

The Product Company

~ Final Data Mart Development Report ~

Team 5

Team Members

Muriel Banze

Sahil Shah

Siddharth Chauhan

Date

27th November 2020

~ Table of Contents ~

I. Data Mart Design Definition	2
1. Universe of Discourse	2
2. Information Package	2
II. Dimensional Model	6
III. Data Staging: ETL – Data Extract File Definitions	7
IV. Data Staging: ETL – Source-to-Target Mappings	14
V. SQL Code – Tables & Constraints	15
VI. Data Staging Activities - ETL	19
1. Data Cleansing	19
2. Data Transformation	22
3. Table Population	23
VII. End User Applications	25
1. Queries	25
2. A View	28
VIII. Handling Slowly Changing Dimensions (SCD)	39

I. Data Mart Design Definition

1. Universe of Discourse

The universe of debate for the data mart is to handle The Product Company's distribution to its consumers across its three branches, namely PEC, TPCE, TPCW.

In order to maximize revenue with lower costs and maintain a stable relationship with vendors, the data mart also handles historical revenues, reports and net profit for all divisions.

2. Information Package

Process Name: TPC and its Division's Financial Results.

Grain: Daily Sales for each customer, Product

Customer	Product	Order_Date	Sales_Date	Junk
Customer_SK	Product_SK	Order_Date_SK	Sales_Date_SK	Junk_SK
CustomerID_NK	ProductID_NK	Order_Date	Sales_Date	Ship_Method
Customer_Name	Product_Name	Order_Year	Sales_Year	Payment_Method
Customer_dept	Price1	Order_Quarter	Sales_Quarter	Order_Method
Customer_suite	Price2	Order_Month	Sales_Month	
Customer_address	UnitCost	Order_Week	Sales_Week	
Customer_city	Supplier_Name	Order_Fiscal_Year	Sales_Fiscal_Year	
Customer_state	Supplier_Address	Order_Fiscal_Quarter	Sales_Fiscal_Quarter	
Customer_zip	Supplier_Attn	Order_Fiscal_Month	Sales_Fiscal_Month	
Customer_custtypeID	Supplier_City	Order_Fiscal_Week	Sales_Fiscal_Week	
Customer_custtypeName	Supplier_State			
Customer_division	Supplier_Zip			
	ProductTypeID			
	TypeDescription			
	BUID			
	Name			
	Abbreviation			
	Division			

Facts: Discounted, Ship_Cost, Amount, Quantity, InvoiceID (DD)

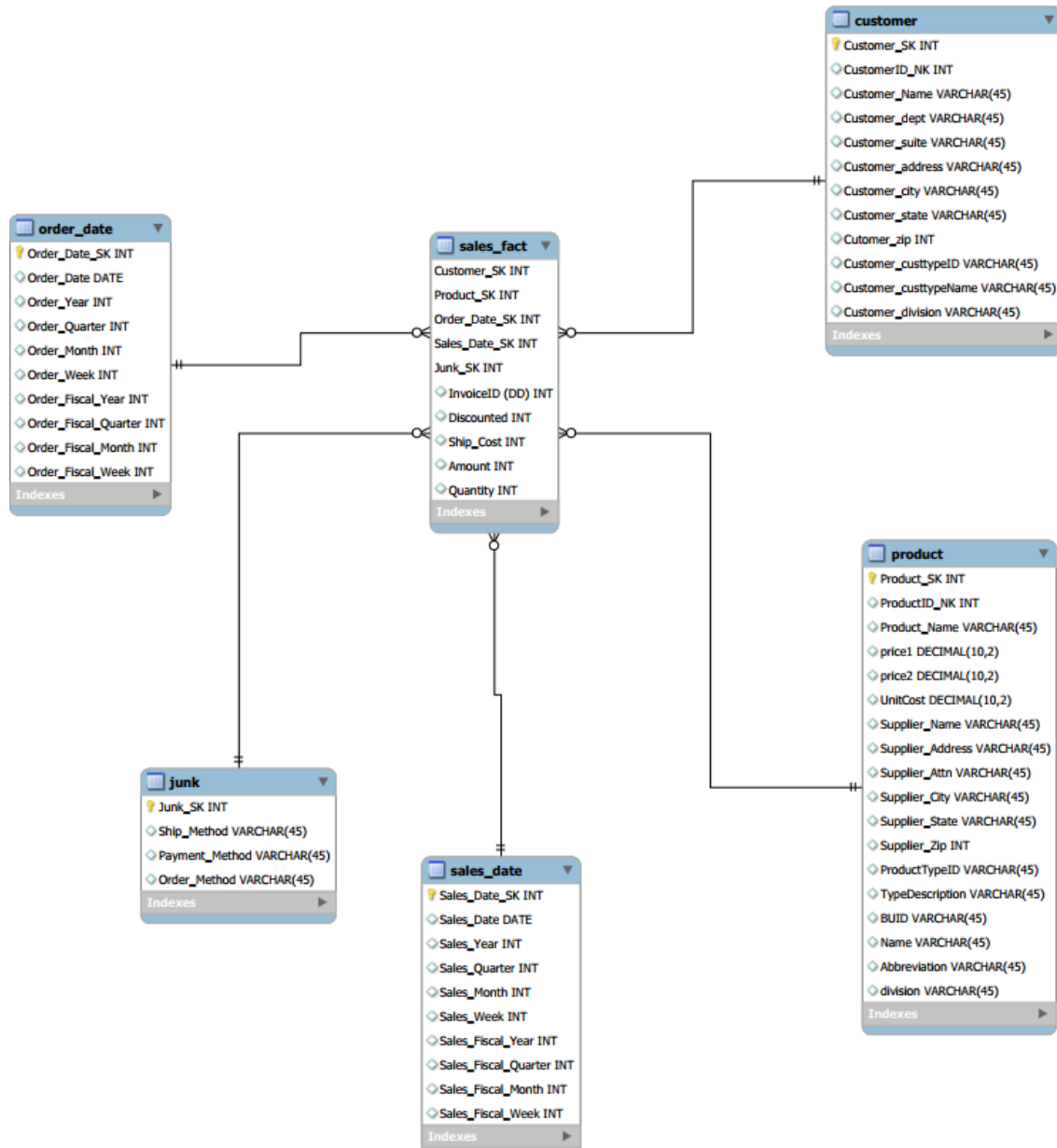
3. Entity Definitions

Entity	Entity Definition (<i>genus differentia</i>)
Customer	<p>The entity holds the all the information of all existing customers in the Organization over all divisions – PEC, TPCW, TPCE.</p> <ol style="list-style-type: none"> 1. Customer_SK – All the unique surrogate keys assigned to each customers present in every department. 2. CustomerID_NK – Natural keys of customers 3. Customer_Name – Name of the customer 4. Customer_dept- Department Number of the customer 5. Customer_suite – Suite Number of the customer 6. Customer_Address – Address of the Customer 7. Customer_City- City of the customer in the Address 8. Customer_State – State where the customers lives 9. Customer_ZIP – Zipcode of the customer’s Address 10. Customer_custtypeID – Code of the CustomerType 11. Customer_custtypeName- Name of the CustomerType 12. Customer_Division- Defines from which division the customer is from PEC, TPCE or TPCW.
Product	<p>The entity holds the all the information of all existing Products in the Organization over all divisions – PEC, TPCW, TPCE</p> <ol style="list-style-type: none"> 1. Product_SK – Surrogate Key of Product Dimension 2. ProductID_NK – Natural Key of Product Dimension 3. Product_Name- Name of the Product 4. Price1 – Price when discount = 0 5. Price2 – Price when discount = 1 6. UnitCost – Cost of the Unit 7. Supplier_Name- Name of the Supplier 8. Supplier_Address – Address of the Supplier 9. Supplier_Attn – The head representative of the supplier. 10. Supplier_City – City when the Supplier is located 11. Supplier_State – State where the Supplier is located 12. Supplier_ZIP – Zipcode of the Address 13. ProductTypeID – ID of the product type 14. TypeDescription – Description of the Product Type 15. BUID – Business Unit ID 16. Name – Name of the Business Unit 17. Abbreviation – Abbreviation of the Business Unit

	18. Division – The division from which companies manufactures and supplies (PEC, TPCE, or TPCW).
Order_Date	<p>It is a role-playing dimension in our dimensional model. It represents order date across all divisions namely (PEC, TPCE, TPCW)</p> <ol style="list-style-type: none"> 1. Order_Date_SK – Surrogate Key of the Order Date 2. Order_Date – Actual date of the Order(MM/DD/YYYY) 3. Order_Year- Year of the Order 4. Order_Quarter- Quarter of the date 5. Order_Month – Month of the Order 6. Order_Week – Week of the Order 7. Order_Fiscal_Year – Fiscal Year of order 8. Order_Fiscal_Quarter – Fiscal Quarter of Order 9. Order_Fiscal_Month – Fiscal Month of Order 10. Order_Fiscal_Week – Fiscal Week of Order
Sales_Date	<p>It is a role-playing dimension in our dimensional model. It represents sales date across all divisions namely (PEC, TPCE, TPCW)</p> <ol style="list-style-type: none"> 1. Sales_Date_SK – Surrogate key of Sales_Date 2. Sales_Date – Actual Sales Date(MM/DD/YYYY) 3. Sales_Year – Year of the Sales Date 4. Sales_Quarter – Quarter of Sales Date 5. Sales_Month – Month of Sales Date 6. Sales_Week – Week of Sales Date 7. Sales_Fiscal_Year – Fiscal Year of Sales Date 8. Sales_Fiscal_Quarter – Fiscal Quarter of Sales Date 9. Sales_Fiscal_Month – Fiscal Month of Sales Date 10. Sales_Fiscal_Week – Fiscal Week of Sales Date
Junk	<p>This is a Junk Dimension since attributes are often flag-like in nature as they do not belong to any dimension. This specifies the form of system of delivery, payment and ordering in all dimensions.</p> <ol style="list-style-type: none"> 1. Junk_SK – Surrogate Key of Junk 2. Ship_Method – Method of Shipping 3. Order_Method – Ordering Method 4. Payment_Method – Method of Payment
Sales_Fact	<p>The fact table connecting all the dimensional tables.</p> <ol style="list-style-type: none"> 1. Customer_SK – The surrogate acting of the Customer, which acts as the Constraint as Foreign and composite primary key in the fact table.

