Part 1.

Task 1.1

Relation A:

1. EmpID, SSN, Email, Phone, Name, Department
2. EmpID, SSN, Email, Phone
3. EmpID. Because it is unique for every employee and the same EmpID cannot be given to more than one person. Making it an unique identifier for each employee
4. No. Since phone number can be used only by one person and would not work on two devices simultaneously.

Relation B:

StudentID is a primary key. CourseCode, Section, Semester, Year will be needed attributes. CourseCode, candidate key because it can be used to link tables that offer course information.

Task 1.2

Student(StudentID, Name, Email, Major, AdvisorID) – AdvisorID is a foreign key

Professor(ProfID, Name, Department, Salary) – Department is a foreign key

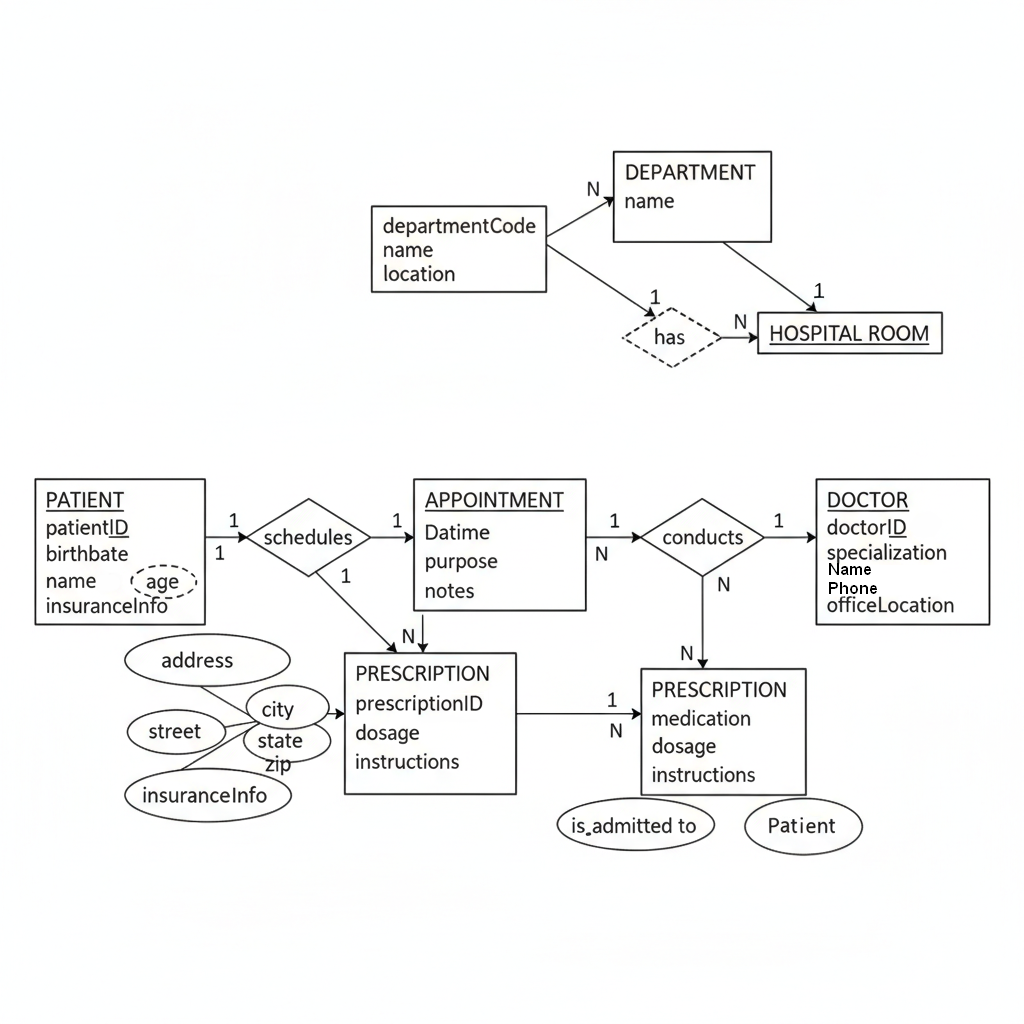
Course(CourseID, Title, Credits, DepartmentCode) – DepartmentCode is a foreign key

Department(DeptCode, DeptName, Budget, ChairID) – ChairID is a foreign key

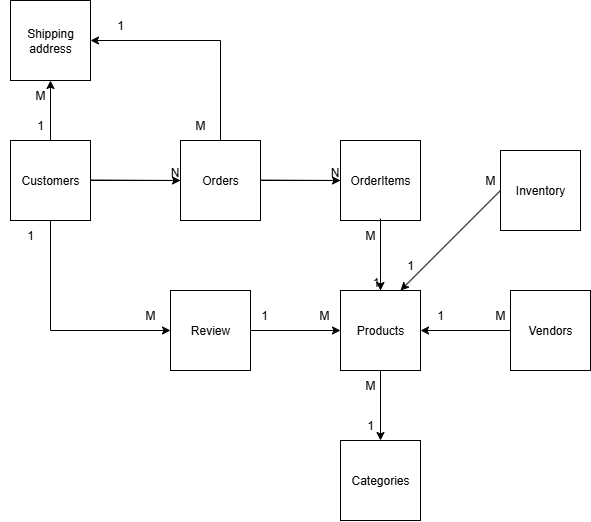
Enrollment(StudentID, CourseID, Semester, Grade) – CourseID is a foreign key

