7. Бизнес и Power BI

Наконец, пришло время получить ответы на наши бизнес вопросы, при помощи базы OLAP, созданной и наполненной данными.

Напомню описанное ранее:

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?

Power Bi

Для получения ответов на эти вопросы я собрал отчет в Power BI. Он уже содержит данные, поэтому ничего извлекать из OLAP не нужно. Файл отчета находится в репозитории.

В целом отчет в Power BI отвечает на вопросы, описанные чуть выше.

Тут я разобью блоки с вопросами по страницам:

1. Страница 1 - блок 1
2. Страница 2 - блоки 2 и 3
3. Страница 3 - блок 4

Бизнес-вопросы

А в оставшейся части этой заметки пожалуй стоит закрыть последний пункт требований:

Задать вопросы и получить ответы с помощью SQL

Все запросы ниже выполняем на стороне базы OLAP, чтобы не “прыгать” между серверами.

Блок 1

- Какова динамика загрузки файлов пользователями с течением времени?

В OLTP:

```
SELECT u.display_name            AS user_name,
       DATE(mf.uploaded_datetime) AS upload_date,
       COUNT(mf.id)               AS files_uploaded_count
FROM oltp_mediafiles mf
         JOIN
     oltp_users u ON mf.user_id = u.id
WHERE DATE(mf.uploaded_datetime) >= '2025-05-01'
GROUP BY u.display_name,
         DATE(mf.uploaded_datetime)
ORDER BY upload_date,
         user_name;
```

В OLAP:

```
SELECT du.display_name             AS user_name,
       dd.full_date                AS upload_date,
       fma.num_files_uploaded_today AS files_uploaded_count
FROM fact_mediafile_activity fma
         JOIN
     dim_user du ON fma.user_key = du.id
         JOIN
     dim_date dd ON fma.date_key = dd.id
WHERE dd.full_date >= '2025-05-01'
ORDER BY upload_date,
         user_name;
```

- Как изменяется общее количество файлов в системе от месяца к месяцу?

В OLTP:

```
SELECT DATE_TRUNC('month', mf.uploaded_datetime) AS month_start,
       COUNT(mf.id)                              AS total_files_at_month_end
FROM oltp_mediafiles mf
WHERE mf.trashed_datetime IS NULL
GROUP BY DATE_TRUNC('month', mf.uploaded_datetime)
ORDER BY month_start;
```

В OLAP:

```
SELECT DATE_TRUNC('month', dd.full_date) AS month_start,
       MAX(fma.total_files)              AS total_files_at_month_end
FROM fact_mediafile_activity fma
         JOIN
     dim_date dd ON fma.date_key = dd.id
```

```
GROUP BY DATE_TRUNC('month', dd.full_date)
ORDER BY month_start;
```

Примечание: Запрос к OLTP будет менее точен, т.к. информация в OLAP хранится с историей.

Блок 2

- Какие альбомы содержат наибольшее количество файлов?

В OLTP:

```
SELECT a.name                 AS album_name,
       COUNT(mfa.mediafile_id) AS mediafile_count
FROM oltp_albums a
         JOIN
     oltp_mediafiles_albums mfa ON a.id = mfa.album_id
         JOIN
     oltp_mediafiles mf ON mfa.mediafile_id = mf.id
WHERE mf.trashed_datetime IS NULL
GROUP BY a.name
ORDER BY mediafile_count DESC
LIMIT 10; -- Top-10 albums by mediafile count
```

В OLAP:

```
SELECT da.name                 AS album_name,
       SUM(fas.mediafile_count) AS total_mediafile_count
FROM fact_album_stats fas
         JOIN
     dim_album da ON fas.album_key = da.id
         JOIN
     dim_date dd ON fas.date_key = dd.id
WHERE dd.full_date = CURRENT_DATE
  AND da.is_deleted = FALSE
GROUP BY da.name
ORDER BY total_mediafile_count DESC
LIMIT 10; -- Top-10 albums by mediafile count
```

Ситуация в OLAP чуть лучше - ETL уничтожает дубликаты в CSV-файлах, предоставленных изначально. Также мы объединяем имена альбомов, т.е. одно и то же название альбома у разных пользователей выливается в сумму в обоих альбомах. Почему? Потому что мы анализируем наиболее популярные имена.

