数据库设计与E-R模型
Database Design & E-R Model

李文根/Wengen Li
Email: lwengen@tongji.edu.cn

先进数据与机器智能系统实验室（ADMIS Lab）
https://admis-tongji.github.io

同济大学 计算机科学与技术学院

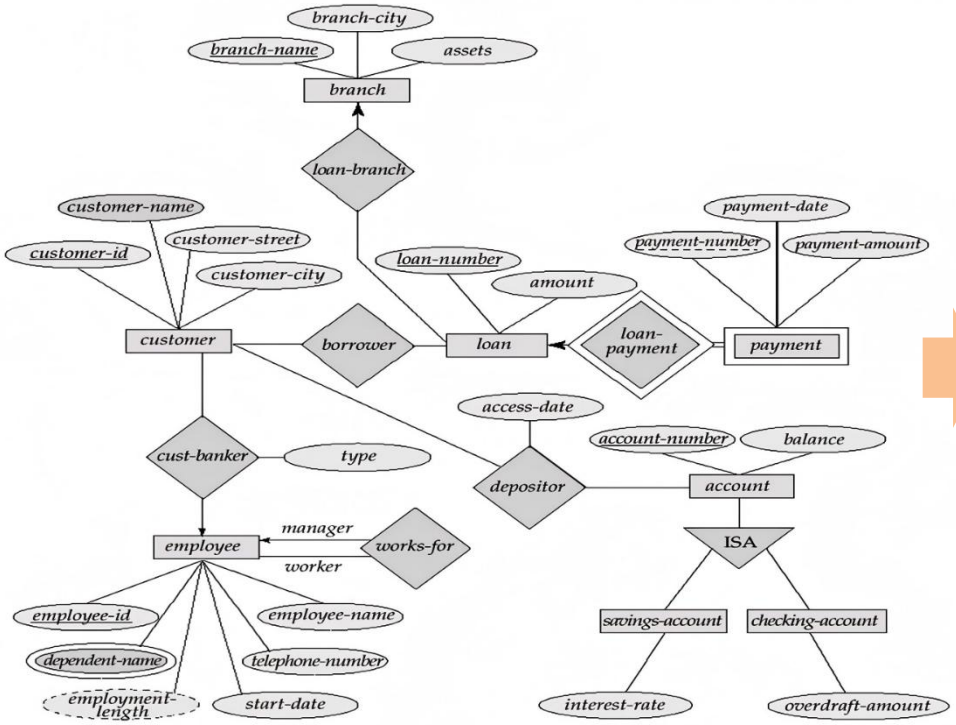
2026年04月

课程概要

同济大学 TONGJI UNIVERSITY

- **Part 0: Overview**
  - Ch1: Introduction
- **Part 1 Relational Languages**
  - Ch2: Relational model
  - Ch3: Introduction to SQL
  - Ch4: Intermediate SQL
  - Ch5: Advanced SQL
- **Part 2 Database Design**
  - **Ch6: Database design via E-R model**
  - Ch7: Relational database design
- **Part 3 Application Design & Development**
  - Ch8: Complex data types
  - Ch9: Application development
- **Part 4 Big Data Analytics**
  - Ch10: Big data
  - Ch11: Data analytics
- **Part 5 Storage Management & Indexing**
  - Ch12: Physical storage systems
  - Ch13: Data storage structures
  - Ch14: Indexing
- **Part 6 Query Processing & Optimization**
  - Ch15: Query processing
  - Ch16: Query optimization
- **Part 7 Transaction Management**
  - Ch17: Transactions
  - Ch18: Concurrency control
  - Ch19: Recovery system
- **Part 8 Parallel & Distributed Database**
  - Ch20: Database system architecture
  - Ch21-23: Parallel & distributed storage, query processing & transaction processing
- **Advanced topics**
  - DB Platform: **OceanBase**, MongoDB, Neo4J
  - RAG, Multimodal retrieval, …

E-R图和关系模式(Banking DB)



E-R图

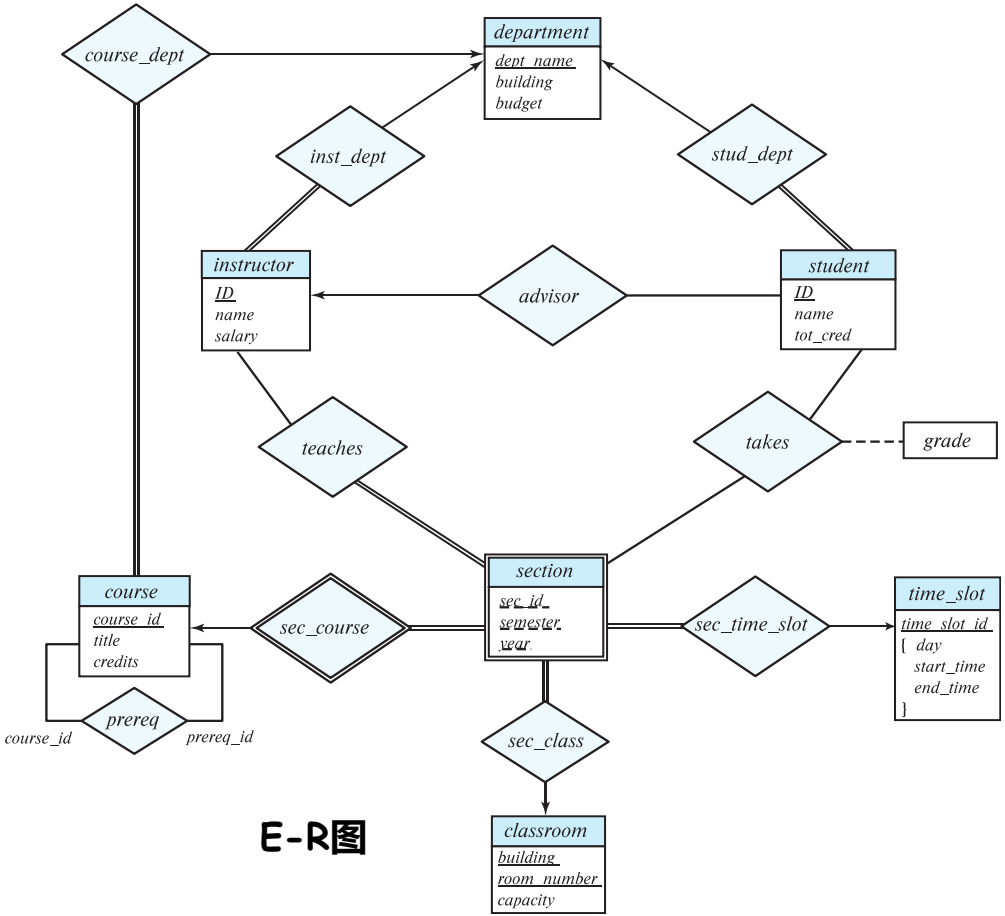
- branch (<u>branch_name</u>, branch_city, assets)
- customer (<u>customer_id</u>, customer_name, customer_street, customer_city)
- loan (<u>loan_number</u>, amount)
- account (<u>account_number</u>, balance)
- employee (<u>employee_id</u>, employee_name, telephone_number, start_date)

- dependent_name (<u>employee_id, dname</u>) (derived from a multivalued attribute)

- account_branch (<u>account_number</u>, branch_name)
- loan_branch (<u>loan_number</u>, branch_name)
- borrower (<u>customer_id, loan_number</u>)
- depositor (<u>customer_id, account_number</u>, access_date)
- cust_banker (<u>customer_id</u>, employee_id, type)
- works_for (<u>worker_employee_id</u>, manager_employee_id)
- Payment (<u>loan_number, payment_number</u>, payment_date, payment_amount)
- savings_account (<u>account_number</u>, interest_rate)
- checking_account (<u>account_number</u>, overdraft_amount)

关系模式

E-R图和关系模式(University DB)



E-R图

classroom(*building*, *room_number*, *capacity*)
department(*dept_name*, *building*, *budget*)
course(*course_id*, *title*, *credits*)
instructor(*ID*, *name*, *salary*)
student(*ID*, *name*, *tot_cred*)
teaches (*ID*, *course_id*, *sec_id*, *semester*, *year*)
takes (*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
prereq (*course_id*, *prereq_id*)
advisor (*s_ID*, *i_ID*)
sec_course (*course_id*, *sec_id*, *semester*, *year*)
sec_time_slot (*course_id*, *sec_id*, *semester*, *year*, *time_slot_id*)
sec_class (*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*)
inst_dept (*ID*, *dept_name*)
stud_dept (*ID*, *dept_name*)
course_dept (*course_id*, *dept_name*)

