The Product Company

~ Final Data Mart Development Report ~

Team # ____1_____

Team Members

___John Warner_____

___Ellie Parobek_____

___Ryan Garcia_____

11/16/2020

ISTE-DW Data Warehousing

~ Table of Contents ~

I. Data Mart Design Definition

1. Universe of Discourse

This data mart will cover the performance measures relevant to the sale of different products made across all three divisions.

2. Information Package

Process Name: Invoicing

Grain: Individual invoice records

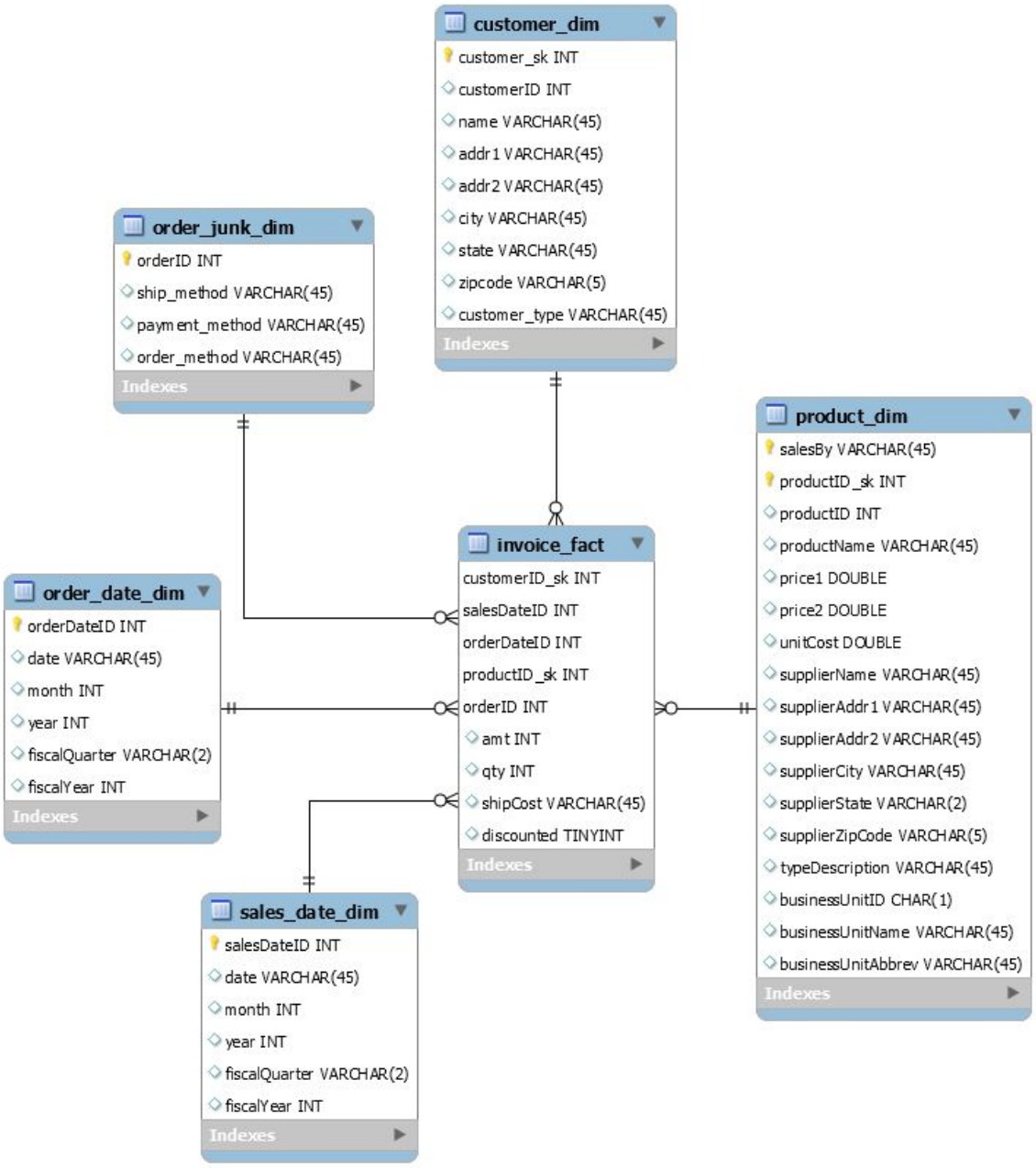
Sales Date	Order Date	Product	Customer	Orders	
Sale Date	Order Date	Product Name	Customer Name	Order Method	
Month	Month	Price1	Addr1	Ship Method	
Year	Year	Price2	Addr2	Payment Method	
		Unit Cost	City		
		Supplier Name	State		
		Supplier Address	Zipcode		
		Supplier City	Customer Type		
		Supplier State			
		Supplier Zipcode			
		Type Description			
		Business Unit Name			
		Business Unit Abbreviation			

Facts: Sales amount, quantity, shipping cost

3. Entity Definitions

Entity	Entity Definition (*genus differentia*)
invoice_fact	A fact table that tracks all the invoices related to products sold.
customer_dim	A dimensional table that holds a customer's personal information.
product_dim	A dimensional table that has information on a particular product.
order_junk_dim	A junk dimensional table that contains order information for an invoice such as order method, payment method and ship method.
order_date_dim	A dimensional table that holds records for all possible dates a product might have been ordered on.
sales_date_dim	A dimensional table that holds records for all possible dates a product might have been sold on.

II. Dimensional Model



III. Data Staging: ETL – Data Extract File Definitions

All loaded files are located under '434Final_Team01_2201/Files Loaded'.

businessUnit.csv: Loaded into the business unit dimension originally, but was later removed and combined with the product dimension (see section X.Appendix - Product Dim).

customer.csv: Contains all customer dimensional data.

invoice.csv: Contains all invoice fact table data.

orderDate.csv: Contains all order date dimensional data.

orderJunk.csv: Contains all order junk dimensional data.

PECproduct.csv: Contains the PEC related product dimensional data.

salesDate.csv: Contains all sales date dimensional data.

TPCEproduct.csv: Contains the TPCE related product dimensional data.

TPCWproduct.csv: Contains the TPCW related product dimensional data.

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

Follow the same format as indicated in "The Data Warehouse ETL Toolkit" by Kimball & Caserta, Fig. 3.1 on page 60. This is available on Books 24x7. The table should be in alphabetical order table name and column name.

