Optimising Retail Company Revenue Using SQL

By Bheemeswararao Aika

Analyze product data for an online sports retail company to optimize revenue.

Objective

This project aims to enhance the revenue of an online retail company by providing actionable recommendations for its marketing and sales teams. It involves analyzing a variety of data types, including numeric, string, and timestamp data related to pricing, revenue, ratings, reviews, and product descriptions.

Additionally, website traffic data will be examined to identify trends and opportunities for improvement. The insights gained will inform strategies to optimize marketing efforts and boost sales performance. Ultimately, the project seeks to drive growth and increase profitability for the company.

SQL Function

SQL functions that I used in this project:

Aggregate Functions
These functions enable the calculation of values across multiple rows, including operations such as sum, average, and count.
Date Manipulation Functions
These functions provide the means to manipulate and extract information from date columns within the database.
Join Functions
These functions facilitate the merging of data from multiple tables based on common columns.
Union Functions
These functions enable the consolidation of results from multiple queries into a single result set.

CTE (Common Table Expression)

This function allows the creation of a temporary result set within a SELECT statement.

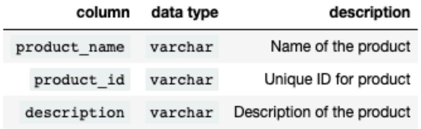
Subquery
These are SELECT statements executed within another SELECT statement.

Data Overview

As a product analyst of an online sports clothing company. The company is specifically interested in how it can improve revenue. We will dive into product data such as pricing, reviews, descriptions, and ratings, as well as revenue and website traffic, to produce recommendations for its marketing and sales teams.

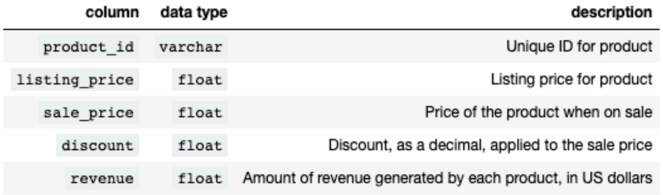
The database provided to us, sports, contains five tables, with product_id being the primary key for all of them.

`info`



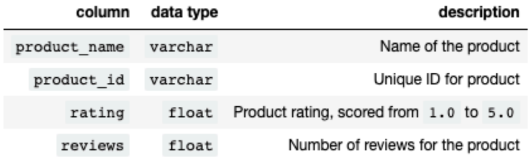
column	data type	description
`product_name`	`varchar`	Name of the product
`product_id`	`varchar`	Unique ID for product
`description`	`varchar`	Description of the product

`finance`



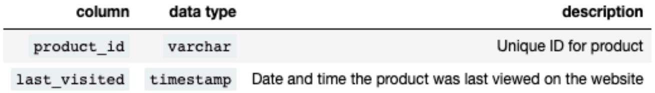
column	data type	description
`product_id`	`varchar`	Unique ID for product
`listing_price`	`float`	Listing price for product
`sale_price`	`float`	Price of the product when on sale
`discount`	`float`	Discount, as a decimal, applied to the sale price
`revenue`	`float`	Amount of revenue generated by each product, in US dollars

`reviews`



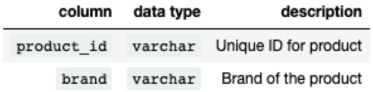
column	data type	description
`product_name`	`varchar`	Name of the product
`product_id`	`varchar`	Unique ID for product
`rating`	`float`	Product rating, scored from `1.0` to `5.0`
`reviews`	`float`	Number of reviews for the product

`traffic`



column	data type	description
`product_id`	`varchar`	Unique ID for product
`last_visited`	`timestamp`	Date and time the product was last viewed on the website

`brands`



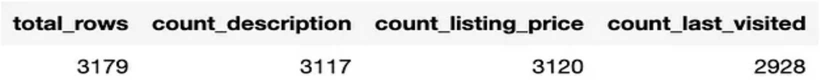
column	data type	description
`product_id`	`varchar`	Unique ID for product
`brand`	`varchar`	Brand of the product

