

**CST2340**

**Database Systems: - Design and Implementation 2020-21**

**Coursework 1: - Part C**

**Advanced Queries**

**Module Tutor: - Ms. Engie Bashir**

**Done By: -**

**Alester D’Costa (M00734829)**

Table of Contents

[Relational Model 3](#_Toc63331296)

[C1. Query using JOIN statements 4](#_Toc63331297)

[C2. Query using JOIN statements along with conditions 5](#_Toc63331298)

[C3. Query using a sub-query 6](#_Toc63331299)

[C4. Query using a self-join 7](#_Toc63331300)

[C5. Query using the Group By and Having commands along with aggregate commands 8](#_Toc63331301)

[C6. Query using the UNION relational algebra commands 9](#_Toc63331302)

[C7. Interactive query 10](#_Toc63331303)

[C8. Query that uses either Exists or Non-Exists 11](#_Toc63331304)

[C9. Query that uses a correlated sub-query 12](#_Toc63331305)

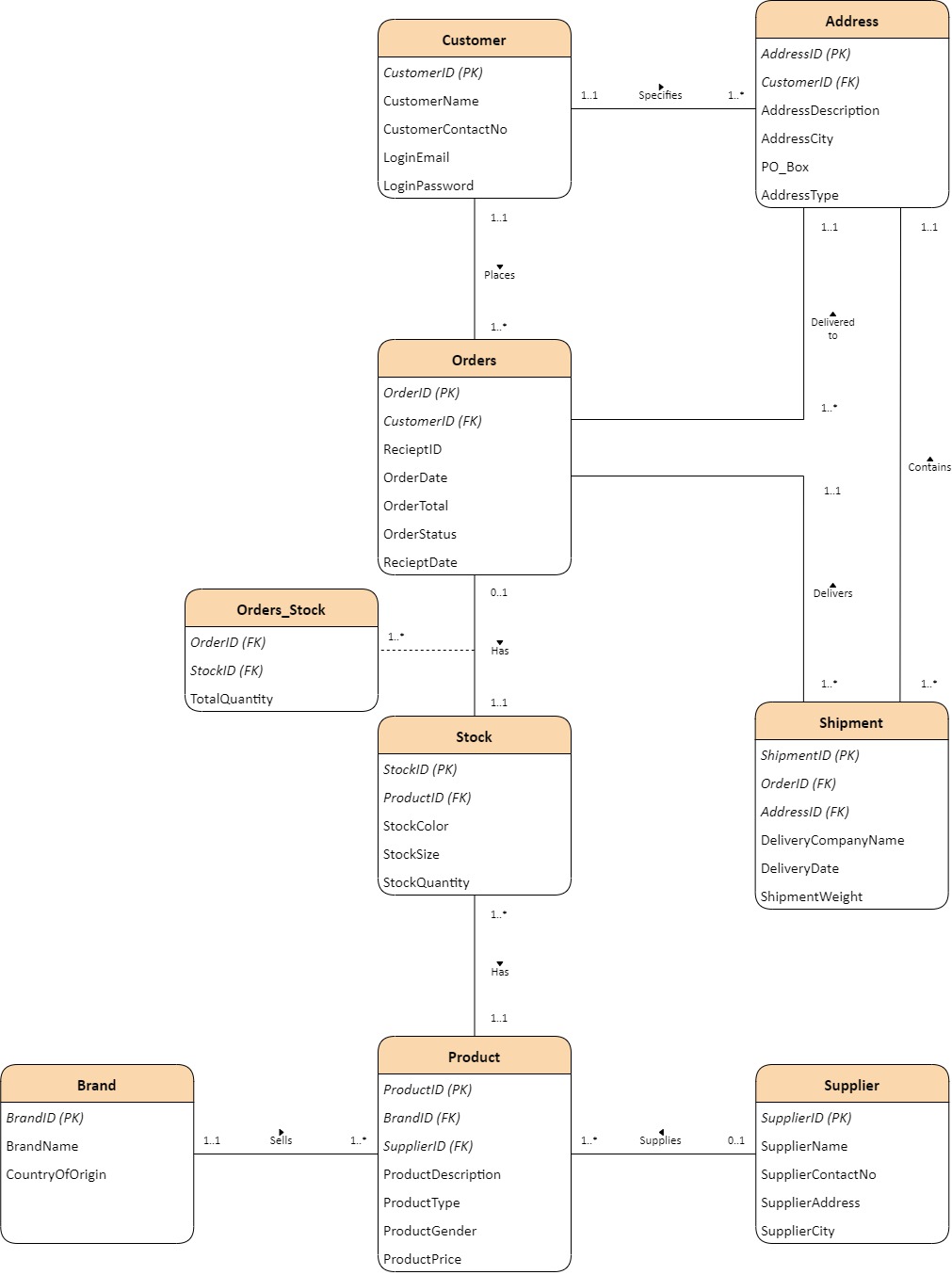
[C10. Query that is an example of a relational algebra divide 13](#_Toc63331306)

