Project Simple ETL with Pandas

Project yang Akan Dikerjakan

Pada proyek kali ini, Anda diminta untuk mengolah data pendaftar hacktahon yang diselenggarkan oleh DQLab bernama DQThon.

Dataset ini terdiri dari 5000 baris data (5000 pendaftar) dengan format CSV (Comma-separated values) dan memiliki beberapa kolom diantaranya:

1. **participant\_id**: ID dari peserta/partisipan hackathon. Kolom ini bersifat unique sehingga antar peserta pasti memiliki ID yang berbeda
2. **first\_name**: nama depan peserta
3. **last\_name**: nama belakang peserta
4. **birth\_date**: tanggal lahir peserta
5. **address**: alamat tempat tinggal peserta
6. **phone\_number**: nomor hp/telfon peserta
7. **country**: negara asal peserta
8. **institute**: institusi peserta saat ini, bisa berupa nama perusahaan maupun nama universitas
9. **occupation**: pekerjaan peserta saat ini
10. **register\_time**: waktu peserta melakukan pendaftaran hackathon dalam second

Namun pada proyek ini nantinya Anda diminta untuk menghasilkan beberapa kolom dengan memanfaatkan kolom-kolom yang ada, sehingga akhir dari proyek ini berupa hasil transformasi data dengan beberapa kolom baru selain dari 10 kolom diatas.

Sebagai pemanasan dalam proyek ini, Anda dipersilakan untuk membuka isi dataset nya dan melihat values nya. Jika sudah siap dengan perjalanan di proyek ini, silakan klik Lanjut.

## Extract

Extract merupakan proses meng-ekstraksi data dari sumber, sumber data ini bisa berupa relational data (SQL) atau tabel, nonrelational (NoSQL) maupun yang lainnya.

Tugas Anda adalah baca terlebih dahulu dataset ini sebagai CSV agar nantinya bisa diolah. Gunakan live code editor untuk menampilkan dataset.

File tersebut bisa diakses di path **dqthon-participants.csv**.

import pandas as pd

df\_participant = pd.read\_csv('dqthon-participants.csv')

## Transform

Transform merupakan proses melakukan transformasi data, atau perubahan terhadap data. Umumnya seperti:

1. Merubah nilai dari suatu kolom ke nilai baru,
2. Menciptakan kolom baru dengan memanfaatkan kolom lain,
3. Transpose baris menjadi kolom (atau sebaliknya),
4. Merubah format data ke bentuk yang lebih standar (contohnya kolom date, maupun datetime yang biasanya memiliki nilai yang tidak standar maupun nomor HP yang biasanya memiliki nilai yang tidak sesuai format standarnya), dan lainnya.

## Transform Bagian I - Kode Pos

Ada permintaan datang dari tim Logistik bahwa mereka membutuhkan kode pos dari peserta agar pengiriman piala lebih mudah dan cepat sampai. Maka dari itu buatlah kolom baru bernama postal\_code yang memuat informasi mengenai kode pos yang diambil dari alamat peserta (kolom address).

Diketahui bahwa kode pos berada di paling akhir dari alamat tersebut.

Note:  
Jika regex yang dimasukkan tidak bisa menangkap pattern dari value kolom address maka akan menghasilkan NaN.

df\_participant['postal\_code'] = df\_participant['address'].str.extract(r'(\d+)$') #Masukkan regex Anda didalam fungsi extract

## Transform Bagian II - Kota

Selain kode pos, mereka juga membutuhkan kota dari peserta.

Untuk menyediakan informasi tersebut, buatlah kolom baru bernama **city** yang didapat dari kolom **address**. Diasumsikan bahwa kota merupakan sekumpulan karakter yang terdapat setelah nomor jalan diikuti dengan \n (newline character) atau dalam bahasa lainnya yaitu enter.

df\_participant['city'] = df\_participant['address'].str.extract(r'(?:\d+)\n+(\w+)') #Masukkan regex Anda didalam fungsi extract

## Transform Bagian III - Github

Salah satu parameter untuk mengetahui proyek apa saja yang pernah dikerjakan oleh peserta yaitu dari git repository mereka.

Pada kasus ini kita menggunakan profil github sebagai parameter nya. Tugas Anda yaitu membuat kolom baru bernama **github\_profile** yang merupakan link profil github dari peserta.