Analysis and Insight

1 Counting the Missing Values

```
SELECT count(*) as total_rows,

count(i.description) as count_description,

   count(f.listing_price) as count_listing_price,

   count(t.last_visited) as count_last_visited

FROM info i

JOIN finance f

ON i.product_id = f.product_id

JOIN traffic t

ON i.product_id = t.product_id;
```

Results



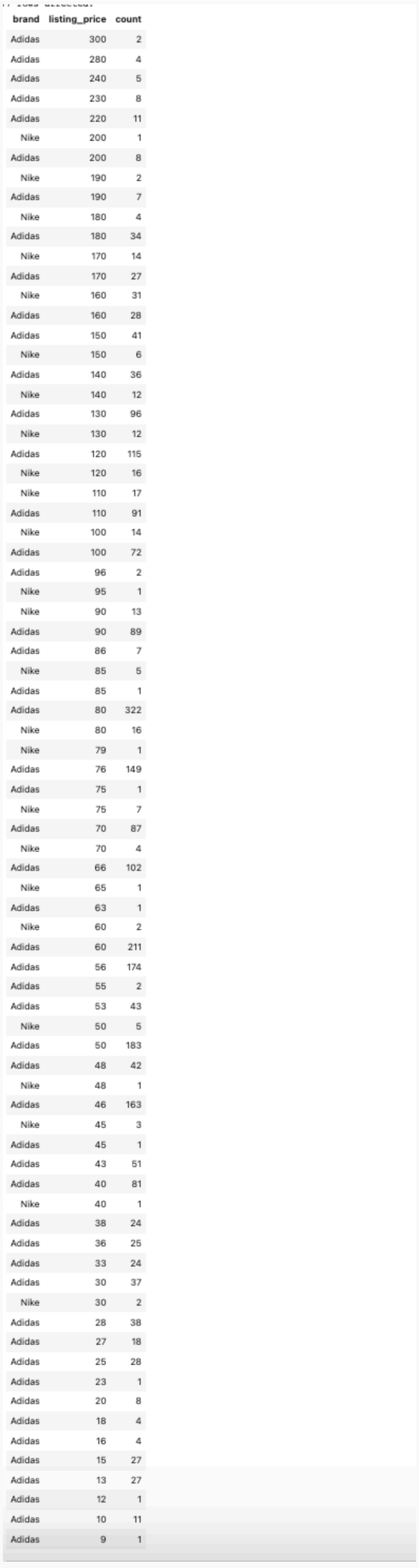
total_rows	count_description	count_listing_price	count_last_visited
3179	3117	3120	2928

We can see the database contains 3,179 products in total. Of the columns we previewed, only one — last_visited — is missing more than five percent of its values. Now let's turn our attention to pricing.

2 Nike vs Adidas pricing

How do the price points of Nike and Adidas products differ? This question helps to build a picture of the company's stock range and customer market.

```
SELECT b.brand,

      CAST(listing_price AS Integer) as listing_price,

      COUNT(f.*)

FROM brands b

JOIN finance f

ON b.product_id = f.product_id

WHERE f.listing_price > 0

GROUP BY b.brand, listing_price

ORDER BY listing_price DESC;
```



```
77 rows affected.
```

brand	listing_price	count
Adidas	300	2
Adidas	280	4
Adidas	240	5
Adidas	230	8
Adidas	220	11
Nike	200	1
Adidas	200	8
Nike	190	2
Adidas	190	7
Nike	180	4
Adidas	180	34
Nike	170	14
Adidas	170	27
Nike	160	31
Adidas	160	28
Adidas	150	41
Nike	150	6
Adidas	140	36
Nike	140	12
Adidas	130	96
Nike	130	12
Adidas	120	115
Nike	120	16
Nike	110	17
Adidas	110	91
Nike	100	14
Adidas	100	72
Adidas	96	2
Nike	95	1
Nike	90	13
Adidas	90	89
Adidas	86	7
Nike	85	5
Adidas	85	1
Adidas	80	322
Nike	80	16
Nike	79	1
Adidas	76	149
Adidas	75	1
Nike	75	7
Adidas	70	87
Nike	70	4
Adidas	66	102
Nike	65	1
Adidas	63	1
Nike	60	2
Adidas	60	211
Adidas	56	174
Adidas	55	2
Adidas	53	43
Nike	50	5
Adidas	50	183
Adidas	48	42
Nike	48	1
Adidas	46	163
Nike	45	3
Adidas	45	1
Adidas	43	51
Adidas	40	81
Nike	40	1
Adidas	38	24
Adidas	36	25
Adidas	33	24
Adidas	30	37
Nike	30	2
Adidas	28	38
Adidas	27	18
Adidas	25	28
Adidas	23	1
Adidas	20	8
Adidas	18	4
Adidas	16	4
Adidas	15	27
Adidas	13	27
Adidas	12	1
Adidas	10	11
Adidas	9	1