	<ol style="list-style-type: none"> 2. Product_SK – The surrogate key of the product which acts as the foreign and composite primary key in the fact table 3. Order_Date_SK - The surrogate key of the Order Date which acts as the foreign and composite primary key in the fact table 4. Sales_Date_SK - The surrogate key of the Sales Date which acts as the foreign and composite primary key in the fact table 5. Junk_SK - The surrogate key of the Junk which acts as the foreign and composite primary key in the fact table 6. Amount – Total Cost of the product 7. Quantity – Quantity of the product that were ordered 8. Discounted – Defines if the product was sold on discounted price or no (0/1) 9. Ship_Cost – The cost of the shipping 10. InvoiceID(DD) – Acts and the degenerate dimension in the fact table
--	---

II. Dimensional Model



III. Data Staging: ETL – Data Extract File Definitions

Data Source – PEC

Index	File_Name	Format	Datatype
1	PECbusiness_unit.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	BUID - String NAME - String ABBREV - String
2	PECcustomer.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	BUID - String NAME - String ABBREV - String
3	PECcustomer_type.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	CUSTTYPEID- String TYPENAME-String
4	PECinvoice.csv	Fields separated by comma (,)	Invoice-Integer, Cust-ID-Integer, salesDate-Date, prodid-integer, amt-Integer, qty-Integer, shipMethod-String, shipCost-Decimal, paymentMethod-String , orderMethod-String, orderDate-Date, discounted-Integer
5	PECmanufacturingCosts.csv	Fields separated by pipe ()	Year- Integer,

			Month- Integer, ProdID- Integer, manufacturingCost- Integer
6	PECproduct_type.csv	Fields enclosed in double quotes (") and separated by semicolon (;). Rows enclosed by double quotes (")	PRODTYPEID-String, TYPEDESCRIPTION-String, BUID- String
7	PECproduct.csv	Fields enclosed in double quotes (") and separated by semicolon (;). Rows enclosed by double quotes (")	prodid- Integer, prodDescription-String, price1- Decimal, price2- Decimal, unitCost- Decimal, supplierName- String, productTypeID- Integer

Data Source – TPCW

Index	File_Name	Format	Data Type
1	TPCWbusiness_unit.csv	Fields enclosed in double quotes (") and separated by semicolon (;)	BUID – String, NAME- String, ABBREV-String
2	TPCWcustomer_type.csv	Fields enclosed in double quotes (") and separated by semicolon (;)	CUSTTYPEID - String, TYPENAME-String
3	TPCWcustomer.csv	Fields enclosed in double quotes (") and	custID-Integer,

		separated by semicolon (;)	name-String, address-String, city-String, state-String, zip-Integer, custType - String
4	TPCWinvoice.csv	Fields separated by comma (,)	Invoice-Integer, custID-Integer, prodID-Integer, salesDate-String, amt-Integer, qty-Integer, discounted-Integer
5	TPCWproduct_type.csv	Fields enclosed in double quotes (") and separated by semicolon (;). Rows enclosed by double quotes (")	PRODTYPEID- String, TYPEDESCRIPTION-String, BUID- String
6	TPCWproduct.csv	Fields enclosed in double quotes (") and separated by semicolon (;). Rows enclosed by double quotes (")	ProductID- Integer, ProductName- String, Price1- Number, Price2- Number, Unit Cost- Number, Supplier Name- String, Supplier Address- String,

			Supplier city- String, Supplier State- String, Supplier zipcode- String, Product Type ID- Integer
--	--	--	--

Data Source – TPCE

Index	File_Name	Format	Data Type
1	business_unit.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	BUID – String, NAME- String, ABBREV-String
2	invoice.csv	Fields separated by comma (,)	InvoiceID-Integer, custID-Integer, salesDate-Date
3	invoice_details.csv	Fields separated by comma (,)	InvoiceID – Integer, prodID- Integer, amt- Decimal, qty- Integer, discounted-Integer
4	customer_type.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	CUSTTYPEID - String, TYPENAME-String
5	customer.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;)	CUSTID-Integer,

			NAME-String, ADDR1-String, ADDR2- String, CITY-String, STATE-String, ZIP-Integer, CUSTTYPEID-String
6	supplier.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;).	SUPPLIERID-Integer, NAME- String, ADDR1- String, ADDR2- String, CITY- String, STATE- String, ZIP- Integer
7	product.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;).	ProductID- Integer, ProductName- String, Price1- Number, Price2- Number, Unit Cost- Number, Supplier Name- String, Supplier Address-String, Supplier city- String, Supplier State- String,

			Supplier zipcode-String, Product Type ID- Integer
8	prod_type.csv	Fields enclosed in double quotes (“”) and separated by semicolon (;).	PRODTYPEID- String, TYPEDESCRIPTION- String, BUID- String

IV. Data Staging: ETL – Source-to-Target Mappings

Reference File : Mapping.xlsx

Table_name	Column_name	TARGET	Table_type	SCD_Type	Database_name	SOURCE	Column_name	Data_type	Transformation
		Data Type				File_name			
						OR Table_name			
customer	Customer_SK	INT	Dimension	0		PECcustomer.csv	CustID	NUMBER	Surrogate key
customer	CustomerID_NK	INT	Dimension	0		TPCWcustomer.csv	custid	NUMBER	Natural key for customer_dimension
					chauhan_salesorder	customer	customerID	INT(11)	Natural key for customer_dimension
customer	Customer_Name	VARCHAR	Dimension	6		PECcustomer.csv	name	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	name	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	name	VARCHAR(50)	Pentaho (Customer.ktr)
customer	Customer_suite	VARCHAR	Dimension	6		PECcustomer.csv	address	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	address	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	addr1	VARCHAR(45)	Pentaho (Customer.ktr)
customer	Customer_dept	VARCHAR	Dimension	6		PECcustomer.csv	address	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	address	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	addr1	VARCHAR(45)	Pentaho (Customer.ktr)
customer	Customer_address	VARCHAR	Dimension	6		PECcustomer.csv	address	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	address	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	addr2	VARCHAR(50)	Pentaho (Customer.ktr)
customer	Customer_city	VARCHAR	Dimension	6		PECcustomer.csv	city	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	city	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	city	VARCHAR(16)	Pentaho (Customer.ktr)
customer	Customer_state	VARCHAR	Dimension	6		PECcustomer.csv	state	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	state	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	state	CHAR(2)	Pentaho (Customer.ktr)
customer	Customer_zip	INT	Dimension	6		PECcustomer.csv	zip	STRING	Pentaho (Customer.ktr)
					TPCWcustomer.csv	customer	zip	STRING	Pentaho (Customer.ktr)
					chauhan_salesorder	customer	zip	CHAR(15)	Pentaho (Customer.ktr)
customer	Customer_custTypeID	VARCHAR	Dimension	2		PECcustomer_type.csv	CUSTTYPEID	STRING	Pentaho (Customer.ktr)

V. SQL Code – Tables & Constraints

Creating `salesorder_5_2201`

Query:

```
CREATE DATABASE IF NOT EXISTS `salesorder_5_2201` /*!40100 DEFAULT  
CHARACTER SET utf8 */ /*!80016 DEFAULT ENCRYPTION='N' */;  
USE `salesorder_5_2201`;
```

Customer

Query:

```
DROP TABLE IF EXISTS `customer`;  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!50503 SET character_set_client = utf8mb4 */;  
CREATE TABLE `customer` (  
  `Customer_SK` int NOT NULL AUTO_INCREMENT,  
  `CustomerID_NK` int DEFAULT NULL,  
  `Customer_Name` varchar(45) DEFAULT NULL,  
  `Customer_dept` varchar(45) DEFAULT NULL,  
  `Customer_suite` varchar(45) DEFAULT NULL,  
  `Customer_address` varchar(45) DEFAULT NULL,  
  `Customer_city` varchar(45) DEFAULT NULL,  
  `Customer_state` varchar(45) DEFAULT NULL,  
  `Cutomer_zip` int DEFAULT NULL,  
  `Customer_custtypeID` varchar(45) DEFAULT NULL,  
  `Customer_custtypeName` varchar(45) DEFAULT NULL,  
  `Customer_division` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`Customer_SK`)  
)
```

Product

Query:

```
DROP TABLE IF EXISTS `product`;  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!50503 SET character_set_client = utf8mb4 */;  
CREATE TABLE `product` (  
  `Product_SK` int NOT NULL AUTO_INCREMENT,  
  `ProductID_NK` int DEFAULT NULL,  
  `Product_Name` varchar(45) DEFAULT NULL,  
  `price1` decimal(10,2) DEFAULT NULL,  
  `price2` decimal(10,2) DEFAULT NULL,  
  `UnitCost` decimal(10,2) DEFAULT NULL,  
  `Supplier_Name` varchar(45) DEFAULT NULL,  
  `Supplier_Address` varchar(45) DEFAULT NULL,  
  `Supplier_Attn` varchar(45) DEFAULT NULL,
```

```

`Supplier_City` varchar(45) DEFAULT NULL,
`Supplier_State` varchar(45) DEFAULT NULL,
`Supplier_Zip` int DEFAULT NULL,
`ProductTypeID` varchar(45) DEFAULT NULL,
`TypeDescription` varchar(45) DEFAULT NULL,
`BUID` varchar(45) DEFAULT NULL,
`Name` varchar(45) DEFAULT NULL,
`Abbreviation` varchar(45) DEFAULT NULL,
`division` varchar(45) DEFAULT NULL,
PRIMARY KEY (`Product_SK`)
)

```

Order Date

Query:

```

DROP TABLE IF EXISTS `order_date`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `order_date` (
  `Order_Date_SK` int NOT NULL AUTO_INCREMENT,
  `Order_Date` date DEFAULT NULL,
  `Order_Year` int DEFAULT NULL,
  `Order_Quarter` int DEFAULT NULL,
  `Order_Month` int DEFAULT NULL,
  `Order_Week` int DEFAULT NULL,
  `Order_Fiscal_Year` int DEFAULT NULL,
  `Order_Fiscal_Quarter` int DEFAULT NULL,
  `Order_Fiscal_Month` int DEFAULT NULL,
  `Order_Fiscal_Week` int DEFAULT NULL,
  PRIMARY KEY (`Order_Date_SK`)
)