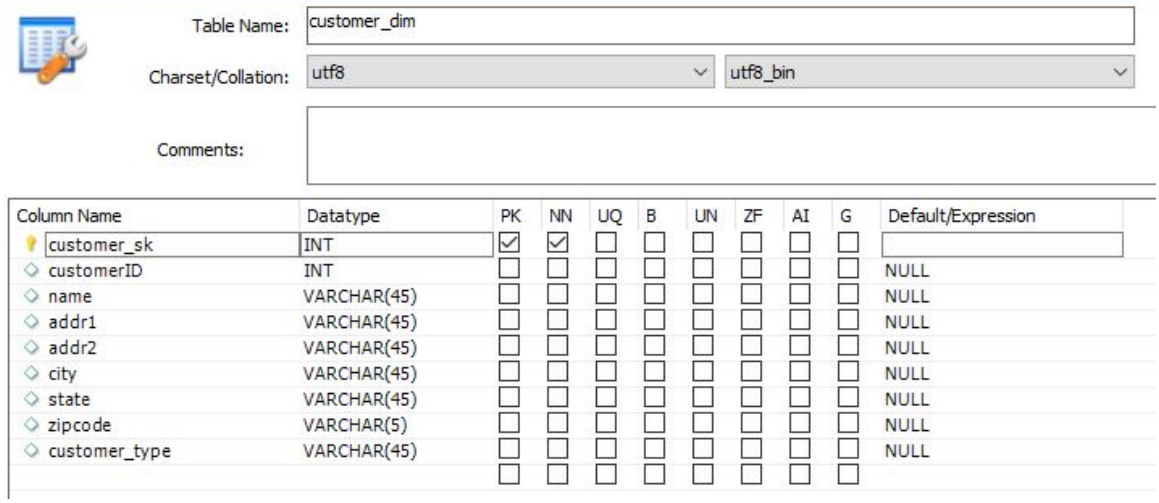
Target					Source				Transformation
Table Name	Column Name	Data Type	Table Type	SCD Type	Database Name	Table Name	Column Name	Data Type	
customer_dim	customer_sk	INT	Dimension	1	N/A	N/A	N/A	NUMBER	Surrogate key.
customer_dim	customerID	INT	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	CUSTID, custID	NUMBER	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	name	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	NAME, name	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	addr1	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	ADDR1, addr	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	addr1	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	ADDR1, addr	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	city	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	CITY, city	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	state	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	STATE, state	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	zipcode	VARCHAR(5)	Dimension	1	TPCE, TPCW, PEC	customer, TPCWcustomer, PECcustomer	ZIP, zip	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
customer_dim	customer_type	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	customer_type, TPCWcustomer_type, PECcustomer_type	TYPENAME	VARCHAR()	Formatted with TPCWcustomer.ktr/PECcustomer.ktr/CustomerMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE customer_dim' for each divisions' customer csv file or using Workbench Import Wizard
product_dim	productID_sk	INT	Dimension	1	N/A	N/A	N/A	NUMBER	Surrogate key.
product_dim	salesBy	VARCHAR(45)	Dimension	1	N/A	N/A	N/A	VARCHAR()	Manually added based on which division (PEC/TPCW/TPCE) the product information came from
product_dim	productID	INT	Dimension	1	TPCE, TPCW, PEC	product, TPCWproduct, PECproduct	PRODID, prodid	NUMBER	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	productName	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	product, TPCWproduct, PECproduct	DESCRIPTION, prodDescription	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	price1	DOUBLE	Dimension	1	TPCE, TPCW, PEC	product, TPCWproduct, PECproduct	PRICE1, price1	NUMBER	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	price2	DOUBLE	Dimension	1	TPCE, TPCW, PEC	product, TPCWproduct, PECproduct	PRICE2, price2	NUMBER	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	unitCost	DOUBLE	Dimension	1	TPCE, TPCW, PEC	product, TPCWproduct, PECproduct	UNITCOST, unitCost	NUMBER	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	supplierName	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	supplier, TPCWproduct, PECproduct	NAME, supplierName	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard

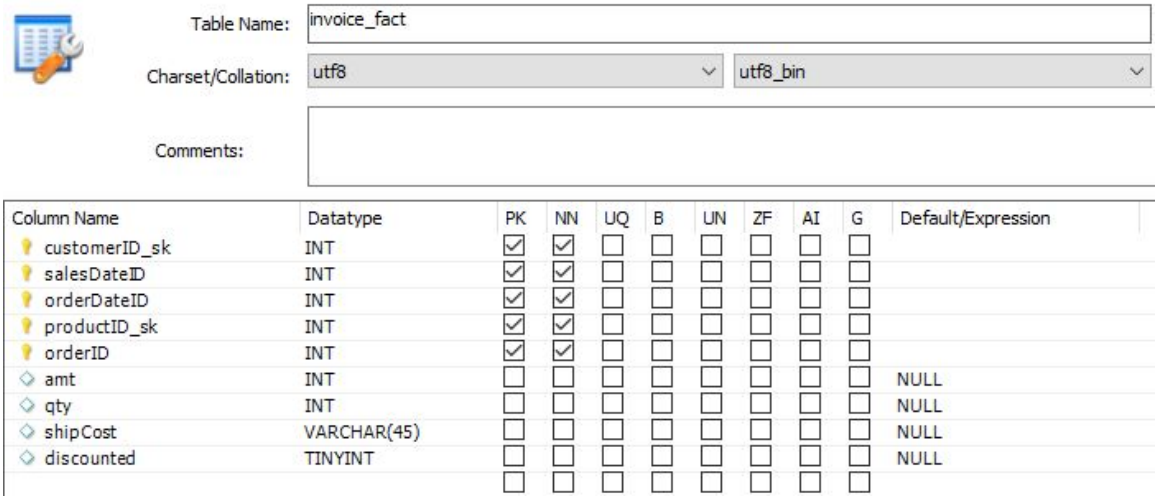
product_dim	supplierAddr1	VARCHAR(45)	Dimension	1	TPCE	supplier	ADDR1	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	supplierAddr2	VARCHAR(45)	Dimension	1	TPCE	supplier	ADDR2	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	supplierCity	VARCHAR(45)	Dimension	1	TPCE	supplier	CITY	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	supplierState	VARCHAR(2)	Dimension	1	TPCE	supplier	STATE	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	supplierZipCode	VARCHAR(45)	Dimension	1	TPCE	supplier	ZIP	VARCHAR()	Formatted with PECProduct.ktr/TPCWProduct.ktr/ProductMerge.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	typeDescription	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	prod_type, TPCWproduct_type, PECproduct_type	TYPEDESCRIPTION	VARCHAR()	Formatted with PECproducttype.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for each division's product csv file or using Workbench Import Wizard
product_dim	businessUnitID	CHAR(1)	Dimension	1	TPCE, TPCW, PEC	business_unit, TCPWbusiness_unit PECbusiness_unit	BUID	VARCHAR()	Formatted with PECbusinessunit.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE product_dim' for businessUnit.csv or using Workbench Import Wizard
product_dim	businessUnitName	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	business_unit, TCPWbusiness_unit PECbusiness_unit	NAME	VARCHAR()	Formatted with PECbusinessunit.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for businessUnit.csv or using Workbench Import Wizard
product_dim	businessUnitAbbrev	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	business_unit, TCPWbusiness_unit PECbusiness_unit	ABBREV	VARCHAR()	Formatted with PECbusinessunit.ktr, then loaded into database using 'LOAD DATA INFILE 'filename.csv' INTO TABLE product_dim' for businessUnit.csv or using Workbench Import Wizard
order_junk_dim	orderID	INT	Junk Dimension	1	N/A	N/A	N/A	N/A	Primary key.
order_junk_dim	order_method	VARCHAR(45)	Junk Dimension	1	PEC	PECinvoice	orderMethod	VARCHAR()	Used Google Sheets to take order_method from invoice data and create csv for junk dimension. Created dimension in database and used 'LOAD DATA INFILE *'filename.csv'* INTO TABLE order_junk_dim' to populate junk dimension or using Workbench Import Wizard
order_junk_dim	ship_method	VARCHAR(45)	Junk Dimension	1	PEC	PECinvoice	shipMethod	VARCHAR()	Used Google Sheets to take ship_method from invoice data and create csv for junk dimension. Created dimension in database and used 'LOAD DATA INFILE *'filename.csv'* INTO TABLE order_junk_dim' to populate junk dimension or using Workbench Import Wizard
order_junk_dim	payment_method	VARCHAR(45)	Junk Dimension	1	PEC	PECinvoice	paymentMethod	VARCHAR()	Used Google Sheets to take payment_method from invoice data and create csv for junk dimension. Created dimension in database and used 'LOAD DATA INFILE *'filename.csv'* INTO TABLE order_junk_dim' to populate junk dimension or using Workbench Import Wizard
sales_date_dim	salesDateID	INT	Dimension	1	N/A	N/A	N/A	NUMBER	Primary key.
sales_date_dim	date	VARCHAR(45)	Dimension	1	TPCE, TPCW, PEC	invoice, TCPWinvoice, PECinvoice	salesDate	DATE	Taken from invoice salesDate and placed into seperate csv file. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE sales_date_dim' or using Workbench Import Wizard
sales_date_dim	month	INT	Dimension	1	TPCE, TPCW, PEC	invoice, TCPWinvoice, PECinvoice	salesDate	DATE	Used Google Sheets formula: =MONTH(B3) to extract month from salesDate. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE sales_date_dim' or using Workbench Import Wizard
sales_date_dim	year	INT	Dimension	1	TPCE, TPCW, PEC	invoice, TCPWinvoice, PECinvoice	salesDate	DATE	Used Google Sheets formula: =YEAR(B3) to extract year from salesDate. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE sales_date_dim' or using Workbench Import Wizard
sales_date_dim	fiscalQuarter	VARCHAR(2)	Dimension	1	TPCE, TPCW, PEC	invoice, TCPWinvoice, PECinvoice	salesDate	DATE	Google Sheets formula: =IF(AND(MONTH(B3)>=1, MONTH(B3)<4), "Q4", IF(AND(MONTH(B3)>=4, MONTH(B3)<7), "Q1", IF(AND(MONTH(B3)>=7, MONTH(B3)<10), "Q2", IF(AND(MONTH(B3)>=10, MONTH(B3)<13), "Q3", "n/a"))))

