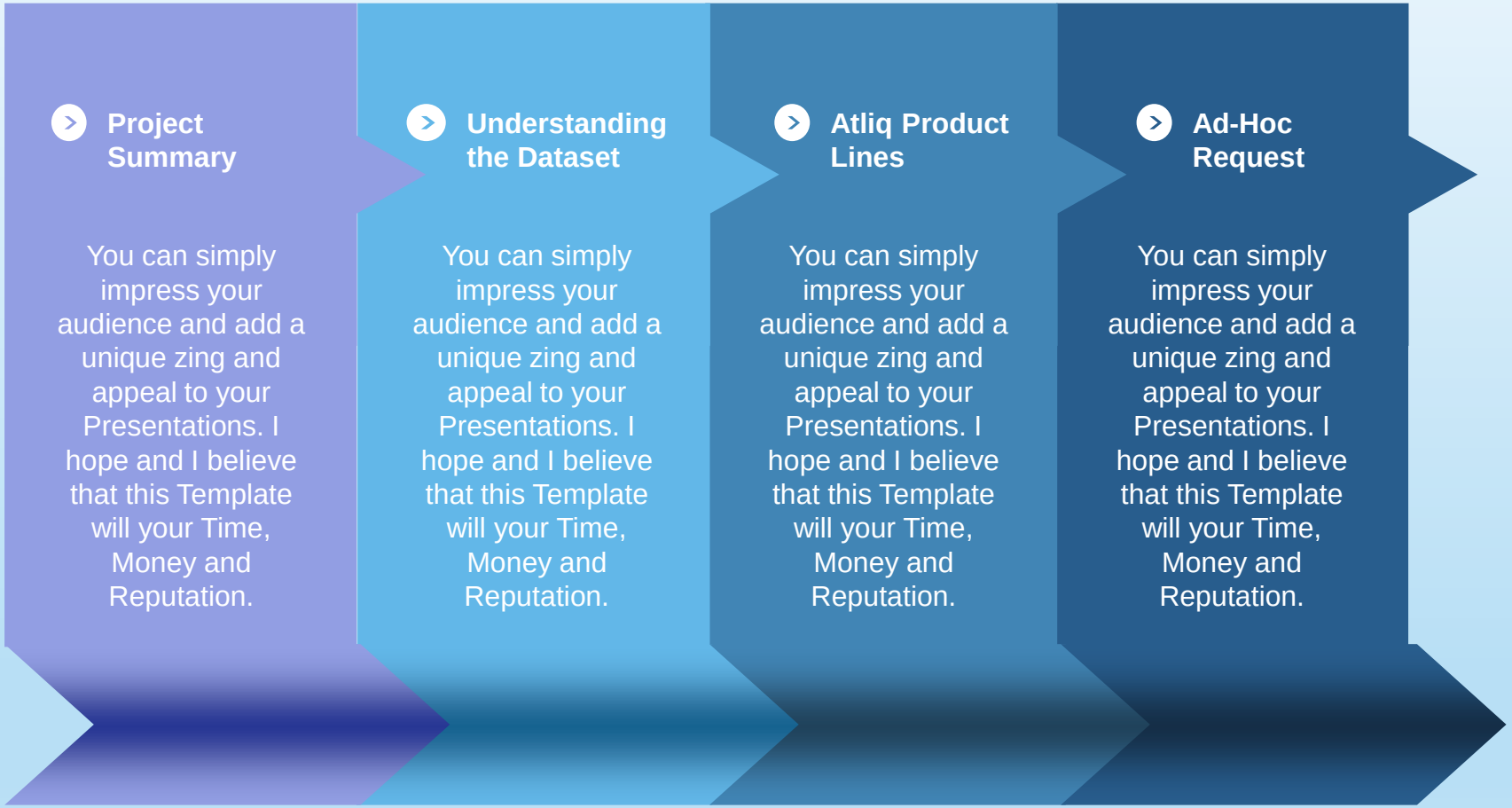
Consumer Goods

AD-Hoc Request

Codesbascis Project Challenge

Presented by Mayank Singh

Table of contents



1 Project Summary

01 About Company

- Atliq Hardware (an imaginary company) is one of India's leading computer hardware producers and has a business well spread in 27 countries across the globe. It has customers in the region like North America (NA), Latin America(LATAM), and Asia-Pacific(APAC)

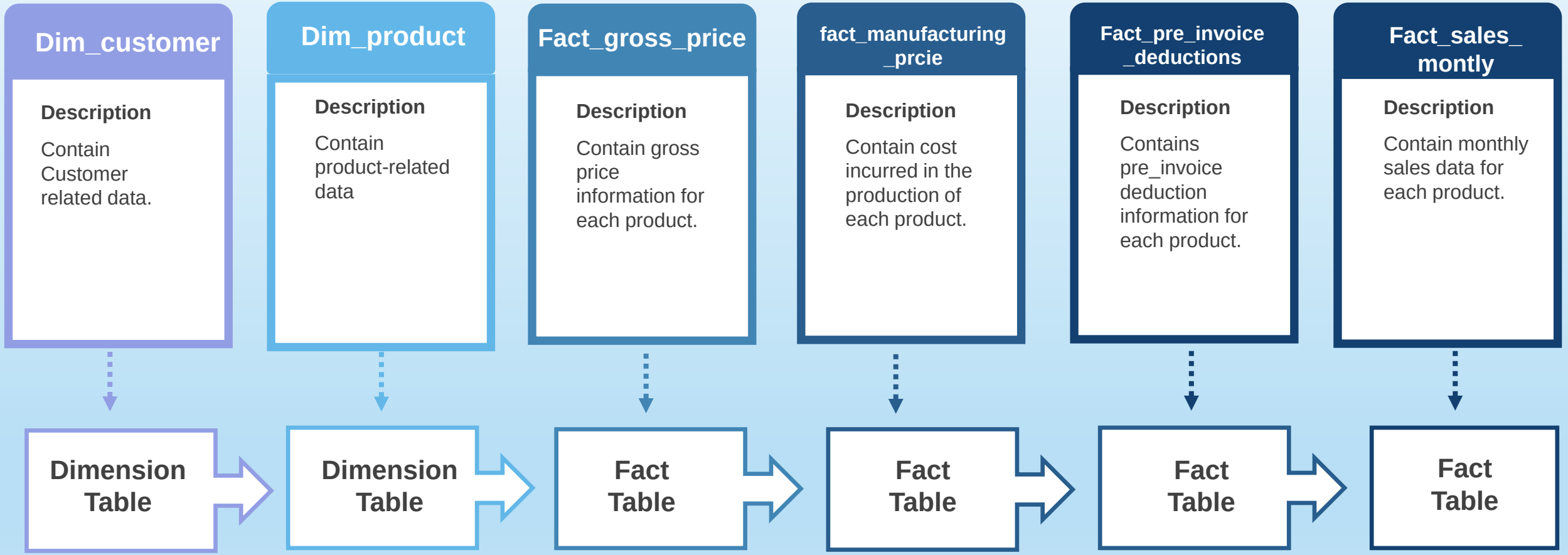
02 Problem Statement

- The management team of the company has identified a need for timely and informed decision-making based on data insights. Consequently, they have requested the data analytics team, to provide ad-hoc solutions to 10 specific problems in order to gain the required insights

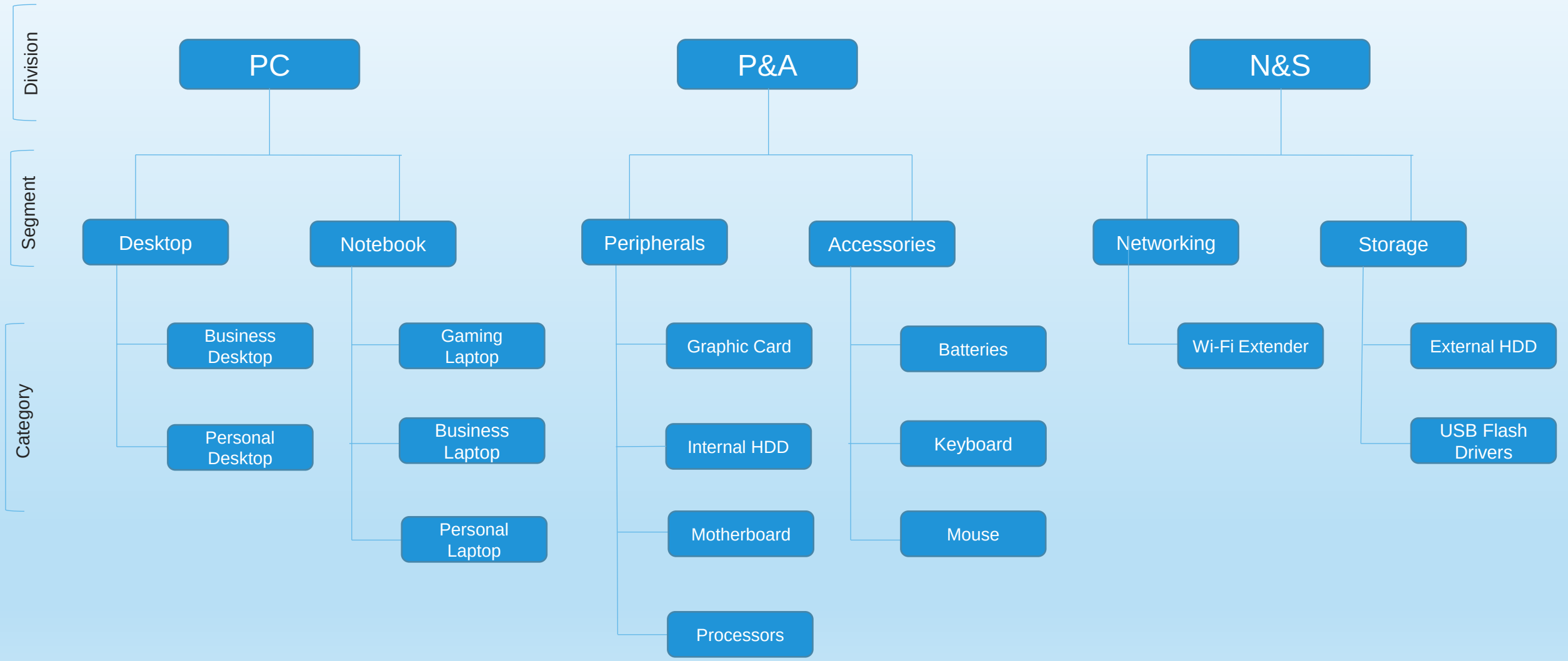
03 Objective

- The task at hand involves executing SQL queries to address 10 significant ad hoc inquiries that provide essential business insights. The intended recipients of the resulting presentation are high-level executives, requiring a thorough and precise depiction of the insights to aid in making critical business decisions.

