Construction Data Lake et Data warehouse

Prérequis:
Ce TP s'inscrit à la suite du TP Ksqldb 2.
Les applications doivent être écrites en Python

Design Data Lake

- Le data lake sera un dossier **data_lake** sur votre machine locale.
- Les données lues depuis Kafka (ksqldb) doivent être partitionnées par date (ou version)
  - Prévoir des possibles ajouts de nouveaux feeds et comment ils seront gérés et stockés
- Chaque stream et table de ksqldb doit être stocké sur le file system
  - Définir et justifier le mode de stockage en fonction du stream et de la table (append, overwrite, ignore etc …)

Design Data Warehouse

- Toutes les Tables dans Ksqldb doivent être également stockées dans une table MySQL
  - Définir clés primaires, clés étrangères et liens entre les tables si besoin à travers un simple diagramme entité-association

Kafka consumers

- Écrire une ou plusieurs applications Kafka consumer pour peupler le data lake et le data warehouse.

Gouvernance et sécurité

- Mécanisme de suppression des données historiques
- Gérez les droits d'accès par utilisateur. Définissez une table dans le data warehouse qui stockera les permissions par dossier du data lake (ce dossier sera checké plus tard lorsqu'on rajoutera la partie API/viz)
- En cas d'ajout d'un nouveau feed dans Kafka, décrivez la procédure à adopter pour avoir le nouveau feed dans le data lake, et décrivez comment vous réutiliserez le travail déjà fait pour accélérer l'intégration du nouveau feed

Orchestration et optimisation

- Utilisez Dataflow et apache beam (pip install apache-beam schedule) pour controller l'exécution des job chaque 10 minutes.
- Utilisez le script python producer pour envoyer des messages dans Kafka. Modifiez ses paramètres pour augmenter le nombre de messages à envoyer.