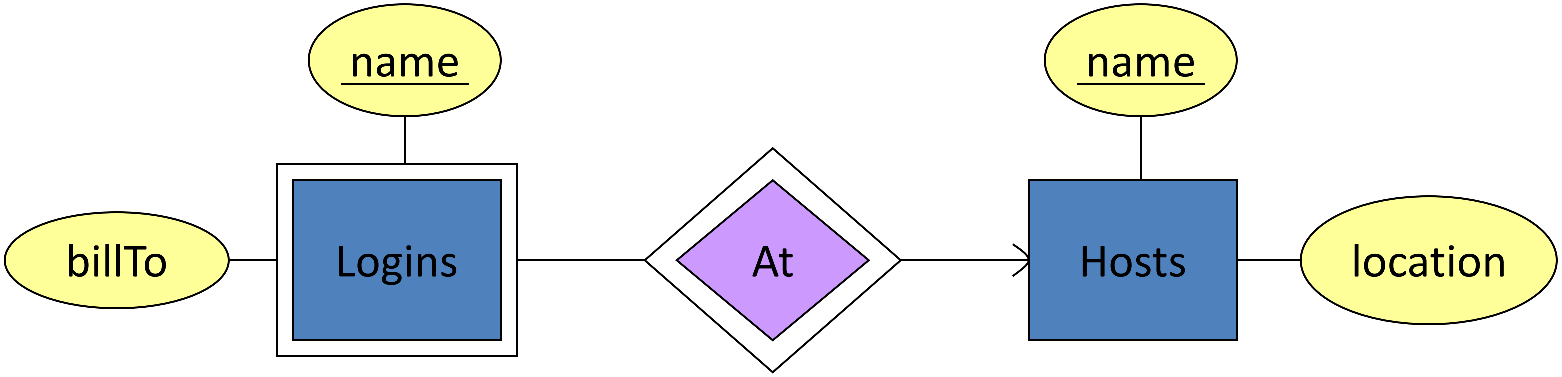
Entity-Relationship Model

1. Entity
2. Attributes
3. Entity Sets
4. Relationship Sets
5. Design Issues
6. Mapping Constraints
7. Weak Entity
8. Keys
9. E-R Diagram
10. Extended E-R Features
11. Design of an E-R Database Schema
12. Reduction of an E-R Schema to Tables

Example: Weak Entity Set -> Relation



Hosts(hostName, location)

Logins(loginName, hostName, billTo)

~~At(loginName, hostName, hostName2)~~

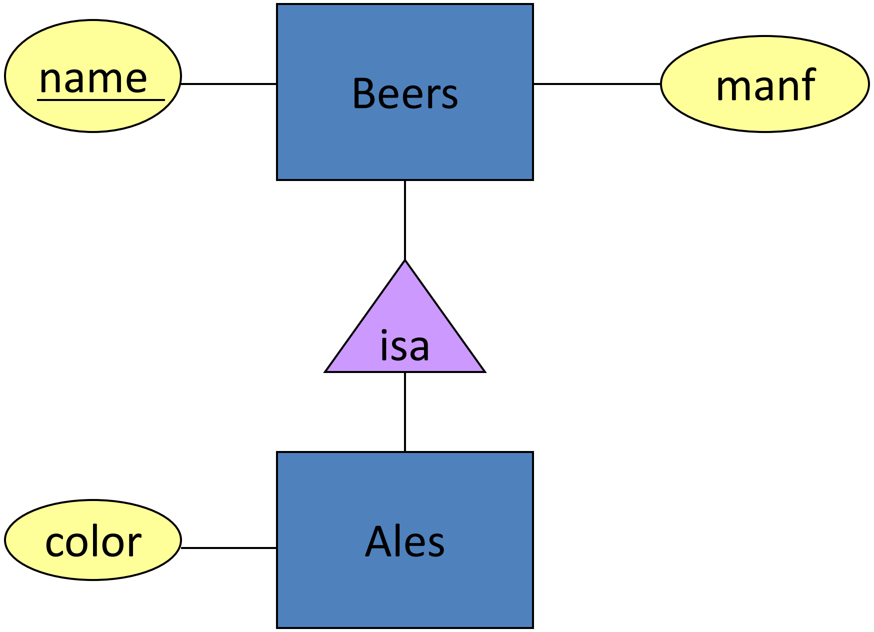
At becomes part of Logins

Must be the same

Subclasses: Three Approaches

1. *Object-oriented* : One relation per subset of subclasses, with all relevant attributes.
2. *Use nulls* : One relation; entities have NULL in attributes that don't belong to them.
3. *E/R style* : One relation for each subclass:
   - Key attribute(s).
   - Attributes of that subclass.

