Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

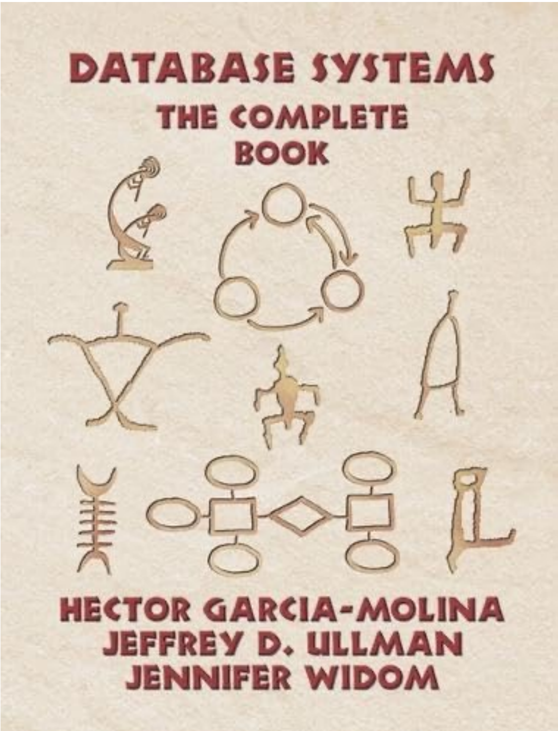
Prof. Dr. Ralf Möller

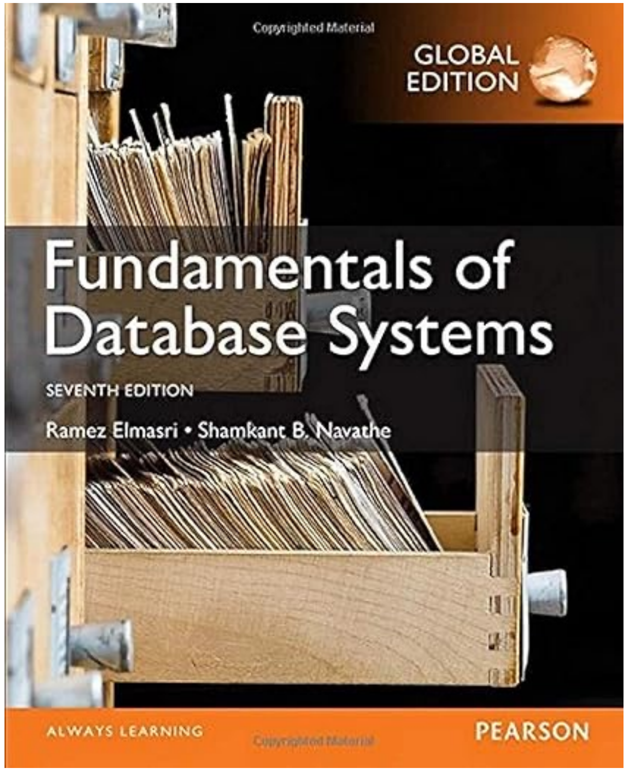
Databases

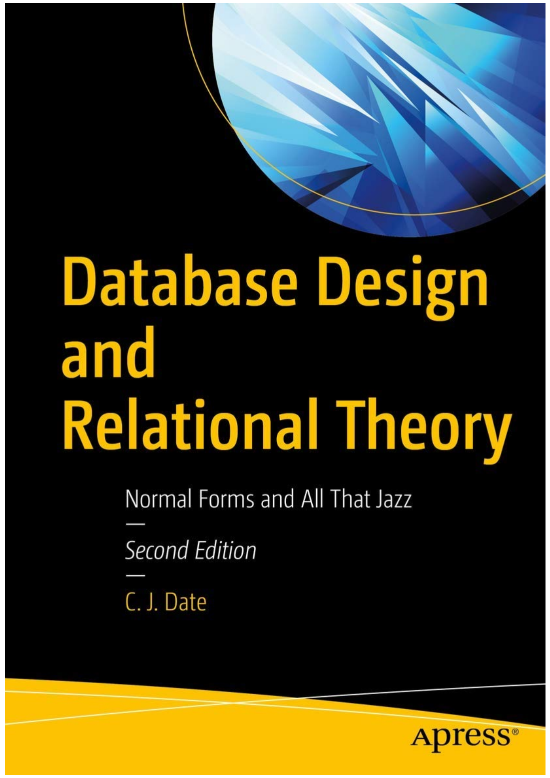
Organizational matters

- **Start**: 16. Oct. 2025
- **Lecture:** each Thursday 1:15 pm – 2:45 pm
- **Seminar:** 3 pm – 3:45 pm (you need to participate)
- **Location:** Warburgstraße 28, Room 4007
- **Web page:** https://www.philosophie.uni-hamburg.de/chai/studium/intellecticsstudiengang.html
- **Register in Moodle**
  - Presentations and Zoom link provided in Moodle

Literature


DATABASE SYSTEMS
THE COMPLETE BOOK
HECTOR GARCIA-MOLINA
JEFFREY D. ULLMAN
JENNIFER WIDOM


Copyrighted Material
GLOBAL EDITION
Fundamentals of Database Systems
SEVENTH EDITION
Ramez Elmasri • Shamkant B. Navathe
ALWAYS LEARNING
PEARSON


Database Design and Relational Theory
Normal Forms and All That Jazz
Second Edition
C. J. Date
Apress

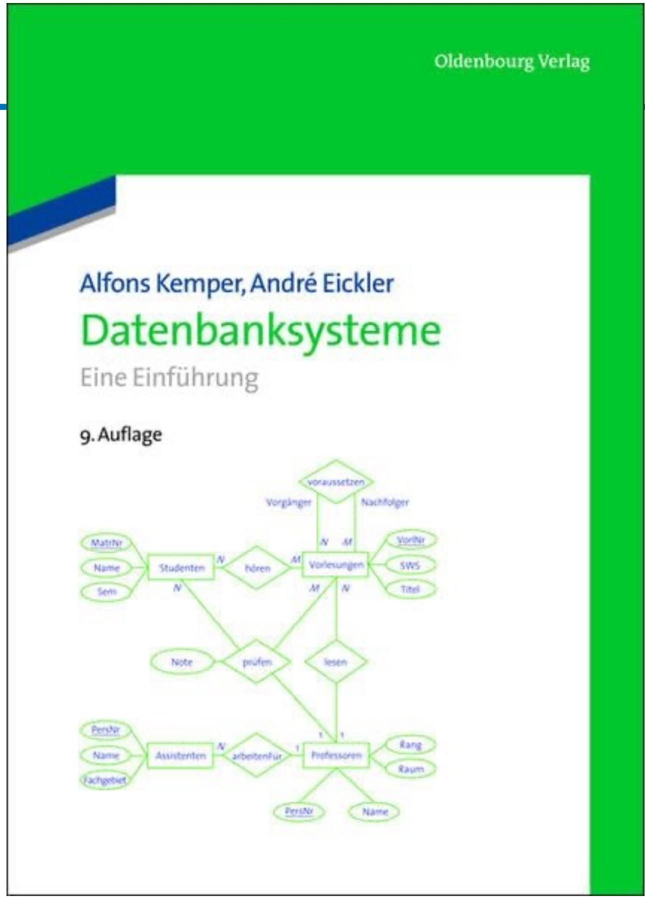
- Literature contains extensive content
- Summarization by this lecture
- Discussion in seminar part
- Available in Moodle
  - Presentations
  - Exercise tasks and solutions

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Acknowledgements

Slides used for this course are partially based on slides provided by Alfons Kemper and André Eickler for their German textbook on database systems



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Overview

1. **Introduction**
   - Models and abstraction
   - Database systems
   - Applications and data independence
2. **Entity-relationship model (ER model)**
3. **Relational data model (RDM)**
   - From ER to RDM
4. **Relational design theory**
   - Relational algebra as a query language
   - Functional dependencies
   - Normal forms
5. **Structured Query Language (SQL)**
   - Data definition
   - Query specification
   - Data manipulation
6. **Query processing**
   - Architecture
   - Indexing
   - Query plans, optimization
7. **Transactions**
   - Transaction processing, schedules, locks
   - Recovery
8. **Other types of databases**
   - Vector databases
   - Data stream query languages

