SCIENCE
PASSION
TECHNOLOGY

Database Systems
02 Conceptual Design

Matthias Boehm

Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMVIT endowed chair for Data Management

Last update: Mar 11, 2019

Announcements/Org

- **Feedback so far**
  - **#1 Video Recording** (5): Record and upload lectures (English, repetition, flexibility, room) ➔ **Expected start: Mar 18**
  - **#2 Questions** (1): Repeat questions for everybody in the room

- **Update SIGMOD Programming Contest 2019** (1)

  Extracurricular Activity

  - Task announced Mar 5: **Radix partition/sort** (10B+90B) http://sigmod19contest.itu.dk/task.shtml
  - **Prizes: $7.000** (winner) / **$3.000** (first runner-up), by Amazon Web Services
  - **Deadline: Apr 25, 2019**

[Viktor Leis, Alfons Kemper, Thomas Neumann: The adaptive radix tree: ARTful indexing for main-memory databases. **ICDE 2013**]

[Matthias Boehm, Benjamin Schlegel, Peter Benjamin Volk, Ulrike Fischer, Dirk Habich, Wolfgang Lehner: Efficient In-Memory Indexing with Generalized Prefix Trees. **BTW 2011**]

Agenda

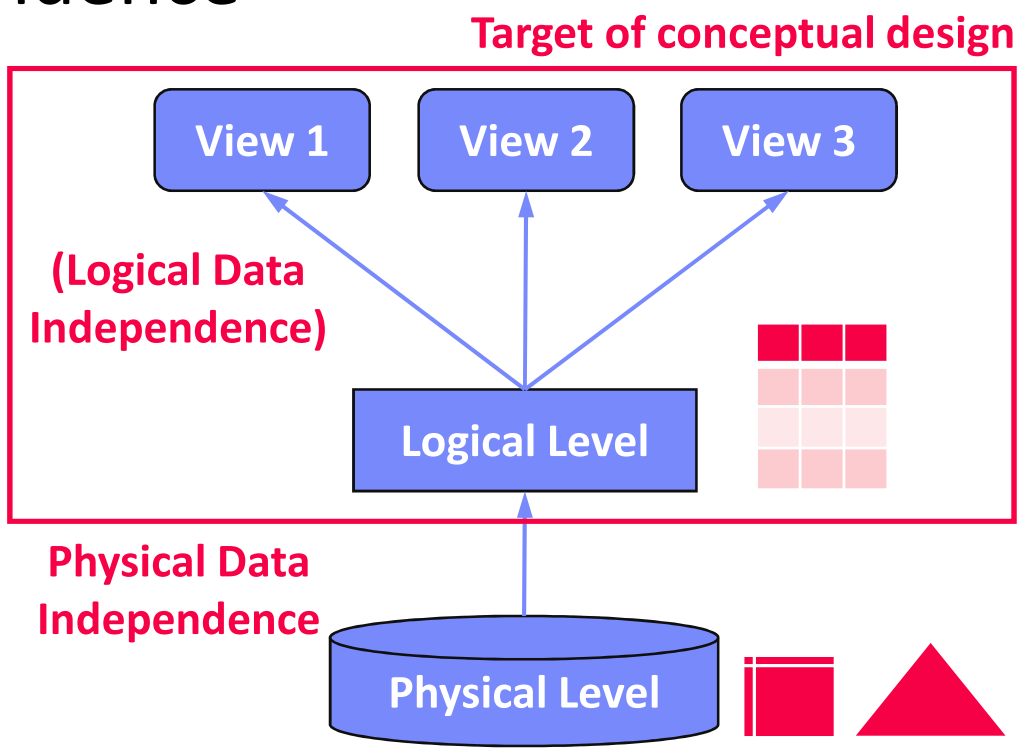
- **DB Design Lifecycle**
- **ER Model and Diagrams**
- **Exercise 01 – Data Modeling**

[**Credit:** Alfons Kemper, André Eickler: Datenbanksysteme - Eine Einführung, 10. Auflage. De Gruyter Studium, de Gruyter Oldenbourg 2015, ISBN 978-3-11-044375-2, pp. 1-879]

DB Design Lifecycle

Recap: Data Independence

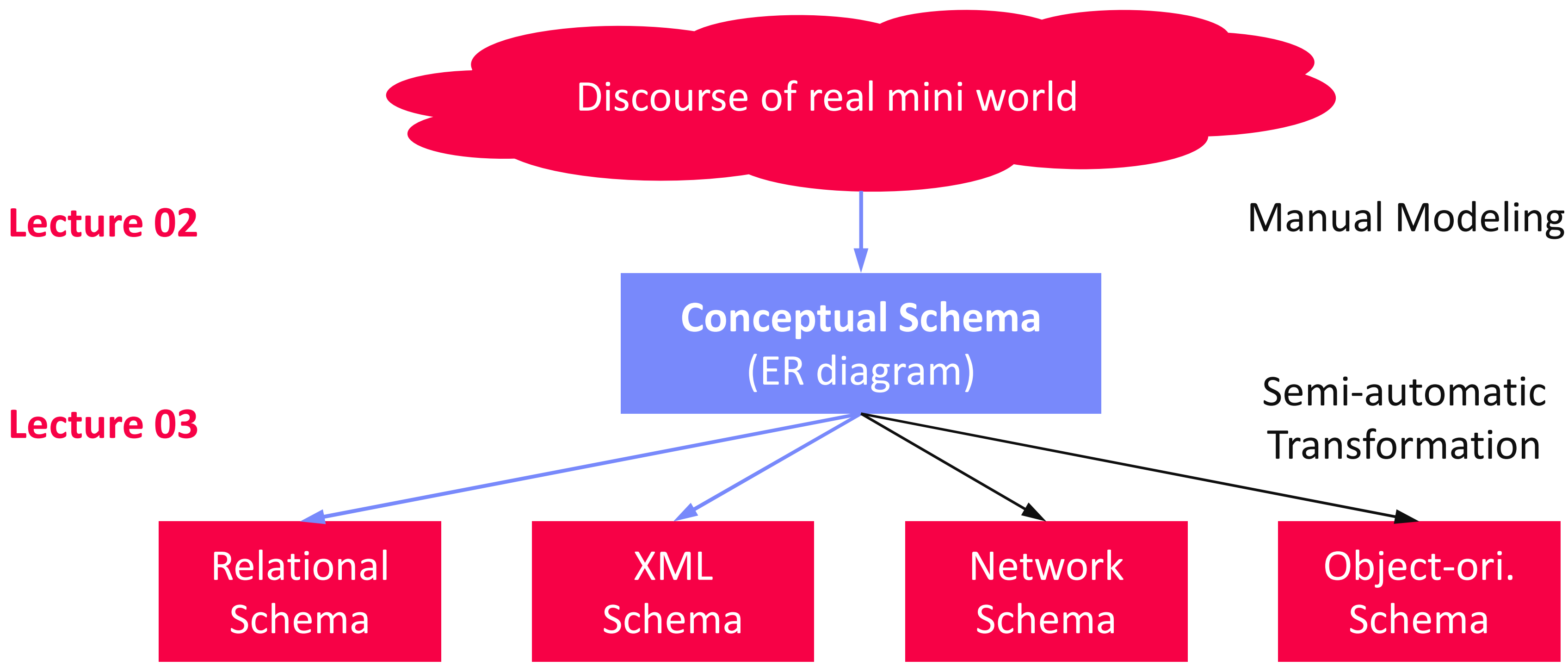
- **Three Layer ANSI-SPARC Architecture**
  - External schemas (external level)
  - Conceptual schema (logical level)
  - Internal schema (physical level)



