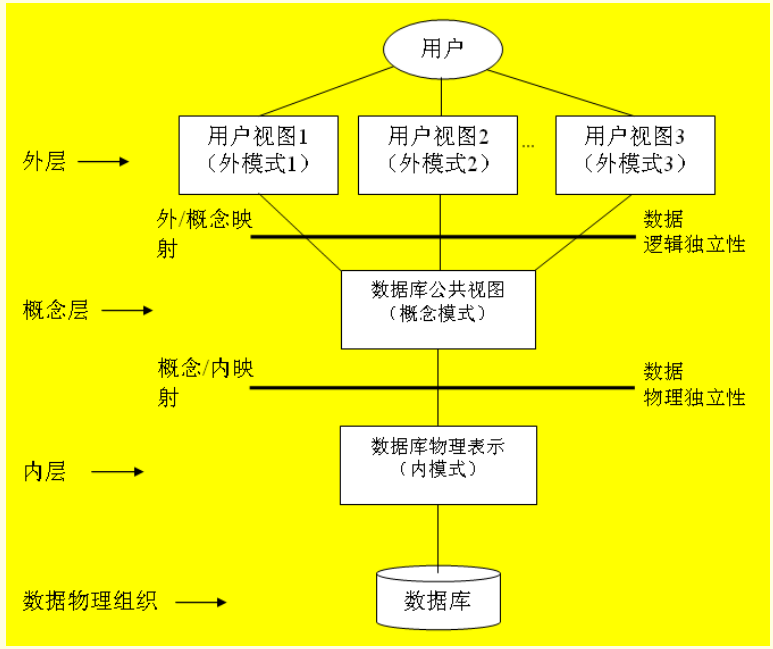
第2章　数据模型

2012. 02

目录 Contents

2.1 数据模型的概念

- **回顾：ANSI-SPARC三层抽象**
  - 数据库模式：数据库的（总体）描述，也称数据库的内涵
  - 数据库中的三种数据模式：
    - 外模式（external schema）或称子模式（subschema）：分别描述数据的不同视图
    - 概念模式（conceptual schema）：描述数据库中所有实体、属性与关系，以及完整性约束
    - 内模式（internal schema）：描述数据库的内部模型，包括数据域与存储记录的定义、表示方法、索引与存储结构等



2.1 数据模型的概念

- 数据模式（结果）需要数据模型（手段）来描述
- **数据模式（data schema）：**运用某种数据模型（手段）对一个企业（enterprise）或组织（organization）的一组相关数据的结构、联系和约束的描述（结果）
- **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/两个方面：
  - 数据的结构 → 静态特性
    - 数据的逻辑/物理结构和数据间的联系
  - 数据中的约束 → 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 →动态特性
    - 如何检索、更新（增、删、改）数据

2.1 数据模型的概念

- **多级数据模型**（multi-level data model）
  - 概念数据模型（conceptual data model）
    - 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）
    - 既面向用户、又面向实现
    - 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）
    - 面向机器世界/实现，描述数据的存储结构。与DBMS、OS、硬件有关

2.1 数据模型的概念

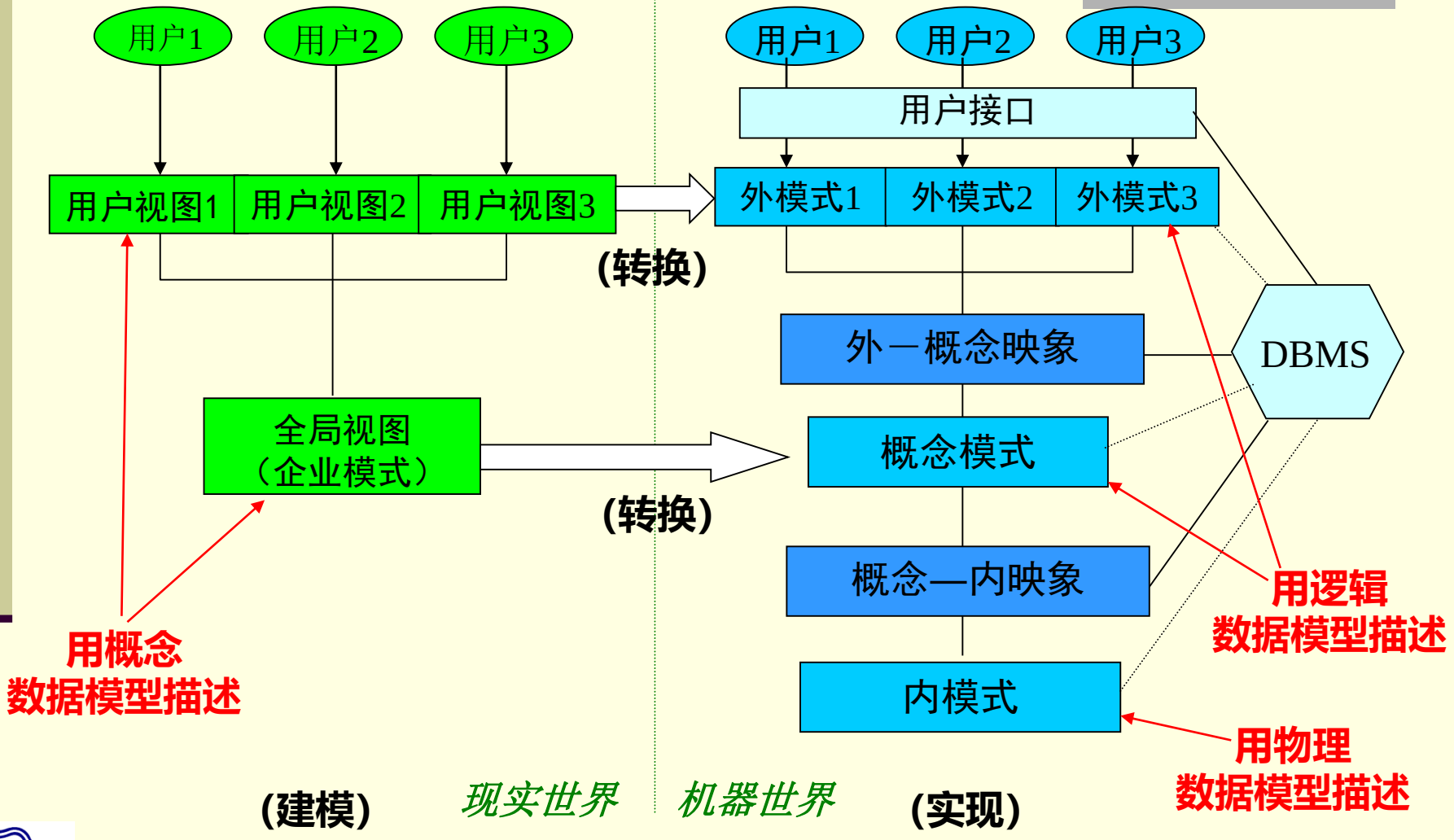
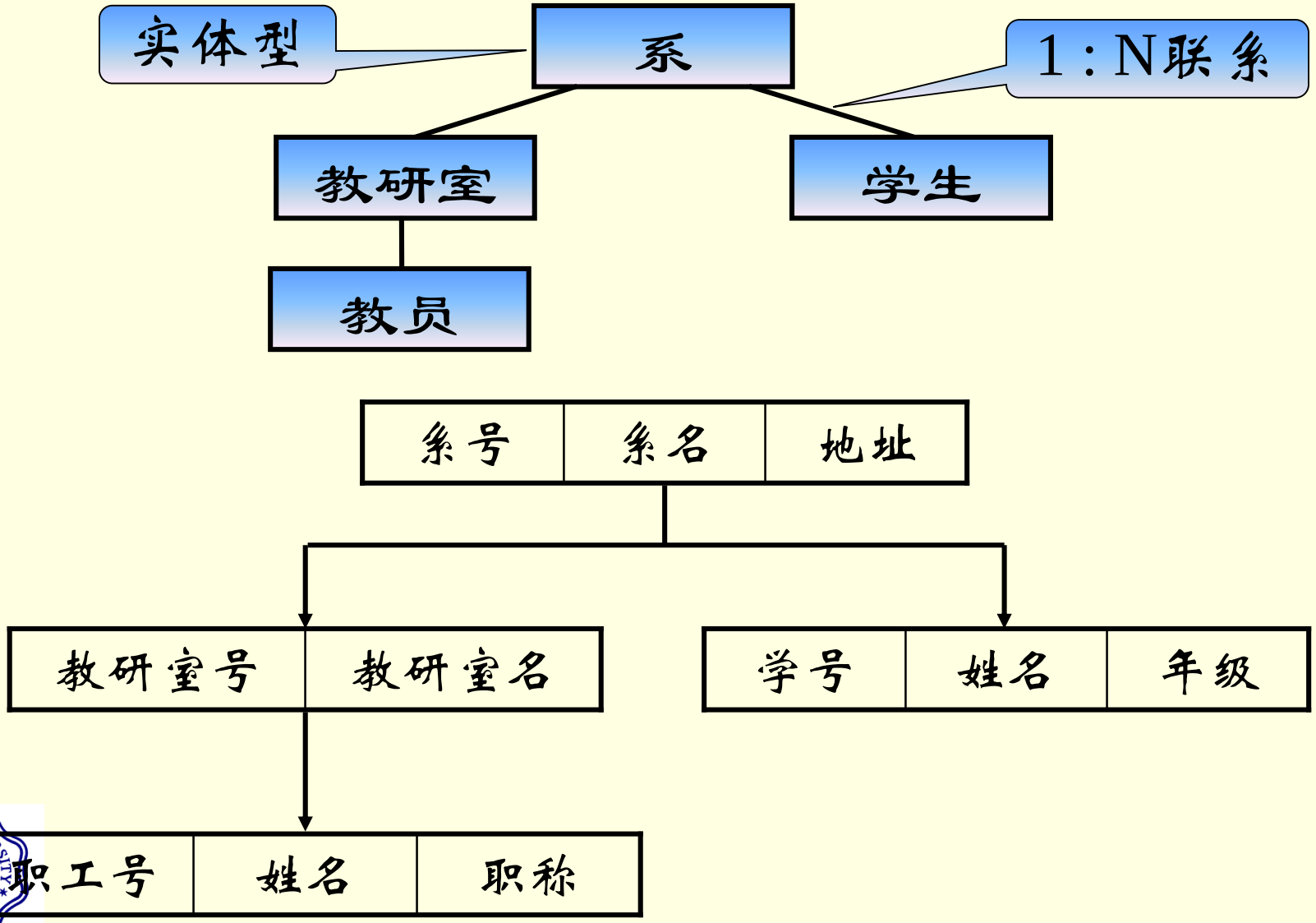
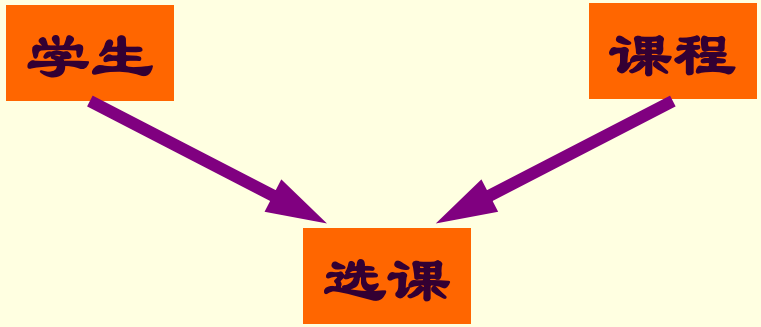


