ENSF 608:
Entity-Relationship (ER) Modelling and Enhanced Entity-Relationship Modelling

Dr. Emily Marasco

Fall 2021

Textbook: Fundamentals of Database Systems, 7th Ed., Elmasri & Navathe

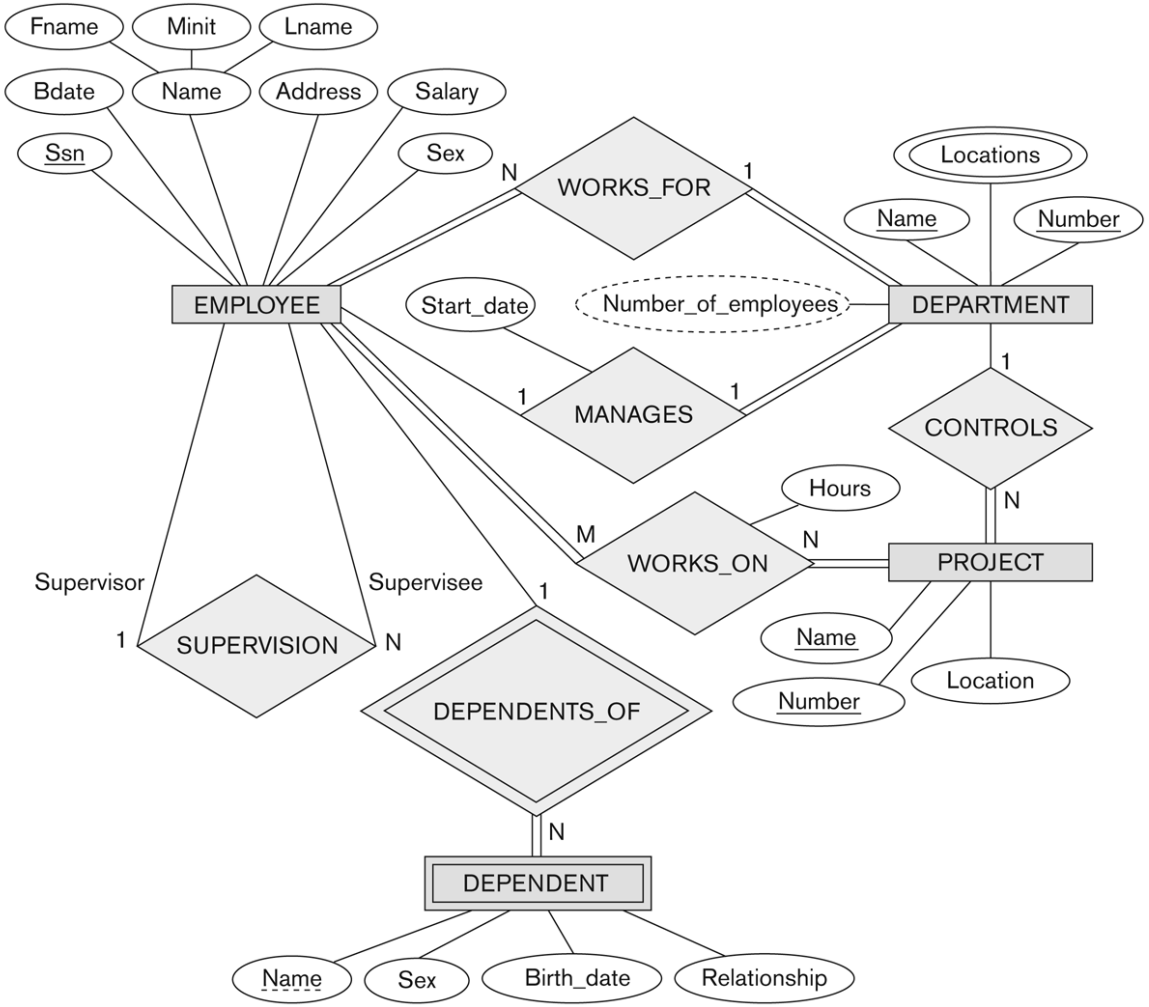
Weak Entity Types (1 of 2)