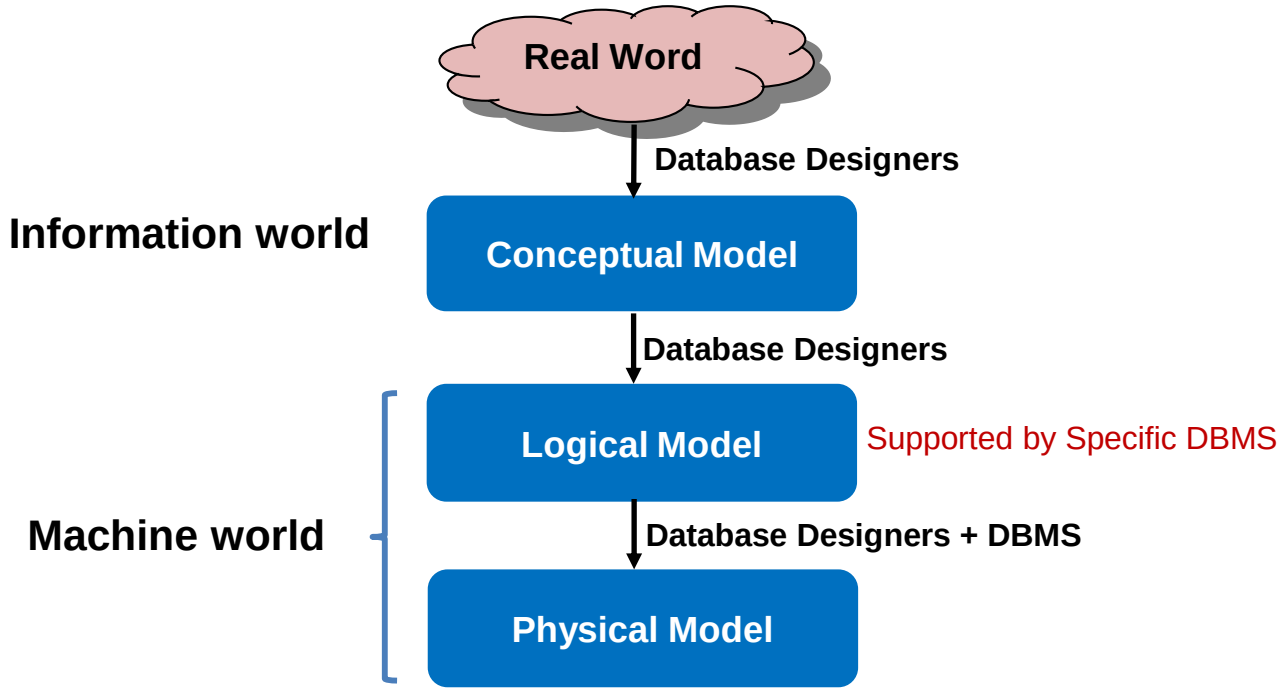
关系模式

目录

开发数据库应用的主要任务

- 关系（表）
- 关系的属性
- 约束

数据抽象(Data Abstraction)



数据库设计(Database Design)

- **Conceptual design(概念设计)**
  - Map a real world organization to a conceptual model
- **Logical design(逻辑设计)**
  - Transform the conceptual model to a logical model
- **Physical design(物理设计)**
  - Instantiate the logical model to physical organization and storage

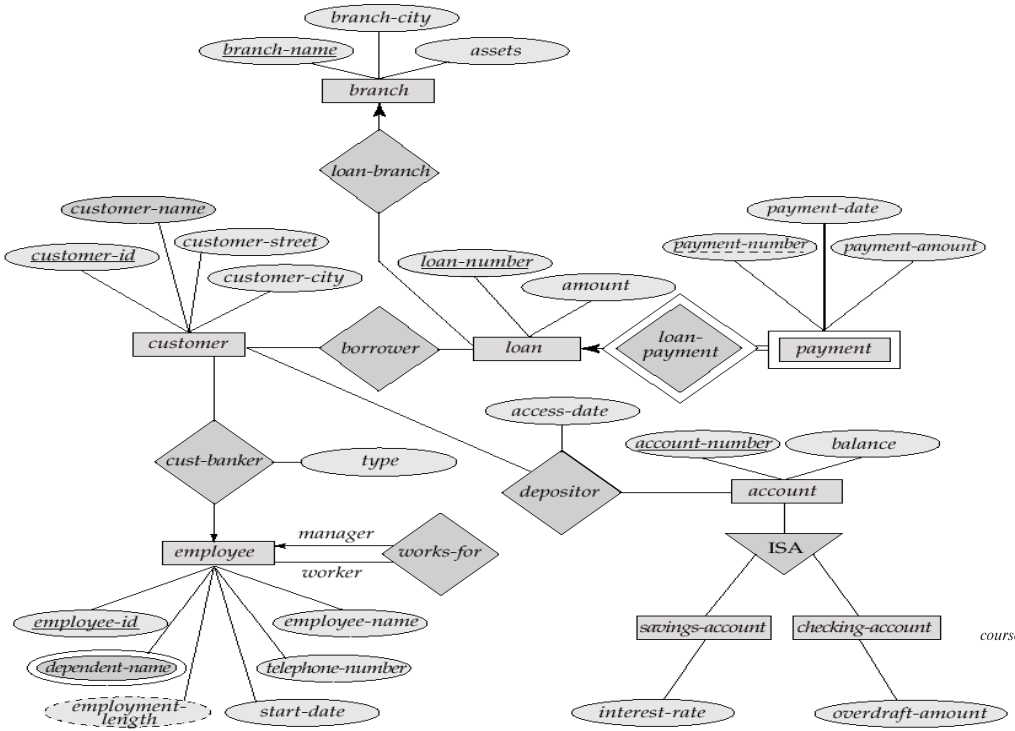
数据库设计(续)

- **Step 1**: Understand the real-world domain being modeled
- **Step 2**: Specify it using a database design model
  - Design models are especially convenient for schema design, but are not necessarily implemented by DBMS
    - Entity-Relationship (E-R) model
    - Object Definition Language (ODL)
- **Step 3**: Translate specification to the data model of DBMS
  - Relational, XML, object-oriented, etc.
- **Step 4**: Create the DBMS schema

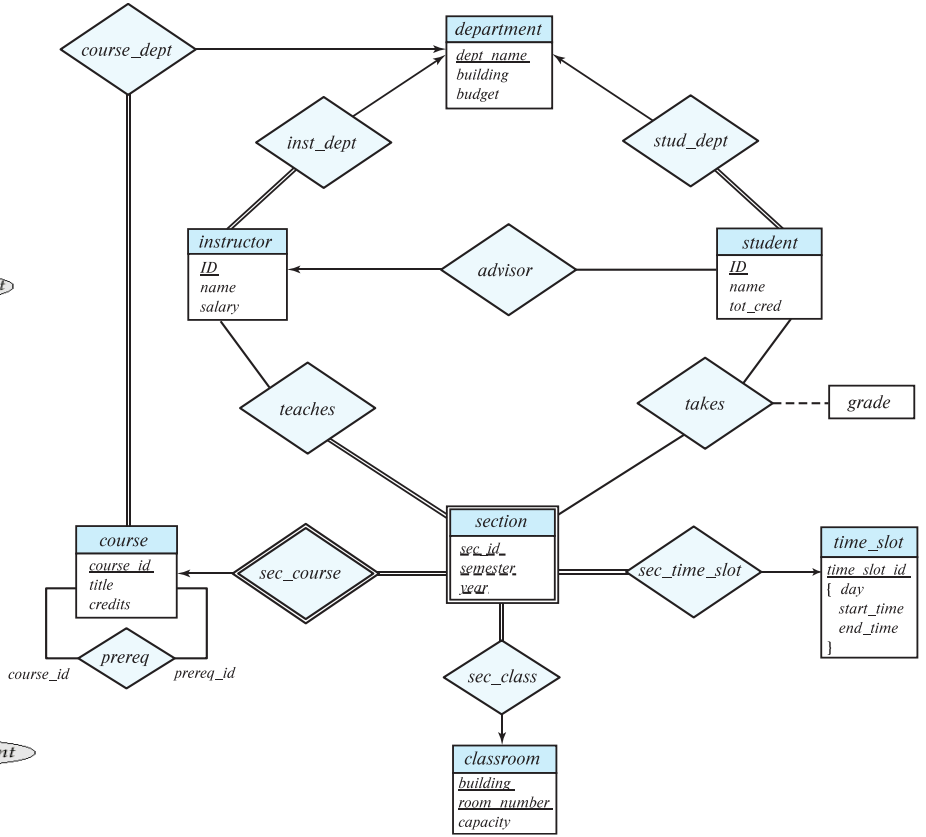
目录

- 设计过程概览
- E-R模型
- 约束
- E-R图
- E-R图转换为关系模式

E-R Diagram



Banking DB



University DB

数据库概念设计(Database Conceptual Design)

- **Conceptual design (E-R Model is used at this stage)**
  - **What** are the entities and relationships?
  - **What** information about these entities and relationships should be stored in the database?
  - **What** are the integrity constraints or business rules that should hold?
  - A database 'schema' in the E-R Model can be represented pictorially using **E-R diagram**
  - An E-R diagram can be then mapped into a relational schema