```

Sales Date

Query:

```

DROP TABLE IF EXISTS `sales_date`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `sales_date` (
  `Sales_Date_SK` int NOT NULL AUTO_INCREMENT,
  `Sales_Date` date DEFAULT NULL,
  `Sales_Year` int DEFAULT NULL,
  `Sales_Quarter` int DEFAULT NULL,
  `Sales_Month` int DEFAULT NULL,
  `Sales_Week` int DEFAULT NULL,
  `Sales_Fiscal_Year` int DEFAULT NULL,

```



```

`Sales_Fiscal_Quarter` int DEFAULT NULL,
`Sales_Fiscal_Month` int DEFAULT NULL,
`Sales_Fiscal_Week` int DEFAULT NULL,
PRIMARY KEY (`Sales_Date_SK`)
)

```

Junk

Query:

```

DROP TABLE IF EXISTS `junk`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `junk` (
  `Junk_SK` int NOT NULL AUTO_INCREMENT,
  `Ship_Method` varchar(45) DEFAULT NULL,
  `Payment_Method` varchar(45) DEFAULT NULL,
  `Order_Method` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`Junk_SK`)
)

```

Sales Fact

Query:

```

DROP TABLE IF EXISTS `sales_fact`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `sales_fact` (
  `Customer_SK` int NOT NULL,
  `Product_SK` int NOT NULL,
  `Order_Date_SK` int NOT NULL,
  `Sales_Date_SK` int NOT NULL,
  `Junk_SK` int NOT NULL,
  `InvoiceID (DD)` int DEFAULT NULL,
  `Discounted` int DEFAULT NULL,
  `Ship_Cost` decimal(10,2) DEFAULT NULL,
  `Amount` decimal(10,2) DEFAULT NULL,
  `Quantity` int DEFAULT NULL,
  PRIMARY KEY
  (`Customer_SK`,`Product_SK`,`Order_Date_SK`,`Sales_Date_SK`,`Junk_SK`),
  KEY `product_fk_idx` (`Product_SK`),
  KEY `salesDate_fk_idx` (`Sales_Date_SK`),
  KEY `orderDate_fk_idx` (`Order_Date_SK`),
  KEY `junk_fk_idx` (`Junk_SK`),
  KEY `customer_fk_idx` (`Customer_SK`),
  CONSTRAINT `customer_fk` FOREIGN KEY (`Customer_SK`) REFERENCES
`ustomer` (`Customer_SK`) ON DELETE NO ACTION ON UPDATE NO ACTION,

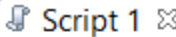
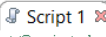
```

```
CONSTRAINT `junk_fk` FOREIGN KEY (`Junk_SK`) REFERENCES `junk`  
(`Junk_SK`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
CONSTRAINT `orderDate_fk` FOREIGN KEY (`Order_Date_SK`) REFERENCES  
`order_Date` (`Order_Date_SK`) ON DELETE NO ACTION ON UPDATE NO  
ACTION,  
CONSTRAINT `product_fk` FOREIGN KEY (`Product_SK`) REFERENCES `product`  
(`Product_SK`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
CONSTRAINT `salesDate_fk` FOREIGN KEY (`Sales_Date_SK`) REFERENCES  
`sales_Date` (`Sales_Date_SK`) ON DELETE NO ACTION ON UPDATE NO ACTION  
)
```

VI. Data Staging Activities - ETL

1. Data Cleansing

DM Table	Attribute	Problem	Resolution Strategy (attach code)																							
Customer																										
PEC_Customer	Custtype Address	<ul style="list-style-type: none">Extra CommasRemoving periods and field names with half names. For eg: St.,Av.,Dr.,Rd Changing Uppercase to lower case for custtype	<ul style="list-style-type: none">Replace in String function Use Constant to a define suite/dept as N/A as they are not present.																							
PEC_Custtype	CusttypeID TypeName	Extra Inverted Commas	Replace in String with setting as empty string.																							
TPCW_Customer	All fields	<ul style="list-style-type: none">Extra Inverted CommasPartial names for custtype field with Edu., Comm., Govt, State.,Partial names in name with Inc., Co, Corp.Partial names in Address field St, Rd, Ave. Lowercase in State with Fl and DC	<ul style="list-style-type: none">Replace in string for every field where the problem exists. <div><div>Fields string</div><table><tr><td>Search</td><td>Replace with</td><td rowspan="11"><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></td></tr><tr><td>Edu</td><td>Education</td></tr><tr><td>State</td><td>State/Local Gov</td></tr><tr><td>(Comm\$(Comm))\$</td><td>Commercial</td></tr><tr><td>Govt</td><td>US Govt</td></tr><tr><td>(Inc Inc.)\$</td><td>Incorporated</td></tr><tr><td>Co.\$</td><td>Company</td></tr><tr><td>Corp\$</td><td>Corporation</td></tr><tr><td>St.\$</td><td>Street</td></tr><tr><td>Rd.\$</td><td>Road</td></tr><tr><td>(Ave\$ Av.\$)</td><td>Avenue</td></tr></table></div>	Search	Replace with	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	Edu	Education	State	State/Local Gov	(Comm\$(Comm))\$	Commercial	Govt	US Govt	(Inc Inc.)\$	Incorporated	Co.\$	Company	Corp\$	Corporation	St.\$	Street	Rd.\$	Road	(Ave\$ Av.\$)	Avenue
Search	Replace with	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>																								
Edu	Education																									
State	State/Local Gov																									
(Comm\$(Comm))\$	Commercial																									
Govt	US Govt																									
(Inc Inc.)\$	Incorporated																									
Co.\$	Company																									
Corp\$	Corporation																									
St.\$	Street																									
Rd.\$	Road																									
(Ave\$ Av.\$)	Avenue																									

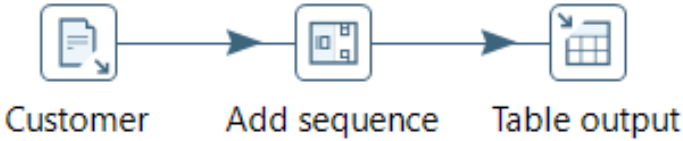
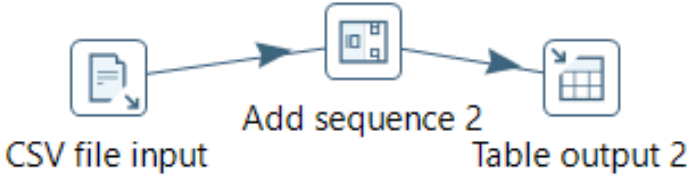
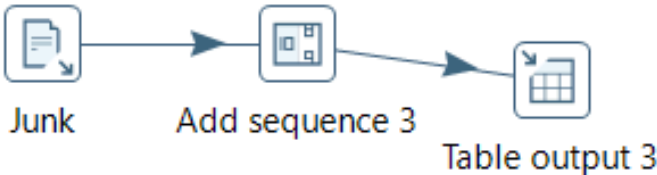
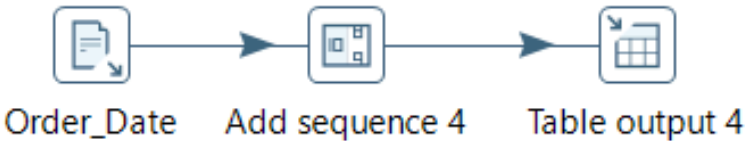
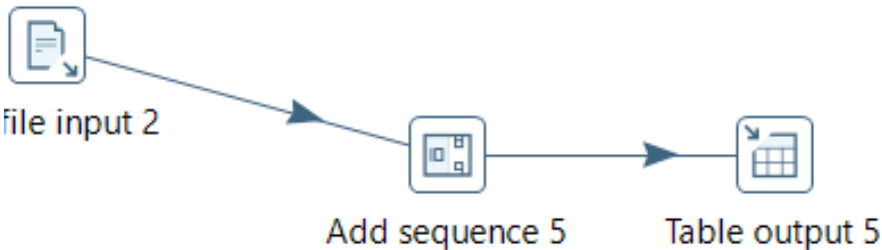
TPCE_Customer	Address	Dept and Suite in address field	<pre>var addr = addr1; //var su = "Suite"; if (addr.indexOf("Suite") != -1) { var suite = addr; } else { var suite = "N/A"; } if (addr.indexOf("Dept #") != -1) { var dept = addr1; } else { var dept = "N/A"; }</pre>												
TPCE_Custtype	typeName	(_) in State/Local Govt	Replace in String function to (/)												
Product															
PEC_product	All fields	<ul style="list-style-type: none">Replace with inverted commas Unit cost are calculated incorrectly.	<ul style="list-style-type: none">Replace in String and Equip to EquipmentWe merge the Product and manufacturing cost based on the Sales-date with respect to month and year. UnitPrice = manufacturingCost/TotalQty.												
PEC_Producttype	All fields	Replace with Inverted Commas	Replace is String and Equip to Equipment												
PEC_BusinessUnit	All fields	<ul style="list-style-type: none">Replace with Inverted Commas Abbreviation Missing in Miscellaneous	<ul style="list-style-type: none">Replace in String <table><tr><th>Search</th><th>Replace with</th><th>Set empty string</th></tr><tr><td>"</td><td></td><td>Y</td></tr><tr><td>"</td><td></td><td>Y</td></tr><tr><td>"</td><td></td><td>Y</td></tr></table>	Search	Replace with	Set empty string	"		Y	"		Y	"		Y
Search	Replace with	Set empty string													
"		Y													
"		Y													
"		Y													
PEC_manufacturingCost	Year	The year consist of only YY	 <pre>//Script here var Year = Year + 2000</pre>												
TPCW_businessUnit	Abbrev	Abbreviation Missing in Miscellaneous	 <pre>//Script here if (name.getString() == "Miscellaneous"){ abbrev.setValue("Misc") }</pre>												
TPCW_prodtype	All fields	Extra Inverted Commas	Replace in String with Set to Empty String												
TPCW_Product	All fields	<ul style="list-style-type: none">Extra Inverted CommasSuite and Dept in Same Address field	<ul style="list-style-type: none">Replace in String with Set to Empty StringSplitting Fields function												