- An entity that does not have a key attribute and that is identification-dependent on another entity type.
- A weak entity must participate in an identifying relationship type with an owner or identifying entity type
- Entities are identified by the combination of:
  - A partial key of the weak entity type
  - The particular entity they are related to in the identifying relationship type

Weak Entity Types (2 of 2)

- **Example:**
  - A DEPENDENT entity is identified by the dependent's first name, **and** the specific EMPLOYEE with whom the DEPENDENT is related
  - Name of DEPENDENT is the **partial key**
  - DEPENDENT is a **weak entity type**
  - EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

Weak Entity Notation

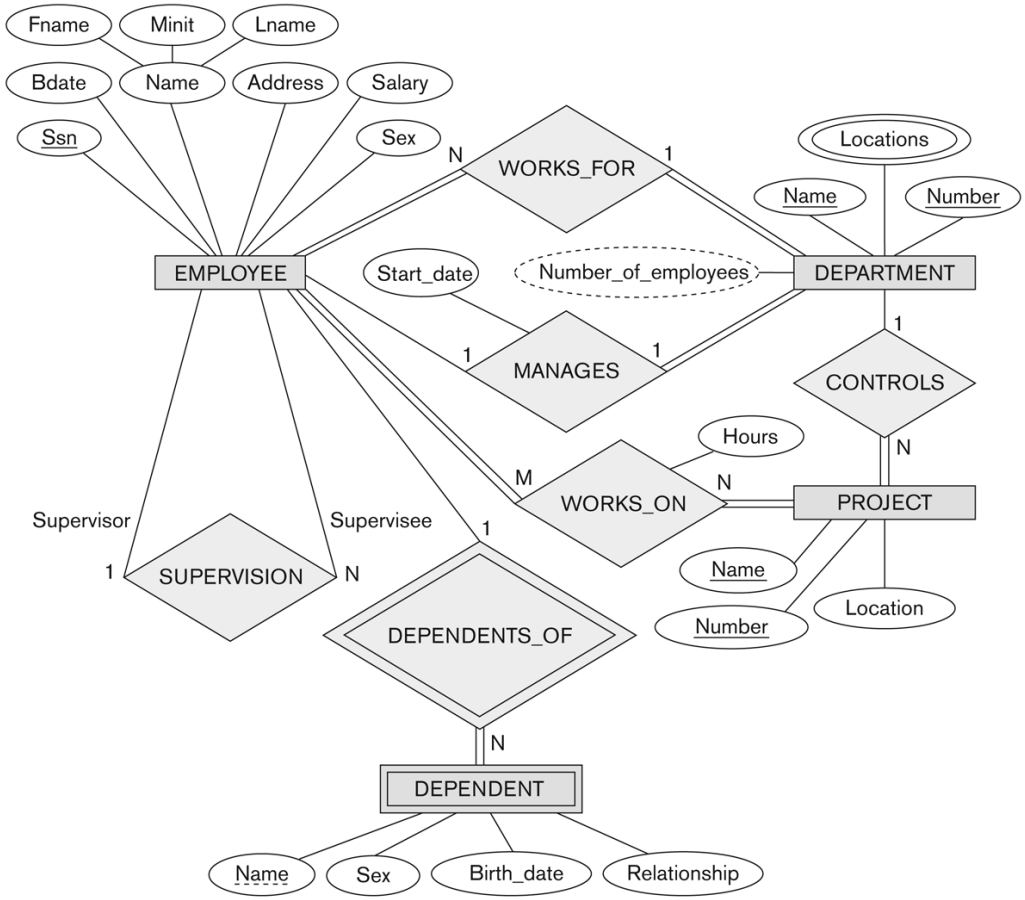


Attributes of Relationship Types

- A relationship type can have attributes:
  - For example, HoursPerWeek of WORKS_ON
  - Its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT.
    - A value of HoursPerWeek depends on a particular (employee, project) combination
  - Most relationship attributes are used with M:N relationships
    - In 1:N relationships, they can be transferred to the entity type on the N-side of the relationship

Example Attribute of a Relationship Type: Hours of WORKS_ON

Figure 3.2 An ER schema diagram for the COMPANY database.



