

# 关系模型



徐辰  
华东师范大学  
数据科学与工程学院  
[cxu@dase.ecnu.edu.cn](mailto:cxu@dase.ecnu.edu.cn)

# 什么是系统？

用户需求

应用软件

系统软件

# 系统的要素

- 功能
- 性能
- 易用性



# 数据模型

- 如何描述信息?
  - 如何描述信息需求?
- 考虑两类数据模型
  - Key-Value Pairs
  - Relational Model by E. F. Codd

# Key-Value Pairs

- 数据组织: <key, value>
- 数据访问方式:
  - Put(key, value)
  - Get(key)
  - Delete(key)
- 例子:

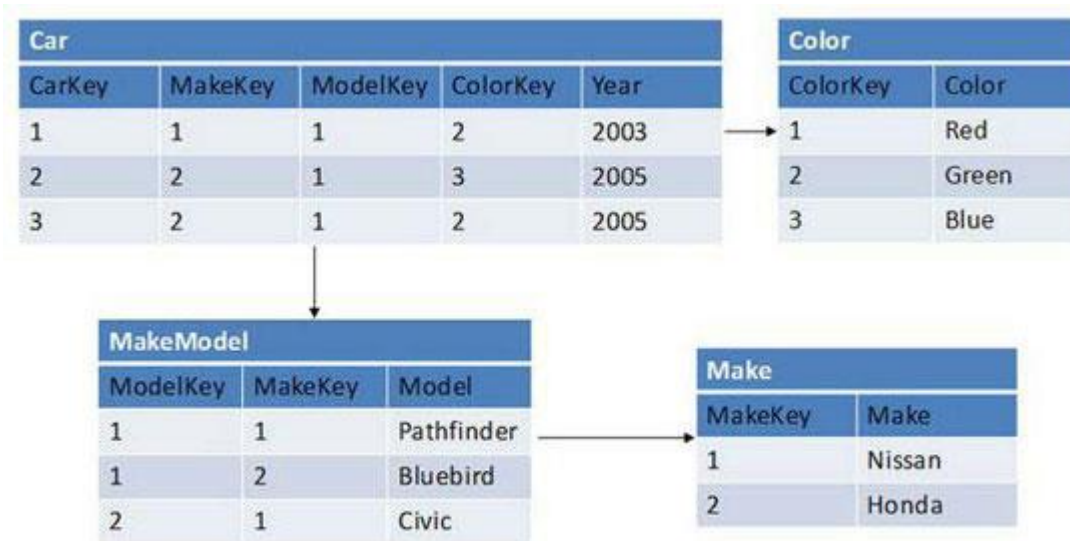
Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year: 2005 Transmission: Auto



# Relational Model

- 数据组织：关系/表
- 数据访问方式：关系代数/SQL
- 例子：

ORACLE®



# 表达能力

- 系统能够满足哪些信息需求？
  - 用户可以向系统表达哪些需求，并从系统直接获得答案？
- 例子
  - 1号汽车是什么颜色的？
  - 2003年的丰田汽车有多少种款式？

# 关系模型具备很强的表达能力

- E. F. Codd提出关系模型时使用的概念 – Relational Calculus
  - Relational Calculus的表达能力比较接近First Order Logic (FOL)
- 不太严谨的表达能力比较
$$KV < Relation < FOL < Turing Machine$$

<https://www.dbis.informatik.uni-goettingen.de/Teaching/DB/db-rel-calc.pdf>



# 关系演算/Relational Calculus

- 元组关系演算
  - $\{ t : \{ \text{name} \} \mid \exists s : \{ \text{name}, \text{wage} \} ( \text{Employee}(s) \wedge s.\text{wage} = 50.000 \wedge t.\text{name} = s.\text{name} ) \}$
  - $\{ t : \{ \text{supplier}, \text{article} \} \mid \exists s : \{ s\#, \text{sname} \} ( \text{Supplier}(s) \wedge s.\text{sname} = t.\text{supplier} \wedge \exists p : \{ p\#, \text{pname} \} ( \text{Product}(p) \wedge p.\text{pname} = t.\text{article} \wedge \exists a : \{ s\#, p\# \} ( \text{Supplies}(a) \wedge s.s\# = a.s\# \wedge a.p\# = p.p\# ) ) \}$
- 域关系演算

