DATABASE SYSTEM PRINCIPLE

- ENTITY-RELATIONSHIP MODEL (PART A)

李旭东 **Li-Xudong**

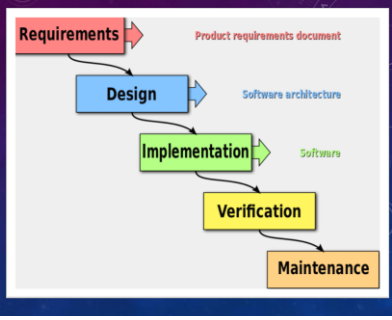
LEEXUDONG@NANKAI.EDU.CN

NANKAI UNIVERSITY

OBJECTIVES

- Design Process
- Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- E-R Notations

SOFTWARE DEVELOPMENT PROCESSES - WATERFALL MODEL



DESIGN PHASES

- (1)The initial phase of database design is to characterize fully the data needs of the prospective database users.
- (2)Next, the designer chooses a data model and, by applying the concepts of the chosen data model, translates these requirements into a conceptual schema of the database.
- (3)A fully developed conceptual schema also indicates the functional requirements of the enterprise.
  - In a "specification of functional requirements", users describe the kinds of operations (or transactions) that will be performed on the data.

DESIGN PHASES (CONT.,)

- (4)The process of moving from an abstract data model to the implementation of the database proceeds in two final design phases.
- (1) Logical Design – Deciding on the database schema.

  Database design requires that we find a "good" collection of relation schemas.
  - Business decision – What attributes should we record in the database?
  - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- (2) Physical Design – Deciding on the physical layout of the database

DESIGN ALTERNATIVES选择

- In designing a database schema, we must ensure that we avoid two major pitfalls:
  - 1.Redundancy
  - A bad design may repeat information
  - 2.Incompleteness
    - A bad design may make certain aspects of the enterprise difficult or impossible to model

DESIGN APPROACHES

- Entity Relationship Model (covered in this chapter)
  - Models an enterprise as a collection of *entities* and *relationships*
  - Represented diagrammatically by an *entity-relationship diagram*
- Normalization Theory (规范化理论)
  - Formalize what designs are bad, and test for them

©LXD

ENTITY-RELATIONSHIP MODEL

©LXD

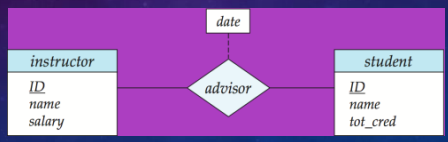
E-R MODEL - DATABASE MODELING

- The E-R data mode was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database.
- The E-R model is very useful in mapping the meanings and interactions of real-world enterprises onto a conceptual schema
  - Because of this usefulness, many database-design tools draw on concepts from the ER model.
- The E-R data model employs three basic concepts:
  - (1)entity sets, (2)relationship sets, (3)attributes.

©LXD

ER MODEL - DATABASE MODELING

- The E-R model also has an associated diagrammatic representation, the ER diagram, which can express the overall logical structure of a database graphically.



©LXD

BIBLIOGRAPHY : E-R MODEL

Dr. Peter Chen（陳品山）at Louisiana State University (LSU)

http://www.csc.lsu.edu/~chen/

- Chen P S. The Entity-Relationship Model—Toward a Unified View of Data[J]. Acm Transactions on Database Systems, 1976, 1(1):9-36.

- Prof. Peter Chen received his Ph.D. from Harvard University and has held regular and visiting faculty appointments at MIT, UCLA and Harvard.
- He is the originator of the **Entity-Relationship Model (ER Model)**, which serves as the foundation of many systems analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems.
- Now "Entity-Relationship Model (ER Model)," "Entity-Relationship Diagram (ER Diagram)," and "Peter Chen" have become commonly used terms in "online" dictionaries, books, articles, web pages, course syllabi, and commercial product brochures.