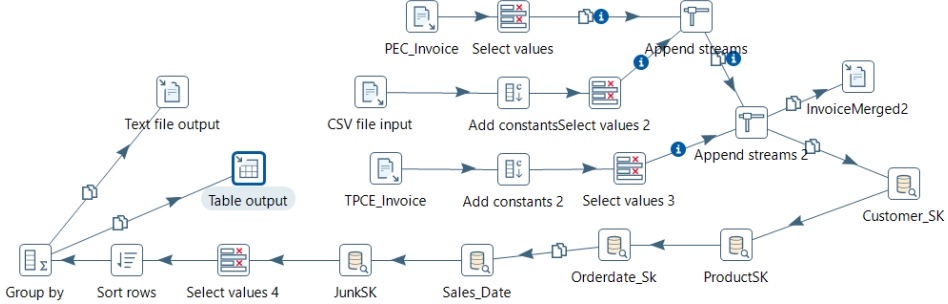
		Lowercase Supplier State	<div>Script 1</div> <div>//Script here</div> <div>var supplier_state = upper(supplier_state)</div>															
TPCE_BusinessUnit	Abbrev	Abbreviation Missing in Miscellaneous	<div>Script 1</div> <div>//Script here</div> <div>if (name.getString() == "Miscellaneous"){ abbrev.setValue("Misc") }</div>															
Order_Date																		
PEC_invoice	Order_date	Two records had black values, because shifting of columns	Used Modified Javascript to replace the values. Other approach was to make changes manually in excel.															
Sales_Date																		
PEC_invoice	saleDate	Incorrect Date and format of date	<ul style="list-style-type: none">Segregating the dates in to year, Month and week. Select values to change the date format to MMddYYYY															
TPCE_invoice	saleDate	Date in ddMMYYYY	Used Modified Java Script															
TPCW_invoice	salesDate	Inconsistencies in date with different formats ddmmyy and ddMMYYYY	<ul style="list-style-type: none">Used Microsoft Excel to make the changes. <div>Script 1</div> <div>//Script here</div> <div>if (invoice.getString() == "3032") { salesDate.setValue("20-08-05") amt.setValue("372") qty.setValue("52") }</div> <div>//Script here</div> <div>if(salesDate.indexOf('/')>-1){ arr2 = salesDate.split('/'); d2 = arr2[0]; m2 = arr2[1]; y2 = arr2[2]; var salesDate = m2 + "/" + d2 + "/" + y2; }</div>															
Junk																		
PEC_invoice	shipMethod	Incorrect names for air, train, truck.	<table><tr><th>Search</th><th>Replace with</th></tr><tr><td>aiir</td><td>air</td></tr><tr><td>trainn</td><td>train</td></tr><tr><td>trick</td><td>truck</td></tr><tr><td>ttrain</td><td>train</td></tr><tr><td>tuck</td><td>truck</td></tr><tr><td>ttran</td><td>train</td></tr></table>		Search	Replace with	aiir	air	trainn	train	trick	truck	ttrain	train	tuck	truck	ttran	train
Search	Replace with																	
aiir	air																	
trainn	train																	
trick	truck																	
ttrain	train																	
tuck	truck																	
ttran	train																	

2. Data Transformation

DM Table	Image Creation Process (attach code)
Customer	<ul style="list-style-type: none"> ● Getting Input from all the sources of the company – PEC, TPCW and TPCE after cleaning them. ● After the cleaning process, we merged them giving each one of them division to identify from where the customers are coming from i.e. PEC, TPCW, TPCE. ● Transformations are sorted from PEC, TPCE and TPCW. ● We assigned the Surrogate keys to each unique records.
Product	<ul style="list-style-type: none"> ● Cleaning all the product files from PEC, TPCE and TPCW we assign the constants for each division. ● Merging them with the feature of Pentaho “Append Streams”, first with PEC and TPCW and Later TPCE. ● We didn’t split the supplier and kept inside the product dimension with name, address and state. ● Added Surrogate keys to unique each unique records in Product dimension.
Order_date	<ul style="list-style-type: none"> ● PEC product dates helped to calculate the Order_Year, Order_Month, Order_Week, Order_Quarter. ● Removed duplicates using unique rows. ● Added new fields in the Order_date dimension with fiscal_year, fiscal_Quarter, fiscal_Month and fiscal_week using javascript. ● Added surrogate keys for each unique dates
Sales_Date	<ul style="list-style-type: none"> ● Using calculator got the separate field for Sales_Year, Sales_Quarter, Sales_Month and Sales_Week for all the divisions (PEC, TPCW and TPCE) ● Same as order date used javascript to get the fiscal_year, fiscal_Quarter, fiscal_Month and fiscal_week. ● Added surrogate keys for each unique date.
Junk	<ul style="list-style-type: none"> ● PEC invoice only has orderMethod, PaymentMethod and shipMethod so extracted from the invoice. ● Added orderMethod, PaymentMethod and shipMethod to TPCE and TPCW with “N/A”. ● Sorted the rows with each unique rows to remove duplicates.

3. Table Population

DM Table	Table Population Process (attach code)
Customer	 <pre> graph LR A[Customer] --> B[Add sequence] B --> C[Table output] </pre>
Product	 <pre> graph LR A[CSV file input] --> B[Add sequence 2] B --> C[Table output 2] </pre>
Order_Date	 <pre> graph LR A[Junk] --> B[Add sequence 3] B --> C[Table output 3] </pre>
Junk	 <pre> graph LR A[Order_Date] --> B[Add sequence 4] B --> C[Table output 4] </pre>
Sale_Date	 <pre> graph LR A[file input 2] --> B[Add sequence 5] B --> C[Table output 5] </pre>

Sales_Fact	
	<ul style="list-style-type: none"> ● Merging each tables using Database Lookup with surrogate keys of Customer, Product, Sales_Date, Order_Date and Junk. ● Matching the records based on Natural Keys and Division in similarity. ● Group by the records for redundant records of the same SK for the population of the fact table.

VII. End User Applications

1. Queries

User Question/Reporting Need						
Report that is showing Top 5 Customers of Each Division with customer_Type, Product description, name and total sales amount. <ol style="list-style-type: none"> 1. PEC 2. TPCE 3. TPCW 						
SQL Code						
1. select c.Customer_Name, c.Customer_custtypeName, p.Product_Name, p.TypeDescription, c.Customer_division, SUM(Amount) as Sales from sales_fact AS sf join customer AS c using(Customer_SK) join product as p using(Product_SK) where c.Customer_division = "PEC" group by c.Customer_Name, C.Customer_state order by sum(Sales) DESC LIMIT 5;						
	Customer_Name	Customer_custtypeName	Product_Name	TypeDescription	Customer_division	Sales
►	Baxter May	Commercial	Enumerator Polishing Equipment	Polishing Equipment	PEC	83894157.00
	Raphael Allison	Commercial	Enumerator Polishing Equipment	Polishing Equipment	PEC	85805253.00
	The Product Company (East)	Commercial	Enumerator Polishing Equipment	Polishing Equipment	PEC	81431842.00
	Austin Ferrell	Commercial	Enumerator Polishing Equipment	Polishing Equipment	PEC	78682286.00
	Serrano	Commercial	Enumerator Polishing Equipment	Polishing Equipment	PEC	84512186.00
2. select c.Customer_Name, c.Customer_custtypeName, p.Product_Name, p.TypeDescription, c.Customer_division, SUM(Amount) as Sales from sales_fact AS sf join customer AS c using(Customer_SK) join product as p using(Product_SK) where c.Customer_division = "TPCE" group by c.Customer_Name, C.Customer_state order by sum(Sales) DESC LIMIT 5;						
	Customer_Name	Customer_custtypeName	Product_Name	TypeDescription	Customer_division	Sales
►	Room Plus Incorporated	Commercial	Flake Photo Equipment	Photo Equipment	TPCE	8826984.00
	Synaptic Pharmaceutical Corp	Commercial	Flake Photo Equipment	Photo Equipment	TPCE	9348858.00
	Railamerica Incorporated	Commercial	Flake Photo Equipment	Photo Equipment	TPCE	10042968.00
	Meridian Resources Corporati	Commercial	Flake Photo Equipment	Photo Equipment	TPCE	9848233.00
	Polymer Group Incorporated	Commercial	Flake Photo Equipment	Photo Equipment	TPCE	9296142.00
3. select c.Customer_Name, c.Customer_custtypeName, p.Product_Name, p.TypeDescription, c.Customer_division, SUM(Amount) as Sales from sales_fact AS sf join customer AS c using(Customer_SK) join product as p using(Product_SK)						

where c.Customer_division = "TPCW"
group by c.Customer_Name, C.Customer_state
order by sum(Sales) DESC LIMIT 5;

	Customer_Name	Customer_custtypeName	Product_Name	TypeDescription	Customer_division	Sales
▶	Santiago Processing	State/Local Gov	Enumerator Polishing Equipment	Polishing Equipment	TPCW	81938844.00
	India-Stuart	US Govt	Enumerator Polishing Equipment	Polishing Equipment	TPCW	84529830.00
	Ferdinand Supply	US Govt	Enumerator Polishing Equipment	Polishing Equipment	TPCW	84725388.00
	Austin Burns	Education	Enumerator Polishing Equipment	Polishing Equipment	TPCW	83812764.00
	Martinez Disposables	US Govt	Enumerator Polishing Equipment	Polishing Equipment	TPCW	85245945.00

Supporting Index(es)

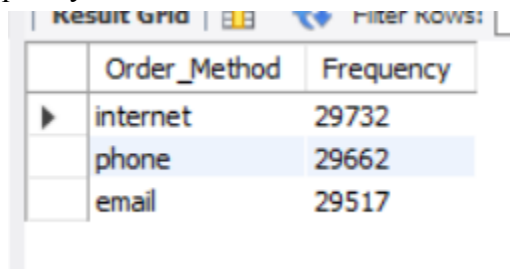
Customer_Name, Customer_custtypeName, Customer_Division(Customer), Product_Name, TypeDescription(Product), Sales(sales_fact)

User Question/Reporting Need

Get the report for most frequent method of ordering a product from PEC division

SQL Code

```
SELECT jk.Order_Method, count(*) 'Frequency'
FROM sales_fact sf JOIN junk jk USING (Junk_SK)
JOIN product p USING (Product_SK)
WHERE p.division = "PEC"
GROUP BY jk.Order_Method
ORDER BY Frequency DESC
```



Order_Method	Frequency
internet	29732
phone	29662
email	29517

Supporting Index(es)

Order_Method(Junk)

User Question/Reporting Need

Report where total percentage is calculated for each payment method i.e. COD, Charge and Cash from PEC.

