MYONGJI UNIVERSITY

ER-Model / UML Data Modeling

Database Design Model

- Entity-Relationship Model (E/R)
- Unified Modeling Language (UML)

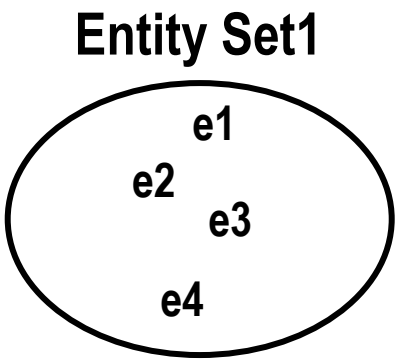
- 특징
  - Graphical
  - Relational Model로 변환 가능

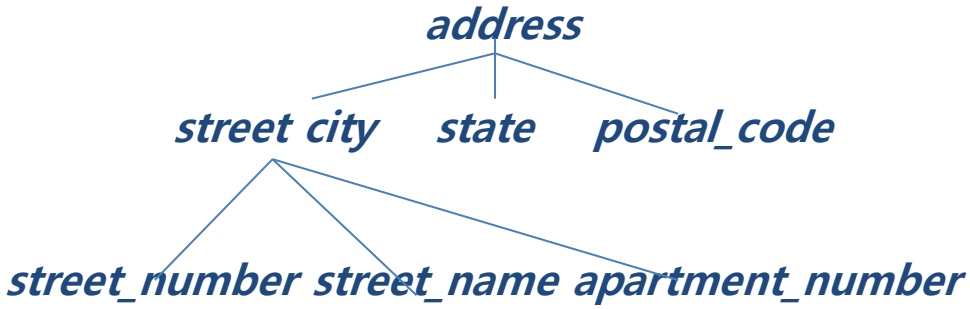
E/R

- 1976년 P. Chen에 의해 제안
  - "The Entity-Relationship Model: Toward a Unified View of Data", ACM Transactions On Database Systems, Jan.1976.
- Data Modeling을 위한 Model

Entity

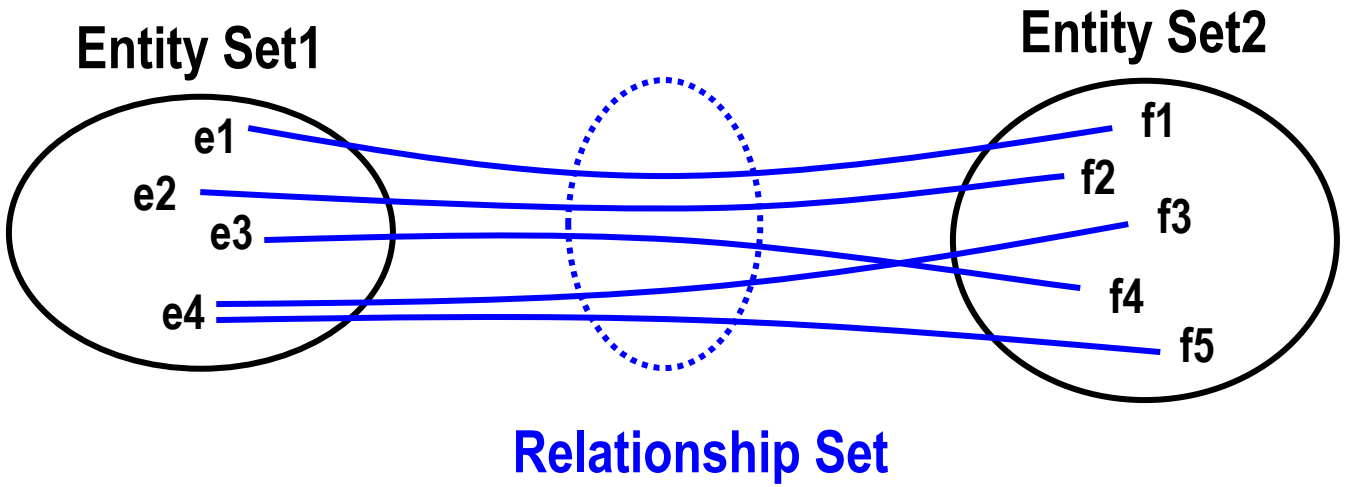
- Entity: 개체
  - 실 세계에 관념적/물리적으로 존재
  - Set of attribute로 표현됨
- Entity Set: 개체 집합
  - 동일한 속성을 갖는 개체의 집합
- Attribute: 속성
  - Simple vs. Composite Attribute
  - Single-valued vs. Multi-valued Attribute
  - Derived Attribute



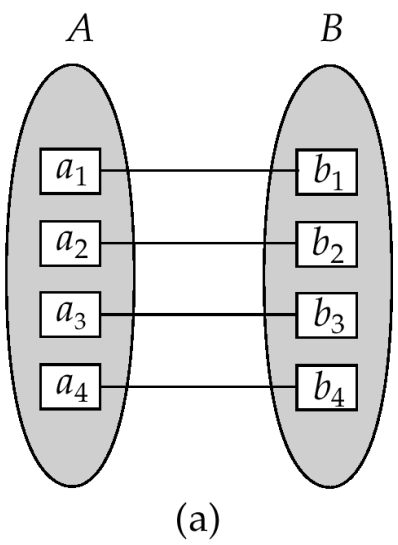


Relationship

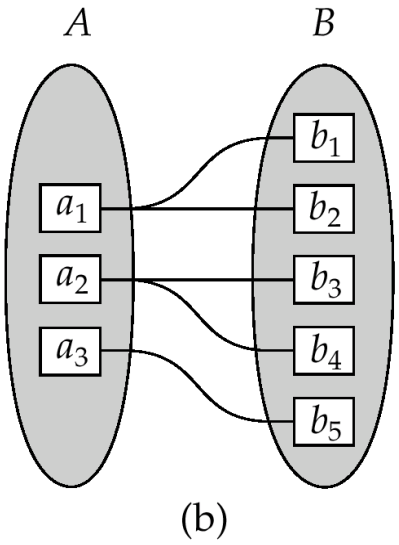
- Entity들의 상호관계
- Relationship set
- Role: Relationship에서 각 entity의 역할
- 둘 이상의 Entity Set 간의 Relationship set이 존재 가능
- Relationship set이 attribute를 갖는 경우도 있음