©LXD

ENTITY SETS

- An **entity** is an object that exists and is distinguishable from other objects
  - Described by a set of attributes
  - Example: specific person, company, event, plant
- An **entity set** is a set of entities of the same type that share the same properties.
  - Example: set of all persons, companies, trees, holidays

©LXD

ENTITY SETS

- An entity is represented by a set of attributes
  - i.e., descriptive properties possessed by all members of an entity set.
  - Example:

    instructor = (*ID, name, street, city, salary*)
    course= (*course_id, title, credits*)
- A subset of the attributes form a **primary key** of the entity set; i.e., uniquely identifying each member of the set

ENTITY SETS -- *INSTRUCTOR* AND *STUDENT*

instructor_ID	instructor_name
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

student-ID	student_name
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

RELATIONSHIP SETS

- A **relationship** is an association among several entities

  Example:
  44553 (Peltier) *advised by* 22222 (Einstein)
  student entity relationship set *instructor* entity
- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

  $\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$

  where $(e_1, e_2, \ldots, e_n)$ is a relationship
  - Example: (44553,22222) $\in$ *advisor*

RELATIONSHIP SET *ADVISOR*

76766 Crick	98988 Tanaka
45565 Katz	12345 Shankar
10101 Srinivasan	00128 Zhang
98345 Kim	76543 Brown
76543 Singh	76653 Aoi
22222 Einstein	23121 Chavez
	44553 Peltier

instructor *student*

RELATIONSHIP SETS (CONT.)

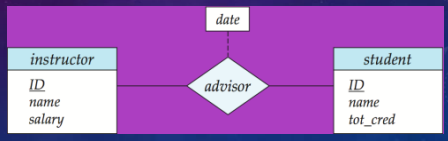
- An attribute can also be associated with a relationship set.
- Example:

76766 Crick		3 May 2008	98988 Tanaka
45565 Katz		10 June 2007	12345 Shankar
10101 Srinivasan		12 June 2006	00128 Zhang
98345 Kim		6 June 2009	76543 Brown
76543 Singh		30 June 2007	76653 Aoi
22222 Einstein		31 May 2007	23121 Chavez
		4 May 2006	44553 Peltier

instructor *student*

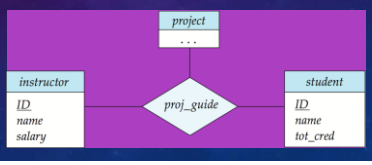
DEGREE度 OF A RELATIONSHIP SET 1/2

- (1)binary relationship二元关系
  - involve two entity sets (or degree two).
  - most relationship sets in a database system are binary.



DEGREE度 OF A RELATIONSHIP SET 2/2

- (2) nonbinary relationship多元关系
  - Relationships between more than two entity sets are rare
    - eg: *students* work on research *projects* under the guidance of an *instructor*.
      - relationship *proj_guide* is a ternary relationship between *instructor*, *student*, and *project*



ATTRIBUTES属性

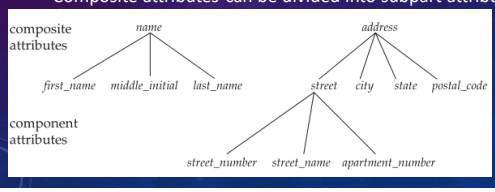
- Domain
  - For each attribute, there is a set of permitted values, called the domain域, or value set值集
- Example
  - the domain of attribute semester might be strings from the set {Fall, Winter, Spring, Summer}

ATTRIBUTE OF AN ENTITY SET

- an attribute of an entity set is a function that maps from the entity set into a domain.
  - Since an entity set may have several attributes, each entity can be described by a set of (attribute, data value) pairs, one pair for each attribute of the entity set.
- Example: instructor
  - {(ID , 76766), (name, Crick), (dept_name, Biology), (salary, 72000)}

ATTRIBUTE TYPES 1/3

- Simple简单 and composite复合 attributes
  - Composite attributes can be divided into subpart attributes



ATTRIBUTE TYPES 2/3

- Single-valued and multivalued attributes
- example
  - The student ID attribute for a specific student entity refers to only one student ID
  - An instructor may have zero, one, or several phone numbers, and different instructors may have different numbers of phones.
    - This type of attribute is said to be multivalued

ATTRIBUTE TYPES 3/3

- Derived attribute派生属性
  - The value for this type of attribute can be derived from the values of other related attributes or entities.
- Suppose that the instructor entity set has an attribute age that indicates the instructor's age
  - If the instructor entity set also has an attribute date of birth, we can calculate age from date of birth and the current date
  - Thus, age is a derived attribute

