Сервис-прокси для обработки запросов (ClickHouse)

Так как в ClickHouse отсутствуют привычные индексы, для оптимального поиска нужно создавать материализованные представления, заточенные под нужды запросов. А также нужен сервис, который будет определять, какой запрос в какое представление отправить.

В данном модуле представлены следующие "оптимизаторы" запросов (приоритет в порядке убывания):

Ключ партицирования	Ключевой атрибут	Таблица
classCode + year	`classCode`	`requestHistoryProxyClassCode`
YYYYMM	`userId`	`requestHistoryProxyUserId`
YYYYMM	`username`	`requestHistoryProxyUsername`
YYYYMM	Все остальные	`requestHistory`

Таким образом, ход работы получается такой:

1) (если используется чтение из kafka) таблица с движком `Kafka` подписывается на топик; 2) (если используется чтение из kafka) материализованное представление считывает данные из `Kafka`, когда они приходят; 3) Данные записываются в основную таблицу; 4) Сразу после вставки, данные читаются в материализованные представления с движком `MergeThree` (по сути, это те же таблицы, только с подпиской).

Особенности sql

При создании материализованных представлений, можно обратить внимание на параметр `POPULATE`. Он отвечает за то, будут ли старые данные читаться в представления или нет.

Кроме того, для экономии места было решено отказаться от данных, которые не используются в поиске.

Выбор ключей партицирования

Для максимальной скорости поиска, необходимы наименьшие ключи партицирования. Кроме того, чтобы данные не мусорились, ключ партицирования должен быть динамически изменяемым. Для этого лучше всего подходит дата. Кроме того, использование даты спасает от другой проблемы: чем больше партиций используется при вставке, тем больше времени занимает вставка. Так как каждый день данные будут отправляться этого же дня, проблем со вставкой не будет.

Остаётся вопрос в использовании даты. Можно брать по дням/месяцам/годам. И в данном случае надо отталкиваться от диапазона поиска. Так как поиск должен идти по минимальному количеству партиций.

Выбор порядка (первичного ключа)

Как говорилось ранее, первичный ключ в таблице только один (не смотря на то, что указать их можно несколько, роль они будут играть только в том случае, когда указан ключ верхний по иерархии). Поэтому выбирать первичный ключ необходимо в зависимости от конкретных целей таблицы. В табличке вверху описано, какие ключи выбирались для той или иной таблицы.