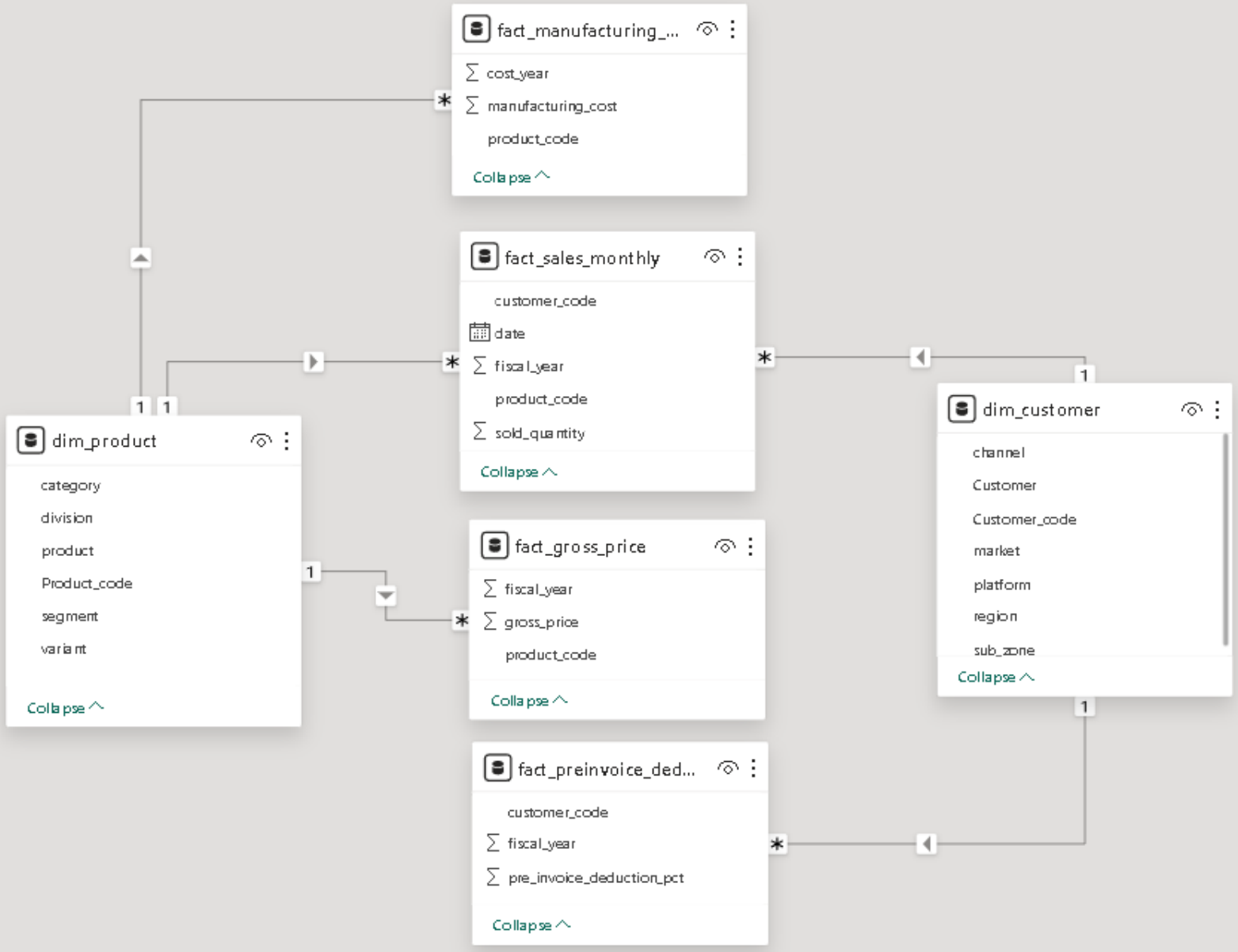

AD-HOC
ANALYSIS

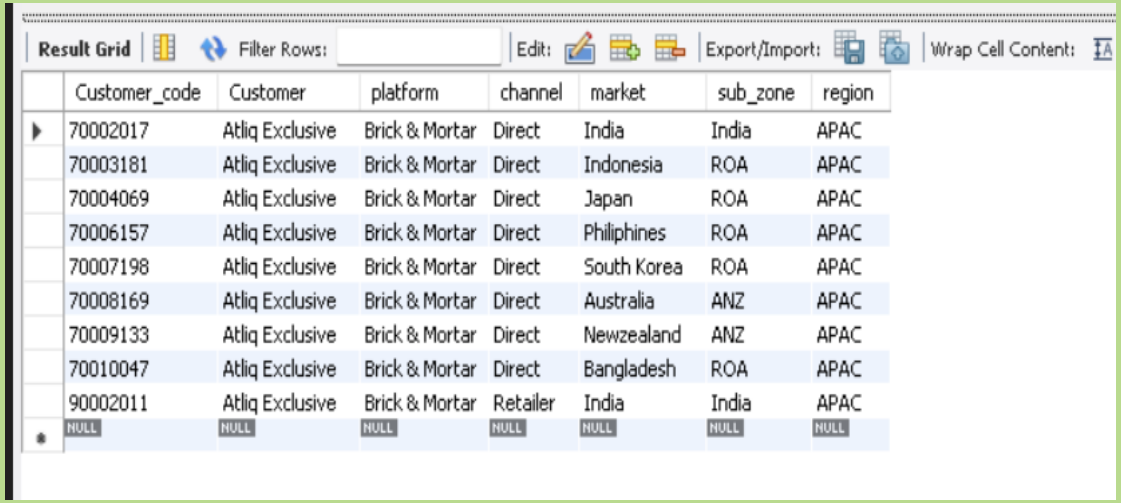
# AGENDA

- Develop a standardized SQL framework for addressing ad hoc requests efficiently
- Decision-Making: SQL Solutions for Instant Insights
- Engaging Visualizations: Creative Presentations for Top-Level Clarity


fact_manufacturing_...
cost_year
manufacturing_cost
product_code
Collapse
fact_sales_monthly
customer_code
date
fiscal_year
product_code
sold_quantity
Collapse
dim_product
category
division
product
Product_code
segment
variant
Collapse
fact_gross_price
fiscal_year
gross_price
product_code
Collapse
dim_customer
channel
Customer
Customer_code
market
platform
region
sub_zone
Collapse
fact_preinvoice_ded...
customer_code
fiscal_year
pre_invoice_deduction_pct
Collapse
1
*