VALUE OF ATTRIBUTE

- null value
  - An attribute takes a null value when an entity does not have a value for it.
  - The null value may indicate “not applicable“不适用的
  - that is, that the value does not exist for the entity

CONSTRAINTS约束

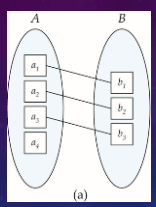
CONSTRAINTS约束

- An E-R enterprise schema may define certain constraints to which the contents of a database must conform
  - (1)mapping cardinalities映射基数
  - (2)participation constraints参与约束
  - (3)Key码

CONSTRAINTS: MAPPING CARDINALITY(映射基数)

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one, One to many, Many to one, Many to many

CONSTRAINTS: MAPPING CARDINALITY

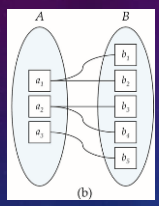


One to one

One to One : An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

CONSTRAINTS: MAPPING CARDINALITY

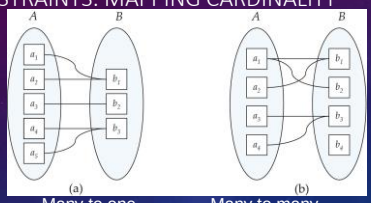


One to many

One-to-many: An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

CONSTRAINTS: MAPPING CARDINALITY



Many to one　　Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

CONSTRAINTS: PARTICIPATION CONSTRAINTS参与约束

- (1)Total: The participation of an entity set E in a relationship set R is said to be total全部 if every entity in E participates in at least one relationship in R.
- (2)Partial: If only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial部分.

CONSTRAINTS: KEY码

- We must have a way to specify how entities within a given entity set are distinguished.
- The values of the attribute values of an entity must be such that they can uniquely identify the entity.
- Superkey超码, candidate key候选码, primary key主码

CONSTRAINTS: KEY OF RELATIONSHIP SET 1/2

- Let R be a relationship set involving entity sets E1 , E2 ,..., En
- Let primary-key(Ei) denote the set of attributes that forms the primary key for entity set Ei
- Assume for now that the attribute names of all primary keys are unique.
- The composition of the primary key for a relationship set depends on the set of attributes associated with the relationship set R.
- (case 1)If the relationship set R has no attributes associated with it, then the set of attributes:

  primary-key(E1) ∪ primary-key(E2) ∪ ··· ∪ primary-key(En)

  describes an individual relationship in set R.

CONSTRAINTS: KEY OF RELATIONSHIP SET 2/2

- (case 2) If the relationship set R has attributes a1 ,a2 ,...,a m associated with it, then the set of attributes:

primary-key(E1) ∪ primary-key(E2) ∪ ··· ∪ primary-key(En) ∪ { a1 ,a2 ,..., am } describes an individual relationship in set R.

- In both of the above cases, the set of attributes

  primary-key(E1) ∪ primary-key(E2) ∪ ··· ∪ primary-key(En)

  forms a superkey for the relationship set.
- For nonbinary relationships, if no cardinality constraints are present then the super key formed is the only candidate key, and it is chosen as the primary key.

REMOVING REDUNDANT ATTRIBUTES IN ENTITY SETS

REMOVING REDUNDANT ATTRIBUTES IN ENTITY SETS

- Suppose we have entity sets:
  - *instructor*, with attributes: *ID*, *name*, *dept_name*, *salary*
  - *department*, with attributes: *dept_name*, *building*, *budget*
- We model the fact that each instructor has an associated department using ***a relationship set*** *inst_dept*
- The attribute *dept_name* appears in both entity sets. Since it is the primary key for the entity set *department*, it replicates information present in the relationship and is therefore redundant in the entity set *instructor* and needs to be removed.

REMOVING REDUNDANT ATTRIBUTES IN ENTITY SETS

A good entity-relationship design does not contain redundant attributes:

- **classroom**: with attributes (*building*, *room_number*, *capacity*).
- **department**: with attributes (*dept_name*, *building*, *budget*).
- **course**: with attributes (*course_id*, *title*, *credits*).
- **instructor**: with attributes (*ID*, *name*, *salary*).
- **section:** with attributes (*course_id*, *sec_id*, *semester*, *year*).
- **student**: with attributes (*ID*, *name*, *tot_cred*).
- **time_slot**: with attributes (*time_slot_id*, {(*day*, *start_time*, *end_time*) }).

