Enhanced Entity-Relationship Model

Enhanced Entity-Relationship Model

- Semantic concepts are incorporated into the original ER model and called the Enhanced Entity-Relationship (EER) model.

- Examples of additional concept of EER model is called specialization / generalization.

- The concept of specialization/generalization is associated with special types of entities known as **superclasses** and **subclasses,** and the process of **attribute inheritance.**

Superclasses and Subclasses

- **Superclass**
  - An entity set that includes one or more distinct subgroupings of its occurrences.
- **Subclass**
  - A distinct subgrouping of occurrences of an entity set.
- Superclass and subclass also may be designated by the terms **Higher-** and **lower-level** entity sets

Superclasses and Subclasses

- The relationship between a superclass and any one of its subclasses is called a **superclass/subclass relationship.**

- Superclass/subclass relationship is one-to-one (1:1).

Superclasses and Subclasses

- Each member of a subclass is also a member of the superclass.

- In other words, the entity in the subclass is the same entity in the superclass, but has a **distinct role**.

- Superclass may contain overlapping or distinct subclasses

- Not all members of a superclass need be a member of a subclass.

Attribute Inheritance

- An entity in a subclass represents same ‘real world’ object as in superclass, and may possess subclass-specific attributes, as well as those associated with the superclass.

- a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked

Specialization

- **Specialization**
  - Process of maximizing differences between members of an entity by identifying their distinguishing characteristics.
  - Melalui proses ini akan diperoleh satu atau beberapa subclass sehingga bisa dipakai untuk menyusun hubungan superclass/subclass.
  - **top-down** approach to defining a set of superclasses and their related subclasses

Specialization

- Specialization is a top-down approach to defining a set of superclasses and their related subclasses.
  - we designate subgroupings within an entity set that are distinctive from other entities in the set.
  - These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled ISA (E.g. *customer* "is a" *person*).

Generalization

- **Generalization**
  - Process of minimizing differences between entities by identifying their common characteristics.
  - The process of generalization is a bottom-up approach, which results in the identification of a generalized superclass from the original entity types.
  - **A bottom-up design process** – combine a number of entity sets that share the same features into a higher-level entity set.

