Sales Analysis for Automobile Company

**Name: Adit Deshpande**

**Student No: 10505324**

**Name: Ali Asgar**

**Student No: 10532445**

**Name: Natraj**

**Student No: 10534918**

Contents

[1. Introduction 3](#_Toc37693588)

[1.2. Reasons for selecting the subject area AND DATA 3](#_Toc37693589)

[1.3. Vision and Goals 4](#_Toc37693590)

[1.4. Key StakeHolders 4](#_Toc37693591)

[1.5. Business requirements 5](#_Toc37693592)

[1.5. Entity Relation Diagram 6](#_Toc37693593)

[2. SCHEMA 6](#_Toc37693594)

[3. ETL 8](#_Toc37693595)

[ETL PROCESS 1: Employee\_Dim 8](#_Toc37693596)

[ETL Process 2: Customer\_Dim 8](#_Toc37693597)

[ETL Process 3 9](#_Toc37693598)

[ETL Process 4: Product\_Dim 9](#_Toc37693599)

[ETL Process 5: Sales\_Fact 9](#_Toc37693600)

[ETL Process 6: Employee\_Fact 10](#_Toc37693601)

[4. VISUALIZATIONS AND REPORTS 11](#_Toc37693602)

[4.1. Visualizations 11](#_Toc37693603)

[4.1.1 Visualization 1. 11](#_Toc37693604)

[4.1.2 Visualization 2 12](#_Toc37693605)

[4.1.3 Visualization 3 13](#_Toc37693606)

[4.1.4 Visualization 4 14](#_Toc37693607)

[4.2. Reports 15](#_Toc37693608)

[4.2.1 Report 1 15](#_Toc37693609)

[4.2.2 Report 2 16](#_Toc37693610)

[4.2.3 Report 3 17](#_Toc37693611)

[4.2.4 Report 4 19](#_Toc37693612)

[5. Graph Databases 23](#_Toc37693613)

[5.1.1 Graph 1 23](#_Toc37693614)

[5.1.2 Graph 2 24](#_Toc37693615)

[5.1.3 Graph 3 25](#_Toc37693616)

[5.1.4 Graph 4 26](#_Toc37693617)

[5.1.5 Graph 5 27](#_Toc37693618)

[5.1.6 Graph 6 28](#_Toc37693619)

[5.1.7 Graph 7 29](#_Toc37693620)

[5.2 COMAPRISON to realtional databases 30](#_Toc37693621)

[6. Conclusions 30](#_Toc37693622)

[7. Bibliography 31](#_Toc37693623)

[Appendix A – SQL Queries and VISUALIZATIONS Code 31](#_Toc37693624)

[Appendix B – Neo 4J code and Sql Queries 35](#_Toc37693625)

# 1. Introduction

Sales reports describes sales details of a company. This company is a fleet management company and deals with sales of large-scale vehicles to various other companies. Fleet management means buying and selling of large number of vehicles. The company buys vehicles from various vendors from all around the world and sell those vehicles to the customer companies situated in different countries who require many numbers of vehicles. This process includes many aspects like customers, vendors, products or vehicles, employees who sale the products. Therefore, a report is needed to analyze various measures involved. The reports are focused on previously mentioned aspects like Customers: Companies which buy vehicles in large amount

Vendors: Companies which provide vehicles to the fleet management company and the sales representative

Employees: Employees which are responsible for the sales these vehicles and work for the fleet management company in various countries.

Data warehousing and creating dimensional schema provides the proper insights about the sales of the company. Once the dimensional model is ready various reports can be generated. These reports help the company to make strategies and improve the sales wherever required.

The statistical reports are easy to read and understand by the company executives. The reports can give following information sale of a product in a country, a product which is popular in the country, progress made by the employee, etc. Using this information profit and loss can also be calculated.

## 1.2. Reasons for selecting the subject area AND DATA

The key intention of the study was to analyze and understand the behavior of customers buying Vehicles in different parts of the world. Also, to figure out which Sales representative is making progress and provides gainful business to the company

The focus of the study was to analyze and gain intuition of the sales, Country which sold the most vehicles and Revenue generated. The other reason for selecting this dataset was to understand which product of the vendor most popular or which vendor is beneficial to the company.

## 1.3. Vision and Goals

Monitoring Revenue Generated:

Company would be able to monitor the revenue generated by different factors. If there is less sale country, then the company can prepare strategies in order to increase sale in that country. Talking about the sales representative employee, if an employee is doing well and the sales by his side is more then he/she can be awarded and similarly the employee whose sale is less he can be boosted or training can be provided to him to increase his/her performance. Also, if the demand of products from the some of the vendors is more then the company can approach with some mutual beneficiary schemes so that vendor and company both earns profits. Thus, revenue generated depending on different factors many decisions can be made by company to achieve profits and increase in progressive way.

Expanding in different Countries:

Analyzing the sales in different countries, company could make new strategies to improve sales in countries where the sales is less. Also, the business can be expanded in other different countries examining customer’s needs.

## 1.4. Key StakeHolders

The Key Stake holders is the Automobile Company as we are analyzing the automobile data, the key reason for this analysis to understand the sales trend and how much sale done by a single vendor to analyze the performance of the vendors. Sales by country will give the company a clear idea as to how and on what scale do, they must expand in that particular country.

## 1.5. Business requirements

* Analysis of Sales Data for the following purpose:  
  To generate the report on the maximum and minimum sales so that it gives an idea in which city the company can expand and hire more individuals for the Sales role.

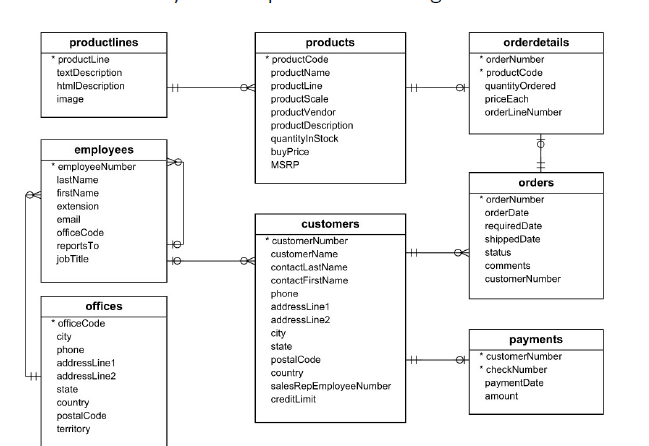
To generate report on the best selling product and promote that product in other cities and countries to boost sales.

* Analysis of Employee Performance:  
  Reports generated from this data will help the company track the performance of the Employee. The company will be able to track the sales made by each employee and the revenue generated by them.

This will help analyze the performance of the employee based on sales and revenue generated and ultimately can be promoted to senior positions to manage the department.

* Analysis of Customers Buying behavior:  
  By the Sales data we can analyze on which day the customer bought the product. Whether it was a weekday or a weekend or any special occasion. Most of the products are sold on the weekday and by this result we can either hire more staff to attend customers or open other branches so that customers can visit other centers.

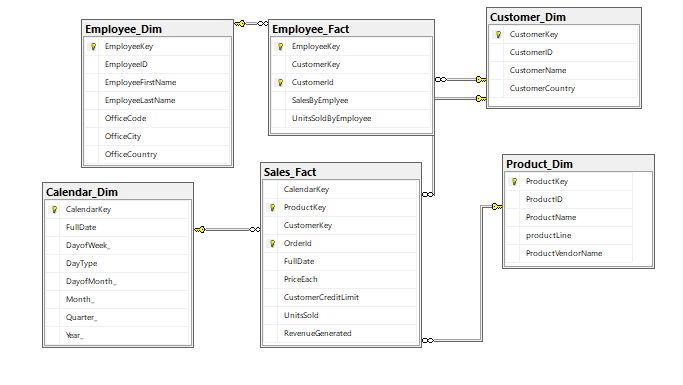
## 1.5. Entity Relation Diagram



# 2. SCHEMA

We have selected sample dataset of a fleet management company Dimensions and fact tables are created in Microsoft SQL Server Management Studio using SQL queries. Refer appendix for SQL queries. The data is separated into dimension tables and fact tables.

Depending on the reasons mentioned above the data set is divided between four dimension tables and two fact tables.



