**Порядок анализа задачи**

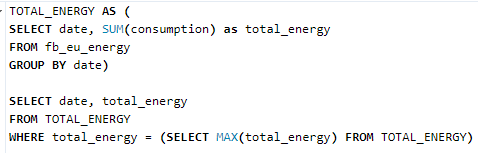
1. Понять какие шаблоны здесь применими (найти максимальное, произвести выборку ссылаясь на другие значения, ...)
2. Понять из условия задачи нужно ли что то выводить для группы полей. Например если просят вывести кол-во сообщений которые отправил гость, то группируем по гостю и выводим SUM()
3. Разбиваем задачу на подблоки с помощью with

По сути все сводится к следующему (не считая базовых конструкций по типу фильтрации в WHERE по полю, SELECT значение, ORDER BY, HAVING, LIMIT)

1. Подзапросы в
   1. SELECT
   2. FROM / или табличным выражениям заместо подзапроса в FROM
   3. WHERE
2. GROUP BY + агрегатные функции
3. JOIN
4. Оконные Функции
5. CASE

Если для решения задачи есть несколько шаблонов, то тогда применять отдельно каждый. Например есть таблица в которой содержаться даты и число. Задача – найти общую сумму чисел и вывести даты с самым максимальным числом. Здесь 2 шаблона

1. Группировка по дате и поиск суммы чисел для каждой даты
2. Через подзапрос выбор MAX() в условии WHERE

****

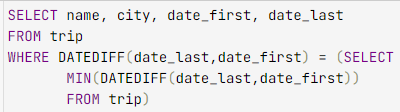
**ПОДЗАПРОСЫ**

**Если нужно найти что то с максимальным/минимальным значнеием, то используем подзапрос MAX()/MIN()**

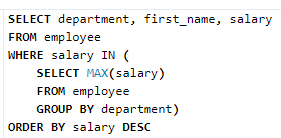
**Find the hour with the highest gasoline cost. Assume there's only 1 hour with the highest gas cost.**

****

Вывести информацию о командировках сотрудника(ов), которые были самыми короткими по времени.

****

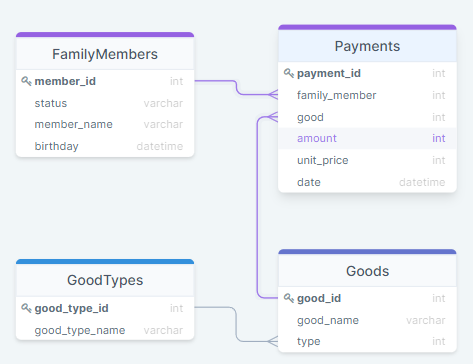
**Find the employee with the highest salary per department. Output the department name, employee's first name along with the corresponding salary.**

****

Используем IN, так как вернется не одно значение (из за того что идет группировка по department, и для каждого department выведится максимальное число по зарплате)

**Если нужно выбрать поля, исключив из выборки другие, пишем в условии WHERE подзапрос того что исключаем**

**Определить, какие товары имеются в таблице Goods, но не покупались в течение 2005 года**



SELECT good\_name

FROM Goods

WHERE good\_id NOT IN (SELECT good FROM Payments

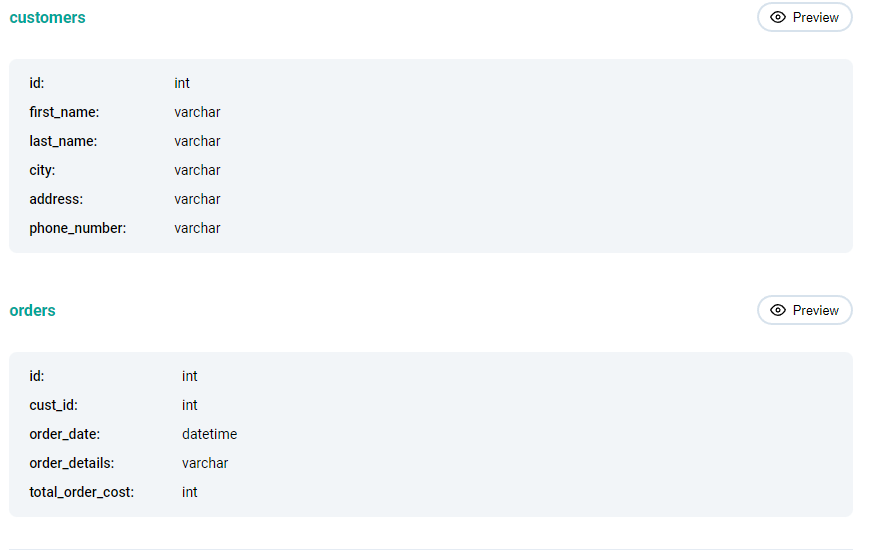
WHERE YEAR(date) = 2005);

