***The Product Company***

**~ Final Data Mart Development Report ~**

Team # 3

Team Members

Darshan Shah

Evlin Steve

Purva Redkar

Date

12/8/2017

ISTE-DW Data Warehousing

~ Table of Contents ~

[I. Data Mart Design Definition 2](#_30j0zll)

[1. Universe of Discourse 2](#_1fob9te)

[2. Information Package 2](#_3znysh7)

[II. Dimensional Model 3](#_2et92p0)

[III. Data Staging: ETL – Data Extract File Definitions 4](#_tyjcwt)

[IV. Data Staging: ETL – Source-to-Target Mappings 5](#_3dy6vkm)

[V. SQL Code – Tables & Constraints 6](#_1t3h5sf)

[VI. Data Staging Activities - ETL 7](#_4d34og8)

[1. Data Cleansing 7](#_2s8eyo1)

[2. Data Transformation 7](#_17dp8vu)

[3. Table Population 7](#_3rdcrjn)

[VII. End User Applications 8](#_26in1rg)

[1. SQL99 Queries 8](#_lnxbz9)

[2. A Views 8](#_35nkun2)

3. Aggregated Data Marts …….……………………………………………........9

VIII. Handling Slowly Changing Dimensions (SCD)……..…………………………. 10

IX. N-M Implementation Option………………………………………………….…11

X Appendix (Fix Lab #3)…………………………………………………………..12

# I. Data Mart Design Definition

## 1. Universe of Discourse

|  |
| --- |
| This Data Mart is for Financial Sales analysis of PEC, TPCW and TPCE company to investigate their financial performance and optimisation for year , quarter, month ,week and daily basis. |

## 2. Information Package

Process Name: Financial analysis and control of PEC, TPCW, TPCE

Grain: Individual Sales for each product

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product | Customer | Supplier | OrderDate | SalesDate | Method |
| Product\_sk | Customersk | SupplierID\_sk | OrderdateID\_sk | SalesdateID\_sk | MethodID\_sk |
| Name | CustomerID | SupplierID\_NK | OrderDate\_Day | Salesdat\_Day | Paymentmethod |
| ABBREV | name | Suppliername | OrderDate\_Quater | Salesdat\_Quarter | ShipMethod |
| prodid | addr1 | Address1 | QrderDate\_week | Salesdat\_week | Ordermethod |
| prodDescription | addr2 | Address2 | OrderDate\_Year | Salesdat\_Year |  |
| price1 | city | City | OrderDate\_month | Salesdat\_month |  |
| price2 | state | State | OrderDate\_nk | Salesdate\_nk |  |
| UnitCost | zip | zip | OrderFiscal\_Date | SalesFiscal\_Date |  |
| producttypeID | custtypeID |  | OrderFiscal\_Week | SalesFiscal\_Week |  |
| TypeDescription | typename |  | Calender\_Quarter | SalesFiscal\_Quarter |  |
| Source | Source |  | OrderFiscal\_Month | SalesFiscal\_Month |  |
| BUID |  |  | OrderFiscal\_Year | SalesFiscal\_Year |  |

Facts: Number of days , Sales Amount, Quantity CostAmount, Discount

3. Entity Definitions

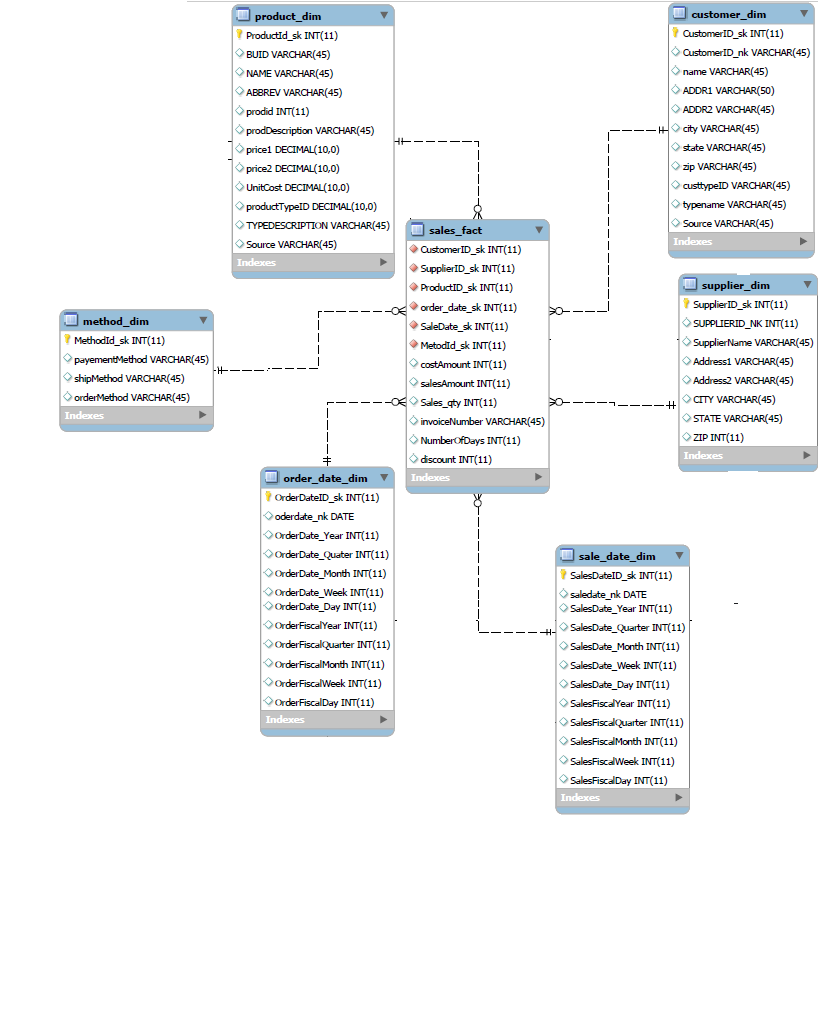
# 

|  |  |
| --- | --- |
| **Entity** | **Entity Definition** (*genus differentia*) |
| Product\_dim | Product Entity defines product details of the products sold by TPCE,TPCW and PEC divisions of the company.  1)Product\_sk:It represents the surrogate key of product dimension table  2)BUID: It represents the unique ID of a business Unit  3)Name: It represents the name of the products .  4) Abbrev: It represents the abbreviations of Business Unit  5)prodid: It is the natural key in product dimension which identifies products from each source.  6)prodDescription: It gives description of each product.  7)price 1: It is the original price of the product  8)price 2 :it is the discounted price of the product.  9)unitcost: It represents unit cost per product.  10) ProductTypeID: It represents the typeID of a particular product type/  11) Typedescription: It represents the description of type of producttype  12)Source:It denotes which source the product belongs to. |
| Customer\_dim | Customer entity is defined as: Customers who are for the TPC-E, TPC-W and PEC  1) CustomerID\_SK: It represents the surrogate key of customer dimension table  2) CustomerID: It represents the unique ID of the each customer which is primary key and natural key in OLTP database and dimension table respectively  3) Name: It represents the name of the customers  4) Addr1: It represents the first address line of customer  5) Addr2: It represents second address line  6) City: It is the city of customer  7) State: It represents the state of customer  8) zip: It is the zip number of the customer  9) custTypeID: It represents the typeID of customer type  10) typename: It represents the name of the customer type according to typeid  11)Source:It represents that customer belongs to which division(TPCE,TPCW,PEC) |
| Supplier | Supplier dimension is defined as the suppliers of the products for 3 divisions.  1)SupplierID\_sk: It represents the surrogate key of supplier dimension table  2)SupplierId\_nk: It represents the unique ID of the each customer which is primary key and natural key in OLTP database and dimension table respectively  3)Supplier Name: It represents the name of the supplier  4) Address1: It represents the first address line of supplier  5) Addr2: It represents second address line  6) City: It is the city of supplier  7) State: It represents the state of supplier  8) zip: It is the zip number of the supplier |
| SalesDate | It represents sales date for each transactions,  SalesdateID\_sk: It represents the surrogate key of Sales dimension table  SalesdateID\_nk:It represents the unique ID of the each transaction which is primary key and natural key in OLTP database and dimension table respectively  Salesdate\_year: It denotes sales Year according to Calender date  Salesdate\_quarter:It denotes sales quarter according to Calender date  Salesdate\_month:It denotes sales month according to Calender date  Salesdate\_week:It denotes sales week according to Calender date  Salesdate\_day:It denotes sales day according to Calender date  SalesFiscal\_year: It denotes sales year according to fiscal date  SalesFiscal\_quarter: It denotes sales quarter according to fiscal date  SalesFiscal\_month: It denotes sales month according to fiscal date  SalesFiscal\_week: It denotes sales week according to fiscal date  SalesFiscal\_Day: It denotes sales date according to fiscal date |
| OrderDate | It represents order date for each transactions,  orderdateID\_sk: It represents the surrogate key of order dimension table  orderdateID\_nk:It represents the unique ID of the each transaction which is primary key and natural key in OLTP database and dimension table respectively  orderdate\_year: It denotes order Year according to Calender date  orderdate\_quarter:It denotes order quarter according to Calender date  orderdate\_month:It denotes order month according to Calender date  orderdate\_week:It denotes order week according to Calender date  orderdate\_day:It denotes order day according to Calender date  orderFiscal\_year: It denotes order year according to fiscal date  orderFiscal\_quarter: It denotes order quarter according to fiscal date  orderFiscal\_month: It denotes order month according to fiscal date  orderFiscal\_week: It denotes order week according to fiscal date  orderFiscal\_Day: It denotes orderdate according to fiscal date |
| Method | It denotes different methods  Methodid\_sk:It represents the surrogate key of method dimension table  Ordermethod: It denotes the method in which order was placed like internet,phone etc  Paymentmethod:It denotes the method in which payment was done like card,cheque etc  Shipmethod:It denotes the method in which order was shipped like air,train etc |
| Sales\_fact | This is the fact table in the star schema. It consists of  1)ProductID\_SK: It is primary key in Product dimension table and in fact table it is foriegn key.We can use it to get details about product dimension  2) CustomerID\_SK: It is primary key in Customer dimension table and in fact table it is foriegn key We can use it to get details about customer dimension.  3) SupplierID\_SK: It is primary key in supplier dimension table and in fact tableit is foriegn key.We can use it to get details about supplier dimension .  4) SalesdateID\_sk: It is primary key in salesdate dimension table and in fact table it is foriegn key.We can use it to get details about Salesdate dimension.  5) OrderDateID\_SK: It is primary key in orderdate dimension table and in fact table it is foriegn key.We can use it to get details about orderdate dimension .  6)MethodID\_sk:It is primary key in method dimension table and in fact tableit is foriegn key.We can use it to get details about method dimension.  7) Cost amount : It is the measure of cost price for the company with respect to the product. Cost price for each product is calculated by  Cost Price= Unit Price \* Quantity.  8) Sales Amount : It is the measure of sales price for each product of the company. It is calculated by considering the discount offered for product.  When discount = 0,  Sales Amount= price1 \* quantity.  When discount =1,  Sales Amount= price2 \* quantity.  9) Sales\_qty : It is the quantity of sales of a product in a company.  10) Number of days: It is a measure to calculate number of days between sales datse and order date.  Number of days=Sales date- Order date.  11)Discount : It is a measure that denotes whether the product is sold in discounted price.  12) Invoice Number: It is the degenerate dimension of the fact table. |

# 

# 

# II. Dimensional Model



# 

1. Star Dimension Model:

There are six dimensions in the sales fact table. They are product,customer, method, order date,sales date and supplier. Among these dimension sales data and order date is role playing date dimension. Method dimension is considered as the junk dimension for our data mart because of the cardinality. Invoice is considered as a degenerate dimension and is included in the fact table. Fact Measures are included in the fact table. There is a cardinal relationship of one to many between fact and dimensions.

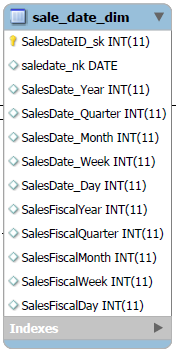
1. Crows Foot Diagram:

Below are screenshots of all the facts and dimensions including all attributes, entities and

primary keys . They are denormalizations of each dimension in the Crows-Foot format.

