A Practical Automated Transformation of Entity Relationship Models to Relational Models

Gregor Grambow and Simon Ruttmann
Dept. of Computer Science - Aalen University
Presented by: Gregor Grambow
gregor.grambow@hs-aalen.de

Gregor Grambow

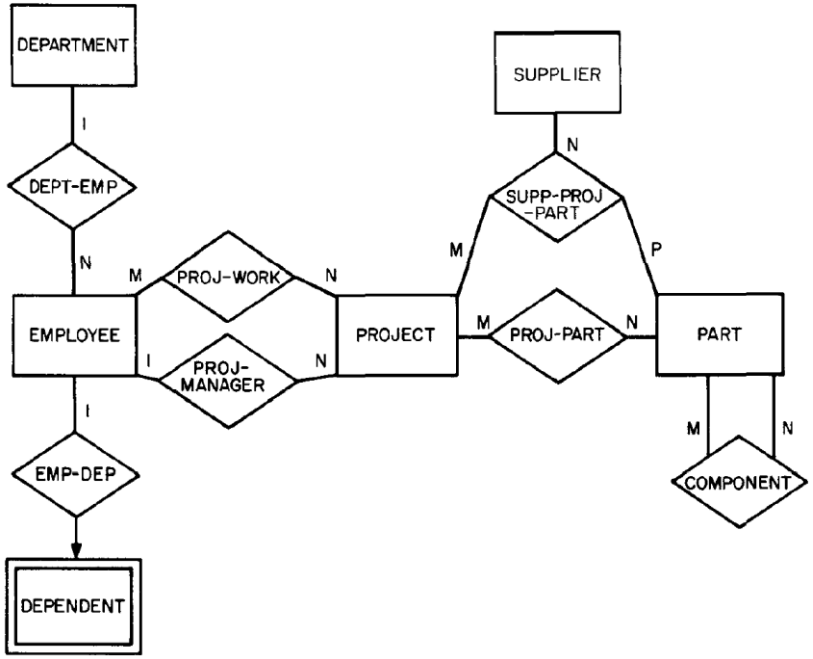
- Professor for modern database technologies and their application at Aalen University
  - Since 2020
- Industry experience
  - Various projects relating to big data, data processing, and process optimization
- Research projects
  - Different projects relating to process optimization in context of Industry 4.0
    - Applied data processing
    - Combination of different data processing technologies
- Research interests
  - Database modeling
  - Graph databases
  - Context processing
  - Process optimization
  - …

Table of Contents

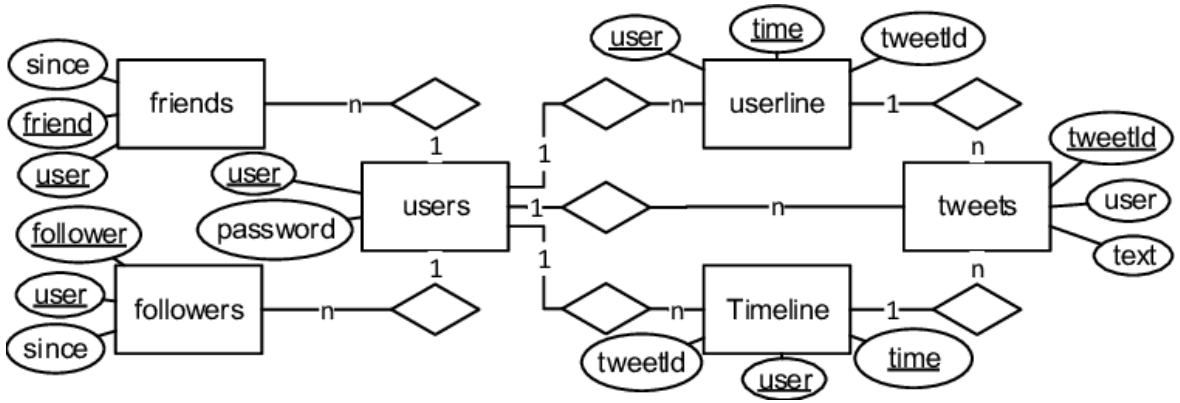
Introduction

Entity Relationship (ER) modeling

- Prevalent option for semantic data modeling
- Introduced in 1976 by Peter Chen
- Focus of active research for decades
- Conversion required for application in relational databases
  - Can be tedious and error-prone
  - Various approaches for standardized transformations have been proposed



An ER Diagram as proposed by Peter Chen
[P. Chen, "The entity-relationship model—toward a unified view of data," ACM Trans. on DB Sys. vol. 1, no. 1, pp. 9-36, 1976.]



