Analyse_csv_pdf

March 2, 2021

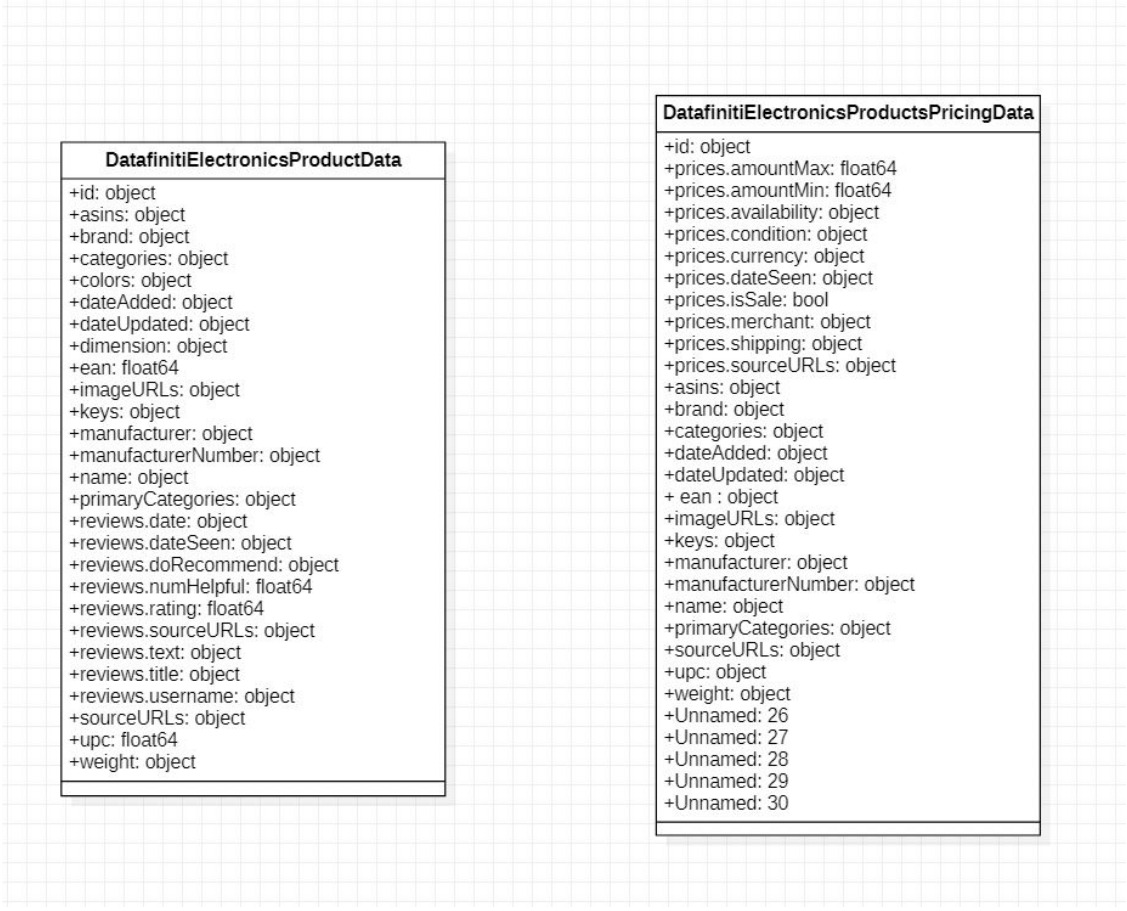
1

Analyse des CSV fournis

1.1 Les deux csv fournis sont les suivant :

- DatafinitiElectronicsProductData
- DatafinitiElectronicsProductsPricingData

1.2 Leurs composition :



Les deux sont composées essentiellement de format string, il y a quand même 4 float dans le premier csv, 2 float et un bool dans la deuxième.

Des colonnes Unnamed sont présente, celles ci pourront être utilisée dans une future gestion de la database.

