



# 0. Abstract

This paper is discussing the dimensional data warehouse. In 1996 Ralph Kimball designed dimensional data ware house that is design or model to help people to develop data models in structure and visualize way. In this report we will see the tactics and strategies for the design, development and implementation of this structure using tools like SSDT, SSMS, R studio, Excel, Neo4j for different kinds of uses of handling and visualize the data. In addition, here will talk about many execution techniques of inserting data in to fact table, how to web scrap data using R, and different types of approaches to system as much smoother.

# 1. Background for the Development of chosen system

The Dataset I am using here is Northwind Trading company. This data set is easily avaiable online but best way to find out the dataaset is by Mircosoft offical site because this dataset is distributed to microsoft by Northwind company. So it non human manupulated and well structred relation dataset. As the dataset is relational it is quiet easy to build data warehouse and analyze, study dataset.

The dataset is the order transaction data of Northwind dataset that sells the human consumamable food product. Which is shipped to many country with their different shipping company partner. Northwind company has many vendor partner that fulfills the order requirement given by northwind company.

I selected this database for building sample warehouse application for three reasons:

1. It is relational data set so its easy for understanding and study about how actual relational database looks.
2. SQL server/SSDT/SSMS has OLEDB complaint, is very easy to import the data into server by importing the Excel file, ETL or SQL Queries that is available on the microsoft official site.
3. The most important reason is, it is excellent example of operational system of database which is used order processing transactional details.

# 2. Scope and Business Objectives

Data warehousing is very important concept for many people for implementing the simple data warehouse or data mart more correctly . This report clears the concept of how I have implemented and created the data warehouse. Ocassionally I will discuss the technical terms wherever necessary.

Here mostly we will talk about the implementation design stage of data warehouse, which inclusding the how data warehoue is important to us, why we created and what is the expectation of data warehouse. The scope of warehouse will demonstrate the business process, dimensional model and physical data model.

In this particular dataset, main focus is to answer all business requirement. Generally quries aries on the what are the different product? Where through it comes? Which one the top selling product? Is the shipping company trustworthy for delivering the product? What is the good are shippers and vendors? And etc.

All these kind of queries’s answer will get through this report.

In my ware house I am finding out the total revenue of suppliers(Vendors). Through this analysis we are ultimately finding out the which suppliers is on the demanding. i.e. which suppliers does work the most and how trust worth he is. We can guess the behaviour/relation of suppliers to his purchasing partner.

In second analysis I am finding out, Prominancy of shippers company. Like how many product they deliverd on\_time, before\_time and late.

Through this analysis northwind can take the decsion in the future, with which shippment company shipment contract should be continued or not? Or they need to add another shipment company if the products are more and getting product deleveries is slower(than shipped date).

Thse are the two main important perspective of creating the data warehouse.

## 2.1. Business Requirements

For fulfilling the First analysis I am using the 4 dimension tables, and my analysis will be updated in the fact table.

**Dimension Tables:**

1. **Product\_dim:** product Dimension table contains, the number of product that Northwind Company sells to customer.
2. **Suppliers\_Dim:** Suppliers Dimension table contains details of Vendor of Northwind Company.
3. **Calender\_Dim:** Calendar Dimension table, I am using for the details of Total revenue generating by suppliers over period of 1996-1998
4. **Orders\_Details\_Dim:** order details dimension table contains the details of Order transaction after purchasing the product. Like price, quantity and discount of product.
5. **Employees\_Dim:** Employee Dimension table contain the details of the employee who works for Northwind Company. details like their name, hire date, where they live, contact details etc.
6. **Shippers\_Dim:** Shippers dimension table contain the details of Northwind company shipper partner details. Like shipper name, contact details etc.
7. **Customer\_Dim:** Customer dimension table contains the details of Northwind company’s customer details like, name, contact details, where they live etc.

**Fact Table:**

1. **Revenue\_suppliers\_fact:** Fact table stores the information of Total revenue generated by supplier’s company. It linked with dimension tables like, product, order\_details and calendar so it is help us to analyse the other factors as well. Such as Product wise total revenue, yearly sales of product or yearly revenue of suppliers. So through this even we can analyse which product should keep for sales or removal from sales? Or which product is on the top trending, which help company to add new customers
2. **Performance fact:** Performance factstores the information about the performance of Shippers Company. Like how many product they delivered to customer late, before\_time or on\_time. Even through this fact table we can analyse the customer behaviour, since how long he is been attached with company, who is the top most buyers and etc.

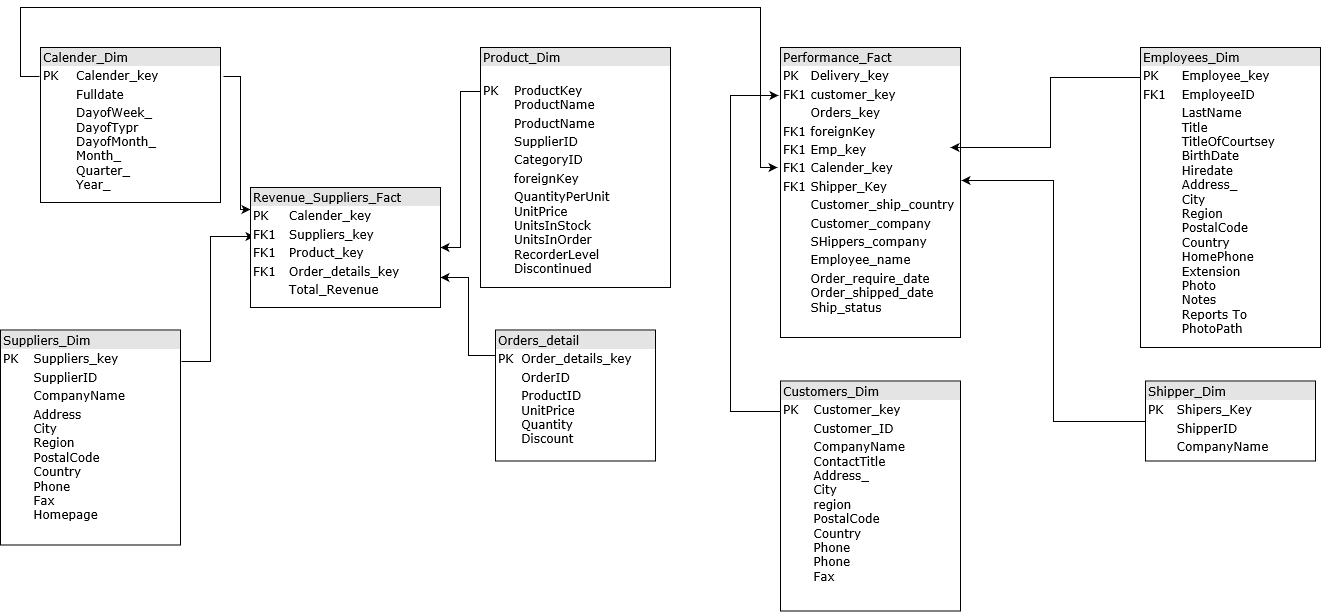
**Business Drivers and their corresponding objectives are shown in the below table**

| **Business Drivers** | **Business Objectives** |
| --- | --- |
| Addition of new customers | Manage the increase in volume |
| Addition/Removal of products | Total Revenue of product suppliers |
| Addition/Removal of shippers | Demand of product and deliveries |

# 3. Relational star schema

**Relational star schema of Northwind data warehouse as below:**

In star schema there 7 dimension tables are using for 2 fact tables. As shown in below diagram.



When we look at star schema its very simple relational model with one to many relationship of dimension tables and fact table. Although star schema is the relational model, it is not normalized form, wherein dimension tables are purposely normalized this one is basic difference between star schema and relational schema for OLTP systems.

From the above diagram, in 1st Revenue suppliers fact table, three foreign key are used i.e. Calender\_key, suppliers\_key, Product\_key which are the primary key of dimension tables, calender\_Dim, Suppliers\_Dim, Product\_Dim. And in column total\_revenue of fact table, total revenue of suppliers are stored. Which helps to business for analyzing the behavior of suppliers.

**Reasons for the design –**

1. All dimension tables are referenced with fact table. Join query can be easily implemented.
2. Changing anomalies can be controlled easily because there are hardly no modifications in the classification data
3. Query (Aggregation) optimize faster, as already dimension tables linked with fact tables, query fetches the faster than other OLTP system.
4. Star schema has simpler query
5. Simple business reporting logic easy to understand even fresher can easily understand

# 4. microsoft SQL server and ETL implementations

## 4.1 Source table creating query **(before creating dimension tables)**

Source table crating table query as follows by table to table

1. **Employee Source table:**

CREATE TABLE "Employees" (

"EmployeeID" "int" IDENTITY (1, 1) NOT NULL ,

"LastName" nvarchar (20) NOT NULL ,

"FirstName" nvarchar (10) NOT NULL ,

"Title" nvarchar (30) NULL ,

"TitleOfCourtesy" nvarchar (25) NULL ,

"BirthDate" "datetime" NULL ,

"HireDate" "datetime" NULL ,

"Address" nvarchar (60) NULL ,

"City" nvarchar (15) NULL ,

"Region" nvarchar (15) NULL ,

"PostalCode" nvarchar (10) NULL ,

"Country" nvarchar (15) NULL ,

"HomePhone" nvarchar (24) NULL ,

"Extension" nvarchar (4) NULL ,

"Photo" "image" NULL ,

"Notes" "ntext" NULL ,

"ReportsTo" "int" NULL ,

"PhotoPath" nvarchar (255) NULL ,

CONSTRAINT "PK\_Employees" PRIMARY KEY CLUSTERED

(

"EmployeeID"

),

CONSTRAINT "FK\_Employees\_Employees" FOREIGN KEY

(

"ReportsTo"

) REFERENCES "dbo"."Employees" (

"EmployeeID"

),

CONSTRAINT "CK\_Birthdate" CHECK (BirthDate < getdate())

)

GO

CREATE INDEX "LastName" ON "dbo"."Employees"("LastName")

GO

CREATE INDEX "PostalCode" ON "dbo"."Employees"("PostalCode")

GO

CREATE TABLE "Categories" (

"CategoryID" "int" IDENTITY (1, 1) NOT NULL ,

"CategoryName" nvarchar (15) NOT NULL ,

"Description" "ntext" NULL ,

"Picture" "image" NULL ,

CONSTRAINT "PK\_Categories" PRIMARY KEY CLUSTERED

(

"CategoryID"

)

)

GO

CREATE INDEX "CategoryName" ON "dbo"."Categories"("CategoryName")

GO

1. **Customers Source table:**

CREATE TABLE "Customers" (

"CustomerID" nchar (5) NOT NULL ,

"CompanyName" nvarchar (40) NOT NULL ,

"ContactName" nvarchar (30) NULL ,

"ContactTitle" nvarchar (30) NULL ,

"Address" nvarchar (60) NULL ,

"City" nvarchar (15) NULL ,

"Region" nvarchar (15) NULL ,

"PostalCode" nvarchar (10) NULL ,

"Country" nvarchar (15) NULL ,

"Phone" nvarchar (24) NULL ,

"Fax" nvarchar (24) NULL ,

CONSTRAINT "PK\_Customers" PRIMARY KEY CLUSTERED

(

"CustomerID"

)

)

GO

CREATE INDEX "City" ON "dbo"."Customers"("City")

GO

CREATE INDEX "CompanyName" ON "dbo"."Customers"("CompanyName")

GO

CREATE INDEX "PostalCode" ON "dbo"."Customers"("PostalCode")

GO

CREATE INDEX "Region" ON "dbo"."Customers"("Region")

GO

CREATE TABLE "Shippers" (

"ShipperID" "int" IDENTITY (1, 1) NOT NULL ,

"CompanyName" nvarchar (40) NOT NULL ,

"Phone" nvarchar (24) NULL ,

CONSTRAINT "PK\_Shippers" PRIMARY KEY CLUSTERED

(

"ShipperID"

)

)

1. **Supplier Source table:**

GO

CREATE TABLE "Suppliers" (

"SupplierID" "int" IDENTITY (1, 1) NOT NULL ,

"CompanyName" nvarchar (40) NOT NULL ,

"ContactName" nvarchar (30) NULL ,

"ContactTitle" nvarchar (30) NULL ,

"Address" nvarchar (60) NULL ,

"City" nvarchar (15) NULL ,

"Region" nvarchar (15) NULL ,

"PostalCode" nvarchar (10) NULL ,

"Country" nvarchar (15) NULL ,

"Phone" nvarchar (24) NULL ,

"Fax" nvarchar (24) NULL ,

"HomePage" "ntext" NULL ,

CONSTRAINT "PK\_Suppliers" PRIMARY KEY CLUSTERED

(

"SupplierID"