An ER Diagram used for schema modeling of current NoSQL systems
[D. Bermbach, S. Müller, J. Eberhardt, and S. Tai "Informed Schema Design for Column Store-Based Database Services," IEEE 8th Int. Conf. on Service-Oriented Computing and Applications, 2015.]

Related Work

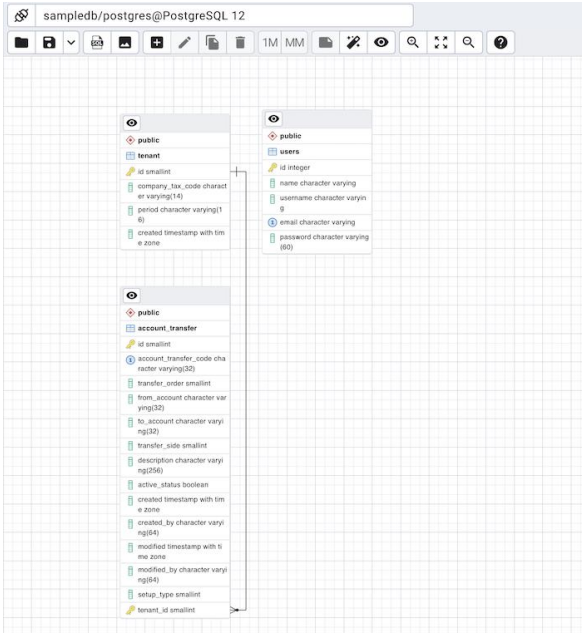
Scientific approaches

- High number of approaches
- Based on different sets of modeling elements
- Rather theoretic – no application

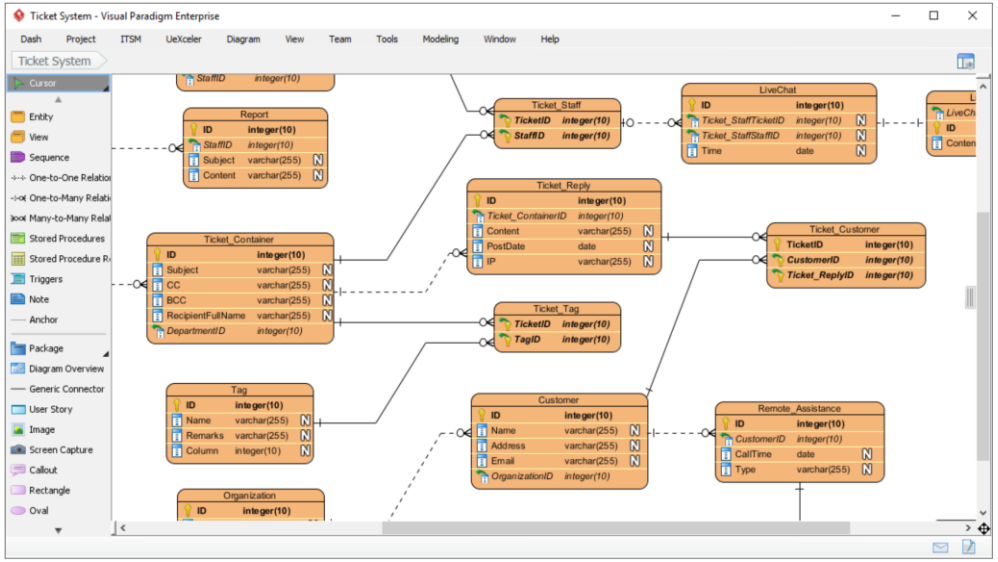
Practical ER editor tools

- High number of tools
- Mostly not proper ER modeling
- No transformation of ER models to relational models

➢ No practical applied transformation in place



pgadmin
[https://www.pgadmin.org/docs/pgadmin4/6.18/erd_tool.html]

