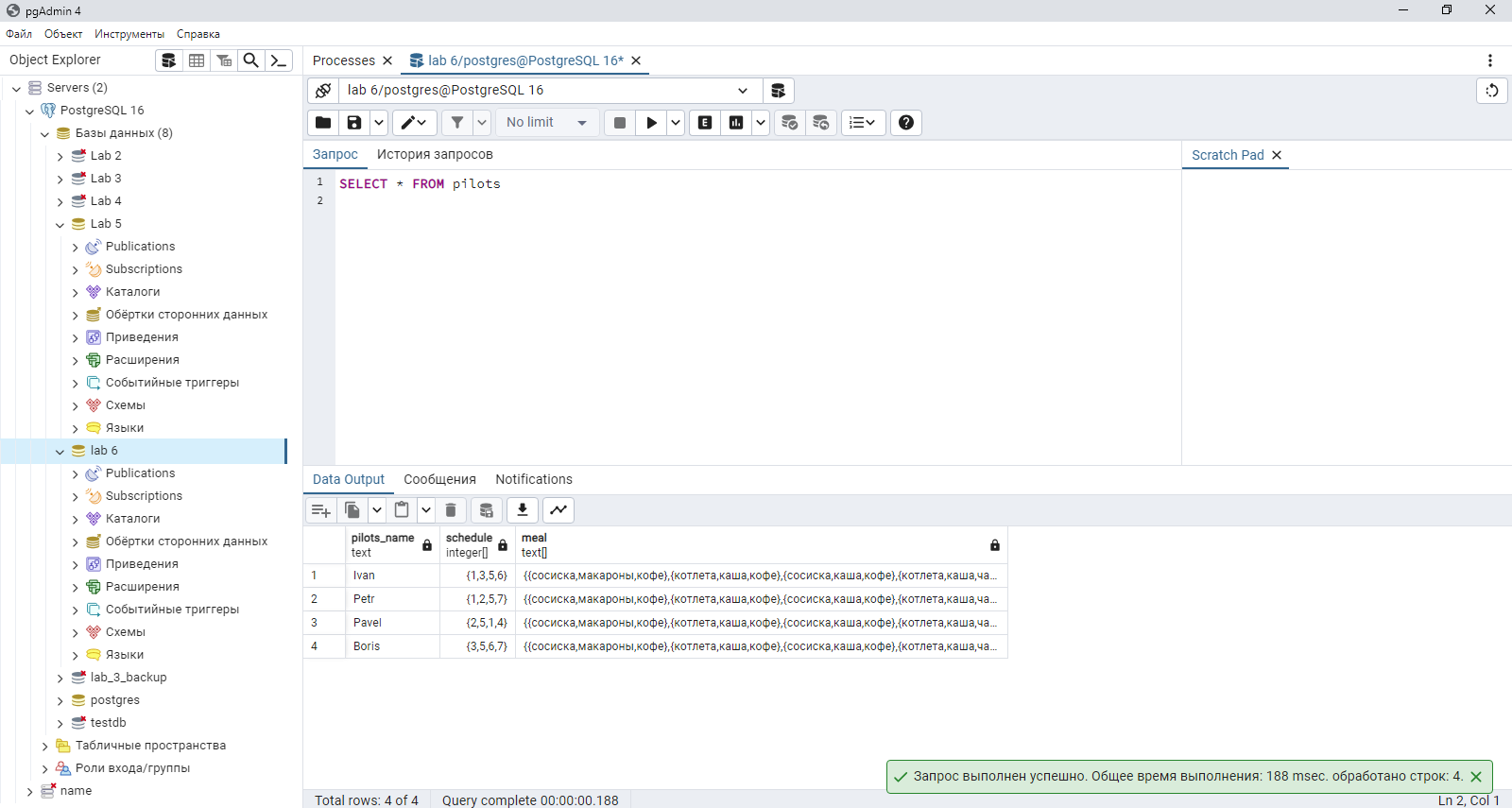
Создайте новую версию таблицы pilots (по четыре дня полетов у каждого пилота) и соответственно измените команду INSERT¸ чтобы в ней содержались литералы *двумерных* массивов. Они будут выглядеть так:

'{ {"сосиска", "макароны", "кофе"},

{"котлета", "каша", "кофе"},

{"сосиска", "каша", "кофе"},

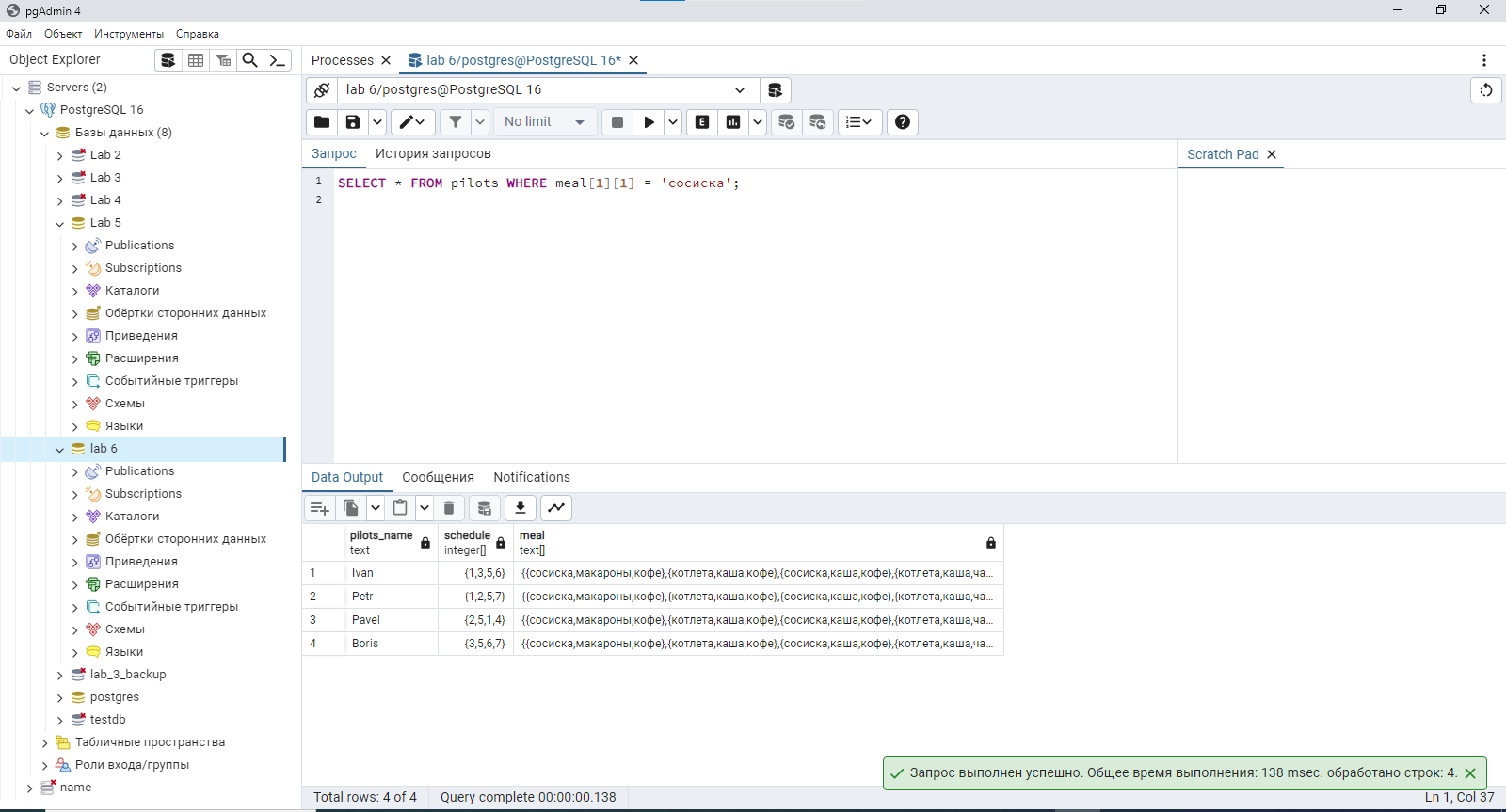
{"котлета", "каша", "чай" } }'::text[ ] [ ]



**Задание 6**. Сделайте ряд выборок и обновлений строк в таблице pilots, созданной в задании 5.

Для обращения к элементам двумерного массива нужно использовать два индекса.