sales_date_dim	fiscalYear	INT	Dimension	1	TPCE, TPCW, PEC	invoice, TCPWinvoice, PECinvoice	salesDate	DATE	Used Google Sheets formula to extract fiscalYear from salesDate: =IF(MONTH(B3)>3,YEAR(B3),YEAR(B3)-1)
order_date_dim	orderDateID	INT	Dimension	1	N/A	N/A	N/A	NUMBER	Primary Key.
order_date_dim	date	VARCHAR(45)	Dimension	1	PEC	PECinvoice	orderDate	DATE	Taken from invoice orderDate and placed into seperate csv file. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE order_date_dim' or using Workbench Import Wizard
order_date_dim	month	INT	Dimension	1	PEC	PECinvoice	orderDate	DATE	Used Google Sheets formula: =MONTH(B3) to extract month from orderDate. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE order_date_dim' or using Workbench Import Wizard
order_date_dim	year	INT	Dimension	1	PEC	PECinvoice	orderDate	DATE	Used Google Sheets formula: =YEAR(B3) to extract year from orderDate. Loaded into database w/ 'LOAD DATA INFILE '*filename.csv*' INTO TABLE order_date_dim' or using Workbench Import Wizard
order_date_dim	fiscalQuarter	VARCHAR(2)	Dimension	1	PEC	PECinvoice	orderDate	DATE	Google Sheets formula: =IF(AND(MONTH(B3)>=1, MONTH(B3)<4), "Q4", IF(AND(MONTH(B3)>=4, MONTH(B3)<7), "Q1", IF(AND(MONTH(B3)>=7, MONTH(B3)<10), "Q2", IF(AND(MONTH(B3)>=10, MONTH(B3)<13), "Q3", "n/a"))))
order_date_dim	fiscalYear	INT	Dimension	1	PEC	PECinvoice	orderDate	DATE	Used Google Sheets formula to extract fiscalYear from orderDate: =IF(MONTH(B3)>3,YEAR(B3),YEAR(B3)-1)
invoice_fact	customerID_sk	INT	Fact	N/A	N/A	N/A	N/A	NUMBER	WHERE invoice_fact.customerID_sk = customer_dim.customer_sk
invoice_fact	productID_sk	INT	Fact	N/A	N/A	N/A	N/A	NUMBER	WHERE invoice_fact.productID_sk = product_dim.productID_sk
invoice_fact	orderID	INT	Fact	N/A	N/A	N/A	N/A	NUMBER	WHERE invoice_fact.orderID_sk = order_junk_dim.orderID
invoice_fact	salesDateID	INT	Fact	N/A	N/A	N/A	N/A	NUMBER	WHERE invoice_fact.salesDateID = sales_date_dim.salesDateID
invoice_fact	orderDateID	INT	Fact	N/A	N/A	N/A	N/A	NUMBER	WHERE invoice_fact.orderDateID = order_date_dim.orderDateID
invoice_fact	amt	INT	Fact	N/A	TPCE, TPCW, PEC	invoice_details, TCPWinvoice, PECinvoice	amt	NUMBER	Formatted with PECinvoice.ktr/TCPWcustomerAndinvoice.ktr/InvoiceMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE invoice_fact_dim' for each divisions' invoice csv file
invoice_fact	qty	INT	Fact	N/A	TPCE, TPCW, PEC	invoice_details, TCPWinvoice, PECinvoice	qty	NUMBER	Formatted with PECinvoice.ktr/TCPWcustomerAndinvoice.ktr/InvoiceMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE invoice_fact_dim' for each divisions' invoice csv file or using Workbench Import Wizard
invoice_fact	shipCost	VARCHAR(45)	Fact	N/A	PEC	PECinvoice	shipCost	NUMBER	Formatted with PECinvoice.ktr/TCPWcustomerAndinvoice.ktr/InvoiceMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE invoice_fact_dim' for each divisions' invoice csv file or using Workbench Import Wizard
invoice_fact	discounted	TINYINT	Fact	N/A	TPCE, TPCW, PEC	invoice_details, TCPWinvoice, PECinvoice	discounted	NUMBER	Formatted with PECinvoice.ktr/TCPWcustomerAndinvoice.ktr/InvoiceMerge.ktr, then loaded into database using 'LOAD DATA INFILE *'filename.csv'* INTO TABLE invoice_fact_dim' for each divisions' invoice csv file or using Workbench Import Wizard

V. SQL Code – Tables & Constraints

All data was loaded with the CSV files stated in section III.Data Staging: ETL – Data Extract File Definitions using the MySQL Workbench 'Table Data Import Wizard'.





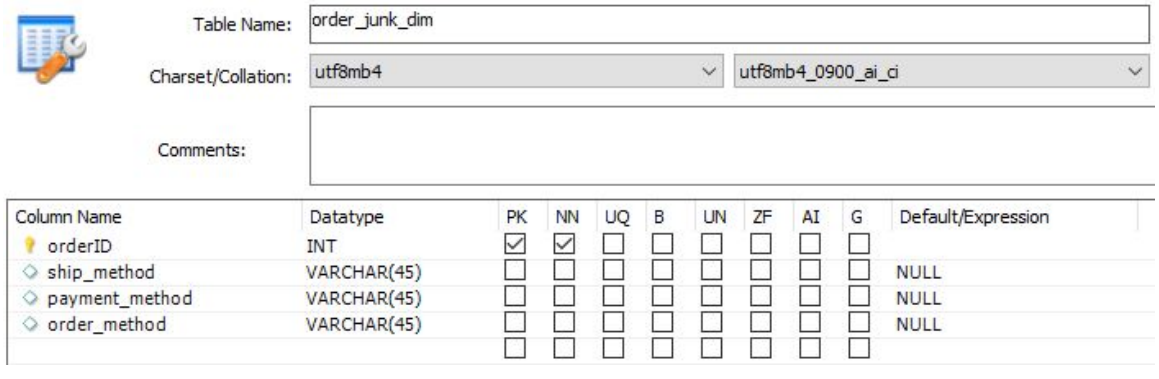


Table Name: product_dim

Charset/Collation: utf8mb4 / utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
salesBy	VARCHAR(45)	☑	☑	☐	☐	☐	☐	☐	☐	
productID_sk	INT	☑	☑	☐	☐	☐	☐	☐	☐	
productID	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
productName	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
price1	DOUBLE	☐	☐	☐	☐	☐	☐	☐	☐	NULL
price2	DOUBLE	☐	☐	☐	☐	☐	☐	☐	☐	NULL
unitCost	DOUBLE	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierName	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierAddr1	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierAddr2	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierCity	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierState	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
supplierZipCode	VARCHAR(5)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
typeDescription	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
businessUnitID	CHAR(1)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
businessUnitName	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
businessUnitAbbrev	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

