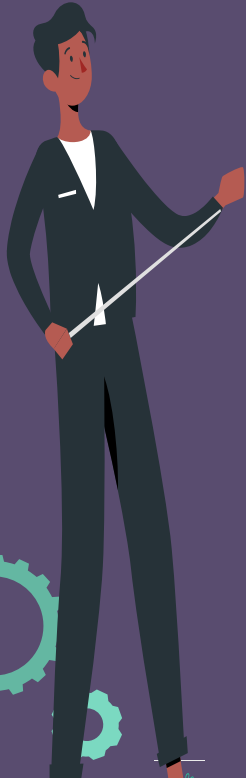




SALES PRESENTATION

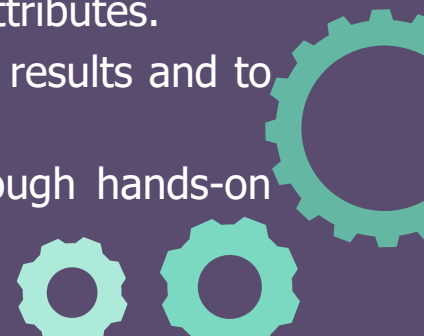
CONTENT

- PROJECT OVERVIEW
- PROJECT'S STEPS
- DATA ANALYSIS DOMAINS
- DATA DESCRIPTION
- SQL ANALYSIS (QUERIES)
- VISUALIZATION (POWER BI)
- KEY INSIGHTS
- REFERENCES





PROJECT OVERVIEW

- ❖ **OBJECTIVE:** The primary objective of this project is to analyze sales data to uncover insights and trends that can help improve business decisions and strategies. By leveraging SQL for data extraction and transformation, and Power BI for data visualization, we aim to provide a comprehensive view of sales performance, customer behavior, and operational efficiency.
 - ❖ **TOOLS:** For the project, Microsoft Excel, MySQL, & Power BI applications are used.
 - ❖ **APPROACH:** Employ data analysis, SQL Analysis and Power BI visualization of the data for better understanding.
 - ❖ **DATASET:** Make use of an extensive dataset that records pertinent attributes.
 - ❖ **METHODOLOGY:** Investigate different SQL queries to get accurate results and to visualize trends and patterns more clearly.
 - ❖ **OUTCOME:** Gain practical SQL & Power BI Visualization skills through hands-on experience in Sales Analysis.
- 



PROJECT STEPS

- ✓ **DATA EXTRACTION:** Extract the necessary dataset from KAGGLE Repository. Load the dataset in MySQL .Use SQL queries to get a overview of the dataset.
- ✓ **DATA CLEANING:** Address any missing, duplicate, data type conversion or inconsistent data. This includes writing SQL scripts to clean the data.
- ✓ **DATA TRANSFORMATION:** Transform the raw data into a format suitable for analysis. This includes normalizing data, creating new calculated fields.
- ✓ **SQL QUERIES:** Multiple SQL Query for better understanding of data and then take the data for visualizations.
- ✓ **CONNECT TO DATA SOURCES:** In Power BI, connect to the SQL database and import the prepared table.
- ✓ **DATA LOADING:** Load the data into Power BI, ensuring that it is refreshed regularly if the data is dynamic.
- ✓ **MEASURE CREATIONS:** Create DAX measures to calculate key performance indicators (KPIs) and other metrics required for the analysis such as columns that will be required in future calculations.
- ✓ **DESIGN DASHBOARDS:** Create interactive dashboards in Power BI. Include visualizations such as charts, graphs, and maps that effectively communicate the data.
- ✓ **INTERPRET DATA:** Analyze the visualizations to draw meaningful insights.
- ✓ **GENERATE RECCOMENDATIONS:** Based on the insights, generate actionable recommendations for the business.

DATA ANALYSE DOMAINS

- 1. SALES ANALYSIS:-** Sales analysis using SQL and Power BI is a robust approach for gaining insights into sales performance & customer behavior. SQL is utilized to query the sales database, extracting relevant data such as Unit Price, Quantity & product specifics. This data is then cleaned and aggregated to identify trends, patterns, and anomalies in sales, such as peak sales times, Percentage sales by City & Category. Once the data is prepared, Power BI is employed for visualization. This data is then processed to calculate key performance indicators (KPIs) such as Total Revenue, Average Order value, Total Product Sold, Total Orders.
- 2. PRODUCT PERFORMANCE ANALYSIS:-** Analyzing the best & worst Products using SQL & Power BI involves a systematic approach to uncovering performance metrics and identifying top-performing and underperforming products. SQL is first employed to query the sales database, retrieving data on individual seller transactions, sales volumes, and Total Revenue. Using Power BI, these KPIs are visualized through interactive dashboards and detailed reports, highlighting the best & worst performers. Visualizations such as Stacked bar charts enable stakeholders to quickly grasp which product excel in terms of revenue and customer satisfaction and which ones lag behind.
- 3. CUSTOMER ANALYSIS:-** With Power BI, businesses can unlock valuable insights into their customer base, driving data-driven decisions to enhance customer satisfaction and loyalty. Power BI's interactive visualizations can help businesses segment their customers based on purchase history, geographic location, and product preferences, allowing for the development of tailored promotions and loyalty programs. businesses can gain a deeper understanding of their customers, improve customer retention, and drive revenue growth.

DATA DESCRIPTION

The four tables that make up the dataset are listed below:-

Table 1:-“Sales_Orders”

- **OrderNumber** :A unique identifier for each sales order.
- **OrderDate**: The date the sales order was created.
- **Ship Date**: The expected date of shipment for the sales order.
- **Customer Name Index**: A unique identifier for the customers.
- **Channel**: The type of sales order, which can be either "Wholesale", "Distributor", or "Export".
- **CurrencyCode**: The currency in which the sales order was made (e.g., USD, NZD, AUD, EUR, GBP).
- **Warehouse Code**: A unique identifier for the warehouses.
- **Delivery Region Index**: A unique identifier for the cities the product is been sold.
- **Product Description Index**: A unique identifier for the product being sold (e.g., FLR025, AXW291).
- **Order Quantity**: The number of units ordered.
- **Unit Price**: The price of each unit of the product.
- **Total Unit Cost**: The cost of total unit of the products.
- **Total Revenue**: The total value of the sales order.

Table 2:-“Customers”

- **Customer Index**: A numerical identifier for each customer, ranging from 1 to 50. This is the primary key of the table, ensuring each customer has a unique ID.
- **Customer Names**: The official name of the customer. This column includes various naming conventions, including abbreviations (e.g., Ltd, Corp), punctuation (e.g., commas, periods), and capitalization variations




Table 3:- “Regions”

- **Index:** Unique identifier for each suburb.
- **Suburb:** Name of the suburb.
- **City:** Name of the city the suburb belongs to
- **postcode:** Postal code of the suburb
- **Longitude:** Geographic longitude coordinate of the suburb.
- **Latitude:** Geographic latitude coordinate of the suburb.
- **Full Address:** A complete address string including street, suburb, and city.

Table 4:- “Products”

- **Index:** A unique numerical identifier for each product. It appears to be a sequential integer starting from 1.
- **Product_Name:** The name of each product.

SQL ANALYSIS (QUERIES)

- This dataset contains detailed information about sales, including specifics about the customers, quantities, pricing, dates, regions, products & categorization details.
- The final results are displayed below after the raw data is fed into SQL tables such as regions, customers, product & sales orders.