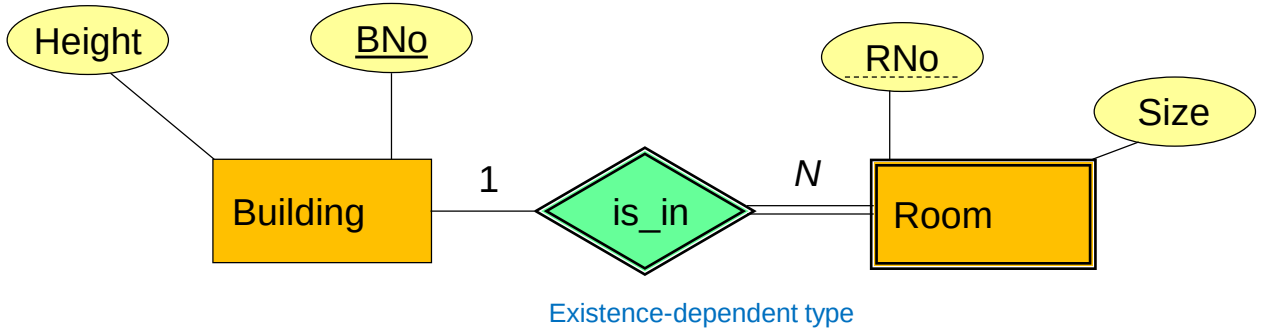


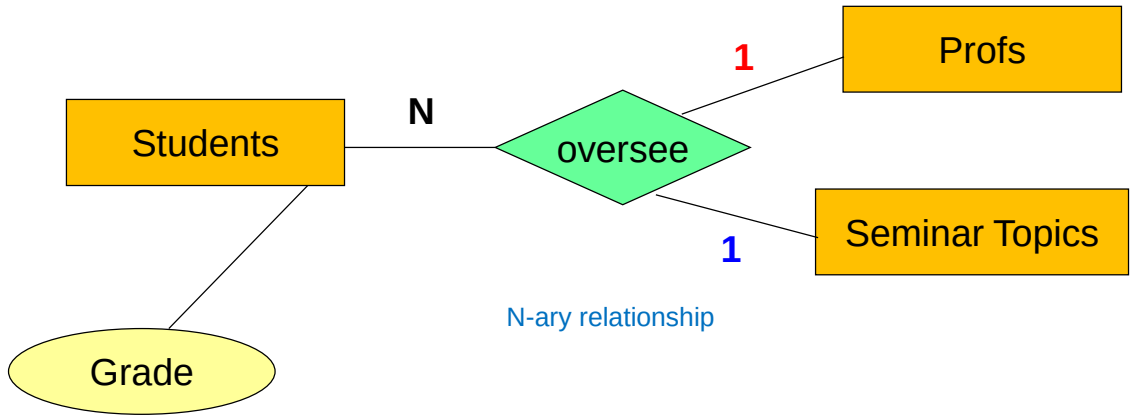
Visual Paradigm
[https://www.visual-paradigm.com/features/database-design-with-erd-tools/]

Definition of the ER model

Based on the original model by Peter Chen with some prevalent extensions

- Modeling elements:
  - Entities
  - Relationships
    - N-ary relationships
  - Attributes
    - For entities and relationships
    - Multi-valued attributes
    - Compound attributes
  - Existence-dependent (weak) types
  - Generalization
    - IsA structures
- Multiple application and combination of elements is possible