Table Name: order_date_dim

Charset/Collation: utf8 / utf8_bin

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
orderDateID	INT	☑	☑	☐	☐	☐	☐	☐	☐	
date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
month	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
year	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
fiscalQuarter	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
fiscalYear	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

Table Name: sales_date_dim

Charset/Collation: utf8 / utf8_bin

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
salesDateID	INT	☑	☑	☐	☐	☐	☐	☐	☐	
date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
month	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
year	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
fiscalQuarter	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
fiscalYear	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

VI. Data Staging Activities - ETL

1. Data Cleansing

DM Table	Attribute	Problem	Resolution Strategy (attach code)
TPCWcustomer	address	Commas and periods	TPCWcustomerAndinvoice.ktr
TPCWcustomer	address	Department #s and suite #s included	TPCWcustomerAndinvoice.ktr
TPCWcustomer	zip	Zip code with 4 digits	TPCWcustomerAndinvoice.ktr
TPCWinvoice	All attributes	Columns shifted	TPCWcustomerAndinvoice.ktr
TPCWinvoice	invoiceID	Record with only invoiceID	TPCWcustomerAndinvoice.ktr
TPCWinvoice	custID	Negative custID	TPCWcustomerAndinvoice.ktr
TPCWinvoice	date	Date in 2 diff formats	TPCWcustomerAndinvoice.ktr
TPCWinvoice	amt	Sales amt incorrectly calculated	TPCWcustomerAndinvoice.ktr
TPCWinvoice	productID	Non-existent product ID (399)	Manually removed
PECinvoice	salesDate	Incorrect date format	PECinvoice.ktr
PECinvoice	orderDate	Incorrect date format	PECinvoice.ktr
PECinvoice	shipMethod	Incorrect shipping method	PECinvoice.ktr
PECinvoice	amt	Sales amounts are incorrect	PECinvoice.ktr
PECinvoice	All Attributes	1 record has column values shifted	Manually corrected using Notepad++
PECproduct	unitCost	Units costs are not properly calculated	PECproduct.ktr & manually running equation for each product
TPCWproduct	SupplierCityAndState	Needs to be split into two attributes	TPCWproduct.ktr
TPCWproduct	All attributes	Duplicate entry	TPCWproduct.ktr
PECcustomer	address	Unnecessary commas	PECcustomer.ktr

2. Data Transformation

DM Table	Image Creation Process (attach code)
salesDate	Used Google Sheets formulas to convert salesDate into unique ids and create month/year columns. Create month column: =LEFT(B1:B, 1) Create year column: =RIGHT(B1:B, 1) Convert date to unique id: = TEXT(C2,"mmddyy")
orderDate	Used Google Sheets formulas to convert salesDate into unique ids and create month/year columns. Create month column: (=LEFT(B1:B, 1) Create year column: (=RIGHT(B1:B, 1) Convert date to unique id: (=ARRAYFORMULA(REGEXREPLACE(A1:A,"/",))
Invoice (TPCW and TPCE)	Used Google Sheets formulas to convert salesDate into unique ids and create month/year columns. Create month column: (=LEFT(B1:B, 1) Create year column: (=RIGHT(B1:B, 1) Convert date to unique id: = TEXT(C2,"mmddyy") Also added shipMethod, shipCost, paymentMethod, orderMethod, and orderDate to be consistent with the other invoice file (PECinvoice). Also updated customerIDs to match the customer table.
Invoice (PEC)	Used Google Sheets formulas to convert salesDate into unique ids and create month/year columns. Create month column: (=LEFT(B1:B, 1) Create year column: (=RIGHT(B1:B, 1) Convert date to unique id: = TEXT(C2,"mmddyy") Also updated customerIDs to match the customer table.
All customer files	Manually merged together to remove duplicates into ‘customer.csv’
Customer	Merged together to form TPCEcustomer which was merged with the other customers to remove duplicates into ‘customer.csv’
Product	Merged together PECproduct and PECproductType with Productmerge.ktr
Product	Merged together TPCEproduct and TPCEproductType with Productmerge.ktr
Product	Merged together TPCWproduct and TPCWproductType with Productmerge.ktr
Invoice (TCPE)	Merged together TPCEinvoice and invoice with InvoiceMerge.ktr
PECProductType	Removed commas, abbreviations, and cleaned up with PECproducttype.ktr
PECBusinessUnit	Removed commas, abbreviations, and cleaned up with PECbusinessunit.ktr

3. Table Population

DM Table	**Table Population Process** (attach code)
Business Unit Junk	businessUnit.csv
Sales Date Dim	salesDate.csv
Order Date Dim	orderDate.csv
Customer Dim	customer.csv
Product Dim	PECproduct.csv, TPCWproduct.csv, TPCEproduct.csv
Invoice Fact	PECinvoice.csv, TPCWinvoice.csv, TPCEinvoice.csv

VII. End User Applications

1. Queries

User Question/Reporting Need
Show the total cost of products for each supplier
SQL Code
SELECT sum(unitCost) as 'Total Cost', supplierName FROM product_dim WHERE supplierName <> '' Group By supplierName;
Supporting Index(es)

Total Cost	supplierName
8333.300000000001	Afg Industries Inc.
5687.3	American General Ventures Inc
8770.7	Black Hills Corp
8522.099999999999	Dollar General Corp
6644.400000000015	Fedders Corp
6844.000000000002	First Bancshares Inc Mo
3811.2000000000003	Immunex Corp
3401.6000000000004	Saurer Gruppe Holding Ag
4141.2	Sinclair Broadcast Group Inc.
3807.6000000000004	Specialty Teleconstructors Inc
1267.6	TPC East
1644.9999999999998	TPC West

User Question/Reporting Need
The most frequent method of ordering a product from PEC
SQL Code
SELECT ship_method, count(ship_method) FROM order_junk_dim INNER JOIN invoice_fact ON order_junk_dim.orderID = invoice_fact.orderID INNER JOIN product_dim ON invoice_fact.productID_sk = product_dim.productID_sk WHERE salesBy LIKE 'PEC' AND ship_method <> '' AND ship_method <> '0' GROUP BY ship_method
Supporting Index(es)

ship_method	count(ship_method)
air	21849
n/a	21781
train	21499
truck	21756

User Question/Reporting Need
Show sales from one division to another
SQL Code
SELECT sum(amt), salesBy FROM invoice_fact JOIN product_dim ON invoice_fact.productID_sk=product_dim.productID_sk GROUP BY salesBy;
Supporting Index(es)

