Entity Relationship Model

Chapter 5

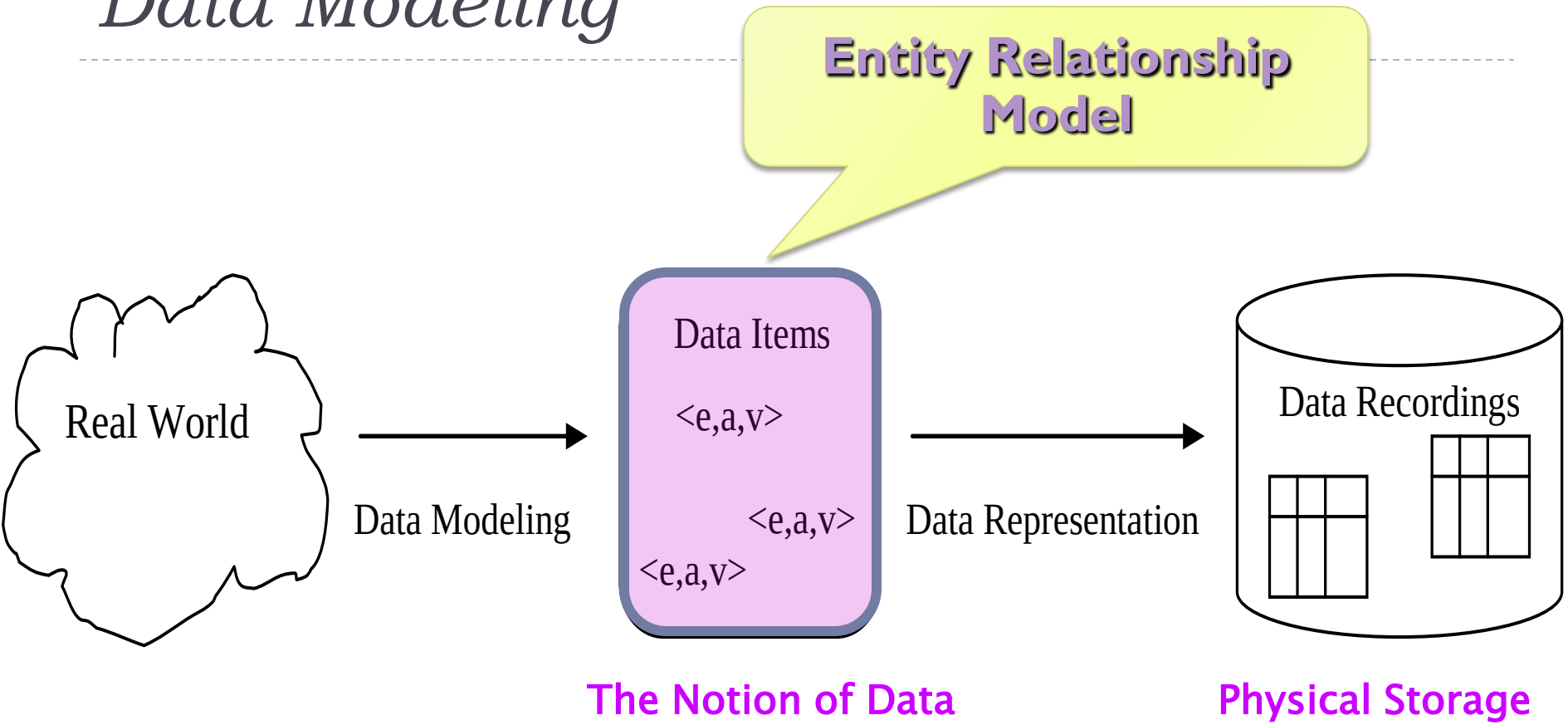
Objectives

- Learn how to analyze real world data
  - Data and properties
  - Relationships
- Represent data in a conceptual model
  - ER Model
  - The first step of database design
- Learn to construct ER diagrams

Outline

- Notation basics
- Understanding entity relationships
- Generalization hierarchies
- Diagram rules

Data Modeling



Database Model

Database application development

1. Requirement analysis
2. Component design
    1. Data modeling
    2. SW design
3. Implementation

- User interview
- Document analysis
    - reports, business rules, etc.
- Stakeholder Meeting

A Library Case

- Collection of Books, index cards
- Members
- Borrowing and Returning books

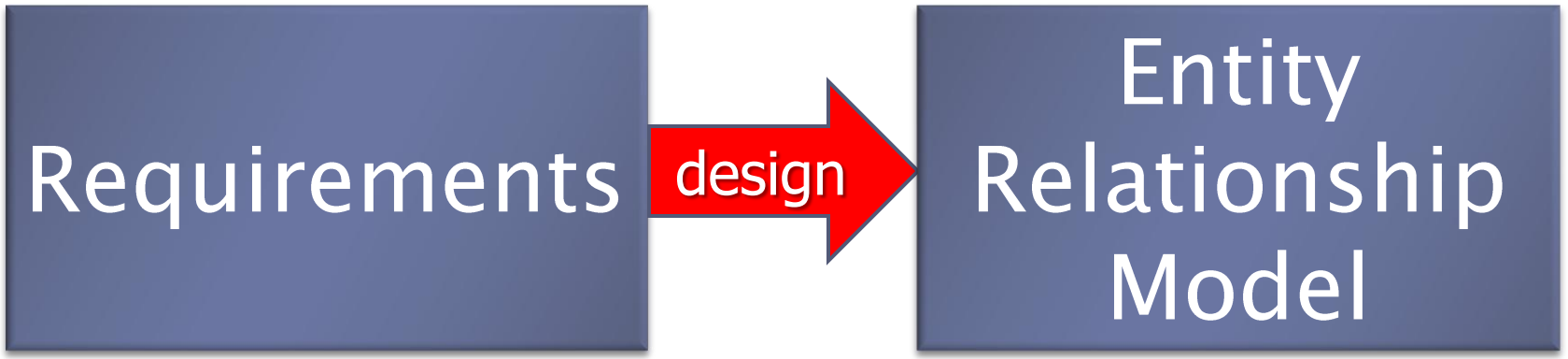
- Tom borrows a database book.

Tom is a member, noun

Borrowing is a process (verb) involving a book and a member

A database book is a book, noun

Design stage



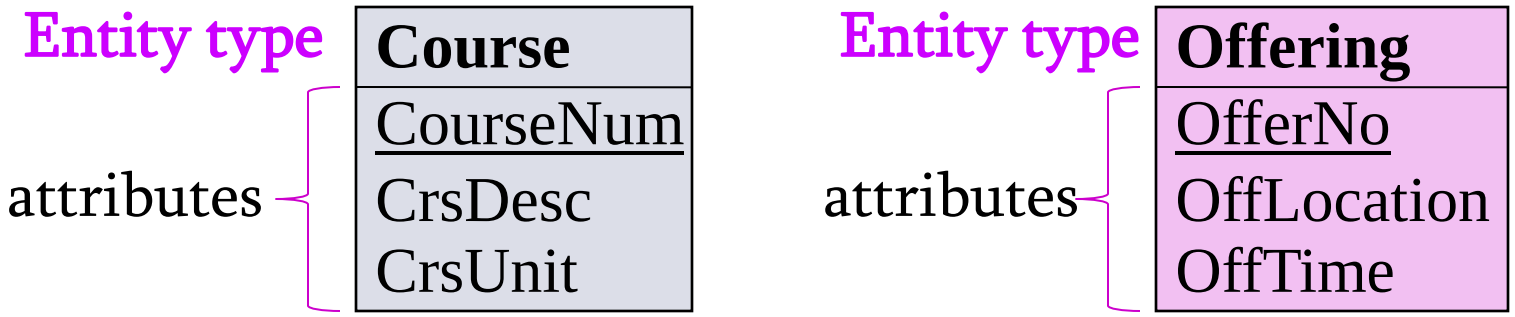
- Conceptual model
- See the overall design of DB
- Independent of database technology

Entity Relationship Model (ER Model)

1. Entities
2. Attributes
3. Primary keys
4. Relationships
5. Cadinality

Entity type (Entity class)

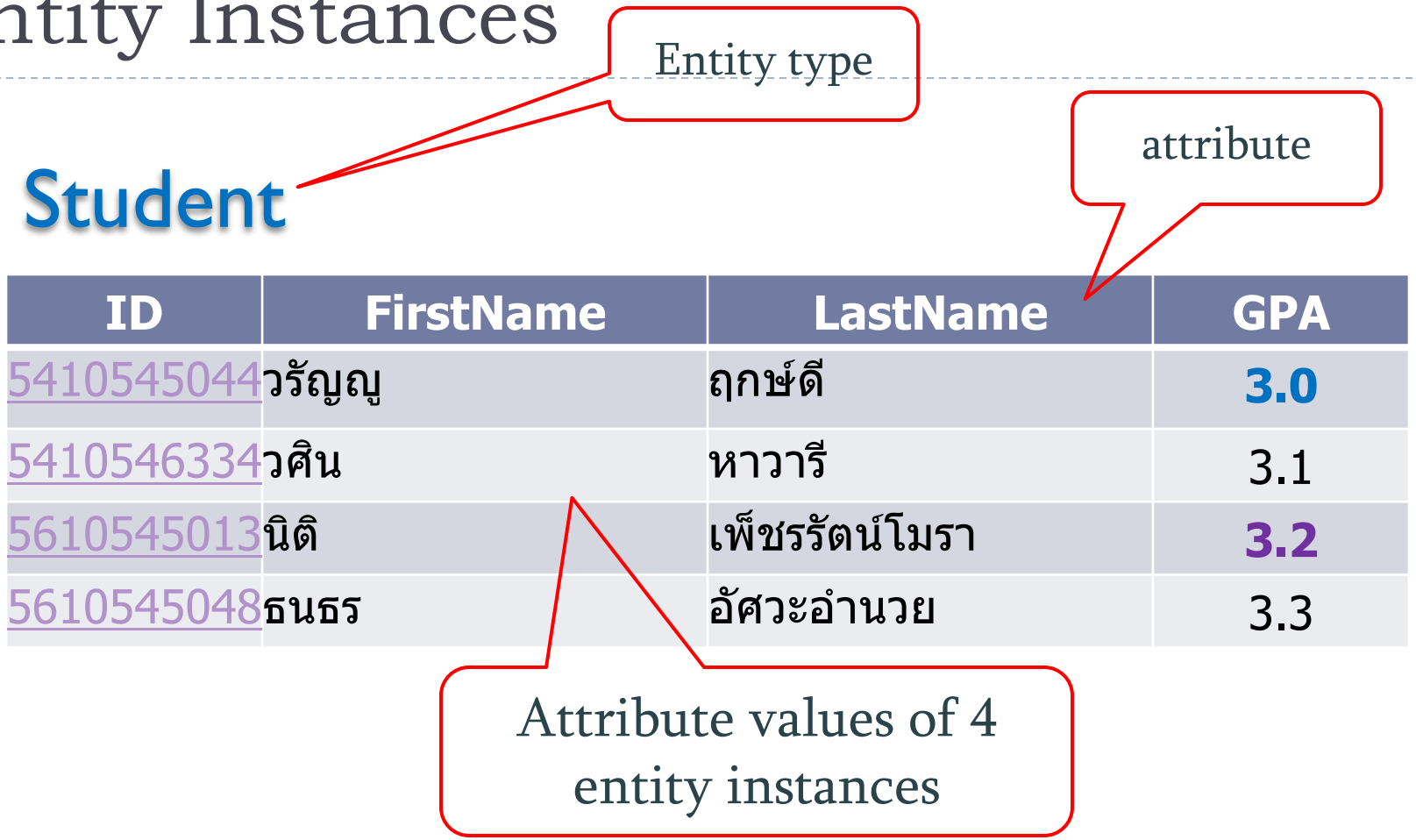
- A collection of things of interest:
  - persons, places, things, events, processes
- Contains attributes with identifier(s)
  - characteristics/properties of things (entities)



