



数据库设计



我们曾经的工作数据库

■ simple movies database

- `Movies(title, year, length, movieType, studioname, producerC)`
- `movieStar(name, address, gender, birthdate)`
- `StarIn(movietitle, movieyear, starname)`
- `MovieExec(name, address, cert, netWorth)`
- `Studio(name, address, presC)`

以上关系模式是如何设计出来的？
这样的设计合理吗？

现实世界

1. 现实世界的个体：

个体是一个实际存在的且可以被识别的事物。

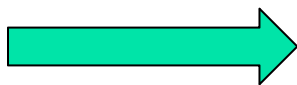
👉 可以是可接触的具体事物，如学生。

👉 也可以是不可接触的抽象概念，如课程。



学生

学习



课程

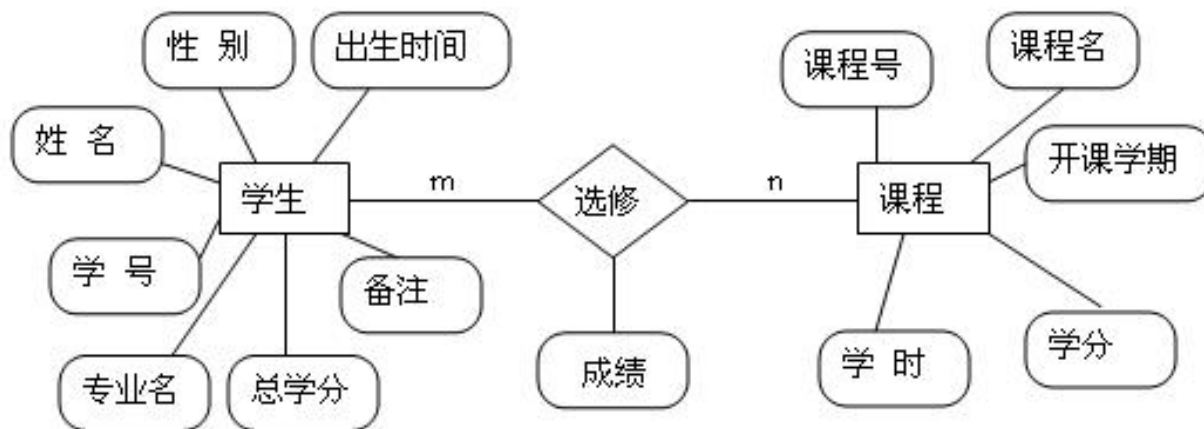
信息世界

2. 信息世界的概念模型

实体(Entity)：客观存在并可相互区分的事物。

属性：实体具有的某一特征。

联系(Relationship)：事物内部以及事物之间的相互关系。



机器世界

3. 机器世界中的关系模型

- (1) 关系模式：是由概念模型转换得到的。
- (2) 关系表：是由关系模式+数据的定义域+完整性约束。

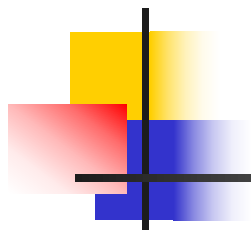
关系模式：

学生(学号，姓名，专业名，性别，出生日期，总学分，备注)

课程(课程号，课程名，开课学期，学时，学分)

成绩(学号，课程号，成绩)

学号	姓名	出生年月	性别	系	年级
970501	张平	198005	女	计算机	3
970502	王鹰	198103	男	计算机	3
970601	周丽	198205	女	机械	3
980209	李强	198011	男	化工	2



现实世界

第一级
抽象



信息世界
概念模型

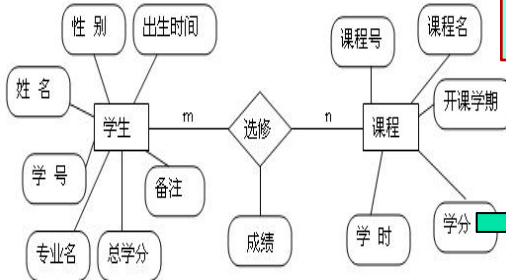
第二级
抽象



机器世界
关系模型



概念设计



逻辑设计

学号	姓名	出生年月	性别	系	年级
970501	张平	198005	女	计算机	3
970502	王鹰	198103	男	计算机	3
970601	周丽	198205	女	机械	3
980209	李强	198011	男	化工	2

数据库设计

用户需求
规格说明

需求分析

确定存储哪些数据，建立哪些应用，常用的操作及对象有哪些等，完整性，安全性

ER模型
ODL

概念结构设计

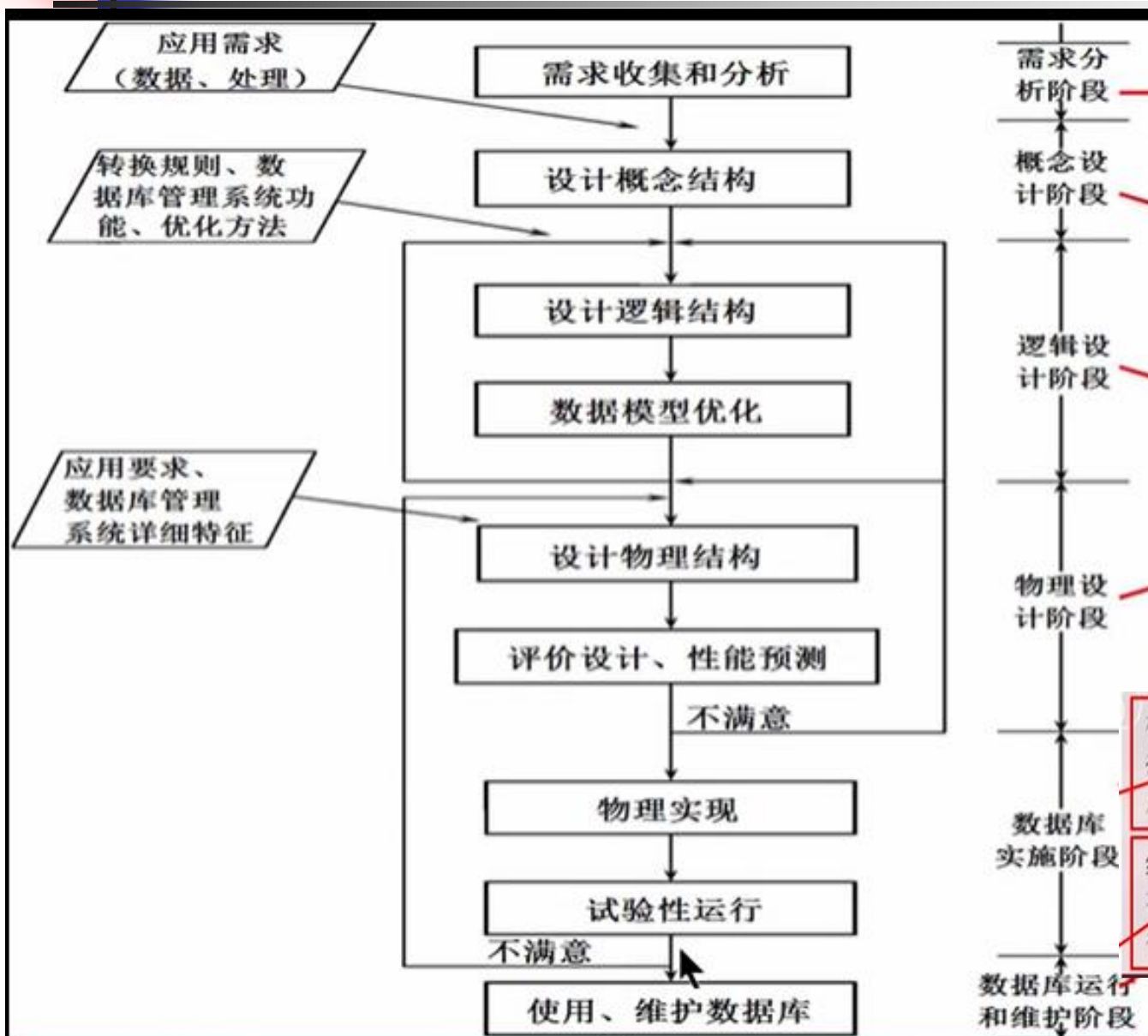
对需求分析所得到数据的更高层的抽象描述：
综合，归纳，抽象

逻辑结构设计

将概念模型映射为某个特定的数据库系统的数据模型，如关系模式

物理结构设计

数据库设计过程


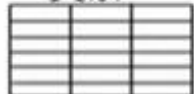


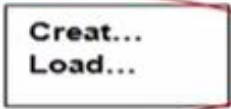


设计一个完善的数据库系统是一个不断反复的过程

根据逻辑设计和物理设计的结果构建数据库，编写与调试应用程序，组织数据入库并进行试运行

经过试运行后即可投入正式运行。在运行过程中必须不断对数据库设计进行评估、调整与修改。

设计阶段的输出

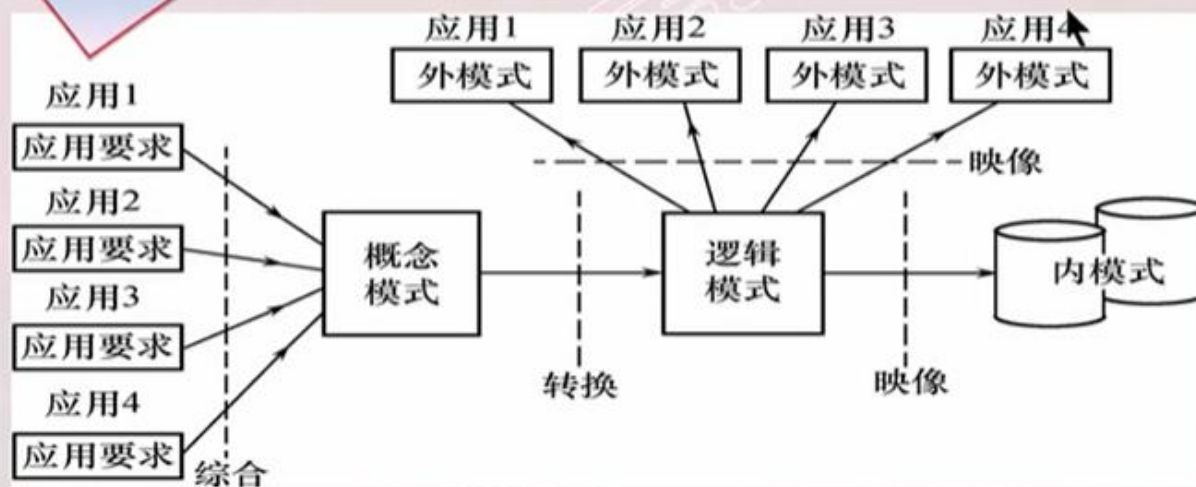
设计阶段	设计描述
需求分析	数字字典、全系统中数据项、数据结构、数据流、数据存储的描述
概念结构设计	概念模型 (E-R 图)  数据字典
逻辑结构设计	某种数据模型 <div> 关系  </div> <div> 非关系  </div>
物理结构设计	存储安排 存取方法选择 存取路径建立 
数据库实施	创建数据库模式 装入数据 数据库试运行 
数据库运行和维护	性能监测、转储/恢复、数据库重组和重构

