CONSUMER AD-HOC INSIGHTS

PRESENTED BY: PRAVESH AGARWAL

Company Overview

AtliQ Hardware is a **Computer Hardware** and **Accessory** manufacterer.

The company manufactures products under 3 major divisions i.e., **Netorking & Storage**, **PC**, **Peripherals & Accessories**.

AtliQ Hardware is operational in **NA**, **LATAM**, **EU** and **APAC** regions.

Objectives

- **AtliQ Hardware** (fictitious corporation) is one of the major computer hardware manufacturers in India, with a strong presence in other nations.
- Nevertheless, the management did note that they do not have sufficient insights to make prompt, wise, and data-informed judgments.
- Plan to expand the data analytics team by adding junior data analysts.
- To assess candidates, Data analytics director, Tony Sharma plans to conduct a **SQL Challenge** to evaluate both tech and soft skills.
- Thecompanyseeksinsights for 10 ad hoc requests.

Company’s Markets

AtliQ Hardware operates across four major regions: **North America (NA)**, **Latin America (LATAM)**, **Europe (EU)**, and **Asia-Pacific (APAC)**. This global presence allows AtliQ to serve diverse customer needs with products in Networking and Storage, PCs, Peripherals and Accessories.

In **NA** and **EU**, AtliQ benefits from strong demand for **Advanced Computing Solutions**. **LATAM** and **APAC**, with their growing economies, offer **Significant Opportunities for Expansion**. This strategic positioning ensures AtliQ's sustained growth and customer satisfaction worldwide.

