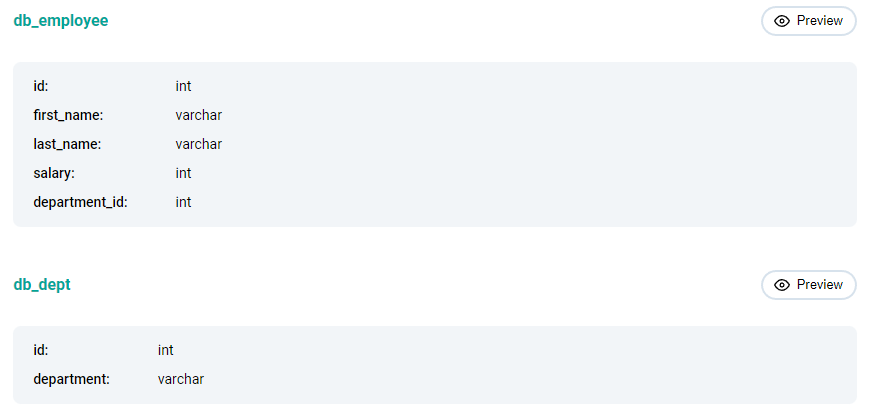
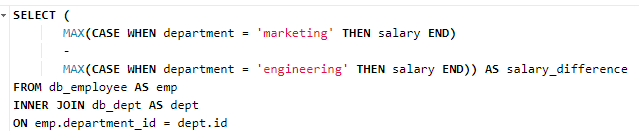
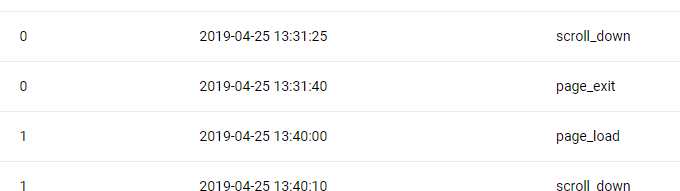
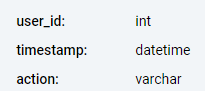
**Если нужно найти разницу, то можем использовать CASE**

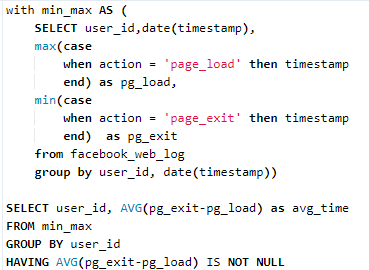
**Write a query that calculates the difference between the highest salaries found in the marketing and engineering departments. Output just the absolute difference in salaries.**



****

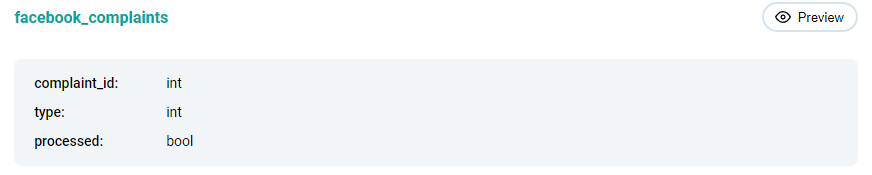
**Calculate each user's average session time. A session is defined as the time difference between a page\_load and page\_exit. For simplicity, assume a user has only 1 session per day and if there are multiple of the same events on that day, consider only the latest page\_load and earliest page\_exit. Output the user\_id and their average session time.**

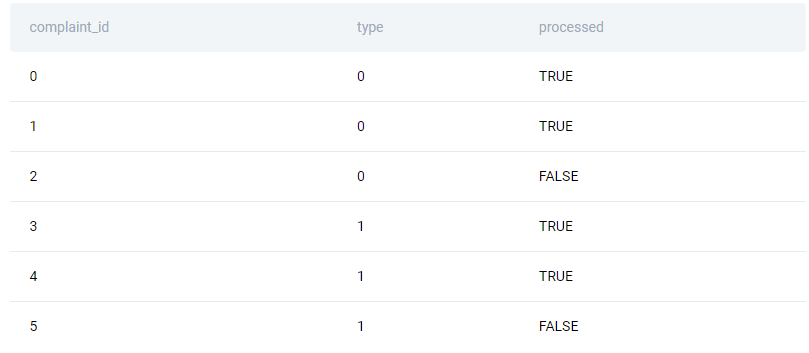


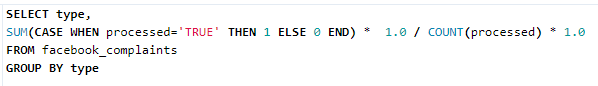


1. Выбираем Максимальное время для каждого юзера, когда страница загружена, если action не равен загруженной странице, то пропускаем
2. Выбираем Минимальное время для каждого юзера, когда страница загружена, если action не равен загруженной странице, то пропускаем
3. Находим разницу для каждого юзера и выводим среднее

