**PLAN TO READ DATA**

* Connect to S3
* Extract data from S3 - csv
  + Combine a month and a day
  + Concact CSV to JSON
* convert CSV to JSON/DICT
* Create an unique key for each spartan

**PATTERN**

{Unique\_key: {Name, Gender, DOB, Email, City, Address, Postcode, Phone-number, Uni, Degree, Inv\_date, Inv\_by}}

**UNIQUE KEY**

"0\_Name"

"Name\_Date"

"Integer"

{"Stillmann\_Castano\_22/08/2019": }

**COLUMNS**

=========================================================================================================================================================

**Name (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

str.title() - capitalize every letter in the string at the beggining of each word

**Gender (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Null = Null

**DOB (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

format: "dd-mm-yyyy" TO "yyyy-mm-dd"

**Email (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(optional)[15:13] Matt Lyons

Allowed characters: letters (a-z), numbers, underscores, periods, and dashes. An underscore, period, or dash must be followed by one or more letter or number.

(optional)[15:13] Matt Lyons

Allowed characters: letters, numbers, dashes. The last portion of the domain must be at least two characters, for example: .com, .org, .cc

**City (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- Question: Is it in the UK

- Can be None value

**Address (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- starts with the number

- use api to verify (???)

**Postcode (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- no spaces

- capital letters and numbers

**Phone-number (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- must be string

- no spaces

- no special characters (+ is OK)

- RANGE: 11 if starts with 0, 13 if stars with +44

**Uni (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- (very very optional)Uni API validator (????)

**Degree (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- if degeree in [1st, 2:1, 2:2, 3rd]

- numpy.isnan() for NaN values

**Inv\_date (int)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- check if is below 31

- can't be 0

**Month (str)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

- "mm yyyy"

- fill missing values using file name

- (?) transforming to a datetime

**Inv\_by (str) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**