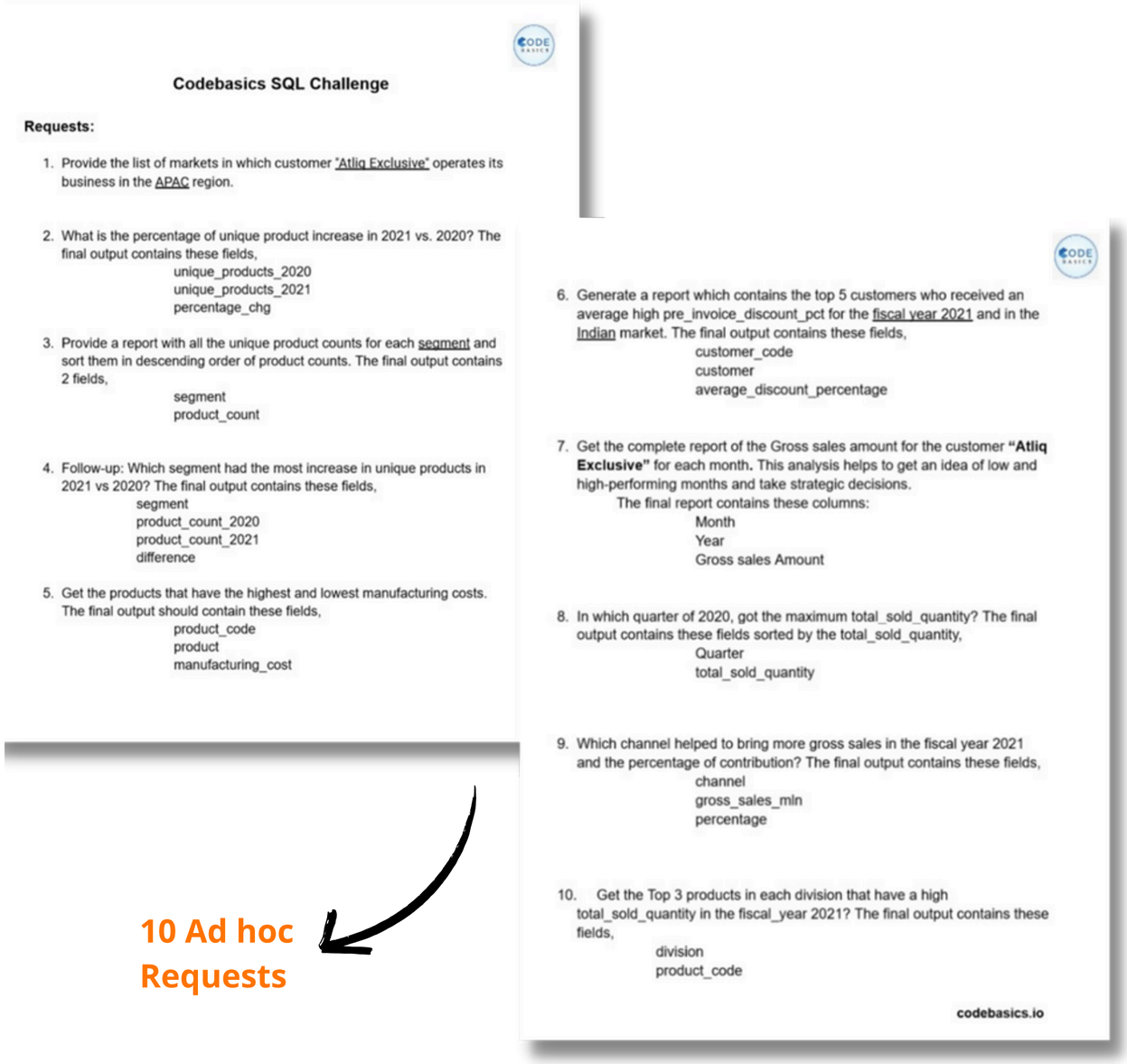
Requests & Tools

Power BI

For Analysis and Visualization

MySQL

For Ad hoc Queries



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

codebasics.io

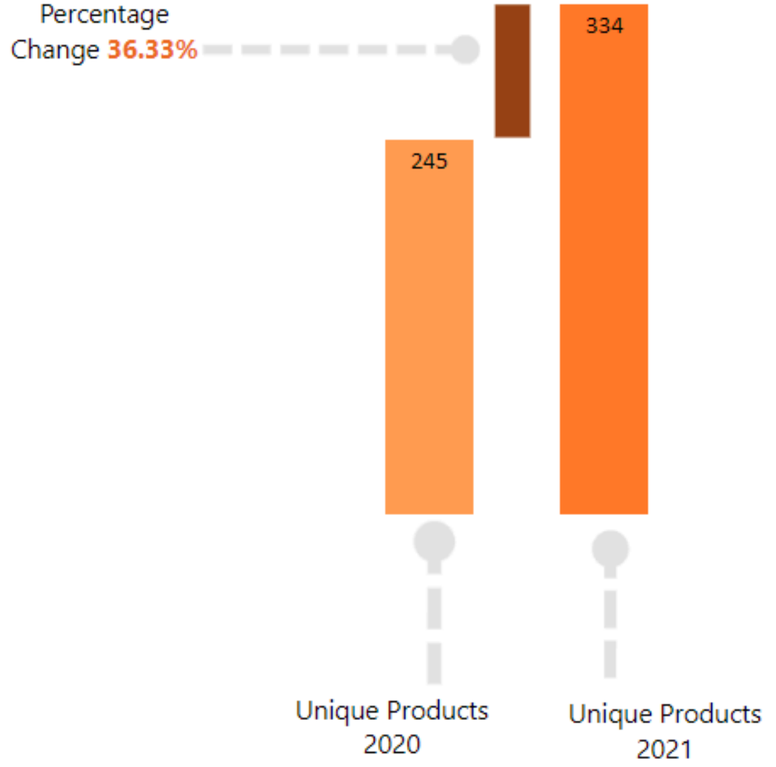
10 Ad hoc Requests

Q1. Provide the list of markets in which customer "Atliq Exclusive" operates its business in the APAC region.

