AD-HOC Requests

1) 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'
ORDER BY market;
```

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

 # Count the number of distinct (unique) products available or sold in the year 2020.

 # Percentage Increase = (Unique Products in 2021 - Unique Products in 2020) / Unique Products in 2020 * 100

```sql
WITH cte_up_20 AS
(SELECT
  COUNT(DISTINCT product_code) AS unique_products_2020
FROM gdb023.fact_sales_monthly
WHERE fiscal_year = '2020'),

cte_up_21 AS
(SELECT
  COUNT(DISTINCT product_code) AS unique_products_2021
FROM gdb023.fact_sales_monthly
WHERE fiscal_year = '2021')

SELECT
  cte_up_20.unique_products_2020,
  cte_up_21.unique_products_2021,
  ROUND((cte_up_21.unique_products_2021 - cte_up_20.unique_products_2020) /
  cte_up_20.unique_products_2020 * 100,2) AS percent_chg
FROM cte_up_20, cte_up_21;
```

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 and product_count

```sql
SELECT
  segment,
  COUNT(DISTINCT product_code) AS product_count

FROM dim_product
GROUP BY segment
ORDER BY product_count DESC;
```

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 and difference

```sql
WITH cte_up_20 AS(
SELECT
  product.segment,
  COUNT(DISTINCT product_code) AS product_2020

FROM dim_product AS product
JOIN fact_sales_monthly AS sales
USING(product_code)

WHERE sales.fiscal_year = 2020
GROUP BY product.segment),

cte_up_21 AS(
SELECT product.segment,
       COUNT(DISTINCT product_code) AS product_2021

FROM dim_product AS product
JOIN fact_sales_monthly AS sales
USING(product_code)

WHERE sales.fiscal_year = 2021
GROUP BY product.segment)

SELECT
  cte_up_20.segment,
  cte_up_20.product_2020,
  cte_up_21.product_2021,
  (cte_up_21.product_2021 - cte_up_20.product_2020) AS difference

FROM cte_up_20 JOIN cte_up_21
USING(segment)

ORDER BY difference DESC;
```

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

```sql
SELECT
  product_code,
  product,
  manufacturing_cost

FROM fact_manufacturing_cost
JOIN dim_product
USING(product_code)
WHERE
    manufacturing_cost = (SELECT max(manufacturing_cost) FROM fact_manufacturing_cost)
OR  manufacturing_cost = (SELECT min(manufacturing_cost) FROM fact_manufacturing_cost)

ORDER BY manufacturing_cost DESC;
```

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 and average_discount_percentage

```sql
SELECT
  customer_code,
  customer,
  ROUND(AVG(pre_invoice_discount_pct)* 100 ,2) AS avg_discount_pct

FROM fact_pre_invoice_deductions
JOIN dim_customer
  USING (customer_code)
WHERE
    market = 'India' AND fiscal_year = 2021
GROUP BY
    customer_code, customer
ORDER BY
    avg_discount_pct DESC
LIMIT 5;
```

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 and Gross sales Amount

 # Gross Sales = Quantity Sold × Price Per Unit

```sql
SELECT
    MONTHNAME(sales.date) AS `month`,
    sales.fiscal_year,
    CONCAT(ROUND(SUM(gross.gross_price * sales.sold_quantity)/
1000000,2),' M') AS gross_sales_mln

FROM fact_sales_monthly AS sales
JOIN fact_gross_price AS gross
    USING(product_code)

JOIN dim_customer
    USING(customer_code)

WHERE customer = 'Atliq Exclusive'
GROUP BY `month`, fiscal_year
ORDER BY fiscal_year;
```

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 and total_sold_quantity

```sql
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 quarters,
    CONCAT(ROUND(SUM(sold_quantity) /1000000,2), ' M') AS sold_quantity_mln

FROM fact_sales_monthly
WHERE fiscal_year = 2020
GROUP BY quarters
ORDER BY sold_quantity DESC;
```

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 and percentage

```sql
WITH temp_table AS
(SELECT
    customer.channel,
    SUM(gross.gross_price * sales.sold_quantity) AS total_sales

FROM dim_customer AS customer
JOIN fact_sales_monthly AS sales
    USING(customer_code)

JOIN fact_gross_price AS gross
    USING(product_code)

WHERE sales.fiscal_year = 2021
GROUP BY customer.channel
ORDER BY total_sales DESC)

SELECT
  channel,
  ROUND((total_sales / 1000000),2) AS total_sales_mln,
  ROUND((total_sales / SUM(total_sales) OVER() ) * 100,2) AS percentage FROM temp_table;
```

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 and rank_order

```sql
WITH temp_table AS
(SELECT
    RANK() OVER(PARTITION BY division ORDER BY SUM(sold_quantity) DESC) AS rnk,
    division,
    product_code,
    CONCAT(product, ' - ', variant) AS product,
    SUM(sold_quantity) AS total_sold_quantity

FROM dim_product
JOIN fact_sales_monthly
    USING(product_code)

WHERE fiscal_year = 2021
GROUP BY division, product_code, product, variant)

SELECT * FROM temp_table
WHERE rnk <= 3;
```