Transformation

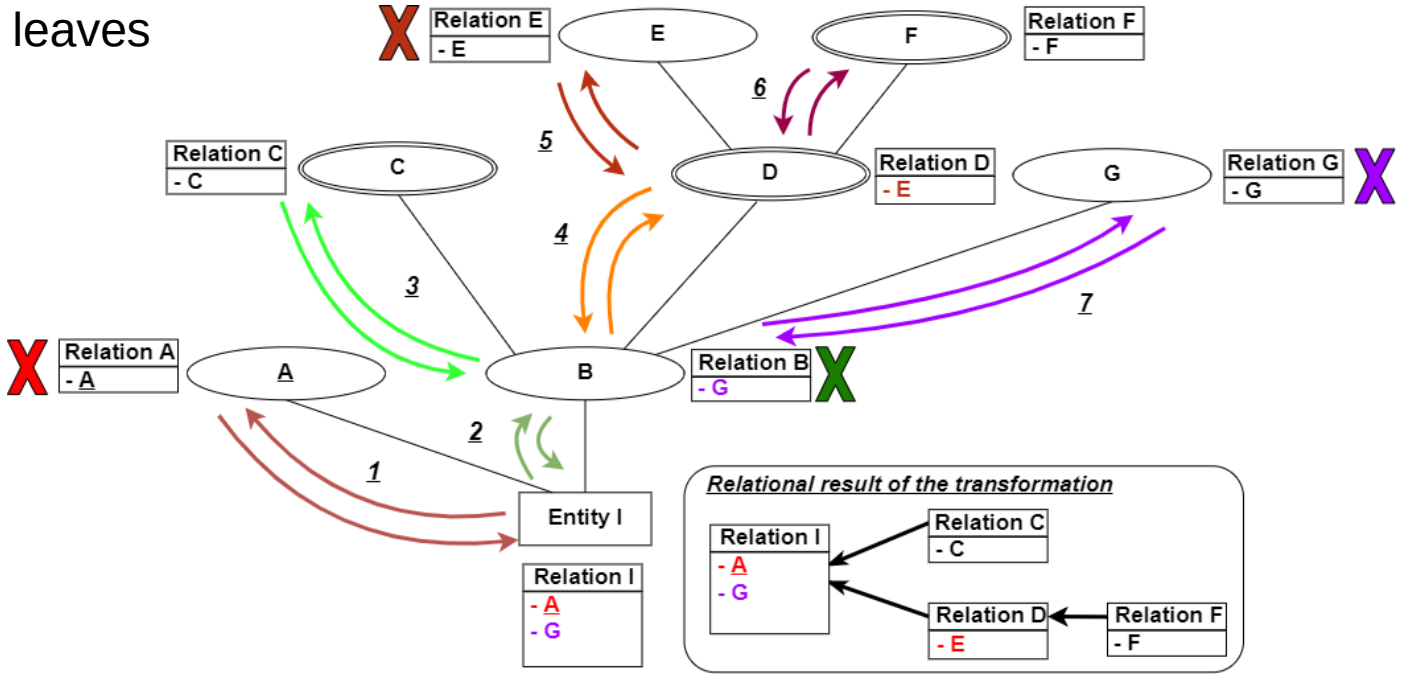
Consists of a set of steps for specific concepts

- Executed sequentially
- Steps:
  - Create a data model structure of ER diagrams
  - Transformation of attributes
  - Transformation of IsA structures
  - Transformation of weak types
