**Case Study: Event Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for an Event Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage events, participants, and registrations.

**Requirements:**

1. **Event Management:**

* Implement the functionality to add, update, and delete event records.
* Each event should have attributes such as event\_id, event\_name, event\_date, location, and max\_participants.

1. **Participant Management:**

* Implement the functionality to manage participants.
* Each participant should have attributes such as participant\_id, name, email, and phone\_number.

1. **Registration Management:**

* Implement the functionality to handle event registrations.
* Each registration should have attributes such as registration\_id, event\_id, participant\_id, and registration\_date.

**Business Functionalities:**

1. **Manage Events:**
   * Create a class Event with attributes event\_id, event\_name, event\_date, location, and max\_participants.
   * Implement methods to add a new event, update event details, and delete an event from the system.
2. **Manage Participants:**
   * Create a class Participant with attributes participant\_id, name, email, and phone\_number.
   * Implement methods to add a new participant, update participant details, and delete a participant.
3. **Manage Registrations:**
   * Create a class Registration with attributes registration\_id, event\_id, participant\_id, and registration\_date.
   * Implement methods to add a new registration, update registration details, and cancel a registration.

**Section 2: MySQL Database Management**

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

**Table Structures:**

1. **Events Table:**

* event\_id: INT, Primary Key
* event\_name: VARCHAR(255)
* event\_date: DATE
* location: VARCHAR(255)
* max\_participants: INT

1. **Participants Table:**

* participant\_id: INT, Primary Key
* name: VARCHAR(255)
* email: VARCHAR(255)
* phone\_number: VARCHAR(20)

1. **Registrations Table:**

* registration\_id: INT, Primary Key
* event\_id: INT, Foreign Key References Events(event\_id)
* participant\_id: INT, Foreign Key References Participants(participant\_id)
* registration\_date: DATE

**Problem Statements:**

* Write a query to find the total number of participants registered for each event.
* Write a query to find the details of events happening in the next month.
* Write a query to find the participants who have registered for more than one event.
* Write a query to find the events that have reached their maximum participant capacity.
* Write a query to find the registration details for a specific participant.