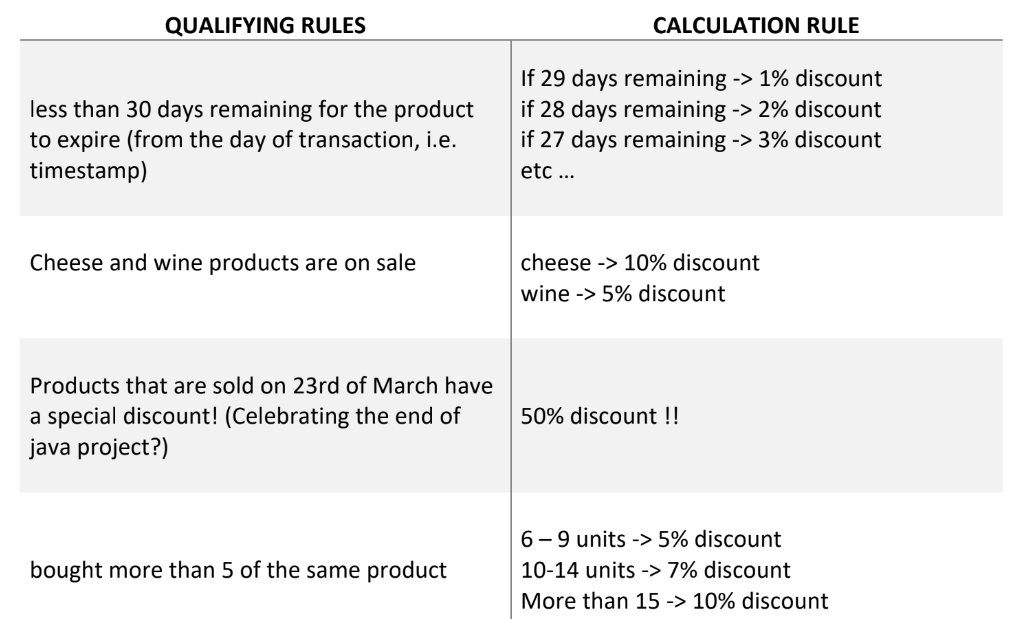
**Rule Engine with Scala**

**The Problem Statement**

A huge retail store wants a rule engine that qualifies orders’ transactions to discounts based

on a set of qualifying rules. And automatically calculates the proper discount based on some

calculation rules as follows:



I read this data using pandas to do simple exploratory data analysis (EDA) on it, to see the data distributions and unique values for each column and so on.

And I retrieve the wine and cheese products string pattern to recognize them using regex in my code logic in Scala.

**On-Sale Products**

**If the product in this wine list products, it will have a 5% discount**

* Wine - White Pinot Grigio'
* 'Wine - Chardonnay Mondavi'
* 'Wine - Magnotta - Belpaese'
* 'Wine - Prosecco Valdobienne'
* 'Wine - Marlbourough Sauv Blanc'
* 'Wine - Saint Emilion Calvet'
* 'Wine - Vidal Icewine Magnotta'
* 'Wine - Bouchard La Vignee Pinot'
* 'Wine - Cousino Macul Antiguas'
* 'Wine - White Schroder And Schyl'
* 'Wine - Red Colio Cabernet'
* 'Wine - White Lindemans Bin 95'
* 'Wine - White Magnotta'
* 'Wine - Casillero Deldiablo'
* 'Wine - Rubyport'
* 'Wine - Soave Folonari'
* 'Wine - Blue Nun Qualitatswein'
* 'Wine - Red Gallo Merlot'
* 'Wine - Vineland Estate Semi - Dry'
* 'Wine - Fume Blanc Fetzer'
* 'Wine - Charddonnay Errazuriz'
* 'Wine - Rosso Toscano Igt'
* 'Wine - Barossa Valley Estate'
* 'Wine - Vovray Sec Domaine Huet'
* 'Wine - Carmenere Casillero Del'
* 'Wine - Placido Pinot Grigo'
* 'Wine - Duboeuf Beaujolais'
* 'Wine - Taylors Reserve'
* 'Wine - Segura Viudas Aria Brut'
* 'Wine - Bourgogne 2002 La'
* 'Wine - Rosso Del Veronese Igt'
* 'Wine - Magnotta - Cab Franc'
* 'Wine - Tribal Sauvignon'
* 'Wine - Shiraz South Eastern'
* 'Wine - Pinot Noir Stoneleigh'
* 'Wine - German Riesling'
* 'Wine - Domaine Boyar Royal'
* 'Wine - Pinot Noir Latour'

