**Задание 2**

Изучите данные и выполните следующие задачи. Выполните задание с помощью SQL.

*Решение оформите в одном Word-файл с текстом запросов на каждое задание.*

Клиенты оставляют заявки на звонок, после чего менеджеры звонят и обсуждают все необходимые вопросы.

Есть 2 таблицы с данными. В таблице tasks лежат заявки от клиентов, в таблице calls лежат звонки от менеджеров клиентам.

📁 **Таблица tasks**

* client\_id (идентификатор клиента)
* created\_datetime (время клиентской заявки)
* title (тема обращения)

📁 **Таблица calls**

* manager\_id (идентификатор менеджера)
* client\_id (идентификатор клиента, которому звонят)
* call\_datetime (время менеджерского звонка)

Обратите внимание, что в Metabase данных таблиц нет. Выдача заданий с “вымышленными” таблицами - это частая практика на собеседованиях и в тестовых заданиях от работодателей.

Чтобы протестировать свой код на корректную работу - можно создать таблицы с данными на [ресурсе](https://sqliteonline.com/) и отработать запросы.

Напишите запросы, которые выводят

**Задание 2.1**

Сколько заявок приходило каждый день в июне 2022 года?

select created\_datetime

, count(\*) as cnt

from tasks

where created\_datetime between '2022-06-01 00:00' and '2022-06-30 23:59'

group by created\_datetime

order by created\_datetime asc

Работаем с таблицей tasks. Прописываем условие, что нам нужны даты только за июнь 2022 года. На вывод делаем группировку по дате и считаем количество по любому из столбцов таблицы.

**Задание 2.2**

Список тем, для которых обращений было больше 10 в апреле 2022 года

select title

, count(title) as cnt

from tasks

where created\_datetime between '2022-04-01 00:00' and '2022-04-30 00:00'

group by title

having count(title) > 10

order by cnt asc

Снова работаем с таблицей tasks. Прописываем с помощью where условие на дату. Нас интересует апрель 2022 года. Просим вывести в итоговой таблице количество любого из столбца исходной таблицы (в данном случае запросили title) с группировкой по теме обращений. С помощью having ставим ограничение больше 10 обращений.

**Задание 2.3**

Список клиентов, которые оставляли заявку, но ни одного звонка от менеджера по ним не было.

select t.client\_id

from tasks as t

full join calls as c

on t.client\_id = c.client\_id

where c.client\_id is null

group by t.client\_id

order by t.client\_id asc

Здесь уже работаем с двумя таблицами. Соединяем их с помощью full join, так как нам нужны данные именно по нуловым строкам. Соединяем по client\_id. Ставим условие пустой строки в client\_id из таблицы calls. Делаем на вывод группировку по client\_id и сортируем по возрастанию.

**Задание 2.4**

Для каждого клиента выведите три его последних обращения и постройте распределение количества этих обращений по теме

select title

, count(rn) as cnt

from

(select client\_id

, created\_datetime

, title

, row\_number() over(partition by client\_id order by created\_datetime desc) as rn

from tasks

) as tasks\_rn

where rn <= 3

group by title

order by cnt asc

Работаем с таблицей tasks. Используем row number в рамках client\_id с сортировкой по created\_datetime. Получаем таблицу с колонками client\_id, created\_datetime, title, rn. В рамках каждого клиента каждому обращению присваивается порядковый номер, порядковый номер попросили присваивать по принципу: последнее обращение по времени получает наивысший ранг, соответственно первое – низший ранг. Присваиваем новой таблице имя tasks\_rn. Делаем к ней условие порядковый номер <= 3 (последние три обращения клиента), считаем их количсетво и группируем их по теме обращения. Сортировка по возрастанию.

**Задание 2.5**

Для каждой темы обращения найдите среднее время, которое проходит между клиентскими обращениями

select title

, avg(delta\_created\_datetime) as avg\_created\_datetime

from

(select client\_id

, title

, created\_datetime

, created\_datetime - lag(created\_datetime) over (partition by title order by created\_datetime asc) as delta\_created\_datetime

from tasks

) as tasks\_new

group by title

Работаем с таблицей tasks. С помощью lag создаем столбец с датой предшествующей дате в текущей строке. Делаем это в пределах темы обращения. Затем из created\_datetime вычитаем созданный столбец. Получаем разность, дельту. На выход делаем группировку по теме обращений и считаем среднее по дельте. Это и будет средним временем, которое проходит между клиентскими обращениями в пределах каждой темы.

**Задание 3**

*Выберите верные, на ваш взгляд, ответы и укажите их в том же Word-файл, где вы сохранили решение по заданию №2.*

1. Известно, что бокрёнок обязательно или грустный, или ушастый, или то и другое вместе.
   * бокрёнок не может быть безухим;
   * бокрёнок не может быть грустным и безухим одновременно;
   * бокрёнок не может быть веселым и безухим одновременно.
2. Если куздру накормить, то она сразу начнет засыпать.
   * если куздра не засыпает, то она не накормлена;
   * если куздра не накормить, то она не будет засыпать;
   * если куздра засыпает, то она была накормлена.
3. Все бокры умеют играть в салки.
   * не бывает бокров, которые не умеют играть в салки;
   * все, кто умеет играть в салки, являются бокрами;
   * не бывает бокров, которые умеют играть в салки.
4. Куздрята бывают либо толстыми, либо худыми. Неправда, что этот куздренок не худой.
   * этот куздренок толстый;
   * этот куздренок средненький;
   * этот куздренок худой.
5. В природе обнаружено более десятка бокрят. Все обнаруженные бокрята сплошь синие.
   * по крайней мере некоторые из бокрят синие;
   * по крайней мере некоторые из бокрят белые;
   * некоторые бокрята (из тех, что уже обнаружены) могут оказаться не синими.