2 Understanding the Dataset



3 Atliq Product Lines



4 Ad hoc Requests

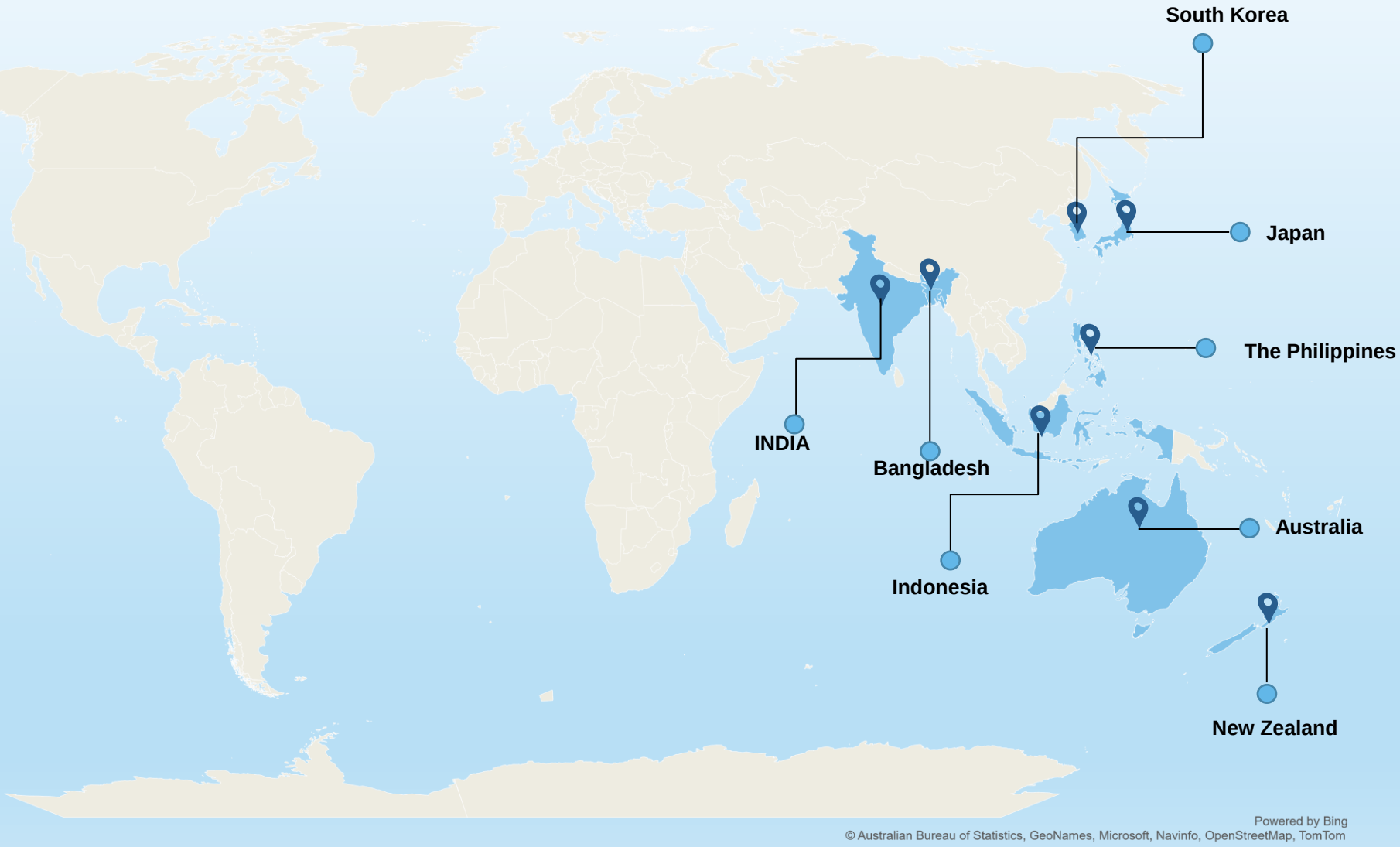
Request 1

List of markets in which customer “Atliq Exclusive” operates its business in the APAC region.

market
India
Indonesia
Japan
Philippines
South Korea
Australia
New Zealand
Bangladesh

Important Insights

"Atliq Exclusive" operates its business in 8 countries of the APAC Region. APAC stands for Asia-Pacific.

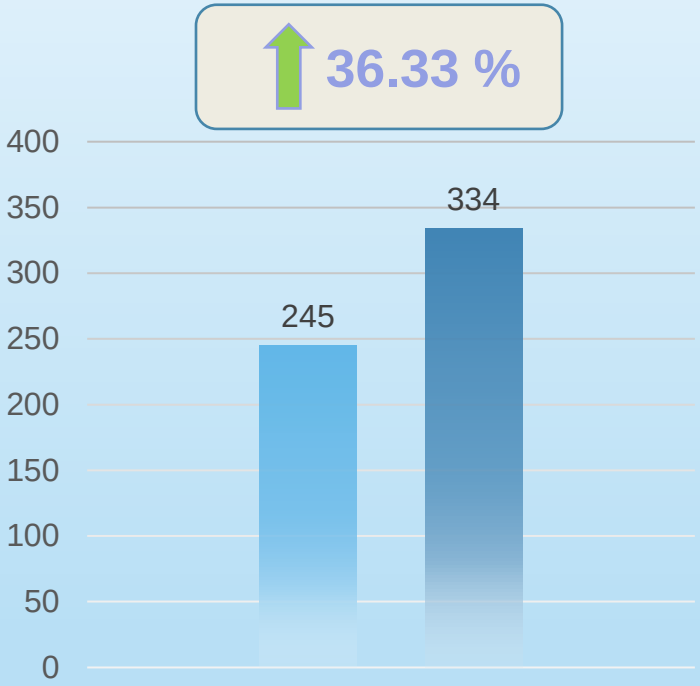


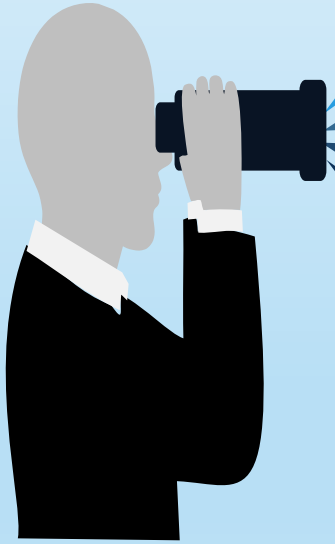
Request 2

Percentage of unique product increase in 2021 vs 2020.

unique_products_2020	unique_products_2021	Percentage chg
245	334	36.33

Important Insights





Atliq Hardware saw a significant rise of 36.33% in the number of unique products recorded in 2021, with 334 products, compared to 245 products in 2020.

This increase suggests that Atliq Hardware remains competitive through the introduction of new products.

Atliq Hardware is committed to enhancing its revenue by enhancing the strengths of its existing products and developing high-demand products.

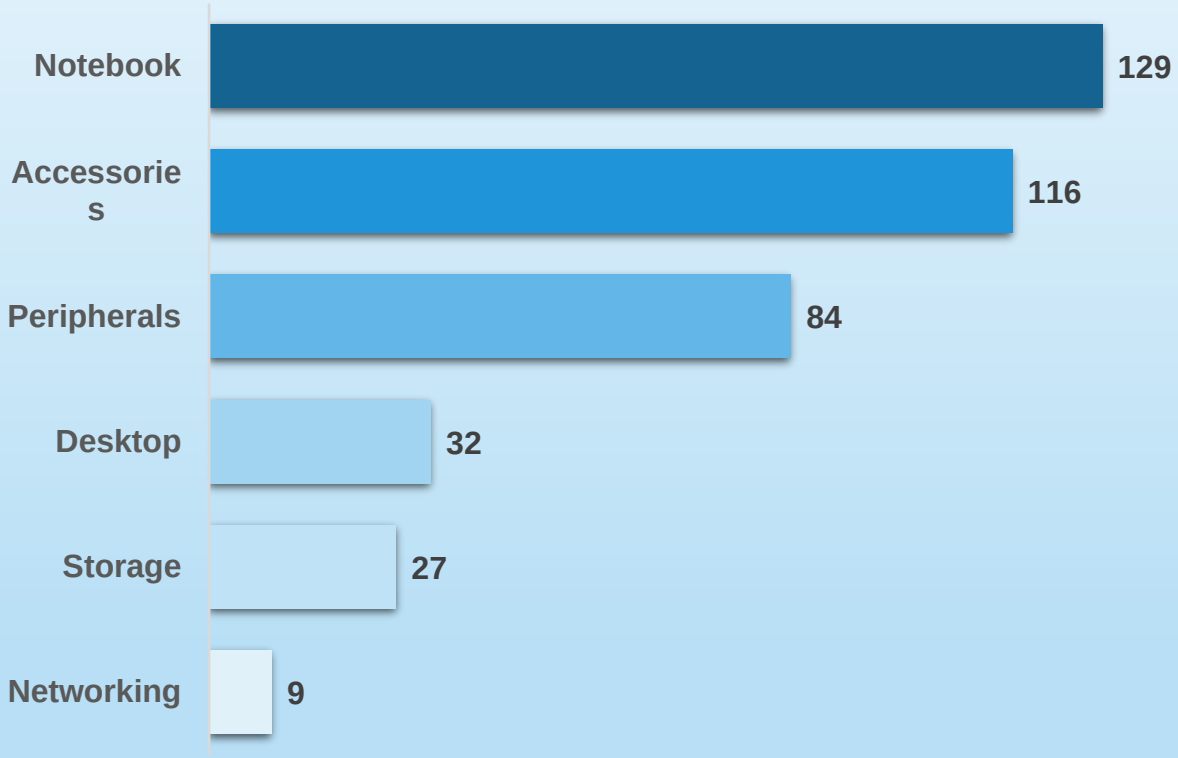
To sustain growth and expand its market share, Atliq Hardware should prioritize product development

Request 3

Total Number of Unique Products for each Segment.