- **Types of Data Independence**
  - **Logical data independence** (external views and applications independent of logical data model)
  - **Physical data independence** (logical data model independent of underlying data organization)

Data Modeling

- **Data Model**
  - Concepts for describing data objects and their relationships (meta model)
  - **Schema:** Description (structure, semantics) of specific data collection



Data Models

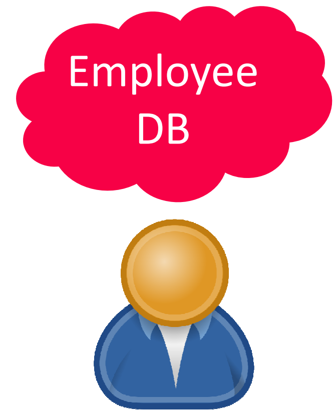
- **Conceptual Data Models**
  - **Entity-Relationship Model (ERM)**, focus on data, ~1975
  - Unified Modeling Language (UML), focus on data and behavior, ~1990

- **Logical Data Models**
  - **Relational**
  - Key-Value
  - Graph
  - Document (XML, JSON)
  - Matrix/Tensor
  - Object-oriented
  - Network
  - Hierarchical

Partly covered in part B (Key-Value, Graph, Document, Matrix/Tensor)

Mostly obsolete (Network, Hierarchical)

Phases of the DB Design Lifecycle



- **#1 Requirements engineering**
  - Collect and analyze data and application requirements
  - → **Specification documents**
- **#2 Conceptual Design** (this lecture)
  - Model data semantics and structure, independent of logical data model
  - → **ER model / diagram**
- **#3 Logical Design** (next lecture)
  - Model data with implementation primitives of concrete data model
  - → **e.g., relational schema** + integrity constraints, views, permissions, etc
- **#4 Physical Design**
  - Model **user-level data organization** in a specific DBMS (and data model)
  - Account for deployment environment and performance requirements

Relevance of Conceptual Design in Practice

- **Analogy ERM-UML**
  - **Model-driven development** (self-documenting, but quickly outdated)
  - **But:** Once data is loaded, data model and schema harder to change

- **Observation: Full-fledged ER modeling rarely used in practice**
  - Often the logical schema (relational schema) is directly created, maintained and used for documentation
  - **Reasons:** redundancy, indirection, single target (relational)
  - Simplified ER modeling used for brainstorming and early ideas

- **Goals**
  - **Understanding of proper database design** from conceptual to physical schema
  - ER modeling as a helpful **tool in database design**
  - Schema transformation and normalization as blueprint for **good designs**

Entity-Relationship (ER) Model and Diagrams

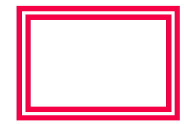
[Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst. 1(1) 1976**]

[Peter P. Chen: The Entity-Relationship Model: Toward a Unified View of Data. **VLDB 1975**]

ER Diagram Components (Chen Notation)

- **Entity Type** (noun)
  - Entities are objects of the real world
  - An entity type (or entity set) represents a collection of entities

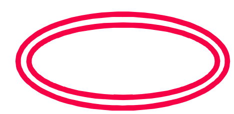
Weak entities



- **Relationship Type** (verb)
  - Relationships are concrete associations of entities
  - Relationship type (or relationship set) or relationship of entity types

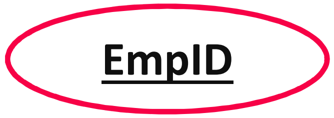
- **Attribute**
  - Entities or relationships are characterized by attribute-value pairs
  - Attribute types (or value sets) describe entity and relationship types
  - Extended attributes: composite, multi-valued, derived

Multi-valued attributes

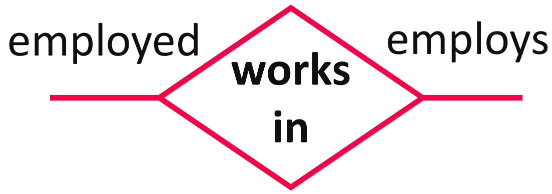


ER Diagram Components (Chen Notation), cont.

- **Keys**
  - Attributes that uniquely identify an entity
  - Every entity type must have such a key
  - Natural or surrogate (artificial) keys

