```
## 1. Calculate the number of orders per month in 2018.
SELECT
  DATE_FORMAT(order_purchase_timestamp, '%Y-%m') AS order_month,
  COUNT(order_id) AS order_count
FROM orders
WHERE
  YEAR(order_purchase_timestamp) = 2018
GROUP BY
  order_month
ORDER BY
  order_month;

  ##2. Find the average number of products per order. grouped by customer city.
WITH ProductsPerOrder AS (
  SELECT
    order_id,
    COUNT(product_id) AS product_count
  FROM order_items
  GROUP BY
    order_id
)
SELECT
  T3.customer_city,
  AVG(T1.product_count) AS avg_products_per_order
FROM ProductsPerOrder 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
  T3.customer_city
ORDER BY
  avg_products_per_order DESC;

 ##3. Calculate the percentage of total revenue contributed by each product
category.
 WITH CategoryRevenue AS (
  SELECT
    T1.product_category,
    SUM(T2.price) AS revenue
  FROM products AS T1
  JOIN order_items AS T2
    ON T1.product_id = T2.product_id
  GROUP BY
    T1.product_category
)
SELECT
  product_category,
  (revenue * 100.0) / (
    SELECT
      SUM(revenue)
    FROM CategoryRevenue
  ) AS percentage_of_total_revenue
FROM CategoryRevenue
ORDER BY
  percentage_of_total_revenue DESC;

  ##4. Identify the correlation between product price and the number of times a
product has been purchased.
  SELECT
  product_id,
  AVG(price) AS average_price,
  COUNT(order_item_id) AS purchase_count
FROM order_items
```

```
GROUP BY
  product_id;

  ##5. Calculate the total revenue generated by each seller, and rank them by
revenue.
  SELECT
  T1.seller_id,
  SUM(T2.price) AS total_revenue,
  RANK() OVER (
    ORDER BY
      SUM(T2.price) DESC
  ) AS revenue_rank
FROM sellers AS T1
JOIN order_items AS T2
  ON T1.seller_id = T2.seller_id
GROUP BY
  T1.seller_id
ORDER BY
  revenue_rank;
```