Département mathématique Informatique

Big Data : Fondements et Architectures de stockage

C

|  |  |  |
| --- | --- | --- |
|  | **ECOLE NORMALE SUPERIEURE**  **DE L’ENSEIGNEMENT TECHNIQUE DE MOHAMMEDIA** |  |

**Examen**

**Professeur :**

**Mr. BOUSSELHAM Abdelmajid**

2ème année II-BDCC

Filière d’ingénieur : Ingénieur informatique, Big Data et Cloud Computing

C

**Rapport**

**2022/2023**

**Réalisé par :**

**Ayoub ETOULLALI**

**SOMMAIRE**

[Introduction 2](#_Toc136621724)

[Travail à faire 3](#_Toc136621725)

[Exercice 1: Manipuler le système de fichiers HDFS 3](#_Toc136621726)

[Exercice 2: 6](#_Toc136621727)

[Partie 1 : Spark SQL 6](#_Toc136621728)

[Partie 2 : Importer et exporter des données avec SQOOP 9](#_Toc136621729)

[Partie 3: Traitement de données en streaming 10](#_Toc136621730)

[Conclusion 13](#_Toc136621731)

Introduction



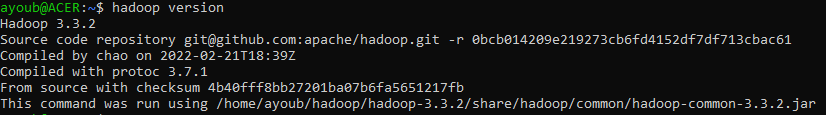
**L'analyse des données et la prise de décision basées sur ces données sont devenues des aspects cruciaux pour de nombreuses entreprises. Dans le domaine du Big Data, la capacité à manipuler, traiter et extraire des informations significatives à partir de grands ensembles de données est essentielle. Cet examen pratique en Big Data vise à évaluer vos compétences dans la manipulation du système de fichiers HDFS ainsi que dans l'utilisation des outils de traitement de données distribuées tels que Spark SQL.**

**Au cours de cet examen, vous serez confronté à différentes tâches liées à la gestion des données à grande échelle. Vous devrez effectuer des opérations telles que la création de répertoires, le chargement de fichiers, l'affichage du contenu des répertoires, la manipulation des fichiers, et bien d'autres encore. Cela vous permettra de mettre en pratique vos connaissances en Big Data et de démontrer votre compréhension des concepts clés.**

