Atliq Hardware Consumer Goods ad–hoc Insights

COMPANY

Atliq Hardware is one of the leading computer hardware producers in India and well expanded in other countries too.

AIM

To help the top management to make data informed decision by providing Insights

DATA UNDERSTANDING

Dataset is available for fiscal year 2020 and 2021

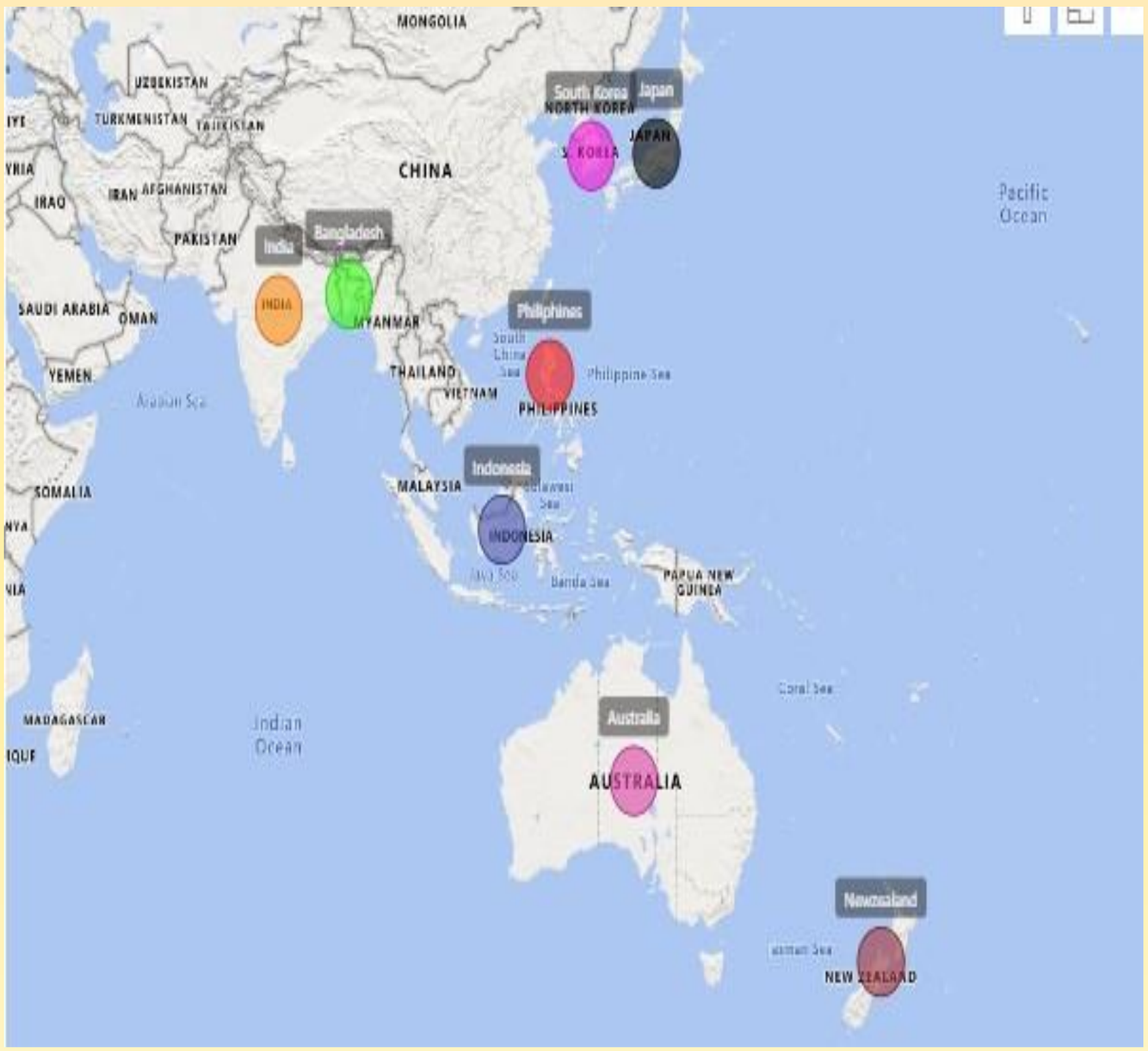
Atliq Hardware's fiscal year starts on 1st September and ends on 31st August

Atliq Hardware database (atliq_hardware_db) has 6 tables :

1) dim_customer: contains customer-related data
2) dim_product: contains product-related data
3) fact_gross_price: contains gross price information for each product
4) fact_manufacturing_cost: contains the cost incurred in the production of each product
5) fact_pre_invoice_deductions: contains pre-invoice deductions information for each product
6) fact_sales_monthly: contains monthly sales data for each product.