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

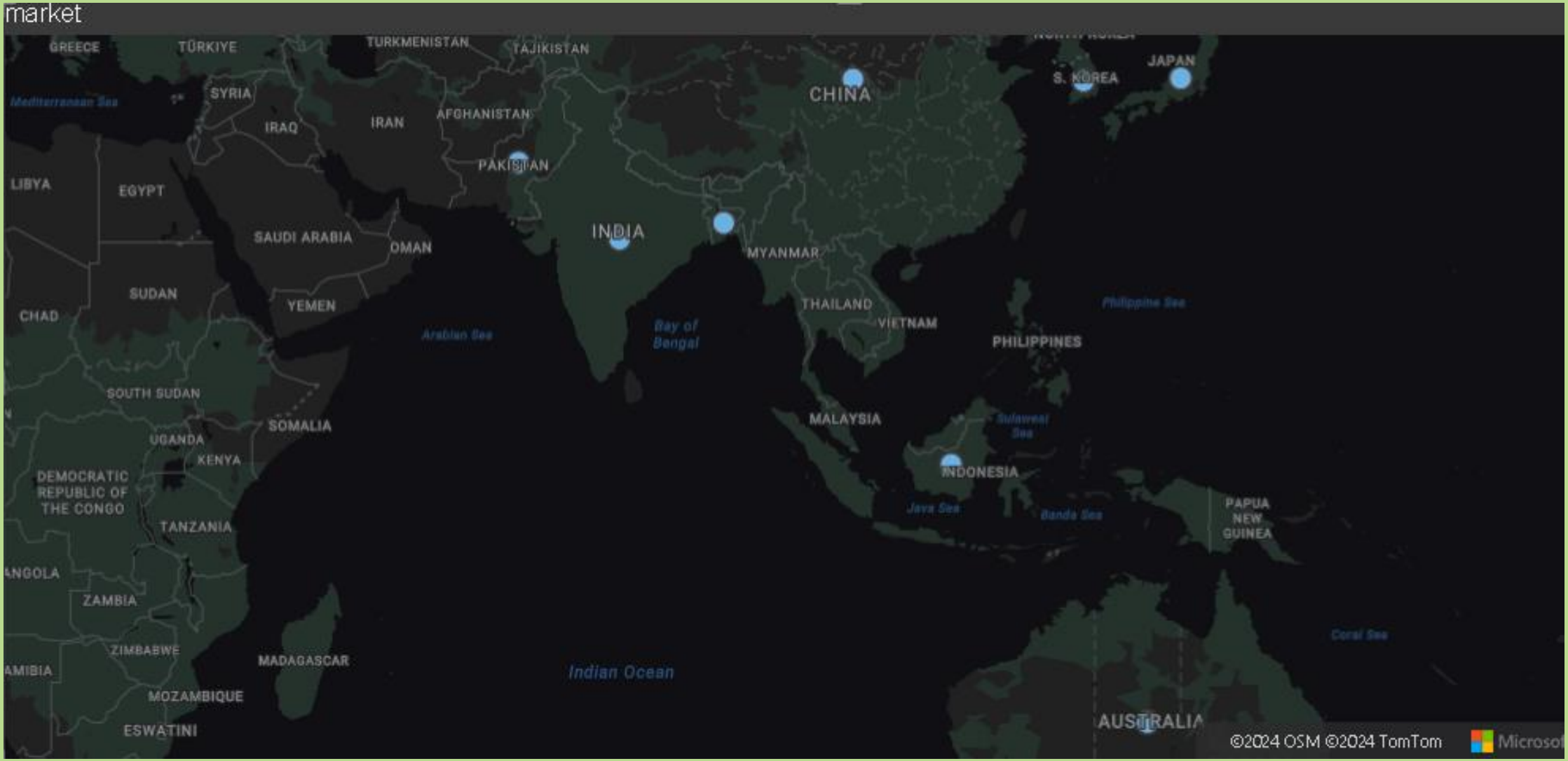
```
1   select *
2    from dim_customer
3    where Customer="Atliq Exclusive" and region="APAC";
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

| Customer_code | Customer | platform | channel | market | sub_zone | region |
|---|---|---|---|---|---|---|
| 70002017 | Atliq Exclusive | Brick & Mortar | Direct | India | India | APAC |
| 70003181 | Atliq Exclusive | Brick & Mortar | Direct | Indonesia | ROA | APAC |
| 70004069 | Atliq Exclusive | Brick & Mortar | Direct | Japan | ROA | APAC |
| 70006157 | Atliq Exclusive | Brick & Mortar | Direct | Philiphines | ROA | APAC |
| 70007198 | Atliq Exclusive | Brick & Mortar | Direct | South Korea | ROA | APAC |
| 70008169 | Atliq Exclusive | Brick & Mortar | Direct | Australia | ANZ | APAC |
| 70009133 | Atliq Exclusive | Brick & Mortar | Direct | Newzealand | ANZ | APAC |
| 70010047 | Atliq Exclusive | Brick & Mortar | Direct | Bangladesh | ROA | APAC |
| 90002011 | Atliq Exclusive | Brick & Mortar | Retailer | India | India | APAC |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