Following are the four dimensions:

|  |  |
| --- | --- |
| Employee\_Dim | Displays employee details |
| Customer\_Dim | Displays information related to customer |
| Calendar\_Dim | Displays timeline of the dataset |
| Product\_Dim | Displays list of products and the name of vendor |

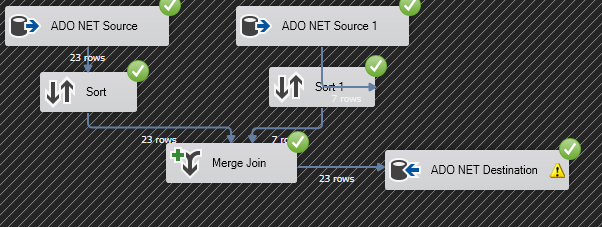
Following are two fact tables

|  |  |
| --- | --- |
| Sales\_Fact | Displays information related to sales |
| Employee\_Fact | Displays information related Employees |

# 3. ETL

For the process of Extraction, Transformation and Loading we have used Microsoft Visual Studio SSIS.

## ETL PROCESS 1: Employee\_Dim



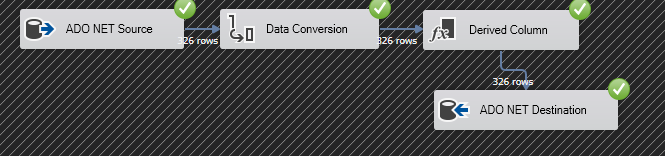
In this ETL process, the data is loaded into Employee\_Dim table using dataset. Here, Data from two tables employees and offices are sorted and joined to the destination.

## ETL Process 2: Customer\_Dim



In this ETL process, the data is loaded into Customer\_Dim from the table customers.

## ETL Process 3



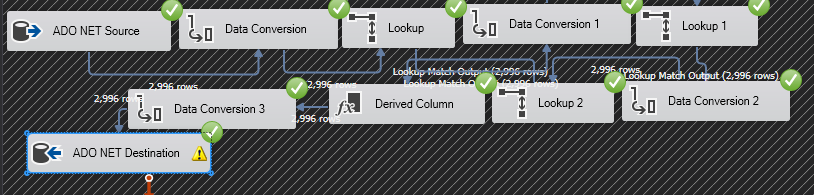
In this ETL process, the data is loaded into Calendar\_Dim using orderedDate field of orders table, the data type of field is converted to match the columns in dimension table. Some columns are derived using orderDate and loaded in Calendar\_Dim.

## ETL Process 4: Product\_Dim



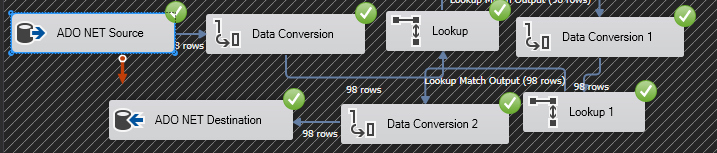
In this ETL process, the data is loaded in Product\_Dim from the table products.

## ETL Process 5: Sales\_Fact



In this ETL process, data from Calendar\_Dim, Customer\_Dim and Product\_Dim are loaded in Sales\_Fact using look up and Data Conversion. Lookup is used to extract the data from the source dimension and pass it to the destination fact.

## ETL Process 6: Employee\_Fact

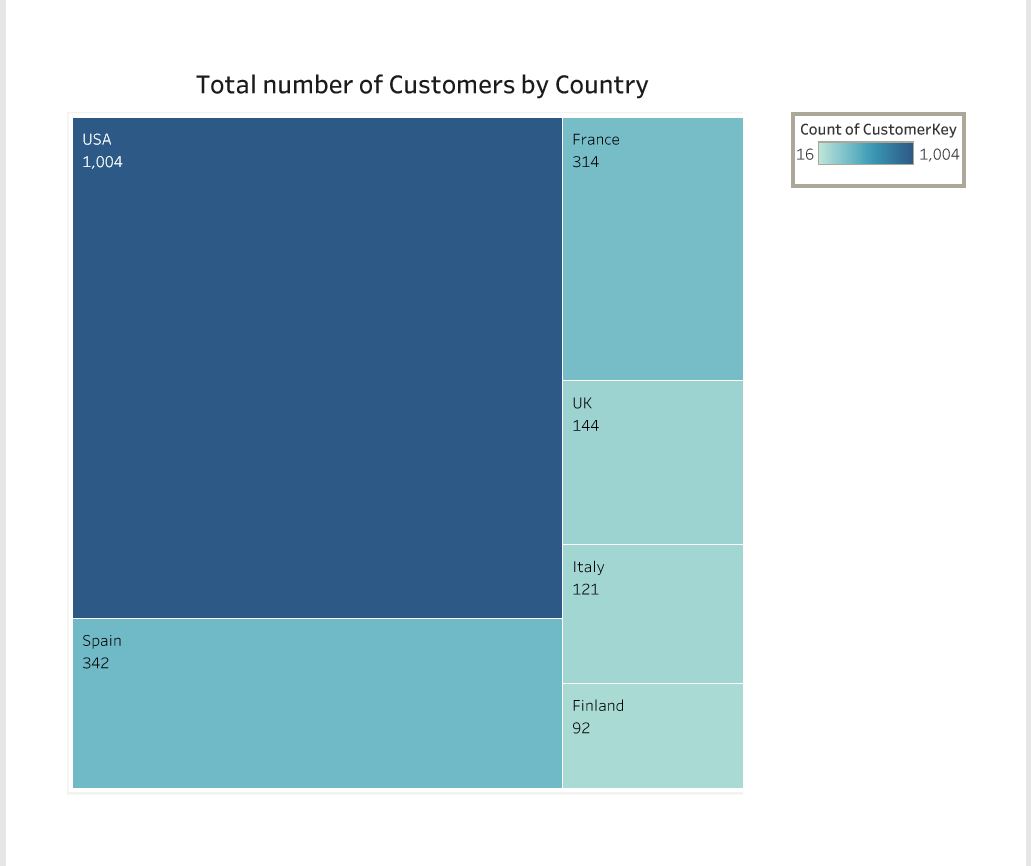


In this ETL process, data from Customer\_Dim and Employee\_Dim are loaded in Employee\_Fact using look up and Data Conversion. Lookup is used to extract the data from the source dimension and pass it to the destination fact.