数据库设计各个阶段产生的设计文档/设计说明

数据库设计各个阶段的数据设计描述

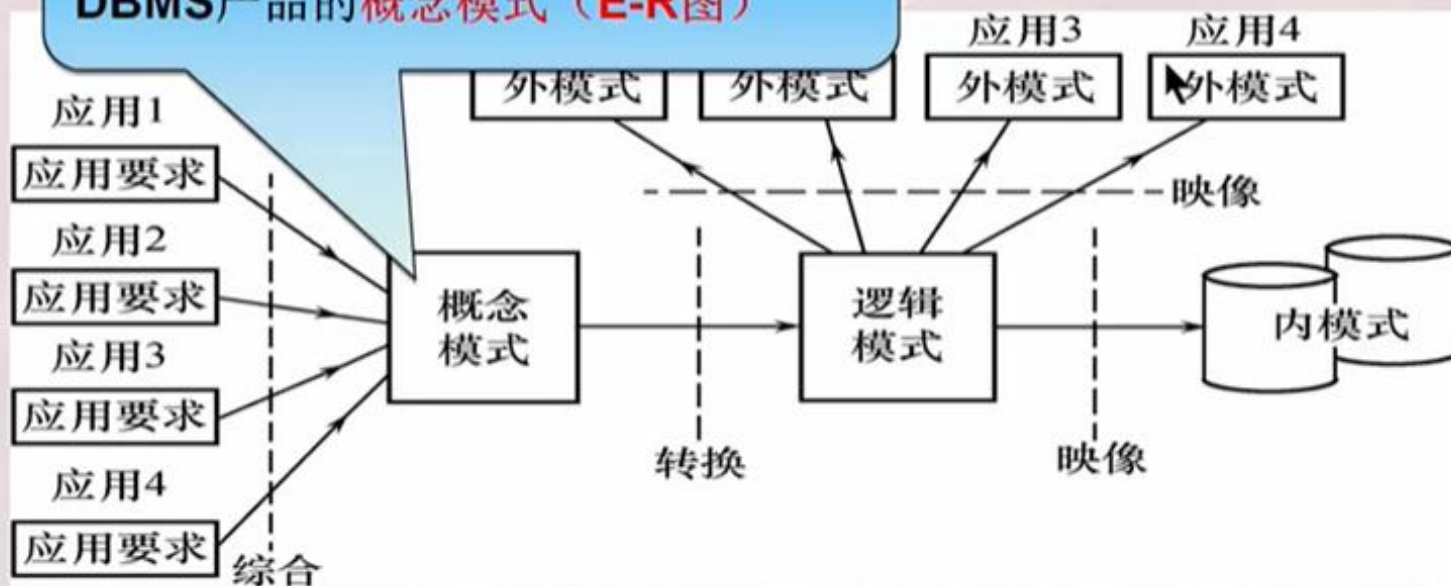
数据库设计各阶段与模式

需求分析阶段：
综合各个用户的应用需求

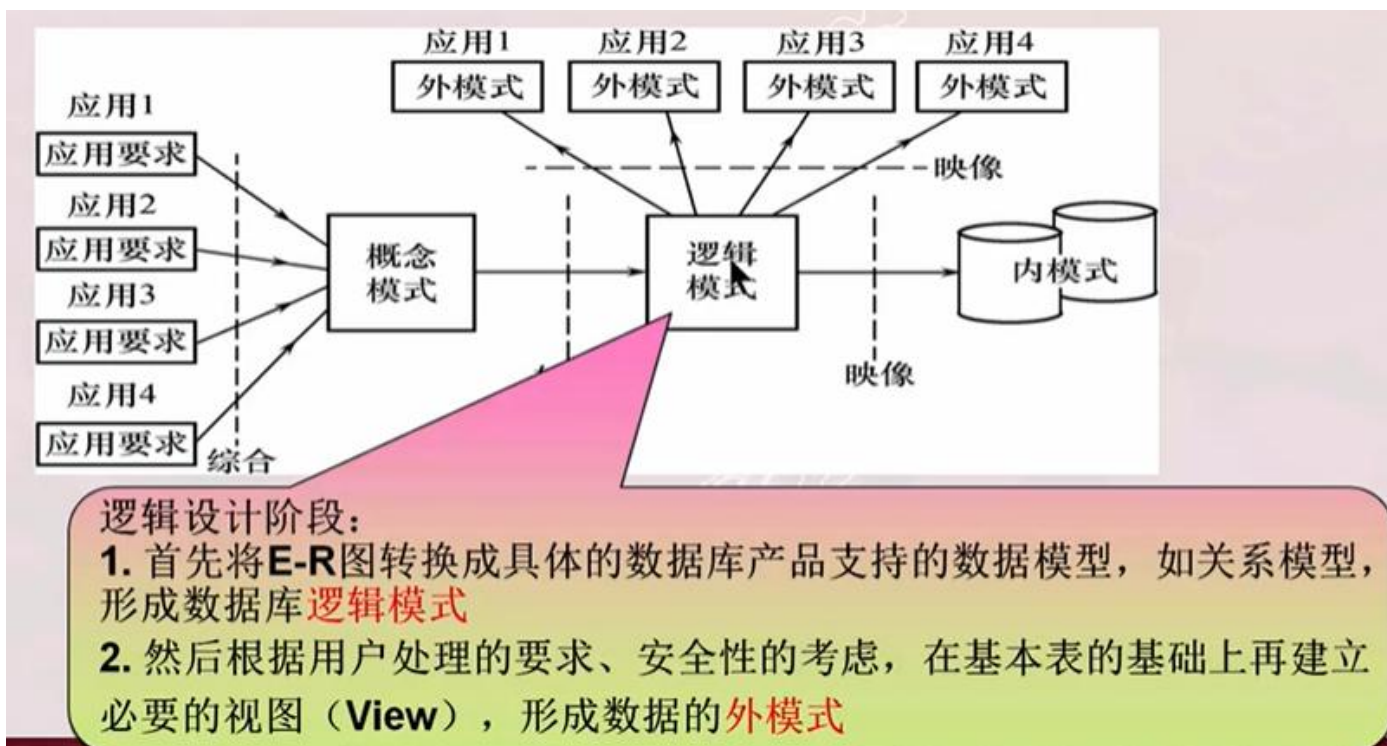


数据库设计各阶段与模式

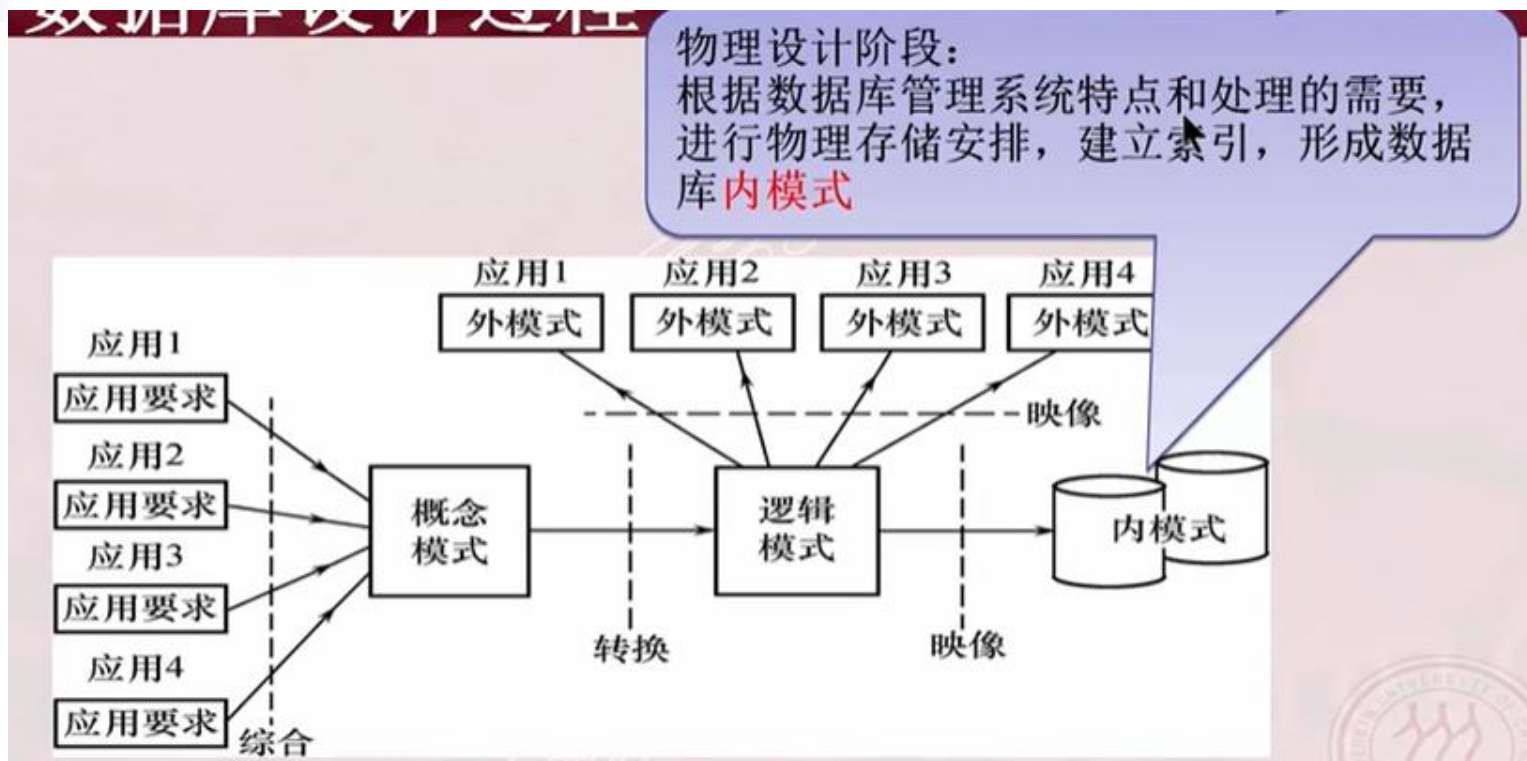
概念设计阶段：
形成独立于机器特点，独立于各个
DBMS产品的**概念模式（E-R图）**



数据库设计各阶段与模式



数据库设计各阶段与模式





数据库的设计原则

■ 避免冗余

- 帐户、客户名、地址、电话
- 贷款号、客户名、地址、电话
- 问题：帐户和贷款中的客户信息重复

■ 避免不完整

- 帐户、客户名、地址、电话
- 贷款、客户名、地址、电话
- 问题：无法输入一个新客户

■ 选择好的设计

- 关系数据理论中讨论

数据库设计方法学

■ 进行概念设计 (E/R 模型)

- E/R图来刻画实体世界
- 将E/R图转化为关系模式

■ 用规范化理论优化设计

- 规范化理论来消除冗余



哪些操作可能导致违反外键约束？

- 删除被参照表中记录
- 更新被参照表中记录
- 添加记录到被参照表中
- 删除参照表里记录
- 更新参照表中记录
- 添加记录到参照表中

✓

拒绝/级联删除/设置为NULL

✓

拒绝/级联修改/设置为NULL

显式说明违约处理方法：

On delete cascade/set null

On update cascade/set null

✓

拒绝

✓

拒绝

D	1990	120	sciFic	S4	104
M1	2004	120	drama	S1	101
spiderman	1997	120	sciFic	S2	102
Star	2002	120	comedy	S4	104
Y	0	NULL	NULL	S4	NULL
Y	2010	NULL	NULL	S4	NULL

11 rows in set (0.00 sec)

mysql> select * from studio;

name	address	presC
S1	NULL	101
S2	NULL	101
S3	Str1	102
S4	NULL	104

4 rows in set (0.02 sec)

mysql> update studio set name = 'X' where name = 'S4';

ERROR 1451 (23000): Can't
T `movies_ibfk_1` FOR

会员卡:

Customer(ID,name,sex,phone,balance) //balance 为余额

→在被参照表stud

产品:

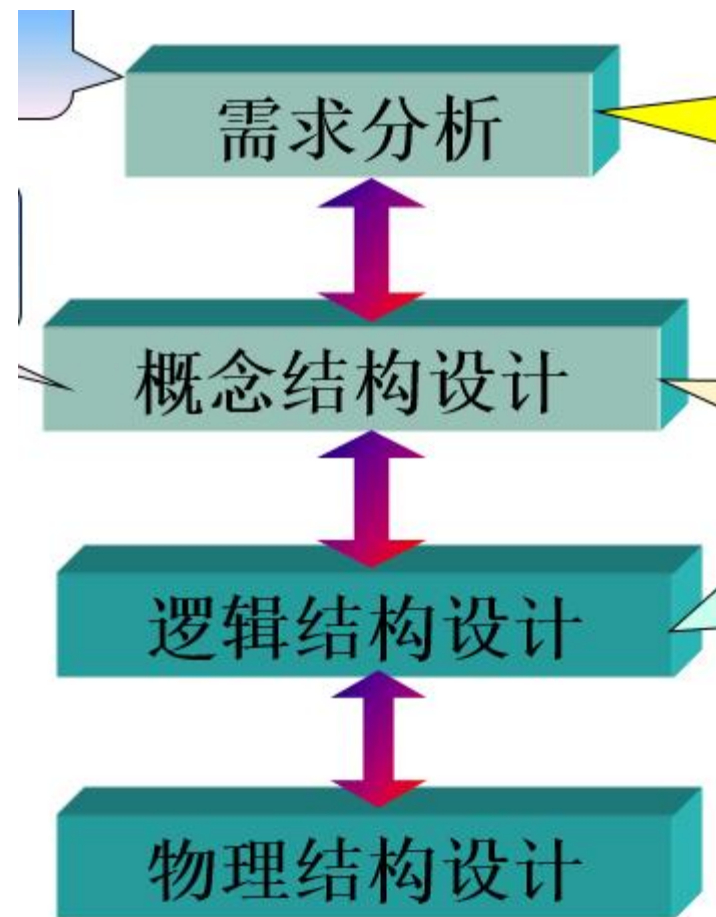
Product(ID,price,amount) //amount 为库存量

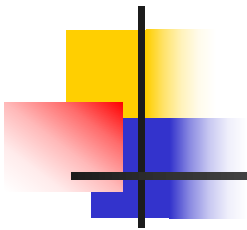
消费记录:

Reord(ID,CID,PID,time,amount,totalPrice) //amount 为数量

主要内容

- 实体/联系图（（E/R图））进行概念设计
- 将E/R设计图转化为逻辑结构（关系模式）
- 规范化理论优化逻辑结构





- 实体/联系图 ((E/R图))
- 将E/R设计图转化为逻辑结构 (关系模式)



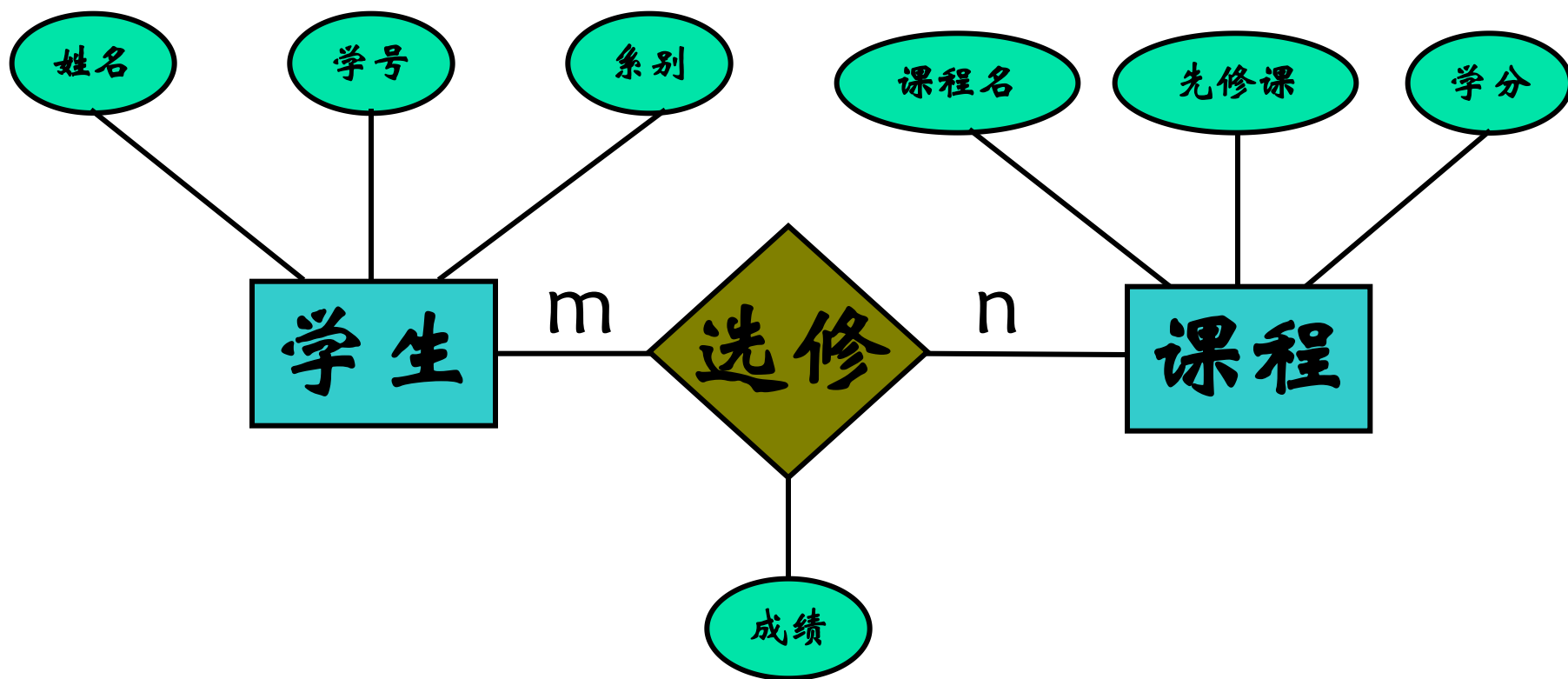
E-R模型

E-R模型：Entity-Relationship Model

- 1976年，由美籍华裔计算机科学家陈品山(Peter P. Chen)发明，这种数据模型典型的用在信息系统设计的第一阶段。
- E-R模型是建立在语义基础上的，即语义制造模型，即与时间/历史等有关。
- **E-R模型的基本观点**：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。
- 作用
 - 将**现实世界的事物和相互关联**进行抽象与描述，强调语义表达能力

E-R图概览

例：学生选修课程





基本概念

■ 实体(Entity)

- 客观存在并可相互区分的事物叫**实体**
- 如学生张三、工人李四、计算机系、数据库概论

■ 属性(Attribute)

- 实体所具有的某一特性称为实体的**属性**
- 一个实体可以由若干个属性来刻画
- 例如，学生可由学号、姓名、年龄、系等组成

■ 域(Domain)

- 属性的取值范围，即**值集**。
- 例如，性别的域为{男,女}，月份的域为{1,2,3,⋯,12}



基本概念

■ 实体型 (Entity Type)

- 实体名与其属性名集合共同构成**实体型**
- 例: 学生 (学号, 姓名, 年龄, 性别, 系, 年级)
- 注意**实体型与实体 (值)**之间的区别, 后者是前者的一个实例
- 如 (9808100, 王平, 21, 男, 计算机系, 2) 是一个实体

■ 实体集 (Entity Set)

- 同型实体的集合称为**实体集**
- 如全体学生



基本概念

customer-id customer- customer- customer-
 name street city

loan amount

321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan



如何唯一确定一个实体

Identifier/key (键, 码) = A set of attributes (属性或属性组) whose values **uniquely identify an entity** (唯一确定一个实体) in its entity set .

■ Example: 如演员名字唯一确定一名演员, “名字”属性就是键



基本概念

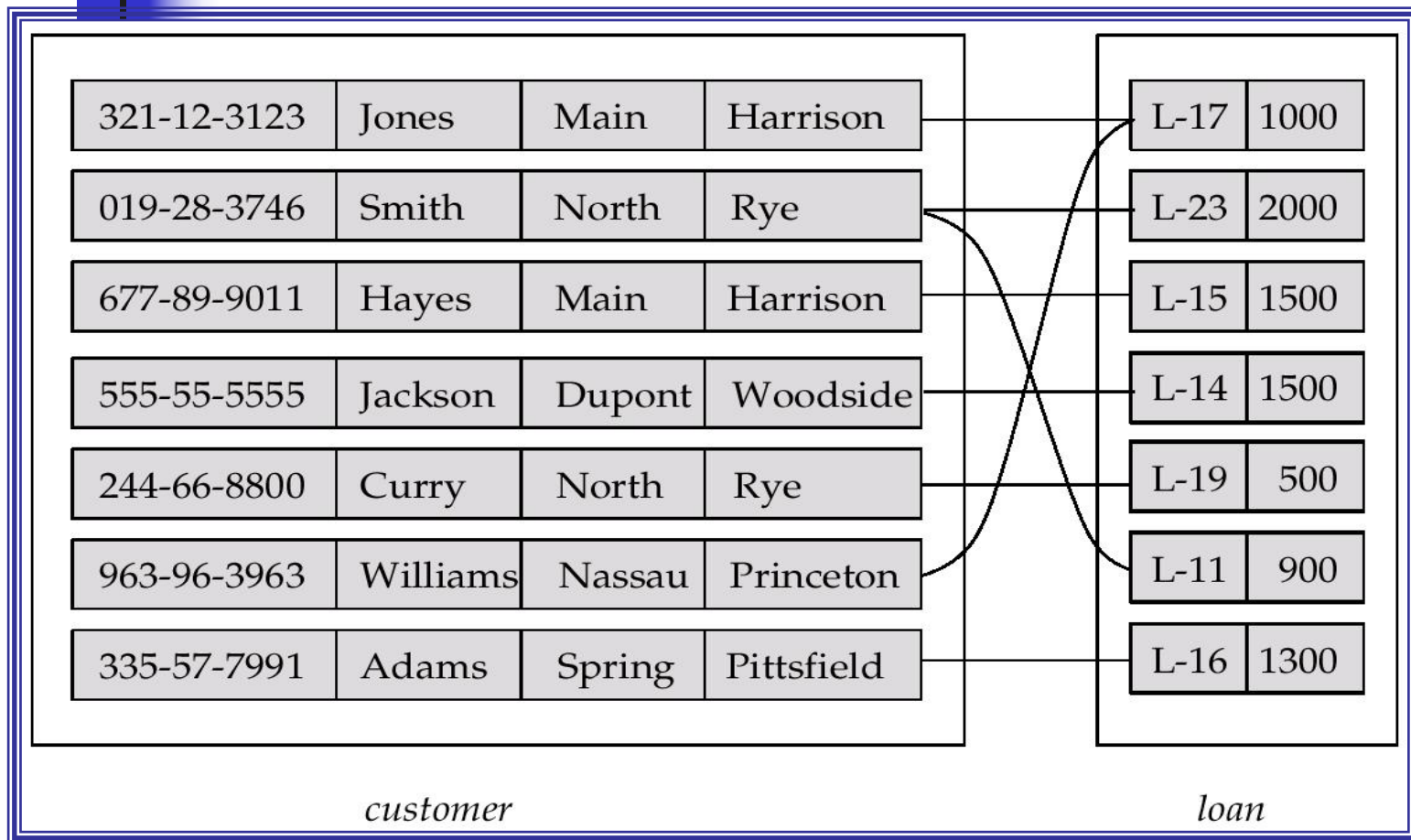
■ 联系 (Relationship)

- 实体（集）之间的关联。

■ Example :

- 联系 stars-in 将实体集 “**moviestar**” 与 “**movies**” 关联在一起，表明一些 moviestar 出演一些 movies.
- 联系 owns 将实体集 “**movies**” 与 “**studio**” 关联起来，表明一些 studio 发行一些 movies.

