

# Electronics Inventory Management Dashboard

## Helping Manual

### 1. Introduction

This manual provides guidance on setting up and using the Electronics Inventory Management Dashboard using SQL and Power BI. The objective is to manage inventory, sales trends, supplier data, and customer insights.

### 2. Project Components

Components:

- SQL Script: Table creation, data insertion and SQL Queries
- Power BI Report: Dashboard visuals.
- Project Proposal: Detailed system overview.

## Functionalities in Dashboard

### 1. Sales Overview Dashboard:

This query calculates key sales metrics: the total number of unique transactions, the total sales revenue after applying discounts, and the average order value. We use **KPIs** in Power BI to visually represent these metrics, providing insights into sales performance and helping to drive data-informed decisions.

```
SELECT
    COUNT(DISTINCT SaleID) AS TotalTransactions,
    SUM(QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) AS TotalSales,
    SUM(QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) / COUNT(DISTINCT SaleID) AS AvgOrderValue
FROM Sales s
JOIN Products p ON s.ProductID = p.ProductID;
```

90	56.49K	627.65
Sum of TotalTransactions	Sum of TotalSales	Sum of AvgOrderValue

## 2. Inventory Levels

Visualize current stock levels for each electronic product, highlighting high-stock items, Medium-stock items, low-stock items for timely restocking .We use a **stacked column chart** in Power BI to visually display these stock levels, allowing for easy comparison across different product categories

```
SELECT p.ProductName, p.StockQuantity, c.CategoryName,
CASE
WHEN p.StockQuantity < 50 THEN 'Low Stock'
WHEN p.StockQuantity BETWEEN 50 AND 100 THEN 'Medium Stock'
ELSE 'High Stock'END AS StockStatus
FROM Products p
JOIN Categories c ON p.CategoryID = c.CategoryID
ORDER BY p.StockQuantity ASC;
```



## 3. Sales by Product Category

Analyzing sales distribution across categories such as smartphones, laptops, headphones, and smartwatches. We use a line and stacked column chart in Power BI to effectively visualize sales performance across categories, enabling easy comparison and trend analysis.

```
SELECT c.CategoryName,
SUM(s.QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) AS TotalSalesValue
FROM Sales s JOIN Products p ON s.ProductID = p.ProductID
JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY TotalSalesValue DESC;
```



#### 4. Top-Selling Products

Identify top 10 best-selling electronic products within a specified period, with filters for category, region, and time. We use a stacked column chart in Power BI to visually display these top products, highlighting their performance across different categories.

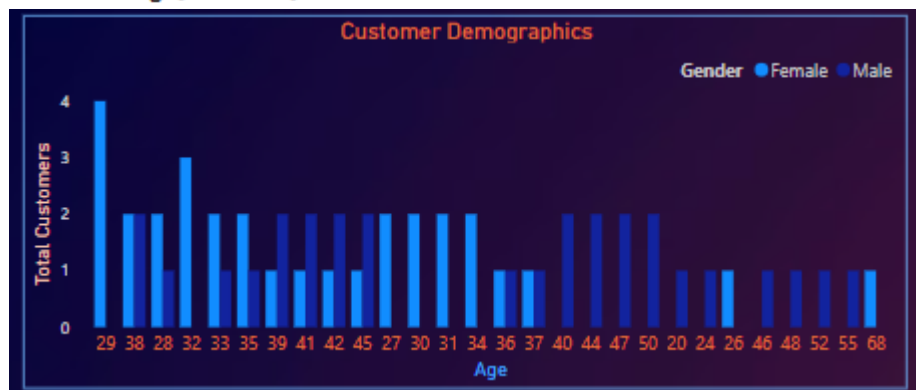
```
SELECT TOP 10 p.ProductName, c.CategoryName,  
SUM(s.QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) AS TotalSalesValue  
FROM Sales s  
JOIN Products p ON s.ProductID = p.ProductID  
JOIN Categories c ON p.CategoryID = c.CategoryID  
GROUP BY p.ProductName, c.CategoryName  
ORDER BY TotalSalesValue DESC;
```



#### 5. Customer Demographics

Analyze customer age and gender to tailor marketing strategies for electronic products. We use a clustered column chart in Power BI to visually represent this data, allowing for easy comparison of customer counts across different age and gender groups.