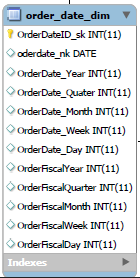
**Sales Date Dimension**

Sales date Dimension has SalesDateID\_sk as surrogate key. Salesdate\_nk is the natural key . Its contains calendar dates as well as Fiscal dates. Sales\_Date\_Dimension is a Role Playing Dimension.



**Order\_Date\_dim**

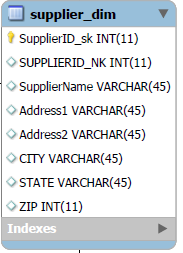
Order date Dimension has OrderDateID\_sk as surrogate key. orderdate\_nk is the natural key . Its contains calendar dates as well as Fiscal dates. order\_Date\_Dim is a Role Playing Dimension.



**supplier Dimension**

SupplierID\_sk is surrogate key and SUPPLIERID\_NK is the natural key. The supplier table is populated with all the supplier details are present in TPCE. TPCW and PEC.

.

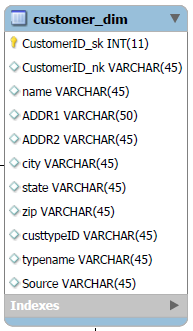


**Customer Dimension**

Customer Dimension has Customer\_sk as the surrogate Key and Customer\_nk as the primary key.

Customer and customer\_type table was combined to get customer dimension.

We added Source in Customer table that maintains, the division name for customer.

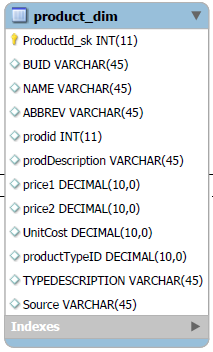


**Product Dimension:**

Product Dimension has ProductId\_sk as the primary key.

The Product Dimension is created with Product table and Product Type.

Source is added to product dimension to analyze from which source the product is got.

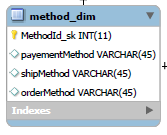


**Method Dimension**

This is the Junk Dimension

This has all the methods that are present for payment, shipment and order.

These methods has low cardinality hence it is made junk



**Fact Table**

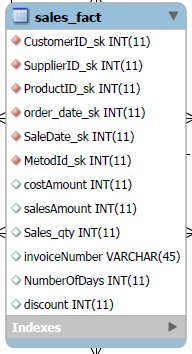
The fact table has following foreign keys

1. CustomerID\_sk that references the customer dimension.
2. SupplierID\_sk that references the Supplier dimension
3. ProductID\_sk that reference the Product dimension
4. order\_date\_sk that reference the Order Date dimension
5. SaleDate\_sk that references the SalesDate dimension
6. MetodId\_sk that references the Method dimension

costAmount, salesAmount Numberof days is measured quantity

invoiceNumber id degenerate Dimension

discount is factless fact



**III. Data Staging: ETL – Data Extract File Definitions**

1.Files:

**TPCE Source Files:**

a)business\_unit.csv: The file format is .csv. The delimiters used in the file is “;”.

b)customer.csv : The file format is .csv. The delimiters used in the file is “;”.

c) customer\_type.csv: The file format is .csv. The delimiters used in the file is “;”.

d) invoice.csv: The file format is .csv. The delimiters used in the file is “,”.

e) invoice\_details.csv: The file format is .csv. The delimiters used in the file is “,”.

f)prod\_type.csv: The file format is .csv. The delimiters used in the file is “;”.

g)product.csv: The file format is .csv. The delimiters used in the file is “;”.

h)The file format is .csv. The delimiters used in the file is “;”.

**PEC Source Files:**

a)PECbusiness\_unit.csv : The file format is .csv. The delimiters used in the file is “;”.

b)PECcustomer.csv: The file format is .csv. The delimiters used in the file is “;”.

c)PECcustomer\_type.csv: The file format is .csv. The delimiters used in the file is “;”.

d) PECinvoice.csv: The file format is .csv. The delimiters used in the file is “,”.

e) PECmanufacturingCosts.csv: The file format is .csv. The delimiters used in the file is “|”.

f) PECproduct.csv: The file format is .csv. The delimiters used in the file is “;”.

g) PECproduct\_type.csv:The file format is .csv. The delimiters used in the file is “;”.

**TPCW Source Files:**

a)TPCWbusiness\_unit.csv:The file format is .csv. The delimiters used in the file is “;”.

b)TPCWcustomer.csv: The file format is .csv. The delimiters used in the file is “;”.

c)TPCWcustomer\_type.csv: The file format is .csv. The delimiters used in the file is “;”.

d)TPCWinvoice.csv: The file format is .csv. The delimiters used in the file is “,”.

e)TPCWproduct.csv: The file format is .csv. The delimiters used in the file is “;”.

f)TPCWproduct\_type.csv: The file format is .csv. The delimiters used in the file is “;”.

2.Data Formats:

# Customer.csv (PEC, TPCE,TPCW)

|  |  |
| --- | --- |
| CustomerID\_sk | Integer |
| name | String |
| addr1 | String |
| addr2 | String |
| city | String |
| state | String |
| zip | Integer |

# 

# Customertype.csv(PEC,TPCE,TPCW)

# 

|  |  |
| --- | --- |
| custtypeID | String |
| typename | String |

# 

# Product.csv(PEC,TPCE,TPCW)

# 

|  |  |
| --- | --- |
| prodid | Integer |
| prodDescription | String |
| price1 | Number |
| price2 | Number |
| unitCost | Number |
| producttypeID | Integer |
| suppliername | String |

# 

# Product\_type.csv (PEC,TPCE,TPCW)

# 

|  |  |
| --- | --- |
| Producttype | Integer |
| Typedescription | String |
| BUID | String |

# 

# Business\_unit.csv (PEC,TPCE,TPCW)

|  |  |
| --- | --- |
| BUID | String |
| NAME | String |
| ABBREV | String |

# Invoice.csv (PEC,TPCE,TPCW)

|  |  |
| --- | --- |
| Invoice | Integer |
| CustID | Integer |
| salesDate | Date |
| prodid | Integer |
| amt | Integer |
| qty | Integer |
| shipMethod | String |
| shipcost | Number |
| paymentMethod | String |
| orderMethod | String |
| orderDate | Date |
| discounted | Integer |

# Manufacturingcost (PEC)

# 

|  |  |
| --- | --- |
| Year | Integer |
| Month | Integer |
| ProdID | Integer |
| manufacturingCost | Integer |

# 

# Supplier.csv (TPCE)

# 

|  |  |
| --- | --- |
| SupplierID | Integer |
| name | String |
| Addr1 | String |
| Addr2 | String |
| City | String |
| State | String |
| zip | Integer |

# 

# Invoice\_details.csv

|  |  |
| --- | --- |
| InvoiceID | Integer |
| ProdID | Integer |
| amt | Number |
| qty | Integer |
| discounted | Integer |

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

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target | | | | | Source | | | | Transformation |
| Table Name | Column  Name | Data Type | Table Type | SCD Type | Division Name | Table Name/ CSV files | Column Name | Data Type |
| Supplier | SupplierID\_SK | INT(11) | Dimension | 0 |  |  |  |  | Generated using random sequence generator to uniquely identify a row. |
| Supplier | Supplier ID | INT(11) | Dimension | 0 | Pec  TPCE  TPCW | PECproduct.csv  supplier.csv  TPCWproduct.csv | Supplier name  Supplier ID  Supplier name | String  Integer    String | Using supplier name, we transformed it to supplierId from supplier table of tpce  It is a natural key that uniquely identifies each customer in PEC source.  Using supplier name, we transformed it to supplierId from supplier table of tpce |
| Supplier | name | VARCHAR (45) | Dimension | 1 | PEC    TPCE  TPCW | PECproduct.csv  supplier.csv  TPCWpoduct.csv | Suppliername  name  Supplier name | String    String  String | supplier name had inc./corp. Using pentaho replace in string changed it to Inc and Corp.  supplier name had inc./corp. Using pentaho replace in string changed it to Inc and Corp.  supplier name had inc./corp. Using pentaho replace in string changed it to Inc and Corp. |
| Supplier | ADDR1 | VARCHAR (50) | Dimension | 0 | TPCE | supplier.csv | Addr1 | String | Checked if it is empty |
| Supplier | ADDR2 | VARCHAR (50) | Dimension | 0 | TPCE | supplier.csv | Addr2 | String | Checked if it is empty |
| Supplier | City | VARCHAR (45) | Dimension | 0 | TPCE | Supplier.csv | City | String | No cleaning required. |
| Supplier | State | VARCHAR (45) | Dimension | 0 | TPCE | Supplier.csv | State | String | No cleaning required |
| Supplier | zip | Varchar(45) | Dimenion | 0 | TPCE | Supplier.csv | Zip | Integer | Checked if the length of 5 is set for all zip codes. |
| Method | MethodId | INT(11) | Dimension | 0 |  |  |  |  | Generated using random sequence generator to uniquely identify a row. |
| Method | ShippingMethod | Varchar(45) | Dimension | 1 | PEC | PECinvoice | shipmethod | String | contained some incorrect  shipping method  . These were corrected using replace string operation in pentaho |
| Method | PaymentMethod | Varchar(45) | Dimension | 0 | PEC | PECinvoice | paymentmethod | String | Checked if it contained empty field |
| Method | OrderMethod | Varchar(45) | Dimension | 1 | PEC | PECinvoice | ordermethod | String | contained some incorrect  order method  . These were corrected using replace string operation in pentaho |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

# V. SQL Code – Tables & Constraints

CREATE DATABASE IF NOT EXISTS `team03` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `team03`;

# Customer Dimension

# 

# DROP TABLE IF EXISTS `customer\_dim`;

# /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

# /\*!40101 SET character\_set\_client = utf8 \*/;

# CREATE TABLE `customer\_dim` (

# `CustomerID\_sk` int(11) NOT NULL,

# `CustomerID\_nk` varchar(45) DEFAULT NULL,

# `name` varchar(45) DEFAULT NULL,

# `ADDR1` varchar(50) DEFAULT NULL,

# `ADDR2` varchar(45) DEFAULT NULL,

# `city` varchar(45) DEFAULT NULL,

# `state` varchar(45) DEFAULT NULL,

# `zip` varchar(45) DEFAULT NULL,

# `custtypeID` varchar(45) DEFAULT NULL,

# `typename` varchar(45) DEFAULT NULL,

# `Source` varchar(45) DEFAULT NULL,

# PRIMARY KEY (`CustomerID\_sk`)

# ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

# /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain

Primary Key (surrogate key):CustomerID\_sk

**Method Dimension**

DROP TABLE IF EXISTS `method\_dim`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `method\_dim` (

`MethodId\_sk` int(11) NOT NULL,

`payementMethod` varchar(45) DEFAULT NULL,

`shipMethod` varchar(45) DEFAULT NULL,

`orderMethod` varchar(45) DEFAULT NULL,

PRIMARY KEY (`MethodId\_sk`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain

Primary Key (surrogate key):MethodId\_sk

**Order Date Dimension**

DROP TABLE IF EXISTS `order\_date\_dim`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `order\_date\_dim` (

`OrderDateID\_sk` int(11) NOT NULL,

`oderdate\_nk` date DEFAULT NULL,

`OrderDate\_Year` int(11) DEFAULT NULL,

`OrderDate\_Quater` int(11) DEFAULT NULL,

`OrderDate\_Month` int(11) DEFAULT NULL,

`OrderDate\_Week` int(11) DEFAULT NULL,

`OrderDate\_Day` int(11) DEFAULT NULL,

`OrderFiscalYear` int(11) DEFAULT NULL,

`OrderFiscalQuarter` int(11) DEFAULT NULL,

`OrderFiscalMonth` int(11) DEFAULT NULL,

`OrderFiscalWeek` int(11) DEFAULT NULL,

`OrderFiscalDay` int(11) DEFAULT NULL,

PRIMARY KEY (`OrderDateID\_sk`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain

Primary Key (surrogate key):OrderDateID\_sk

**Product Date Dimension**

DROP TABLE IF EXISTS `product\_dim`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `product\_dim` (

