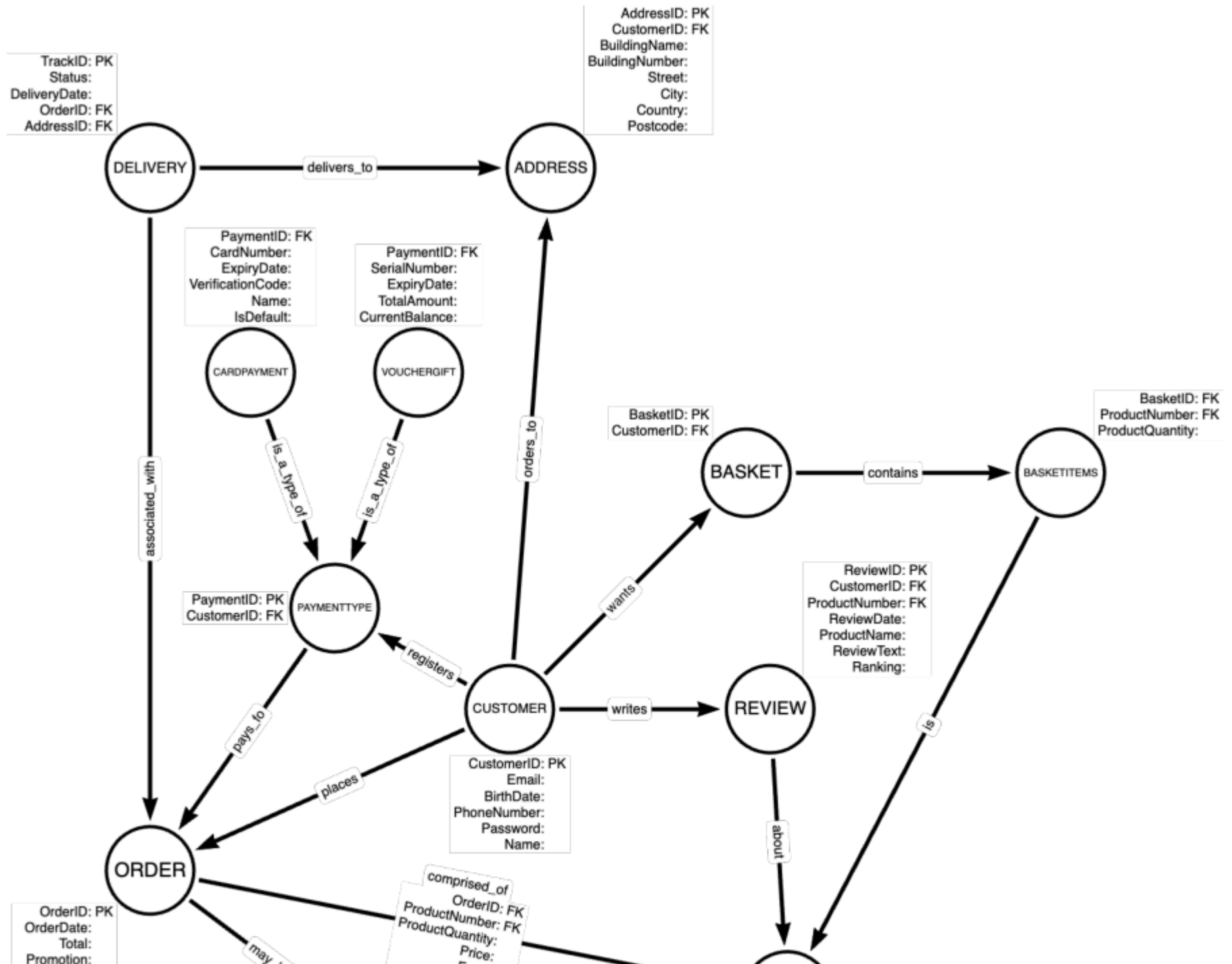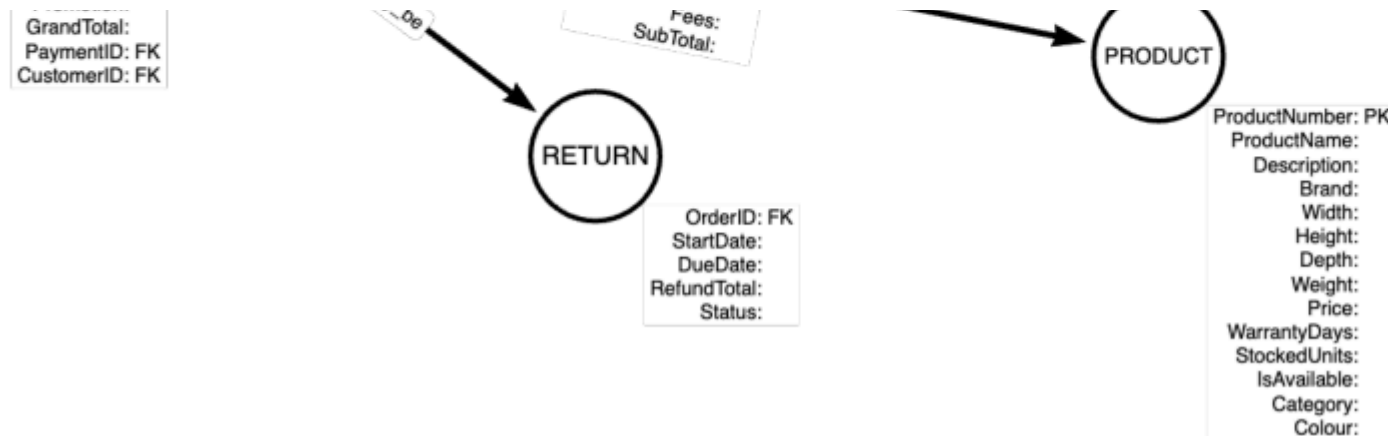# Question 2 Neo4j

2a) *Graph Database*

I have imported an image of my Property Graph Model and displayed it below. To ensure that it's accessible by anyone, I have put the image in the same folder and used a relative filepath to access it. I created the Property Graph Model using Arrows.app and adapting it from my ER Model.

```python
import matplotlib.pyplot as plt

# displaying the image
img = plt.imread("data/PropertyGraphModel.png")
plt.figure(figsize=(12, 12))
plt.imshow(img)
plt.axis('off')
plt.show()
```

DELIVERY
- TrackID: PK
- Status:
- DeliveryDate:
- OrderID: FK
- AddressID: FK

ADDRESS
- AddressID: PK
- CustomerID: FK
- BuildingName:
- BuildingNumber:
- Street:
- City:
- Country:
- Postcode:

CARDPAYMENT
- PaymentID: FK
- CardNumber:
- ExpiryDate:
- VerificationCode:
- Name:
- IsDefault:

VOUCHERGIFT
- PaymentID: FK
- SerialNumber:
- ExpiryDate:
- TotalAmount:
- CurrentBalance:

PAYMENTTYPE
- PaymentID: PK
- CustomerID: FK

BASKET
- BasketID: PK
- CustomerID: FK

BASKETITEMS
- BasketID: FK
- ProductNumber: FK
- ProductQuantity:

REVIEW
- ReviewID: PK
- CustomerID: FK
- ProductNumber: FK
- ReviewDate:
- ProductName:
- ReviewText:
- Ranking:

CUSTOMER
- CustomerID: PK
- Email:
- BirthDate:
- PhoneNumber:
- Password:
- Name:

ORDER
- OrderID: PK
- OrderDate:
- Total:
- Promotion:

comprised_of
- OrderID: FK
- ProductNumber: FK
- ProductQuantity:
- Price:

Relationships:
- DELIVERY delivers_to ADDRESS
- DELIVERY associated_with ORDER
- CARDPAYMENT is_a_type_of PAYMENTTYPE
- VOUCHERGIFT is_a_type_of PAYMENTTYPE
- CUSTOMER orders_to ADDRESS
- CUSTOMER registers PAYMENTTYPE
- CUSTOMER wants BASKET
- CUSTOMER writes REVIEW
- CUSTOMER places ORDER
- PAYMENTTYPE pays_to ORDER
- BASKET contains BASKETITEMS
- BASKETITEMS is ...
- REVIEW about ...
- ORDER may ... comprised_of

GrandTotal:
PaymentID: FK
CustomerID: FK

...De

Fees:
SubTotal:

PRODUCT

RETURN

OrderID: FK
StartDate:
DueDate:
RefundTotal:
Status:

ProductNumber: PK
ProductName:
Description:
Brand:
Width:
Height:
Depth:
Weight:
Price:
WarrantyDays:
StockedUnits:
IsAvailable:
Category:
Colour:

For setting up the Neo4j database and executing queries, I used the code we learned in Seminar 9 as a guideline. First, I imported the libraries required for this question and then to set up the connection, I created a cluster on AuraDB and noted the URI, user, and password which I then used in the Graph Database Driver. Finally, I copied the function used in the seminar which allows me to be more efficient with my code by only needing to access the function to run the query.

In [7]:
```python
# importing necessary libraries
from neo4j import GraphDatabase
from neo4j.exceptions import ServiceUnavailable
import logging

# creating connection
URI = "neo4j+s://5e29471e.databases.neo4j.io"
user = "neo4j"
password = "2QiW7i2LiStEqozOK2upyqa5nzXdjSp-oW1lAMNPJRw"
AUTH = (user, password)

driver = GraphDatabase.driver(URI, auth=AUTH)
driver.verify_connectivity()

# Function to run a single query
def run_query(query, params=None):
    with driver.session() as session:
        return session.run(query, params)
```

2b) _Populating Data_

Creating Queries to Load Data

To populate the data, I used data that I had saved on Google Sheets. The data is from the previous project and is a mix of random generation and AI prompts on Mockaroo.com as well as some manual data generation and checking to ensure that foreign keys match up and data/relationships stay consistent. To ensure that the queries properly access the data, the Sheets are publically accessible by link and converted to CSV format.

I imported the data using _LOAD CSV WITH HEADERS_ command to read the data and _MERGE_ to match it. For each node, I created a seperate query and checked the headers to make sure the names of the node properties were accurate and used _SET_ . I also kept the datatypes as accurate as I could by using functions like _toFloart()_ and _toInteger()_ Then I created relationships between nodes using _MERGE_ again.

Finally, I used the function _run_query_ to run all of the queries.

Considerations:

While I was importing the data, I initially intended to import first and then create the relationships in a different step but I realised that I would have to load the data again and it was more efficient to just create the relationships alongside importing the data.

Another issue I came across was an error saying "CypherSyntaxError: {code: Neo.ClientError.Statement.SyntaxError} {message: WITH is required between MERGE and MATCH (line 7, column 1 (offset: 306))}". To resolve this, I had to google it and got this link: https://stackoverflow.com/questions/32156721/cypher-with-is-required-between-create-and-match. Which told me that I can't use _MERGE_ and _MATCH_ clauses directly without seperating them with _WITH_ .