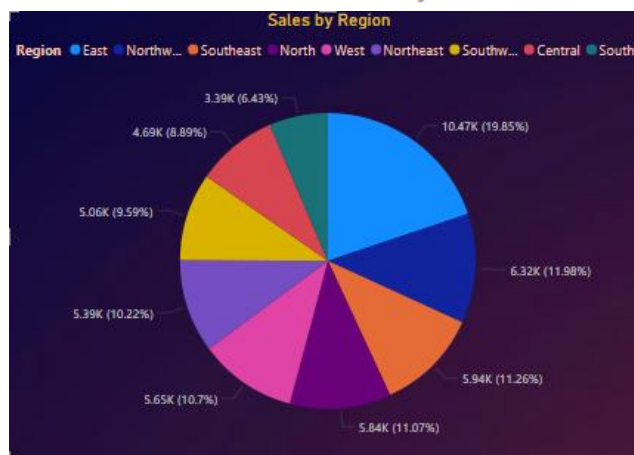
```
SELECT Gender, Age, COUNT(*) AS CustomerCount  
FROM Customers  
GROUP BY Gender, Age  
ORDER BY Age, Gender;
```



## 6. Sales by Region

Tracking sales performance across different regions or cities to optimize inventory allocation and promotional campaigns. We use a pie chart in Power BI to visually represent these sales distributions across regions, providing a clear overview of regional performance.

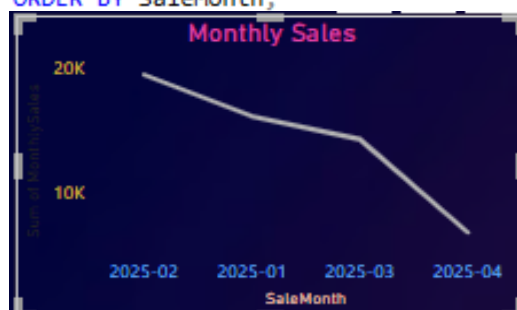
```
SELECT r.RName,  
SUM(s.QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) AS TotalSalesValue  
FROM Sales s  
JOIN Products p ON s.ProductID = p.ProductID  
join Region r on r.RId=s.RId  
GROUP BY r.RName  
ORDER BY TotalSalesValue DESC;
```



## 7. Monthly Sales Trend

Showing a line chart that illustrates sales trends over time, allowing users to drill down into daily, weekly, or monthly data. We use a line chart in Power BI to visually display these monthly sales trends, enabling easy identification of patterns and fluctuations over time.

```
SELECT FORMAT(SaleDate, 'yyyy-MM') AS SaleMonth,  
SUM(QuantitySold * Products.UnitPrice * (1-DiscountPercent/100)) AS MonthlySales  
FROM Sales JOIN Products ON Sales.ProductID = Products.ProductID  
GROUP BY FORMAT(SaleDate, 'yyyy-MM')  
ORDER BY SaleMonth;
```



## 8. Supplier Performance Analysis

Analyze supplier contributions by products supplied, sales volume, revenue generated, and quality metrics (returns and warranty claims). Identifies top-performing suppliers and potential quality issues. We use a clustered column chart in Power BI to visually represent these metrics, facilitating comparisons of supplier performance.

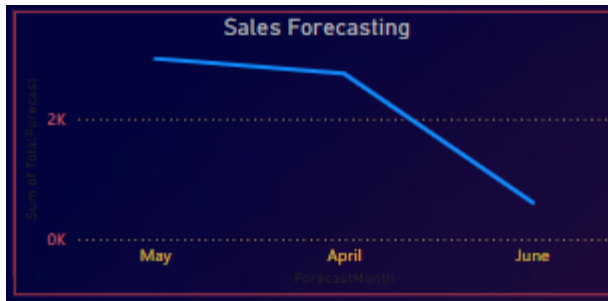
```
SELECT s.SupplierName,
       count(DISTINCT ps.ProductID) AS ProductsSupplied,
       sum(sa.QuantitySold) AS UnitsSold,
       sum(sa.QuantitySold * p.UnitPrice) AS RevenueGenerated,
       count(DISTINCT r.ReturnID) AS ReturnsCount,
       count(DISTINCT w.WarrantyID) AS WarrantyClaims,
       (count(DISTINCT r.ReturnID) * 100.0 / sum(sa.QuantitySold)) AS ReturnRate
FROM Suppliers s
join ProductSupplier ps ON s.SupplierID = ps.SupplierID
join Products p ON ps.ProductID = p.ProductID
join Sales sa ON p.ProductID = sa.ProductID
left join Returns r ON sa.SaleID = r.SaleID
left join Warranties w ON p.ProductID = w.ProductID AND sa.CustomerID = w.CustomerID
group BY s.SupplierName
order BY RevenueGenerated DESC;
```



## 9. Monthly Sales Forecasting

Showing historical sales data to create forecasts for upcoming months or quarters, aiding in inventory planning. We use a line chart in Power BI to visually display these monthly forecasts, enabling easy analysis of projected sales patterns over time.

```
SELECT ForecastMonth, SUM(ForecastedSales) AS TotalForecast
FROM SalesForecast
GROUP BY ForecastMonth;
```



## 10. Customer Segmentation

Segment customers based on purchasing behavior related to electronics, enabling targeted marketing campaigns. We use a stacked column chart in Power BI to visually represent these metrics, allowing for easy comparison of total sales across different customer segments.