Segment	Product count
Notebook	129
Accessories	116
Peripherals	84
Desktop	32
Storage	27
Networking	9

Important Insights



The highest contributing segment to sales is Notebook, Accessories, and Peripherals, whereas the lowest contributing segment is Desktop, Storage, and Networking.

The average sales of the top 3 contributors were (109.66), while the average sales of the bottom 3 contributors were (22.6).

To sustain production levels, Atliq Hardware should prioritize allocating resources to the top-performing segments.

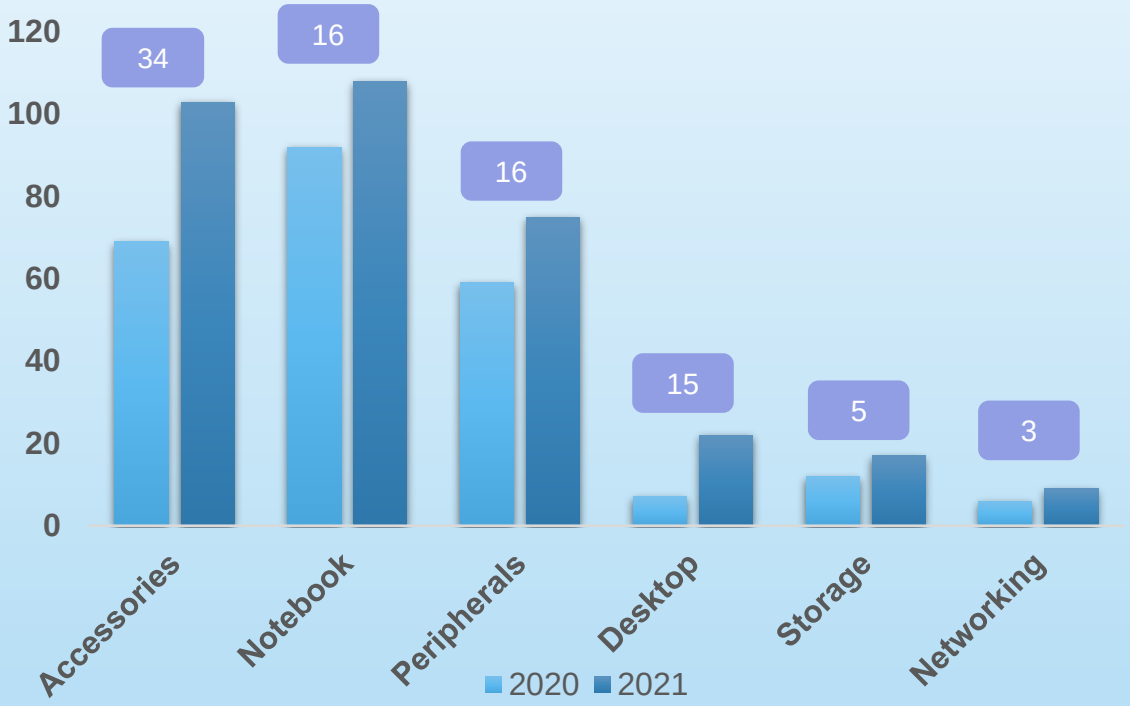
In addition to that, Atliq Hardware should investigate why Desktop, Storage, and Networking have fewer unique products and develop strategies to increase their sales, which would lead to market expansion and revenue growth.

Request 4

The segment that had the most increase in unique products in 2020 vs 2021

segment	2020	2021	difference
Accessories	69	103	34
Notebook	92	108	16
Peripherals	59	75	16
Desktop	7	22	15
Storage	12	17	5
Networking	6	9	3

Important Insights



There has been a significant surge in the number of unique products in the accessories segment, with the addition of 34 new unique products.

The notebook and peripherals segments have also witnessed an increase in unique products, with each segment adding 16 new unique products.

Request 5

The Product that had the highest and lowest manufacturing cost.

product code	product	Manufacturing cost
A2118150101	AQ Master wired x1 Ms	0.892
A6120110206	AQ HOME Allin1 Gen 2	240.5364

Product with the highest and lowest manufacturing cost.

The personal desktop, Product AQ HOME Allin1 Gen 2, incurs the highest manufacturing cost among all the products.

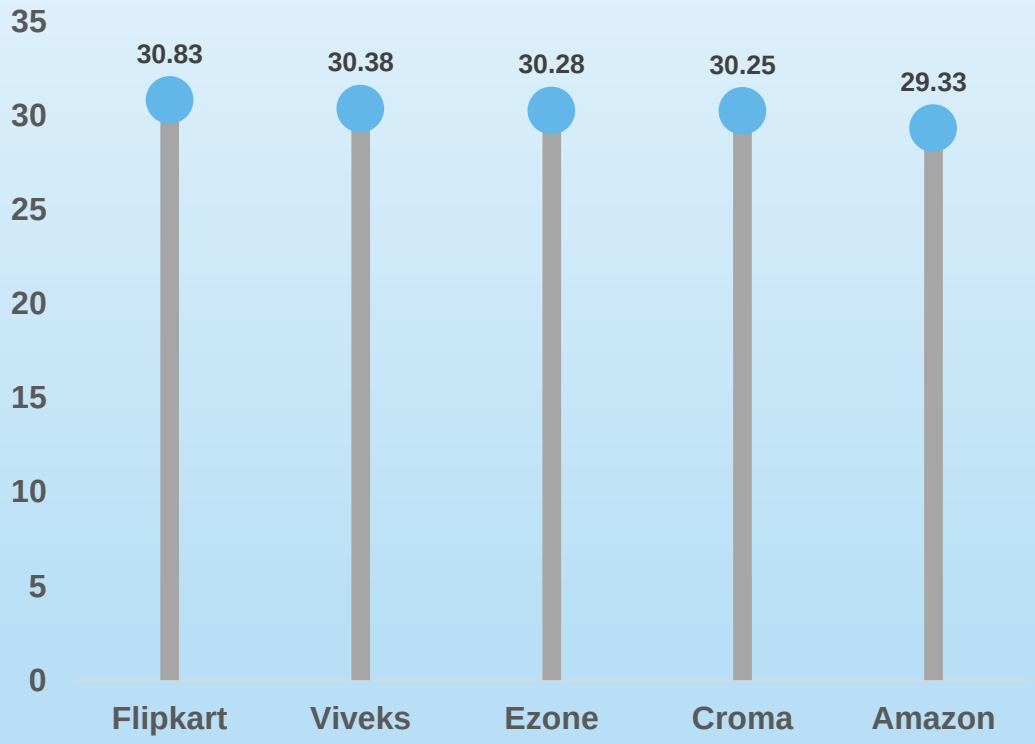
Among all the products, the manufacturing cost of Product AQ Master wired x1 Ms, a wired mouse, is the lowest.

Request 6

Top 5 customers who received an average high pre-invoice discount percent for the fiscal year 2021 and in the Indian market.

Customer code	customer	Average discount percentage
90002009	Flipkart	30.83
90002006	Viveks	30.38
90002003	Ezone	30.28
90002002	Croma	30.25
90002016	Amazon	29.33

Important Insights



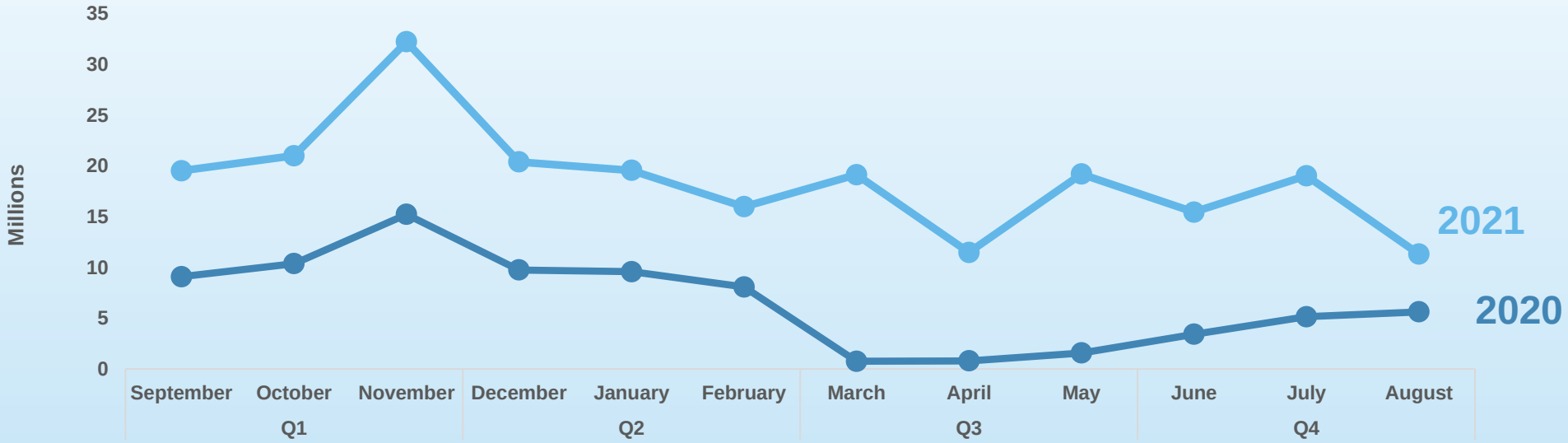
- The company's pre-invoice discount remained consistent for its top 5 customers in 2021, with a difference of only 1.5% between the highest and lowest discount received.
- This could be a deliberate strategy on Atliq Exclusive's part to maintain parity between its major customers and ensure that none of them feel they are receiving a better or worse deal than the others.
- Among these customers, Flipkart received the highest discount, followed by Viveks. while Amazon received the lowest discount in the Indian market.
- It is worth noting, however, that the discount percentage being offered is quite high across the board, with all of Atliq Exclusive's major customers receiving discounts of over 29%. This could suggest that Atliq Exclusive is facing pressure from its competitors and needs to offer significant discounts in order to remain competitive.

