Советы и рекомендации по проекту

**1. DDL**

С DDL все ОК, но для витрины следует добавить UNIQUE для сочетания курьер — год — месяц

**2. заполнение stg**

В спринте вы грузили данные из источников не парся JSON, потому что он был сложной структуры и не соответствовал структуре таблицы

В проекте же JSON такой структуры, что из него можно сразу же получить таблицу, можно делать и так (в рекомендации к проекту сказано грузить JSON целиком и это также правильно. Ниже я просто предлагаю еще один из возможных вариантов, который может показаться тебе удобнее)

Для загрузки можно воспользоваться методом execute\_values библиотеки psycopg2

Код будет выглядеть приблизительно так:

```python

# получаем название колонок

columns = ','.join([i for i in response[0]])

# SQL запрос на вставку

sql = f"INSERT INTO {pg\_schema}.{pg\_table\_source} (columns) VALUES %s"

# делаем список списков из значений словаря. response - результат get-запроса в JSON формате

values = [[value for value in response[i].values()] for i in range(len(response))]

execute\_values(cursor, sql, values)

```

**3. логика цикла while с длиной 21**

Не совсем понятно, почему ты решаешь так делать? А если останется меньше 20-ти записей и API выдаст, например, 15 значений? Следует останавливать цикл, когда запрос придет пустым, т.е. длина будет равна 0

**4. использование date.today()**

date.today()- это текущая физическая дата, а для того, чтобы запрос мог отрабатывать используя ту дату, за которую ты его запускаешь, следует использовать шаблоны Airflow (например {{ds}} - дату запуска дага)

https://airflow.apache.org/docs/apache-airflow/stable/templates-ref.html

**5. заполнение DDS**

Здорово, что используешь постгресоператор для данных действий, но загрузка данных идет неоптимально

Условиями уникального ключа ты не допускаешь дублей. Но при этом при каждом запуске дага ты берешь все данные из stg. Так ETL процессы не строят, они должны быть инкрементальными, т.е. обрабатывать только новую информацию, которая не было обработана ранее (и внутри deliveries скрипт для курьеров)

Ниже оставлю коммент, как, на мой взгляд, стоит выстраивать ETL процесс

**6. заполнение витрины**

Та же ситуация, перезапись - код с полным очищением и загрузкой всех данных может быть рабочим вариантом, но точно неоптимальным. А на большх данных просто может не хватит ресурсов

**А теперь к общим рекомендациям по построению ETL процесса**

Инкрементальный ETL процесс проще всего построить, логически понять и мониторить, если он построен на датах. Т.е. процесс стоит на расписании и забирает данные за определенный временной промежуток

Для этого можно поставит расписание не раз в 15 минут, а, например, раз в день

1. При заполнении stg слоя мы можем:

-брать данные за конкретный временной промежуток, API позволяет делать сортировку по датам

-дату брать из контекста Airflow ({{ds}}). Это позволит нам перезапускать ETL процесс за любые временные промежутки не залезая в сам код. Если, конечно, соблюдать условие идемпотентности

-к сожалению, в информации о курьерах нет временных отметок. В реальной рабочей ситуации следовало бы просто обратиться к разработчикам, чтобы добавили поле. В текущем проекте можно либо сделать вспомогательную таблицу, либо каждый день перезаписывать данные. Не думаю, что количество курьеров может дорасти до такого числа, что перезаписать таблицу станет прямо накладно:))

2. При заполнении dds и cdm у нас идет перенос данных исключительно в SQL. Значит можно обойтись только SQL запросами, не строя сложную логику на питоне. Также с помощью контекста Airflow грузить данные за определенный временной промежуток

Такой ETL процесс будет предельно понятен. Один запуск означает загрузку данных за вчерашний день

И его легко мониторить и исправлять. Если вдруг в какой-то из дней была проблема с API, то просто в веб-интерфейсе Airflow перезапускаешь даг за эту дату, все данные обновляются и становятся актуальными (для этого обязательно использовать контекст Airflow для обращения к датам и обеспечивать в коде условие идемпотентности)