第二章 数据模型

2.1 数据模型的概念

在数据建模时，**数据模式（data schema）【结果】**是用**数据模型（data model）【手段】**来描述的

1.多级数据模型（multilevel data models）

概念数据模型（conceptual data model）：概念化结构，面向现实世界/用户，与DBMS无关 e.g. E-R模型、O-O模型
逻辑数据模型（logical data model）：逻辑结构，面向用户、面向实现，与DBMS有关 e.g. 网状模型、层次模型、关系模型、O-O模型
物理数据模型（physical data model）：物理存储结构，面向机器世界/实现，与DBMS、OS、硬件有关

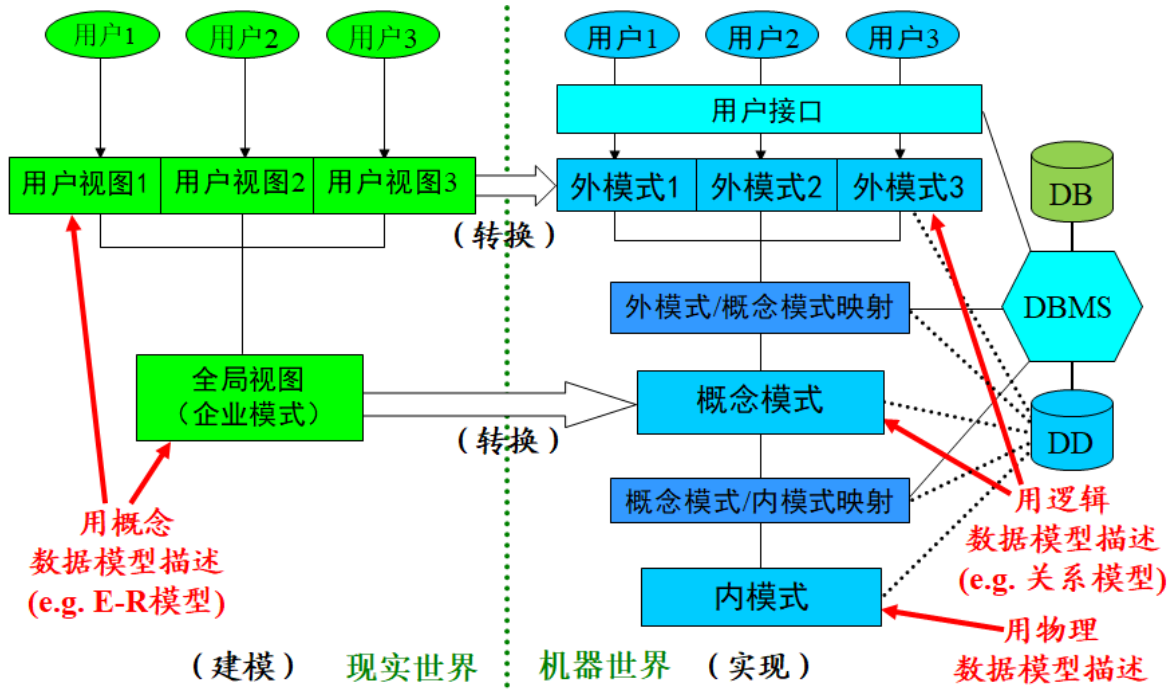


Figure: The DB Modeling, Schema Conversion and Implementation Process

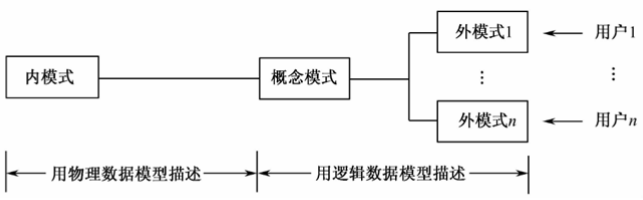


图 1-4　数据模式的分级

学生记录可以定义为图 1-3(a)的形式，这是数据模式。而图 1-3(b)是其一个实例。

姓名	学号	性别	出生年份	籍贯	系别	入学时间

(a)

王彤	0709135	女	1989	江苏	计算机	2007

(b)

