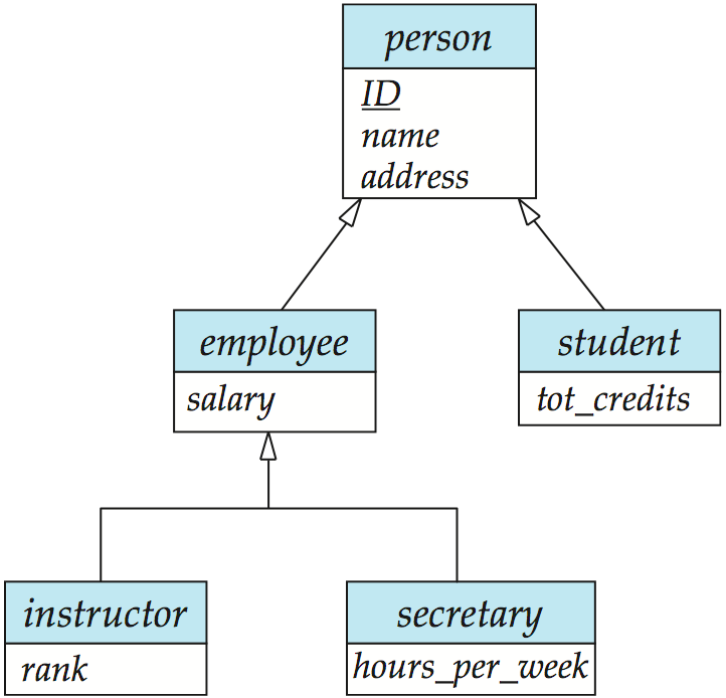
扩展的E-R特性

- 特化
  - 自顶向下设计过程中，确定实体集中的一个具有特殊性质的子集
  - 这些子集称为低层实体集，它们具有特殊的属性或者参加特殊的联系
  - 属性继承 ：低层实体集继承它连接的高层实体集的所有属性及参加的联系