Request 7

Gross Sales Amount for "Atliq Exclusive" for each month in 2020 and 2021.

month	2021	2020
November	32247289.79	15231895
October	21016218.21	10378638
December	20409063.18	9755795.1
January	19570701.71	9584951.9
September	19530271.3	9092670.3
May	19204309.41	1586964.5
March	19149624.92	766976.45
July	19044968.82	5151815.4
February	15986603.89	8083995.6
June	15457579.66	3429736.6
April	11483530.3	800071.95
August	11324548.34	5638281.8

Important Insights

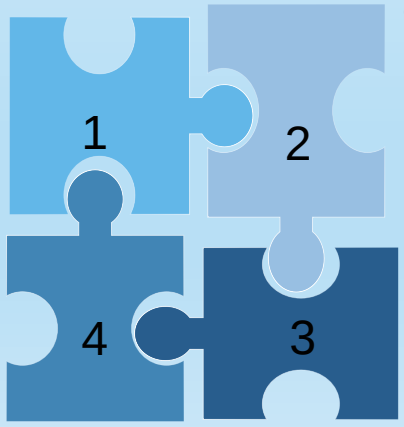


Sales Growth

The gross sales of Atliq Exclusive have shown significant growth in 2021 compared to 2020. In particular, Q1 and Q3 of 2021 have seen a substantial increase in sales compared to the same quarter in 2020. This could be due to several factors, including an increase in consumer demand or effective marketing and sales strategies.

Seasonal Trend

There appears to be a seasonal trend in the gross sales of Atliq Exclusive, with higher sales during the months of September to November, which fall in Q1, and lower sales during the months of March to May, which fall in Q3. This trend may be due to various reasons, including seasonal shopping trends or market conditions.



Monthly Variations

There are significant variations in gross sales from month to month, with some months showing much higher sales than others. For example, November 2020 and November 2021 both had much higher sales than the preceding months, which may indicate successful promotions or marketing campaigns during that period.

Impact of Covid-19

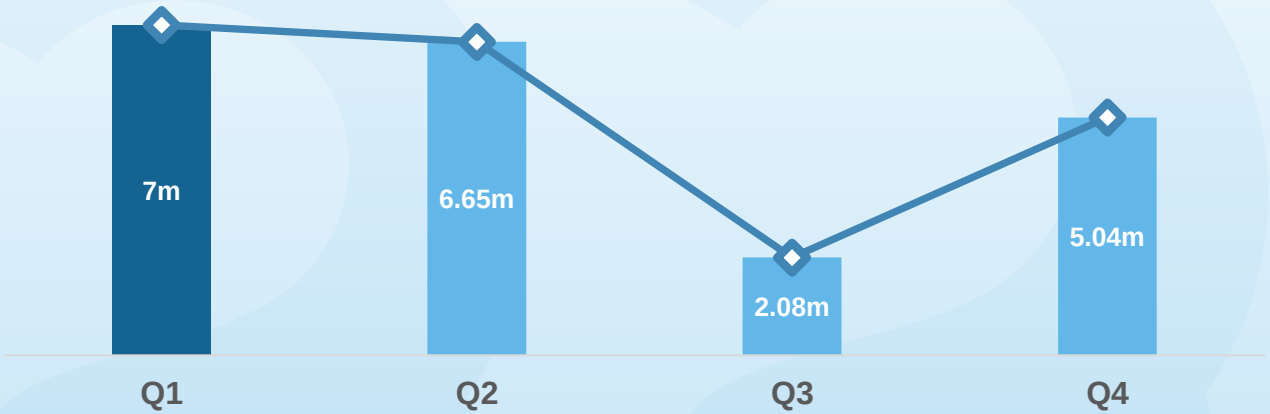
The data shows that Atliq Exclusive experienced a significant dip in sales during the months of March and April 2020, which may have been due to the COVID-19 pandemic and related lockdowns. However, the sales rebounded in the later months of 2020 and continued to grow in 2021.

Request 8

Maximum total sold quantity in 2020 quarter-wise.

Quarter	Total Sold Quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

Important Insights



The total sold quantity increased significantly in Q4 compared to Q3, indicating a strong finish to the year for Atliq Exclusive. This could be due to various reasons, including effective marketing and sales strategies or seasonal shopping trends.

The total sold quantity of Atliq Exclusive during 2020 is highest during Q1, followed by Q4, Q2, and Q3, respectively. This suggests that Q1 was the strongest quarter for Atliq Exclusive in terms of sales.

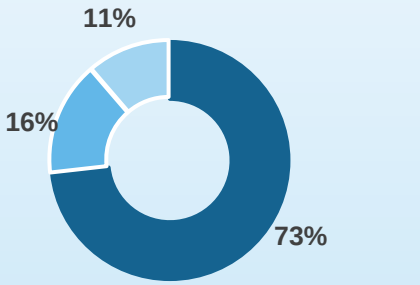
Overall, the data suggests that Atliq Exclusive's total sold quantity showed seasonal trends, with higher sales during Q1 and Q4, and lower sales during Q2 and Q3. The comparison of sales across quarters shows that Q1 was the strongest quarter for Atliq Exclusive in terms of sales.

Request 9

Channel that contributes to most gross sales in million in 2021 and their percentage of contribution.

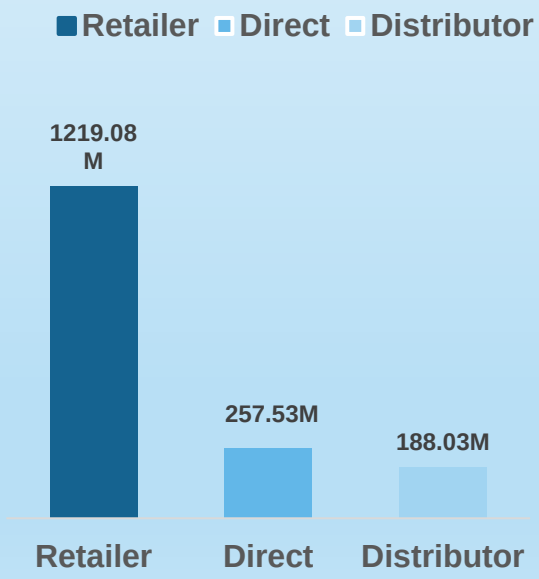
channel	Gross sales Mn	percentage
Retailer	1219.08M	73.23%
Direct	257.53M	15.47%
Distributor	188.03M	11.30%

Important Insights



The Retailer channel has contributed the highest gross sales of 1219.08 million, which is 73.22% of the total sales. This indicates that the Retailer channel is the most significant contributor to the sales of Atliq Exclusive in 2021.

1.The Direct and Distributor channels together have contributed around 26.79% of the total sales, which shows that Atliq Exclusive also depends on these channels to a significant extent.



1.The Retailer channel has contributed more than two-thirds of the total sales, which indicates that Atliq Exclusive heavily relies on this channel for its sales.

Overall, the data suggest that the Retailer channel is the most important channel for Atliq Exclusive in terms of gross sales contribution, and the company needs to maintain a strong relationship with this channel to continue its growth and success.

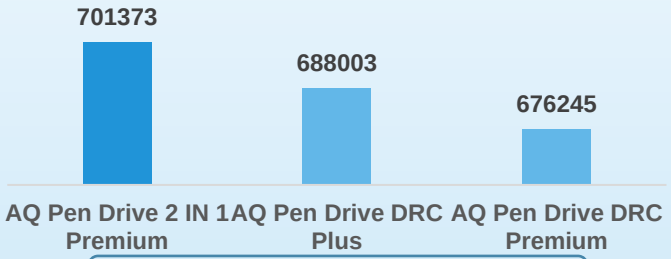
Request 10

Top 3 products in each division that has the highest told sold quantity in the fiscal year 2021.

division	Product code	product	Total sold quantity	Rank order
N & S	A6720160103	AQ Pen Drive 2 IN 1 Premium	701373	1
N & S	A6818160202	AQ Pen Drive DRC Plus	688003	2
N & S	A6819160203	AQ Pen Drive DRC Premium	676245	3
P & A	A2319150302	AQ Gamers Ms Standard 2	428498	1
P & A	A2520150501	AQ Maxima Ms Standard 1	419865	2
P & A	A2520150504	AQ Maxima Ms Plus 2	419471	3
PC	A4218110202	AQ Digit Standard Blue	17434	1
PC	A4319110306	AQ Velocity Velocity Plus Red	17280	2
PC	A4218110208	AQ Digit Digit Premium Misty Green	17275	3

Important Insights

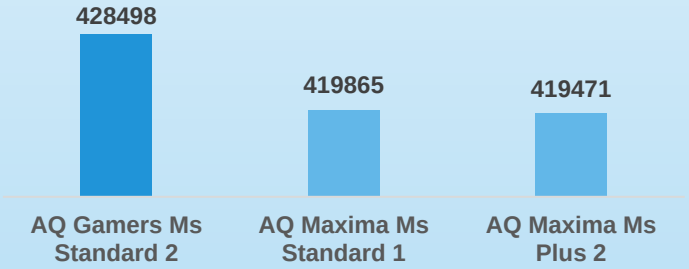
Based on the provided data, it is clear that “Atliq Exclusive” products have been well-received in the market with high total sold quantities across all divisions.



Network & Storage (N&S)

Finally, in the PC division, the AQ Digit Standard Blue, AQ Velocity Velocity Plus Red, and AQ Digit Digit Premium Misty Green were the top 3 products with the highest total sold quantity, suggesting a strong demand for high-quality computer accessories in this division.

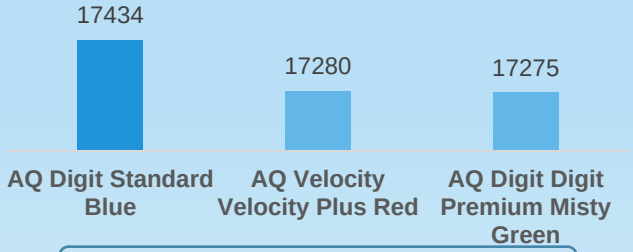
In the N & S division, the AQ Pen Drive 2 IN 1 Premium, AQ Pen Drive DRC Plus, and AQ Pen Drive DRC Premium were the top 3 products with the highest total sold quantity. Indicating a strong demand for premium-quality pen drives in this division.



