Zharylkassyn Tomiris

Database Systems Laboratory Work

Week 2: Relational Model & Keys

Part 1: Key Identification Exercises

Task 1.1: Superkey and Candidate Key Analysis

- Superkeys: {EmpID}, {SSN}, {Email}, {Phone}, {EmpID, Name}, {Email, Department}
- Candidate keys: {EmpID}, {SSN}, {Email}, {Phone}
- Primary key: **EmpID**, because it is stable and company-controlled.
- Two employees cannot have the same phone number in this table, but in real life they might.

Relation B: Course Registration

- The minimum attributes for the primary key are **(StudentID, CourseCode, Section, Semester, Year)**.
- Each attribute is needed:
- **StudentID** - to know which student.
- **CourseCode** - to know which course.
- **Section** - because a course can have many sections.
- **Semester + Year** - because the same student can take the same course again in another semester.
- Extra candidate key: **(CourseCode, Section, Semester, Year)** can be a key for the course section itself, because it shows one unique offering with fixed credits.

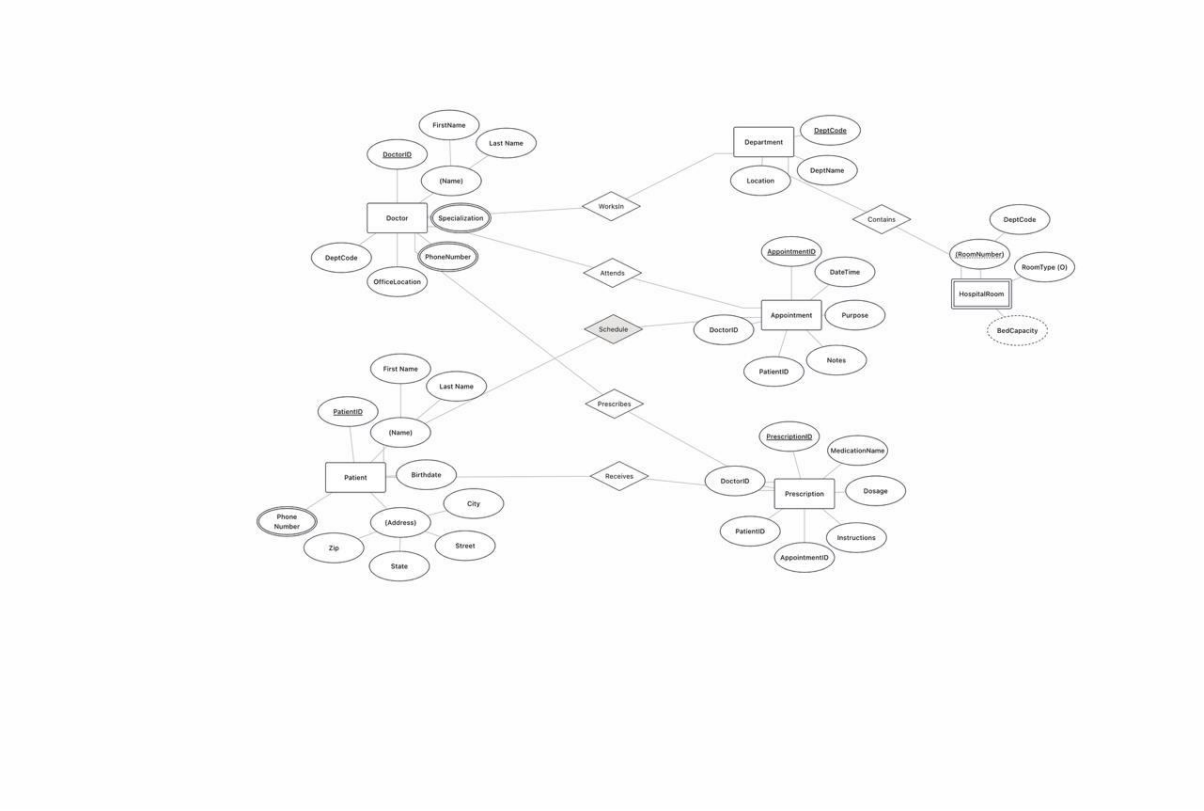
Task 1.2: Foreign Key Design

- **Student - Professor**
- AdvisorID is a foreign key, references **Professor(ProfID)**.