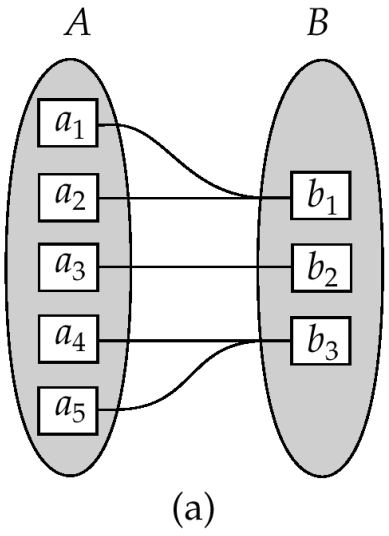
Cardinality



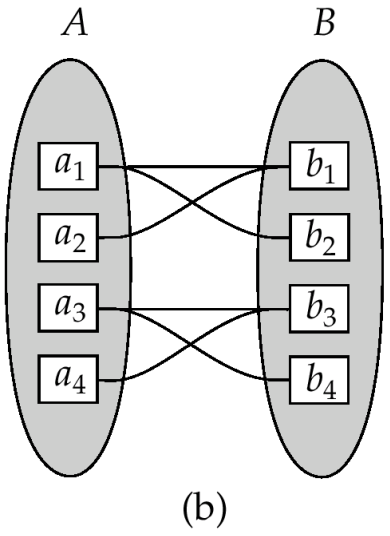
One to one



One to many



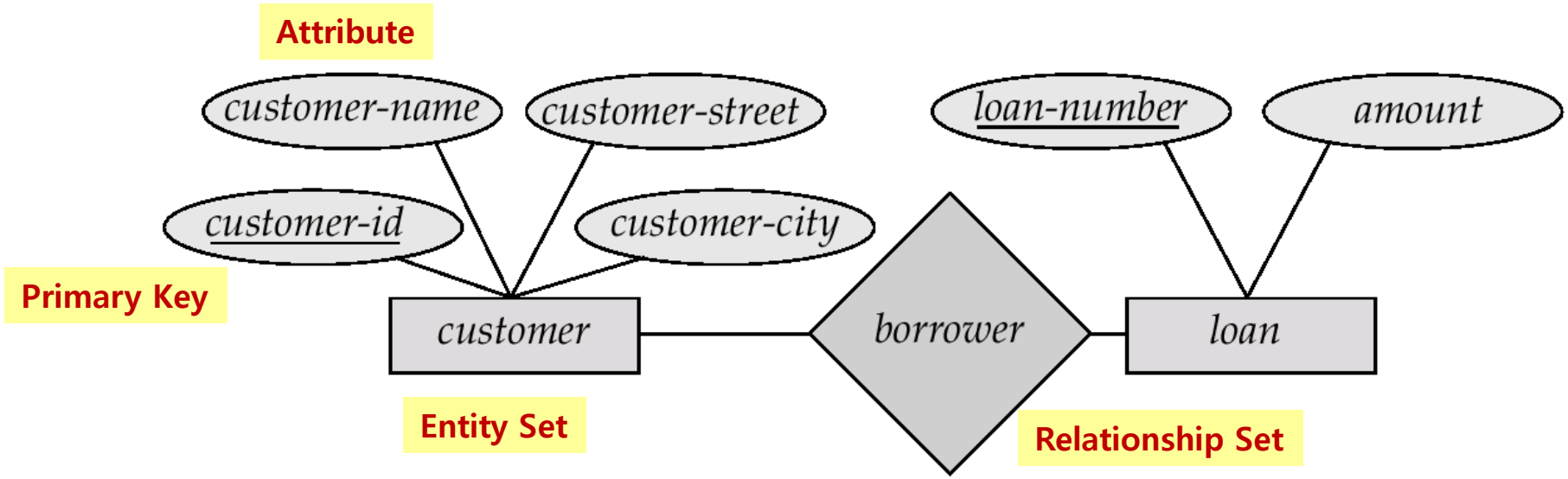
Many to one



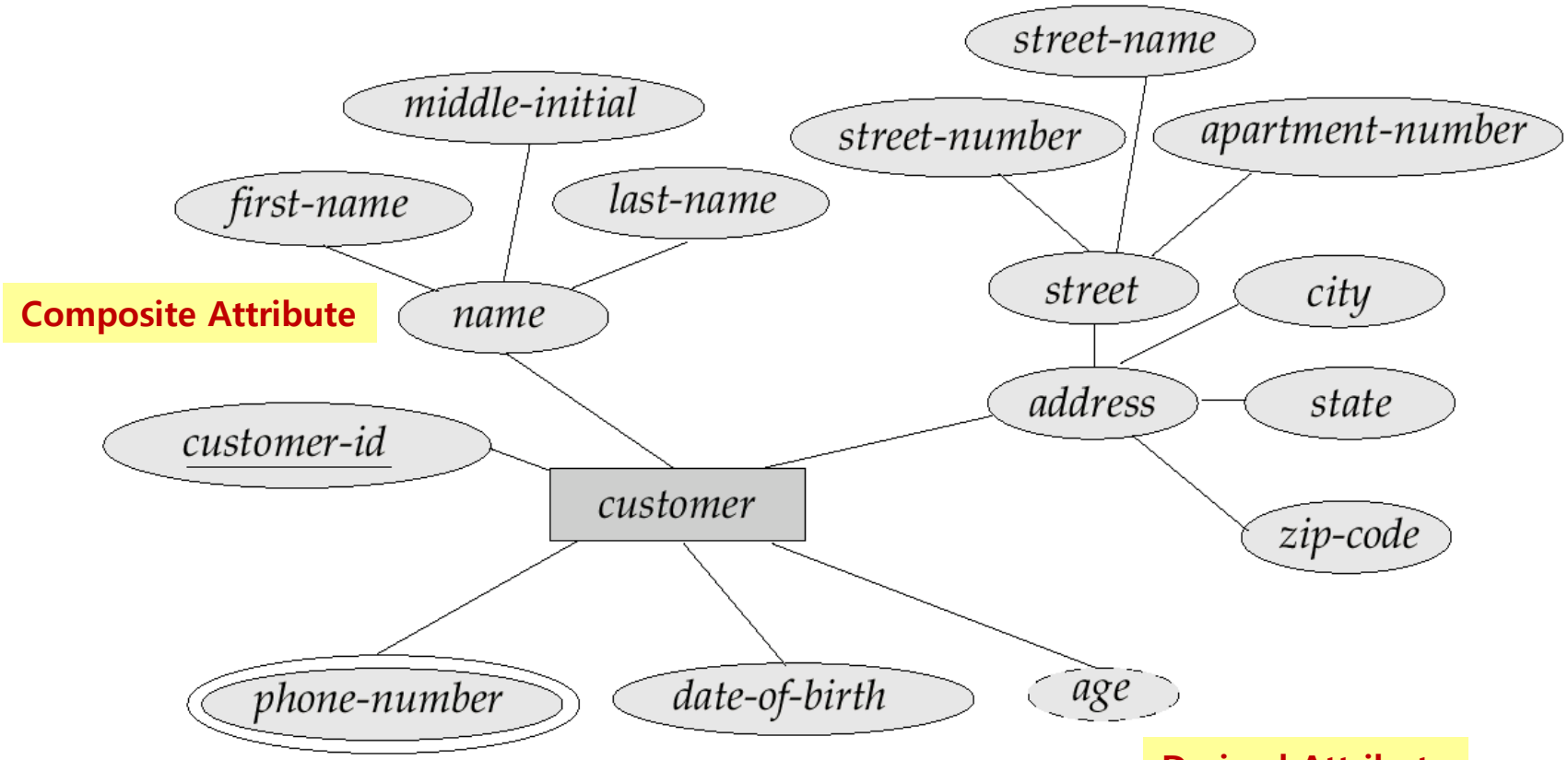
Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set (total vs. partial)

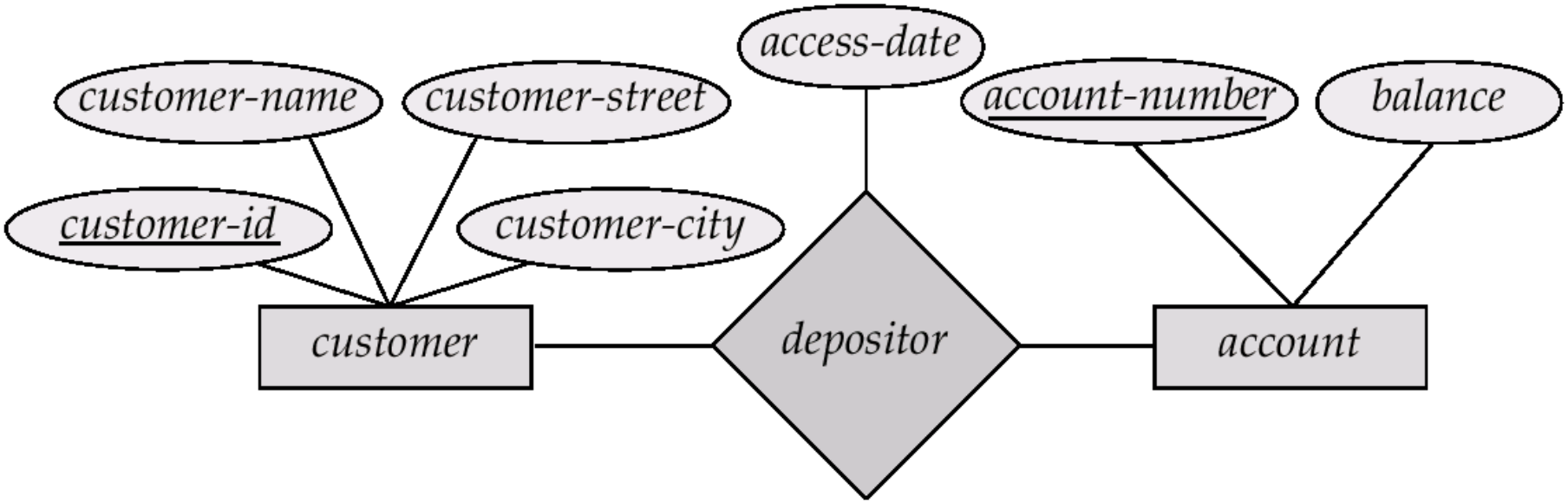
E-R Diagram: Entity Set & Relationship Set



