# LabWork1 - Final Solutions

Part 1: Key Identification

Task 1.1 (Employee): - Example superkeys: {EmpID}, {SSN}, {Email}, {Phone}, {EmpID, Name}, {SSN, Phone} - Candidate keys: {EmpID}, {SSN}, {Email}, {Phone} - Primary key choice: EmpID (stable, unique identifier) - Can two employees share a phone number? Based on the sample data, no, phone numbers are unique. Task 1.1 (Registration): - Primary key = {StudentID, CourseCode, Section, Semester, Year} - Each attribute is necessary to uniquely identify a registration record. - Candidate keys: same as above. Task 1.2 (Foreign Keys): - Student(AdvisorID) → Professor(ProfID) - Course(DepartmentCode) → Department(DeptCode) - Department(ChairID) → Professor(ProfID) - Enrollment(StudentID) → Student(StudentID) - Enrollment(CourseID) → Course(CourseID)

Part 2: ER Diagrams

Task 2.1 (Hospital Management System): - Entities: Patient (strong), Doctor (strong), Department (strong), Appointment (weak), Prescription (weak), Room (strong) - Attributes: Patient(PatientID, Name, Birthdate, Address, Phones, Insurance), Doctor(DoctorID, Name, Specializations, Phone, Office), Department(DeptCode, DeptName, Location), Appointment(DateTime, Purpose, Notes), Prescription(Medication, Dosage, Instructions), Room(RoomNumber, DeptCode) - Relationships: Patient–Appointment–Doctor (M:N), Doctor–Department (N:1), Room–Department (N:1), Prescription (Doctor–Patient, M:N) Task 2.2 (E-commerce Platform): - Entities: Customer, Order, Product, Category, Vendor, Review, Inventory, OrderItem (weak) - Relationships: Customer–Order (1:N), Order–OrderItem (1:N), Product–Category (N:1), Product–Vendor (N:1), Product–Customer through Review (M:N with attributes) - Weak entity: OrderItem (depends on Order and Product) - Example many-to-many: Product–Customer through Review with attributes (rating, comment)
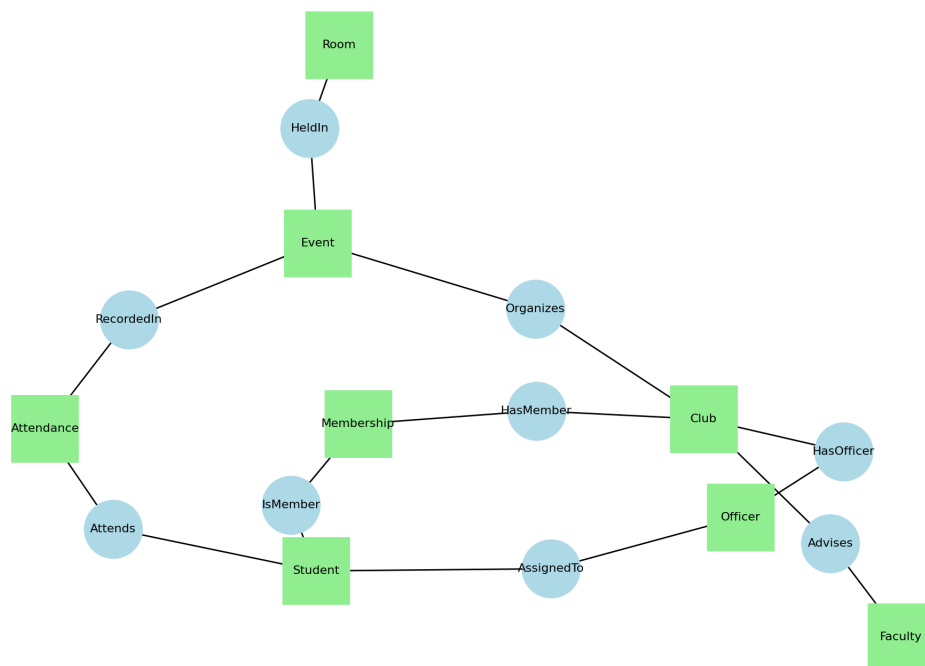
Hospital Management System
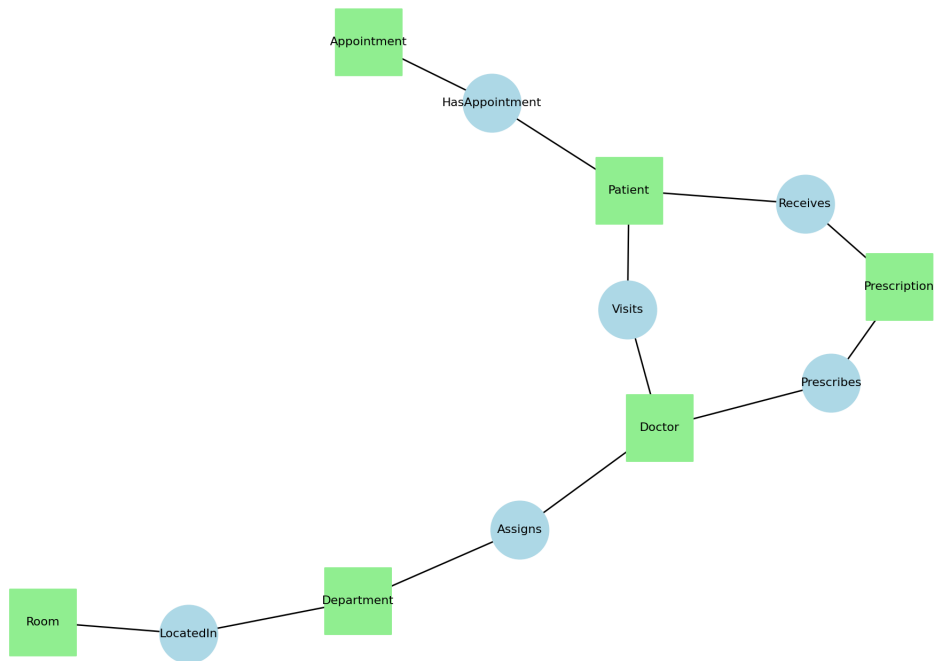
E-commerce Platform
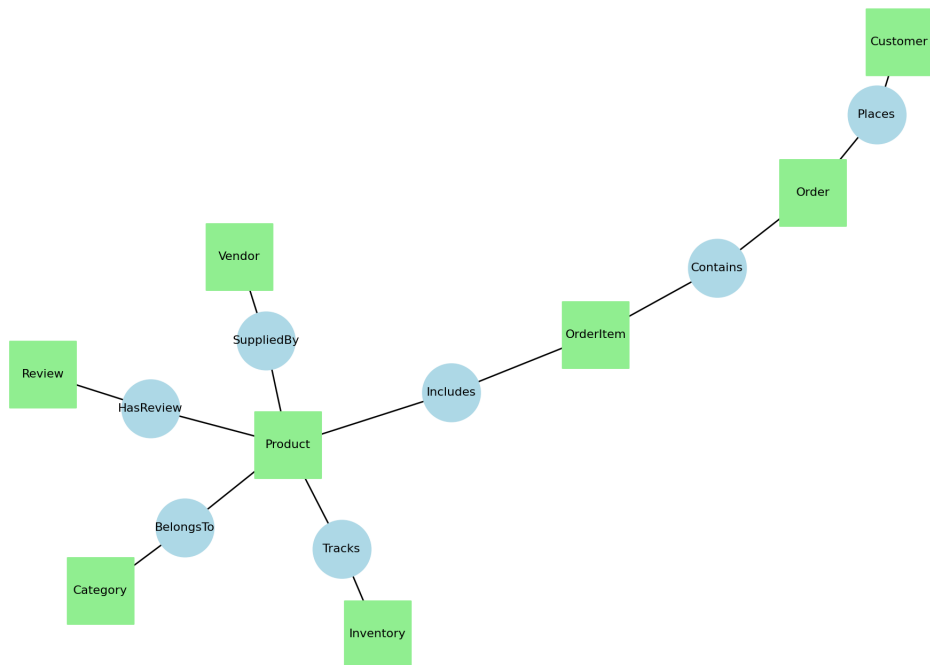
Student Clubs System

