--Drop procedure

IF OBJECT\_ID('dbo.UpdateAddress') IS NOT NULL

DROP PROCEDURE dbo.UpdateAddress

--Drop views

IF OBJECT\_ID('dbo.RegistryDetail') IS NOT NULL

DROP VIEW dbo.RegistryDetail

IF OBJECT\_ID('dbo.MonthlyOrders') IS NOT NULL

DROP VIEW dbo.MonthlyOrders

IF OBJECT\_ID('dbo.MonthlyProfit') IS NOT NULL

DROP VIEW dbo.MonthlyProfit

IF OBJECT\_ID('dbo.CategoryCount') IS NOT NULL

DROP VIEW dbo.CategoryCount

IF OBJECT\_ID('dbo.BrandCount') IS NOT NULL

DROP VIEW dbo.BrandCount

IF OBJECT\_ID('dbo.ProductByAge') IS NOT NULL

DROP VIEW dbo.ProductByAge

IF OBJECT\_ID('dbo.VendorPriceCompare') IS NOT NULL

DROP VIEW dbo.VendorPriceCompare

--Drop tables

IF OBJECT\_ID('ReferralList') IS NOT NULL

DROP TABLE dbo.ReferralList

IF OBJECT\_ID('RegistryProduct') IS NOT NULL

DROP TABLE dbo.RegistryProduct

IF OBJECT\_ID('RegistryPermission') IS NOT NULL

DROP TABLE dbo.RegistryPermission

IF OBJECT\_ID('PersonVendorRating') IS NOT NULL

DROP TABLE dbo.PersonVendorRating

IF OBJECT\_ID('OrderPayment') IS NOT NULL

DROP TABLE dbo.OrderPayment

IF OBJECT\_ID('Commission') IS NOT NULL

DROP TABLE dbo.Commission

IF OBJECT\_ID('OrderDetail') IS NOT NULL

DROP TABLE dbo.OrderDetail

IF OBJECT\_ID('ProductOrder') IS NOT NULL

DROP TABLE dbo.ProductOrder

IF OBJECT\_ID('Registry') IS NOT NULL

DROP TABLE dbo.Registry

IF OBJECT\_ID('OrderStatus') IS NOT NULL

DROP TABLE dbo.OrderStatus

IF OBJECT\_ID('Product') IS NOT NULL

DROP TABLE dbo.Product

IF OBJECT\_ID('Vendor') IS NOT NULL

DROP TABLE dbo.Vendor

IF OBJECT\_ID('Person') IS NOT NULL

DROP TABLE dbo.Person

--create database tables

CREATE TABLE Vendor(

--create columns

VendorID int identity

, VendorName varchar(50) NOT NULL

, VendorDescription varchar(256)

, VendorAddress varchar(256) NOT NULL

, CustomerSupportEmail varchar(50)

, CustomerSupportPhone varchar(15) NOT NULL

--place constraints

, CONSTRAINT PKVendorID PRIMARY KEY (VendorID)

)

--create database tables

CREATE TABLE Product(

--create columns

ProductID int identity

, ProductName varchar(50) NOT NULL

, VendorID int not null

, ProductDescription varchar(256) NOT NULL

, ProductDetail varchar(256) NOT NULL

, ProductCategory varchar(30)

, ProductBrand varchar(30) NOT NULL

, Color varchar(30) NOT NULL

, ProductInStock char(1) not null -- will be answered with Y / N

, QuantityInStock int

, VendorPrice int

, EstimatedShippingDays int NOT NULL

--place constraints

, CONSTRAINT PKProductID PRIMARY KEY (ProductID)

, CONSTRAINT FK1\_Product FOREIGN KEY(VendorID) REFERENCES Vendor(VendorID)

)

--create database tables

CREATE TABLE Person(

--create columns

PersonID int identity

, UserName varchar(50) NOT NULL

, FirstName varchar(100) NOT NULL

, LastName varchar(100) NOT NULL

, PersonAge int NOT NULL --NEW ADD

, StreetAddress varchar(50)

, StreetAddress2 varchar(50) NOT NULL

, CityAddress varchar(30)

, StateAddress varchar(30)

, CountryAddress varchar(30)

, ZipAddress varchar(15)

, PhoneNumber varchar(30)

, EmailAddress varchar(50)

, FutureSpouse varchar(50)

--place constraints

, CONSTRAINT PKPersonID PRIMARY KEY (PersonID)

, CONSTRAINT U1\_Person UNIQUE(UserName)

)

--create database tables

CREATE TABLE Registry(

--create columns

RegistryID int identity

, PersonID int NOT NULL

, SpouseID int not null

--place constraints

, CONSTRAINT PKRegistryID PRIMARY KEY (RegistryID)

, CONSTRAINT FK1\_Registry FOREIGN KEY(PersonID) REFERENCES Person(PersonID)

, CONSTRAINT FK2\_Registry FOREIGN KEY(PersonID) REFERENCES Person(PersonID)

)

--create database tables

CREATE TABLE RegistryProduct(

--create columns

RegistryProductID int identity

, RegistryID int NOT NULL

, ProductID int not null

, FulfilledStatus varchar(20)