E-R Diagram: Attributes Types

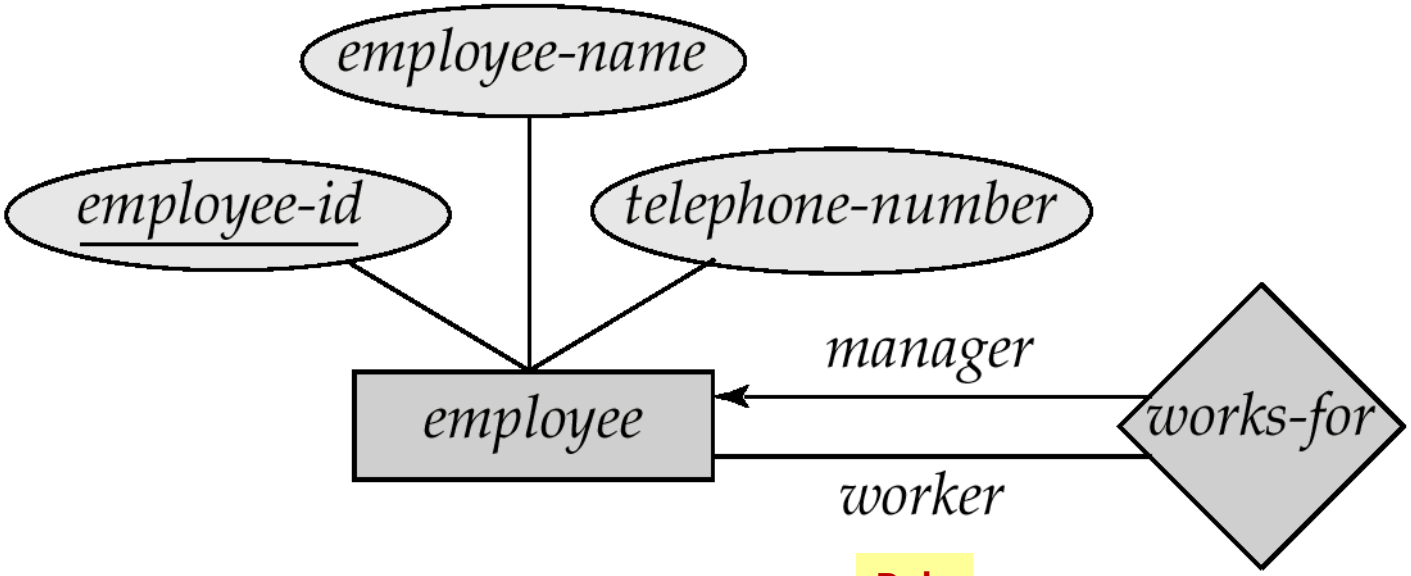


E-R Diagram: Relationship Set with Attribute



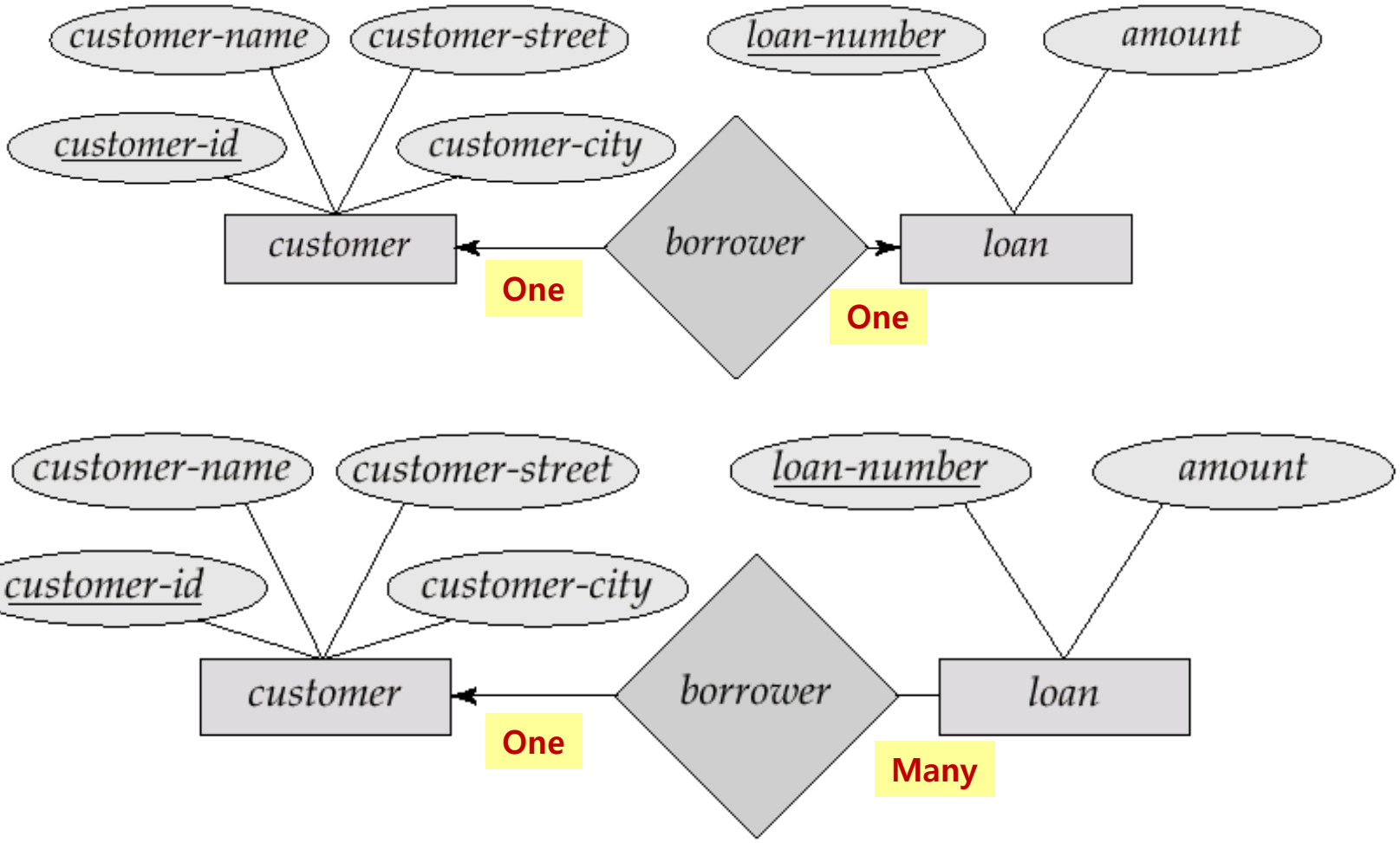
Relationship Set with Attribute

E-R Diagram: Roles

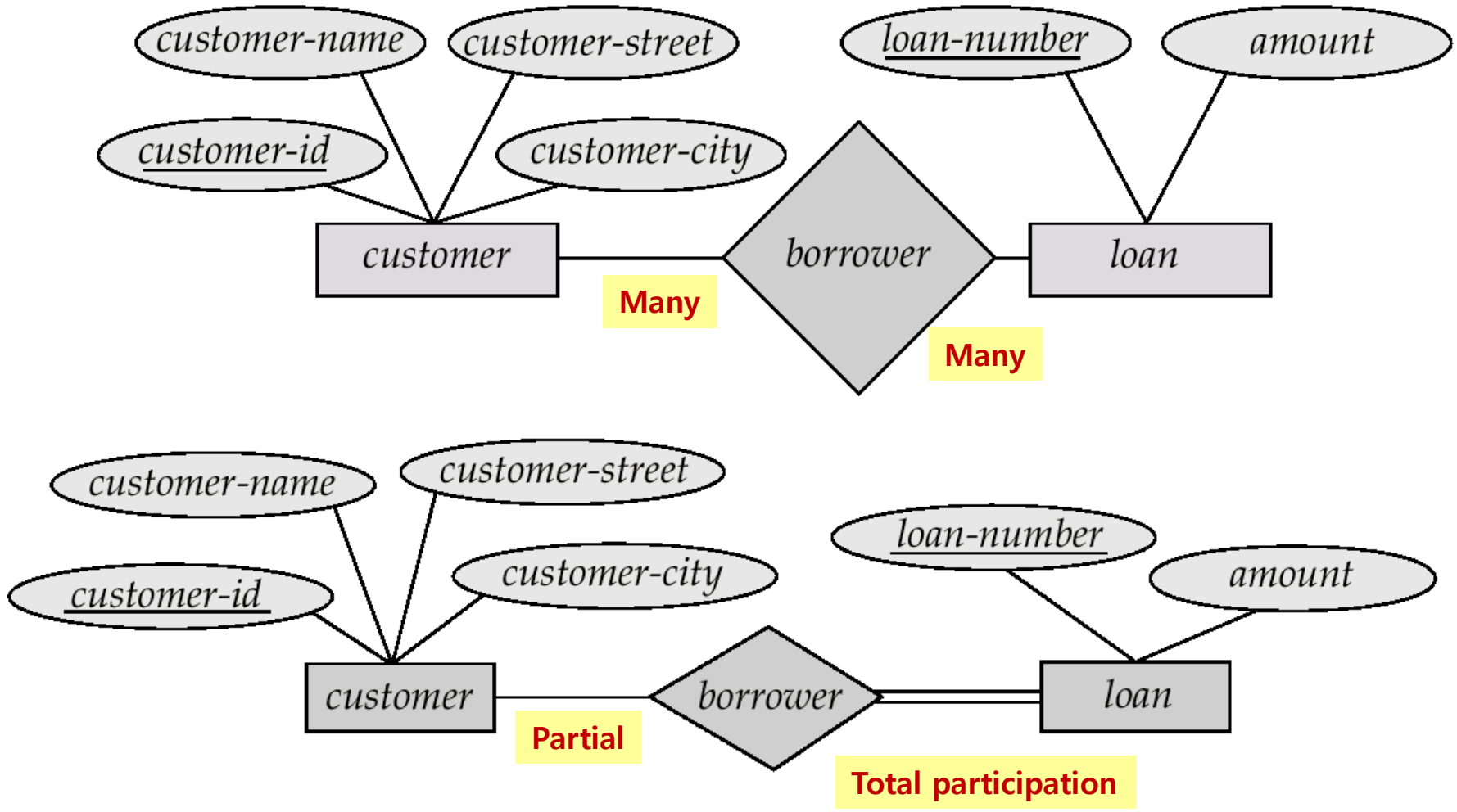


Role

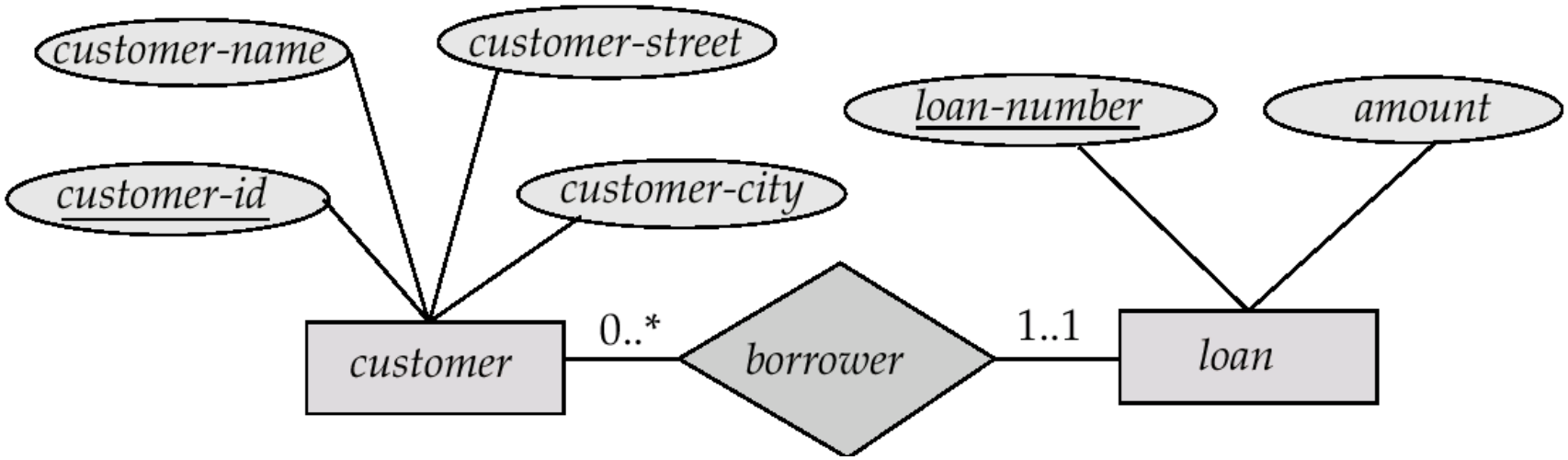
E-R Diagram: Cardinality



E-R Diagram: Cardinality



Alternative Notation for Cardinality Limits