Peripherals and Accessories (P&A)

Overall, these insights provide us with valuable information about “Atliq Exclusive” customers' preferences and the products that are performing well in the market.

Similarly, in the P & A division, the AQ Gamers Ms Standard 2, AQ Maxima Ms Standard 1, and AQ Maxima Ms Plus 2 were the top 3 products with the highest total sold quantity, highlighting the popularity of gaming mouse in this division.



Personal Computer (P&C)

It is crucial for “Atliq Exclusive” to continue to invest in these products and further improve their quality to meet the evolving needs of our customers.

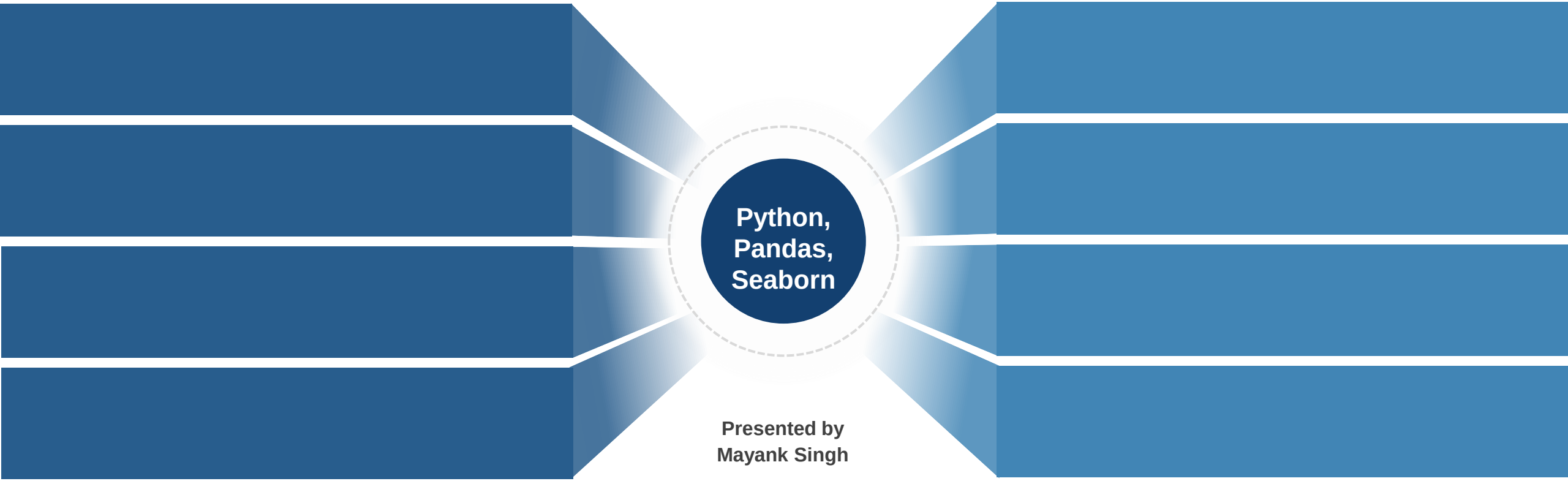
KODE BASICS

Dhaval Patel

Hemanand Vadivel

THANK YOU

Atliq Exclusive Consumer Goods



Presented by
Mayank Singh

Ad-hoc request 1

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

In [5]:
```
engine = create_engine('mysql://root:123456789@localhost:3306/gdb023')

# Establish a connection to the database
connection = engine.connect()

query = text("""
    SELECT DISTINCT market
    FROM dim_customer
    WHERE customer = 'Atliq Exclusive'
    AND region = 'APAC'
""")

# Use the connection object to execute the query
df_market = pd.read_sql_query(query, connection)



print(df_market)
```

```
        market
0        India
1    Indonesia
2        Japan
3  Philiphines
4  South Korea
5    Australia
6   Newzealand
7   Bangladesh
```

In [6]:
```
import geopandas as gpd

# Read the shapefile containing the geographical data
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Merge the markets data with the geographical data
merged = world.merge(df_market, how='inner', left_on='name', right_on='market')

# Plot the map
fig, ax = plt.subplots(figsize=(10, 6))
merged.plot(column='market', cmap='Paired', ax=ax)
ax.set_title('Markets in APAC for Atliq Exclusive')
ax.set_axis_off()
plt.show()
```

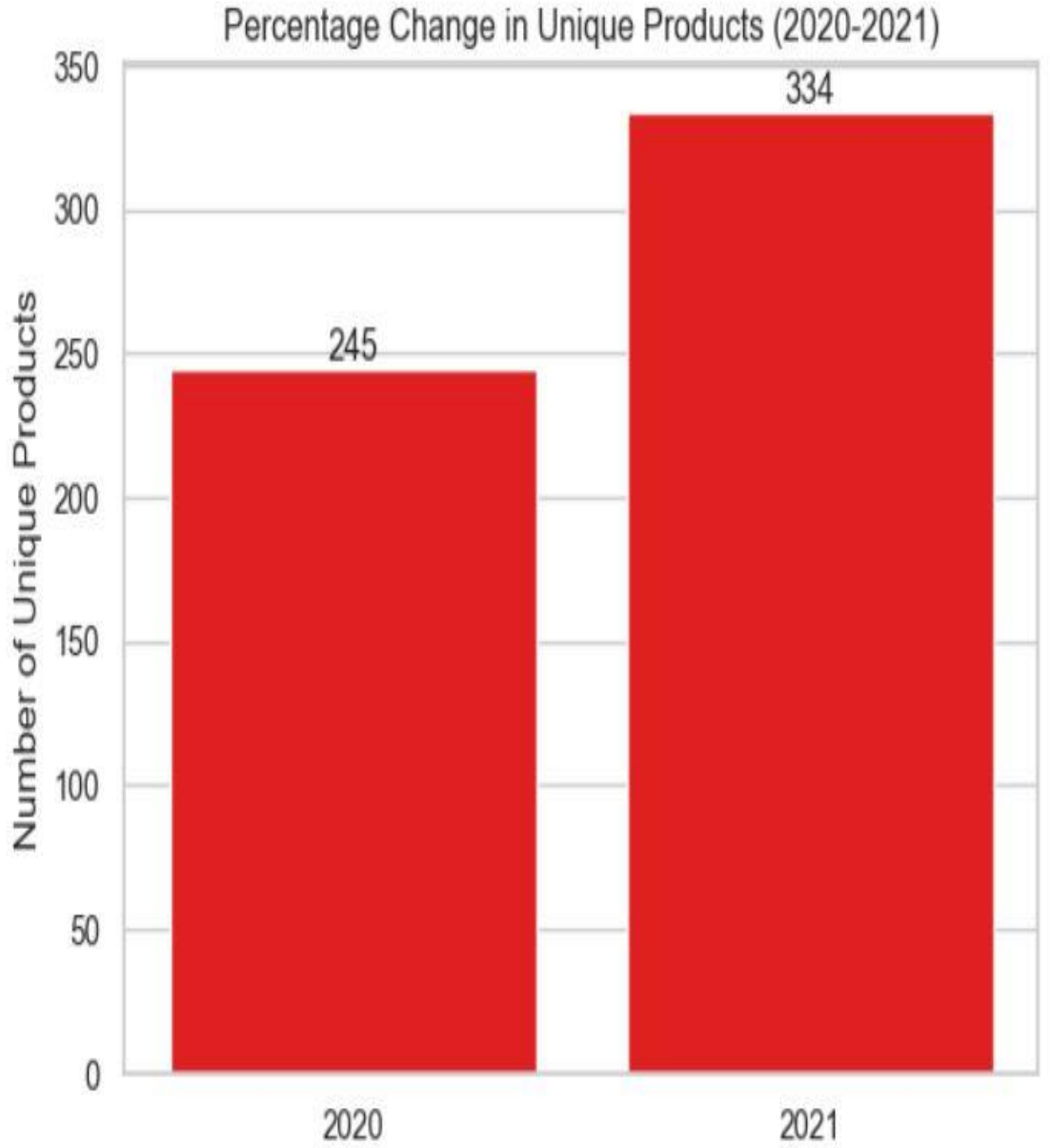
Ad - hoc request 2

percentage of unique product increase in 2021 vs. 2020

In [6]:
```
query = text("""
WITH cte AS ( SELECT
COUNT(DISTINCT CASE WHEN fiscal_year = 2020 THEN product_code END) AS unique_products_2020,
COUNT(DISTINCT CASE WHEN fiscal_year = 2021 THEN product_code END) AS unique_products_2021
FROM fact_sales_monthly )
SELECT unique_products_2020, unique_products_2021,
ROUND((unique_products_2021 - unique_products_2020)/unique_products_2020 * 100,2) AS percentage_chg
FROM cte;
""")

# Use the connection object to execute the query
df_unique_product = pd.read_sql_query(query, connection)


print(df_unique_product)
```

	unique_products_2020	unique_products_2021	percentage_chg
0	245	334	36.33

In [8]:
```
sns.set(style="whitegrid")

# creating a barplot
ax = sns.barplot(x=["2020", "2021"], y=[245, 334], color="red",
                 data=df_unique_product,
                 errwidth=0)

# adding data label
ax.bar_label(ax.containers[0])

# setting the y-axis label
plt.ylabel("Number of Unique Products")

# setting the title
plt.title("Percentage Change in Unique Products (2020-2021)")

# display the plot
plt.show()
```



