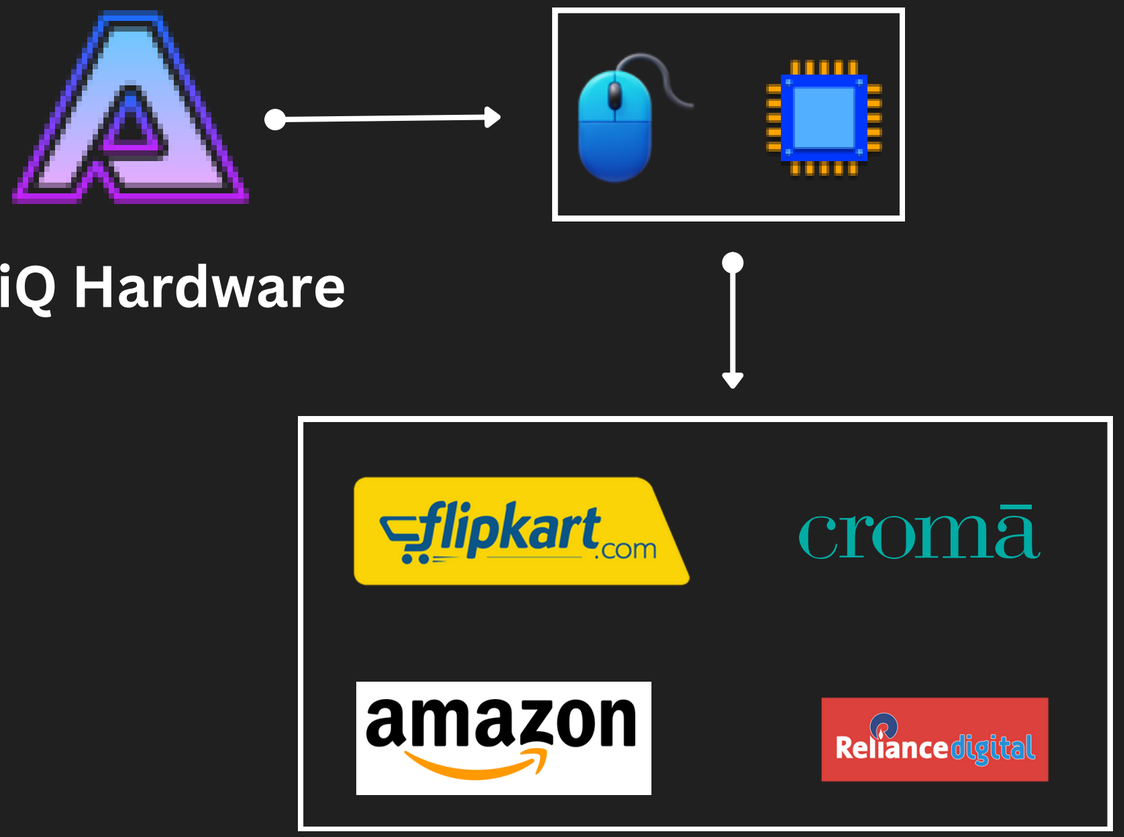
# WELCOME

ADHOC ANALYSIS
WITH MYSQL 

DOMAIN:CONSUMER GOODS

Sree Vignesh

# BUSINESS MODEL




# *PROBLEM OVERVIEW*

Atliq Hardwares (imaginary company) is one of the leading computer hardware producers in India and well expanded in other countries too.

But the management noticed that they do not get enough insights to make quick and smart data-informed decisions.

# *OBJECTIVE*

- Address all ad-hoc requests.
- Uncover hidden insights to meet the business needs.

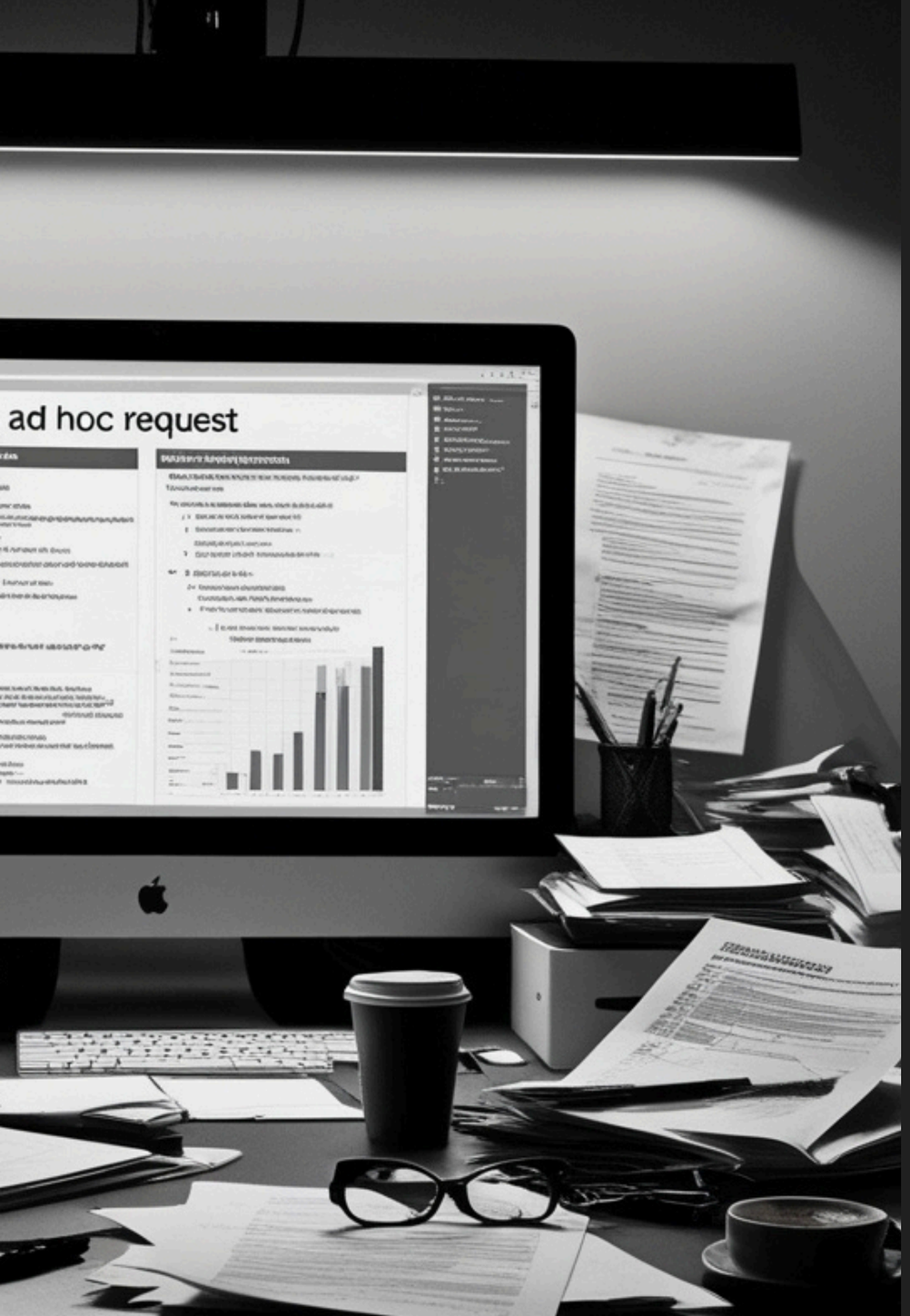

# *TOOLS USED*

For Ad-Hoc Analysis

For Visualization

For Presentation

For Screen Recording




# *REQUEST 1*

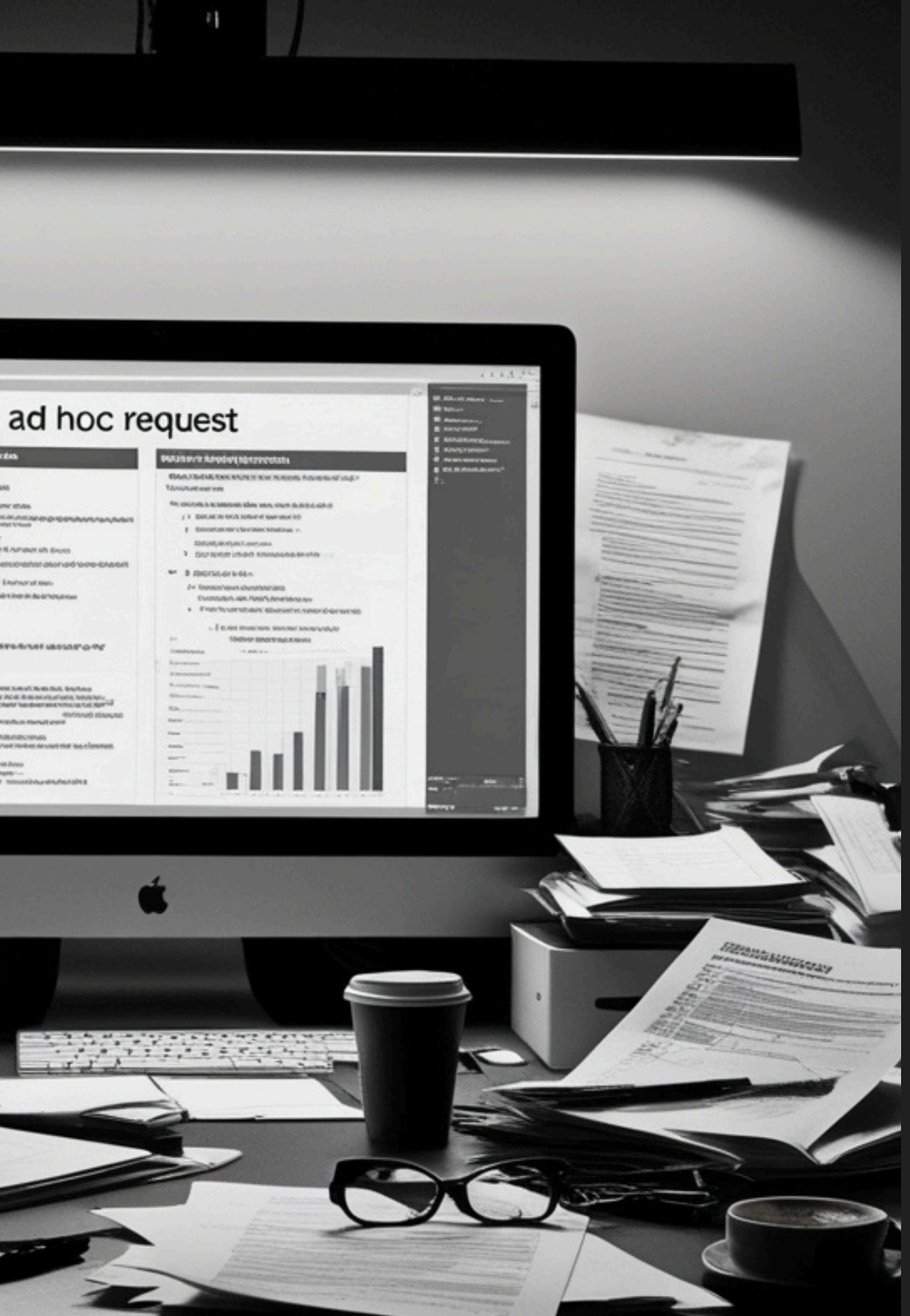
Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region