E-R模型

- A “watered-down” object-oriented design model
- Primarily a design model, not implemented by any major DBMS
- **Three concepts**
  - 实体集（Entity set）
  - 属性（Attribute）
  - 联系集（Relationship set）

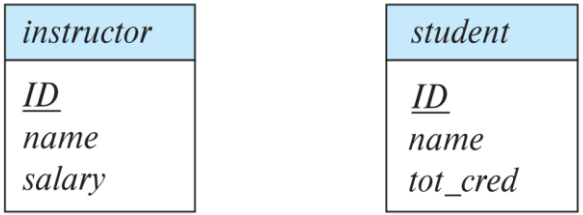
Peter Pin-Shan Chen（陈品山）

- Dr. Peter P. Chen is the originator of the Entity-Relationship Model (E-R Model), and the founder of ER international conference
- The E-R model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems

Peter Chen, ***The Entity-Relationship Model--Toward a Unified View of Data***
ACM Transactions on Database Systems (TODS), Vol. 1, No. 1, March 1976, Pages 9 - 36

实体集（Entity Sets）

- **A database can be modeled as**
  - a collection of entities
  - relationship among entities
- An entity is an object that exists and is distinguishable from other objects
  - E.g., specific person, company, event, and university
- Entities have attributes
  - E.g., people have names and addresses
- An entity set is a set of entities of the same type that share the same properties
  - E.g., the set of all persons, companies, trees, holidays



实体集(续)

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

instructor

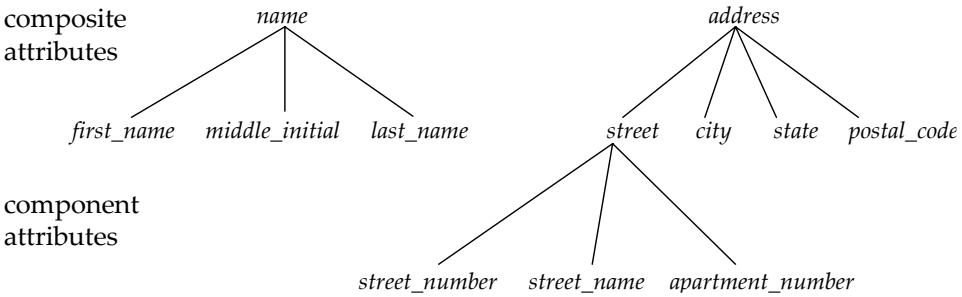
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

属性(Attributes)

- An entity contains **a set of attributes**, and these attributes are possessed by all members of an entity set
  - ***instructor***: *ID, name, salary*
  - ***student***: *ID, name, tot_cred*

- **Attribute types**
  - Simple and composite attributes (复合属性)
  - Single-valued and multi-valued attributes (多值属性)
  - Derived attributes (派生属性)



instructor
ID
name
first_name
middle_initial
last_name
address
street
street_number
street_name
apt_number
city
state
zip
{ phone_number }
date_of_birth
age ( )

联系集(Relationship Sets)

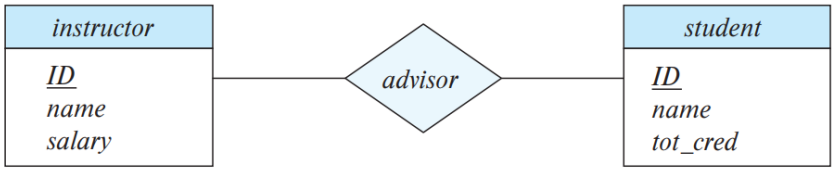
- A relationship is an association among several entities
  - E.g.,

44553 (Peltier) *advisor* *22222 (Einstein)*
student entity *relationship set* *instructor entity*

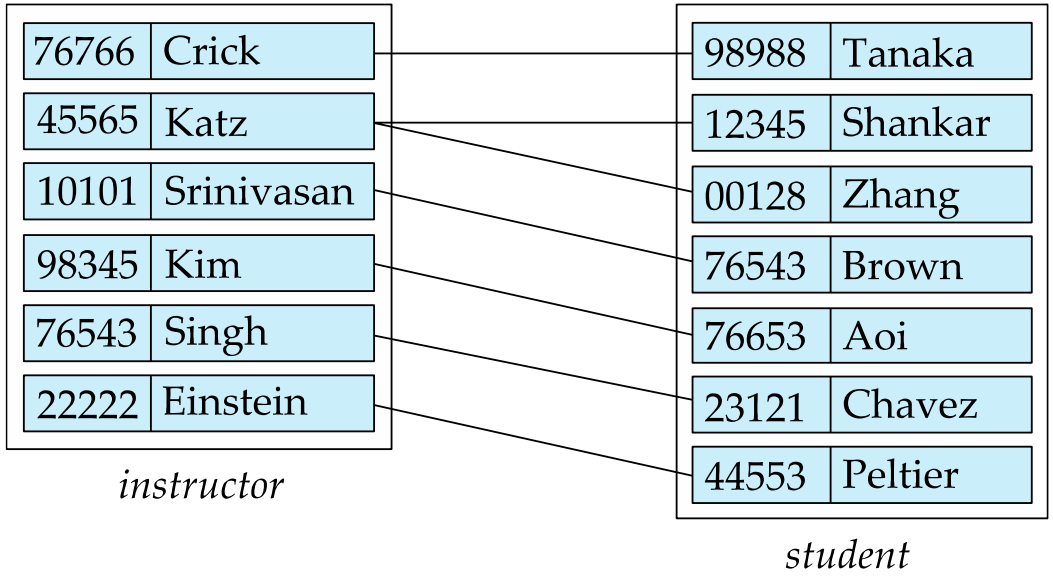
- A relationship set is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

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

where $(e_1, e_2, \ldots, e_n)$ is a relationship, e.g., (44553, 22222) ∈ advisor

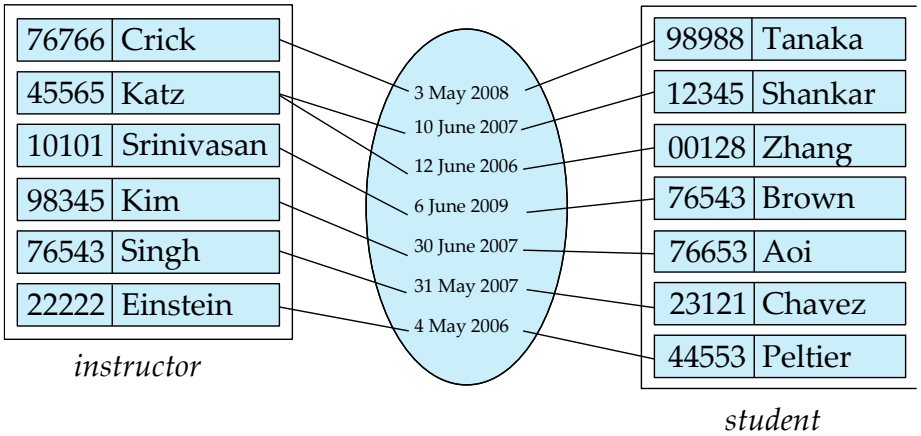


联系集advisor


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

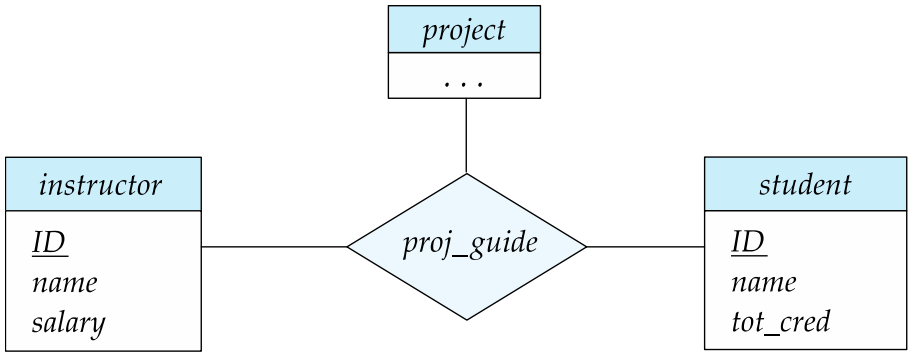
联系集（续）

- A relationship set may also contain some attributes
- For instance, the advisor relationship between the entity sets **instructor** and **student** may include the attribute **date** which records when the student started being associated with the advisor



联系集的度/阶

- **The number of entity sets that participate in a relationship set**
  - Relationship sets that involve two entity sets are binary (二元的)
  - Relationship sets may involve more than two entity sets, which is rare
    - E.g., E-R diagram with a ternary relationship (三元联系)