Ad - hoc request 3

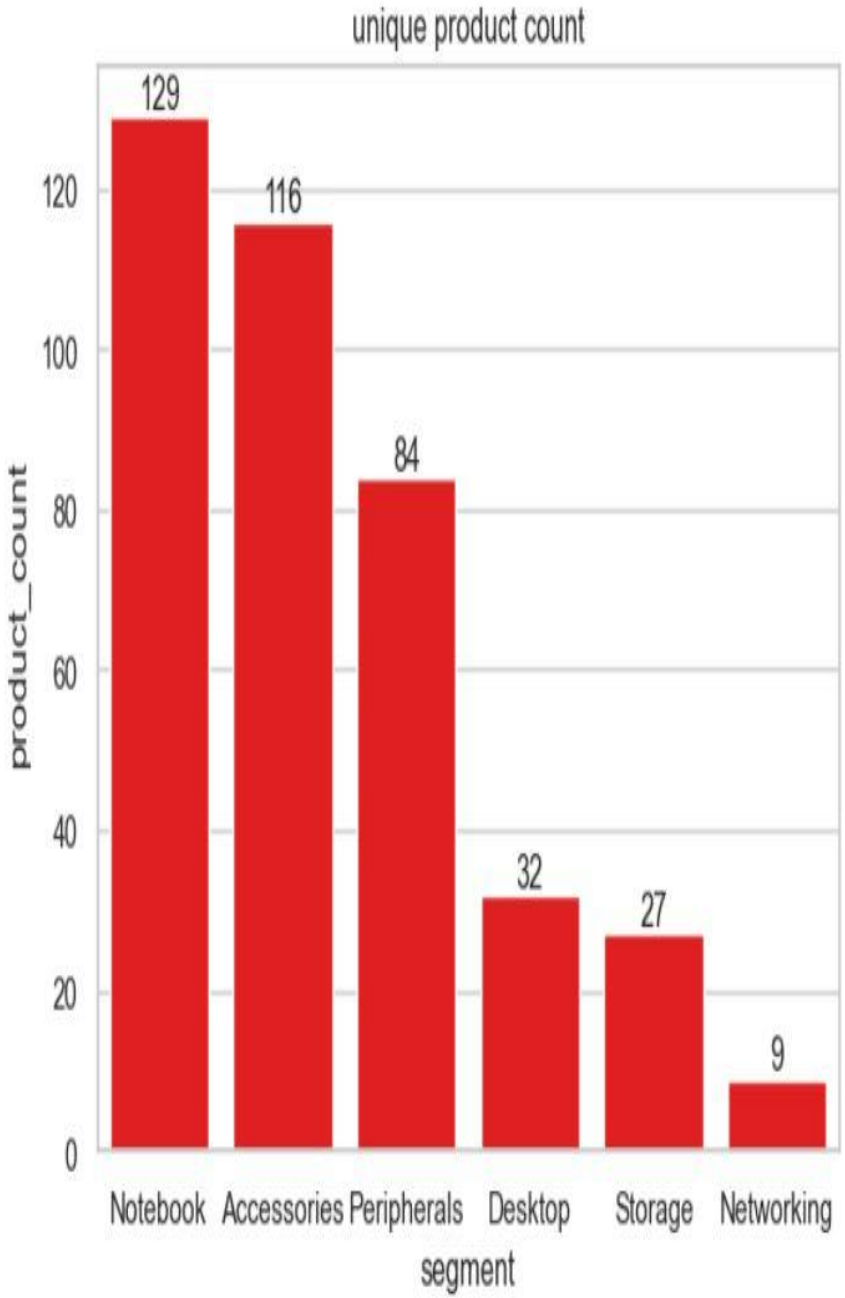
all the unique product counts for each segment and sort them in descending order of product counts

In [9]:
```
query = text("""
SELECT segment, COUNT(DISTINCT product_code) AS product_count
FROM dim_product
GROUP BY segment
ORDER BY product_count DESC;
""")

# Use the connection objecct to execute the query
df_up_count = pd.read_sql_query(query, connection)

df_up_count
```

Out[9]:

	segment	product_count
0	Notebook	129
1	Accessories	116
2	Peripherals	84
3	Desktop	32
4	Storage	27
5	Networking	9

In [10]:
```
sns.set(style="whitegrid")

# creating a barplot
ax = sns.barplot(x='segment', y='product_count', color='red',
                 data=df_up_count,
                 errwidth=0)

# adding data label
ax.bar_label(ax.containers[0])

# setting the title
plt.title("unique product count")

# display the plot
plt.show()
```



Ad- hoc request 4

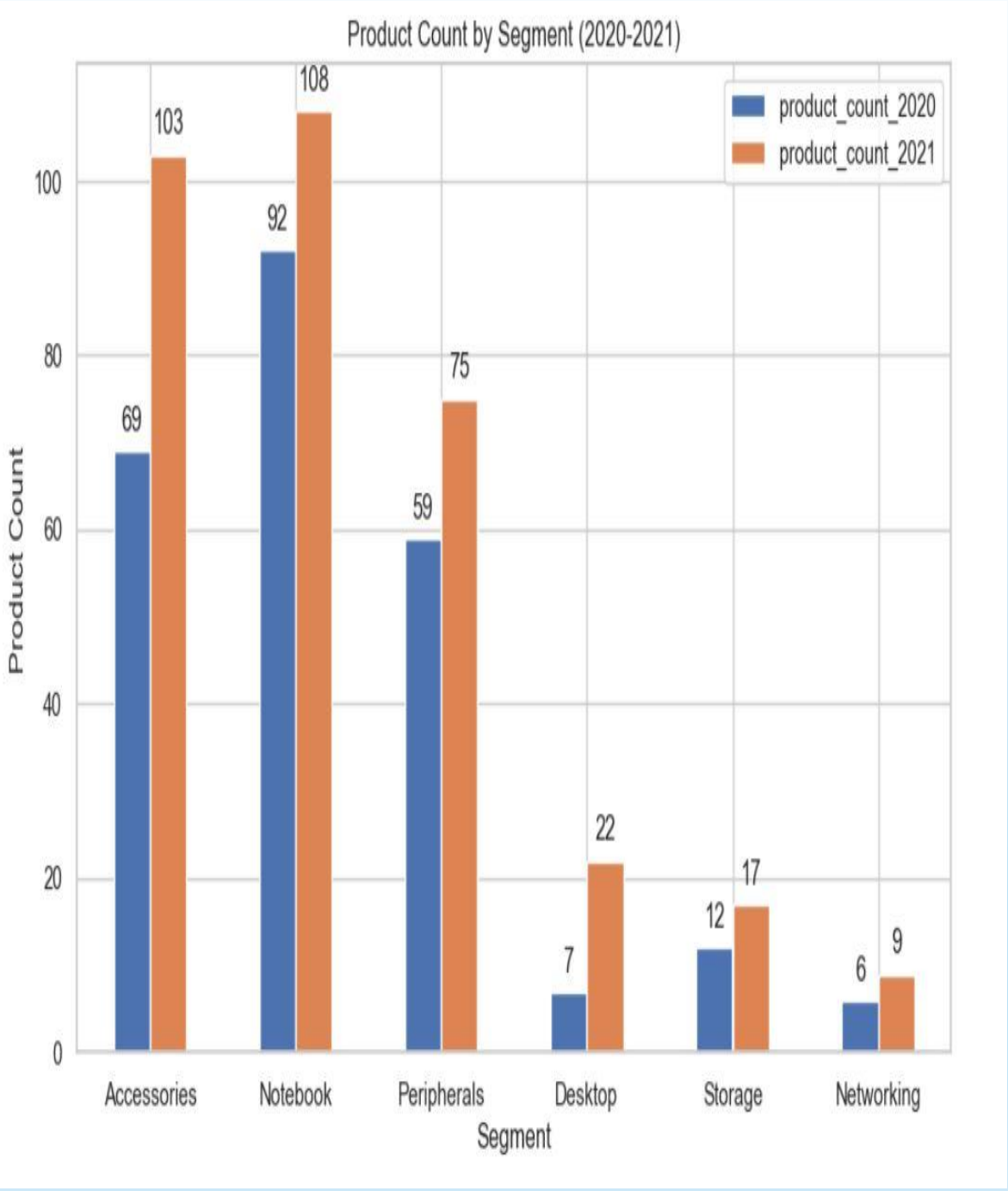
Follow-up: Which segment had the most increase in unique products in 2021 vs 2020?

In [11]:
```
query = text("""
WITH cte AS ( SELECT b.segment,
COUNT(DISTINCT CASE WHEN a.fiscal_year = 2020 THEN a.product_code END) AS product_count_2020,
COUNT(DISTINCT CASE WHEN a.fiscal_year = 2021 THEN a.product_code END) AS product_count_2021
FROM fact_sales_monthly a
INNER JOIN dim_product b
ON a.product_code = b.product_code
GROUP BY b.segment )
SELECT segment, product_count_2020, product_count_2021,
(product_count_2021 - product_count_2020) AS difference
FROM cte
ORDER BY difference DESC;
""")

# Use the connection objecct to execute the query
df_up_increase = pd.read_sql_query(query, connection)

df_up_increase
```

Out[11]:

	segment	product_count_2020	product_count_2021	difference
0	Accessories	69	103	34
1	Notebook	92	108	16
2	Peripherals	59	75	16
3	Desktop	7	22	15
4	Storage	12	17	5
5	Networking	6	9	3

In [13]:
```
sns.set(style="whitegrid")
ax = df_up_increase[['product_count_2020', 'product_count_2021']].plot(kind='bar', figsize=(10, 6))
ax.set_xlabel('Segment')
ax.set_ylabel('Product Count')
ax.set_title('Product Count by Segment (2020-2021)')

# replace x-axis labels with segment names
ax.set_xticklabels(df_up_increase['segment'], rotation=0)

# add the data labels
for i, p in enumerate(ax.patches):
    ax.annotate(str(p.get_height()),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')

# display the plot
plt.show()
```



