PROJECT NAME :- Analyse product data for an online sports retail company to optimize revenue

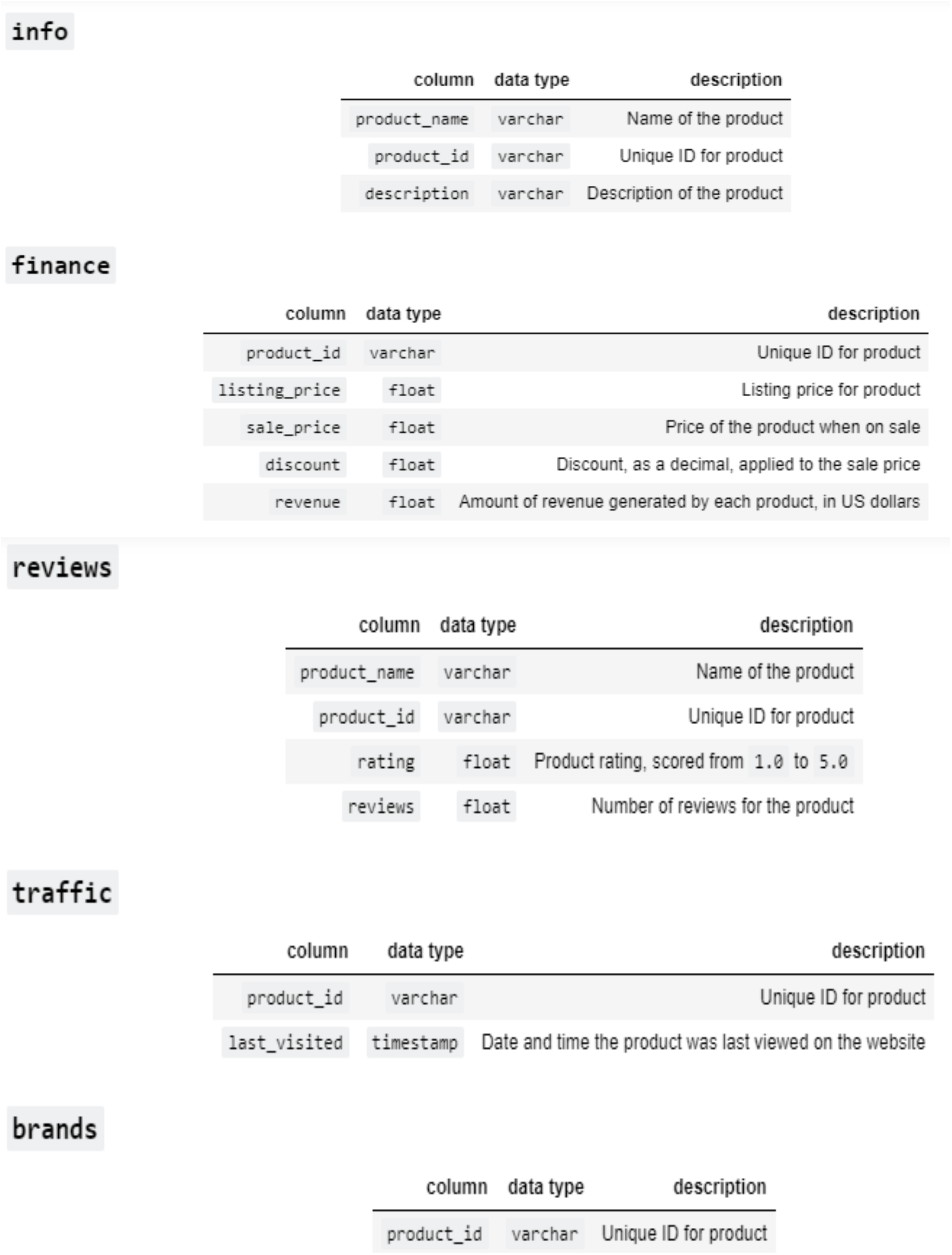
Project Description

Sports clothing is a booming sector!
In this notebook, I will use my SQL skills to analyse product data for an online sports retail company.
In this project, I'll will work with **numeric, string, and timestamp data** on pricing and revenue, ratings, reviews, descriptions, and website traffic.
And I will be using techniques such as **aggregation**, **cleaning**, **labelling**, **Common Table Expressions(CTE),** and **correlation** to produce recommendations on how the company can maximize revenue.

Task List

1. Count the total number of products, along with the number of non-missing values in description, listing price, and last visited
2. Find out how listing price varies between Adidas and Nike products.
3. Create labels for products grouped by price range and brand.
4. Calculate the correlation between reviews and revenue.
5. Split description into bins in increments of one hundred characters, and calculate average rating by for each bin.
6. Count the number of reviews per brand per month.
7. Create the footwear CTE, then calculate the number of products and average revenue from these items.
8. Copy the code used to create footwear then use a filter to return only products that are not in the CTE.

The sports clothing and athleisure industry is a massive market, valued at approximately $193 billion in 2021, with strong growth projections for the coming decade. In this notebook, I will be playing the role of a product analyst for an online sports clothing company. The company is specifically interested in how it can improve revenue. I 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

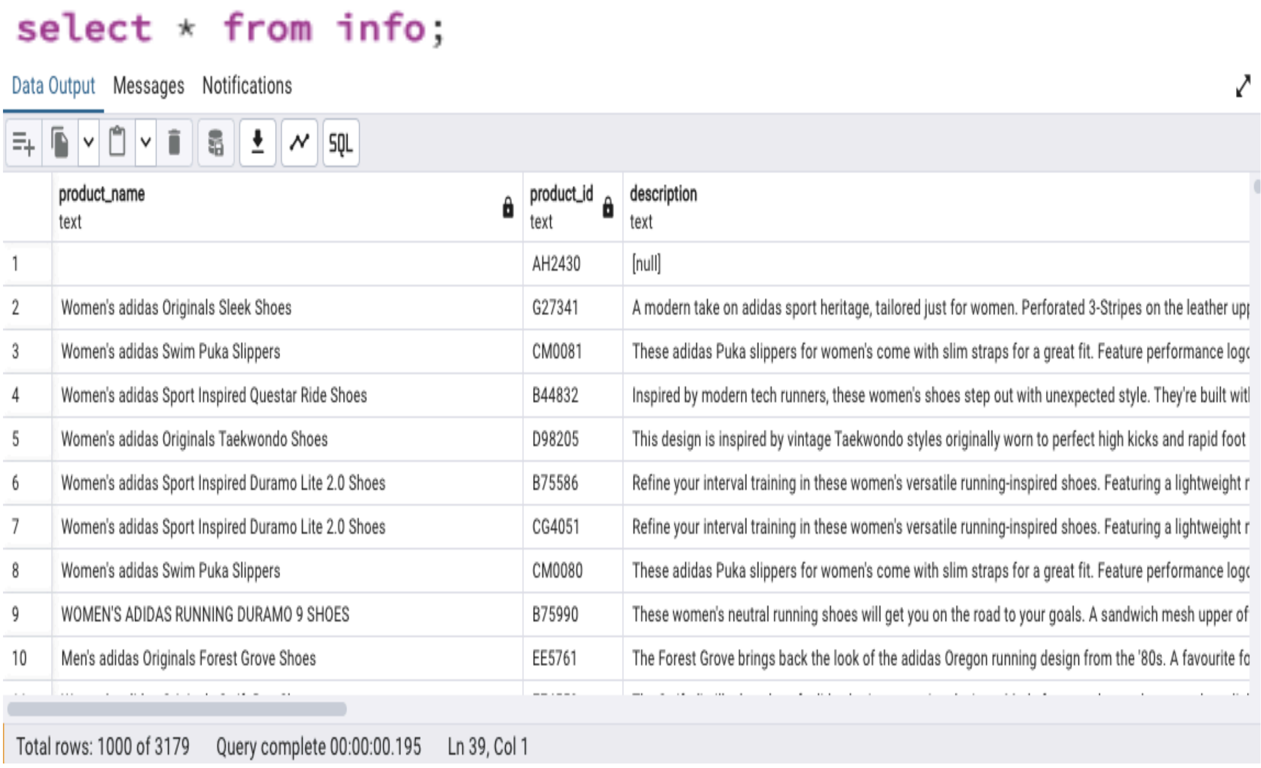
column	data type	description
product_id	varchar	Unique ID for product
last_visited	timestamp	Date and time the product was last viewed on the website

brands

column	data type	description
product_id	varchar	Unique ID for product

The tables have following columns and rows when the “**select**” command is executed.