**Find the rate of processed tickets for each type. – Если нужно что то посчитать, то можем сделать SUM, и привести с помощью CASE значение в поле к 1 или 0**



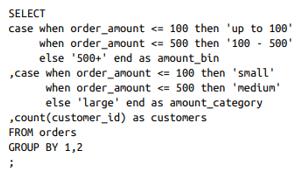




1. Преобразуем булевое TRUE в 1, и FALSE в 0. Переводим это в float тип
2. Делим общее кол-во TRUE на общее кол-во строк
3. Группируем по типу

**Если нужно сгруппировать данные**

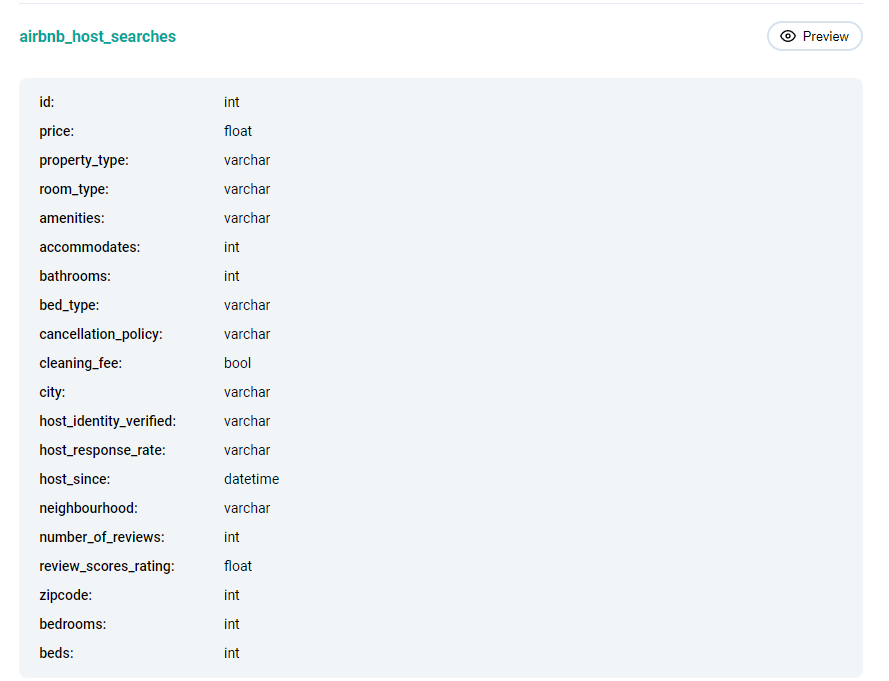
**As an example, imagine we are considering discounted shipping offers and we want to know how many customers will be affected. We can group order\_amount into three buckets using a CASE statement:**

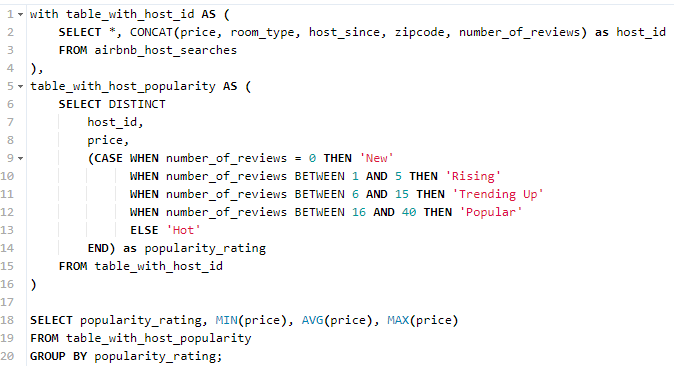
****

Тем самым мы создадим отдельные группы по которым после будет группировать

**You’re given a table of rental property searches by users. The table consists of search results and outputs host information for searchers. Find the minimum, average, maximum rental prices for each host’s popularity rating. The host’s popularity rating is defined as below: 0 reviews: New 1 to 5 reviews: Rising 6 to 15 reviews: Trending Up 16 to 40 reviews: Popular more than 40 reviews: Hot**

**Tip: The id column in the table refers to the search ID. You'll need to create your own host\_id by concating price, room\_type, host\_since, zipcode, and number\_of\_reviews.**

