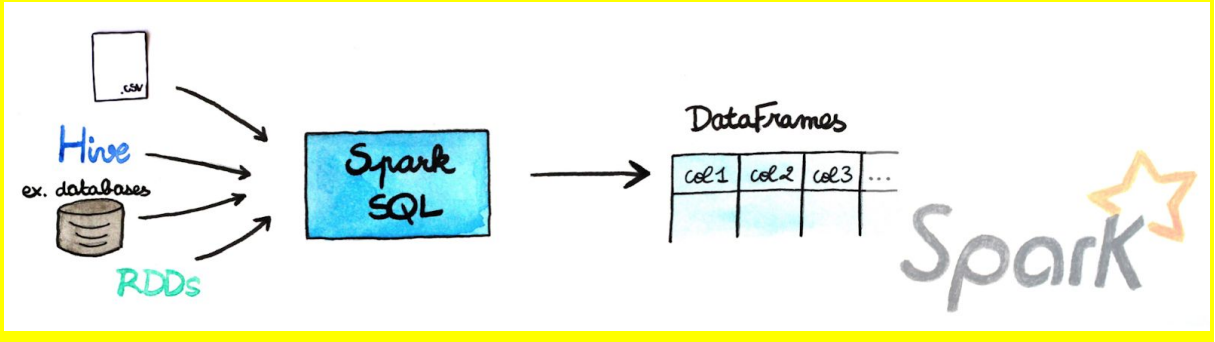
Spark SQL-Java Application: Read CSV file into Data Frame and Execute some Queries With Spark SQL And Java

I. Introduction:

In this documentation, we are focused to parse data from a CSV file, perform some queries and output the result in the output using the Spark Core and Spark SQL APIs, and also Java.



II. Technologies:

- Java 8
- Spark Core 3.0.1
- Spark SQL 3.0.1
- Maven
- Intellij IDEA

III. Implement some queries using java and spark:

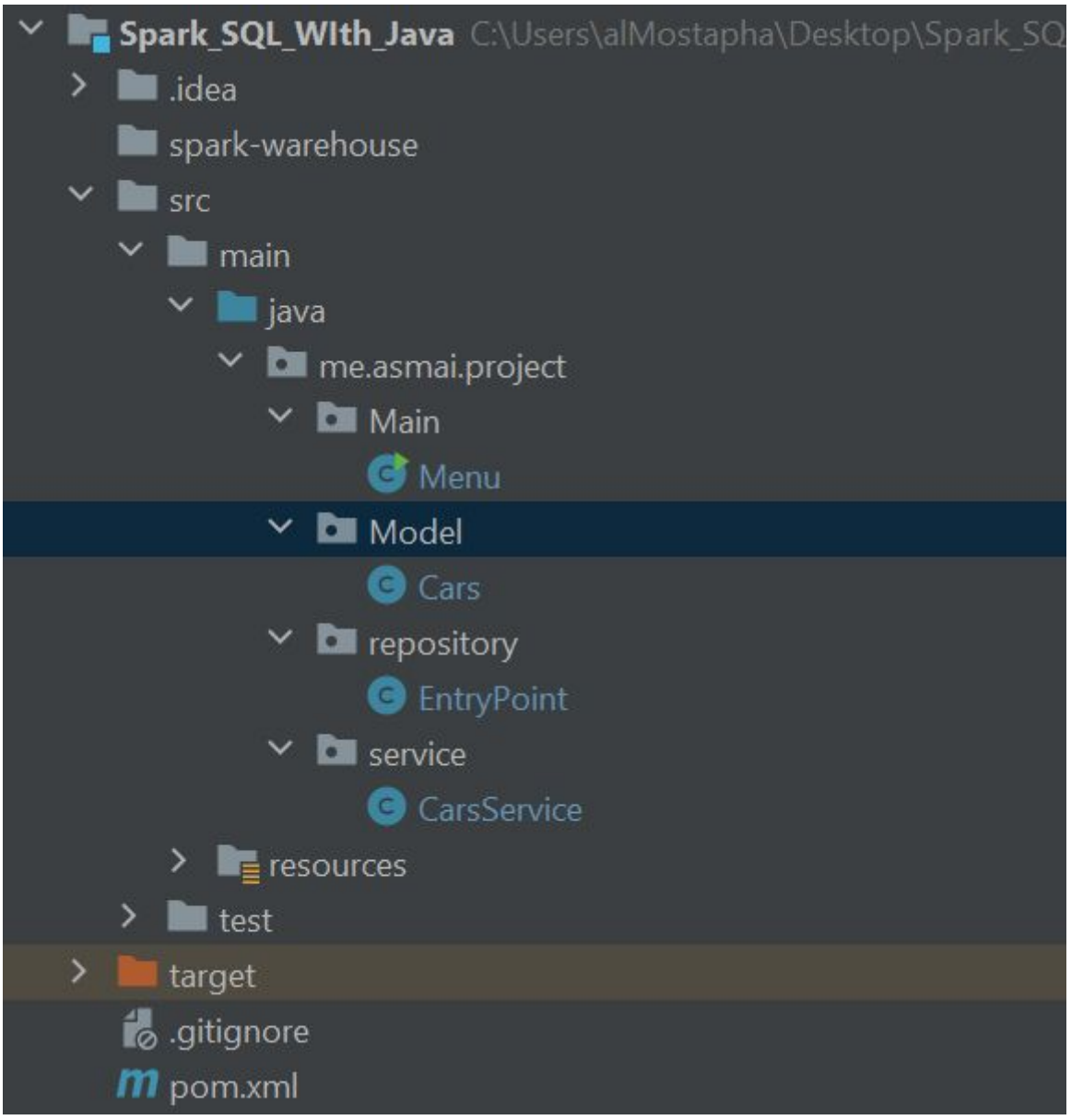
-Let's have a look at the **cars** dataset which we will use for this queries:

Car	MPG	Cylinders	Displacemen	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet Ch	18.0	8	307.0	130.0	3504	12.0	70	US
Buick Skylark	15.0	8	350.0	165.0	3693	11.5	70	US
Plymouth Sat	18.0	8	318.0	150.0	3436	11.0	70	US
AMC Rebel S	16.0	8	304.0	150.0	3433	12.0	70	US
Ford Torino	17.0	8	302.0	140.0	3449	10.5	70	US
Ford Galaxie	15.0	8	429.0	198.0	4341	10.0	70	US
Chevrolet Im	14.0	8	454.0	220.0	4354	9.0	70	US
Plymouth Fu	14.0	8	440.0	215.0	4312	8.5	70	US
Pontiac Catal	14.0	8	455.0	225.0	4425	10.0	70	US
AMC Ambass	15.0	8	390.0	190.0	3850	8.5	70	US
Citroen DS-2	0	4	133.0	115.0	3090	17.5	70	Europe
Chevrolet Ch	0	8	350.0	165.0	4142	11.5	70	US
Ford Torino (	0	8	351.0	153.0	4034	11.0	70	US
Plymouth Sat	0	8	383.0	175.0	4166	10.5	70	US
AMC Rebel S	0	8	360.0	175.0	3850	11.0	70	US
Dodge Challe	15.0	8	383.0	170.0	3563	10.0	70	US
Plymouth 'Cu	14.0	8	340.0	160.0	3609	8.0	70	US
Ford Mustan	0	8	302.0	140.0	3353	8.0	70	US
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

-These are **queries** to be exported:

- Get all Cars from csv file
- Get Cars By Model
- Get Model of Cars By less HorsePower
- Get Cars sorted by Model and HorsePower.
- Get Cars between two models and Origin and sorted by HorsePower
- Get Cars by Origin and sorted by Model

IV. Project Structure :



V. Setup Dependencies on pom.xml:

After Adding the below dependencies on **pom.xml**, It will download all the required packages.

```xml
<groupId>org.example</groupId>
<artifactId>Test-Spark-With-Java</artifactId>
<version>1.0-SNAPSHOT</version>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <configuration>
                <source>11</source>
                <target>11</target>
            </configuration>
        </plugin>
    </plugins>
</build>

<dependencies>
    <!-- Dependency of Apache Spark Core-->
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.12</artifactId>
        <version>3.0.1</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-sql -->
    <dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-sql_2.12</artifactId>
        <version>3.0.1</version>
    </dependency>
</dependencies>
```

VI. Configure Log4j file on spark console :

I'd like to stop various **INFO messages** that are coming on the spark console to get just the result on the console without logging messages.