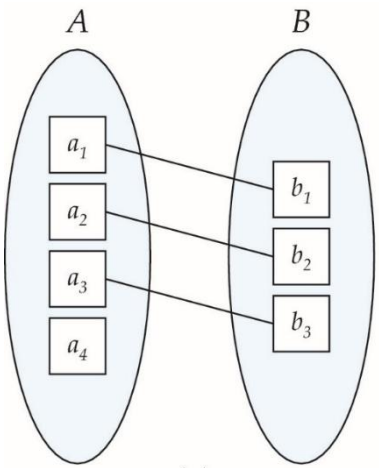


目录

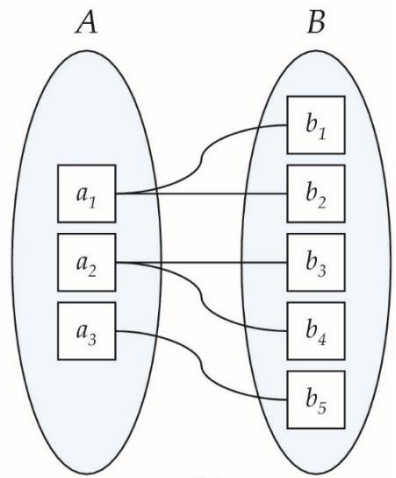
- 设计过程概览
- E-R模型
- 约束
- E-R图
- E-R图转换为关系模式

映射基数(Mapping Cardinalities)

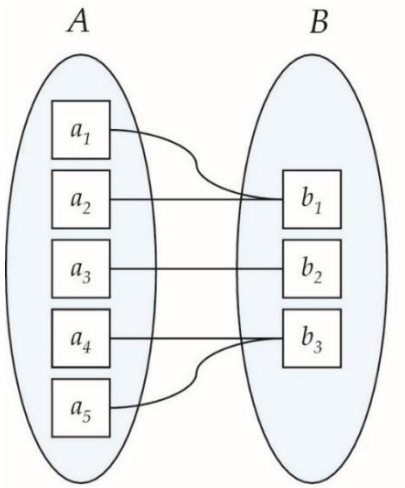
- 一个实体通过关系集可以关联的实体数量
- 给定两个实体集A和B：



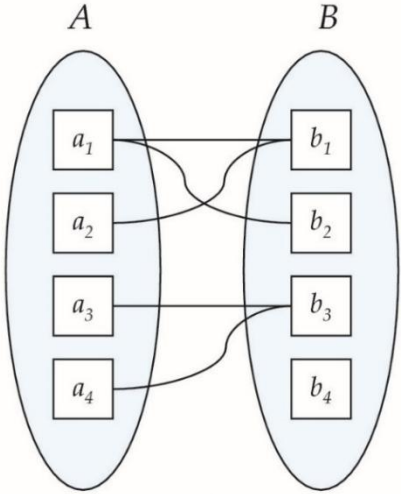
一对一



一对多



多对一



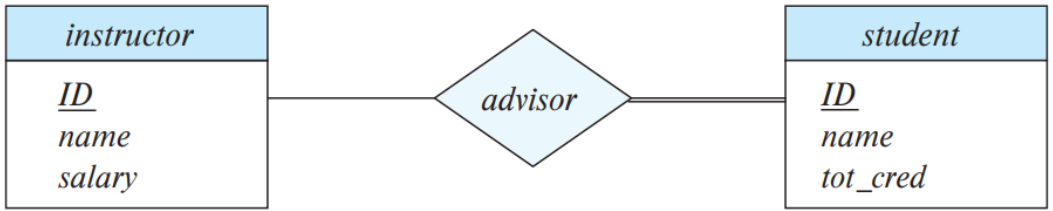
多对多

映射基数(续)

- For a binary relationship set, the mapping cardinality is one of the following types
  - **One to one (1对1):** 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
  - **One to many (1对多):** 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
  - **Many to one (多对1):** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A
  - **Many to many (多对多):** An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A

参与约束(Participation Constraints)

- **Total participation**
  - Every entity in the entity set participates in at least one relationship in the relationship set
  - E.g., 每个student实体通过advisor联系同至少一名instructor关联，student在联系集advisor中是全部参与
- **Partial participation**
  - Some entities may not participate in any relationship in the relationship set
  - E.g., 有的instructor可能不指导学生，所以instructor在联系集advisor中是部分参与



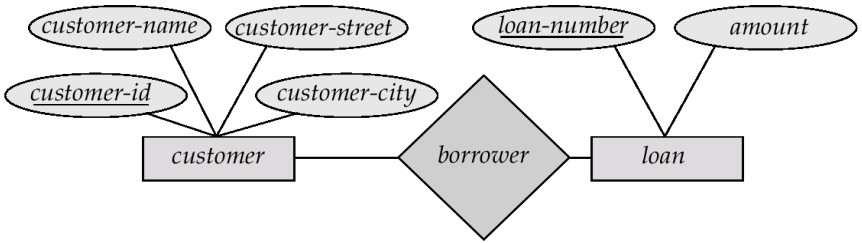
目录

- 设计过程概览
- E-R模型
- 约束
- E-R图
- E-R图转换为关系模式

E-R图

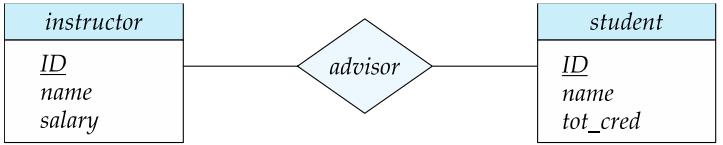
风格A

- Rectangles represent entity sets
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
  - Double ellipses represent multi-valued attributes
  - Dashed ellipses denote derived attributes
- Underline indicates primary key attributes



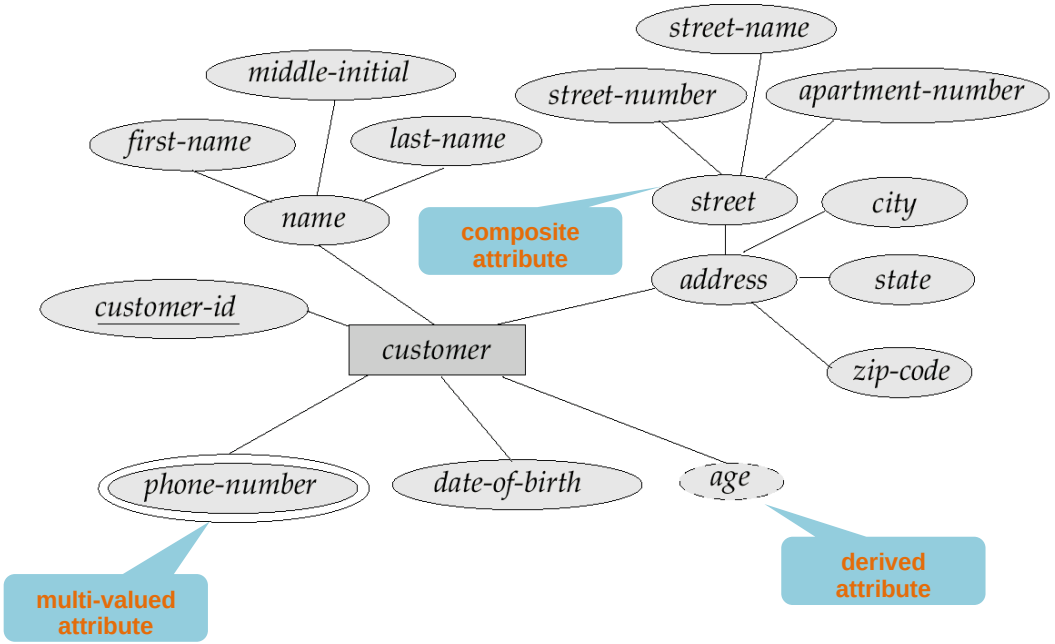
风格B

- Entity sets can be represented graphically as follows:
  - Rectangles represent entity sets.
  - Attributes listed inside entity rectangle
  - Underline indicates primary key attributes
- Diamonds represent relationship sets



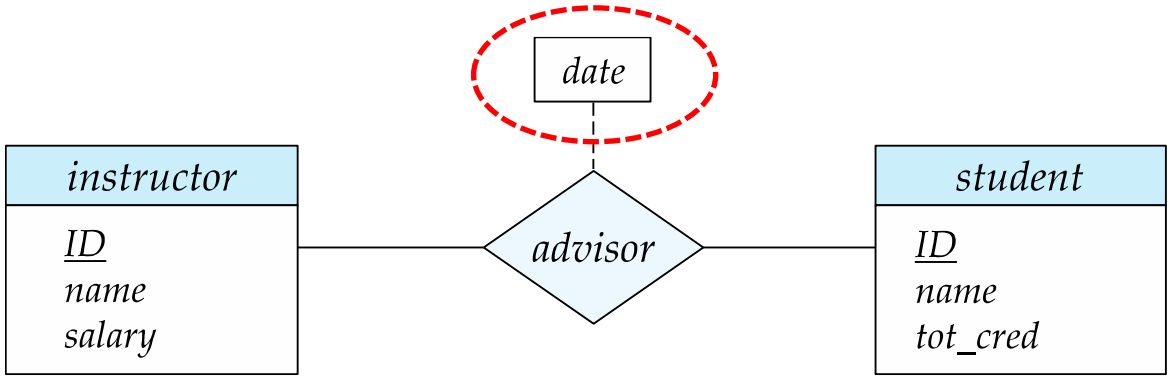
E-R图(续)