[Appendixes 14](#_Toc63331307)

[a. Appendix A 14](#_Toc63331308)

[b. Appendix B 17](#_Toc63331309)

# Relational Model



# C1. Query using JOIN statements

/\*Question C1: - Query using JOIN Statements on three different tables\*/

/\*List customer details (ID and Name), the order details (ID, order date and status) and the shipment details (delivery date delivery company name) in which each customer has placed an order and is arranged in the ascending order of OrderID\*/

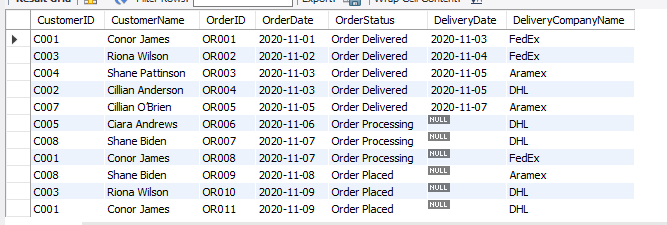
SELECT C.CustomerID, C.CustomerName, O.OrderID, O.OrderDate, O.OrderStatus, SH.DeliveryDate, SH.DeliveryCompanyName

FROM Customer AS C

INNER JOIN Orders AS O ON C.CustomerID=O.CustomerID

INNER JOIN Shipment AS SH ON O.OrderID=SH.OrderID

ORDER BY OrderID ASC;



# C2. Query using JOIN statements along with conditions

/\*Question C2: - Query using JOIN Statements on three different tables with conditions\*/

/\*List brand details (ID and name) and product description for each product supplied by the supplier with the name JD Sports\*/

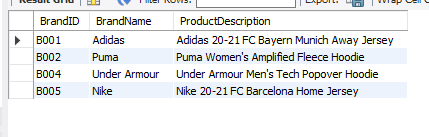
SELECT B.BrandID, B.BrandName, P.ProductDescription

FROM Supplier AS SU

INNER JOIN Product AS P ON SU.SupplierID=P.SupplierID

INNER JOIN Brand AS B ON P.BrandID=B.BrandID

AND SU.SupplierName="JD Sports";



# C3. Query using a sub-query

/\*Question C3: - Query using a sub-query\*/

/\*List all customers who have not placed any order\*/

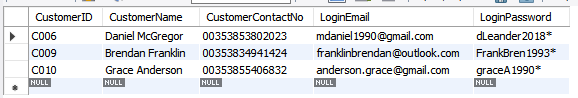
SELECT \*

FROM Customer

WHERE CustomerID NOT IN

(SELECT CustomerID

FROM Orders);



# C4. Query using a self-join

/\*Question C4: - Query using a self-join\*/

/\*List all products which have the same product type as product description “Levi’s Penelope Denim Shirt”\*/

SELECT P2.\*

FROM Product AS P1, Product AS P2

WHERE P1.ProductType=P2.ProductType

AND P1.ProductDescription="Levi’s Penelope Denim Shirt"

AND P2.ProductDescription<>"Levi’s Penelope Denim Shirt";



