

Task 1.1: Superkey and Candidate Key Analysis

A.

1. Superkeys: (EmpID), (SSN), (Email), (EmpID, Name), (SSN, Name), (EmpID, SSN)

2. Candidate keys: (EmpID), (SSN), (Email)

3. EmpID, because SSN could be changed with document update, and email could be changed with domain change. EmpID is the most stable candidate key.

4. They can't have 2 identical personal phone numbers, but they could have got 2 identical corporate phone numbers via VoIP

B.

1. (StudentID, CourseCode, Section, Semester, Year)

2. StudentID → identify student

CourseCode + Section → to exclude all repeating registration on different sections of course

Semester + Year → to exclude all repeating courses in the semester

3. Grade and Credits couldn't be candidate keys

Task 1.2: Foreign Key Design

Student.AdvisorID → Professor.ProfID

Student.Major → Department.DeptCode

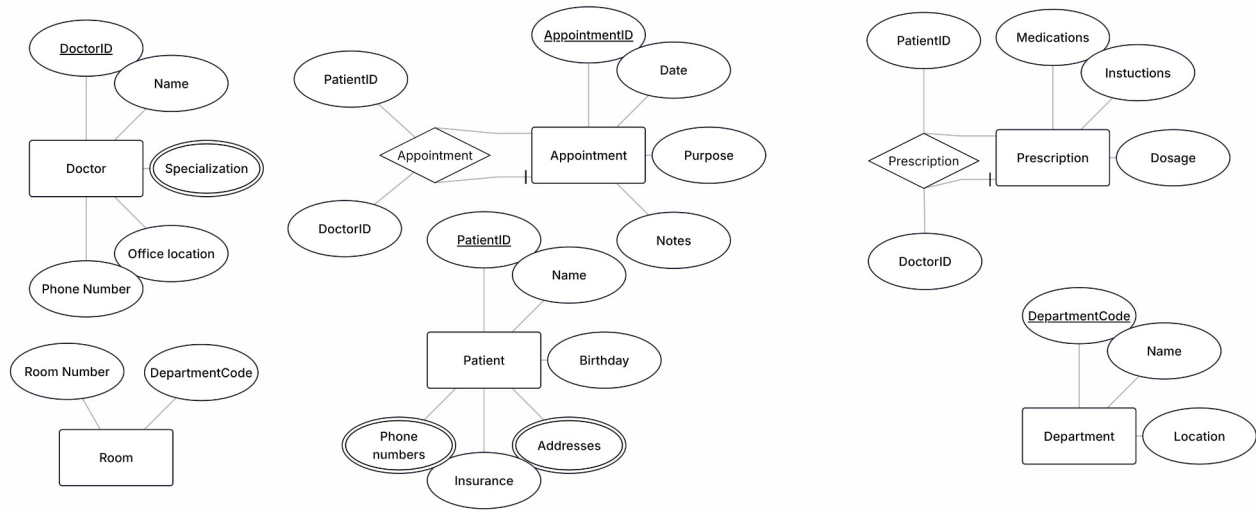
Course.DepartmentCode → Department.DeptCode

Department.ChairID → Professor.ProfID

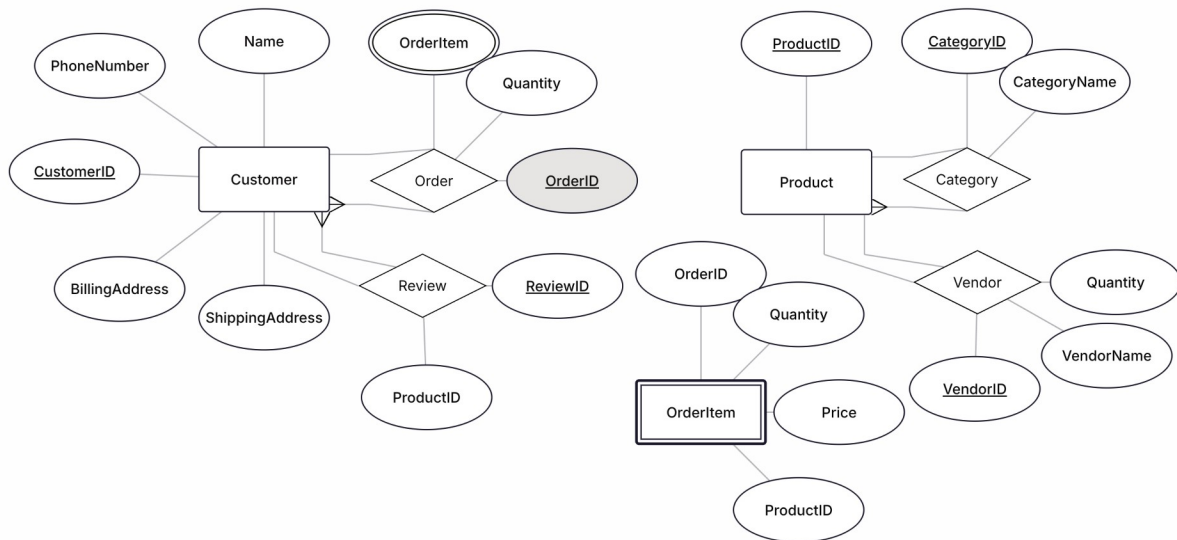
Enrollment.StudentID → Student.StudentID

