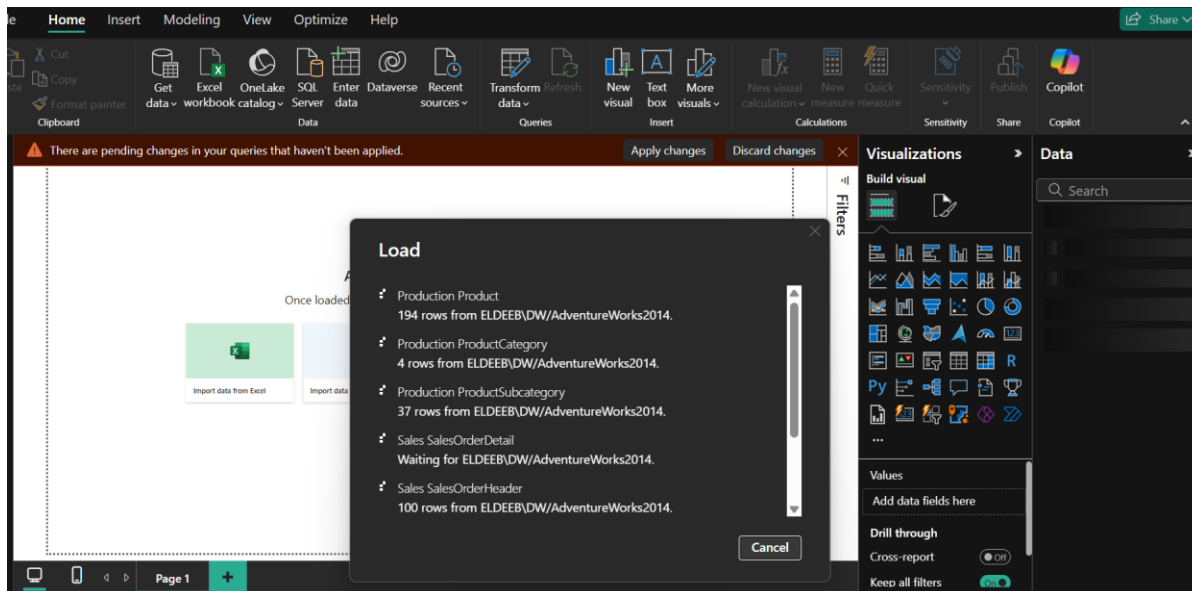
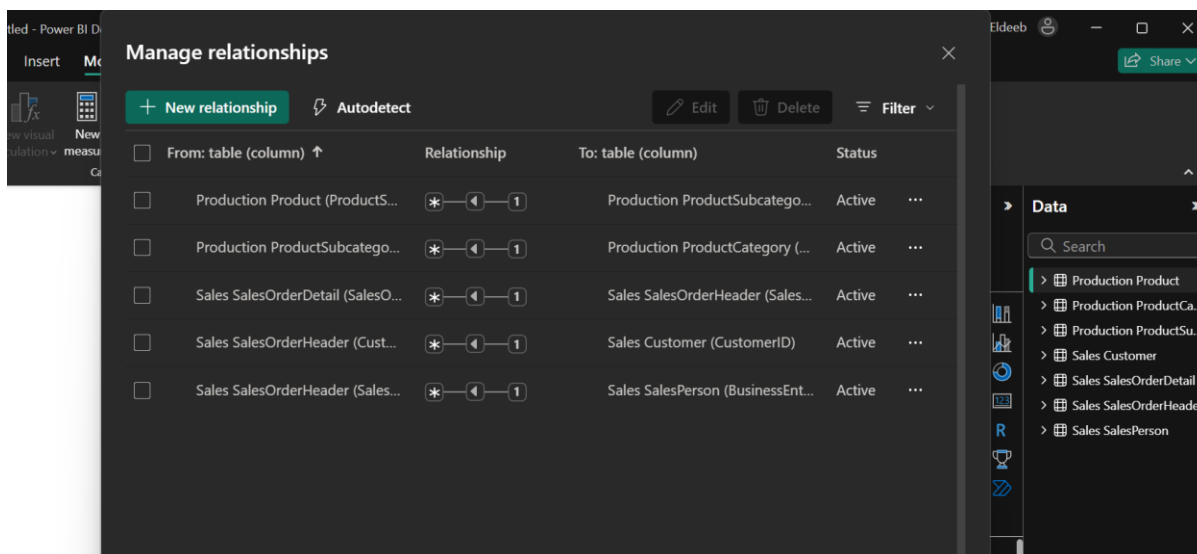


- Task 1.1:

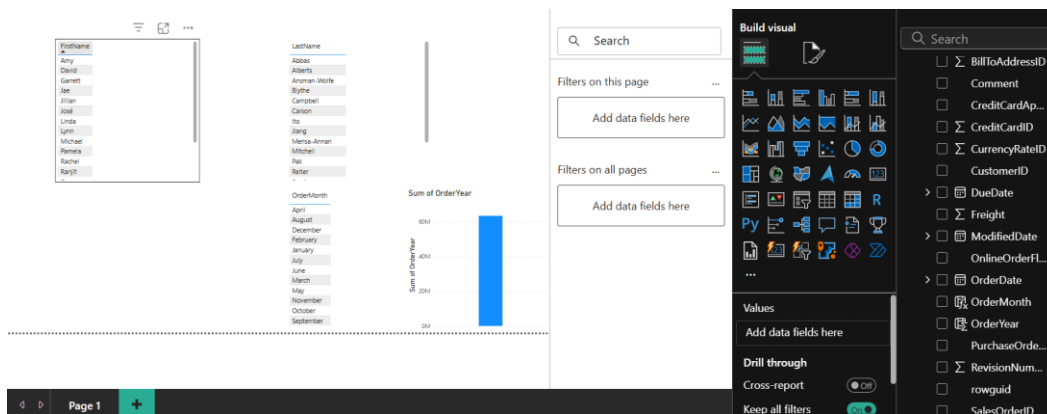
Started by loading the necessary tables into my project.



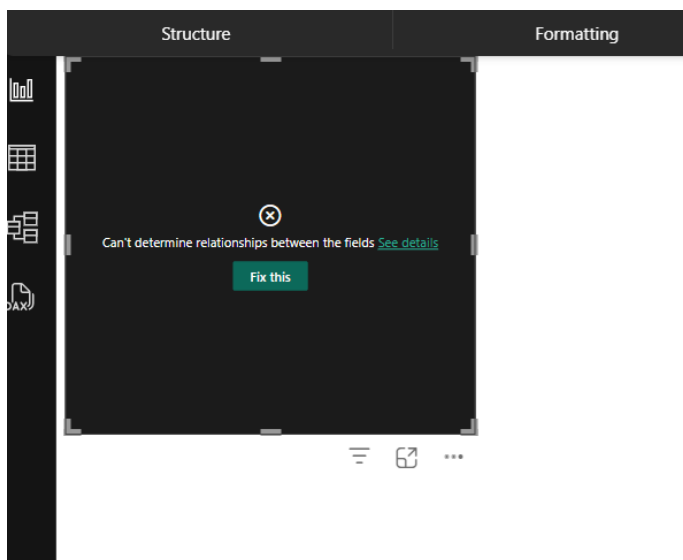
Then I verified the existence of the relationships between tables.



In the next step I tried to create a visualization for each of the column fields by dragging them to the workspace.



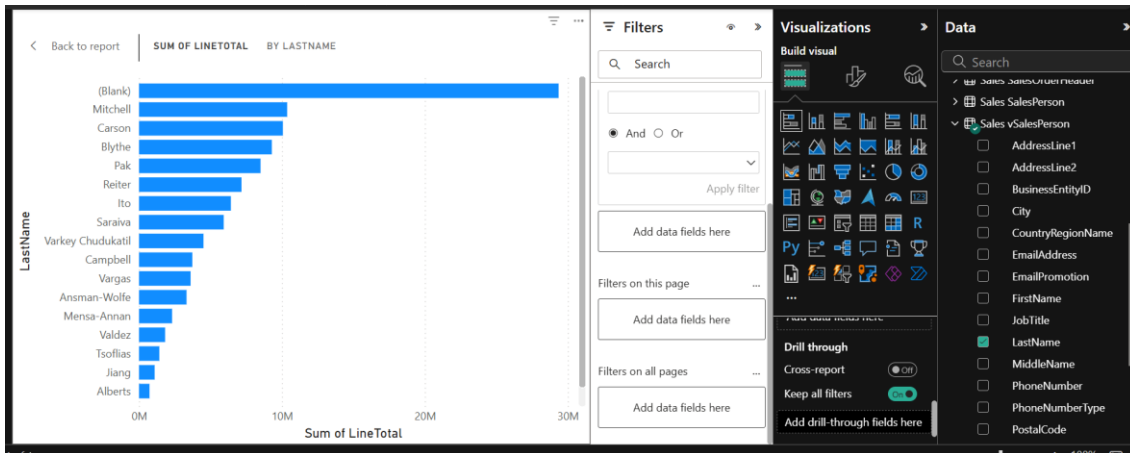
Then I tried to create one visualization containing all of these columns but I got an error.



