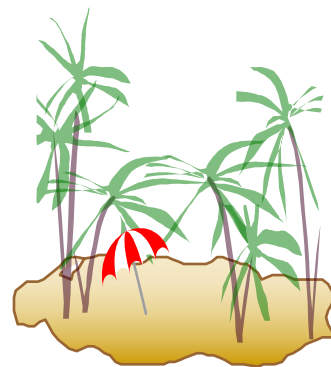
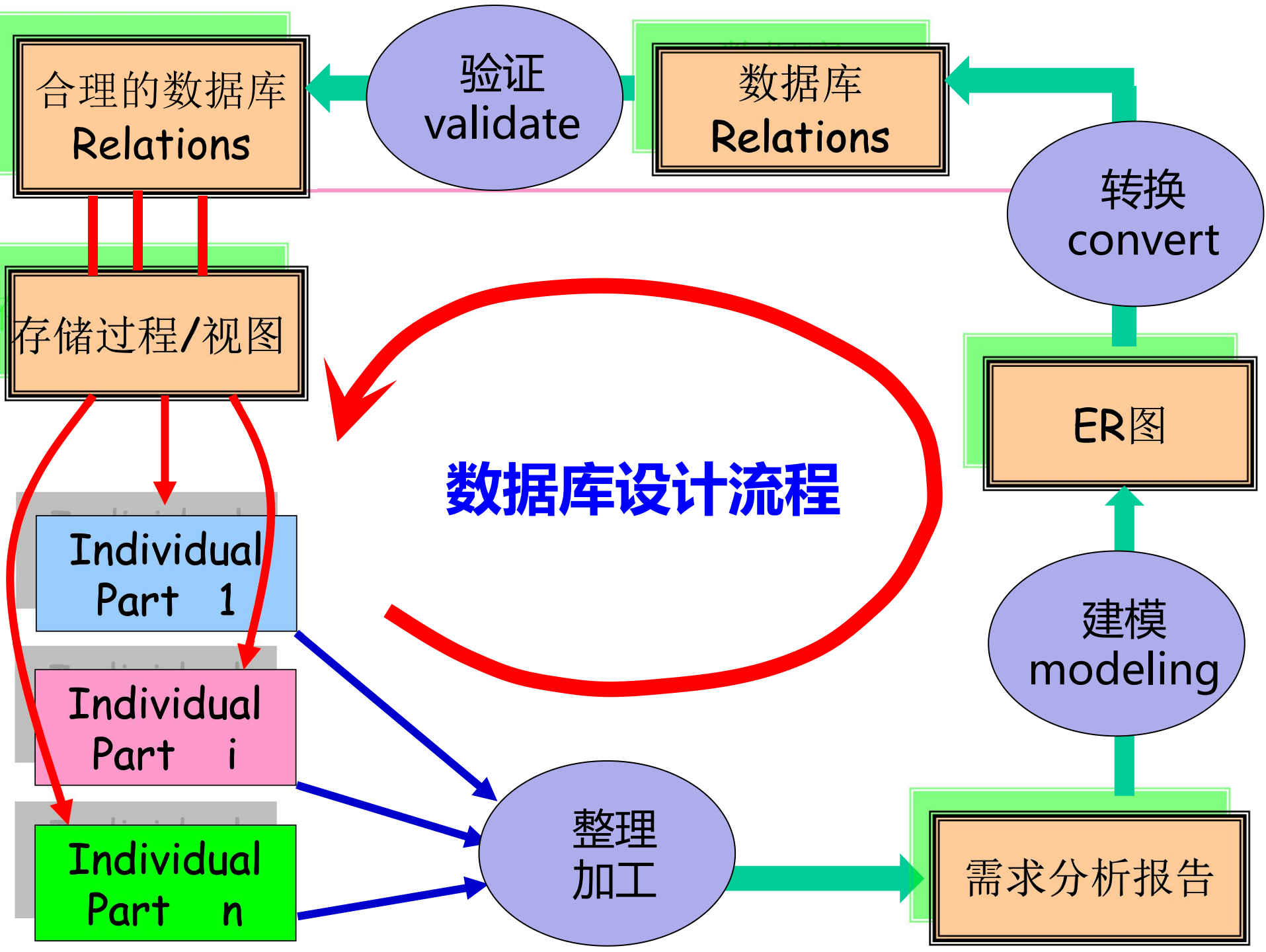


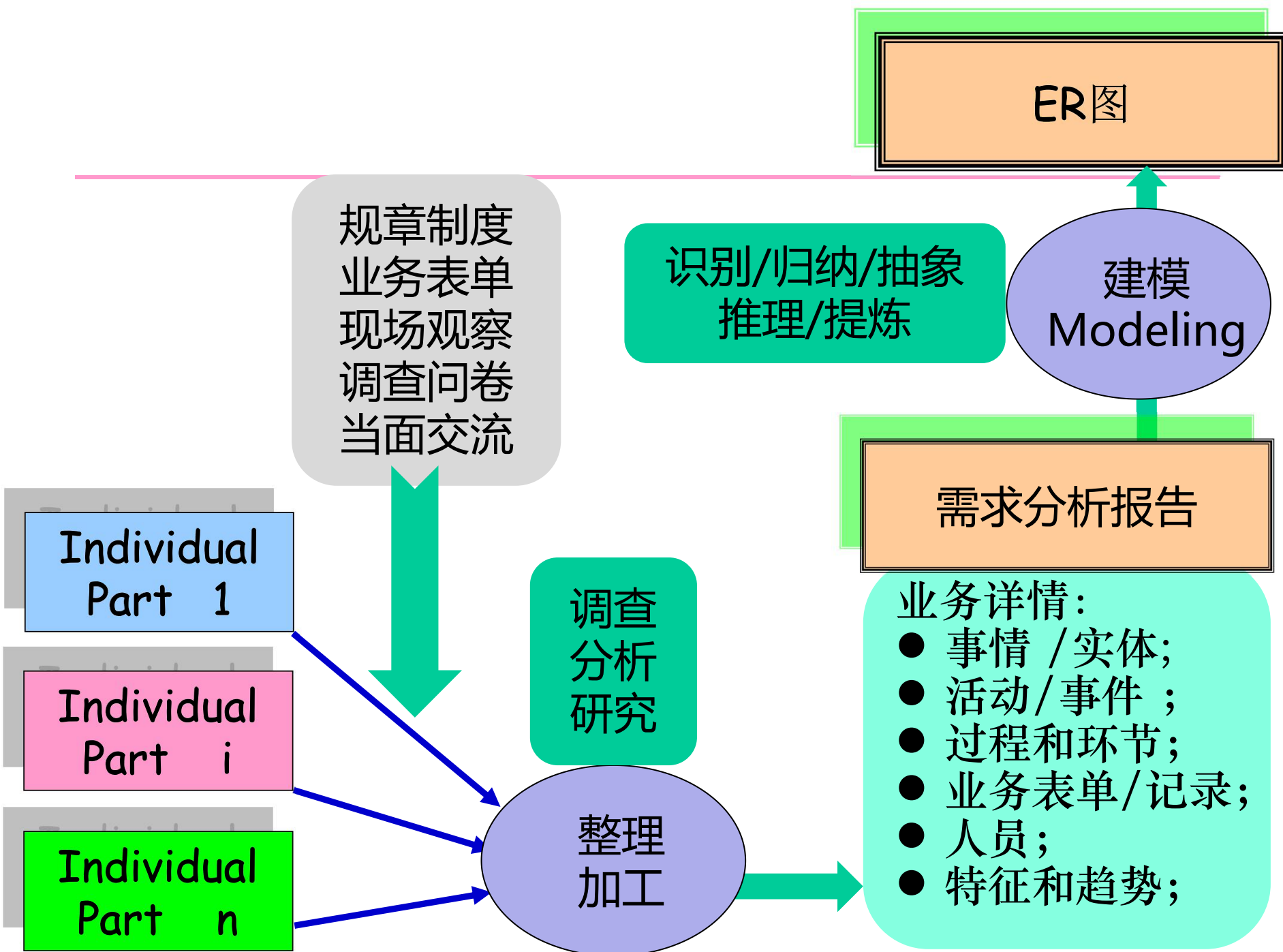
数据库设计

section 2 实体-关系建模

- 信息科学与工程学院
 - 杨金民
- 2017.12







数据库设计目标

- **满足业务需求**：全覆盖；需求获取**全面**，**分析到位**；
- **数据正确性**：尽量少的**冗余**，无操作**异常**，**一致**。设计正确。
- **可伸缩性**：只要数据库中存在数据根子，任何业务需求都能满足，想要的业务表单都能组合/合成出来；

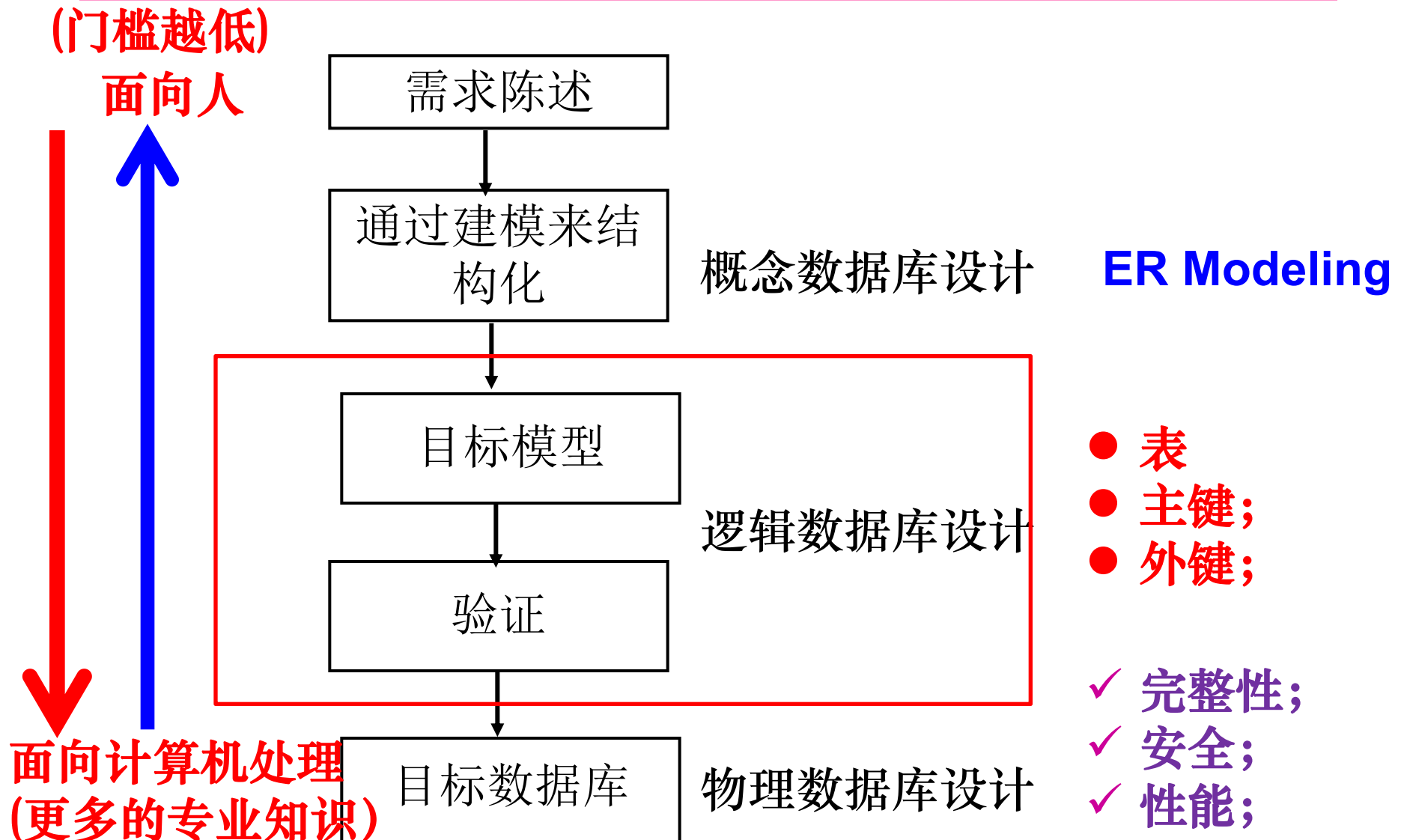
涉及的三个问题：

- 1) 覆盖性问题：确定一个单位有哪些数据项（字段）？
- 2) 划分问题：确定哪些字段构成一个表？
- 3) 关系问题：确定表之间有什么关系？

需求分析

- **需求分析报告:** 以文本或者图形方式来陈述企业的业务活动, 以及其中的处理流程, 环节, 记录, 流转特性, 诉求和期望。
- **优点:** 门槛低, 易于理解, 方便交流。在认识和把握企业特性, 业务, 流程, 记录, 数据等方面易于沟通交流。以求全面, 准确的认识和理解企业, 以及。
- **不足:** 在结构和关系上模糊, 不精准, 理解和认识上易于出现偏差, 让计算机处理难;
- 例如: 1) 6除2; 2) 中国队大胜美国队, 中国队大败美国队;

数据库设计的过程



建模(modeling)

对一个企业的数据进行结构化处理，准确地表达数据的内涵；

基本要素：

- 概念及其定义；
- 表达概念所用的符号；
- 有关处理的规则；

例如： $(6 \div 2 + 3) \times 4$ ；



需求分析报告案例

湖南大学本科教学的业务情况

每个学期的教学过程是：老师首先提出开课申请，一个老师可以开一门或者多门课。一门课程也允许有多个老师同时开；

然后学生对老师开出的课进行选课，学生信息有学号，姓名，性别，班级，所在学院；

一个学生一个学期可以自由决定是否选课，也可选多门课。修完一门课后得到一个考试分数，如果考试不及格，可以以后重选、重修。





典型的教学管理业务表单(1)

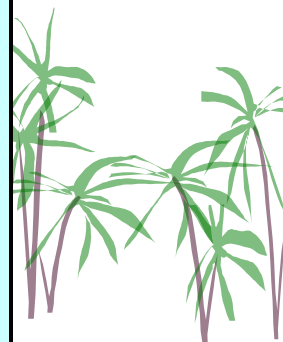
学生名册

学院名称：软件学院 学院编号：01

地址：湖南大学软件大楼

姓名	学号	性别	出生日期
周山	2008043101	男	1990/12/14
汪兵	2008043214	男	1992/02/21
张珊	2008043332	女	1988/07/09
刘丽	2008043315	女	1988/01/29

院长：林亚平





典型的教学管理业务表单(2)

课程成绩单

课程名称：数据库系统与设计 课程编号：001
学 期：2010/01

姓名	学号	成绩
周山	2008043101	78
汪兵	2008043214	65
张珊	2008043332	81
刘丽	2008043315	92

上课老师：张三





典型的教学管理业务表单(3)

上课老师名单汇总表

学院名称：软件学院

学 期：2010/01

教师姓名	课程名称	选课学生人数
周波	数据库系统	54
杨柳	数据库系统	66
张景	操作系统	75
马征	面向对象设计	92



数据库设计的过程

三步曲：

为什么要三步，而不是一步？

概念数据库设计；从业务需求采用高级建模方法建模；



逻辑数据库设计；将高级模型转换为关系模型；



物理数据库设计；采用特定的DBMS产品实现数据库；

概念数据库设计(ER modeling)

非结构化数据 演变 带语义的结构化数据

高级语言(规范)
常用
易于理解和沟通
独立于任何数据模型

基本概念和要素:

实体(Entities) 抽象 实体类(Entity type)

关系(Relationships) 抽象 关系类(Relationship type)
属性(Attributes)

企业数据库实例

路桥工程公司，承接路桥工程业务，公司有多个工程部门，每个工程部门，有一些员工，公司承接的工程项目，交由某个工程部门负责承建完成，要求使用计算机对工程项目，人员，部门进行管理，为结算，绩效考核，公司状况，发展提供帮助。

表（实体类型及其实例）

Employee Relation

ENo	ENAME	BDate	Title	Salary	SuperNo	DNo
E1	J. Doe	01-05-75	EE	30000	E2	null
E2	M. Smith	06-04-66	SA	50000	E5	D3
E3	A. Lee	07-05-66	ME	40000	E7	D2
E4	J. Miller	09-01-50	PR	20000	E6	D3
E5	B. Casey	12-25-71	SA	50000	E8	D3
E6	L. Chu	11-30-65	EE	30000	E7	D2
E7	R. Davis	09-08-77	ME	40000	E8	D1
E8	J. Jones	10-11-72	SA	50000	null	D1

