4. Вопросы к данным, OLAP

Думу думать будем

Речь - про аналитику. Что такого можно спросить полезного с точки зрения бизнеса в таком приложении?

Сохраняя конфиденциальность пользовательских данных - почти ничего. Но к счастью в данном случае сам факт аналитики стоит против этого, поэтому ничего страшного.

Вопросы к данным

Блок 1: Затраты на сервер?

Ответы на эти вопросы помогут спланировать расходы на инфраструктуру проекта (S3, в данном случае)

- Какова динамика загрузки файлов пользователями с течением времени?
- Как изменяется общее количество файлов в системе от месяца к месяцу?

Блок 2: Как пользователи используют альбомы?

С введением тегов становится важно отследить, пользуются ли пользователи альбомами или нет. Почему это важно? Потому что такие быстрые фильтры поиска могут впринципе уничтожить необходимость в любых других категоризациях внутри такого простого приложения. Некоторым пользователям проще набрать слово, чем копаться в большом списке альбомов.

Помимо прочего, информация о конкретных альбомах может помочь унифицировать систему для всех: для чего такого забористого не хватает тегов, что пользователи продолжают использовать альбомы? А насколько изменится число файлов в альбомах, если мы введем новую фичу с тегами или что-то поменяем в интерфейсе?

- Какие альбомы содержат наибольшее количество файлов?
- Как изменяется число файлов в альбомах в течение времени?

Блок 3: Дубликаты

Далеко не все пользователи часто с ними сталкиваются. Благодаря наличию информации о дубликатах можно выделить тех, кто наиболее вероятно сталкивается с проблемами. Это может быть полезно - мы можем запросить фидбэк у пользователей, которые часто имеют дело с дубликатами. Почему это важно? Потому что это позволит понять, насколько стоит инвестировать время и ресурсы в поддержку и улучшение алгоритма по поиску дубликатов в системе.

- Каков процент дубликатов у конкретного пользователя?

Блок 4: Теги

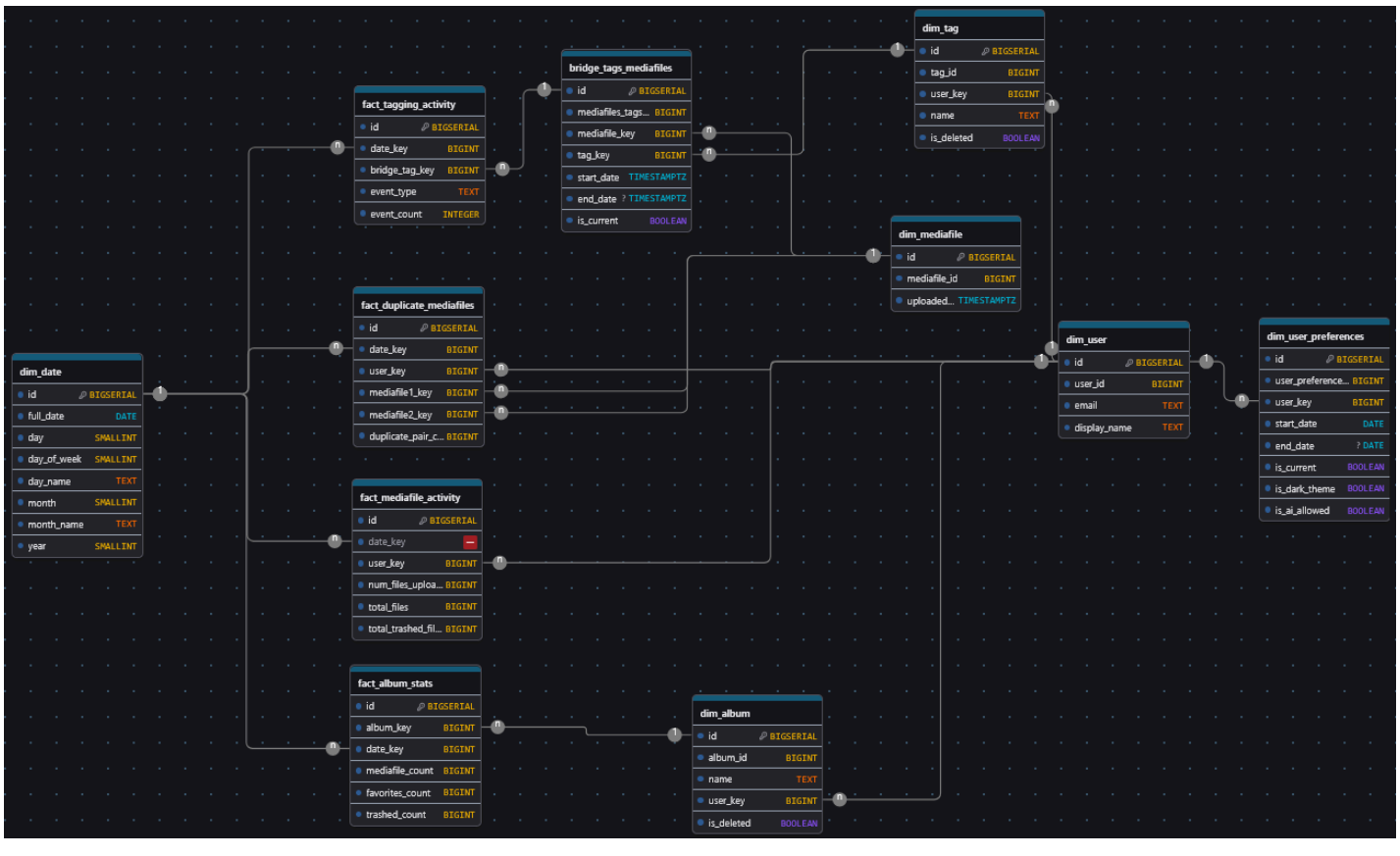
Теги - достаточно новая фича, чтобы вызывать общие опасения с точки зрения бизнеса. Пользователи очень медленно вникают в новые вещи, поэтому важно отследить, для чего именно они применяют эту возможность и в случае если что-то не так - расширить или дополнить общедоступную документацию к приложению.