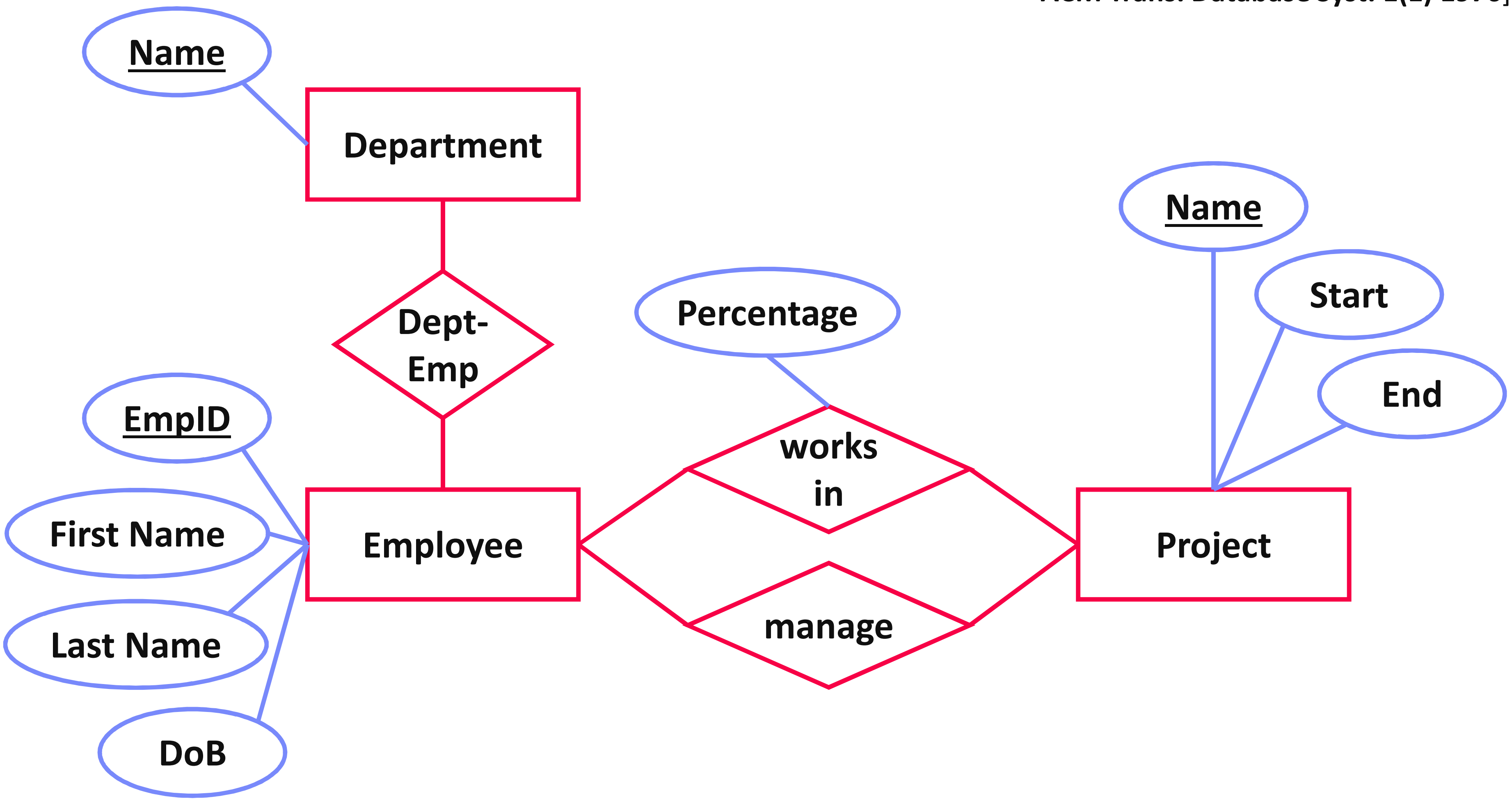


- **Role**
  - Optional description of relationship types
  - Useful for recursive relationships



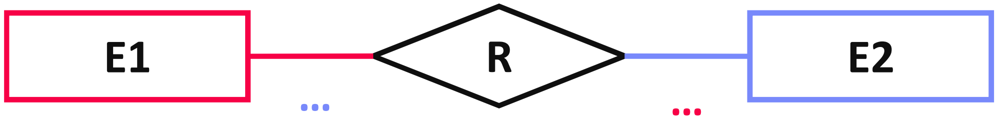
An EmployeeDB Example

[Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst. 1(1) 1976**]



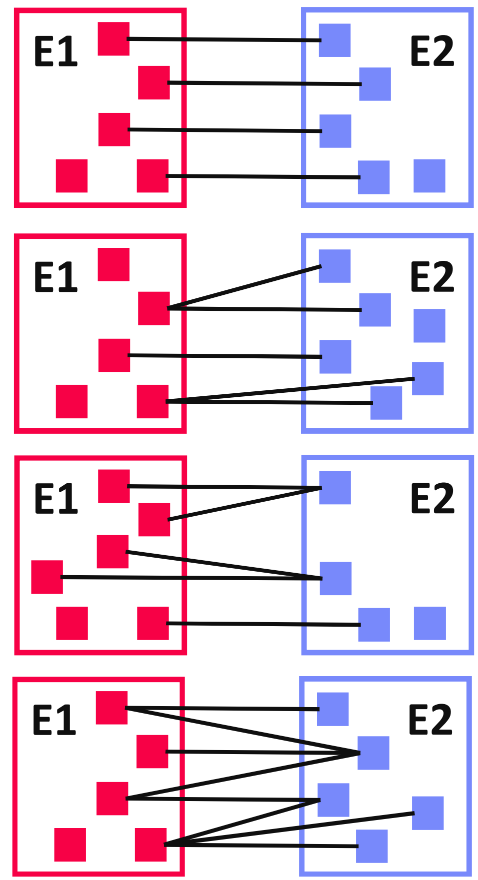
Multiplicity (Mapping Cardinalities)

1 .. [0,1]
N … [0,1,N]



$R \subseteq E1 \times E2$

- **1:1 (one-to-one)**
  - Each e1 relates to at most one e2
  - Each e2 relates to at most one e1
- **1:N (one-to-many)**
  - Each e1 relates to many e2 (0,1,…N)
  - Each e2 relates to at most one e1
- **N:1 (many-to-one)**
  - Symmetric to 1:N
- **M:N (many-to-many)**
  - Each e1 relates to many e2 (0,1,…N)
  - Each e2 related to many e1 (0,1,…N)