# QUERY & OUTPUT

```sql
SELECT DISTINCT
    market
FROM
    dim_customer
WHERE
    customer LIKE '%Atliq Exclusive%'
        AND region LIKE '%APAC%';
```

Result Grid

| market |
|---|
| India |
| Indonesia |
| Japan |
| Philiphines |
| South Korea |
| Australia |
| Newzealand |
| Bangladesh |




# *REQUEST 2*

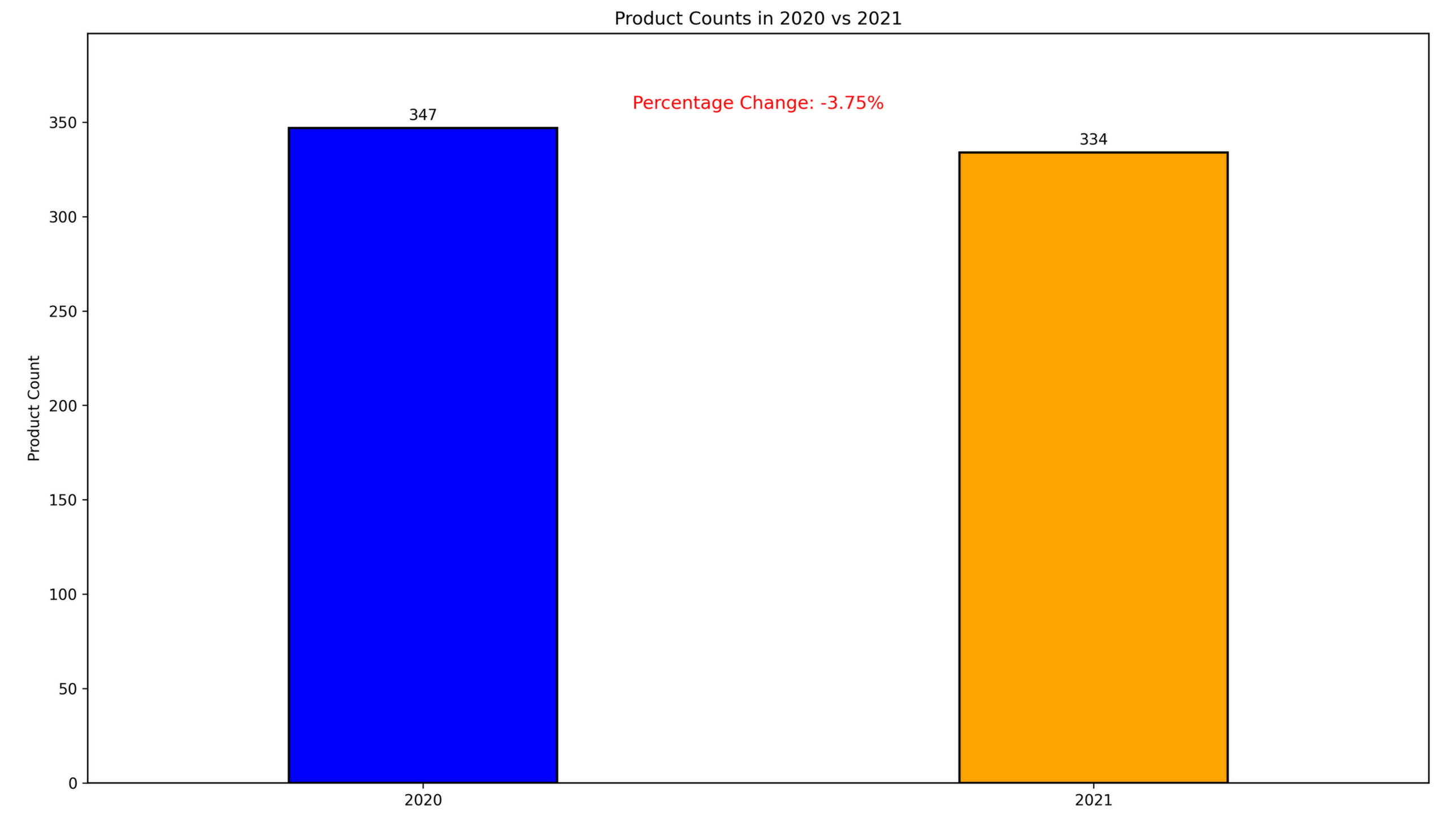
What is the percentage of unique product increase in 2021 vs. 2020?
The final output contains these fields,
-unique_products_2020
-unique_products_2021
- percentage_chg

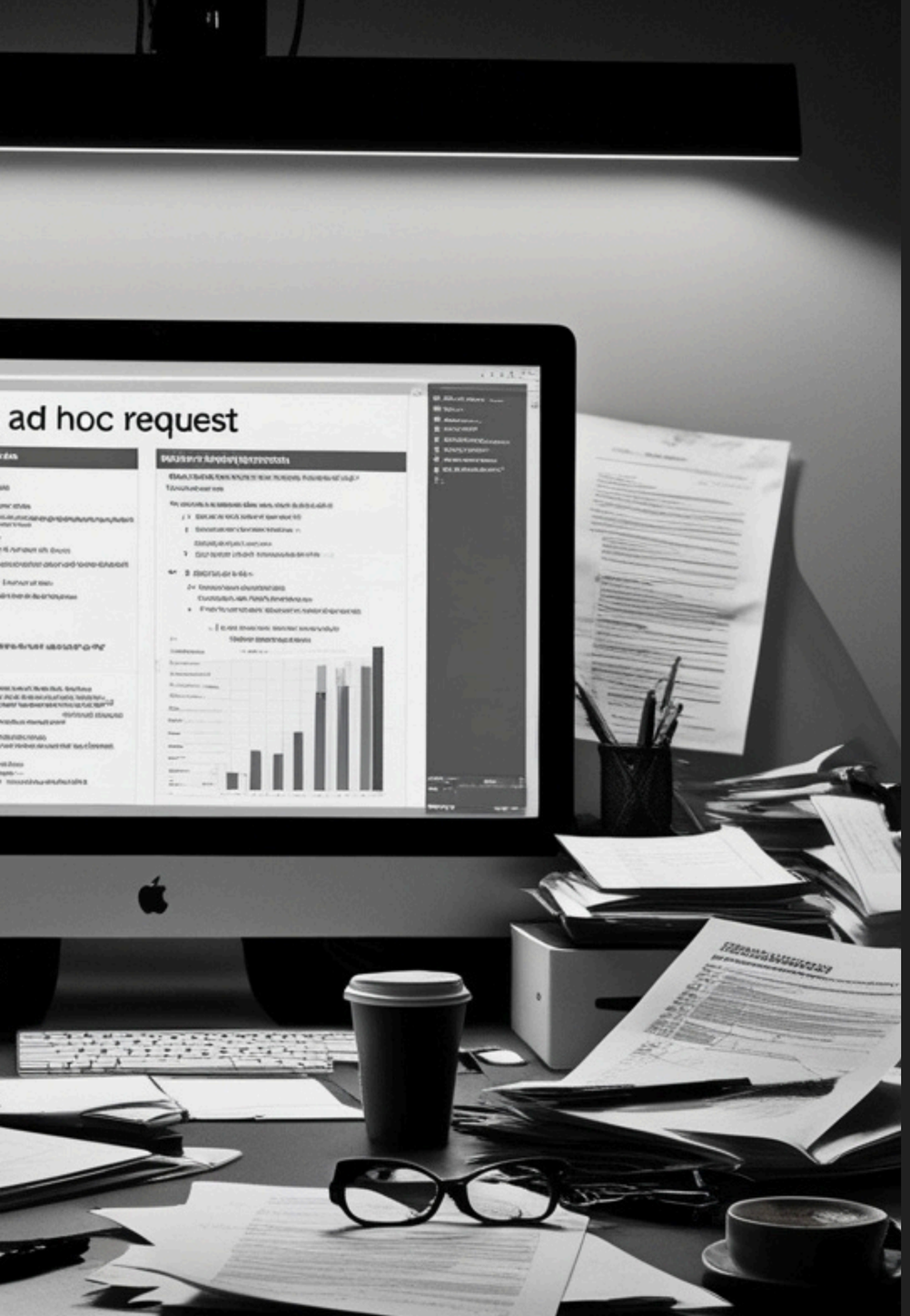
# QUERY & OUTPUT

```sql
WITH cte1 AS (
    SELECT
        COUNT(DISTINCT product_code) AS count_2021
    FROM
        gdb023.fact_sales_monthly
    WHERE
        EXTRACT(YEAR FROM date) = 2021
),
cte2 AS (
    SELECT
        COUNT(DISTINCT product_code) AS count_2020
    FROM
        gdb023.fact_sales_monthly
    WHERE
        EXTRACT(YEAR FROM date) = 2020
)
SELECT
    cte1.count_2021,
    cte2.count_2020,
    ROUND(
        ((cte1.count_2021 - cte2.count_2020) * 100.0 / cte2.count_2020), 2
    ) AS percentage_chg
FROM
    cte1, cte2;
```