It turns out there are 77 unique prices for the products in our database.

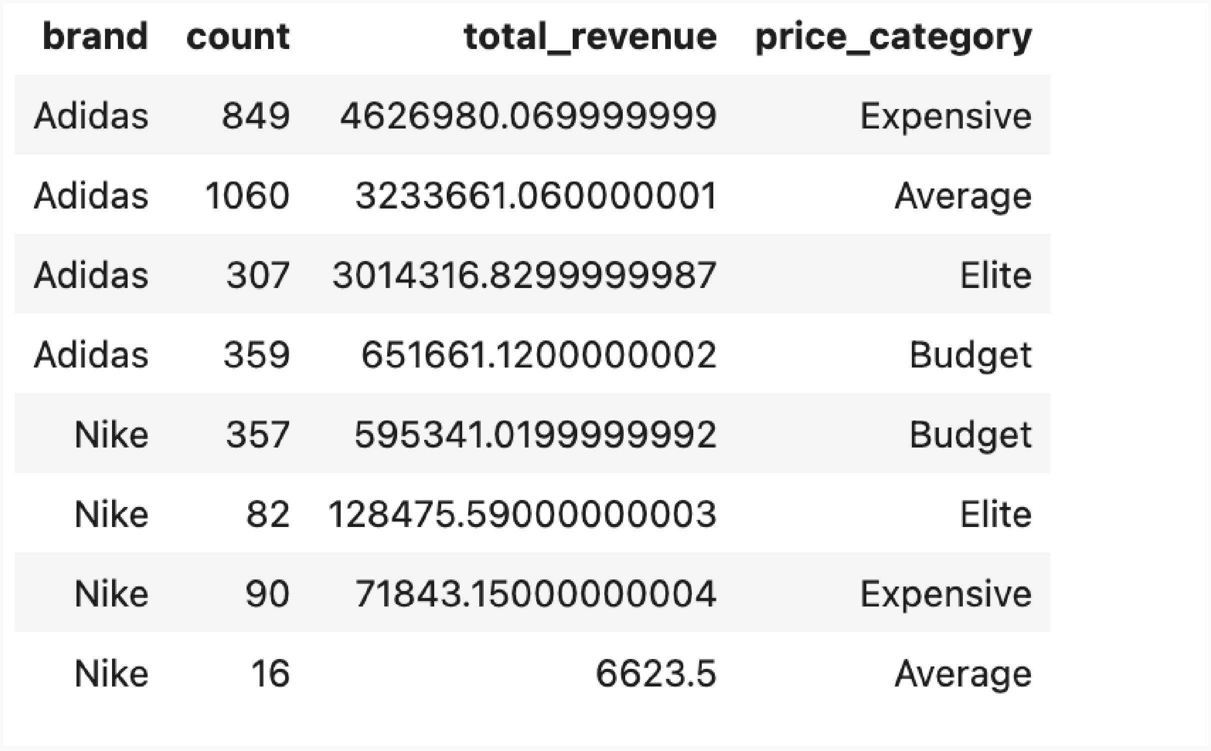
3 Labeling price ranges

Assigning labels to different price ranges by brand and its revenue

```
SELECT b.brand,
    COUNT(f.*),
    SUM(f.revenue) as total_revenue,
    CASE WHEN f.listing_price < 42 THEN 'Budget'
         WHEN f.listing_price >= 42 AND f.listing_price < 74 THEN 'Average'
         WHEN f.listing_price >= 74 AND f.listing_price < 129 THEN 'Expensive'
        ELSE 'Elite'
    END AS price_category
FROM brands b
JOIN finance f
ON b.product_id = f.product_id
WHERE b.brand IS NOT NULL
```

```
GROUP BY b.brand, price_category
```

```
ORDER BY total_revenue DESC;
```



