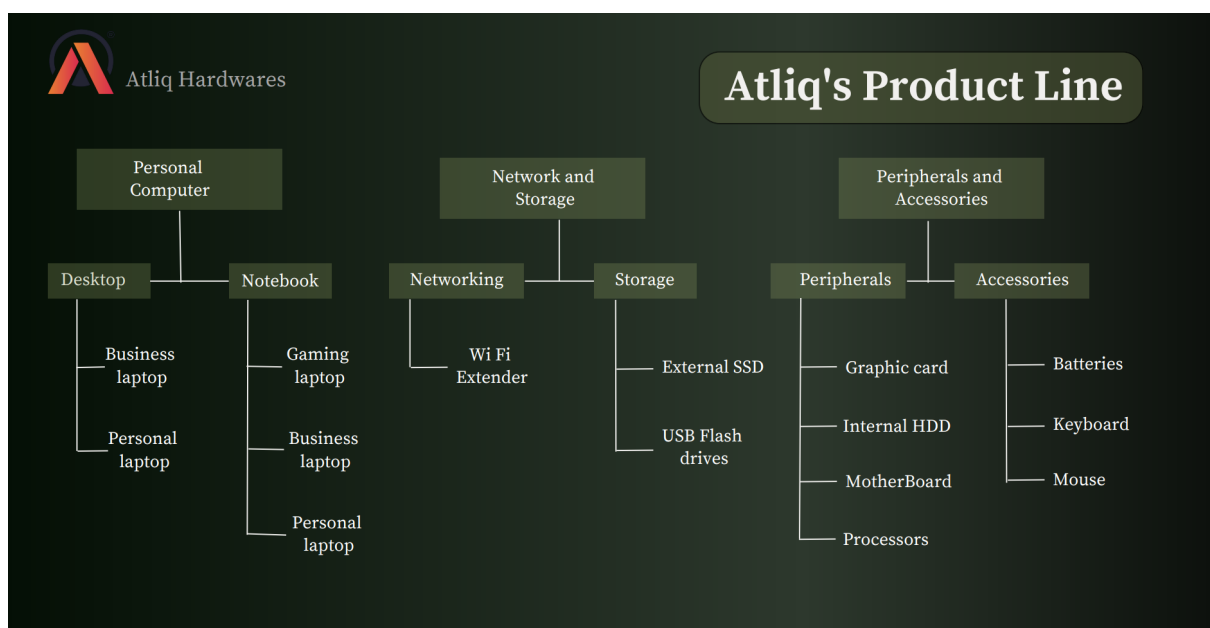


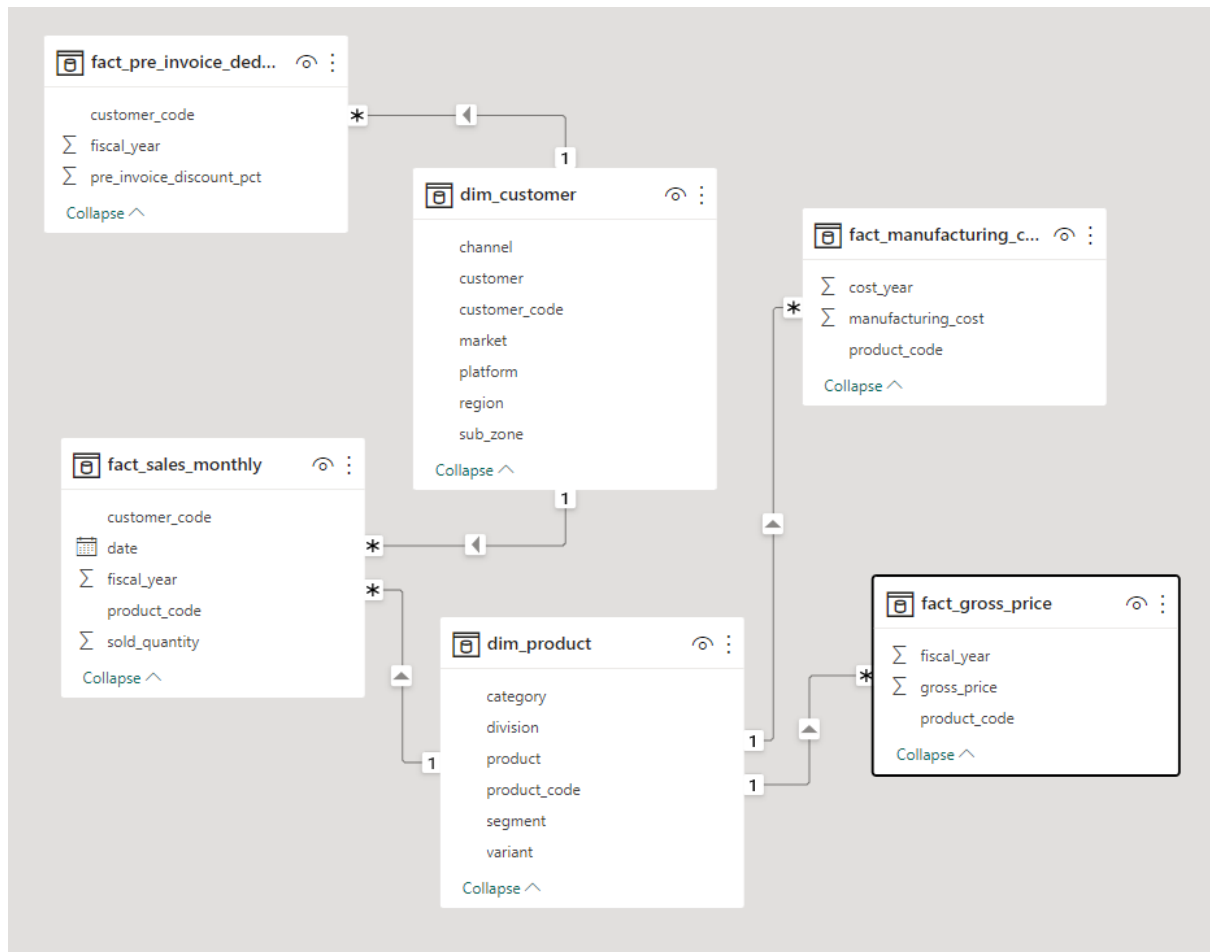


Atliq Hardware

Consumer Goods Insights



Atliq Hardware operates in several markets in the Asia Pacific region, including India, Indonesia, Japan, the Philippines, South Korea, Australia, New Zealand, and Bangladesh. The company offers a range of products, including notebooks, accessories, peripherals, desktops, storage devices, and networking equipment.



To generate meaningful insights for the company, we analyzed Atliq Hardware's data using SQL. The data we used includes the following tables:

- **Dim_product**: contains information about each product, including the product code, name, price, and category.
- **Dim_customer**: contains information about each customer, including the customer name, region, and market.
- **Fact_sales_monthly**: contains information about sales by month, including the product code, customer name, and sales figures.
- **Fact_gross_price**: This is a fact table that contains data related to the gross price of each product. This table typically contains information about the price at which each product is sold to customers before any deductions or discounts are applied.
- **Fact_manufacturing_cost**: This is another fact table that contains data related to the manufacturing cost of each product during production.
- **Fact_pre_invoice_deduction**: This is a fact table that contains data related to the pre-invoice deduction of each product.

Ad-hoc Requests, Queried Results, Insights and Visualization



Request 01:

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

```

SELECT
    DISTINCT market
-- Selects unique values from the 'market' column
FROM
    dim_customer
-- Specifies the table 'dim_customer' to query from
WHERE
    customer = 'Atliq Exclusive' AND region = 'APAC';
-- Filters the results by customer and region

```