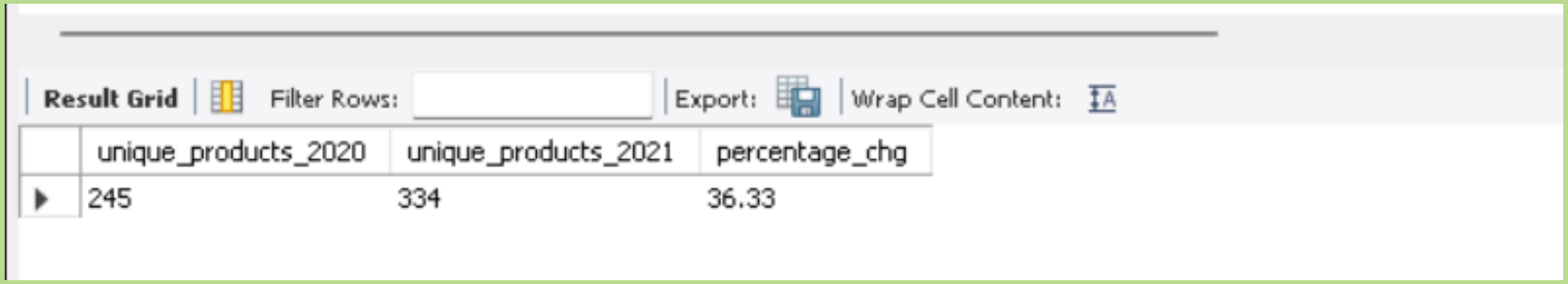
market


GREECE
TÜRKIYE
TURKMENISTAN
TAJIKISTAN
CHINA
S. KOREA
JAPAN
Mediterranean Sea
SYRIA
IRAQ
IRAN
AFGHANISTAN
PAKISTAN
LIBYA
EGYPT
SAUDI ARABIA
OMAN
INDIA
MYANMAR
SUDAN
CHAD
YEMEN
Arabian Sea
Bay of Bengal
THAILAND
VIETNAM
Philippine Sea
PHILIPPINES
SOUTH SUDAN
SOMALIA
MALAYSIA
Sulawesi Sea
UGANDA
KENYA
INDONESIA
DEMOCRATIC REPUBLIC OF THE CONGO
Java Sea
Banda Sea
PAPUA NEW GUINEA
TANZANIA
ANGOLA
ZAMBIA
Coral Sea
ZIMBABWE
AMIBIA
MADAGASCAR
Indian Ocean
MOZAMBIQUE
ESWATINI
AUSTRALIA
©2024 OSM ©2024 TomTom
Microsof

# 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
With product as (SELECT p.*,f.fiscal_year
FROM dim_product p
join fact_gross_price f
on p.Product_code=f.product_code)

SELECT
    count(CASE WHEN fiscal_year  = 2020 THEN product END) as unique_products_2020,
    count(CASE WHEN fiscal_year  = 2021 THEN product END) as unique_products_2021,
    round(((count(CASE WHEN fiscal_year = 2021 THEN product END) - count(CASE WHEN fiscal_year  = 2020 THEN product END))
    / count(CASE WHEN fiscal_year  = 2020 THEN product END)) * 100,2) as percentage_chg
FROM product
WHERE fiscal_year IN (2020, 2021);
```