Example: Subclass -> Relations



Object-Oriented

name	manf
Bud	Anheuser-Busch

Beers

name	manf	color
Summerbrew	Pete’s	dark

Ales

Good for queries like “find the color of ales made by Pete’s.”

E/R Style

name	manf
Bud	Anheuser-Busch
Summerbrew	Pete’s

Beers

name	color
Summerbrew	dark

Ales

Good for queries like “find all beers (including ales) made by Pete’s.”

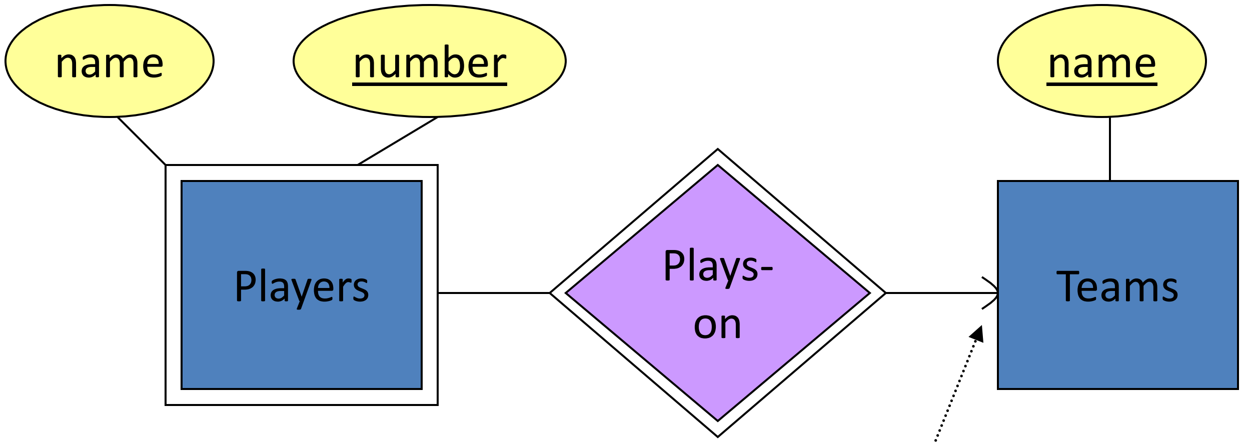
Using Nulls

name	manf	color
Bud	Anheuser-Busch	NULL
Summerbrew	Pete’s	dark

Beers

Saves space unless there are *lots*
of attributes that are usually NULL.

In E/R Diagrams



Note: must be rounded because each player needs a team to help with the key.

- Double diamond for *supporting* many-one relationship.
- Double rectangle for the weak entity set.

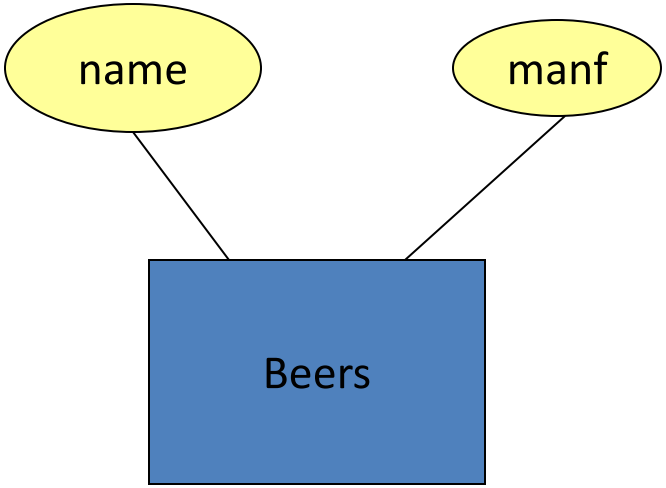
E/R Diagrams in a General

- In an entity-relationship diagram:
  - Entity set = rectangle.
  - Attribute = oval, with a line to the rectangle representing its entity set.