Part 4: Normalization

Task 4.1 (StudentProject): - Functional Dependencies: StudentID → StudentName, StudentMajor; ProjectID → ProjectTitle, ProjectType; SupervisorID → SupervisorName, SupervisorDept; {StudentID, ProjectID} → Role, HoursWorked, StartDate, EndDate - Problems: redundancy in Student, Project, and Supervisor info; update, insert, delete anomalies - 1NF: ensure atomic attributes - 2NF: decompose by removing partial dependencies - 3NF: remove transitive dependencies (e.g., SupervisorID → Dept) Task 4.2 (CourseSchedule): - Primary key: {StudentID, CourseID, TimeSlot} - FDs: StudentID → StudentMajor; CourseID → CourseName; InstructorID → InstructorName; Room → Building; {CourseID, TimeSlot} → InstructorID, Room - Not in BCNF; decompose into Student, Course, Instructor, Section, Enrollment

Part 5: Design Challenge

Task 5.1 (Student Clubs System): - Entities: Club, Student, Faculty, Event, Room, Membership, Officer - Relationships: Student–Club (M:N via Membership), Club–Faculty (1:N), Club–Event (1:N), Event–Room (1:1), Event–Attendance (M:N with Student) - Weak entity: Officer (depends on Club) - Example queries: 1. Find all students who are officers in the Computer Science Club 2. List all events scheduled for next week with their room reservations 3. Show total budget and expenses for each club