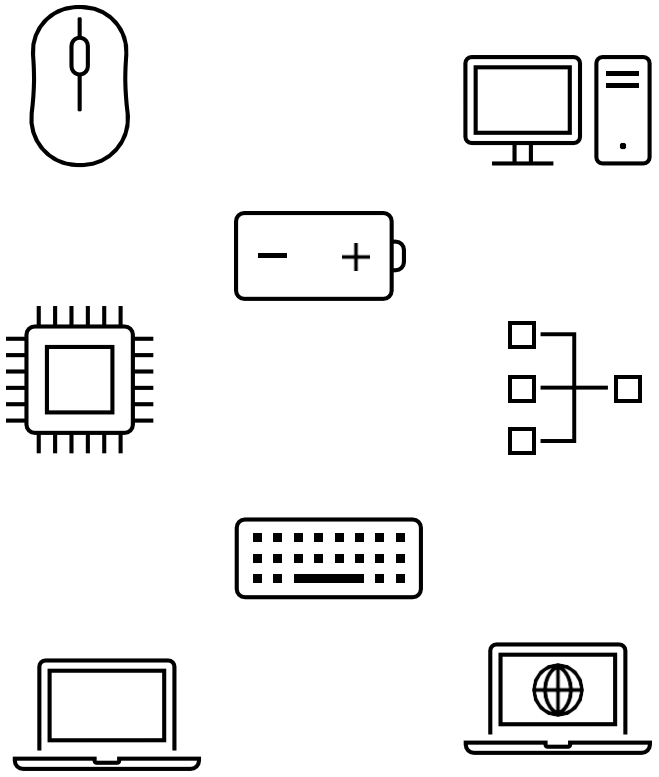
Consumer Goods
Queries & Insights

SQL, Power BI, Power Point



Aasdeep Singh

Why?

Problem Statement / Task

What?

About the company

How?

Data, request and tools

Why?

Problem Statement:

Atliq Hardwares (a fictional company) is a prominent computer hardware manufacturer with a global presence.

The need : Enhanced data insights to facilitate swift and informed decision-making.

Adding a data analytics team and hiring new analysts.

Seeking individuals proficient in both technical and soft skills.

To assess these competencies, initiated a SQL challenge as part of the hiring process, aiming to gain a comprehensive understanding of candidates' capabilities."

Task :

1. There are 10 ad hoc requests for which the business needs insights.
2. Write SQL query to answer these requests.

3. Visualize the output and give insights.
4. Create a presentation for the top-level management who are target audience.

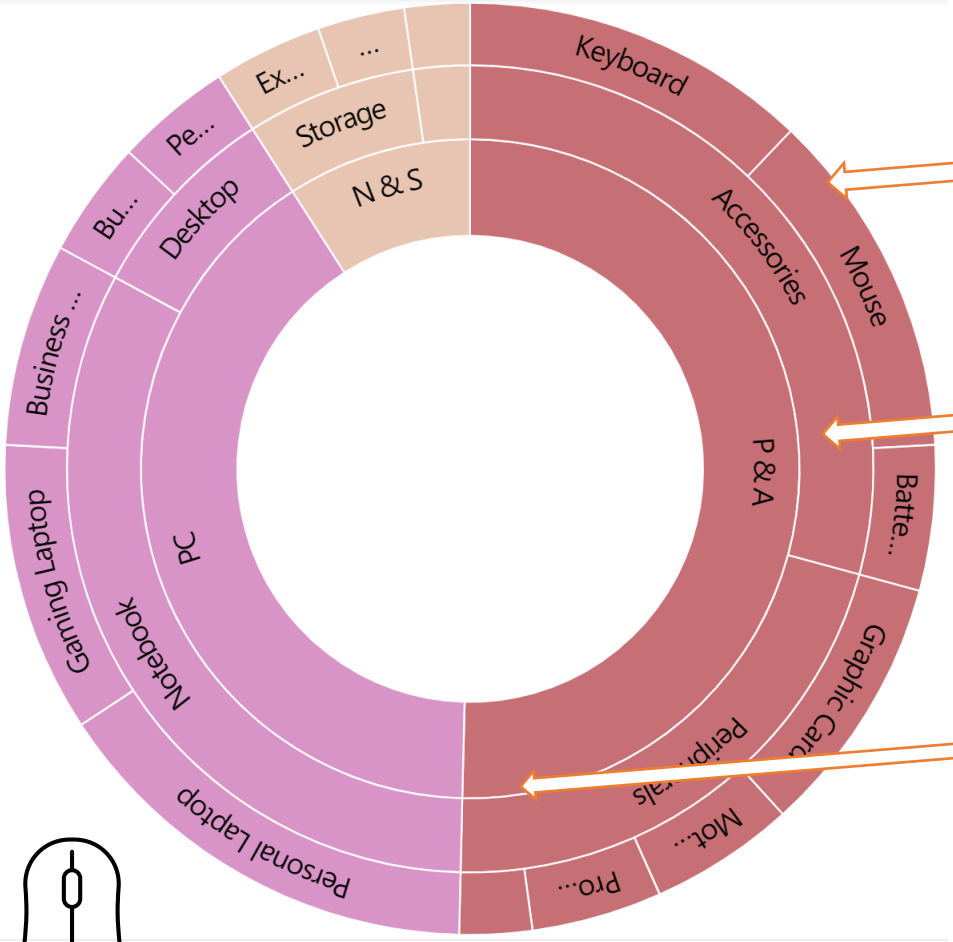
What?