REMOVING REDUNDANT ATTRIBUTES IN ENTITY SETS

- **inst_dept**: relating instructors with departments.
- **stud_dept**: relating students with departments.
- **teaches**: relating instructors with sections.
- **takes**: relating students with sections, with a descriptive attribute *grade*.
- **course_dept**: relating courses with departments.
- **sec_course**: relating sections with courses.
- **sec_class**: relating sections with classrooms.
- **sec_time_slot**: relating sections with time slots.
- **advisor**: relating students with instructors.
- **prereq**: relating courses with prerequisite courses.

ENTITY-RELATIONSHIP DIAGRAMS

ENTITY SETS

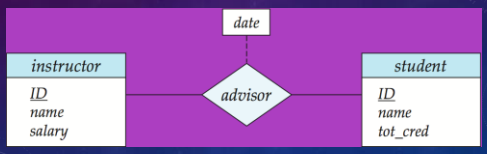
Entities can be represented graphically as follows:

- Rectangles represent entity sets.
- Attributes listed inside entity rectangle
- Underline indicates primary key attributes

RELATIONSHIP SETS

Diamonds represent relationship sets.

RELATIONSHIP SETS WITH ATTRIBUTES



CARDINALITY CONSTRAINTS映射基数

- We express cardinality constraints by drawing the following line between the relationship set and the entity set
  - either a directed line (→), signifying "one,"
  - or an undirected line (—), signifying "many,"

CARDINALITY CONSTRAINTS映射基数

- One-to-one relationship between an *instructor* and a *student* :
  - A student is associated with at most one *instructor* via the relationship *advisor*
  - A *student* is associated with at most one *department* via *stud_dept*

CARDINALITY CONSTRAINTS映射基数

- One-to-many relationship between an *instructor* and a *student*
  - an instructor is associated with several (including 0) students via *advisor*
  - a student is associated with at most one instructor via advisor,

CARDINALITY CONSTRAINTS映射基数: MANY-TO-MANY RELATIONSHIP

- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*

CARDINALITY CONSTRAINTS映射基数: TOTAL AND PARTIAL PARTICIPATION

Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

participation of *student* in *advisor* relation is total

- every *student* must have an associated instructor

Partial participation: some entities may not participate in any relationship in the relationship set

Example: participation of *instructor* in *advisor* is partial

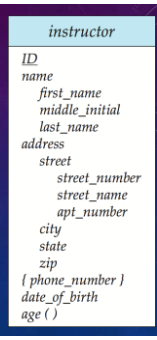
NOTATION FOR EXPRESSING MORE COMPLEX CONSTRAINTS

A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality

- A minimum value of 1 indicates total participation.
- A maximum value of 1 indicates that the entity participates in at most one relationship
- A maximum value of * indicates no limit.

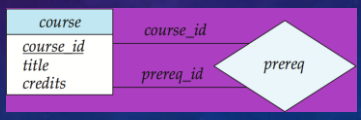
NOTATION FOR EXPRESSING MORE COMPLEX CONSTRAINTS

- composite attribute name
- a multivalued attribute: phone number
- a derived attribute age



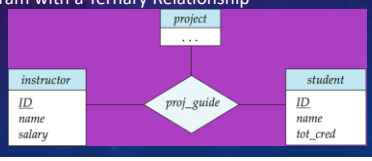
ROLES角色

- Entity sets of a relationship need not be distinct
  - Each occurrence of an entity set plays a "role" in the relationship
- The labels "*course_id*" and "*prereq_id*" are called **roles**.



