

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Уфимский государственный авиационный технический университет"**

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Базы данных

Отчет по лабораторной работе № 3

Тема: «Основы SQL-запросов. JOIN, агрегирование и группировка,
подзапросы»

Группа ПМ-353	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Ямилева А.М.			

Уфа 2022

Теоритическая часть

Для подключения базы данных используется команда

```
psql -d demo -U postgres
```

Для создания таблиц в языке SQL служит команда CREATE TABLE.

Для удаления таблицы служит команда DROP TABLE;

Для выполнения вводу данных в таблицу служит команда INSERT. Ее упрощенный формат таков:

```
INSERT INTO имя-таблицы [( имя-атрибута, имя-атрибута, ... )] VALUES  
( значение-атрибута, значение-атрибута, ... );
```

Для выборки информации из таблиц базы данных служит команда SELECT. Ее синтаксис, упрощенный до предела, таков:

```
SELECT имя-атрибута, имя-атрибута, ... FROM имя-таблицы;
```

Теперь мы ознакомимся с командой UPDATE, предназначенной для обновления данных в таблицах. Ее упрощенный синтаксис таков:

```
UPDATE имя-таблицы SET имя-атрибута1 = значение-атрибута1, имя-  
атрибута2 = значение-атрибута2, ... WHERE условие;
```

Типы данных СУБД PostgreSQL

Группа числовых типов данных включает в себя целый ряд разновидностей: целочисленные типы, числа фиксированной точности, типы данных с плавающей точкой, последовательные типы (serial). В составе целочисленных типов находятся следующие представители: smallint, integer, bigint. Если атрибут таблицы имеет один из этих типов, то он позволяет хранить только целочисленные данные. При этом перечисленные типы различаются по количеству байтов, выделяемых для хранения данных.

Числа фиксированной точности представлены двумя типами — numeric и decimal. Однако они являются идентичными по своим возможностям. Для задания значения этого типа используются два базовых понятия: масштаб (scale) и точность (precision). Масштаб показывает число значащих цифр, стоящих справа от десятичной точки (запятой). Точность указывает общее число цифр как до десятичной точки, так и после нее.

Стандартные представители строковых типов — это типы character varying(n) и character(n), где параметр указывает максимальное число символов в строке, которую можно сохранить в столбце такого типа. При работе с многобайтовыми кодировками символов, например UTF-8, нужно учитывать, что речь идет о символах, а не о байтах. Если сохраняемая строка

символов будет короче, чем указано в определении типа, то значение типа `character` будет дополнено пробелами до требуемой длины, а значение типа `character varying` будет сохранено так, как есть.

PostgreSQL поддерживает все типы данных, предусмотренные стандартом SQL для даты и времени. Даты обрабатываются в соответствии с григорианским календарем, причем это делается даже в тех случаях, когда дата относится к тому моменту времени, когда этот календарь в данной стране еще не был принят. Для этих типов данных предусмотрены определенные форматы для ввода значений и для вывода. Причем эти форматы могут не совпадать.

В результате объединения типов даты и времени получается интегральный тип — временная отметка. Этот тип существует в двух вариантах: с учетом часового пояса — `timestamp with time zone`, либо без учета часового пояса — `timestamp`. Для первого варианта существует сокращенное наименование — `timestampz`, которое является расширением PostgreSQL. При вводе и выводе значений этого типа данных используются соответствующие форматы ввода и вывода даты и времени.

Для получения значения текущей временной отметки (т. е. даты и времени в одном 'значении) служит функция `current_timestamp`.

Последним типом является `interval`, который представляет продолжительность отрезка времени между двумя моментами времени. Его формат ввода таков: `quantity unit [quantity unit ...] direction`

Логический (`boolean`) тип может принимать три состояния: истина и ложь, а так же неопределенное состояние, которое можно представить значением `NULL`. Таким образом, тип `boolean` реализует трехзначную логику.

Практическая часть

Задание 1

```
demo=# SELECT count( * ) FROM tickets;
count
-----
366733
(1 строка)

demo=# SELECT count( * ) FROM tickets WHERE passenger_name LIKE '% %';
count
-----
366733
(1 строка)

demo=# SELECT count( * ) FROM tickets WHERE passenger_name LIKE '% % %';
count
-----
0
(1 строка)

demo=# SELECT count( * ) FROM tickets WHERE passenger_name LIKE '% % %';
count
-----
366733
(1 строка)
```

Задание 2

```
demo=# WHERE passenger_name LIKE '% _____';
passenger_name
-----
ILYA POPOV
VLADIMIR POPOV
PAVEL GUSEV
LEONID ORLOV
EVGENIY GUSEV
NIKOLAY FOMIN
EKATERINA ILINA
ANTON POPOV
ARTEM BELOV
VLADIMIR POPOV
ALEKSEY ISAEV
EMIL ISAEV
SERGEY ISAEV
IVAN POPOV
ALEKSANDR ORLOV
ALEKSEY ORLOV
```