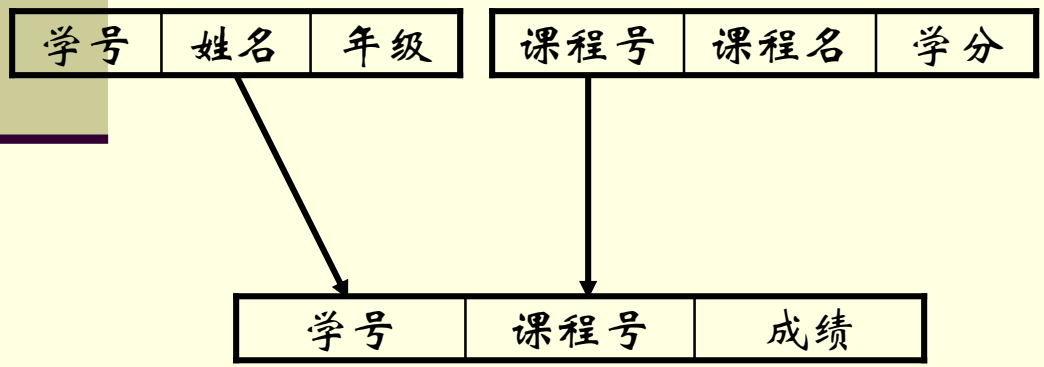
Figure: The DB Modeling, Schema Conversion and Implementation Process

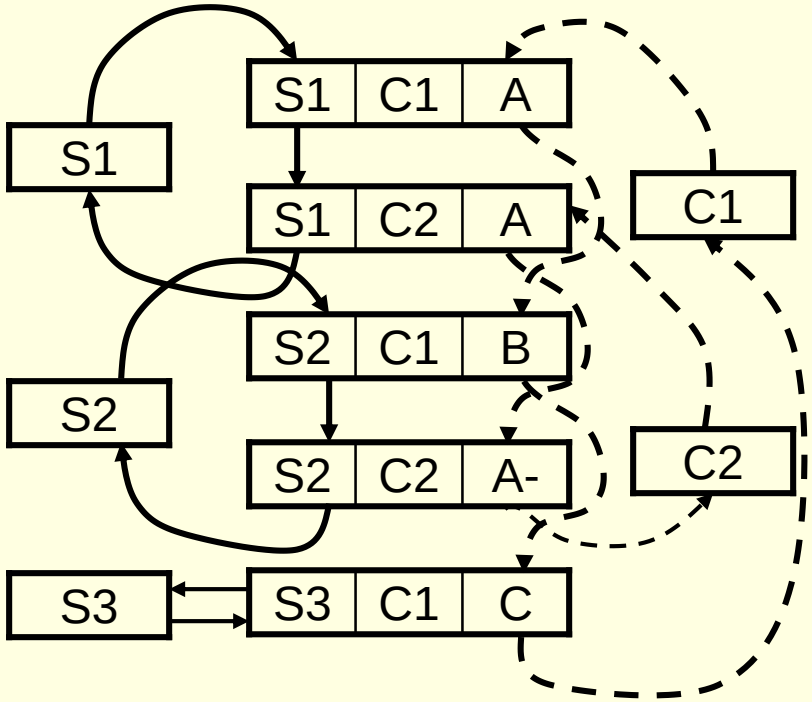
2.1 层次数据模型



2.3 网状数据模型


学生
课程
选课


学号
姓名
年级
课程号
课程名
学分
学号
课程号
成绩


S1
S2
S3
S1 C1 A
S1 C2 A
S2 C1 B
S2 C2 A-
S3 C1 C
C1
C2

目录 Contents

2.4 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。
- 定义：**属性**和**域**
  - 要描述现实世界中的一个事物，常常取其若干特征来表示，每个特征称为**属性(Attibute)**。
  - 每个属性对应一个值的集合，作为其取值范围，称为该属性的**域(Domain)**。
  - e.g. “性别”是人的属性，其域是{男，女} or {M，F} or {男，女，未知，变性}
  - **[注]**：a. 域中的值是**原子数据(Atomic Data)**，这种限制称**第一范式(First Normal Form,1NF)**条件；如域中的值可以是**组合数据(Aggregated Data)**，称**非第一范式(Non-First Normal Form,NF²)**条件.
  - b. 关系模型中允许某些属性的值未知或无值，此时，用**空值NULL**表示。

2.4 关系数据模型

- 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}。
  - 其中，每个元素(d1, d2, …, dn) 称作一个**n元组**(n-tuple)。元素中每个值叫作一个**分量**(Component)。
  - [注]：笛卡尔积可表示成一个**二维表**(Table)。

2.4 关系数据模型

- **e.g.** D1: 男人={王兵，李平}，D2: 女人={丁美，吴芳}，
  D3: 儿童={王小兵，李小平，吴小芳}

  则，D1×D2×D3={(王兵,丁美,王小兵), (王兵,丁美, 李小平),
  (王兵,丁美,吴小芳),(王兵,吴芳,王小兵),
  ……
  (李平,吴芳,李小平),(李平,吴芳,吴小芳)}

  其可表示成一个二维表：

列(Column)

D_1	D_2	D_3
王兵	丁美	王小兵
王兵	丁美	李小平
王兵	丁美	吴小芳
…	…	…
李平	吴芳	李小平
李平	吴芳	吴小芳

行(Row)