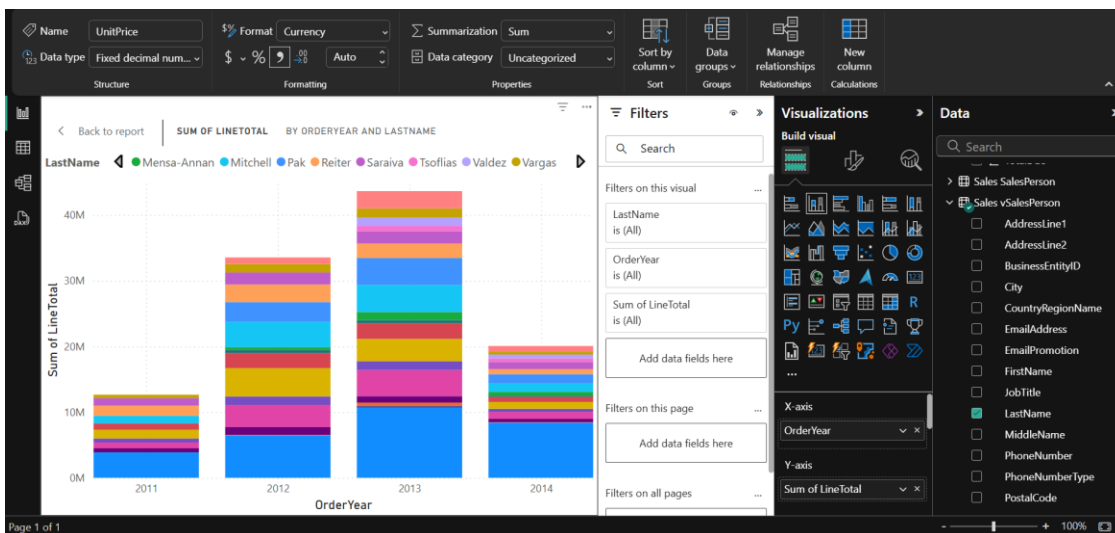
This error is due the impossible determination of the relationships between the fields chosen.

- Task 1.2&1.3:

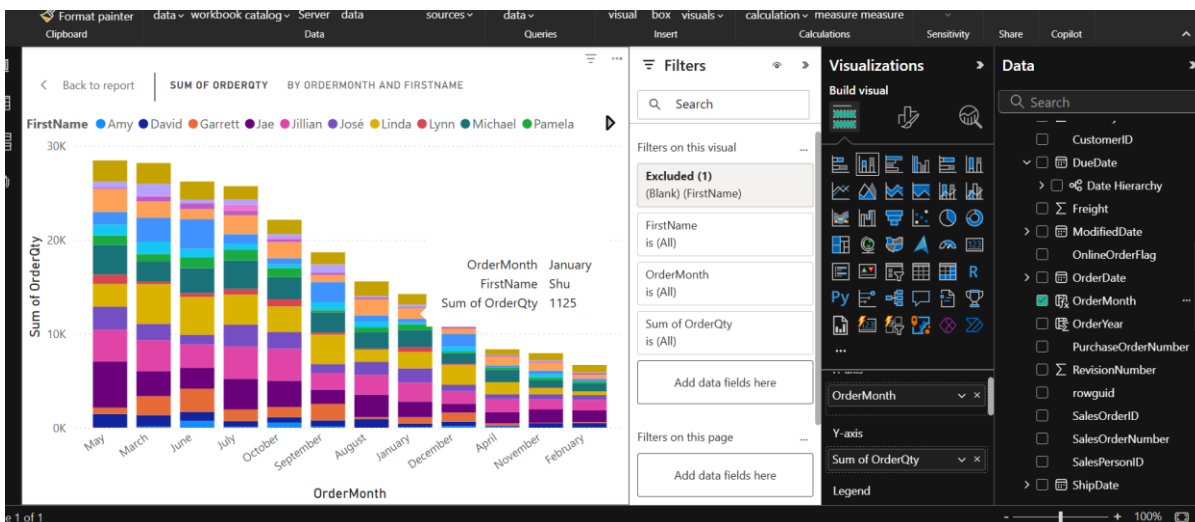
Here I created a visual for the Sales representative LastName and the sum(LineTotal).



This visual is stacked column chart for the yearly total sales for the sales representatives by the last name.



In this visual I have created a bar chart to show the number of sales assigned to the first name of the salesperson in every month.



- Task 2.1 → 2.5:

We don't need all the products attributes or every attribute of the chosen tables to get a final OBT containing the data we need, in this task I have used the necessary tables to get an OBT of the important info about the sales as shown.

The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is a SELECT statement that joins several tables: Production.Product (p), Production.ProductSubcategory (psc), Production.ProductCategory (pc), Sales.SalesOrderHeader (soh), Sales.SalesOrderDetail (sod), and SalesPerson (vsp). The query selects various attributes including ProductID, ProductName, CategoryName, SubcategoryName, ProductPrice, ProductColor, WeightCategory, RepID, and RepName. It also includes a CASE statement to categorize products by weight (LIGHT, MEDIUM, HEAVY). The results are displayed in a table with 10 columns and 7 rows.

```
SELECT
p.ProductID,
p.Name AS ProductName,
pc.Name AS CategoryName,
psc.Name AS SubcategoryName,
p.ListPrice AS ProductPrice,
p.Color AS ProductColor,
CASE
WHEN p.Weight < 5 THEN 'LIGHT'
WHEN p.Weight BETWEEN 5 AND 20 THEN 'MEDIUM'
ELSE 'HEAVY'
END AS 'WeightCategory',
vsp.BusinessEntityID AS RepID,
vsp.FirstName + ' ' + vsp.LastName AS RepName,
DATEPART(YEAR,soh.OrderDate) AS OrderYear,
DATEPART(MONTH,soh.OrderDate) AS OrderMonth,
sod.LineTotal AS SalesValue,
sod.OrderQty,
sod.UnitPriceDiscount AS Discount
FROM Sales.SalesOrderDetail sod
JOIN Production.Product p ON sod.ProductID = p.ProductID
JOIN Production.ProductSubcategory psc ON p.ProductSubcategoryID = psc.ProductSubcategoryID
JOIN Production.ProductCategory pc ON psc.ProductCategoryID = pc.ProductCategoryID
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
JOIN Sales.SalesPerson vsp ON soh.SalesPersonID = vsp.BusinessEntityID
```