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/C:/Users/alMostapha/.m2/repository/org/apache/spark/spark-
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
+--------------------+----+---------+-----------+----------+------+------------+-----+------+
|                 Car| MPG|Cylinders|Displacement|Horsepower|Weight|Acceleration|Model|Origin|
```

I edit the **log4j.properties** file in order to stop these messages. Here are the contents of **log4j.properties:**

```
#Stop INFO messages displaying on Spark console to get just the result expected
log4j.rootCategory=ERROR, console
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.target=System.err
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern=%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n
```

VII. Define Data Model:

In the **model** package, we define **Cars** class.
model/Cars.class:

```
public class Cars {

    private String car;

    private double MPG;

    private int Cylinders;

    private double Displacement;

    private double Horsepower;

    private double Weight;

    private double Acceleration;

    private double Model;

    private String Origin;

    public Cars(String car, double MPG, int cylinders, double displacement, double horsepower, double weight, double acceleration, double model, String origin) {...}

    public String getCar() { return car; }

    public void setCar(String car) { this.car = car; }

    public double getMPG() { return MPG; }

    public void setMPG(double MPG) { this.MPG = MPG; }
```

```
    public String getCar() { return car; }

    public void setCar(String car) { this.car = car; }

    public double getMPG() { return MPG; }

    public void setMPG(double MPG) { this.MPG = MPG; }

    public double getCylinders() { return Cylinders; }

    public void setCylinders(int cylinders) { Cylinders = cylinders; }

    public double getDisplacement() { return Displacement; }

    public void setDisplacement(double displacement) { Displacement = displacement; }

    public double getHorsepower() { return Horsepower; }

    public void setHorsepower(double horsepower) { Horsepower = horsepower; }

    public double getWeight() { return Weight; }

    public void setWeight(double weight) { Weight = weight; }

    public double getAcceleration() { return Acceleration; }

    public void setAcceleration(double acceleration) { Acceleration = acceleration; }

    public double getModel() { return Model; }

    public void setModel(double model) { Model = model; }
```

VIII. Create a Repository to working with Dataframe(cars.csv):

Let's create a repository to interact with **Orders** from the csv file.
In the **repository** package, create a class **EntryPoint** which is responsible for reading **CSV file** and loading the data into a **spark dataframe** with a custom schema.

```
import org.apache.spark.sql.types.StructType;

public class EntryPoint {

    public EntryPoint() { }

    public static SparkSession sparkSession(){
        return SparkSession
                .builder()
                .appName(" Application with Spark SQL and Java")
                .master("local[*]")
                .getOrCreate();
    }

    public static Dataset<Cars> getDataset(){
        Encoder<Cars> carsEncoder = Encoders.bean(Cars.class);

        Dataset<Cars> carsDataset = sparkSession().read()
                .option("header","true")
                .option("treatEmptyValuesAsNulls", "true")
                .option("inferSchema", "true")
                .option("mode","DROPMALFORMED")
                .option("delimiter",";")
                .csv( path: "src/main/resources/cars.csv")
                .as(carsEncoder);

        carsDataset.registerTempTable( tableName: "cars");
        return carsDataset;
    }
}
```

IX. Create a Spark Service:

CarsService class uses **Repository/EntryPoint** class for 5 functions:

→ `getAllCars(int numberRows)`: Get all Cars from csv file
→ `getCarsByModel(double model)`: Get Cars By Model

→ `getModelOfCarsByLessHorsePower()`: Get only the Model of Cars which have less HorsePower.
→ `getCarsSortedByModelAndHorsePower()`**:** Get Cars sorted by Model and HorsePower.
→`getCarsBetweenTwoModelsOfAnOriginAndSortedByHorsePower(double model1, double model2, String origin`**:** Get Cars between two models and Origin and sorted by HorsePower.
→ `getCarsByOriginAndSortedByModel(String origin)` : Get Cars by Origin and sorted by Model.

Here is the code of **service/CarsService.java:**