1)

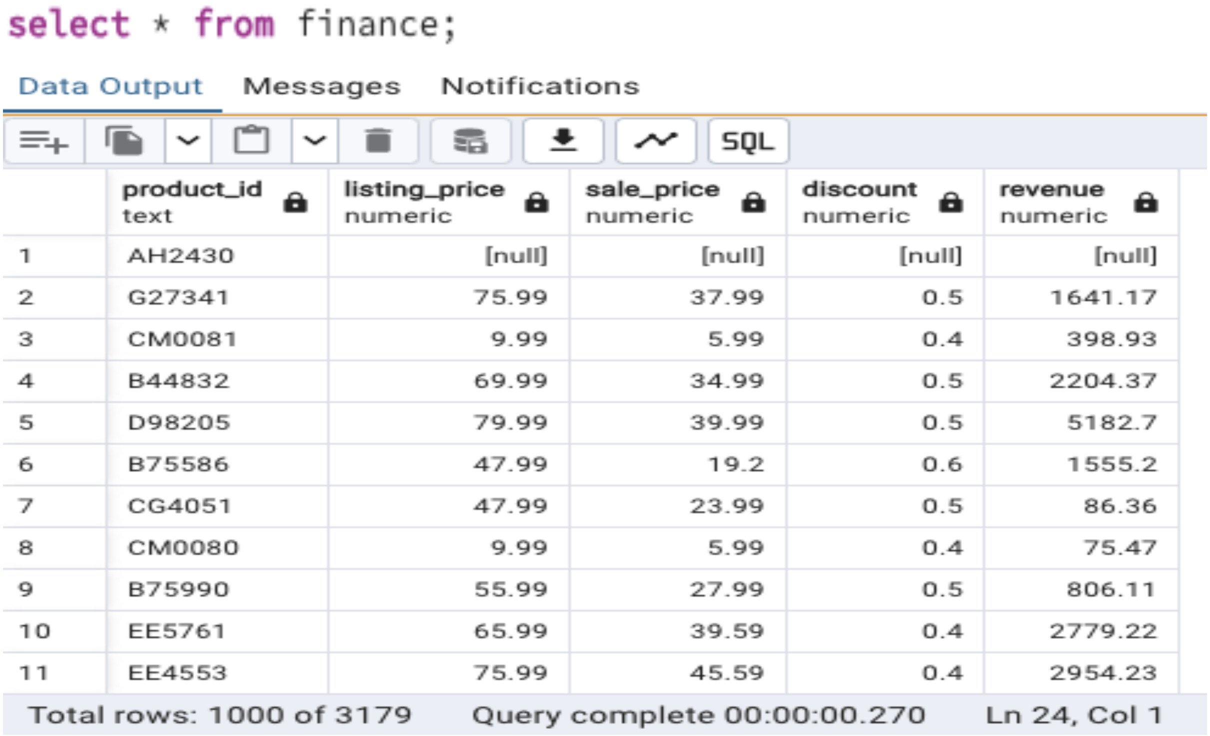


```
select * from info;
```

	product_name text	product_id text	description text
1		AH2430	[null]
2	Women's adidas Originals Sleek Shoes	G27341	A modern take on adidas sport heritage, tailored just for women. Perforated 3-Stripes on the leather upp
3	Women's adidas Swim Puka Slippers	CM0081	These adidas Puka slippers for women's come with slim straps for a great fit. Feature performance logo
4	Women's adidas Sport Inspired Questar Ride Shoes	B44832	Inspired by modern tech runners, these women's shoes step out with unexpected style. They're built witl
5	Women's adidas Originals Taekwondo Shoes	D98205	This design is inspired by vintage Taekwondo styles originally worn to perfect high kicks and rapid foot
6	Women's adidas Sport Inspired Duramo Lite 2.0 Shoes	B75586	Refine your interval training in these women's versatile running-inspired shoes. Featuring a lightweight r
7	Women's adidas Sport Inspired Duramo Lite 2.0 Shoes	CG4051	Refine your interval training in these women's versatile running-inspired shoes. Featuring a lightweight r
8	Women's adidas Swim Puka Slippers	CM0080	These adidas Puka slippers for women's come with slim straps for a great fit. Feature performance logo
9	WOMEN'S ADIDAS RUNNING DURAMO 9 SHOES	B75990	These women's neutral running shoes will get you on the road to your goals. A sandwich mesh upper of
10	Men's adidas Originals Forest Grove Shoes	EE5761	The Forest Grove brings back the look of the adidas Oregon running design from the '80s. A favourite fo

Total rows: 1000 of 3179   Query complete 00:00:00.195   Ln 39, Col 1

2)