Notation for Constraints on Relationships

- Cardinality ratio (of a binary relationship): 1:1, 1:N, N:1, or M:N
  - Shown by placing appropriate numbers on the relationship edges.
- Participation constraint (on each participating entity type): total (called existence dependency) or partial.
  - Total shown by double line, partial by single line.
- **Note**: These are easy to specify for Binary Relationship Types.

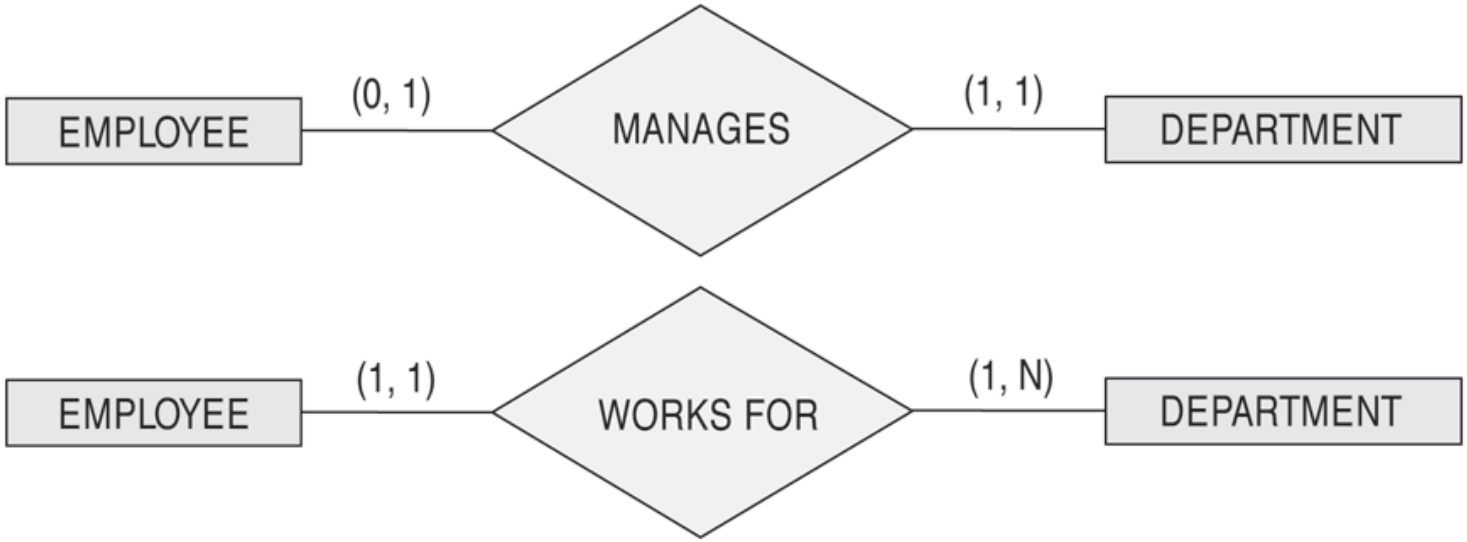
Alternative (Min, Max) Notation for Relationship Structural Constraints: (1 of 2)

- Specified on each participation of an entity type E in a relationship type R
- Specifies that each entity e in E participates in at least **min** and at most **max** relationship instances in R
- Default(no constraint): min=0, max=n (signifying no limit)
- Must have $\min \leq \max, \min \geq 0, \max \geq 1$
- Derived from the knowledge of mini-world constraints

Alternative (Min, Max) Notation for Relationship Structural Constraints: (2 of 2)

