

# 第三章

# 关系模型

简单即是有力

-----燧人氏语录

Albert Einstein: Any intelligent fool can make things bigger and more complex...

It takes a touch of genius and a lot of courage to move in the opposite direction

牛顿：把简单的事情考虑得很复杂，可以发现新领域；  
把复杂的现象看得很简单，可以发现新定律



$$E = MC^2$$

电子 汤姆逊

质子 卢瑟福

中子 查德威克

慢中子 费米

裂变 哈恩 迈特纳

U235 波尔

曼哈顿工程 奥本海默

小男孩 广岛

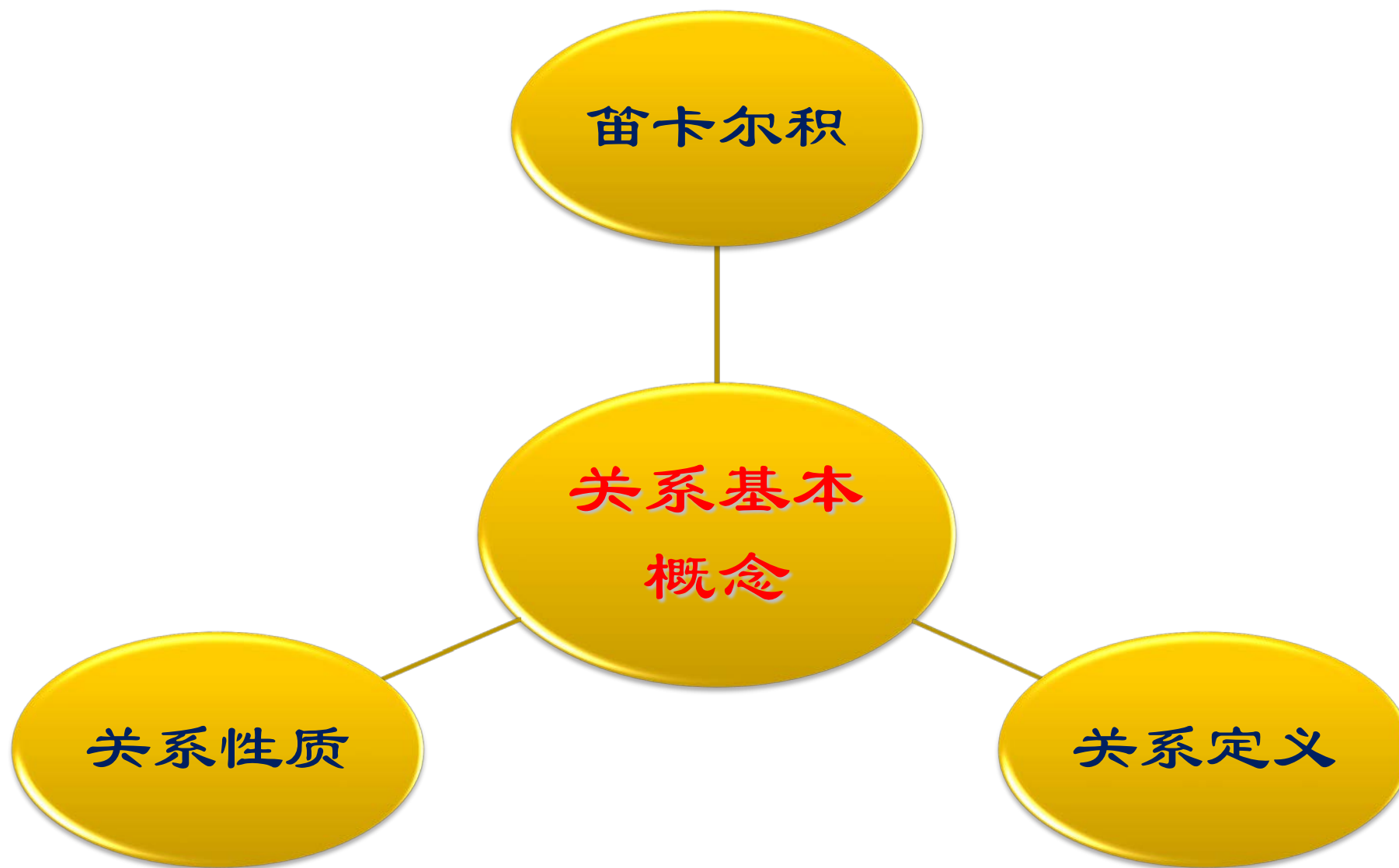
居里家族



# TABLE

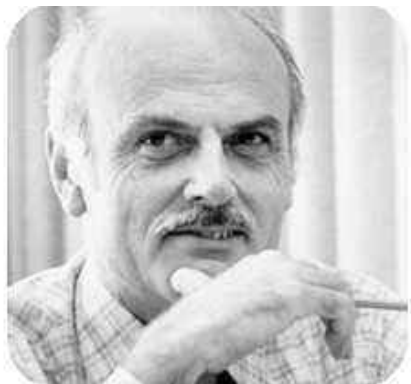
*An arrangement of words, numbers, or signs, or combinations of them, as in parallel columns, to exhibit a set of facts or relations in a definite, compact, and comprehensive form; a synopsis or scheme.*

— — Webster's Dictionary of the English Language





# 关系模型的诞生



E.F.Codd于70年代初提出  
关系数据理论，他因此获  
得1981年的ACM图灵奖

关系理论建立在集合代数  
理论基础之上，有着坚实  
的数学基础

1970: "A Relational Model of data for Large Shared Data Banks"



提出关系代数和关系演算



1972: 1NF, 2NF, 3NF



1974: BCNF