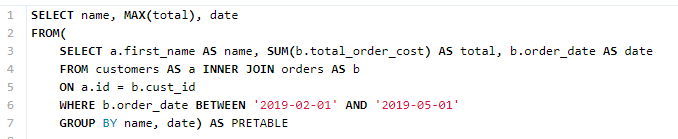
Дело в том, что не все продукты есть в Payments. А нам надо найти продукты которые не купили в 2005. Если фильтровать только по таблице Payments, то будут выбраны значения которые не купили в 2005, но купили в другой год. А таким образом выводятся так же и те значения которые вообще никогда не купили

**Если нужно найти что то с максимальным/минимальным значнеием, но для этого заранее нужно посчитать те самые значения из которых будем выбирать – используя подзапрос**

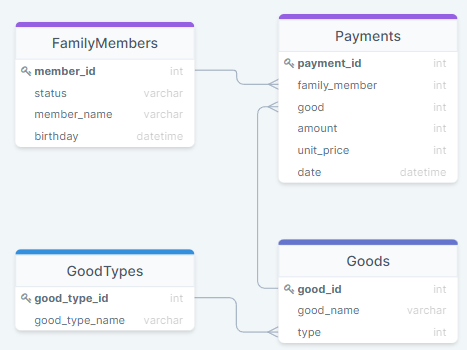
**Когда мы используем подзапрос в FROM, то возвращаемая таблица должна иметь alias**

Find the customer with the highest daily total order cost between 2019-02-01 to 2019-05-01. If customer had more than one order on a certain day, sum the order costs on daily basis. Output customer's first name, total cost of their items, and the date.





**Найдите самый дорогой деликатес (delicacies) и выведите его стоимость**



SELECT g.good\_name, p.unit\_price

FROM Goods AS g

JOIN Payments AS p ON (g.good\_id=p.good)

JOIN GoodTypes AS gt ON (g.type=gt.good\_type\_id)

WHERE p.unit\_price = (SELECT **MAX**(p.unit\_price)

FROM Payments AS p

JOIN Goods AS g ON (g.good\_id=p.good)

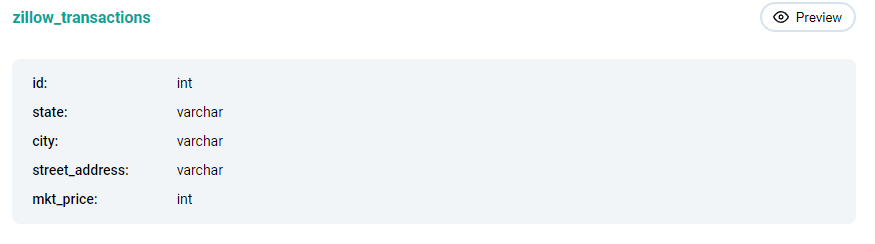
JOIN GoodTypes AS gt ON (g.type=gt.good\_type\_id)

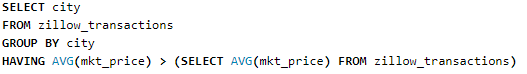
WHERE gt.good\_type\_name='delicacies');

В условии WHERE мы фильтруем максимальный ценник только по деликатесам. Так как это вложенный запрос, то мы должны опять объединить все таблицы. Если же написать:

WHERE gt.good\_type\_name='delicacies' AND p.unit\_price = (SELECT **MAX**(p.unit\_price) FROM Payments) – то выдаст ошибку, так как будет браться максимальный ценник по изначальной таблице, а не только по деликатесам

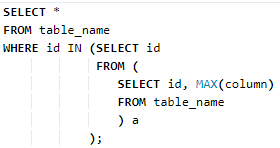
**Write a query that identifies cities with higher than average home prices when compared to the national average. Output the city names.**

****

****

1. Выводим город, когда средняя цена для города > средняя цена для нации (нация это средняя цена по всей таблице)

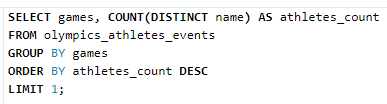
**Если нужно отфильтровать по условию WHERE, но возвращаемая таблица имеет несколько значений**