Result Grid | Filter Rows: | Export: | W

| count_2021 | count_2020 | percentage_chg |
|---|---|---|
| 334 | 347 | -3.75 |


Product Counts in 2020 vs 2021
Percentage Change: -3.75%
347
334
Product Count
0
50
100
150
200
250
300
350
2020
2021




# *REQUEST 3*

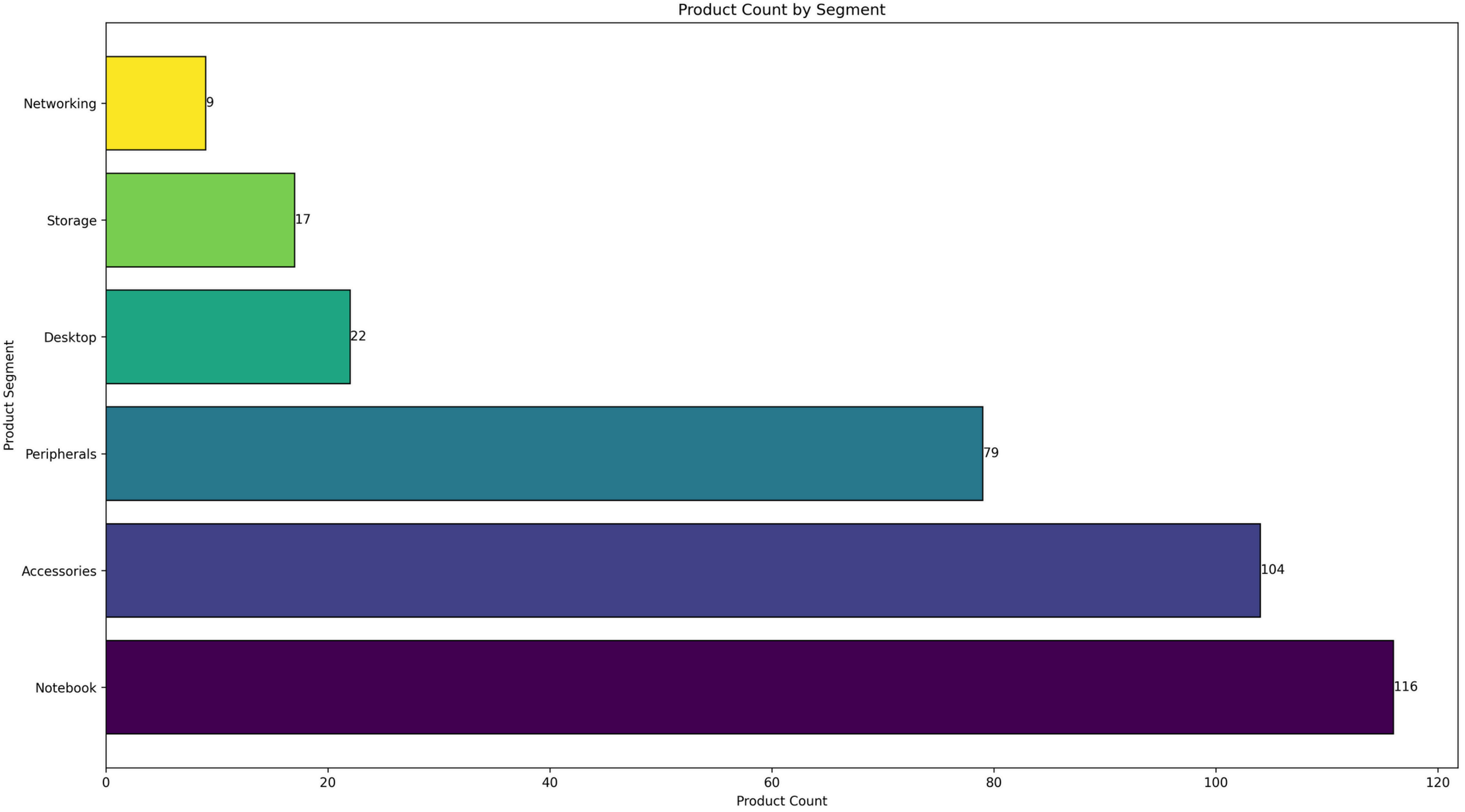
Provide a report with all the unique product counts for each segment and sort them in descending order of product counts.
The final output contains 2 fields,
-segment
-product_count

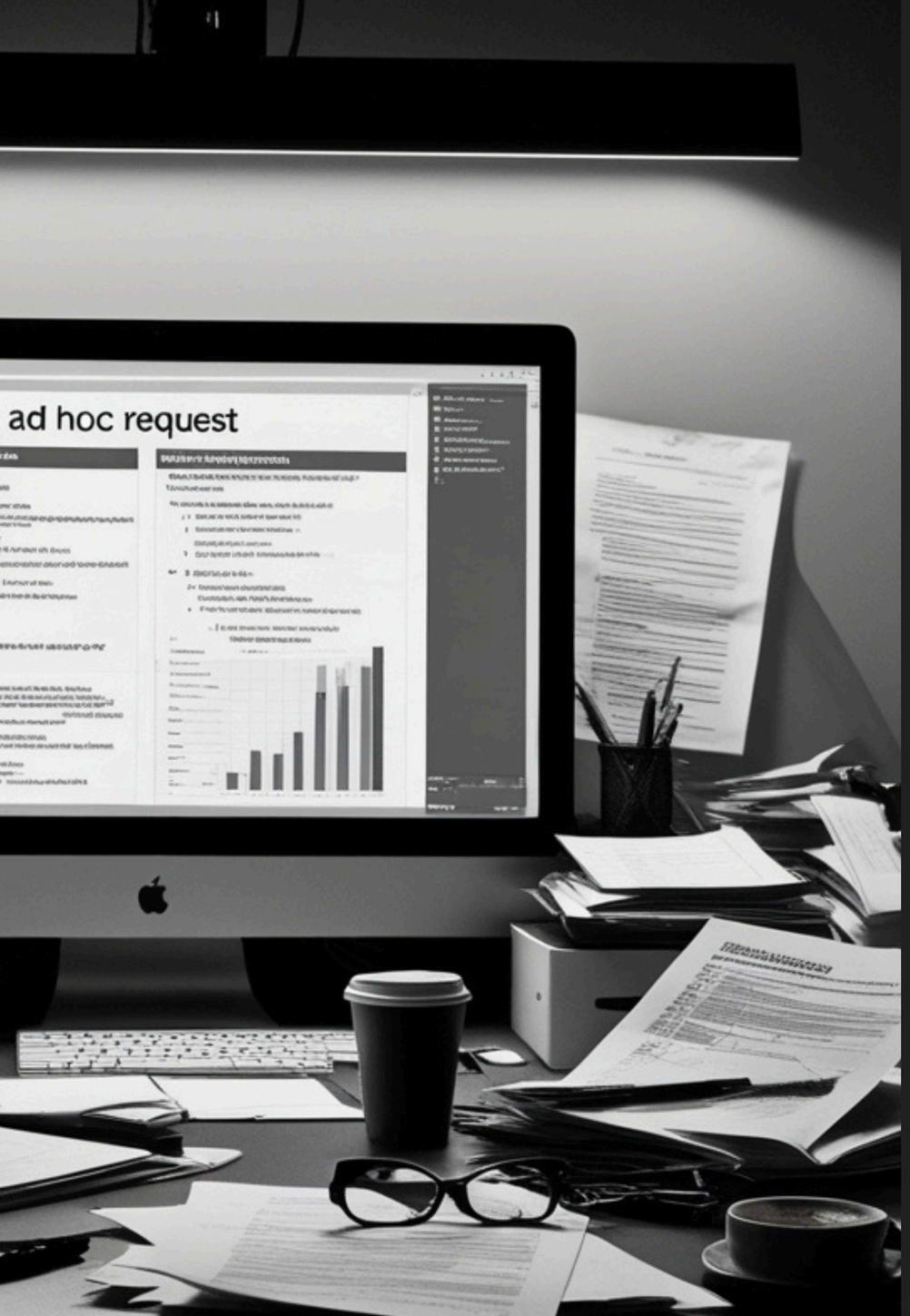
# QUERY & OUTPUT

```sql
select p.segment,
count( distinct s.product_code) as product_count
from fact_sales_monthly s
join dim_product p
on s.product_code = p.product_code
group by p.segment
order by product_count desc
```

Result Grid | Filter Rows:

| segment | product_count |
|---|---|
| Notebook | 116 |
| Accessories | 104 |
| Peripherals | 79 |
| Desktop | 22 |
| Storage | 17 |
| Networking | 9 |


Product Count by Segment
Networking
Storage
Desktop
Peripherals
Accessories
Notebook
9
17
22
79
104
116
Product Segment
0
20
40
60
80
100
120
Product Count




# *REQUEST 4*

Which segment had the most increase in unique products in 2021 vs 2020?

