-- Drop the tables

-- Order Does not matter for these tables as they do not have foreign key constraints

DROP TABLE CartProduct;

DROP TABLE ProductCategory;

DROP TABLE ShoppingCart;

DROP TABLE OrderItem;

-- Later Drop these in Order

DROP TABLE OrderTable;

DROP TABLE CategoryTable;

DROP TABLE Product;

DROP TABLE Customer;

DROP TABLE Address;

DROP TABLE CountryTable;

DROP TABLE ProvinceTable;

-- CREATE THE TABLES:

-- Address table

CREATE TABLE ProvinceTable (

ProvinceID NUMBER(5) PRIMARY KEY,

Province VARCHAR2(50),

City VARCHAR2(50)

);

CREATE TABLE CountryTable (

CountryID NUMBER(5) PRIMARY KEY,

Country VARCHAR2(50),

PostalCode VARCHAR2(10),

ProvinceID NUMBER(5) REFERENCES ProvinceTable(ProvinceID)

);

CREATE TABLE Address (

AddressID NUMBER(5) PRIMARY KEY,

StreetNumber VARCHAR2(10),

StreetName VARCHAR2(100),

CountryID NUMBER(5) REFERENCES CountryTable(CountryID)

);

-- Customer table

CREATE TABLE Customer (

CustomerID NUMBER(5) PRIMARY KEY,

Username VARCHAR2(50) NOT NULL,

Passwd VARCHAR2(50) NOT NULL,

Email VARCHAR2(100),

FirstName VARCHAR2(50),

LastName VARCHAR2(50),

PhoneNumber VARCHAR2(20),

AddressID NUMBER(5) REFERENCES Address(AddressID)

);

-- Product table

CREATE TABLE Product (

ProductID NUMBER(5) PRIMARY KEY,

ProductName VARCHAR2(100) NOT NULL,

ProductDesc VARCHAR2(1000),

Price NUMBER(10, 2) NOT NULL,

Stock NUMBER(5) NOT NULL,

ImageURL VARCHAR2(255)

);

-- Category table

CREATE TABLE CategoryTable (

CategoryID NUMBER(5) PRIMARY KEY,

CategoryName VARCHAR2(100) NOT NULL

);

-- ProductCategory table

CREATE TABLE ProductCategory (

ProductCategoryID NUMBER(5) PRIMARY KEY,

ProductID NUMBER(5) REFERENCES Product(ProductID),

CategoryID NUMBER(5) REFERENCES CategoryTable(CategoryID)

);

-- Order table

CREATE TABLE OrderTable (

OrderID NUMBER(5) PRIMARY KEY,

CustomerID NUMBER(5) REFERENCES Customer(CustomerID),

AddressID NUMBER(5) REFERENCES Address(AddressID),

OrderDate DATE NOT NULL,

OrderTime TIMESTAMP NOT NULL,

TotalPrice NUMBER(10, 2) NOT NULL

);

-- OrderItem table

CREATE TABLE OrderItem (

OrderItemID NUMBER(5) PRIMARY KEY,

OrderID NUMBER(5) REFERENCES OrderTable(OrderID),

ProductID NUMBER(5) REFERENCES Product(ProductID),

Quantity NUMBER(5) NOT NULL,

Subtotal NUMBER(10, 2) NOT NULL

);

-- Create the ShoppingCart table

CREATE TABLE ShoppingCart (

CustomerID NUMBER(5) PRIMARY KEY,

CreationDate DATE NOT NULL,

CreationTime TIMESTAMP NOT NULL

);

-- CartProduct table

CREATE TABLE CartProduct (

CartProductID NUMBER(5) PRIMARY KEY,

CustomerID NUMBER(5) REFERENCES ShoppingCart(CustomerID),

ProductID NUMBER(5) REFERENCES Product(ProductID),

Quantity NUMBER(5) NOT NULL

);

-- POPULATE THE TABLES:

-- Insert data into the Province table

INSERT INTO ProvinceTable (ProvinceID, Province, City)

VALUES (1, 'Ontario', 'Toronto');

INSERT INTO ProvinceTable (ProvinceID, Province, City)

VALUES (2, 'New York', 'New York');

INSERT INTO ProvinceTable (ProvinceID, Province, City)

VALUES (3, 'California', 'Los Angeles');

INSERT INTO ProvinceTable (ProvinceID, Province, City)

VALUES (4, 'Illinois', 'Chicago');

INSERT INTO ProvinceTable (ProvinceID, Province, City)

VALUES (5, 'Texas', 'Houston');

-- Insert data into the Country table

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (1, 'Canada', 'P7H0A8', 1);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (2, 'USA', '10001', 2);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (3, 'USA', '90001', 3);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (4, 'USA', '60601', 4);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (5, 'USA', '77001', 5);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (6, 'Canada', 'M1Z001', 1);