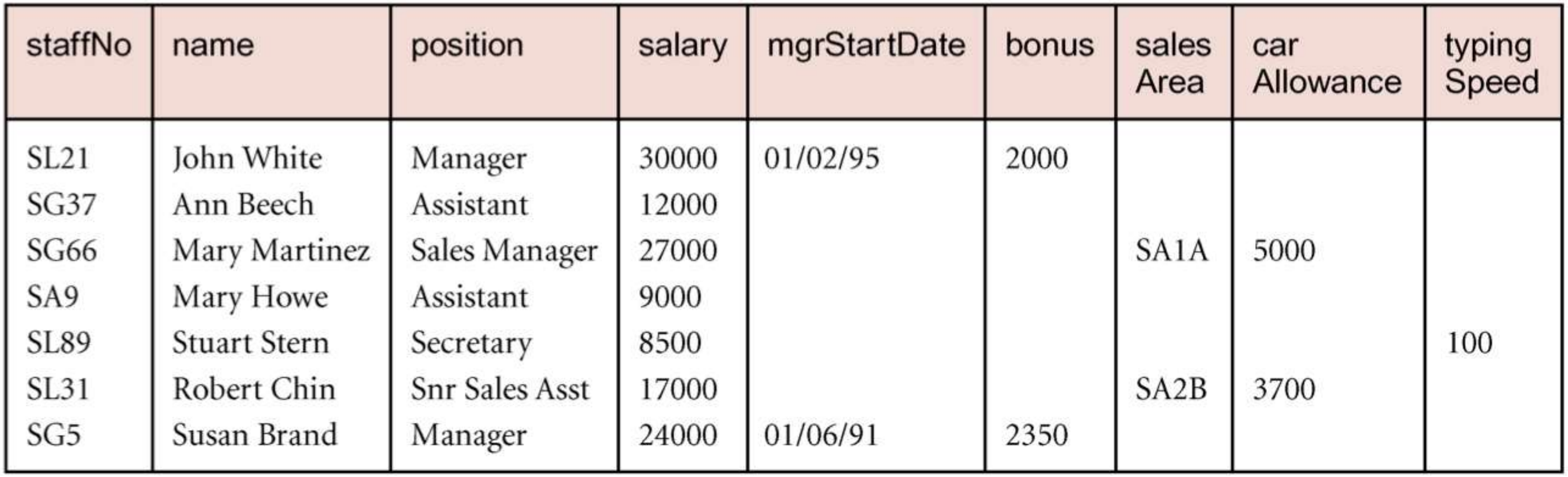
Generalization

- Langkah-langkah dasar untuk melakukkan generalisasi adalah seperti berikut
  - Cari atribut yang berlaku untuk semua tipe entitas dan bentuk superclass yang berisikan atribut-atribut tersebut. Tentukan pula kunci primernya.
  - Bentuk masing-masing subclass yang hanya berisi atribut yang tidak terdapat pada superclass
  - Gambarkan hubungan superclass/subclass-nya.

Specialization And Generalization

- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
- The terms specialization and generalization are used interchangeably