)

)

GO

CREATE INDEX "CompanyName" ON "dbo"."Suppliers"("CompanyName")

GO

CREATE INDEX "PostalCode" ON "dbo"."Suppliers"("PostalCode")

GO

1. **Orders Source table:**

CREATE TABLE "Orders" (

"OrderID" "int" IDENTITY (1, 1) NOT NULL ,

"CustomerID" nchar (5) NULL ,

"EmployeeID" "int" NULL ,

"OrderDate" "datetime" NULL ,

"RequiredDate" "datetime" NULL ,

"ShippedDate" "datetime" NULL ,

"ShipVia" "int" NULL ,

"Freight" "money" NULL CONSTRAINT "DF\_Orders\_Freight" DEFAULT (0),

"ShipName" nvarchar (40) NULL ,

"ShipAddress" nvarchar (60) NULL ,

"ShipCity" nvarchar (15) NULL ,

"ShipRegion" nvarchar (15) NULL ,

"ShipPostalCode" nvarchar (10) NULL ,

"ShipCountry" nvarchar (15) NULL ,

CONSTRAINT "PK\_Orders" PRIMARY KEY CLUSTERED

(

"OrderID"

),

CONSTRAINT "FK\_Orders\_Customers" FOREIGN KEY

(

"CustomerID"

) REFERENCES "dbo"."Customers" (

"CustomerID"

),

CONSTRAINT "FK\_Orders\_Employees" FOREIGN KEY

(

"EmployeeID"

) REFERENCES "dbo"."Employees" (

"EmployeeID"

),

CONSTRAINT "FK\_Orders\_Shippers" FOREIGN KEY

(

"ShipVia"

) REFERENCES "dbo"."Shippers" (

"ShipperID"

)

)

GO

CREATE INDEX "CustomerID" ON "dbo"."Orders"("CustomerID")

GO

CREATE INDEX "CustomersOrders" ON "dbo"."Orders"("CustomerID")

GO

CREATE INDEX "EmployeeID" ON "dbo"."Orders"("EmployeeID")

GO

CREATE INDEX "EmployeesOrders" ON "dbo"."Orders"("EmployeeID")

GO

CREATE INDEX "OrderDate" ON "dbo"."Orders"("OrderDate")

GO

CREATE INDEX "ShippedDate" ON "dbo"."Orders"("ShippedDate")

GO

CREATE INDEX "ShippersOrders" ON "dbo"."Orders"("ShipVia")

GO

CREATE INDEX "ShipPostalCode" ON "dbo"."Orders"("ShipPostalCode")

GO

1. **Products Source table**

CREATE TABLE "Products" (

"ProductID" "int" IDENTITY (1, 1) NOT NULL ,

"ProductName" nvarchar (40) NOT NULL ,

"SupplierID" "int" NULL ,

"CategoryID" "int" NULL ,

"QuantityPerUnit" nvarchar (20) NULL ,

"UnitPrice" "money" NULL CONSTRAINT "DF\_Products\_UnitPrice" DEFAULT (0),

"UnitsInStock" "smallint" NULL CONSTRAINT "DF\_Products\_UnitsInStock" DEFAULT (0),

"UnitsOnOrder" "smallint" NULL CONSTRAINT "DF\_Products\_UnitsOnOrder" DEFAULT (0),

"ReorderLevel" "smallint" NULL CONSTRAINT "DF\_Products\_ReorderLevel" DEFAULT (0),

"Discontinued" "bit" NOT NULL CONSTRAINT "DF\_Products\_Discontinued" DEFAULT (0),

CONSTRAINT "PK\_Products" PRIMARY KEY CLUSTERED

(

"ProductID"

),

CONSTRAINT "FK\_Products\_Categories" FOREIGN KEY

(

"CategoryID"

) REFERENCES "dbo"."Categories" (

"CategoryID"

),

CONSTRAINT "FK\_Products\_Suppliers" FOREIGN KEY

(

"SupplierID"

) REFERENCES "dbo"."Suppliers" (

"SupplierID"

),

CONSTRAINT "CK\_Products\_UnitPrice" CHECK (UnitPrice >= 0),

CONSTRAINT "CK\_ReorderLevel" CHECK (ReorderLevel >= 0),

CONSTRAINT "CK\_UnitsInStock" CHECK (UnitsInStock >= 0),

CONSTRAINT "CK\_UnitsOnOrder" CHECK (UnitsOnOrder >= 0)

)

GO

CREATE INDEX "CategoriesProducts" ON "dbo"."Products"("CategoryID")

GO

CREATE INDEX "CategoryID" ON "dbo"."Products"("CategoryID")

GO

CREATE INDEX "ProductName" ON "dbo"."Products"("ProductName")

GO

CREATE INDEX "SupplierID" ON "dbo"."Products"("SupplierID")

GO

CREATE INDEX "SuppliersProducts" ON "dbo"."Products"("SupplierID")

GO

1. **Order Details Source table:**

CREATE TABLE "Order Details" (

"OrderID" "int" NOT NULL ,

"ProductID" "int" NOT NULL ,

"UnitPrice" "money" NOT NULL CONSTRAINT "DF\_Order\_Details\_UnitPrice" DEFAULT (0),

"Quantity" "smallint" NOT NULL CONSTRAINT "DF\_Order\_Details\_Quantity" DEFAULT (1),

"Discount" "real" NOT NULL CONSTRAINT "DF\_Order\_Details\_Discount" DEFAULT (0),

CONSTRAINT "PK\_Order\_Details" PRIMARY KEY CLUSTERED

(

"OrderID",

"ProductID"

),

CONSTRAINT "FK\_Order\_Details\_Orders" FOREIGN KEY

(

"OrderID"

) REFERENCES "dbo"."Orders" (

"OrderID"

),

CONSTRAINT "FK\_Order\_Details\_Products" FOREIGN KEY

(

"ProductID"

) REFERENCES "dbo"."Products" (

"ProductID"

),

CONSTRAINT "CK\_Discount" CHECK (Discount >= 0 and (Discount <= 1)),

CONSTRAINT "CK\_Quantity" CHECK (Quantity > 0),

CONSTRAINT "CK\_UnitPrice" CHECK (UnitPrice >= 0)

)

## 4.2 Dimension and fact table creating queries

**1. Revenue suppliers fact table:**

use northwind\_DW

GO

create table Revenue\_Suppliers\_fact

(

Calender\_Key int,

suppliers\_key int,

product\_key int,

Order\_key int,

Total\_Revenue int,

CONSTRAINT FK\_product\_key FOREIGN KEY (product\_key)

REFERENCES [dbo].[Products\_Dim](product\_key),

CONSTRAINT FK\_suppliers\_key FOREIGN KEY (suppliers\_key)

REFERENCES [dbo].[Suppliers\_dim](suppliers\_key),

CONSTRAINT FK\_Order\_key FOREIGN KEY (Order\_key)

REFERENCES [dbo].[Orders](OrderID),

CONSTRAINT FK\_calender\_key FOREIGN KEY (Calender\_Key)

REFERENCES[dbo].[Calendar\_Dim](CalendarKey)

)

1. **Performance fact table:**

create table Performance\_fact

(

delivery\_key int not null identity primary key,

customer\_key int null,

orders\_key int,

Emp\_key int null,

calender\_key int null,

shipper\_key int null,

customer\_ship\_country nvarchar(25) null,

Customer\_company nvarchar(25) null,

shippers\_company nvarchar(25) null,

Employee\_name nvarchar(25) null,

order\_require\_date datetime NULL,

order\_shipped\_date datetime null,

ship\_status nvarchar(25) null,

CONSTRAINT FK\_customer\_key FOREIGN KEY (customer\_key)

REFERENCES [dbo].[Customers\_Dim](customer\_key),

CONSTRAINT FK\_Emp\_key FOREIGN KEY (Emp\_key)

REFERENCES [dbo].[Employees\_Dim](Employee\_key),

CONSTRAINT FK\_shipper\_key FOREIGN KEY (shipper\_key)

REFERENCES [dbo].[Shippers\_Dim](shippers\_key),

FOREIGN KEY (calender\_key) REFERENCES [dbo].[Calendar\_Dim](CalendarKey)

)

UPDATE [dbo].[Performance\_fact]

SET [Performance\_fact].[ship\_status] =

CASE

WHEN ([order\_require\_date] = [order\_shipped\_date])

THEN 'On\_time'

WHEN ([order\_require\_date] < [order\_shipped\_date])

THEN 'Before\_Time'

WHEN ([order\_require\_date] > [order\_shipped\_date])

THEN 'Late'

END

1. **Shippers Dimension table:**

CREATE TABLE Shippers\_Dim (

shippers\_key int identity not null primary key,

ShipperID int NULL ,

CompanyName nvarchar (40) NOT NULL ,

Phone nvarchar (24) NULL

)

1. **Order details Dimension table:**

CREATE TABLE Order\_Details\_Dim (

Order\_Details\_key int Identity Not Null Primary Key,

OrderID int NOT NULL,

ProductID int NOT NULL ,

UnitPrice money NOT NULL,

Quantity smallint not null,

Discount real NOT NULL)

1. **Suppliers Dimension table:**

/\*Suppliers table creating query\*/

CREATE TABLE Suppliers\_dim (

suppliers\_key int identity not null primary key,

SupplierID int NOT NULL ,

CompanyName nvarchar (40) NOT NULL ,

ContactName nvarchar (30) NULL ,

ContactTitle nvarchar (30) NULL ,

Address nvarchar (60) NULL ,

City nvarchar (15) NULL ,

Region nvarchar (15) NULL ,

PostalCode nvarchar (10) NULL ,

Country nvarchar (15) NULL ,

Phone nvarchar (24) NULL ,

Fax nvarchar (24) NULL ,

HomePage ntext NULL)

1. **Products Dimension table**

CREATE TABLE Products\_Dim (

product\_key int identity not null primary key,

ProductID int NOT NULL ,

ProductName nvarchar (40) NOT NULL,

SupplierID int NULL ,

CategoryID int NULL ,

QuantityPerUnit nvarchar (20) NULL ,

UnitPrice money NULL,

UnitsInStock smallint NULL,

UnitsOnOrder smallint NULL,

ReorderLevel smallint NULL,

Discontinued bit NOT NULL)

1. **Calendar Dimension table:**

CREATE TABLE Calendar\_Dim

(

CalendarKey INT NOT NULL IDENTITY,

FullDate DATETIME,

DayofWeek\_ CHAR(15),

DayType CHAR(20),

DayofMonth\_ INT,

Month\_ CHAR(10),

Quarter\_ CHAR(2),

Year\_ INT,

PRIMARY KEY (CalendarKey));

1. **Customers Dimension table:**

CREATE TABLE Customers\_Dim (

customer\_key int not null identity primary key,

CustomerID nchar (5) NOT NULL ,

CompanyName nvarchar (40) NOT NULL ,

ContactName nvarchar (30) NULL ,

ContactTitle nvarchar (30) NULL ,

Address\_ nvarchar (60) NULL ,

City nvarchar (15) NULL ,

Region nvarchar (15) NULL ,

PostalCode nvarchar (10) NULL ,

Country nvarchar (15) NULL ,

Phone nvarchar (24) NULL ,

Fax nvarchar (24) NULL)

