E/R Design

CSE462 Database Concepts

Demian Lessa

Department of Computer Science and Engineering
State University of New York, Buffalo

March 21 – April 01, 2011

Outline

High-Level Design

Types of questions we must answer during high-level design.

- What information needs to be stored?
- How do information elements relate to each other?
- Which constraints should be assumed?

High-Level Design

Design approach.

- A conceptual model of the data is produced.
- The model is a formal notation.
- Not used directly for database implementation.
- Conceptual models are mapped to logical ones (e.g., logical schema).
- Logical models are mapped to physical ones (e.g., physical schema).

Next...

1. E/R Design
   - Entity/Relationship Model
   - Design Principles
   - Constraints
   - Weak Entity Sets
   - E/R Notation Summary
   - Mapping to Relational Schemas

Entity/Relationship Model

- Entity/Relationship (E/R) is one of many high-level design approaches.
- Others: Unified Modeling Language, Object Definition Language, etc.
- ER has many variants, each with slightly different notation.
- We will follow the one in the book, with a few extensions.
- The structure of the data is represented graphically as an E/R Diagram.
- E/R diagrams consists mainly of: entity sets, attributes, relationships.

Entity and Entity Set

Definition: Entity and Entity Set

An entity is an object of some sort, somewhat similar to objects in an OOPL. A collection of similar entities is an entity set, and its role is analogous to that of classes in an OOPL. Entity sets are usually implemented as relations in the relational model, however, not all relations in the final schema correspond to entity sets.

Attribute

Definition: Attribute

Entity sets have associated attributes, which are the individual properties of every entity in that set. Attribute types can be one of the following:

- primitive (e.g., atomic values such as strings or integers)
- composite (e.g., tuples with primitive components)
- set of values (e.g., set of some primitive or composite)
- derived (e.g., age is derived from date of birth)

Only primitive attributes are allowed in relational schemas. Attributes of an entity usually map to attributes in a relation, but that is not a requirement.

Relationship

Definition: Relationship

A relationship is a connection among two or more entity sets. Binary relationships involve exactly two entity sets and are by far the most common. In principle, however, a relationship may involve any number of entity sets. Some relationships are implemented as relations, others as attributes.

ER Diagram

Graphical representation: ER Diagrams.

- Entity sets are represented as rectangles.
- Attributes are represented as ovals.
- Relationships are represented as diamonds.
- Edges connect entity sets to their attributes.
- Edges also connect relationships to their entity sets (and attributes).

Example: ER Diagram

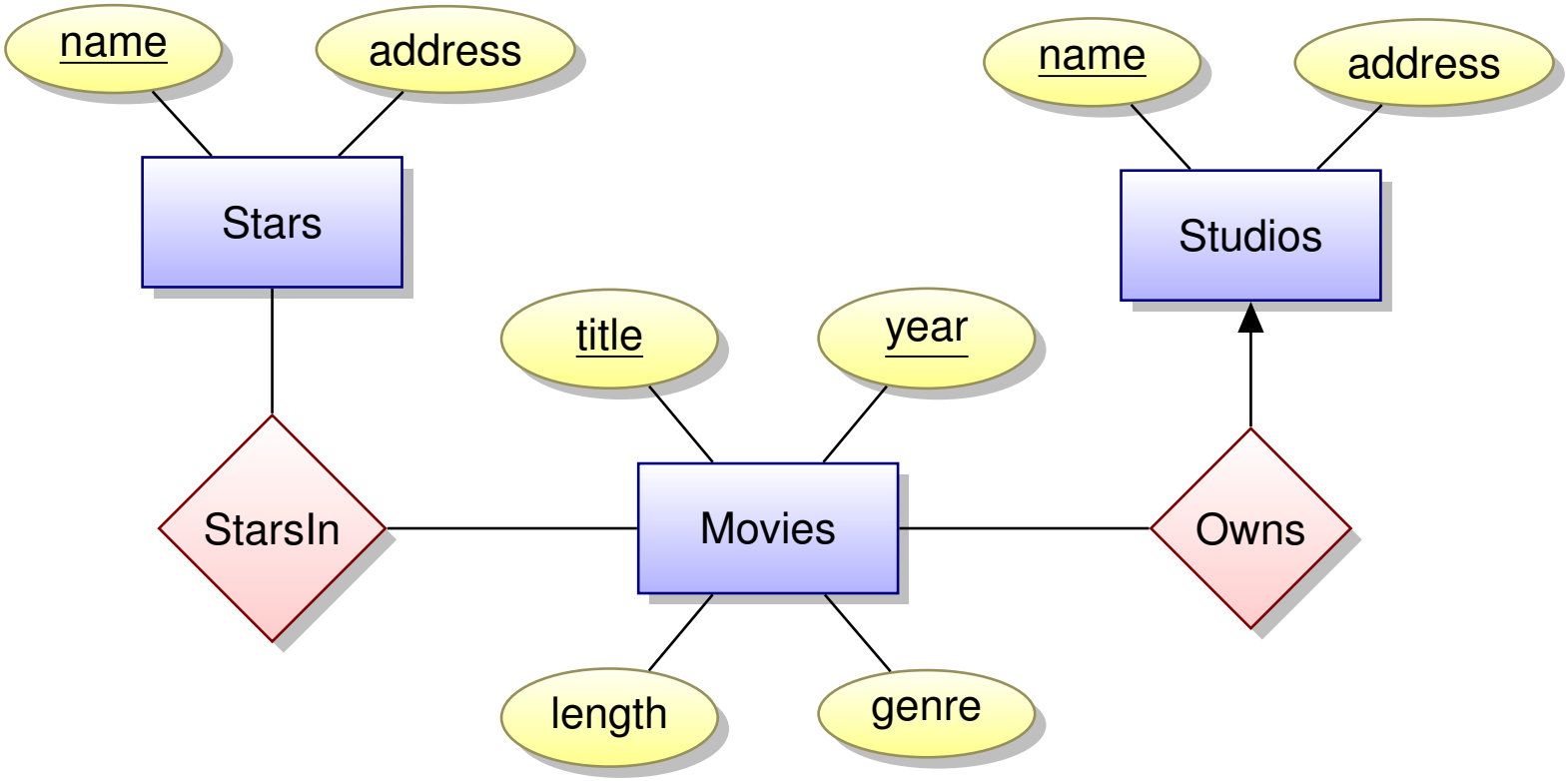

Figure: An ER model for the Movies database represented diagrammatically.