E-R Design Decisions

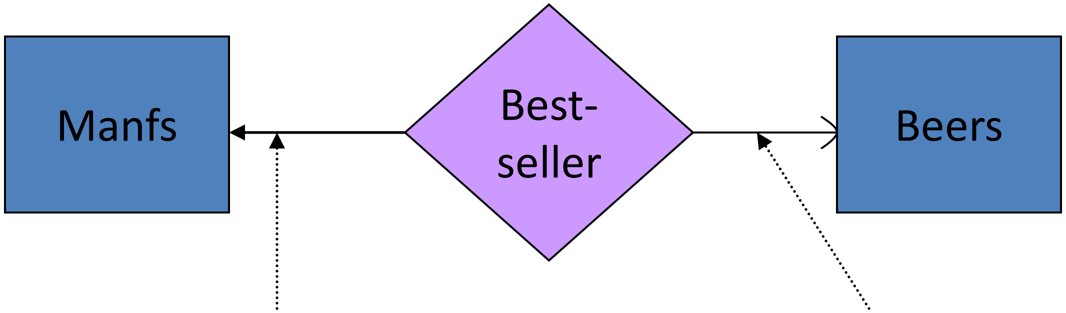
- The use of an attribute or entity set to represent an object.
- Whether a real-world concept is best expressed by an entity set or a relationship set.
- The use of a ternary relationship versus a pair of binary relationships.
- The use of a strong or weak entity set.
- The use of specialization/generalization – contributes to modularity in the design.
- The use of aggregation – can treat the aggregate entity set as a single unit without concern for the details of its internal structure.

Example:



- Entity set Beers has two attributes, name and manf (manufacturer).
- Each Beers entity has values for these two attributes, e.g. (Bud, Anheuser-Busch)

First Look E/R Diagram



A beer is the best-seller for 0 or 1 manufacturer. **(CARDINALITY)**

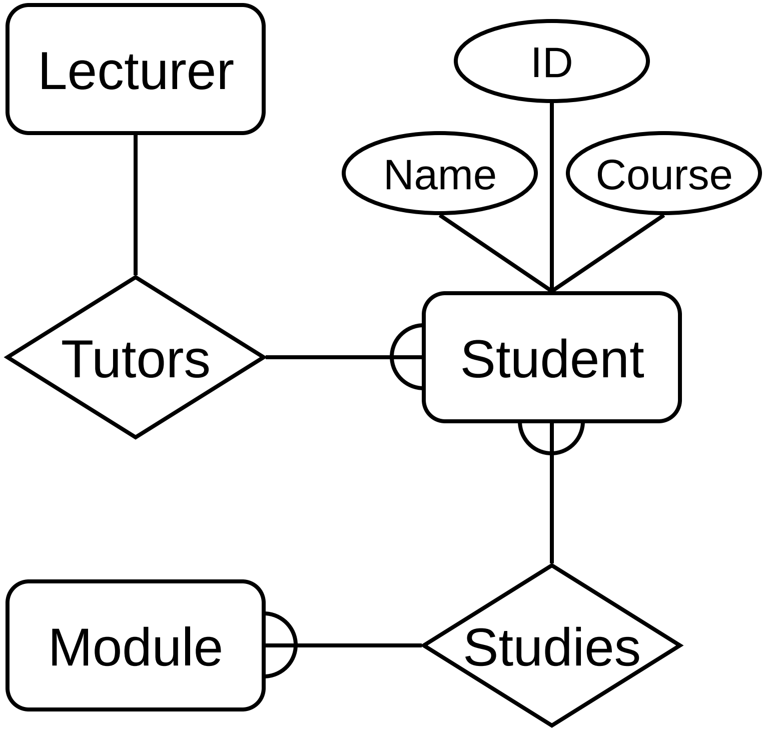
A manufacturer has exactly one best seller. **(CARDINALITY)**

Caution: Reduction of an E-R Schema to Tables

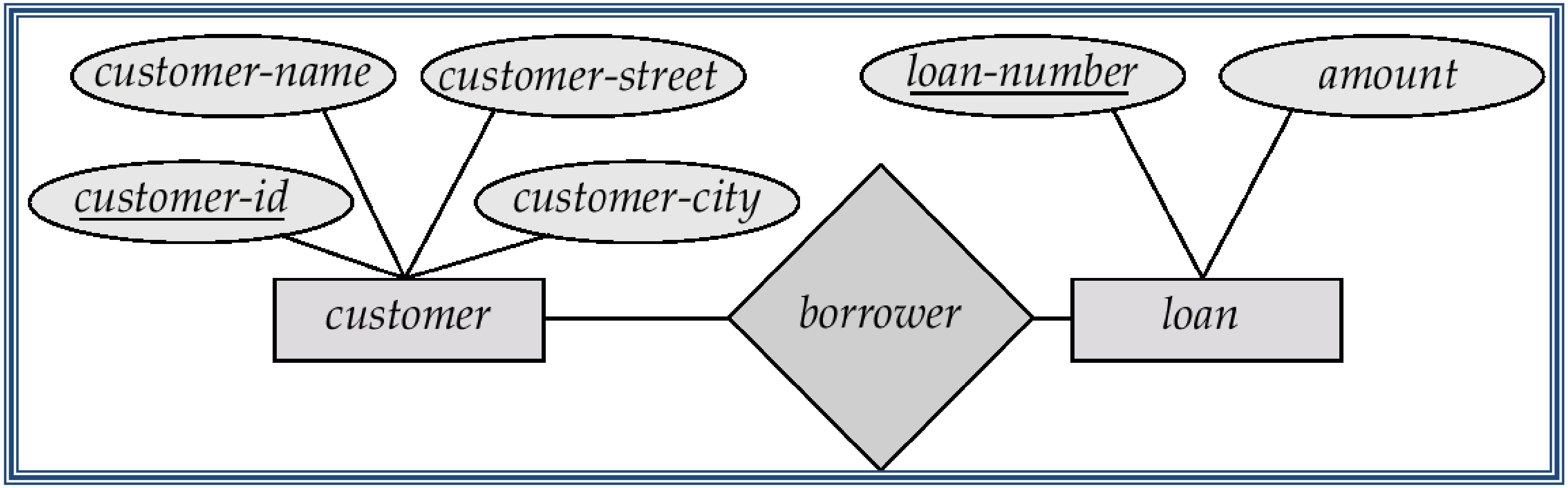
- Primary keys allow entity sets and relationship sets to be expressed uniformly as *tables* which represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of tables.
- For each entity set and relationship set there is a unique table which is assigned the name of the corresponding entity set or relationship set.
- Each table has a number of columns (generally corresponding to attributes), which have unique names.
- Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram.