图 1-3　数据模式及其实例

2.数据模式（data schema）

运用某种**数据模型【手段】**对一个企业（enterprise）/组织（organization）的一组数据的结构、联系和约束的描述**【结果】**

3.数据模型（data model）

用来描述数据的一组概念和定义，这种描述包括三个要素/两种特性：
数据的结构→数据的**静态特性** 数据的逻辑/物理结构和数据间的联系
对数据的约束→数据的**静态特性** 语义施加在数据上的约束（称完整性约束）
数据上的操作→数据的**动态特性** 定义在数据上的操作，即：数据查询、更新（增、删、改）
后面将从三要素角度介绍关系数据模型；从结构、约束角度介绍E-R数据模型

2.2 关系数据模型

一、结构

是以集合论中的关系（relation）概念为基础的。

1.定义：属性和域

要描述现实世界中的一个事物（实体），常常取其若干特征来表示，每个特征称为**属性（attribute）**
每个属性对应一个值的集合，作为该属性取值范围，称为该属性的**域（domain）**
e.g. 姓名、性别、年龄…是人的属性 “性别”的域是{男,女} or {M, F} or {男,女,未知,变性}…
注：a. 域中的值必须是**原子数据（atomic data）**，称这种限制为**满足第一范式（first normal form,1NF）**条件；若域中的值是**非原子数据**，即**组合（aggregated）数据**，则称为**非第一范式（non-first normal form, NF2）**条件
b. 允许某些属性的值未知或无值，用**空值（NULL）**表示

2.定义：笛卡尔积、元组和分量

给定一组域D1，D2，…，Dn，这些域中可以有相同的，它们的**笛卡尔积（Cartesian product）**定义为以下集合：
D1×D2×…×Dn□={ (d1, d2, …, dn) | di∈Di, i=1, 2, …, n }。
以上集合中的每个元素(d1, d2, …, dn) 称作一个**n元组（n-tuple）**，简称**元组（tuple）**；元素中每个值di 称作一个**分量（component）**

注：笛卡尔积可表示成一个**二维表（table）**，即：由行、列所组成的平坦表格，如Excel表单。

3.定义：关系

笛卡尔积D1×D2×…×Dn的某个子集称为定义在域D1, D2, …, Dn上的一个**关系（relation）**，用r表示，r ⊆ D1×D2×…×Dn
对关系也要区分型（type）和值（value），**关系模式**用R(A1/D1, A2/D2, …, An/Dn) 或 R(A1, A2, …, An)表示，R称为关系的**模式名**，Ai为关系的**属性名**，n为关系的**元（arity）或目（degree）**
注： a. 关系的性质：【关系可表示成二维表】
传统上，关系必须满足1NF条件；
行、列的次序无所谓；
任意两行不能全同；
列是**同质的（homogeneous）**，即值来自于相同域
b. 关系既可用来描述**实体**，又可用来描述**实体间的联系**

4.定义：键和超键

关系中满足如下两个条件的属性（组）称为此关系的**候选键（candidate key）**，简称为**键（key）**：

a. 决定性条件：这个属性（组）的值唯一地（uniquely）决定了其他属性的值（因而也决定/标识了整个元组）；

b. 最小性条件：这个属性（组）的任何真子集（proper subset）均不满足决定性条件。

若键由关系中所有属性所组成，则称为**全键（all key）**。

关系中包含（候选）键的属性（组）称为**超键（superkey, i.e., the superset of a key）。超键 ⊇ 键 或 键 ⊆ 超键**

e.g. 学生(学号, 姓名, 性别, 专业, 入学年度, 身份证号)

键1={学号}，键2={身份证号}；{学号}、{身份证号}也是超键。而{学号, 性别}、{学号, 身份证号}...也是超键，但不是键。

5.定义：主属性和非主属性

包含在某个关系中任何一个（候选）键中的属性称为此关系的**主属性（prime attribute）**；不包含在任何（候选）键中的属性称为**非主属性（non-prime attribute）**

6.定义：主键和外键

在关系模式机器实现时，从一个关系中（多个）键中选定一个作为此关系模式的键，称被选定者为**主键（primary key, PK）**；其他键称为**候补键（alternate key）**