Ad - hoc request 6

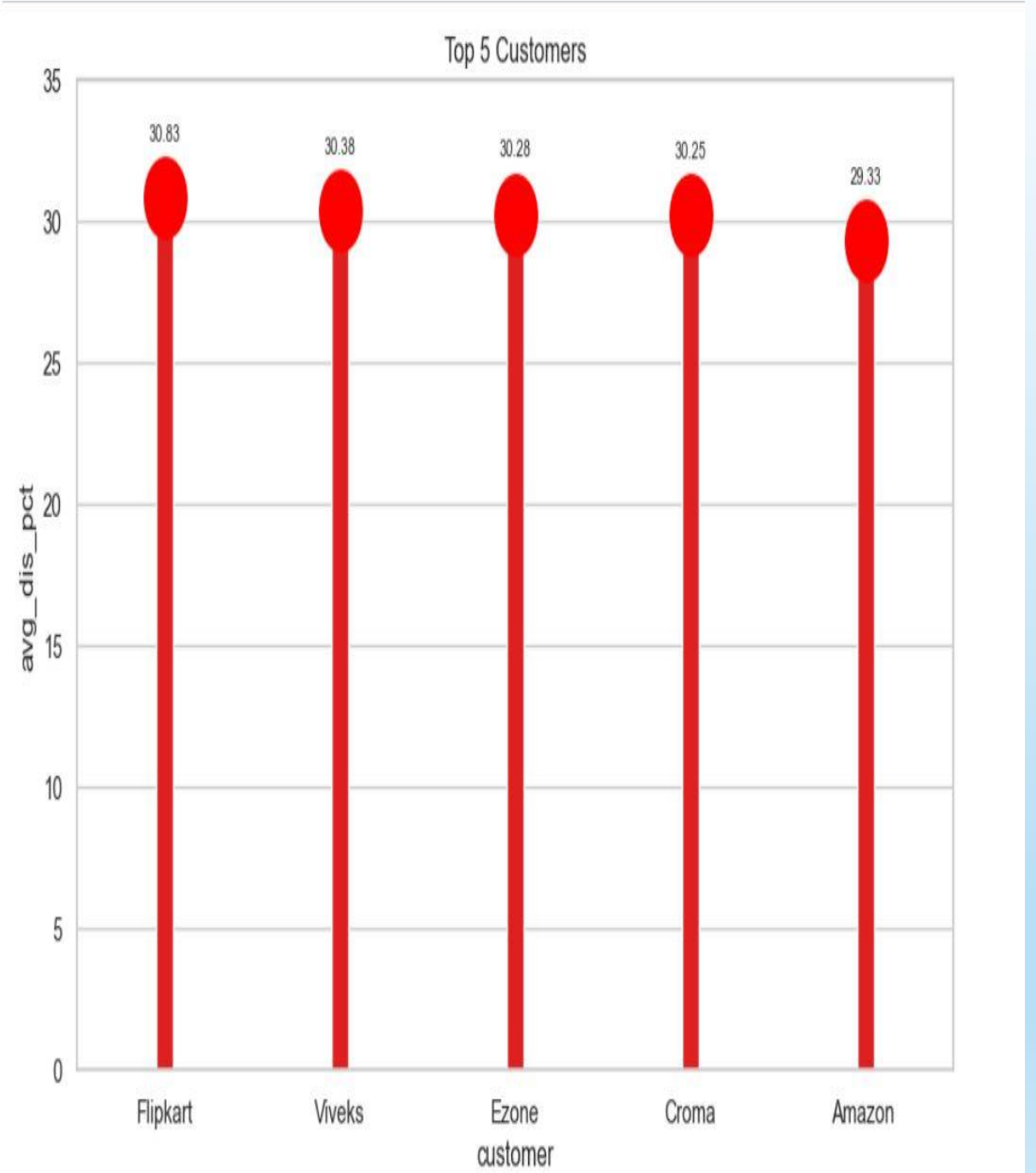
top 5 customers who received an average high pre_invoice_discount_pct for the fiscal year 2021 and in the Indian market.

In [16]:
```
query = text("""
SELECT a.customer_code, a.customer,
ROUND(AVG(b.pre_invoice_discount_pct)*100, 2) AS average_discount_percentage
FROM dim_customer a
JOIN fact_pre_invoice_deductions b
ON a.customer_code = b.customer_code
WHERE a.market = 'India'
AND fiscal_year = 2021
GROUP BY a.customer_code, a.customer
ORDER BY average_discount_percentage DESC
LIMIT 5;
""")

# Use the connection object to execute the query
df_preinvoice_discount = pd.read_sql_query(query, connection)

df_preinvoice_discount
```

Out[16]:

	customer_code	customer	average_discount_percentage
0	90002009	Flipkart	30.83
1	90002006	Viveks	30.38
2	90002003	Ezone	30.28
3	90002002	Croma	30.25
4	90002016	Amazon	29.33

In [17]:
```
# Set the figure size
plt.figure(figsize=(10, 6))

# Creating a lollipop chart using stripplot() and barplot()
sns.stripplot(x="customer", y="average_discount_percentage", data=df_preinvoice_discount, jitter=False, size=28, color='red')
sns.barplot(x='customer', y='average_discount_percentage', data=df_preinvoice_discount, width=0.1, color='red')

# Adding title and axis labels
plt.title("Top 5 Customers")
plt.xlabel("customer")
plt.ylabel("avg_dis_pct")

# Setting the lower limit of the y-axis to 0 and upper limit to 35
plt.ylim(0, 35)

# Add data labels
for i, point in enumerate(df_preinvoice_discount['average_discount_percentage']):
    plt.text(i, point+2, str(round(point, 2)), ha='center', fontsize=8)

# Show the plot
plt.show()
```



Ad - hoc request 7

the Gross sales amount for the customer “Atliq Exclusive” for each month.

In [18]:

```
query = text("""
SELECT MONTH(date) AS months, a.fiscal_year,
SUM(gross_price * sold_quantity) AS gross_sales
FROM fact_sales_monthly a
JOIN dim_customer b USING(customer_code)
JOIN fact_gross_price c USING(product_code)
WHERE customer = 'Atliq exclusive'
GROUP BY months, fiscal_year
ORDER BY a.fiscal_year, months;
""")

# Use the connection object to execute the query
df_GrossSales_Amount = pd.read_sql_query(query, connection)

# View the resulting DataFrame
df_GrossSales_Amount
```

In [20]:

```
# Create a dataframe from the table
df = pd.DataFrame({
    'months': [1,2,3,4,5,6,7,8,9,10,11,12,1,2,3,4,5,6,7,8,9,10,11,12],
    'fiscal_year': [2020]*12 + [2021]*12,
    'gross_sales': [9584951.94,8083995.55,766976.45,800071.95,1586964.48,3429736.57,
                    5151815.4,5638281.83,9092670.34,10378637.6,15231894.97,9755795.06,
                    19570701.71,15986603.89,19149624.92,11483530.3,19204309.41,15457579.66,
                    19044968.82,11324548.34,19530271.3,21016218.21,32247289.79,20409063.18]
})

# Pivot the table to create separate columns for 2020 and 2021
df_pivot = df.pivot_table(index='months', columns='fiscal_year', values='gross_sales')

# Reset the index and rename the columns
df_pivot = df_pivot.reset_index().rename(columns={2020: '2020 Gross Sales', 2021: '2021 Gross Sales'})

# Convert gross_sales to millions with 2 decimal places
df_pivot['2020 Gross Sales'] = df_pivot['2020 Gross Sales'] / 1000000
df_pivot['2021 Gross Sales'] = df_pivot['2021 Gross Sales'] / 1000000

# Plot a line chart
sns.set_style("darkgrid")
sns.lineplot(x='months', y='2020 Gross Sales', data=df_pivot, label='2020')
sns.lineplot(x='months', y='2021 Gross Sales', data=df_pivot, label='2021')

# Add data labels
for i in range(len(df_pivot)):
    plt.annotate(f"{df_pivot['2020 Gross Sales'][i]:,.2f}", (df_pivot['months'][i], df_pivot['2020 Gross Sales'][i]),
                 textcoords="offset points", xytext=(0,10), ha='center')
    plt.annotate(f"{df_pivot['2021 Gross Sales'][i]:,.2f}", (df_pivot['months'][i], df_pivot['2021 Gross Sales'][i]),
                 textcoords="offset points", xytext=(0,10), ha='center')

plt.title('Gross Sales by Month and Year')
plt.xlabel('Month')
plt.ylabel('Gross Sales (Millions)')
plt.legend()
plt.show()
```

Ad - hoc request 8

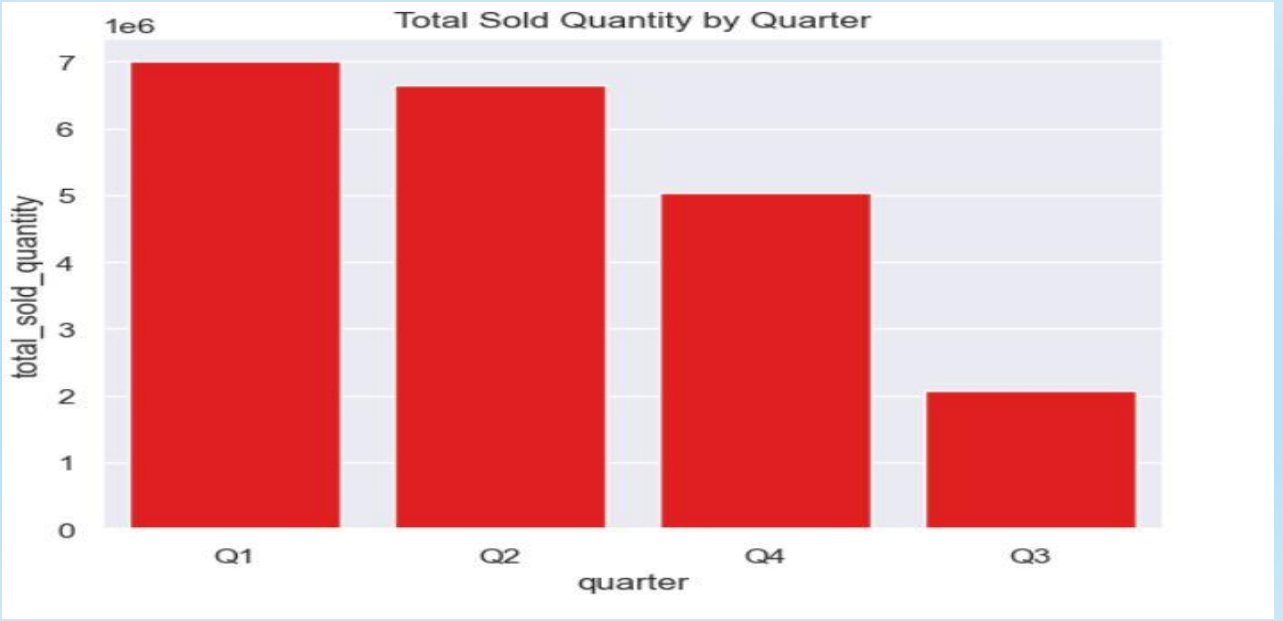
In which quarter of 2020, got the maximum total_sold_quantity?