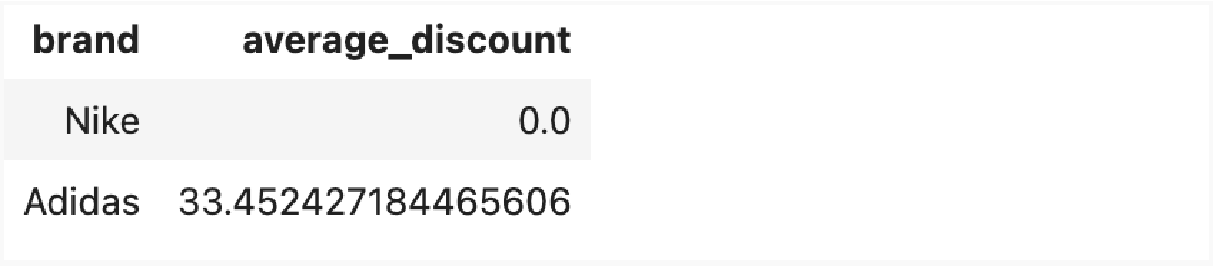
brand	count	total_revenue	price_category
Adidas	849	4626980.069999999	Expensive
Adidas	1060	3233661.060000001	Average
Adidas	307	3014316.8299999987	Elite
Adidas	359	651661.1200000002	Budget
Nike	357	595341.0199999992	Budget
Nike	82	128475.59000000003	Elite
Nike	90	71843.15000000004	Expensive
Nike	16	6623.5	Average

Adidas items generate more total revenue regardless of price category! Specifically, "Elite" Adidas products priced $129 or more typically generate the highest revenue, so the company can potentially increase revenue by shifting their stock to have a larger proportion of these products!

4 Average discount by brand

To understand revenue better, let's take a look at the discount, which is the percent reduction in the listing_price when the product is actually sold. We would like to know whether there is a difference in the amount of discount offered between brands, as this could be influencing revenue.

```
SELECT b.brand,
    AVG(discount)*100 as average_discount

FROM brands b

JOIN finance f

ON b.product_id = f.product_id

WHERE b.brand IS NOT NULL

GROUP BY b.brand;
```



brand	average_discount
Nike	0.0
Adidas	33.452427184465606

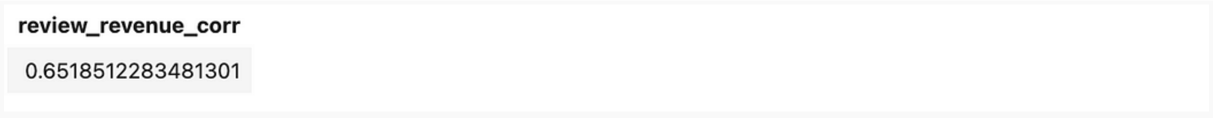
Strangely, no discount is offered on Nike products! In comparison, not only do Adidas products generate the most revenue, but these products are also heavily discounted!

To improve revenue further, the company could try to reduce the amount of discount offered on Adidas products, and monitor sales volume to see if it remains stable. Alternatively, it could try offering a small discount on Nike products. This would reduce average revenue for these products, but may increase revenue overall if there is an increase in the volume of Nike products sold.

5 Correlation between revenue and reviews

Now explore whether relationships exist between the columns in our database. We will check the strength and direction of a correlation between revenue and reviews.

```
SELECT CORR(f.revenue, r.reviews) AS review_revenue_corr
FROM finance f
JOIN reviews r
ON f.product_id = r.product_id;
```



```
review_revenue_corr
0.6518512283481301
```