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

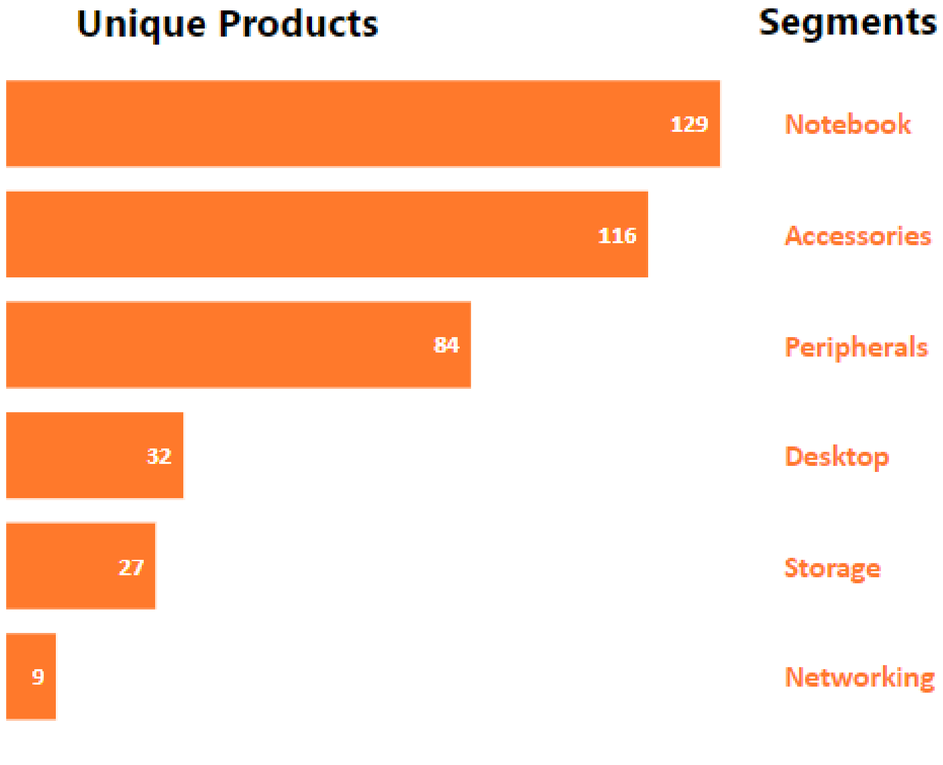
Q2. What is the percentage of unique product increase in 2021 vs. 2020? The final output contains these fields: unique_products_2020, unique_products_2021 and percentage_chg

```sql
WITH
    unique_products_2020 AS (
        SELECT
            COUNT( DISTINCT product_code ) AS unique_products_2020
        FROM
            fact_sales_monthly
        WHERE
            fiscal_year = 2020
    ),
    unique_products_2021 AS (
        SELECT
            COUNT( DISTINCT product_code ) AS unique_products_2021
        FROM
            fact_sales_monthly
        WHERE
            fiscal_year = 2021
    )
SELECT
    up20.unique_products_2020,
    up21.unique_products_2021,
    ROUND( ( (unique_products_2021 - unique_products_2020) * 100 ) / unique_products_2020, 2 ) AS percentage_chg
FROM
    unique_products_2020 up20,
    unique_products_2021 up21;
```