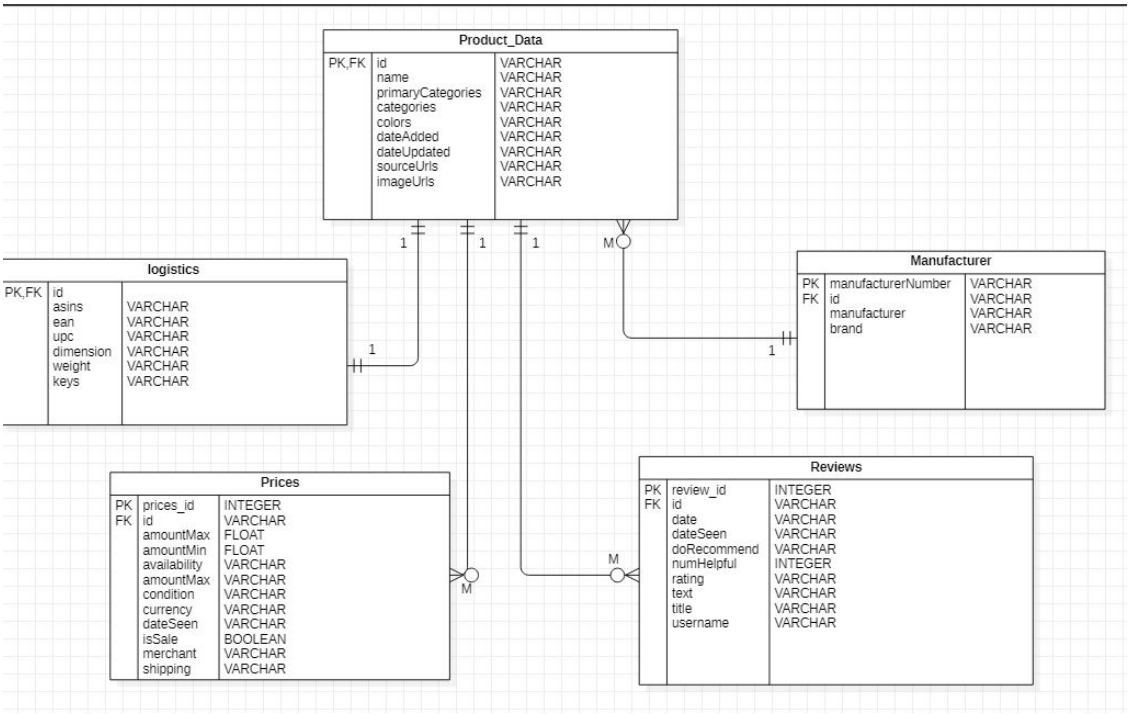
1.3 Les types de données :

```
RangeIndex: 7299 entries, 0 to 7298
Data columns (total 27 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   7299 non-null   object
 1   asins                7299 non-null   object
 2   brand                7299 non-null   object
 3   categories           7299 non-null   object
 4   colors               5280 non-null   object
 5   dateAdded            7299 non-null   object
 6   dateUpdated          7299 non-null   object
 7   dimension            6090 non-null   object
 8   ean                  2951 non-null   float64
 9   imageURLs            7299 non-null   object
 10  keys                 7299 non-null   object
 11  manufacturer         4632 non-null   object
 12  manufacturerNumber   7299 non-null   object
 13  name                 7299 non-null   object
 14  primaryCategories    7299 non-null   object
 15  reviews.date         7238 non-null   object
 16  reviews.dateSeen     7299 non-null   object
 17  reviews.doRecommend  5908 non-null   object
 18  reviews.numHelpful   5813 non-null   float64
 19  reviews.rating       7135 non-null   float64
 20  reviews.sourceURLs   7299 non-null   object
 21  reviews.text         7294 non-null   object
 22  reviews.title        7295 non-null   object
 23  reviews.username     7299 non-null   object
 24  sourceURLs           7299 non-null   object
 25  upc                  7299 non-null   float64
 26  weight               7299 non-null   object
```