`ProductId\_sk` int(11) NOT NULL,

`BUID` varchar(45) DEFAULT NULL,

`NAME` varchar(45) DEFAULT NULL,

`ABBREV` varchar(45) DEFAULT NULL,

`prodid` int(11) DEFAULT NULL,

`prodDescription` varchar(45) DEFAULT NULL,

`price1` decimal(10,0) DEFAULT NULL,

`price2` decimal(10,0) DEFAULT NULL,

`UnitCost` decimal(10,0) DEFAULT NULL,

`productTypeID` decimal(10,0) DEFAULT NULL,

`TYPEDESCRIPTION` varchar(45) DEFAULT NULL,

`Source` varchar(45) DEFAULT NULL,

PRIMARY KEY (`ProductId\_sk`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain

Primary Key (surrogate key):ProductId\_sk

**SALES DATE DIMENSION**

DROP TABLE IF EXISTS `sale\_date\_dim`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `sale\_date\_dim` (

`SalesDateID\_sk` int(11) NOT NULL,

`saledate\_nk` date DEFAULT NULL,

`SalesDate\_Year` int(11) DEFAULT NULL,

`SalesDate\_Quarter` int(11) DEFAULT NULL,

`SalesDate\_Month` int(11) DEFAULT NULL,

`SalesDate\_Week` int(11) DEFAULT NULL,

`SalesDate\_Day` int(11) DEFAULT NULL,

`SalesFiscalYear` int(11) DEFAULT NULL,

`SalesFiscalQuarter` int(11) DEFAULT NULL,

`SalesFiscalMonth` int(11) DEFAULT NULL,

`SalesFiscalWeek` int(11) DEFAULT NULL,

`SalesFiscalDay` int(11) DEFAULT NULL,

PRIMARY KEY (`SalesDateID\_sk`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain:

Primary key (Surrogate Key) : SalesDateID\_sk

**Sales Fact**

DROP TABLE IF EXISTS `sales\_fact`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `sales\_fact` (

`CustomerID\_sk` int(11) NOT NULL,

`SupplierID\_sk` int(11) NOT NULL,

`ProductID\_sk` int(11) NOT NULL,

`order\_date\_sk` int(11) NOT NULL,

`SaleDate\_sk` int(11) NOT NULL,

`MetodId\_sk` int(11) NOT NULL,

`costAmount` int(11) DEFAULT NULL,

`salesAmount` int(11) DEFAULT NULL,

`Sales\_qty` int(11) DEFAULT NULL,

`invoiceNumber` varchar(45) DEFAULT NULL,

`NumberOfDays` int(11) DEFAULT NULL,

`discount` int(11) DEFAULT NULL,

KEY `cust\_idx` (`CustomerID\_sk`),

KEY `prod\_idx` (`ProductID\_sk`),

KEY `sup\_idx` (`SupplierID\_sk`),

KEY `sales\_idx` (`SaleDate\_sk`),

KEY `order\_idx` (`order\_date\_sk`),

KEY `meth\_idx` (`MetodId\_sk`),

CONSTRAINT `cust` FOREIGN KEY (`CustomerID\_sk`) REFERENCES `customer\_dim` (`CustomerID\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `meth` FOREIGN KEY (`MetodId\_sk`) REFERENCES `method\_dim` (`MethodId\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `order` FOREIGN KEY (`order\_date\_sk`) REFERENCES `order\_date\_dim` (`OrderDateID\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `prod` FOREIGN KEY (`ProductID\_sk`) REFERENCES `product\_dim` (`ProductId\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `sales` FOREIGN KEY (`SaleDate\_sk`) REFERENCES `sale\_date\_dim` (`SalesDateID\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `sup` FOREIGN KEY (`SupplierID\_sk`) REFERENCES `supplier\_dim` (`SupplierID\_sk`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Foreign Keys

CustomerID\_sk,MetodId\_sk,order\_date\_sk,SaleDate\_sk,SupplierID\_sk

**Supplier Dimenion**

DROP TABLE IF EXISTS `supplier\_dim`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `supplier\_dim` (

`SupplierID\_sk` int(11) NOT NULL,

`SUPPLIERID\_NK` int(11) DEFAULT NULL,

`SupplierName` varchar(45) DEFAULT NULL,

`Address1` varchar(45) DEFAULT NULL,

`Address2` varchar(45) DEFAULT NULL,

`CITY` varchar(45) DEFAULT NULL,

`STATE` varchar(45) DEFAULT NULL,

`ZIP` int(11) DEFAULT NULL,

PRIMARY KEY (`SupplierID\_sk`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

Constrain:

Primary key (Surrogate Key) : SupplierID\_sk

# VI. Data Staging Activities - ETL

## 1. Data Cleansing

|  |  |  |  |
| --- | --- | --- | --- |
| **DM Table** | **Attribute** | **Problem** | **Resolution Strategy** (attach code) |
| TPCW -CUSTOMER | STATE | Replaced to abbrevation for consistency | Replace in String from Pentaho is used to change abbreviations of state to state names. |
| TPCW -CUSTOMER | Address | Abbreviation used for Road as RD, Street as St,Avenue as Av or Ave Drive as Dr | Replace in String is used to replace change abbreviations used for address queries. |
| TPCW -CUSTOMER | ZIP | ZIP codes of different length | Used Excel and Pentaho file input step to make 5 digits for zip codes |
| TPCW -CUSTOMER | Address | Presence of inadequate dots and commas replaced. | Replace in String from Pentaho is used to remove dots and commas. |
| TPCW -CUSTOMER | TYPENAME | State is changed to State/Local Government for conformity | Pentaho Replace in String |
| TPCW -CUSTOMER | TYPENAME | Govt is changed to-US Government for conformity | Pentaho Replace in String |
| TPCW -CUSTOMER | TYPENAME | Edu-Education | Pentaho Replace in String |
| TPCW -CUSTOMER | TYPENAME | Comm-Commercial | Pentaho Replace in String |
| TPCW -CUSTOMER-TYPE | TYPENAME | Abbreviation expanded | Pentaho Replace in String |
| TPCW -CUSTOMER-TYPE | Source | To denote source division of customer | Add Constant |
| PECcustomer-Table | Address | Abbreviation used for Road as Rd, Street as St,Avenue as Av or Ave Drive as Dr | Pentaho Replace in String is used to replace change abbreviations used for address queries. |
| PECcustomer-Table | Address | Presence of inadequate dots and commas replaced. | Pentaho Replace in String |
| PECcustomer-Table | Address | Address is divided into two fields address1 and address 2. | Modified Java Script from Pentaho is used. |
| PECcustomer-Table | custtype | Removing uppercase letters | Pentaho Replace in String |
| PECcustomer-Table | TYPENAME | Abbreviation Used | Pentaho Replace in String |
| PECcustomer-Table | custtype | Deleted | Pentaho Select Values |
| PECcustomer-Table | ZIP | ZIP codes of different length | Used Excel and Pentaho file input step to make 5 digits for zip codes |
| customer\_TPCE | ZIP | ZIP codes of different length | Used Excel and Pentaho file input step to make 5 digits for zip codes |
| PECcustomer-Table | Source | To denote source division of customer | Add Constant |
| customer\_TPCE | address | Trim space before string | String operations trim left |
| customer\_type\_TPCE | Source | To denote source division of customer | Add Constant |
| PECinvoice.csv | orderDate | Presence of wrong delimiters | Used excel to use consistent delimiters |
| PECinvoice.csv | orderDate | Presence of “=” | Used excel to remove illegal characters |
| PECinvoice.csv | OrderMethod | Incorrect Shipping Methods | Used Excel to identify incorrect order method and changed using excel |
| PECinvoice.csv | Shipping Method | Incorrect Shipping Methods | Used Excel to identify incorrect shipping method and changed using excel |
| PECinvoice.csv | orderDate | Presence of wrong date formats | Used excel to format date |
| PECinvoice.csv | SalesDate | Presence of wrong delimiters | Used excel to use consistent delimiters |
| PECinvoice.csv | SalesDate | Presence of wrong date formats | Used excel to format date |
| TPCWinvoice.csv | custID | Occurrence of negative value | Used absloute value |
| TPCWinvoice.csv | custID,prodID,amt,qty | Occurrence of header | Removed row using excel |
| TPCWinvoice.csv | discounted | Mislocation of value in discounted field | Changed misplaced value using excel |
| PEC\_product |  | unit price NULL | set to default 0 |
| PEC\_product | prodDescp | prodDescription has equip and equipment | equip to equipment |
| PEC\_product |  | "" | removed the "" |
| PEC\_product | Unit Cost | Occurrences of null value | Used manufacturing details and calculated unit cost for each product. Calculator from pentaho is used for calculation. |
| PEC\_product\_type |  | "" | removed the "" |
| PEC\_product\_type | prodDescp | type has equip | changed it to equipment |
| PEC\_product\_type/PEC\_product | product type | product type duplicate after merging product and product type | removed prod type |
| TPCW\_product |  | unit price NULL | set to default 0 |
| TPCW\_product | prodDescription | prodDescription has equip and equipment | equip to equipment |
| TPCW\_product | All Attributes | Presence of duplicate records | Cleaned using excel |
| TPCW\_product |  | "" | removed the "" |
| PEC\_product\_type |  | "" | removed the "" |
| TPCW\_product | prodDescp | type has equip | changed it to equipment |
| TPCW\_product | product | product type duplicate after merging product and product type | removed prod type |
| TPCE\_product | prodDescription | prodDescription has equip and equipment | equip to equipment |
| TPCE,PEC.TPCW\_PRODUCT |  | inconsitancy in headers | changed to follow same format |
| TPCE\_product |  | Had inconsistent dataformat | Changed it to for proper format |