	ProductID	ProductName	CategoryName	SubcategoryName	ProductPrice	ProductColor	WeightCategory	RepID	RepName
1	776	Mountain-100 Black, 42	Bikes	Mountain Bikes	3374.99	Black	HEAVY	274	Stephen Jia
2	776	Mountain-100 Black, 42	Components	Mountain Bikes	3374.99	Black	HEAVY	274	Stephen Jia
3	776	Mountain-100 Black, 42	Clothing	Mountain Bikes	3374.99	Black	HEAVY	274	Stephen Jia
4	776	Mountain-100 Black, 42	Accessories	Mountain Bikes	3374.99	Black	HEAVY	274	Stephen Jia
5	714	Long-Sleeve Logo Jersey, M	Bikes	Jerseys	49.99	Multi	HEAVY	274	Stephen Jia
6	714	Long-Sleeve Logo Jersey, M	Components	Jerseys	49.99	Multi	HEAVY	274	Stephen Jia
7	714	Long-Sleeve Logo Jersey, M	Clothing	Jerseys	49.99	Multi	HEAVY	274	Stephen Jia

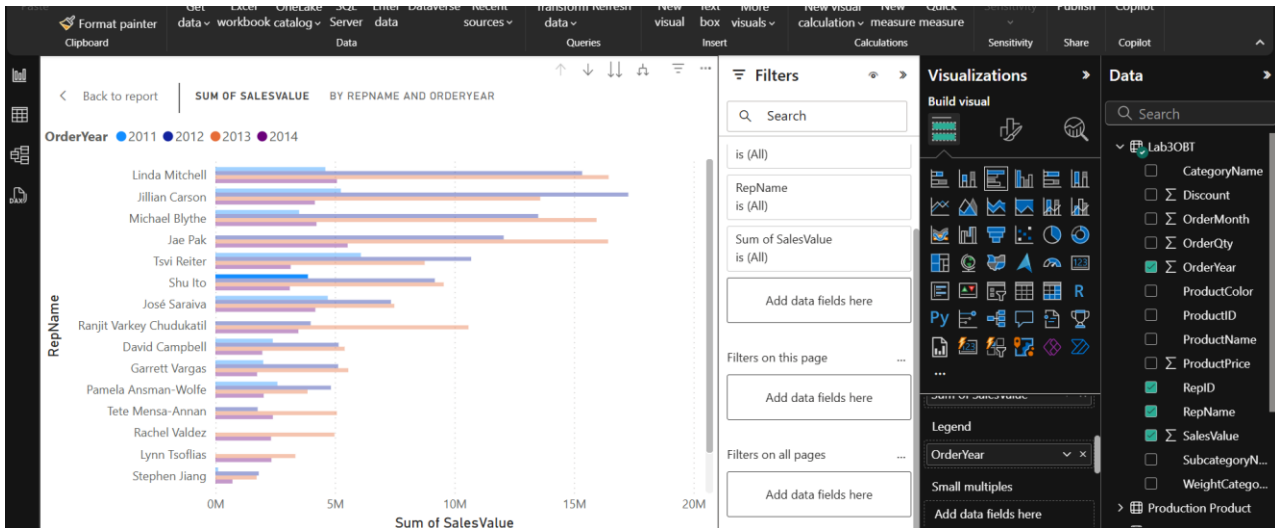
- Task 2.6:

Then I created a view and used it in PowerBI to visualize it:

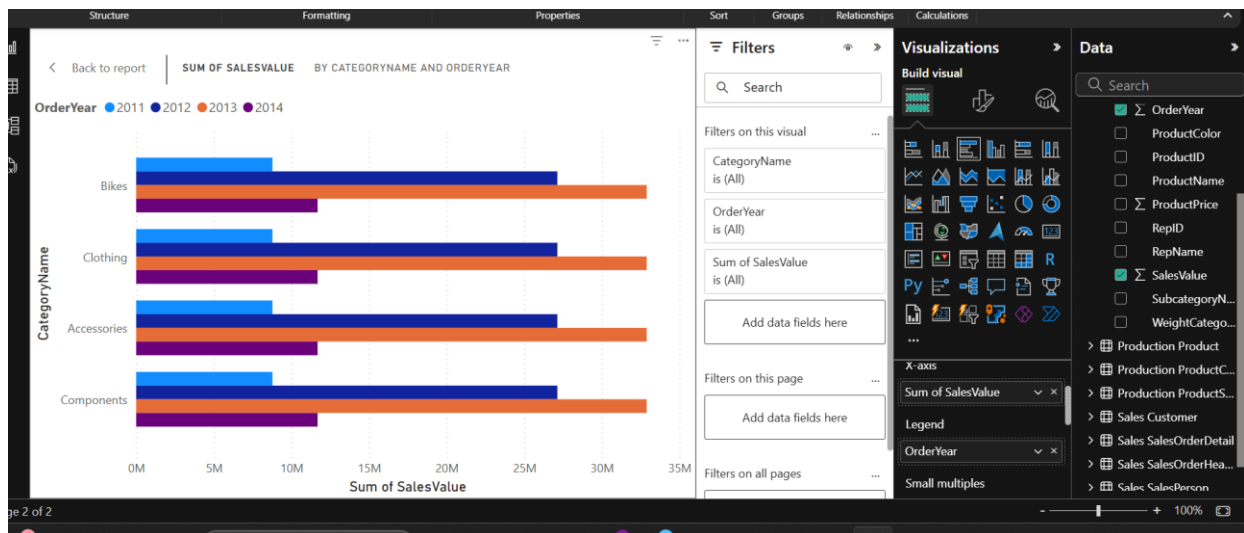
The screenshot shows the PowerBI Desktop interface. The main area displays a table visualization with columns: ProductName, CategoryName, RepName, OrderYear, OrderMonth, and SalesValue. The Fields pane on the right shows the data sources and the fields used in the visualization. The table contains 15 rows of data.