- E-R diagram with composite, multivalued, and derived attributes



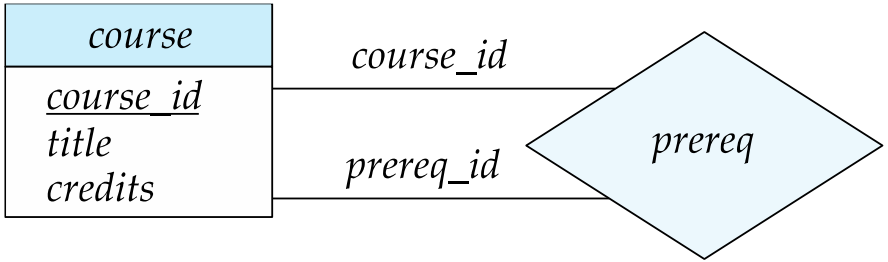
instructor
ID
name
first_name
middle_initial
last_name
address
street
street_number
street_name
apt_number
city
state
zip
{ phone_number }
date_of_birth
age ( )

联系集的属性


date
instructor
ID
name
salary
advisor
student
ID
name
tot_cred

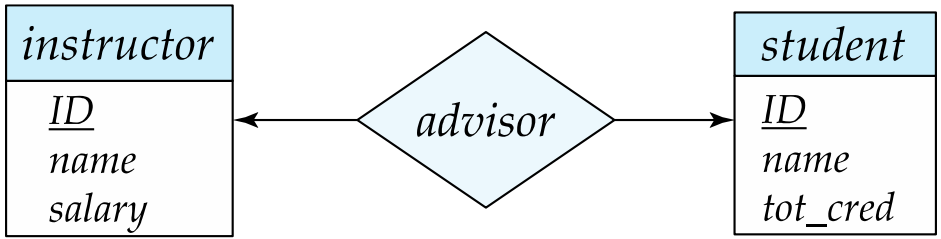
角色(Roles)

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



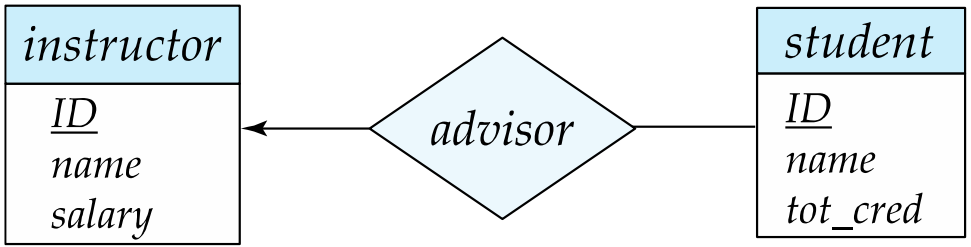
映射基数约束（Cardinality Constraints）

- We express cardinality constraints by drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.
- **一对一联系**
  - A student is associated with at most one instructor via the relationship advisor, and vice versa



一对多联系

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

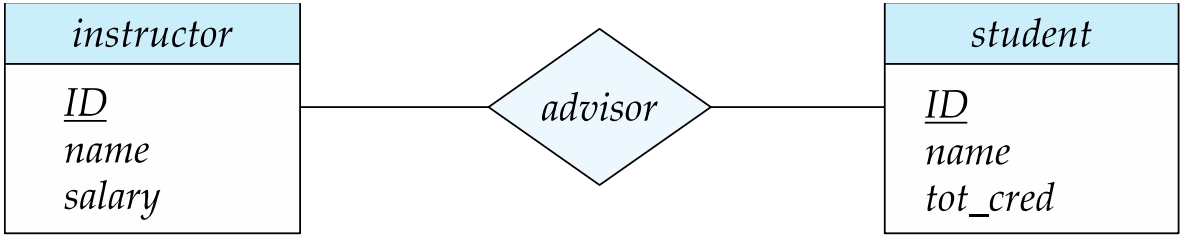


多对一联系

- **Many-to-one relationship**
  - an instructor is associated with at most one student via advisor
  - a student is associated with several (including 0) instructors via advisor

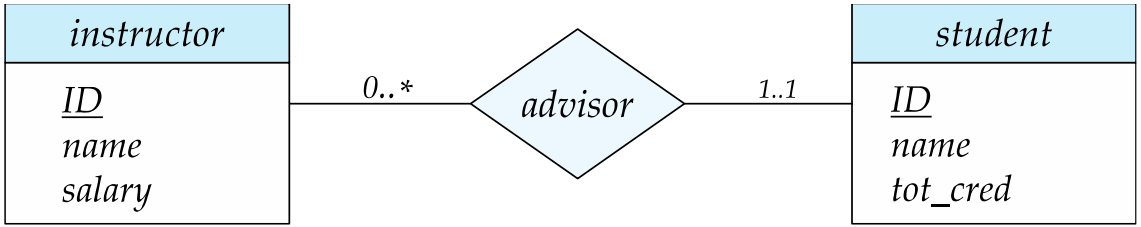
多对多联系

- **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



映射基数约束（续）

- 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
- **Example**
  - Each instructor can advise 0 or more students, and each student must have 1 advisor and cannot have multiple advisors



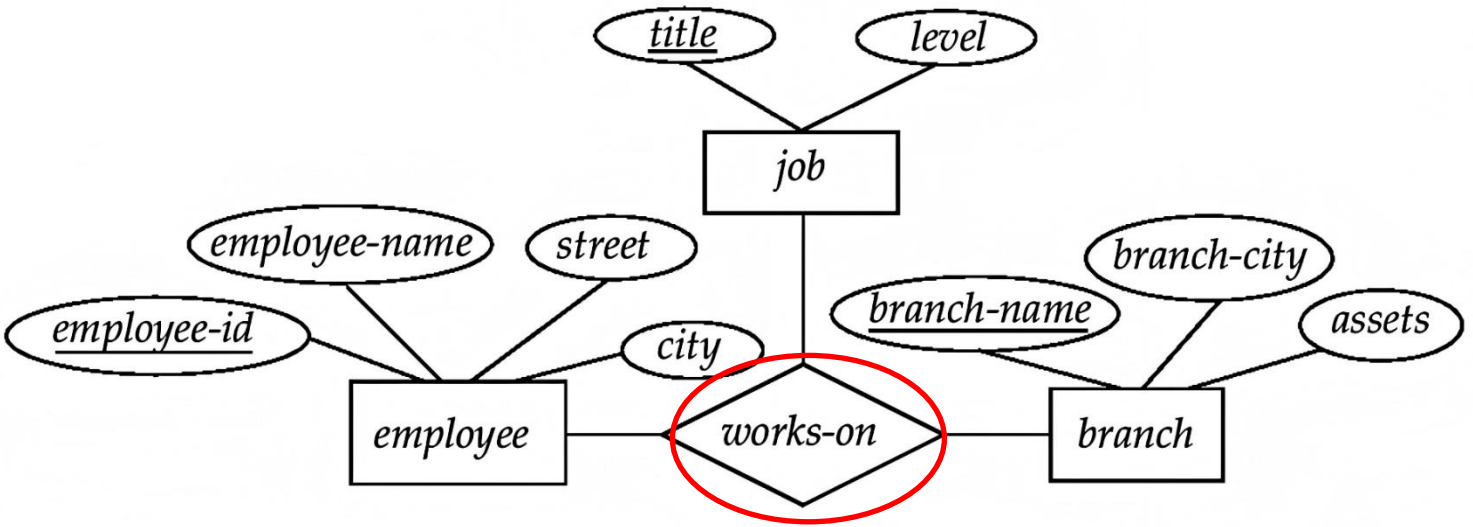
Keys (键/码)

- **超码（Superkey)**
  - A superkey of an entity set is a set of one or more attributes whose values uniquely determine each entity
- **候选码（Candidate key)**
  - A candidate key of an entity set is a **minimal superkey**
    - student_id is a candidate key of student
    - account_number is a candidate key of account
- **主码（Primary key)**
  - Although several candidate keys may exist, one of the candidate keys is selected to be the primary key

联系集的码

- The **combination of the primary keys** of the participating entity sets forms a superkey of a relationship set
  - (customer_id, account_number) is the super key of depositor
- Must consider the mapping cardinality of the relationship set when deciding the candidate keys
- Need to consider the semantics of relationship set in selecting the primary key in case of more than one candidate key