Atliq Hardwares

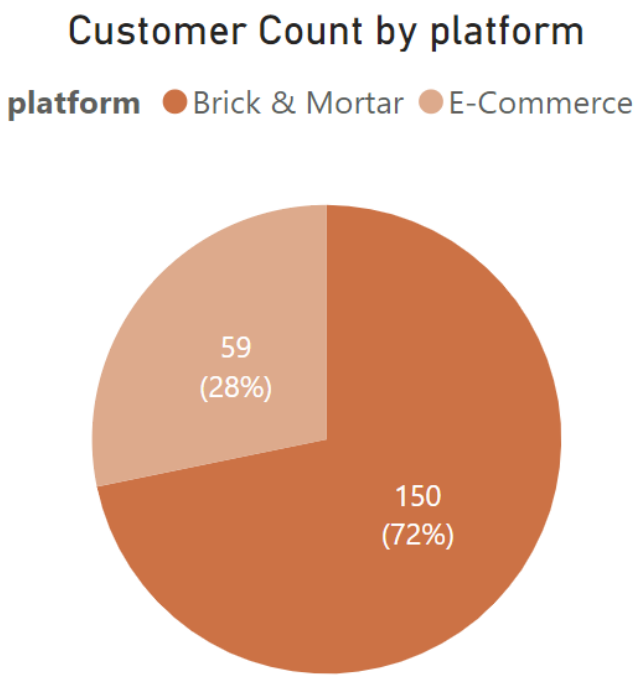
Products

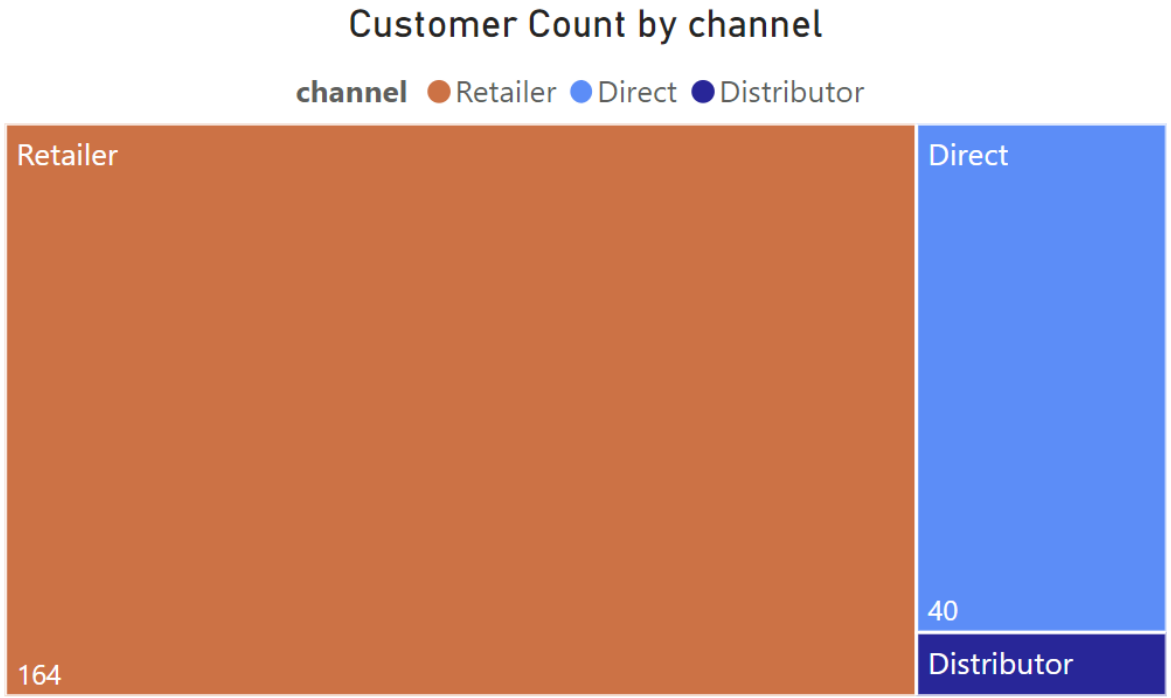
Fiscal Year

FY2020 = September 2019 - August 2020
FY2021 = September 2020 - August 2021



Customers


Customer Count by platform
platform
Brick & Mortar
E-Commerce
59
(28%)
150
(72%)


Customer Count by channel
channel
Retailer
Direct
Distributor
Retailer
164
Direct
40
Distributor

Region & Markets

Codebasics SQL Challenge

Requests:

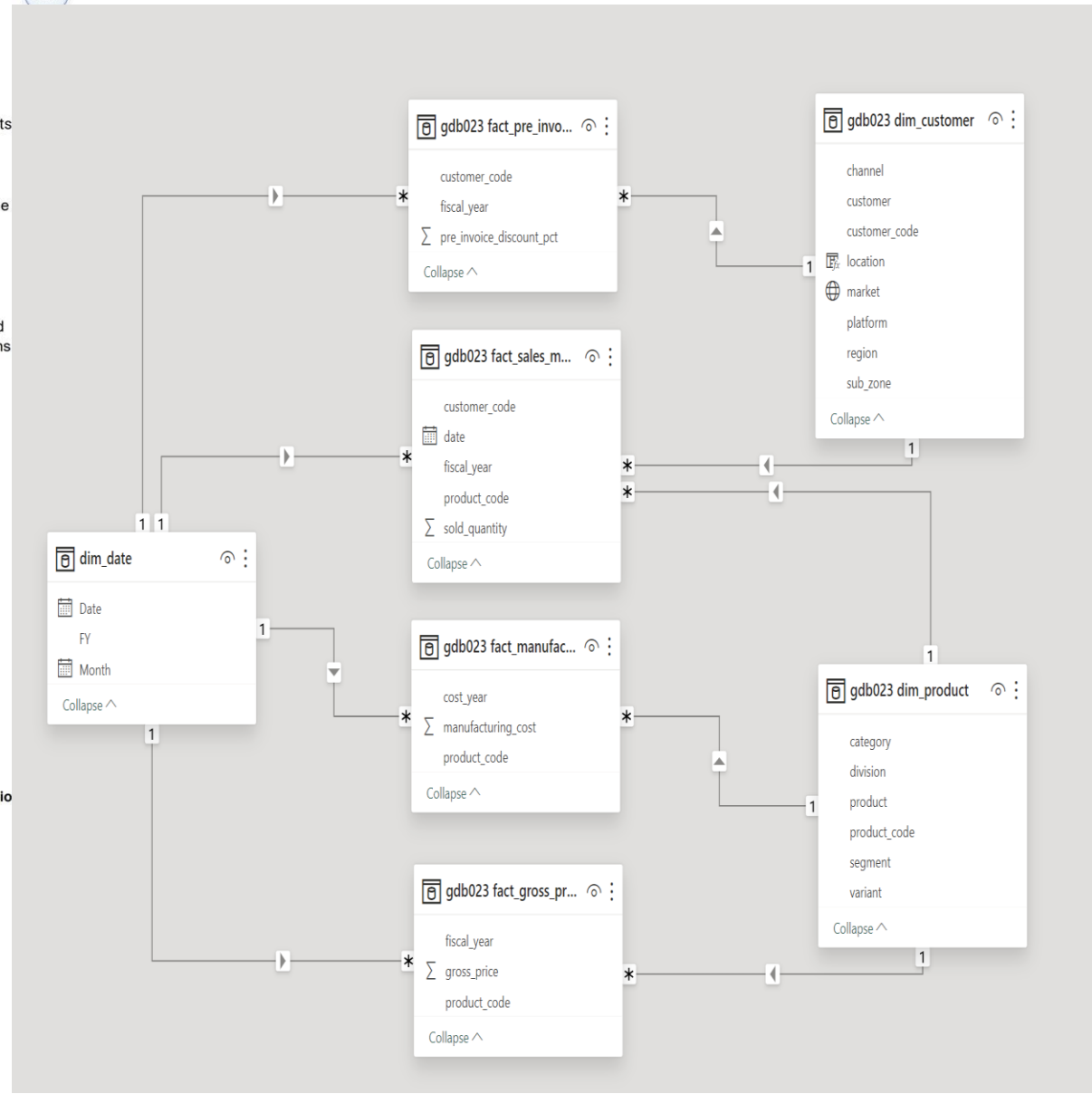
1. Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region.

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

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

4. Follow-up: 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

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



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

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

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

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

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

How?

Tools used

MYSQL

POWER BI

POWER POINT

YOUTUBE

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

```sql
SELECT
    DISTINCT(market)
FROM dim_customer
WHERE customer = "Atliq Exclusive"
        AND region = 'APAC'
ORDER BY market ASC;
```

market
Australia
Bangladesh
India
Indonesia
Japan
Newzealand
Philiphines
South Korea

Explanation:

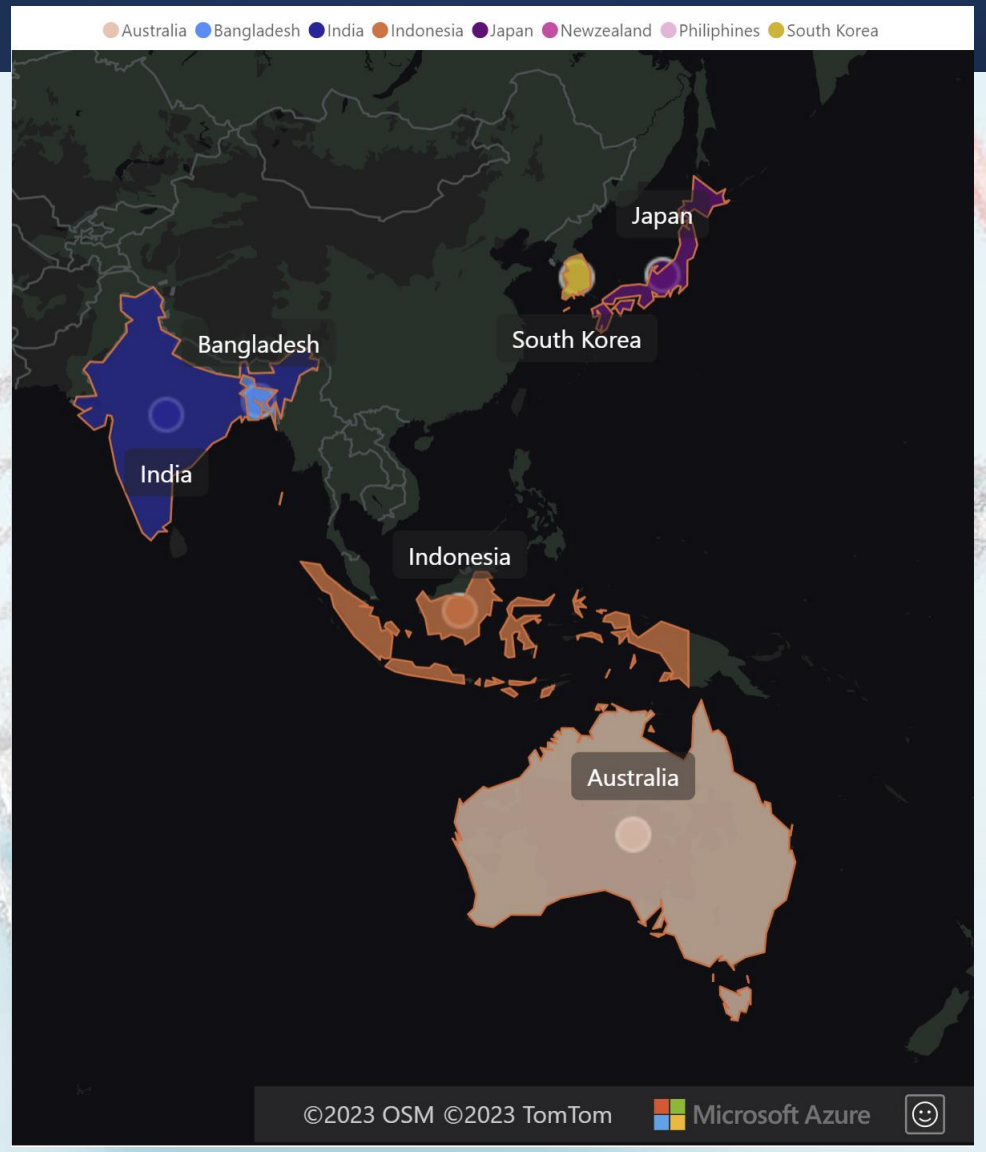
- We select all the **distinct market** from the customer table.
- We use where class to give the condition - customer = **"Atliq Exclusive"** and region = **APAC**.
- To display the list of countries in alphabetical order, use order market by ascending.

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

Insights:

Atliq Exclusive has established its market in **8 countries** in the APAC region with ‘India’ being its base market.

	market
▶	Australia
	Bangladesh
	India
	Indonesia
	Japan
	Newzealand
	Philiphines
	South Korea



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

```sql
WITH unique_products 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
    * ,
    CONCAT(ROUND((unique_products_2021-unique_products_2020)*100/unique_products_2020,2),"%") AS
percentage_chg
 FROM unique_products;
```

Explanation:

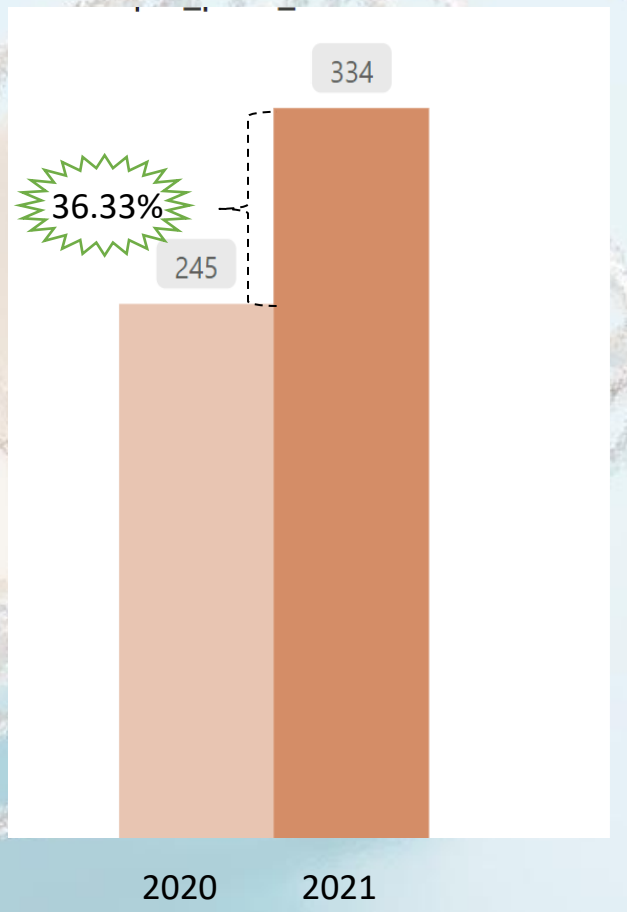
- From the sales table, we count the number of unique products for the fiscal year 2020 & 2021 separately using **Case statements**.
- With this result stored as a **CTE** named unique products, we find the percentage of unique products increase from 2020 to 2021.
- **Round** the results to two decimal and **concat** a percentage symbol for aesthetic reason.

	unique_products_2020	unique_products_2021	percentage_chg
▶	245	334	36.33%

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

	unique_products_2020	unique_products_2021	percentage_chg
▶	245	334	36.33%



Insights:

- The number of unique products in 2020 was 245 whereas in 2021 it has increased to 334.
- we note a 36.33% increase in the number of unique products.
- The company has a good growth rate and is capturing new markets.

Provide a report with all the unique product counts for each segment.