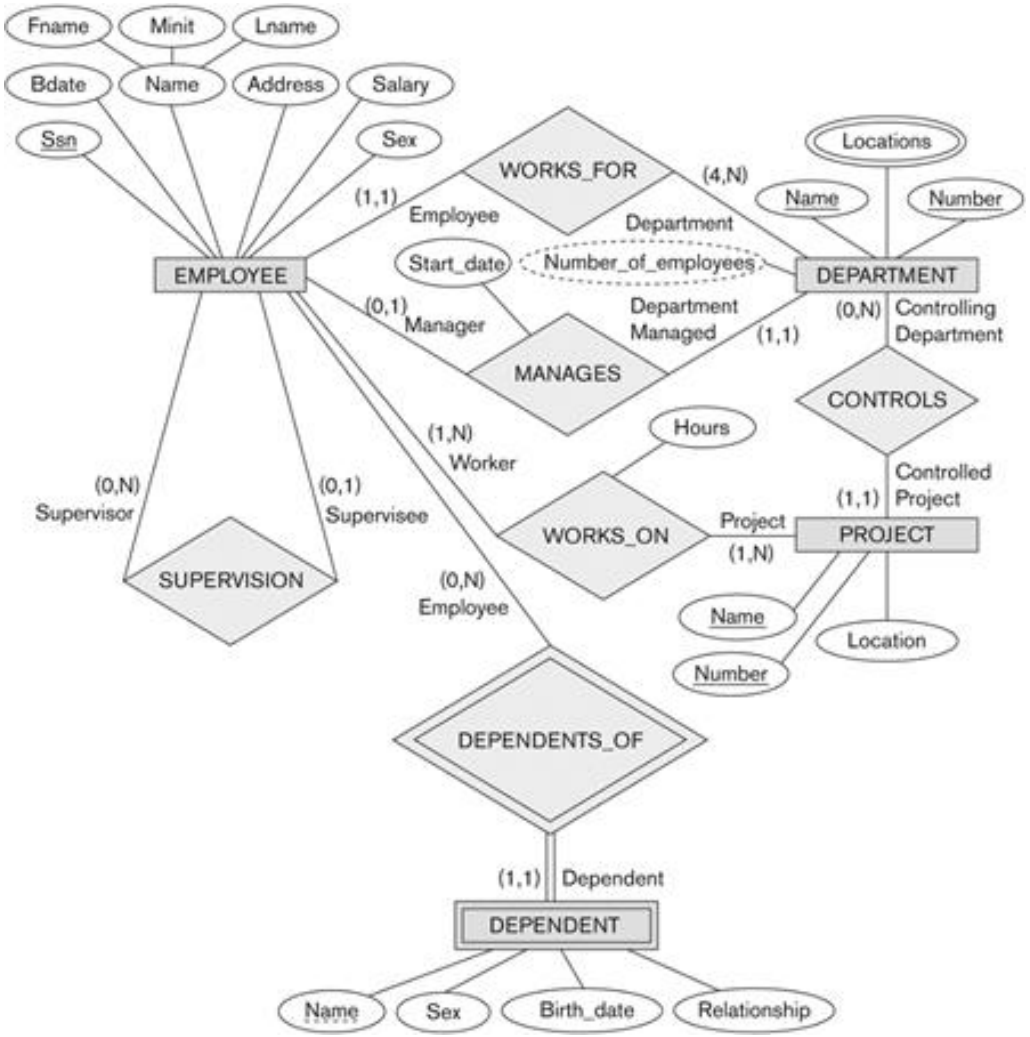
- Examples:
  - A department has exactly one manager and an employee can manage at most one department.
    - Specify (0,1) for participation of EMPLOYEE in MANAGES
    - Specify (1,1) for participation of DEPARTMENT in MANAGES
  - An employee can work for exactly one department but a department can have any number of employees.
    - Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
    - Specify (0,n) for participation of DEPARTMENT in WORKS_FOR

The (Min,Max) Notation for Relationship Constraints



Read the min, max numbers next to the entity type and looking **away from** the entity type