若一个关系A中某个属性（组）不是本关系的键，但它的值引用了其他关系（或本关系）B中某个键的值，则称此属性（组）为本关系的**外键（foreign key, FK）**。A称为施引关系（referencing relation），B称为被引关系（referenced relation）

二、约束

1.语法约束

关系模式R(A1/D1, A2/D2, ..., An/Dn)的定义实际上仅指出了R的任一实例（关系）r中的每个元组应满足的**语法约束**:

r = {t1, t2, ..., tm} ⊆ D1×D2×...×Dn

上式中每个元组 t = (v1, v2, ..., vn) ∈ D1×D2×...×Dn,

其中，vi ∈ Di，i=1, 2, ..., n

在实现时，域Di通常用数据类型（及取值规则）来约束。

但是，数据是有**语义约束的（即完整性约束）**：

设 rc 是R的所有满足完整性约束的元组的集合，显然应： r ⊆ rc ⊆ D1×D2×...×Dn

因此，一个好的DBMS应尽可能地具备**完整性约束的定义和检查机制。"定义"**在模式定义时申明；"检查"在数据库初始加载及事后更新时进行。

2.完整性约束的类型

域完整性约束（domain integrity constraints）

属性值应在域中取值

属性值是否可为NULL？（**由语义所决定**）

实体完整性约束（entity integrity constraints）

每个关系应有一个PK，每个元组的PK值应唯一，且不能为NULL

引用完整性约束（referential integrity constraints）

一个关系中的FK值必须引用（另一个关系或本关系中）实际存在的PK值，否则只能暂时取NULL（称**悬空引用**）

一般完整性约束 / 业务规则（business rules）

由特定应用领域中的业务规则所决定，由用户明确地自定义

迄今为止还没有一个DBMS能全面实现一般完整性约束，但总的趋势是朝这个方向努力。

三、操作

有两类/三种在表达能力上等价的关系操作，称为“纯”（“pure”）查询语言，而不是商用数据库语言（SQL）

关系代数（relational algebra)

过程性的（procedural），由一组操作所组成：传统的**集合运算**（并、交、差、笛卡尔积，等）和**关系专用操作**（选择、投影、连接、除，等），每个操作以一个或多个关系为输入，以结果关系为输出

关系演算（relational calculus)

非过程性的（nonprocedural），使用谓词逻辑（predicate logic）来定义所需的结果。根据变量是元组（tuple）还是域（domain），进一步区分为：**元组关系演算 vs. 域关系演算**

1.关系代数操作 - 筛选型操作

选择（selection）

选出关系r中满足<选择条件>的元组，构成结果关系（横向筛选，一元操作）

б <选择条件> (r) = { t | t ∈ r AND <选择条件> }

投影（projection）

选出关系r中<属性表>所列出的诸属性列的值，构成结果关系（纵向筛选，一元操作）

Π <属性表> (r) = { t [<属性表>] | t ∈ r }

2.关系代数操作 - 传统的集合运算

要求参与操作的关系并兼容（union compatibility），即关系具有相同的元/目、且对应的属性域相同。

并（union）r∪s = { t | t ∈ r OR t ∈ s }

差（difference）r − s = {t | t ∈ r AND (t ∉ s) }

交（intersection）不是独立的操作：r∩s = r − (r − s)

3.关系代数操作 - 拼接型操作

笛卡尔积（Cartesian product）

r×s = { <t, g> | t ∈ r AND g ∈ s }

序偶<t, g>称元组t与元组g的**拼接（concatenation）**

r×s的元为nr+ns，结果关系中的元组数为|r|x|s|。

连接（join）： 从两个关系 r 和 s 的笛卡尔积的所有元组拼接中选出满足<连接条件>者，构成结果关系：

r ⋈ <连接条件> s = б<连接条件> (r×s)

<连接条件>的一般形式为：C1 AND C2 AND ... AND Ck，其中，Ci形如Ai θ Bi，而且，Ai, Bi分别为r和s中的属性，θ为关系运算符（relational operator）：{<, <=, >, >=, =, !=}

