**GA Project 2 – SQL Query Log**

**/\*COUNT (\*) and COUNT DISTINCT To Check for Duplicates\*/**

SELECT

COUNT (order\_id), COUNT (DISTINCT order\_id)

FROM orders

;

Result:

This checks if the number of rows in the Orders table and the number of unique order\_ids in the table are he same. If not, then that means each order has multiple records.

**Chart 1) OP1**

**/\*Best Performing Product Segments (Count of Instances Ordered)\*/**

SELECT

pro.category, COUNT (order\_id) as orders

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a count of all the total item order requests for each Product Category. By comparing this we can see which is the best product category throughout the whole dataset.

**Chart 2) OP2**

**/\*Top Performing Product Subcategories (Count of Order Item Requests)\*/**

SELECT

pro.sub\_category, COUNT(\*)

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a chart showing a comparison of Top Performing Subcategories in terms of the total count of order requests they get.

**Chart 3) SP11**

**/\*Total Product Category Sales Chart\*/**

SELECT

pro.category, round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

Creates a chart showing the Total Sales Value of each Product Category ($) so that we can compare their total values.

**Chart 4 ) PP11**

**/\*Total Product Category Profit Chart\*/**

SELECT

pro.category, round(SUM(profit)+0) as profit

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 1 DESC

;

Result:

Creates a chart showing the Total Profit Value of each Product Category ($). We can compare these to see which performs best.

**Chart 5) SP10**

**/\*Quarterly Sales for Each Product Category\*/**

SELECT

TO\_CHAR(DATE\_TRUNC('quarter', order\_date), 'YYYY-MM') AS quarter, pro.category,

round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1, 2

ORDER BY 1

;

Result:

This creates a chart showing the sales values of each product category every quarter. I did this to check for any seasonal trends for sales.

**Chart 6) PP10**

**/\*Quarterly Profit for Each Product Category\*/**

SELECT

TO\_CHAR(DATE\_TRUNC('quarter', order\_date), 'YYYY-MM') AS quarter, pro.category,

round(SUM(profit)+0) as profit

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1, 2

ORDER BY 1

;

Result:

This chart shows the profit values of each product category every quarter. I did this to check for any seasonal trends for profit.

**Chart 7) D1**

**/\*Discount Ranges - MIN, MAX & AVG\*/**

SELECT

pro.category, MIN(discount), MAX (discount), AVG(discount)

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a chart showing the Minimum, Maximum and Average Discount Percentage rates offered in each Product Category and so that we can compare the figures for each.

**Chart 8) SP12**

**/\*Total Sales per Subcategory Chart\*/**

SELECT

pro.sub\_category, round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a chart showing the Total Sales per Subcategory. This lets us look at what product area brings the most sales revenue.

**Chart 9) PP12**

**/\*Total Profit per Subcategory Chart\*/**

SELECT

pro.sub\_category, round(SUM(profit)+0) as profit

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a chart showing the Total Profit per Subcategory. This lets us look at what product area brings the most profit value.

**Chart 10) ---PP14**

**/\*Profit Vs Sub-Category for Technology in November 2019\*/**

SELECT

pro.sub\_category, round(SUM(profit)+0) as profit

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

WHERE order\_date BETWEEN '2019-11-01' AND '2019-11-30'

AND pro.category = 'Technology'

GROUP BY 1

ORDER BY 2 DESC

;

Result:

This creates a chart showing the amount of profit that each Technology Subcategory made in November 2019 in particular. I probed into this to find out what caused the November 2019 sales and profit spike.

**Chart 11) SP5**

**/\*Top 10 Items – Sales\*/**

SELECT

pro.category, product\_name, round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1, 2

ORDER BY 3 DESC

LIMIT 10

;

Result:

This creates a chart showing the overall top 10 performing items in terms of Sales. I made this chart so that I can compare it to the Top 10 items in terms of profit chart.

**Chart 12) PP6**

**/\*Top 10 Items – Profit\*/**

SELECT

pro.category, product\_name, round(SUM(profit)+0) as profit

FROM orders o

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1, 2

ORDER BY 3 DESC

LIMIT 10

;

Result:

This creates a chart showing the overall top 10 performing items in terms of Profit. I made this chart so that I can compare it to the Top 10 items in terms of sales chart.