Table 1:-“Sales_Orders”

OrderNumber	OrderDate	Ship_Date	Customer_Name	Channel	Currency	Warehouse_Code	Delivery_Region_Index	Product	Order_Quantity	Unit_Price	Total_Unit_Cost	Total_Revenue
SO - 000225	2017-01-01	2017-01-13	28	Wholesale	NZD	AXW291	71	11	6	2499.1	1824.34	14994.6
SO - 0002911	2017-01-02	2017-01-15	37	Wholesale	EUR	AXW291	2	2	6	911.2	665.18	5467.2
SO - 0003378	2017-01-01	2017-01-06	7	Distributor	NZD	AXW291	54	7	11	2351.7	1269.92	25868.7
SO - 0003527	2017-01-02	2017-01-19	8	Wholesale	USD	AXW291	77	9	11	1701.8	1429.51	18719.8
SO - 0003901	2017-01-01	2017-01-05	12	Wholesale	NZD	AXW291	58	13	5	1728.6	1019.87	8643
SO - 0004792	2017-01-02	2017-01-13	14	Wholesale	NZD	AXW291	82	11	6	167.5	83.75	1005
SO - 0005126	2017-01-01	2017-01-17	5	Wholesale	USD	AXW291	29	7	6	978.2	684.74	5869.2
SO - 0005414	2017-01-02	2017-01-05	47	Wholesale	USD	AXW291	71	7	7	1835.8	1413.57	12850.6
SO - 0005614	2017-01-01	2017-01-07	27	Export	NZD	AXW291	31	6	7	2338.3	1028.85	16368.1
SO - 0005781	2017-01-01	2017-01-15	44	Wholesale	GBP	AXW291	78	9	8	2291.4	1260.27	18331.2

Table 2:-“Customers”

Customer_Index	Customer_Names
1	Avon Corp
2	WakeFern
3	Elorac, Corp
4	ETUDE Ltd
5	Procter Corp
6	PEDIFIX, Corp
7	New Ltd
8	Medsep Group
9	Ei
10	21st Ltd

Table 4:- “Products”

Index	Product_Name
1	Product 1
2	Product 2
3	Product 3
4	Product 4
5	Product 5
6	Product 6
7	Product 7
8	Product 8
9	Product 9
10	Product 10

Table 3:- “Regions”

	Index	Suburb	City	postcode	Longitude	Latitude	Full_Address
▶	1	Freemans Bay	Auckland	1011	174.748652	-36.855732	Picton Street, Freemans Bay, Auckland
	2	Nightcaps	Southland	9630	168.028823	-45.9703	Lyne Street, Nightcaps, Southland
	3	Northcote	North Shore	627	174.755505	-36.804712	McBreen Avenue, Northcote, North Shore
	4	Bay View	Napier	4104	176.871662	-39.440389	Ferguson Street South, Bay View, Napier
	5	Parklands	Christchurch	8083	172.705997	-43.472699	Forest Drive, Parklands, Christchurch
	6	Hamilton East	Hamilton	3216	175.305496	-37.781657	Rochford Court, Hamilton, Hamilton
	7	Te Kuiti	Waitomo	3910	175.163086	-38.330973	Alexandra Street, Te Kuiti, Waitomo
	8	Opaheke	Papakura	2113	174.947631	-37.077401	Rhonda Place, Opaheke, Papakura
	9	North East Valley	Dunedin	9010	170.527081	-45.846837	Selwyn Street, North East Valley, Dunedin
	10	Whangamata	Thames-Coromandel	3620	175.883257	-37.219172	Pipi Road, Whangamata, Thames-Coromandel

DATA CLEANING & TRANSFORMING

Once the data is entered into SQL, we see that part of the data is distorted. Specifically, the date columns are not formatted correctly, and the data type is incorrect. The SQL query for this is provided below:-

```
-- Orderdate
UPDATE sales_orders
SET OrderDate = DATE_FORMAT(STR_TO_DATE(OrderDate, '%d-%m-%Y'), '%Y-%m-%d');

alter table sales_orders modify column OrderDate Date;

-- ship date
UPDATE sales_orders
SET Ship_Date = DATE_FORMAT(STR_TO_DATE(Ship_Date, '%d-%m-%Y'), '%Y-%m-%d');

alter table sales_orders modify column Ship_Date Date;
```

Second, it becomes clear that three columns are dealing with the data type issue. Unit pricing, total revenue, and total unit cost are the three columns. Since the columns contain special characters, we must first figure out how to change the character before changing the data type. The following is the SQL code:-

```
-- Unit Price
UPDATE Sales_orders
SET Unit_Price = REPLACE(Unit_Price, ',', '');

ALTER TABLE sales_orders MODIFY COLUMN Unit_Price float;

-- Total Unit cost
UPDATE Sales_orders
SET Total_Unit_Cost = REPLACE(Total_Unit_Cost, ',', '');

ALTER TABLE sales_orders MODIFY COLUMN Total_Unit_Cost float;

-- Total Revenue
UPDATE Sales_orders
SET Total_Revenue = REPLACE(Total_Revenue, ',', '');

ALTER TABLE sales_orders MODIFY COLUMN Total_Revenue float;
```

KEY PERFORMANCE INDICATOR'S



- 1) **Total Cost:** It refers to the complete expense incurred in the production and delivery of goods or services.
- 2) **Total Orders:** The total number of orders placed.
- 3) **Average Order value:** The average amount spent per order, calculated by dividing the total revenue by the total number of orders.
- 4) **Total Revenue:** The sum of the total price of all product orders.
- 5) **Total Profit:** By subtracting the total cost from the total revenue, the profit is computed.
- 6) **Current Year Sales:** To calculate current year sales, sum the revenue from all sales transactions made within the current year.
- 7) **Previous Year Sales:** To calculate previous year sales, sum the total revenue from all sales transactions made during the entire previous fiscal year.
- 8) **YOY%:** Year-over-Year percentage change is calculated as $(\text{Current Year's Value} - \text{Previous Year's Value}) / \text{Previous Year's Value} \times 100\%$
- 9) **Total Products Sold:** The Sum Of the quantities of all products sold.
- 10) **Average Product Per Order:** The average number of products sold per order, calculated by dividing the total number of products sold by the total number of orders.

```

-- Total Cost
SELECT round(sum(Order_Quantity * Total_Unit_Cost),2) as Total_Cost from sales_orders;

-- Total Orders
select count(distinct(OrderNumber)) as Total_Orders from sales_orders;

-- Average Order value
SELECT round(SUM(Total_Revenue) / COUNT(DISTINCT OrderNumber),2) AS Average_Order_Value FROM sales_orders;

-- Total Revenue
select round(sum(Total_Revenue),2) as Total_Revenue from sales_orders;

-- Total Profit
Select round(sum(Total_Revenue -(Order_Quantity * Total_Unit_Cost)),2) as Total_Profit from Sales_Orders;

-- Current Year Sales
WITH CurrentYearSales AS (SELECT COALESCE(round(SUM(Total_Revenue), 0),0) AS CY_Sales FROM sales_orders WHERE YEAR(OrderDate) = '2019'),
-- Previous Year Sales
PreviousYearSales AS (SELECT COALESCE(round(SUM(Total_Revenue), 0),0) AS PY_Sales FROM sales_orders WHERE year(OrderDate) = '2018')
-- YoY Percentage
SELECT CY_Sales,PY_Sales, round((CY_Sales - PY_Sales) / PY_Sales * 100,2) as YoY_Percentage FROM CurrentYearSales, PreviousYearSales;

-- Total Products Sold =done
Select sum(Order_Quantity) as Total_Products_Sold from sales_orders;

-- Avg Product Per Order
select Round(sum(Order_Quantity)/COUNT(DISTINCT OrderNumber),2) AS Avg_Product_per_Order from sales_orders;

```

OUTPUTS

Total_Cost
96784028.16

Total_Orders
7991

Average_Order_Value
19343.4

Total_Revenue
154573140.57

Total_Profit
57789112.41

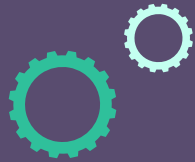
OUTPUT

CY_Sales	PY_Sales	YoY_Percentage
48528944	53463662	-9.23

OUTPUTS

Total_Products_Sold
67579

Avg_Product_per_Order
8.46



TOTAL ORDERS BY WAREHOUSE CODE

- **Total orders by Warehouse Code:-** Total orders by warehouse code refer to the aggregate number of orders processed and fulfilled by each specific warehouse within a distribution network. This metric is crucial for understanding the operational efficiency, capacity utilization, and performance of individual warehouses. By analyzing the total orders by warehouse code, businesses can identify which warehouses are handling the highest volumes, detect potential bottlenecks or inefficiencies, and make informed decisions about inventory distribution and resource allocation.

