**Case Study: Inventory Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for an Inventory Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage the inventory of products.

**Requirements:**

1. **Product Management**:
   * Implement the functionality to add, update, and delete products.
   * Each product should have attributes such as product\_id, name, category, price, and quantity\_in\_stock.
2. **Inventory Operations**:
   * Implement the functionality to check the stock level of a product.
   * Implement the functionality to update the stock level after sales or restocking.
3. **Reporting**:
   * Implement the functionality to generate a report of products that are low in stock (quantity\_in\_stock < 10).

**Business Functionalities:**

1. **Add/Update/Delete Products**:
   * Create a class Product with attributes product\_id, name, category, price, and quantity\_in\_stock.
   * Implement methods to add a new product, update existing product details, and delete a product from the inventory.
2. **Stock Level Check and Update**:
   * Implement a method to check the stock level of a product by product\_id.
   * Implement a method to update the stock level when a product is sold or restocked.
3. **Low Stock Report**:
   * Implement a method to generate a list of products that are low in stock (quantity\_in\_stock < 10).

**Section 2: MySQL Database Management**

Design a MySQL database schema to support the Inventory Management System and provide solutions for the given problem statements for querying the database.

**Table Structures:**

1. **Products Table**:
   * product\_id: INT, Primary Key
   * name: VARCHAR(100)
   * category: VARCHAR(50)
   * price: DECIMAL(10, 2)
   * quantity\_in\_stock: INT
2. **Sales Table**:
   * sale\_id: INT, Primary Key
   * product\_id: INT, Foreign Key References Products(product\_id)
   * quantity\_sold: INT
   * sale\_date: DATE
3. **Restocks Table**:
   * restock\_id: INT, Primary Key
   * product\_id: INT, Foreign Key References Products(product\_id)
   * quantity\_restocked: INT
   * restock\_date: DATE

**Problem Statements:**

1. Write a query to find the total quantity of each product sold.
2. Write a query to find the product name and total quantity sold for each product.
3. Write a query to find the names of products that have never been sold.
4. Write a query to find the products that have been restocked more than 5 times.
5. Write a query to find the product names and their current stock levels for products that have sold more than 20 units.