## 2. Data Transformation

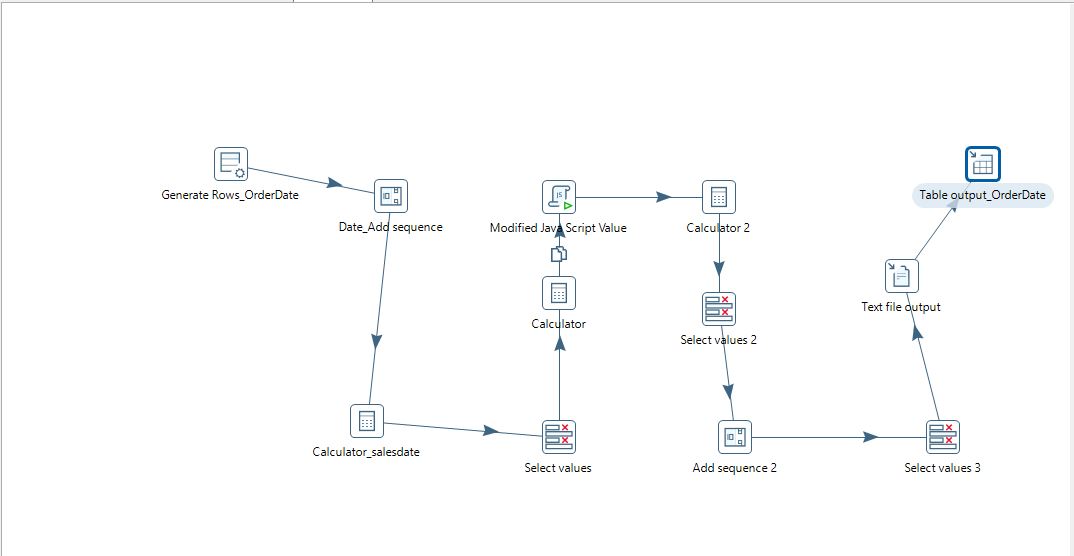
|  |  |
| --- | --- |
| DM Table | Table Population Process |
| Method dim | 1. Fields with least cardinalities are grouped into junk dimension.  Junk dimensions are Payment Type, Ship Method and Order Method.    2. Add sequence to generate unique numbers as surrogate key for Method dimension.  3. Used select values to rename column to make consistent with table in database. |
| Product dim | 1. Product from all sources is joined with product type is so that BUID, productType description, unit cost ,price1 and price 2 are merged together in the same table.  3. Product from TPCE source had BUID in business unit table. Product tableis merged with business unit table to retrieve BUID.  2. Unit cost is calculated for each productID by dividing total cost / total quantity of each product with respect to month and year. It is found that unit cost of product is same for all year.  Unit Cost= Total Cost/ Total quantity  Combined all the product tables of TPCE ,Tpcw and PEC in one table into one  3. Merged products for all sources  4. Add sequence to generate unique numbers as surrogate key for Product dimension dimension. |
| Customer | 1. Customers files are merged with customer types for every sources.  2. Appended customer details from TPCW,TPCE and PEC.  3. Added new field called source to identify source of each record.  4 Created unique surrogate key for all records.  5. Used select values to remove redundant columns after merging customer type and customer table of TPCE,TPCW and PEC. |
| Order Date Dimension | 1.Generated all dates from 1/1/2003 with a limit of 10000 using pentaho.    2. Calculater Year,Quarter ,Month,Week and day from order date using calculator.  3. Using Modified Java Script of Pentaho to calculate fiscal date  Formula Used:  FiscalOrderDate.setValue(getFiscalDate(FiscalOrder.getDate(),"01.04."));    4. Fiscal Year, Fiscal Quarter, Fiscl Month, Fiscal Week and Fiscal Day is calculated  from the fiscal date created in the previous step using calculator of pentaho.    5. Surrogate key is generated to uniquely identify all columns. |
| Sales Date Dimension | 1.Generated all dates from 1/1/2003 with a limit of 10000 using pentaho    2. Calculater Year,Quarter ,Month,Week and day from sales date using calculator.    3. Using Modified Java Script of Pentaho to calculate fiscal date  Formula Used:    FiscalSalesDate.setValue(getFiscalDate(FiscalSales.getDate(),"01.04."))    4. Fiscal Year, Fiscal Quarter, Fiscl Month, Fiscal Week and Fiscal Day is calculated  from the fiscal date created in the previous step using calculator of pentaho.    5. Surrogate key is generated to uniquely identify all columns. |
| Supplier | 1. Suppliers from TPEC,TPCW Product and PEC product are merged together.    2.Supplier name, id, address 1, address 2 ,state,zip and city are selected using using select filter.    3. PEC is added as supplier to the supplier table.    4. Surrogate key is generated to uniquely identify all columns. |
| Sales Fact Table | Transformation of Sales fact table is divided into three different process depending upon the source.    PEC :  1. Extracted PEC invoice and customer dimension to join the fact table using stream lookup with customerID and customer\_sk is retrieved.  2. Product OF PEC is joined with fact table using stream lookup with productID and product\_sk is retrieved.  3. Order date dimension is joined with fact table using stream lookup with orderdateID and orderdate\_sk is retrieved.  4. Method dimension is extracted and joined with fact table using stream lookup with order method , sales method and ship method and then method\_sk is retrieved.  5. Sales date dimension is joined with fact table using stream lookup with Sales dateID and sales date\_sk is retrieved.  6. After supplier merging with product of PEC , stream lookup is used to retrieve supplier sk.  7. Fact Measure Number of days is calculated using Sales date - Order Date.  8. Cost amount is calculated using unit price \* quantity.  10. Sales amount is calculated with price 1 or price 2 with respective to discount.    TPCE :  1. Extracted TPCE invoice and customer dimension of TPCE to join the fact table using stream lookup with customerID and then customer\_sk is retrieved.  2. Product of TPCE is joined with fact table using stream lookup with productID and product\_sk is retrieved.  3. Order date dimension is joined with fact table using stream lookup with orderdateID and orderdate\_sk is retrieved.  4 Sales date dimension is joined with fact table using stream lookup with Sales dateID and sales date\_sk is retrieved.  5. After supplier merging with product of TPCE , stream lookup is used to retrieve supplier sk.  7. Fact Measure Number of days is calculated using Sales date - Order Date.  8. Cost amount is calculated using unit price \* quantity.  10. Sales amount is calculated with price 1 or price 2 with respective to discount.    TPCW:  1. Extracted TPCW invoice and customer dimension of TPCW to join the fact table using stream lookup with customerID and then customer\_sk is retrieved.  2. Product of TPCW is joined with fact table using stream lookup with productID and product\_sk is retrieved.  3. Order date dimension is joined with fact table using stream lookup with orderdateID and orderdate\_sk is retrieved.  4 Sales date dimension is joined with fact table using stream lookup with Sales dateID and sales date\_sk is retrieved.  5. After supplier merging with product of TPCW, stream lookup is used to retrieve supplier sk.  7. Fact Measure Number of days is calculated using Sales date - Order Date.  8. Cost amount is calculated using unit price \* quantity.  10. Sales amount is calculated with price 1 or price 2 with respective to discount. |

## 3.Table Population

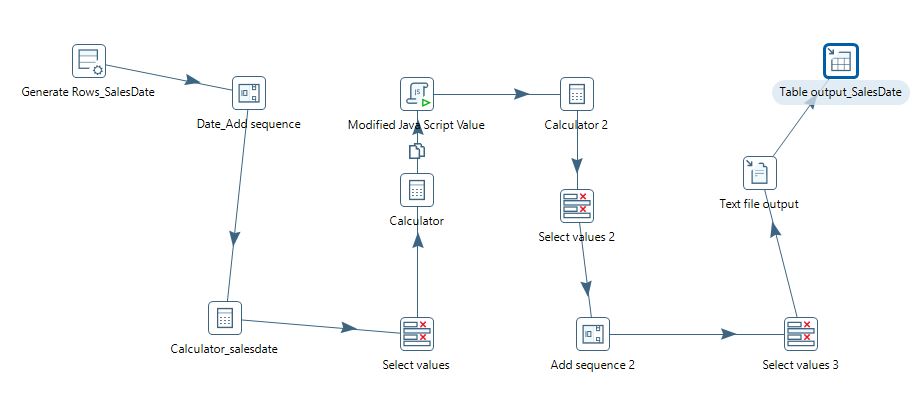
|  |  |
| --- | --- |
| DM | **Table Population Process** (attach code) |
| Customer | 1. Used table Output Step in Pentaho to connect to populate customer dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |
| Product | 1. Used table Output Step in Pentaho to connect to populate Product dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |
| Method | 1. Used table Output Step in Pentaho to connect to populate Method dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |
| Supplier | 1. Used table Output Step in Pentaho to connect to populate Supplier dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |
| Sales Date | 1. Used table Output Step in Pentaho to connect to populate Sales Date dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |
| Order Date | 1. Used table Output Step in Pentaho to connect to populate Order Date dimension.  2. SQL connection is created in Pentaho and loaded data and tested.  3. After previewing results data is loaded to database. |

1. Transformation:

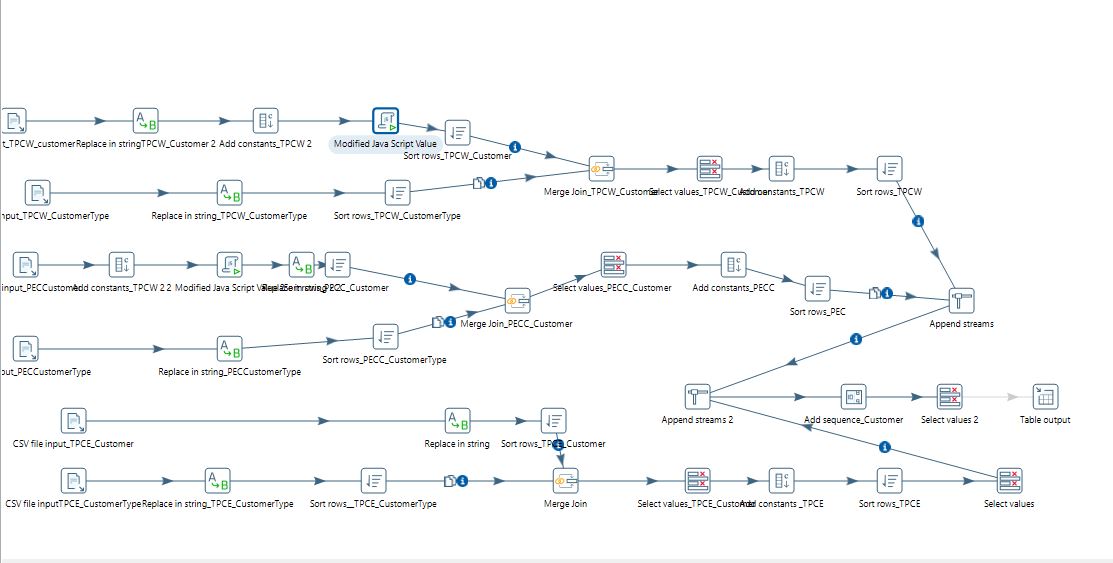
Order Date Dimension



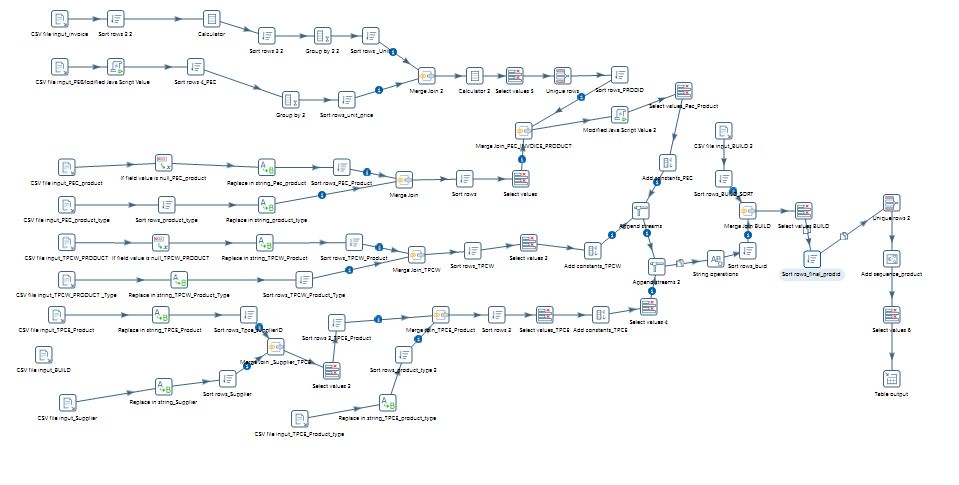
Sales Date Dimension:



Customer Dimension

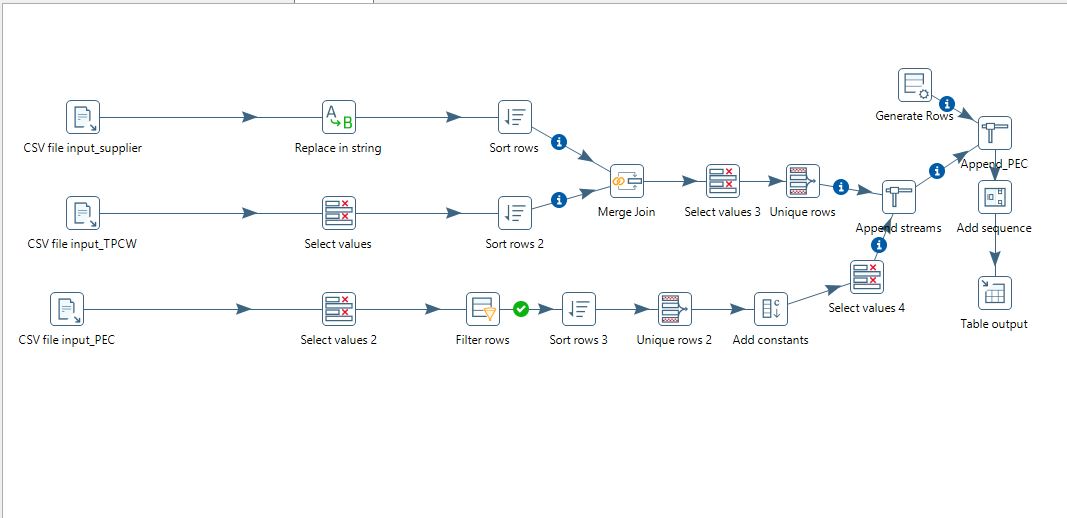


Product Dimension:

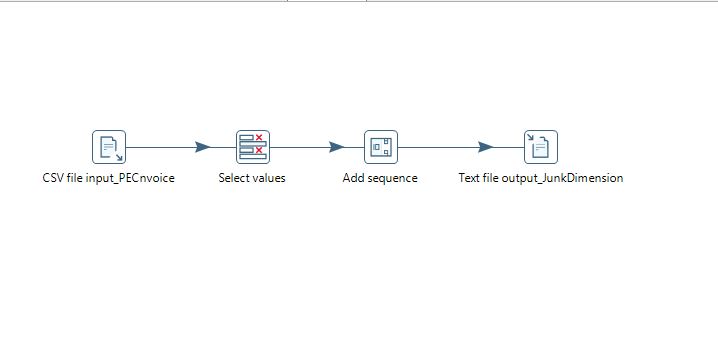


# 

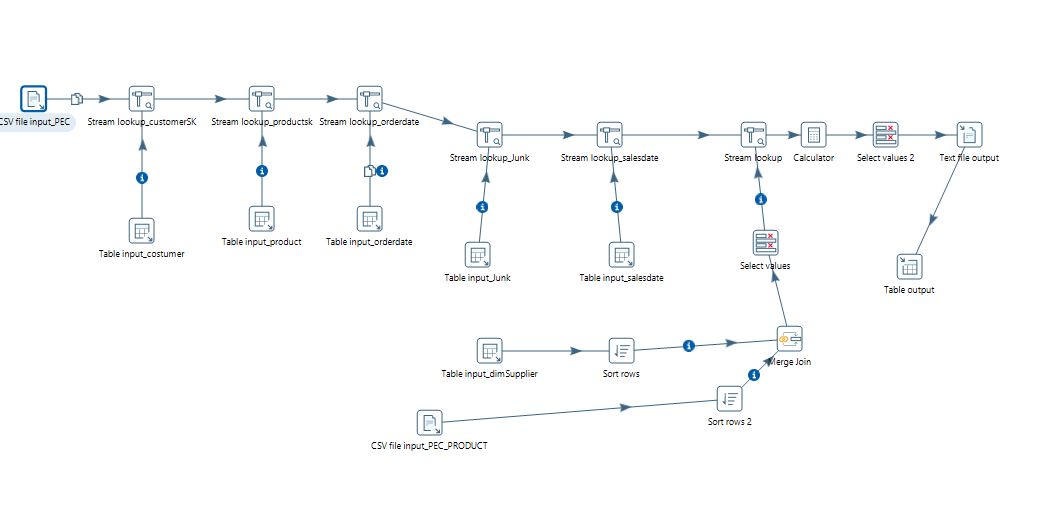
Supplier Dimension



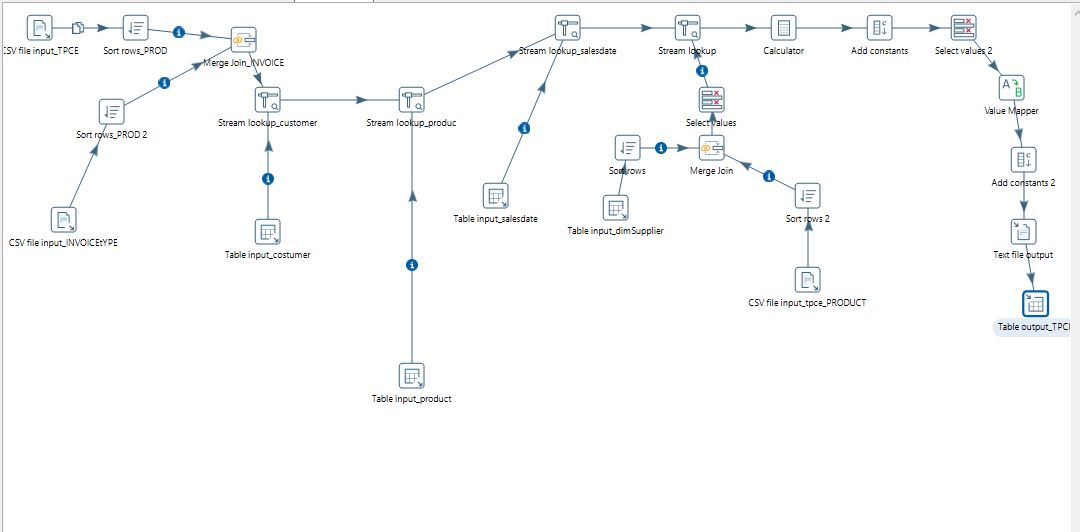
Method Dimension :



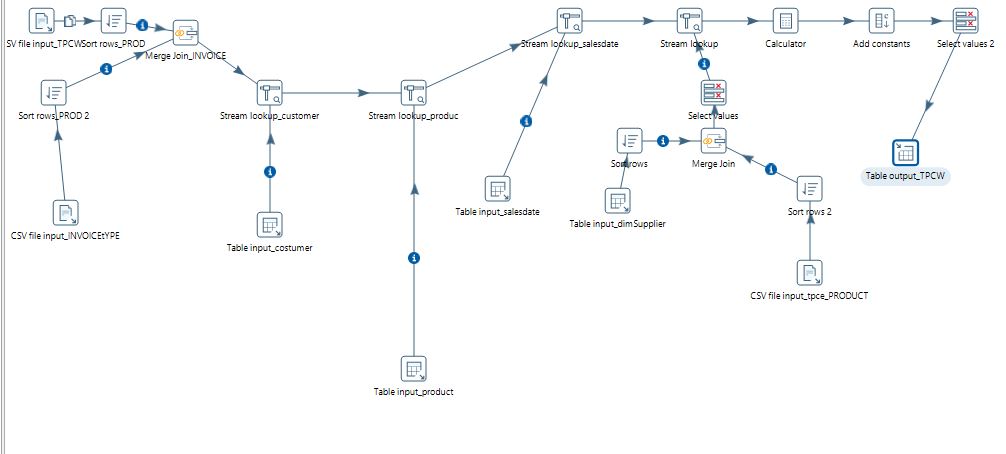
PEC Transform for Fact Table



TPCE - Transform for Fact Table



TPCW – Transform for Fact



# 

# 

# 

# VII. End User Applications

## 1. Queries

1.1 Query 1

User question :

Sales by type of customer, by state, by product type, by business unit.

select cust.state,cust.typename,prod.BUID,prod.TYPEDESCRIPTION,fact.costAmount,fact.salesAmount,fact.Sales\_qty,fact.profit from

(SELECT CustomerID\_sk ,salesAmount,costAmount,Sales\_qty,(salesAmount-costAmount) as profit ,ProductID\_sk FROM team03.sales\_fact) as fact

join

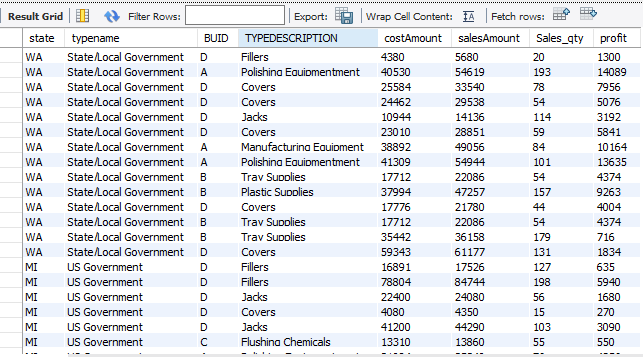
(SELECT ProductId\_sk, BUID,TYPEDESCRIPTION FROM team03.product\_dim) as prod

join

(SELECT CustomerID\_sk ,state, typename FROM team03.customer\_dim) as cust

on fact.CustomerID\_sk= cust.CustomerID\_sk and fact.ProductID\_sk = prod.ProductId\_sk;

Results:



Supporting Index(es): ProductID\_sk(from fact and product Dimension) , CustomerID\_sk (from customerDimension and fact table)

1.2 Query 2

User question : A report that shows the sales, and costs associated with each customer or customer type on an annual, quarterly, monthly or weekly basis.

Results:

select cust.name,Sale.SalesDate\_Year, Sale.SalesDate\_Month ,

Sale.SalesDate\_Quarter ,Sale.SalesDate\_Week, sum(Fact.salesAmount) , sum(Fact.costAmount) from

((SELECT CustomerID\_sk ,name FROM team03.customer\_dim) as cust

join

(SELECT SalesDateID\_sk , SalesDate\_Year, SalesDate\_Month , SalesDate\_Quarter , SalesDate\_Week

FROM team03.sale\_date\_dim) as Sale

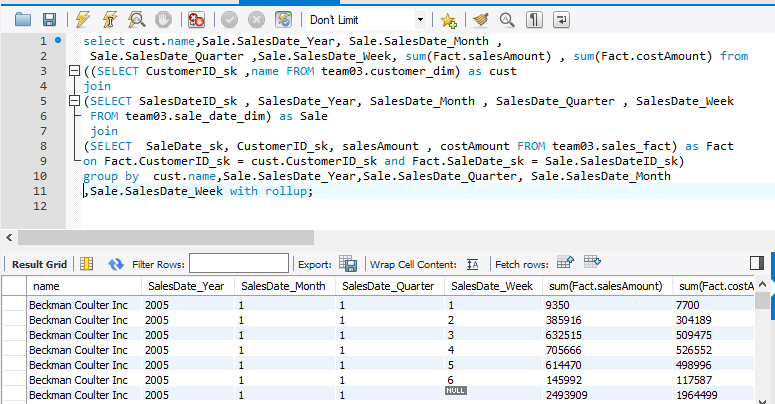
join

(SELECT SaleDate\_sk, CustomerID\_sk, salesAmount , costAmount FROM team03.sales\_fact) as Fact

on Fact.CustomerID\_sk = cust.CustomerID\_sk and Fact.SaleDate\_sk = Sale.SalesDateID\_sk)

group by cust.name,Sale.SalesDate\_Year,Sale.SalesDate\_Quarter, Sale.SalesDate\_Month

,Sale.SalesDate\_Week with rollup;



Supporting Index(es): .CustomerID\_sk(from Customer Dimension and Fact table) and SaleDate\_sk (sales Dimension and Fact)

1.3 Query 3

User Question: what is the average number of products supplied by each supplier?

Results:

Select SupplierName, avg(prod.ProductId\_sk)as Average\_Product\_supplied from

((SELECT ProductId\_sk FROM team03.product\_dim) as prod

join

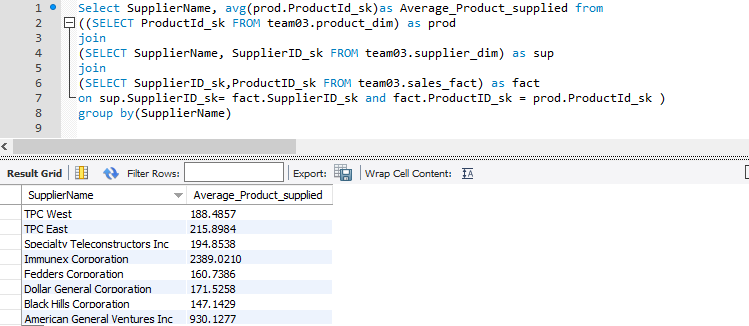
(SELECT SupplierName, SupplierID\_sk FROM team03.supplier\_dim) as sup

join

(SELECT SupplierID\_sk,ProductID\_sk FROM team03.sales\_fact) as fact

on sup.SupplierID\_sk= fact.SupplierID\_sk and fact.ProductID\_sk = prod.ProductId\_sk )

group by(SupplierName)



supporting index : SupplierID\_sk( from supplier Dimension and fact table) and ProductID\_sk( product Dimension and fact table)

## A View

A view shows the sales, and costs associated with each customer or customer type

on an annual, quarterly, monthly or weekly basis

.

