GROUP BY

используется для объединения результатов выборки по одному или нескольким столбцам, работает с агрегаторами:COUNT, SUM, MIN, MAX, AVG. Неуверен но вроде бы если есть WHERE то GROUP BY ИДЁТ ПОСЛЕ НЕГО.

пример:

**SELECT snum, SUM(amt) AS amt**

**FROM orders**

**GROUP BY snum;** <--вернёт уникальные значения snum и каждому из этих значений сумму из amt значений.

пример: Выбрать каждого автора и посчитать сколько у каждого автора книг

**SELECT author, sum(amount), count(amount)**

**FROM book**

**GROUP BY author;**

пояснения:

как показано ниже, GROUP BY группирует одинаковые значения из указанного столбца (здесь author) в группы а затем выщитывает (берёт) только уникальное из каждой группы и потом проходит по таблице и складывает все значения amount где автор соответствует выбранному из группы. Другими словами например автор из группы Булгаков как уникальное значение, значит проходимся по таблице и если в ряду в столбце author стоит Булгаков то складываем значение из amount



Булгаков

Достаевский

Есенин

author | sum(amount)

| Булгаков М.А. | 8

| Достоевский Ф.М. | 12

| Есенин С.А. | 15

Пример:

Singer Album Year Sale

The Prodigy Invaders Must Die 2008 1200000

Drowning Pool Sinner 2001 400000

Massive Attack Mezzanine 1998 2300000

The Prodigy Fat of the Land 1997 600000

The Prodigy Music For Generation 1994 1500000

Massive Attack 100th Window 2003 1200000

Drowning Pool Full Circle 2007 800000

Massive Attack Danny The Dog 2004 1900000

Drowning Pool Resilience 2013 500000

**SELECT Singer, SUM(Sale) AS AllSales**

**FROM Artists**

**GROUP BY Singer;**

результат:

Drowning Pool 1700000

Massive Attack 5400000

The Prodigy 3300000

Пояснения: здесь происходит тоже что описано в примере выше. GROUP BY разбил всех Singer на группы (одному Singer одна группа) в каждую группу внёс все соответстующие строки из таблицы, затем из каждой группы взял только одно значение, тем самым получилось уникальный set Singer-ов.

**SELECT Singer, MAX(Year) AS Last\_Album\_Year**

**FROM Artists**

**GROUP BY Singer;**

результат:

Singer Last\_Album\_Year

Drowning Pool 2013

Massive Attack 2004

The Prodigy 2008

Предложение GROUP ВУ позволяет определять подмножество значений отдельно­го поля в терминах другого поля и применять функции агрегирования к полученному подмножеству.

я тут почитал некоторое время спустя и честно говоря нихуя не понятно из этого описания, надо писать чтоб хотябы самому потом понятно было а не заумные слова. Ещо приведём пример с таблицей pc, в ней есть модель, и цена на одинаковую модель может быть разной в зависимости откомплектации. Надо пощитать сколько штук моделей существует (производится) и какая средняя цена для каждой модели

SELECT pc.model, COUNT(model) as Qty\_model, AVG(price) as Avg\_price FROM PC group by model;

model Qty\_model Avg\_price

1121 3 850.0000

1232 4 425.0000

1233 4 875.0000

1260 1 350.0000

Мы применили group by к колонке model, количество моделей 1121 =3 шт. средняя цена 850.0000 и т.д. Если убрать выражение group by то произойдёт

ошибка. Агрегатные функции используются с group by,

Посчитать стоимость всех экземпляров каждого автора без учета книг «Идиот» и «Белая гвардия». В результат включить только тех авторов, у которых суммарная стоимость книг (без учета книг «Идиот» и «Белая гвардия») более 5000 руб. Вычисляемый столбец назвать Стоимость. Результат отсортировать по убыванию стоимости.

**SELECT author, SUM(amount\*price) AS Стоимость**

**FROM book**

**WHERE title!='Идиот' AND title!='Белая гвардия'**

**GROUP BY author**

**HAVING sum(amount\*price)>5000**

**ORDER BY author DESC**

author | Стоимость

| Есенин С.А. | 9750.00

| Достоевский Ф.М. | 7202.03

**Группировка данных по нескольким столбцам**

В разделе **GROUP BY** можно указывать несколько столбцов, разделяя их запятыми. Тогда к одной группе будут относиться записи, у которых значения столбцов, входящих в группу, равны (т.е. одинаковые значения). Рассмотрим группировку по нескольким столбцам на примере следующего запроса:

**SELECT name, number\_plate, violation, count(\*)**

**FROM fine**

**GROUP BY name, number\_plate, violation;**

name | number\_plate | violation | count(\*)

| Баранов П.Е. | Р523ВТ | Превышение скорости(от 40 до 60) | 2

| Абрамова К.А. | О111АВ | Проезд на запрещающий сигнал | 2

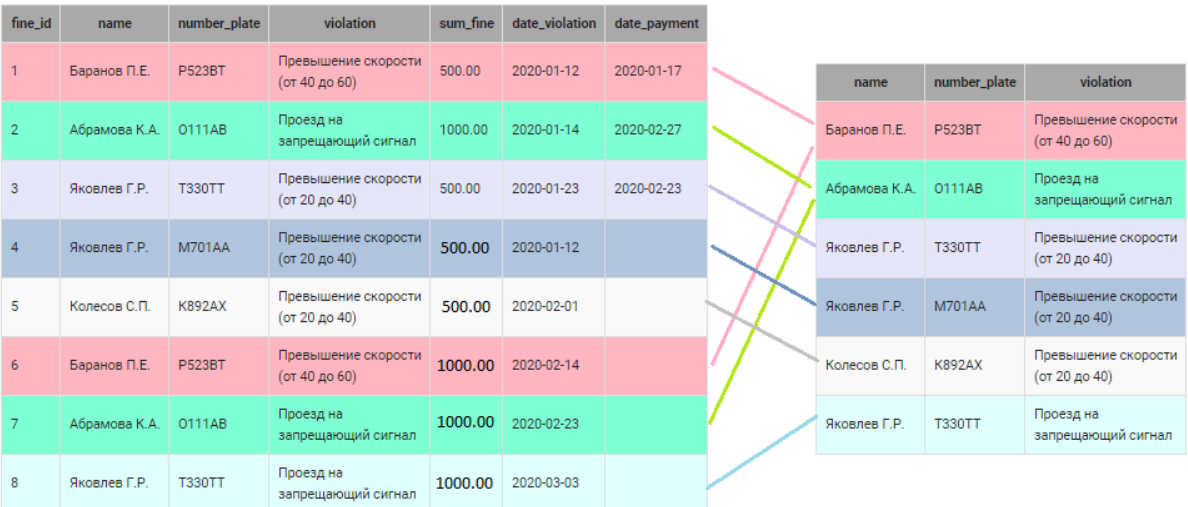
| Яковлев Г.Р. | Т330ТТ | Превышение скорости(от 20 до 40) | 1

| Яковлев Г.Р. | М701АА | Превышение скорости(от 20 до 40) | 1

| Колесов С.П. | К892АХ | Превышение скорости(от 20 до 40) | 1

| Яковлев Г.Р. | Т330ТТ | Проезд на запрещающий сигнал | 1

1. Сначала записи таблицы fine разделяются на группы. В каждую группу включаются строки, у которых равны значения в столбцах name, number\_plate и violation соответственно. Получается 6 групп.



2. Затем вычисляется функция count(\*), которая определяет количество записей в каждой группе. Получается, что к первым двум группам относятся по две записи, ко всем остальным - по одной.

Пример: Вывести фамилию, номер машины и нарушение только для тех водителей, которые на одной машине нарушили одно и то же правило два и более раз. При этом учитывать все нарушения, независимо от того оплачены они или нет. Информацию отсортировать в алфавитном порядке, сначала по фамилии водителя, потом по номеру машины и, наконец, по нарушению.

**SELECT name, number\_plate, violation**

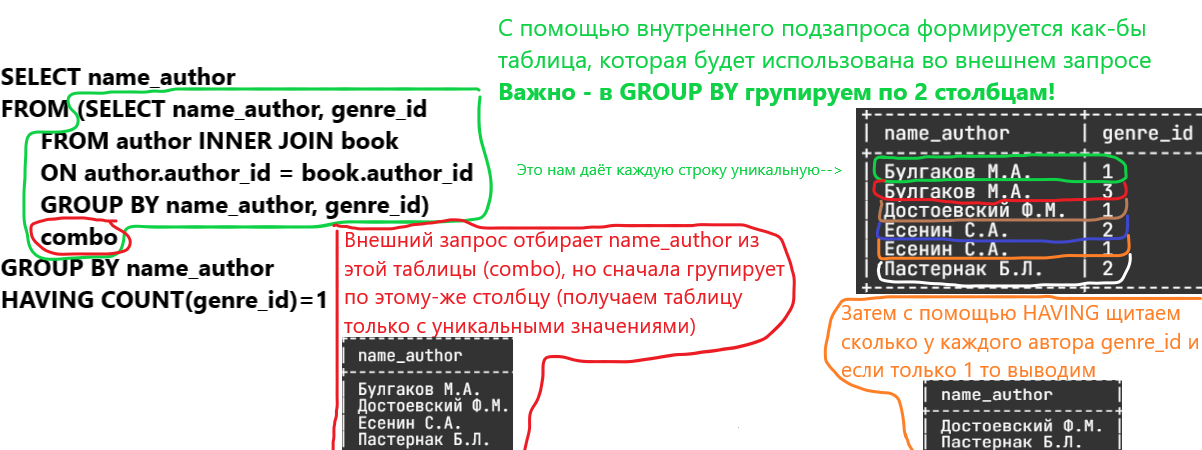
**FROM fine**

**GROUP BY name, number\_plate, violation** ← группировка сразу по 3 столбцам

**HAVING COUNT(number\_plate)>1 ←** having можно только после GROUP BY

**ORDER BY name, number\_plate, violation**

Пример: Вывести в алфавитном порядке всех авторов, которые пишут только в одном жанре.



Из интересного:

- Есть вложенный подзапрос у которого есть JOIN

- у вложенного подзапроса есть имя ‘combo’ это важно, иначе FROM не сработает.

- в вложенном подзапросе группируем по 2 столбцам