CONSUMER AD-HOC INSIGHTS

AtliQ Hardware

Presented by - Priya Jangid

AtliQ Hardware

About the Company

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

It offers products across three main divisions:

- **Networking & Storage**
- **PC**
- **Peripherals & Accessories**

AtliQ Hardware operates globally in :

- **North America (NA)**
- **Latin America (LATAM)**
- **Europe (EU)**
- **Asia-Pacific (APAC) regions.**

AtliQ Hardware

Objective

- AtliQ Hardware (imaginary company) is a leading computer hardware producer in India with a global presence.
- However, the management noticed that they do not get enough insights to make quick and smart data-informed decisions.
- So to expand their data analytics team, they’re conducting a SQL challenge to assess candidates' technical expertise and problem-solving abilities.
- So, the company seeks insights for 10 ad-hoc requests.

AtliQ Hardware

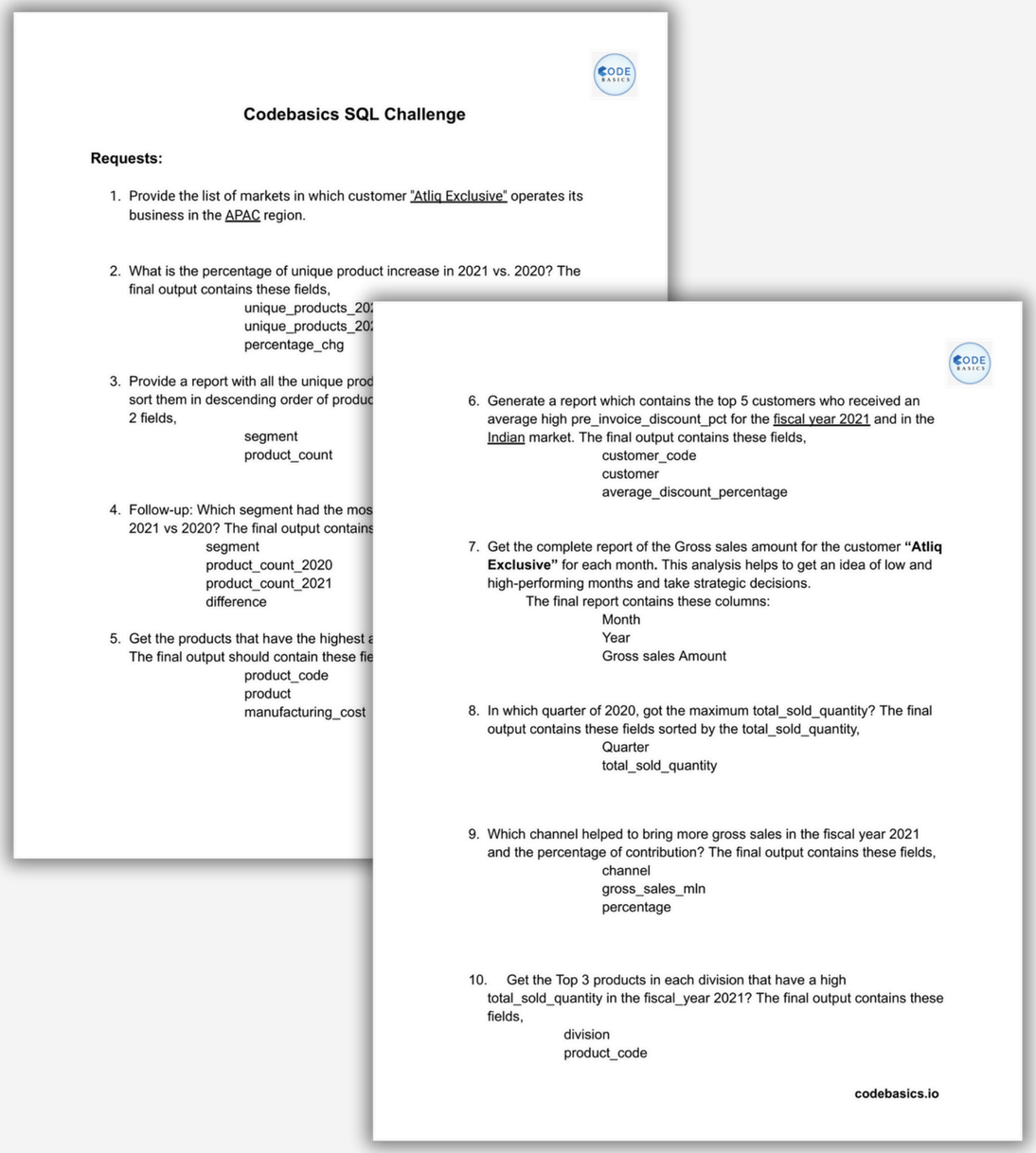
Ad-Hoc Requests & Tools

MySQL®

For Ad-Hoc Queries

Power BI

For Visualization



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_20
   unique_products_20
   percentage_chg

