Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

Département Mathématiques et Informatique

« Ingénierie Informatique : Big Data et Cloud Computing » II-BDCC2

DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

Spark Sql

Filière :
« Ingénierie Informatique : Big Data et Cloud Computing »
II-BDCC$_2$

Créé par : **MOHAMED AIT LAHCEN**

Lien vers le code source sur github :

https://github.com/mohamed-ait/Spark-TPs/tree/main/sparkSql

● L'énoncé de l'exercice :

```
Employe :

   id :long
   name :string
   phone : string
   salary : double
   age : int
   departement : string


   1-affciher les employés ayant un age entre 30 et 35
   2-affciher la moyenne des salaires de chaque dpartement
   3-afficher le nombre de salariés par département
   4-afficher le salaire maximum de tous les départements
```

● Les fichiers qui contiennent les données à traiter :

1. Fichier CSV :

id	name	phone	salary	age	departement
1	employe1	0625124584	10000	35	informatique
2	employe2	0625123584	10000	45	physique
3	employe3	0625121584	20000	35	informatique
4	employe4	0625155584	15000	25	economique
5	employe5	0625124584	15000	25	economique
6	employe6	0625125484	25000	50	physique

2. Fichier JSON :

```json
[
  {
    "id" : 1,
    "name" : "employe1",
    "phone" : "0625015242",
    "salary" : 10000,
    "age" : 30,
    "departement" : "informatique"
},
  {
    "id" : 2,
    "name" : "employe2",
    "phone" : "0625142154",
    "salary" : 15000,
    "age" : 32,
    "departement" : "mathematique"
},
```

- **La déclaration de la classe sérialisé :**

```java
public class Employe implements Serializable {
    3 usages
    private long id;
    3 usages
    private String name;
    3 usages
    private String phone;
    3 usages
    private double salary;
    3 usages
    private long age;
    3 usages
    private String departement;
    mohamed-ait
    public Employe() {

    }
```

- **La récupération et le stockage des données dans un dataFrame :**

1. Cas du fichier CSV :

```
SparkConf conf = new SparkConf().setAppName("saprk sql").setMaster("local[*]");
SparkSession ss=SparkSession.builder().
        appName("TP Spark SQL").
        master("local[*]").getOrCreate();
Encoder<Employe> employeEncoder= Encoders.bean(Employe.class);
Dataset<Employe> ds = ss.read().option("header", true).option("inferSchema",true)
        .option("delimiter",",")
        .csv( path: "employes.csv").as(employeEncoder);        //ds.printSchema();
```

2. Cas du fichier JSON:

```
mohamed-ait *
public static void main(String[] args) {
    SparkConf conf = new SparkConf().setAppName("saprk sql").setMaster("local[*]");
    SparkSession ss=SparkSession.builder().
            appName("TP Spark SQL").
            master("local[*]").getOrCreate();
    Encoder<Employe> employeEncoder= Encoders.bean(Employe.class);
    Dataset<Employe> ds = ss.read().option("multiline", true).json( path: "employes.json").as(employeEncoder);
```

1-La liste des employée qui ont un age entre 30 et 35 :

```
//les employés qui ont un age entre 30 et 35
ds.filter((FilterFunction<Employe>) employeeBean -> employeeBean.getAge()>=30 && employeeBean.getAge()<=35).show();
```

```
|age| departement| id|    name|     phone|salary|
+---+------------+---+--------+----------+------+
| 30|informatique|  1|employe1|0625015242| 10000|
| 32|mathematique|  2|employe2|0625142154| 15000|
| 34|  économique|  3|employe3|0625015242|  9000|
+---+------------+---+--------+----------+------+
```

2-La moyenne des salaires par département :

```
//la moyenne des salaires par departement :
ds.groupBy(col( colName: "departement")).avg( ...colNames: "salary").show();
```

```
+-------------+-----------+
| departement|avg(salary)|
+-------------+-----------+
|mathematique|    15000.0|
|informatique|    10000.0|
|  économique|     9000.0|
+-------------+-----------+
```

3-Le nombre des employés par département :

```
//le nombre des employés par département :
ds.groupBy(col( colName: "departement")).count().show();
```

```
+-------------+-----+
|  departement|count|
+-------------+-----+
|mathematique|     1|
|informatique|     1|
|   économique|     1|
+-------------+-----+
```

4-Le salaire maximum de tous les départements :

```
// le salaire maximum de tous les départements :
ds.select(max( columnName: "salary")).show();
```

```
+-----------+
|max(salary)|
+-----------+
|      15000|
+-----------+
```