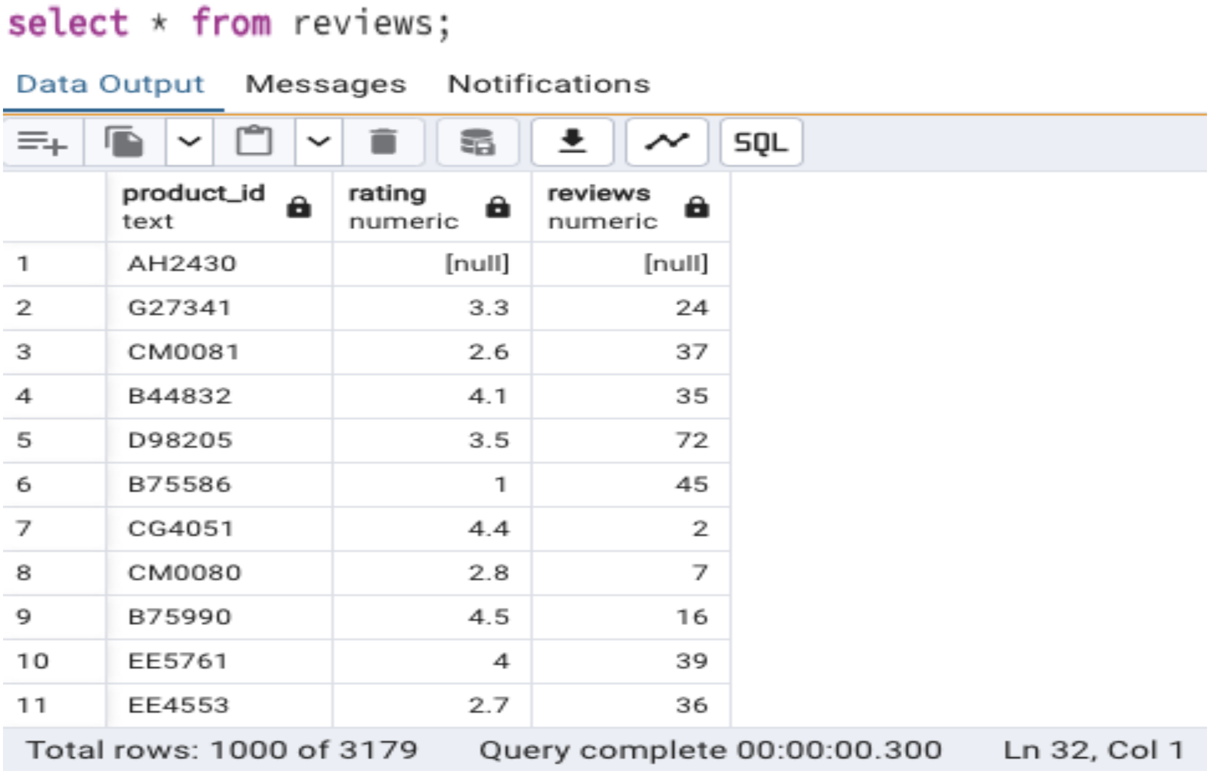


```
select * from finance;
```

	product_id text	listing_price numeric	sale_price numeric	discount numeric	revenue numeric
1	AH2430	[null]	[null]	[null]	[null]
2	G27341	75.99	37.99	0.5	1641.17
3	CM0081	9.99	5.99	0.4	398.93
4	B44832	69.99	34.99	0.5	2204.37
5	D98205	79.99	39.99	0.5	5182.7
6	B75586	47.99	19.2	0.6	1555.2
7	CG4051	47.99	23.99	0.5	86.36
8	CM0080	9.99	5.99	0.4	75.47
9	B75990	55.99	27.99	0.5	806.11
10	EE5761	65.99	39.59	0.4	2779.22
11	EE4553	75.99	45.59	0.4	2954.23

Total rows: 1000 of 3179   Query complete 00:00:00.270   Ln 24, Col 1

3)

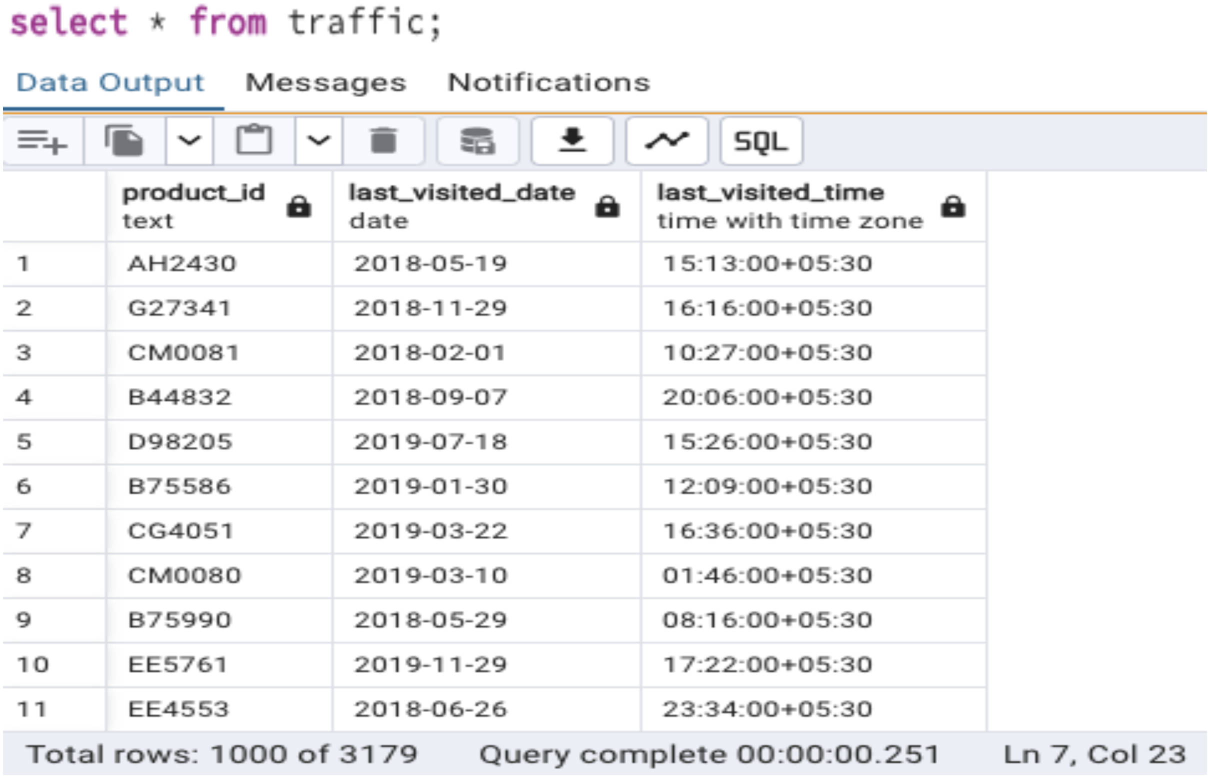


```sql
select * from reviews;
```

	product_id text	rating numeric	reviews numeric
1	AH2430	[null]	[null]
2	G27341	3.3	24
3	CM0081	2.6	37
4	B44832	4.1	35
5	D98205	3.5	72
6	B75586	1	45
7	CG4051	4.4	2
8	CM0080	2.8	7
9	B75990	4.5	16
10	EE5761	4	39
11	EE4553	2.7	36

Total rows: 1000 of 3179   Query complete 00:00:00.300   Ln 32, Col 1

4)