1. **Employees Dimension table:**

CREATE TABLE Employees\_Dim (

Employee\_key int Not Null Identity Primary Key,

EmployeeID int NOT NULL ,

LastName nvarchar (20) NOT NULL ,

FirstName nvarchar (10) NOT NULL ,

Title nvarchar (30) NULL ,

TitleOfCourtesy nvarchar (25) NULL ,

BirthDate datetime NULL ,

HireDate datetime NULL ,

Address nvarchar (60) NULL ,

City nvarchar (15) NULL ,

Region nvarchar (15) NULL ,

PostalCode nvarchar (10) NULL ,

Country nvarchar (15) NULL ,

HomePhone nvarchar (24) NULL ,

Extension nvarchar (4) NULL ,

Photo image NULL ,

Notes ntext NULL ,

ReportsTo int NULL ,

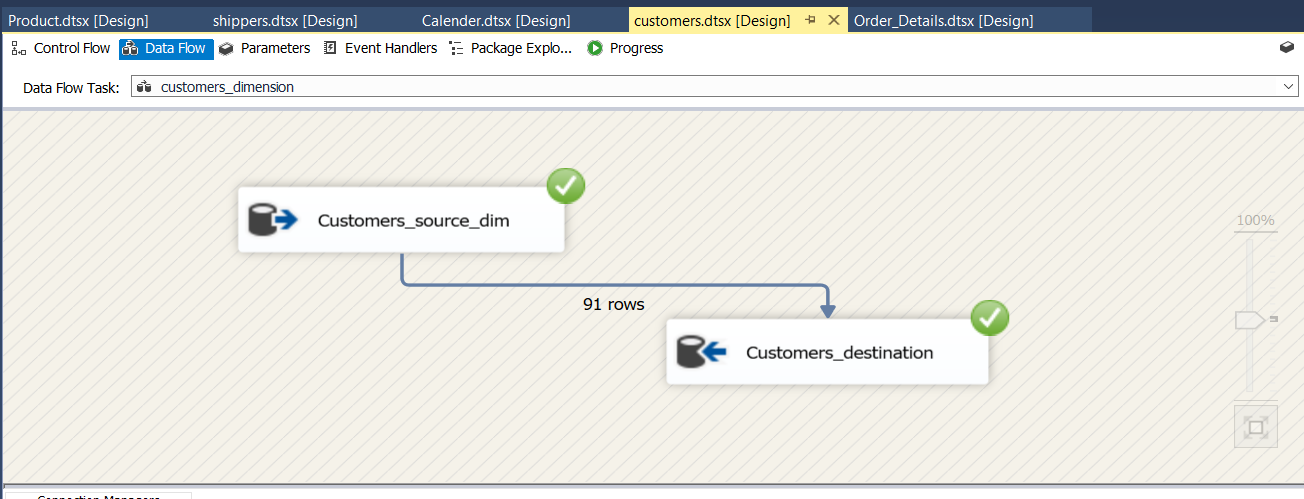
PhotoPath nvarchar (255) NULL)

## 4.3 Implementation of etl

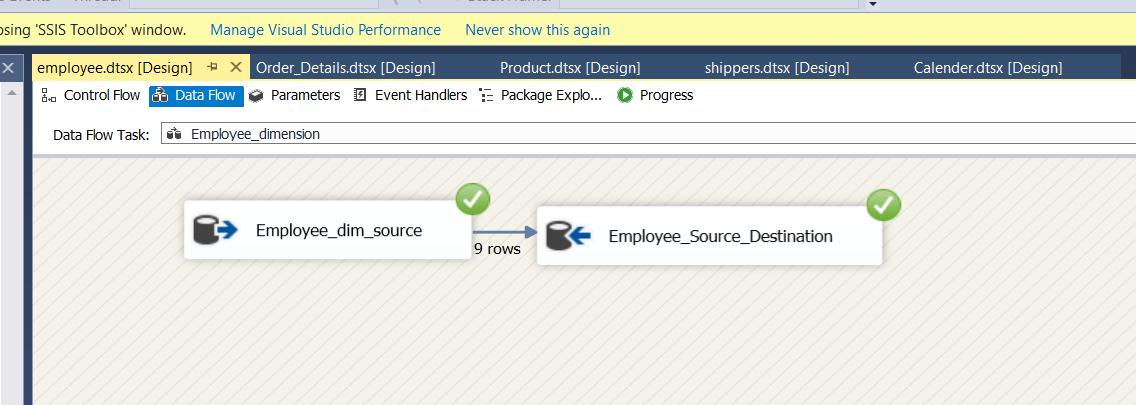
In the staging of implementation of ETL process we using SSDT tool for implementation of ETL.

In which we are extracting the from source table. Transforming it and loading to dimension table and fact. The step by step ETL execution of every dimension table and fact table’s screen are as follows.