The final output contains these fields,

-segment

-product_count_2020

-product_count_2021

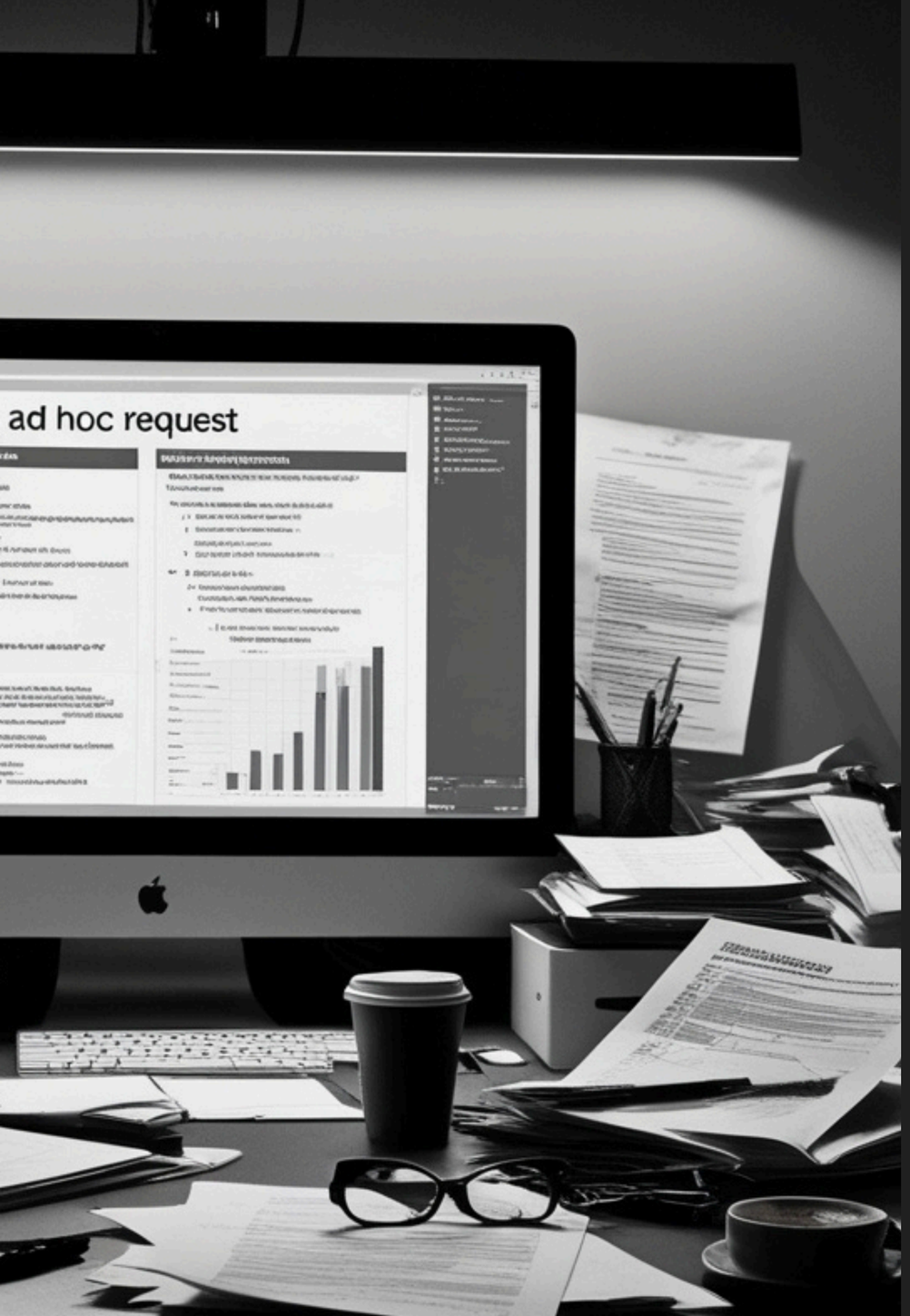
-difference

# QUERY & OUTPUT

```sql
WITH cte1 AS (
    SELECT p.segment,
    COUNT(DISTINCT s.product_code) AS product_count_21
    FROM fact_sales_monthly s
    JOIN dim_product p ON s.product_code = p.product_code
    WHERE EXTRACT(YEAR FROM s.date) = 2021
    GROUP BY p.segment),
cte2 AS (
    SELECT p.segment,
    COUNT(DISTINCT s.product_code) AS product_count_20
    FROM fact_sales_monthly s
    JOIN dim_product p ON s.product_code = p.product_code
    WHERE EXTRACT(YEAR FROM s.date) = 2020
    GROUP BY p.segment)
SELECT
    cte2.segment,
    cte1.product_count_21,
    cte2.product_count_20,
    Abs(cte1.product_count_21-cte2.product_count_20) as abs_difference,
    ROUND(
        (abs((cte1.product_count_21 - cte2.product_count_20)) * 100.0 / abs(cte2.product_count_20)), 2
    ) AS abs_percentage_change
FROM cte1
JOIN cte2 ON cte1.segment = cte2.segment
order by abs_difference desc;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

| segment | product_count_21 | product_count_20 | abs_difference | abs_perc |
|---|---|---|---|---|
| Notebook | 108 | 116 | 8 | 6.90 |
| Peripherals | 75 | 79 | 4 | 5.06 |
| Accessories | 103 | 104 | 1 | 0.96 |
| Desktop | 22 | 22 | 0 | 0.00 |
| Networking | 9 | 9 | 0 | 0.00 |
| Storage | 17 | 17 | 0 | 0.00 |




# *REQUEST 5*

Get the products that have the highest and lowest manufacturing costs.
The final output should contain these fields,

-product_code

-product

-manufacturing_cost

# QUERY & OUTPUT

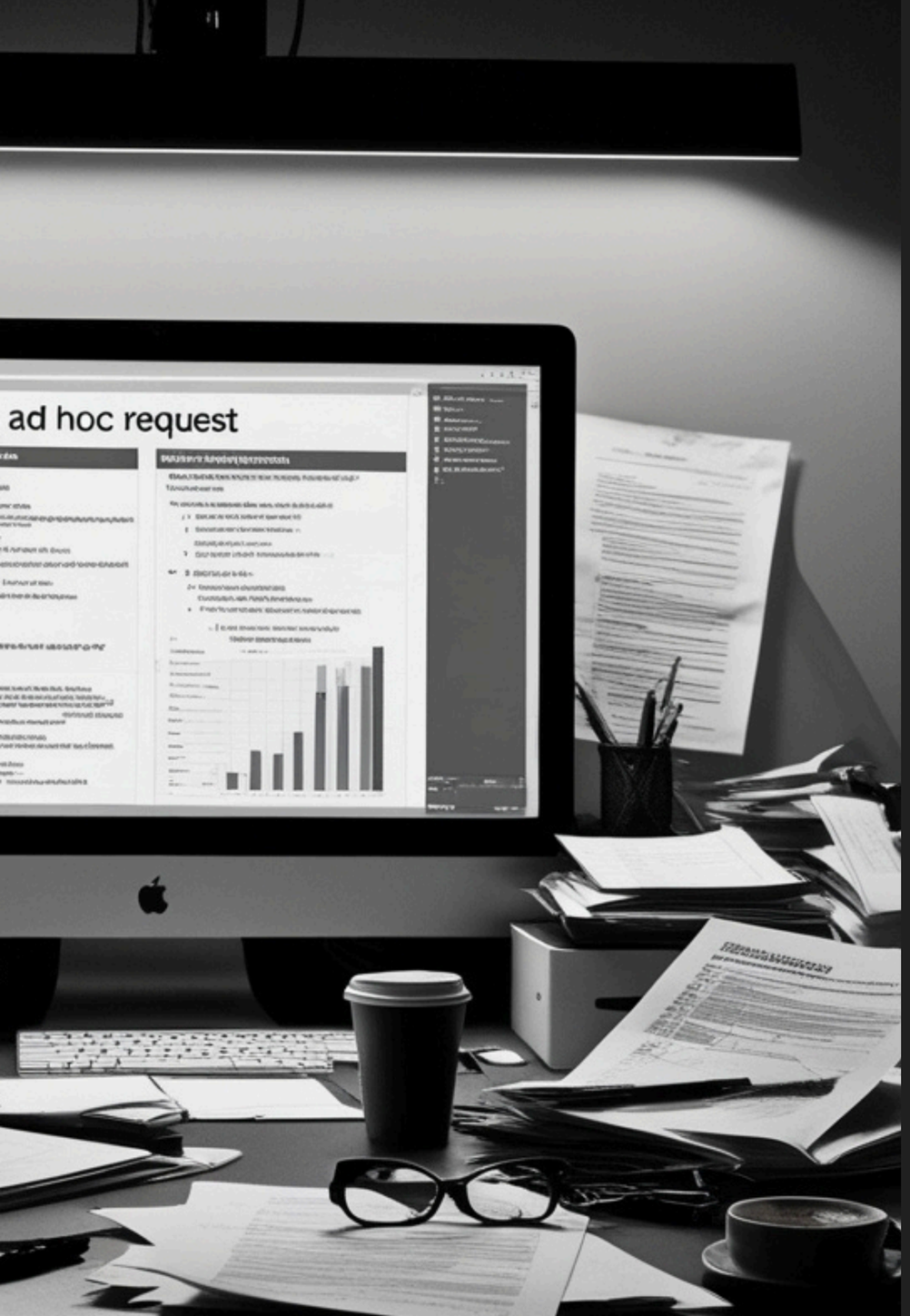
```sql
select * from (SELECT p.product_code, p.product,
    m.manufacturing_cost
FROM dim_product p
JOIN fact_manufacturing_cost m
ON p.product_code = m.product_code
ORDER BY m.manufacturing_cost ASC
LIMIT 1) as min_price

UNION ALL

select * from (SELECT p.product_code, p.product,
    m.manufacturing_cost
FROM dim_product p
JOIN fact_manufacturing_cost m
ON p.product_code = m.product_code
ORDER BY m.manufacturing_cost DESC
LIMIT 1) as max_price;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

| product_code | product | manufacturing_cost |
|---|---|---|
| A2118150101 | AQ Master wired x1 Ms | 0.8920 |
| A6120110206 | AQ HOME Allin1 Gen 2 | 240.5364 |




# *REQUEST 6*

Generate a report which contains the top 5 customers who received an average high pre_invoice_discount_pct for the fiscal year 2021 and in the Indian market. The final output contains these fields,