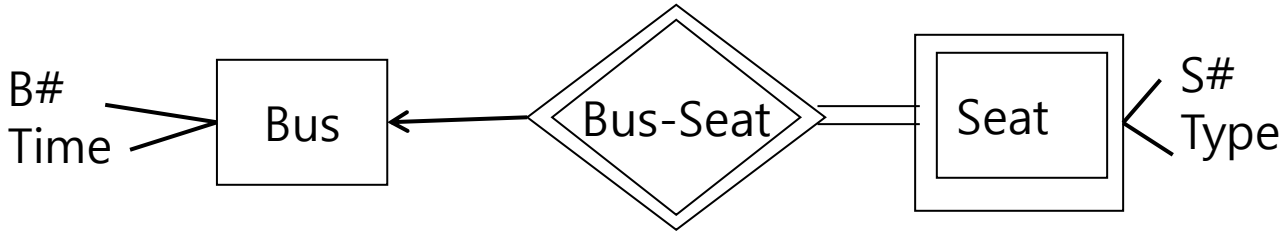


customer는 0개에서 무한대의 loan이 있음

loan은 반드시 customer가 1개 있음

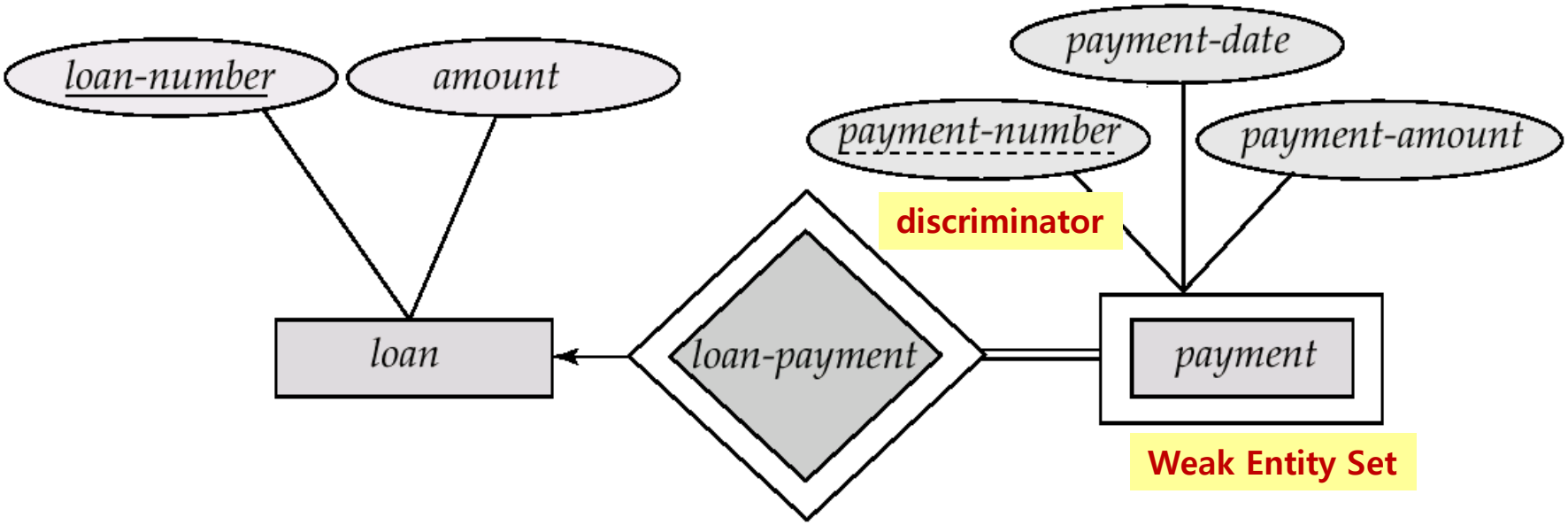
Weak Entity Sets

- Strong entity: 자신의 attribute만으로 PK를 만들 수 있음
- Weak entity: 자신의 attribute만으로 PK를 만들지 못함



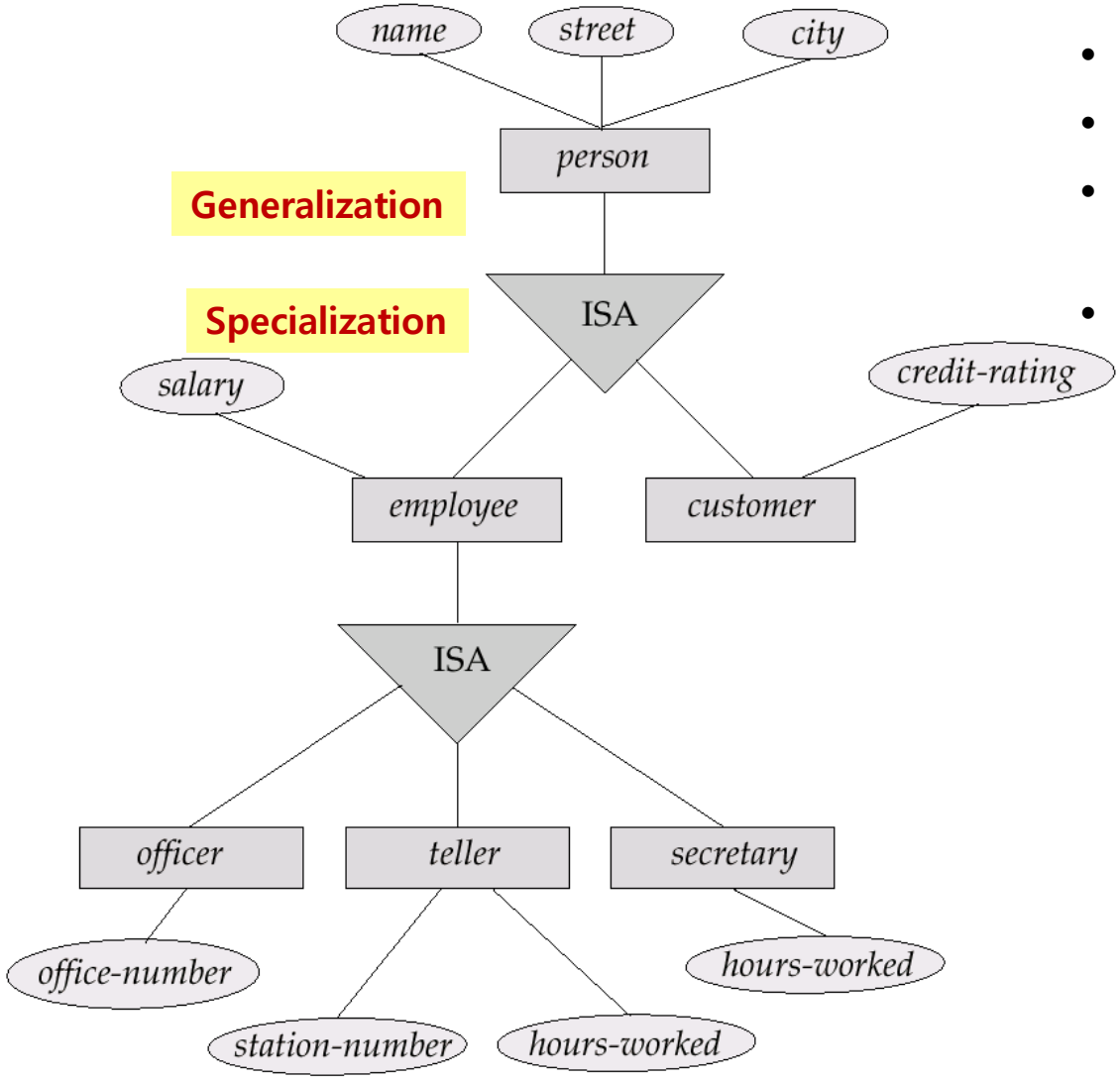
  - A weak entity set is dependent on a strong entity set
  - Primary key of a weak entity set