Entity/Relationship Diagrams

- E/R Models are often represented as E/R diagrams that
  - Give a conceptual view of the database
  - Are independent of the choice of DBMS
  - Can identify some problems in a design

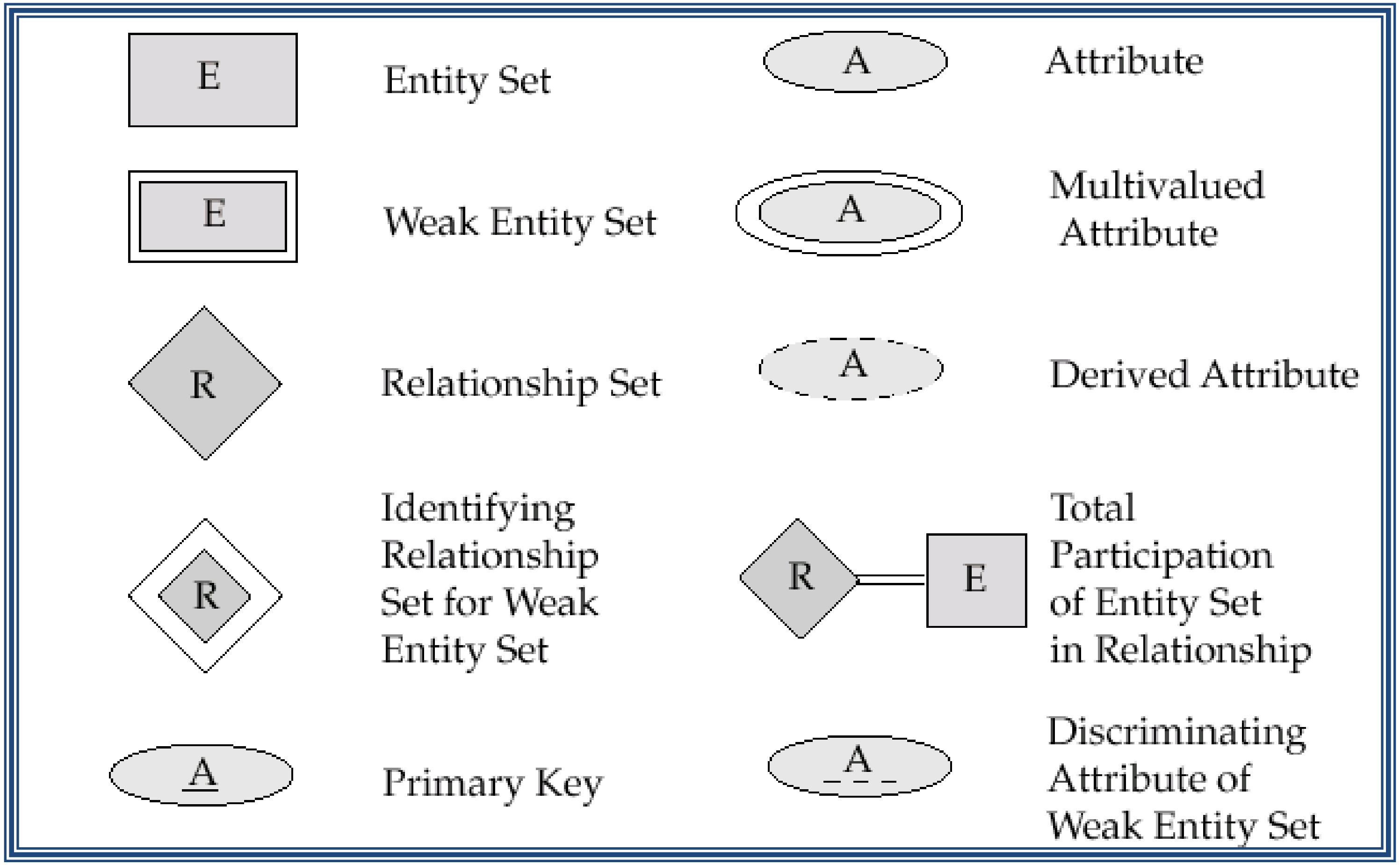


E-R Diagrams Contd..