2.4 关系数据模型

- 定义：**关系**
  - D1×D2×…×Dn的某个子集称定义在域D1,D2,…,Dn上的一个**关系(Relation)**。
  - 用R(A1 / D1,A2 / D2,…,An / Dn)或R(A1,A2,…,An)表示。
  - 其中，R为**关系名**，Ai为**属性名**，n为关系的**目**(Degree)。
  - 对关系也要区分其型(Type)和值(Value), 其型称为关系的**模式**(Schema)/**内涵**(Intension); 其值称为关系的**实例**(Instance)/**外延**(Extension)。
  - **[注]**：a. 关系的性质：
    - 关系必须满足要求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）。
  - b. 关系既可用来表示一个实体，又用来表示实体间的联系。

2.4 关系数据模型

- 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)**，或简称**键(Key):**
    - a. 决定性条件：这个属性（组）的值唯一地决定了其他属性的值（因而也决定了整个元组）；
    - b. 最小性条件：这个属性（组）的任何真子集均不满足决定性条件。
  - 关系中包含键的属性（组）称为**超键(Superkey, i.e. superset of a key)**。
  - **e.g.** 学生（学号，姓名，性别，专业，入学年度，**ID-no**）
  - key1={学号}，key2={ **ID-no** }。而{学号，性别}、{学号，**ID-no** }不是key，而是superkey。{学号}、{ ID-no }也是superkey。

2.4 关系数据模型

- 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)。
  - 若一个关系中某个属性（组）并不是本关系的主键，但它的值引用了其他关系（或本关系）的主键的值，则称这个属性（组）为本关系的**外键**(Foreign Key, FK)。
  - 若键由所有的属性构成，则此属性组称为**全键**(all key)。
  - 包含在任何一个候选键中的属性称为主属性**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) 。

2.4 关系数据模型

- e.g.

学生(学号, 姓名, 班级)　　PK = {学号}

课程(课程号, 课程名, 学分)　　PK = {课程号}

学生选课(学号，课程号，成绩）　　PK = {学号，课程号}

FK1 = {学号}，FK2 = {课程号}。

- e.g.

部门(部门号，部门名，地点)　　PK = {部门号}

职工(职工号，姓名，工种，主管经理，所在部门号) PK = {职工号}

FK1 = {主管经理}，FK2 = {所在部门号}

2.4 关系数据模型

二、约束

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

$$r = \{t\} \subseteq D1 \times D2 \times \ldots \times Dn$$

在实现时，域Di通常用数据类型(及取值规则)来表示。

- 但是，数据是有**语义约束**的（即**完整性约束**）。
- 设r_c是R的所有满足完整性约束的元组的集合，显然应：

$$r = \{t\} \subseteq r_c \subseteq D1 \times D2 \times \ldots \times Dn$$

- 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。

2.4 关系数据模型

- 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
  - **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。
  - **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，否则只能取NULL。
  - **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。
  - 迄今为止还没有一个DBMS完全实现了所有的完整性约束，但总的趋势是朝这个方向努力。

2.4 关系数据模型

- 三、操作
  - 有两类/三种表达能力上等价的操作：
    - **关系代数操作**(Relational Algebra Operations)
      - 用关系代数来表示。包括传统的集合运算(并、交、差、笛卡尔积、外并，etc.)和关系专用操作(选择、投影、连接、除、外连接，etc.)
    - **关系演算**(Relational Calculus)
      - 用谓词演算来表示。根据变量是元组还是域进一步区分为：
      - **元组关系演算**(Tuple Relational Calculus)
      - **域关系演算**(Domain Relational Calculus)

2.4 关系数据模型

- **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，构成一个新的关系。（横向筛选）
    - $\sigma_{<选择条件>}(r) = \{t \mid t \in r \text{ AND } <选择条件>\}$
    - e.g. $\sigma_{deptno = 2 \text{ AND } job = \text{'salesman'}}(emp)$
  - **投影(Projection)**
    - 选出关系r中由<属性表>所示的诸属性列值，构成一个新的关系。（纵向筛选）
    - $\Pi_{<属性表>}(r) = \{t [<属性表>] \mid t \in r\}$
    - e.g. $\Pi_{job, sal} (emp)$

2.4 关系数据模型

- **关系代数操作—传统的集合运算**
  - **要求参与运算的关系并兼容(Union Compatibility), 即具有相同的目、且对应的属性域相同。**
  - **并(Union)**
    - $r \cup s = \{ t \mid t \in r \text{ OR } t \in s \}$
    - **e.g.** $(б_{deptno = 2 \text{ AND } job = 'salesman'}(emp)) \cup (б_{deptno = 1}(emp))$
  - **差(Difference)**
    - $r - s = \{t \mid t \in r \text{ AND } (t \notin s) \}$
    - **e.g.** $(б_{deptno = 2}(emp)) - (б_{job = 'manager'}(emp))$
  - **交(Intersection)**
    - **不是独立的操作。因为$r \cap s = r-(r-s)$**

2.4 关系数据模型

- **关系代数操作—拼接型操作**
  - **笛卡尔积(Cartesian Product)**
    - **$r \times s = \{ <t, g> \mid t \in r \text{ AND } g \in s \}$**
    - **序偶<t, g>称元组t与元组g的拼接(Concatenation)**
    - **$r \times s$的目为n_r+n_s，元组数为$|r| \times |s|$。**
    - **e.g. emp×dept**

2.4 关系数据模型

- **关系代数操作—拼接型操作**
  - **连接(Join)**：从两个关系 r 和 s 的所有元组拼接中选出满足<连接条件>者，构成一个新关系。
  - $r \bowtie_{<连接条件>} s = \sigma_{<连接条件>}(r \times s)$
  - <连接条件>的一般形式为：C_1 AND C_2 AND … AND C_k

    其中，C_i形式为: $A_i \theta B_i$, A_i, B_i分别为r, s中的属性。θ为关系运算符。
  - 故连接也称θ连接(Theta-join)。当θ为“=”时，有两类特殊连接：
    - 等连接(Equijoin)：在连接结果中保留两关系中重复的属性列。
    - 自然连接(Natural Join)：在连接结果中只保留两关系中重复属性列中之一。
  - 在实际中，连接常指自然连接。
  - e.g. emp $\bowtie_{emp.deptno = dept.deptno}$ dept
  - 另外，因$r \times s = r \bowtie_{TURE} s$，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。

2.4 关系数据模型

- **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，{**б, П,** ∪, **—,** × }或{**б, П,** ∪, **—,** ⋈ }是**关系完备操作集**。而支持关系完备操作集的DBMS称**关系完备的**(Relationally Complete)。
- 目前，大部分SQL RDBMS e.g. Oracle, Sybase, SQL Server 是关系完备的，而大部分PC数据库系统 e.g. Foxbase, FoxPro 不是关系完备的。

2.4 关系数据模型

- 关系代数操作表达式对应一颗**语法树/查询树**

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	salesmam	3500	2

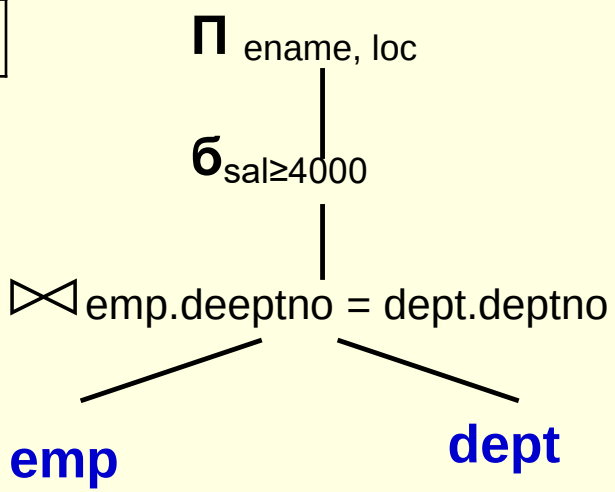
关系 dept:

deptno	dname	Loc
1	财务部	Shanghai
2	市场部	Nanjing

结果：

ename	Loc
张三	Shanghai
李四	Nanjing

语法查询树：



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

目录 Contents

2.5 对传统数据模型的评价

- 一、传统数据模型
  - **传统数据模型**(Traditional Data Models)
    - 层次、网状、关系模型。
  - **非传统数据模型 / 后关系模型**(Post Relational Data Models)
    - ER (Entity/Relationship)、O-O (Object-Oriented)、基于逻辑的(Logic)模型。

2.5 对传统数据模型的评价

- 二、评价
  - **肯定之处**
    - 向用户提供了统一的数据模型。
    - 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。
    - 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，数据管理技术不能不说是一个飞跃。尤其是关系数据模型(RDBMS),在量大面广的联机事务处理(OLTP)系统中基本上能满足应用的需求。