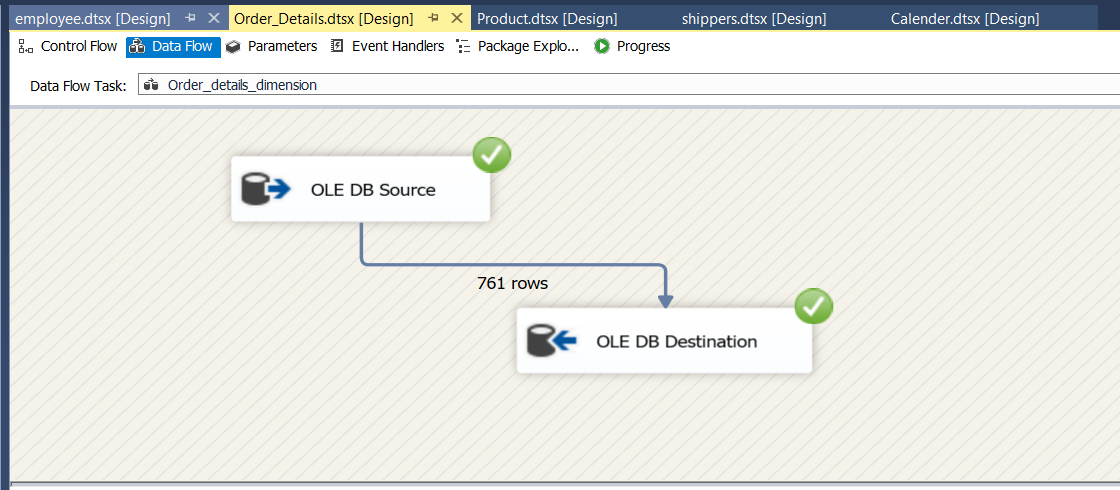
1. **Customer Dim:**

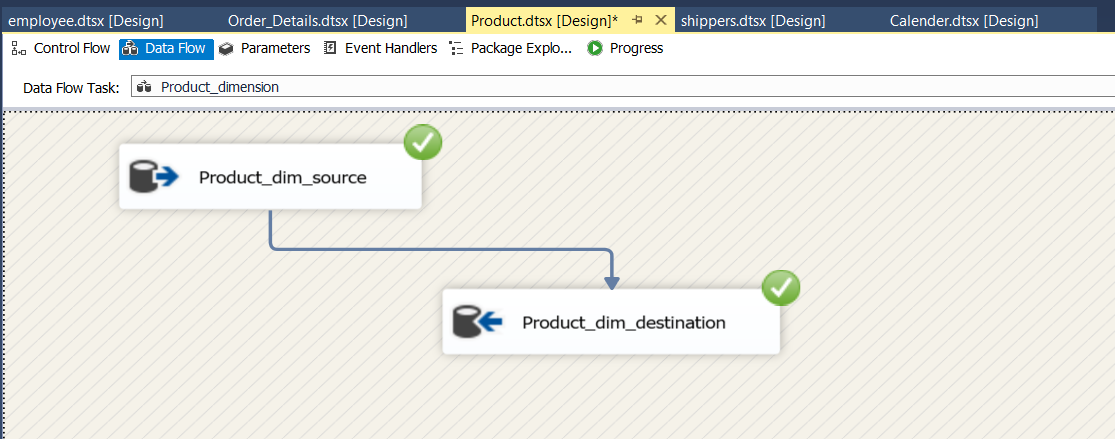
****

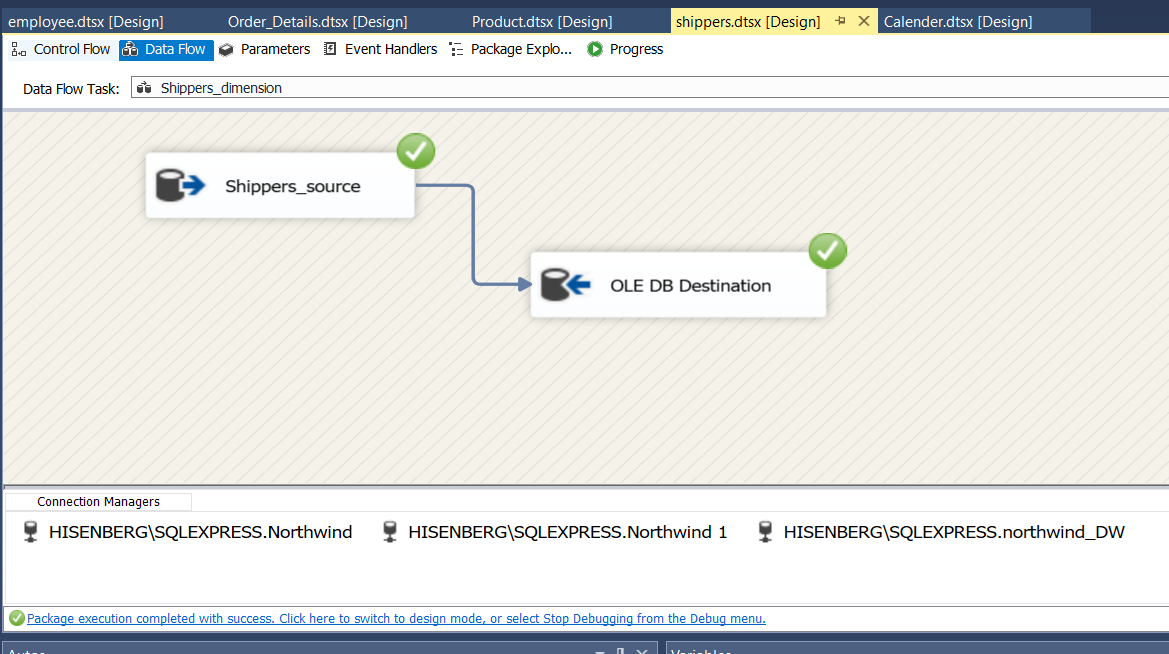
1. **Employee Dim:**

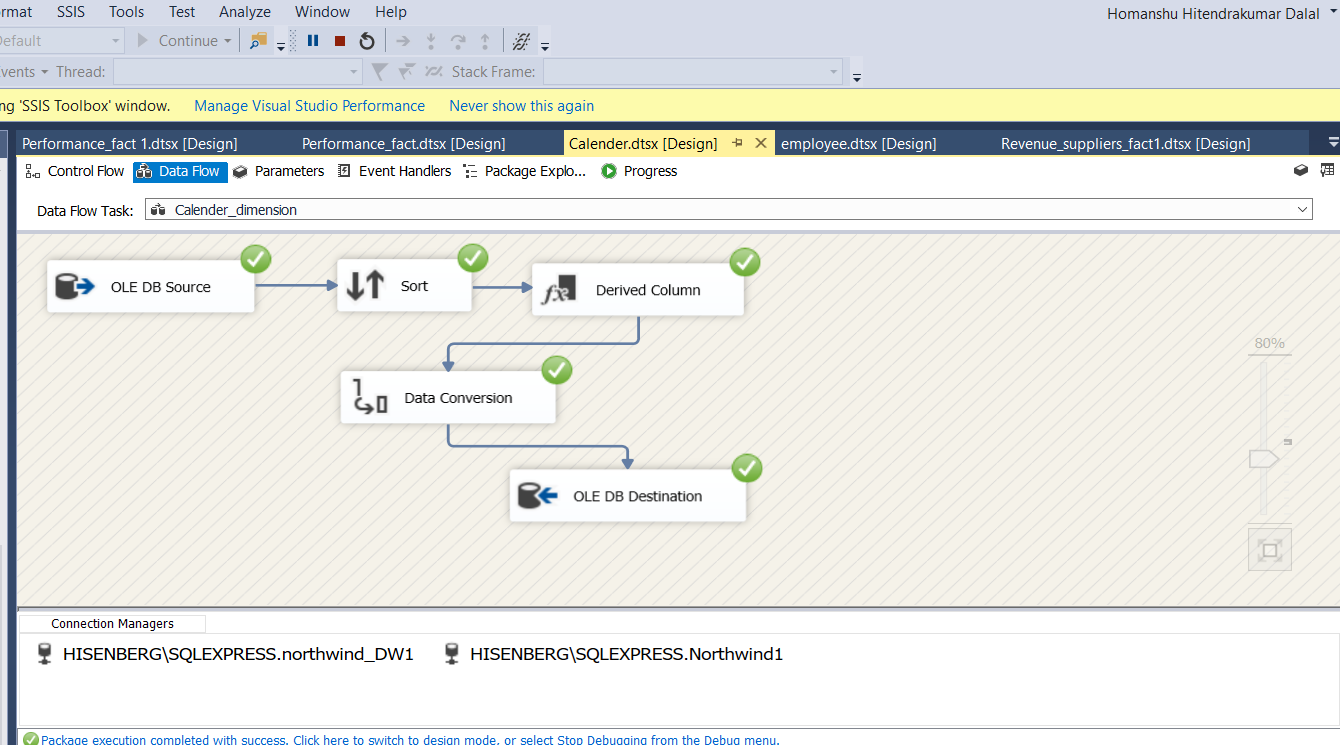
****

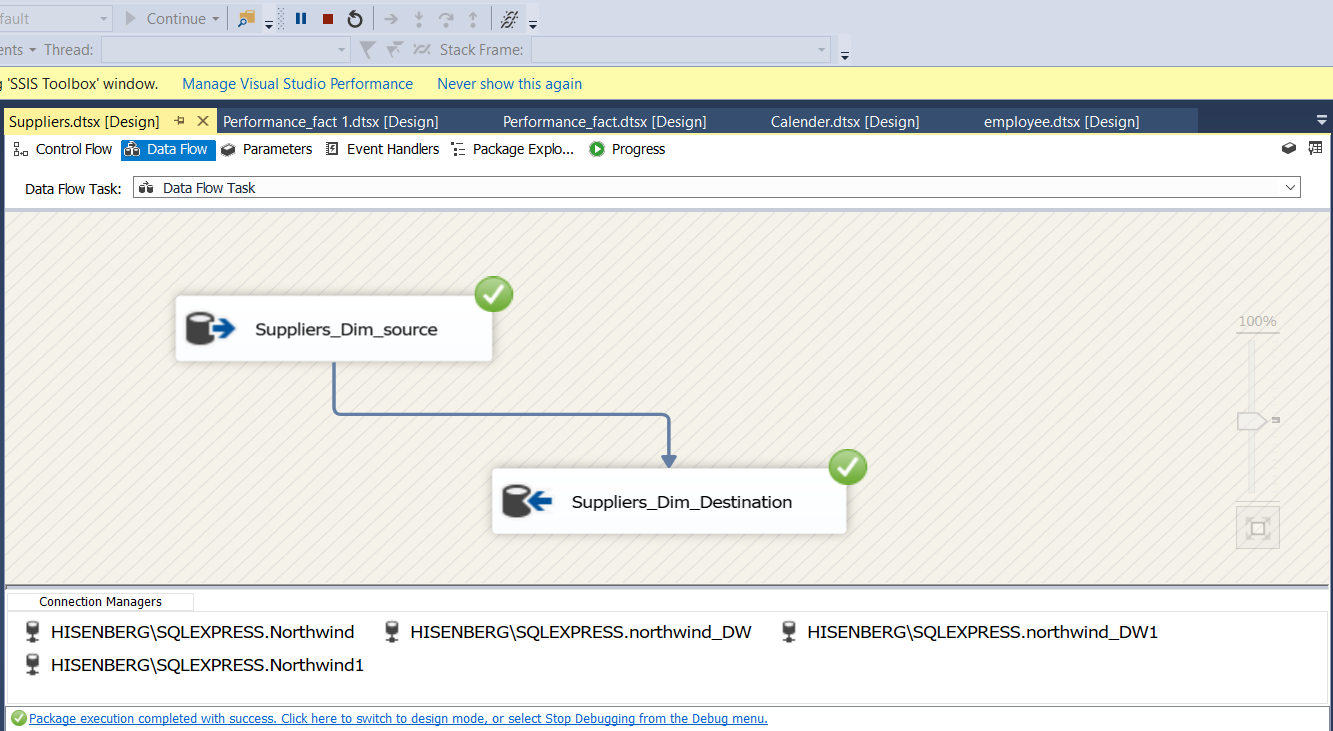
1. **Order Details Dim:**

****

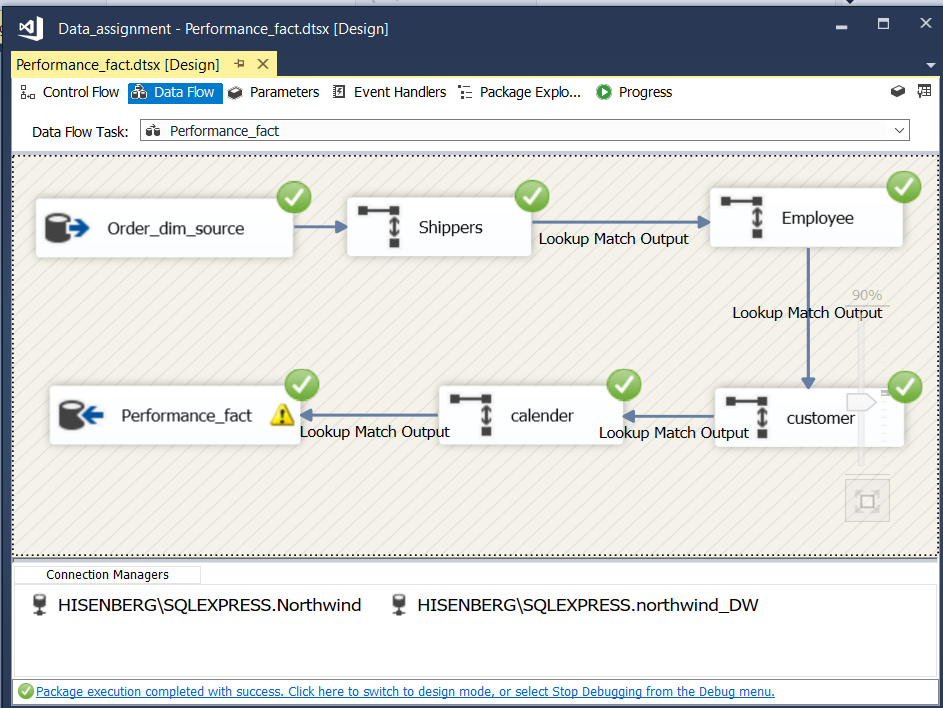
1. **Product\_Dim: **
2. **Shippers Dim:**

****

1. **Calendar\_Dim: **
2. **Supplier Dim:**

****

1. **Performance Revenue Fact:**



**UPDATE [dbo].[Performance\_fact]**

**SET [Performance\_fact].[ship\_status] =**

**CASE**

**WHEN ([order\_require\_date] = [order\_shipped\_date])**

**THEN 'On\_time'**

**WHEN ([order\_require\_date] < [order\_shipped\_date])**

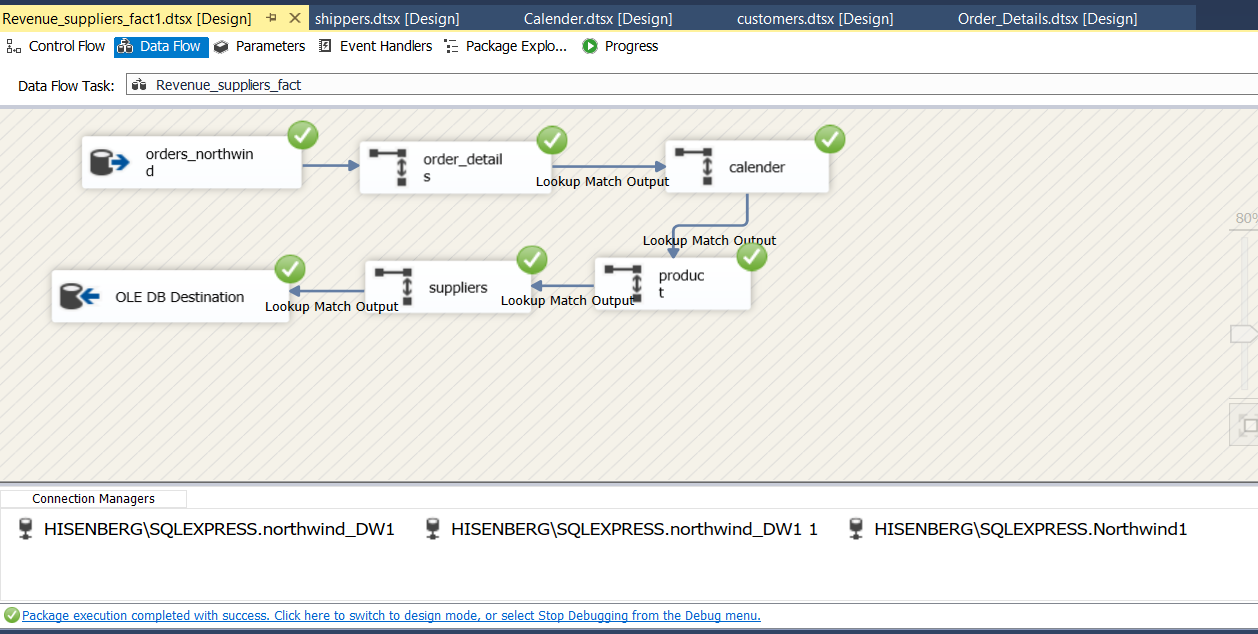
**THEN 'Before\_Time'**

**WHEN ([order\_require\_date] > [order\_shipped\_date])**

**THEN 'Late'**

**END**

1. **Supplier Revenue fact:**

****

# 5. Reports and visualization

In the given project SSRS is using for generating report of warehouse and R studio is using for visualization. The each report and their visualization in R of screen shot as follows.

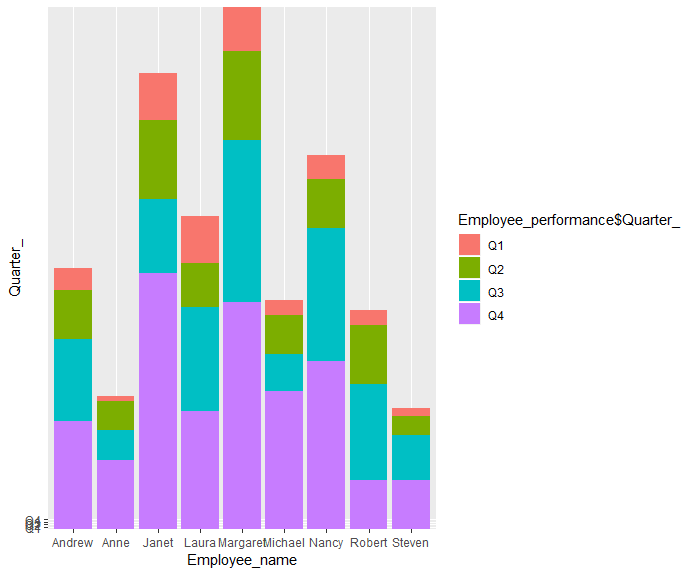
1. **Employee Performance of year 1997:**

Below report showing performance of employee which works for Northwind Company.

The below report telling us the how many number of order shipment handled by the employee. Wherein Margaret is best performer year of 1997 employee of company.