NON-BINARY RELATIONSHIP SETS

- Most relationship sets are binary
- There are occasions when it is more convenient to represent relationships as non-binary.
- E-R Diagram with a Ternary Relationship



CARDINALITY CONSTRAINTS ON TERNARY RELATIONSHIP三元联系

- We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint
  - For exampe, an arrow from *proj_guide* to *instructor* indicates each student has at most one guide for a project
- If there is more than one arrow, there are two ways of defining the meaning.
  - Example (cont.,)

CARDINALITY CONSTRAINTS ON TERNARY RELATIONSHIP

- If there is more than one arrow, there are two ways of defining the meaning.
  - For example, a ternary relationship *R* between *A*, *B* and *C* with arrows to *B* and *C* could mean
    1. Each *A* entity is associated with a unique entity from *B* and *C* or
    2. Each pair of entities from (*A, B*) is associated with a unique *C* entity, and each pair (*A, C*) is associated with a unique *B*
  - Each alternative has been used in different formalisms
  - To avoid confusion we outlaw more than one arrow

WEAK ENTITY SETS弱实体集

- Consider a *section* entity, which is uniquely identified by a *course_id*, *semester, year*, and *sec_id*.
- Clearly, section entities are related to course entities. Suppose we create a relationship set *sec_course* between entity sets *section* and *course*.
- Note that the information in *sec_course* is redundant, since *section* already has an attribute *course_id*, which identifies the course with which the section is related.
- **One option** to deal with this redundancy is to get rid of the relationship *sec_course*; however, by doing so the relationship between *section* and *course* becomes implicit in an attribute, which is not desirable.

©LXD

WEAK ENTITY SETS (CONT.)

- An alternative way to deal with this redundancy is to not store the attribute *course_id* in the *section* entity and to only store the remaining attributes section_id, *year*, and *semester.*
- However, the entity set *section* then does not have enough attributes to identify a particular *section* entity uniquely; although each *section* entity is distinct, sections for different courses may share the same *section_id*, *year*, and *semester*.
- To deal with this problem, we treat the relationship ***sec_course*** as a special relationship that provides extra information, in this case, the *course_id*, required to identify *section* entities uniquely.

©LXD

WEAK ENTITY SETS (CONT.)

- The notion of **weak entity set** formalizes the above intuition.
- A weak entity set is one whose existence is dependent on another entity, called its **identifying entity**;
- Instead of associating a primary key with a weak entity, we use the identifying entity, along with extra attributes called **discriminator分辨符** to uniquely identify a weak entity.
- An entity set that is not a weak entity set is termed a **strong entity set强实体集**

©LXD

WEAK ENTITY SETS (CONT.)

- Every weak entity must be associated with an identifying entity标识实体
- that is, the weak entity set is said to be **existence dependent存在依赖** on the identifying entity set.
- The identifying entity set标识实体集(属主实体集) is said to **own拥有** the weak entity set that it identifies. The relationship associating the weak entity set with the identifying entity set is called the **identifying relationship标识性联系**.
- Note that the relational schema we eventually create from the entity set *section* does have the attribute *course_id*, for reasons that will become clear later, even though we have dropped the attribute *course_id* from the entity set *section*.

©LXD

WEAK ENTITY SETS (CONT.)

- In E-R diagrams, a weak entity set is depicted via a double rectangle.
- We underline the discriminator分辨符 of a weak entity set with a dashed line.
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.
- The primary key of a weak entity set is formed by the primary key of the identifying entity set, plus the weak entity set's discriminator

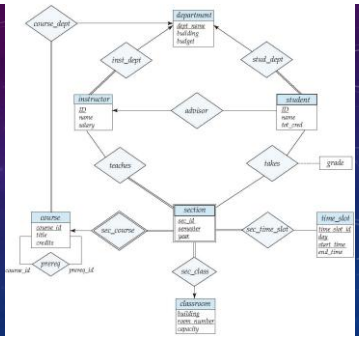
©LXD

WEAK ENTITY SETS (CONT.)

- Primary key for *section* – (*course_id, sec_id, semester, year*)