, ProductQuantity int

--place constraints

, CONSTRAINT PKRegistryProductID PRIMARY KEY (RegistryProductID)

, CONSTRAINT FK1\_RegistryProduct FOREIGN KEY(RegistryID) REFERENCES Registry(RegistryID)

, CONSTRAINT FK2\_RegistryProduct FOREIGN KEY(ProductID) REFERENCES Product(ProductID)

)

--create database tables

CREATE TABLE OrderStatus(

--create columns

OrderStatusID int identity

, OrderStatus varchar(20) NOT NULL

--place constraints

, CONSTRAINT PKOrderStatusID PRIMARY KEY (OrderStatusID)

)

--create database tables

CREATE TABLE ProductOrder(

--create columns

OrderID int identity

, PersonID int NOT NULL

, RegistryID int NOT NULL

, ProductID int NOT NULL

, DeliveryAddress varchar(256) not null

, SenderAddress varchar(256) not null

, ReturnAddress varchar(256) not null

, ReturnInstructions varchar (256) not null

, ShippingMethod varchar (30)

, OrderQuantity int not null

, OrderSubtotal int

, OrderPaymentType varchar(50) not null

, CommissionRate decimal(10,2)

, OrderMonth varchar(15)

--place constraints

, CONSTRAINT PKOrderID PRIMARY KEY (OrderID)

, CONSTRAINT FK1\_ProductOrder FOREIGN KEY(PersonID) REFERENCES Person(PersonID)

, CONSTRAINT FK2\_ProductOrder FOREIGN KEY(RegistryID) REFERENCES Registry(RegistryID)

, CONSTRAINT FK3\_ProductOrder FOREIGN KEY(ProductID) REFERENCES Product(ProductID)

)

--create database tables

CREATE TABLE ReferralList(

--create columns

ReferralID int identity

, ReferorID int not null

, RefereeID int not null

--place constraints

, CONSTRAINT PKReferralID PRIMARY KEY (ReferralID)

, CONSTRAINT FK1\_ReferralList FOREIGN KEY(ReferorID) REFERENCES Person(PersonID)

, CONSTRAINT FK2\_ReferralList FOREIGN KEY(RefereeID) REFERENCES Person(PersonID)

)

--create database tables

CREATE TABLE PersonVendorRating(

--create columns

PersonVendorID int identity

, PersonID int NOT NULL

, VendorID int NOT NULL

, Rating decimal(10,2)

--place constraints

, CONSTRAINT PKPersonVendorID PRIMARY KEY (PersonVendorID)

, CONSTRAINT FK1\_PersonVendorRating FOREIGN KEY(PersonID) REFERENCES Person(PersonID)

, CONSTRAINT FK2\_PersonVendorRating FOREIGN KEY(VendorID) REFERENCES Vendor(VendorID)

)

--create database tables

CREATE TABLE RegistryPermission(

--create columns

PermissionID int identity

, PersonID int not null

, RegistryID int not null

, PermissionStatus varchar(20) NOT NULL

--place constraints

, CONSTRAINT PKPermissionID PRIMARY KEY (PermissionID)

, CONSTRAINT FK1\_RegistryPermission FOREIGN KEY(PersonID) REFERENCES Person(PersonID)

, CONSTRAINT FK2\_RegistryPermission FOREIGN KEY(RegistryID) REFERENCES Registry(RegistryID)

)

--create 10 vendors and insert into vendor table

INSERT INTO Vendor(VendorName, VendorDescription, VendorAddress, CustomerSupportEmail, CustomerSupportPhone)

VALUES ('Dream Vendor', 'We serve all areas of registry needs from bath to bed', '517 W Sequoia Ave, Chicago, IL 60606', 'support@dreamvendor.com', '3125575032')

,('First Rate Gifts', 'Top quality and gifts only at First Rate Gifts. We were established by a family for your family. We offer somewhat competitive pricing.', '212 W School Ave, Dallas, TX, 77104', 'support@firstrategifts.com', '4045578293')

,('Home Living', 'Trusted home living goods since 1690. We specialize in living room furnature and hae other items on hand too. Come check out a show room near you', '500 E 17th Ave, Chicago, IL 60606', 'support@homeliving.com', '3124084009')

,('Silverware Only', 'Boutique store specializing in silverware only. Our mission is to make the worlds first trendy spork', '90 Knife Street, Ann Arbor, MI, 48104', 'support@silverwareonly.com', '2489820402')

,('Bed, Kitchen & Nothing Beyond', 'New compeitior to Bed, Bath and Beyond. Why go to a store that claims to have a beyond section? We are honest about our lack of the "beyond"', '75 W 75th Avenue, New York, NY, 10777', 'support@bknb.com', '2482272277')

,('Luxury Living', 'Our luxury good are second to none - you pay for what you get', '68 South Forest Street, Aurora, CA, 50211', 'support@luxuryliving.com', '2026575043')

,('Thrifty Gifty', 'Our registry gifts are affordable, medium quality and delivered with a smile', '17 E Carson Ave, Carson City, NV, 32401', 'support@thriftygifty.com', '8016676677')

,('Bloominghalls', 'Rebranded recently from the popular home store. Different name, same products and services', '5 John Galt Ave, Chicago, IL 60606', 'support@bloominghalls.com', '3124562020')

,('Handcrafted', 'All our suppliers are local crafts people. No two pieces will be alike', '45 E Lake Street, Chicago, IL 60606', 'support@handcrafted.com', '6300428004')

,('Solo Cup', 'Fancy registry gifts available at affordable rates', '202 Industrial Ave, Chicago, IL 60606', 'support@solocup.com', '3124809080')

--insert a variety of products into the product table to eventually be purchased, added to registries, etc.