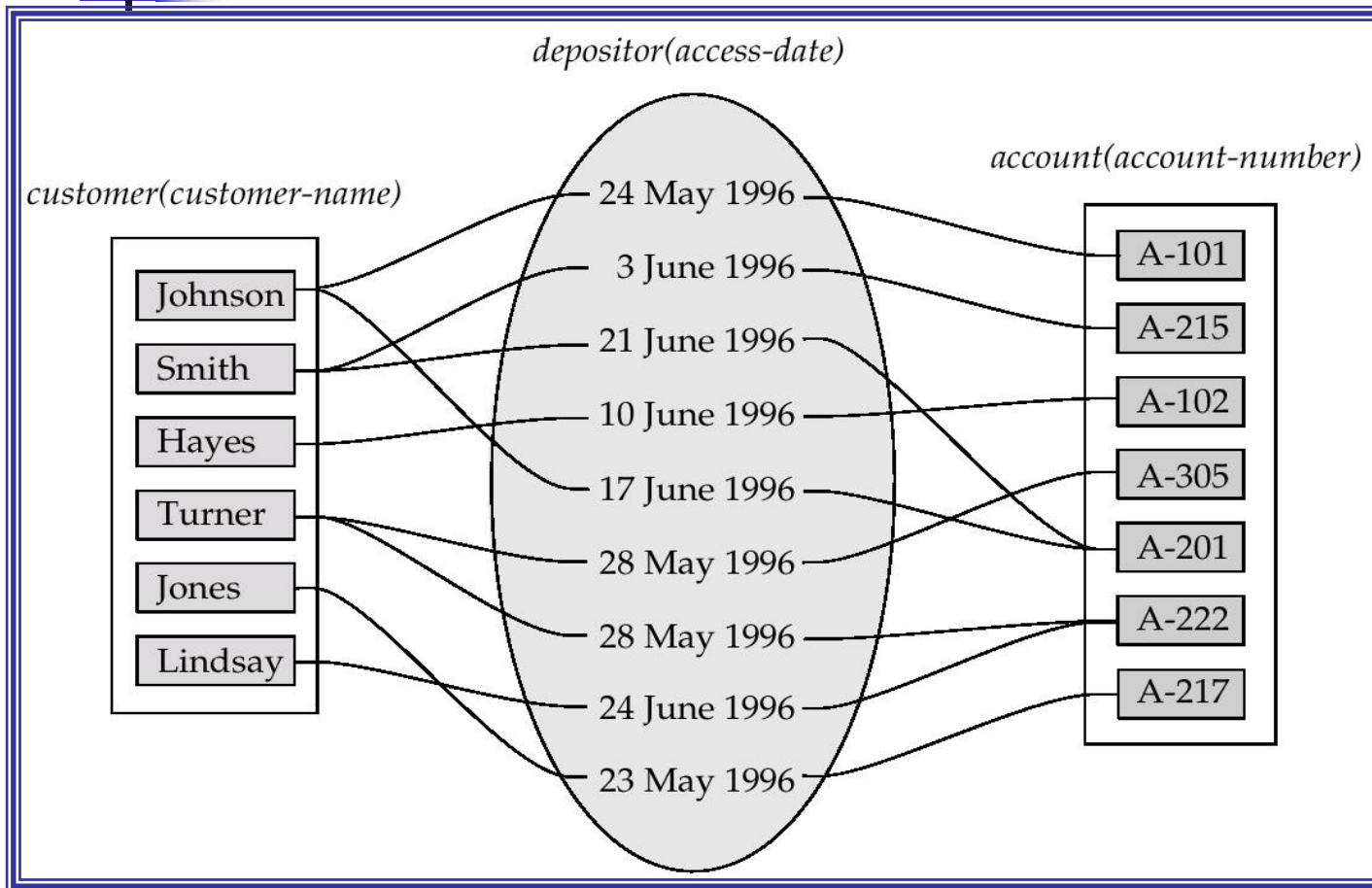
基本概念



联系集 *borrower*

■ 同类联系的集合称为联系集

基本概念



- 联系也可以有属性，中间属性属于联系，不属于任何实体。

实体集customer与account之间的联系集depositor具有属性access-date

E-R模型:符号



长方形：表示实体集，
内注实体名



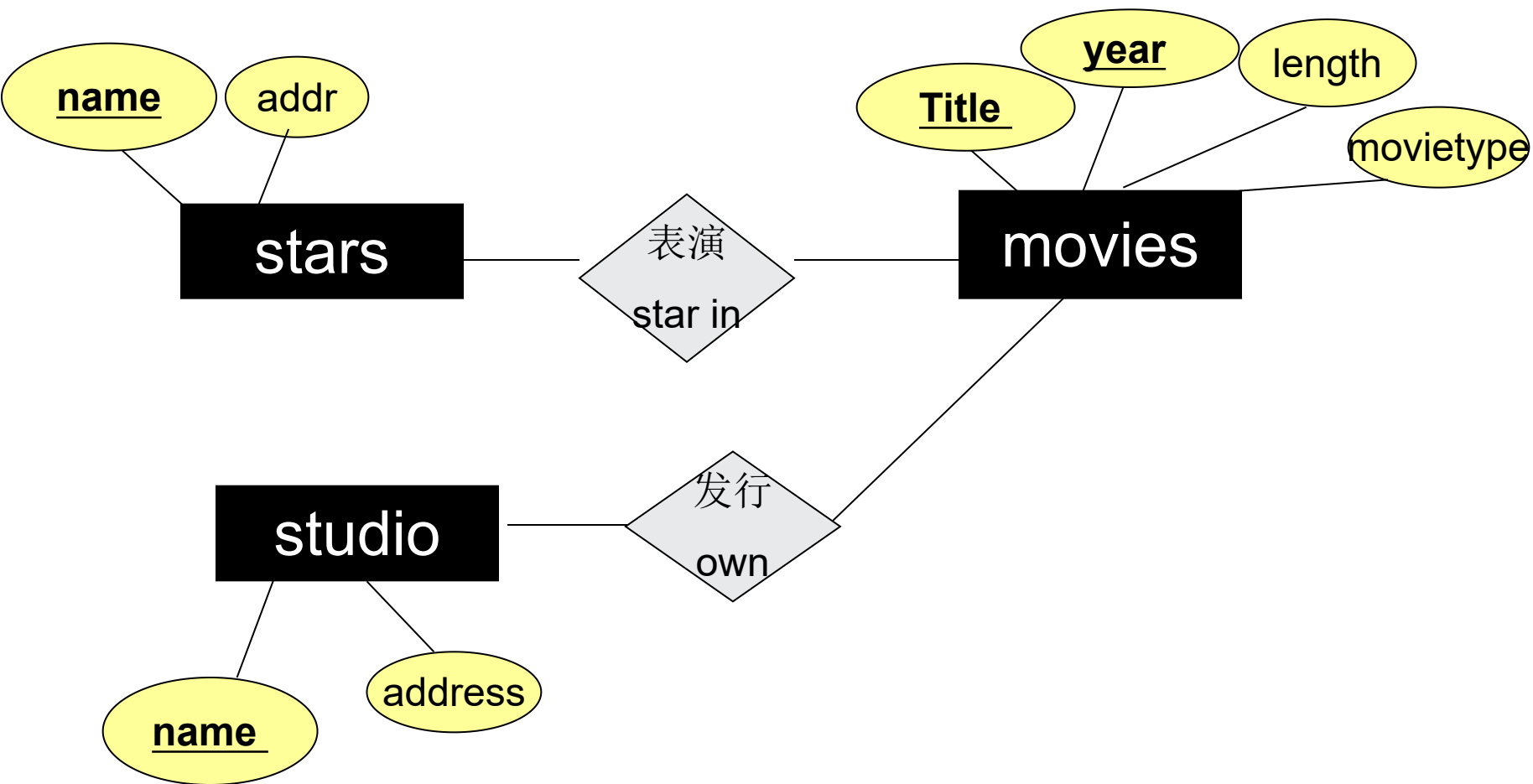
菱形：表示实体间的联系，
内注联系名



椭圆：表示实体的属性，**作为主键的属性或属性组用下划线标记**

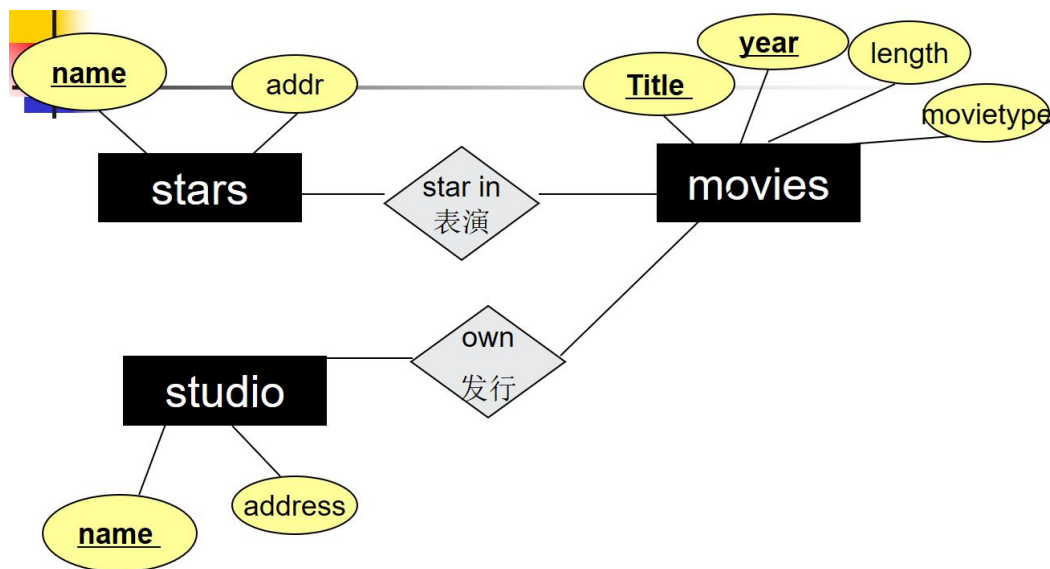


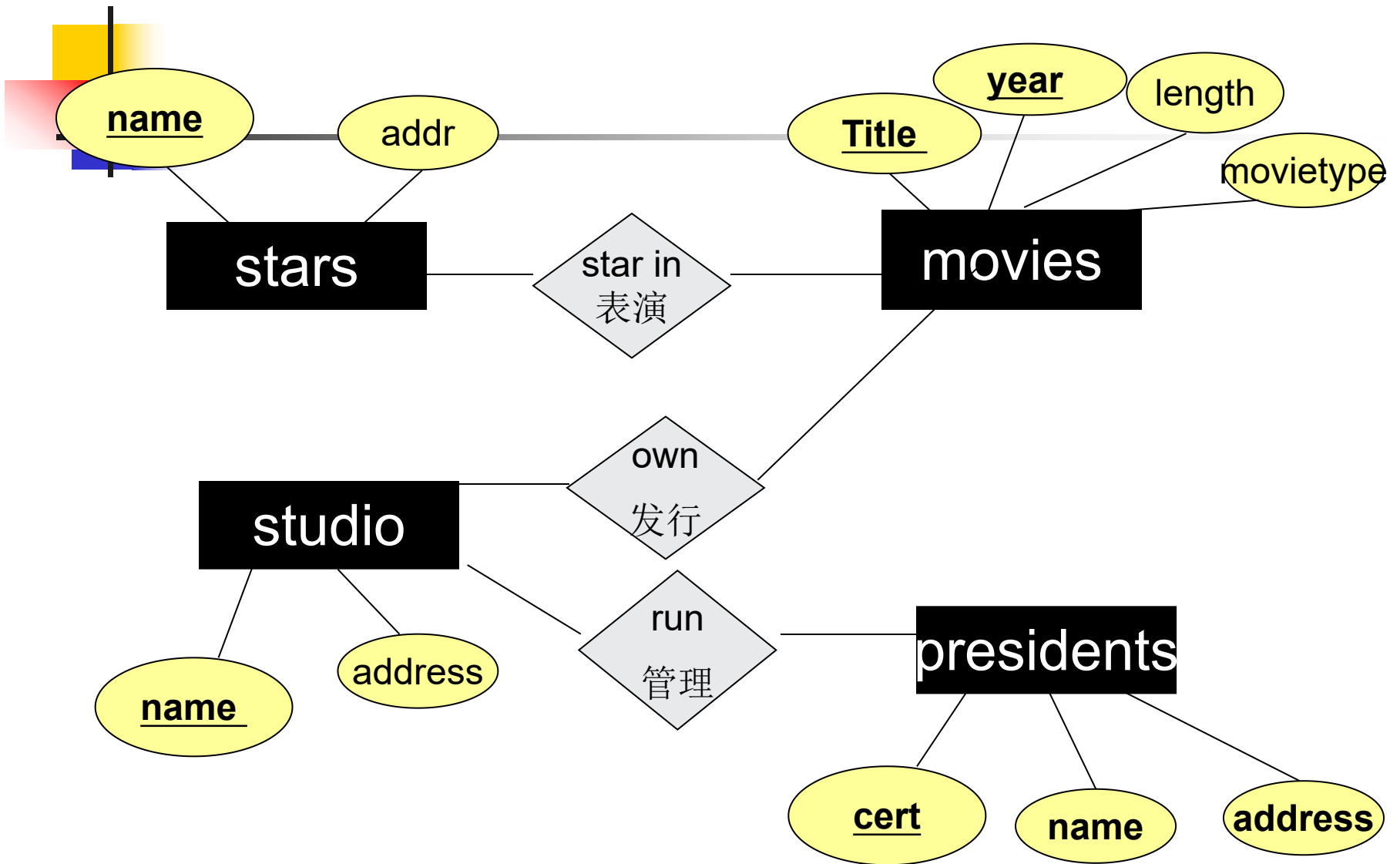
无向边：表示**将实体与相关的属性和联系相连**；将实体集和联系也用边相连



讨论

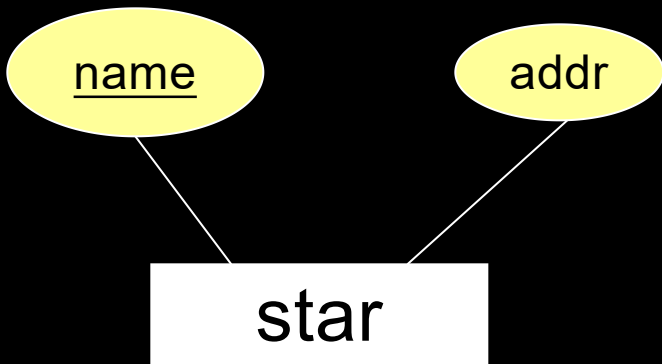
- 在movies数据库中，添加新的实体集以及联系来体现“公司由一些人来管理”。