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

```sql
select
      distinct(market)
from gdb023.dim_customer
where
      customer = 'Atliq Exclusive'
      and region = 'APAC';
```

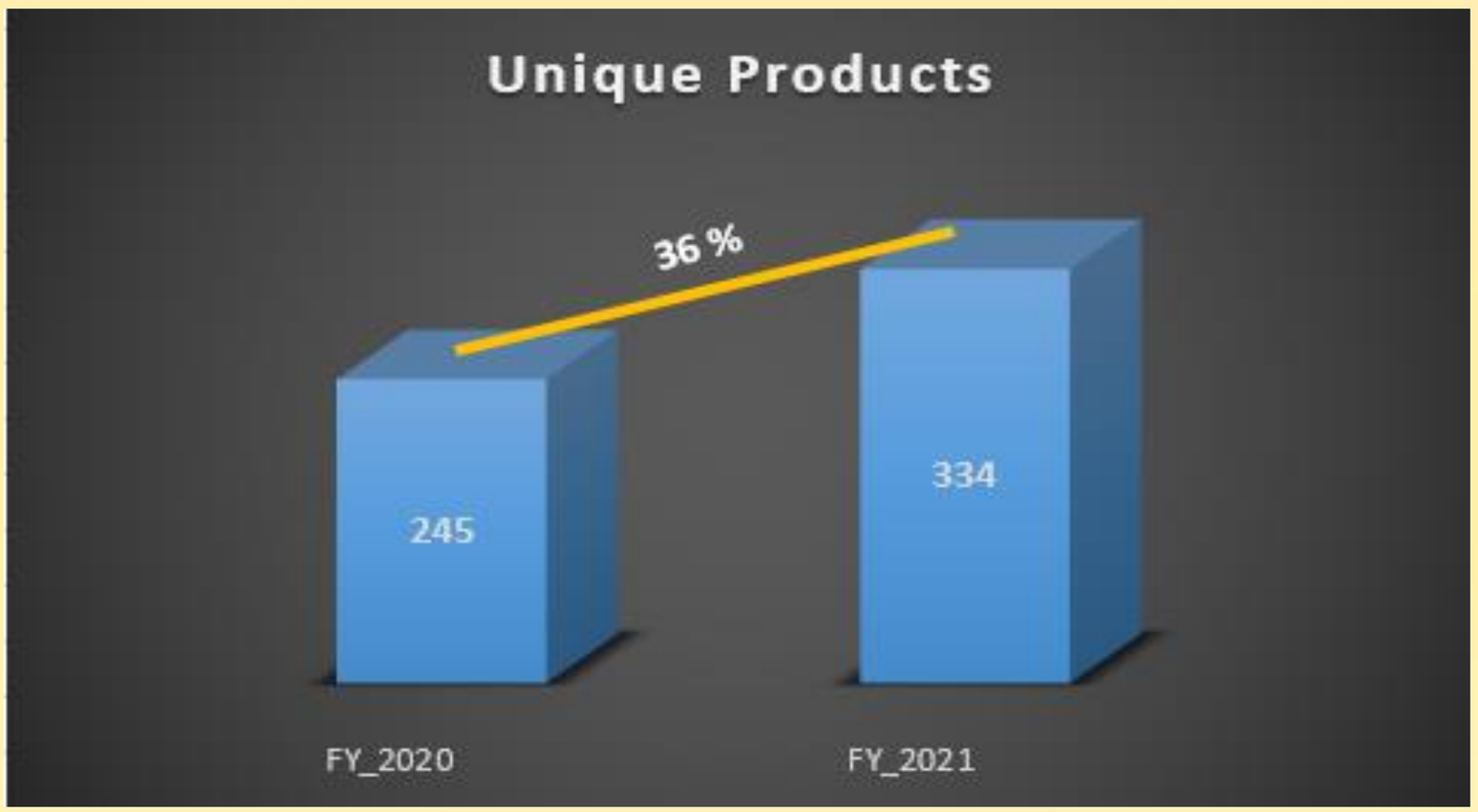


Result

- India
- Australia
- New Zealand
- Bangladesh
- Indonesia
- Japan
- 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**

```
    with cte as (
    select
        sum(case when cost_year = 2021 then 1 else 0 end) as unique_products_2021,
        sum(case when cost_year = 2020 then 1 else 0 end) as unique_products_2020
    from gdb023.fact_manufacturing_cost
)

select
    unique_products_2020,
    unique_products_2021,
    concat(round(((unique_products_2021 - unique_products_2020)/unique_products_2020)*100,2),'%') as
percentage_chg
from cte;
```