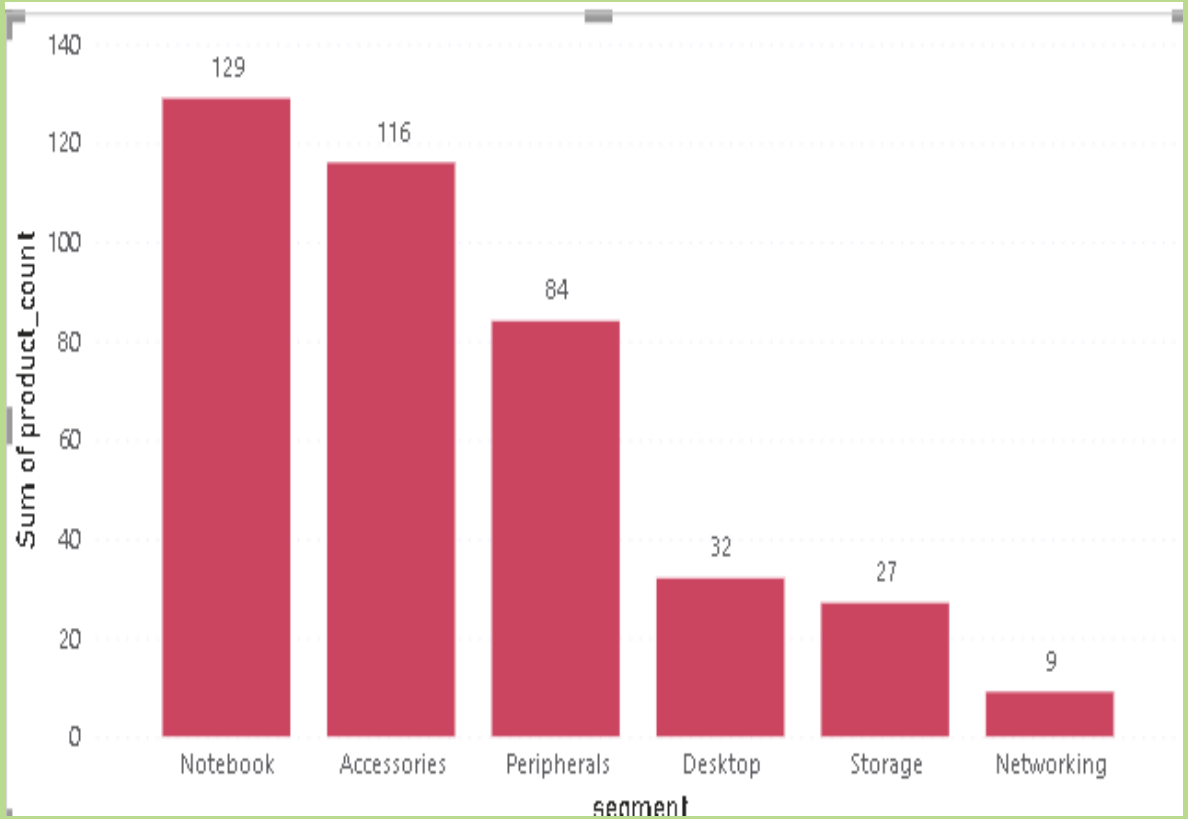



# 3.Provide a report with all the unique product counts for each segment and sort them in descending order of product counts. The final output contains 2 fields, segment product_count

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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



# 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
with product as ( SELECT p.*,f.fiscal_year
FROM dim_product p
join fact_gross_price f
on p.Product_code=f.product_code
)
select distinct segment,
    (select  count( product) from product where fiscal_year="2021") as product_count_2021,
    (select  count(product) from product where fiscal_year="2020") as product_count_2020,
     ((select  count(product) from product where fiscal_year="2021") -(select  count(product) from product where fiscal_year="2020"))
        as Difference
     from product;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| segment | product_count_2021 | product_count_2020 | Difference |
|---|---|---|---|
| Peripherals | 334 | 245 | 89 |
| Accessories | 334 | 245 | 89 |
| Notebook | 334 | 245 | 89 |
| Desktop | 334 | 245 | 89 |
| Storage | 334 | 245 | 89 |
| Networking | 334 | 245 | 89 |