General rules: 实体集→关系

每个实体集转化为一个关系，关系属性就是实体集属性



Star(name,addr)

star

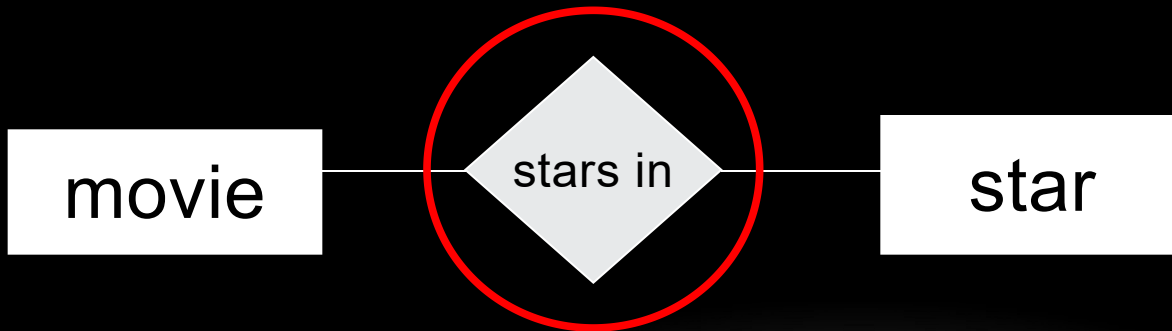
name	addr
carrie fisher	123 maple st.
mark hamill	123 maple st.
harrison ford	789 palm dr

实体实例

General rules:联系→关系

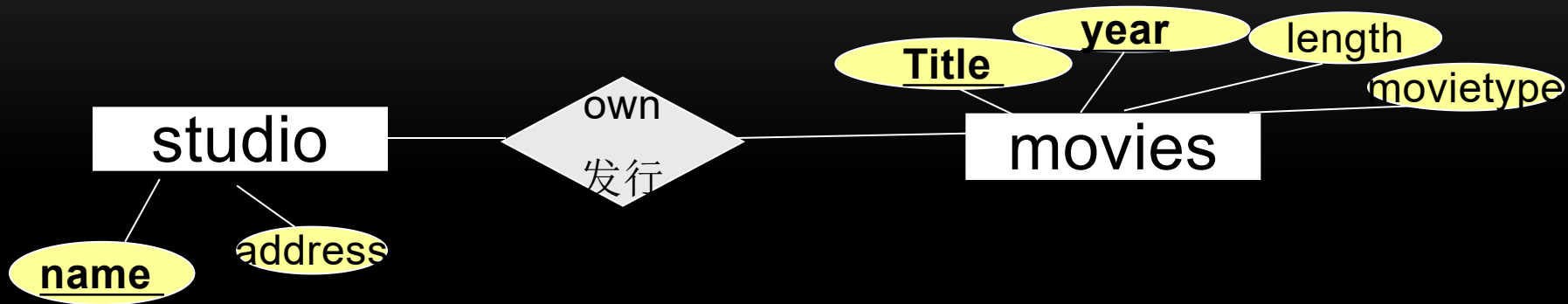
每个联系转化为一个关系，属性由该联系所连接的所有实体主键及自身属性构成

movieTitle ▼	movieYear ▼	starName ▼
Empire Strikes Back	1980	Harrison Ford
Star Wars	1977	Carrie Fisher
Star Wars	1977	Harrison Ford
Star Wars	1977	Mark Hamill
Terms of Endearment	1983	Debra Winger
Terms of Endearment	1983	Jack Nicholson



`starsIn(movietitle ,movieyear ,starname)`

讨论



- 将概念设计以 General rules 转化为关系模式
 - `studio(name,address)`
 - `movies(title,year,length,movietype)`
 - `own(title,year,studioname)`

发现问题？

movies

↑ title	↑ year	length	movieType
Empire Strikes Back	1980	111	drama
Gone With the Wind	1938	238	drama
Harry Potter and the Order of the...	2007	120	
Pretty Woman	1990	119	drama
Star Trek	1979	132	sciFic
Star Trek: Nemesis	2002	116	sciFic
Terms of Endearment	1983	132	drama
The Man Who Wasn't There	2001	116	comedy
The Rescuers	1977	120	Animation
The Usual Suspects	1995	106	drama

own

↑ title	↑ year	studioName
Empire Strikes Back	1980	Fox
Gone With the Wind	1938	MGM
Harry Potter and the Order of the...	2007	Warner Bros.
Pretty Woman	1990	Disney
Star Trek	1979	Paramount
Star Trek: Nemesis	2002	Paramount
Terms of Endearment	1983	MGM
The Man Who Wasn't There	2001	USA Entertainm.
The Rescuers	1977	Disney
The Usual Suspects	1995	MGM

studio

↑ name	address
Disney	Disney Boulevard
Fox	Fox Boulevard
MGM	MGM Boulevard
Paramount	Paramount Boulevard
USA Entertainm.	USA Entertainm Boulevard
Warner Bros.	NULL



E-R模型——联系的多样性



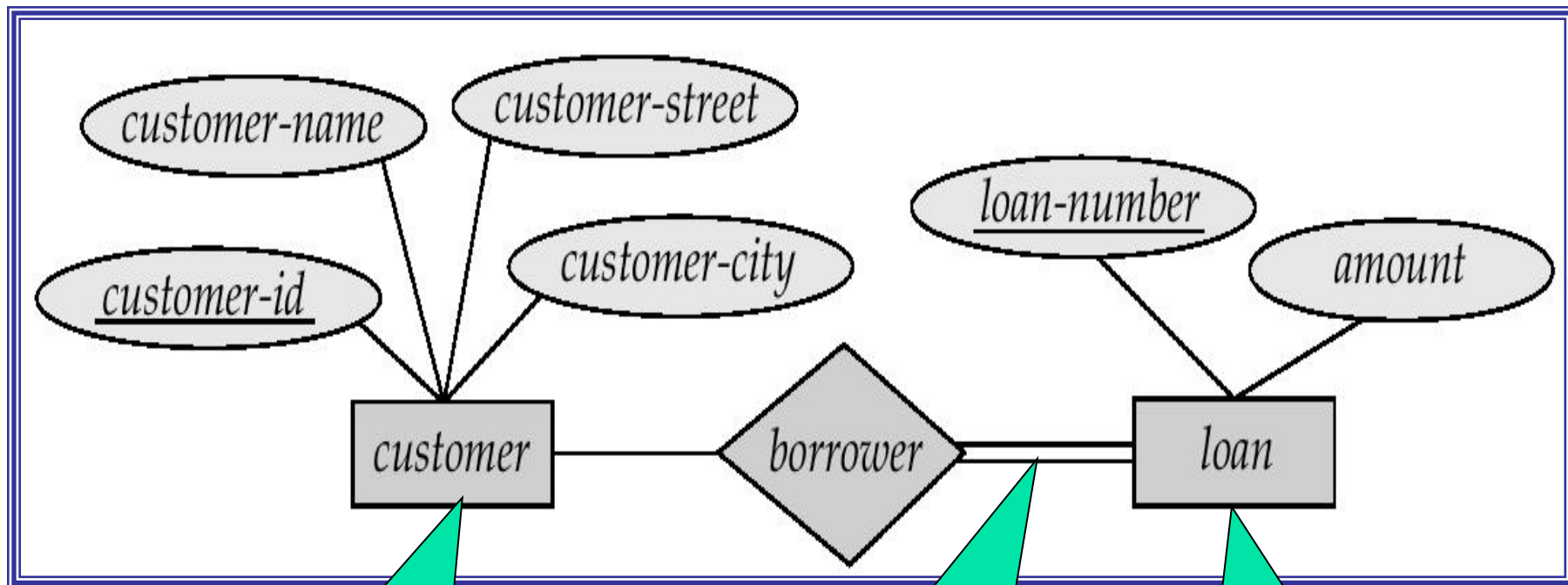
实体参与联系

■ 参与 (Participation)

- 实体集之间的关联称为参与，即**实体参与联系**
- 如“A”发行了电影“M1”，表示实体“A”与“M1”**参与**了联系“own”

■ 部分参与 VS 全部参与

- 如果实体集E中的每个实体都参与到联系集R中的至少一个联系，则称E**全部参与**R
- 如果实体集E中只有部分实体参与到联系集R的联系中，则称E**部分参与**R
- 如“movie”与“studio”之间的“own”联系，
“studio”实体集部分参与，
而“movie”实体集完全参与



customer部分参与
与联系borrower

用双线连接联系
与完全参与的实体

Loan完全参与
联系borrower



基本概念

■ 联系的元或度 (Degree)

■ 参与联系的实体集的个数称为联系的元

➤ 一元联系：同一实体集合内的实体间的联系

如：班长管理学生；领导管理员工；一个零件由多个零件组成。

➤ 二元联系：两个不同实体集合的实体间联系

如：学生选修课程；学生借阅图书。

➤ 多元联系：两个以上不同实体集合的实体间的联系

如：某工程项目需要多个供应商提供多种零件



联系的基数

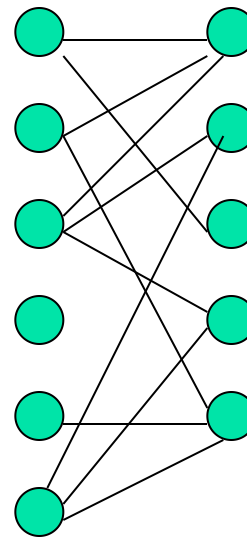
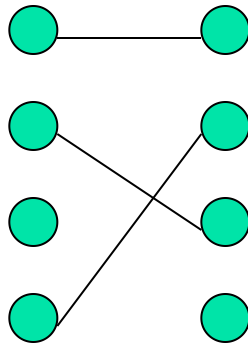
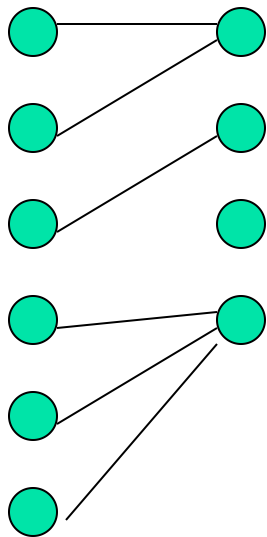
■ 映射的基数 (Mapping Cardinalities)

- 实体之间的联系的数量，即一个实体通过一个联系集能与另一实体集相关联的实体的数目
- 可以有一对一的 (1:1)，一对多的 (1:m)，多对多的 (m:n) 几种情况

