

# Retail Transaction Analysis — Query Report

[Tableau Dashboard](#)

## ○ Customer Demographics

### 1. Age Group Distribution.

```
SELECT
    CASE WHEN age < 20 THEN '15-19'
        WHEN age >= 20 AND age < 30 THEN '20-29'
        WHEN age >= 30 AND age < 40 THEN '30-39'
        WHEN age >= 40 AND age < 50 THEN '40-49'
        WHEN age >= 50 AND age < 60 THEN '50-59'
        WHEN age >= 60 THEN '60 or older' END AS age_group,
    COUNT(age) AS age_distribution
FROM retail_data
WHERE age IS NOT NULL
GROUP BY age_group;
```

Data Output Messages Notifications

	age_group	age_distribution
1	15-19	19326
2	20-29	124939
3	30-39	43946
4	40-49	56213
5	50-59	28743
6	60 or older	28666

Total rows: 6 of 6    Query complete 00:00:00.376

### 2. Gender Distribution.

```
SELECT gender, COUNT(gender) AS gender_distribution
FROM retail_data
WHERE gender IS NOT NULL
GROUP BY gender
ORDER BY gender_distribution DESC;
```

Data Output Messages Notifications

	gender	gender_distribution
1	Female	114093
2	Male	187596

Total rows: 2 of 2    Query complete 00:00:00.155

### 3. Gender Distribution of Each Age Group.

```
SELECT
    CASE WHEN age < 20 THEN '15-19'
        WHEN age >= 20 AND age < 30 THEN '20-29'
        WHEN age >= 30 AND age < 40 THEN '30-39'
        WHEN age >= 40 AND age < 50 THEN '40-49'
        WHEN age >= 50 AND age < 60 THEN '50-59'
        WHEN age >= 60 THEN '60 or older' END AS age_group,
    gender,
    COUNT(gender) AS count_gender
FROM retail_data
WHERE gender IS NOT NULL and age IS NOT NULL
GROUP BY age_group, gender;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has three columns: 'age\_group\_text' (text), 'gender\_character' (character), and 'count\_gender' (bigint). The data shows the count of females and males across different age groups. The rows are numbered 1 to 12.

	age_group_text	gender_character	count_gender
1	15-19	Female	7699
2	15-19	Male	11608
3	20-29	Female	43492
4	20-29	Male	81325
5	30-39	Female	17422
6	30-39	Male	26485
7	40-49	Female	22635
8	40-49	Male	33513
9	50-59	Female	11399
10	50-59	Male	17312
11	60 or older	Female	11373
12	60 or older	Male	17253

Total rows: 12 of 12 Query complete 00:00:00.256

#### 4. Income Level Distribution.

```
SELECT income, COUNT(income) AS income_distribution
FROM retail_data
WHERE income IS NOT NULL
GROUP BY income;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'income' (character) and 'income\_distribution' (bigint). The data shows the count of customers in High, Low, and Medium income levels. The rows are numbered 1 to 3.

	income	income_distribution
1	High	75229
2	Low	96257
3	Medium	130230

Total rows: 3 of 3 Query complete 00:00:00.241

#### 5. Customer Segment Distribution.

```
SELECT customer_segment, COUNT(customer_segment) AS count_customer_segment
FROM retail_data
WHERE customer_segment IS NOT NULL
GROUP BY customer_segment
ORDER BY customer_segment DESC;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'customer\_segment' (character) and 'count\_customer\_segment' (bigint). The data shows the count of customers in Regular, Premium, and New segments. The rows are numbered 1 to 3.

	customer_segment	count_customer_segment
1	Regular	146219
2	Premium	64387
3	New	91185

Total rows: 3 of 3 Query complete 00:00:00.194

#### 6. Customer Segment Distribution of Each Age Group, ranking from the highest to lowest count within each age group.