  - Transformation of relationships
  - Cascade primary keys for attributes

Transformation: Attributes

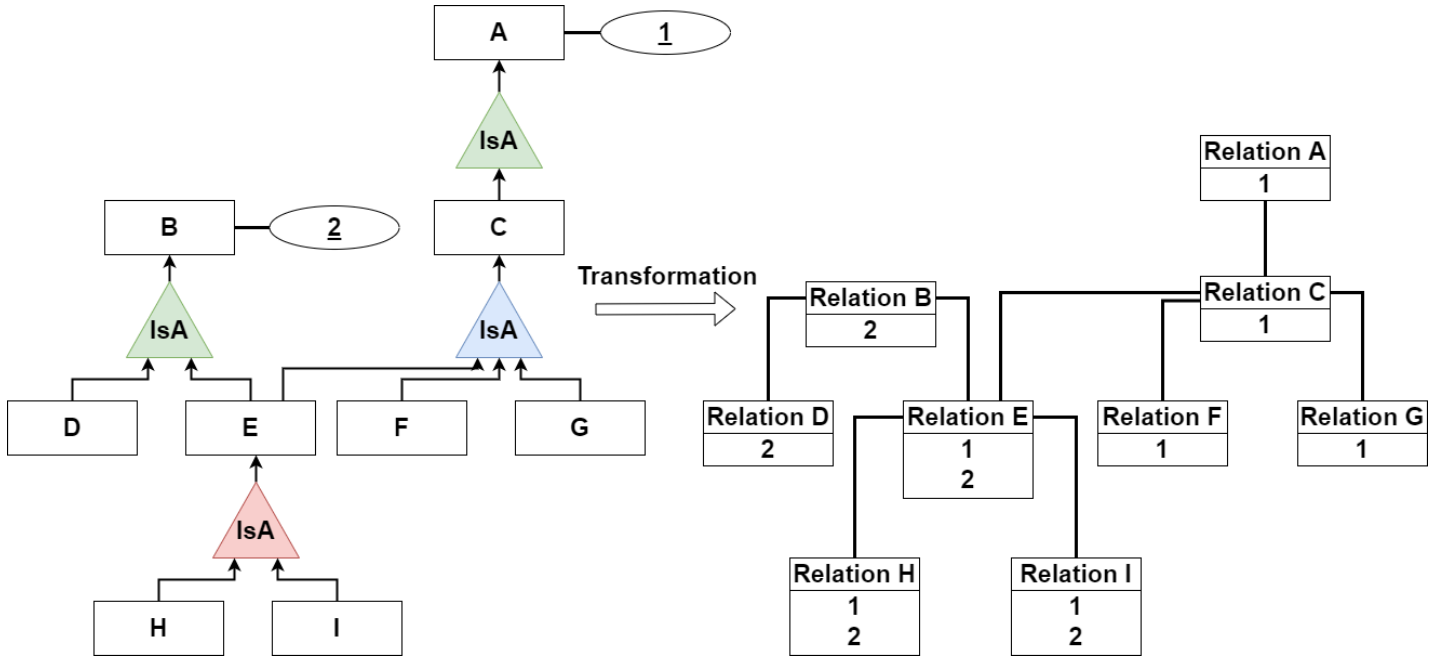
- Attributes stored in a tree structure
- Multi-valued and compound attributes can be contained in any combination