# 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
with cte1 as ( SELECT
    m.Product_code,
    concat('$',round(m.manufacturing_cost,2)) as manufacturing_cost,
    p.product
FROM fact_manufacturing_cost m
right join dim_product p
on m.product_code=p.product_code )
select Product_code,
      product,manufacturing_cost from cte1
where manufacturing_cost = (select max(manufacturing_cost) from cte1) or
manufacturing_cost= (select min(manufacturing_cost) from cte1)
order by  manufacturing_cost desc;
```

- **Highest Manufacturing_cost:- AQ Genx[99.54$]**
- **Lowest Manufacturing_cost:- AQ Master Weird [0.89$]**

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

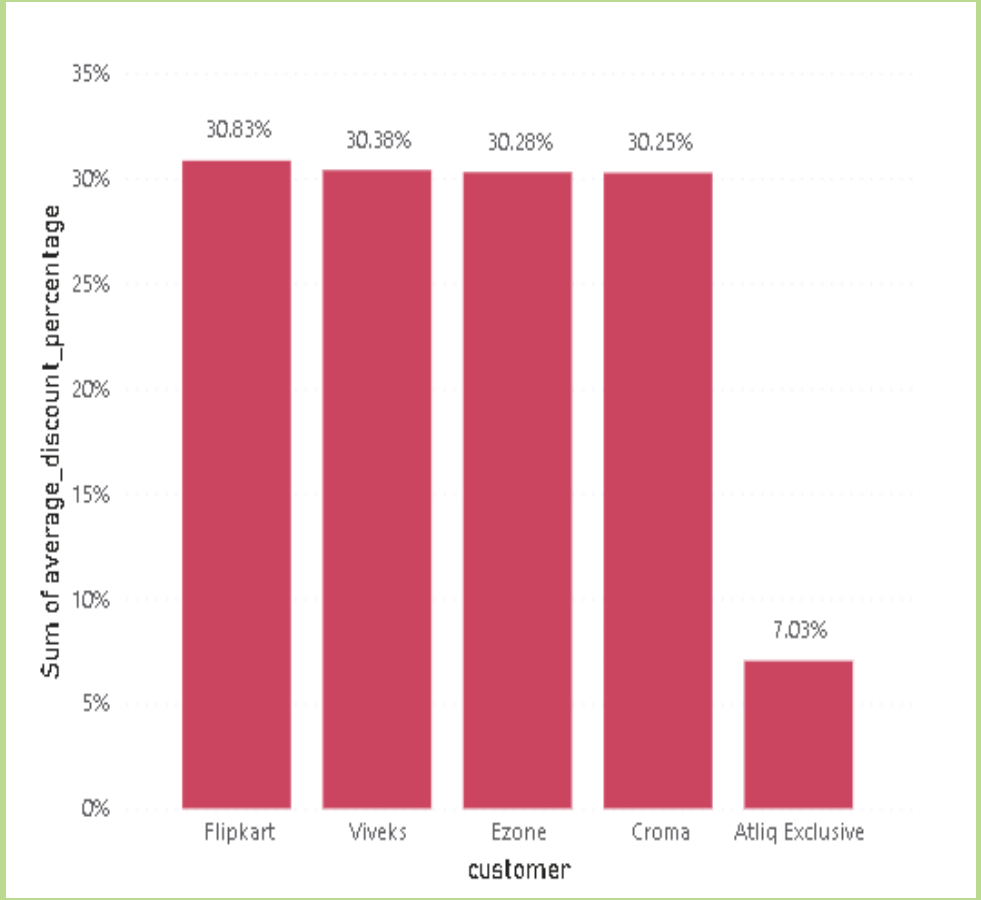
| Product_code | product | manufacturing_cost |
|---|---|---|
| A4520110504 | AQ Gen X | $99.54 |
| A2118150101 | AQ Master wired x1 Ms | $0.89 |

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
SELECT a.customer_code ,
       b.customer,
       CONCAT(ROUND(AVG(pre_invoice_deduction_pct)*100,2),'%') AS average_discount_percentage
FROM fact_preinvoice_deduction AS a
INNER JOIN
dim_customer AS b
ON a.customer_code = b.customer_code
WHERE market = 'India'
AND fiscal_year = 2021
GROUP BY b.customer, b.customer_code
ORDER BY average_discount_percentage DESC
LIMIT 5;
select * from fact_preinvoice_deduction
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| customer_code | customer | average_discount_percentage |
|---|---|---|
| 70002017 | Atliq Exclusive | 7.03% |
| 90002009 | Flipkart | 30.83% |
| 90002006 | Viveks | 30.38% |
| 90002003 | Ezone | 30.28% |
| 90002002 | Croma | 30.25% |



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
with cte as(SELECT f.product_code,
      monthname(f.date) as Months,month(f.date) as month_number,
      year(f.date) as Year,
      (f.sold_quantity*g.gross_price) as Gross_sales
FROM fact_sales_monthly f
join fact_gross_price g
on f.product_code=g.product_code
join dim_customer c
on f.Customer_code=c.Customer_code
where Customer='Atliq Exclusive')
select months,year,
      concat(round(sum(Gross_sales)/1000000,2),'M') as gross_sales
 from cte
 group by year,months
 order by year,months desc
;
```