Example: ER Diagram

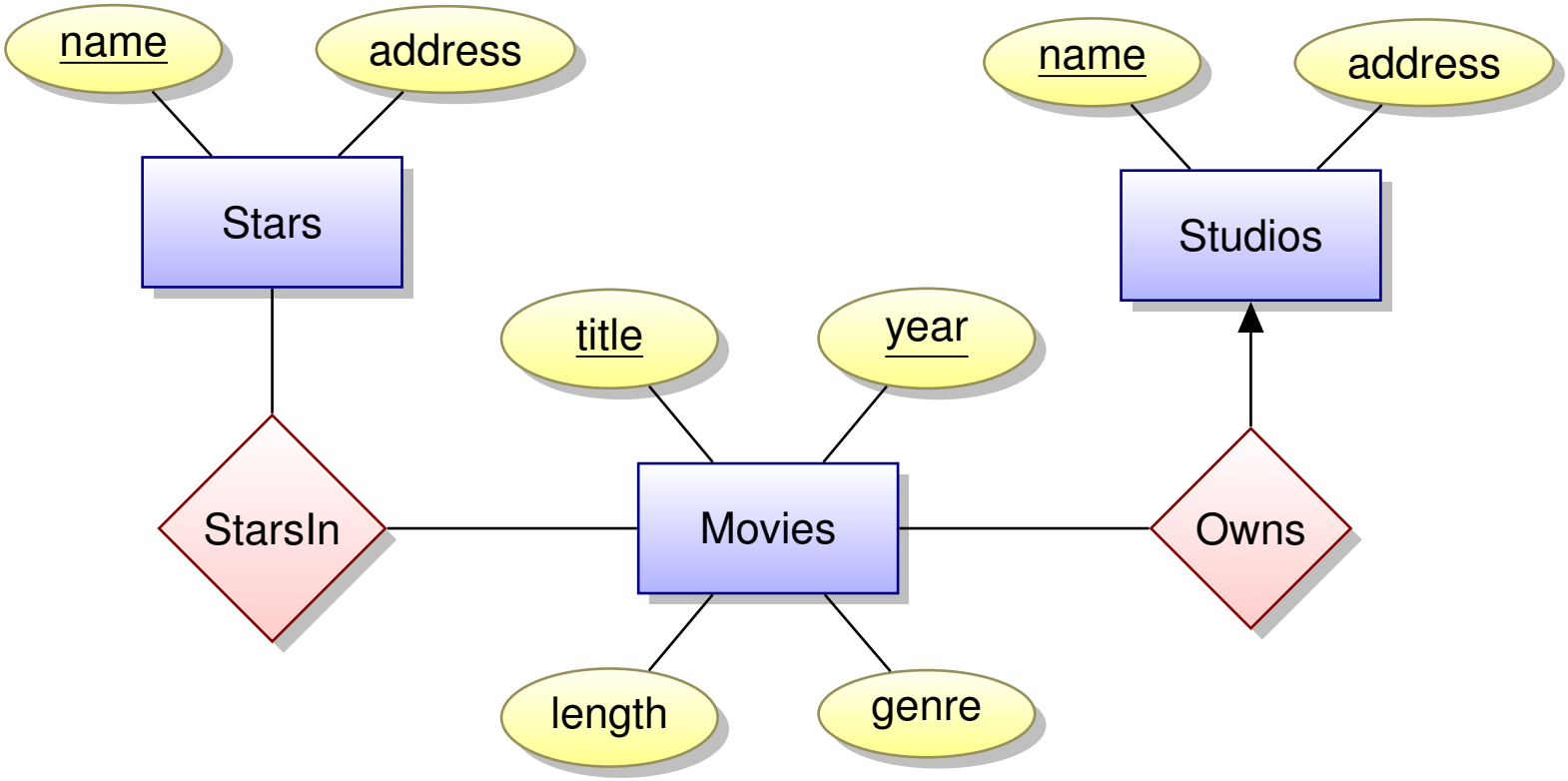

Figure: An ER model for the Movies database represented diagrammatically.

- What are the entities, attributes, and relationships of this model?

Example: ER Diagram

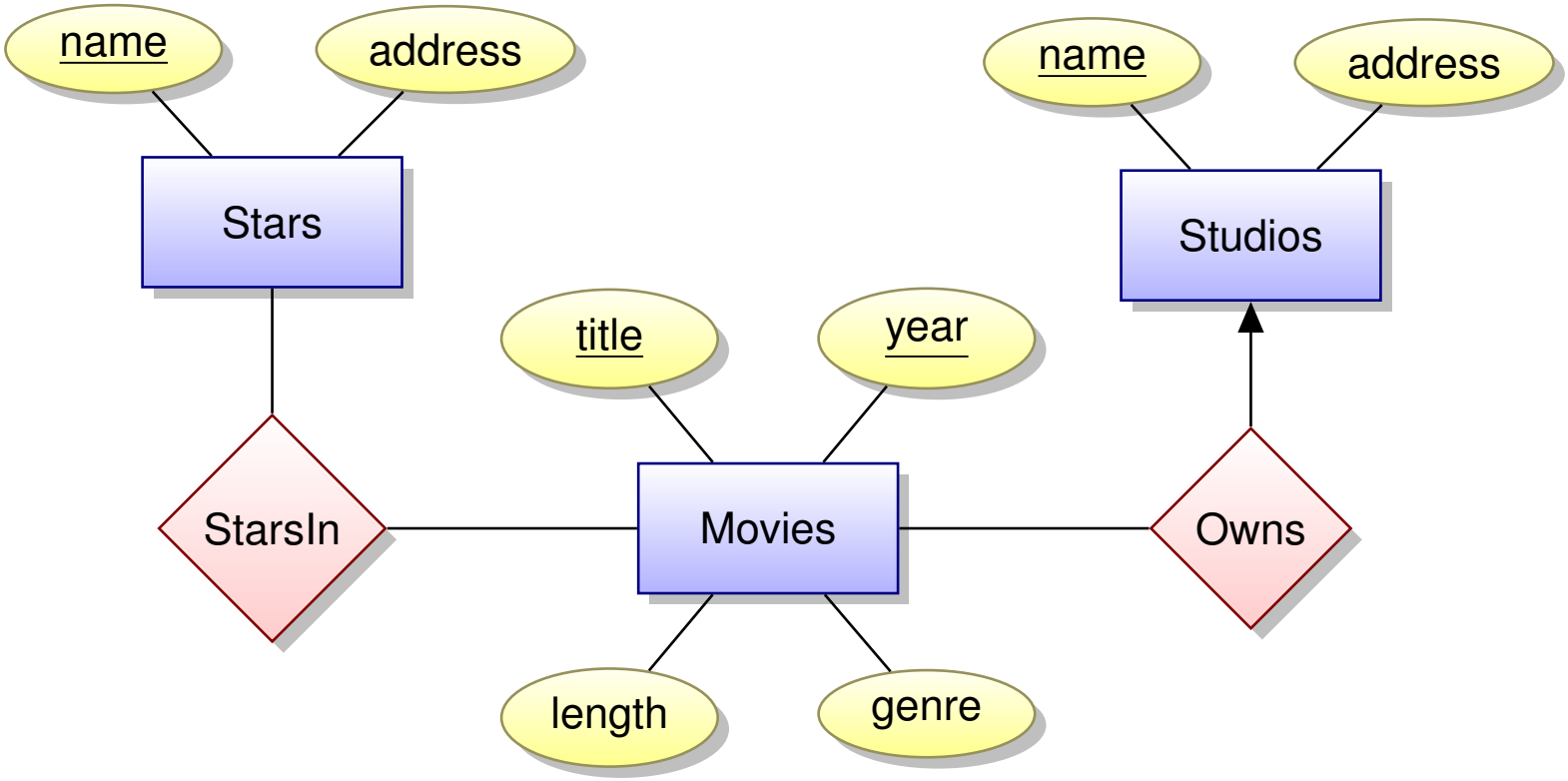

Figure: An ER model for the Movies database represented diagrammatically.

- What are the entities, attributes, and relationships of this model?
- What does the relationship *StarsIn* model?