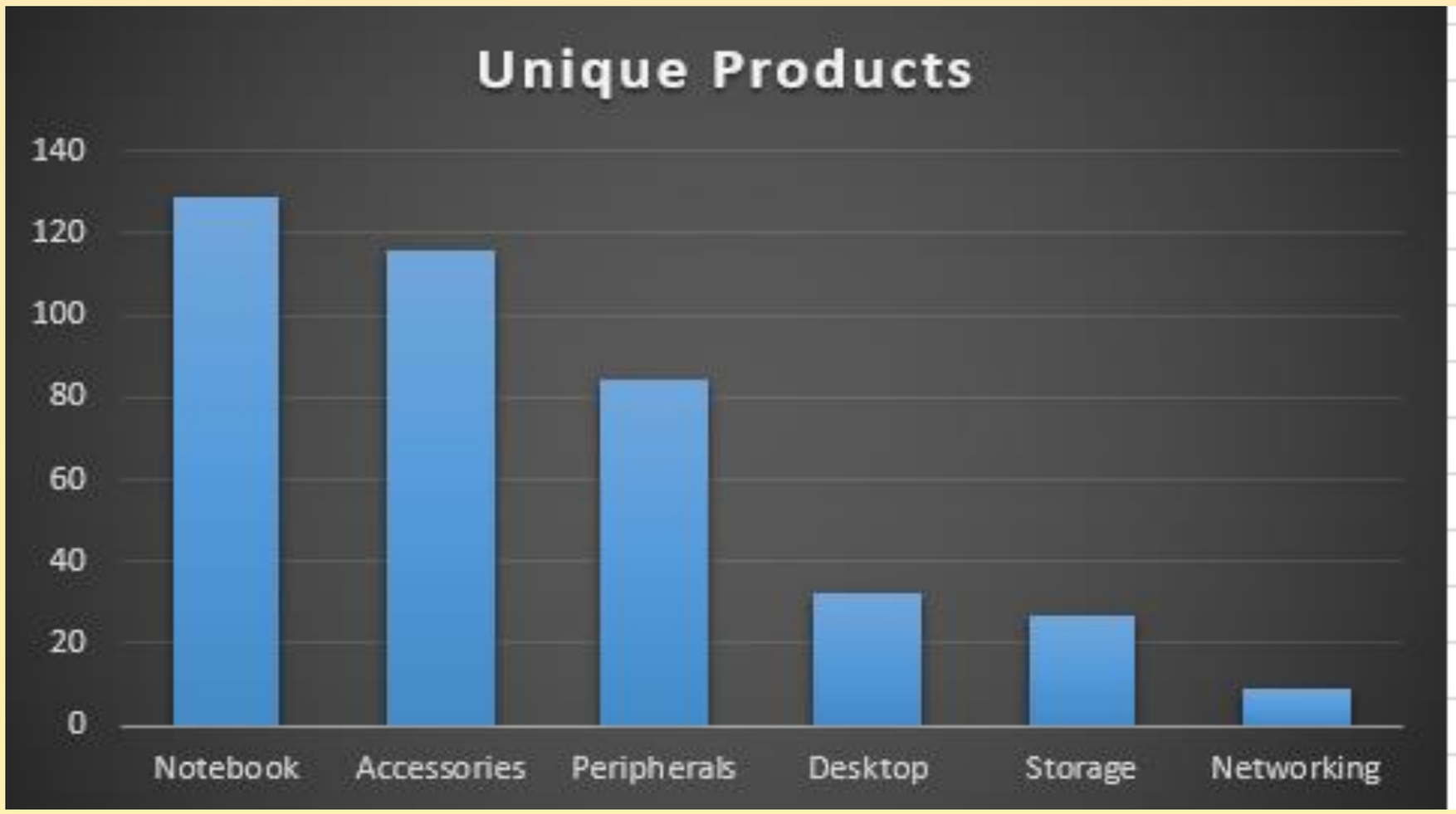
Result

There is 36% increase in unique products from FY 2020 to FY 2021

- **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
select
    segment,
    count(distinct(product_code)) as product_count
from gdb023.dim_product
group by segment
order by product_count desc;
```

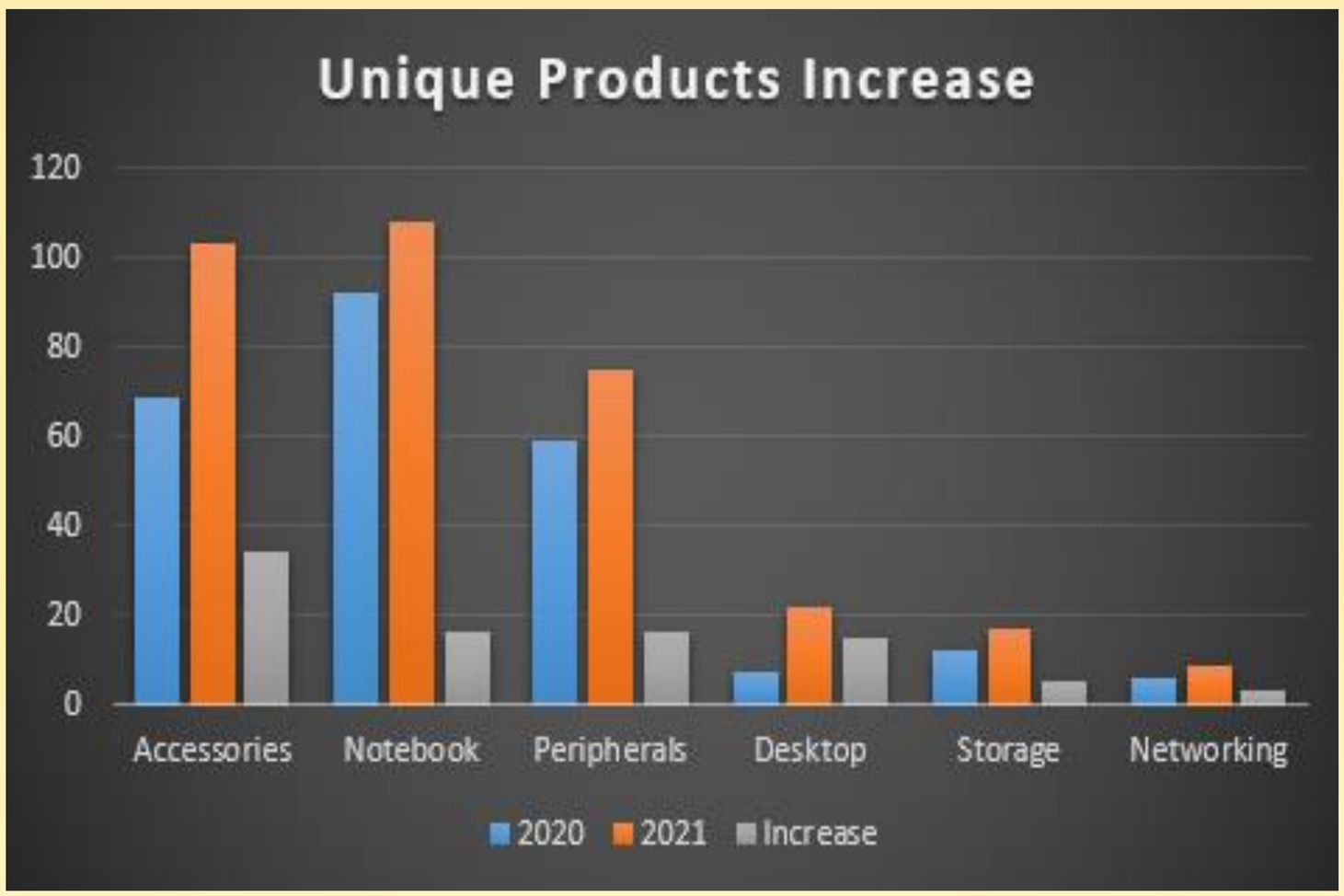
Result



Notebook segment has produced maximum unique products while Networking has the least

- **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 cte as (
    select b.segment,
        sum(case when a.cost_year = 2021 then 1 else 0 end) as unique_products_2021,
        sum(case when a.cost_year = 2020 then 1 else 0 end) as unique_products_2020
    from gdb023.fact_manufacturing_cost a
    join gdb023.dim_product b
        on a.product_code = b.product_code
    group by b.segment
)

select
    segment,
    unique_products_2020,
    unique_products_2021,
    (unique_products_2021 - unique_products_2020) as difference
from cte;
```