连接也称**θ连接（theta-join）**。当θ为“=”时，有两类特殊的连接：

等连接 （equi-join）：在连接结果中保留两个关系中重复的属性列

自然连接 （natural join）：在连接结果中只保留重复属性列之一

连接常指**自然连接**。

由于r×s = r ⋈ TRUE s，故笛卡尔积与连接不是相互独立的操作，实际中常取其一，而且连接更有意义！

4.关系代数操作 - 其他操作

除（division）、外连接（outer join）、外并（union）
与前述操作不独立；而且很少使用

综上所述，**集合{б, Π, ∪, −, ×}或{б, Π, ∪, −, ⋈}是关系完备（relationally complete）**操作集。支持关系完备操作集的DBMS称**关系完备的 DBMS**，或者说DBMS具有**关系完备性/relational completeness**

目前，大部分SQL RDBMS, 如：Oracle, IBM DB2, SQL Server是关系完备的，而大部分PC数据库系统, 如：FoxBase, FoxPro 不是关系完备的。

四、例子

职工关系emp:

empno	ename	job	sal	deptno
25	张三	accountant	4000	1
30	李四	manager	5000	2
31	王五	salesman	3000	2
32	赵六	salesman	3500	2

部门关系 dept:

deptno	dname	loc
1	财务部	Shanghai
2	市场部	Nanjing

（1）б deptno = 2 AND job = ‘salesman’ (emp)

empno	ename	job	sal	deptno
31	王五	salesman	3000	2
32	赵六	salesman	3500	2

（2）**Π** job, sal (emp)

job	sal
accountant	4000
manager	5000
salesman	3000
salesman	3500

（3）(б deptno = 2 AND job = ‘salesman’ (emp)) ∪ (б deptno = 1 (emp))

empno	ename	job	sal	deptno
25	张三	accountant	4000	1
31	王五	salesman	3000	2
32	赵六	salesman	3500	2

（4） (б deptno = 2 (emp)) − (б job = ‘manager’ (emp))

empno	ename	job	sal	deptno
31	王五	salesman	3000	2
32	赵六	salesman	3500	2

(5) emp×dept

empno	ename	job	sal	deptno
25	张三	accountant	4000	1
30	李四	manager	5000	2
31	王五	salesman	3000	2
32	赵六	salesman	3500	2

deptno	dname	loc
1	财务部	Shanghai
2	市场部	Nanjing

拼接出4x2 = 8个元组

(6)

- 等/自然连接 $emp \bowtie_{emp.deptno = dept.deptno} dept$ 的结果

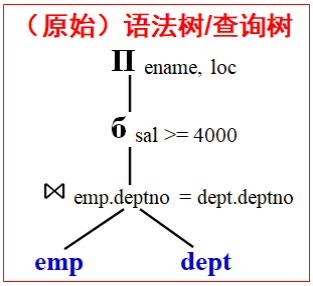
保留/去除重复属性列的值

empno	ename	job	sal	deptno	deptno	dname	loc
25	张三	accountant	4000	1	1	财务部	Shanghai
30	李四	manager	5000	2	2	市场部	Nanjing
31	王五	salesman	3000	2	2	市场部	Nanjing
32	赵六	salesman	3500	2	2	市场部	Nanjing

(7) 关系代数表达式对应一颗语法树/查询树（用于查询优化）

$$\Pi_{ename,\ loc}(\sigma_{sal >= 4000}(emp \bowtie_{emp.deptno = dept.deptno} dept))$$

（原始）语法树/查询树



ename	loc
张三	Shanghai
李四	Nanjing

2.3 对传统数据模型的评价

一、传统数据模型

1.传统数据模型（traditional data models）

层次模型
网状模型
关系模型（已学过）

2.非传统数据模型：后关系模型（post relational data models）

面向对象（object-oriented, O-O）模型
对象-关系（object-relational）模型
实体-联系（entity-relationship, E-R ）模型（讲解）

二、评价

1.肯定之处

向用户提供了统一的数据模型（如：关系模型）；
数据与程序之间具有相当程度的独立性；
向用户提供了统一的数据库语言（如：SQL）；
DBMS在数据共享性、安全性、完整性及故障恢复等方面提供了足够的保障。
总之，从文件系统到数据库系统，数据管理技术是一个飞跃！尤其是基于关系模型的RDBMS，在量大面广的联机事务处理（online transaction processing, OLTP）系统中基本上能满足应用的需求。

