Data Engineer Task

Your task is to ingest schemaless JSON into a SQL database and run some basic queries on the data. The input data represents a NoSQL store used by the order service, which processes orders that come into a digital ordering system from various channels.

We would like you to spend no more than 3 hours on the task. This may not be enough time to complete all parts, but make your best effort and get as far as you can in the allotted time. You can submit your answers via zipped files or a link to a repository.

Input

The input is a zipped standard JSON file consisting of an array of nested orders with various attributes. The majority of our ETL processes are written in Python 3, so a script written in Python 3 would be preferred, but other languages can also be used. You are free to use any libraries.

Note: the names of fields in the JSON are not always semantic, i.e. they may not mean the same thing as they do in the real world. This should not affect the extraction of the data, or the understanding of the specific queries.

Processing

The file contains a nested structure and it can be assumed that the destination of the data is a SQL based data warehouse, so some normalisation would be required for performance reasons. It is up to you to decide the level of normalisation, and we will discuss your reasoning in the interview. Some columns may need to be transformed to make it easier for a data analyst to interpret, it is your decision where you would want to do this. A relational schema needs to be created and the data loaded into the database. You can use any SQL based database, however for simplicity, SQLite is a quick solution to get you started. You are free to use another database if you find it easier to get started with, i.e. installing MySQL or PostgreSQL in a Docker container.

Queries

For this part, write and show us the SQL queries that would get the following information out:

1. How many total orders were there in the dataset?
2. How many orders were from each channel (requestedFrom column is what shows the channel)
3. How many items were sold for each hour of the day for each tenant?
4. What were the top 5 items sold for each tenant?
5. What were the items for each tenant that were sold more than 5 of?
6. Which order UUIDs had multiples of the same bundle?