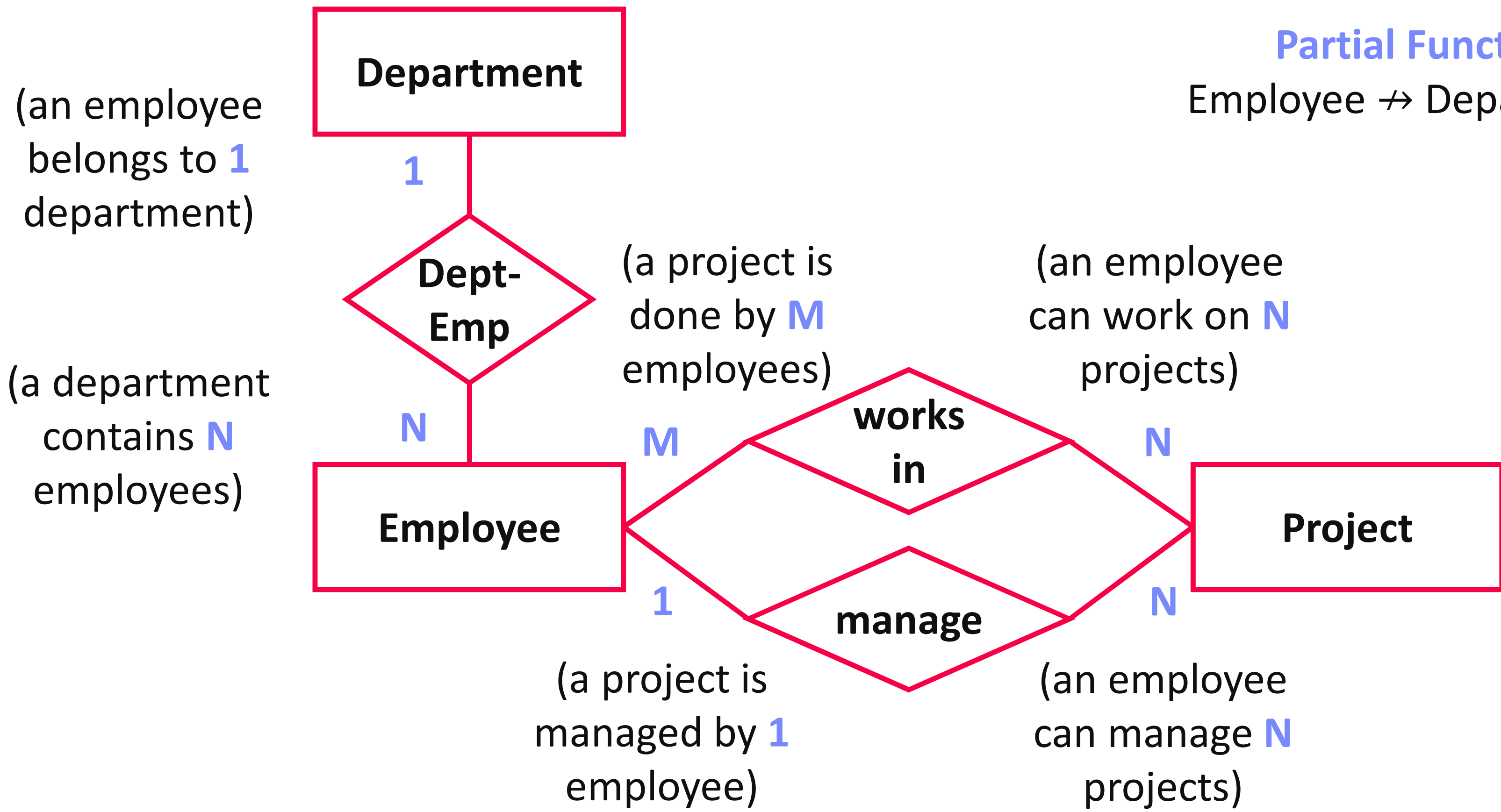


An EmployeeDB Example, cont.

[Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst. 1(1) 1976**]

Partial Function

Employee ↛ Department



Multiplicity in Modified Chen (MC) Notation

- **Extension:** C ("choice"/"can") to model 0 or 1, while 1 means exactly 1 and M means at least 1.

4 alternatives (1, C, M, CM)
→ 2^4 = 16 combinations
(symmetric combinations omitted)

- **1:1** – [1] to [1]
- **1:C** – [1] to [0 or 1]
- **1:M** – [1] to [at least 1]
- **1:MC** – [1] to [arbitrary many]

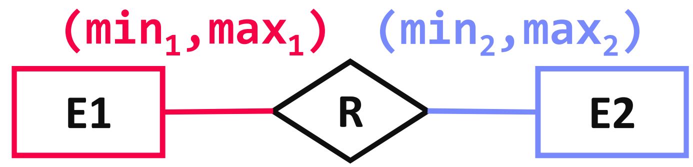
- **C:C** – [0 or 1] to [0 or 1] → **see 1:1 in Chen**
- **C:M** – [0 or 1] to [at least 1]
- **C:MC** – [0 or 1] to [arbitrary many] → **see 1:N in Chen**

- **M:M** – [at least 1] to [at least 1]
- **M:MC** – [at least 1] to [arbitrary many]

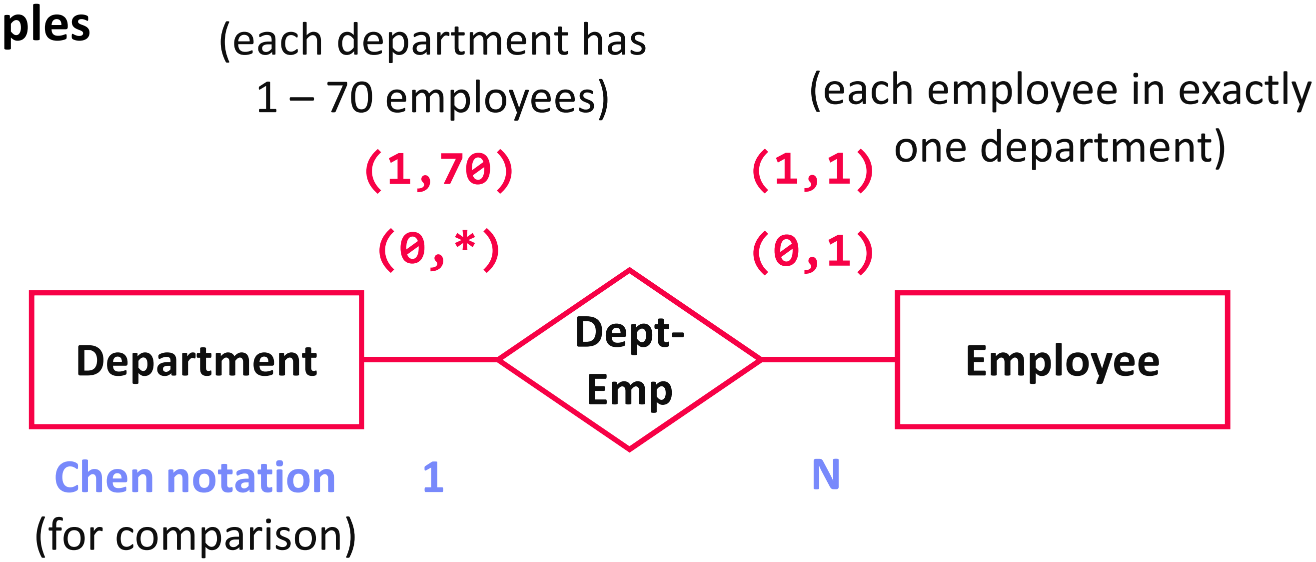
- **MC:MC** – [arbitrary many] to [arbitrary many] → **see M:N in Chen**

(min,max)-Notation

- **Alternative Cardinality Notation**
  - **Indicate concrete min/max constraints**
    (each entity is part of at least/at most x relationships)
  - Chen and (min,max) notation generally incomparable
  - **Wildcard *** indicates arbitrary many (i.e., N)