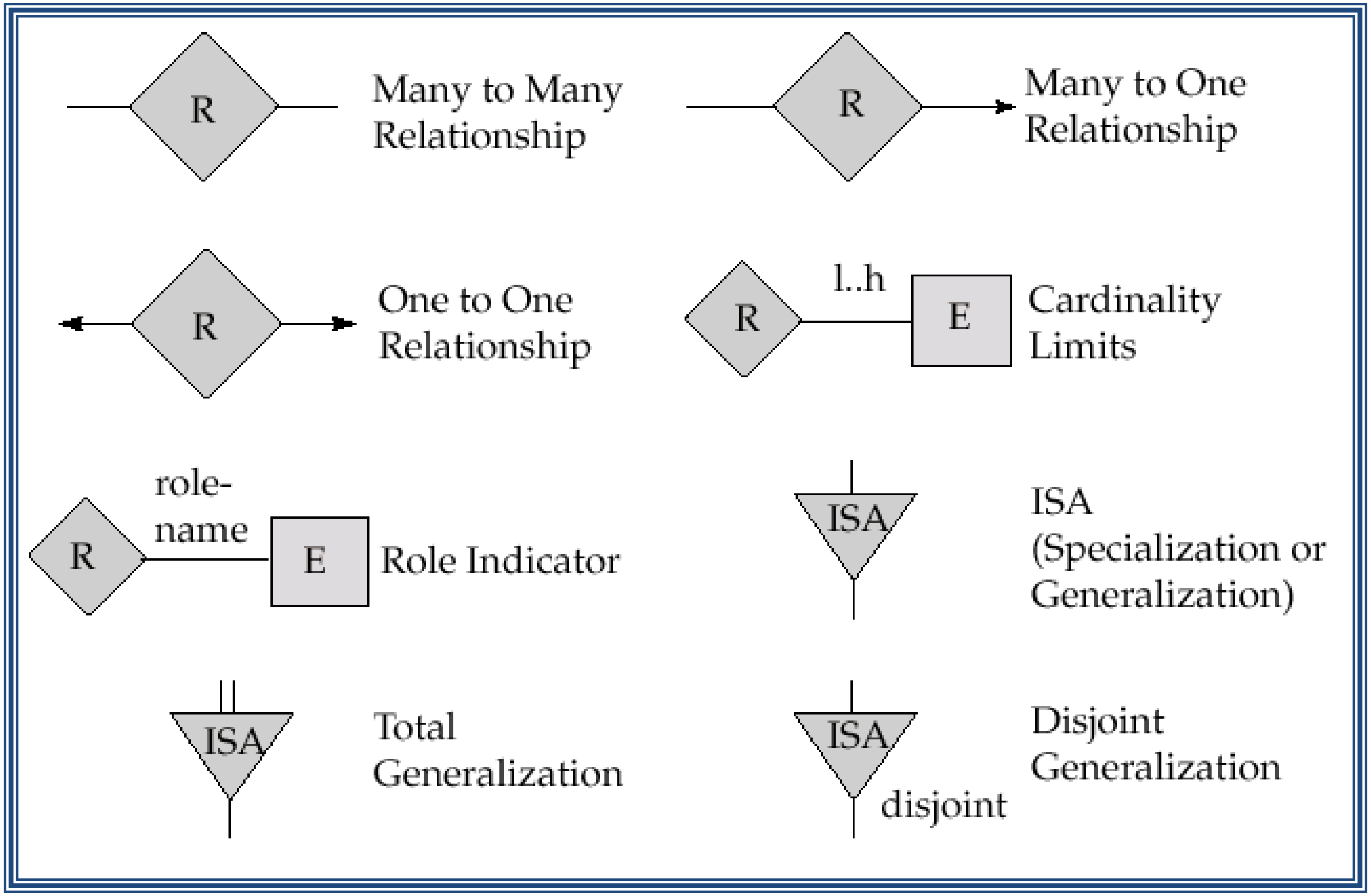


- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Ellipses** represent attributes
  - **Double ellipses** represent multivalued attributes.
  - **Dashed ellipses** denote derived attributes.
- **Underline** indicates primary key attributes (will study later)