sum(amt)	salesBy
0	0
4681944327	PEC
6536089398	TPCE
6542294370	TPCW

2. A View

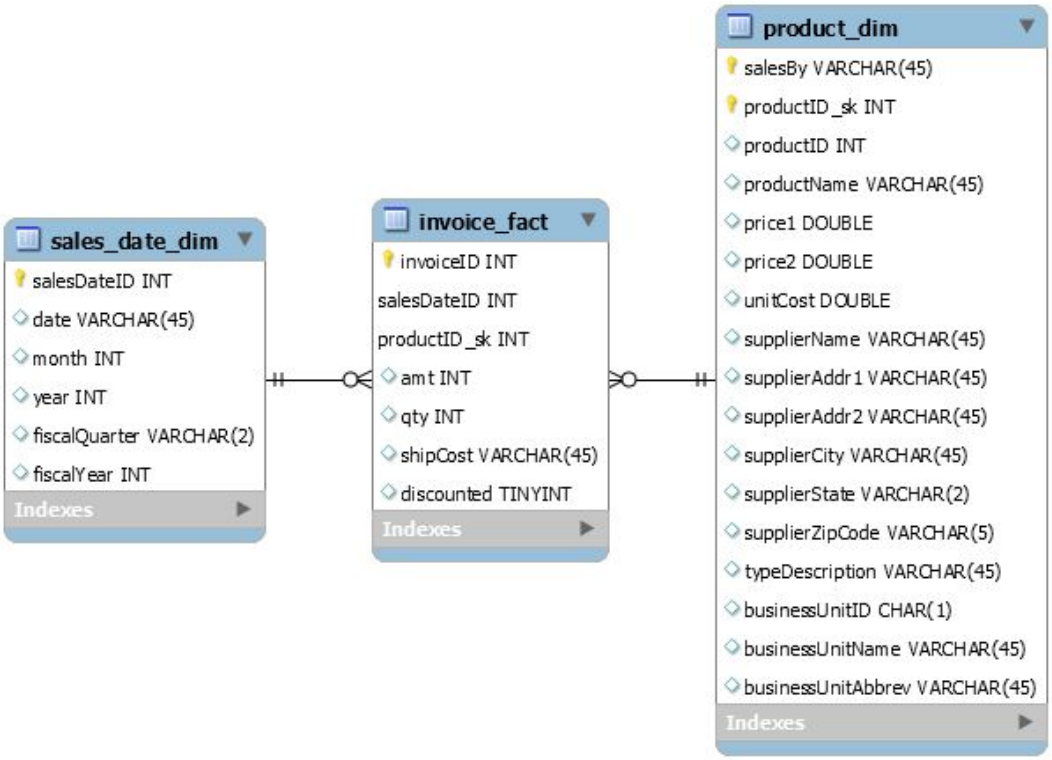
This generates a report of the total sales for each business unit and sorts them by year.

SELECT sum(amt) as 'Sales Amount',
product_dim.businessUnitName,sales_date_dim.year FROM invoice_fact join
product_dim on invoice_fact.productID_sk=product_dim.productID_sk
join sales_date_dim on invoice_fact.salesDateID=sales_date_dim.salesDateID
group by product_dim.businessUnitName, sales_date_dim.year ORDER BY
sales_date_dim.year;

Sales Amount	businessUnitName	year
363811352	Chemicals	5
584523055	Disposable Supplies	5
712410030	Miscellaneous	5
1345853508	Processing Equipment	5
713227799	Miscellaneous	6
1382763076	Processing Equipment	6
579749498	Disposable Supplies	6
370037473	Chemicals	6
1368034168	Processing Equipment	7
724017033	Miscellaneous	7
590910023	Disposable Supplies	7
376466859	Chemicals	7
1375551630	Processing Equipment	8
722907224	Miscellaneous	8
374539532	Chemicals	8
585169972	Disposable Supplies	8
372121789	Chemicals	9
571097639	Disposable Supplies	9
718207098	Miscellaneous	9
1353855372	Processing Equipment	9
303560449	Chemicals	10
1130171052	Processing Equipment	10
482892568	Disposable Supplies	10
601472182	Miscellaneous	10
0	Miscellaneous	11
1261434	Processing Equipment	11

3. Aggregated Data Marts

Lost Dimension



By dropping the Customer, Order Junk, and Order Date dimensions, we can create queries that relate directly between product, invoice, and sales date. This information can be used for selecting information that directly relates between product and sales date; such as product name and supplier names connected with their fiscal quarters and years.
Query Example #1:

```
SELECT productName, fiscalYear, sum(amt) as 'totalAmount'
FROM invoice_fact
JOIN product_dim
ON invoice_fact.productID_sk = product_dim.productID_sk
JOIN sales_date_dim
ON invoice_fact.salesDateID = sales_date_dim.salesDateID
GROUP BY productName, fiscalYear
ORDER BY sum(amt) DESC
LIMIT 5
```

This query will select the top 5 products by their amount sold and in which fiscal year that total amount was sold. This can help with determining which product sells the most total amount and in which year it was at its best.

productName	fiscalYear	totalAmount
Defeated Tray Supplies	2007	76017038
Planetesimal Manufacturing Equip	2006	76016389
Defeated Tray Supplies	2005	75795189
Defeated Tray Supplies	2006	75024610
Defeated Tray Supplies	2009	73952132

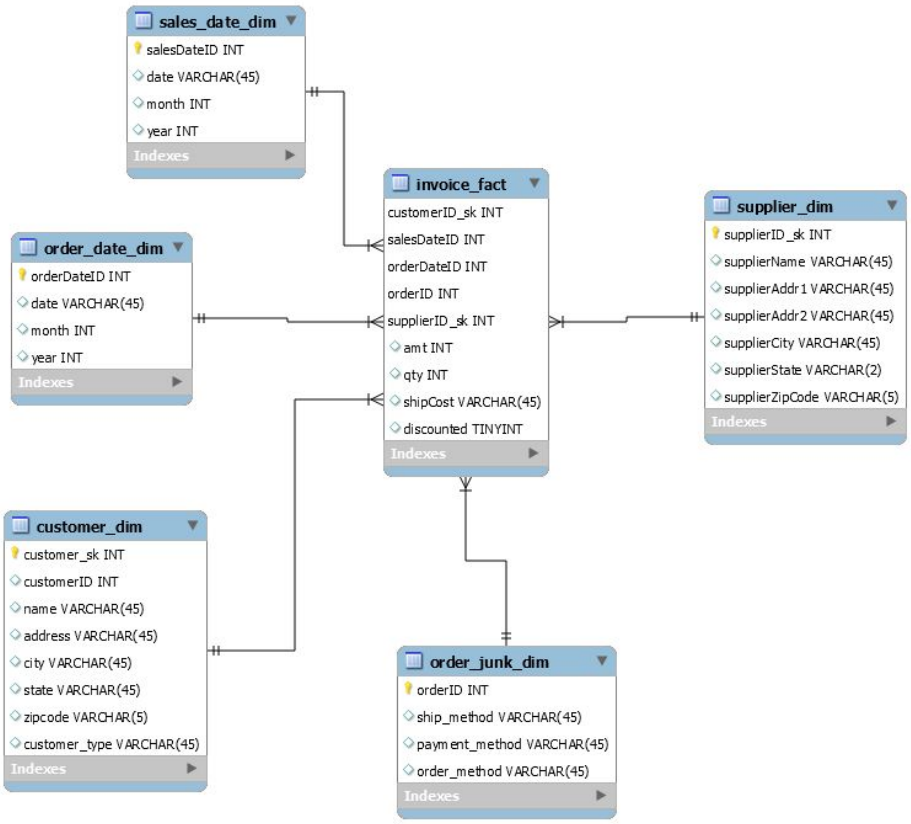
Query Example #2:

```
SELECT businessUnitName, fiscalQuarter, sum(qty) as 'totalQuantity'
FROM invoice_fact
JOIN product_dim
ON invoice_fact.productID_sk = product_dim.productID_sk
JOIN sales_date_dim
ON invoice_fact.salesDateID = sales_date_dim.salesDateID
GROUP BY businessUnitName,  fiscalQuarter
ORDER BY fiscalQuarter, businessUnitName ASC
```