The final output contains 2 fields - segment & product_count. Sort them in descending order of product counts.

```sql
SELECT
    segment,
    COUNT(DISTINCT(product_code)) AS product_count
FROM dim_product
GROUP BY segment
ORDER BY product_count DESC;
```

segment	product_count
Notebook	129
Accessories	116
Peripherals	84
Desktop	32
Storage	27
Networking	9

Explanation:

- From product table, we choose the column segment and count the distinct number of product as product count.
- Since we use an aggregate function, we have to use **group by.**
- **T**he final result is **ordered by** the product count in a **desc**ending order

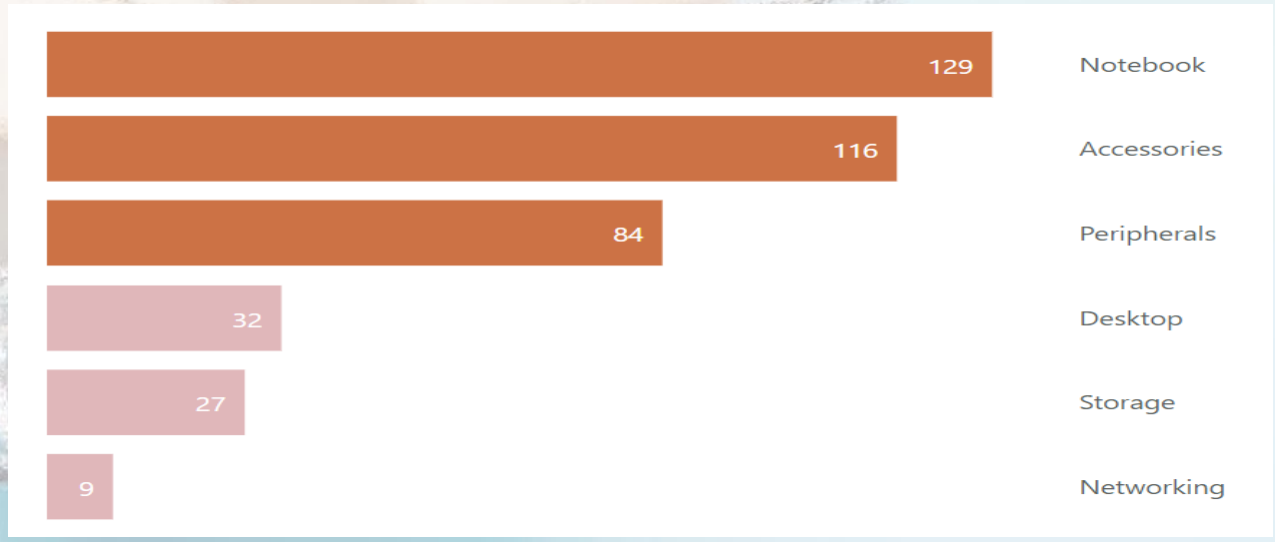
Provide a report with all the unique product counts for each segment.

The final output contains 2 fields - segment & product_count. Sort them in descending order of product counts.

segment	product_count
Notebook	129
Accessories	116
Peripherals	84
Desktop	32
Storage	27
Networking	9

Insights:

- **Notebook, Accessories and peripherals** have good and growing number of unique product.
- These 3 account **for 83% of all unique products**.
- Desktop, storage and networking needs to concentrate on increasing the number of products offered.



Follow-up: 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

```sql
WITH unique_products AS (
    SELECT
        p.segment,
        COUNT(DISTINCT CASE WHEN fiscal_year=2020 THEN s.product_code END) AS product_count_2020,
        COUNT(DISTINCT CASE WHEN fiscal_year=2021 THEN s.product_code END) AS product_count_2021
    FROM fact_sales_monthly s
    JOIN dim_product p
    USING (product_code)
    GROUP BY p.segment
    )
SELECT
    * ,
    (product_count_2021-product_count_2020) AS difference
FROM unique_products
ORDER BY difference DESC;
```

segment	product_count_2020	product_count_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

Explanation:

- We find the **unique product count** for each fiscal year like before using **CASE statements.**
- **Group by** segment to get the same for each segment.
- using the above as **a CTE**, find the difference in the product count and display the result.

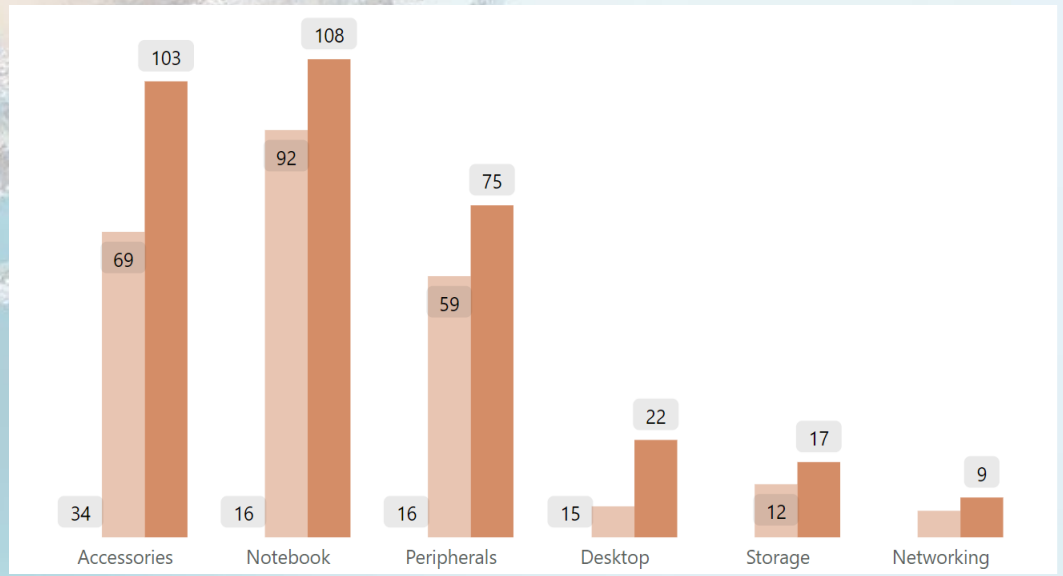
Follow-up: Which segment had the most increase in unique products in 2021 vs 2020?

