Task 1.1

Relation A

1. {EmpID} {SNN} {Email} {Phone} {EmpID, SNN} {Email, Phone}
2. EmpID, SNN, Email, Phone
3. I would choose EmpID just because i like it more than other candidate keys
4. No, two employees cannot have the same phone number because if they could have, there wouldn't be any point in using phone numbers

Relation B

1. StudentID, CourseCode, Section, Semester, Year
2. StudentID is used to understand which student is registered, CourseCode needed because same student can take different courses, Section is neccesary because there can be different course sections, Semester is needed because students can take same courses but in the different sections, Year is needed because there can be same semesters but only different years
3. No candidate keys

Task 1.2

Student(**StudentID**, Name, Email, Major, AdvisorID)
Professor(**ProfID**, Name, Department, Salary)
Course(**CourseID**, Title, Credits, DepartmentCode)
Department(**DeptCode**, **DeptName**, Budget, ChairID)
Enrollment(StudentID, CourseID, **Semester**, Grade)

Foreign keys: Department, DepartmentCode, StudentID, CourseID

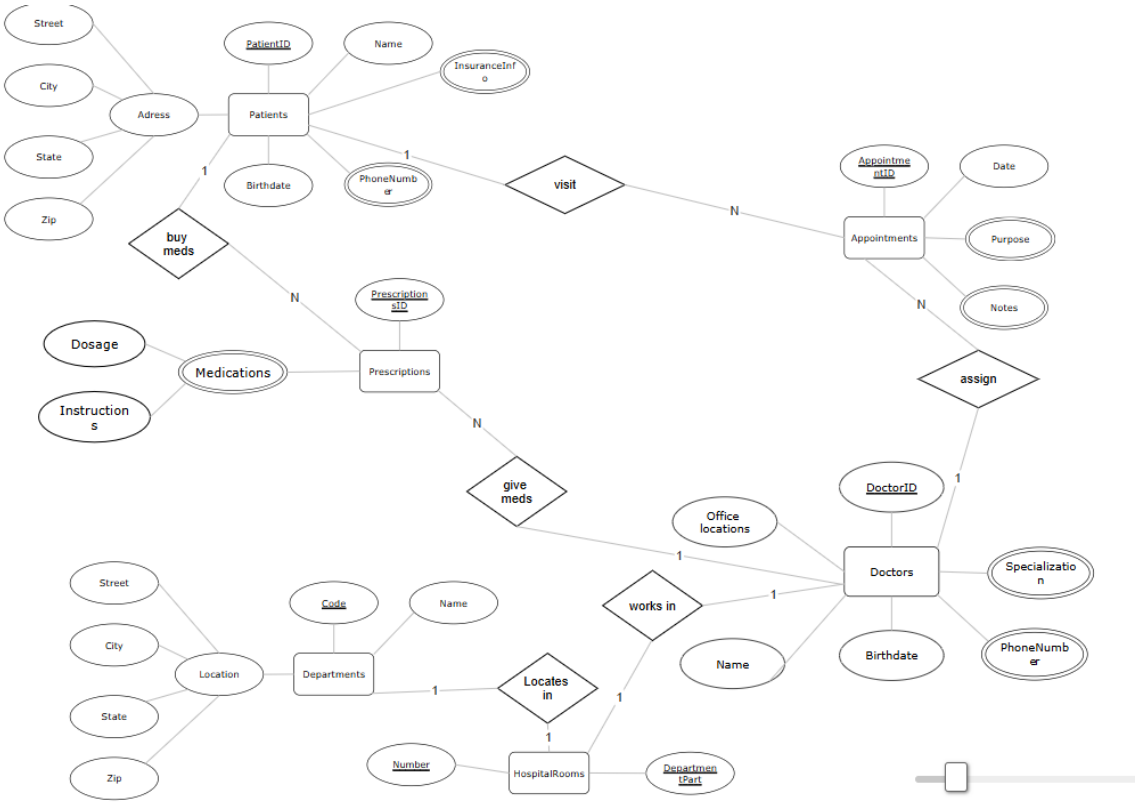
Task 2.1

1. Patients (strong), Doctors (strong), Departments (strong), Appointments (weak), Prescriptions (weak), Hospital Rooms (weak)
2. **Patients** – PatientID (simple), Name (simple), Birthdate (simple), Adress (composite), PhoneNumber (multi-valued), Insurance (multi-valued)
   Doctors – DoctorID (simple), Name (simple), Specialization (multi-valued), PhoneNumber (multi-valued), OfficeLocation (composite)
   Departments – DeptCode (simple), Name (simple), Location (composite)
   Appointments - PatientName, DocName (simple), Date (simple), Time (simple), Purpose (multi-valued), Notes (multi-valued)