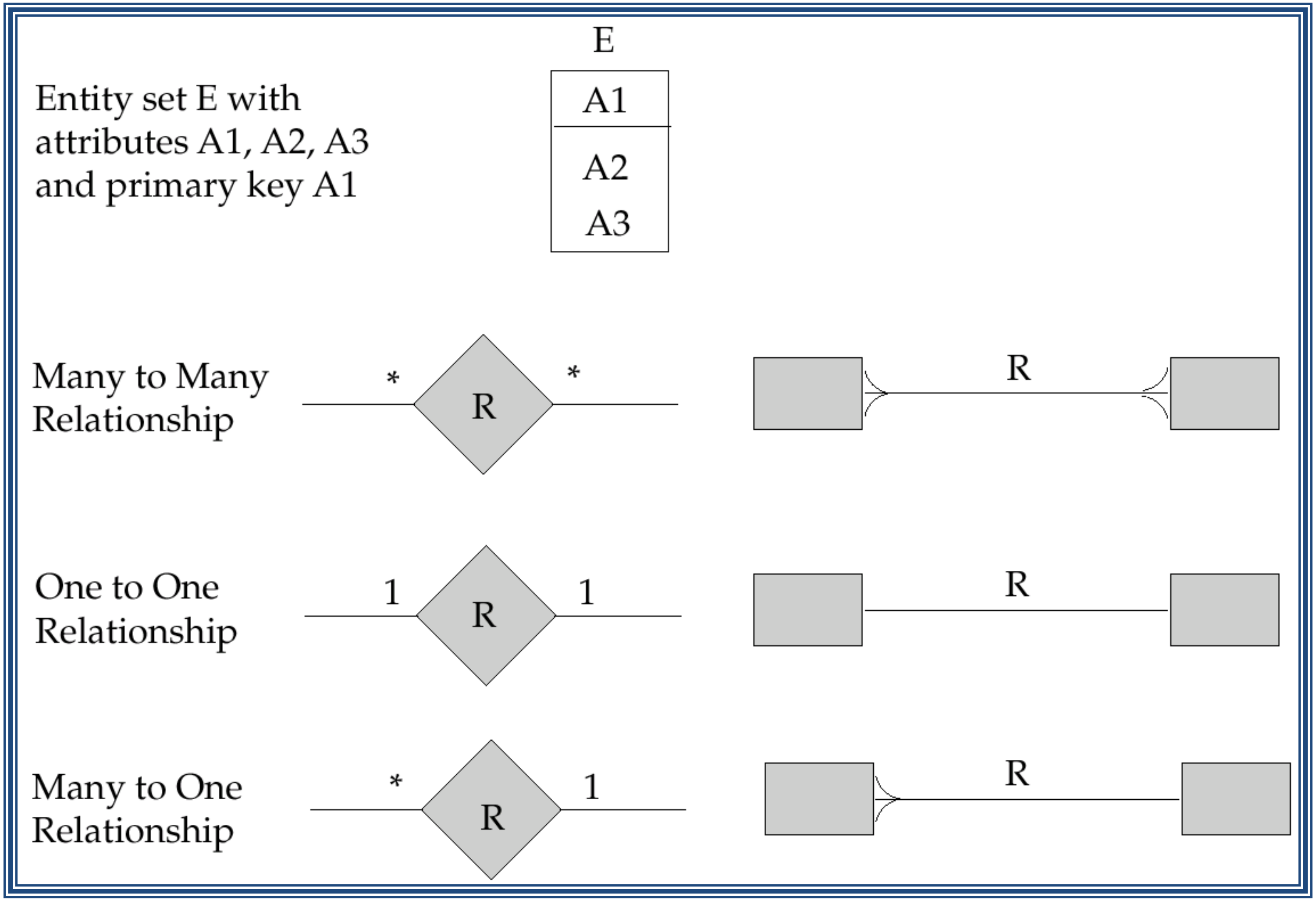
Summary of Symbols Used in E-R Notation


E
Entity Set
A
Attribute
E
Weak Entity Set
A
Multivalued Attribute
R
Relationship Set
A
Derived Attribute
R
Identifying Relationship Set for Weak Entity Set
R
E
Total Participation of Entity Set in Relationship
A
Primary Key
A
Discriminating Attribute of Weak Entity Set

Summary of Symbols (Cont.)



