Chapter 6

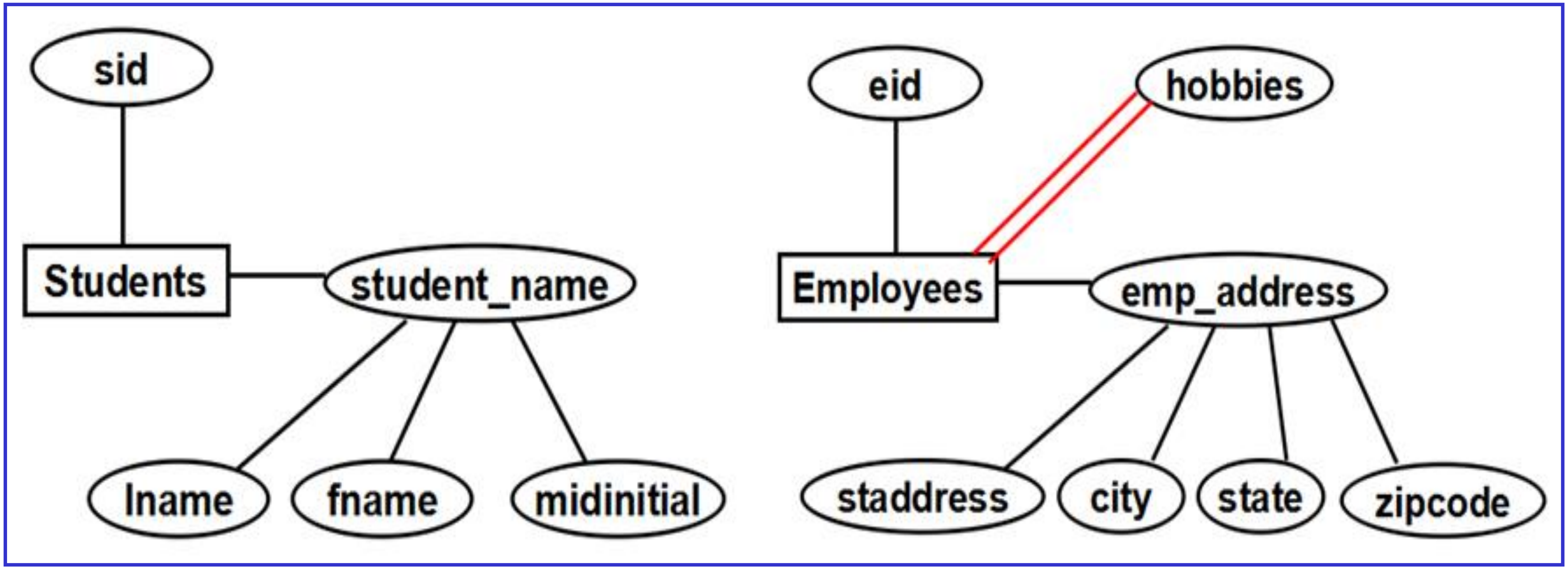
Database Design

Contents

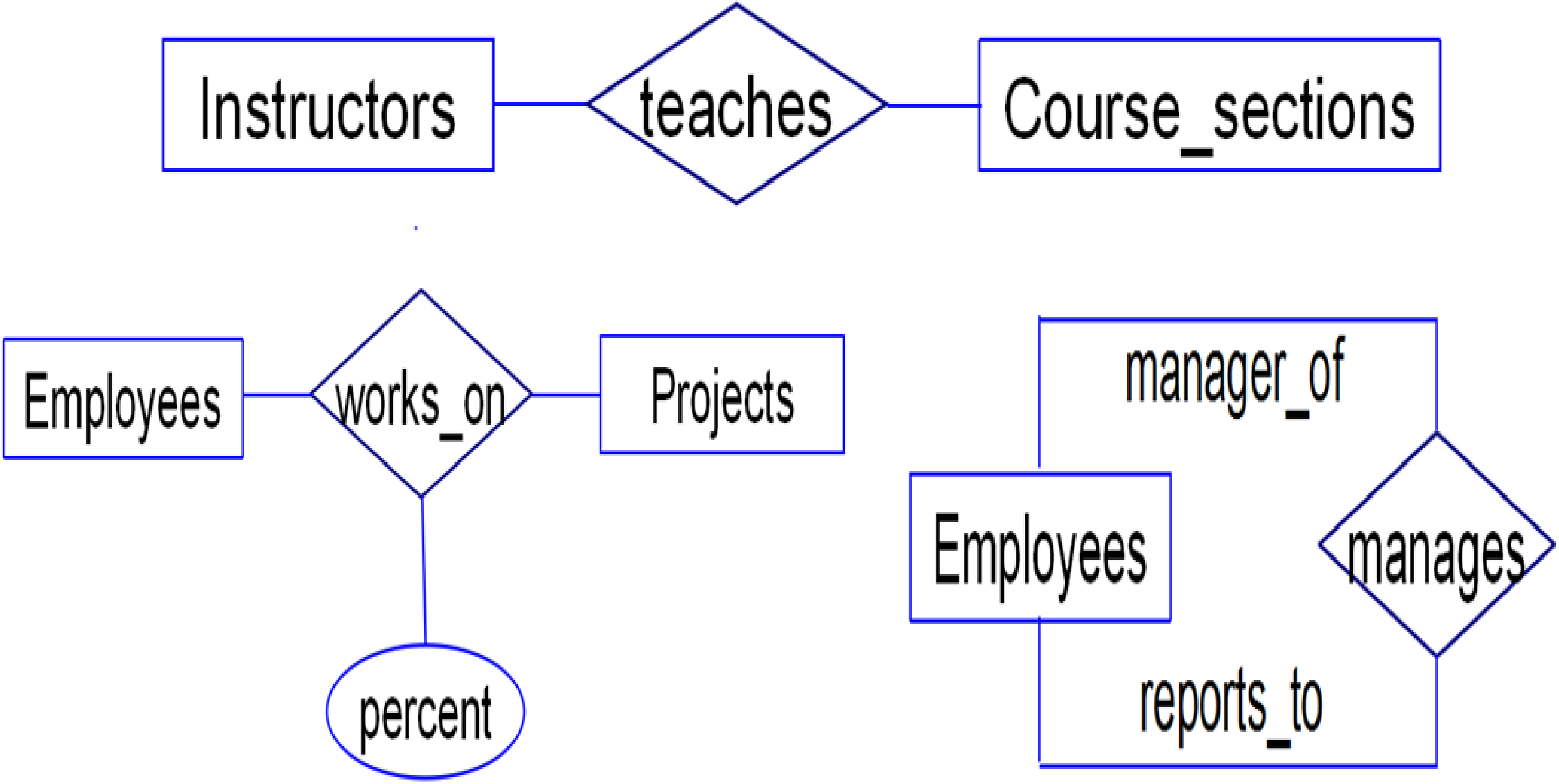
Review of E-R Model (1)