INSERT INTO product(ProductName, VendorID, ProductDescription, ProductDetail, ProductCategory, ProductBrand, Color, ProductInStock, QuantityInStock, VendorPrice, EstimatedShippingDays)

VALUES ('Plate Set', '1', 'Wonderful plate set, high quality, popular', '5 piece, Wood material, Dish washer safe', 'Dining', 'Lenox', 'White', 'Y', '100', '43', '5')

,('Plate Set', '2', 'Wonderful plate set, high quality, popular', '5 piece, Wood material, Dish washer safe', 'Dining', 'Lenox', 'White', 'Y', '50', '134', '5')

,('Plate Set', '6', 'Wonderful plate set, high quality, popular', '5 piece, Wood material, Dish washer safe', 'Dining', 'Lenox', 'White', 'Y', '25', '135', '5')

,('Silverware Set', '4', 'Wonderful silverware set, high quality, popular', '30 piece set, dishwasher friendly', 'Dining', 'Wusthof', 'Silver', 'Y', '50', '109', '5')

,('Silverware Set', '4', 'Wonderful silverware set, high quality, popular', '30 piece set, dishwasher friendly', 'Dining', 'Wusthof', 'Bronze', 'Y', '100', '49', '5')

,('Toaster', '8', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'Breville', 'Silver', 'Y', '50', '150', '7')

,('Toaster', '8', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'KitchenAid', 'Silver', 'Y', '25', '55', '7')

,('Toaster', '2', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'Breville', 'Silver', 'Y', '50', '1', '7')

,('Toaster', '2', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'KitchenAid', 'Silver', 'Y', '100', '209', '7')

,('Toaster', '5', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'Breville', 'Silver', 'Y', '50', '168', '7')

,('Toaster', '5', 'Will not burn your toast - guaranteed', '40L, 25H, 15W. 18V battery powered', 'Kitchen', 'KitchenAid', 'Silver', 'Y', '25', '160', '7')

,('Wine Fridge', '5', 'Keep your wine cool the old fashioned way', '18 bottle fridge, solar powered', 'Home', 'KitchenAid', 'Silver', 'Y', '50', '83', '10')

,('Bottle Opener', '9', 'Open your bottles in style with this indestructable bottle opener', 'Glass bottle opener', 'Kitchen', 'KitchenAid', 'Black', 'Y', '100', '3', '3')

,('Ice Cream Scoop', '7', 'Be the envy of all your friends with this ice cream scoop', 'Cast iron, 8 inches long', 'Kitchen', 'KitchenAid', 'Silver', 'Y', '50', '78', '3')

,('Vacuum Cleaner', '3', 'Tidy up all the unwanted pet hair', 'robotic, built in sensor', 'Home', 'iRobot', 'Black', 'Y', '25', '159', '10')

,('Vacuum Cleaner', '8', 'Will pick up at least 40% of your messes', 'Pet hair edition', 'Home', 'Dyson', 'Blue', 'Y', '50', '51', '10')

,('Coffee Maker', '6', 'Wanting to start your day with a cup of joe? Look no further', '10 cup maker', 'Kitchen', 'Ninja', 'Silver', 'Y', '100', '213', '7')

,('Coffee Maker', '7', 'Professional coffee at half the price', 'single serve, 18 inches tall by 8 inches wide', 'Kitchen', 'Keurig', 'Black', 'Y', '50', '56', '7')

,('Non-Stick Cookware', '9', 'Nothing will stick to these pans. Ever.', '5 piece set including lids', 'Kitchen', 'Cuisinart', 'Black', 'Y', '25', '131', '7')

,('Wine Glass', '10', 'Pour yourself the perfect glass with these French inspired wine glasses', '4 pieces in set, each sold seperately', 'Kitchen', 'Lenox', 'Clear', 'Y', '50', '234', '3')

,('Comforter Set', '6', 'Slide into your dreams with this set', 'Includes down comforter, queen size', 'Bed & Bath', 'Calvin Klein', 'Grey', 'Y', '100', '201', '5')

,('Coffee Mug', '7', 'Pairs perfectly with a Keurig. Lets face it, what good is coffee without a place to put it?', '8 lbs, heavy duty', 'Kitchen', 'KitchenAid', 'Red', 'Y', '50', '239', '3')

,('Knife Block', '8', 'Slice through fruit and packaging that is difficult to open', '34 piece set, new never-dull technology', 'Kitchen', 'Wusthof', 'Natural', 'Y', '25', '19', '3')