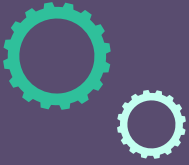
```
SELECT s.Warehouse_Code, COUNT(s.OrderNumber) AS Total_Orders
FROM Sales_Orders s
GROUP BY s.Warehouse_Code
order by Total_Orders desc;
```

OUTPUT

Warehouse_Code	Total_Orders
AXW291	3756
GUT930	1850
NXH382	1569
FLR025	816



MONTHLY AVERAGE REVENUE



- **Average Total Revenue By Month:-** Average Total Revenue by Month refers to the mean amount of money generated from sales over each month within a specified period, typically a year. This metric helps businesses understand their revenue patterns, identify trends, and make informed decisions.

```
SELECT DATE_FORMAT(OrderDate, '%b') as Month, ROUND(AVG(Total_Revenue), 2) as Avg_Total_Revenue
FROM sales_orders
GROUP BY DATE_FORMAT(OrderDate, '%b'), MONTH(OrderDate)
ORDER BY MONTH(OrderDate);
```

OUTPUT

Month	Avg_Total_Revenue
Jan	19240.95
Feb	18931.35
Mar	20033.23
Apr	19305.8
May	19687.86
Jun	18886.9
Jul	18538.17
Aug	19133.23
Sep	19267
Oct	18776.18



TOTAL PROFIT BY CITY

- **Total Profit By City:-** "Total Profit by City" refers to the aggregate profit generated from business operations within a specific city. Analyzing profit by city allows businesses to identify which locations are most profitable, enabling more informed strategic decisions regarding resource allocation, marketing efforts, and potential expansion or contraction.

```
SELECT r.city,round(sum(Total_Revenue -(Order_Quantity * Total_Unit_Cost)),2) AS Total_Profit FROM sales_orders AS s
INNER JOIN regions AS r
ON s.Delivery_Region_Index= r.Index
GROUP BY r.city
ORDER BY Total_Profit DESC;
```

OUTPUT

city	Total_Profit
Christchurch	4333779.51
Hamilton	4304665.88
Waitakere	4180236.22
Manukau	3644071.59
Auckland	2853433.75
Napier	2846350.87
Palmerston North	2638189.85
South Waikato	2467584.26
Whangarei	1812421.22
North Shore	1763762.27

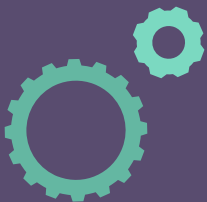
TOTAL PRODUCTS SOLD BY CHANNEL & CURRENCY CODE

- **The Amount of Products Sold by Currency Code & Channel:-** "Total Products Sold by Channel & Currency Code" is a detailed metric that provides insights into the quantity of products sold through different sales channels and specifies the currency used in the transactions. This breakdown helps businesses understand their sales performance across various channels, identify which channels are the most effective, and analyze revenue in the context of different currencies.

```
SELECT Channel,currency_code,SUM(Order_Quantity) AS Total_Products_Sold FROM Sales_orders
GROUP BY channel,currency_code
ORDER BY Total_Products_Sold desc;
```

OUTPUT

Channel	currency_code	Total_Products_Sold
Wholesale	NZD	13880
Wholesale	USD	10526
Distributor	NZD	8001
Distributor	USD	6297
Wholesale	AUD	5946
Export	NZD	3742
Distributor	AUD	3282
Wholesale	GBP	3051
Wholesale	EUR	2928
Export	USD	2830



PERCENTAGE REVENUE CALCULATION

- **% Total Revenue By City:-** The percentage of total revenue by city represents the proportion of overall revenue that each city contributes to a business's total income. It provides valuable insights into geographic revenue distribution, highlighting which cities are major revenue drivers versus those contributing less to the overall income. This information helps businesses focus resources on high-performing markets, tailor marketing strategies to local preferences, and optimize operational efficiencies based on regional performance trends.

```
SELECT r.city, ROUND(SUM(s.Total_Revenue) * 100.0 / (SELECT SUM(Total_Revenue) FROM Sales_Orders), 2) AS Percentage_Revenue
FROM Sales_Orders AS s
INNER JOIN regions AS r
ON s.Delivery_Region_Index = r.Index
GROUP BY r.city;
```

OUTPUT

city	Percentage_Revenue
Queenstown-Lakes	0.9
Christchurch	7.46
Hamilton	7.43
Timaru	1.88
Waitakere	7.24
Manukau	6.31
Southland	2.01
South Waikato	4.26
Gisborne	1.1
Wanganui	2.14

AVERAGE COST & REVENUE BY PRODUCT NAME



- **Avg Cost & Avg Revenue by Product Name:-** Analyzing average cost and revenue by product name involves calculating the average financial metrics associated with each specific product or service offered by a business. Average Cost helps businesses understand the per-unit cost efficiency of each product line, aiding in pricing strategies and profitability assessments. On the other hand, average revenue per product name reflects the average income generated from selling each unit of a specific product. By analyzing these averages, businesses can pinpoint strengths and weaknesses across their product portfolio, guiding strategic decisions aimed at enhancing profitability and market competitiveness.

```
SELECT p.Product_Name,ROUND(AVG(Total_Revenue),0) as Avg_Total_Revenue,ROUND(AVG(Order_Quantity * Total_Unit_Cost),0) AS Avg_Total_Cost
FROM sales_orders AS s
INNER JOIN Products AS p
ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name
ORDER BY Avg_Total_Cost DESC;
```

OUTPUT

Product_Name	Avg_Total_Revenue	Avg_Total_Cost
Product 12	19931	12803
Product 8	19489	12356
Product 6	19202	12310
Product 3	19045	12303
Product 14	20551	12300
Product 5	19700	12241
Product 7	19360	12172
Product 11	19418	12169
Product 13	18720	12148
Product 1	19398	12095



BEST/WORST PRODUCTS BY TOTAL REVENUE

- **Top 5 Products by Revenue:** This SQL code shows the Top 5 best products based on Total Revenue. This numbers will help us identify the most popular products options.
- **Bottom 5 Products by Revenue:** This SQL code shows the Bottom 5 products based on Total Revenue. This numbers will help us to identify the not so popular products options.

```
SELECT p.Product_Name, round(SUM(s.Total_Revenue),0) AS Total_Revenue
FROM Sales_Orders s
JOIN Products p ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name
ORDER BY Total_Revenue DESC
limit 5;
```


Product_Name	Total_Revenue
Product 7	25710124
Product 1	25489299
Product 2	22846497
Product 11	20621809
Product 5	17021156



OUTPUTS

```
SELECT p.Product_Name, ROUND(SUM(s.Total_Revenue),0) AS Total_Revenue
FROM Sales_Orders s
JOIN Products p ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name
ORDER BY Total_Revenue ASC
LIMIT 5;
```

Product_Name	Total_Revenue
Product 4	2857289
Product 3	3066282
Product 12	3109289
Product 10	3114006
Product 14	3144263



PRODUCTS PERFORMANCE

- **Products Performance By Channel:-** Product performance by channel refers to the assessment of how well a product is selling across different distribution channels it also shows the Total products sold through each channel. This analysis helps businesses optimize their marketing strategies, allocate resources efficiently, and tailor their product offerings to better meet consumer demand across various channels, ultimately enhancing overall profitability and market competitiveness.
- **Products Performance By Currency Code:-** Product performance by currency code assesses how well a particular product or range of products is performing in different currencies. It then shows the products sold through each currencies. This analysis helps businesses identify strong-performing products in specific currencies, optimize pricing strategies based on currency fluctuations, and make informed decisions regarding inventory management, marketing campaigns, and expansion opportunities in targeted geographical areas.