**R visualization of employee performance report as below:**

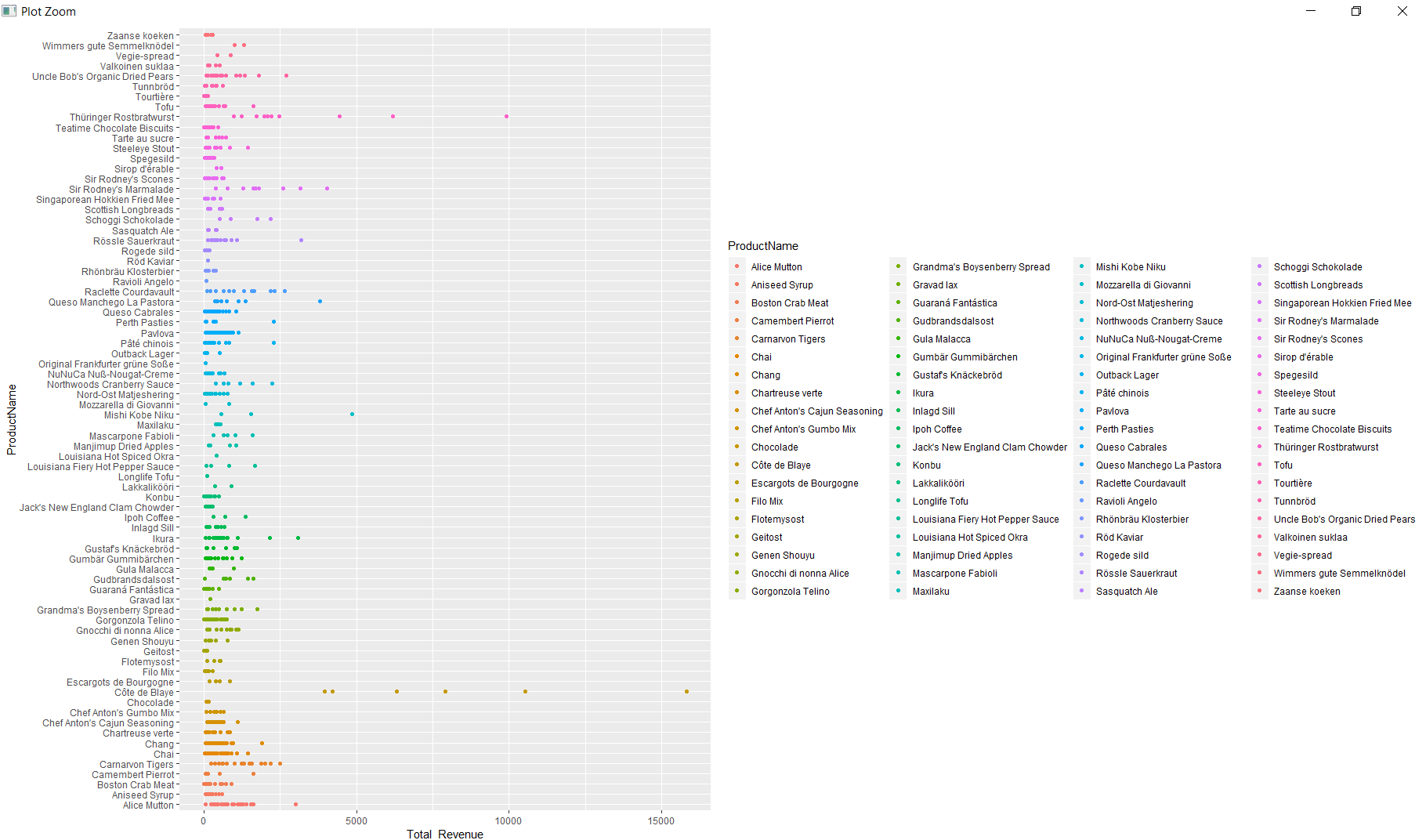
****

1. **Product wise Revenue:**

Below report showing the Revenue of different product for period 1996-1998. And corresponding to their company name, **“Cote de Blaye”** is the highest revenue generator product of Northwind Company.



**The visualization of above report as below:**

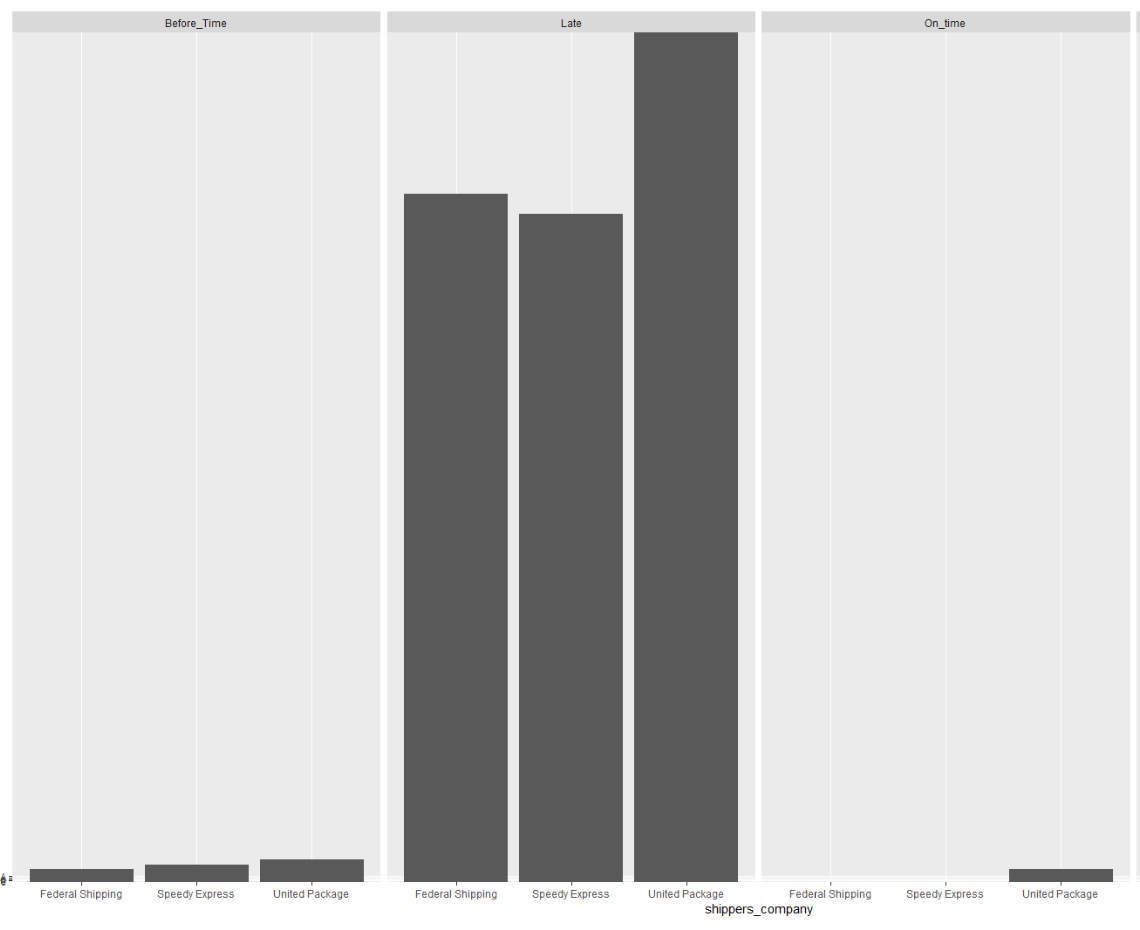
****

1. **Shipper company delivery status(Performance):**

The below report showing which company has delivered the product on on\_time, before\_time and Late. i.e. their work efficiency.



**The visualization of above report as below:**



1. **Suppliers Revenue:**

The below report showing the revenue of suppliers for period 1996-1998. Wherein **“Aux joyeux ecclesiastiques”** is at the top for revenue generator.



**The visualization of report is as below:**

In Visualization we are interested to see top 10 suppliers of year 1997



# 6. XML and XML schema

An XML schema is the structural layout of an XML document, expressed in terms of constraints and contents of the document. Constraints are expressed using a combination of the following:

* Grammatical rules governing the order of elements
* Data types governing an element and content attribute
* Boolean predicates that the content has to satisfy
* Specialized rules including uniqueness and referential integrity constraints

## 6.1 XML code of total revenue fact

**XML of total revenue fact table:**

**<?xml version="1.0"?>**

**<Total\_Revenue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**

**xsi:noNamespaceSchemaLocation="Total\_Revenue.xsd">**

**<Calender\_Dim>**

**<Calender\_key>1</Calender\_key>**

**<FullDate>1996-07-04T00:00:00.000</FullDate>**

**<DayofWeek\_>Weekday </DayofWeek\_>**

**<DayType>Thursday </DayType>**

**<DayofMonth\_>4</DayofMonth\_>**

**<Month\_>7</Month\_>**

**<Quarter\_>Q3</Quarter\_>**

**<Year\_>1996</Year\_>**

**</Calender\_Dim>**

**<Calender\_Dim>**

**<Calender\_key>2</Calender\_key>**

**<FullDate>1996-07-04T00:00:00.000</FullDate>**

**<DayofWeek\_>Weekday </DayofWeek\_>**

**<DayType>Friday</DayType>**

**<DayofMonth\_>5</DayofMonth\_>**

**<Month\_>7</Month\_>**

**<Quarter\_>Q3</Quarter\_>**

**<Year\_>1996</Year\_>**

**</Calender\_Dim>**

**<Suppliers\_Dim>**

**<suppliers\_key>5</suppliers\_key>**

**<SupplierID>5</SupplierID>**

**<CompanyName>Cooperativa de Quesos 'Las Cabras'</CompanyName>**

**<ContactName>Antonio del Valle Saavedra</ContactName>**

**<ContactTitle>Export Administrator</ContactTitle>**

**<Address>Calle del Rosal 4</Address>**

**<City>Oviedo</City>**

**<Region>Asturias</Region>**

**<PostalCode>33007</PostalCode>**

**<Country>Spain</Country>**

**<Phone>985987654</Phone>**

**<Fax>NULL</Fax>**

**<HomePage>NULL</HomePage>**

**</Suppliers\_Dim>**

**<Suppliers\_Dim>**

**<suppliers\_key>6</suppliers\_key>**

**<SupplierID>5</SupplierID>**

**<CompanyName>Mayumi's</CompanyName>**

**<ContactName>Mayumi Ohno</ContactName>**

**<ContactTitle>Marketing Representative</ContactTitle>**

**<Address>92 Setsuko Chuo-ku</Address>**

**<City>Osaka</City>**

**<Region>NULL</Region>**

**<PostalCode>545</PostalCode>**

**<Country>Japan</Country>**

**<Phone>064317877</Phone>**

**<Fax>NULL</Fax>**

**<HomePage>Mayumi's (on the World Wide Web)#http://www.microsoft.com/accessdev/sampleapps/mayumi.htm#</HomePage>**

**</Suppliers\_Dim>**

**<Products\_Dim>**

**<Product\_key>11</Product\_key>**

**<ProductID>11</ProductID>**

**<ProductName>Queso Cabrales</ProductName>**

**<SupplierID>5</SupplierID>**

**<CategoryID>4</CategoryID>**

**<QuantityPerUnit>1 kg pkg.</QuantityPerUnit>**

**<UnitPrice>21.00</UnitPrice>**

**<UnitsInStock>22</UnitsInStock>**

**<UnitsOnOrder>30</UnitsOnOrder>**

**<ReorderLevel>30</ReorderLevel>**

**<Discontinued>0</Discontinued>**

**</Products\_Dim>**

**<Products\_Dim>**

**<Product\_key>14</Product\_key>**

**<ProductID>11</ProductID>**

**<ProductName>Tofu</ProductName>**

**<SupplierID>6</SupplierID>**

**<CategoryID>7</CategoryID>**

**<QuantityPerUnit>40 - 100 g pkgs.</QuantityPerUnit>**

**<UnitPrice>23.25</UnitPrice>**

**<UnitsInStock>35</UnitsInStock>**

**<UnitsOnOrder>0</UnitsOnOrder>**

**<ReorderLevel>0</ReorderLevel>**

**<Discontinued>0</Discontinued>**

**</Products\_Dim>**

**<Order\_Details\_Dim>**

**<Order\_Details\_key>1</Order\_Details\_key>**

**<OrderID>10248</OrderID>**

**<ProductID>11</ProductID>**

**<UnitPrice>14.00</UnitPrice>**

**<Quantity>12</Quantity>**

**<Discount>0</Discount>**

**</Order\_Details\_Dim>**

**<Order\_Details\_Dim>**

**<Order\_Details\_key>4</Order\_Details\_key>**

**<OrderID>10249</OrderID>**

**<ProductID>14</ProductID>**

**<UnitPrice>18.60</UnitPrice>**

**<Quantity>9</Quantity>**

**<Discount>0</Discount>**

**</Order\_Details\_Dim>**

**<Revenue\_Suppliers\_fact>**

**<Calender\_key>1</Calender\_key>**

**<suppliers\_key>5</suppliers\_key>**

**<Product\_key>11</Product\_key>**

**<Order\_Details\_key>1</Order\_Details\_key>**

**<Total\_Revenue>168</Total\_Revenue>**

**</Revenue\_Suppliers\_fact>**

**<Revenue\_Suppliers\_fact>**

**<Calender\_key>2</Calender\_key>**

**<suppliers\_key>6</suppliers\_key>**

**<Product\_key>14</Product\_key>**

**<Order\_Details\_key>4</Order\_Details\_key>**

**<Total\_Revenue>167</Total\_Revenue>**

**</Revenue\_Suppliers\_fact>**

**</Total\_Revenue>**

## 6.2 XML schema of total revenue fact

<?xml version="1.0"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="Total\_Revenue">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Calender\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Suppliers\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Products\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Order\_Details\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Revenue\_Suppliers\_fact" minOccurs="1" maxOccurs="unbounded" />