- E-R Model
  - Entity
  - Attribute
    - Identifier, Descriptor, Composite Attribute, Multi-Valued Attribute
  - Relationship
    - Card(E, R) = (x, y)
      - *single-valued/multi-valued participation*
      - *mandatory/optional participation*
    - One-to-One, Many-to-Many, Many-to-One

Examples of E-R diagrams (1)



Examples of E-R diagrams (2)



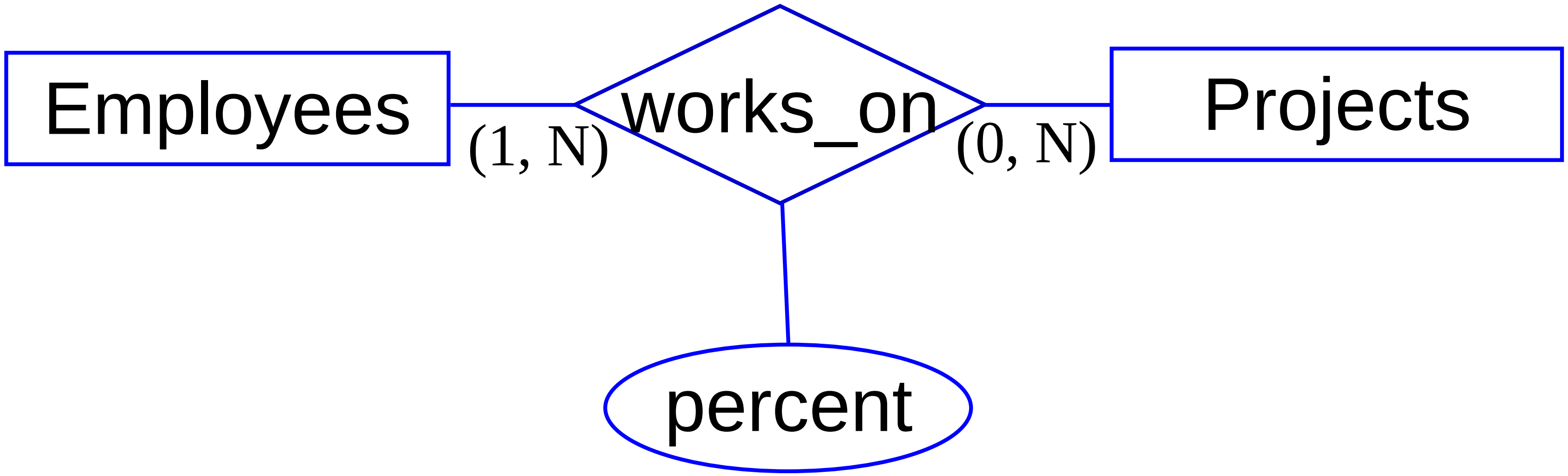
Review of E-R Model (2)

Transform E-R Model to Relations

- Rule 1: Entity
- Rule 2: Multi-valued Attribute
- Rule 3: N-N Relationships
- Rule 4. N-1 Relationships
- Rule 5: 1-1 Relationships: Optional on one side
- Rule 6: 1-1 Relationships: Mandatory on both sides