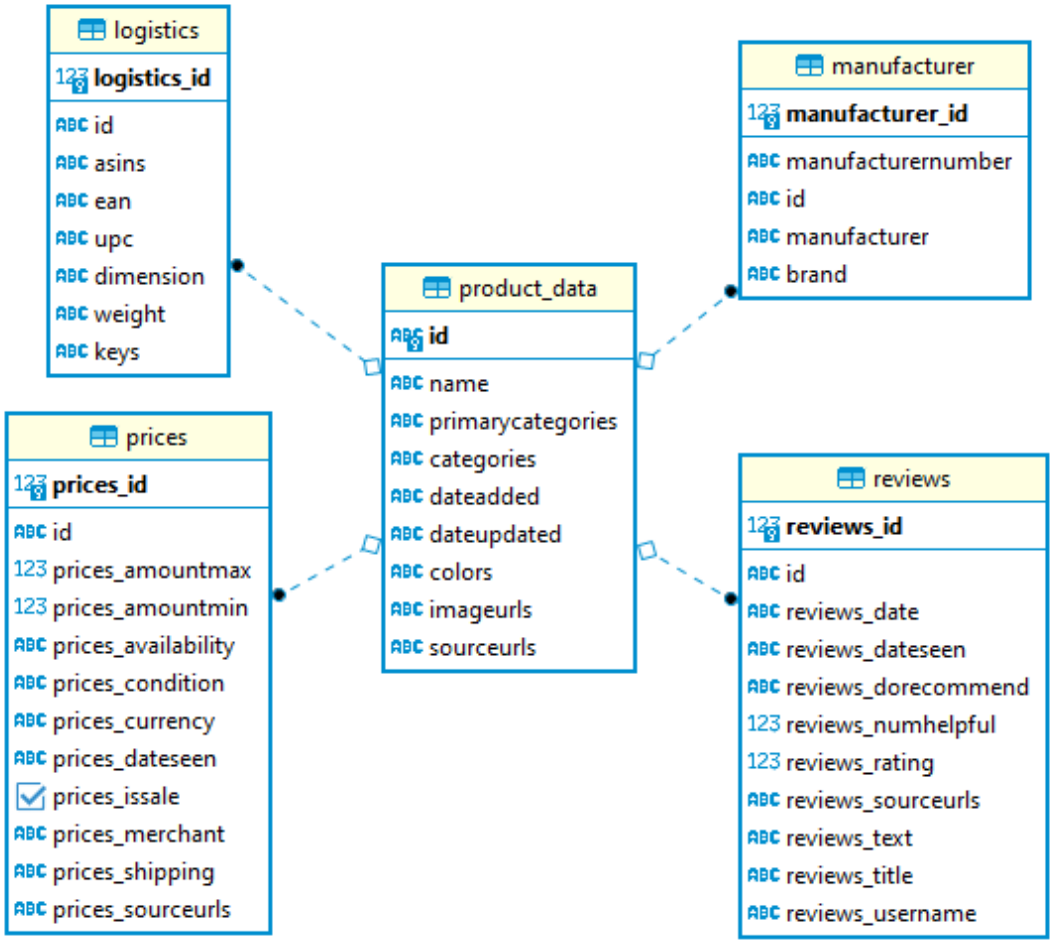
```
RangeIndex: 7249 entries, 0 to 7248
Data columns (total 31 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   7249 non-null   object
 1   prices.amountMax     7249 non-null   float64
 2   prices.amountMin     7249 non-null   float64
 3   prices.availability  7249 non-null   object
 4   prices.condition     7249 non-null   object
 5   prices.currency      7249 non-null   object
 6   prices.dateSeen      7249 non-null   object
 7   prices.isSale        7249 non-null   bool
 8   prices.merchant      7249 non-null   object
 9   prices.shipping      4277 non-null   object
 10  prices.sourceURLs    7249 non-null   object
 11  asins                7249 non-null   object
 12  brand                7249 non-null   object
 13  categories           7249 non-null   object
 14  dateAdded            7249 non-null   object
 15  dateUpdated          7249 non-null   object
 16  ean                  1543 non-null   object
 17  imageURLs            7249 non-null   object
 18  keys                 7249 non-null   object
 19  manufacturer         3235 non-null   object
 20  manufacturerNumber   7249 non-null   object
 21  name                 7249 non-null   object
 22  primaryCategories    7249 non-null   object
 23  sourceURLs           7249 non-null   object
 24  upc                  7249 non-null   object
 25  weight               7249 non-null   object
 26  Unnamed: 26          39 non-null     object
 27  Unnamed: 27          18 non-null     object
 28  Unnamed: 28          6 non-null      float64
 29  Unnamed: 29          12 non-null     object
 30  Unnamed: 30          6 non-null      object
```