In [8]:
```
load_customer = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
CREATE (:Customer {
    CustomerID: toInteger(row.CustomerID),
    Email: row.Email,
    BirthDate: row.BirthDate,
    PhoneNumber: row.PhoneNumber,
    Password: row.Password,
    Name: row.Name
```

```
});
'''

load_address = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (a:Address {AddressID: toInteger(row.AddressID)})
WITH a, row
MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})
MERGE (c)-[:HAS_ADDRESS]->(a)
SET a.BuildingName = row.BuildingName,
    a.BuildingNumber = row.BuildingNumber,
    a.Street = row.Street,
    a.City = row.City,
    a.Country = row.Country,
    a.Postcode = row.Postcode
'''

load_billingaddress = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (ba:BillingAddress {AddressID: toInteger(row.AddressID)})
WITH ba, row
MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})
MERGE (c)-[:HAS_BILLING_ADDRESS]->(ba)
'''

load_shippingaddress = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (sa:ShippingAddress {AddressID: toInteger(row.AddressID)})
WITH sa, row
MATCH (a:Address {AddressID: toInteger(row.AddressID)})
MERGE (sa)-[:SHIPPED_TO]->(a)
SET sa.DeliveryInstructions = row.DeliveryInstructions
'''

load_paymenttype = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (pt:PaymentType {PaymentID: toInteger(row.PaymentID)})
MERGE (c:Customer {CustomerID: toInteger(row.CustomerID)})
MERGE (c)-[:HAS_PAYMENT]->(pt)
'''

load_cardpayment = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (cp:CardPayment {PaymentID: toInteger(row.PaymentID)})
```

```
WITH cp, row
MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})
MERGE (c)-[:MADE_CARD_PAYMENT]->(cp)
WITH cp, row
MATCH (p:PaymentType {PaymentID: toInteger(row.PaymentID)})
MERGE (cp)-[:IS_A]->(p)
SET cp.CardNumber = row.CardNumber,
    cp.ExpiryDate = row.ExpiryDate,
    cp.VerificationCode = row.VerificationCode,
    cp.Name = row.Name,
    cp.IsDefault = CASE WHEN row.IsDefault = 'true' THEN true ELSE false END
'''

load_vouchergift = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (vg:VoucherGift {PaymentID: toInteger(row.PaymentID)})
SET vg.SerialNumber = row.SerialNumber,
    vg.ExpiryDate = row.ExpiryDate,
    vg.TotalAmount = toFloat(row.TotalAmount),
    vg.CurrentBalance = toFloat(row.CurrentBalance)
WITH vg, row
MATCH (pt:PaymentType {PaymentID: toInteger(row.PaymentID)})
MERGE (pt)-[:IS_A]->(vg)
'''

load_product = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (p:Product {ProductNumber: toInteger(row.ProductNumber)})
SET p.ProductName = row.ProductName,
    p.Brand = row.Brand,
    p.Width = toFloat(row.Width),
    p.Height = toFloat(row.Height),
    p.Depth = toFloat(row.Depth),
    p.Weight = toFloat(row.Weight),
    p.Price = toFloat(row.Price),
    p.WarrantyDays = toInteger(row.WarrantyDays),
    p.StockedUnits = toInteger(row.StockedUnits),
    p.IsAvailable = (row.IsAvailable = 'TRUE'),
    p.Description = row.Description
'''

load_productrelations = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MATCH (p:Product {ProductNumber: toInteger(row.ProductNumber)}), (c:Category {CategoryID: toInteger(row.CategoryID)})
```

```
    MERGE (pc:ProductCategory {ProductNumber: toInteger(row.ProductNumber), CategoryID: toInteger(row.CategoryID)})
    MERGE (p)-[:BELONGS_TO]->(c)
    MERGE (p)-[:HAS_CATEGORY]->(pc)
    '''

    load_productcategory = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MATCH (p:Product {ProductNumber: toInteger(row.ProductNumber)}), (c:Category {CategoryID: toInteger(row.CategoryID)})
    MERGE (pc:ProductCategory {ProductNumber: toInteger(row.ProductNumber), CategoryID: toInteger(row.CategoryID)})
    MERGE (p)-[:BELONGS_TO]->(c)
    MERGE (p)-[:HAS_CATEGORY]->(pc)
    '''

    load_category = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MERGE (c:Category {CategoryID: toInteger(row.CategoryID), CategoryName: row.CategoryName})
    '''

    load_productcolour = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MATCH (p:Product {ProductNumber: toInteger(row.ProductNumber)}), (c:Colour {ColourID: toInteger(row.ColourID)})
    MERGE (pc:ProductColour {ProductNumber: toInteger(row.ProductNumber), ColourID: toInteger(row.ColourID)})
    MERGE (p)-[:HAS_COLOUR]->(c)
    MERGE (p)-[:HAS_COLOUR]->(pc)
    '''

    load_colour = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MERGE (c:Colour {ColourID: toInteger(row.ColourID), ColourName: row.ColourName})
    '''

    load_basket = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MERGE (b:Basket {BasketID: toInteger(row.BasketID), CustomerID: toInteger(row.CustomerID)})
    WITH row, b
    MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})
    MERGE (c)-[:HAS_BASKET]->(b)
    '''

    load_basketitems = '''
    LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
    MERGE (bi:BasketItem {BasketID: toInteger(row.BasketID), ProductNumber: toInteger(row.ProductNumber)})
    WITH row, bi
    MERGE (b:Basket {BasketID: toInteger(row.BasketID)})
```

```
MERGE (p:Product {ProductNumber: toInteger(row.ProductNumber)})
MERGE (b)-[:CONTAINS_ITEM]->(bi)
MERGE (bi)-[:IS_OF_PRODUCT]->(p)
SET bi.ProductQuantity = toInteger(row.ProductQuantity)
'''

load_order = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (o:Order {OrderID: toInteger(row.OrderID)})
SET o.OrderDate = row.OrderDate,
    o.Promotion = toFloat(row.Promotion),
    o.Total = toFloat(row.Total),
    o.GrandTotal = toFloat(row.GrandTotal)
WITH o, row
MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})
MERGE (c)-[:PLACED_ORDER]->(o)
'''

load_orderproduct = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (op:OrderProduct {OrderID: toInteger(row.OrderID), ProductNumber: toInteger(row.ProductNumber)})
WITH row, op
MATCH (o:Order {OrderID: toInteger(row.OrderID)}), (p:Product {ProductNumber: toInteger(row.ProductNumber)})
MERGE (o)-[:CONTAINS_PRODUCT]->(p)
MERGE (op)-[:IS_PRODUCT_IN_ORDER]->(p)
SET op.ProductQuantity = toInteger(row.ProductQuantity),
    op.Price = toFloat(row.Price),
    op.Fees = toFloat(row.Fees),
    op.Subtotal = toFloat(row.Subtotal)
'''


load_delivery = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (d:Delivery {TrackID: row.TrackID, OrderID: toInteger(row.OrderID)})
WITH row, d
MATCH (a:Address {AddressID: toInteger(row.AddressID)})  // Ensure to match the Address node
MERGE (d)-[:DELIVERED_TO]->(a)
SET d.Status = row.Status,
    d.DeliveryDate = row.DeliveryDate
'''

load_review = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
```

```
MERGE (r:Review {ReviewID: toInteger(row.ReviewID)})
WITH row, r
MATCH (c:Customer {CustomerID: toInteger(row.CustomerID)})  // Match Customer by CustomerID
MATCH (p:Product {ProductNumber: toInteger(row.ProductNumber)})  // Match Product by ProductNumber
MERGE (r)-[:WROTE]->(c)
MERGE (r)-[:REVIEWS]->(p)
SET r.ReviewDate = row.ReviewDate,
    r.ReviewText = row.ReviewText,
    r.Ranking = toInteger(row.Ranking),
    r.ProductName = row.ProductName
'''


load_return = '''
LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
MERGE (r:Return {OrderID: toInteger(row.OrderID)})
WITH row, r
MATCH (o:Order {OrderID: toInteger(row.OrderID)})
MERGE (r)-[:FOR_ORDER]->(o)
SET r.StartDate = row.StartDate,
    r.DueDate = row.DueDate,
    r.RefundTotal = toFloat(row.RefundTotal),
    r.Status = row.Status
'''
```

Running the import queries

```python
In [9]:   with driver.session(database="neo4j") as session:
              session.run(load_customer)
              session.run(load_address)
              session.run(load_billingaddress)
              session.run(load_shippingaddress)
              session.run(load_paymenttype)
              session.run(load_cardpayment)
              session.run(load_vouchergift)
              session.run(load_product)
              session.run(load_productrelations)
              session.run(load_productcategory)
              session.run(load_category)
              session.run(load_productcolour)
              session.run(load_colour)
              session.run(load_basket)
              session.run(load_basketitems)
```

```
        session.run(load_order)
        session.run(load_orderproduct)
        session.run(load_delivery)
        session.run(load_review)
        session.run(load_return)
```

2c) _Customers and Orders_

I first checked that the data was loaded correctly and that the relationships between the Order, Product, and OrderProduct nodes were consistent and matched properly. The query and output verify this.

Next, I had to write code to compute the Grand Total for each order as this is a derived value. I do this by matching order to product and getting orderproduct so I can access all the attributes. First, I have a calculated price which is simply the unit price of each product in the order multiplied by it's product quantity. Next, to get the subtotal, I subtract the fees associated with each product order. Then to get the total, I add all the subtotals of the individual products in an order. Finally, I account for promotion which I had in the previous assignment defined to be a certain percentage off an entire order after fees so to calculate the grand total of an order, I multiple total by (1 - promotion). Then, to ensure that this calculation was done correctly, I used _RETURN_ with _LIMIT_ set to 5 to check the output.

Then I used a query by first matching the relevant nodes and adding the _COLLECT_ function which aggregates the list of products and their quantities. I also ordered by OrderID to make the output logical. For displaying the result, in order to make it both visually pleasing and easy to read, I made use of Python functions like _join_ wiht comma and printing a divider line.