*Не забывайте, что по умолчанию номера индексов начинаются с единицы.*



**Задание 7**. Создайте таблицу pilot\_hobbies по примеру из лекции:

CREATE TABLE pilot\_hobbies

( pilot\_name text,

hobbies jsonb);

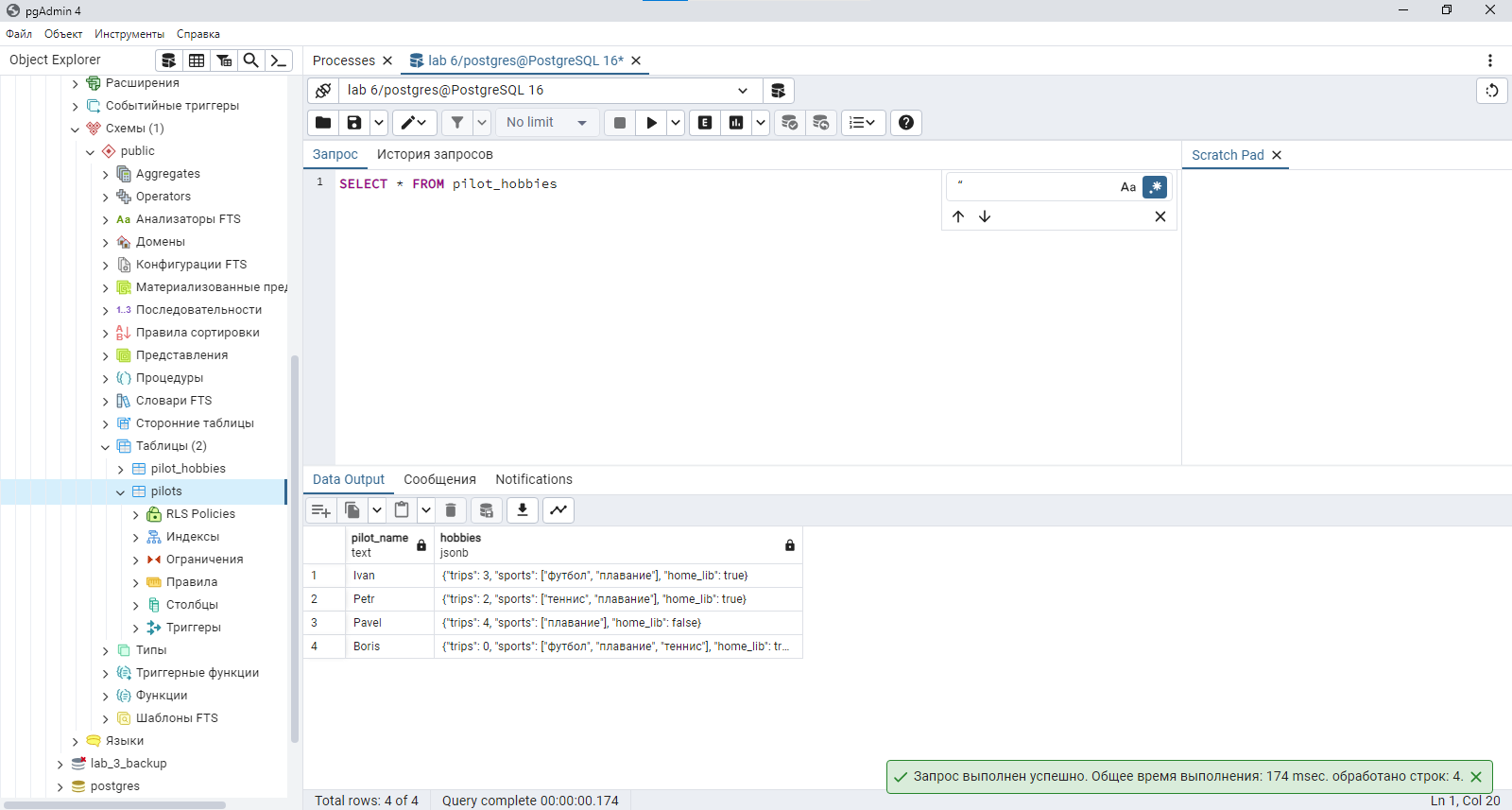
INSERT INTO pilot\_hobbies VALUES

(‘Ivan’, ‘{“sports”: [“футбол”, “плавание”], “home\_lib”: true, “trips”: 3}’ :: jsonb),

(‘Petr’, ‘{“sports”: [“теннис”, “плавание”], “home\_lib”: true, “trips”: 2}’ :: jsonb),