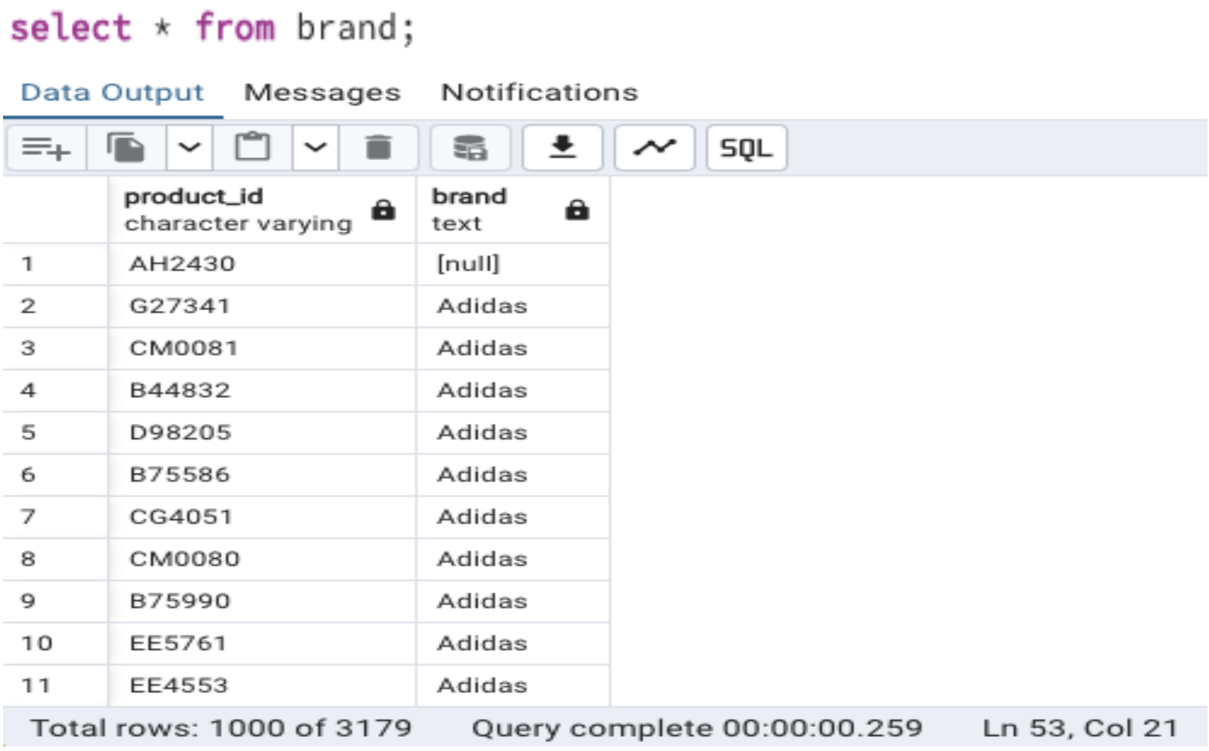


```sql
select * from traffic;
```

	product_id text	last_visited_date date	last_visited_time time with time zone
1	AH2430	2018-05-19	15:13:00+05:30
2	G27341	2018-11-29	16:16:00+05:30
3	CM0081	2018-02-01	10:27:00+05:30
4	B44832	2018-09-07	20:06:00+05:30
5	D98205	2019-07-18	15:26:00+05:30
6	B75586	2019-01-30	12:09:00+05:30
7	CG4051	2019-03-22	16:36:00+05:30
8	CM0080	2019-03-10	01:46:00+05:30
9	B75990	2018-05-29	08:16:00+05:30
10	EE5761	2019-11-29	17:22:00+05:30
11	EE4553	2018-06-26	23:34:00+05:30

Total rows: 1000 of 3179   Query complete 00:00:00.251   Ln 7, Col 23

5)



```
select * from brand;
```

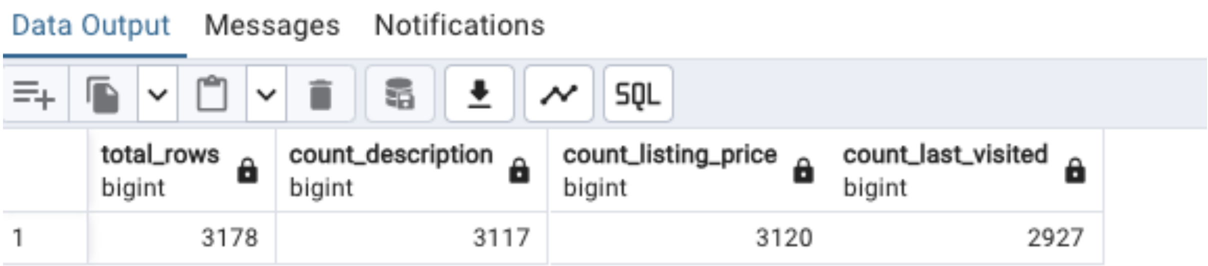
	product_id character varying	brand text
1	AH2430	[null]
2	G27341	Adidas
3	CM0081	Adidas
4	B44832	Adidas
5	D98205	Adidas
6	B75586	Adidas
7	CG4051	Adidas
8	CM0080	Adidas
9	B75990	Adidas
10	EE5761	Adidas
11	EE4553	Adidas

```
Total rows: 1000 of 3179     Query complete 00:00:00.259     Ln 53, Col 21
```

Now let’s start working on the “Task Lists”.

1. Counting missing values

```
--1)
SELECT COUNT(info) as total_rows,
COUNT(info.description) as count_description, COUNT(finance.listing_price) as count_listing_price,
COUNT(traffic.last_visited_date) as count_last_visited
FROM info
INNER JOIN traffic
ON traffic.product_id = info.product_id
INNEr JOIN finance
ON finance.product_id = info.product_id;
```

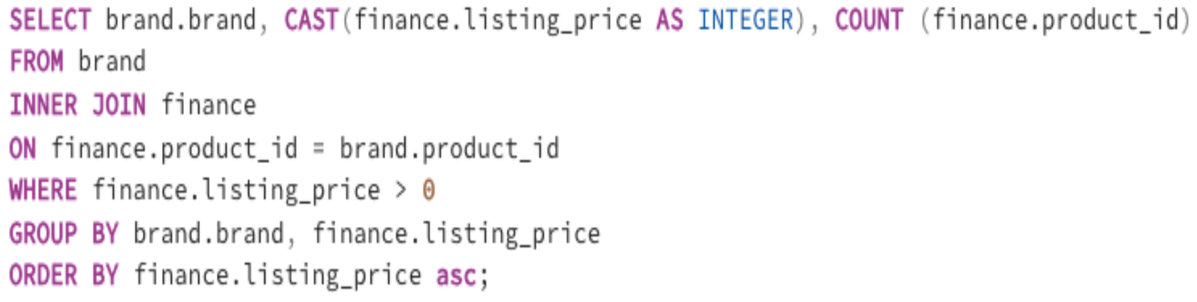


	total_rows bigint	count_description bigint	count_listing_price bigint	count_last_visited bigint
1	3178	3117	3120	2927

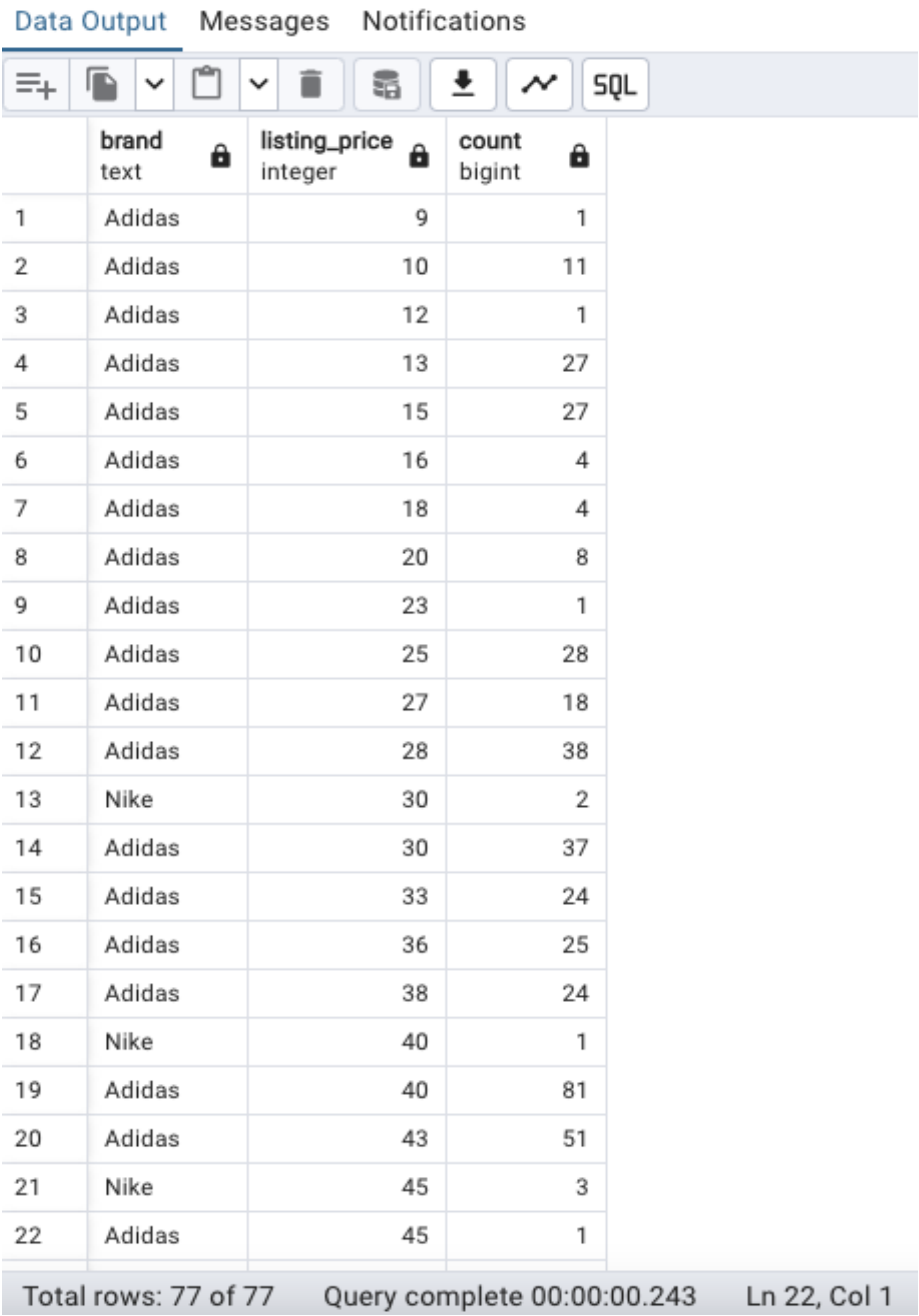
1) **`total_rows`**: The total number of rows or entries in the analysed dataset is 3,179 rows.
2) **`count_description`**: The total number of entries in the **description** column that have non-empty values is 3,117 entries.
3) **`count_listing_price`**: The total number of entries in the **listing_price** column that have non-empty values is 3,120 entries.
4) **`count_last_visited`**: The total number of entries in the **last_visited** column that have non-empty values is 2,928 entries.