This code selects distinct markets from the `dim_customer` table where the customer is 'Atliq Exclusive' and the region is 'APAC'. The `DISTINCT` keyword is used to remove duplicate market values from the results.

Output:

```

+-----+
| market |
+-----+
| India   |
| Indonesia |
| Japan   |
| Philippines |
| South Korea |
| Australia |
| Newzealand |
| Bangladesh |
+-----+

```



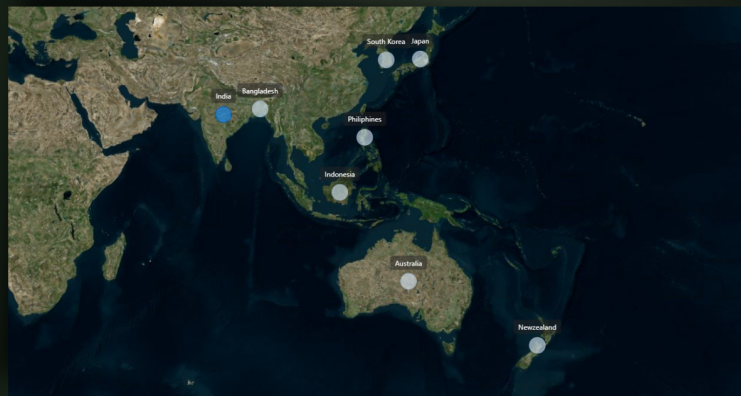
Atliq Hardware

Our Exclusive store has established its presence in 8 major markets within the APAC region.

```

+-----+
| market |
+-----+
| India   |
| Indonesia |
| Japan   |
| Philippines |
| South Korea |
| Australia |
| Newzealand |
| Bangladesh |
+-----+

```



Kamal Dave



Request 02:

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

```
-- Define two common table expressions (CTEs) to get the count of unique products sold in
-- 2020 and 2021 respectively
WITH
  unique_products_2020 AS (
    SELECT
      COUNT(DISTINCT product_code) AS count
    FROM
      fact_sales_monthly
    WHERE
      fiscal_year = 2020
  ),
  unique_products_2021 AS (
    SELECT
      COUNT(DISTINCT product_code) AS count
    FROM
      fact_sales_monthly
    WHERE
      fiscal_year = 2021
  ),
  -- Define another CTE to get the percentage change in unique products sold from
  -- 2020 to 2021
  percentage_change AS (
    SELECT
      (((unique_products_2021.count - unique_products_2020.count) / unique_products_2020.count) * 100) AS percentage_chng
    FROM
      unique_products_2020,
      unique_products_2021
  )
-- Combine the CTEs in the final SELECT statement to get the count of unique products
-- sold in 2020 and 2021, along with the percentage change
SELECT
  unique_products_2020.count AS unique_products_2020,
  unique_products_2021.count AS unique_products_2021,
  percentage_change.percentage_chng
FROM
  unique_products_2020,
  unique_products_2021,
  percentage_change;
```

The code first defines two CTEs to get the count of unique products sold in 2020 and 2021 respectively from the fact_sales_monthly table.

