**Case Study: E-commerce Order Management System using Oracle SQL and PL/SQL**

**Problem Statement:**

Design and implement an E-commerce Order Management System using Oracle SQL and PL/SQL. The system will be used to manage customers, products, and orders. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle order creation, order updates, and order cancellations.

**Requirements:**

1. **Customer Management**:
   * Implement the functionality to add, update, delete, and search for customers.
   * Ensure that each customer has attributes such as CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, and ADDRESS.
2. **Product Management**:
   * Implement the functionality to add, update, delete, and search for products.
   * Ensure that each product has attributes such as PRODUCT\_ID, PRODUCT\_NAME, CATEGORY, PRICE, and STOCK\_QUANTITY.
3. **Order Management**:
   * Implement the functionality to create, update, and cancel orders.
   * Track orders with attributes such as ORDER\_ID, CUSTOMER\_ID, ORDER\_DATE, TOTAL\_AMOUNT, and ORDER\_STATUS.

**Tasks:**

1. **Design the Database Schema**:
   * Create the Customers, Products, and Orders tables with the appropriate fields and constraints.
   * Define primary keys and foreign keys to maintain data integrity.
2. **Populate the Database with Sample Data**:
   * Insert sample records into the Customers, Products, and Orders tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to create an order. The procedure should check product availability, update stock quantity, and insert a record into the Orders table.
   * Create a procedure to update an order. The procedure should allow modifying the order details, including the products and quantities ordered.
   * Create a procedure to cancel an order. The procedure should update the order status and restore the stock quantities of the products in the order.

**Expected Outcomes:**

1. **Customers Table**:
   * Contains all information about the customers registered in the e-commerce system.
2. **Products Table**:
   * Contains details of all products available for sale.
3. **Orders Table**:
   * Tracks the orders placed by customers, including order dates, total amounts, and statuses.
4. **PL/SQL Procedures**:
   * Efficiently manage the creation, updating, and cancellation of orders, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the Customers, Products, and Orders tables.
2. SQL scripts to insert sample data into the tables.
3. PL/SQL scripts for the procedures to create, update, and cancel orders.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **Customers Table**:
   * **CUSTOMER\_ID**: Number, Primary Key
   * **FIRST\_NAME**: Varchar2(50)
   * **LAST\_NAME**: Varchar2(50)
   * **EMAIL**: Varchar2(100)
   * **PHONE\_NUMBER**: Varchar2(15)
   * **ADDRESS**: Varchar2(200)
2. **Products Table**:
   * **PRODUCT\_ID**: Number, Primary Key
   * **PRODUCT\_NAME**: Varchar2(100)
   * **CATEGORY**: Varchar2(50)
   * **PRICE**: Number
   * **STOCK\_QUANTITY**: Number
3. **Orders Table**:
   * **ORDER\_ID**: Number, Primary Key
   * **CUSTOMER\_ID**: Number, Foreign Key References Customers(CUSTOMER\_ID)
   * **ORDER\_DATE**: Date
   * **TOTAL\_AMOUNT**: Number
   * **ORDER\_STATUS**: Varchar2(20)

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the tables.
* **Develop**: Write PL/SQL procedures for creating, updating, and canceling orders.
* **Test**: Test the procedures with various scenarios (e.g., creating an order, updating an order, canceling an order, ensuring proper stock updates).