- **Examples**



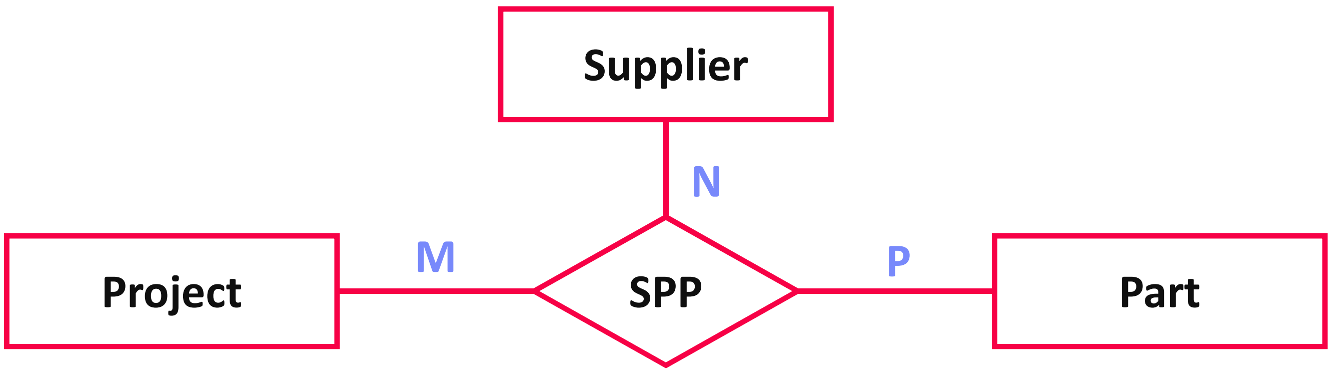
Weak Entity Types

- **Existence Dependencies**
  - Entities **E2** whose existence depends on the other entities **E1**
  - Visualized as a special rectangle with double border
  - Primary key is contains primary key of **E1**
  - Relationship between strong and weak entity types **1:N** (sometimes **1:1**)

- **Examples**
  - Dependents of an employee (spouse, children)
  - Rooms of a building

N-ary Relationships

- **Use of n-ary relationships**
  - Relationship type among multiple entity types
  - N-ary relationship can be converted to binary relationships
  - Design choice: **simplicity** and **consistency constraints**

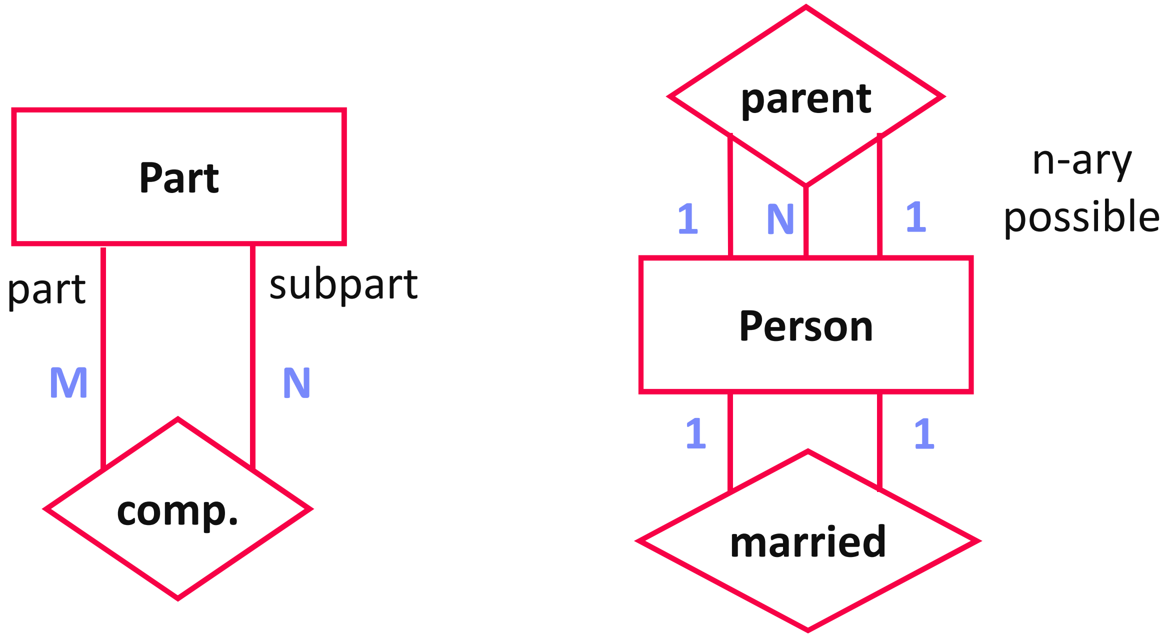


- **Multiplicity**
  - 1 Project and 1 Supplier → supply P parts
  - 1 Project and 1 Part → supplied by N suppliers (**1 instead of N?**)
  - 1 Supplier and 1 Part → supply for M projects

Recursive Relationships

- **Definition**
  - Recursive relationships are relations between entities of the same type
  - Use roles to differentiate cardinalities

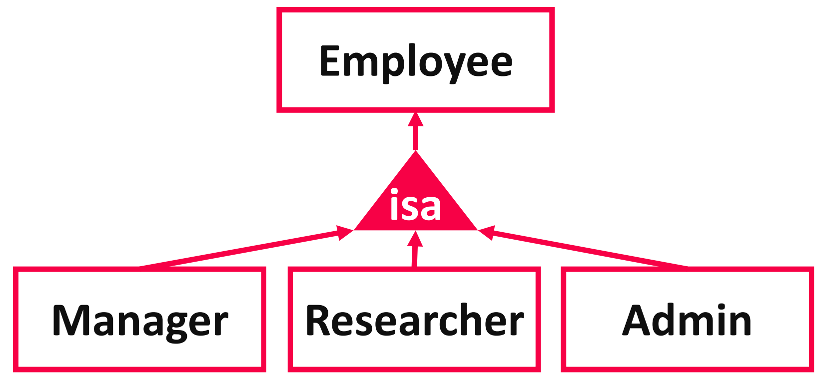
- **Examples**



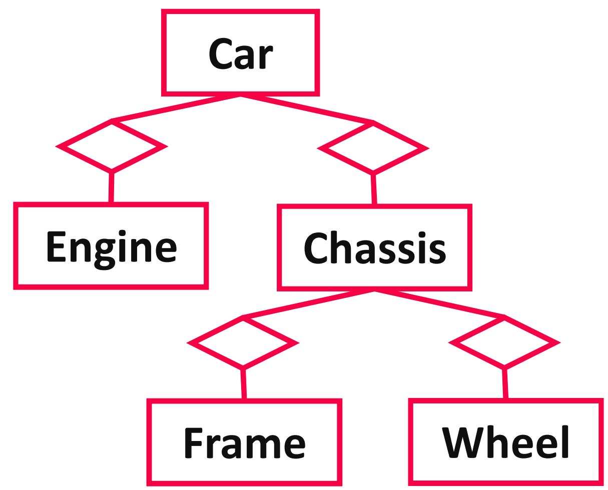
- **Beware of [at least 1] constraints in recursive relationships** (e.g., (min,max)-notation, or MC notation)

Specialization and Aggregation

- **Specialization via Subclasses**
  - Tree of specialized entity types (no multi-inheritance)
  - Graphical symbol: triangle (or hexagon, or subset)
  - Each entity of subclass is entity of superclass, but not vice versa

