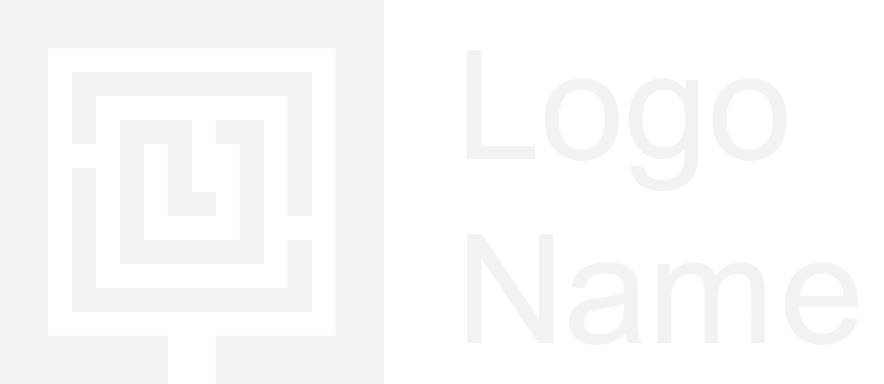
|  |
| --- |
| PROPOSAL AND MARKETING PLAN  Capstone Project: Data Exploration in SQL |
| UNICORN |



Table

Description automatically generated

To get started right away, just tap any placeholder text (such as this) and start typing to replace it with your own.

Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.

Find even more easy-to-use tools on the Insert tab, such as to add a hyperlink or insert a comment.

✏️

**Questions:**

1. **How many customers do we have in the data?**

SELECT Count (\*)

FROM customers

A picture containing background pattern

Description automatically generated

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

SELECT shipping\_city,SUM(order\_profits),EXTRACT(YEAR FROM order\_date) AS Year\_of\_profit

FROM Orders AS o

JOIN order\_details AS od

ON o.order\_id = od.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = '2015'

GROUP BY 1,3

ORDER BY 2 DESC

LIMIT 5;

Graphical user interface, application

Description automatically generated

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

SELECT shipping\_city,SUM(order\_profits),EXTRACT(YEAR FROM order\_date) AS Year\_of\_profit

FROM Orders AS o

JOIN order\_details AS od

ON o.order\_id = od.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = '2015'

GROUP BY 1,3

ORDER BY 2 DESC

LIMIT 5;

Graphical user interface, application

Description automatically generated

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

SELECT  COUNT (DISTINCT shipping\_city) AS different\_cities\_nos

FROM orders

Graphical user interface, application, Teams

Description automatically generated

1. **Show the total spent by customers from low to high.**

SELECT c.customer\_id,c.customer\_name,SUM(od.order\_sales) AS total\_spent

FROM order\_details AS od

JOIN orders AS o

ON o.order\_id = od.order\_id

JOIN customers AS c

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

GROUP BY 1,2

ORDER BY 3

LIMIT 10;

Graphical user interface

Description automatically generated

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

SELECT o.shipping\_city, SUM(od.order\_profits) AS most\_profit

FROM order\_details AS od

JOIN orders AS o

ON o.order\_id = od.order\_id

WHERE o.shipping\_state = 'Tennessee'

GROUP BY 1

ORDER BY SUM(od.order\_profits) DESC

LIMIT 5;

Table

Description automatically generated

1. **What’s the average annual profit for that city across all years?**

SELECT o.shipping\_city, ROUND(AVG(od.order\_profits),2) AS avg\_profit

FROM order\_details AS od

JOIN orders AS o

ON o.order\_id = od.order\_id

WHERE o.shipping\_city = 'Lebanon'

GROUP BY 1

ORDER BY AVG(od.order\_profits) DESC;

Graphical user interface, text, application

Description automatically generated

1. **What is the distribution of customer types in the data?**

SELECT DISTINCT customer\_segment AS customerè\_types, count(\*)

FROM customers

GROUP BY 1;

Graphical user interface, application, Teams

Description automatically generated

1. **What’s the most profitable product category on average in Iowa across all years?**

SELECT p.product\_category,o.shipping\_state,AVG(od.order\_profits) AS avg\_profitable\_category

FROM product AS p

JOIN order\_details AS od

ON p.product\_id = od.product\_id

JOIN orders AS o

ON o.order\_id = od.order\_id

WHERE o.shipping\_state = 'Iowa'

GROUP BY 1,2

ORDER BY AVG(od.order\_profits) DESC;

Graphical user interface, application

Description automatically generated

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

SELECT p.product\_category,p.product\_name,SUM(od.quantity) AS total\_quantity,

EXTRACT(YEAR FROM order\_date) AS Year\_of\_profit

FROM product AS p

JOIN order\_details AS od

ON p.product\_id = od.product\_id

JOIN orders AS o

ON o.order\_id = od.order\_id

WHERE EXTRACT (YEAR FROM o.order\_date) = '2016' AND p.product\_category = 'Furniture'

