Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Новосибирский государственный технический университет»

Кафедра вычислительной техники

**Отчет**

По лабораторной работе №3

По дисциплине: «Базы данных»

Использование агрегатных функций в среде PostgreSQL. Массивы.

Вариант 10

Выполнили: Павлюк А.С., Преподаватель: Харюткина С. А.

Казанцев К. О.

Группа: АВТ-008

Факультет: АВТ

Новосибирск, 2022

Оглавление

[Цель работы 3](#_Toc113394637)

[Задание 3](#_Toc113394638)

[Структура базы данных 4](#_Toc113394639)

[Ход работы 4](#_Toc113394640)

[Вывод 10](#_Toc113394641)

[Приложение 11](#_Toc113394642)

Цель работы

Изучить базовые операции по работе с массивами. Изучить синтаксис команд. Приобрести навыки работы с агрегатными функциями в PostgreSQL.

Задание

Ознакомиться с теоретическими сведениями о создании массивов. Создать таблицу с полем-массивом, таблицу с полем, содержащим многомерный массив. Выполнить вставку значений в созданные таблицы (минимум по 6 записей в каждой). Выполнить выборку из созданных таблиц (в том числе продемонстрировать предотвращение выборки NULL в массивах). Осуществить выборку с использованием среза. Продемонстрировать работу функции array\_dims(). Выполнить обновление данных в созданных таблицах. Осуществить модификацию среза массива, отдельного элемента массива. Проанализировать полученную в результате выполнения операций информацию.

*Вариант 10:* заполнить базу данных туристического агентства (при необходимости). Найти авиатуры с минимальной стоимостью. Найти железнодорожные туры с максимальной стоимостью. Найти количество автобусных туров. Найти среднюю стоимость туров в город Москву. Найти общую стоимость туров, выполненных руководителями туров со стажем работы более 10 лет. Продемонстрировать результаты работы.