3. Provide a report with all the unique prod
   sort them in descending order of produc
   2 fields,
   segment
   product_count

4. Follow-up: Which segment had the mos
   2021 vs 2020? The final output contains
   segment
   product_count_2020
   product_count_2021
   difference

5. Get the products that have the highest a
   The final output should contain these fie
   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

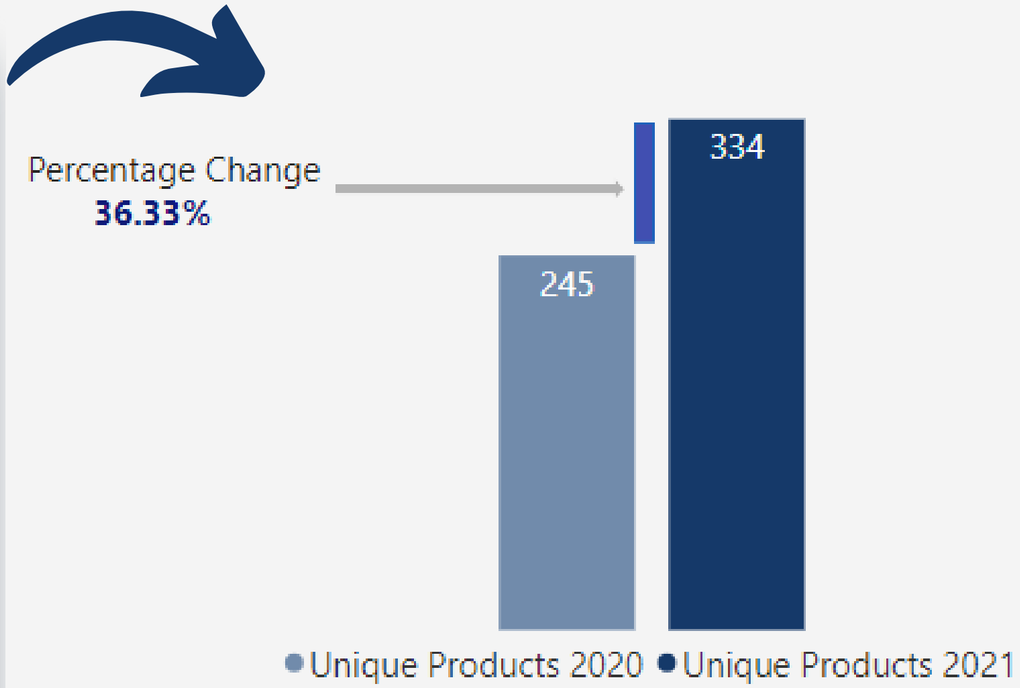
Q1: Provide list of markets in which customer “Atliq Exclusive” operates its business in the APAC region.