表（实体类型及其实例）（2）

Department Relation

DNo	DName	MgrNo
D1	Management	E8
D2	Consulting	E7
D3	Accounting	E5
D4	Development	null

Project Relation

PNo	PName	Budget
P1	Instruments	150000
P2	DB Develop	135000
P3	Budget	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

表（关系类型及其实例）（3）

WorksOn Relation

ENo	PNo	Resp	Duration
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36

实体-关系建模(E-R modelling)

❖ 概念(concepts):

- ❖ 实体Entity;
- ❖ 关系Relationship;
 - 属性Attribute;

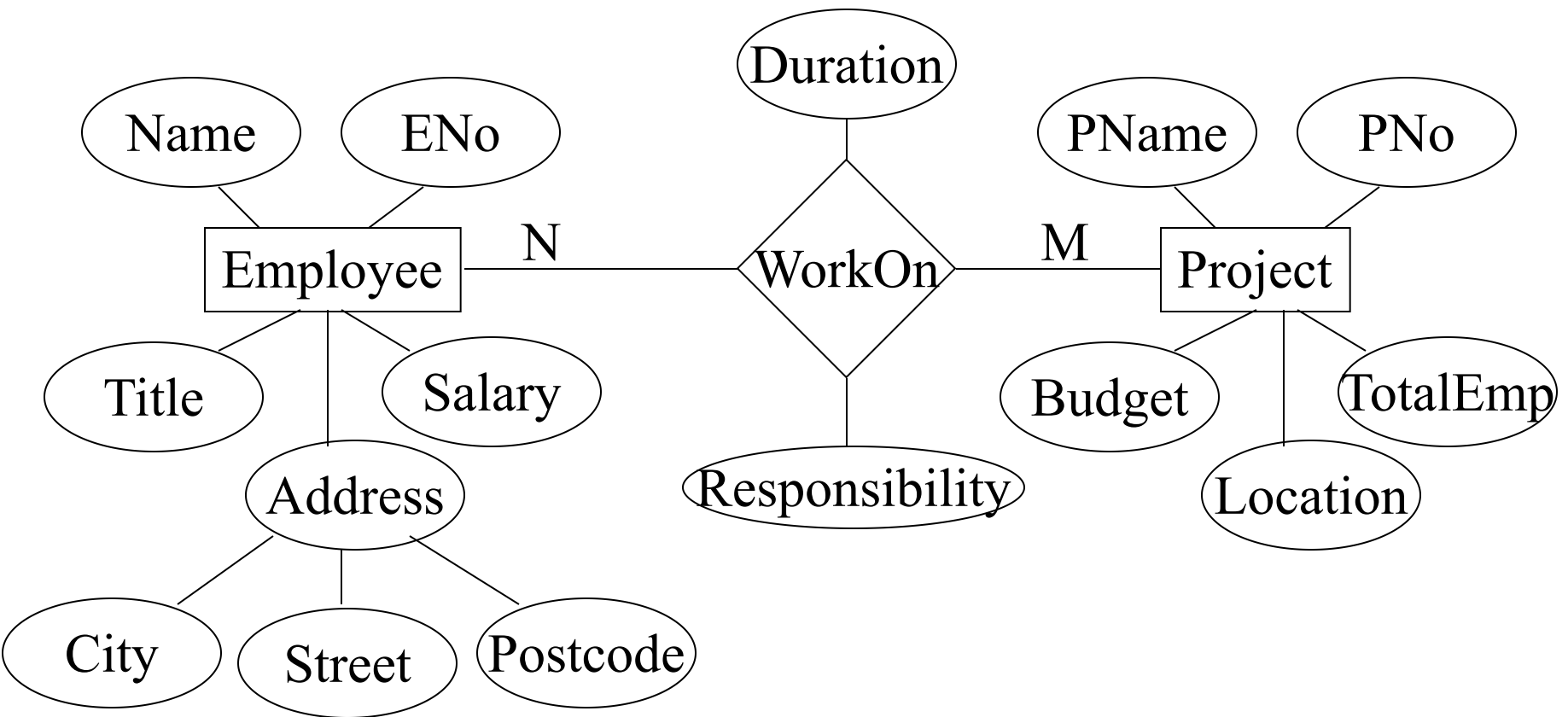
❖ 约束(Constraints):

- ❖ 作用于实体上的约束:
 - 实体标识(Entity identification);
- ❖ 作用于关系上的约束:
 - 度(degree),
 - 参(participation);
 - 基(Cardinality);

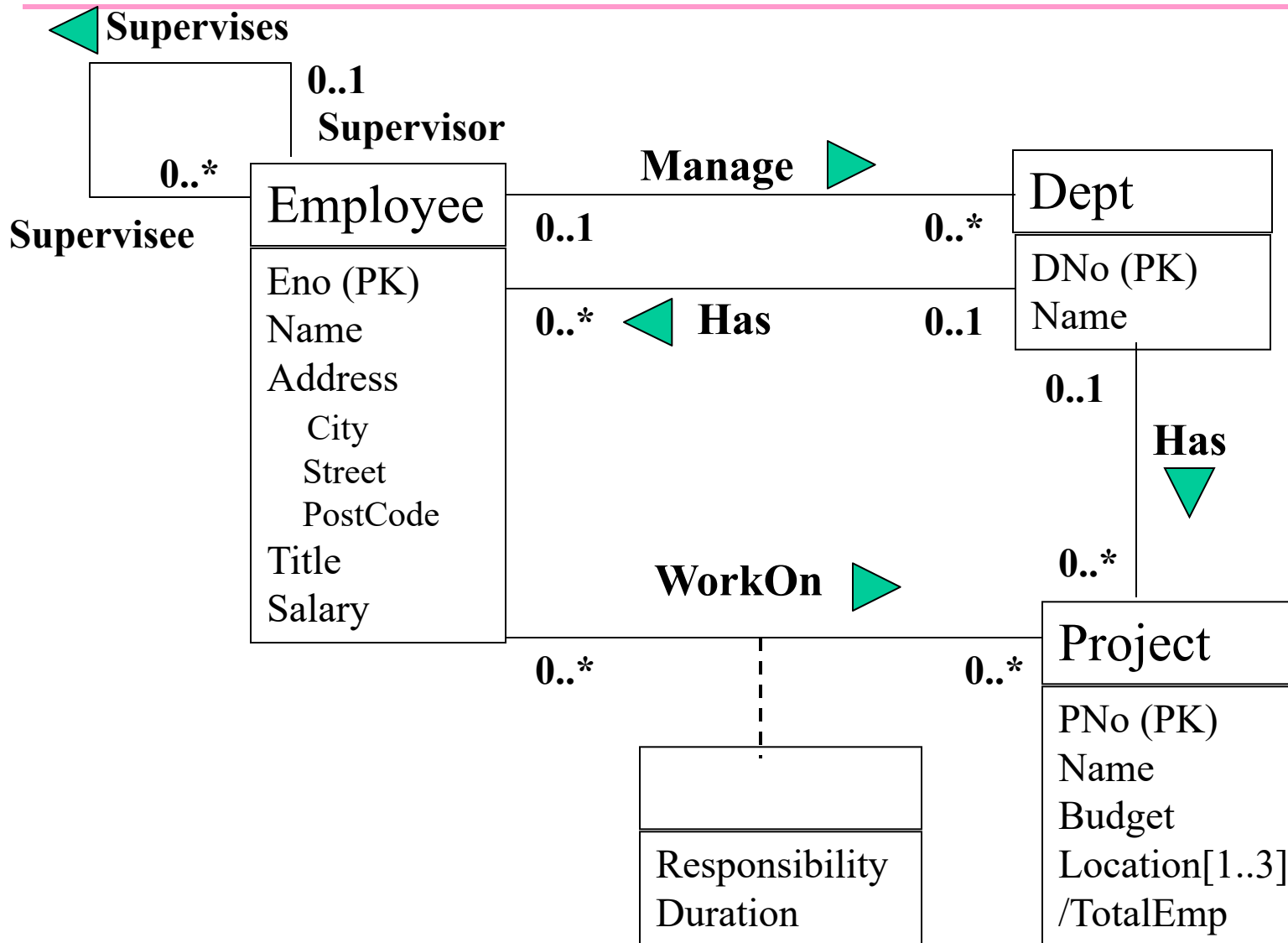
❖ 符号(Symbol);

1976年由Peter chen 提出.

ER Model 例子 (旧标记)



ER Model 例子 (UML 标记)



实体类型(Entity Types)

- ❖ 通过归纳和抽象得出的概念。对应于面向对象编程中的class
- 注意：
 - 实体可以是现实世界的可见实体，例如：职员，学院；也可以是抽象概念，例如：项目(project)，会议。
- ❖ 实体实例(entity instance) 是实体类型(entity type)的一个实例对象。
 - 例如，实体类型：员工(Employee)，
 - 实体实例： 'E1 – 杨金民'。
- ❖ 实体实例可构成一个集合；

实体类型的符号表示

- ❖ 实体类型用长方形表示，其名称放在长方形中.其名称用单数名称，第一个字母大写;
- ❖ 旧规范和XML规范一样

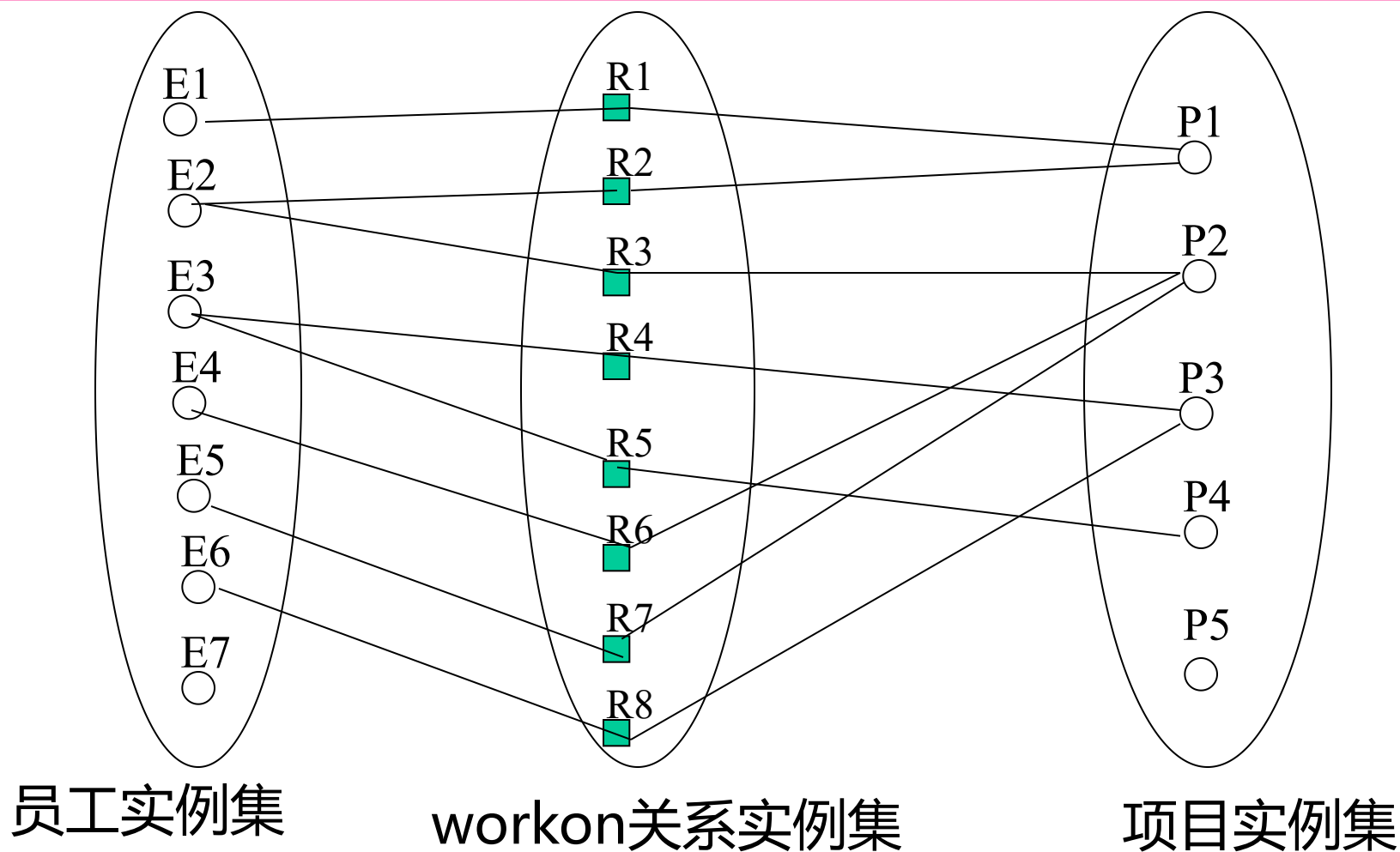
Employee

Project

关系类型

- ❖ 实体类型之间的联系——关系类型(**relationship type**).
- ❖ 例如, **WorksOn** 是员工类型(Employee)和项目类型(Project)之间的一种关系。而**(E1,P1): 'E1' works on project 'P1'** 是该关系类型的一个实例.
- ❖ 在两个实体类型之间可能存在多个关系类型。

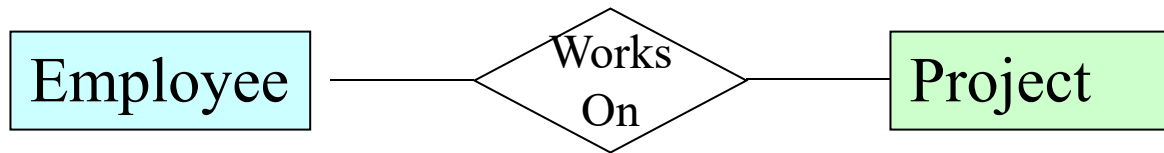
WorkOn实例



多对多关系

关系类型的标记

- ❖ 旧标记中，菱形表示关系类型。



- ❖ In UML, 两个实体类型之间的一条边线表示一个关系类型，边上标签为关系类型的名称，附加上一个箭头来表达关系的逻辑。

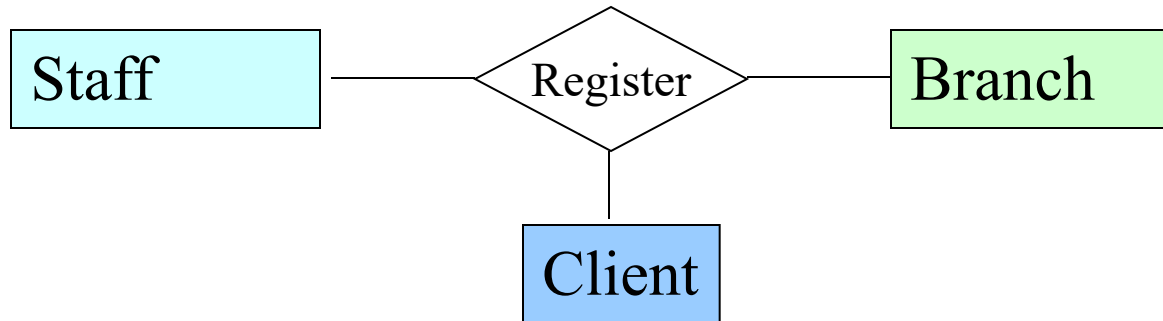


- ❖ 关系的名称通常为动词或者动词短语，第一个字母大写

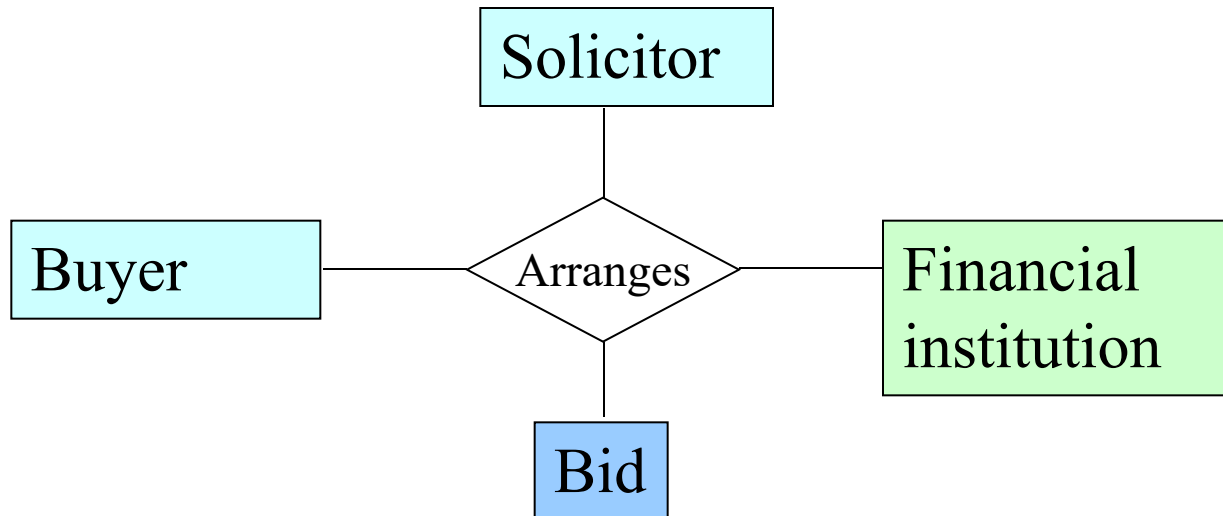
关系的度(Degree)

- ❖ 一个关系类型涉及的实体类型的数量.
 - 例如, WorksOn 关系的度是2, 它涉及2个实体类型: Employee, Project.
- ❖ 度为2的关系叫二元(binary)关系, 度为3的关系叫三元(ternary)关系,, 度为N的关系叫N元(n-ary)关系.
- ❖ 对于度 ≥ 3 的关系类型在UML标记中, 也是用菱形来表示关系类型.

