Решение команды № 21 GlowByte Autumn Hack 2022

Краткое резюме:

Мы получили большое удовольствие, приняв участие в GlowByte Autumn Hack 2022. Задача была очень интересной, по ходу погружения она для нас открывалась с новых сторон. При работе над ее решением нам пришлось изучить много нового. К сожалению, нам не удалось довести решение до конца в силу разных обстоятельств (нехватки времени по личным и рабочим причинам, к тому же мы в какой-то момент слишком погрузились в детали и потратили много времени на решение этих вопросов и выбор инструментов). Тем не менее, мы бы хотели описать проделанную нами работу.

Описание решения:

Для подключения к базам данных мы используем библиотеки psycopg2 и sqlalchemy. Мы создаем подключение и отправляем туда запрос на создание таблиц. SQL -запрос, создающий таблицы расположен в модуле info.

Далее мы подключаемся к удаленным ftp серверам с помощью защищенного соединения. Нам не удалось найти другое решение с безопасным соединением, кроме как создания локальных csv файлов, с которыми мы в дальнейшем будем работать.

С помощью метода increment_load_new_data класса FactRiders мы загружаем данные в нашу целевую базу. Мы передаем туда аргументы: входящее подключение, исходящее подключение, источник чтения, а также названия таблиц и параметров к ним, также целевой дата фрейм. В этом методе реализована следующая идея инкрементальной загрузки: мы создаем sourse_upd_df pandas dataframe и читаем туда все данные из источника. Создаем отдельный target dataframe и туда считываем данные, уже загруженные базы. И построчно их сравниваем. При этом если мы находим отличия возможны следующие действия: внесение изменений новой строкой, либо же в случае модификации данных мы должны внести строку новой строкой, но предыдущему значению поменять значение в поле end_dt. К сожалению, до конца эту идею мы не успели реализовать, но аналогичный sql запрос при подаче в терминале непосредственно в базу справлялся с этой задачей. Также на этом этапе мы отслеживаем удаленные строки deleted. По задумке, если такие строки находились мы должны были поднимать deleted_flg или же менять поле end_dt.

То же самое происходит с csv файлом. Данные из него мы загружаем в датафрейм, трансформируем (переименовываем колонки, объединяем дату и время в один столбец итд), сохраняем в таком формате, который требуется для fact_payments. После этого мы задаем параметры для целевой таблицы и передаем данные в тот же метод increment_load_new_data.

К сожалению, мы не успели реализовать для всех целевых таблиц, но в целом решение использовать один метод для инкремнтальной загрузки, меняя только параметры -аргументы, кажется нам интересным.

Если бы у нас была еще пара дней мы бы реализовали следующее. Данные из xml файлов у нас собираются в один csv файл. Из него мы читаем в дата фрейм. Также вложенным SQL запросом мы выгружаем только те события, по которым есть данные о завершении или отмене поездки. При этом мы отправляем SQL запрос в rides чтобы получить данные об этих поездках. Все это собираем в один дата фрейм и отправляем в целевую базу.

Заключение:

Мы получили большое удовольствие от участия в GlowByte Autumn Hack 2022. Задание было комплексным и в ходе его выполнения мы узнали много нового, в том числе прочитали много интересного о том, что мы пока не использовали в работе, например DAG Airflow, SQL server Batch mode итд. К сожалению, нам не хватило времени решить кейс до конца, но возможно мы вернемся к нему в свободное время.