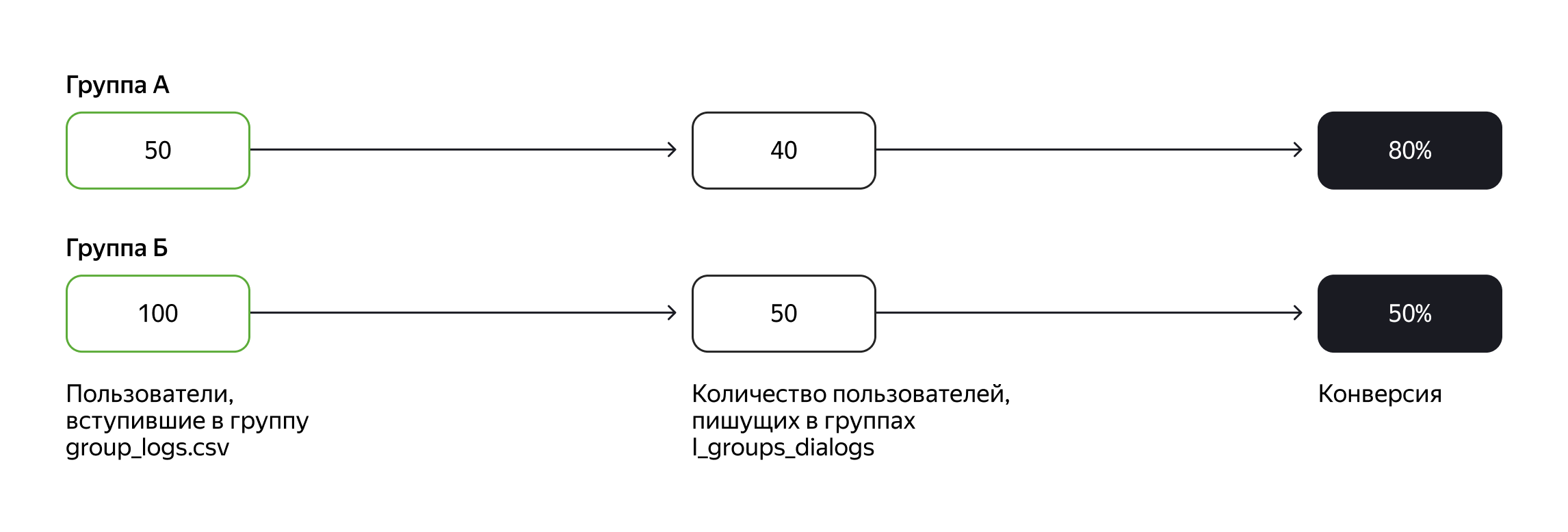
**Проектная работа по аналитическим базам данных**

Вы построили аналитическое хранилище и протестировали его в деле: выяснили возраст активной аудитории. Теперь вам предстоит пойти дальше — развить свою БД и ответить на следующий вопрос бизнеса. В рамках проекта вам нужно будет расширить модель данных, проанализировать новую информацию и помочь маркетологам дать эффективную рекламу соцсети в интернете.

**Описание задачи**

Чтобы привлечь новых пользователей, маркетологи хотят разместить на сторонних сайтах рекламу сообществ с высокой активностью. Вам нужно определить группы, в которых начала общаться большая часть их участников. В терминологии маркетинга их бы назвали пабликами с высокой конверсией в первое сообщение.

На схеме ниже видно, что в группе А конверсия выше, чем в Б. Хотя в группе А сейчас общается только 40 пользователей соцсети, а в Б — 50, доля активных в А выше, ведь в ней всего 50 человек. В то время как в группе Б сообщения написали уже 50 участников, но это лишь половина от общего количества — 100. Значит, если в обе группы вступит одинаковое число людей, эффективнее сработает сообщество А, потому что оно лучше вовлекает своих участников. Получается, что для рекламы соцсети стоит выбрать группу А и другие паблики с высокой конверсией. Ваша задача — выявить и перечислить маркетологам такие сообщества.