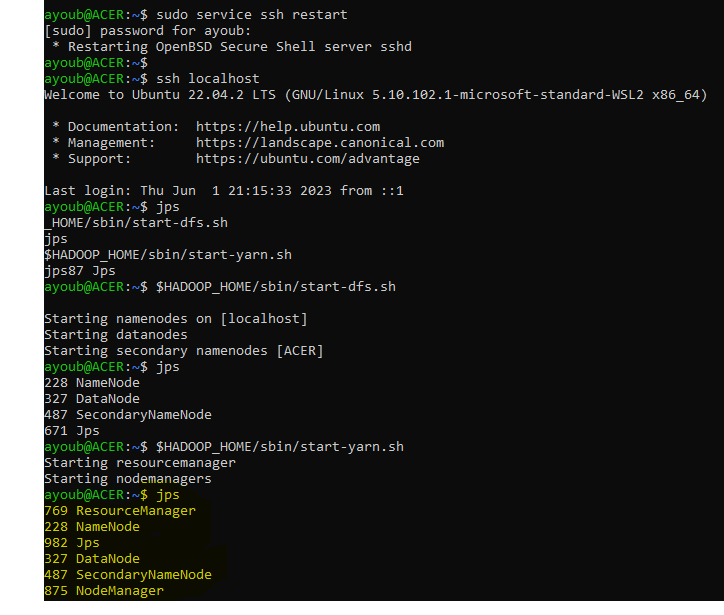
Travail à faire

Exercice 1: Manipuler le système de fichiers HDFS

1. Vérifiez la version Hadoop.

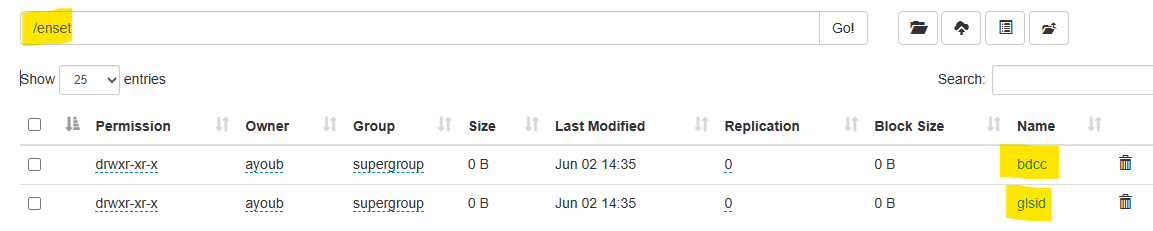


1. Démarrez HDFS et vérifiez qu’il est en cours d’exécution.



1. Créez deux nouveaux répertoires nommés /enset/bddc et /enset/glsid sur HDFS.

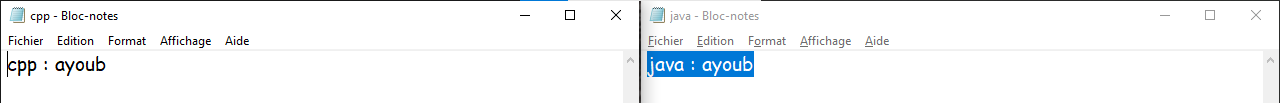




4. Créez un nouveau fichier java.txt contenant 10 lignes et cpp.txt contenant 10 lignes sur

votre système local.