Alternative E-R Notations



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

Many to Many Relationship

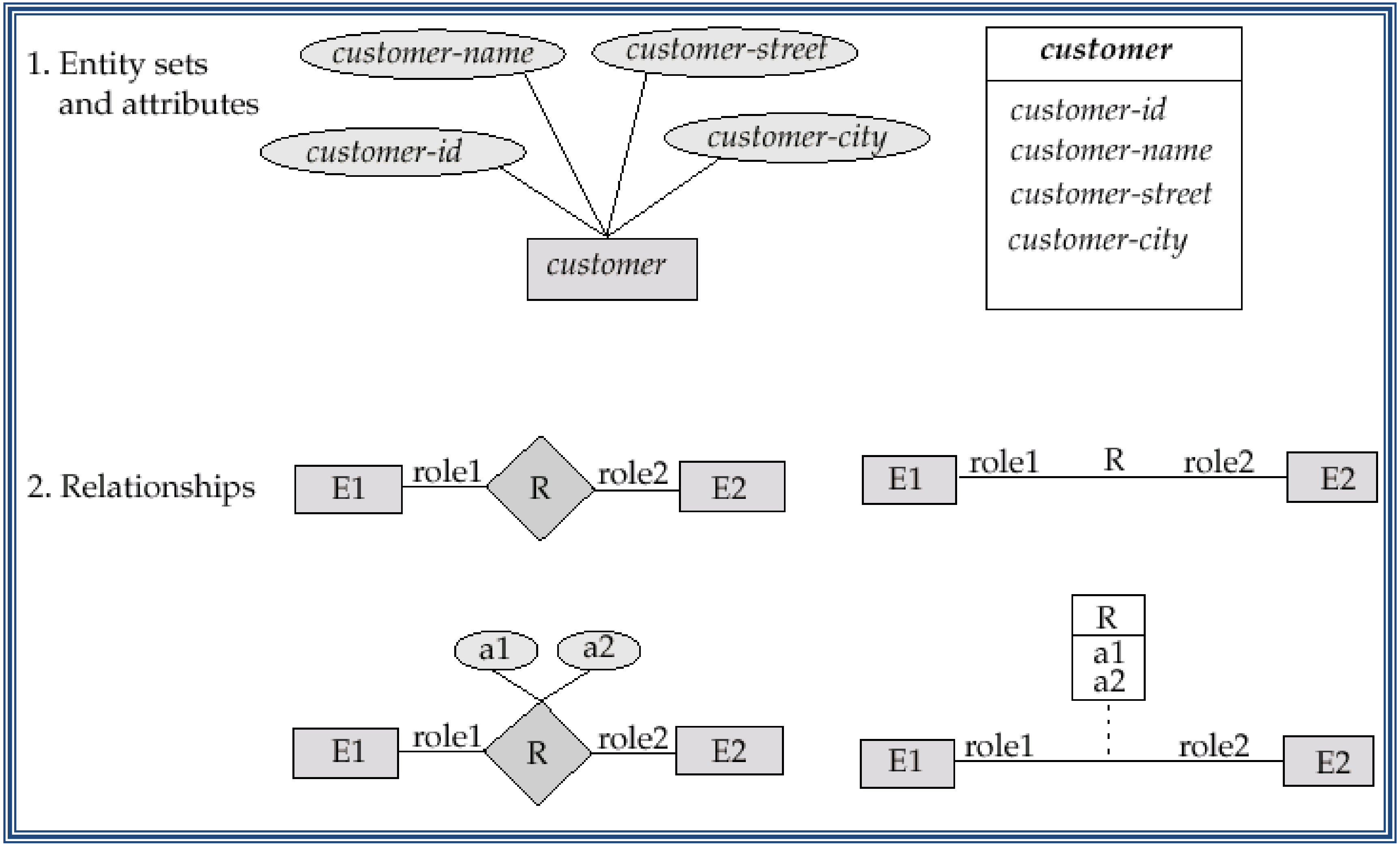
One to One Relationship

Many to One Relationship

UML

- UML: Unified Modeling Language
- UML has many components to graphically model different aspects of an entire software system
- UML Class Diagrams correspond to E-R Diagram, but several differences.