Similarly, this query will show the total quantity of products sold by business unit name for each fiscal quarter. This can help with displaying how much total quantity each business unit sells in each fiscal quarter.

businessUnitName	fiscalQuarter	totalQuantity
Chemicals	Q1	1348005
Disposable Supplies	Q1	2487519
Miscellaneous	Q1	3071115
Processing Equipment	Q1	5666345
Chemicals	Q2	1356457
Disposable Supplies	Q2	2502373
Miscellaneous	Q2	3102696
Processing Equipment	Q2	5744521
Chemicals	Q3	1242491
Disposable Supplies	Q3	2353330
Miscellaneous	Q3	2784670
Processing Equipment	Q3	5163914
Chemicals	Q4	1314111
Disposable Supplies	Q4	2480743
Miscellaneous	Q4	3055949
Processing Equipment	Q4	5655468

Shrunken Dimension



The shrunken method was used to create the supplier_dim table by replacing the original product_dim table with a rolled up version of itself. The use case for this is now the supplier_dim be used to determine metrics associated specifically with suppliers. Using this method improves performance when users run queries associated with supplier information.

Query Example #1:

```
SELECT supplier_dim.supplierName, sum(invoice_fact.amt)
FROM invoice_fact
JOIN supplier_dim
ON invoice_fact.supplierID_sk = supplier_dim.supplierID_sk
GROUP BY supplierName
HAVING sum(invoice_fact.amt) > 0
ORDER BY sum(invoice_fact.amt) ASC
LIMIT 10
```

Results:

	supplierName	Total Sales
▶	Specialty Teleconstructors Inc	670795820
	Sinclair Broadcast Group Inc.	770142180
	Immunex Corp	845016614
	Saurer Gruppe Holding Ag	964675377
	Black Hills Corp	1414811294
	Afg Industries Inc.	1481323737
	Fedders Corp	1567884325
	First Bancshares Inc Mo	1596644012
	American General Ventures Inc	1793519913
	Dollar General Corp	1917854216

This query shows the top ten suppliers whose products had the least amount of sales. This is useful in determining which suppliers are not as integral to the company's profits.

Query Example #2:

```
SELECT supplier_dim.supplierName, sum(invoice_fact.qty)
FROM invoice_fact
JOIN supplier_dim
ON invoice_fact.supplierID_sk = supplier_dim.supplierID_sk
JOIN sales_date_dim
ON invoice_fact.salesDateID = sales_date_dim.salesDateID
WHERE sales_date_dim.month = 1 AND sales_date_dim.year = 10
GROUP BY supplierName
ORDER BY sum(invoice_fact.qty) DESC
LIMIT 5
```

Results:

	supplierName	Total Quantity
▶	American General Ventures Inc	83533
	Dollar General Corp	75280
	Afg Industries Inc.	63569
	Fedders Corp	60061
	First Bancshares Inc Mo	59626

This query shows the top 5 suppliers who had the most quantity sold in January of 2010. This is useful in seeing how well a supplier is doing in terms of performance and it also helps determine if more stock needs to be bought from certain suppliers.

Collapsed Dimension

The collapse method is used to compress the data mart into something that can easily be used for simple reports. This collapse is used for basic price checking and for how many items are sold.

Query Example #1:

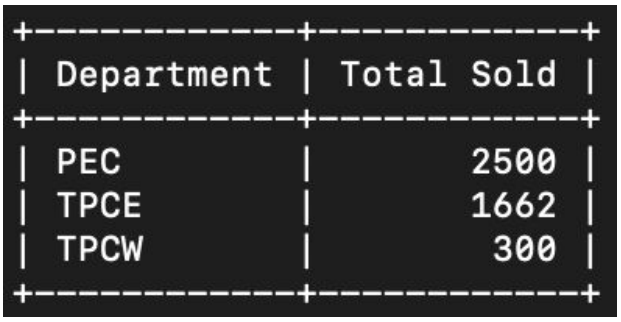
SELECT productName, sum(qty) as 'Total Sold' from invoice_fact GROUP BY productName;

productName	Total Sold
Barrels Flushing Chemicals	84
Bluest Fillers	41
Bomber Photo Equip	29
Bomber Photo Equip.	144
Carelessly Freezing Chemicals	19
Chalmers Polishing Equip.	43
Chalmers Polishing Equipment	177
Commendation Fillers	301
Consideration Tray Supplies	169
Courthouses Manufacturing Equip	150
Coward Covers	159
Decided Tray Supplies	105
Defeated Tray Supplies	188
Denigrating Polishing Equip	70
Denigrating Polishing Equip.	190
Disallow Manufacturing Equip	84

This just gets the total number of a product sold through all of the invoices

Query Example #2:

SELECT salesBy as 'Department', sum(qty) as 'Total Sold' from invoice_fact GROUP BY salesBy;



Department	Total Sold
PEC	2500
TPCE	1662
TPCW	300

This gets the total number of products sold by each department.

VIII. Handling Slowly Changing Dimensions (SCD)

The type of SCD should be determined based on the dimensions purpose in the data mart. For example, our customer dimension should have items be overwritten which would be type 1 while our product dimension would have type 2 because we shouldn't delete products even if there is an error.

Customer Type 0

customer_sk	customerID	name	city	state	customer_type
1	1	Beverly Equi	Bloomington	Hawaii	State
2	2	Clio-Hutchins	Butler	South Carolir	State
3	3	YuliChem	Laguna Woods	Kentucky	Comm
4	4	Austin Burns	Alexandria	DC	Edu
5	5	Saint Andrev	North Miami	Fl	Govt
6	6	Austin Ferrel	Pullman	Pennsylvania	Comm
7	7	Hop Adams	Sun Valley	New Hampsl	Edu
8	8	Mullins Inc.	Birmingham	Nebraska	State
9	9	Meridian Res	Fall River	Massachuset	Comm
10	10	Byron Chemi	Salt Lake City	Indiana	Comm
11	11	Beasley	Idabel	Massachuset	Comm
12	12	Processing E	Naperville	Illinois	Comm
13	13	Zena Machin	San Juan	Washington	State
14	14	The Final Frc	Biddeford	Missouri	State
15	15	Merritt Long	West Haven	Wyoming	Comm
16	16	Martinez Dis	Tucson	District of Co	Govt
17	17	Camera Platt	Houston	Texas	State
18	18	Cross	Hopkinsville	New Hampsl	Govt
19	19	Firstfed Ame	Ward Prairie	Texas	Edu
20	20	Price Rivers	Sault Ste. Marie	Mississippi	State
21	21	Santos LLC	Billings	Indiana	Comm
22	22	Santiago Pro	Hanahan	Alaska	State
23	23	Chemfix Tecl	Pasadena Hills	Missouri	State
24	24	Chantale-Hut	New Iberia	New Hampsl	State
25	25	Garrison Wo	Reno	West Virgini	Edu