```
SELECT
CASE WHEN age < 20 THEN '15-19'
WHEN age >= 20 AND age < 30 THEN '20-29'
WHEN age >= 30 AND age < 40 THEN '30-39'
WHEN age >= 40 AND age < 50 THEN '40-49'
WHEN age >= 50 AND age < 60 THEN '50-59'
WHEN age >= 60 THEN '60 or older' END AS age_group,
customer_segment, COUNT(age)
FROM retail_data
```

```

WHERE customer_segment IS NOT NULL AND age IS NOT NULL
GROUP BY customer_segment, age_group
ORDER BY age_group, COUNT(age) DESC;

```

Data Output Messages Notifications

The screenshot shows a table with three columns: 'age\_group' (text), 'customer\_segment' (character), and 'count' (bigint). The data is ordered by count in descending order. The first few rows are:

	age_group	customer_segment	count
1	15-19	Regular	11375
2	15-19	New	6425
3	15-19	Premium	1517
4	20-29	Regular	77146
5	20-29	New	34094
6	20-29	Premium	13656
7	30-39	Regular	15867
8	30-39	New	14225
9	30-39	Premium	13805
10	40-49	Regular	19882
11	40-49	New	18310
12	40-49	Premium	17970
13	50-59	Regular	10975
14	50-59	New	9029
15	50-59	Premium	8709
16	60 or older	Regular	10910
17	60 or older	New	9052
18	60 or older	Premium	8671

Total rows: 18 of 18 | Query complete 00:00:00.163

## 7. Age Group Distribution of Each Customer Segment, ranking from the highest to lowest count within each customer segment.

```

SELECT customer_segment,
CASE WHEN age < 20 THEN '15-19'
     WHEN age >= 20 AND age < 30 THEN '20-29'
     WHEN age >= 30 AND age < 40 THEN '30-39'
     WHEN age >= 40 AND age < 50 THEN '40-49'
     WHEN age >= 50 AND age < 60 THEN '50-59'
     WHEN age >= 60 THEN '60 or older' END AS age_group,
COUNT(age)

```

```

FROM retail_data
WHERE customer_segment IS NOT NULL AND age IS NOT NULL
GROUP BY customer_segment, age_group
ORDER BY customer_segment, COUNT(age) DESC;

```

Data Output Messages Notifications

The screenshot shows a table with three columns: 'customer\_segment' (character), 'age\_group' (text), and 'count' (bigint). The data is ordered by count in descending order. The first few rows are:

	customer_segment	age_group	count
1	New	20-29	34094
2	New	40-49	18310
3	New	30-39	14225
4	New	60 or older	9052
5	New	50-59	9029
6	New	15-19	6425
7	Premium	40-49	17970
8	Premium	30-39	13805
9	Premium	20-29	13656
10	Premium	50-59	8709
11	Premium	60 or older	8671
12	Premium	15-19	1517
13	Regular	20-29	77146
14	Regular	40-49	19882
15	Regular	30-39	15867
16	Regular	15-19	11375
17	Regular	50-59	10975
18	Regular	60 or older	10910

Total rows: 18 of 18 | Query complete 00:00:00.708

**8. Transaction Counts By Countries, ranking from the highest to lowest count.**

```
SELECT country, COUNT(country)
FROM retail_data
WHERE country IS NOT NULL
GROUP BY country
ORDER BY COUNT(country) DESC;
```

Data Output Messages Notifications

The screenshot shows a database query results interface. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with icons for file operations like new, open, save, and print. The main area displays a table with two columns: 'country' (character type) and 'count' (bigint type). The table has 5 rows, each representing a country and its transaction count. The data is sorted by count in descending order. At the bottom of the table, it says 'Total rows: 5 of 5' and 'Query complete 00:00:00.454'.

	country character	count bigint
1	USA	95268
2	UK	63122
3	Germany	52880
4	Australia	45377
5	Canada	45359

- **Purchasing Behaviors**

**9. Total Purchases By Age Group, ranking from the highest to lowest purchase.**