SQL Code

```
SELECT a.pm 'Payment Method', a.totaleachinvoices 'Total of Each Invoices', b.total
'Total Invoices',
format(100*a.totaleachinvoices/b.total,1) "Percentage"
FROM (SELECT j.Payment_Method 'pm', COUNT(*) 'totaleachinvoices'
FROM sales_fact sf
```

```

JOIN junk j USING(Junk_SK)
GROUP BY j.Payment_Method) a,
(SELECT COUNT(*) 'total'
FROM sales_fact) b
LIMIT 3;

```

	Payment Method	Total of Each Invoices	Total Invoices	Percentage
▶	cod	29390	272694	10.8
	charge	30088	272694	11.0
	cash	29433	272694	10.8

Supporting Index(es)

Payment_Method(Junk)

User Question/Reporting Need

Report that defines the total sales with respect to Customer Type and respective Sales Year.

SQL Code

```

Select Customer_custtypeName, Sales_Year, Sales_Quarter, Sales_Month, SUM(Amount)
as Sales
FROM sales_fact as sf
JOIN sales_date as sd USING(Sales_Date_SK)
JOIN customer as cd USING (Customer_SK)
GROUP BY Customer_custtypeName, Sales_Year, Sales_Quarter
ORDER BY SUM(Amount) DESC;

```

	Customer_custtypeName	Sales_Year	Sales_Quarter	Sales_Month	Sales
▶	Commercial	2010	3	7	95557574.00
	Commercial	2009	4	11	95515801.00
	Commercial	2007	2	4	95075284.00
	Commercial	2006	3	7	93555908.00
	Commercial	2008	1	3	93549970.00
	Commercial	2005	3	7	93306943.00
	Commercial	2009	2	6	92931021.00
	Commercial	2007	4	10	92798994.00
	Commercial	2007	3	7	92752891.00
	Commercial	2010	1	2	92322390.00

Supporting Index(es)

**Customer_typeName(Customer), Sales_Year,
Sales_Quarter,Sales_Month(Sales_Date), Amount(sales_fact)**

2. A View

The view reports the sales report for each year, sales, cost and calculate the profit with respect to division to see who did good in all those years.

a) PEC

Query for the view:

```
CREATE VIEW PEC_gross_profit_year AS
SELECT view1.division, view1.Year, view1.Sales, view1.Costs,
(view1.Sales-view1.Costs) 'Gross Profit'
FROM
(SELECT p.division, s.Sales_Year 'Year', SUM(salesfact.Amount) 'Sales',
SUM(p.UnitCost * salesfact.Quantity) 'Costs'
FROM sales_fact salesfact JOIN product p USING (Product_SK)
JOIN sales_date s USING (Sales_Date_SK)
WHERE p.division = "PEC"
GROUP BY s.Sales_Year) view1;
```

Query:

```
select * from PEC_gross_profit_year;
```

	division	Year	Sales	Costs	Gross Profit
▶	PEC	2005	564083168.00	396825564.00	167257604.00
	PEC	2006	574128232.00	403812152.00	170316080.00
	PEC	2007	577889346.00	408156211.00	169733135.00
	PEC	2008	578299992.00	406754801.00	171545191.00
	PEC	2009	567165471.00	398699897.00	168465574.00
	PEC	2010	479426562.00	338008974.00	141417588.00

b) TPCE

Query for the view:

```
CREATE VIEW TPCE_gross_profit_year AS
SELECT view2.division, view2.Year, view2.Sales, view2.Costs,
(view2.Sales-view2.Costs) 'Gross Profit'
FROM
(SELECT p.division, s.Sales_Year 'Year', SUM(salesfact.Amount) 'Sales',
SUM(p.UnitCost * salesfact.Quantity) 'Costs'
FROM sales_fact salesfact JOIN product p USING (Product_SK)
JOIN sales_date s USING (Sales_Date_SK)
WHERE p.division = "TPCE"
GROUP BY s.Sales_Year) view2;
```

Query:

```
select * from TPCE_gross_profit_year;
```

	division	Year	Sales	Costs	Gross Profit
▶	TPCE	2005	56084582.00	45138300.00	10946282.00
	TPCE	2006	54218064.00	43639617.00	10578447.00
	TPCE	2007	56582255.00	45520308.00	11061947.00
	TPCE	2008	55532132.00	44706429.00	10825703.00
	TPCE	2009	55485156.00	44661869.00	10823287.00
	TPCE	2010	55591130.00	44734699.00	10856431.00
	TPCE	2011	46531217.00	37468424.00	9062793.00

c) **TPCW**

Query for the view:

```
CREATE VIEW TPCW_gross_profit_year AS
SELECT view3.division, view3.Year, view3.Sales, view3.Costs,
(view3.Sales-view3.Costs) 'Gross Profit'
FROM
(SELECT p.division, s.Sales_Year 'Year', SUM(salesfact.Amount) 'Sales',
SUM(p.UnitCost * salesfact.Quantity) 'Costs'
FROM sales_fact salesfact JOIN product p USING (Product_SK)
JOIN sales_date s USING (Sales_Date_SK)
WHERE p.division = "TPCW"
GROUP BY s.Sales_Year) view3;
```

Query:

```
select * from TPCW_gross_profit_year;
```

	division	Year	Sales	Costs	Gross Profit
▶	TPCW	2005	570042388.00	460438204.00	109604184.00
	TPCW	2006	573386105.00	463263359.00	110122746.00
	TPCW	2007	574476430.00	464351362.00	110125068.00
	TPCW	2008	576153429.00	465621735.00	110531694.00
	TPCW	2009	572925081.00	462855303.00	110069778.00
	TPCW	2010	468365619.00	378395222.00	89970397.00

d) **Overall all the divisions**

```
CREATE VIEW alldivisions_gross_profit_year AS
SELECT view4.division, view4.Year, view4.Sales, view4.Costs,
(view4.Sales-view4.Costs) 'Gross Profit'
FROM
(SELECT p.division, s.Sales_Year 'Year', SUM(salesfact.Amount) 'Sales',
SUM(p.UnitCost * salesfact.Quantity) 'Costs'
FROM sales_fact salesfact JOIN product p USING (Product_SK)
JOIN sales_date s USING (Sales_Date_SK)
GROUP BY s.Sales_Year) view4;
```

Query:

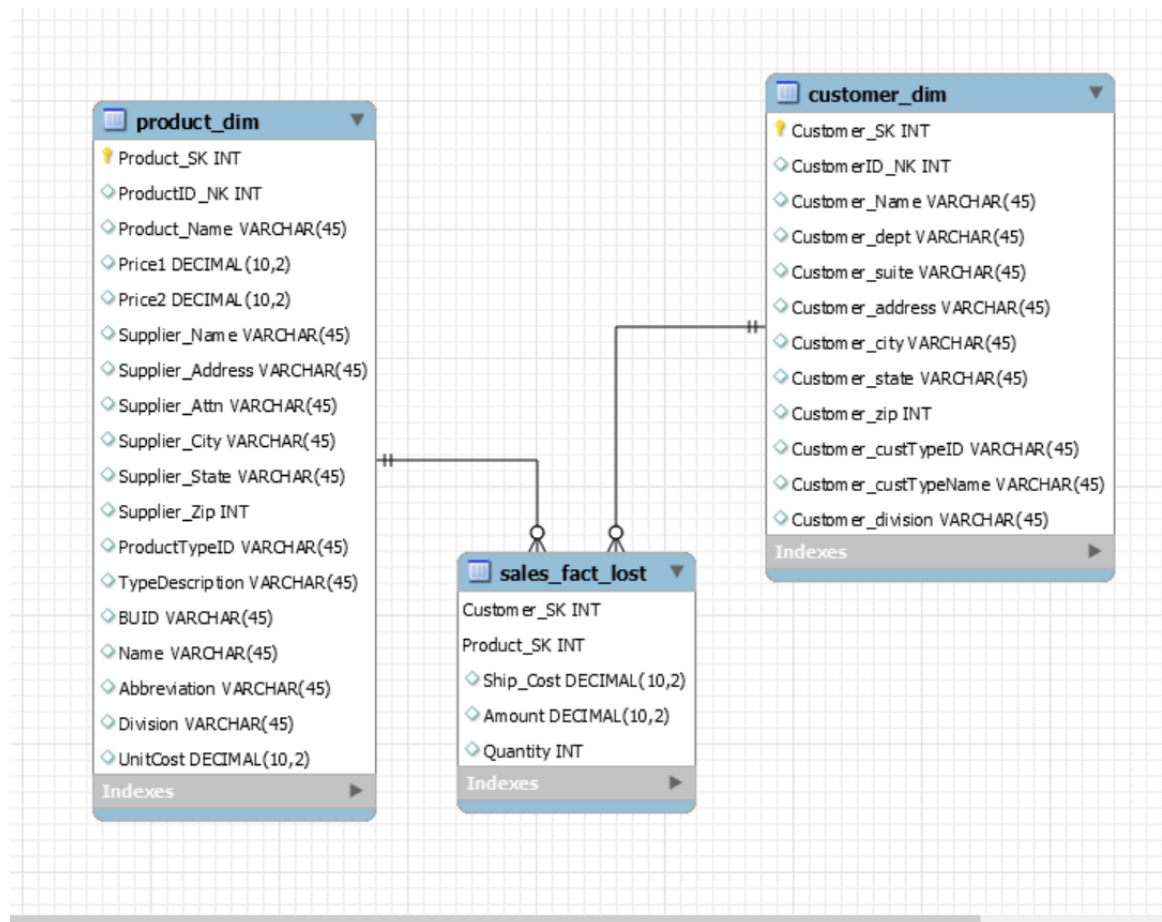
```
select * from alldivisons_gross_profit_year;
```

Year	Sales	Costs	Gross Profit
2005	1190210138.00	902402068.00	287808070.00
2006	1201732401.00	910715128.00	291017273.00
2007	1208948031.00	918027881.00	290920150.00
2008	1209985553.00	917082965.00	292902588.00
2009	1195575708.00	906217069.00	289358639.00
2010	1003383311.00	761138895.00	242244416.00
2011	46531217.00	37468424.00	9062793.00

3. Aggregated Mata Marts

- **Lost Aggregated Data Mart**

ERD:



Aggregation method: Lost Dimension

Use Case: In this use case we kept Customer and Product dimension. In lost aggregated dimension, one or more dimensions are removed, in this use case we removed sales date, Order date, Junk Dimension. The remaining dimensions can be used in finding the most popular product and the top customers.