扩展的E-R特性

- 特化用从特化实体指向另一方实体的空心箭头来表示。这种关系为ISA关系，代表“is a”（“是一个”）。例如，一个教师“是一个”雇员

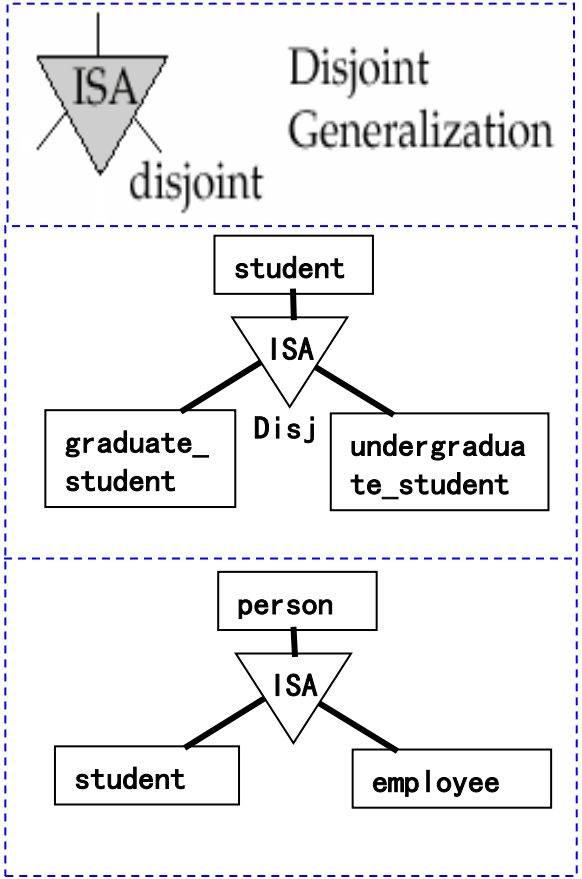


扩展的E-R特性

- 概化
  - 自底向上设计过程中，将若干共享相同特性的实体集组合成一个高层实体集
  - 特化与泛化简单互逆：它们在E-R图中以相同方式表示

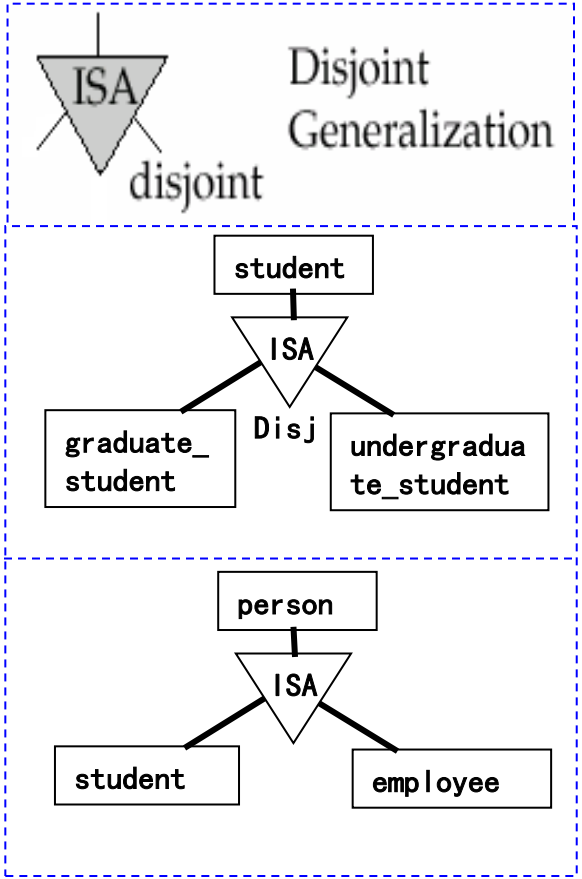
对特化/概化的设计约束

- 关于哪些实体可以是给定低层实体集的成员的约束
  - 条件定义的
    - 只有满足*student_type*=“研究生”的实体才允许属于*graduate_student*实体
  - 用户定义的
    - 大学雇员属于不同的工作组



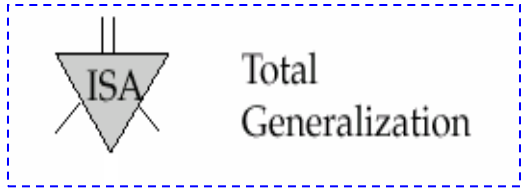
对特化/概化的设计约束

- 关于实体在单个概化中是否可以属于多于一个低层实体集的约束
  - 不相交
    - 一个实体只能属于一个低层实体集
    - 在E-R图中ISA三角形旁边加注disjoint
  - 重叠
    - 一个实体可以属于多个低层实体集



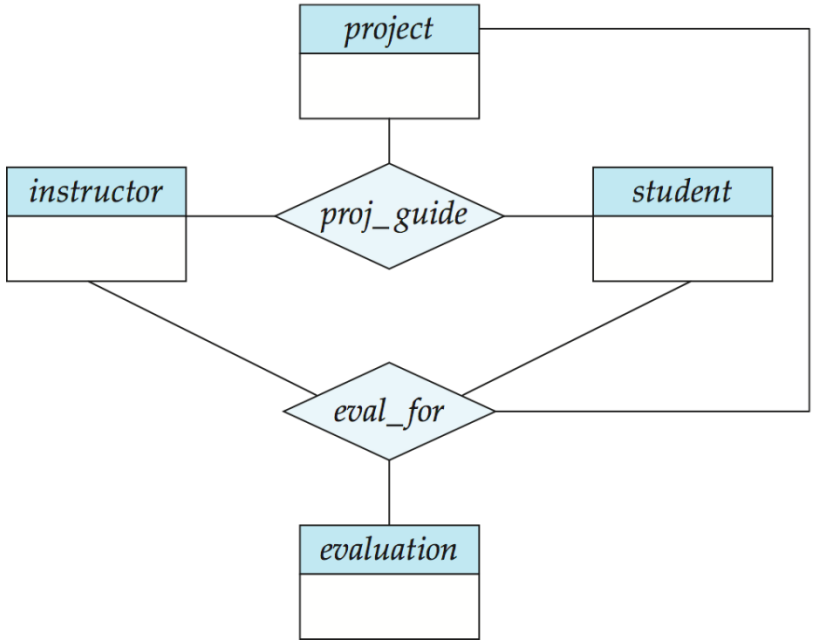
对特化/概化的设计约束

- 完备性约束：说明高层实体集中的实体是否必须至少属于一个低层实体集
  - 全部概化或特化：每个高层实体必须属于一个低层实体集
  - 部分概化或特化：允许一些高层实体不属于任何低层实体集（默认的）



