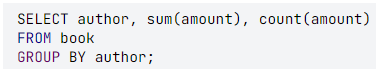
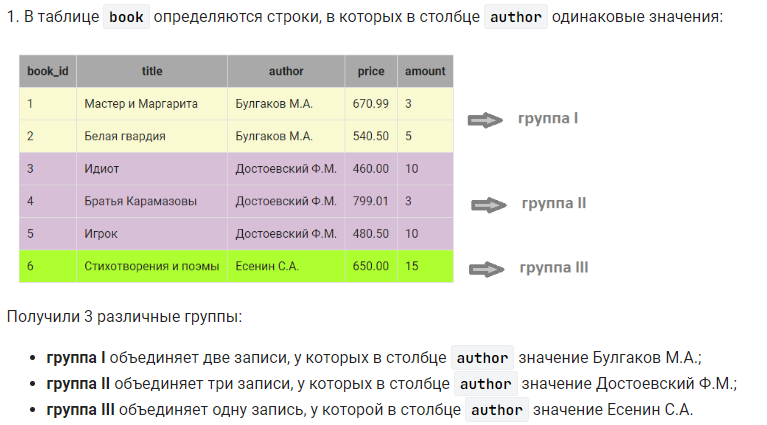
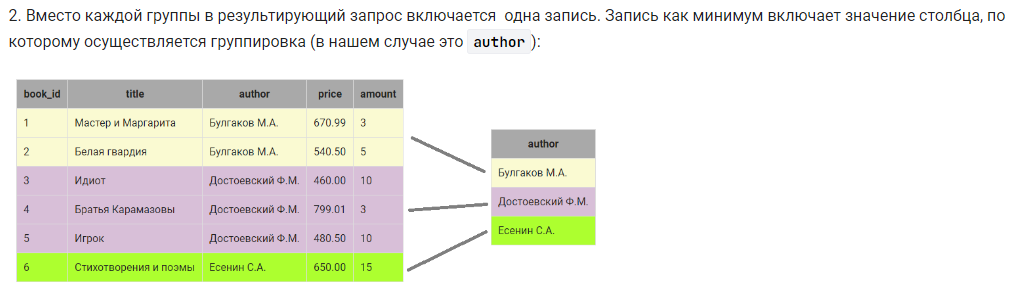
**GROUP BY**

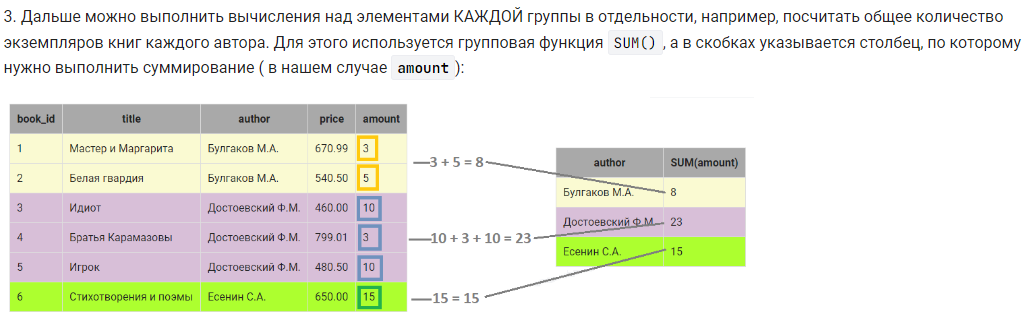
По сути “схлопывает” одинаковые атрибуты. Например если в таблице есть колонка author, и в ней 2 автора, но они дублируются много раз, то GROUP BY author вернет всего 2 уникальных автора, а другие колонки можно будет проссумировать или выполнить с ними другие операции

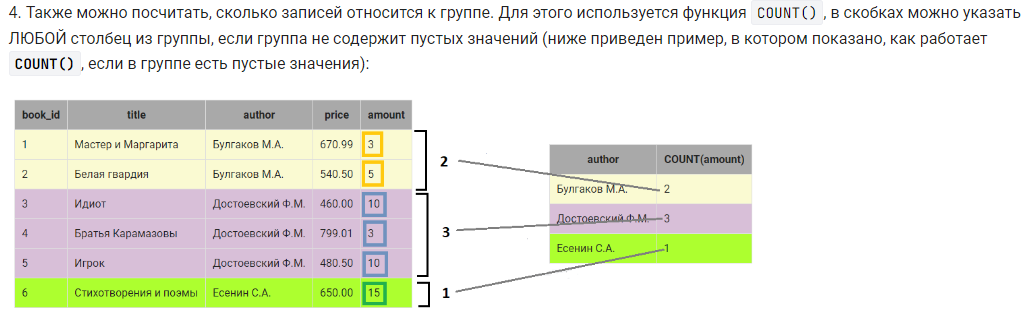
Подробно рассмотрим, как осуществляется группировка данных по некоторому столбцу и вычисления над группой на следующем примере:











**Мы обязательно должны выводить в SELECT такие же колонки, которые группируем в GROUP BY, исключение составляют агрегатные функции – COUNT, MIN, MAX, SUM, AVG.**

* **Те колонки которые мы выводим, но они не учавствуют в группировке, ОБЯЗАТЕЛЬНО должны иметь какую-либо функцию (это логично, так как если сгрупируются другие колонки, то кол-во значений другой колонки будет так же различно, и не понятно будет что выводить, если не использовать функцию) – выведится в таком случае рандомное значение из этого поля**
* **нельзя агрегировать ту же колонку, по которой группируете - есть что-то есть в агрегации, то в GROUP BY этих столбцов быть не должно**

КАК НЕЛЬЗЯ ДЕЛАТЬ – здесь мы НЕ группируем по Price, а должны, так как к ней не применена никакая функция. И вторая ошибка, мы группируем по той колонке, которую агрегируем

SELECT Price, COUNT(ProductID)

FROM Products

GROUP BY ProductID

Этот вариан верный, хотя мы и группируем по колонке, которую агрегируем, НО мы ее так же выводим отдельно – это верно

SELECT Price, COUNT(Price)

FROM Products

GROUP BY Price

- Группирует по столбцу. (одинаковые значения группируются). Возвращает количество товаров по одной цене.

SELECT Price, COUNT(ProductID)

FROM Products

GROUP BY Price

- Можно так же группировать по номеру

SELECT Price, COUNT(ProductID)

FROM Products

GROUP BY 1

- Можно так же группировать те колонки, которые мы не выводим

SELECT Price, COUNT(ProductID)

FROM Products

GROUP BY Price, Customer\_id

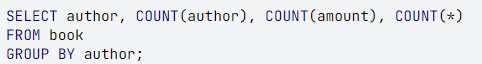
- Группирует по цвету волос только женских персонажей

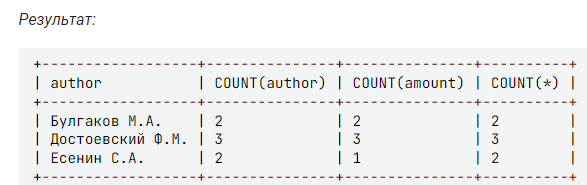
SELECT hair, COUNT(\*)

FROM superheroes

WHERE gender='Female Characters'

GROUP BY hair





COUNT(author) – если применяем функцию к конкретному стобцу, то он “схлопывает” каждое уникальное значение и для этого уникального значения считает то поле, которое мы указали. В данном случае это author.

COUNT(amount) – кол-во строк с кол-вом книг для каждого автора. У Есенина 2 книги, но кол-во проставлено только для 1 поля, а второе поле – NULL, по этому оно не считается

- **Группировка по 2 столбцам**. Группирует вначале компании производители комиксов (Marvel или DC), а после группирует по их отношению к добру или злу

SELECT universe, align, COUNT(\*)

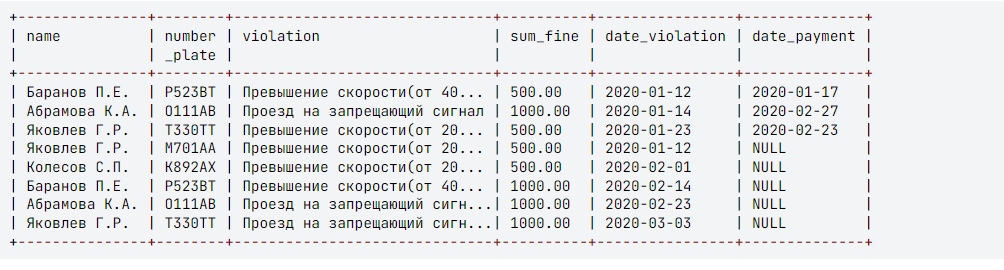
FROM superheroes

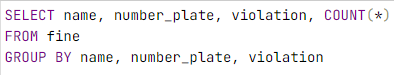
GROUP BY universe, align

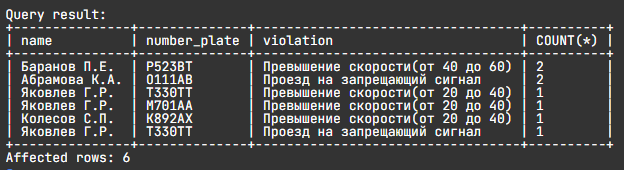
**Если выполнить группировку только по universe, то “схлопнутся все уникальные вселенные” (DC, MARVEL) и на выходе будет всего 2 строки, align же возьмется рандомный, так как он не попадает под группировку, а значений в этом столбце много. Но зачастую выводится ошибка, говорящая о том что нужно сгруппировать колонку**