In [12]:
```python
# query to check data is matched correctly
checker_order_product = '''
MATCH (o:Order)-[:CONTAINS_PRODUCT]->(p:Product)
MATCH (op:OrderProduct {OrderID: o.OrderID, ProductNumber: p.ProductNumber})
RETURN o.OrderID AS OrderID,
       p.ProductNumber AS ProductNumber,
       op.ProductQuantity AS ProductQuantity,
       op.Fees AS Fees,
       op.Price AS Price
LIMIT 5
'''

# running query
with driver.session(database="neo4j") as session:
    result_op = session.run(checker_order_product)
```

```
        for record in result_op:
            print(record)
```

```
<Record OrderID=30 ProductNumber=1 ProductQuantity=8 Fees=3.02 Price=None>
<Record OrderID=36 ProductNumber=1 ProductQuantity=10 Fees=8.18 Price=None>
<Record OrderID=209 ProductNumber=1 ProductQuantity=8 Fees=7.22 Price=None>
<Record OrderID=7 ProductNumber=2 ProductQuantity=4 Fees=7.06 Price=None>
<Record OrderID=13 ProductNumber=2 ProductQuantity=6 Fees=9.94 Price=None>
```

In [13]:
```python
# query to calculate grand totals of orders
calculate = '''
// calculating individual product prices
MATCH (o:Order)-[:CONTAINS_PRODUCT]->(p:Product)
MATCH (op:OrderProduct {OrderID: o.OrderID, ProductNumber: p.ProductNumber})
SET op.CalculatedPrice = p.Price * op.ProductQuantity

// calculating subtotals
SET op.Subtotal = op.CalculatedPrice - op.Fees

// calculating totals
WITH o, SUM(op.Subtotal) AS OrderTotal
SET o.Total = OrderTotal

// calculating grandtotal with promotion
SET o.GrandTotal = o.Total * (1 - o.Promotion)

RETURN o.OrderID AS OrderID, o.Total AS Total, o.GrandTotal AS GrandTotal
LIMIT 5
'''

# Execute the query
with driver.session(database="neo4j") as session:
    result_op = session.run(calculate)
    for record in result_op:
        print(record)
```

```
<Record OrderID=30 Total=13514.919999999998 GrandTotal=2567.8347999999987>
<Record OrderID=36 Total=11161.199999999999 GrandTotal=4464.48>
<Record OrderID=209 Total=5406.38 GrandTotal=594.7017999999999>
<Record OrderID=7 Total=5360.8099999999995 GrandTotal=696.9053>
<Record OrderID=13 Total=8421.14 GrandTotal=3284.2446>
```

In [14]:
```python
# query to get list of product and quantities in order
list_products_query = '''
```

```
MATCH (c:Customer)-[:PLACED_ORDER]->(o:Order)-[:CONTAINS_PRODUCT]->(p:Product)
MATCH (op:OrderProduct {OrderID: o.OrderID, ProductNumber: p.ProductNumber})
WITH c.Name AS CustomerName,
     o.OrderID AS OrderID,
     COLLECT(p.ProductName) AS ProductsInOrder,
     COLLECT(op.ProductQuantity) AS Quantities,
     o.GrandTotal AS GrandTotal
RETURN CustomerName, OrderID, ProductsInOrder, Quantities, GrandTotal
ORDER BY OrderID
'''

# running the query
with driver.session(database="neo4j") as session:
    result = session.run(list_products_query)
    for record in result:
        customer_name = record['CustomerName']
        products = ', '.join(record['ProductsInOrder'])
        quantities = ', '.join(map(str, record['Quantities']))
        print(f"\nCustomer Name: {customer_name} \nOrderID: {record['OrderID']} \nProducts: {products}\n"
        f"Respective Quantities: {quantities}\nGrandTotal: {record['GrandTotal']}")
        print('-' * 35) # for aesthetic purposes
```

Customer Name: Calhoun Dewett
OrderID: 1
Products: mixer, scale, ring, keychain, laundry basket
Respective Quantities: 2, 6, 9, 8, 8
GrandTotal: 2589.4487

Customer Name: Sheilah Hanhard
OrderID: 2
Products: blender, air fryer, scarf, shower caddy, storage bin
Respective Quantities: 6, 10, 8, 6, 7
GrandTotal: 11219.817000000001

Customer Name: Kahlil Ismay
OrderID: 3
Products: hair dryer, bath mat, saucepan, lotion, necklace
Respective Quantities: 10, 7, 4, 5, 1
GrandTotal: 12315.7699

Customer Name: Chase McIlmorow
OrderID: 4
Products: candle holder, shoe rack, umbrella, makeup bag, laptop case
Respective Quantities: 2, 7, 6, 9, 4
GrandTotal: 10848.801599999999

Customer Name: Ritchie Morgon
OrderID: 5
Products: mixer, ring, lotion, laptop case, socks
Respective Quantities: 4, 8, 9, 2, 1
GrandTotal: 5570.950400000001

Customer Name: Lester Cristofor
OrderID: 6
Products: blender, air fryer, makeup bag, saucepan, brush set
Respective Quantities: 4, 8, 10, 8, 5
GrandTotal: 5317.356

Customer Name: Loise Arni
OrderID: 7

Products: baking sheet, hair dryer, scale, pot holder, clothes hanger
Respective Quantities: 4, 1, 6, 4, 7
GrandTotal: 696.9053
_____

Customer Name: Nat Kensit
OrderID: 8
Products: blender, umbrella, laptop case, necklace, watch
Respective Quantities: 9, 2, 8, 6, 3
GrandTotal: 3284.319600000001
_____

Customer Name: Aura Sebborn
OrderID: 9
Products: blender, hair dryer, t-shirt, t-shirt, brush set
Respective Quantities: 4, 7, 2, 10, 2
GrandTotal: 7338.3912
_____

Customer Name: Mignonne Dougill
OrderID: 10
Products: mixing bowl, makeup bag, broom, towel rack, laundry basket
Respective Quantities: 5, 8, 4, 7, 5
GrandTotal: 2394.0950000000003
_____

Customer Name: Humfrey Laybourn
OrderID: 11
Products: scale, socks, storage bin, socks, juicer
Respective Quantities: 4, 8, 4, 8, 4
GrandTotal: 6174.856000000002
_____

Customer Name: Vidovic Rudolf
OrderID: 12
Products: water bottle, pot holder, saucepan, lotion, t-shirt
Respective Quantities: 2, 8, 4, 10, 7
GrandTotal: 3576.3099999999995
_____

Customer Name: Sky Keppy
OrderID: 13
Products: baking sheet, candle holder, candle holder, brush set, pencil case
Respective Quantities: 6, 2, 1, 8, 4

GrandTotal: 3284.2446

Customer Name: Brandie Stirton
OrderID: 14
Products: saucepan, t-shirt, laptop case, watch, candle holder
Respective Quantities: 7, 4, 4, 8, 6
GrandTotal: 9598.7137

Customer Name: Linoel Bradley
OrderID: 15
Products: air fryer, umbrella, ring, t-shirt, candle holder
Respective Quantities: 4, 7, 7, 8, 2
GrandTotal: 6440.3636

Customer Name: Anthony Garber
OrderID: 16
Products: mixer, air fryer, water bottle, socks, towel rack
Respective Quantities: 8, 2, 10, 7, 5
GrandTotal: 6440.721299999999

Customer Name: Samuele Stoacley
OrderID: 17
Products: blender, mixer, water bottle, water bottle, storage bin
Respective Quantities: 10, 2, 10, 2, 5
GrandTotal: 15652.906199999998

Customer Name: Kameko Rich
OrderID: 18
Products: mixing bowl, lotion, t-shirt, storage bin, candle holder
Respective Quantities: 9, 6, 8, 1, 5
GrandTotal: 8594.942399999998

Customer Name: Matilda Warham
OrderID: 19
Products: candle holder, pot holder, saucepan, brush set, watch
Respective Quantities: 8, 8, 5, 3, 7
GrandTotal: 12239.338800000003

Customer Name: Theodore Mallinder
OrderID: 20
Products: umbrella, scale, trash can, laptop case, laundry basket
Respective Quantities: 10, 8, 1, 2, 7
GrandTotal: 577.5732000000005

Customer Name: Ferne Babst
OrderID: 21
Products: shoe rack, air fryer, ring, candle holder
Respective Quantities: 8, 1, 2, 10
GrandTotal: 2993.4857999999995

Customer Name: Linette Biddlestone
OrderID: 22
Products: mixer, hat, saucepan, broom, necklace
Respective Quantities: 4, 8, 1, 5, 7
GrandTotal: 6217.5650000000005

Customer Name: Nanon Yorath
OrderID: 23
Products: mixer, water bottle, lotion, necklace, kitchen towel
Respective Quantities: 2, 3, 8, 2, 8
GrandTotal: 3727.849499999999

Customer Name: Rosalinda Trice
OrderID: 24
Products: bath mat, water bottle, saucepan, candle holder, candle holder
Respective Quantities: 7, 2, 2, 10, 8
GrandTotal: 7894.4976

Customer Name: Siobhan Hamper
OrderID: 25
Products: socks, ironing board, pot holder, pencil case
Respective Quantities: 8, 3, 4, 10
GrandTotal: 3115.5681999999997

Customer Name: Baillie Crips

OrderID: 26
Products: scale, shower caddy, candle holder
Respective Quantities: 8, 5, 9
GrandTotal: 2053.8581999999997
_____

Customer Name: Krista Gerckens
OrderID: 27
Products: air fryer, water bottle, broom
Respective Quantities: 5, 5, 2
GrandTotal: 2166.4509
_____

Customer Name: Howey Cargill
OrderID: 28
Products: water bottle, ring, necklace
Respective Quantities: 2, 1, 9
GrandTotal: 4302.2480000000005
_____

Customer Name: Anabal Davydochkin
OrderID: 29
Products: umbrella, mixing bowl, necklace
Respective Quantities: 3, 10, 2
GrandTotal: 9789.7056
_____

Customer Name: Jo-anne Chretien
OrderID: 30
Products: cutlery set, water bottle, ironing board
Respective Quantities: 8, 4, 6
GrandTotal: 2567.8347999999987
_____

Customer Name: Colleen Reeders
OrderID: 31
Products: pencil case, candle holder, pencil case
Respective Quantities: 8, 1, 5
GrandTotal: 3176.8260000000005
_____

Customer Name: Allys Bourdas
OrderID: 32
Products: ironing board, candle holder, laundry basket

Respective Quantities: 2, 1, 7
GrandTotal: 531.9570000000001
————————————————————————————————

Customer Name: Carley Hiddsley
OrderID: 33
Products: toothbrush holder, water bottle, keychain
Respective Quantities: 4, 7, 7
GrandTotal: 8101.0158
————————————————————————————————

Customer Name: Lotti Blackborn
OrderID: 34
Products: candle holder, ring, necklace
Respective Quantities: 9, 9, 10
GrandTotal: 11420.032799999999
————————————————————————————————

Customer Name: Kyla Bartles
OrderID: 35
Products: mixing bowl, water bottle, ring
Respective Quantities: 1, 10, 6
GrandTotal: 1342.4152000000001
————————————————————————————————

Customer Name: Marinna Sleaford
OrderID: 36
Products: cutlery set, shoe rack, socks
Respective Quantities: 10, 6, 1
GrandTotal: 4464.48
————————————————————————————————

Customer Name: Poul Poutress
OrderID: 37
Products: pencil case, brush set, towel rack
Respective Quantities: 6, 1, 6
GrandTotal: 3755.7144000000003
————————————————————————————————