- Only single-valued attributes:
  - Simple post-order traversal checking for leaves
- Including multi-valued attributes:
  - Standalone relation must be created for each one
  - Relations must be linked



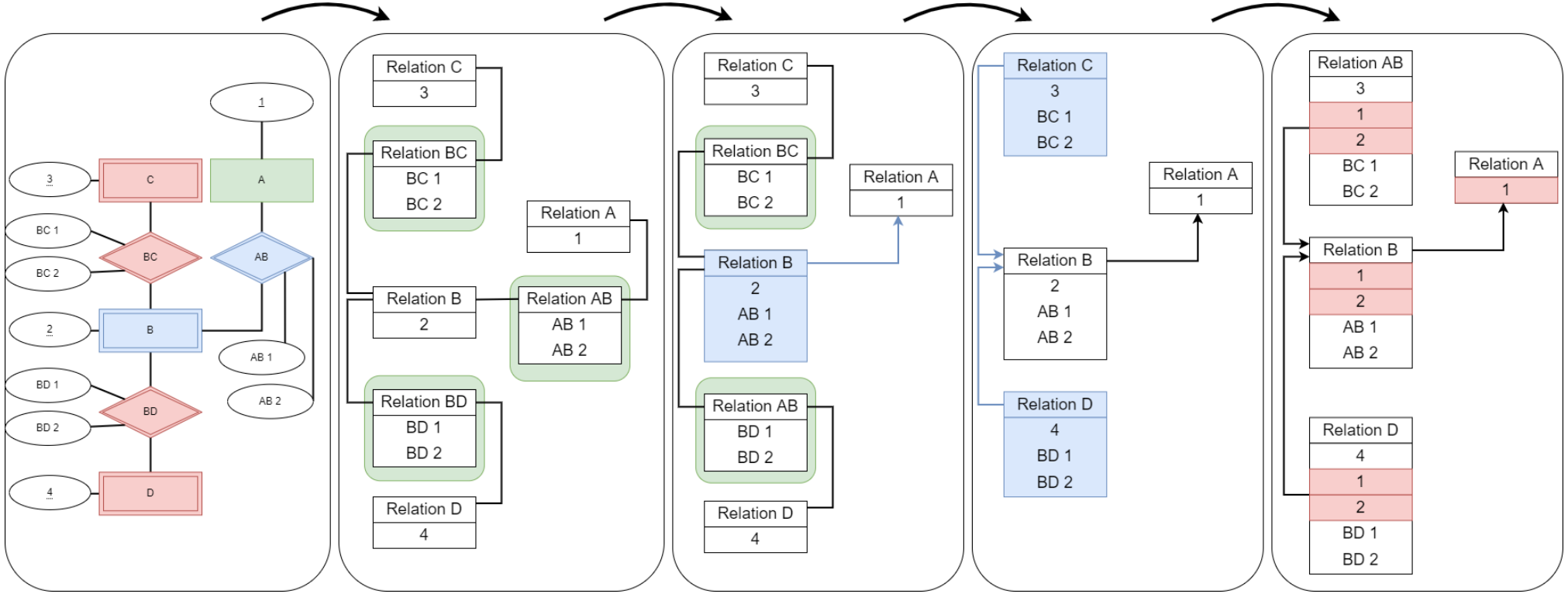
Transformation: IsA structures

- IsA structures are realized by means of foreign key dependencies between the subtypes and supertypes
- Subtypes inherit all primary keys from the supertype
  - Inherited primary keys refer to the upper type as foreign keys
- In case of multiple linked IsA structures
  - ”Higher level” IsA structures have to be translated first
  - That way, lower levels receive primary keys



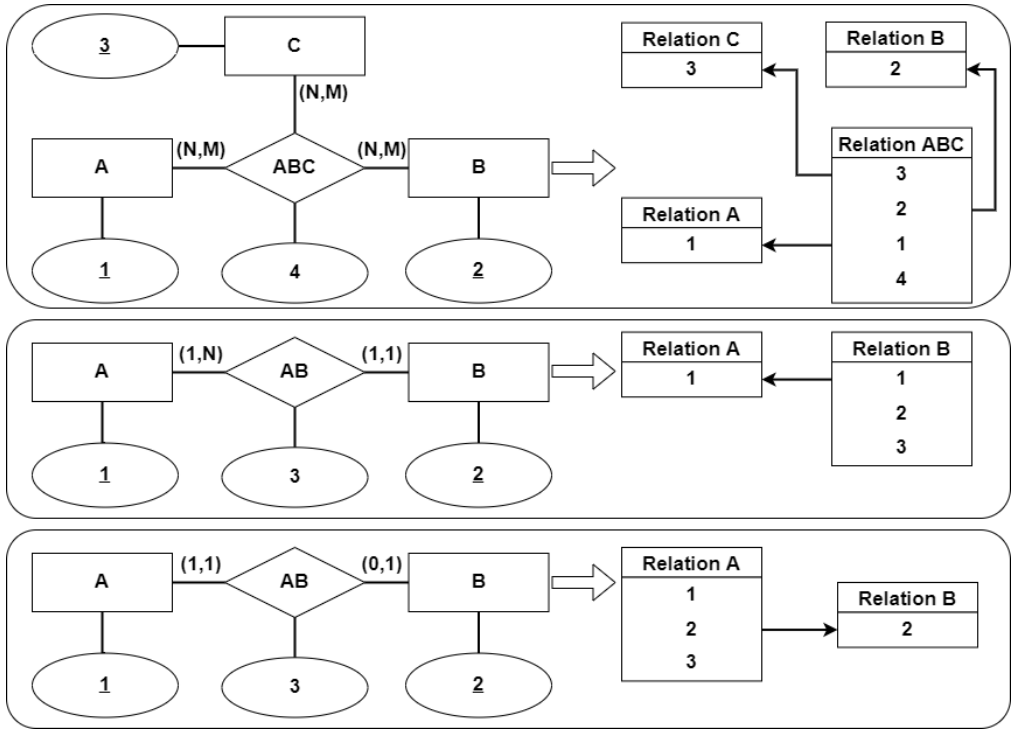
Transformation: Weak types

- Before translating weak types the relations for entities including attributes must be in place
  - This includes IsA structures for potentially inherited attributes
- First, relations are created for all weak relationships and entities
  - After that, they are merged step by step