</xsd:sequence>

</xsd:complexType>

<xsd:key name="Calender\_key">

<xsd:selector xpath="Calender\_Dim/Calender\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Calender\_key\_ref" refer="Calender\_key">

<xsd:selector xpath="Revenue\_Suppliers\_fact/Calender\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="suppliers\_key">

<xsd:selector xpath="Suppliers\_Dim/suppliers\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="suppliers\_key\_ref" refer="suppliers\_key">

<xsd:selector xpath="Revenue\_Suppliers\_fact/suppliers\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="Product\_key">

<xsd:selector xpath="Products\_Dim/Product\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Product\_key\_ref" refer="Product\_key">

<xsd:selector xpath="Revenue\_Suppliers\_fact/Product\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="Order\_Details\_key">

<xsd:selector xpath="Order\_Details\_Dim/Order\_Details\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Order\_Details\_ref" refer="Order\_Details\_key">

<xsd:selector xpath="Revenue\_Suppliers\_fact/Order\_Details\_key" />

<xsd:field xpath="." />

</xsd:keyref>

</xsd:element>

<xsd:element name="Calender\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Calender\_key" type="xsd:int" />

<xsd:element name="FullDate" type="xsd:dateTime" />

<xsd:element name="DayofWeek\_" type="xsd:string" />

<xsd:element name="DayType" type="xsd:string" />

<xsd:element name="DayofMonth\_" type="xsd:int" />

<xsd:element name="Month\_" type="xsd:int" />

<xsd:element name="Quarter\_" type="xsd:string" />

<xsd:element name="Year\_" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Suppliers\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="suppliers\_key" type="xsd:int" />

<xsd:element name="SupplierID" type="xsd:int" />

<xsd:element name="CompanyName" type="xsd:string" />

<xsd:element name="ContactName" type="xsd:string" />

<xsd:element name="ContactTitle" type="xsd:string" />

<xsd:element name="Address" type="xsd:string" />

<xsd:element name="City" type="xsd:string" />

<xsd:element name="Region" type="xsd:string" />

<xsd:element name="PostalCode" type="xsd:int" />

<xsd:element name="Country" type="xsd:string" />

<xsd:element name="Phone" type="xsd:int" />

<xsd:element name="Fax" type="xsd:string" />

<xsd:element name="HomePage" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Products\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Product\_key" type="xsd:int" />

<xsd:element name="ProductID" type="xsd:int" />

<xsd:element name="ProductName" type="xsd:string" />

<xsd:element name="SupplierID" type="xsd:int" />

<xsd:element name="CategoryID" type="xsd:int" />

<xsd:element name="QuantityPerUnit" type="xsd:string" />

<xsd:element name="UnitPrice" type="xsd:float" />

<xsd:element name="UnitsInStock" type="xsd:int" />

<xsd:element name="UnitsOnOrder" type="xsd:int" />

<xsd:element name="ReorderLevel" type="xsd:int" />

<xsd:element name="Discontinued" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Order\_Details\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Order\_Details\_key" type="xsd:int" />

<xsd:element name="OrderID" type="xsd:int" />

<xsd:element name="ProductID" type="xsd:int" />

<xsd:element name="UnitPrice" type="xsd:float" />

<xsd:element name="Quantity" type="xsd:int" />

<xsd:element name="Discount" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Revenue\_Suppliers\_fact">

<xsd:complexType>

<xsd:all>

<xsd:element name="Calender\_key" type="xsd:int" />

<xsd:element name="suppliers\_key" type="xsd:int" />

<xsd:element name="Product\_key" type="xsd:int" />

<xsd:element name="Order\_Details\_key" type="xsd:int" />

<xsd:element name="Total\_Revenue" type="xsd:int" />

</xsd:all>

</xsd:complexType>

</xsd:element>

</xsd:schema>

## 6.3 XML of performance fact

<?xml version="1.0"?>

<performance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="performance.xsd">

<Customers\_Dim>

<customer\_key>85</customer\_key>

<CustomerID>VINET</CustomerID>

<CompanyName>Vins et alcools Chevalier </CompanyName>

<ContactName>Paul Henriot </ContactName>

<ContactTitle>Accounting Manager</ContactTitle>

<Address\_>59 rue de l'Abbaye</Address\_>

<City>Reims</City>

<Region>NULL</Region>

<PostalCode>51100</PostalCode>

<Country>France</Country>

<Phone>26471510</Phone>

<Fax>26471511</Fax>

</Customers\_Dim>

<Orders>

<OrderID>10248</OrderID>

<CustomerID>VINET</CustomerID>

<EmployeeID>5</EmployeeID>

<OrderDate>1996-07-04T00:00:00</OrderDate>

<RequiredDate>1996-08-01T00:00:00</RequiredDate>

<ShippedDate>1996-07-16T00:00:00</ShippedDate>

<ShipVia>3</ShipVia>

<Freight>32.38</Freight>

<ShipName>Vins et alcools Chevalier</ShipName>

<ShipAddress>59 rue de l'Abbaye</ShipAddress>

<ShipCity>Reims</ShipCity>

<ShipRegion>NULL</ShipRegion>

<ShipPostalCode>51100</ShipPostalCode>

<ShipCountry>France</ShipCountry>

</Orders>

<Employees\_Dim>

<Employee\_key>5</Employee\_key>

<EmployeeID>5</EmployeeID>

<LastName>Buchanan </LastName>

<FirstName>Steven</FirstName>

<Title>Sales Manager</Title>

<TitleOfCourtesy>Mr.</TitleOfCourtesy>

<BirthDate>1955-03-04T00:00:00</BirthDate>

<HireDate>1993-10-17T00:00:00</HireDate>

<Address\_>14 Garrett Hill</Address\_>

<City>London</City>

<Region>NULL</Region>

<PostalCode>SW18JR</PostalCode>

<Country>UK</Country>

<HomePhone>715554848</HomePhone>

<Extension>3453</Extension>

<Photo></Photo>

<Notes>Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree in 1976. Upon joining the company as a sales representative in 1992, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London. He was promoted to sales manager in March 1993. Mr. Buchanan has completed the courses "Successful Telemarketing" and "International Sales Management." He is fluent in French.</Notes>

<ReportsTo> 2</ReportsTo>

<PhotoPath>http://accweb/emmployees/buchanan.bmp</PhotoPath>

</Employees\_Dim>

<Calender\_Dim>

<Calender\_key>1</Calender\_key>

<FullDate>1996-07-04T00:00:00</FullDate>

<DayofWeek\_>Weekday </DayofWeek\_>

<DayType>Thursday </DayType>

<DayofMonth\_>4</DayofMonth\_>

<Month\_>7</Month\_>

<Quarter\_>Q3</Quarter\_>

<Year\_>1996</Year\_>

</Calender\_Dim>

<Shippers\_Dim>

<shippers\_key>3</shippers\_key>

<ShipperID>3</ShipperID>

<CompanyName>Federal Shipping</CompanyName>

</Shippers\_Dim>

<Performance\_fact>

<delivery\_key>1</delivery\_key>

<customer\_key>85</customer\_key>

<orders\_key>10248</orders\_key>

<Emp\_key>5</Emp\_key>

<calender\_key>1</calender\_key>

<shipper\_key>3</shipper\_key>

<customer\_ship\_country>Reims</customer\_ship\_country>

<Customer\_company>Vins et alcools Chevalier</Customer\_company>

<shippers\_company>Federal Shipping</shippers\_company>

<Employee\_name>Steven</Employee\_name>

<order\_require\_date>1996-08-01T00:00:00</order\_require\_date>

<order\_shipped\_date>1996-07-16T00:00:00</order\_shipped\_date>

<ship\_status>Late</ship\_status>

</Performance\_fact>

</performance>

## 6.4 xml schema of performance fact

<?xml version="1.0"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="performance">

<xsd:complexType>

<xsd:sequence>

<xsd:element ref="Customers\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Orders" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Employees\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Calender\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Shippers\_Dim" minOccurs="1" maxOccurs="unbounded" />

<xsd:element ref="Performance\_fact" minOccurs="1" maxOccurs="unbounded" />

</xsd:sequence>

</xsd:complexType>

<xsd:key name="customer\_key">

<xsd:selector xpath="Customers\_Dim/customer\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="customer\_key\_ref" refer="customer\_key">

<xsd:selector xpath="Performance\_fact/customer\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="Orders\_key">

<xsd:selector xpath="Orders/OrderID" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Orders\_key\_ref" refer="Orders\_key">

<xsd:selector xpath="Performance\_fact/orders\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="Employee\_key">

<xsd:selector xpath="Employees\_Dim/Employee\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Employee\_key\_ref" refer="Employee\_key">

<xsd:selector xpath="Performance\_fact/Emp\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="Calender\_key">

<xsd:selector xpath="Calender\_Dim/Calender\_key" />

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="Calender\_key\_ref" refer="Calender\_key">

<xsd:selector xpath="Performance\_fact/calender\_key" />

<xsd:field xpath="." />

</xsd:keyref>

<xsd:key name="shippers\_key">

<xsd:selector xpath="Shippers\_Dim/shippers\_key"/>

<xsd:field xpath="." />

</xsd:key>

<xsd:keyref name="shipper\_key\_ref" refer="shippers\_key">

<xsd:selector xpath="Performance\_fact/shipper\_key"/>

<xsd:field xpath="." />

</xsd:keyref>

</xsd:element>

<xsd:element name="Customers\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="customer\_key" type="xsd:int" />

<xsd:element name="CustomerID" type="xsd:string" />

<xsd:element name="CompanyName" type="xsd:string" />

<xsd:element name="ContactName" type="xsd:string" />

<xsd:element name="ContactTitle" type="xsd:string" />

<xsd:element name="Address\_" type="xsd:string" />

<xsd:element name="City" type="xsd:string" />

<xsd:element name="Region" type="xsd:string" />

<xsd:element name="PostalCode" type="xsd:int" />

<xsd:element name="Country" type="xsd:string" />

<xsd:element name="Phone" type="xsd:int" />

<xsd:element name="Fax" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Orders">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="OrderID" type="xsd:int" />

<xsd:element name="CustomerID" type="xsd:string" />

<xsd:element name="EmployeeID" type="xsd:int" />

<xsd:element name="OrderDate" type="xsd:dateTime" />

<xsd:element name="RequiredDate" type="xsd:dateTime" />

<xsd:element name="ShippedDate" type="xsd:dateTime" />

<xsd:element name="ShipVia" type="xsd:int" />

<xsd:element name="Freight" type="xsd:float" />

<xsd:element name="ShipName" type="xsd:string" />

<xsd:element name="ShipAddress" type="xsd:string" />

<xsd:element name="ShipCity" type="xsd:string" />

<xsd:element name="ShipRegion" type="xsd:string" />

<xsd:element name="ShipPostalCode" type="xsd:int" />

<xsd:element name="ShipCountry" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Employees\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Employee\_key" type="xsd:int" />

<xsd:element name="EmployeeID" type="xsd:int" />

<xsd:element name="LastName" type="xsd:string" />

<xsd:element name="FirstName" type="xsd:string" />

<xsd:element name="Title" type="xsd:string" />

<xsd:element name="TitleOfCourtesy" type="xsd:string" />

<xsd:element name="BirthDate" type="xsd:dateTime" />

<xsd:element name="HireDate" type="xsd:dateTime" />

<xsd:element name="Address\_" type="xsd:string" />

<xsd:element name="City" type="xsd:string" />

<xsd:element name="Region" type="xsd:string" />

<xsd:element name="PostalCode" type="xsd:string" />

<xsd:element name="Country" type="xsd:string" />

<xsd:element name="HomePhone" type="xsd:string" />

<xsd:element name="Extension" type="xsd:int" />

<xsd:element name="Photo" type="xsd:string" />

<xsd:element name="Notes" type="xsd:string" />

<xsd:element name="ReportsTo" type="xsd:int" />

<xsd:element name="PhotoPath" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Calender\_Dim">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Calender\_key" type="xsd:int" />

<xsd:element name="FullDate" type="xsd:dateTime" />

<xsd:element name="DayofWeek\_" type="xsd:string" />

<xsd:element name="DayType" type="xsd:string" />

<xsd:element name="DayofMonth\_" type="xsd:int" />

<xsd:element name="Month\_" type="xsd:int" />

<xsd:element name="Quarter\_" type="xsd:string" />

<xsd:element name="Year\_" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Shippers\_Dim">

<xsd:complexType>

<xsd:all>

<xsd:element name="shippers\_key" type="xsd:int" />

<xsd:element name="ShipperID" type="xsd:int" />

<xsd:element name="CompanyName" type="xsd:string" />

</xsd:all>

</xsd:complexType>

</xsd:element>

<xsd:element name="Performance\_fact">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="delivery\_key" type="xsd:int" />

<xsd:element name="customer\_key" type="xsd:int" />

<xsd:element name="orders\_key" type="xsd:int" />

<xsd:element name="Emp\_key" type="xsd:int" />

<xsd:element name="calender\_key" type="xsd:int" />

<xsd:element name="shipper\_key" type="xsd:int" />

<xsd:element name="customer\_ship\_country" type="xsd:string" />

<xsd:element name="Customer\_company" type="xsd:string" />

<xsd:element name="shippers\_company" type="xsd:string" />

<xsd:element name="Employee\_name" type="xsd:string" />

<xsd:element name="order\_require\_date" type="xsd:dateTime" />

<xsd:element name="order\_shipped\_date" type="xsd:dateTime" />

<xsd:element name="ship\_status" type="xsd:string" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

# 7. graph database

For the given project we are using neo4j tool for creating the graph database. Comparison of graph database and relational database that we will discuss here.