关系类型的度



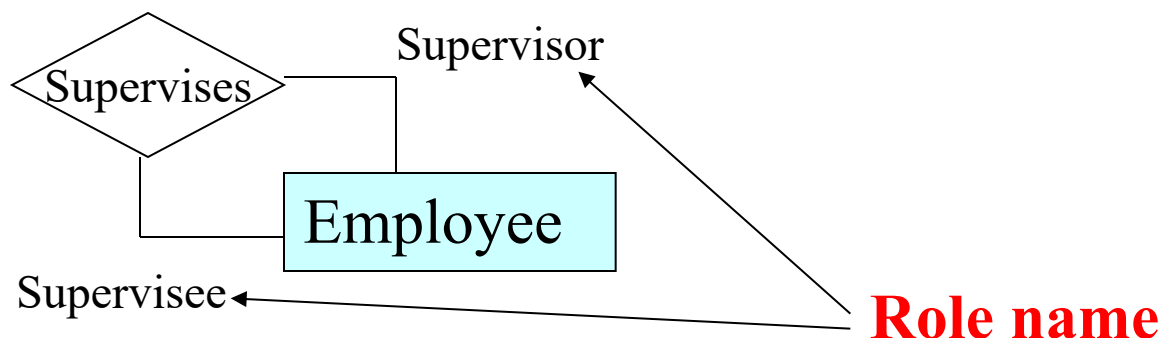
Registration of a client by a staff at a branch



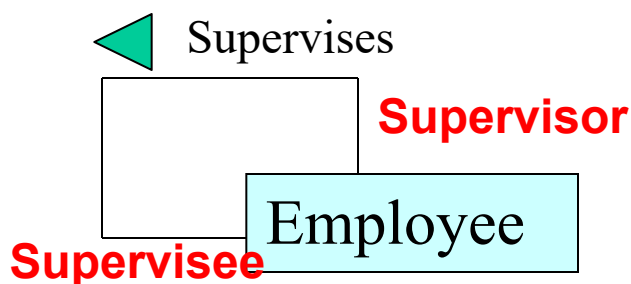
递归关系(Recursive Relationships)

一个**实体类型**可以充当不同的**角色**，同一个实体类型的**不同角色**之间可能存在关系。例如：

旧标记：



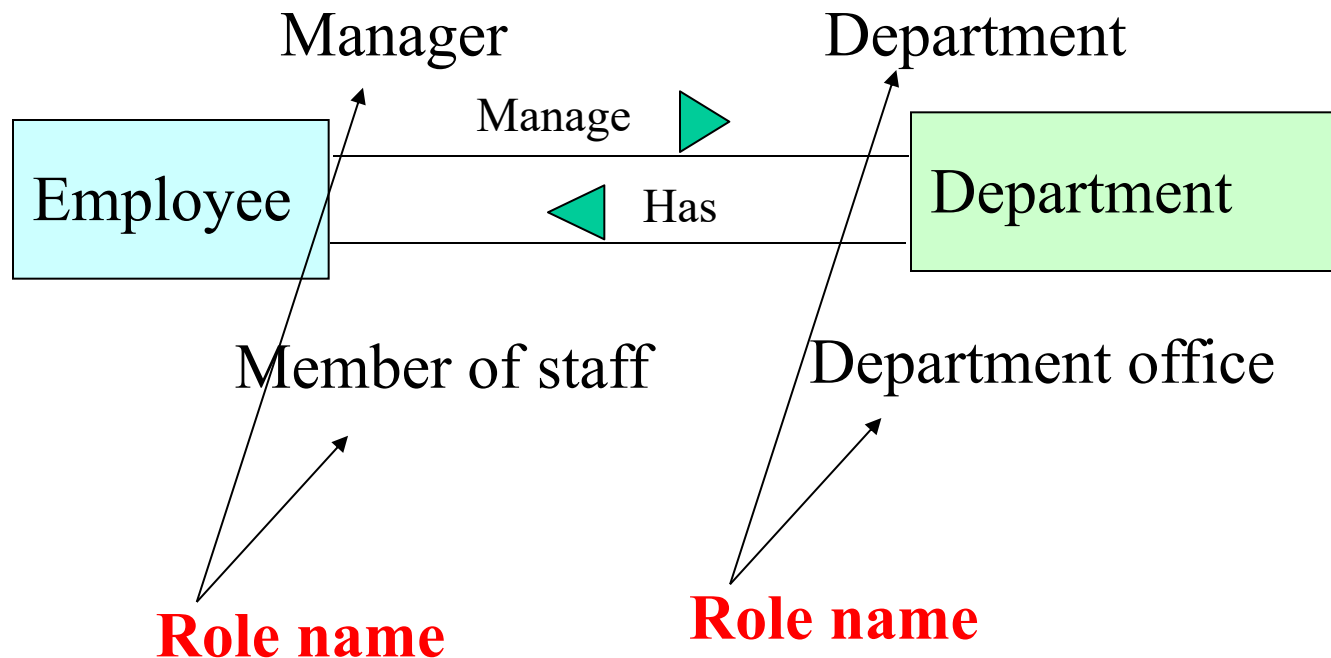
UML标记：



递归关系

(Recursive Relationships)(2)

一个**实体类型**可以充当**不同的角色**，因此两个实体类型之间可能存在多种关系。同一个实体类型在不同的关系中扮演不同的角色。例如：



属性(Attributes)

❖ 实体类型或者关系类型都有属性。例如，Employee 实体类型的属性有：name, salary, title等。属性名称以小写字母开头。属性有域(**domain**)的概念。

简单属性的类型

❖ 简单属性与组合属性：

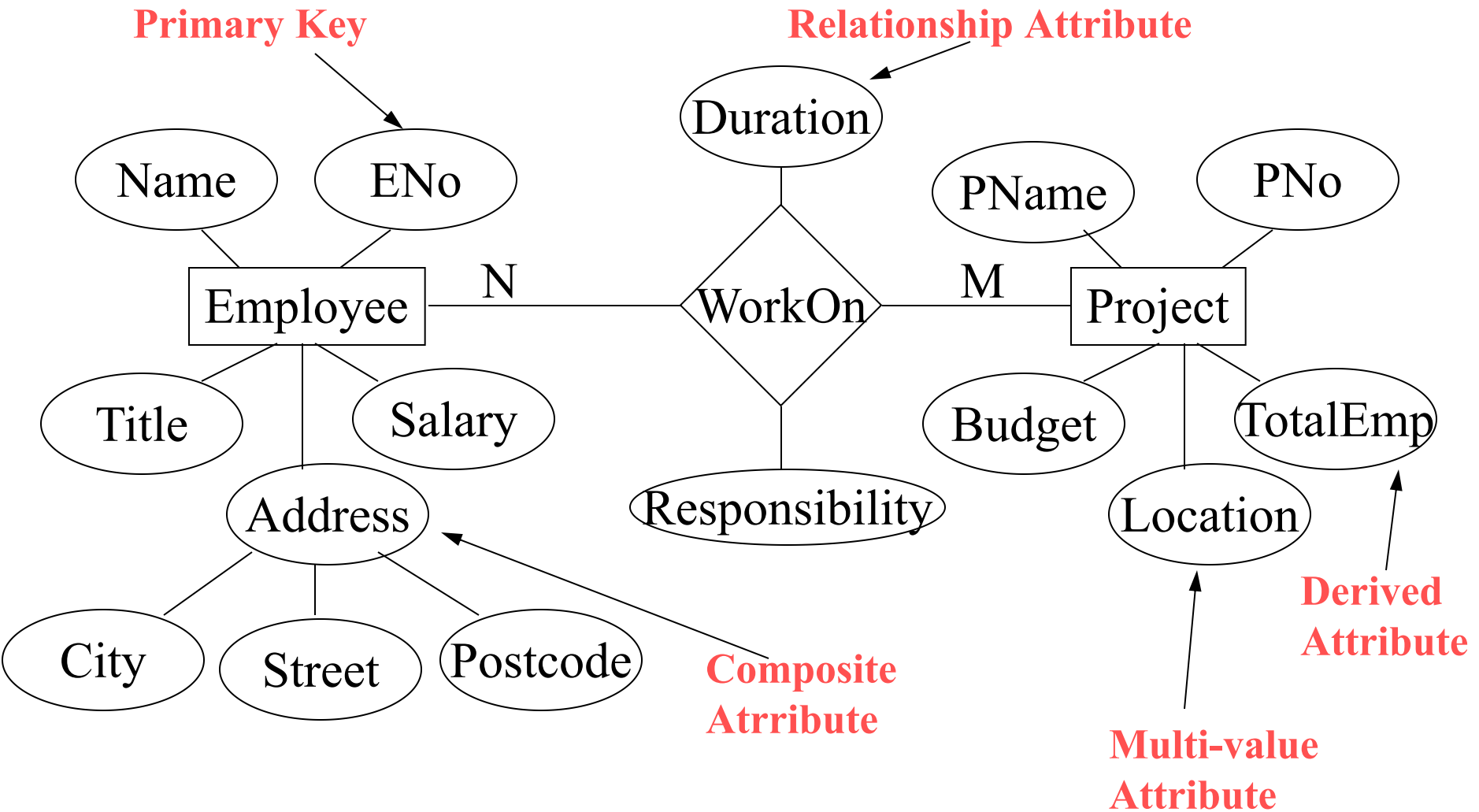
- 例如，**工资**是一个简单属性，**通信地址**则是一个组合属性，它由 postcode , street, city 三个组件组成。

❖ **Question:** 姓名是简单属性还是组合属性？

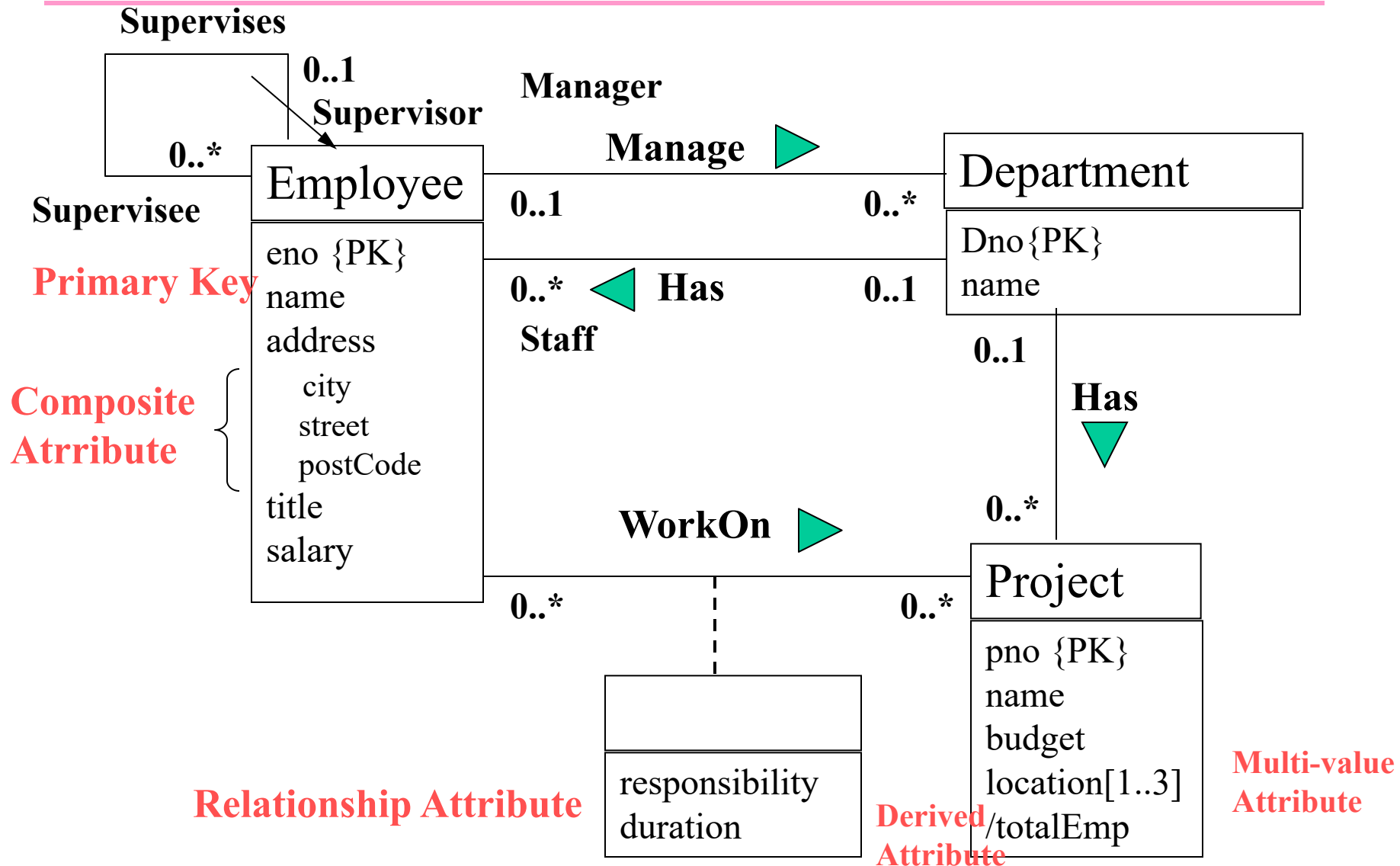
❖ 单值属性与多值属性：

- 例如**工资**是单值属性，**电话号码**是多值属性：有手机号码，家里电话，办公电话；
- **推导属性**：能够由其它属性计算得出的属性；例如，**年龄**，能由出生日期计算而得；

ER Model 例子(旧标记)



ER Model 例子 (UML 标记)



约束(Constraints)