目录 Contents

2.6 E-R数据模型

- 1976年，Peter Pin-Shan Chen在论文：**The entity-relationship model—toward a unified view of data,** in: *ACM Transactions on Database Systems (TODS)*, Volume 1, Issue 1, March 1976, Pages: 9–36. 中首先提出ER模型时有三个目的，现主要作为数据建模（即**DB概念设计**）的有力工具。Peter Pin-Shan Chen模型称基本ER模型，后来有许多扩充版本，称**扩充ER模型**(Extended ER Data Model, EER)。
- 观点
  - 世界是由一组称作实体的基本对象和这些对象之间的联系构成的。

2.6 E-R数据模型

一、基本ER模型

三种抽象

- **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，是对某个具体事物的一种描述。

    e.g. 实体集S = {e | e是学生}；而张三,李四,王五,…∈S,是一个个学生。
  - 实体键&实体主键：同关系模型中的概念。

    e.g. “学号”是“学生”实体的实体键 / 实体主键。
  - 弱实体：不能独立存在，依附与其他实体集中的某个实体（称所有者实体）。其实体键必须包含其所有者实体的键。

    e.g. “职工”与“家属”，“电影厂”与“摄制组”。

2.6 E-R数据模型

- **属性(Attribute)一对事物（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，专业，…是学生的属性。
  - 非原子属性：
    - 组合属性/元组属性

      e.g. 通讯地址：（邮编，街区）

      街区：（街道名，号码，住宅号）
    - 多值属性/集合属性

      e.g. 选修课程：{C语言，DB，汇编}

2.6 E-R数据模型

- **联系**(Relationship)—对事物之间某种关系的一种抽象。
  - 如学生与老师间的授课关系，学生与学生间有班长关系。
  - 联系也可以有属性，如学生与课程之间有选课联系，每个选课联系都有一个成绩作为其属性。
  - **联系集**：事物之间同类联系所组成的一个集合。
  - **联系**：联系集中的一个实例。

    e.g.

    联系集MARR(M,F)= {<e1,e2>|e1∈M∩e2∈F∩e1与e2是夫妻}；而<张三, 李梅>, <王五, 赵丽>, …,∈MARR是一对对夫妻。

2.6 E-R数据模型

- **联系**(Relationship)
  - **联系的语义约束**—结构约束(Structural Constraints)
    - **基数比**(Cardinality Ratio)

对联系R（E1,E2,…,En），

当n = 2时，二元联系(Binary Relationship)：1:1, 1:N, M:N

当n > 2时，多元联系(Multiway Relationship)

e.g.三元联系：1：1：1，1：1：P，M：N：P, etc.

当n = 1时, 自联系/递归联系(Recursive Relationship):1:1, 1:N, M:N

- **参与度**(Participation)

对联系R（E1，E2，…，En）中的某个Ei，

若 ei ∈Ei均参与联系R，则称实体集Ei是全参与的(Total);

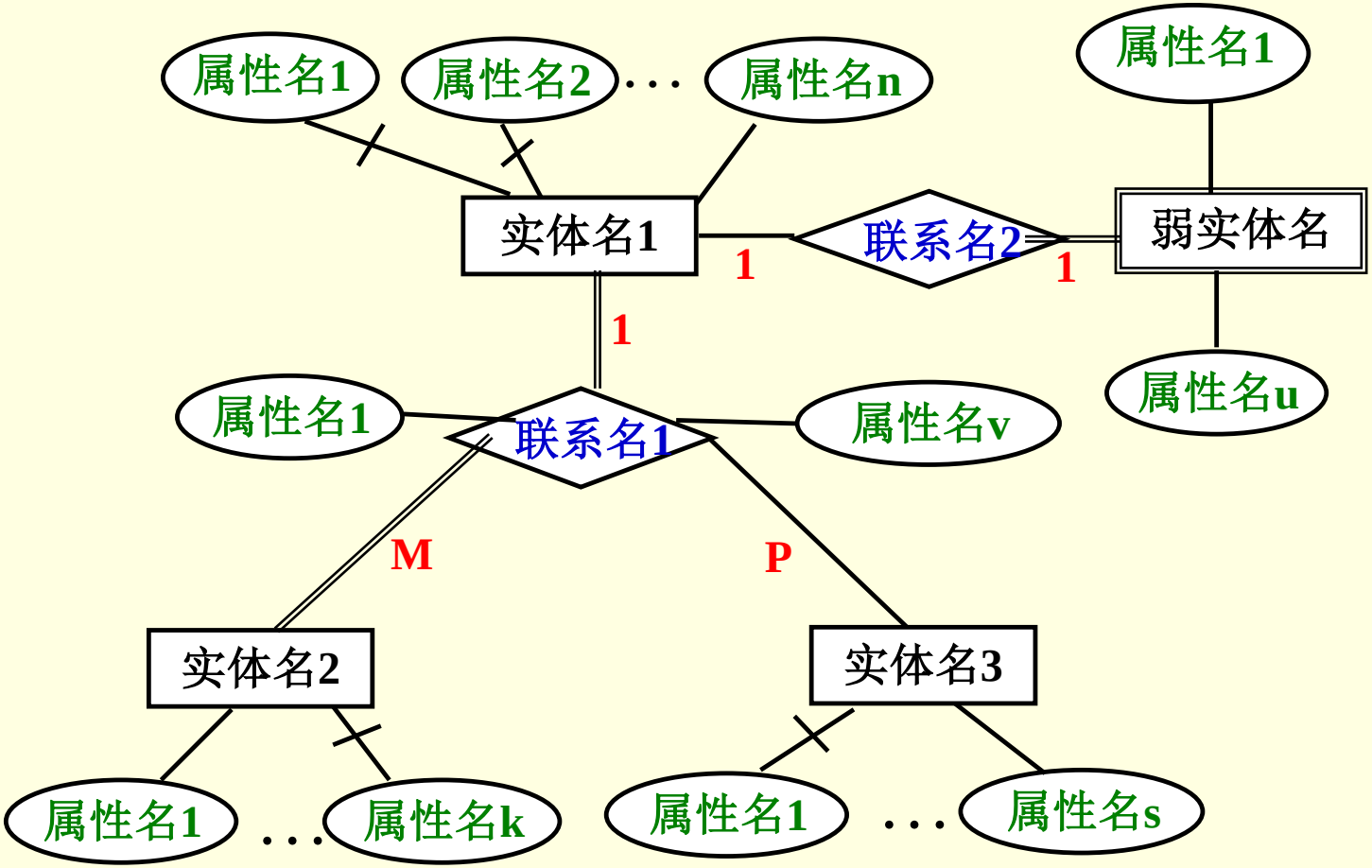
若 ei ∈Ei不参与联系R，则称实体集Ei是部分参与的(Partial)。

2.6 E-R数据模型

- 一、基本ER模型
  - ER数据模式与ER图
    - 运用ER模型对一个企业的全体数据进行建摸得出的结果程ER数据模式，简称ER模式(ER Schema)，其常用ER图(ER Diagram)的形式表示。