# 4. VISUALIZATIONS AND REPORTS

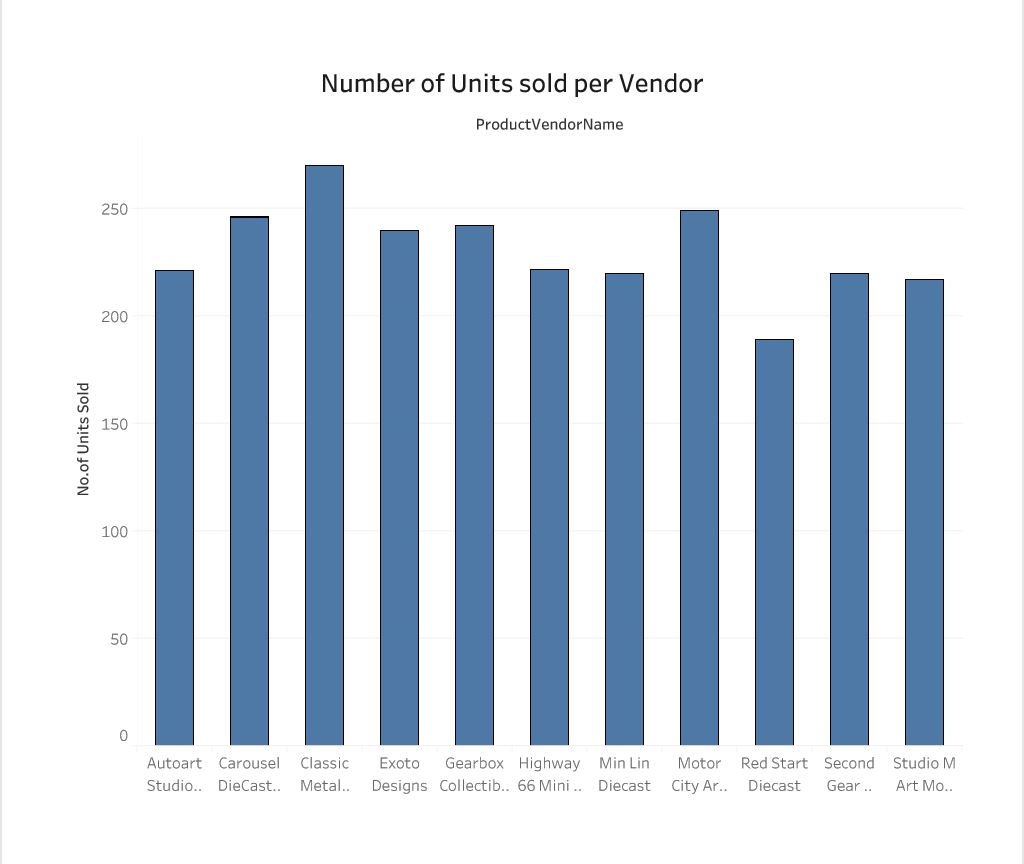
## 4.1. Visualizations

## 4.1.1 Visualization 1.



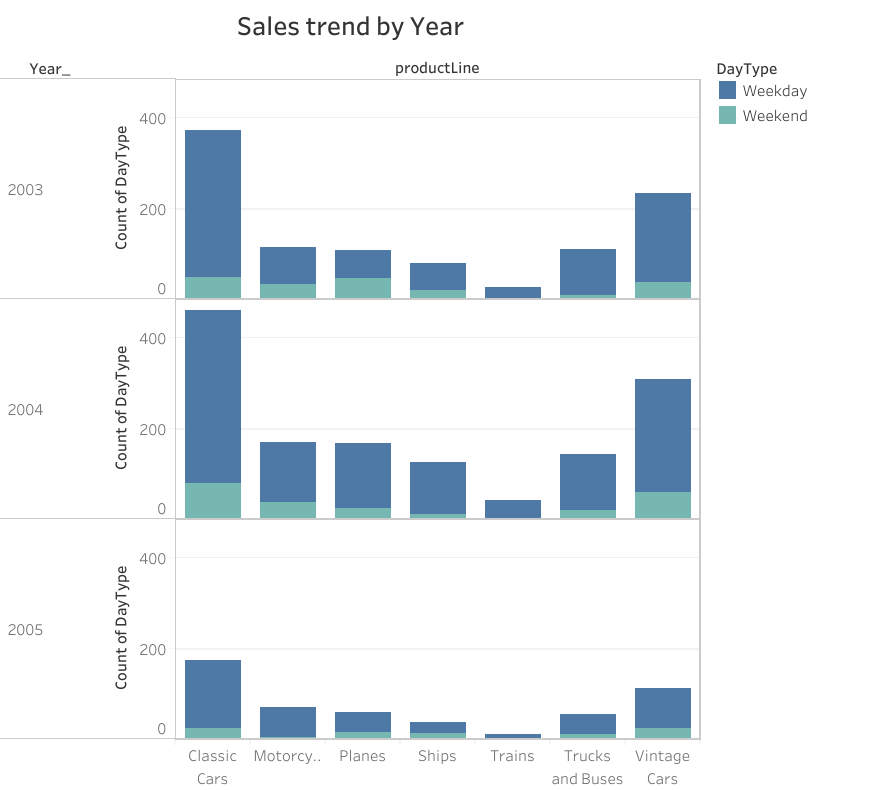
Here we are visualizing the total number of customers by country. As most of our customers are from USA and Europe that is the reason, we have used Tree map to get a clear idea and count of the customers. We can see in the above diagram that most of the customers are from the United States, Spain, France and other parts of Europe.

## 4.1.2 Visualization 2



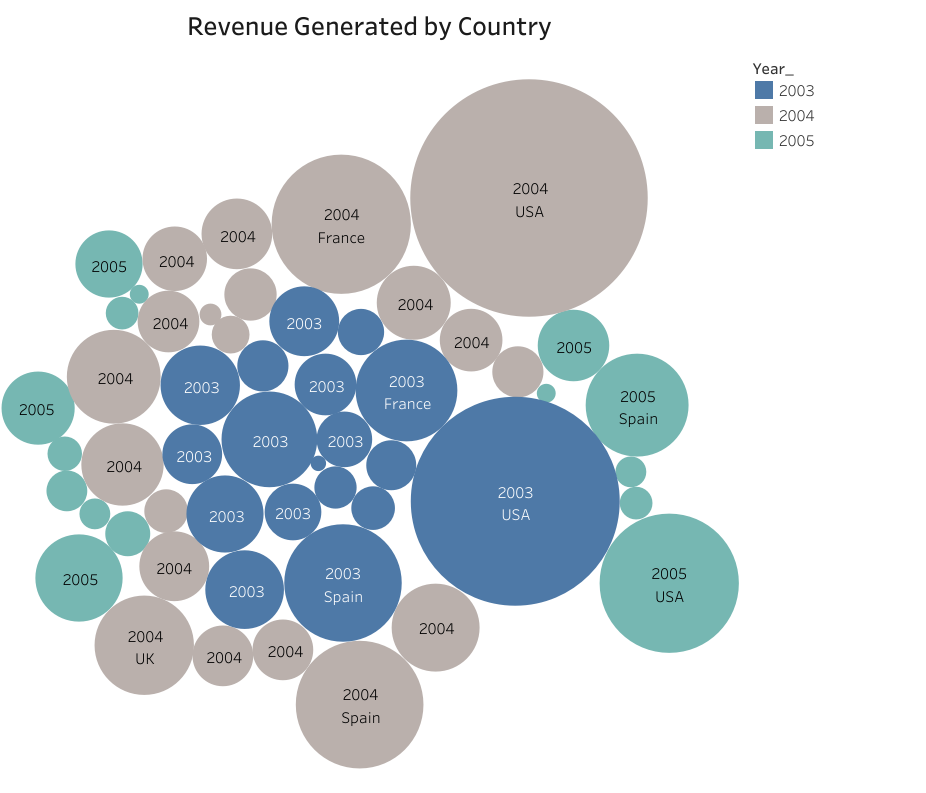
Here we are visualizing the Number of Units sold per Vendor. The bar graph shows the total number of units sold by the vendors. On the X- axis are the name of the Vendors. By the above graph we can clearly see number of units sold by each vendor. We can see that Classic Metal Creations has sold the maximum number of units.

## 4.1.3 Visualization 3

.

Here we are visualizing Sales by year. The dataset has the sales data of 3 years. We have used bar graph to visualize the data. The graph shows the Sales trend by Weekday or Weekend. By this we can predict the most sales happening on Weekend or Weekday. As we can see in the above graph that for the year 2003, Classic Cars were sold mostly on the Weekday and Planes were mostly sold on the weekends. Most of the vehicles were sold on the Weekdays more than weekends this means we can boost the productivity in the Weekdays rather than Weekends.

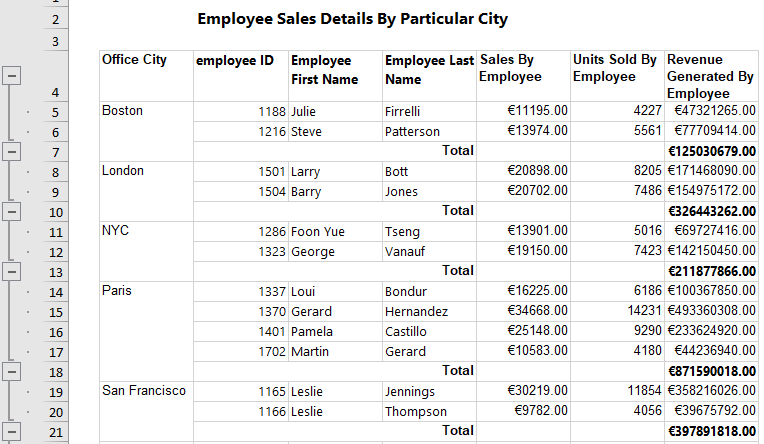
## 4.1.4 Visualization 4



The above graph shows the Revenue Generated by country. We have used Bubble graph to visualize this data. The bigger bubble shows name and year of the country which has generated the most revenue in that particular year. There are 3 different colors used for years. Bigger bubble shows the most revenue generated by year, similarly the smaller bubbles show the least revenue generated.