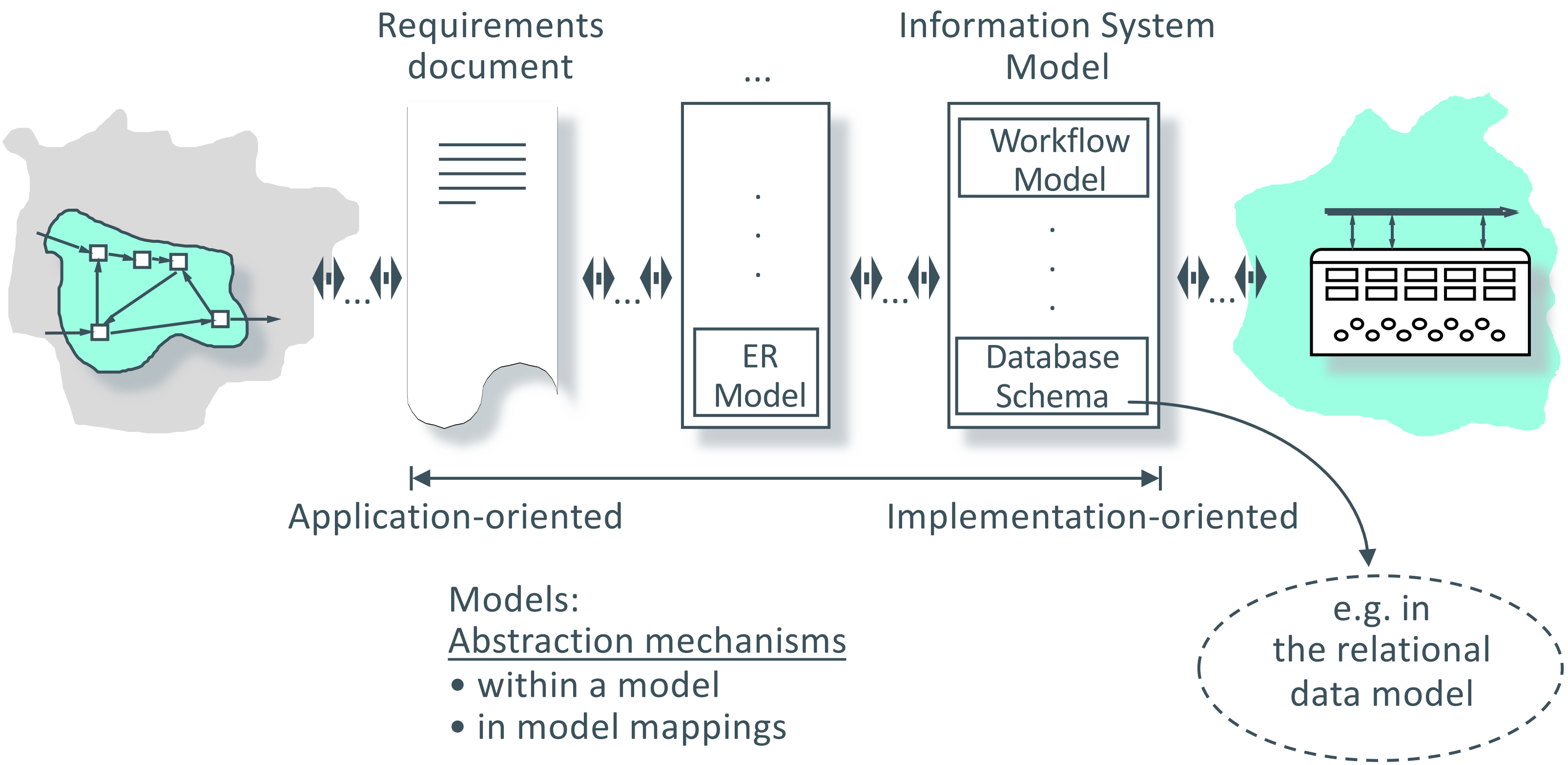
Chapter 1: Introduction

Characteristics of data in databases

- Long data retention period (years, decades)
- Regular structures
- Big data, growing (Giga-, Tera-, Petabytes)
- Access from multiple application programs

Decimal		
Value	SI	
1000	k	kilo
1000^2	M	mega
1000^3	G	giga
1000^4	T	tera
1000^5	P	peta
1000^6	E	exa
1000^7	Z	zetta
1000^8	Y	yotta

Models and Abstraction



Models:
<u>Abstraction mechanisms</u>
- within a model
- in model mappings

P. Chen, The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems 1 (1), pp. 9–36, **1976**
E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Vol. 13, No. 6, pp. 377-387, **1970**

First example of (relational) database

- Database of a winery

Table Winery

Rack	Type	Year	N_bottles
2	Franken	2009	5
1	Baden	2006	3
4	Rheinhessen	2007	10
1	Mosel	2013	2
2	Franken	2010	10

E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Vol. 13, No. 6, pp. 377-387, **1970**

A first query

- Return information about wines such that there are at least 4 bottles

Winery

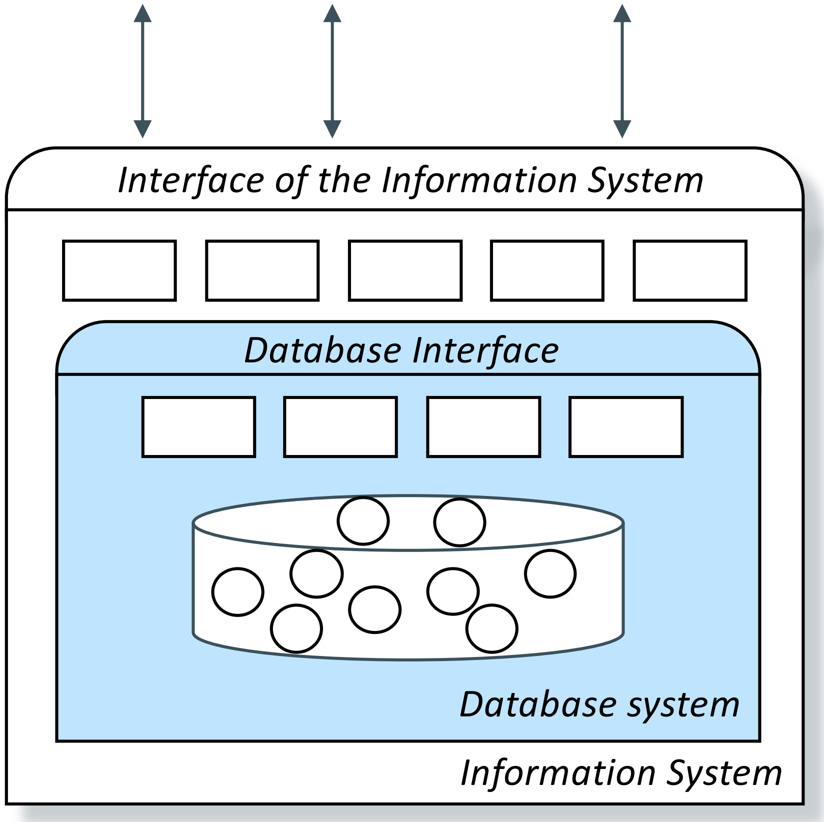
Rack	Type	Year	N_bottles
2	Franken	2009	5
1	Baden	2006	3
4	Rheinhessen	2007	10
1	Mosel	2013	2
2	Franken	2010	10

```
SELECT Type, Rack, Year
FROM Winery
WHERE N_bottles >= 4
```