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

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

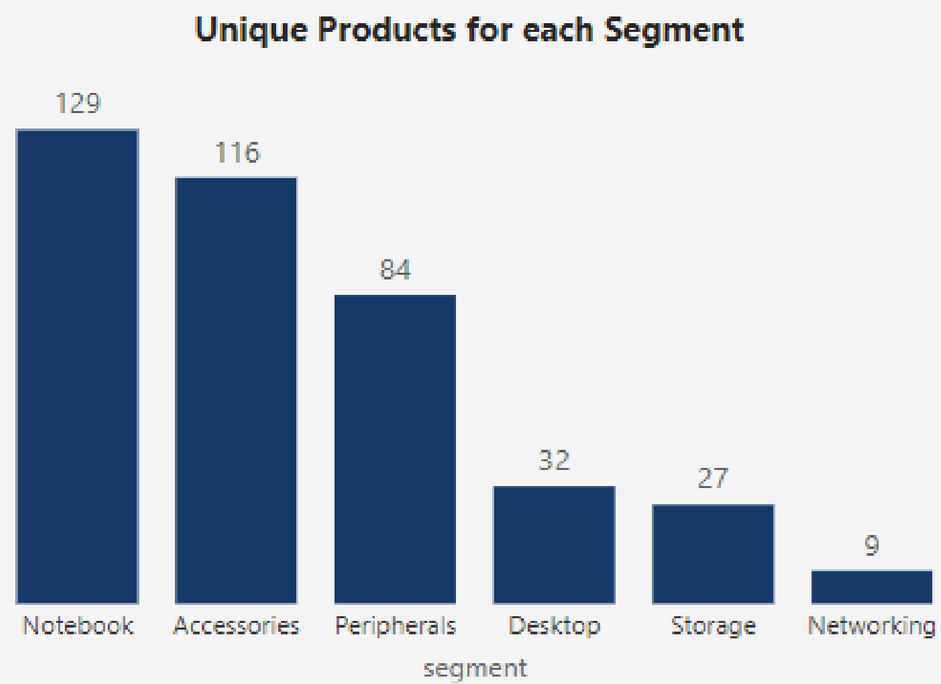
unique_products_2020, unique_products_2021, percentage_chg.

```sql
WITH
unique_prod_20 AS (
SELECT
    COUNT(DISTINCT product_code) AS unique_product_20
FROM fact_sales_monthly
WHERE fiscal_year = 2020
),
unique_prod_21 AS (
SELECT
    COUNT(DISTINCT product_code) AS unique_product_21
FROM fact_sales_monthly
WHERE fiscal_year = 2021
)
SELECT a.unique_product_20,
       b.unique_product_21,
       ROUND((b.unique_product_21 - a.unique_product_20)*100/a.unique_product_20, 2) AS percentage_chg
FROM unique_prod_20 a,
  unique_prod_21 b;
```



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, product_count.

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



Q4: 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
Desktop	7	22	15
Networking	6	9	3
Notebook	92	108	16
Peripherals	59	75	16
Storage	12	17	5

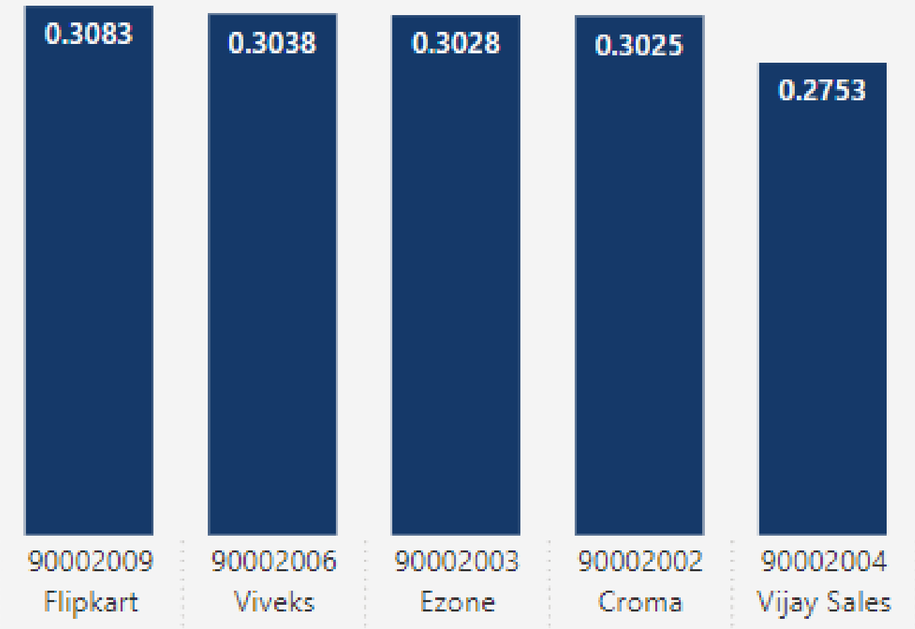
```sql
WITH
prod_count_2020 AS (
SELECT p.segment,
            COUNT(DISTINCT s.product_code) AS product_count_20
FROM fact_sales_monthly s
JOIN dim_product p
USING(product_code)
WHERE s.fiscal_year = 2020
GROUP BY p.segment
),
prod_count_2021 AS (
SELECT p.segment,
            COUNT(DISTINCT s.product_code) AS product_count_21
FROM fact_sales_monthly s
JOIN dim_product p
USING(product_code)
WHERE s.fiscal_year = 2021
GROUP BY p.segment
)
SELECT  a.segment,
            a.product_count_20,
            b.product_count_21,
        (b.product_count_21 - a.product_count_20) AS difference
FROM prod_count_2020 a
JOIN prod_count_2021 b
ON a.segment = b.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 p.product,
       m.product_code,
       m.manufacturing_cost
FROM fact_manufacturing_cost m
JOIN dim_product p
ON m.product_code = p.product_code
WHERE m.manufacturing_cost IN (
                               SELECT MAX(manufacturing_cost) FROM fact_manufacturing_cost
                               UNION
                               SELECT MIN(manufacturing_cost) FROM fact_manufacturing_cost
                              )
ORDER BY m.manufacturing_cost DESC;
```