Specialization And Generalization Example

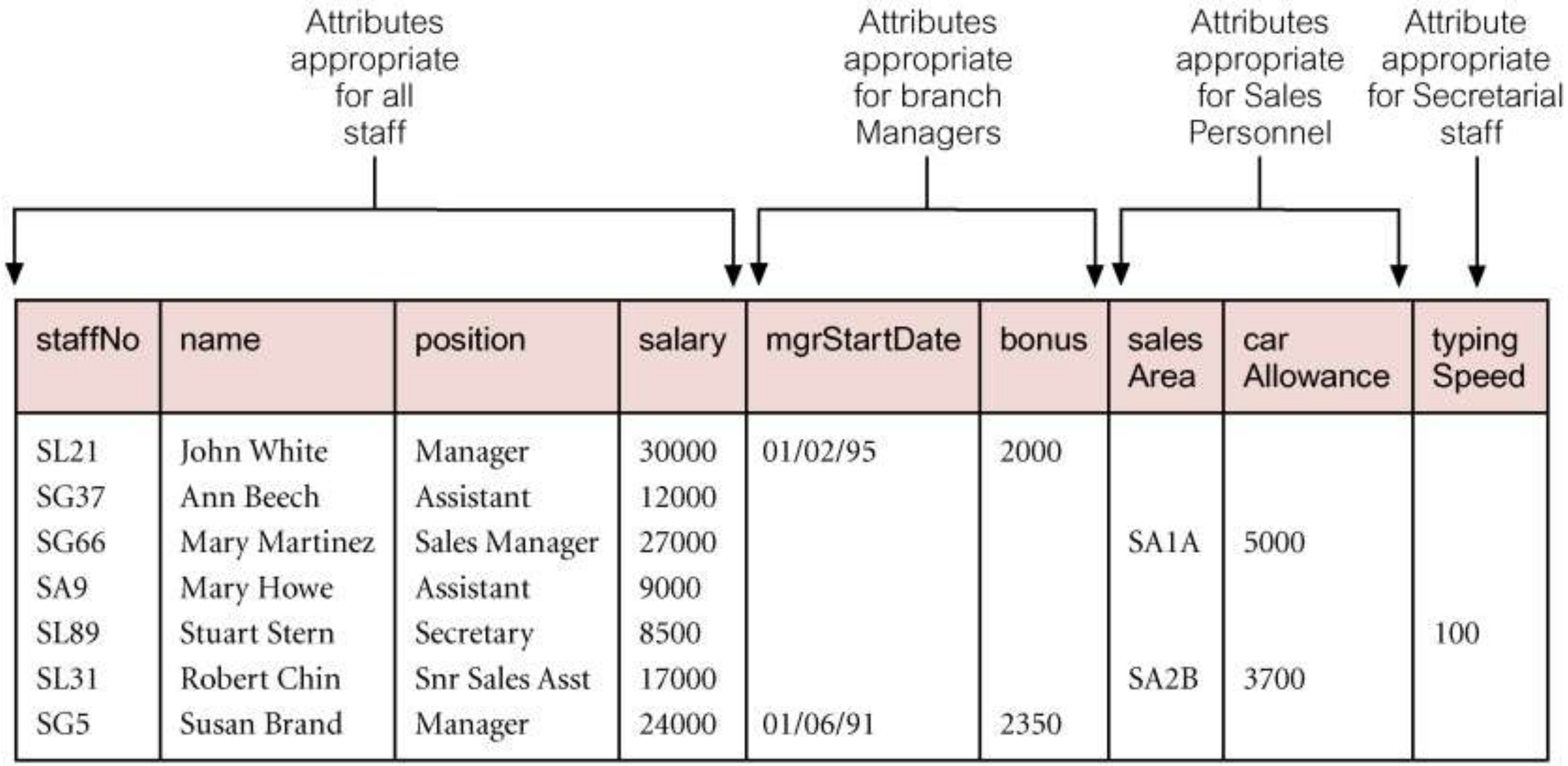


staffNo	name	position	salary	mgrStartDate	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31	Robert Chin	Snr Sales Asst	17000			SA2B	3700	
SG5	Susan Brand	Manager	24000	01/06/91	2350			

Specialization And Generalization Example

- Manager, SalesPersonnel, and Secretary as subclasses of a specialized Staff superclass
- Manager, Secretary, and SalesPersonnel as subclassess of generalized Staff superclass
- **IS-A hierarchy**
  - Manager IS-A (member of) Staff).

Specialization And Generalization Example



Attributes appropriate for all staff

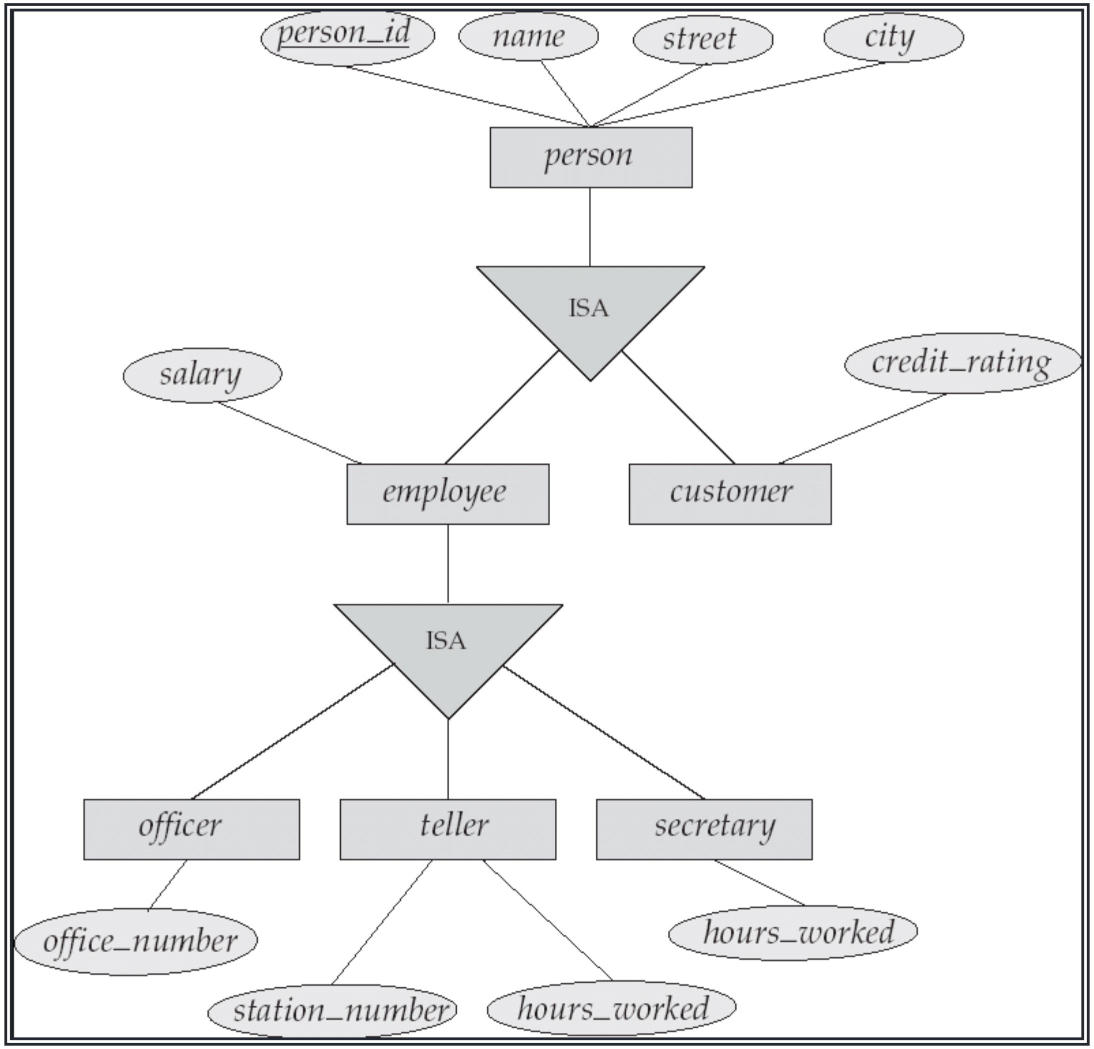
Attributes appropriate for branch Managers