Структура базы данных

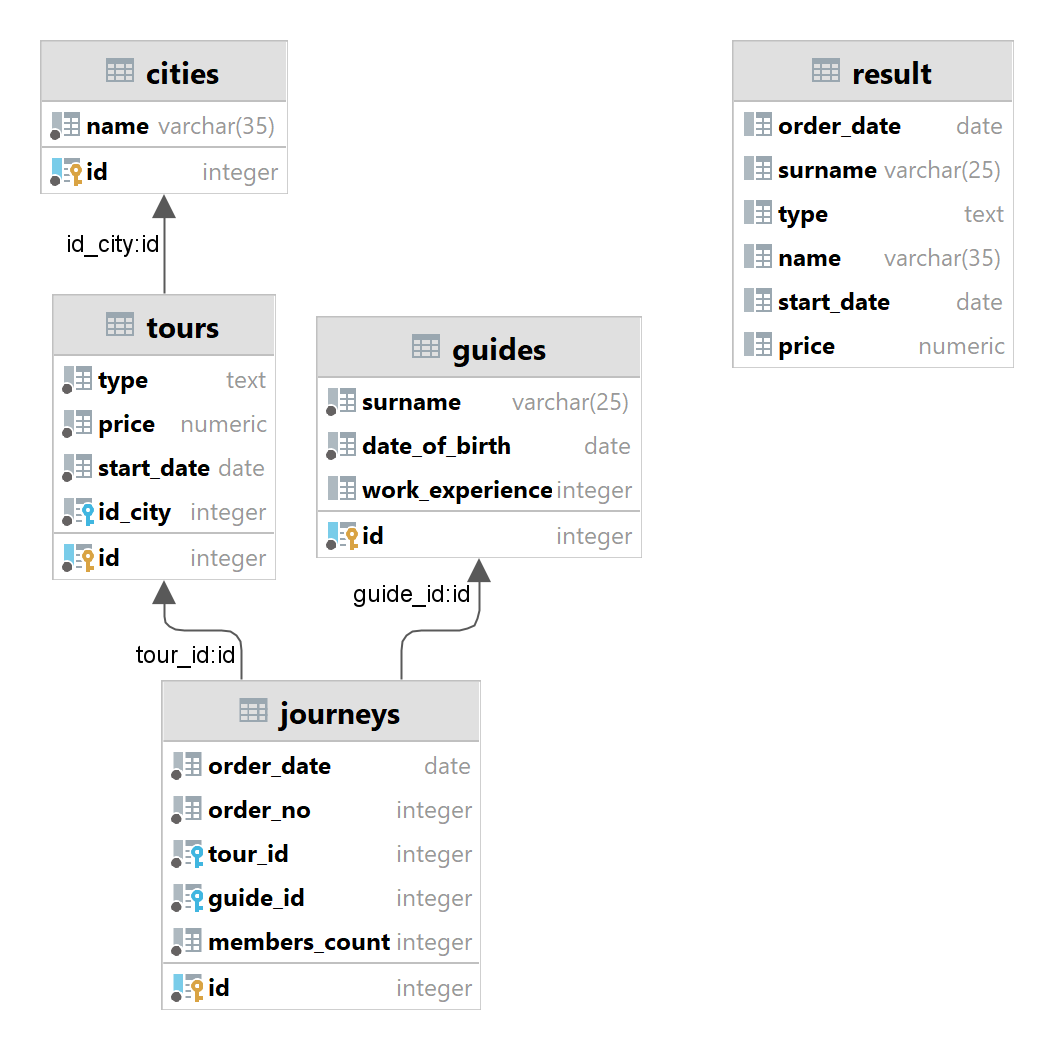


Рисунок 1. Структура базы данных

Ход работы

Поля данных PostgreSQL вместо отдельных величин могут содержать конструкции, называемые массивами.

Чтобы создать простейшее поле-массив, включите в команду CREATE TABLE или ALTER TABLE пару квадратных скобок после имени поля. Квадратные скобки показывают, что вместо одного значения в поле может храниться массив указанного типа.

Например, команда для создания поля single\_array типа type выглядит так:

single\_array type[] -- Одномерный массив

Дополнительные квадратные скобки определяют многомерные массивы, то есть «массивы массивов». Пример:

multi\_array type[][] -- Многомерный массив

Теоретически в квадратных скобках можно указать целое число, чтобы созданный массив имел фиксированный размер (то есть всегда состоял из п элементов по указанному измерению и не более). Но в текущей реализации это не учитывается.

Создадим таблицу контактов с полем массивом и двумерным массивом.

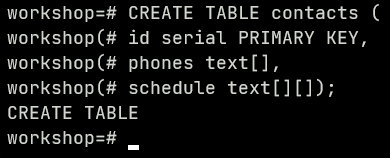


Рисунок 2. Создание таблицы с массивами

Приведем пример запроса для заполнения одномерного массива и двумерного:

INSERT INTO contacts VALUES (default, '{"(383)-123-45-67", "(383)-890-12-34", "(383)-567-89-01"}', '{{"09:00", "18:00"}, {"09:00", "18:00"}, {"09:00", "18:00"}, {"10:00", "18:00"}, {"10:00", "18:00"}}');

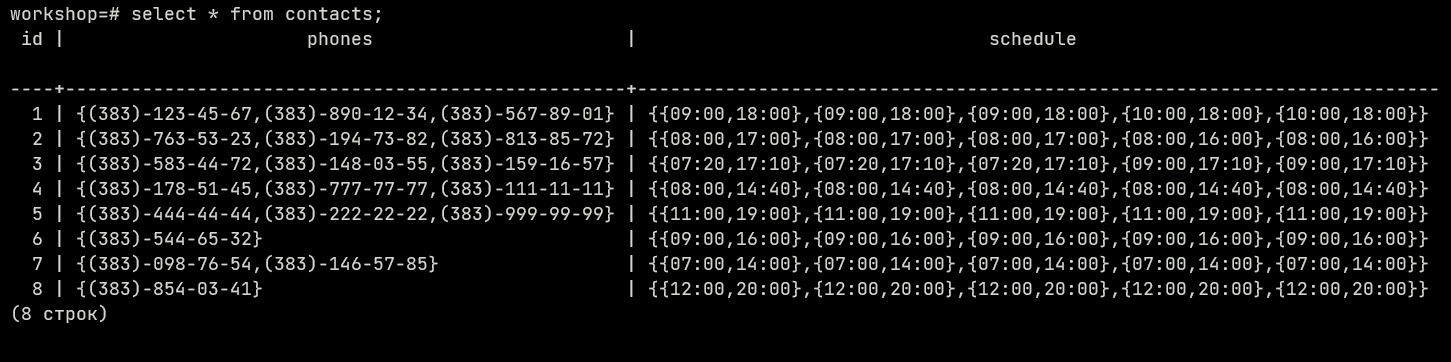


Рисунок 3. Заполнение массивов в таблице

Продемонстрируем результаты выборок полей массивов.