**Graph database:**

Very simply, a graph database is a database designed to treat the relationships between data as equally important to the data itself. It is intended to hold data without constricting it to a pre-defined model. Instead, the data is stored like we first draw it out – showing how each individual entity connects with or is related to others.

The primary difference is that in a graph database, the relationships are stored at the individual record level, while in a relational database, the structure is defined at a higher level (the table definitions).

This has important ramifications:

* A relational database is much faster when operating on huge numbers of records. In a graph database, each record has to be examined individually during a query in order to determine the structure of the data, while this is known ahead of time in a relational database.
* Relational databases use less storage space, because they don't have to store all of those relationships.

Storing all of the relationships at the individual-record level only makes sense if there is going to be a lot of variation in the relationships; otherwise you are just duplicating the same things over and over. This means that graph databases are well-suited to irregular, complex structures. But in the real world, most databases require regular, relatively simple structures. This is why relational databases predominate

**7.1Neo4j creating Node Label creating query query:**

1. **Employees\_Dim label node:**

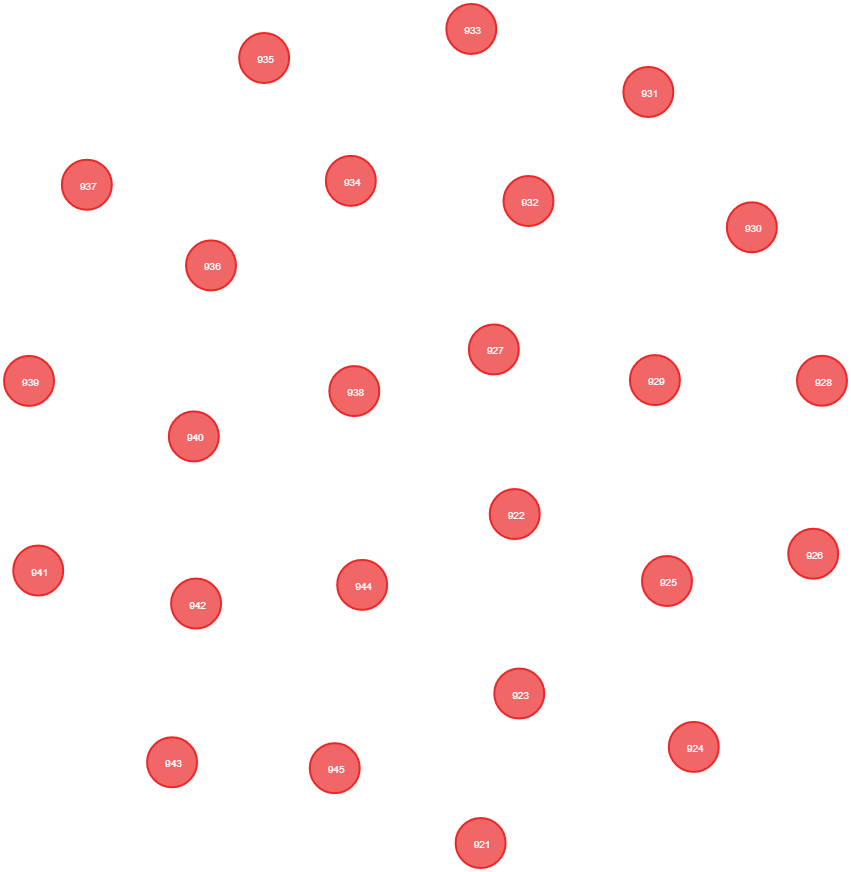
load csv with headers from "file:///EmployeeDim\_csv.csv" as row CREATE (employee\_Dim:Employees\_dim) SET employee\_Dim = row.emp\_key

:row.emp\_key

,name:row.FirstName

}

return employee\_Dim



1. **Customer Dim label node:**

load csv with headers from "file:///Customers\_Dim.csv" as row CREATE (customer\_dim:Customers\_Dim) SET customer\_dim= row.customer\_key

:row.customer\_key

,name:row.CompanyName

}

return customer\_dim



1. **Shipper Dim label node:**

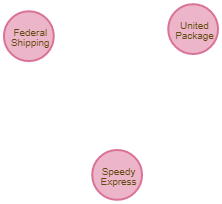
load csv with headers from "file:///Shippers\_Dim.csv" as row CREATE (shipper\_dim:Shippers\_Dim) SET shipper\_dim= row.shippers\_key

:row.shippers\_key

,name:row.CompanyName

}

return shipper\_dim



1. **Order Dim label node:**

load csv with headers from "file:///orders\_Dim.csv" as row CREATE (order\_dim:Order\_Dim) SET order\_dim= row.orders\_key

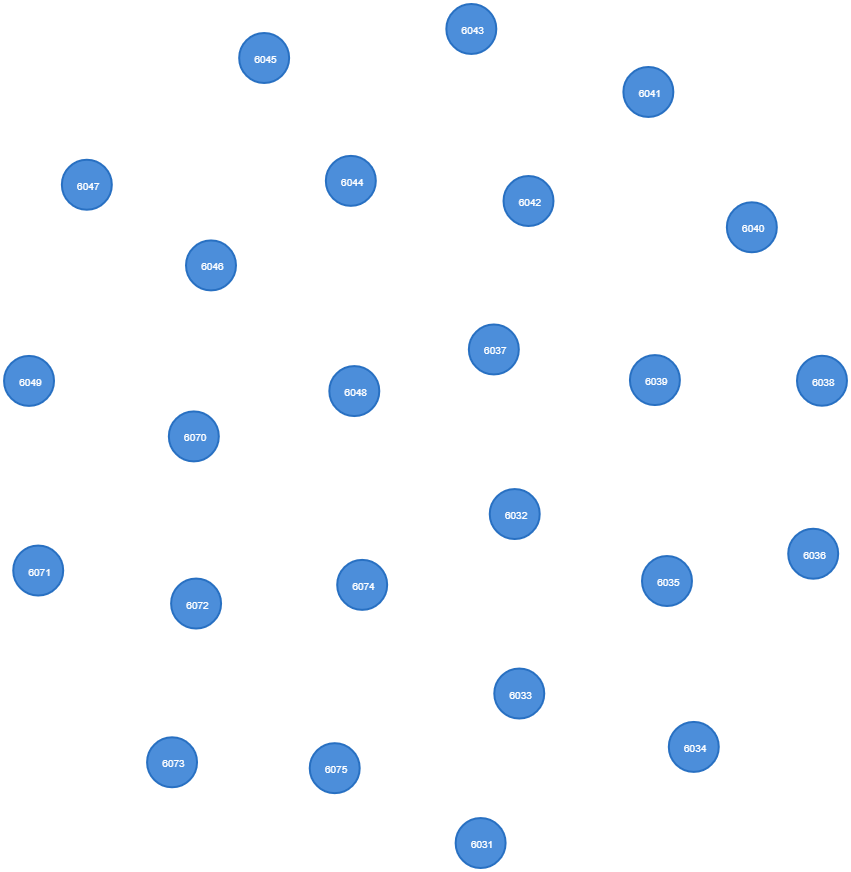
:row.orders\_key

,name:row.RequiredDate

,name:row.ShippedDate

}

return order\_dim



1. **Fact label node creating query:**

load csv with headers from "file:///Performance\_fact.csv" as row CREATE (performance:Performance\_fact)

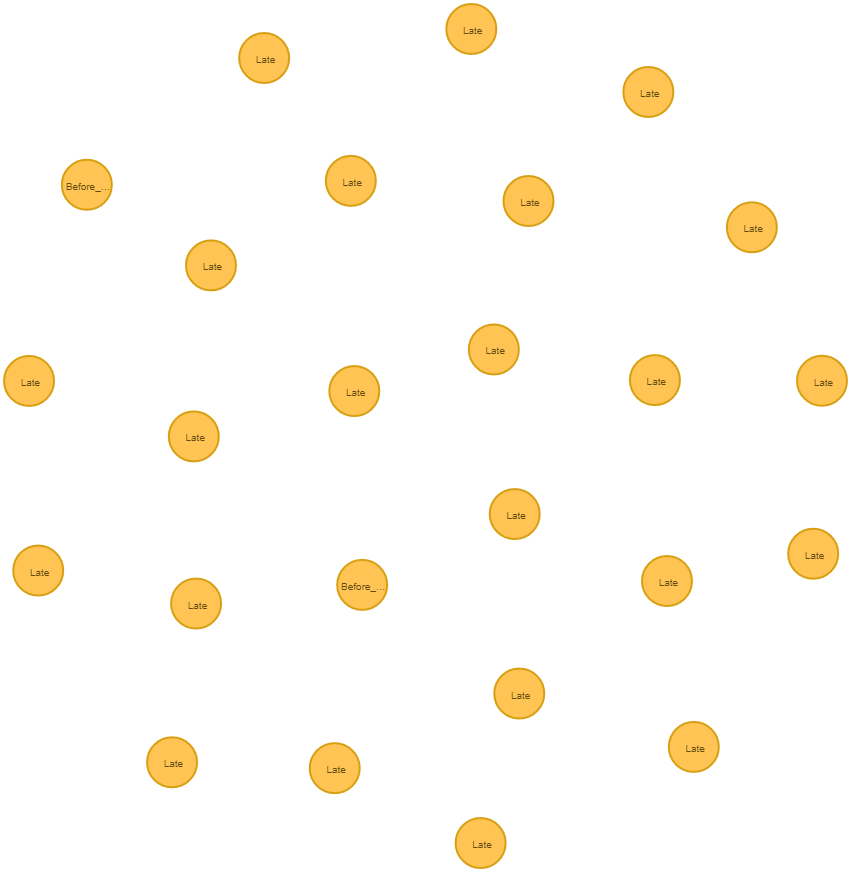
SET performance= row{delivery\_key:row.delivery\_key,customer\_key:row.customer\_key,orders\_key:row.orders\_key,Emp\_key:row.Emp\_key,

shipper\_key:row.shipper\_key,customer\_ship\_country:row.customer\_ship\_country,

customer\_ship\_country:row.customer\_ship\_country,Customer\_company:row.Customer\_company,shippers\_company:row.shippers\_company,