Entity Instance

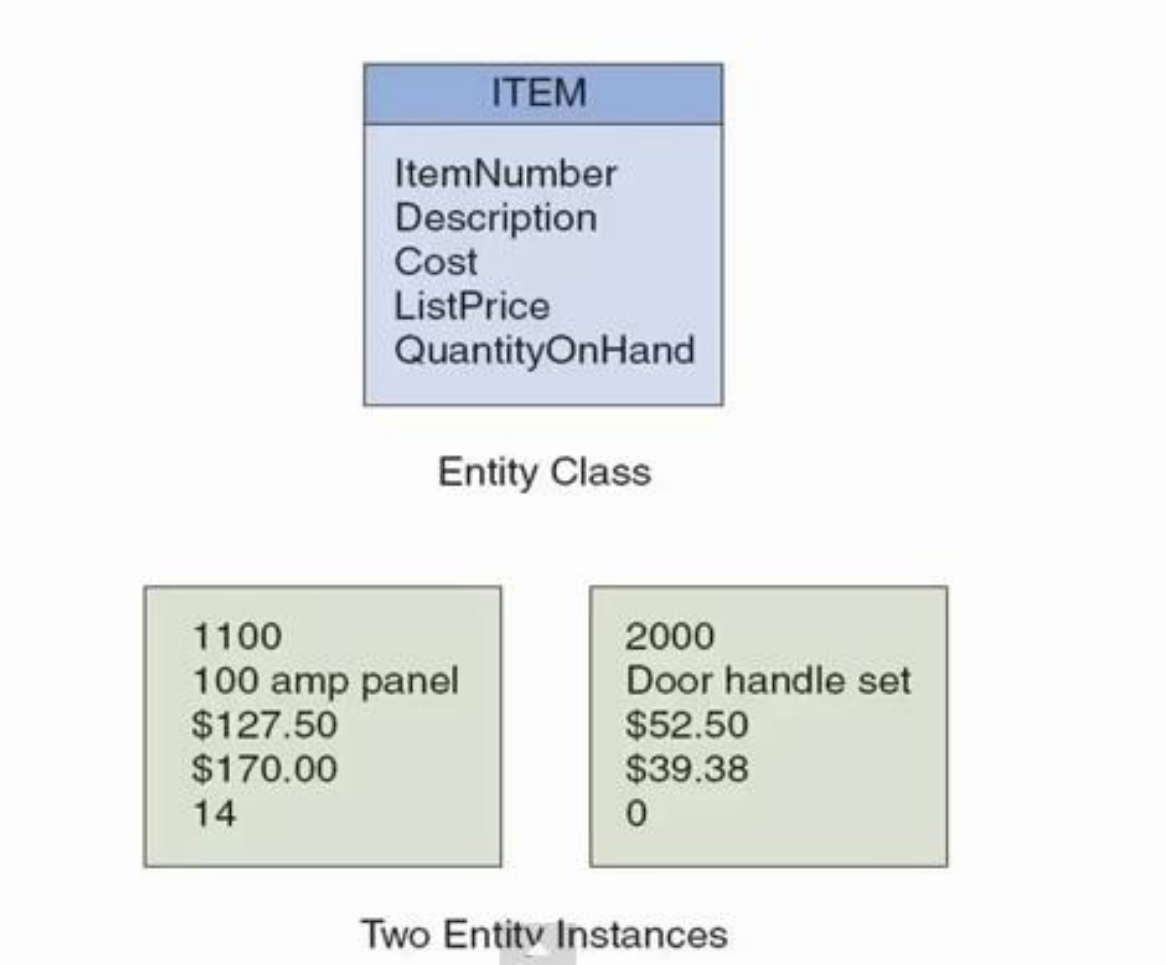
- An entity
- An entity instance is a specific occurrence of an entity type.
  - a member of an entity type

Entity Instances



Members of a set must have the same type
Student = {s1, s2, s3, s4}

Example


ITEM
ItemNumber
Description
Cost
ListPrice
QuantityOnHand
Entity Class
1100
100 amp panel
$127.50
$170.00
14
2000
Door handle set
$52.50
$39.38
0
Two Entity Instances

Attributes

- Describe characteristics of entities
- An entity type has attributes which together describe the entity

  Student (ID, FirstName, LastName, *GPA*)
- Each attribute has data type and other properties
  - Key or non-key

Primary keys

- **Entity instances have identifiers (keys)**
- **Keys are a type of attribute**

Entity type : Student

- Contains attributes: ID, FirstName, LastName, GPA
- **Primary Key : ID**
  - **Unique value**
  - **ID can identify a particular instance in the entity type**

Types of Keys

- Uniqueness
  - Keys may be unique or non-unique
  - If the key is unique, the data value for the key must be unique among all instances of the entity
    - Candidate keys
    - Primary keys (one for each entity type)
- Composite
  - A composite key consists of two or more attributes
    - E.g., FirstName + LastName

Entity Type vs. Entity Instance

- An entity type is a description of the structure and format of occurrences of the entity
- An entity instance is a specific occurrence of an entity type.

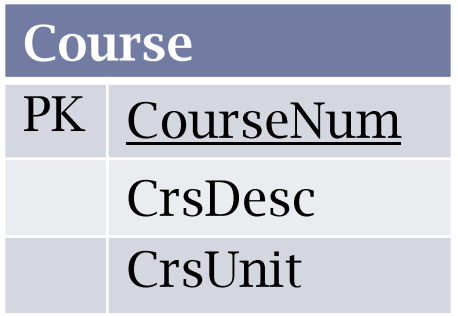
Graphical presentation

•Entity type

•Conceptual design

•Entity type showing only the primary key attribute

•Preliminary design



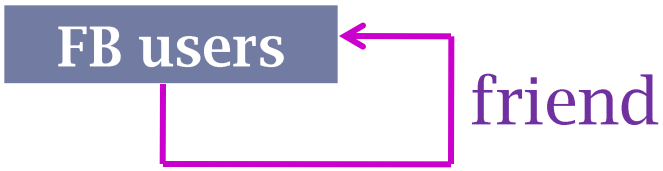
•Entity type showing all attributes

•Final stage

Relationships

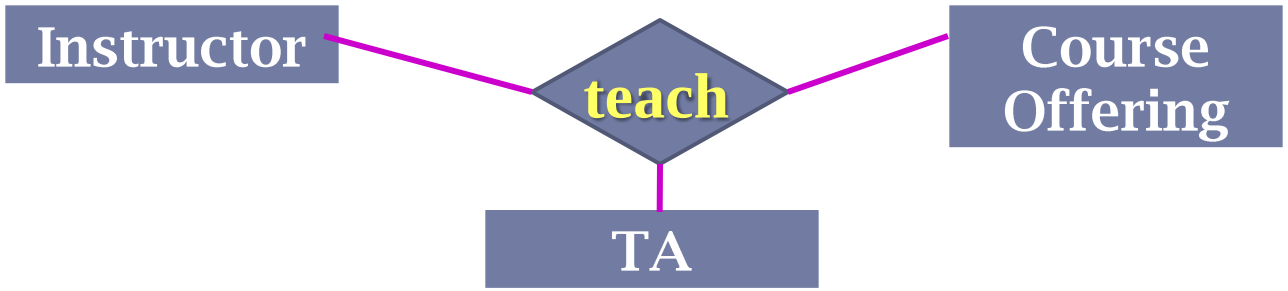
- Association among entity instances:
  - have specific names
- Usually between two entity types
  - Unary
  - Binary
  - Ternary
  - etc. multiple entity types

1.Unary Relationship



2.Binary Relationship

3.Ternary Relationship



Types of Binary Relationships

- One to one
- One to many
- Many to one
- Many to many

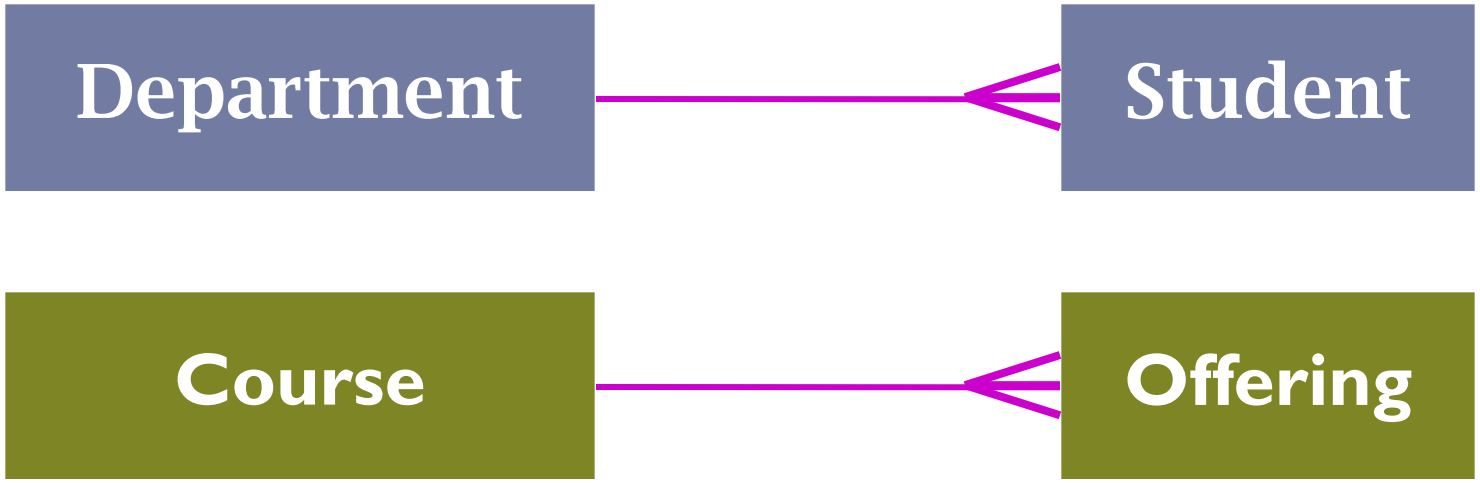
One-to-One Relationship

- **Unary or binary**
- **One-to-one binary relationship**
  - **An entity instance in one entity type is related to an entity instance in another entity type**
    - Example
      - A student may have no more than one locker
      - A locker may only be used by one student

Student — Locker

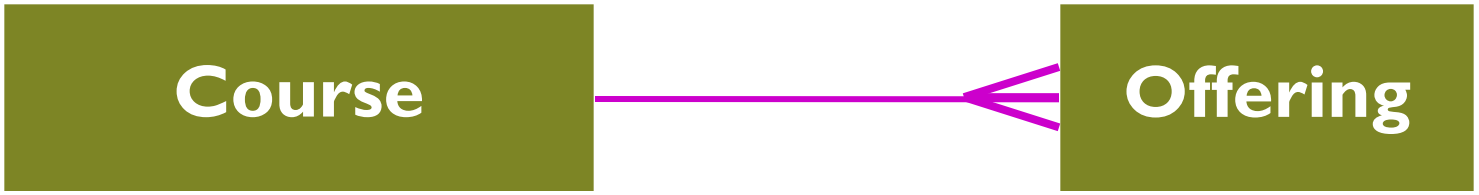
One-to-Many Binary Relationship

- 1:N (one-to-many)
  - An instance in one entity type is related to many entity instances in another entity type
    - A student studies in one department
    - A department can have many students



Bidirectional Relationship

- Can be used to navigate in both directions
- Two names
  - Course to Offering: Has, Provides *
  - Offering to Course: IsProvidedFor
- Which name to use: try to use active* verb;
  - not always possible



Ex. Course 351 has three offerings

N:1, Offering to Course

1:N, Course to Offering

1:N relationship

Course

ID	Name
101	Database
102	OS
103	Network

Offering

OfferNo	Semester	Year
1	1	2014
2	2	2015
3	1	2015

1.Each course has many offerings

ID	Name
101	Database
102	OS
103	Network

OfferNo	Semester	Year
1	1	2014
2	2	2015
3	1	2015

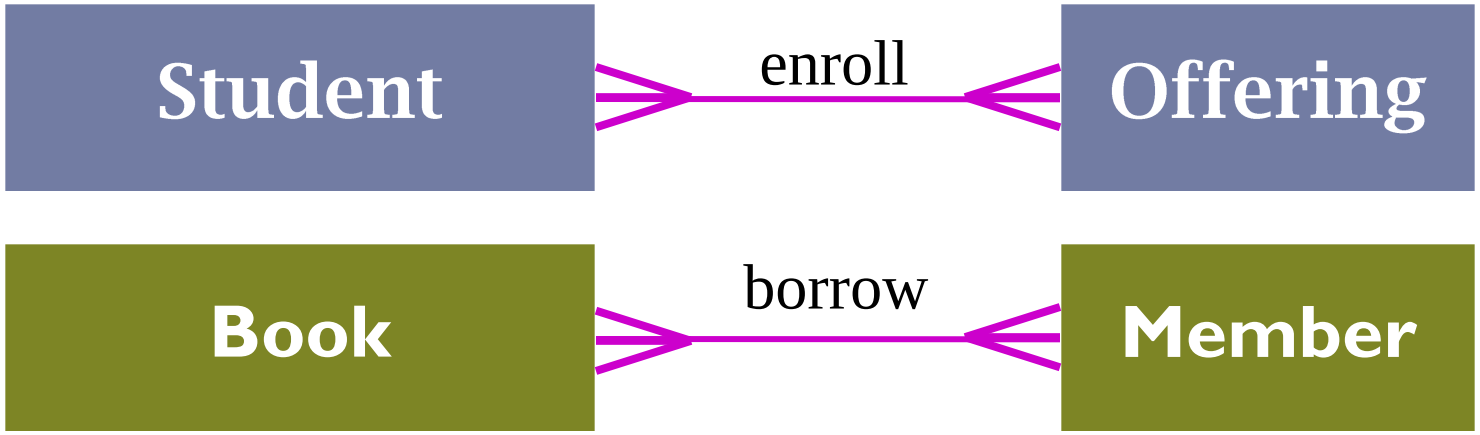
2.Each offering is provided for only one course

1:N

Many-to-Many Binary Relationship

- N:M (conceptual many-to-many)
  - An instance in entity type A can be related to many entity instances in entity type B

  and
  - An instance in entity type B can be related to many entity instances in entity type A



N:M relationship

Student

ID	Name
101	Sarah
102	Tom
103	Tim

Offering

OfferNo	Semester	Year
1	1	2014
2	2	2015
3	1	2015

1.Each student can enroll in **many** offerings

ID	Name
101	Sarah
102	Tom
103	Tim

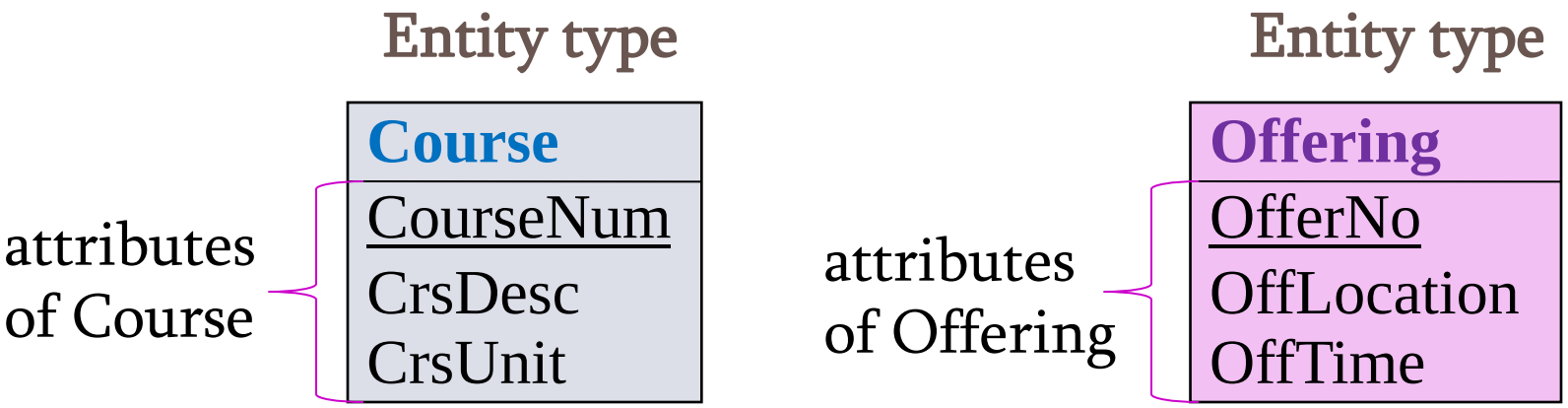
OfferNo	Semester	Year
1	1	2014
2	2	2015
3	1	2015

2.Each offering can be enrolled by **many** students

N:M

How to represent relationships?