```
SELECT
    CASE WHEN age < 20 THEN '15-19'
        WHEN age >= 20 AND age < 30 THEN '20-29'
        WHEN age >= 30 AND age < 40 THEN '30-39'
        WHEN age >= 40 AND age < 50 THEN '40-49'
        WHEN age >= 50 AND age < 60 THEN '50-59'
        WHEN age >= 60 THEN '60 or older' END AS age_group,
    SUM(total_purchases) AS total_purchase
FROM retail_data
WHERE total_purchases IS NOT NULL AND age IS NOT NULL
GROUP BY age_group
ORDER BY total_purchase DESC;
```

Data Output Messages Notifications		
	age_group text	total_purchase bigint
1	20-29	667703
2	40-49	305131
3	30-39	238730
4	50-59	157362
5	60 or older	156916
6	15-19	91867

Total rows: 6 of 6    Query complete 00:00:00.142

**10. Total Amount of Spending and Average Spending Per Purchase By Each Customer Segment, ranking from the highest to lowest amount.**

```
SELECT customer_segment, ROUND(SUM(total_amount),-3) AS amount_spent,
    ROUND(AVG(amount),2) AS avg_spending_each_time
FROM retail_data
WHERE customer_segment IS NOT NULL
GROUP BY customer_segment
ORDER BY amount_spent DESC;
```

Data Output Messages Notifications		
	customer_segment character	amount_spent numeric
1	Regular	200190000
2	New	124727000
3	Premium	87795000

Total rows: 3 of 3    Query complete 00:00:00.207

**11. Total Purchases By Product Category Purchased By the 'Premium' Customer Segment, ranking from the highest to lowest purchase.**

```
SELECT product_category, SUM(total_purchases) AS total_purchase
FROM retail_data
WHERE product_category IS NOT NULL
GROUP BY product_category, customer_segment
HAVING customer_segment = 'Premium'
ORDER BY COUNT(product_category) DESC;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'product\_category' (character) and 'total\_purchase' (bigint). The data shows five rows: Grocery (82361), Electronics (81342), Clothing (60838), Books (60097), and Home Decor (59578).

	product_category character	total_purchase bigint
1	Grocery	82361
2	Electronics	81342
3	Clothing	60838
4	Books	60097
5	Home Decor	59578

Total rows: 5 of 5 Query complete 00:00:00.152

## 12. Total Purchases By Product Category Purchased By the ‘Regular’ Customer Segment, ranking from the highest to lowest purchase.

```
SELECT product_category, SUM(total_purchases) AS total_purchase
FROM retail_data
WHERE product_category IS NOT NULL
GROUP BY product_category, customer_segment
HAVING customer_segment = 'Regular'
ORDER BY COUNT(product_category) DESC;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'product\_category' (character) and 'total\_purchase' (bigint). The data shows five rows: Electronics (191483), Grocery (164114), Clothing (142383), Books (142089), and Home Decor (142377).

	product_category character	total_purchase bigint
1	Electronics	191483
2	Grocery	164114
3	Clothing	142383
4	Books	142089
5	Home Decor	142377

Total rows: 5 of 5 Query complete 00:00:00.147

## 13. Total Purchases By Product Category Purchased By the ‘New’ Customer Segment, ranking from the highest to lowest purchase.

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'product\_category' (character) and 'total\_purchase' (bigint). The data shows five rows: Grocery (110949), Electronics (108763), Books (89941), Clothing (90488), and Home Decor (89224).

	product_category character	total_purchase bigint
1	Grocery	110949
2	Electronics	108763
3	Books	89941
4	Clothing	90488
5	Home Decor	89224

Total rows: 5 of 5 Query complete 00:00:00.148

o ***Product Sales***

**14. Top-performing Product Categories in terms of total purchases, ranking from the highest to lowest purchase.**

```
SELECT product_category, SUM(total_purchases) AS total_purchase  
FROM retail_data  
WHERE product_category IS NOT NULL  
GROUP BY product_category  
ORDER BY total_purchase DESC;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'product\_category' (character) and 'total\_purchase' (bigint). The data shows five rows: Electronics (381850), Grocery (357655), Clothing (293930), Books (292337), and Home Decor (291399). The total number of rows is 5, and the query took 0:00:00.156 to complete.