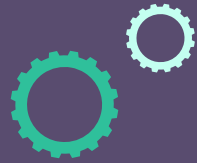
```
SELECT p.Product_Name, s.Channel, SUM(s.Order_Quantity) AS Total_Products_Sold
FROM Sales_Orders s
INNER JOIN Products p ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name, s.Channel
ORDER BY Total_Products_Sold DESC;
```

Product_Name	Channel	Total_Products_Sold
Product 7	Wholesale	6174
Product 1	Wholesale	5891
Product 2	Wholesale	5271
Product 11	Wholesale	4868
Product 5	Wholesale	3827

```
SELECT p.Product_Name, s.currency_code, SUM(s.Order_Quantity) AS Total_Products_Sold
FROM Sales_Orders s
INNER JOIN Products p ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name, s.currency_code
ORDER BY Total_Products_Sold DESC;
```

Product_Name	currency_code	Total_Products_Sold
Product 7	NZD	4306
Product 1	NZD	4056
Product 2	NZD	3801
Product 11	NZD	3377
Product 1	USD	3241

OUTPUTS



BEST/WORST PRODUCTS BY TOTAL ORDERS

- **Top 5 Products by Orders:** This SQL code shows the Top 5 best products based on Total Orders. This numbers will help us identify the most popular products options.
- **Bottom 5 Products by Orders:** This SQL code shows the Bottom 5 products based on Total Orders. This numbers will help us to identify the not so popular products options.

```
SELECT p.Product_Name,count(DISTINCT(OrderNumber)) AS Total_Orders FROM
sales_orders AS s
INNER JOIN products AS p
ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name
ORDER BY Total_Orders DESC
LIMIT 5;
```

Product_Name	Total_Orders
Product 7	1328
Product 1	1314
Product 2	1171
Product 11	1062
Product 5	864

```
SELECT p.Product_Name,count(DISTINCT(OrderNumber)) AS Total_Orders FROM
sales_orders AS s
INNER JOIN products AS p
ON s.Product_Description_Index = p.Index
GROUP BY p.Product_Name
ORDER BY Total_Orders ASC
LIMIT 5;
```

Product_Name	Total_Orders
Product 14	153
Product 12	156
Product 3	161
Product 10	163
Product 4	163

OUTPUTS



CUSTOMER BEHAVIOUR

- **Customer Purchase Pattern:-** Understanding customer behavior involves analyzing factors such as demographics, psychographics, buying patterns, and preferences to predict and influence future purchasing decisions. With the help of this SQL query, we can determine which customers buy more products, how much is shipped to them, and how much each client contributes to our bottom line.

```
SELECT c.customer_Names,count(DISTINCT(OrderNumber)) AS Total_Orders , sum(Order_Quantity) AS Total_Products_Sold,  
ROUND(SUM(Total_Revenue -(Order_Quantity * Total_Unit_Cost)),0) AS Total_Profit  
FROM sales_orders s  
INNER JOIN customers c  
ON c.Customer_Index = s.Customer_Name_Index  
GROUP BY c.customer_Names  
ORDER BY Total_Profit DESC;
```

OUTPUT

customer_Names	Total_Orders	Total_Products_Sold	Total_Profit
Medline	210	1754	1506669
Pure Group	175	1547	1402517
OUR Ltd	176	1492	1398369
Apollo Ltd	178	1553	1392406
OHTA'S Corp	173	1504	1355517

TOP CUSTOMERS BY REVENUE

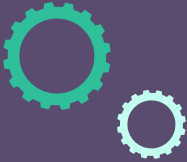
- **Top 10 Customers By Total Revenue:-** It refers to a business strategy or analysis that identifies and ranks the ten customers who contribute the highest total revenue to a company over a specified period, typically a fiscal year. This analysis helps businesses understand which customers are the most significant in terms of financial impact, allowing them to prioritize resources such as marketing efforts, customer service enhancements, and personalized engagement strategies towards these key accounts. By focusing on the top revenue-generating customers, companies can strengthen relationships, improve retention rates, and potentially uncover opportunities for further growth and profitability. This approach also aids in strategic decision-making, such as pricing adjustments or product development tailored to meet the needs and preferences of these valuable customers.

```
SELECT c.Customer_Names, round(SUM(s.Total_Revenue),0) AS Total_Revenue
FROM Sales_Orders s
INNER JOIN Customers c ON s.Customer_Name_Index= c.Customer_Index
GROUP BY c.Customer_Names
ORDER BY Total_Revenue DESC
Limit 10;
```

OUTPUT

Customer_Names	Total_Revenue
Medline	4081486
Pure Group	3821593
OUR Ltd	3676035
Eminence Corp	3643427
Apotheca, Ltd	3608493
Apollo Ltd	3600359
OHTA'S Corp	3533801
Ole Group	3462882
Victory Ltd	3407814
Ohio	3347682

UNIT COST BY CUSTOMERS



- **Total unit cost by customers:-** It refers to the combined expenses incurred by a business to produce and deliver a single unit of product, as perceived and influenced by the customer. Understanding total unit cost is crucial for businesses as it directly impacts pricing strategies and profitability. By analyzing these costs from the customer's perspective, companies can ensure competitive pricing while maintaining quality and profitability standards, ultimately enhancing customer satisfaction and optimizing financial performance.

```
SELECT c.Customer_Names, ROUND(SUM(s.Total_Unit_Cost),2) AS Total_Unit_Cost
FROM Sales_Orders s
INNER JOIN Customers c ON s.Customer_Name_Index = c.Customer_Index
GROUP BY c.Customer_Names
ORDER BY Total_Unit_Cost DESC;
```

OUTPUT

Customer_Names	Total_Unit_Cost
Medline	308855.59
Eminence Corp	279896.2
Apotheca, Ltd	276517.25
Pure Group	268695.78
OUR Ltd	268167.06
Ole Group	259739.71



Connecting SQL Server to Power BI is an essential step in leveraging the powerful data visualization capabilities of Power BI to analyze sales data stored in MySQL server. This connection allows for the seamless integration of robust data management with interactive reporting, provide valuable insights into Sales trend, product performance and customer's product preferences.

Benefits:

- **Real-time Data Analysis:** With DirectQuery, Power BI can query data directly from SQL Server, providing up-to-date insights without the need for frequent data imports.
- **Comprehensive Reporting:** By combining queries and views from SQL Server, Power BI enables the creation of detailed and comprehensive reports that cover various aspects of sales.
- **Interactive Dashboards:** Power BI's interactive features allow users to filter data dynamically, drill down into details, and uncover hidden trends in sales data.
- **Improved Decision-Making:** The insights derived from Power BI reports can inform strategic decisions such as inventory management, marketing campaigns, & customer engagement strategies.