Type	Rack	Year
Franken	2	2009
Rheinhessen	4	2007
Franken	2	2010

Database Systems

- For realizing information systems

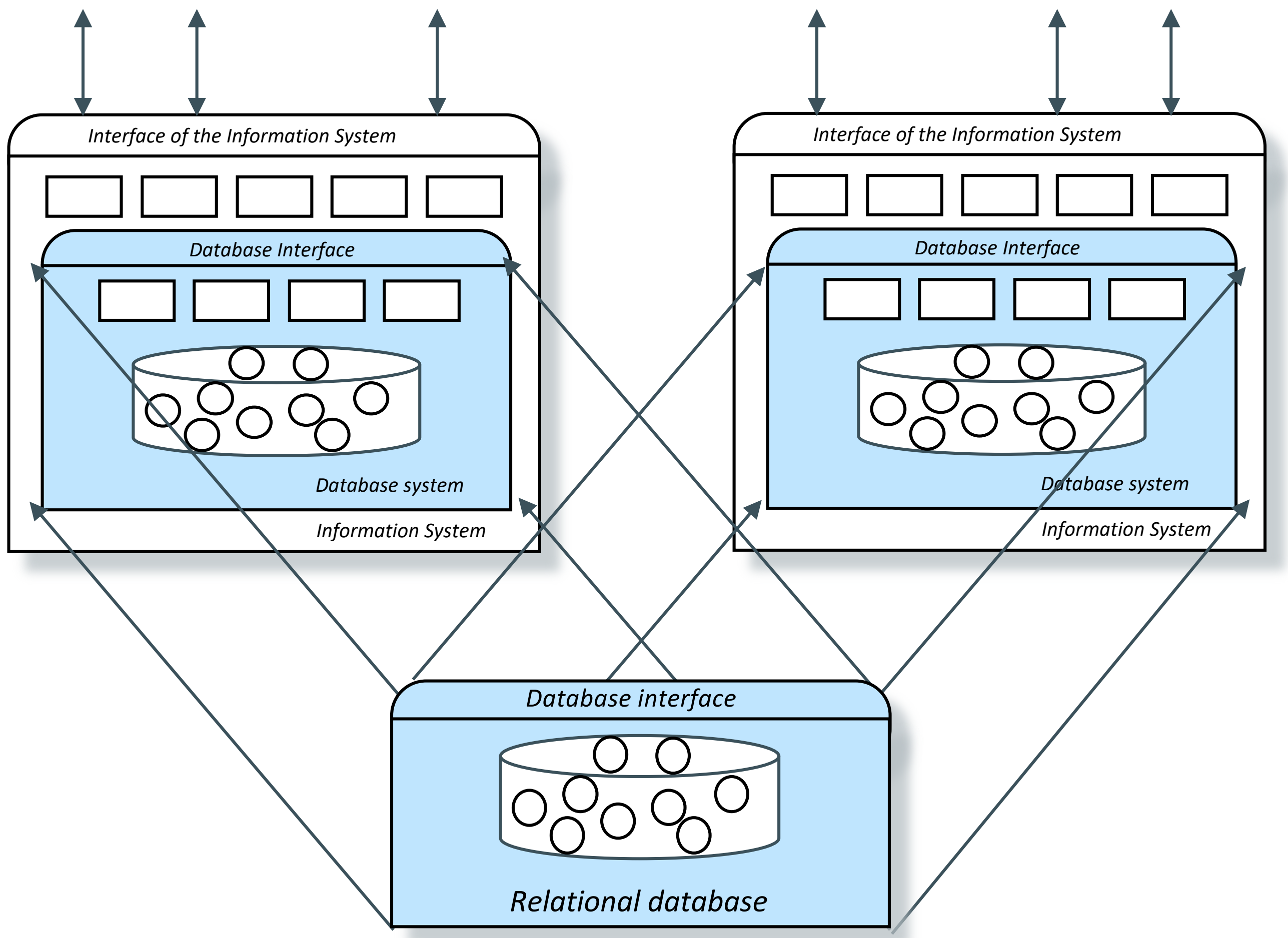


Services for
information representation,
processing, and
integrity assurance

Services of the database system:
Data storage and querying
Integrity maintenance
(with database schema)

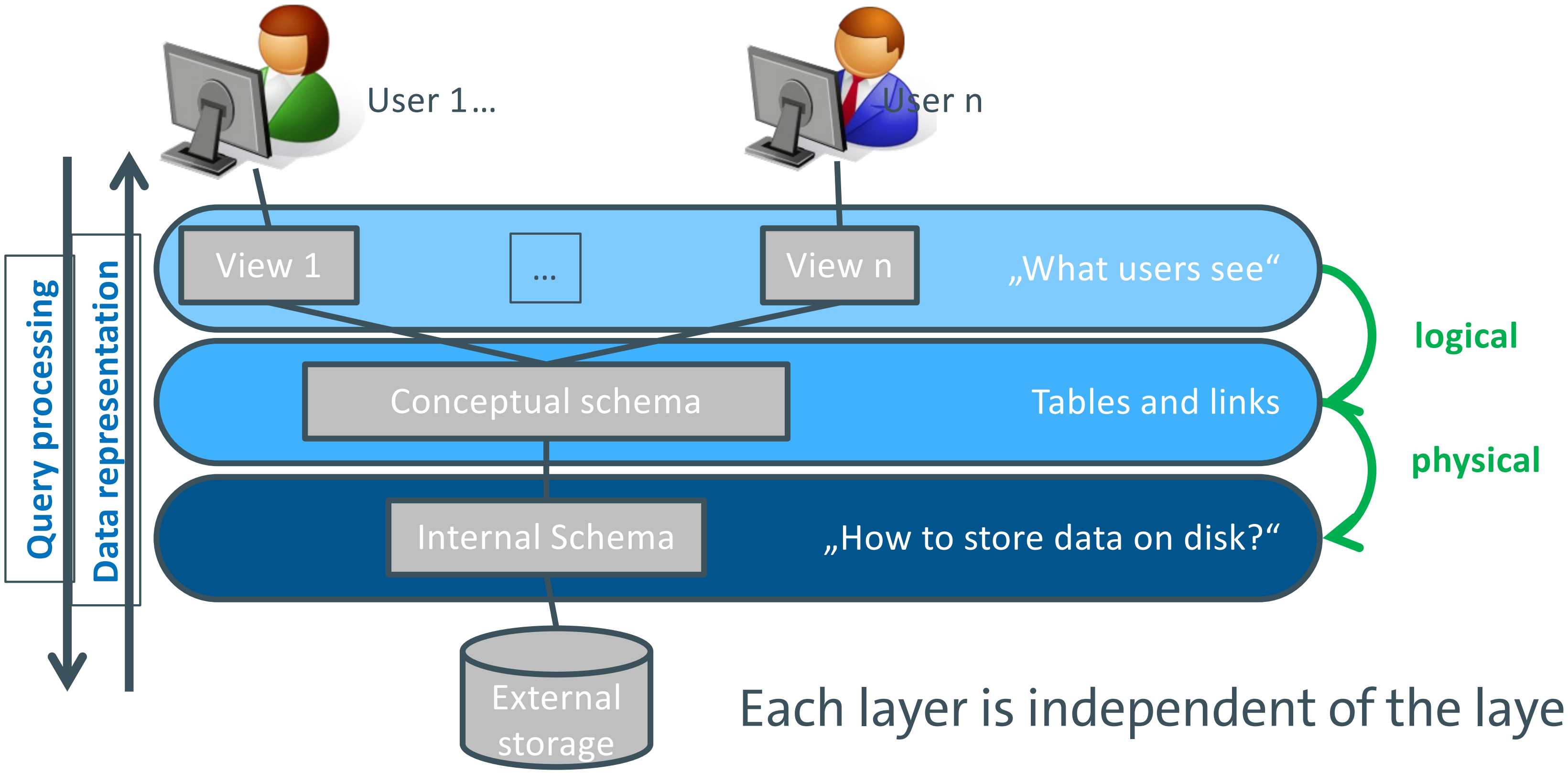
Database state

Views on data



ANSI-SPARC-Architecture

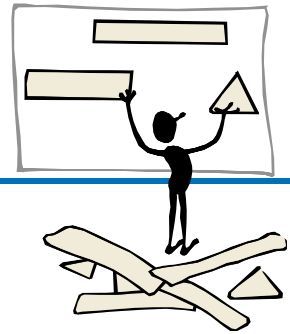
ANSI: **A**merican **N**ational **S**tandards **I**nstitute
SPARC: **S**tandards **P**lanning **a**nd **R**equirement **C**ommittee



Each layer is independent of the layers below

ANSI/X3/SPARC Study Group on Data Base Management Systems, Interim Report. FDT, ACM SIGMOD bulletin. Volume 7, No. 2, **1975**

Final report of the ANSI/X3/SPARC DBS-SG relational database task group. M. Brodie, J.W.. Schmidt (Ed.), SIGMOD Rec. 12, 4, **1982**