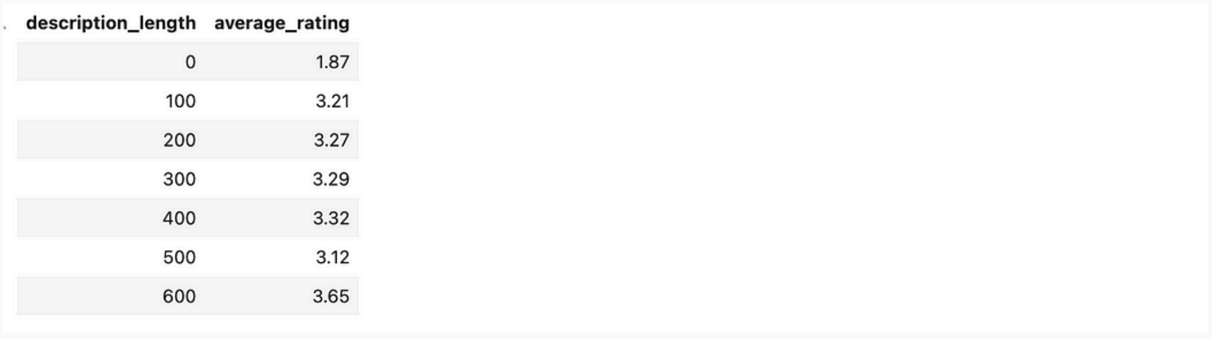
Interestingly, there is a strong positive correlation between revenue and reviews. This means, potentially, if we can get more reviews on the company's website, it may increase sales of those items with a larger number of reviews.

6 Ratings and reviews by product description length

Perhaps the length of a product’s description might influence a product's rating and reviews — if so, the company can produce content guidelines for listing products on their website and test if this influences revenue. Let's check this out!

```
SELECT TRUNC(Length(description)/100.0) *100 as description_length,
    ROUND(AVG(CAST(rating AS numeric)),2) as  average_rating
FROM info i
JOIN reviews r
```

```
ON i.product_id = r.product_id

WHERE description IS NOT NULL

GROUP BY description_length

ORDER BY description_length;
```



description_length	average_rating
0	1.87
100	3.21
200	3.27
300	3.29
400	3.32
500	3.12
600	3.65

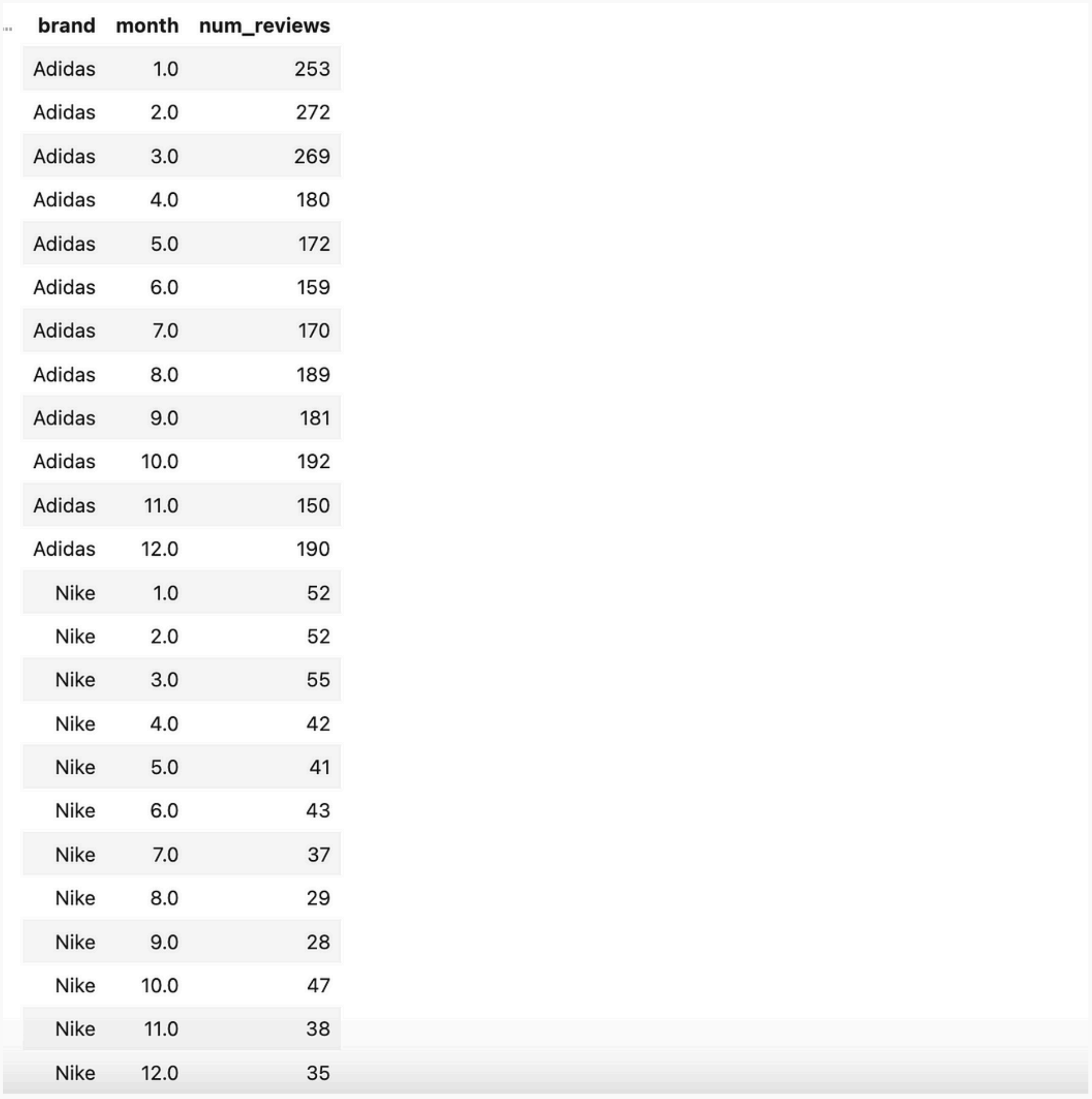
Unfortunately, there doesn’t appear to be a clear pattern between the length of a product’s description and its rating

