***The Product Company***

**~ Final Data Mart Development Report ~**

Team # 6

Team Members

Adwait Kirtikar

Nandakumar Gunalan

Samita Pradhan

Siva Datta Koneru

12/11/2015

ISTE   Data Warehousing

~ Table of Contents ~

[I.  \_ Data Mart Design Definition](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.30j0zll)

[1. Universe of Discourse](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.1fob9te)

[2. Information Package](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.3znysh7)

[II.  \_ Dimensional Model](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.2et92p0)

[III.  \_ Data Staging: ETL – Data Extract File Definitions](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.tyjcwt)

[IV.  \_ Data Staging: ETL – Source-to-Target Mappings](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.3dy6vkm)

[V.  \_ SQL Code – Tables & Constraints](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.1t3h5sf)

[VI.  \_ Data Staging Activities - ETL](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.4d34og8)

[1. Data Cleansing](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.2s8eyo1)

[2. Data Transformation](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.17dp8vu)

[3. Table Population](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.3rdcrjn)

[VII.  \_ End User Applications](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.26in1rg)

[1. SQL99 Queries](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.lnxbz9)

[2. A Views \_](https://docs.google.com/document/d/1ZyDXJrotihs80Ury3Rk3bvzpdVWbC958ZN0vYbYpM6Q/edit#heading=h.35nkun2)

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

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

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



# I.  Data Mart Design Definition

## 1. Universe of Discourse

|  |
| --- |
| Financial analysis data mart will investigate financial performance of The product company to optimize sales while keeping the cost down. Using this data mart they can manage relationship with their suppliers and they can analyze their sales for specific time period to help them improve profit. |

## 2. Information Package

Process Name: Financial analysis

Grain: The number of products sold for each customer in a day.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Customer | Product | Supplier | OrderDate | SaleDate | salesfact | miscellaneous |
| Customer\_SK | Product\_SK | Supplier\_SK | OrderDate\_SK | SaleDate\_SK | InvoiceID | misc\_SK |
| CustID | ProductID | supplier\_ID | Year | Year | Customer\_SK | ID |
| Name | ProductName | Name | Quarter | Quarter | OrderDate\_SK | shipMethod |
| Address | ProductTypeId | State | Month | Month | SaleDate\_SK | paymentMethod |
| City | TypeDescription | City | fiscalYear | fiscalYear | Product\_SK | orderMethod |
| State | BUID | ZIP | fiscalQuarter | fiscalQuarter | Supplier\_SK |  |
| Zip | divisionID | SupplierAddress | fiscalMonth | fiscalMonth | misc\_SK |  |
| Custtype | BUAbbrev | SupplierAttn | orderDate | saledate | amt |  |
| TypeName | BUName |  |  |  | qty |  |
| DivId |  |  |  |  | discounted |  |
|  |  |  |  |  | daysToComplete |  |
|  |  |  |  |  | UnitCost |  |
|  |  |  |  |  | Price |  |

Facts : amt, quantity, discounted, no of days to complete order, unit cost, price.

**3. Entity Definitions**

|  |  |
| --- | --- |
| **Entity** | **Entity Definition** (*genus differentia*) |
| **salesdate** | This table contains the data of sales record |
| **orderdate** | This table contains the data of order record |
| **Customer** | This table contains the customer records |
| **supplier** | Is the table which stores the supplier records. |
| **product** | Is the table which stores the product details. |
| **Miscellaneous** | Is the table which contains the records of the following three attributes.  shipMethod - contains the information of shipping method.  paymentmethod - contains the record of payment method.  orderMethod - contains the record of order method. |

# II.  Dimensional Model

# *https://lh5.googleusercontent.com/89vVBfwt6ql_Xu6YeOEE1SfTowpeOjr8b377v06S8-qPfvlIc5atIDlHn5HgeV5kc6bv8epenAOEDhJei16OYkj0xUa0a6fGXFcsSUVrPj0M98PKsGrL_AiatOgH9mxO22hqshUi*

# III.

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

* files
* data formats

Data dictionary TPC W

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Table name** | **Data type** | **Description** | **Sample 1** | **Sample 2** | **File Format** |
| BUID | TPCWbusiness\_unit | CHAR(1) | Business unit ID |  |  | CSV |
| NAME |  | VARCHAR(45) | Business unit name |  |  |  |
| ABBREV |  | VARCHAR(45) | Business unit Abbrevation |  |  |  |
|  |  |  |  |  |  |  |
| CUSTTYPEID | customer\_type | CHAR(1) | it represents the customer type ID | C | E | CSV |
| TYPENAME |  | VARCHAR(45) | It represents the type of the customer | Commercial | Education |  |
|  |  |  |  |  |  |  |
| CUSTID | TPCWcustomer | INT(11) | It represents the customer ID of TPC E |  |  | CSV |
| NAME |  | VARCHAR(50) | It represents the customer name of TPC E |  |  |  |
| Address |  | VARCHAR(50) | It is the second part of the address |  |  |  |
| CITY |  | VARCHAR(50) | It is the city of the customer |  |  |  |
| State |  | CHAR(2) | It is the code of the state that the customer stays in |  |  |  |
| ZIP |  | CHAR(45) | It is the zip code of the customer address |  |  |  |
| CUSTTYPEID |  | CHAR(1) | It represents the customer type ID of customer in TPC E |  |  |  |
|  |  |  |  |  |  |  |
| PRODTYPEID | TPCWprod\_type | VARCHAR(45) | It represents the product type ID | 3 |  | CSV |
| TYPEDESCRIPTION |  | VARCHAR(45) | It represents the product description | Polishing Equipmen | Polishing Equipmen |  |
| BUID |  | CHAR(1) | Business unit ID | A | B |  |
|  |  |  |  |  |  |  |
| ProductID | TPCWproduct | INTEGER(11) | It is the ID of the product table of TPCW |  |  | CSV |
| Product Name |  | VARCHAR(45) | It is the product name |  |  |  |
| Price1 |  | VARCHAR(45) | It is the price without discount |  |  |  |
| Price2 |  | VARCHAR(45) | It is the price with discount |  |  |  |
| Unit Cost |  | VARCHAR(45) |  |  |  |  |
| Supplier Name |  | VARCHAR(45) | It is the name of the supplier |  |  |  |
| Supplier Address |  | VARCHAR(45) | It is the address of the supplier |  |  |  |
| Supplier city |  | VARCHAR(45) | It is the city of the supplier |  |  |  |
| Supplier State |  | CHAR(2) | It is the state code of the state in which the supplier is located |  |  |  |
| Supplier zipcode |  | CHAR(5) | It is the zip code of the supplier in which he is located |  |  |  |
|  |  |  |  |  |  |  |
| Invoice | TPCWsales\_new | INTEGER(11) | It is the invoice ID |  |  | CSV |
| custID |  | INTEGER(11) | It us the ID of the customer |  |  |  |
| prodID |  | INTEGER(11) | It is the ID of the product |  |  |  |
| Salesdate |  | DATE | it is the actual sales date |  |  |  |
| Amt |  | INTEGER(11) | amount of the sales |  |  |  |
| Qty |  | INTEGER(11) | quantity of the sales |  |  |  |
| Discounted |  | BOOLEAN | It is a flag for discounted or not discounted |  |  |  |

