Data Engineering - Assignment 03

This exercise involves parsing data from a JSON data source, extracting key pieces of information, and flattening those into a tabular format that can be used for analysis. The data for this exercise is real-world healthcare price transparency data from United Health Services. As you explore this file, pay close attention to all the nested objects and the complexity that creates in understanding and parsing the underlying data.

The assignment is to parse key information from this file as shown in the data examples below using both SQL (in a Snowflake / dbt project) and in a Python notebook. Start with whichever tool provides you the most comfort in exploring the data, then implement in the other tool also.

Remember to submit your solution using GitHub in a folder named “ex03”.

# Input Data

You can find the sample input file in the class GitHub repository under the ex03 folder in a file named “negotiated\_rates.json”. Below is an annotated explanation of a segment of that file.

|  |  |
| --- | --- |
| A screen shot of a computer program  AI-generated content may be incorrect. | The first few elements are basic header information.  The main body of the JSON is a list called “out\_of\_network”, these are the items and rates associated with those items.  Note the key attributes: billing\_code\_type and billing\_code  For each item that can be billed at an out of network rate, you’ll find a list of “allowed\_amounts”…  Which are specific to a particular service\_code, billing\_class, and for a particular list of providers that are identified by NPIs. |

# Output Format

Our goal is to create a list of rates for each of the entries under “out\_of\_network.” That is, the above example would produce the following key fields in a single record:

name E-STIM 1/> AREAS OTH THAN WND CARE…

billing\_code\_type HCPCS

billing\_code G0283

description Electrical stimulation…

service\_code 11

billing\_class professional

allowed\_amount 8.78

billed\_charge 10.0

npi 1184090458

If any of the JSON objects that are lists have multiple values, you’ll need to explode out another record to represent that combination of key fields. Remember that our goal is to produce a simple, rectangular, tabular view of the data.

In both your Python notebook and SQL/dbt solutions, this will probably require multiple steps.