



> Конспект > 2 урок > SQL

> Оглавление

1. `GROUP BY`
2. `HAVING`
3. Агрегатные функции

> GROUP BY

Как правило, в базах данных данные хранятся с максимальной детализацией, порой даже излишней. Например: чеки из магазинов, которые поступают почти сразу после совершения покупки. Каждый такой чек имеет дату и время совершения покупки с точностью до секунд. И таких чеков у вас огромное количество. Но задача – показать продажи определенного товара по месяцам. Что делать в таком случае? Специально для решения таких задач используется блок группировки – `GROUP BY`.

`GROUP BY` – группирует/агрегирует данные по какому-либо полю. В блоке `GROUP BY` задаются поля, по которым будет сгруппирован результат. Каждый столбец из `SELECT` должен использоваться либо в группировке, либо внутри агрегатной функции, но не в обоих.

– нельзя агрегировать ту же колонку, по которой группируете - есть что-то есть в агрегации, то в `GROUP BY` этих столбцов быть не должно

– если в блоке `SELECT` у нас есть столбцы под агрегатной функцией, то не вошедшие в агрегацию столбцы должны быть в `GROUP BY`

Примеры запросов:

```
-- группируем по Source и считаем число строк
SELECT
    Source,
    COUNT(*) AS NumRows
FROM
    installs
GROUP BY
    Source
LIMIT 100
```

Пример как работает группировка с суммой:

– исходная табличка

▼ Страна	Аа Город	☰ Район	☰ Улица	⌘ Кол-во домов
Россия	<u>Москва</u>	Центральный	Ул. 1	56
Россия	<u>Москва</u>	Центральный	Ул. 2	73
Россия	<u>Самара</u>	Центральный	Ул. 3	28
Россия	<u>Псков</u>	Центральный	Ул. 4	18
Россия	<u>Самара</u>	Вокзал	Ул. 5	39
Россия	<u>Псков</u>	Вокзал	Ул. 6	1
Россия	<u>Суздаль</u>	Вокзал	Ул. 7	99
Россия	<u>Владимир</u>	Аэропорт	Ул. 8	71
Беларусь	<u>Минск</u>	Аэропорт	Ул. 9	9
Беларусь	<u>Минск</u>	Аэропорт	Ул. 10	15
Беларусь	<u>Брест</u>	Рынок	Ул. 11	44

– Считаем, сколько в какой стране домов:

<u>Аа</u> Страна	<u>#</u> Сумма по полю Кол-во домов
<u>Россия</u>	385
<u>Беларусь</u>	114

> HAVING

С блоком `GROUP BY` периодически используют оператор `HAVING` (аналогичен блоку `WHERE`).

`HAVING` — позволяет фильтровать данные при использовании `GROUP BY`, или применительно к результирующей таблице. Главным отличием от `WHERE` является то, что `WHERE` выполняется перед агрегацией (группировкой), а `HAVING` — после.

Колонку с результатом агрегации можно передавать как используя алиас, так и через запись агрегатной функции. То есть:

```
SELECT
    uid,
    COUNT(*) AS NumRows
FROM
    users
GROUP BY
    uid
HAVING
    COUNT(*) > 10 -- ссылаемся на функцию COUNT(*)
```

```
SELECT
    uid,
    COUNT(*) AS NumRows
FROM
    users
GROUP BY
    uid
HAVING
    NumRows > 10 -- ссылаемся на алиас NumRows
```

Также в `HAVING` можно использовать агрегатные функции, которые не используются в результирующей таблице:

```

SELECT
    uid,
    COUNT(*) AS NumRows
FROM
    users
GROUP BY
    uid
HAVING
    NumRows > 10 -- ссылаемся на алиас NumRows
    AND SUM(clicks) > 100 -- добавляем еще одно условие, при этом не добавляем в SELECT

```

> Агрегатные функции

Агрегатные функции используются в блоке `SELECT` и в качестве параметра принимают наименования поля. К ним относятся:

- `COUNT()` – количество значений поля (без параметра считает кол-во строк, с полем – кол-во без NULL значений)
- `COUNT (DISTINCT)` – количество уникальных значений поля (`uniqExact()` и `count(DISTINCT)` – одно и тоже)
- `AVG()` – вычисляет среднее
- `MIN()` – показывает минимальное значение
- `MAX()` – показывает максимальное значение
- `SUM()` – показывает сумму элементов

Такие функции вычисляют заказанное значение на уровне группировки, заданном в секции `GROUP BY`.

Примеры:

1.

```

SELECT
    Source,
    uniqExact(DeviceID) AS installs
FROM
    installs
GROUP BY
    Source
LIMIT 100

```

```

+-----+-----+
| Source | installs |
+-----+-----+
| Source_1 | 20      |
| Source_19 | 3148    |
| Source_10 | 4       |
| Source_2 | 1393    |
| Source_20 | 9       |
+-----+-----+

```

2.

```

SELECT
    uniqExact(DeviceID) AS devices_one,
    COUNT(DISTINCT DeviceID) AS devices_two
FROM
    devices

```

```

+-----+-----+
| devices_one | devices_two |
+-----+-----+
| 9406636     | 9406636     |
+-----+-----+

```

Больше агрегатных функций в ClickHouse: [здесь](#)