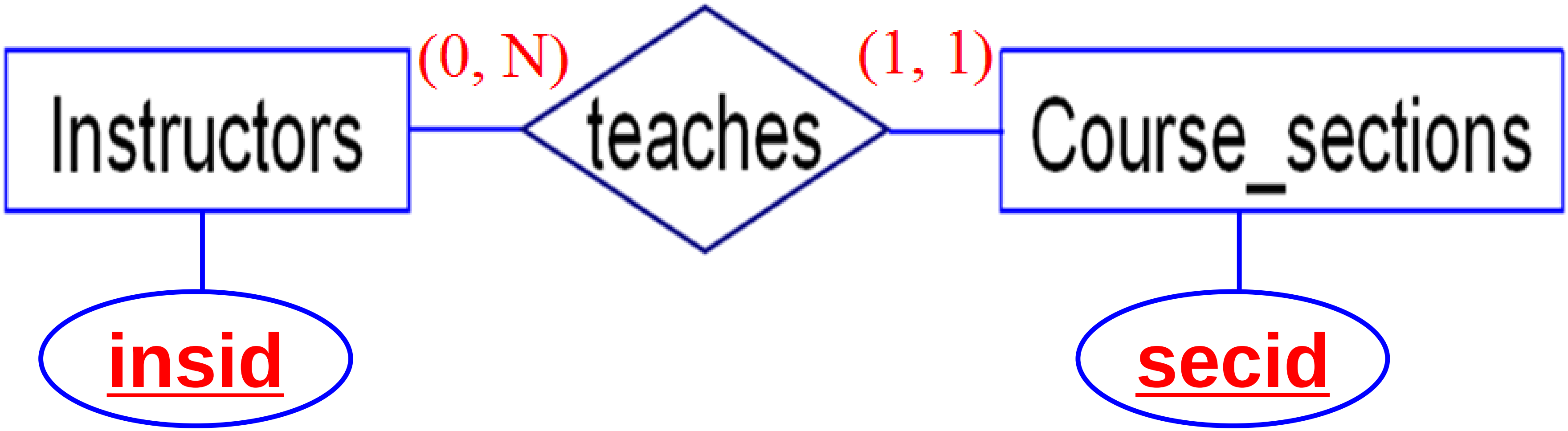
Rule 3: N-N Relationships

- **Employees(<u>eid</u>, straddr, city, ……)**
- **Projects(<u>prid</u>, proj_name, due_date)**



- **works_on(<u>eid, prid</u>, percent)**

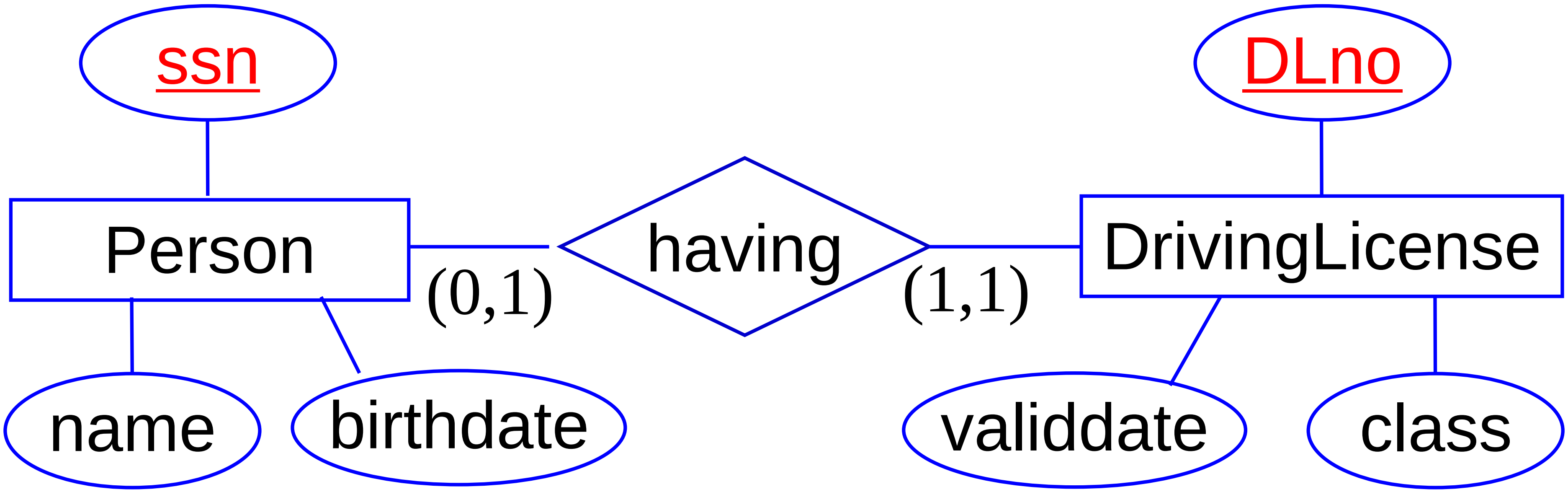
Rule 4. N-1 Relationships



Instructors(<u>insid</u>, Iname, ……)

Course_sections(<u>secid</u>, <u>*insid*</u>, course, …)