Data dictionary PEC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Table name** | **Data type** | **Description** | **Sample 1** | **Sample 2** | **File Format** |
| BUID | PECWbusiness\_unit | CHAR(1) | Business unit ID |  |  | CSV |
| NAME |  | VARCHAR(45) | Business unit name |  |  |  |
| ABBREV |  | VARCHAR(45) | Business unit Abbrevation |  |  |  |
|  |  |  |  |  |  |  |
| CUSTTYPEID | PECcustomer\_type | CHAR(1) | it represents the customer type ID | C | E | CSV |
| TYPENAME |  | VARCHAR(45) | It represents the type of the customer | Commercial | Education |  |
|  |  |  |  |  |  |  |
| CUSTID | PECWcustomer | INT(11) | It represents the customer ID of PEC |  |  | CSV |
| NAME |  | VARCHAR(50) | It represents the customer name of PEC |  |  |  |
| Address |  | VARCHAR(50) | It is the second part of the address |  |  |  |
| CITY |  | VARCHAR(50) | It is the city of the customer |  |  |  |
| state |  | CHAR(2) | It is the code of the state that the customer stays in |  |  |  |
| ZIP |  | CHAR(45) | It is the zip code of the customer address |  |  |  |
| CUSTTYPEID |  | CHAR(1) | It represents the customer type ID of customer in PEC |  |  |  |
|  |  |  |  |  |  |  |
| YEAR | PECmanufacturingCosts |  | It is the year field |  |  | CSV |
| MONTH |  |  | It is the month field |  |  |  |
| prodID |  |  | It is the productID |  |  |  |
| manufacturingCost |  |  | It is the manufacturing cost for the product at that time of the year and month |  |  |  |
|  |  |  |  |  |  |  |
| PRODTYPEID | PECproduct\_type | VARCHAR(45) | It represents the product type ID |  |  | CSV |
| TYPEDESCRIPTION |  | VARCHAR(45) | It represents the product description |  |  |  |
| BUID |  | CHAR(1) | Business unit ID |  |  |  |
|  |  |  |  |  |  |  |
| prodid | PECproduct |  | Its is the natural key of product table |  |  | CSV |
| prodDescription |  |  | Gives the description of the product |  |  |  |
| price1 |  |  | It is the price that is not discounted |  |  |  |
| price2 |  |  | Is the price that is after discount. |  |  |  |
| unitCost |  |  | is the unit cost of the product |  |  |  |
| supplierName |  |  | It is the name of the supplier of the product |  |  |  |
| productTypeID |  |  | Is the ID of the product |  |  |  |
|  |  |  |  |  |  |  |
| invoice | PECsales\_new | INTEGER(11) | It is the invoice ID |  |  | CSV |
| custID |  | INTEGER(11) | It is the natural key f the customer |  |  |  |
| salesDate |  | DATE | It is the sales date |  |  |  |
| prodID |  | INTEGER(11) | It is the natural key of the product |  |  |  |
| amt |  | Decimal(10,2) | It represents the amount |  |  |  |
| qty |  | INTEGER(11) | It represents the quantity |  |  |  |
| shipMethod |  | VARCHAR(45) | It represents the shipping method |  |  |  |
| shipCost |  | VARCHAR(45) | It represents the shipping cost |  |  |  |
| paymentMethod |  | VARCHAR(45) | It represents the payment method |  |  |  |
| orderMethod |  | VARCHAR(45) | It represents the order method |  |  |  |
| orderDate |  | VARCHAR(45) | It is the order date |  |  |  |
| discounted |  | BOOLEAN | It is a flag to indicate wether it is dicounted or not |  |  |  |

Data dictionary TPC E

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Table name** | **Data type** | **Description** | **Sample 1** | **Sample 2** | **File Format** |
| BUID | business\_unit | CHAR(1) | Business unit ID |  |  | CSV |
| NAME |  | VARCHAR(45) | Business unit name |  |  |  |
| ABBREV |  | VARCHAR(45) | Business unit Abbrevation |  |  |  |
|  |  |  |  |  |  |  |
| CUSTTYPEID | customer\_type | CHAR(1) | it represents the customer type ID | C | E | CSV |
| TYPENAME |  | VARCHAR(45) | It represents the type of the customer | Commercial | Education |  |
|  |  |  |  |  |  |  |
| CUSTID | customer | INT(11) | It represents the customer ID of TPC E | 0 | 1 | CSV |
| NAME |  | VARCHAR(50) | It represents the customer name of TPC E | Shaw Brothers | United Healthcare Corporation |  |
| Addr1 |  | VARCHAR(50) | It is one part of the address |  |  |  |
| Addr2 |  | VARCHAR(50) | It is the second part of the address |  |  |  |
| CITY |  | VARCHAR(50) | It is the city of the customer | Bridgeport | Columbia |  |
| ZIP |  | CHAR(45) | It is the zip code of the customer address | 29260 | 49236 |  |
| CUSTTYPEID |  | CHAR(1) | It represents the customer type ID of customer in TPC E | F | S |  |
|  |  |  |  |  |  |  |
| invoiceID | invoice\_details | INTEGER(10) | It represents the invoice ID | 11 | 20 | CSV |
| prodID |  | INTEGER(10) | It represents the product ID | 1 | 2 |  |
| amt |  | VARCHAR(29) | it is the cost for that specific invoice ID | 329.94 | 2565.54 |  |
| qty |  | INTEGER(10) | It is the quantity of the products | 18 | 2 |  |
| discounted |  | BOOLEAN | It is a flag for representing it as discounted or not | 0 | 1 |  |
|  |  |  |  |  |  |  |
| invoiceID | invoice | INTEGER(10) | It represents the invoice ID |  |  |  |
| custID |  | INT(11) | It represents the customer ID of TPC E |  |  |  |
| salesDate |  | DATE | It is the date of the invoice created |  |  |  |
|  |  |  |  |  |  |  |
| PRODTYPEID | prod\_type | VARCHAR(45) | It represents the product type ID | 3 |  | CSV |
| TYPEDESCRIPTION |  | VARCHAR(45) | It represents the product description | Polishing Equipmen | Polishing Equipmen |  |
| BUID |  | CHAR(1) | Business unit ID | A | B |  |
|  |  |  |  |  |  |  |
| PRODID | product | INTEGER(11) | Its is the natural key of product table | 49 | 50 | CSV |
| DESCRIPTION |  | VARCHAR(45) | Gives the description of the product | Visage Flushing Chemicals |  |  |
| PRICE1 |  | VARCHAR(45) | It is the price that is not discounted |  |  |  |
| PRICE2 |  | VARCHAR(45) | Is the price that is after discount. |  |  |  |
| PRODTYPEID |  | VARCHAR(45) | Is the ID of the product |  |  |  |
| UNITCOST |  | DECIMAL(10,2) | is the unit cost of the product |  |  |  |
| SUPPLIERID |  | INTEGER(11) | Is the supplier ID of the the supplier |  |  |  |
|  |  |  |  |  |  |  |
| SUPPLIERID | supplier | INTEGER(11) | Is the supplier ID of the the supplier |  |  | CSV |
| NAME |  | VARCHAR(50) | The name of the supplier |  |  |  |
| ADDR1 |  | VARCHAR(50) | Is the first part of the address of the supplier |  |  |  |
| ADDR2 |  | VARCHAR(50) | Is the second part of the address of the supplier |  |  |  |
| CITY |  | VARCHAR(50) | It is city of the supplier |  |  |  |
| STATE |  | CHAR(2) | It is the state code of the supplier |  |  |  |
| ZIP |  | CHAR(5) | It is the zip code of the supplier |  |  |  |