# 关系模型一统江湖

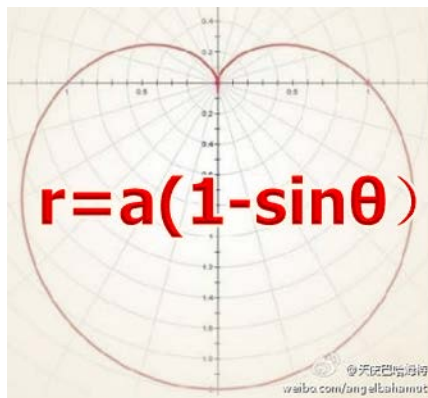
## 早期代表系统

- System R：由IBM研制
- INGRES：由加州Berkeley分校研制

## 目前主流的商业数据库系统

- Oracle, SQL Server, DB2
- MySQL, PostgreSQL
- OceanBase, GaussDB, 达梦
- Access, SQLite

# 笛卡尔积 (Cartesian Product)



## 域 (Domain)

具有相同数据类型的一组值的集合  
如整数集合、字符串集合、全体学生集合

一组域  $D_1, D_2, \dots, D_n$  的笛卡尔积为:

$$D_1 \times D_2 \times \dots \times D_n = \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i = 1, \dots, n \}$$



# 笛卡尔积

笛卡尔积的元素  $(d_1, d_2, \dots, d_n)$

称作 **n元组** (tuple)

元组的每一个值  $d_i$

称作 **分量** (component)

若  $D_i$  的基数为  $m_i$

则笛卡尔积的 **基数** 为  $\prod_{i=1}^n m_i$

# 笛卡尔积：可能的世界

老师集合  $T = \{t_1, t_2\}$

学生集合  $S = \{s_1, s_2, s_3\}$

课程集合  $C = \{c_1, c_2\}$

- $T \times S \times C$  是个三元组集合，元组个数为  $2 \times 3 \times 2$
- 它是所有可能的(老师，学生，课程)元组集合

笛卡尔积可表为二维表的形式

老师	学生	课程
$t_1$	$s_1$	$c_1$
$t_1$	$s_1$	$c_2$
$t_1$	$s_2$	$c_1$
...	...	...
$t_2$	$s_3$	$c_2$