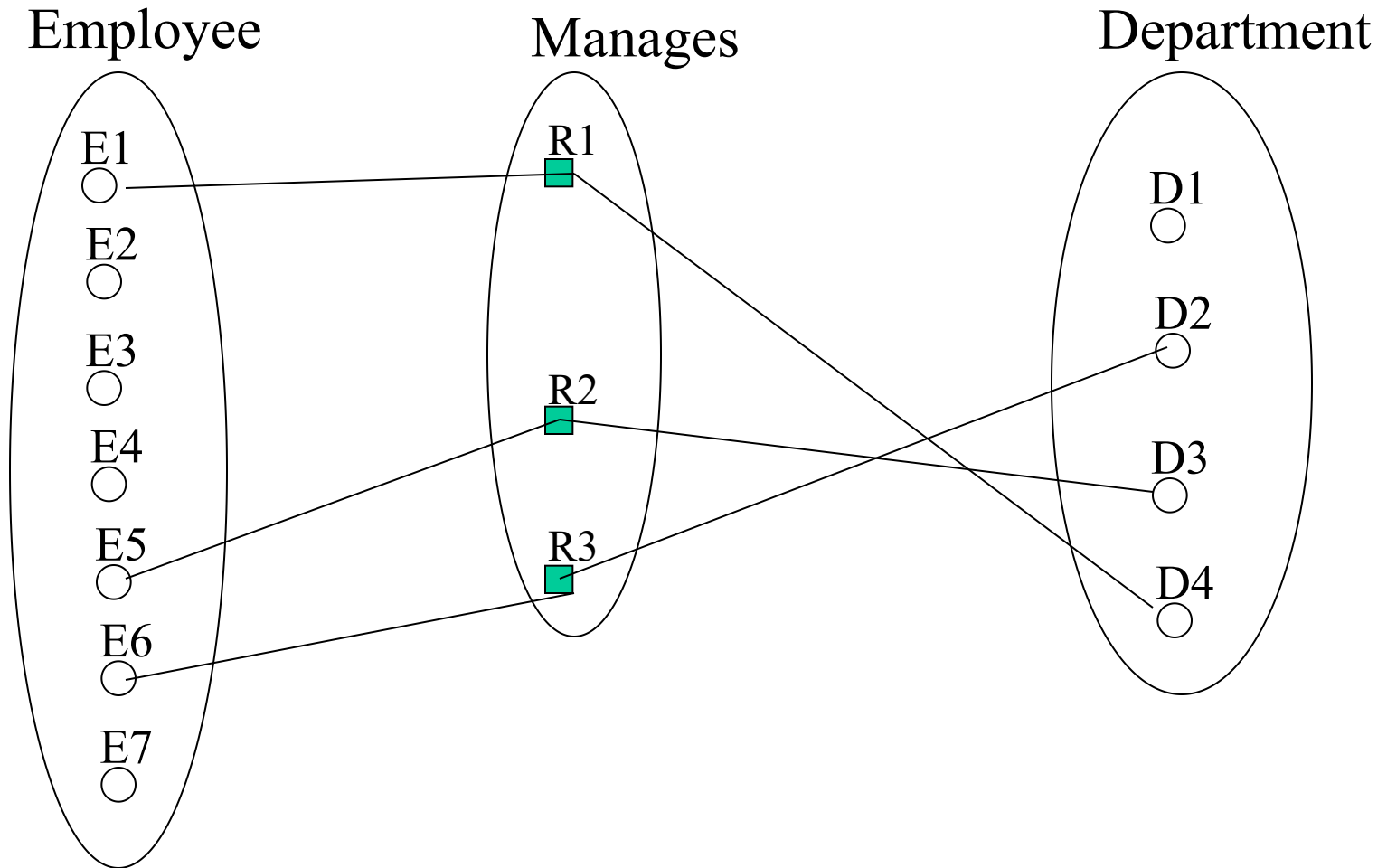
❖ 作用于**实体类型**上的**约束**:

标识属性(primarykey), 实例集中, 标识其实例的属性;

❖ 作用于**关系类型**的**约束**:

对于二元关系, 共有三类: 1对1关系, 1对多关系, 多对多关系; 进一步细化为两极: 参(participation)和基(Cardinality);

一对一关系



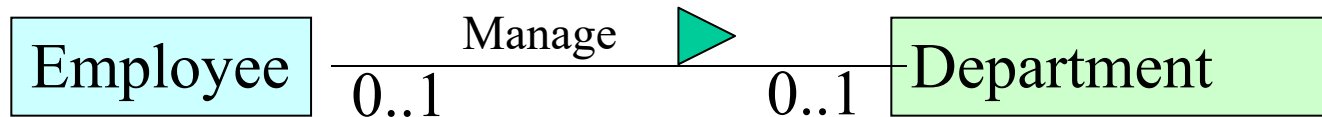
AS manager role

一对一关系

I旧标记:



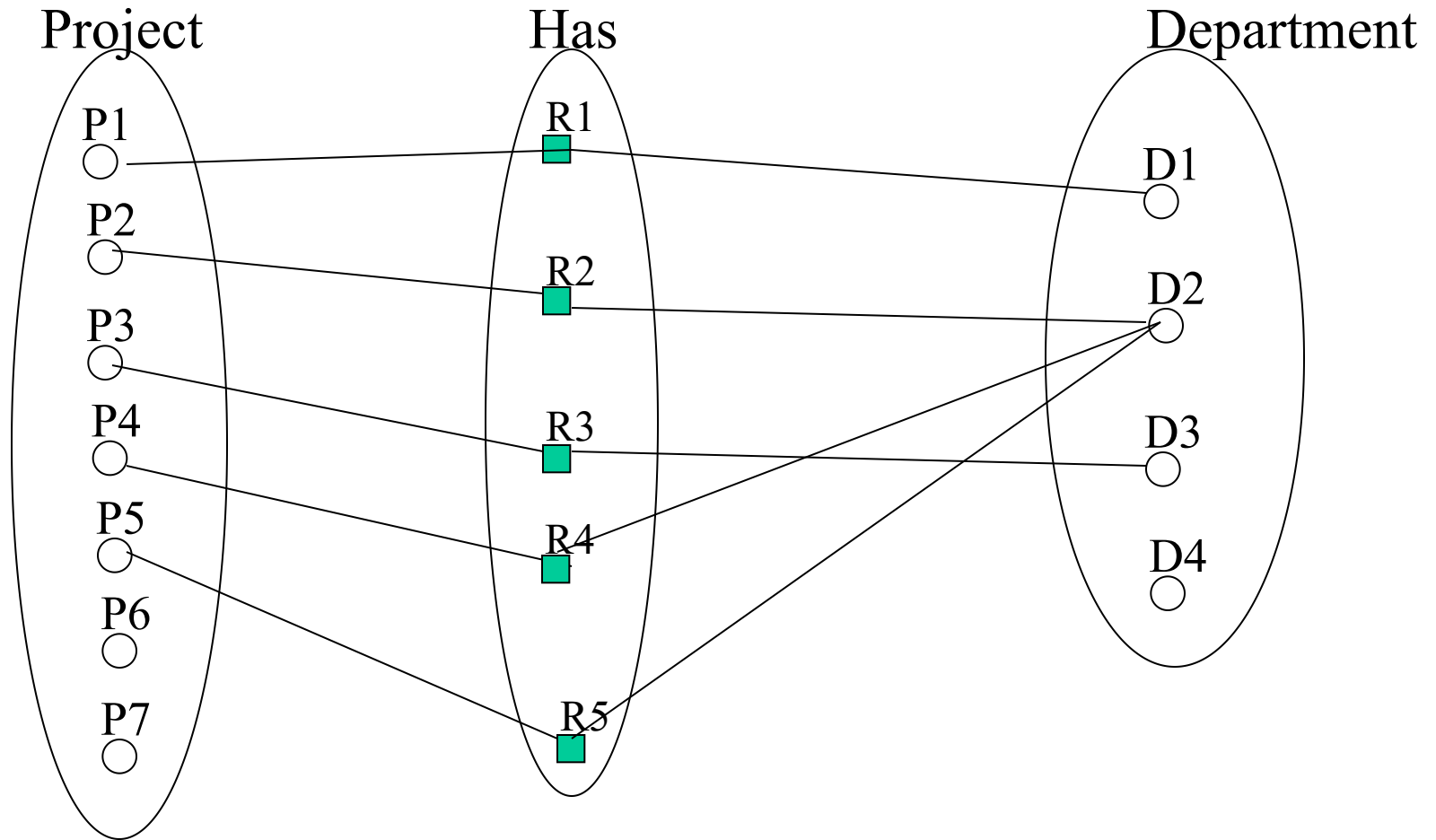
UML 标记:



Each department
may have a manager.

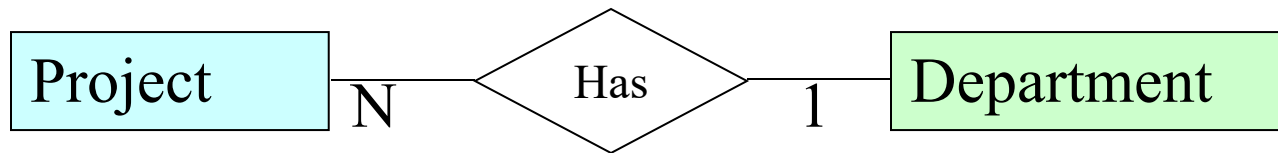
Each employee manages
zero or one department.

一对多关系

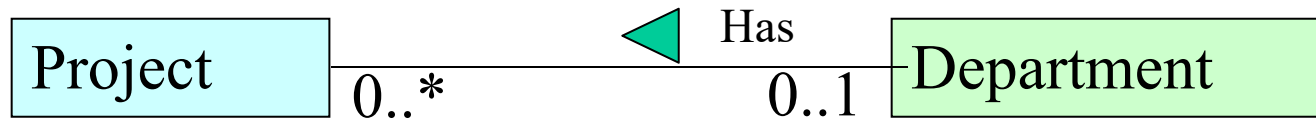


一对多关系

旧标记:



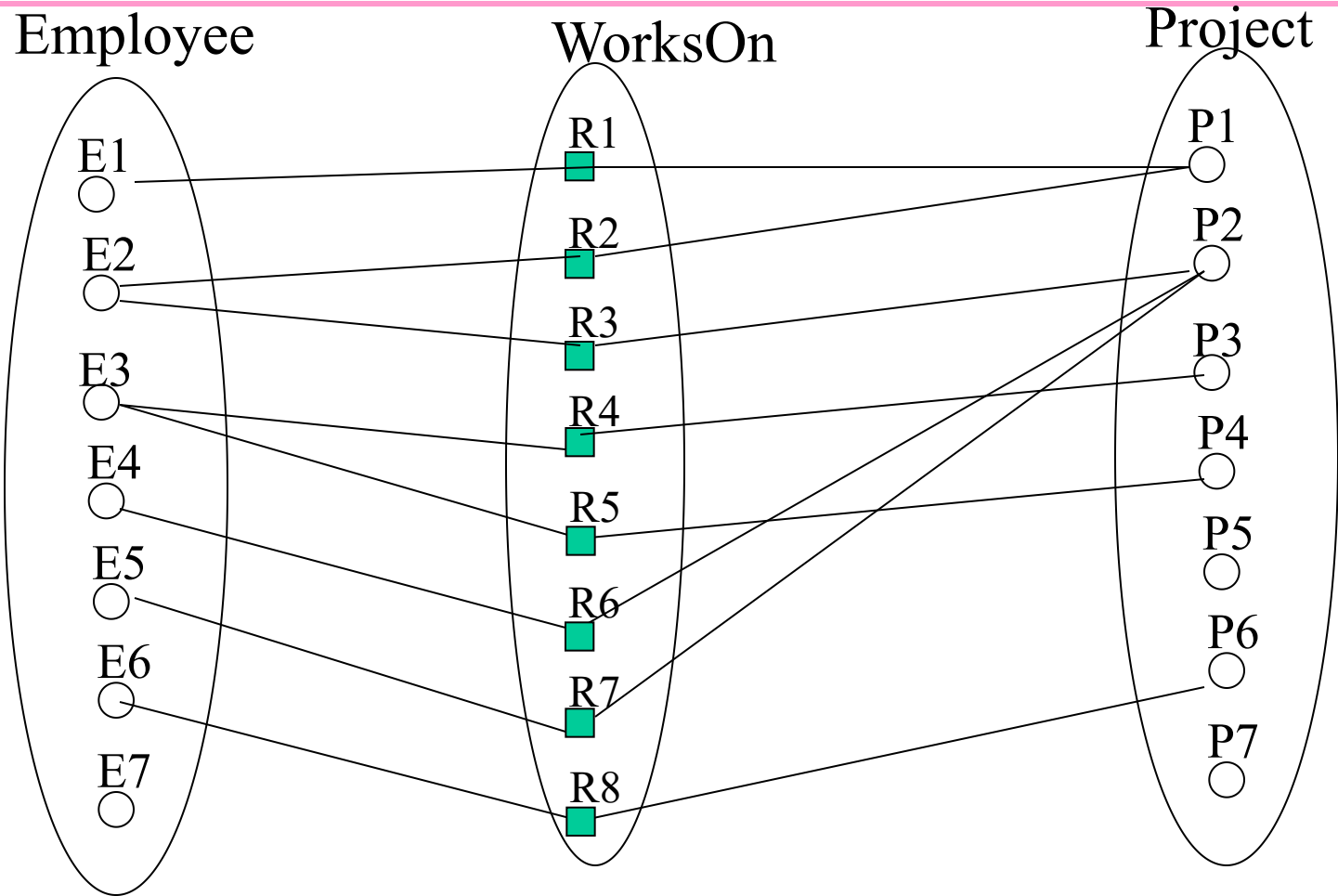
UML 标记:



Each department
Has zero or more projects.

Each project has
zero or one departments

多对多关系

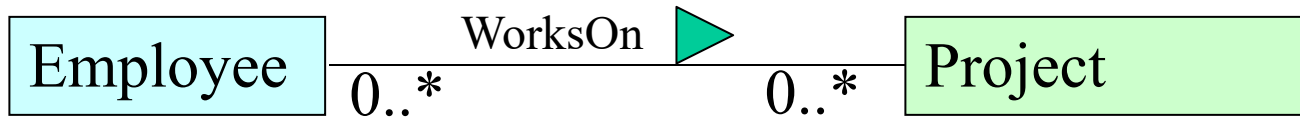


多对多关系

旧标记:



UML标记:

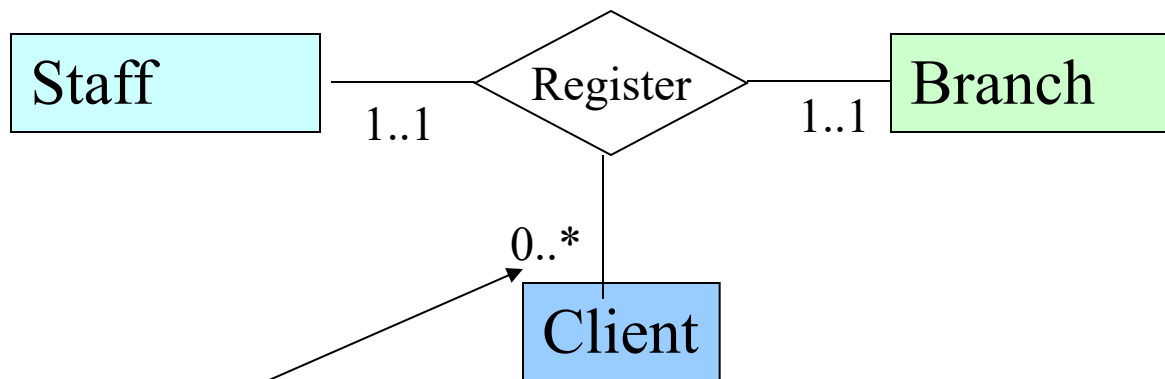


Each project has
zero or more employees.

Each employee works on
zero or more projects.

多元关系中的约束标记方法

对于多元关系，它至少有三个端，当标记某一端的约束时，其它端的实例对象数都设为1，然后从最悲观和最乐观视角来确定约束。



For each Branch/Staff combination,
have zero or more clients.

ER建模案例演示

Please design a database for bachelor teaching (本科教学) of Hunan university. Each college(学院) offers at least 3 courses for students. A college has a name and a building location, and has some teachers. Each course has a name, course id, and course hours, and textbook. Each teacher can teach most one course in a semester(学期), but can teach different course in different semester. Each student has an id, name, sex, birth date, family address, phone. Students can enroll in (选修)one or more courses of Hunan university in a semester (学期). Once a student completes a course, he or she receives a grade (分数). If a student fails to pass the examination, he or she can enroll in the course once again later. Each teacher has name, id, sex, rank(职称), and has emails for communication with students.

概念抽取

Construct a university database where:

- ❖ Each **student** has an id, name, sex, birth date, and grade.
- ❖ Each **teacher** has a name and is in a department.
- ❖ Each **department** offers some courses and has lecturers. A department has a name and a building location.
- ❖ Each **course** has a name and number and may have multiple sections.
- ❖ A lecturer apply to offer a course for student to enroll.
- ❖ Students **enroll** in courses. They may only enroll in a course once. Once a student completes a course, they receive a grade.