2. Nike vs Adidas pricing

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.
How do the price points of Nike and Adidas products differ? Answering this question can help us build a picture of the company's stock range and customer market. We will run a query to produce a distribution of the **listing_price** and the count for each price, grouped by **brand**.



```
SELECT brand.brand, CAST(finance.listing_price AS INTEGER), COUNT (finance.product_id)
FROM brand
INNER JOIN finance
ON finance.product_id = brand.product_id
WHERE finance.listing_price > 0
GROUP BY brand.brand, finance.listing_price
ORDER BY finance.listing_price asc;
```



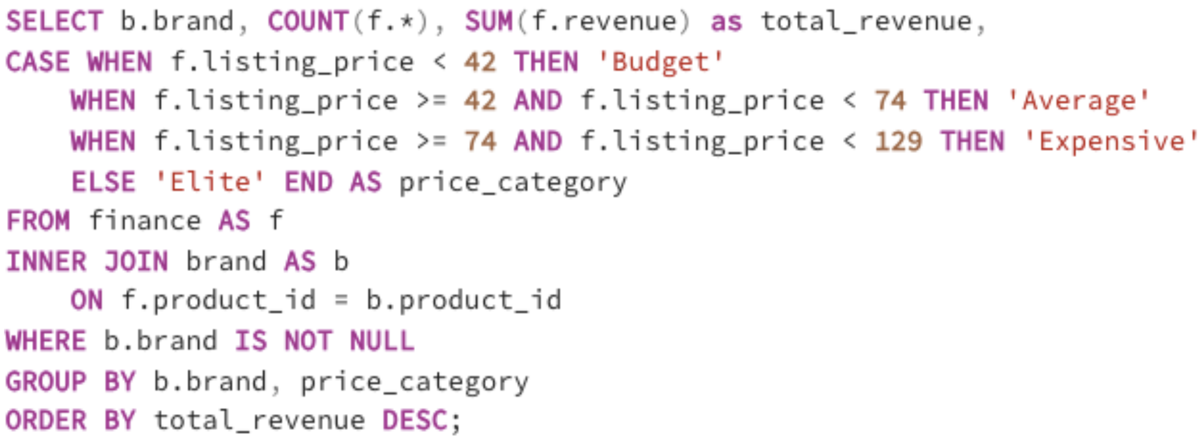
	brand text	listing_price integer	count bigint
1	Adidas	9	1
2	Adidas	10	11
3	Adidas	12	1
4	Adidas	13	27
5	Adidas	15	27
6	Adidas	16	4
7	Adidas	18	4
8	Adidas	20	8
9	Adidas	23	1
10	Adidas	25	28
11	Adidas	27	18
12	Adidas	28	38
13	Nike	30	2
14	Adidas	30	37
15	Adidas	33	24
16	Adidas	36	25
17	Adidas	38	24
18	Nike	40	1
19	Adidas	40	81
20	Adidas	43	51
21	Nike	45	3
22	Adidas	45	1

Total rows: 77 of 77   Query complete 00:00:00.243   Ln 22, Col 1

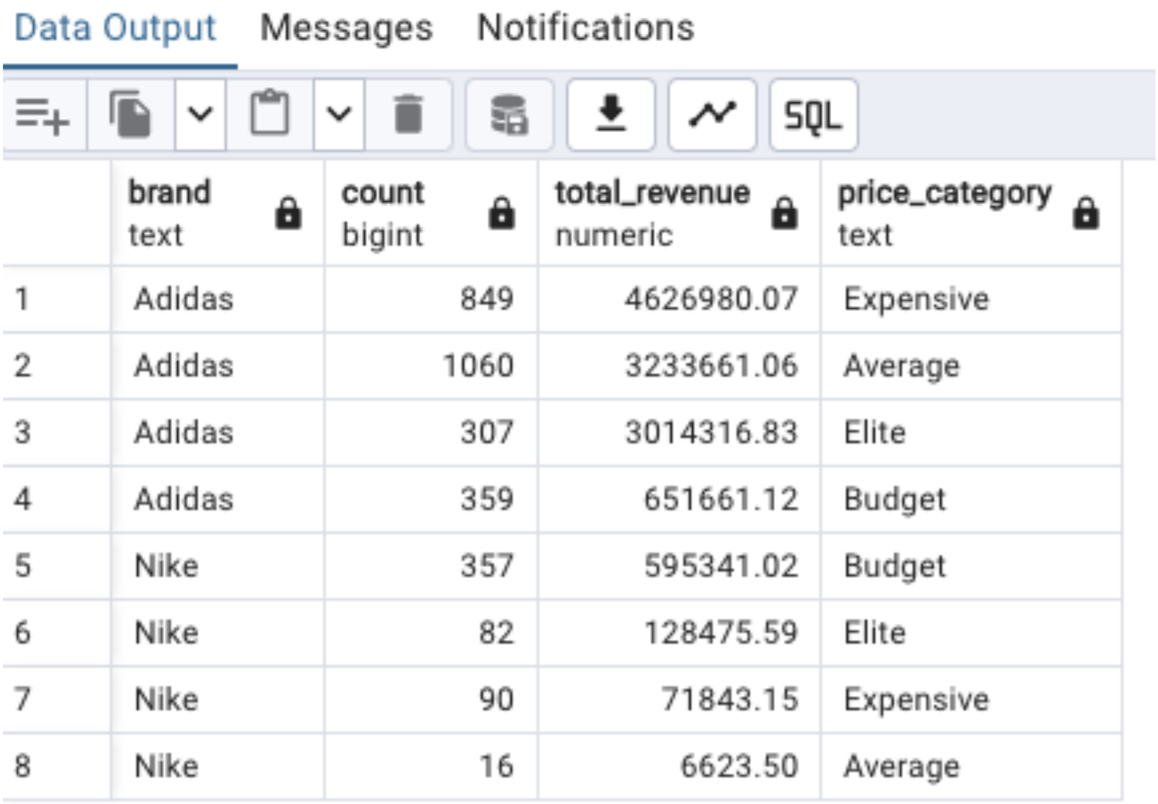
And the result continues till 77th row, as be seen in output of the query.

