**Scenario 1 (University Sport Complex System)**

The university's Sports Complex currently relies on manual logbooks and spreadsheets to manage facilities, equipment, coaches, and bookings. This outdated approach often leads to problems such as double bookings, missing records, and difficulty tracking equipment maintenance.

You have been appointed as a Database Designer to develop a conceptual database design for the said university sports complex.

The proposed system should:

- Maintain a clear catalogue of facilities and equipment.
- Allow members (students, staff, and external visitors) to register and make bookings.
- Enable coaches to be assigned to training sessions.
- Prevent booking conflicts.
- Track maintenance schedules for both equipment and facilities.

**Scenario 2 (Campus Carpool Management System)**

A large university aims to ease traffic congestion and encourage sustainable transportation by implementing a Campus Carpool Management System. The platform will allow students and staff to offer or request rides within the campus or to nearby locations.

At present, carpool arrangements are made informally via social media and group chats, often resulting in confusion, overlapping bookings, and limited accountability.

You have been assigned to design a conceptual database for this new system to ensure efficient ride matching, clear scheduling, and reliable tracking of participants.

What the system shall do:

- Users register with their university ID, name, role (student/staff), and contact information.
- Drivers can create ride offers, specifying date, time, route, number of seats, and pick-up points.
- Passengers can search and book available rides.

- The system prevents double-booking of seats.
- Cancellations are logged, and users with repeated cancellations may be flagged.
- Feedback and ratings can be left for drivers and passengers after each ride.
- Monthly reports can be generated to show the most active drivers, most popular routes, and seat utilization rates.

## Task 1 (Relational Database Theory – Requirement analysis) – 10%

1. Requirement Analysis
    - **a)** Identify at least 8 functional and 4 non-functional requirements.
    - **b)** Clearly state any assumptions made during the design process.

## Task 2 (Relational Database Theory – Conceptual design) – 20%

1. Entity & Attribute Identification **(10%)**
    - **a)** Identify at least 6 or more entities from the scenario chosen.
    - **b)** List attributes for each entity, including suitable data types.
    - **c)** Specify the Primary Key (PK) for each entity.

2. Entity–Relationship Diagram **(10%)**
    - **a)** Create an enhanced ERD (EERD) using Chen's model.
    - **b)** Indicate all relationships, cardinalities, and participation constraints.

## Task 3 (Relational Database Theory – Logical design) – 10%

1. Relational Schema Conversion (10%)
    - **a)** Convert the EERD above into a relational schema.
    - **b)** Define all Primary Keys (PK) and Foreign Keys (FK).

## Task 4 (Relational Database Theory – Physical design) – 9%

1. Justification

**If you choose scenario 1, answer the following:**

**a)** Explain how your design meets university sport complex system requirements and prevents double bookings.

**If you choose scenario 2, answer the following:**

**a)** Explain how your design supports the campus's sustainability goals, ensures seat allocation accuracy, and improves accountability for ride-sharing.

## Task 5 (Access control) – 12%

**a)** Propose at least three (3) user roles for the system and explain the purpose or responsibilities of each role.

**b)** Clearly state the type of access (e.g., read, insert, update, delete) each user role has for the respective database table.

## Task 6 (Database Integrity) – 12%

Explain how your database enforces the following types of integrity:

- Entity integrity
- Referential integrity
- Domain integrity

## Task 7 (Database transaction) – 12%

Discuss how your system design maintains each of the ACID properties such as atomicity, consistency, isolation, and durability during transaction processing.