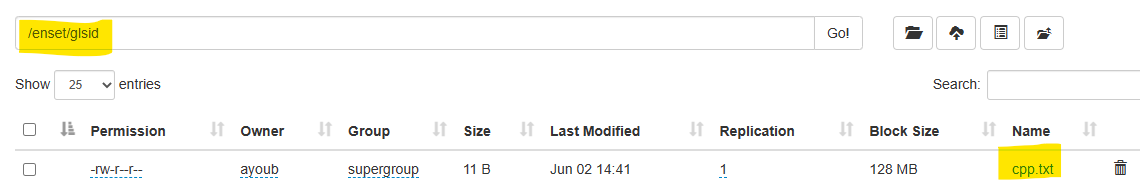


1. Charger le fichier java.txt dans /enset/bddc et cpp.txt dans /enset/glsid sur HDFS.

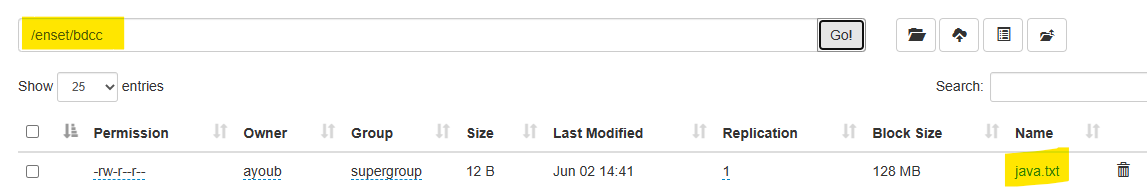


1. Afficher le contenu du répertoire /enset/bddc et /enset/glsid.

hdfs dfs -ls -R /enset/glsid



hdfs dfs -ls -R /enset/bddc

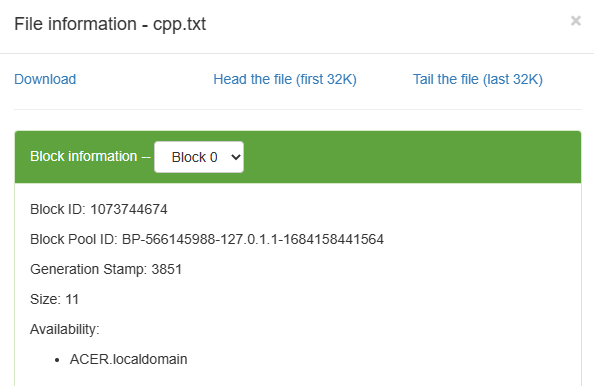


1. Affichez le contenu du fichier java.txt qui se trouve dans HDFS.



1. Déterminez la taille du fichier cpp.txt qui se trouve dans HDFS.

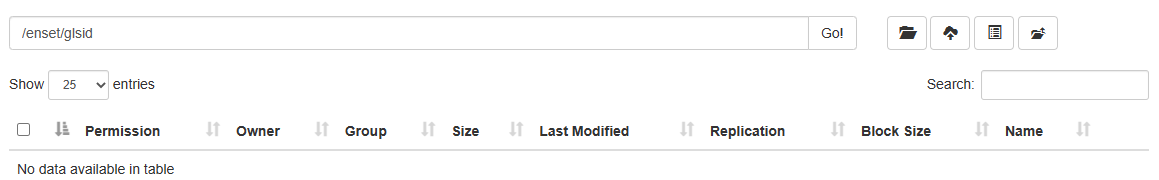




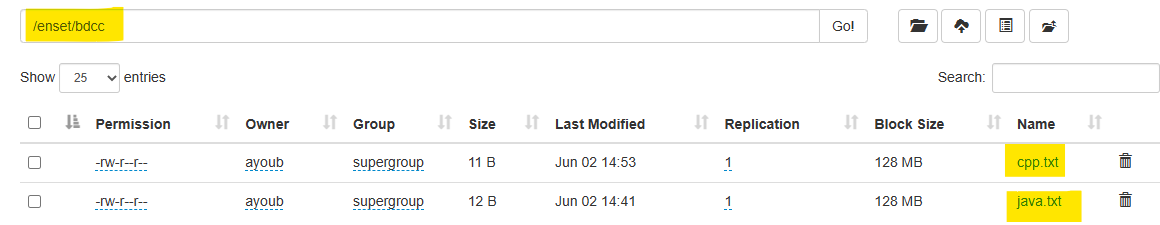
1. Déplacez le fichier cpp.txt vers /enset/bddc et vérifier si le fichier est bien déplacé.



**Vider /enset/glsid**

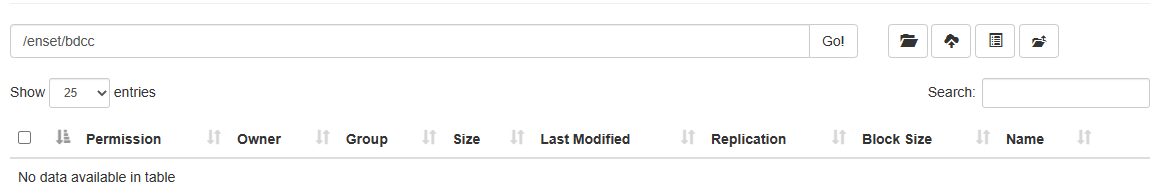


**Remplir /enset/bdcc**



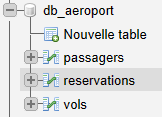
10. Supprimez les fichiers java.txt et cpp.txt dans HDFS.

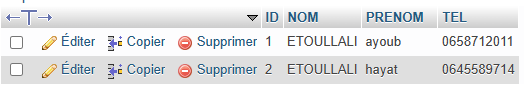


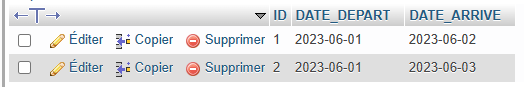


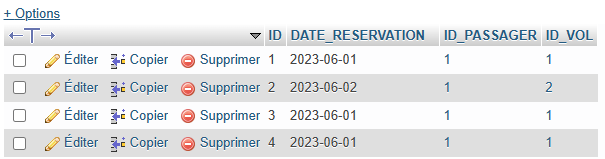
Exercice 2:

## Partie 1 : Spark SQL



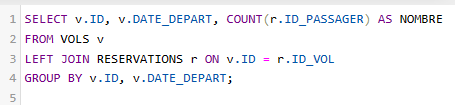


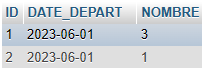




1. Afficher pour charque vol, le nombre de passagers selon le format d’affichage suivant :

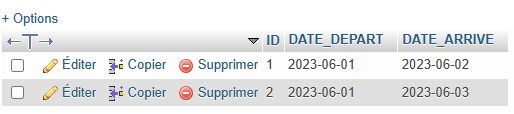
ID\_VOL |DATE DEPART| NOMBRE





2. Afficher la liste des vols en cours selon le format d’affichage suivant :

ID\_VOL |DATE DEPART| DATE ARRIVE

**Problème dans Xampp**

**Voir le code tel que la vérification des requetés sont dans MySQL ci-dessous**

La réponse dans les deux versions : Dataframes et QuerySql

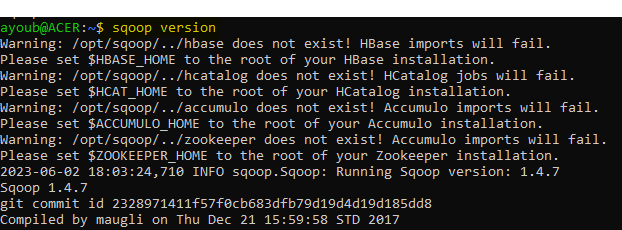
Dataframes

package ma.enset.dataMysql;  
  
import org.apache.log4j.Level;  
import org.apache.log4j.Logger;  
import org.apache.spark.sql.\*;  
  
public class Dataframes {  
  
 public static void main(String[] args) {  
  
 Logger.*getLogger*("org").setLevel(Level.*OFF*);  
  
 SparkSession spark = SparkSession.*builder*()  
 .appName("Spark SQL")  
 .config("spark.master", "local")  
 .getOrCreate();  
  
 Dataset<Row> vols\_df = spark.read()  
 .format("jdbc")  
 .option("url", "jdbc:mysql://localhost:3306/db\_aeroport")  
 .option("dbtable", "VOLS")  
 .option("user", "root")  
 .option("password", "")  
 .load();  
  
 Dataset<Row> passagers\_df = spark.read()  
 .format("jdbc")  
 .option("url", "jdbc:mysql://localhost:3306/db\_aeroport")  
 .option("dbtable", "PASSAGERS")  
 .option("user", "root")  
 .option("password", "")  
 .load();  
  
 Dataset<Row> reservations\_df = spark.read()  
 .format("jdbc")  
 .option("url", "jdbc:mysql://localhost:3306/db\_aeroport")  
 .option("dbtable", "RESERVATIONS")  
 .option("user", "root")  
 .option("password", "")  
 .load();  
  
 // Question 1: Afficher le nombre de passagers par vol  
 Dataset<Row> resultat\_question1 = vols\_df.join(reservations\_df, vols\_df.col("ID\_VOL").equalTo(reservations\_df.col("ID\_VOL")), "left")  
 .groupBy(vols\_df.col("ID\_VOL"), vols\_df.col("DATE\_DEPART"))  
 .agg(functions.*count*(reservations\_df.col("ID\_PASSAGER")).alias("NOMBRE"));  
  
 resultat\_question1.show();  
  
 // Question 2: Afficher la liste des vols en cours  
 Dataset<Row> resultat\_question2 = vols\_df.select(vols\_df.col("ID\_VOL"), vols\_df.col("DATE\_DEPART"), vols\_df.col("DATE\_ARRIVE"));  
 resultat\_question2.show();  
  
 // Fermeture de la session Spark  
 spark.stop(); }}

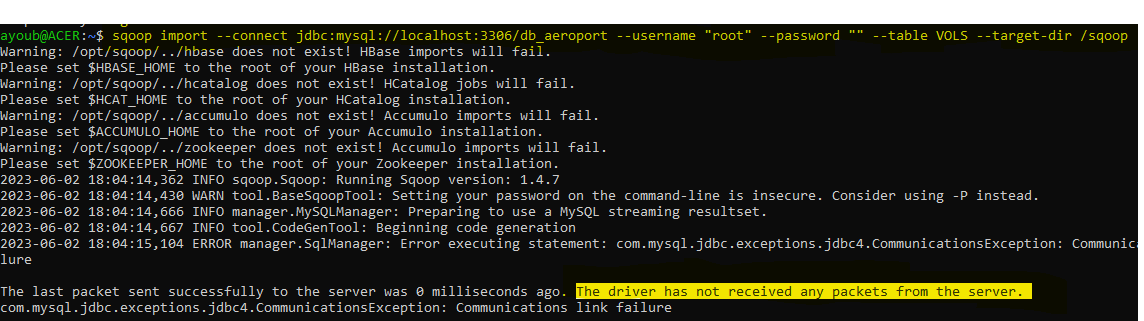
QuerySql

package ma.enset.dataMysql;  
  