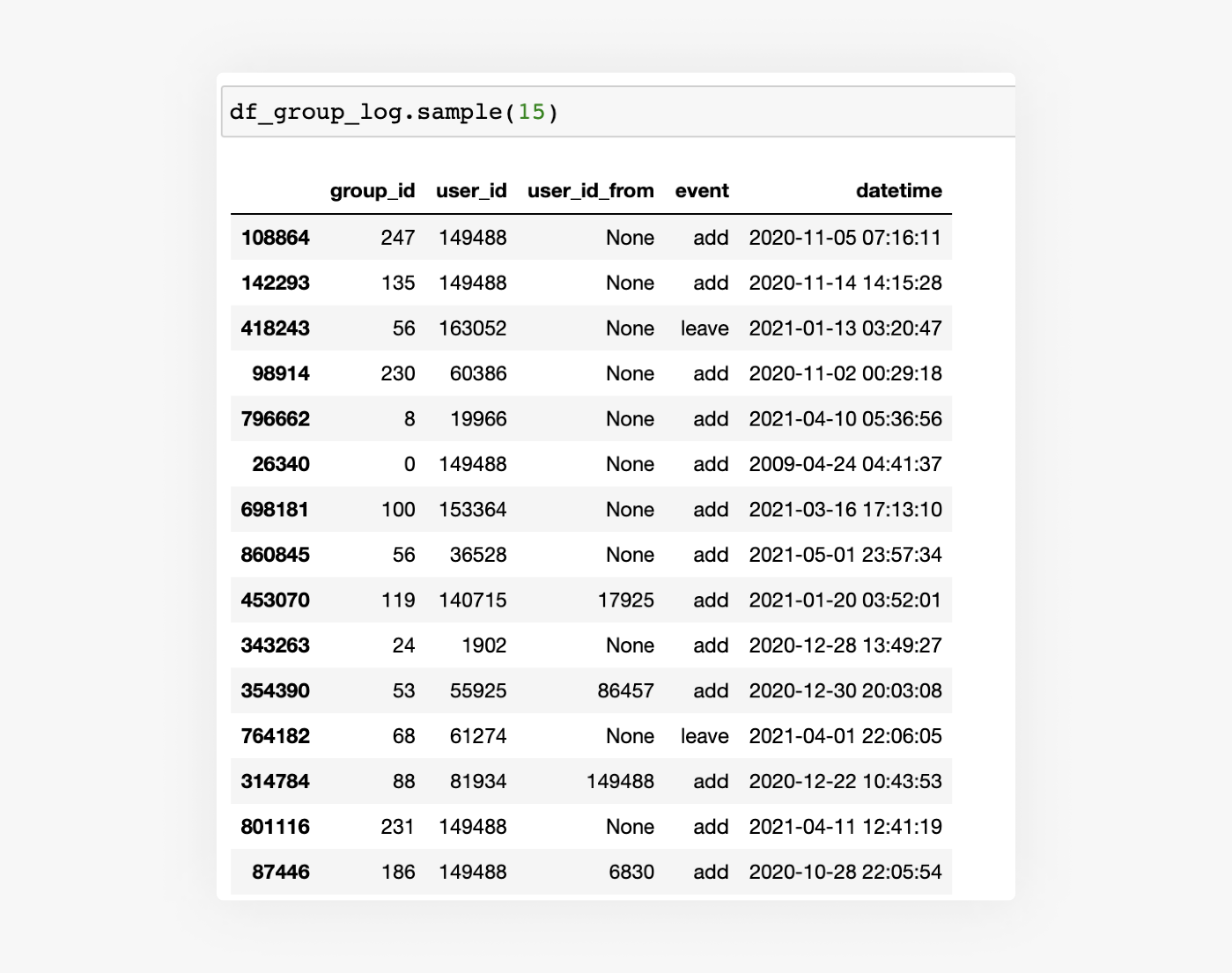


Необходимо:

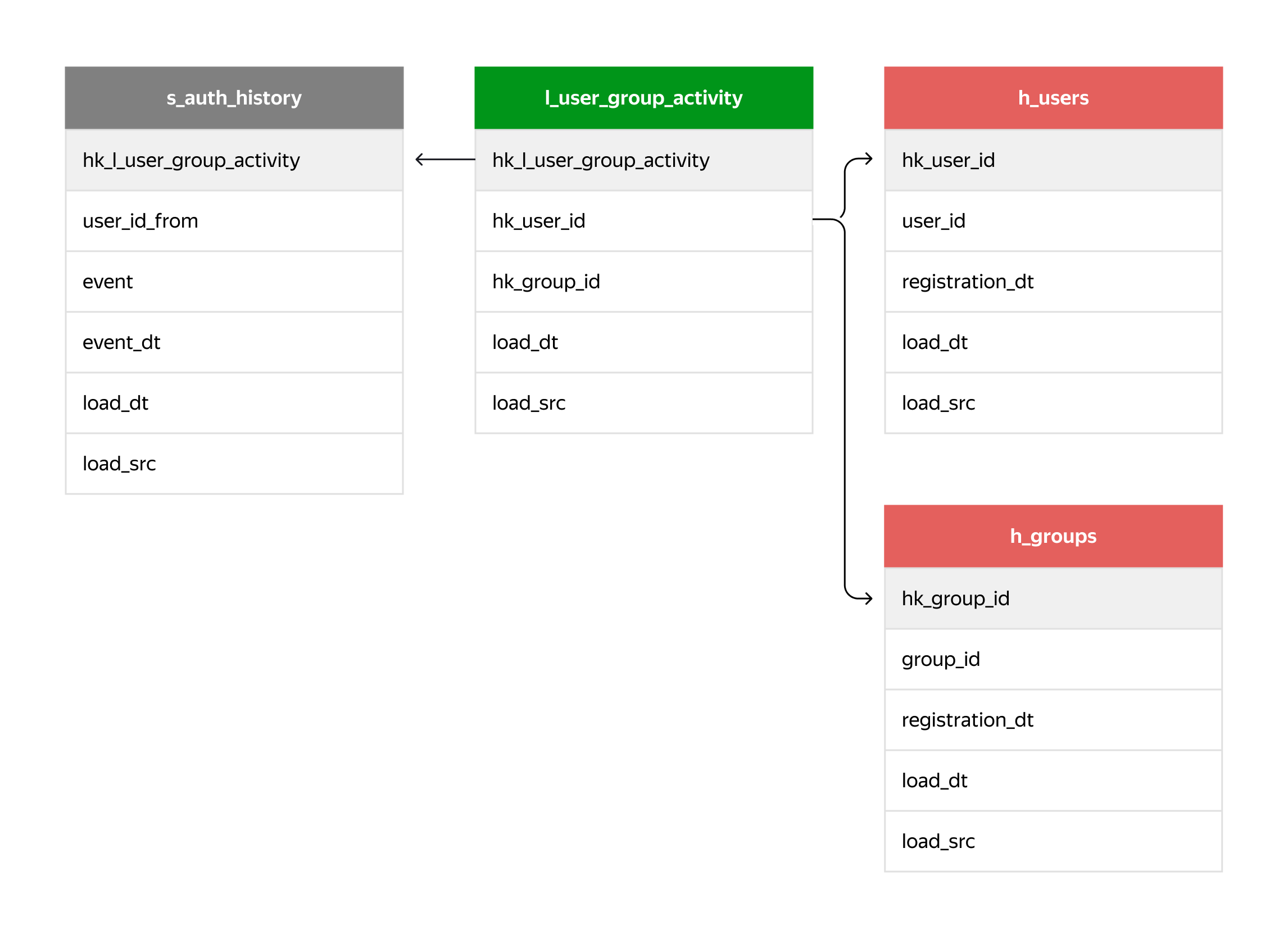
1. Перенести из S3 в staging-слой новые данные о входе и выходе пользователей из групп — файл group\_log.csv.

Лог работы групп group\_log.csv содержит:

1. group\_id — уникальный идентификатор группы;
2. user\_id — уникальный идентификатор пользователя;
3. user\_id\_from — поле для отметки о том, что пользователь не сам вступил в группу, а его добавил другой участник. Если пользователя пригласил в группу кто-то другой, поле будет непустым.
4. event — действие, которое совершено пользователем user\_id. Возможны следующие варианты:
   * create — пользователь создал группу;
   * add — пользователь user\_id вступил сам или был добавлен в группу;
   * leave — пользователь user\_id покинул группу.
5. datetime — время совершения event.