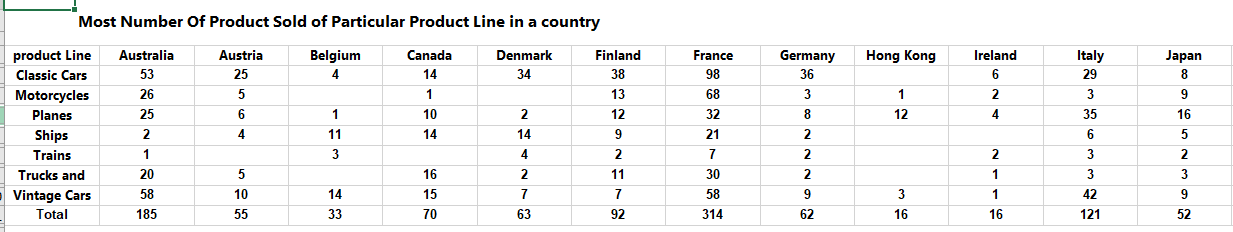
# 4.2. Reports

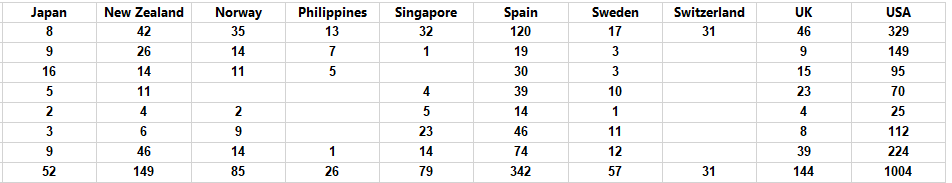
## 4.2.1 Report 1



This is the report of sales made by employees in different cities. Here we can see that as there are a greater number of employees working in Paris Sales in Paris is more than any of the cities. We can conclude that if the number of employees is increased in Tokyo then there are chances of generating more revenue.

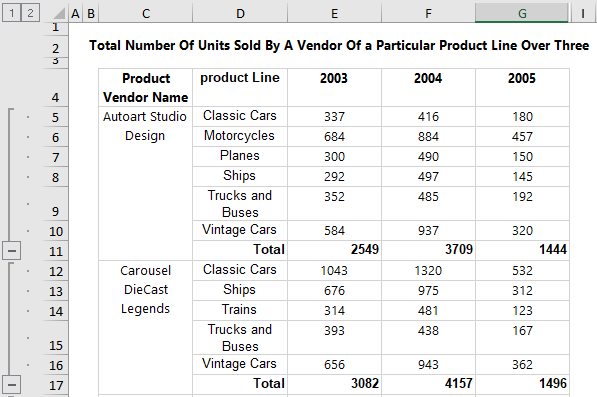
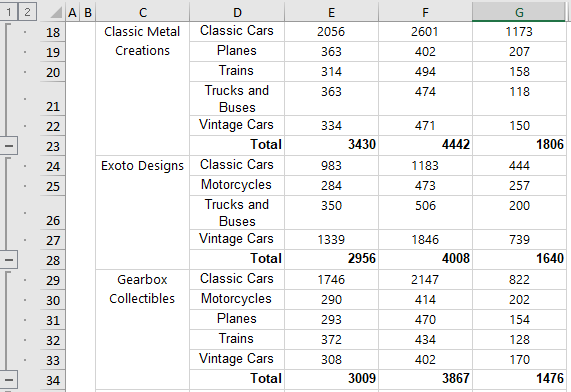
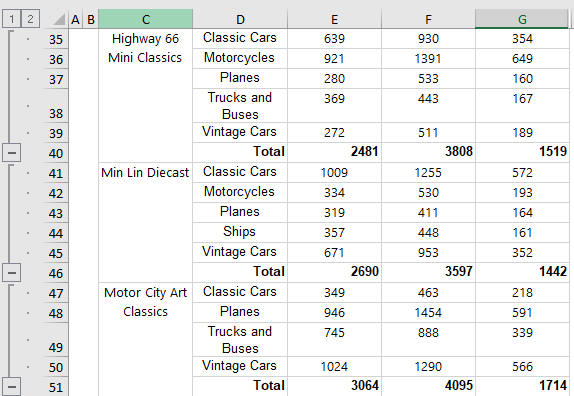
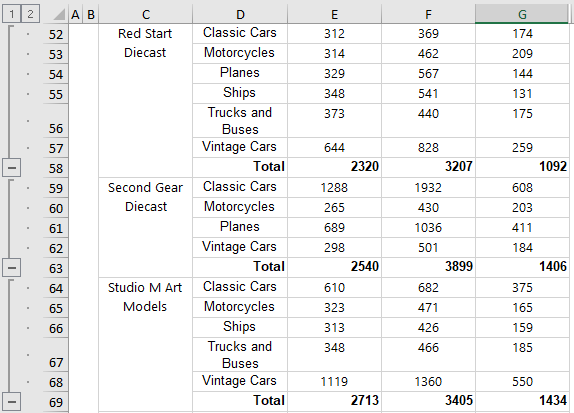
## 4.2.2 Report 2

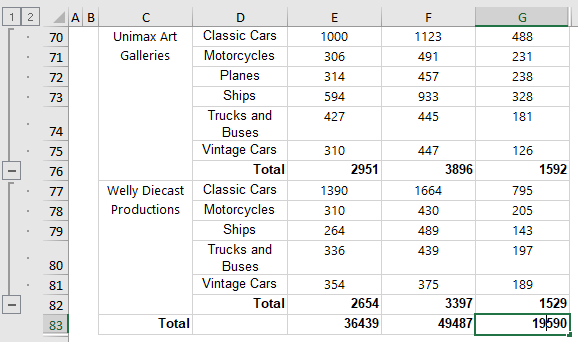




This is the report for sales of the product of different product lines in different countries. From this report, the company can try to promote the products in countries where the sales are less. For, example Hong Kong is a tourist place also a coastal area, considering this product like classic cars and ships and be promoted over there to increase the sales.