Diketahui bahwa profil github mereka merupakan gabungan dari **first\_name** dan **last\_name** yang sudah di-lowercase.

df\_participant['github\_profile'] = 'https://github.com/' + df\_participant['first\_name'].str.lower() + df\_participant['last\_name'].str.lower()

## Transform Bagian IV - Nomor Handphone

Jika kita lihat kembali, ternyata nomor handphone yang ada pada data csv kita memiliki format yang berbeda-beda. Maka dari itu, kita perlu untuk melakukan cleansing pada data nomor handphone agar memiliki format yang sama. Anda sebagai Data Engineer diberi privilege untuk menentukan format nomor handphone yang benar. Pada kasus ini mari kita samakan format nya dengan aturan:

1. Jika awalan nomor HP berupa angka 62 atau +62 yang merupakan kode telepon Indonesia, maka diterjemahkan ke 0.
2. Tidak ada tanda baca seperti kurung buka, kurung tutup, strip⟶ ()-
3. Tidak ada spasi pada nomor HP Nama kolom untuk menyimpan hasil cleansing pada nomor HP yaitu **cleaned\_phone\_number**
4. #Masukkan regex anda pada parameter pertama dari fungsi replace
5. df\_participant['cleaned\_phone\_number'] = df\_participant['phone\_number'].str.replace(r'^(\+62|62)', '0')
6. df\_participant['cleaned\_phone\_number'] = df\_participant['cleaned\_phone\_number'].str.replace(r'[()-]', '')
7. df\_participant['cleaned\_phone\_number'] = df\_participant['cleaned\_phone\_number'].str.replace(r'\s+', '')

## Transform Bagian V - Nama Tim

Dataset saat ini belum memuat nama tim, dan rupanya dari tim Data Analyst membutuhkan informasi terkait nama tim dari masing-masing peserta.

Diketahui bahwa nama tim merupakan gabungan nilai dari kolom **first\_name, last\_name, country dan institute.**

Tugas Anda yakni buatlah kolom baru dengan nama **team\_name** yang memuat informasi nama tim dari peserta.

def func(col):

abbrev\_name = "%s%s"%(col['first\_name'][0],col['last\_name'][0]) #Singkatan dari Nama Depan dan Nama Belakang dengan mengambil huruf pertama

country = col['country']

abbrev\_institute = '%s'%(''.join(list(map(lambda word: word[0], col['institute'].split())))) #Singkatan dari value di kolom institute

return "%s-%s-%s"%(abbrev\_name,country,abbrev\_institute)

df\_participant['team\_name'] = df\_participant.apply(func, axis=1)

## Transform Bagian VI - Email

Setelah dilihat kembali dari data peserta yang dimiliki, ternyata ada satu informasi yang penting namun belum tersedia, yaitu email.

Anda sebagai Data Engineer diminta untuk menyediakan informasi email dari peserta dengan aturan bahwa format email sebagai berikut:

Format email:

xxyy@aa.bb.[ac/com].[cc]

Keterangan:

xx -> nama depan (first\_name) dalam lowercase

yy -> nama belakang (last\_name) dalam lowercase

aa -> nama institusi

Untuk nilai bb, dan cc mengikuti nilai dari aa. Aturannya:

- Jika institusi nya merupakan Universitas, maka

bb -> gabungan dari huruf pertama pada setiap kata dari nama Universitas dalam lowercase

Kemudian, diikuti dengan .ac yang menandakan akademi/institusi belajar dan diikuti dengan pattern cc

- Jika institusi bukan merupakan Universitas, maka

bb -> gabungan dari huruf pertama pada setiap kata dari nama Universitas dalam lowercase

Kemudian, diikuti dengan .com. Perlu diketahui bahwa pattern cc tidak berlaku pada kondisi ini

cc -> merupakan negara asal peserta, adapun aturannya:

- Jika banyaknya kata pada negara tersebut lebih dari 1 maka ambil singkatan dari negara tersebut dalam lowercase

- Namun, jika banyaknya kata hanya 1 maka ambil 3 huruf terdepan dari negara tersebut dalam lowercase

Contoh:

Nama depan: Citra

Nama belakang: Nurdiyanti

Institusi: UD Prakasa Mandasari

Negara: Georgia

Maka,Email nya: citranurdiyanti@upm.geo

-----------------------------------

Nama depan: Aris

Nama belakang: Setiawan

Institusi: Universitas Diponegoro

Negara: Korea Utara

Maka, Email nya: arissetiawan@ud.ac.ku

def func(col):

first\_name\_lower = col['first\_name'].lower()

last\_name\_lower = col['last\_name'].lower()

institute = ''.join(list(map(lambda word: word[0], col['institute'].lower().split()))) #Singkatan dari nama perusahaan dalam lowercase

if 'Universitas' in col['institute']:

if len(col['country'].split()) > 1: #Kondisi untuk mengecek apakah jumlah kata dari country lebih dari 1

country = ''.join(list(map(lambda word: word[0], col['country'].lower().split())))

else:

country = col['country'][:3].lower()

return "%s%s@%s.ac.%s"%(first\_name\_lower,last\_name\_lower,institute,country)

return "%s%s@%s.com"%(first\_name\_lower,last\_name\_lower,institute)

df\_participant['email'] = df\_participant.apply(func, axis=1)

## Transform Bagian VII - Tanggal Lahir

MySQL merupakan salah satu database yang sangat populer dan digunakan untuk menyimpan data berupa tabel, termasuk data hasil pengolahan yang sudah kita lakukan ini nantinya bisa dimasukkan ke MySQL.