In summary, the integration of SQL Server with Power BI for sales analysis empowers businesses to transform raw sales data into actionable insights, driving better decision-making and enhancing overall business performance.

DAX FORMULA'S

We can now construct new measures & columns using the DAX formulas after connecting SQL to Power Bi. Data Analysis Expressions (DAX) is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel. DAX formulas include functions, operators, and values to perform advanced calculations and queries on data in related tables and columns in tabular data models.

In accordance with the demand, we have computed new measures and columns. Below are the dax formulas for the new measures & columns:-

NEW MEASURES

1. Average Order Value =

```
ROUND(SUM('capstone1 sales_orders'[Total_Revenue]) /  
DISTINCTCOUNT('capstone1 sales_orders'[OrderNumber]),2)
```

2. Total Orders =

```
DISTINCTCOUNT('capstone1 sales_orders'[OrderNumber])
```

3. Total Products sold =

```
SUM('capstone1 sales_orders'[Order_Quantity])
```

4. Avg Products Per Order =

```
'capstone1 sales_orders'[Total_Products_Sold]/  
'capstone1 sales_orders'[Total_Orders]
```

5. SumSales =

```
SUMX('capstone1 sales_orders',  
'capstone1 sales_orders'[Order_Quantity] * 'capstone1 sales_orders'[Unit_Price])
```

6. CY Sales =

```
TOTALYTD('capstone1 sales_orders'[SumSales],  
'capstone1 sales_orders'[OrderDate].[Date])
```

7. PY Sales =

```
CALCULATE('capstone1 sales_orders'[SumSales],  
PREVIOUSYEAR(DATESYTD('capstone1 sales_orders'[OrderDate].[Date]))))
```

8. YOY% =

```
DIVIDE('capstone1 sales_orders'[CY_Sales] - 'capstone1 sales_orders'[PY_sales],  
'capstone1 sales_orders'[PY_sales])
```

NEW COLUMNS

1. Total Cost =

```
'capstone1 sales_orders'[Order_Quantity]*  
'capstone1 sales_orders'[Total_Unit_Cost]
```

2. Total Profit =

```
'capstone1 sales_orders'[Total_Revenue]-'capstone1 sales_orders'[Total Cost]
```

Now that all the metrics and columns have been created, we can move on with creating interactive dashboards. You can view the reports in the next three slides.