  = primary key of its dominant entity set + its discriminator
    - **B# + S#**



PK for payment: (loan-number, payment-number)

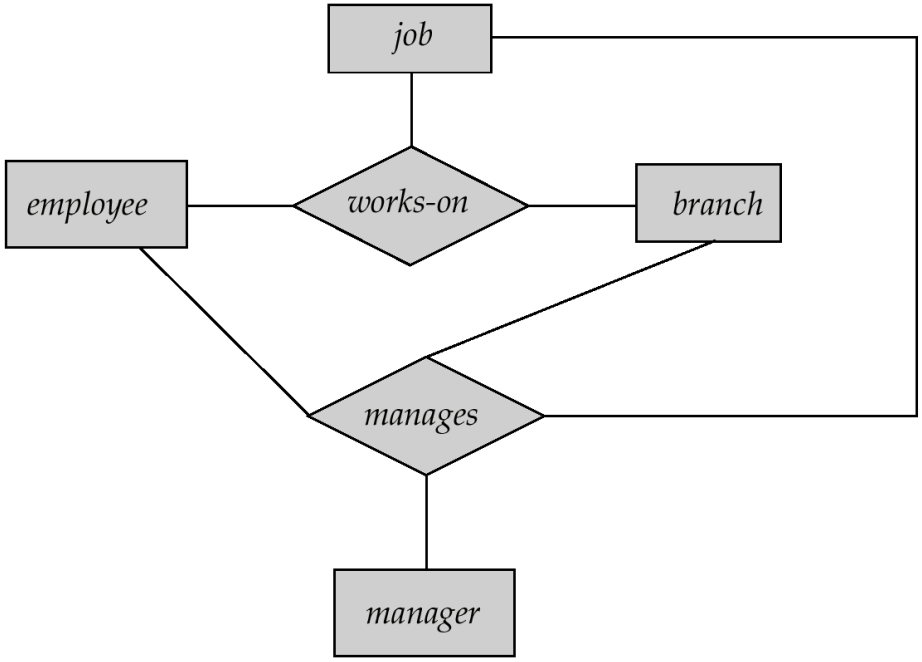
Specialization & Generalization



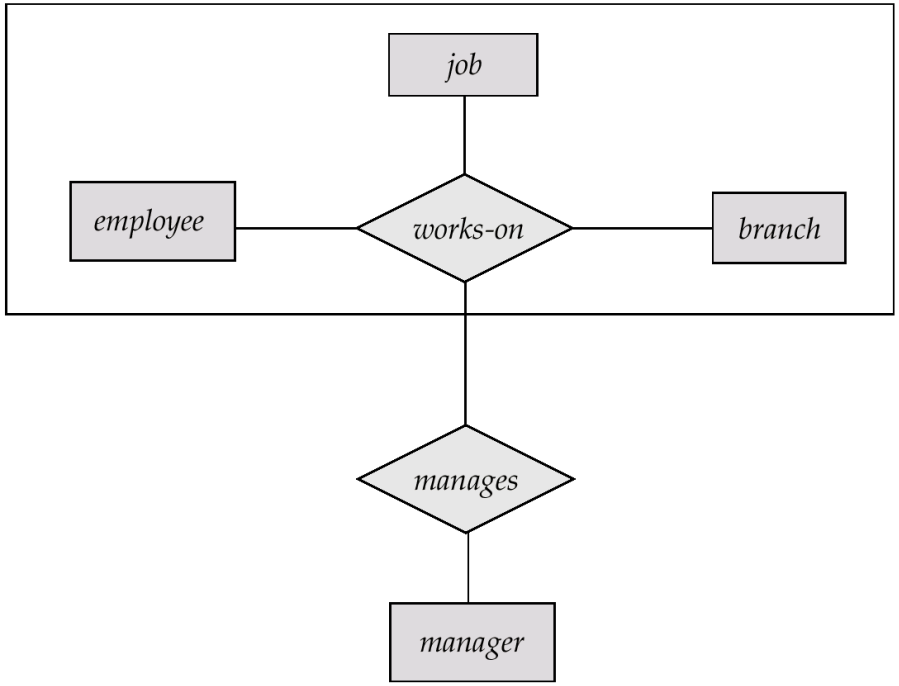
- Generalization: Super-Entity
- Specialization: Sub-Entity
- Inheritance
  - 상위 Entity의 Attribute는 상속됨
- Types of generalization (super-sub entities)
  - Disjoint vs. Overlapping
    - E면서 C인 P가 있나?
  - Total vs. Partial
    - E도 C도 아닌 P가 있나?