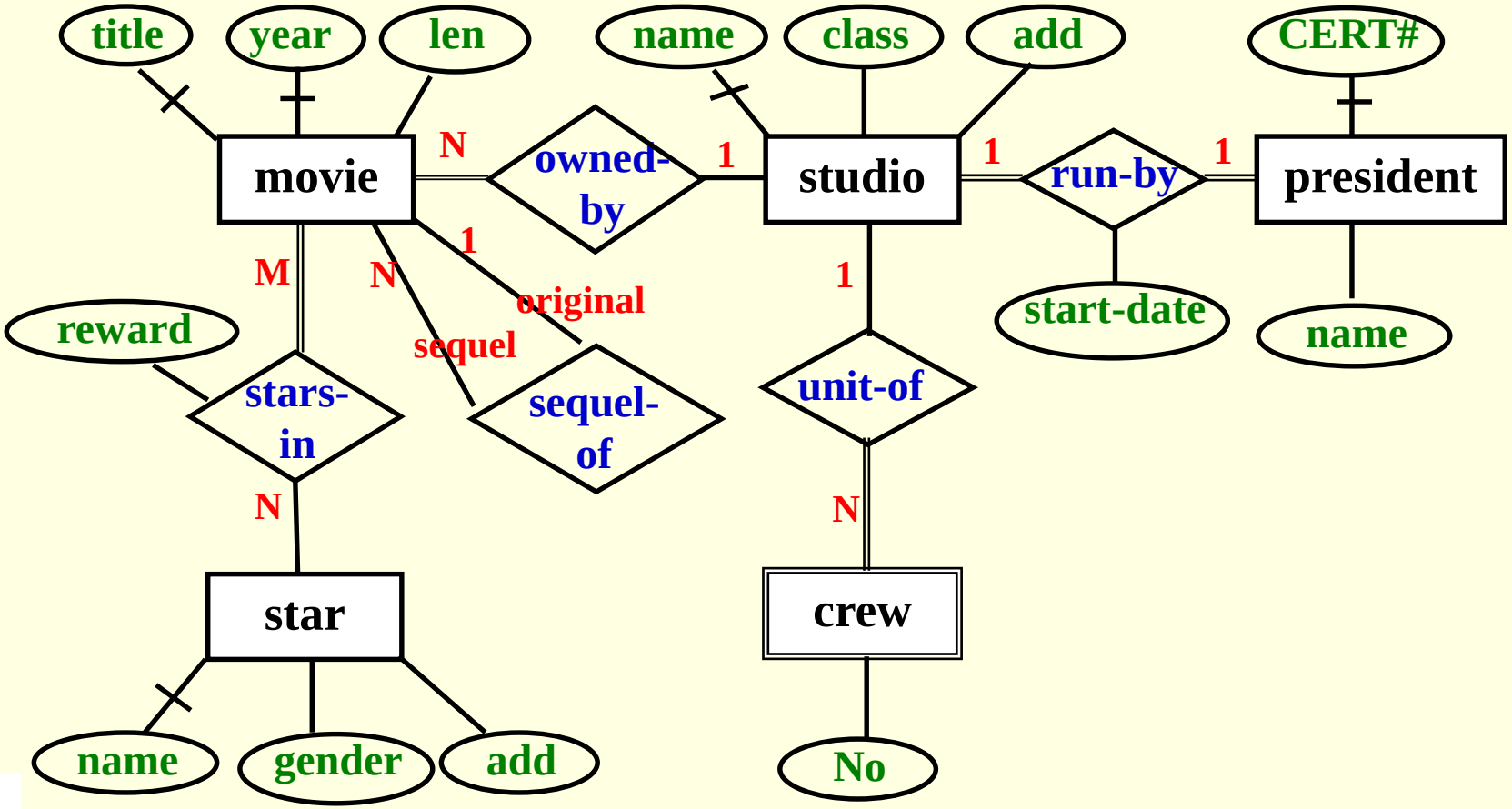
2.6 E-R数据模型

- ER图的符号(Notation)



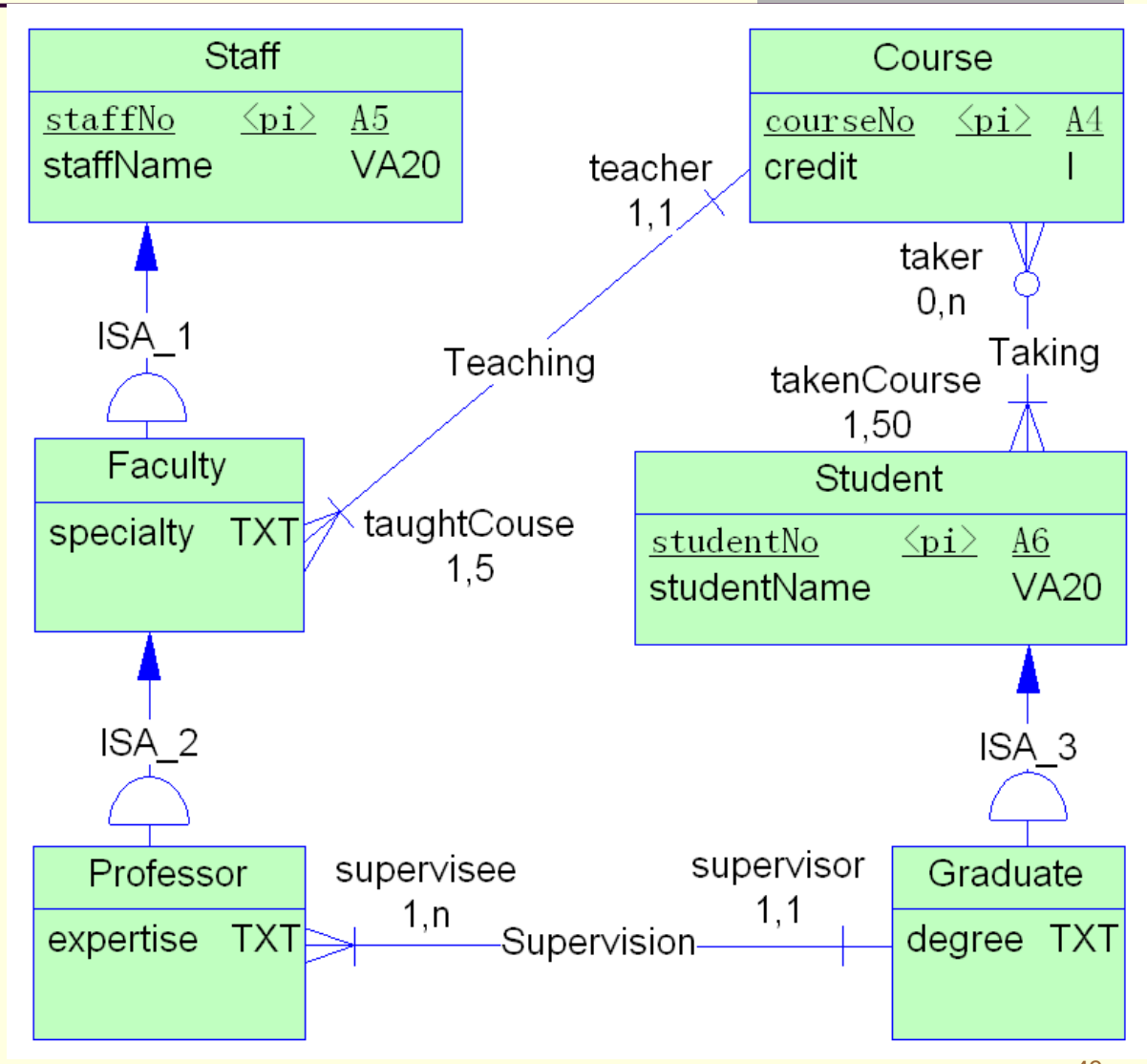
2.6 E-R数据模型

- ER图例子：从Stanford Univ.’s textbook中拷贝的ER图



2.6 E-R数据模型

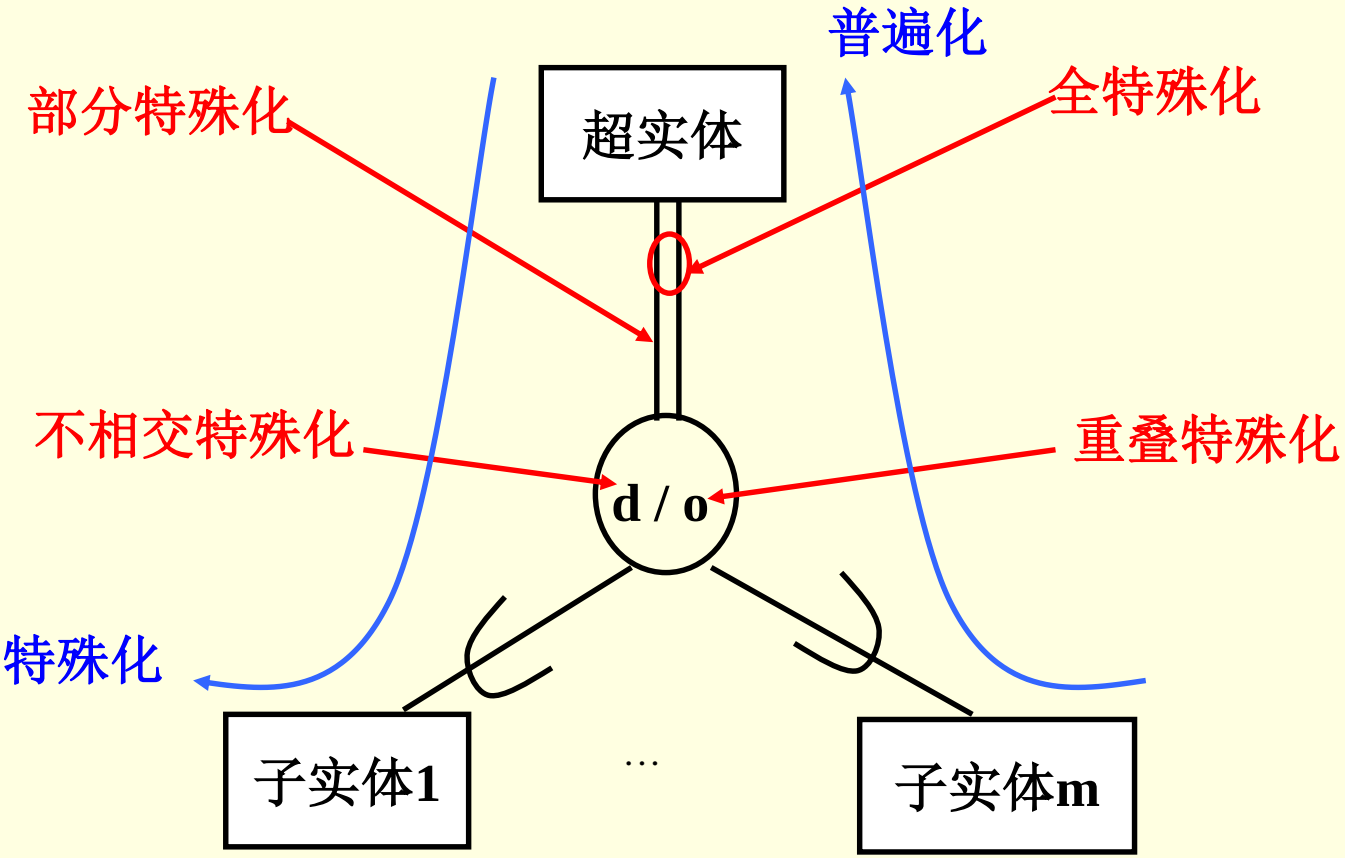
- ER图例子：用ER Modeling tool 创建的ER图
- e.g., Sybase's PowerDesigner 9.5