Creation of Tables: For this kind of aggregation we created the tables using MySQL Workbench and used “Forward Engineering” to further create the data mart

File: LostAggregation.mwb

Output of Fact Table:

```
use lostaggregated;
select * from sales_fact_lost;
```

Customer_SK	Product_SK	Ship_Cost	Amount	Quantity
4321	4321	10.00	3280.00	20
4321	4322	8.00	34353.00	99
4321	4323	19.00	47337.00	93
4321	4324	9.00	82992.00	182
4321	4325	12.00	41296.00	178
4321	4326	4.00	53110.00	94
4321	4327	12.00	46190.00	155
4321	4328	18.00	57222.00	198
4321	4329	17.00	55900.00	100
4321	4330	11.00	61090.00	149
4321	4331	12.00	22100.00	65
4321	4332	20.00	92660.00	164
4321	4333	2.00	4290.00	10
4321	4334	21.00	17536.00	64
4321	4335	1.00	41132.00	182
4321	4336	13.00	8400.00	42
4321	4337	16.00	6448.00	26
4321	4338	22.00	24924.00	124
4321	4339	31.00	54972.00	108
4321	4340	7.00	36195.00	127
4321	4341	7.00	34056.00	172
4321	4342	16.00	2400.00	20
4321	4343	3.00	33432.00	168
4321	4344	12.00	65875.00	155

Creation and Inserting to Aggregates data mart:

Customer Dimension:

```
#inserting data into customer lost dim
Insert into lostaggregated.customer_dim(Customer_SK, CustomerID_NK, Customer_Name, Customer_dept, Customer_suite,
Customer_address, Customer_city, Customer_state, Customer_zip, Customer_custTypeID, Customer_custtypeName,
Customer_division)
select Customer_SK, CustomerID_NK, Customer_Name, Customer_dept, Customer_suite,
Customer_address, Customer_city, Customer_state, Customer_zip, Customer_custTypeID, Customer_custtypeName,
Customer_division from salesorder_5_before.customer;
```

Product Dimension:

```
#inserting data into product lost dimension
Insert into lostaggregated.product_dim(Product_SK, ProductID_NK, Product_Name, Price1, Price2,
supplier_Name, supplier_Address, supplier_Attn,supplier_City,supplier_State,supplier_Zip, productTypeID,
TypeDescription,BUID, Name, Abbreviation, Division,UnitCost)
select Product_SK, ProductID_NK, Product_Name, Price1, Price2,
supplier_Name, supplier_Address, supplier_Attn,supplier_City,supplier_State,supplier_Zip, productTypeID,
TypeDescription,BUID, Name, Abbreviation, division,UnitCost from salesorder_5_before.product;
```


Lost Aggregation Fact Table:

```
#insert into sales fact lost dim
Insert ignore into lostaggregated.sales_fact_lost (Customer_SK, Product_SK, Ship_Cost, Amount, Quantity)
select Customer_SK, Product_SK, Ship_Cost, Amount, Quantity from salesorder_5_before.sales_fact
group by Customer_SK,Product_SK;
```

USER QUERIES:

```
#QUERY1:
use lostaggregated;
select cust.customer_Name as 'Customer Name', p.typeDescription as
'Product Description',sum(fact.quantity) as 'Number of Products',
sum(fact.Amount) as 'total amount'
from sales_fact_lost fact
inner join Customer_dim cust on cust.Customer_SK=fact.Customer_SK
inner join product_dim p on p.product_SK=fact.Product_SK
group by cust.Customer_Name
order by sum(fact.Amount) DESC limit 10;
```

Customer Name	Product Description	Number of Products	total amount
Firstfed America Bancorp Inc	Polishing Equipment	16762	6022102.00
Blevins	Polishing Equipment	16743	6011286.00
Zena Machines	Polishing Equipment	16458	5964214.00
Hop Adams	Polishing Equipment	16013	5816072.00
Ronan French	Polishing Equipment	15844	5730843.00
Emerson Electric Company	Polishing Equipment	15599	5572643.00
Mullins Incorporated	Polishing Equipment	15559	5517444.00
Austin Ferrell	Polishing Equipment	15389	5500878.00
Cross	Polishing Equipment	14751	5381095.00
Beverly Equipment	Polishing Equipment	15149	5215236.00

```
#QUERY2:
select p.Product_Name as 'Product Name', sum(f.ship_Cost) as
'Shipping cost', sum(f.Amount) as 'Total sales'
from sales_fact_lost f
inner join product_dim p on p.product_SK=f.product_SK
group by p.product_Name
order by f.ship_Cost DESC limit 10;
```

Product Name	Shipping cost	Total sales
Bellowing Polishing Equipment	543.00	4199233.00
Habitually Manufacturing Equipment	489.00	4666517.00
Commendation Fillers	545.00	3894048.00
Allis Polishing Equipment	534.00	1640805.00
Travelings Photo Chemicals	507.00	2355201.00
Chromium Photo Equipment	551.00	4826453.00
Tailor Jacks	465.00	4477477.00
Concentrators Polishing Equipment	475.00	5517311.00
Sulkiness Covers	456.00	3145048.00
Birthdays Manufacturing Equipment	477.00	1722266.00

- **Shrunken Aggregation data :**

ERD:

Aggregation Method: Shrunken Dimension

Use Case: In this use case we have used the shrunk dimension and made the few dimension lost

Creation of Tables: For this kind of aggregation we created the tables using MySQL Workbench and used “Forward Engineering” to further create the data mart

File: shrunkenAggregate.mwb

Shruken Dimension & Grain: Monthly

```
#inserting data into customer shrunken dim
Insert into shrunkenaggregaion.customer_dim(Customer_SK, CustomerID_NK,
Customer_Name, Customer_dept, Customer_suite,
Customer_address, Customer_city, Customer_state, Customer_zip,
Customer_custTypeID, Customer_custTypeName,
Customer_division)
select Customer_SK, CustomerID_NK, Customer_Name, Customer_dept,
Customer_suite,
Customer_address, Customer_city, Customer_state,
Cutomer_zip, Customer_custttTypeID, Customer_custttTypeName,
Customer_division from salesorder_5_before.customer;
```

```
#inserting data into product shrunken dimension
Insert into shrunkenaggregaion.product_dim(Product_SK, ProductID_NK, Product_Name, Price1, Price2,
supplier_Name, supplier_Address, supplier_Attn,supplier_City,supplier_State,supplier_Zip, productTypeID,
TypeDescription,BUID, Name, Abbreviation, Division,UnitCost)
select Product_SK, ProductID_NK, Product_Name, Price1, Price2,
supplier_Name, supplier_Address, supplier_Attn,supplier_City,supplier_State,supplier_Zip, productTypeID,
TypeDescription,BUID, Name, Abbreviation, division,UnitCost from salesorder_5_before.product;

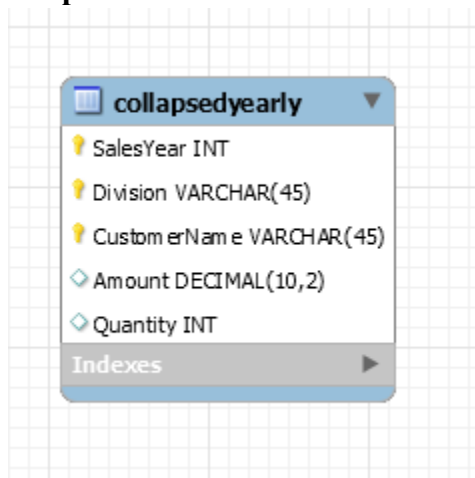
#insert into sales shrunken date dim
insert into shrunkenaggregaion.salesdateshrunken_dim(sales_Year,sales_Quarter,
sales_Month,sales_fiscal_Year,sales_fiscal_Quarter,sales_fiscal_Month)
select Sales_Year, Sales_Quarter, Sales_Month,Sales_Fiscal_Year, Sales_Fiscal_Quarter,
Sales_Fiscal_Month from salesorder_5_before.sales_date;
```

- **Collapsed Aggregation**

Aggregation Method: Collapsed Aggregation Dimension

Use Case: In collapsed aggregated Data mart we removed the surrogate key and use the attributes of useful dimensions and combines them into fact table. Here we used sales_year from sales_date dimension, Division and CustomerName from Customer dimension.

Sample output:



```
USE COLLAPSEDAGGREGATE;
select * from collapsedyearly;
```

	SalesYear	Division	CustomerName	Amount	Quantity
►	2005	PEC	Ann Lee	15056088.00	41032
	2005	PEC	Atkins	14536400.00	40087
	2005	PEC	Austin Ferrell	12883264.00	36563
	2005	PEC	Baxter May	14898842.00	41851
	2005	PEC	Beverly Equipment	13822151.00	38488
	2005	PEC	Blevins	14191868.00	40604
	2005	PEC	Cain	13865988.00	39284
	2005	PEC	Clare Baird	15454228.00	42082
	2005	PEC	Colton Maldonado	13539753.00	37349
	2005	PEC	Cross	13785265.00	39988
	2005	PEC	Dakota Mills	15609526.00	41717
	2005	PEC	Emerson Electric Compa...	12304889.00	33974
	2005	PEC	Ferengi Treasures	13533823.00	37978
	2005	PEC	Firstfed America Bancor...	16000796.00	43963
	2005	PEC	Gemma Castro	14204247.00	40110
	2005	PEC	Googol	14118557.00	40551
	2005	PEC	Hammett Farley	13428399.00	36691
	2005	PEC	Haynes	15296075.00	43792
	2005	PEC	Hop Adams	13840711.00	38770
	2005	PEC	Knight	15289183.00	43129
	2005	PEC	Knox Reid	13037342.00	36661
	2005	PEC	Kuame Barnes	15138579.00	41446
	2005	PEC	Mallory Lynch	13945788.00	39402
	2005	PEC	Martin Donaldson	13510697.00	37773
	2005	PEC	Martinez Disposables	13583844.00	38926
	2005	PEC	Maya Brewer	14655397.00	40165
	2005	PEC	Melvin House	12678743.00	35365
	2005	PEC	Mullins Incorporated	13291027.00	37694
	2005	PEC	Pewter Gym	14876256.00	42246
	2005	PEC	Raphael Allison	15027957.00	41144
	2005	PEC	Ronan French	12690689.00	35467