# 关系代数、关系演算和SQL

- 本质一样
  - 关系演算是逻辑表示。
  - 关系代数是实现方案。
  - SQL是用户界面。

# 关系模型能提供较强的智能

- 智能体现在需求表述的简洁性上。
- 声明式语言 / Declarative Language
  - 声明式语言是简洁的，因而具有较高的智能。
  - 例子
    - SQL / Relational Calculus
    - Prolog
    - SPARQL

# 关系模型的使用

- 设计关系模式 (schema)
- 用SQL语言描述信息需求

# 关系模式的设计

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	<del>洗发水</del>	<del>20元</del>	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	<del>张三</del>	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

# 差的数据模式带来的问题

- 数据冗余 (Redundancy)
- 更新异常 (Update Anomaly)
- 插入异常 (Insertion Anomaly)
- 删除异常 (Deletion Anomaly)

**那什么是好的数据模式？**

# 第一范式 (1<sup>st</sup> Normal Form)

- 关系模式一定满足第一范式(1NF)。

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	洗发水	20元	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	张三	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

# 函数依赖 (Functional Dependency)

- 设 $R(U)$ 为属性集 $U$ 上的关系模式。 $X, Y$ 是 $U$ 的子集。若对于 $R(U)$ 的任意一个在现实世界中可能的关系 $r$ ,  $r$ 中不可能存在两个元组在 $X$ 上的属性值相等, 而在 $Y$ 上的属性值不等, 则称 **$Y$ 函数依赖于 $X$** , 记作 $X \rightarrow Y$ 。



# 函数依赖 (Functional Dependency)

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	洗发水	20元	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	张三	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

订单号, 客户号  $\rightarrow$  客户号

订单号, 客户号, 商品号  $\rightarrow$  商品号, 商品名称

订单号  $\rightarrow$  客户号

订单号  $\rightarrow$  客户名字

商品号  $\rightarrow$  商品名称

商品号  $\rightarrow$  商品名称, 价格

} 完全函数依赖

# 完全函数依赖 (Full FD)

- 在 $R(U)$ 中, 如果 $X \rightarrow Y$ , 并且对于 $X$ 的任意一个真子集 $X'$ ,  $X' \rightarrow Y$ 都不成立, 则称 **$Y$ 完全依赖于 $X$** 。

# 键/码 (Key)

- 设 $R(U)$ 为属性集 $U$ 上的关系模式。 $X$ 是 $U$ 的子集。如果 $X \rightarrow U$ ，并且不存在 $X$ 的子集 $X'$ 使得 $X' \rightarrow U$ ，那么 $X$ 为 $R(U)$ 的**键/码** (key)。
- 在 $R(U)$ 中可能存在多个键，我们人为指定其中的一个键为**主键** (primary key)。
- 包含在任意一个键中的属性，称为**主属性** (prime attribute)。

# 第二范式 (2nd Normal Form)

- 定义：关系模式R满足第二范式，当且仅当，R满足第一范式，并且，R的每一个非主属性都完全依赖于R的每一个键。

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	洗发水	20元	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	张三	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

# 第二范式 (2nd Normal Form)

- 不满足第二范式的关系模式：

学号	学院	住址	课程号	成绩
00001	信息	知行1	MA815	89
00002	金融	品元3	MA815	69
00003	法律	东风6	L762	90
00001	信息	知行1	CS453	90
...	...	...	...	...

学号, 课程号 → 成绩

学号 → 学院

学院 → 住址

# 第三范式 (3rd Normal Form)

- 定义：关系模式R满足第三范式，当且仅当，R中不存在这样的键X，属性组Y和非主属性Z，使得 $X \rightarrow Y$ ， $Y \rightarrow Z$ 成立，且 $Y \rightarrow X$ 不成立。

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	洗发水	20元	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	张三	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

# 第三范式 (3rd Normal Form)

- 满足第三范式的关系模式：

客户号	客户名字
007	张三
010	小明
025	大宝
...	...

商品号	商品名称	价格
0815	洗发水	20元
0762	优盘	50元
0453	电吹风	150元
...	...	...

订单号	客户号	商品号	数量	日期
00001	007	0815	1	2013-1-12
00002	010	0815	1	2012-12-23
00003	025	0762	10	2012-12-15
00004	007	0453	1	2012-12-1
...	...	...	...	...