****

**GROUP BY - ИСПОЛЬЗУЕМ GROUP BY, ЕСЛИ НУЖНО НАЙТИ ЧТО ТО ДЛЯ КАЖДОЙ ГРУППЫ (совокупность уникальных значений в столбце)**

**Если нужно посчитать кол-во уникальных значений**

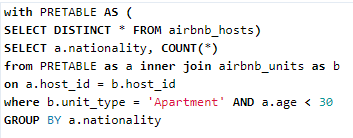
1. Find the Olympics with the highest number of athletes. The Olympics game is a combination of the year and the season, and is found in the 'games' column. Output the Olympics along with the corresponding number of athletes.



1. Find the number of apartments per nationality that are owned by people under 30 years old. Output the nationality along with the number of apartments. Sort records by the apartments count in descending order.

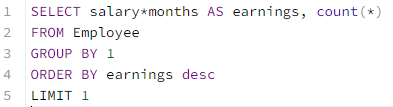
В таблице airbnb\_hosts содержатся дубликаты

****

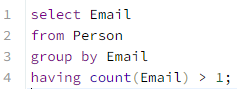
****

Если нужно выбрать **общее кол-во чего либо**, то используем **COUNT(\*)** , совместно с **GROUP BY** той колонки, кол-во которой нужно посчитать

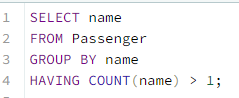
**Выбрать кол-во людей с наибольшей зарплатой**



**Найти дубликаты**

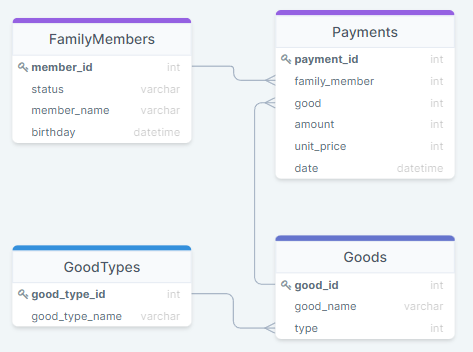


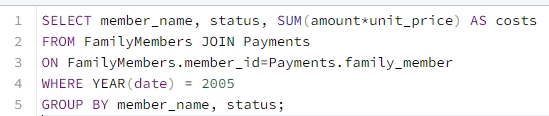
**Вывести имена людей, у которых есть полный тёзка среди пассажиров**



Если нужно выбрать **общую сумму чего либо**, то используем **SUM(\*)** , совместно с **GROUP BY** той колонки, сумму которой нужно посчитать

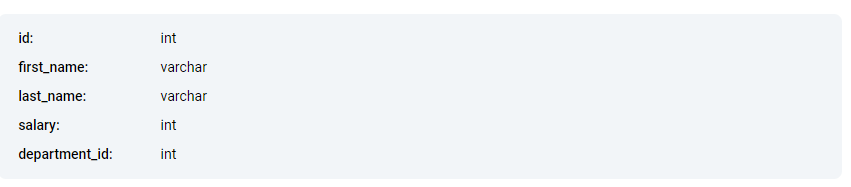
**Определить, сколько потратил в 2005 году каждый из членов семьи**

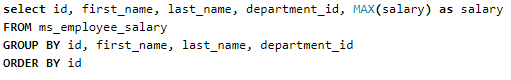




**Если нужно найти максимальное значение для каждой группы**

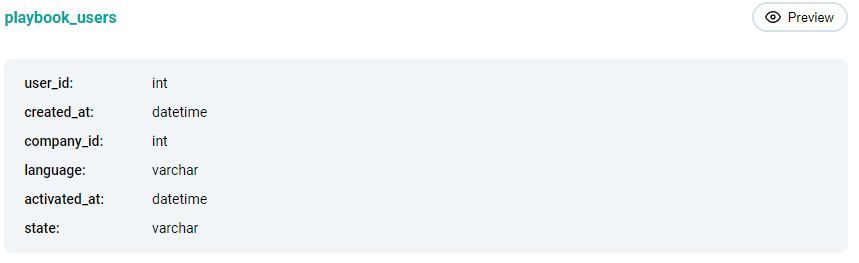
We have a table with employees and their salaries, however, some of the records are old and contain outdated salary information. Find the current salary of each employee assuming that salaries increase each year. Output their id, first name, last name, department ID, and current salary. Order your list by employee ID in ascending order.