Customer Name: Janette Beckhouse
OrderID: 38
Products: storage bin, laptop case, laundry basket
Respective Quantities: 2, 6, 5
GrandTotal: 424.9742999999999

```
——————————————————————————————————

Customer Name: Kelsey Skace
OrderID: 39
Products: hair dryer, toothbrush holder, air fryer
Respective Quantities: 5, 7, 5
GrandTotal: 7010.204
——————————————————————————————————

Customer Name: Ardenia Boyet
OrderID: 40
Products: candle holder, scale, saucepan
Respective Quantities: 3, 4, 6
GrandTotal: 1450.1016
——————————————————————————————————

Customer Name: Barbara Goding
OrderID: 41
Products: water bottle, storage bin, socks
Respective Quantities: 1, 6, 2
GrandTotal: 1470.3275999999998
——————————————————————————————————

Customer Name: Ximenes Jendas
OrderID: 42
Products: socks, pot holder, kitchen towel
Respective Quantities: 5, 2, 4
GrandTotal: 475.6576
——————————————————————————————————

Customer Name: Andreas Bassindale
OrderID: 43
Products: laptop case, towel rack, necklace
Respective Quantities: 4, 1, 6
GrandTotal: 4557.2841
——————————————————————————————————

Customer Name: Reece Carleton
OrderID: 44
Products: storage bin, socks, brush set
Respective Quantities: 9, 6, 4
GrandTotal: 1924.7256000000002
——————————————————————————————————
```

Customer Name: Harlen Vorley
OrderID: 45
Products: hair dryer, bath mat, ring
Respective Quantities: 3, 10, 10
GrandTotal: 5316.629500000001

Customer Name: Rozalie Bosence
OrderID: 46
Products: scale, makeup bag, brush set
Respective Quantities: 7, 2, 9
GrandTotal: 2401.3176000000003

Customer Name: Lorette Trathan
OrderID: 47
Products: socks, keychain, candle holder
Respective Quantities: 4, 6, 1
GrandTotal: 3383.6884999999997

Customer Name: Corinna Spoors
OrderID: 48
Products: socks, hat, kitchen towel
Respective Quantities: 8, 7, 7
GrandTotal: 687.6891

Customer Name: Kathryn Hemerijk
OrderID: 49
Products: clothes hanger, ring, laptop case
Respective Quantities: 4, 7, 8
GrandTotal: 5017.6144

Customer Name: Lester Cristofor
OrderID: 50
Products: baking sheet, water bottle, brush set
Respective Quantities: 3, 2, 2
GrandTotal: 2383.4835000000003

Customer Name: Loise Arni
OrderID: 51

Products: air fryer, ring
Respective Quantities: 1, 3
GrandTotal: 338.07059999999996

Customer Name: Nat Kensit
OrderID: 52
Products: makeup bag, kitchen towel
Respective Quantities: 8, 8
GrandTotal: 2022.2971999999997

Customer Name: Aura Sebborn
OrderID: 53
Products: scarf, keychain
Respective Quantities: 10, 6
GrandTotal: 4132.5424

Customer Name: Mignonne Dougill
OrderID: 54
Products: mixing bowl, hat
Respective Quantities: 8, 2
GrandTotal: 3610.7100000000005

Customer Name: Humfrey Laybourn
OrderID: 55
Products: toothbrush holder, ring
Respective Quantities: 5, 3
GrandTotal: 1293.3728

Customer Name: Vidovic Rudolf
OrderID: 56
Products: candle holder, water bottle
Respective Quantities: 2, 10
GrandTotal: 3167.4299999999994

Customer Name: Sky Keppy
OrderID: 57
Products: air fryer, water bottle
Respective Quantities: 7, 2

GrandTotal: 5159.988000000001

Customer Name: Brandie Stirton
OrderID: 58
Products: laptop case, kitchen towel
Respective Quantities: 3, 6
GrandTotal: 115.38119999999999

Customer Name: Linoel Bradley
OrderID: 59
Products: mixing bowl, scarf
Respective Quantities: 6, 10
GrandTotal: 4504.0956

Customer Name: Anthony Garber
OrderID: 60
Products: toothbrush holder, mixing bowl
Respective Quantities: 1, 7
GrandTotal: 2572.3568

Customer Name: Samuele Stoacley
OrderID: 61
Products: toothbrush holder, pot holder
Respective Quantities: 4, 2
GrandTotal: 3006.7206

Customer Name: Kameko Rich
OrderID: 62
Products: blender, candle holder
Respective Quantities: 7, 7
GrandTotal: 4999.643000000001

Customer Name: Matilda Warham
OrderID: 63
Products: water bottle, watch
Respective Quantities: 8, 10
GrandTotal: 3816.0096

Customer Name: Theodore Mallinder
OrderID: 64
Products: toothbrush holder, laptop case
Respective Quantities: 6, 2
GrandTotal: 4610.155500000001

Customer Name: Vi Geldard
OrderID: 65
Products: mixing bowl, water bottle
Respective Quantities: 8, 6
GrandTotal: 7367.092200000002

Customer Name: Adiana Eakin
OrderID: 66
Products: toothbrush holder, laundry basket
Respective Quantities: 2, 1
GrandTotal: 0.0

Customer Name: Gary Vonasek
OrderID: 67
Products: mixer, pot holder
Respective Quantities: 6, 5
GrandTotal: 3240.2699999999995

Customer Name: Fonsie Westfalen
OrderID: 68
Products: blender, brush set
Respective Quantities: 8, 5
GrandTotal: 3251.4729

Customer Name: Mary Knox
OrderID: 69
Products: saucepan, watch
Respective Quantities: 6, 9
GrandTotal: 2370.848499999999

Customer Name: Evania de Mullett

OrderID: 70
Products: toothbrush holder, makeup bag
Respective Quantities: 6, 6
GrandTotal: 3872.7002999999995

Customer Name: Thornton Votier
OrderID: 71
Products: hair dryer, water bottle
Respective Quantities: 2, 5
GrandTotal: 55.711800000000046

Customer Name: Benedick Wimmer
OrderID: 72
Products: clothes hanger, laundry basket
Respective Quantities: 6, 8
GrandTotal: 9.566400000000007

Customer Name: Riki Roobottom
OrderID: 73
Products: mixer, juicer
Respective Quantities: 9, 6
GrandTotal: 3405.4524

Customer Name: Queenie Twopenny
OrderID: 74
Products: socks, brush set
Respective Quantities: 10, 8
GrandTotal: 3339.0966000000003

Customer Name: Jeanette Glasscoe
OrderID: 75
Products: saucepan, laundry basket
Respective Quantities: 3, 9
GrandTotal: 237.68689999999998

Customer Name: Dasya Batha
OrderID: 76
Products: scale, makeup bag

Respective Quantities: 1, 10
GrandTotal: 3902.4955

Customer Name: Tiff Prophet
OrderID: 77
Products: hair dryer, scarf
Respective Quantities: 2, 8
GrandTotal: 651.4656000000001

Customer Name: Kalindi Kebell
OrderID: 78
Products: saucepan, clothes hanger
Respective Quantities: 4, 1
GrandTotal: 1258.119

Customer Name: Tamarah Tumbelty
OrderID: 79
Products: air fryer, juicer
Respective Quantities: 3, 8
GrandTotal: 3038.5288

Customer Name: Della Fantini
OrderID: 80
Products: mixing bowl, socks
Respective Quantities: 2, 6
GrandTotal: 2095.4778

Customer Name: Peggie Dinneen
OrderID: 81
Products: ironing board, laundry basket
Respective Quantities: 7, 6
GrandTotal: 2612.4345

Customer Name: Sax Jannex
OrderID: 82
Products: umbrella, scale
Respective Quantities: 4, 1
GrandTotal: 497.7270000000001

```
----------------------------------------

Customer Name: Matty Parks
OrderID: 83
Products: scarf, candle holder
Respective Quantities: 5, 7
GrandTotal: 488.4628
----------------------------------------

Customer Name: Tom Cracker
OrderID: 84
Products: shoe rack, saucepan
Respective Quantities: 8, 3
GrandTotal: 469.3680000000004
----------------------------------------

Customer Name: Cam Spooner
OrderID: 85
Products: air fryer, clothes hanger
Respective Quantities: 4, 4
GrandTotal: 406.84459999999996
----------------------------------------

Customer Name: Orson Parmer
OrderID: 86
Products: mixing bowl, ring
Respective Quantities: 2, 7
GrandTotal: 922.5450000000001
----------------------------------------

Customer Name: Creighton Gerrans
OrderID: 87
Products: umbrella, ironing board
Respective Quantities: 8, 10
GrandTotal: 8112.848800000002
----------------------------------------

Customer Name: Brina Frow
OrderID: 88
Products: umbrella
Respective Quantities: 10
GrandTotal: 4919.39
----------------------------------------
```

Customer Name: Sallee McWhannel
OrderID: 89
Products: t-shirt, candle holder
Respective Quantities: 5, 9
GrandTotal: 414.4175999999998

Customer Name: Orrin Kilfeder
OrderID: 90
Products: baking sheet, shoe rack
Respective Quantities: 8, 5
GrandTotal: 3237.6798

Customer Name: Terrel Edis
OrderID: 91
Products: trash can, clothes hanger
Respective Quantities: 8, 1
GrandTotal: 744.9407999999999

Customer Name: Salome Bruhnicke
OrderID: 92
Products: candle holder, ring
Respective Quantities: 1, 9
GrandTotal: 1217.7099999999998

Customer Name: Maggee Gallie
OrderID: 93
Products: baking sheet, umbrella
Respective Quantities: 7, 8
GrandTotal: 1698.1360000000002

Customer Name: Lil Gare
OrderID: 94
Products: umbrella, clothes hanger
Respective Quantities: 6, 2
GrandTotal: 3899.9295000000006

Customer Name: Creight Shovel
OrderID: 95

Products: t-shirt, clothes hanger
Respective Quantities: 6, 2
GrandTotal: 809.8896000000001

Customer Name: Rainer Torpie
OrderID: 96
Products: baking sheet, towel rack
Respective Quantities: 5, 10
GrandTotal: 4032.3737999999994

Customer Name: Alyse Drover
OrderID: 97
Products: hat, trash can
Respective Quantities: 9, 1
GrandTotal: 166.56640000000002

Customer Name: Harlen Vorley
OrderID: 98
Products: candle holder, lotion
Respective Quantities: 3, 5
GrandTotal: 1492.3818

Customer Name: Rozalie Bosence
OrderID: 99
Products: baking sheet, hair dryer
Respective Quantities: 1, 1
GrandTotal: 1094.232

Customer Name: Lorette Trathan
OrderID: 100
Products: clothes hanger, necklace
Respective Quantities: 4, 10
GrandTotal: 7009.990000000002

Customer Name: Corinna Spoors
OrderID: 101
Products: clothes hanger
Respective Quantities: 2

GrandTotal: 213.95340000000002

Customer Name: Kathryn Hemerijk
OrderID: 102
Products: towel rack
Respective Quantities: 4
GrandTotal: 514.8948999999999

Customer Name: Aymer Sidnell
OrderID: 103
Products: hat
Respective Quantities: 10
GrandTotal: 808.7844

Customer Name: Jonell Biggs
OrderID: 104
Products: lotion
Respective Quantities: 1
GrandTotal: 12.48300000000001

Customer Name: Ronnie Dadds
OrderID: 105
Products: hair dryer
Respective Quantities: 10
GrandTotal: 4694.7064