1.4 UML :

1.4.1 Architecture de départ



1.4.2 Architecture de la database finale :

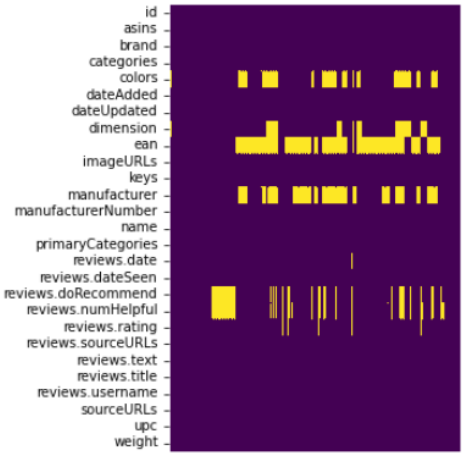


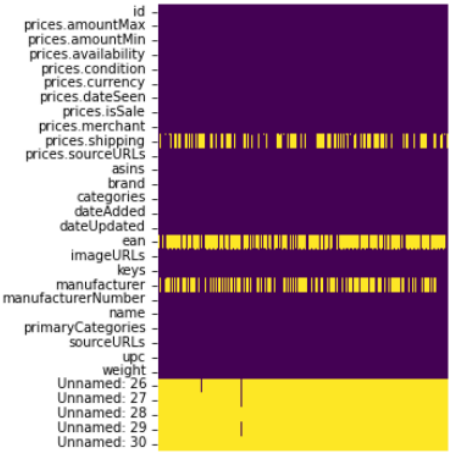
Le choix de cette architecture SQL est de pouvoir indexé et gérer nos données proprement.

Les colonnes unnamed on été écartée, car celles ci me sont inutile pour l'instant.

Dans un version v2 de la database, les primary key (hors product_data [table mère]) seront ajouté à la table product_data pour créé des liens fort.

1.5 Les données compliquées :





Comme nous pouvons le voir juste au dessus, de nombreuses données manque à l'appelle, j'ai décider quand même de les utilisée pour montrer que le jeux de données doit être complété avec d'autre datasets.

Mais malheureusement pas que..
Nous pouvons nous apercevoir que de nombreuse colonnes possèdes des données différentes mais qui veulent sensiblement dire la même chose.

Par exemple :

	prices_availability	nbrid
1	32 available	1
2	7 available	1
3	FALSE	1
4	In Stock	3 172
5	More on the Way	91
6	No	4
7	Out Of Stock	115
8	Retired	1
9	sold	22
10	Special Order	109
11	TRUE	663
12	undefined	40
13	yes	893
14	Yes	2 136
15	[NULL]	50

Un autre exemple, auquel ce rajoute un autre problème qui est un peu plus embêtant, nous remarquons que des données on été décalées :

	prices_condition	nbrid
1	5/16" Ring Terminal, 3 ft. 8 GA Black Ground Cable, 6 ft. Split Loom Tubing, Depth: 6.5" (165mm) (top) 11.2" (285mm) (bottom	1
2	Manufacturer refurbished	55
3	new	699
4	New	6 226
5	New Kicker BT2 41IK5BT2V2 Wireless Bluetooth USB Audio System Black + Remote, Power Supply (volts, ampere): 24, 2.9, Squa	2
6	New other (see details)	92
7	pre-owned	3
8	refurbished	2
9	Refurbished	11
10	Seller refurbished	15
11	Used	143
12	[NULL]	50

Il y avais aussi un autre léger problème, le nom des colonnes qui comportaient des points.

Celles ci ont été remplacer grâce à cette méthode :

```
df3.columns = df3.columns.str.replace(".", "_")
```

Les points ont été remplacer par des underscore ” _ ”, pour éviter tout conflit avec python.

Toutes les indications sur la propreté des données que comporte les csv fournis sont données essentiellement à titre indicatif, en effet, dans mes capacités actuelle, il mais impossible de procéder à un nettoyage rapide et propre. Seulement la modification des noms de colonne à été faites, les autres modification apparaîtront dans une autre version du projet.

Merci.