- События с тегами: как часто их удаляют и добавляют?
- Какие имена тегов самые популярные?

OLAP

Теперь, когда мы определились, что именно нам нужно - пришло время архитектуры.

У меня получилось нечто такое (схема - ниже). Я не назову это идеальным решением, но в контексте требований, представленных к OLAP - наверное оно не самое плохое.



Опять же - схема доступна в интерактивном режиме, загрузите сюда:

https://www.drawdb.app/editor

Вот этот файл (Файл→Импорт→JSON):

> Photos_app_olap.json

Скрипт генерации OLAP доступен в репозитории:

```
./db_creation_scripts/olap/olap_create.sql
```

Много про схему

- dim_date - вещь понятная и простая, храним даты в максимально удобном формате
- dim_user - храним данные пользователей. Полные, потому что как минимум один из наших аналитических вопросов затрагивает данные пользователя целиком
- dim_user_preferences - аналогично описанному ранее, количество настроек имеет свойство расти. A SCD Type 2 в данном случае поможет нам отслеживать динамику изменения этих самых предпочтений в зависимости от бизнес-необходимости. Т.е. если мы вводим новую фичу - это может как-то повлить на юзеров. Например, они решат её отключить через настройки
- dim_mediafile - не содержит некоторых значений, потому что они либо не очень полезны в контексте базы, либо уже представлены в общей статистике в одной из таблиц фактов (ниже)
- dim_tag и dim_album - вполне похожи на себя. Из важного мы тут храним флажок is_deleted. Т.к. у нас нету корзины для тегов или альбомов и они удаляются сразу - неплохо бы для галочки хранить тот факт, что альбома или тега больше не существует, чтобы не учитывать его в аналитике. Т.е. в OLTP удаляется сразу, а ETL это дело отслеживает и мы получаем какую-никакую историю
- bridge_tags_mediafiles - вот тут наверное будет много текста. Поскольку наш OLTP достаточно минималистичен, а мы можем захотеть анализировать теги подробнее (в будущем, на основе результатов начальной аналитики) - предлагается вести историю связей в формате SCD Type 2. Вкупе со следующей таблицей это позволяет вести аналитику достаточно глубокую, сравнительно с фактовой таблицей альбомов.
- fact_tagging_activity - помимо очевидного, содержит два поля: event_type (тип события с тегом) и event_count (больше сервисное поле для подсчета суммы, всегда равно 1). Соль в следующем: добавляем два ивента: “Tag

created” и “Tag removed”. Всё, теперь наша история работает на нас и мы можем отслеживать с большой точностью, как наши пользователи пользуются нашими “высокими” технологиями с механизмом тегов

- fact_duplicate_mediafiles - содержит ссылки на оба медиафайла, образующих пару дубликатов, а также аналогичный подход с полем duplicate_pair_count, всегда равным 1. Позволяет задавать бизнес-вопросы касаемо дубликатов.
- fact_mediafile_activity - аналитика по медиафайлам в общем. Содержит уже три полезных поля:
  - total_files - общее число файлов пользователя на конкретную дату
  - total_files_uploaded - общее число загрузок этого пользователя за сегодня
  - total_trashed_files - общее число файлов в корзине этого пользователя
- fact_album_stats - анализируем альбомы.
  - mediafile_count - общее число файлов в альбоме на конкретную дату у конкретного пользователя
  - favorites_count - общее число файлов в альбоме на конкретную дату, которые также помечены как “любимые” у этого пользователя
  - trashed_count - общее число файлов в корзине пользователя на конкретную дату

Некоторое время назад я упоминал, что мы убрали некоторые поля из dim_mediafile. И вот, они появились тут, в таблице фактов - чтобы оптимизировать аналитику.

Отмечу, что таблицы фактов содержат ссылки на dim_date, т.е. гранулярность фактов у нас получается по дням. Это полезно, потому что позволяет экономить немного на хранимых данных, не собирая возможно ошибочные события. Например: пользователь неверно указывает тег к файлу и сразу же удаляет его, выставляя другой. Если бы мы записывали все события - это было бы бесполезно. Но записывая информацию на уровне дня, вероятность таких проблем снижается - мы получаем конечный результат “теггирования”, а не промежуточный.

Вывод

База OLAP сделана с небольшим запасом под будущее. Помимо вопросов уже упомянутых, мы сможем задавать и другие, не расширяя базу до необходимости. Это позволит проекту перешагнуть этап MVP чуть более бодро. Как только аналитика по базовому функционалу будет готова - будет понятно, что стоит делать по проекту дальше.