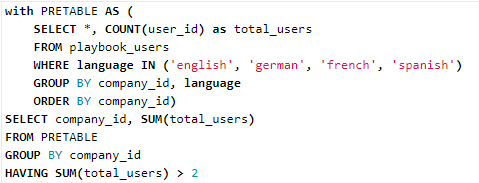




Если нужно найти сумму чего то, когда есть условие найти для **n**, когда **m**

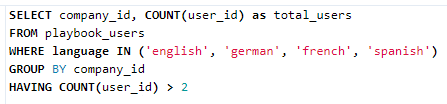
Find ids of **companies** that have more than **2 users who speak** English, German, French, or Spanish.





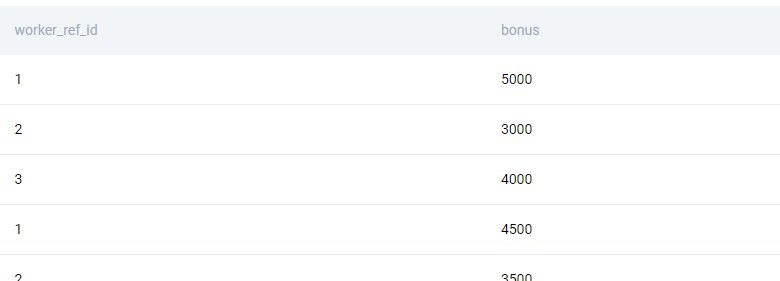
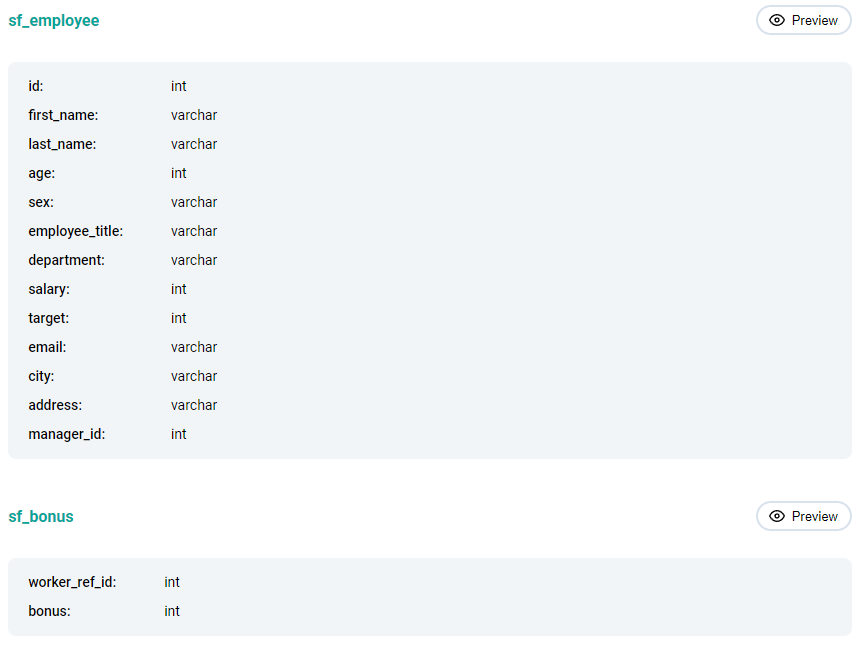
1. Добавим фильтрацию WHERE
2. "схлопнуть" (сгруппировать) все компании
3. Расширить "схлопывание" группировкой так же по языку - тем самым будет список компаний и всех языков
4. Посчитаем кол-во юзеров для каждогй компании и языка на котором они говорят COUNT(user\_id)
5. Из полученной таблицы сгруппировать по компаниям и проссумировать кол-во языков
6. Отфильтруем только те компании где кол-во юзеров более 2 с

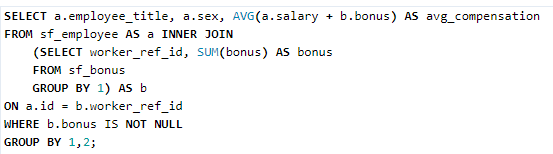
Второй вариант – более простой и лучший. В варианте выше работает кейс, когда мы сгруппировали по 2 колонкам и более (company\_id, language), то имеем кол-во юзеров для каждой компании и языка. И далее мы опять группируем по компании, что бы найти кол-во всех сотрудников. Группировка по языку в данном случае лишняя, по этому можно обойтись и без нее, сгруппировав только по компании