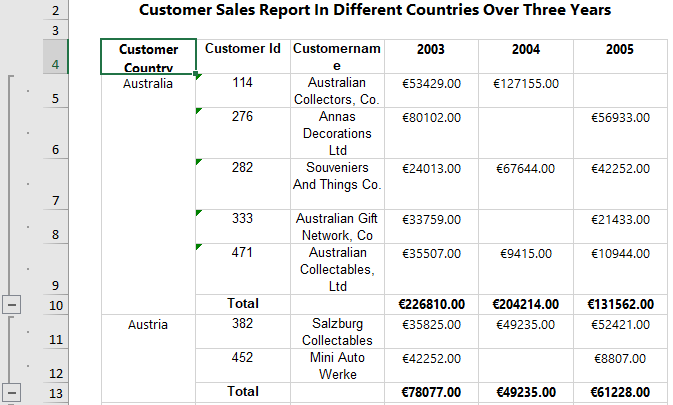
## 4.2.3 Report 3

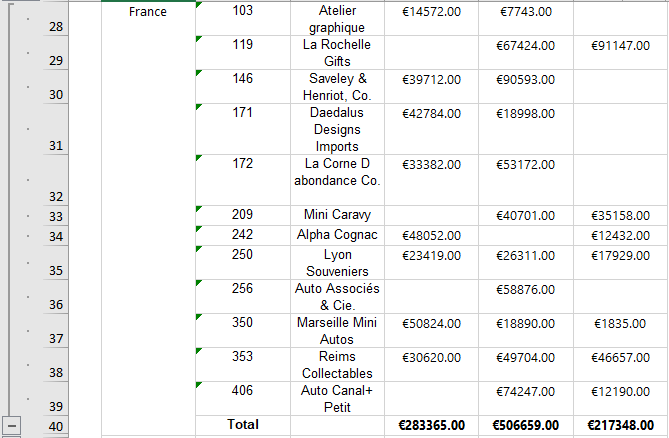


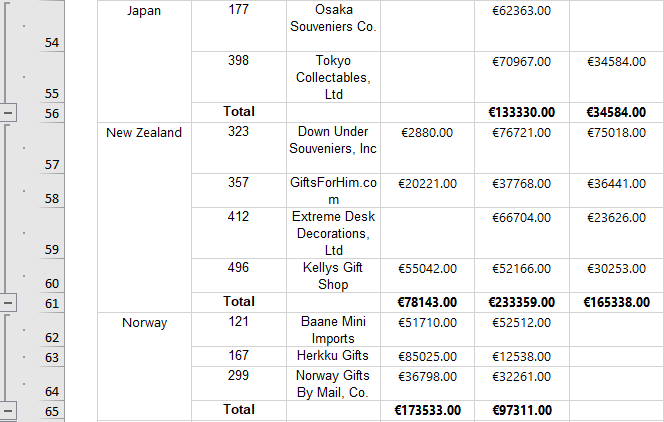
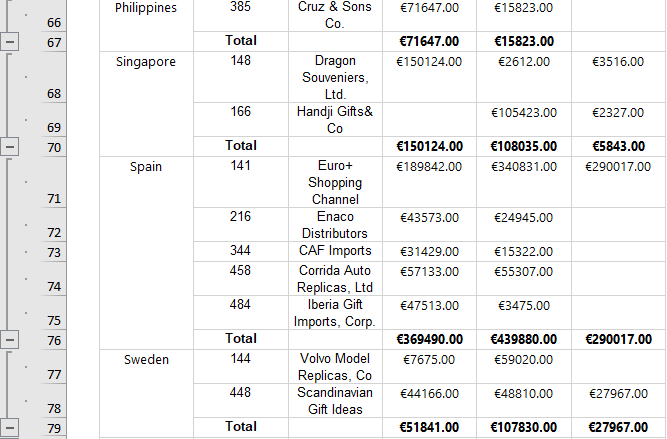
This report is of a number of units of different product lines sold by different vendors to the company over a period of three years from 2003 to 2005. From this company can compare the products bought from the vendors and take necessary measures which would be beneficial to both vendors and the company.

## 4.2.4 Report 4

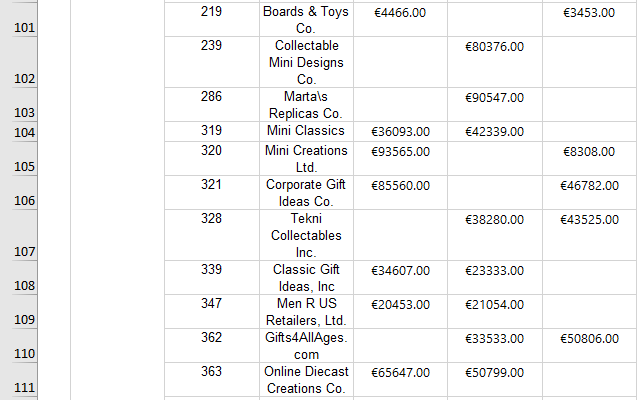






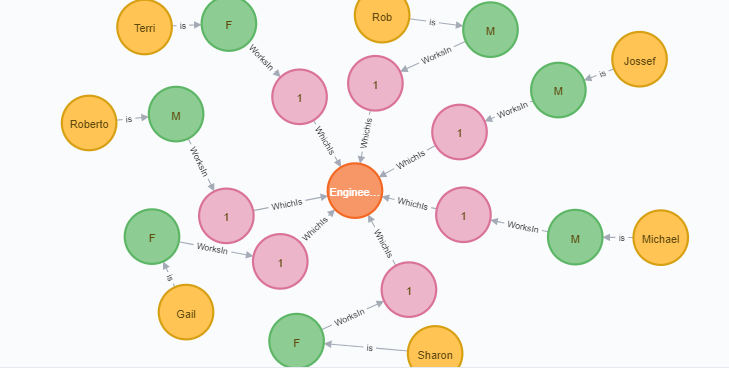


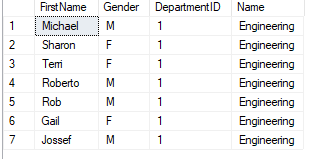


This report shows information about customer's sales in different countries. From the report we can see that in some countries the company has only one customer. The Company can look at this situation as an opportunity to develop more business in countries like Hong Kong, Ireland, Switzerland, etc.

# 5. Graph Databases

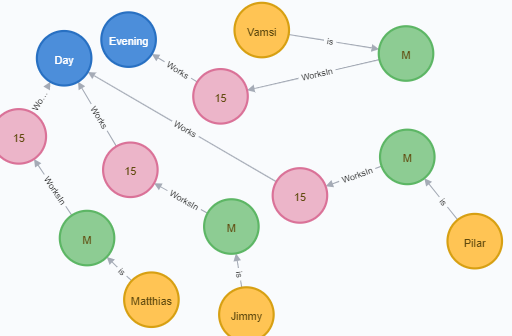
## 5.1.1 Graph 1

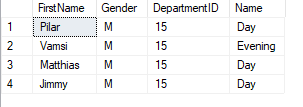




This Graph shows the Employee categorized into Male and Female who works in the Engineering Department. There are two nodes that are from the Department table and one node from the Person table.

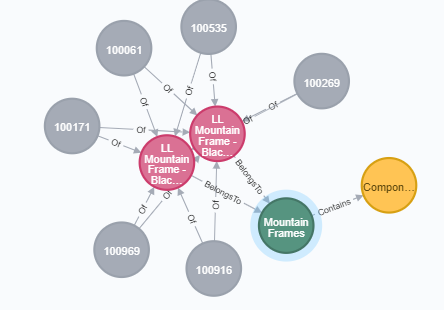
## 5.1.2 Graph 2

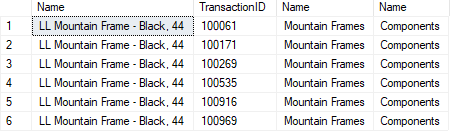




This Graph shows shifts in Male employees whose DepartmentID is 15. It is very clear from the Graph that 3 of the employees do Day shift and one employee does evening shift.

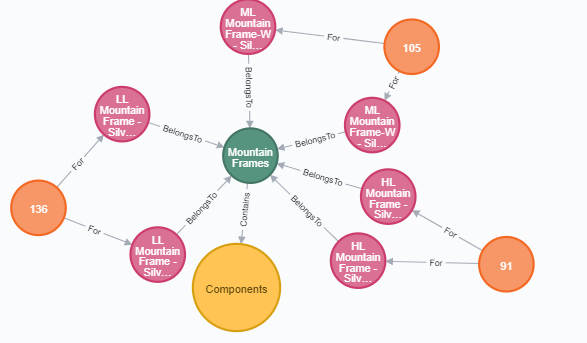
## 5.1.3 Graph 3

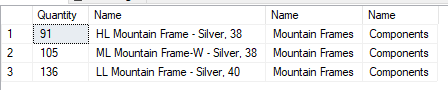




This Graph Shows Transaction History of a Product’s Category and Subcategory. The grey node has TransactionID of Name LL Mountain Frame Black which belongs to Sub Category Mountain Frames and Category Components.

## 5.1.4 Graph 4

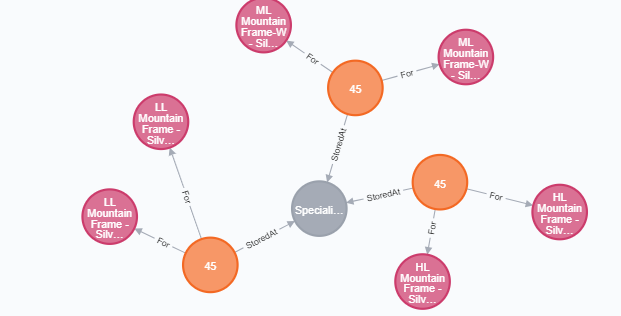


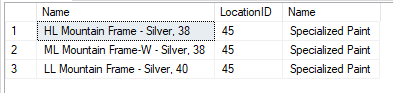


The above graph shows the Category, Sub Category, and Name of the Product with the Quantity. Mountain Frames is a Category that has LL Mountain Silver as Sub Category of Product Component and the

Quantityof those products are in the outer nodes. The Relationship gives a clear idea of Category and Subcategory.