定理：满足  
第三范式的  
模式必满足  
第二范式

# 第二范式与第三范式

- 满足第三范式意味着R中的函数依赖不传递
- R的每一个非主属性都完全依赖于R的每一个键



# BC范式 (Boyce Codd Normal Form)

- 定义：关系模式R满足BC范式，当且仅当，对任意一个属性集A，如果存在不属于A一个属性X，使得X函数依赖于A，那么所有R的属性都函数依赖于A。
- 定理：任何满足BC范式的关系模式都满足第三范式

# BC范式 (Boyce Codd Normal Form)

- 同时满足第三范式和BC范式:

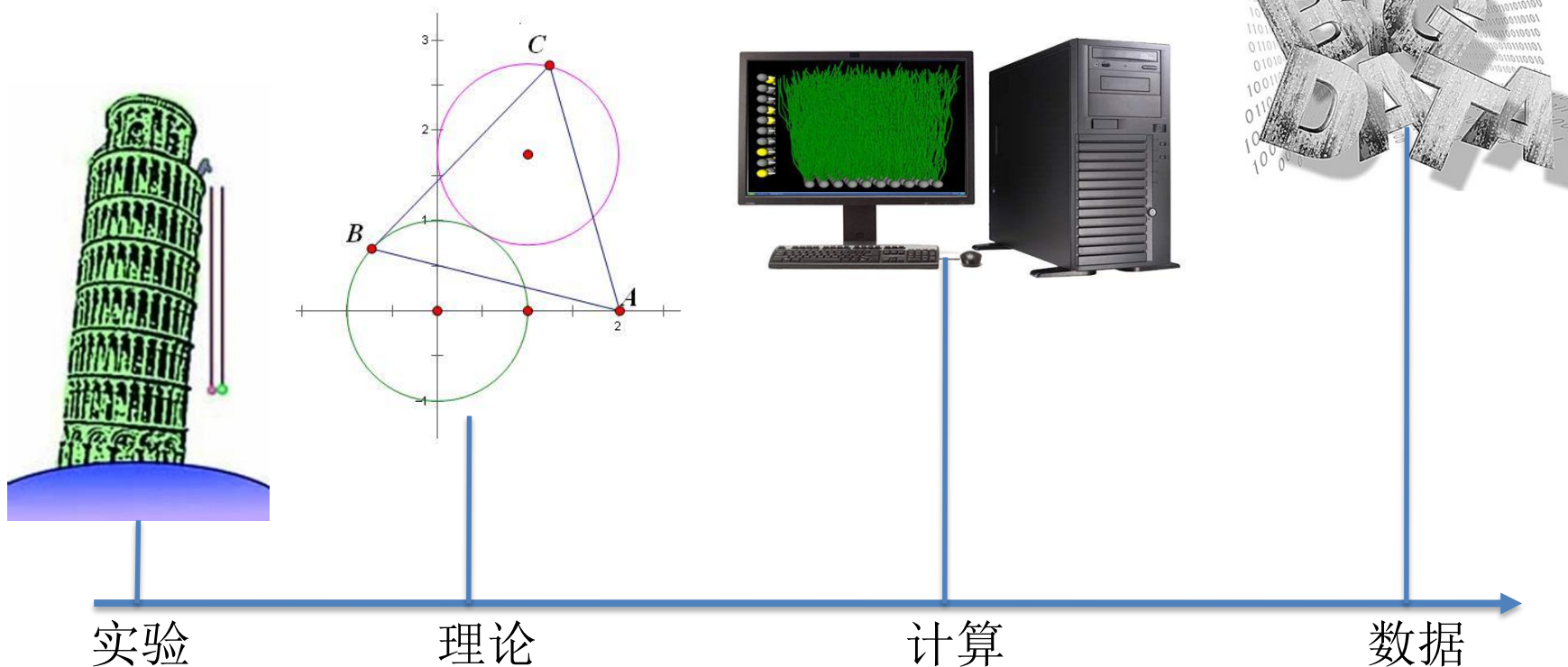
客户号	客户名字
007	张三
010	小明
025	大宝
...	...

商品号	商品名称	价格
0815	洗发水	20元
0762	优盘	50元
0453	电吹风	150元
...	...	...

订单号	客户号	商品号	数量	日期
00001	007	0815	1	2013-1-12
00002	010	0815	1	2012-12-23
00003	025	0762	10	2012-12-15
00004	007	0453	1	2012-12-1
...	...	...	...	...