Example: ER Diagram

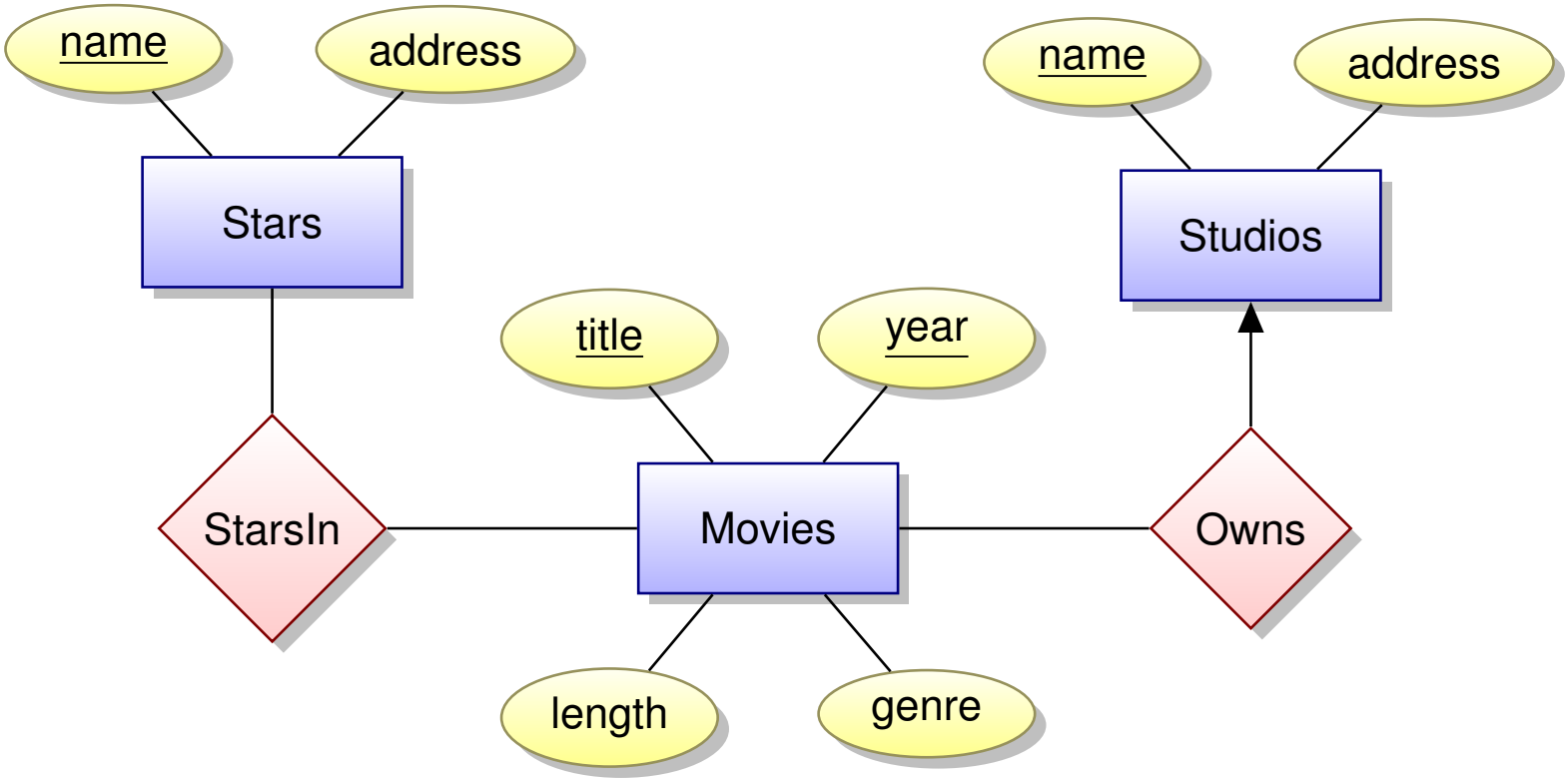

Figure: An ER model for the Movies database represented diagrammatically.

- What are the entities, attributes, and relationships of this model?
- What does the relationship *StarsIn* model?
- What does the relationship *Owns* model?

Relationship Multiplicity

Suppose a relationship `R` connects entity sets `E` and `F`. Then,

- `R` is many-one from `E` to `F` if each entity in `E` can be connected to at most one entity in `F`.
- `R` is one-one if it is many-one from both `E` to `F` and `F` to `E`.
- `R` is many-many if it is not many-one from either `E` to `F` or `F` to `E`.

Relationship Multiplicity

Entity sets are connected to relationship diamonds as follows:

- Let `R` be a many-one relationship from `E` to `F`. `R`'s diamond is connected to `F`'s rectangle with an arrow pointing to `F` and to `E`'s rectangle with a simple edge.

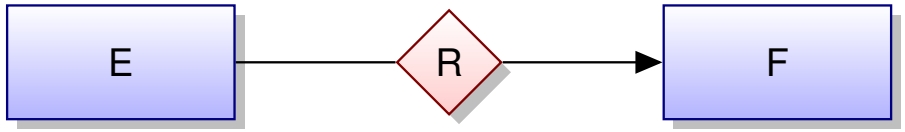

Figure: Many-one relationship `R` from `E` to `F`.

- Let `R` be a one-one relationship between `E` and `F`. `R`'s diamond is connected to both `F`'s and `E`'s rectangles with arrows pointing to each of the rectangles.

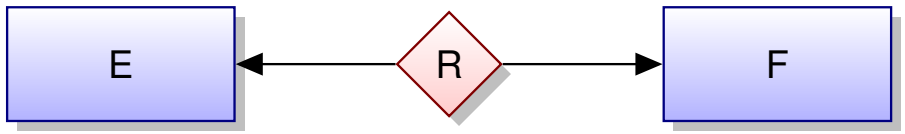

Figure: One-one relationship `R` between `E` and `F`.

Example: Relationship Multiplicity

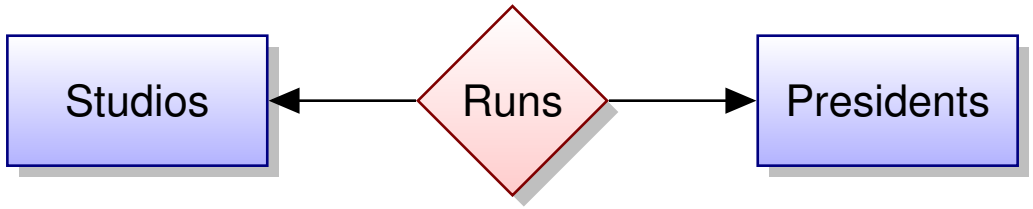

Figure: A one-one relationship.

Example: Relationship Multiplicity

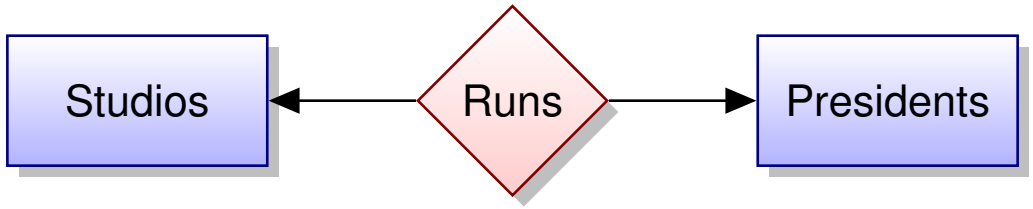

Figure: A one-one relationship.

- How would you interpret the *Runs* relationship?

Multiway Relationship

- The E/R model also supports multiway relationships.
- The multiway relationship diamond connects all involved entities.
- An arrow pointing to an entity set $\mathtt{E}$ in a multiway relationship indicates that if we pick one entity from each of the other participating entity sets, they are associated to at most one $\mathtt{E}$ entity.

Example: Multiway Relationship

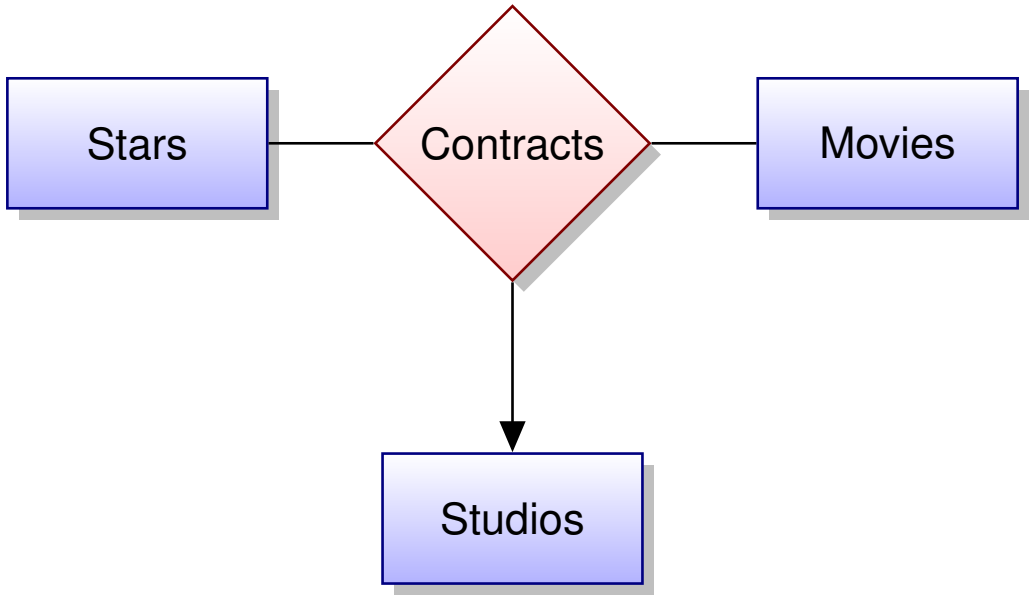

Figure: A three-way relationship.