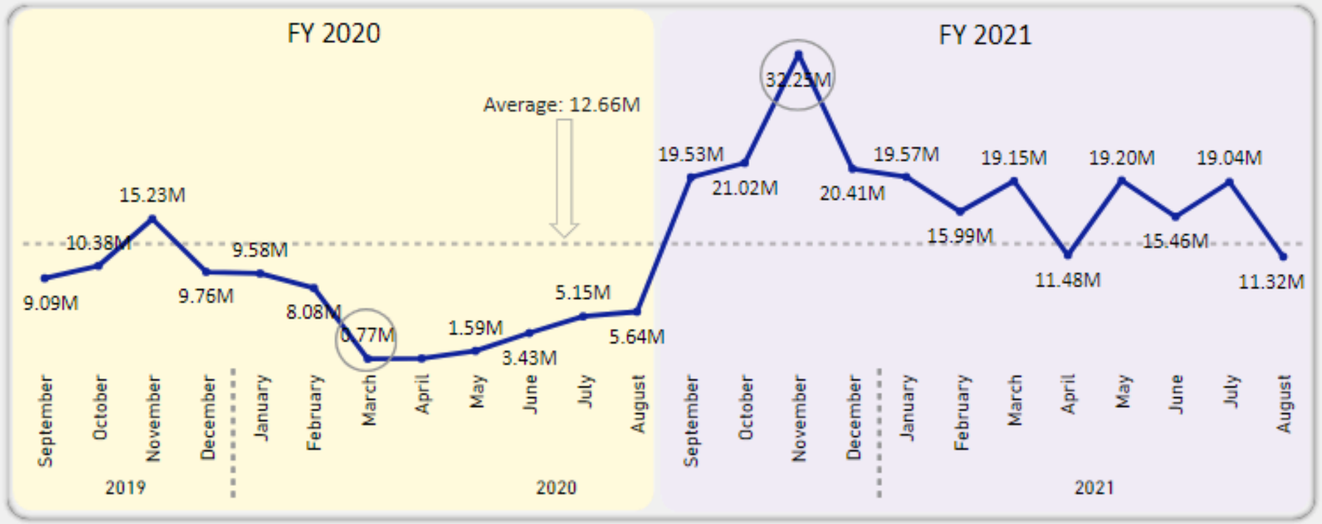
Q6: Generate a report which contains the top 5 customers who received an average high pre_invoice_discount_pct for the fiscal year 2021 and in the Indian market. The final output contains these fields:
customer_code, customer, average_discount_percentage.

```sql
SELECT c.customer,
       p.customer_code,
       AVG(p.pre_invoice_discount_pct) AS avg_discount_pct
FROM fact_pre_invoice_deductions p
JOIN dim_customer c
USING (customer_code)
WHERE c.market = "India"
           AND
      fiscal_year = 2021
GROUP BY c.customer
ORDER BY p.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. The final report contains these columns: Month, Year, Gross sales Amount.