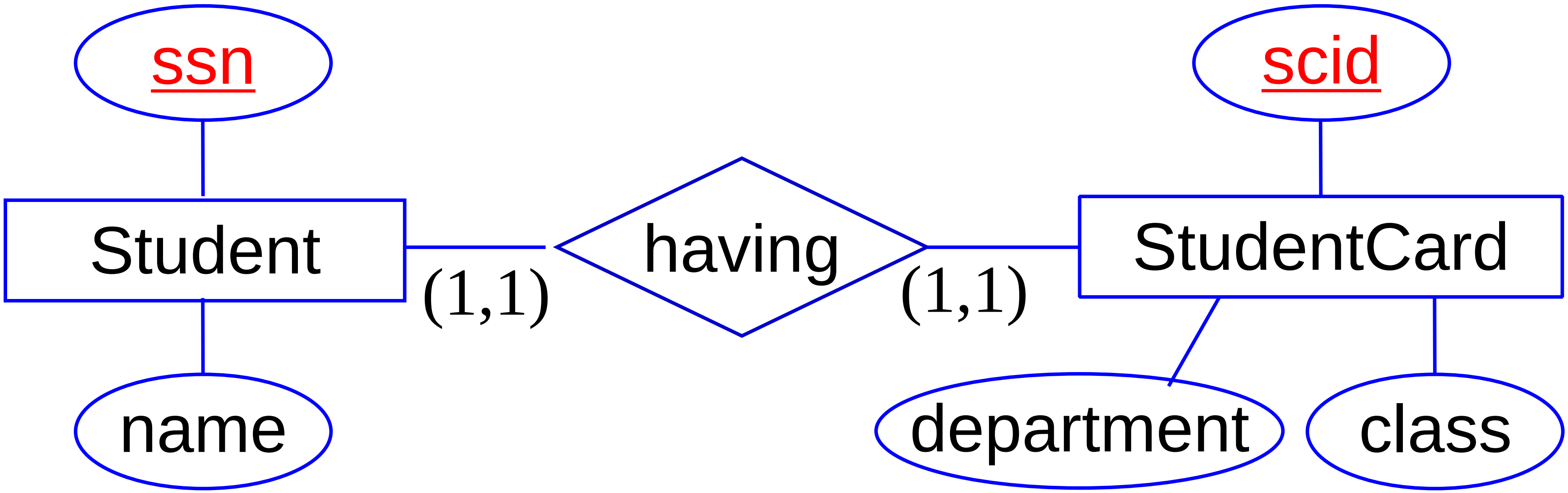
Rule 5: 1-1 Relationships: Optional on one side



Person(<u>ssn</u>, name, birthdate)

DrivingLicense(<u>DLno</u>, validdate, class, <u>ssn</u>)

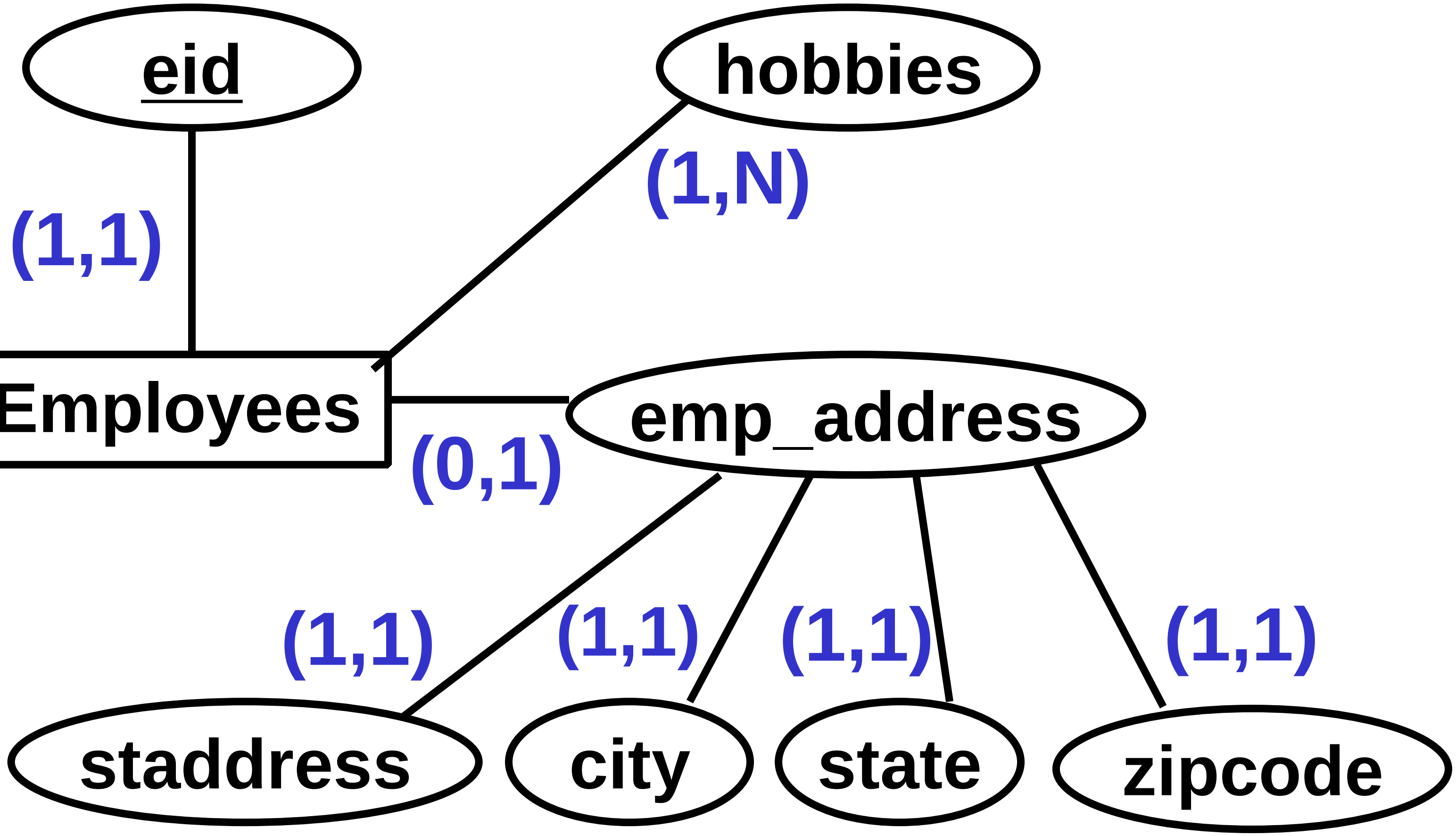
Rule 6: 1-1 Relationships: Mandatory on both sides



