The Company

Geowox is developing ground-breaking automated real estate valuation solutions for leading financial sector clients. We are a high growth startup, scaling to 10-15 employees in 2020. We are the winner of the 2019 DatSci awards, and backed by Irish and European innovation funds.

The role

Your responsibility as Software Engineer (Data & Analytics) is to customise, maintain and improve our ETL job pipeline infrastructure. This represents an opportunity to grow and develop your skills within a fast-paced, innovative, data-rich environment.

The assignment

Introduction

Accurate valuations are a function of combining the best data with the best algorithms. A critical part of the entire process is related to the quality of data given as input to our models. Past transactions represent the most important source of data for training and evaluating our models and provide transparent and accurate valuation.

For this assignment you are responsible to create an ETL job pipeline to extract data from the Property price register. The register contains information related to residential sale transactions.

The provider does not have any API and all the sales transactions reside in a .zip file on the website.

Scope of the assignment:

- Showcase your software engineering ability to create an end-to-end ETL process to refresh our internal property transactional data

You will have access to the following resources:

- Sales transactions publicly available on the Property price register (PPR)
- Latest snapshot of the above data ingested by Geowox:
  File name: ppr_current.csv. You can download it here
- Geowox table structure, data types, naming conventions guidelines

Overview of the expected end-to-end ETL process:

- extract.py
  - Scope:
    Programmatically download the latest file available on the PPR website and save it as a .zip file on a dedicated S3 bucket. You can simulate the storage on an S3 bucket folder/key, e.g. bucket_name/extract/ppr.csv. Moto server may help you to simulate some AWS services needed
  - Tips:
    The process in production runs automatically and should be able to handle errors (e.g. website down) without compromising the rest of the ETL pipeline
- transform.py
  - Scope:
    When a new file is uploaded on the S3, a new process is triggered. The scope of this process is to transform the raw data in a more structured and cleaned format ready to be loaded in the final data environment. See the Appendix to view what kind of transformations are needed (you can simulate the storage on an S3 bucket folder/key, e.g. bucket_name/transform/ppr.csv)

    - Tips:
      Please follow naming and conventions for data type and column names and any suggested data transformation of one or more columns.
- load.py
    - Scope:
      Programmatically refresh the current snapshot file available in S3, by adding/updating/removing anything that changed between the two snapshots
    - Tips:
      Keep in mind that, as best practices, several controls should be engineered or taken into consideration before applying any irreversible change to the main file.

Things to keep in mind:

- Most of the time we deal with data where no or little documentation is provided. One of our challenges is to find out what could be the meaning of the data and their integrity, who is the data provider, who else is using this data, etc.
- Google search is always the best place to start with this type of researches

⋆ Please don't forget, your code should be production-ready, clean and tested!

Happy coding :)!

Appendix

Naming conventions guidelines - Columns Suffix

- _date = date format 'dd-mm-yyyy'
- _value = integer/numeric number
- _ind = indicator, interval [0,1]. 0=No, 1=Yes

Naming conventions guidelines - Columns names

- Must be lowercase
- Must not contain spaces or special characters

Table structure:

Column name	Data type
id	serial
sales_date	date
month_start	date
address	varchar(255)
county	varchar(255)
sales_value	integer
not_full_market_price_ind	integer
vat_exclusive_ind	integer
new_home_ind	integer
quarantine_ind	integer
quarantine_code	varchar(255)

Required data transformation

- String standardization
    - Apply best practices to any text column present in the dataset
- Calculated fields
    - Month_start: Date DD/MM/YYYY, based on the first day of the month
    - New_home_ind: [0/1] indicator. Identifies new home transactions
    - Quarantine_ind: [0/1] indicator. Identifies suspicious data.
        - e.g. Not unique record
        - e.g. Not Irish counties
    - Quarantine_code: Short description of the quarantine reason