

USE superstore;

-- 1. Show each customer along with their complete address in one table.

```
SELECT * FROM customers
      INNER JOIN customer_addresses
      ON customers.postal_code = customer_addresses.postal_code;
```

-- 2. Find any orders that do not have a postal code available for shipment.

```
SELECT * FROM orders JOIN customers ON orders.customer_id = customers.customer_id
WHERE postal_code IS NULL;
```

-- 3. Find the customer that has placed the most orders.

```
SELECT customer_name, COUNT(order_id) AS top_customer FROM orders
JOIN customers ON orders.customer_id = customers.customer_id
GROUP BY customer_name ORDER BY top_customer DESC LIMIT 1;
```

-- 4. Find the total number of orders placed by each customer segment in the year 2016.

```
SELECT segment, COUNT(order_id) AS total_orders FROM orders
JOIN customers ON orders.customer_id = customers.customer_id
WHERE order_date >= '2016-01-01' AND order_date <= '2016-12-31'
GROUP BY segment ORDER BY total_orders DESC ;
```

-- -----

```
SELECT segment, COUNT(order_id) AS total_orders FROM customers
JOIN orders ON customers.customer_id = orders.customer_id
WHERE YEAR(order_date) = 2016 GROUP BY segment ORDER BY total_orders DESC;
```

-- 5. Show any customers that have not yet placed an order.

```
SELECT customer_name FROM customers LEFT JOIN orders ON customers.customer_id =
orders.customer_id
WHERE orders.order_id IS NULL;
```

-- 6. Find the name of the top-rated customer that has spent the greatest amount of money.

```
SELECT customer_name, SUM(products.sales) AS topRated_customer FROM customers
```

```
JOIN orders ON customers.customer_id = orders.customer_id
JOIN order_products ON orders.order_id = order_products.order_id
JOIN products ON products.product_id = order_products.product_id
GROUP BY customer_name ORDER BY top_rated_customer DESC LIMIT 1;
```

-- 7. Find the distribution of customer segments in each city.

```
SELECT city, COUNT(*) AS segment_count FROM customers
JOIN customer_addresses ON customers.postal_code = customer_addresses.postal_code
GROUP BY city ORDER BY segment_count DESC;
```

-- -----

-- 8. Find the total sales volume per product category.

```
SELECT category, SUM(sales) AS total_sales FROM products
JOIN product_categories ON products.sub_category = product_categories.sub_category
GROUP BY category ORDER BY total_sales DESC;
```

-- 9. Find the names of any cities that have not placed any orders yet.

```
SELECT city FROM customer_addresses
JOIN customers ON customer_addresses.postal_code = customers.postal_code
JOIN orders ON customers.customer_id = orders.customer_id
WHERE orders.customer_id IS NULL ;
```

-- -----

```
SELECT DISTINCT city FROM customer_addresses
LEFT JOIN customers ON customer_addresses.postal_code = customers.postal_code
WHERE customer_ID IS NULL;
```

-- 10. Re-create the Table:

```
SELECT * FROM customer_addresses LEFT JOIN customers
ON customer_addresses.postal_code = customers.postal_code LEFT JOIN orders
ON customers.customer_id = orders.customer_id LEFT JOIN order_products
ON orders.order_id = order_products.order_id LEFT JOIN products
ON order_products.product_id = products.product_id LEFT JOIN product_categories
ON products.sub_category = product_categories.sub_category;
```

