Database Systems Laboratory Work
Week 2: Relational Model & Keys

Part 1: Key Identification Exercises

Task 1.1: Superkey and Candidate Key Analysis

1)Superkeys-any set of attributes that uniquely identifies a string.
1.EmpID
2.SSN
3.Email
4.Phone
5.EmpID,Name
6.SSN,Department
2)Candidate keys:
1.EmpID
2.SSN
3.Email
3) Which candidate key would you choose as primary key and why?
We can choose EmpID as primary key,due to this is an identifier that belongs to each individual, is stable and does not change, while the ssn and email can be changed
4) Can two employees have the same phone number? Justify your answer based on the data shown.
The table shows that each employee has their own number.
But in real life it is possible that one number is used by several employees. Phone is not a unique attribute, which means it cannot be a candidate key.

Relation B: Course Registration

1. Determine the minimum attributes needed for the primary key :
Minimum attributes needed for the primary key StudentID, CourseCode, Section, Semester, Credits.
2. Explain why each attribute in your primary key is necessary :
   Because according to the business-rule
- A student can take the same course in different semesters
- A student cannot register for the same course section in the same semester
- Each course section in a semester has a fixed credit value
StudentID-individual identifier,CourseCode-Indicates which course you are registering for,Section-prohibits registration for courses of the same section in the same semester, Semester- prohibits same courses,Credits-has fixed value.
3. Identify any additional candidate keys (if they exist):
   They don't exist

Task 1.2: Foreign Key Design

1)Student(AdvisorID->Professor(ProfID))
2)Professor(Department->Department(DeptCode))
3)Course(DepartmentCode->Department(DeptCode))

4)Department(ChairiD->Professor(ProfID))
5)Enrollment(StudentID->Student(StudentID))
(CourseID->Course(CourseID))

Part 2: ER Diagram Construction
Task 2.1: Hospital Management System

1. Identify all entities (specify which are strong and which are weak)
   Strong entities(have own Primary keys):
   -Patient
   -Doctor
   -Department
   -HospitalRoom
   Weak entities(depends on other Primary keys):
   -Appointment
   -Prescription

2. Identify all attributes for each entity (classify as simple, composite, multi-valued, or derived) :

Patient:*PatientID(PK)***,***Name(simple)***,***Birthdate(simple)***,***Address (composite: Street, City, State, Zip)***,***Phone (multi-valued)***,***InsuranceInfo(simple)*

Doctor:*DoctorID(PK)***,***Name(simple)***,***Specialization(multi-valued)***,***Phone(multi-valued)***,***OfficeLocation(simple)*

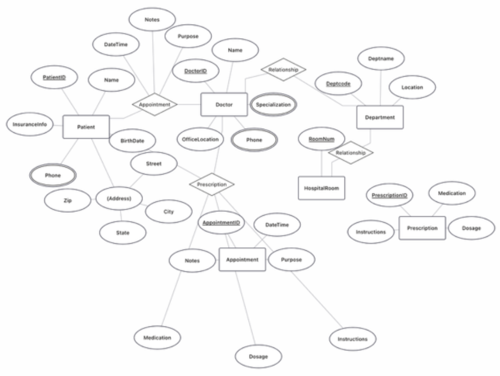
Department:*DeptCode(PK)***,***DeptName(simple)***,***Location(simple)*

HospitalRoom:*RoomNumber(PK)*
Appointment (weak entity):*AppointmentID (PK),PatientID (FK → Patient),DoctorID (FK → Doctor)***,***DateTime (simple)***,***Purpose (simple)***,***Notes (simple)*

Prescription (weak entity):*PrescriptionID (PK),PatientID (FK → Patient)***,***DoctorID (FK → Doctor),Medication (simple)***,***Dosage (simple)***,***Instructions (simple)*

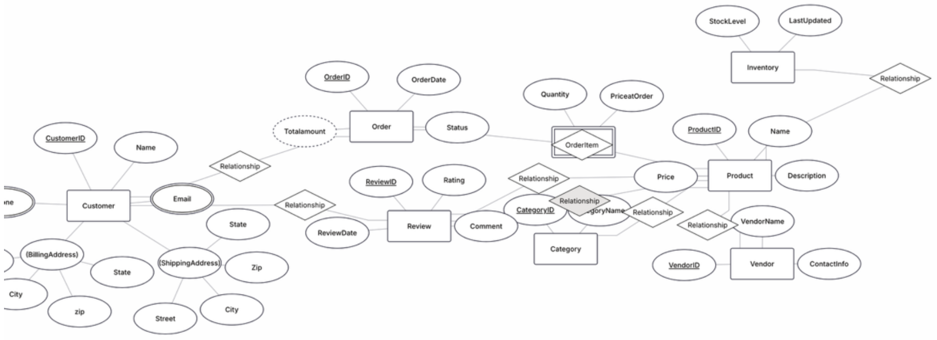
3.Identify all relationships with their cardinalities (1:1, 1:N, M:N)
1)Department-Doctor:1:N(One department contains many doctors,a doctor works in one department)
2)Department-HospitalRoom:1:N(One department contains many rooms,a room belongs to one department)
3)Patient-Appointment-Doctor:M:N(A patient may have many appointments with different doctors,a doctor may see many patients)
4)Doctor-Prescription-Patient:M:N(A doctor can prescribe many prescriptions,a patient can have many prescriptions from different doctors)

4)ER diagarm



Task 2.2: E-commerce Platform

1)ER Diagram



2)Weak entity is OrderItem, since it cannot exist without Order and Product (OrderID, ProductID).