三元联系的E-R图

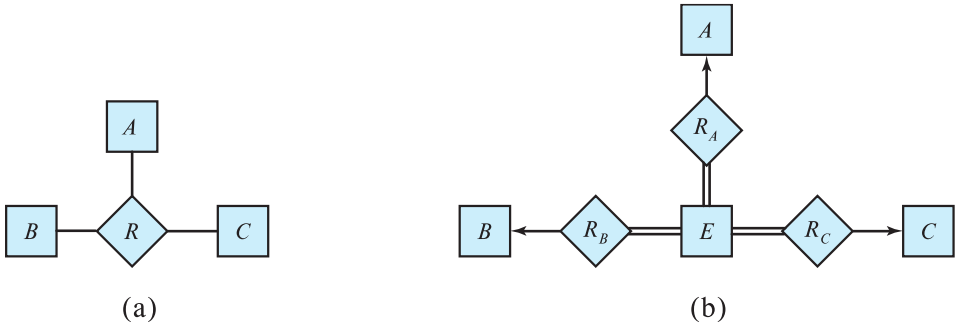

title
level
job
employee-name
street
branch-city
branch-name
assets
employee-id
city
employee
works-on
branch

二元联系 vs. 非二元联系

- Some relationships that appear to be non-binary may be better represented using binary relationships
  - E.g. A ternary relationship parents, relating a child to his/her father and mother, can be replaced by two binary relationships, i.e., father and mother
    - Using two binary relationships allows partial information (e.g., only mother being known)
  - But there are some relationships that are naturally non-binary
    - E.g. works-on, proj_guide

非二元联系的转换

- Any non-binary relationship can be represented using binary relationships by creating an **artificial entity set**
  - Replace R between entity sets A, B and C with an entity set E, and three relationship sets:
    - R_A, relating E and A
    - R_B, relating E and B
    - R_C, relating E and C
  - Create a special identifying attribute for E, and add any attributes of R to E
  - For each relationship (a_i , b_i , c_i) in R
    - add a new entity e_i in the entity set E
    - add (e_i, a_i) to R_A
    - add (e_i, b_i) to R_B
    - add (e_i, c_i) to R_C



弱实体集（Weak Entity Sets）

- **Entity set section**
  - uniquely identified by a course_id, semester, year, and sec_id.
  - related to course entities

- **Relationship set sec_course between entity sets section and course**
  - 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
  - **Solution A**: 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
  - **Solution B**: Not store the attribute course_id in the section entity and 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
- **Way out**
  - Treat sec_course as **a special relationship** that provides extra information, i.e., the course_id, required to identify section entities uniquely
  - The existence of a **weak entity set** is dependent on another entity, called its identifying entity (标识性实体)
  - 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

弱实体集（续）

- **Strong entity set (强实体集)**
  - An entity set that is not a weak entity set is termed a strong entity set.
- **Weak entity set & identifying entity**
  - Every weak entity must be associated with an identifying entity. The weak entity set is said to be existence dependent on the identifying entity set (**标识性实体集**).
  - The identifying entity set is said to own 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**)

弱实体集（续）

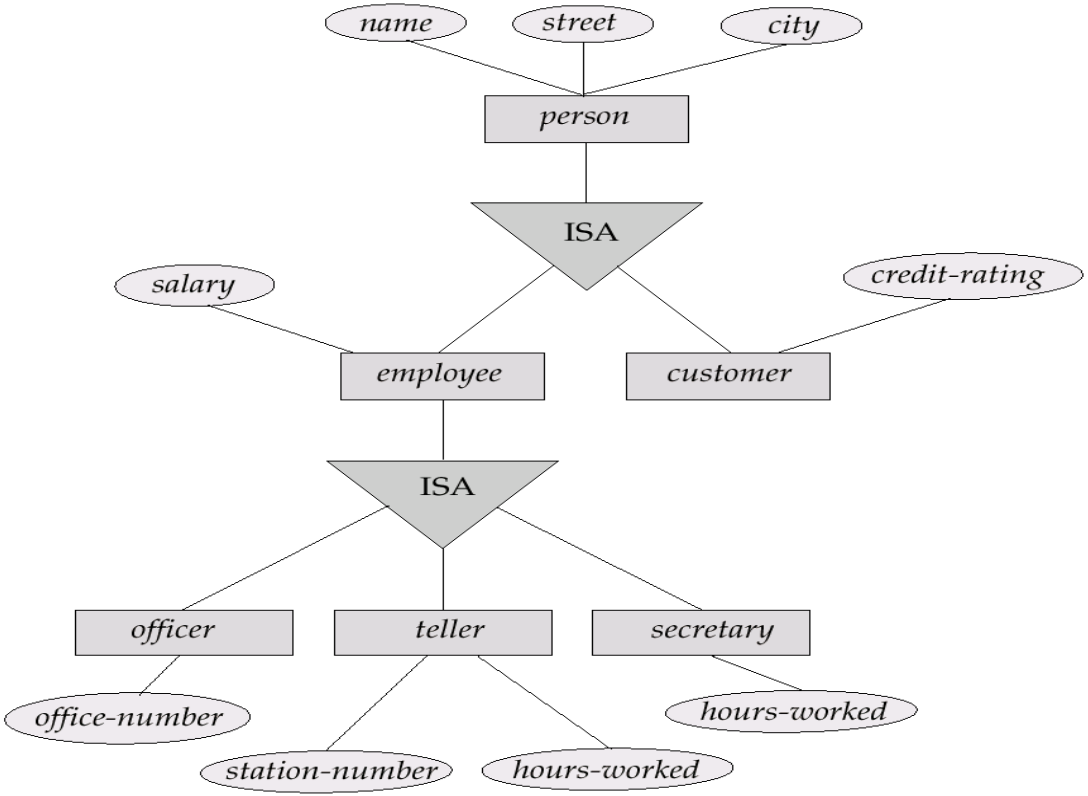
- In E-R diagrams, a weak entity set is depicted via a double rectangle
- The discriminator of a weak entity set is underlined with a dashed line
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond
- Primary key for section – (**course_id**, sec_id, semester, year)

特化（Specialization）

- **自上而下的设计过程**
  - 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

- **Attribute inheritance（属性继承）**
  - A lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked

特化的示例


name
street
city
person
ISA
salary
credit-rating
employee
customer
ISA
officer
teller
secretary
office-number
station-number
hours-worked
hours-worked

泛化（Generalization）

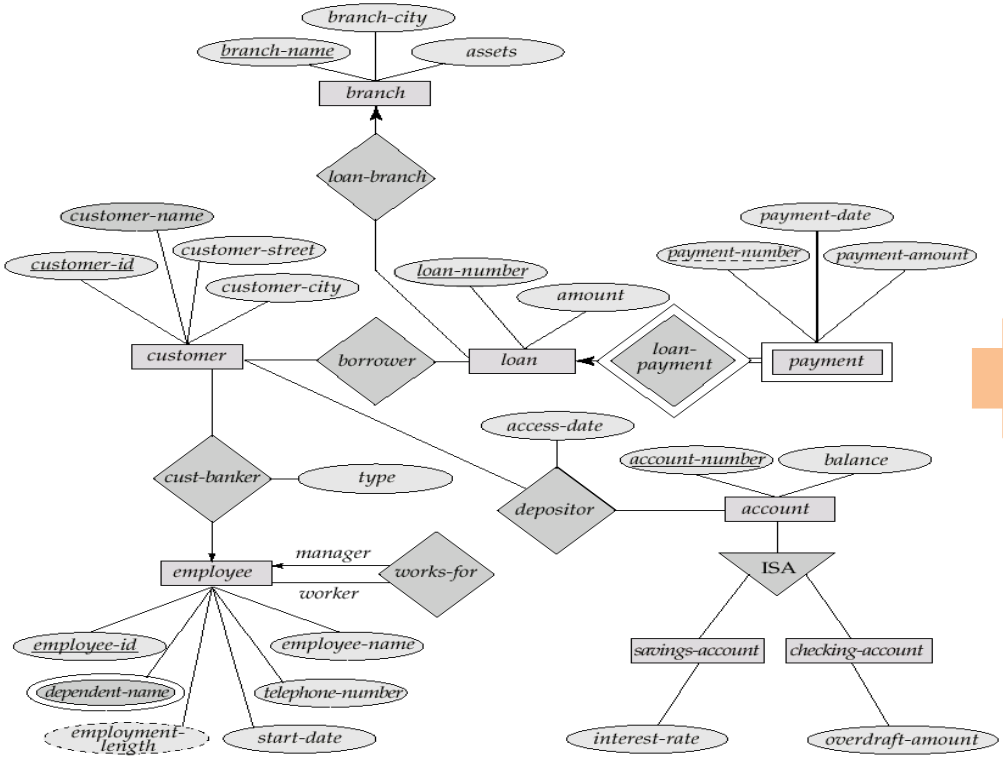
- **自下而上的设计过程**
  - Combine a number of entity sets that share the same features into a higher-level entity set
- **特化与泛化**
  - Specialization and generalization are inversions of each other
  - They are represented in the same way in an E-R diagram

