In [6]:
```
import sqlite3
import pandas as pd
```

In [7]:
```
# Load the data with pandas
df_orders = pd.read_csv('orders.csv')
df_customers = pd.read_csv('customers.csv')
df_line_items = pd.read_csv('line_items.csv')
```

In [8]:
```
# Make the connection to sqlite3 and make a db
con = sqlite3.connect("interview.db")
cur = con.cursor()
```

In [9]:
```
# # drop data into database into three tables
df_orders.to_sql("orders", con)
df_customers.to_sql("customers", con)
df_line_items.to_sql("line_items", con)
```

1. How many orders were completed in 2018?

In [10]:
```
cur.execute("""SELECT COUNT(*) AS num_order
                    FROM orders
                    WHERE order_timestamp >= '2018-01-01'
                            AND order_timestamp <= '2018-12-31';""")
print('The number of orders in 2018 is ', cur.fetchall()[0][0])
```

```
The number of orders in 2018 is  9219
```

2. How many orders were completed in 2018 containing at least 10 units?

In [11]:
```
'''There are multiple lines of items for each order_id
that is why we grouped by order_id in line_items table and then sum all the quanitities.'''

cur.execute("""SELECT COUNT(*) AS num_order
                    FROM orders AS o
                        JOIN (SELECT order_id, sum(quantity) AS total_quantity
                                    FROM line_items
                                    GROUP BY order_id) AS gl
                            ON o.order_id = gl.order_id
                    WHERE o.order_timestamp >= '2018-01-01'
                            AND o.order_timestamp <= '2018-12-31'
                            AND gl.total_quantity >= 10;""")

print('The number of orders in 2018 with at least 10 units is ', cur.fetchall()[0][0])
```

```
The number of orders in 2018 with at least 10 units is  5147
```

3. How many customers have ever purchased a medium sized sweater with a discount?

In [12]:
```
'''
According to the question the conditions are customers with
                                                        product_category = 'Sweater'
                                                        size = 'M'
                                                        quantity = 1
                                                        discount > 0
'''
cur.execute("""SELECT COUNT(DISTINCT o.customer_uid) AS num_customers
                    FROM orders AS o
                    JOIN line_items AS l ON o.order_id = l.order_id
                    WHERE l.product_category = 'Sweater'
                            AND l.size = 'M'
                            AND l.quantity = 1
                            AND 1.0*o.discount > 0;""")

print('Number of customers that have purchased a medium sized sweater with a discount is ', cur.fetchall()[0][0])
```

```
Number of customers that have purchased a medium sized sweater with a discount is  528
```

4. How profitable was our most profitable month?

In [13]:
```
'''
profit = ((quantity*selling_price)*(1-discount)*(1-returned)) +
            shipping_revenue -
            (quantity*supplier_cost) -
            (shipping_cost)
'''
cur.execute("""SELECT ROUND(SUM(profit), 2) as total_month_profit
            FROM(SELECT o.order_id, ((l.total_sale*(1-o.discount)*(1-o.returned)) +
                                o.shipping_revenue - l.total_supplier_cost - (shipping_cost)) AS profit,
            strftime('%Y', order_timestamp) AS year, strftime('%m', order_timestamp) AS month
            FROM orders AS o
            JOIN (SELECT order_id, SUM(quantity*selling_price) AS total_sale,
                                    SUM(quantity*supplier_cost) AS total_supplier_cost
                    FROM line_items
                    WHERE (selling_price IS NOT NULL) AND (supplier_cost IS NOT NULL)
                    GROUP BY order_id)
             AS l ON o.order_id = l.order_id)
            GROUP BY year, month
            ORDER BY total_month_profit DESC
            LIMIT 1;""")
print('The profit of the most profitable month is ', cur.fetchall()[0][0])
```

```
The profit of the most profitable month is  55714.25
```

5. What is the return rate for business vs. non-business customers?

In [19]:

```
cur.execute("""SELECT c.is_business, ROUND(1.0*AVG(o.returned), 3) AS rate_of_return
            FROM customers AS c
            JOIN orders AS o ON c.customer_uid = o. customer_uid
            GROUP BY c.is_business;""")

rows = cur.fetchall()
print(f'Rate of return for business customers is {rows[1][1]} and for non-business customers is {rows[0][1]}')
```

```
Rate of return for business customers is 0.067 and for non-business customers is 0.049
```

In [ ]: