DNSC 6305 Data Management Final Project

State Agency Amazon Spending, A Washington State Case

Ask 1: Project and Dataset Overview

Overview

The Department of Enterprise Services (DES) is proactively sharing in-depth insights into the Amazon expenditures of state agencies, sourced from the Washington State Amazon Business account. This dataset exclusively contains completed orders, excluding any that are currently in progress or have been canceled. It encompasses order data from the period spanning 2017 to 2022, offering a comprehensive snapshot of financial transactions during this five-year timeframe. With an extensive compilation comprising more than 486,000 records, this dataset offers a detailed and intricate view of the diverse purchases conducted by various departments throughout Washington State.

Data Source

We have retrieved the dataset from Data.Gov and it was lasted updated on December 2, 2022. We have provided a link to the dataset below:

- https://catalog.data.gov/dataset/state-agency-amazon-spend

Significance and Relevance

Strategic Insights for Amazon: This dataset serves as a valuable resource for Amazon. It offers insights into trends and seasonality within orders, aiding in the optimization of inventory and marketing strategies. Additionally, it provides valuable information about popular product categories, allowing for adjustments in product offerings to align with consumer preferences. Furthermore, it helps identify key competitors, enabling Amazon to tailor its sales strategies effectively and maintain a competitive edge in the market.

Dimensional Modeling Capability

We performed an initial exploration and have determined that this dataset is well-suited for dimensional modeling and in-depth analytical analysis. It encompasses a collection of over 486,000 rows of transactional data, featuring numerical fields tailored for the fact table and four distinct sets of categorical data optimized for the dimensional table.

Data Analysis Objectives

1. **What are the notable time series trends that characterize purchases from Amazon by Washington Agencies over the years?**

   - In which months do Washington Agencies exhibit higher spending patterns on Amazon purchases?
   - Over each year, which Washington Agencies demonstrate the highest expenditures on Amazon orders? What are the trends for the top spending agencies (is spending increasing, decreasing, etc.)?
   - How do the Net Purchase Totals from all Washington Agencies fluctuate yearly, and what factors contribute to these trends?

   ### Business Value:

   These analyses aim to maximize revenue by leveraging seasonal trends:

   a. **Optimizing Advertising Strategies:** Understanding seasonal spending trends helps Amazon tailor advertising campaigns ahead of peak purchasing periods. By identifying high-expenditure months, targeted campaigns can be deployed to capitalize on increased purchasing behavior.

   b. **Stimulating Higher Spending:** Offering discounts and specialized pricing plans during low-order months can encourage greater spending from agencies that may otherwise be less active.

   c. **Efficient Marketing Alignment:** Aligning marketing efforts with agency spending patterns ensures that Amazon maximizes revenue potential throughout the year. This approach optimizes the allocation of marketing resources based on historical data and trends.

2. **What trends emerge in recurring orders, bulk orders, and customer attrition among the individual Washington Agencies that spend the most on Amazon purchases?**

- Which Washington Agencies have recorded the highest order volumes on Amazon during the years?
- Among Washington Agencies, which ones are placing Bulk Orders (orders with item quantities exceeding 100) on Amazon?
- Which Washington Agencies previously ordered from Amazon but have stopped?

Business Value:

These analyses provide valuable insights and opportunities for Amazon:

a. **Streamlining Inventory and Predicting Bulk Orders:** By identifying agencies with high order volumes, Amazon can optimize inventory management and predict bulk orders more accurately. This ensures consistent availability of inventory to meet customer demands.

b. **Promotions for Bulk Orders:** Recognizing agencies that place bulk orders allows Amazon to create targeted discounts and promotions for such orders. This consistent availability of inventory and promotions enhances Amazon's reputation as a preferred vendor and improves customer satisfaction.

c. **Re-engaging Attrited Agencies:** Identifying agencies that have stopped purchasing from Amazon allows the company to calculate the potential net loss incurred. Amazon can then develop strategies to re-engage these agencies by offering tailored discounts and promotions, potentially recovering lost revenue and expanding its customer base.

3. **Who are the primary competitor sellers to Amazon's own products on Amazon.com, and what are their sales offerings, magnitudes of sales, and competitive product categories?**

- Identify the top 5 competitor sellers that Agencies are sourcing from on Amazon.com and quantify their net totals. How have they changed over the past few years?
- Among the top 5 competitor sellers on Amazon.com, what product categories do they predominantly sell? Are they offering discounts that are giving them an advantage in this process?
- What are the agencies that the top 5 competitor sellers have the most business with? That all competitors in the data have the most business with?

Business Value:

These analyses offer strategic advantages for Amazon:

a. **Competitor Threat Assessment:** Identifying top competitor sellers and quantifying their net totals provides Amazon with insights into the biggest competitive threats. It also allows for a comparison of competitor revenue per product relative to Amazon's revenue per product.

b. **Targeted Promotions and Discounts:** Knowledge of competitor product categories can help Amazon pinpoint specific product lines to offer promotions and discounts. This strategy can increase revenue and market share for Amazon's own products in those categories.

c. **Market Share Expansion:** Understanding which agencies engage most with competitor sellers enables Amazon to target promotions and discounts effectively. This approach can help Amazon gain market share with these agencies and potentially reduce competition's influence on their purchases.

Overcoming Challenges in Data Analysis

Dealing with a substantial dataset naturally presented its own set of challenges, and we addressed them proactively:

1. **Handling Missing Values:**

   - In our dataset, some columns had missing values, either completely or partially. For columns where all values were absent, like "Payment Amount," we opted to remove these columns. In cases where at least half of the values were missing, we carefully considered their relevance to our data analysis.
   - It's noteworthy that a significant part of our analysis hinges on the relationship between Agency Names and orders. Roughly 70,000 transactions were associated with an agency name, while the remaining 410,000 lacked this information. Consequently, our analysis relies on this subset of the dataset when dealing with Agency orders.
2. **Causation Vs. Correlation:**

   - We adhere to a critical principle that distinguishes between causation and correlation in our analysis. We refrain from making unwarranted assumptions about causation without conducting in-depth analyses and gaining a comprehensive understanding of the underlying factors.
   - Recognizing and mitigating misinterpretations in these relationships is of paramount importance, as erroneous conclusions could potentially lead to misguided decisions.
3. **Dealing with Duplicate Values:**

- During our analysis of seller names, we encountered instances where different text representations referred to the same entity. For instance, "Amazon" and "Amazon.com" denoted identical entities but were represented differently across various years. In these cases, we made the decision to consolidate these values into a single representation to accurately reflect one seller entity.

Ask 2: Data Wrangling and Dimensional Modeling

Setup - Agency Amazon Spend

```
In [1]: %load_ext sql
```

```
In [2]: !dropdb -U student Final_Project
```

```
In [3]: !createdb -U student Final_Project
```

```
In [4]: %sql postgresql://student@/Final_Project
```

Loading the data directly from Data.Gov

```
In [5]: !wget -O amzn_spend.csv https://data.wa.gov/api/views/agvw-ch2s/rows.csv
```

```
--2023-12-10 16:48:30--  https://data.wa.gov/api/views/agvw-ch2s/rows.csv
Resolving data.wa.gov (data.wa.gov)... 52.206.140.199, 52.206.68.26, 52.206.140.205
Connecting to data.wa.gov (data.wa.gov)|52.206.140.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘amzn_spend.csv’

amzn_spend.csv          [                <=> ] 102.76M  3.71MB/s    in 25s

2023-12-10 16:48:56 (4.05 MB/s) - ‘amzn_spend.csv’ saved [107757551]
```

After loading the data, we need to make sure that all the observations and column head have been downloaded successfully.

In [6]:
```
!wc -l amzn_spend.csv
```

```
486274 amzn_spend.csv
```

Show the columns of the Amazon table for reference

In [7]:
```
!csvcut -n amzn_spend.csv
```

```
  1: Order Date
  2: Agency Name
  3: Payment Date
  4: Payment Amount
  5: Shipment Date
  6: Product Category
  7: ASIN
  8: Product Title
  9: UNSPSC
 10: Brand Code
 11: Brand
 12: Manufacturer
 13: Item model number
 14: Part number
 15: Product Condition
 16: Listed PPU
 17: Purchase PPU
 18: Item Quantity
 19: Item Subtotal
 20: Item Shipping & Handling
 21: Item Promotion
 22: Item Tax
 23: Item Net Total
 24: Discount Program
 25: Pricing Discount applied ($ off)
 26: Pricing Discount applied (% off)
 27: Seller Name
```

Table Creation

In [8]:
```
!head -n 10000 amzn_spend.csv | csvstat
```

```
/home/ubuntu/.local/lib/python3.8/site-packages/agate/table/from_csv.py:74: RuntimeWarning: Error sniffing CS
V dialect: Could not determine delimiter
  1. "Order Date"

        Type of data:          DateTime
        Contains null values:  False
        Unique values:         81
        Smallest value:        2017-10-09 00:00:00
        Largest value:         2022-06-30 00:00:00
        Most common values:    2022-06-15 00:00:00 (773x)
                               2022-06-09 00:00:00 (736x)
                               2022-06-16 00:00:00 (601x)
                               2022-06-13 00:00:00 (598x)
                               2022-06-23 00:00:00 (596x)

  2. "Agency Name"

        Type of data:          Text
        Contains null values:  True (excluded from calculations)
        Unique values:         59
        Longest value:         64 characters
        Most common values:    CHILDREN YOUTH AND FAMILIES DEPT OF (2360x)
                               SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS) (1914x)
                               CORRECTIONS DEPARTMENT OF (1129x)
                               FISH AND WILDLIFE DEPARTMENT OF (594x)
                               PARKS AND RECREATION COMMISSION WASHINGTON STATE (427x)

  3. "Payment Date"

        Type of data:          DateTime
        Contains null values:  True (excluded from calculations)
        Unique values:         153
        Smallest value:        2017-10-09 00:00:00
        Largest value:         2022-07-25 00:00:00
        Most common values:    2022-06-16 00:00:00 (595x)
                               2022-06-20 00:00:00 (575x)
                               2022-06-14 00:00:00 (524x)
                               2022-06-15 00:00:00 (508x)
                               2022-06-22 00:00:00 (456x)

  4. "Payment Amount"

        Type of data:          Boolean
```

```
      Contains null values:  True (excluded from calculations)
      Unique values:         1
      Most common values:    None (9999x)

5. "Shipment Date"

      Type of data:          DateTime
      Contains null values:  True (excluded from calculations)
      Unique values:         156
      Smallest value:        2017-10-09 00:00:00
      Largest value:         2022-07-25 00:00:00
      Most common values:    2022-06-16 00:00:00 (655x)
                             2022-06-14 00:00:00 (518x)
                             2022-06-15 00:00:00 (477x)
                             2022-06-22 00:00:00 (449x)
                             2022-06-20 00:00:00 (411x)

6. "Product Category"

      Type of data:          Text
      Contains null values:  False
      Unique values:         55
      Longest value:         49 characters
      Most common values:    Office Product (1046x)
                             Home Improvement (692x)
                             Health and Beauty (631x)
                             Book (607x)
                             Kitchen (604x)

7. "ASIN"

      Type of data:          Text
      Contains null values:  False
      Unique values:         7320
      Longest value:         10 characters
      Most common values:    B086Q92563 (46x)
                             B01HR87JKG (43x)
                             B08LVBD1YB (28x)
                             B014W20C90 (27x)
                             B07S4LDHJM (26x)

8. "Product Title"
```

```
        Type of data:          Text
        Contains null values:  False
        Unique values:         55
        Longest value:         49 characters
        Most common values:    Office Product (1046x)
                               Home Improvement (692x)
                               Health and Beauty (631x)
                               Book (607x)
                               Kitchen (604x)

  9. "UNSPSC"

        Type of data:          Number
        Contains null values:  True (excluded from calculations)
        Unique values:         846
        Smallest value:        10000000
        Largest value:         95000000
        Sum:                   459938086957
        Mean:                  46327365.729
        Median:                47130000
        StDev:                 10595585.529
        Most common values:    55101500 (600x)
                               53100000 (320x)
                               53131600 (217x)
                               43191601 (202x)
                               55111514 (163x)

 10. "Brand Code"

        Type of data:          Text
        Contains null values:  True (excluded from calculations)
        Unique values:         1888
        Longest value:         5 characters
        Most common values:    None (4923x)
                               LOGAM (127x)
                               GRAR9 (105x)
                               AMZSM (96x)
                               MICT9 (66x)

 11. "Brand"

        Type of data:          Text
        Contains null values:  True (excluded from calculations)
```

```
       Unique values:         4288
       Longest value:         49 characters
       Most common values:    None (354x)
                              Logitech (150x)
                              Amazon Basics (144x)
                              Microsoft (130x)
                              Amazon (94x)

12. "Manufacturer"

       Type of data:          Text
       Contains null values:  True (excluded from calculations)
       Unique values:         4389
       Longest value:         97 characters
       Most common values:    None (686x)
                              Logitech (133x)
                              Amazon Basics (131x)
                              Microsoft (129x)
                              Graco (74x)

13. "Item model number"

       Type of data:          Text
       Contains null values:  True (excluded from calculations)
       Unique values:         4438
       Longest value:         40 characters
       Most common values:    None (3441x)
                              307_US_Email (65x)
                              SVS-00001 (46x)
                              CF101-L-A (43x)
                              1 (31x)

14. "Part number"

       Type of data:          Text
       Contains null values:  True (excluded from calculations)
       Unique values:         5660
       Longest value:         44 characters
       Most common values:    None (1844x)
                              1 (65x)
                              SVS-00001 (46x)
                              CF101-L-A (43x)
                              unknown (33x)
```

```
15. "Product Condition"

       Type of data:          Text
       Contains null values:  False
       Unique values:         6
       Longest value:         24 characters
       Most common values:    New (9947x)
                              Used - Very good (25x)
                              Used - Like new (12x)
                              Used - Good (11x)
                              Used - Acceptable (3x)

16. "Listed PPU"

       Type of data:          Number
       Contains null values:  False
       Unique values:         2049
       Smallest value:        0
       Largest value:         4999
       Sum:                   735772.63
       Mean:                  73.585
       Median:                31.99
       StDev:                 168.338
       Most common values:    19.99 (292x)
                              39.99 (288x)
                              29.99 (257x)
                              49.99 (194x)
                              14.99 (189x)

17. "Purchase PPU"

       Type of data:          Number
       Contains null values:  False
       Unique values:         2948
       Smallest value:        0
       Largest value:         4999
       Sum:                   603887.37
       Mean:                  60.395
       Median:                25.98
       StDev:                 133.408
       Most common values:    19.99 (147x)
                              9.99 (133x)
                              14.99 (124x)
                              39.99 (120x)
                              29.99 (116x)

18. "Item Quantity"

       Type of data:          Number
       Contains null values:  False
       Unique values:         53
       Smallest value:        0
       Largest value:         200
       Sum:                   24037
       Mean:                  2.404
       Median:                1
       StDev:                 5.036
       Most common values:    1 (6642x)
                              2 (1405x)
                              3 (507x)
                              4 (445x)
                              5 (237x)

19. "Item Subtotal"

       Type of data:          Number
       Contains null values:  True (excluded from calculations)
       Unique values:         3925
       Smallest value:        1.88
       Largest value:         9998
       Sum:                   1083669.01
       Mean:                  108.399
       Median:                41.49
       StDev:                 276.302
       Most common values:    19.99 (98x)
                              9.99 (94x)
                              14.99 (88x)
                              39.99 (88x)
                              29.99 (84x)

20. "Item Shipping & Handling"

       Type of data:          Number
       Contains null values:  True (excluded from calculations)
       Unique values:         132
```

```
        Smallest value:        0
        Largest value:         839.88
        Sum:                   5544.3
        Mean:                  0.56
        Median:                0
        StDev:                 10.5
        Most common values:    0 (9675x)
                               None (94x)
                               3.99 (16x)
                               2.99 (11x)
                               4.99 (10x)

 21. "Item Promotion"

        Type of data:          Number
        Contains null values:  True (excluded from calculations)
        Unique values:         284
        Smallest value:        -208.8
        Largest value:         0
        Sum:                   -5313.78
        Mean:                  -9.768
        Median:                -2.76
        StDev:                 19.914
        Most common values:    None (9455x)
                               0 (16x)
                               -0.75 (13x)
                               -2.99 (12x)
                               -10 (11x)

 22. "Item Tax"

        Type of data:          Number
        Contains null values:  True (excluded from calculations)
        Unique values:         2023
        Smallest value:        0
        Largest value:         946.88
        Sum:                   95344.62
        Mean:                  9.536
        Median:                3.6
        StDev:                 23.881
        Most common values:    0 (363x)
                               1.6 (61x)
                               2 (54x)
```

```
                              1.32 (46x)
                              1 (44x)

23. "Item Net Total"

       Type of data:          Number
       Contains null values:  True (excluded from calculations)
       Unique values:         5634
       Smallest value:        0
       Largest value:         11019.88
       Sum:                   1179244.15
       Mean:                  117.948
       Median:                45.1
       StDev:                 299.378
       Most common values:    1365.24 (39x)
                              25 (28x)
                              303.44 (25x)
                              15.31 (24x)
                              9.84 (21x)

24. "Discount Program"

       Type of data:          Text
       Contains null values:  True (excluded from calculations)
       Unique values:         3
       Longest value:         30 characters
       Most common values:    None (7389x)
                              Quantity pricing (2609x)
                              Every Day Low Business Pricing (1x)

25. "Pricing Discount applied ($ off)"

       Type of data:          Number
       Contains null values:  True (excluded from calculations)
       Unique values:         821
       Smallest value:        0.01
       Largest value:         1305
       Sum:                   41612.64
       Mean:                  23.391
       Median:                7.8
       StDev:                 66.529
       Most common values:    None (8220x)
                              10 (28x)
                              2 (25x)
                              7 (25x)
                              10.33 (25x)

 26. "Pricing Discount applied (% off)"

        Type of data:          Number
        Contains null values:  True (excluded from calculations)
        Unique values:         1112
        Smallest value:        0
        Largest value:         82.84
        Sum:                   1943.472
        Mean:                  1.092
        Median:                0.224
        StDev:                 5.662
        Most common values:    None (8220x)
                               0.129 (25x)
                               0.42 (21x)
                               0.15 (20x)
                               0.11 (19x)

 27. "Seller Name"

        Type of data:          Text
        Contains null values:  False
        Unique values:         3650
        Longest value:         50 characters
        Most common values:    Amazon.com (2873x)
                               Amazon (1337x)
                               Tracfone Wireless Inc. (57x)
                               Happydeal-US (43x)
                               AnkerDirect (33x)

Row count: 9999
```

In [9]:

```
%%sql
DROP TABLE IF EXISTS amazon;

CREATE TABLE amazon (
    order_date DATE,
    agency_name VARCHAR,
    payment_date DATE,
    payment_amount NUMERIC,
    shipment_date DATE,
    product_category VARCHAR,
    asin VARCHAR,
    product_title VARCHAR,
    unspsc VARCHAR,
    brand_code VARCHAR,
    brand VARCHAR,
    manufacturer VARCHAR,
    item_model_number VARCHAR,
    part_number VARCHAR,
    product_condition VARCHAR,
    listed_ppu NUMERIC,
    purchase_ppu NUMERIC,
    item_quantity NUMERIC,
    item_subtotal NUMERIC,
    item_shipping_handling NUMERIC,
    item_promotion NUMERIC,
    item_tax NUMERIC,
    item_net_total NUMERIC,
    discount_program VARCHAR,
    pricing_discount_dollar NUMERIC,
    pricing_discount_percentage VARCHAR,
    seller_name VARCHAR
)
```

```
 * postgresql://student@/Final_Project
Done.
Done.
```

Out[9]: []

In [10]:

```
%%sql

COMMENT ON TABLE amazon IS
'The Amazon table contains all the order transactions from the Washington State Agencies on Amazon from 2017-
'Next, we are going to decribe all the columns of the table that we have used in the analysis.';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[10]: []

In [11]:

```
%%sql

COMMENT ON COLUMN amazon.order_date IS 'Order Date: The detailed date and time an order was placed on Amazon'
COMMENT ON COLUMN amazon.agency_name IS 'Agency Name: The State Agency making the order';
COMMENT ON COLUMN amazon.asin IS 'ASIN: A unique product identifier on Amazon.com';
COMMENT ON COLUMN amazon.brand IS 'Brand: The product brand';
COMMENT ON COLUMN amazon.manufacturer IS 'Manufacturer: The product manufacturer';
COMMENT ON COLUMN amazon.listed_ppu IS 'Listed Price per Unit: The listed price per unit for each item ordere
COMMENT ON COLUMN amazon.purchase_ppu IS 'Purchase Price per Unit: The purchase price per unit for each item
COMMENT ON COLUMN amazon.item_quantity IS 'Item Quantity: The number of items per order';
COMMENT ON COLUMN amazon.item_subtotal IS 'Item Subtotal: The item price without tax added';
COMMENT ON COLUMN amazon.item_shipping_handling IS 'Item Shiping and Handling: The shipping and handling fee'
COMMENT ON COLUMN amazon.item_tax IS 'Item Tax: The item tax';
COMMENT ON COLUMN amazon.item_net_total IS 'Item Net Total: The item price including the tax';
COMMENT ON COLUMN amazon.pricing_discount_dollar IS 'Pricing Discount: The dollar amount discounted from the
COMMENT ON COLUMN amazon.seller_name IS 'Seller Name: The name of the seller that sold the item on Amazon.com
```

```
 * postgresql://student@/Final_Project
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
Done.
```

Out[11]:

```
[]
```

In [12]:

```
!pwd
```

```
/home/ubuntu/notebooks/Final Project
```

After confirming the working directory, we load the data to the amazon table we created

In [13]:

```
%%sql
COPY amazon FROM '/home/ubuntu/notebooks/Final Project/amzn_spend.csv'
CSV
HEADER;
```

```
 * postgresql://student@/Final_Project
486273 rows affected.
```

```
Out[13]: []
```

Confirm that the same amount of rows transfered by running a count SQL query

```
In [14]: %%sql

SELECT COUNT(*) from amazon
```

```
 * postgresql://student@/Final_Project
1 rows affected.
```

Out[14]:

count
486273

```
In [15]: %%sql

COMMENT ON TABLE amazon IS
'We have confirmed that all the observations have been transfered successfully';
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[15]: []
```

```
In [16]: %%sql

SELECT * from amazon
LIMIT 5
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[16]:

	order_date	agency_name	payment_date	payment_amount	shipment_date	product_category	asin	product_title	unspsc
	2022-06-30	NATURAL RESOURCES DEPARTMENT OF	2022-07-01	None	2022-07-01	Personal Computer	B07TSQFD7X	Personal Computer	43211900
	2022-06-30	NATURAL RESOURCES DEPARTMENT OF	2022-07-03	None	2022-07-03	CE	B095YBDCP2	CE	43211609
	2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	2022-07-01	None	2022-07-01	Kitchen	B09TR685YD	Kitchen	52120000
	2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	2022-07-01	None	2022-07-01	Kitchen	B097P2H4C6	Kitchen	49101609
	2022-06-30	FISH AND WILDLIFE DEPARTMENT	2022-07-01	None	2022-07-01	Apparel	B07CRL2KC9	Apparel	53100000

We decided to drop the following columns for the attached reasons

1. PAYMENT AMOUNT - NO VALUES
2. SHIPMENT DATE - NOT RELEVANT TO OUR ANALYSIS
3. UNSPSC - UNSURE WHAT IT REFERS TO
4. BRAND CODE - NOT RELEVANT TO OUR ANALYSIS, TOO MANY EMPTY CELLS
5. ITEM MODEL NUMBER - NOT RELEVANT TO OUR ANALYSIS
6. PART NUMBER - NOT RELEVANT TO OUR ANALYSIS
7. PRODUCT CONDITION - NOT RELEVANT TO OUR ANALYSIS, ALMOST ALL ARE 'NEW'
8. PRODUCT TITLE - SAME AS PRODUCT CATEGORY
9. PRICING DISCOUNT APPLIED (% off) - NOT RELEVANT TO OUR ANALYSIS, USING PRICING DISCOUNT APPLIED ($ OFF)
10. PAYMENT DATE - NOT RELEVANT TO OUR ANALYSIS
11. DISCOUNT PROGRAM - NOT RELEVANT TO OUR ANALYSIS
12. ITEM PROMOTION - NOT RELEVANT TO OUR ANALYSIS

```
In [17]: %%sql

         COMMENT ON TABLE amazon IS
         'Drop the following columns for the attached reasons, PAYMENT AMOUNT - NO VALUES, SHIPMENT DATE - NOT RELEVAN
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[17]: []
```

```
In [18]: %%sql

         ALTER TABLE amazon
             DROP COLUMN payment_amount,
             DROP COLUMN payment_date,
             DROP COLUMN shipment_date,
             DROP COLUMN unspsc,
             DROP COLUMN brand_code,
             DROP COLUMN item_model_number,
             DROP COLUMN product_title,
             DROP COLUMN part_number,
             DROP COLUMN product_condition,
             DROP COLUMN pricing_discount_percentage,
             DROP COLUMN discount_program,
             DROP COLUMN item_promotion;
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[18]: []
```

Show the table to confirm the columns were dropped

```
In [19]: %%sql

         SELECT * from amazon
         LIMIT 5
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[19]:

order_date	agency_name	product_category	asin	brand	manufacturer	listed_ppu	purchase_ppu	item_quantity	item
2022-06-30	NATURAL RESOURCES DEPARTMENT OF	Personal Computer	B07TSQFD7X	ASUS	Asus	319	297.69	1	
2022-06-30	NATURAL RESOURCES DEPARTMENT OF	CE	B095YBDCP2	LOBKIN	LOBKIN	9.99	7.29	2	
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	Kitchen	B09TR685YD	ADXCO	ADXCO	22.99	22.99	1	
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	Kitchen	B097P2H4C6	OuMuaMua	Hawaiian Luau Party Porch Sign	10.99	10.99	2	
2022-06-30	FISH AND WILDLIFE DEPARTMENT	Apparel	B07CRL2KC9	Hajoyful	Hajoyful	24.98	24.98	1	

In [20]:

```
%%sql

COMMENT ON TABLE amazon IS
'Show the table to confirm the columns were dropped';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[20]: []

An additional observation in the original data was that for some purchases where Amazon was the vendor, the seller name was listed as just "Amazon", and in other cases, it was listed as "Amazon.com". The following query seeks to eliminate this discepency by rewriting all of the seller names as "Amazon.com" for purchases where Amazon was the vendor.

In [21]:

```
%%sql

UPDATE amazon
```

```
SET seller_name = 'Amazon.com'
WHERE seller_name IN ('Amazon', 'Amazon.com');
```

```
 * postgresql://student@/Final_Project
214915 rows affected.
```

Out[21]: []

In [22]:

```
%%sql

COMMENT ON TABLE amazon IS
'An additional observation in the original data was that for some purchases where Amazon was the vendor, the
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[22]: []

We also discovered that the company, "VARIDESK", had it's seller_name switched to "Vari." in this data file during the year 2021. Therefore, this code is used to clean the data so that the seller name is consistent across all records. This will give a more accurate display of the net totals across the five years of records in the file.

- After careful observation, none of the other top 5 competitors have this issue in the data set, and the analysis will be carried out accordingly

In [23]:

```
%%sql

UPDATE amazon
SET seller_name = 'VARIDESK'
WHERE seller_name IN ('VARIDESK', 'Vari.');
```

```
 * postgresql://student@/Final_Project
490 rows affected.
```

Out[23]: []

In [24]:

```
%%sql

COMMENT ON TABLE amazon IS
'We also discovered that the company, "VARIDESK", had its seller_name switched to "Vari." in this data file d
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[24]: []
```

```
In [25]: %%sql

         SELECT COUNT(*) from amazon;
```

```
 * postgresql://student@/Final_Project
1 rows affected.
```

```
Out[25]:   count
          486273
```

```
In [26]: %%sql

         COMMENT ON TABLE amazon IS
         'After adjusting the sellers names, we confirm that the same amount of observations remain';
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[26]: []
```

Item_net_total only includes item_subtotal - (item_tax + item_shipping_handling). Creating new pricing_after_discount column to include pricing_discount_dollar for purchases that had a discount applied to the order.

```
In [27]: %%sql
         UPDATE amazon
         SET pricing_discount_dollar = COALESCE(pricing_discount_dollar, 0.0);
```

```
 * postgresql://student@/Final_Project
486273 rows affected.
```

```
Out[27]: []
```

```
In [28]: %%sql

         COMMENT ON TABLE amazon IS
         'Item_net_total only includes item_subtotal - (item_tax + item_shipping_handling). Creating new pricing_after
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[28]: []
```

In [29]:

```
%%sql
ALTER TABLE amazon
DROP COLUMN IF EXISTS pricing_after_discount;

ALTER TABLE amazon
ADD COLUMN pricing_after_discount NUMERIC;

UPDATE amazon
SET pricing_after_discount = (item_net_total - pricing_discount_dollar);
```

```
 * postgresql://student@/Final_Project
Done.
Done.
486273 rows affected.
```

Out[29]:

```
[]
```

Checking new pricing_after_discount column is consistent with item_net_total and pricing_discount_dollar

In [30]:

```
%%sql

SELECT item_net_total, pricing_discount_dollar, pricing_after_discount from amazon
LIMIT 5
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[30]:

item_net_total	pricing_discount_dollar	pricing_after_discount
15.86	0.0	15.86
24.85	0.0	24.85
23.76	0.0	23.76
26.98	0.0	26.98
12.95	0.0	12.95

In [31]:

```
%%sql

COMMENT ON TABLE amazon IS
'Checking new pricing_after_discount column is consistent with item_net_total and pricing_discount_dollar';
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[31]: []
```

```
In [32]: !pwd
```

```
/home/ubuntu/notebooks/Final Project
```

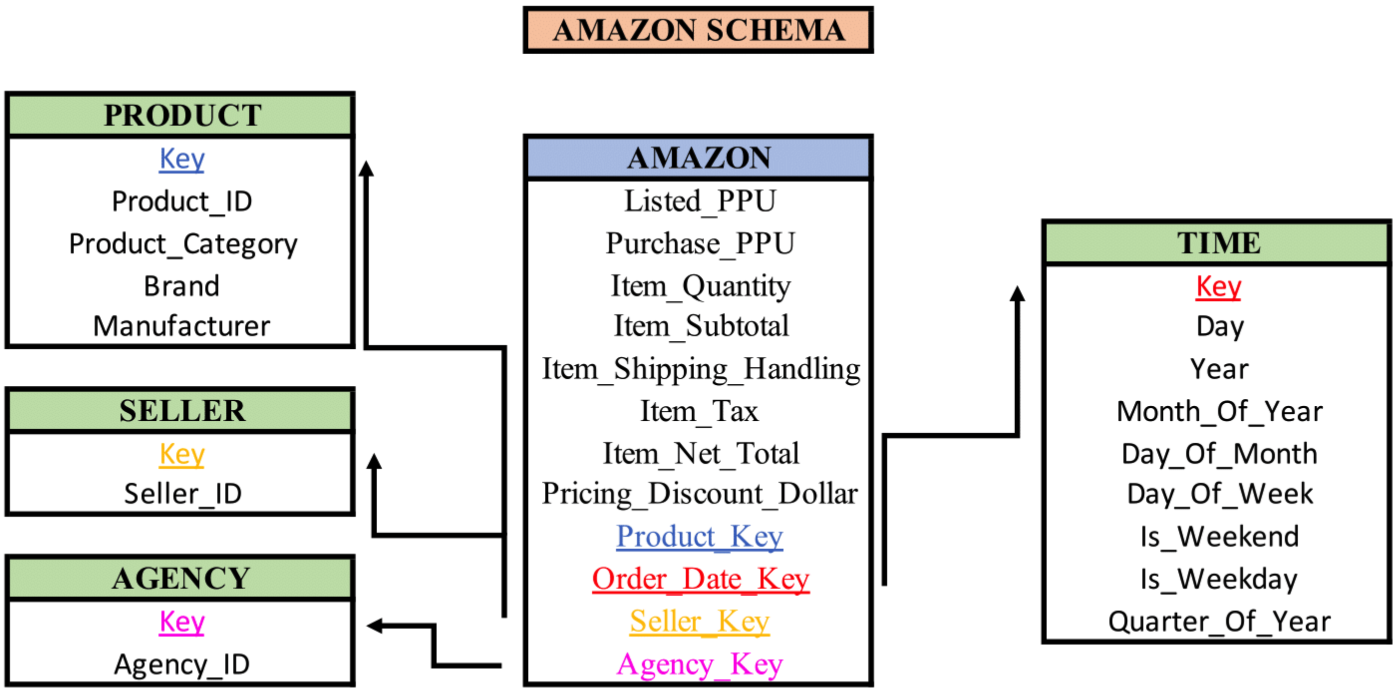
Now that the data wrangling is complete, we will create the dimensional table for the modeling

We have uploaded the Amazon Schema below to reference the Dimensional tables we will be building next.

- Note: The schema file must be in your working directory for the image to display. We have inlcuded the file in the project submission folder.

```
In [33]: from IPython.display import Image
         Image(filename='/home/ubuntu/notebooks/Final Project/Schema.png')
```

Out[33]:



Product Dimension Table

In [34]:

```
%%sql
DROP TABLE IF EXISTS product CASCADE;

CREATE TABLE product (
    key SERIAL PRIMARY KEY,
    Product_ID VARCHAR(15) UNIQUE,
    Product_Category VARCHAR(100),
    Brand VARCHAR(100),
    Manufacturer VARCHAR(500)
);
```

```
 * postgresql://student@/Final_Project
Done.
Done.
```

Out[34]: []

In [35]:

```
%%sql

COMMENT ON TABLE product IS
'The product dimensional table includes the key, the unique product ID, the category name, the brand name, an
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[35]: []

In [36]:

```
%%sql
INSERT INTO product (Product_ID)
SELECT DISTINCT ASIN AS Product_ID
FROM amazon;
```

```
 * postgresql://student@/Final_Project
226521 rows affected.
```

Out[36]: []

In [37]:

```
%%sql
UPDATE product
SET Product_Category = amazon.Product_Category,
    Brand = amazon.Brand,
    Manufacturer = amazon.Manufacturer
FROM amazon
WHERE product.Product_ID = amazon.ASIN;
```

```
 * postgresql://student@/Final_Project
226521 rows affected.
```

Out[37]: []

In [38]:

```
%%sql
SELECT * FROM product
ORDER BY key
LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[38]:

key	product_id	product_category	brand	manufacturer
1	B00PXN0BK8	Home	Bigso	Bigso
2	B076D365RV	CE	Hermitshell	Hermitshell
3	B06VY1DFQN	Baby Product	Evenflo	Evenflo
4	B01JSHFS3Y	Office Product	TUL	OFFICE DEPOT
5	B00G3Q5O7O	Lawn & Patio	Parafilm	Parafilm

In [39]:

```
%%sql

COMMENT ON TABLE product IS
'We can see that the data has transefered successfully. We now create the product foreign key';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[39]: `[]`

Creating the product foreign key to put into our amazon fact table

In [40]:

```
%%sql
ALTER TABLE amazon
ADD COLUMN product_key INTEGER,
ADD CONSTRAINT fk_product
    FOREIGN KEY (product_key)
    REFERENCES product (key);
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[40]: `[]`

In [41]:

```
%%sql
UPDATE amazon
SET product_key = product.key
FROM product
WHERE amazon.ASIN = product.product_id;
```

```
 * postgresql://student@/Final_Project
486273 rows affected.
```

Out[41]: []

Dropping columns from fact table represented by product key

In [42]:

```
%%sql

ALTER TABLE amazon
DROP COLUMN ASIN,
DROP COLUMN Product_Category,
DROP COLUMN Brand,
DROP COLUMN Manufacturer;
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[42]: []

In [43]:

```
%%sql

COMMENT ON TABLE product IS
'After creating the foreign key, we drop the columns from the fact table represented by the product key';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[43]: []

Checking product key populated into amazon fact table

In [44]:

```
%%sql
SELECT * FROM amazon LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[44]:

order_date	agency_name	listed_ppu	purchase_ppu	item_quantity	item_subtotal	item_shipping_handling	item_tax	item_net_total
2021-10-19	None	29.99	20.99	1	20.99	0	1.97	22.96
2021-06-24	None	42.99	32.99	1	32.99	0	2.94	35.93
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	22.99	22.99	1	22.99	0	1.86	24.85
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	22.99	22.99	1	22.99	0	1.86	24.85
2021-10-19	None	89	89	1	89	0	7.92	96.92
2021-10-19	None	125.99	125.99	1	125.99	0	11.21	137.2
2022-06-30	LABOR AND INDUSTRIES WASHINGTON STATE	10.5	10.5	1	10.5	0	0.86	11.36
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	10.99	10.99	2	21.98	0	1.78	23.76
2022-06-30	FISH AND WILDLIFE DEPARTMENT OF	24.98	24.98	1	24.98	0	2	26.98
2022-06-30	FISH AND WILDLIFE DEPARTMENT	11.99	11.99	1	11.99	0	0.96	12.95

Order Date Dimension Table

- Creating new variables such as is_weekend, quarter_of_year, and day_of_week

In [45]:

```
%%sql
DROP TABLE IF EXISTS order_date;
```

```
CREATE TABLE order_date (
    key SERIAL PRIMARY KEY,
    day CHAR(10),
    year INTEGER,
    month_of_year_str VARCHAR(12),
    month_of_year INTEGER,
    day_of_month INTEGER,
    day_of_week_str CHAR(9),
    day_of_week INTEGER,
    is_weekend BOOLEAN,
    is_weekday BOOLEAN,
    quarter_of_year INTEGER
);
```

```
 * postgresql://student@/Final_Project
Done.
Done.
```

Out[45]: []

In [46]:

```
%%sql

COMMENT ON TABLE order_date IS
'The order date table dimensinal table included the date of each order. We have added other unique identifier
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[46]: []

In [47]:

```
%%sql
INSERT INTO order_date (day, year, month_of_year_str, month_of_year, day_of_month,
                  day_of_week_str, day_of_week, is_weekend, is_weekday,
                  quarter_of_year)
SELECT DISTINCT TO_CHAR(order_date, 'YYYY-MM-DD') AS day,
    CAST(TO_CHAR(order_date, 'YYYY') AS INTEGER) AS year,
    TO_CHAR(order_date, 'Month') AS month_of_year_str,
    CAST(TO_CHAR(order_date, 'MM') AS INTEGER) AS month_of_year,
    CAST(TO_CHAR(order_date, 'DD') AS INTEGER) AS day_of_month,
    TO_CHAR(order_date, 'Day') AS day_of_week_str,
    CAST(TO_CHAR(order_date, 'D') AS INTEGER) AS day_of_week,
    CASE WHEN CAST(TO_CHAR(order_date, 'D') AS INTEGER) IN (1, 7)
        THEN TRUE
        ELSE FALSE
    END AS is_weekend,
    CASE WHEN CAST(TO_CHAR(order_date, 'D') AS INTEGER) NOT IN (1, 7)
        THEN TRUE
        ELSE FALSE
    END AS is_weekday,
    CAST(TO_CHAR(order_date, 'Q') AS INTEGER) AS quarter_of_year
FROM amazon;
```

```
 * postgresql://student@/Final_Project
1759 rows affected.
```

```
Out[47]: []
```

Order_Date Table

```
In [48]: %%sql
SELECT * FROM order_date
LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[48]:

key	day	year	month_of_year_str	month_of_year	day_of_month	day_of_week_str	day_of_week	is_weekend	is_weekday	quart
1	2018-05-04	2018	May	5	4	Friday	6	False	True	
2	2018-07-12	2018	July	7	12	Thursday	5	False	True	
3	2021-04-08	2021	April	4	8	Thursday	5	False	True	
4	2021-02-05	2021	February	2	5	Friday	6	False	True	
5	2022-02-25	2022	February	2	25	Friday	6	False	True	
6	2020-06-29	2020	June	6	29	Monday	2	False	True	
7	2019-05-11	2019	May	5	11	Saturday	7	True	False	
8	2021-01-09	2021	January	1	9	Saturday	7	True	False	
9	2019-10-27	2019	October	10	27	Sunday	1	True	False	
	2019-									

Checking the correctness of the days of the week and whether it is a weekend or not

In [49]:

```
%%sql
SELECT DISTINCT day_of_week_str, day_of_week, is_weekend, is_weekday
FROM order_date
ORDER BY day_of_week;
```

```
 * postgresql://student@/Final_Project
7 rows affected.
```

Out[49]:

day_of_week_str	day_of_week	is_weekend	is_weekday
Sunday	1	True	False
Monday	2	False	True
Tuesday	3	False	True
Wednesday	4	False	True
Thursday	5	False	True
Friday	6	False	True
Saturday	7	True	False

In [50]:

```
%%sql
ALTER TABLE amazon
ADD COLUMN order_date_key INTEGER,
ADD CONSTRAINT fk_order_date
    FOREIGN KEY (order_date_key)
    REFERENCES order_date (key);
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[50]: []

In [51]:

```
%%sql
UPDATE amazon
SET order_date_key = order_date.key
FROM order_date
WHERE TO_CHAR(amazon.order_date, 'YYYY-MM-DD') = order_date.day;
```

```
 * postgresql://student@/Final_Project
486273 rows affected.
```

Out[51]: []

In [52]:

```
%%sql

COMMENT ON TABLE order_date IS
'Add the order_date foreign key and check of the key was populated into the amazon fact table. After which, w
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[52]: []
```

```
In [53]: %%sql
         ALTER TABLE amazon
         DROP COLUMN order_date;
```

```
 * postgresql://student@/Final_Project
Done.
```

```
Out[53]: []
```

Checking order_date key populated into amazon fact table

```
In [54]: %%sql
         SELECT * FROM amazon
         LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[54]:

agency_name	listed_ppu	purchase_ppu	item_quantity	item_subtotal	item_shipping_handling	item_tax	item_net_total	pricing_dis
None	29.99	20.99	1	20.99	0	1.97	22.96	
None	42.99	32.99	1	32.99	0	2.94	35.93	
FISH AND WILDLIFE DEPARTMENT OF	22.99	22.99	1	22.99	0	1.86	24.85	
FISH AND WILDLIFE DEPARTMENT OF	22.99	22.99	1	22.99	0	1.86	24.85	
None	89	89	1	89	0	7.92	96.92	

Seller and Agency Dimension Tables

```
In [55]: %%sql
         DROP TABLE IF EXISTS seller;
```

```
CREATE TABLE seller (
    key SERIAL PRIMARY KEY,
    seller_id VARCHAR(200) UNIQUE
);

DROP TABLE IF EXISTS agency;

CREATE TABLE agency (
    key SERIAL PRIMARY KEY,
    agency_id VARCHAR(200) UNIQUE
);
```

```
 * postgresql://student@/Final_Project
Done.
Done.
Done.
Done.
```

Out[55]: []

In [56]:

```
%%sql

COMMENT ON TABLE seller IS
'The seller table includes the key and the seller name for each order placed';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[56]: []

In [57]:

```
%%sql

COMMENT ON TABLE agency IS
'The agency table includes the key and the agency name for each order placed';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[57]: []

In [58]:

```
%%sql
INSERT INTO seller (seller_id)
SELECT DISTINCT seller_name AS seller_id
FROM amazon;
```

```
 * postgresql://student@/Final_Project
62157 rows affected.
```

Out[58]: []

In [59]:

```
%%sql
ALTER TABLE amazon
ADD COLUMN seller_key INTEGER,
ADD CONSTRAINT fk_seller
    FOREIGN KEY (seller_key)
    REFERENCES seller (key);
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[59]: []

In [60]:

```
%%sql
UPDATE amazon
SET seller_key = seller.key
FROM seller
WHERE amazon.seller_name = seller.seller_id;
```

```
 * postgresql://student@/Final_Project
486218 rows affected.
```

Out[60]: []

In [61]:

```
%%sql
INSERT INTO agency (agency_id)
SELECT DISTINCT agency_name AS agency_id
FROM amazon;
```

```
 * postgresql://student@/Final_Project
62 rows affected.
```

Out[61]: []

In [62]:

```
%%sql
ALTER TABLE amazon
DROP COLUMN IF Exists agency_key,
ADD COLUMN agency_key INTEGER,
ADD CONSTRAINT fk_agency
    FOREIGN KEY (agency_key)
    REFERENCES agency (key);
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[62]: []

In [63]:

```
%%sql
UPDATE amazon
SET agency_key = agency.key
FROM agency
WHERE amazon.agency_name = agency.agency_id;
```

```
 * postgresql://student@/Final_Project
69779 rows affected.
```

Out[63]: []

In [64]:

```
%%sql

COMMENT ON TABLE agency IS
'We added the foreign key and we checked to see if the transfer to the final fact table was successful.';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[64]: []

In [65]:

```
%%sql

COMMENT ON TABLE seller IS
'We added the foreign key and we checked to see if the transfer to the final fact table was successful.';
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[65]: []

Seller Table

In [66]:

```
%%sql
SELECT * FROM seller
LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[66]:

key	seller_id
1	None
2	ORDERIN US
3	PRINY
4	Hard to Find Items
5	Buy-Parts-Here
6	DL-TECH
7	AOKWIT
8	ZHIXINGHEYI store
9	Early Buy
10	Artistry by LA

Agency Table

In [67]:

```
%%sql
SELECT * FROM agency
LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[67]:

key	agency_id
1	None
2	LOTTERY COMMISSION WASHINGTON STATE
3	ECOLOGY DEPARTMENT OF
4	CONSOLIDATED TECHNOLOGY SERVICES
5	PUBLIC INSTRUCTION SUPERINTENDENT OF
6	PARKS AND RECREATION COMMISSION WASHINGTON STATE
7	HORSE RACING COMMISSION WASHINGTON
8	NATURAL RESOURCES DEPARTMENT OF
9	FINANCIAL INSTITUTIONS DEPARTMENT OF
10	WINE COMMISSION

In [68]:

```
%%sql
ALTER TABLE amazon
DROP COLUMN seller_name,
DROP COLUMN agency_name;
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[68]:

```
[]
```

Final Fact Table

- Populated with the 4 dimension table keys
- We can see that all the keys have transfered successfully and we are ready for data analysis

In [69]:

```
%%sql
SELECT * FROM amazon
LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[69]:

listed_ppu	purchase_ppu	item_quantity	item_subtotal	item_shipping_handling	item_tax	item_net_total	pricing_discount_dollar	p
29.99	20.99	1	20.99	0	1.97	22.96	0.0	
42.99	32.99	1	32.99	0	2.94	35.93	0.0	
89	89	1	89	0	7.92	96.92	0.0	
125.99	125.99	1	125.99	0	11.21	137.2	0.0	
22.99	22.99	1	22.99	0	1.86	24.85	0.0	
22.99	22.99	1	22.99	0	1.86	24.85	0.0	
10.5	10.5	1	10.5	0	0.86	11.36	0.0	
10.99	10.99	2	21.98	0	1.78	23.76	0.0	
24.98	24.98	1	24.98	0	2	26.98	0.0	

In [70]:

```
%%sql

COMMENT ON TABLE amazon IS
'This now represent the final stage of the Amazon Fact Table. In inlcudes all the facts and dimensions and is
```

```
 * postgresql://student@/Final_Project
Done.
```

Out[70]:

```
[]
```

Ask 3: Data Analysis

1. What are the notable time series trends that characterize purchases from Amazon by Washington Agencies over the years?

a. In which months of the 2022 fiscal year do Washington Agencies exhibit higher spending patterns on Amazon purchases?

Business Need: Pinpointing peak spending months by Washington Agencies on Amazon allows for targeted advertising during these periods. Offering discounts and tailored pricing plans during low-order months incentivizes agencies to increase their spending beyond typical patterns.

In [71]:

```
%%sql
SELECT
    od.month_of_year AS month,
    SUM(amazon.item_net_total) AS total_monthly_spending
FROM
    amazon
JOIN
    order_date od ON amazon.order_date_key = od.key
JOIN
    agency ON amazon.agency_key = agency.key
GROUP BY
    od.month_of_year
ORDER BY
    total_monthly_spending DESC;
```

```
 * postgresql://student@/Final_Project
8 rows affected.
```

Out[71]:

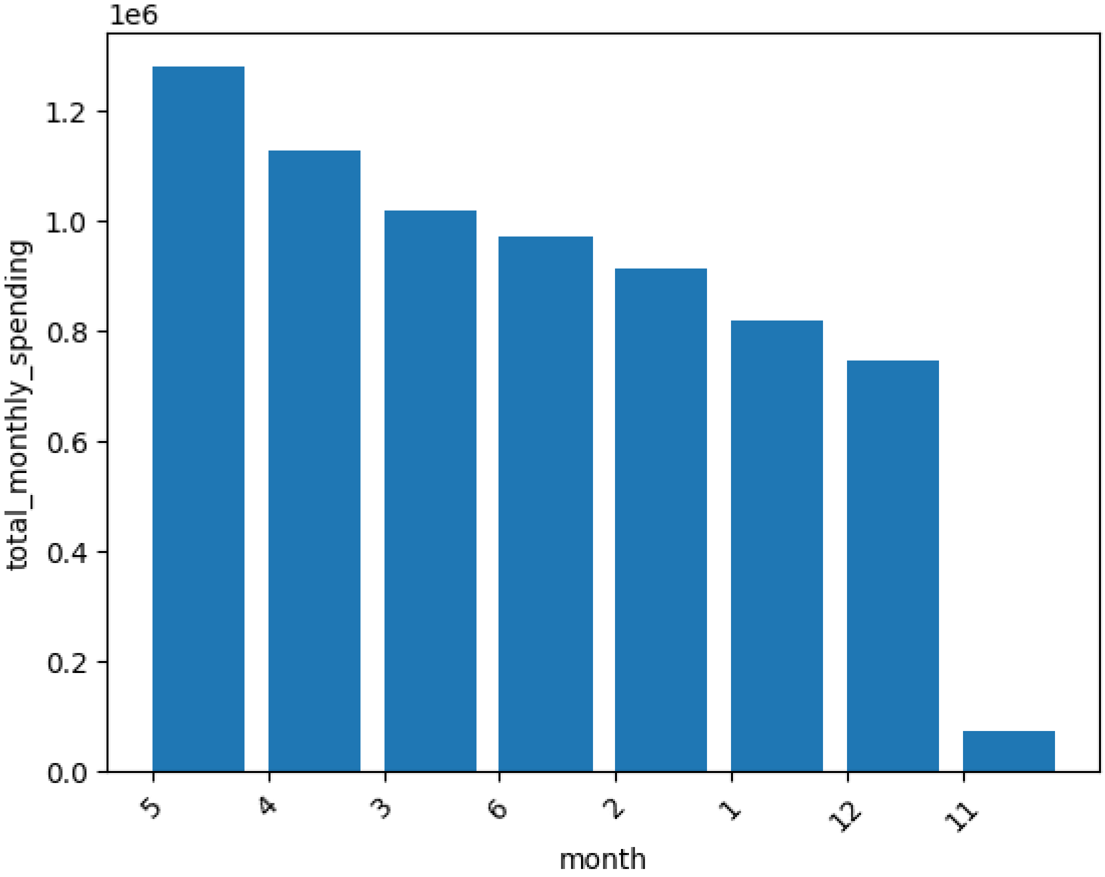
month	total_monthly_spending
5	1277761.39
4	1127827.14
3	1016801.08
6	968615.28
2	911949.16
1	816127.28
12	746745.85
11	73520.42

In [72]:

```
%matplotlib inline
_.bar(width = 0.8,align = 'edge')
```

Out[72]:

```
<BarContainer object of 8 artists>
```



Analysis: The analysis of 2022 fiscal year spending patterns among Washington Agencies on Amazon indicates a peak expenditure in May, totaling just over 1.27 million dollars. This peak is consistent with elevated sales in April and March. Contrary to the fiscal year starting in July, the high spending trend suggests agencies are allocating their budgets toward the calendar year-end, likely to prevent reductions and fully utilize available funds.

Given these insights, targeting advertising efforts towards Washington Agencies during peak spending months (March, April, May) would be strategic for Amazon. Additionally, offering discounts or tailored business pricing plans during low-spending periods (July, September, November) can incentivize purchases during these typically quieter months.

b. During these high spend months, which Washington Agencies tend to spend the most on ordering from Amazon?

Business Need: Identifying the top spending Washington Agencies during high expenditure months on Amazon provides critical insights for targeted marketing strategies. The consistent leadership of the Department of Social and Health Services and the Department of Children, Youth, and Families in March, April, and May signifies their substantial purchasing influence. To maximize sales, targeted advertising campaigns should prioritize these high-spending agencies with significant purchasing power. Simultaneously, offering discounts to agencies with lower spending tendencies can stimulate increased purchases during these peak months.

May

In [73]:

```
%%sql
SELECT
    a.agency_id,
    SUM(o.item_net_total) AS total_spend_for_month
FROM
    amazon AS o
JOIN
    order_date AS od ON o.order_date_key = od.key
JOIN
    agency AS a ON o.agency_key = a.key
WHERE
    od.month_of_year = 5
GROUP BY
    a.agency_id
ORDER BY
    total_spend_for_month DESC
LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[73]:

agency_id	total_spend_for_month
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	293475.51
CHILDREN YOUTH AND FAMILIES DEPT OF	229636.88
CORRECTIONS DEPARTMENT OF	221836.55
FISH AND WILDLIFE DEPARTMENT OF	68954.19
LABOR AND INDUSTRIES WASHINGTON STATE	50612.71

April

In [74]:

```
%%sql
SELECT
    a.agency_id,
    SUM(o.item_net_total) AS total_spend_for_month
FROM
    amazon AS o
JOIN
    order_date AS od ON o.order_date_key = od.key
JOIN
    agency AS a ON o.agency_key = a.key
WHERE
    od.month_of_year = 4
GROUP BY
    a.agency_id
ORDER BY
    total_spend_for_month DESC
LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[74]:

agency_id	total_spend_for_month
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	295998.70
CHILDREN YOUTH AND FAMILIES DEPT OF	207356.42
CORRECTIONS DEPARTMENT OF	147737.60
FISH AND WILDLIFE DEPARTMENT OF	71738.06
NATURAL RESOURCES DEPARTMENT OF	49330.08

March

In [75]:

```
%%sql
SELECT
    a.agency_id,
    SUM(o.item_net_total) AS total_spend_for_month
FROM
    amazon AS o
JOIN
    order_date AS od ON o.order_date_key = od.key
JOIN
```

```
    agency AS a ON o.agency_key = a.key
WHERE
    od.month_of_year = 3
GROUP BY
    a.agency_id
ORDER BY
    total_spend_for_month DESC
LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[75]:

agency_id	total_spend_for_month
CHILDREN YOUTH AND FAMILIES DEPT OF	233240.78
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	197650.75
CORRECTIONS DEPARTMENT OF	133157.34
FISH AND WILDLIFE DEPARTMENT OF	55356.23
TRANSPORTATION DEPARTMENT OF	45939.96

Analysis: Examining spending patterns across high expenditure months reveals consistent dominance by certain Washington Agencies. The Department of Social and Health Services and the Department of Children, Youth, and Families consistently outpace others in March, April, and May, surpassing the Department of Corrections. Leveraging this insight, targeted advertising initiatives should emphasize these top spenders, acknowledging their substantial purchasing capacity. Additionally, incentivizing lower-spending agencies with discounts during these high-spend periods could effectively prompt increased purchases, thereby optimizing Amazon's sales strategy across varying agency spending behaviors.

c. How do the Net Purchase Totals from all Washington Agencies fluctuate yearly, and what factors contribute to these trends?

Business Need: Understanding the annual fluctuations in Net Purchase Totals from Washington Agencies is critical for Amazon's strategic planning. By comprehending the factors driving these variations, Amazon can tailor its approach to capitalize on peak spending periods and mitigate downturns. Insights into the influences behind these fluctuations, such as fiscal year-end trends, external events like the COVID-19 pandemic, or shifts in agency priorities, empower Amazon to adapt its sales and marketing strategies accordingly. This knowledge enables Amazon to offer targeted solutions, anticipate spending behaviors, and optimize offerings to ensure consistent and maximized sales to Washington Agencies throughout the year.

```
In [76]: %%sql
SELECT
    EXTRACT(YEAR FROM CAST(od.day AS DATE)) AS year,
    SUM(amazon.item_net_total) AS total_net_purchase
FROM
    amazon
JOIN
    order_date od ON amazon.order_date_key = od.key
JOIN
    agency a ON amazon.agency_key = a.key
GROUP BY
    EXTRACT(YEAR FROM CAST(od.day AS DATE))
ORDER BY
    year DESC;
```

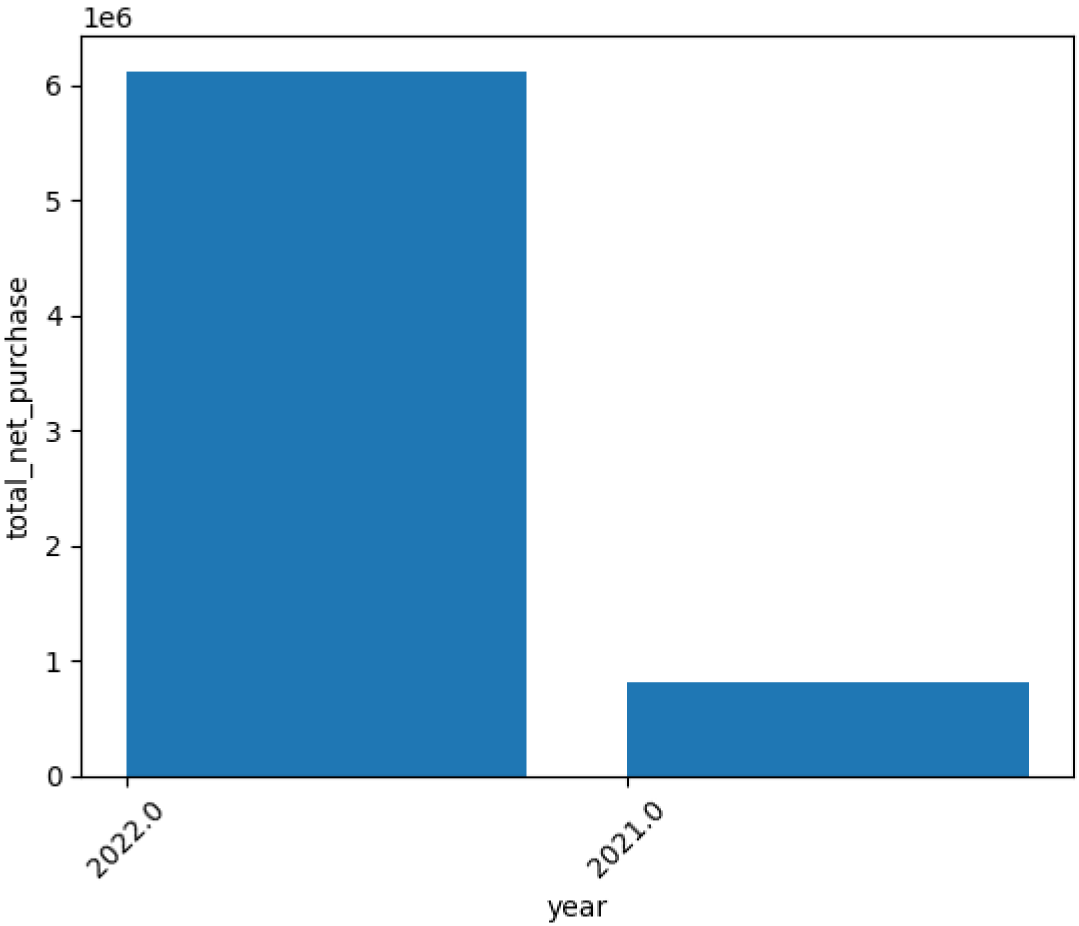
```
 * postgresql://student@/Final_Project
2 rows affected.
```

Out[76]:

year	total_net_purchase
2022.0	6119081.33
2021.0	820266.27

```
In [77]: %matplotlib inline
_.bar(width = 0.8,align = 'edge')
```

```
Out[77]: <BarContainer object of 2 artists>
```



Analysis: In 2021, there was a significant drop in total net purchases, possibly influenced by the disruptive effects of COVID-19. This impact likely redirected spending priorities for government agencies like the Washington Agencies, emphasizing healthcare and disease management needs.

However, the scenario shifted notably in 2022, with a substantial increase in total net purchases to $6,119,081.33, surpassing the preceding year's figures. This surge indicates a recovery or shift in spending patterns, highlighting the resilience or altered priorities within Washington Agencies post-pandemic. Understanding these fluctuations is crucial for Amazon to adapt strategies and sustain or augment sales to Washington Agencies.

2. What trends emerge in recurring orders, bulk orders, and customer attrition among the individual Washington Agencies that spend the most on Amazon purchases?

a. Which Washington Agencies have recorded the highest order volumes on Amazon during the Fiscal Year 2022?

Business Need: Identifying top buyers among Washington Agencies is crucial for Amazon's retention strategy, as these clients significantly contribute to overall profits. Maintaining consistent engagement with these agencies is essential to ensure continued patronage. Offering loyalty incentives and tailored advertising is pivotal to solidify relationships with these high-volume purchasers.

In [78]:

```
%%sql
SELECT
    COUNT(*) AS order_count,
    a.agency_id
FROM
    amazon AS o
JOIN
    agency AS a ON o.agency_key = a.key
JOIN
    order_date AS od ON o.order_date_key = od.key
WHERE
    EXTRACT(YEAR FROM CAST(od.day AS DATE)) = 2022
GROUP BY
    a.agency_id
ORDER BY
    order_count DESC
LIMIT 10;
```

```
 * postgresql://student@/Final_Project
10 rows affected.
```

Out[78]:

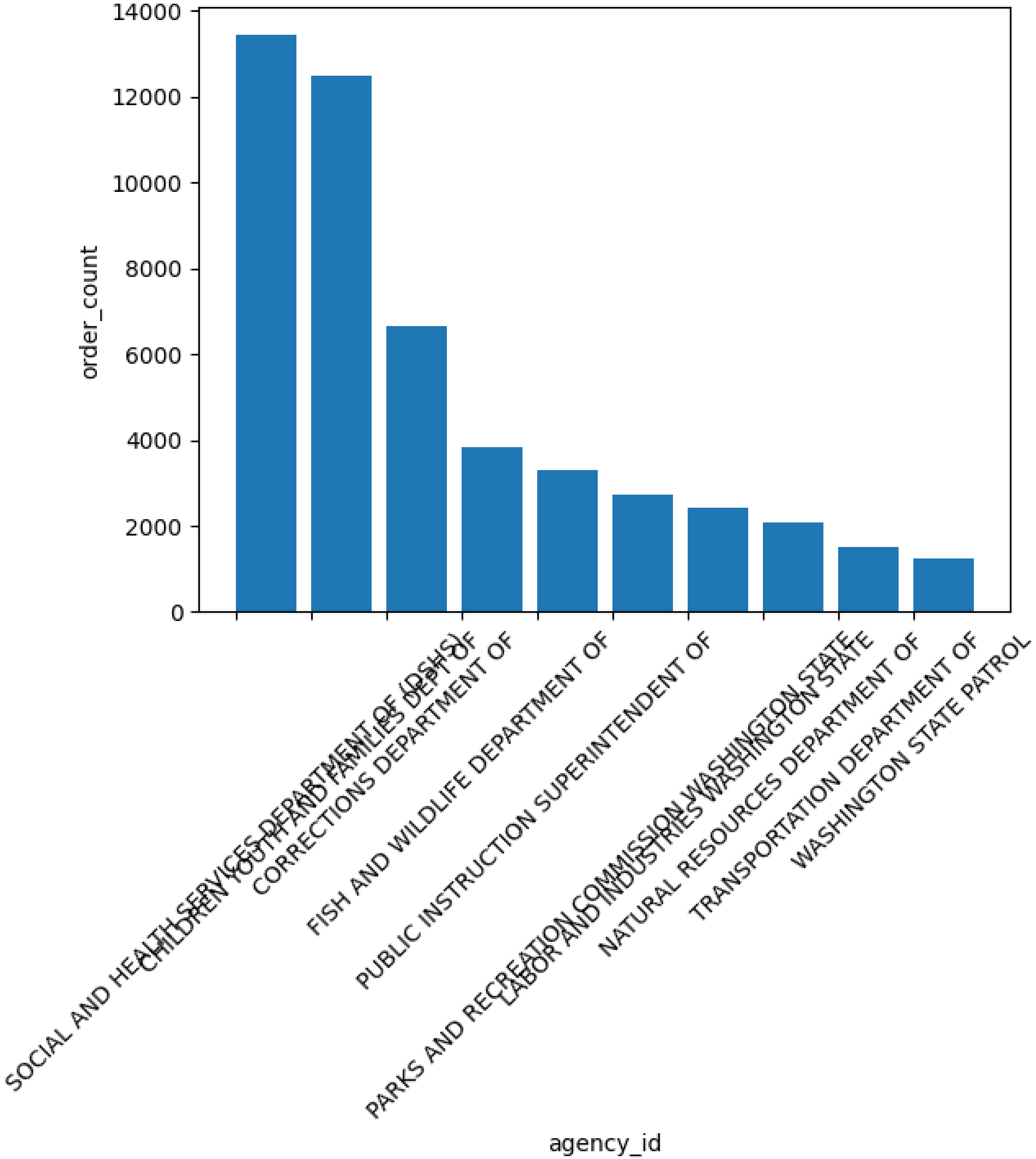
order_count	agency_id
13417	SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)
12485	CHILDREN YOUTH AND FAMILIES DEPT OF
6642	CORRECTIONS DEPARTMENT OF
3840	FISH AND WILDLIFE DEPARTMENT OF
3281	PUBLIC INSTRUCTION SUPERINTENDENT OF
2744	PARKS AND RECREATION COMMISSION WASHINGTON STATE
2413	LABOR AND INDUSTRIES WASHINGTON STATE
2090	NATURAL RESOURCES DEPARTMENT OF
1505	TRANSPORTATION DEPARTMENT OF
1231	WASHINGTON STATE PATROL

In [79]:

```
%matplotlib inline
_.bar(width = 0.8,align = 'edge')
```

Out[79]:

```
<BarContainer object of 10 artists>
```



order_count
14000
12000
10000
8000
6000
4000
2000
0
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)
CHILDREN YOUTH AND FAMILIES DEPT OF
CORRECTIONS DEPARTMENT OF
FISH AND WILDLIFE DEPARTMENT OF
PUBLIC INSTRUCTION SUPERINTENDENT OF
PARKS AND RECREATION COMMISSION WASHINGTON STATE
LABOR AND INDUSTRIES WASHINGTON STATE
NATURAL RESOURCES DEPARTMENT OF
TRANSPORTATION DEPARTMENT OF
WASHINGTON STATE PATROL
agency_id

Analysis: The Department of Social and Health Services emerges as the top buyer among Washington Agencies based on order volume through Amazon. Sustaining advertising efforts directed at these agencies is imperative to maintain their loyalty and consistent orders. Additionally, implementing loyalty discounts specifically tailored for these top purchasing agencies can further cement their relationship with Amazon. However, delving deeper into purchase patterns reveals that while some agencies place fewer individual orders, they might compensate by making bulk purchases. Understanding this distinction aids in refining strategies to cater to both frequent and bulk-buying agencies, ensuring comprehensive retention efforts across different buying behaviors

b. Among Washington Agencies, which ones are placing Bulk Orders (orders with item quantities exceeding 100) on Amazon?