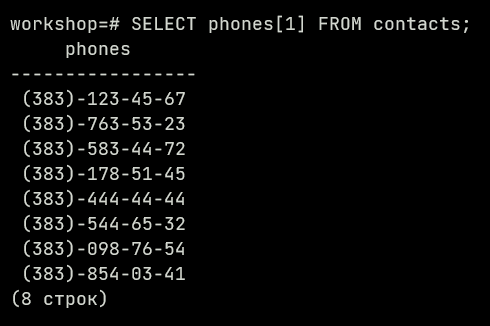


Рисунок 4. Выборка элементов одномерного массива

Если выйти за пределы массива, выборка будет состоять из NULL значений.

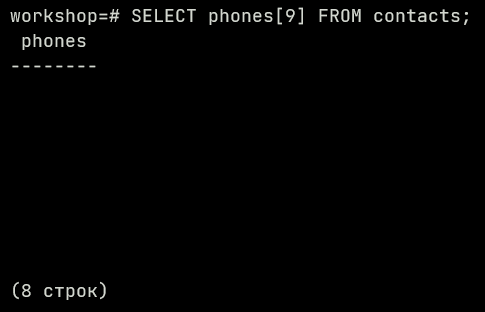


Рисунок 5. Выборка элементов за пределами массива

Это можно предотвратить, добавив условие IS NOT NULL.

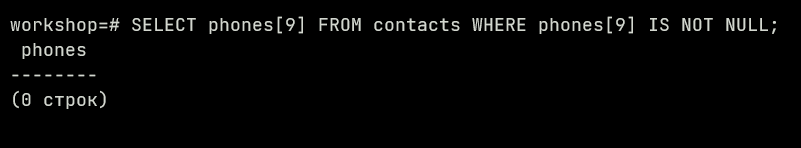


Рисунок 6. Предотвращение выборки NULL в массивах

Получим время начала работы первого контакта со вторника по четверг с помощью среза.

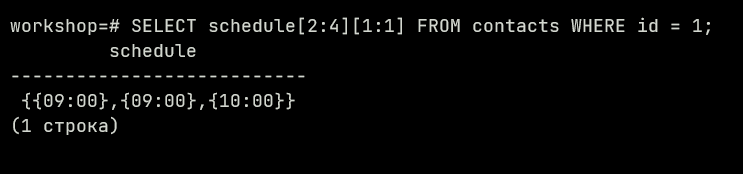


Рисунок 7. Срез двумерного массива

Теперь найдем третьи телефоны всех контактов.

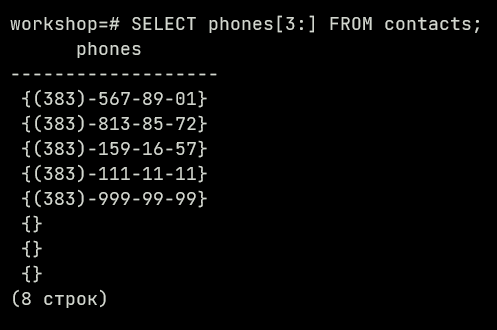


Рисунок 8. Срез одномерного массива

Функция array\_dims() возвращает текущие размеры значения массива.

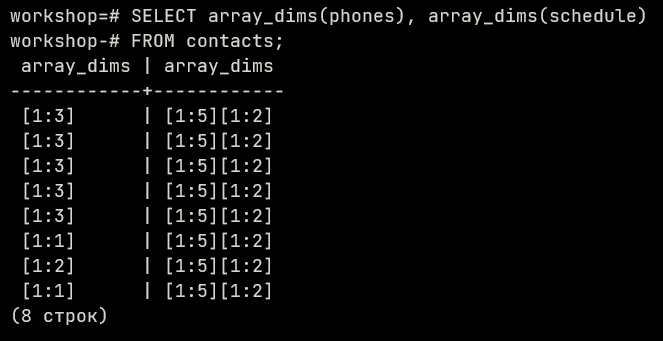


Рисунок 9. Пример работы функции array\_dims()

Обновим срез массива часов работы контактов. Установим начало рабочего дня в 12:00 в четверг и пятницу.

