```
## 1. Calculate the moving average of order values for each customer over their
order history.
WITH OrderValues AS (
  SELECT
    T2.customer_id,
    T2.order_purchase_timestamp,
    SUM(T1.payment_value) AS order_value
  FROM payments AS T1
  JOIN orders AS T2
    ON T1.order_id = T2.order_id
  GROUP BY
    T2.customer_id,
    T2.order_purchase_timestamp
)
SELECT
  customer_id,
  order_purchase_timestamp,
  order_value,
  AVG(order_value) OVER (
    PARTITION BY
      customer_id
    ORDER BY
      order_purchase_timestamp
    ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
  ) AS moving_average_3_orders
FROM OrderValues
ORDER BY
  customer_id,
  order_purchase_timestamp;

 ##2. Calculate the cumulative sales per month for each year.
WITH MonthlySales AS (
  SELECT
    DATE_FORMAT(T1.order_purchase_timestamp, '%Y') AS order_year,
    DATE_FORMAT(T1.order_purchase_timestamp, '%Y-%m') AS order_month,
    SUM(T2.price) AS monthly_sales
  FROM orders AS T1
  JOIN order_items AS T2
    ON T1.order_id = T2.order_id
  GROUP BY
    order_year,
    order_month
)
SELECT
  order_year,
  order_month,
  monthly_sales,
  SUM(monthly_sales) OVER (
    PARTITION BY
      order_year
    ORDER BY
      order_month
  ) AS cumulative_sales
FROM MonthlySales
ORDER BY
  order_year,
  order_month;

  ## 3. Calculate the year-over-year growth rate of total sales.
  WITH AnnualSales AS (
  SELECT
    YEAR(T1.order_purchase_timestamp) AS order_year,
    SUM(T2.price) AS yearly_sales
  FROM orders AS T1
```

```
  JOIN order_items AS T2
    ON T1.order_id = T2.order_id
  GROUP BY
    order_year
)
SELECT
  order_year,
  yearly_sales,
  (
    (yearly_sales - LAG(yearly_sales) OVER (
      ORDER BY
        order_year
    )) / LAG(yearly_sales) OVER (
      ORDER BY
        order_year
    )
  ) * 100 AS yoy_growth_rate_percentage
FROM AnnualSales
ORDER BY
  order_year;

  ##4. Calculate the retention rate of customers, defined as the percentage of
customers who make another purchase within 6 months of their first purchase.
  WITH CustomerOrders AS (
  SELECT
    T1.customer_unique_id,
    T2.order_purchase_timestamp
  FROM Customers AS T1
  JOIN orders AS T2
    ON T1.customer_id = T2.customer_id
), FirstPurchaseDates AS (
  SELECT
    customer_unique_id,
    MIN(order_purchase_timestamp) AS first_purchase_date
  FROM CustomerOrders
  GROUP BY
    customer_unique_id
), RetainedCustomers AS (
  SELECT DISTINCT
    T1.customer_unique_id
  FROM FirstPurchaseDates AS T1
  JOIN CustomerOrders AS T2
    ON T1.customer_unique_id = T2.customer_unique_id
  WHERE
    T2.order_purchase_timestamp > T1.first_purchase_date AND
T2.order_purchase_timestamp <= DATE_ADD(T1.first_purchase_date, INTERVAL 6
MONTH)
)
SELECT
  (
    (
      SELECT
        COUNT(*)
      FROM RetainedCustomers
    ) * 100.0
  ) / (
    SELECT
      COUNT(*)
    FROM FirstPurchaseDates
  ) AS retention_rate_percentage;

  ##5. Identify the top 3 customers who spent the most money in each year.
  WITH CustomerOrderSpending AS (
  SELECT
```

```
    T1.order_id,
    SUM(T1.payment_value) AS order_value
  FROM payments AS T1
  GROUP BY
    T1.order_id
), CustomerAnnualSpending AS (
  SELECT
    YEAR(T2.order_purchase_timestamp) AS order_year,
    T3.customer_unique_id,
    SUM(T1.order_value) AS total_spent
  FROM CustomerOrderSpending AS T1
  JOIN orders AS T2
    ON T1.order_id = T2.order_id
  JOIN Customers AS T3
    ON T2.customer_id = T3.customer_id
  GROUP BY
    order_year,
    T3.customer_unique_id
), RankedCustomers AS (
  SELECT
    order_year,
    customer_unique_id,
    total_spent,
    RANK() OVER (
      PARTITION BY
        order_year
      ORDER BY
        total_spent DESC
    ) AS customer_rank
  FROM CustomerAnnualSpending
)
SELECT
  order_year,
  customer_unique_id,
  total_spent,
  customer_rank
FROM RankedCustomers
WHERE
  customer_rank <= 3
ORDER BY
  order_year,
  customer_rank;
```