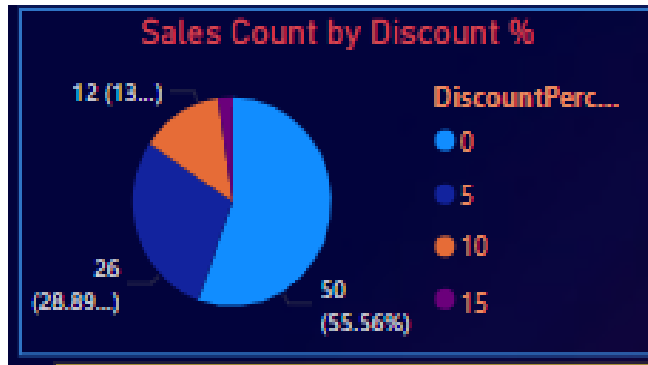
```
SELECT cs.SegmentName, count(DISTINCT c.CustomerID) AS CustomerCount,
sum(s.QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent/100))) AS TotalSalesValue
FROM Customers c
JOIN CustomerSegments cs ON c.SegmentID = cs.SegmentID
JOIN Sales s ON c.CustomerID = s.CustomerID
JOIN Products p ON s.ProductID = p.ProductID
GROUP BY cs.SegmentName
```



## 11. Sales Discount Performance Metrics

Provide an analysis of how discounts impact order volume by displaying the discount percentage alongside the count of orders for each discount level. We use a pie chart in Power BI to visually display the distribution of order counts across different discount levels, highlighting the impact of discounts on sales.

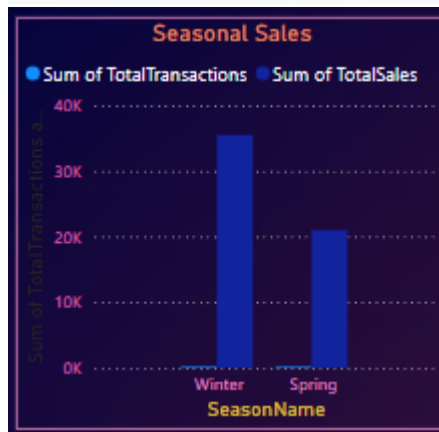
```
SELECT DiscountPercent, COUNT(*) AS OrderCount
FROM Sales
GROUP BY DiscountPercent
ORDER BY DiscountPercent;
```



## 12. Seasonal Sales Analysis

Analyzes sales patterns by season, summarizing transactions and total sales for each seasonal period. It categorizes sales data into four distinct seasons—Winter, Spring, Autumn, Summer based on the month of the sale date. This approach provides insights into how sales fluctuate throughout the year, helping to identify seasonal trends in consumer behavior. We use

```
SELECT se.SeasonName, COUNT(DISTINCT s.SaleID) AS TotalTransactions,
SUM(s.QuantitySold * (p.UnitPrice * (1 - s.DiscountPercent / 100))) AS TotalSales
FROM Sales s
JOIN Products p ON s.ProductID = p.ProductID
JOIN Seasons se ON
    (MONTH(s.SaleDate) IN (12, 1, 2) AND se.SeasonName = 'Winter') OR
    (MONTH(s.SaleDate) IN (3, 4, 5) AND se.SeasonName = 'Spring') OR
    (MONTH(s.SaleDate) IN (6, 7, 8) AND se.SeasonName = 'Summer') OR
    (MONTH(s.SaleDate) IN (9, 10, 11) AND se.SeasonName = 'Autumn')
GROUP BY se.SeasonName
```

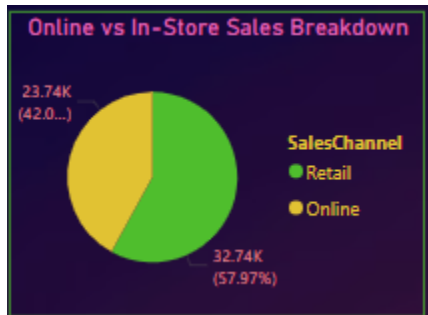




### 13. Sales Channel Analysis

Comparing sales performance between online and in-store channels for electronic products. We use a pie chart to visually represent the distribution of total sales across different sales channels, highlighting the contributions of each channel to overall sales performance.

```
SELECT SalesChannel,  
SUM(QuantitySold * (p.UnitPrice * (1 - DiscountPercent/100))) AS TotalSalesValue  
FROM Sales s JOIN Products p ON s.ProductID = p.ProductID  
GROUP BY SalesChannel;
```