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target |  |  |  |  |  | Source |  |  |  |  |  |
| Table Name | Column Name | Data Type | Table Type | SCD Type |  | Database Name | Table Name | Column Name | Data Type |  | Transformatio |
| Product | Product\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate key |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | ProductID | INTEGER(11) | Dimension |  |  | TPC E Excelsheet | product | PRODID | INTEGER(11) |  | Natural Key of product dimension |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWproduct | ProductID | INTEGER(11) |  | Natural Key of product dimension |
|  |  |  |  |  |  | PEC Excelsheet | PECproduct | prodid | INTEGER(11) |  | Natural Key of product dimension |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | ProductName | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | product | DESCRIPTION | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWproduct | Product Name | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECproduct | prodDescription | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | PRODTYPEID | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | prod\_type | PRODTYPEID | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWprod\_type | PRODTYPEID | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECproduct\_type | PRODTYPEID | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | TYPEDESCRIPTION | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | prod\_type | TYPEDESCRIPTION | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWprod\_type | TYPEDESCRIPTION | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECproduct\_type | TYPEDESCRIPTION | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | BUID | CHAR(1) | Dimension |  |  | TPC E Excelsheet | business\_unit |  | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWbusiness\_unit |  | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECWbusiness\_unit |  | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | dividionID | INTEGER(11) | Dimension | 4 |  |  |  |  |  |  | TPC E - 1, TPC W - 2 PEC - 3 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | BUAbbrev | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | business\_unit | ABBREV | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWbusiness\_unit | ABBREV | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECWbusiness\_unit | ABBREV | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | BUName | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | business\_unit | NAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWbusiness\_unit | NAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECWbusiness\_unit | NAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Supplier | Supplier\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate\_SK |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | SupplierID | INTEGER(11) | Dimension |  |  | TPC E Excelsheet | supplier | SUPPLIERID | INTEGER(11) |  | Natural Key of supplier dimension |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | NAME | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | supplier | NAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECproduct | supplierName | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWproduct | Supplier Name | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | ZIP | CHAR(5) | Dimension | 4 |  | TPC W Excelsheet | TPCWproduct | Supplier zipcode | CHAR(5) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | supplier | ZIP | CHAR(5) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | City | VARCHAR(50) | Dimension | 4 |  | TPC W Excelsheet | TPCWproduct | Supplier City | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | supplier | CITY | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | State | CHAR(2) | Dimension | 4 |  | TPC W Excelsheet | TPCWproduct | Supplier State | CHAR(2) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | supplier | STATE | CHAR(2) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Supplier Attn | VARCHAR(45) | Dimension | 4 |  | TPC E Excelsheet | supplier | ADDR1 | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Supplier Address | VARCHAR(45) | Dimension | 4 |  | TPC W Excelsheet | TPCWproduct | Supplier Address | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | ADDR2 | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Orderdate | OrderDate\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate Key of Order date dimension |
|  | order date | DATE | Dimension | 4 |  | PEC Excelsheet | PECsales\_new | orderDate | DATE |  |  |
|  | YEAR | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | Month | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | quarter | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | fiscalYear | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  | fiscalMonth | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  | fiscalQuarter | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Salesdate | SalesDate\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate Key of Sales date dimension |
|  | SalesDate | DATE | Dimension | 4 |  | TPC W Excelsheet | TPCWsales\_new | salesdate |  |  | Transformation using pentaho |
|  |  |  |  |  |  | PEC Excelsheet | PECsales\_new | salesdate |  |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | invoice | salesdate |  |  | Transformation using pentaho |
|  | YEAR | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | Month | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | quarter | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using pentaho |
|  | fiscalYear | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  | fiscalMonth | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  | fiscalQuarter | VARCHAR(45) | Dimension | 4 |  |  |  |  |  |  | Transformation using excel |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Customer | Customer\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate Key of Customer dimension |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | CUSTID | INTEGER(11) | Dimension |  |  | PEC Excelsheet | PECWcustomer | CUSTID | INTEGER(11) |  | Natural key of Customer dimension |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | CUSTID | INTEGER(11) |  | Natural key of Customer dimension |
|  |  |  |  |  |  | TPC E Excelsheet | customer | CUSTID | INTEGER(11) |  | Natural key of Customer dimension |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | NAME | VARCHAR(50) | Dimension | 6 |  | PEC Excelsheet | PECWcustomer | NAME | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | NAME | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | NAME | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | Address | VARCHAR(50) | Dimension | 1 |  | PEC Excelsheet | PECWcustomer | Address | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | Address | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | Addr1 | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  | Addr2 | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | CITY | VARCHAR(50) | Dimension | 4 |  | PEC Excelsheet | PECWcustomer | CITY | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | CITY | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | CITY | VARCHAR(50) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | STATE | CHAR(2) | Dimension | 4 |  | PEC Excelsheet | PECWcustomer | STATE | CHAR(2) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | STATE | CHAR(2) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | STATE | CHAR(2) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | ZIP | CHAR(45) | Dimension | 4 |  | PEC Excelsheet | PECWcustomer | ZIP | CHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | TPCWcustomer | ZIP | CHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | ZIP | CHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | CUSTTYPE | CHAR(1) | Dimension | 2 |  | PEC Excelsheet | customer\_type | CUSTTYPEID | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | customer\_type | CUSTTYPEID | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | CUSTTYPEID | CHAR(1) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | TYPENAME | VARCHAR(45) | Dimension | 2 |  | PEC Excelsheet | customer\_type | TYPENAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC W Excelsheet | customer\_type | TYPENAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  | TPC E Excelsheet | customer | TYPENAME | VARCHAR(45) |  | Transformation using pentaho |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | DivID | VARCHAR(45) | Dimension |  |  |  |  |  |  |  | TPC E - 1, TPC W - 2 PEC - 3 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| miscellaneous | misc\_SK | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Surrogate key of the miscellaneous dimension |
|  | ID | INTEGER(11) | Dimension |  |  |  |  |  |  |  | Natural Key |
|  | shipMethod | VARCHAR(45) | Dimension | 4 |  | PEC Excelsheet | PECsales\_new | shipMethod | VARCHAR(45) |  | Select shipping method from PECSales\_New |
|  | paymentMethod | VARCHAR(45) | Dimension | 4 |  | PEC Excelsheet | PECsales\_new | paymentMethod | VARCHAR(45) |  | Select payment method from PECSales\_New |
|  | orderMethod | VARCHAR(45) | Dimension | 4 |  | PEC Excelsheet | PECsales\_new | orderMethod | VARCHAR(45) |  | Select order method from PECSales\_New |
|  |  |  |  |  |  |  |  |  |  |  |  |
| salesfact | Product\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate key |
|  | Supplier\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate\_SK |
|  | OrderDate\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate Key of Order date dimension |
|  | SalesDate\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate Key of Sales date dimension |
|  | Customer\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate Key of Customer dimension |
|  | misc\_SK | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | Surrogate key of the miscellaneous dimension |
|  | invoice | INTEGER(11) | Fact |  |  |  |  |  | INTEGER(11) |  | degenerate dimension |
|  | amt | Decimal(10,2) | Fact | 4 |  |  | PECsales\_new |  | Decimal(10,2) |  | Transformation using pentaho |
|  |  |  |  |  |  |  | TPCWsales\_new |  |  |  | Transformation using pentaho |
|  | discounted | BOOLEAN | Fact | 4 |  |  | PECsales\_new |  | BOOLEAN |  | Transformation using pentaho |
|  |  |  |  |  |  |  | TPCWsales\_new |  |  |  | Transformation using pentaho |
|  | daysToComplete | INTEGER(11) | Fact | 4 |  |  |  |  | INTEGER(11) |  | Transformation using pentaho |
|  | UnitCost | Decimal(10,2) | Fact | 4 |  |  |  |  | Decimal(10,2) |  | Transformation using pentaho |
|  | Price | VARCHAR(45) | Fact | 4 |  |  |  |  | VARCHAR(45) |  | Transformation using pentaho |

# 

# V.  SQL Code – Tables & Constraints

# *Table creation • Constraints*

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema lab3

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema lab3

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `lab3` DEFAULT CHARACTER SET utf8 ;

USE `lab3` ;

-- -----------------------------------------------------

-- Table `lab3`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`customer` (

 `Customer\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `CUSTID` INT(11) NULL DEFAULT NULL COMMENT '',

 `NAME` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 `Addr1` VARCHAR(100) NULL DEFAULT NULL COMMENT '',

 `CITY` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 `STATE` CHAR(2) NULL DEFAULT NULL COMMENT '',

 `ZIP` CHAR(45) NULL DEFAULT NULL COMMENT '',

 `CUSTTYPE` CHAR(1) NULL DEFAULT NULL COMMENT '',

 `TYPENAME` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `DivID` INT(11) NULL DEFAULT NULL COMMENT '',

 `Addr2` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`Customer\_SK`)  COMMENT '',

 INDEX `custState` (`STATE` ASC)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 122

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`product` (

 `Product\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `ProductID` INT(11) NOT NULL COMMENT '',

 `ProductName` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `PRODTYPEID` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `TYPEDESCRIPTION` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `BUID` CHAR(1) NULL DEFAULT NULL COMMENT '',

 `divisionID` INT(11) NULL DEFAULT NULL COMMENT '',

 `BUAbbrev` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `BUName` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`Product\_SK`)  COMMENT '',

 INDEX `division` (`divisionID` ASC)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 150

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`saledate`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`saledate` (

 `SaleDate\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `salesDate` DATE NULL DEFAULT NULL COMMENT '',

 `Year` YEAR NULL DEFAULT NULL COMMENT '',

 `Month` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `quarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalYear` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalMonth` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalQuarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `day` INT(11) NULL DEFAULT NULL COMMENT '',

 `fiscalDay` INT(11) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`SaleDate\_SK`)  COMMENT '',

 INDEX `saleD` (`salesDate` ASC)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 2496

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`supplier`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`supplier` (

 `Supplier\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `supplierID` INT(11) NOT NULL COMMENT '',

 `Name` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 `ZIP` CHAR(5) NULL DEFAULT NULL COMMENT '',

 `City` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 `State` CHAR(2) NULL DEFAULT NULL COMMENT '',

