Functional plan:

Q1:

Input list of orders

Output one attribute from each part of the list

Output the number of different values in the list

Q2:

Input list of orders

Output all unique products (finding unique products in a list)

Output total amount of those products

Q3:

Input list of orders

Output the customer and product from the list IF:

Customer has ordered more than a set amount

(conditional statements in haskell)

Q4:

Input list of orders

Input list of deliveries

IF:

Ordered product > delivered product

THEN:

Output list of product

Output difference in quantity

Q5:

Inputting

Can input a list by inputting a set of tuples

Tuples = Lecture 3

Eg Orders [ (“Customer1”, “Product1”, 5), (“Customer4”, “Product3”, 89) ]

Can use tuples for comparison:

Comparing ordered vs delivered in Q4

Can do

Compare (1,True,'c') (1,True,'d') = LESS THAN

Can use “fst” and “snd” to get first and second values in orders:

Fst (1, 5) = 1

Snd (1, 5) = 5

Format for inputting values:

numProducts1 [(Order "Cus1" "prod1" 59)]

numProducts1 [(Order "Cus1" "prod1" 59), (Order "Cus2" "prod3" 23)]

[ (Order "cust1" "charger" 1.0), (Order "cust2" "phone" 4.0), (Order "cust4" "keyboard" 1.0), (Order "cust5" "tv" 14.0), (Order "cust2" "charger" 1.0), (Order "cust1" "phone" 6.0) ]

[ (Order "cust1" "charger" 1.0),

(Order "cust2" "phone" 4.0),

(Order "cust4" "keyboard" 1.0),

(Order "cust5" "tv" 14.0),

(Order "cust2" "charger" 1.0),

(Order "cust1" "phone" 6.0) ]

[ (Delivery "charger" 1.0), (Delivery "phone" 2.0), (Delivery "keyboard" 1.0), (Delivery "tv" 14.0), (Delivery "wallet" 3.0), (Delivery "phone" 1.0) ]

[ (Delivery "charger" 1.0),

(Delivery "phone" 2.0),

(Delivery "keyboard" 1.0),

(Delivery "tv" 14.0),

(Delivery "wallet" 3.0),

(Delivery "phone" 1.0) ]

Shit to do:

Test

A screenshot of a computer

Description automatically generated with medium confidence