**Project Overview:**

In this Power BI project, I will be creating an end-to-end interactive and dynamic car sales dashboard. The dashboard is designed to provide a comprehensive overview of the company's performance and insights for stakeholders, including KPIs, charts, and a detailed grid view. The primary focus is on building a dynamic and visually appealing dashboard using Power BI.

* **Sales Overview:**
  + KPIs:
    - Year-to-date sales
    - Month-to-date total sales
    - Year-on-year growth in total sales
    - Difference between year-to-date sales and previous year-to-date sales
    - Year-to-date average price analysis
    - Month-to-date total cars sold
    - Year-on-year growth in cars sold
    - Difference between year-to-date cars sold and previous year-to-date cars sold
  + Dynamic Filtering:
    - Users can interactively filter data by body style, dealer, transmission, color, and more to analyze the sales performance from various angles.
* **Charts:**
  + Year-to-Date Sales Weekly Trend:
    - Visualizing the weekly trend of year-to-date sales, highlighting the maximum value each week.
  + Year-to-Date Sales Analysis:
    - Breakdown of year-to-date sales by body style, color, and dealer region.
  + Company-wise Sales Trend:
    - Analyzing the sales trend for different car brands/companies.
  + Detailed Grid View:
    - Displaying a detailed grid view of individual car sales data with export functionality.

**Project Functionalities:**

* Connecting Excel files to Power BI.
* Implementing DAX (Data Analysis Expressions) functions for calculations.
* Utilizing time intelligence functions for date-related analysis.
* Building custom charts and visualizations.
* Implementing dynamic filters and conditional formatting.
* Constructing interactive and dynamic dashboards in Power BI.

**Key Learning Areas:**

* Data modeling and quality checking.
* Building calendar tables for time-based analysis.
* Implementing DAX functions for complex calculations.
* Creating visually appealing and informative charts.
* Designing dynamic dashboards with interactive filters.
* Exporting data and generating insights for stakeholders.

**DAX Functions Used in the Project:**

* Avg Price:
  + Function:SUM(car\_data[Price ($)]) / COUNT(car\_data[Car\_id])
  + Description: Calculates the average price of cars based on the sum of prices divided by the count of unique car IDs.
* Avg Price Color:
  + Function:IF([Avg Price Difference]>0, "Green", "Red")
  + Description: Determines the color (Green or Red) based on whether the average price difference is positive or negative.
* Avg Price Difference:
  + Function: [YTD Avg Price] - [PYTD Avg Price]
  + Description: Calculates the difference between Year-to-Date (YTD) and Previous Year-to-Date (PYTD) average prices.
* MTD Avg Price:
  + Function: TOTALMTD([Avg Price], 'Calendar Table'[Date])
  + Description: Computes the Month-to-Date (MTD) average price using the total MTD function over the calendar date.
* MTD Avg Price KPI:
  + Function: CONCATENATE("MTD Avg Price : ", FORMAT([MTD Avg Price] / 1000, "$0.00K"))
  + Description: Concatenates a string indicating MTD Avg Price along with formatting the value in thousands.
* MTD KPI:
  + Function:CONCATENATE("MTD Total Sales : ", FORMAT([MTD Total Sales] / 1000000, "$0.00M"))
  + Description: Concatenates a string indicating MTD Total Sales along with formatting the value in millions.
* MTD Total Sales:
  + Function: TOTALMTD(SUM(car\_data[Price ($)]), 'Calendar Table'[Date])
  + Description: Computes the Month-to-Date (MTD) total sales using the total MTD function over the calendar date.
* PYTD Avg Price:
  + Function: CALCULATE([Avg Price], SAMEPERIODLASTYEAR('Calendar Table'[Date]))
  + Description: Calculates the Previous Year-to-Date (PYTD) average price using the calculate function with the same period last year.
* PYTD Total Sales:
  + Function: CALCULATE(SUM(car\_data[Price ($)]), SAMEPERIODLASTYEAR('Calendar Table'[Date]))
  + Description: Calculates the Previous Year-to-Date (PYTD) total sales using the calculate function with the same period last year.
* Sales Difference:
  + Function: [YTD Total Sales] - [PYTD Total Sales]
  + Description: Calculates the difference between Year-to-Date (YTD) and Previous Year-to-Date (PYTD) total sales.
* Sales Difference Color:
  + Function:IF([Sales Difference]>0, "Green", "Red")
  + Description: Determines the color (Green or Red) based on whether the sales difference is positive or negative.
* YoY Avg Price Growth:
  + Function: [Avg Price Difference] / [PYTD Avg Price]
  + Description: Calculates the Year-on-Year (YoY) growth percentage for average prices.
* YoY Sales Growth:
  + Function:[Sales Difference] / [PYTD Total Sales]
  + Description: Calculates the Year-on-Year (YoY) growth percentage for total sales.
* YTD Avg Price:
  + Function: TOTALYTD([Avg Price], 'Calendar Table'[Date])
  + Description: Computes the Year-to-Date (YTD) average price using the total YTD function over the calendar date.
* YTD Total Sales:
  + Function: TOTALYTD(SUM(car\_data[Price ($)]), 'Calendar Table'[Date])
  + Description: Computes the Year-to-Date (YTD) total sales using the total YTD function over the calendar date.

YTD Cards Sold:

* Function: TOTALYTD(COUNT(car\_data[Car\_id]), 'Calendar Table'[Date])
* Description: Computes the Year-to-Date (YTD) total count of unique car IDs using the total YTD function over the calendar date.

PYTD Cars Sold:

* Function:CALCULATE(COUNT(car\_data[Car\_id]), SAMEPERIODLASTYEAR('Calendar Table'[Date]))
* Description: Calculates the Previous Year-to-Date (PYTD) count of unique car IDs using the calculate function with the same period last year.

Cars Sold Difference:

* Function: [YTD Cards Sold] - [PYTD Cars Sold]
* Description: Calculates the difference between Year-to-Date (YTD) and Previous Year-to-Date (PYTD) car counts.

Cars Sold Color:

* Function:IF(car\_data[Cars Sold Difference]>0, "Green", "Red")
* Description: Determines the color (Green or Red) based on whether the car count difference is positive or negative.

YoY Car Sold Growth:

* Function: [Cars Sold Difference] / [PYTD Cars Sold]
* Description: Calculates the Year-on-Year (YoY) growth percentage for car counts.

MTD Cars Sold:

* Function: TOTALMTD(COUNT(car\_data[Car\_id]), 'Calendar Table'[Date])
* Description: Computes the Month-to-Date (MTD) total count of unique car IDs using the total MTD function over the calendar date.

MTD Cars Sold KPI:

* Function: CONCATENATE("MTD Cars Sold : ", FORMAT([MTD Cars Sold] / 1000, "0.00K"))
* Description: Concatenates a string indicating MTD Cars Sold along with formatting the value in thousands.

Total Sales:

* Function: SUM(car\_data[Price ($)])
* Description: Calculates the total sales amount by summing the 'Price ($)' column in the 'car\_data' table.

Max Point:

* Function: IF(MAXX(ALLSELECTED('Calendar Table'[Week]), [Total Sales]) = [Total Sales], MAXX(ALLSELECTED('Calendar Table'[Week]), [Total Sales]), blank())
* Description: Determines the maximum sales point for a given week, returning the sales value if it is the maximum point; otherwise, returns blank.