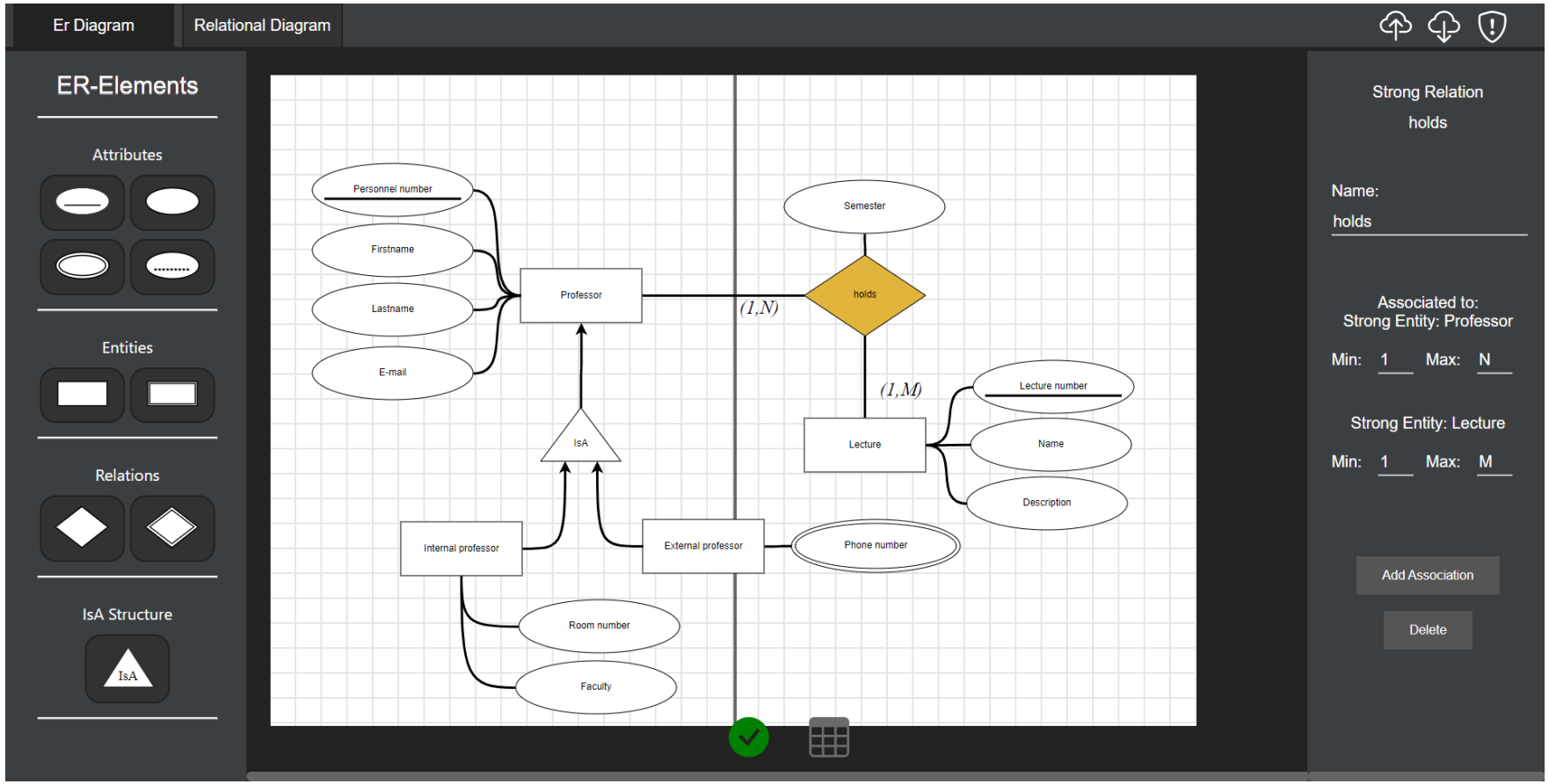
Transformation: Relationships

- Prior to this step, each relationship has its own relation
- This step transforms different types of relationships in a different way:
  - N:M relationships
    - The primary keys of the connected entities are added to the relation
  - 1:N
    - The relationship is resolved by merging the relation of the relationship with the entities relation on the side with the higher cardinality
  - 1:1
    - Translation performed analogously to 1:N relationships, and the entity that receives the foreign keys and relation attributes must be specified for this purpose
    - As Min-Max notation is used, optionality has to be considered



