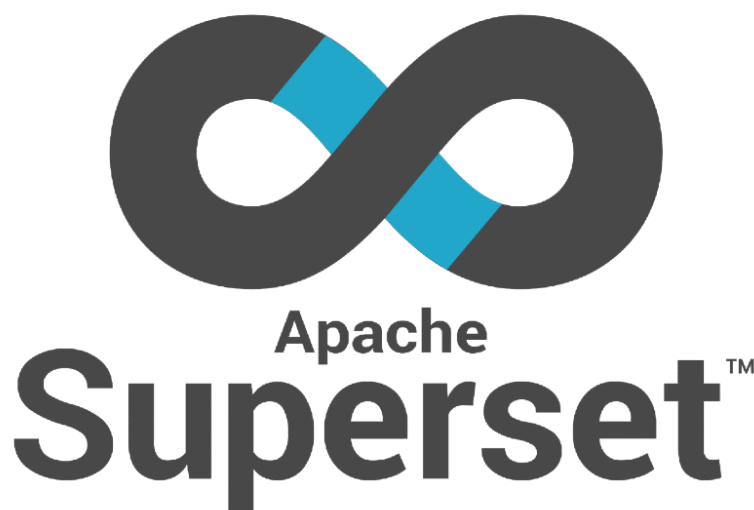




> ПЕРВЫЙ ДАШБОРД

> Что такое Superset?



Apache Superset — это BI-платформа для создания интерактивных визуализаций и дашбордов. Может быть интегрирована с широким набором баз данных (полный список смотрите [в документации](#)) и развёрнута на собственном сервере.

Инструмент **бесплатный**.

Документацию, примеры визуализаций, ссылки на источники информации и ресурсы для обучения можно найти на [официальном сайте](#).

Мы будем пользоваться нашим собственным Superset. Его можно найти по [этой ссылке](#) либо во вкладке **Инструменты**.

Когда зайдёте, вы увидите что-то похожее на это:

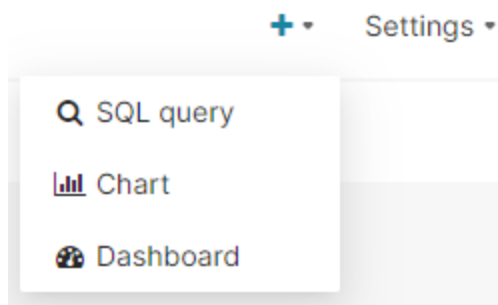
Home

- > Recents
- > Dashboards
- > Charts
- > Saved queries

Здесь можно видеть, что вы недавно создавали или просматривали, а также создавать новые объекты. Всего их три типа:

1. **Дашборды** — интерактивные доски со всей необходимой бизнесу информацией.
2. **Диаграммы** — основной контент дашборда; представляют собой графики, таблицы и другие интерактивные элементы.
3. **Сохранённые запросы** — SQL-запросы, которые вы делали ранее и решили оставить на потом.

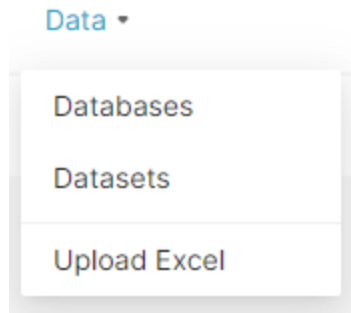
Всё это можно создавать либо в трёх разделах снизу, либо на панели инструментов сверху, либо через кнопку в правом верхнем углу рядом с настройками:



Давайте посмотрим на основные элементы панели инструментов.

> Data

Наиболее простой и непритязательный раздел панели инструментов, касающийся доступа к данным:



Он показывает:

1. Какие **базы данных** нам доступны. В нашем случае база данных всего лишь одна, посвящённая именно *Симулятору Аналитика*.
2. Какие **данные** нам доступны. Здесь можно найти все те наборы данных, которые **находятся** внутри вышеуказанной базы данных, которые вы **создали** в процессе работы над ними и которые вы **загрузили** с компьютера.

Как вы могли понять по последней опции, вы также можете создавать свой собственный набор данных, сохраняя его в Excel-формате и **загружать прямо на Superset**.

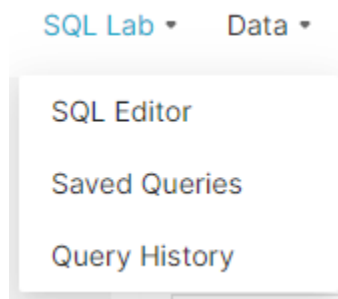
| | |
|----------------|--|
| Table Name * | <input type="text" value="Table Name"/> Name of table to be created from excel data. |
| Excel File * | <div>Выберите файл</div> Файл не выбран Select a Excel file to be uploaded to a database. |
| Sheet Name | <input type="text" value="Sheet Name"/> Strings used for sheet names (default is the first sheet). |
| Database | <div>▼</div> |
| Schema | <input type="text" value="Schema"/> Specify a schema (if database flavor supports this). |
| Table Exists * | <div>Fail ▼</div> If table exists do one of the following: Fail (do nothing), Replace (drop and recreate table) or Append (insert data). |

Superset имеет разные опции, связанные с прочтением данных (обозначение десятичных долей, какие колонки прочесть как даты, какие значения считать пропусками и т.д.). Однако среди них есть три обязательных:

- **Название** таблицы;
- **Файл**, который вы будете загружать;
- Что делать, если такая таблица **уже существует** (варианты: не делать ничего, перезаписать таблицу с нуля, присоединить данные к оригинальной таблице).

> SQL Lab

Раз уж мы работаем с базами данных, то куда же без SQL? В нашем случае это Clickhouse, документацию которого вы можете видеть [тут](#). Обратите внимание на **регистр** при использовании некоторых функций!



Здесь мы можем создавать наши запросы к базе данных, просматривать сохранённые запросы, а также их историю.

Зачем это нужно? Данные не всегда имеют тот формат, с которым нам удобно работать. Многие функции Superset имеют встроенные возможности по преобразованию, агрегации и фильтрации данных, но иногда их возможностей либо не хватает, либо писать такой запрос в маленьком окошке неудобно. Поэтому лучше заранее придать данным нужную форму, а потом приступить к их визуализации.

Самое интересное для нас — **SQL Editor**, который и позволяет работать с таблицами. Что в нём можно делать:

- Выбирать конкретную таблицу и делать её предпросмотр.

SEE TABLE SCHEMA 5 IN SIMULATOR

Select table or type table name



feed_actions



| | |
|-----------|----------|
| user_id | UInt32 |
| post_id | UInt32 |
| action | String |
| time | DATETIME |
| gender | Int8 |
| age | Int16 |
| country | String |
| city | String |
| os | String |
| source | String |
| exp_group | Int8 |

COPY TO CLIPBOARD

Filter results

100 rows returned

| user_id | post_id | action | time | gender | age | country | city | os | source | exp_group |
|---------|---------|--------|---------------------|--------|-----|---------|-----------------|---------|---------|-----------|
| 950 | 10 | view | 2021-08-31T00:13:53 | 1 | 20 | Russia | Novocheboksarsk | iOS | ads | 0 |
| 950 | 10 | like | 2021-08-31T00:15:50 | 1 | 20 | Russia | Novocheboksarsk | iOS | ads | 0 |
| 106919 | 10 | view | 2021-08-31T00:18:44 | 1 | 26 | Russia | Severodvinsk | Android | organic | 4 |
| 533 | 10 | view | 2021-08-31T00:19:06 | 1 | 17 | Russia | Sergiyev Posad | Android | ads | 3 |
| 588 | 71 | view | 2021-08-31T00:22:45 | 0 | 37 | Russia | Chernolesskoye | Android | ads | 2 |
| 528 | 10 | view | 2021-08-31T00:22:51 | 1 | 23 | Ukraine | Irpin | iOS | ads | 1 |
| 528 | 71 | view | 2021-08-31T00:24:24 | 1 | 23 | Ukraine | Irpin | iOS | ads | 1 |
| 528 | 71 | like | 2021-08-31T00:24:29 | 1 | 23 | Ukraine | Irpin | iOS | ads | 1 |
| 106801 | 71 | view | 2021-08-31T00:26:54 | 0 | 30 | Russia | Kemerovo | iOS | organic | 2 |
| 106801 | 71 | like | 2021-08-31T00:28:02 | 0 | 30 | Russia | Kemerovo | iOS | organic | 2 |
| 106496 | 0 | view | 2021-08-31T00:29:50 | 0 | 49 | Russia | Tambov | iOS | organic | 1 |
| 106496 | 0 | like | 2021-08-31T00:31:20 | 0 | 49 | Russia | Tambov | iOS | organic | 1 |

- Писать и выполнять сами запросы. Тут же можно задавать лимит вывода (без необходимости писать руками LIMIT), сохранять запрос на будущее и включать/выключать автодополнение.

1 SELECT *
2 FROM simulator.feed_actions

RUN

LIMIT: 100
00:00:00.35

SAVE AS
COPY LINK
...