Company ER Schema Diagram Using (Min, Max) Notation



Relationships of Higher Degree

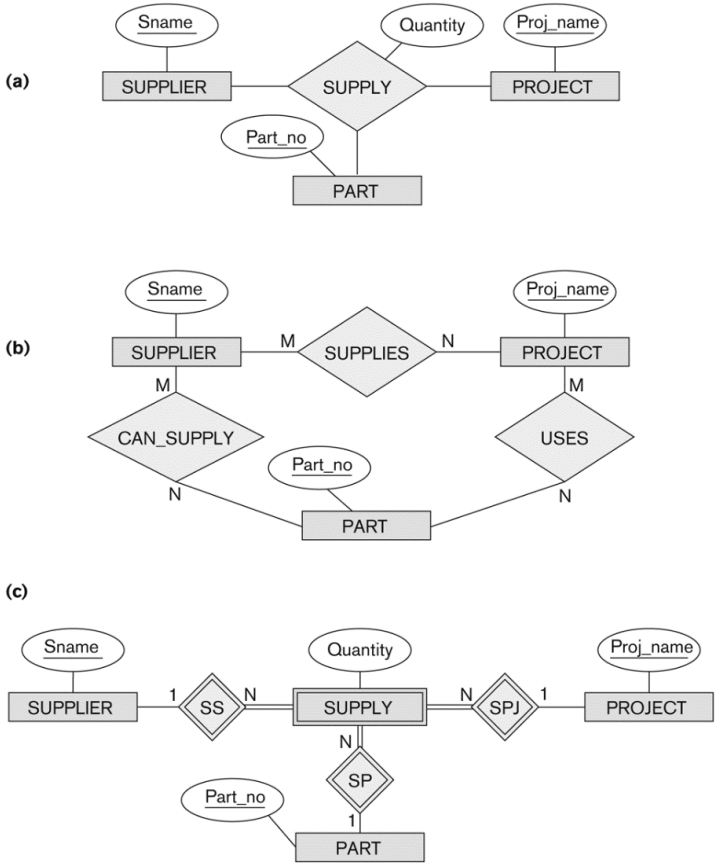
- Relationship types of degree 2 are called binary
- Relationship types of degree 3 are called ternary and of degree n are called n-ary
- In general, an n-ary relationship is not equivalent to n binary relationships
- Constraints are harder to specify for higher-degree relationships (n > 2) than for binary relationships

Discussion of N-Ary Relationships

- If a particular binary relationship can be derived from a higher-degree relationship at all times, then it is redundant
- In general, 3 binary relationships can represent different information than a single ternary relationship
- If needed, the binary and n-ary relationships can all be included in the schema design
- In some cases, a ternary relationship can be represented as a weak entity if the data model allows a weak entity type to have multiple identifying relationships (and hence multiple owner entity types)

Example of a Ternary Relationship

Figure 3.17 Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.



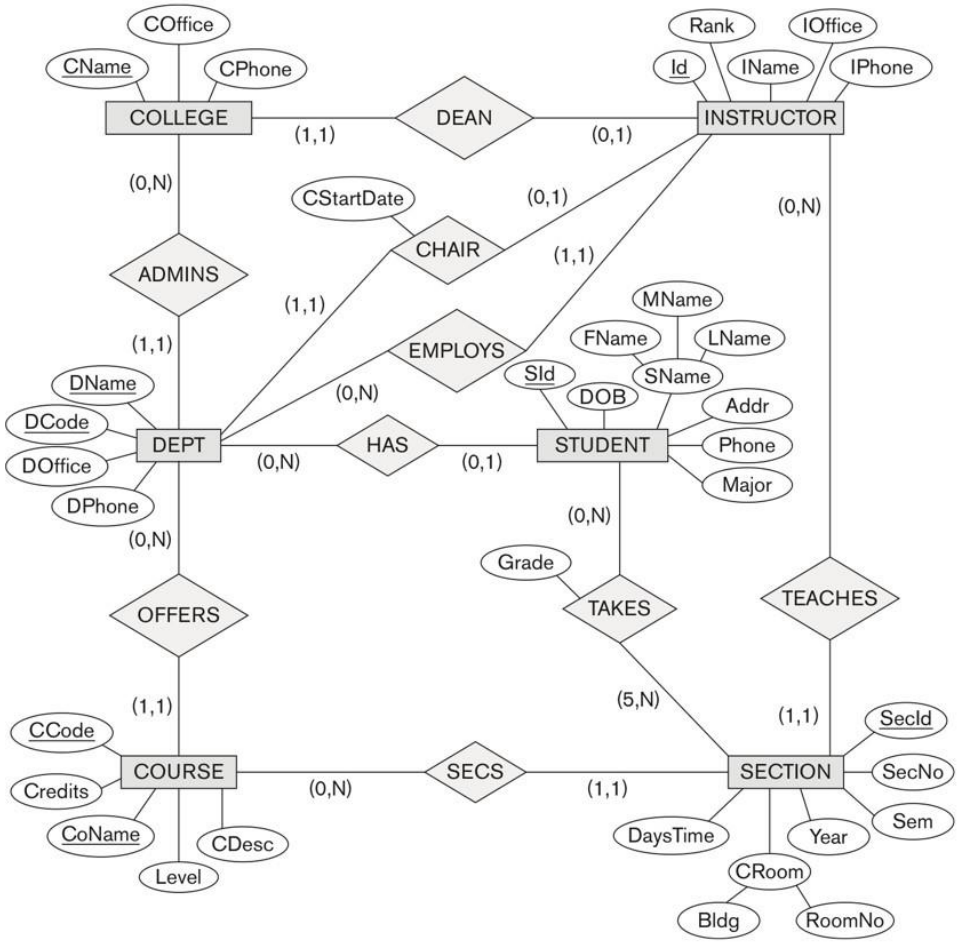
Displaying Constraints on Higher-Degree Relationships