# 关系：笛卡尔积的子集

笛卡尔积  $D_1 \times D_2 \times \cdots \times D_n$  的子集称作

在域  $D_1, D_2, \dots, D_n$  上的 **关系**

用  $R(D_1, D_2, \dots, D_n)$  表示

$R$  是关系的名字， $n$  是关系的度或目

- **guanxi** 关系 （社交网络系统及有影响力的人际关系系统）
- **add oil** 加油
- **hongbao** 红包
- **Gelivable** 给力的；不可思议的
- **Funny mud pee**

# 关系：实际的世界

关系是笛卡尔积中**有意义的**子集

有意义的体现

关系名体现了

现实实体或联系

关系集合包含了真正存在  
的实体或发生的联系

Teach( 老师、学生、课程)  
Assistant( 老师、学生、课程)

老师	学生	课程
t <sub>1</sub>	s <sub>1</sub>	c <sub>1</sub>
t <sub>1</sub>	s <sub>1</sub>	c <sub>2</sub>
t <sub>1</sub>	s <sub>2</sub>	c <sub>1</sub>
t <sub>2</sub>	s <sub>3</sub>	c <sub>2</sub>

元组

属性

# 过江之鲫：关系实例的数目

$D1 := \{A, B, C, D, E\}$

$D2 := \{1, 2, 3, 4\}$

$D3 := \{\text{甲}, \text{乙}, \text{丙}, \text{丁}\}$