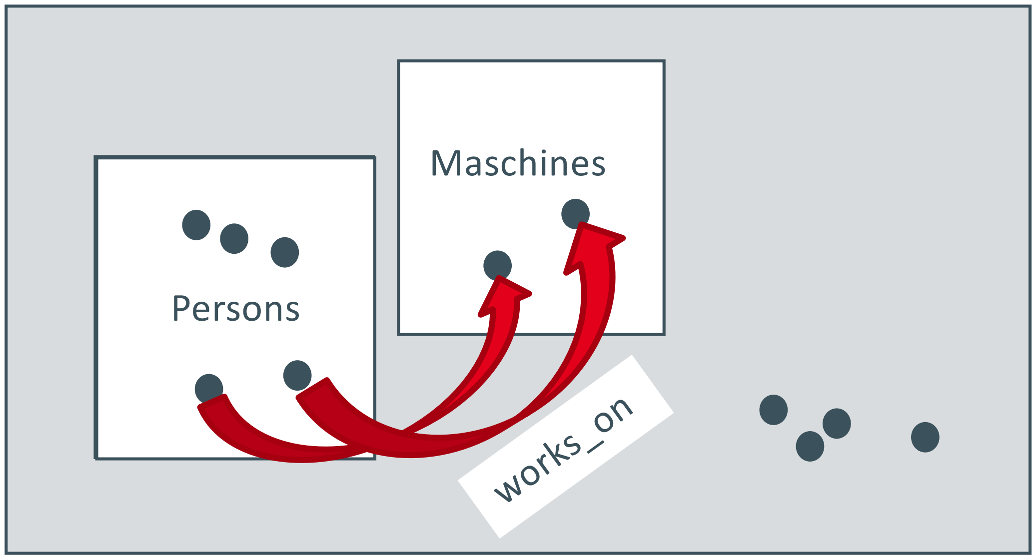


Desideratum: Data independence

- Logical independence
  - Conceptual schema can be changed without requirement to change views
    - Addition of attributes or whole tables
    - Changes in table structure
- Physical independence
  - Internal schema can be changed w/o requirement to change conceptual schema
    - Changes to storage location
    - Changes to storage format
    - Creating of index for query answering optimization

Chapter 2: Entity-Relationship Modeling

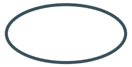
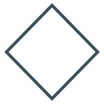
- Graphical notation (for discussions)
  - Objects (aka entities, individuals, or instances) with similar properties are collected into sets (aka object types, entity types or named classes)
  - Each object can be member of multiple classes
  - The set of all members of a class (or an entity type) is called its extension
  - Objects can be set into relation to each other
  - Relations are sets of tuples (aka relationships)
    { (●,●), (●,●) , ... }
  - Relationships are described by relationship types

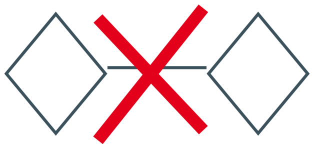
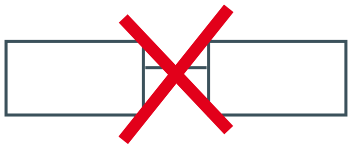


All objects (universe)

● stands for an object

Elements of ER-Diagrams

- Object type  (aka entity type or class; stands for set of objects)
- Value type  (represents basis data types, e.g. numbers or strings; stands for set of values)
- Relation type  (sets of tuples of objects)
- Edges connect object types with value types or relations types

 und 

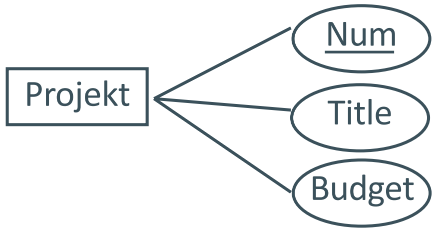
Connections between symbols of the same types are not allowed

P. Chen, The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems 1 (1), pp. 9–36, **1976**

Entity types and attributes

Beispiele:

- Each project is described by
  - a number
  - a title
  - a Budget



Winery example

Rack	Type	Year	N_bottles
2	Franken	2009	5
1	Baden	2006	3
4	Rheinhessen	2007	10
1	Mosel	2013	2
2	Franken	2010	10

- Mathematical meaning: An entity ● is a tuple of values

- Tuples can be characterized by unique values: E.g., there is only one project with a certain number
  - Underlined attributes are called “keys”
  - Keys are underlined in diagrams (see Num)

Generalization or specialization

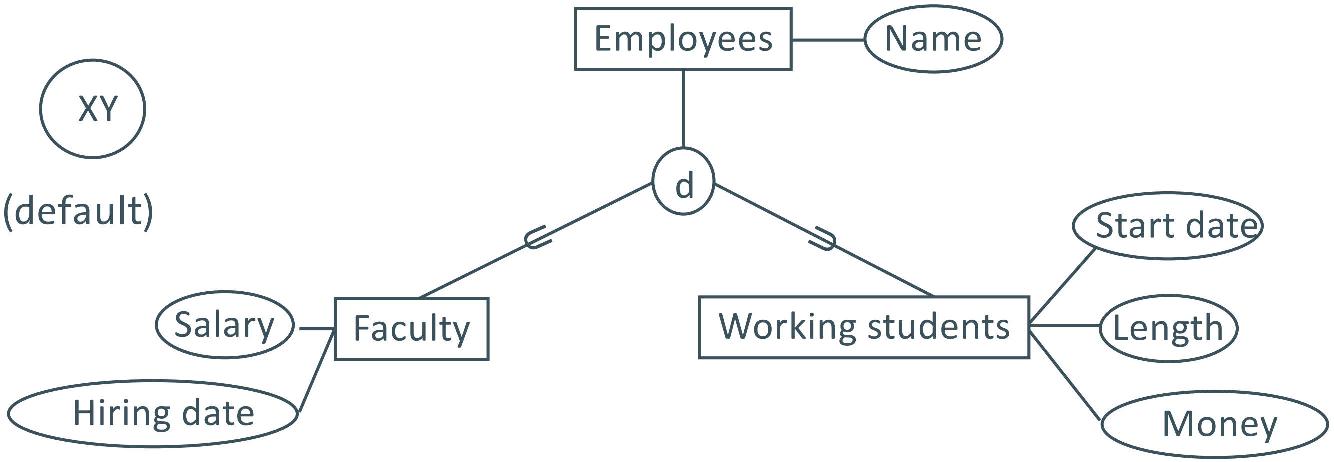
Example:

- Faculty and working students are employees. Permanent employees are not working students.
- The number of employees is equal to the union of the sets of permanent employees and working students, i.e., an employee is either a permanent employee or a working student.