	product_category	total_purchase
1	Electronics	381850
2	Grocery	357655
3	Clothing	293930
4	Books	292337
5	Home Decor	291399

**15. Top-performing Product Categories in terms of the total amount of spending by customers, ranking from the highest to the lowest amount.**

```
SELECT product_category, ROUND(SUM(total_amount),-3) AS total_amount_spent  
FROM retail_data  
WHERE product_category IS NOT NULL  
GROUP BY product_category  
ORDER BY total_amount_spent DESC;
```

Data Output Messages Notifications

A screenshot of a database query results window. The table has two columns: 'product\_category' (character) and 'total\_amount\_spent' (numeric). The data shows five rows: Electronics (97539000), Grocery (91184000), Clothing (74902000), Books (74672000), and Home Decor (74362000). The total number of rows is 5, and the query took 0:00:00.151 to complete.

	product_category	total_amount_spent
1	Electronics	97539000
2	Grocery	91184000
3	Clothing	74902000
4	Books	74672000
5	Home Decor	74362000

**16. Top-performing Product Brands in terms of total purchases, ranking from the highest to lowest purchase.**

```
SELECT product_brand, SUM(total_purchases) AS total_purchase  
FROM retail_data  
WHERE product_brand IS NOT NULL  
GROUP BY product_brand  
ORDER BY total_purchase DESC;
```

Data Output Messages Notifications

A screenshot of a database management system interface showing a results table. The table has two columns: 'product\_brand' and 'total\_purchase'. The 'product\_brand' column contains brand names, and the 'total\_purchase' column contains their total purchase count. The data is ordered from highest to lowest.

	product_brand	total_purchase
1	Pepsi	162369
2	Samsung	98915
3	Zara	98433
4	Coca-Cola	98381
5	Sony	98078
6	Nike	97916
7	HarperCollins ...	97696
8	Adidas	97584
9	Home Depot ...	97483
10	Bed Bath & Bey...	97467
11	Penguin Books ...	97344
12	Random House...	97216
13	Nestle	96908
14	Apple	96472
15	IKEA	96459
16	Whirlpool	40115
17	Mitsubhisi	36214
18	BlueStar	12022

Total rows: 18 of 18 | Query complete 00:00:00.308

## 17. Top-performing Product Brands Within Each Product Category in terms of total amount of spending by customers, ranking from the highest to lowest amount.

```
SELECT product_category, product_brand, SUM(total_purchases) AS total_purchase,
       ROUND(SUM(total_amount),-3) AS total_amount_spent
  FROM retail_data
 WHERE product_category IS NOT NULL AND product_brand IS NOT NULL
 GROUP BY product_category, product_brand
 ORDER BY product_category, total_amount_spent DESC;
```

Data Output Messages Notifications

A screenshot of a database management system interface showing a results table. The table has four columns: 'product\_category', 'product\_brand', 'total\_purchase', and 'total\_amount\_spent'. The data is ordered by product\_category and total\_amount\_spent.

	product_category	product_brand	total_purchase	total_amount_spent
1	Books	HarperCollins ...	97608	24919000
2	Books	Penguin Books ...	97245	24872000
3	Books	Random House...	97109	24783000
4	Clothing	Nike	97830	25011000
5	Clothing	Zara	98337	24991000
6	Clothing	Adidas	97451	24817000
7	Electronics	Samsung	98819	25387000
8	Electronics	Sony	98022	25102000
9	Electronics	Apple	96403	24630000
10	Electronics	Whirlpool	40115	10092000
11	Electronics	Mitsubhisi	36214	9198000
12	Electronics	BlueStar	12022	3074000
13	Grocery	Pepsi	162284	41382000
14	Grocery	Coca-Cola	98273	25020000
15	Grocery	Nestle	96767	24708000
16	Home Decor	Bed Bath & Bey...	97372	24990000
17	Home Decor	Home Depot ...	97367	24772000
18	Home Decor	IKEA	96353	24524000