扩展的E-R特性

- 聚集
  - 考虑三元联系*proj_guide*
  - 现在假设每位在项目上指导学生的教师需要记录月评估报告

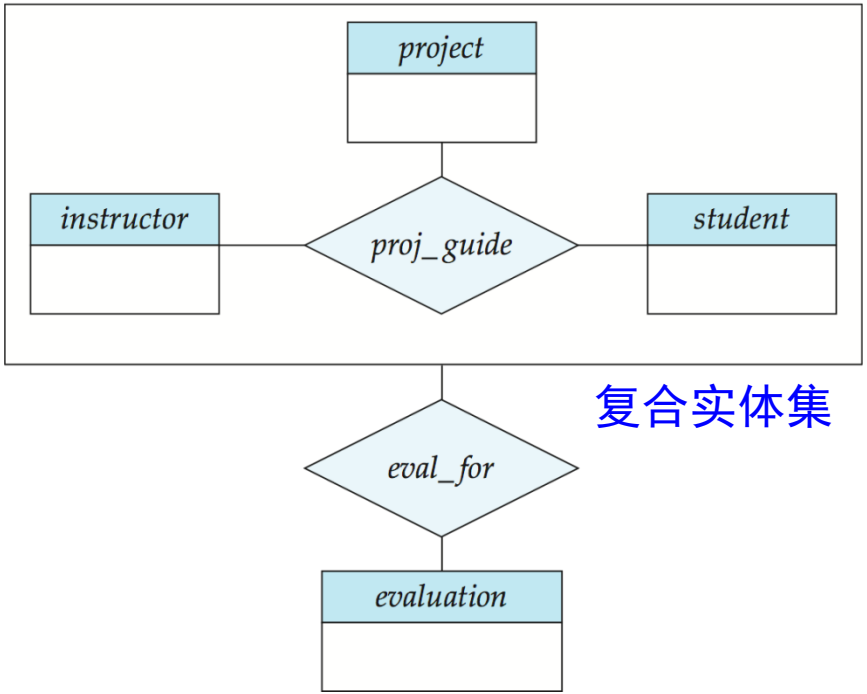


聚集

- 联系集*eval_for*和*proj_guide*表达了重叠信息
  - 每个*eval_for*联系对应一个*proj_guide*联系
  - 然而，某些*proj_guide*联系可能不对应任何*eval_for*联系，因此我们不能丢掉*proj_guide*联系
- 通过聚集消除这种冗余
  - 将联系视为一个抽象实体
  - 从而允许联系之间的联系
  - 联系抽象为新实体

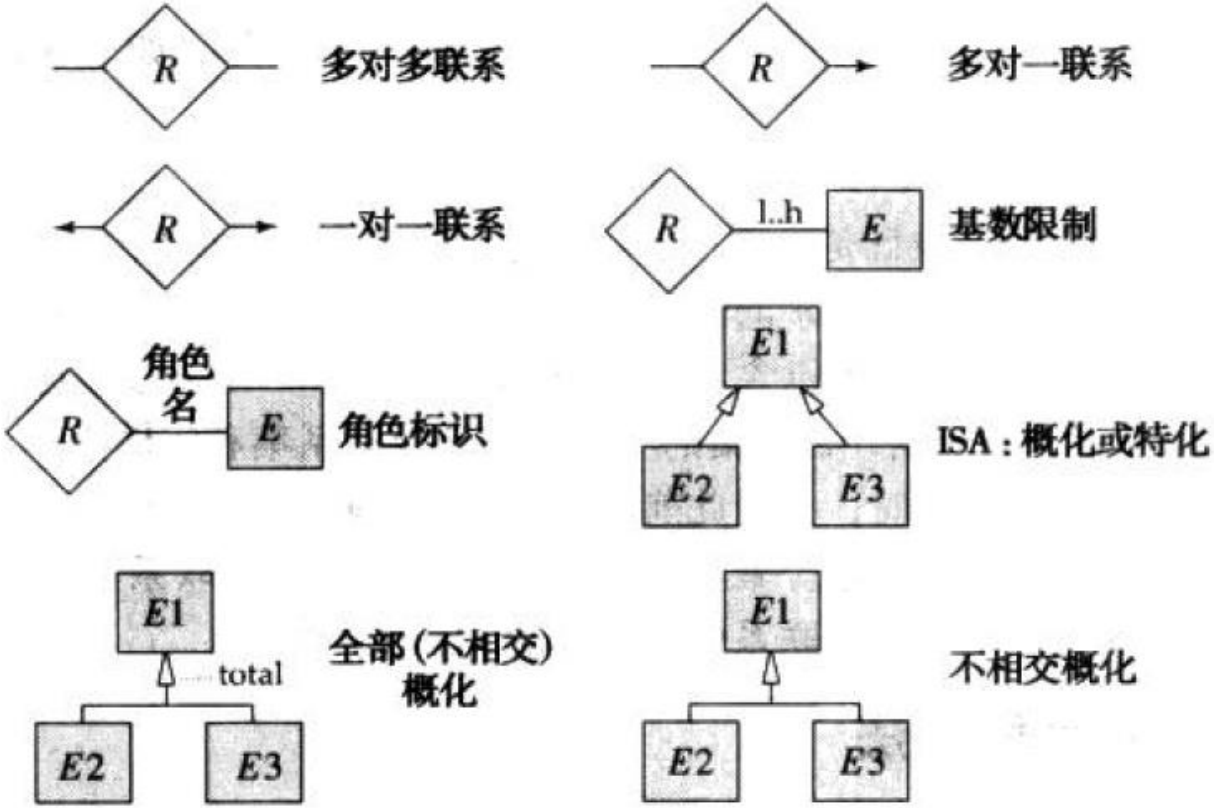
聚集

- 在没有引入冗余的情况下，下图表达了：
  - 一个学生在某个项目上由某个导师指导
  - 一个学生，导师，项目的组合可能有一相关的评估



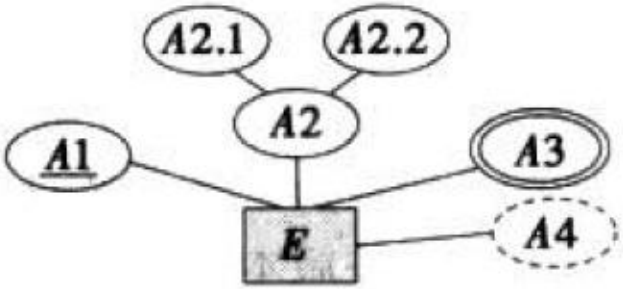
E-R图表示法中使用的符号小结

E-R图表示法中使用的符号小结



其他可选择的E-R图表示法

实体集*E*包含
简单属性*A*1、
复合属性*A*2、
多值属性*A*3、
派性属性*A*4、
以及主码*A*1

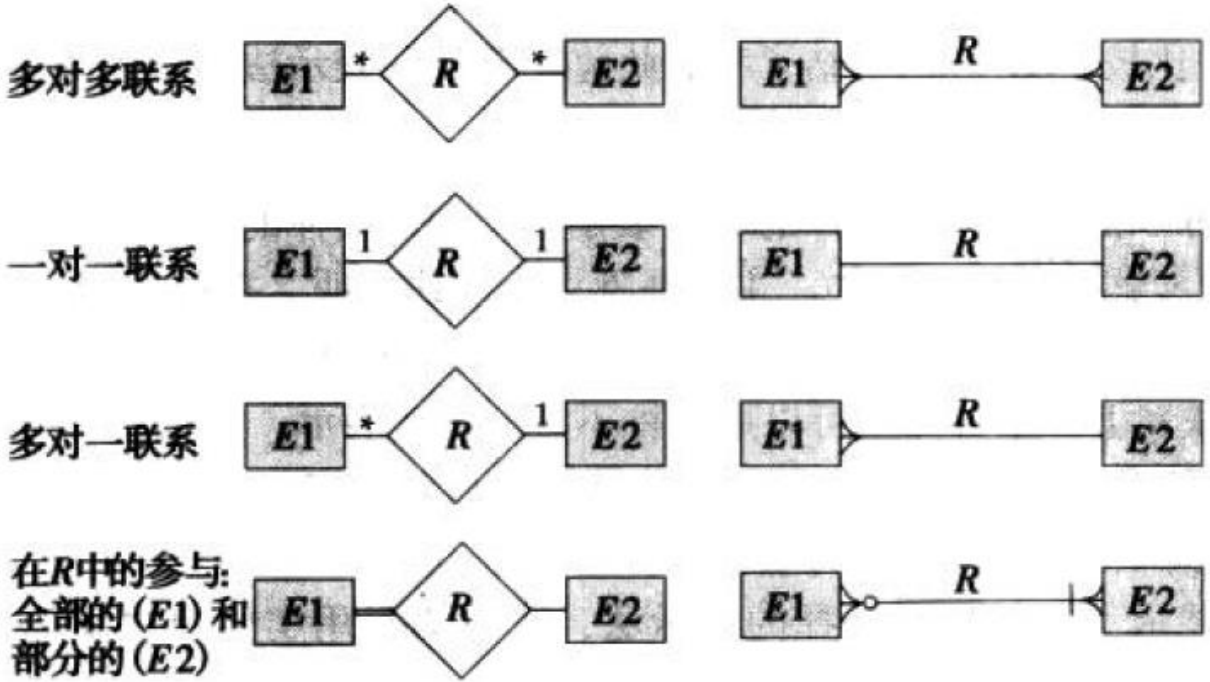


弱实体集

概化

全部概化

其他可选择的E-R图表示法

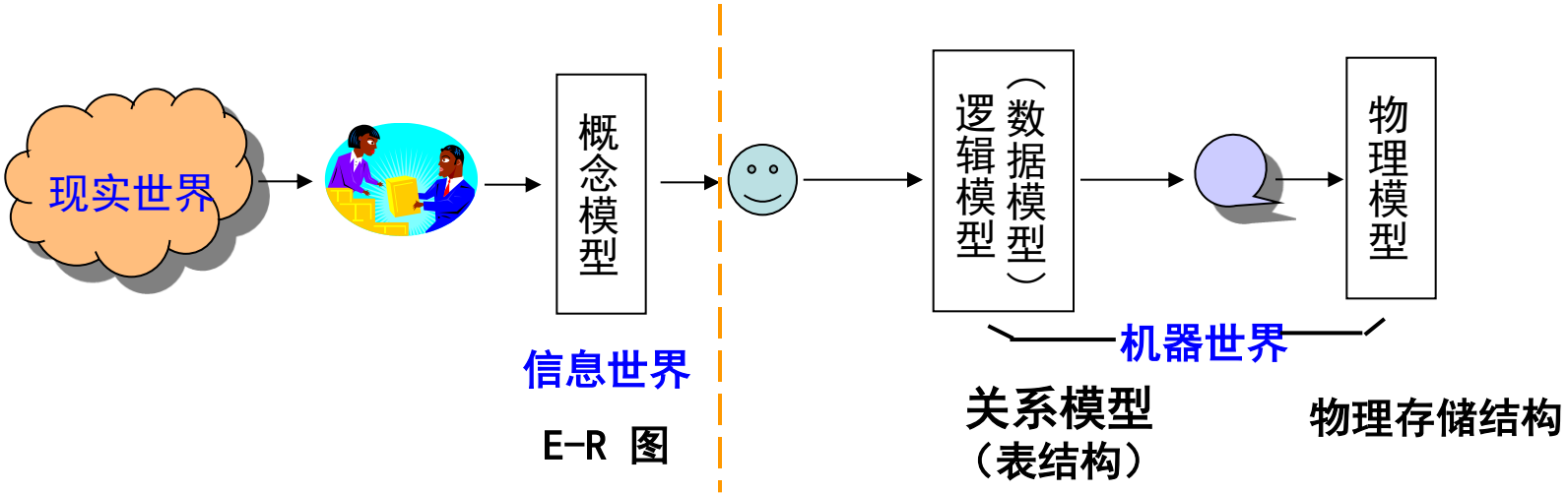


设计数据库的E-R模式

- 需求分析
  - 需要什么样的数据、应用程序和业务
- 概念数据库设计 ★
  - 使用E-R模型或类似的高层次数据模型，描述数据
- 逻辑数据库设计
  - 将概念设计转换为某个DBMS所支持的数据模型
  - 关系标准化，检查冗余和相关的异常关系结构
- 物理数据库设计
  - 索引，集群和数据库调优