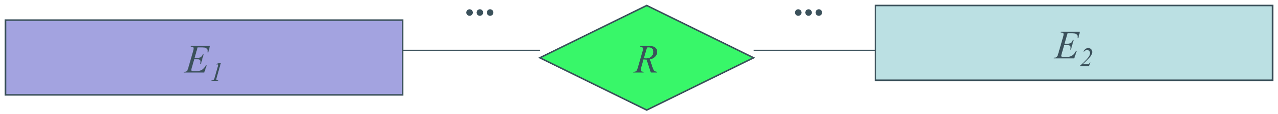
Graphical element

XY

- X=d: disjoint
- X=o: overlapping (default)
- Y=c: covering

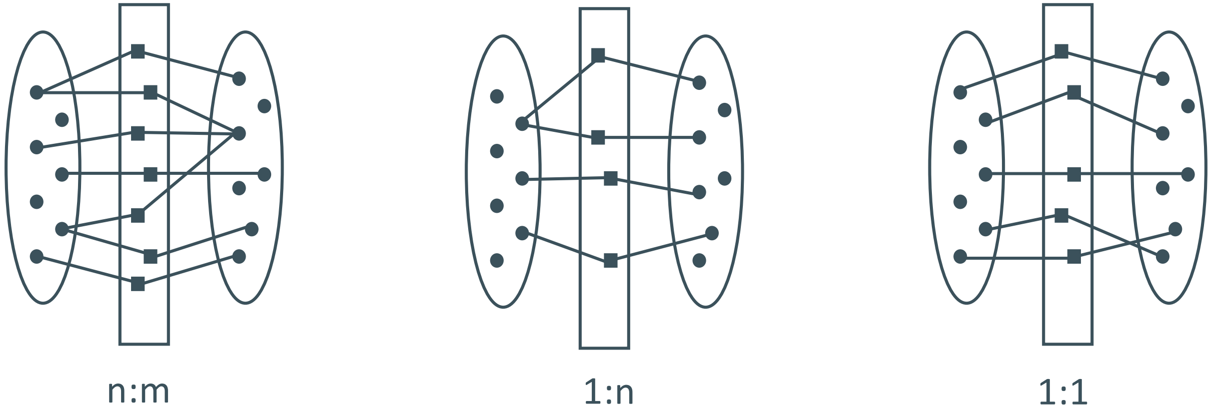


Relationship types



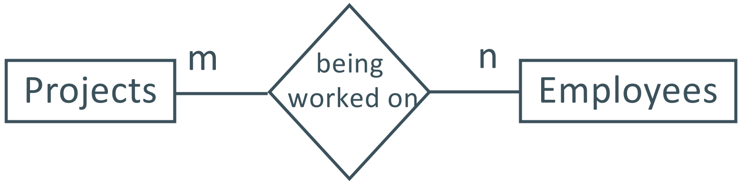
$R \subseteq E_1 \times E_2$

- Objects can be related to one another:
  - Binary (or even ternary, …) Relationships associate two (or three) classes and their respective objects
  - In general: n-ary relations
- Functionality specifications define restrictions on how respective objects are associated (m:n is default if nothing is specified)



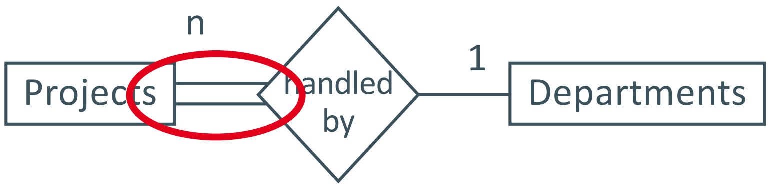
Relationship types: Example

- Example: Employees work in projects. An employee can work on many projects. A project can be worked on by many employees.



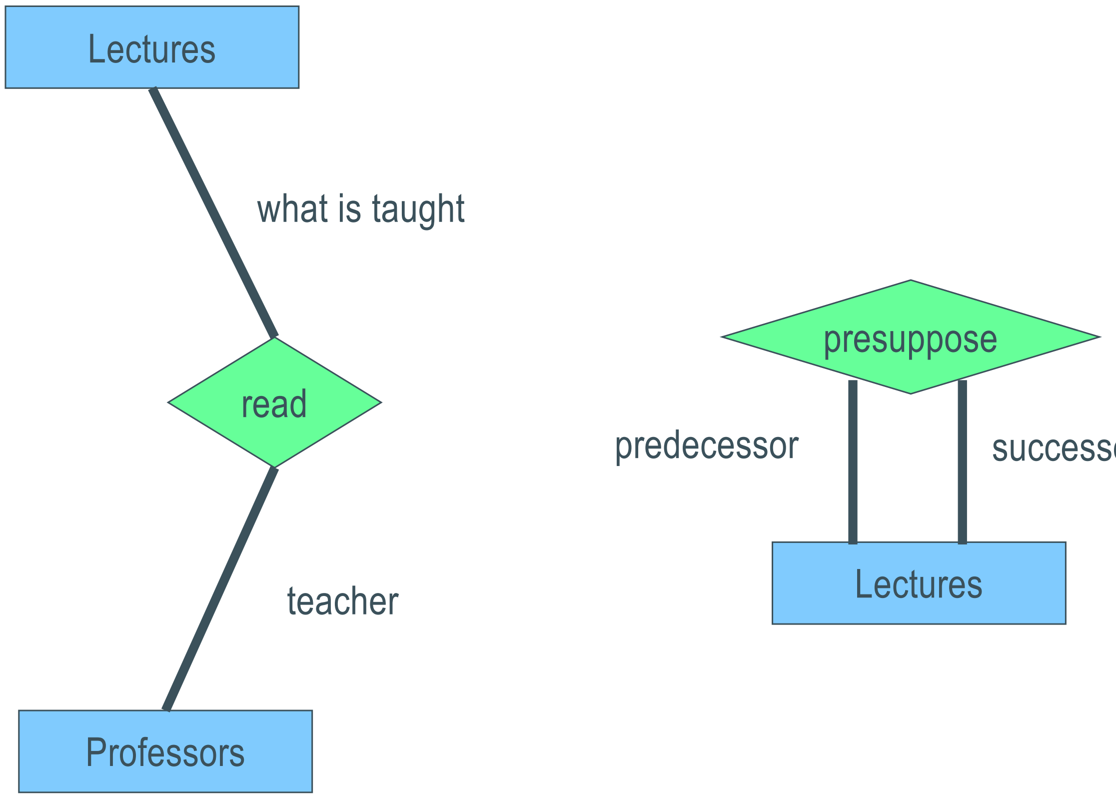
- How to specify that **for every entity of an entity type there is an association**?

- Example: Projects are handled by a single department. Every project must be associated with a department (and a department can handle multiple projects).



Roles

- Names attached to edges attached to relationship types



Teaching goal 1:

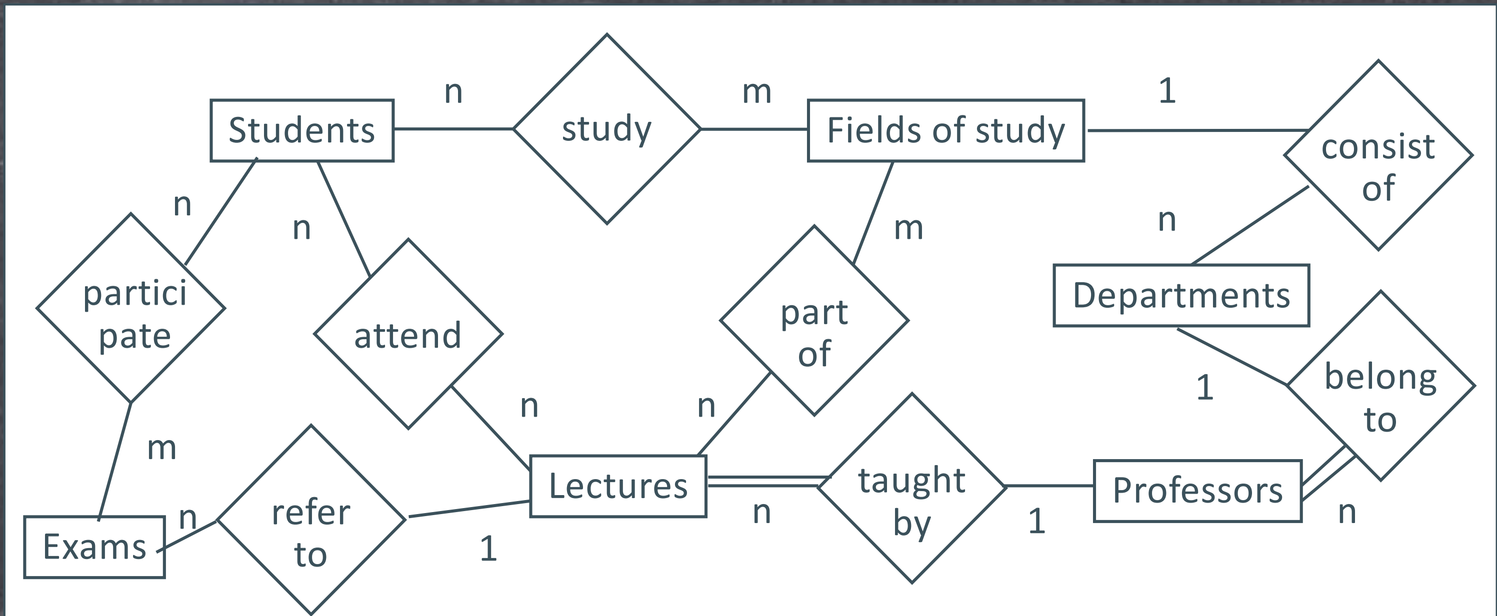
- Given: "Requirements definition"

  - A university offers various lectures that are part of several subjects.
  - These lectures are held by exactly one lecturer.
  - Each lecturer is a member of exactly one department.
  - A department has several subjects.
  - The lectures are attended by students who have enrolled in one or more subjects.
  - Several exams are offered for each lecture, which are taken by the students.