ProductName	CategoryName	RepName	OrderYear	OrderMonth	SalesValue
Patch Kit/8 Patches	Accessories	Amy Alberts	2013	10	4.12
Patch Kit/8 Patches	Bikes	Amy Alberts	2013	10	4.12
Patch Kit/8 Patches	Clothing	Amy Alberts	2013	10	4.12
Patch Kit/8 Patches	Components	Amy Alberts	2013	10	4.12
Bike Wash - Dissolver	Accessories	Amy Alberts	2013	6	9.54
Bike Wash - Dissolver	Bikes	Amy Alberts	2013	6	9.54
Bike Wash - Dissolver	Clothing	Amy Alberts	2013	6	9.54
Bike Wash - Dissolver	Components	Amy Alberts	2013	6	9.54
AWC Logo Cap	Accessories	Amy Alberts	2012	9	10.37
AWC Logo Cap	Bikes	Amy Alberts	2012	9	10.37
AWC Logo Cap	Clothing	Amy Alberts	2012	9	10.37
AWC Logo Cap	Components	Amy Alberts	2012	9	10.37
AWC Logo Cap	Accessories	Amy Alberts	2013	6	10.79
Racing Socks, L	Accessories	Amy Alberts	2013	12	10.79
Racing Socks, M	Accessories	Amy Alberts	2013	12	10.79
AWC Logo Cap	Bikes	Amy Alberts	2013	6	10.79
Racing Socks, L	Bikes	Amy Alberts	2013	12	10.79
Racing Socks, M	Bikes	Amy Alberts	2013	12	10.79

- Task 2.7:



- Task 2.8:



- Task 3:

I created a new query and did a merge for several tables then removed unnecessary column to get my new table DIM_Product.

Table: RenameColumns(#"Reordered Columns",{{"Name", "ProductName"}})

	ProductID	ProductName	CategoryName	SubcategoryName	Color
1	680	HL Road Frame - Black, 58	Components	Road Frames	Black
2	706	HL Road Frame - Red, 58	Components	Road Frames	Red
3	707	Sport-100 Helmet, Red	Accessories	Helmets	Red
4	708	Sport-100 Helmet, Black	Accessories	Helmets	Black
5	709	Mountain Bike Socks, M	Clothing	Socks	White
6	710	Mountain Bike Socks, L	Clothing	Socks	White
7	711	Sport-100 Helmet, Blue	Accessories	Helmets	Blue
8	712	AWC Logo Cap	Clothing	Caps	Multi
9	713	Long-Sleeve Logo Jersey, S	Clothing	Jerseys	Multi
10	714	Long-Sleeve Logo Jersey, M	Clothing	Jerseys	Multi
11	715	Long-Sleeve Logo Jersey, L	Clothing	Jerseys	Multi
12	716	Long-Sleeve Logo Jersey, XL	Clothing	Jerseys	Multi
13	717	HL Road Frame - Red, 62	Components	Road Frames	Red
14	718	HL Road Frame - Red, 44	Components	Road Frames	Red
15	719	HL Road Frame - Red, 48	Components	Road Frames	Red

Here I have merged several columns from several tables to get the final table required having the info about the sales representatives and their address and location, performing add column from example to have 'FullName' column.