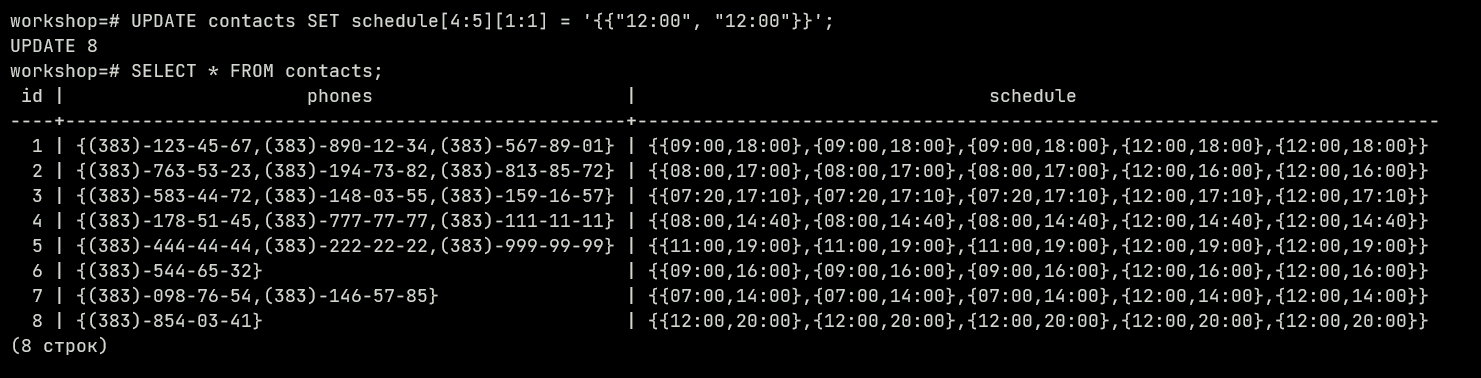


Рисунок 10. Обновление среза массива

Изменим второй телефон у пятого контакта.

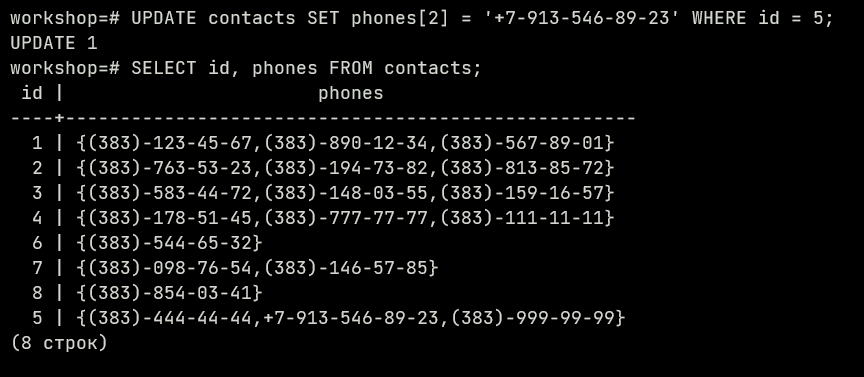


Рисунок 11. Обновление элемента массива

Поработав с массивами, можно сделать вывод, что одномерные массивы допустимы в схеме таблиц. Что касается многомерных массивов, как мне кажется, лучше выносить их в отдельные таблицы, чтобы пользователь мог легче обрабатывать информацию.

Перейдем к агрегатным функциям.

В SQL существует ряд специальных стандартных функций (SQL-функций). Кроме специального случая *COUNT(\*)* каждая из этих функций 10

оперирует совокупностью значений столбца некоторой таблицы и создает единственное значение.

Найдем авиатуры с минимальной стоимостью.

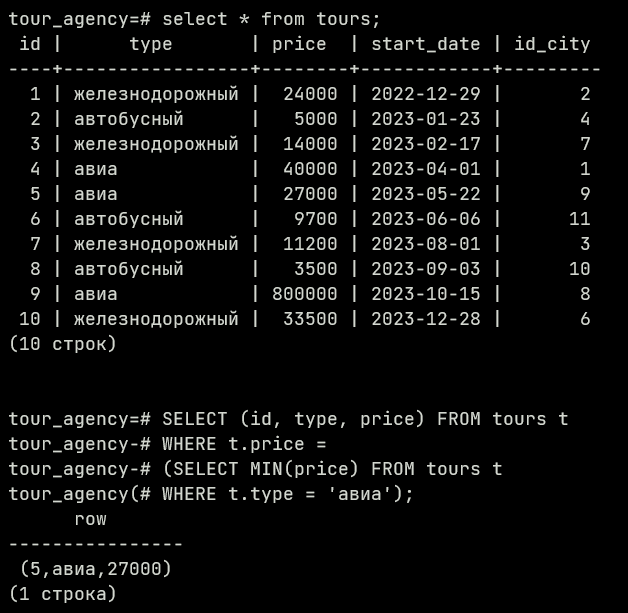


Рисунок 12. авиатуры с минимальной стоимостью

Далее найдем железнодорожные туры с максимальной стоимостью.

