Lab: Creating SQL Tables and Loading Data in Oracle SQL Developer

Objective:

To create SQL tables and load data in three different pluggable databases (PDBs) - PDB1_CDBLAB, PDB2_CDBLAB, and PDB3_CDBLAB - based on a standard business use case.

Use Case:

We will create a simple business use case for a retail company managing customer orders. The tables will be:

1. **Customers** in PDB1_CDBLAB
2. **Products** in PDB2_CDBLAB
3. **Orders** in PDB3_CDBLAB

Steps:

Step 1: Connect to the PDBs in Oracle SQL Developer (Should be done already)

1. **Open Oracle SQL Developer.**
2. **Connect to PDB1_CDBLAB, PDB2_CDBLAB, and PDB3_CDBLAB:**
   - In the Connections pane, right-click on `PDB1_CDBLAB` and select `Connect`.
   - Repeat the process for `PDB2_CDBLAB` and `PDB3_CDBLAB`.

Step 2: Create the `Customers` Table in PDB1_CDBLAB

1. **In the SQL Worksheet for `PDB1_CDBLAB`, execute the following SQL statement:**

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR2(50),
    LastName VARCHAR2(50),
    Email VARCHAR2(100),
    PhoneNumber VARCHAR2(15)
);
```

2. **Load sample data into the `Customers` table:**

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber) VALUES (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890');
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber) VALUES (2, 'Jane', 'Smith', 'jane.smith@example.com', '098-765-4321');
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber) VALUES (3, 'Alice', 'Johnson', 'alice.johnson@example.com', '555-123-4567');
```

3. **Commit the changes:**

```
COMMIT;
```

Step 3: Create the `Products` Table in PDB2_CDBLAB

1. **In the SQL Worksheet for `PDB2_CDBLAB`, execute the following SQL statement:**

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR2(100),
```

```
    Price DECIMAL(10, 2),
    StockQuantity INT
);
```

2. **Load sample data into the `Products` table:**

```
INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (1,
'Laptop', 999.99, 50);
INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (2,
'Smartphone', 699.99, 200);
INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (3,
'Tablet', 299.99, 150);
```

3. **Commit the changes:**

```
COMMIT;
```

Step 4: Create the `Orders` Table in PDB3_CDBLAB

1. **In the SQL Worksheet for `PDB3_CDBLAB`, execute the following SQL statement:**

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    OrderDate DATE,
    Quantity INT
);
```

2. **Load sample data into the `Orders` table:**

```
INSERT INTO Orders (OrderID, CustomerID, ProductID, OrderDate, Quantity) VALUES (1, 1,
1, TO_DATE('2023-07-15', 'YYYY-MM-DD'), 1);
INSERT INTO Orders (OrderID, CustomerID, ProductID, OrderDate, Quantity) VALUES (2, 2,
2, TO_DATE('2023-07-16', 'YYYY-MM-DD'), 2);
INSERT INTO Orders (OrderID, CustomerID, ProductID, OrderDate, Quantity) VALUES (3, 3,
3, TO_DATE('2023-07-17', 'YYYY-MM-DD'), 3);
```

3. **Commit the changes:**

```
COMMIT;
```

Step 5: Verify the Data

1. **Query the `Customers` table in PDB1_CDBLAB:**

```
SELECT * FROM Customers;
```

2. **Query the `Products` table in PDB2_CDBLAB:**

```
SELECT * FROM Products;
```

3. **Query the `Orders` table in PDB3_CDBLAB:**

```
SELECT * FROM Orders;
```

Conclusion:

This lab provided step-by-step instructions to create tables in three different PDBs and load them with sample data. The `Customers` table in PDB1_CDBLAB, the `Products` table in PDB2_CDBLAB, and the `Orders` table in PDB3_CDBLAB represent a standard business use case for managing customer orders in a retail environment.