Total rows: 18 of 18 | Query complete 00:00:00.216

## 18. Top-performing Product Types Within Each Product Brand Under Each Product Category in terms of total purchases, ranking from the highest to lowest purchase.

```
SELECT product_category, product_brand, product_type, SUM(total_purchases) AS total_purchase
  FROM retail_data
 WHERE product_category IS NOT NULL AND product_brand IS NOT NULL AND product_type IS NOT
        NULL
 GROUP BY product_category, product_brand, product_type
 ORDER BY product_category, product_brand, total_purchase DESC;
```

Data Output Messages Notifications

	product_category	product_brand	product_type	total_purchase
	character	character	character varying (50)	bigint
1	Books	HarperCollins	Thriller	33262
2	Books	HarperCollins	Fiction	32243
3	Books	HarperCollins	Non-Fiction	32103
4	Books	Penguin Books	Non-Fiction	32678
5	Books	Penguin Books	Children's	32430
6	Books	Penguin Books	Fiction	32137
7	Books	Random House	Literature	32439
8	Books	Random House	Fiction	32411
9	Books	Random House	Non-Fiction	32259
10	Clothing	Adidas	T-shirt	33033
11	Clothing	Adidas	Shoes	32711
12	Clothing	Adidas	Jacket	31707
13	Clothing	Nike	Shorts	32934
14	Clothing	Nike	T-shirt	32494
15	Clothing	Nike	Shoes	32402
16	Clothing	Zara	Dress	32818
17	Clothing	Zara	Shirt	32781
18	Clothing	Zara	Jeans	32738
19	Electronics	Apple	Smartphone	32293
20	Electronics	Apple	Laptop	32141
21	Electronics	Apple	Tablet	31969
22	Electronics	BlueStar	BlueStar AC	12022
23	Electronics	Mitsubishi	Mitsubishi 1.5 Ton 3 Star Split AC	36214
24	Electronics	Samsung	Television	33128
25	Electronics	Samsung	Smartphone	33064
26	Electronics	Samsung	Tablet	32627
27	Electronics	Sony	Smartphone	33280
28	Electronics	Sony	Headphones	32501
29	Electronics	Sony	Television	32241
30	Electronics	Whirepool	Fridge	40115
31	Grocery	Coca-Cola	Water	33207
32	Grocery	Coca-Cola	Juice	32977
33	Grocery	Coca-Cola	Soft Drink	32089
34	Grocery	Nestle	Coffee	33004
35	Grocery	Nestle	Chocolate	32398
36	Grocery	Nestle	Snacks	31365
37	Grocery	Pepsi	Water	98056
38	Grocery	Pepsi	Juice	32147
39	Grocery	Pepsi	Soft Drink	32081
40	Home Decor	Bed Bath & Bey...	Kitchen	33149
41	Home Decor	Bed Bath & Bey...	Bathroom	32316
42	Home Decor	Bed Bath & Bey...	Bedding	31907
43	Home Decor	Home Depot	Decorations	32829
44	Home Decor	Home Depot	Furniture	32527
45	Home Decor	Home Depot	Tools	32011
46	Home Decor	IKEA	Lighting	32630
47	Home Decor	IKEA	Decorations	32100
48	Home Decor	IKEA	Furniture	31623

Total rows: 48 of 48 Query complete 00:00:00.283

- ***Customer Satisfaction***

### 19. Average Ratings Received By Each Product Brand.

```
SELECT product_brand, ROUND(avg(ratings),1) AS avg_ratings
FROM retail_data
WHERE product_brand IS NOT NULL
GROUP BY product_brand
ORDER BY avg_ratings DESC;
```

	product_brand	avg_ratings
	character	numeric
1	BlueStar	4.4
2	Mitsubhisi	3.9
3	Whirepool	3.5
4	Pepsi	3.3
5	HarperCollins	3.1
6	Home Depot	3.1
7	IKEA	3.1
8	Nestle	3.1
9	Nike	3.1
10	Penguin Books	3.1
11	Random House	3.1
12	Samsung	3.1
13	Sony	3.1
14	Adidas	3.1
15	Zara	3.1
16	Apple	3.1
17	Bed Bath & Bey...	3.1
18	Coca-Cola	3.1