Example: Multiway Relationship

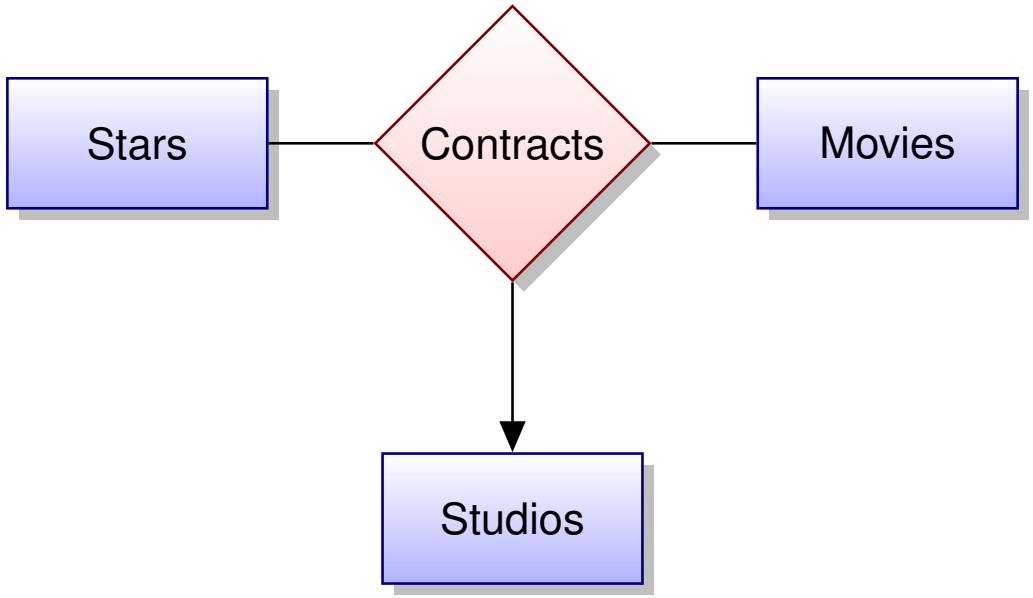

Figure: A three-way relationship.

- How would you interpret the *Contracts* relationship?

Roles in Relationships

- What if an entity set $\mathtt{E}$ appears more than once in a relationship?
- How can we distinguish amongst the different participations of $\mathtt{E}$?

Roles in Relationships

- What if an entity set E appears more than once in a relationship?
- How can we distinguish amongst the different participations of E?
- For each line connecting E to the relationship, label the role of E.

Example: Roles in Relationships

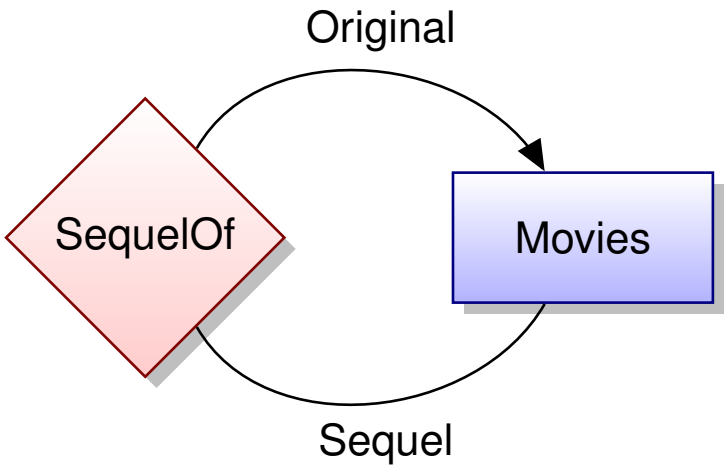

Figure: A relationship with roles.

Example: Roles in Relationships

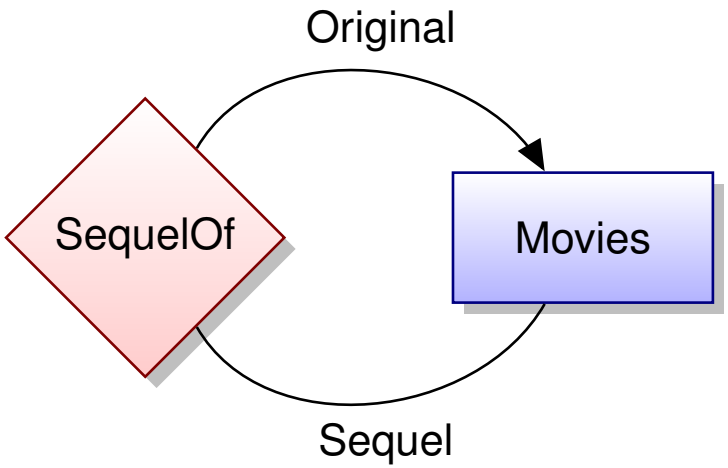

Figure: A relationship with roles.

- How would you interpret the *SequelOf* relationship?

Relationship Attributes

- Sometimes we must associate information with a relationship.
- The information is not provided the involved entity sets.
- We model the information as attributes of the relationship.
- Use the same graphical notation as for entity set attributes.

