Ce projet est cofinancé par l'Union européenne

L'EUROPE S'ENGAGE en région Auvergne-Rhône-Alpes avec le FSE

La Région Auvergne-Rhône-Alpes

Modèles de données

source: xkcd

ITÉRATION 1

Petit tour des données extraites

OBJECTIF

- Faire un point sur les données collectées depuis le début du module
- Revoir l'UML

1.1 — Une drôle de journée

1,5 heure— Présentiel

Cette journée vous paraît plutôt banale, jusqu'à qu'une notification mail trouble l'une de vos premières pauses café de la semaine!
Cette notification est un mail venant des personnes qui vous ont missionné il y a quelque temps après avoir vu vos talents en scraping!

Dans leur mail, ces personnes mentionnent quelques excuses comme quoi elles étaient occupées et n'ont pas pensé à vous contacter pour connaître l'avancement de ce projet.

Elles vous demandent de leur envoyer un résumé contenant les différentes données collectées. Après un moment à procrastiner, vous décidez que le mieux serait de leur faire un schéma pour décrire le contenu des différents fichiers. Vous vous rappelez soudainement du module de base de données et des diagrammes de classe UML, vous décidez d'utiliser ce formalisme afin de représenter les données collectées.

Consignes

- Utilisez un outil de "dessin" (un papier et un crayon suffisent) pour schématiser les données que vous avez collectées depuis le début du module.
- Formalisez ce schéma en utilisant les diagrammes de classe UML (les relations entre les différents fichiers
- Faites apparaître les relations entre les différentes classes du diagramme fait précédemment.
    - Quelles sont les clés primaires des différentes classes?
    - Des colonnes contiennent elles des tableaux?
    - Des données redondantes sont-elles présentes dans votre diagramme?

RESSOURCES

- Outils de dessin:
  - https://excalidraw.com/
  - https://app.diagrams.net/
- Quelques rappels sur l'UML
  - https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2015/modeling/uml-class-diagrams-reference
  - https://www.youtube.com/watch?v=LmS4Y99fNaQ
  - https://www.youtube.com/watch?v=X89KLfrNOPo

LIVRABLES

❏ Le diagramme UML

ITÉRATION 2

Optimisation de la structure des données collectées

OBJECTIF

- Optimiser la structure de stockage des données.
- Découvrir les formes normales de base de données

2.1 —Normalisation du modèle de données

1h — Présentiel

Il est temps maintenant d'optimiser la structure des données extraites afin de limiter la redondance des données et de simplifier leur vérification à partir de celles-ci.

Pour cela il existe une méthode pour caractériser les relations entre les différentes entités présentes dans les données, celle-ci consiste à enlever les redondances et à créer des colonnes dite "atomique"; c'est-à-dire ne contenant pas de tableau ou de liste. Ces paramètres sont décrits dans ce que l'on appelle des "formes normales".

Il existe plusieurs formes normales différentes, les quatres les plus courantes sont les trois premières et celle de Boyce-Codd. Une modélisation respectant la troisième forme normale respectera aussi les deux premières.

Consignes

- Renseignez vous sur les formes normales
    - dans un mémo décrivez avec vos mots:
        - La première forme normale
        - La seconde forme normale
        - La troisième forme normale
        - La forme normale de Boyce-Codd
- À quelle forme normale correspond votre diagramme réalisé précédemment ?
- À partir du diagramme fait précédemment:
    - Adaptez le pour qu'il respecte la première forme normale
    - Faites de même pour les seconde et troisième formes normales.

RESSOURCES

- Description des formes normales: https://fr.wikipedia.org/wiki/Forme_normale

- https://www.databasestar.com/database-normalization/
- https://phlonx.com/resources/nf3/

LIVRABLES

- ❏ Le mémo
- ❏ Les diagrammes adaptés

2.2 —Changement du modèle de données

2h — Présentiel

Maintenant que vous avez un schéma évitant trop de redondance et permettant de vérifier plus facilement l'intégrité des données collectées, vous pensez maintenant à adapter vos scripts python faits précédemment pour respecter un de vos diagrammes en seconde ou troisième forme normale.

Cependant, adapter tous les scripts vous semble être une tâche un peu longue et complexe, et si le schéma devait changer dans un futur proche, cela impacterait toutes vos routines de collecte. Vous décidez de garder vos scripts de collecte intacts.

L'idée va être de créer plusieurs scripts de migration, permettant de passer chaque fichier créé dans le format désiré, il faudra donc créer un fichier csv par classe présente dans le diagramme.

Consignes

- Rangez vos fichiers python dans deux répertoires:
  - Un répertoire nommé collect qui contiendra les scripts de collecte
  - Un répertoire nommé migrations qui contiendra les scripts de migration
- Créer les scripts nécessaires pour migrer les fichiers collectés précédemment pour que le schéma de données corresponde à un des diagrammes de classe de forme normale 2 ou 3.

Livrables

➔ les données collectées dans les différents fichiers
➔ Une première base de données structurée en table