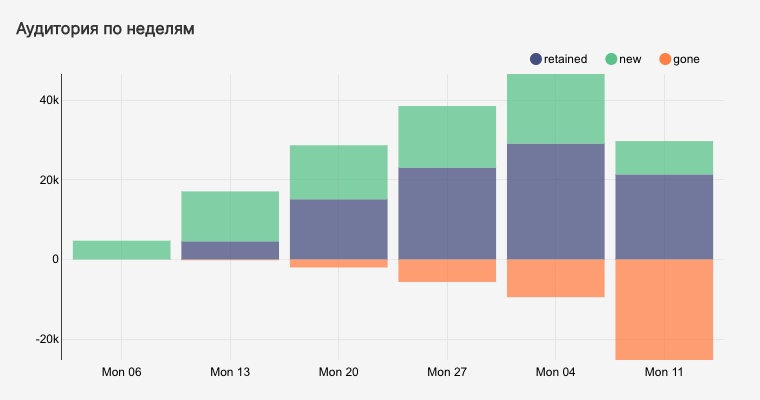
**Задания**

Задание 3

В лекции я показывал график, который позволяет взглянуть на активную аудиторию с точки зрения новых, старых и ушедших пользователей. Попробуйте самостоятельно построить такой график:



На графике мы отобразим активную аудиторию по неделям, для каждой недели выделим три типа пользователей.

Новые — первая активность в ленте была на этой неделе.

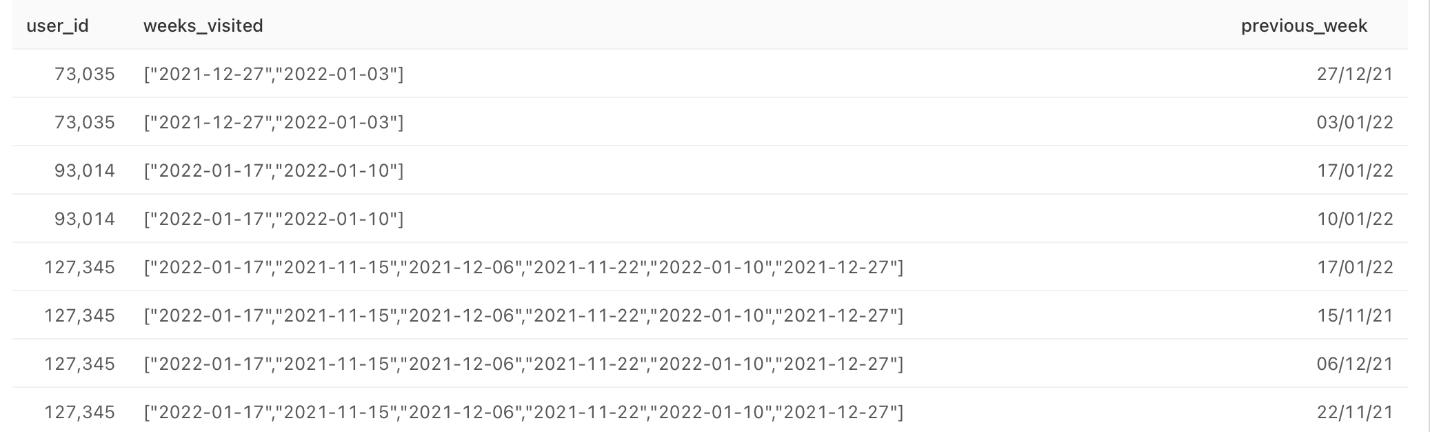
Старые — активность была и на этой, и на прошлой неделе.

Ушедшие — активность была на прошлой неделе, на этой не было.

Для выполнения этого задания есть ряд методологических советов, которые помогут разобраться, если непонятно с чего начать.

Прежде всего, лучше не стараться сделать всё сразу, а выполнить две подзадачи: сначала получить запрос для ушедших пользователей, а затем для новых и старых. Получить уникальные записи user\_id и дата активности достаточно легко напрямую из таблицы, но к этому также необходимо добавить столбец, состоящий из массивов, которые будут содержать все даты активностей для конкретного user\_id. Массив будет назван weeks\_visited, а дату активности - previous\_week, так как будем использовать для старта отсчета именно предыдущую неделю, когда достоверно известно, что пользователь был активен.