Example: Relationship Attributes

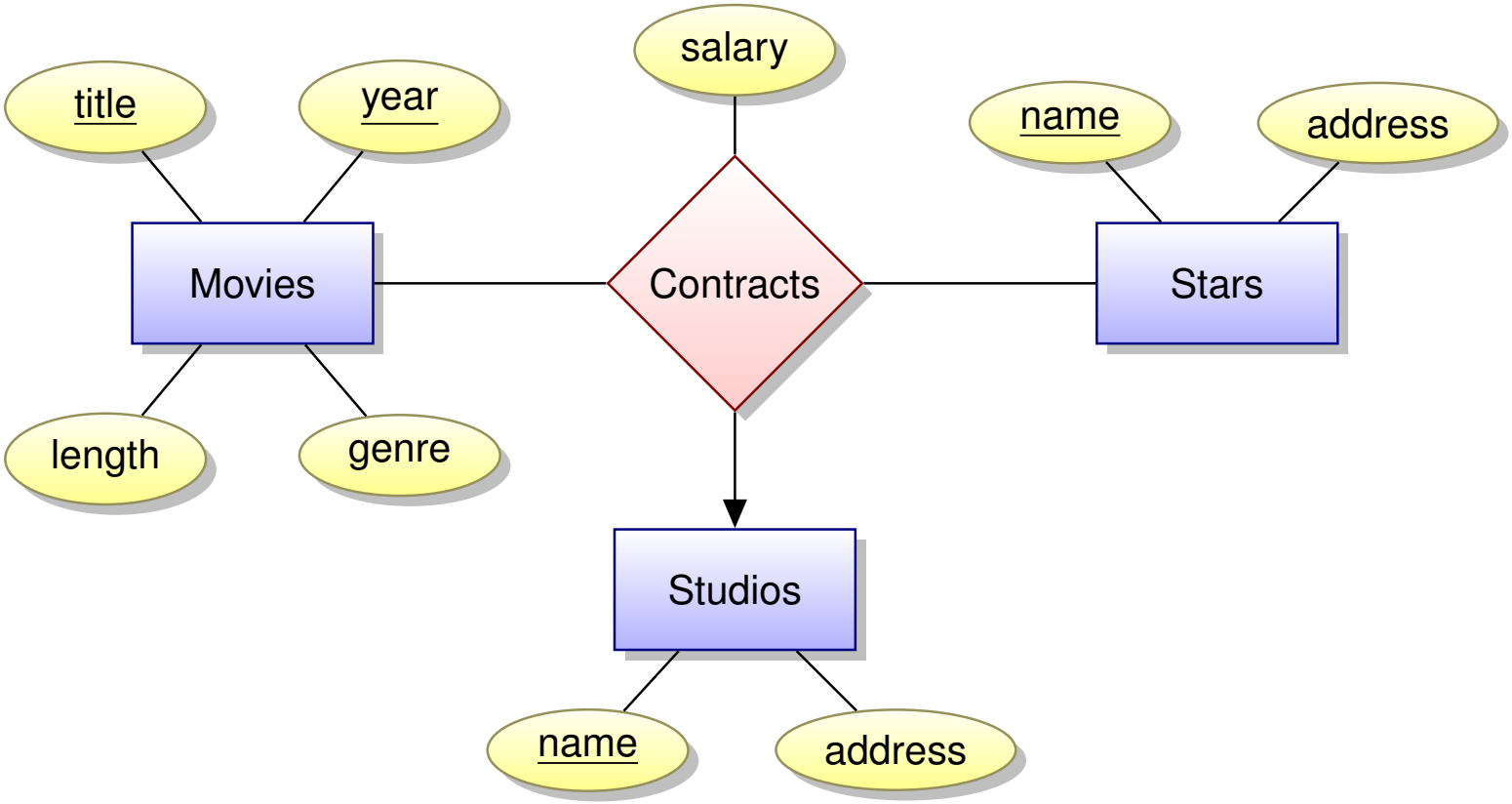

Figure: A relationship with an attribute.

Example: Relationship Attributes

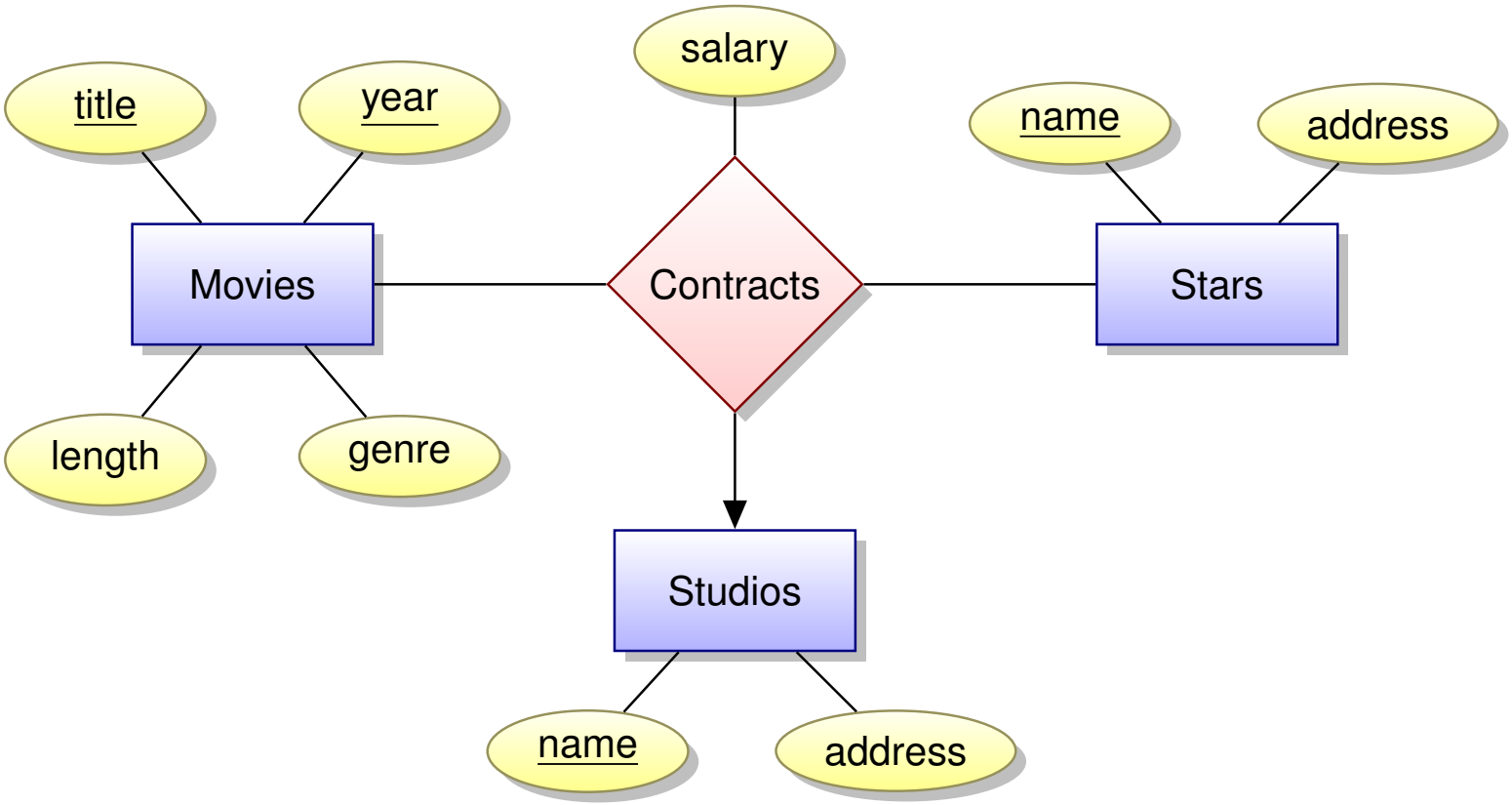

Figure: A relationship with an attribute.

- We can always factor out relationship attributes. (How?)