(‘Pavel’, ‘{“sports”: [“плавание”], “home\_lib”: false, “trips”: 4}’ :: jsonb),

(‘Boris’, ‘{“sports”: [“футбол”, “плавание”, “теннис”], “home\_lib”: true, “trips”: 0}’ :: jsonb);



*Примечание: jsonb в версии 9.3 не существует! Вместо jsonb используйте json.*

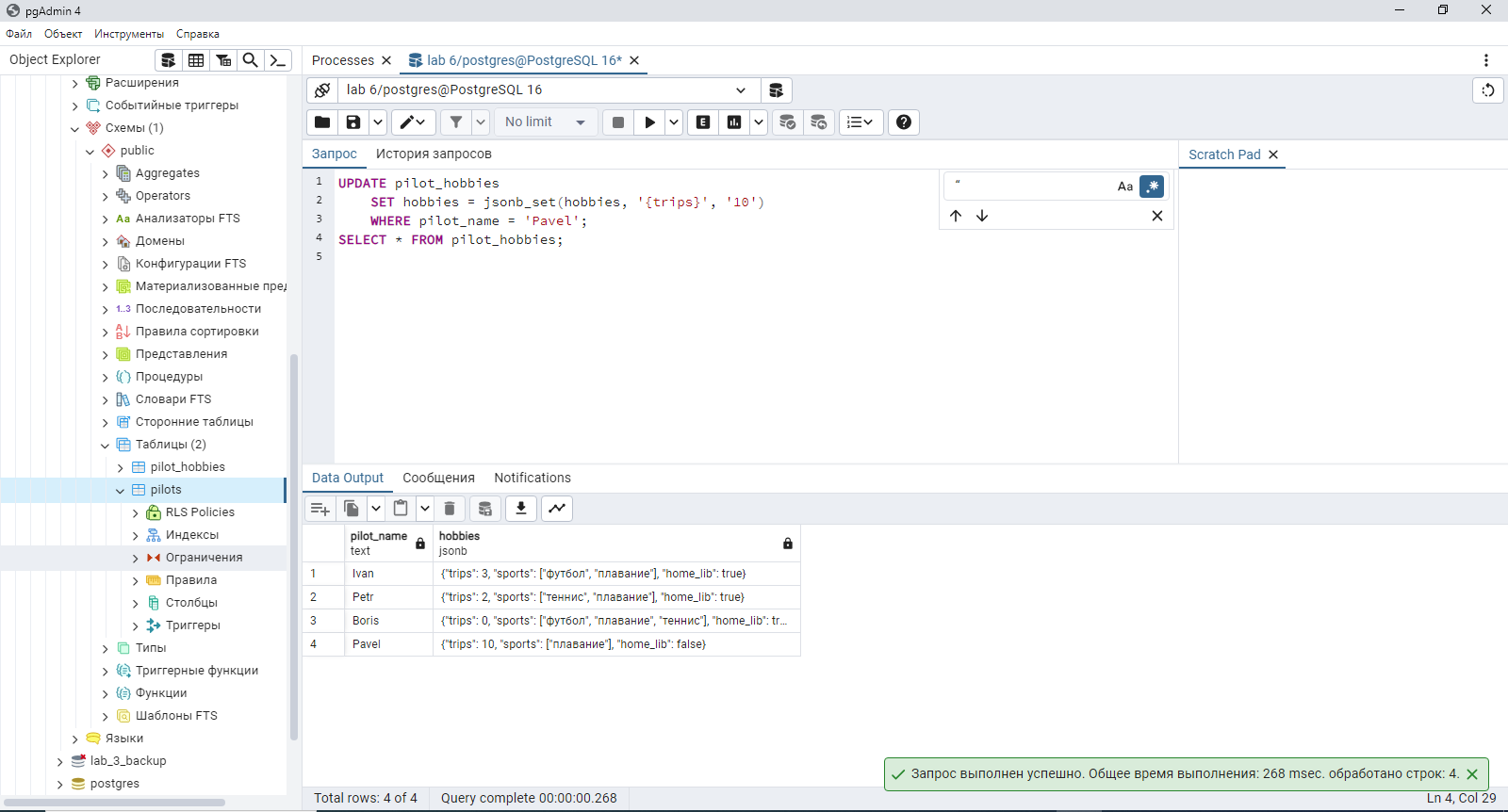
**Задание 8**. Обновить скалярные значения, например по ключу trips:

UPDATE pilot\_hobbies

SET hobbies = jsonb \_set(hobbies, ‘{trips}’, '10’)

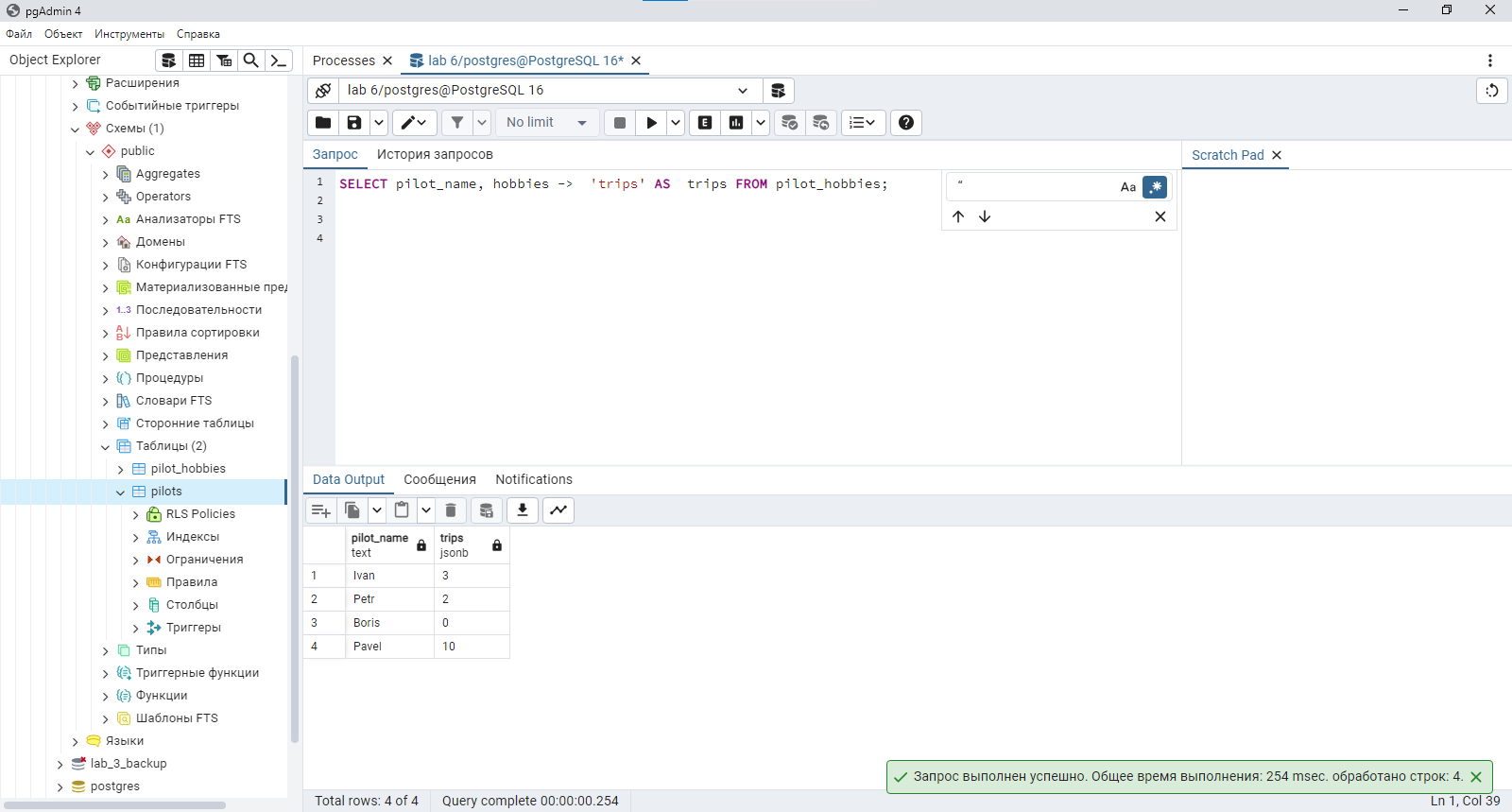
WHERE pilot\_name = ‘Pavel’;

*Пояснения: второй параметр функции – это путь в пределах JSON-объекта. Он теперь представляет собой лишь имя ключа. Однако, его необходимо заключить в фигурные скобки. Третий параметр – этот новое значение. Хотя оно числовое, но все равно требуется записать его в одинарных кавычках.*