Customer Name: Nisse Willisch
OrderID: 106
Products: necklace
Respective Quantities: 4
GrandTotal: 742.4838999999998

Customer Name: Federica Bickardike
OrderID: 107
Products: laundry basket
Respective Quantities: 5
GrandTotal: 78.33599999999998

```
Customer Name: Anson Americi
OrderID: 108
Products: scarf
Respective Quantities: 7
GrandTotal: 1573.8143999999998
----------------------------------

Customer Name: Cam Spooner
OrderID: 109
Products: bath mat
Respective Quantities: 10
GrandTotal: 16.8944
----------------------------------

Customer Name: Orson Parmer
OrderID: 110
Products: laptop case
Respective Quantities: 7
GrandTotal: 18.13080000000002
----------------------------------

Customer Name: Creighton Gerrans
OrderID: 111
Products: lotion
Respective Quantities: 1
GrandTotal: 2.535300000000002
----------------------------------

Customer Name: Brina Frow
OrderID: 112
Products: makeup bag
Respective Quantities: 5
GrandTotal: 1671.4782
----------------------------------

Customer Name: Sallee McWhannel
OrderID: 113
Products: hair dryer
Respective Quantities: 4
GrandTotal: 1566.5772
----------------------------------

Customer Name: Orrin Kilfeder
```

OrderID: 114
Products: necklace
Respective Quantities: 9
GrandTotal: 8371.001000000002

Customer Name: Terrel Edis
OrderID: 115
Products: brush set
Respective Quantities: 1
GrandTotal: 59.26080000000002

Customer Name: Salome Bruhnicke
OrderID: 116
Products: mixing bowl
Respective Quantities: 9
GrandTotal: 3836.7816000000003

Customer Name: Maggee Gallie
OrderID: 117
Products: juicer
Respective Quantities: 9
GrandTotal: 1618.9952999999998

Customer Name: Lil Gare
OrderID: 118
Products: water bottle
Respective Quantities: 5
GrandTotal: 3338.8415999999997

Customer Name: Creight Shovel
OrderID: 119
Products: pencil case
Respective Quantities: 4
GrandTotal: 380.23019999999997

Customer Name: Nisse Willisch
OrderID: 120
Products: t-shirt

Respective Quantities: 10
GrandTotal: 397.9788000000003

Customer Name: Federica Bickardike
OrderID: 121
Products: air fryer
Respective Quantities: 1
GrandTotal: 509.7664000000001

Customer Name: Anson Americi
OrderID: 122
Products: water bottle
Respective Quantities: 2
GrandTotal: 16.123500000000014

Customer Name: Cam Spooner
OrderID: 123
Products: water bottle
Respective Quantities: 1
GrandTotal: 590.8888000000001

Customer Name: Orson Parmer
OrderID: 124
Products: lotion
Respective Quantities: 6
GrandTotal: 604.8159

Customer Name: Creighton Gerrans
OrderID: 125
Products: watch
Respective Quantities: 3
GrandTotal: 2512.7547999999997

Customer Name: Brina Frow
OrderID: 126
Products: laundry basket
Respective Quantities: 1
GrandTotal: 0.36300000000000027

```
------------------------------------

Customer Name: Sallee McWhannel
OrderID: 127
Products: shoe rack
Respective Quantities: 4
GrandTotal: 2240.9464000000003
------------------------------------

Customer Name: Orrin Kilfeder
OrderID: 128
Products: saucepan
Respective Quantities: 7
GrandTotal: 2003.3227000000004
------------------------------------

Customer Name: Terrel Edis
OrderID: 129
Products: lotion
Respective Quantities: 8
GrandTotal: 1136.454
------------------------------------

Customer Name: Colleen Reeders
OrderID: 130
Products: candle holder
Respective Quantities: 7
GrandTotal: 73.00349999999999
------------------------------------

Customer Name: Allys Bourdas
OrderID: 131
Products: socks
Respective Quantities: 7
GrandTotal: 196.1575
------------------------------------

Customer Name: Carley Hiddsley
OrderID: 132
Products: shower caddy
Respective Quantities: 4
GrandTotal: 1298.4582999999998
------------------------------------
```

Customer Name: Lotti Blackborn
OrderID: 133
Products: broom
Respective Quantities: 5
GrandTotal: 174.79680000000005

Customer Name: Kyla Bartles
OrderID: 134
Products: water bottle
Respective Quantities: 5
GrandTotal: 1695.0990000000002

Customer Name: Lester Cristofor
OrderID: 135
Products: umbrella
Respective Quantities: 3
GrandTotal: 1186.77

Customer Name: Loise Arni
OrderID: 136
Products: ironing board
Respective Quantities: 2
GrandTotal: 324.316

Customer Name: Nat Kensit
OrderID: 137
Products: candle holder
Respective Quantities: 8
GrandTotal: 58.72959999999999

Customer Name: Aura Sebborn
OrderID: 138
Products: ironing board
Respective Quantities: 2
GrandTotal: 48.70650000000004

Customer Name: Mignonne Dougill
OrderID: 139

Products: keychain
Respective Quantities: 7
GrandTotal: 389.76959999999985

Customer Name: Humfrey Laybourn
OrderID: 140
Products: necklace
Respective Quantities: 1
GrandTotal: 545.1824000000001

Customer Name: Vidovic Rudolf
OrderID: 141
Products: ring
Respective Quantities: 10
GrandTotal: 1374.336

Customer Name: Sky Keppy
OrderID: 142
Products: shoe rack
Respective Quantities: 9
GrandTotal: 1246.8312999999996

Customer Name: Brandie Stirton
OrderID: 143
Products: brush set
Respective Quantities: 7
GrandTotal: 2269.377

Customer Name: Orrin Kilfeder
OrderID: 144
Products: laptop case
Respective Quantities: 9
GrandTotal: 425.67800000000005

Customer Name: Terrel Edis
OrderID: 145
Products: air fryer
Respective Quantities: 4

GrandTotal: 70.50090000000007

Customer Name: Salome Bruhnicke
OrderID: 146
Products: saucepan
Respective Quantities: 6
GrandTotal: 3355.5059999999994

Customer Name: Maggee Gallie
OrderID: 147
Products: storage bin
Respective Quantities: 6
GrandTotal: 73.4712

Customer Name: Lil Gare
OrderID: 148
Products: pot holder
Respective Quantities: 6
GrandTotal: 473.50669999999997

Customer Name: Creight Shovel
OrderID: 149
Products: necklace
Respective Quantities: 9
GrandTotal: 8545.9716

Customer Name: Rainer Torpie
OrderID: 150
Products: socks
Respective Quantities: 4
GrandTotal: 771.2991000000001

Customer Name: Maggee Gallie
OrderID: 151
Products: bath mat
Respective Quantities: 8
GrandTotal: 64.14750000000001

Customer Name: Lil Gare
OrderID: 152
Products: brush set
Respective Quantities: 3
GrandTotal: 753.7098

Customer Name: Creight Shovel
OrderID: 153
Products: candle holder
Respective Quantities: 7
GrandTotal: 2502.6614999999997

Customer Name: Nisse Willisch
OrderID: 154
Products: socks
Respective Quantities: 9
GrandTotal: 135.7756

Customer Name: Federica Bickardike
OrderID: 155
Products: clothes hanger
Respective Quantities: 5
GrandTotal: 4387.3375

Customer Name: Ronnie Dadds
OrderID: 156
Products: baking sheet
Respective Quantities: 5
GrandTotal: 2985.0015999999996

Customer Name: Sherman Schoenrock
OrderID: 157
Products: mixer
Respective Quantities: 7
GrandTotal: 2604.745

Customer Name: Barris Bog

OrderID: 158
Products: storage bin
Respective Quantities: 7
GrandTotal: 27.201000000000004

Customer Name: Anthe Whitham
OrderID: 159
Products: saucepan
Respective Quantities: 1
GrandTotal: 232.25649999999996

Customer Name: Cam Spooner
OrderID: 160
Products: shoe rack
Respective Quantities: 5
GrandTotal: 582.4496000000001

Customer Name: Orson Parmer
OrderID: 161
Products: ring
Respective Quantities: 4
GrandTotal: 410.499

Customer Name: Creighton Gerrans
OrderID: 162
Products: blender
Respective Quantities: 1
GrandTotal: 512.1472

Customer Name: Brina Frow
OrderID: 163
Products: clothes hanger
Respective Quantities: 8
GrandTotal: 270.0225

Customer Name: Lester Cristofor
OrderID: 164
Products: laptop case

Respective Quantities: 2
GrandTotal: 125.26500000000001

Customer Name: Loise Arni
OrderID: 165
Products: t-shirt
Respective Quantities: 1
GrandTotal: 208.47329999999997

Customer Name: Nat Kensit
OrderID: 166
Products: laundry basket
Respective Quantities: 3
GrandTotal: 27.5865

Customer Name: Aura Sebborn
OrderID: 167
Products: socks
Respective Quantities: 7
GrandTotal: 106.97720000000001

Customer Name: Mignonne Dougill
OrderID: 168
Products: pot holder
Respective Quantities: 6
GrandTotal: 200.20299999999997

Customer Name: Humfrey Laybourn
OrderID: 169
Products: candle holder
Respective Quantities: 5
GrandTotal: 13.532399999999999

Customer Name: Vidovic Rudolf
OrderID: 170
Products: saucepan
Respective Quantities: 2
GrandTotal: 822.2377999999999

```
------------------------------------

Customer Name: Sky Keppy
OrderID: 171
Products: candle holder
Respective Quantities: 9
GrandTotal: 124.56670000000001
------------------------------------

Customer Name: Brandie Stirton
OrderID: 172
Products: air fryer
Respective Quantities: 10
GrandTotal: 470.9319999999998
------------------------------------

Customer Name: Linoel Bradley
OrderID: 173
Products: blender
Respective Quantities: 9
GrandTotal: 4651.392
------------------------------------

Customer Name: Anthony Garber
OrderID: 174
Products: mixing bowl
Respective Quantities: 3
GrandTotal: 86.12500000000007
------------------------------------

Customer Name: Samuele Stoacley
OrderID: 175
Products: saucepan
Respective Quantities: 2
GrandTotal: 119.44169999999995
------------------------------------

Customer Name: Kameko Rich
OrderID: 176
Products: trash can
Respective Quantities: 5
GrandTotal: 701.2199999999999
------------------------------------
```

```
Customer Name: Matilda Warham
OrderID: 177
Products: candle holder
Respective Quantities: 6
GrandTotal: 413.51039999999995
-----------------------------------

Customer Name: Federica Bickardike
OrderID: 178
Products: hat
Respective Quantities: 2
GrandTotal: 33.140800000000006
-----------------------------------

Customer Name: Salome Bruhnicke
OrderID: 179
Products: kitchen towel
Respective Quantities: 9
GrandTotal: 20.396000000000015
-----------------------------------

Customer Name: Maggee Gallie
OrderID: 180
Products: bath mat
Respective Quantities: 4
GrandTotal: 6.448100000000001
-----------------------------------

Customer Name: Lil Gare
OrderID: 181
Products: pot holder
Respective Quantities: 10
GrandTotal: 663.2025
-----------------------------------

Customer Name: Creight Shovel
OrderID: 182
Products: ring
Respective Quantities: 1
GrandTotal: 251.64
-----------------------------------

Customer Name: Rainer Torpie
OrderID: 183
```