Example: Relationship Attributes

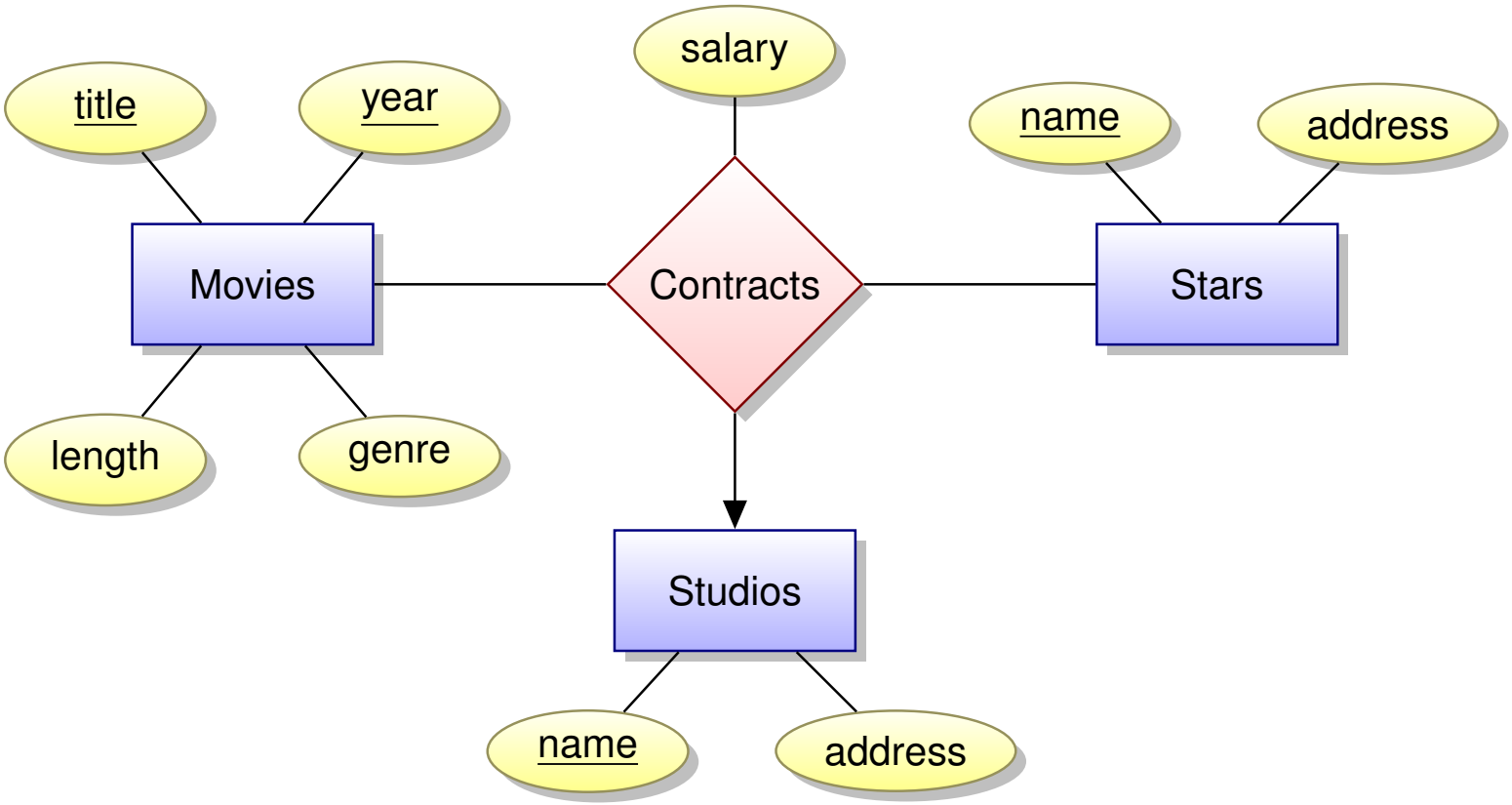

Figure: A relationship with an attribute.

- We can always factor out relationship attributes. (How?)
- Create a new entity set to hold the attributes. Include the new entity set in the relationship.

Simplifying Multiway Relationships

- Multiway relationships may be converted to a collection of binary ones.
- Create a connecting entity set to replace the multiway relationship.
- Create a many-one relationship from the connecting entity set to each of the participating entity sets (or roles).

Subclass Relationship

- Sometimes a subset `F` of a given entity set `E` has special properties (not present in `E`) associated to all its entities.
- We say `F` is a subclass of `E` (and `E` a superclass of `F`).
- An isa relationship connects an entity set with its subclasses.
- An entity may have representatives in a tree of entity sets, related by isa relationships (resembling inheritance relationships in OOPLs).
- In the ER diagram, isa relationships are represented as triangles.
- The subclass connects to one side of the triangle while the superclass connects to opposite point.
- isa relationships are one-one, but no arrows are drawn.

Example: Subclass Relationship

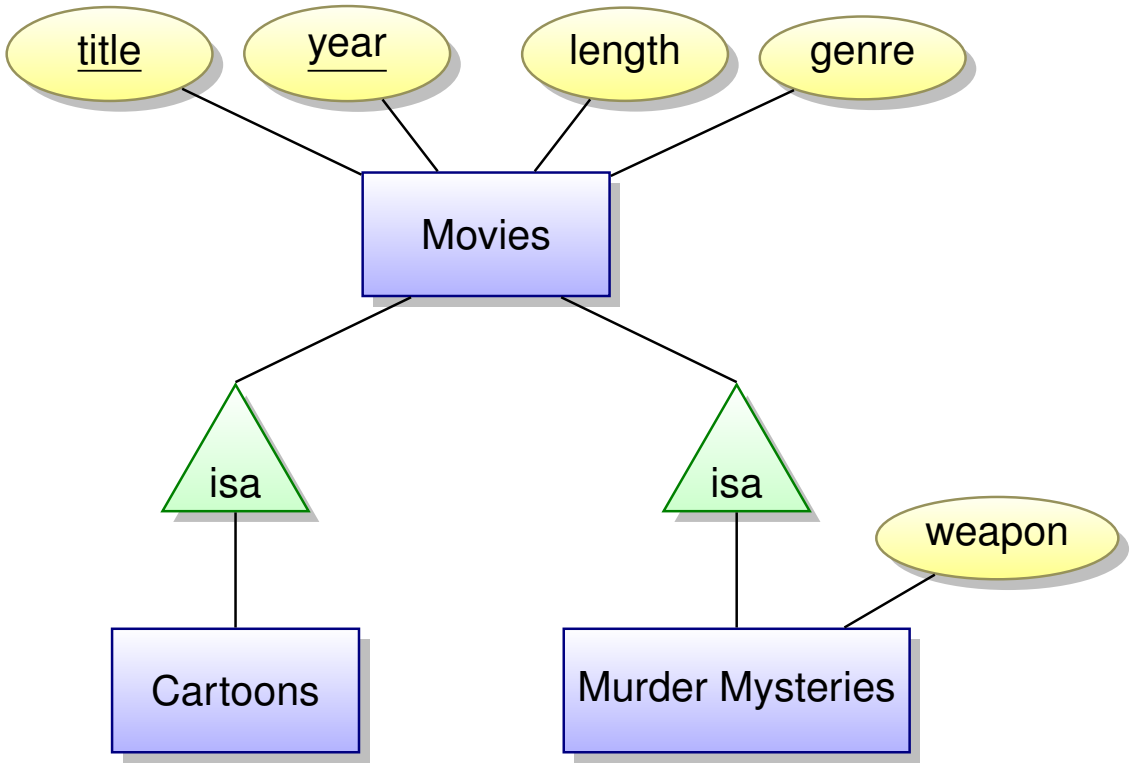

Figure: Subclass (isa) relationships.

- Murder Mysteries and Cartoons need not be disjoint.
- Some movies are neither Murder Mysteries nor Cartoons.