3. Labelling price ranges

It turns out there are 77 unique prices for the products in our database, which makes the output of our last query quite difficult to analyse.
Let's build on our previous query by assigning labels to different price ranges, grouping by **brand** and **label**. We will also include the total **revenue** for each price range and **brand**.



```
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 finance AS f
INNER JOIN brand AS b
    ON f.product_id = b.product_id
WHERE b.brand IS NOT NULL
GROUP BY b.brand, price_category
ORDER BY total_revenue DESC;
```

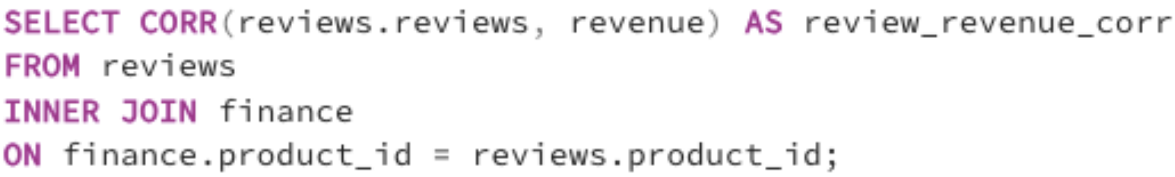


	brand text	count bigint	total_revenue numeric	price_category text
1	Adidas	849	4626980.07	Expensive
2	Adidas	1060	3233661.06	Average
3	Adidas	307	3014316.83	Elite
4	Adidas	359	651661.12	Budget
5	Nike	357	595341.02	Budget
6	Nike	82	128475.59	Elite
7	Nike	90	71843.15	Expensive
8	Nike	16	6623.50	Average

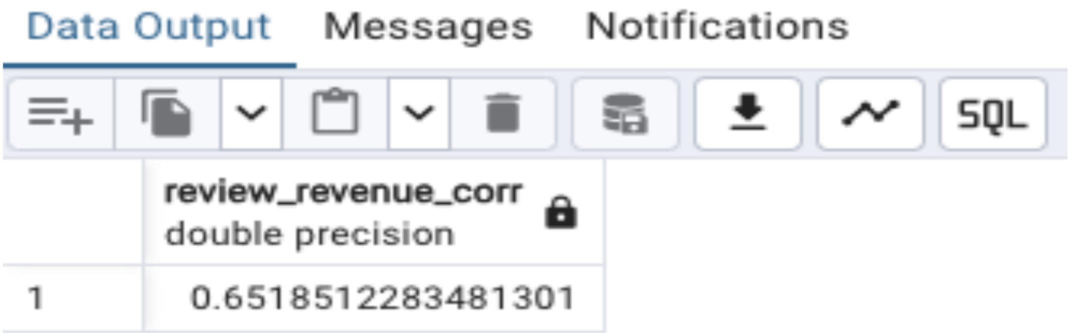
The majority of Adidas products fall into the “Average” category, totalling 1060 products. However, the brand also excels in the “Expensive” category, generating a total revenue of $4,626,980.07. Additionally, Adidas presents luxury products in the “Elite” category, although the quantity is lower (307 products), they contribute significantly to a total revenue of $3,014,316.83. Affordable products are not neglected, as the brand offers 359 products in the “Budget” category, even though the total revenue from this category is lower. On the other hand, the brand “Nike” also exhibits price variation in its products. Products in the “Budget” category are the primary choice for customers, amounting to 357 products, representing the majority of sales. Despite their large quantity, the total revenue from this category is lower compared to the “Adidas” brand, totalling $595,341.02. On the flip side, Nike also presents luxury products in the “Elite” category, with a smaller quantity (82 products), yet generating a total revenue of $128,475.59. Notably, Nike’s products in the “Expensive” and “Average” categories also contribute significantly to the total revenue.

4. Correlation between revenue and reviews

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.



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



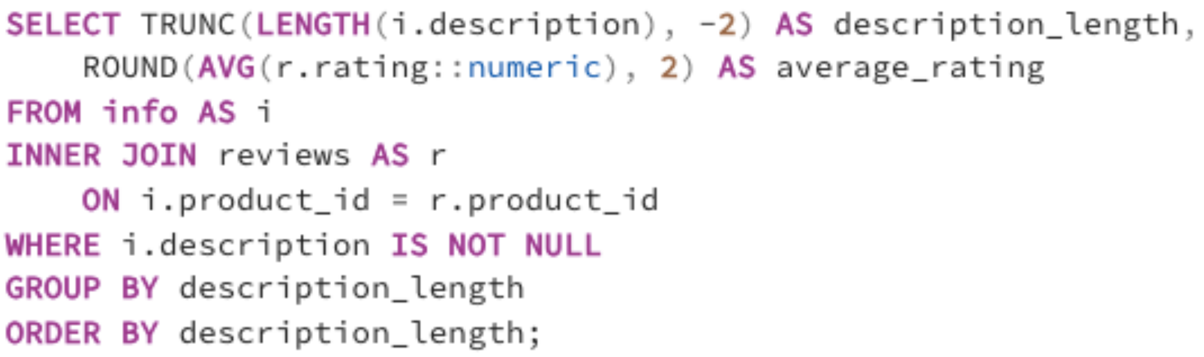
	review_revenue_corr double precision
1	0.6518512283481301

The correlation result between revenue and reviews is approximately 0.651. This indicates a moderate positive relationship between the two variables. With a correlation value exceeding 0.5, it can be concluded that the higher the number of reviews for a product, the tendency is for an increase in the product's revenue as well. However, it's important to note that correlation does not imply causation, meaning it cannot be determined whether more reviews directly cause an increase in revenue or vice versa.

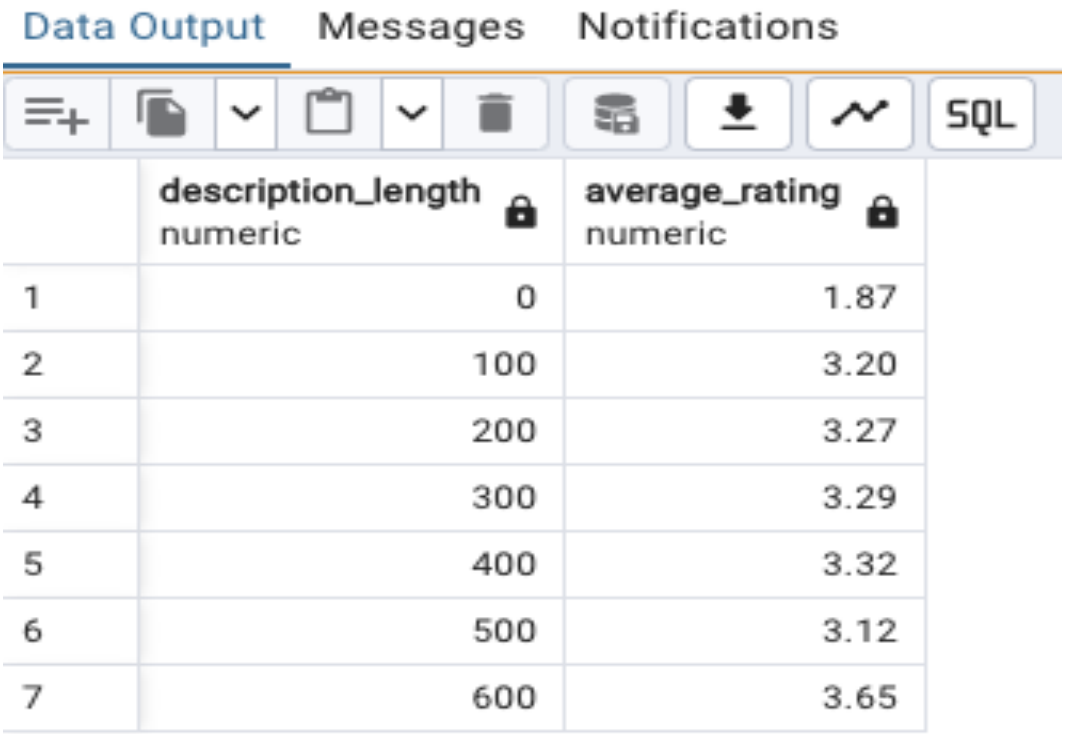
5. Ratings and reviews by product description length

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.