APC	FirstName	APC	LastName	APC	FullName	APC	AddressLine1	APC	City	APC	StateProvince	APC	CountryName
	517 distinct, 319 unique		752 distinct, 581 unique		874 distinct, 748 unique		297 distinct, 296 unique		38 distinct, 18 unique		17 distinct, 11 unique		7 distinct, 1 unique
1	Ken		Sánchez		Ken Sánchez		4350 Minute Dr.		Newport Hills		Washington		United States
2	Terri		Duffy		Terri Duffy		7559 Worth Ct.		Renton		Washington		United States
3	Roberto		Tamburello		Roberto Tamburello		2137 Birchwood Dr		Redmond		Washington		United States
4	Rob		Walters		Rob Walters		5678 Lakeview Blvd.		Minneapolis		Minnesota		United States
5	Gail		Erickson		Gail Erickson		9435 Breck Court		Bellevue		Washington		United States
6	Jossef		Goldberg		Jossef Goldberg		5670 Bel Air Dr.		Renton		Washington		United States
7	Dylan		Miller		Dylan Miller		7048 Laurel		Kenmore		Washington		United States
8	Diane		Margheim		Diane Margheim		475 Santa Maria		Everett		Washington		United States
9	Gigi		Matthew		Gigi Matthew		7808 Brown St.		Bellevue		Washington		United States
10	Michael		Raheem		Michael Raheem		1234 Seaside Way		San Francisco		California		United States
11	Ovidiu		Cracium		Ovidiu Cracium		5458 Gladstone Drive		Kenmore		Washington		United States
12	Thierry		D'Hers		Thierry D'Hers		1970 Napa Ct.		Bothell		Washington		United States
13	Janice		Galvin		Janice Galvin		3397 Rancho View Drive		Redmond		Washington		United States
14	Michael		Sullivan		Michael Sullivan		6510 Hacienda Drive		Renton		Washington		United States
15	Sharon		Salavaria		Sharon Salavaria		7165 Brock Lane		Renton		Washington		United States

To create a table contains the dates and each row representing a unique day I use a blank query and then advanced editor to input the DAX query to create the table.

Advanced Editor

Query1

Display Options ?

```

let
    StartDate = #date(2010, 1, 1),
    EndDate = #date(2030, 12, 31),
    DateList = List.Dates(StartDate, Number.From(EndDate - StartDate) + 1, #duration(1,0,0,0)),
    TableFromList = Table.FromList(DateList, Splitter.SplitByNothing(), {"Date"}),
    ChangeType = Table.TransformColumnTypes(TableFromList,{{"Date", type date}},),
    AddColumns = Table.AddColumn(ChangeType, "Year", each Date.Year([Date]), Int64.Type),
    AddMonth = Table.AddColumn(AddColumns, "Month", each Date.Month([Date]), Int64.Type),
    AddMonthName = Table.AddColumn(AddMonth, "Month Name", each Date.ToText([Date], "MMMM"), type text),
    AddDay = Table.AddColumn(AddMonthName, "Day", each Date.Day([Date]), Int64.Type),
    AddWeekday = Table.AddColumn(AddDay, "Day of Week", each Date.DayOfWeekName([Date]), type text),
    AddQuarter = Table.AddColumn(AddWeekday, "Quarter", each "Q" & Number.ToText(Date.QuarterOfYear([Date])), type text),
    AddYearMonth = Table.AddColumn(AddQuarter, "Year-Month", each Date.ToText([Date], "yyyy-MM"), type text)
in
    AddYearMonth
        
```

✓ No syntax errors have been detected.

Done Cancel

= Table.AddColumn(AddQuarter, "Year-Month", each Date.ToText([Date], "yyyy-MM"), type text)					
Date	123 Year	123 Month	APC Month Name	123 Day	
1000 distinct, 1000 unique	3 distinct, 0 unique	12 distinct, 0 unique	12 distinct, 0 unique	31 distinct, 0 unique	
1	01/01/2010	2010	1 January		
2	02/01/2010	2010	1 January		
3	03/01/2010	2010	1 January		
4	04/01/2010	2010	1 January		
5	05/01/2010	2010	1 January		
6	06/01/2010	2010	1 January		
7	07/01/2010	2010	1 January		
8	08/01/2010	2010	1 January		
9	09/01/2010	2010	1 January		
10	10/01/2010	2010	1 January		
11	11/01/2010	2010	1 January		
12					