 `SupplierAttn` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `SupplierAddr1` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 `SupplierAddr2` VARCHAR(50) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`Supplier\_SK`)  COMMENT '',

 INDEX `suppState` (`State` ASC)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`lostsalesfact`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`lostsalesfact` (

 `SaleDate\_SK` INT(11) NOT NULL COMMENT '',

 `amt` DECIMAL(8,2) NULL DEFAULT NULL COMMENT '',

 `Product\_SK` INT NOT NULL COMMENT '',

 `Supplier\_SK` INT NOT NULL COMMENT '',

 PRIMARY KEY (`SaleDate\_SK`, `Product\_SK`, `Supplier\_SK`)  COMMENT '',

 INDEX `product\_sk\_idx` (`Product\_SK` ASC)  COMMENT '',

 INDEX `supplier\_sk\_idx` (`Supplier\_SK` ASC)  COMMENT '',

 CONSTRAINT `product\_sk`

   FOREIGN KEY (`Product\_SK`)

   REFERENCES `lab3`.`product` (`Product\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `saledate\_sk`

   FOREIGN KEY (`SaleDate\_SK`)

   REFERENCES `lab3`.`saledate` (`SaleDate\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `supplier\_sk`

   FOREIGN KEY (`Supplier\_SK`)

   REFERENCES `lab3`.`supplier` (`Supplier\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`miscellaneous`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`miscellaneous` (

 `misc\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `shipMethod` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `paymentMethod` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `orderMethod` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`misc\_SK`)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 38

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`orderdate`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`orderdate` (

 `OrderDate\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `orderDate` DATE NULL DEFAULT NULL COMMENT '',

 `Year` YEAR NULL DEFAULT NULL COMMENT '',

 `Month` INT(11) NULL DEFAULT NULL COMMENT '',

 `Quarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `Fiscalyear` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalMonth` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `FiscalQuarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `day` INT(11) NULL DEFAULT NULL COMMENT '',

 `fiscalDay` INT(11) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`OrderDate\_SK`)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 2159

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`ordermonth`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`ordermonth` (

 `OrderMonth\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `Year` YEAR NULL DEFAULT NULL COMMENT '',

 `Month` INT(11) NULL DEFAULT NULL COMMENT '',

 `Quarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `Fiscalyear` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalMonth` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `FiscalQuarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`OrderMonth\_SK`)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 128

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`salemonth`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`salemonth` (

 `SaleMonth\_SK` INT(11) NOT NULL AUTO\_INCREMENT COMMENT '',

 `Year` YEAR NULL DEFAULT NULL COMMENT '',

 `Month` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `quarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalYear` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalMonth` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `fiscalQuarter` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 PRIMARY KEY (`SaleMonth\_SK`)  COMMENT '',

 INDEX `saleM` (`Month` ASC)  COMMENT '')

ENGINE = InnoDB

AUTO\_INCREMENT = 85

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`salesfact`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`salesfact` (

 `InvoiceiD` VARCHAR(45) NOT NULL COMMENT '',

 `Customer\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `OrderDate\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `SaleDate\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `Product\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `Supplier\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `misc\_SK` INT(11) NOT NULL DEFAULT '-1' COMMENT '',

 `amt` DECIMAL(10,2) NULL DEFAULT NULL COMMENT '',

 `qty` DECIMAL(8,2) NULL DEFAULT NULL COMMENT '',

 `discounted` BIT(1) NULL DEFAULT NULL COMMENT '',

 `daysToComplete` INT(11) NULL DEFAULT NULL COMMENT '',

 `UnitCost` DECIMAL(10,2) NULL DEFAULT NULL COMMENT '',

 `Price` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `internalSale` BIT(1) NULL DEFAULT b'1' COMMENT '',

 PRIMARY KEY (`Customer\_SK`, `OrderDate\_SK`, `SaleDate\_SK`, `Product\_SK`, `Supplier\_SK`, `misc\_SK`, `InvoiceiD`)  COMMENT '',

 INDEX `discounted` (`discounted` ASC, `internalSale` ASC)  COMMENT '',

 INDEX `OrderDate\_Sk\_idx` (`OrderDate\_SK` ASC)  COMMENT '',

 INDEX `SaleDate\_SK\_idx` (`SaleDate\_SK` ASC)  COMMENT '',

 INDEX `Product\_SK\_idx` (`Product\_SK` ASC)  COMMENT '',

 INDEX `Supplier\_SK\_idx` (`Supplier\_SK` ASC)  COMMENT '',

 INDEX `misc\_SK\_idx` (`misc\_SK` ASC)  COMMENT '',

 CONSTRAINT `Customer\_SK`

   FOREIGN KEY (`Customer\_SK`)

   REFERENCES `lab3`.`customer` (`Customer\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `OrderDate\_Sk`

   FOREIGN KEY (`OrderDate\_SK`)

   REFERENCES `lab3`.`orderdate` (`OrderDate\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `SaleDate\_SK`

   FOREIGN KEY (`SaleDate\_SK`)

   REFERENCES `lab3`.`saledate` (`SaleDate\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `Product\_SK`

   FOREIGN KEY (`Product\_SK`)

   REFERENCES `lab3`.`product` (`Product\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `Supplier\_SK`

   FOREIGN KEY (`Supplier\_SK`)

   REFERENCES `lab3`.`supplier` (`Supplier\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `misc\_SK`

   FOREIGN KEY (`misc\_SK`)

   REFERENCES `lab3`.`miscellaneous` (`misc\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `lab3`.`salesfactmonth`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `lab3`.`salesfactmonth` (

 `Customer\_SK` INT(11) NOT NULL COMMENT '',

 `OrderMonth\_SK` INT(11) NOT NULL COMMENT '',

 `SaleMonth\_SK` INT(11) NOT NULL COMMENT '',

 `Product\_SK` INT(11) NOT NULL COMMENT '',

 `Supplier\_SK` INT(11) NOT NULL COMMENT '',

 `misc\_SK` INT(11) NOT NULL COMMENT '',

 `amt` DECIMAL(10,2) NULL DEFAULT NULL COMMENT '',

 `qty` DECIMAL(8,2) NULL DEFAULT NULL COMMENT '',

 `discounted` BIT(1) NOT NULL COMMENT '',

 `avgDaysToComplete` INT(11) NULL DEFAULT NULL COMMENT '',

 `UnitCost` DECIMAL(10,2) NULL DEFAULT NULL COMMENT '',

 `Price` VARCHAR(45) NULL DEFAULT NULL COMMENT '',

 `internalSale` BIT(1) NOT NULL COMMENT '',

 PRIMARY KEY (`discounted`, `internalSale`, `Customer\_SK`, `OrderMonth\_SK`, `SaleMonth\_SK`, `Product\_SK`, `Supplier\_SK`, `misc\_SK`)  COMMENT '',

 INDEX `customer\_sk\_idx` (`Customer\_SK` ASC)  COMMENT '',

 INDEX `OrderMonth\_SK\_idx` (`OrderMonth\_SK` ASC)  COMMENT '',

 INDEX `SaleMonth\_SK\_idx` (`SaleMonth\_SK` ASC)  COMMENT '',

 INDEX `Product\_SK\_idx` (`Product\_SK` ASC)  COMMENT '',

 INDEX `Supplier\_Sk\_idx` (`Supplier\_SK` ASC)  COMMENT '',

 INDEX `misc\_SK\_idx` (`misc\_SK` ASC)  COMMENT '',

 CONSTRAINT `customer\_sk`

   FOREIGN KEY (`Customer\_SK`)

   REFERENCES `lab3`.`customer` (`Customer\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `OrderMonth\_SK`

   FOREIGN KEY (`OrderMonth\_SK`)

   REFERENCES `lab3`.`ordermonth` (`OrderMonth\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `SaleMonth\_SK`

   FOREIGN KEY (`SaleMonth\_SK`)

   REFERENCES `lab3`.`salemonth` (`SaleMonth\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `Product\_SK`

   FOREIGN KEY (`Product\_SK`)

   REFERENCES `lab3`.`product` (`Product\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `Supplier\_Sk`

   FOREIGN KEY (`Supplier\_SK`)

   REFERENCES `lab3`.`supplier` (`Supplier\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION,

 CONSTRAINT `misc\_SK`

   FOREIGN KEY (`misc\_SK`)

   REFERENCES `lab3`.`miscellaneous` (`misc\_SK`)