7. Reviews by month and brand

As we know a correlation exists between reviews and revenue, one approach the company could take is to run experiments with different sales processes encouraging more reviews from customers about their purchases, such as by offering a small discount on future purchases.

Let’s take a look at the volume of reviews by month to see if there are any trends or gaps we can look to exploit.

```
SELECT b.brand,

    DATE_PART('month', last_visited) as month,
    COUNT(r.*) as num_reviews
FROM brands b
JOIN traffic t
ON b.product_id = t.product_id
JOIN reviews r
ON r.product_id = t.product_id
WHERE b.brand IS NOT NULL
GROUP BY b.brand, DATE_PART('month', last_visited)
HAVING DATE_PART('month', last_visited) IS NOT NULL
ORDER BY b.brand, DATE_PART('month', last_visited);
```



brand	month	num_reviews
Adidas	1.0	253
Adidas	2.0	272
Adidas	3.0	269
Adidas	4.0	180
Adidas	5.0	172
Adidas	6.0	159
Adidas	7.0	170
Adidas	8.0	189
Adidas	9.0	181
Adidas	10.0	192
Adidas	11.0	150
Adidas	12.0	190
Nike	1.0	52
Nike	2.0	52
Nike	3.0	55
Nike	4.0	42
Nike	5.0	41
Nike	6.0	43
Nike	7.0	37
Nike	8.0	29
Nike	9.0	28
Nike	10.0	47
Nike	11.0	38
Nike	12.0	35

Looks like product reviews are highest in the first quarter of the calendar year, so there is scope to run experiments aiming to increase the volume of reviews in the other nine months!