**And if it in the following cheese products list, it will have a 10% discount:**

* 'Cheese Cloth No 60'
* 'Cheese - Taleggio D.o.p.'
* 'Cheese - La Sauvagine'
* 'Cheese - Brick With Pepper'
* 'Cheese - Cheddar With Claret'
* 'Cheese - Camembert'
* 'Cheese - Gouda'
* 'Cheese - Goat'
* 'Cheese - Havarti Salsa'
* 'Cheese - Woolwich Goat Log'
* 'Cheese - Briedanish'
* 'Cheese - Parmesan Grated'

**The functions that I created:**

**Day Remaining Qualifying Rule**

days\_remaining\_qualifier – it returns the number of remaining days of a product to be expired

days\_remaining\_discount\_calculator – it returns the discount as double value

days\_remaining\_discount\_percentage – it returns the discount as percentage

**On-Sale Products Qualifying Rule**

onSaleProducts\_qualifier – it checks if the product is a wine or cheese or otherwise and map the required discount amount on each one

**Special Discount for Products Sold in 23th Of March**

marchSpecialDiscount – it checks if the sold date is in 23th of March, and if it is apply on it the proper discount value

**Quantity of Products Sold**

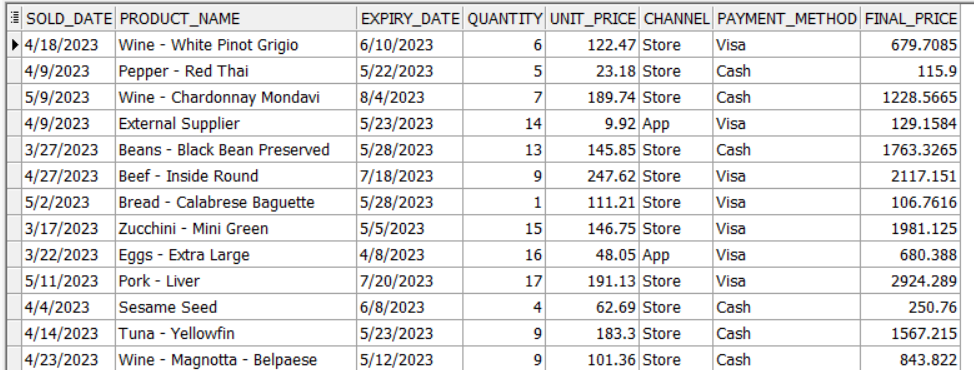
quantitySoldDiscount – if chekcs the number of quantities sold of each product and calculate the discount according to the quantity range it falls into as mentioned in the problem statement.

**Database Connection**

I calculated the final discount by taking the average of the two maximum discount values of each product, then I created a table RetailSore that holds the following columns:

* SOLD\_DATE
* PRODUCT\_NAME
* EXPIRY\_DATE
* QUANTITY
* UNIT\_PRICE
* CHANNEL
* PAYMENT\_METHOD
* FINAL\_PRICE

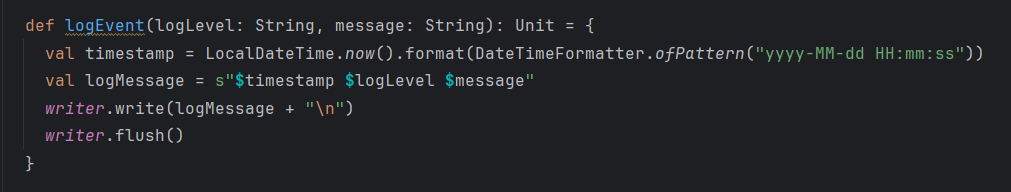
The connection that I used is Oracle connection using JDBC oracle drive, then I insert into this table the values of source file csv file with the FINAL\_PRICE column that was calculated through the project code.