```
public class CarsService {

    public void getAllCars(int numberRows) { getDataset().show(numberRows); }

    public void getCarsByModel(double model) {
        Dataset<Cars> cars = getDataset().filter("Model == \"" + model + "\"");
        cars.show((int) getDataset().count());
    }


    public void getModelOfCarsByLessHorsePower() {
        Dataset<Row> cars = sparkSession().sql( sqlText: "SELECT Model,MIN(Horsepower) FROM cars GROUP BY Model");
        cars.show((int) getDataset().count());
    }

    public void getCarsSortedByModelAndHorsePower(){
        Dataset<Cars> cars = getDataset().sort( sortCol: "Model", ...sortCols: "Horsepower"); // Or create a Comparator
        cars.show((int) getDataset().count());
    }

    public void getCarsBetweenTwoModelsOfAnOriginAndSortedByHorsePower(double model1, double model2, String origin){
        Dataset<Cars> cars = getDataset().filter((FilterFunction<Cars>) car -> car.getOrigin().equals(origin))
        .filter("Model >= \"" + model1 + "\"")
        .filter("Model <= \"" + model2 + "\"")
        .sort( sortCol: "Horsepower");
        cars.show((int) getDataset().count());
    }

    public void getCarsByOriginAndSortedByModel(String origin){
        Dataset<Cars> cars = getDataset().filter((FilterFunction<Cars>) car -> car.getOrigin().equals(origin))
        .sort( sortCol: "Model");
        cars.show((int) getDataset().count());
    }
```

X. Creating a Menu Driven Program :

Let's create a **Menu** class under package **Main** to obtain input from a user by displaying a list of options.

main/Menu.java:

```java
public class Menu {

    public static Scanner scanner = new Scanner(System.in);
    public static CarsService carsService = new CarsService();

    public static void main(String[] args) {

        try {
            int menuOption = 0;
            double model1;
            double model;
            double model2;
            String origin;

            do {
                menuOption = showMenu();

                switch (menuOption) {

                    case 1:
                        System.out.println("Enter the rows number of Cars that you want preview : ");
                        int numberRows = scanner.nextInt();
                        carsService.getAllCars(numberRows);
                        break;
                    case 2:
                        System.out.println("Enter the model of car :  Ex= 70");
                        scanner.nextLine();//// Adding nextLine just to discard the old \n character
                        model = scanner.nextInt();
                        carsService.getCarsByModel(model);
                        break;
                    case 3:
                        carsService.getModelOfCarsByLessHorsePower();
```

```java
                    break;
                case 2:
                    System.out.println("Enter the model of car :  Ex= 70");
                    scanner.nextLine();//// Adding nextLine just to discard the old \n character
                    model = scanner.nextInt();
                    carsService.getCarsByModel(model);
                    break;
                case 3:
                    carsService.getModelOfCarsByLessHorsePower();
                    break;
                case 4:
                    carsService.getCarsSortedByModelAndHorsePower();
                    break;
                case 5:
                    System.out.println("Enter the model of the car : Ex: 80");
                    scanner.nextLine();
                    model1 = scanner.nextInt();
                    System.out.println("Enter the second model of the car : Ex: 81");
                    scanner.nextLine();
                    model2 = scanner.nextInt();
                    System.out.println("Enter the origin of the car : Ex: US");
                    scanner.nextLine();
                    origin = scanner.nextLine();
                    carsService.getCarsBetweenTwoModelsOfAnOriginAndSortedByHorsePower(model1,model2,origin);
                    break;
                case 6:
                    System.out.println("Enter the origin of the car : Ex: Japan");
                    scanner.nextLine();
                    origin = scanner.nextLine();
                    carsService.getCarsByOriginAndSortedByModel(origin);
                case 7:
                    System.out.println("Quitting Program...");
                    break;
                default:
```

```
                default:
                    System.out.println("Sorry, please enter a valid Option");
                }

            } while (menuOption != 7);