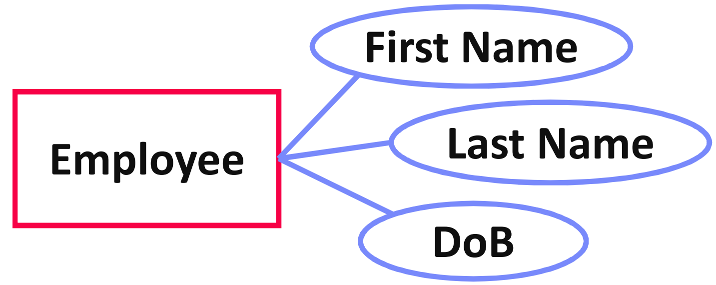


- **Aggregation** (is not specialization)
  - **#1:** Recursive relationship types, or
  - **#2:** Explicit tree of entity and relationship types
  - Design choice: number of types known and finite, and heterogeneous attributes

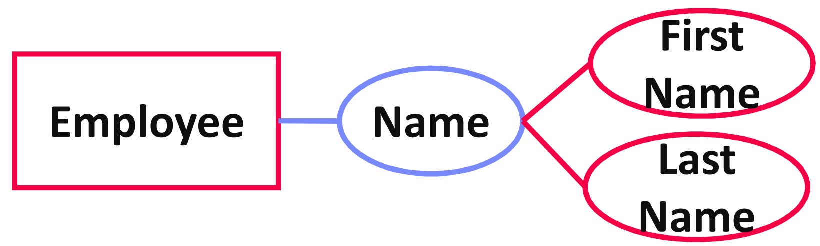


Types of Attributes

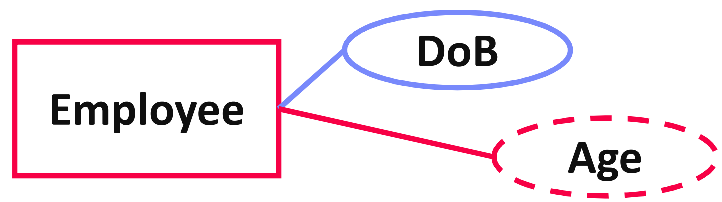
- **Atomic Attributes**
  - Basic, single-valued attributes



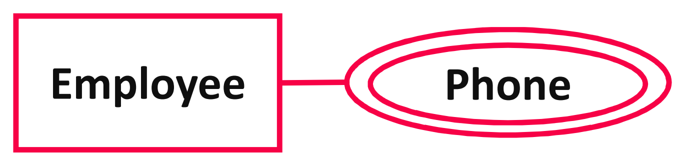
- **Composite Attributes**
  - Attributes as structured data types
  - Can be represented as a hierarchy



- **Derived Attributes**
  - Attributes derived from other data
  - Examples: Number of employees in dep, employee age, employee yearly salary

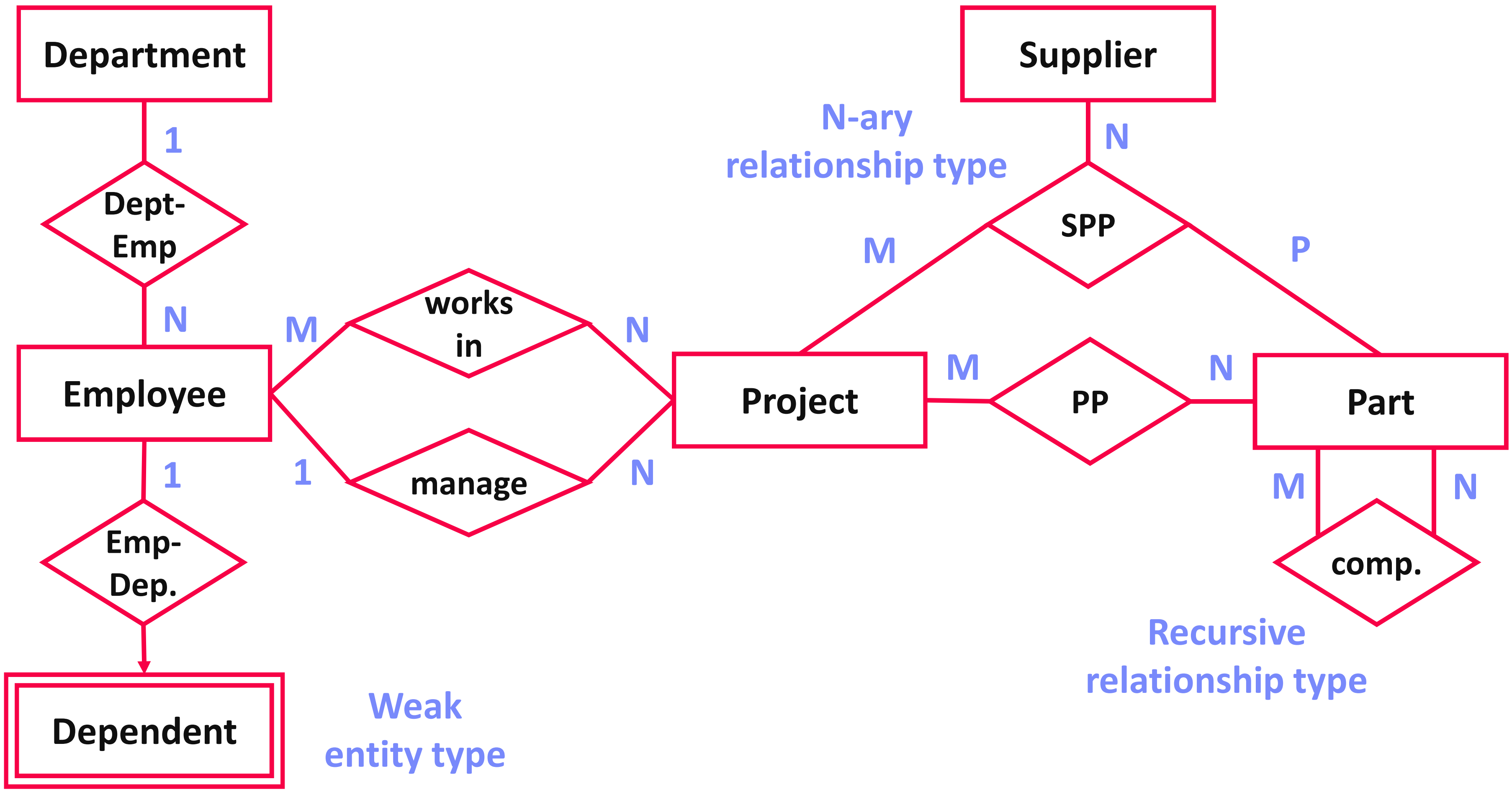


- **Multi-valued Attributes**
  - Attributes with list of homogeneous entries



An EmployeeDB Example, cont.