```sql
SELECT
    MONTHNAME(s.date) AS month,
    s.fiscal_year AS fiscal_year,
    ROUND(SUM(s.sold_quantity * g.gross_price), 2) AS gross_sales_amount
FROM
    fact_sales_monthly s
JOIN
    fact_gross_price g
USING (product_code, fiscal_year)
JOIN
    dim_customer c
ON s.customer_code = c.customer_code
WHERE
    c.customer = 'Atliq Exclusive'
GROUP BY month, fiscal_year
ORDER BY fiscal_year;
```

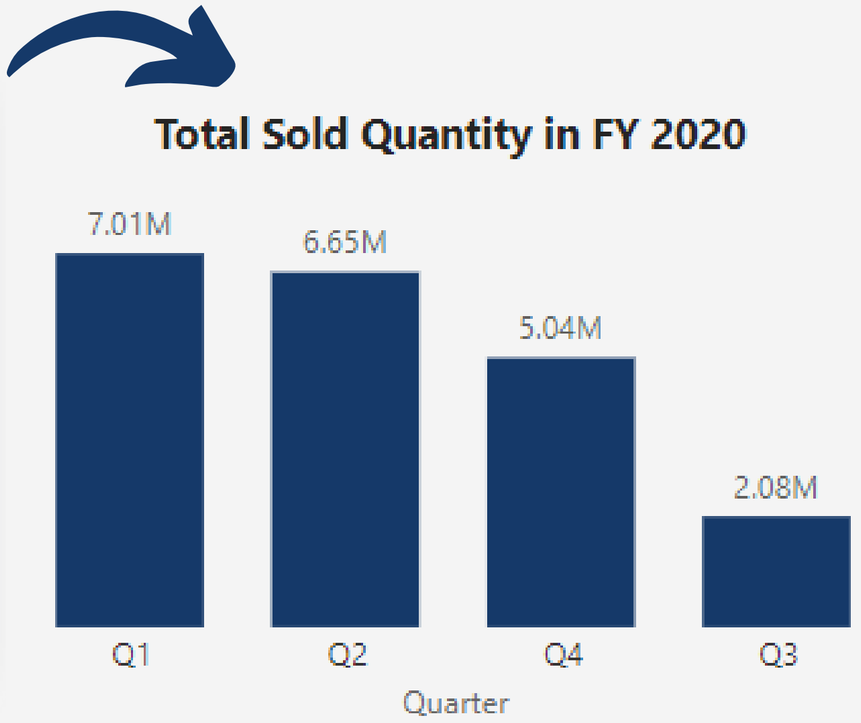


Gross sales amount report for **Atliq Exclusive** by month

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.

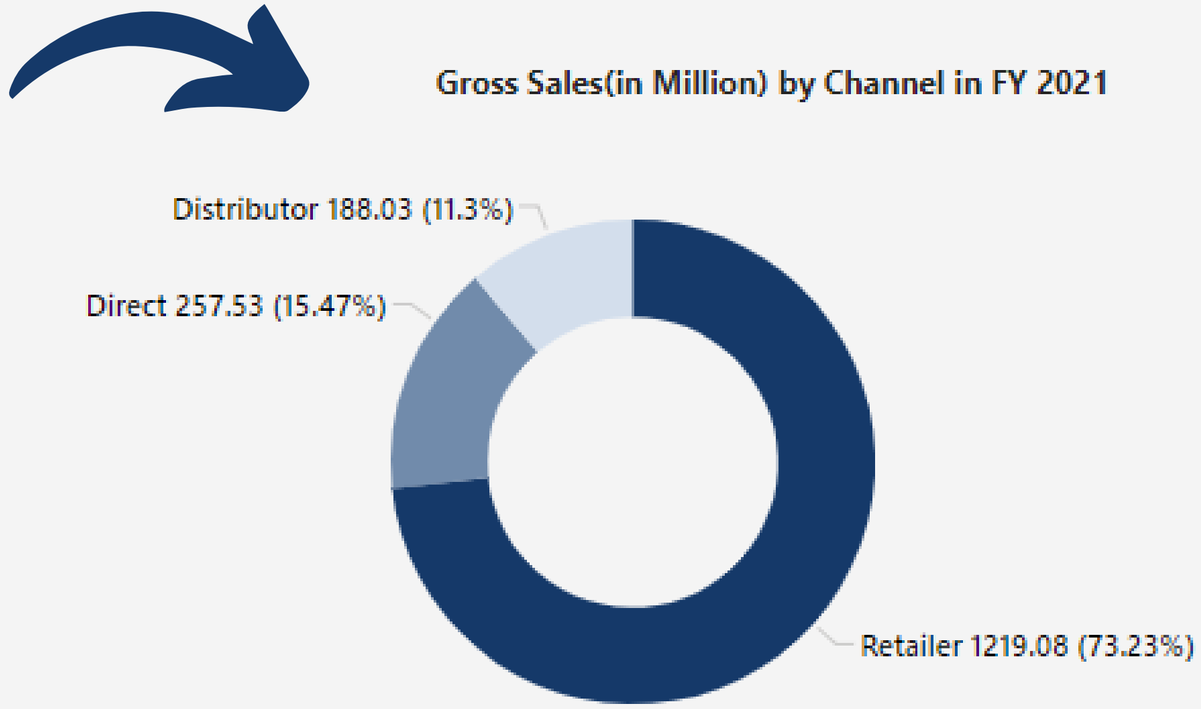
```sql
CREATE DEFINER=`root`@`localhost` FUNCTION `get_quarter`(
calendar_date DATE
) RETURNS CHAR(2) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    DECLARE month_no TINYINT;
    DECLARE qtr CHAR(2);
    SET month_no = MONTH(calendar_date);