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost`

SQL SECURITY DEFINER VIEW `Customer\_Aggreation` AS

SELECT cus.name, sdate.SalesDate\_Year,sdate.SalesDate\_Month ,sdate.SalesDate\_Week,sdate.SalesDate\_Day,

sum(fact.costAmount),sum(fact.salesAmount),sum(fact.profit) FROM

(SELECT name,CustomerID\_sk FROM team03.customer\_dim) as cus

join

(SELECT SalesDateID\_sk, SalesDate\_Year,SalesDate\_Month ,SalesDate\_Day, SalesDate\_Week FROM team03.sale\_date\_dim) as sdate

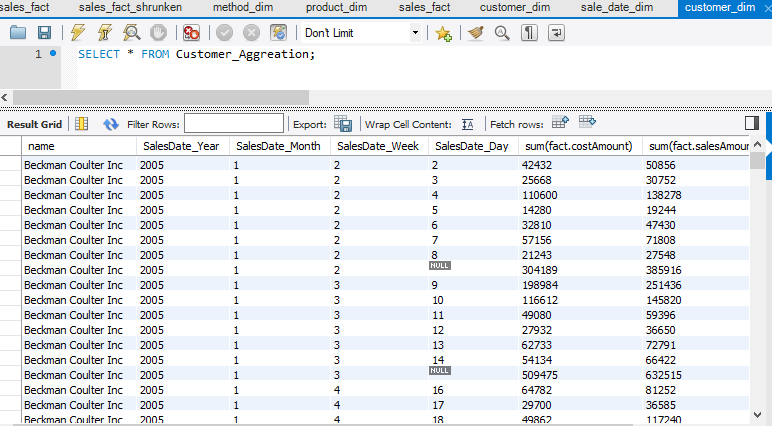
join

(SELECT CustomerID\_sk,SaleDate\_sk , costAmount,salesAmount,(salesAmount - costAmount) profit FROM team03.sales\_fact) fact

ON fact.CustomerID\_sk = cus.CustomerID\_sk and fact.SaleDate\_sk = sdate.SalesDateID\_sk

group by cus.name, sdate.SalesDate\_Year,sdate.SalesDate\_Month ,sdate.SalesDate\_Week,sdate.SalesDate\_Day with rollup;

Select \* From Customer\_Aggreation



3. Aggregated Mata Marts

Implemented Lost, Shrunken and collapsed aggregated dimension for the data mart. In aggregate fact tables facts are pre calculated at higher level of granuality. This is done to improve user experience and run time

3.1 Lost Aggregated Data Mart

In lost dimension aggregated data marts one or more dimensions are completely missing. The lost dimension we chose are Customer,OrderDate, Sales Date and Method dimension. The facts measure that appear in the table have been aggregated across the lost dimensions.

Insert Query for Sales Lost fact Table:

insert into team03.sales\_fact\_lost

(ProductID\_sk,SupplierID\_sk,costAmount,salesAmount,Sales\_qty,NumberOfDays)

(

SELECT ProductID\_sk,SupplierID\_sk,sum(costAmount),sum(salesAmount),sum(Sales\_qty),sum(NumberOfDays)

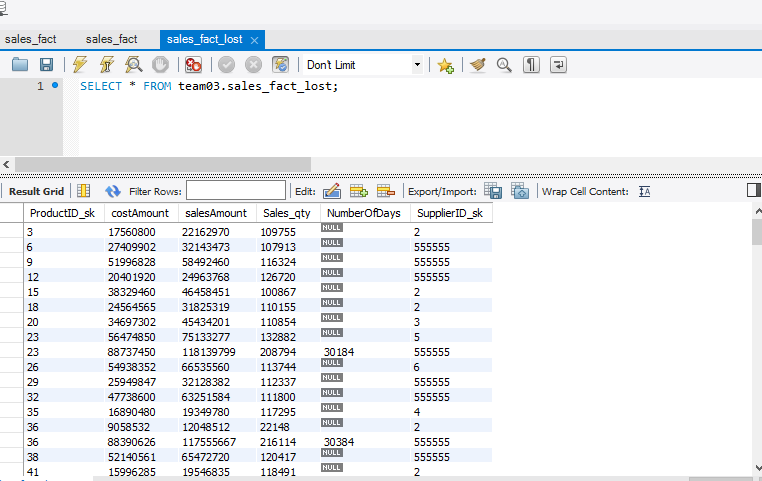
FROM team03.sales\_fact group by ProductID\_sk,SupplierID\_sk

)

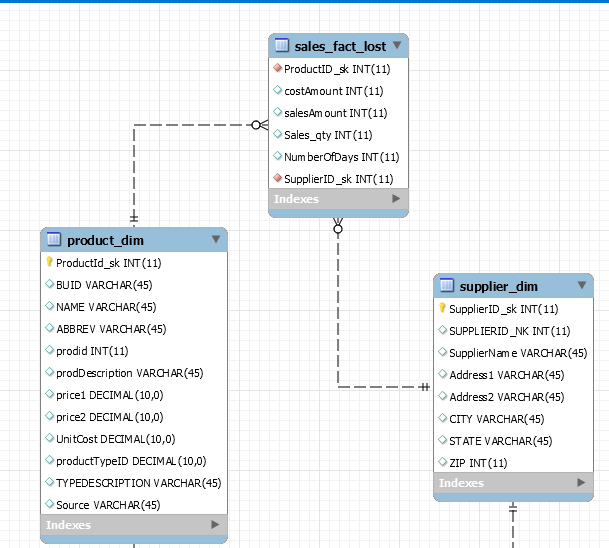
Select Statement for Lost Dimension.

Select statement is used to display fact measures such as Cost Amount , Sales Amount , Sales quantity ,Number of days for every product and supplier. The fact measures are aggregated over the lost dimensions.

Select \* from team03.sales\_fact\_lost,



ER diagram for lost dimension



Queries for Lost Dimension

1. Query1

Use Case:

Query result finds the product for which a supplier could have maximum amount of sales.The results answers the questions such as which product is important to a supplier in terms of sales amount. The results are ordered by decreasing order of sales amount.The select statement provides suppplierID, Supplier Name, ProductID , Product name and maximum amount of sales of supplier.

SELECT a.SupplierID\_sk, b.SupplierName, a.ProductID\_sk, c.NAME , salesAmount from

(select ProductID\_sk, SupplierID\_sk ,salesAmount,Sales\_qty,

costAmount FROM team03.sales\_fact\_lost) as a join

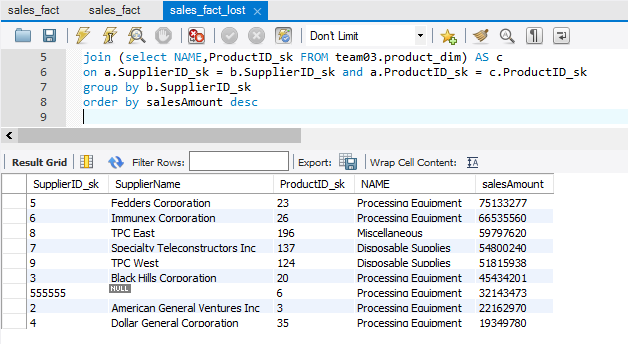
(select SupplierName , SupplierID\_sk from team03.supplier\_dim) as b

join (select NAME,ProductID\_sk FROM team03.product\_dim) AS c

on a.SupplierID\_sk = b.SupplierID\_sk and a.ProductID\_sk = c.ProductID\_sk

group by b.SupplierID\_sk

order by salesAmount desc



2. Query 2

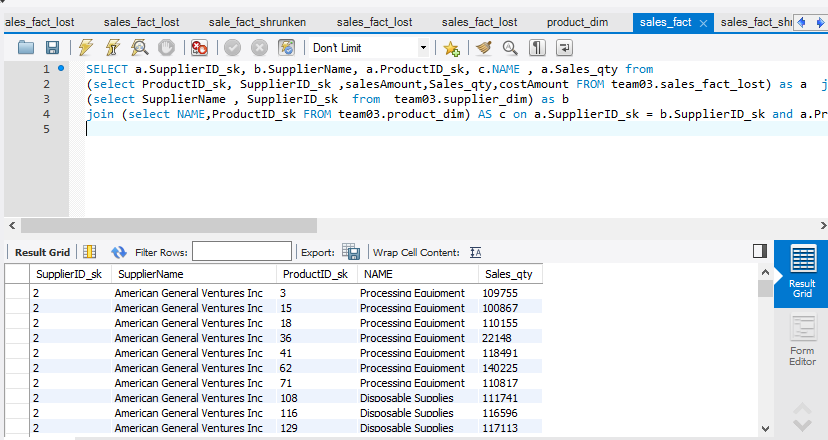
Use Case:

The query finds the quantity of sales every supplier will have on each product and also the quantity of sales associated with each product on a supplier. The select statements retrieves supplieriID, supplier name, productID, product name and the quantity of sales associated with each product and supplier.

SELECT a.SupplierID\_sk, b.SupplierName, a.ProductID\_sk, c.NAME , a.Sales\_qty from

(select ProductID\_sk, SupplierID\_sk ,salesAmount,Sales\_qty,costAmoun,t FROM team03.sales\_fact\_lost) as a join(select SupplierName , SupplierID\_sk from team03.supplier\_dim) as b

join (select NAME,ProductID\_sk FROM team03.product\_dim) AS c on a.SupplierID\_sk = b.SupplierID\_sk and a.ProductID\_sk = c.ProductID\_sk



**3. 2 Shrunken Dimension**

In shrunken dimension one or more dimension is replaced by its own rollup versions. Sales Data dimension is rolleup in such a way that the lowest grain is changed from day to month. Shrunken dimension provides a higher level of summary. Sales date dimension rolled up to month will give monthly analysis of sales. In shrunken dimension. In shrunken dimension method dimension, order date ,product and supplier dimension is lost.

Insert queries for shrunken dimension.

insert into team03.sales\_fact\_shrunken

(YearMonth,CustomerID\_sk,SaleDate\_month\_sk, CostAmount,SalesAmount,Sale\_qty,NumberOfDays)

SELECT concat(SalesDate\_Year,"/" ,SalesDate\_Month),F.CustomerID\_sk,A.SalesDateID\_sk,SUM(F.costAmount),sum(F.salesAmount),SUM(F.Sales\_qty),sum(F.NumberOfDays) from

(SELECT CustomerID\_sk, order\_date\_sk, SaleDate\_sk, costAmount ,Sales\_qty , NumberOfDays,salesAmount

FROM team03.sales\_fact ) F

join

( select SalesDateID\_sk ,SalesDate\_Year , SalesDate\_Month from team03.sale\_date\_dim

group by SalesDate\_Year ,SalesDate\_Month

) as A

ON (A.SalesDateID\_sk) = F.SaleDate\_sk

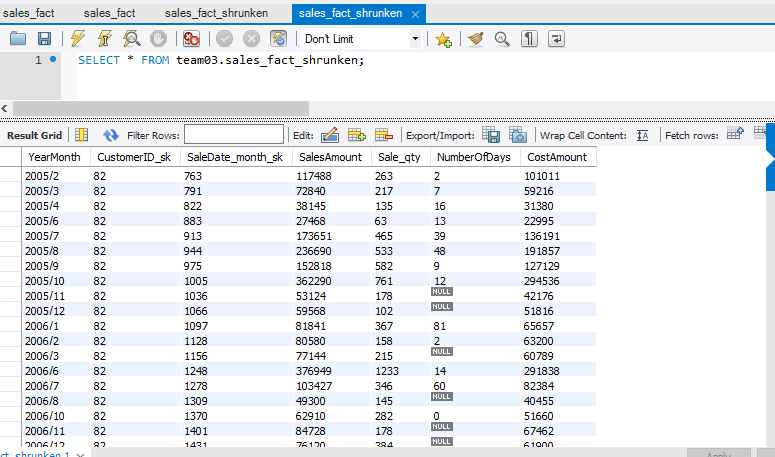
GROUP BY CustomerID\_sk,A.SalesDate\_Year ,A.SalesDate\_Month

;

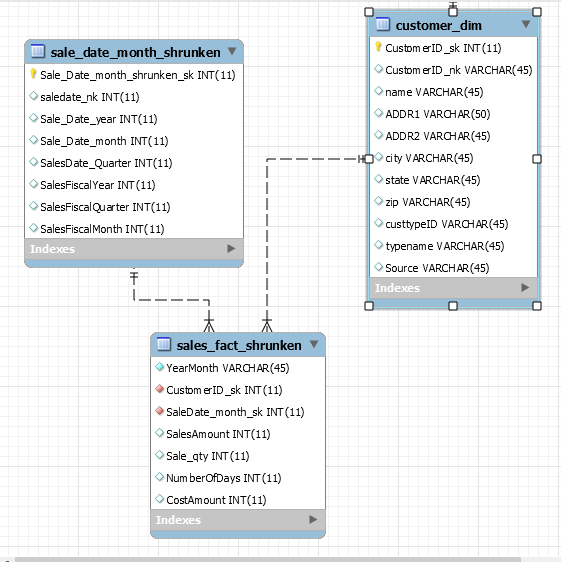
Select statement for shrunken fact table.

Query finds monthly sales measures such as sales quantity, sales amount, cost amount and number of days associated with customers.

