Data engineer - Homework: Music royalties

Background:

We are building a music royalty portfolio. In evaluating which songs present an attractive investment opportunity, we will require a significant volume of data from unrelated sources that will need to be tied together. This will form the backbone of the business and is fundamental to gaining a competitive advantage.

As a data engineer, it will be your responsibility to design and develop this data ecosystem, as well as ensuring referential integrity across the data sets.

Exercise 1: Ingestion and structuring of PDF royalty statements

An important part of due-diligence will be assessing the quality of cash flows related to the music royalties. Cash flows are recorded in royalty statements, which are collated by music Administrators. The statements provide a snapshot of an artists' income by song and separates cash flow by geography and sources (e.g. Spotify, Apple Music, etc.).

Due to there being a number of administrators working in this space, statements will be provided in a range of non-standardised formats and structures. We expect a large number to be in the format of PDFs.

Your first task is to build an ingestion workflow in Python for a single format of PDF (attached).

Points we are looking to see:

- Dynamic ingestion and transformation of data held in PDF into a format that you believe most suitable
- Testing to ensure complete import is successful
- Testing for underlying data quality
- *Ideally testing results written out into an easy to understand output (PDF) but priority is understanding your approach to testing*
- Output tables imported into a staging area that will allow immediate interrogation. This staging area can be in whichever tool you believe most suitable. There can also be intermediate steps if you wish
- All imported PDFs should be assigned a unique ID
- Thought should be given to naming conventions of tables and output files/directories.

Exercise 2: Standardisation of royalty statements

Given that royalty statements are going to be delivered in non-standardised formats, but will also need to be processed in bulk, have a think about how you would:

- Design an overarching workflow approach that can process different data formats (assume all csv for this exercise), without being able to tell from the filename
- Standardise all the royalty statements into a common format, if required

Be ready to talk about this in interview – A couple of slides with your thoughts may help but are not essential.

Exercise 3: Build out the data universe of artists and recordings

There is a finite universe of artists and their associated royalties that we are interested in. We will identify these through various themes evaluated using 3rd party data. As part of this there will be a

requirement to match datasets to a set of common metadata (in our case we will use MusicBrainz for this exercise - https://musicbrainz.org/doc/About).

You have been provided with a random selection of 250 of the top performing songs in the past six decades. Using the Music Brainz API, please can you:

- Match as many billboard artists to their relevant MusicBrainz ID as possible (Not using manual methods)
- Extract all recordings for the matched artists
- For each recording and associated works, capture the contributing artists/writers/composers
- Structure data in a way that could be utilised in a relational database, ensuring the data you deem important is included
- Ensure the code to incorporate into any SQL/NOSQL databases are included, even if you do not have direct access for this assessment

Please provide the code, as well as the output tables you have created.

######
We have set up API credentials for you below.

```
import requests
#pip install musicbrainzngs
#https://python-musicbrainzngs.readthedocs.io/en/v0.7.1/ ##<<< Documentation
import musicbrainzngs
musicbrainzngs.auth("test_application", "xqL4RTKh@7#9P4cG")
musicbrainzngs.set_useragent
musicbrainzngs.set_useragent("test_application", "0.1", "http://example.com/music")
#####
```

Exercise 4: Structuring a database

Given you have now seen 3 types of data inputs (Royalty statements; Billboard lists; MusicBrainz – Assortment of…), please design a simple schema that incorporates the data from the previous exercises and would allow comprehensive querying for an end user. This can be in the data management tool of choice you deem most suitable based on your experience, but please outline the tool name.

You can present this on a small number of slides. You do not have to build the structure in your selected tool.

Outputs, IDs, and naming conventions from previous exercises should be referenced.

---------- END OF EXERCISES ----------

Important

We are interested in assessing your production ready code, so please structure as you would for BAU processes.