Products: air fryer
Respective Quantities: 1
GrandTotal: 581.6943

Customer Name: Maggee Gallie
OrderID: 184
Products: necklace
Respective Quantities: 2
GrandTotal: 331.69380000000007

Customer Name: Lil Gare
OrderID: 185
Products: umbrella
Respective Quantities: 9
GrandTotal: 1642.8876999999998

Customer Name: Creight Shovel
OrderID: 186
Products: laptop case
Respective Quantities: 7
GrandTotal: 138.72220000000002

Customer Name: Rainer Torpie
OrderID: 187
Products: brush set
Respective Quantities: 9
GrandTotal: 388.95360000000034

Customer Name: Alyse Drover
OrderID: 188
Products: pot holder
Respective Quantities: 3
GrandTotal: 83.62239999999998

Customer Name: Harlen Vorley
OrderID: 189
Products: umbrella
Respective Quantities: 4

GrandTotal: 507.1248000000001

Customer Name: Rozalie Bosence
OrderID: 190
Products: blender
Respective Quantities: 6
GrandTotal: 2905.608

Customer Name: Lorette Trathan
OrderID: 191
Products: broom
Respective Quantities: 6
GrandTotal: 209.63040000000004

Customer Name: Corinna Spoors
OrderID: 192
Products: scale
Respective Quantities: 10
GrandTotal: 583.5840000000001

Customer Name: Kathryn Hemerijk
OrderID: 193
Products: lotion
Respective Quantities: 9
GrandTotal: 2142.0728000000004

Customer Name: Aymer Sidnell
OrderID: 194
Products: candle holder
Respective Quantities: 1
GrandTotal: 224.8158

Customer Name: Jonell Biggs
OrderID: 195
Products: candle holder
Respective Quantities: 8
GrandTotal: 3556.0483999999997

Customer Name: Ronnie Dadds
OrderID: 196
Products: umbrella
Respective Quantities: 4
GrandTotal: 2280.4344

Customer Name: Nisse Willisch
OrderID: 197
Products: socks
Respective Quantities: 3
GrandTotal: 88.264

Customer Name: Federica Bickardike
OrderID: 198
Products: mixer
Respective Quantities: 2
GrandTotal: 1276.9882

Customer Name: Anson Americi
OrderID: 199
Products: storage bin
Respective Quantities: 7
GrandTotal: 27.487600000000004

Customer Name: Cam Spooner
OrderID: 200
Products: candle holder
Respective Quantities: 4
GrandTotal: 491.02000000000004

Customer Name: Orson Parmer
OrderID: 201
Products: umbrella
Respective Quantities: 7
GrandTotal: 1777.104

Customer Name: Creighton Gerrans

OrderID: 202
Products: shoe rack
Respective Quantities: 8
GrandTotal: 3906.5287999999996

Customer Name: Brina Frow
OrderID: 203
Products: broom
Respective Quantities: 4
GrandTotal: 8.703300000000008

Customer Name: Sallee McWhannel
OrderID: 204
Products: mixer
Respective Quantities: 7
GrandTotal: 104.1916000000001

Customer Name: Orrin Kilfeder
OrderID: 205
Products: saucepan
Respective Quantities: 8
GrandTotal: 1543.7744

Customer Name: Lester Cristofor
OrderID: 206
Products: ironing board
Respective Quantities: 10
GrandTotal: 6917.512500000001

Customer Name: Loise Arni
OrderID: 207
Products: water bottle
Respective Quantities: 5
GrandTotal: 725.8860000000002

Customer Name: Nat Kensit
OrderID: 208
Products: keychain

```
Respective Quantities: 8
GrandTotal: 1161.9335999999998
----------------------------------

Customer Name: Nisse Willisch
OrderID: 209
Products: cutlery set
Respective Quantities: 8
GrandTotal: 594.7017999999999
----------------------------------

Customer Name: Federica Bickardike
OrderID: 210
Products: clothes hanger
Respective Quantities: 1
GrandTotal: 72.1791
----------------------------------

Customer Name: Anson Americi
OrderID: 211
Products: water bottle
Respective Quantities: 7
GrandTotal: 3162.0030000000006
----------------------------------

Customer Name: Cam Spooner
OrderID: 212
Products: toothbrush holder
Respective Quantities: 2
GrandTotal: 73.38120000000006
----------------------------------

Customer Name: Federica Bickardike
OrderID: 213
Products: brush set
Respective Quantities: 2
GrandTotal: 0.0
----------------------------------

Customer Name: Anson Americi
OrderID: 214
Products: scarf
Respective Quantities: 10
GrandTotal: 1206.0858
```

```
————————————————————————————————

Customer Name: Cam Spooner
OrderID: 215
Products: watch
Respective Quantities: 9
GrandTotal: 934.3521
————————————————————————————————

Customer Name: Orson Parmer
OrderID: 216
Products: clothes hanger
Respective Quantities: 5
GrandTotal: 327.14849999999996
————————————————————————————————

Customer Name: Creighton Gerrans
OrderID: 217
Products: clothes hanger
Respective Quantities: 1
GrandTotal: 117.0516
————————————————————————————————

Customer Name: Brina Frow
OrderID: 218
Products: watch
Respective Quantities: 10
GrandTotal: 6700.5611
————————————————————————————————

Customer Name: Sallee McWhannel
OrderID: 219
Products: shower caddy
Respective Quantities: 8
GrandTotal: 2672.49
————————————————————————————————

Customer Name: Samuele Stoacley
OrderID: 220
Products: umbrella
Respective Quantities: 8
GrandTotal: 1650.883
————————————————————————————————
```

Customer Name: Kameko Rich
OrderID: 221
Products: brush set
Respective Quantities: 7
GrandTotal: 236.66939999999994

Customer Name: Matilda Warham
OrderID: 222
Products: ring
Respective Quantities: 4
GrandTotal: 806.5595000000001

Customer Name: Theodore Mallinder
OrderID: 223
Products: scarf
Respective Quantities: 9
GrandTotal: 457.0384000000002

Customer Name: Anthe Whitham
OrderID: 224
Products: air fryer
Respective Quantities: 10
GrandTotal: 3001.4316000000003

Customer Name: Cam Spooner
OrderID: 225
Products: scarf
Respective Quantities: 1
GrandTotal: 65.04749999999999

Customer Name: Orson Parmer
OrderID: 226
Products: candle holder
Respective Quantities: 6
GrandTotal: 20.352000000000004

Customer Name: Maggee Gallie
OrderID: 227

Products: makeup bag
Respective Quantities: 2
GrandTotal: 340.88680000000005

Customer Name: Lil Gare
OrderID: 228
Products: candle holder
Respective Quantities: 5
GrandTotal: 2035.5833

Customer Name: Creight Shovel
OrderID: 229
Products: candle holder
Respective Quantities: 4
GrandTotal: 137.3021999999999

Customer Name: Nisse Willisch
OrderID: 230
Products: kitchen towel
Respective Quantities: 2
GrandTotal: 36.679

Customer Name: Federica Bickardike
OrderID: 231
Products: juicer
Respective Quantities: 4
GrandTotal: 1492.4817000000003

Customer Name: Anson Americi
OrderID: 232
Products: umbrella
Respective Quantities: 4
GrandTotal: 285.7670999999999

Customer Name: Cam Spooner
OrderID: 233
Products: mixing bowl
Respective Quantities: 4

```
GrandTotal: 391.19380000000007
----------------------------------

Customer Name: Orson Parmer
OrderID: 234
Products: necklace
Respective Quantities: 10
GrandTotal: 8414.2142
----------------------------------

Customer Name: Creighton Gerrans
OrderID: 235
Products: ironing board
Respective Quantities: 2
GrandTotal: 307.87409999999994
----------------------------------

Customer Name: Brina Frow
OrderID: 236
Products: pencil case
Respective Quantities: 6
GrandTotal: 1032.7115
----------------------------------

Customer Name: Maggee Gallie
OrderID: 237
Products: clothes hanger
Respective Quantities: 7
GrandTotal: 330.9145
----------------------------------

Customer Name: Lil Gare
OrderID: 238
Products: hat
Respective Quantities: 7
GrandTotal: 478.59499999999997
----------------------------------

Customer Name: Creight Shovel
OrderID: 239
Products: pencil case
Respective Quantities: 2
GrandTotal: 247.2799
----------------------------------
```

Customer Name: Rainer Torpie
OrderID: 240
Products: candle holder
Respective Quantities: 1
GrandTotal: 269.31800000000004

Customer Name: Alyse Drover
OrderID: 241
Products: ironing board
Respective Quantities: 8
GrandTotal: 845.8177000000001

Customer Name: Harlen Vorley
OrderID: 242
Products: hair dryer
Respective Quantities: 10
GrandTotal: 3077.344

Customer Name: Rozalie Bosence
OrderID: 243
Products: mixer
Respective Quantities: 8
GrandTotal: 238.17080000000024

Customer Name: Lorette Trathan
OrderID: 244
Products: lotion
Respective Quantities: 5
GrandTotal: 129.42099999999996

Customer Name: Corinna Spoors
OrderID: 245
Products: baking sheet
Respective Quantities: 2
GrandTotal: 292.39439999999996

Customer Name: Kathryn Hemerijk

OrderID: 246
Products: pencil case
Respective Quantities: 2
GrandTotal: 315.1916

Customer Name: Aymer Sidnell
OrderID: 247
Products: air fryer
Respective Quantities: 3
GrandTotal: 109.87080000000002

Customer Name: Jonell Biggs
OrderID: 248
Products: t-shirt
Respective Quantities: 6
GrandTotal: 4059.8992

Customer Name: Maggee Gallie
OrderID: 249
Products: laptop case
Respective Quantities: 1
GrandTotal: 66.946

Customer Name: Lil Gare
OrderID: 250
Products: brush set
Respective Quantities: 2
GrandTotal: 201.70350000000002

Customer Name: Creight Shovel
OrderID: 251
Products: shower caddy
Respective Quantities: 7
GrandTotal: 2058.0845999999997

