Promo parsing challenge

Goal

The goal of this exercise is to learn the required tools for working @ Daltix on the promo parsing project.

There are:

- Working with a PostgreSQL database, simple queries are more than enough.
- Working with Python & Pandas.
- Writing regular expressions.
- Working with Git & Github.

Github

This exercise needs to be handed in via Github, you do this by:

- Create a new branch from develop.
- Create a folder with your name (name_surname) in the solutions/promo_parsing directory in which you should store your solution (notebook + text file with queries).
- Commit your work on your branch. Create a new folder under solutions with your name.
- Create a pull-request to merge your branch into master when you are done. Assign to Simon for a review.

Postgres

Postgres is a SQL database, more info on how to install the database on your machine:
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-14-04

It's best to use psql because every once in a while it can be handy to inspect the data.

Concretely the following should be done in this part of the challenge:

- Install PostgreSQL
- Create a database, name it daltix.
- Create a table named promo_strings with the following columns:
  - shop TEXT NOT NULL
  - promo_string TEXT NOT NULL

- Postgres allows you to load a CSV file into a table using the \copy command, use it to load the *promotions.csv* into your newly created table.

Now you should have a database which contains a table promo_strings which holds some data.

Write some queries to answer the following questions & add them in your solution:

- How many different shops are there?
- How many promo strings are there per shop?

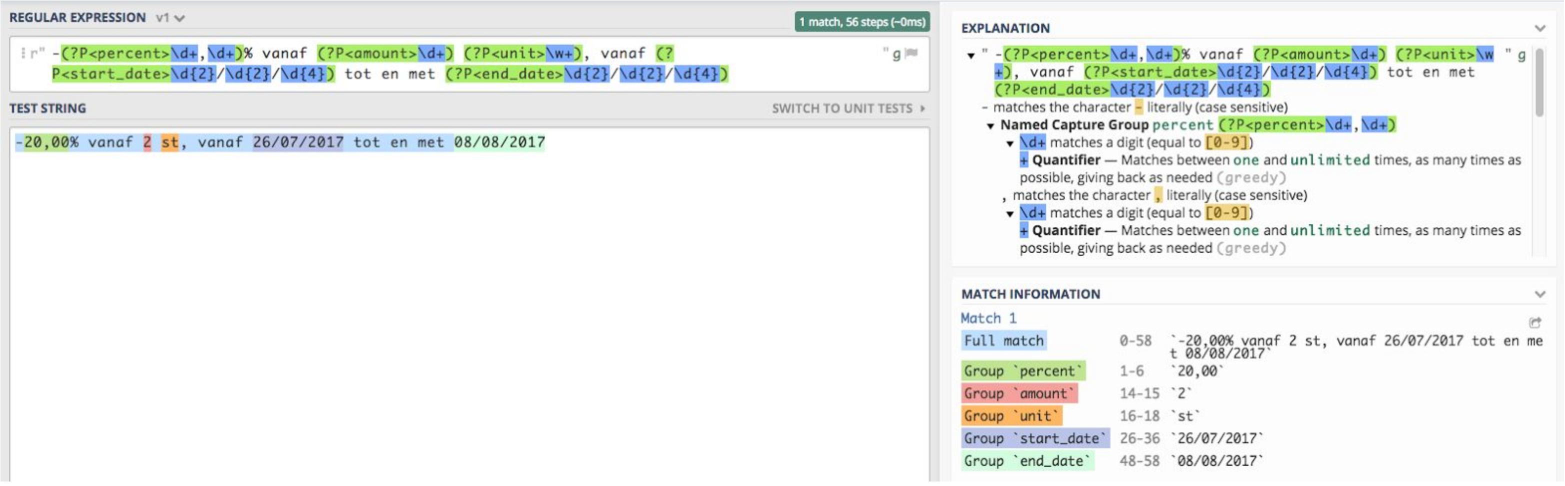
Python, pandas & regex

Pandas is a very popular data manipulation library in Python. Surely you don't need to be fluent with it, however you will need to be able to do some basic operations with it.

The following can be an interesting 10-minute tutorial:
https://pandas.pydata.org/pandas-docs/stable/10min.html

We split promotion strings into pieces using regular expressions, these might look scary at first but after you take some time to learn how they work they are actually often easy for our use-case.

Example



```
r" -(?P<percent>\d+,\d+)% vanaf (?P<amount>\d+) (?P<unit>\w+), vanaf (?P<start_date>\d{2}/\d{2}/\d{4}) tot en met (?P<end_date>\d{2}/\d{2}/\d{4}) " g
```

```
-20,00% vanaf 2 st, vanaf 26/07/2017 tot en met 08/08/2017
```

```
" -(?P<percent>\d+,\d+)% vanaf (?P<amount>\d+) (?P<unit>\w+), vanaf (?P<start_date>\d{2}/\d{2}/\d{4}) tot en met (?P<end_date>\d{2}/\d{2}/\d{4}) " g
```

- matches the character - literally (case sensitive)

Named Capture Group percent (?P<percent>\d+,\d+)

\d+ matches a digit (equal to [0-9])

+ Quantifier — Matches between one and unlimited times, as many times as possible, giving back as needed (greedy)

, matches the character , literally (case sensitive)

\d+ matches a digit (equal to [0-9])

+ Quantifier — Matches between one and unlimited times, as many times as possible, giving back as needed (greedy)

```
Match 1
Full match           0-58   `-20,00% vanaf 2 st, vanaf 26/07/2017 tot en me
                            t 08/08/2017`
Group `percent`      1-6    `20,00`
Group `amount`       14-15  `2`
Group `unit`         16-18  `st`
Group `start_date`   26-36  `26/07/2017`
Group `end_date`     48-58  `08/08/2017`
```

Tip: zoom-in to make this more readable.

You can open the above example via this link:
https://regex101.com/r/Nt0Nqn/1

In the above example we split the promo string
'-20,00% vanaf 2 st, vanaf 26/07/2017 tot en met 08/08/2017'
into various components using named matching groups. A named matching group has the following format: *(?P<name>...)*

This is a useful technique to group parts of a string together.
As you can see we have defined the following groups:

- Percent: the amount of % you get.
- Amount: how many of the product do you have to buy in order to get the percentage?

- Unit: what is the unit of the above amount? Liters, kg? In this case it is st, which means stuks.
- Start/end dates: from when till when is this promotion valid.

By using named matching groups & pandas we can create a *DataFrame* in which the columns are the names of our matching groups and the rows contain their value.

See the example here:
https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.str.extract.html#pandas.Series.str.extract

Concretely for this task you have to fill in the TODOs in the promo_parsing.ipynb, this involves:

- **Install Python 3, jupyter notebook, psycopg2, pandas, numpy.**
- **Open the Jupyter notebook.**
- **Set the correct information so that a DB connection can be made..**
- **Write regexes so that you can parse every promotion string.**
    - **Pay attention to using correct named matching groups, you can reuse the ones defined above & make up some extra ones in case you need some.**
    - **Also avoid using the * character in your regexes, that allows for too much variation, better to be strict in what you expect (e.g numbers where you expect numbers, strings where you expect strings, ...).**