- Как изменяется число файлов в альбомах в течение времени?

В OLTP:

```
SELECT a.name                    AS album_name,
       DATE(mf.uploaded_datetime) AS snapshot_date,
       COUNT(DISTINCT mf.id)      AS mediafile_count_on_date
FROM oltp_albums a
         JOIN
     oltp_mediafiles_albums mfa ON a.id = mfa.album_id
         JOIN
     oltp_mediafiles mf ON mfa.mediafile_id = mf.id
WHERE mf.trashed_datetime IS NULL
GROUP BY a.name,
         DATE(mf.uploaded_datetime)
ORDER BY album_name,
         snapshot_date;
```

В OLAP:

```
SELECT da.name      AS album_name,
       dd.full_date AS snapshot_date,
       fas.mediafile_count
FROM fact_album_stats fas
         JOIN
     dim_album da ON fas.album_key = da.id
         JOIN
     dim_date dd ON fas.date_key = dd.id
WHERE da.is_deleted = FALSE
ORDER BY album_name,
         snapshot_date;
```

В данном случае результат OLTP опять неточен, т.к. мы храним историю в OLAP. Вполне возможно, что данные в OLTP могут быть удалены (к примеру, я уже упоминал флажок is_delete для альбомов).

Блок 3

- Каков процент дубликатов у конкретного пользователя?

В OLTP:

```
SELECT u.display_name AS user_name,
       COUNT(mf.id)   AS total_active_files,       -- total files that are not trashed
       COUNT(
           DISTINCT CASE
                        WHEN od.mediafile_1_id IS NOT NULL
                            OR od.mediafile_2_id IS NOT NULL
                            THEN mf.id
           END)       AS files_in_duplicate_pairs, -- files that are part of duplicates
       (CAST
        (COUNT(DISTINCT CASE
                        WHEN od.mediafile_1_id IS NOT NULL
                            OR od.mediafile_2_id IS NOT NULL
                            THEN mf.id
                END)
            AS NUMERIC) * 100.0 / COUNT(mf.id)) AS percentage_duplicates -- percentage of files that are part of duplicates
FROM oltp_users u
         JOIN
     oltp_mediafiles mf ON u.id = mf.user_id
         LEFT JOIN
     oltp_duplicates od ON mf.id = od.mediafile_1_id OR mf.id = od.mediafile_2_id
WHERE mf.trashed_datetime IS NULL
  AND u.id = 1 -- Insert real user ID here
GROUP BY u.display_name;
```

В OLAP:

```
SELECT
    du.display_name AS user_name,
    dd.full_date AS snapshot_date,
    fma.total_files AS total_active_files,
    -- Duplicate files in pairs for the user on the specific date
    COALESCE(duplicate_files.files_in_duplicate_pairs, 0) AS files_in_duplicate_pairs,
    -- Duplicate percentage
    (CAST(COALESCE(duplicate_files.files_in_duplicate_pairs, 0) AS NUMERIC) * 100.0 / NULLIF(fma.total_files, 0)) AS perc
FROM
    fact_mediafile_activity fma
```

```
    JOIN
  dim_user du ON fma.user_key = du.id
    JOIN
  dim_date dd ON fma.date_key = dd.id
    LEFT JOIN
    -- Subquery to get the count of files in duplicate pairs for the user on the specific date
    (
      SELECT
        fdf.date_key,
        fdf.user_key,
        COUNT(DISTINCT fdf.mediafile1_key) + COUNT(DISTINCT fdf.mediafile2_key) AS files_in_duplicate_pairs
      FROM
        fact_duplicate_mediafiles fdf
      GROUP BY
        fdf.date_key,
        fdf.user_key
    ) duplicate_files ON fma.date_key = duplicate_files.date_key AND fma.user_key = duplicate_files.user_key
WHERE
  dd.full_date = CURRENT_DATE -- For today's date
 AND du.id = 1 -- Insert real user ID here
GROUP BY
  du.display_name,
  dd.full_date,
  fma.total_files,
  duplicate_files.files_in_duplicate_pairs;
```

Тут мы делим количество уникальных файлов в дубликатах на число активных (не в корзине) файлов пользователя. Запрос к OLAP более точен и может быть выполнен на конкретную дату.