- In E-R model, each entity type consists of its properties only !



Where is the relationship?

Data Representation

- Database models
  - e.g., Relational data model
- Entity Relationship Model
  - Conceptual model
  - No physical data representation
  - Entity Relationship Diagram
    - rectangles represents entity types and attributes
    - relationships represented by lines linking entities

Cardinality

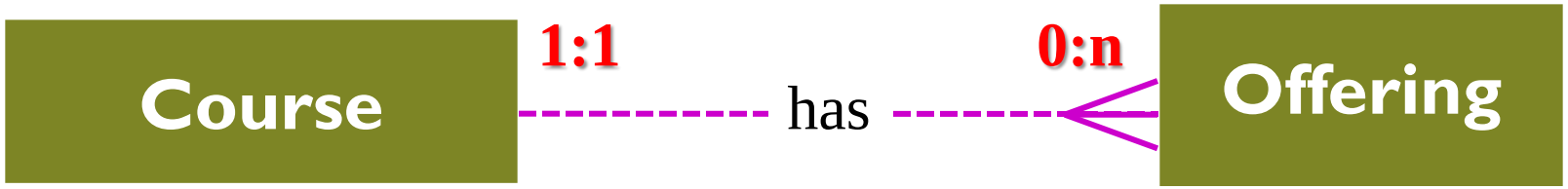
Data Validation

Cardinality of Relationships

- Cardinality means count
  - as a number of members in a set
- Specify the number of entity instances that can participate in a relationship instance
  - minimum : 0-n
  - maximum : 1- n

0,1,2,3,… or functional

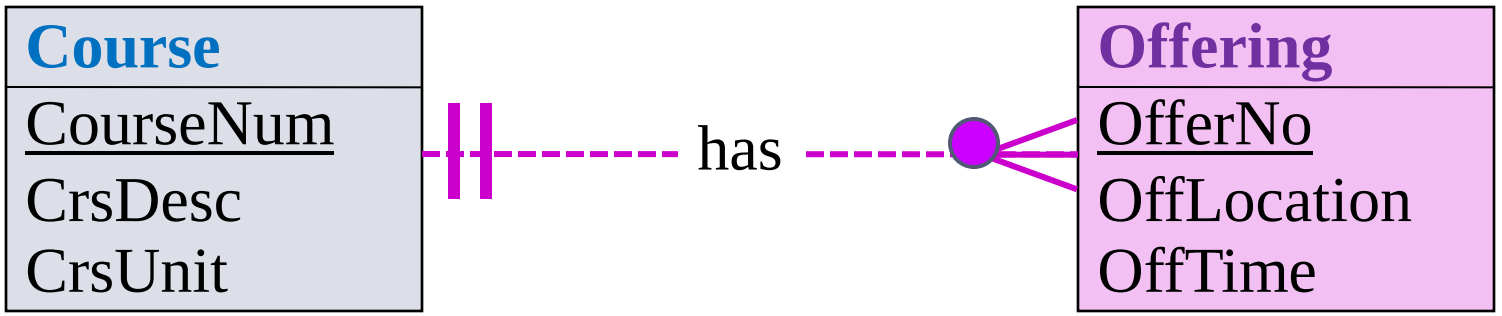
Questions about data validation



1. How many times that a course can be offered?
   - Minimum = 0 (a course can exist without relating to any offering)
   - Maximum = N (no upper bound)
     - can be 1,2,3,10 or functional
2. Does each offering has to be related to any course?
   - Yes, it is
   - Each course relates to a minimum of one course and a maximum of one course as well.

E-R Diagram

- Represent E-R model
- Entity and attributes
  - a rectangle with entity name and attributes
  - underlined attribute is a primary key
- Relationship
  - dashed line with a name
- Cardinality



Cardinality Notation (ERD)

- A vertical line represents one (1)
- A circle represents zero (0)
- A crow's foot represent many (N)
- Any integers
- Function names

Crow's foot

max:min

min:**max**

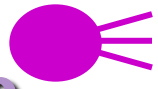
1:0

0:1

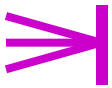
1:1

1:1

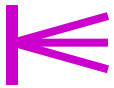
n:0



0:n

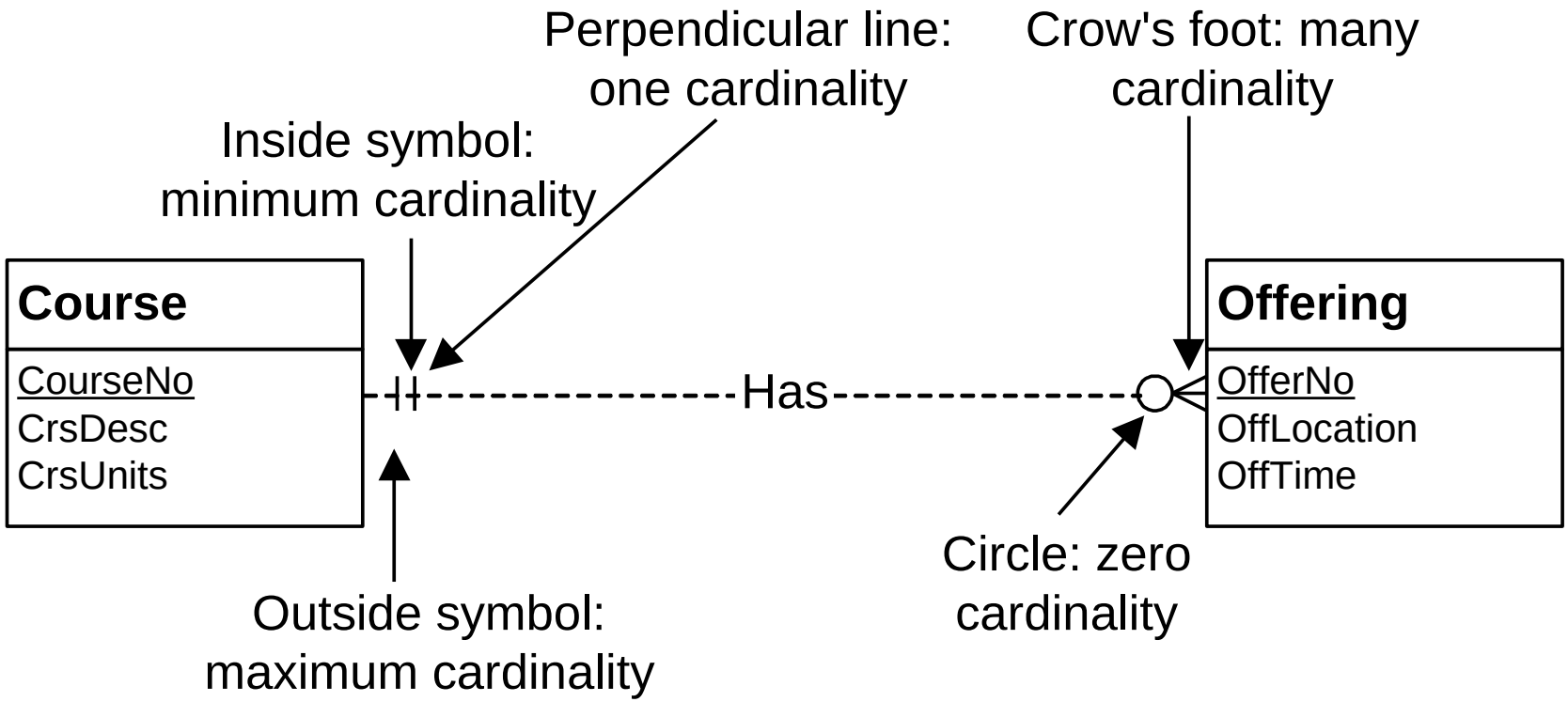


n:1



1:n

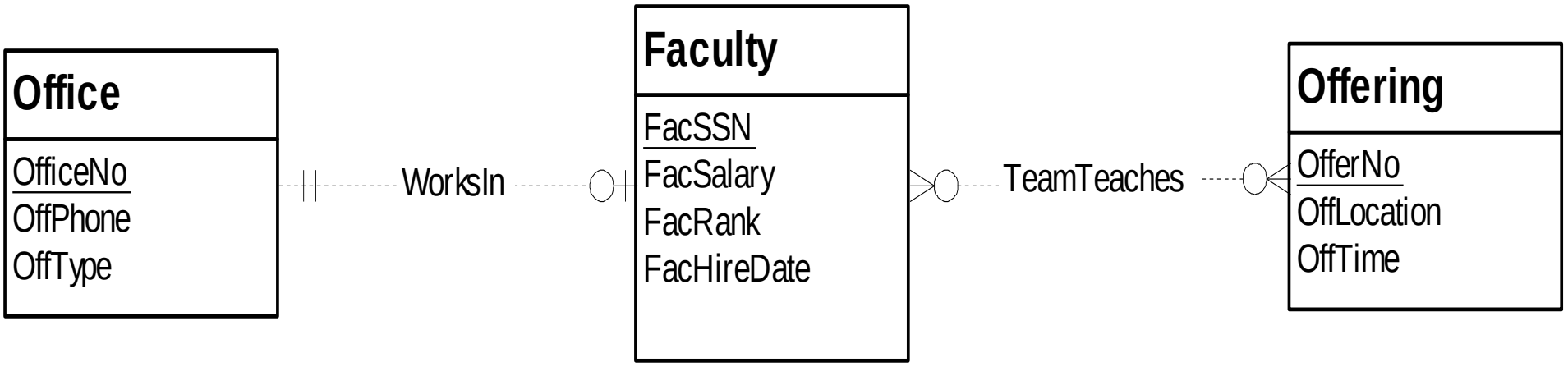
Cardinality Notation



Summary of Cardinalities

Classification	Cardinality Restrictions
Mandatory	Minimum cardinality ≥ 1
Optional	Minimum cardinality = 0
Functional or single-valued	Maximum cardinality = 1
1-M	Maximum cardinality = 1 in one direction and Maximum cardinality > 1 in the other direction.
M-N	Maximum cardinality is > 1 in both directions.
1-1	Maximum cardinality = 1 in both directions.

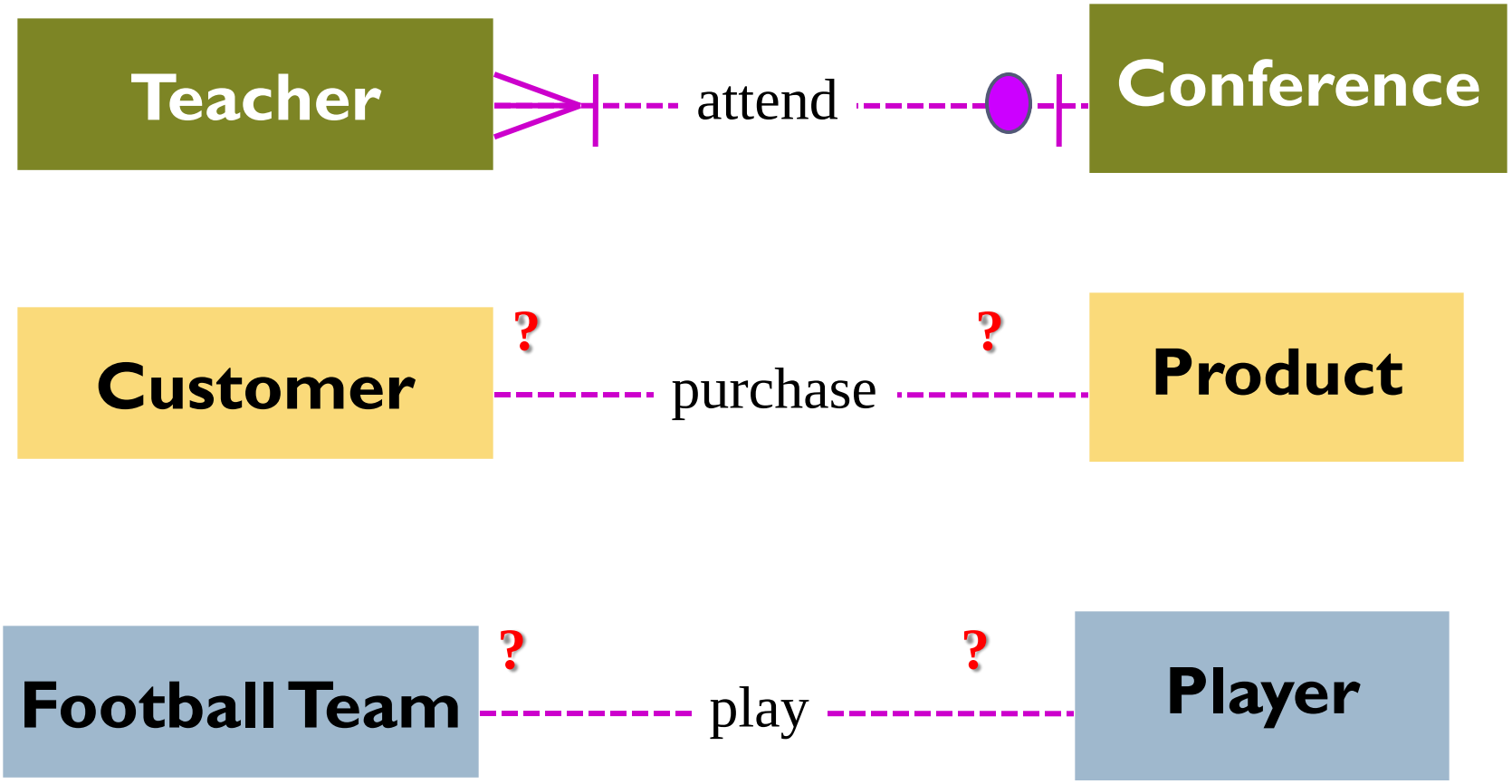
More Relationship Examples



WorksIn:
- 1-1
- Optional: office can be empty
- Mandatory: faculty must be assigned to an office