Student(ssn, name, scid, department, class)

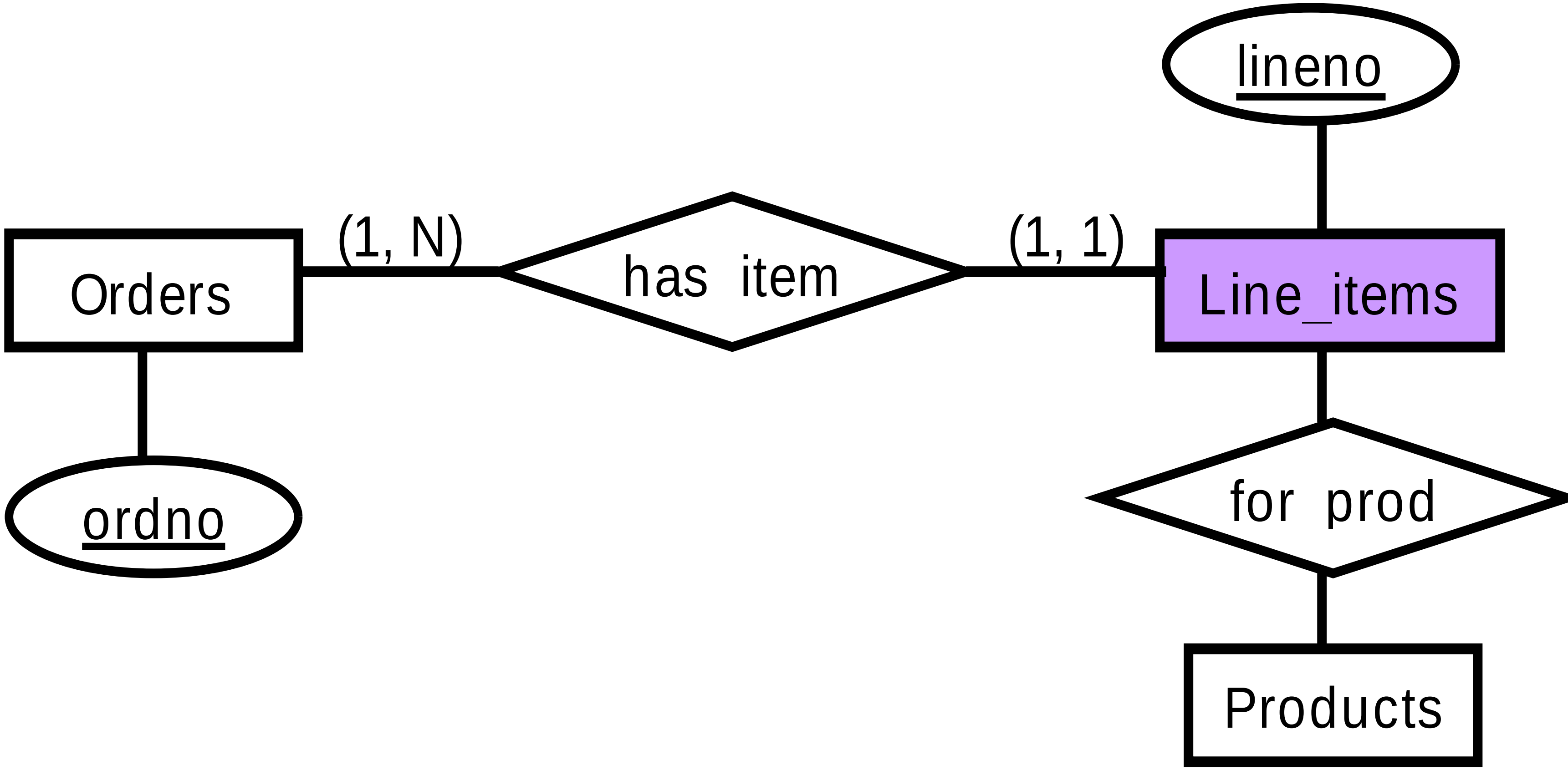
Review of E-R Model (3)

