Scala Discounts Rule Engine

Omar Ashraf

Scala

Problem Statement:

A rule engine is needed to evaluate order transactions for discounts based on various conditions such as product expiry, product type, special events, and quantity purchased in a large retail store.

Qualifying Rules and Calculation Rules:

1. **Less than 30 days remaining for product expiry:**
    - Discounts based on remaining days:
        - 29 days: 1% discount
        - 28 days: 2% discount
        - ...and so on.
2. **Cheese and Wine products on sale:**
    - Cheese: 10% discount
    - Wine: 5% discount
3. **Products sold on March 23rd:**
    - Special discount: 50%
4. **More than 5 units of the same product:**
    - 6-9 units: 5% discount
    - 10-14 units: 7% discount
    - More than 15 units: 10% discount
5. **App Usage:**
    - quantity: 1, 2, 3, 4, 5 -> discount 5%
    - quantity 6, 7, 8, 9, 10 -> discount 10%
    - quantity 11, 12, 13, 14, 15 -> discount 15%
    - ...and so on.
6. **Visa Card Usage:**
    - 5% discount

Technical Approach:

1. **Functional Programming:** Core logic implemented using pure functions with an emphasis on immutability and predictability.
2. **Data Processing:**
    - Read order transactions from a CSV file.
    - Apply qualifying rules to determine discounts.
    - Calculate final prices after applying discounts.

3. **Database Interaction:**
    - Insert processed data into a database table.
    - Utilize JDBC for database connectivity.
4. **Logging:**
    - Log engine events in a file with timestamps and log levels.

Implemented Discount Rules:

1. **Day Remaining Qualifying Rule:**
    - Function: LessThan30Days
    - Description: Checks if the remaining days for a product to expire is less than 30.
2. **On-Sale Products Qualifying Rule:**
    - Function: CheeseAndWine
    - Description: Identifies whether a product is eligible for discount based on being a wine or cheese product.
3. **Special Discount for Products Sold on 23rd of March:**
    - Function: ProductSoldOn23March
    - Description: Applies a special discount if a product is sold on the 23rd of March.
4. **Quantity of Products Sold:**
    - Function: MoreThan5Products
    - Description: Applies a special discount if a product is sold on the 23rd of March.
5. **Through App Rule:**
    - Function: ThroughApp
    - Description: Applies discount based on the quantity.
6. **Using Visa:**
    - Function: UsingVisa
    - Description: Applies discount based on visa payment method.

Database Interaction:

- Utilizes Oracle JDBC driver for database connectivity.
- Inserts processed data into the orders table, including detail such as order date, expiry date, product category, quantity, unit price, channel, payment method, discount, and total price.

Logging Mechanisms:

- Function: log_event
- Description: Writes information about engine rule interactions into the logs.txt file.
- Structure: Logs timestamp, log level, and message for each rule interaction event.

Usage:

1. Data Input:

- Ensure that Scala and necessary dependencies are installed.
- Place the order data CSV file (TRX1000.csv) in the src/main/resources directory.
- Update database connection details in the main object.
- Run the main object to execute the application.
- Check the logs for information about the execution process.

2. Configuration:

- Update the database connection details (URL, username, password) in the application.
- Create table as follows:

```
CREATE TABLE Orders (
  Order_Date timestamp,
  Product_Name VARCHAR(255),
  Expiry_Date DATE,
  Quantity INT,
  Unit_Price DECIMAL(10,2),
  Channel VARCHAR(50),
  Payment_Method VARCHAR(50),
  Discount DECIMAL(4,3),
  Final_Price DECIMAL(10,2)
);
```

3. Execution:

- Run the application to process order transactions, calculate discounts, and insert data into the database.

4. Verification:

- Check the orders table in the database for inserted records with calculated discounts and total prices.
- Review the logs.txt file for logged engine rule interactions and any error messages.

```
Timestamp: 2024-05-08T17:10:44.330Z     LogLevel: Info          message: Starting Application
Timestamp: 2024-05-08T17:10:44.352Z     LogLevel: Event         message: Opening TRX1000.csv
Timestamp: 2024-05-08T17:10:44.475Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.488Z     LogLevel: Info          message: Customer qualified for discount and qualified for 2 rules
Timestamp: 2024-05-08T17:10:44.489Z     LogLevel: Info          message: Customer discount: 0.05
Timestamp: 2024-05-08T17:10:44.489Z     LogLevel: Debug         message: Time taken to calculate discount: 12.3723 milliseconds
Timestamp: 2024-05-08T17:10:44.489Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.489Z     LogLevel: Info          message: Didn't qualify for discount
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Debug         message: Time taken to calculate discount: 0.1625 milliseconds
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Info          message: Customer qualified for discount and qualified for 2 rules
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Info          message: Customer discount: 0.05
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Debug         message: Time taken to calculate discount: 0.2734 milliseconds
Timestamp: 2024-05-08T17:10:44.494Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Info          message: Customer qualified for discount and qualified for 2 rules
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Info          message: Customer discount: 0.11
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Debug         message: Time taken to calculate discount: 0.230901 milliseconds
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Info          message: Customer qualified for discount and qualified for 1 rules
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Info          message: Customer discount: 0.07
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Debug         message: Time taken to calculate discount: 0.190501 milliseconds
Timestamp: 2024-05-08T17:10:44.495Z     LogLevel: Debug         message: Start processing
Timestamp: 2024-05-08T17:10:44.496Z     LogLevel: Info          message: Customer qualified for discount and qualified for 2 rules
Timestamp: 2024-05-08T17:10:44.496Z     LogLevel: Info          message: Customer discount: 0.05
Timestamp: 2024-05-08T17:10:44.496Z     LogLevel: Debug         message: Time taken to calculate discount: 0.1983 milliseconds
Timestamp: 2024-05-08T17:10:44.496Z     LogLevel: Debug         message: Start processing
```

```
Timestamp: 2024-05-13T06:57:17.871Z     LogLevel: Debug         message: Time taken to calculate discount: 0.0874 milliseconds
Timestamp: 2024-05-13T06:57:17.871Z     LogLevel: Debug         message: Writing data to csv
Timestamp: 2024-05-13T06:57:17.920Z     LogLevel: Debug         message: Data wrote to csv
Timestamp: 2024-05-13T06:57:17.920Z     LogLevel: Debug         message: Inserting Data In DB
Timestamp: 2024-05-13T06:57:24.410Z     LogLevel: Debug         message: Data Inserted In DB
Timestamp: 2024-05-13T06:57:24.413Z     LogLevel: Debug         message: Connection Closed with DB
Timestamp: 2024-05-13T06:57:24.413Z     LogLevel: Debug         message: Time taken to insert in DB: 6492.05 millisecond
Timestamp: 2024-05-13T06:57:24.413Z     LogLevel: Debug         message: Time taken for the app: 7806.3694 millisecond
```

5. Dependencies:

- Java JDBC for database connection.