TeamTeaches:
- M-N
- Optional in both directions

More Relationship Examples



More on Relationships

Next week

Understanding Relationships

1. Identification dependency
2. M-N relationships with attributes
3. Self identifying relationships
4. M-way relationships
5. Equivalence between M-N and 1-M relationships

Weak and Strong Entities

- Weak entity
  - Can not exist without an existence of an instance of another entity
- Strong Entity
  - Any entity that is not a weak entity is called a strong entity

Room cannot exist unless associated building exists

Building

A room must be in a building

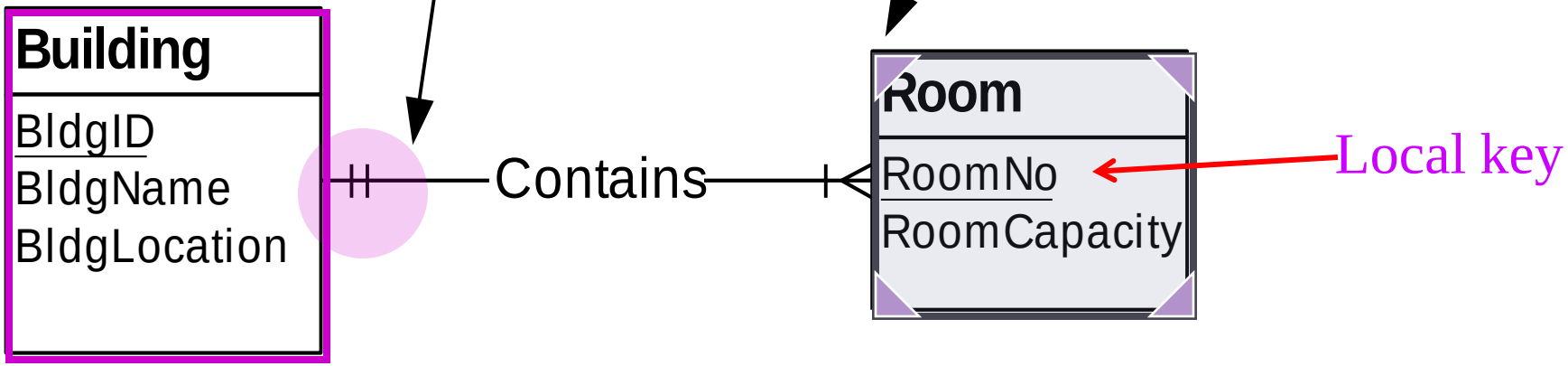
1.Identification Dependency

- Identification dependency involves existence dependency:
  - Weak entity is existent dependent on other entity
  - A weak entity borrows all or part of its primary key from other entity types
- Concept
  - Closely related entities: physical containment
  - Ex: Province & district, order-orderline

Identifying Relationship

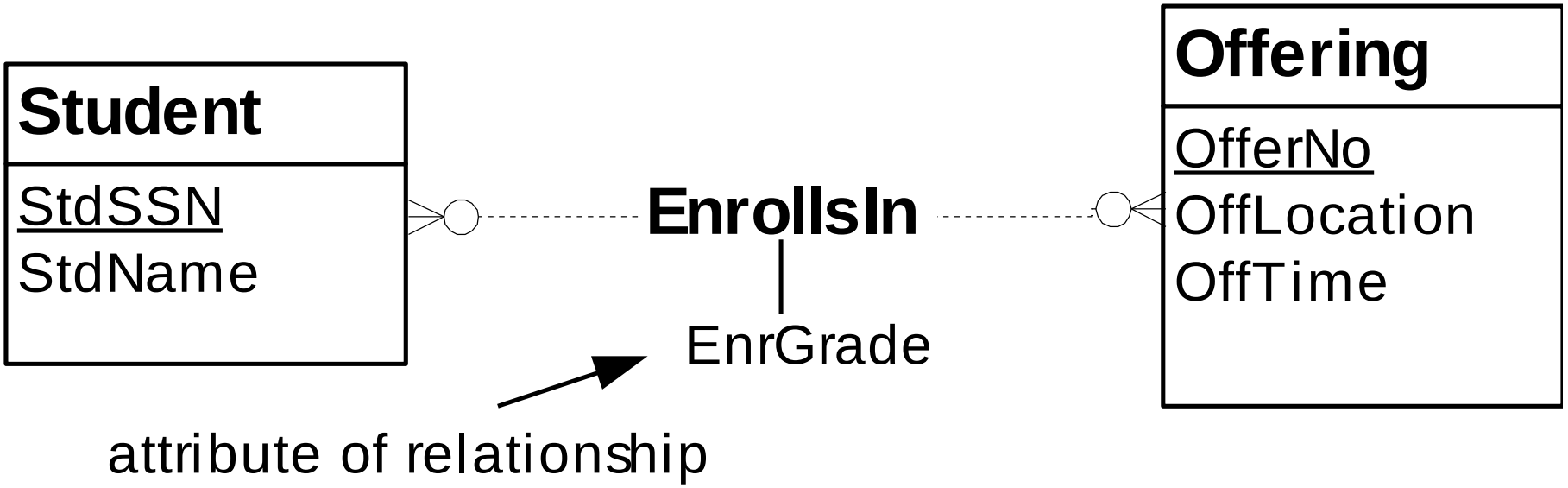
Identification Dependency Symbols:

- Solid relationship line for identifying relationships *Indicates the source of PK*
- Diagonal lines in the corners denote weak entities. *Borrows part or all of PK*



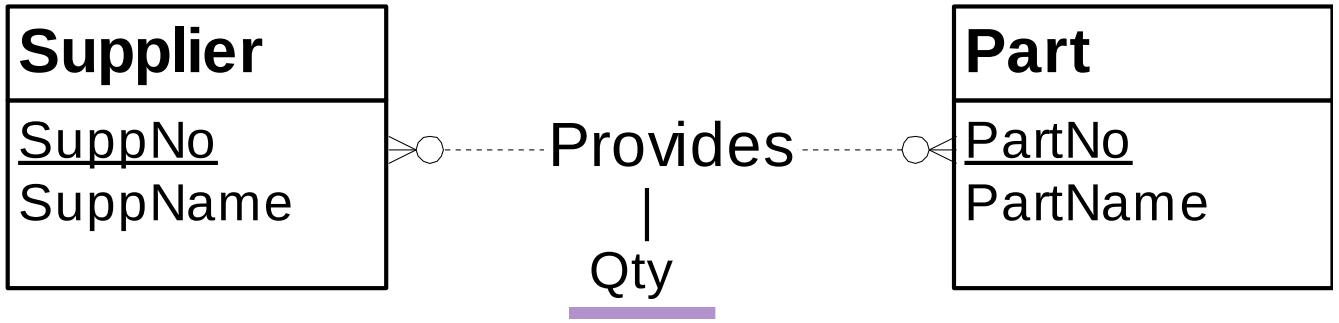
PK of Room is a combination of RoomNo (local key) and BldgID (borrowed attribute)

2.M-N Relationships with Attributes

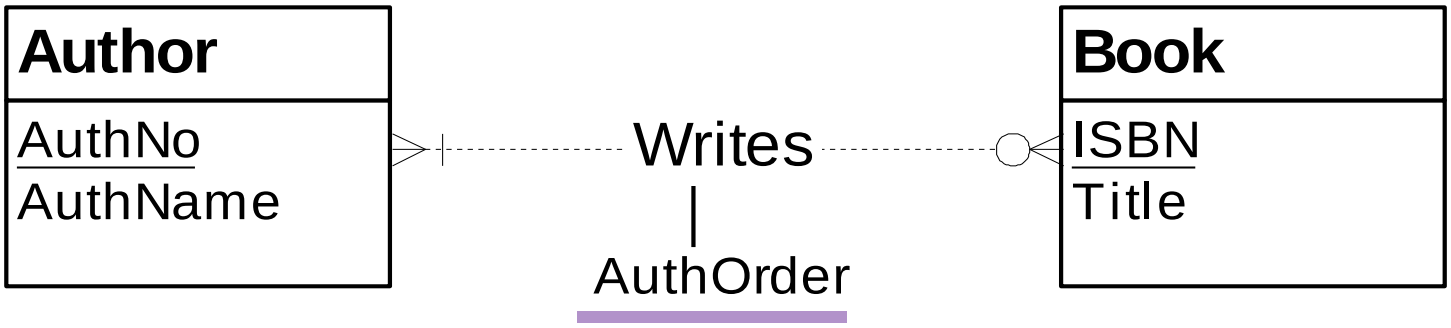


M-N Relationships with Attributes (II)

