Немного опишу свои шаги по тестовому:

1. SQL

Сначала было сложно приступить к заданию по SQL, но потом посидел, помозговал и начал декомпозировать задачу. В целом задание не очень сложное, реализация не прям идеальна, но главное, что показал все этапы - от создания, до наполнения и вывода нужных нам данных.

Проблемы были с выбором правильных типов, чтоб БД занимала меньше места. Далее понял ошибку, что нельзя генерировать разные "id" в таблицах и надо идти от того, что мы сначала сгенерировали "id" и к нему уже генерируем другие данные. Оставил разный возраст на "client\_id", т.к. предположил, что могут быть аккаунты, которые используют несколько человек, наверно это неправильно, но другую логику применил в питоне. Также были всякие нюансы по понимаю группировок и вывода нужных данных. Можно было бы добавить ещё "constraint'ы", но я не стал, т.к. время уже решил не терять на этом этапе, могу в целом голосом это проговорить (об этом очевидном моменте подумал перед отправкой задания).

В целом SQL корректировал раза 4-5.

1. Python

С питоном было проще, потому что в целом скелет того, как это должно быть на выходе уже был. Тут я уже исправил вариант, чтоб на один "client\_id" был один возраст и пол. Также предусмотрел генерацию данных для клиентов по уникальному полю "client\_id", чтоб не было дублированных "id" с разными данными. Выбирал как предоставить данные лучше будет - в своднике или же в переменную датафрейм запихать. При попытке построить сводник упёрся в нехватку памяти, когда накидываешь группировку по 6-7 полям разным, то там объём таблицы резко возрастает. В итоге сделал, что 1 таблицу общую собираю, а во второй группирую данные по первой таблице и потом их соединяю по полю. В целом всё

1. Docker

Сходу недооценил объём по docker'у. У меня в целом имелся опыт с виртуализацией (OpenStack, VMware, VB). Начал разбираться с нуля тут, потому что никогда докер не поднимал самостоятельно. Казалось бы уже очевидные моменты вызывали очень много вопросов и приходилось в некоторых моментах погружаться до видео с индусами)) откуда эти ребята вообще столько интересного знают)

docker-compose написал, стал разворачивать сервисы и столкнулся с проблемой, что у меня nifi не видит базу PostgreSQL, начал разбираться и вообще закопался. Видимо я не пробросил сетку между ними, но решил сдать работу, т.к. время уже поджимало. Хотя, для меня не очень понятно, почему сервисы, которые поднимаются в докере не имеют сетевой связности между контейнерами, думал что виртуальная сеть между ними автоматически создаётся. Я накидал очень простую схему, она в целом неверна, но понимаю данный процесс в голове, но реализовать не смог, т.к. упёрся в связность.

Процесс примерно:

Создаём бакет в минио -> в него закидываем нужный нам файл -> идём в nifi и создаём процесс "GetFile", который должен забирать из стораджа minio -> закидываем .jar файлик на nifi container для коннекта -> создаём таблицу в PostgreSQL -> добавляем процесс PutDatabaseRecord (описываем в нём креды для базы в DBCP коннекторе) -> Funnel -> выполняем процесс. Общими мазками думаю логика примерно такая.

Можно ещё "вылизывать" код сидеть, но надо вовремя остановиться (кмк это очень важный скилл :) ).

В целом задание мне очень понравилось, смог ощутить себя отчасти архитектором, дата инженером и поработать с продуктами, которые было интересно изучить. Спасибо большое за возможность! :)