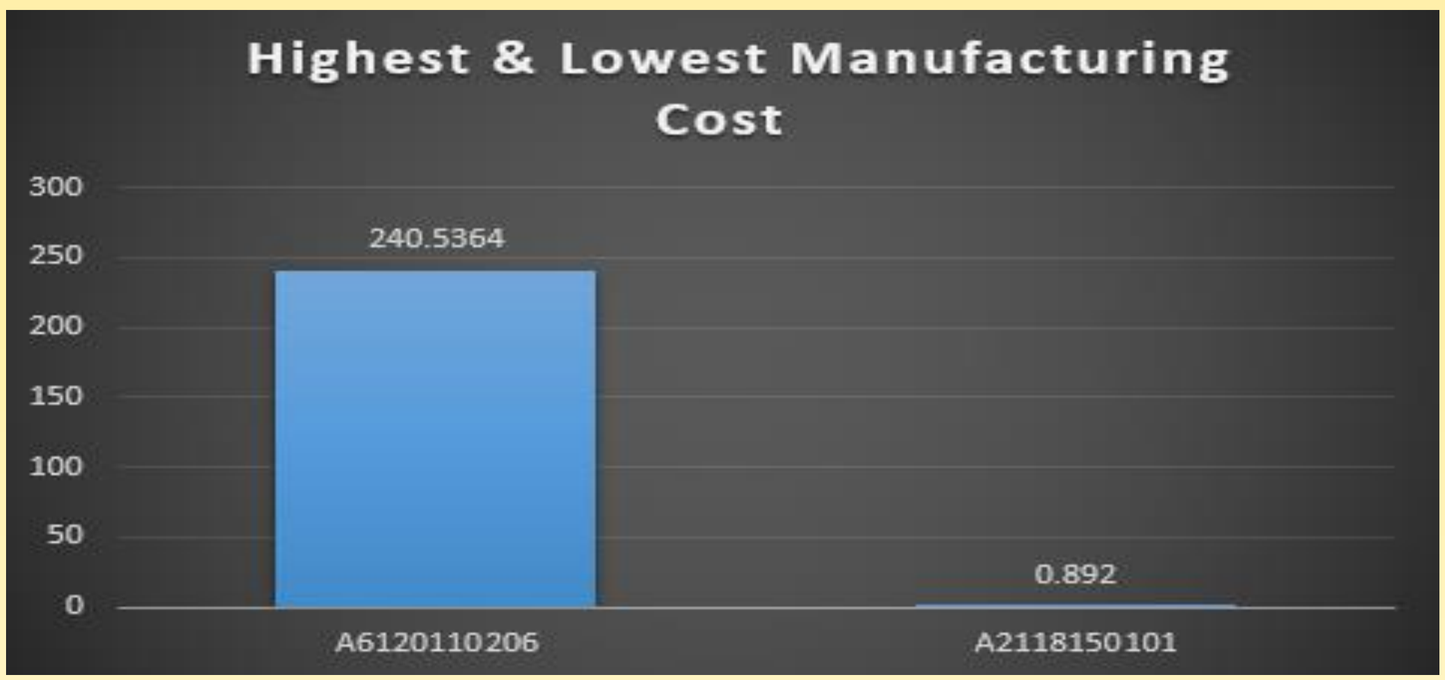


Result

- Company produced more unique products in every segment in 2021 vs 2020
- Accessories segment has most increase in unique products in 2021 vs 2020