It then defines another CTE to calculate the percentage change in unique products sold from 2020 to 2021 by dividing the difference in counts by the count in 2020.

Finally, it combines the CTEs in the final SELECT statement to display the count of unique products sold in 2020 and 2021, along with the percentage change.

unique_products_2020	unique_products_2021	percentage_chng
245	334	36.3265



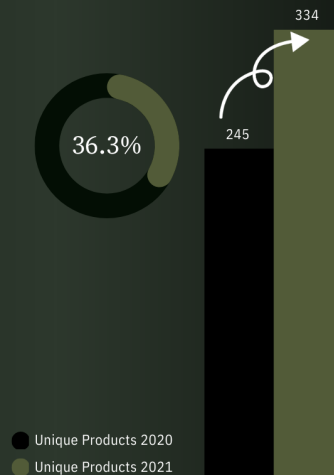
Product Growth Comparison: 2020 vs 2021

unique_products_2020	unique_products_2021	percentage_chng
245	334	36.3265

There is a significant increase in the number of unique products sold between 2020 and 2021, with a 36.33% increase

with a total of 334 unique products sold in 2021 compared to 245 unique products in 2020.

This growth in the number of unique products sold can be considered a positive sign for the business, as it shows an increase in product variety and potential revenue streams.



Kamal Dixit

The insights from this table are:

1. There was an overall increase in the number of unique products from 2020 to 2021, with 334 unique products in 2021 compared to 245 in 2020.
2. The percentage change between the two years is 36.3265%, which is a significant increase.
3. This increase in unique products could be due to a number of factors, such as an expansion in the product line, increased demand for new products, or a change in production processes.
4. It is important to continue tracking the number of unique products in future years to see if this trend continues or if there are any fluctuations in product offerings.



Request 03:

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

```
-- This SQL query selects the number of distinct products in each segment from the dim_product table, and sorts the results in descend
-- The 'segment' column represents the product category or segment, and the 'product_code' column represents the unique identifier for
-- The COUNT DISTINCT function is used to count the number of unique products in each segment, and the GROUP BY clause groups the resu
-- The ORDER BY clause sorts the results in descending order based on the count of distinct products.
select segment, count(distinct product_code) as product_count
from dim_product
group by segment
order by product_count desc;
```

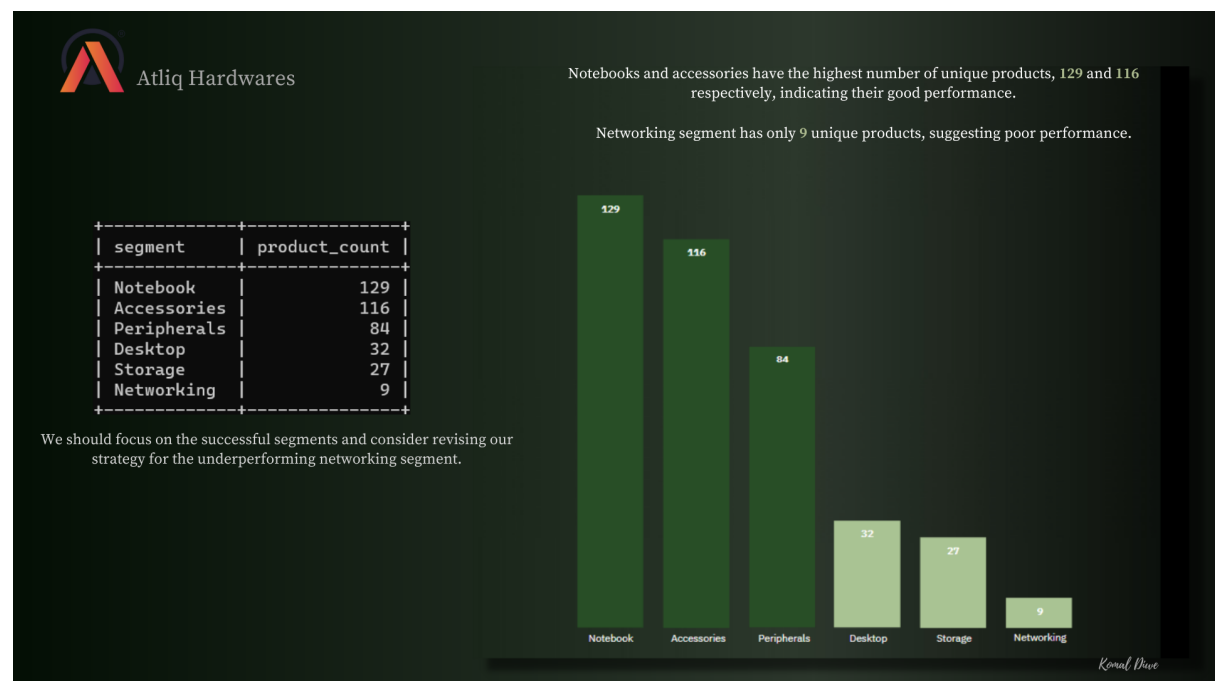
This SQL query retrieves data on the count of unique products in each segment from the dim_product table. The results are grouped by the 'segment' column, which represents the product category or segment, and the 'product_code' column is used

as the unique identifier for each product.

The COUNT DISTINCT function is used to count the number of unique products in each segment, and the GROUP BY clause groups the results by segment. The ORDER BY clause is then used to sort the results in descending order based on the count of distinct products.

This query can be useful for identifying which product segments have the most distinct products, which can be important information for various business decisions such as inventory management, marketing, and sales strategies.