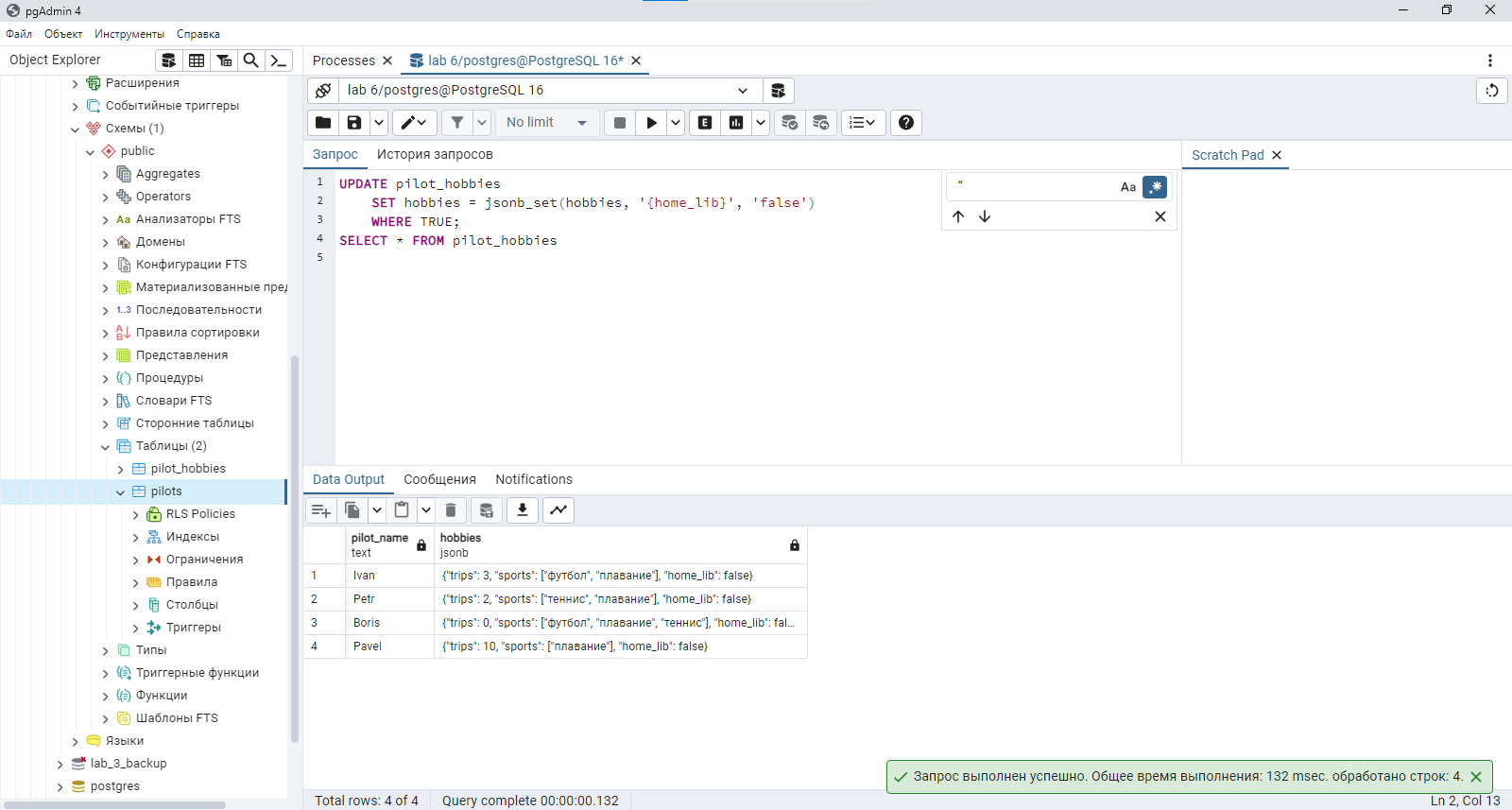
**

Проверьте выполненные изменения командой:

SELECT pilot\_name, hobbies -> ‘trips’ AS trips FROM pilot\_hobbies;