# C5. Query using the Group By and Having commands along with aggregate commands

/\*Question C5: - Query using the Group By and Having commands along with aggregate commands\*/

/\*List the customer details (ID and name) and sum of the order total where customers have placed more than one order\*/

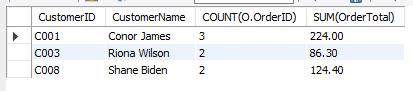
SELECT C.CustomerID, C.CustomerName, COUNT(O.OrderID), SUM(OrderTotal)

FROM Customer AS C

INNER JOIN Orders AS O ON C.CustomerID=O.CustomerID

GROUP BY C.CustomerID, C.CustomerName

HAVING COUNT(O.OrderID)>1;



# C6. Query using the UNION relational algebra commands

/\*Question C6: - Query using the UNION relational algebra commands\*/

/\*List those customers who have specified their address type as company or list those customers whose orders are to be delivered by the delivery company DHL or both\*/

SELECT C.CustomerName

FROM Customer AS C, Address AS A

WHERE C.CustomerID=A.CustomerID

AND A.AddressType="Company"

UNION

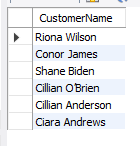
SELECT C.CustomerName

FROM Customer AS C, Orders AS O, Shipment AS SH

WHERE C.CustomerID=O.CustomerID

AND O.OrderID=SH.OrderID

AND SH.DeliveryCompanyName="DHL";



# C7. Interactive query

/\*Question C7: - Execute an interactive query that includes joining at least two tables\*/

/\*List the brand names and product details (description, type, gender and price) whose price is less than 25 euros and is ordered by price in ascending order\*/

/\*Run\*/

SET @ProductPrice=25;

/\*Execution of query\*/

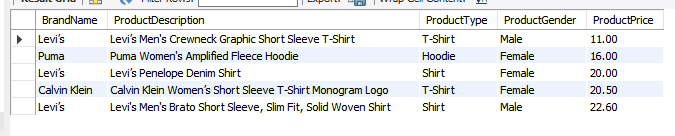
SELECT B.BrandName, P.ProductDescription, P.ProductType, P.ProductGender, P.ProductPrice

FROM Brand AS B

INNER JOIN Product AS P ON B.BrandID=P.BrandID

WHERE ProductPrice<@ProductPrice

ORDER BY ProductPrice ASC;



# C8. Query that uses either Exists or Non-Exists

/\*Question C8: - Query that uses either Exists or Non-Exists\*/

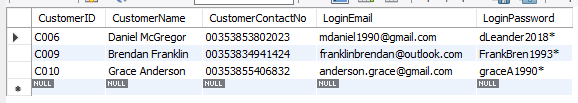
/\*List all those customers who have not specified an address\*/

SELECT \*

FROM Customer AS C

WHERE NOT EXISTS (SELECT \* from Address AS A

WHERE C.CustomerID=A.CustomerID);



# C9. Query that uses a correlated sub-query

/\*Question C9: - Query that uses a correlated sub-query\*/

/\*List the brand name and product details (ID, description, price) in which the price of a product is greater than the average price of products within their brand\*/

SELECT BrandName, ProductID, ProductDescription, ProductPrice

FROM Product AS P1

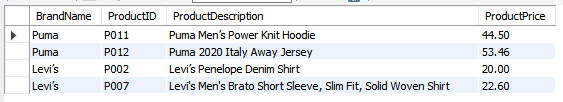
INNER JOIN Brand AS B1 ON P1.BrandID=B1.BrandID

WHERE ProductPrice >

(SELECT AVG(ProductPrice)

FROM Product AS P2

WHERE P1.BrandID=P2.BrandID);



# C10. Query that is an example of a relational algebra divide

/\*Question C10: - Query that is an example of relational algebra divide\*/

/\*List all the names of those brands whose products are supplied by every supplier\*/

