Exp. No. : 03

Date : / /2024

Multimodule Data Engineering Project

AIM :

To implement a multimodule data engineering project to work around with the csv files in java using maven build tool

PROCEDURE :

Creating parent module :

- Create the parent module by navigating to the where you want the project to be
- Run the following command to create the parent project

  $ mvn archetype:generate -DgroupId=lab.ost.csvproj -DartifactId=csv-project archetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
- Navigate into the parent module and delete the App.java and its associated test files.

Modifying the pom.xml :

- Open the *pom.xml* in the parent module directory
- Add a *<modules>* section to specify the child modules as :

```
<modules>
  <module>data-ingestion</module>
  <module>transformation</module>
  <module>data-storage</module>
  <module>data-analytics</module>
</modules>
```

Creating Child Modules :

- Use the below maven command to create a new module.

  $ mvn archetype:generate -DgroupId=lab.ost.csvproj -DartifactId=<module-name> -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
- Update the child module's pom.xml to include the parent reference as :

```
<parent>
  <groupId>lab.ost.csvproject</groupId>
  <artifactId>csv-project</artifactId>
  <version>1.0</version>
</parent>
```

Repeat the above steps for each child module(Data Ingestion,Data Transformation,Data Storage,Data analytics)

Implementing Functionality in Child Modules :

1. Data Ingestion Module :
   - Add functionality to read data from a CSV file using in-built file handling methods.
   - Write the parsed data to a temporary location and pass the data to the next module.
2. Data Transformation Module **:**
   - Implement filtering or transformation logic based on specific criteria.
   - Ensure the transformed data is prepared for storage or analysis and pass the data to the next module.
3. Data Storage Module :
   - Save the transformed data to a output file in the user desired path.
4. Data Analytics Module :

- Perform Data Analysis, such as generating summaries (i.e. Average Salary) from the transformed data

5. Main Module:
   - Create a main runner class in a new module to invoke the workflow.
   - Call the Data Ingestion, Data Transformation, Data storage and Data analytics modules in sequence based on the user input.

Compiling the Multimodule Project :

- Run the maven Command to compile all modules and ensure there are no errors and all modules are compiled successfully
  $ mvn clean install
- Execute the multimodule project from the main class using the below command.
  $ mvn exec:java -Dexec.mainClass=”lab.ost.csvproj.”
- Use the below maven command to package the project into a executable jar.
  $ java -cp target/csv-project-1.0.jar ost.lab.csvproj.

Program:

1. **Data Ingestion Module (DataIngestion.java):**
   Reads and parses a CSV file.

```
package com.example.dataengineering.ingestion;

import java.io.*;
import java.util.*;

public class DataIngestion {
    public List<String[]> readCSV(String filePath) {
        List<String[]> data = new ArrayList<>();
        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            String header = br.readLine();  // Skip header
            String line;
            while ((line = br.readLine()) != null) {
                data.add(line.split(","));
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return data;
    }
}
```

2. **Data Transformation Module (DataTransformation.java):**
   Filters rows based on criteria.

```
package com.example.dataengineering.transformation;

import java.util.*;

public class DataTransformation {
    public List<String[]> filterByCountry(List<String[]> data, String country) {
        return data.stream()
```

```
                .filter(row -> row[2].equalsIgnoreCase(country))
                .collect(Collectors.toList());
    }
}
```

3. Data Storage Module (DataStorage.java):
Writes processed data to a file.

```
package com.example.dataengineering.storage;

import java.io.*;
import java.util.*;

public class DataStorage {
    public void writeCSV(List<String[]> data, String outputPath) {
        try (FileWriter writer = new FileWriter(outputPath)) {
            for (String[] row : data) {
                writer.append(String.join(",", row)).append("\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

4. Data Analytics Module (DataAnalytics.java):
Computes analytical metrics.

```
package com.example.dataengineering.analytics;

import java.util.*;

public class DataAnalytics {
    public double calculateAverageSalary(List<String[]> data) {
        return data.stream()
                .mapToDouble(row -> Double.parseDouble(row[4]))
                .average()
                .orElse(0);
    }
}
```

5. Main Class (Main.java):
Integrates all modules.

```
package com.example;

import com.example.dataengineering.ingestion.*;
import com.example.dataengineering.transformation.*;
import com.example.dataengineering.storage.*;
```

```
import com.example.dataengineering.analytics.*;

import java.util.*;

public class Main {
   public static void main(String[] args) {
      String inputFile = "data/input.csv";
      String outputFile = "data/output.csv";

      DataIngestion ingestion = new DataIngestion();
      DataTransformation transformation = new DataTransformation();
      DataStorage storage = new DataStorage();
      DataAnalytics analytics = new DataAnalytics();

      List<String[]> data = ingestion.readCSV(inputFile);
      List<String[]> filteredData = transformation.filterByCountry(data, "USA");

      storage.writeCSV(filteredData, outputFile);

      double avgSalary = analytics.calculateAverageSalary(filteredData);
      System.out.println("Average Salary in the USA: $" + avgSalary);
   }
}
```

Preparation	
Observation	
Output	
Viva	
Record	
Total	

RESULT:

Hence the multimodule data engineering project in java using maven build tool has been implemented and the output has been verified successfully.

OUTPUT:

```
C:\Users\Deepika\MavenProjects\data-engineering-parent\data-ingestion>mvn exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] ----------< com.example.dataengineering:data-ingestion >----------
[INFO] Building data-ingestion 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- exec-maven-plugin:3.4.0:java (default-cli) @ data-ingestion ---
Average Salary in the USA: $58600.0
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.485 s
[INFO] Finished at: 2024-11-22T20:32:51+05:30
[INFO] ------------------------------------------------------------------------

 1,John Doe,USA,30,70000
 3,Bob Johnson,USA,45,500 00
 5,Tom White,USA,35,60000
 8,Sarah Silver,USA,29,55000
 9,Saran Silver,USA,21,58000
Average Salary in the USA: $58600.0
```