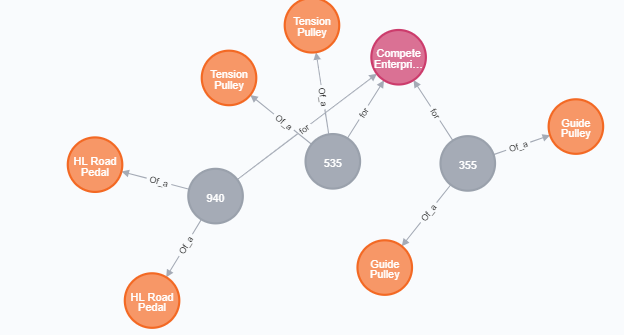
## 5.1.5 Graph 5

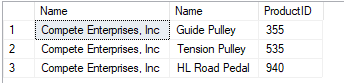




This Graph shows the Category of Products and the location at which they are stored. For the above graph “LL Mountain Frame Silver” is stored at Location ID “45” which is “Specialized Paint”.

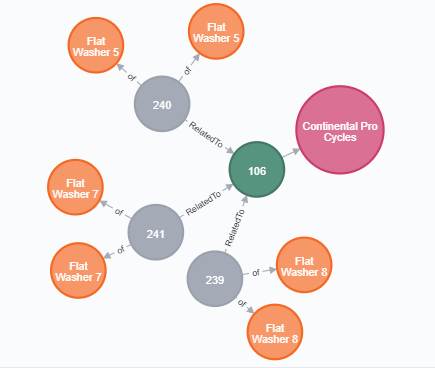
## 5.1.6 Graph 6

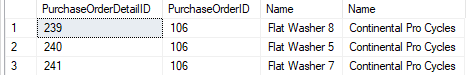




The above graph is a relationship about an order for the Purchasing Vendor Complete Enterprise. “Guide Pulley” of Product ID “355” has ordered by the company “Compete Enterprise”.

## 5.1.7 Graph 7





The graph shows and order placed by a vendor. The difference between the previous graph and this graph is that in this particular graph there is two Product ID. One is a ProductOrderDetailID and the other is ProductOrderID which is like an ID of an ID. For Example, For the order of “Flat Washer 7” of Order Detail ID “241” which is related to Order ID “106” was placed by the Purchasing Vendor “Continental Pro Cycles”.

## 5.2 COMAPRISON to realtional databases

* The key difference between relational database and graphical data base is that the relationships are sorted at different levels in graphical database whereas in relational database the structure is defined at in tables.
* A relational database is much faster as compared to graphical database when operating on huge volume of data or records. But in graphical database each record must be checked individually in a query in order to obtain structure or the graph of the data.
* Relational database uses less space as compared to graphical database because relational database does not store all the relationships

# 6. Conclusions

We took the Sales data of Automobile Company and performed ETL on the Dataset. Generated SSRS reports which give insights about the company’s Sales and Performance of the Vendors which will be helpful for the company to take many business decisions. The visualization reports can be used to study and take decisions to expand the business in other parts. With the help of Sales reports company can analyze which Vendor is performing so that it can expand in that City or State

# 7. Bibliography

Website: Mysqltutorial

URL: <https://www.mysqltutorial.org/mysql-sample-database.aspx>

Neo4j Documentation: <https://neo4j.com/docs/>

# Appendix A – SQL Queries and VISUALIZATIONS Code

**SQL Queries:**

create database CA\_DW

use CA\_DW

CREATE TABLE Product\_Dim

( ProductKey INT NOT NULL IDENTITY,

ProductID VARCHAR(255),

ProductName VARCHAR(255),

productLine VARCHAR(255),

ProductVendorName VARCHAR(255),

PRIMARY KEY (ProductKey));

GO

CREATE TABLE Customer\_Dim

(CustomerKey INT NOT NUll IDENTITY,

CustomerID VARCHAR(255),

CustomerName CHAR(255),

CustomerCountry CHAR(255),

PRIMARY KEY (CustomerKey));

GO

CREATE TABLE Employee\_Dim

(EmployeeKey INT NOT NUll IDENTITY,

EmployeeID INT,

EmployeeFirstName CHAR(255),

EmployeeLastName CHAR(255),

OfficeCode INT,

OfficeCity VARCHAR(255),

OfficeCountry VARCHAR(255),

PRIMARY KEY (EmployeeKey));

GO

CREATE TABLE Calendar\_Dim

(

CalendarKey INT NOT NULL IDENTITY,

FullDate DATE,

DayofWeek\_ CHAR(255),

DayType CHAR(255),

DayofMonth\_ INT,

Month\_ CHAR(255),

Quarter\_ CHAR(255),

Year\_ INT,

PRIMARY KEY (CalendarKey));

GO

GO

CREATE TABLE Sales\_Fact

(

CalendarKey INT,

ProductKey INT,

CustomerKey INT,

OrderId VARCHAR(10),

FullDate DATE,

PriceEach INT,

CustomerCreditLimit INT,

UnitsSold INT,

RevenueGenerated INT,

PRIMARY KEY(ProductKey, OrderId),

FOREIGN KEY (Calendarkey) REFERENCES Calendar\_Dim (CalendarKey),

FOREIGN KEY (Productkey) REFERENCES Product\_Dim (ProductKey),

FOREIGN KEY (CustomerKey) REFERENCES Customer\_Dim(CustomerKey)

);

CREATE TABLE Employee\_Fact

(

EmployeeKey INT,

CustomerKey INT,

CustomerId INT,

SalesByEmplyee INT,

UnitsSoldByEmployee INT,

PRIMARY KEY(EmployeeKey,CustomerId),

FOREIGN KEY (EmployeeKey) REFERENCES Employee\_Dim (EmployeeKey),

FOREIGN KEY (CustomerKey) REFERENCES Customer\_Dim(CustomerKey)

);

**Visualization Code:**CREATE PROC [dbo].[CAQueryReport1]

AS

select employeeID, EmployeeFirstName,EmployeeLastName,sum(SalesByEmplyee) as SalesByEmployee,sum(UnitsSoldByEmployee) as UnitsSoldByEmployee,OfficeCity

from Employee\_Dim ED, Employee\_Fact EF where ED.EmployeeKey=EF.EmployeeKey

group by EmployeeID, EmployeeFirstName,EmployeeLastName,OfficeCity

GO

CREATE PROC [dbo].[CAQueryReport2]

AS

select productLine, CustomerCountry,ProductName, count(productLine) as NoOfProducts from Product\_Dim PD, Customer\_Dim CD, Sales\_Fact SF

where PD.ProductKey=SF.ProductKey and CD.CustomerKey=SF.CustomerKey group by CustomerCountry,productLine,ProductName

GO

Create PROC [dbo].[CAQueryReport3]

AS

select productLine,ProductVendorName,Year\_,sum(UnitsSold) as NumberofUnitsSold from Product\_Dim PD, Calendar\_Dim CD, Sales\_Fact SF

where PD.ProductKey=SF.ProductKey and CD.CalendarKey=SF.CalendarKey group by productLine,ProductVendorName,Year\_

GO

Create PROC [dbo].[CAQueryReport4]

AS

Select CustomerId,sum(RevenueGenerated) RevenueGenerated,Year\_,CustomerCountry,Customername from Customer\_Dim CD,Sales\_Fact SF,Calendar\_Dim CAD

where CD.CustomerKey=SF.CustomerKey and CAD.CalendarKey=SF.CalendarKey group by CustomerId,Year\_,CustomerCountry,CustomerName

**Customers.CustomerID;**

# Appendix B – Neo 4J code and Sql Queries

Loading the CSV file in Neo4j.