-customer_code

-customer

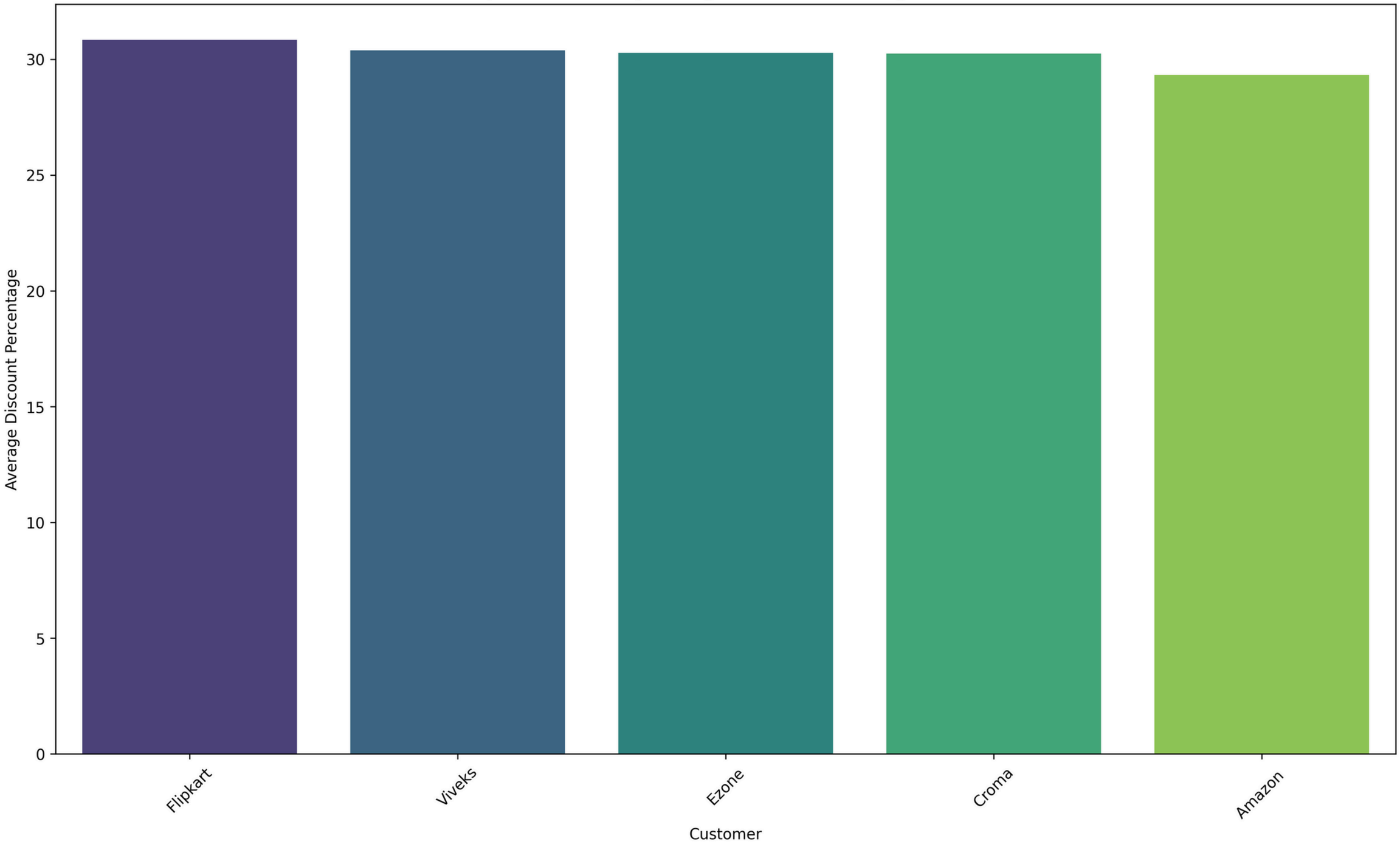
-average_discount_percentage

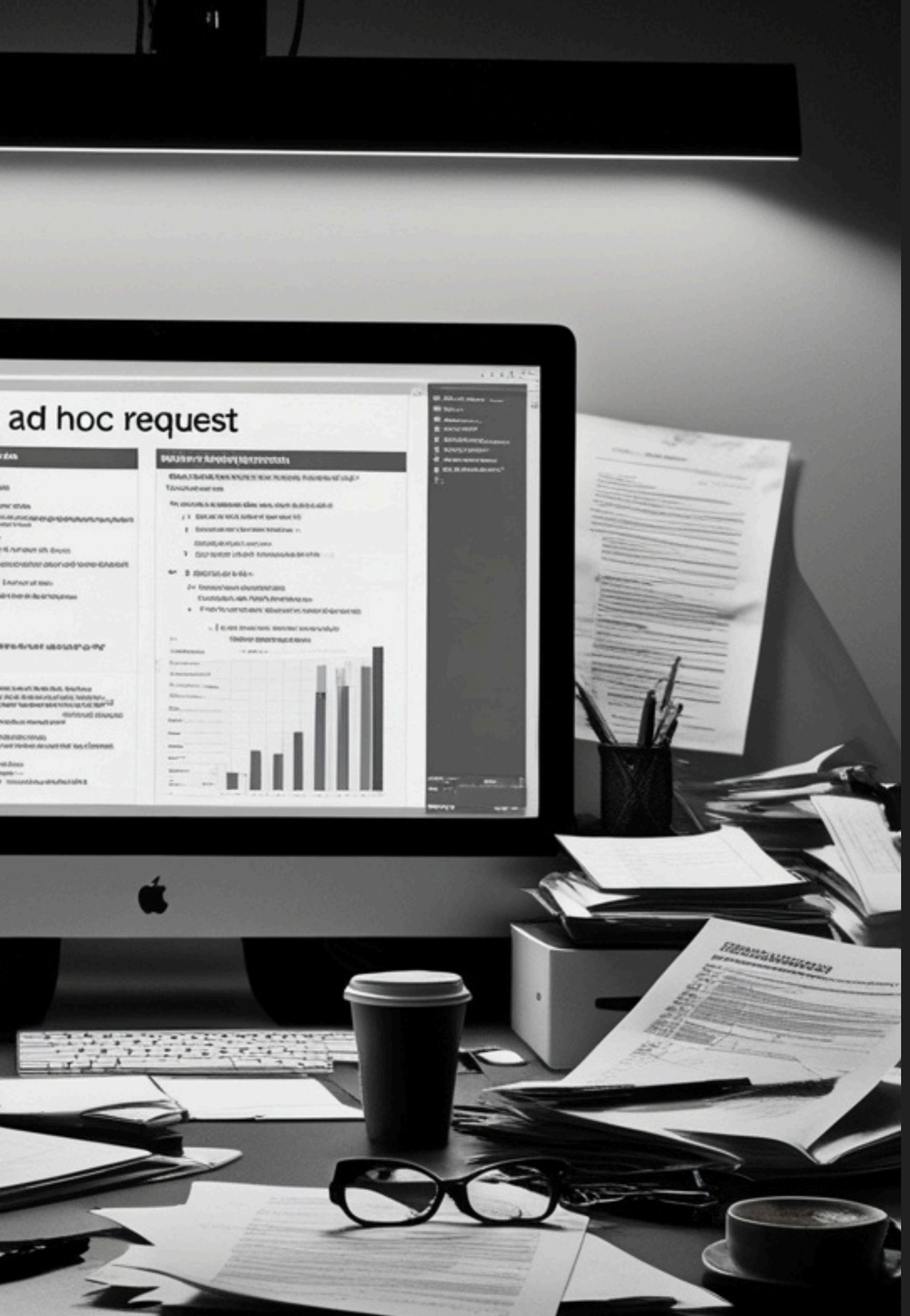
# QUERY & OUTPUT

```sql
SELECT
    c.customer_code,
    c.customer,
    ROUND(AVG(pid.pre_invoice_discount_pct), 4) AS average_discount_percentage
FROM
    dim_customer c
        JOIN
    fact_pre_invoice_deductions pid ON c.customer_code = pid.customer_code
WHERE
    pid.fiscal_year = 2021
        AND c.market = 'India'
GROUP BY c.customer_code , c.customer
ORDER BY average_discount_percentage DESC
LIMIT 5
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

| customer_code | customer | average_discount_percentage |
|---|---|---|
| 90002009 | Flipkart | 0.3083 |
| 90002006 | Viveks | 0.3038 |
| 90002003 | Ezone | 0.3028 |
| 90002002 | Croma | 0.3025 |
| 90002016 | Amazon | 0.2933 |


Top 5 Customers by Average Discount Percentage (2021)
Average Discount Percentage
0
5
10
15
20
25
30
Flipkart
Viveks
Ezone
Croma
Amazon
Customer




# *REQUEST 7*

Get the complete report of the Gross sales amount for the customer “Atliq Exclusive” for each month. This analysis helps to get an idea of low and high-performing months and take strategic decisions. The final report contains these columns:

-Month

-Year

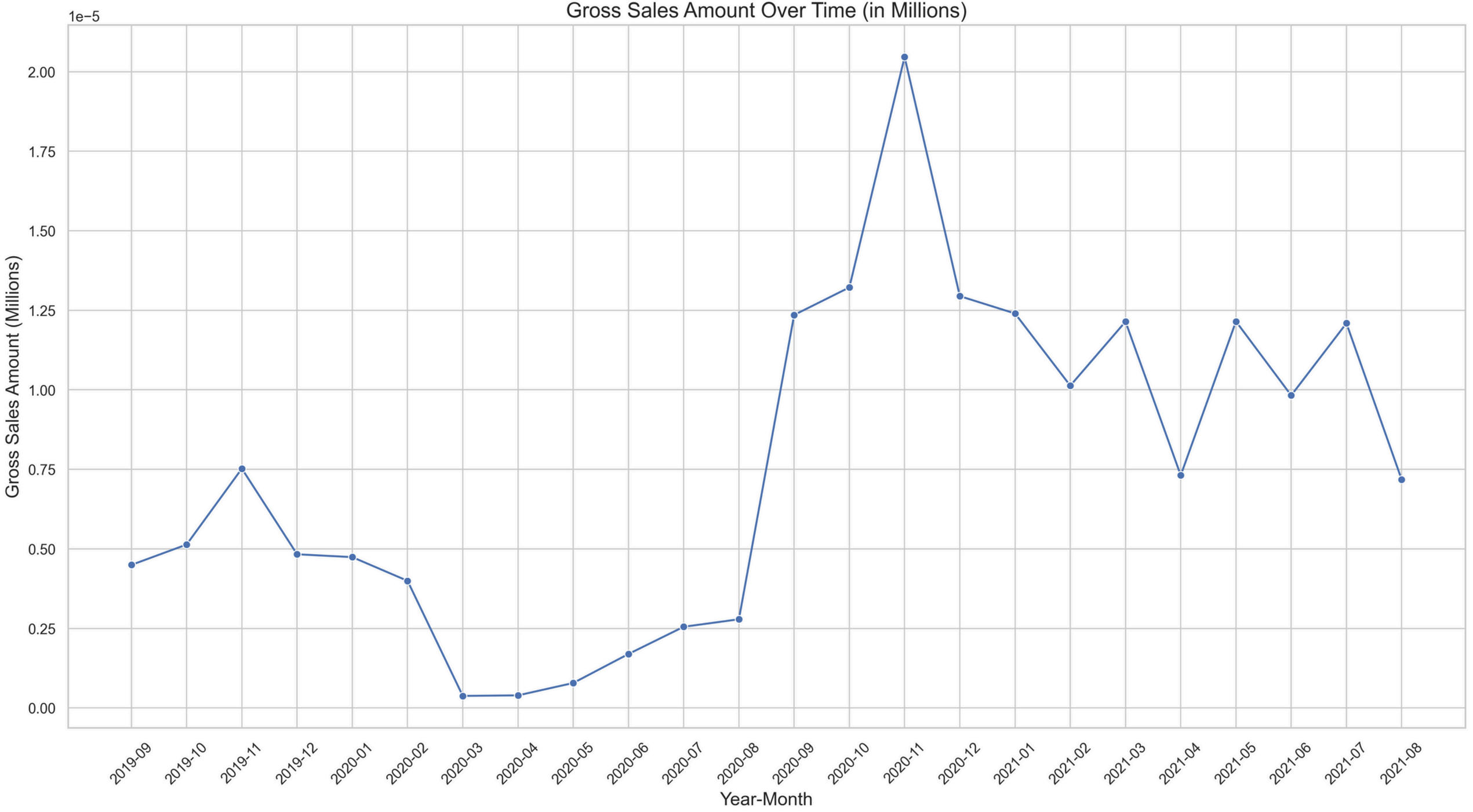
-Gross sales Amount

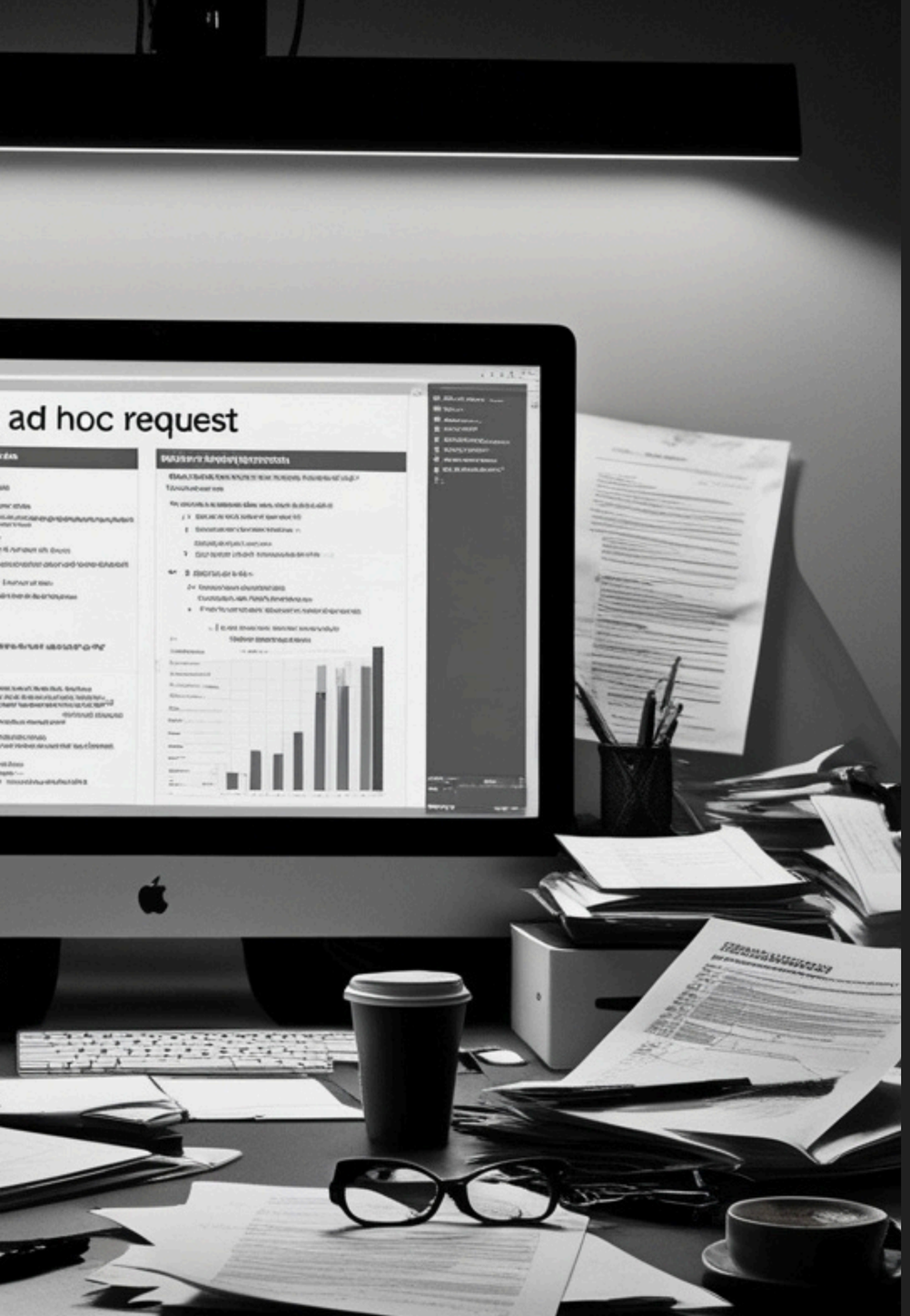
# QUERY & OUTPUT