Employee\_name:row.Employee\_name,ship\_status:row.ship\_status}

return performance

****

**7.2 Creating constraint query:**

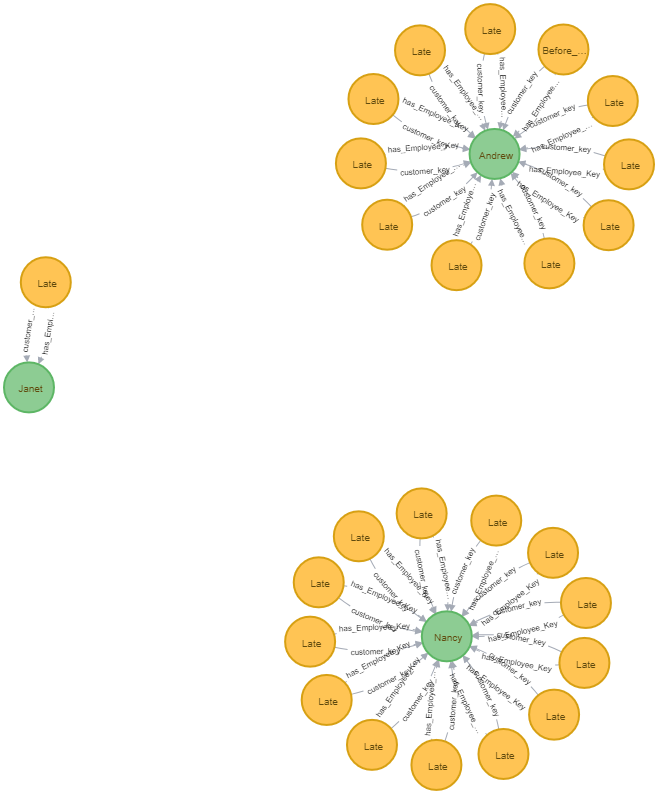
1. CREATE CONSTRAINT ON (s:EmployeeDim\_csv) ASSERT f.emp\_key IS UNIQUE
2. CREATE CONSTRAINT ON (s:Customers\_Dim) ASSERT f.customer\_key IS UNIQUE
3. CREATE CONSTRAINT ON (s:Shippers\_Dim) ASSERT f.shippers\_key IS UNIQUE

**7.3 Node Relationship matching query:**

1. **Employee Dim node and Performance fact query:**

MATCH (n:EmployeeDim\_csv),(m:Performance\_fact)

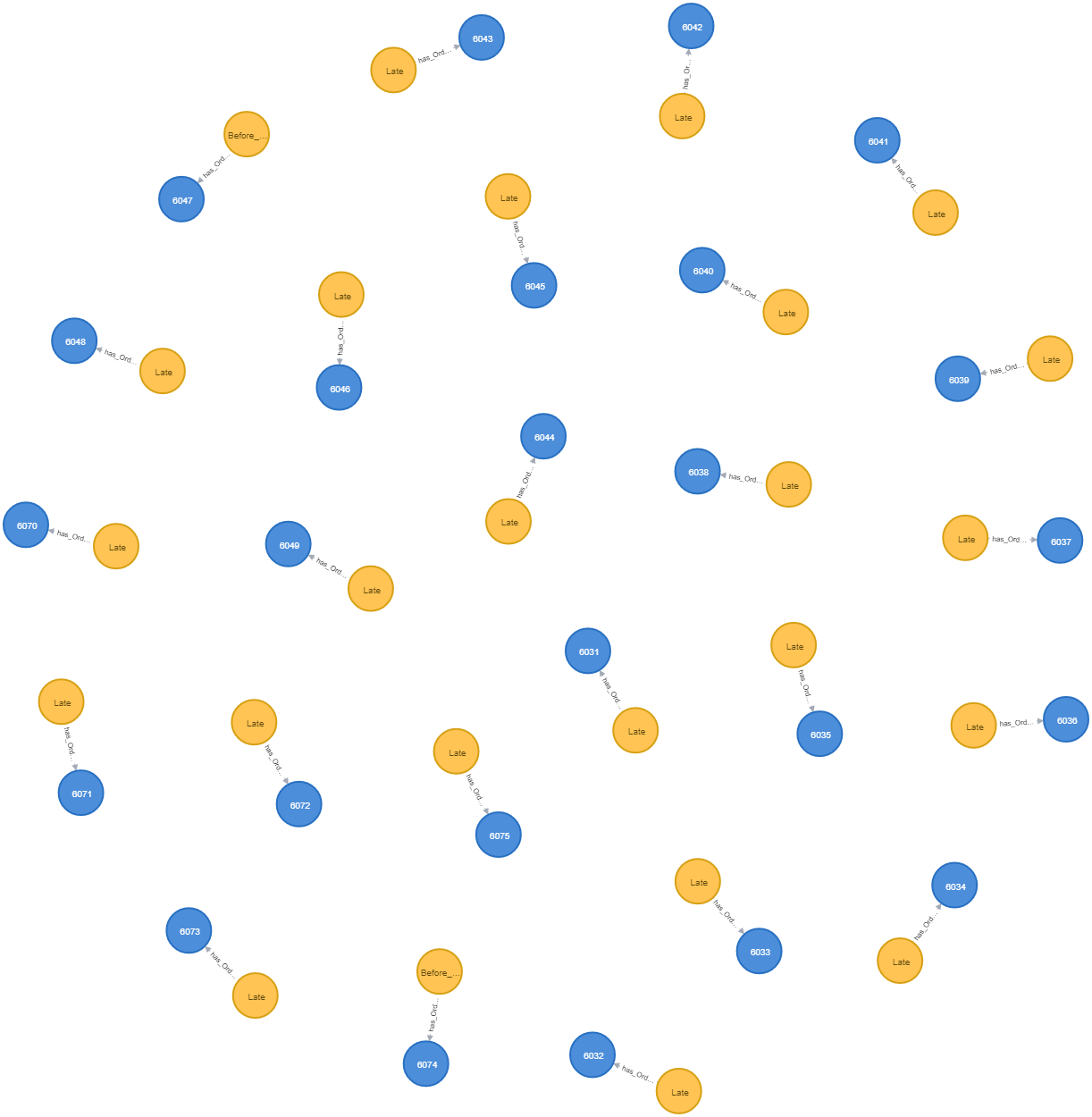
where n.EmployeeID=m.Emp\_key CREATE (m)-[r:has\_Employee\_Key]->(n) return n,r,m



1. **Order Dim node and Performance fact query:**

MATCH (n:Order\_Dim),(m:Performance\_fact)

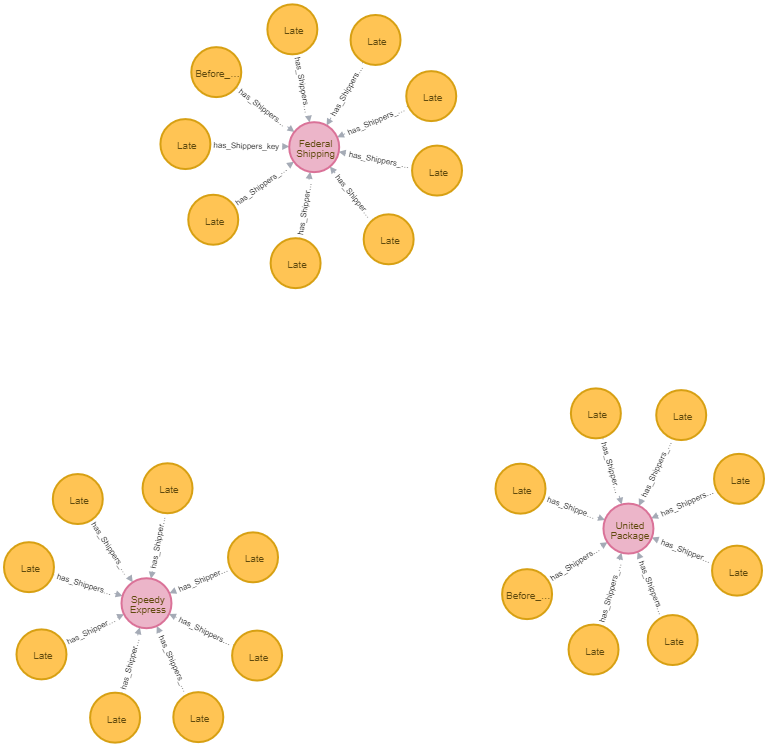
where n.orders\_key=m.orders\_key CREATE (m)-[r:has\_order\_Key]->(n) return n,r,m



1. **Shipper Dim node and Performance fact query:**

MATCH (n:Shippers\_Dim),(m:Performance\_fact)

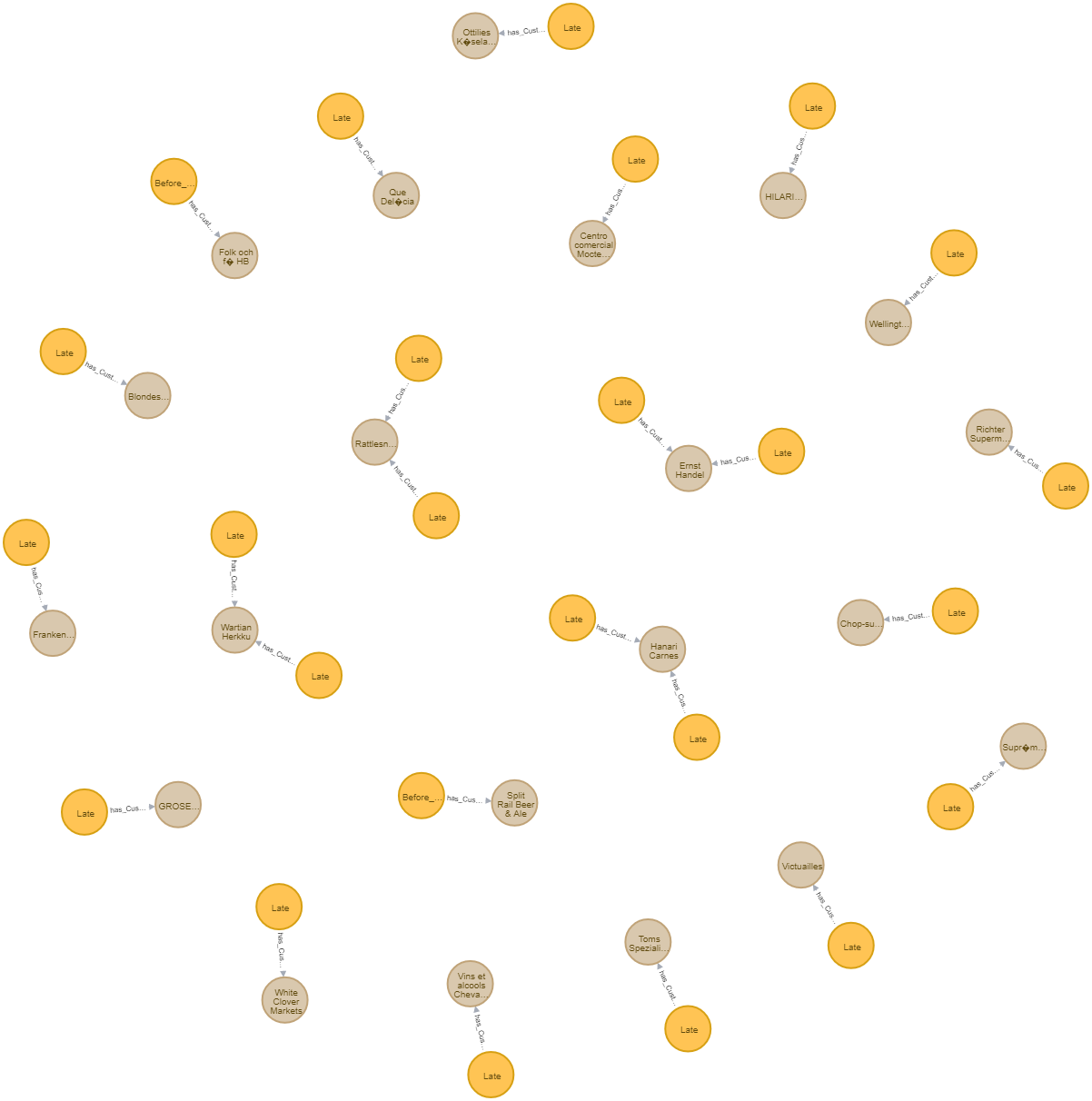
where n.shippers\_key=m.shippers\_key CREATE (m)-[r:has\_shipper\_Key]->(n) return n,r,m

****

1. **customer Dim node and Performance fact query:**

MATCH (n:Customers\_Dim),(m:Performance\_fact)

where n.customer\_key=m.customer\_key CREATE (m)-[r:has\_Customer\_Key]->(n) return n,r,m

****

* 1. **All dimension label node with fact node matching query:**

match(n:Performance\_fact),(c:EmployeeDim\_csv),(x:Shippers\_Dim) where n.Emp\_key=c.EmployeeID

and n.shipper\_key=x.shippers\_key return n,c,x limit 2000