特化和泛化

- One entity set may have multiple specializations based on different features
  - E.g., permanent-employee vs. temporary-employee, officer vs. secretary vs. teller
  - Each particular employee would be
    - a member of one of permanent-employee or temporary-employee, and
    - a member of one of officer, secretary or teller
- The ISA relationship is also referred to as superclass-subclass relationship

目录

转换为关系模式



E-R图

- branch = (branch_name, branch_city, assets)
- customer = (customer_id, customer_name, customer_street, customer_city)
- loan = (loan_number, amount)
- account = (account_number, balance)
- employee = (employee_id, employee_name, telephone_number, start_date)

- dependent_name = (employee_id, dname) (derived from a multivalued attribute)

- account_branch = (account_number, branch_name)
- loan_branch = (loan_number, branch_name)
- borrower = (customer_id, loan_number)
- depositor = (customer_id, account_number, access_date)
- cust_banker = (customer_id, employee_id, type)
- works_for = (worker_employee_id, manager_employee_id)
- payment =(loan_number, payment_number, payment_date, payment_amount)
- savings_account = (account_number, interest_rate)
- checking_account = (account_number, overdraft_amount)

关系模式

转换为关系模式（续）

- **Reduction of an E-R diagram to tables**
  - In general, for each entity set and relationship set, there is a unique table
  - Each table has a number of attributes
  - Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram

实体集→关系表

- A strong entity set is reduced to a table with the same attributes

customer-id	*customer-name*	*customer-street*	*customer-city*
019-28-3746	Smith	North	Rye
182-73-6091	Turner	Putnam	Stamford
192-83-7465	Johnson	Alma	Palo Alto
244-66-8800	Curry	North	Rye
321-12-3123	Jones	Main	Harrison
335-57-7991	Adams	Spring	Pittsfield
336-66-9999	Lindsay	Park	Pittsfield
677-89-9011	Hayes	Main	Harrison
963-96-3963	Williams	Nassau	Princeton

Customer

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Instructor

复合和多值属性的转换

- **Composite attributes** are flattened out by creating a separate attribute for each component attribute
- A multi-valued attribute M of an entity E is represented by a separate table EM
  - Table EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M
  - Each value of the multivalued attribute maps to a separate row of the table EM

弱实体集的转换

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set
  - E.g., **还款记录**

loan-number	*payment-number*	*payment-date*	*payment-amount*
L-11	53	7 June 2001	125
L-14	69	28 May 2001	500
L-15	22	23 May 2001	300
L-16	58	18 June 2001	135
L-17	5	10 May 2001	50
L-17	6	7 June 2001	50
L-17	7	17 June 2001	100
L-23	11	17 May 2001	75
L-93	103	3 June 2001	900
L-93	104	13 June 2001	200

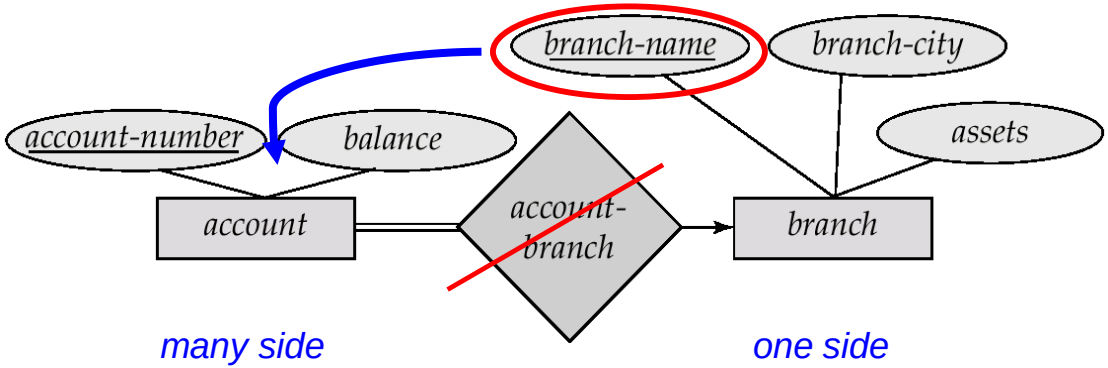
联系集→关系表

- **Many-to-many relationship set**
  - Represented as a table with columns for the primary keys of the two participating entity sets, and the attributes of the relationship set.
  - E.g., table for the relationship set borrower

customer-id	*loan-number*
019-28-3746	L-11
019-28-3746	L-23
244-66-8800	L-93
321-12-3123	L-17
335-57-7991	L-16
555-55-5555	L-14
677-89-9011	L-15
963-96-3963	L-17

联系集→关系表（续）

- **Many-to-one and one-to-many relationship sets**
  - Can be represented by adding an extra attribute to the many side, containing the primary key of the one side
  - E.g., instead of creating a table for relationship account-branch, add an attribute branch-name to the entity set account



联系集→关系表（续）

- **One-to-one relationship sets**
  - either side can be chosen to act as the “many” side

特化→关系表

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

table	table attributes
person	*name, street, city*
customer	*name, credit-rating*
employee	*name, salary*

  - **Drawback**: Querying information about entities, e.g., employee, requires accessing two tables

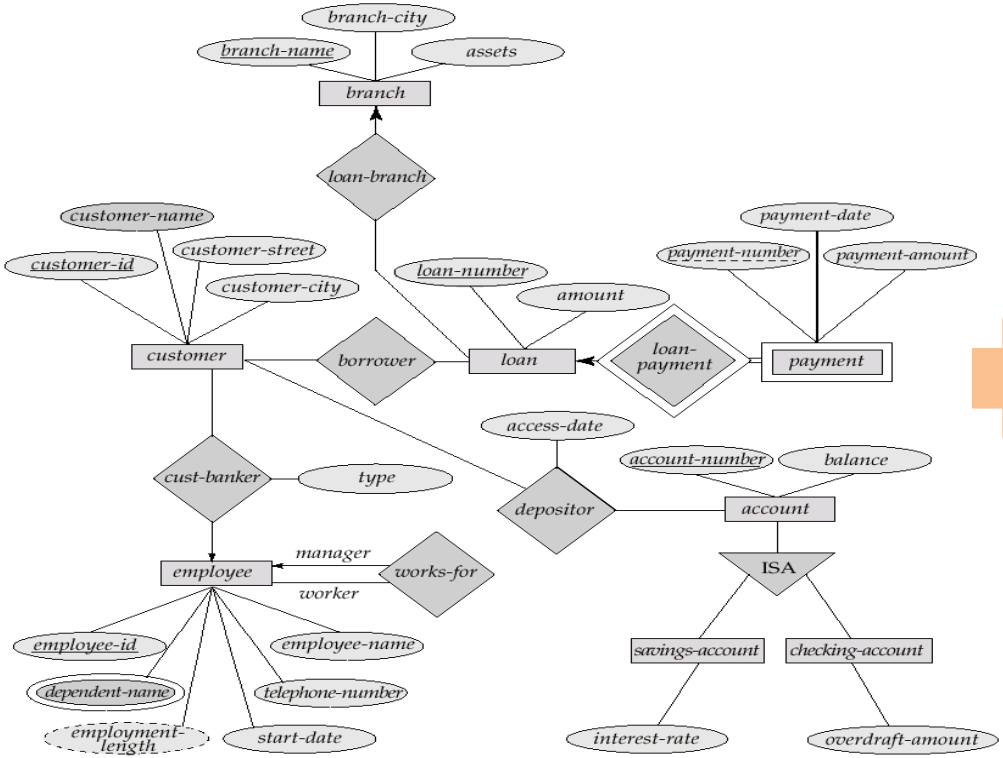
特化→关系表（续）

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

table	table attributes
person	*name, street, city*
customer	*name, street, city, credit-rating*
employee	*name, street, city, salary*

  - **Drawback:** street and city are stored redundantly for customers and employees