In [21]:
```
query = text("""
WITH quarterly_sales AS (
SELECT
CASE
WHEN MONTH(date) IN (9,10,11) THEN 'Q1'
WHEN MONTH(date) IN (12,1,2) THEN 'Q2'
WHEN MONTH(date) IN (3,4,5) THEN 'Q3'
WHEN MONTH(date) IN (6,7,8) THEN 'Q4'
END AS quarter,
SUM(sold_quantity) AS total_sold_quantity
FROM fact_sales_monthly
WHERE fiscal_year = 2020
GROUP BY quarter
)
SELECT quarter, total_sold_quantity
FROM quarterly_sales
ORDER BY total_sold_quantity DESC;
-- LIMIT 1;
""")

# Execute the query and store the results in a DataFrame
df_totalsold_quantity = pd.read_sql_query(query, connection)

df_totalsold_quantity
```

Out[21]:

	quarter	total_sold_quantity
0	Q1	7,005,619.00
1	Q2	6,649,642.00
2	Q4	5,042,541.00

In [25]:
```
sns.set(style="darkgrid")

# creating a barplot
ax = sns.barplot(x='quarter', y='total_sold_quantity', color="red",
                 data=df_totalsold_quantity,
                 errwidth=0)

# setting the y-axis label
plt.ylabel("total_sold_quantity")
plt.xlabel("quarter")

# setting the title
plt.title("Total Sold Quantity by Quarter")

# display the plot
plt.show()
```



Ad - hoc request 9

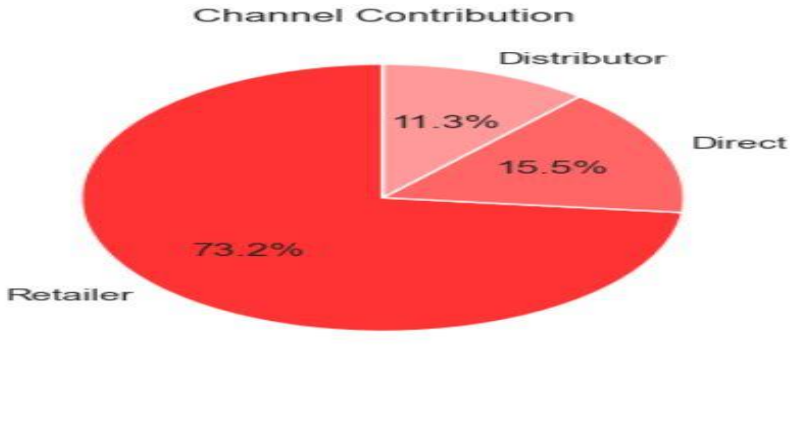
Channel that contributes to most gross sales in Million in 2021 and their percentage of contribution

In [26]:
```
query = text("""
WITH gross_sales_cte AS (
SELECT
c.channel,
ROUND(SUM(b.gross_price * a.sold_quantity) / 1000000, 2) AS gross_sales_mln
FROM fact_sales_monthly a
JOIN fact_gross_price b ON a.product_code = b.product_code
JOIN dim_customer c ON a.customer_code = c.customer_code
AND a.fiscal_year = b.fiscal_year
WHERE a.fiscal_year = 2021
GROUP BY c.channel
)
SELECT
channel,
gross_sales_mln,
ROUND(gross_sales_mln / SUM(gross_sales_mln) OVER() * 100, 2) AS percentage
FROM gross_sales_cte
ORDER BY gross_sales_mln DESC;
""")

# Execute the query and store the results in a DataFrame
df_channel_contribution = pd.read_sql_query(query, connection)

df_channel_contribution
```

Out[26]:

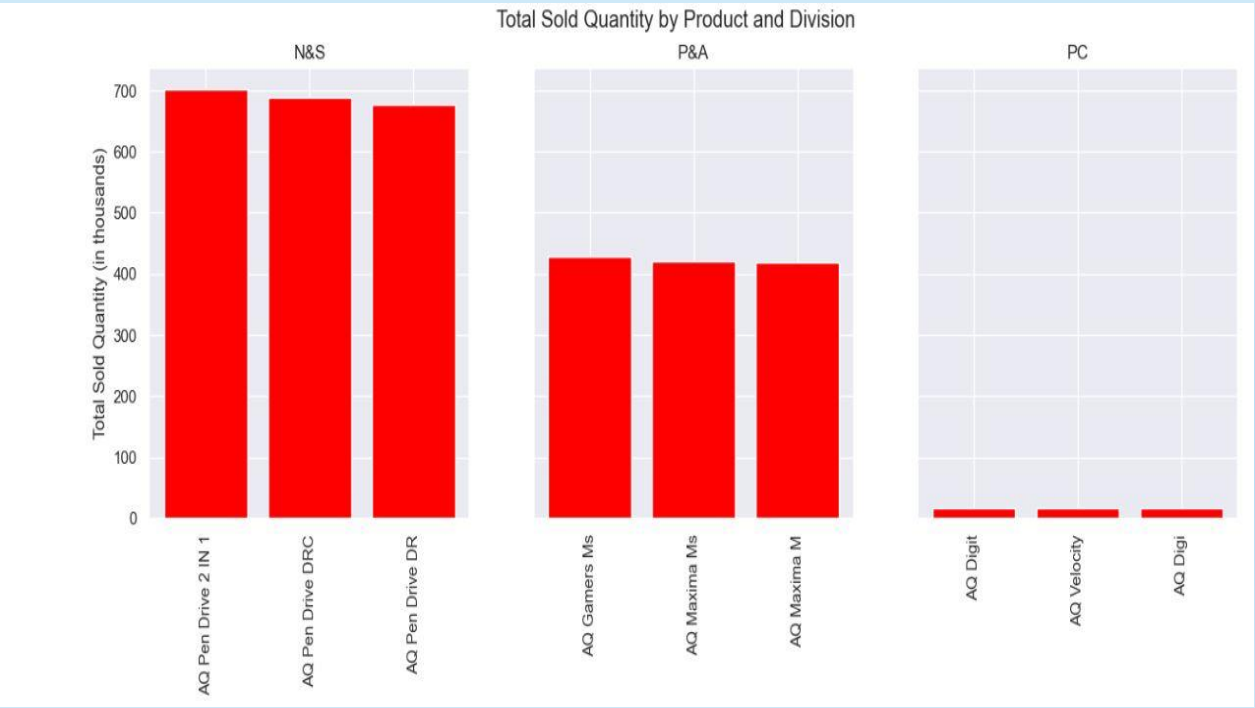
	channel	gross_sales_mln	percentage
0	Retailer	1,219.08	73.23
1	Direct	257.53	15.47
2	Distributor	188.03	11.30

In [27]:
```
sns.set(style="darkgrid")

# Set the figure size
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10, 4))

# creating a barplot
sns.barplot(x='channel', y='gross_sales_mln', color="red",
            data=df_channel_contribution, errwidth=0, ax=ax1)

# setting the title
ax1.set_title("Highest Sales by Channel")

# adding data label
ax1.bar_label(ax1.containers[0])

# Create a list of colors for the pie chart
colors = ['#ff3333', '#ff6666', '#ff9999']

# Create a pie chart
ax2.pie(df_channel_contribution['percentage'], labels=df_channel_contribution['channel'], colors=colors,
        autopct='%1.1f%%', startangle=90)

# Add a title to the chart
ax2.set_title('Channel Contribution')

# Show the chart
plt.show()
```



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

In [28]:
```
query = text("""
WITH cte AS (
    SELECT
        b.division,
        b.product_code,
        b.product,
        SUM(a.sold_quantity) AS total_sold_quantity,
        RANK() OVER (PARTITION BY b.division ORDER BY SUM(a.sold_quantity) DESC) AS rank_order
    FROM fact_sales_monthly a
    JOIN dim_product b ON a.product_code = b.product_code
    WHERE a.fiscal_year = 2021
    GROUP BY b.division, b.product_code, b.product
)
SELECT division, product_code, product, total_sold_quantity, rank_order
FROM cte
WHERE rank_order <= 3;
""")

# Execute the query and store the results in a DataFrame
df_division = pd.read_sql_query(query, connection)

df_division
```



In [29]:
```
# Create a DataFrame from the given data
df = pd.DataFrame({
    'division': ['N & S', 'N & S', 'N & S', 'P & A', 'P & A', 'P & A', 'PC', 'PC', 'PC'],
    'product': ['AQ Pen Drive 2 IN 1', 'AQ Pen Drive DRC', 'AQ Pen Drive DR', 'AQ Gamers Ms', 'AQ Maxima Ms',
                'AQ Maxima M', 'AQ Digit', 'AQ Velocity', 'AQ Digi'],
    'total_sold_quantity': [701373.0, 688003.0, 676245.0, 428498.0, 419865.0, 419471.0, 17434.0, 17280.0, 17275.0],
})

# Divide the data into three different data frames based on the 'division' column
n_s_df = df[df['division'] == 'N & S']
p_a_df = df[df['division'] == 'P & A']
pc_df = df[df['division'] == 'PC']
```

In [30]:
```
# Create a figure with 3 subplots
fig, axs = plt.subplots(1, 3, figsize=(15, 5), sharey=True)

# Plot a bar graph for the N&S data frame
axs[0].bar(n_s_df['product'], n_s_df['total_sold_quantity']/1000, color='red')
axs[0].set_title('N&S')
axs[0].set_ylabel('Total Sold Quantity (in thousands)')

# Plot a bar graph for the P&A data frame
axs[1].bar(p_a_df['product'], p_a_df['total_sold_quantity']/1000, color='red')
axs[1].set_title('P&A')

# Plot a bar graph for the PC data frame
axs[2].bar(pc_df['product'], pc_df['total_sold_quantity']/1000, color='red')
axs[2].set_title('PC')

# Set the x-axis labels for all subplots
for ax in axs:
    ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
# Add a title for the entire figure
fig.suptitle('Total Sold Quantity by Product and Division')
# Display the plot
plt.show()
```


THANK YOU