Attributes appropriate for Sales Personnel

Attribute appropriate for Secretarial staff

staffNo	name	position	salary	mgrStartDate	bonus	sales Area	car Allowance	typing Speed
SL21	John White	Manager	30000	01/02/95	2000			
SG37	Ann Beech	Assistant	12000					
SG66	Mary Martinez	Sales Manager	27000			SA1A	5000	
SA9	Mary Howe	Assistant	9000					
SL89	Stuart Stern	Secretary	8500					100
SL31	Robert Chin	Snr Sales Asst	17000			SA2B	3700	
SG5	Susan Brand	Manager	24000	01/06/91	2350			

Specialization And Generalization Example



Specialization And Generalization

- Can have multiple specializations of an entity set based on different features.
- E.g. *permanent_employee* vs. *temporary_employee*, in addition to *officer* vs. *secretary* vs. *teller*
- Each particular employee would be
  - a member of one of *permanent_employee* or *temporary_employee*,
  - and also a member of one of *officer*, *secretary*, or *teller*

Design Constraints on a Specialization/Generalization

- **Disjointness constraint** -- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.
  - **Disjoint**
    - an entity can belong to only one lower-level entity set
    - Noted in E-R diagram by writing disjoint next to the ISA triangle
  - **Overlapping**
    - an entity can belong to more than one lower-level entity set