2.不足之处

以记录（record）为基础，不能很好地面向用户和应用：记录以实现方便为出发点，刻板地描述各种实体（entity）□——只能"削足适履"！
不能以自然的方式（natural way）表示实体之间的联系（relationships between entities）：实体间联系以面向实现的方式或非显式的方式来表示
语义贫乏（semantically poor）：无法明确、显式地描述实体间联系的语义
数据类型少（few data types），难以满足应用需要：不支持用户自定义（user-defined）数据类型、复杂数据类型、取值规则

2.4 E-R数据模型

1976年，Peter Pin-Shan Chen（陳品山）在其论文：The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), Volume 1, Issue 1, March 1976, Pages: 9–36 中首先提出E-R模型时，有三个目的（统一的、中间、概念数据模型，详见教材Page 33）。后来（一直到现在）， E-R模型主要用作数据建模（即DB概念设计）的有力工具。
Peter Chen模型称基本E-R模型（basic E-R model），后来有许多扩充版本，称扩充的E-R模型（extended E-R model, EER）。

一、基本E-R模型

1.三种抽象

①实体（entity）与弱实体（weak entity）：对事物的一种抽象
实体集（entity set）：对同类事物的一种抽象
实体（entity）：实体集中的一个实例，是对某类事物中某个具体事物的一种描述
e.g. 实体集 students = {e | e是学生}；而其实例是具体的学生，例如：张三, 李四∈students，都是具体的学生
实体键&实体主键：与关系模型中的相关概念对应
e.g. 学号是学生实体（集）的实体键（可选为实体主键）
弱实体：不能独立存在，依附于其他实体集中的某个实体（称所有者实体）。弱实体键必须包含其所有者实体的键
e.g. "职工"与"家属"

②属性（attribute）：对事物（或事物间联系）特征的一种抽象
原子属性：不可再分的数据项。
e.g. 学号，姓名，性别，等□ 是学生的原子属性
非原子属性：

组合属性/元组属性

e.g. 通讯地址：(邮编，省，市，街区地址)

街区地址：(街名，号码，公寓号)