Customer Name: Rainer Torpie
OrderID: 252
Products: candle holder

```
Respective Quantities: 10
GrandTotal: 84.67559999999999
----------------------------------

Customer Name: Alyse Drover
OrderID: 253
Products: juicer
Respective Quantities: 6
GrandTotal: 248.32169999999994
----------------------------------

Customer Name: Harlen Vorley
OrderID: 254
Products: lotion
Respective Quantities: 6
GrandTotal: 387.97249999999997
----------------------------------

Customer Name: Rozalie Bosence
OrderID: 255
Products: pot holder
Respective Quantities: 1
GrandTotal: 35.088
----------------------------------

Customer Name: Lorette Trathan
OrderID: 256
Products: makeup bag
Respective Quantities: 6
GrandTotal: 809.5842
----------------------------------

Customer Name: Corinna Spoors
OrderID: 257
Products: shower caddy
Respective Quantities: 5
GrandTotal: 1668.4574999999998
----------------------------------

Customer Name: Samuele Stoacley
OrderID: 258
Products: mixing bowl
Respective Quantities: 6
GrandTotal: 2557.3438
```

```
_____

Customer Name: Kameko Rich
OrderID: 259
Products: candle holder
Respective Quantities: 6
GrandTotal: 26.481999999999992
_____

Customer Name: Matilda Warham
OrderID: 260
Products: shoe rack
Respective Quantities: 2
GrandTotal: 1267.7292
_____

Customer Name: Theodore Mallinder
OrderID: 261
Products: pencil case
Respective Quantities: 4
GrandTotal: 478.36719999999997
_____

Customer Name: Kameko Rich
OrderID: 262
Products: clothes hanger
Respective Quantities: 4
GrandTotal: 526.9362
_____

Customer Name: Matilda Warham
OrderID: 263
Products: storage bin
Respective Quantities: 3
GrandTotal: 5.109000000000001
_____

Customer Name: Theodore Mallinder
OrderID: 264
Products: makeup bag
Respective Quantities: 7
GrandTotal: 1333.1904
_____
```

```
Customer Name: Aymer Sidnell
OrderID: 265
Products: candle holder
Respective Quantities: 3
GrandTotal: 103.45600000000002
-----------------------------------

Customer Name: Jonell Biggs
OrderID: 266
Products: keychain
Respective Quantities: 6
GrandTotal: 219.57389999999995
-----------------------------------

Customer Name: Maggee Gallie
OrderID: 267
Products: blender
Respective Quantities: 7
GrandTotal: 1640.4865000000002
-----------------------------------

Customer Name: Lil Gare
OrderID: 268
Products: baking sheet
Respective Quantities: 8
GrandTotal: 1217.5449999999998
-----------------------------------

Customer Name: Creight Shovel
OrderID: 269
Products: necklace
Respective Quantities: 10
GrandTotal: 9590.5348
-----------------------------------

Customer Name: Rainer Torpie
OrderID: 270
Products: hair dryer
Respective Quantities: 8
GrandTotal: 4063.204199999999
-----------------------------------

Customer Name: Alyse Drover
OrderID: 271
```

Products: pot holder
Respective Quantities: 2
GrandTotal: 47.7549

Customer Name: Harlen Vorley
OrderID: 272
Products: pencil case
Respective Quantities: 1
GrandTotal: 34.27020000000001

Customer Name: Brina Frow
OrderID: 273
Products: candle holder
Respective Quantities: 10
GrandTotal: 5414.964800000001

Customer Name: Sallee McWhannel
OrderID: 274
Products: laundry basket
Respective Quantities: 6
GrandTotal: 67.1549

Customer Name: Creighton Gerrans
OrderID: 275
Products: candle holder
Respective Quantities: 5
GrandTotal: 2125.2439999999997

Customer Name: Brina Frow
OrderID: 276
Products: watch
Respective Quantities: 2
GrandTotal: 1376.78

Customer Name: Lester Cristofor
OrderID: 277
Products: laundry basket
Respective Quantities: 5

GrandTotal: 49.2102

Customer Name: Loise Arni
OrderID: 278
Products: makeup bag
Respective Quantities: 3
GrandTotal: 106.91189999999997

Customer Name: Nat Kensit
OrderID: 279
Products: necklace
Respective Quantities: 3
GrandTotal: 1085.3358

Customer Name: Aura Sebborn
OrderID: 280
Products: juicer
Respective Quantities: 1
GrandTotal: 257.60010000000005

Customer Name: Mignonne Dougill
OrderID: 281
Products: clothes hanger
Respective Quantities: 2
GrandTotal: 44.579100000000004

Customer Name: Humfrey Laybourn
OrderID: 282
Products: towel rack
Respective Quantities: 3
GrandTotal: 616.2022

Customer Name: Vidovic Rudolf
OrderID: 283
Products: bath mat
Respective Quantities: 7
GrandTotal: 8.727599999999999

Customer Name: Sky Keppy
OrderID: 284
Products: hat
Respective Quantities: 7
GrandTotal: 178.26239999999999

Customer Name: Anson Americi
OrderID: 285
Products: candle holder
Respective Quantities: 10
GrandTotal: 1129.7347

Customer Name: Cam Spooner
OrderID: 286
Products: clothes hanger
Respective Quantities: 3
GrandTotal: 340.46039999999994

Customer Name: Orson Parmer
OrderID: 287
Products: clothes hanger
Respective Quantities: 7
GrandTotal: 4587.2887

Customer Name: Creighton Gerrans
OrderID: 288
Products: keychain
Respective Quantities: 6
GrandTotal: 554.0037

Customer Name: Brina Frow
OrderID: 289
Products: storage bin
Respective Quantities: 5
GrandTotal: 64.848

Customer Name: Samuele Stoacley

OrderID: 290
Products: hair dryer
Respective Quantities: 2
GrandTotal: 520.6487999999999

Customer Name: Kameko Rich
OrderID: 291
Products: socks
Respective Quantities: 2
GrandTotal: 411.4320000000001

Customer Name: Matilda Warham
OrderID: 292
Products: candle holder
Respective Quantities: 8
GrandTotal: 2908.8068000000003

Customer Name: Theodore Mallinder
OrderID: 293
Products: clothes hanger
Respective Quantities: 10
GrandTotal: 1189.3024

Customer Name: Matilda Warham
OrderID: 294
Products: laptop case
Respective Quantities: 3
GrandTotal: 207.68000000000004

Customer Name: Theodore Mallinder
OrderID: 295
Products: makeup bag
Respective Quantities: 2
GrandTotal: 653.3677

Customer Name: Orson Parmer
OrderID: 296
Products: air fryer

```
Respective Quantities: 7
GrandTotal: 1193.7154000000005
----------------------------------

Customer Name: Creighton Gerrans
OrderID: 297
Products: candle holder
Respective Quantities: 3
GrandTotal: 1320.9840000000002
----------------------------------

Customer Name: Brina Frow
OrderID: 298
Products: shower caddy
Respective Quantities: 9
GrandTotal: 3971.8205999999996
----------------------------------

Customer Name: Cam Spooner
OrderID: 299
Products: t-shirt
Respective Quantities: 6
GrandTotal: 1511.3978000000002
----------------------------------

Customer Name: Orson Parmer
OrderID: 300
Products: air fryer
Respective Quantities: 3
GrandTotal: 25.453200000000024
----------------------------------
```

2d) _Customers and Baskets_

To get the relevant information, I wrote a query based primarily on _MATCH_ . As this is only for customers who have payment types thats is a VoucherGift, by matching those, I automatically filter to get only relevant data. For those customers, I then retrieve the products they have in their Baskets by matching customer to basket to basket item and then product. I used _COLLECT_ to aggregate the data and group the product names into a single list. Then, I used _RETURN_ to display customer email, name, birthday, gift card balance, and the products in their baskets. To make sure the results are readible, I formatted the print function to make the output visually simple and ordered.

```python
In [15]:   # query to get relevant customer details
           customer_gift = '''
           MATCH (c:Customer)-[:HAS_PAYMENT]->(pt:PaymentType)-[:IS_A]->(vg:VoucherGift),
                 (c)-[:HAS_BASKET]->(b:Basket)-[:CONTAINS_ITEM]->(bi:BasketItem)-[:IS_OF_PRODUCT]->(p:Product)
           RETURN c.Email AS Email,
                  c.Name AS Name,
                  c.BirthDate AS Birthday,
                  vg.CurrentBalance AS GiftCardBalance,
                  COLLECT(p.ProductName) AS ProductsInBasket
           '''

           # running query
           with driver.session(database="neo4j") as session:
               result = session.run(customer_gift)
               for record in result:
                   print(f"\nEmail: {record['Email']}\nName: {record['Name']}\nBirthday: {record['Birthday']}\n"
                         f"Gift Card Balance: {record['GiftCardBalance']}\nProducts in Basket: {record['ProductsInBasket']}")
                   print('-' *19)
```

Email: ipanons1r@hostgator.com
Name: Ileane Panons
Birthday: 1957-01-06
Gift Card Balance: 428.86
Products in Basket: ['hat']

Email: shampero@paypal.com
Name: Siobhan Hamper
Birthday: 1958-02-27
Gift Card Balance: 105.32
Products in Basket: ['makeup bag', 'clothes hanger']

Email: cspooner2c@hibu.com
Name: Cam Spooner
Birthday: 1997-04-05
Gift Card Balance: 361.58
Products in Basket: ['clothes hanger']

Email: abassindale16@reuters.com
Name: Andreas Bassindale
Birthday: 1988-08-08
Gift Card Balance: 473.48
Products in Basket: ['pencil case', 'cutlery set']

Email: msleafordz@discovery.com
Name: Marinna Sleaford
Birthday: 2000-10-12
Gift Card Balance: 372.83
Products in Basket: ['towel rack', 't-shirt']

Email: ltrathan1a@buzzfeed.com
Name: Lorette Trathan
Birthday: 1966-04-26
Gift Card Balance: 161.0
Products in Basket: ['hair dryer', 'clothes hanger']

Email: shampero@paypal.com
Name: Siobhan Hamper

Birthday: 1958-02-27
Gift Card Balance: 173.95
Products in Basket: ['makeup bag', 'clothes hanger']

Email: lgare2l@unicef.org
Name: Lil Gare
Birthday: 1983-09-03
Gift Card Balance: 193.29
Products in Basket: ['laundry basket']

Email: rcarleton17@pcworld.com
Name: Reece Carleton
Birthday: 1982-02-02
Gift Card Balance: 158.9
Products in Basket: ['pot holder', 'saucepan']

Email: abassindale16@reuters.com
Name: Andreas Bassindale
Birthday: 1988-08-08
Gift Card Balance: 112.43
Products in Basket: ['pencil case', 'cutlery set']

Email: lblackbornx@yale.edu
Name: Lotti Blackborn
Birthday: 1952-08-04
Gift Card Balance: 17.06
Products in Basket: ['laundry basket']

Email: dbatha23@utexas.edu
Name: Dasya Batha
Birthday: 1967-10-25
Gift Card Balance: 96.7
Products in Basket: ['ring']

Email: rtorpie2n@disqus.com
Name: Rainer Torpie
Birthday: 1991-08-05
Gift Card Balance: 329.31

Products in Basket: ['kitchen towel']

Email: adavydochkins@dell.com
Name: Anabal Davydochkin
Birthday: 2003-07-15
Gift Card Balance: 1.02
Products in Basket: ['kitchen towel', 'clothes hanger']

Email: adrover2o@woothemes.com
Name: Alyse Drover
Birthday: 2002-10-18
Gift Card Balance: 425.28
Products in Basket: ['socks']