Design Constraints on a Specialization/Generalization

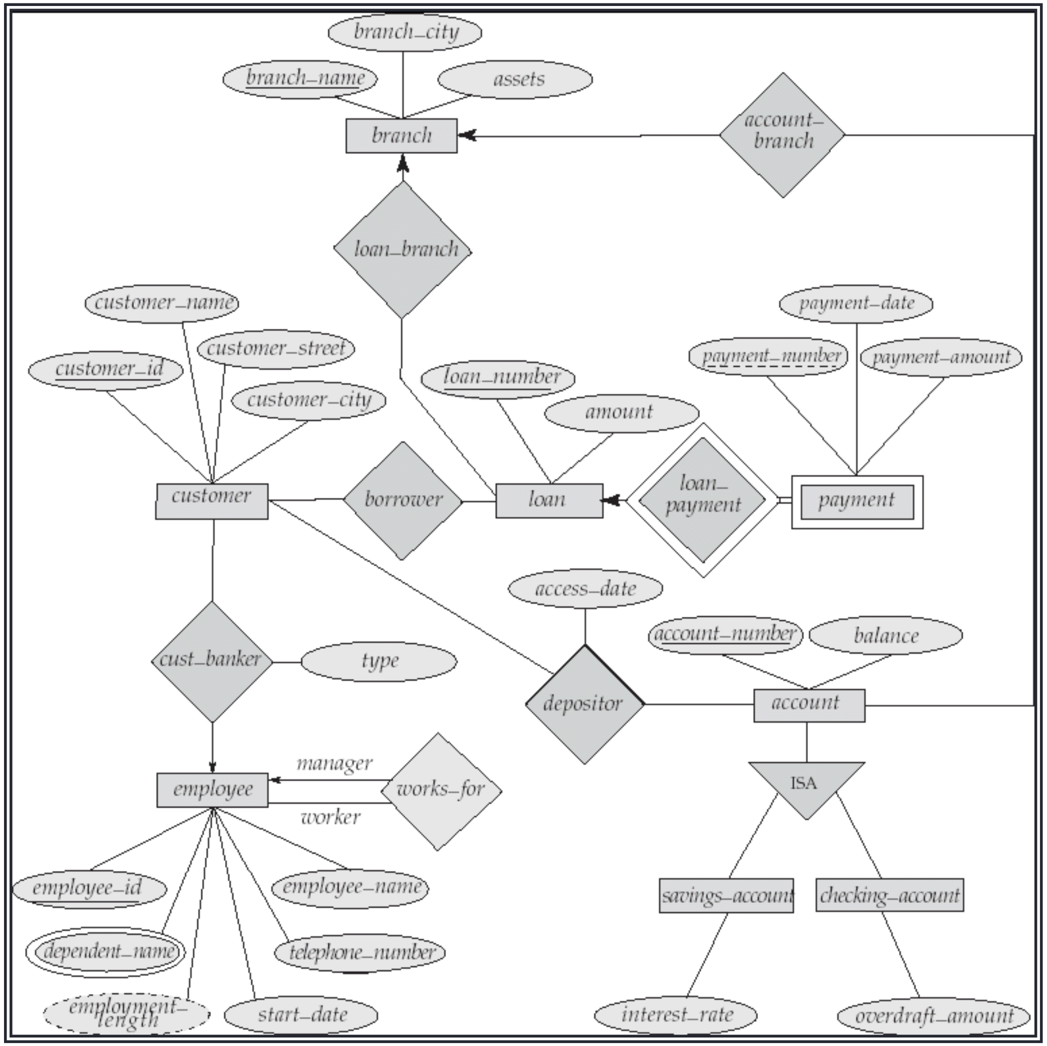
- **Completeness constraint** -- specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.
  - **total**: an entity must belong to one of the lower-level entity sets
  - **partial**: an entity need not belong to one of the lower-level entity sets