RESULTS QUERY HISTORY PREVIEW: 'FEED_ACTIONS'
Autocomplete ☒

- Работать с результатами выполнения этих запросов. Можно фильтровать просматриваемые данные, копировать в буфер обмена, скачивать их на компьютер в формате CSV и самое главное — сразу переходить к их визуализации с помощью кнопки **EXPLORE**:

EXPLORE

DOWNLOAD TO CSV

COPY TO CLIPBOARD

Filter results

100 rows returned The number of rows displayed is limited to 100 by the limit dropdown.

| user_id | post_id | action | time | gender | age | country | city | os | source | exp_group |
|---------|---------|--------|---------------------|--------|-----|---------|-----------|---------|---------|-----------|
| 33138 | 2484 | view | 2021-10-03T05:44:11 | 1 | 15 | Russia | Volgograd | Android | ads | 4 |
| 25919 | 2385 | view | 2021-10-03T05:44:53 | 1 | 19 | Russia | Moscow | iOS | ads | 1 |
| 33138 | 2484 | like | 2021-10-03T05:45:03 | 1 | 15 | Russia | Volgograd | Android | ads | 4 |
| 33138 | 2502 | view | 2021-10-03T05:45:22 | 1 | 15 | Russia | Volgograd | Android | ads | 4 |
| 133957 | 2419 | view | 2021-10-03T05:45:27 | 0 | 24 | Russia | Biysk | iOS | organic | 3 |
| 33138 | 2418 | view | 2021-10-03T05:45:31 | 1 | 15 | Russia | Volgograd | Android | ads | 4 |
| 25919 | 2266 | view | 2021-10-03T05:45:33 | 1 | 19 | Russia | Moscow | iOS | ads | 1 |
| 133957 | 2498 | view | 2021-10-03T05:45:34 | 0 | 24 | Russia | Biysk | iOS | organic | 3 |
| 25919 | 2491 | view | 2021-10-03T05:45:35 | 1 | 19 | Russia | Moscow | iOS | ads | 1 |
| 133957 | 2485 | view | 2021-10-03T05:45:35 | 0 | 24 | Russia | Biysk | iOS | organic | 3 |
| 133957 | 2321 | view | 2021-10-03T05:45:39 | 0 | 24 | Russia | Biysk | iOS | organic | 3 |
| 139336 | 2520 | view | 2021-10-03T05:45:43 | 0 | 30 | Russia | Voronezh | Android | organic | 3 |

При переходе к визуализации Superset позволяет сохранить результат в виде виртуального набора данных либо перезаписать существующий.

Save or Overwrite Dataset



Save this query as a virtual dataset to continue exploring

☒ Save as new

Query simulator.feed_actions 10/30/2021 18:44:32

☐ Overwrite existing

Select or type dataset name

SAVE & EXPLORE

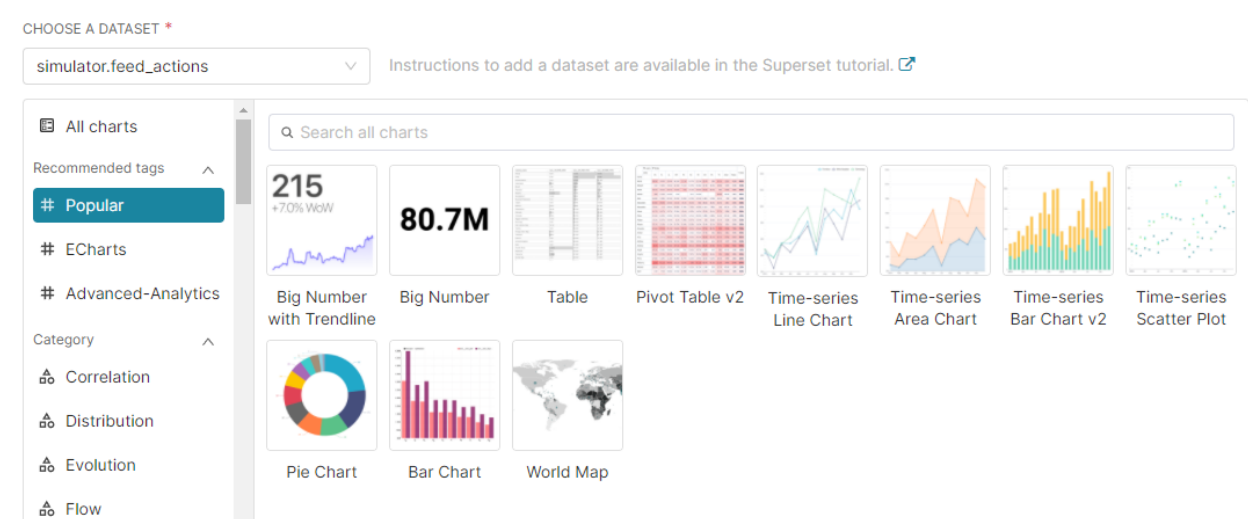
После этого мы можем перейти к самому главному, ради чего создавался Superset :)

> Charts

Именно здесь мы и создаём диаграммы, важные для нашего бизнеса!

Superset предоставляет широкий набор визуализаций, рассмотреть все из которых не представляется возможным. Поэтому сделаем общий обзор возможностей Superset.

Первым делом необходимо выбрать тот набор данных, которые вы планируете визуализировать. Перед вами будет следующий набор вариантов:



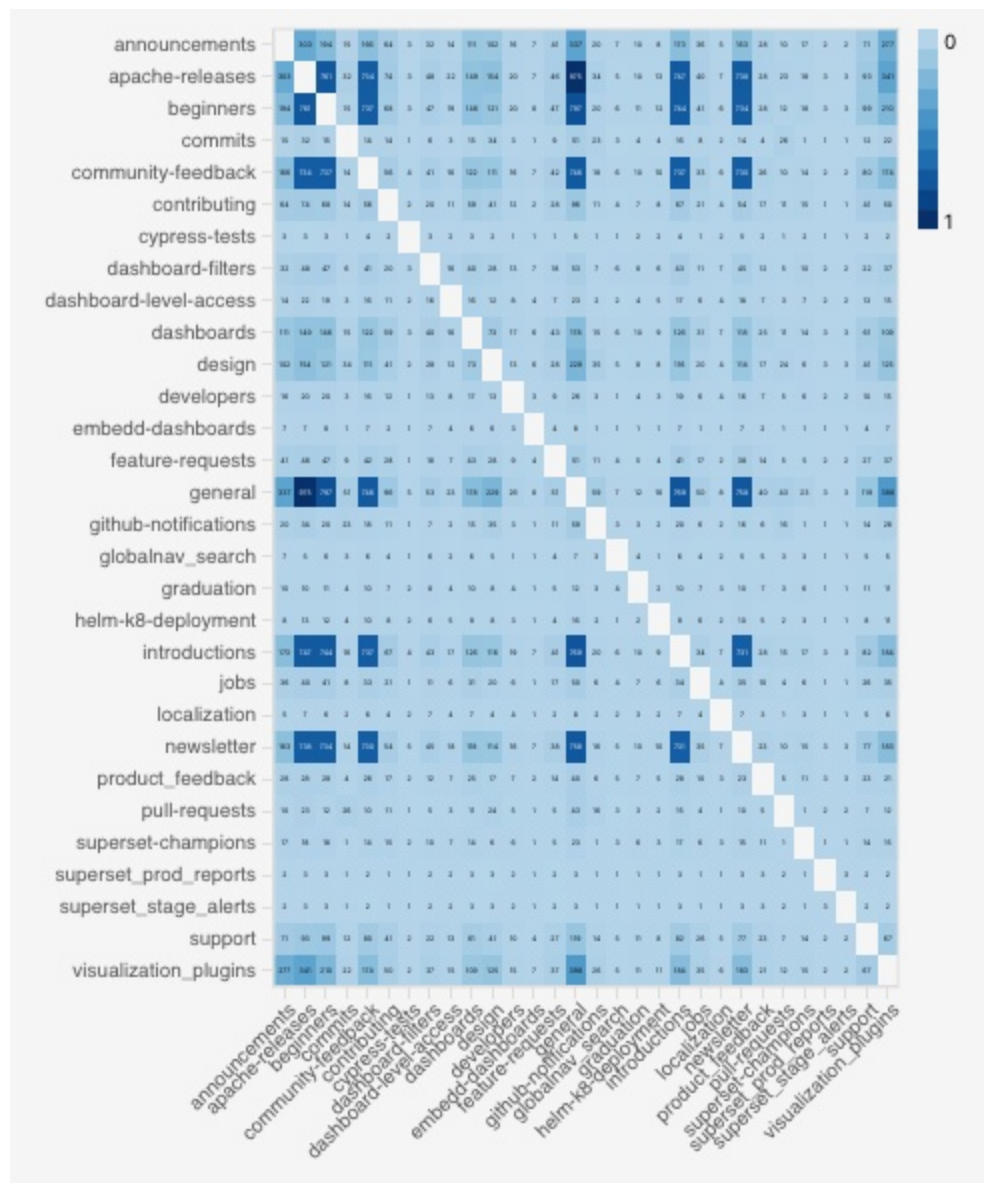
Диаграммы делятся по **категориям** и по **тегам**.

Категории обозначают основную идею — для визуализации чего используют именно такой график или таблицу.