            // Exiting message when user decides to quit Program
            System.out.println("Thanks for using this Program...");

        } catch (Exception ex) {
            System.out.println("Sorry problem occured within Program");
            scanner.next();
        } finally {
            scanner.close();
        }
    }
    public static int showMenu() {
        int option = 0;

        System.out.println("Menu:");
        System.out.println("1. Get All Cars form CSV file ");
        System.out.println("2. Get Cars By Model");
        System.out.println("3. Get Model of Cars By Less HorsePower");
        System.out.println("4. Get Cars Sorted by Model and HorsePower");
        System.out.println("5. Get Cars By Origin and Between Two Models and Sorted by HorsePower");
        System.out.println("6. Get Cars by Origin and Sorted by Model");
        System.out.println("7. Quit Program");

        // Getting query number from above menu
        System.out.println("Enter the number of Query from above...");
        option = scanner.nextInt();

        return option;
    }
```

XI. Output:

While executing each query , you will be able to see below its content in the console.

1. ***Menu:***

```
Menu:
1. Get All Cars form CSV file
2. Get Cars By Model
3. Get Model of Cars By Less HorsePower
4. Get Cars Sorted by Model and HorsePower
5. Get Cars By Origin and Between Two Models and Sorted by HorsePower
6. Get Cars by Origin and Sorted by Model
7. Quit Program
Enter the number of Query from above...
```

2. ***First Query:***

```
Enter the number of Query from above...
1
Enter the rows number of Cars that you want preview :
30
```

Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origin
Chevrolet Chevell...	18.0	8	307.0	130.0	3504	12.0	70	US
Buick Skylark 320	15.0	8	350.0	165.0	3693	11.5	70	US
Plymouth Satellite	18.0	8	318.0	150.0	3436	11.0	70	US
AMC Rebel SST	16.0	8	304.0	150.0	3433	12.0	70	US
Ford Torino	17.0	8	302.0	140.0	3449	10.5	70	US
Ford Galaxie 500	15.0	8	429.0	198.0	4341	10.0	70	US
Chevrolet Impala	14.0	8	454.0	220.0	4354	9.0	70	US
Plymouth Fury iii	14.0	8	440.0	215.0	4312	8.5	70	US
Pontiac Catalina	14.0	8	455.0	225.0	4425	10.0	70	US
AMC Ambassador DPL	15.0	8	390.0	190.0	3850	8.5	70	US
Citroen DS-21 Pallas	0.0	4	133.0	115.0	3090	17.5	70	Europe
Chevrolet Chevell...	0.0	8	350.0	165.0	4142	11.5	70	US
Ford Torino (sw)	0.0	8	351.0	153.0	4034	11.0	70	US
Plymouth Satellit...	0.0	8	383.0	175.0	4166	10.5	70	US
AMC Rebel SST (sw)	0.0	8	360.0	175.0	3850	11.0	70	US
Dodge Challenger SE	15.0	8	383.0	170.0	3563	10.0	70	US
Plymouth 'Cuda 340	14.0	8	340.0	160.0	3609	8.0	70	US
Ford Mustang Boss...	0.0	8	302.0	140.0	3353	8.0	70	US
Chevrolet Monte C...	15.0	8	400.0	150.0	3761	9.5	70	US
Buick Estate Wago...	14.0	8	455.0	225.0	3086	10.0	70	US
Toyota Corolla Ma...	24.0	4	113.0	95.0	2372	15.0	70	Japan

3. Second Query :