2.6 E-R数据模型

二、扩充ER模型

特殊化(Specialization)与普遍化(Generalization)

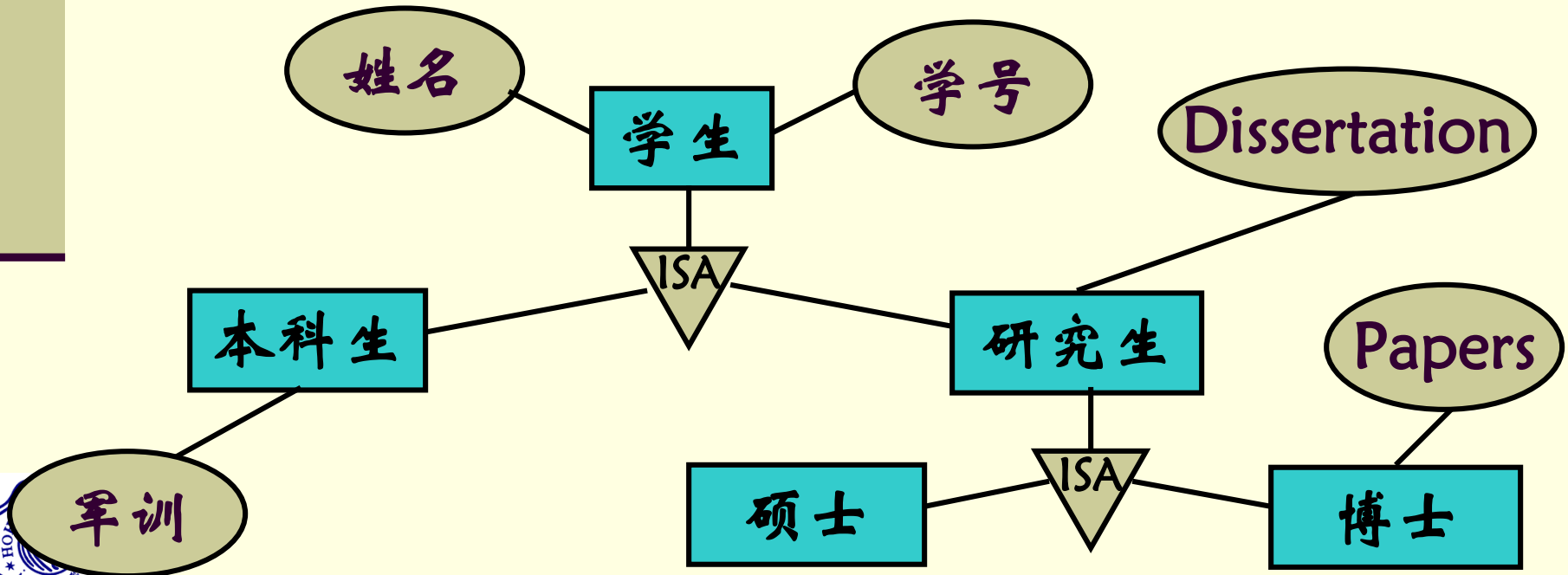


特殊化

- 自顶向下、逐步求精的数据库设计过程
  - 实体集中某些子集具有区别于该实体集内其它实体的特性，可以根据这些差异特性对实体集进行分组，这一分组的过程称作特殊化
- 父类 → 子类
- 子类＝特例＝更小的实体集＝更多的属性
  - 一个银行帐号可以有存款帐号、贷款帐号
  - 学生可以有研究生、本科生

特殊化

- 特殊化在E-R图中的表示
  - 特殊化用标记为ISA的图形来表示
  - ISA = “is a”，表示高层实体和低层实体之间的“父类－子类”联系



普遍化

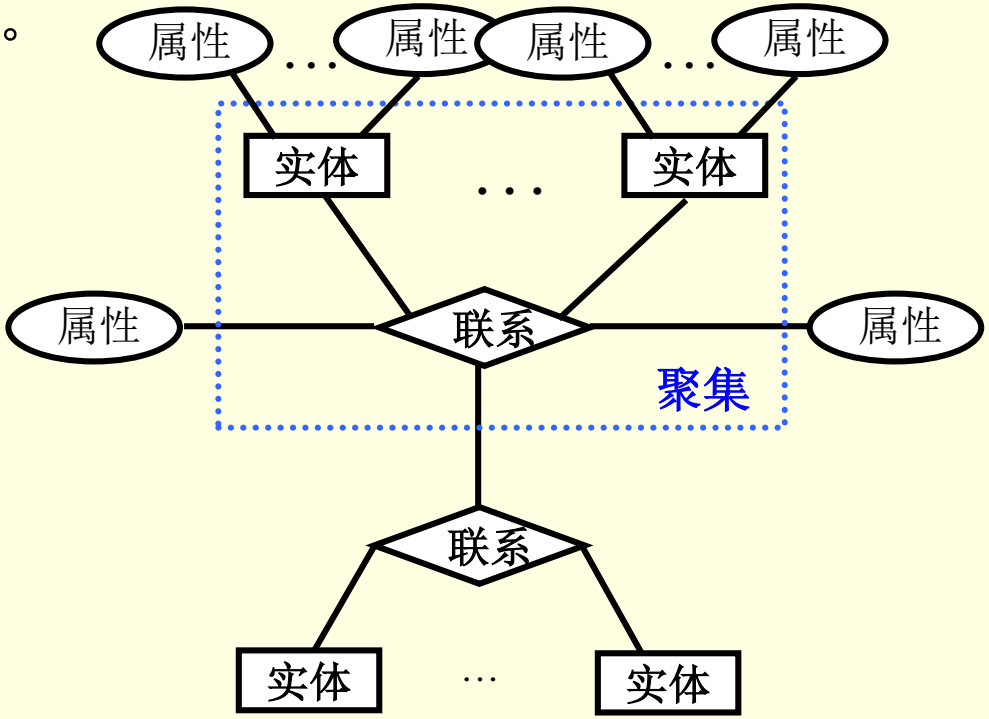
- 自底向上、逐步合成的数据库设计过程
  - 各个实体集根据共有的性质，合成一个较高层的实体集。
  - 普遍化是一个高层实体集与若干个低层实体集之间的包含关系
- 普遍化 Vs 特殊化
  - 普遍化与特殊化是个互逆的过程，在E-R图中的表示方法是相同的。
  - 特殊化强调同一实体集内不同实体之间的差异，普遍化强调不同实体集之间的相似性
  - 反映了数据库设计的不同方法

