Part 1: Exploration with SQL

Q1. How many customers do we have in the data?

```
SELECT
        COUNT(customer_id) AS total_customers
FROM customers
```

Q2. What was the city with the most profit for the company in 2015?

```
SELECT
        o.shipping_city,
        SUM(od.order_profits) AS total_profits
FROM orders o
JOIN order_details od
        ON o.order_id = od.order_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2015
GROUP BY o.shipping_city
ORDER BY total_profits DESC
LIMIT 1
```

Q3. In 2015, what was the most profitable city's profit?

```
SELECT
        MAX(total_profits) AS most_profitable_city_profit
FROM (
 SELECT
        o.shipping_city,
        SUM(od.order_profits) AS total_profits
FROM orders o
JOIN order_details od
        ON o.order_id = od.order_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2015
GROUP BY o.shipping_city
)
        AS city_profits
```

Q4. How many different cities do we have in the data?

```
SELECT
        DISTINCT shipping_city
FROM orders

SELECT
        COUNT(DISTINCT shipping_city) AS total_cities
FROM orders
```

Q5. Show the total spent by customers from low to high.

```
SELECT
        o.customer_id,
        ROUND(SUM(od.order_sales * (1 - od.order_discount))::numeric, 2) AS discounted_price
FROM order_details od
JOIN orders o
 ON o.order_id = od.order_id
GROUP BY o.customer_id
ORDER BY discounted_price ASC;
```

Q6. What is the most profitable city in the State of Tennessee?

```
SELECT
        o.shipping_city,
        SUM(od.order_profits) AS total_profits
FROM orders o
JOIN order_details od
        ON o.order_id = od.order_id
WHERE o.shipping_state = 'Tennessee'
GROUP BY o.shipping_city
ORDER BY total_profits DESC
LIMIT 1;
```

Q7. What's the average annual profit for that city across all years?

```
SELECT
        o.shipping_city,
        ROUND(AVG(od.order_profits)::numeric,2) AS avg_annual_profit
FROM orders o
JOIN order_details od
        ON o.order_id = od.order_id
WHERE o.shipping_city = 'Lebanon'
GROUP BY o.shipping_city;
```

Q8. What is the distribution of customer types in the data?

```
SELECT
        customer_segment,
        COUNT(*) AS total_customers
FROM customers
GROUP BY customer_segment;
```

Q9. What's the most profitable product category on average in Iowa across all years?

```
SELECT
        p.product_category,
        AVG(od.order_profits) AS most_avg_profits
FROM product p
JOIN order_details od
        ON p.product_id = od.product_id
JOIN orders o
        ON od.order_id = o.order_id
WHERE o.shipping_state = 'Iowa'
GROUP BY p.product_category
ORDER BY most_avg_profits DESC
LIMIT 1;
```

Q10. What is the most popular product in that category across all states in 2016?

```
SELECT
 p.product_name,
        COUNT(od.quantity) AS total_sales
FROM product p
JOIN order_details od
        ON p.product_id = od.product_id
JOIN orders o
        ON od.order_id = o.order_id
WHERE p.product_category = 'Furniture' AND EXTRACT(YEAR FROM o.order_date) = 2016
GROUP BY p.product_name
ORDER BY total_sales DESC
LIMIT 1;
```

Q11. Which customer got the most discount in the data? (in total amount)

```
SELECT
        c.customer_id,
 c.customer_name,
 ROUND(SUM(od.order_discount)::numeric,2) AS total_discount
FROM customers c
JOIN orders o USING (customer_id)
JOIN order_details od USING (order_id)
GROUP BY c.customer_id, c.customer_name
ORDER BY total_discount DESC
LIMIT 1;
```

Q12. How widely did monthly profits vary in 2018?

```
SELECT
        DATE_PART('month', o.order_date) AS month,
 SUM(od.order_profits) AS total_profits
FROM orders o
JOIN order_details od
ON o.order_id = od.order_id
WHERE DATE_PART('year', o.order_date) = 2018
GROUP BY DATE_PART('month', o.order_date)
ORDER BY month DESC;
```

Q13. Which was the biggest order regarding sales in 2015?

```
SELECT
        od.order_id,
 SUM(od.order_sales) AS total_sales
FROM order_details od
JOIN orders o
        ON o.order_id = od.order_id
WHERE DATE_PART('year', o.order_date) = 2015
GROUP BY od.order_id
ORDER BY total_sales DESC
LIMIT 1;
```