The final output contains these fields:-- segment, product_count_2020, product_count_2021, difference

segment	product_count_2020	product_count_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

Insights:

- **Accessories is the fastest growing** segment with increase in unique product count of 34.
- Followed by notebook, peripherals and desktop withan increase of 16 each and 15 respectively.
- **Storage and networking are the least growing** segments in terms of increasing the unique product counts.



Get the products that have the highest and lowest manufacturing costs.

The final output should contain these fields: product_code, product, manufacturing_cost

```sql
SELECT
    p.product_code,
    p.product,
    CONCAT("$ ",ROUND(m.manufacturing_cost,2)) AS manufacturing_cost
FROM fact_manufacturing_cost m
JOIN dim_product p
USING (product_code)
WHERE m.manufacturing_cost = (select max(manufacturing_cost) from fact_manufacturing_cost)
    OR m.manufacturing_cost = (select min(manufacturing_cost) from fact_manufacturing_cost)
ORDER BY manufacturing_cost DESC;
```

$240.54

$ 0.89

	product_code	product	manufacturing_cost
▶	A6120110206	AQ HOME Allin1 Gen 2	$ 240.54
	A2118150101	AQ Master wired x1 Ms	$ 0.89

Explanation:

- We **join** the tables - manufacturing cost and products.
- Using separate **subqueries** to get the maximum and minimum manufacturing cost.
- This value is used in **where clause** to filter the result.

Generate a report which contains the top 5 customers who received an average high pre invoice discount pct for the fiscal year 2021 in the Indian market.

The final output contains these fields: customer_code, customer, average_discount_percentage

```sql
SELECT
    c.customer_code,
    c.customer,
    CONCAT(ROUND(AVG(pre_invoice_discount_pct*100),2)," %")
                            AS average_discount_percentage
FROM dim_customer c
JOIN fact_pre_invoice_deductions pre
USING (customer_code)
WHERE fiscal_year = 2021 AND market = "india"
GROUP BY c.customer_code,c.customer
ORDER BY AVG(pre_invoice_discount_pct) DESC
LIMIT 5;
```

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 %

Explanation:

- We **join** the required tables and give filter conditions in **where clause**.
- Since we need to calculate an **aggregate fn, use group by**.
- To get the **top 5**, order by descending and limit the results to 5 rows.

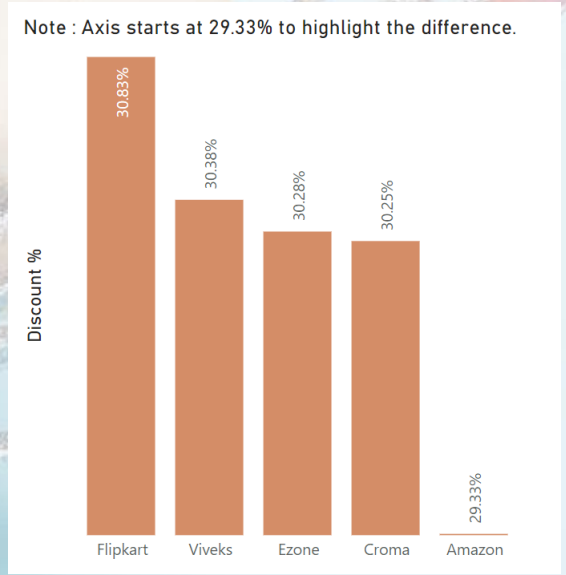
Generate a report which contains the top 5 customers who received an average high pre invoice discount pct for the fiscal year 2021 in the Indian market.

The final output contains these fields: customer_code, customer, average_discount_percentage

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 %

Insights :

- 5 customers get an approximately the same range of discount percentage.
- This is not ideal but just to study the difference even further I have chosen to start the axis of the bar chart at 29.33 percentage
- We can note that Flipkart has maximum discount percentage of 30.83.
- Viveks, Ezone and chroma have almost the same % of about 30.3
- Amazon has the lowest among the top five with 29.33% discount.