,('Cast Iron Skillet', '3', 'Which amateur chef actually needs one of these? Overpay and still burn your food', 'make from iron that was cast', 'Kitchen', 'Le Creuset', 'Green', 'Y', '50', '245', '5')

,('Pillow Case', '1', 'Drift into your dreams', 'Silk, oversized', 'Bed & Bath', 'Laura Ashley', 'Pink', 'Y', '100', '116', '3')

,('Measuring Cup Set', '10', 'Pour the perfect amount of anything with these measuring cups', '4 cupsizes 1 cup, 1/2 cup, 1/4 cup, 1/8 cup', 'Kitchen', 'KitchenAid', 'White', 'Y', '50', '164', '3')

,('Towel Set', '6', 'Dry off in style', '12 piece set including hand towels', 'Bed & Bath', 'Nautica', 'Royal Blue', 'Y', '25', '62', '7')

--Insert people into the person table. The people will have registries, place orders, etc.

INSERT INTO Person(UserName, FirstName,LastName, PersonAge, StreetAddress, StreetAddress2, CityAddress, StateAddress, CountryAddress, ZipAddress, PhoneNumber, EmailAddress, FutureSpouse)

VALUES ('vengefulSausage3', 'Tyler', 'Durden', '28', '317 E 1st Street', '', 'Chicago', 'IL', 'USA', '60606', '3128357781', 'tylerdurden@gmail.com', '')

,('pacifiedIcecream7', 'Joe', 'Exotic', '42', '1 Federal Way', '', 'Chicago', 'IL', 'USA', '60606', '3128990091', 'joeexotic@gmail.com', 'amazedOtter4')

,('amazedOtter4', 'Carole', 'Baskin', '43', '42 Bigcat Lane', '', 'Orlando', 'FL', 'USA', '32789', '8095574032', 'carolebaskin@bigcatrescue.com', 'pacifiedIcecream7')

,('lovesickSardines2', 'Lauren', 'Anderson', '25', '137 Chicago Ave', 'Apt #2', 'Chicago', 'IL', 'USA', '60606', '3127728837', 'laurenanderson@aol.com', 'grizzledChowder1')

,('grizzledChowder1', 'Jim', 'Sanders', '29', '137 Chicago Ave', 'Apt #2', 'Chicago', 'IL', 'USA', '60606', '3128982203', 'jimsanders1@gmail.com', 'lovesickSardines2')

,('grumpyRat1', 'Mike', 'Fernandez', '55', '222 W 5th Street', '', 'Houston', 'TX', 'USA', '77002', '4018384718', 'mikefernandez@hotmail.com', 'troubledCod7')

,('troubledCod7', 'Julia', 'Brookfield', '32', '122 E Oak', '', 'Detroit', 'MI', 'USA', '48104', '2489820402', 'juliabrookfield@work.com', 'grumpyRat1')

,('dopeyPiglet7', 'Michelle', 'Branch', '31', '55 Elm Street', '#203', 'Bozeman', 'MT', 'USA', '59718', '6156713232', 'michellebranch55@gmail.com', 'dreadfulDog4')

,('dreadfulDog4', 'Jordan', 'Rodgers', '34', '55 Elm Street', '#203', 'Bozeman', 'MT', 'USA', '59718', '6156711211', 'jordanrodgers6@gmail.com', 'dopeyPiglet7')

,('giddyTuna7', 'Ron', 'Swanson', '20', '5 Wood Ave', '', 'Pawnee', 'IN', 'USA', '46013', '2189884250', 'ronswanson@yahoo.com', ''

)

--the registry table has largely been scaled back. Registries are being created for a person and their spouse

INSERT INTO Registry(PersonID, SpouseID)

VALUES ('2', '3'), ('4', '5'), ('6', '7'), ('8', '9')

--add detail to registry product. this is a way to associate many products with one registry

INSERT INTO RegistryProduct(RegistryID, ProductID, FulfilledStatus, ProductQuantity)

VALUES ('1', '1', 'Fulfilled', '10')

,('1', '4', 'Unfulfilled', '6')

,('1', '6', 'Fulfilled', '1')

,('1', '12', 'Fulfilled', '1')

,('1', '13', 'Fulfilled', '1')

,('1', '14', 'Unfulfilled', '1')

,('1', '15', 'Fulfilled', '1')

,('1', '22', 'Unfulfilled', '8')

,('1', '23', 'Fulfilled', '1')

,('1', '24', 'Unfulfilled', '2')

,('2', '1', 'Fulfilled', '8')

,('2', '4', 'Fulfilled', '8')

,('2', '6', 'Fulfilled', '1')

,('2', '12', 'Unfulfilled', '1')

,('2', '13', 'Fulfilled', '1')

,('2', '14', 'Unfulfilled', '1')

,('2', '15', 'Fulfilled', '1')

,('2', '25', 'Unfulfilled', '4')

,('2', '26', 'Fulfilled', '1')

,('2', '27', 'Unfulfilled', '4')

,('3', '12', 'Fulfilled', '1')

,('3', '13', 'Unfulfilled', '1')

,('3', '14', 'Fulfilled', '1')

,('3', '15', 'Fulfilled', '1')