```sql
SELECT
    YEAR(s.date) AS Year,
    MONTH(s.date) AS Month,
    SUM(s.sold_quantity * g.gross_price) AS Gross_sales_Amount
FROM
    fact_sales_monthly s
        JOIN
    fact_gross_price g ON s.product_code = g.product_code
        AND s.fiscal_year = g.fiscal_year
        JOIN
    dim_customer c ON s.customer_code = c.customer_code
WHERE
    c.customer LIKE '%Atliq Exclusive%'
GROUP BY Year , Month
ORDER BY Year , Month;
```

Result Grid | Filter Rows: | Expo

| Year | Month | Gross_sales_Amount |
|---|---|---|
| 2019 | 9 | 4496259.6724 |
| 2019 | 10 | 5135902.3467 |
| 2019 | 11 | 7522892.5608 |
| 2019 | 12 | 4830404.7285 |
| 2020 | 1 | 4740600.1605 |
| 2020 | 2 | 3996227.7661 |
| 2020 | 3 | 378770.9700 |
| 2020 | 4 | 395035.3535 |
| 2020 | 5 | 783813.4238 |
| 2020 | 6 | 1695216.6008 |
| 2020 | 7 | 2551159.1584 |
| 2020 | 8 | 2786648.2601 |
| 2020 | 9 | 12353509.7938 |
| 2020 | 10 | 13218636.1966 |
| 2020 | 11 | 20464999.0997 |
| 2020 | 12 | 12944659.6509 |
| 2021 | 1 | 12399392.9788 |
| 2021 | 2 | 10129735.5675 |
| 2021 | 3 | 12144061.2501 |
| 2021 | 4 | 7311999.9547 |
| 2021 | 5 | 12150225.0139 |
| 2021 | 6 | 9824521.0110 |
| 2021 | 7 | 12092346.3245 |


Gross Sales Amount Over Time (in Millions)
1e−5
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Gross Sales Amount (Millions)
2019-09
2019-10
2019-11
2019-12
2020-01
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01
2021-02
2021-03
2021-04
2021-05
2021-06
2021-07
2021-08
Year-Month




# *REQUEST 8*

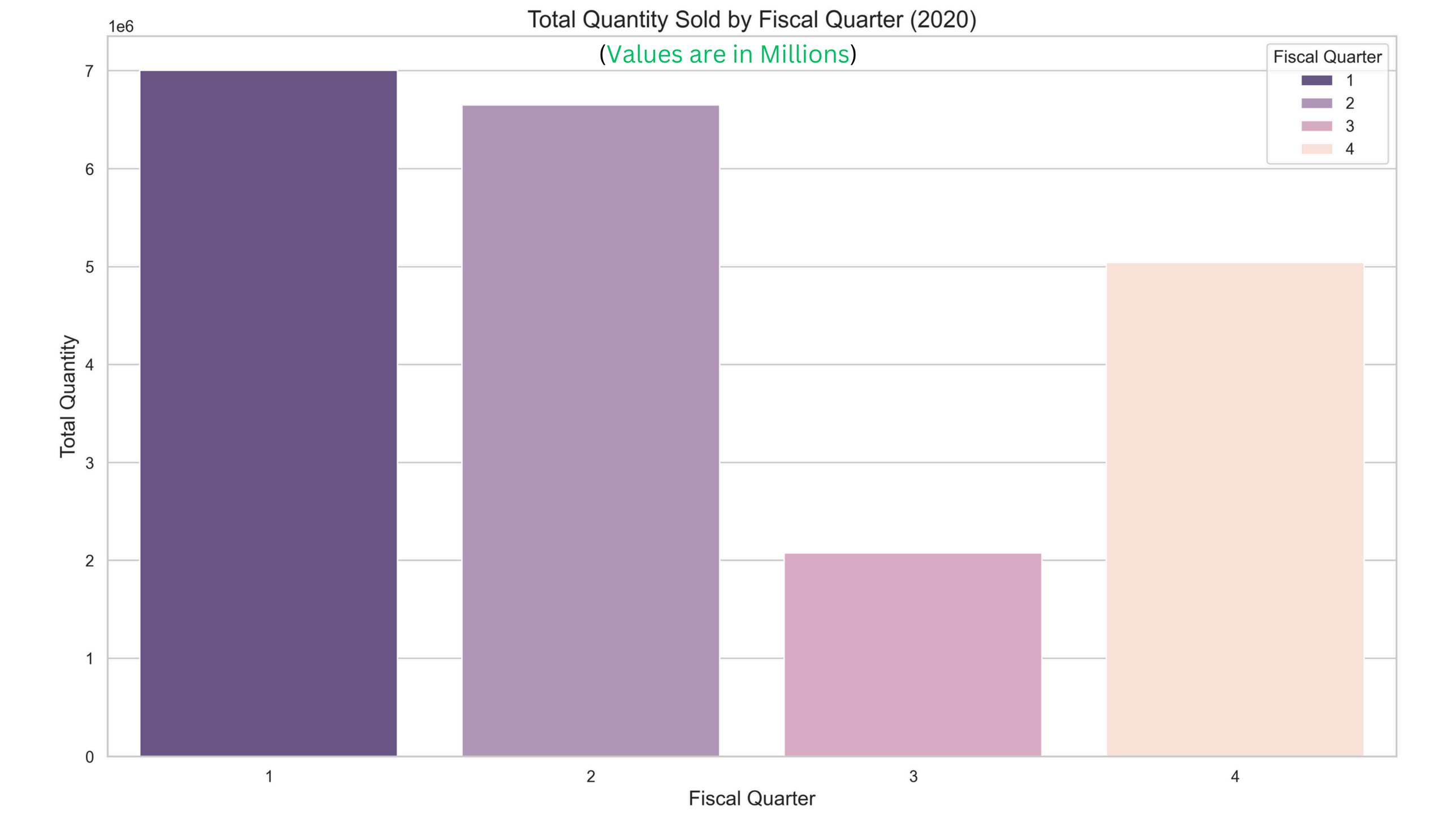
In which quarter of 2020, got the maximum total_sold_quantity?
The final output contains these fields sorted by the total_sold_quantity,
-Quarter
-total_sold_quantity

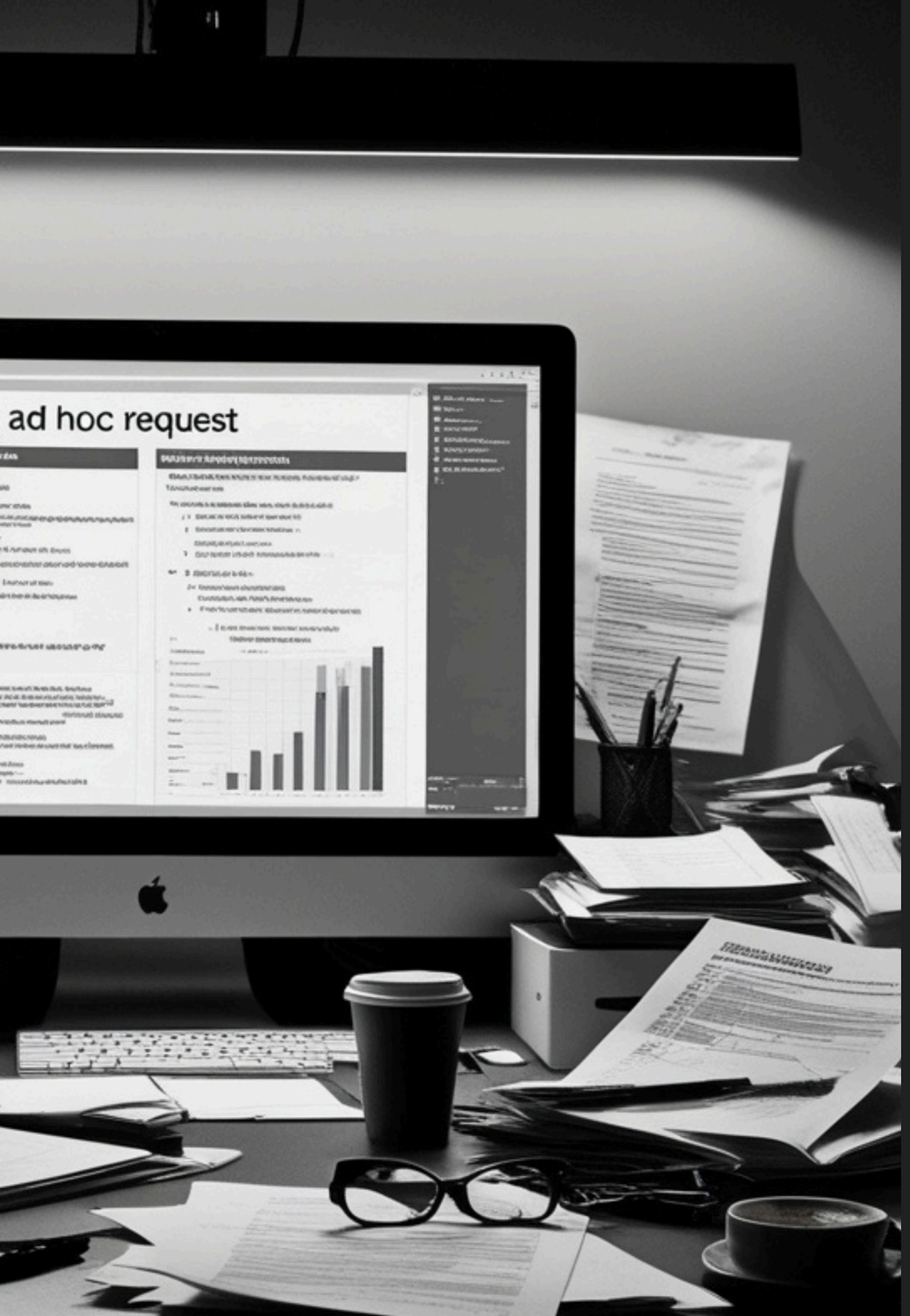
# QUERY & OUTPUT