Задание 3

```
demo=# SELECT count( * ) FROM tickets WHERE passenger_name SIMILAR TO '% %';
count
-----
366733
(1 строка)

demo=# SELECT count( * ) FROM tickets WHERE passenger_name SIMILAR TO '(A|B)%';
count
-----
71511
(1 строка)

demo=# SELECT passenger_name FROM tickets WHERE passenger_name SIMILAR TO '(A|B)%';
passenger_name
-----
ARTUR GERASIMOV
ALINA VOLKOVA
ALEKSANDR TIKHONOV
ALEKSANDR KUZNECOV
ANTON BONDARENKO
ALLA TARASOVA

demo=# SELECT passenger_name FROM tickets WHERE passenger_name SIMILAR TO '_+(M|R) %';
passenger_name
-----
ARTUR GERASIMOV
MAKSIM ZHUKOV
ALEKSANDR TIKHONOV
ALEKSANDR KUZNECOV
VIKTOR IVANOV
ALEKSANDR DENISOV
VLADIMIR POPOV
VLADIMIR ANTONOV
VLADIMIR SEMENOV
ALEKSANDR NAUMOV
VLADIMIR MIRONOV
ALEKSANDR SMIRNOV
EGOR NIKOLAEV
VLADIMIR AFANASEV
ALEKSANDR MAKSIMOV
ALEKSANDR FEDOROV
```

Задание 5

```
demo=# SELECT coalesce(1,2,3,4,5);
coalesce
-----
1
(1 строка)

demo=# SELECT coalesce(NULL,NULL,3,4,5);
coalesce
-----
3
(1 строка)
```

```
demo=# SELECT nullif(1,2);
nullif
-----
      1
(1 строка)

demo=# SELECT nullif(1,1);
nullif
-----
(1 строка)
```

```
demo=# SELECT greatest(1,2,3,4,5);
greatest
-----
      5
(1 строка)

demo=# SELECT least(1,2,3,4,5);
least
-----
      1
(1 строка)
```

Задание 6

```
demo=# SELECT departure_city,arrival_city,model FROM routes,aircrafts WHERE aircrafts.model LIKE ('Боинг %') AND aircrafts.aircraft_code=routes.aircraft_code;
```

departure_city	arrival_city	model
Сочи	Москва	Боинг 777-300
Краснодар	Новокузнецк	Боинг 737-300
Горно-Алтайск	Ставрополь	Боинг 737-300
Ставрополь	Горно-Алтайск	Боинг 737-300
Магадан	Минеральные Воды	Боинг 767-300
Минеральные Воды	Магадан	Боинг 767-300
Москва	Мирный	Боинг 737-300
Мирный	Москва	Боинг 737-300
Екатеринбург	Комсомольск-на-Амуре	Боинг 767-300
Комсомольск-на-Амуре	Екатеринбург	Боинг 767-300
Москва	Петропавловск-Камчатский	Боинг 767-300
Петропавловск-Камчатский	Москва	Боинг 767-300
Анапа	Новокузнецк	Боинг 737-300
Новокузнецк	Анапа	Боинг 737-300
Москва	Владивосток	Боинг 767-300
Владивосток	Москва	Боинг 767-300
Москва	Хабаровск	Боинг 767-300
Хабаровск	Москва	Боинг 767-300
Москва	Минеральные Воды	Боинг 737-300
Минеральные Воды	Москва	Боинг 737-300
Москва	Самара	Боинг 767-300

Задание 7

```
demo=# SELECT departure_city,arrival_city,model FROM routes r,aircrafts a WHERE a.model='Боинг 777-300' AND a.aircraft_code=r.aircraft_code;
```

departure_city	arrival_city	model
Сочи	Москва	Боинг 777-300
Москва	Новосибирск	Боинг 777-300
Новосибирск	Москва	Боинг 777-300
Москва	Сочи	Боинг 777-300
Москва	Екатеринбург	Боинг 777-300
Екатеринбург	Москва	Боинг 777-300
Москва	Новосибирск	Боинг 777-300
Новосибирск	Москва	Боинг 777-300
Москва	Пермь	Боинг 777-300
Пермь	Москва	Боинг 777-300

(10 строк)

```
demo=# SELECT departure_city,arrival_city,model FROM routes r,aircrafts a WHERE a.model='Боинг 777-300' AND a.aircraft_code=r.aircraft_code AND departure_city < arrival_city;
```

departure_city	arrival_city	model
Москва	Новосибирск	Боинг 777-300
Москва	Сочи	Боинг 777-300
Екатеринбург	Москва	Боинг 777-300
Москва	Новосибирск	Боинг 777-300
Москва	Пермь	Боинг 777-300

(5 строк)

Задание 9

```
demo=# SELECT departure_city , arrival_city , count( * )
demo=# FROM routes
demo=# WHERE departure_city = 'Москва'
demo=# AND arrival_city = 'Санкт-Петербург'
demo=# GROUP BY departure_city, arrival_city
demo=# ORDER BY count DESC;
```

departure_city	arrival_city	count
Москва	Санкт-Петербург	12

(1 строка)

Список литературы

1. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.