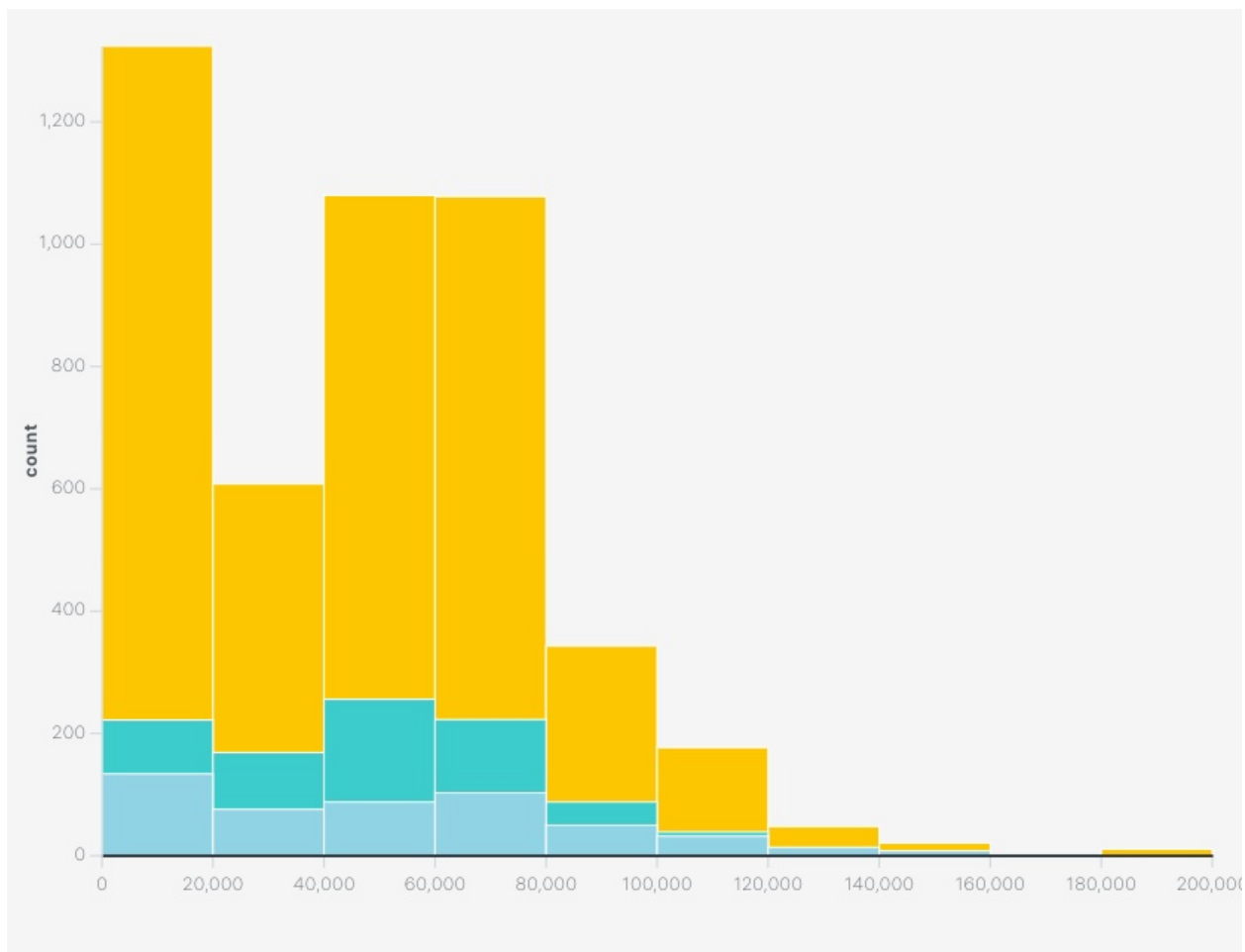
Теги обозначают какое-то свойство диаграммы — она 2D или 3D, изображает тренд, используется для сравнения величин и т.д.

Категорий гораздо меньше, чем тегов, поэтому пройдемся по ним. Можете заметить, что границы между ними порой очень условные, но тем не менее эти категории полезны, поскольку образуют некоторую систему.

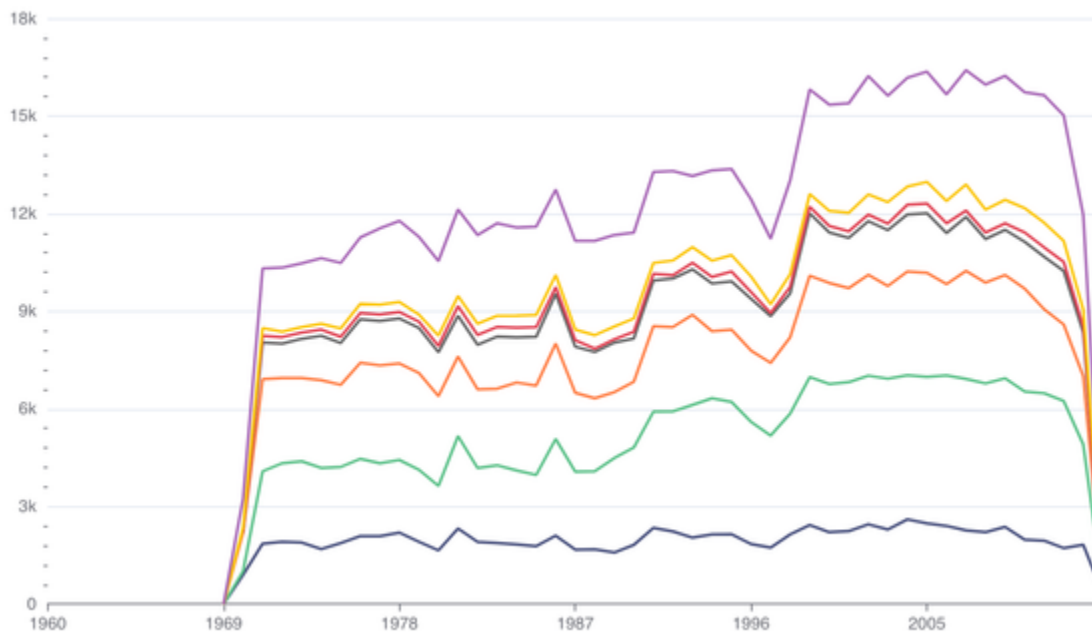
- **Correlation** — подходит для визуализации связей между разными величинами. **Тепловая карта** является самым очевидным вариантом такого рода визуализации. Однако есть и менее очевидные варианты трактовки "связи" — например, таблица с парными t-тестами.



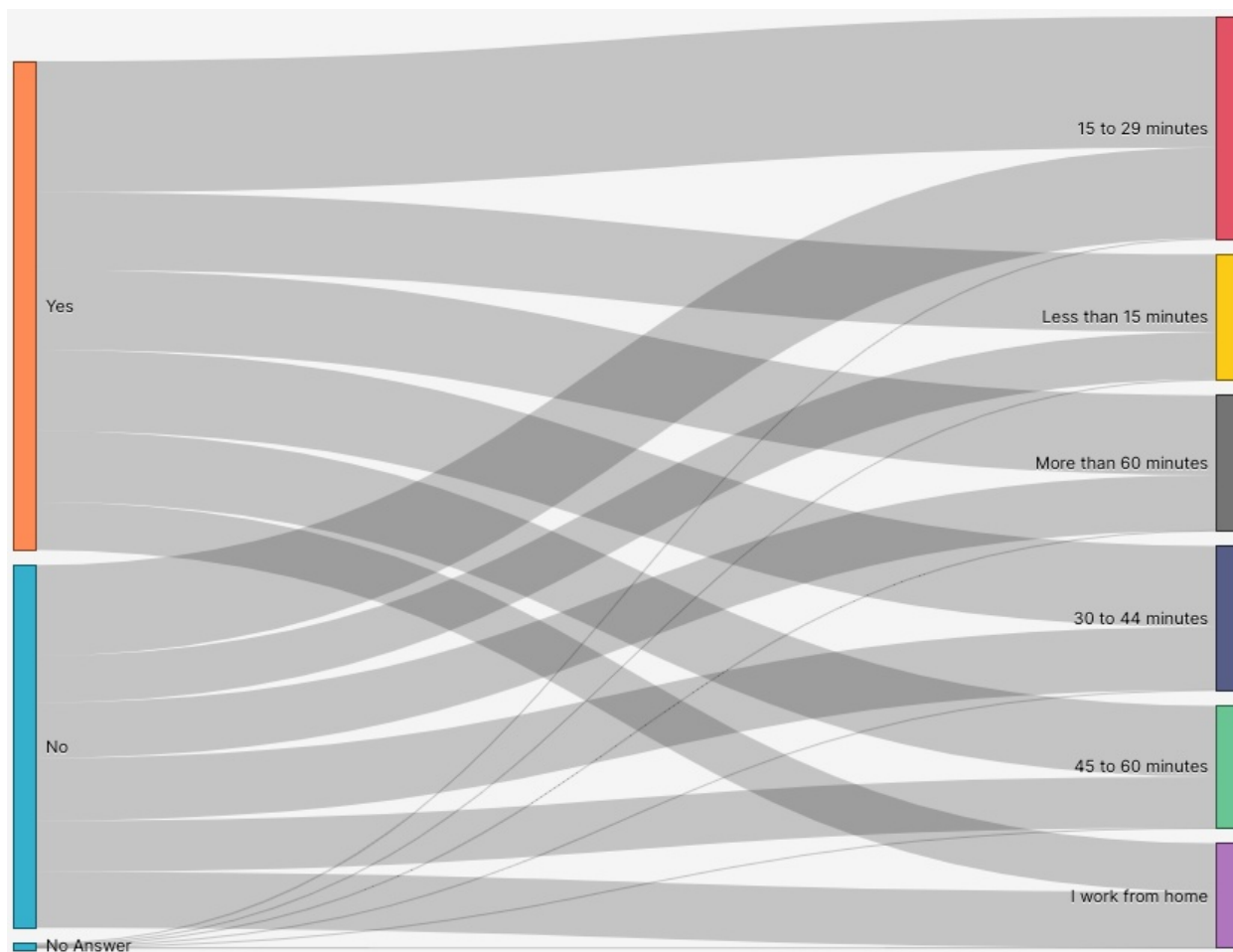
- **Distribution** — подходит для распределений величин. Для единичной величины есть **гистограмма**, для нескольких групп — **боксплот**, для изменений распределения во времени — **horizon chart**.



- **Evolution** — подходит для визуализации величин, которые изменяются со временем. Здесь большинство графиков, связанных с временными рядами, например, **линейная диаграмма**.



