TP Révision : **Spark Scala**

Exercice 1 : Premiers pas avec Spark et Scala

- Créer une instance de SparkSession.
- Charger un fichier texte simple (ex : fichier .txt).
- Compter le nombre total de lignes dans le fichier.
- Afficher les 10 premières lignes du fichier.

Exercice 2 : Traitement basique avec les RDDs

- Charger un fichier texte en RDD.
- Réaliser un Word Count : compter le nombre d'occurrences de chaque mot.
- Trier les résultats par fréquence décroissante.
- Sauvegarder les résultats dans un fichier texte (ou dossier de sortie).

Exercice 3 : Manipulation de DataFrames et Datasets

- Charger un fichier CSV en DataFrame (ex : données de ventes, logs, etc.).
- Afficher le schéma (colonnes et types de données).
- Filtrer les données selon une condition (ex : quantité > 1000).
- Effectuer une agrégation (ex : somme des ventes par région).
- Enregistrer le DataFrame filtré ou agrégé au format Parquet.

Exercice 4 : Jointures et opérations avancées

- Charger deux DataFrames (ex : clients.csv et commandes.csv).
- Effectuer des jointures : inner join, left join, etc.
- Identifier les clients sans commande.
- Calculer la moyenne des montants de commande par client.

Exercice 5 : Utilisation des fonctions SQL dans Spark

- Créer une table temporaire (temp view) à partir d'un DataFrame.
- Écrire des requêtes SQL pour filtrer, agréger ou trier les données.
- Utiliser des fonctions SQL comme :
- date_format() : formater une date

- substring() : extraire une sous-chaîne
- countDistinct() : compter les valeurs uniques

Exercice 6 : Traitement de données temporelles

- Charger un jeu de données contenant des timestamps (dates/heures).
- Calculer des différences de temps entre événements (ex : temps entre commande et livraison).
- Grouper les données par jour, semaine ou mois.
- Visualiser les tendances (par exemple en exportant les données vers CSV, puis avec Excel, Python, ou Power BI).

Exercice 7 : Optimisation et partitionnement

- Étudier l'impact du partitionnement sur les performances de traitement.
- Répartir manuellement les données avec repartition() et coalesce().
- Utiliser la mise en cache ou la persistance (cache(), persist()) pour réutiliser les DataFrames sans recalcul.

Exercice 8 : Traitement de données en streaming (avancé)

- Lire un flux de données en streaming (ex : logs qui arrivent en temps réel).
- Appliquer un traitement continu (ex : filtrage, comptage, agrégation par fenêtre temporelle).
- Écrire les résultats dans un sink : console, fichier, base de données, etc.