# 大数据第四范式!=数据库范式

- Jim Gray：人类自古以来，在科学研究上先后历经了实验、理论、计算和数据四种范式

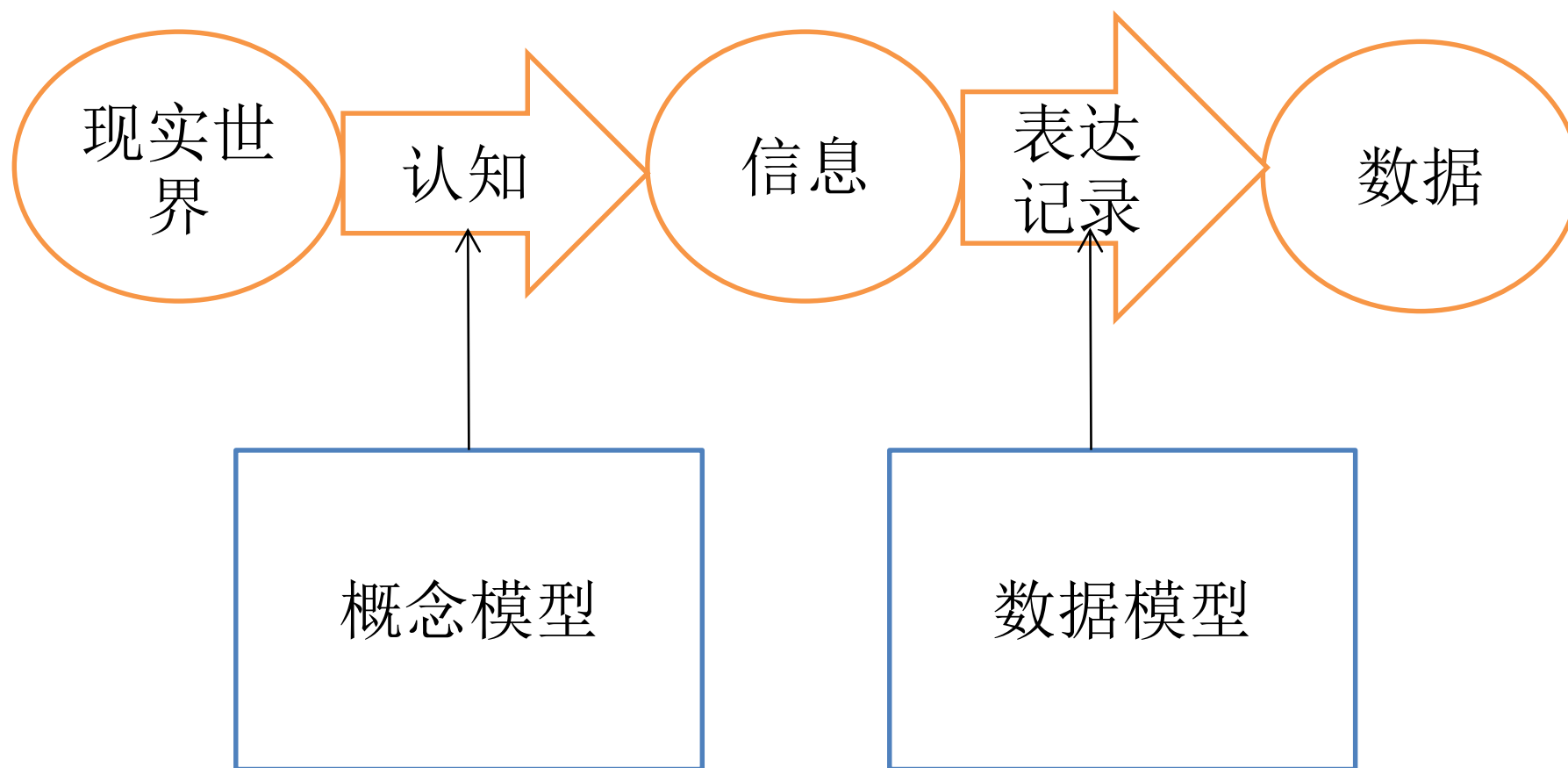


**如何设计出好的模式？**

# 数据库设计流程

- 需求分析
  - 了解用户需求，确定软件的基本功能。
- 概念模型设计
  - 确定数据库需要记录哪些信息，并通过概念进行描述。
- 逻辑结构设计
  - 确定数据库模式（关系模式）。
- 数据库物理设计
  - 确定数据的存储方式、索引的使用、系统配置等等。
- 数据库实施
  - 安装数据库、导入数据、调试、运行。