- **Cardinality of Attributes (x, y)**



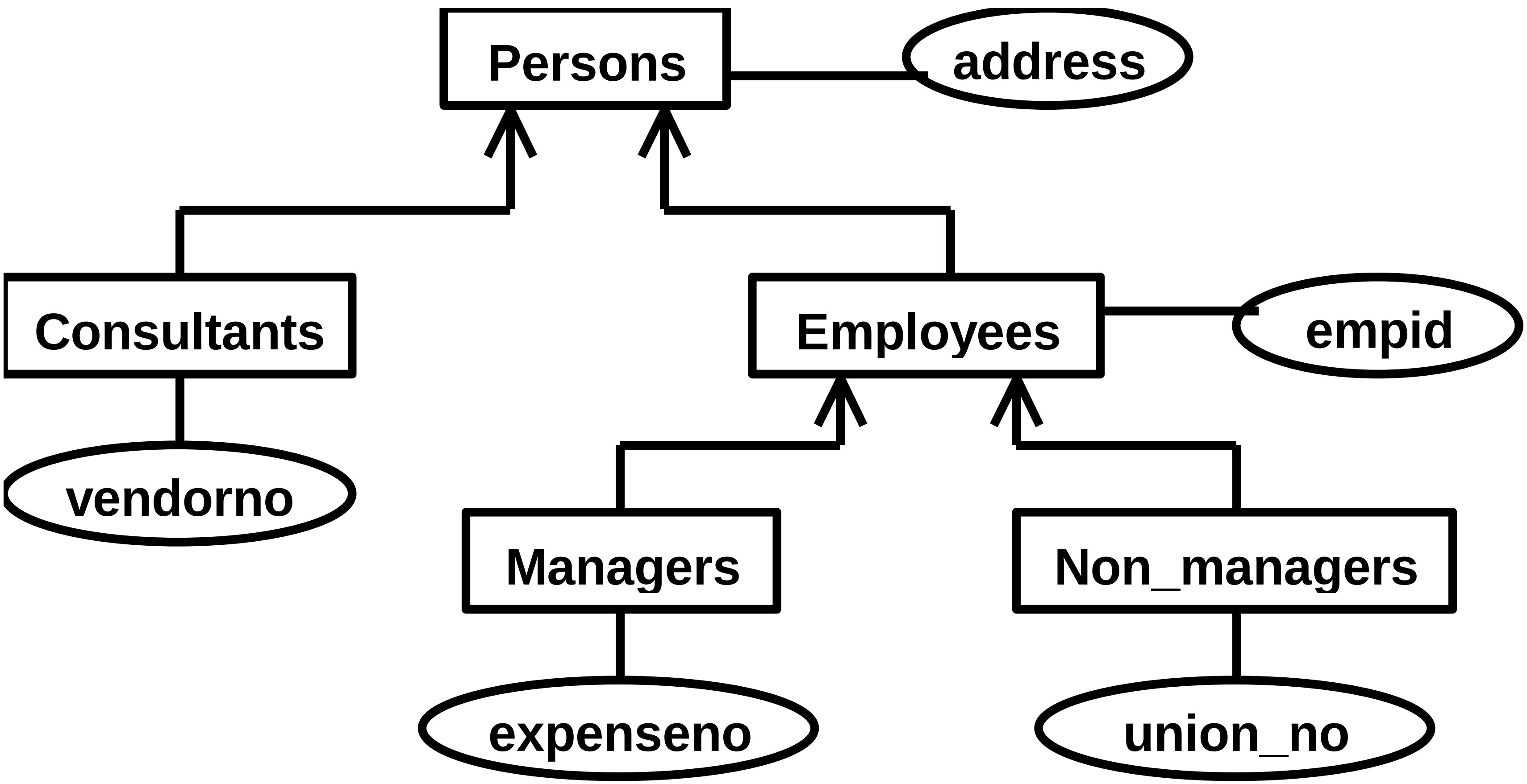
Review of E-R Model (4)

❑Weak Entities



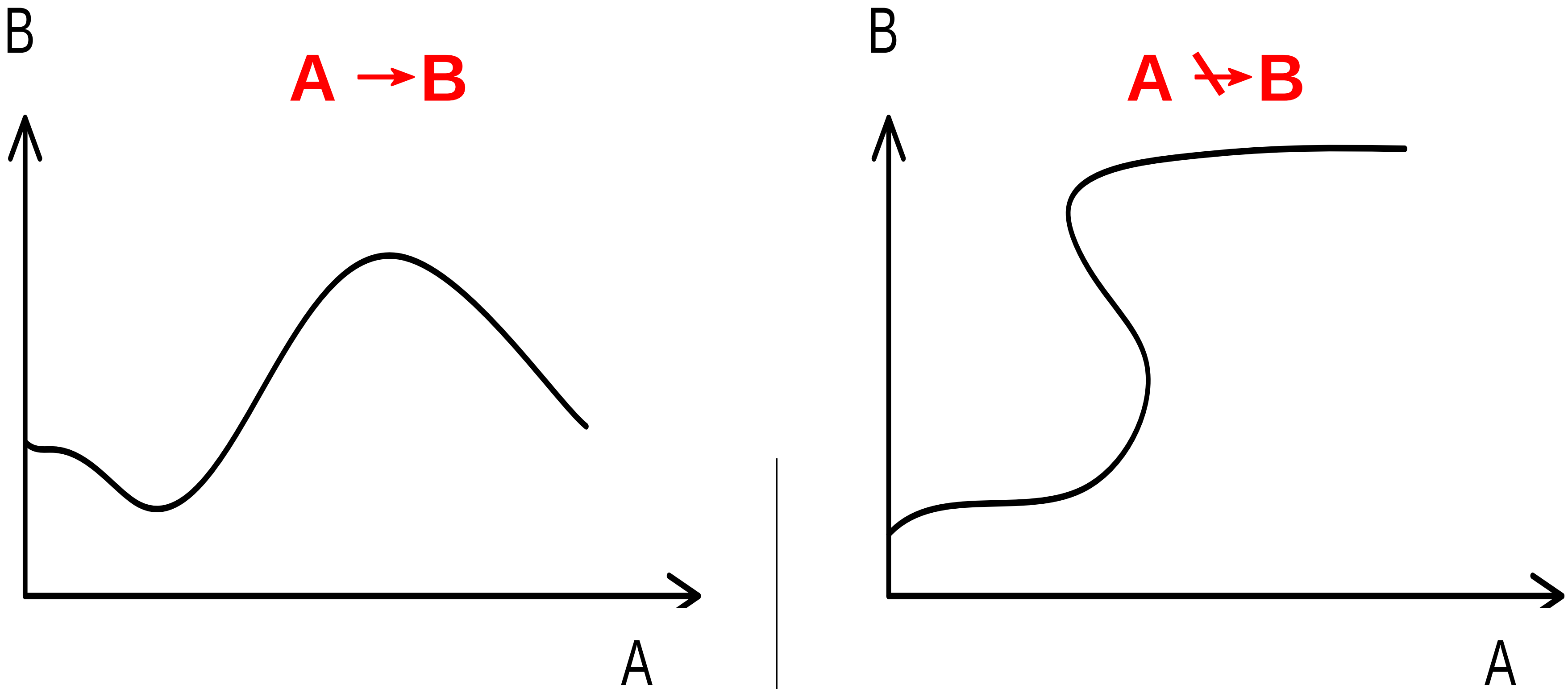
Review of E-R Model (5)