INSERT INTO CountryTable (CountryID, Country, PostalCode, ProvinceID)

VALUES (7, 'USA', '90101', 2);

-- Insert data into the Address table

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (1, '123', 'Main St', 1);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (2, '456', 'Oak Avenue', 2);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (3, '789', 'Cedar Lane', 3);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (4, '1010', 'Maple Street', 4);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (5, '234', 'Pine Avenue', 5);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (6, '134', 'Brooke Road',1);

INSERT INTO Address (AddressID, StreetNumber, StreetName, CountryID)

VALUES (7, '134', 'Brooklyn Street', 2);

-- Insert data for Customer table

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (1, 'user1', 'password1', 'user1@example.com', 'John', 'Doe', '657-123-4567', 1);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (2, 'user2', 'password2', 'user2@example.com', 'Alice', 'Smith', '987-654-3210', 2);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (3, 'user3', 'password3', 'user3@example.com', 'Bob', 'Johnson', '555-123-4567', 3);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (4, 'user4', 'password4', 'user4@example.com', 'Emily', 'Brown', '444-789-0123', 4);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (5, 'user5', 'password5', 'user5@example.com', 'Grace', 'Lee', '111-222-3333', 5);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (6, 'user6', 'password6', 'user6@example.com', 'Bob', 'Marley', '111-444-3333', 6);

INSERT INTO Customer (CustomerID, Username, Passwd, Email, FirstName, LastName, PhoneNumber, AddressID)

VALUES (7, 'user7', 'password7', 'user7@example.com', 'Bob', 'Dylan', '000-434-2333', 7);

-- Insert data forProduct table

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (1, 'Men''s Cotton Shirt', 'High-quality cotton shirt for men', 29.99, 150, 'shirt.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (2, 'Women''s Leather Handbag', 'Stylish leather handbag for women', 79.99, 80, 'handbag.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (3, 'Organic All-Purpose Flour', 'Certified organic wheat flour', 5.99, 200, 'flour.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (4, 'Coca-Cola Classic', 'Classic carbonated soft drink', 1.99, 500, 'cocacola.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (5, 'Apple iPhone 13', 'Latest model with advanced features', 999.99, 30, 'iphone.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (6, 'Samsung 55" 4K Smart TV', 'Crystal-clear 4K Ultra HD television', 799.99, 10, 'tv.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (7, 'Bluetooth Wireless Earbuds', 'Wireless earbuds with long battery life', 49.99, 100, 'earbuds.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (8, 'Dell Inspiron Laptop', 'Powerful laptop for work and entertainment', 899.99, 40, 'laptop.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (9, 'Fitness Tracker Watch', 'Track your health and fitness activities', 29.99, 120, 'fitness.jpg');

INSERT INTO Product (ProductID, ProductName, ProductDesc, Price, Stock, ImageURL)

VALUES (10, 'Chocolate Chip Cookies', 'Delicious homemade chocolate chip cookies', 3.99, 300, 'cookies.jpg');

-- Insert data for the CategoryTable

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (1, 'Clothing');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (2, 'Fashion Accessories');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (3, 'Electronics');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (4, 'Food');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (5, 'Fitness Equipment');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (6, 'Beverages');

INSERT INTO CategoryTable (CategoryID, CategoryName)

VALUES (7, 'Home Appliances');

-- Insert data ProductCategory table

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (1, 1, 1);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (2, 1, 2);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (3, 2, 1);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (4, 3, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (5, 4, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (6, 4, 6);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (7, 5, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (8, 5, 2);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (9, 6, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (10, 6, 7);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (11, 7, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (12, 8, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (13, 9, 2);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (14, 9, 3);

INSERT INTO ProductCategory (ProductCategoryID, ProductID, CategoryID)

VALUES (15, 10, 4);

-- Insert data for OrderTable

INSERT INTO OrderTable (OrderID, CustomerID, AddressID, OrderDate, OrderTime, TotalPrice)

VALUES (1, 1, 1, TO\_DATE('2022-09-25', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP, 10.99);

INSERT INTO OrderTable (OrderID, CustomerID, AddressID, OrderDate, OrderTime, TotalPrice)

VALUES (2, 2, 2, TO\_DATE('2020-09-16', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP, 39.99);

INSERT INTO OrderTable (OrderID, CustomerID, AddressID, OrderDate, OrderTime, TotalPrice)

VALUES (3, 3, 3, TO\_DATE('2023-01-07', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP, 17.49);

INSERT INTO OrderTable (OrderID, CustomerID, AddressID, OrderDate, OrderTime, TotalPrice)

VALUES (4, 4, 4, TO\_DATE('2023-10-18', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP, 24.95);

INSERT INTO OrderTable (OrderID, CustomerID, AddressID, OrderDate, OrderTime, TotalPrice)