Q3. 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 and product_count

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

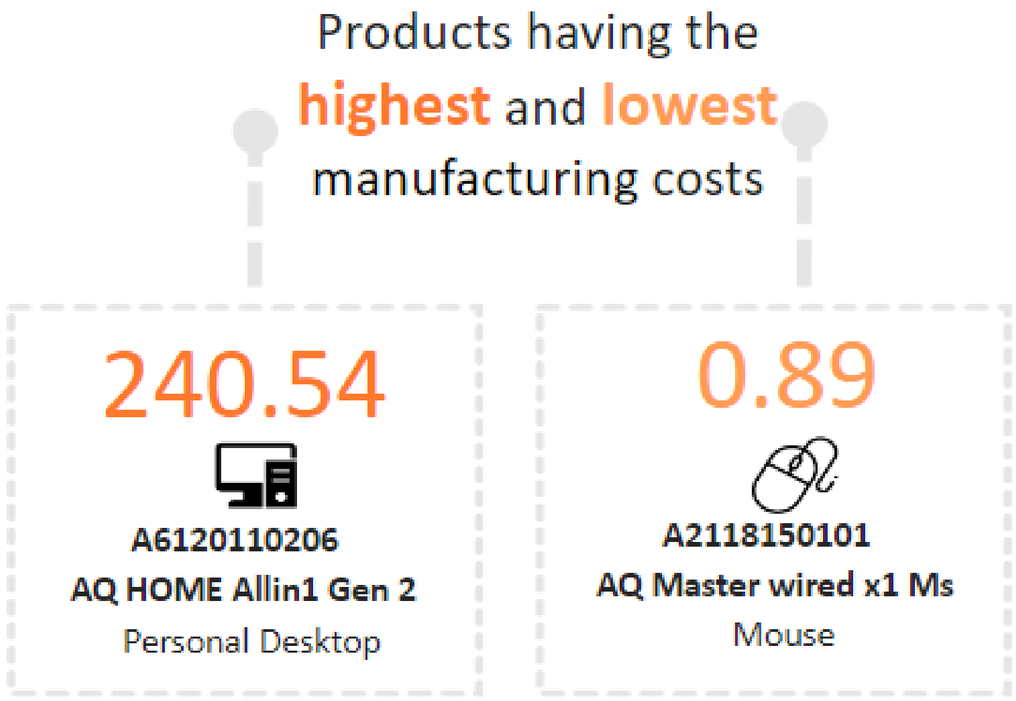


Q4. Follow-up: Which segment had the most increase in unique products in 2021 vs 2020? The final output contains these fields:
segment
product_count_2020
product_count_2021
difference

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

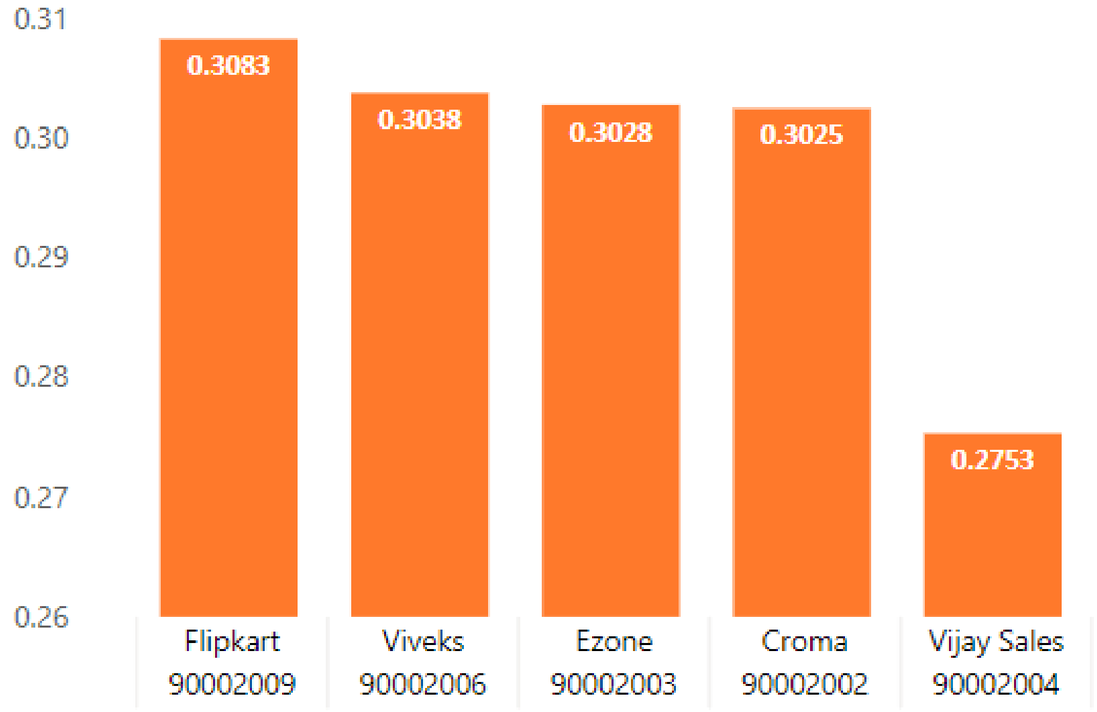
```sql
WITH
    count_2020 AS(
        SELECT
            dp.segment,
            COUNT( DISTINCT fsm.product_code ) AS product_count_2020
        FROM
            dim_product dp
        JOIN
            fact_sales_monthly fsm
                ON
                    dp.product_code = fsm.product_code
        WHERE
            fsm.fiscal_year = 2020
        GROUP BY
            dp.segment
    ),
    count_2021 AS(
        SELECT
            dp.segment,
            COUNT( DISTINCT fsm.product_code ) AS product_count_2021
        FROM
            dim_product dp
        JOIN
            fact_sales_monthly fsm
                ON
                    dp.product_code = fsm.product_code
        WHERE
            fsm.fiscal_year = 2021
        GROUP BY
            dp.segment
    )
SELECT
    c20.segment, c20.product_count_2020,
    c21.product_count_2021,
    ( c21.product_count_2021 - c20.product_count_2020 ) AS difference
FROM
    count_2020 c20
JOIN
    count_2021 c21
        ON
            c20.segment = c21.segment
ORDER BY
    difference DESC;
```