,('3', '21', 'Fulfilled', '1')

,('3', '22', 'Fulfilled', '4')

,('3', '23', 'Fulfilled', '1')

,('3', '24', 'Unfulfilled', '1')

,('3', '25', 'Fulfilled', '2')

,('3', '27', 'Unfulfilled', '6')

,('4', '1', 'Fulfilled', '4')

,('4', '4', 'Unfulfilled', '6')

,('4', '6', 'Fulfilled', '1')

,('4', '14', 'Fulfilled', '1')

,('4', '15', 'Unfulfilled', '1')

,('4', '22', 'Fulfilled', '5')

,('4', '23', 'Unfulfilled', '1')

,('4', '24', 'Fulfilled', '2')

,('4', '25', 'Unfulfilled', '2')

,('4', '27', 'Fulfilled', '4')

--One important component of the system is the ability for a person to rate a vendor - this will inform future purchases, decisions

INSERT INTO PersonVendorRating(PersonID, VendorID, Rating)

VALUES ('1', '2', '5') ,('2', '3', '5') ,('3', '5', '2') ,('4', '5', '5') ,('5', '8', '3')

,('6', '8', '3') ,('7', '5', '1') ,('8', '2', '4') ,('9', '5', '4') ,('10', '5', '5')

,('1', '7', '1') ,('2', '4', '3') ,('3', '10', '2') ,('4', '9', '1') ,('5', '6', '3')

,('6', '3', '3') ,('7', '1', '2') ,('8', '2', '2') ,('9', '8', '2') ,('10', '2', '5')

,('1', '8', '2') ,('2', '10', '3') ,('3', '6', '3') ,('4', '3', '1') ,('5', '10', '1')

,('6', '1', '5') ,('7', '8', '5') ,('8', '7', '4') ,('9', '8', '5') ,('10', '10', '1')

,('1', '4', '3') ,('2', '5', '2') ,('3', '10', '5') ,('4', '4', '4') ,('5', '2', '3')

,('6', '3', '4') ,('7', '6', '1') ,('8', '7', '4') ,('9', '4', '4') ,('10', '8', '3')

,('1', '9', '5') ,('2', '6', '3') ,('3', '8', '5') ,('4', '4', '2') ,('5', '4', '1')

,('6', '7', '4') ,('7', '7', '2') ,('8', '7', '3') ,('9', '10', '4') ,('10', '10', '3')

,('1', '5', '5') ,('2', '3', '5') ,('3', '2', '4') ,('4', '1', '4') ,('5', '10', '5')

,('6', '3', '2') ,('7', '5', '3') ,('8', '1', '2') ,('9', '2', '5') ,('10', '6', '1')

,('1', '3', '4') ,('2', '4', '2') ,('3', '6', '5') ,('4', '2', '5') ,('5', '6', '1')

,('6', '2', '1') ,('7', '5', '3') ,('8', '8', '1') ,('9', '6', '1') ,('10', '2', '5')

,('1', '5', '4') ,('2', '10', '3') ,('3', '9', '4') ,('4', '6', '5') ,('5', '4', '5')

,('6', '10', '5') ,('7', '10', '3') ,('8', '3', '5') ,('9', '4', '3') ,('10', '7', '1')

-- update new orders for the month of June 2020

INSERT INTO ProductOrder(PersonID, RegistryID, ProductID, DeliveryAddress, SenderAddress, ReturnAddress, ReturnInstructions, ShippingMethod, OrderQuantity, OrderPaymentType, OrderMonth)

VALUES ('1', '1', '2', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '6', 'VISA', 'June')

,('1', '1', '6', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '5', 'VISA', 'June')

,('1', '1', '3', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '1', 'VISA', 'June')

,('2', '1', '13', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'USPS', '1', 'VISA', 'June')

,('2', '1', '4', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'USPS', '4', 'VISA', 'June')

,('3', '2', '13', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '42 Bigcat Lane Orlando FL USA 32789 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '5', 'VISA', 'June')

,('4', '2', '1', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'UPS', '6', 'Discover', 'June')

,('4', '2', '2', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'USPS', '1', 'American Express', 'June')

,('5', '2', '7', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'USPS', '4', 'PayPal', 'June')

,('5', '2', '11', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '6', 'VISA', 'June')

,('5', '3', '16', '222 W 5th Street Houston TX USA 77002', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '3', 'VISA', 'June')

,('6', '3', '11', '222 W 5th Street Houston TX USA 77002', '222 W 5th Street Houston TX USA 77002 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '2', 'Discover', 'June')

,('7', '3', '8', '222 W 5th Street Houston TX USA 77002', '122 E Oak Detroit MI USA 48104 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '4', 'VISA', 'June')

,('8', '3', '13', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'UPS', '3', 'VISA', 'June')

,('8', '3', '8', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'FedEx', '3', 'VISA', 'June')

,('9', '4', '5', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '3', 'VISA', 'June')

,('9', '4', '4', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'USPS', '5', 'VISA', 'June')

,('10', '4', '21', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'UPS', '5', 'Discover', 'June')

,('10', '4', '14', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'FedEx', '1', 'VISA', 'June')