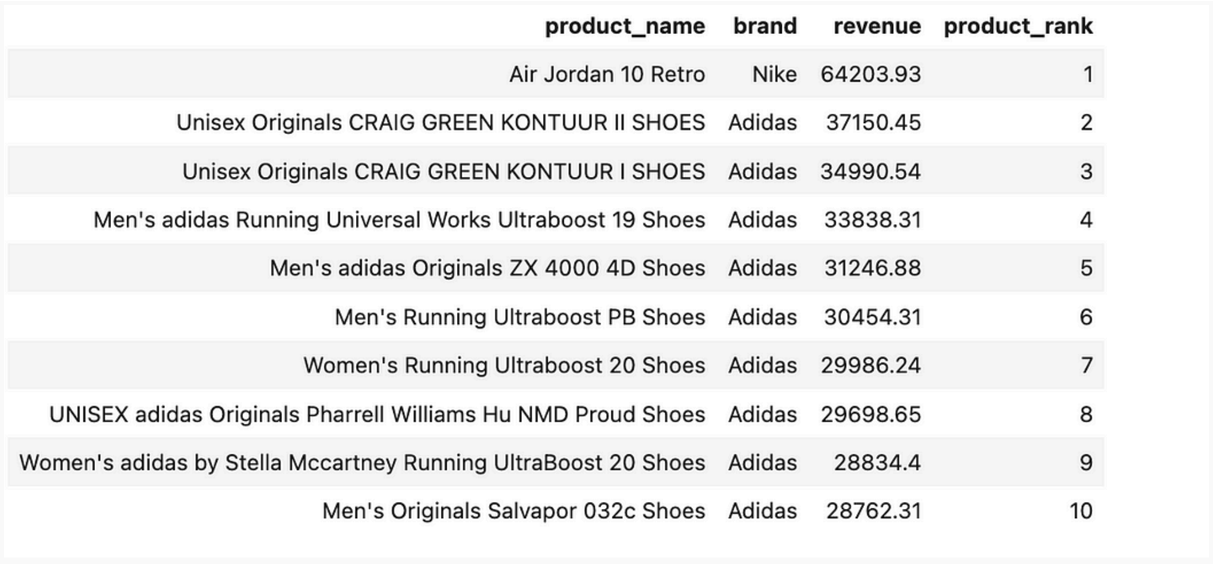
8 Top Revenue Generated Products with Brands

So far, we have been primarily analyzing Adidas vs Nike products. Now, let’s switch our attention to the top revenue generated products.

```
WITH highest_revenue_product AS
```

```
(
  SELECT i.product_name,
         b.brand,
         revenue
  FROM finance f
  JOIN info i
  ON f.product_id = i.product_id
  JOIN brands b
  ON b.product_id = i.product_id
  WHERE product_name IS NOT NULL
    AND revenue IS NOT NULL
    AND brand IS NOT NULL
)
```

```
SELECT product_name,

    brand,

    revenue,

     RANK() OVER (ORDER BY revenue DESC) AS product_rank

FROM highest_revenue_product

LIMIT 10;
```



product_name	brand	revenue	product_rank
Air Jordan 10 Retro	Nike	64203.93	1
Unisex Originals CRAIG GREEN KONTUUR II SHOES	Adidas	37150.45	2
Unisex Originals CRAIG GREEN KONTUUR I SHOES	Adidas	34990.54	3
Men's adidas Running Universal Works Ultraboost 19 Shoes	Adidas	33838.31	4
Men's adidas Originals ZX 4000 4D Shoes	Adidas	31246.88	5
Men's Running Ultraboost PB Shoes	Adidas	30454.31	6
Women's Running Ultraboost 20 Shoes	Adidas	29986.24	7
UNISEX adidas Originals Pharrell Williams Hu NMD Proud Shoes	Adidas	29698.65	8
Women's adidas by Stella Mccartney Running UltraBoost 20 Shoes	Adidas	28834.4	9
Men's Originals Salvapor 032c Shoes	Adidas	28762.31	10

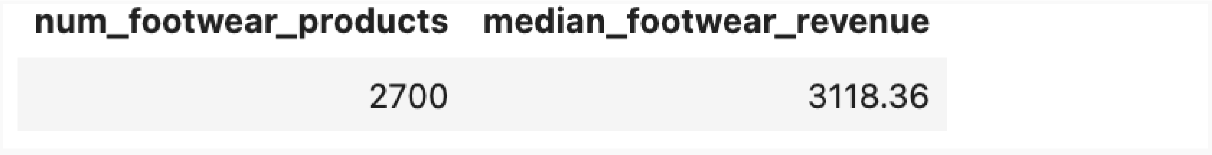
Highest revenue generated by Nike brand which is around $64k, when closely look into top revenue products, the shoes product generating highest revenue. Let’s check the footwear product performance.