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



- Notebooks and accessories are the most popular product segments, with a total of 245 products sold between them.
- The least popular product segments are networking, storage, and desktop, with only a small number of products sold in each category.
- It may be worth considering ways to increase sales in the less popular product segments, such as by offering promotions or expanding the range of products available.
- Further analysis could be done to understand the profitability of each product segment, as this would be an important factor in deciding which segments to focus on.



Request 04:

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

```
with
f_2020 as (
    select segment, product_code
    from dim_product
    join fact_sales_monthly using (product_code)
    where fiscal_year = 2020
),
f_2021 as (
    select segment, product_code
    from dim_product
    join fact_sales_monthly using (product_code)
    where fiscal_year = 2021
),
f_2020_agg as (
    select segment, count(distinct product_code) as product_count_2020
    from f_2020
    group by segment
),
f_2021_agg as (
    select segment, count(distinct product_code) as product_count_2021
    from f_2021
    group by segment
)
select
    f_2020_agg.segment,
    f_2020_agg.product_count_2020,
    f_2021_agg.product_count_2021,
    (f_2021_agg.product_count_2021 - f_2020_agg.product_count_2020) as difference
from f_2020_agg
join f_2021_agg using (segment)
order by difference desc;
```

This SQL code uses common table expressions (CTEs) to calculate the number of distinct products sold in each product segment for the fiscal years 2020 and 2021.

First, CTEs `f_2020` and `f_2021` are created to select the segment and product code columns from the `dim_product` and `fact_sales_monthly` tables, filtering for fiscal years 2020 and 2021 respectively.

Then, two more CTEs `f_2020_agg` and `f_2021_agg` are created to calculate the number of distinct products sold in each segment for the fiscal years 2020 and 2021 respectively.

Finally, the `SELECT` statement joins `f_2020_agg` and `f_2021_agg` using the segment column and calculates the difference between the number of distinct products sold in 2021 and 2020. The results are sorted in descending order based on the difference.

In summary, this SQL code is used to compare the number of distinct products sold in each product segment between two fiscal years and identify the segments with the biggest changes in product count.

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



- Accessories and Notebook are the two segments with the highest product count in both 2020 and 2021.
- All segments experienced an increase in product count from 2020 to 2021, with Accessories having the highest increase (34 products).
- Desktop, Storage, and Networking are the segments with the lowest product count, and while they also had an increase in product count from 2020 to 2021, the increase was relatively small.



Request 05:

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

The final output should contain these fields:

product_code
product
manufacturing_cost

```
-- This SQL query selects the product code, product name, and manufacturing cost of products in the "dim_product" table
-- The query joins the "fact_manufacturing_cost" table using the "product_code" column
-- It then filters the results to include only the rows where the "manufacturing_cost" column is equal to either the maximum or minimum
SELECT product_code, product, manufacturing_cost
FROM dim_product
JOIN fact_manufacturing_cost USING (product_code)
WHERE manufacturing_cost IN (
```



```

SELECT MAX(manufacturing_cost) FROM fact_manufacturing_cost
UNION
SELECT MIN(manufacturing_cost) FROM fact_manufacturing_cost
);

```

This SQL query retrieves information about products from two tables: "dim_product" and "fact_manufacturing_cost".

The "SELECT" statement specifies that we want to retrieve the "product_code", "product", and "manufacturing_cost" columns from the joined tables.


The "FROM" clause specifies that we want to join the "dim_product" table with the "fact_manufacturing_cost" table using the "product_code" column.

The "WHERE" clause filters the results to only include rows where the "manufacturing_cost" column is either the maximum or minimum value in the "fact_manufacturing_cost" table. The "IN" operator is used to specify that we want to include rows where the "manufacturing_cost" column matches either of these two values.

To find the maximum and minimum values of "manufacturing_cost" in the "fact_manufacturing_cost" table, two subqueries are used with the "UNION" operator to combine the results. The "MAX()" and "MIN()" functions are used to retrieve the maximum and minimum values, respectively.

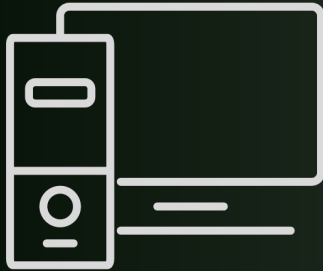
Finally, the results are returned with the "ORDER BY" clause, which sorts the rows in ascending order of "product_code".

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


Atliq Hardware

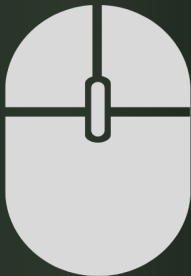
product_code	product	manufacturing_cost
A2118150101	AQ Master wired x1 Ms	0.8920
A6120110206	AQ HOME Allin1 Gen 2	240.5364

Highest manufacturing cost



AQ HOME Allin1 Gen 2 (Plus 3)
Personal Desktop
240.54\$

Lowest manufacturing cost



AQ Master wired x1 Ms (Standard 1)
Mouse
0.89\$

Kamal Dave

The manufacturing cost of "AQ Master wired x1 Ms" is 0.8920, which is relatively low compared to the cost of "AQ HOME Allin1 Gen 2" which is 240.5364.

This could be useful in determining the pricing strategy for the products or in identifying which products have higher production costs and therefore may require more attention in terms of cost management.