Enrollment.CourseID → Course.CourseID

Task 2.1



Task 2.2



OrderItem is weak, because it doesn't make sense without Order;
Customer → Review → Products have many-to-many relationship

Task 4.1

1.

StudentID \rightarrow StudentName, StudentMajor

ProjectID \rightarrow ProjectTitle, ProjectType

SupervisorID \rightarrow SupervisorName, SupervisorDept

(StudentID, ProjectID) \rightarrow Role, HoursWorked, StartDate, EndDate

2.

Update anomaly (Change of Supervisor Name need changes in many tables)

Insert anomaly (Impossible to create project without student nor supervisor)

Delete anomaly (Deletion of student deletes all projects)

3. Table already 1NF

4. 2NF

Primary keys: StudentID, ProjectID

Partial dependencies: StudentID \rightarrow StudentName, StudentMajor; ProjectID \rightarrow ProjectTitle, ProjectType

Student(StudentID, StudentName, StudentMajor)

Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)

Supervisor(SupervisorID, SupervisorName, SupervisorDept)

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

5. Transitive dependency: ProjectID \rightarrow SupervisorID \rightarrow SupervisorDept

Task 4.2: Advanced Normalization

1. Primary key: (TimeSlot, Room)

2.

StudentID \rightarrow StudentMajor

CourseID \rightarrow CourseName

InstructorID \rightarrow InstructorName

(TimeSlot, Room) \rightarrow Building

(CourseID, TimeSlot, Room) \rightarrow InstructorID

3. Not in bcnf, many transitive dependency

4.

Student(StudentID, Major)

Course(CourseID, CourseName)

Instructor(InstructorID, Name)

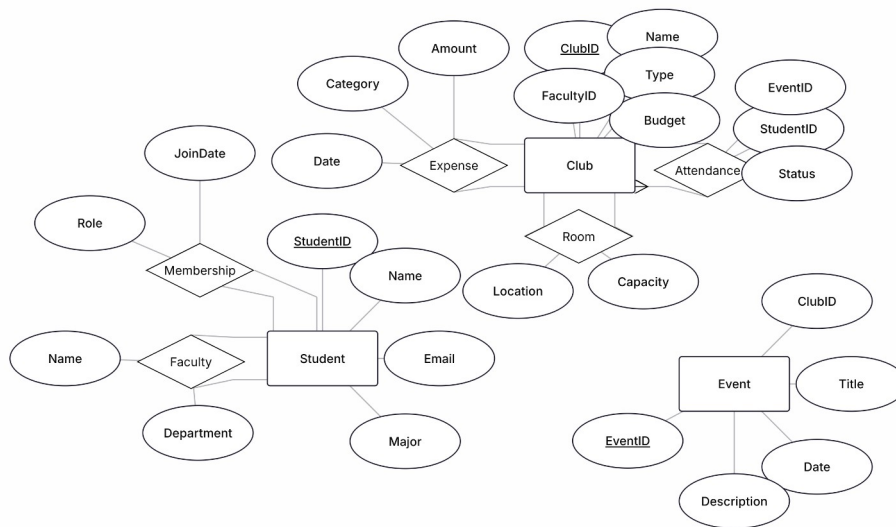
Room(Room, Building)

Section(CourseID, InstructorID, TimeSlot, Room)

Registration(StudentID, CourseID, TimeSlot)

5. All entities share all attributes from initial table, so no data loss.

Task 5.1: Real- World Application



3. Club officer position replaced with just role in student membership, so students can have multiple roles within one club among all clubs.

4.

Find all students who are officers in the Computer Science Club.

List all events scheduled for next week with their room reservations.

Show the total budget and expenses for each student club.