1- compter le nombre  de transactions des 3 derniers mois  sans cache et avec cache et comparer les performances à travers la spark UI ou  une librairie python.

2- supprimer le cache et faites une jointure avec la table clients sans optimiser la jointure et après optimisation. Comparer les performances

3- associer à la table transactions les bons labels associés aux codes de la colonne "mcc"

4- Créer 2 vues temporaires, une pour les transactions frauduleuses et une pour les non frauduleuses

5- Enrichissez le dataframe transactions avec quelques données clients et les informations sur les fraudes ensuite stocker le résultat sur disque de facon déséralisée

6-top 10 clients de l'année 2010 (ceux qui ont le plus dépensé) et stocker sur une table Hive (built in)  en disque

7- top 3 clients chaque mois de l'année 2010 (ceux qui ont le plus dépensé)

8- Nous voulons créer un datamart qui contiendra des métriques sur les clients sur les données de 2010. Nous voulons, l'année, l'idClient, nombre de transactions effectuées, somme moyenne depensée, somme maximale depensée, le mois où il y a le plus de dépenses, la date de la dernière transaction. Ce datamart doit être stocké dans une base de donnée de votre choix

ces données doivent être accessible en API par un appel GET /users/:id/transaction-metrics

l'API doit être faite avec fast API et permettre la pagination

9-  Utiliser une partie des données du fichier "train fraud transactions" pour entrainer un modèle que vous ferez valider avec l''autre partie des données. Le modèle doit prédire si une transaction est frauduleuse ou pas. Utiliser au moins 3 algorithmes

10- partitionner les données brutes des transactions par mois de transaction dans un dossier  "bronze\_transactions\_by\_month" (pour le stockage)

11- je veux connaitre la moyenne des dépenses en ligne et surplace. (pensez vous qu'on peut optimiser votre requête ?) Si oui comment ?

12 - Le résultat de la question 8 nous voulons un stockage de fichiers avec 5 partitions