Q1: Top 3 Most Profitable Customers by Region (Last 12 Months)

```
WITH last_12_months AS (SELECT *
                        FROM orders
                          WHERE `Order Date` >= (SELECT MAX(`Order Date`) FROM orders) -
INTERVAL 12 MONTH),

orders_not_returned AS (SELECT o.*
                        FROM last_12_months o
                        LEFT JOIN returns r
                        ON o.`Order ID` = r.`Order ID`
                        WHERE r.`Order ID` IS NULL),

customer_profit AS ( SELECT `Customer Name`, `Region`, SUM(Profit) AS total_profit
                     FROM orders_not_returned
                     GROUP BY `Customer Name`, `Region`
                     HAVING SUM(Profit) > 0 ),

ranked_customers AS (SELECT `Customer Name`,`Region`, total_profit,
                      DENSE_RANK() OVER (PARTITION BY `Region`ORDER BY total_profit DESC)
AS rank_by_region
                     FROM customer_profit)

SELECT `Region`, `Customer Name`, total_profit, rank_by_region
FROM ranked_customers
WHERE rank_by_region <= 3;
```

Q2: Identify Products with Increasing Monthly Sales Trend (Last 6 Months)

```
WITH last_6_months AS (SELECT * FROM orders
                                 WHERE STR_TO_DATE(`Order Date`, '%Y-%m-%d') >= (SELECT
MAX(STR_TO_DATE(`Order Date`, '%Y-%m-%d')) FROM orders) - INTERVAL 6 MONTH),

monthly_sales AS (SELECT `product name`,DATE_FORMAT(STR_TO_DATE(`Order Date`,
'%Y-%m-%d'), '%Y-%m') AS sales_month,SUM(sales) AS total_sales
                   FROM last_6_months
                   GROUP BY `product name`, sales_month),

valid_products AS (SELECT `product name`
                   FROM monthly_sales
                   GROUP BY `product name`
                   HAVING COUNT(DISTINCT sales_month) = 6),

with_lag AS (SELECT m.`product name`, m.sales_month, m.total_sales,
                      LAG(m.total_sales) OVER (PARTITION BY m.`product name` ORDER BY
m.sales_month) AS prev_sales
              FROM monthly_sales m JOIN valid_products v
```

```
               ON m.`product name` = v.`product name`),

growth_flags AS (SELECT *,CASE WHEN prev_sales IS NOT NULL AND total_sales > prev_sales
THEN 1 ELSE 0 END AS growth_flag
                  FROM with_lag),
count_growth AS (SELECT `product name`, COUNT(*) AS months_with_growth
                FROM growth_flags
                WHERE growth_flag = 1
                GROUP BY `product name`)

SELECT `product name`
FROM count_growth
WHERE months_with_growth = 5;
```

Q3: Calculate Return Rate by Product Sub-Category and Highlight Risky Ones

```
WITH return_info AS (SELECT o.`Sub-Category` AS subcategory,
                    COUNT(DISTINCT o.`Order ID`) AS total_orders,
                    COUNT(DISTINCT r.`Order ID`) AS total_returns
                     FROM orders o
                    LEFT JOIN returns r
                    ON o.`Order ID` = r.`Order ID`
                    GROUP BY o.`Sub-Category`)

SELECT   subcategory, ROUND(total_returns / total_orders, 3) AS return_rate,
    CASE
        WHEN (total_returns / total_orders) > 0.25 THEN 'High Risk'
        ELSE 'Acceptable'
    END AS risk_flag
FROM return_info
ORDER BY return_rate DESC;
```

Q4: Customer Purchase Frequency Segmentation

```
SELECT  `Customer ID`, `Customer Name`,
         COUNT(DISTINCT `Order ID`) AS num_orders,
        CASE
            WHEN COUNT(DISTINCT `Order ID`) <= 2 THEN 'Low'
            WHEN COUNT(DISTINCT `Order ID`) BETWEEN 3 AND 5 THEN 'Moderate'
            ELSE 'High'
        END AS segment
FROM orders
GROUP BY `Customer ID`, `Customer Name`
ORDER BY num_orders DESC;
```

Q5: Region-Wise Delayed Delivery Analysis

```
SELECT
```

```
    Region,
    ROUND(AVG(DATEDIFF(`Ship Date`, `Order Date`)), 2) AS avg_delay
FROM orders
GROUP BY Region
 HAVING avg_delay > 4;
```

