

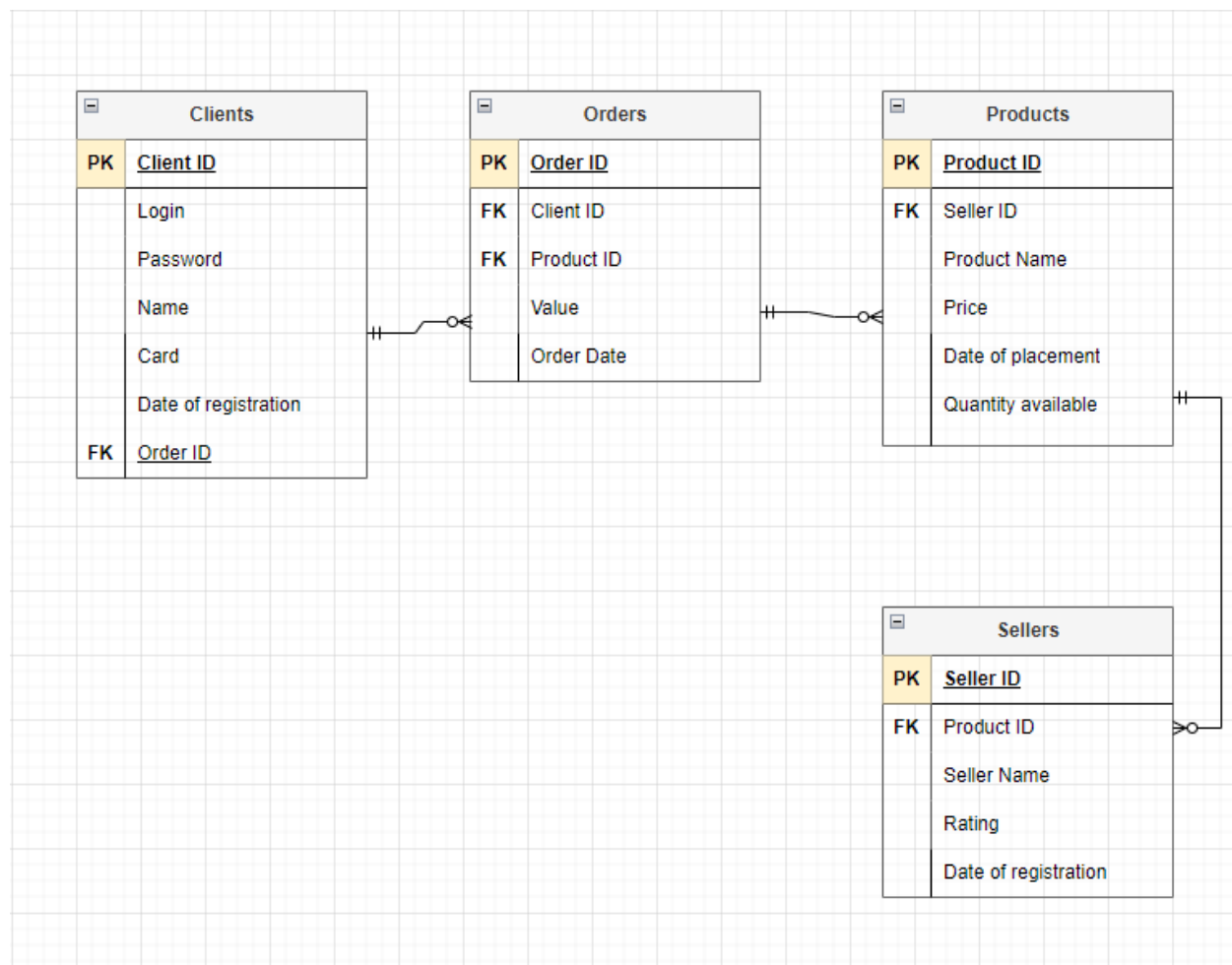
1. Мой выбор – колоночная СУБД ClickHouse.
2. Какие требования к нашей БД?
  - Хранение огромного количества информации
  - Возможность проведения сложных аналитических запросов
  - Аналитика в реальном времени
  - Легкая масштабируемость.

