Lab W05 – Logical Design and Normalization

Objective: You will translate the ERD diagram into a normalized relational schema and justify design decisions using normalization theory.

In earlier labs, you explored the problem domain, outlined system requirements, and created a conceptual ERD to model the necessary data. Now, in this lab, you will transform that conceptual model into a logical relational design and assess it through normalization principles. This stage acts as a crucial link between high-level design and physical implementation. The aim is not just to generate tables but also to ensure your design maintains data integrity, reduces redundancy, and aligns with business rules.

Note: Do not use generative AI (considered a violation of academic integrity).

Relational Schema Design

Using your ERD, produce a logical relational schema using the relational schema notation we covered in the lectures. Your schema must include:

- A relation (table) for each entity and associative entity
- Clearly identified primary keys
- Foreign keys representing relationships
- Attributes derived from the ERD, placed in appropriate relations

At this stage:

- Do not include DBMS-specific syntax
- Do not choose data types
- Do not include indexes

Functional Dependencies

For each relation in your schema:

- Identify the relevant functional dependencies (full, partial, transitive, multivalued)
- Clearly indicate determinants and dependent attributes using the normalization notation in the lectures.
- Justify any non-obvious dependencies.

You are not required to list trivial dependencies, but you should include all dependencies that affect normalization decisions. This section demonstrates your understanding of how attributes relate to one another, not just how tables are structured.

Normalization Analysis

Using the functional dependencies identified above, analyze each relation and normalize your schema to Third Normal Form (3NF) or Boyce-Codd Normal Form (BCNF) where appropriate.

For each relation:

- State the highest normal form it currently satisfies
- Identify violations, if any
- Show how the relation is decomposed to reach the target normal form
- If a relation is intentionally left below BCNF:
    - Clearly explain why

    - Describe the trade-offs involved
    - Explain how integrity will be preserved

Blindly normalizing without justification will not receive full credit.

Design Justification and Trade-Offs

Write a short narrative explaining:

- How your logical design supports the business rules
- Where normalization improves data integrity
- Where strict normalization may introduce practical drawbacks

This section should demonstrate that you understand normalization as a design tool, not a mechanical exercise.

Evaluation Criteria

Grading Breakdown

- Objective Criteria: 65%
- Subjective Evaluation: 35%

Both parts matter. A neat but incorrectly modeled diagram will not score well, and neither will a technically correct schema that is unclear, incomplete, or poorly reasoned.

Objective Criteria (65 points)

The objective portion of the grade is based on whether the ERD correctly applies Crow's Foot notation and fundamental data modeling principles. This includes:

- Correct mapping from ERD to relations
- Correct identification of keys and dependencies
- Correct application of normalization rules

Subjective Evaluation (35 points)

- Reasonableness of decomposition decisions
- Quality of explanations and justifications
- Consistent with earlier requirements and business rules

Submission

- Use one of the provided templates.
- Name the document 'lastSchema' or 'lastlastSchema' like we did in the previous lab. Upload and commit the diagram to the labs folder.
- Place "done" in the Moodle submission textbox for Lab W05.