- **Flow** — концептуально довольно близко к Correlation, но акцент на "поток" некоторой информации из одного места в другое. Если видите кучу элементов, связанных ленточками разной толщины — это оно. К таким в первую очередь относится **диаграмма Санкея**.

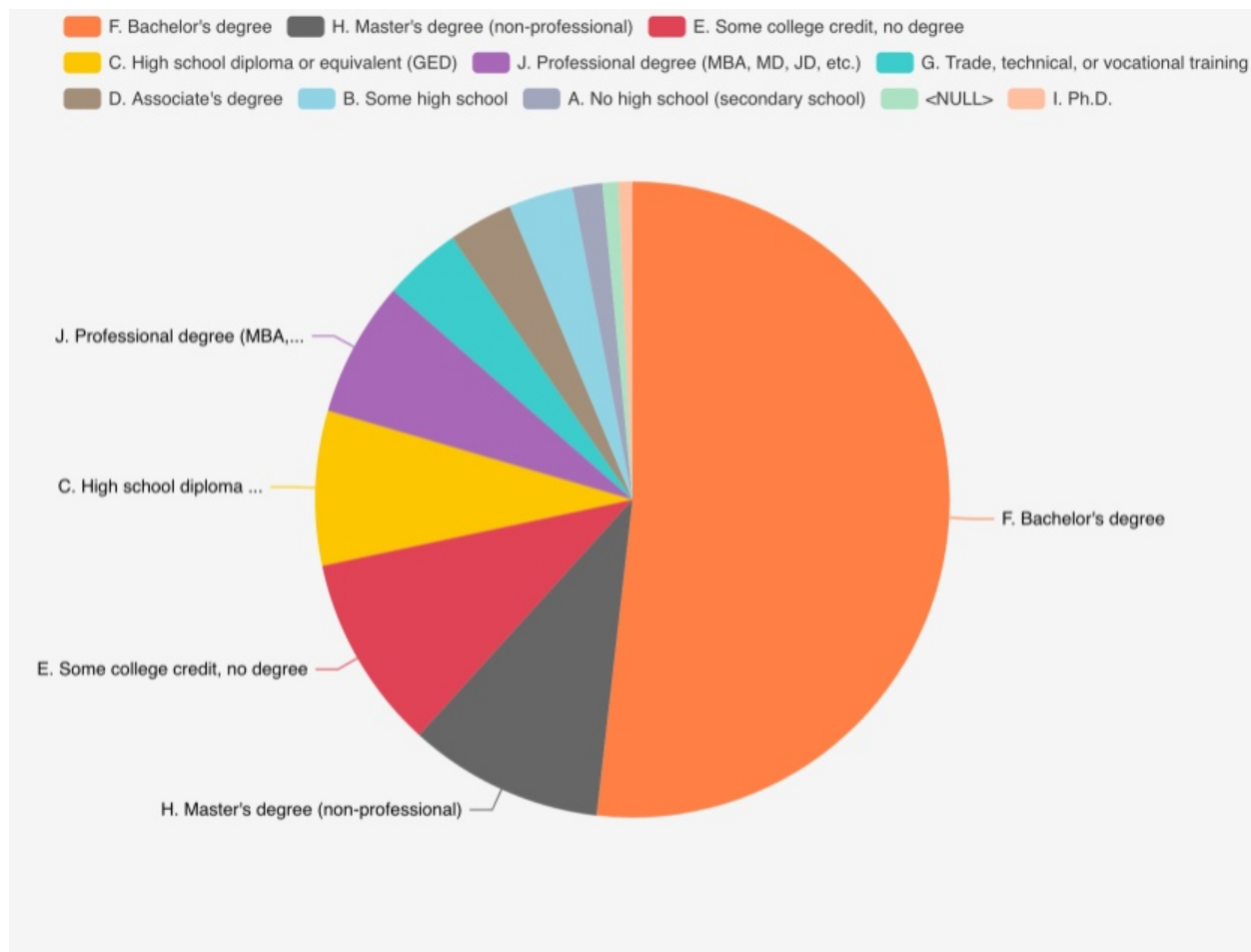


- **KPI** — наиболее простые и неприхотливые показатели "для менеджеров", отражающие конкретные изменения наиболее важных метрик. Как правило, это просто **численная величина**, но могут быть и дополнительные элементы для наглядности.

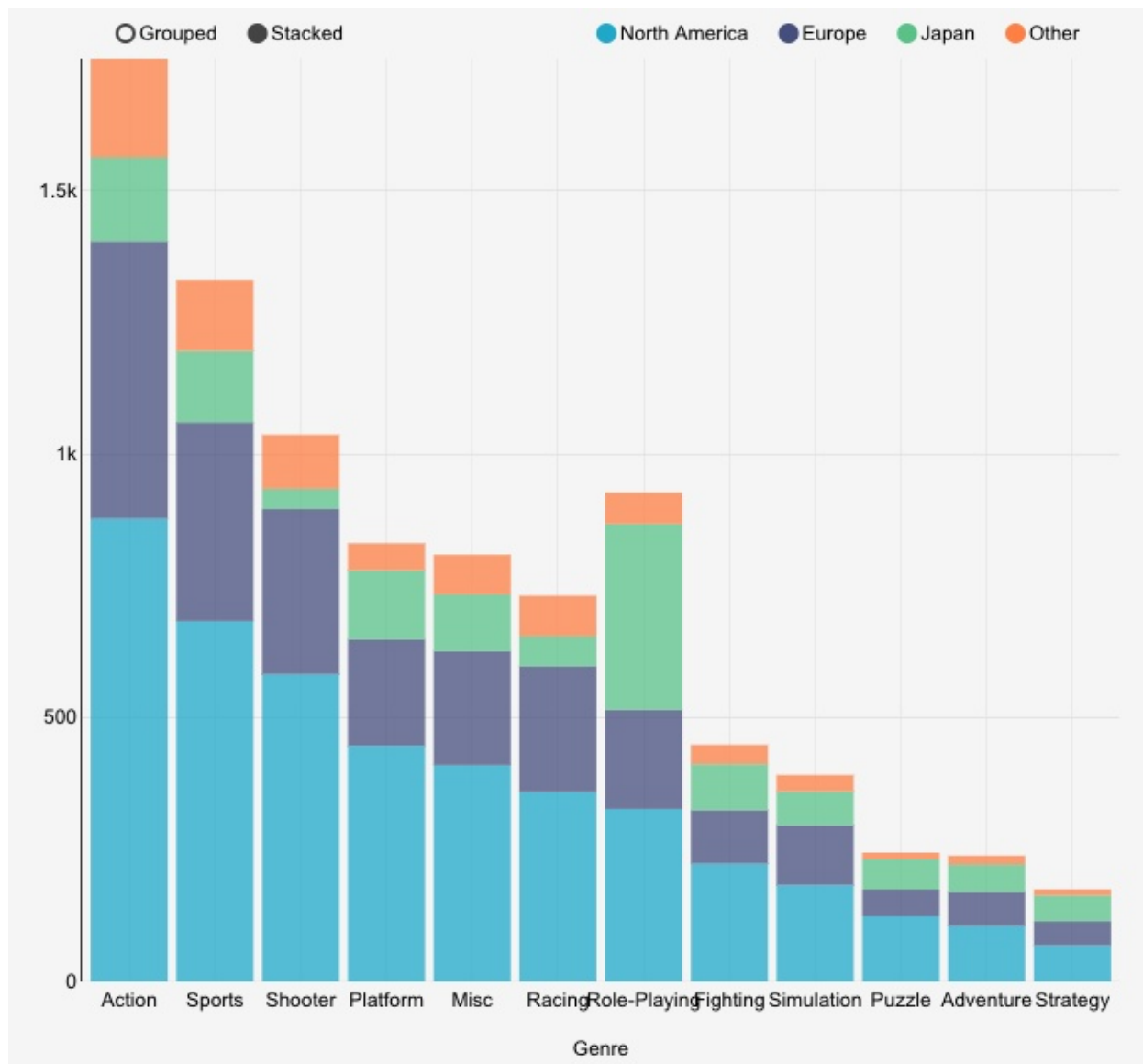
4.73k

Developers

-
- **Map** — как следует из названия, это самые разные карты.



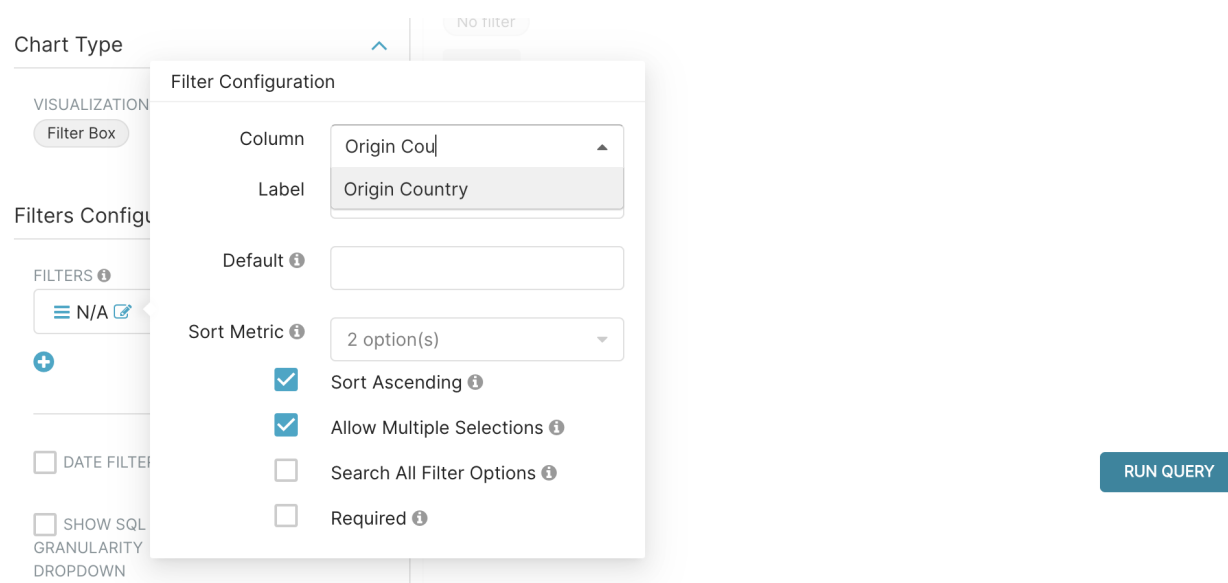
- **Ranking** — позволяет отвечать на вопросы "что больше, а что меньше". Брат-близнец предыдущей категории. Наиболее характерный график — **столбиковая диаграмма**.