**Задание 9**. Самостоятельно выполните изменение значения по ключу home\_lib в одной из строк.



**Задание 10**. Объекты JSON в разных строках таблицы могут иметь различные наборы ключей.

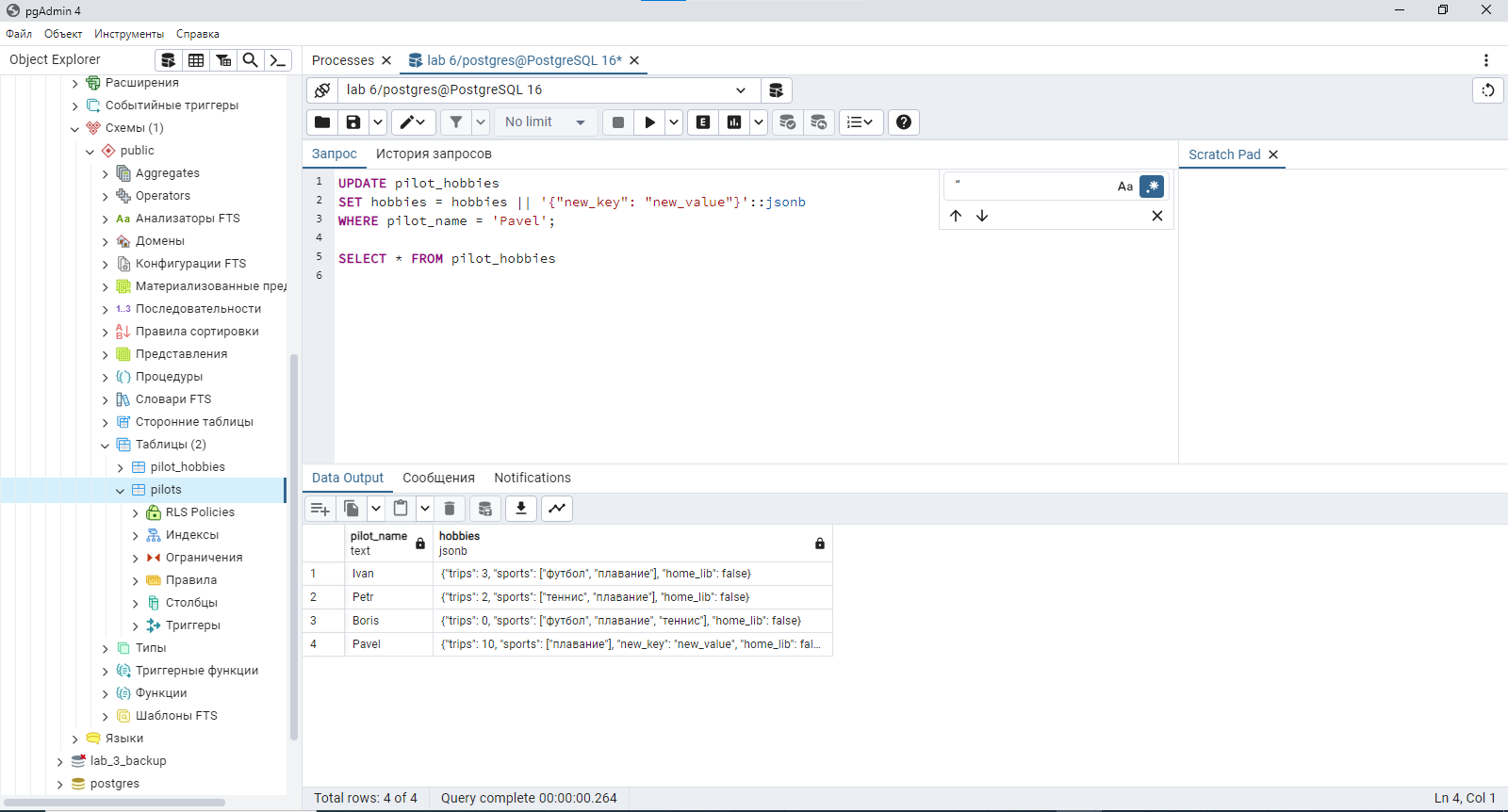
Добавьте дополнительный ключ и соответствующее ему значение в JSON-объект какой-нибудь строки таблицы pilots.

Используйте оператор | |.

UPDATE pilot\_hobbies

SET hobbies = hobbies || '{"new\_key": "new\_value"}'::jsonb

WHERE pilot\_name = 'Pavel';



**Задание 11**. Объекты JSON позволяют добавлять в них новые ключи, но также и удалять из них существующие.

Удалите один из ключей из JSON-объекта какой-нибудь строки таблицы pilots.