[Peter P. Chen: The Entity-Relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst. 1(1) 1976**]

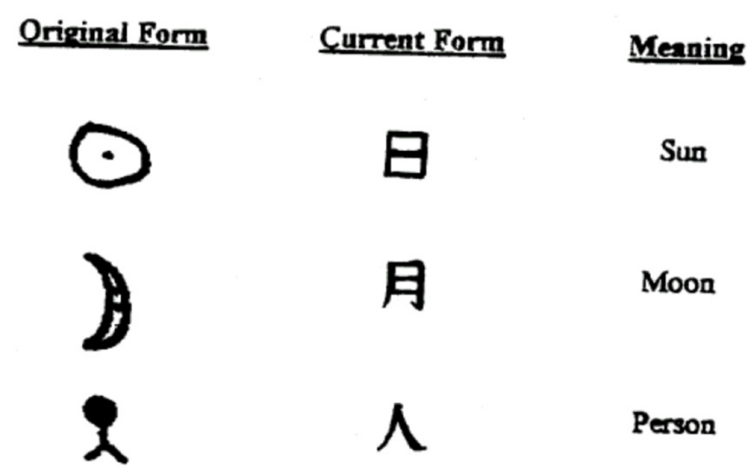


Excursus: Influence of Chinese Characters?

"What does the Chinese character construction principles have to do with ER modeling? The answer is: both Chinese characters and the ER model are trying to model the world – trying to use graphics to represent the entities in the real world. […]"

[Peter Pin-Shan Chen: Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. **Software Pioneers 2002**]

- **Chinese characters representing real-world entities**



- **Composition of two Chinese characters**

日 (sun) + 月 (moon) = 明 (Bright/ Brightness by light)

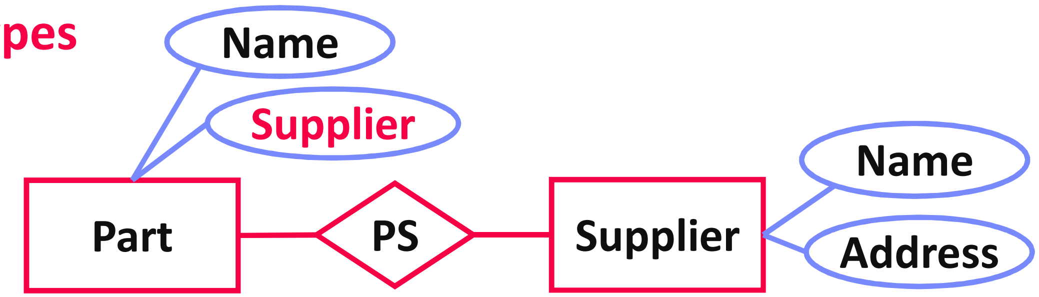
Design Decisions

Avoid redundancy
Avoid unnecessary complexity

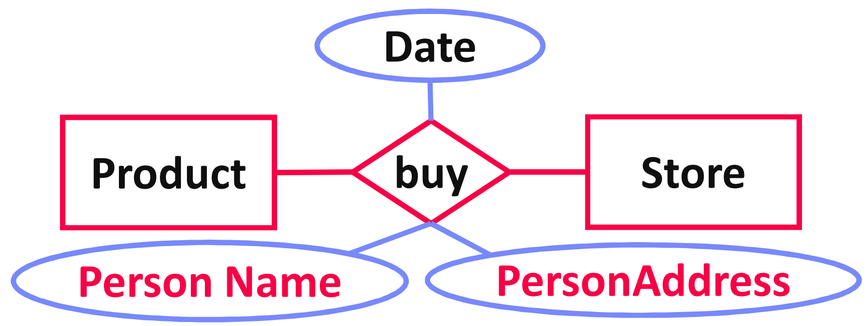
- **Meta-Level:**
  - Which notations to use (Chen, modified Chen, (min,max)-notation)?
- **Entities**
  - What are the entity types (entity vs relationship vs attribute)?
  - What are the attributes of each entity type?
  - What are key attributes (one or many)?
  - What are weak entities (with partial keys)?
- **Relationships**
  - What are the relationship types between entities (binary, n-ary)?
  - What are the attributes of each relationship type?
  - What are the cardinalities?
- **Attributes**
  - What are composite, multi-valued, or derived attributes?

Design Decisions – Examples of Poor Choices

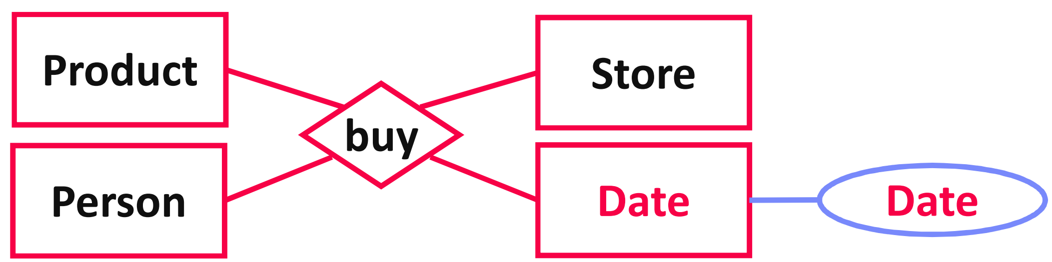
- **#1 Overuse of weak entity types**
- **#2 Redundant attributes**
  - **Redundant supplier name** in Part and Supplier



- **#3 Repeated information**
  - **Missing person entity type** → redundancy per purchase