Get the complete report of Gross sales amount for “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

```sql
SELECT
    MONTHNAME(s.date) as month,
    YEAR(s.date) AS YEAR,
    CONCAT("$",ROUND(SUM(g.gross_price*s.sold_quantity)/1000000,2))
                                            AS Gross_sales_amount_mln
FROM fact_sales_monthly s
JOIN fact_gross_price g
    USING (product_code,fiscal_year)
JOIN dim_customer c
    USING (customer_code)
WHERE c.customer = "Atliq Exclusive"
GROUP BY YEAR,MONTH(DATE),MONTHNAME(date)
ORDER BY YEAR;
```

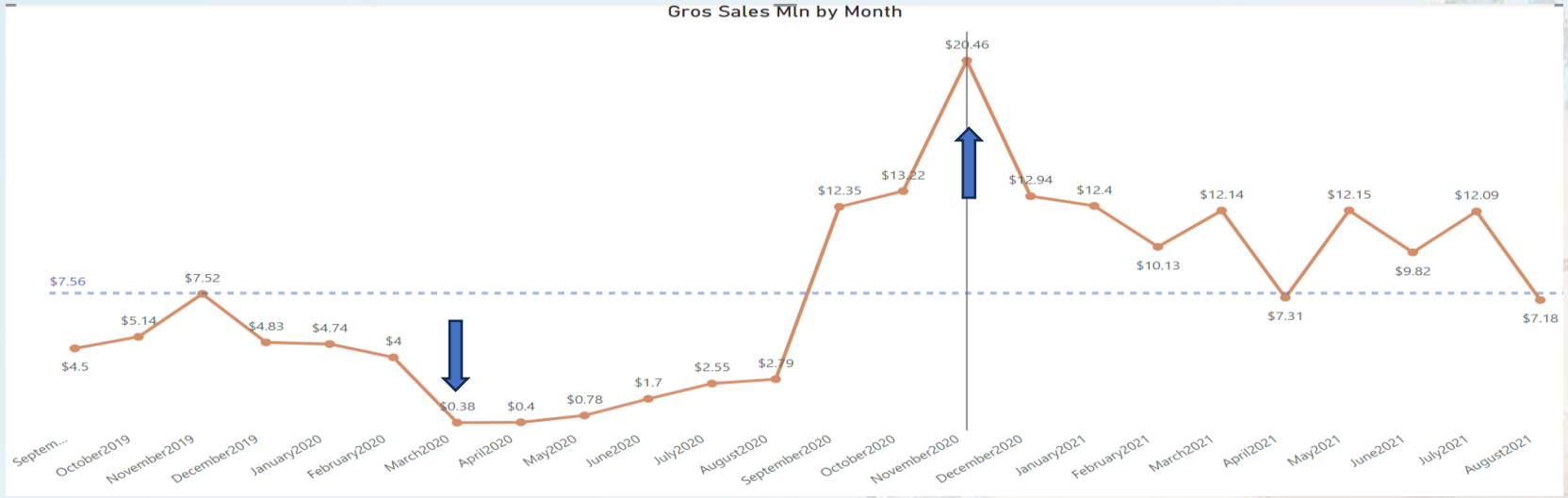
month	YEAR	Gross_sales_amount_mln
September	2019	$4.50
October	2019	$5.14
November	2019	$7.52
December	2019	$4.83
January	2020	$4.74
February	2020	$4.00
March	2020	$0.38
April	2020	$0.40
May	2020	$0.78
June	2020	$1.70
July	2020	$2.55
August	2020	$2.79
September	2020	$12.35
October	2020	$13.22
November	2020	$20.46
December	2020	$12.94
January	2021	$12.40
February	2021	$10.13
March	2021	$12.14
April	2021	$7.31
May	2021	$12.15
June	2021	$9.82
July	2021	$12.09
August	2021	$7.18

Explanation:

- **Gross sales amount** is obtained using gross price and sold quantity. To filter the data for the given customer, I am using customer also.
- We need to **perform multiple joins.**
- Since we are asked to generate report with data for each month, we use the time function, **month name to get month from the date**.
- I want to display the final report in the time sequence, so I have also taken the year from the date.

Get the complete report of Gross sales amount for “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



	month	YEAR	Gross_sales_amount_mln
▶	September	2019	$4.50
	October	2019	$5.14
	November	2019	$7.52
	December	2019	$4.83
	January	2020	$4.74
	February	2020	$4.00
	March	2020	$0.38
	April	2020	$0.40
	May	2020	$0.78
	June	2020	$1.70
	July	2020	$2.55
	August	2020	$2.79
	September	2020	$12.35
	October	2020	$13.22
	November	2020	$20.46
	December	2020	$12.94
	January	2021	$12.40
	February	2021	$10.13
	March	2021	$12.14
	April	2021	$7.31
	May	2021	$12.15
	June	2021	$9.82
	July	2021	$12.09
	August	2021	$7.18

Insights:

- The trend of gross sales amount over the months can be classified into two distinct parts:
  - **Sales data from fiscal year 2020 where it is below the average line**,
  - the part where it is mostly **above the average line for the fiscal year 2021**.
- The **minimum value is seen in March 2020** can be attributed to the COVID and it remains there because there was not much sales made during this period.
- From September, we can see a drastic increase in the gross sales, and we can see the **peak in November 2020. This can be attributed to the holiday season** right after COVID when people started purchasing. **The company has maintained their gross sales above the average line** after picking back up.

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.

```sql
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
ORDER BY total_sold_quantity DESC;
```

Quarter	Calendar Months
Q1 2020	Sept, Oct, Nov 2019
Q2 2020	Dec2019, Jan, Feb 2020
Q3 2020	Mar, Apr, May 2020
Q4 2020	Jun, Jul, Aug 2020

Quarter	total_sold_quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

Explanation:

- Since the **fiscal year for Atliq starts in September, we cannot use the built in QUARTER function** from SQL.
- I'm getting the month number from date and comparing it to numbers (9,10 or11). If it matches, then it is Q1 when it is in (12,1 or2) then it is Q2 and so on.
- Using **case statement** and above logic, we get quarter.
- We filter out the values based on fiscal year 2020.