二元联系（两个实体型之间）

■ 二元联系的多样性可以分为

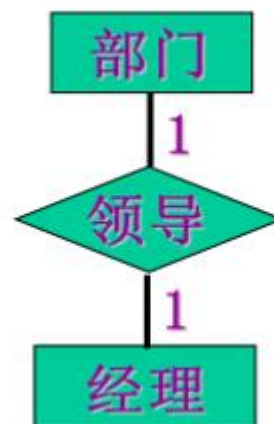
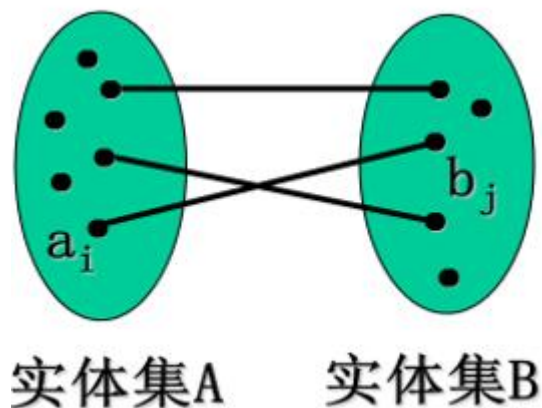
- 一对一
- 一对多（多对一）
- 多对多



一对一联系

(1) 1:1 联系:

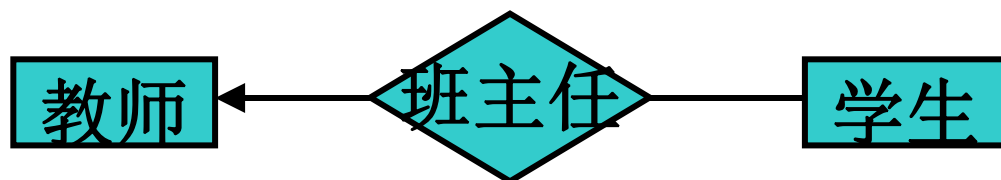
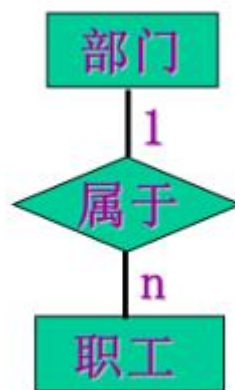
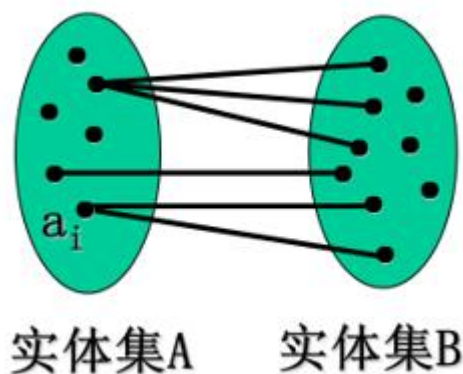
- 若对于实体集**A**中的一个实体，实体集**B**中**至多有一个实体**与之联系，反之亦然，则称实体集**A**与实体集**B**具有一对一联系，记为**1:1**。
- **1:1**联系同时起着在一个实体集中的实体标识另一个实体集中实体的作用。



一对多联系

(2) 一对多联系：

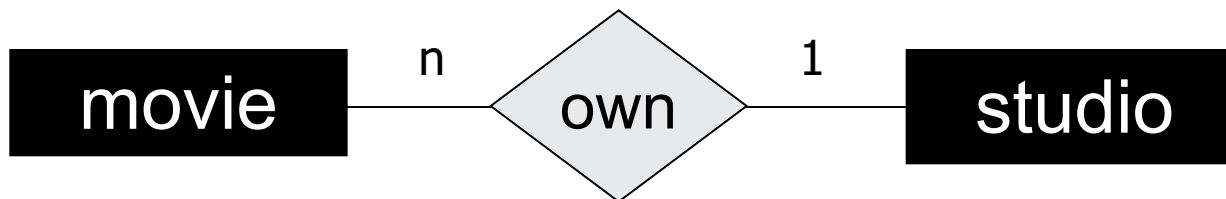
对于实体集A中的一个实体，实体集B中有n个实体与之联系，反之，对于实体集B中的一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B具有一对多的联系



多对一联系

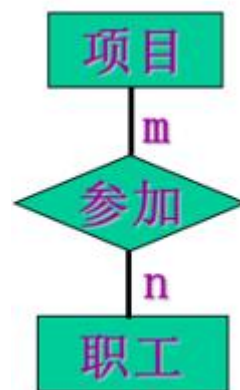
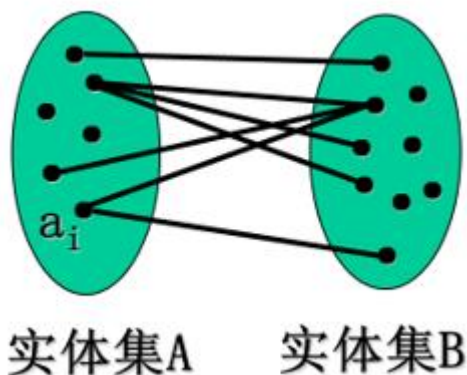
多对一联系：

- Example: 一个studio 可以发行多个movies, 但是每个movie只被一个studio发行.

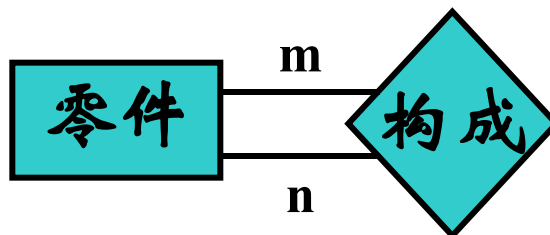
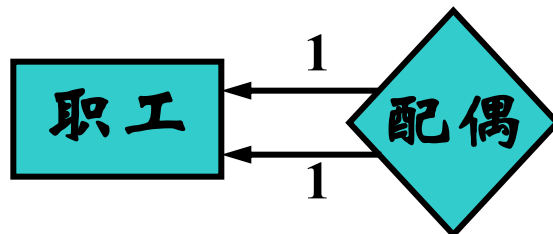
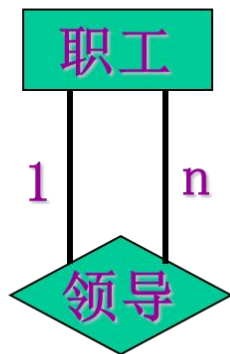
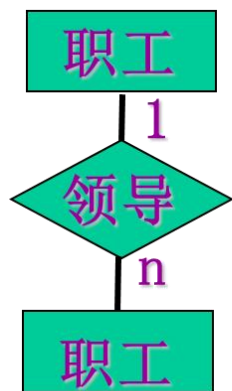


多对多联系

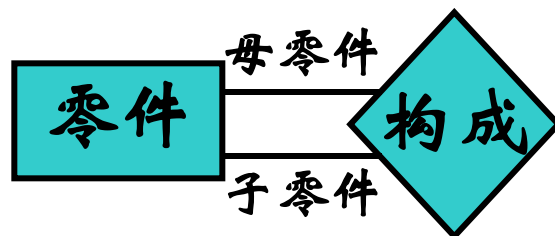
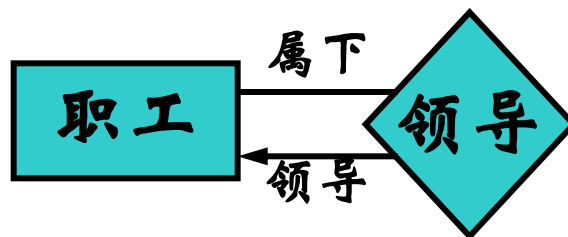
(3) $m:n$ 联系：若对于实体集A中的每一个实体，实体集B中有n个实体与之联系，反之，对于实体集B中的每一个实体，实体集A中有m个实体与之联系，则称实体集A与实体集B具有多对多的联系，记为 $m:n$ 。



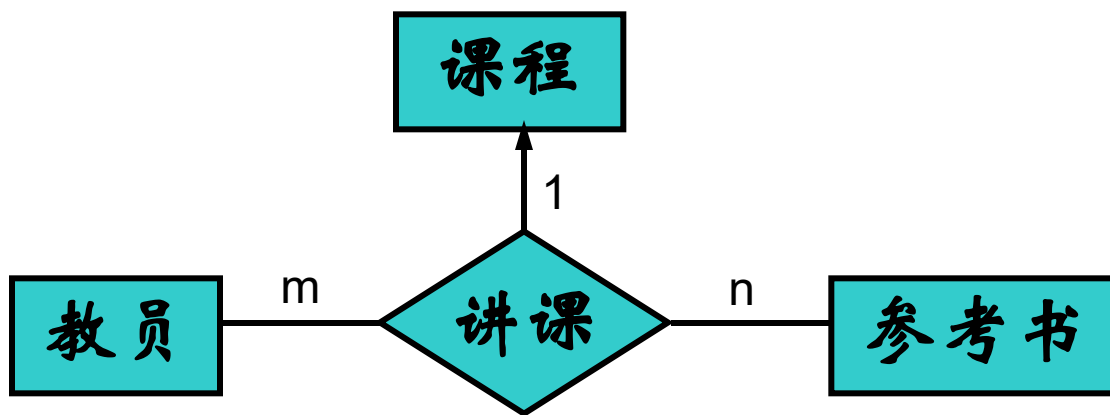
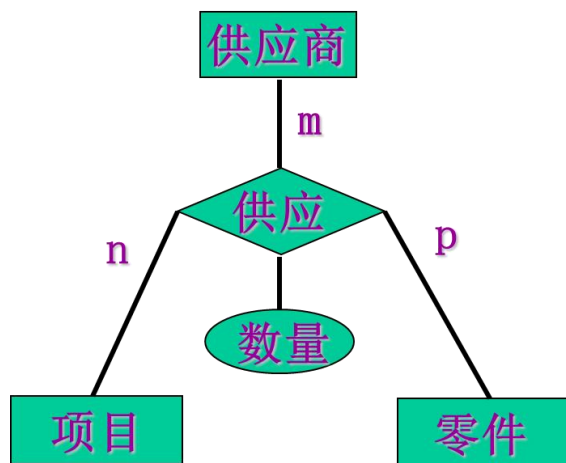
一元联系（单个实体型内）



实体在联系中的作用称为实体的**角色**：当同一个实体集不止一次参与一个联系集时，可显式指明其角色。

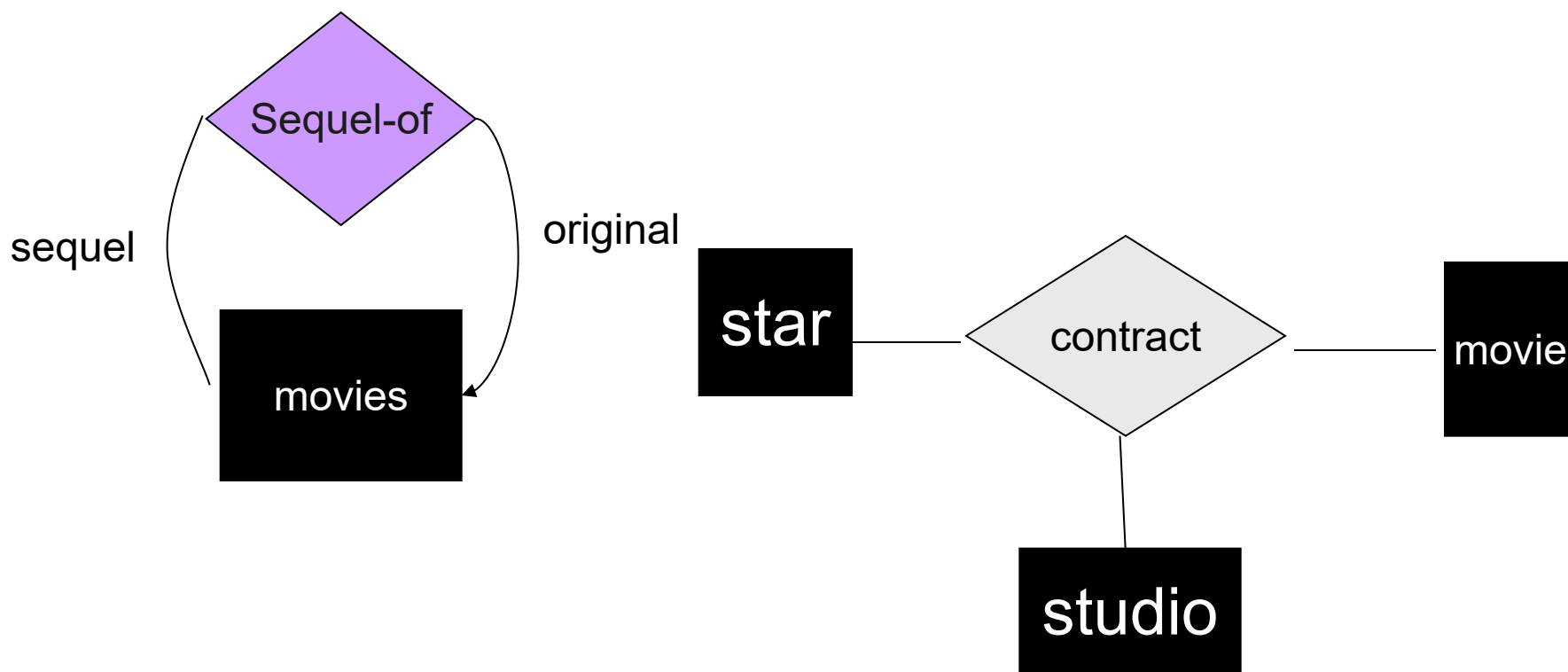


多元联系（多个实体型内）



讨论

- (1) **movies** 实体集内部有某种联系吗？如何表达
- (2) **movieStar**, **Movies**, **Studio** 之间的签约关系如何表达？