   ON DELETE NO ACTION

   ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SET SQL\_MODE=@OLD\_SQL\_MODE;

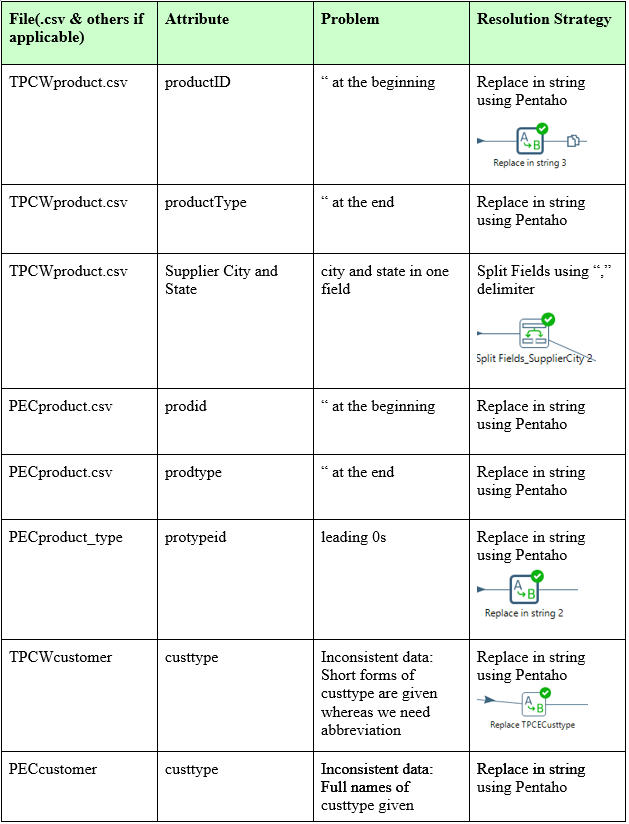
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

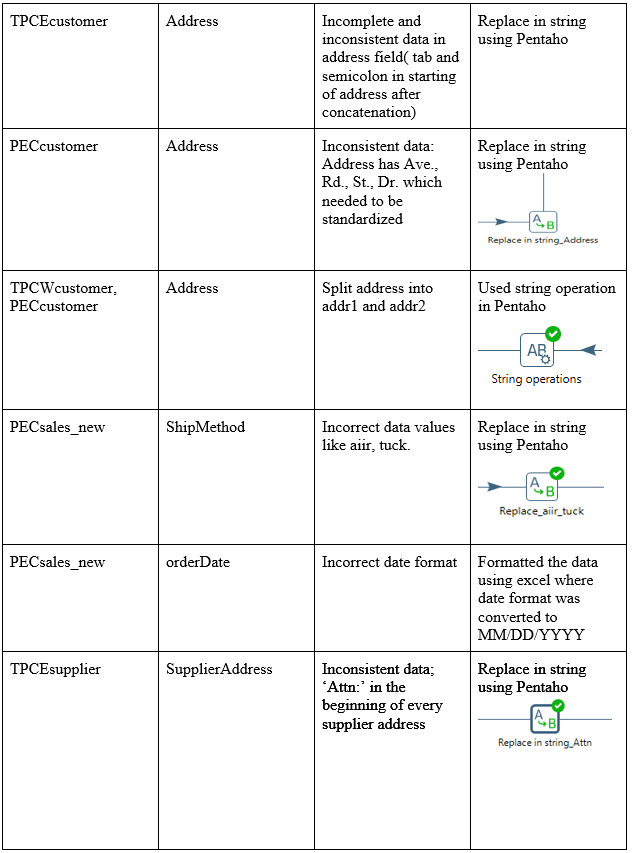
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

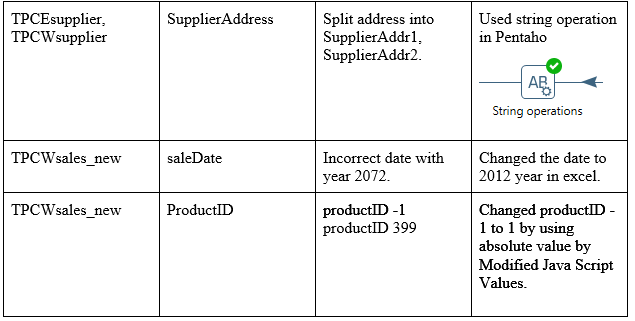
# VI.  Data Staging Activities - ETL

Which attributes were cleansed • Cleansing process – manual processes or code • Data problems identified – by extract file and field • Problem resolution descriptions

## 1. Data Cleansing

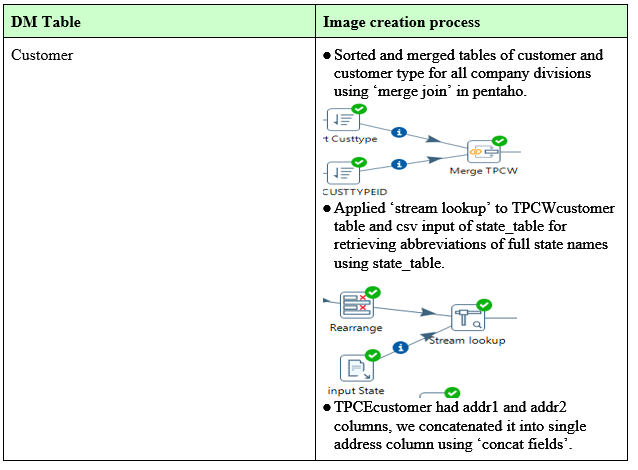


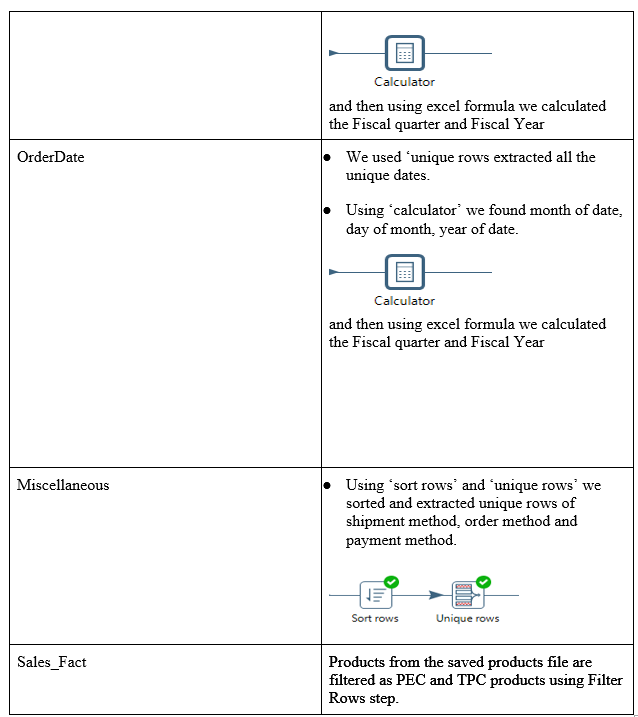
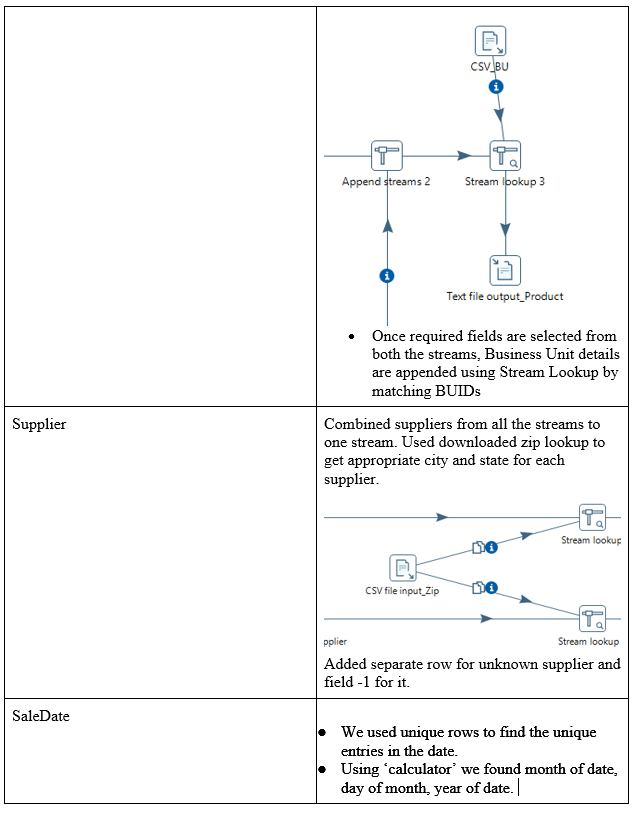
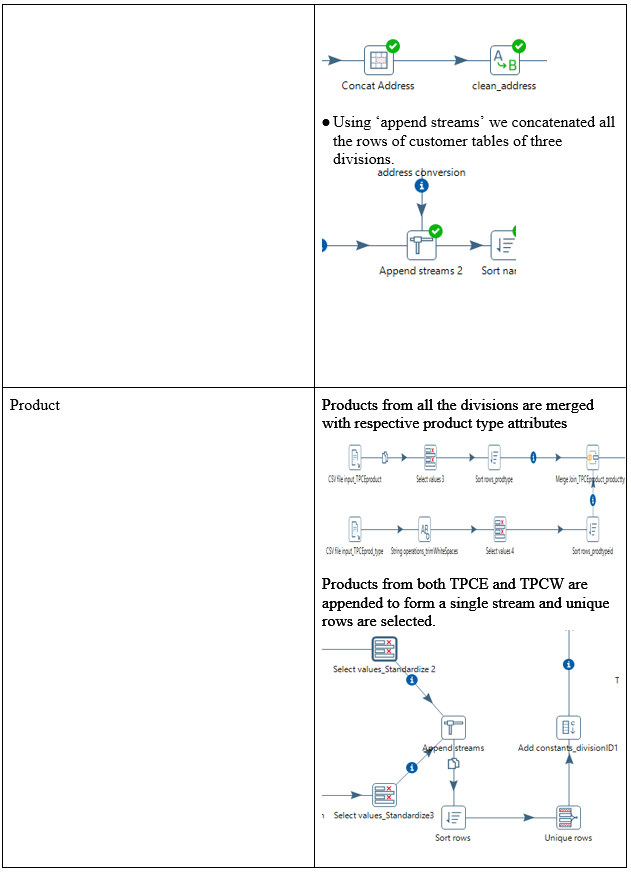