**Если выполнить группировку по universe и align, то будет соотношение каждой уникальной вселенной к каждому уникальному align.**

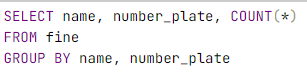
Есть следующая таблица

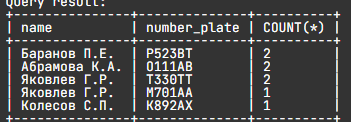
****

****

****

Видно, что для Яковлев Г.Р. для номера Т330ТТ есть 2 уникальных нарушения, из за этого это 2 разные записи. И еще есть уникальный номер М701АА. Тем самым у Яковлева 3 уникальных записи



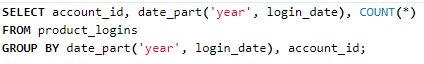
****

Если не группировать по нарушению, то у Яковлева 2 записи для Т330ТТ и 1 запись для М701АА. 2 записи для Т330ТТ это как раз разные нарушения, но из этой выборки этого не видно, так как мы не группируем по нарушениям (таблица соответственно не “экспандится” что бы в нее влезло больше данных)

**How many accounts have performed a login in the year 2016?**



Мы группируем по дате, что бы найти логин ВСЕХ аккаунтов. Если мы сгруппируем по дате и аккаунту, то будет виден логин для каждой даты И каждого аккаунта – по сути сколько раз в день логинился КАЖДЫЙ юзер



**ВЫРАЖЕНИЯ С АГРЕГАТНЫМИ ФУНКЦИЯМИ**

- Выводит общее количество появлений каждого типа персонажей. Если применять SUM без GROUP BY, то проссумируются все строки. А используя группировку – суммируется по группам

SELECT align, COUNT(\*), SUM(appearances)

FROM superheroes

GROUP BY align

- Выводит сколько раз появлялись наименее популярные и наиболее популярные супергерои за каждый год. Опять же, если не использовать GROUP BY, то выведется сумма всех строк

SELECT year, MIN(appearances), MAX(appearances)

FROM superheroes

GROUP BY year

- Проекты с одинаковыми именами группирует и выводит их суммарны бюджет

SELECT Project\_name, SUM(Budget), COUNT(\*) AS ‘Total’

FROM project

GROUP BY Project\_name

HAVING TOTAL > 1

**HAVING**

- HAVING это WHERE, но только для агрегатных функций. Нужен что бы задавать условия для агрегатных функций

SELECT hair, COUNT(\*) AS Total

FROM superheroes

WHERE gender = 'Female Characters'

GROUP BY hair

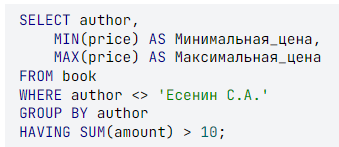
HAVING COUNT(\*) > 10

Заметим, что в предложении HAVING нельзя использовать псевдоним (Total ), используемый для именования значений агрегатной функции в предложении SELECT. Дело в том, что предложение SELECT, формирующее выходной набор запроса, выполняется предпоследним перед предложением ORDER BY. Ниже приведен порядок обработки предложений в операторе SELECT:

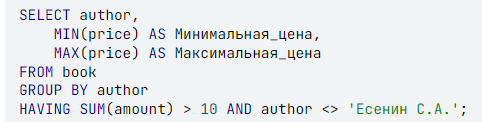
1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

Этот порядок не соответствует синтаксическому порядку общего представления оператора SELECT, который ближе к естественному языку:

Вывести максимальную и минимальную цену книг каждого автора, кроме Есенина, количество экземпляров книг которого больше 10.



Другим способом решения примера является запрос:



Не смотря на то что результат будет одинаковым, так делать **не рекомендуется**. «Потому что как написано - запрос сначала выбирает всех авторов, потом выводит данные, рассчитывая минимальное и максимальное значение цены для каждого, и только после всего убирает Есенина. Можно убрать Есенина в данном случае раньше и не использовать ресурсы базы для расчета его минимального и максимального значения, как это сделано в первом варианте. На небольшой базе быстродействия не ощутить, но если выполнять такое на продуктивной, то второй вариант значительно проигрывает

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.

