第4章: 概念数据库设计

Conceptual Database Design

邹兆年

哈尔滨工业大学
计算机科学与技术学院
海量数据计算研究中心
电子邮件: znzou@hit.edu.cn

2020年春

教学内容[1]

1. 数据库设计的过程
2. 实体-联系模型(ER模型)
   - 与实体相关的概念
   - 与联系相关的概念
3. 实体-联系图(ER图)
4. 数据库概念设计案例
5. 增强实体-联系图(EER图)

[1]课件更新于2020年3月10日

4.1 数据库设计的过程

Overview of Database Design Process

数据库设计的过程

- 数据密集型应用的设计有两个主要任务
  - 数据库设计: 设计数据库的模式(本课程的内容)
  - 应用程序设计: 设计访问数据库的应用程序(软件工程课的内容)
- 数据库设计的过程
  1. 概念设计(conceptual design): 设计数据库的(抽象)概念模式 (第4章)
  2. 逻辑设计(logical design): 根据所使用的DBMS，设计数据库的逻辑模式 (第5章)
  3. 物理设计(physical design): 根据性能需求和应用访问数据库的特点，设计数据库的物理模式(内模式) (第6章)

4.2 实体-联系模型

The Entity-Relationship (ER) Model

实体-联系模型(ER模型)

- 实体-联系模型(the entity-relationship model): 简称ER模型，是概念数据库设计阶段使用的一种重要概念模型，用于将现实世界抽象为实体(entity)及实体间的联系(relationship)
- ER模型提供了数据建模所需的多种概念
  - 与实体相关的概念
  - 与联系相关的概念
- 实体-联系图(the entity-relationship diagram): 简称ER图，是ER模型的一种图形化表示方法

ER模型概念

- 与实体相关的概念
  - 实体
  - 属性
  - 键
  - 实体型、实体集
  - 弱实体型
- 与联系相关的概念
  - 联系
  - 联系型、联系集
  - 联系型的约束
  - 联系型的属性
  - 多元联系

例: COMPANY数据库设计需求

- The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.
- Each department controls a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.
- We store each EMPLOYEE's social security number (SSN), address, salary, sex, and birthdate.
  - Each employee works for one department but may work on several projects.
  - We keep track of the number of hours per week that an employee currently works on each project.
  - We also keep track of the direct supervisor of each employee.
- Each employee may have a number of DEPENDENTs.
  - For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

4.2.1 与实体相关的概念

Concepts Related to Entities

实体(Entity)与属性(Attribute)

- 实体(entity): 数据库中表示的现实世界中的具体对象或事物
  - 例: 雇员John Smith、科研部、ProductX项目
- 属性(attribute): 用于刻画实体的特性
  - 例: 雇员实体具有姓名、性别、出生日期、身份证号、住址等属性
  - 每个实体的每个属性都具有一个值，全部属性值共同刻画了该实体
  - 每个属性都关联着它的取值域(或数据类型)

属性的类型

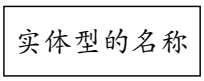
- 简单属性(simple attribute): 具有原子(atomic)属性值的属性，其属性值不可再分
  - 例如，性别、身份证号、民族等
- 复合属性(composite attribute): 由多个成分构成的属性
  - 姓名→ (姓，名)
  - 地址→ (省，市，区，街道，门牌号，邮编)
  - 复合属性的某个成分还可以是复合的
- 多值属性(multi-valued attribute): 一个实体可具有多个值的属性
  - 例如，电话号码、论文作者等，记作{电话号码}、{论文作者}
- 派生属性(derived attribute): 由其他属性派生出来的属性
  - 例如，年龄可根据出生日期算出
- 复合属性和多值属性可以是相互嵌套的
  - 例如，已取得的学位是一个复合多值属性，该属性可具有多个值，每个值由4部分构成，记作{(学校，专业，学位，日期)}

实体型与实体集

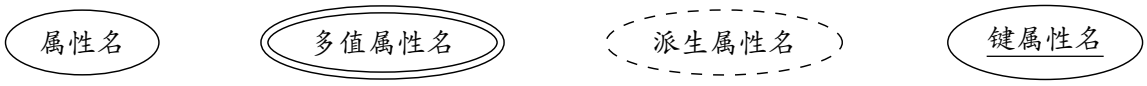
- 实体型(entity type): 具有相同属性的实体共同具有的类型
- 键属性(key attribute): 同一实体型的任意实体都具有不同值的属性
  - 例如，雇员实体型的身份证号属性
  - 键属性可以是复合属性
  - 一个实体型可以具有多个键属性
- 实体集(entity set): 当前存储在数据库中的某实体型的实例的集合
- 实体 vs. 实体型 vs. 实体集
  - 实体: 对象
  - 实体型: 类型
  - 实体集: 对象集合

实体型的ER图表示

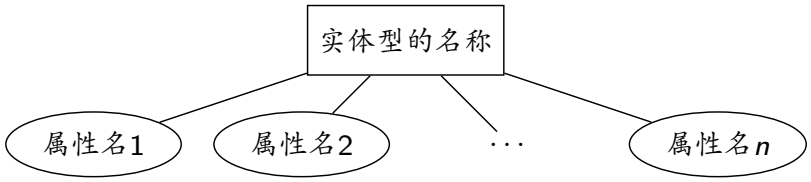
- 实体型表示为矩形



- 属性表示为椭圆
  - 简单属性表示为实线椭圆
  - 多值属性表示为双实线椭圆
  - 派生属性表示为虚线椭圆
  - 键属性的属性名加下划线

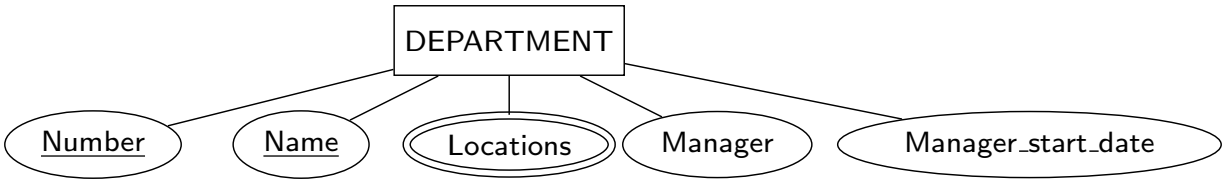


- 实体型与其属性用线连接起来



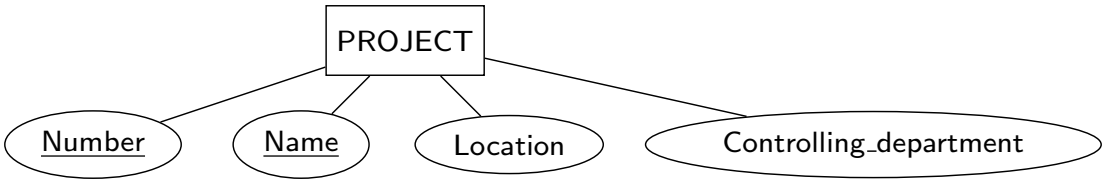
案例: COMPANY数据库的初步设计

The company is organized into DEPARTMENTs. Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager. A department may have several locations.



案例: COMPANY数据库的初步设计(续)

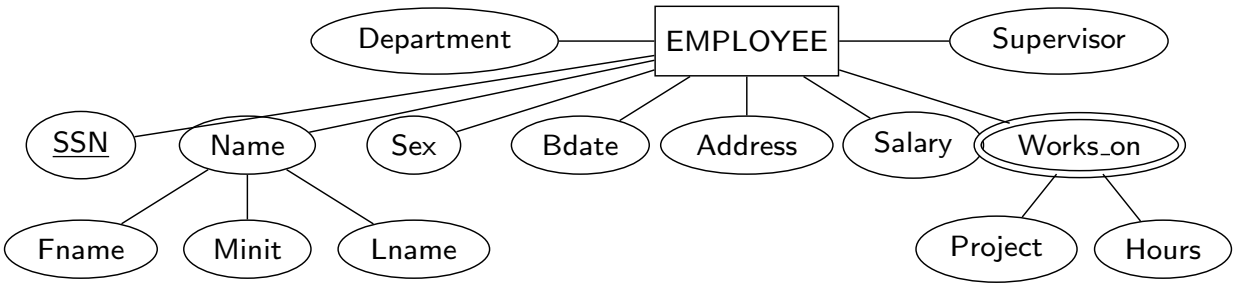
Each department *controls* a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.



案例: COMPANY数据库的初步设计(续)

We store each EMPLOYEE's social security number (SSN), address, salary, sex, and birthdate.

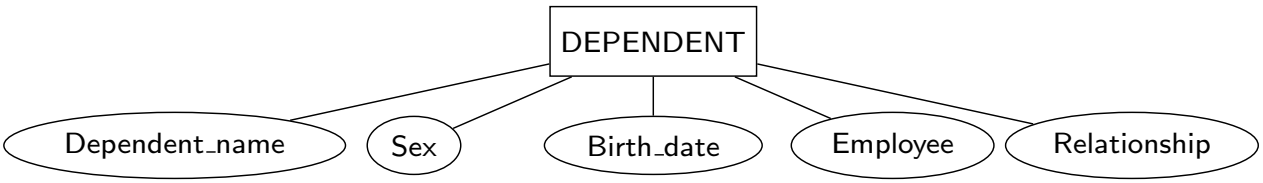
- Each employee *works for* one department but may *work on* several projects.
- We keep track of the number of hours per week that an employee currently works on each project.
- We also keep track of the *direct supervisor* of each employee.



案例: COMPANY数据库的初步设计(续)

Each employee may *have* a number of DEPENDENTs.

- For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.



案例: COMPANY数据库的初步设计(续)

- 初步设计方案并不完善: 只有实体
- 需求中很多内容需要表示为实体间的联系

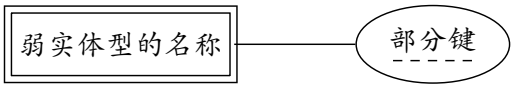
弱实体型、弱实体集

等讲完与联系相关的概念后再讲 ▸▸ 与联系相关的概念

- 弱实体型(weak entity type): 没有键属性的实体型
  - EMPLOYEE John Smith's dependent has no SSN
  - DEPENDENT是一个弱实体
- 标识实体型(identifying entity type)/属主实体型(owner entity type): 由于弱实体型没有键属性，需要依赖于其他实体型进行区分
  - John Smith's dependent is identified by John Smith
  - EMPLOYEE是DEPENDENT的标识实体型
- 标识联系型(identifying relationship type): 弱实体型与其标识实体型通过标识联系型关联
  - DEPENDENT和EMPLOYEE实体型通过标识联系型DEPENDENT_OF联系起来
- 部分键(partial key): 用于区分和同一标识实体相关联的弱实体的属性集合
  - John Smith has several dependents who have different names
  - DEPENDENT实体型的Name属性是部分键
- 用"弱实体型的部分键+标识实体型的主键"来区分不同弱实体

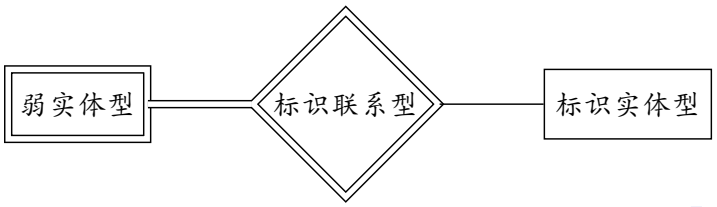
弱实体型的ER图表示

- 弱实体型表示为双实线矩形，部分键加虚下划线



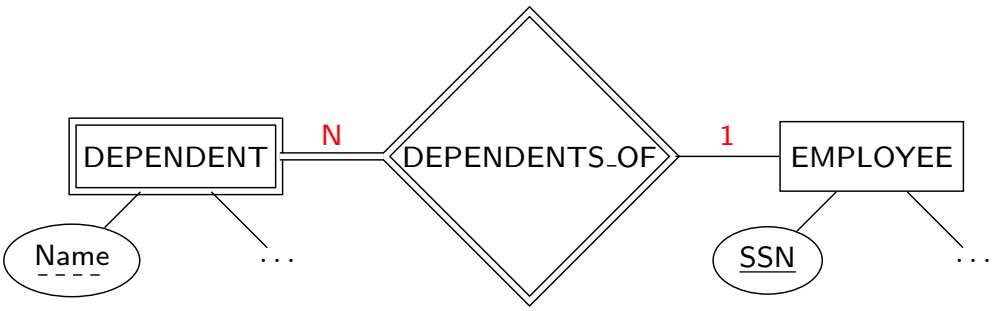
- 标识联系型表示为双实线菱形

- 弱实体型和标识实体型分别由直线连到标识联系型上
  - 弱实体型全部参与到标识联系型中(为什么?)



案例: COMPANY数据库设计的细化

- Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.



4.2.2 与联系相关的概念

Concepts Related to Relationships

联系、联系型、联系集

- 联系(relationship): 一个联系表示多个实体之间有意义的关联关系
  - EMPLOYEE John Smith works on the ProductX PROJECT
  - EMPLOYEE Franklin Wong manages the Research DEPARTMENT
- 联系型(relationship type): 同一类联系共同具有的类型
  - The WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate
  - The MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate
- 联系型的度(degree): 参与到一个联系型中的实体型的个数
  - 联系型WORKS_ON和MANAGES的度都是2
  - 度为2的联系称作二元联系(binary relationship)
  - 度为3的联系称作三元联系(ternary relationship)
  - 度为n的联系称作n元联系(n-ary relationship)
- 联系集(relationship set): 数据库中当前存储的联系型的实例的集合

联系、联系型、联系集(续)

- 实体型: EMPLOYEE, PROJECT
- 实体集: $\{e_1, e_2, \ldots\}$, $\{p_1, p_2, \ldots\}$
- 联系型: WORKS_ON
- 联系集: $\{r_1, r_2, \ldots\}$
- WORKS_ON是二元联系型，参与实体型有EMPLOYEE和PROJECT
- 实体e_1和p_1参与到联系r_1中，实体e_2和p_1参与到联系r_2中，...

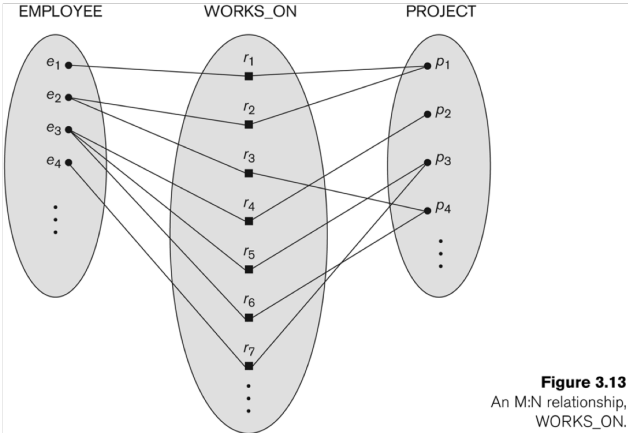
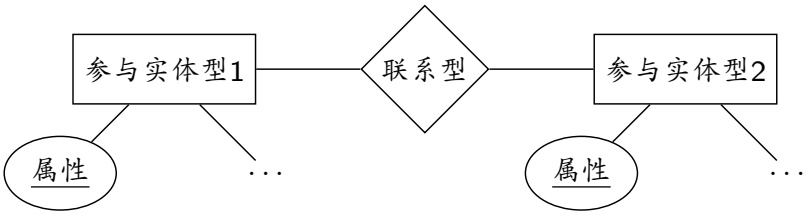


Figure 3.13
An M:N relationship, WORKS_ON.

联系型的ER图表示

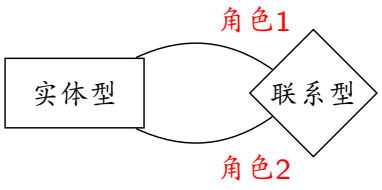
- 联系型表示为菱形

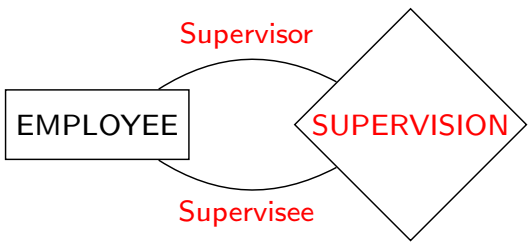
- 用直线将联系型与参与到联系型中的实体型连接起来



联系型的ER图表示(续)

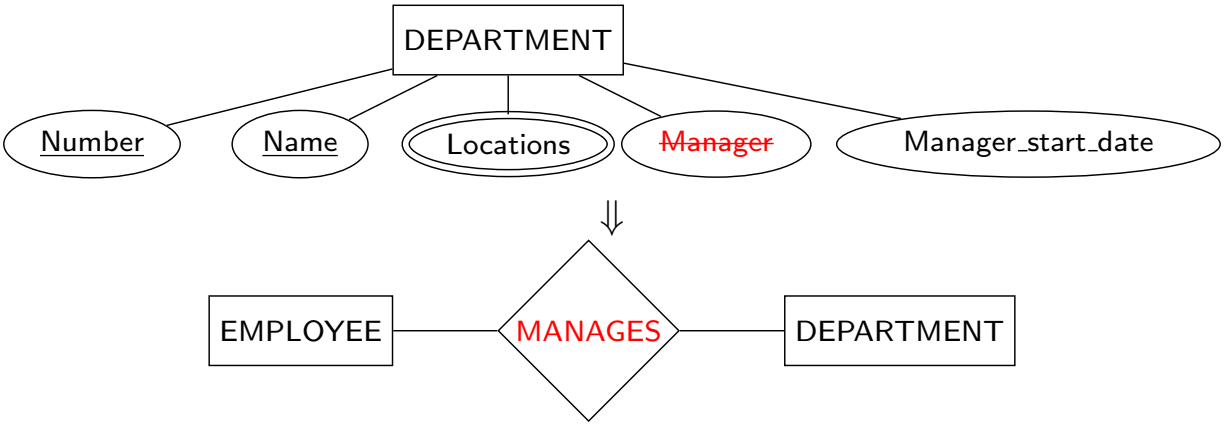
- 相同的参与实体型之间可以存在多种不同的联系型
- 一个实体型可以和自身参与到同一个联系型中
  - EMPLOYEE Jill is the direct supervisor of EMPLOYEE Jack
  - 该实体型在该联系型中担任不同的角色(role)
  - 需要在ER图中标出实体型参与在联系型中的角色





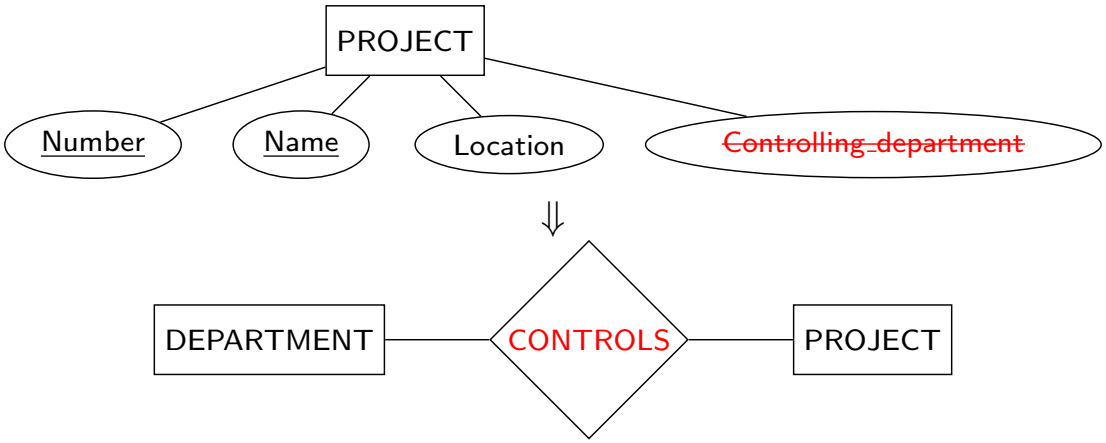
案例: COMPANY数据库设计的细化

- The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.



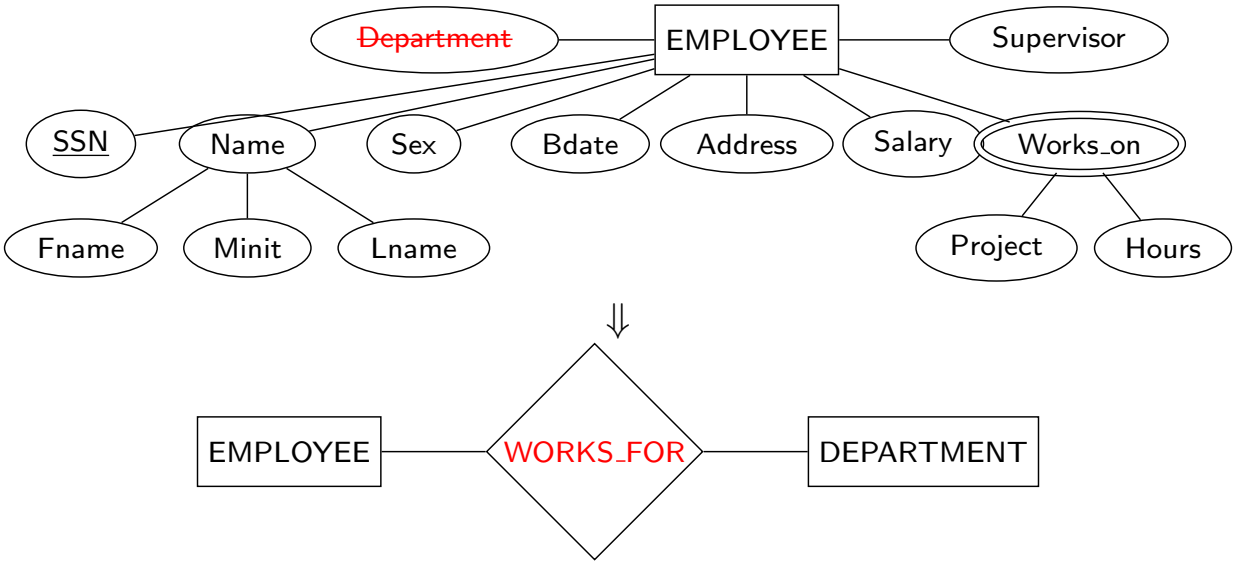
案例: COMPANY数据库设计的细化(续)

- Each department controls a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.



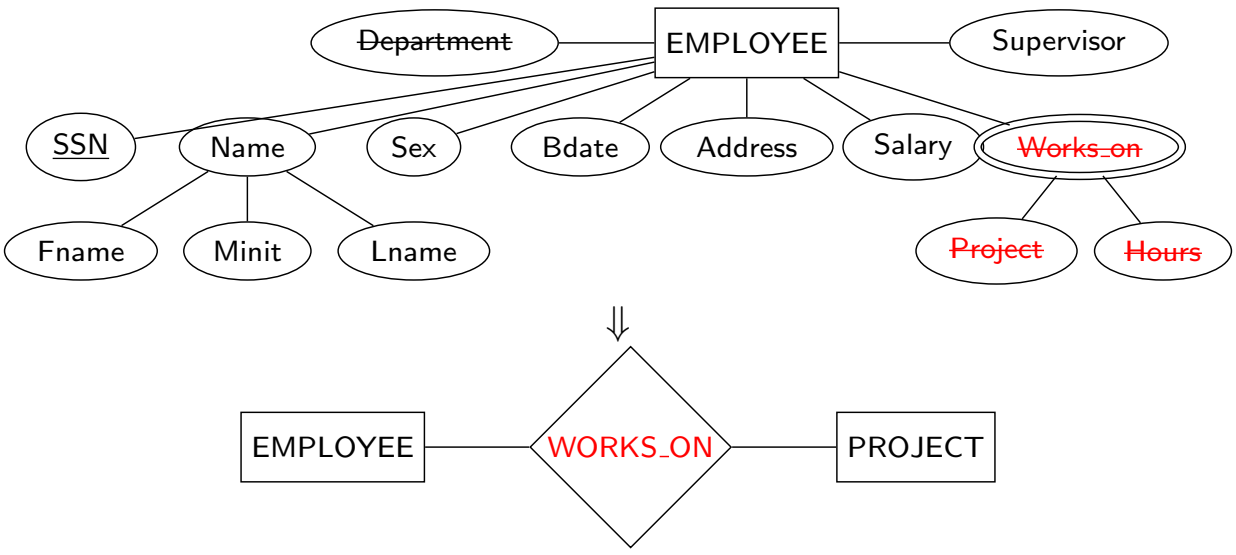
案例: COMPANY数据库设计的细化(续)

- Each employee works for one department but may work on several projects.



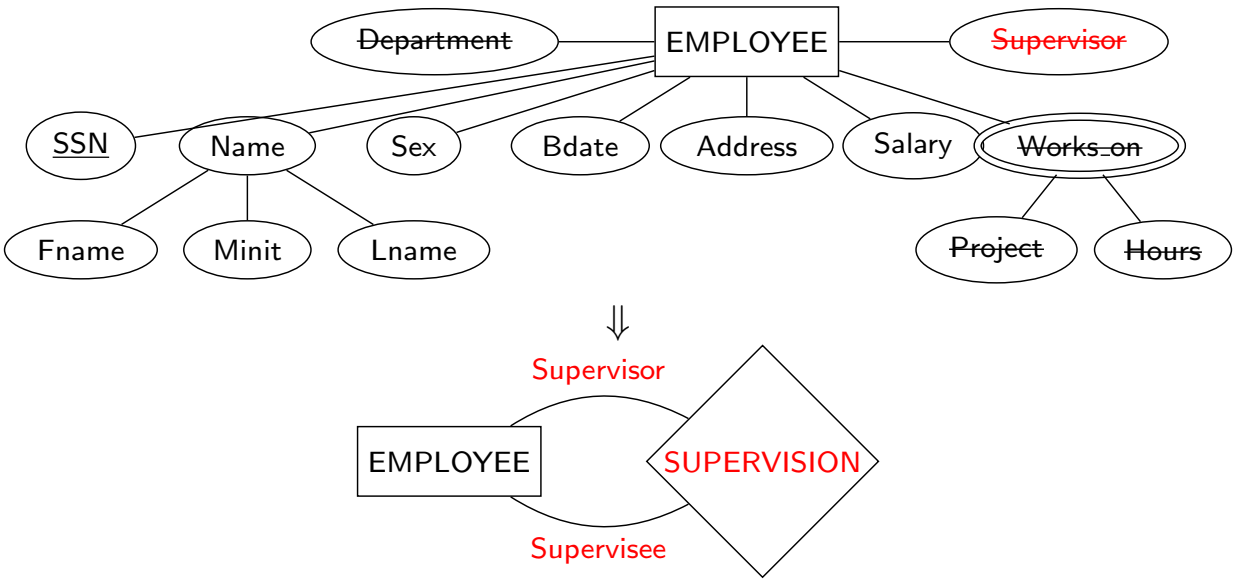
案例: COMPANY数据库设计的细化(续)

- Each employee works for one department but may work on several projects.



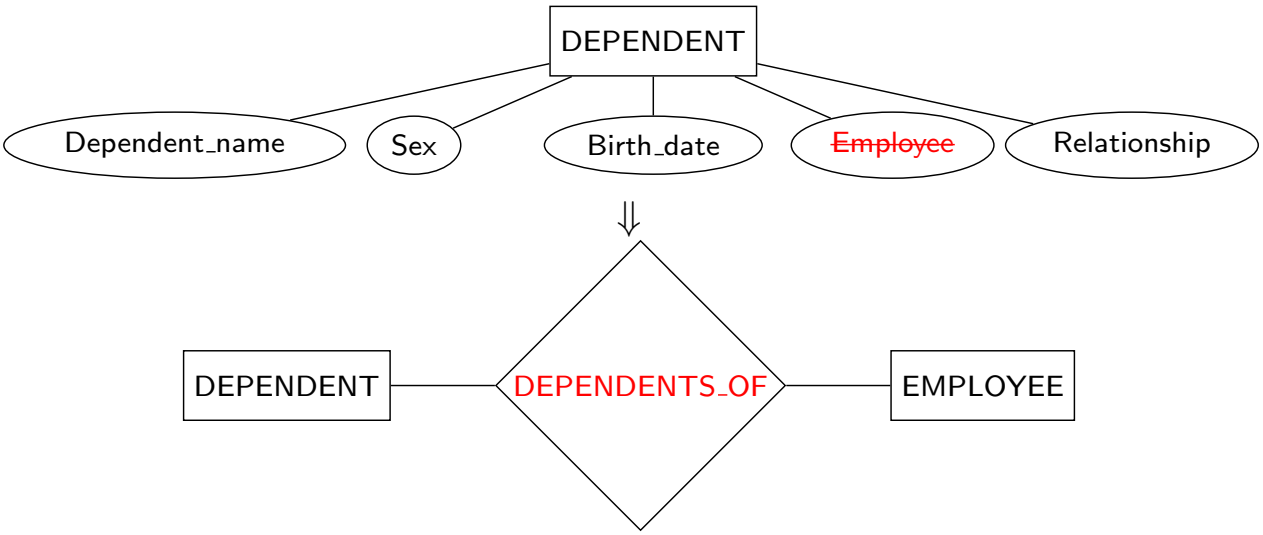
案例: COMPANY数据库设计的细化(续)

- We also keep track of the direct supervisor of each employee.



案例: COMPANY数据库设计的细化(续)

- Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

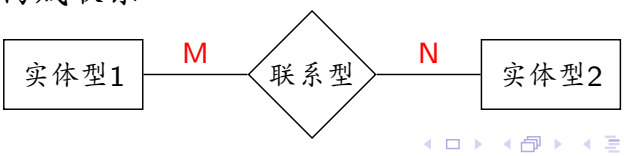


联系型的约束

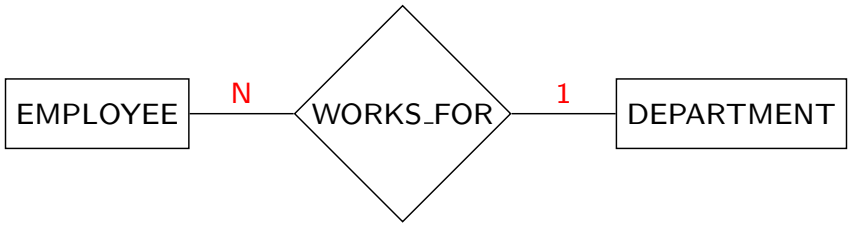
- 基数比(cardinality ratio): 刻画实体型参与到联系型中的最大基数
  - 1对1 (1:1): 对实体型1的任意实体e，最多只有1个实体型2的实体与e构成联系；对实体型2的任意实体e'，最多只有1个实体型1的实体与e'构成联系

  - 多对1 (N:1): 对实体型1的任意实体e，最多只有1个实体型2的实体与e构成联系；对实体型2的任意实体e'，最多有$N > 1$个实体型1的实体与e'构成联系

  - 多对多(M:N): 对实体型1的任意实体e，最多有$N > 1$个实体型2的实体与e构成联系；对实体型2的任意实体e'，最多有$M > 1$个实体型1的实体与e'构成联系



联系型的约束(续)



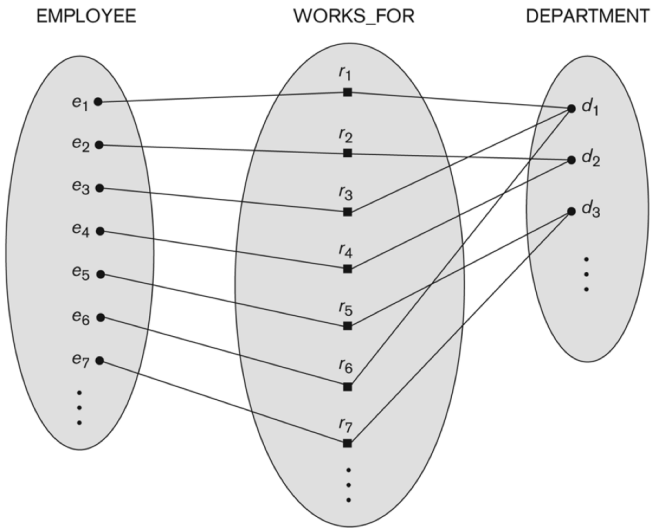
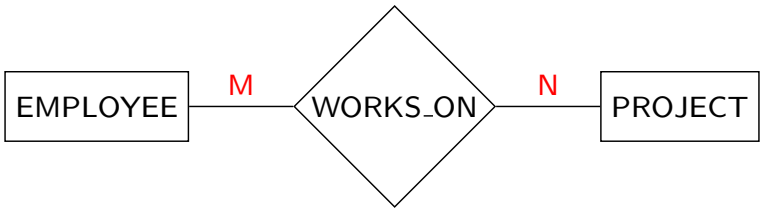


Figure 3.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

联系型的约束(续)



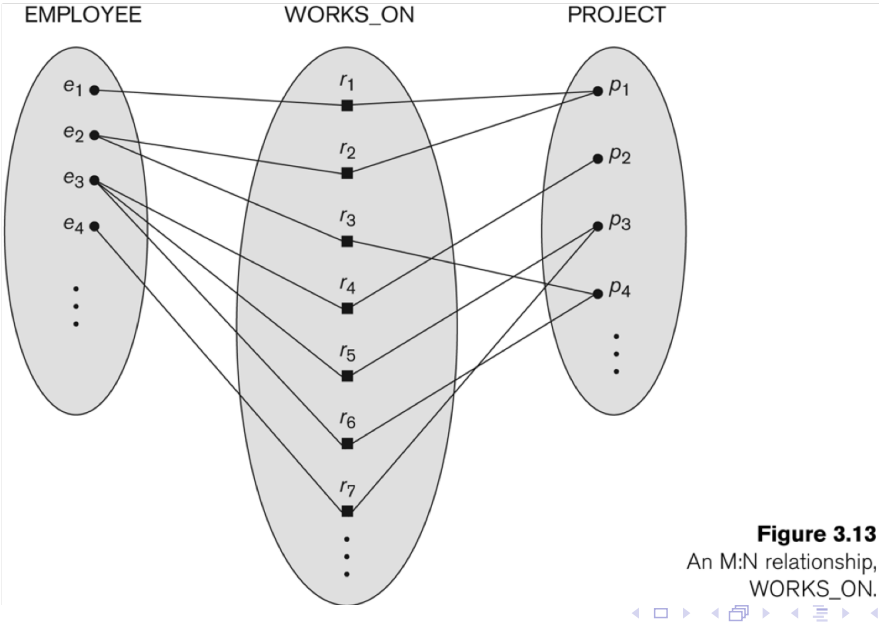
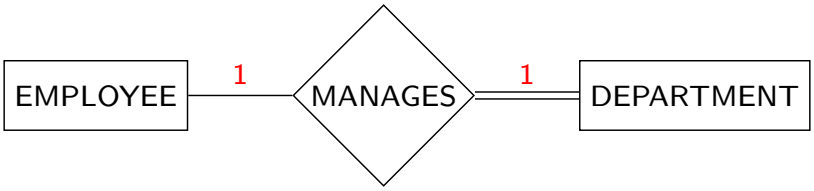


Figure 3.13
An M:N relationship, WORKS_ON.

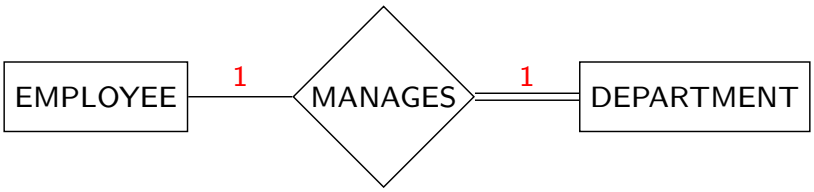
联系型的约束(续)

- 存在依赖约束(existence dependency constraint)/参与度约束(participation constraint): 刻画实体型参与到联系型中的最小基数(即一个实体最少参与到几个联系中)
  - 0个(部分参与): 在ER图中表示为单线
  - ≥ 1个(全部参与): 在ER图中表示为双线

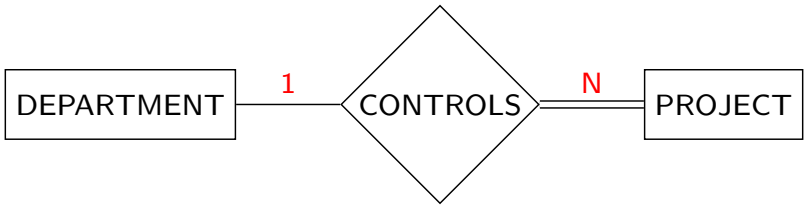


案例: COMPANY数据库设计的细化

- The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.

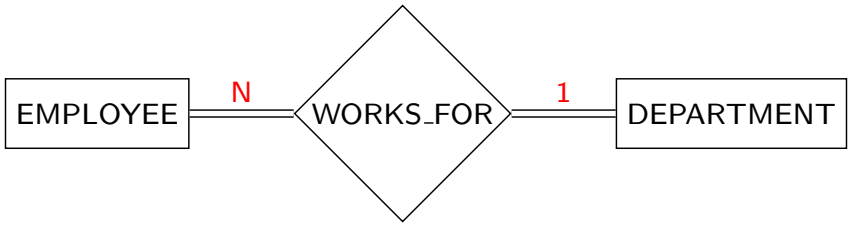


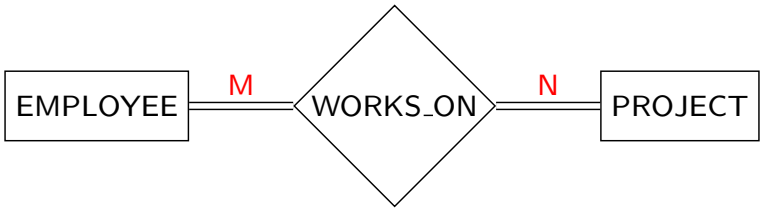
- Each department controls a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.



案例: COMPANY数据库设计的细化(续)

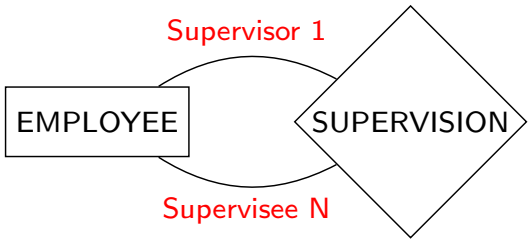
- Each employee works for one department but may work on several projects.



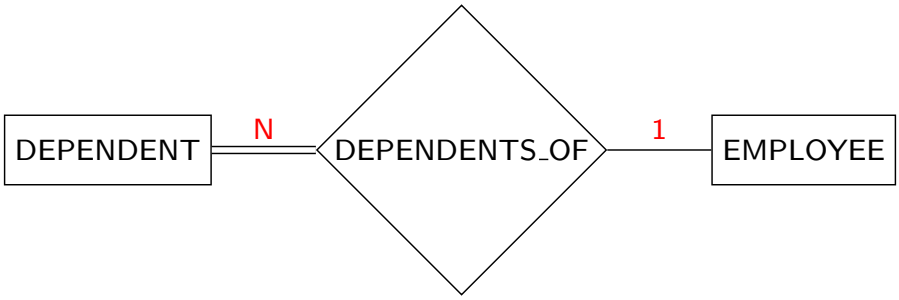


案例: COMPANY数据库设计的细化(续)

- We also keep track of the direct supervisor of each employee.

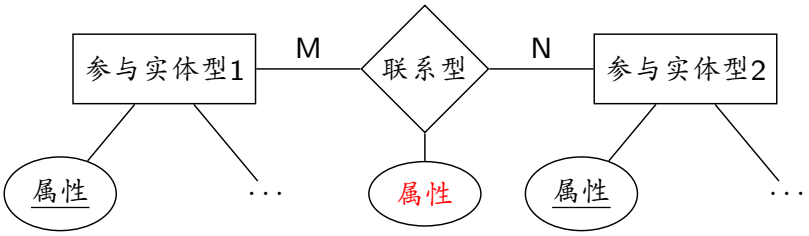


- Each employee may have a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

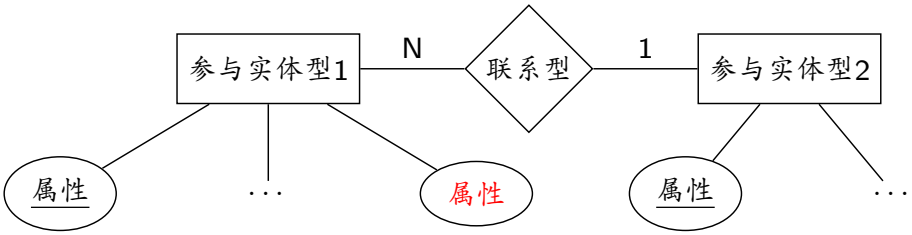


联系型的属性

- 联系型可以具有属性
  - EMPLOYEE John Smith works on the ProducX Project 40 hours/week
- 在ER图中，用直线将联系型与联系型的属性连接起来

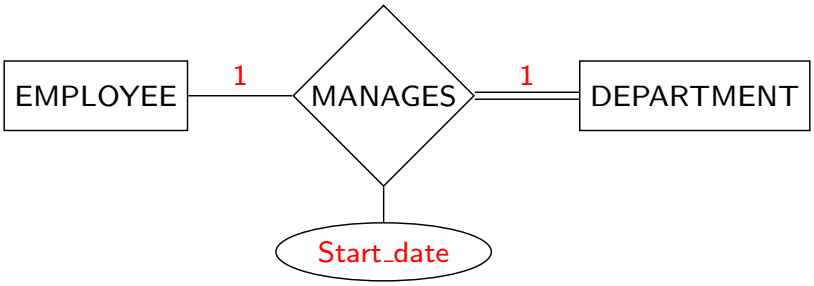


- 联系型的属性主要出现在M:N联系型中
- N:1联系型的属性可以被移到N一方的实体型中



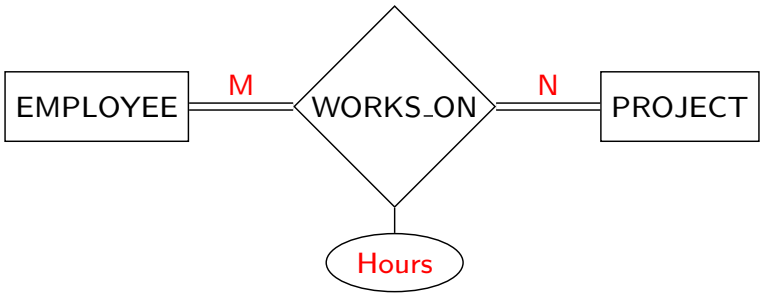
案例: COMPANY数据库设计的细化

- The company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. A department may have several locations.



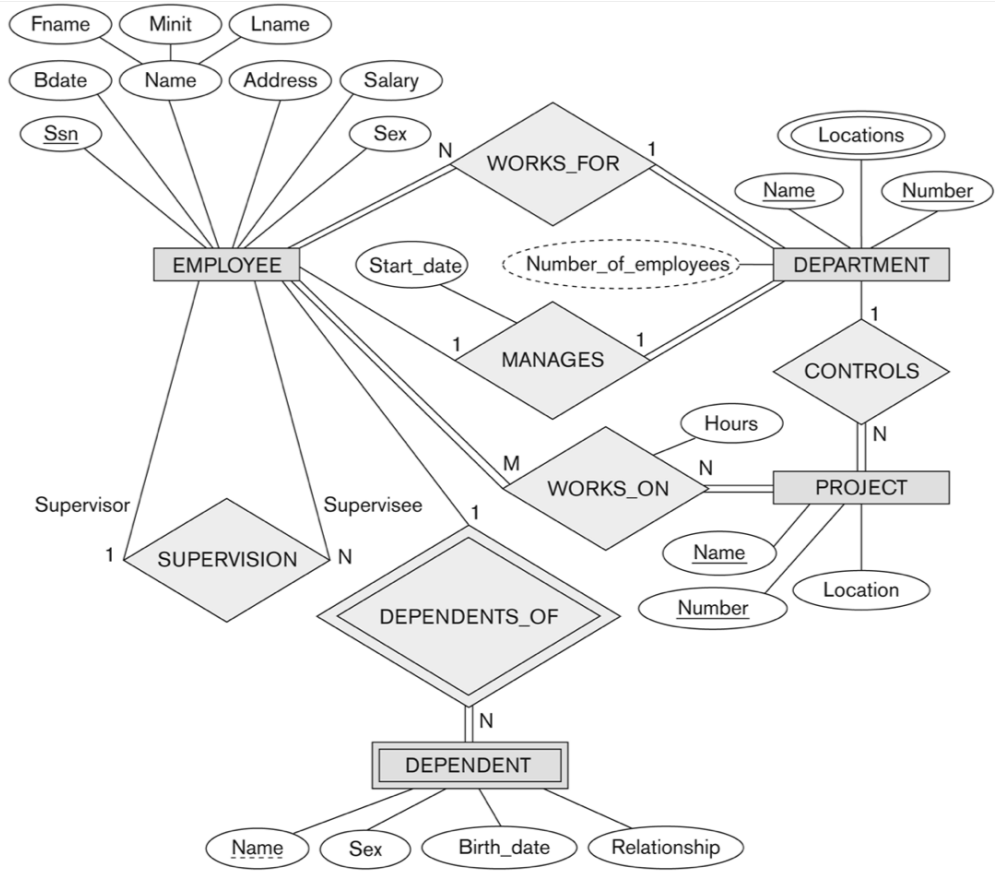
案例: COMPANY数据库设计的细化(续)

- We keep track of the number of hours per week that an employee currently works on each project.



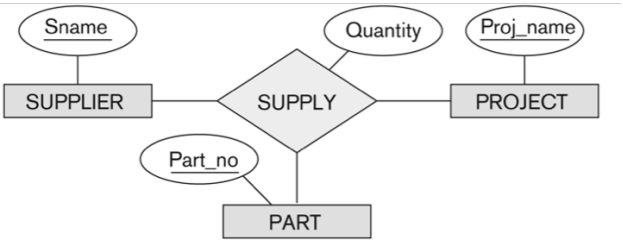
转到弱实体型 弱实体型

案例: COMPANY概念数据库设计

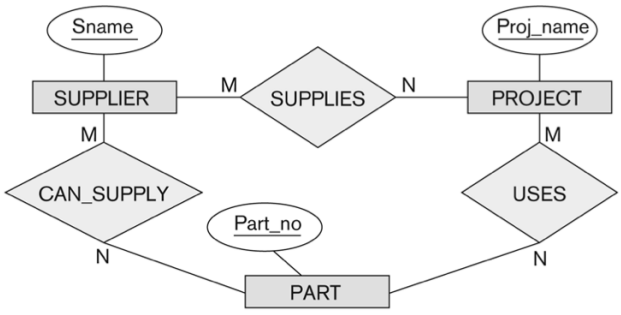


多元联系(Higher-degree Relationships)

- 多元联系(higher-degree relationships): 3个以上实体参与的联系
- 一个n元联系通和n个二元联系所表示的意义通常是不同的



1个三元联系型



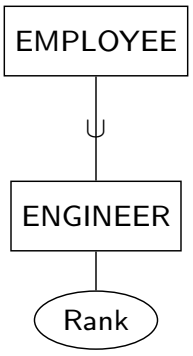
3个二元联系型

4.3 增强ER模型

Enhanced ER Model

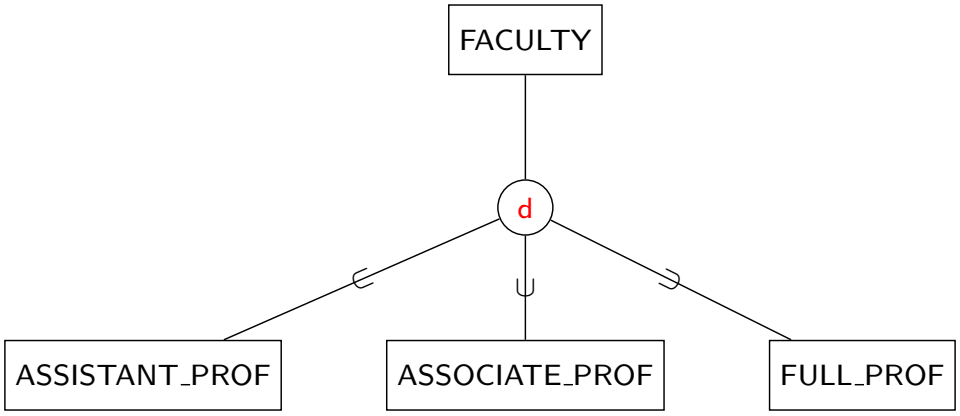
子类(subclass)/超类(superclass)

- 实体型E的一部分实体构成了一个有特殊含义的子集，这部分实体的实体型E'称为E的子类(subclass)，同时E称为E'的超类(superclass)
  - 超类: EMPLOYEE
  - 子类: Engineer, DBA
- 子类继承了父类的全部属性及父类所参与的全部联系型
- 子类/超类联系的表示方法:



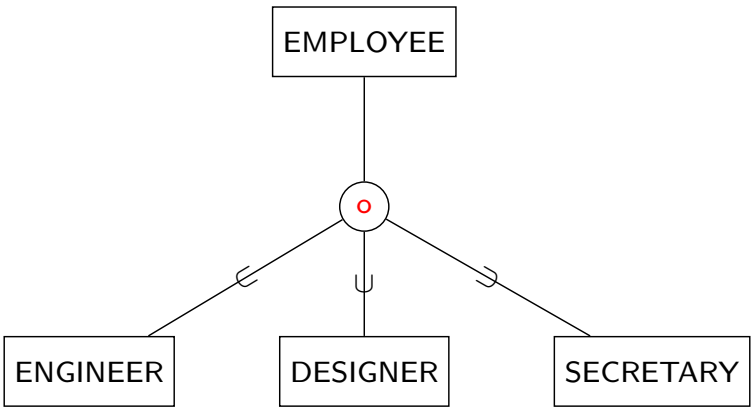
不相交子类

- 如果超类的每个实体属于最多一个子类，则子类是不相交的(disjoint)
  - 超类: FACULTY
  - 子类: ASSISTANT_PROF, ASSOCIATE_PROF, FULL_PROF



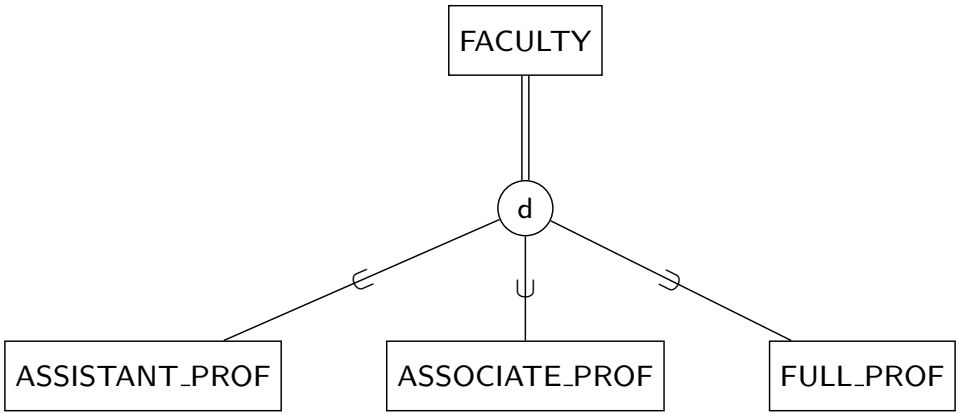
重叠子类

- 如果超类的每个实体可以属于多个子类，则子类是重叠的(overlap)
  - 超类: EMPLOYEE
  - 子类: ENGINEER, DESIGNER, SECRETARY



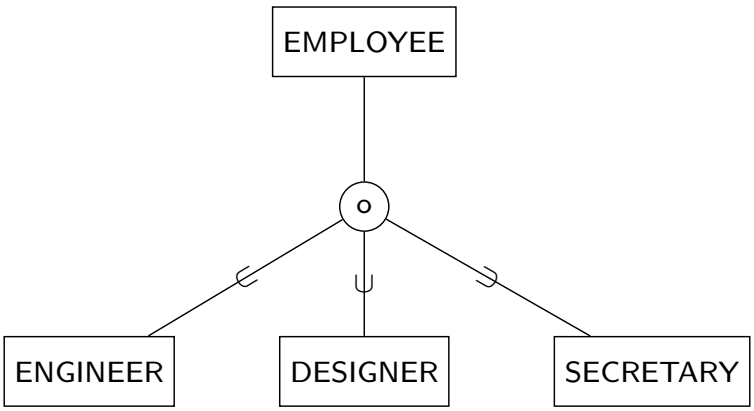
全部特化(Total Specialization)

- 超类的每个实体必须属于至少一个子类
  - 超类: FACULTY
  - 子类: ASSISTANT_PROF, ASSOCIATE_PROF, FULL_PROF



部分特化(Partial Specialization)

- 超类的某些实体可以不属于任何子类
  - 超类: EMPLOYEE
  - 子类: ENGINEER, DESIGNER, SECRETARY



总结

1. 数据库设计的过程: 概念数据库设计→ 逻辑数据库设计→ 物理数据库设计
2. 实体-联系模型(ER模型)
   - 与实体相关的概念
     - ★ 实体
     - ★ 属性: 简单属性、复合属性、多值属性、派生属性
     - ★ 键属性
     - ★ 实体型、实体集
     - ★ 弱实体型: 标识实体型、标识联系型、部分键
   - 与联系相关的概念
     - ★ 联系
     - ★ 联系型、联系集
     - ★ 联系型的约束: 基数比(1:1、N:1、M:N)、参与度约束(全部参与、部分参与)
     - ★ 联系型的属性
     - ★ 多元联系
3. 实体-联系图(ER图)[2]
4. 增强实体-联系图(EER图)

[2]课件中的ER图使用Tikz绘制http://www.texample.net/tikz/examples/er-diagram/

习题

1. 数据库设计过程分为哪些阶段? 为什么不宜从需求出发直接在DBMS上创建关系?
2. 说明实体、实体型、实体集的区别
3. 为什么弱实体型全部参与到标识联系型中?
4. 举例说明1个三元联系通和3个二元联系表示的意义通常是不同的
5. 什么情况下1个三元联系通和3个二元联系表示的意义是相同的?
6. Design a database for a bank, including information about customers and their accounts.
   - Information about a customer includes their name, address, phone, and Social Security number (SSN).
   - A customer can have a set of addresses (which are street-city-state triples), and at each address there is a set of phones.
   - Accounts have numbers, types (e.g., savings, checking) and balances. Also record the customer who own an account.
   - An account can have only one customer.

致谢

- 感谢李一鸣同学指出了课件中的错误