E-R模型向关系模式的转换

- 一对一联系：



职工（职工号，姓名...）
部门（部门号，部门名...）
管理（职工号，部门号）



合并：若联系一方全部参与，则将联系另一方的码作为全部参与一方的属性。

职工（职工号，姓名...）
部门（部门号，部门名，**管理职工号**）

E-R模型向关系模式的转换

- 一对多联系：



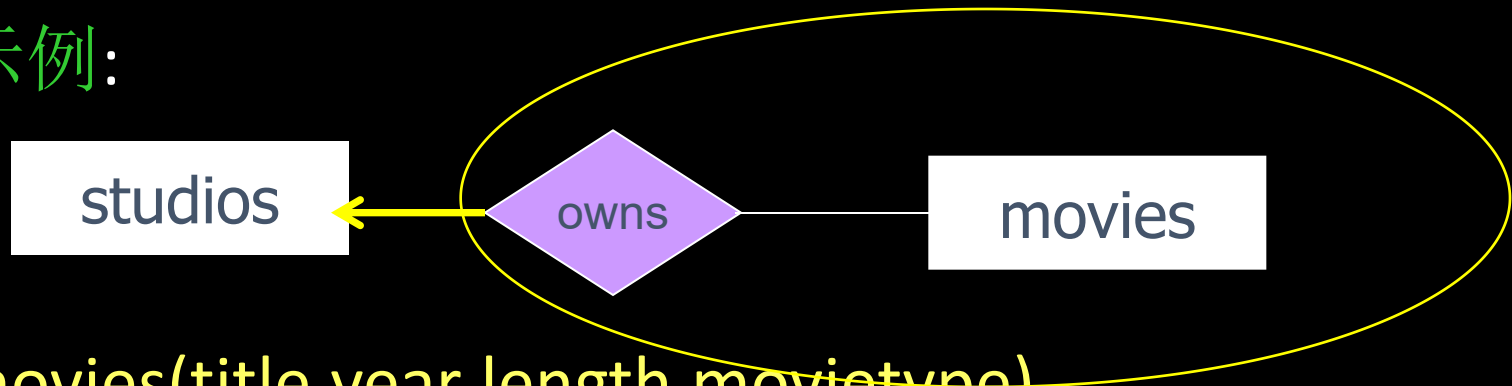
学生(学生号, 学生名, 系号)
教师(教师号, 姓名...)
班主任(教师号, 学生号)



将单方参与一方实体的码作为多方参与实体的属性。

学生(学生号, 学生名, 系号, 班主任教工号)
教师(教师号, 姓名...)

- 示例:

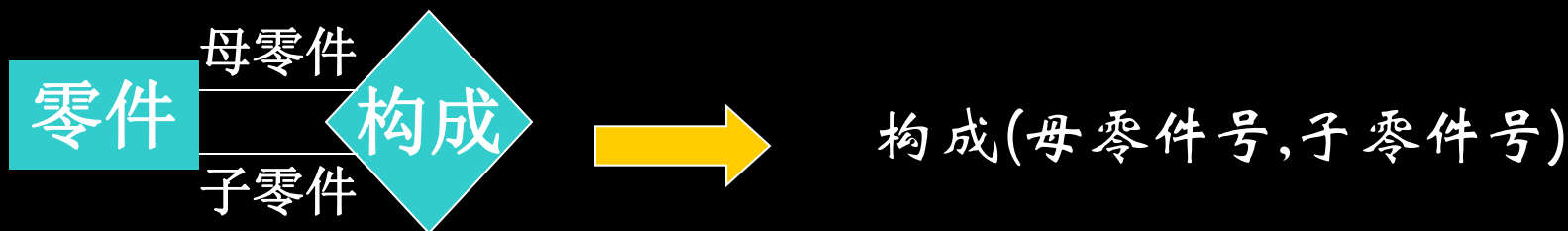
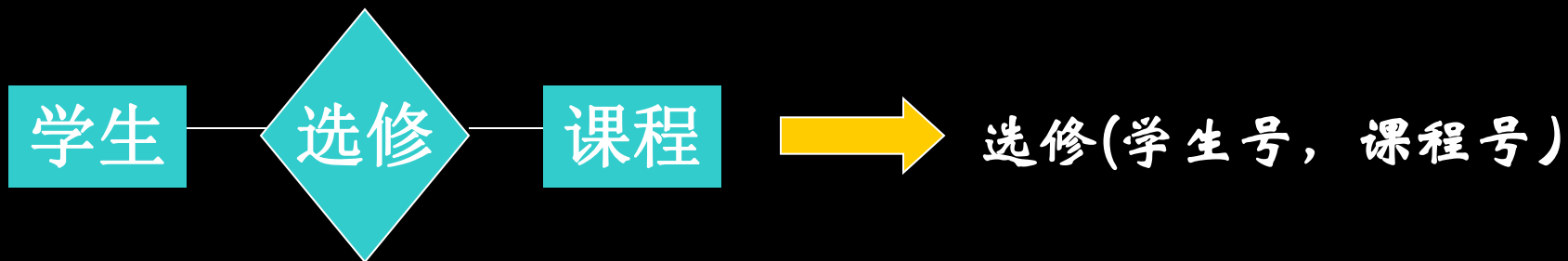


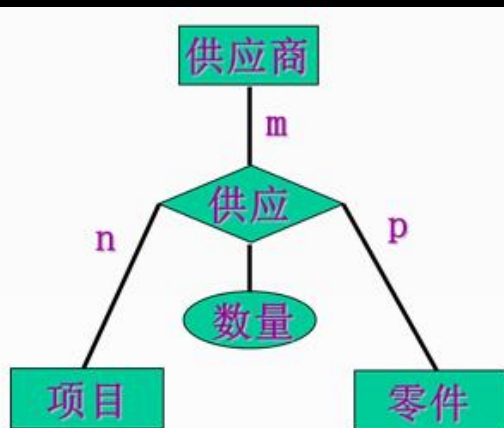
- `movies(title,year,length,movietype)` ,
- `studio(name, addr)`
- `owns(title,year, name)`
- after :
 - `movies1(title,year,length,movietype, studioname).`
 - `studio(name, addr)`

E-R模型向关系模式的转换

- 多对多联系：

将联系定义为新的关系，属性为参与双方的码。





供应商 (供应商号,)

项目 (项目号,)

零件 (零件号,)

订单 (供应商号, 项目号, 零件号, 数量)

供应商 (供应商号,)

项目 (项目号,)

零件 (零件号,)

订单 (ID, 供应商号, 项目号, 零件号, 数量)



构建E-R模型

1. 确定实体：实体往往是需求中的名词
2. 确定实体的属性及键（主属性）
3. 确定实体的联系：联系的基数和属性

讨论

例：以下是用户需要分析所得出的语义，请根据语义画出E-R图

- (1) 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中；仓库有仓库号、仓库类型和面积，零件有零件号、名称、规格、单价、描述等属性。
- (2) 一个职工只能在一个仓库工作，一个仓库有多个职工当保管员；职工有职工号、姓名、性别、职务、出生年月属性。
- (3) 职工之间有领导与被领导关系，仓库主任领导若干保管员；
- (4) 一个供应商可以供应若干项目多种零件；而一个项目可以使用不同供应商供应的多种零件；一种零件可由不同供应商供给多个工程项目。供应商有供应商号、姓名、住址、电话、帐号，项目有项目号、预算、开工日期等属性。



1. 确定实体：实体往往是需求中的名词

供应商

仓库

职工

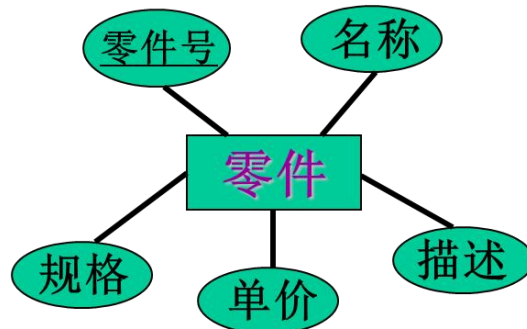
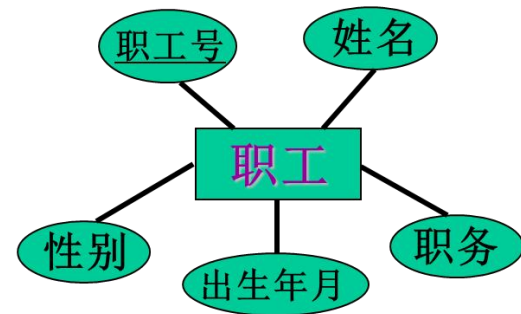
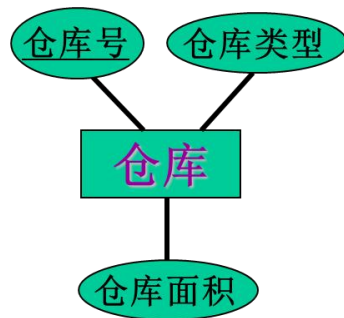
项目

零件

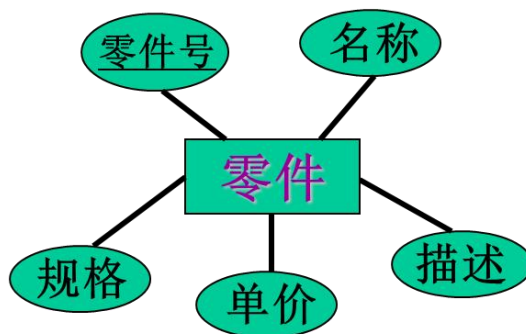
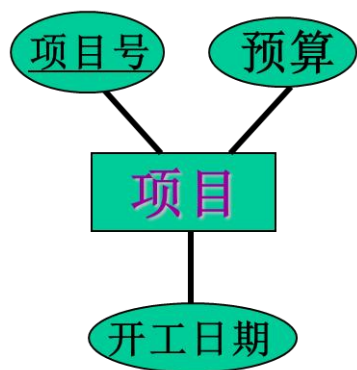
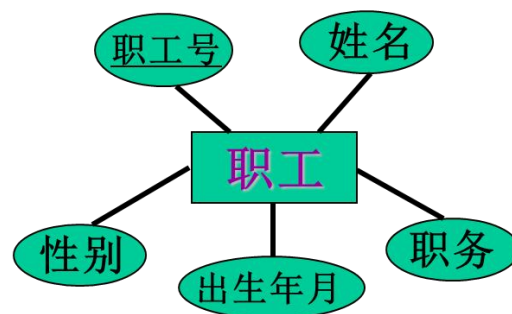
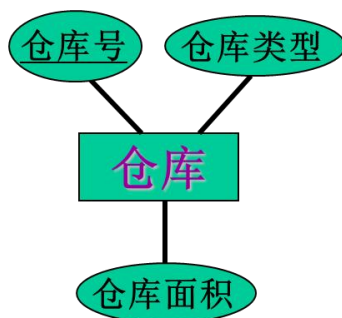
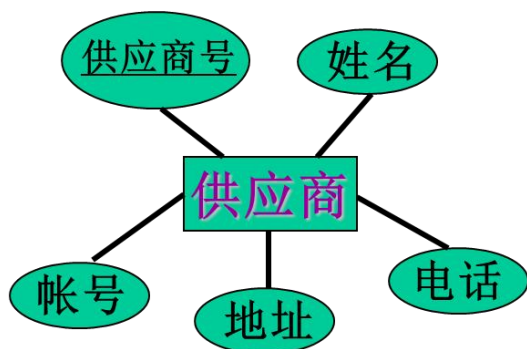
(1) 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中；
仓库有仓库号、仓库类型和面积，零件有零件号、名称、规格、单价、描述等属性。