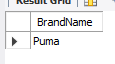
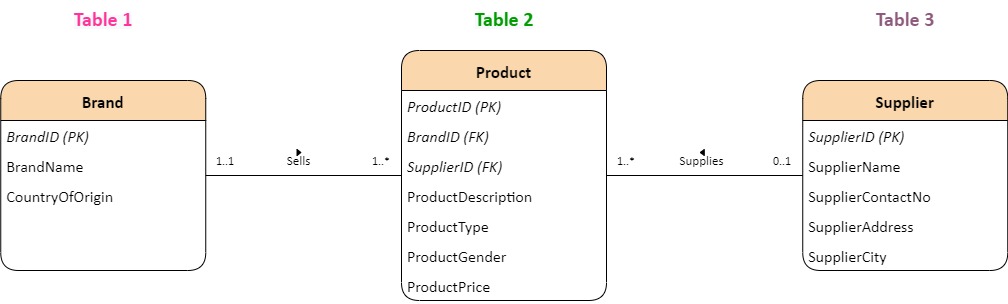
SELECT BrandName from Brand AS B

WHERE NOT EXISTS (SELECT \* from Supplier AS SU

WHERE NOT EXISTS (SELECT \* from Product AS P

WHERE B.BrandID=P.BrandID

and P.SupplierID=SU.SupplierID));



# Appendixes

The appendix has been attached in this coursework as a result of more records being inserted from Part B of the coursework

## **Appendix A**

/\*CREATE TABLE statement\*/

/\*Creates the Brand table\*/

CREATE TABLE Brand(

BrandID varchar(255),

BrandName varchar(30),

CountryOfOrigin varchar(30),

CONSTRAINT pk\_BrandID PRIMARY KEY(BrandID)

);

/\*Creates the Supplier table\*/

CREATE TABLE Supplier(

SupplierID varchar(255),

SupplierName varchar(60),

SupplierContactNo varchar(30),

SupplierAddress varchar(100),

SupplierCity varchar(30),

CONSTRAINT pk\_SupplierID PRIMARY KEY (SupplierID)

);

/\*Creates the Customer table\*/

CREATE TABLE Customer(

CustomerID varchar(255),

CustomerName varchar(60),

CustomerContactNo varchar(30),

LoginEmail varchar(30),

LoginPassword varchar(25),

CONSTRAINT pk\_CustomerID PRIMARY KEY(CustomerID)

);

/\*Creates the Address table\*/

CREATE TABLE Address(

AddressID varchar(255),

CustomerID varchar(255),

AddressDescription varchar(100),

AddressCity varchar(20),

PO\_Box varchar(5),

AddressType varchar(20),

CONSTRAINT pk\_AddressID PRIMARY KEY(AddressID),

CONSTRAINT fk\_CustomerID FOREIGN KEY(CustomerID) REFERENCES Customer(CustomerID)

);

/\*Creates the Product table\*/

CREATE TABLE Product(

ProductID varchar(255),

BrandID varchar(255),

SupplierID varchar(255),

ProductDescription varchar(100),

ProductType varchar(20),

ProductGender varchar(20),

ProductPrice decimal(10,2),

CONSTRAINT pk\_ProductID PRIMARY KEY(ProductID),

CONSTRAINT fk\_BrandID FOREIGN KEY(BrandID) REFERENCES Brand(BrandID),

CONSTRAINT fk\_SupplierID FOREIGN KEY(SupplierID) REFERENCES Supplier(SupplierID)

);

/\*Creates the Stock table\*/

CREATE TABLE Stock(

StockID varchar(255),

ProductID varchar(255),

StockColor varchar(20),

StockSize varchar(20),

StockQuantity int,

CONSTRAINT pk\_StockID PRIMARY KEY(StockID),

CONSTRAINT fk\_ProductID FOREIGN KEY(ProductID) REFERENCES Product(ProductID)

);

/\*Creates the Orders table\*/

CREATE TABLE Orders(

OrderID varchar(255),

CustomerID varchar(255),

RecieptID varchar(255),

OrderDate date,

OrderTotal decimal(10,2),

OrderStatus varchar(30),

RecieptDate date,

CONSTRAINT pk\_OrderID PRIMARY KEY(OrderID),

CONSTRAINT fk2\_CustomerID FOREIGN KEY(CustomerID) REFERENCES Customer(CustomerID)