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

```
(select
      a.product_code,
      a.product,
      b.manufacturing_cost
from gdb023.dim_product a
right join gdb023.fact_manufacturing_cost b
      on a.product_code = b.product_code
order by b.manufacturing_cost desc
limit 1)
union
(select
      a.product_code,
      a.product,
      b.manufacturing_cost
from gdb023.dim_product a
right join gdb023.fact_manufacturing_cost b
      on a.product_code = b.product_code
order by b.manufacturing_cost asc
limit 1);
```



Result

AQ Home Alin 1 Gen 2 – Highest
AQ Master wired x1 Ms - Lowest

- **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.**

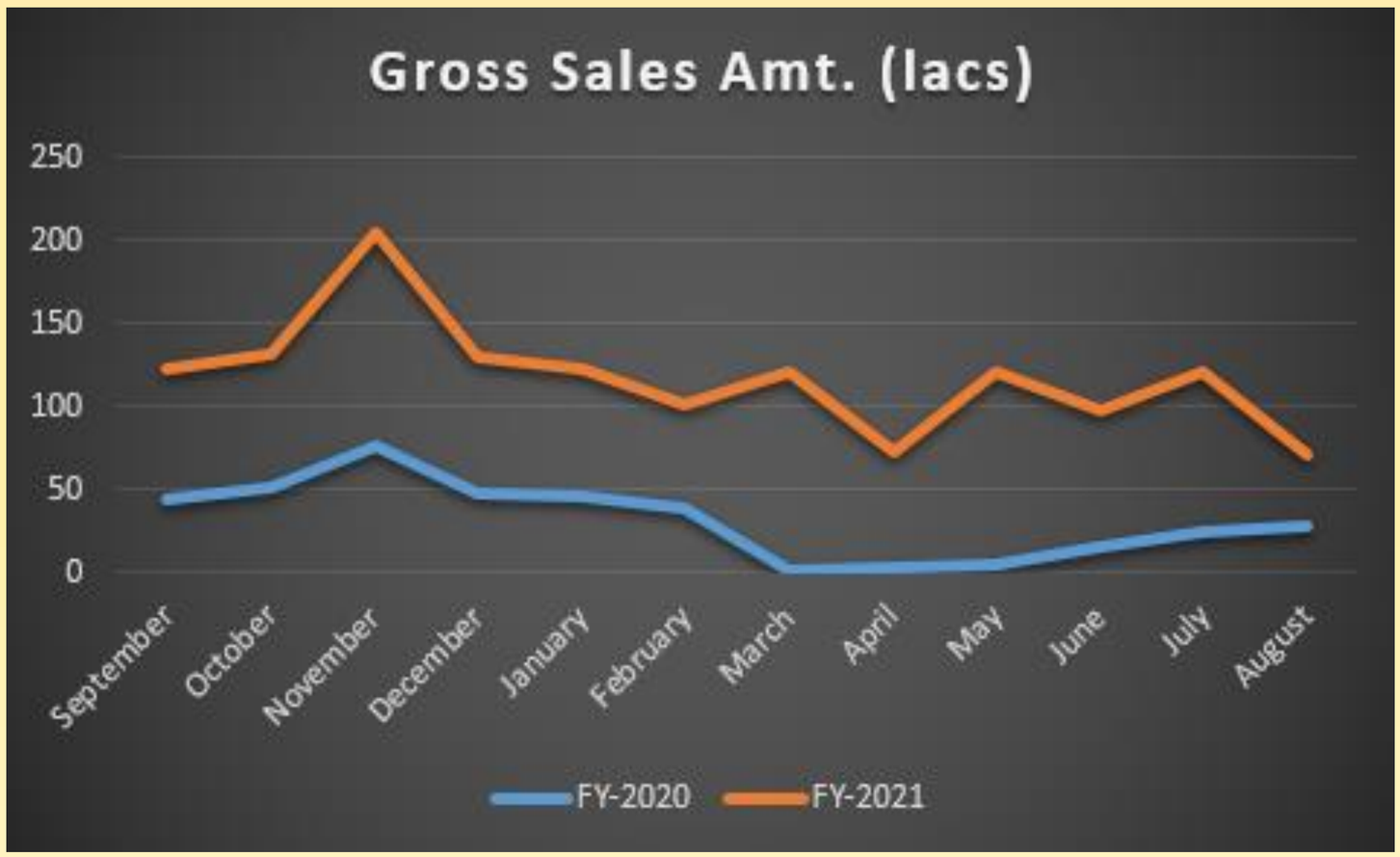
```sql
with cte as (
select
a.customer_code,
c.customer,
round(sum(b.pre_invoice_discount_pct*d.gross_price*a.sold_quantity)
        /sum(d.gross_price*a.sold_quantity)*100,2) as disc_pct
from gdb023.fact_sales_monthly a
join gdb023.fact_pre_invoice_deductions b
    on a.customer_code = b.customer_code
    and a.fiscal_year = b.fiscal_year
join gdb023.dim_customer c
    on a.customer_code = c.customer_code
join gdb023.fact_gross_price d
    on d.product_code = a.product_code
    and d.fiscal_year = a.fiscal_year
where a.fiscal_year = 2021 and c.market = 'India'
group by a.customer_code, c.customer
order by disc_pct desc
)

select
customer_code,
customer,
concat(disc_pct,'%') as discount_pct
from cte
limit 5;
```