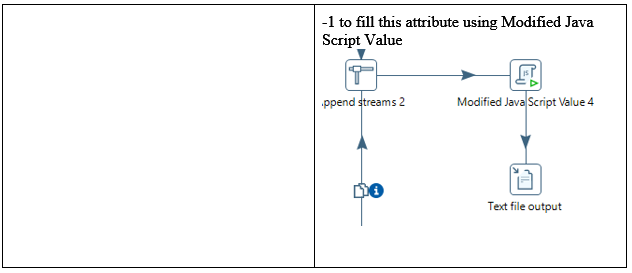
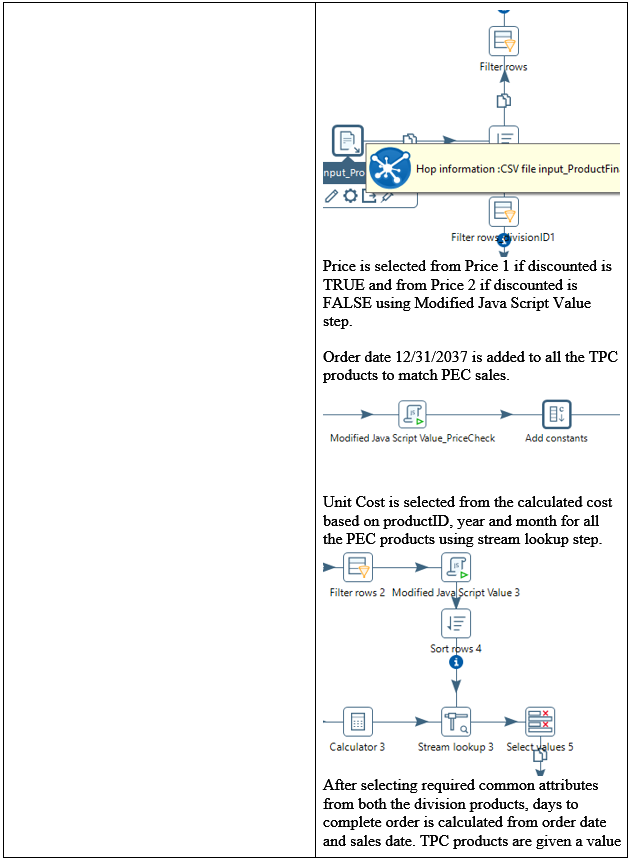


## 2. Data Transformation

Process details • Code







## 3. Table Population

Process details • Code included

|  |  |
| --- | --- |
| **DM Table** | **Table Population Process** (attach code) |
| Product | Final saved product file is read in Pentaho to filter the required fields and is loaded into database by using Table Output feature. An integer Surrogate Key is generated by the database automatically. Product\_SK for not found products are given -1 value. |
| Supplier | Concatenated supplier file is loaded into database using Table Output step in Pentaho. All the unknown suppliers are give -1 surrogate key |
| Customer | Final customer. -1 surrogate key is added for unknown customers |
| Order Date | Unique order dates retrieved from all the sales are added to database using Pentaho. Default or null date entry is given a surrogate key -1 with date 12/31/2037 ( oldest date possible in MySQL is 01/2038). |
| Sales Date | Unique saledates identified from all the sales facts are added to the table with automated surrogate key generation |
| Sales Fact | Sales fact file is read into Pentaho using CSV file Input step. Values for each dimension table are matched using respective natural keys and the surrogate keys are retrieved using Stream Lookup step in Pentaho. The required fields are selected using Select Values and loaded to database using Table Output step. |
| Miscellaneous | Miscellaneous dimension file saved from retrieved unique rows from PEC sales is used to populate Miscellaneous dimension. Natural keys are manually allocated in Excel and the final table in loaded into database using Pentaho. Miscellaneous natural key for default or null values is given as -1. |

# VII.  End User Applications

## 1. Queries

|  |
| --- |
| **User Question/Reporting Need** |
| /\* Customer Ranking for Each Product \*/ |
| **SQL Code** |
| Select a.Product\_SK, a.Customer\_SK, a.CustomerTotal, count(b.CustomerTotal) Rank  from  (select Product\_SK, Customer\_SK, sum(amt) CustomerTotal  from salesfact group by Product\_Sk,Customer\_SK) a  join  (select Product\_SK, Customer\_SK, sum(amt)  CustomerTotal  from salesfact group by Product\_Sk,Customer\_SK) b  using (Product\_SK) where a.CustomerTotal <= b.CustomerTotal  group by a.Product\_SK, a.Customer\_SK  order by 1,4,2;  Sample output:  https://lh5.googleusercontent.com/uAVdSD1kyWHz65gvlheyAy4X6zzcWenLo0qBw0M4aJdsyVhlhBLvn7Py4GmXDaJQLaHYdnQN74Zp6WPOkaHAr6yOKgwkz6QQ4ICs_v6A6xIkHK0Qa63zI0fCMpSNVBZtsRpG-6jk |

|  |
| --- |
| **User Question/Reporting Need** |
| /\* Finding product and customer total for every product and customer.\*/ |
| **SQL Code** |
| select a.Product\_SK, a.Customer\_SK, CustomerTotal, ProductTptal  from  (select Product\_SK, Customer\_SK, sum(amt) CustomerTotal  from salesfact group by Product\_SK, Customer\_SK) a  join  (select Product\_SK, sum(amt) ProductTptal  from salesfact group by Product\_SK) b  using (Product\_SK)  order by 1,2,3;  Sample output:  https://lh6.googleusercontent.com/arBKt7jNm07YvCACNPIaiHYnaDVwwG7wuXJaYolcqZAWhiqfFFENBomABREkA-WbV69Ik60qAOz9r_RNYYf7qzi0NNe6_ESOuYvy0TVOu8YviJLYYeUZE2ByNeOlurh3RVekMcvX |

|  |
| --- |
| **User Question/Reporting Need** |
| /\* Comparision of Sales quarter-to-quarter based on calendar year  \*/ |
| **SQL Code** |
| select a.quarter q1, a.amount amt1, b.quarter q2, b.amount amt2  from  (select Quarter, sum(amt) amount  from saledate d join salesfact f on d.SaleDate\_SK = f.OrderDate\_SK  group by quarter) a  join  (select Quarter, sum(amt) amount  from saledate d join salesfact f on d.SaleDate\_SK = f.OrderDate\_SK  group by quarter) b  on a.Quarter != b.Quarter  group by q1, q2;  Sample output:  https://lh5.googleusercontent.com/O8E85hJ8gCJ1MLvt6ja23g0nZ_6Ry7yQpfKm_lagrqCl7N-vkT-hx-CT7_jC4eqv7jAJrLWTjVPEuqsRaV_-agM0QeC7KUCSj0l6xlHTadokTNhkn4c0xe7-hAG6IF_pim6YxXE- |