REPORT – 1: SALES REPORT



SALES REPORT

Jan/17 - Dec/19

Customer Names

All

Suburb

All

Post Code

All

Year

All

96.78M

Total Cost

7991

Total Orders

19,343.40

Average Order Value

154.57M

Total Revenue

57.79M

Total Profit

48.53M

CY Sales

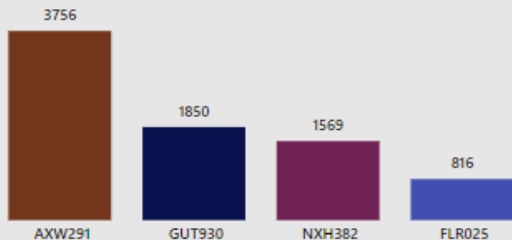
53.46M

PY sales

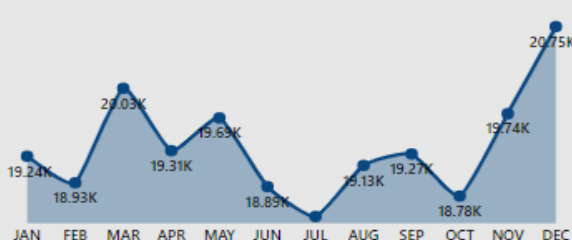
-9.23%

YOY%

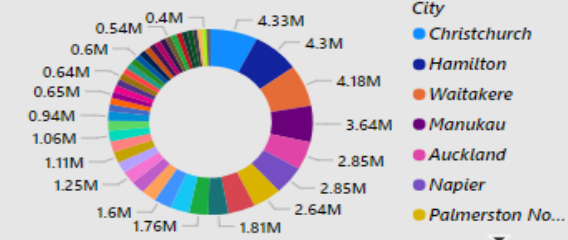
Total Orders By Warehouse Code



Avg Total Revenue by Month

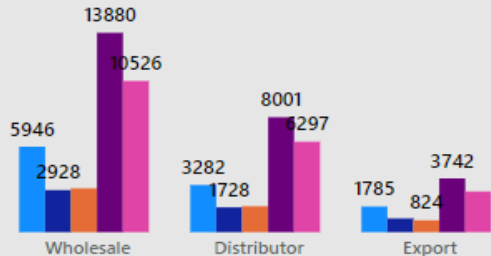


Total Profit By City



Total Products Sold By Channel & Currency Code

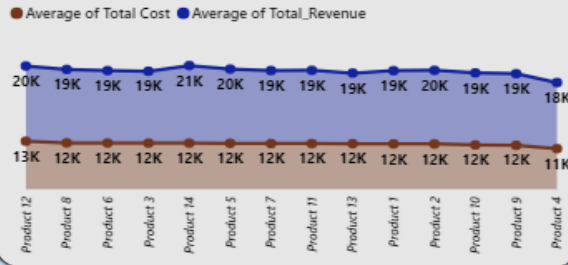
Currency Code: AUD, EUR, GBP, NZD, USD



%Total Revenue By City



Avg Total Cost & Avg Total Revenue By Product Name



REPORT – 2: PRODUCT PERFORMANCE REPORT



City

All

Customer Names

All

PRODUCTS PERFORMANCE REPORT

Jan/17 - Dec/19

Suburb

All

Warehouse Code

All

7991

Total Orders

67579

Total Products Sold

8.46

Avg Products Per Order

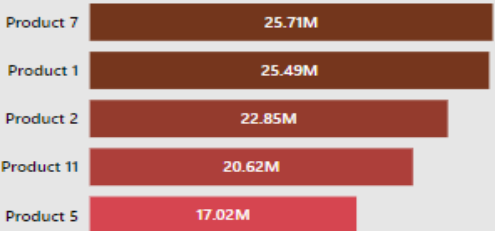
19,343.40

Average Order Value

154.57M

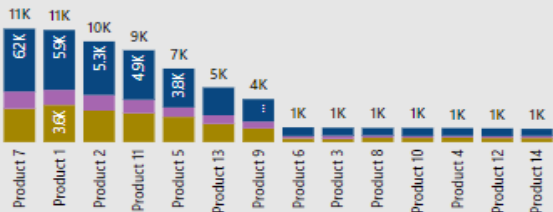
Total Revenue

Top 5 Products By Revenue

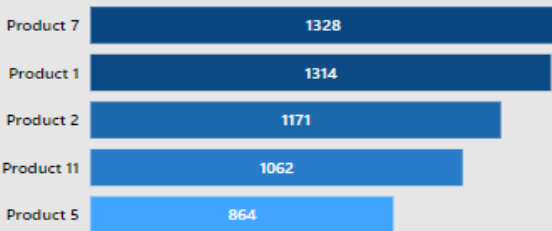


Products Performance By Channel

Channel ● Distributor ● Export ● Wholesale



Top 5 Products By Total Orders

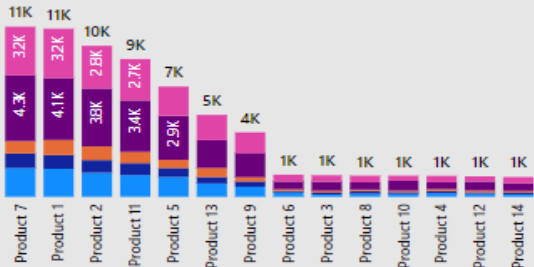


Bottom 5 Products By Revenue

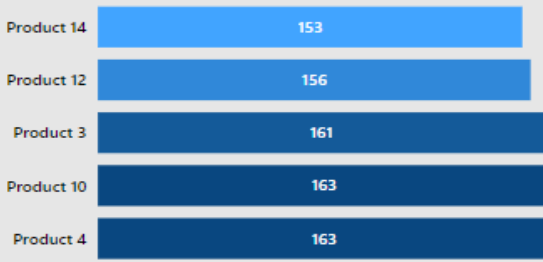


Products Performance By Currency Code

Currency Code ● AUD ● EUR ● GBP ● NZD ● USD



Bottom 5 Products By Total Orders



REPORT – 3: CUSTOMER ANALYSIS



CUSTOMER ANALYSIS

Jan/17-Jan/19

FILTERS

Year

All

Month

All

City

All

Channel

☐ Distributor

☐ Export

☐ Wholesale

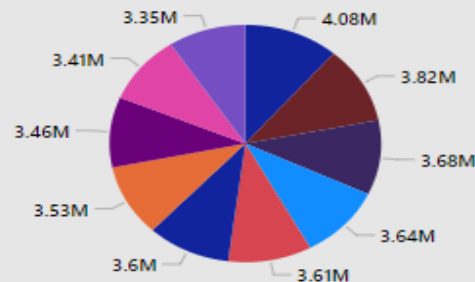
WarehouseCode

All

Customer's Purchase Pattern

Customer_Names	Total Orders	Total Products Sold	Total Profit
Medline	210	1754	15,06,669.31
Pure Group	175	1547	14,02,517.29
OUR Ltd	176	1492	13,98,368.61
Apollo Ltd	178	1553	13,92,406.16
OHTA'S Corp	173	1504	13,55,516.68
Apotheca, Ltd	179	1486	13,37,765.81
Eminence Corp	186	1572	13,32,935.50
Ohio	164	1376	13,18,352.27
ETUDE Ltd	167	1435	13,14,732.71
Linde	157	1398	12,86,557.86
Victory Ltd	176	1515	12,63,720.89
Ei	171	1467	12,61,536.30
Prasco Group	162	1364	12,31,436.19
Ole Group	171	1478	12,27,354.35
Rochester Ltd	142	1242	12,15,109.37
Weimei Corp	161	1346	12,12,960.32
Sundial	163	1360	12,09,882.20

Top 10 Customers By Total Revenue



Customer Names

● Medline

● Pure Group

● OUR Ltd

● Eminence Corp

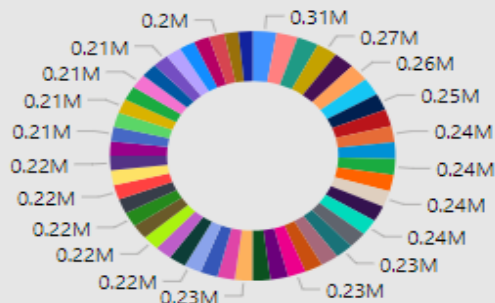
● Apotheca, Ltd

● Apollo Ltd

● OHTA'S Corp

● Ole Group

Total Unit Cost By Customers



Customer Names

● 12

● 18

● 29

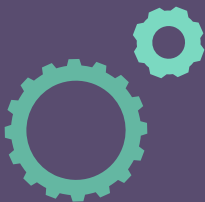
● 17

● 34

● 13

● 11

● 32



KEY INSIGHTS

1. KPI'S REQUIREMENT:

FROM REPORT
1: SALES REPORT

96.78M

Total Cost

7991

Total Orders

19,343.40

Average Order Value

154.57M

Total Revenue

57.79M

Total Profit

48.53M

CY Sales

53.46M

PY sales

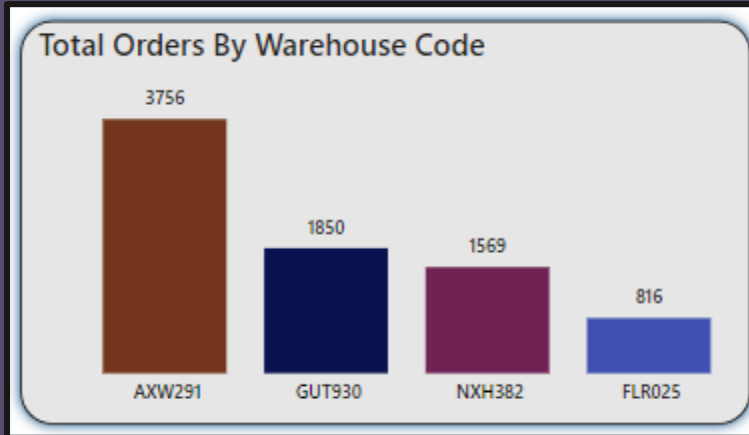
-9.23%

YOY%

KEY PERFORMANCE INDICATOR:-

- **Total Cost** : The Expenses incurred in the production & delivery of goods are **96.78M**.
- **Total Orders**: The total number of orders placed is **7,991**.
- **Average Order value**: The average amount spent per order is **19,343.40**.
- **Total Revenue**: The sum of the total price of all products orders is **154.57M**.
- **Total Profit**: The actual profit is **57.79M**.
- **CY Sales**: The Sum of the revenue from all sales transactions made within **2019** is **48.53M**.
- **PY Sales**: The Sum of the revenue from all sales transactions made within **2018** is **53.46M**.
- **YOY%**: The difference between current year and previous year is **-9.23%**.

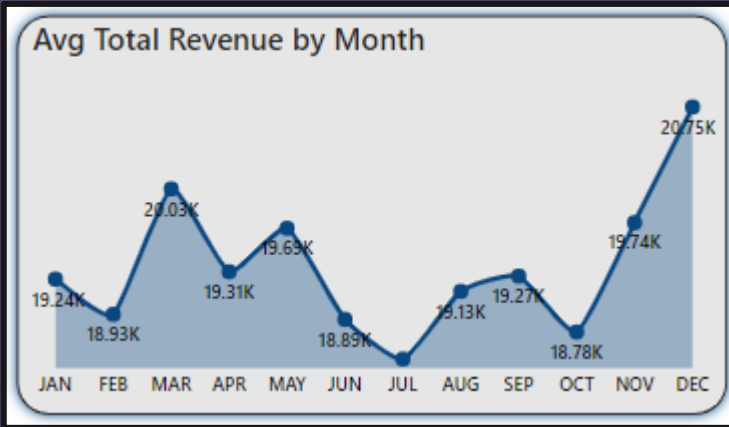
2. STACKED COLUMN CHART: TOTAL ORDERS BY WAREHOUSE CODE



The purpose of this stacked column chart is to display total orders by warehouse code. It is evident from the chart that **AXW291**, with **3,756** orders, is the Warehouse that proceed & fulfill the highest orders, followed by GUT930(with **1850** orders) ,NXH382(with **1569** orders) . The FLR025(with **816** orders) is the warehouse that fulfill the least orders to the customers.