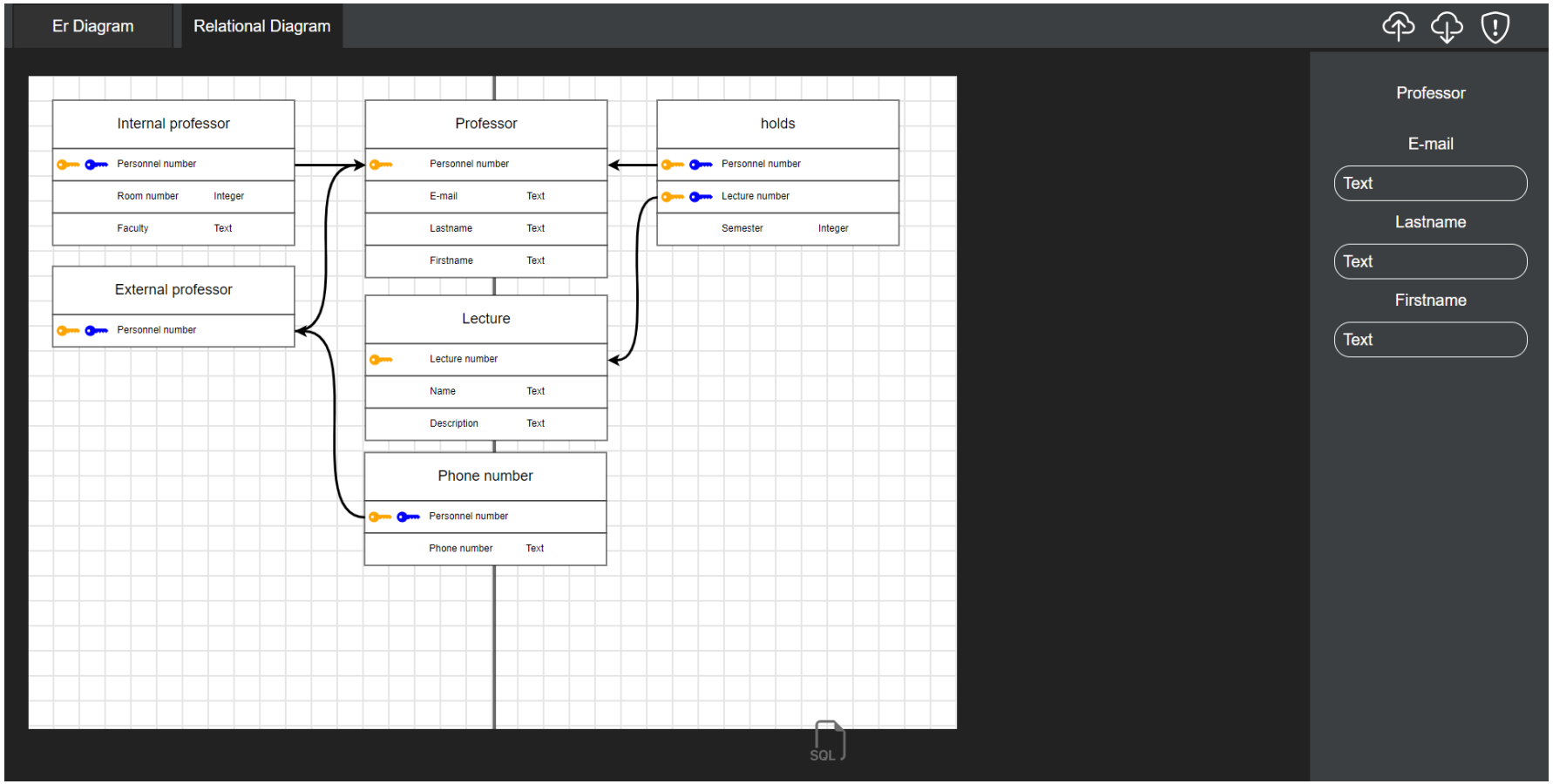
Evaluation

- Focus: Practical applicability
- A web-based editor was created implementing the algorithms
  - ER modeling
  - Validation process for ER diagrams
  - Automated transformation to relational models



Evaluation

- The editor also contains:
  - Visualization of the transformed relational model
  - Save & Load functionality
  - Generation of SQL statements from the relational model
- Editor was used for preliminary tests with different models
- Broader evaluation with user groups is part of future work



Conclusion

- ER modeling is still prevalent
- Many approaches dealt with ER modeling and transformation to relational models
  - Practical application was never achieved

- We created an automated ER-to-Relational transformation approach
  - Extended Peter Chen‘s model with most prevalent concepts
  - Transformation algorithms for all of these concepts

- We evaluated the practical applicability
  - Created and released a graphical editor capable of
    - Visually modeling ER models
    - Automatic transformation to relational models and SQL