Total rows: 18 of 18 | Query complete 00:00:00.284

### 20. Feedback Counts For Each Product Brand Under the 'Electronics' Product Category.

```
SELECT product_brand, feedback, COUNT(feedback) AS feedback_count
FROM retail_data
WHERE product_brand IS NOT NULL AND feedback IS NOT NULL
GROUP BY product_brand, feedback, product_category
HAVING product_category = 'Electronics'
ORDER BY product_brand, COUNT(feedback) DESC;
```

	product_brand	feedback	feedback_count
	character	character	bigint
1	Apple	Excellent	6353
2	Apple	Good	4692
3	Apple	Average	4111
4	Apple	Bad	2889
5	BlueStar	Excellent	2030
6	BlueStar	Good	226
7	Mitsubhisi	Good	4171
8	Mitsubhisi	Excellent	2553
9	Samsung	Excellent	6569
10	Samsung	Good	4774
11	Samsung	Average	4177
12	Samsung	Bad	2818
13	Sony	Excellent	6498
14	Sony	Good	4762
15	Sony	Average	4128
16	Sony	Bad	2908
17	Whirepool	Good	7292
18	Whirepool	Average	149
19	Whirepool	Excellent	3
20	Whirepool	Bad	1

Total rows: 20 of 20 | Query complete 00:00:00.194

## 21. Feedback Counts For Each Product Brand Under the ‘Grocery’ Product Category.

```
SELECT product_brand, feedback, COUNT(feedback) AS feedback_count
FROM retail_data
WHERE product_brand IS NOT NULL AND feedback IS NOT NULL
GROUP BY product_brand, feedback, product_category
HAVING product_category = 'Grocery'
ORDER BY product_brand, feedback_count DESC;
```

Data Output			
	product_brand character	feedback character	feedback_count bigint
1	Coca-Cola	Excellent	6458
2	Coca-Cola	Good	4758
3	Coca-Cola	Average	4196
4	Coca-Cola	Bad	2941
5	Nestle	Excellent	6335
6	Nestle	Good	4734
7	Nestle	Average	4200
8	Nestle	Bad	2803
9	Pepsi	Good	16798
10	Pepsi	Excellent	6452
11	Pepsi	Average	4108
12	Pepsi	Bad	2908

Total rows: 12 of 12 Query complete 00:00:00.238

## 22. Feedback Counts For Each Product Brand Under the ‘Clothing’ Product Category.

```
SELECT product_brand, feedback, COUNT(feedback) AS feedback_count
FROM retail_data
WHERE product_brand IS NOT NULL AND feedback IS NOT NULL
GROUP BY product_brand, feedback, product_category
HAVING product_category = 'Clothing'
ORDER BY product_brand, feedback_count DESC;
```

Data Output			
	product_brand character	feedback character	feedback_count bigint
1	Adidas	Excellent	6377
2	Adidas	Good	4760
3	Adidas	Average	4088
4	Adidas	Bad	2975
5	Nike	Excellent	6425
6	Nike	Good	4690
7	Nike	Average	4078
8	Nike	Bad	2917
9	Zara	Excellent	6365
10	Zara	Good	4771
11	Zara	Average	4293
12	Zara	Bad	2911

Total rows: 12 of 12 Query complete 00:00:00.377

## 23. Feedback Counts For Each Product Brand Under the ‘Books’ Product Category.

```
SELECT product_brand, feedback, COUNT(feedback) AS feedback_count
FROM retail_data
WHERE product_brand IS NOT NULL AND feedback IS NOT NULL
GROUP BY product_brand, feedback, product_category
HAVING product_category = 'Books'
ORDER BY product_brand, feedback_count DESC;
```