1. Добавим фильтрацию WHERE
2. "схлопнуть" (сгруппировать) все компании
3. Посчитаем кол-во юзеров для каждогй компании
4. Отфильтруем только те компании где кол-во юзеров более 2 с

**Find the average total compensation based on employee titles and gender. Total compensation is calculated by adding both the salary and bonus of each employee. However, not every employee receives a bonus so disregard employees without bonuses in your calculation. Employee can receive more than one bonus. Output the employee title, gender (i.e., sex), along with the average total compensation.**

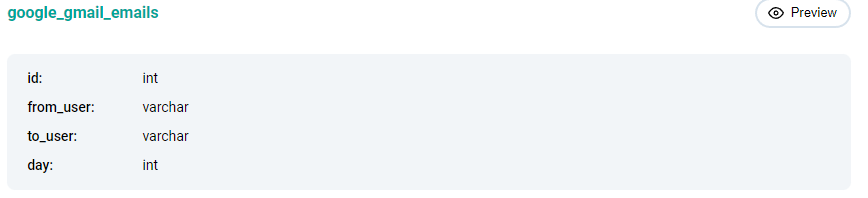


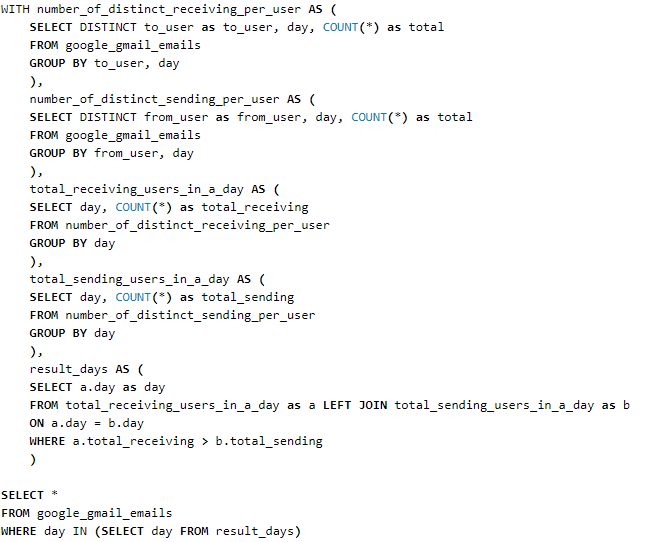


В таблице sf\_bonus – бонусы для каждого сотрудника встречаются несколько раз, по этому их нужно так же сгруппировать и найти их сумму

**Порой нужно в начале посчитать общее кол-во чего то для каждой сгруппированной колонки, а потом уже общее кол-во**

Find all records from days when the number of distinct users receiving emails was greater than the number of distinct users sending emails