Итак, колоночные СУБД отлично подходят для записи большого количества информации и масштабируемость у них – линейная, что является несомненным плюсом. Для аналитических запросов КликХаус использует простой и понятный инструмент - язык SQL. Это очередной плюс в пользу именно нашего выбора в отличие от какой нибудь MongoDB. И последний аргумент в пользу КликХауса – это скорость. Скорость, с которой эта СУБД позволяет обрабатывать запросы довольно хороша, и при использовании шардирования позволяет увеличить скорость запросов до тысячных доли секунды. Естественно, после определенной цифры рост скорости становится трудным, поскольку в дело вступают задержки сети, но это не отменяет того факта, что КликХаус является одной из самых быстрых СУБД.

3. Таблицы, колонки и связи описываются примерно так же, как и в PostgreSQL. За исключением того, что всегда указывается как правило дата.

Ссылка на диаграмму:

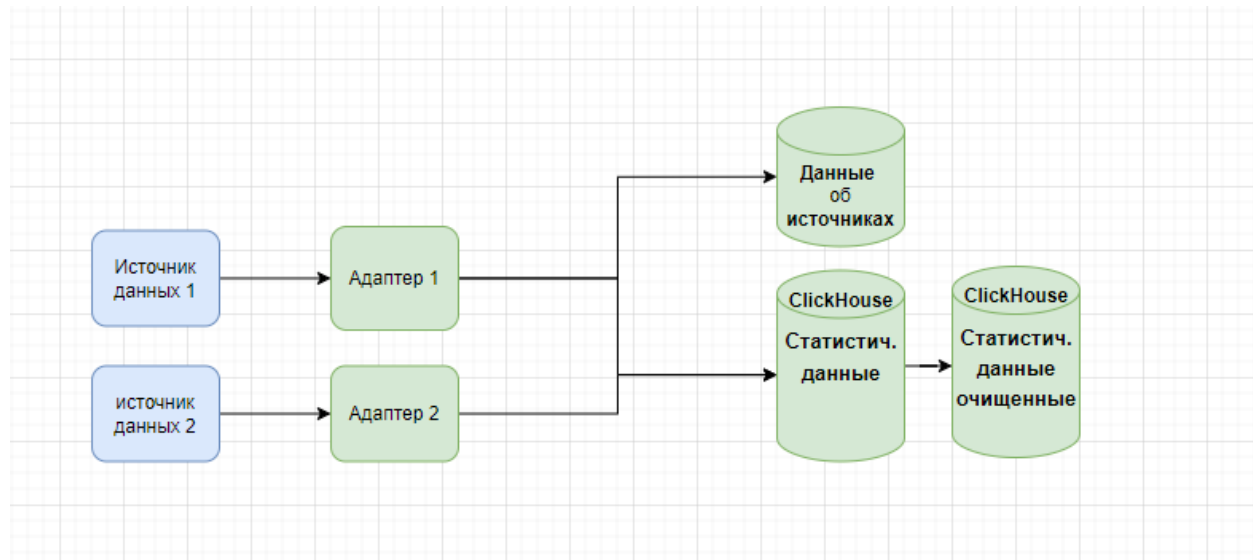
[https://drive.google.com/file/d/1b2HI8sr9eVcWYpJ14yVa\\_DoeEDR3pnV/view?usp=sharing](https://drive.google.com/file/d/1b2HI8sr9eVcWYpJ14yVa_DoeEDR3pnV/view?usp=sharing)



4. Сбор данных из разных источников осуществляется посредством написания адаптера. По-хорошему, неплохо бы иметь по одному адаптеру на каждый источник данных, будь то сайт, база данных, или апишка.

Ссылка на диаграмму:

[https://drive.google.com/file/d/1HpFnf1ZK4fvhbZmx\\_vuOZESB8rV8TAD8/view?usp=sharing](https://drive.google.com/file/d/1HpFnf1ZK4fvhbZmx_vuOZESB8rV8TAD8/view?usp=sharing)



5. Например, использование Kafka и нескольких адаптеров. Kafka поможет сделать нагрузку более распределенной по всем имеющимся адаптерам и не потерять информацию. В нашем случае, при «миллионах записей» это особенно критично. Иными словами – непрерывный большой объем данных, который требуется обработать, Kafka будет держать в себе, пока мы не вычитаем все необходимые данные и не отсортируем. Итак, говоря о схеме – это может быть некий сервис логики, отправляющий запросы, Kafka, хранящие сообщения, и адаптеры, которые занимаются обработкой запросов и отправкой данных из того или иного источника.