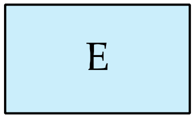
E-R Notations
UML Models

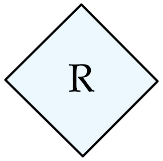
E-R Design Decisions

- 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.

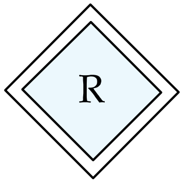
Summary of Symbols Used in E-R Notation



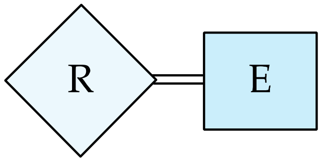
entity set



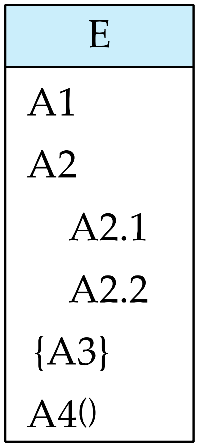
relationship set



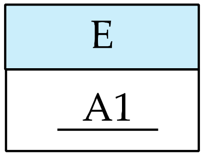
identifying
relationship set
for weak entity set



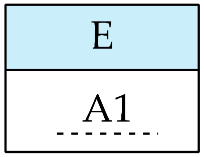
total participation
of entity set in
relationship



attributes:
simple (A1),
composite (A2) and
multivalued (A3)
derived (A4)

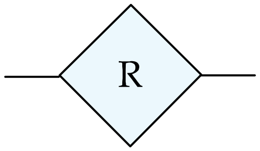


primary key

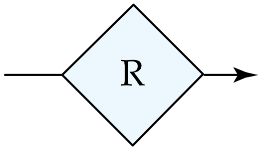


discriminating
attribute of
weak entity set

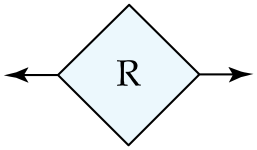
Symbols Used in E-R Notation



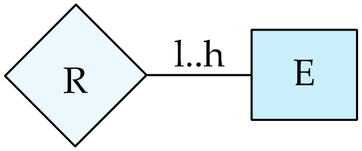
many-to-many
relationship



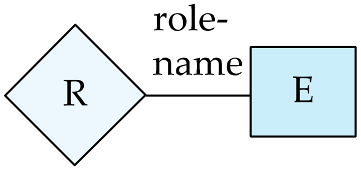
many-to-one
relationship



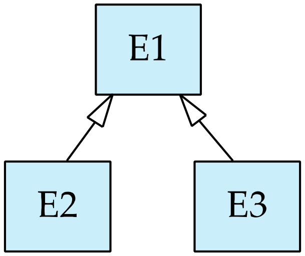
one-to-one
relationship



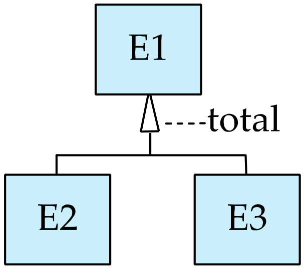
cardinality
limits



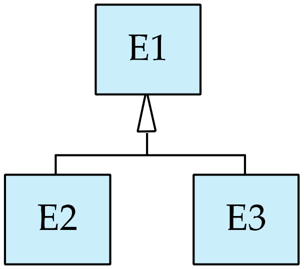
role indicator



ISA: generalization
or specialization



total (disjoint)
generalization

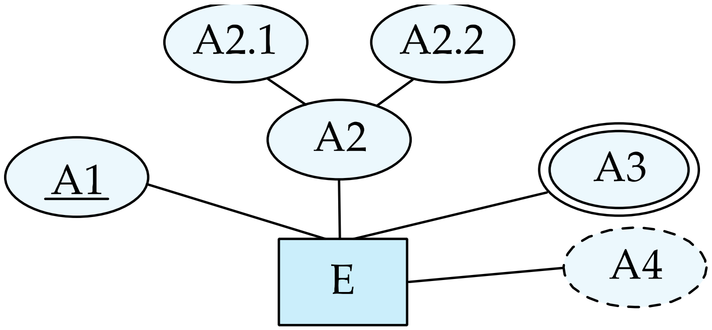


disjoint
generalization

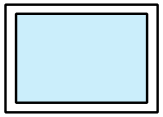
Alternative ER Notations

- Chen, IDE1FX, …

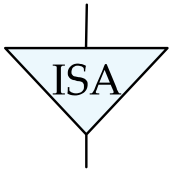
entity set E with
simple attribute A1,
composite attribute A2,
multivalued attribute A3,
derived attribute A4,
and primary key A1