## 2. A View

* We created a view ‘SalesDescription’ using following SQL query:

Create view SalesDescription AS

select customer.CUSTTYPE, customer.STATE, product.TYPEDESCRIPTION, product.BUID

from customer, product

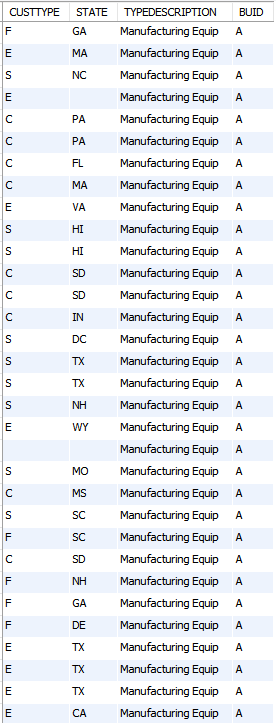
where productID = 4;

* Using this view company can obtain data of sales by type of customer, by state, by product type and by business unit associated with specific product ID.

We can view data from this view using following select query:

select \* from SalesDescription;

* For this view we got the following output:



3. Aggregated Data Marts

# SalesFactMonth - Shrunken Dimension

# This dimension can be used to extract monthly sales facts. DaysToComplete attribute is transformed to avgDaysToComplete with the mean value. All the other attributes are either summed or left as they are. New table saleMonth is shrunken from saleDate.

# Aggregation Method used: shrunken dimension

# CODE:

# insert into salemonth(Year, Month, quarter,Fiscalyear, fiscalMonth, fiscalQuarter)

# select distinct Year,Month,quarter,Fiscalyear,fiscalMonth,FiscalQuarter from saledate ;

# insert into salesfactmonth(Customer\_SK,OrderDate\_SK,SaleMonth\_SK,Product\_SK,Supplier\_SK,misc\_SK,amt,qty,discounted,avgDaysToComplete,UnitCost,Price,internalSale)

# select Customer\_SK,OrderDate\_SK,k.saleMonth\_SK,Product\_SK,Supplier\_SK,misc\_SK,sum(amt),sum(qty),discounted,avg(daysToComplete),avg(UnitCost),sum(Price),internalSale

# from salesfact

# join saledate using (saledate\_sk)

# join salemonth k using (Month, Year)

# group by Customer\_SK,OrderDate\_SK,k.saleMonth\_SK,Product\_SK,Supplier\_SK,misc\_SK,discounted,internalSale ;

**Queries:**

**To find total amount of sales of products supplied by various suppliers in a year**

select Name, Year, Month, amt

from salesfactmonth

join Supplier using(Supplier\_SK)

join salemonth using (salemonth\_sk)

where Year = 2005 and supplier\_SK<>-1;

**Use Case:**

Query can be used by analysts to analyse supplier product sales in a month or a year.

**Show sales  from  one  division  to  another**

select Year, Month, ProductName, qty

from salesfactMonth

join salemonth using(saleMonth\_sk)

join product using(product\_sk)

where internalSale=0;

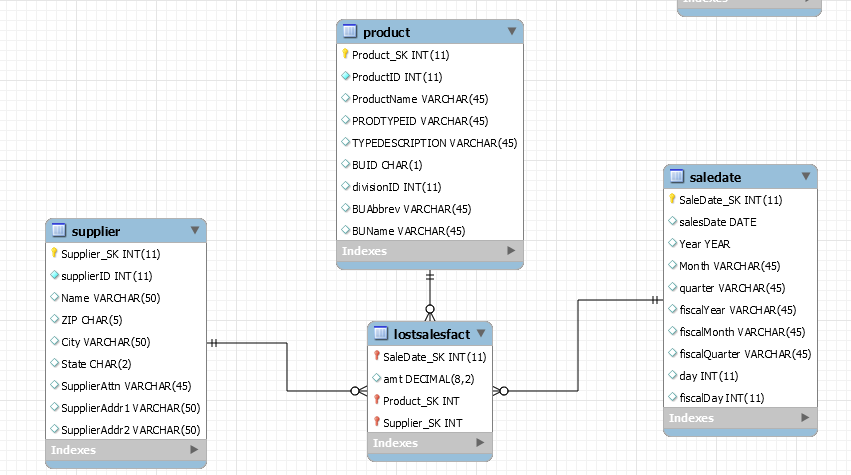
Query can be used to know the  internal sales of the organization.

**2. Aggregation Method - Lost Dimension**

lostsalesfact –

insert into lostsalesfact(SaleDate\_SK, Product\_SK, Supplier\_SK, amt)

# select distinct Year,Month,quarter,Fiscalyear,fiscalMonth,FiscalQuarter from saledate ;



**Query:** Total Cost of products supplied by a supplier in a year.

Use case- Query can be used by analysts to analyze highest revenue obtained by suppliers.

select product\_SK, supplier\_SK, amt, year

from lostsalesfact

join supplier using (supplier\_SK)

join product using (product\_SK)

join SaleDate using (salesdate\_SK)

group by year;

**Query:**

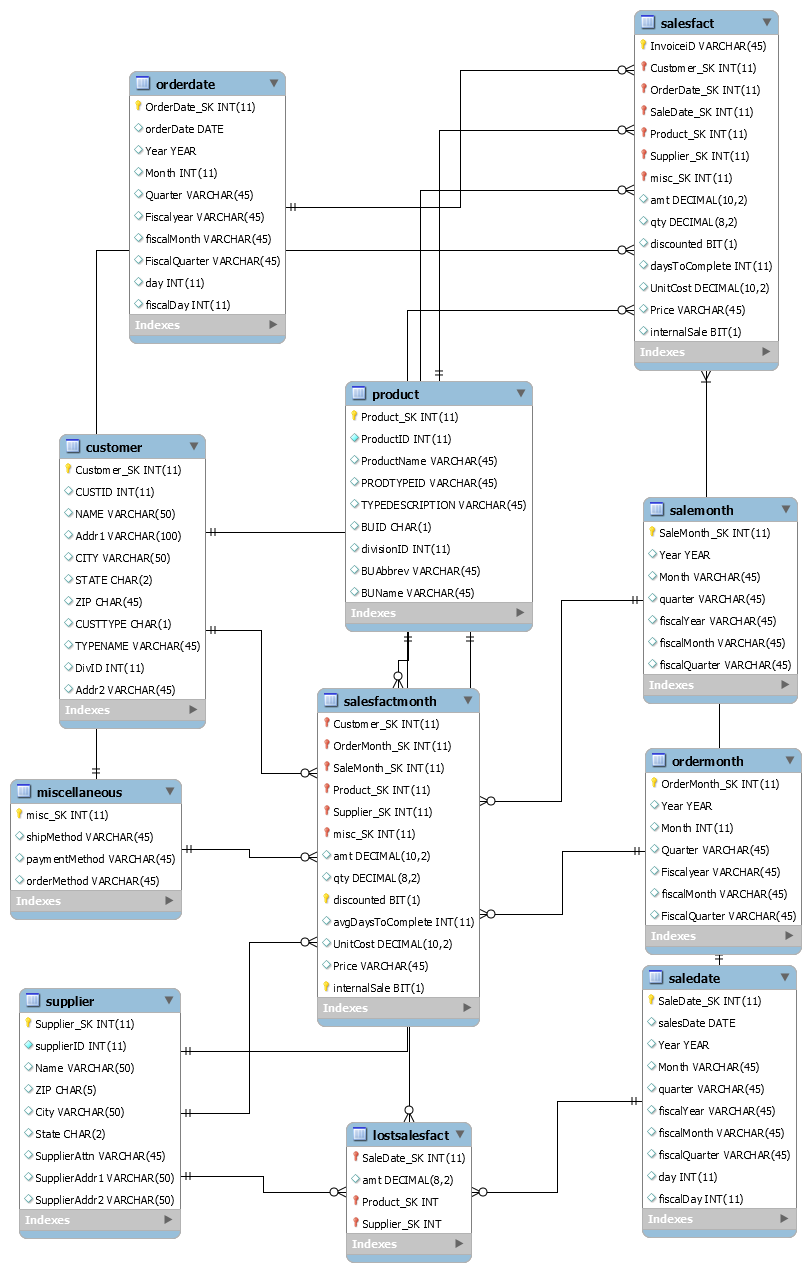
Use Case - With this query we can find out cost of products sold in each day using which we can find daily profit.