- **Table** — таблички. Подходят, когда важна не столько визуализация, сколько конкретные данные.

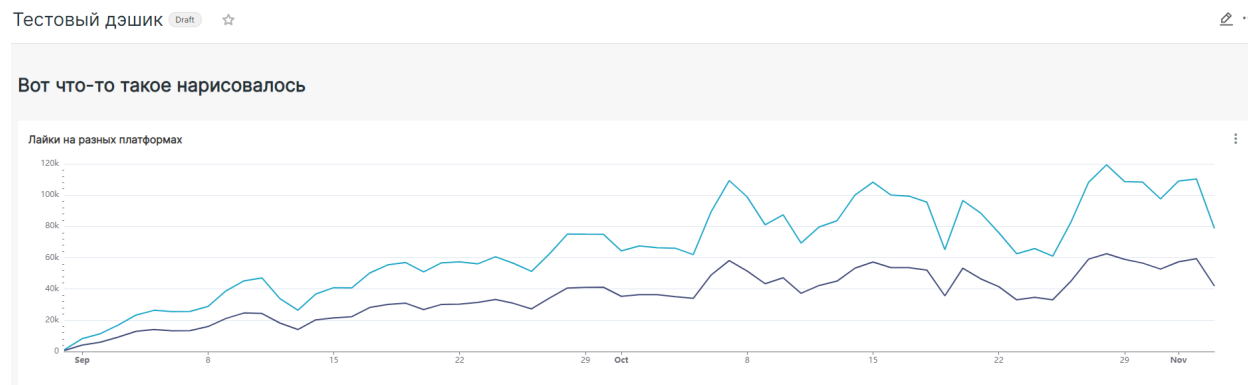
| name | height | mass | hair_color | skin_color | eye_color | birth_year | gender | homeworld | species |
|-----------------------|--------|-----------|---------------|------------------|-----------|------------|---------------|------------|----------------|
| Luke Skywalker | | 172 77 | blond | fair | blue | 19BBY | male | Tatooine | Human |
| C-3PO | | 167 75 | N/A | gold | yellow | 112BBY | N/A | Tatooine | Droid |
| R2-D2 | | 96 32 | N/A | white, blue | red | 33BBY | N/A | Naboo | Droid |
| Darth Vader | | 202 136 | none | white | yellow | 41.9BBY | male | Tatooine | Human |
| Leia Organa | | 150 49 | brown | light | brown | 19BBY | female | Alderaan | Human |
| Owen Lars | | 178 120 | brown, grey | light | blue | 52BBY | male | Tatooine | Human |
| Beru Whitesun lars | | 165 75 | brown | light | blue | 47BBY | female | Tatooine | Human |
| R5-D4 | | 97 32 | N/A | white, red | red | N/A | N/A | Tatooine | Droid |
| Biggs Darklighter | | 183 84 | black | light | brown | 24BBY | male | Tatooine | Human |
| Obi-Wan Kenobi | | 182 77 | auburn, white | fair | blue-gray | 57BBY | male | Stewjon | Human |
| Anakin Skywalker | | 188 84 | blond | fair | blue | 41.9BBY | male | Tatooine | Human |
| Wilhuff Tarkin | | 180 N/A | auburn, grey | fair | blue | 64BBY | male | Eriadu | Human |
| Chewbacca | | 228 112 | brown | N/A | blue | 200BBY | male | Kashyyyk | Wookiee |
| Han Solo | | 180 80 | brown | fair | brown | 29BBY | male | Corellia | Human |
| Greedo | | 173 74 | N/A | green | black | 44BBY | male | Rodia | Rodian |
| Jabba Desilijic Tiure | | 175 1,358 | N/A | green-tan, brown | orange | 600BBY | hermaphrodite | Nal Hutta | Hutt |
| Wedge Antilles | | 170 77 | brown | fair | hazel | 21BBY | male | Corellia | Human |
| Jek Tono Porkins | | 180 110 | brown | fair | blue | N/A | male | Bestine IV | Human |
| Yoda | | 66 17 | white | green | brown | 896BBY | male | N/A | Yoda's species |
| Palpatine | | 170 75 | grey | pale | yellow | 82BBY | male | Naboo | Human |
| Boba Fett | | 183 78.2 | black | fair | brown | 31.5BBY | male | Kamino | Human |
| IG-88 | | 200 140 | none | metal | red | 15BBY | none | N/A | Droid |

- **Tools** — состоит ровно из одного элемента, а именно — **окошка фильтрации**. Полезен исключительно как вспомогательный инструмент к остальным визуализациям, т.к. позволяет отображать конкретные куски данных.



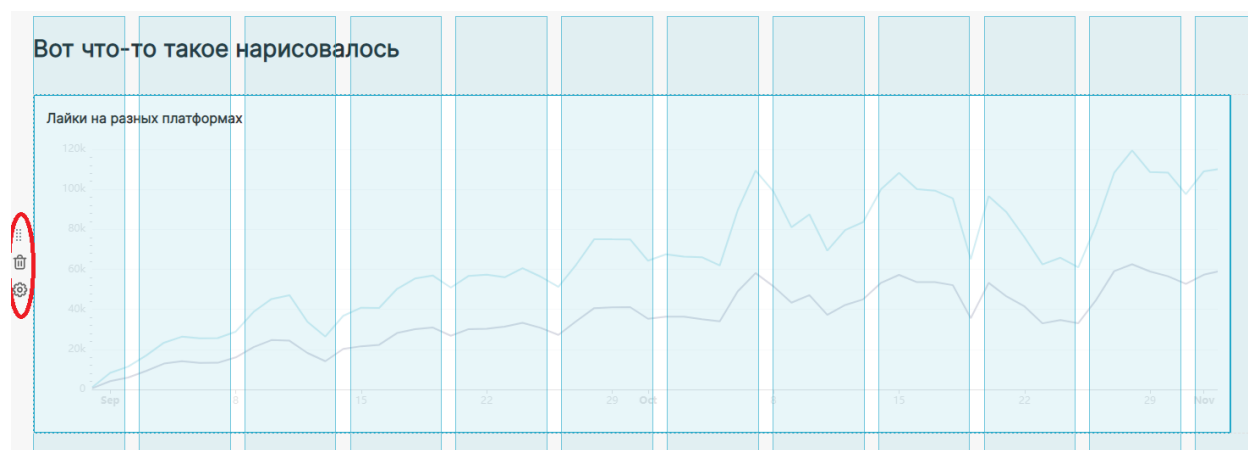
> Dashboards

Основная причина, по которой мы обычно пользуемся Superset :) Здесь мы помещаем созданные нами графики на дашборд, который можно давать в пользование другим:

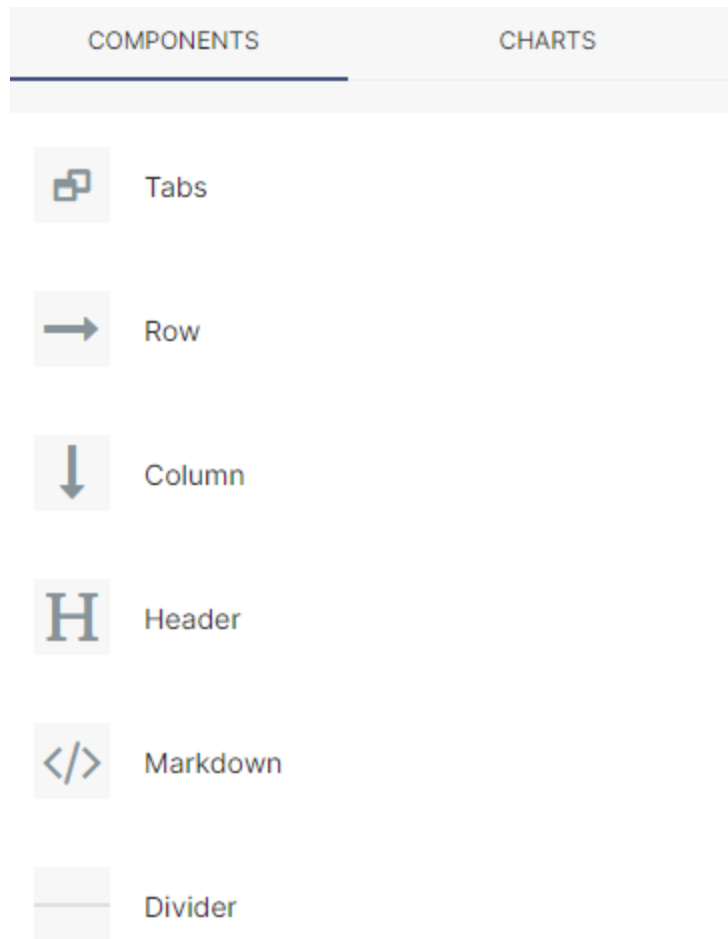


Обратите внимание на тег **Draft** рядом с названием дашборда — при нажатии на него дашборд становится доступным для просмотра остальным! В любой момент можно вернуться обратно, если не хотите, чтобы дашборд отображался в общем списке.