VALUES (5, 5, 5, TO\_DATE('2023-02-27', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP, 35.99);

-- Insert data into the OrderItem table

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (1, 1, 2, 2, 21.98);

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (2, 1, 5, 3, 59.97);

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (3, 2, 3, 1, 7.49);

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (4, 2, 4, 2, 49.90);

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (5, 3, 5, 4, 63.96);

INSERT INTO OrderItem (OrderItemID, OrderID, ProductID, Quantity, Subtotal)

VALUES (6, 4, 8, 1, 33.96);

-- Insert data into the ShoppingCart table

INSERT INTO ShoppingCart (CustomerID, CreationDate, CreationTime)

VALUES (1, TO\_DATE('2020-09-25', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP);

INSERT INTO ShoppingCart (CustomerID, CreationDate, CreationTime)

VALUES (2, TO\_DATE('2023-09-26', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP);

INSERT INTO ShoppingCart (CustomerID, CreationDate, CreationTime)

VALUES (3, TO\_DATE('2023-09-27', 'YYYY-MM-DD'), CURRENT\_TIMESTAMP);

-- Insert data into the CartProduct table

INSERT INTO CartProduct (CartProductID, CustomerID, ProductID, Quantity)

VALUES (1, 1, 1, 3);

INSERT INTO CartProduct (CartProductID, CustomerID, ProductID, Quantity)

VALUES (2, 1, 2, 2);

INSERT INTO CartProduct (CartProductID, CustomerID, ProductID, Quantity)

VALUES (3, 2, 3, 1);

-- Simple Queries for A4

-- List all attributes of all products

SELECT \* FROM Product;

SELECT \* FROM Address;

SELECT \* FROM cartproduct;

SELECT \* FROM categorytable;

SELECT \* FROM customer;

SELECT \* FROM orderitem;

SELECT \* FROM ordertable;

SELECT \* FROM productcategory;

SELECT \* FROM shoppingcart;

---- List Product Name and their Price for all products

--SELECT 'The Product: ', ProductName, ' costs ', Price

--FROM Product;

--

---- List the Unique First Name of all the customers

--SELECT DISTINCT FirstName FROM Customer;

--

---- List all attributes of products that costs more than $100.0 or

---- have more than 200 items in stock.

--SELECT \* FROM Product

--WHERE Price > 100.0 OR Stock > 200;

--

---- List all the Subtotal of each product purchased by Customer1.

--SELECT Subtotal AS Purchases\_by\_Customer1 FROM OrderItem

--WHERE OrderID = 1;

--

---- List the Order ID and Total Price of all orders sorted by Order

---- Date in ascending order (Earliest to Latest).

--SELECT OrderID, TotalPrice FROM OrderTable

--ORDER BY OrderDate;

--

---- Count the number of customers living in each City from all the

---- addresses.

--SELECT COUNT(AddressID) AS Number\_Of\_Customers, City FROM Address

--GROUP BY City;

--

---- List the username and password of customer with an id greater than 3

--SELECT username, passwd

--FROM customer

--WHERE customerid>3;

--

---- List the names of products that are popular (customers got more than one of)

--SELECT productname

--FROM product, cartproduct

--Where quantity > 1

-- AND cartproduct.productid=product.productid;

--

---- List the customer id of customers that have spent less than $25 on their order

--SELECT 'Customer ID is: ', customerid

--FROM ordertable

--WHERE totalprice < 25;

--

---- List the order IDs for orders that spent more than $50

--SELECT orderid

--FROM orderitem

--WHERE subtotal>50;

--

---- List information of customers with an active shopping cart

--SELECT c.\*

--FROM customer c, shoppingcart s

--WHERE c.customerid=s.customerid;

--

---- List the address id of customers that live in Toronto

--SELECT addressid

--FROM address

--WHERE city='Toronto';

--

---- List all product ID's that are within the Clothing category

--SELECT productid

--FROM productcategory

--WHERE categoryid='1';

--

---- List the category ID for the food category

--SELECT categoryid

--FROM categorytable

--WHERE categoryname='food';

--

---- Count the number of products in each category.

--SELECT CategoryName, COUNT(ProductCategoryID) AS NumberOfProducts

--FROM CategoryTable

--LEFT JOIN ProductCategory ON CategoryTable.CategoryID = ProductCategory.CategoryID

--GROUP BY CategoryName;

--

---- List the total number of products in each shopping cart.

--SELECT ShoppingCart.CustomerID, COUNT(Quantity) AS TotalProductsInCart

--FROM ShoppingCart

--INNER JOIN CartProduct ON ShoppingCart.CustomerID = CartProduct.CustomerID

--GROUP BY ShoppingCart.CustomerID;

--

---- List the products along with their quantities in the shopping cart of a specific customer.