a) *Provides* relationship

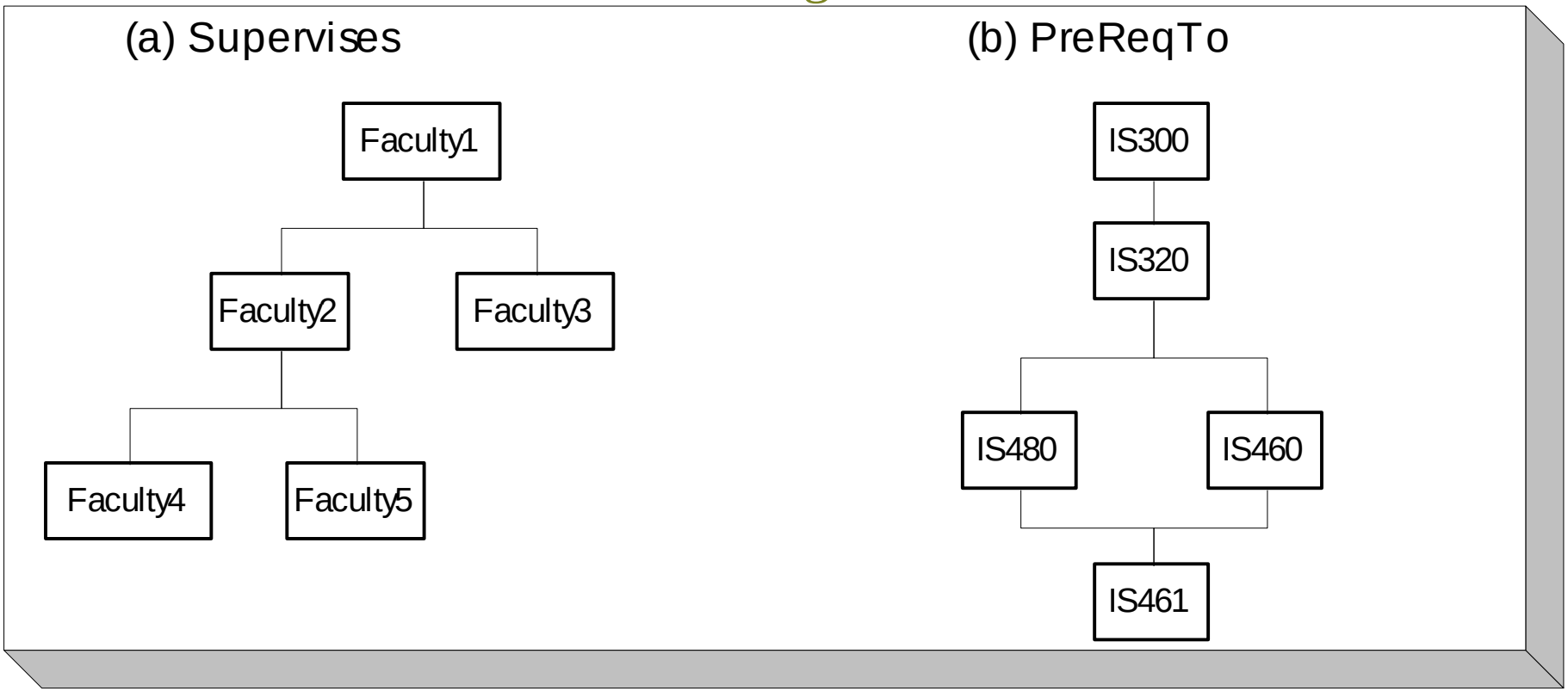


b) *Writes* relationship



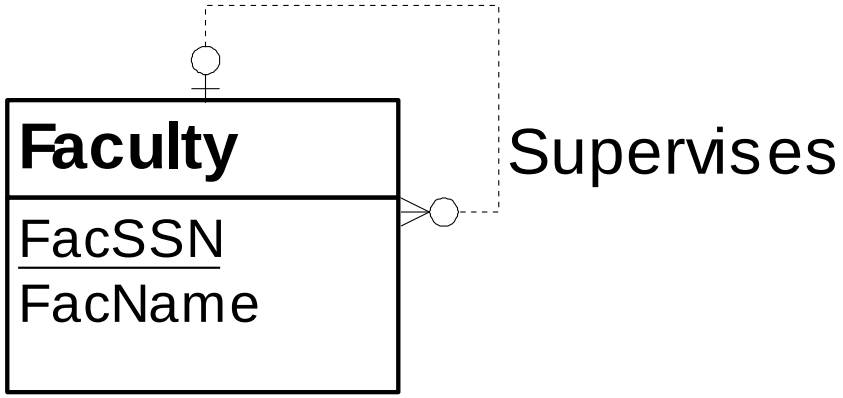
3. Self-Identifying Relationships

Instance Diagrams

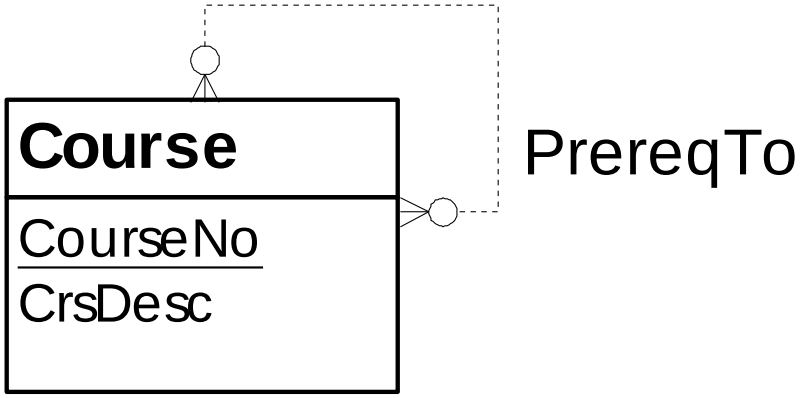


ERD for Self-Referencing Relationships

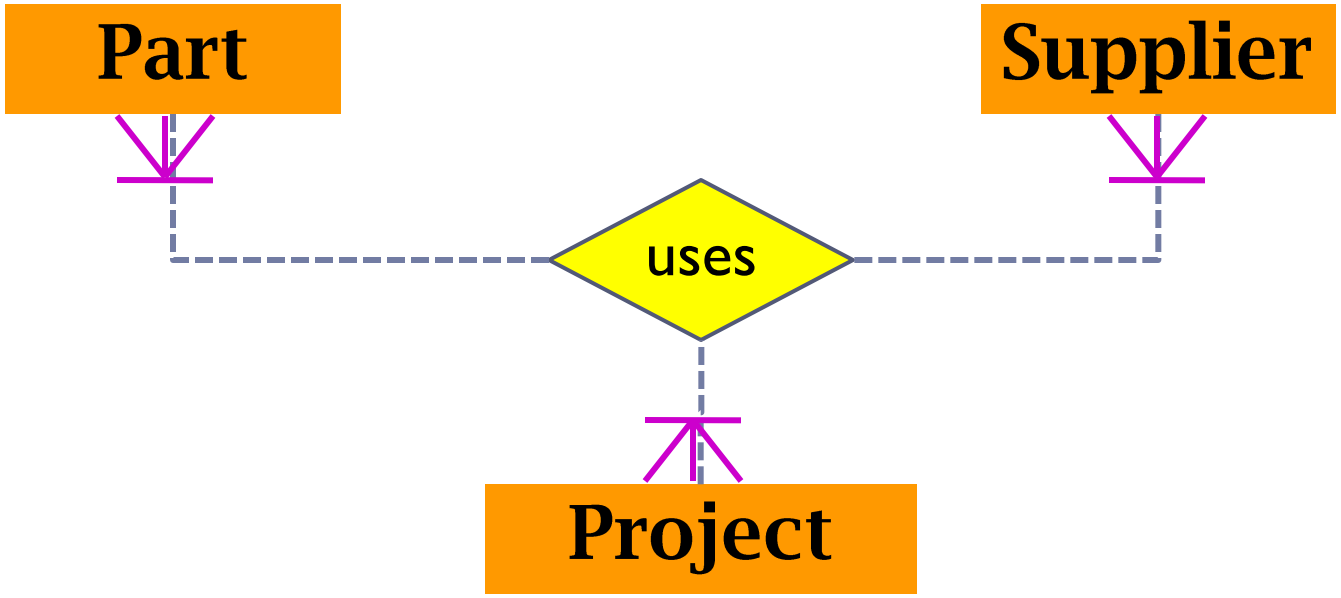
a) manager-subordinate



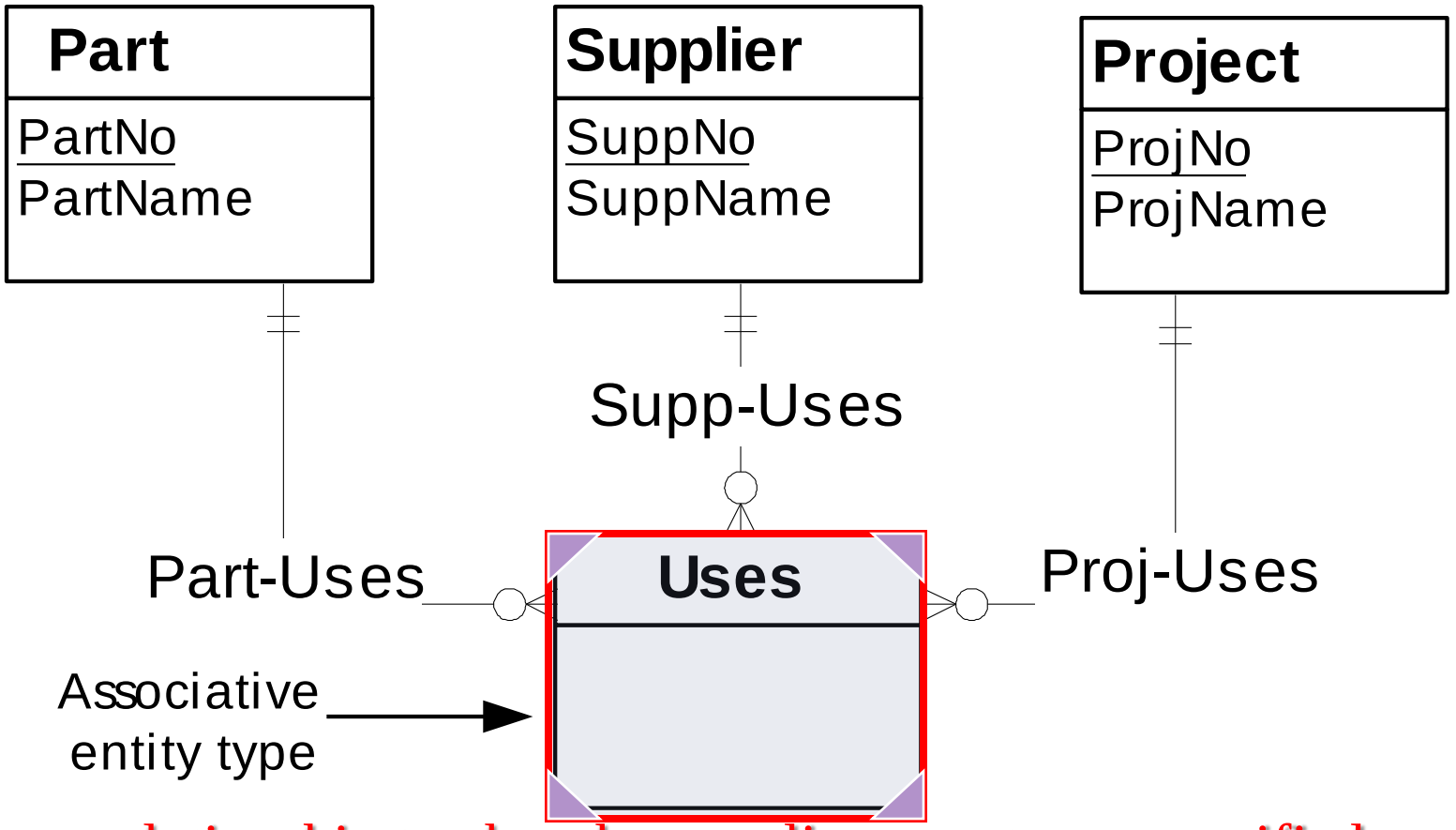
b) course prerequisites



4.M-way Relationships



Associative Entity Types for M-way Relationships



3 way relationship tracks who supplies a part on a specified project

Example

Uses

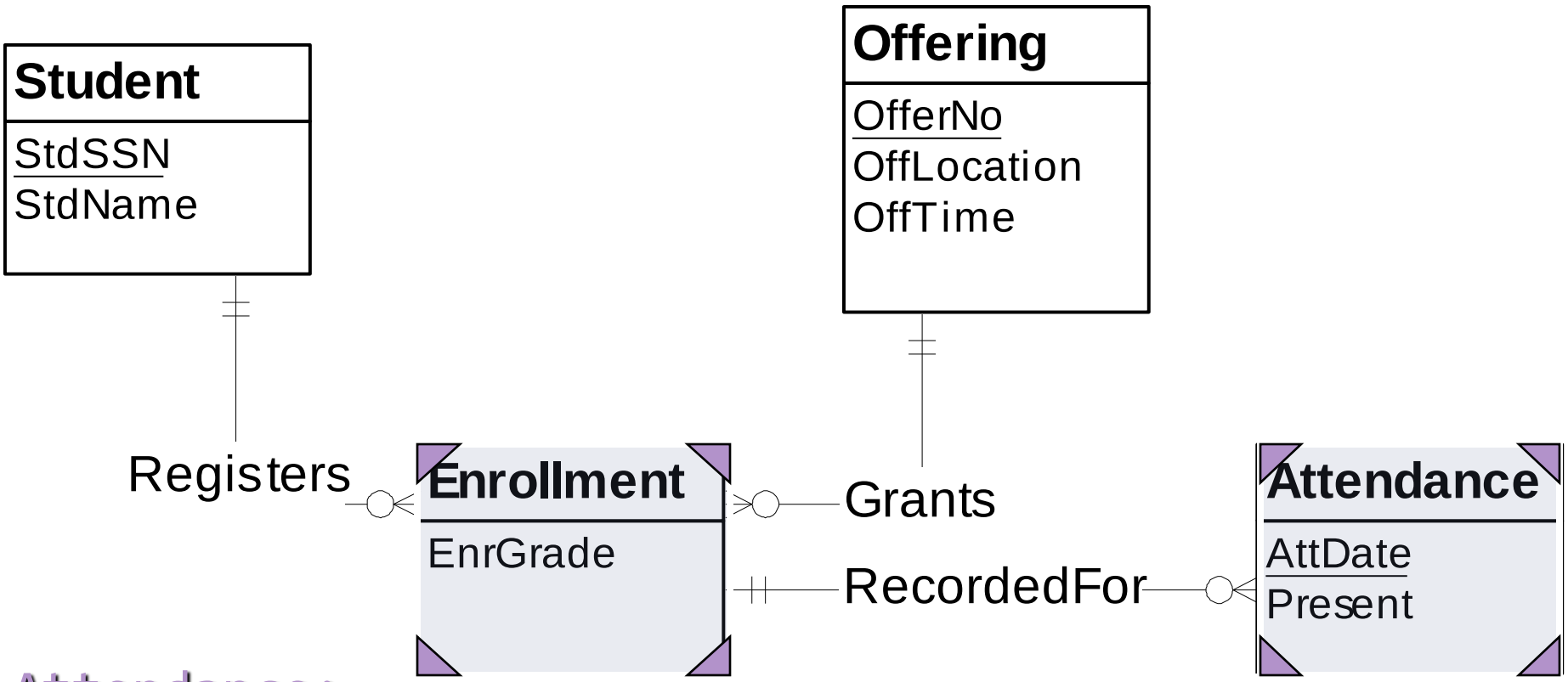
Project	Part	Supplier
Pancake	Egg	7-11
Pancake	Milk	Lotus
Pancake	Flour	7-11
Pizza	Flour	7-11
Pizza	Chicken breast	BigC

Note: Primary keys must be used in this associative entity

5.Relationship Equivalence

- Replace M-N relationship
  - Associative entity type
  - Two identifying 1-M relationships
- M-N relationship versus associative entity type
  - Largely preference
  - Associative entity type is more flexible in some situations

Associative Entity Type Example



Attendance:

- Weak entity
- PK: Combination of AttDate and PK of Enrollment

Must use associative entity type for Enrollment rather than M-N relationship

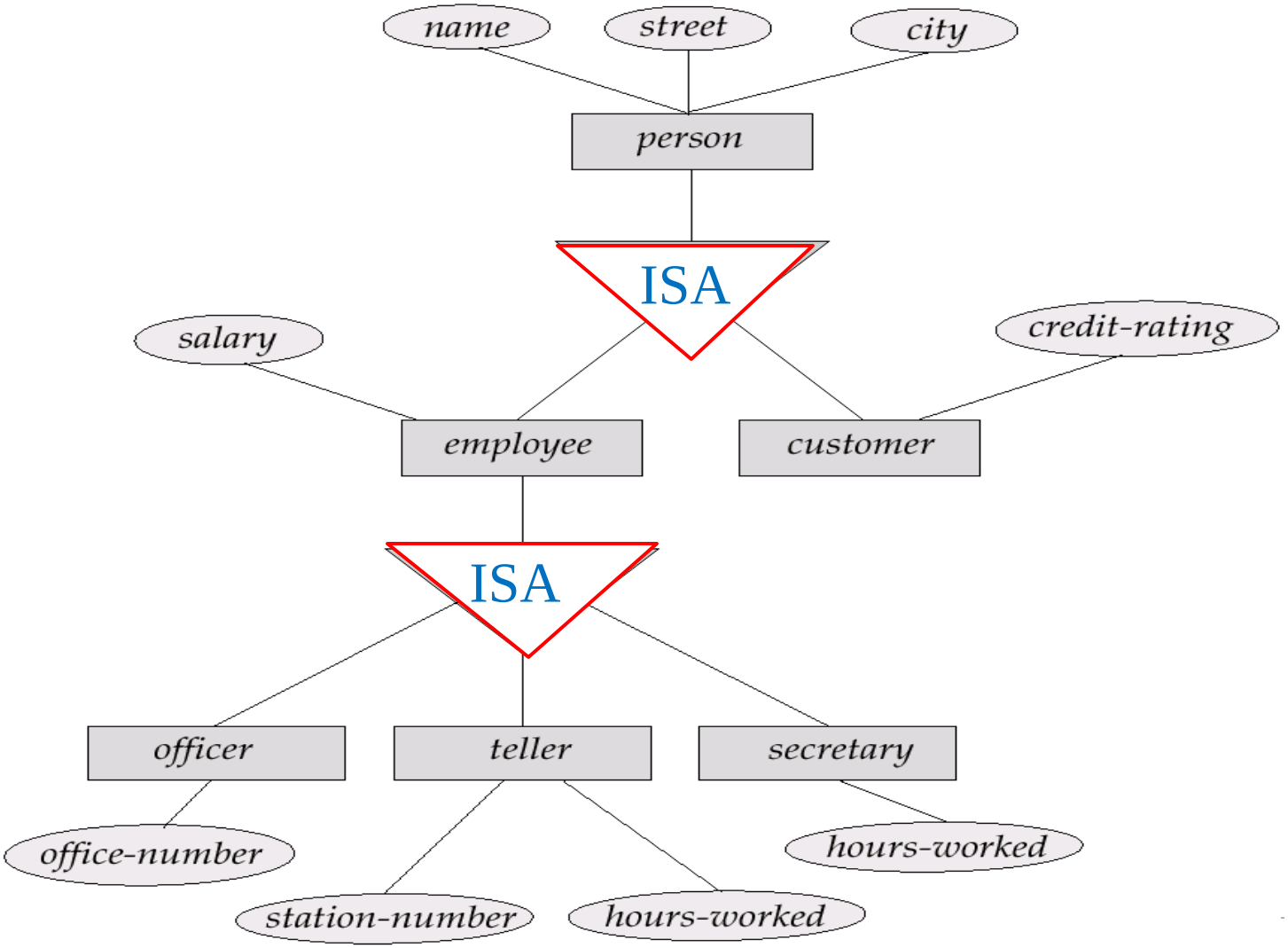
Generalization

Specialization

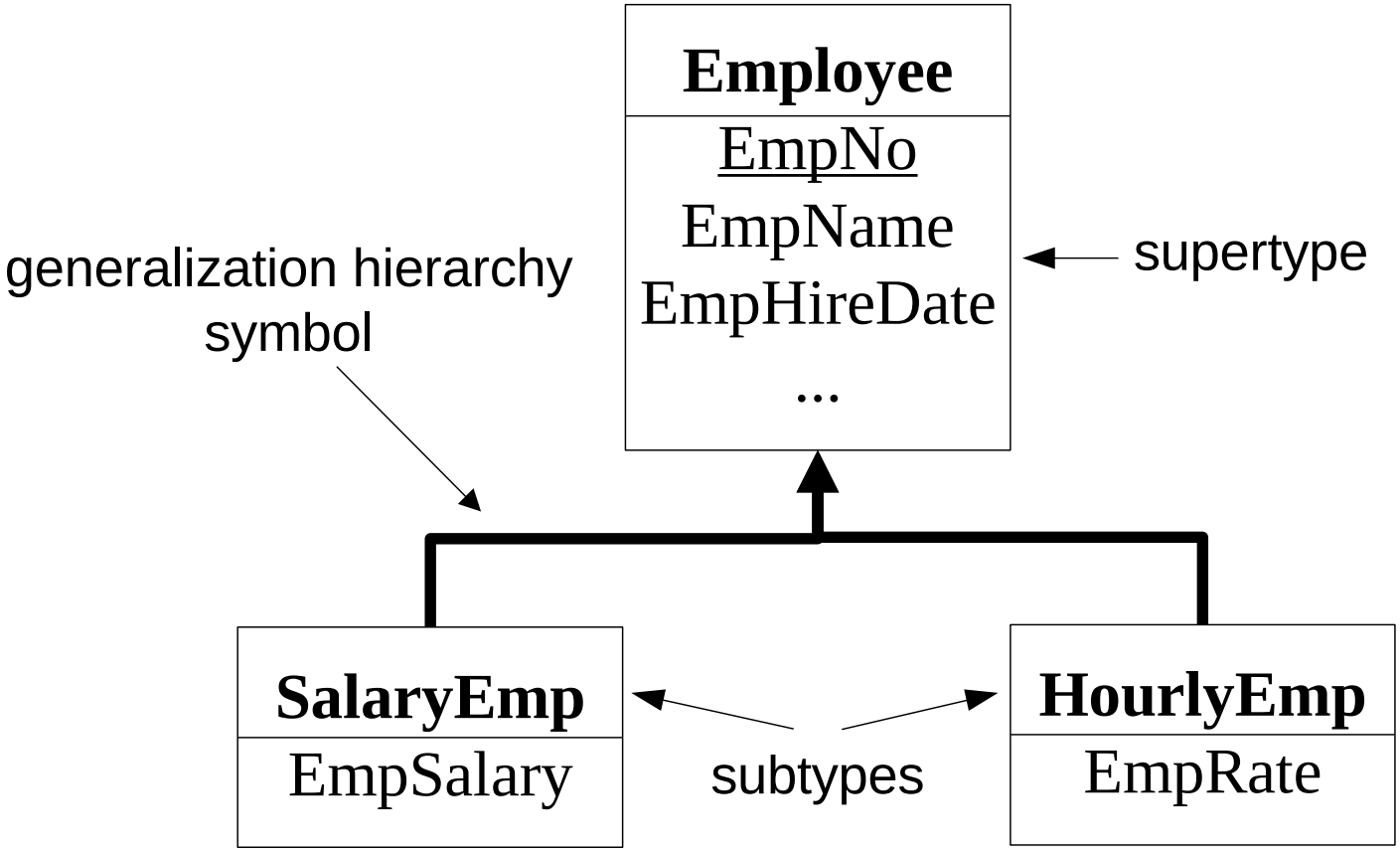
Generalization

- A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set.
- 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.

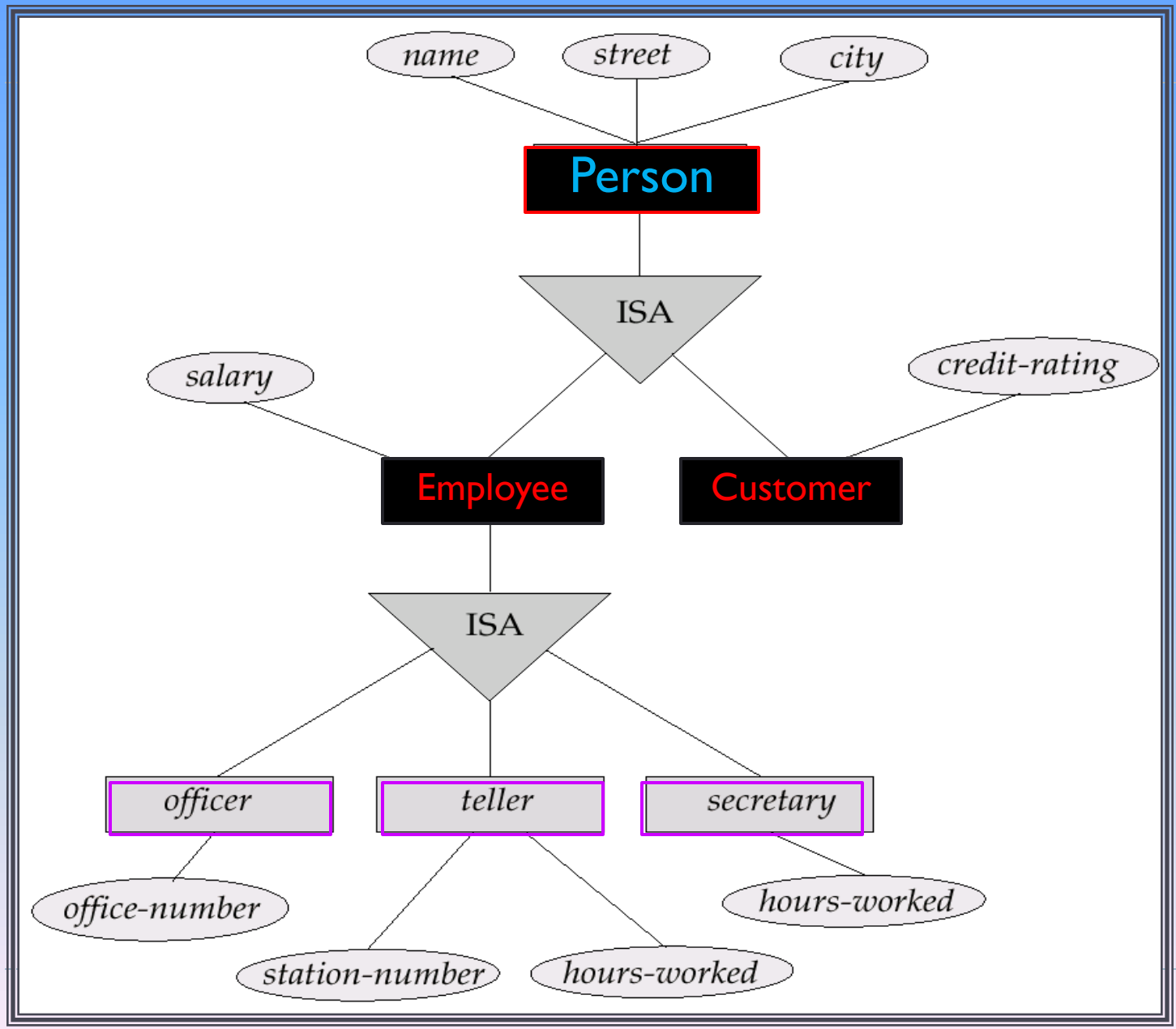
- The **ISA** relationship also referred to as **superclass - subclass** relationship



Generalization Hierarchies



Example

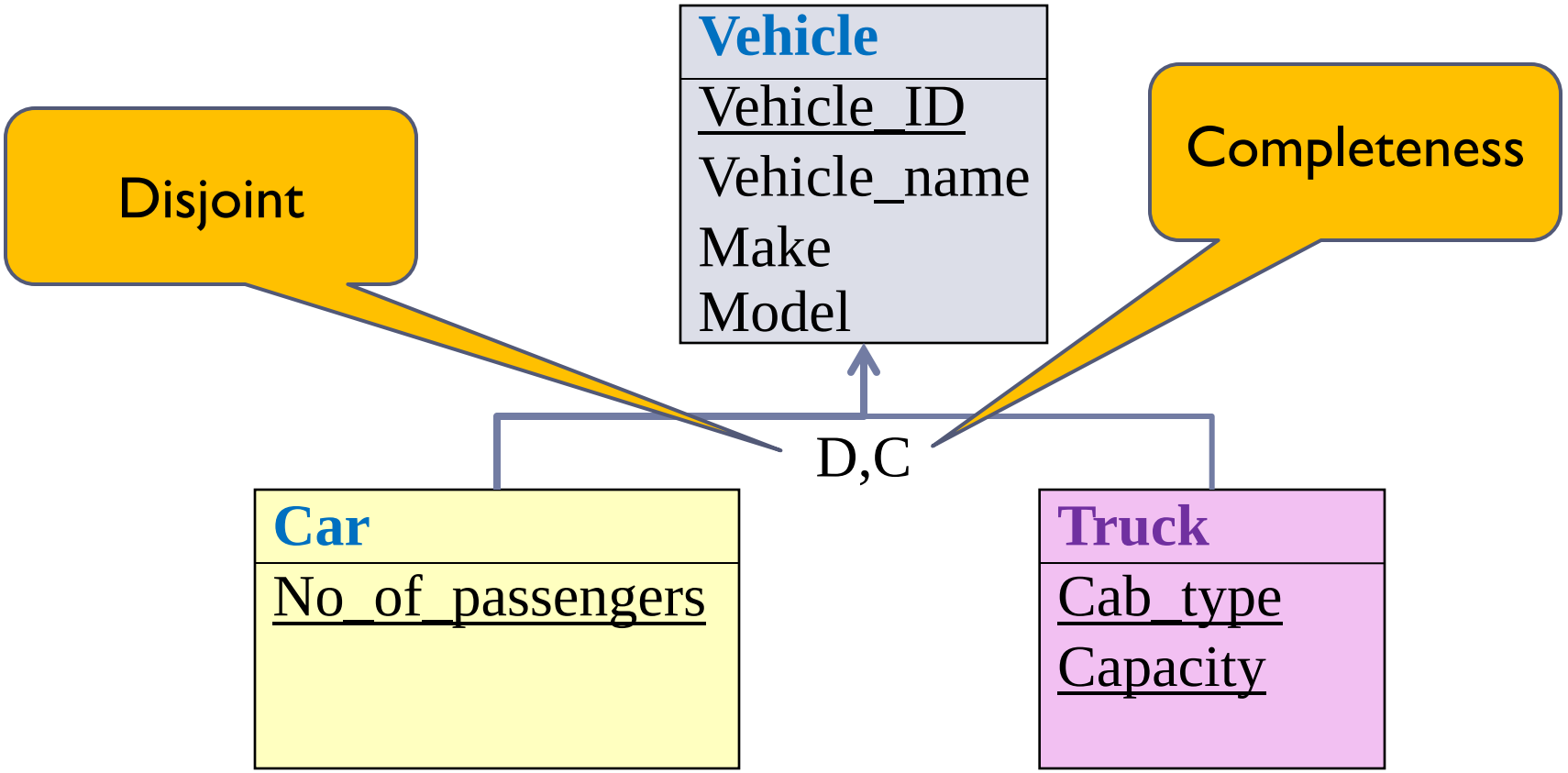

name
street
city
Person
ISA
salary
credit-rating
Employee
Customer
ISA
officer
teller
secretary
office-number
station-number
hours-worked
hours-worked