1. Создать в слое постоянного хранения таблицы для новых данных. Например, по схеме:



1. Перенести новые данные из staging-области в слой DDS.
2. Рассчитать конверсионные показатели для десяти самых старых групп:
   * количество пользователей, вступивших в группу, — cnt\_added\_users;
   * количество участников группы, которые написали хотя бы одно сообщение, — cnt\_users\_in\_group\_with\_messages;
   * конверсию в первое сообщение из вступления в группу — group\_conversion.

**Инструкция по выполнению проекта**

**Шаг 1. Загрузить данные из S3**

Напишите DAG в Airflow, чтобы подключиться к бакету sprint6 в S3 и выгрузить файл group\_log.csv в папку data.

**Шаг 2. Создать таблицу group\_log в Vertica**

Заведите таблицу group\_log в схеме \*\_STAGING с учётом формата данных в скачанном файле.

**Шаг 3. Загрузить данные в Vertica**

Реализуйте DAG, который будет считывать из папки data файл group\_log.csv и загружать его в созданную таблицу group\_log в схеме \*\_\_STAGING.

Подсказка

**Шаг 4. Создать таблицу связи**

Добавьте в схему \*\_\_DWH таблицу связи l\_user\_group\_activity с такими полями:

* hk\_l\_user\_group\_activity — основной ключ типа INT;
* hk\_user\_id — внешний ключ типа INT, который связан с основным ключом хаба MY\_\_DWH.h\_users;
* hk\_group\_id — внешний ключ типа INT, который связан с основным ключом хаба MY\_\_DWH.h\_groups;
* load\_dt — временная отметка типа DATETIME о том, когда были загружены данные;
* load\_src — данные об источнике типа VARCHAR(20).

**Шаг 5. Создать скрипты миграции в таблицу связи**

Напишите SQL-скрипт миграции данных из \*\_\_STAGING.group\_log в линк l\_user\_group\_activity.

Вставьте в таблицу l\_user\_group\_activity данные с помощью INSERT и запроса из таблиц \*\_\_STAGING.group\_log . Вам потребуется сделать JOIN хэш-ключей из таблицы хабов пользователей \*\_\_DWH.h\_users и таблицы групп \*\_\_DWH.h\_groups.

Заготовка кода:

INSERT INTO MY\_\_DWH.l\_user\_group\_activity(hk\_l\_user\_group\_activity, hk\_user\_id,hk\_group\_id,load\_dt,load\_src)

select distinct

\*\*\*Ваш код здесь\*\*\*

from MY\_\_STAGING.group\_log as gl

left join MY\_\_DWH.h\_users \*\*\*Ваш код здесь\*\*\*

left join MY\_\_DWH.h\_groups \*\*\*Ваш код здесь\*\*\*

;

**Шаг 6. Создать и наполнить сателлит**

Добавьте в схему \*\_\_DWH сателлит s\_auth\_history и наполните его данными из \*\_\_STAGING.group\_log и необходимых таблиц в хранилище через скрипт миграции.

Таблицу-сателлит создайте с набором полей:

* hk\_l\_user\_group\_activity — внешний ключ к ранее созданной таблице связей типа INT. Напомним, в сателлитах нет первичных ключей.
* user\_id\_from — идентификатор того пользователя, который добавил в группу другого. Если новый участник вступил в сообщество сам, это поле пустое. Задайте атрибуту user\_id\_from тип INT и наполните его из исходных загруженных данных \*\_\_STAGING.group\_log.
* event — событие пользователя в группе;
* event\_dt — дата и время, когда совершилось событие;
* load\_dt и load\_src — знакомые вам технические поля.