E-R Diagram With Aggregation

- Redundant Relationship

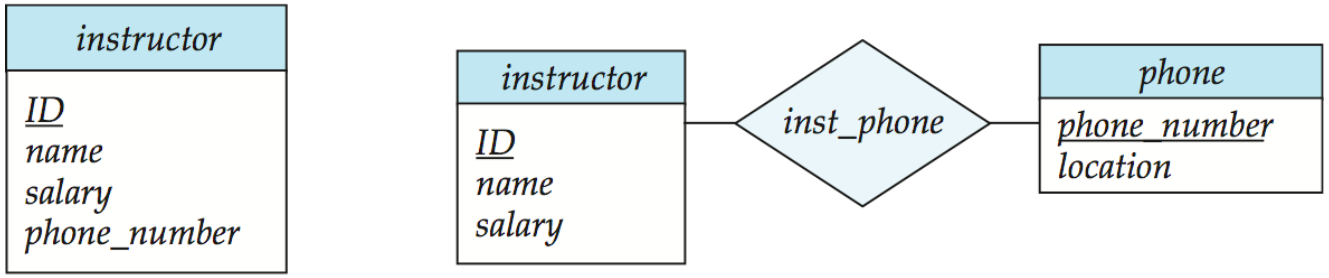


- 여러 Entity Set과 Relationship Set을 하나의 Entity Set으로 합침



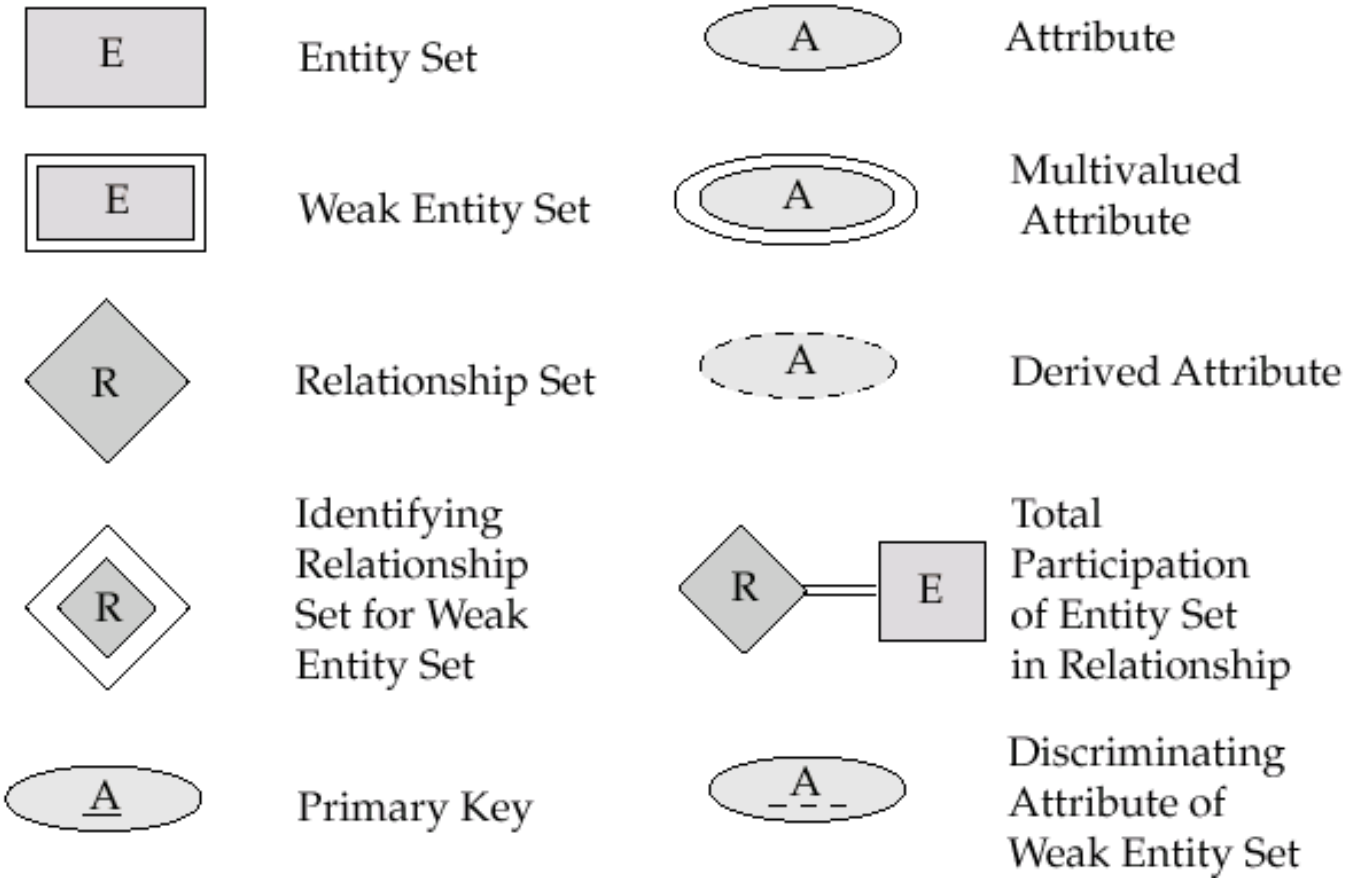
Design Issues

- Entity vs. Attribute

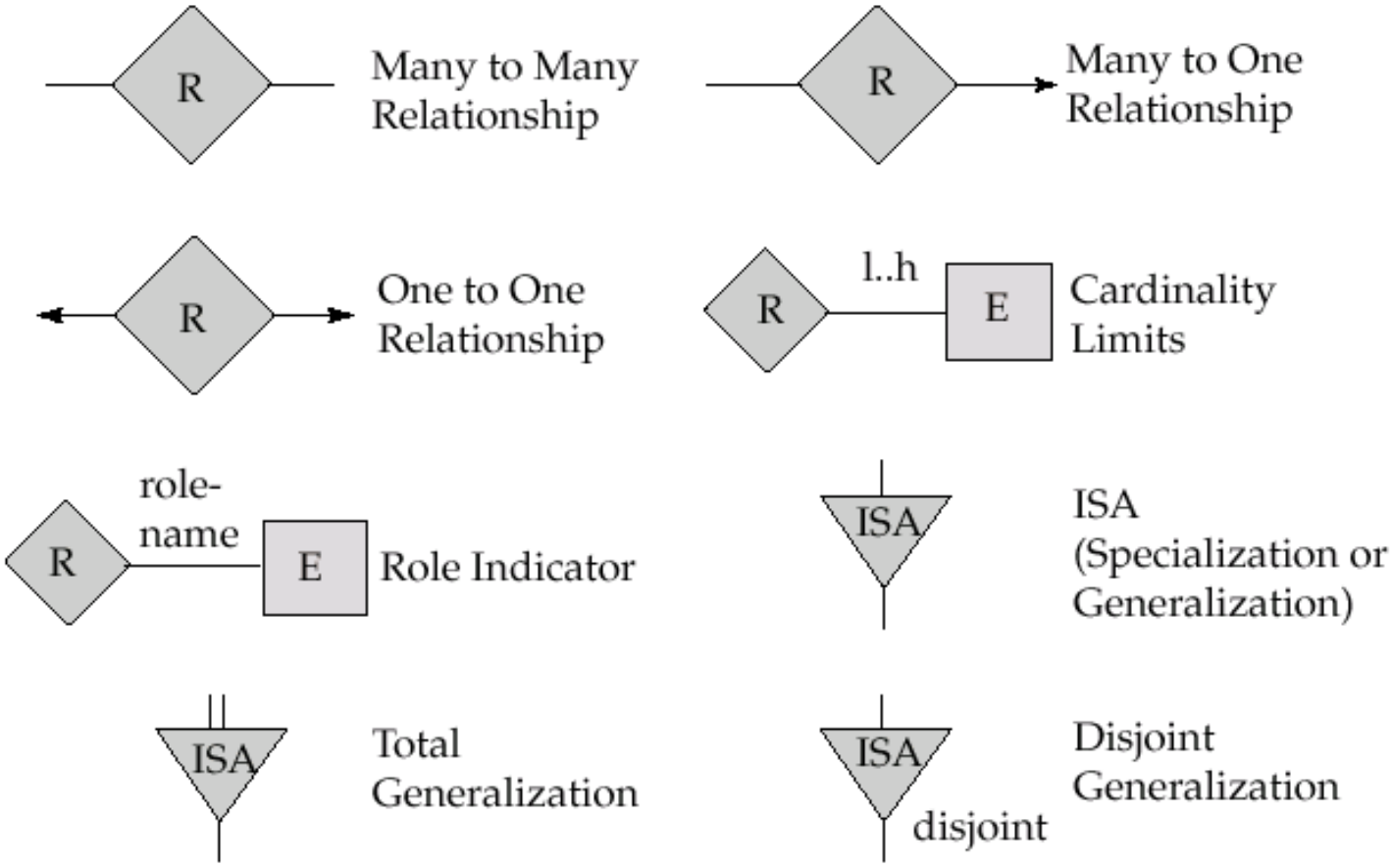


- Entity vs. Relationship
- Binary vs. n-ary Relationship

Summary of Symbols

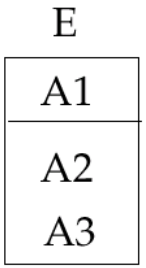


Summary of Symbols (Cont.)

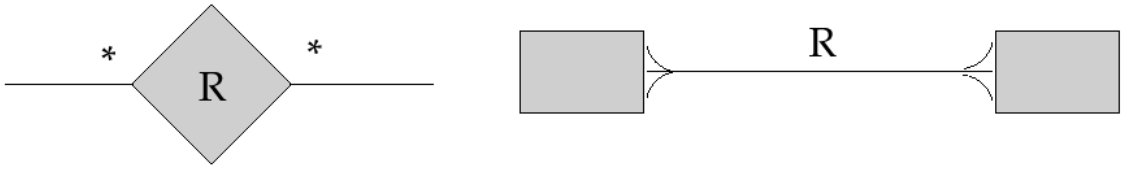


Alternative E-R Notations

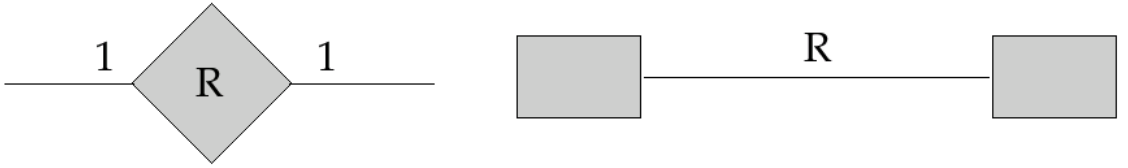
Entity set E with attributes A1, A2, A3 and primary key A1



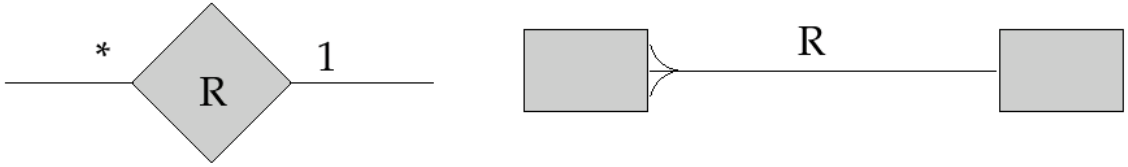
Many to Many Relationship



One to One Relationship

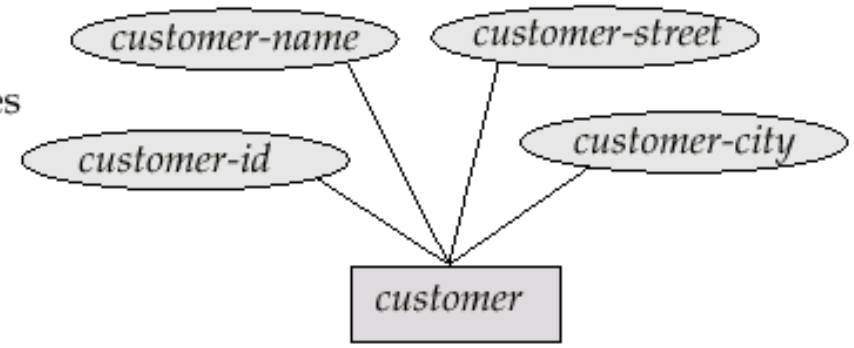


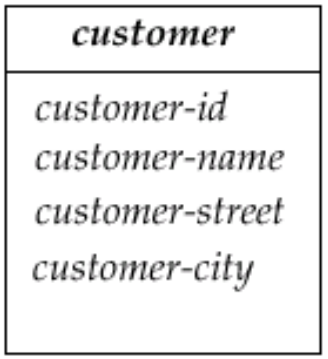
Many to One Relationship



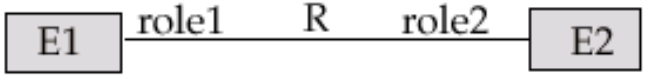
Alternative Notations: UML Class Diagram Notation