```
Enter the number of Query from above...
2
Enter the model of car :  Ex= 70
71
```

Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model	Origin
Datsun PL510	27.0	4	97.0	88.0	2130	14.5	71	Japan
Chevrolet Vega 2300	28.0	4	140.0	90.0	2264	15.5	71	US
Toyota Corolla	25.0	4	113.0	95.0	2228	14.0	71	Japan
Ford Pinto	25.0	4	98.0	0.0	2046	19.0	71	US
Volkswagen Super ...	0.0	4	97.0	48.0	1978	20.0	71	Europe
AMC Gremlin	19.0	6	232.0	100.0	2634	13.0	71	US
Plymouth Satellit...	16.0	6	225.0	105.0	3439	15.5	71	US
Chevrolet Chevell...	17.0	6	250.0	100.0	3329	15.5	71	US
Ford Torino 500	19.0	6	250.0	88.0	3302	15.5	71	US
AMC Matador	18.0	6	232.0	100.0	3288	15.5	71	US
Chevrolet Impala	14.0	8	350.0	165.0	4209	12.0	71	US
Pontiac Catalina ...	14.0	8	400.0	175.0	4464	11.5	71	US
Ford Galaxie 500	14.0	8	351.0	153.0	4154	13.5	71	US
Plymouth Fury iii	14.0	8	318.0	150.0	4096	13.0	71	US
Dodge Monaco (sw)	12.0	8	383.0	180.0	4955	11.5	71	US
Ford Country Squi...	13.0	8	400.0	170.0	4746	12.0	71	US
Pontiac Safari (sw)	13.0	8	400.0	175.0	5140	12.0	71	US
AMC Hornet Sporta...	18.0	6	258.0	110.0	2962	13.5	71	US
Chevrolet Vega (sw)	22.0	4	140.0	72.0	2408	19.0	71	US
Pontiac Firebird	19.0	6	250.0	100.0	3282	15.0	71	US
Ford Mustang	18.0	6	250.0	88.0	3139	14.5	71	US
Mercury Capri 2000	23.0	4	122.0	86.0	2220	14.0	71	US
Opel 1900	28.0	4	116.0	90.0	2123	14.0	71	Europe
Peugeot 304	30.0	4	79.0	70.0	2074	19.5	71	Europe
Fiat 124B	30.0	4	88.0	76.0	2065	14.5	71	Europe

4. *Third Query :*

```
Enter the number of Query from above...
3
+-----+---------------+
|Model|min(Horsepower)|
+-----+---------------+
|   78|           48.0|
|   81|            0.0|
|   76|           52.0|
|   72|           54.0|
|   77|           58.0|
|   82|            0.0|
|   80|            0.0|
|   73|           46.0|
|   70|           46.0|
|   75|           53.0|
|   71|            0.0|
|   79|           65.0|
|   74|            0.0|
+-----+---------------+
```

5. Fourth Query :

```
Enter the number of Query from above...
4
+-------------------+----+---------+------------+----------+------+------------+-----+------+
|                Car| MPG|Cylinders|Displacement|Horsepower|Weight|Acceleration|Model|Origin|
+-------------------+----+---------+------------+----------+------+------------+-----+------+
|Volkswagen 1131 D...|26.0|        4|        97.0|      46.0|  1835|        20.5|   70|Europe|
|       Ford Maverick|21.0|        6|       200.0|      85.0|  2587|        16.0|   70|    US|
|         Peugeot 504|25.0|        4|       110.0|      87.0|  2672|        17.5|   70|Europe|
|        Datsun PL510|27.0|        4|        97.0|      88.0|  2130|        14.5|   70| Japan|
|         Audi 100 LS|24.0|        4|       107.0|      90.0|  2430|        14.5|   70|Europe|
|         AMC Gremlin|21.0|        6|       199.0|      90.0|  2648|        15.0|   70|    US|
|            Saab 99e|25.0|        4|       104.0|      95.0|  2375|        17.5|   70|Europe|
|Toyota Corolla Ma...|24.0|        4|       113.0|      95.0|  2372|        15.0|   70| Japan|
|     Plymouth Duster|22.0|        6|       198.0|      95.0|  2833|        15.5|   70|    US|
|          AMC Hornet|18.0|        6|       199.0|      97.0|  2774|        15.5|   70|    US|
|            BMW 2002|26.0|        4|       121.0|     113.0|  2234|        12.5|   70|Europe|
|Citroen DS-21 Pallas| 0.0|        4|       133.0|     115.0|  3090|        17.5|   70|Europe|
|Chevrolet Chevell...|18.0|        8|       307.0|     130.0|  3504|        12.0|   70|    US|
|         Ford Torino|17.0|        8|       302.0|     140.0|  3449|        10.5|   70|    US|
|Ford Mustang Boss...| 0.0|        8|       302.0|     140.0|  3353|         8.0|   70|    US|
|       AMC Rebel SST|16.0|        8|       304.0|     150.0|  3433|        12.0|   70|    US|
|Chevrolet Monte C...|15.0|        8|       400.0|     150.0|  3761|         9.5|   70|    US|
|  Plymouth Satellite|18.0|        8|       318.0|     150.0|  3436|        11.0|   70|    US|
|    Ford Torino (sw)| 0.0|        8|       351.0|     153.0|  4034|        11.0|   70|    US|
|  Plymouth 'Cuda 340|14.0|        8|       340.0|     160.0|  3609|         8.0|   70|    US|
|   Buick Skylark 320|15.0|        8|       350.0|     165.0|  3693|        11.5|   70|    US|
|Chevrolet Chevell...| 0.0|        8|       350.0|     165.0|  4142|        11.5|   70|    US|
| Dodge Challenger SE|15.0|        8|       383.0|     170.0|  3563|        10.0|   70|    US|
|  AMC Rebel SST (sw)| 0.0|        8|       360.0|     175.0|  3850|        11.0|   70|    US|
|Plymouth Satellit...| 0.0|        8|       383.0|     175.0|  4166|        10.5|   70|    US|
|  AMC Ambassador DPL|15.0|        8|       390.0|     190.0|  3850|         8.5|   70|    US|
```

6. *Fifth Query:*