```sql
SELECT
  CASE
    WHEN MONTH(date) IN (9, 10, 11) THEN 1
    WHEN MONTH(date) IN (12, 1, 2) THEN 2
    WHEN MONTH(date) IN (3, 4, 5) THEN 3
    ELSE 4
  END AS fiscal_quarter,
  SUM(sold_quantity) AS total_quantity
FROM fact_sales_monthly
WHERE fiscal_year = 2020
GROUP BY fiscal_quarter;
```

Result Grid | Filter Rows:

| fiscal_quarter | total_quantity |
|---|---|
| 1 | 7005619 |
| 2 | 6649642 |
| 3 | 2075087 |
| 4 | 5042541 |


Total Quantity Sold by Fiscal Quarter (2020)
(Values are in Millions)
1e6
Total Quantity
0
1
2
3
4
5
6
7
Fiscal Quarter
1
2
3
4




# *REQUEST 9*

Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution?

The final output contains these fields,

-channel

-gross_sales_mln

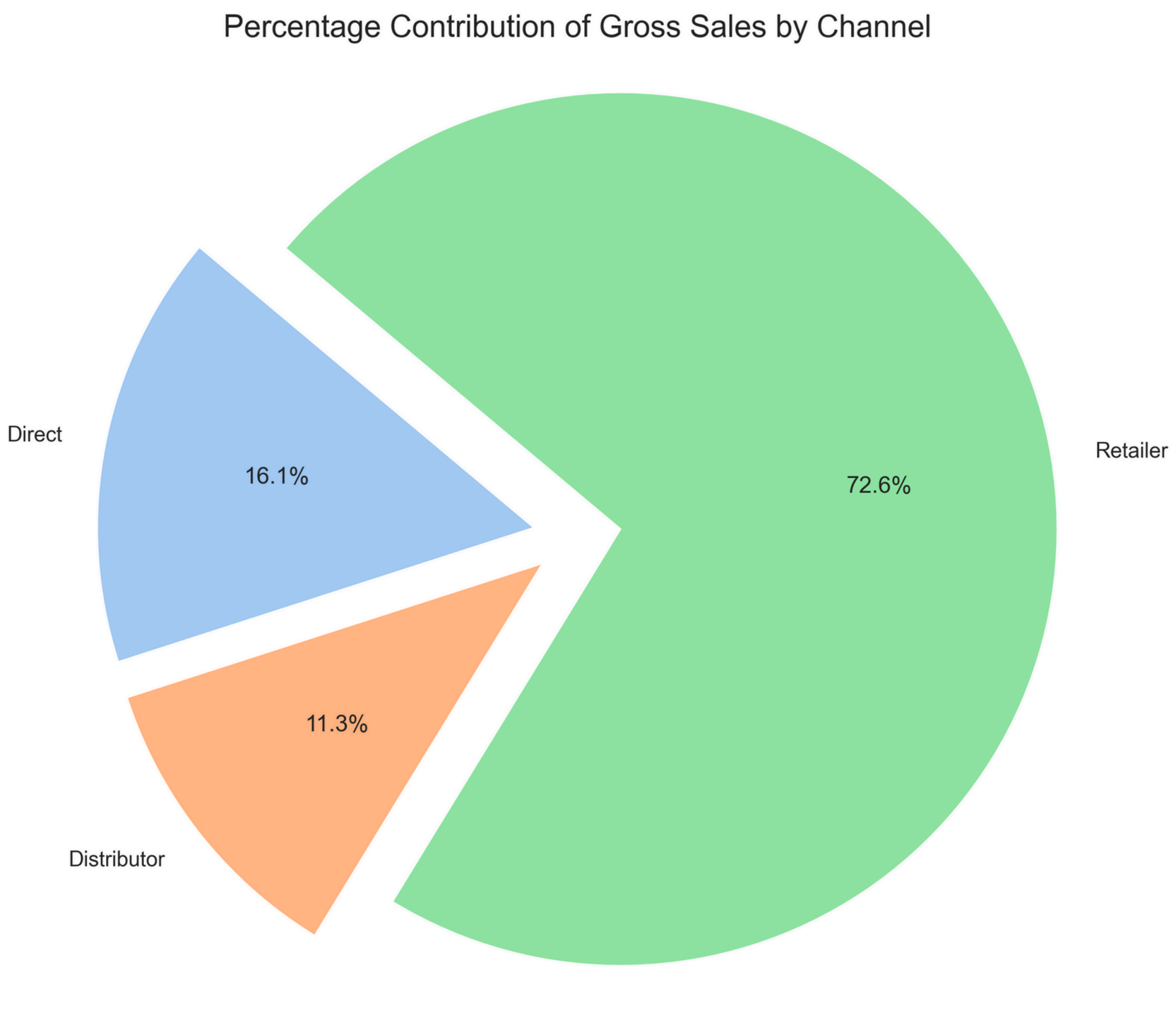
-percentage

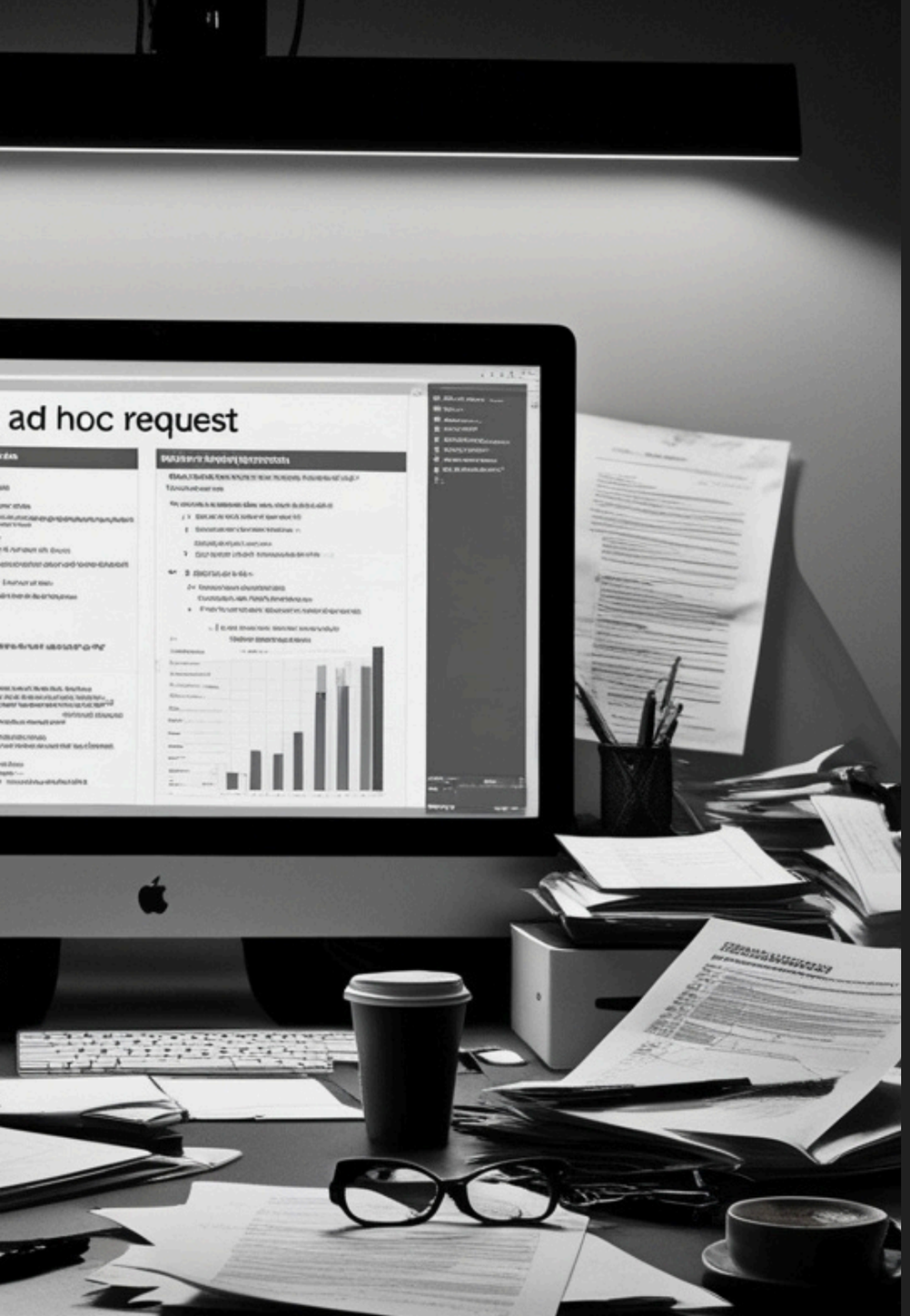
# QUERY & OUTPUT