);

/\*Creates the Shipment Table\*/

CREATE TABLE Shipment(

ShipmentID varchar(255),

OrderID varchar(255),

AddressID varchar(255),

DeliveryCompanyName varchar(30),

DeliveryDate date,

ShipmentWeight decimal(10,2),

CONSTRAINT pk\_ShipmentID PRIMARY KEY(ShipmentID),

CONSTRAINT fk\_OrderID FOREIGN KEY(OrderID) REFERENCES Orders(OrderID),

CONSTRAINT fk\_AddressID FOREIGN KEY(AddressID) REFERENCES Address(AddressID)

);

/\*Creates the Orders\_Stock table\*/

CREATE TABLE Orders\_Stock(

OrderID varchar(255),

StockID varchar(255),

TotalQuantity int,

CONSTRAINT fk2\_OrderID FOREIGN KEY(OrderID) REFERENCES Orders(OrderID),

CONSTRAINT fk\_StockID FOREIGN KEY(StockID) REFERENCES Stock(StockID)

);

## **Appendix B**

/\*INSERT INTO statement\*/

/\*Load Brand with data\*/

INSERT INTO Brand VALUES

("B001", "Adidas", "Germany");

INSERT INTO Brand VALUES

("B002", "Puma", "Germany");

INSERT INTO Brand VALUES

("B003", "Levi’s", "United States");

INSERT INTO Brand VALUES

("B004", "Under Armour", "United States");

INSERT INTO Brand VALUES

("B005", "Nike", "United States");

INSERT INTO Brand VALUES

("B006", "Armani", "Italy");

INSERT INTO Brand VALUES

("B007", "Calvin Klein", "United States");

/\*Extract all records from Brand table\*/

SELECT \* from Brand;

/\*Load Supplier with data\*/

INSERT INTO Supplier VALUES

("SU001", "Caramba", "0035316001000", "Caramba Headquarters, Abbey Street", "Dublin");

INSERT INTO Supplier VALUES

("SU002", "JD Sports", "0035321006321", "JD Sports Complex, St. Patrick Street", "Cork");

INSERT INTO Supplier VALUES

("SU003", "Next", "0035342305810", "The Wallace Building, Wellington Street", "Dundalk");

INSERT INTO Supplier VALUES

("SU004", "ELIE Wholesale", "0035314009393", "The Golden Tower, Amiens Street", "Dublin");

/\*Extract all records from Supplier table\*/

SELECT \* from Supplier;

/\*Load Customer with data\*/

INSERT INTO Customer VALUES

("C001", "Conor James", "00353835674367", "conor.james@gmail.com", "Jamesalisson2020");

INSERT INTO Customer VALUES

("C002", "Cillian Anderson", "00353879648299", "cillian1989@outlook.com", "cillianA121\*");

INSERT INTO Customer VALUES

("C003", "Riona Wilson", "00353867272829", "rionaw1986@yahoo.com", "Wilson.R86\*");

INSERT INTO Customer VALUES

("C004", "Shane Pattinson", "00353833622445", "patto2020@outlook.com", "Pattinson88\*");

INSERT INTO Customer VALUES

("C005", "Ciara Andrews", "00353832405373", "ciara2013@gmail.com", "CiaraFrancis2013");

INSERT INTO Customer VALUES

("C006", "Daniel McGregor", "00353853802023", "mdaniel1990@gmail.com", "dLeander2018\*");

INSERT INTO Customer VALUES

("C007", "Cillian O’Brien", "00353871293223", "cillianob2001@yahoo.com", "obrienC2001\*");

INSERT INTO Customer VALUES

("C008", "Shane Biden", "00353833833024", "shane.biden@gmail.com", "Shaneisthebest2001");

INSERT INTO Customer VALUES

("C009", "Brendan Franklin", "00353834941424", "franklinbrendan@outlook.com", "FrankBren1993\*");

INSERT INTO Customer VALUES

("C010", "Grace Anderson", "00353855406832", "anderson.grace@gmail.com", "graceA1990\*");