- The (min, max) constraints can be displayed on the edges – however, they do not fully describe the constraints
- Displaying a 1, M, or N indicates additional constraints
  - An M or N indicates no constraint
  - A 1 indicates that an entity can participate in at most one relationship instance **that has a particular combination of the other participating entities**
- In general, both (min, max) and 1, M, or N are needed to describe fully the constraints
- Overall, the constraint specification is difficult and possibly ambiguous when we consider relationships of a degree higher than two.

Another Example: A University Database

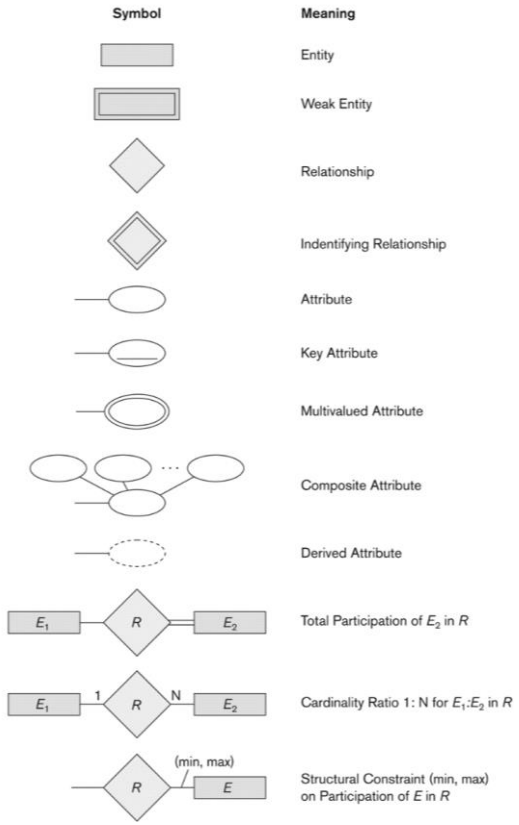
- To keep track of the enrollments in classes and student grades, another database is to be designed.
- It keeps track of the COLLEGEs, DEPARTMENTs within each college, the COURSEs offered by departments, and SECTIONs of courses, INSTRUCTORS who teach the sections etc.
- These entity types and the relationships among these entity types are shown on the next slide in Figure 3.20.

University Database Conceptual Schema



Summary of Notation for ER Diagrams

Figure 3.14 Summary of the notation for ER diagrams.



UML Class Diagrams

- Represent classes (similar to entity types) as large rounded boxes with three sections:
  - Top section includes entity type (class) name
  - Second section includes attributes
  - Third section includes class operations (operations are not in basic ER model)
- Relationships (called associations) represented as lines connecting the classes
  - Other UML terminology also differs from ER terminology
- Used in database design and object-oriented software design
- UML has many other types of diagrams for software design

UML Class Diagram for Company Database Schema

Figure 3.16 The COMPANY conceptual schema in UML class diagram notation.