Populating Collapsed Aggregate Data Mart:

```

INSERT INTO collapsedyearly(salesYear,Division,CustomerName,Amount,Quantity)
select sales_Year, CASE when Customer_division="PEC" then "PEC"
when Customer_division="TPCE" then "TPC EAST"
when Customer_division="TPCW" then "TPC WEST"
END as "DivisionName", Customer_Name, sum(Amount),sum(Quantity)
from salesorder_5_before.sales_fact f join salesorder_5_before.sales_date sd
using (sales_Date_SK) join salesorder_5_before.Customer c on f.Customer_SK=c.Customer_SK
group by sd.Sales_Year,c.Customer_Division,c.Customer_Name;

```

USE QUERIES:

#Query 1:

```
SELECT *FROM ( SELECT Division
                ,CustomerName
                ,sum(Amount) `Total Sales` ,RANK() OVER (
                    PARTITION BY Division ORDER BY sum(Amount) DESC
                ) `Ranking`
            FROM collapsedyearly y
            GROUP BY Division
                ,CustomerName
        ) q1
WHERE Ranking <= 5;
```

	Division	CustomerName	Total Sales	Ranking
▶	PEC	Clare Baird	87632356.00	1
	PEC	PEC Center Gym	87073644.00	2
	PEC	Maya Brewer	86935577.00	3
	PEC	Kuame Barnes	86340099.00	4
	PEC	Dakota Mills	86046793.00	5
	TPC EAST	Gte Corporation	10989815.00	1
	TPC EAST	Greenman Technologies Incorp	10851712.00	2
	TPC EAST	Camera Platforms Internation	10526697.00	3
	TPC EAST	Helpmate Robotics Incorporat	10499580.00	4
	TPC EAST	Seattle Filmworks Incorporat	10336360.00	5
	TPC WEST	Price Rivers	87867821.00	1
	TPC WEST	Ashton Washington	87257163.00	2
	TPC WEST	Mullins Incorporated	86490465.00	3
	TPC WEST	Pulitzer Publishing Company	85959640.00	4
	TPC WEST	Byron Chemicals	85695086.00	5

SalesYear	Division	Sales by Year	Yearly Sales as per Division	Percentage OF Yearly Division Sales
2005	PEC	1190210138.00	564083168.00	47.39
2005	TPC EAST	1190210138.00	56084582.00	4.71
2005	TPC WEST	1190210138.00	570042388.00	47.89
2006	PEC	1201732401.00	574128232.00	47.78
2006	TPC EAST	1201732401.00	54218064.00	4.51
2006	TPC WEST	1201732401.00	573386105.00	47.71
2007	PEC	1208948031.00	577889346.00	47.80
2007	TPC EAST	1208948031.00	56582255.00	4.68
2007	TPC WEST	1208948031.00	574476430.00	47.52
2008	PEC	1209985553.00	578299992.00	47.79
2008	TPC EAST	1209985553.00	55532132.00	4.59
2008	TPC WEST	1209985553.00	576153429.00	47.62
2009	PEC	1195575708.00	567165471.00	47.44
2009	TPC EAST	1195575708.00	55485156.00	4.64
2009	TPC WEST	1195575708.00	572925081.00	47.92
2010	PEC	1003383311.00	479426562.00	47.78
2010	TPC EAST	1003383311.00	55591130.00	5.54
2010	TPC WEST	1003383311.00	468365619.00	46.68
2011	TPC EAST	46531217.00	46531217.00	100.00

VIII. Handling Slowly Changing Dimensions (SCD)

- Created Sample Slowly Changing Dimension records in Product and Customer dimension

Source File : SCD_Data/SCD1_Data

Rows : 30

product_id	product_name	price1	price2	unitcost	productty	buid	name	abbrev	division
64	Berwick Fillers-TPCE-Version	584	488	463	12	D	Miscellaneous	Misc	TPCE
64	Berwick Fillers -V2	583	487	463	12	D	Miscellaneous	Misc	TPCW
52	Identical Freezing Chemicals	578	465	447	11	C	Hard Chemicals	Chemicals	TPCW
52	Identical Freezing Chemicals	588	466	447	11	C	Chemicals	Chemical	TPCE
97	Stickiest Paper Supplies	578	503	468	6	B	Disposable Supplies	Supply	TPCE
97	Stickiest Paper Supplies	579	502	467	6	B	Disposable Supplies	Supplies	TPCW
49	Visage Flushing Chemicals	568	510	467	10	C	Chemicals	Chemical	TPCE
49	Visage Flushing Chemicals	567	517	466	10	C	Chemicals	Chemicals	TPCW
94	Beliefs Freezing Chemicals	577	444	455	11	C	Chemicals	Chemical	TPCE
94	Beliefs Freezing Chemicals	566	383	454	11	C	Chemicals	Chemicals	TPCW
35	Chromium Photo Equipment	566	430	425	1	A	Processing Equipment	Equipment	TPCE
21	Denigrating Polishing Equipment	566	494	427	3	A	Processing Equipment	Equipment	TPCE
6	Denigrating Polishing Equipment-PEC	565	454	402	3	A	Processing Equipment	Equipment	PEC
12	Chromium Photo Equipment	535	457	326	1	A	Processing Equipment-HQ	Equipment	PEC
21	Denigrating Polishing Equipment	585	454	427	3	A	Processing Equipment	Equipment	TPCW
35	Chromium Photo Equipment	585	477	425	1	A	Processing Equipment	Equipment	TPCW
10	Defeated Tray Supplies	590	477	448	8	B	Chemicals	Supply	TPCE
42	Carelessly Freezing Chemicals	560	478	426	11	C	Chemicals	Chemical	TPCE
9	Defeated Tray Supplies	559	476	448	8	B	Disposable Supplies	Supplies	PEC
10	Defeated Tray Supplies -TPCW	559	476	448	8	B	Disposable Supplies	Supplies	TPCW
42	Carelessly Freezing Chemicals	559	467	425	11	C	Hard Chemicals	Chemicals	TPCW
28	Wonderingly Covers	547	467	453	13	D	Miscellaneous	Misc	TPCW
28	Wonderingly Covers-Version2	547	467	453	13	D	Miscellaneous	Misc	TPCE
55	Bumblers Plastic Supplies	545	466	447	7	B	Disposable Supplies	Supplies	TPCW
86	Soya Cleaning Supplies	406	345	345	5	B	Disposable Supplies	Supplies	TPCW
87	Suing Manufacturing Equipment	482	426	401	2	A	Processing Equipment	Equipment	TPCW
21	Denigrating Polish Equipment	566	454	427	3	A	Processing Equipment	Equipment	TPCE
93	Bellowing Polishing Equipment	503	451	447	3	A	Processing Equipment-HQ	Equipment	TPCE
63	Weeks Polishing Equipment	199	174	169	3	A	Processing Equipment-HQ	Equipment	TPCE
68	Loaves Polishing Equipment	426	361	354	3	A	Processing Equipment-HQ	Equipment	TPCE

- **Highlighted Parts:**

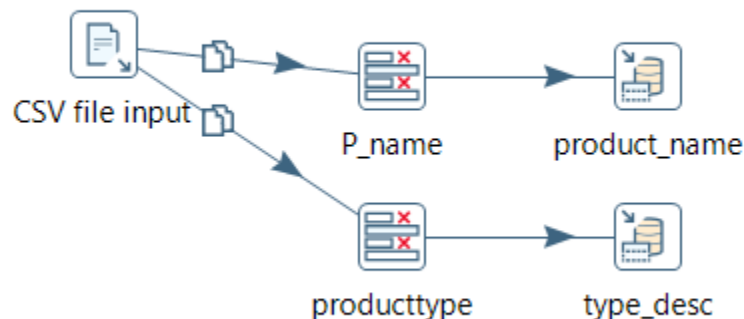
Product_Name (Yellow) : SCD1 Implementation

Name(Orange): SCD1 Implementation

- **SCD Type 1:**

The implantation of the type1 on columns Product_name and name from Product table as history is not necessary to be kept as it is and it can change overtime.

We implemented SCD1 using the Pentaho, the file is SCD1.ktr, where we used the feature *Output->Insert/Update to implement type1*



- **Output:**

product_SK	ProductID_NK	division	Product_Name	p
4550	29	TPCE	Bluest Fillers	50
4551	64	TPCE	Berwick Fillers-TPCE-Version	58
4552	74	TPCE	Flanker Fillers	27
4553	5	TPCE	Automobiles Fillers	28