Блок 4

- События с тегами: как часто их удаляют и добавляют?

В OLTP:

В OLTP это в целом неподъемная задача. Теоретически можно было бы сделать топ-10 наиболее активных тегов на основе updated_datetime, но это скорее к следующему запросу.

В OLAP:

```
SELECT dd.full_date        AS event_date,
       fta.event_type,
       SUM(fta.event_count) AS total_events
FROM fact_tagging_activity fta
       JOIN
     dim_date dd ON fta.date_key = dd.id
GROUP BY dd.full_date,
       fta.event_type
ORDER BY event_date,
       fta.event_type;
```

- Какие имена тегов самые популярные?

В OLTP:

```
SELECT t.name                      AS tag_name,
       COUNT(DISTINCT mfa.mediafile_id) AS mediafile_count -- distinct files with the tag
FROM oltp_tags t
       JOIN
     oltp_mediafiles_tags mfa ON t.id = mfa.tag_id
       JOIN
```

```
    oltp_mediafiles mf ON mfa.mediafile_id = mf.id
WHERE mf.trashed_datetime IS NULL -- not trashed
GROUP BY t.name
ORDER BY mediafile_count DESC
LIMIT 10; -- Top-10 tags by mediafile count
```

B OLAP:

```
SELECT dt.name                        AS tag_name,
       COUNT(DISTINCT fta.bridge_tag_key) AS active_tag_links_count
FROM fact_tagging_activity fta
         JOIN
     bridge_tags_mediafiles btm ON fta.bridge_tag_key = btm.id
         JOIN
     dim_tag dt ON btm.tag_key = dt.id
         JOIN
     dim_date dd ON fta.date_key = dd.id
WHERE dd.full_date = (SELECT MAX(full_date) FROM dim_date WHERE full_date <= CURRENT_DATE)
  AND dt.is_deleted = FALSE
GROUP BY dt.name
ORDER BY active_tag_links_count DESC
LIMIT 10; -- Top-10 tags by mediafile count
```

Wrapping Up

Я признаю, что запросы для OLAP в конечном итоге не выглядят менее “массивно”, сравнительно с их собратьями для OLTP. Однако есть несколько моментов, которые делают OLAP-аналитику более выгодным решением в конкретном (нашем) случае:

- Нагрузка на сервер. Люди в наше время хранят супер много фото и видео, раз уж подписки на крупные сервисы за 2 ТБ начали стоить всего 10$. Аналитика и работа приложения в одной базе приводят к дополнительной, нежелательной нагрузке.
- История. То, что не очень важно для работы приложения “здесь и сейчас” - обретает смысл для аналитики в “лонг-ране”. Как видно из некоторых запросов - мы можем не только спросить про “здесь и сейчас”. Мы можем больше - заглянуть в прошлое.
- Агрегация. Несмотря на массивность запросов к обоим базам, в OLAP у нас всё равно есть преимущество - заранее рассчитанные метрики в таблицах фактов. Да, количество JOIN всё ещё оставляет желать лучшего. Но у нас в руках уже лежит нужное нам число, не требуется пересчитывать его заново.

Таким образом, реализация OLAP в таких условиях всё ещё полезна и нужна. И да, я признаю - это заняло очень много времени и сил, чтобы совместить контекст моего псевдо-приложения и условия задания. Но это как минимум было полезно и продуктивно, спасибо вам за эту возможность :)