



Consumer Goods Ad - Hoc Insights



INTRODUCTION

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



PROBLEM STATEMENT

- ❖ Management do not get enough insights to make quick and data-informed decisions.
- ❖ Their senior data analyst Tony Sharma want someone which is good at both tech and soft skills , so he decided to conduct SQL challenge which help him to understand both the skill.
- ❖ He provided 10 ad-hoc requests pdf for which the business need insights for which you have to run a SQL query.



Request 1

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

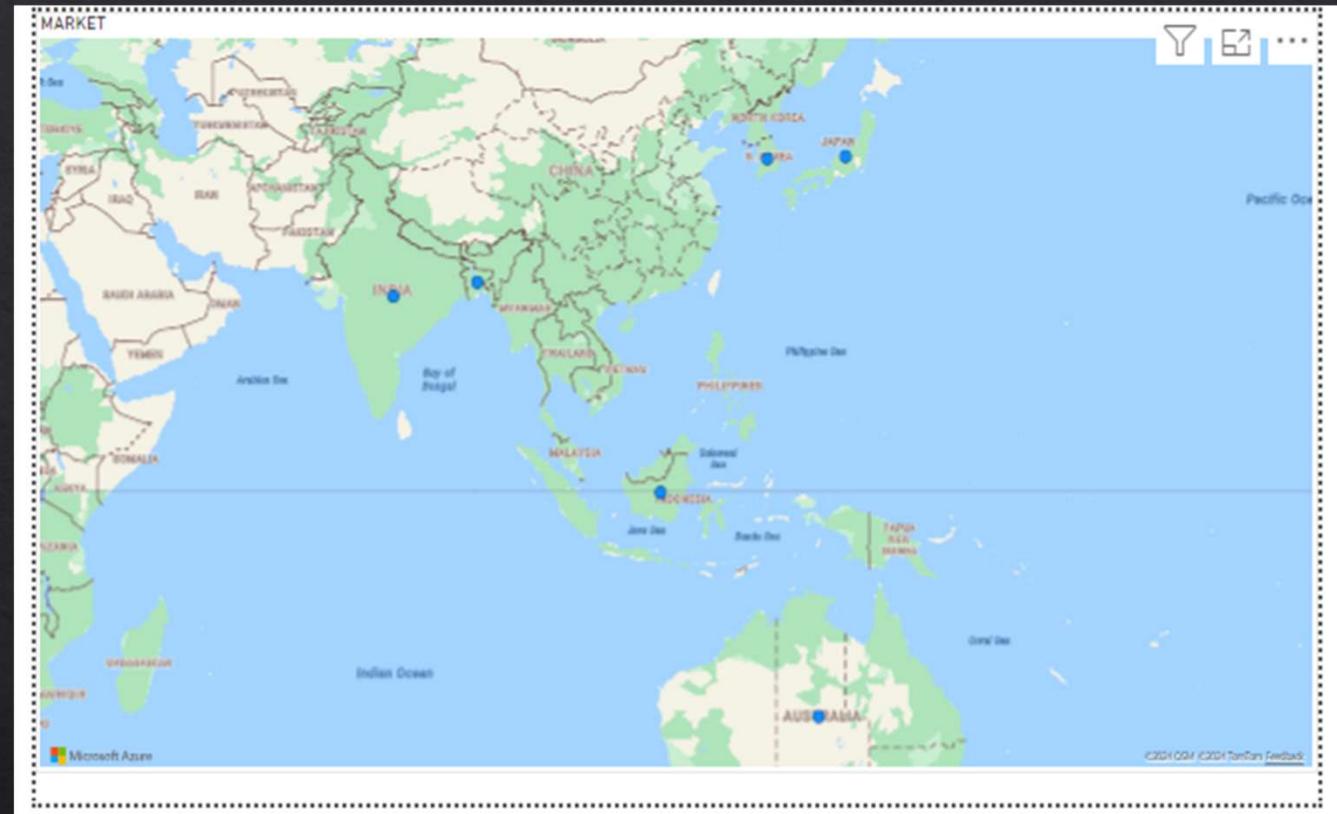
SQL query:-

Select

MARKET from dim_customer

where customer = "atliq exclusive"

and region = "APAC";



There are 8 countries in which AtliQ Hardware operate.



Request 2

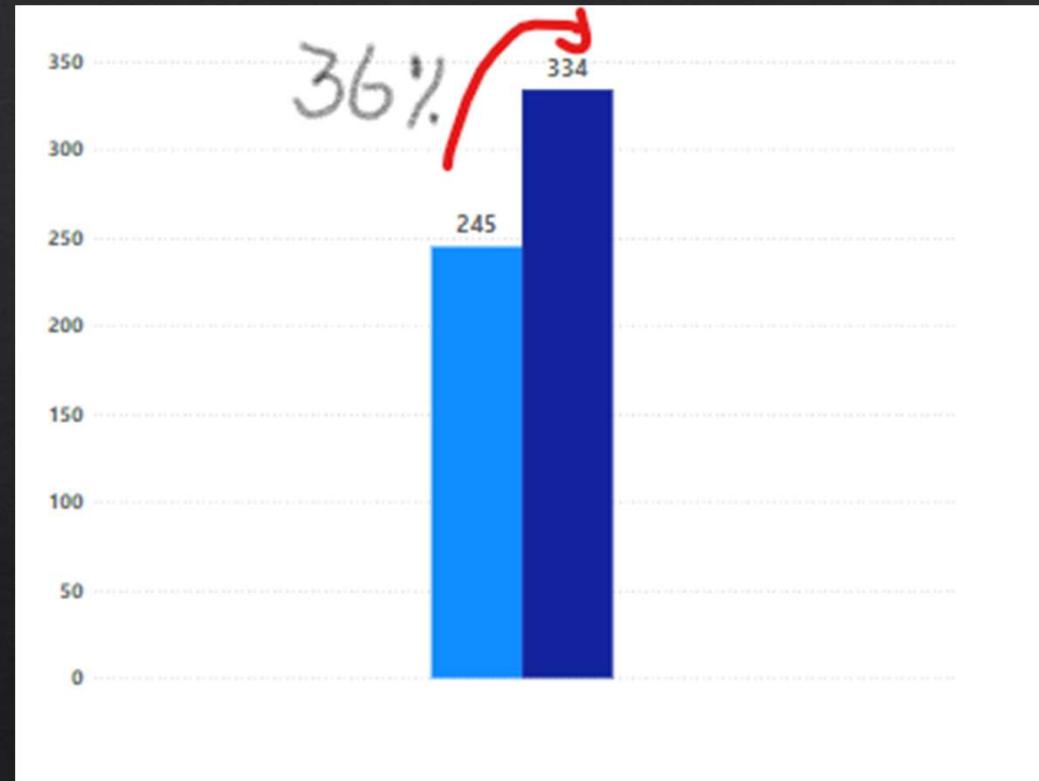
2. What is the percentage of unique product increase in 2021 vs. 2020? The final output contains these fields, unique_products_2020 unique_products_2021 percentage_chg?

SQL Query:-

```
with cte1 as
(select count(distinct(product_code)) as unique_products_2020
from fact_sales_monthly
where fiscal_year = 2020),
cte2 as
( select count(distinct(product_code)) as unique_products_2021 from fact_sales_monthly
where fiscal_year = 2021)
select unique_products_2020,unique_products_2021,
round((unique_products_2021 - unique_products_2020)*100/unique_products_2020,2)
as percentage_chg
from cte2
cross join cte1;
```



	unique_products_2020	unique_products_2021	percentage_chg
•	245	334	36.33



The unique products in 2020 is 245 products but in 2021 it changes to 334 . So percentage chg of unique products from 2020 to 2021 is 36.33%.



Request 3

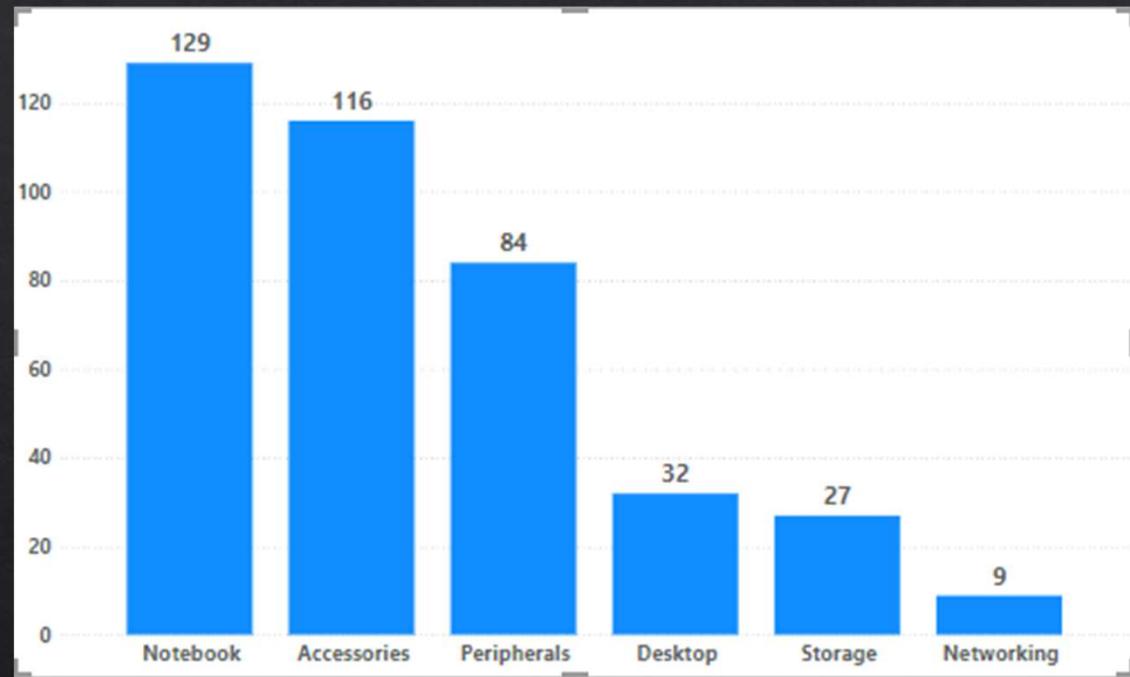
3. Provide a report with all the unique product counts for each segment and sort them in descending order of product counts. The final output contains 2 fields, segment , product_count ?

SQL Query:-

```
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



- ❑ There are 6 segment in product.
- ❑ Notebook segment has the highest product count.
- ❑ Networking has the lowest product count .



Request 4

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 ?

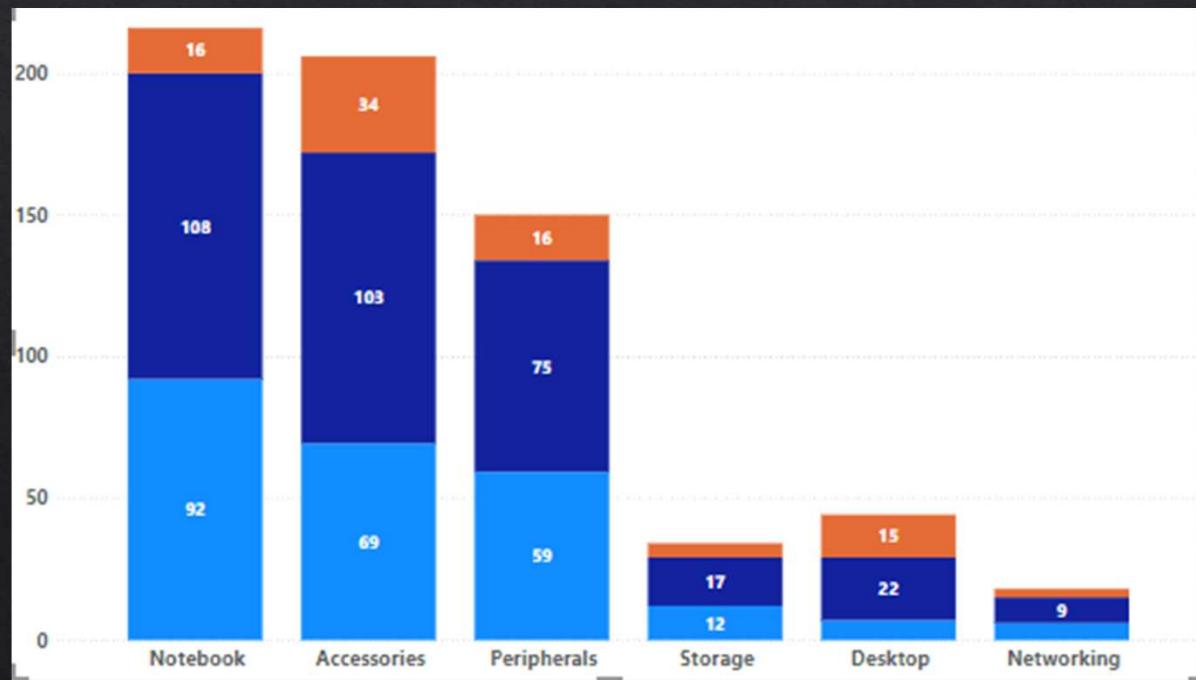
SQL Query:-

```
with cte1 as
(select p.segment , s.fiscal_year , s.product_code
from dim_product p
join fact_sales_monthly s
on p.product_code = s.product_code),
cte2 as
(select segment ,
count(distinct(case when fiscal_year = 2020 then product_code end)) as product_count_2020,
count(distinct(case when fiscal_year = 2021 then product_code end)) as product_count_2021
from cte1
group by segment)
select * ,(product_count_2021 - product_count_2020) as difference
from cte2
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

From this , we got the know about the product count in segment in year 2020 and 2021 and get the difference of the product count i.e. how many products count in the segment in year 2020 and in year 2021.





Request 5

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

SQL Query:-

```
(select m.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 manufacturing_cost desclimit 1)  
Union  
(select m.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 manufacturing_cost asc  
limit 1);
```

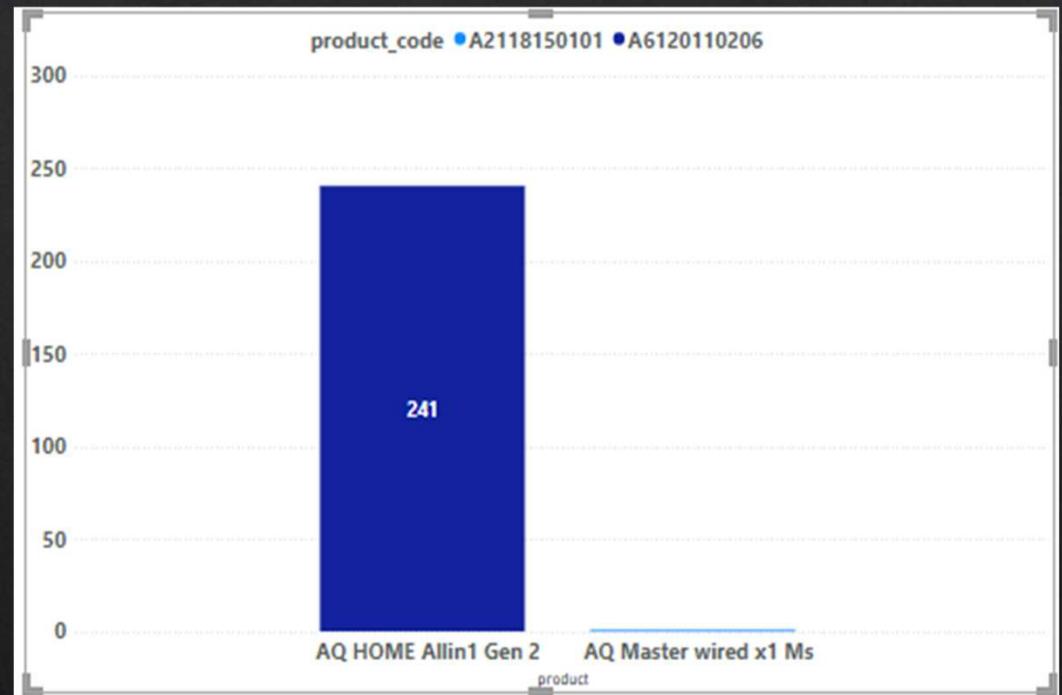


	product_code	product	manufacturing_cost
▶	A6120110206	AQ HOME Allin1 Gen 2	240.5364
	A2118150101	AQ Master wired x1 Ms	0.8920

From this query , we got to know about the product and their product code who have lowest and highest manufacturing costs.

Lowest manufacturing cost = 0.892

Highest manufacturing cost = 240.5364





Request 6

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?

SQL Query:-

```
with cte1 as
( select c.customer_code , c.customer , p.pre_invoice_discount_pct
from dim_customer c
join fact_pre_invoice_deductions p
on c.customer_code = p.customer_code
where fiscal_year = 2021 and market = "india")
select customer_code, customer, concat(round(avg(pre_invoice_discount_pct)*100,2), "%") as
avg_discount_pct
from cte1
group by customer_code , customer
order by avg(pre_invoice_discount_pct)*100 desc
limit 5;
```

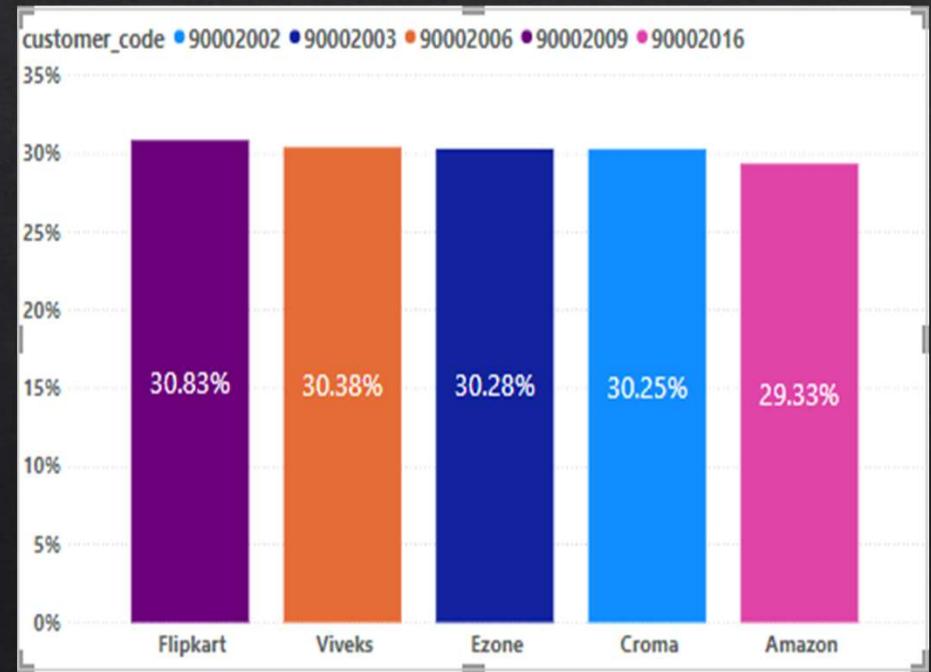


	customer_code	customer	avg_discount_pct
▶	90002009	Flipkart	30.83%
	90002006	Viveks	30.38%
	90002003	Ezone	30.28%
	90002002	Croma	30.25%
	90002016	Amazon	29.33%

In this query , we have to find which top 5 customers got the highest average discount percentage in India in year 2021 .

Flipkart , Viveks , Ezone , Croma , Amazon

This are the top 5 customers and got average discount percentage approximately same .





Request 7

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?

SQL Query:-

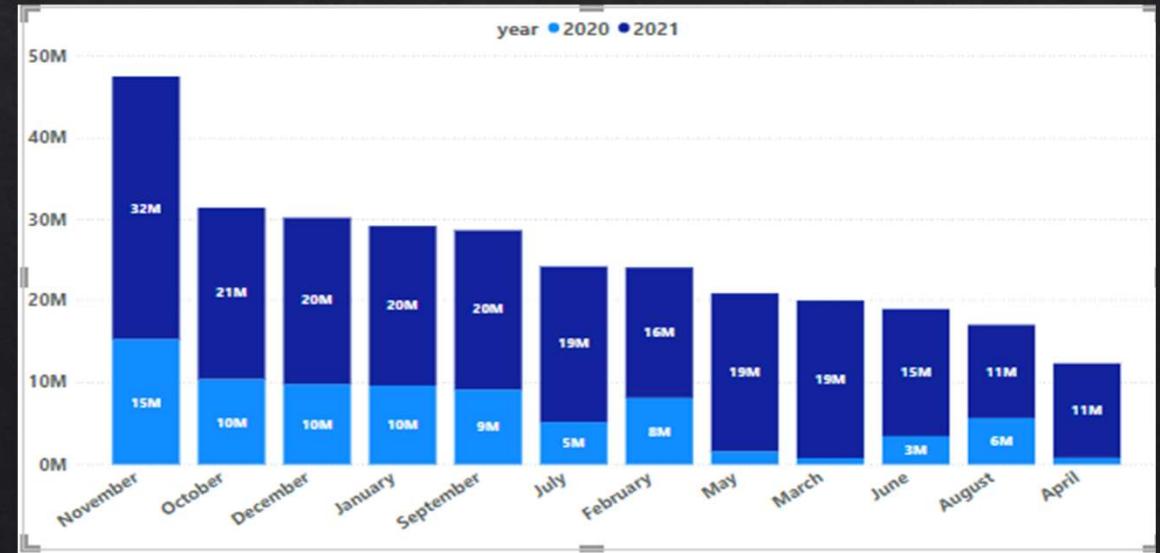
```
select monthname(s.date)as month,
s.fiscal_year as year ,
sum(sold_quantity*gross_price) as gross_sales
from fact_sales_monthly s
join fact_gross_price g
on s.product_code=g.product_code
join dim_customer c
on s.customer_code=c.customer_code
where customer = "atliq exclusive"
group by month , year
order by year asc;
```



	month	year	gross_sales
▶	September	2020	9092670.3392
	October	2020	10378637.5961
	November	2020	15231894.9669
	December	2020	9755795.0577
	January	2020	9584951.9393
	February	2020	8083995.5479
	March	2020	766976.4531
	April	2020	800071.9543
	May	2020	1586964.4768
	June	2020	3429736.5712
	July	2020	5151815.4020
	August	2020	5638281.8287
	September	2021	19530271.3028

	month	year	gross_sales
	August	2020	5638281.8287
	September	2021	19530271.3028
	October	2021	21016218.2095
	November	2021	32247289.7946
	December	2021	20409063.1769
	January	2021	19570701.7102
	February	2021	15986603.8883
	March	2021	19149624.9239
	April	2021	11483530.3032
	May	2021	19204309.4095
	June	2021	15457579.6626
	July	2021	19044968.8164
	August	2021	11324548.3409

In this query , we find the gross sales in the months in year 2020 and 2021.





Request 8

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?

SQL Query:-

```
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;
```



	Quarter	total_sold_quantity
▶	Q1	7005619
	Q2	6649642
	Q3	2075087
	Q4	5042541

In this query , we find total sold quantity in year 2020 but group by quarter .

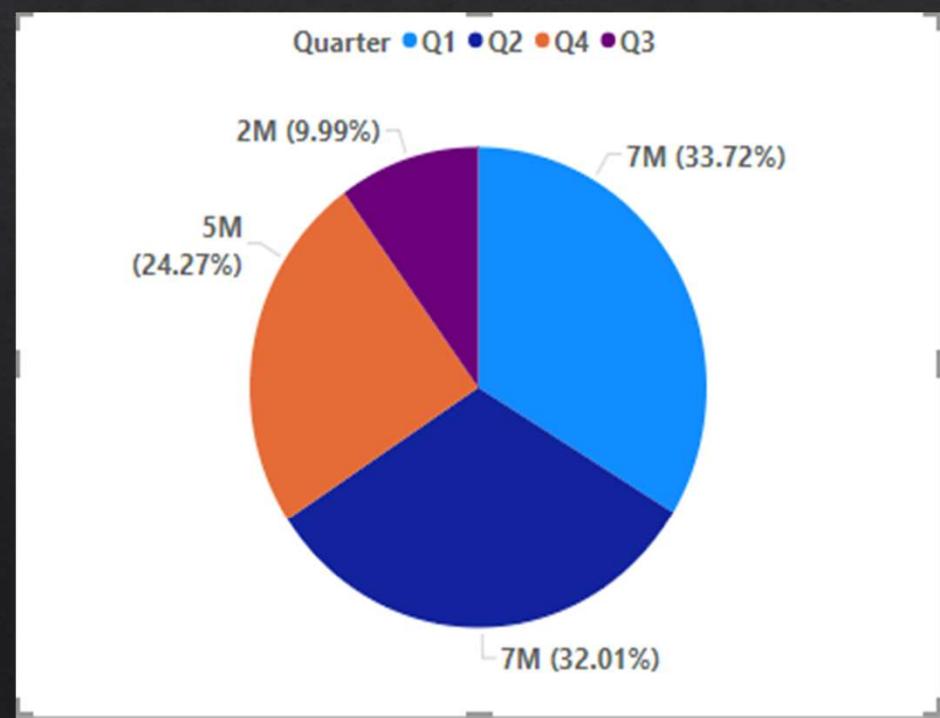
In AtliQ Hardware , fiscal year from September to August .

So September to November --- Q1

December to February ----- Q2

March to May ----- Q3

June to August ----- Q4





Request 9

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?

SQL Query:-

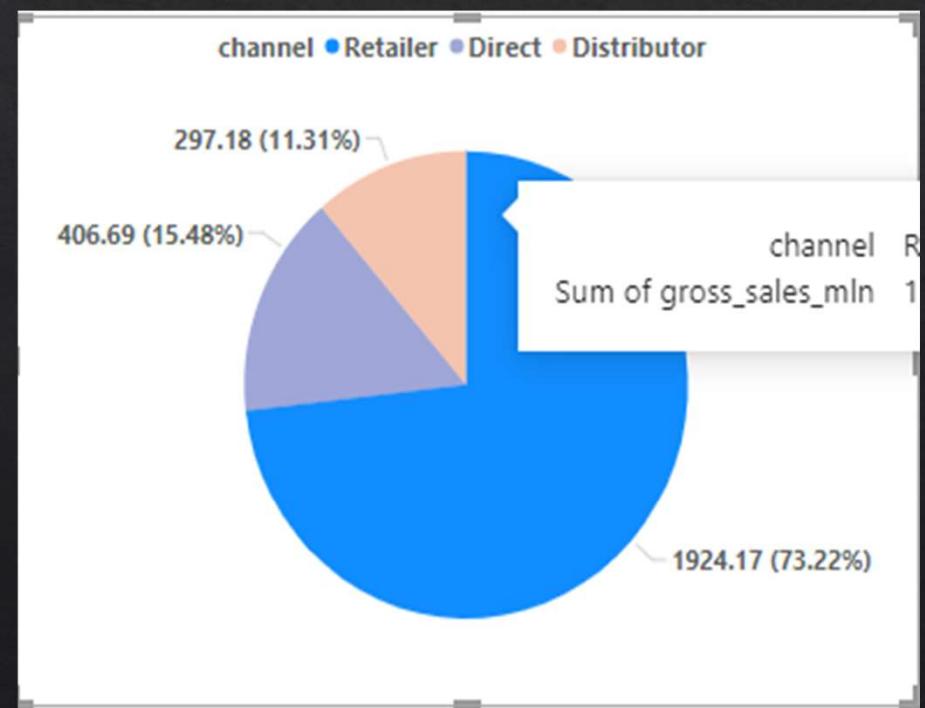
```
with cte1 as
  (select channel , round(sum((gross_price*sold_quantity))/1000000,2) as gross_sales_mln
   from fact_sales_monthly s
   join fact_gross_price g
     using (product_code)
   join dim_customer c
     using (customer_code)
   where s.fiscal_year = 2021
   group by channel)
  select * , concat(round(gross_sales_mln*100/
  (select sum(gross_sales_mln) from cte1),2),"%) as pct_contribution
  from cte1
  order by pct_contribution desc;
```



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

In this query , we find which channel give us how much sales and their contribution in percentage in total .

We find from retailer channel give us the highest gross sales which contribute almost 73% in gross sales .
Then followed by direct channel and then distributor channel.





Request 10

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 ?

SQL Query:-

```
with cte1 as
(select division , s.product_code ,product ,
sum(sold_quantity) as total_sold_quantity,
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)
group by division , product , product_code)
select * from cte1
where rank_order in (1,2,3)
order by division;
```



division	product_code	product	total_sold_quantity	rank_order
N & S	A6720160103	AQ Pen Drive 2 IN 1	115922	1
N & S	A6818160201	AQ Pen Drive DRC	1128104	2
N & S	A6419160301	AQ Clx1	729696	3
P & A	A2319150302	AQ Gamers Ms	683634	1
P & A	A2219150204	AQ Master wireless x1 Ms	682321	2
P & A	A2319150306	AQ Gamers Ms	681531	3
PC	A4218110202	AQ Digit	26012	1
PC	A4319110306	AQ Velocity	25978	2
PC	A4118110107	AQ Aspiron	25963	3

In this query , we have to find the top 3 products in each division with their product code on the basis of total sold quantity , also with rank order.