---- In this query, we used tablesm aliases cp and p for a more advanced looking query

--SELECT cp.CustomerID, p.ProductName, cp.Quantity

--FROM CartProduct cp

--JOIN Product p ON cp.ProductID = p.ProductID

--WHERE cp.CustomerID = 1;

--

---- List customers who made orders and their respective cities:

--SELECT c.CustomerID, c.FirstName || ' ' || c.LastName AS CustomerName, a.City

--FROM Customer c

--JOIN Address a ON c.AddressID = a.AddressID

--WHERE c.CustomerID IN (SELECT DISTINCT CustomerID FROM OrderTable);

--

---- List products and their categories:

--SELECT p.ProductName, ct.CategoryName

--FROM Product p

--JOIN ProductCategory pc ON p.ProductID = pc.ProductID

--JOIN CategoryTable ct ON pc.CategoryID = ct.CategoryID;

--

---- List customers and their total spending:

--SELECT c.CustomerID, c.FirstName || ' ' || c.LastName AS CustomerName, SUM(o.TotalPrice) AS TotalSpending

--FROM Customer c

--LEFT JOIN OrderTable o ON c.CustomerID = o.CustomerID

--GROUP BY c.CustomerID, c.FirstName, c.LastName;

--

---- List customers who have both placed orders and added products to their shopping cart:

--SELECT c.CustomerID, c.FirstName || ' ' || c.LastName AS CustomerName

--FROM Customer c

--WHERE EXISTS (SELECT 1 FROM OrderTable o WHERE o.CustomerID = c.CustomerID)

--AND EXISTS (SELECT 1 FROM CartProduct cp WHERE cp.CustomerID = c.CustomerID);

--

--

----This query joins the OrderTable, Customer, and Address tables and orders the results by the order date in descending order.

--SELECT o.OrderID, o.OrderDate, o.TotalPrice, c.FirstName || ' ' || c.LastName AS CustomerName, a.City, a.Province, a.Country

--FROM OrderTable o

--JOIN Customer c ON o.CustomerID = c.CustomerID

--JOIN Address a ON o.AddressID = a.AddressID

--ORDER BY o.OrderDate DESC;

--

--

--

---- VIEWS:

--

---- This view combines information from the Customer, OrderTable, and OrderItem tables to provide details about customer orders.

--CREATE VIEW CustomerOrderDetails AS

--SELECT o.OrderID, o.OrderDate, o.TotalPrice, c.FirstName || ' ' || c.LastName AS CustomerName, p.ProductName, oi.Quantity, oi.Subtotal

--FROM OrderTable o

--JOIN Customer c ON o.CustomerID = c.CustomerID

--JOIN OrderItem oi ON o.OrderID = oi.OrderID

--JOIN Product p ON oi.ProductID = p.ProductID;

--

--

---- This view shows products that are popular (ordered more than once) along with the total quantity ordered.

--CREATE VIEW PopularProducts AS

--SELECT p.ProductID, p.ProductName, SUM(oi.Quantity) AS TotalOrdered

--FROM Product p

--JOIN OrderItem oi ON p.ProductID = oi.ProductID

--GROUP BY p.ProductID, p.ProductName

--HAVING SUM(oi.Quantity) > 1;

--

--

---- This view counts the number of customers in each city.

--CREATE VIEW CustomerCityCount AS

--SELECT a.City, COUNT(c.CustomerID) AS CustomerCount

--FROM Customer c

--JOIN Address a ON c.AddressID = a.AddressID

--GROUP BY a.City;

--

--

---- This view lists customers with active shopping carts.

--CREATE VIEW ActiveShoppingCarts AS

--SELECT c.CustomerID, c.FirstName || ' ' || c.LastName AS CustomerName, s.CreationDate, s.CreationTime

--FROM Customer c

--JOIN ShoppingCart s ON c.CustomerID = s.CustomerID;

--

--

---- This view identifies high-value customers based on their total spending.

--CREATE VIEW HighValueCustomers AS

--SELECT c.CustomerID, c.FirstName || ' ' || c.LastName AS CustomerName, SUM(o.TotalPrice) AS TotalSpending

--FROM Customer c

--LEFT JOIN OrderTable o ON c.CustomerID = o.CustomerID

--GROUP BY c.CustomerID, c.FirstName, c.LastName

--HAVING SUM(o.TotalPrice) > 100;

--

--

---- This view identifies product categories with high total sales based on the quantity sold.

--CREATE VIEW HighValueProductCategories AS

--SELECT pc.CategoryID, ct.CategoryName, SUM(oi.Quantity) AS TotalQuantitySold

--FROM ProductCategory pc

--JOIN OrderItem oi ON pc.ProductID = oi.ProductID

--JOIN CategoryTable ct ON pc.CategoryID = ct.CategoryID

--GROUP BY pc.CategoryID, ct.CategoryName

--HAVING SUM(oi.Quantity) > 50;