第一步：标识实体

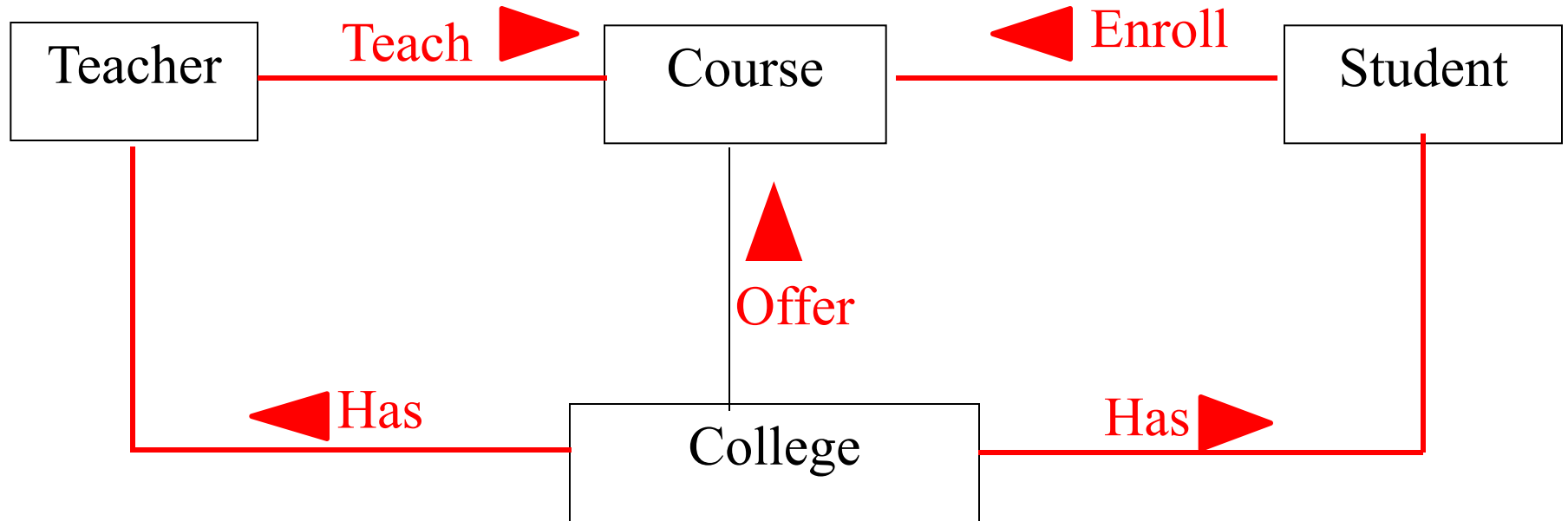
Teacher

Course

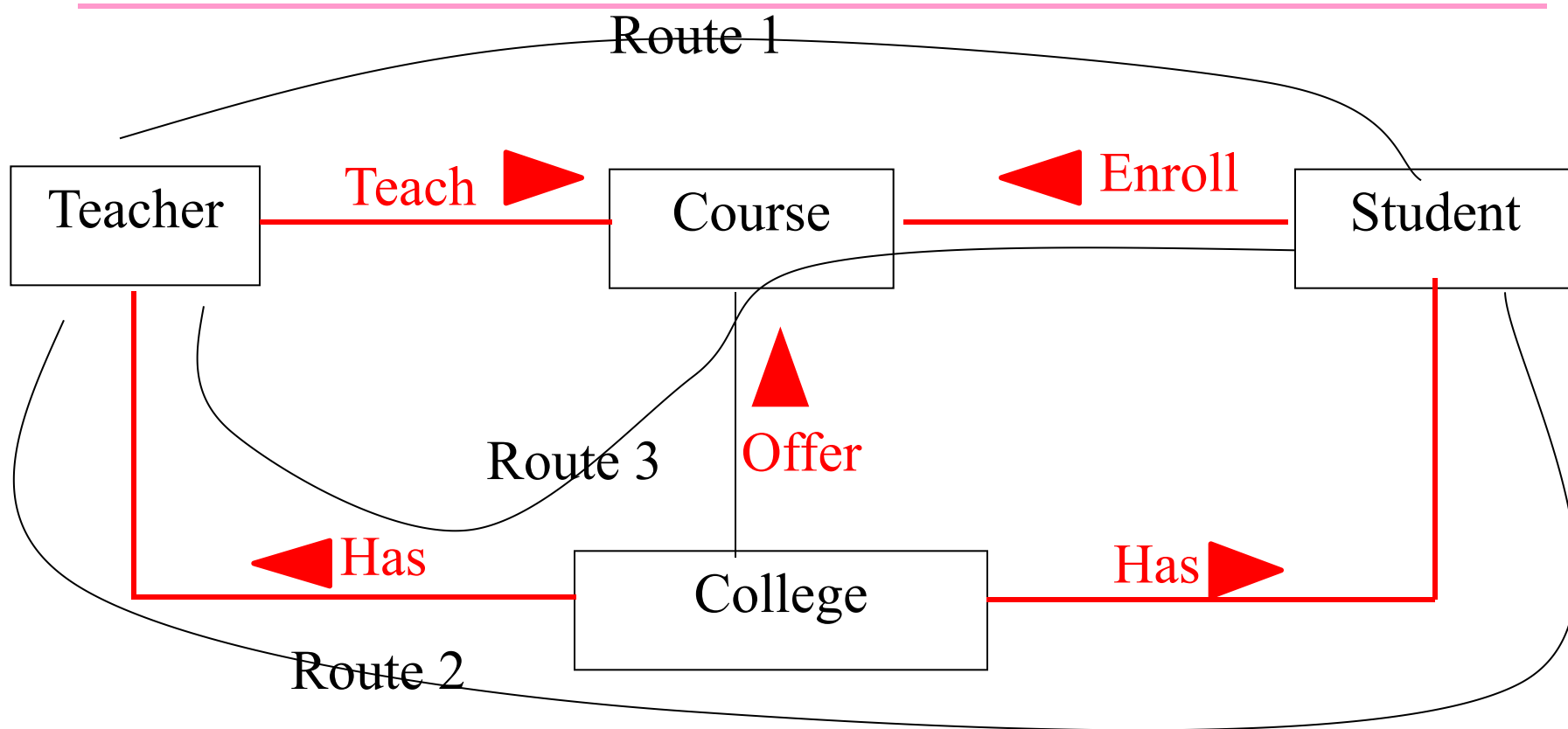
Student

College

第二步：标识关系



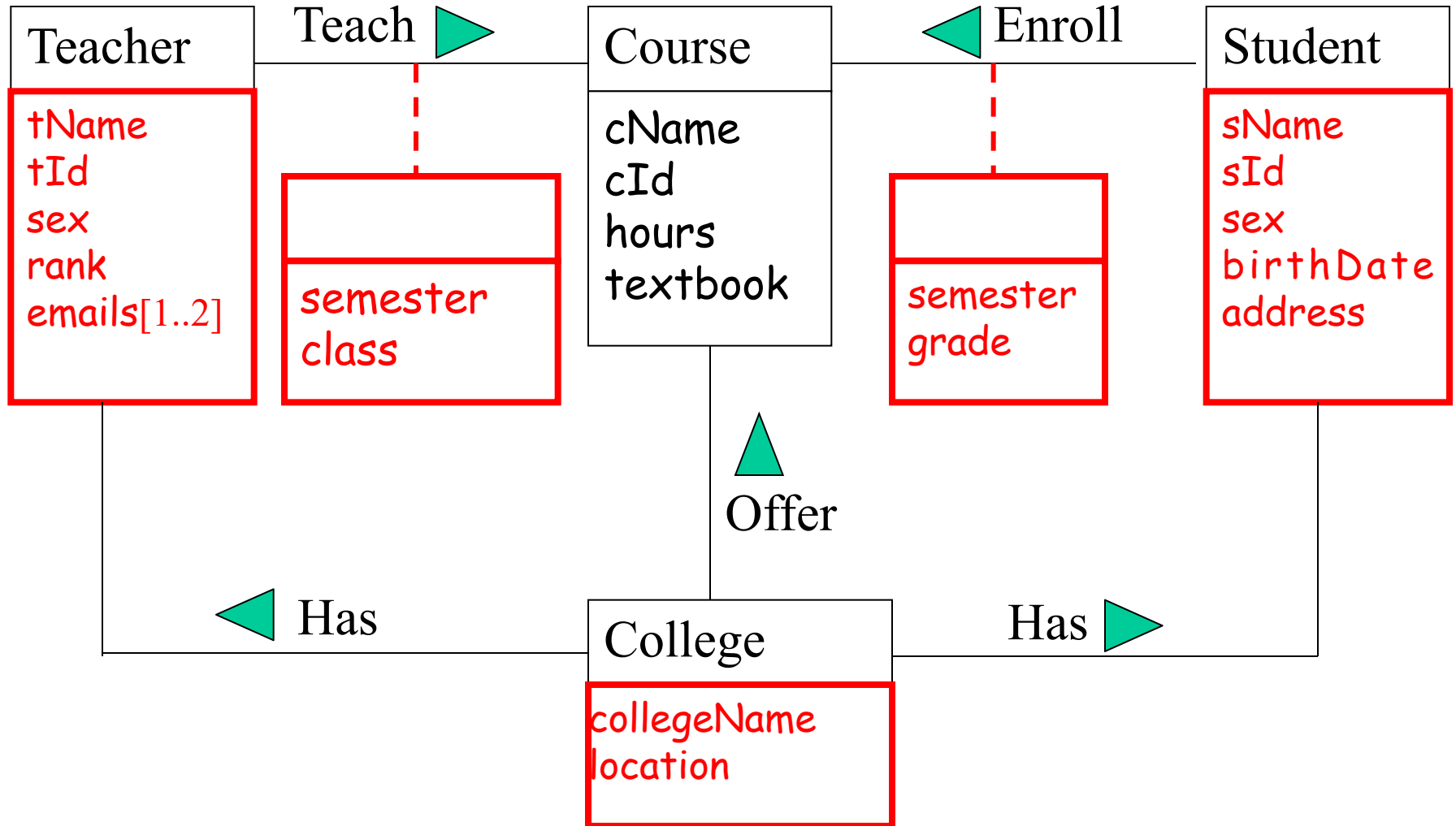
检查关系



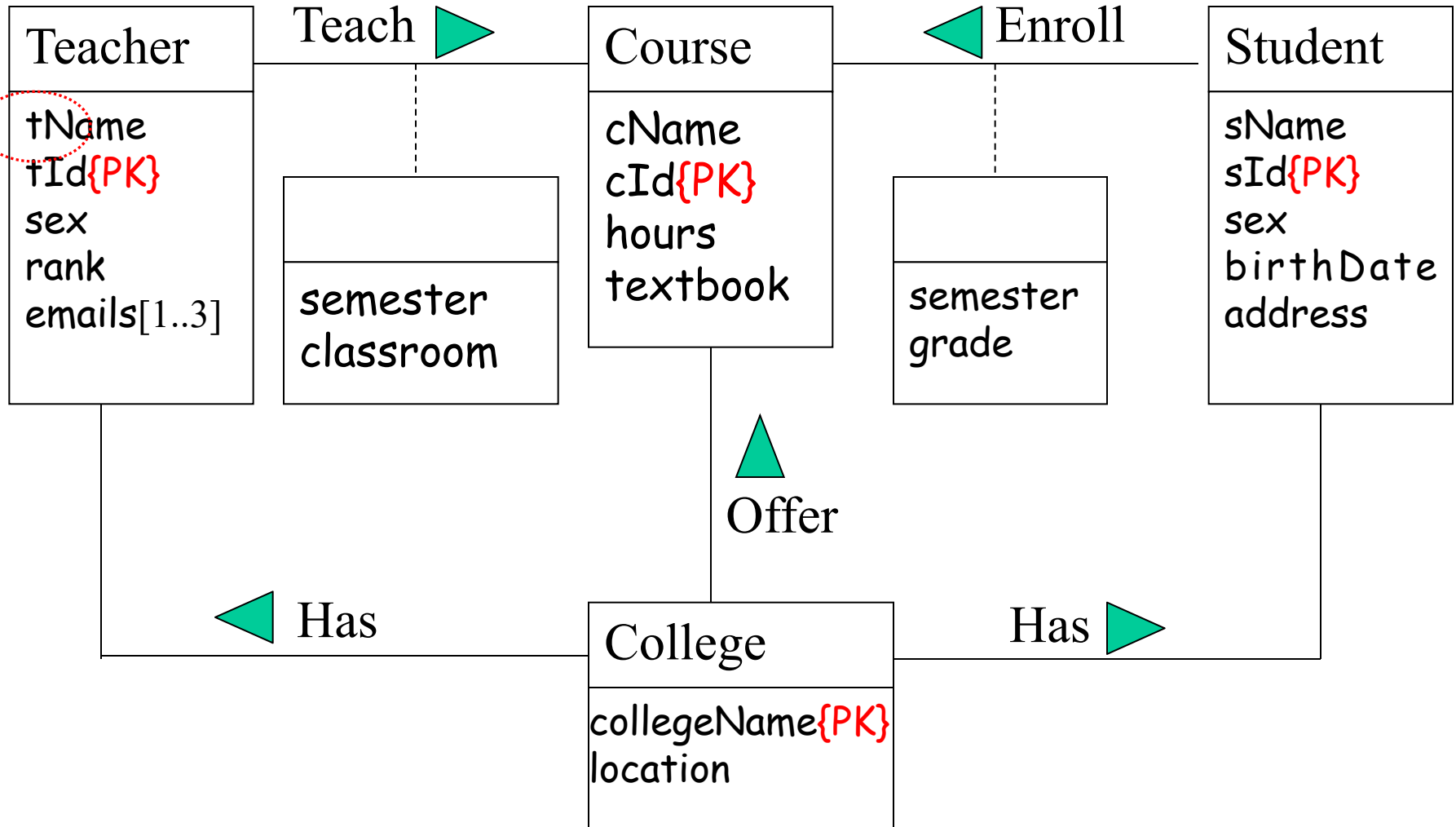
Teach $\bowtie$ Enroll = route 1;

But Has $\bowtie$ Has $\neq$ route 2; Has $\bowtie$ Offer $\neq$ Teach;

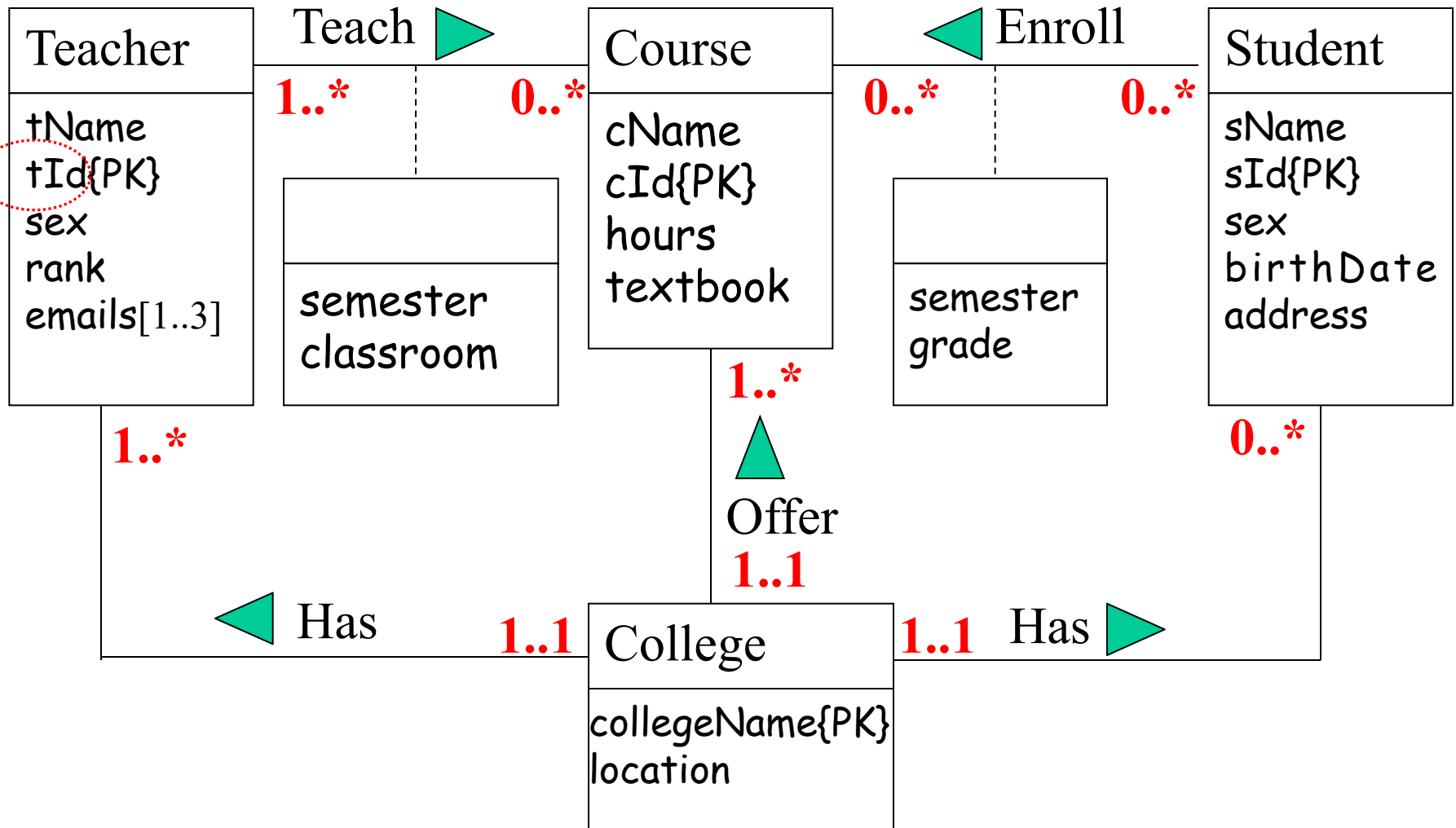
第三步：标识属性



第四步：标识实体约束



第五步：标识关系约束



Practice Question ?

