Database Systems Laboratory Work

Week 2: Relational Model & Keys

Prerequisites: Understanding of basic database concepts
Tools Required: ERDPlus, Draw.io (or similar ER tool), DB Fiddle, pen and paper

---

Overview

This laboratory work provides hands-on experience with relational database fundamentals including key identification, ER modeling, schema mapping, and normalization techniques. You will work through progressively complex scenarios that mirror real-world database design challenges.

Part 1: Key Identification Exercises

Task 1.1: Superkey and Candidate Key Analysis

Given the following relations, identify all superkeys and candidate keys:

Relation A: Employee

```
Employee(EmpID, SSN, Email, Phone, Name, Department, Salary)

Sample Data:
```

EmpID	SSN	Email	Phone	Name	Department	Salary
101	123-45-6789	john@company.com	555-0101	John	IT	75000
102	987-65-4321	mary@company.com	555-0102	Mary	HR	68000
103	456-78-9123	bob@company.com	555-0103	Bob	IT	72000

Your Tasks: 1. List at least 6 different superkeys 2. Identify all candidate keys 3. Which candidate key would you choose as primary key and why? 4. Can two employees have the same phone number? Justify your answer based on the data shown.

Relation B: Course Registration

```
Registration(StudentID, CourseCode, Section, Semester, Year, Grade, Credits)

Business Rules:
- 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
```

Your Tasks: 1. Determine the minimum attributes needed for the primary key 2. Explain why each attribute in your primary key is necessary 3. Identify any additional candidate keys (if they exist)

Task 1.2: Foreign Key Design

Design the foreign key relationships for this university system:

Given Tables:

```
Student(StudentID, Name, Email, Major, AdvisorID)
Professor(ProfID, Name, Department, Salary)
Course(CourseID, Title, Credits, DepartmentCode)
Department(DeptCode, DeptName, Budget, ChairID)
Enrollment(StudentID, CourseID, Semester, Grade)
```

Your Tasks: 1. Identify all foreign key relationships

Part 2: ER Diagram Construction

Task 2.1: Hospital Management System

Scenario: Design a database for a hospital management system.

Requirements:

- **Patients** have unique patient IDs, names, birthdates, addresses (street, city, state, zip), phone numbers (multiple allowed), and insurance information
- **Doctors** have unique doctor IDs, names, specializations (can have multiple), phone numbers, and office locations
- **Departments** have department codes, names, and locations
- **Appointments** track which patient sees which doctor at what date/time, the purpose of visit, and any notes
- **Prescriptions** track medications prescribed by doctors to patients, including dosage and instructions
- **Hospital Rooms** are numbered within departments (room 101 in Cardiology is different from room 101 in Neurology)

Your Tasks: 1. Identify all entities (specify which are strong and which are weak) 2. Identify all attributes for each entity (classify as simple, composite, multi-valued, or derived) 3. Identify all relationships with their cardinalities (1:1, 1:N, M:N) 4. Draw the complete ER diagram using proper notation 5. Mark primary keys

Task 2.2: E-commerce Platform

Scenario: Design a simplified e-commerce database.

Requirements:

- **Customers** place **Orders** for **Products**
- **Products** belong to **Categories** and are supplied by **Vendors**
- **Orders** contain multiple **Order Items** (quantity and price at time of order)
- **Products** have reviews and ratings from customers
- Track **Inventory** levels for each product
- **Shipping addresses** can be different from customer billing addresses

Your Tasks: 1. Create a complete ER diagram 2. Identify at least one weak entity and justify why it's weak 3. Identify at least one many-to-many relationship that needs attributes

---

Part 4: Normalization Workshop

Task 4.1: Denormalized Table Analysis

Given Table:

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

Your Tasks: 1. **Identify functional dependencies:** List all FDs in the format A → B 2. **Identify problems:** - What redundancy exists in this table? - Give specific examples of update, insert, and delete anomalies 3. **Apply 1NF:** Are there any 1NF violations? How would you fix them? 4. **Apply 2NF:** - What is the primary key of this table? - Identify any partial dependencies - Show the 2NF decomposition 5. **Apply 3NF:** - Identify any transitive dependencies - Show the final 3NF decomposition with all table schemas

Task 4.2: Advanced Normalization

Given Table:

```
CourseSchedule(StudentID, StudentMajor, CourseID, CourseName,
               InstructorID, InstructorName, TimeSlot, Room, Building)
```

Business Rules:

- Each student has exactly one major
- Each course has a fixed name
- Each instructor has exactly one name
- Each time slot in a room determines the building (rooms are unique across campus)
- Each course section is taught by one instructor at one time in one room
- A student can be enrolled in multiple course sections

Your Tasks: 1. Determine the primary key of this table (hint: this is tricky!) 2. List all functional dependencies 3. Check if the table is in BCNF 4. If not in BCNF, decompose it to BCNF showing your work 5. Explain any potential loss of information in your decomposition

---

Part 5: Design Challenge

Task 5.1: Real-World Application

Scenario: Your university wants to track student clubs and organizations with the following requirements:

System Requirements:

- Student clubs and organizations information
- Club membership (students can join multiple clubs, clubs have multiple members)
- Club events and student attendance tracking
- Club officer positions (president, treasurer, secretary, etc.)
- Faculty advisors for clubs (each club has one advisor, faculty can advise multiple clubs)
- Room reservations for club events
- Club budget and expense tracking

Your Tasks: 1. Create a complete ER diagram for this system 2. Convert your ER diagram to a normalized relational schema 3. Identify at least one design decision where you had multiple valid options and explain your choice 4. Write 3 example queries that your database should support (in English, not SQL)

Example Query Format: - "Find all students who are officers in the Computer Science Club" - "List all events scheduled for next week with their room reservations"

Lab Deliverables

What to Submit:

1. **Complete Solutions:** Detailed answers to all tasks with clear explanations
2. **ER Diagrams:** Hand-drawn or digital diagrams for Tasks 2.1, 2.2, and 5.1
3. **Normalization Work:** Step-by-step decomposition showing all intermediate steps
4. **Relational Schemas:** Complete table definitions with primary keys, foreign keys, and data types

Submission Format:

- **File Format:** PDF document with clear headings for each task
- **Diagrams:** Include all ER diagrams (scanned hand-drawings or digital files)
- **Tables:** Use clear formatting for relational schemas
- **Page Limit:** No strict limit, but aim for clarity and conciseness

Common Mistakes to Avoid:

- Forgetting to underline primary keys in relational schemas
- Missing foreign key relationships between tables
- Stopping normalization at 2NF instead of continuing to 3NF
- ER diagrams missing cardinality constraints or participation indicators
- Weak entities without proper composite keys including owner's key
- Confusing superkeys with candidate keys

Additional Resources:

- Database textbook chapters on ER modeling and normalization
- Online ER diagram tools: Draw.io, Lucidchart, ERDPlus
- Course slides and lecture notes

Good luck! This laboratory work will provide you with hands-on experience in the fundamental skills every database designer needs. Take your time to understand each concept thoroughly, as these form the foundation for all advanced database topics