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**.



```
SELECT TRUNC(LENGTH(i.description), -2) AS description_length,
    ROUND(AVG(r.rating::numeric), 2) AS average_rating
FROM info AS i
INNER JOIN reviews AS r
    ON i.product_id = r.product_id
WHERE i.description IS NOT NULL
GROUP BY description_length
ORDER BY description_length;
```



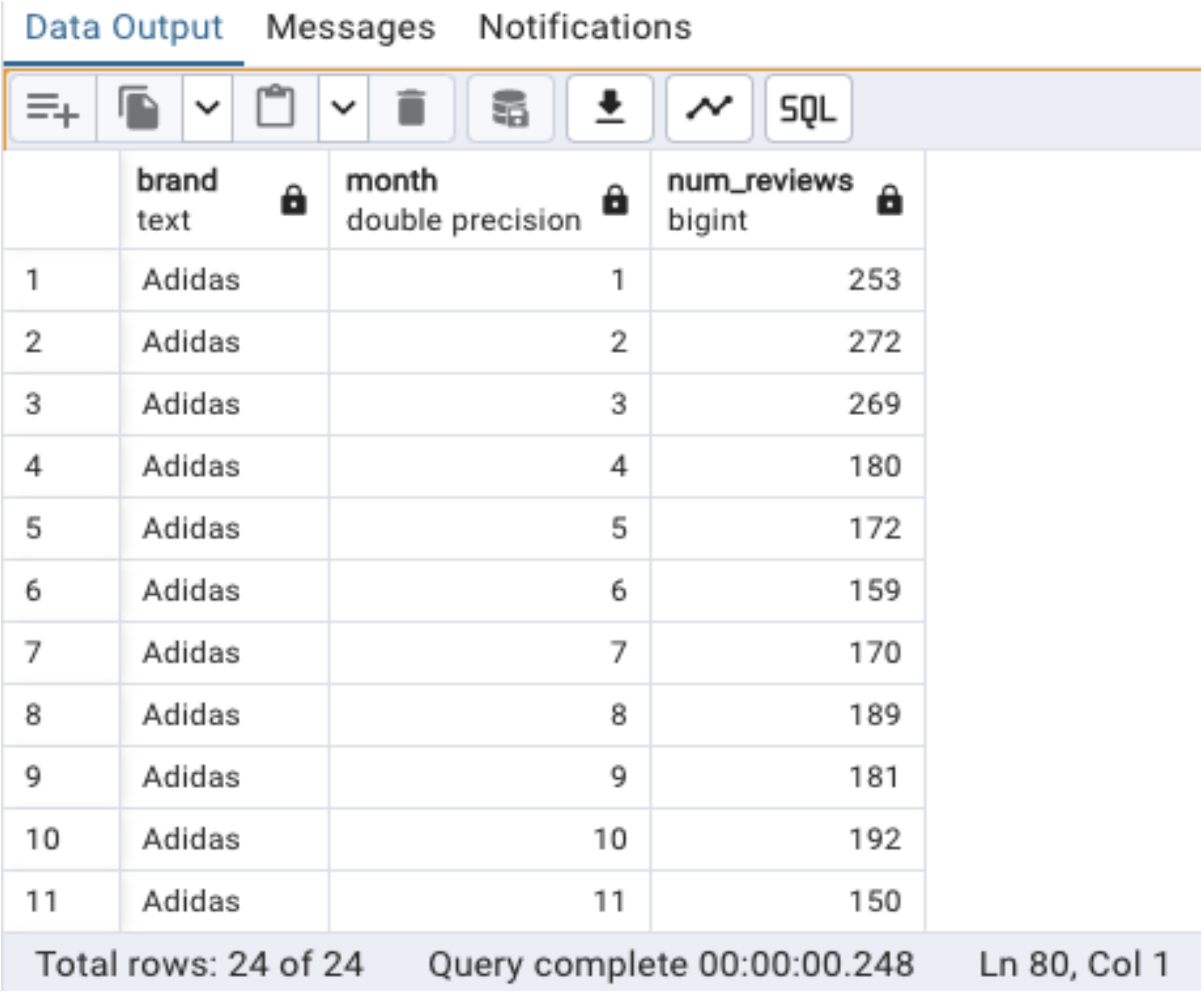
	description_length numeric	average_rating numeric
1	0	1.87
2	100	3.20
3	200	3.27
4	300	3.29
5	400	3.32
6	500	3.12
7	600	3.65

The results indicate a relationship between the length of product descriptions (**description_length**) and the average product rating (**average_rating**). It is evident that as the length of product descriptions increases, the average rating tends to rise. For instance, products with a description length of 600 characters have a higher average rating (3.65) compared to products with shorter descriptions.

6. 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.

```
SELECT b.brand, DATE_PART('month', t.last_visited_date) AS month, COUNT(r.*) AS num_reviews
FROM brand AS b
INNER JOIN traffic AS t
    ON b.product_id = t.product_id
INNER JOIN reviews AS r
    ON t.product_id = r.product_id
GROUP BY b.brand, month
HAVING b.brand IS NOT NULL
    AND DATE_PART('month', t.last_visited_date) IS NOT NULL
ORDER BY b.brand, month;
```



	brand text	month double precision	num_reviews bigint
1	Adidas	1	253
2	Adidas	2	272
3	Adidas	3	269
4	Adidas	4	180
5	Adidas	5	172
6	Adidas	6	159
7	Adidas	7	170
8	Adidas	8	189
9	Adidas	9	181
10	Adidas	10	192
11	Adidas	11	150

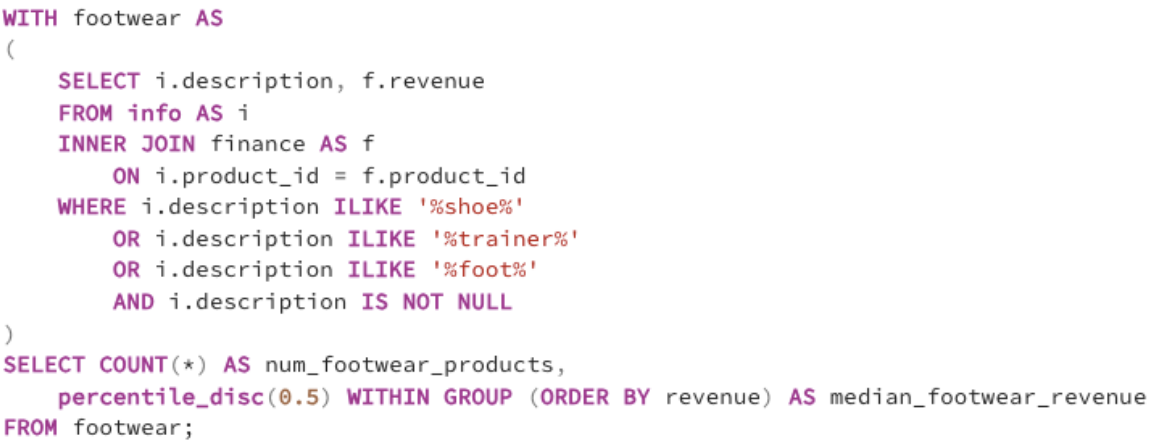
```
Total rows: 24 of 24    Query complete 00:00:00.248    Ln 80, Col 1
```

On the Adidas side, there’s an interesting variation in their review counts from month to month:
- Starting from January, Adidas received 253 reviews.
- The review count increased to 272 in February.
- March and April also showed high review counts with 269 and 180 reviews, respectively.
- May and June followed with 172 and 159 review counts.
- Then, July, August, and September recorded 170, 189, and 181 reviews.
- October exhibited a significant increase with 192 reviews, followed by November with 150 reviews.
- The year ended with December having 190 reviews.
On the Nike side, the review trend also captures attention:
- In January and February, Nike received an equal number of reviews, 52 each.
- March showed an increase to 55 reviews.
- Subsequent months like April and May recorded 42 and 41 reviews.
- June displayed a slight increase with 43 reviews.
- July had 37 reviews, while August and September had 29 and 28 reviews, respectively.