❑ Generalization Hierarchies



Review of Functional Dependency

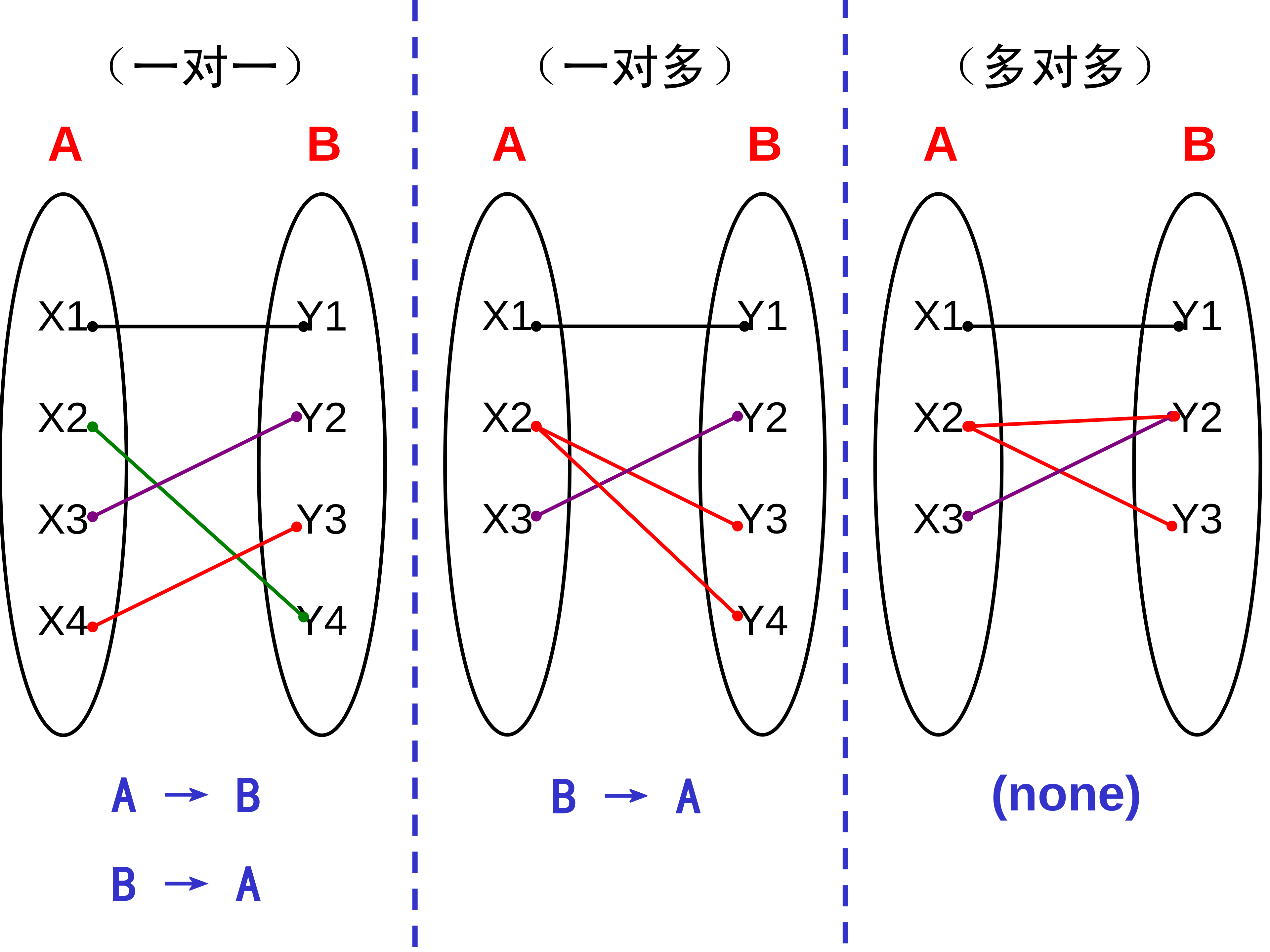
- **Functional Dependency (FD, 函数依赖)**
  - Def. of FD
  - Armstrong’s Axioms (Armstrong公理)
- **Minimal Cover (最小覆盖)**
  - Closure of a Set of FDs (函数依赖集的闭包)
  - FD Set Cover (函数依赖集的覆盖)
  - Equivalence of two sets of FDS (函数依赖集的等价)
  - Algorithm 6.6.13：最小覆盖计算算法
- **Closure of a Set of Attributes (属性集的闭包)**
  - Algorithm 6.6.12：属性集闭包计算算法



A functionally determines B. Each value of A corresponds to only one value of B.

A does not functionally determine B. Some values of A correspond to more than one value of B.