2.6 E-R数据模型

二、扩充ER模型

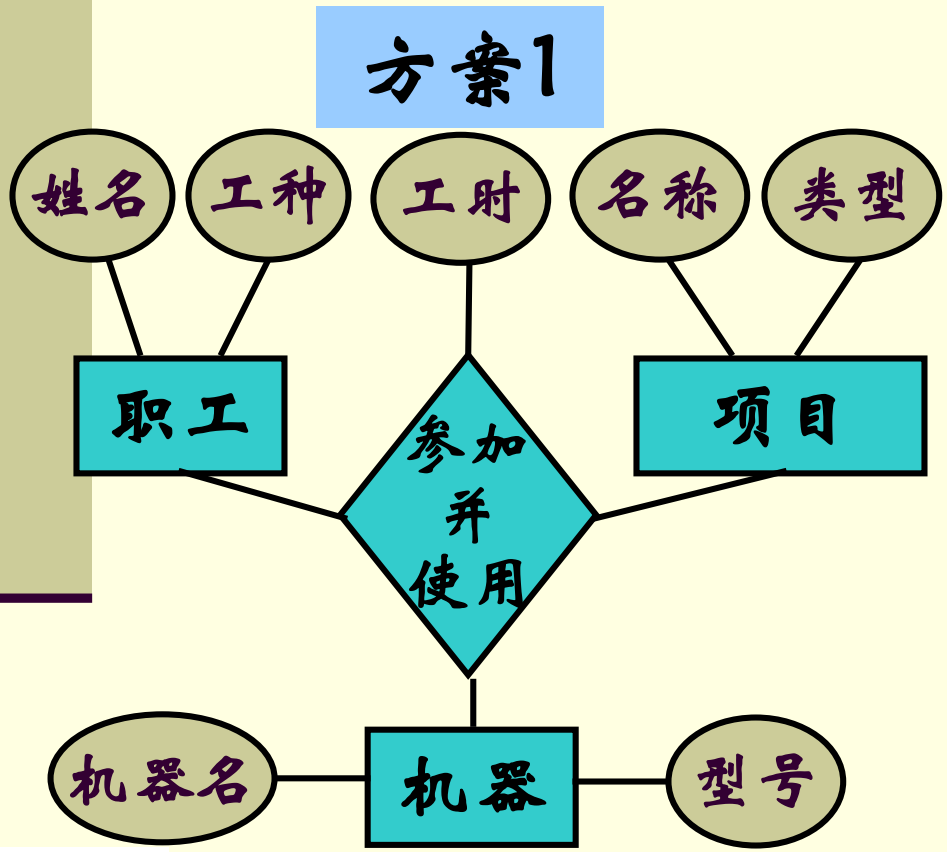
聚集(Aggregation)

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



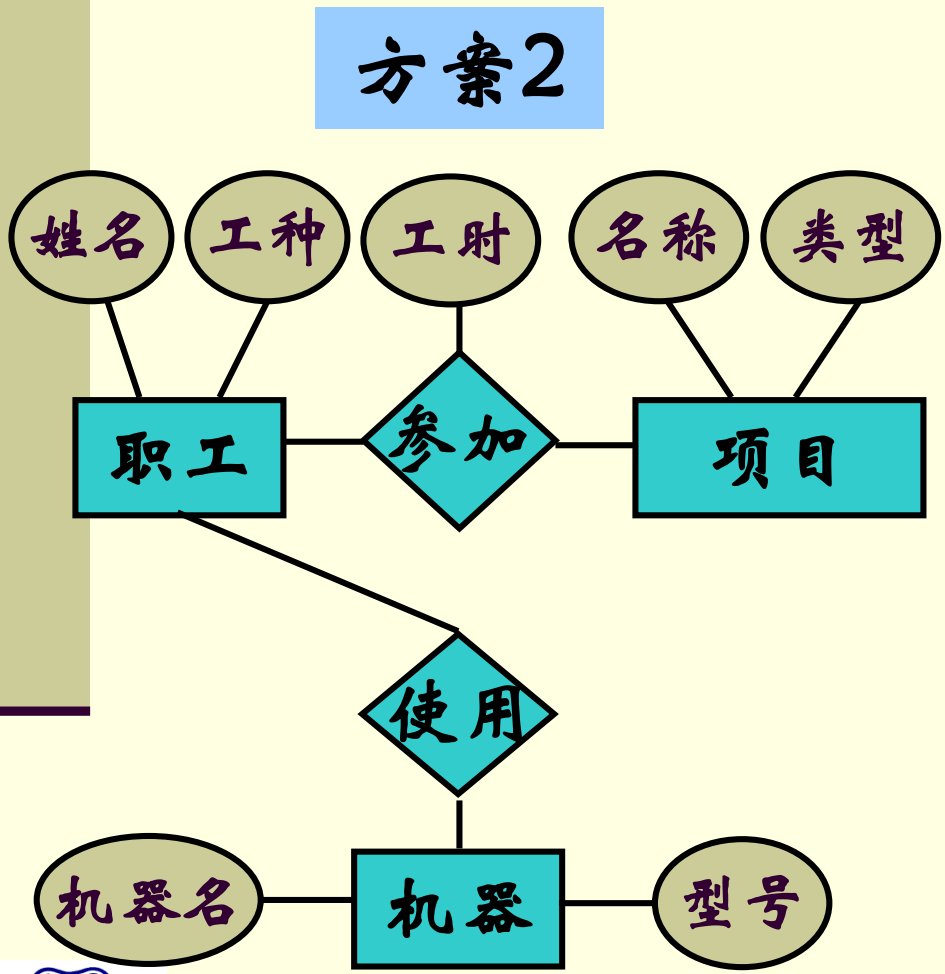
聚集

- 实例：职工参加项目，并在此过程中可能使用机器



职工号	项目号	机器号	工时
e1	j1	m1	3
e1	j1	m2	3
e1	j2	m3	4
e2	j1	m1	5
e3	j2	null	4
e4	j2	null	4
e5	j2	null	6
e6	j2	null	5

聚集



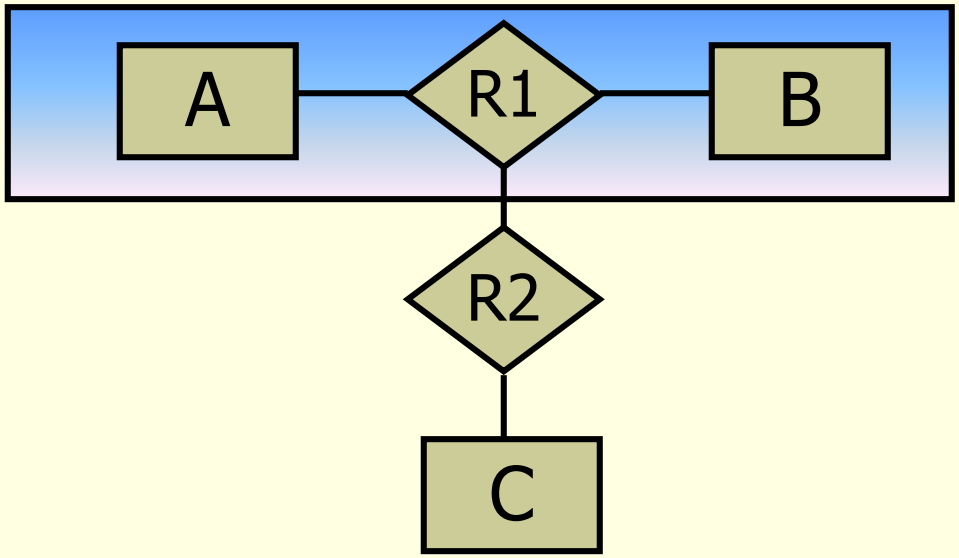
职工号	项目号	工时
e1	j1	3
e1	j2	4
e2	j1	5
e3	j2	4
e4	j2	4
e5	j2	6
e6	j2	5