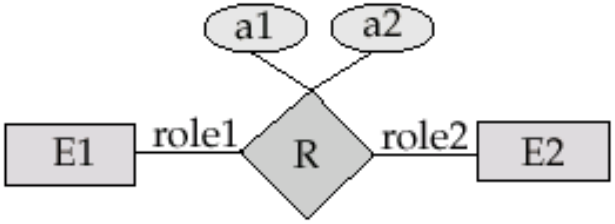
1. Entity sets and attributes

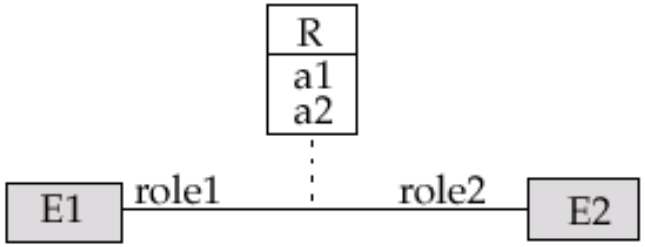




2. Relationships



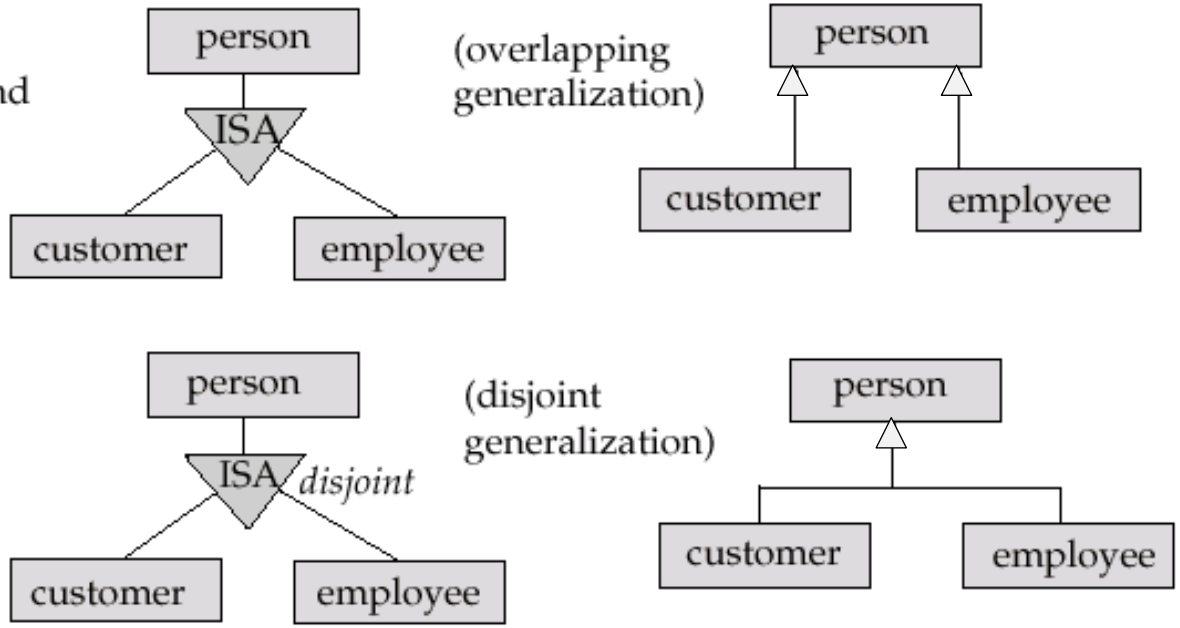




UML Class Diagram Notation (Cont.)

3. Cardinality constraints

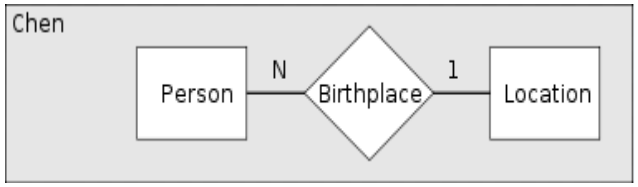
4. Generalization and Specialization

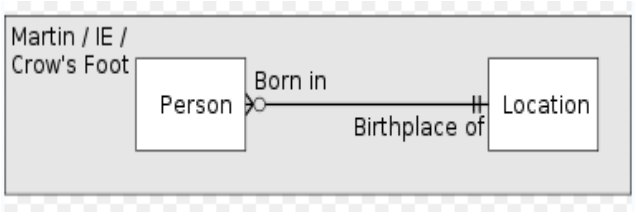


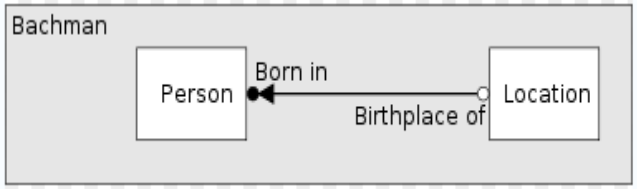
- Note reversal of position in cardinality constraint depiction!
- Think of customer & Loan case as an example.

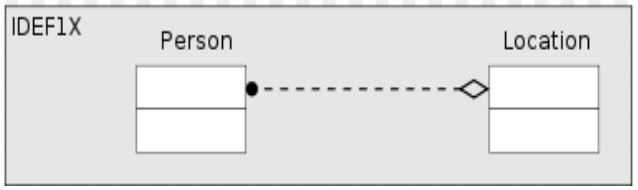
Other ERD Conventions

- Other ERD conventions do exist
  - **IE/Crow's Foot/Martin**
    - IE(Information Engineering)
    - 일반적으로 가장 널리 사용되는 방식
  - Bachman notation
  - IDEF1X(Integration DEFinition for Information Modeling)
    - IDEF0: functional model, IDEF1: Information/Data model
  - (min, max) notation and etc.









출처: Wikipedia

Crow's foot Cardinality/Modality

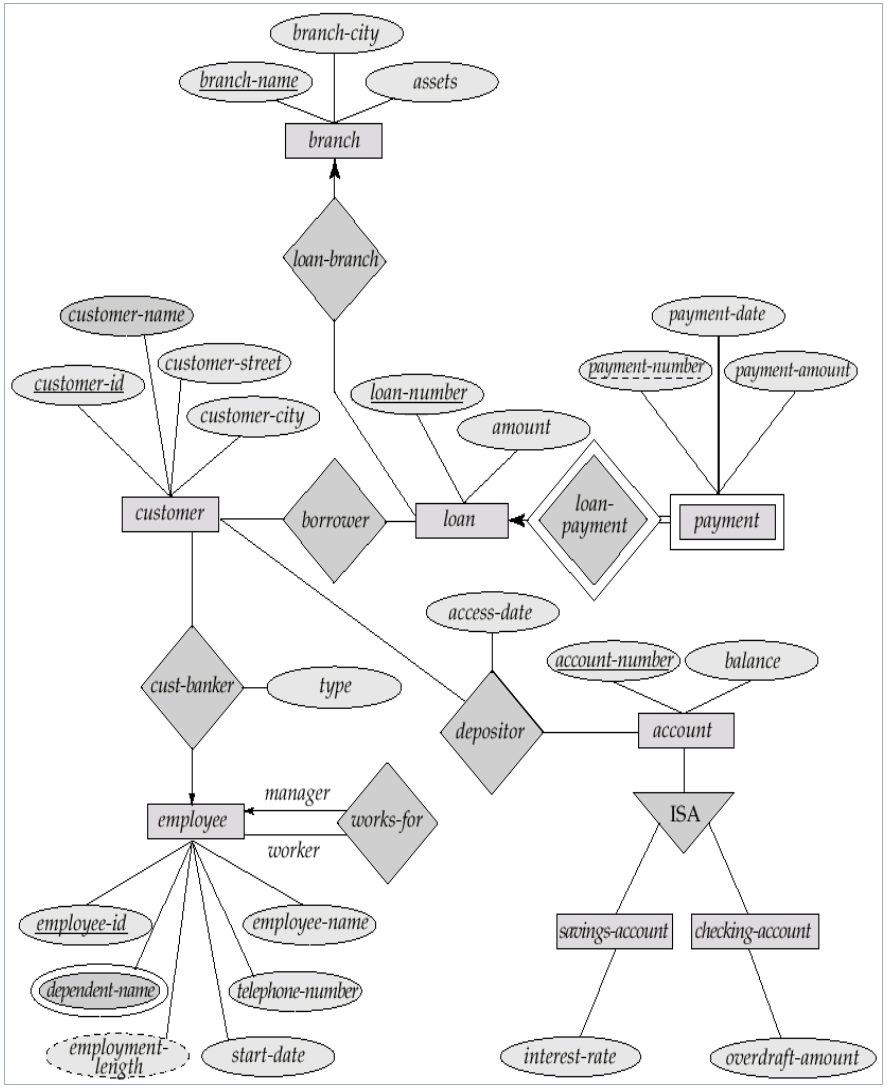
Zero or more (cardinality many, modality 0)