```sql
WITH sales_data AS (
    SELECT
        c.channel,
        round(SUM(s.sold_quantity * g.gross_price) / 1000000,2) AS gross_sales_mln
    FROM
        fact_sales_monthly s
    JOIN
        dim_customer c ON s.customer_code = c.customer_code
    JOIN
        fact_gross_price g ON s.product_code = g.product_code
                           AND s.fiscal_year = g.fiscal_year
    GROUP BY
        c.channel
)
SELECT
    channel,
    gross_sales_mln,
    round((gross_sales_mln / SUM(gross_sales_mln)over()) * 100,2) AS pct_contribution
FROM
    sales_data;
```

Result Grid | Filter Rows: | Export: | W

| channel | gross_sales_mln | pct_contribution |
|---|---|---|
| Direct | 353.96 | 16.08 |
| Distributor | 248.47 | 11.29 |
| Retailer | 1598.16 | 72.62 |


Percentage Contribution of Gross Sales by Channel
Direct
16.1%
Distributor
11.3%
Retailer
72.6%




# *REQUEST 10*

Get the Top 3 products in each division that have a high total_sold_quantity in the fiscal_year 2021?

The final output contains these fields,

-division

-product_code

-product

-total_sold_quantity

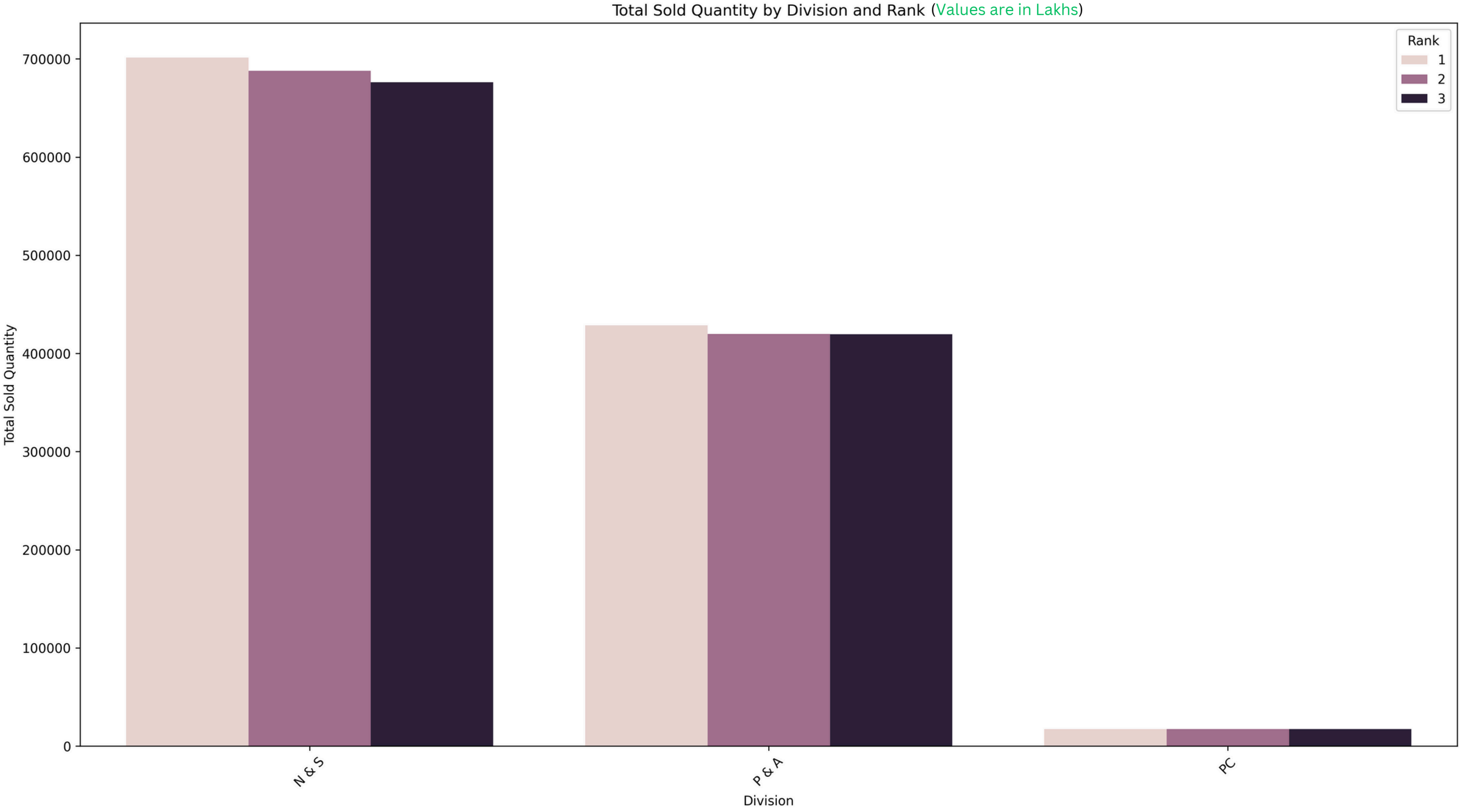
-rank_order

# QUERY & OUTPUT

```sql
WITH cte1 AS (
    SELECT
        p.division AS division,
        p.product_code,
        p.product,
        SUM(s.sold_quantity) AS total_sold_quantity
    FROM
        fact_sales_monthly s
    JOIN
        dim_product p ON s.product_code = p.product_code
    JOIN
        fact_gross_price g ON s.product_code = g.product_code
                          AND s.fiscal_year = g.fiscal_year
    WHERE
        s.fiscal_year = 2021
    GROUP BY
        p.division, p.product_code, p.product
),
ranked_products AS (
    SELECT
        division,
        product_code,
        product,
        total_sold_quantity,
        DENSE_RANK() OVER (PARTITION BY division ORDER BY total_sold_quantity DESC) AS _rank
    FROM
        cte1
)
SELECT *
FROM ranked_products
WHERE
    _rank IN (1, 2, 3);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| division | product_code | product | total_sold_quantity | _rank |
|---|---|---|---|---|
| N & S | A6720160103 | AQ Pen Drive 2 IN 1 | 701373 | 1 |
| N & S | A6818160202 | AQ Pen Drive DRC | 688003 | 2 |
| N & S | A6819160203 | AQ Pen Drive DRC | 676245 | 3 |
| P & A | A2319150302 | AQ Gamers Ms | 428498 | 1 |
| P & A | A2520150501 | AQ Maxima Ms | 419865 | 2 |
| P & A | A2520150504 | AQ Maxima Ms | 419471 | 3 |
| PC | A4218110202 | AQ Digit | 17434 | 1 |
| PC | A4319110306 | AQ Velocity | 17280 | 2 |
| PC | A4218110208 | AQ Digit | 17275 | 3 |


Total Sold Quantity by Division and Rank (Values are in Lakhs)
Rank
1
2
3
700000
600000
500000
400000
300000
200000
100000
0
Total Sold Quantity
N & S
P & A
PC
Division

# THANK YOU