All changes made manually in excel

Customer Type 1

customer_sk	customerID	name	city	state	customer_type
1	1	Beverly Equi	Bloomington	Hawaii	State
2	2	Clio-Hutchins	Butler	South Carolir	State
3	3	YuliChem	Laguna Woods	Texas	Comm
4	4	Austin Burns	Alexandria	DC	Edu
5	5	Saint Andrev	North Miami	Fl	Govt
6	6	Austin Ferrel	Pullman	Pennsylvania	Comm
7	7	Hop Adams	Sun Valley	New Hampsl	Edu
8	8	Mullins Inc.	Birmingham	Nebraska	State
9	9	Meridian Res	Fall River	Massachuse	Comm
10	10	Byron Chemi	Salt Lake City	Indiana	Comm
11	11	Beasley	Idabel	Massachuse	Comm
12	12	Processing E	Naperville	Illinois	Comm
13	13	Zena Machin	San Juan	Washington	State
14	14	The Final Fro	Biddeford	Missouri	State
15	15	Merritt Long	West Haven	Wyoming	Comm
16	16	Martinez Dis	Tucson	District of Co	Govt
17	17	Camera Platf	Houston	Texas	State
18	18	Cross	Hopkinsville	New Hampsl	Govt
19	19	Firstfed Ame	Ward Prairie	Texas	Edu
20	20	Price Rivers	Sault Ste. Marie	Mississippi	State
21	21	Santos LLC	Billings	Indiana	Comm
22	22	Santiago Pro	Hanahan	Alaska	State
23	23	Chemfix Tecl	Pasadena Hills	Missouri	State
24	24	Chantale-Huf	New Iberia	New Hampsl	State
25	25	Garrison Wo	Reno	West Virgini	Edu

All changes made manually in excel

Product Type 0

salesBy	productID_sk	productID	productName	unitCost	typeDescription
PEC	1	1	Enumerator Polishing Equip.	120	Polishing Equip.
PEC	2	2	Planetesimal Manufacturing Equip	250	Manufacturing Equip.
PEC	3	3	Tailor Jacks	399.7	Jacks
PEC	4	4	Miniaturizing Manufacturing Equip	325	Manufacturing Equip.
PEC	5	5	Flake Photo Equip.	170	Photo Equip.
TPCE	1	1	Enumerator Polishing Equip	135.4	Polishing Equip
TPCE	2	2	Tailor Jacks	399.7	Jacks
TPCE	3	3	Sortie Covers	206.8	Covers
TPCE	4	4	Embodying Cleaning Supplies	207.1	Cleaning Supplies
TPCE	5	5	Automobiles Fillers	218.5	Fillers
TPCE	6	6	Planetesimal Manufacturing Equip	285.5	Manufacturing Equip
TPCE	7	7	Commendation Fillers	364.8	Fillers
TPCE	8	8	Septembers Manufacturing Equip	517.8	Manufacturing Equip
TPCE	9	9	Millimeters Flushing Chemicals	241.8	Flushing Chemicals
TPCE	10	10	Defeated Tray Supplies	448.3	Tray Supplies
TPCW	1	1	Enumerator Polishing Equipment	135.4	Polishing Equip
TPCW	2	2	Tailor Jacks	399.7	Jacks
TPCW	3	3	Sortie Covers	206.8	Covers
TPCW	4	4	Embodying Cleaning Supplies	207.1	Cleaning Supplies
TPCW	5	5	Automobiles Fillers	218.5	Fillers
TPCW	6	6	Planetesimal Manufacturing Equip	285.5	Manufacturing Equip
TPCW	7	7	Commendation Fillers	364.8	Fillers
TPCW	8	8	Septembers Manufacturing Equip	517.8	Manufacturing Equip
TPCW	21	21	Denigrating Polishing Equipment	427.2	Polishing Equip

All changes made manually in excel

Product Type2

salesBy	productID_sk	productID	productName	unitCost	typeDescription
PEC	1	1	Enumerator Polishing Equip.	120	Polishing Equip.
PEC	2	2	Planetesimal Manufacturing Equip	250	Manufacturing Equip.
PEC	3	3	Tailor Jacks	399.7	Jacks
PEC	4	4	Miniaturizing Manufacturing Equip	325	Manufacturing Equip.
PEC	5	5	Flake Photo Equip.	170	Photo Equip.
PEC	6	5	Flake Photo Equip.	200	Photo Equip.
TPCE	1	1	Enumerator Polishing Equip	135.4	Polishing Equip
TPCE	2	2	Tailor Jacks	399.7	Jacks
TPCE	3	3	Sortie Covers	206.8	Covers
TPCE	4	4	Embodying Cleaning Supplies	207.1	Cleaning Supplies
TPCE	5	5	Automobiles Fillers	218.5	Fillers
TPCE	6	6	Planetesimal Manufacturing Equip	285.5	Manufacturing Equip
TPCE	7	7	Commendation Fillers	364.8	Fillers
TPCE	8	8	Septembers Manufacturing Equip	517.8	Manufacturing Equip
TPCE	9	9	Millimeters Flushing Chemicals	241.8	Flushing Chemicals
TPCE	10	10	Defeated Tray Supplies	448.3	Tray Supplies
TPCW	1	1	Enumerator Polishing Equipment	135.4	Polishing Equip
TPCW	2	2	Tailor Jacks	399.7	Jacks
TPCW	3	3	Sortie Covers	206.8	Covers
TPCW	4	4	Embodying Cleaning Supplies	207.1	Cleaning Supplies
TPCW	5	5	Automobiles Fillers	218.5	Fillers
TPCW	6	6	Planetesimal Manufacturing Equip	285.5	Manufacturing Equip
TPCW	7	7	Commendation Fillers	364.8	Fillers
TPCW	8	8	Septembers Manufacturing Equip	517.8	Manufacturing Equip
TPCW	21	21	Denigrating Polishing Equipment	427.2	Polishing Equip

All changes made manually in excel

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

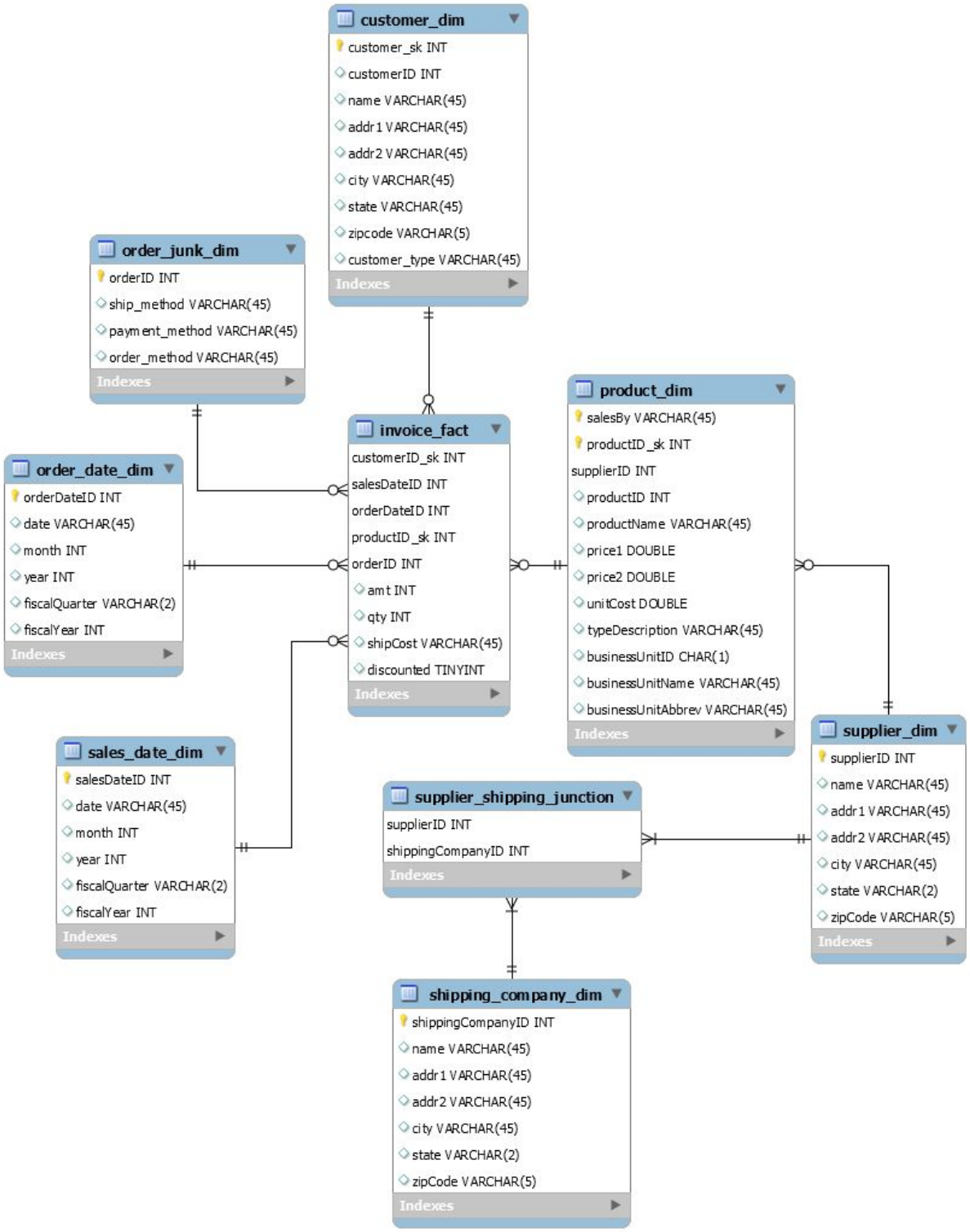
Instead of just linking the supplier and shipping company tables together, the implementation decided upon was to utilize a Junction/Joining Table. First, the supplier information was pulled out of the product dimension and placed into its own dimension - supplier_dim. Then, the shipping company dimension was created with shipping company data - shippping_company_dim. Finally, we can connect these two dimensions using the junction table - supplier_shipping_junction. This table includes the primary keys from both the supplier and the shipping company, allowing them to be connected in a many-to-many relationship.