Request 06:

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

```
SELECT
    dim_customer.customer_code, -- select the customer code from the "dim_customer" table
    customer, -- select the customer name from the "dim_customer" table
    round(((pre_invoice_discount_pct)*100), 2) AS average_discount_percentage -- calculate the average pre-invoice discount percentag
FROM
    fact_pre_invoice_deductions -- join the "fact_pre_invoice_deductions" table
JOIN dim_customer ON fact_pre_invoice_deductions.customer_code = dim_customer.customer_code -- join the "dim_customer" table usin
WHERE
    fiscal_year = 2021 -- filter the results to include only rows where the fiscal year is 2021
    AND market = 'India' -- filter the results to include only rows where the market is India
GROUP BY
    customer_code, customer -- group the results by customer code and customer name
ORDER BY
    average_discount_percentage DESC -- sort the results in descending order based on the average discount percentage
LIMIT 6; -- limit the results to the top 6 rows
```

- The SELECT statement lists the columns we want to retrieve: customer_code, customer, and an average discount percentage (calculated as the pre-invoice discount percentage multiplied by 100 and rounded to 2 decimal places).
- The FROM clause specifies the tables we are querying from: fact_pre_invoice_deductions and dim_customer.
- The JOIN condition connects the two tables using the customer_code column.
- The WHERE clause filters the results to only include data for the fiscal year 2021 and the market of India.
- The GROUP BY clause groups the results by customer_code and customer, which means that we will get one row for each unique combination of these two columns.
- The ORDER BY clause sorts the results in descending order of average_discount_percentage.
- Finally, the LIMIT clause limits the output to only the first six rows of the result set.

So the overall purpose of this query is to find the top six customers in India with the highest average pre-invoice discount percentage for the fiscal year 2021.

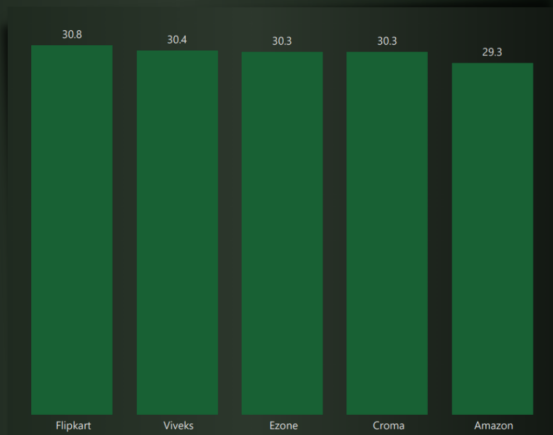
customer_code	customer	average_discount_percentage
90002009	Flipkart	30.83
90002006	Viveks	30.38
90002003	Ezone	30.28
90002002	Croma	30.25
90002016	Amazon	29.33
90002011	Atliq Exclusive	27.93



Top 5 Customers with Highest Average Pre-Invoice Discount Percentage
in the Indian Market for Fiscal Year 2021

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

In 2021, the top 5 customers were offered similar pre-invoice discount percentages, with Flipkart being the most heavily discounted customer in the Indian market at 30.83%.



Kamal Dixit

- The top customers with the highest average discount percentage are Flipkart, Viveks, and Ezone, all with an average discount percentage of over 30%.
- Croma and Atliq Exclusive have a slightly lower average discount percentage than the top three, but are still within the range of 27-30%.
- Amazon has the lowest average discount percentage among the listed customers, at 29.33%.



Request 07:

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

```
-- This SQL query selects the gross sales amount for each month and year for the customer 'Atliq Exclusive'.
-- The EXTRACT function is used to extract the month and year from the 'date' column in the fact_sales_monthly table.
-- The ROUND function is used to round the gross sales amount to 2 decimal places.
SELECT
    EXTRACT(MONTH FROM fact_sales_monthly.date) AS Month,
    EXTRACT(YEAR FROM fact_sales_monthly.date) AS Year,
    ROUND(SUM((gross_price * sold_quantity)), 2) as gross_sales_amount
FROM
    fact_sales_monthly
JOIN
    dim_customer USING (customer_code)
JOIN
    fact_gross_price USING (product_code)
WHERE
    dim_customer.customer = 'Atliq Exclusive' -- Filter the results to include only the rows where the customer name is 'Atliq Exclusive'
GROUP BY
    Month, -- Group the results by month.
    Year -- Group the results by year.
ORDER BY
```

```
Year ASC, -- Order the results by year in ascending order.
Month ASC; -- Order the results by month in ascending order.
```

This SQL query retrieves monthly gross sales amount for a specific customer 'Atliq Exclusive'. Here's a breakdown of what it does:

1. The **SELECT** statement retrieves three columns:
 - **EXTRACT(MONTH FROM fact_sales_monthly.date) AS Month** extracts the month value from the 'date' column of the fact_sales_monthly table and aliases it as 'Month'.
 - **EXTRACT(YEAR FROM fact_sales_monthly.date) AS Year** extracts the year value from the 'date' column of the fact_sales_monthly table and aliases it as 'Year'.
 - **ROUND(SUM((gross_price * sold_quantity)), 2) as gross_sales_amount** calculates the gross sales amount by multiplying the 'gross_price' and 'sold_quantity' columns of the fact_gross_price table, summing them up and rounding the result to two decimal places. The result is aliased as 'gross_sales_amount'.
2. The **FROM** statement includes the fact_sales_monthly table.
3. The **JOIN** statements join the dim_customer and fact_gross_price tables using the 'customer_code' and 'product_code' columns, respectively.
4. The **WHERE** statement filters the results to only include data for the customer 'Atliq Exclusive'.
5. The **GROUP BY** statement groups the results by month and year.
6. The **ORDER BY** statement sorts the results in ascending order of year and month.