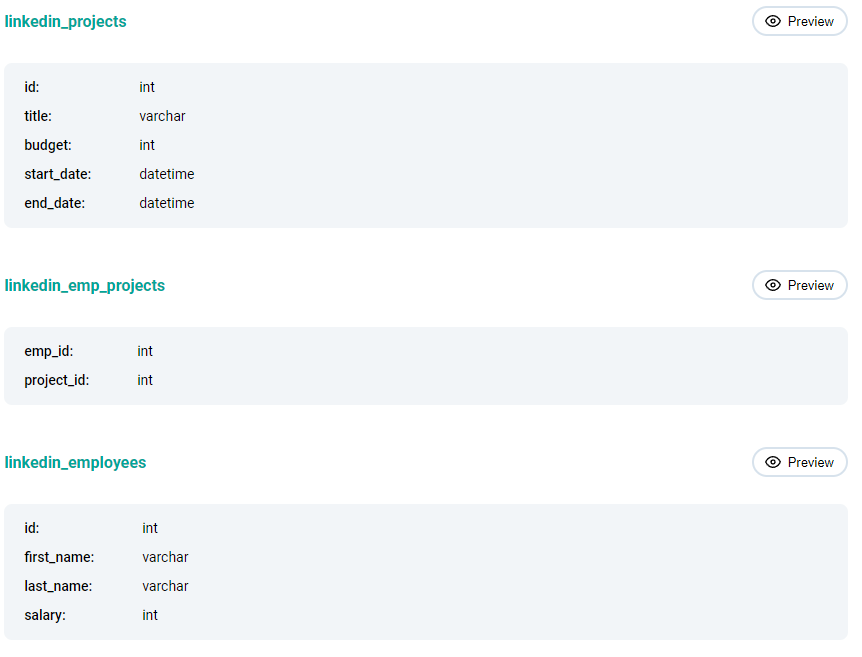


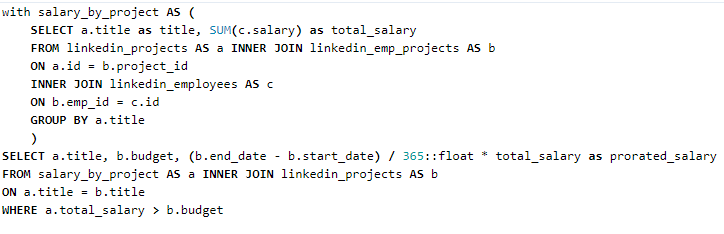
1. В начале находим сколько раз каждый юзер в день получал сообщения и сколько раз каждый юзер в день отправлял сообщения
2. Потом находим общее кол-во сообщений в день для тех кто отправлял и получа лсообщения
3. После используя JOIN находим дни когда получали сообщения больше чем отправляли

**Identify projects that are at risk for going overbudget. A project is considered to be overbudget if the cost of all employees assigned to the project is greater than the budget of the project.**

**You'll need to prorate the cost of the employees to the duration of the project. For example, if the budget for a project that takes half a year to complete is $10K, then the total half-year salary of all employees assigned to the project should not exceed $10K. Salary is defined on a yearly basis, so be careful how to calculate salaries for the projects that last less or more than one year.**

**Output a list of projects that are overbudget with their project name, project budget, and prorated total employee expense (rounded to the next dollar amount).**





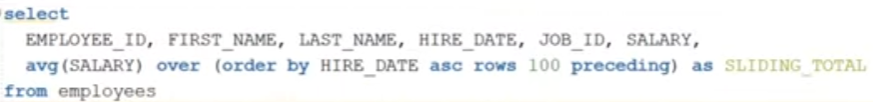
1. Находим общуу зарплату сотрудников по проекту с помощью with и группировки по проекту
2. Выбираем только те проекты где зарплата сотрудников больше чем бюджет при помощи WHERE
3. С помощью ::float можн оуказать какой тип должен получиться после деления

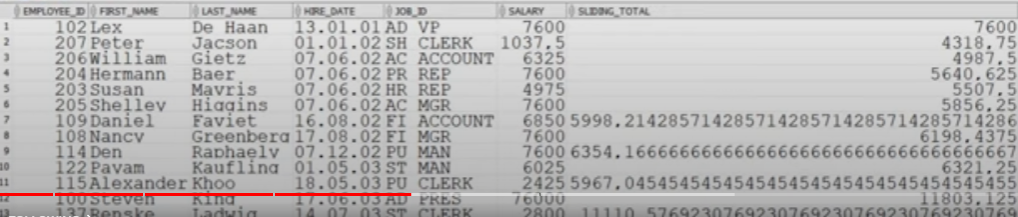
**Оконные функции**

**Если нужно посчитать что то (найти накопительную сумму/скользящее среднее/…) в течении какого то периода времени**

**SUM(**поле**)** **OVER(**[PARTITION BY поле] **ORDER BY поле** [ROWS/RANGE …]**)**

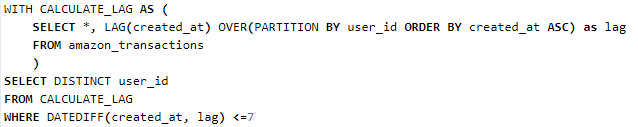
Допустим нужно найти зарплату сотрудников, и среднюю зарплату всех принятых сотрудников за последние 100 дней





**Если нужно забирать данные прошлой строки (например дату прошлой покупки относительно текущей)**

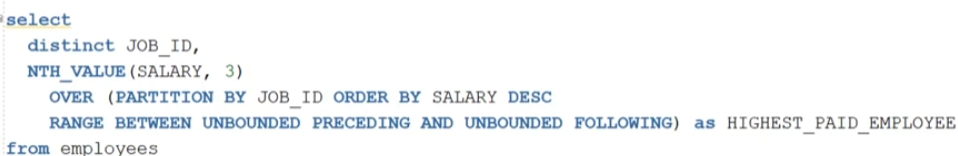
Write a query that'll identify returning active users. A returning active user is a user that has made a second purchase within 7 days of any other of their purchases. Output a list of user\_ids of these returning active users.