$D1 \times D2 \times D3$  包含多少个关系？

$\{A, B, C, D, E, 1, 2, 3, 4, \text{甲}, \text{乙}, \text{丙}, \text{丁}\}$

划分为多少个域，每个域多少个元素，  
使得笛卡尔积包含的关系数目最多？

# 关系的性质

*P1:*

列是同质的，是同一类型的数据，即每一列中的分量来自同一域

*P2:*

不同的列可以来自同一域，每列必须有不同的属性名

*P3:* 行列的顺序无关紧要

*P4:*

任意两个元组不能完全相同（集合内不能有相同的两个元素）

*P5:*

每一分量必须是不可再分的数据，称其为满足第一范式（1NF）的关系



# 关系的性质

## 列同质

Tno	Sno	Cno
t1	s1	c1
t1	t2	c2

不同质违反了关系对实体联系的表达，以及对实体属性的刻画

## 来自同一域的不同列

Eno	ENAME	MGR
e1	TOM	e2
e2	jerry	null

什么时候两个列会来自同一个域？

- 一元联系
- 类型相同的属性

# 关系的性质

分量不可再分

嵌套关系

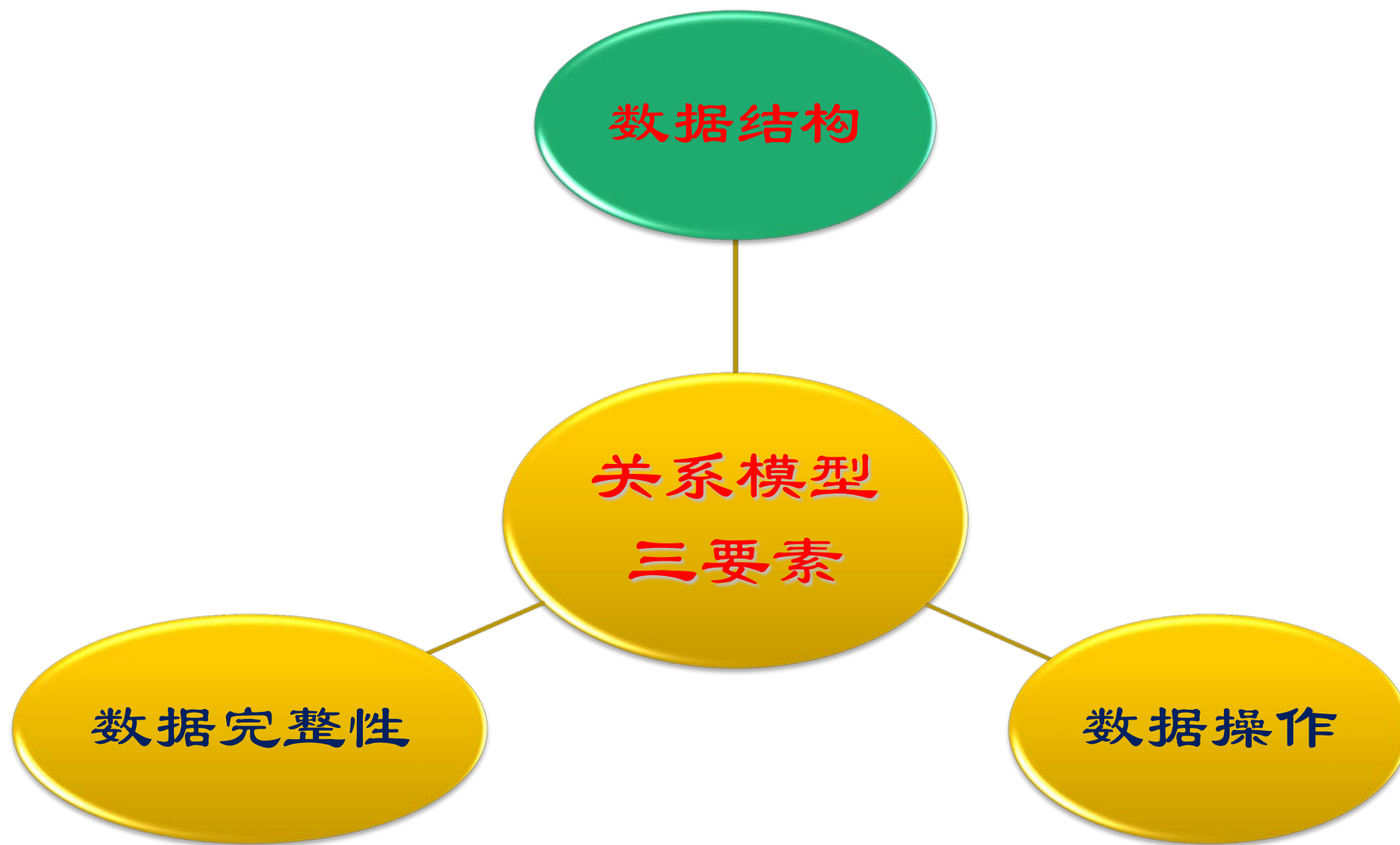
Sno	Cno
s1	{c1,c2,c3}

1NF关系

Sno	Cno
s1	c1
s1	c2
s1	c3

从查询的角度，深入分析嵌套关系的利弊

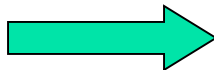
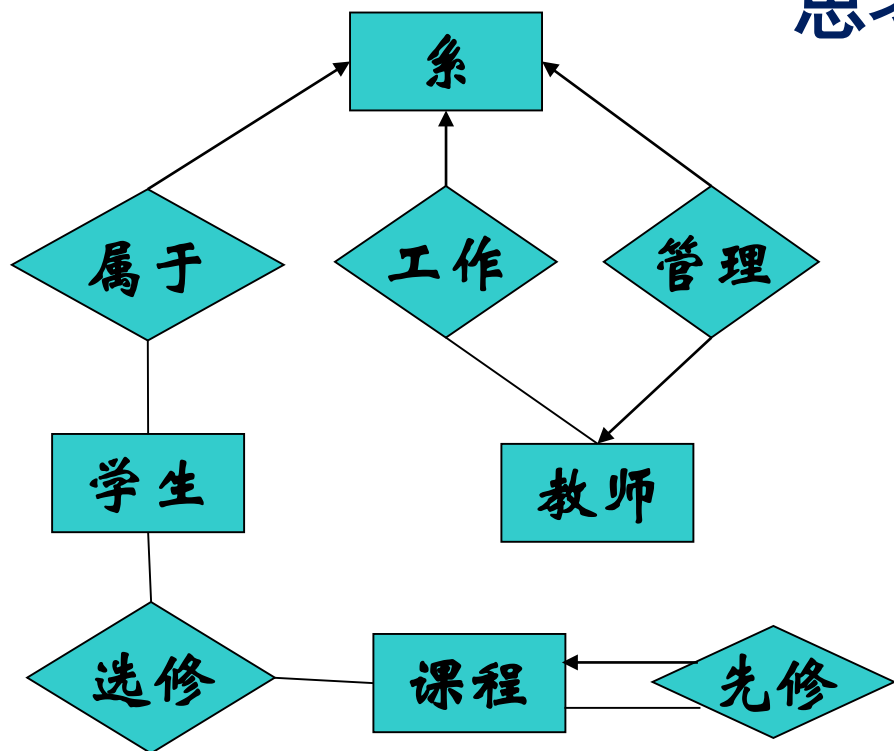
声明性查询 VS 路径式查询