Overall, this query provides a summary of monthly gross sales amount for the customer 'Atliq Exclusive' over time, making it easy to spot trends and fluctuations.

Month	Year	gross_sales_amount
9	2019	9092670.34
10	2019	10378637.60
11	2019	15231894.97
12	2019	9755795.06
1	2020	9584951.94
2	2020	8083995.55
3	2020	766976.45
4	2020	800071.95
5	2020	1586964.48
6	2020	3429736.57
7	2020	5151815.40
8	2020	5638281.83
9	2020	19530271.30
10	2020	21016218.21
11	2020	32247289.79
12	2020	20409063.18
1	2021	19570701.71
2	2021	15986603.89
3	2021	19149624.92
4	2021	11483530.30
5	2021	19204309.41
6	2021	15457579.66
7	2021	19044968.82
8	2021	11324548.34

Monthly **Gross Sales** from September 2019 to August 2021

The sales amount was relatively **stable from September 2019 to February 2020**, but then dropped significantly in March and April 2020, which may be related to the COVID-19 pandemic.

However, the sales amount then began to **recover from May 2020** and has generally increased since then.

The sales amount in **November 2020** was much **higher** than in previous months, at over 32 million dollars.



Kamal Datta

shows the gross sales amount for each month from September 2019 to August 2021.

- The gross sales amount was highest in November 2020, at 32.2 million, and lowest in March 2020, at 767 thousand.
- There is a clear dip in sales between November 2019 and January 2020, with sales gradually recovering in the following months before a steep increase starting in September 2020.
- There is a noticeable drop in sales in April 2021, followed by a recovery in May and a subsequent decline in June and July.
- Sales in August 2021 are relatively low compared to the same month in previous years, potentially indicating a recent decrease in demand.



Request 08:

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

```
-- This code uses a WITH clause to create a CTE (Common Table Expression) called 'quarters', which calculates the quarter of the year
WITH quarters AS (
  SELECT *,
    CASE
      WHEN MONTH(date) IN (9, 10, 11) THEN 'Q1' -- If the month is September, October, or November, then it's in Q1.
      WHEN MONTH(date) IN (12, 1, 2) THEN 'Q2' -- If the month is December, January, or February, then it's in Q2.
      WHEN MONTH(date) IN (3, 4, 5) THEN 'Q3' -- If the month is March, April, or May, then it's in Q3.
      WHEN MONTH(date) IN (6, 7, 8) THEN 'Q4' -- If the month is June, July, or August, then it's in Q4.
    END AS Quarter
  FROM fact_sales_monthly
  WHERE fiscal_year = 2020 -- Only select the data for the fiscal year 2020.
)

-- This code selects the quarter and the total sold quantity for each quarter from the 'quarters' table and groups the data by the qua
-- The total sold quantity is calculated by summing the 'sold_quantity' column.
-- The results are then ordered in descending order by the total sold quantity.
```

```
SELECT Quarter, SUM(sold_quantity) AS total_sold_quantity
FROM quarters
GROUP BY Quarter
ORDER BY total_sold_quantity DESC;
```

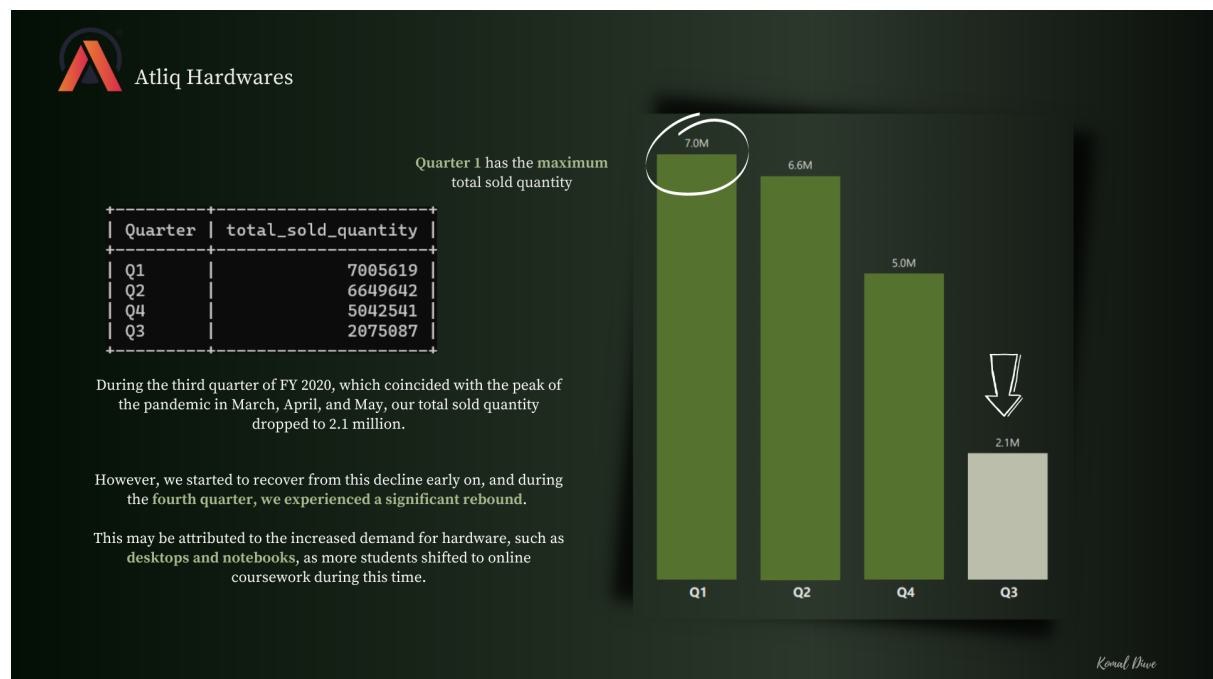
The code is using a common table expression (CTE) to create a temporary table named "quarters". The "quarters" table is derived from the "fact_sales_monthly" table, where only the data for the fiscal year 2020 is selected.