Result

Flipkart, Viveks, Ezon, Chroma, Amazon were top 5 customers to get the highest pre_invoice_discount_pct

- **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
select
    month(c.date) as Month,
    c.fiscal_year as FY,
    round((c.sold_quantity)*(b.gross_price),2)
        as Gross_sales_Amount
from gdb023.fact_sales_monthly c
join gdb023.dim_customer a
    on a.customer_code = c.customer_code
join gdb023.fact_gross_price b
    on b.product_code = c.product_code
    and b.fiscal_year = c.fiscal_year
where
    a.customer = 'Atliq Exclusive'
group by Month, FY
order by FY;
```

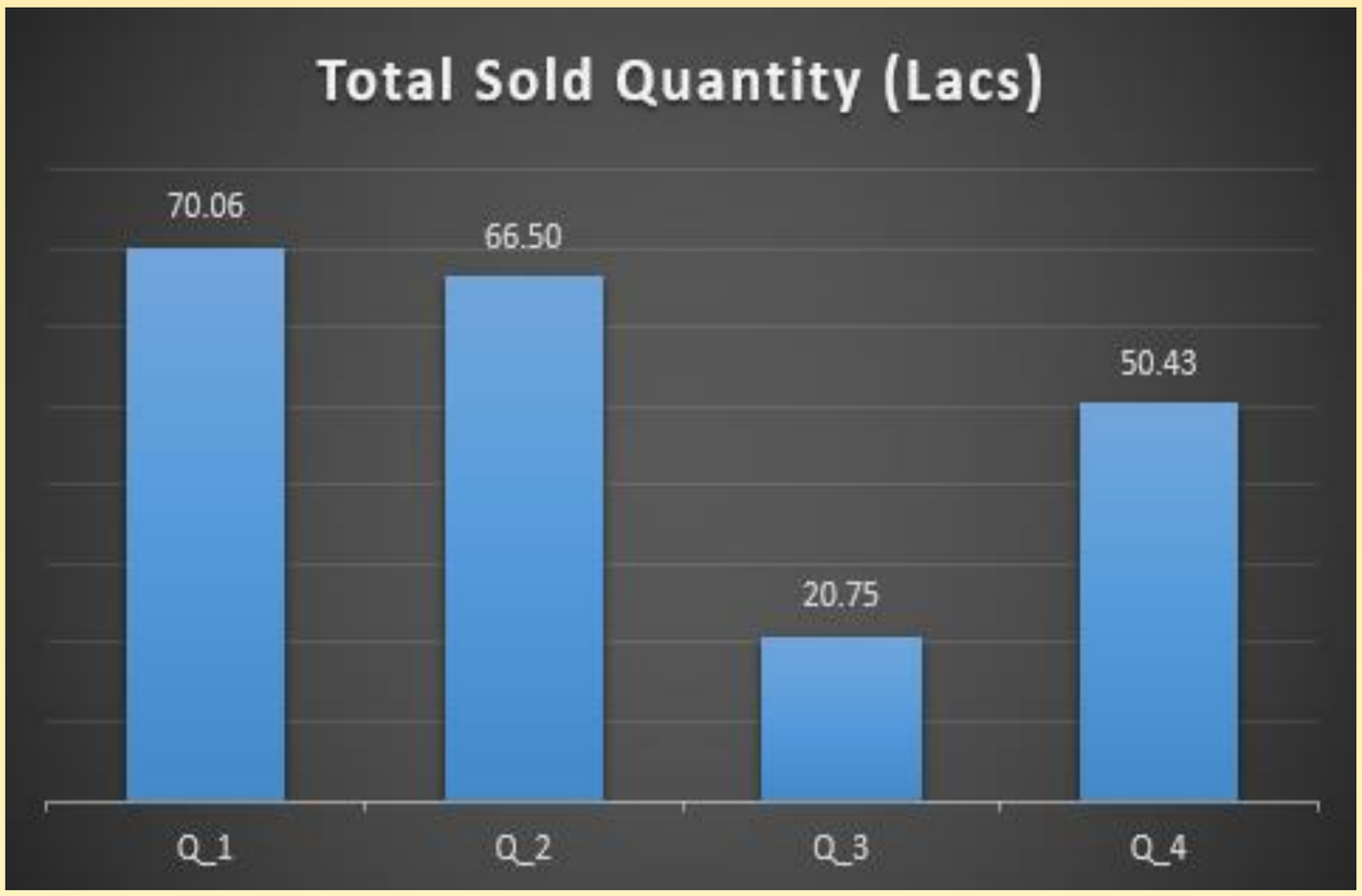


Result

Each year November had the highest Gross Sales.

- **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
with cte as(
    select
        month(date) as Month,
        sold_quantity,
    case
        when month(date) > 8 and month(date) < 12 then 'Q_1'
        when month(date) > 11 or month(date) < 3 then 'Q_2'
        when month(date) > 2 and month(date) < 6 then 'Q_3'
        else 'Q_4'
    end as Quarter
    from gdb023.fact_sales_monthly
    where
        fiscal_year = 2020)

select
    Quarter,
    sum(sold_quantity) as total_sold_quantity
from cte
group by Quarter
order by total_sold_quantity desc;
```



Result

- Q_1 had the highest sales quantity
- Q_3 had the lowest sales quantity

- **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
        a.channel,
        round(sum((b.gross_price)*(c.sold_quantity))/1000000,2) as gross_sales_mln
    from gdb023.fact_sales_monthly c
    join gdb023.dim_customer a
        on a.customer_code = c.customer_code
    join gdb023.fact_gross_price b
        on b.product_code = c.product_code
        and b.fiscal_year = c.fiscal_year
    where
        c.fiscal_year = 2021
    group by a.channel
)

select
    channel,
    gross_sales_mln,
    concat(round(((gross_sales_mln)/(select sum(gross_sales_mln) from cte))*100,2),'%') as percentage
from cte
group by channel
order by gross_sales_mln desc;
```

Result

- Distributer did highest sales
- Retailer did lowest sales

- **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
with cte as (
    select
        a.division,
        b.product_code,
        a.product,
        sum(b.sold_quantity) as total_sold_quantity,
        rank() over (
                partition by division
                order by sum(b.sold_quantity) desc) rank_order
    from gdb023.fact_sales_monthly b
    join gdb023.dim_product a
        on a.product_code = b.product_code
    where
        b.fiscal_year = 2021
    group by a.division, b.product_code, a.product
)

select
    division,
    product_code,
    product,
    total_sold_quantity,
    rank_order
from cte
```