1. Создаем таблицу, где для каждой новой покупки будем рядом выводить дату последней покупки. Группируем по юзеру
2. Из полученной таблицы выбираем уникальных юзеров, где разница между текущей покупкой и прошлой меньше или равна 7

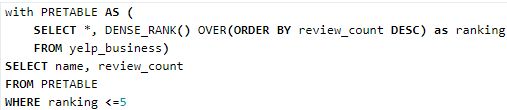
**Если нужно вывести проранжированную N строку**

Например нужно получить третью по велечине зарплату по каждому отделу



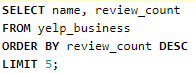
**Если нужно вывести ранк компании**

Find the top 5 businesses with most reviews. Assume that each row has a unique business\_id such that the total reviews for each business is listed on each row. Output the business name along with the total number of reviews and order your results by the total reviews in descending order.



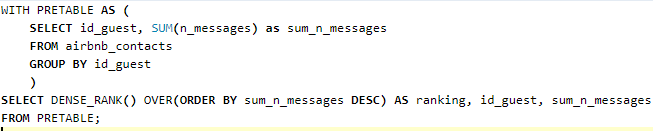
1. Создаем таблицу где всем строкам будет присвоен ранк, ранк сортируем от наибольшего к наименьшему, тем самым строки где ранка больше всего будут сверху (*нужно использовать табличное выражение или подзапрос, так как создается отдельная колонка – ranking, и where ее не видит сразу*?)
2. Выбираем из созданной таблицы нужные поля, и указываем условие WHERE

В принципе эту задачу можно решить и так



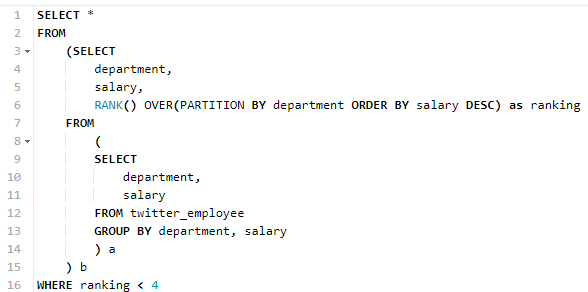
Rank guests based on the number of messages they've exchanged with the hosts. Guests with the same number of messages as other guests should have the same rank. Do not skip rankings if the preceding rankings are identical. Output the rank, guest id, and number of total messages they've sent. Order by the highest number of total messages first.





1. В начале “схлопываем” (группируем) гостей, и считаем сумму их сообщений
2. После высчитываем ранк этого гостя

**Find the top three distinct salaries for each department. Output the department name and the top 3 distinct salaries by each department. Order your results alphabetically by department and then by highest salary to lowest.**



В подзапроса **a** оставляем только уникальные строки

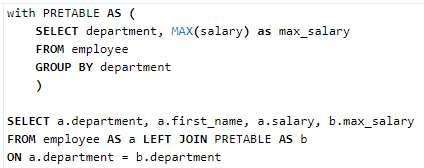
В подзапроса **b** ранжируем строки по зарплате для каждого департамента

В главном запросе фильтруем только топ 3

**Если нужно посчитать кол-во чего то для всех полей, не изменяя структуру полей (не уменьшая кол-во строк, а просто добавить новую колонку со значениями)**

Нужно посчитать максимальную зарплату для каждого отдела и рядом с зарплатой каждого сотрудника указать максимальную зарплату по отделу

**Решение через JOIN и Табличные выражения – не предпочтительно, лучше оконные функции**

****

* 1. Находим максимальную зарплату по отделу в табличном выражении
  2. Добавляем колонку с максимальной зарплатой левым джойном
  3. Связываем таблицы именно по department, так как для него вычислялась максимальная зарплата

