

РАБОТА С SQL-ЗАПРОСАМИ

| ВВЕДЕНИЕ

На этом уроке мы узнаем

- Как выглядит структура SQL запроса
- Какие есть особенности синтаксиса запросов в базе Postgres

На практике разберём

- Примеры простейших запросов

| SQL-ЗАПРОС

SQL-запрос

это набор команд для работы с реляционными базами данных, который определяет какие данные и из какого источника необходимо получить, а также каким образом эти данные отфильтровать или преобразовать

SQL запрос для формирования выборки данных из базы содержит **следующие стандартные инструкции**:

о о о

```
SELECT column1 [, column2 , ...]
FROM table1 [, table2, ... ]
[ WHERE conditions ]
[ GROUP BY column1 [, column2, ... ] ]
[ HAVING conditions ]
[ ORDER BY column1 [, column2, ... ] [ ASC | DESC ] ]
```

- 01** В первую очередь **вводится ключевое слово SELECT**, после которого указываются название колонок (столбцов) таблицы данных, которые необходимо извлечь, то есть то, что мы хотим получить. В базе PostgreSQL названия столбцов должны вводиться в кавычках, другие СУБД могут иметь иные особенности синтаксиса
- 02** Далее **вводится ключевое слово FROM**, после которого указывается название таблицы, из которой будут извлекаться данные. В базе PostgreSQL название таблицы должно содержать её местонахождение внутри схемы базы данных

В нашем примере базы данных вкладка Tables находится внутри схемы public, поэтому для выбора нужно таблицы нужно указать вначале слово public а затем через точку наименование таблицы, из которой мы хотим получить данных, например: public.customers

03 Команды SELECT и FROM, определяющие что и откуда необходимо выгрузить, должны содержаться в запросе в обязательном порядке, инач запрос просто не будет работать. Остальные элементы запроса могут применяться опционально по необходимости

Сразу разберем примеры простейших SQL-запросов, содержащих только команды SELECT и FROM (переход на базу данных)

| ПРИМЕРЫ

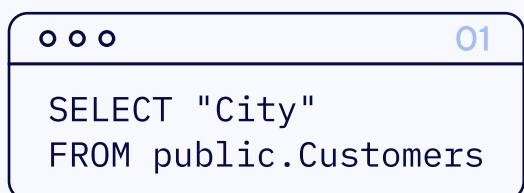
Выбрать из таблицы Customers (клиенты) данные о городах проживания клиентов (столбец «City»)

Простейший способ создать SQL-запрос, позволяющий посмотреть, что находится в той или иной таблице базы данных – найти эту таблицы в списке таблиц во вкладке Tables, затем щёлкнуть на ней правой кнопкой мыши и в пункте меню View/Edit Data выбрать ALL Rows

В этом случае автоматически вверху создаётся новая вкладка с SQL-запросом, выгружающим все данные из таблицы, а внизу мы увидим результат отработки этого запроса. Однако, в таком варианте мы не можем редактировать сформированный автоматически запрос, если попробуем что-либо изменить, то увидим, что это сделать не удаётся

Для того чтобы создавать свой собственный произвольный SQL-запрос, нам необходимо воспользоваться функцией Query Tool, которая находится на верхней панели слева, либо по нажатию клавиш Alt Shift Q

После нажатия откроется новая вкладка Query для нашей базы Northwind, в которой мы может создавать уже свой собственный запрос



```
01  
SELECT "City"  
FROM public.Customers
```

По нажатию клавиши F5 можно запустить введённый запрос и посмотреть на полученный нами результат

Вверху мы можем также написать комментарий, который раскрывает суть введенного SQL-запроса. Для этого мы можем сверху ввести текст, например: «1. Выбрать данные о городах проживания клиентов» и добавить слева два символа дефиса. Как мы видим, после добавления двух дефисов у нас текст окрасился в бледно-красный цвет, то есть стал закомментирован. Благодаря этому, интерпретатор кода понимает, что этот фрагмент текста не является исполняемым кодом, а является комментарием

Как видим, у нас в таблице есть достаточно много повторяющихся строк. Предположим, нас интересует только Выбрать из таблицы Customers (клиенты) только уникальные названия городов проживания клиентов (столбец «City»)

о о о 02

```
SELECT DISTINCT "City"  
FROM public.Customers
```

Если мы хотим выбрать все данные из выбранной таблицы, но надо использовать после SELECT символ «*»

о о о 03

```
SELECT *  
FROM public.Customers
```

Также после SELECT можно использовать агрегатные функции для получения суммы, среднего или иных значений по выбранным числовым столбцам таблицы. Пример: получить из таблицы order_details средний размер скидки и сумму количества проданных товаров:

о о о

```
SELECT AVG("Discount"), SUM("Quantity")  
FROM public.order_details
```

Чтобы изменить названия столбцов в результате запроса, можно добавить после столбца конструкцию AS «NewName» для переименования колонки:



○○○

04

```
SELECT AVG("Discount") AS "Mid_discount",
SUM("Quantity") AS "Sum_quantity"
FROM public.order_details
```

После этого заголовки столбцов переименовались в указанные нами названия. Под названиями колонок указывается тип данных итогового столбца: для средней скидки тип данных – число с плавающей точкой («double precision»), для столбца Количество – тип данных целое число (обозначается как bigint)

Тип данных столбца при желании также можно изменить. Например, мы хотим представить сумму по столбцу количество тоже в виде числа с плавающей точкой. Для этого можно использовать ключевое слово CAST с присвоением выбранного нами типа данных:

○○○

05

```
SELECT AVG("Discount") AS "Mid_discount",
CAST(SUM("Quantity") AS float) AS "Sum_quantity"
FROM public.order_details
```

После запуска этого запроса мы видим, что типа данных столбца количество поменялся с bigint на double precision

Также мы можем округлить полученные значения до 2 знаков. Для этого необходимо воспользоваться функцией ROUND, а после вычисления среднего значения функцией AVG указать ::numeric

○○○

06

```
SELECT ROUND(AVG("Discount")::numeric, 2) AS
"Mid_discount", CAST(SUM("Quantity") AS float) AS
"Sum_quantity"
FROM public.order_details
```

Наконец, мы можем выгрузить полученные нами результаты в csv-файл и посмотреть его через excel. Для этого нужно нажать кнопку Save results to excel file на панели в окошке Data Output, либо клавишу F8 на клавиатуре

После выгрузки мы можем открыть полученный результат через excel, после чего можно отправить результат и продолжить работу с ним. Также отметим, что зачастую данные из csv-файла в excel открываются без разделения по столбцам. В этом случае нужно проделать следующую операцию:

Выделить столбец с данными, затем на вкладке Данные открыть раздел Текст по столбцам, затем выбрать разделитель запятая, нажать Далее и Готово, после чего получаем таблицу разделённую на разные столбцы

I ИТОГ

- ☑ Вы узнали о функционале pgAdmin для создания собственных SQL-запросов, разобрали содержание SQL-запроса, и на практике научились создавать простейшие запросы на языке SQL для решения различных поставленных задач