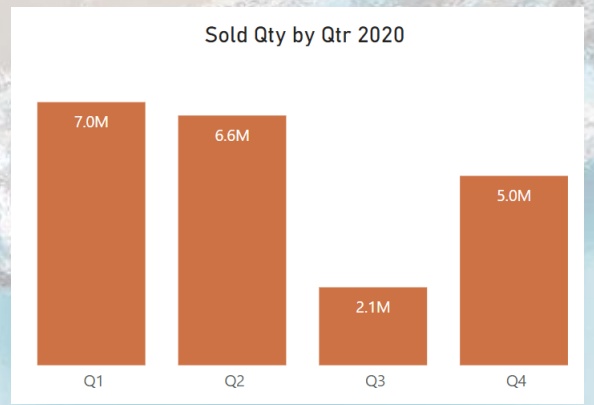
In which quarter of 2020, got the maximum total_sold_quantity?

The final output contains these fields: sorted by the total_sold_quantity, Quarter, total_sold_quantity.

Insights:

- The fiscal year of Atliq starts from September.
- I have given a table showing the quarter with the corresponding months. As you can see the **Q3 corresponds to March April and May of 2020** which is when COVID started spreading for most of the world.
- We can see from the graph the direct result of that. **Q3 sales is at the minimum** with 2.1 million quantities sold.
- We can also see the **Q4 sales have improved** 5 million which is a good thing that our company bounced back quickly from the effect of COVID.

Quarter	Calendar Months
Q1 2020	Sept, Oct, Nov 2019
Q2 2020	Dec2019, Jan, Feb 2020
Q3 2020	Mar, Apr, May 2020
Q4 2020	Jun, Jul, Aug 2020



Quarter	total_sold_quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087

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

```sql
WITH CTE AS (
SELECT
    c.channel,
    ROUND(SUM((gross_price*sold_quantity))/1000000,2) AS gross_sales_mln
FROM fact_sales_monthly s
JOIN dim_customer c
    USING (customer_code)
JOIN fact_gross_price g
        USING (product_code)
 WHERE s.fiscal_year = 2021
 GROUP BY c.channel
 ORDER BY gross_sales_mln DESC)
 SELECT
    *,
    CONCAT(ROUND(gross_sales_mln*100/SUM(gross_sales_mln) over(),2)," %") as percentage
FROM CTE;
```

channel	gross_sales_mln	percentage
Retailer	1924.17	73.22 %
Direct	406.69	15.48 %
Distributor	297.18	11.31 %

Explanation:

- We compute the **gross sales from gross price and sold quantity** of the records from 2021, group it by channel to get gross sales million per channel for the fiscal year 2021.
- The obtained result is stored in a CTE.
- To find the percentage contribution of each channel we have to **use window function over()** on the gross sales million obtained in the above CTE.

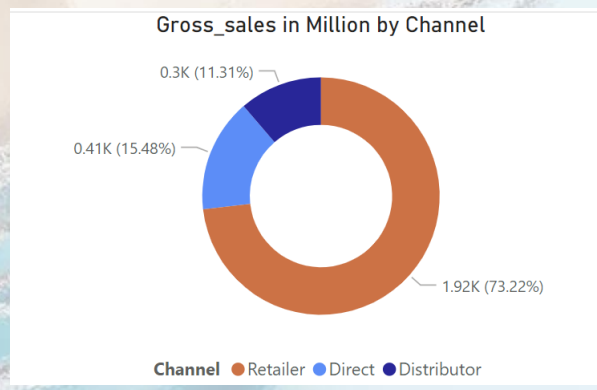
Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution?

The final output contains these fields: channel, gross_sales_mln, percentage

Insights:

- we have **only three channels** in our data. To represent the percentage contribution of each channel for the given year, I chose to **use the donut chart.**
- We can see that a retailer has been our major part major source of sales accounting almost 75%.
- The other channels, direct and distributor have a similar market share with 15.5% and 11.5% respectively.

	channel	gross_sales_mln	percentage
▶	Retailer	1924.17	73.22 %
	Direct	406.69	15.48 %
	Distributor	297.18	11.31 %



Get the Top 3 products in each division that have a high sold quantity in the fiscal year 2021?

The final output contains these fields: division, product_code, product, total_sold_quantity, rank_order

```sql
WITH CTE AS(
SELECT
    p.division,
    p.product_code,
    p.product,
    SUM(sold_quantity) AS total_sold_quantity,
    DENSE_RANK() OVER(PARTITION BY division
                ORDER BY SUM(sold_quantity) DESC) AS rank_order
FROM dim_product P
JOIN fact_sales_monthly S
        USING (product_code)
WHERE fiscal_year = 2021
GROUP BY   p.division, p.product_code, p.product
)
SELECT
    *
 FROM CTE
 WHERE rank_order ⩽3
```

division	product_code	product	total_sold_quantity	rank_order
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

Explanation:

- Calculate the total sold quantity for the fiscal year 2021 per product.
- To find the top three products for each division, we give this result a rank using **window function, DENSE_RANK** where it is partitioned by division and ordered by sold quantity.
- Store the above result in a **CTE.**
- **W**e get the final result where I filter the rank to display only those values where the rank order<=3.

Thank You

Aasdeep Singh