Lab 1

Task 1.1

Relation A

Superkeys

1) {EmpID}

2){SSN}

3){Email}

4){Phone}

5){EmpID, Name}

6){SSN, Department}

7){Email, Salary, Department}

Candidate keys

1) {EmpID}

2){SSN}

3){Email}

4) {Phone}

Relation B

1)Primary key = (StudentID, CourseCode, Section, Semester, Year)

2)Each attribute is necessary to prevent duplicate registrations and to rfollow the business rules:

StudentID → to identify students

CourseCode → to identify courses

Section → to identify which class offering

Semester → to identify when in the year

Year → to distinguishe terms across years

3)No extra CK except the (StudentID, CourseCode, Section, Semester, Year)

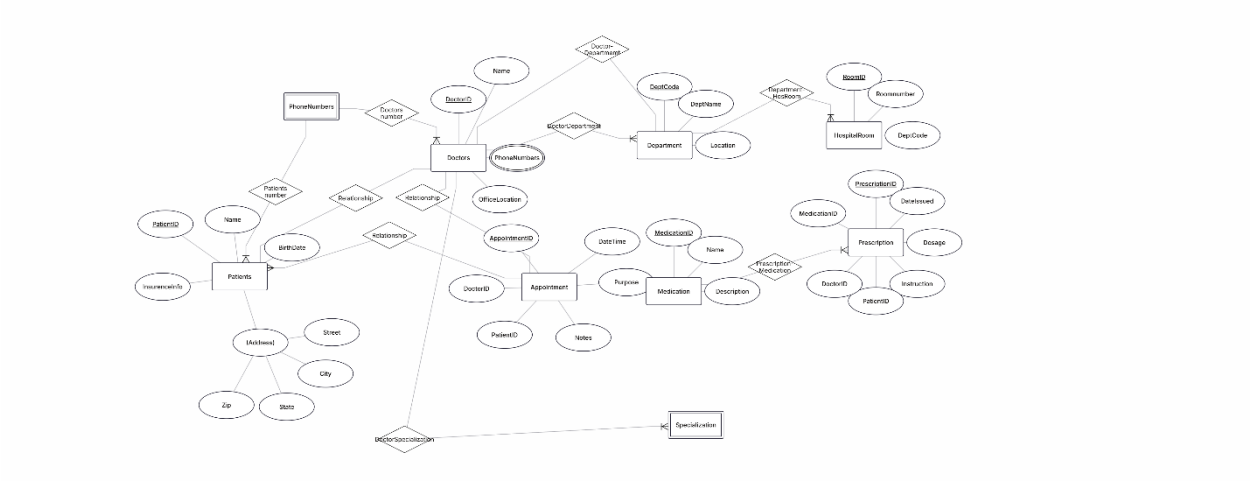
Task 1.2

Foreign Key Relationships:

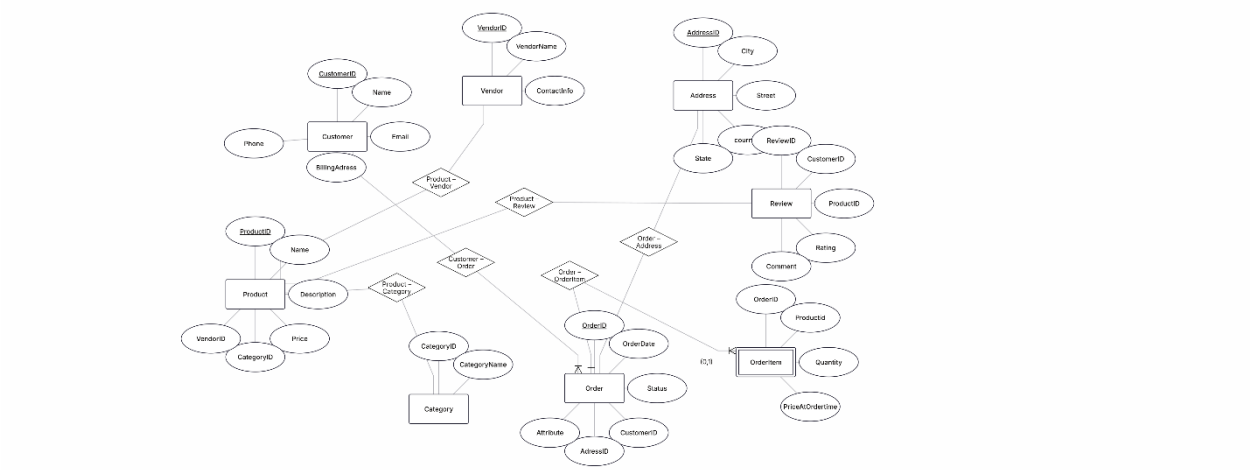
1. Student.Major → Department.DeptCode
2. Student.AdvisorID → Professor.ProfID
3. Professor.Department → Department.DeptCode
4. Course.DepartmentCode → Department.DeptCode

5. Department.ChairID → Professor.ProfID
6. Enrollment.StudentID → Student.StudentID
7. Enrollment.CourseID → Course.CourseID

Task 2.1



2.2



4.1

FDs:

- StudentID → StudentName, StudentMajor
- SupervisorID → SupervisorName, SupervisorDept
- ProjectID → ProjectTitle, ProjectType, SupervisorID
- (StudentID, ProjectID) → Role, HoursWorked, StartDate, EndDate

Problems:

- Redundancy (students, supervisors, projects info repeated).
- Update anomaly, Insert anomaly, Delete anomaly.

1NF: Already satisfied (all atomic values).

2NF: Primary key = (StudentID, ProjectID). Removed partial dependencies by decomposition.

3NF: Removed transitive dependency (Supervisor info separated).

Final Tables:

- Student(StudentID, StudentName, StudentMajor)
- Supervisor(SupervisorID, SupervisorName, SupervisorDept)
- Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)
- StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

4.2

Sure — super short, lab-ready answers:

1. **Primary key:** (StudentID, CourseID, TimeSlot, Room).
2. **Functional dependencies:**
    - StudentID → StudentMajor
    - CourseID → CourseName
    - InstructorID → InstructorName
    - Room → Building
    - (CourseID, TimeSlot, Room) → InstructorID
    - (StudentID, CourseID, TimeSlot, Room) → {StudentMajor, CourseName, InstructorID, InstructorName, Building}
3. **BCNF?** No. Violated because FDs like StudentID → StudentMajor, CourseID → CourseName, Room → Building, etc., have left sides that are not superkeys.
4. **BCNF decomposition (concise):**
    - Student(StudentID PK, StudentMajor)
    - Course(CourseID PK, CourseName)
    - Instructor(InstructorID PK, InstructorName)
    - Room(Room PK, Building)
    - CourseSection(CourseID, TimeSlot, Room, InstructorID) — PK (CourseID,TimeSlot,Room)
    - Enrollment(StudentID, CourseID, TimeSlot, Room) — PK (StudentID,CourseID,TimeSlot,Room) (FK → Student, CourseSection)

Want this even shorter (one-line each) or ready-to-paste into your lab?

5.1