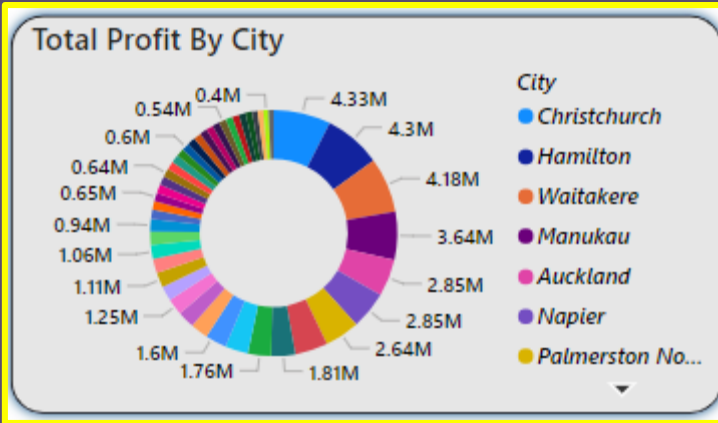
3. AREA CHART: AVERAGE TOTAL REVENUE BY MONTH



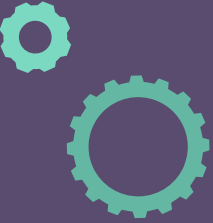
This Area Chart was created to display the Average Total revenue for each month. The graph makes it evident that the top three months of the year are **December** with average revenue of **20.75K**, **March** with **20.03k** average Revenue & November with 19.74k. **February** & **October** saw the least average revenue, at 18.93K & 18.78K respectively. This indicates that the majority of revenue is generated in **December** and then **March**.



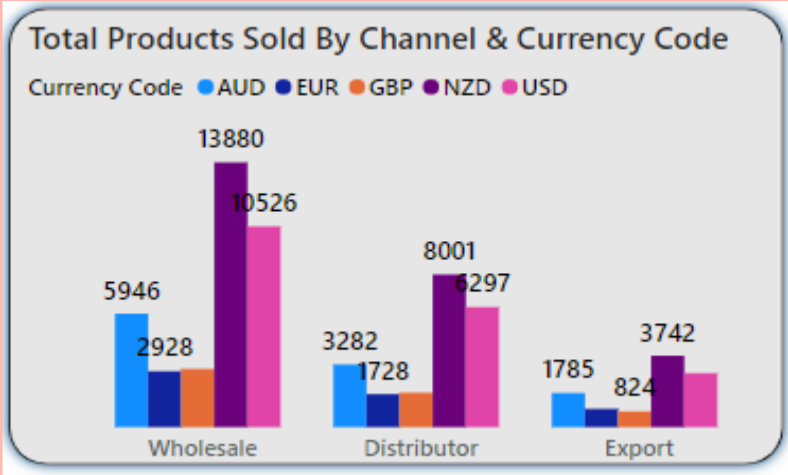
4. DONUT CHART: TOTAL PROFIT BY CITY



Donut charts is made to display how profit is made among different cities. The graphs clearly show how profits are made throughout the cities; at **4.33 million**, **Christchurch** is the city that generates the most profits. **Waitakere** and **Hamilton** are in third and second place, respectively, with 4.18 and 4.3 million. In this manner, we can concentrate our efforts on cities that generate lower profits.



5. CLUSTERED COLUMN CHART: TOTAL PRODUCTS SOLD BY CHANNEL & CURRENCY CODE



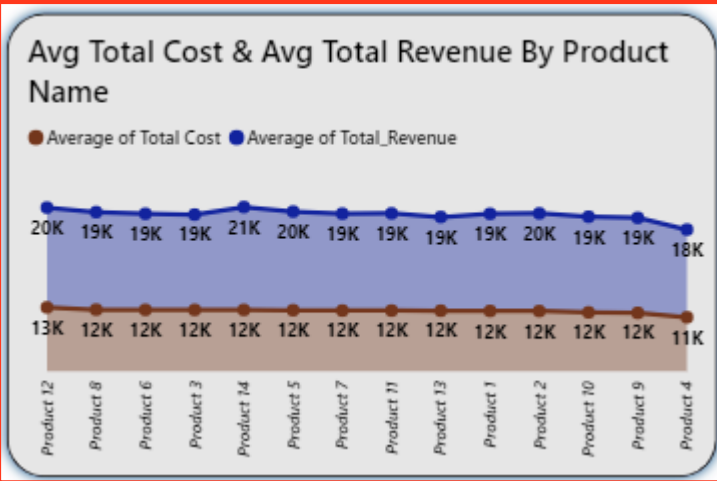
The products sold by channel and currency are shown in the clustered column chart. It is evident that the sold goods is bifurcated first by channel and subsequently by currency code. With **13,880** products, the product in the **wholesale** channel is sold in **New Zealand dollars**. Then, **10,526** products were sold using **USD**. The final two currencies are EUR and AUD. Additionally, the **distributor** channel sold the most number of products in **NZD**, followed by **USD**, with 8001 and 6297 products respectively. The final **export** channel trades primarily in **NZD** and **USD** rather than AUD and EUR. This Shows that the business mainly deals in **NZD** and **USD**.

6. MAP: % TOTAL REVENUE BY CITY



This map is designed to show the location that helps the business generate the most revenue, expressed as a percentage. The city that generates the highest revenue for the business is **Queenstown-Lakes**, which brings in **9%** of the total revenue. **Christchurch** and **Hamilton** come in second & third, with 7.46% & 7.43% of the revenue, respectively. Then rest of the cities you can see in the map. This helps the decision maker to focus on location that are less contributing in the business.

7. STACKED AREA CHART: AVG TOTAL COST & AVG TOTAL REVENUE BY PRODUCT NAME



Analyzing average cost and revenue by product name involves calculating the average financial metrics associated with each specific product or service offered by a business. It shows that every products Average revenue is more than the Average total cost of each product. **Product 12** has the most Average cost (**13K**) and generates the highest revenue (**20K**) . It shows every products is performs well according to there cost and revenue . **Product 2** is the one with a high Average revenue but the cost is less than Product 12 .

FROM REPORT 2: PRODUCT PERFORMANCE REPORT

1. KPI'S REQUIREMENT:

67579

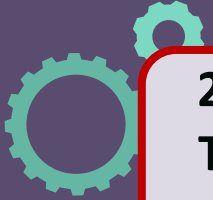
Total Products Sold

8.46

Avg Products Per Order

KEY PERFORMANCE INDICATOR:-

- **Total Products sold** : The total product sold during the business are **67,579** units.
- **Avg Products Per Order**: The average number of products sold per order is **8.46** units.



2 & 3 STACKED BAR CHART: TOP & BOTTOM PRODUCTS BY REVENUE

Top 5 Products By Revenue



Bottom 5 Products By Revenue



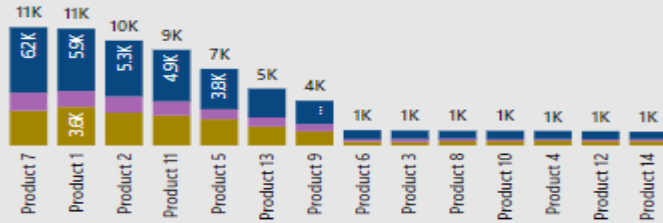
The top five products in terms of revenue are displayed in the first visual expression. With **25.71M**, **Product 7** bring in the most money. Product 1(**25.49M**), Product 2 (22.85M), & Product 5(17.02M) products are next in line.

In the second visual expression, the lowest five products in terms of revenue are shown. **Product 4** makes the least money, with **2.86M**. The next in line, in minimum revenue, are the Product 3 (3.07M), Product 12 (3.11M), Product 10 (3.11M), & Product 14 (3.14M).