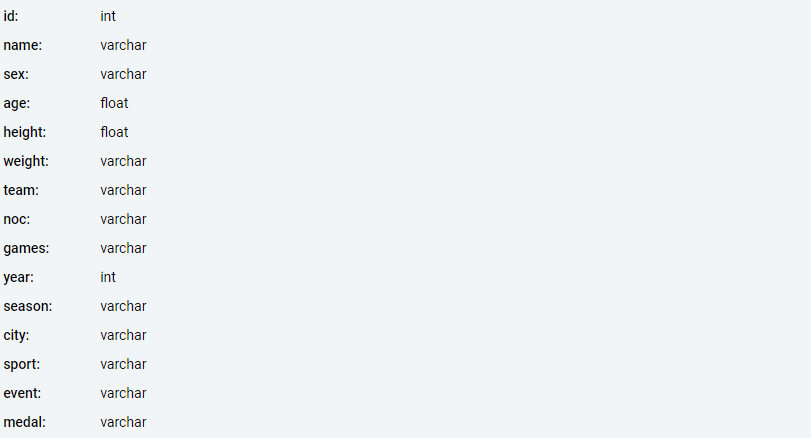


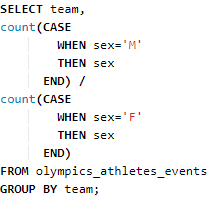


1. Сами создаем host\_id
2. Создаем группы для кол-ва отзывов. Так же оставляем только уникальные значения. Так как изначальная таблица – логи поиска, то один и тот же host\_id может повторяться много раз
3. Находим min, avg, max для каждой группы

**Если нужно посчитать только определенные данные**

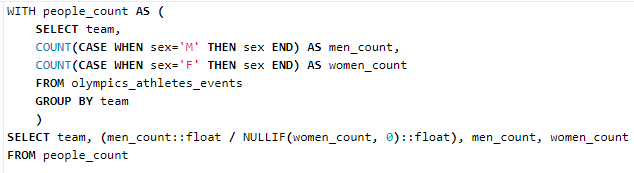
Допустим, нам нужно посчитать соотношение кол-ва спортсменов мужчин к женщинам в каждой из команд

****

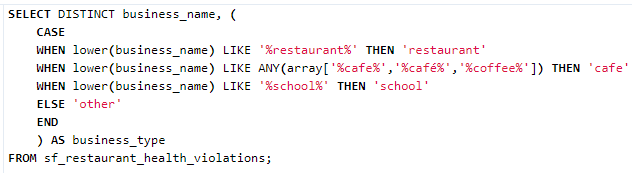
****

Но такой запрос выведет ошибку, так как женщин может быть 0, а делить на 0 нельзя. Для этого используем **NULLIF(column\_name, 0)**

Если так же добавить в выборку отдельно поля как “кол-во мужчин” и “кол-во женщин”, то запрос можно переписать вот так.

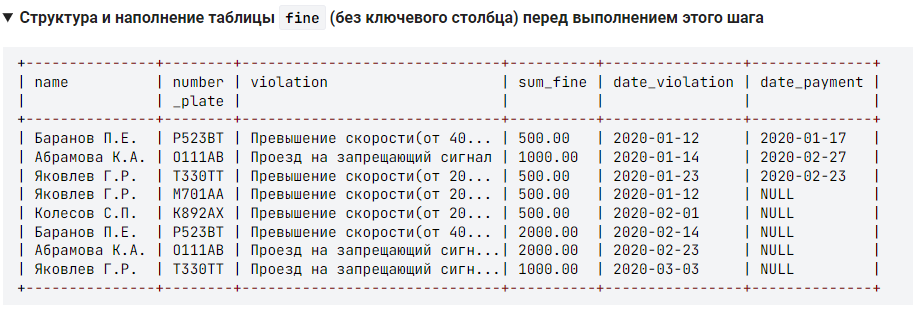
****

**Classify each business as either a restaurant, cafe, school, or other. A restaurant should have the word 'restaurant' in the business name. For cafes, either 'cafe', 'café', or 'coffee' can be in the business name. 'School' should be in the business name for schools. All other businesses should be classified as 'other'. Output the business name and the calculated classification.**



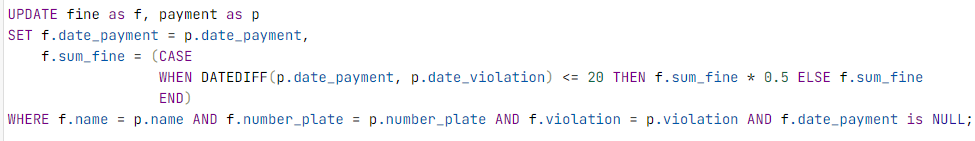
1. DISTINCT так как нам не нужны дубликаты
2. Не нужно писать …THEN business\_name = ‘restaurant’, так как это операция сравнения и она выведет TRUE / FALSE, здесь мы просто явно указываем что должно быть выведено





