Lab Activity

A prestigious university is revamping its student examination management system to modernize the process of conducting and managing examinations for a diverse student body. The new system is expected to handle all aspects of student assessments, including various types of examinations and evaluations. Here are the details,

1. **Exam Types-** the University conducts different types of exams, such as regular written exams, practical exams, oral exams, and project-based assessments. Each exam type has specific requirements and evaluation criteria.
2. **Course Offerings-** the University offers a wide range of courses across various disciplines. Each course has its own set of exams. Professors and instructors are responsible for defining the exams for their respective courses.
3. **Student Registrations-** Students register for courses and, consequently, the associated exams. The system should facilitate student registration and maintain a record of enrolled courses and their associated exams.
4. **Exam Scheduling-** The system needs to schedule exams, ensuring that there are no overlaps in exam timings for students who are registered for multiple courses. Professors and administrative staff are responsible for setting exam dates, times, and locations.
5. **Exam Administration-** On the day of the exam, professors, teaching assistants, and invigilators play a role in administering the exams, ensuring adherence to university rules and regulations. The system should provide a means to record attendance and monitor the exam process.
6. **Grading and Evaluation-** After the exams are completed, professors and teaching assistants evaluate student responses. Some exams may involve multiple-choice questions, while others require subjective assessments. The system should facilitate grading and provide tools for professors to input scores and comments.
7. **Results Publication-** The University needs to publish exam results securely. Students should be able to access their results, and professors need access to individual and aggregate student performance data for further analysis.
8. **Exception Handling-** The system must handle exceptional cases such as make-up exams for students who missed an exam due to illness or other valid reasons.
9. **Data Security-** Protecting the integrity and security of exam data is crucial. The system should have robust security measures to prevent unauthorized access or tampering with exam records.
10. **Reporting and Analytics-** The system should provide reporting and analytics features for the university administration to gain insights into student performance and course effectiveness.

To design the logical model, ER model, and generate DDL for this scenario, you will need to define entities, attributes, relationships, and business rules for each aspect of the scenario. The resulting database should efficiently store and manage information related to courses, exams, student registrations, exam scheduling, exam administration, grading, and exam results. Additionally, it should allow for scalability to accommodate the university's evolving examination needs.

Entities and Attributes	
Courses • Course_ID (Primary Key) • Course_Name • Department • Instructor_ID (Foreign Key)	Registrations • Registration_ID (Primary Key) • Student_ID (Foreign Key) • Course_ID (Foreign Key) • Registration_Date
Exams • Exam_ID (Primary Key) • Course_ID (Foreign Key) • Exam_Type • Exam_Date • Start_Time • End_Time • Location	Exam_Enrollments • Enrollment_ID (Primary Key) • Student_ID (Foreign Key) • Exam_ID (Foreign Key) • Attendance
Students • Student_ID (Primary Key) • First Name • Last Name • Email • Registration_Date	Make-Up Exams • Make_Up_ID (Primary Key) • Student_ID (Foreign Key) • Exam_ID (Foreign Key) • Make_Up_Date • Reason

LAB TASK

A university campus is looking to modernize its hostel mess management system to efficiently cater to the dining needs of its students and staff. The existing system is manual and poses challenges in managing meal plans, dietary preferences, and billing. The university is keen on implementing a robust database system to streamline hostel mess operations. Here are the details of the scenario:

1. **Hostel Residents**
   The university has multiple hostels, each housing a different set of students and staff. Hostel residents are expected to register for a meal plan. The system should maintain a database of residents with their details, including hostel ID, room number, and contact information.

2. **Meal Plans**
   The system offers various meal plans to residents, allowing them to choose from options like full board, half board, or specific meal packages. Meal plans may also include dietary preferences or restrictions, such as vegetarian, vegan, or allergen-free meals.

3. **Mess Operations**
   The hostel mess serves meals at specific timings and in designated dining areas. The system should manage mess operations, including meal scheduling, menu planning, and food inventory.

4. **Billing and Payments**
   The system should generate bills for meal plans and allow residents to make payments. It should keep a record of billing cycles, payment history, and overdue payments.

5. **Meal Booking**
   Residents should be able to book their meals in advance. They can specify the meal plan, dietary preferences, and any guest meals required. The system should provide a way to confirm or modify bookings.

6. **Dietary Preferences and Allergies**
   The database should store dietary preferences, restrictions, and allergies of residents to ensure that meals are prepared according to individual needs.

7. **Guest Meals**
   Occasionally, residents may have guests or visitors who need meal accommodations. The system should handle guest meal bookings and charges accordingly.

8. **Feedback and Complaints**
   Residents should have a channel to provide feedback on the quality of meals and report any issues or complaints. The system should track and address these concerns.

9. **Inventory Management**
   The hostel mess needs to manage food inventory efficiently. It should monitor the availability of ingredients, place orders for supplies, and ensure minimal food wastage.

10. **Reports and Analytics**
    The system should offer reporting and analytics features for administrators to assess meal plan popularity, billing trends, and feedback analysis.

To design the logical model, ER model, and generate DDL for this scenario, you will need to define entities, attributes, relationships, and business rules for each aspect of the scenario. The resulting database should efficiently store and manage information related to residents, meal plans, billing, meal bookings, dietary preferences, and feedback. Additionally, it should allow for scalability to accommodate the university's changing dining needs.