Lab Problem 1: Trigger for Automatic Status Update in Orders Table

Problem Statement:

You are managing an e-commerce system that tracks orders. There is a table called `Orders` with the following structure:

```sql
CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_status VARCHAR(20),  -- status can be 'Pending', 'Shipped', 'Delivered'
    order_date DATE,
    delivery_date DATE
);
```

Write a trigger that automatically changes the `order_status` from 'Pending' to 'Delivered' when the `delivery_date` is updated.

Requirements:

1. The trigger should be fired AFTER UPDATE on the `Orders` table.

2. If the `delivery_date` is updated to a non-NULL value, the `order_status` should be updated to 'Delivered'.

3. Test the trigger by updating a sample order's `delivery_date`.

Expected Output:

- After the `delivery_date` is set, the `order_status` of that order should be automatically updated to 'Delivered'.

Lab Problem 2: Trigger for Stock Management

Problem Statement:

You manage a retail inventory system with two tables: `Products` and `Sales`. Each time a sale is made, the system should automatically decrease the stock of the sold product.

1. Products Table:

```sql
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(100),
    stock_quantity INT
);
```

2. Sales Table:

```sql
CREATE TABLE Sales (
    sale_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT,
    quantity_sold INT,
    sale_date DATE
);
```

Write a trigger that automatically reduces the stock of a product in the `Products` table after a sale is recorded in the `Sales` table.

Requirements:

1. The trigger should be fired AFTER INSERT on the `Sales` table.

2. The trigger should subtract the `quantity_sold` from the corresponding product's `stock_quantity` in the `Products` table.

3. Ensure that the stock is updated only if the product has sufficient quantity (optional challenge).

Expected Output:

- After inserting a new sale, the `stock_quantity` in the `Products` table should decrease accordingly.

Lab Problem 3: Trigger for Logging Changes to Employee Salaries

Problem Statement:

You are managing an employee database. Any changes to an employee's salary should be logged for audit purposes.

1. Employees Table:

```sql
CREATE TABLE Employees (
    employee_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    salary DECIMAL(10, 2)
);
```

2. Salary_Audit Table:

```sql
CREATE TABLE Salary_Audit (
    audit_id INT PRIMARY KEY AUTO_INCREMENT,
    employee_id INT,
    old_salary DECIMAL(10, 2),
    new_salary DECIMAL(10, 2),
    change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Write a trigger that logs any changes to an employee’s salary in the `Salary_Audit` table.

Requirements:

1. The trigger should be fired AFTER UPDATE on the `Employees` table.

2. The trigger should log the `old_salary`, `new_salary`, and the `employee_id` of the employee whose salary was changed.

3. Test the trigger by updating the salary of an employee and verifying that the audit log is populated correctly.

Expected Output:

- After updating an employee's salary, the `Salary_Audit` table should contain an entry recording the `old_salary`, `new_salary`, and `employee_id`.