The "quarters" table includes an additional column "Quarter" that categorizes each row into one of four quarters based on the month of the "date" column. This is done using a "CASE" statement that assigns the appropriate quarter based on the month.

After creating the "quarters" table, the code performs a query that selects the "Quarter" column and the sum of "sold_quantity" for each quarter. The result set is grouped by the "Quarter" column and ordered in descending order based on the total sold quantity.

Overall, the query calculates the total sold quantity for each quarter of the fiscal year 2020 and returns the results grouped by quarter in descending order of total sold quantity.

Quarter	total_sold_quantity
Q1	7005619
Q2	6649642
Q4	5042541
Q3	2075087



- The total sold quantity of products is highest in Q1 and lowest in Q3.
- Q1 and Q2 combined have the highest total sold quantity of products.
- Q3 has the lowest total sold quantity of products.
- Q4 has a lower total sold quantity of products than Q1 and Q2, but higher than Q3.

These insights can be used to identify seasonal trends in the sales of the company's products and to make strategic decisions related to inventory management, production, and marketing.



Request 09:

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

```
WITH channel_gross AS (  
  -- select the channel and the total gross sales in millions of USD for each channel  
  SELECT  
    dim_customer.channel,  
    ROUND(SUM(gross_price * sold_quantity), 2) AS gross_sales_mln  
  FROM fact_sales_monthly  
  JOIN dim_customer ON fact_sales_monthly.customer_code = dim_customer.customer_code  
  JOIN fact_gross_price ON fact_sales_monthly.product_code = fact_gross_price.product_code  
  WHERE fact_sales_monthly.fiscal_year = 2021 -- filter the fact_sales_monthly table to only include data from 2021  
  GROUP BY dim_customer.channel -- group the results by channel  
  ORDER BY gross_sales_mln DESC -- order the results by gross sales in descending order  
)  
-- select the channel, gross sales in millions of USD, and the percentage of each channel's gross sales out of the total gross sales  
SELECT  
  channel,  
  gross_sales_mln,  
  ROUND((gross_sales_mln * 100 / sum(gross_sales_mln) over()), 3) AS percentage  
FROM channel_gross; -- use the "channel_gross" subquery as the source for the results
```

This code first creates a common table expression (CTE) called "channel_gross" that calculates the total gross sales in millions of USD for each channel based on the fact_sales_monthly, dim_customer, and fact_gross_price tables. The CTE filters the fact_sales_monthly table to only include data from the fiscal year 2021, groups the results by channel, and orders the results by gross sales in descending order.

The code then uses the "channel_gross" CTE as the source for the results, selecting the channel, gross sales in millions of USD, and the percentage of each channel's gross sales out of the total gross sales. The percentage is calculated using a window function that calculates the sum of the gross sales across all channels, allowing the percentage for each channel to be calculated as a ratio of the channel's gross sales to the total gross sales.

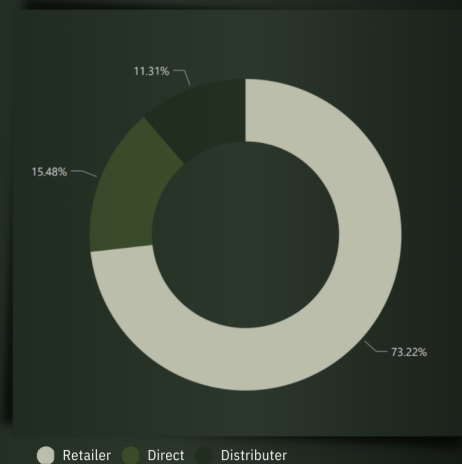
channel	gross_sales_mln	percentage
Retailer	1924170397.91	73.217
Direct	406686873.90	15.475
Distributor	297175879.72	11.308



Sales Distribution by Channel

channel	gross_sales_mln	percentage
Retailer	1924170397.91	73.217
Direct	406686873.90	15.475
Distributor	297175879.72	11.308

The majority of sales, 73.217%, came from retailers, while 15.475% came directly from the company and 11.308% came through distributors.



Kamal Dixit

- The Retailer channel had the highest gross sales amount,
- The Direct channel had an approximately 15.475% of the total gross sales amount.
- The Distributor channel is approximately 11.308% of the total gross sales amount.
- The Retailer channel is the largest sales channel, contributing to more than 73% of the total gross sales amount.