,('10', '4', '17', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '2', 'VISA', 'June')

--update orders for the month of July 2020

INSERT INTO ProductOrder(PersonID, RegistryID, ProductID, DeliveryAddress, SenderAddress, ReturnAddress, ReturnInstructions, ShippingMethod, OrderQuantity, OrderPaymentType, OrderMonth)

VALUES ('1', '1', '16', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '6', 'VISA', 'July')

,('1', '1', '10', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '5', 'VISA', 'July')

,('1', '1', '8', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '1', 'VISA', 'July')

,('2', '1', '25', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'USPS', '1', 'VISA', 'July')

,('2', '1', '6', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'USPS', '4', 'VISA', 'July')

,('3', '2', '23', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '42 Bigcat Lane Orlando FL USA 32789 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '5', 'VISA', 'July')

,('4', '2', '27', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'UPS', '6', 'Discover', 'July')

,('4', '2', '3', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'USPS', '1', 'American Express', 'July')

,('5', '2', '23', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'USPS', '4', 'PayPal', 'July')

,('5', '2', '15', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '6', 'VISA', 'July')

,('5', '3', '3', '222 W 5th Street Houston TX USA 77002', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '3', 'VISA', 'July')

,('6', '3', '1', '222 W 5th Street Houston TX USA 77002', '222 W 5th Street Houston TX USA 77002 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '2', 'Discover', 'July')

,('7', '3', '24', '222 W 5th Street Houston TX USA 77002', '122 E Oak Detroit MI USA 48104 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '4', 'VISA', 'July')

,('8', '3', '27', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'UPS', '3', 'VISA', 'July')

,('8', '3', '24', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'FedEx', '3', 'VISA', 'July')

,('9', '4', '10', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '3', 'VISA', 'July')

,('9', '4', '13', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'USPS', '5', 'VISA', 'July')

,('10', '4', '4', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'UPS', '5', 'Discover', 'July')

,('10', '4', '13', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'FedEx', '1', 'VISA', 'July')

,('10', '4', '16', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '2', 'VISA', 'July')

--insert orders for august 2020

INSERT INTO ProductOrder(PersonID, RegistryID, ProductID, DeliveryAddress, SenderAddress, ReturnAddress, ReturnInstructions, ShippingMethod, OrderQuantity, OrderPaymentType, OrderMonth)

VALUES ('1', '1', '27', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '6', 'VISA', 'August')

,('1', '1', '12', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '5', 'VISA', 'August')

,('1', '1', '5', '1 Federal Way Chicago IL USA 60606', '317 E 1st Street Chicago IL USA 60606 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '1', 'VISA', 'August')

,('2', '1', '10', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'USPS', '1', 'VISA', 'August')

,('2', '1', '16', '1 Federal Way Chicago IL USA 60606', '1 Federal Way Chicago IL USA 60606 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'USPS', '4', 'VISA', 'August')

,('3', '2', '24', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '42 Bigcat Lane Orlando FL USA 32789 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '5', 'VISA', 'August')

,('4', '2', '8', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'UPS', '6', 'Discover', 'August')

,('4', '2', '27', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'USPS', '1', 'American Express', 'August')

,('5', '2', '18', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'USPS', '4', 'PayPal', 'August')

,('5', '2', '15', '137 Chicago Ave Apt #2 Chicago IL USA 60606', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '6', 'VISA', 'August')

,('5', '3', '13', '222 W 5th Street Houston TX USA 77002', '137 Chicago Ave Apt #2 Chicago IL USA 60606 ', '517 W Sequoia Ave, Chicago, IL 60606', 'Send it on back', 'USPS', '3', 'VISA', 'August')

,('6', '3', '13', '222 W 5th Street Houston TX USA 77002', '222 W 5th Street Houston TX USA 77002 ', '212 W School Ave, Dallas, TX, 77104', 'Return ASAP', 'USPS', '2', 'Discover', 'August')

,('7', '3', '9', '222 W 5th Street Houston TX USA 77002', '122 E Oak Detroit MI USA 48104 ', '500 E 17th Ave, Chicago, IL 60606', 'Only 20% of your money will be returned', 'USPS', '4', 'VISA', 'August')

,('8', '3', '13', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '90 Knife Street, Ann Arbor, MI, 48104', 'Good luck finding the place', 'UPS', '3', 'VISA', 'August')

,('8', '3', '16', '222 W 5th Street Houston TX USA 77002', '55 Elm Street #203 Bozeman MT USA 59718 ', '75 W 75th Avenue, New York, NY, 10777', 'Easy come, easy go', 'FedEx', '3', 'VISA', 'August')

,('9', '4', '25', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '68 South Forest Street, Aurora, CA, 50211', 'dont register for it next time', 'USPS', '3', 'VISA', 'August')

,('9', '4', '25', '55 Elm Street #203 Bozeman MT USA 59718', '55 Elm Street #203 Bozeman MT USA 59718 ', '17 E Carson Ave, Carson City, NV, 32401', 'tell your friends about us', 'USPS', '5', 'VISA', 'August')

,('10', '4', '19', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '5 John Galt Ave, Chicago, IL 60606', 'perfect gifts', 'UPS', '5', 'Discover', 'August')

,('10', '4', '18', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '45 E Lake Street, Chicago, IL 60606', 'Go Blue', 'FedEx', '1', 'VISA', 'August')

,('10', '4', '9', '55 Elm Street #203 Bozeman MT USA 59718', '5 Wood Ave Pawnee IN USA 46013 ', '202 Industrial Ave, Chicago, IL 60606', 'Come back again', 'USPS', '2', 'VISA', 'August')

--Created dynamic update fields to multiply the order's quantity according to the vendor's price.