4 & 5 STACKED COLUMN CHART: PRODUCT PERFORMANCE BY CHANNEL & CURRENCY CODE

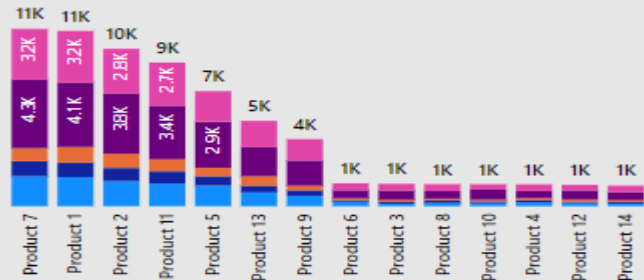
Products Performance By Channel

Channel ● Distributor ● Export ● Wholesale



Products Performance By Currency Code

Currency Code ● AUD ● EUR ● GBP ● NZD ● USD



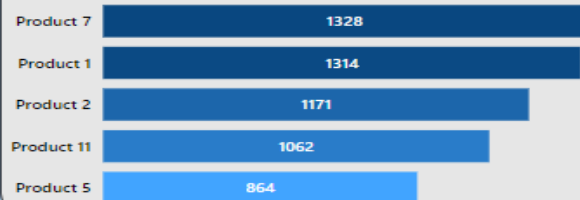
The first visual representation shows the product performance by **channel**. Every product show the total products sold according to channel . **Product 7 (11K)** is the most sold product. Then we can see that channel wholesale sells most product 7 with **6.2K**. Then export does the most sales. It goes for the rest of the products in the graph.

In the second visual expression, it represents the product performance by **currency code** . Every product show the total products sold according to Currency Code . **Product 7 (11K)** is the most sold product. Then we can see that currency code NZD sells most product 7 with **4.3K** . Then USD does the most sales. It goes for the rest of the products in the graph.

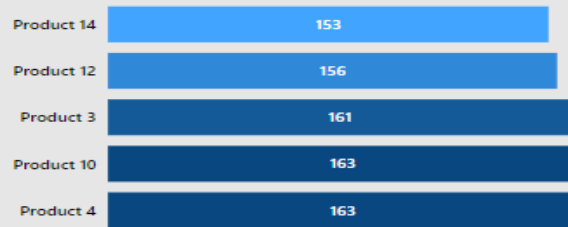


6 & 7 STACKED BAR CHART: TOP & BOTTOM PRODUCTS BY ORDERS

Top 5 Products By Total Orders



Bottom 5 Products By Total Orders



The top five products in terms of orders placed are displayed in the first visual expression. With **1328**, **Product 7** is the product that is ordered mostly. Product 1(**1314**), Product 2 (**1171**), Product 11(1062) & Product 5(**864**) products are next in line in terms of Orders.

In the second visual expression, the lowest five products in terms of order placed are shown. **Product 14** makes the least orders, with **153**. The next in line, in minimum orders, are the Product 12 (**156**), Product 3 (**161**), Product 10 (**163**), & Product 4 (**163**).

FROM REPORT 3: CUSTOMER ANALYSIS

1. TABLE: CUSTOMER'S PURCHASE PATTERN

Customer's Purchase Pattern

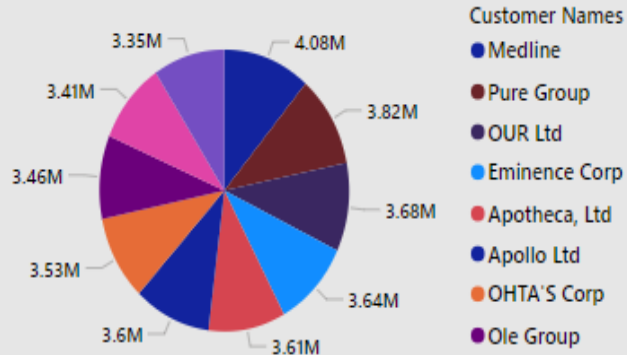
Customer_Names	Total Orders	Total Products Sold	Total Profit
Medline	210	1754	15,06,669.31
Pure Group	175	1547	14,02,517.29
OUR Ltd	176	1492	13,98,368.61
Apollo Ltd	178	1553	13,92,406.16
OHTA'S Corp	173	1504	13,55,516.68
Apotheca, Ltd	179	1486	13,37,765.81
Eminence Corp	186	1572	13,32,935.50
Ohio	164	1376	13,18,352.27
ETUDE Ltd	167	1435	13,14,732.71
Linde	157	1398	12,86,557.86
Victory Ltd	176	1515	12,63,720.89
Ei	171	1467	12,61,536.30
Prasco Group	162	1364	12,31,436.19
Ole Group	171	1478	12,27,354.35
Rochester Ltd	142	1242	12,15,109.37
Weimei Corp	161	1346	12,12,960.32
Sundial	163	1360	12,09,882.20

This table makes it easy to monitor how customers behave when making purchases. This provides comprehensive customer information, allowing us to view the products they have ordered and the profit the business has made from each individual consumer. We can clearly see the top 3 customers according to the table are **Medline** , **Pure Group** and **OUR LTD**. Medline ordered **210** times , they get **1754** products for these orders and the money that our business makes is **1.5M**. This same goes for the rest customers.



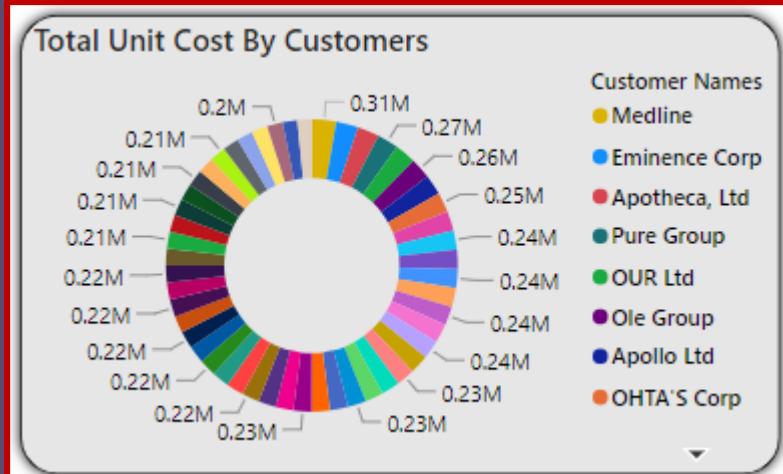
2. PIE CHART:TOP 10 CUSTOMERS BY TOTAL REVENUE

Top 10 Customers By Total Revenue



We can examine the top 10 customers by total revenue in this pie chart. It displays the type of clients who generate the most income for the company. We can clearly see the top 3 customers according to the table are **Medline** , **Pure Group** and **OUR LTD**. The revenue that we generated from Medline is **4.08M** and for the rest it is **3.82M** and **3.68M**. The rest of the chart shows other customers that help to make the money for the business or company.

3. DONUT CHART: TOTAL UNIT COST BY CUSTOMERS



It refers to the combined expenses incurred by a business to produce and deliver a single unit of product, as perceived and influenced by the customer.

As we saw before Medline is the Top customer for the business here also the most Total unit cost spend is on **Medline** of **0.31M**. We can see the rest of the 44 customers . This helps us to understand the value of the customers.



REFERENCES

Data Source:

yesrahulkr. "Sales Analysis Report on Power BI." Kaggle, 2020. Available at: Kaggle Sales Analysis Report on Power BI

Data Access Method:

The dataset was directly downloaded using the Kaggle with the following link:
<https://www.kaggle.com/datasets/yesrahulkr/sales-analysis-report-on-power-bi>

The dataset can also be accessed and downloaded using the Kaggle API with the following command:

[`kaggle datasets download -d yesrahulkr/sales-analysis-report-on-power-bi`](#)

This commands allows for easy programmatic access and integration of the dataset into data analysis workflows.

A close-up photograph of a computer keyboard. The central focus is a large, rectangular blue key with the words "Thank you!" printed in white, italicized, sans-serif font. Surrounding this key are several standard black keyboard keys with white symbols: an "M" key to the top left, a "% ù" key to the top, a "£ , ' " key to the top right, a large arrow key to the far right, a "+ =" key to the bottom left, and a key with a white triangle symbol below it. The keyboard is set against a light gray background.

Thank you!