Q5. Get the products that have the highest and lowest manufacturing costs. The final output should contain these fields:
product_code
product
manufacturing_cost



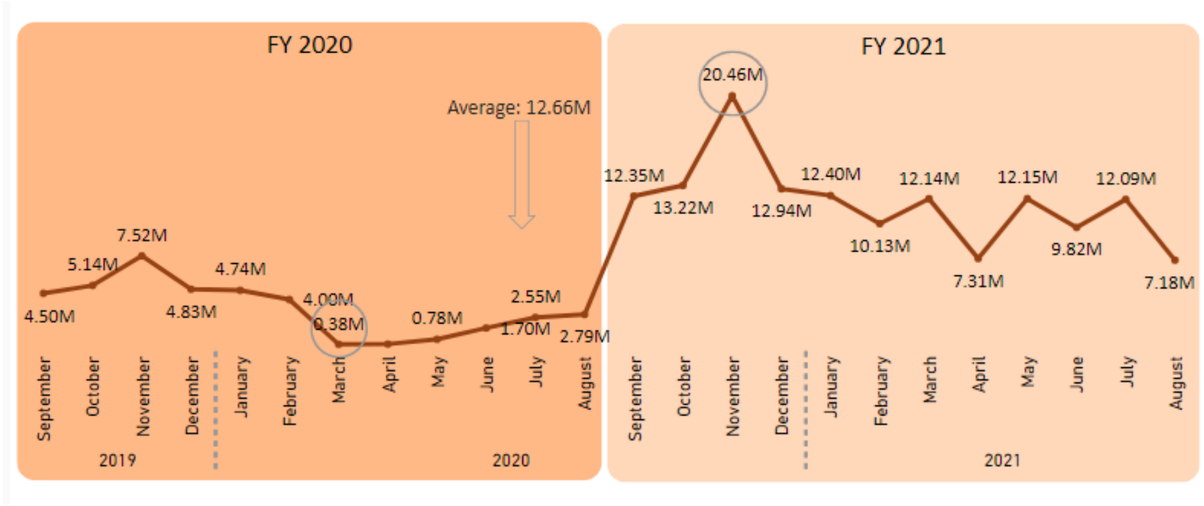

```sql
SELECT
    fmc.product_code,
    dp.product,
    fmc.manufacturing_cost
FROM
    fact_manufacturing_cost fmc
JOIN
    dim_product dp
        ON
            dp.product_code = fmc.product_code
WHERE
    fmc.manufacturing_cost IN
        (
            SELECT
                MAX(manufacturing_cost)
            FROM
                fact_manufacturing_cost

            UNION

            SELECT
                MIN(manufacturing_cost)
            FROM
                fact_manufacturing_cost
        )
ORDER BY
    fmc.manufacturing_cost DESC;
```