Business Need: Identifying Washington Agencies that frequently place bulk orders on Amazon is essential for Amazon's tailored pricing strategies. Offering Quantity Pricing or Business Pricing to these agencies is crucial to encourage continued bulk purchasing behavior, ensuring sustained business relationships.

In [80]:

```
%%sql
SELECT
    a.agency_id,
    COUNT(*) as bulk_order_count
FROM
    amazon AS o
JOIN
    agency AS a ON o.agency_key = a.key
WHERE
    o.item_quantity > 100
GROUP BY
    a.agency_id
ORDER BY
    bulk_order_count DESC;
```

```
 * postgresql://student@/Final_Project
8 rows affected.
```

Out[80]:

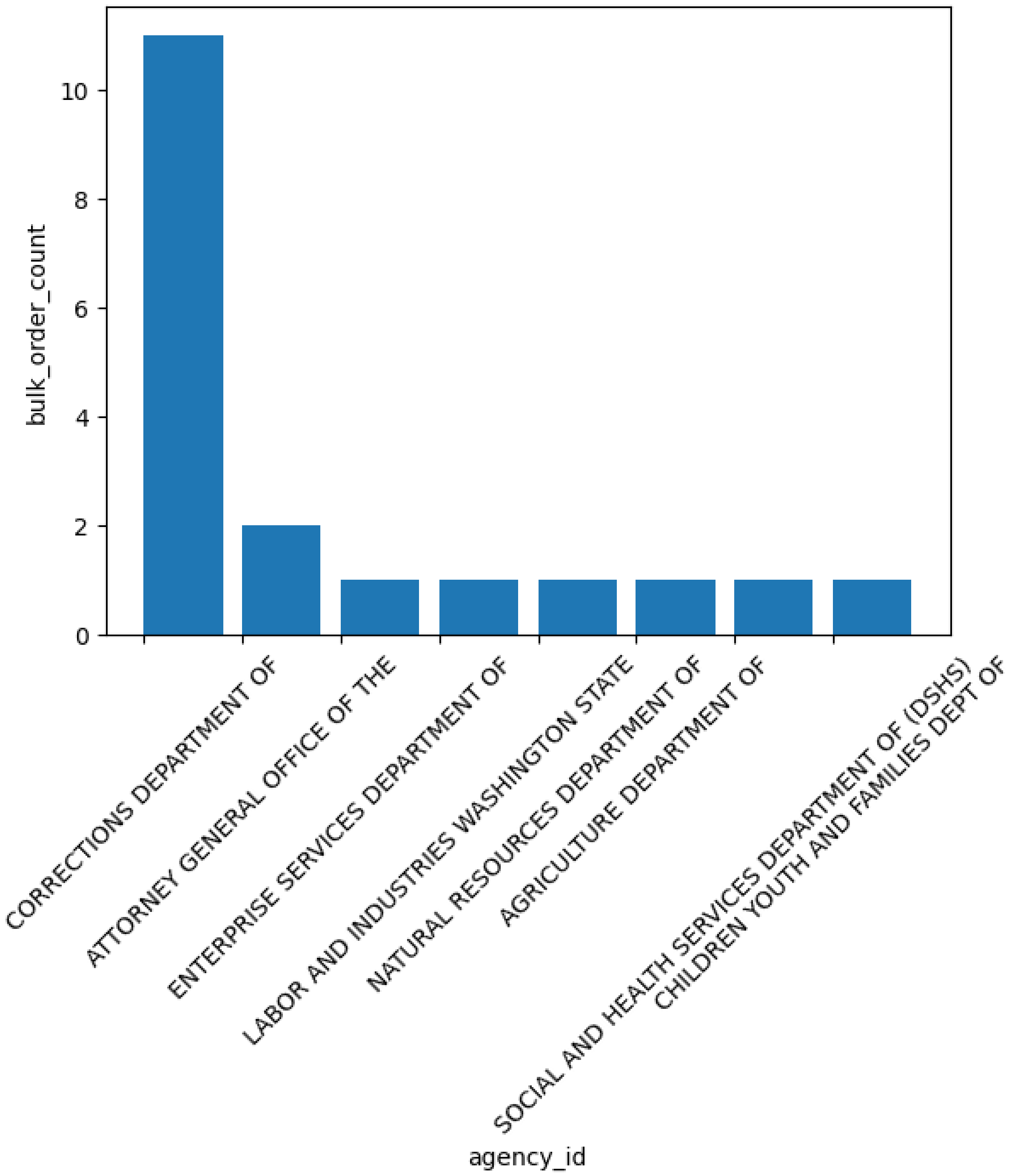
agency_id	bulk_order_count
CORRECTIONS DEPARTMENT OF	11
ATTORNEY GENERAL OFFICE OF THE	2
ENTERPRISE SERVICES DEPARTMENT OF	1
LABOR AND INDUSTRIES WASHINGTON STATE	1
NATURAL RESOURCES DEPARTMENT OF	1
AGRICULTURE DEPARTMENT OF	1
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	1
CHILDREN YOUTH AND FAMILIES DEPT OF	1

In [81]:

```
%matplotlib inline
_.bar(width = 0.8,align = 'edge')
```

Out[81]:

```
<BarContainer object of 8 artists>
```


bulk_order_count
0
2
4
6
8
10
CORRECTIONS DEPARTMENT OF
ATTORNEY GENERAL OFFICE OF THE
ENTERPRISE SERVICES DEPARTMENT OF
LABOR AND INDUSTRIES WASHINGTON STATE
NATURAL RESOURCES DEPARTMENT OF
AGRICULTURE DEPARTMENT OF
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)
CHILDREN YOUTH AND FAMILIES DEPT OF
agency_id

Analysis: Agencies like the Department of Corrections regularly placing bulk orders should be targeted for Quantity Pricing discounts to incentivize their ongoing bulk purchases. Additionally, proposing a subscription model for these high-volume orders could be advantageous. Shifting these agencies towards a subscription-based purchasing approach might reduce attrition rates associated with large one-time purchases, providing a more consistent revenue stream for Amazon while accommodating the agencies' budgeting preferences. This strategy aims to retain high-profit orders while mitigating potential spending attrition.

c. Which Washington Agencies previously ordered from Amazon but have stopped?

Business Need: Assessing the net gain or loss of Washington Agencies as Amazon customers is crucial to understand the overall impact on revenue. This evaluation aids Amazon in determining the value of re-engaging with agencies that stopped ordering, as well as in strategizing to retain newly acquired customers. Insights gained from this analysis help shape incentives and engagement tactics for customer retention and acquisition.

In [82]:

```sql
%%sql
SELECT
    a.agency_id,
    string_agg(CAST(year AS VARCHAR), ',') AS years,
    COUNT(year) AS count
FROM (
    SELECT
        a.agency_id,
        od.year,
        COUNT(*) AS order_count
    FROM
        amazon AS o
    JOIN
        agency AS a ON o.agency_key = a.key
    JOIN
        order_date AS od ON o.order_date_key = od.key
    GROUP BY
        a.agency_id, od.year
) AS A
GROUP BY
    a.agency_id
ORDER BY
    count
LIMIT 11;
```

```
 * postgresql://student@/Final_Project
11 rows affected.
```

Out[82]:

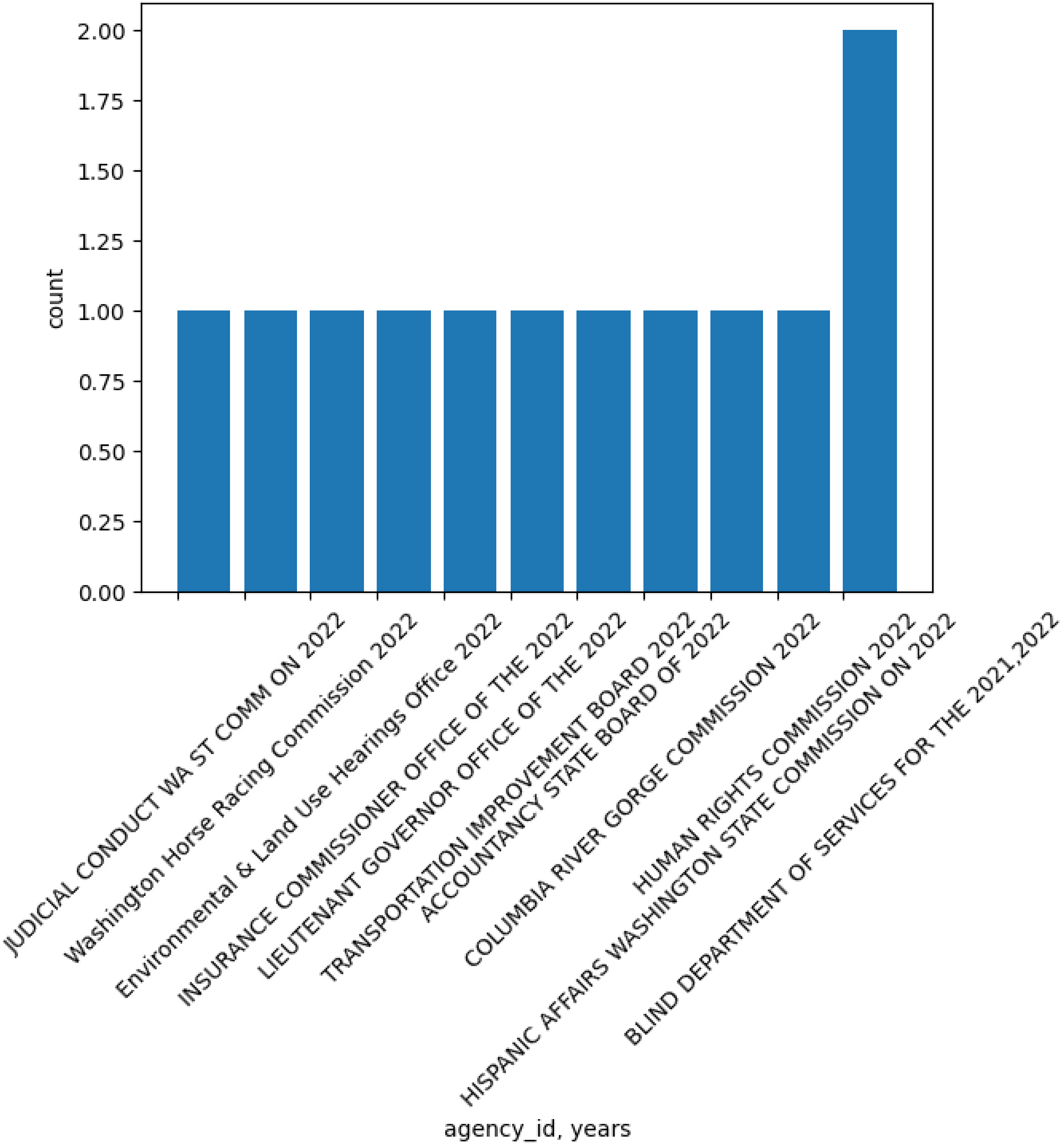
agency_id	years	count
JUDICIAL CONDUCT WA ST COMM ON	2022	1
Washington Horse Racing Commission	2022	1
Environmental & Land Use Hearings Office	2022	1
INSURANCE COMMISSIONER OFFICE OF THE	2022	1
LIEUTENANT GOVERNOR OFFICE OF THE	2022	1
TRANSPORTATION IMPROVEMENT BOARD	2022	1
ACCOUNTANCY STATE BOARD OF	2022	1
COLUMBIA RIVER GORGE COMMISSION	2022	1
HISPANIC AFFAIRS WASHINGTON STATE COMMISSION ON	2022	1
HUMAN RIGHTS COMMISSION	2022	1
BLIND DEPARTMENT OF SERVICES FOR THE	2021,2022	2

In [83]:

```
%matplotlib inline
_.bar(width = 0.8,align = 'edge')
```

Out[83]:

```
<BarContainer object of 11 artists>
```


count
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
JUDICIAL CONDUCT WA ST COMM ON 2022
Washington Horse Racing Commission 2022
Environmental & Land Use Hearings Office 2022
INSURANCE COMMISSIONER OFFICE OF THE 2022
LIEUTENANT GOVERNOR OFFICE OF THE 2022
TRANSPORTATION IMPROVEMENT BOARD 2022
ACCOUNTANCY STATE BOARD OF 2022
COLUMBIA RIVER GORGE COMMISSION 2022
HUMAN RIGHTS COMMISSION 2022
HISPANIC AFFAIRS WASHINGTON STATE COMMISSION ON 2022
BLIND DEPARTMENT OF SERVICES FOR THE 2021,2022
agency_id, years

Analysis: Several agencies, including the Commission on Judicial Conduct, Transportation Improvement Board, State Board of Accountancy, Human Rights Commission, and others, did not engage with Amazon in 2021 but initiated orders in 2022. This surge indicates a potential increase in Amazon's customer base during that period, possibly attributed to post-COVID spending trends. Calculating the spending from these agencies provides a clear understanding of the net gain or loss for Amazon. This insight is pivotal in assessing the impact of customer acquisition and attrition, guiding strategies to either regain lost customers or further incentivize and retain the newly acquired ones.

3. Who are the primary competitor sellers to Amazon's own products on Amazon.com, and what are their sales offerings, magnitudes of sales, and competitive product categories?

```
a. Identify the top 5 competitor sellers that Agencies are sourcing from on Amazon.com and
quantify their net totals.
```

- **Note:** In conducting this analysis, I discovered that the company, "VARIDESK", had it's seller_name switched to "Vari." in this data file during the year 2021. Therefore, this code is used to clean the data so that the seller name is consistent across all records. This will give a more accurate display of the net totals across the five years of records in the file.
    - After careful observation, none of the other top 5 competitors have this issue in the data set, and the analysis will be carried out accordingly

Identifying Amazon's top 5 competitors based on the sum of sum of sales per item [sum(item_net_total)], listed as "total_net_sales":

In this query, the coalesce function was used to sum the up the net sales because there are a few records that have null values for the "item_net_total" column in the original csv file. Without it, the total_net_sales calculation results in a null value.