Consider the university database developed before. 标记如下作用于关系类型上的约束:

A department must offer at least 2 courses and no more than 20 courses. A Course is offered by only one department.

- ❖ A course may have multiple sections, but always has at least one section.
- ❖ A student may enroll for courses (but does not have to).
- ❖ A lecturer may be in multiple departments (at least 1), and a department must have at least 3 lecturers.
- ❖ A course is taught by at least one lecturer, but may be taught by more than one. A lecturer does not have to teach.
- ❖ A student may only enroll in a course once. (Not keeping track of history of student enrollments.)

Homework

[Http://ss.hnu.cn/database/index.htm](http://ss.hnu.cn/database/index.htm);

先看案例库中的case C, 然后仔细看案例分析, 自己做一遍案例分析中的内容, 然后独立完成案例库中的Case D:

要求:

- a) ER model;
- b) Relational model;
- c) DDL for the relational model;
- d) DML for the questions (a) - (n);
- e) Sample data;
- f) Web user interface;

强实体类型和弱实体类型

强实体类型，是指它的实例的存在不依赖于任何其他实体类型的实例；

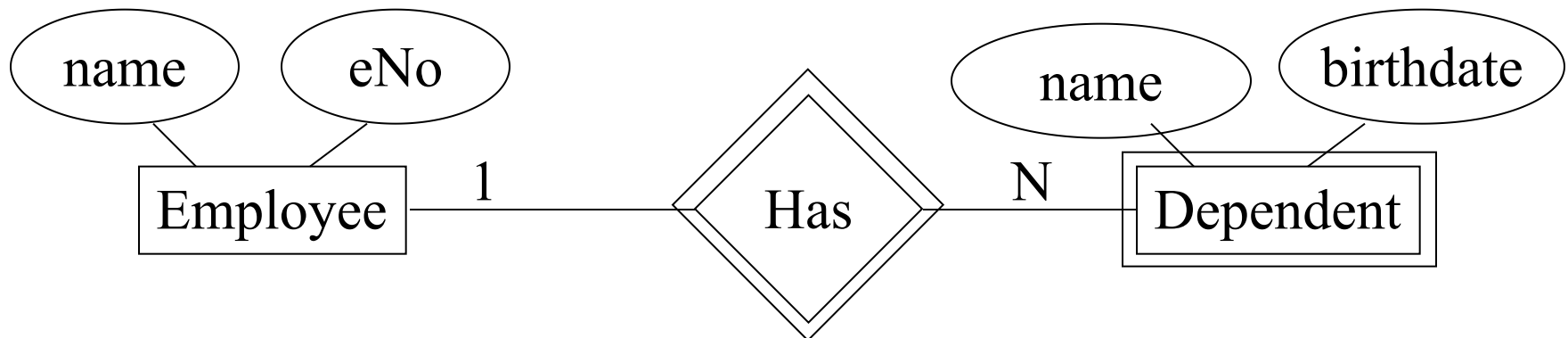
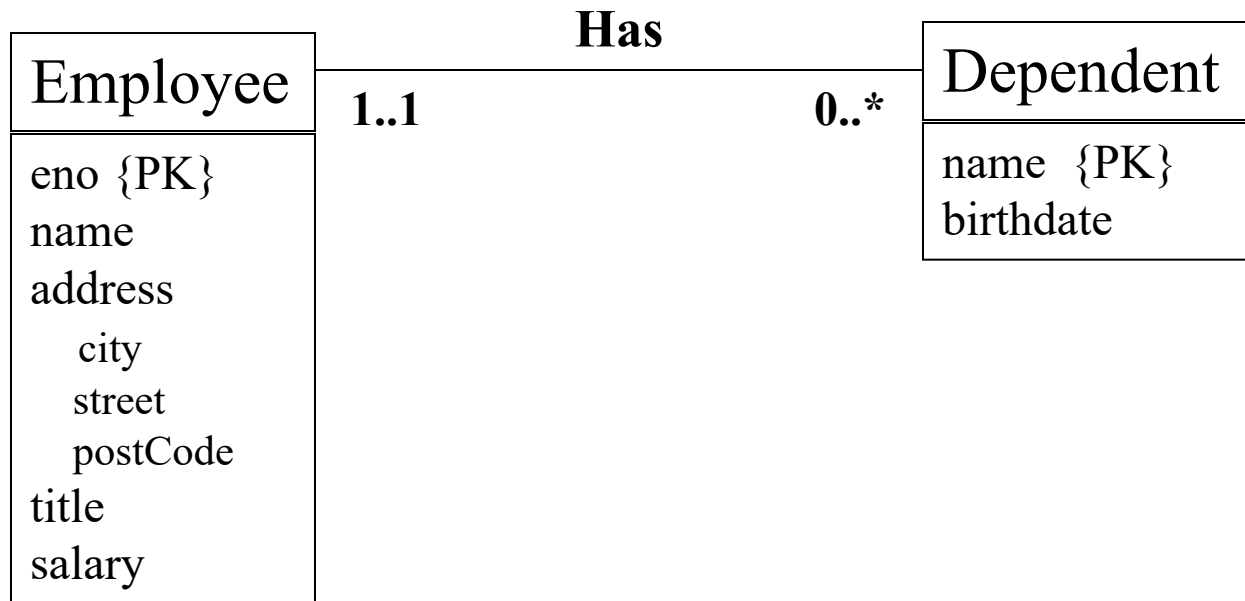
它有自己独立的主键，唯一性地标识它的实例；

弱实体类型，是指它的实例的存在依赖于任何其它实体类型的实例；

它的主键包括两个部分：

- 1) 它所依赖的实体类型的主键；
- 2) 在此基础上，再加上它自己的标识性属性；

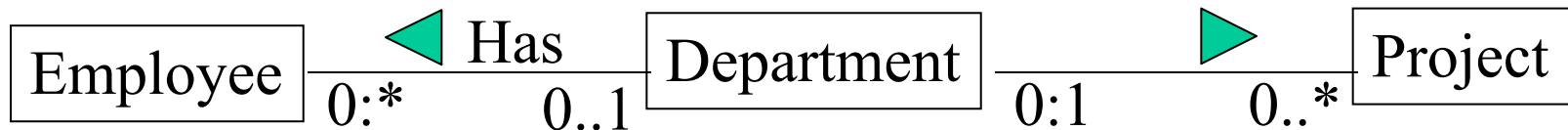
Weak Entities in ER Diagrams



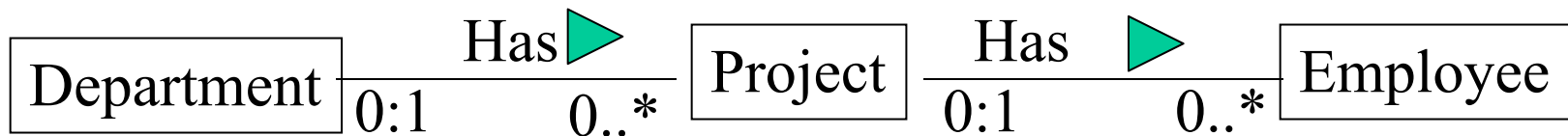
建模陷阱(Modeling Traps)

要注意两类连接陷阱:

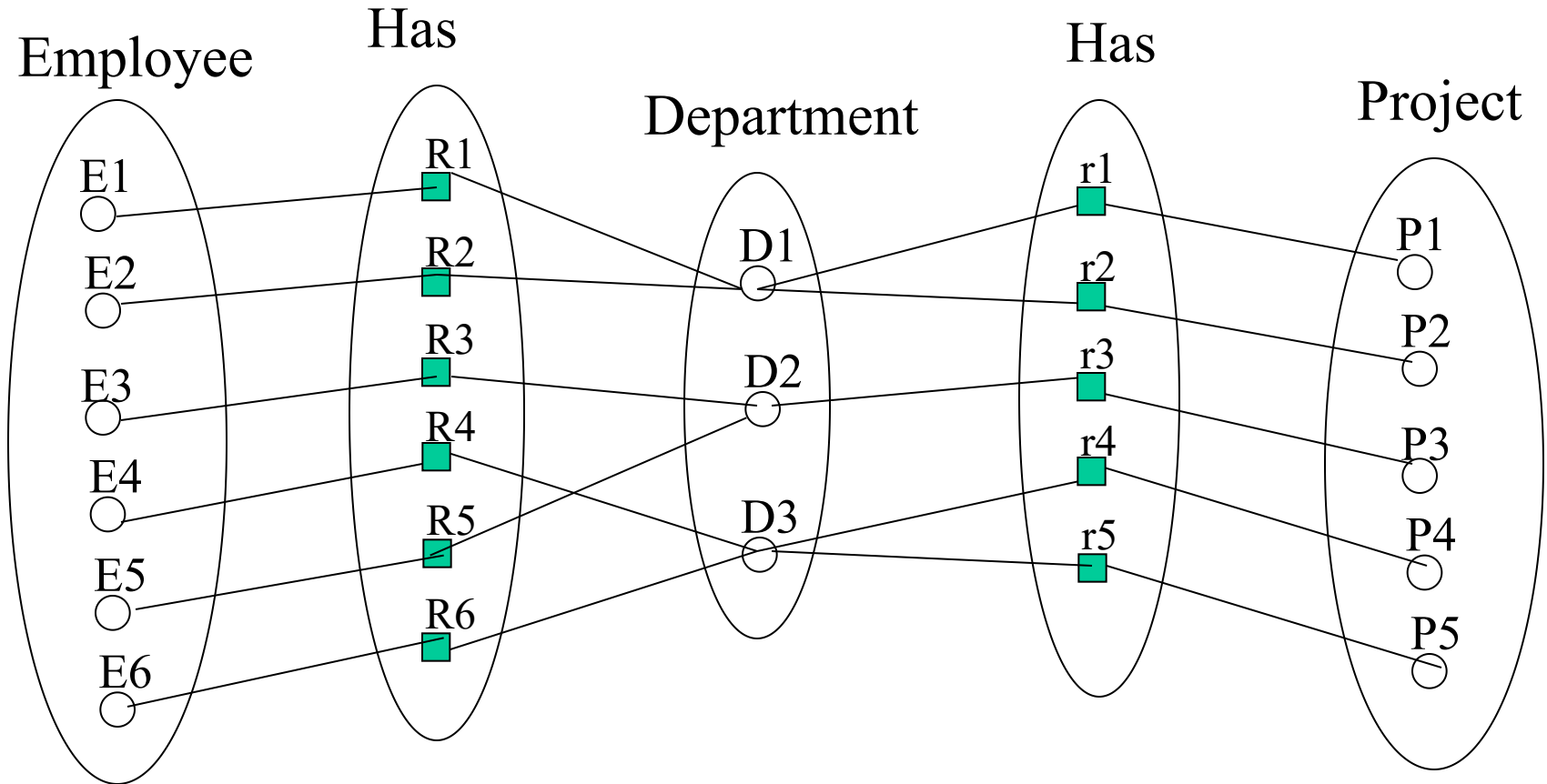
- 扇子陷阱(Fan traps):



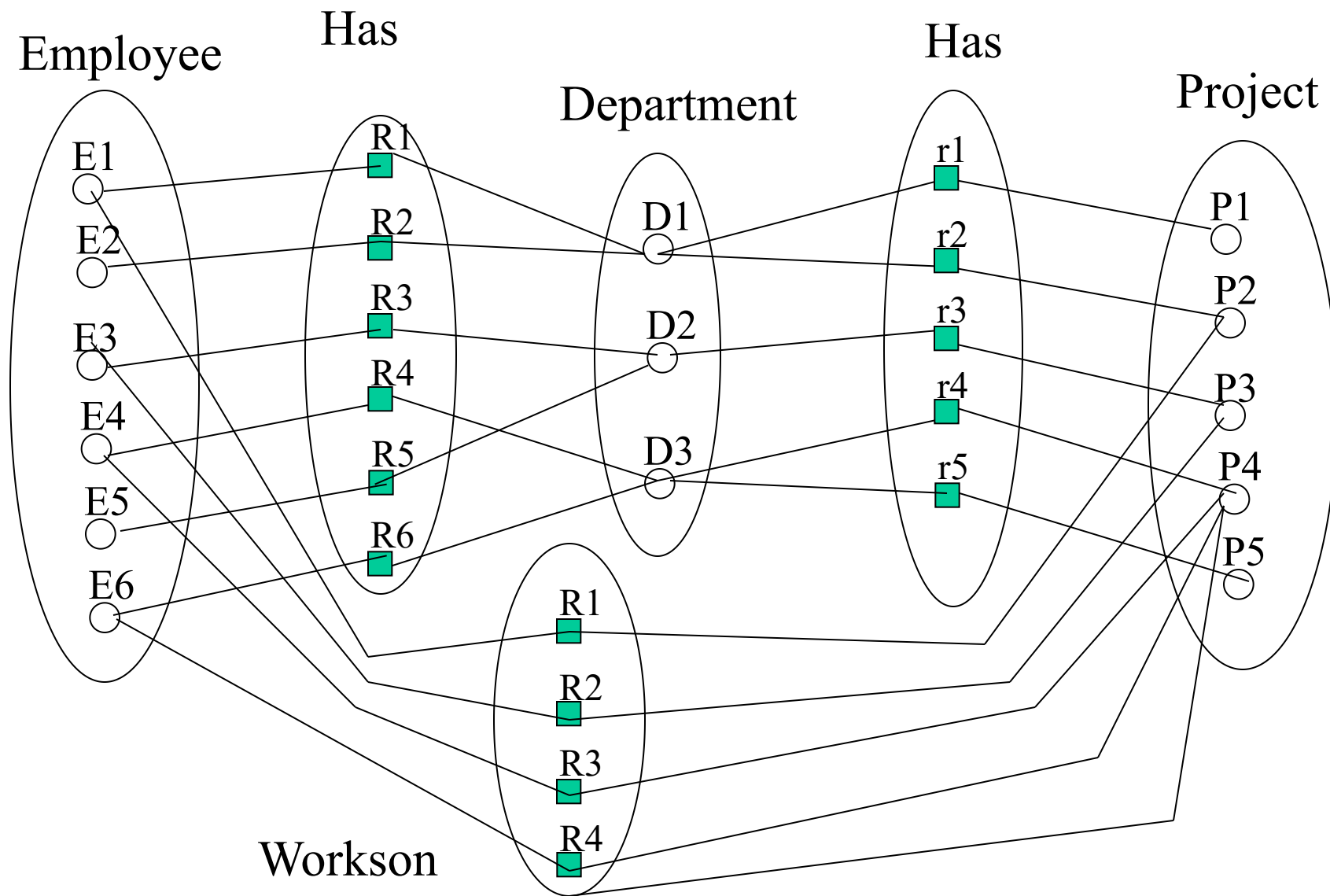
- 裂口陷阱(Chasm traps)