Specialization and Generalization Hierarchies and Lattices

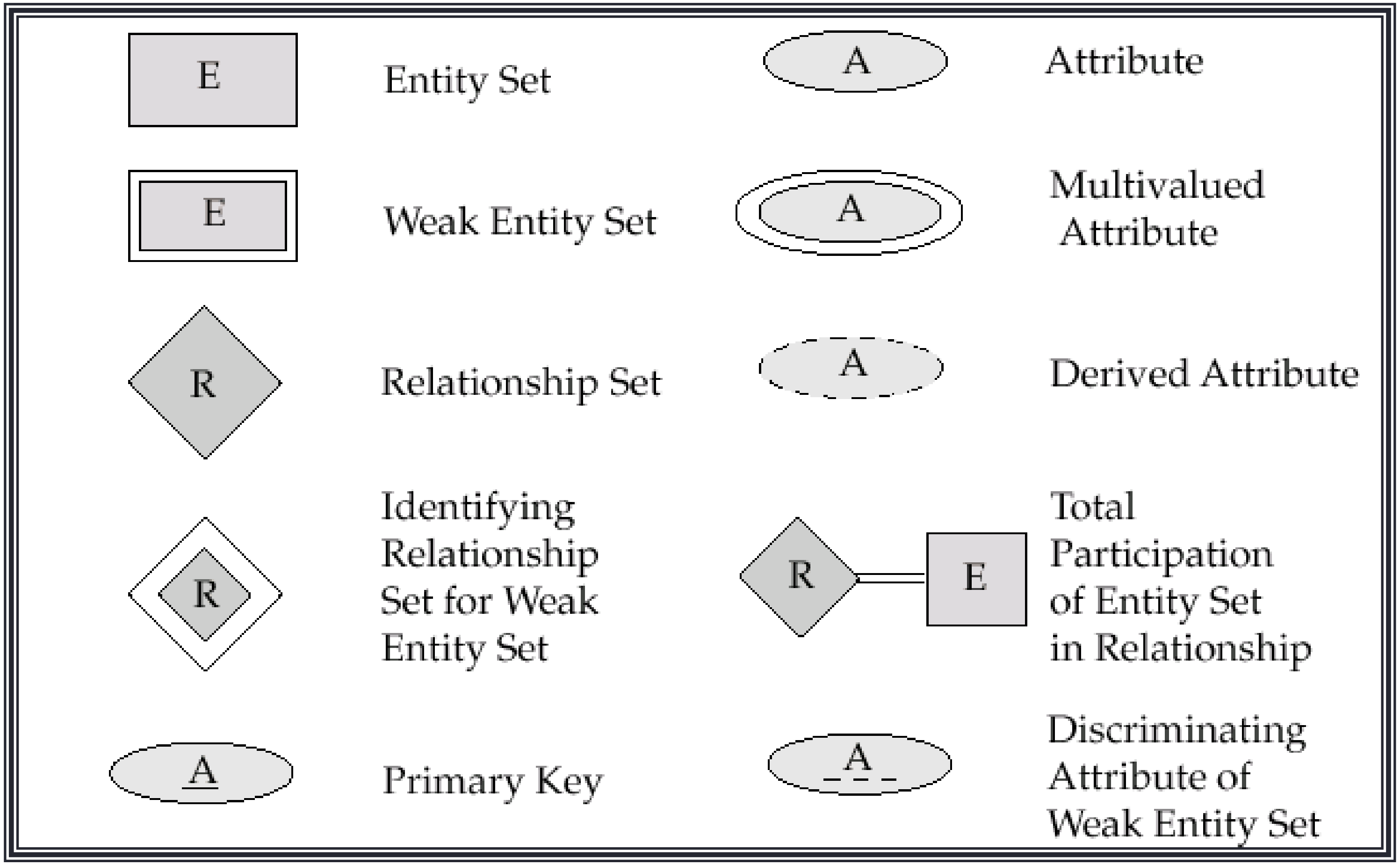
- **Specialization hierarchy** (hierarki tipe) untuk menyatakan keadaan yang menggambarkan bahwa suatu subclass hanya mempunyai satu superclass tetapi sebuah subclass juga bisa bertindak sebagai supeclass bagi subclass lainnya

- **Specialization lattice**, a subclass can be a subclass in *more than one class/subclass relationship.*