```
Enter the number of Query from above...
5
Enter the model of the car : Ex: 80
80
Enter the second model of the car : Ex: 81
81
Enter the origin of the car : Ex: US
Japan
+--------------------+----+---------+------------+----------+------+------------+-----+------+
|                 Car| MPG|Cylinders|Displacement|Horsepower|Weight|Acceleration|Model|Origin|
+--------------------+----+---------+------------+----------+------+------------+-----+------+
|      Toyota Starlet|39.1|        4|        79.0|      58.0|  1755|        16.9|   81| Japan|
|Toyota Corolla Te...|38.1|        4|        89.0|      60.0|  1968|        18.8|   80| Japan|
|    Honda Civic 1300|35.1|        4|        81.0|      60.0|  1760|        16.1|   81| Japan|
|       Toyota Tercel|37.7|        4|        89.0|      62.0|  2050|        17.3|   81| Japan|
|           Mazda GLC|46.6|        4|        86.0|      65.0|  2110|        17.9|   80| Japan|
|          Datsun 210|40.8|        4|        85.0|      65.0|  2110|        19.2|   80| Japan|
|      Datsun 210 MPG|37.0|        4|        85.0|      65.0|  1975|        19.4|   81| Japan|
|          Datsun 310|37.2|        4|        86.0|      65.0|  2019|        16.4|   80| Japan|
|           Subaru DL|33.8|        4|        97.0|      67.0|  2145|        18.0|   80| Japan|
|              Subaru|32.3|        4|        97.0|      67.0|  2065|        17.8|   81| Japan|
| Honda Civic 1500 gl|44.6|        4|        91.0|      67.0|  1850|        13.8|   80| Japan|
|         Mazda GLC 4|34.1|        4|        91.0|      68.0|  1985|        16.0|   81| Japan|
|        Honda Accord|32.4|        4|       107.0|      72.0|  2290|        17.0|   80| Japan|
|           Mazda 626|31.6|        4|       120.0|      74.0|  2635|        18.3|   81| Japan|
|      Toyota Corolla|32.2|        4|       108.0|      75.0|  2265|        15.2|   80| Japan|
|       Honda Prelude|33.7|        4|       107.0|      75.0|  2210|        14.4|   81| Japan|
|           Mazda 626|31.3|        4|       120.0|      75.0|  2542|        17.5|   80| Japan|
|      Toyota Corolla|32.4|        4|       108.0|      75.0|  2350|        16.8|   81| Japan|
|Toyota Corolla Li...|29.8|        4|       134.0|      90.0|  2711|        15.5|   80| Japan|
|Datsun 510 Hatchback|37.0|        4|       119.0|      92.0|  2434|        15.0|   80| Japan|
|       Mazda RX-7 GS|23.7|        3|        70.0|     100.0|  2420|        12.5|   80| Japan|
|        Datsun 200SX|32.9|        4|       119.0|     100.0|  2615|        14.8|   81| Japan|
|     Toyota Cressida|25.4|        6|       168.0|     116.0|  2900|        12.6|   81| Japan|
|   Datsun 810 Maxima|24.2|        6|       146.0|     120.0|  2930|        13.8|   81| Japan|
|       Datsun 280-ZX|32.7|        6|       168.0|     132.0|  2910|        11.4|   80| Japan|
+--------------------+----+---------+------------+----------+------+------------+-----+------+
```

7. *Sixth Query :*