Размеры и положение графиков можно регулировать их перетаскиванием и растягиванием. Также при выборе объекта слева посередине есть небольшая панель инструментов, через которую можно удалять и менять настройки отображения:



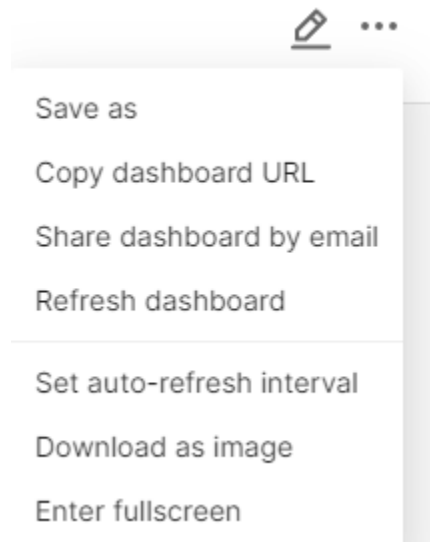
В окошке справа вы можете выбирать диаграммы, которые хотите разместить на дашборде (вкладка **Charts**), а также разные сопроводительные элементы:



- **Вкладки (Tabs)** — позволяет разместить графики на нескольких вкладках, которые можно переключать; полезен для экономии места, особенно если графики объединены каким-то общим принципом (например, DAU, WAU и MAU из лекции вполне можно было разместить в трёх разных вкладках и переключать их в зависимости от необходимого временного разрешения).
- **Ряд (Row)** и **колонка (Column)** — создают свободное место под новые диаграммы; полезны для организации элементов дашборда (горизонтально либо вертикально).
- **Заголовок (Header)** — имя раздела дашборда; можно менять размер текста, но только указанными в Superset градациями.
- **Маркдаун-разметка (Markdown)** — позволяет писать текст произвольного форматирования, используя соответствующую разметку; применяется для аннотации дашборда текстом.

- **Разделитель (Divider)** — незаменимый компонент, если графики нужно как-то отделить друг от друга в целях более удачной компоновки.

После создания дашборд можно снова редактировать (нажав на иконку в виде карандаша). Также можно сохранять его копию, сохранять его как изображение и давать ссылку на него:

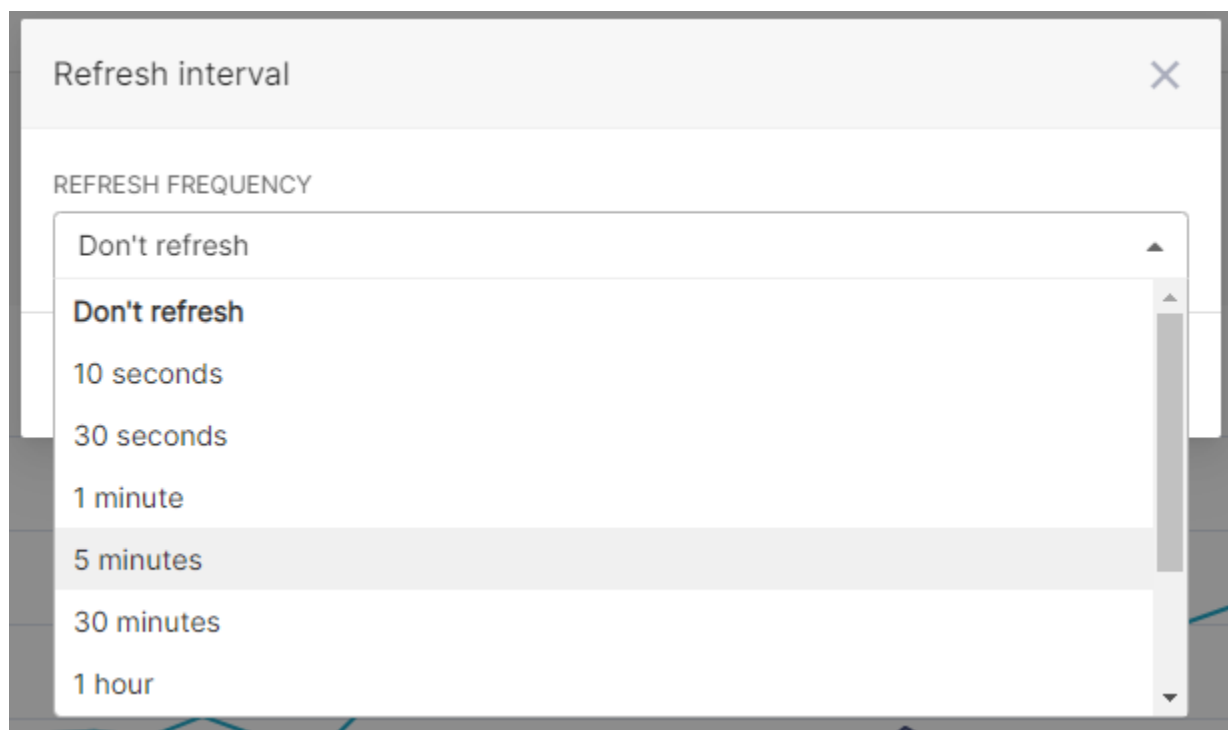


Здесь также есть другой важный функционал, о котором мы поговорим в следующем шаге :)

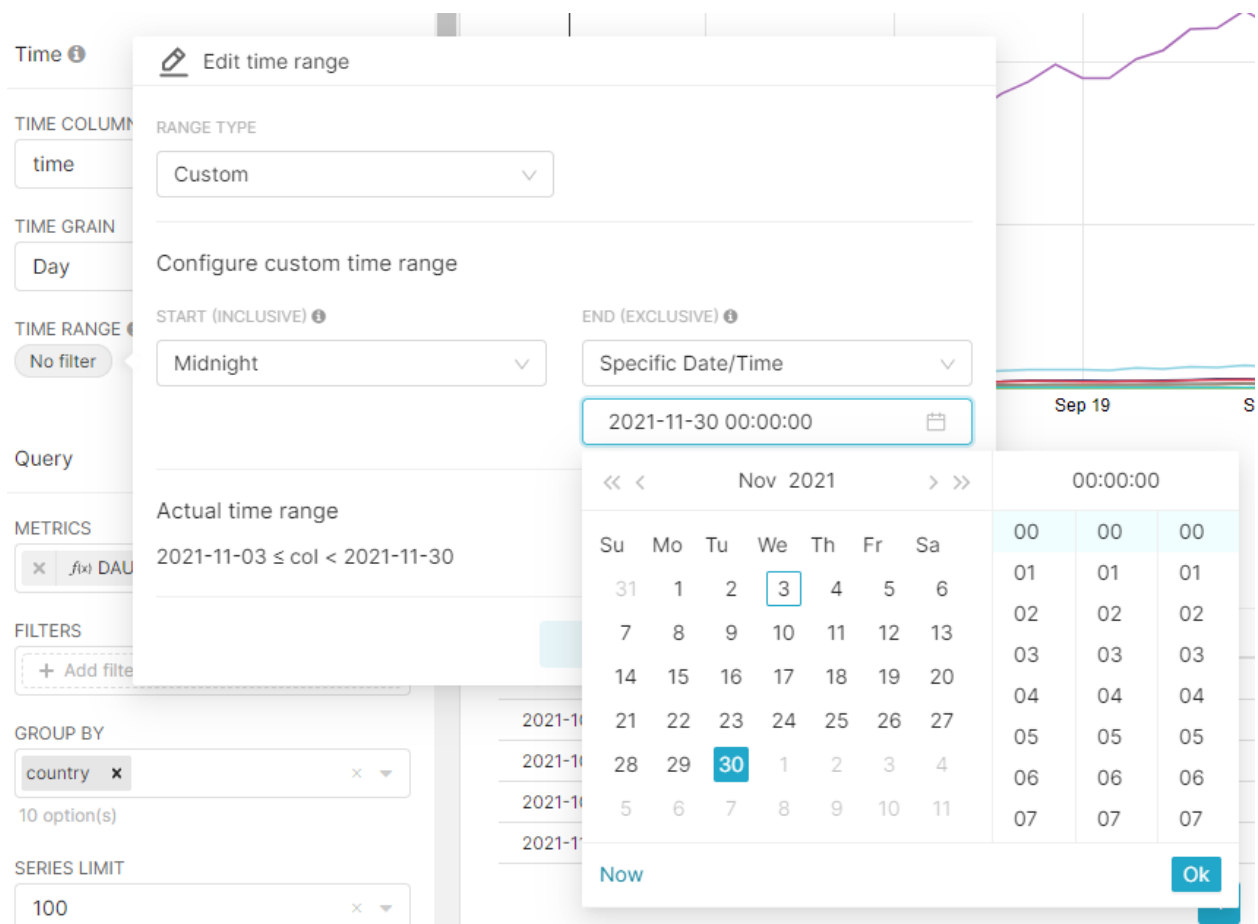
> **Оперативные данные**

Особенно важная интерактивная фишка дашборда — обновление графика, когда меняются связанные с ним данные. Например, с момента создания дашборда прошёл месяц, за этот месяц база данных пополнилась новыми наблюдениями, и мы хотим видеть их на том же графике в дашборде. Особенно это хорошо работает в контексте **оперативных данных** — это данные за короткий промежуток времени, чтобы следить за динамикой данных "в моменте" (как это делал Анатолий в лекции на 15-минутных интервалах).