Email: lcristofor5@pagesperso-orange.fr
Name: Lester Cristofor
Birthday: 1998-10-27
Gift Card Balance: 297.73
Products in Basket: ['air fryer', 'storage bin', 'laundry basket']

Email: ljoscelyn2r@globo.com
Name: Luigi Joscelyn
Birthday: 1967-09-11
Gift Card Balance: 436.98
Products in Basket: ['saucepan']

Email: fbickardike1h@vinaora.com
Name: Federica Bickardike
Birthday: 1992-02-22
Gift Card Balance: 488.36
Products in Basket: ['umbrella']

Email: cshovel2m@craigslist.org
Name: Creight Shovel
Birthday: 1952-10-19
Gift Card Balance: 197.3
Products in Basket: ['shoe rack']

Email: chiddsleyw@elegantthemes.com
Name: Carley Hiddsley
Birthday: 1998-05-20
Gift Card Balance: 81.48
Products in Basket: ['socks', 'scale']
_____

Email: lcristofor5@pagesperso-orange.fr
Name: Lester Cristofor
Birthday: 1998-10-27
Gift Card Balance: 476.6
Products in Basket: ['air fryer', 'storage bin', 'laundry basket']
_____

Email: creedersu@weebly.com
Name: Colleen Reeders
Birthday: 1971-10-26
Gift Card Balance: 476.4
Products in Basket: ['cutlery set', 'baking sheet']
_____

Email: amattock2p@alexa.com
Name: Avie Mattock
Birthday: 2004-02-07
Gift Card Balance: 21.25
Products in Basket: ['kitchen towel']
_____

Email: cshovel2m@craigslist.org
Name: Creight Shovel
Birthday: 1952-10-19
Gift Card Balance: 14.04
Products in Basket: ['shoe rack']
_____

Email: ppoutress10@nytimes.com
Name: Poul Poutress
Birthday: 2001-01-27
Gift Card Balance: 327.24
Products in Basket: ['candle holder', 't-shirt']
_____

Email: cdewett0@hugedomains.com

Name: Calhoun Dewett
Birthday: 1/26/1969
Gift Card Balance: 264.95
Products in Basket: ['broom', 'clothes hanger', 'kitchen towel']
———————————————————

Email: fwestfalen1v@linkedin.com
Name: Fonsie Westfalen
Birthday: 1961-03-23
Gift Card Balance: 175.58
Products in Basket: ['makeup bag']
———————————————————

Email: mknox1w@github.io
Name: Mary Knox
Birthday: 1980-02-07
Gift Card Balance: 270.86
Products in Basket: ['umbrella']
———————————————————

Email: mparks2a@ehow.com
Name: Matty Parks
Birthday: 1986-05-10
Gift Card Balance: 114.02
Products in Basket: ['towel rack']
———————————————————

Email: jglasscoe22@pinterest.com
Name: Jeanette Glasscoe
Birthday: 1953-11-14
Gift Card Balance: 352.34
Products in Basket: ['air fryer']
———————————————————

Email: fwestfalen1v@linkedin.com
Name: Fonsie Westfalen
Birthday: 1961-03-23
Gift Card Balance: 179.81
Products in Basket: ['makeup bag']
———————————————————

Email: pdinneen28@marriott.com
Name: Peggie Dinneen
Birthday: 1955-10-02

Gift Card Balance: 221.06
Products in Basket: ['clothes hanger']
───────────────────

Email: sstoacleyg@scribd.com
Name: Samuele Stoacley
Birthday: 1992-07-02
Gift Card Balance: 120.62
Products in Basket: ['storage bin', 'shoe rack', 'laundry basket']
───────────────────

Email: abassindale16@reuters.com
Name: Andreas Bassindale
Birthday: 1988-08-08
Gift Card Balance: 288.7
Products in Basket: ['pencil case', 'cutlery set']
───────────────────

Email: sjannex29@histats.com
Name: Sax Jannex
Birthday: 2000-08-08
Gift Card Balance: 468.38
Products in Basket: ['toothbrush holder']
───────────────────

Email: skeppyc@hhs.gov
Name: Sky Keppy
Birthday: 1978-08-18
Gift Card Balance: 180.4
Products in Basket: ['candle holder', 'makeup bag', 'socks']
───────────────────

Email: oparmer2d@princeton.edu
Name: Orson Parmer
Birthday: 1950-02-11
Gift Card Balance: 185.54
Products in Basket: ['mixing bowl']
───────────────────

Email: nkensit7@bravesites.com
Name: Nat Kensit
Birthday: 1967-08-11
Gift Card Balance: 212.59
Products in Basket: ['t-shirt', 'mixing bowl', 't-shirt']

```
--------------------

Email: bstirtond@quantcast.com
Name: Brandie Stirton
Birthday: 1994-10-05
Gift Card Balance: 442.09
Products in Basket: ['baking sheet', 't-shirt', 'kitchen towel']
--------------------

Email: lbiddlestonel@narod.ru
Name: Linette Biddlestone
Birthday: 1997-01-30
Gift Card Balance: 409.03
Products in Basket: ['storage bin', 'saucepan']
--------------------

Email: jbiggs1e@infoseek.co.jp
Name: Jonell Biggs
Birthday: 1962-01-11
Gift Card Balance: 285.45
Products in Basket: ['pencil case']
--------------------

Email: ctohill2q@blogs.com
Name: Colline Tohill
Birthday: 1977-08-03
Gift Card Balance: 314.22
Products in Basket: ['cutlery set']
--------------------

Email: ppoutress10@nytimes.com
Name: Poul Poutress
Birthday: 2001-01-27
Gift Card Balance: 92.79
Products in Basket: ['candle holder', 't-shirt']
--------------------

Email: oparmer2d@princeton.edu
Name: Orson Parmer
Birthday: 1950-02-11
Gift Card Balance: 385.86
Products in Basket: ['mixing bowl']
--------------------
```

```
Email: tvotier1y@plala.or.jp
Name: Thornton Votier
Birthday: 1981-08-09
Gift Card Balance: 335.01
Products in Basket: ['water bottle']
-------------------

Email: rdadds1f@buzzfeed.com
Name: Ronnie Dadds
Birthday: 1992-06-29
Gift Card Balance: 79.7
Products in Basket: ['hair dryer']
-------------------

Email: jglasscoe22@pinterest.com
Name: Jeanette Glasscoe
Birthday: 1953-11-14
Gift Card Balance: 143.87
Products in Basket: ['air fryer']
-------------------

Email: khemerijk1c@omniture.com
Name: Kathryn Hemerijk
Birthday: 1995-11-12
Gift Card Balance: 142.61
Products in Basket: ['storage bin', 'towel rack']
-------------------

Email: creedersu@weebly.com
Name: Colleen Reeders
Birthday: 1971-10-26
Gift Card Balance: 318.12
Products in Basket: ['cutlery set', 'baking sheet']
-------------------

Email: abourdasv@nydailynews.com
Name: Allys Bourdas
Birthday: 1954-02-22
Gift Card Balance: 106.16
Products in Basket: ['scarf', 'laptop case']
-------------------
```

2e) *Top Products by Category*

To run the following query, I had to first ensure that the Product, Category, and OrderProduct nodes were correctly established with the proper relationships.

Then in the query I matched the relevant nodes and merged with relationships. I calculated the total sales by multiplying all products with their respective qunatities in orders and then using *SUM* to add individual product quantities together. I used *ORDER BY _ DESC* to sort by total sales and *COLLECT* to group the products by category. To extract only the top two products by category, I had to use the *UNWIND* function which flattens the the list of top products into indiviudal rows on which I can use the interval [0..2] to get the top two products per category. Finally, I ordered by category and printed the result, formatting the print function to make it easy to read and understand.

```
In [16]:    # checking product_relations
            product_relations = '''
            LOAD CSV WITH HEADERS FROM 'https://docs.google.com/spreadsheets/d/1KDL71aLV5ay4rZzTFN5u8RplovR3trOl8QzDaGF9FsI/export
            MATCH (p:Product {ProductNumber: toInteger(row.ProductNumber)}), (c:Category {CategoryID: toInteger(row.CategoryID)})
            MERGE (pc:ProductCategory {ProductNumber: toInteger(row.ProductNumber), CategoryID: toInteger(row.CategoryID)})
            MERGE (p)-[:BELONGS_TO]->(c)
            MERGE (p)-[:HAS_CATEGORY]->(pc)
            '''

            # running
            with driver.session(database="neo4j") as session:
                result_2c = session.run(product_relations)
```

```
In [17]:    topcategories = '''
            MATCH (p:Product)-[:BELONGS_TO]->(c:Category)
            MATCH (op:OrderProduct)-[:IS_PRODUCT_IN_ORDER]->(p)
            WITH c.CategoryName AS Category, p.ProductName AS Product, SUM(op.ProductQuantity) AS TotalSales
            ORDER BY TotalSales DESC
            WITH Category, COLLECT({Product: Product, TotalSales: TotalSales})[0..2] AS TopProducts
            UNWIND TopProducts AS TopProduct
            RETURN Category, TopProduct.Product AS Product, TopProduct.TotalSales AS TotalSales
            ORDER BY Category
            '''

            # printing title
            print(' ' * 8, "Top 2 Products in each Category and Total Sales")

            # running query
            with driver.session(database="neo4j") as session:
                result_2c = session.run(topcategories)
                for record in result_2c:
```

```python
        print(f"\nCategory: {record['Category']}\nProduct: {record['Product']}\nTotal Sales: {record['TotalSales']}")
        print('-' * 20)
```

Top 2 Products in each Category and Total Sales

Category: Bathroom
Product: makeup bag
Total Sales: 78
_____

Category: Bathroom
Product: bath mat
Total Sales: 53
_____

Category: Cleaning
Product: broom
Total Sales: 26
_____

Category: Clothing
Product: socks
Total Sales: 106
_____

Category: Clothing
Product: clothes hanger
Total Sales: 86
_____

Category: Electronics
Product: air fryer
Total Sales: 70
_____

Category: Electronics
Product: hair dryer
Total Sales: 65
_____

Category: Home Decor
Product: candle holder
Total Sales: 233
_____

Category: Home Decor
Product: blender

Total Sales: 71

Category: Jewelry
Product: necklace
Total Sales: 101

Category: Jewelry
Product: ring
Total Sales: 100

Category: Kitchen
Product: air fryer
Total Sales: 89

Category: Kitchen
Product: mixing bowl
Total Sales: 80

Category: Personal Care
Product: water bottle
Total Sales: 116

Category: Personal Care
Product: umbrella
Total Sales: 103

Category: Stationery
Product: water bottle
Total Sales: 116

Category: Stationery
Product: scale
Total Sales: 55

Category: Storage

```
Product: mixing bowl
Total Sales: 80
--------------------

Category: Storage
Product: laundry basket
Total Sales: 76
--------------------
```

Closing session

In [19]: `session.close`

Out[19]: `<bound method Session.close of <neo4j._sync.work.session.Session object at 0x168cb4ad0>>`