Q6. 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 and average_discount_percentage




```sql
WITH
    pid21 AS(
        SELECT
            customer_code, pre_invoice_discount_pct
        FROM
            fact_pre_invoice_deductions
        WHERE
            fiscal_year = 2021
    ),
    market AS(
        SELECT
            customer, customer_code
        FROM
            dim_customer
        WHERE
            market = 'India'
    )
SELECT
    dis.customer_code,
    m.customer,
    AVG(dis.pre_invoice_discount_pct) AS average_discount_percentage
FROM
    pid21 dis
JOIN
    market m
        ON
            m.customer_code = dis.customer_code
GROUP BY
    m.customer
ORDER BY
    pre_invoice_discount_pct DESC
LIMIT
    5;
```

Q7. 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 and Gross sales Amount



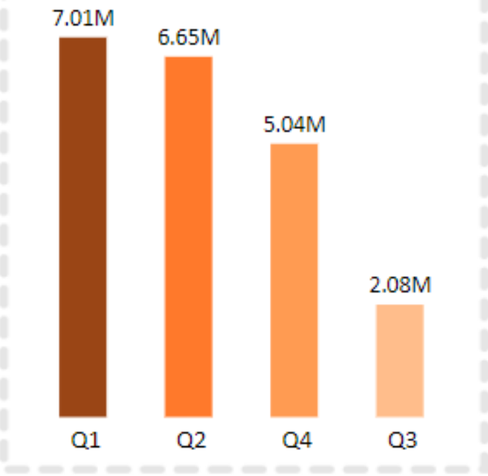

```sql
SELECT
    DATE_FORMAT( fsm.date, '%M (%Y)' ) AS Month,
    fsm.fiscal_year AS Fiscal_Year,
    ROUND( SUM( (fsm.sold_quantity * fgp.gross_price) ), 2 ) AS Gross_Sales_Amount
FROM
    fact_sales_monthly fsm
JOIN
    dim_customer dc
        ON
            dc.customer_code = fsm.customer_code
JOIN
    fact_gross_price fgp
        ON
            fgp.product_code = fsm.product_code
            AND
            fgp.fiscal_year = fsm.fiscal_year
WHERE
    dc.customer = 'Atliq Exclusive'
GROUP BY
    Month,
    Fiscal_Year
ORDER BY
    Fiscal_Year;
```

