**Case Study: University Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for a University Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage students, courses, and enrollments.

**Requirements:**

1. **Student Management**:
   * Implement the functionality to add, update, and delete student records.
   * Each student should have attributes such as student\_id, name, age, email, and major.
2. **Course Management**:
   * Implement the functionality to manage course information.
   * Each course should have attributes such as course\_id, course\_name, instructor, credits, and max\_students.
3. **Enrollment Management**:
   * Implement the functionality to handle student enrollments in courses.
   * Each enrollment should have attributes such as enrollment\_id, student\_id, course\_id, and enrollment\_date.

**Business Functionalities:**

1. **Add/Update/Delete Students**:
   * Create a class Student with attributes student\_id, name, age, email, and major.
   * Implement methods to add a new student, update existing student details, and delete a student from the system.
2. **Manage Courses**:
   * Create a class Course with attributes course\_id, course\_name, instructor, credits, and max\_students.
   * Implement methods to add a new course, update course details, and delete a course.
3. **Manage Enrollments**:
   * Create a class Enrollment with attributes enrollment\_id, student\_id, course\_id, and enrollment\_date.
   * Implement methods to add a new enrollment, update enrollment details, and cancel an enrollment.

**Section 2: MySQL Database Management**

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

**Table Structures:**

1. **Students Table**:
   * student\_id: INT, Primary Key
   * name: VARCHAR(100)
   * age: INT
   * email: VARCHAR(100)
   * major: VARCHAR(50)
2. **Courses Table**:
   * course\_id: INT, Primary Key
   * course\_name: VARCHAR(100)
   * instructor: VARCHAR(100)
   * credits: INT
   * max\_students: INT
3. **Enrollments Table**:
   * enrollment\_id: INT, Primary Key
   * student\_id: INT, Foreign Key References Students(student\_id)
   * course\_id: INT, Foreign Key References Courses(course\_id)
   * enrollment\_date: DATE

**Problem Statements:**

1. Write a query to find the total number of students enrolled in each course.
2. Write a query to find the names and emails of students who are enrolled in more than three courses.
3. Write a query to find the courses that have not reached their maximum student capacity.
4. Write a query to find the students who are enrolled in courses taught by a specific instructor.
5. Write a query to find the details of enrollments made in the last semester.