**Lab Exercise 6- Using Basic Accumulate Functions in Drools**

**Objective:**

To create a Drools project that demonstrates the use of basic accumulate functions such as sum, max, and min to perform aggregation operations on a set of data.

**Step 1: Set Up the Project**

* Ensure you have a Maven project set up with Drools dependencies. Use your existing project setup or create a new one.

**Step 2: Define Java Model Classes**

**Create a Product Class:**

In the src/main/java/com/example/model directory, create a Product.java file with the following content:

package com.example.model;

public class Product {

private String name;

private double price;

public Product(String name, double price) {

this.name = name;

this.price = price;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

@Override

public String toString() {

return "Product{name='" + name + "', price=" + price + "}";

}

}

**Step 3: Create DRL Rules with Accumulate Functions**

**Create the Rule File:**

In the src/main/resources directory, create a file named productRules.drl with the following content:

package com.example.rules

import com.example.model.Product

rule "Total Price of All Products"

when

$totalPrice : Double() from accumulate(

Product($price : price),

sum($price)

)

then

System.out.println("The total price of all products is $" + $totalPrice);

end

rule "Maximum Price of Products"

when

$maxPrice : Double() from accumulate(

Product($price : price),

max($price)

)

then

System.out.println("The maximum price of any product is $" + $maxPrice);

end

rule "Minimum Price of Products"

when

$minPrice : Double() from accumulate(

Product($price : price),

min($price)

)

then

System.out.println("The minimum price of any product is $" + $minPrice);

end

rule "Average Price of Products"

when

$avgPrice : Double() from accumulate(

Product($price : price),

average($price)

)

then

System.out.println("The average price of all products is $" + $avgPrice);

end

**Explanation of the Rules:**

* Total Price of All Products: Uses the sum accumulate function to calculate the total price of all products.
* Maximum Price of Products: Uses the max accumulate function to find the highest price among the products.
* Minimum Price of Products: Uses the min accumulate function to find the lowest price among the products.
* Average Price of Products: Uses the average accumulate function to calculate the average price of all products.

**Step 4: Update the Main Application**

**Modify the DroolsTest Class:**

Update the DroolsTest.java file to include the new Product class:

package com.example;

import com.example.model.Product;

import org.kie.api.KieServices;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

public class DroolsTest {

public static void main(String[] args) {

// Load the knowledge base

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

KieSession kSession = kContainer.newKieSession("ksession-rules");

// Create sample products

Product product1 = new Product("Laptop", 1500);

Product product2 = new Product("Tablet", 300);

Product product3 = new Product("Phone", 800);

// Insert the products into the session

kSession.insert(product1);

kSession.insert(product2);

kSession.insert(product3);

// Fire all rules

kSession.fireAllRules();

// Dispose the session

kSession.dispose();

}

}

**Step 5: Run the Application**

**Compile and Run:**

Compile and run the DroolsTest class.

**Expected Output:**

The console should display output similar to the following:

The total price of all products is $2600.0

The maximum price of any product is $1500.0

The minimum price of any product is $300.0

The average price of all products is $866.6666666666666