Q8. 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

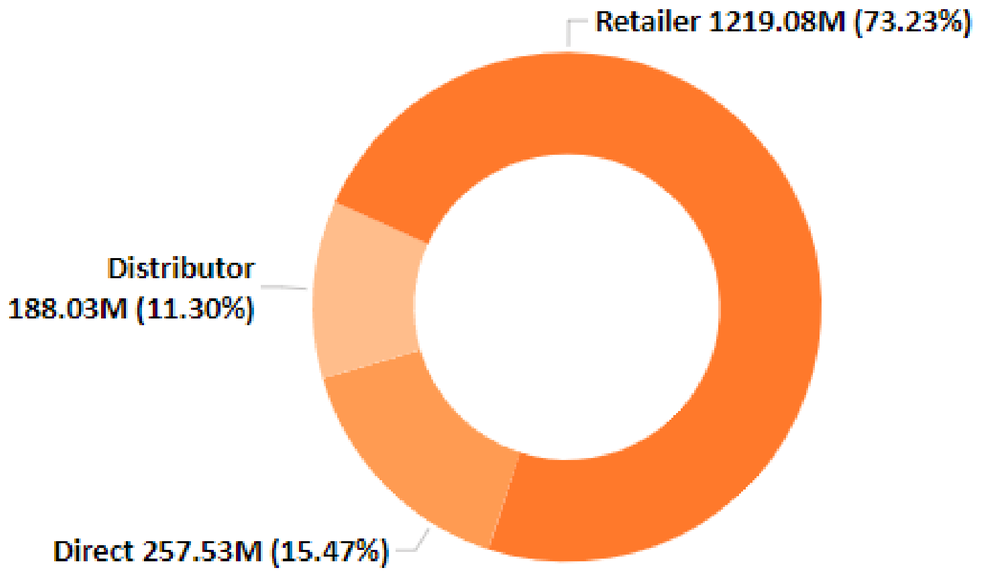
```sql
SELECT
    get_fiscal_quarter(date) as Quarter,
    SUM( sold_quantity ) AS total_sold_quantity
FROM
    fact_sales_monthly
WHERE
    fiscal_year = 2020
GROUP BY
    Quarter
ORDER BY
    total_sold_quantity DESC;
```

```sql
CREATE DEFINER=`root`@`localhost` FUNCTION `get_fiscal_quarter`(
        calender_date DATE
) RETURNS char(2) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    DECLARE fiscal_date INT;
    DECLARE fiscal_quarter CHAR(2);
    SET fiscal_date =
        DATE_ADD(
            calender_date,
            INTERVAL 4 MONTH
        );
    SET fiscal_quarter =
        CONCAT(
            'Q',
            QUARTER( fiscal_date )
        );
    RETURN fiscal_quarter;
END
```

Month	Quarter	Total Sold Quantity
September	Q1	1.76M
October	Q1	2.19M
November	Q1	3.05M
January	Q2	1.76M
February	Q2	1.70M
December	Q2	3.18M
March	Q3	0.24M
April	Q3	0.82M
May	Q3	1.02M
June	Q4	1.56M
July	Q4	1.69M
August	Q4	1.79M