LOAD CSV WITH HEADERS FROM "file:///HumanResources.Shift.csv" AS row CREATE (s:Shift) SET s=row{ShiftID:row.ShiftID,Name:row.Name,StartTime:row.StartTime,EndTime:row.EndTime,ModifiedDate:row.ModifiedDate} RETURN s

CREATE CONSTRAINT ON (s:Shift) ASSERT s.ShiftID IS UNIQUE

LOAD CSV WITH HEADERS FROM "file:///HumanResources.Department.csv" AS row CREATE (d:Department) SET d=row{DepartmentID:row.DepartmentID,Name:row.Name,GroupName:row.GroupName,ModifiedDate:row.ModifiedDate} RETURN d

CREATE CONSTRAINT ON (d:Department) ASSERT d.DepartmentID IS UNIQUE

LOAD CSV WITH HEADERS FROM "file:///HumanResources.EmployeeDepartmentHistory.csv" AS row CREATE (edh:EmployeeDepartmentHistory) SET edh=row{BusinessEntityID:row.BusinessEntityID,DepartmentID:row.DepartmentID,ShiftID:row.ShiftID,ModifiedDate:row.ModifiedDate,StartDate:row.StartDate,EndDate:row.EndDate} RETURN edh

CREATE INDEX ON :EmployeeDepartmentHistorydh(BusinessEntityID,DepartmentID,StartDate,ShiftID)

LOAD CSV WITH HEADERS FROM "file:///HumanResources.Employee.csv" AS row CREATE (e:Employee) SET e=row{BusinessEntityID:row.BusinessEntityID,NationalIDNumber:row.NationalIDNumber,LoginID:row.LoginID,OrganizationaNode:row.OrganizationNode,OrganizationLevel:row.OrganizationLevel,JobTitle:row.JobTitle,BirthDate:row.BirthDate,MaritalStatus:row.MaritalStatus,Gender:row.Gender,HireDate:row.HireDate,SalariedFlag:row.SalariedFlag,VacationHours:row.VacationHours,SickLeaveHours:row.SickLeaveHours,CurrentFlag:row.CurrentFlag,RowGuid:row.rowguid,ModifiedDate:row.ModifiedDate} RETURN e

CREATE CONSTRAINT ON (e:Employee) ASSERT e.BusinessEntityID IS UNIQUE

load csv with headers from "file:///HumanResources.EmployeePayHistory.csv" as row create (eph:EmployeePayHistory) set eph = row{BusinessEntityID:row.BusinessEntityID, RateChangeDate:row.RateChangeDate, Rate:row.Rate, PayFrequency: row.PayFrequency, ModifiedDate: row.ModifiedDate} return eph

CREATE INDEX ON :EmployeePayHistory(BusinessEntityID,RateChangeDate)

load csv with headers from "file:///Person.Person.csv" as row create (P:Person) set P = row{BusinessEntityID:row.BusinessEntityID,PersonType:row.PersonType,NameStyle:row.NameStyle,Title:row.Title,FirstName:row.FirstName,MiddleName:row.MiddleName,LastName:row.LastName,Suffix:row.Suffix,EmailPromotion:row.EmailPromotion,AdditionalContactInfo:row.AdditionalContactInfo,Demographics:row.Demographics,rowguid:row.rowguid,ModifiedDate:row.ModifiedDate} return P

CREATE CONSTRAINT ON (P:Person) ASSERT P.BusinessEntityID IS UNIQUE

LOAD CSV WITH HEADERS FROM "file:///Production.Product.csv" AS row CREATE (P:Product) SET P=row{ProductID:row.ProductID,Name:row.Name,ProductNumber:row.ProductNumber,MakeFlag:row.MakeFlag,FinishedGoodsFlag:row.FinishedGoodsFlag,Color:row.Color,SafetyStockLevel:row.SafetyStockLevel,ReorderPoint:row.ReorderPoint,StandardCost:row.StandardCost,ListPrice:row.ListPrice,Size:row.Size,SizeUnitMeasureCode:row.SizeUnitMeasureCode,WeightUnitMeasureCode:row.WeightUnitMeasureCode,Weight:row.Weight,DaysToManufacture:row.DaysToManufacture,ProductLine:row.ProductLine,Class:row.Class,Style:row.Style,ProductSubcategoryID:row.ProductSubcategoryID,ProductModelID:row.ProductModelID,SellStartDate:row.SellStartDate,SellEndDate:row.SellEndDate,DiscontinuedDate:row.DiscontinuedDate,rowguid:row.rowguid,ModifiedDate:row.ModifiedDate} RETURN P

LOAD CSV WITH HEADERS FROM "file:///Production.ProductSubCategory.csv" AS row CREATE (PSC:ProductSubCategory) SET PSC=row{ProductSubcategoryID:row.ProductSubcategoryID,ProductCategoryID:row.ProductCategoryID,Name:row.Name,rowguid:row.rowguid,ModifiedDate:row.ModifiedDate} RETURN PSC

LOAD CSV WITH HEADERS FROM "file:///Production.ProductCategory.csv" AS row CREATE (PC:ProductCategory) SET PC=row{ProductCategoryID:row.ProductCategoryID,Name:row.Name,rowguid:row.rowguid,ModifiedDate:row.ModifiedDate} RETURN PC

LOAD CSV WITH HEADERS FROM "file:///Production.TransactionHistory.csv" AS row CREATE (TH:TransactionHistory) SET TH=row{TransactionID:row.TransactionID,ProductID:row.ProductID,ReferenceOrderID:row.ReferenceOrderID,TransactionDate:row.TransactionDate,TransactionType:row.TransactionType,Quantity:row.Quantity,ActualCost:row.ActualCost,ModifiedDate:row.ModifiedDate,ReferenceOrderLineID:row.ReferenceOrderLineID} RETURN TH

LOAD CSV WITH HEADERS FROM "file:///Production.ProductInventory.csv" AS row CREATE (PI:ProductInventory) SET PI=row{ProductID:row.ProductID,LocationID:row.LocationID,Shelf:row.Shelf,Bin:row.Bin,Quantity:row.Quantity,rowguid:row.rowguid,ModifiedDate:row.ModifiedDate} RETURN PI

LOAD CSV WITH HEADERS FROM "file:///Production.Location.csv" AS row CREATE (L:Location) SET L=row{LocationID:row.LocationID,Name:row.Name,CostRate:row.CostRate,Availability:row.Availability,ModifiedDate:row.ModifiedDate} RETURN L

LOAD CSV WITH HEADERS FROM "file:///Purchasing.ProductVendor.csv" AS row CREATE (PV:ProductVendor) SET PV=row{BusinessEntityID:row.BusinessEntityID,ProductID:row.ProductID,AverageLeadTime:row.AverageLeadTime,StandardPrice:row.StandardPrice,LastReceiptCost:row.LastReceiptCost,LastReceiptDate:row.LastReceiptDate,MinOrderQty:row.MinOrderQty,MaxOrderQty:row.MaxOrderQty,OnOrderQty:row.OnOrderQty,UnitMeasureCode:row.UnitMeasureCode,ModifiedDate:row.ModifiedDate} RETURN PV

LOAD CSV WITH HEADERS FROM "file:///Purchasing.Vendor.csv" AS row CREATE (V:Vendor) SET V=row{BusinessEntityID:row.BusinessEntityID,AccountNumber:row.AccountNumber,Name:row.Name,CreditRating:row.CreditRating,PreferredVendorStatus:row.PreferredVendorStatus,ActiveFlag:row.ActiveFlag,PurchasingWebServiceURL:row.PurchasingWebServiceURL,ModifiedDate:row.ModifiedDate} RETURN V

LOAD CSV WITH HEADERS FROM "file:///Purchasing.PurchaseOrderDetail.csv" AS row CREATE (POD:PurchaseOrderDetail) SET POD=row{PurchaseOrderID:row.PurchaseOrderID,PurchaseOrderDetailID:row.PurchaseOrderDetailID,DueDate:row.DueDate,OrderQty:row.OrderQty,ProductID:row.ProductID,UnitPrice:row.UnitPrice,LineTotal:row.LineTotal,ReceivedQty:row.ReceivedQty,RejectedQty:row.RejectedQty,StockedQty:row.StockedQty,ModifiedDate:row.ModifiedDate} RETURN POD

LOAD CSV WITH HEADERS FROM "file:///Purchasing.PurchaseOrderHeader.csv" AS row CREATE (POH:PurchaseOrderHeader) SET POH=row{PurchaseOrderID:row.PurchaseOrderID,ShipMethodID:row.ShipMethodID,VendorID:row.VendorID,RevisionNumber:row.RevisionNumber,Status:row.Status,OrderDate:row.OrderDate,ShipDate:row.ShipDate,SubTotal:row.SubTotal,TaxAmt:row.taxAmt,Freight:row.Freight,TotalDue:row.TotalDue,ModifiedDate:row.ModifiedDate} RETURN POH

Relationship

MATCH(P:Person),(E:Employee) WHERE P.BusinessEntityID=E.BusinessEntityID CREATE (P)-[r:is]->(E) return P,r,E

MATCH(EDH:EmployeeDepartmentHistory),(E:Employee) WHERE EDH.BusinessEntityID=E.BusinessEntityID CREATE (E)-[r:WorksIn]->(EDH) return E,r,EDH