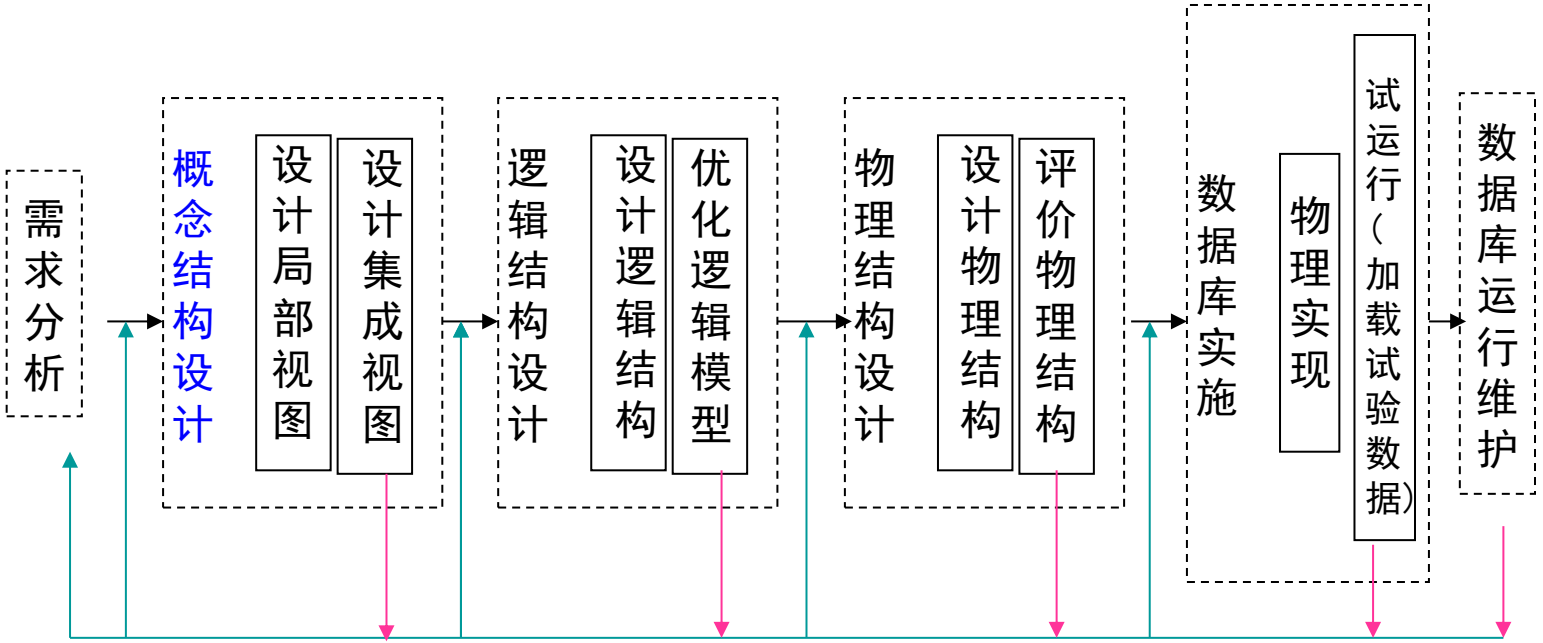
设计数据库的E-R模式

- 数据库的设计步骤（补充）



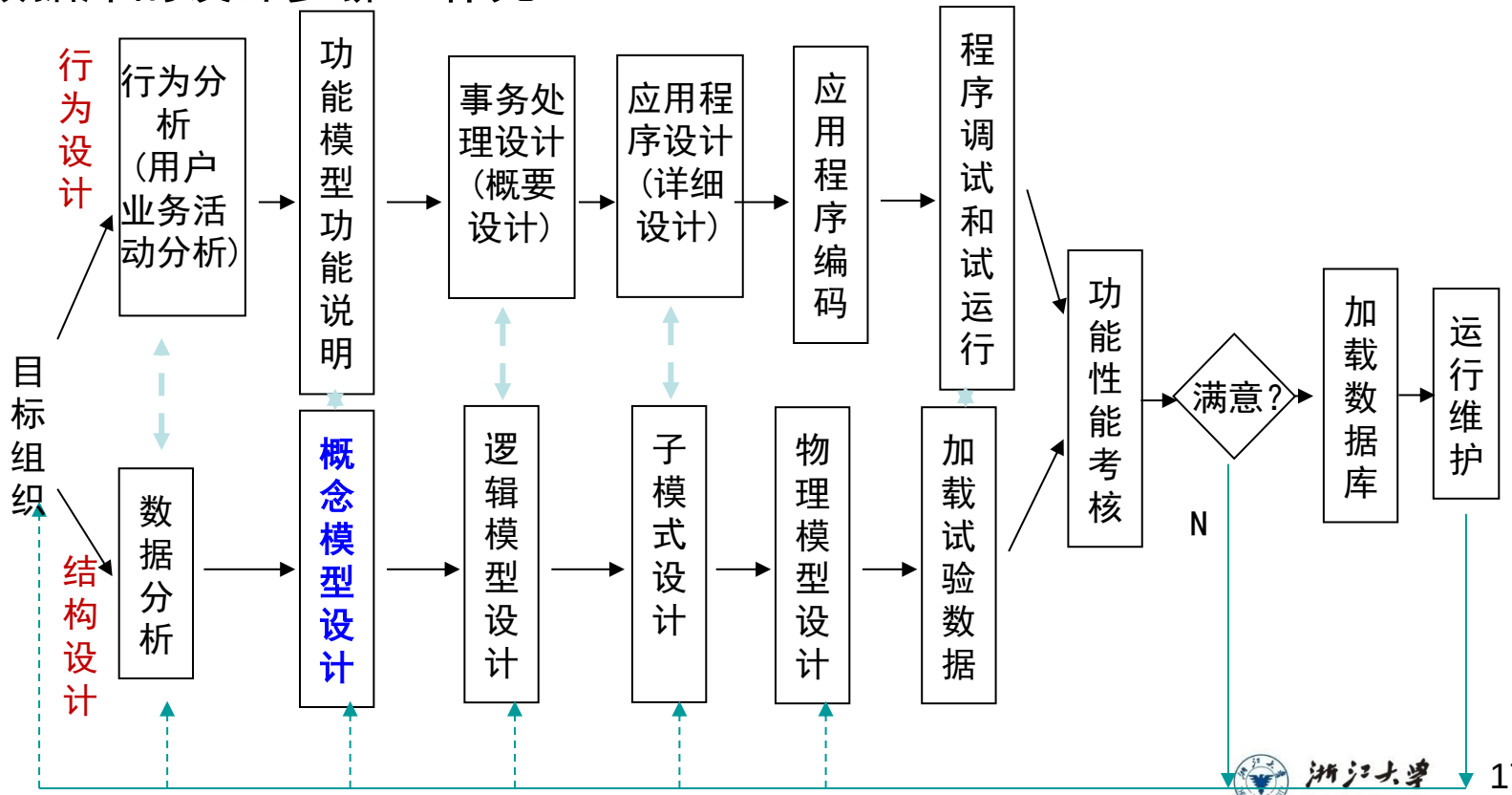
设计数据库的E-R模式

- 数据库的设计步骤（补充）



设计数据库的E-R模式

- 数据库的设计步骤（补充）



E-R设计决策

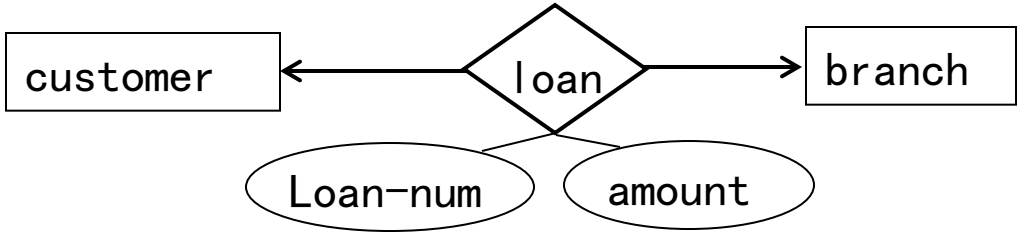
- 用属性还是实体集来表示对象
  - *instructor*(*ID*, *name*, …, *phone*)，优点：简单。但多个电话怎么处理？电话的其他属性？
  - *instructor*(*ID*, *name*, …)