GROUP BY 1,2,4

ORDER BY SUM(od.quantity) DESC;

Graphical user interface

Description automatically generated with medium confidence

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

SELECT c.customer\_id,c.customer\_name, SUM((order\_sales\*order\_discount)/(1-order\_discount)) AS total\_discount

FROM customers AS c

JOIN orders AS o

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

JOIN order\_details AS od

ON o.order\_id = od.order\_id

GROUP BY 1,2

ORDER BY 3 DESC

LIMIT 5;

Graphical user interface, application

Description automatically generated

1. **How widely did monthly profits vary in 2018?**

SELECT EXTRACT(MONTH FROM o.order\_date) AS MONTH, SUM(od.order\_profits) AS monthly\_profits,LAG (SUM(od.order\_profits),1,0) OVER(ORDER BY EXTRACT(MONTH FROM o.order\_date))

AS lag, (SUM(od.order\_profits)-LAG (SUM(od.order\_profits),1,0) OVER(ORDER BY EXTRACT(MONTH FROM o.order\_date))) AS variance\_monthly

FROM orders AS o

JOIN order\_details AS od

ON o.order\_id = od.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = '2018'

GROUP BY 1

ORDER BY 1

Table

Description automatically generated with medium confidence

1. **Which order was the highest in 2015?**

SELECT p.product\_subcategory,od.order\_id,SUM(od.order\_sales) AS total\_sales

FROM orders AS o

JOIN order\_details AS od

ON o.order\_id = od.order\_id

JOIN product AS p

ON p.product\_id = od.product\_id

WHERE EXTRACT(YEAR FROM order\_date) = '2015'

GROUP BY 1,2

ORDER BY 3 DESC

LIMIT 1;

Chart

Description automatically generated

1. **What was the rank of each city in the East region in 2015?**

SELECT o.shipping\_city,

SUM(od.quantity) AS total\_quantity, rank() OVER (Order by SUM(od.quantity)DESC) AS city\_rank

FROM orders AS o

JOIN order\_details AS od

ON o.order\_id = od.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = '2015' AND o.shipping\_region = 'East'

GROUP BY 1

ORDER BY 3;

Graphical user interface, application

Description automatically generated

1. **Display customer names for customers who are in the segment ‘Consumer’ or ‘Corporate.’ How many customers are there in total?**

SELECT customer\_id,customer\_name,COUNT(\*)

FROM customers

WHERE customer\_segment IN ('Consumer', 'Corporate')

GROUP BY 1,2

Graphical user interface, application

Description automatically generated with medium confidence

SELECT COUNT(\*)

FROM customers

WHERE

customer\_segment IN ('Consumer', 'Corporate');

Graphical user interface, application

Description automatically generated

1. **Calculate the difference between the largest and smallest order quantities for product id ‘100.’**

select MAX(quantity) AS largest\_order,MIN(quantity) AS smallest\_order,

(MAX(quantity)-MIN(quantity)) AS difference

FROM order\_details

WHERE product\_id = '100'

Graphical user interface, application, Teams

Description automatically generated

1. **Calculate the percent of products that are within the category ‘Furniture.’**

WITH t1 AS

(SELECT product\_category,COUNT(\*) AS count\_category

FROM product

GROUP BY 1),

t2 AS

(SELECT COUNT(\*) AS total\_category

FROM product)

SELECT product\_category, ROUND(t1.count\_category\*1.0/t2.total\_category\*100,2) AS percentage\_category

FROM t1,t2

**OR**

select

(SELECT COUNT(\*) AS count\_category FROM product where product\_category = 'Furniture')\*1.0/

(SELECT COUNT(\*) AS count\_category FROM product) as Furniture\_percentage

Graphical user interface, application

Description automatically generated

1. **Display the number of duplicate products based on their product manufacturer.**

**Example: A product with an identical product manufacturer can be considered a duplicate.**

select product\_manufacturer,COUNT(\*) AS num\_of\_duplicates

FROM product

GROUP BY 1

HAVING count(\*)>1

Graphical user interface, application

Description automatically generated

1. **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\_subcategory) AS total\_products

FROM product

GROUP BY 1

ORDER BY 2 DESC;

Graphical user interface, application

Description automatically generated

SELECT product\_subcategory,COUNT(product\_subcategory) AS total\_products

FROM product

GROUP BY 1

ORDER BY 1;

Graphical user interface, application

Description automatically generated

1. **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 1

HAVING SUM(quantity) >= '100'

Graphical user interface, chart

Description automatically generated with medium confidence

**21 Join all database tables into one dataset that includes all unique columns and download it as a .csv file**

SELECT \*

FROM order\_details

NATURAL JOIN orders

NATURAL JOIN customers

NATURAL JOIN product