Example: Subclass Relationship

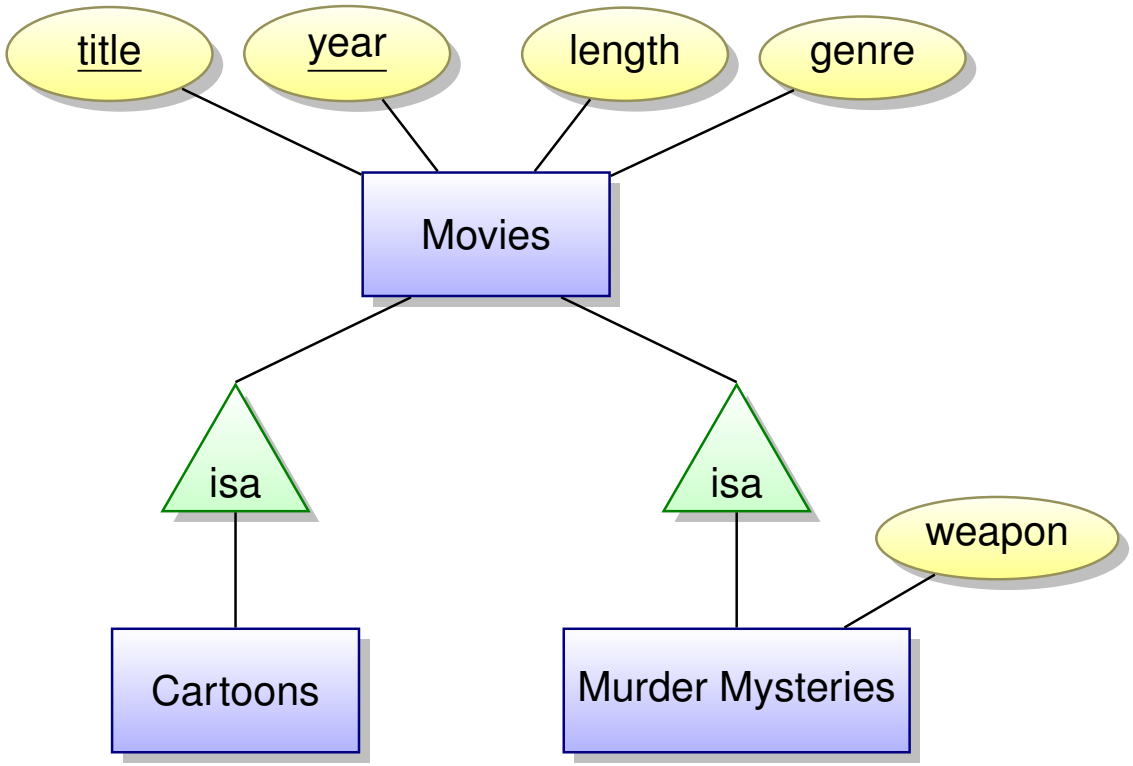

Figure: Subclass (isa) relationships.

- Murder Mysteries and Cartoons need not be disjoint.
- Some movies are neither Murder Mysteries nor Cartoons.
- What attributes and relationships do the involved entities have?

To-Do

- Discuss problems 4.1.1-4.1.10.
- On your own: read section 4.1 from chapter #4 in the textbook and go over the remaining problems.

Next...

Towards Good Designs

- Faithfulness to the specifications.
- Avoid redundancy– anomalies creep into databases during the design phase!
- Simplicity: stick to the specifications, avoid unnecessary elements.
- Do not create new relationships if they can be inferred from existing ones.
- Choose right between attributes and entity set/relationship.
- Attributes are simple and easy to understand. However, they cannot participate in relationships and only provide partial information about an entity.

To-Do

- Go over the examples and problems in this section.
- On your own: read section 4.2 from chapter #4 in the textbook and go over the remaining problems.

Next...

1. E/R Design
   - Entity/Relationship Model
   - Design Principles
   - Constraints
   - Weak Entity Sets
   - E/R Notation Summary
   - Mapping to Relational Schemas

To-Do

- We have seen how to represent keys.
- We will skip referential integrity (4.3.3) and degree constraints (4.3.4).
- On your own: read section 4.3 from chapter #4 in the textbook and go over the problems.

Next...

Motivation

- Some entity sets do not carry enough identifying information of their own.
- This may be caused by a part-of style relationship with some other entity set.
- Or by entity sets resulting from the decomposition of multiway relationships.
- Entity sets of this type are called weak entity sets.

Weak Entity Set

- Have zero or more attributes of their own, but no key.
- May have a partial key: a set of discriminator attribute(s).
- Is identified by the combination of its partial key and borrowed keys from all identifying relationships it maintains with other entity sets.

Weak Entity Set: Representation

- A weak entity set is represented by a rectangle with a double border.
- Discriminator attributes are underlined using a dashed underline.
- An identifying relationship is represented by a diamond with a double border.

To-Do

- Go over the examples and problems in this section.
- On your own: read section 4.4 from chapter #4 in the textbook and go over the remaining problems.

Next...

1. E/R Design
   - Entity/Relationship Model
   - Design Principles
   - Constraints
   - Weak Entity Sets
   - E/R Notation Summary
   - Mapping to Relational Schemas