Query Settings

PROPERTIES

Name

Dim_Date

All Properties

APPLIED STEPS

StartDate

EndDate

DateList

TableFromList

ChangeType

AddColumns

AddMonth

AddMonthName

AddDay

AddWeekday

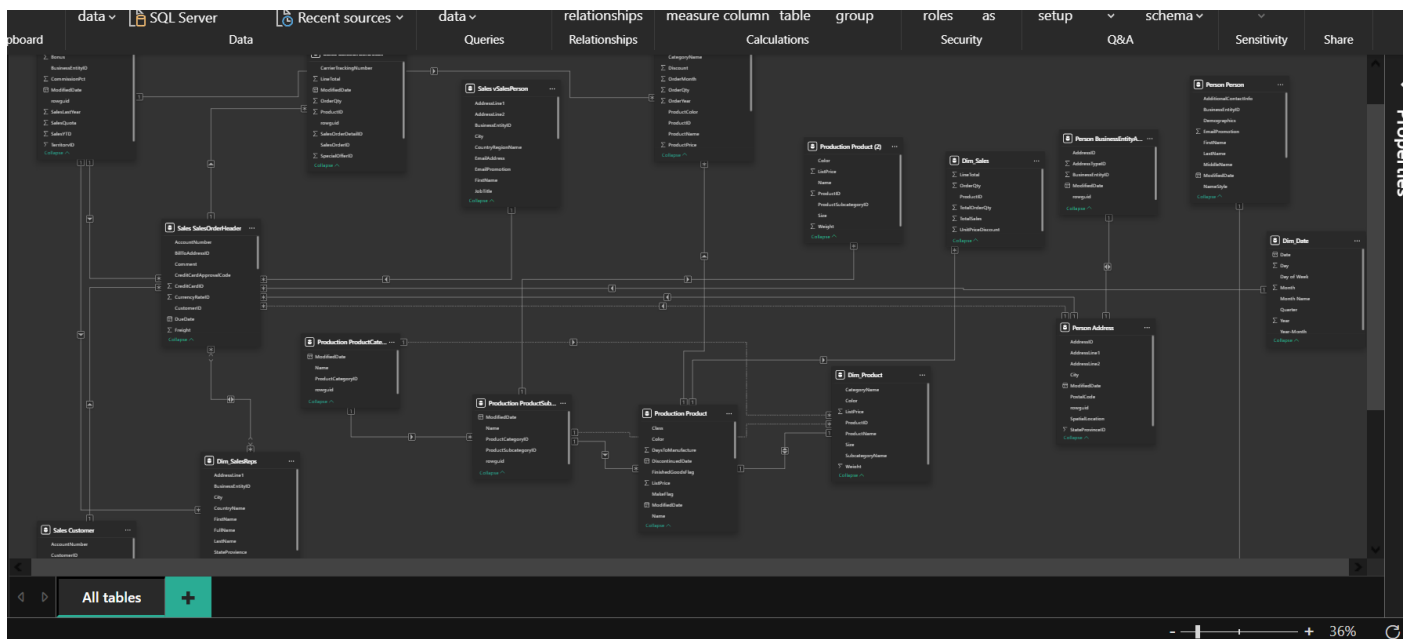
AddQuarter

✕ AddYearMonth

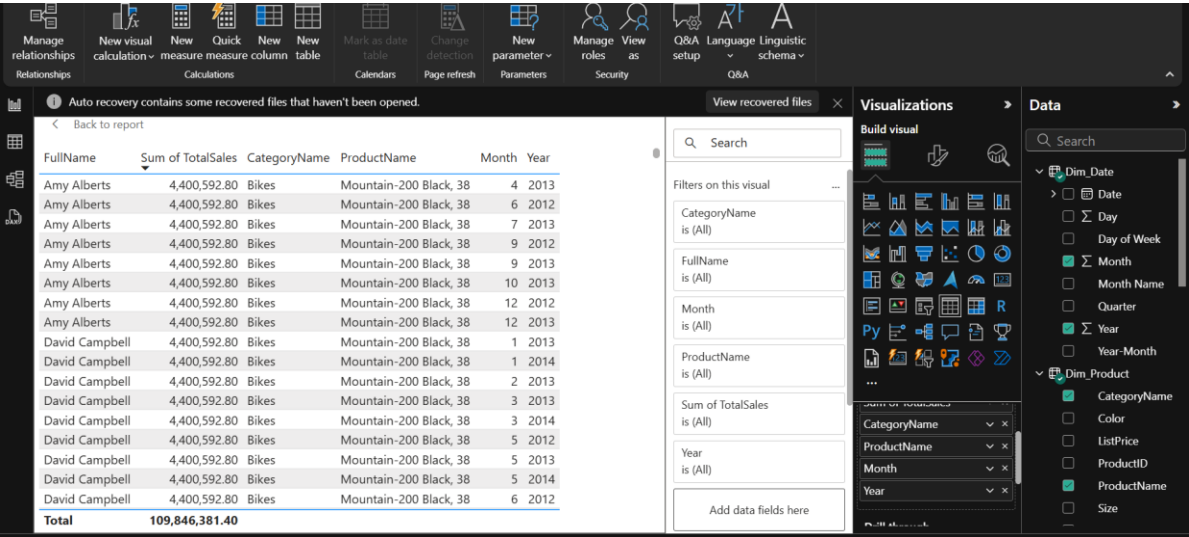
To create the table that contains each row represent a unique sales I started by merging the SaleaOrderDetail with Product tables (inner JOIN using the ProductID). Then I used a group by to perform an aggregation function on the required columns.

	ProductID	LineTotal	OrderQty	UnitPriceDiscount	TotalSales	TotalOrderQty
1	716	59.988	2	0.00	1079.784	36
2	716	57.6808	2	0.00	5479.676	190
3	716	89.982	3	0.00	2339.532	78
4	716	115.3616	4	0.00	8075.312	280
5	716	28.8404	1	0.00	2682.1572	93
6	716	29.994	1	0.00	479.904	16
7	716	149.97	5	0.00	6598.68	220
8	716	201.8828	7	0.00	2624.4764	91
9	716	173.0424	6	0.00	3979.9752	138
10	716	299.94	10	0.00	1199.76	40
11	716	119.976	4	0.00	3599.28	120
12	716	269.946	9	0.00	2969.406	99
13	716	49.99	1	0.00	20645.87	413
14	716	86.5212	3	0.00	7354.302	255
15	716	239.952	8	0.00	2399.52	80