Select amt, SalesDate

From lostsalesfact

Join saledate using saledate\_SK

Group by saledate;



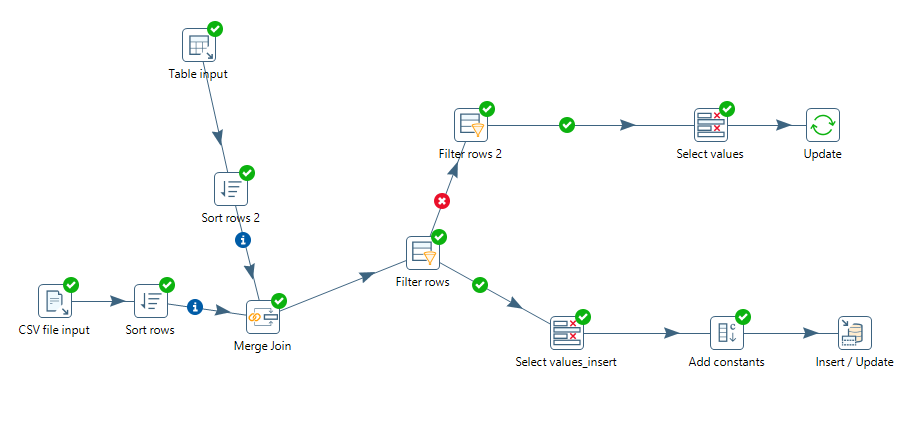
**VIII. Handling Slowly Changing Dimensions (SCD)**

The three SCDs implemented in the design are type 1 on customer address (addr1, addr2, city, state, zip), type 2 on customer type information (custtype and typename) and type 6 on customer name. The SCDs are implemented by Pentaho. The incoming customer data stream is used in pentaho to identify records needed to be updated and inserted.

**SCD type 1**:

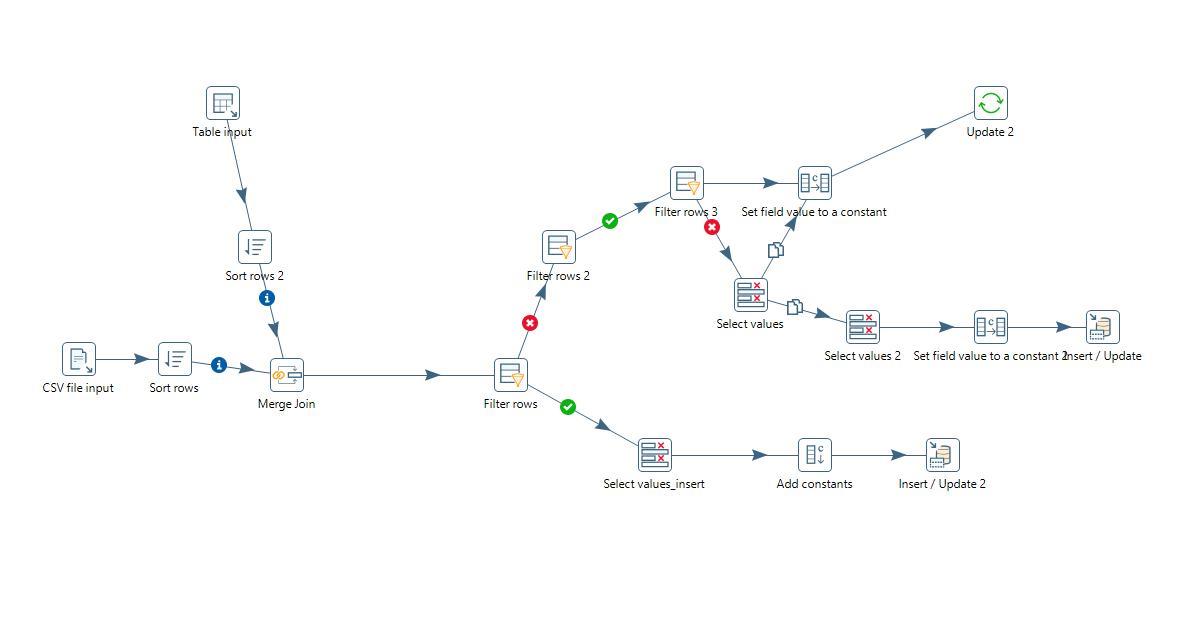
The slowly changing dimension type 1 is used to update customer address information. The attributes addr1, addr2, city, state, zip are replaced on update. The customer information is matched using customer ID and the division the customer belongs to.

Whenever there is any change in customer’s address company does not need their old address, as type 1 SCD does not maintain history we implemented it on address, city, state, zip fields of customer dimension.



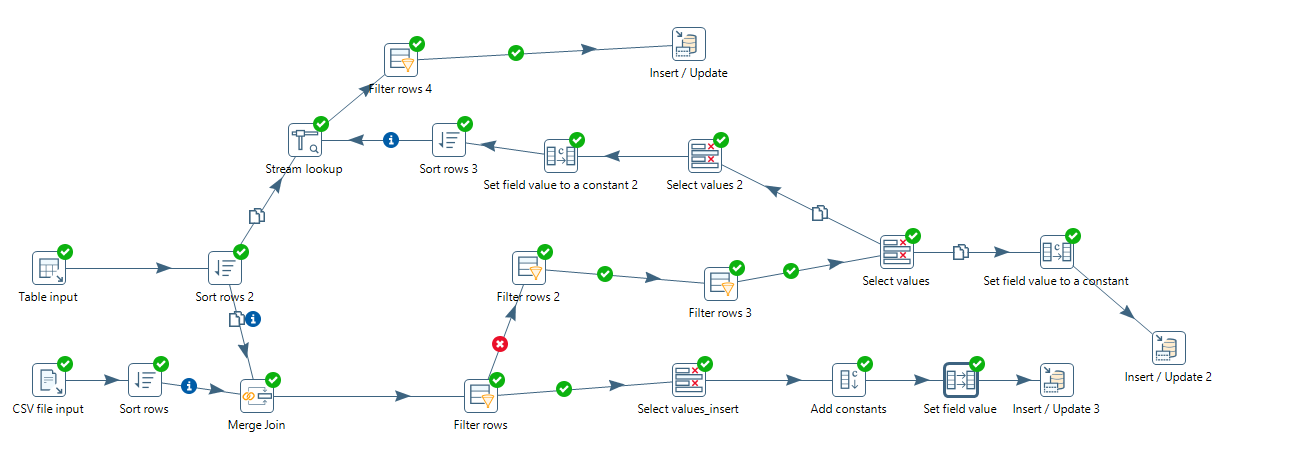
**SCD type 2:**

The type 2 SCD is implemented on Customer Type information. Since the customers usually do not change the type in this case, a change in the customer type information is needed to be stored. So maintaining a new record for every change helps in monitoring previous sales by the customer before changing the type. A change in customer type and customer type name updates the existing end dates of the customers and adds a new row to represent current version of the data.



**SCD type 6:**

Customer Name is considered a type 6 SCD in this senario. Wheneever a customer name is changed, we need to record all possible information. Since it is a unusual situation, the frequency of the update happening is rare which makes it worth the complexity required to follow type 6. Upon a change in name other than the current name, all the name values throughout the table gets updated to the new one and a completely new record is added to represent the current snapshot. Updating the name also causes changes in end date of the currently active row.

****

(We have attached out test file and results in SCD folder.)

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

* **Implementation strategy:**

As shipping company supports more than one supplier, we have many to many relationship between supplier and shipping company.  To implement many to many relationship between two entities we have three methods:

1) Boolean column method

2) Multiple column method

3) Bridge table method.

We referred Kimball’s design tip # 166 about potential bridges.

Thus, for implementing this many to many relation, we propose a bridge table method. We have two entities here, supplier and shipping company. We added a bridge table between these two tables and joined them using their respective surrogate keys. As according to Kimball bridge table is a powerful and scalable implementation approach for handling multivalued dimensions.

* **ER diagram:**