Part 2

Task 2.1



Task 2.2



Part 4

Task 4.1

Dependencies:

StudentID -> StudentName, StudentMajor

ProjectID -> ProjectTitle, ProjectType, SupervisorID

SupervisorID -> SupervisorName, SupervisorDept

(StudentID, ProjectID) -> Role, HoursWorked, StartDate, EndDate

ProjectID -> SupervisorID -> SupervisorName, SupervisorDept

Redundancy:

Repetition of student info (StudentName, Major)

Repetition of project info (Title, Type, Supervisor)

Repetition of supervisor info (Name, Dept)

Anomalies:

Update anomaly: If Supervisor "Dr. Smone" changes department, we must update every row for all his projects.

Insert anomaly: We can’t add a new student or supervisor unless they are linked to a project.

Delete anomaly: If the only student working on Project X leaves, deleting that row also deletes the project and supervisor info.

1NF:

It is already in 1NF; All values are atomic

2NF:

Primary key – StudentID

Composite key – ProjectID

Partial dependencies:

StudentID → StudentName, StudentMajor

ProjectID → ProjectTitle, ProjectType, SupervisorID

SupervisorID → SupervisorName, SupervisorDept

Final 2NF:

Student(StudentID, StudentName, StudentMajor)

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)

Supervisor(SupervisorID, SupervisorName, SupervisorDept)

StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

3NF:

Student(StudentID, StudentName, StudentMajor) - PK: StudentID

Supervisor(SupervisorID, SupervisorName, SupervisorDept) - PK: SupervisorID

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID (FK)) - PK: ProjectID - FK: SupervisorID -> Supervisor

StudentProject(StudentID (FK), ProjectID (FK), Role, HoursWorked, StartDate, EndDate) -PK: (StudentID, ProjectID) - FK: StudentID -> Student - FK: ProjectID -> Project

Task 4.2

1. Primary key - (StudentID, CourseID, TimeSlot, Room)
2. Functional dependencies:

StudentID -> StudentMajor

CourseID -> CourseName

InstructorID -> InstructorName

Room -> Building

(CourseID, TimeSlot, Room) -> InstructorID

1. Table is not in BCNF

Student(StudentID PK, StudentMajor) Separate with PK StudentID

Course(CourseID PK, CourseName) Separate with PK CourseID

Instructor(InstructorID PK, InstructorName) Separate with PK InstructorID

RoomInfo(Room PK, Building) Separate with PK Room and Building

Decompose into:

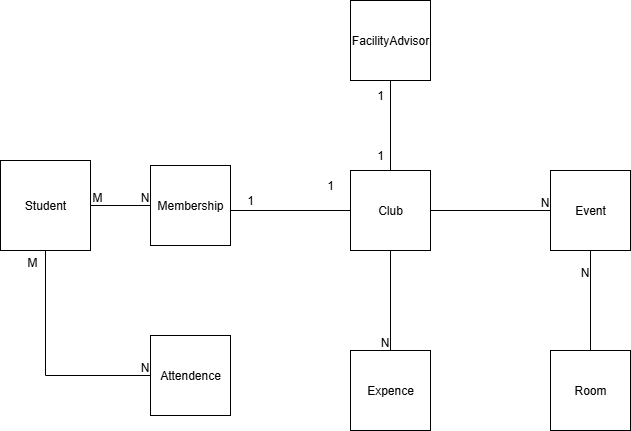
Section(CourseID, TimeSlot, Room, InstructorID) — PK: (CourseID, TimeSlot, Room); FKs: CourseID -> Course, Room -> RoomInfo, InstructorID -> Instructor

Enrollment(StudentID, CourseID, TimeSlot, Room) — PK: (StudentID, CourseID, TimeSlot, Room); FK: Student(StudentID) and Section(CourseID,TimeSlot,Room)

1. Decomposition is lossless because everything intersects with each other.

Part 5

Task 5.1



Normalized Relational Schema (3NF)

Student(StudentID PK, Name, Major, Year)

Club(ClubID PK, ClubName, Description, Budget, FacultyID FK)

FacultyAdvisor(FacultyID PK, FacultyName, Department, Email)

Membership(StudentID FK, ClubID FK, JoinDate, OfficerPosition,

PK = (StudentID, ClubID))

Event(EventID PK, ClubID FK, EventName, EventDate, EventType, RoomID FK)

Attendance(EventID FK, StudentID FK, Status,

PK = (EventID, StudentID))

Room(RoomID PK, Building, Capacity)

Expense(ExpenseID PK, ClubID FK, Amount, ExpenseDate, Category, Description)

3.

Facility Advisor could be several people for one club. Because we often see, how in clubs there are several leaders and primes. In fact, only one person must be a facility advisor.

4.

Find all students who attended the event in the Chess Club.

List all upcoming events for clubs advised by Dr. Dre.

Show the total expenses for each club in the current semester.