/\*Extract all records from Customer table\*/

SELECT \* from Customer;

/\*Load Address with data\*/

INSERT INTO Address VALUES

("AD001", "C001", "The Dorms Building, Building No: 15, Room No: 201, Henry Street", "Dublin", null, "Residential");

INSERT INTO Address VALUES

("AD002", "C003", "Crown Company, Building No: 43, Shandon Street", "Cork", "38923", "Company");

INSERT INTO Address VALUES

("AD003", "C001", "Leyton Traders, Greenwood Building, Building No: 99, Amiens Street", "Dublin", "28289", "Company");

INSERT INTO Address VALUES

("AD004", "C004", "The Den, Building No: 92, Flat No: 117, 14th Baker Street", "Dublin", null, "Residential");

INSERT INTO Address VALUES

("AD005", "C008", "The Italian Knights Building, Building No: 10, Flat No: 204, Leinster Street", "Dundalk", null, "Residential");

INSERT INTO Address VALUES

("AD006", "C005", "Oaklands Building, Building No: 11, Room No: 303, Leinster Street", "Dundalk", "39923", "Residential");

INSERT INTO Address VALUES

("AD007", "C008", "Francesco and Sons, Building No: 94, Baggot Street", "Dublin", "12937", "Company");

INSERT INTO Address VALUES

("AD008", "C002", "The Old Barn, Building No: 23, Flat No: 109, 24-London Street", "Galway", null, "Residential");

INSERT INTO Address VALUES

("AD009", "C007", "Leyton Traders, Greenwood Building, Building No: 99, Amiens Street", "Dublin", null, "Company");

INSERT INTO Address VALUES

("AD010", "C002", "Frankston Building, Building No: 33, Flat No: 401, Thames Street", "Cork", "84834", "Residential");

/\*Extract all records from Address table\*/

SELECT \* from Address;

/\*Load Product with data\*/

INSERT INTO Product VALUES

("P001", "B001", "SU002", "Adidas 20-21 FC Bayern Munich Away Jersey", "T-Shirt", "Male", 65.50);

INSERT INTO Product VALUES

("P002", "B003", "SU001", "Levi’s Penelope Denim Shirt", "Shirt", "Female", 20.00);

INSERT INTO Product VALUES

("P003", "B002", "SU002", "Puma Women's Amplified Fleece Hoodie", "Hoodie", "Female", 16.00);

INSERT INTO Product VALUES

("P004", "B003", "SU004", "Levi’s Men's Crewneck Graphic Short Sleeve T-Shirt", "T-Shirt", "Male", 11.00);

INSERT INTO Product VALUES

("P005", "B007", "SU003", "Calvin Klein Women’s Short Sleeve T-Shirt Monogram Logo", "T-Shirt", "Female", 20.50);

INSERT INTO Product VALUES

("P006", "B006", "SU001", "Armani Exchange Men's Crew Neck Logo Tee", "T-Shirt", "Male", 28.30);

INSERT INTO Product VALUES

("P007", "B003", "SU004", "Levi's Men's Brato Short Sleeve, Slim Fit, Solid Woven Shirt", "Shirt", "Male", 22.60);

INSERT INTO Product VALUES

("P008", "B004", "SU002", "Under Armour [Men's Tech Popover Hoodie](https://www.amazon.com/Under-Armour-1274511-Popover-Hoodie/dp/B00YSPQ5CU/ref=sr_1_9?crid=XCNF42QMN4N3&dchild=1&keywords=under+armour+hoodie+mens&qid=1606650929&sprefix=under+armour+hoo%2Caps%2C352&sr=8-9)", "Hoodie", "Male", 35.20);

INSERT INTO Product VALUES

("P009", "B002", "SU003", "Puma Unisex NRG Triblend Graphic T-Shirt", "T-Shirt", "Unisex", 25.30);

INSERT INTO Product VALUES

("P010", "B005", "SU002", "Nike 20-21 FC Barcelona Home Jersey", "T-Shirt", "Male", 60.00);