SELECT \* FROM team03.sales\_fact\_shrunken;



ER diagram for shrunken dimension



**Queries**

**1. Query 1**

**Use Case:**

The query results identifies the importance of each customer city with respect to sales profit. Results of the query find answers to the questions such as which city is important to the company in a particular month of a year with respect to sales amount for the company. The results are ordered in decreasing order of sales amount.

The select statement provides year, month, city and the sales amount.

# select S.Sale\_Date\_month,S.Sale\_Date\_year, cus.city,salesAmount from

# ((SELECT CustomerID\_sk, SaleDate\_month\_sk,salesAmount FROM team03.sales\_fact\_shrunken) as C

# join

# (select Sale\_Date\_month,Sale\_Date\_year ,Sale\_Date\_month\_shrunken\_sk from team03.sale\_date\_month\_shrunken) as S

# join

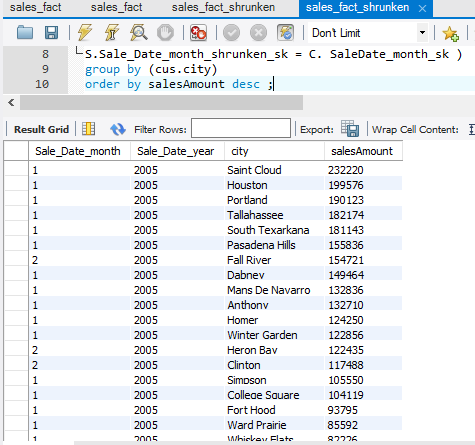
# (SELECT CustomerID\_sk,name,city FROM team03.customer\_dim) as cus

# on cus.CustomerID\_sk = C.CustomerID\_sk and

# S.Sale\_Date\_month\_shrunken\_sk = C. SaleDate\_month\_sk )

# group by (cus.city)

order by salesAmount desc ;

******

2**. Query 2**

**Use Case:**

The query results answers questions such as which all customers are important to the company with respect to profit in each month of a year . The results are ordered in terms of decreasing order of profit so that the the most important customers are in the first few lines of results.

The select statement shows Sales Month, Sales year, Name of customers and profit .

select S.Sale\_Date\_month,S.Sale\_Date\_year, cus.name,(C.salesAmount-C.CostAmount) as Profit from

((SELECT CustomerID\_sk, SaleDate\_month\_sk,salesAmount,CostAmount FROM team03.sales\_fact\_shrunken) as C

join

(select Sale\_Date\_month,Sale\_Date\_year ,Sale\_Date\_month\_shrunken\_sk from team03.sale\_date\_month\_shrunken) as S

join

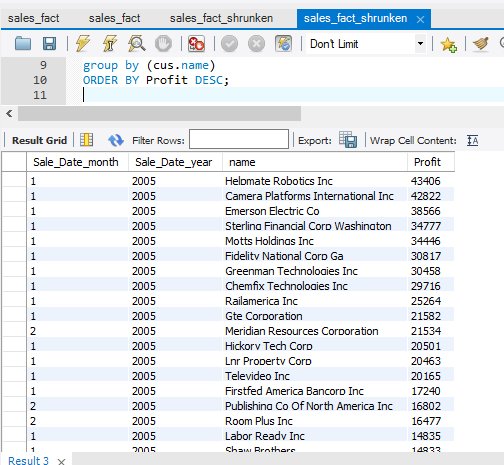
(SELECT CustomerID\_sk,name,city FROM team03.customer\_dim) as cus

on cus.CustomerID\_sk = C.CustomerID\_sk and

S.Sale\_Date\_month\_shrunken\_sk = C. SaleDate\_month\_sk )

group by (cus.name)

ORDER BY Profit DESC;



**3.3 Collapsed Dimension**

In collapsed dimension aggregation is performed on Time dimension( Sales Date and Order Date) and product dimension. Supplier,Customer and Method dimension is lost in collapsed aggregated dimesnsion.

Insert statement for Collapsed dimension

insert into team03.sales\_fact\_colapsed ( order\_date\_sk,SaleDate\_sk,

costAmount,salesAmount,Sales\_qty,

ProductId\_sk,NAME, ABBREV,prodDescription)

(SELECT a.order\_date\_sk,a.SaleDate\_sk,a.costAmount,a.salesAmount,a.Sales\_qty,

a.ProductId\_sk,p.NAME, p.ABBREV,p.prodDescription from

(SELECT CustomerID\_sk,ProductId\_sk,SupplierID\_sk, order\_date\_sk,SaleDate\_sk,MetodId\_sk,sum(costAmount) as costAmount,

sum(salesAmount) as salesAmount ,sum(Sales\_qty) as Sales\_qty

FROM team03.sales\_fact group by ProductId\_sk) as a

join

(SELECT ProductId\_sk,BUID,NAME, ABBREV,prodid,prodDescription,price1

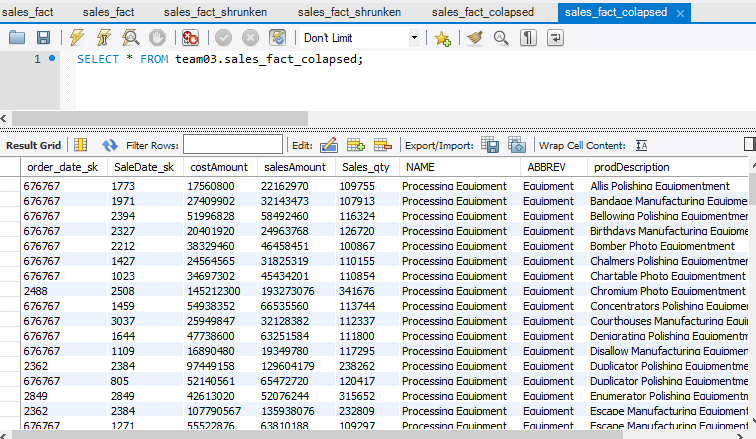
,price2,UnitCost,productTypeID,TYPEDESCRIPTION,Source FROM team03.product\_dim) as p

on p.ProductId\_sk= a.ProductId\_sk)

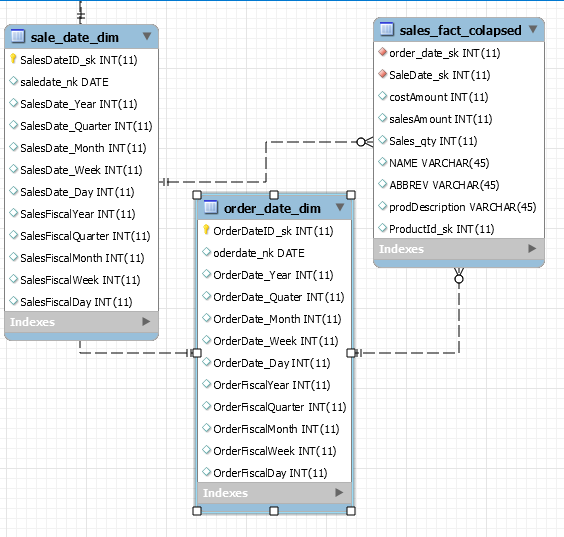
Select Statement for collapsed dimension

To display fact measures and product descriptions of each each product aggregated with product, sales date and order date.

SELECT \* FROM team.03.sales\_fact\_collapsed;



ER diagram for collapsed dimension



**Queries**

**1. Query 1**

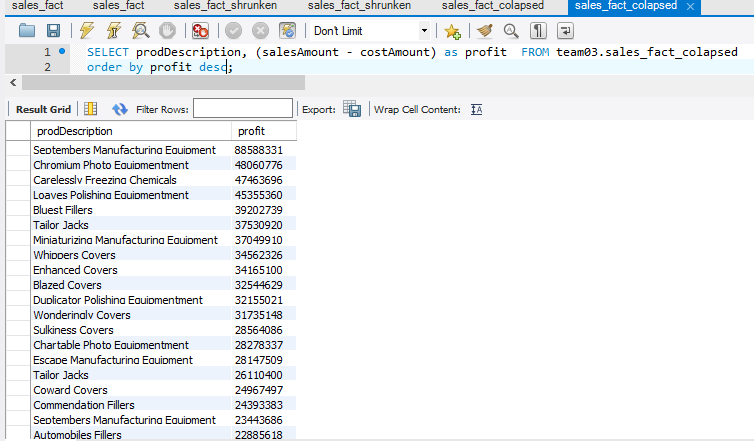
**Use Case:**

The query answers the question how much profit the company earned for each product with its descriptions.From this query it is able identify importance of product with respect to profit.

The select statement provides product description and profit associated with each product. The results are ordered by profit in descending order.

SELECT prodDescription, (salesAmount - costAmount) as profit FROM team03.sales\_fact\_colapsed

order by profit desc;



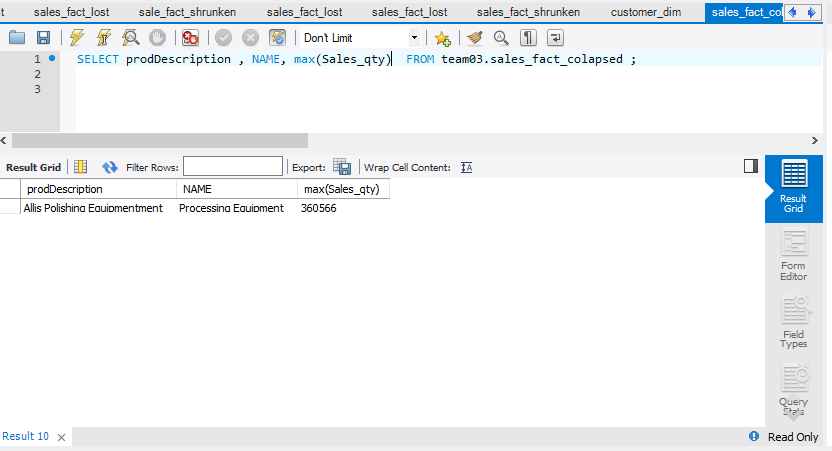
**2. Query 2**

**Use Case:**

The query identifies the product with its description. The product with maximum sales quantity is returned. This query answers question that which product has maximum sales in terms of quantity for the company.

The select statement provides product description, product name and the maximum sales quantity .

SELECT prodDescription , NAME, max(Sales\_qty) FROM team03.sales\_fact\_colapsed ;



# VIII. Handling Slowly Changing Dimensions (SCD)

SCD Implementation

* To implement SCD considered 25 records.
* SCD TYPE 1 and TYPE 2 are implemented

the following tables shows the Attributes that were considered for SCD changed

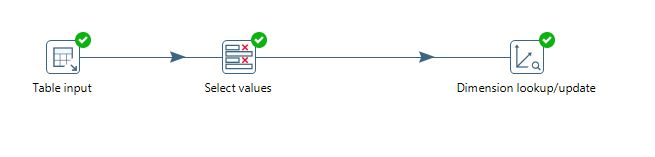
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Attribute | SCD Type | old | New |
| 1 | TypeName  State  city | Type 1  Type 2  Type 2 | State/Local Government  HI  Bloomington | State Government  CA  Fremont |
| 2 | TypeName  name | Type 1  Type 2 | State/Local Government  Clio-Hutchinson | State Government  Clio-Hutchinson Corporation |
| 3 | State  city | Type 2  Type 2 | KY  Laguna Woods | FL  Orlando |
| 4 | State  city | Type 2  Type 2 | DC  Alexandria | NY  Rochester |
| 5 | CustTypeID  name | Type 1  Type 2 | F  Saint Andrews Golf Corp | U  Saint Andrews |
| 6 | State  city | Type 2  Type 2 | PA  Pullman | MA  Boston |
| 7 | name | Type 2 | Hop Adams | Richard Adams |
| 8 | TypeName  name | Type 1  Type 2 | State/Local Government  Mullins Inc | State Government  Mullins Corporation |
| 9 | city | Type 2 | Fall River | Boston |
| 10 | name | Type 2 | Byron Chemicals | Byron  Corporation |
| 11 | city | Type 2 | Idabel | Littleton |
| 12 | State  city | Type 2  Type 2 | IL  Naperville | MA  Lowell |
| 13 | TypeName  State  city | Type 1  Type 2  Type 2 | State/Local Government  San Juan  WA | State Government  San Jose  CA |
| 14 | TypeName  Name | Type 1  Type 2 | State/Local Government  The Final Frontier | State Government  The Final Technologies |
| 15 | State  city | Type 2  Type 2 | DC  West Haven | AZ  Chandler |
| 16 | CustTypeID  city | Type 1  Type 2 | F  Tucson | U  Vancouver |
| 17 | TypeName  city | Type 1  Type 2 | State/Local Government  Houston | State Government  Austin |
| 18 | Name  City  CustTypeID | Type2  Type2  Type 1 | Cross  Hopkinsville  F | Cross Inc  Spuistraat  U |
| 19 | City  Name | Type 2  Type2 | Ward Prairie  Firstfed America Bancorp Inc | Austin  Firstfed America |
| 20 | TypeName  Type 2  type2 | Type 1  State  city | State/Local Government  MI  Sault Ste. Marie | State Government  NY  Buffalo |
| 21 | Name  city | Type 2  Type 2  Type 2 | Santos LLC.  Billings | Santos Corporation  Bloomington |
| 22 | TypeName  name | Type 1  Type 2 | State/Local Government  Santiago Processing | State Government  Santiago Inc |
| 23 | TypeName  name | Type 1  Type 2 | State/Local Government  Chemfix Technologies Inc | State Government  Chemfix Technologies |
| 24 | TypeName  name | Type 1  Type 2 | State/Local Government  Chantale-Huffman Inc. | State Government  Chantale-Huffman Corporation |
| 25 | Name | Type 2 | Garrison Works | Garrison Corporation |