**Chart 13) SSEG12**

**/\*Sales for Each Product Category by Customer Segment\*/**

SELECT

pro.category, c.segment, round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN customers c

ON o.customer\_id = c.customer\_id

LEFT JOIN products pro

ON o.product\_id = pro.product\_id

GROUP BY 1, 2

ORDER BY 1

;

Result:

This creates a chart showing a comparison of the different Customer Segment sales revenue by the respective product categories that were purchased.

**Chart 14) SSEG10**

**/\*Top 25 Country Customer Segments In Terms of Sales ($)\*/**

SELECT

c.segment, r.country, round(SUM(sales)+0) as sales

FROM orders o

LEFT JOIN customers c

ON o.customer\_id = c.customer\_id

LEFT JOIN regions r

ON o.region\_id = r.region\_id

GROUP BY 1, 2

ORDER BY 3 DESC

LIMIT 25

;

Result:

This creates a chart showing the top 25 best performing country customer segments in terms of sales. I did this to find out from what country and what customer segment most of the sales revenue is coming from

**/\*Total Sales Figure Overall for the Superstore Business=245883674.76\*/**

SELECT SUM(sales)

FROM orders

;

Result:

Shows the sum of all the sales the business has made. I worked this out to use the figure in my presentation slides.

**Chart 15) RET2**

**/\*Count of Returns Reasons for Order Item Return Requests By Customer Segments\*/**

SELECT

reason\_returned, c.segment, COUNT(o.order\_id) as return\_reason\_count

FROM orders o

INNER JOIN customers c

ON o.customer\_id = c.customer\_id

INNER JOIN returns r

ON o.order\_id = r.order\_id

GROUP BY 1, 2

ORDER BY 1

;

Result:

This chart shows a Count of Return Reasons for Order Item Returns Requests by Customer Segments. I made this to find out which consumer segment makes the most order item return requests and for what reason.

**Chart 16) RET1**

**/\* Return Reason Count, Return Reason Count per Customer Segment and Total Order Item Requests Charts\*/**

**/\*Return Reason Count\*/**

SELECT

r.reason\_returned, COUNT(o.order\_id) as return\_reason\_count

FROM orders o

INNER JOIN customers c

ON o.customer\_id = c.customer\_id

INNER JOIN returns r

ON o.order\_id = r.order\_id

GROUP BY 1

ORDER BY 1

;

Result:

This creates a chart showing a count of the different returns reasons so that we can compare their values. I used it in my total percentage cell chart for order item return requests.

**/\*Total Return Item Requests = 51250\*/**

SELECT

COUNT(r.order\_id) as total\_return\_item\_requests

FROM orders o

INNER JOIN returns r

ON o.order\_id = r.order\_id

ORDER BY 1

;

Result:

This creates a figure showing a total count of all the return item requests. I used it in my total percentage cell chart for order item return requests.

**/\*Return Reason Count per Customer Segment\*/**

SELECT

reason\_returned, c.segment, COUNT(o.order\_id) as return\_reason\_count

FROM orders o

INNER JOIN customers c

ON o.customer\_id = c.customer\_id

INNER JOIN returns r

ON o.order\_id = r.order\_id

GROUP BY 1, 2

ORDER BY 1

;

Result:

This creates a chart showing the Return Reason count for each reason by each customer segment. I made this so that I can work out the total return requests value as a percentage of orders from each segment. I used it in my total percentage cell chart for order item return requests.

**/\*Total Return Requests Per Segment\*/**

SELECT

c.segment, COUNT(o.order\_id) as return\_reason\_count

FROM orders o

INNER JOIN customers c

ON o.customer\_id = c.customer\_id

INNER JOIN returns r

ON o.order\_id = r.order\_id

GROUP BY 1

ORDER BY 1

;

Result:

This creates a total return requests count for each consumer segment. I used it in my total percentage cell chart for order item return requests.

**/\*Total Order Item Requests Per Segment\*/**

SELECT c.segment,

COUNT(\*) AS order\_requests

FROM orders o

LEFT JOIN customers c

ON o.customer\_id = c.customer\_id

GROUP BY 1

ORDER BY 1

;

Result:

This creates a total order item requests per segment chart soe that I can compare the different segment orders to the total order figures. I used it in my total percentage cell chart for order item return requests.

**/\*Total Order Item Requests Count\*/**

SELECT

COUNT (order\_id)

FROM orders

;

Result:

This creates a total order item request figure. I used it in my total percentage cell chart for order item return requests.