--Because the vendor sets their own price and that information is stored in the product table the is a call to the price according to the product ID

UPDATE ProductOrder

SET OrderSubtotal = (ProductOrder.OrderQuantity \* (SELECT VendorPrice FROM Product

where Product.ProductID=ProductOrder.ProductID))

--Similar to above this is an update statement to multiply the order subtotal \* the 0.02 commission rate

UPDATE ProductOrder

SET CommissionRate = (ProductOrder.OrderSubtotal \* 0.02)

--A vendor has informed us that they have moved their office and shipping facility so we will need to update their profile

--Subsequently, the vendor's address will also need to be updated in the order form's return address field

--Following this line of logic the same would hold true for a person updating their address - in a little we will

--introduce a function to streamline the update of a person's address in a different table

UPDATE Vendor

SET VendorAddress = '84 Minor Fifth Street, Denver, CO USA 80204'

WHERE VendorID = 4

--Update product order table with the newly provided vendor address.

UPDATE ProductOrder

SET ReturnAddress = '84 Minor Fifth Street, Denver, CO USA 80204'

WHERE ReturnAddress = '90 Knife Street, Ann Arbor, MI, 48104'

--A new person has joined Remote Registry

INSERT INTO Person(UserName, FirstName,LastName, PersonAge, StreetAddress, StreetAddress2, CityAddress, StateAddress, CountryAddress, ZipAddress, PhoneNumber, EmailAddress, FutureSpouse)

VALUES ('modestshark8', 'Chris', 'Loo', '29', '252 W Green Ave', '', 'Charleston', 'SC', 'USA', '20290', '4038809981', 'cloo@gmial.com', null)

--But they accidentally mistyped their street address and needs updating

UPDATE Person

SET StreetAddress = '252 W Blue Ave'

WHERE Person.UserName ='modestshark8'

--Because there are many moving pieces to get from a person to their registry and the items that make up that registry

--it is necessary to create a repeatable view to answer a simple question like "what items are on my registry?"

GO

CREATE VIEW RegistryDetail

AS

SELECT

Person.UserName as UserName

, Person.FirstName + ' ' + Person.LastName as FullName

, Product.ProductName as ProductName

, Product.ProductCategory as Category

, Product.ProductBrand as Brand

, Product.ProductDescription as ProductDescription

, Product.ProductDetail as Detail

, RegistryProduct.ProductQuantity as Quantity

, RegistryProduct.FulfilledStatus as FulfilledStatus

FROM Registry

JOIN RegistryProduct on RegistryProduct.RegistryID = Registry.RegistryID

JOIN Product on Product.ProductID = RegistryProduct.ProductID

JOIN Person on Person.PersonID = Registry.PersonID

WHERE Person.UserName = 'grumpyRat1' --specifically defining the user so only a single registry is returned

GO

SELECT\*FROM RegistryDetail --call newly created view and view all fields

--Second we anticipate that end users will want to see a monthly snapshot of orders and ultimately the commission rate

--this view will allow a user to automatically query sales for any given month

GO

CREATE VIEW MonthlyOrders

AS

SELECT

Person.FirstName + ' ' + Person.LastName as FullName

, Product.ProductName as ProductName

, ProductOrder.OrderQuantity as Quantity

, ProductOrder.OrderSubtotal as Subtotal

, ProductOrder.CommissionRate as Commission

FROM ProductOrder

JOIN Person on Person.PersonID = ProductOrder.PersonID

JOIN Product on Product.ProductID = ProductOrder.ProductID

WHERE ProductOrder.OrderMonth = 'July' --specifically defining the month so only a single month's snapshot is returned

GO

SELECT\*FROM MonthlyOrders --call newly created view and view all fields

--Because we know people will move addresses we want there to be a flow between user input and other columns the

--address may be stored. Initially, I will update an address field in the Person table

Update Person

Set StreetAddress = '222 Wabash',StreetAddress2 = 'Unit S', CityAddress = 'Chicago', StateAddress = 'IL'

,CountryAddress = 'USA', ZipAddress = '60606'

WHERE personID = 2

--Additionally, if that person was to order a product in the future we would want the detail from the person

--table to pass into the concatenated sender address within the product order table

GO

CREATE PROCEDURE UpdateAddress(@SenderAddress varchar(256))

AS

BEGIN

UPDATE ProductOrder SET SenderAddress = Person.StreetAddress +' '+Person.StreetAddress2 +' '+ Person.CityAddress +' '+