多值属性/集合属性

e.g. 选修课程：{C语言，C++语言，Java语言}

③联系（relationship）：对事物之间某种关系的一种抽象

联系集（relationship set）：事物之间同类联系所组成的一个集合

联系（relationship）：联系集中的一个实例

e.g. 联系集 married (M,F) = □ { <e1,e2> | e1∈M ∧ e2∈F ∧ e1与e2是夫妻 }；

而<张三, 李梅>, <王五, 赵丽>∈ married (M,F)，是一对对具体的夫妻

联系也有属性

e.g. 联系集 married (M,F) 可以有一个属性： 婚礼日期wedding_date，

<张三, 李梅>的wedding_date = May 1, 2018

2.联系的语义约束（semantic constraints）

①基数比约束（cardinality ratio constraints）

对联系 R(E1, E2, ..., En),

当n = 2时，二元联系（binary relationship）其基数比可以是：1:1，1:N，M:N

当n > 2时，多元联系（multiway relationship），如三元联系 其基数比可以是：1:1:1，1:1:P，M:N:P，等

当n = 1时，自联系/递归联系（recursive relationship） 其基数比可以是：1:1，1:N，M:N

②参与约束（participation constraints）

对联系 R(E1, E2, ..., En) 中的某个实体集Ei，

若所有实体 ei ∈Ei 均参与联系R，则称实体集Ei是全参与的（total participation）

若存在实体ei ∈Ei 不参与联系R，则称实体集Ei是部分参与的（partial participation）

3.E-R数据模式与E-R图

运用前述E-R数据模型对一个企业/机构的全体数据进行建模后所得的结果称为E-R数据模式，通常简称为E-R模式（E-R schema）

E-R模式常用直观的E-R图（E-R diagram）来表示

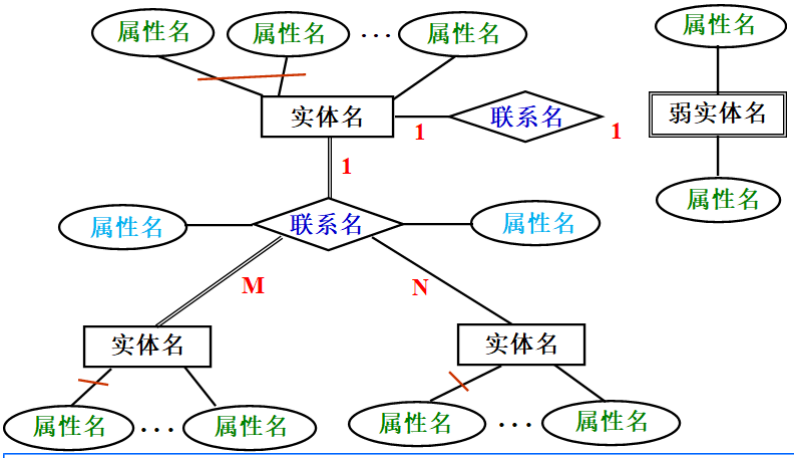
E-R图有各种符号体系（notation），教材中只是其中一种（基本上是Peter Chen的符号体系）