INSERT INTO Product VALUES

("P011", "B002", "SU001", "Puma Men’s Power Knit Hoodie", "Hoodie", "Male", 44.50);

INSERT INTO Product VALUES

("P012", "B002", "SU004", "Puma 2020 Italy Away Jersey", "T-Shirt", "Male", 53.46);

/\*Extract all records from Product table\*/

SELECT \* from Product;

/\*Load Stock with data\*/

INSERT INTO Stock VALUES

("S001", "P001", "White", "Small", 20);

INSERT INTO Stock VALUES

("S002", "P001", "White", "Medium", 10);

INSERT INTO Stock VALUES

("S003", "P001", "White", "Large", 10);

INSERT INTO Stock VALUES

("S004", "P001", "White", "Extra Large", 0);

INSERT INTO Stock VALUES

("S005", "P002", "Blue", "Small", 10);

INSERT INTO Stock VALUES

("S006", "P002", "Blue", "Medium", 5);

INSERT INTO Stock VALUES

("S007", "P002", "Blue", "Large", 0);

INSERT INTO Stock VALUES

("S008", "P003", "Black", "Small", 10);

INSERT INTO Stock VALUES

("S009", "P003", "Black", "Medium", 10);

INSERT INTO Stock VALUES

("S010", "P003", "Black", "Large", 8);

INSERT INTO Stock VALUES

("S011", "P004", "White", "Small", 10);

INSERT INTO Stock VALUES

("S012", "P004", "White", "Medium", 10);

INSERT INTO Stock VALUES

("S013", "P004", "White", "Large", 5);

INSERT INTO Stock VALUES

("S014", "P004", "White", "Extra Large", 5);

INSERT INTO Stock VALUES

("S015", "P005", "White", "Small", 15);

INSERT INTO Stock VALUES

("S016", "P005", "White", "Medium", 10);

INSERT INTO Stock VALUES

("S017", "P005", "White", "Large", 5);

INSERT INTO Stock VALUES

("S018", "P006", "Black", "Small", 0);

INSERT INTO Stock VALUES

("S019", "P006", "Black", "Medium", 10);

INSERT INTO Stock VALUES

("S020", "P006", "Black", "Large", 12);

INSERT INTO Stock VALUES

("S021", "P007", "Pink", "Small", 0);

INSERT INTO Stock VALUES

("S022", "P007", "Pink", "Medium", 10);

INSERT INTO Stock VALUES

("S023", "P007", "Pink", "Large", 10);

INSERT INTO Stock VALUES

("S024", "P008", "Gray", "Small", 0);

INSERT INTO Stock VALUES

("S025", "P008", "Gray", "Medium", 10);

INSERT INTO Stock VALUES

("S026", "P008", "Gray", "Large", 5);

INSERT INTO Stock VALUES

("S027", "P009", "White", "Small", 0);

INSERT INTO Stock VALUES

("S028", "P009", "White", "Medium", 1);

INSERT INTO Stock VALUES

("S029", "P009", "White", "Large", 2);

INSERT INTO Stock VALUES

("S030", "P010", "Red", "Small", 2);

INSERT INTO Stock VALUES

("S031", "P010", "Red", "Medium", 5);

INSERT INTO Stock VALUES

("S032", "P010", "Red", "Large", 2);

INSERT INTO Stock VALUES

("S033", "P011", "Black", "Small", 1);

INSERT INTO Stock VALUES

("S034", "P011", "Black", "Medium", 2);

INSERT INTO Stock VALUES

("S035", "P011", "Black", "Large", 2);

INSERT INTO Stock VALUES

("S036", "P012", "White", "Small", 1);

INSERT INTO Stock VALUES

("S037", "P012", "Black", "Medium", 3);

INSERT INTO Stock VALUES

("S038", "P012", "Black", "Large", 1);

/\*Extract all records from Stock table\*/

SELECT \* from Stock;

/\*Load Orders with data\*/

INSERT INTO Orders VALUES

("OR001", "C001", "R001", "2020-11-01", 131.00, "Order Delivered", "2020-11-03");

INSERT INTO Orders VALUES

("OR002", "C003", "R002", "2020-11-02", 20.00, "Order Delivered", "2020-11-04");