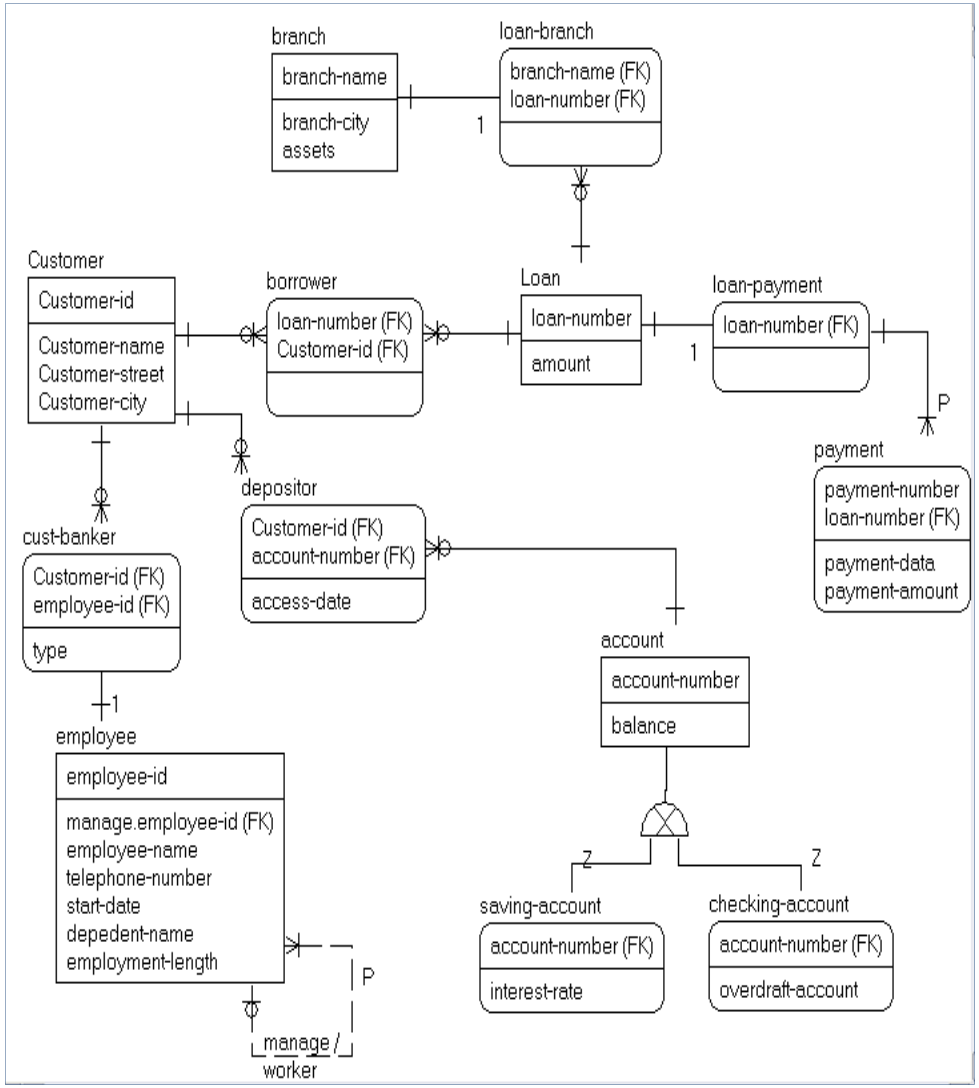
One or more (cardinality many, modality 1)

One and only one (cardinality 1, modality 1)

Zero or one (cardinality 1, modality 0)

Examples – ERwin (cont.)





Design of an ER DB Schema

- Decisions to be made
  - attribute or entity
  - entity set or relationship set
  - ternary relationship or binary relationship
  - strong or weak entity set
  - generalization and specialization
  - aggregation

Design of an ER DB Schema (Revisited)

- Design phases
  - Requirement specification
    - identify data needs of user
  - Conceptual design
    - translate into a conceptual schema
    - Usually ER model
  - Logical design
    - map onto the implementation data model of the DBMS
    - Usually relational model
  - Physical design
    - specify physical features of the database (issues pertaining to performance rather than information contents; index, sequential order, etc.)

  - Read through the Banking Enterprise example (pdf file @tahiti)

Reducing ER schema to tables

- Basic rule
  - each entity set => unique table
  - each relationship set => unique table
- Strong entity set E with attributes $a_1, \ldots, a_n$
  - table E with n distinct columns each of which corresponds to a_i

ER schema to tables (Cont.)

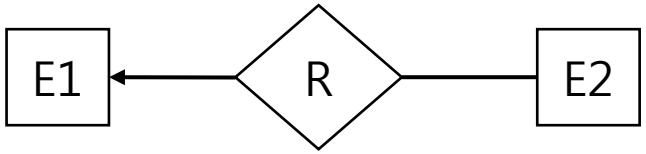
- A strong entity set reduces to a table with the same attributes.

customer-id	*customer-name*	*customer-street*	*customer-city*
019-28-3746	Smith	North	Rye
182-73-6091	Turner	Putnam	Stamford
192-83-7465	Johnson	Alma	Palo Alto
244-66-8800	Curry	North	Rye
321-12-3123	Jones	Main	Harrison
335-57-7991	Adams	Spring	Pittsfield
336-66-9999	Lindsay	Park	Pittsfield
677-89-9011	Hayes	Main	Harrison
963-96-3963	Williams	Nassau	Princeton

ER schema to tables (Cont.)

- Relationship set R
  - involving $E_1, \ldots, E_k$
    => table R with columns corresponding to
    $PK(E_1) \cup \ldots \cup PK(E_k) \cup attr(R)$

  - if one-to-many or one-to-one



    => add columns representing $PK(E_1) \cup attr(R)$

    to table representing E_2

Representing Weak Entity Sets

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

loan-number	*payment-number*	*payment-date*	*payment-amount*
L-11	53	7 June 2001	125
L-14	69	28 May 2001	500
L-15	22	23 May 2001	300
L-16	58	18 June 2001	135
L-17	5	10 May 2001	50
L-17	6	7 June 2001	50
L-17	7	17 June 2001	100
L-23	11	17 May 2001	75
L-93	103	3 June 2001	900
L-93	104	13 June 2001	200

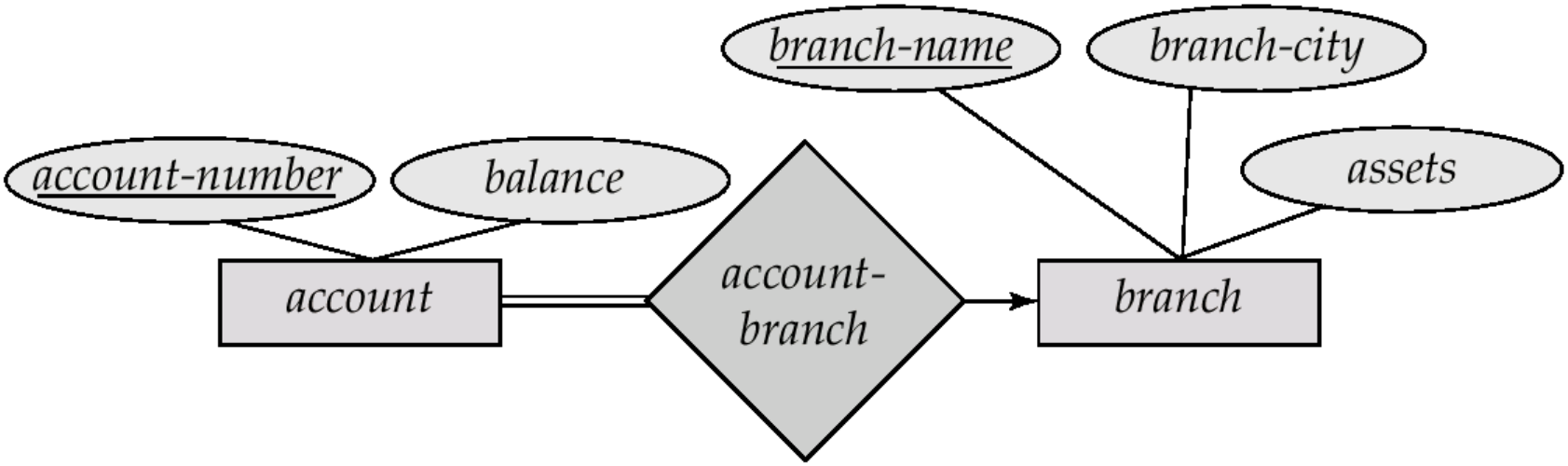
Representing Relationship Sets as Tables

- many-to-many relationship set:
  as a table with the primary keys of the two participating entity sets, + any descriptive attributes

customer-id	*loan-number*
019-28-3746	L-11
019-28-3746	L-23
244-66-8800	L-93
321-12-3123	L-17
335-57-7991	L-16
555-55-5555	L-14
677-89-9011	L-15
963-96-3963	L-17

Redundancy of Tables

- Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the many side, containing the primary key of the one side



Generalization

- Account <= savings, checking
- 방법1: General Case
  - 각각 별도의 relation으로 (sub-entity는 super의 PK를 FK로)
  - E.g., account(acc-number, balance),
    savings(acc-number, int-rate),
    checking(acc-number, overdraft-amt)
- 방법2: If disjoint and complete
  - 모두 Subclass로!
  - E.g., savings(acc-number, balance, int-rate),
  - checking(acc-number, balance, overdraft-amt)
- 방법3: 대부분의 attribute가 중복일 경우
  - 모두 Superclass로!
  - 경우에 따라 sub class의 type을 attribute로