HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Méthodes d'accès aux données

Projet

DATA MODEL AND ANALYTIC QUERIES

Étudiants	Gilliand Loris - Tutic Mateo - Wachter Luc
Nom du professeur	Fatemi Nastaran
Nom des assistants	Meier Christopher Hochet Guillaume
Date	15.11.2019

Table des matières

1. Introduction

Dans ce document, vous trouverez une description du modèle de données graphe ainsi que du modèle de données document, tels que nous les avons définis jusque-là. Vous trouverez aussi une liste des requêtes que nous aimerions implémenter et les noeuds et relations qu'elles utilisent.

2. Technologie

Pour le projet, nous avons décidé d'utiliser `ArangoDB`. Cette technologie permet de regrouper le modèle de données graphe et le modèle de données document (entre autres), et propose son propre langage de requêtes.

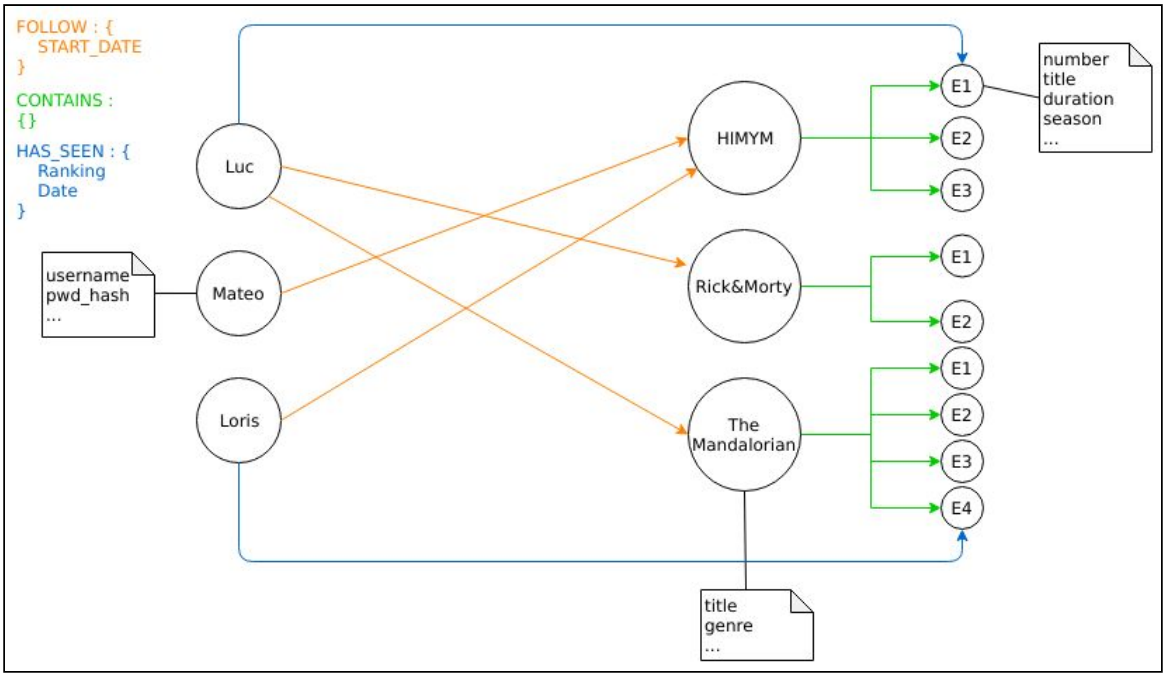
Elle semble adaptée à un tel projet et nous permettrait d'expérimenter avec une technologie un peu différente des technologies de base de données mono-modèle.

3. Modèle de données graphe

Vous trouverez dans la figure suivante une représentation basique d'un graphe possible avec notre modèle de données. On y trouve trois ensembles distincts de sommets, représentant respectivement les *utilisateurs*, les *séries* et leurs *épisodes*.

Les arcs sont également de trois types, et représentent les relations suivantes :

- l'utilisateur **suit** cette série
- l'utilisateur **a vu** cette série
- la série **contient** cet épisode.



4. Modèle de données document

Les sommets de type `User` contiendront les informations nous permettant d'identifier un utilisateur du bot. Ces informations nous seront disponibles à travers l'API de Telegram (https://python-telegram-bot.readthedocs.io/en/stable/telegram.user.html).

Les sommets de type `Series` contiendront les informations relatives aux séries récupérées depuis une API ouverte. Les champs qui nous intéressent sont principalement l'id, le titre, le genre et la date de sortie.

Les sommets de type `Episode` contiendront bien sûr les informations relatives aux épisodes des séries. Les champs qui nous intéressent sont surtout l'id, le numéro d'épisode, le titre, la longueur, la date de sortie et la saison.

5. Requêtes

Concernant les requêtes, plusieurs ont été décrites dans le cahier des charges. Ces dernières sont de complexité moyenne et seront développées selon le temps à disposition. Au minimum les requêtes suivantes seront disponibles :

Description de la requête	**Noeuds et relations impliqués**
Récupérer une liste de séries en fonction du titre.	Série (données consommées).
Récupérer la liste des séries suivies par l'utilisateur.	Série, utilisateur, relation FOLLOWS.
Récupérer la liste des épisodes vus dans une série donnée.	Série, épisode, utilisateur, relation HAS_SEEN.
S'enregistrer en tant qu'utilisateur (lors du lancement de la discussion avec le bot).	Utilisateur.
Suivre une série.	Utilisateur, série, relation FOLLOWS.
Marquer un épisode comme vu (et le noter).	Utilisateur, série, épisode, relation HAS_SEEN.

Afin d'implémenter une requête plus complexe, qui utiliserait mieux les capacités de notre base de données basée graphes, nous avons pensé à la requête suivante.
Quelles séries que je suis sont aussi suivies par un utilisateur donné ? Il s'agirait de trouver tous les chemins allant d'un utilisateur A à un utilisateur B (sans prendre en compte la direction de l'arc).