1. Total Orders

```
SELECT
    COUNT(*) AS Total_orders
FROM orders_dataset;
```

2. Check status of orders

```
SELECT
    order_status,
    COUNT(*) AS status_count
FROM orders_dataset
GROUP BY order_status
ORDER BY status_count DESC;
```

3. Total Revenue of all orders and delivered orders

```
SELECT
    SUM(payment_value) as total_revenue
FROM payments_dataset;
```

4. Total Revenue per order status and number of payment types

```
-- use VIEW table order_payment_summary

SELECT

        o.order_status,

        COUNT(total_paid) AS num_payment,

        SUM(p.total_paid) as total_paid_orders

FROM orders_dataset o

JOIN order_payment_summary p ON o.order_id = p.order_id

GROUP BY o.order_status

ORDER BY num_payment DESC ,total_paid_orders;
```

5. Total Unique Customers

```
-- Each customer with unique id may have placed order multiple times but system assigned new customer id per transaction

--customer_id>customer_unique_id


SELECT

        COUNT (DISTINCT customer_unique_id) AS Total_Unique_Customers

FROM customers_dataset;
```

6. Average Items per Order

```
SELECT

        AVG(price)

FROM order_items_dataset;
```

7. Average Review Score

```
SELECT
	AVG(review_score)
FROM reviews_dataset;
```

8. Average Delivery Time (in days)

```
SELECT
	AVG(
		EXTRACT(
			EPOCH FROM(order_delivered_customer_date - order_purchase_timestamp)/86400 )) AS Avg_delivery_time
FROM orders_dataset
WHERE order_status = 'delivered';
```

9. Top 10 Cities by Number of Orders

```
SELECT
	customer_city,
	COUNT(customer_id) as number_of_orders,
	DENSE_RANK() OVER (ORDER BY COUNT(customer_id) DESC) as Rank_number
FROM customers_dataset
GROUP BY customer_city
LIMIT 10;
```

10. Products Category by Quantity Sold

```
SELECT
        count(o.order_id) as order_count,
        t.product_category_name_english
FROM order_items_dataset o
JOIN products_dataset p ON o.product_id = p.product_id
JOIN product_translation t ON p.product_category_name = t.product_category_name
GROUP BY 2
ORDER BY 1 DESC
```

11. Top Sellers by Revenue

```
SELECT
        seller_id,
        SUM(price) as total_sales
FROM order_items_dataset
GROUP BY seller_id
ORDER BY total_sales DESC
LIMIT 1;
```

12. On-Time Delivery Rate using subquery and CTE

--USING CTE

```
WITH OTDR as (
	SELECT
		COUNT(*) as total_delivery,
		COUNT(*) FILTER(
			WHERE order_delivered_customer_date <=
order_estimated_delivery_date
			) as ontime_delivery
	FROM orders_dataset
)
SELECT
	ROUND(ontime_delivery::DECIMAL / total_delivery, 2) AS Ontime_delivery_rate
FROM OTDR
```

--USING SUBQUERY

```
SELECT
	ROUND(ontime_delivery::DECIMAL / total_delivery, 2) AS Ontime_delivery_rate
FROM(
SELECT
		COUNT(*) as total_delivery,
		COUNT(*) FILTER(
			WHERE order_delivered_customer_date <=
order_estimated_delivery_date
			) as ontime_delivery
	FROM orders_dataset)
```

13. Return/Cancellation Rate

```
SELECT
    ROUND(C.cancelled_orders::decimal /T.total_orders,2) as cancellation_percentage
    FROM
        (SELECT
                count(*) as cancelled_orders
            FROM orders_dataset
            WHERE order_status = 'canceled') as C,
        (SELECT
                count(*) as total_orders
            FROM orders_dataset) as T
```

14. Top Selling Product by Month

1st CTE: get sales per month by product category, 2nd CTE: use 1st CTE to create ranking within month, final select: filter by sales with rank 1 only

```
WITH category_monthly_sales AS (
   SELECT
      TO_CHAR(d.order_purchase_timestamp, 'YYYY-MM') AS order_month,
      t.product_category_name_english,
      SUM(o.price)  AS total_sales
   FROM order_items_dataset o
   JOIN orders_dataset d ON o.order_id = d.order_id
   JOIN products_dataset p ON o.product_id = p.product_id
   JOIN product_translation t ON p.product_category_name = t.product_category_name
   GROUP BY order_month, t.product_category_name_english
),
ranked_categories AS (
   SELECT *,
        RANK() OVER (PARTITION BY order_month ORDER BY total_sales DESC) AS category_rank
   FROM category_monthly_sales
)
SELECT
   order_month,
   product_category_name_english,
   total_sales
FROM ranked_categories
WHERE category_rank = 1
ORDER BY order_month;
```