# 概念模型



# 日常生活中的概念模型

- “类别” 是常见的概念模型
  - 植物、动物
  - 小学、中学、大学
- “度量” 是另一种概念模型
  - 强弱
  - 高低

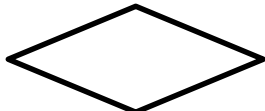
# 用于数据库设计的概念模型：ER图

- E: Entity (实体)
- R: Relationship (联系)
- ER图模型由Peter Chen于1976年提出。
- 基本概念：
  - 世界由实体组成。
  - 实体由属性刻画。
  - 实体之间存在联系。
  - 实体可分类，类别确定实体的属性。

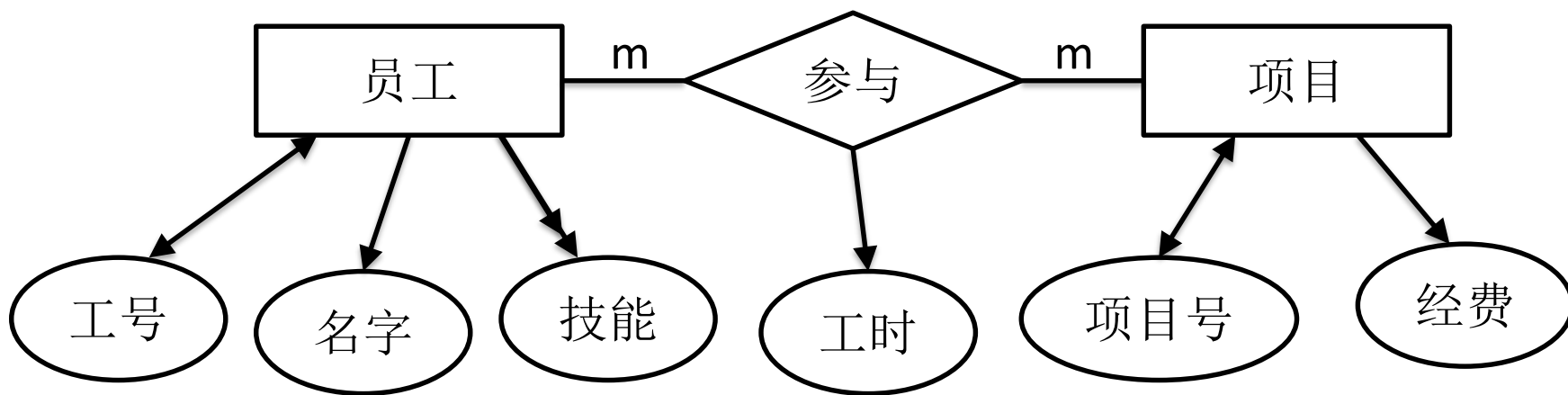





# ER图的表示方法

实体类: 

联系类: 

属性: 