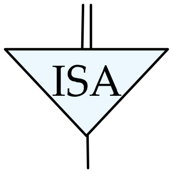
weak entity set



generalization

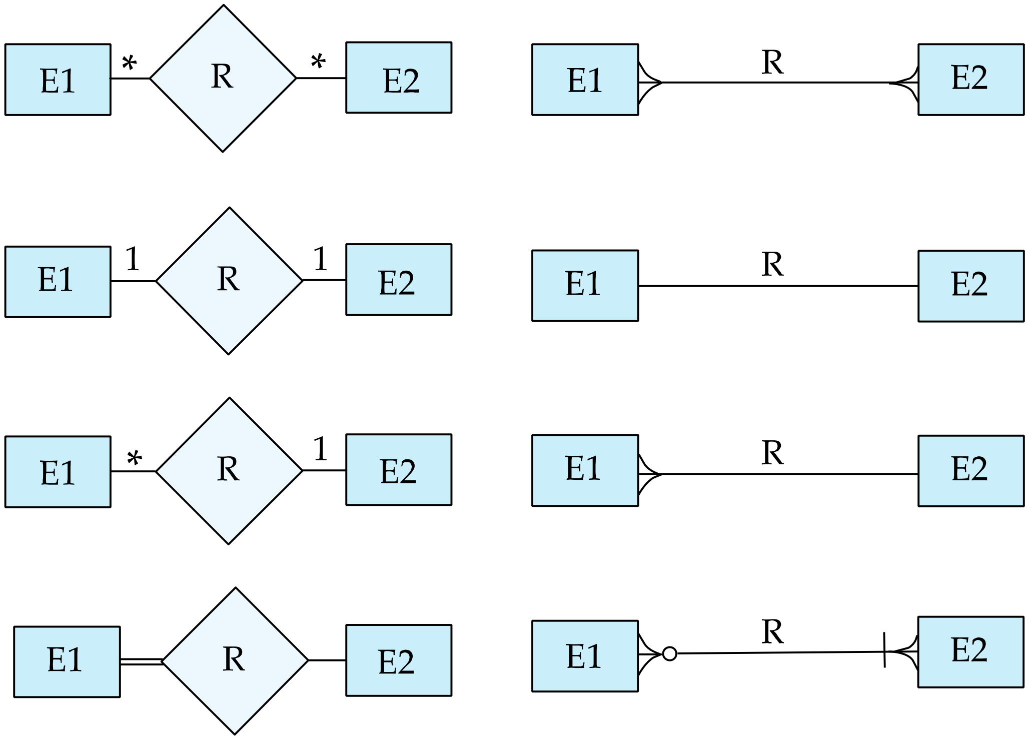


total
generalization



Alternative ER Notations

Chen **IDE1FX (Crows feet notation)**



many-to-many relationship

one-to-one relationship

many-to-one relationship

participation
in R: total (E1)
and partial (E2)

UML

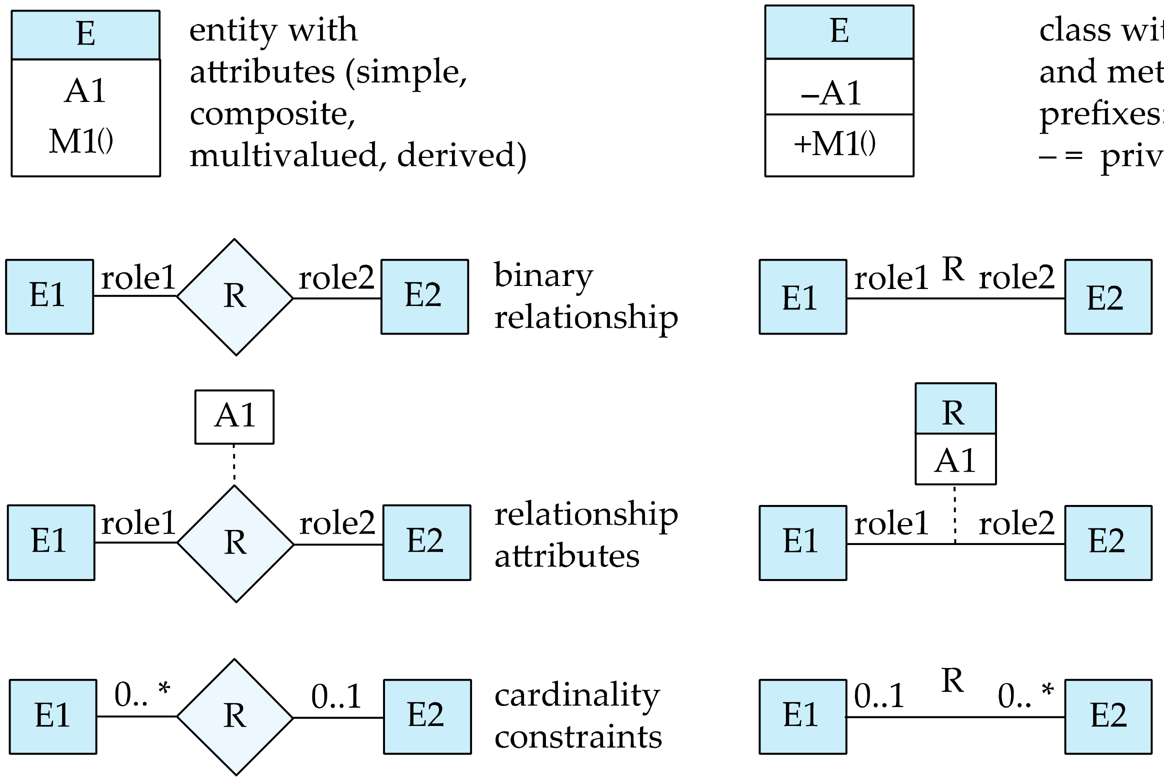
- Data representation, user interactions with the system, specification of functional modules of the system and their interaction
- Software specification language by OMG