E-R Diagram for a Banking Enterprise



Summary of Symbols Used in E-R Notation



Entity Set

Attribute

Weak Entity Set

Multivalued Attribute

Relationship Set

Derived Attribute

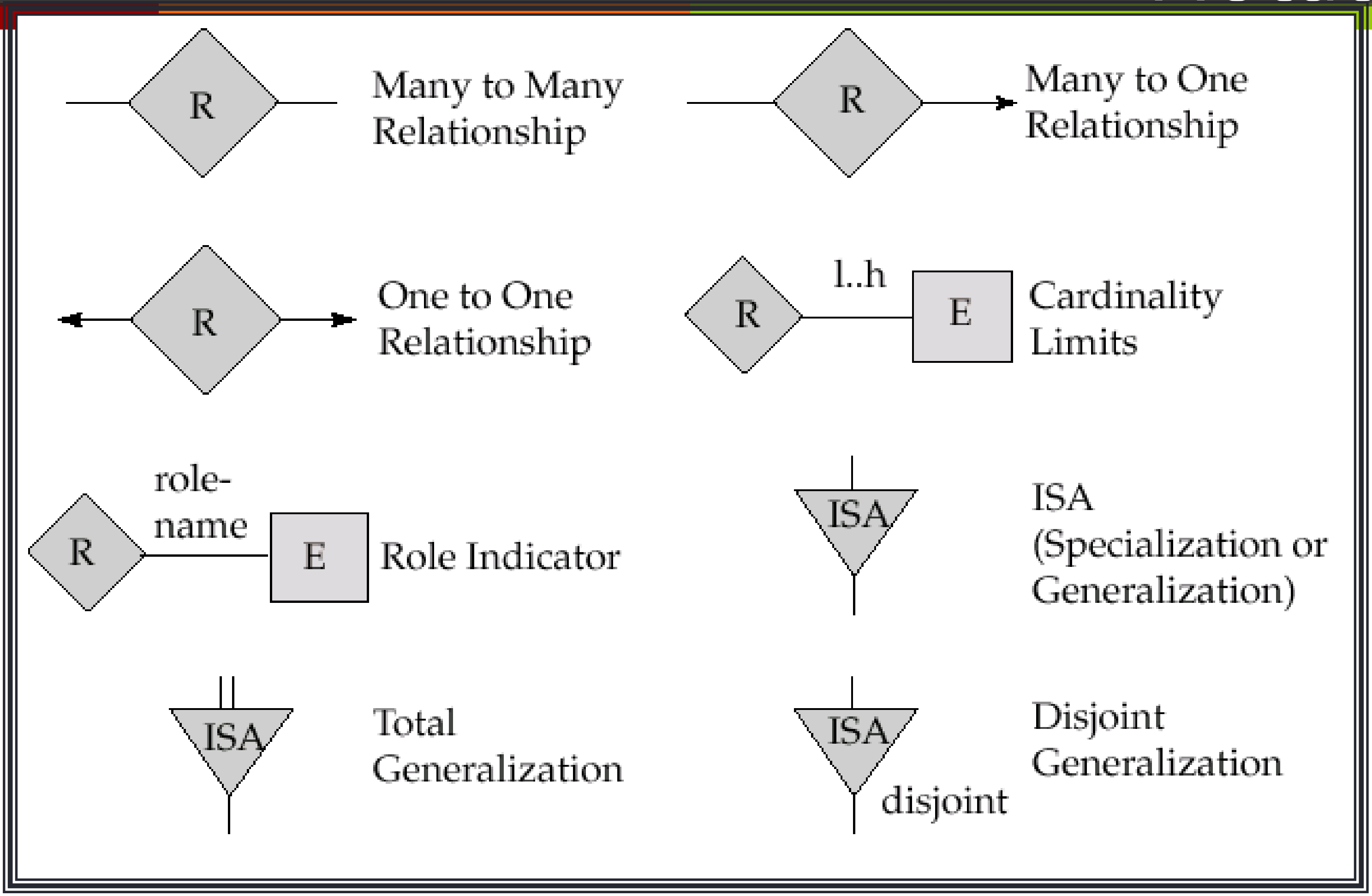
Identifying Relationship Set for Weak Entity Set