- Find: Suitable ER Diagram

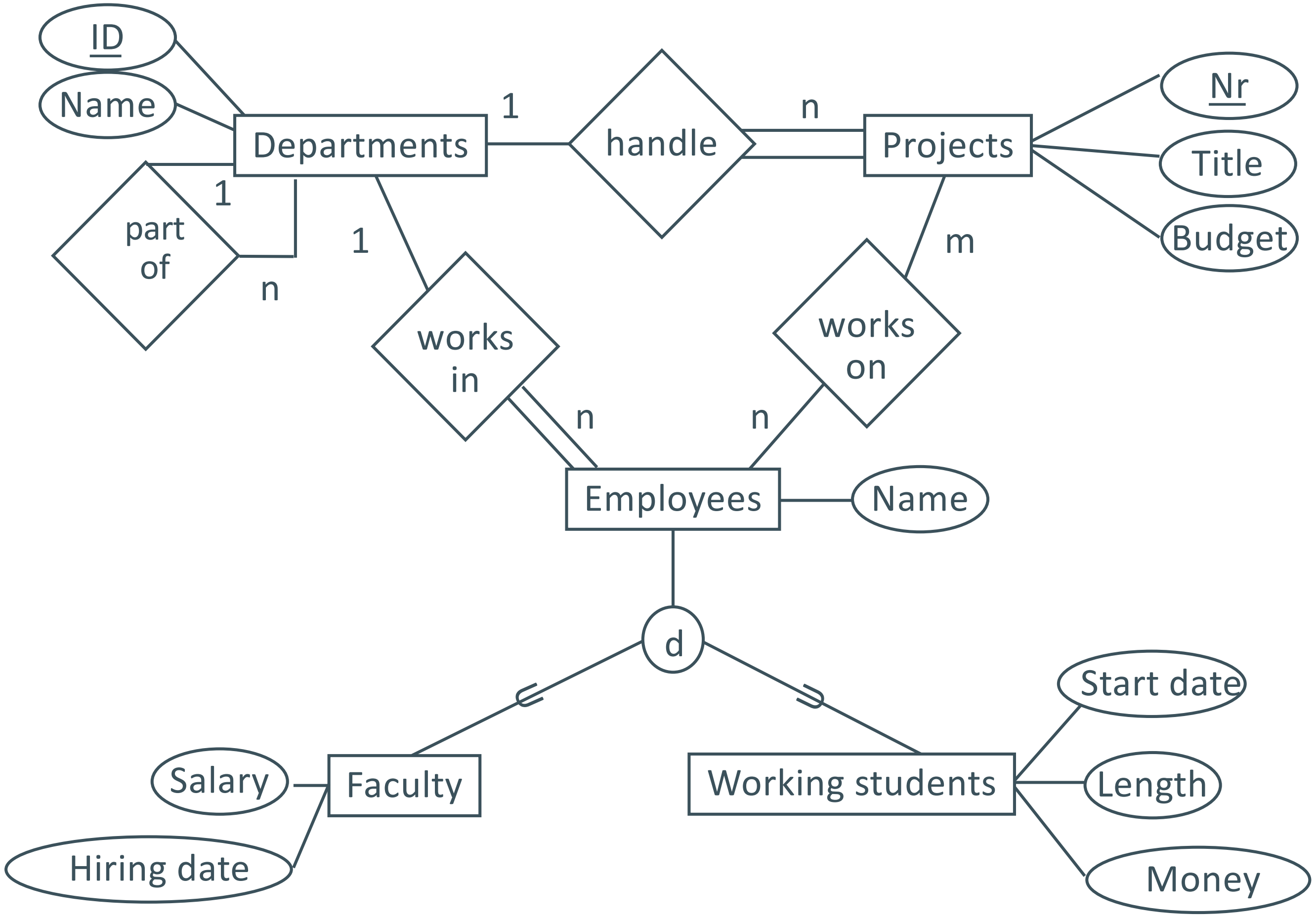
Teaching goal 1:

- Given: "Requirements definition"



- Solution: Suitable ER Diagram

Teaching goal 2: Explain ER Diagram



Identification and keys

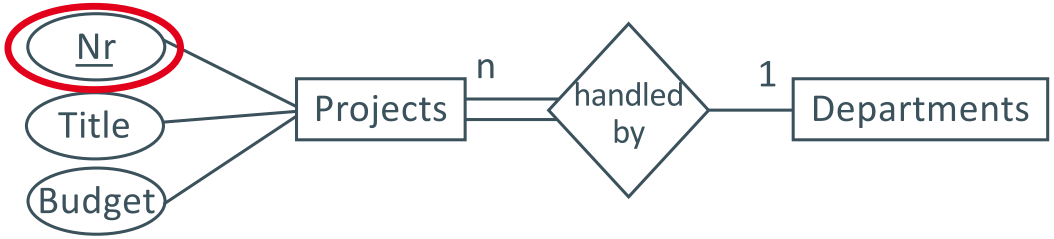
- Keys are attributes (or combinations of attributes) such that in a certain class there exists just one object with a certain value (or certain combination of values)
- Keys can be used to identify entities
- There can be multiple candidates for keys (primary keys, secondary keys, ...)