Data Output Messages Notifications

A screenshot of a database interface showing a table of data. The table has three columns: 'product\_brand' (character), 'feedback' (character), and 'feedback\_count' (bigint). The data consists of 12 rows, each with a row number (1-12) and values for the three columns. The 'feedback' column contains values 'Excellent', 'Good', 'Average', and 'Bad'. The 'feedback\_count' column shows varying numbers of reviews for each brand.

	product_brand	feedback	feedback_count
	character	character	bigint
1	HarperCollins	Excellent	6427
2	HarperCollins	Good	4801
3	HarperCollins	Average	4205
4	HarperCollins	Bad	2894
5	Penguin Books	Excellent	6365
6	Penguin Books	Good	4715
7	Penguin Books	Average	4203
8	Penguin Books	Bad	2827
9	Random House	Excellent	6374
10	Random House	Good	4718
11	Random House	Average	4126
12	Random House	Bad	2867

Total rows: 12 of 12 Query complete 00:00:00.231

## 24. Feedback Counts For Each Product Brand Under the 'Home Decor' Product Category.

```
SELECT product_brand, feedback, COUNT(feedback) AS feedback_count
FROM retail_data
WHERE product_brand IS NOT NULL AND feedback IS NOT NULL
GROUP BY product_brand, feedback, product_category
HAVING product_category = 'Home Decor'
ORDER BY product_brand, feedback_count DESC;
```

Data Output Messages Notifications

A screenshot of a database interface showing a table of data. The table has three columns: 'product\_brand' (character), 'feedback' (character), and 'feedback\_count' (bigint). The data consists of 12 rows, each with a row number (1-12) and values for the three columns. The 'feedback' column contains values 'Excellent', 'Good', 'Average', and 'Bad'. The 'feedback\_count' column shows varying numbers of reviews for each brand.

	product_brand	feedback	feedback_count
	character	character	bigint
1	Bed Bath & Beyond	Excellent	6430
2	Bed Bath & Beyond	Good	4724
3	Bed Bath & Beyond	Average	4201
4	Bed Bath & Beyond	Bad	2859
5	Home Depot	Excellent	6322
6	Home Depot	Good	4709
7	Home Depot	Average	4193
8	Home Depot	Bad	2890
9	IKEA	Excellent	6215
10	IKEA	Good	4859
11	IKEA	Average	4087
12	IKEA	Bad	2802

Total rows: 12 of 12 Query complete 00:00:00.456

- **Transaction Logistics**

## 25. Top Chosen Shipping Methods, ranking from the highest to lowest count.

```
SELECT shipping_method, COUNT(shipping_method)
FROM retail_data
WHERE shipping_method IS NOT NULL
GROUP BY shipping_method
ORDER BY COUNT(shipping_method) DESC;
```

Data Output		
	shipping_method	count
1	Same-Day	104155
2	Express	102354
3	Standard	95160

Total rows: 3 of 3    Query complete 00:00:00.375

## 26. Top Chosen Payment Methods, ranking from the highest to lowest count.

```
SELECT payment_method, COUNT(payment_method)
FROM retail_data
WHERE payment_method IS NOT NULL
GROUP BY payment_method
ORDER BY COUNT(payment_method) DESC;
```

Data Output		
	payment_method	count
1	Credit Card	90115
2	Debit Card	76790
3	Cash	73807
4	PayPal	60997

Total rows: 4 of 4    Query complete 00:00:00.152

## 27. Order Status By Count.

```
SELECT order_status, COUNT(order_status)
FROM retail_data
WHERE order_status IS NOT NULL
GROUP BY order_status;
```

Data Output		
	order_status	count
1	Delivered	130448
2	Pending	49101
3	Processing	57198
4	Shipped	65024

Total rows: 4 of 4    Query complete 00:00:00.147