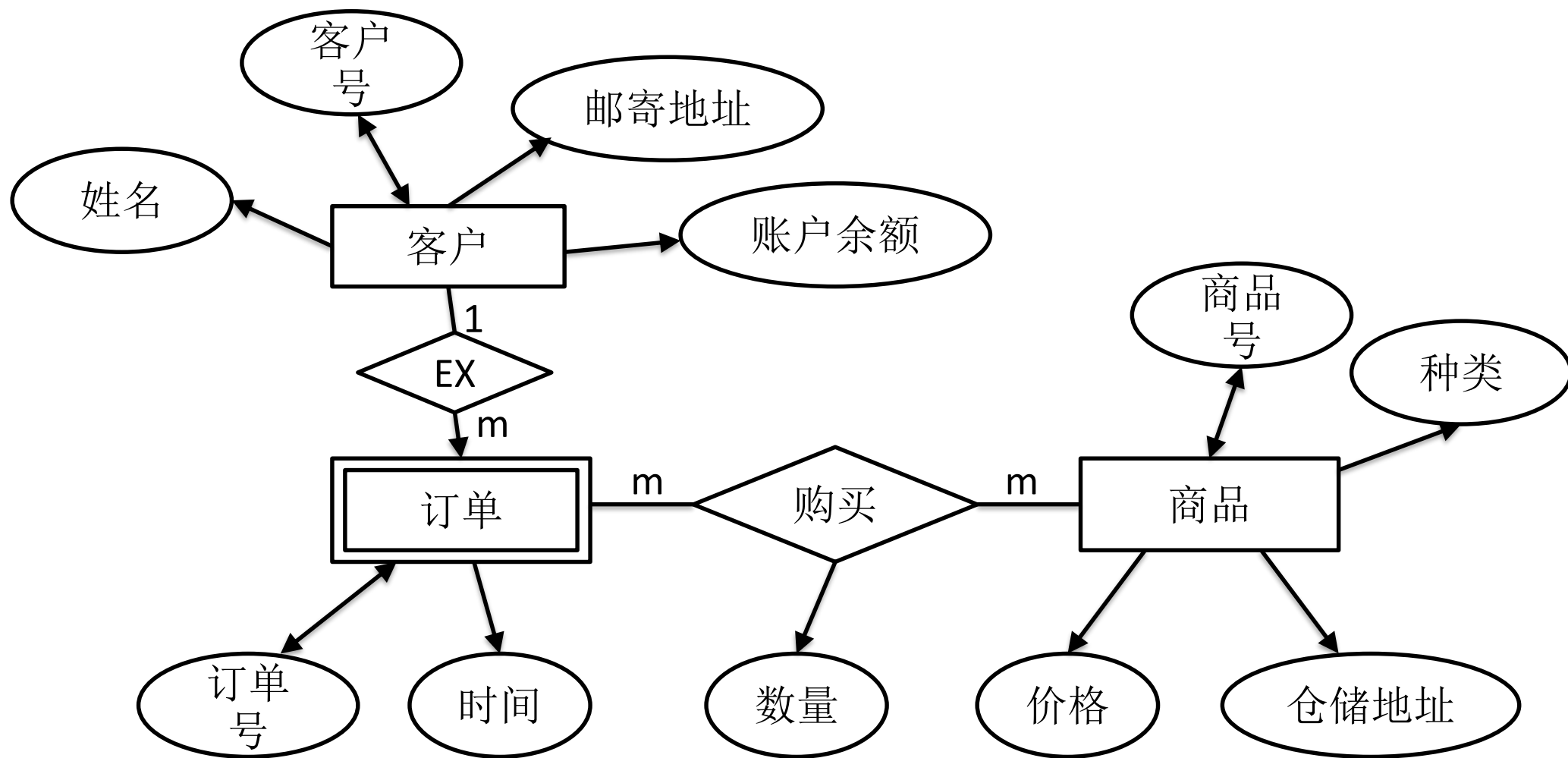


 唯一属性  
 单值属性  
 多值属性

# 用ER图对以下场景进行数据建模

- 网上购物的场景：
  1. 系统需要查询客户信息，包括：客户姓名、邮寄地址、账户余额。
  2. 系统需要打印购物信息，包括：购买商品的种类和数量、购买客户的姓名和地址、购买时间。
  3. 系统需要查询商品信息，包括：商品种类、商品价格、仓储地址。