In [84]:

```
%%sql

SELECT
    s.seller_id,
    COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales
FROM
    seller s, amazon amz
WHERE
    s.key = amz.seller_key
GROUP BY
```

```
    s.seller_id
ORDER BY
    total_net_sales DESC
LIMIT 6;
```

```
 * postgresql://student@/Final_Project
6 rows affected.
```

Out[84]:

seller_id	total_net_sales
Amazon.com	17842264.55
VARIDESK	242604.77
AnkerDirect	169637.61
Tracfone Wireless Inc.	124844.94
ARF Tech	120191.00
Beach Camera Same Day Shipping	116226.72

In [85]:

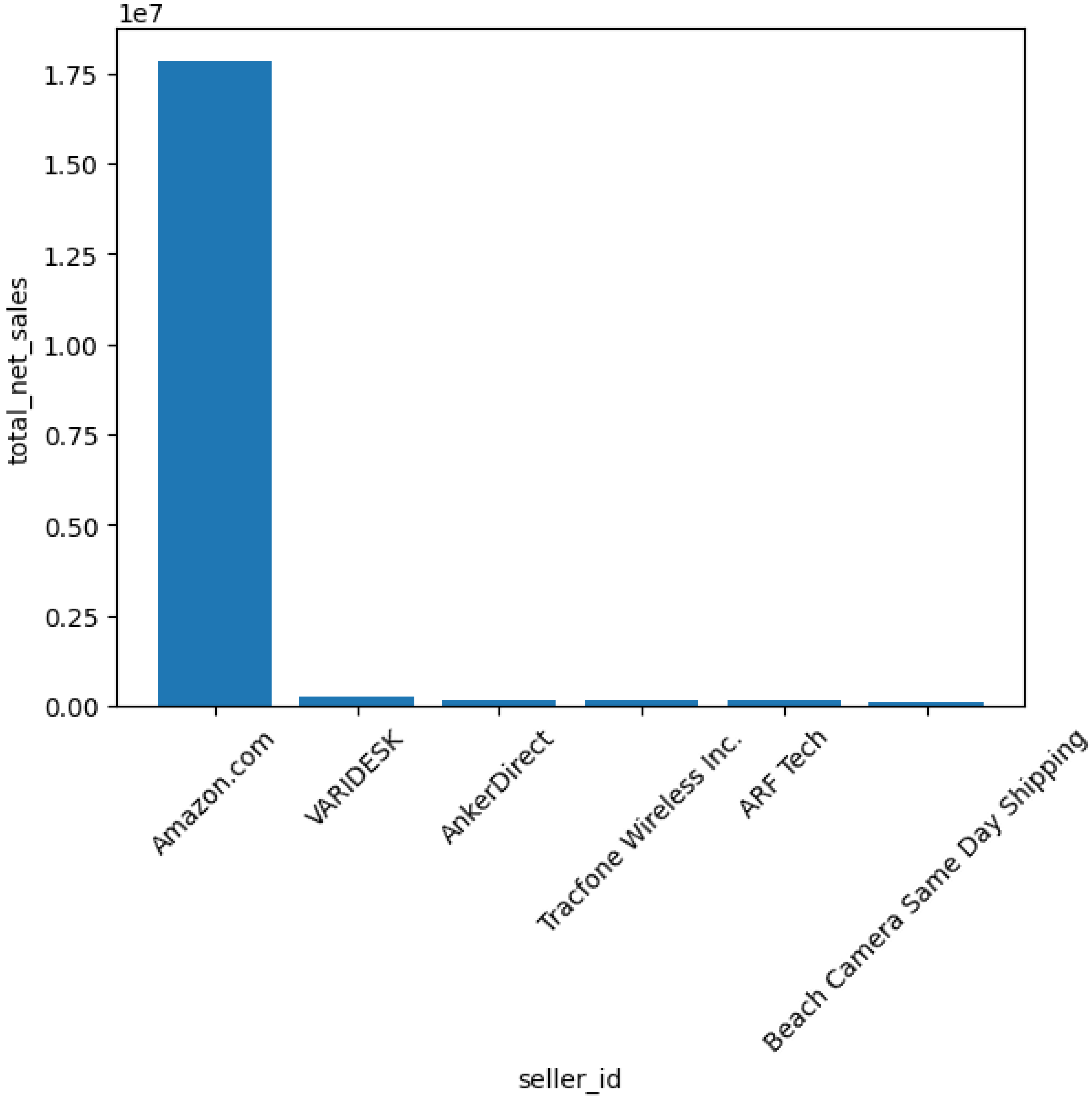
```
%matplotlib inline
```

In [86]:

```
figsize=(100, 100)

_.bar()
```

Out[86]:

```
<BarContainer object of 6 artists>
```



- It is clear that Amazon is the leading seller to Washington Agencies in this data set by two orders of magnitude. Agencies have made a total of 17,842,624.55 in purchases compared to the next highest seller, VARIDESK, of which purchases total in 242,604.77. AnkerDirect, Tracfone Wireless Inc., ARF Tech, and Beach Camera Same Day Shipping are the subsequent competitors with the highest net sales in the 100s of 1000s of dollars. These are the companies that Amazon should be most concerned with in terms of market share in total revenue from sales. They are the biggest threat to Amazon's market share and profit.

Quantifying the total net sales of competitors by year:

This query is focused on the change in total net sales from 2017-2021 of Amazon's top 5 competitors. This analysis does not include the year 2022 because the data in this set only covers up to June of 2022. By using the “IN” function under the “WHERE” section of the query, we specified that we want to see the total net sales by year for only these competitors.

In [87]:

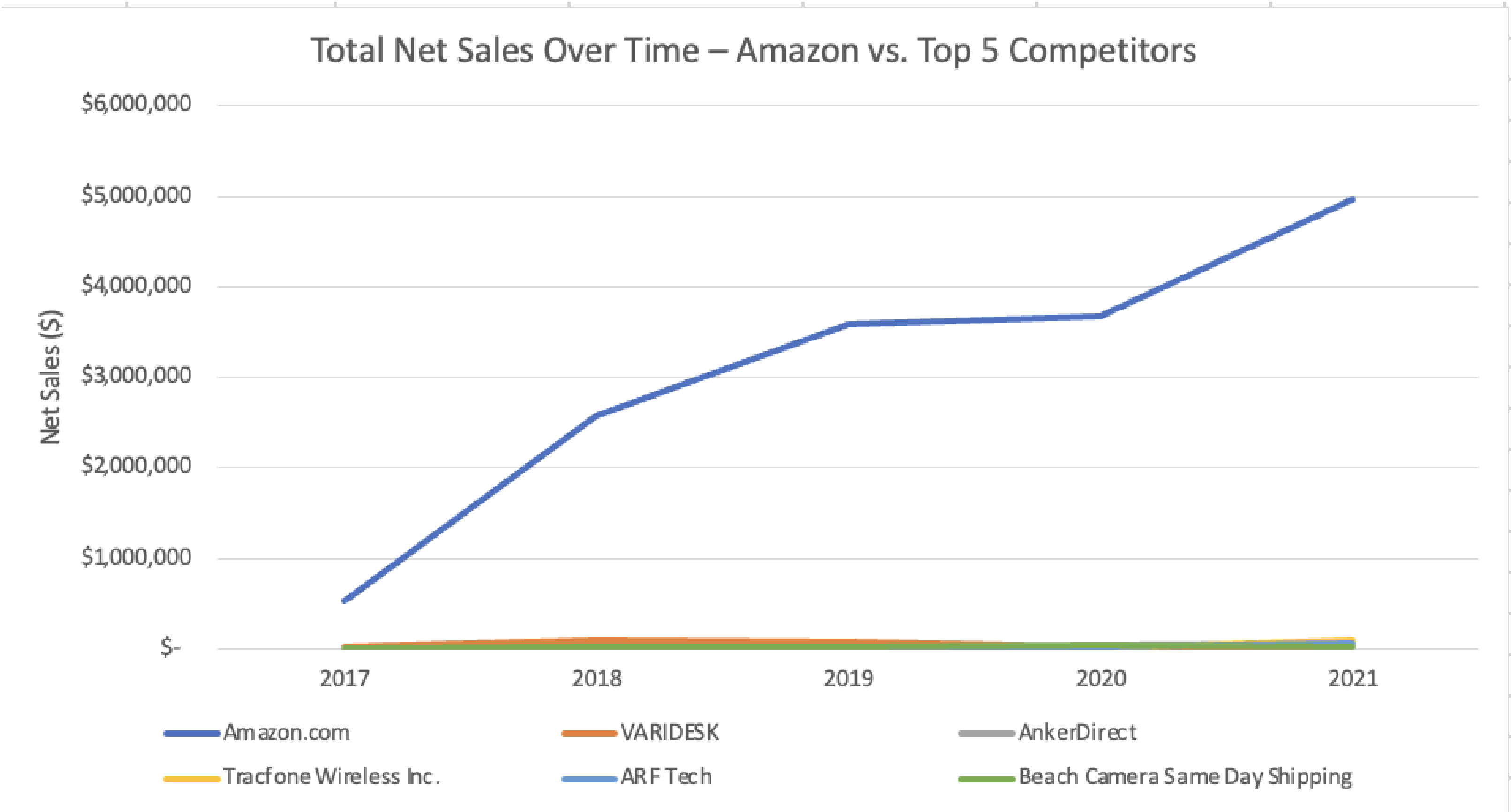
```
%%sql

SELECT
    s.seller_id,
    o.year,
    COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales
FROM
    seller s, order_date o, amazon amz
WHERE
    s.key = amz.seller_key and
    o.key = amz.order_date_key and
    s.seller_id IN ('Amazon.com', 'VARIDESK', 'AnkerDirect', 'Tracfone Wireless Inc.', 'ARF Tech', 'Beach Cam
GROUP BY
    s.seller_id, o.year
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
33 rows affected.
```

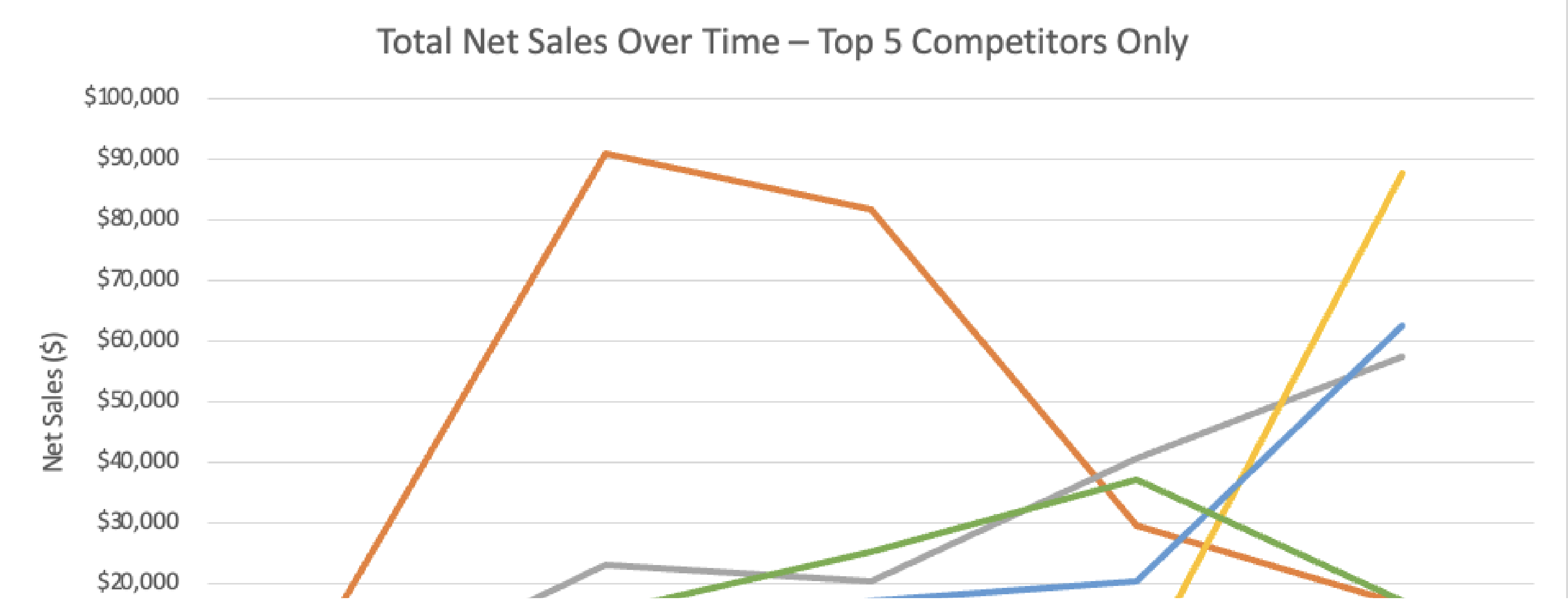
Out[87]:

seller_id	year	total_net_sales
Amazon.com	2021	4975449.82
Amazon.com	2020	3675412.66
Amazon.com	2019	3593555.48
Amazon.com	2018	2578902.81
Amazon.com	2022	2482565.59
Amazon.com	2017	536378.19
VARIDESK	2018	91462.61
Tracfone Wireless Inc.	2021	87670.44
VARIDESK	2019	85261.38
ARF Tech	2021	62669.53
AnkerDirect	2021	57516.54
AnkerDirect	2020	40652.04
Beach Camera Same Day Shipping	2020	37174.80
Tracfone Wireless Inc.	2022	32462.60
VARIDESK	2020	29645.86
AnkerDirect	2022	25613.26
Beach Camera Same Day Shipping	2019	25248.81
AnkerDirect	2018	23153.96
AnkerDirect	2019	20465.92
ARF Tech	2020	20459.71
Beach Camera Same Day Shipping	2022	18074.45
Beach Camera Same Day Shipping	2021	17297.67
ARF Tech	2019	17154.26
VARIDESK	2021	16697.73
VARIDESK	2017	16215
Beach Camera Same Day Shipping	2018	15414.21

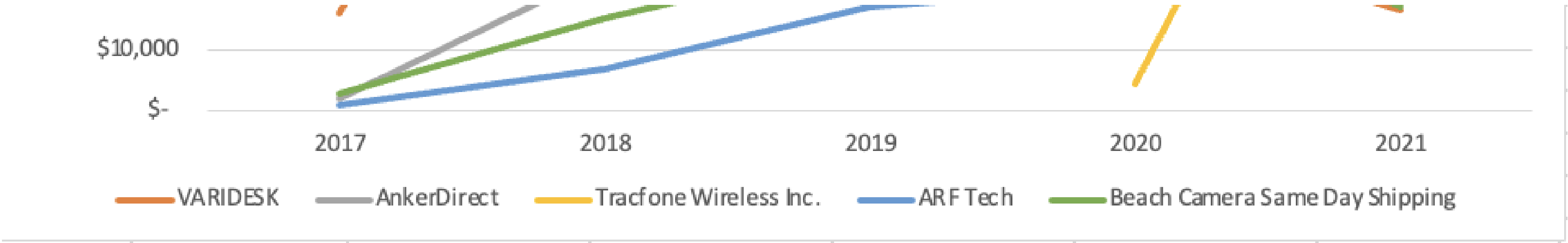
seller_id	year	total_net_sales
ARF Tech	2022	12012.90
ARF Tech	2018	6938.67
Tracfone Wireless Inc.	2020	4711.90
VARIDESK	2022	3322.19
Beach Camera Same Day Shipping	2017	3016.78
AnkerDirect	2017	2235.89
ARF Tech	2017	955.93



Total Net Sales Over Time – Amazon vs. Top 5 Competitors
$6,000,000
$5,000,000
$4,000,000
$3,000,000
$2,000,000
$1,000,000
$-
Net Sales ($)
2017
2018
2019
2020
2021
Amazon.com
VARIDESK
AnkerDirect
Tracfone Wireless Inc.
ARF Tech
Beach Camera Same Day Shipping



Total Net Sales Over Time – Top 5 Competitors Only
$100,000
$90,000
$80,000
$70,000
$60,000
$50,000
$40,000
$30,000
$20,000
Net Sales ($)



- The visuals convey that Agency's purchases from Amazon as a seller have significantly increased since 2021 on a much larger scale compared to its competitors. Nonetheless, the visual provides helpful insight on competitor trends. While VARIDESK has had the highest net sales of any competitors, but since 2018, its sales have significantly slowed. Similarly, Beach Camera Same Day Shipping has had decreases in sales from agencies since 2020. Amazon should be most concerned with emerging competitors including AnkerDirect, ARF Tech, and Tracfone Wireless Inc. That have seen spikes in net sales in the past few years. These are likely to become the most threatening competitors if net sales continue to increase as the graphs portray

Total net sales YTD from data set (Jan 1 – Jun 30 2022):

This analysis takes a deeper dive into the trends of amazon and its competitors in sales during the first half of 2022. The order date year is filtered so that total_net_sales from each month in 2022 are displayed.

```
In [88]: %%sql

SELECT
    s.seller_id,
    o.month_of_year,
    o.year,
    COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales
FROM
    seller s, order_date o, amazon amz
WHERE
    s.key = amz.seller_key and
    o.key = amz.order_date_key and
    s.seller_id IN ('Amazon.com', 'VARIDESK', 'AnkerDirect', 'Tracfone Wireless Inc.', 'ARF Tech', 'Beach Cam
    o.year = 2022
GROUP BY
    s.seller_id, o.year, o.month_of_year
```