INSERT INTO Orders VALUES

("OR003", "C004", "R003", "2020-11-03", 22.00, "Order Delivered", "2020-11-05");

INSERT INTO Orders VALUES

("OR004", "C002", "R004", "2020-11-03", 85.30, "Order Delivered", "2020-11-05");

INSERT INTO Orders VALUES

("OR005", "C007", "R005", "2020-11-05", 28.30, "Order Delivered", "2020-11-07");

INSERT INTO Orders VALUES

("OR006", "C005", null, "2020-11-06", 40.00, "Order Processing", null);

INSERT INTO Orders VALUES

("OR007", "C008", null, "2020-11-07", 67.80, "Order Processing", null);

INSERT INTO Orders VALUES

("OR008", "C001", null, "2020-11-07", 35.20, "Order Processing", null);

INSERT INTO Orders VALUES

("OR009", "C008", null, "2020-11-08", 56.60, "Order Placed", null);

INSERT INTO Orders VALUES

("OR010", "C003", null, "2020-11-09", 66.30, "Order Placed", null);

INSERT INTO Orders VALUES

("OR011", "C001", null, "2020-11-09", 57.80, "Order Placed", null);

/\*Extract all records from Orders table\*/

SELECT \* from Orders;

/\*Load Shipment with data\*/

INSERT INTO Shipment VALUES

("SH001", "OR001", "AD001", "FedEx", "2020-11-03", 0.44);

INSERT INTO Shipment VALUES

("SH002", "OR002", "AD002", "FedEx", "2020-11-04", 0.10);

INSERT INTO Shipment VALUES

("SH003", "OR003", "AD004", "Aramex", "2020-11-05", 0.22);

INSERT INTO Shipment VALUES

("SH004", "OR004", "AD008", "DHL", "2020-11-05", 0.47);

INSERT INTO Shipment VALUES

("SH005", "OR005", "AD009", "Aramex", "2020-11-07", 0.22);

INSERT INTO Shipment VALUES

("SH006", "OR006", "AD006", "DHL", null, 0.36);

INSERT INTO Shipment VALUES

("SH007", "OR007", "AD007", "DHL", null, 0.27);

INSERT INTO Shipment VALUES

("SH008", "OR008", "AD003", "FedEx", null, 0.30);

INSERT INTO Shipment VALUES

("SH009", "OR009", "AD005", "Aramex", null, 0.44);

INSERT INTO Shipment VALUES

("SH010", "OR010", "AD002", "DHL", null, 0.45);

INSERT INTO Shipment VALUES

("SH011", "OR011", "AD001", "DHL", null, 0.39);

/\*Extract all records from Shipment table\*/

SELECT \* from Shipment;

/\*Load Orders\_Stock with data\*/

INSERT INTO Orders\_Stock VALUES

("OR001", "S002", 2);

INSERT INTO Orders\_Stock VALUES

("OR002", "S005", 1);

INSERT INTO Orders\_Stock VALUES

("OR003", "S012", 2);

INSERT INTO Orders\_Stock VALUES

("OR004", "S032", 1);

INSERT INTO Orders\_Stock VALUES

("OR004", "S029", 1);

INSERT INTO Orders\_Stock VALUES

("OR005", "S019", 1);

INSERT INTO Orders\_Stock VALUES

("OR006", "S005", 2);

INSERT INTO Orders\_Stock VALUES

("OR007", "S023", 3);

INSERT INTO Orders\_Stock VALUES

("OR008", "S025", 1);

INSERT INTO Orders\_Stock VALUES

("OR009", "S019", 2);

INSERT INTO Orders\_Stock VALUES

("OR010", "S015", 2);

INSERT INTO Orders\_Stock VALUES

("OR010", "S028", 1);

INSERT INTO Orders\_Stock VALUES

("OR011", "S022", 1);

INSERT INTO Orders\_Stock VALUES

("OR011", "S025", 1);

/\*Extract all records from Orders\_Stock table\*/

SELECT \* from Orders\_Stock;