Another, simpler method of many-to-many relationships is to just link the two tables together. But this means we cannot have the primary keys stored in either table, requiring either a column that contains multiple values or multiple columns for each unique ID. This is very much not ideal as it will make SQL queries much more difficult and harder to maintain. Adding new suppliers or shipping companies would prove difficult and updating existing records would be a challenge and require unique SQL queries for each shipping company / supplier.

Works Cited

Airtable's guide to many-to-many relationships. (n.d.). Retrieved November 28, 2020, from https://support.airtable.com/hc/en-us/articles/218734758-Airtable-s-guide-to-many-to-many-relationships

Brumm, B. (2019, June 19). How to Handle a Many-to-Many Relationship in Database Design - DZone Database. Retrieved November 28, 2020, from https://dzone.com/articles/how-to-handle-a-many-to-many-relationship-in-datab


customer_dim
customer_sk INT
customerID INT
name VARCHAR(45)
addr1 VARCHAR(45)
addr2 VARCHAR(45)
city VARCHAR(45)
state VARCHAR(45)
zipcode VARCHAR(5)
customer_type VARCHAR(45)
Indexes
order_junk_dim
orderID INT
ship_method VARCHAR(45)
payment_method VARCHAR(45)
order_method VARCHAR(45)
Indexes
invoice_fact
customerID_sk INT
salesDateID INT
orderDateID INT
productID_sk INT
orderID INT
amt INT
qty INT
shipCost VARCHAR(45)
discounted TINYINT
Indexes
product_dim
salesBy VARCHAR(45)
productID_sk INT
supplierID INT
productID INT
productName VARCHAR(45)
price1 DOUBLE
price2 DOUBLE
unitCost DOUBLE
typeDescription VARCHAR(45)
businessUnitID CHAR(1)
businessUnitName VARCHAR(45)
businessUnitAbbrev VARCHAR(45)
Indexes
order_date_dim
orderDateID INT
date VARCHAR(45)
month INT
year INT
fiscalQuarter VARCHAR(2)
fiscalYear INT
Indexes
sales_date_dim
salesDateID INT
date VARCHAR(45)
month INT
year INT
fiscalQuarter VARCHAR(2)
fiscalYear INT
Indexes
supplier_shipping_junction
supplierID INT
shippingCompanyID INT
Indexes
supplier_dim
supplierID INT
name VARCHAR(45)
addr1 VARCHAR(45)
addr2 VARCHAR(45)
city VARCHAR(45)
state VARCHAR(2)
zipCode VARCHAR(5)
Indexes
shipping_company_dim
shippingCompanyID INT
name VARCHAR(45)
addr1 VARCHAR(45)
addr2 VARCHAR(45)
city VARCHAR(45)
state VARCHAR(2)
zipCode VARCHAR(5)
Indexes

X. Appendix (Fix Lab #3 Problems)

PECcustomer.csv

- Manually created addr1 and addr2 to split any department numbers, suite numbers, and PO boxes into addr2.

TPCWproduct.csv:

- Manually removed the following duplicate products:
  - Escape Manufacturing Equip.
  - Curiouser Cleaning Supplies
  - Optima Cleaning Supplies
  - Measured Photo Chemicals

TPCWcustomer.csv

- Manually created addr1 and addr2 to split any department numbers, suite numbers, and PO boxes into addr2.
- Manually prefixed zip codes with 4 digits with 0's instead of 1's.

TPCWinvoice.csv:

- Incorrected shifted column- manually updated invoice with invoiceID = 3032.
- Manually removed the record with only invoiceID, ID = 26511.
- Fixed date format to match the 'PECinvoiceClean.csv' date format.

Uniformity among all files:

- Fixed using Google Sheets to have all files use the shorthand version with periods.

Customer Dim:

- Updated state field to have all states use abbreviations using Google Sheets in all caps.
- Manually created addr1 and addr2 to split any department numbers, suite numbers, and PO boxes into addr2.
- Duplications removed.

Product Dim:

- Added business unit information to the product dimension using the following SQL code:

```
CREATE TABLE new_product_dim
(
        SELECT product_dim.salesBy, product_dim.productID_sk,
                product_dim.productID, product_dim.productName, product_dim.price1,
                product_dim.price2, product_dim.unitCost,
                product_dim.supplierName, product_dim.supplierAddr1,
                product_dim.supplierAddr2, product_dim.supplierCity,
                product_dim.supplierState, product_dim.supplierZipCode,
                product_dim.typeDescription, business_unit_junk.businessUnitID,
```

```
            business_unit_junk.name as businessUnitName,
            business_unit_junk.abbrev as businessUnitAbbrev
        FROM product_dim
        INNER JOIN business_unit_junk
        ON product_dim.businessUnitID_sk = business_unit_junk.businessUnitID_sk
        ORDER BY product_dim.salesBy, product_dim.productID_sk
);
```

Then removed the business_unit_junk dim and replaced the old product_dim with the new_product_dim.

- 'SalesBy' field.
- Updated state field to have all states use abbreviations using Google Sheets in all caps.
- TPC West,TPC East, and PEC listed as suppliers when appropriate.

Order Date Dim:

- Added fiscal date through Google Sheets..

Sales Date Dim:

- Added fiscal date through Google Sheets.

Junk Dim:

- Junk dimension updated from business_unit_junk to order_junk_dim.

Fact Table:

- Added a null record for each dimension that did not already have one.

Load:

- 195,198 records in fact table.

Query 3:

- Fixed in section 'VII.End User Applications - Queries'.