Мы можем обновить дашборд вручную через **Refresh dashboard**, но также можем задать автоматическое обновление дашборда через определённые временные интервалы. Это делается через **Set auto-refresh interval**:



Если график содержит шкалу времени (что характерно для временных рядов), мы можем задавать на самом графике диапазон, в котором будут отображаться наши данные. В этом нам поможет опция **time range**:



> В чём идея задачи?

В компании пока не выстроены аналитические процессы, мы будем начинать с нуля. Хорошо, что все необходимые инструменты для анализа данных у нас под рукой.

Первое, с чего нужно начать, — это базовые дашборды, покрывающие ключевые метрики продукта. Было бы ошибкой сразу пытаться начать решать продуктовые задачи, погружаться в анализ причин оттока пользователей, придумывать A/B-тесты и т.д. Мы к этому ещё вернёмся, но сначала нужно закрыть самый первый этап аналитики в компании. А именно — сделать так, чтобы мы могли без труда и ручных выгрузок отвечать на самые простые вопросы о нашем приложении!

Сколько у нас активных пользователей в день, неделю, месяц? Сколько у нас лайков и просмотров постов за последние сутки, за последний час, за последние пятнадцать минут? Какие посты у нас самые популярные? Какие юзеры находятся в топе по просмотрам? Можем ли мы посмотреть эти данные в разрезе

операционной платформы или пола пользователей? Именно на такие вопросы мы должны научиться быстро отвечать на первом этапе.

К сожалению, даже в больших технологических компаниях часто можно увидеть следующую картину. Поступает очень простой вопрос: сколько уникальных пользователей поставили лайк за прошедшую неделю? Этот вопрос переправляют аналитику данных, он начинает конструировать SQL-запрос к нескольким таблицам, чтобы решить поставленную задачу. В лучшем случае код запроса сохранится, тогда в следующий раз можно будет его использовать снова. В худшем случае уже через неделю другой аналитик будет делать всё то же самое.

Главным инструментом, который помогает компании отвечать на указанные выше вопросы, служит BI-система. Не Jupyter, не ручные выгрузки и не что-либо ещё.

Часто внедрение BI осложняется тем, что архитектура хранилища данных плохо подходит для аналитических запросов. Однако дата-инженеры нашей компании уже постарались и настроили сбор данных в ClickHouse, подключив все необходимые BI-инструменты.

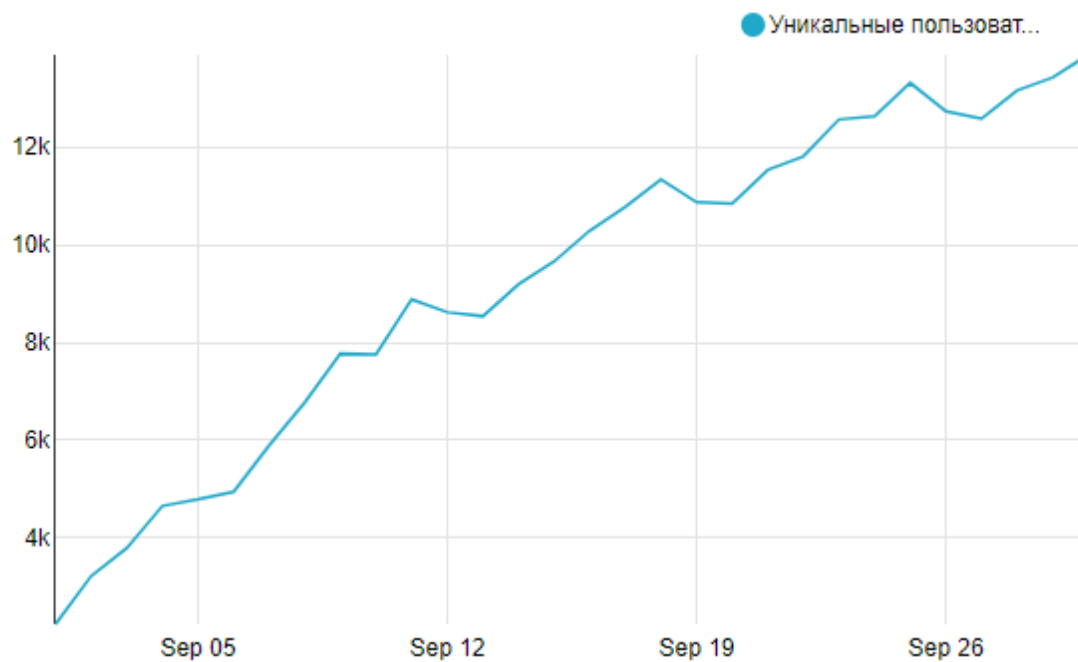
Теперь задача стоит уже перед аналитиками — необходимо построить удобные и простые в использовании дашборды.

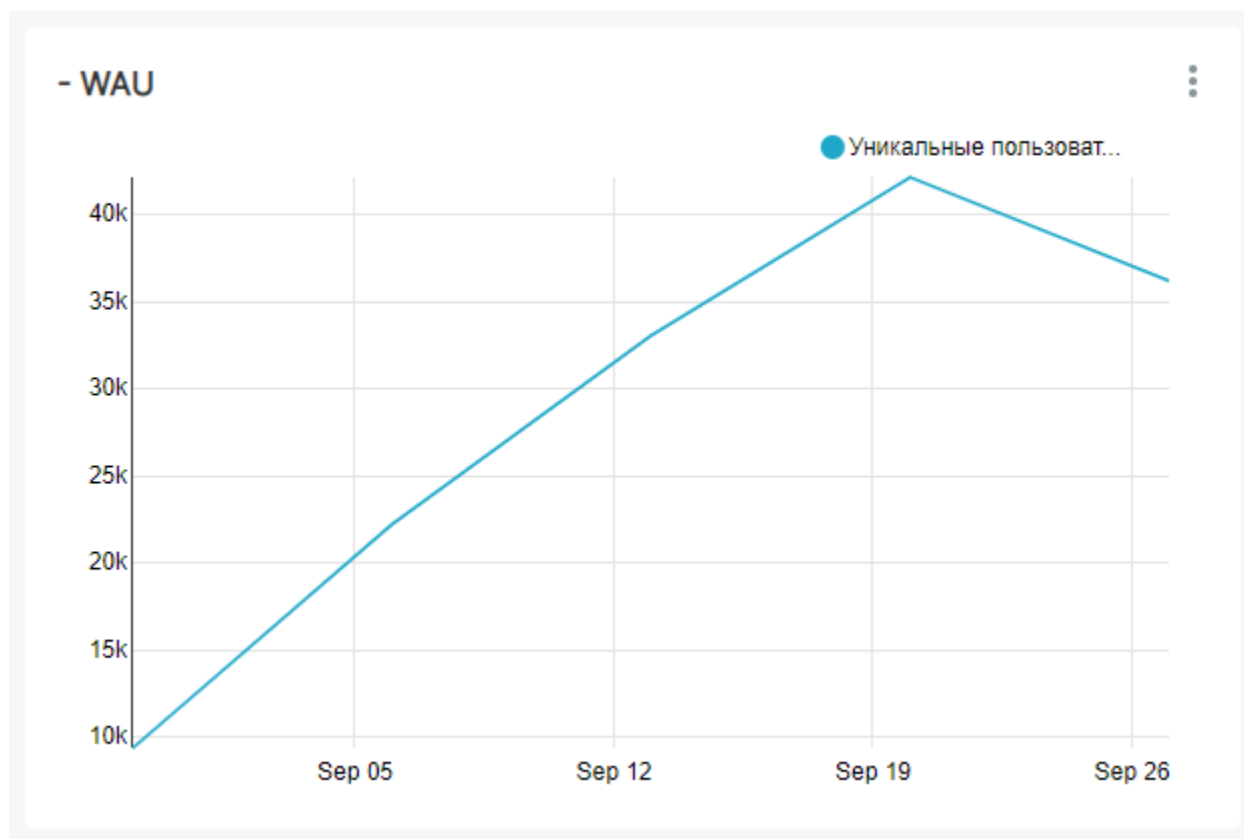
> Что поместить на дашборд?