SCD Type 2

Source : SCD1_Data

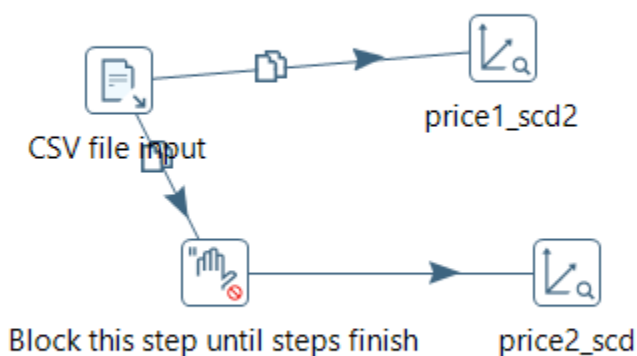
Rows:30

	A	B	C	D	E	F	G	H	
1	product_id	product_name	price1	price2	unitcost	productty	buid	name	abbrev
2	64	Berwick Fillers-TPCE-Version	584	488	463	12	D	Miscellaneous	Misc
3	64	Berwick Fillers -V2	583	487	463	12	D	Miscellaneous	Misc
4	52	Identical Freezing Chemicals	578	465	447	11	C	Hard Chemicals	Chemicals
5	52	Identical Freezing Chemicals	588	466	447	11	C	Chemicals	Chemical
6	97	Stickiest Paper Supplies	578	503	468	6	B	Disposable Supplies	Supply
7	97	Stickiest Paper Supplies	579	502	467	6	B	Disposable Supplies	Supplies
8	49	Visage Flushing Chemicals	568	510	467	10	C	Chemicals	Chemical
9	49	Visage Flushing Chemicals	567	517	466	10	C	Chemicals	Chemicals
10	94	Beliefs Freezing Chemicals	577	444	455	11	C	Chemicals	Chemical
11	94	Beliefs Freezing Chemicals	566	383	454	11	C	Chemicals	Chemicals
12	35	Chromium Photo Equipment	566	430	425	1	A	Processing Equipment	Equipment
13	21	Denigrating Polishing Equipment	566	494	427	3	A	Processing Equipment	Equipment
14	6	Denigrating Polishing Equipment-PEC	565	454	402	3	A	Processing Equipment	Equipment
15	12	Chromium Photo Equipment	535	457	326	1	A	Processing Equipment-HQ	Equipment
16	21	Denigrating Polishing Equipment	585	454	427	3	A	Processing Equipment	Equipment
17	35	Chromium Photo Equipment	585	477	425	1	A	Processing Equipment	Equipment
18	10	Defeated Tray Supplies	590	477	448	8	B	Chemicals	Supply
19	42	Carelessly Freezing Chemicals	560	478	426	11	C	Chemicals	Chemical
20	9	Defeated Tray Supplies	559	476	448	8	B	Disposable Supplies	Supplies
21	10	Defeated Tray Supplies -TPCW	559	476	448	8	B	Disposable Supplies	Supplies
22	42	Carelessly Freezing Chemicals	559	467	425	11	C	Hard Chemicals	Chemicals
23	28	Wonderingly Covers	547	467	453	13	D	Miscellaneous	Misc
24	28	Wonderingly Covers-Version2	547	467	453	13	D	Miscellaneous	Misc
25	55	Bumblers Plastic Supplies	545	466	447	7	B	Disposable Supplies	Supplies
26	86	Soya Cleaning Supplies	406	345	345	5	B	Disposable Supplies	Supplies
27	87	Suing Manufacturing Equipment	482	426	401	2	A	Processing Equipment	Equipment
28	21	Denigrating Polish Equipment	566	454	427	3	A	Processing Equipment	Equipment
29	93	Bellowing Polishing Equipment	503	451	447	3	A	Processing Equipment-HQ	Equipment
30	63	Weeks Polishing Equipment	199	174	169	3	A	Processing Equipment-HQ	Equipment
31	68	Loaves Polishing Equipment	426	361	354	3	A	Processing Equipment-HQ	Equipment
32									

- **Highlighted Part:**

- Price1 (Blue) : SCD type 2
- Price2 (Green) : SCD type 2

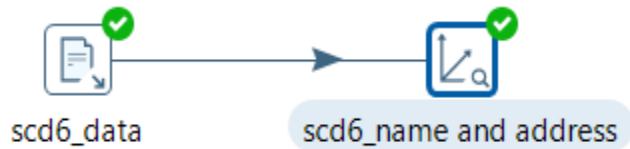
Implementing the price1 and price2 to keep the track of the changes in the prices with their version numbers. Implemented using the *Dimension lookup/Update* for Type2.



act_SK	ProductID_NK	division	Product_Name	price1	price1_Version	price1_Valid_From	price1_Valid_Through	price2	p
	10	TPCW	Defeated Tray Supplies -TPCW	559	1	1900-01-01	2200-01-01	476	NU
	42	TPCW	Carelessly Freezing Chemicals	559	1	1900-01-01	2200-01-01	467	NU
	28	TPCW	Wonderingly Covers	547	1	1900-01-01	2200-01-01	467	NU
	28	TPCE	Wonderingly Covers-Version2	547	1	1900-01-01	2200-01-01	467	NU
	55	TPCW	Bumblers Plastic Supplies	545	1	1900-01-01	2200-01-01	466	NU
	86	TPCW	Soya Cleaning Supplies	406	1	1900-01-01	2200-01-01	345	NU
	87	TPCW	Suing Manufacturing Equipment	482	1	1900-01-01	2200-01-01	426	NU

SCD Type 6

We implemented SCD Type-6 on the columns name and address from the customer_dimension. We did so because the history of these attributes is important and should be maintained. Also, Type 6 SCD adds a current field which helps us to determine the current record and the date it is valid till. We implemented SCD-6 using Pentaho transformation (scd6.ktr). In the transformation, we used the step *Data Warehouse → Dimension lookup/ update* to implement SCD Type-6.



	Customer_SK	CustomerID_NK	Customer_name	name_current	address_current	Customer_address	row_Valid_From	row_Valid_Through
▶	4336	7	Martin Donaldson	Rupa Associates - 3	270 At Road	2704 At Road	1900-01-01	2020-11-26
	4385	7	Hop Adams	NULL	NULL	704 Nisl Road	1900-01-01	2222-01-01
	4412	7	Fedders Corporation	NULL	NULL	3311 Blatantly Circle	1900-01-01	2222-01-01
	4468	7	Rupa Associates	Rupa Associates - 3	270 At Road	2704 At Road	2020-11-26	2020-11-29
	4474	7	Rupa Associates - 2	Rupa Associates - 3	270 At Road	24 At Road	2020-11-29	2020-11-29
	4475	7	Rupa Associates - 3	Rupa Associates - 3	270 At Road	270 At Road	2020-11-29	2200-01-01
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

IX. Many-to-Many (N-M) Relationship Implementation Option

A Many-to-Many relationship is defined as a relationship between a parent and the child tables in a database. A parent row consists of multiple child rows in the other table. In a relational database design, many-to-many relationships are not permitted due to the following issues:

- a. Causes data redundancy
- b. Difficulty in inserting, updating, and deleting the data.

In many real-world applications N:M relationships are often used and normalizing the fact table is not an option.

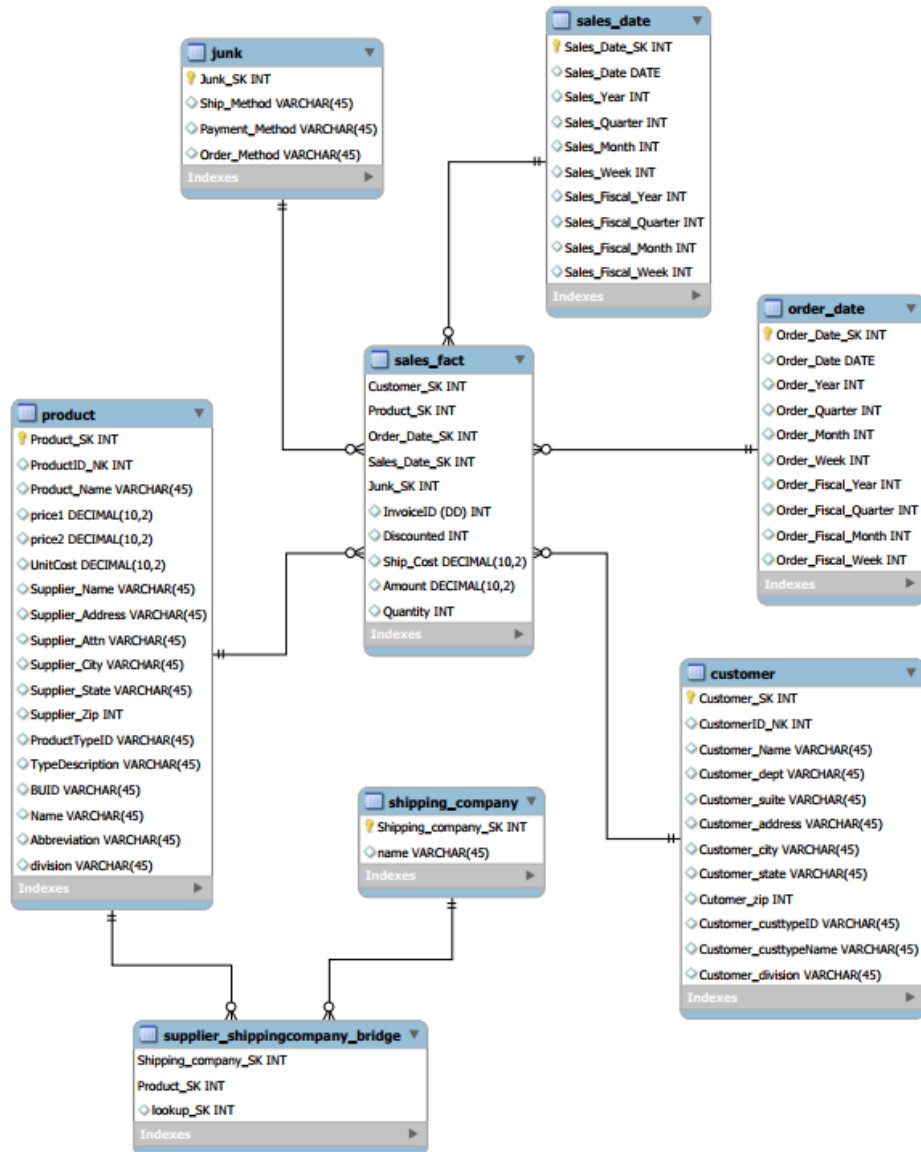
Several approaches exist in dealing Many-to-Many relationships such as:

- a. Joining or Bridging tables.
- b. Lowering the grain of the fact table
- c. De-normalizing the Dimension Table by Positional-Flag Attributes
- d. De-normalizing the Dimension Table by Non-Positional attributes & a Concatenated Field

For our model we have decided to implement the Bridge method. Here a look-up table is created which consists of the surrogate keys for both the Product and Shipping company. We intent to find details of the supplier which in our case is present in the Product dimension. Here, many-to-many relationship can occur since there is a possibility of suppliers have multiple shipping contracts. Likewise, the shipping company can have multiple suppliers. The Kimball's method is a better approach since it minimizes redundancy. In this case the issue lies in assigning weights allotted to a contractor. The bridge method would reduce redundancy by making sure these weights do not exceed 1.

References:

Rowen, W., Song, I. Y., Medsker, C., & Ewen, E. (2001). An analysis of many-to-many relationships between fact and dimension tables in dimensional modeling. In *International Workshop on Design and Management of Data Warehouses (DMDW 2001)*, Interlaken Switzerland (pp. 1-13).



X. Appendix (Fix Lab #3 Problems)

Rationale for Final Schema Design: The designing of the final schema design, we merged the Supplier in the Product itself to keep the track the whole product values as the same. The reason behind it, because the two divisions can sell the same product, but prices and supplier can be different, so its easy to keep intact to track down the information.