职工号	机器号
e1	m1
e1	m2
e1	m3
e2	m1

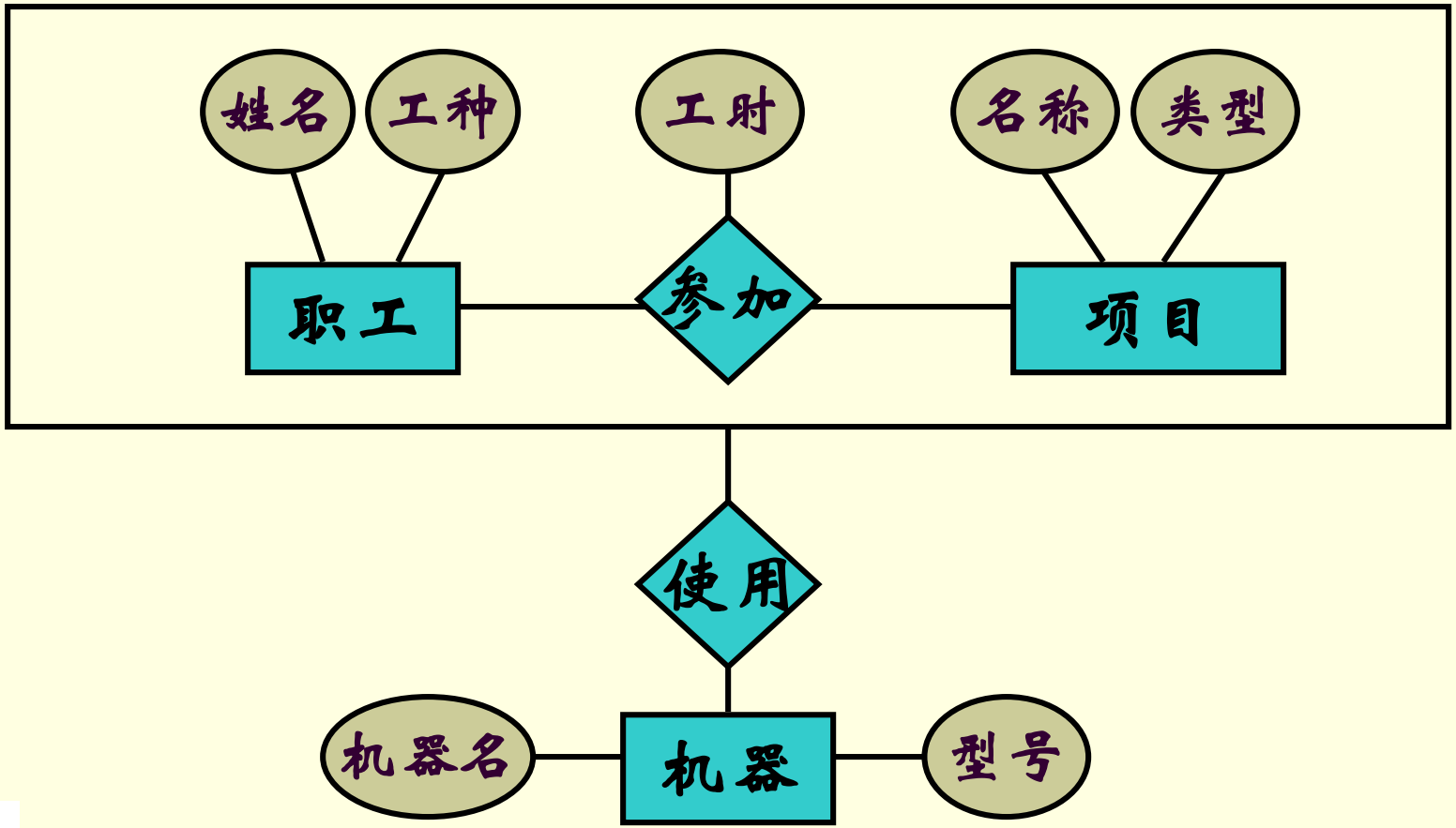
e1在j1项目中使用什么机器?

聚集

- 聚集是一种抽象，通过它联系被作为高层实体集
- 实体集A与B以及它们的联系可与另一实体集C发生联系



聚集



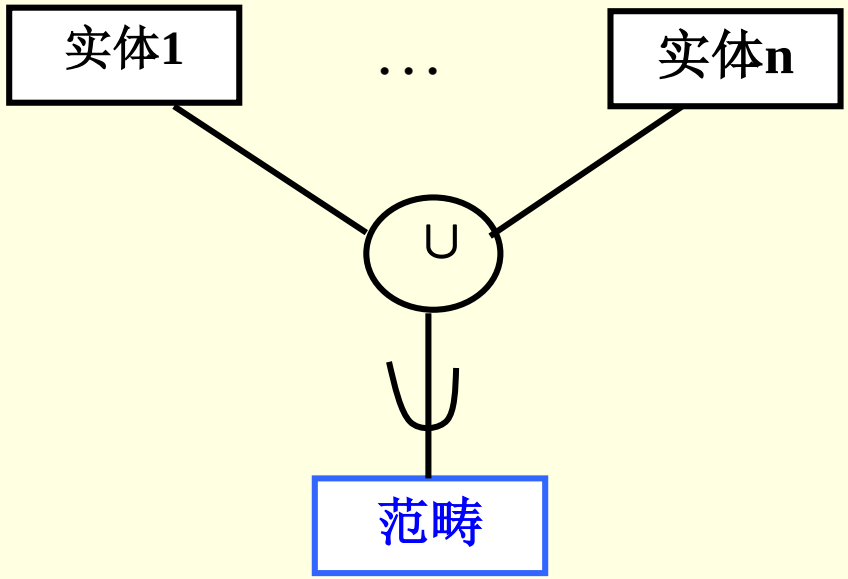
聚集

职工号	项目号	工时
e1	j1	3
e1	j2	4
e2	j1	5
e3	j2	4
e4	j2	4
e5	j2	6
e6	j2	5

职工号	项目号	机器号
e1	j1	m1
e1	j1	m2
e1	j2	m3
e2	j1	m1

2.6 E-R数据模型

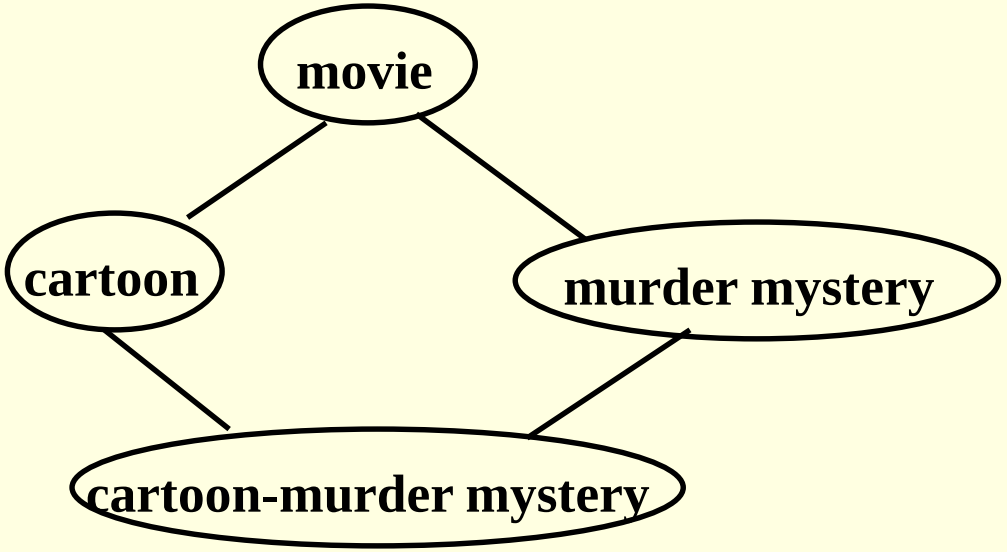
- 二、扩充ER模型
  - **范畴(Category)**
    - **不同类型的实体组成新实体，称范畴。**



目录 Contents

2.7 面向对象数据模型

- 一、基本概念
  - **对象**：世界由对象所组成。对象由OID标识。
  - **类**：具有相似特性(Properties)(属性、联系、方法)的对象组成类。
  - **子类和超类**：类的子集可定义为子类，原类为超类。通过继承（Inheritance）可形成**类层次结构**。



2.7 面向对象数据模型

- 一、基本概念
  - Object-Oriented Data Model
  - Object-Oriented Database Management System (OODBMS)
  - Object Definition Language (ODL)
  - Object Query Language (OQL)

2.7 面向对象数据模型

二、参考文献

http://www.odmg.org

Object Data Management Group
The Standard for Storing Objects

- **ODMG status:**
- The ODMG group completed its work on object data management standards in 2001 and was disbanded. The final release of the ODMG standard can be found here (The ODMG 3.0 specification) including the overall data model, and the Smalltalk and C++ bindings for ODMG. The ODMG Java binding has been superceded by Java Data Objects (JDO: http://java.sun.com/products/jdo/).

2.7 面向对象数据模型

- 二、参考文献

OBJECT MANAGEMENT GROUP

http://www.omg.org

- The OMG is currently working on a "4th generation" object database standard to reflect recent changes in object database technology. More information on the OMG work can be found at http://www.odbms.org/about_news_20060218.html.