Inheritance

- Subtypes inherit attributes of supertypes *(direct and indirect)*
- Allows **abbreviation** of attribute list
- Applies to code (methods) as well as attributes (data)

Reduce the amount of code by inheriting code from similar objects

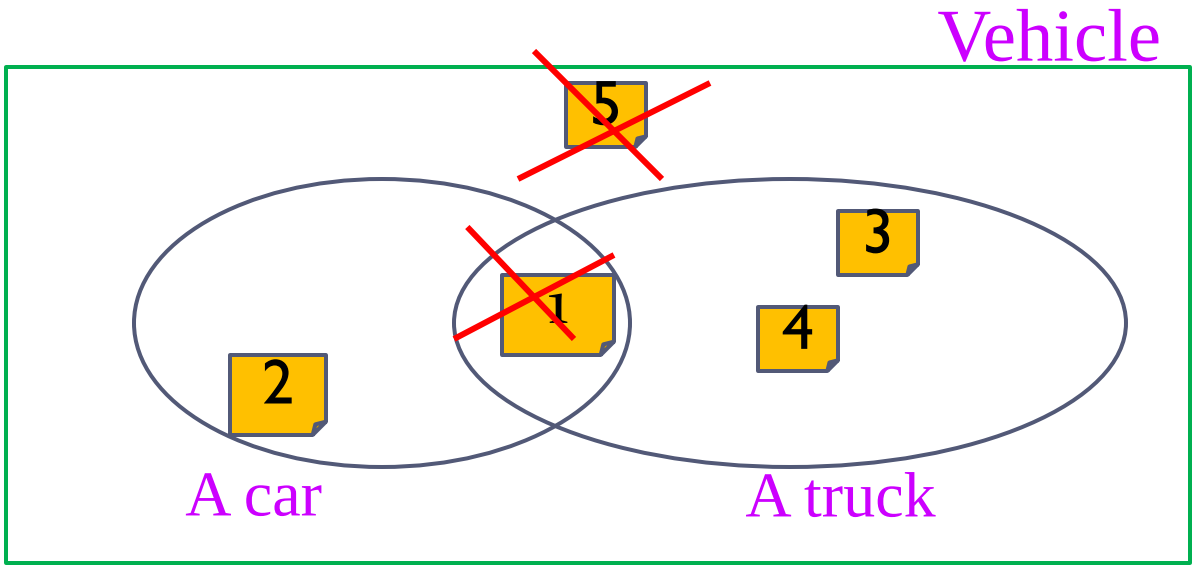
Generalization Constraints



Disjoint: A vehicle can not be both a car and a truck.

Complete: Every vehicle must be either a car or a truck.

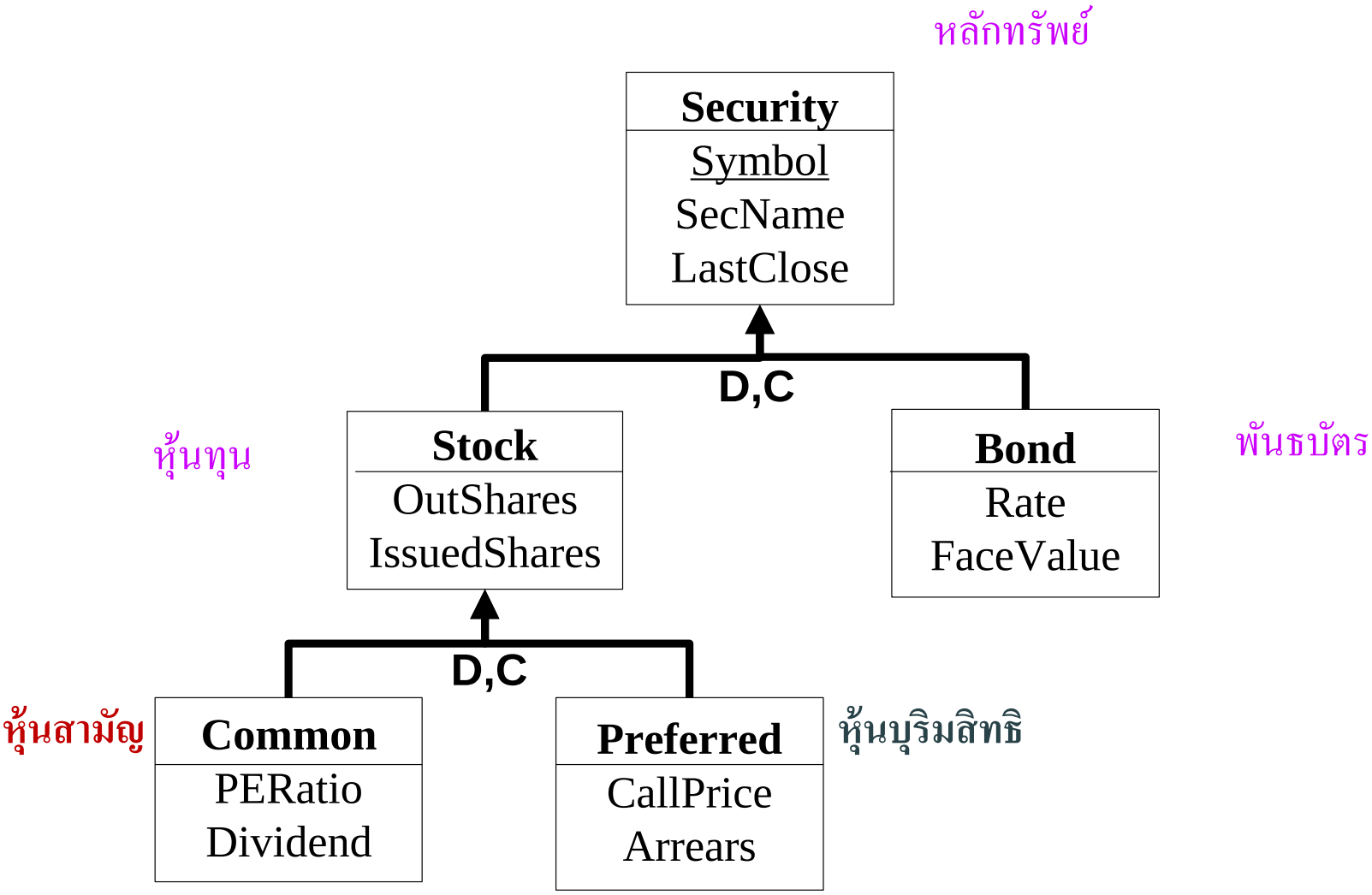
Generalization Constraints



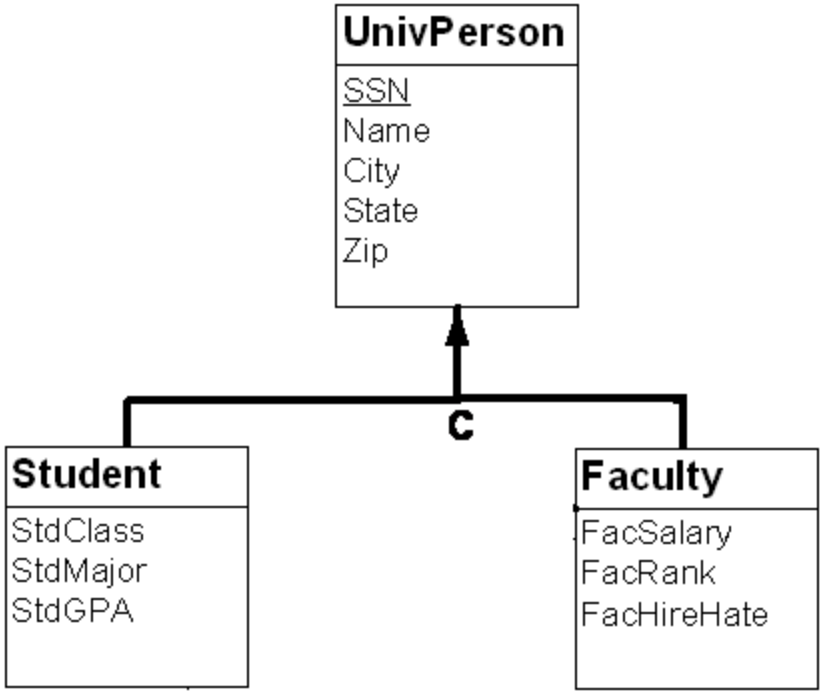
Disjoint: A vehicle can not be both a car and a truck (1).

Complete: Every vehicle must be either a car or a truck (5).

Multiple Levels of Generalization



Generalization Constraints



Complete: Everyone must be either a student or a faculty member.

A faculty member can be a student too so it is not disjointed

Comprehensive Example

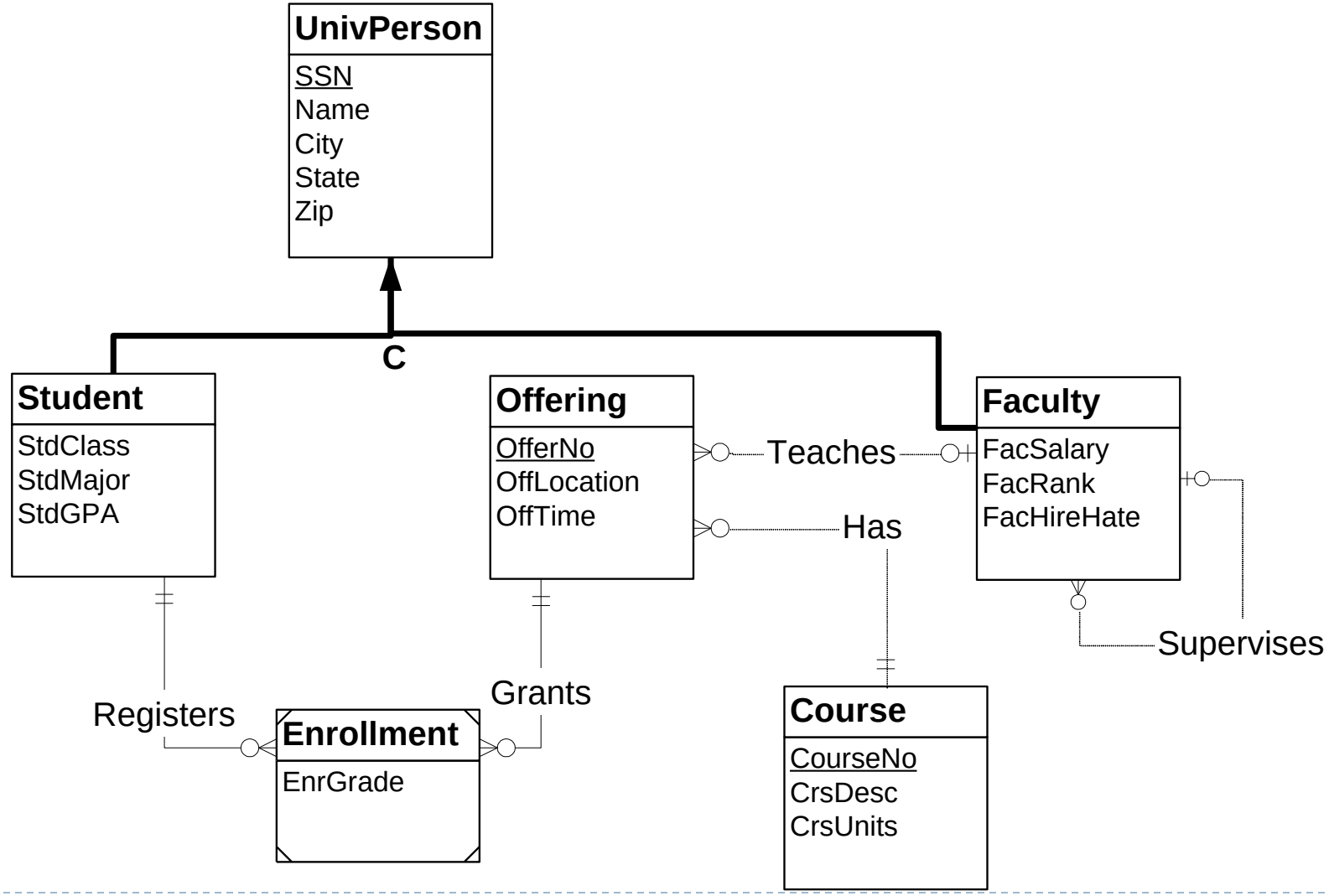


Diagram Rules

- Ensure that ERD notation is correctly used
- Similar to syntax rules for a computer language
- Completeness rules: no missing specifications
- Consistency rules: no conflicts among specifications