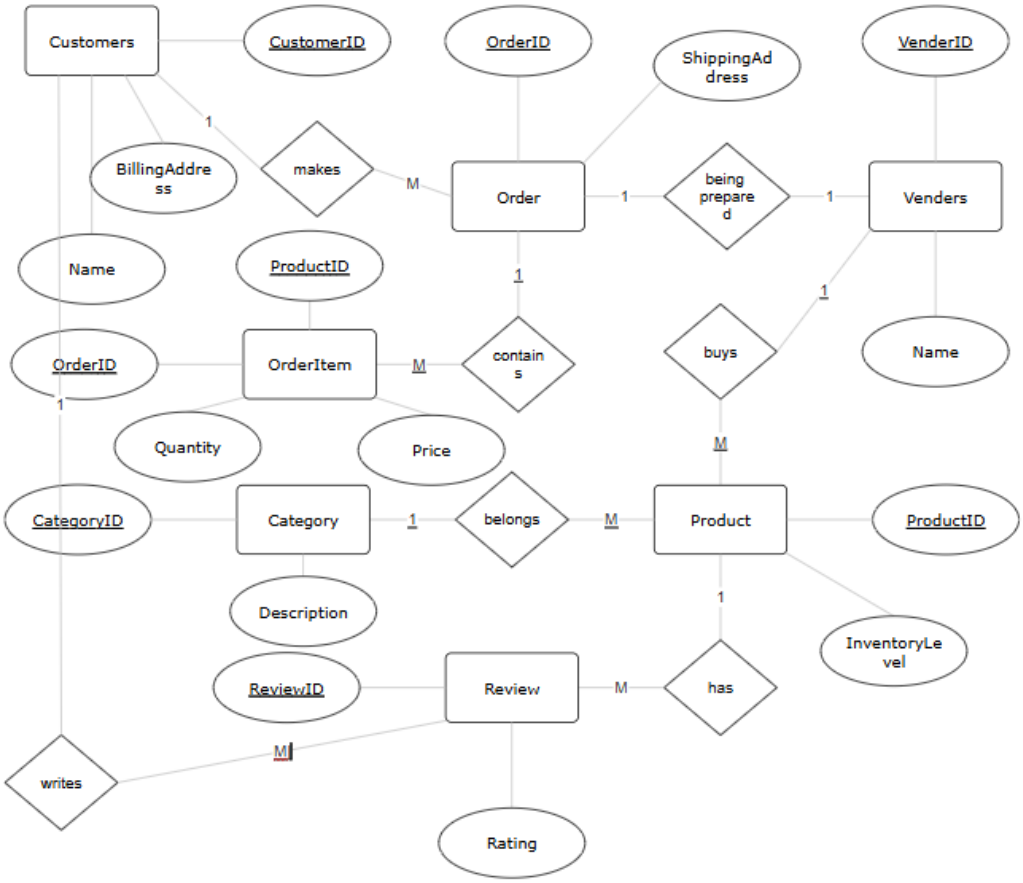
Prescriptions – Medications (multi-valued), Dosage (simple, derived), Instructions (complex)

Hospital Rooms – Room (simple), DeptName (simple)

3. Patients – Doctors (M:N)
   Doctors – Department (M:1)
   Doctors – Appointments (1:N)
   Prescriptions – Patients (1:1)
   Department – Hospital Rooms (1:N)

4. 