### 14. Best-Selling Categories

Ranking product categories based on the total units sold, providing insights into market demand trends. This information guides inventory procurement and category-level sales strategies. We use a stacked column chart in Power BI to visually represent the total units sold across different categories, allowing for easy comparison and analysis.

```
SELECT c.CategoryName, SUM(s.QuantitySold) AS TotalUnitsSold  
FROM Categories c  
JOIN Products p ON c.CategoryID = p.CategoryID  
JOIN Sales s ON p.ProductID = s.ProductID  
GROUP BY c.CategoryName  
ORDER BY TotalUnitsSold DESC;
```



### 15. Warranty and Return Tracking

Monitoring warranty claims and product returns across different categories, providing insights into product performance and customer satisfaction. It calculates the return rate



for each category, helping identify areas for improvement. We use a **clustered column chart** in Power BI to visually display the return counts and warranty claims by category, facilitating easy comparison

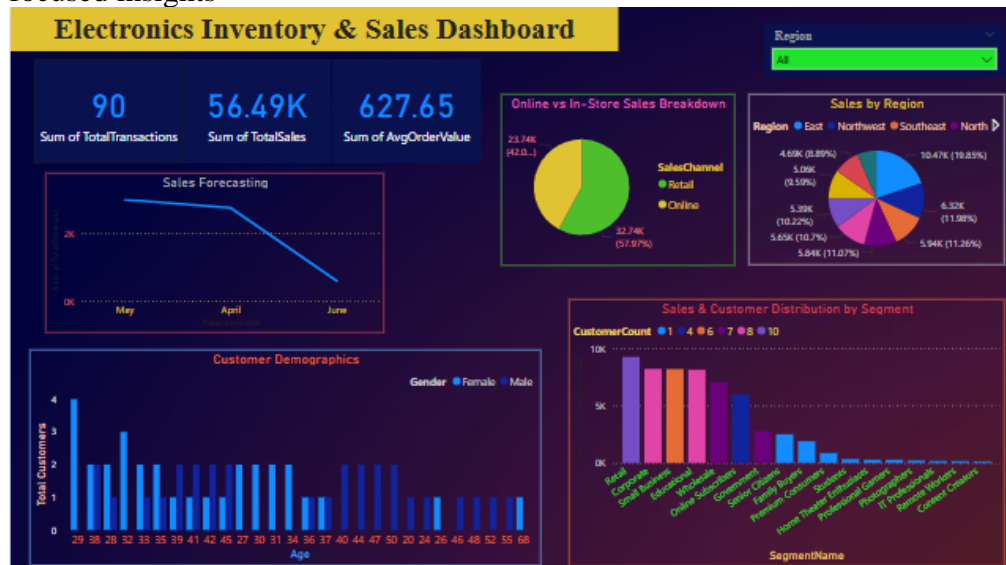
```
SELECT c.CategoryName, COUNT(DISTINCT r.ReturnID) AS ReturnCount,
COUNT(DISTINCT w.WarrantyID) AS WarrantyCount,
COUNT(DISTINCT r.ReturnID) * 100.0 / COUNT(DISTINCT s.SaleID) AS ReturnRate
FROM Products p
JOIN Categories c ON p.CategoryID = c.CategoryID
LEFT JOIN Sales s ON p.ProductID = s.ProductID
LEFT JOIN Returns r ON s.SaleID = r.SaleID
LEFT JOIN Warranties w ON p.ProductID = w.ProductID
GROUP BY c.CategoryName
ORDER BY ReturnRate DESC;
```



## Filters in Dashboard

### Page 1

The **Region Filter** allows users to dynamically analyze data by geographic segments (e.g., Country, State, City). Selecting a region updates all linked visuals, enabling focused insights



Page 2

The **Categories filter** allows users to select specific product categories, dynamically updating the dashboard visuals to reflect data related only to the selected category. This interaction provides insights into sales performance, inventory levels, and stock status for the chosen category, helping users make informed decisions.



Page 3

The **Categories filter** enables users to focus on specific product categories, instantly updating the visuals to display relevant data such as warranty claims and return counts. This functionality allows for a deeper analysis of customer satisfaction and product performance within the selected category, enhancing decision-making capabilities.