**Решение через Оконные функции**

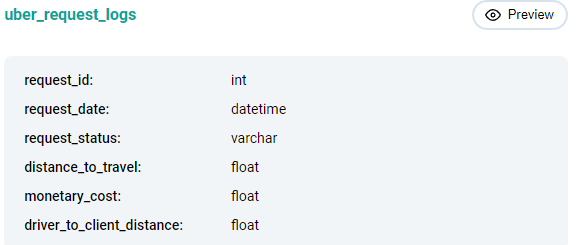


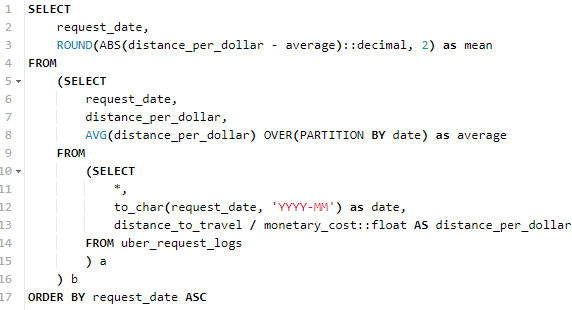
Если поменять условие и вывести не максимальную разплату по отделу, а допусти максимальную у сотрудника, и прилинковать именно ее, то запрос будет выглядеть так

**Решение через Оконные функции**



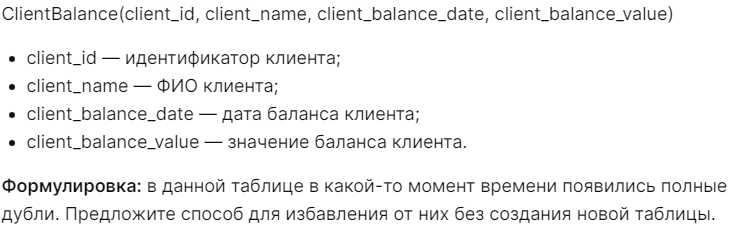
**Найти разницу между (отношением длины поездки к ее сумме за день) и (сумме среднего отношения длины поездки и ее сумме за месяц и год)**

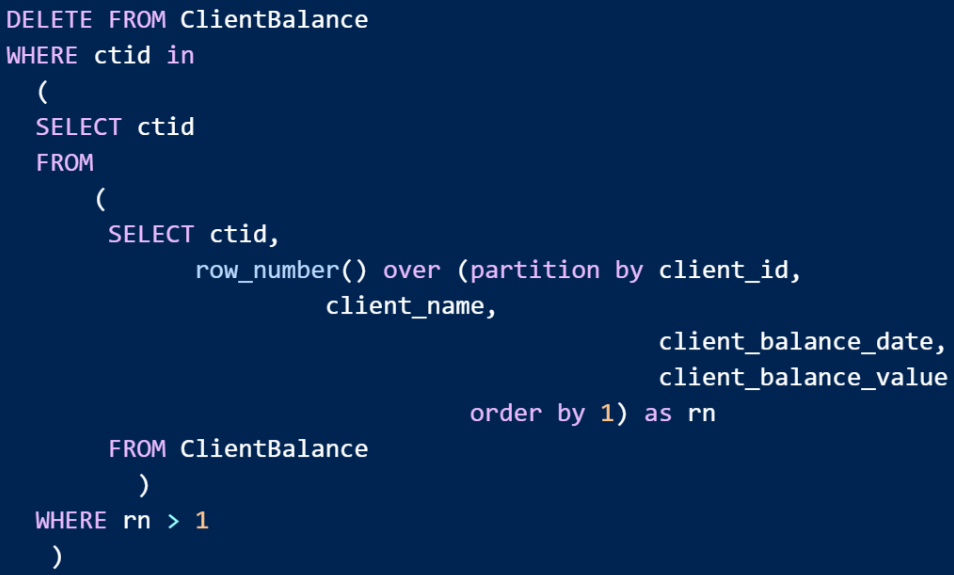
****

****

Используя AVG(..) OVER(PARTITION BY date) – мы находим среднее для каждой уникальной даты формата “Год-Месяц”, и если для каждой даты формата “Год-Месяц-День” мы указываем найденное среднее для этого месяца

**Если нужно найти дубликаты**

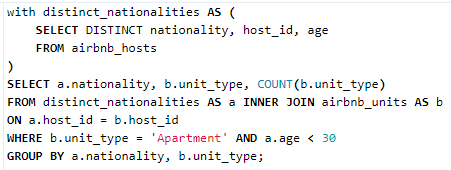
****

****

При выполнении оконной функции row\_number(), мы группируем все поля, тем самым для каждой строки будет проставляться номер от 1 до N кол-ва одинаковых строк

**DISTINCT**

**Find the number of apartments per nationality that are owned by people under 30 years old. Output the nationality along with the number of apartments. Sort records by the apartments count in descending order.**



1. **В начале** создаем таблицу, в которой будут только уникальные юзеры
2. После уже производим джойны и группировки

**Write a query that returns the number of unique users per client per month**