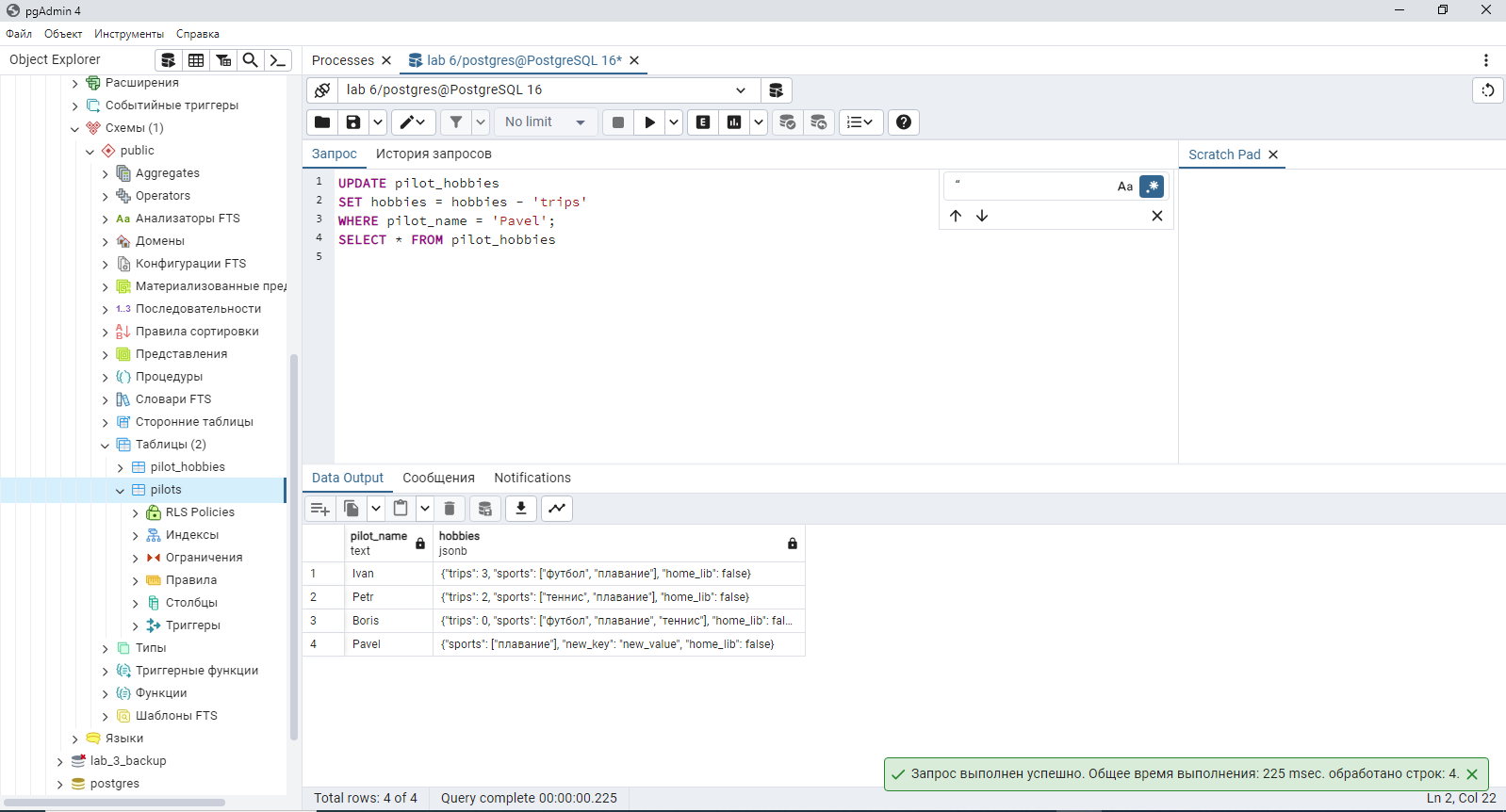
Соответствующее ему значение будет удалено, так как без ключа оно не может существовать!

Воспользуйтесь оператором -.

UPDATE pilot\_hobbies

SET hobbies = hobbies - 'trips'

WHERE pilot\_name = 'Pavel';

SELECT \* FROM pilot\_hobbies****

Вывод: в хоже лабораторной работы я изучил типы данных массив и JSON-объект.