Completeness Rules

- Primary Key Rule: all entity types have a PK (direct, indirect, or inherited)
- Naming Rule: all entity types, relationships, and attributes have a name
- Cardinality Rule: cardinality is specified in both directions for each relationship
- Entity Participation Rule: all entity types participate in an at least one relationship except for entity types in a generalization hierarchy
- Generalization Hierarchy Participation Rule: at least one entity type in a generalization hierarchy participates in a relationship

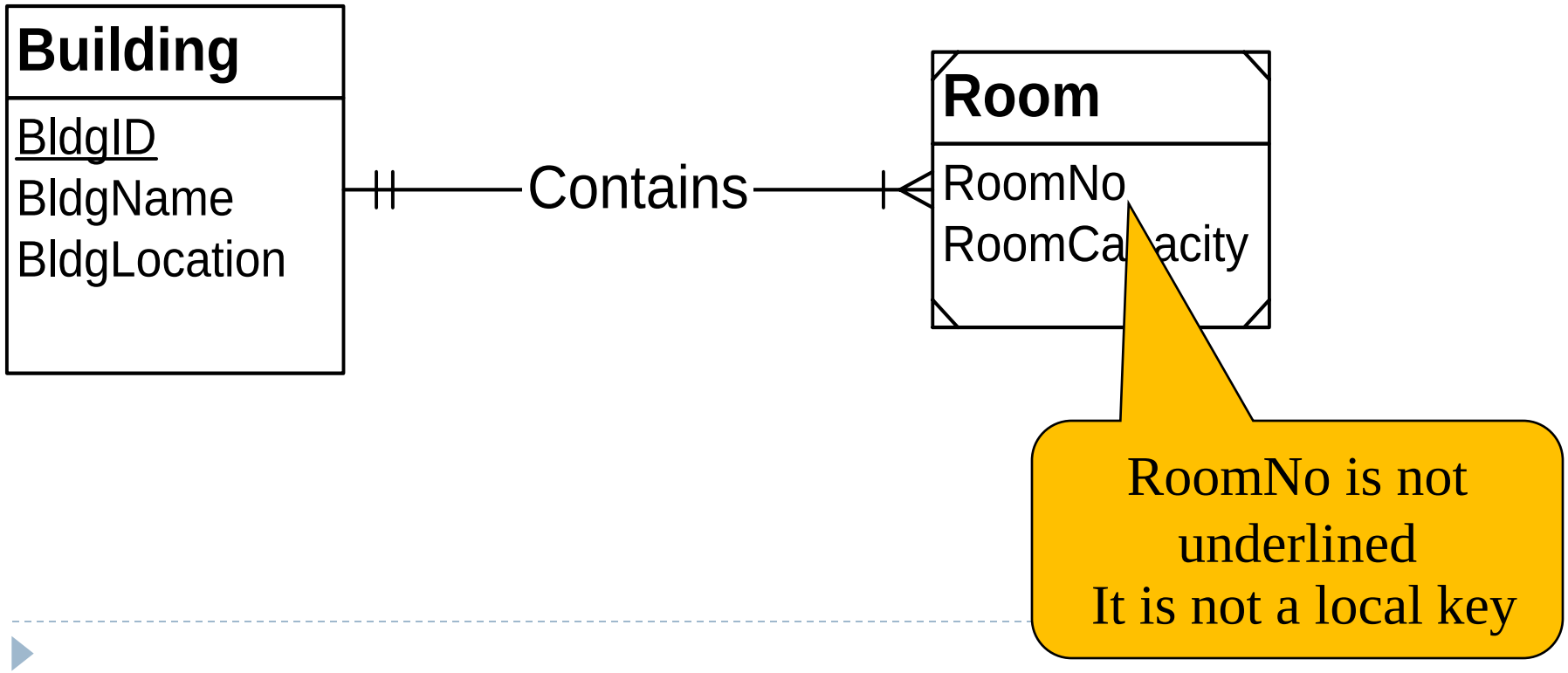
Primary Key Rule Issue

- Primary key rule is simple in most cases
- For some weak entities, the PK rule is subtle
  - Weak entity with only one 1-M identifying relationship
  - Weak entity must have a local key to augment the borrowed PK from the parent entity type
  - Violation of PK rule if local key is missing

PK Rule Violation Example

PK rule violation

- A single 1-M identifying relationship
- Room does not have a local key.



Naming Consistency Rules

- Entity Name Rule: entity type names must be unique
- Attribute Name Rule: attribute names must be unique within each entity type and relationship
- Inherited Attribute Rule: attribute names in a subtype do not match inherited (direct or indirect) attribute names.
  - Attribute names should not be the same as other attributes of entity types in the same hierarchy

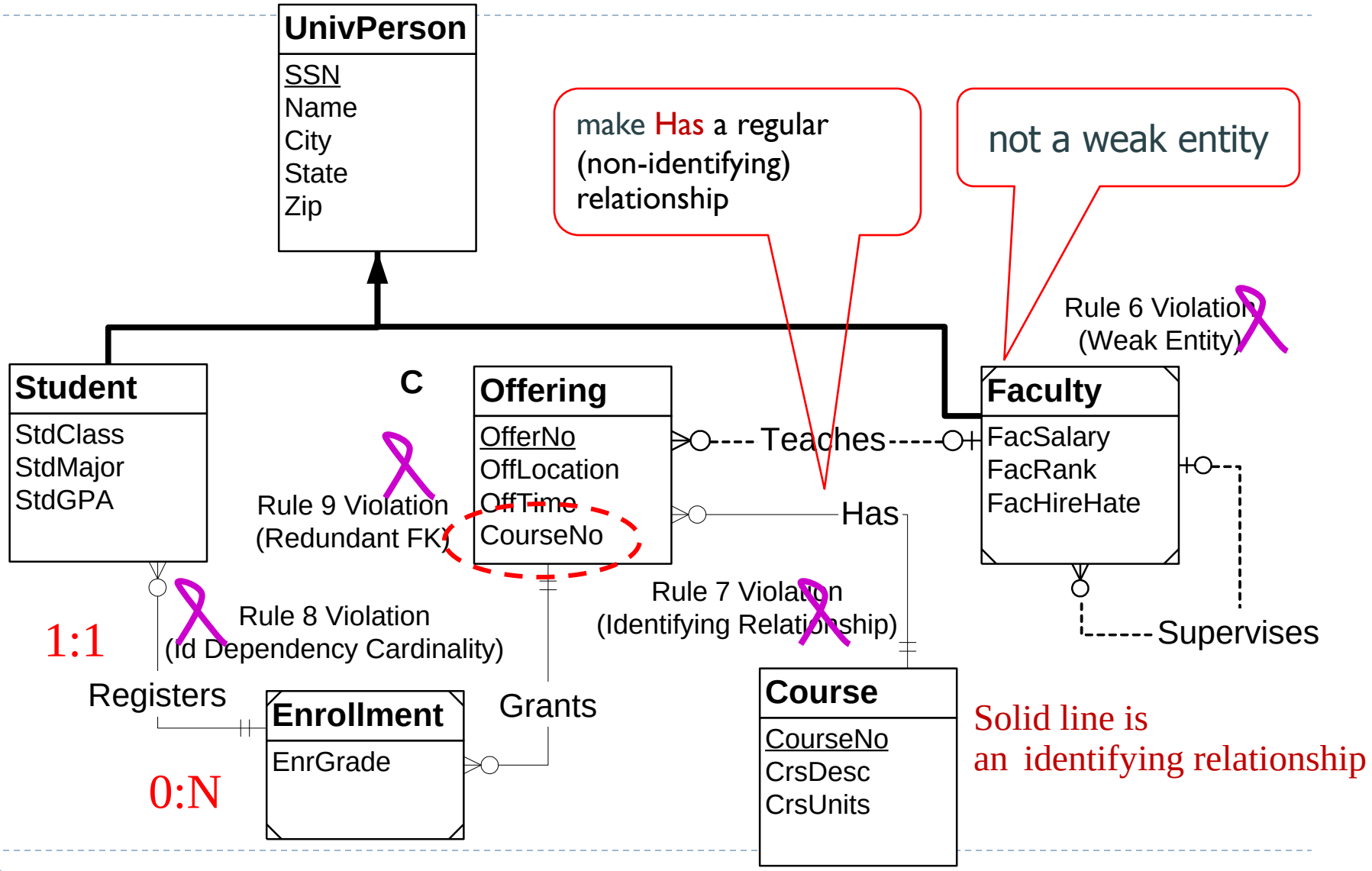
Connection Consistency Rules

- Relationship/Entity Connection Rule: relationships connect two entity types (not necessarily distinct) e.g. ,supervise
- Relationship/Relationship Connection Rule: relationships are not connected to other relationships
- Redundant Foreign Key Rule: foreign keys are not used.(Use FKs in the relational model, not in ERDs)

Identification Dependency Rules

- Weak entity rule: weak entities have at least one identifying relationship (borrow PK)
- Identifying relationship rule: at least one participating entity type must be weak for each identifying relationship
- Identification dependency cardinality rule: the minimum and maximum cardinality must equal 1 for a weak entity in all identifying relationships

Example of Diagram Errors



Explanation

Rule 8: Identification Dependency Cardinality

- The min/max cardinality of the Registers relationship should be (1,1) near Student
- Resolution: reverse the cardinalities on the Registers relationship

Rule 9: Redundant foreign key rule

- CourseNo in Offering is redundant with the Has relationship
- Resolution: remove the CourseNo attribute in Offering

Explanation

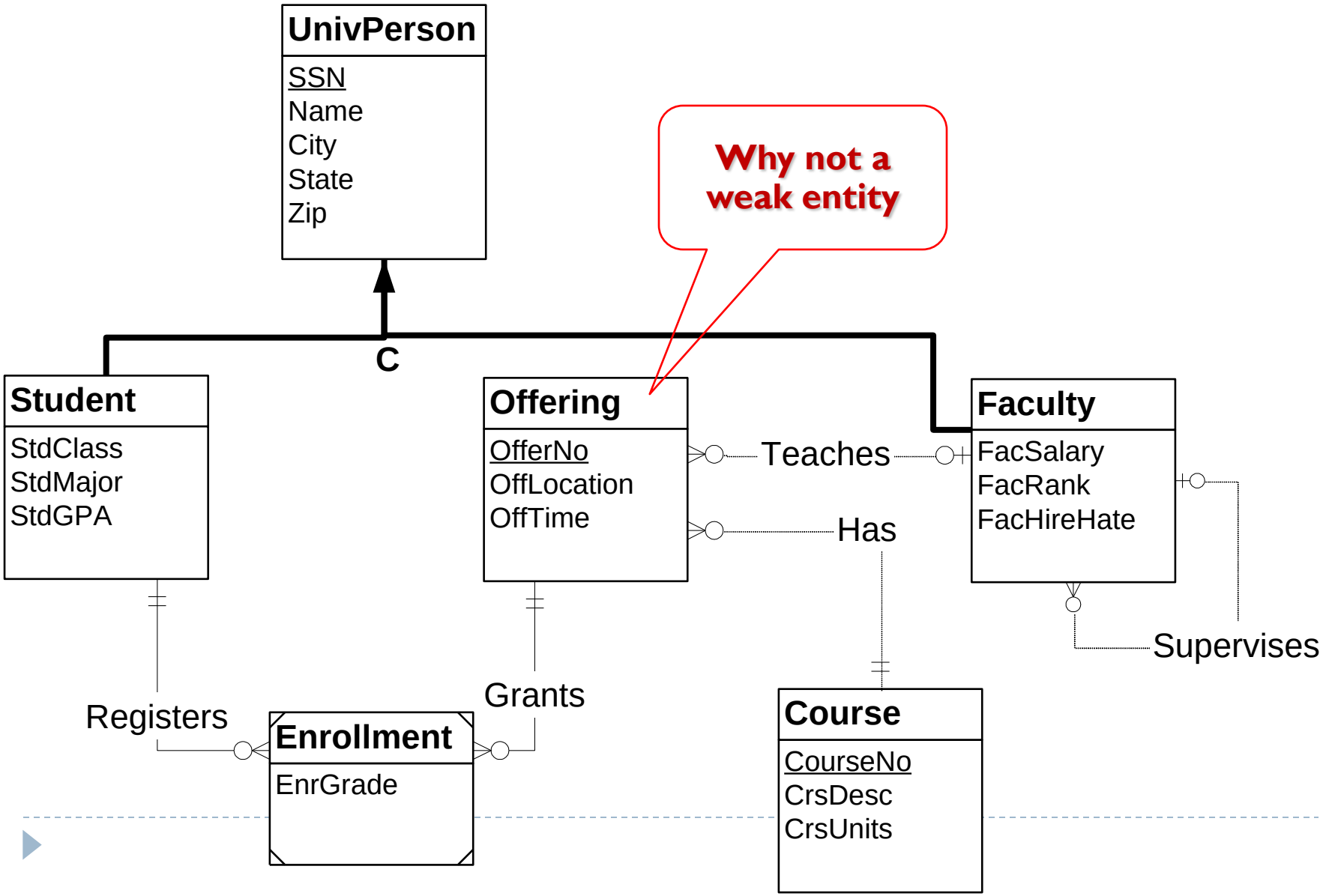
Rule 6: Weak entity rule violation

- Faculty is specified as a weak entity **but** it is not involved in any identifying relationships
- Resolution: remove weak entity symbols

Rule 7: Identifying relationship rule violation

- *Has* is an identifying relationship but neither Offering nor Course is a weak entity
- Resolution: make Has a regular (non-identifying) relationship
- : or making an entity type, course offering, weak

Corrected ERD



ER Assistant

- the ER Assistant supports the diagram rules
- http://er-assistant.software.informer.com/

Summary

- Data modeling is an important skill
- Crow's Foot ERD notation is widely used
- Use notation precisely
- Use the diagram rules to ensure structural consistency and completeness
- Understanding the ERD notation is a prerequisite to applying the notation on business problems