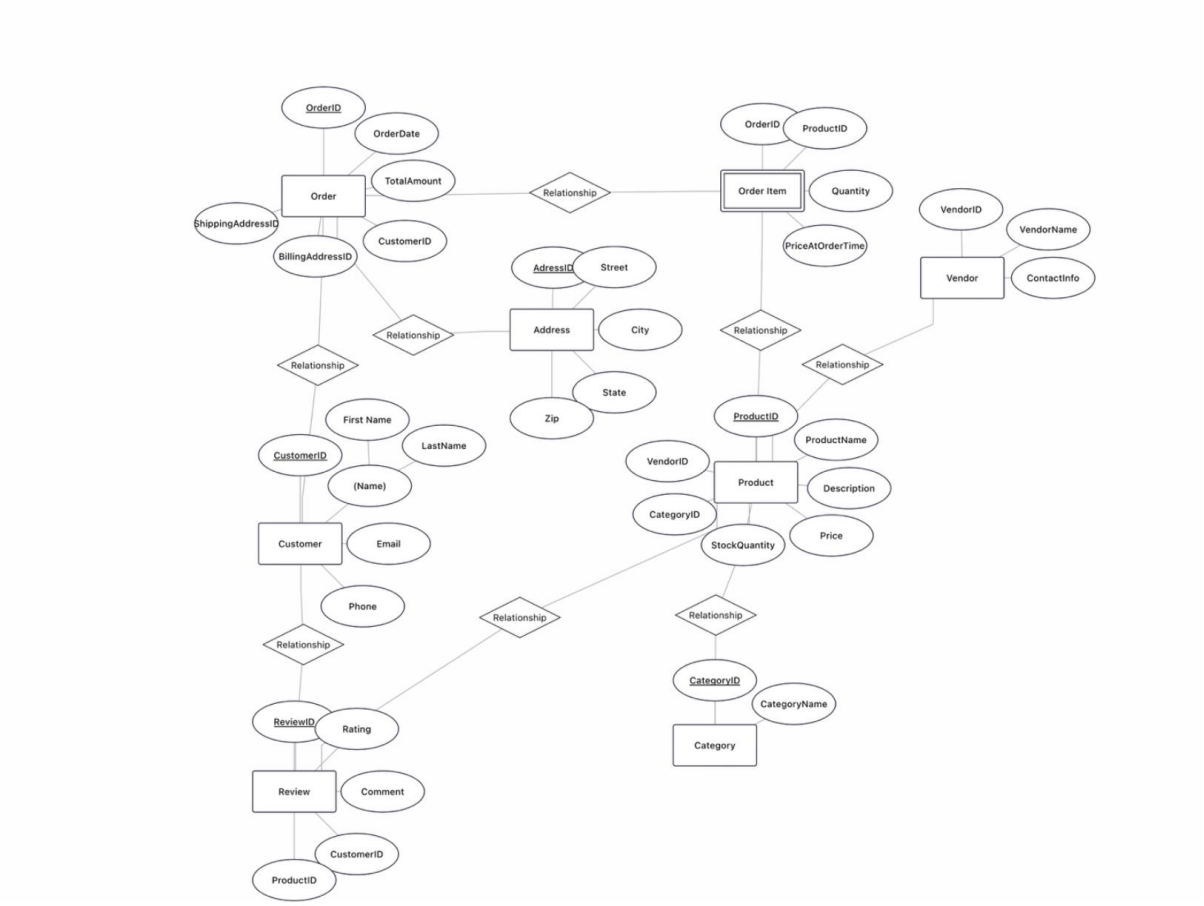
- **Course - Department**
- DepartmentCode is a foreign key, references **Department(DeptCode)**.
- **Department - Professor**
- ChairID is a foreign key, references **Professor(ProfID)**.
- **Enrollment - Student**
- StudentID is a foreign key, references **Student(StudentID)**.
- **Enrollment - Course**
- CourseID is a foreign key, references **Course(CourseID)**.

Part 2: ER Diagram Construction

Task 2.1: Hospital Management System



Task 2.2: E-commerce Platform



Part 4: Normalization Workshop

Task 4.1: Denormalized Table Analysis

1. Functional Dependencies (FDs):

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

2. Problems:

- **Redundancy:** student data, project data, and supervisor data are repeated.
- **Update anomaly:** if a supervisor's department changes, many rows must be updated.
- **Insert anomaly:** cannot insert a project without a student.
- **Delete anomaly:** deleting the last student removes project information.

3. 1NF:
Table is already in 1NF (all attributes are atomic).

4. 2NF:

- PK = (StudentID, ProjectID).
- Partial dependencies exist (Student data depends only on StudentID; project data depends only on ProjectID).
- **Decomposition (2NF):**
    - Student(StudentID, StudentName, StudentMajor)
    - Project(ProjectID, ProjectTitle, ProjectType, SupervisorID, StartDate, EndDate)
    - Participation(StudentID, ProjectID, Role, HoursWorked)

5. 3NF:

- Transitive dependency: ProjectID → SupervisorID → SupervisorName, SupervisorDept.
- **Decomposition (3NF):**
    - Student(StudentID, StudentName, StudentMajor)
    - Supervisor(SupervisorID, SupervisorName, SupervisorDept)
    - Project(ProjectID, ProjectTitle, ProjectType, SupervisorID, StartDate, EndDate)
    - Participation(StudentID, ProjectID, Role, HoursWorked)

Task 4.2: Advanced Normalization

1. Primary Key:
(StudentID, TimeSlot, Room)
→ A row = “a student enrolled in a section”; a section is uniquely identified by (TimeSlot, Room).

2. Functional Dependencies:

- StudentID → StudentMajor
- CourseID → CourseName

- InstructorID → InstructorName
- Room → Building
- (TimeSlot, Room) → CourseID, InstructorID

3. BCNF Check:
Not in BCNF. Many FDs have determinants that are not superkeys.

4. BCNF Decomposition:

- Student(StudentID, StudentMajor)
- Course(CourseID, CourseName)
- Instructor(InstructorID, InstructorName)
- RoomInfo(Room, Building)
- Section(TimeSlot, Room, CourseID, InstructorID)
- Enrollment(StudentID, TimeSlot, Room)

5. Loss of Information:

- Decomposition is **lossless** (can reconstruct the original table by joins).
- All dependencies are preserved (each FD appears in one of the new relations).

Part 5: Design Challenge

Task 5.1: Real-World Application