Необходимо:

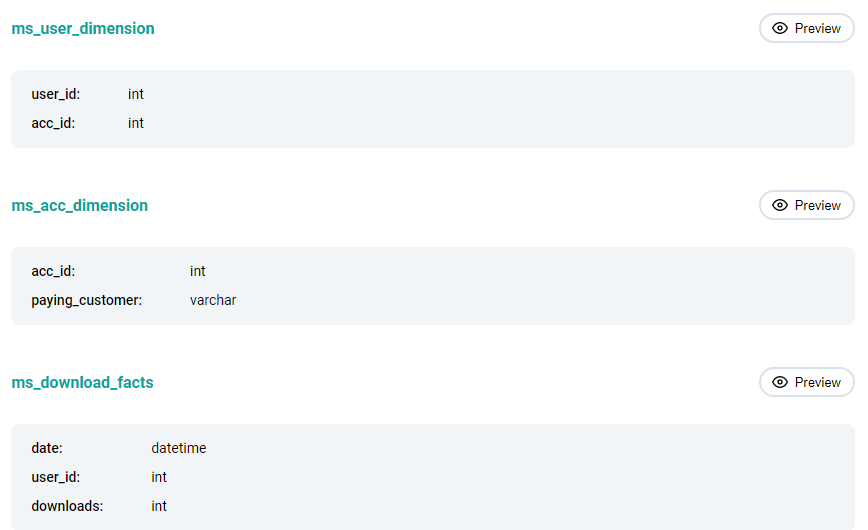
* в таблицу **fine**занести дату оплаты соответствующего штрафа из таблицы **payment;**
* уменьшить начисленный штраф в таблице **fine** в два раза  (только для тех штрафов, информация о которых занесена в таблицу **payment**) , если оплата произведена не позднее 20 дней со дня нарушения.

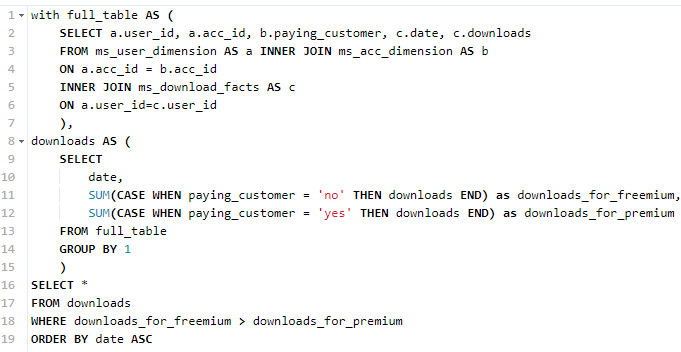


* 1. Выбираем 2 таблицы и соединяем их в WHERE по нужным значениям. Так же добавляем фильтрацию что проставить только там где date\_payment = Null
  2. Для date\_payment в таблице fine устанавливаем дату как в таблице payment
  3. Сумму штрафа вычисляем так – разница между датой платежей должна быть менее или равно 20 дней. В этом случае уменьшить ее в 2 раза. Если условие другое, то оставить такой же. Если не указать условие ELSE, то получится **f.sum\_fine = Null**, а нам это не подходит.

**Find the total number of downloads for paying and non-paying users by date. Include only records where non-paying customers have more downloads than paying customers. The output should be sorted by earliest date first and contain 3 columns date, non-paying downloads, paying downloads.**

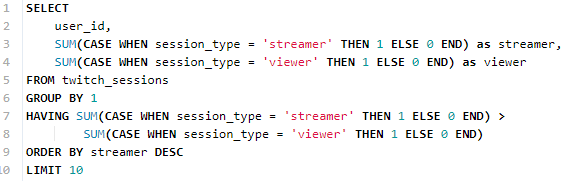
1. Find the total number of downloads for paying and non-paying users by date
2. Include only records where non-paying customers have more downloads than paying customers
3. The output should be sorted by earliest date first and contain 3 columns date, non-paying downloads, paying downloads





1. Объединяем все таблицы
2. Считаем сумму загрузок для премиум и обычных пользователей для каждого дня
3. Фильтурем даты

**List the top 10 users who accumulated the most sessions where they had more streaming sessions than viewing. Return the user\_id, number of streaming sessions, and number of viewing sessions**



1. Считаем кол-во streamer и viewer для каждого пользователя.
2. Фильтруем те значения где кол-во streamer сессий > viewer сессий