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

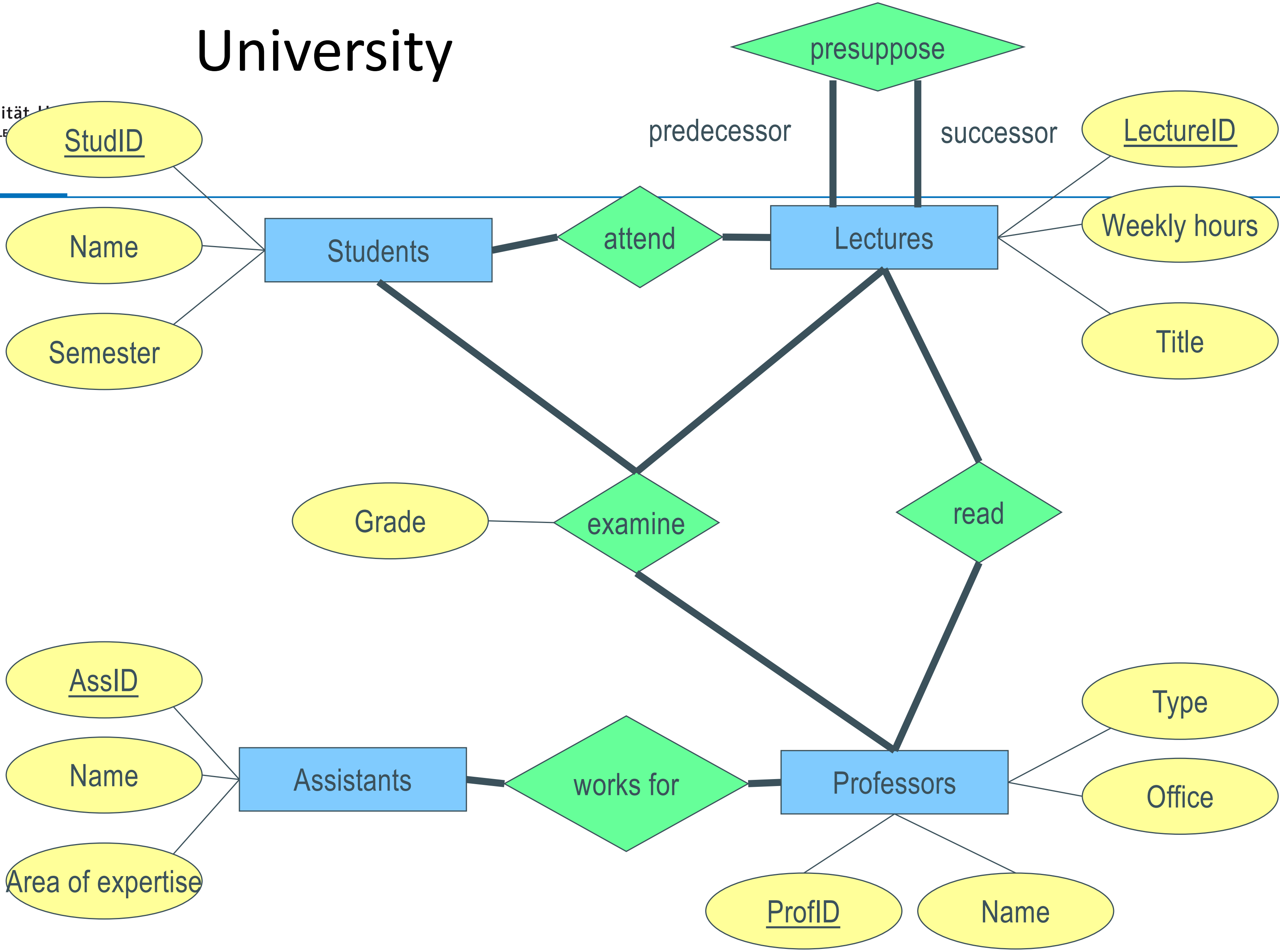
CHAI
Humanities-Centered AI

Identification and keys

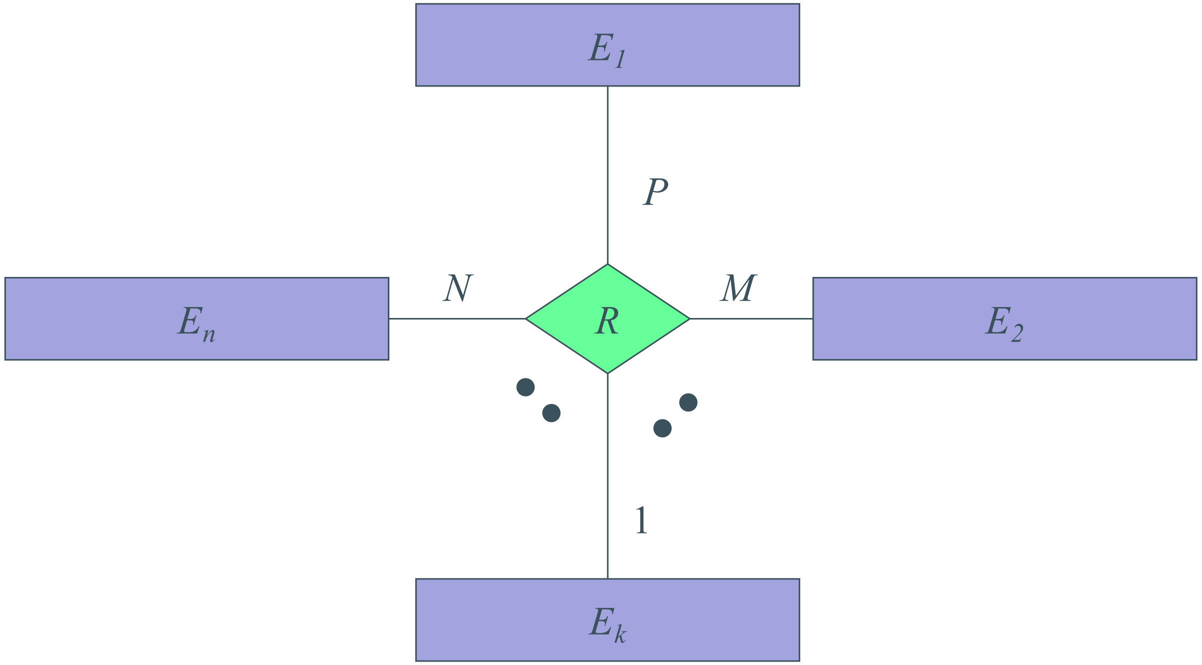
Example: Projects can be uniquely identified with a number



University

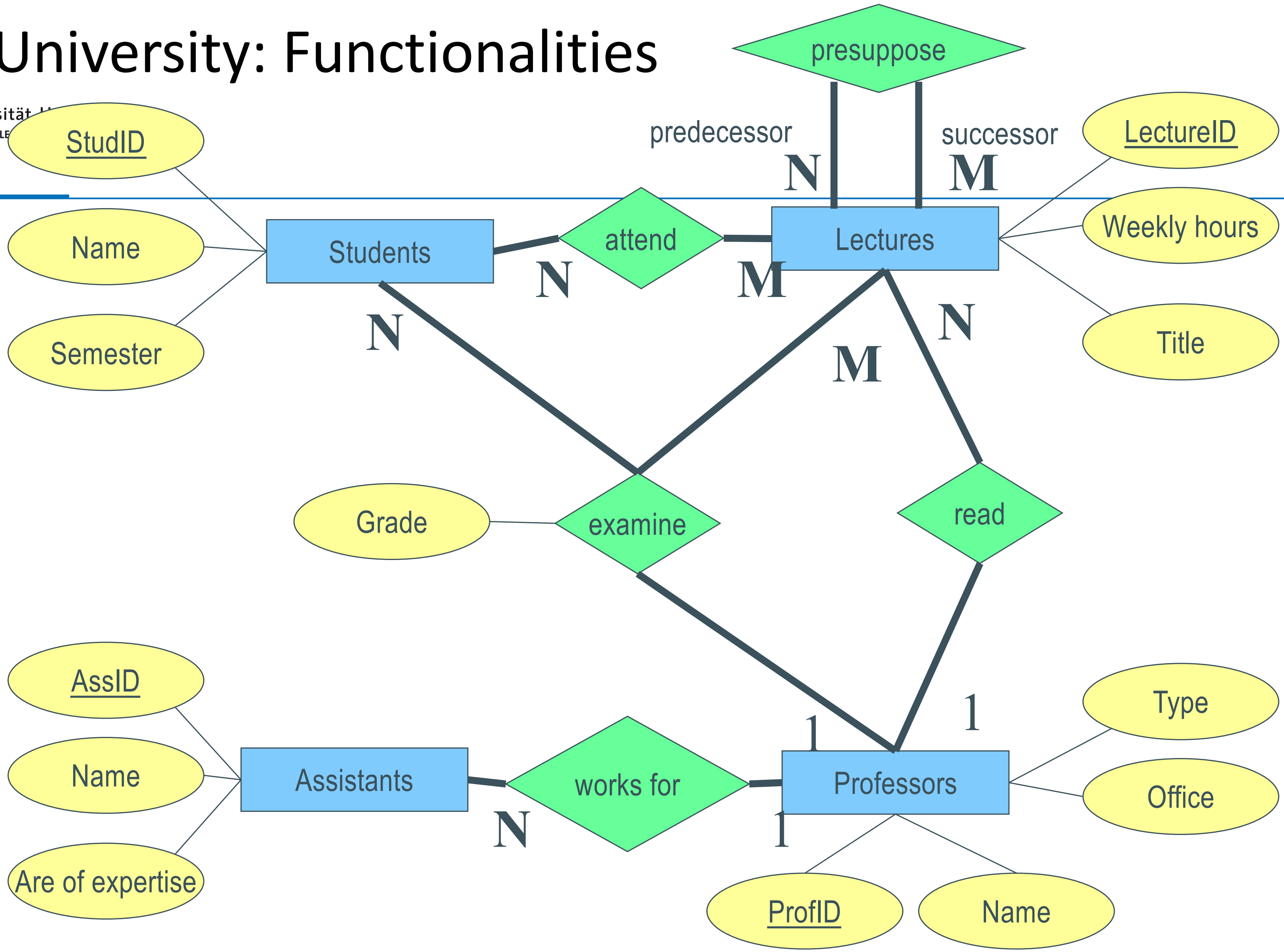

presuppose
predecessor
successor
StudID
Name
Semester
Students
attend
Lectures
LectureID
Weekly hours
Title
Grade
examine
read
AssID
Name
Area of expertise
Assistants
works for
Professors
Type
Office
ProfID
Name