SCD Transformation:

The Transformation is shown below :

The SCD’s were implemented with Dimensional Lookup

In Dimensional Look up punchthrough was used for SCD type 1 and for Type 2 Insert was used

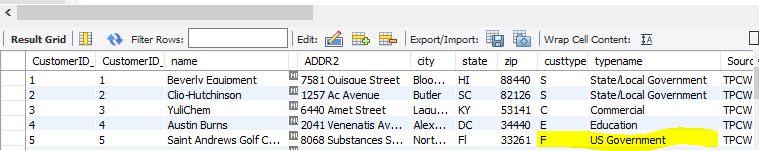


SCD Type 1 results

In this custtype is changed to U from F wherere the type is us government . This is done with SCD type 1 Because we have considered that the type has been wrongly written as F and hence the company doesn't want to keep the track of the the type name.

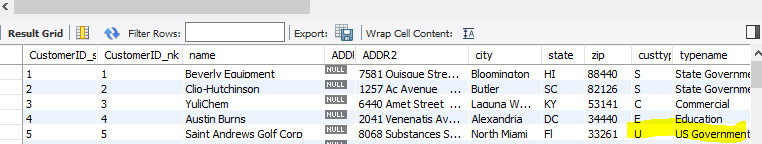
SELECT \* FROM team03.customercopy;

Before SCD was Implemented



After Implentation

SELECT \* FROM team03.customercopy;



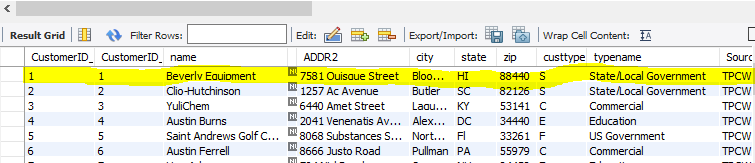
SCD Type 2

In type 2 we have considered the change of state from HI to CA.

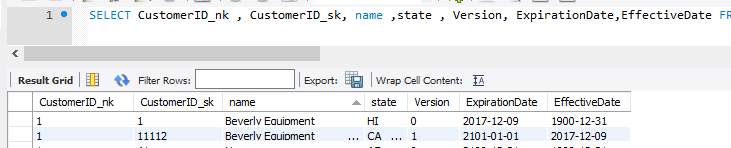
for this a new record is inserted with version 1 that says the version is the current verison

(the heighest version is the heighest version)

before

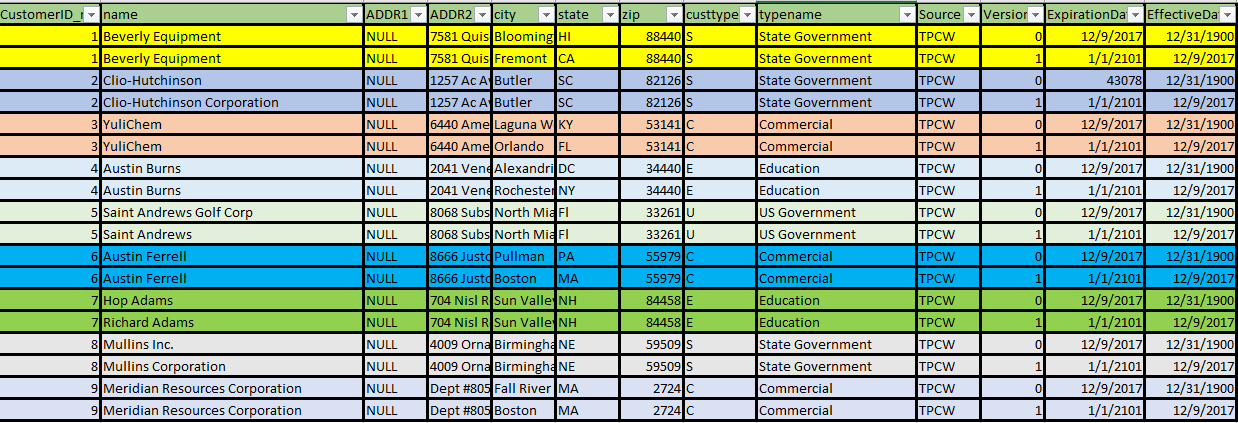


SELECT CustomerID\_nk , CustomerID\_sk, name ,state , Version, ExpirationDate, EffectiveDate FROM team03.customercopy where customerID\_nk = 1;



below is the scree shot of some SCD 2 implementation in an execl sheet that has been exported after quering customer copy table

SELECT \* FROM team03.customercopy;



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

Usually one row in a dimension, can have many rows in the fact table, but one row in the fact table should belong to only one dimensional row. Sometimes, it may happen that Fact table is associated with more than one value in a dimension. To capture this scenario we need to implement N:M relationship in a data mart.

The different approaches for implementing many-to-many relationships.

a) The Bridge Table

b) De-normalizing the Dimension Table by Positional-Flag Attributes

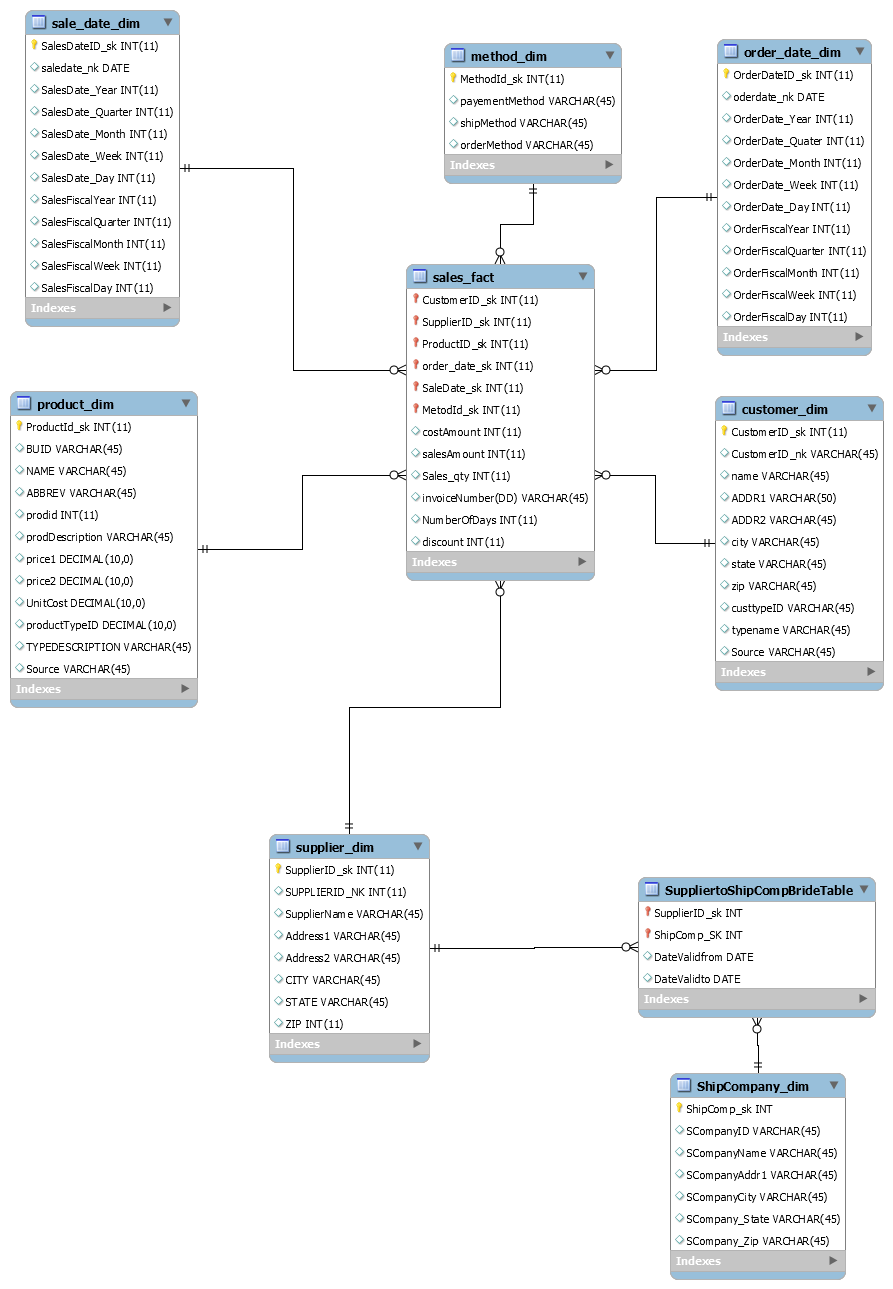
c) De-normalizing the Dimension Table by Non-Positional attributes and a Concatenated Field

d) Lowering the Grain of the Fact Table

We believe Kimball’s method of Bridge table will be the best as it has least redundancy as compared to other methods and it is a ideal solution when there is no limit on the maximum count of many-to-many relationships (Il-Yeol,William, Carl, & Edward, 2001)

If the present Data Mart that we are implementing is considered, Supplier needs to ship a product and to ship this, the supplier is associated with many shipping companies and one shpping company can be associated with multiple suppliers.

We have made a suppliertoshipcompBrigetable which is consistig of surrogate keys from Supplier\_dim and ShipCompany\_dim. Apart from that we have added valid to and valid from date attribute because we feel that supplier and shipping company contract would be for limited time and it would keep changing



References

Il-Yeol, S., William, R., Carl, M., & Edward, E. (2011). An Analysis of

Many-to-Many Relationships Between Fact and Dimension Tables in Dimensional

Modeling. Conference: Proceedings of the 3rd Intl. Workshop on Design and

Management of Data Warehouses, DMDW'2001, Interlaken, Switzerland, June 4,

2001. Retrieved from <https://www.researchgate.net/publication/>

X. Appendix (Fix Lab #3 Problems)

|  |  |
| --- | --- |
| comments | correction Steps |
| Incorrect Shipping Methods | the PECinvoice contained some incorrect shipping method like aiir ,tick .  These were corrected using string operation in pentaho |
| Incorrect order Methods | the PECinvoice contained some incorrect  order method  . These were corrected using replace string operation in pentaho |
| Did not Recalculet Slaes amount | the Sales amount was not present in CSV. Its has been added to CSV files |
| Zero not added to 4 digit zip codes | Zip code were made to be preceding with zero if they are not 5 digit,  Input csv in pentaho had option to append zeros |
| did not split the address field | the Address field is not split into addr1 and addr2 |
| PEC not listed | pec is not an entry in supplier table. Done using pentaho |
| duplicates Present in Product(TPCW) | the duplicated were removed just befor inserting it to dimesion  table by a tranformation called unique, that filters the duplicate rows |
| Named dimension JUNK | the dimension is renamed to method |
| Fiscal date not present in  SalesDate and Order Date | Fiscal date is added to both the Dimensions |
| Junk has more dim | removed duplicated and reloaded the Junk dimension |
| week missing in Sales | Week has been added to sales dime |