- **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.

- **Class diagram.** A class diagram is similar to an E-R diagram, they relate to E-R diagrams.
- **Use case diagram**. Use case diagrams show the interaction between users and the system, in particular the steps of tasks that users perform (such as withdrawing money or registering for a course).
- **Activity diagram**. Activity diagrams depict the flow of tasks between various components of a system.

ER vs. UML Class Diagrams

ER Diagram Notation | **Equivalent in UML**



entity with attributes (simple, composite, multivalued, derived)

class with simple attributes and methods (attribute prefixes: + = public, – = private, # = protected)

binary relationship

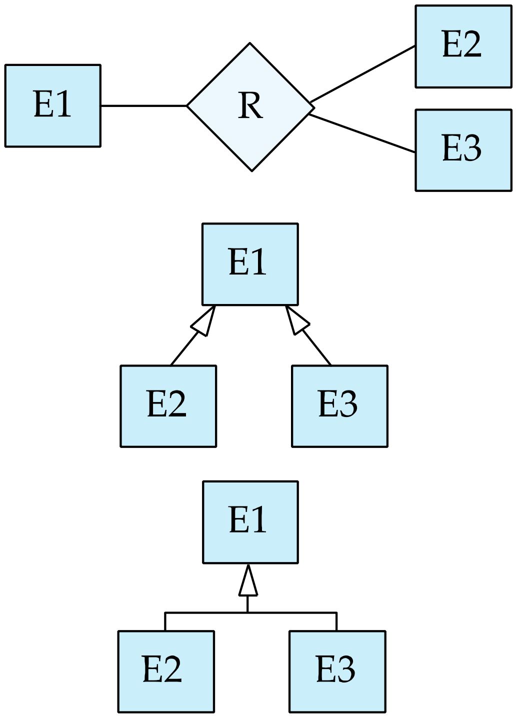
relationship attributes

cardinality constraints

* Note reversal of position in cardinality constraint depiction

ER vs. UML Class Diagrams