Entity Sets

Figure: Strong

Figure: Weak

Attributes

Figure: Simple

Figure: Derived

Figure: Multivalued

Figure: Key

Figure: Discriminator

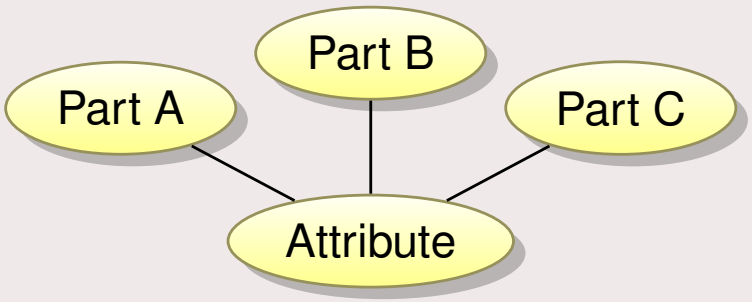

Figure: Composite

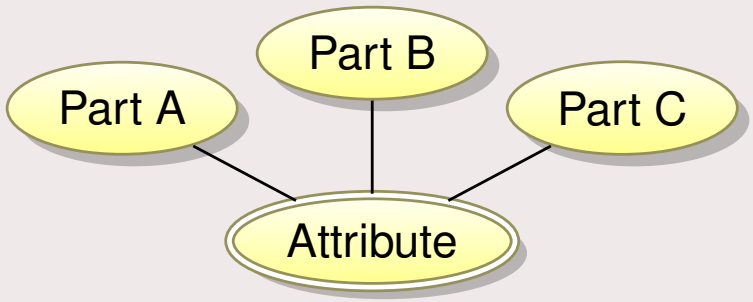

Figure: Multivalued, Composite

Relationships

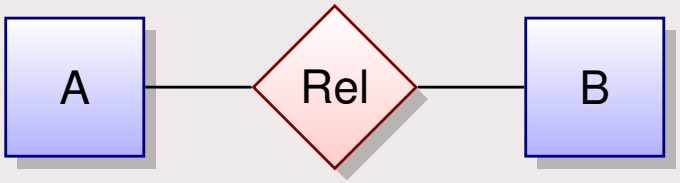

Figure: Binary, Many-Many

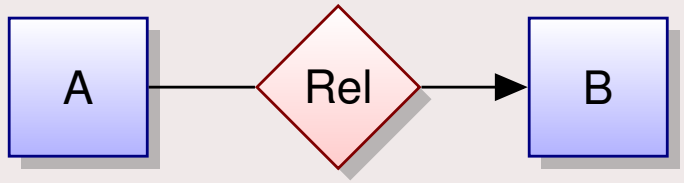

Figure: Binary, Many-One

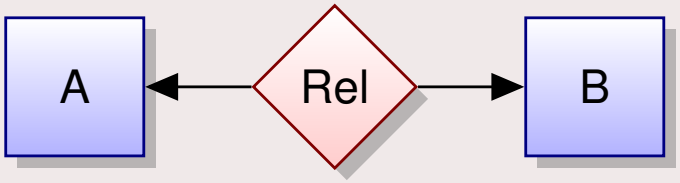

Figure: Binary, One-One

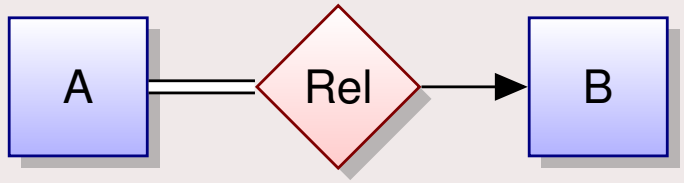

Figure: Binary, Many-One, Total

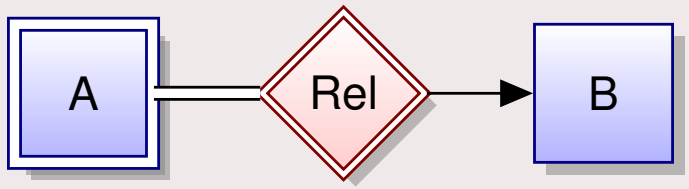

Figure: Binary, Many-One, Identifying, Total

Is-A Relationships

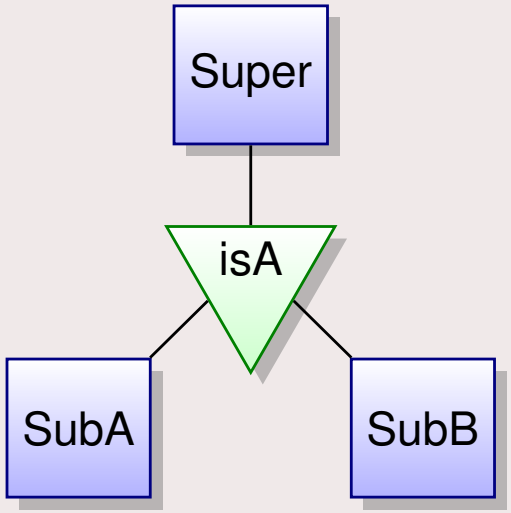

Figure: Simple

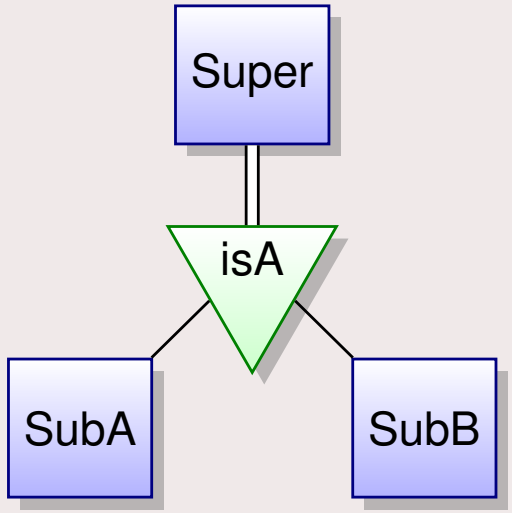

Figure: Total

Is-A Relationships

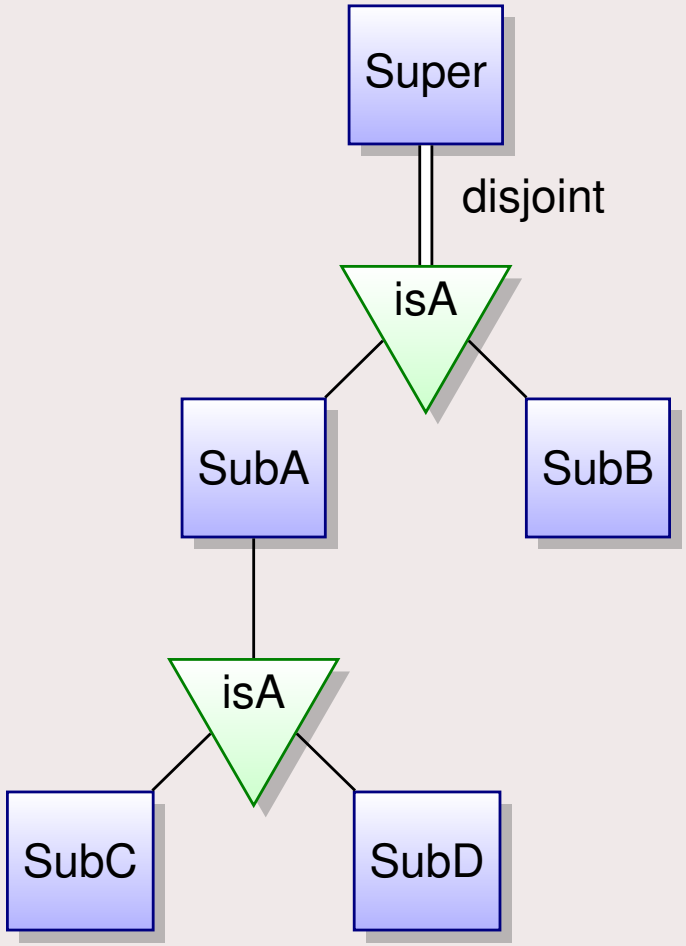

Figure: Two-Level, Total and Disjoint on Super

Next...

1. E/R Design
   - Entity/Relationship Model
   - Design Principles
   - Constraints
   - Weak Entity Sets
   - E/R Notation Summary
   - Mapping to Relational Schemas

In a Nutshell...

ER Element	Relational Schema Element(s)
Entity Set	Relation
1:1 or 1:N Relationship	Foreign key (or relationship relation)
N:M Relationship	Relationship relation and two foreign keys
N-ary Relationship	Relationship relation and *n* foreign keys
Simple Attribute	Attribute
Composite Attribute	Set of attributes
Multivalued Attribute	Relation and foreign key

Table: Mapping ER Models to Relational Schemas.

1. Mapping Attributes

- A simple attribute maps to a single relational attribute.
- A composite attribute maps to a set of relational attributes, one per component.
- A multivalued attribute maps to a separate relation with a borrowed key from the source relation and a partial key from the attribute itself (this works much like a weak entity).

2. Mapping Regular Entity Sets

- Map the regular entity set E to relation R_E.
- R_E schema: Attrs(E).
- R_E primary key: Key(E).

3. Mapping Weak Entity Sets

- Let E be a weak entity identified by $E_1, \ldots, E_n$. Map E to relation R_E.
- R_E schema: $\bigcup_i$ Key(E_i) $\bigcup$ Attrs(E).
- R_E primary key: $\bigcup_i$ Key(E_i) $\bigcup$ Discriminator(E).
- R_E foreign keys: one for each E_1, ..., E_n.

4. Mapping Specialization/Generalization

- Let E be a specialization of G. Map E to relation R_E.
- R_E schema: Key(G) $\bigcup$ Attrs(E).
- R_E primary key: Key(G).
- R_E foreign keys: one for G.

5. Mapping Binary Relationships

- Let S be a relationship involving entity sets E_1 and E_2. Map S to relation R_S.
- R_S schema: Key(E_1) $\bigcup$ Key(E_2) $\bigcup$ Attrs(S).
- R_S primary key:
  - if S is many-many: Key(E_1) $\bigcup$ Key(E_2);
  - if S is many-one: Key(E_1);
  - if S is one-one: choose Key(E_1) or Key(E_2).
- R_S foreign keys: one for E_1 and one for E_2.
- The rules above generalize for n-ary relationships.

6. Mapping N-Ary Relationships

- Let S be a relationship involving entity sets $E_1, \ldots, E_n$. Map S to relation R_S.
- R_S schema: $\bigcup_i$ Key(E_i) $\bigcup$ Attrs(S).
- R_S primary key: $\bigcup_j$ Key(E_j) for all E_j participating in S with cardinality > 1.
- R_S foreign keys: one for each E_1, …, E_n.

7. Optimization

- Let R_1 and R_2 be two relations obtained from the mapping above.
- We may be able to combine R_1 and R_2 into a relation R_3.
- Requirement: R_1 and R_2 come from the same entity set and Key(R_1) = Key(R_2).
- R_3 schema: Attrs(R_1) $\bigcup$ Attrs(R_2).
- R_3 primary key: Key(R_1).
- R_3 foreign keys: the union of the foreign keys of R_1 and R_2 (except those of R_1 that referred to R_2 and vice-versa).

8. DDL Generation

- Generate a `CREATE TABLE` DDL statement for each table.
- Define reasonable data types for each attribute.
- Define `NOT NULL` constraints for mandatory attributes.
- Define `NOT NULL` constraints for total participation in relationships.
- Define check constraints for attributes, if necessary.
- Define primary keys, unique keys (if any), and foreign keys (if any).

To-Do

- Go over the examples and problems in this section.
- On your own: read section 4.5 from chapter #4 in the textbook and go over the remaining problems.

To-Do

- Go over the provided example.
- Go over additional examples.