Q6: Find Customers Who Ordered the Same Product More Than Once But on Different Dates

```
SELECT
    `Customer ID`,
    `Product Name`,
    COUNT(DISTINCT `Order Date`) AS distinct_order_dates
FROM orders
GROUP BY `Customer ID`, `Product Name`
HAVING COUNT(DISTINCT `Order Date`) > 1;
```

Q7: Best-Selling Product by Profit Margin (Adjusted for Shipping Cost)

```
WITH orders_cleaned AS (SELECT o.*
                        FROM orders o
                        LEFT JOIN returns r ON o.`Order ID` = r.`Order ID`
                        WHERE r.`Order ID` IS NULL),

product_profit AS (SELECT `Category` , `Product Name`, SUM(Sales) AS total_sales ,
SUM(`Shipping Cost`) AS total_shipping_cost,
                   SUM(Quantity) AS total_quantity,
                               (SUM(Sales) - SUM(`Shipping Cost`)) / SUM(Quantity) AS
net_profit_per_unit
                   FROM orders_cleaned
                   GROUP BY `Category`, `Product Name`),

ranked_products AS (SELECT *,
                        RANK() OVER (PARTITION BY Category ORDER BY net_profit_per_unit
DESC) AS rnk
                     FROM product_profit)

SELECT Category , `Product Name`,
ROUND(net_profit_per_unit, 2) AS net_profit_per_unit
FROM ranked_products
WHERE rnk = 1
ORDER BY net_profit_per_unit DESC;
```

Q8: Monthly Loss Detection Report

```
SELECT
    DATE_FORMAT(`Order Date`, '%Y-%m') AS sales_month,
    `Sub-Category`,
    ROUND(SUM(Profit), 2) AS total_loss,
    COUNT(*) AS loss_order_count
FROM orders
GROUP BY sales_month, `Sub-Category`
HAVING total_loss < 0
ORDER BY sales_month, total_loss ASC;
```

Q9: Year-on-Year Growth in Total Orders Per Region

```
WITH  orders_by_year  AS  (SELECT   Region  ,  YEAR(`Order  Date`)  AS  order_year  ,
COUNT(DISTINCT `Order ID`) AS total_orders
                        FROM orders
                        GROUP BY Region, order_year),

orders_with_lag AS (SELECT  Region , order_year , total_orders,
                     LAG(total_orders) OVER (PARTITION BY Region ORDER BY order_year) AS
prev_year_orders
                    FROM orders_by_year ),

growth_calc AS (SELECT  Region , order_year , total_orders , prev_year_orders,
                  ROUND(100.0 * (total_orders - prev_year_orders) / prev_year_orders, 2)
AS growth_percent
                FROM orders_with_lag
                WHERE prev_year_orders IS NOT NULL )

SELECT  Region , order_year , growth_percent
FROM growth_calc
WHERE growth_percent > 20
ORDER BY growth_percent DESC;
```

Q10: Repeat Purchase Scorecard (RFM-lite)

```
WITH customer_stats AS (SELECT  `Customer ID`, MAX(`Order Date`) AS last_order_date ,
COUNT(DISTINCT `Order ID`) AS frequency,
                         SUM(Sales) AS monetary
                         FROM orders
                         GROUP BY `Customer ID`),

rfm_base  AS  (SELECT  `Customer  ID`,  DATEDIFF((SELECT  MAX(`Order  Date`)  FROM  orders),
last_order_date) AS recency,
             frequency , monetary
             FROM customer_stats),

rfm_ranked AS (SELECT *,
               RANK() OVER (ORDER BY recency ASC) AS r_rank,
               RANK() OVER (ORDER BY frequency DESC) AS f_rank,
               RANK() OVER (ORDER BY monetary DESC) AS m_rank
               FROM rfm_base),

rfm_scored AS ( SELECT *, (r_rank + f_rank + m_rank) AS rfm_score
                FROM rfm_ranked)

SELECt `Customer ID`, recency , frequency ,
ROUND(monetary, 2) AS monetary,
r_rank, f_rank, m_rank, rfm_score
FROM rfm_scored
ORDER BY rfm_score ASC
LIMIT 10;
```