Figure 6.18 Graphical Depiction of Functional Dependency



（一对一）
A
B
X1
X2
X3
X4
Y1
Y2
Y3
Y4
A → B
B → A
（一对多）
A
B
X1
X2
X3
Y1
Y2
Y3
Y4
B → A
（多对多）
A
B
X1
X2
X3
Y1
Y2
Y3
(none)

Review of Armstrong's Axioms

❑ Armstrong's Axioms

Rule 1(自反规则): If $Y \subseteq X$，then $X \rightarrow Y$

Rule 2(传递规则): If $X \rightarrow Y$ and $Y \rightarrow Z$，then $X \rightarrow Z$

Rule 3(增广规则): If $X \rightarrow Y$, then $XZ \rightarrow YZ$

❑ Some Implications of Armstrong's

Rule 4(合并规则): If $X \rightarrow Y$ and $X \rightarrow Z$，then $X \rightarrow YZ$

Rule 5(分解规则): If $X \rightarrow YZ$，then $X \rightarrow Y$ and $X \rightarrow Z$

Rule 6(伪传递规则):If $X \rightarrow Y$，and $WY \rightarrow Z$，then $XW \rightarrow Z$

Rule 7(聚积规则): If $X \rightarrow YZ$ and $Z \rightarrow W$，then $X \rightarrow YZW$

Review of Minimal Cover

- **Closure of a Set of FDs (函数依赖集F的闭包)**
  - $F^+ = \{ X \rightarrow A \mid F \models X \rightarrow A \}$
- **FD Set Cover (函数依赖集的覆盖)**
  - **F covers G iff $G \subseteq F^+$**
- **Equivalence of two sets of FDS (函数依赖集的等价)**
  - **F covers G and G covers F**

Review of Functional Dependency

- **Closure of a Set of Attributes (属性集的闭包)**
  - **Algorithm 6.6.12：属性集闭包计算算法**

- **Minimal Cover (最小覆盖)**
  - **Algorithm 6.6.13：最小覆盖计算算法**

Review of Closure of a Set of Attributes

- **Def. 6.6.11:** $X^+_F = \{ A \mid X \rightarrow A \in F^+ \}$

- **algorithm 6.6.12**

a) $X^+ := X$;

b) repeat

  $oldX^+ := X^+$;

  for each functional dependency $Y \rightarrow Z$ in F do

    if $Y \subseteq X^+$ then $X^+ := X^+ \cup Z$;

c) until ($oldX^+ = X^+$)

Algorithm 6.6.13 : Minimal Cover

- **step 1: From the set F of FDs, we create an equivalent set H of FDs, with only single attributes on the right side.**
- **step 2: From the set H of FDs, successively remove individual FDs that are inessential in H.**
- **step 3: From the set H of FDs, successively replace individual FDs with FDs that have a smaller number of attributes on the left-hand side, as long as the result does not change H^+.**
- **step 4: From the remaining set of FDs, gather all FDs with equal left-hand sides and use the union rule to create an equivalent set of FDs M where all left-hand sides are unique.**

Review of Normalization

- **The process of normalization**
  - **Decompositions of table T**
  - **$Head(T) = Head(T_1) \cup Head(T_2) \cup \ldots\ldots \cup Head(T_k)$**
- **Lossless Decomposition (无损分解)**
  - **$T \equiv T_1 \infty T_2 \infty \ldots \infty T_k$**
  - **Theorem 6.7.3 & 6.7.4：无损分解的判定定理**
- **Lossy Decomposition (有损分解)**
  - **$T \subset T_1 \infty T_2 \infty \ldots \infty T_k$**

Content of next

- **Def. 6.8.3 FD Preserved (依赖保持性)**
- **Superkey & Key**
  - **Algorithm to Find Candidate Key**
  - **PRIME ATTRIBUTE (主属性)**
  - **NON-PRIME ATTRIBUTE (非主属性)**
- **Normal Forms:**
  - **2NF, 3NF, BCNF**
- **Algorithm 6.8.8**

An Algorithm to Find Candidate Key

- Given a table T with a set F of FDs

```
1. set K := Head(T) ;
2. for each attribute A in K
   {
      compute (K – A)F+ ;
      if (K – A)F+ contains all the attributes in T, then
      {
         set K := K – { A } ;
      }
   }
```

❑ BCNF和3NF定义的对比

❑ BCNF

- for any FD $X \rightarrow A$ in F^+ that lies in T (all attributes of X and A in T), A is a single attribute not in X, then X must be a superkey for T

❑ 3NF

- for any FD $X \rightarrow A$ implied by F that lies in T, if A is a single non-prime attribute not in X, then X must be a superkey for T.

- **3NF和2NF定义的对比**
- **3NF**
  - **for any X→A implied by F that lies in T, if A is a single non-prime attribute not in X, then X must be a superkey for T.**
- **2NF**
  - **for any X → A implied by F that lies in T, if A is a single non-prime attribute not in X, then X is not properly contained in any key of T.**

6.8 Normal Forms

- **Algorithm 6.8.8**
  1. replace F with minimal cover of F;
  2. S = Φ;
  3. for all $X \rightarrow Y$ in F

     if, for all $Z \in S$, $X \cup Y \not\subseteq Z$

     then $S = S \cup \text{Heading}(X \cup Y)$

     end for
  4. If, for all candidate keys K for T:

     for all $Z \in S$, $K \not\subseteq Z$

     then choose a candidate key K and set $S = S \cup \text{Heading}(K)$