E-R图和关系模式-Banking DB



E-R图

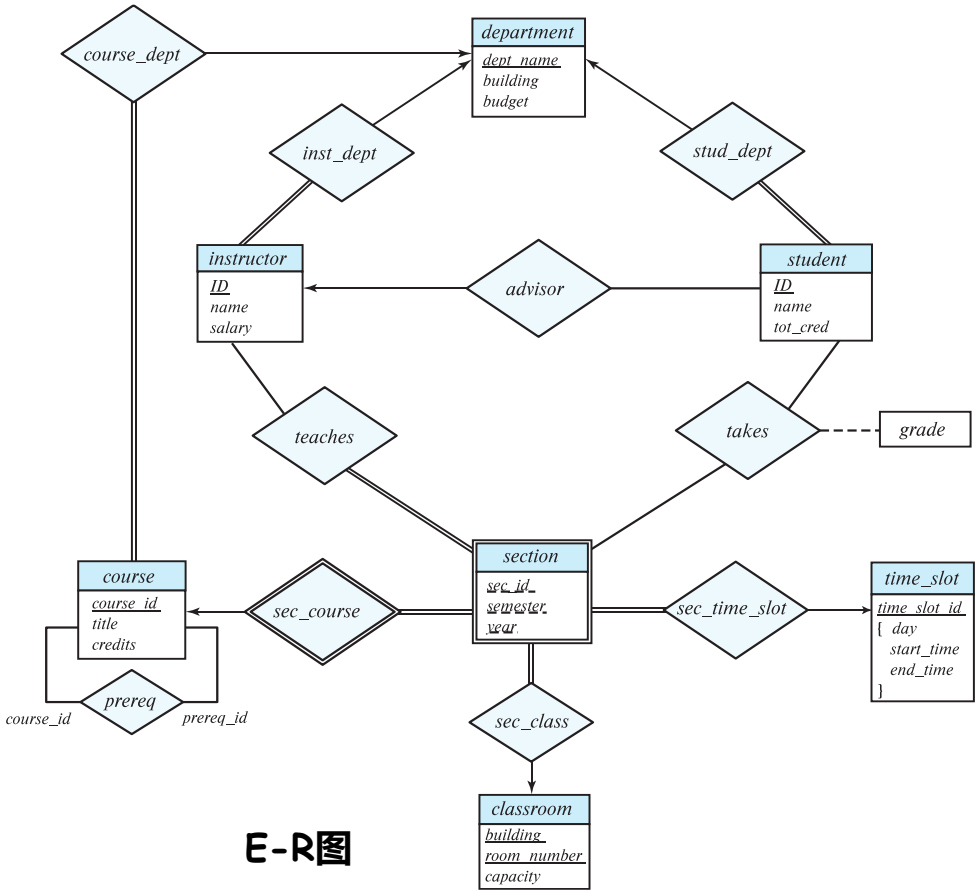
- branch (<u>branch_name</u>, branch_city, assets)
- customer (<u>customer_id</u>, customer_name, customer_street, customer_city)
- loan (<u>loan_number</u>, amount)
- account (<u>account_number</u>, balance)
- employee (<u>employee_id</u>, employee_name, telephone_number, start_date)

- dependent_name (<u>employee_id, dname</u>) (derived from a multivalued attribute)

- account_branch (<u>account_number</u>, branch_name)
- loan_branch (<u>loan_number</u>, branch_name)
- borrower (<u>customer_id, loan_number</u>)
- depositor (<u>customer_id, account_number</u>, access_date)
- cust_banker (<u>customer_id, employee_id</u>, type)
- works_for (<u>worker_employee_id</u>, manager_employee_id)
- Payment (<u>loan_number, payment_number</u>, payment_date, payment_amount)
- savings_account (<u>account_number</u>, interest_rate)
- checking_account (<u>account_number</u>, overdraft_amount)

关系模式

E-R图和关系模式-University DB



E-R图

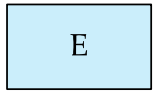
classroom(*building*, *room_number*, *capacity*)
department(*dept_name*, *building*, *budget*)
course(*course_id*, *title*, *credits*)
instructor(*ID*, *name*, *salary*)
student(*ID*, *name*, *tot_cred*)
teaches (*ID*, *course_id*, *sec_id*, *semester*, *year*)
takes (*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
prereq (*course_id*, *prereq_id*)
advisor (*s_ID*, *i_ID*)
sec_course (*course_id*, *sec_id*, *semester*, *year*)
sec_time_slot (*course_id*, *sec_id*, *semester*, *year*, *time_slot_id*)
sec_class (*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*)
inst_dept (*ID*, *dept_name*)
stud_dept (*ID*, *dept_name*)
course_dept (*course_id*, *dept_name*)

关系模式

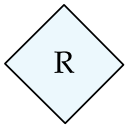
数据库设计的不同阶段

- **S1: Requirement analysis**
  - Data storage requirement
  - Functional requirements analysis
    - Describe the operations that will be performed on the data
- **S2: Conceptual design (E-R Model)**
- **S3: Logical implementation**
  - Mapping from conceptual model to implementation model
  - E.g., relational model, OO model
- **S4: Physical implementation**
  - Specify physical features of the database, e.g., buffer size, index…

E-R图中的符号

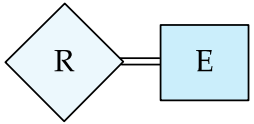


entity set

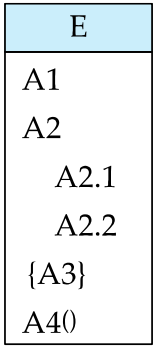


relationship set

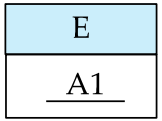
identifying
relationship set
for weak entity set



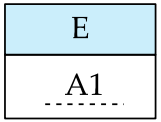
total participation
of entity set in
relationship



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

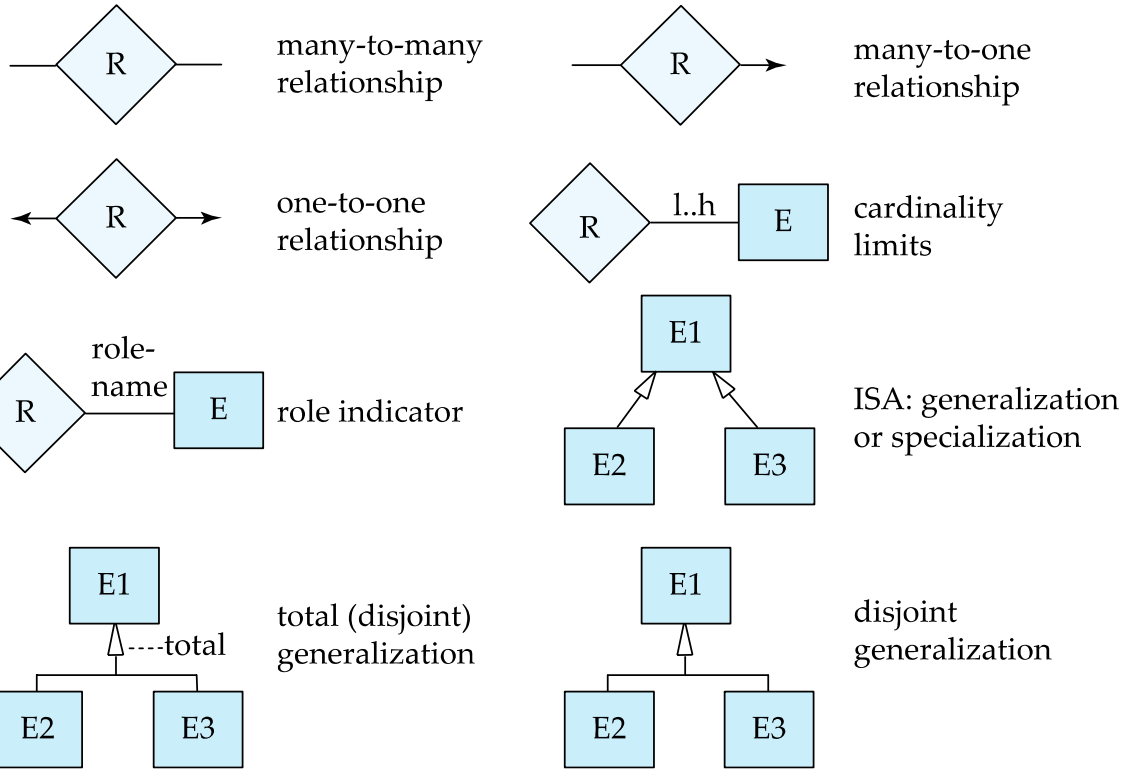


primary key



discriminating
attribute of
weak entity set

E-R图中的符号（续）


R
many-to-many relationship
R
many-to-one relationship
R
one-to-one relationship
R
l..h
E
cardinality limits
R
role-name
E
role indicator
E1
E2
E3
ISA: generalization or specialization
E1
----total
E2
E3
total (disjoint) generalization
E1
E2
E3
disjoint generalization

E-R模型设计工具

- **Freedgo Design**
  - https://www.freedgo.com/
- **Lucidchart**
  - https://www.lucidchart.com/pages/
- **Visual Paradigm**
  - https://www.visual-paradigm.com/cn/
- **Edrawmax**
  - https://www.edrawsoft.cn/

E-R模型总结

- Conceptual design follows requirements analysis
  - Yield a high-level description of data to be stored
- E-R model is popular for conceptual design
  - Constructs are expressive, close to the way people think about their applications
  - Basic constructs: **entities**, **relationships**, and **attributes** (of entities and relationships)
  - Additional constructs: weak entities, ISA hierarchies
- **Integrity constraints in E-R model**
  - Key constraints, participation constraints, and overlap/covering constraints for ISA hierarchies. Some foreign key constraints are also implicit in the definition of a relationship set
  - Some constraints (e.g., **functional dependencies**) cannot be expressed in the E-R model

E-R模型总结（续）

- **E-R design is subjective**
  - There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:
    - Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies

- **Ensuring good database design**
  - The generated relational schema should be analyzed and further refined
  - FD information and normalization techniques are useful (《数据库系统概念》第7章)