Task 2.2



1.
2. OrderItems because it exists only if Order exists
3. Order – Products

Task 4.1

1. StudentID -> StudentName
   StudentID -> StudentMajor
   ProjectID -> ProjectTitle
   ProjectID -> ProjectType
   SupervisorID -> SupervisorName
   SupervisorID -> SupervisorDept
   StudentID + ProjectID -> Role, HoursWorked, StartDate, EndDate

2. Redundancy: StudentName, StudentMajor, ProjectTitle, ProjectType repeating every time
   Update Anomaly: change of Supervisor's name leads to changing it everywhere
   Insert Anomaly: new project cannot be created without students or supervisor
   Delete Anomaly: deleting one student will lead to deleting his project also

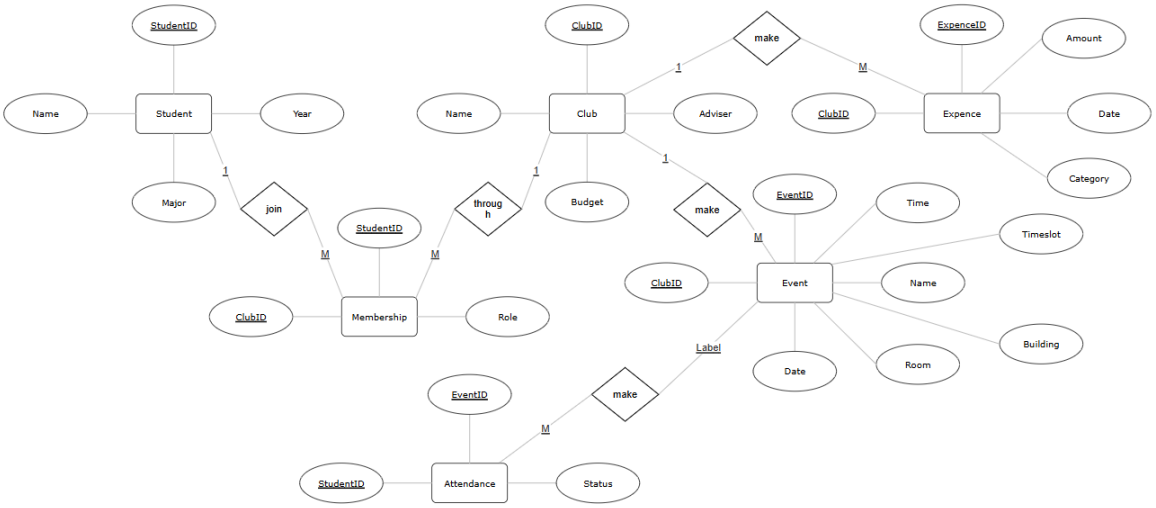
3. 1NF: ProjectID, ProjectTitle and ProjectType should be in Project because there may be several projects made by one student (multi-valued).
4. 2NF: StudentName and StudentMajor depends only on StundentID, same for ProjectID with ProjectName and ProjectTitle, SupervisorID with SupervisorName and SupervisorDept

5. 3NF:
   Student(StudentID, StudentName, StudentMajor)
   Supervisor(SupervisorID, SupervisorName, SupervisorDept)
   Project(ProjectID, ProjectTitle, ProjectType)
   StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

Task 4.2

1. (StudentID, CourseID)
2. StudentID → StudentMajor
   CourseID → CourseName, InstructorID, TimeSlot, Room
   InstructorID → InstructorName
   Room → Building
3. Not in BCNF
4. Student(StudentID, StudentMajor)
   Instructor(InstructorID, InstructorName)
   Room(Room, Building)
   Course(CourseID, CourseName, InstructorID, TimeSlot, Room)
   Enrollment(StudentID, CourseID)
5. no information lost

Task 5.1



1.
2. Student (StudentID PK, Name, Major, Year)

Club (ClubID PK, Name, Budget, Advisor)
Membership (StudentID PK, ClubID PK, Role, FK(StudentID)→Student, FK (ClubID)→Club)
Event (EventID PK, ClubID FK→Club, Name, Date, Time, Room, Building, TimeSlot)
Attendance (EventID PK, StudentID PK, Status, FK(EventID)→Event, FK (StudentID)→Student)
Expense (ExpenseID PK, ClubID FK→Club, Amount, Date, Category)

3. We could have done a special entity for Roles
4. List all students by the role, show all organizated events by the club and calculate overall budget