矩形表示实体，双线矩形表示弱实体

菱形表示实体间的联系，用线来连接实体与联系

单线/双线表示实体的部分/全参与，线上标注基数比

椭圆表示实体/联系的属性，用单线来连接实体/联系与属性，实体键（属性）进一步有横线标识



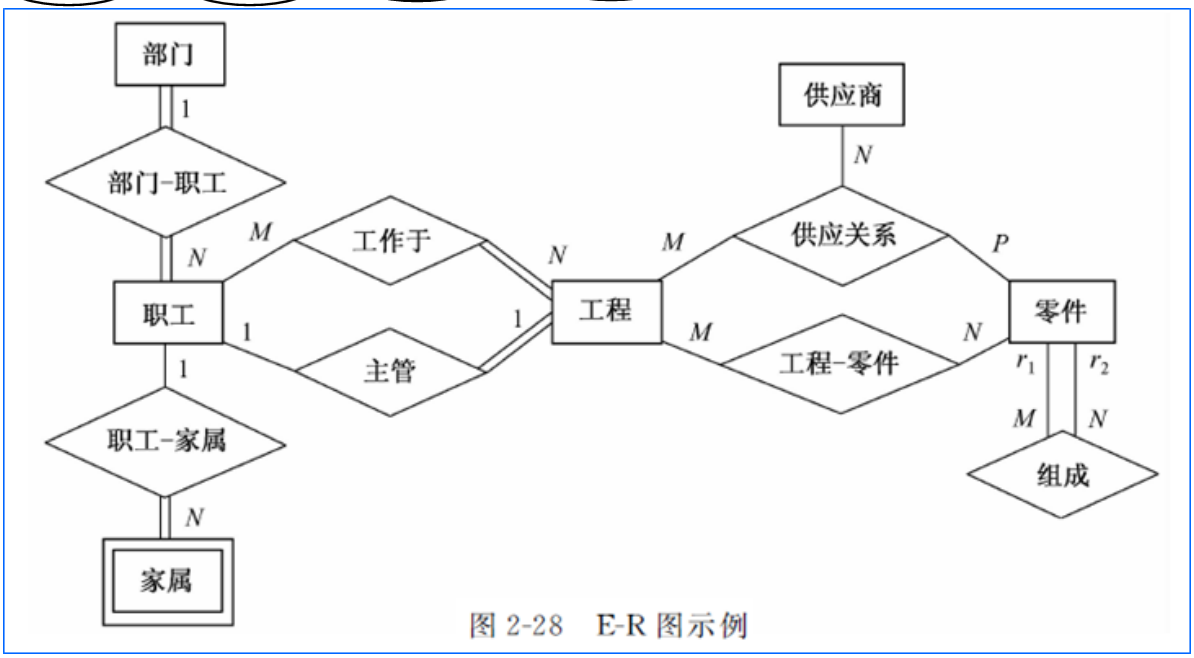


图 2-28　E-R 图示例

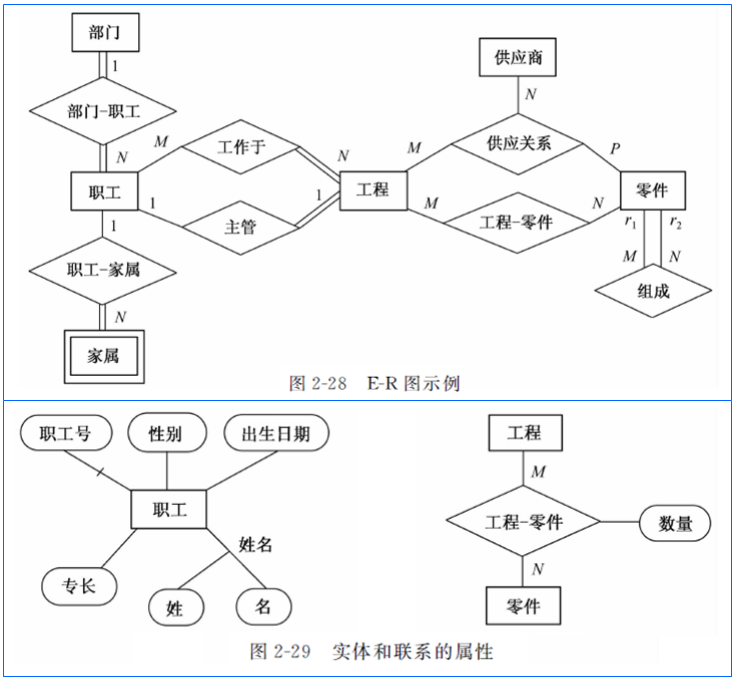


图 2-28　E-R 图示例

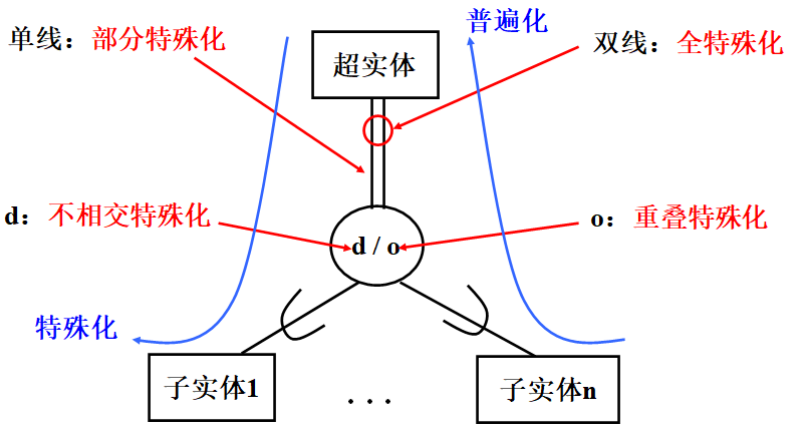
图 2-29　实体和联系的属性

E-R建模的设计原则

选择合适的抽象（appropriate abstract）
忠实性（faithfulness）
避免冗余（avoiding redundancy）
简单性（simplicity）

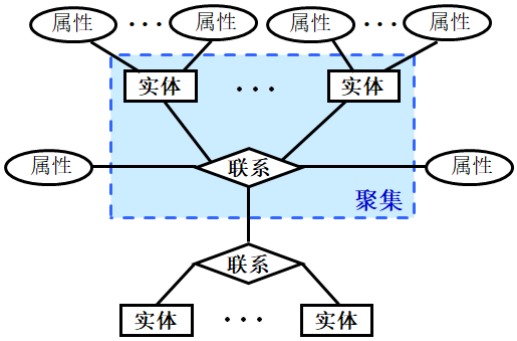
二、扩充E-R模型

1.特殊化（specialization）与普遍化（generalization）



2.聚集（aggregation）

参与某个联系的全部实体组成一个新实体（称聚集），其属性集是所有这些实体的属性及这个联系的属性之并集。
聚集可象一般实体一样参与联系，即：联系也可参与联系了！



3.范畴（category）

不同类型的实体组成新实体（称范畴）。这样，范畴也可参与联系了！