ER Diagram Notation

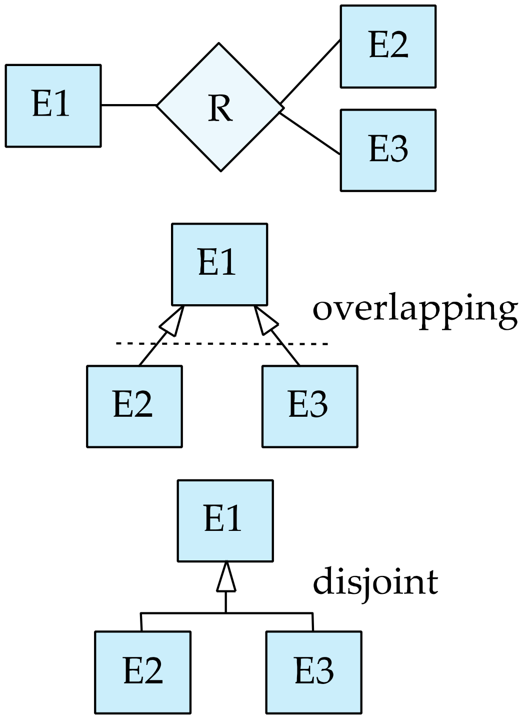


n-ary relationships

overlapping generalization

disjoint generalization

Equivalent in UML

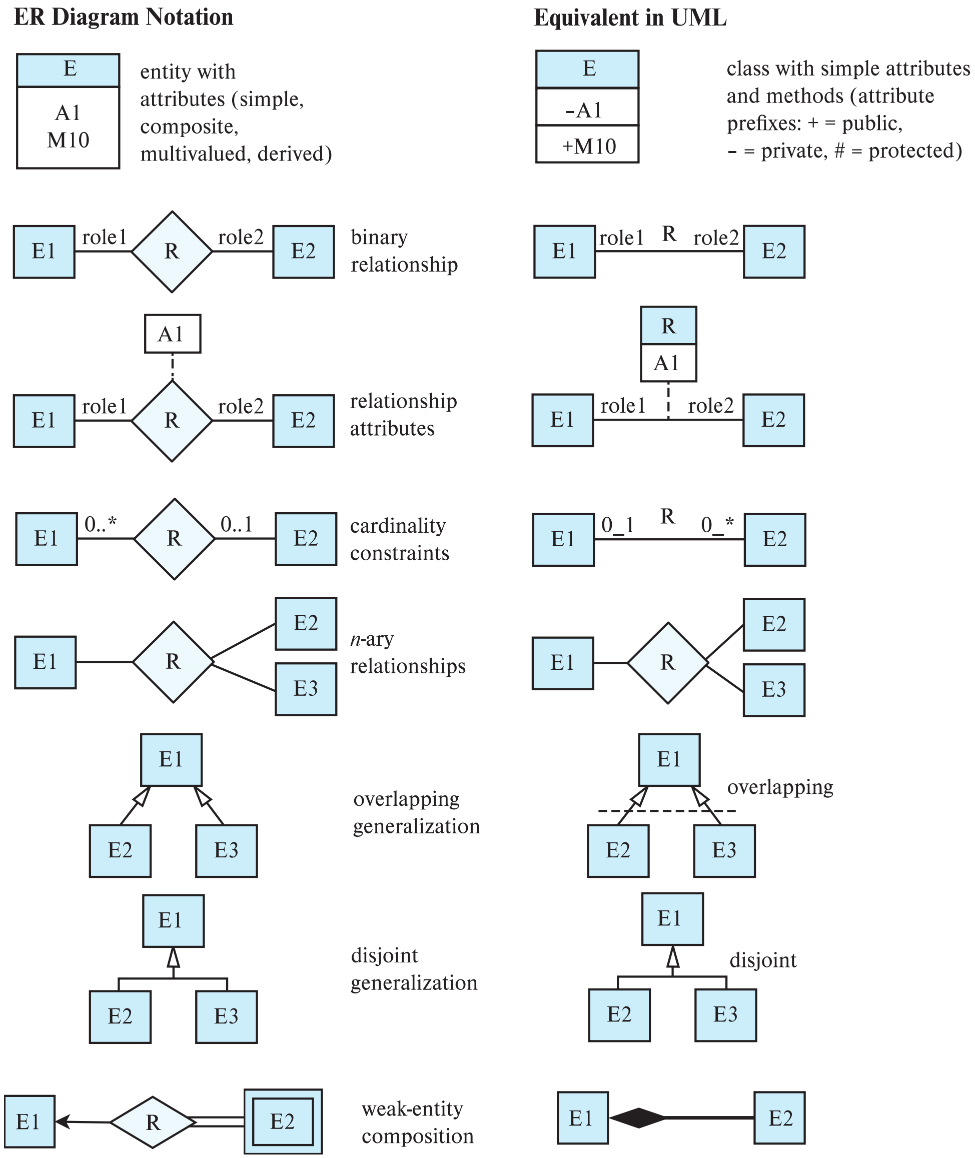


* Generalization can use merged or separate arrows independent of disjoint/overlapping

UML Class 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.

ER vs. UML Class Diagrams



ER Diagram Notation

Equivalent in UML

entity with attributes (simple, composite, multivalued, derived)

class with simple attributes and methods (attribute prefixes: + = public, – = private, # = protected)

binary relationship

relationship attributes

cardinality constraints

n-ary relationships

overlapping generalization

overlapping

disjoint generalization

disjoint

weak-entity composition

Other Aspects of Database Design

- Functional Requirements
- Data Flow, Workflow
- Schema Evolution (fundamental, temporary constraints)