Для наполнения сателлита используйте основной источник загруженных данных \*\_\_STAGING.group\_log в связке с необходимыми таблицами:

* MY\_\_DWH.h\_groups — для связи по хеш-ключу группы данных из MY\_\_DWH.l\_user\_group\_activity;
* MY\_\_DWH.h\_users — для связи по хэш-ключу пользователя данных из MY\_\_DWH.l\_user\_group\_activity.

Заготовка кода вставки данных:

INSERT INTO MY\_\_DWH.s\_auth\_history(hk\_l\_user\_group\_activity, user\_id\_from,event,event\_dt,load\_dt,load\_src)

select \*\*\*Ваш Код здесь\*\*\*

from MY\_\_STAGING.group\_log as gl

left join MY\_\_DWH.h\_groups as hg on gl.group\_id = hg.group\_id

left join MY\_\_DWH.h\_users as hu on gl.user\_id = hu.user\_id

left join MY\_\_DWH.l\_user\_group\_activity as luga on hg.hk\_group\_id = luga.hk\_group\_id and hu.hk\_user\_id = luga.hk\_user\_id

;

**Шаги 7. Подготовить CTE для ответов бизнесу**

Вы мигрировали данные и разложили их в реализованной модели данных для оптимального хранения. Теперь вам нужно реализовать две временные таблицы (Common Table Expression — CTE) для оптимального обращения к данным. К ним вы обратитесь, чтобы получить нужные ответы о доле пользователей, которые пишут в группах, относительно пользователей, которые просто вступили в группу.

**Шаг 7.1. Подготовить CTE user\_group\_messages**

Начните с таблицы, которая поможет с анализом доли пользователей, которые активны в группах (**user\_group\_messages**) — пишут туда что-то.

1. Напишите SQL-запрос, который создаст CTE user\_group\_messages, а внутри:
   * реализуйте вложенный запрос, который будет возвращать хэш-ключ каждой группы hk\_group\_id и количество пользователей в группе, которые написали хотя бы раз — cnt\_users\_in\_group\_with\_messages
   * не забудьте, что нужно учитывать только уникальных пользователей
2. Запустите запрос к данным из user\_group\_messages для подтверждения корректности рассчётов.

Заготовка кода:

with user\_group\_messages as (

select

\*\*\*Ваш Код здесь\*\*\*

)

select hk\_group\_id,

cnt\_users\_in\_group\_with\_messages

from user\_group\_messages

order by cnt\_users\_in\_group\_with\_messages

limit 10

;

**Шаг 7.2. Подготовить CTE user\_group\_log**

Создайте временную таблицу, из которой сможете получать количество пользователей, вступивших в группы.

1. Напишите SQL-запрос, который создаст CTE user\_group\_log, а внутри:
   * реализуйте вложенный запрос, который вернёт хэш-ключ каждой группы hk\_group\_id и количество пользователей в группе, которые в неё просто вступили cnt\_added\_users
   * не забудьте учесть только уникальных пользоватей
   * возьмите только 10 самых ранних созданных групп (таблица MY\_\_DWH.h\_groups, самые ранние группы по значению в поле registration\_dt)
   * для получения всех событий добавления в группу event = 'add’ удобно использовать таблицу MY\_\_DWH.s\_auth\_history
2. Запустите запрос к данным из user\_group\_log для подтверждения корректности рассчётов.

Заготовка кода:

with user\_group\_log as (

select \*\*\*Ваш Код здесь\*\*\*

)

select hk\_group\_id

,cnt\_added\_users

from user\_group\_messages

order by cnt\_added\_users

limit 10

;

**Шаг 7.3. Написать запрос и ответить на вопрос бизнеса**

Временные таблицы вы написали, теперь можно обратиться к ним, чтобы вывести нужные бизнесу ответы.

1. Напишите SQL-запрос, который выведет по десяти самым старым группам:
   * Хэш-ключ группы hk\_group\_id
   * Количество новых пользователей группы (event = add). Назовите поле cnt\_added\_users
   * Количество пользователей группы, которые написали хотя бы одно сообщение. Назовите поле cnt\_users\_in\_group\_with\_messages
   * Долю пользователей группы, которые начали общаться. Назовите выводимое поле group\_conversion
2. Отсортируйте результаты по убыванию значений поля group\_conversion.