# 关系模型：数据结构

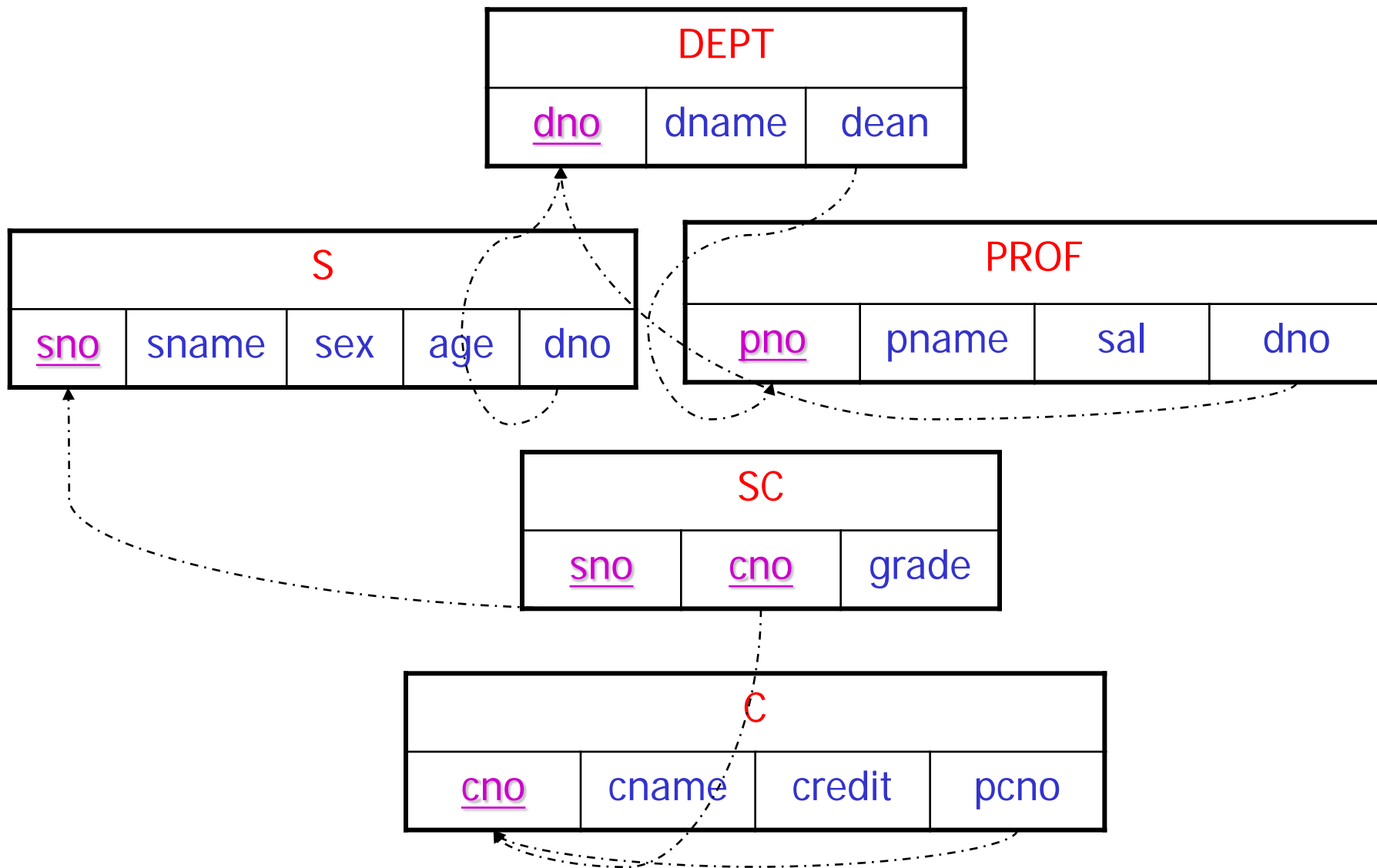
单一的数据结构——关系  
实体集、联系都表示成关系

思考：单一数据结构带来的好处是什么？



DEPT(dno , dname , dean)  
S(sno , sname, sex, age, dno)  
C(cno , cname , pcno , credit)  
SC(sno,cno , grade)  
PROF(pno , pname, sal, dno)

# 关系模型：数据结构



# 关系模型：码

## 候选码 (Candidate Key)

- 关系中的一个属性组，其值能唯一标识一个元组
- 若从属性组中去掉任何一个属性，它就不具有这一性质了，这样的属性组称作候选码

(如DEPT中的dno, dname都可作为候选码)

任何一个候选码中的属性称作**主属性**

(如SC中的sno, cno, DEPT中的dno, dname)



# 关系模型：码

## 主码 (PK: Primary Key)

进行数据库设计时，从一个关系的  
多个候选码中选定一个作为主码

(如可选定dno作为DEPT的主码)

## 外码 (FK: Foreign Key)

关系R中的一个属性组，它不是R的码，但它与另一个关系S的码相对应，称这个属性组为R的外码

(如S关系中的dno属性)

## 关系模式：

关系的描述，记作 $R(A_1, A_2, \dots, A_n)$ ，包括：

- 关系名、关系中的属性名
- 属性向域的映象，通常说明为属性的类型、长度等
- 属性间的数据依赖关系，比如在特定的时间和教室只能安排一门课

关系模式是型，是稳定的

## 关系：

某一时刻对应某个关系模式的内容(元组的集合)