import org.apache.log4j.Level;  
import org.apache.log4j.Logger;  
import org.apache.spark.sql.Dataset;  
import org.apache.spark.sql.Row;  
import org.apache.spark.sql.SparkSession;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class QuerySql {  
  
 public static void main(String[] args) {  
  
 Logger.*getLogger*("org").setLevel(Level.*OFF*);  
 SparkSession ss = SparkSession  
 .*builder*()  
 .master("local[\*]")  
 .appName("Traitement de données stockées dans Mysql [Query SQL]")  
 .getOrCreate();  
  
 Map<String, String> options = new HashMap<>();  
 options.put("driver", "com.mysql.cj.jdbc.Driver");  
 options.put("url", "jdbc:mysql://localhost:3306/db\_aeroport");  
 options.put("user", "root");  
 options.put("password", "");  
  
 // Question 1: Afficher le nombre de passagers par vol  
 Dataset<Row> df1 = ss.read().format("jdbc")  
 .options(options)  
 .option("query", "SELECT v.ID, v.DATE\_DEPART, COUNT(r.ID\_PASSAGER) AS NOMBRE\n" +  
 "FROM VOLS v\n" +  
 "LEFT JOIN RESERVATIONS r ON v.ID = r.ID\_VOL\n" +  
 "GROUP BY v.ID, v.DATE\_DEPART;\n")  
 .load();  
 df1.show();  
 // Question 2: Afficher la liste des vols en cours  
 Dataset<Row> df2 = ss.read().format("jdbc")  
 .options(options)  
 .option("query", "SELECT ID\_VOL, DATE\_DEPART, DATE\_ARRIVE\n" +  
 "FROM VOLS;\n")  
 .load();  
 df2.show();  
 }  
}

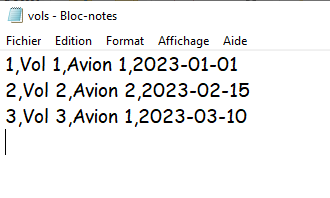
## Partie 2 : Importer et exporter des données avec SQOOP



**Sqoop ça ne marche pas, j’ai un problème dans MySQL**



Voici le fichier vols.txt



**Sqoop ça ne marche pas**

**Voir le code ci-dessous :**

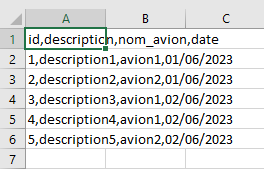
1. Importation des données de la table VOLS dans HDFS :

sqoop import --connect jdbc:mysql://localhost:3306/db\_aeroport --username "root" --password "" --table VOLS --target-dir /sqoop

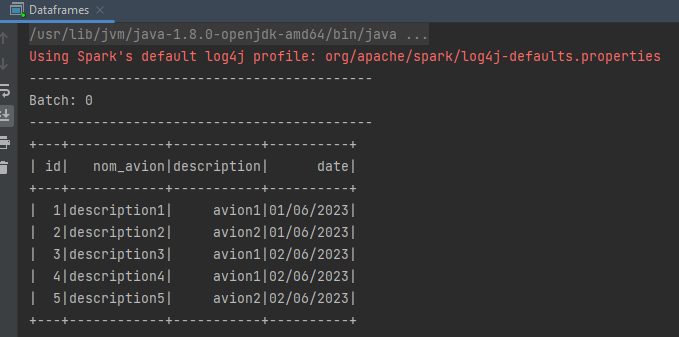
1. Exportation des données du fichier vols.txt vers la table VOLS en utilisant Sqoop :

sqoop export --connect jdbc:mysql://localhost:3306/db\_aeroport --username "root" --password "" --table VOLS --export-dir /aeroport/vols.txt