Result Grid | Filter Rows:

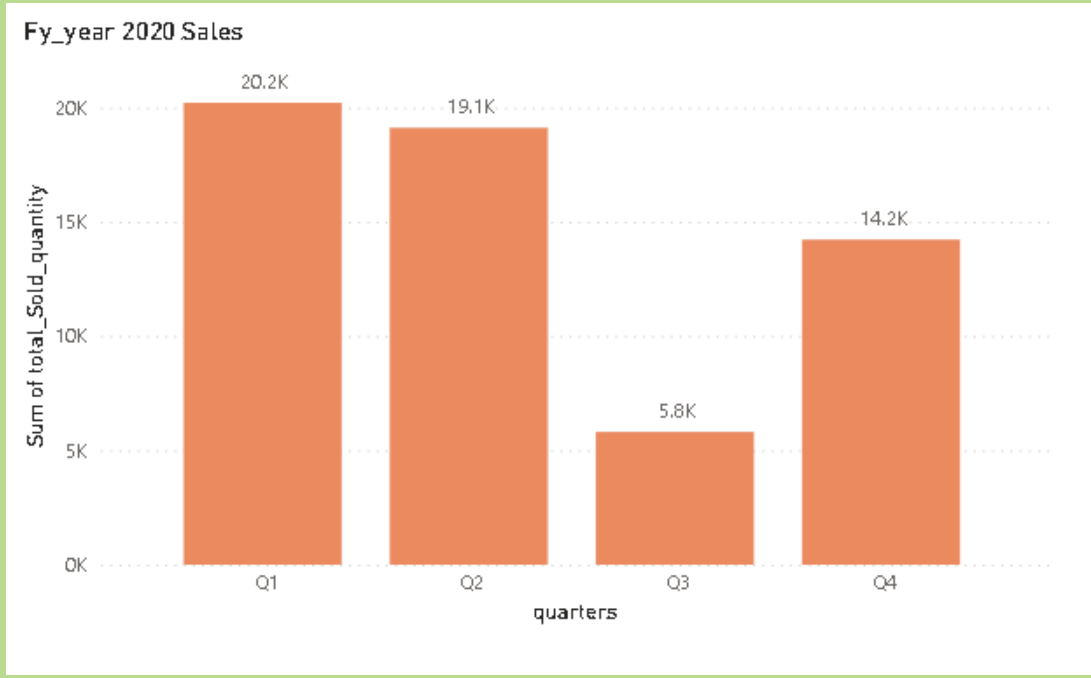
| Months | Year | gross_sales |
|---|---|---|
| September | 2019 | 0.27M |
| October | 2019 | 0.29M |
| November | 2019 | 0.46M |
| December | 2019 | 0.28M |
| September | 2020 | 0.46M |
| October | 2020 | 0.49M |
| November | 2020 | 0.63M |
| May | 2020 | 0.05M |
| March | 2020 | 0.02M |
| June | 2020 | 0.10M |
| July | 2020 | 0.15M |
| January | 2020 | 0.28M |
| February | 2020 | 0.24M |
| December | 2020 | 0.39M |
| August | 2020 | 0.18M |
| April | 2020 | 0.02M |
| May | 2021 | 0.49M |
| March | 2021 | 0.44M |
| June | 2021 | 0.38M |
| July | 2021 | 0.44M |
| January | 2021 | 0.43M |
| February | 2021 | 0.35M |
| August | 2021 | 0.26M |
| April | 2021 | 0.23M |

# 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
SELECT
    case
    when date between '2019-09-01' and '2019-11-01' then 'Q1'
    when date between '2019-12-01' and '2020-02-01' then 'Q2'
    when date between '2020-03-01' and '2020-05-01' then 'Q3'
    when date between '2020-06-01' and '2020-09-01' then 'Q4'
end as quarters,
   sum(sold_quantity) as total_Sold_quantity
FROM fact_sales_monthly where fiscal_year=2020
group by quarters
order by total_sold_quantity desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

