brief-etl-docker

August 7, 2024

0.0.1 Brief - Projet ETL avec Docker

Description du projet Bienvenue à ce projet d'intégration et de traitement des données (ETL). Dans ce projet, vous allez travailler avec un fichier CSV contenant des données de ventes issues d'un supermarché fictif appelé SuperStore. Vous utiliserez ces données pour modéliser une base de données relationnelle dans MySQL et créer un pipeline ETL complet en utilisant Docker.

Description des données Le fichier CSV fourni, `SuperStoreRawData.csv`, contient des informations sur les commandes de produits effectuées par des clients. Voici une description des colonnes présentes dans ce fichier :

- `Order ID` : Identifiant unique de chaque commande.
- `Order Date` : Date à laquelle la commande a été passée.
- `Ship Date` : Date à laquelle la commande a été expédiée.
- `Ship Mode` : Mode d'expédition utilisé.
- `Customer ID` : Identifiant unique du client ayant passé la commande.
- `Customer Name` : Nom du client.
- `Segment` : Segment de marché auquel appartient le client.
- `Sales Rep` : Représentant des ventes associé à la commande.
- `Product ID` : Identifiant unique du produit commandé.
- `Product Name` : Nom du produit.
- `Category` : Catégorie du produit.
- `Sub-Category` : Sous-catégorie du produit.
- `Sales` : Montant des ventes pour le produit.
- `Quantity` : Quantité commandée du produit.
- `Discount` : Remise appliquée sur le produit.
- `Profit` : Profit réalisé sur la vente du produit.
- `City` : Ville de l'adresse de livraison.
- `State` : État de l'adresse de livraison.
- `Postal Code` : Code postal de l'adresse de livraison.
- `Region` : Région de l'adresse de livraison.
- `Sales Team` : Équipe de ventes.
- `Sales Team Manager` : Manager de l'équipe de ventes.
- `Location ID` : Identifiant unique pour l'emplacement de la livraison.

Le fichier contient des problèmes courants de données tels que : - Valeurs manquantes dans certaines cellules. - Lignes dupliquées. - Valeurs aberrantes. - Formats de données incohérents. - Valeurs incorrectes (par exemple, valeurs négatives dans des colonnes où elles ne devraient pas être).

Objectif du projet Votre tâche est de créer un pipeline ETL complet pour ces données. Le projet doit inclure les étapes suivantes :

1. **Nettoyage des données** :
   - Identifier et gérer les valeurs manquantes.
   - Supprimer les doublons.
   - Détecter et traiter les valeurs aberrantes.
   - Uniformiser les formats de données.
   - Corriger les valeurs incorrectes.
2. **Modélisation de la base de données** :
   - Concevoir un schéma relationnel pour la base de données en définissant les tables, les clés primaires et les clés étrangères nécessaires.
   - Assurer que la base de données soit optimisée pour les opérations de requête et d'analyse.
3. **Écriture du script SQL** :
   - Rédiger un script SQL permettant la création de la base de données MySQL et de toutes les tables nécessaires.
   - Inclure les contraintes de clés primaires et étrangères dans le script SQL.
   - Insérer les données nettoyées du fichier CSV dans les tables appropriées.
4. **Configuration de Docker** :
   - Créer un environnement Docker comprenant les services nécessaires :
     - **MySQL** : Service de base de données MySQL pour stocker les données.
     - **Adminer** : Interface web pour gérer la base de données MySQL.
     - **Jupyter Notebook** : Environnement pour l'analyse des données et l'exécution de scripts SQL.
   - Configurer les volumes et les réseaux Docker pour permettre la communication entre les services.
   - Assurer que les données soient correctement chargées dans MySQL lors de l'initialisation des conteneurs.

Étapes à suivre

1. **Nettoyage des données** :
   - Utiliser un outil de traitement de données (par exemple, Python avec pandas) pour nettoyer les données.
   - Identifier et gérer les valeurs manquantes.
   - Supprimer les doublons.
   - Détecter et traiter les valeurs aberrantes.
   - Uniformiser les formats de données.
   - Corriger les valeurs incorrectes.
2. **Modélisation de la base de données** :
   - Identifier les entités principales et leurs attributs.
   - Définir les relations entre les entités (un-à-un, un-à-plusieurs, plusieurs-à-plusieurs).
   - Créer un diagramme ERD (Entity-Relationship Diagram) pour représenter visuellement le schéma de la base de données.
3. **Écriture du script SQL** :
   - Créer un fichier `init.sql` pour la création des tables et l'insertion des données nettoyées.
4. **Configuration de Docker** :
   - Écrire un fichier `docker-compose.yml` pour définir les services Docker nécessaires.

- Configurer les fichiers Docker nécessaires pour chaque service.
- Assurer que tous les services peuvent démarrer correctement et communiquer entre eux.

5. **Chargement des données et analyse** :
   - Utiliser le service Jupyter Notebook pour se connecter à la base de données MySQL.
   - Effectuer des requêtes SQL pour analyser les données chargées.
   - Produire des visualisations et des rapports sur les données.

Livrables attendus

1. **Données nettoyées** : Fichier CSV contenant les données nettoyées en expliquant et montrant comment vous avez nettoyé les données.
2. **Diagramme ERD** : Représentation visuelle du schéma de la base de données.
3. **Script SQL** : Fichier `init.sql` correctement commenté et fonctionnel pour créer les tables et insérer les données nettoyées.
4. **Configuration Docker** : Fichier `docker-compose.yml` et fichiers Docker associés pour chaque service.
5. **Documentation** : Explication détaillée des étapes suivies pour le nettoyage des données, la modélisation de la base de données, l'écriture du script SQL et la configuration de Docker.
6. **Analyse des données** : Fichiers Jupyter Notebook montrant les analyses effectuées sur les données nettoyées et chargées.

Ressources Utiles

- Documentation MySQL : https://dev.mysql.com/doc/
- Documentation Docker : https://docs.docker.com/
- Documentation Jupyter Notebook : https://jupyter.org/documentation

Conseils

- Assurez-vous de bien comprendre les relations entre les entités avant de créer les tables.
- Testez vos scripts SQL dans un environnement de test avant de les exécuter dans Docker.
- Utilisez des volumes Docker pour la persistance des données afin de ne pas perdre les données lorsque les conteneurs sont redémarrés.
- Documentez chaque étape de votre projet pour faciliter la compréhension et la reproduction du travail.