```
Enter the number of Query from above...
6
Enter the origin of the car : Ex: Japan
Japan
+--------------------+----+---------+------------+----------+------+------------+-----+------+
|                 Car| MPG|Cylinders|Displacement|Horsepower|Weight|Acceleration|Model|Origin|
+--------------------+----+---------+------------+----------+------+------------+-----+------+
|Toyota Corolla Ma...|24.0|        4|       113.0|      95.0|  2372|        15.0|   70| Japan|
|        Datsun PL510|27.0|        4|        97.0|      88.0|  2130|        14.5|   70| Japan|
| Toyota Corolla 1200|31.0|        4|        71.0|      65.0|  1773|        19.0|   71| Japan|
|        Datsun PL510|27.0|        4|        97.0|      88.0|  2130|        14.5|   71| Japan|
|         Datsun 1200|35.0|        4|        72.0|      69.0|  1613|        18.0|   71| Japan|
|      Toyota Corolla|25.0|        4|       113.0|      95.0|  2228|        14.0|   71| Japan|
|Toyota Corolla Ma...|23.0|        4|       120.0|      97.0|  2506|        14.5|   72| Japan|
|Toyota Corolla 16...|27.0|        4|        97.0|      88.0|  2100|        16.5|   72| Japan|
|Toyota Corolla Ha...|24.0|        4|       113.0|      95.0|  2278|        15.5|   72| Japan|
|     Mazda RX2 Coupe|19.0|        3|        70.0|      97.0|  2330|        13.5|   72| Japan|
|     Datsun 510 (sw)|28.0|        4|        97.0|      92.0|  2288|        17.0|   72| Japan|
|          Datsun 610|22.0|        4|       108.0|      94.0|  2379|        16.5|   73| Japan|
|           Mazda RX3|18.0|        3|        70.0|      90.0|  2124|        13.5|   73| Japan|
|      Toyota Mark II|20.0|        6|       156.0|     122.0|  2807|        13.5|   73| Japan|
|        Toyota Camry|20.0|        4|        97.0|      88.0|  2279|        19.0|   73| Japan|
|         Datsun B210|31.0|        4|        79.0|      67.0|  1950|        19.0|   74| Japan|
|              Subaru|26.0|        4|       108.0|      93.0|  2391|        15.5|   74| Japan|
| Toyota Corolla 1200|32.0|        4|        71.0|      65.0|  1836|        21.0|   74| Japan|
|         Honda Civic|24.0|        4|       120.0|      97.0|  2489|        15.0|   74| Japan|
|      Toyota Corolla|31.0|        4|        76.0|      52.0|  1649|        16.5|   74| Japan|
|          Datsun 710|32.0|        4|        83.0|      61.0|  2003|        19.0|   74| Japan|
|      Toyota Corolla|29.0|        4|        97.0|      75.0|  2171|        16.0|   75| Japan|
```

8. ***Quit Program:***

```
Enter the number of Query from above...
7
Quitting Program...
Thanks for using this Program...

Process finished with exit code 0
```

XII. Wrapping Up:

In this project, we have created a spark application using **Spark Core** and **Spark SQL** with **Java**. Here, we have loaded the CSV file into **Data Frame** without using any external package. Also, The CSV format is the common file format which gets used as a source file in most cases.
If you want to test the examples above, you will find my Github code link:
[Read CSV file into DataFrame And Perform some Queries]