扇子陷阱(Fan traps)例子



现有的两个关系类型尽管把员工实体类型和项目实体类型连接起来量，但是已有的两个关系并不能精准刻画出员工与项目之间的关系



扇子陷阱(Fan traps)例子

Employee

E1	D1
E2	D1
E3	D2
E4	D3
E5	D2
E6	D3

Project

P1	D1
P2	D1
P3	D2
P4	D3
P5	D3

通过关系传递
推导出来的关系

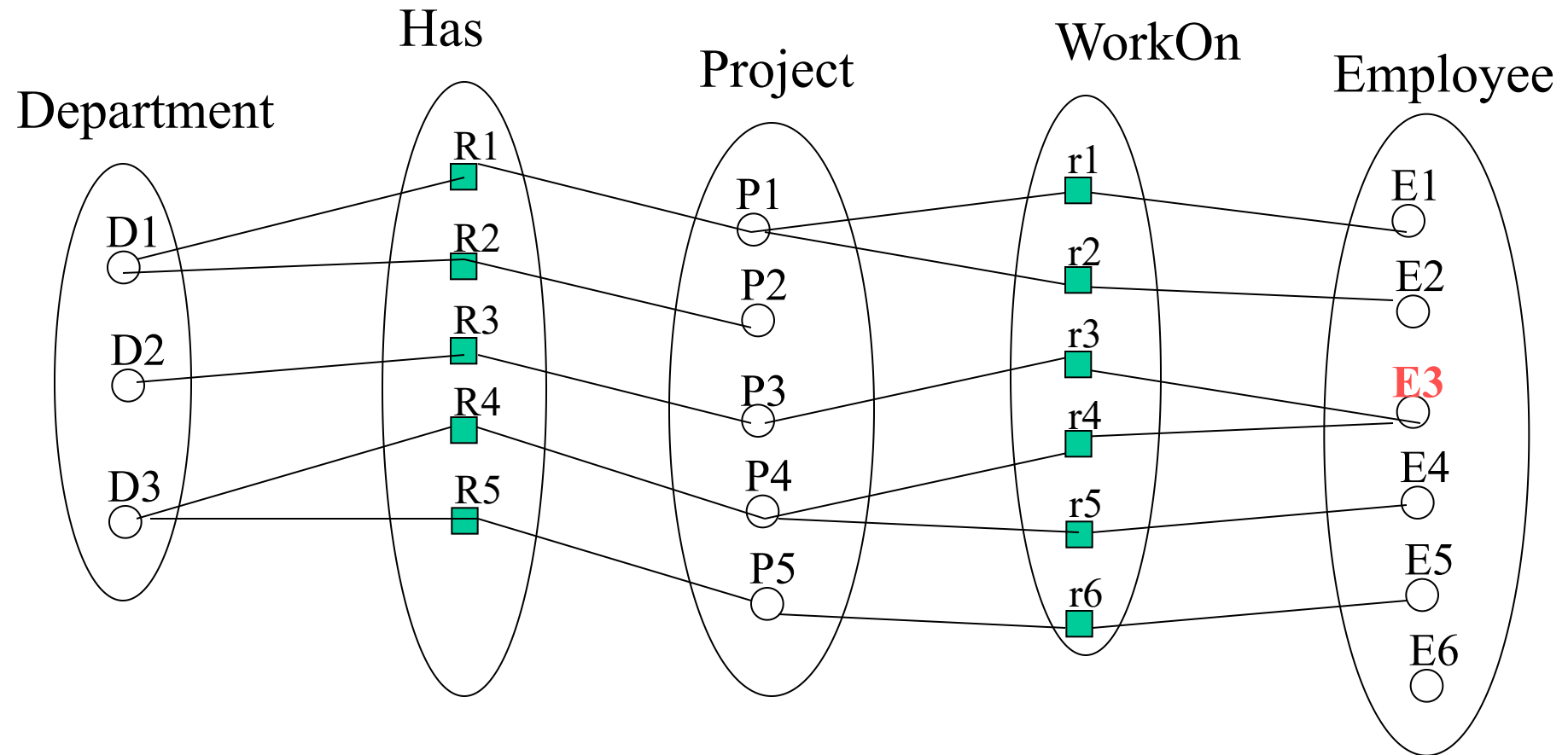


Employee $\bowtie$ Project

E1	D1	P1
E1	D1	P2
E2	D1	P1
E2	D1	P2
E3	D2	P3
E4	D3	P4
E4	D3	P5
E5	D2	P3
E6	D3	P4
E6	D3	P5

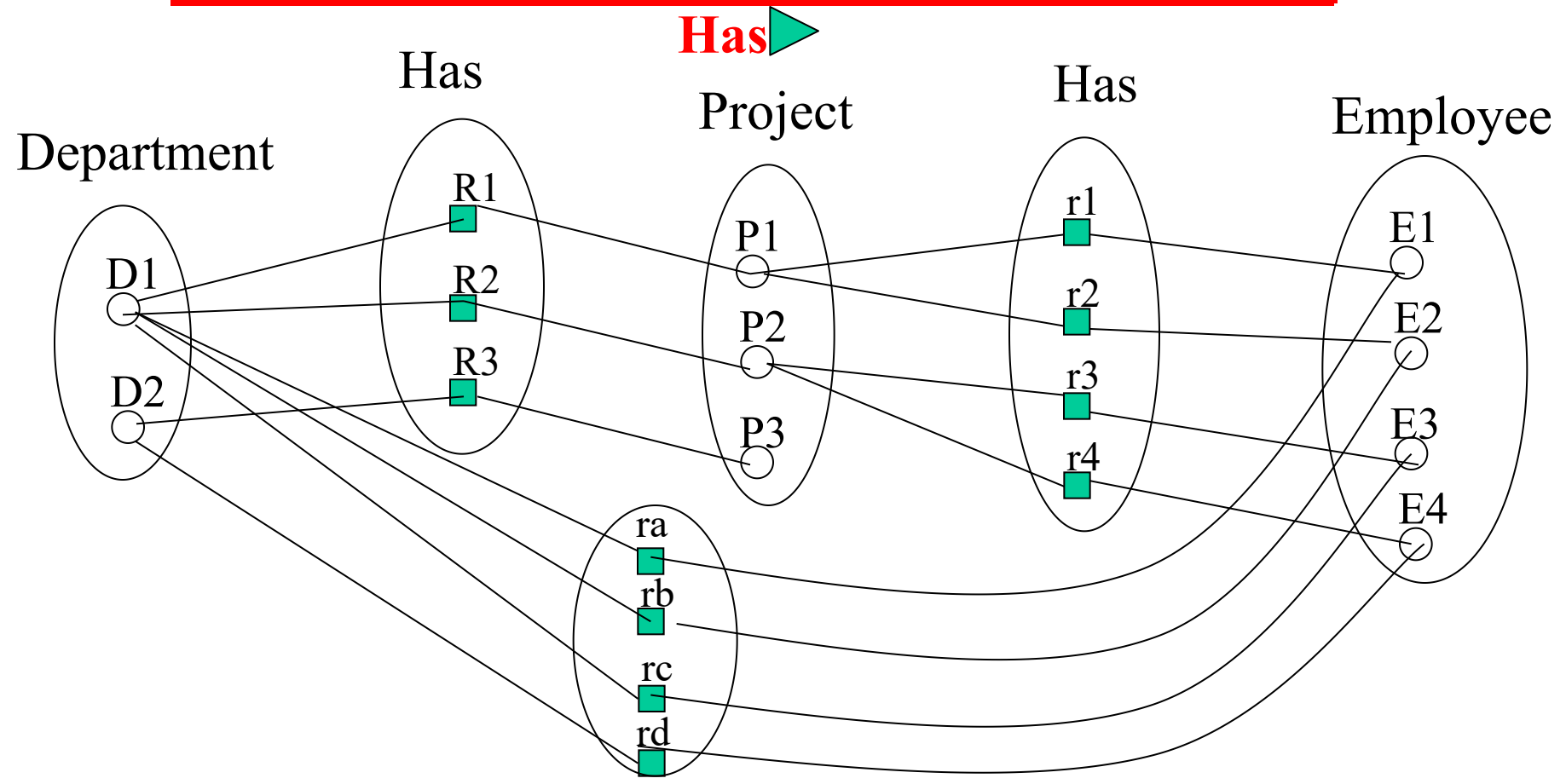
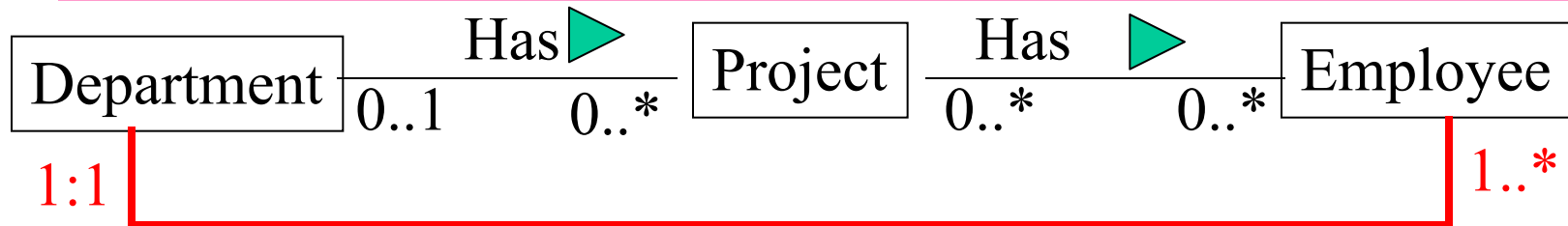
这些推导出来的实例并不是
员工参与项目的真实实例

Chasm Trap 例子



- 员工 E3属于哪个部门?
- D2 部门有哪些员工?

Chasm Trap 例子

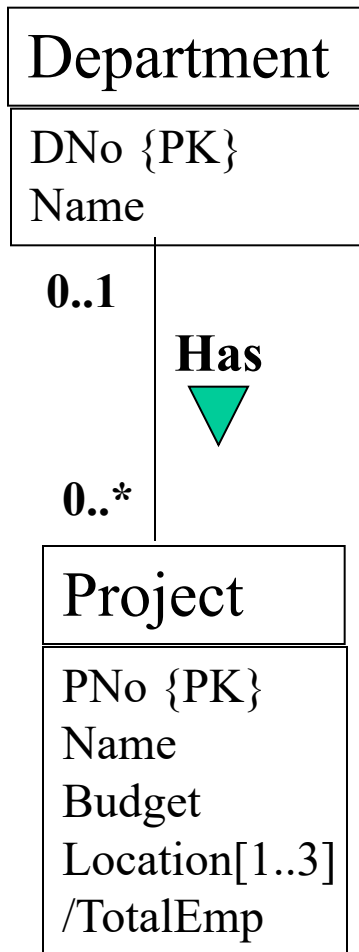


ER建模中应该注意的事项

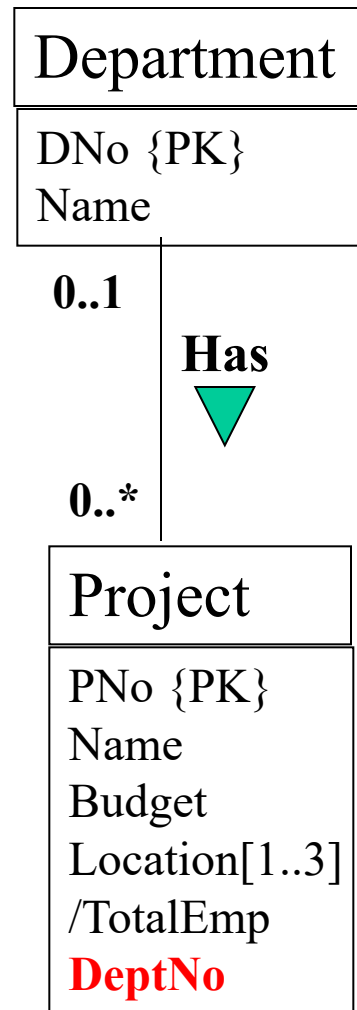
- 1) 避免冗余，不要有多处出现指向同一个概念的属性，实体，关系；
- 2) 能用属性表达的地方不要用实体来表达；

例子

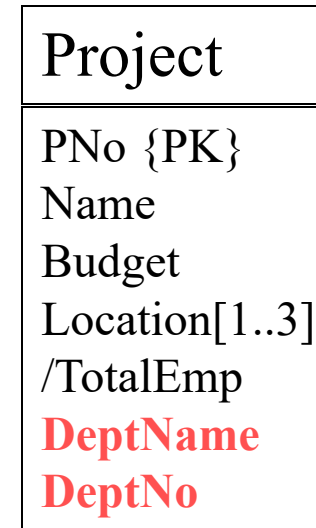
Good



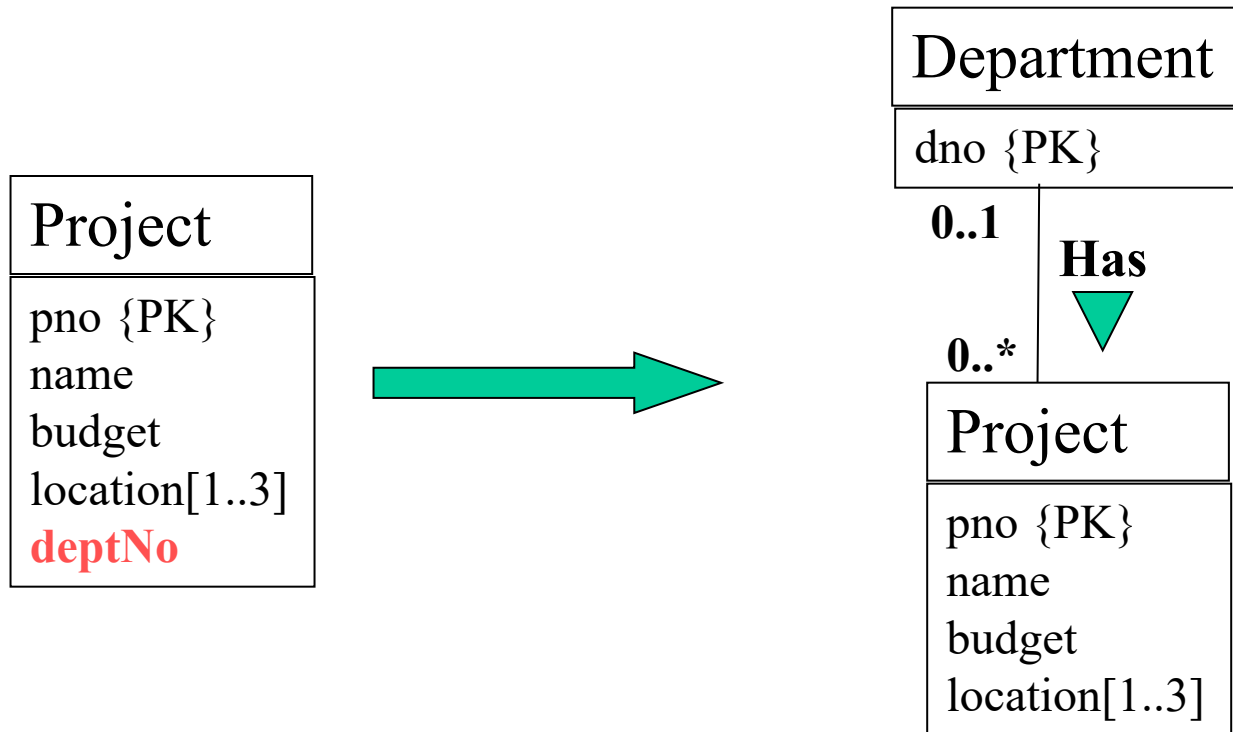
Bad



Also bad



处理方法



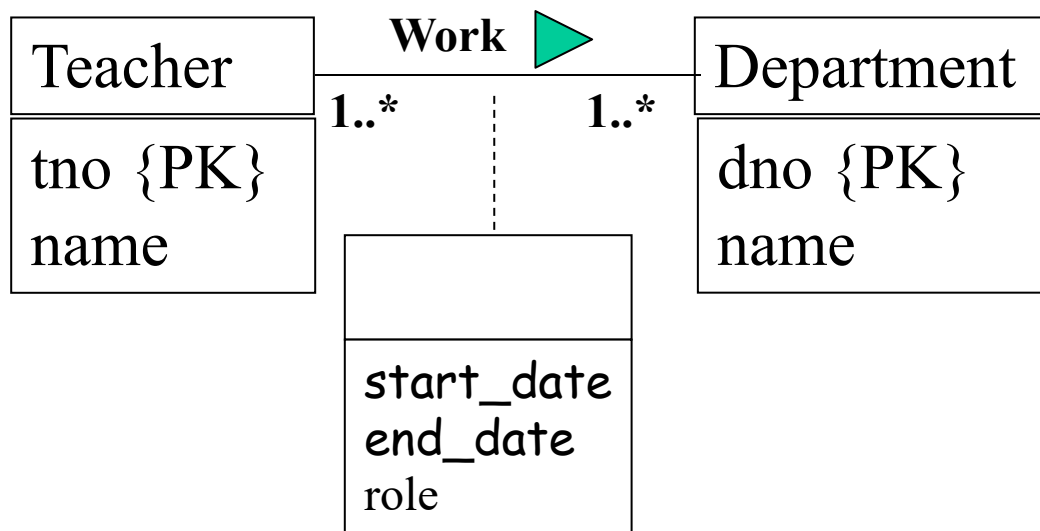
ER建模面临的挑战

实体，关系，属性的界限并不分明，不知道什么时候把一个概念模型化成实体，还是关系，还是属性；

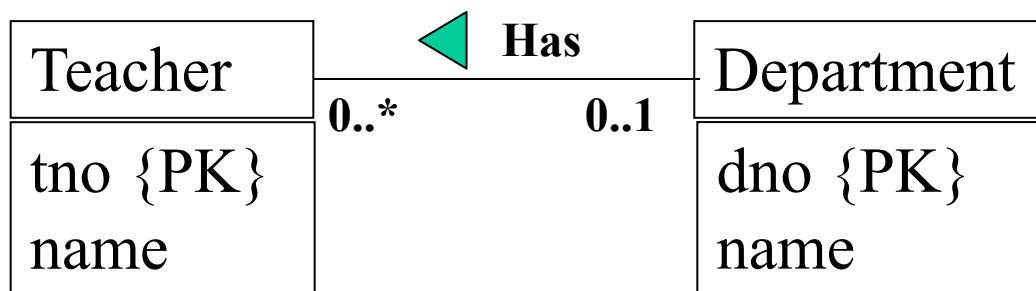
技巧：

- **实体**是**名词**，表**事物**；
- **关系**是**动词**，表**事件**；
- 对**三元关系**和**三元以上的关系**，将其**模型化**为一个**实体**，而且是一个**弱实体**。该弱实体依存于所关联的实体类型。**许多建模工具只允许二元关系，不支持多元关系。**

对关系的认识



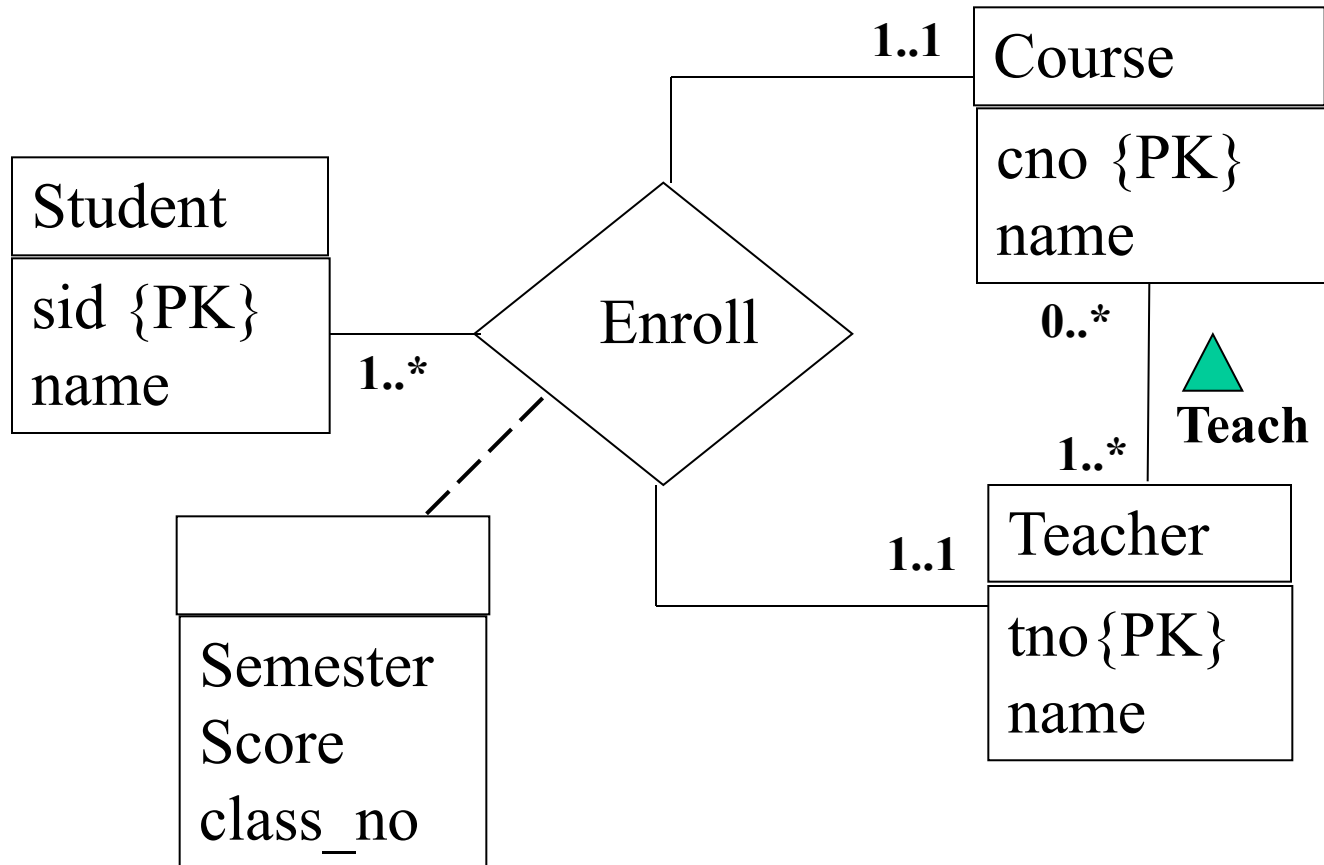
历史记录



当前状态

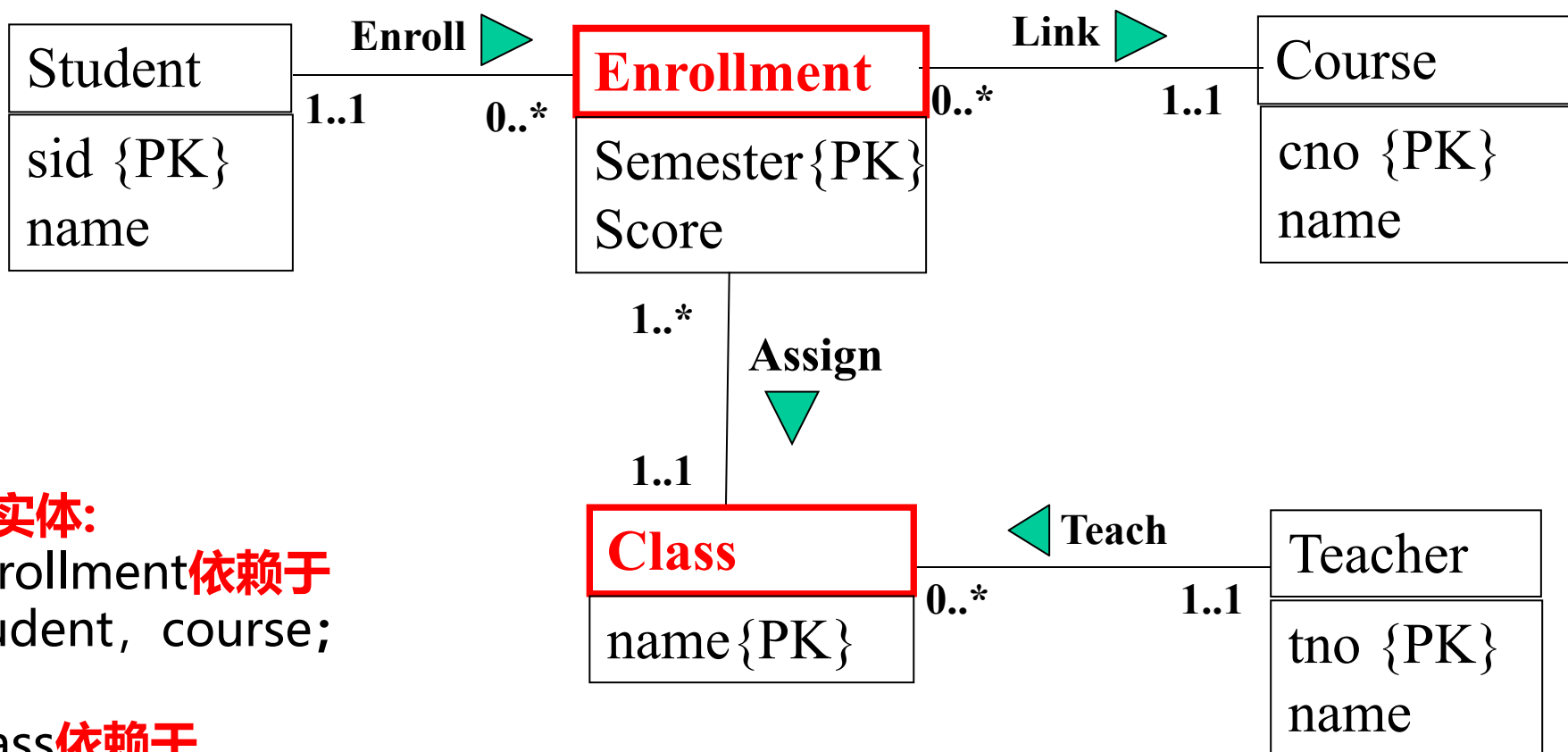
对多元关系的处理

❖ 学生选课.



把关系模型化成实体

❖ 变成二元关系处理。



弱实体:
enrollment依赖于
student, course;

Class依赖于
enrollment

ER Modeling技巧

实体之间的有**从属关系**，使得实体之间构成了**树状结构图**，此时必须要考虑它们之间是否存在**依存关系**。如果是，就要表示**强实体**，与**弱实体**。

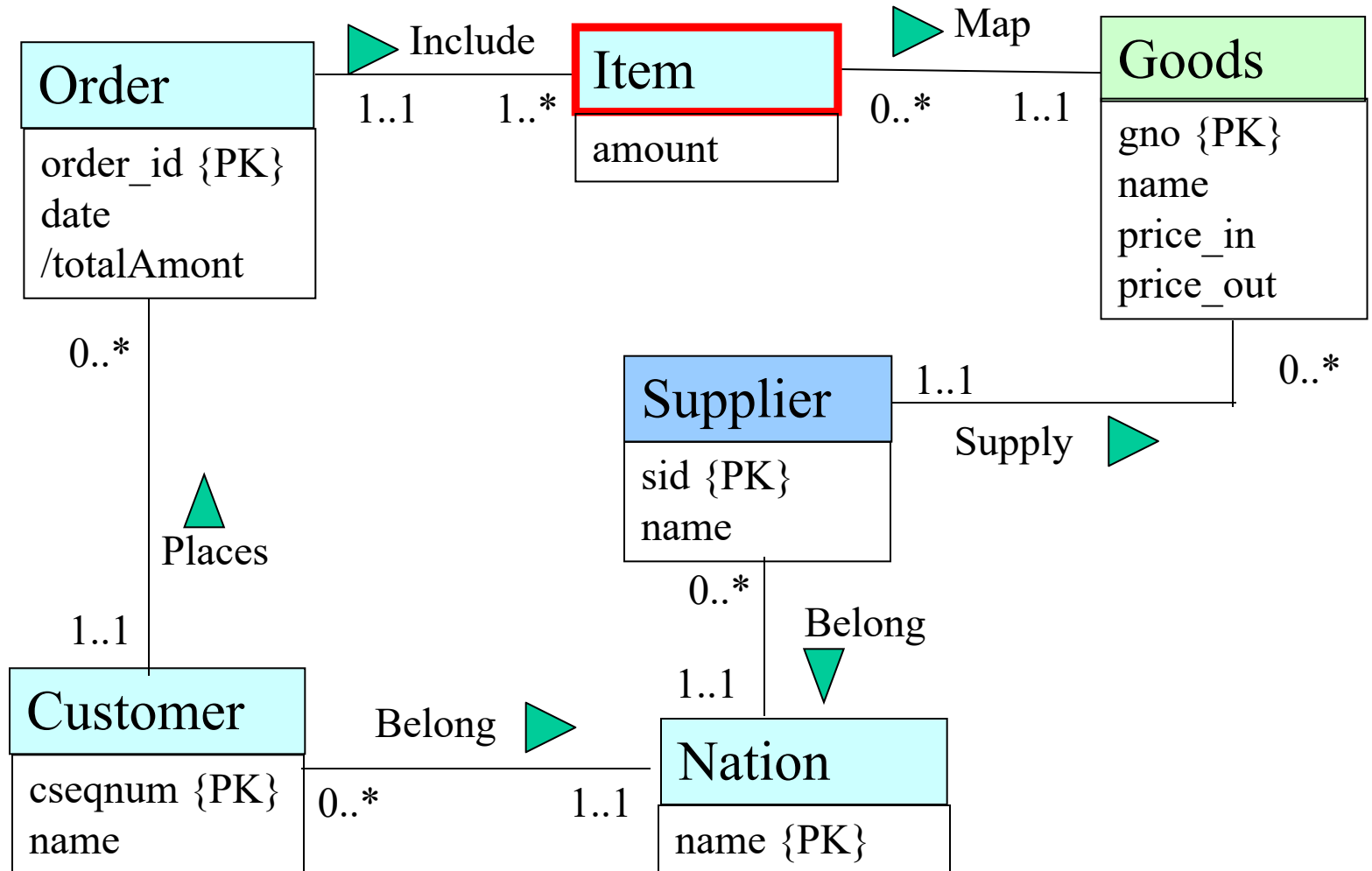
对于关系，有两种含义。第一种含义表达抽象概念之间相互联系这种特性，即表达**当前状态**；第二种含义表达事件的发生，即表达**历史记录**。

ER Modeling随堂测试

Business of a mall is as follows, please design a ER diagram for its database:

- a) Each **order** has a numeric key, a customer, a date, a total order amount, and **a list of goods**.
- b) A **goods** in an order has an amount and a price paid by a customer, and is supplied by a certain **supplier**.
- c) Each goods has a numeric key, a name, and a price and may be supplied by multiple suppliers.
- d) A supplier has a key, a name, and a phone and may supply multiple parts.
- e) A **customer** has a key and a name.
- f) Each supplier and customer is located in a **nation**.

ER Diagram Designing



ER Design Question #2

Construct an invoice database where:

- a) An **invoice** is written by a **sales representative** for a single **customer** and has a unique ID. An invoice has a date and total amount and is comprised of multiple detail lines, containing a product ID, price and quantity.
- b) Each **sales representative** has a name and can write many invoices, but any invoice is written by a single representative.
- c) Each **customer** has a unique id, name, and address and can request many invoices.
- d) **Products** have descriptions and weights and are supplied by **vendors**. Each product has a unique name for a particular vendor.
- e) A **vendor** has an id and an address.

ER Design Question #3

Construct a fish store database where:

- a) A fish store maintains a number of **tanks**, each with a number, name, volume and color.
- b) Each tank contains a number of **fish**, each with an id, name, color, and weight.
- c) Each fish is of a particular **species**, which has a id, name, and preferred food.
- d) Each individual fish has a number of **events** in its life, involving a date and a note relating to the event.

总结

- a) **概念数据库设计是一种对需求的高级归纳和抽象**，它涉及实体，关系，属性三个概念—— entities, relationships, and attributes.
- b) ER建模中有三个概念，两个约束，以及概念与约束的表达符号。
- c) 实体，关系都有类型和实例两个概念。类型对应于编程中的 class，实例对应于class的实例对象。表的模式对应于class的定义，表中的行对应于实例。