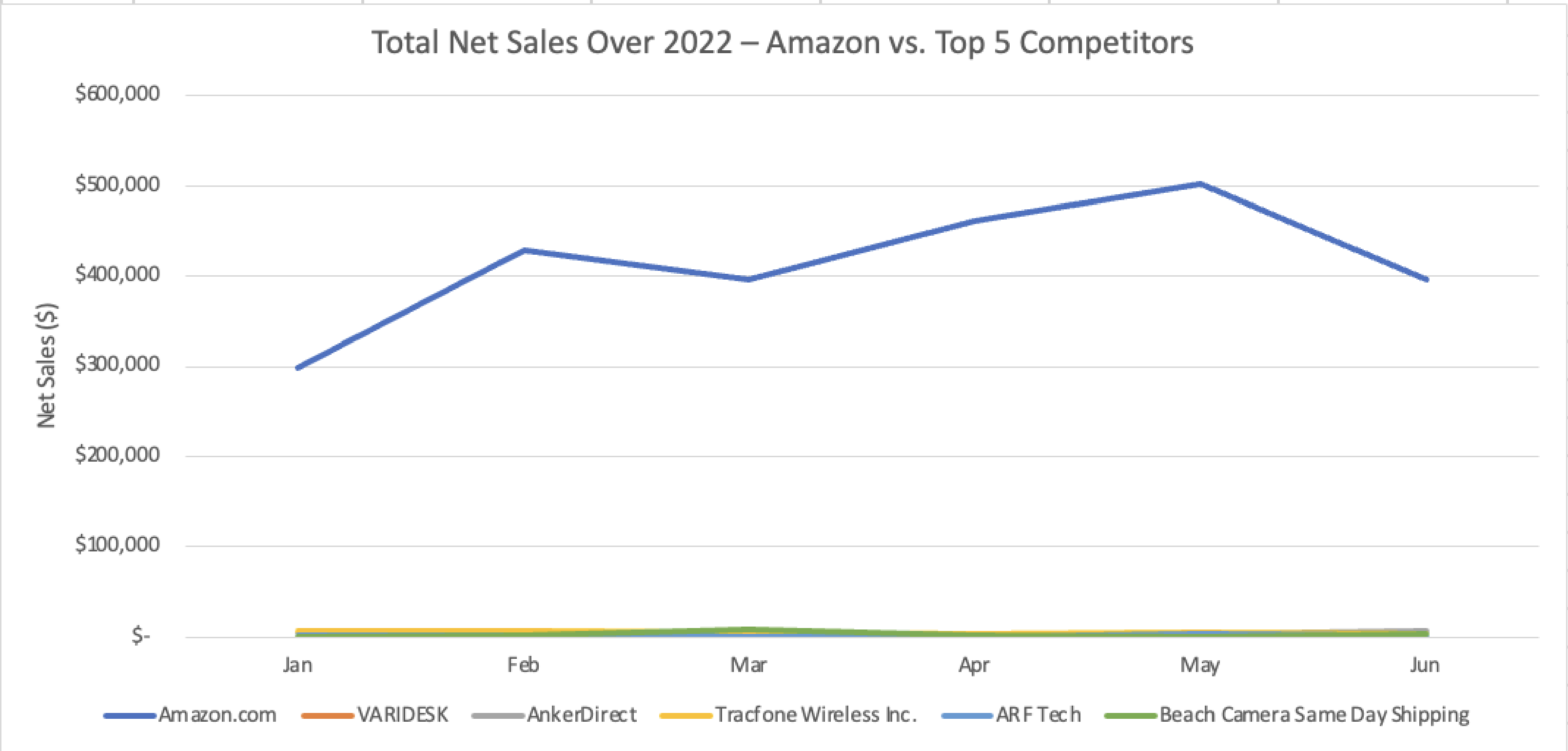
```
ORDER BY
    seller_id;
```

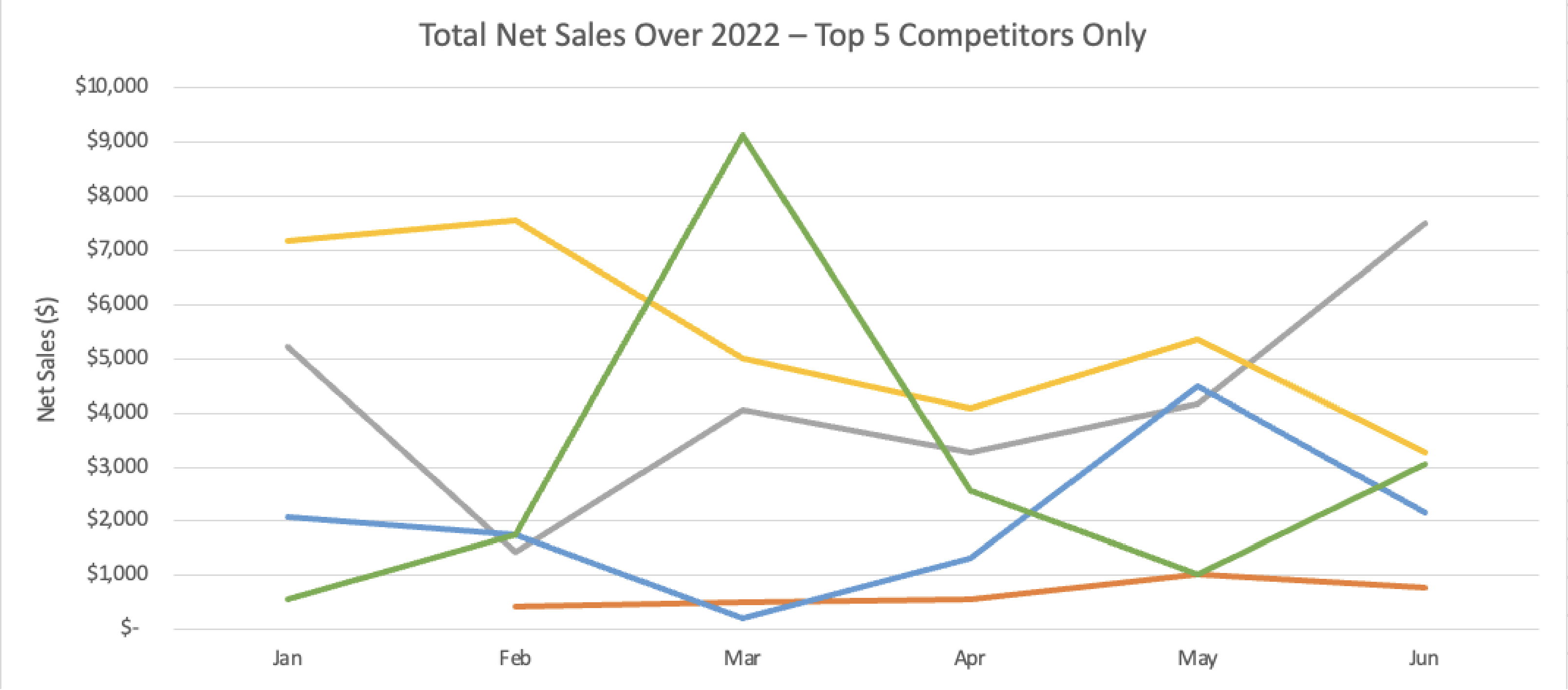
```
 * postgresql://student@/Final_Project
35 rows affected.
```

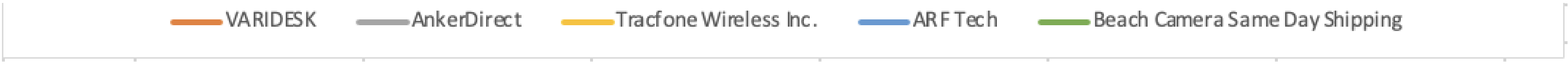
Out[88]:

seller_id	month_of_year	year	total_net_sales
ARF Tech	1	2022	2073.31
ARF Tech	2	2022	1750.33
ARF Tech	3	2022	210.05
ARF Tech	4	2022	1314.27
ARF Tech	5	2022	4503.00
ARF Tech	6	2022	2161.94
Amazon.com	1	2022	298386.57
Amazon.com	2	2022	428220.66
Amazon.com	3	2022	396420.85
Amazon.com	4	2022	460519.33
Amazon.com	5	2022	502613.89
Amazon.com	6	2022	396404.29
AnkerDirect	1	2022	5213.34
AnkerDirect	2	2022	1423.58
AnkerDirect	3	2022	4054.96
AnkerDirect	4	2022	3267.21
AnkerDirect	5	2022	4160.70
AnkerDirect	6	2022	7493.47
Beach Camera Same Day Shipping	1	2022	553.83
Beach Camera Same Day Shipping	2	2022	1742.99
Beach Camera Same Day Shipping	3	2022	9126.36
Beach Camera Same Day Shipping	4	2022	2576.62
Beach Camera Same Day Shipping	5	2022	1019.48
Beach Camera Same Day Shipping	6	2022	3055.17
Tracfone Wireless Inc.	1	2022	7171.39
Tracfone Wireless Inc.	2	2022	7566.64

seller_id	month_of_year	year	total_net_sales
Tracfone Wireless Inc.	3	2022	5010.96
Tracfone Wireless Inc.	4	2022	4071.86
Tracfone Wireless Inc.	5	2022	5360.19
Tracfone Wireless Inc.	6	2022	3281.56
VARIDESK	2	2022	427.81
VARIDESK	3	2022	518.84
VARIDESK	4	2022	573.26
VARIDESK	5	2022	1021.84
VARIDESK	6	2022	780.44


Total Net Sales Over 2022 – Amazon vs. Top 5 Competitors
$600,000
$500,000
$400,000
$300,000
$200,000
$100,000
$-
Net Sales ($)
Jan
Feb
Mar
Apr
May
Jun
Amazon.com
VARIDESK
AnkerDirect
Tracfone Wireless Inc.
ARF Tech
Beach Camera Same Day Shipping


Total Net Sales Over 2022 – Top 5 Competitors Only
$10,000
$9,000
$8,000
$7,000
$6,000
$5,000
$4,000
$3,000
$2,000
$1,000
$-
Net Sales ($)
Jan
Feb
Mar
Apr
May
Jun



- This analysis is fairly consistent with the previous analysis by year. Amazon's competitors are maintaining high sales consistent with trends in previous years. This graph also gives some insights of peak sales months for Amazon's top competitors. It seems that these companies have seasonal trends where March, May, and June are peak months. Also, these trends in 2022 confirm that VERIDESK has seen significant decreases in net sales and is much less of a threat to Amazon.

Identifying the number of products sold and the average price per unit of the top 5 competitors:

The following query takes the total_net_sales and divides it by the count of the product_id to calculate the avg_revenue_per_unit of a purchase by agencies from each seller.

In [89]:
```
%%sql

SELECT
    s.seller_id,
    COUNT(p.product_id) AS number_of_products_sold,
    COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales,
    COALESCE(SUM(amz.item_net_total), 0) / COUNT(p.product_id) AS avg_revenue_per_unit
FROM
    seller s, product p, amazon amz
WHERE
    s.key = amz.seller_key and
    p.key = amz.product_key
GROUP BY
     s.seller_id
ORDER BY
    total_net_sales DESC
LIMIT 6;
```

```
 * postgresql://student@/Final_Project
6 rows affected.
```

Out[89]:

seller_id	number_of_products_sold	total_net_sales	avg_revenue_per_unit
Amazon.com	214915	17842264.55	83.0200988762999325
VARIDESK	490	242604.77	495.1117755102040816
AnkerDirect	1765	169637.61	96.1119603399433428
Tracfone Wireless Inc.	1059	124844.94	117.8894617563739377
ARF Tech	370	120191.00	324.8405405405405405
Beach Camera Same Day Shipping	200	116226.72	581.1336000000000000

- A key obeservation is that all of Amazon's top competitors have a higher average revenue per product than Amazon does. Most of these companies sell a very small quantity of products at a high price point, for example, Beach Camera Same Day Shipping, AnkerDirect, VARIDESK, and ARF Tech having avg_revenue_per_unit of 581, 495, and 325 respectively. These companies tend to have control over tech items and other high-end goods in which they can sell few at a higher price to generate large sales. Amazon should consider shifting its focus towards the luxury/high end goods markets to compete in sales with these companies. The next question seeks to better understand the specific product lines that these companies are selling.

Results vs. Our Expectations

- In answering this question, we were expecting that all competitor sellers to Amazon would have a significant deficit in terms of the total net sales, but what we were not expecting was a difference in over two orders of magnitude (millions of dollars vs. thousands of dollars). We also expected that these competitors have had a more consistent history of sales since 2017, but to our surprise, VARIDESK, the competitor with the largest total net sales behind Amazon for example, has seen a significant plateau in sales since 2018, while Tracfone Wireless Inc., as another example, has only started selling its products on Amazon since 2020 and already the competitor with the third highest total net sales compared to Amazon. And finally, we were expecting top competitor sellers to Amazon to have a similar average revenue per product to Amazon (around 83 and on the lower end of the spectrum) while selling a large quantity of items. We found that the opposite was true. The competitors with the largest total net sales were actually those that sold more premium, high-end products worth multiple 100s of dollars.

b. Among the top 5 competitor sellers on Amazon.com, what product categories do they predominantly sell? Are they offering discounts that are giving them an advantage in this

process?

Showing the most popular product categories and the total net sales in these categories of the top 5 competitors:

The following query uses the analytic function, “RANK()”, takes and stores the total net sales in all of the product categories, and then ranks the categories by the total net sales from largest to smallest for each seller. After using the “IN” command to specify that we only want to view the results of Amazon and their top competitors, the query shows the product category with the highest total net sales and their respective values in sales.

In [90]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    total_net_sales
FROM
    RankedCategories
WHERE
    category_rank = 1 and
    seller_id IN ('Amazon.com', 'VARIDESK', 'AnkerDirect', 'Tracfone Wireless Inc.', 'ARF Tech', 'Beach Camer
ORDER BY
    total_net_sales DESC
LIMIT 6;
```

```
 * postgresql://student@/Final_Project
6 rows affected.
```

Out[90]:

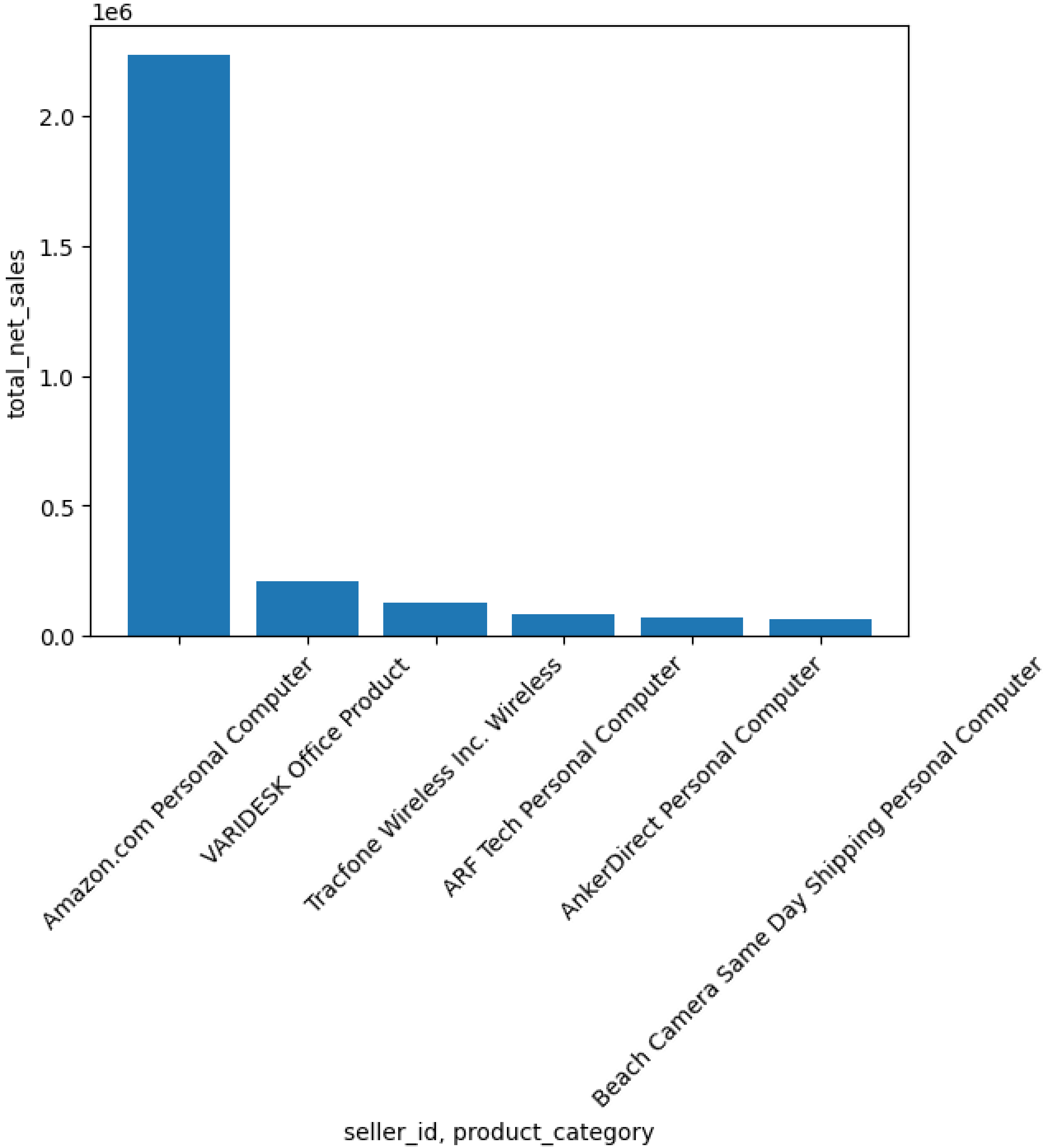
seller_id	product_category	total_net_sales
Amazon.com	Personal Computer	2235100.74
VARIDESK	Office Product	213011.99
Tracfone Wireless Inc.	Wireless	124844.94
ARF Tech	Personal Computer	82303.04
AnkerDirect	Personal Computer	68271.63
Beach Camera Same Day Shipping	Personal Computer	62277.31

In [91]:

```
figsize=(100, 100)

_.bar()
```

Out[91]:

```
<BarContainer object of 6 artists>
```


1e6
2.0
1.5
1.0
0.5
0.0
total_net_sales
Amazon.com Personal Computer
VARIDESK Office Product
Tracfone Wireless Inc. Wireless
ARF Tech Personal Computer
AnkerDirect Personal Computer
Beach Camera Same Day Shipping Personal Computer
seller_id, product_category

- Upon viewing the product category results, for all companies, more than 50% of the total net sales come from purchases of items in their top categories. Therefore, Amazon should focus on targeting these product categories for their top competitors to help diminish competitor sales in these key categories.

- It is important to note that the “Personal Computer” product category is the most competitive among Amazon’s competitors. 3 of the 5 biggest competitors–ARF Tech, AnkerDirect, and Beach Camera Same Day Shipping–have had most of their net sales come from this category of products which poses the largest threat to amazon as this is also the category where Amazon has the largest net sales. Amazon should also consider selling the same or similar top computer models and accessories as these business do in order to provide at least similar options and take some of their sales away. The greater variety of Amazon’s computer products and the more similar they are to alternative company’s product offerings, then Amazon will likely be able to take away some sales. Pairing this strategy with Amazon adding special discounts or advertisements/promotions to similar items to competitors in this product category can help to increase its market share of sales as well. Slightly lower prices or packaged deals for the same products will lead to Amazon becoming the preferred brand. If Amazon can reduce the sales of the “Personal Computer” product category that is key to its competitors' revenues, it will significantly hurt 3 of its competitors’ sales to agencies.

- One final strategic move Amazon can take is to provide same day shipping for as many “Personal Computer” product category items as possible. This is one of the key marketing strategies for the Beach Camera Same Day Shipping seller–it is in their very name. By offering this option, Amazon can take away a key selling point of this competitor’s marketing strategy.

- As Tracfone Wireless Inc. has been an increasingly threatening competitor in its sales of items in the “Wireless” product categories, it is key that Amazon also focus on increasing sales, promotions, discounts, and consumer ratings for their own products in this category. If Amazon can step in before this company becomes a leading seller in the category, it can reduce the risk of losing its market share.

- VARIDESK poses a much smaller threat in the “Office Product” product category as its sales have slowed substantially since 2018. Amazon should not spend too much time with this product category.

Top 5 competitors and their top product categories:

The following queries were written as a follow up to look at the other top 5 product categories in sales for Amazon and each of the top 5 competitors. We also included analysis on the proportion of total sales (%) that falls into each category to see the other types of products they sell. Are there any other product categories that might threaten Amazon's sales?

Amazon:

In [92]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('Amazon.com')
GROUP BY
    seller_id, product_category
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[92]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
Amazon.com	Personal Computer	2235100.74	26.83
Amazon.com	Book	1806364.36	21.68
Amazon.com	Office Product	1574499.08	18.90
Amazon.com	Business, Industrial, & Scientific Supplies Basic	1516241.48	18.20
Amazon.com	CE	1197898.74	14.38

VARIDESK:

In [93]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('VARIDESK')
GROUP BY
    seller_id, product_category
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[93]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
VARIDESK	Office Product	213011.99	87.95
VARIDESK	Furniture	21729.72	8.97
VARIDESK	Kitchen	3922.34	1.62
VARIDESK	Home Improvement	2061.05	0.85
VARIDESK	PC Accessory	1485	0.61

AnkerDirect:

In [94]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('AnkerDirect')
GROUP BY
    seller_id, product_category
```

```
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[94]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
AnkerDirect	Personal Computer	68271.63	41.91
AnkerDirect	Wireless	40174.79	24.66
AnkerDirect	CE	38651.31	23.73
AnkerDirect	Network Media Player	10857.73	6.67
AnkerDirect	Home Theater	4949.19	3.04

Tracfone Wireless Inc.

In [95]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('Tracfone Wireless Inc.')
GROUP BY
    seller_id, product_category
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
1 rows affected.
```

Out[95]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
Tracfone Wireless Inc.	Wireless	124844.94	100.00

ARF Tech:

In [96]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('ARF Tech')
GROUP BY
    seller_id, product_category
```

```
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[96]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
ARF Tech	Personal Computer	82303.04	68.88
ARF Tech	CE	28840.75	24.14
ARF Tech	Wireless	5252.07	4.40
ARF Tech	Photography	1698.79	1.42
ARF Tech	PC Accessory	1396.87	1.17

Beach Camera Same Day Shipping:

In [97]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    SUM(total_net_sales) AS total_net_sales,
    ROUND((SUM(total_net_sales) / SUM(SUM(total_net_sales)) OVER ()) * 100, 2) AS proportion_of_total_net_sal
FROM
    RankedCategories
WHERE
    category_rank <= 5 and
    seller_id IN ('Beach Camera Same Day Shipping')
GROUP BY
    seller_id, product_category
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[97]:

seller_id	product_category	total_net_sales	proportion_of_total_net_sales
Beach Camera Same Day Shipping	Personal Computer	62277.31	57.43
Beach Camera Same Day Shipping	Photography	16954.77	15.64
Beach Camera Same Day Shipping	Home Theater	13124.67	12.10
Beach Camera Same Day Shipping	CE	12644.04	11.66
Beach Camera Same Day Shipping	Wireless	3430.52	3.16

As you can see, Amazon itself sells a wide array of products to agencies with its top product categories as "Personal Computer", "Book", "Office Products", "Business, Industrial, & Scientific Supplies Basic", and "CE". It is good to see that a large proportion of Amazon's sales are towards the "Personal Computer" and "Office Products" categories–the dominant categories of 4 of the 5 top competitors.

An area of concern is that sales of products in the "Wireless" category do not show up in Amazon's top product categories. The "Wireless" product category is the top product category in sales for Tracfone Wireless Inc. comprising 100% of sales, and the second highest product category in over 24% of the sales for AnkerDirect. This emphasizes the importance of Amazon entering the "Wireless" product category market, especially as sales to agencies have boomed for Tracfone Wireless Inc. since 2021. And finally, 4 of the 5 competitors have a siginificant proportion (15-25%) of total net sales for products in the "CE" product category. This is another category of products that Amazon should target in the market.

Results vs. Our Expectations

- We are suprised that so many of the companies have a significant proportion of their total_net_sales under the "Wireless" product category. In conducting this analysis, we expected most companies to dominate their sales in 1 product category. In contrary to our belief, a few of the competitors had a significant proportion of sales in other categories that Amazon needs to be aware of. Particularly, the idea that AnkerDirect's total net sales in its top product category of "Personal Computer" comprised <50% of its total sales was the most surprising.

Average discounts from top competitors by most prominent product category:

This query is similar to the previous one, but in addition to showing the sellers with the total net sales for the product category with the highest overall sales, it also shows the average revenue per unit and the average discount on the unit sold in these categories.

In [98]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        p.product_category,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
        ROUND(COALESCE(SUM(amz.item_net_total), 0) / COUNT(p.product_id), 2) AS avg_revenue_per_unit,
        ROUND(AVG(amz.pricing_discount_dollar), 2) AS avg_discount_on_unit
    FROM
        seller s, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        p.key = amz.product_key
    GROUP BY
        s.seller_id, p.product_category
)
SELECT
    seller_id,
    product_category,
    total_net_sales,
    avg_revenue_per_unit,
    avg_discount_on_unit
FROM
    RankedCategories
WHERE
    category_rank = 1 and
    seller_id IN ('Amazon.com', 'VARIDESK', 'AnkerDirect', 'Tracfone Wireless Inc.', 'ARF Tech', 'Beach Camer
ORDER BY
    total_net_sales DESC
LIMIT 6;
```

```
 * postgresql://student@/Final_Project
6 rows affected.
```

Out[98]:

seller_id	product_category	total_net_sales	avg_revenue_per_unit	avg_discount_on_unit
Amazon.com	Personal Computer	2235100.74	157.51	1.47
VARIDESK	Office Product	213011.99	522.09	0.00
Tracfone Wireless Inc.	Wireless	124844.94	117.89	0.00
ARF Tech	Personal Computer	82303.04	330.53	15.58
AnkerDirect	Personal Computer	68271.63	107.18	0.04
Beach Camera Same Day Shipping	Personal Computer	62277.31	732.67	76.37

- The results give some insight into the marketing strategies of Amazon's competitors– ARF Tech, AnkerDirect, and Beach Camera Same Day Shipping–regarding sales of items in the "Personal Computer" product category. ARF Tech and Beach Camera Same Day Shipping both use significantly high discounts (an average of 4% and 10% off respectively) to lower their prices and attract customers, while Amazon has an average discount of <1% for items in this category. Again, it is important to apply some promotions and discounts for these particular items to provide similar offers to its competitors.

Results vs. Our Expectations

- In analyzing the top product categories of items sold by competitors to agencies, we guessed that the top category would involve some kind of technological products, but we did not expect 3 of the 5 top competitors, along with Amazon, to have the most sales in the "Personal Computer" category. We assumed that each competitor would have found its own niche product category in which Amazon had fewer of its own products on the market so that the competitors could dominate the market share in these categories. Instead, we discovered that Amazon was already facing a lot of competition in the "Personal Computer" product category which is why we recommend that they issue discounts and promotions to become the preferred brand of products in this category. Finally, given that the purchasers are Washington State Agencies, we assumed that most of the competitors would pitch their products with heavy discounts in order to appeal to budget friendly and cost effective products. Besides ARF Tech's and Beach Camera Same Day Shipping's relatively high average discounts per unit, most other competitor sellers did not use discounts as a marketing tactic for their products.

c. What are the agencies that competitor sellers have the most business with?

The agency that spends the most on purchases from each of Amazon's top 5 competitors

The following query uses the analytic function, “RANK()”, in a similar manner to previous queries, except in this code, the SQL code looks for the agency under which each of the top 5 competitors had the highest total net sales.

In [99]:

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        a.agency_id,
        COUNT(p.product_id) AS number_of_products_sold,
        COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales,
        RANK() OVER (PARTITION BY s.seller_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC
    FROM
        seller s, agency a, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        a.key = amz.agency_key and
        p.key = amz.product_key and
        s.seller_id IN ('Amazon.com', 'VARIDESK', 'AnkerDirect', 'Tracfone Wireless Inc.', 'ARF Tech', 'Beach
    GROUP BY
        s.seller_id, a.agency_id
)
SELECT
    seller_id,
    agency_id,
    total_net_sales,
    number_of_products_sold
FROM
    RankedCategories
WHERE
    category_rank = 1
ORDER BY
    total_net_sales DESC
LIMIT 6;
```

```
 * postgresql://student@/Final_Project
6 rows affected.
```

Out[99]:

seller_id	agency_id	total_net_sales	number_of_products_sold
Amazon.com	CHILDREN YOUTH AND FAMILIES DEPT OF	843853.38	8083
Tracfone Wireless Inc.	CHILDREN YOUTH AND FAMILIES DEPT OF	38452.67	427
ARF Tech	SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	10402.14	13
AnkerDirect	AGRICULTURE DEPARTMENT OF	6333.22	13
Beach Camera Same Day Shipping	ENTERPRISE SERVICES DEPARTMENT OF	6016.75	2
VARIDESK	CORRECTIONS DEPARTMENT OF	1442.32	4

- These results aid Amazon in developing a better profile of its top competitors to identify the target agency for their business. As you can see, Amazon's own top agency purchaser is the Dept of Children, Youth, and Families with net sales totaling over 843,000, and the next highest total net sales of any competitor to an agency comes from Tracfone Wireless Inc. who has made almost 40,000 in sales to the same agency. The total net sales given to any individual agency substantially decreases with each competitor. The only other top competitor that has received 10,000 or more from an agency is ARF Tech having its total net sales of about $10,402 from the Department of Social and Health Services (DSHS).

The agencies that spend that most on purchases from all competitors in the dataset:

In [100…

```
%%sql

WITH RankedCategories AS (
    SELECT
        s.seller_id,
        a.agency_id,
        COUNT(p.product_id) AS number_of_products_sold,
        COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) AS total_net_sales,
        ROW_NUMBER() OVER (PARTITION BY a.agency_id ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0
    FROM
        seller s, agency a, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        a.key = amz.agency_key and
        p.key = amz.product_key and
        s.seller_id NOT IN ('Amazon.com')
    GROUP BY
        s.seller_id, a.agency_id
)
SELECT
    agency_id,
    SUM(total_net_sales) AS total_net_sales,
    SUM(number_of_products_sold) AS total_products_sold
FROM
    RankedCategories
WHERE
    category_row_number = 1
GROUP BY
    agency_id
ORDER BY
    total_net_sales DESC
LIMIT 5;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[100]:

agency_id	total_net_sales	total_products_sold
CHILDREN YOUTH AND FAMILIES DEPT OF	38452.67	427
SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	19912.8	4
ATTORNEY GENERAL OFFICE OF THE	17469.10	7
TRANSPORTATION DEPARTMENT OF	16849.55	3
HUMAN RIGHTS COMMISSION	15852.10	2

- It is noteworthy that the Dept of Children, Youth, and Families along with the DSHS are the agencies from which Amazon has the highest total_net_sales, but these are also the agencies that are the #1 and 2 purchasers from all of Amazon's competitors on its platform.

The Top 5 agencies that make purchases from Amazon as the seller:

In [101…

```
%%sql

WITH RankedAgencies AS (
    SELECT
        s.seller_id,
        a.agency_id,
        COALESCE(SUM(amz.item_net_total), 0) AS total_net_sales,
        COUNT(p.product_id) AS number_of_products_sold,
        RANK() OVER (ORDER BY COALESCE(SUM(COALESCE(amz.item_net_total, 0)), 0) DESC) AS agency_rank
    FROM
        seller s, agency a, product p, amazon amz
    WHERE
        s.key = amz.seller_key and
        a.key = amz.agency_key and
        p.key = amz.product_key and
        s.seller_id = 'Amazon.com'
    GROUP BY
        s.seller_id, a.agency_id
)
SELECT
    seller_id,
    agency_id,
    total_net_sales,
    number_of_products_sold
FROM
    RankedAgencies
WHERE
    agency_rank <= 5
ORDER BY
    total_net_sales DESC;
```

```
 * postgresql://student@/Final_Project
5 rows affected.
```

Out[101]:

seller_id	agency_id	total_net_sales	number_of_products_sold
Amazon.com	CHILDREN YOUTH AND FAMILIES DEPT OF	843853.38	8083
Amazon.com	SOCIAL AND HEALTH SERVICES DEPARTMENT OF (DSHS)	474969.27	6131
Amazon.com	CORRECTIONS DEPARTMENT OF	366284.25	2958
Amazon.com	FISH AND WILDLIFE DEPARTMENT OF	120249.77	1584
Amazon.com	NATURAL RESOURCES DEPARTMENT OF	116342.59	903

- This query takes a look at the top 5 agencies that purchase the most from Amazon. The total net sales for each of these individual agencies is significant. Again, it is noteworthy that the Dept of Children, Youth, and Families along with the DSHS are the agencies from which Amazon has the highest total_net_sales, but these are also the agencies that are the #1 purchasers from top competitors Tracfone Wireless Inc. and ARF Tech. The Department of Corrections is another agency that VARIDESK, another top competitor, has the highest total net sales for. This agency is the third largest purchaser from Amazon.

- Above all other agencies, it is key for Amazon to target the Dept of Children, Youth, and Families and the DSHS. 2 of Amazon's top 5 competitors have these agencies as their largest purchasers, and all competing sellers in the data set have the highest sales with these agencies. If Amazon could target promotions and discounts to the most popular items sold to these agencies by its competitors (by referencing its competitors' product categories with the highest net sales of its from the previous analysis), or if they could orchestrate specialized deals or discounts with these agencies only for Amazon branded products, it would eliminate much of the competition in sales to these critical agency buyers.

Results vs. Our Expectations

- In conducting this analysis, we thought that there would not be much significance in the agencies that are the top purchasers from Amazon and its competitor sellers because over half of the records contain a null under the agency_id in this data set. However, it is interesting to find that almost 1.5 million of Amazon's total net sales of roughly 18 million can be linked to just two agencies: Dept of Children, Youth, and Families, and the DSHS. When it comes down to targeting the highest purchasing agencies and beating out competitors to agencies for which they have the highest total net sales, the agencies tend to overlap much more than we thought. Because Amazon already has a presence in the market in similar product categories and with many of the same agency buyers, it will be easier to target their competitors with superior product discounts, promotions, and offerings.

Project Conclusion

1. From our analyses we can conclude that Amazon can tailor marketing efforts to coincide with high-expenditure months and incentivizing purchases during traditionally low-spending periods with discounts.

2. By identifying trends in recurring and bulk orders from top-spending agencies, Amazon can better manage inventory and predict demand.

3. Understanding customer attrition enables Amazon to develop re-engagement strategies for agencies that have ceased orders.

4. Analyzing the performance and offerings of competitor sellers allows Amazon to devise specific promotions and discounts, undercutting competitors in key product categories thereby increasing its market share.