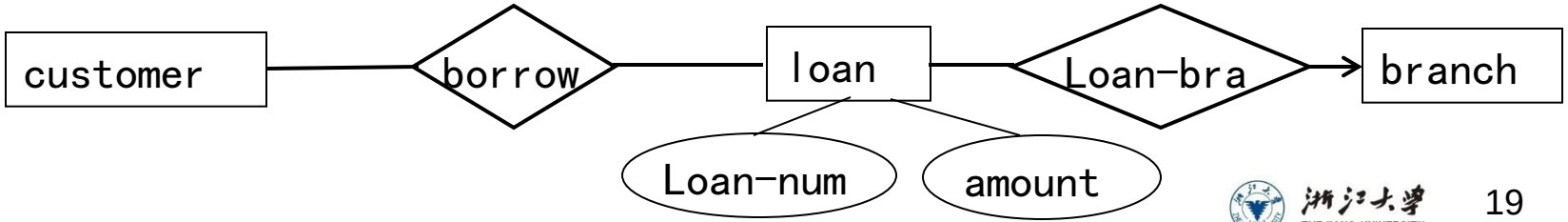
    phone(*phone-num*, *location*, *type*, *color*)

    ins-phone(*i_ID*, *phone-num*)
  - 若一个对象只对其名字及单值感兴趣，则可作为属性，如*性别*；若一个对象除名字外，本身还有其他属性需描述，则该对象应定义为实体集。如*电话*，*住址*，*部门*
  - 一个对象不能同时作为实体和属性
  - 一个实体集不能与另一实体集的属性相关联，只能实体与实体相联系

E-R设计决策

- 用实体集还是用联系集
  - 二个对象之间发生的动作用“relationship set”表示
  - 还需要考虑映射基数的影响
  - 例，考虑*branch*、*loan*、*customer*，如果一个客户在一个分支机构有多个贷款账户，那么将会影响E-R设计





E-R设计决策

- 用实体属性还是用联系
  - *student*(*sid*, *name*, *sex*, *age*, …, *supervisor-id*, *supervisor-name*, *supervisor-position*, …, *class*, *monitor*)
  - 要从对象的语义独立性和减少数据冗余考虑

    student(*sid*, *name*, *sex*, *age*, …);
    supervisor(*sup-id*, *name*, *position*, …);
    stu-sup(*sid*, *sup-id*, *from*, *to*);
    class(*classno*, *specialty*, *monitor*, *stu-num*);
    stu-class(*sid*, *classno*);

E-R设计决策

- 用n元联系还是二元联系
- 用强实体集还是弱实体集
- 特化/概化的使用，有助于设计的模块化
- 聚集的使用，将聚集实体集视为单个单元，从而不必关心其内部结构的细节

大学数据库的设计

- 获取系统需求
  - *classroom*、*department*、*course*、*instructor*、*section*、*student* 、*time_slot*
- 实体集设计
  - *classroom* (*building*, *room_number*, *capacity*)
  - *department* (*dept_name*, *building*, *budget*)
  - *course* (*course_id*, *title*, *credits*)
  - *instructor* (*ID*, *name*, *salary*)
  - *section* (*course_id*, *sec_id*, *semester*, *year*)
  - *student* (*ID*, *name*, *tot cred*)
  - *time_slot* (*time_slot_id*, {(*day*, *start_time*, *end_time*)})

- 联系集设计
  - *inst_dept*：关联教师和系
  - *stud_dept* ：关联学生和系
  - *teaches*：关联教师和开课
  - *takes*：关联学生和开课，包含描述性属性*grade*
  - *course_dept*：关联课程和系
  - *sec_course*：关联开课和课程
  - *sec_class*：关联开课和教室
  - *sec_time_slot*：关联开课和时段
  - *advisor*：关联学生和教师
  - *prereq*：关联课程和先修课程

大学数据库的设计

- E-R图