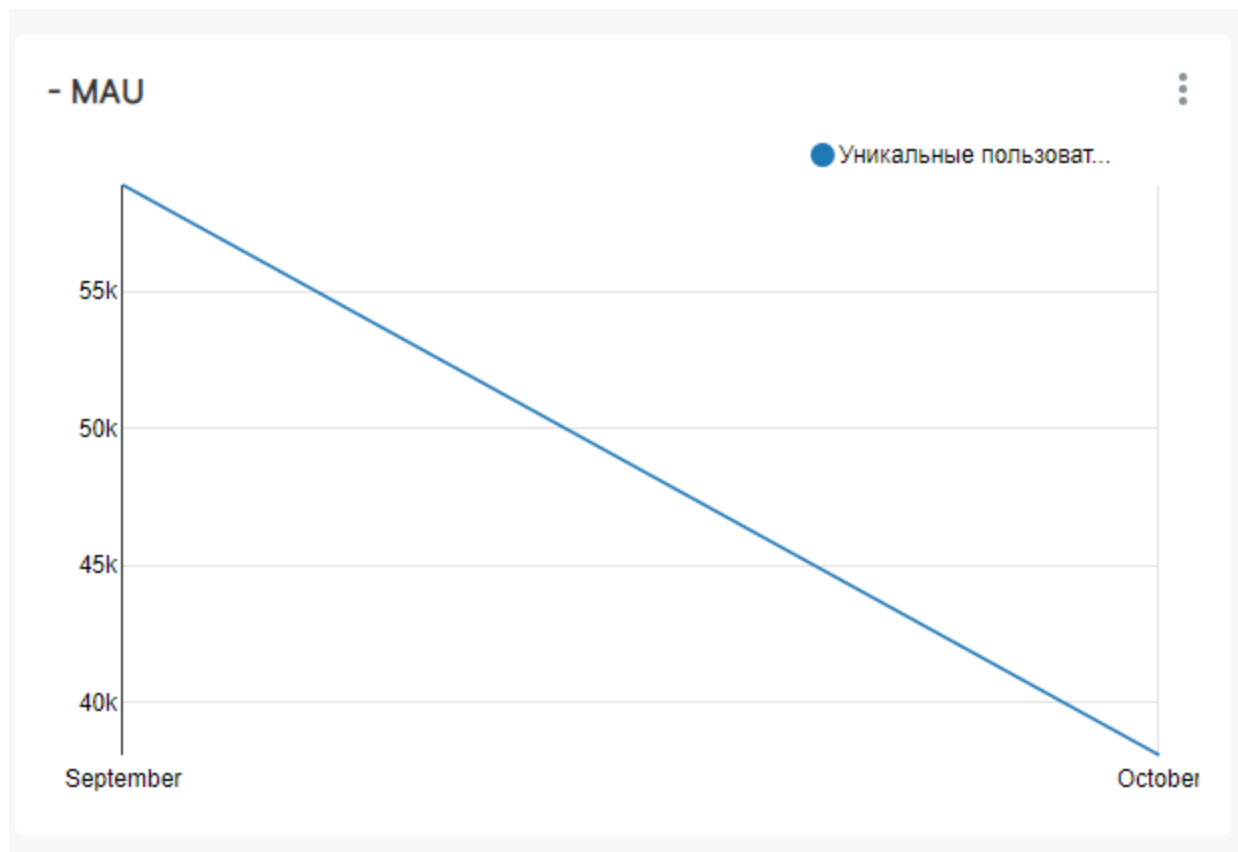
Давайте приступим к созданию дашборда для сервиса просмотра постов. Как мы уже обсудили выше, на первом уровне нам нужно ответить на вопрос "сколько?". Начнём с аудиторных метрик — сколько активных пользователей у нас было за конкретный период: день, неделю и месяц. Это DAU, WAU и MAU соответственно.

Аудиторные метрики (DAU, WAU и MAU) — это одни из самых важных метрик продукта. Внимания заслуживают также их комбинации, например, отношение DAU к MAU. Чем ближе значение этой метрики к единице, тем большая часть активной дневной аудитории пользуется сервисом каждый день в течение месяца.

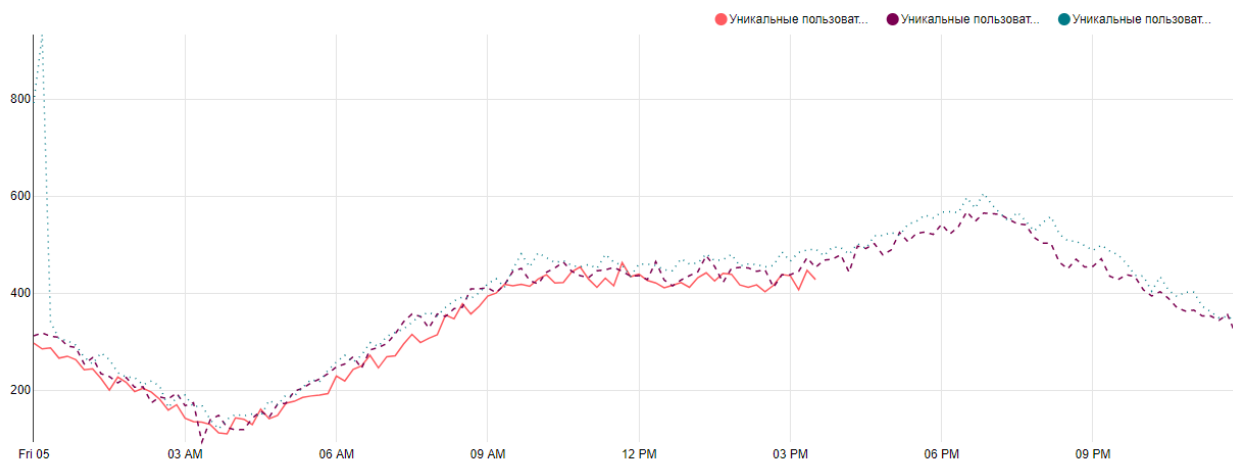
- DAU







Также полезно иметь метрику активных пользователей. Мы можем построить график уникальных активных пользователей по 15-минутным интервалам. Резкое изменение на этом графике, которое не вписывается в дневной тренд, может быть сигналом о том, что в продукте что-то сломалось. Хорошо настроенные дашборды позволяют сразу понять, где именно произошла проблема: например, сломались лайки только на новой версии приложения на iOS. Обнаружив проблему, можно написать коллегам-разработчикам, что нужно срочно решить проблему на новой версии.



Дальше необходимо добавить фильтры для анализа метрик в различных срезах. Вернёмся к структуре таблицы — мы имеем довольно много информации о взаимодействии пользователей с постами. В частности, мы знаем характеристики пользователей: пол, возраст, геолокация, устройство.

TIME RANGE

No filter

OS

Type or Select [os]

GENDER

Type or Select [gender]

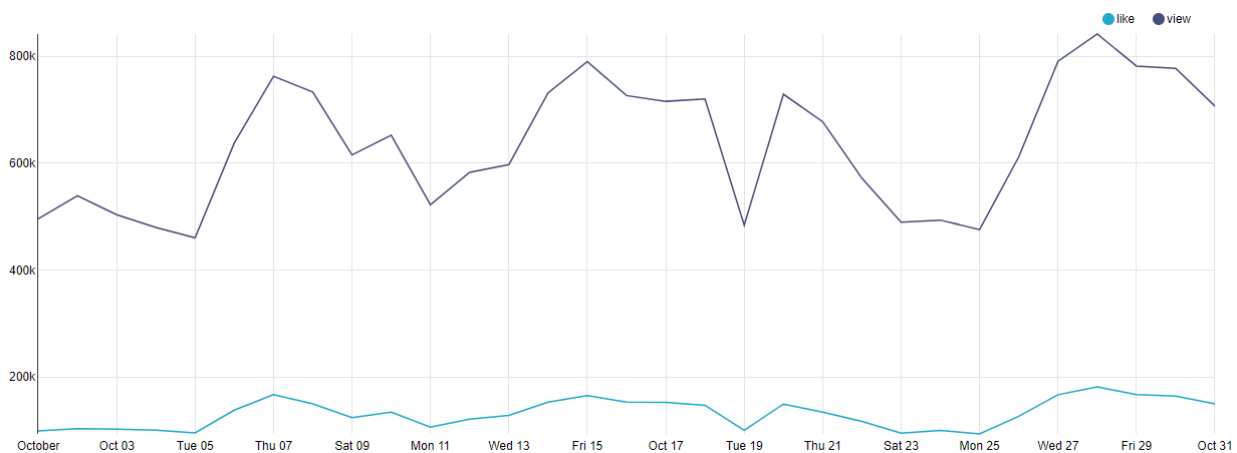
APPLY

Далеко не все свойства постов или юзеров нужно добавлять в срезы на дашборд. Помним, что мы пока хотим отвечать на самые базовые вопросы, а перегруженный дашборд зачастую может только мешать восприятию.

Отдельная задача — понять, какие срезы являются важными, а какие нет. Что важнее: дата публикации или тематика поста? Сходу ответить на этот вопрос довольно тяжело, и понимание приходит только со временем в процессе работы с продуктом. На первом этапе лучше всего спросить у более опытных коллег и будущих пользователей вашего дашборда, на что они в первую очередь хотели бы

смотреть в данных. На первом этапе давайте добавим социально-демографические показатели по юзерам и тематику поста. При необходимости мы легко сможем дополнить этот список.

Теперь перейдём к постам. Давайте также посчитаем число просматриваемых постов за день, неделю, месяц. И добавим метрику количества просмотренных постов в реальном времени. Если с ней что-то пойдёт не так, мы увидим резкое увеличение или падение на графике, это также может быть сигналом, что сервис работает некорректно. При этом не любое аномальное изменение обязательно говорит о проблемах — возможно, вышел очень популярный пост, который стянул на себя внимание большого количества пользователей.



Мы заложили основу дашборда, в рамках стажировки вы можете смело дополнять и модифицировать его. Ваша задача — сделать схожий дашборд для сервиса отправки сообщений.

> Небольшое отступление

Обратите внимание: в нашем случае и само событие взаимодействия с постом, и характеристики пользователей/постов хранятся в одной таблице. У вас может возникнуть закономерный вопрос: насколько это вообще оправдано? Например, один юзер просмотрел 1000 постов в день, сгенерировались 1000 событий, и 1000 раз мы записали один и тот же пол/возраст пользователя. Разве не логичнее было бы хранить в сырых логах только сами события, а свойства пользователей и постов записывать в отдельные таблицы?

Такая архитектура аналитических хранилищ может вам встретиться на практике, и будьте готовы к тому, что вам могут потребоваться дополнительные джойны с таблицами, где хранятся свойства объектов. Но вы также можете встретиться и с нашим решением, когда даже сырые данные обогащаются информацией об объектах. ClickHouse довольно хорошо сжимает данные в колонках, и тот факт, что у нас есть колонка возраста пользователей, не сказывается на скорости работы с данными, но при этом сильно упрощает процесс анализа.