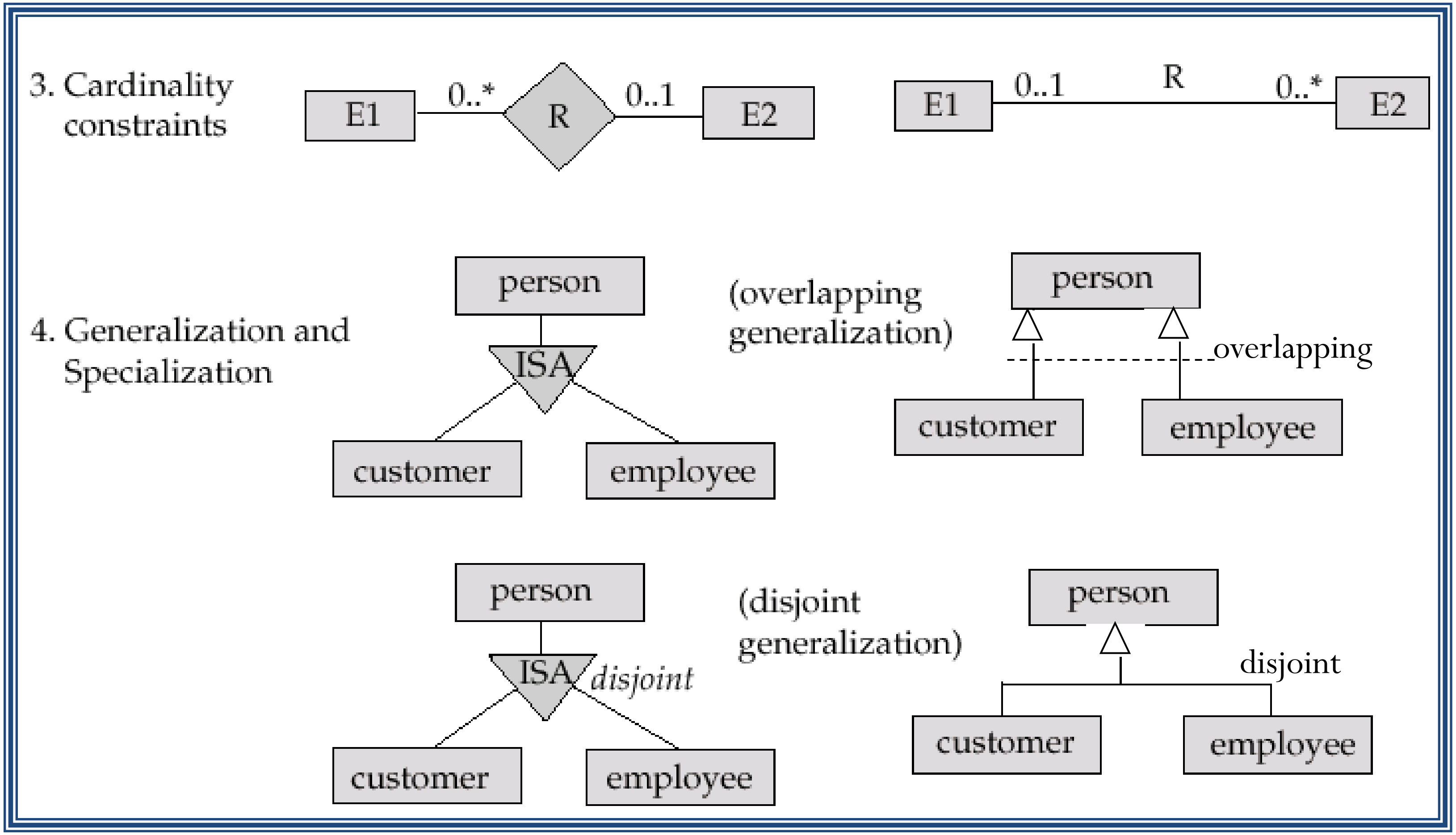
Summary of UML Class Diagram Notation



UML Class Diagrams (Contd.)

- Entity sets are shown as boxes, and attributes are shown within the box, rather than as separate ellipses in E-R diagrams.
- Binary relationship sets are represented in UML by just drawing a line connecting the entity sets. The relationship set name is written adjacent to the line.
- The role played by an entity set in a relationship set may also be specified by writing the role name on the line, adjacent to the entity set.
- The relationship set name may alternatively be written in a box, along with attributes of the relationship set, and the box is connected, using a dotted line, to the line depicting the relationship set.
- Non-binary relationships drawn using diamonds, just as in ER diagrams

UML Class Diagram Notation (Cont.)



*Note reversal of position in cardinality constraint depiction

*Generalization can use merged or separate arrows independent of disjoint/overlapping

UML Class Diagrams (Contd.)

- Cardinality constraints are specified in the form *l*..*h*, where *l* denotes the minimum and *h* the maximum number of relationships an entity can participate in.
- Beware: the positioning of the constraints is exactly the reverse of the positioning of constraints in E-R diagrams.
- The constraint 0..* on the *E*2 side and 0..1 on the *E*1 side means that each *E*2 entity can participate in at most one relationship, whereas each *E*1 entity can participate in many relationships; in other words, the relationship is many to one from *E*2 to *E*1.
- Single values, such as 1 or * may be written on edges; The single value 1 on an edge is treated as equivalent to 1..1, while * is equivalent to 0..*.