Person.StateAddress +' '+ Person.CountryAddress +' '+Person.ZipAddress

from ProductOrder

JOIN person on person.personID = productorder.personID

JOIN Registry on Registry.PersonID = Person.PersonID

WHERE Registry.RegistryID = 1

END

GO

--execute the about function on the sender address column. this command will update the Sender

DECLARE @updatecomboaddress varchar(256)

SET @updatecomboaddress = (SELECT SenderAddress FROM ProductOrder

WHERE OrderID=6)

EXEC UpdateAddress @updatecomboaddress

--select \* from Person

--select \* from Product

--join PersonVendorRating on PersonVendorRating.VendorID = Vendor.VendorID

select \* from registry

JOIN RegistryProduct on RegistryProduct.RegistryID = Registry.RegistryID

JOIN Product on Product.ProductID = RegistryProduct.ProductID

--Data Question 1: "What product categories are most popular based on volume different categories appear on registries?

--Second component to the question: "What brands are most popular? (based on volume different brands appear on registries"\

--I actually want this to be a view because I am going to leverage Power Query in Excel to create a supporting chart

-- + I only feel like double clicking on the view I want...

--Within the view is a select statement that counts the products in the table "Product" according to their category

GO

CREATE VIEW CategoryCount AS

SELECT

Product.ProductCategory,

COUNT(Product.ProductCategory) as ProductCount

FROM RegistryProduct

JOIN Product ON Product.ProductID = RegistryProduct.ProductID

GROUP BY Product.ProductCategory --The select statement is grouped by category to make the output easier to interpret

GO

--Similar to the above logic, I am creating a repeatable view to make my life easier. Within this view is a select statement

--to analyze the brands that are most popular

GO

CREATE VIEW BrandCount AS

SELECT

Product.ProductBrand,

COUNT(Product.ProductBrand) as BrandCount

FROM RegistryProduct

JOIN Product ON Product.ProductID = RegistryProduct.ProductID

GROUP BY Product.ProductBrand --The select statement is grouped by category to make the output easier to interpret

GO

--Data Question 2: "What is the average age of a person with a registry?"

--Second component to the quetion: "What product categories are most popular by age?"

--I am taking the average but I have to link this information to the registry table seeing my question is about

--age relative to those who have a registry. The result is 38 years old with a registry. Note the following select on

--all users...whether they have a registry or not:

--SELECT

--AVG(Person.PersonAge) as AverageAge

--FROM Person

--Yields an average age of 33

SELECT

AVG(Person.PersonAge) as AverageAge

FROM Registry

JOIN Person on Registry.PersonID = Person.PersonID

--This select statement is reaching across multiple tables with joins

--The ultimate goal is linking the Person table (containing age) to the Product table (containing product details)

--These two tables are linked via the registryproduct table and the registry table

--Similar to the above, I am going to turn this into a view so I can easily access it in Power Query

GO

CREATE VIEW ProductByAge AS

SELECT

Product.ProductCategory,

Person.PersonAge,

COUNT(Product.ProductCategory) as ProductCount

FROM RegistryProduct

JOIN Product ON Product.ProductID = RegistryProduct.ProductID

JOIN Registry ON Registry.RegistryID = RegistryProduct.RegistryID

JOIN Person ON Person.PersonID = Registry.PersonID

GROUP BY Person.PersonAge , Product.ProductCategory --This grouping is to ensure the results are able to be interpreted

GO

--Data Question 3: "What are the top three vendors based on their rating by people?"

--Selecting the top three results according to the criteria of best average rating

--For the sake of interpretation it is important to pull the vendors name into this select statement

--so it is easy for the person to interpret the vendor that owns the rating

SELECT

TOP 3

AVG(PersonVendorRating.Rating) as VendorAverageRating,

Vendor.VendorName

FROM PersonVendorRating

JOIN Vendor on Vendor.VendorID = PersonVendorRating.VendorID

GROUP BY Vendor.VendorName

ORDER BY VendorAverageRating DESC

--Data Question 4: "Are any vendors overpriced relative to their peers for the same product?"

GO

--In this statement I am selecting columns to be able to identify the vendor, identify the product and pull back the pricing

--with this data I will be able to create a report to see side-by-side whether any of the vendors charge more for the same product

CREATE VIEW VendorPriceCompare AS

SELECT

Product.ProductName,

Product.VendorPrice,

Vendor.VendorName

FROM Product

JOIN Vendor on Vendor.VendorID = Product.VendorID

GO

--Data question 5: "What is the revenue generated by the commission on a month by month basis?"

--Second component: "What is the total revenue for all time periods?"

--This view will create for convenience is the Monthly Profit view. This will sum all the underlying

--commissions on underlying orders and group by month to provide a nice snapshot of monthly commissions earned

CREATE VIEW MonthlyProfit AS

SELECT ProductOrder.OrderMonth,

SUM(ProductOrder.CommissionRate) as TotalCommission

FROM ProductOrder

GROUP BY OrderMonth

GO

SELECT \* FROM MonthlyProfit

--Finally, this following select statement can be used to pull down the sum of all the commission earned for all time periods

SELECT

SUM(ProductOrder.CommissionRate) as TotalCommission

FROM ProductOrder