Attribution Pipeline Orchestration (Data Engineer position test challenge)

Task

This challenge tests your ability to design and code a relatively small data pipeline.
In the end it should perform the following steps:
- Query Data from a database (we provide a zipped SQLite DB in 'challenge.zip')
- Transform data as is necessary
- Send transformed data to an API which returns you attribution results
- Write attribution results to the database
- Query and export data from the database

You can imagine that this pipeline should run in separate steps/tasks via an orchestration tool such as Airflow. So please design your code such that the steps can be separated easily. Generally our data engineering code is written in Python using Pandas for data aggregation/transformations.

Tables in the challenge.db file

* session_sources:
  * *session_id*: unique identifier of this session
  * *user_id*: user identifier
  * *event_date*: UTC date when the session happened
  * *event_time*: UTC time when the session happened
  * *channel_name*: traffic channel that started this session (e.g. 'Email')
  * *holder_engagement*: indicator column (0 or 1) displaying whether the session should count for holder enagement in the attribution model (you need to pass this information to the API)
  * *closer_engagement*: indicator column (0 or 1) displaying whether the session should count for closer enagement in the attribution model (you need to pass this information to the API)
  * *impression_interaction*: indicator column (0 or 1) displaying whether the session is an impression in the attribution model (you need to pass this information to the API)

* conversions:
   * *conv_id*: unique identifier of this conversion
   * *user_id*: user identifier
   * *conv_date*: UTC date when the conversion happened
   * *conv_time*: UTC time when the conversion happened
   * *revenue*: conversion amount (i.e. how much revenue the company earned through this conversion, you can assume its in Euro)

* session_costs :
   * *session_id*: unique identifier of this session
   * *cost*: marketing costs of this session (you can assume its in Euro)

* attribution_customer_journey :
   * *conv_id*: unique identifier of this conversion
   * *session_id*: unique identifier of this session
   * *ihc*: attribution value of this session_id for this conv_id (in range of 0-1, you get this back from the IHC API)

Note 1: this table does not exist by default and needs to be created using the challenge_db_create.sql file and subsequently filled after data has been received from the IHC API

Note 2: the sum of 'ihc' column in the 'attribution_customer_journey' should be equal to 1 (100%) for each 'conv_id'

* channel_reporting :
   * *channel_name*: traffic channel that started this session (e.g. 'Email')
   * *date*: date when the session happened (i.e. *event_date* from session_sources)
   * *cost*: sum of marketing costs of those session (you can assume its in Euro)
   * *ihc*: sum of *ihc* for the sessions on the given date and channel
   * *ihc_revenue*: sum of (*ihc* x *revenue* (from conversions table)) for the sessions on the given date and channel

Note 3: this table does not exist by default and needs to be created using the challenge_db_create.sql file and subsequently filled after data has been received from the IHC API and has been combined

Steps
1. Familiarize yourself with the IHC attribution model (https://ihc-attribution.com/ihc-data-driven-attribution-model/)
   - Create a free IHC test account (be aware of the API limits for test accounts)
3. Write Python code to query and build customer journeys out of the two tables: *session_sources* and *conversions*
   - for each *conv_id* you need to get all sessions for the given *user_id* that happened before the conversion timestamp
4. Transform customer journey into list(s) of dictionaries and send to IHC API (https://ihc-attribution.com/marketing-attribution-api/)
   - you need to send the data in chunks as the API is limited to the amount of data that can be processed at once (see 'LIMITS AND QUOTAS' here https://ihc-attribution.com/marketing-attribution-api/)
5. Write the data received back from the IHC API to the table *attribution_customer_journey*
6. Fill the table *channel_reporting* by querying the now filled four tables: *session_sources*, *session_costs*, *conversions* and *attribution_customer_journey*
7. Create a .csv file of *channel_reporting* and add the following two columns:
   * *CPO*: (cost per order) showing the amount of marketing costs for the given date and channel that was spent on getting one attributed (IHC) order
   * *ROAS*: (return on ad spend) showing revenue earned for each Euro you spend on marketing

Deliverable

- Python code that performs the required steps
   - bonus points if the pipeline accepts a time-range as input
- CSV file of the exported *channel_reporting* table from step 6 (with the two additional columns)
- A small report of how you designed the pipeline, the assumptions you took and what could be improved