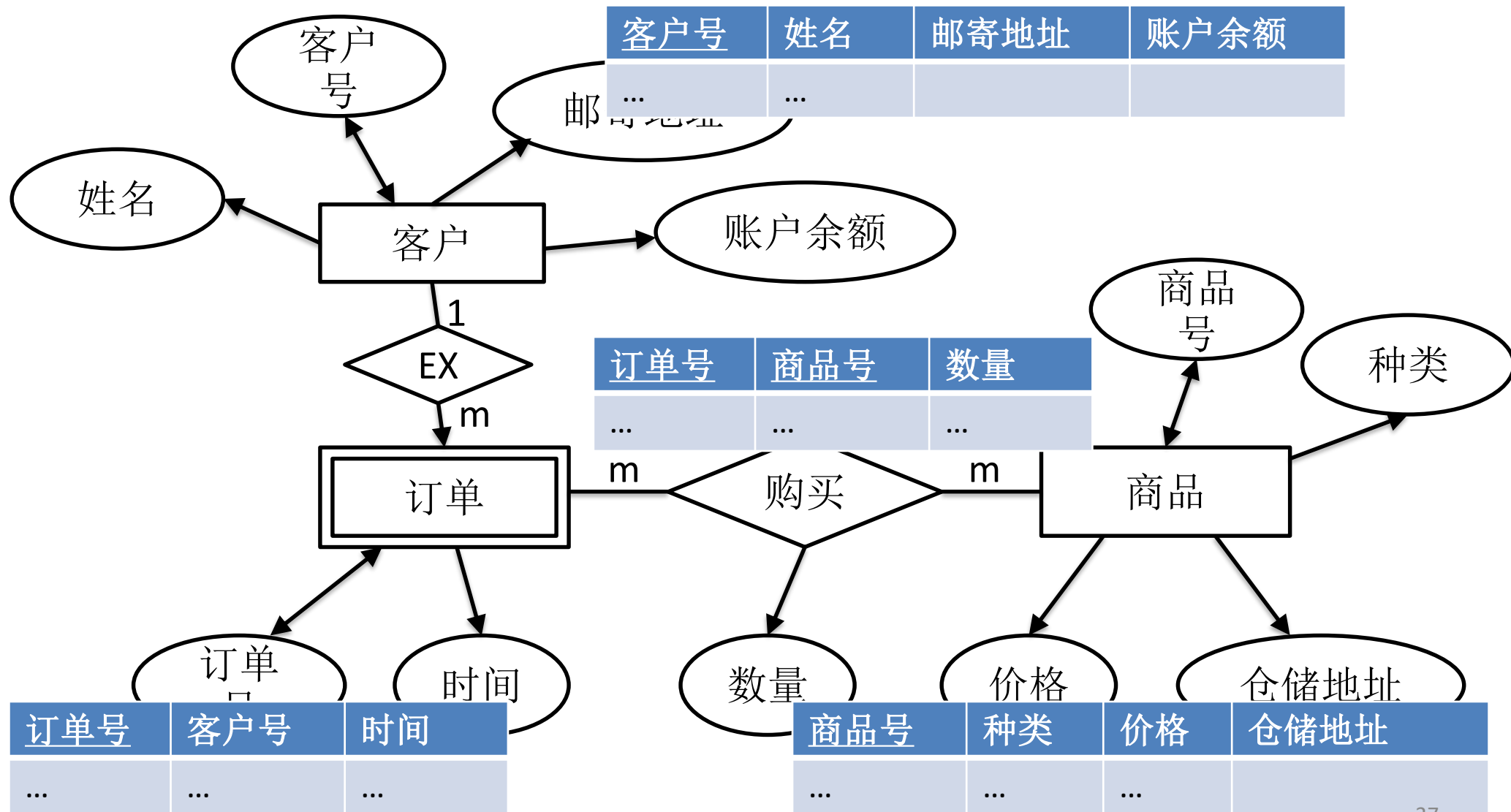
# 一个合理的ER图



# 从ER图到关系模式

1. 补充ER图的信息
  - 定义所有实体类和联系类的ID属性。
2. 将每个实体类转换为一个或多个关系
  - ID属性和所有的唯一属性和单值属性组成一个关系。
  - 每一个多值属性都和ID属性组成一个关系。
3. 将每个联系类转换为一个或多个关系
  - ID属性和所有的唯一属性和单值属性组成一个关系。
  - 每一个多值属性都和ID属性组成一个关系。
  - 对于一对多的联系，可以将联系表与起确定作用的实体表合并。
4. 设置外键约束
  - 联系类、子类、弱实体类都需要设置外键约束。

# 网上购物的场景的关系模式



# 通过ER图能得出以下的模式吗？

订单号	客户号	客户名字	商品号	商品名称	价格	数量	日期
00001	007	张三	0815	洗发水	20元	1	2013-1-12
00002	010	小明	0815	洗发水	20元	1	2012-12-23
00003	025	大宝	0762	优盘	50元	10	2012-12-15
00004	007	张三	0453	电吹风	150元	1	2012-12-1
...	...	...	...	...	...	...	...

# 总结

- 关系模型的设计兼顾表达能力与自动化程度。
- 关系模型在使用中的经验总结进一步提升了它的实用性。

# 思考题

- 既然关系模型的设计如此周全，为什么会有key-value数据库的存在？



# Credits

- Slides from
  - 周烜
  - 林子雨