Functionalities with *n*-ary relationship types

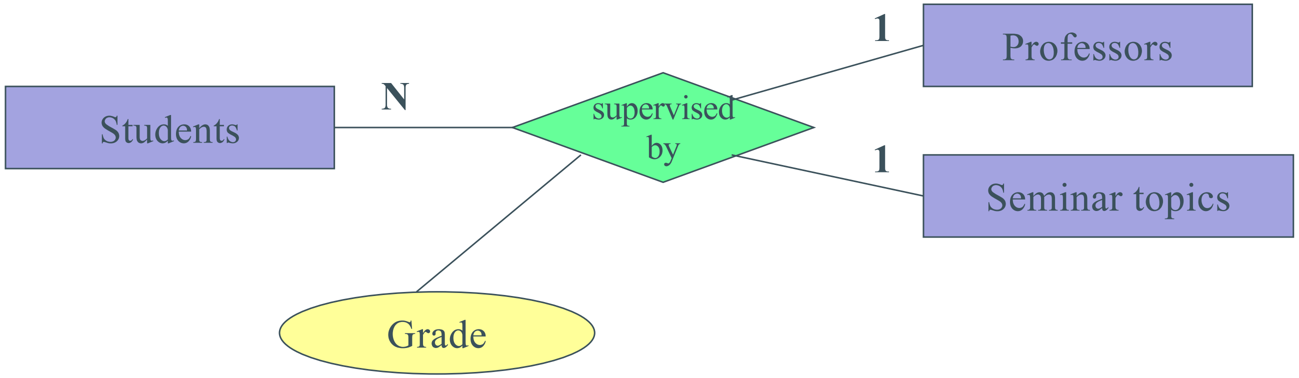


$$R : E_1 \times \ldots \times E_{k-1} \times E_{k+1} \times \ldots \times E_n \longrightarrow E_k$$

University: Functionalities



More complex example: *supervised by*



Textual notation:

supervised by : Professors x Students $\longrightarrow$ Seminar topics

supervised by : Seminar topics x Students $\longrightarrow$ Professors

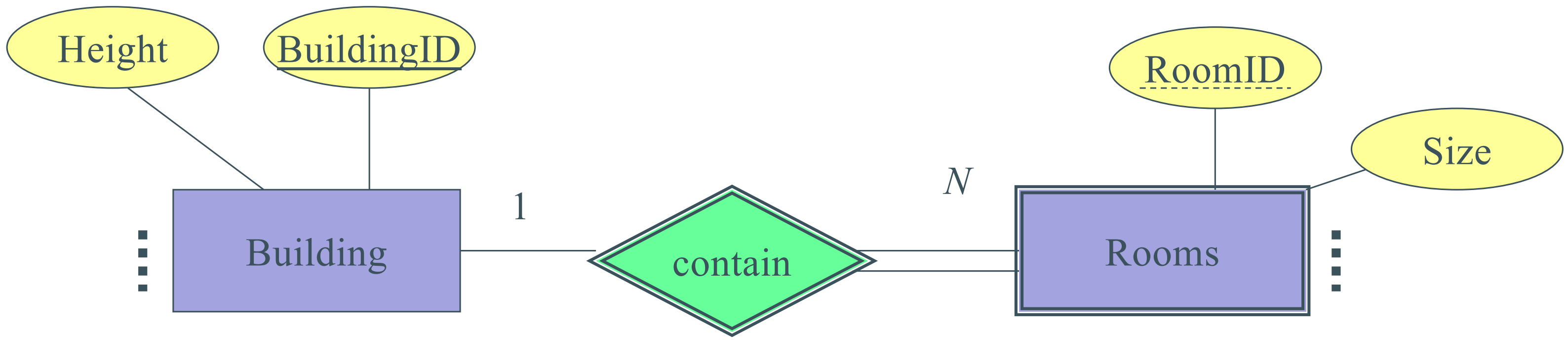
Enforced consistency conditions

1. Students may only “complete” one seminar topic with the same professor (so that a broad spectrum is covered).
2. Students may only work on the same seminar topic once—they may not repeat a seminar topic that has already been assigned by other professors.

The following database states are still possible:

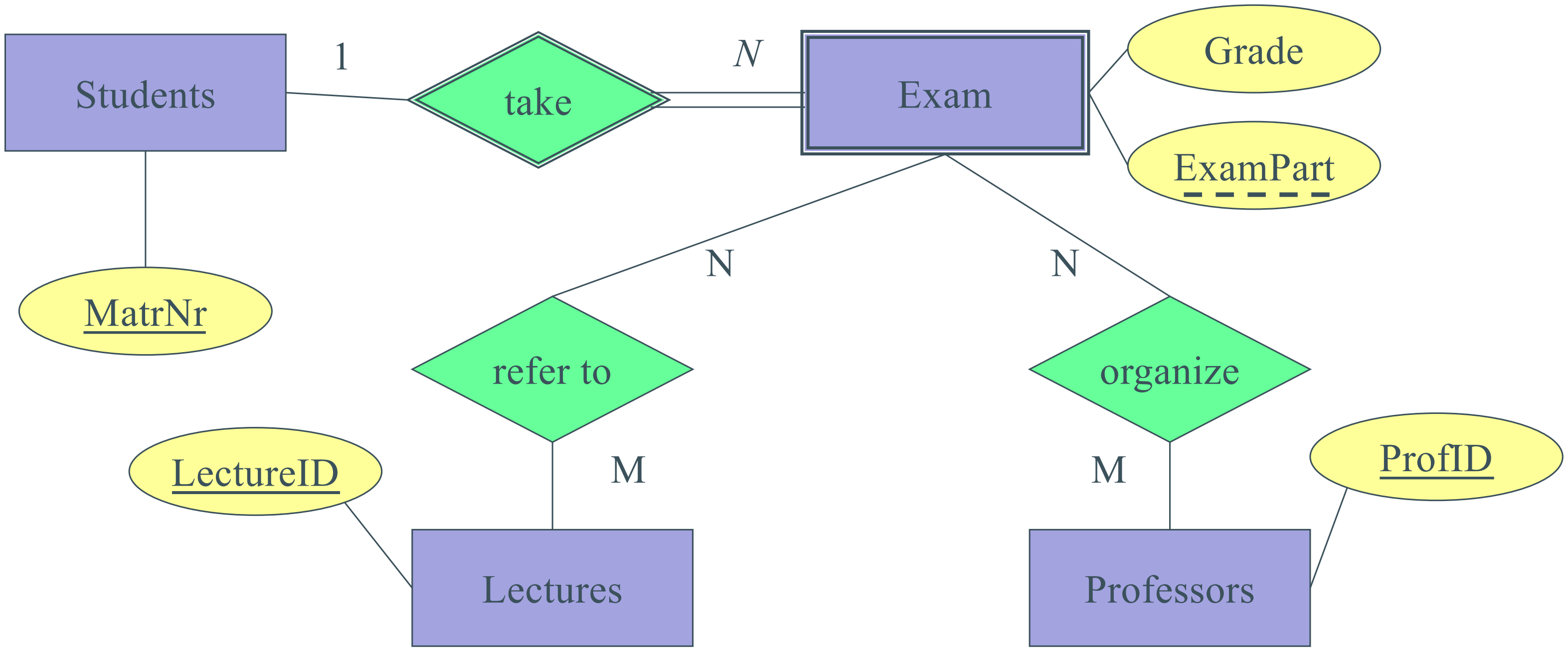
- Professors can “reuse” the same seminar topic—that is, assign the same topic to several students.
- A topic can be assigned by several professors – but to different students.

Weak Entities



- RoomID is only unique within a building.
- The key for Rooms is: BuildingID and RoomID.
- The relationship between “strong” and “weak” types is always 1:N (or 1:1 in rare cases).
- Why can't this be an N:M relationship?

Exam as weak entity type



Multiple examiners in one exam

Multiple lectures are covered in one exam

Summary and outlook

1. **Introduction**
   - Models and abstraction
   - Database systems
   - Applications and data independence
2. **Entity-relationship model (ER model)**
3. **Relational data model (RDM)**
   - From ER to RDM
4. **Relational design theory**
   - Relational algebra as a query language
   - Functional dependencies
   - Normal forms
5. **Structured Query Language (SQL)**
   - Data definition
   - Query specification
   - Data manipulation
6. **Query processing**
   - Architecture
   - Indexing
   - Query plans, optimization
7. **Transactions**
   - Transaction processing, schedules, locks
   - Recovery
8. **Other types of databases**
   - Vector databases
   - Data stream query languages