MATCH(EDH:EmployeeDepartmentHistory),(D:Department) WHERE EDH.DepartmentID=D.DepartmentID CREATE (EDH)-[r:WhichIs]->(D) return EDH,r,D

MATCH(EDH:EmployeeDepartmentHistory),(S:Shift) WHERE EDH.ShiftID=S.ShiftID CREATE (EDH)-[r:Works]->(S) return EDH,r,S

MATCH(EDH:EmployeeDepartmentHistory)-[r:Belongs]->(D:Department) return EDH,r,D

MATCH(P:Product),(PSC:ProductSubCategory) WHERE P.ProductSubcategoryID=PSC.ProductSubcategoryID CREATE (P)-[r:BelongsTo]->(PSC) return P,r,PSC

MATCH(PSC:ProductSubCategory),(PC:ProductCategory) WHERE PSC.ProductCategoryID=PC.ProductCategoryID CREATE (PSC)-[r:Contains]->(PC) return PSC,r,PC

MATCH(PI:ProductInventory),(P:Product) WHERE PI.ProductID=P.ProductID CREATE (PI)-[r:For]->(P) return PI,r,P

MATCH(PV:ProductVendor),(P:Product) WHERE PV.ProductID=P.ProductID CREATE (PV)-[r:Of\_a]->(P) return PV,r,P

MATCH(PV:ProductVendor),(V:Vendor) WHERE PV.BusinessEntityID=V.BusinessEntityID CREATE (PV)-[r:for]->(V) return PV,r,V

MATCH(POD:PurchaseOrderDetail),(P:Product) WHERE POD.ProductID=P.ProductID CREATE (POD)-[r:of]->(P) return POD,r,P

MATCH(POD:PurchaseOrderDetail),(POH:PurchaseOrderHeader) WHERE POD.PurchaseOrderID=POH.PurchaseOrderID CREATE (POD)-[r:RelatedTo]->(POH) return POD,r,POH

MATCH(POH:PurchaseOrderHeader),(V:Vendor) WHERE POH.VendorID=V.BusinessEntityID CREATE (POH)-[r:To]->(V) return POH,r,V

Query1

match(P:Person)-[a:is]->(E:Employee)-[r:WorksIn]->(EDH:EmployeeDepartmentHistory)-[b:WhichIs]->(D:Department{Name:"Engineering"}) return P,a,E,r,EDH,b,D

match(P:Person)-[a:is]->(E:Employee{Gender:"M"})-[r:WorksIn]->(EDH:EmployeeDepartmentHistory{DepartmentID:"15"})-[b:Works]->(S:Shift) return P,a,E,r,EDH,b,S

Query 2

match(P:Person)-[a:is]->(E:Employee{Gender:"M"})-[r:WorksIn]->(EDH:EmployeeDepartmentHistory{DepartmentID:"15"})-[b:Works]->(S:Shift) return P,a,E,r,EDH,b,S

Query 3

match(TH:TransactionHistory)-[a:Of]->(P:Product{ProductID:"925"})-[r:BelongsTo]->(PSC:ProductSubCategory)-[b:Contains]->(PC:ProductCategory) return TH,a,P,r,PSC,b,PC

Query 4

match(PI:ProductInventory{LocationID:"45"})-[a:For]->(P:Product)-[r:BelongsTo]->(PSC:ProductSubCategory)-[b:Contains]->(PC:ProductCategory) return PI,a,P,r,PSC,b,PC

Query 5

match(P:Product)<-[a:For]-(PI:ProductInventory{LocationID:"45"})-[r:StoredAt]->(L:Location) return P,a,PI,r,L

Query 6

match(V:Vendor)<-[a:for]-(PV:ProductVendor{BusinessEntityID:"1508"})-[r:Of\_a]->(P:Product) return V,a,PV,r,P

Query 7

Match(P:Product)<-[a:of]-(POD:PurchaseOrderDetail{PurchaseOrderID:"106"})-[r:RelatedTo]->(POH:PurchaseOrderHeader)-[b:To]->(V:Vendor) return P,a,POD,r,POH,b,V

**SQL Queries:**

Query 1

select [Person.Person].FirstName,Gender,[HumanResources.Department].DepartmentID,Name

from

[HumanResources.Department],[HumanResources.EmployeeDepartmentHistory],[HumanResources.Employee],[Person.Person]

where [HumanResources.Department].DepartmentID=[HumanResources.EmployeeDepartmentHistory].DepartmentID and

[HumanResources.EmployeeDepartmentHistory].BusinessEntityID=[HumanResources.Employee].BusinessEntityID and [HumanResources.Employee].BusinessEntityID=[Person.Person].BusinessEntityID and [HumanResources.Department].Name='Engineering'

Query 2

select [Person.Person].FirstName,[HumanResources.Employee].Gender,

[HumanResources.EmployeeDepartmentHistory].DepartmentID,[HumanResources.Shift].Name

from

[Person.Person],[HumanResources.Employee],[HumanResources.EmployeeDepartmentHistory],[HumanResources.Shift]

where [Person.Person].BusinessEntityID=[HumanResources.Employee].BusinessEntityID and

[HumanResources.Employee].BusinessEntityID=[HumanResources.EmployeeDepartmentHistory].BusinessEntityID and

[HumanResources.EmployeeDepartmentHistory].ShiftID=[HumanResources.Shift].ShiftID and

[HumanResources.Employee].Gender='M' and

[HumanResources.EmployeeDepartmentHistory].DepartmentID=15

Query 3

select [Production.Product].Name,[Production.TransactionHistory].TransactionID,[Production.ProductSubCategory].Name, [Production.ProductCategory].Name

from [Production.TransactionHistory],[Production.Product],[Production.ProductSubCategory],[Production.ProductCategory]

where [Production.TransactionHistory].ProductID=[Production.Product].ProductID and

[Production.Product].ProductSubcategoryID=[Production.ProductSubCategory].ProductSubcategoryID and

[Production.ProductSubCategory].ProductCategoryID=[Production.ProductCategory].ProductCategoryID and [Production.Product].ProductID=925

Query 4

Select [Production.ProductInventory].Quantity,[Production.Product].Name,[Production.ProductSubCategory].Name,[Production.ProductCategory].Name from [Production.ProductInventory],[Production.Product],[Production.ProductSubCategory],[Production.ProductCategory] where [Production.ProductInventory].ProductID=[Production.Product].ProductID and

[Production.Product].ProductSubcategoryID=[Production.ProductSubCategory].ProductSubcategoryID and [Production.ProductCategory].ProductCategoryID=[Production.ProductSubCategory].ProductCategoryID and LocationID=45

Query 5

Select [Production.Product].Name,[Production.ProductInventory].LocationID,[Production.Location].Name

from [Production.Product],[Production.ProductInventory],[Production.Location] where [Production.Product].ProductID=[Production.ProductInventory].ProductID and

[Production.ProductInventory].LocationID=[Production.Location].LocationID and [Production.ProductInventory].LocationID=45

Query 6

select [Purchasing.Vendor].Name,[Production.Product].Name,[Purchasing.ProductVendor].ProductID

from [Purchasing.Vendor],[Purchasing.ProductVendor],[Production.Product]

where [Purchasing.Vendor].BusinessEntityID=[Purchasing.ProductVendor].BusinessEntityID and [Purchasing.ProductVendor].ProductID=[Production.Product].ProductID and [Purchasing.ProductVendor].BusinessEntityID=1508

Query 7

select [Purchasing.PurchaseOrderDetail].PurchaseOrderDetailID,[Purchasing.PurchaseOrderHeader].PurchaseOrderID,

[Production.Product].Name, [Purchasing.Vendor].Name

from [Production.Product],[Purchasing.PurchaseOrderDetail],[Purchasing.PurchaseOrderHeader],[Purchasing.Vendor]

where [Production.Product].ProductID=[Purchasing.PurchaseOrderDetail].ProductID and [Purchasing.PurchaseOrderDetail].PurchaseOrderID=[Purchasing.PurchaseOrderHeader].PurchaseOrderID and [Purchasing.PurchaseOrderHeader].VendorID=[Purchasing.Vendor].BusinessEntityID and [Purchasing.PurchaseOrderDetail].PurchaseOrderID=106