(2) 一个职工只能在一个仓库工作，一个仓库有多个职工当保管员；
职工有职工号、姓名、性别、职务、出生年月属性。

(3) 职工之间有领导与被领导关系，仓库主任领导若干保管员；

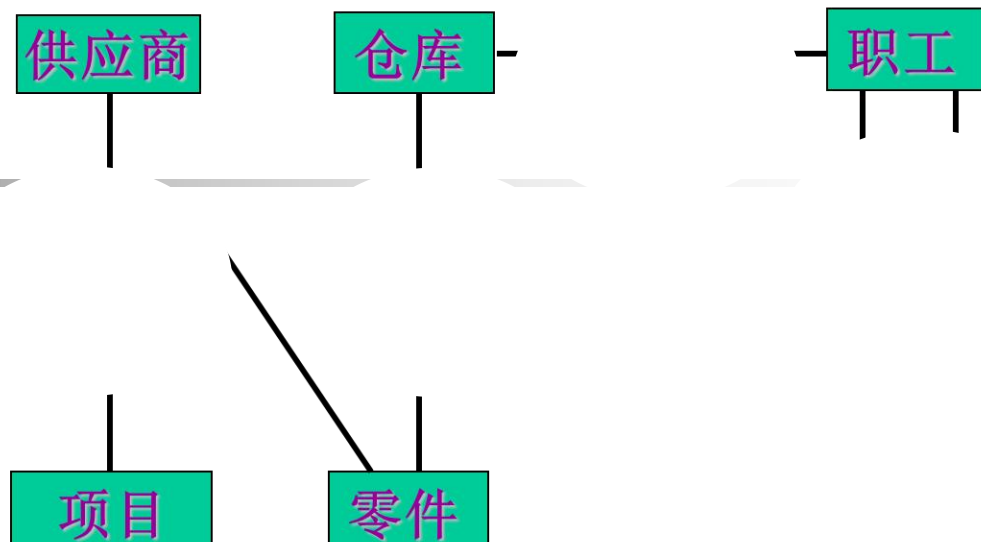


2. 确定实体的属性及键（主属性）



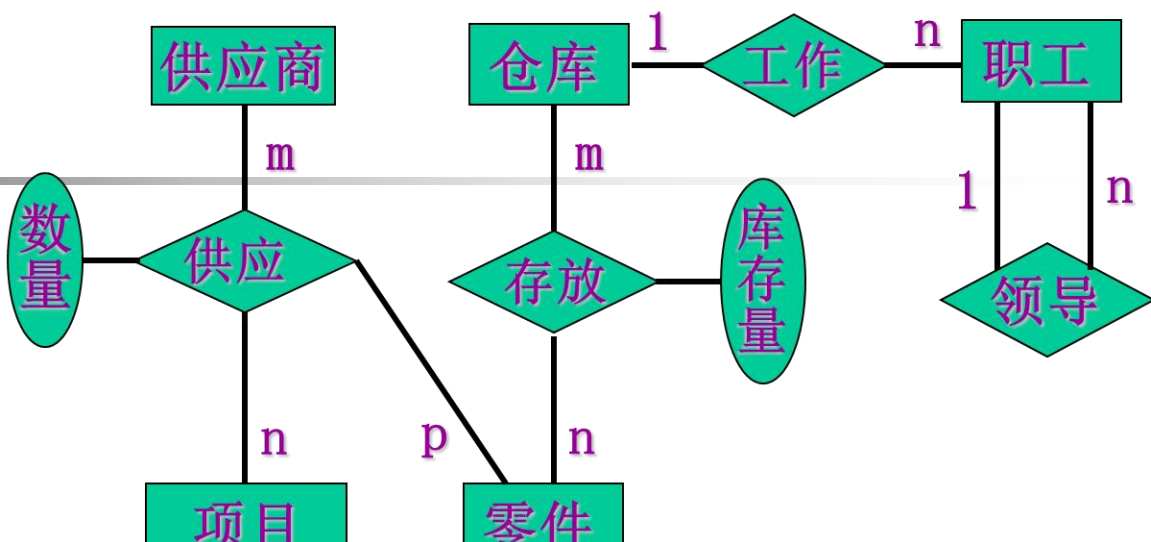


3. 确定实体的联系



- (1) 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中；仓库有仓库号、仓库类型和面积，零件有零件号、名称、规格、单价、描述等属性。
- (2) 一个职工只能在一个仓库工作，一个仓库有多个职工当保管员；职工有职工号、姓名、性别、职务、出生年月属性。
- (3) 职工之间有领导与被领导关系，仓库主任领导若干保管员；
- (4) 一个供应商可以供应若干项目多种零件；而一个项目可以使用不同供应商供应的多种零件；一种零件可由不同供应商供给多个工程项目。供应商有供应商号、姓名、住址、电话、帐号，项目有项目号、预算、开工日期等属性。

4. 确定联系的基数和属性



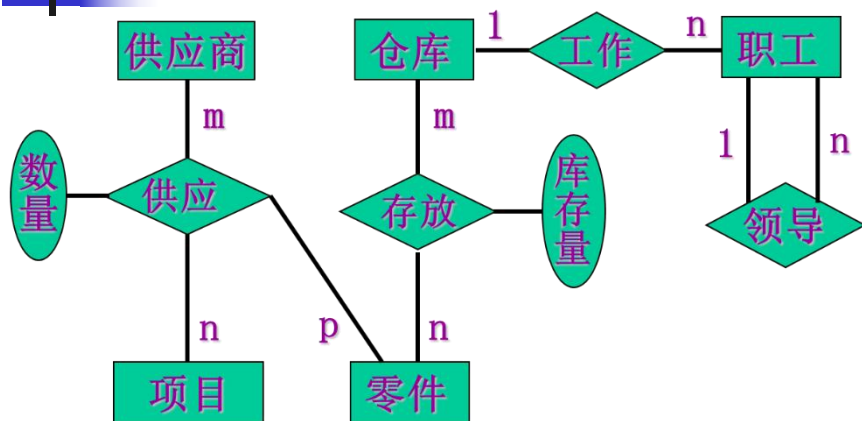
(1) 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中；仓库有仓库号、仓库类型和面积，零件有零件号、名称、规格、单价、描述等属性。

(2) 一个职工只能在一个仓库工作，一个仓库有多个职工当保管员；职工有职工号、姓名、性别、职务、出生年月属性。

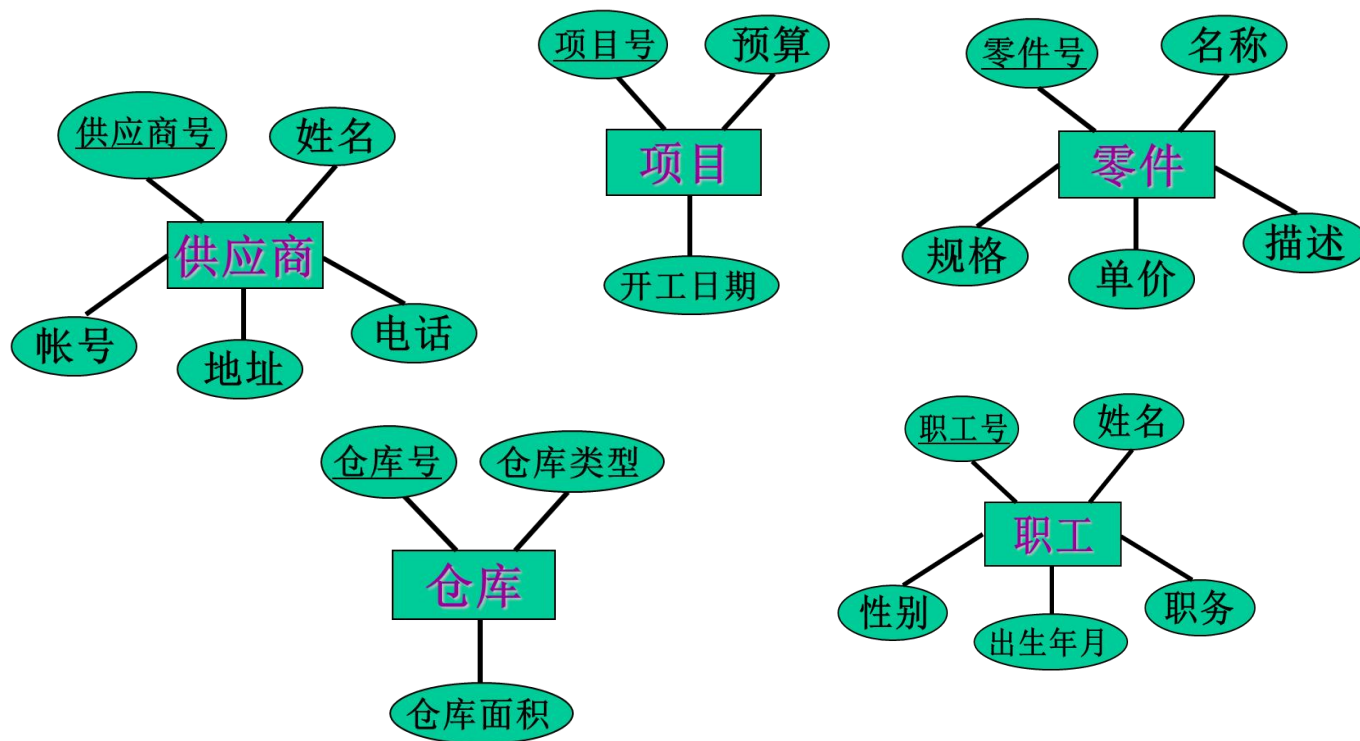
(3) 职工之间有领导与被领导关系，仓库主任领导若干保管员；

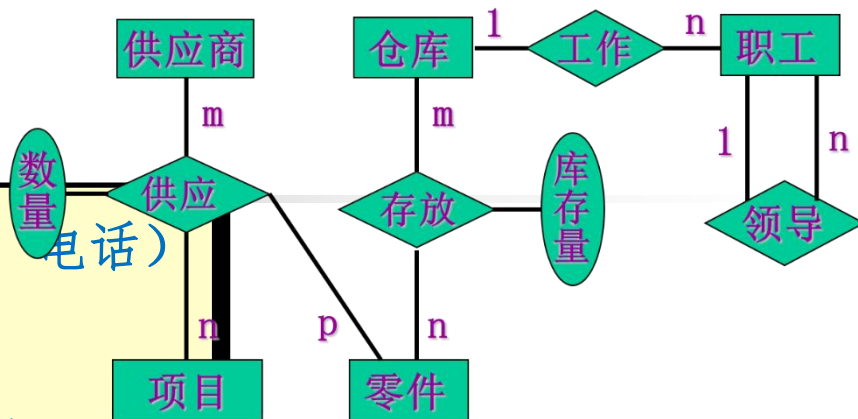
(4) 一个供应商可以供应若干项目多种零件；而一个项目可以使用不同供应商供应的多种零件；一种零件可由不同供应商供给多个工程项目。供应商有供应商号、姓名、住址、电话、帐号，项目有项目号、预算、开工日期等属性。

讨论



对应的关系模式？





供应商（供应商号，姓名，帐号，地址，电话）

项目（项目号，预算，开工日期）

零件（零件号，名称，规格，单价，描述）

职工（职工号，姓名，性别，职务，出生年月）

仓库（仓库号，仓库类型，仓库面积）

供应（供应商号，项目号，零件号，数量）

存放（零件号，仓库号，库存量）

工作（职工号，……，仓库号）

管理（职工号，……，领导）

职工（职工号，姓名，性别，职务，出生年月，仓库号，领导）