    CASE
    WHEN month_no IN (9, 10, 11) THEN SET qtr = "Q1";
    WHEN month_no IN (12, 1, 2) THEN SET qtr = "Q2";
    WHEN month_no IN (3, 4, 5) THEN SET qtr = "Q3";
    ELSE SET qtr = "Q4";
    END CASE;

RETURN qtr;
END
```

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

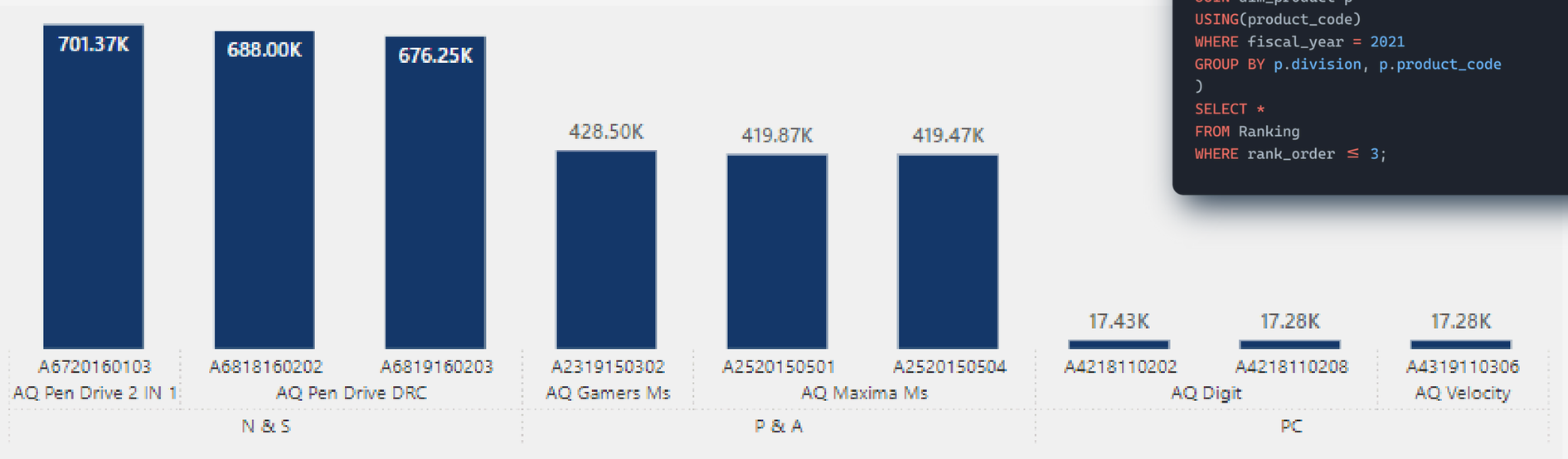


Q9: Which channel helped to bring more gross sales in the fiscal year 2021 and the percentage of contribution? The final output contains these fields: channel, gross_sales_mln, percentage.

```sql
WITH
channel_sales_2021 AS (
SELECT c.channel,
       ROUND(SUM(s.sold_quantity*g.gross_price/1000000), 2) AS gross_sales_mln
FROM fact_sales_monthly s
JOIN dim_customer c
ON s.customer_code = c.customer_code
JOIN fact_gross_price g
ON g.product_code = s.product_code
            AND
    g.fiscal_year = s.fiscal_year
WHERE s.fiscal_year = 2021
GROUP BY c.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 cs.*,
       ROUND(cs.gross_sales_mln*100/ts.total_gross_sales_mln, 2) AS percentage
FROM channel_sales_2021 cs,
     total_sales_2021 ts;
```



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, rank_order.




```sql
WITH
Ranking AS (
SELECT p.division,
       p.product_code,
       p.product,
       SUM(sold_quantity) AS total_sold_quantity,
       DENSE_RANK() OVER(PARTITION BY division ORDER BY SUM(sold_quantity) DESC) AS rank_order
FROM fact_sales_monthly s
JOIN dim_product p
USING(product_code)
WHERE fiscal_year = 2021
GROUP BY p.division, p.product_code
)
SELECT *
FROM Ranking
WHERE rank_order ≤ 3;
```

THANK YOU