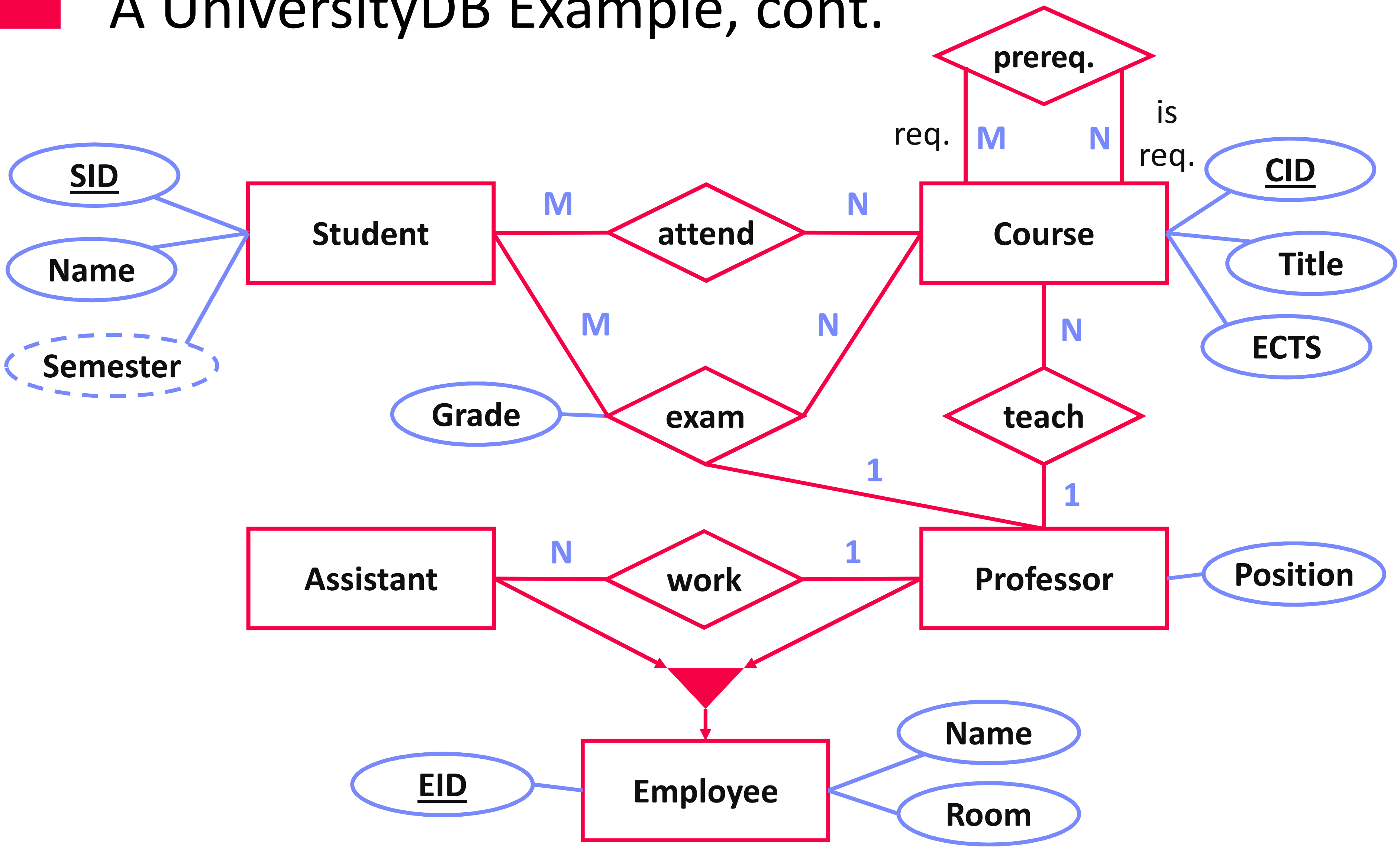
- **#4 Unnecessary Complexity**
  - **Unnecessary entity type Date**
  - Avoid single-attribute entity types unless in many relationships



A UniversityDB Example

- **Discourse of Real Mini World**
  - **Students** (with SID, name, and semester) attend **courses** (CID, title, ECTS), and take graded exams per course
  - **Professors** teach courses, **assistants** work for professors
  - Course may have other course as prerequisites
  - Both professors and assistants are university **employees** (EID, name, and room number); professors also have a position
- **Task: Create an ER diagram in Chen notation**
  - Include entity types, relationship types, attributes, and generalizations
  - Mark primary keys, roles for recursive relationships, and derived attributes

A UniversityDB Example, cont.



Exercise 01 – Data Modeling

Published: **Mar 11, 2019**

Deadline: **Apr 02, 2019**

Exercises: Soccer World Cup 1954-2014

- **Dataset**
  - Public-domain, derived (parsed, cleaned) from **Openfootball Worldcup Dataset**
  - Clone or download your copy from https://github.com/tugraz-isds/datasets.git

- **Exercises**
  - 01 Data modeling (relational schema)
  - 02 Data ingestion and SQL query processing
  - 03 Tuning, query processing, and transaction processing
  - 04 Large-scale data analysis (distributed data ingestions and query processing)

1954_2014_Squads.csv: The Squads file contains tl structure and examples look as follows.

```
#Year, Host_Country, Country, Jersey_Number,
1998,France,Austria,14,FW,Hannes Reinmayr,Stu
2014,Brazil,Germany,1,GK,Manuel Neuer,Bayern
2014,Brazil,Germany,11,FW,Miroslav Klose,Lazi
```

1954_2014_Matches.csv: The Matches file contains and examples look as follows.

```
#Year, Host_Country, Match_ID, Type, Date, Lo
2006,Germany,572,Group A,Wed Jun/14,Signal Id
2010,South Africa,684,Round of 16,Sun Jun/27
2014,Brazil,761,Final,Sun Jul/13 16:00,Estádi
```

1954_2014_Goals.csv: The Goals file contains the g time of the game. It's detailed structure and exam

```
#Year, Host_Country, Match_ID, Team, Player,
2014,Brazil,760,Netherlands,Daley Blind,17
2014,Brazil,760,Netherlands,Georginio Wijnald
2014,Brazil,761,Germany,Mario Götze,113
```

Task 1.1: ER Modeling (12/25 points)

- **ER Diagram in Modified Chen Notation**
  - Discourse: Tournament, Country, Team, Player, Club, Match
  - Create the ER diagram in presentation/data modeling tools
  - Model entity types, relationship types, attribute types, cardinalities, and keys
  - **Note:** The ER diagram allows for alternative modeling choices but you'll loose points for factual mistakes are poor design choices

- **Alternative Cardinalities**
  - Create a list of all relationship types of your ER diagram in (min,max)-notation
  - Use the following format:

  `<entity1> (min,max) – <relationship> – (min,max) <entity2>`

- **Expected result** (for all three subtasks)
  - `DBExercise01_<studentID>.pdf`

Task 1.2: Mapping ER → Relational (8/25 points)

- **Relational Schema**
  - Map your ER diagram into a relational schema (diagram or SQL script)
  - Include relations, typed attributes, primary/foreign key constraints, and NULL constraints

- **Additional Constraints**
  - List of at least 4 additional semantic/domain constraints

Task 1.3: Relational Normalization (5/25 points)

- **3NF Relational Schema**
  - Bring your relational schema into third normal form
  - Explain with reference to specific relations why this schema is in 3NF

- **Requirement for completion**
  - **Submitted on time** (in total at most 7 late days)
  - **13/25 points**

Conclusions and Q&A

- **Summary**
  - DB Design lifecycle from requirements to physical design
  - Entity-Relationship (ER) Model and Diagrams
- **Importance of Good Database Design**
  - Poor database design ➔ **development and maintenance costs**, as well as performance problems
  - Once data is loaded, **schema changes very difficult** (data model, or conceptual and logical schema)
- **Exercise 1: Data Modeling**
  - Published Mar 11, 2019; deadline: Apr 02, 2019
  - Recommendation: start with task 1.1 this week; ask questions in upcoming lectures or on news group
- **Next lecture (Mar 18): 03 Data Models and Normalization**