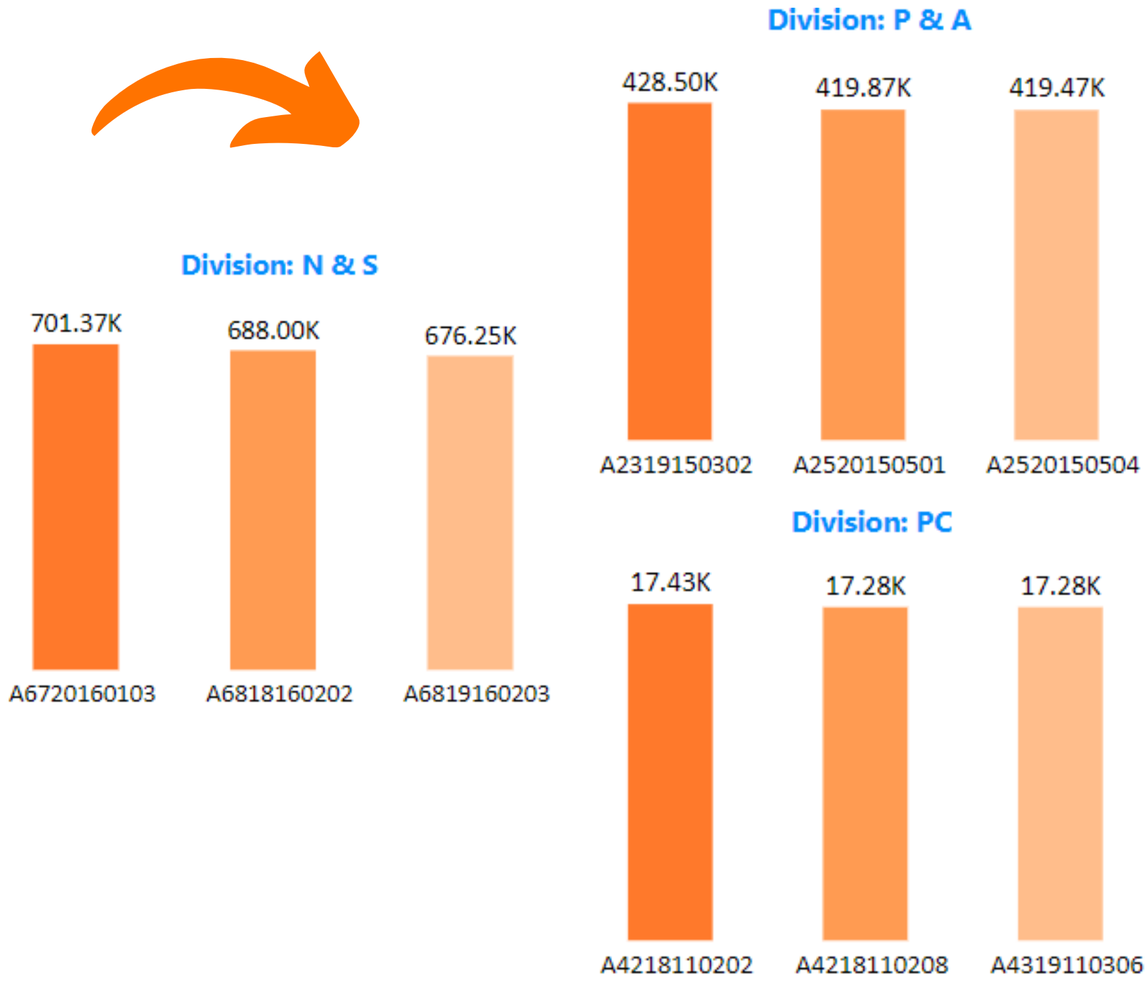
Q9. Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution? The final output contains these fields: channel, gross_sales_mln and percentage




```sql
WITH
    channel_sales_2021 AS(
        SELECT
            dc.channel,
            ROUND( SUM( (fsm.sold_quantity * fgp.gross_price) / 1000000 ), 2 ) AS gross_sales_mln
        FROM
            fact_sales_monthly fsm
        JOIN
            dim_customer dc
                ON
                    fsm.customer_code = dc.customer_code
        JOIN
            fact_gross_price fgp
                ON
                    fsm.product_code = fgp.product_code
                    AND
                    fsm.fiscal_year = fgp.fiscal_year
        WHERE
            fsm.fiscal_year = 2021
        GROUP BY
            dc.channel
        ORDER BY
            gross_sales_mln DESC
    ),
    total_sales_2021 AS(
        SELECT
            SUM(gross_sales_mln) AS total_gross_sales_mln
        FROM
            channel_sales_2021
    )
SELECT
    cs21.channel,
    CONCAT( cs21.gross_sales_mln, 'M' ) AS gross_sales_mln,
    CONCAT( ROUND( ( (cs21.gross_sales_mln * 100) / ts21.total_gross_sales_mln ), 2 ), '%' ) AS percentage
FROM
    channel_sales_2021 cs21,
    total_sales_2021 ts21;
```

Q10. 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 and rank_order

```sql
WITH
    division_sales_2021 AS(
        SELECT
            dp.division, dp.product_code,
            CONCAT( dp.product, ' (', dp.variant, ')') AS product,
            SUM( fsm.sold_quantity ) AS total_sold_quantity
        FROM fact_sales_monthly fsm
        JOIN dim_product dp
                ON
                    fsm.product_code = dp.product_code
        WHERE
            fsm.fiscal_year = 2021
        GROUP BY
            division,
            dp.product_code,
            dp.product
    ),
    sales_rank_2021 AS(
        SELECT
            *,
            DENSE_RANK() OVER( PARTITION BY division ORDER BY total_sold_quantity DESC ) AS rank_order
        FROM
            division_sales_2021
    )
    SELECT
        *
    FROM
        sales_rank_2021
    WHERE
        rank_order ≤ 3;
```



THANK YOU