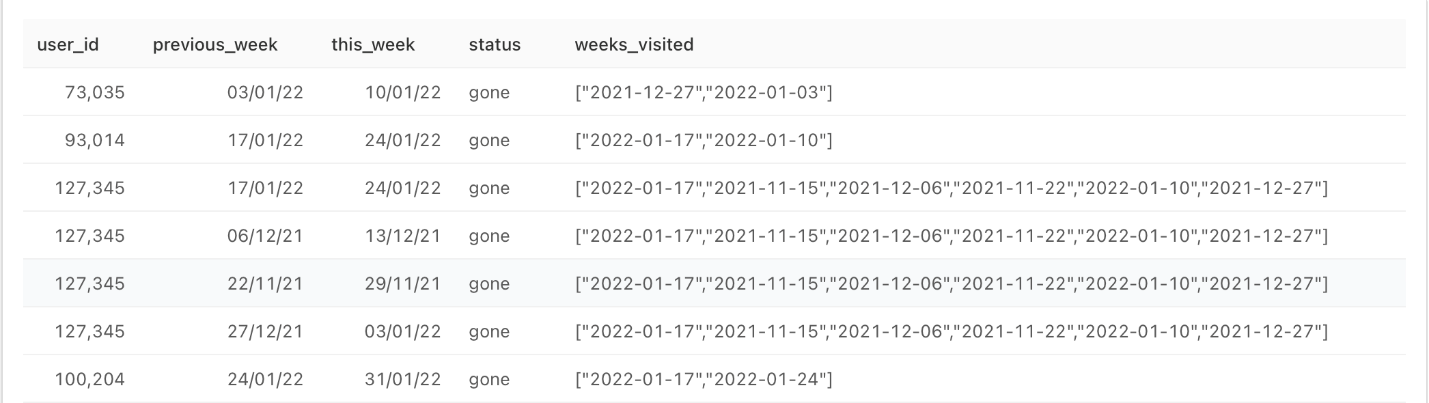
Пример:



Обычно, получить агрегацию, сохраняя изначальную структуру таблицы, можно с помощью оконных в функций, но в ClickHouse это можно сделать с помощью group by, groupUniqArray, arrayJoin.

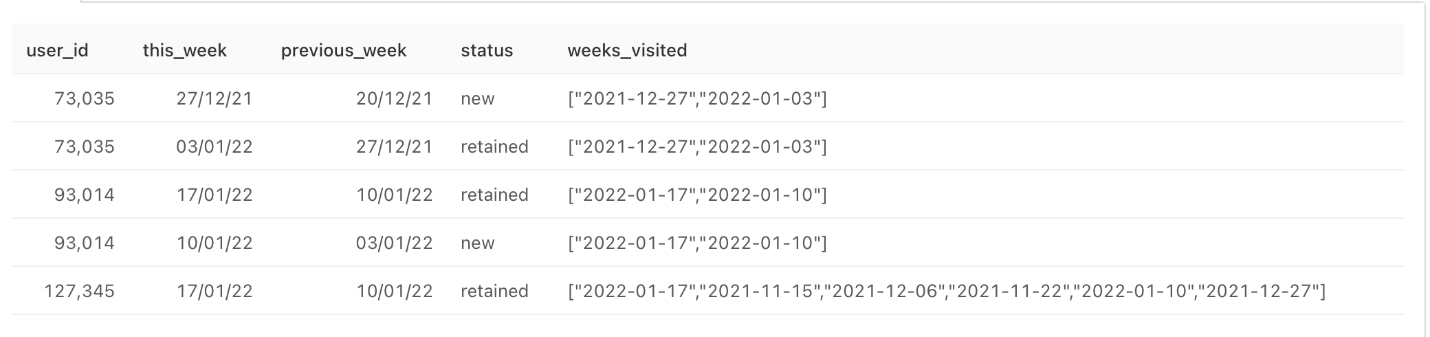
Чтобы теперь получить текущую неделю this\_week, для которой мы производим расчет, достаточно просто воспользоваться функцией addWeeks и добавить одну неделю к previous\_week, а затем проверить наличие this\_week в weeks\_visited с помощью функции has. Если такая дата отсутствует, то это, как раз, будет обозначать то, что пользователь был активен до этого, но на неделе this\_week стал ушедшим.

Пример: пользователь с id 73035 был активен в первую неделю января 2022, но на следующей неделе уже не посещал новостную ленту. Обратите внимание, что здесь для всех записей был добавлен столбец status с константным значением gone. Достигается это в SQL достаточно просто в блоке SELECT - 'gone' as status



Далее таблица сохраняется в подзапрос, произоводится группировка по дате и статусу, считается количество уникальных пользователей и **умножается на -1**, так как эти пользователи являются ушедшими.

Осталось аналогичным образом получить **новых** и **старых** пользователей: для этого пригодится предыдущий запрос, который собирал все даты активностей, с небольшой поправкой на то, что теперь рассматриваемая неделя является не предыдущей, а текущей, то есть пользователь активен на текущей неделе - факт, проверяется активность на предыдущей и присваивается статус, тут его уже нельзя обозначить константой. Достаточно также воспользоваться функциями has и функцией if.



С помощью group by по статусу и текущей неделе также получается посчитать количество уникальных user\_id, но теперь уже положительное, так как эти пользователи пришли в приложение. Для того, чтобы объединить две таблицы можно воспользоваться блоком UNION ALL, который объединяет две таблицы со схожей структурой в одну.