Total Participation of Entity Set in Relationship

Primary Key

Discriminating Attribute of Weak Entity Set

Summary of Symbols Used in E-R Notation



Many to Many Relationship

Many to One Relationship

One to One Relationship

Cardinality Limits

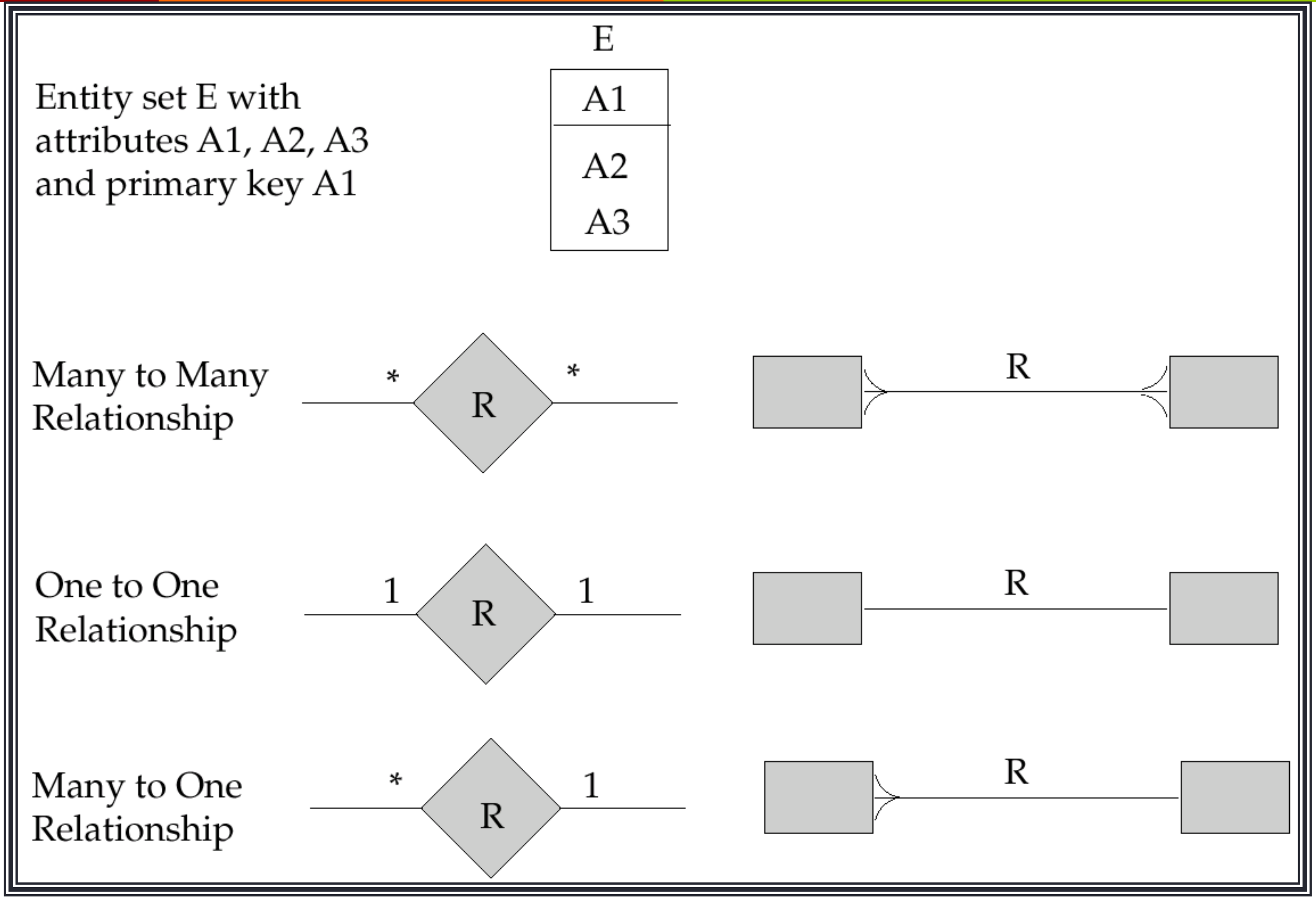
Role Indicator

ISA (Specialization or Generalization)

Total Generalization

Disjoint Generalization

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

Reduction to Relation Schemas

- Primary keys allow entity sets and relationship sets to be expressed uniformly as *relation schemas* that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of schemas.
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.
- Each schema has a number of columns (generally corresponding to attributes), which have unique names.

Representing Entity Sets as Schemas

- A strong entity set reduces to a schema with the same attributes.
- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

  payment =

  (*<u>loan_number</u>*, *<u>payment_number</u>*, *payment_date*, *payment_amount*)

Representing Relationship Sets as Schemas

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.

- Example: schema for relationship set borrower

  borrower = (*customer_id, loan_number*)

Representing Specialization via Schemas

- Method 1:
  - Form a schema for the higher-level entity
  - Form a schema for each lower-level entity set, include primary key of higher-level entity set and local attributes

schema	attributes
person	*name, street, city*
customer	*name, credit_rating*
employee	*name, salary*

  - Drawback: getting information about, an *employee* requires accessing two relations, the one corresponding to the low-level schema and the one corresponding to the high-level

Representing Specialization as Schemas

- Method 2:
  - Form a schema for each entity set with all local and inherited attributes

schema	attributes
person	*name, street, city*
customer	*name, street, city, credit_rating*
employee	*name, street, city, salary*

  - If specialization is total, the schema for the generalized entity set (*person*) not required to store information
    - Can be defined as a "view" relation containing union of specialization relations
    - But explicit schema may still be needed for foreign key constraints
  - Drawback: *street* and *city* may be stored redundantly for people who are both customers and employees