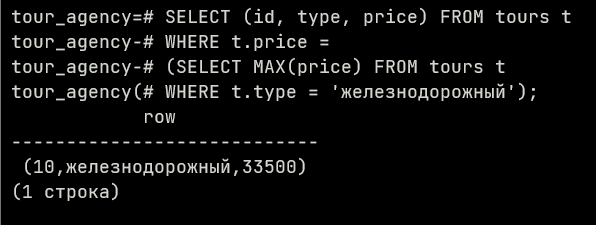


Рисунок 13. Железнодорожные туры с максимальной стоимостью

Найдем количество автобусных туров.

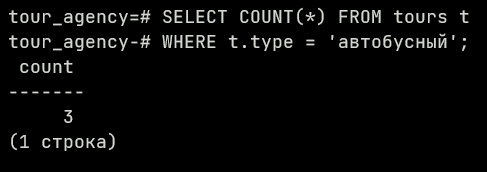


Рисунок 14. Количество автобусных туров

Теперь найдем среднюю стоимость туров в Москву.

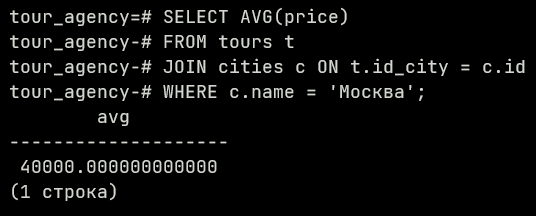


Рисунок 15. Средняя стоимость туров в Москву

Найдем общую стоимость туров, выполненных руководителями со стажем работы более 10 лет. Для этого добавим столбец стажа в таблицу гидов.

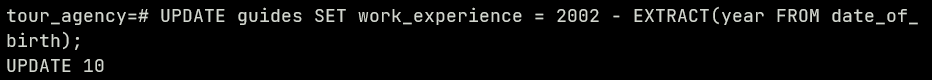


Рисунок 16. Добавление столбца стажа работы

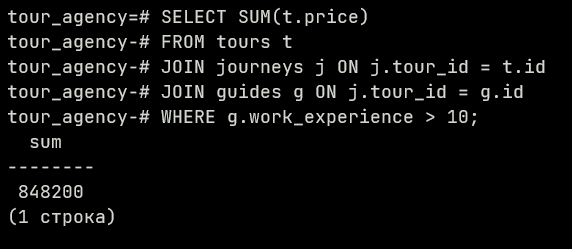


Рисунок 17. Общая стоимость туров, выполненных гидами со стажем более 10 лет

Вывод

В результате выполнения лабораторной работы были изучены базовые операции по работе с массивами. Также были приобретены навыки работы с агрегатными функциями в PostgreSQL.

Приложение

CREATE TABLE contacts (

id serial PRIMARY KEY,

phones text[],

schedule text[][]);

INSERT INTO contacts VALUES (default, '{"(383)-123-45-67", "(383)-890-12-34", "(383)-567-89-01"}', '{{"09:00", "18:00"}, {"09:00", "18:00"}, {"09:00", "18:00"}, {"10:00", "18:00"}, {"10:00", "18:00"}}');

SELECT phones[1] FROM contacts;

SELECT schedule[2:4][1:1] FROM contacts WHERE id = 1;

SELECT array\_dims(phones), array\_dims(schedule) FROM contacts;

UPDATE contacts SET schedule[4:5][1:1] = '{{"12:00", "12:00"}}';

UPDATE contacts SET phones[2] = '+7-913-546-89-23' WHERE id = 5;

SELECT (id, type, price) FROM tours t

WHERE t.price =

(SELECT MIN(price) FROM tours t

WHERE t.type = 'авиа');

SELECT (id, type, price) FROM tours t

WHERE t.price =

(SELECT MAX(price) FROM tours t

WHERE t.type = 'железнодорожный');

SELECT COUNT(\*) FROM tours t

WHERE t.type = 'автобусный';

SELECT AVG(price)

FROM tours t

JOIN cities c ON t.id\_city = c.id

WHERE c.name = 'Москва';

ALTER TABLE guides ADD COLUMN work\_experience integer;

UPDATE guides SET work\_experience = 2002 - EXTRACT(year FROM date\_of\_birth);

SELECT SUM(t.price)

FROM tours t

JOIN journeys j ON j.tour\_id = t.id

JOIN guides g ON j.tour\_id = g.id

WHERE g.work\_experience > 10;