Meskipun begitu, ada suatu aturan dari MySQL terkait format tanggal yang bisa mereka terima yaitu YYYY-MM-DD dengan keterangan:

* YYYY: 4 digit yang menandakan tahun
* MM: 2 digit yang menandakan bulan
* DD: 2 digit yang menandakan tanggal

Contohnya yaitu: 2021-04-07

Jika kita lihat kembali pada kolom tanggal lahir terlihat bahwa nilai nya belum sesuai dengan format DATE dari MySQL

(Lihat lebih detail di: <https://www.mysqltutorial.org/mysql-date/>).

Oleh karena itu, lakukanlah formatting terhadap kolom birth\_date menjadi YYYY-MM-DD dan simpan di kolom yang sama.

df\_participant['birth\_date'] = pd.to\_datetime(df\_participant['birth\_date'], format='%d%b%Y')

## Transform Bagian VII - Tanggal Daftar Kompetisi

Selain punya aturan mengenai format DATE, MySQL juga memberi aturan pada data yang bertipe DATETIME yaitu YYYY-MM-DD HH:mm:ss dengan keterangan:

* YYYY: 4 digit yang menandakan tahun
* MM: 2 digit yang menandakan bulan
* DD: 2 digit yang menandakan tanggal
* HH: 2 digit yang menandakan jam
* mm: 2 digit yang menandakan menit
* ss: 2 digit yang menandakan detik

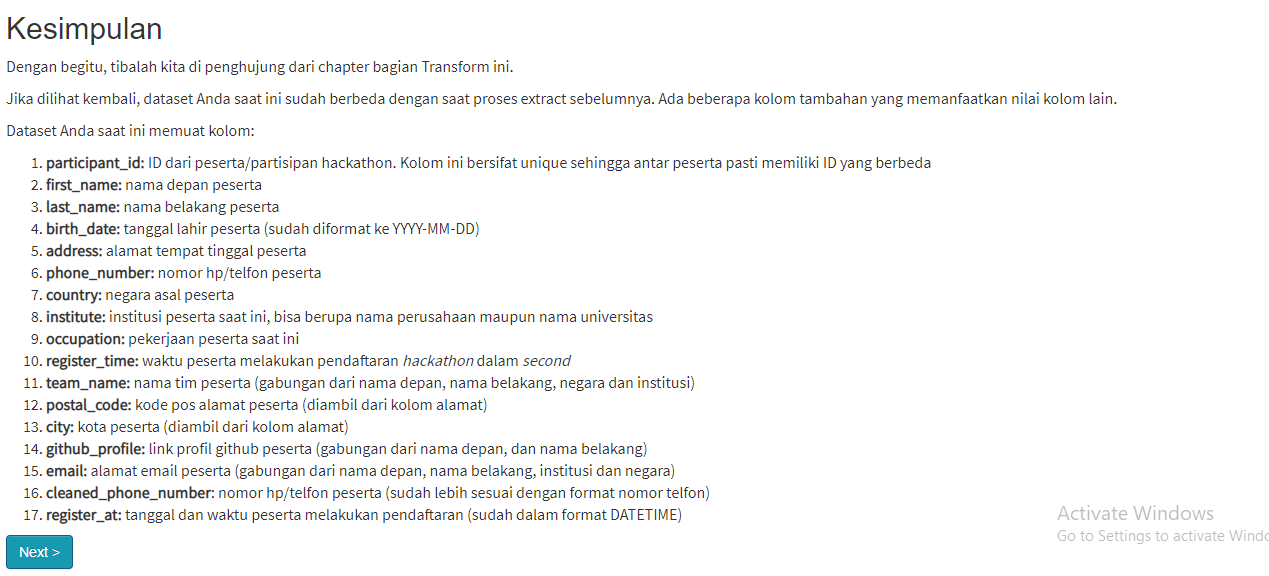
Contohnya yaitu: 2021-04-07 15:10:55

Karena data kita mengenai waktu registrasi peserta (register\_time) belum sesuai format yang seharusnya.

Maka dari itu, tugas Anda yaitu untuk merubah register\_time ke format DATETIME sesuai dengan aturan dari MySQL.

Simpanlah hasil tersebut ke kolom register\_at.

df\_participant['register\_at'] = pd.to\_datetime(df\_participant['register\_time'], unit='s')



## Load

Pada bagian load ini, data yang sudah ditransformasi sedemikian rupa sehingga sesuai dengan kebutuhan tim Analyst dimasukkan kembali ke database yaitu Data Warehouse (DWH). Biasanya, dilakukan pendefinisian skema database terlebih dahulu seperti:

1. Nama kolom
2. Tipe kolom
3. Apakah primary key, unique key, index atau bukan
4. Panjang kolomnya

Karena umumnya Data Warehouse merupakan database yang terstruktur sehingga mereka memerlukan skema sebelum data nya dimasukkan.

Pandas sudah menyediakan fungsi untuk memasukkan data ke database yaitu to\_sql() .

Detail dari fungsi tersebut bisa dilihat pada dokumentasi Pandas: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_sql.html>