©LXD

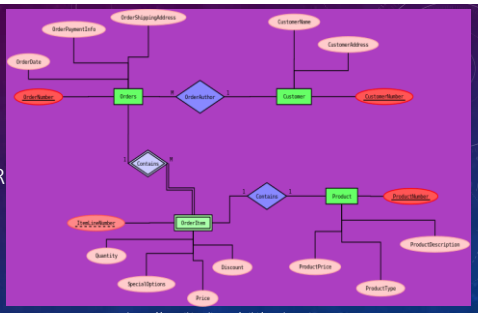
E-R DIAGRAM FOR A UNIVERSITY ENTERPRISE



ENTITY-RELATIONSHIP DIAGRAMS

- E-R diagram can express the overall logical structure of a database graphically
- Basic structure:
  - **Rectangles divided into two parts** represent entity sets. The first part, which in this textbook is shaded blue, contains the name of the entity set. The second part contains the names of all the attributes of the entity set.
  - **Diamonds** represent relationship sets.
  - **Undivided rectangles** represent the attributes of a relationship set. Attributes that are part of the primary key are underlined.
  - **Lines** link entity sets to relationship sets.
  - **Dashed lines** link attributes of a relationship set to the relationship set.
  - **Double lines** indicate total participation of an entity in a relationship set.
  - **Double diamonds** represent identifying relationship sets linked to weak entity sets (we discuss identifying relationship sets and weak entity sets later,

E-R DIAGRAM FOR A CUSTOMER ORDERS

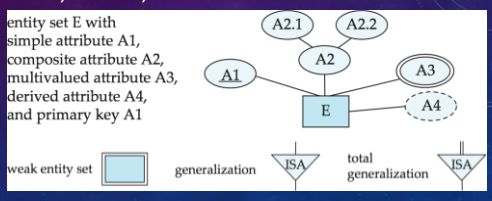


From https://en.wikipedia.org/wiki/Weak_entity

ALTERNATIVE ER NOTATIONS

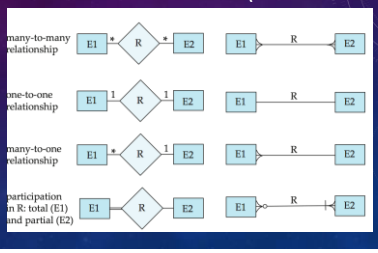
ALTERNATIVE ER NOTATIONS

- Chen, IDE1FX, …



ALTERNATIVE ER NOTATIONS

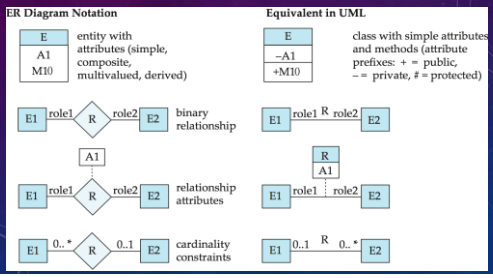
Chen **IDE1FX (Crows feet notation)**



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.

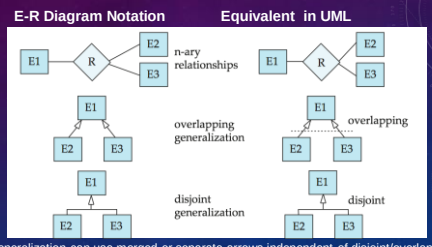
ER VS. UML CLASS DIAGRAMS



UML CLASS DIAGRAMS

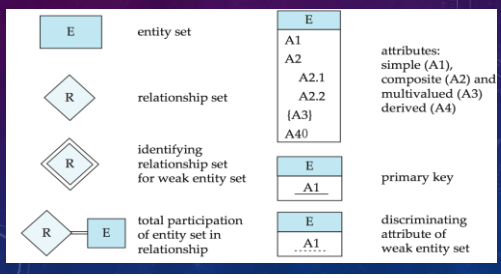
- Binary relationship sets are represented in UML by just drawing a line connecting the entity sets. The relationship set name is written adjacent to the line.
- The role played by an entity set in a relationship set may also be specified by writing the role name on the line, adjacent to the entity set.
- 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.

E-R VS. UML CLASS DIAGRAMS

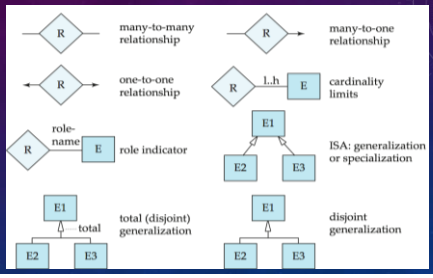


*Generalization can use merged or separate arrows independent of disjoint/overlapping

SUMMARY OF SYMBOLS USED IN E-R NOTATION



SUMMARY OF SYMBOLS USED IN E-R NOTATION



SUMMARY

- Design Process
- Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- E-R Notations

Q&A?

THANKS!

leexudong@nankai.edu.cn