Q14: What was the rank of each city in the East region in 2015 in quantity?

```
SELECT
        o.shipping_city AS City,
  SUM(od.quantity) AS total_quantity,
  RANK() OVER (ORDER BY SUM(od.quantity) DESC) AS City_Rank
FROM orders o
JOIN order_details od
        ON o.order_id = od.order_id
WHERE o.shipping_region='East' AND
DATE_PART('year', o.order_date) = 2015
GROUP BY o.shipping_city
ORDER BY City_Rank;
```

Q15. Display customer names for customers who are in the segment 'Consumer' or 'Corporate.' How many customers are there in total?

```
        — - Solution 1
SELECT
 customer_name,
```

```
 customer_segment
FROM customers
WHERE customer_segment IN ('Consumer', 'Corporate');

SELECT
 COUNT(DISTINCT customer_id) AS total_customers
FROM customers
WHERE customer_segment IN ('Consumer', 'Corporate');

        – - Solution 2
SELECT
        COUNT(customer_id) AS total_customers,
 customer_name,
 customer_segment
FROM customers
WHERE customer_segment IN ('Consumer' , 'Corporate')
GROUP BY customer_segment, customer_name;

        -- Solution 3 With Window Function

WITH customer_list AS (
 SELECT
  customer_name,
  customer_segment
 FROM customers
 WHERE customer_segment IN ('Consumer', 'Corporate')
)
SELECT
 customer_name,
 customer_segment,
 (SELECT COUNT(*) FROM customer_list) AS total_customers
FROM customer_list;

        – - Solution 4
SELECT
(SELECT COUNT(customer_id) FROM customers WHERE customer_segment = 'Consumer') AS
consumer_count,
(SELECT COUNT(customer_id) FROM customers WHERE customer_segment = 'Corporate') AS
corporate_count,
COUNT(customer_id) AS total_customers
FROM customers
WHERE customer_segment IN ('Consumer', 'Corporate')
```

```
GROUP BY 1;
```

Q16. Calculate the difference between the largest and smallest order quantities for product id '100'.

```
SELECT
 od.product_id,
 MAX(od.quantity) - MIN(od.quantity) AS quantity_difference
FROM order_details od
WHERE od.product_id = 100
GROUP BY od.product_id;
```

Q17. Calculate the percent of products that are within the category 'Furniture.'

```
SELECT
   ROUND(COUNT(product_id) * 100.0 / (SELECT COUNT(product_id) FROM product), 1) AS
percent_of_furniture_products
FROM product
WHERE product_category = 'Furniture';
```

Q18. Display the number of product manufacturers with more than 1 product in the product table.

```
SELECT
        product_manufacturer,
        COUNT(product_manufacturer) AS total_manufacturers
FROM product
GROUP BY product_manufacturer
HAVING COUNT(*) > 1
ORDER BY total_manufacturers DESC;
```

– Another way to get cumulative value of all the manufacturers

```
SELECT
 COUNT(product_manufacturer) AS total_manufacturers
FROM (
 SELECT
  product_manufacturer
 FROM product
 GROUP BY product_manufacturer
 HAVING COUNT(*) > 1) AS count_of_manufacturers_count;
```

Q19. Show the product_subcategory and the total number of products in the subcategory.
--Show the order from most to least products and then by product_subcategory name ascending.

```
SELECT
        product_subcategory,
        COUNT(product_id) AS total_products
FROM product
GROUP BY product_subcategory
ORDER BY total_products DESC, product_subcategory ASC;

--With subquery
SELECT
        product_subcategory,
        total_products
FROM (
        SELECT
        product_subcategory,
                COUNT(*) AS total_products
        FROM product
        GROUP BY product_subcategory)
AS subquery
ORDER BY total_products DESC, product_subcategory ASC;
```

Q20. Show the product_id(s), the sum of quantities, where the total sum of its product quantities is greater than or equal to 100.

```
SELECT
        product_id,
 SUM(quantity) AS total_quantity
FROM order_details
GROUP BY product_id
HAVING SUM(quantity)>=100
ORDER BY total_quantity DESC;
```

Bonus Question: Join all database tables into one dataset that includes all unique columns and download it as a .csv file.

```
SELECT c.customer_id, c.customer_name, c.customer_segment,
p.product_id, p.product_name, p.product_category, p.product_subcategory,
p.product_manufacturer,
od.order_details_id, od.order_id, od.quantity, od.order_discount, od.order_profits,
od.order_profit_ratio,
od.order_sales,
o.order_id, o.order_date, o.shipping_city, o.shipping_state, o.shipping_region, o.shipping_country,
```

```
o.shipping_postal_code, o.shipping_date, o.shipping_mode
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_details od ON o.order_id = od.order_id
JOIN product p ON od.product_id = p.product_id
ORDER BY c.customer_id ASC;
```