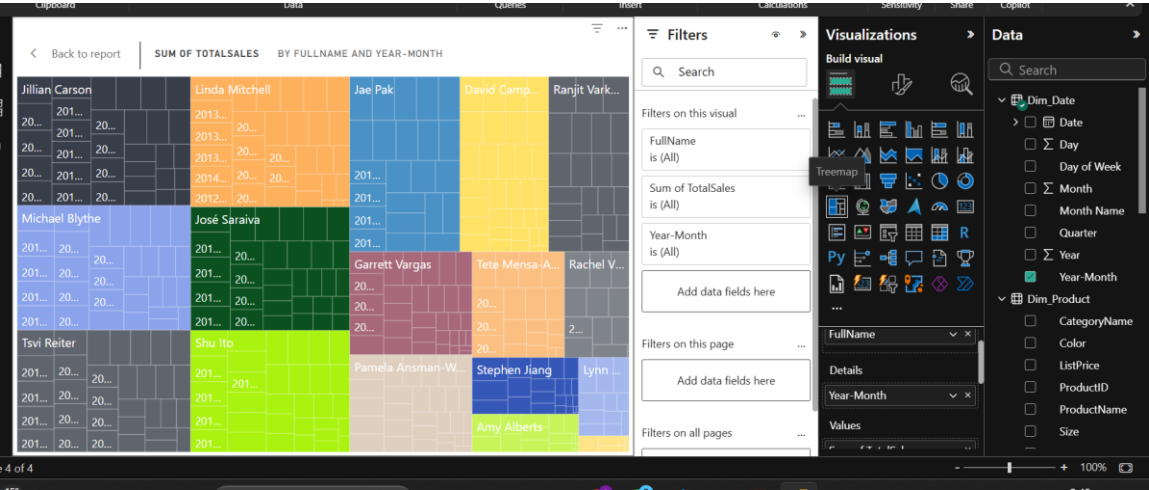
Later on I assigned and managed the new relationships between the new dimensional and fact tables with the database tables.



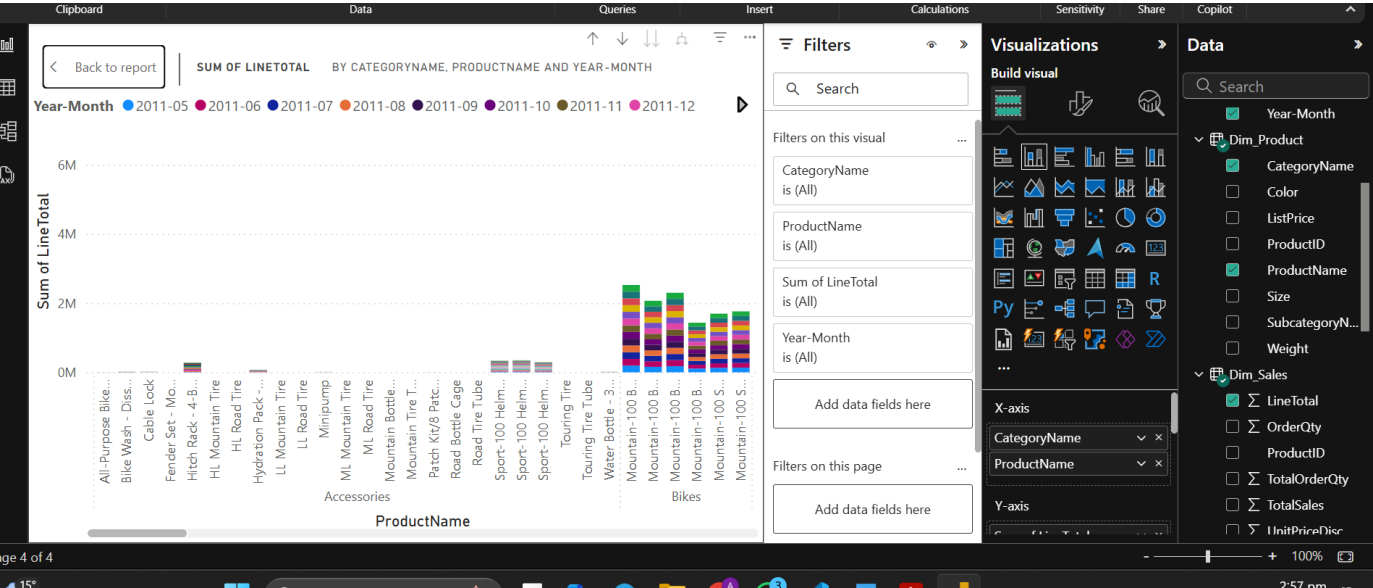
Here I implemented visualization for the dimensional and fact tables



Here the visualization to show the performance of the sales representative with categorizing their sales based on the month and year.



This visualization shows the most sold product with details about the time frame and the category and the sum of the total sales



- Task 4:

First, I will create a Pre-Christmas flag in the Date Table to mark sales from December 1st to 24th. Then, I will link the Date Table to the Sales Table using OrderDate. Next, I will add a DAX measure to calculate total sales for Pre-Christmas days. After that, I will create a bar chart to compare sales by product category and a line chart to see trends over time. Finally, I will add a year slicer so I can filter by year.