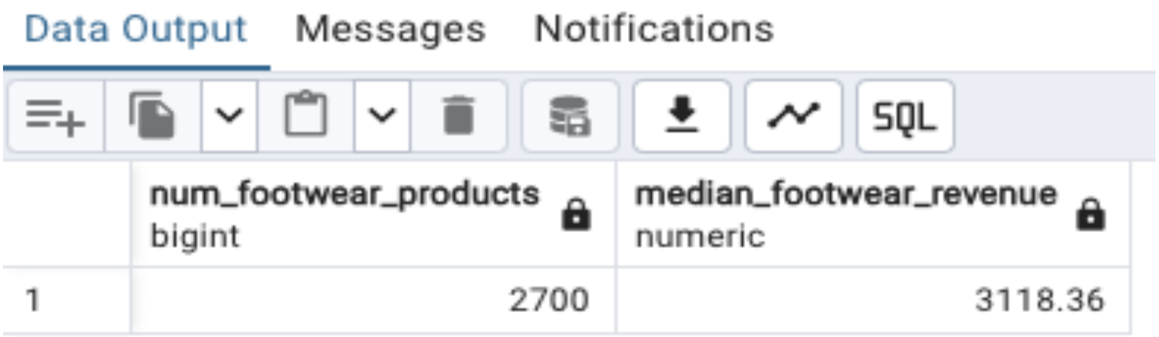
- October showed a drastic increase with 47 reviews.
- November and December then decreased again with 38 and 35 reviews each.

7. Footwear product performance

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!
So far, we have been primarily analysing Adidas vs Nike products. Now, let's switch our attention to the type of products being sold. As there are no labels for product type, we will create a Common Table Expression (CTE) that filters **description** for keywords, then use the results to 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 AS i
    INNER JOIN finance AS 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;
```



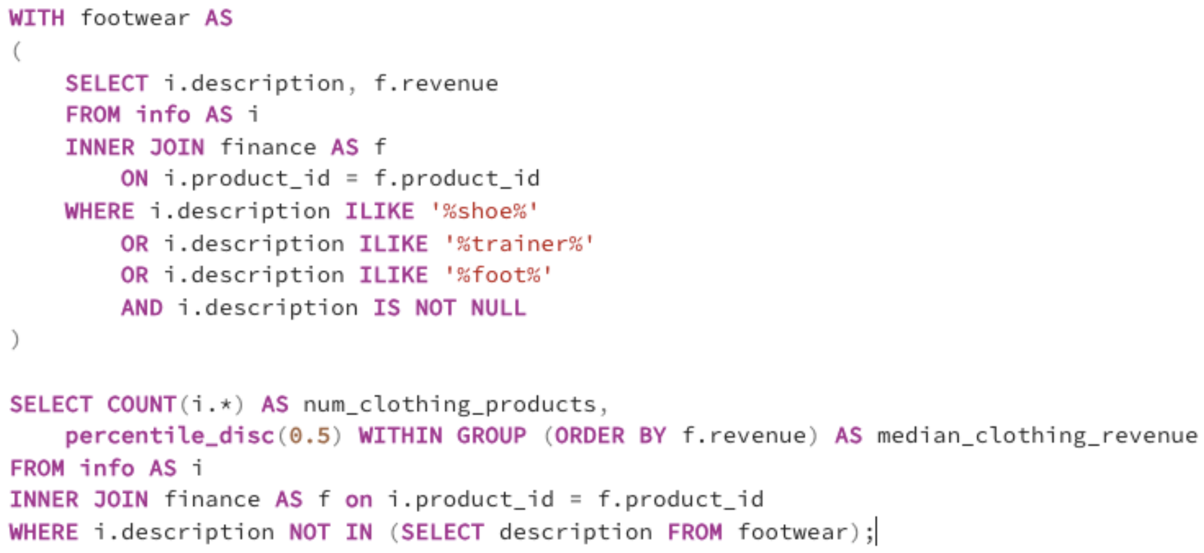
	num_footwear_products bigint	median_footwear_revenue numeric
1	2700	3118.36

The results indicate that there are 2,700 footwear products in the analysed dataset. The median revenue for these footwear products is $3,118.36. This suggests that half of the footwear products have revenue above $3,118.36 and the other half have revenue below that figure.

8. Clothing product performance

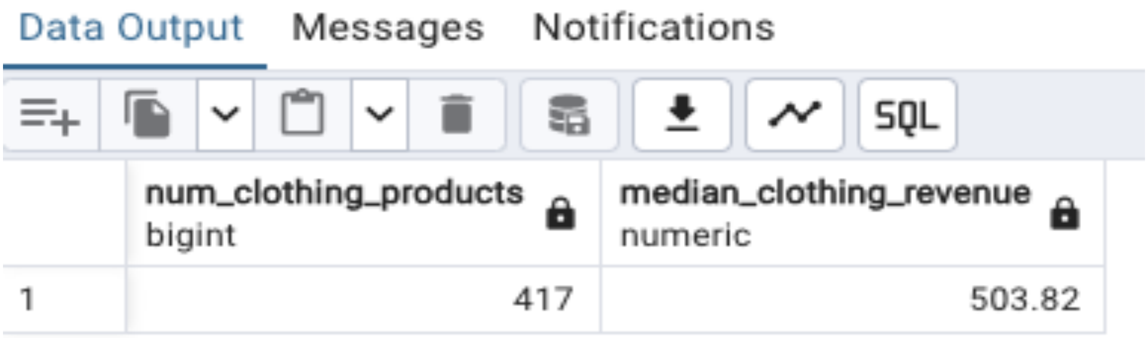
Recalling from the first task that we found there are 3,117 products without missing values for **description**. Of those, 2,700 are footwear products, which accounts for around 85% of the company's stock. They also generate a median revenue of over $3000 dollars!

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, for our final task, let's examine how this differs to clothing products. 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 AS i
    INNER JOIN finance AS 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 f.revenue) AS median_clothing_revenue
FROM info AS i
INNER JOIN finance AS f on i.product_id = f.product_id
WHERE i.description NOT IN (SELECT description FROM footwear);
```



	num_clothing_products bigint	median_clothing_revenue numeric
1	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. Focusing on product quality, customer service, and holistic marketing strategies can help improve reviews and revenue.
3. Analysing factors that influence monthly review fluctuations and planning appropriate marketing strategies.
4. Continuously monitoring product categories like footwear and clothing and making relevant price adjustments or marketing strategies.
5. Using this data as a foundation to design more effective and customer-oriented business strategies.
6. All of these recommendations can assist the brand in enhancing product performance, increasing revenue, and providing a better experience to customers.