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

```
-- Define a common table expression called `division_sales`
WITH division_sales AS (
  SELECT
    dp.division, -- Select the `division` column from `dim_product`
    fsm.product_code, -- Select the `product_code` column from `fact_sales_monthly`
    dp.product, -- Select the `product` column from `dim_product`
    SUM(fsm.sold_quantity) AS total_sold_quantity, -- Calculate the total `sold_quantity` and give it an alias
    RANK() OVER (PARTITION BY dp.division ORDER BY SUM(fsm.sold_quantity) DESC) AS rank_order -- Rank the products within their division
  FROM fact_sales_monthly fsm -- Join the `fact_sales_monthly` table
  JOIN dim_product dp ON fsm.product_code = dp.product_code -- Join the `dim_product` table
  WHERE fsm.fiscal_year = 2021 -- Only consider sales in the year 2021
  GROUP BY dp.division, fsm.product_code, dp.product -- Group the results by `division`, `product_code`, and `product`
)

-- Select data from the `division_sales` CTE and return the top three products for each division
SELECT
  division_sales.division, -- Select the `division` column from the CTE
  division_sales.product_code, -- Select the `product_code` column from the CTE
```



```

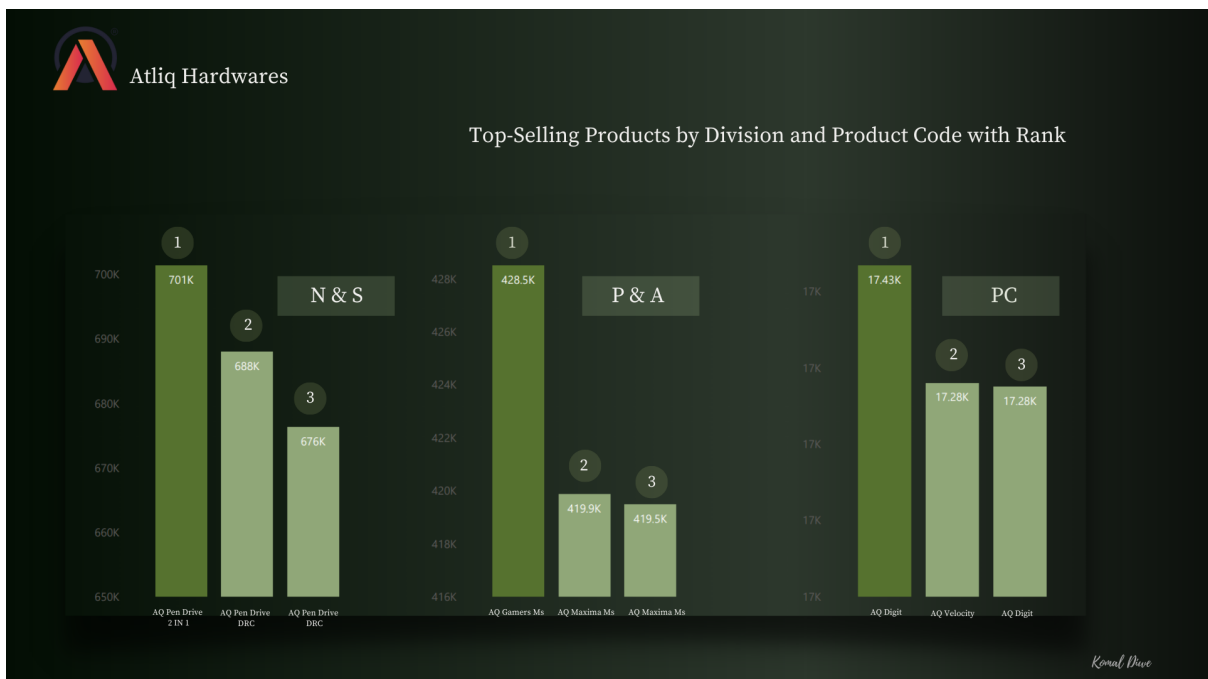
division_sales.product, -- Select the 'product' column from the CTE
division_sales.total_sold_quantity, -- Select the 'total_sold_quantity' column from the CTE
division_sales.rank_order -- Select the 'rank_order'

```

This SQL code has two parts:

1. The first part defines a common table expression (CTE) called `division_sales`. It selects the `division`, `product_code`, `product` and the sum of `total_sold_quantity` from the `fact_sales_monthly` table and the `dim_product` table, filtered by the year 2021. It groups the result by `division`, `product_code`, and `product`. The `RANK()` function is used to calculate the rank of each product within its division, based on the total sold quantity, and the result is assigned to the `rank_order` column.
2. The second part of the code selects data from the `division_sales` CTE, and returns only the top three products for each division, based on their rank order.

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





Top-Selling Products by Division and Product Code with Rank

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	676205	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	A4319110206	AQ Velocity	17200	2
PC	A4218110208	AQ Digit	17275	3

The product "AQ Pen Drive 2 IN 1" sold the highest quantity overall, followed by "AQ Pen Drive DRC" and "AQ Gamers Ms".

The division "PC" has the three lowest selling products.

Kamal Dixit

These insights show the best-selling products by division based on their total sold quantity and rank order.

1. N & S division: The top 3 best-selling products in this division are all AQ Pen Drives. The AQ Pen Drive 2 IN 1 has the highest total sold quantity, followed by two variations of the AQ Pen Drive DRC.
2. P & A division: The best-selling product in this division is AQ Gamers Ms, which has the highest total sold quantity. The next two products in rank order are both variations of AQ Maxima Ms.
3. PC division: The best-selling product in this division is AQ Digit, followed by AQ Velocity and another variation of AQ Digit.