

ЧАСТЬ 1

ИСПОЛЬЗОВАНИЕ GROUP BY, HAVING И АГРЕГИРУЮЩИХ ФУНКЦИЙ

| ВВЕДЕНИЕ

На этом уроке мы разберём

- Работу с запросами, содержащими группировки данных посредством GROUP BY
- Использование агрегирующих функций и фильтров по ним посредством HAVING
- Основы работы с документацией базы данных Postgres для нахождения необходимых для работы функций и их описания

| GROUP BY И HAVING

Операторы GROUP BY и HAVING в SQL используются для агрегации данных и фильтрации агрегированных результатов в соответствии с заданными критериями

Давайте рассмотрим их предназначение и область применения

Оператор GROUP BY

используется для группировки строк в результирующем наборе данных на основе значений в одном или нескольких столбцах. После применения GROUP BY данные разделяются на группы, где каждая группа содержит уникальные комбинации значений, указанные в операторе GROUP BY. Оператор GROUP BY используется совместно с агрегатными функциями (например, SUM, COUNT, AVG), чтобы агрегировать данные внутри каждой группы

Оператор HAVING

применяется в SQL после оператора GROUP BY для фильтрации результатов агрегации. Он позволяет устанавливать условия для отбора групп, основываясь на результатах агрегации. Оператор HAVING действует как фильтр по результатам группировки, оставляя только те группы, которые удовлетворяют заданным условиям. Это позволяет фильтровать группы по значениям агрегирующих функций. И в этом главное отличие оператора HAVING от рассмотренного нами ранее оператора WHERE, который позволяет фильтровать таблицу только по значениям исходных колонок

! Этот момент нужно чётко запомнить

в случае если запрос содержит группировку данных, то фильтрацию нужно осуществлять посредством HAVING, если не содержит группировки – то посредством WHERE

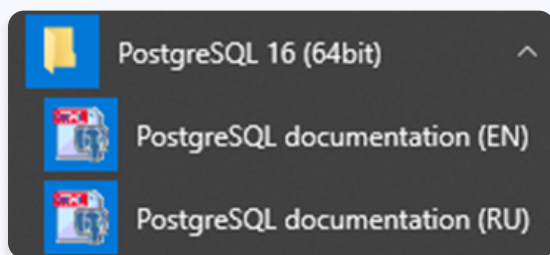
Что из себя представляют агрегирующие функции языка SQL и где можно получить о них полную информацию?

Поиск доступных для использования агрегирующих функций можно осуществлять в документации конкретной базы данных

Документация СУБД Postgres может быть получена по ссылке

<file:///C:/Program%20Files/PostgreSQL/16/doc/ru/sql.html>

Также открыть документацию можно, зайдя в раздел документация в папке Postgres в меню ПУСК и выбрав Документация RU



В документации дается полное и детальное описание использования языка SQL в PostgreSQL : описание общего синтаксиса SQL, того как создавать структуры для хранения данных, как наполнять базу данных и как выполнять запросы к ней. Также в документации перечислены существующие типы данных и функции, применяемые с командами SQL

Для поиска нужных нам агрегирующих функций необходимо в документации перейти во вкладку 9.21 «Агрегатные функции» раздела

Функции и операторы

<file:///C:/Program%20Files/PostgreSQL/16/doc/ru/functions-aggregate.html>

Перейдя в этот раздел мы увидим полную таблицу, содержащую данные обо всех имеющихся агрегатных функциях. Как мы видим, они делятся на несколько основных типов:

01 Агрегатные функции общего назначения – это наиболее часто используемые функции. В первую очередь это функции sum – суммирование, count – подсчёт количества строк, avg – расчет среднего значения и так далее. Также рядом с каждой агрегатной функций указывается тип данных показателя, который возвращается в результате применения этой агрегатной функции к столбцу

- 02** Агрегатные функции для статистических вычислений
– это в первую очередь функции расчёт математического ожидания, стандартного отклонения и дисперсии совокупности данных
- 03** Сортирующие агрегатные функции
- 04** Гипотезирующие агрегатные функции
- 05** Операции группировки

| ПРИМЕРЫ

Перейдём к применению группировок и агрегатных функций на примере конкретных практических задач

- 01** Сколько клиентов проживают в Лондоне?

```
ooo
```

```
SELECT COUNT(*) AS count_customers
FROM public.customers
WHERE "City" LIKE 'London'
```

- 02** Найдите суммарное количество единиц проданной товара с ID равном 1

```
ooo
```

```
SELECT SUM("Quantity") AS Sum
FROM Order_Details
WHERE "ProductID" = '1'
```

- 03** Найдите прибыль по 1 товару

```
ooo
```

```
SELECT SUM("Quantity"*"UnitPrice"*(1-"Discount")) AS Sum
FROM Order_Details
WHERE "ProductID" = '1'
```

- 04** Найдите суммарную прибыль по каждому товару

```
ooo
```

```
SELECT "ProductID", SUM("Quantity"*"UnitPrice"*(1-"Discount")) AS Sum
FROM Order_Details
GROUP BY "ProductID"
ORDER BY "ProductID"
```

I ИТОГ

- ✓ По итогам урока мы научились пользоваться документацией по базе данных PostgreSQL, находить и применять в запросах агрегирующие функции, получать выгрузки данных из базы в требуемой группировке посредством использования оператора GROUP BY