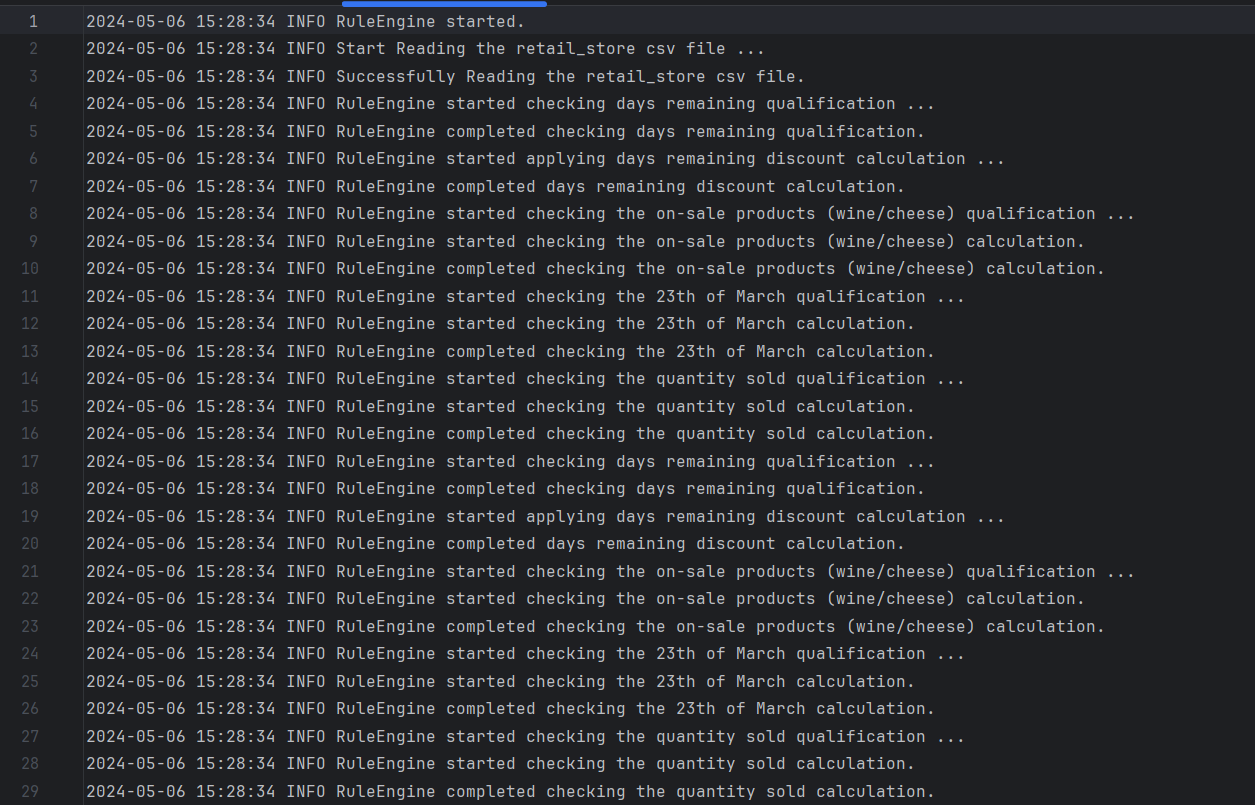
**Logging the Engine Rule Interactions**

Finally, I created a function that write into ‘rules\_engine.log’ file. It writes the INFO operations happened in the Engine Rule.

**Here is the function:**



**Here is the ‘rules\_engine.log’ file structure:**



**Logging of data insertion into the database table**