9 Footwear product performance

Lets find out how much of the company’s stock consists of footwear products and the median revenue generated by these items.

```
with footwear AS

( SELECT i.description,

       f.revenue

 FROM info i

 INNER JOIN finance f

 ON i.product_id = f.product_id

 WHERE i.description ILIKE '%shoe%'

      OR i.description ILIKE '%trainer%'

      OR i.description ILIKE '%foot%'

      AND i.description IS NOT NULL

)

select COUNT(*) as num_footwear_products,

     percentile_disc(0.5) WITHIN GROUP(ORDER BY revenue) as median_footwear_revenue
```

```
FROM footwear;
```



num_footwear_products	median_footwear_revenue
2700	3118.36

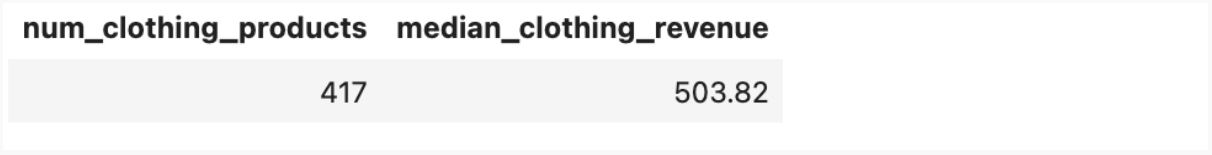
We found there are 3,117 products without missing values for description. Of those, 2,700 are footwear products, which accounts for around 85% of the company's stock. They also generate a median revenue of over $3000 dollars!

This is interesting, but we have no point of reference for whether footwear's median_revenue is good or bad compared to other products. So, let's examine how this differs to clothing products.

10 Clothing product performance

We will re-use footwear, adding a filter afterward to count the number of products and median_revenue of products that are not in footwear.

```
with footwear AS

( SELECT i.description,

       f.revenue

 FROM info i

 INNER JOIN finance f

 ON i.product_id = f.product_id
 WHERE i.description ILIKE '%shoe%'

      OR i.description ILIKE '%trainer%'

      OR i.description ILIKE '%foot%'

      AND i.description IS NOT NULL

)

select COUNT(i.*) as num_clothing_products,

     percentile_disc(0.5) WITHIN GROUP(ORDER BY revenue) as
median_clothing_revenue

FROM info i

INNER JOIN finance f

ON i.product_id = f.product_id

WHERE i.description NOT IN (select description from footwear);
```



num_clothing_products	median_clothing_revenue
417	503.82

The results indicate the presence of 417 clothing products in the analyzed dataset. The median revenue of these clothing products is $503.82. This means that half of the clothing products have revenue above $503.82, and the other half have revenue below that figure.

Conclusion :

1. The brand needs to explore opportunities to develop products in the “Expensive” and “Elite” categories that have higher revenue potential.
2. Also, highest revenue generated Products are from footwear section, brand should focus on giving less discounts on footwear and more discounts on clothing that will increase sales and revenue for clothing section as well.
3. Continuously monitoring product section like footwear and clothing and making relevant price adjustments or marketing strategies.
4. Focusing on product quality, customer service, and holistic marketing strategies can help improve reviews and revenue.
5. Analyzing factors that influence monthly review fluctuations and planning appropriate marketing strategies.
6. Using this data as a foundation to design more effective and customer-oriented business strategies.
7. All of these recommendations can assist the brand in enhancing product performance, increasing revenue, and providing a better experience to customers.

Suggestions

To enhance revenue and customer experience, the brand should consider the following strategies:

1. **Product Development:** Explore opportunities in the "Expensive" and "Elite" categories to tap into higher revenue potential.
2. **Pricing Strategy:** Focus on reducing discounts for footwear, which generates the highest revenue, while offering more discounts on clothing to boost sales in that category.
3. **Continuous Monitoring:** Regularly assess both footwear and clothing sections for pricing adjustments and marketing strategies based on performance data.
4. **Quality and Service:** Prioritize product quality and customer service alongside holistic marketing efforts to improve reviews and overall revenue.
5. **Review Analysis:** Investigate the factors influencing monthly review changes and adjust marketing strategies accordingly.
6. **Data-Driven Strategies:** Utilize collected data to design effective, customer-focused business strategies.

Implementing these recommendations can significantly improve product performance, increase revenue, and enhance the customer experience.