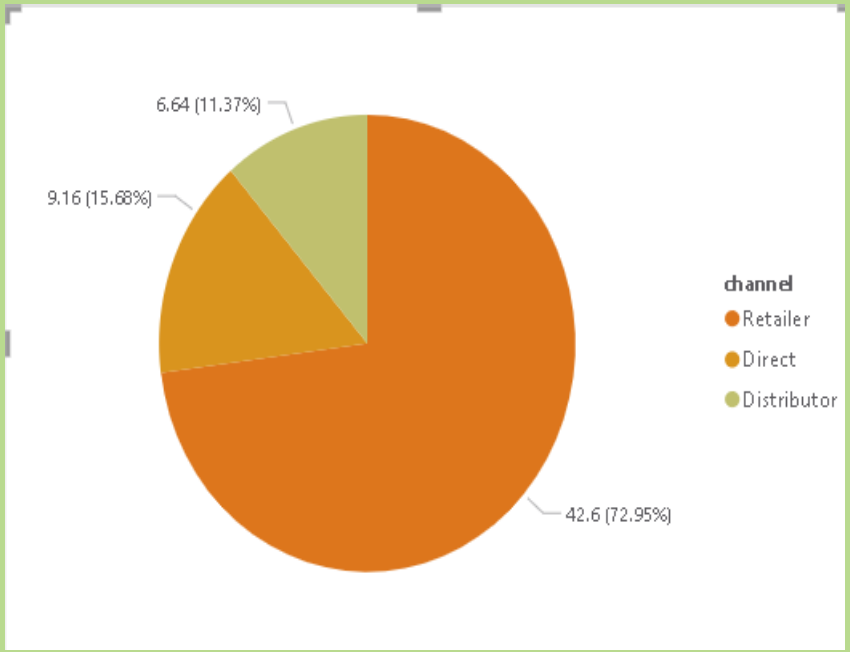
| quarters | total_Sold_quantity |
|---|---|
| Q1 | 20208 |
| Q2 | 19115 |
| Q4 | 14223 |
| Q3 | 5811 |



# 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
with cte as(select c.channel,
      sum((s.sold_quantity*g.gross_price)) as total_gross_sales
from fact_sales_monthly s
left join fact_gross_price g  on s.product_code=g.product_code
right join  dim_customer c  on c.customer_code=s.customer_code
where s.fiscal_year=2021
group by c.channel
order by total_gross_sales desc)

select channel,
       concat(round(total_gross_sales/1000000,2),'M') as Gross_sales,
       concat(round(total_gross_sales/sum(total_gross_sales)over() *100,2),'%') as percantage
from cte;
```




Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| channel | Gross_sales | percantage |
|---|---|---|
| Retailer | 42.60M | 72.95% |
| Direct | 9.16M | 15.69% |
| Distributor | 6.64M | 11.36% |

# 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

```sql
WITH Output1 AS
(
SELECT P.division, FS.product_code, P.product, SUM(FS.sold_quantity) AS Total_sold_quantity
FROM dim_product P JOIN fact_sales_monthly FS
ON P.product_code = FS.product_code
WHERE FS.fiscal_year = 2021
GROUP BY  FS.product_code, division, P.product
),
Output2 AS
(
SELECT division, product_code, product, Total_sold_quantity,
        RANK() OVER(PARTITION BY division ORDER BY Total_sold_quantity DESC) AS 'Rank_Order'
FROM Output1
)
 SELECT Output1.division, Output1.product_code, Output1.product, Output2.Total_sold_quantity, Output2.Rank_Order
 FROM Output1 JOIN Output2
 ON Output1.product_code = Output2.product_code
WHERE Output2.Rank_Order IN (1,2,3)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| division | product_code | product | Total_sold_quantity | Rank_Order |
|---|---|---|---|---|
| PC | A4319110306 | AQ Velocity | 17280 | 1 |
| PC | A4319110304 | AQ Velocity | 17010 | 2 |
| PC | A4419110402 | AQ Elite | 16847 | 3 |