3)Customer-Product,we need review(attributes:rating,comment,review,reviewDate)

Part 4: Normalization Workshop

Task 4.1: Denormalized Table Analysis

1. 1)StudentID(StudentName,StudentMajor)
2)ProjectiD(projecttitle,projecttype,supervisorId)

3)SupervisorID(SupervisorName,supervisordept)

4)StudentId,projected(role,hoursworked,startdate,enddate)

2.Redundancy:

1)StudentName,StudentMajor are repeated for each student project.

ProjectTitle,ProjectType are repeated for each project member.

SupervisorName,SupervisorDept are repeated for each project under thia supervisor.

2)Update,insert,delete anomalies

If the SupervisorName changes, you will have to change it in all rows.

You cannot add a new student without a project.

You cannot add a new project without a participant.

If the last student leaves the project, we lose the project data.

3)Primary keys:

StudentID,ProjectID

Partial dependencies: StudentID → StudentName, StudentMajor, ProjectID → ProjectTitle, ProjectType, SupervisorID

StudentID, StudentName, StudentMajor
ProjectID, ProjectTitle, ProjectType, SupervisorID
SupervisorID, SupervisorName, SupervisorDept
StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate

4) Transitive dependencies: SupervisorID → SupervisorName, SupervisorDept
Student(StudentID, StudentName, StudentMajor)
Project(ProjectID, ProjectTitle, ProjectType, SupervisorID)
Supervisor(SupervisorID, SupervisorName, SupervisorDept)
StudentProject(StudentID, ProjectID, Role, HoursWorked, StartDate, EndDate)

Task 4.2: Advanced Normalization

1)Primary key:StudentID,CourseID,TimeSlot
2) StudentID-StudentMajor
CourseID - CourseName

InstructorID -InstructorName
(TimeSlot, Room) -Building
(CourseID, TimeSlot, Room) – InstructorID
3)In BCNF every FD must have super key, in our case we don't have superkey.So,we conclude that its not BCNF.
4)*Student(StudentID, StudentMajor) -> (StudentID, CourseID, CourseName, InstructorID, InstructorName, TimeSlot, Room, Building)*
Course(CourseID, CourseName)-> (StudentID, CourseID, InstructorID, InstructorName, TimeSlot, Room, Building)

Instructor(InstructorID, InstructorName)-> (StudentID, CourseID, InstructorID, TimeSlot, Room, Building)

RoomSchedule(TimeSlot, Room, Building)-> (StudentID, CourseID, InstructorID, TimeSlot, Room)

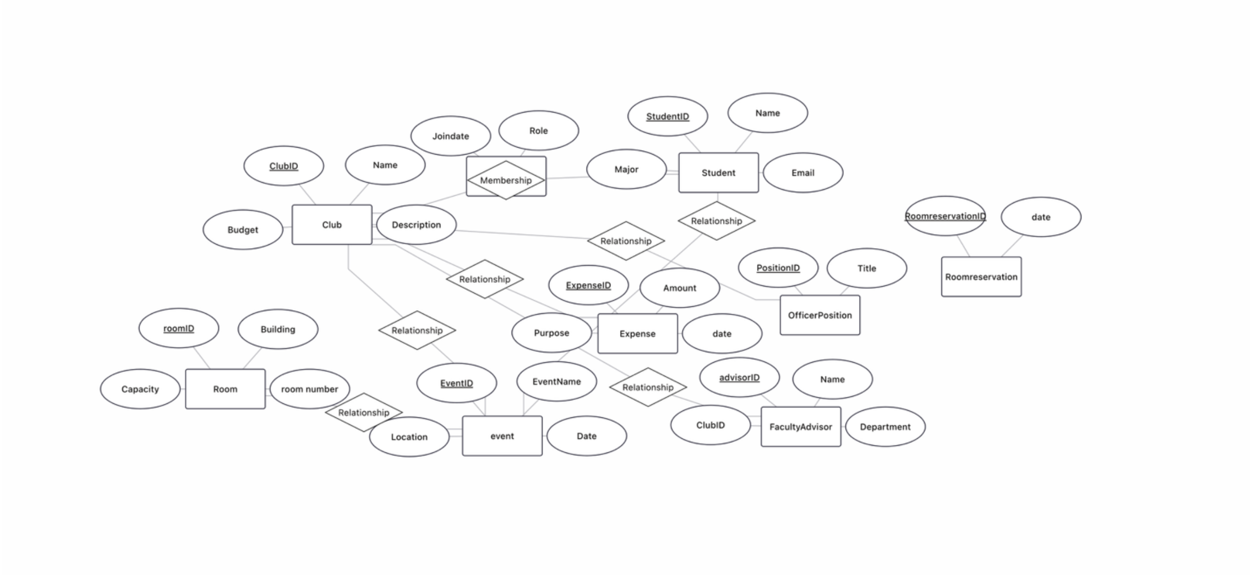
Enrollment(StudentID, CourseID, InstructorID, TimeSlot, Room)

Now in each table all dependencies come from keys, which means the schema is in BCNF.

5) When decomposed, the links between CourseSection and Instructor/Room remain via FK, so the information is preserved.

Task 5.1: Real-World Application

1)



3) How to store club officers (president, treasurer, secretary, etc.)?

Move OfficerPosition (PositionID, ClubID, StudentID, Title) to a separate table.It is more flexible: you can store several officers of the same position (for example, two "Event Managers").

4) Show all clubs that share at least one member in common.

Find the faculty advisor for each club.

Show the total expenses per club for the current semester.