## Partie 3: Traitement de données en streaming

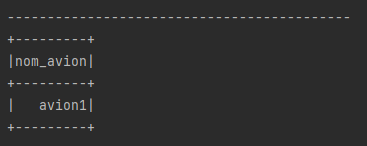




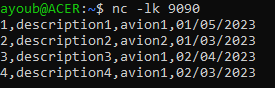


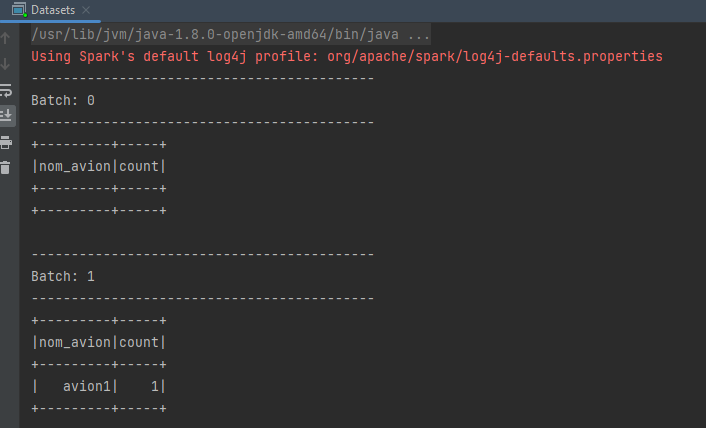
1. Afficher d’une manière continue l’avion ayant plus d’incidents.

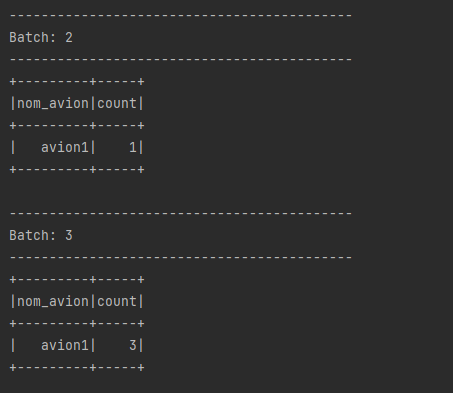
Ici c’est l’avion « avion1 » (3 incidents)



**Avec DataStreaming :**





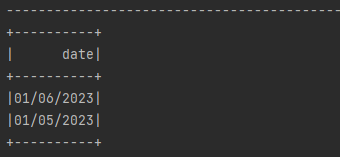


2. Afficher d’une manière continue les deux mois de l’année en cours où il a y avait moins d’incidents.

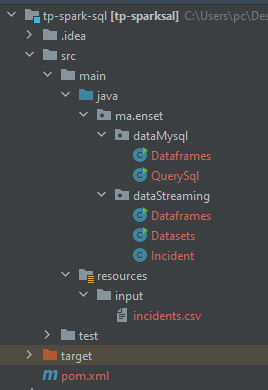
J’ai changé la base de données :

id,description,nom\_avion,date  
1,description1,avion1,01/05/2023  
2,description2,avion2,01/03/2023  
3,description3,avion1,01/04/2023  
4,description4,avion1,01/03/2023  
5,description5,avion2,01/06/2023  
6,description6,avion2,01/04/2023

Alors les deux mois de l’année en cours où il a y avait moins d’incidents sont les mois 05 et 06



**Projet :**



Conclusion

**Au cours de cet examen, j'ai pu manipuler le système de fichiers HDFS, charger des fichiers, créer des répertoires et effectuer diverses opérations sur les données. J'ai également utilisé Spark SQL pour l'analyse et le traitement des données distribuées, ce qui m'a permis d'extraire des informations utiles et de prendre des décisions éclairées.**

**Cette expérience m'a permis de mieux comprendre les défis et les opportunités liés à la gestion des données à grande échelle. J'ai réalisé l'importance de pouvoir travailler efficacement sur des ensembles de données massifs et de savoir utiliser des outils tels que Spark pour optimiser les performances et obtenir des résultats précis.**