关系是某一时刻的值，是随时间不断变化的

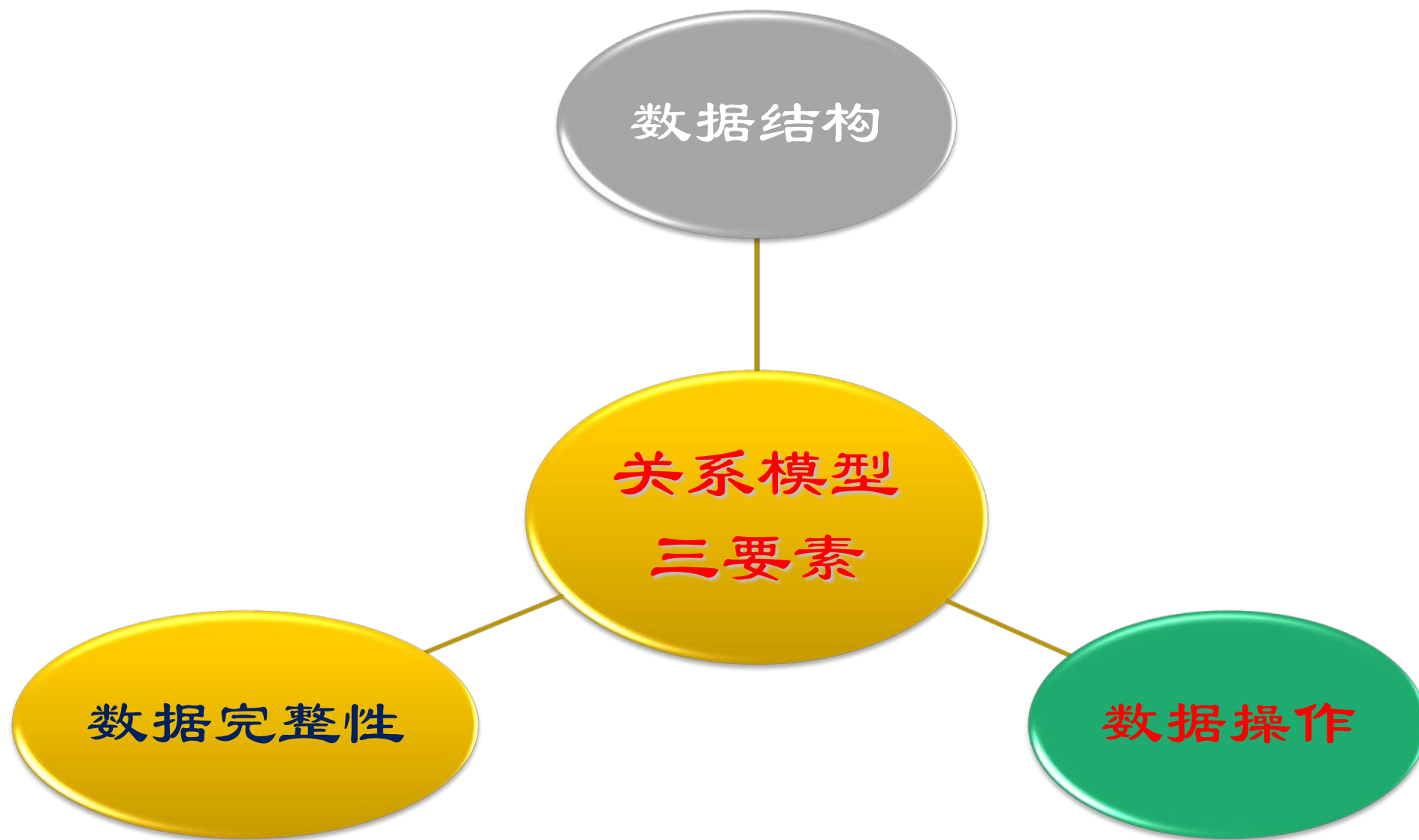
# 关系数据库的构成

## 关系数据库的型：

是关系模式的集合，即数据库描述。称作数据库的**内涵**(Intension)

## 关系数据库的值：

是某一时刻关系的集合。称作数据库的**外延**(Extension)



# 关系操作

关系操作是集合操作

操作的对象及结果都是集合

是一次一集合(Set-at-a-time)的方式

非关系型的数据操作方式是一次一记录

(Record-at-a-time)

# 关系数据语言的特点

## 一体化

对象单一，都是关系，因此操作符也单一

## 非过程化

用户只需提出“做什么”，无须说明“怎么做”

存取路径的选择和操作过程由系统自动完成

## 面向集合的存取方式

操作对象是一个或多个关系，结果是一个新的关系  
(一次一关系)

非关系系统是一次一记录的方式



# 抽象的关系模型查询语言

关系  
代数

用预定义操作算子的  
执行序列来表达查询

关系  
验算

用谓词来表达查询，只  
需描述所需信息的特性

元组关系验算

谓词变元是元组

域关系验算

谓词变元是属性列

# 具体系统数据库中的关系操作语言

SQL

介于关系代数和关系演算之间

由IBM公司在研制System R时提出

QUEL

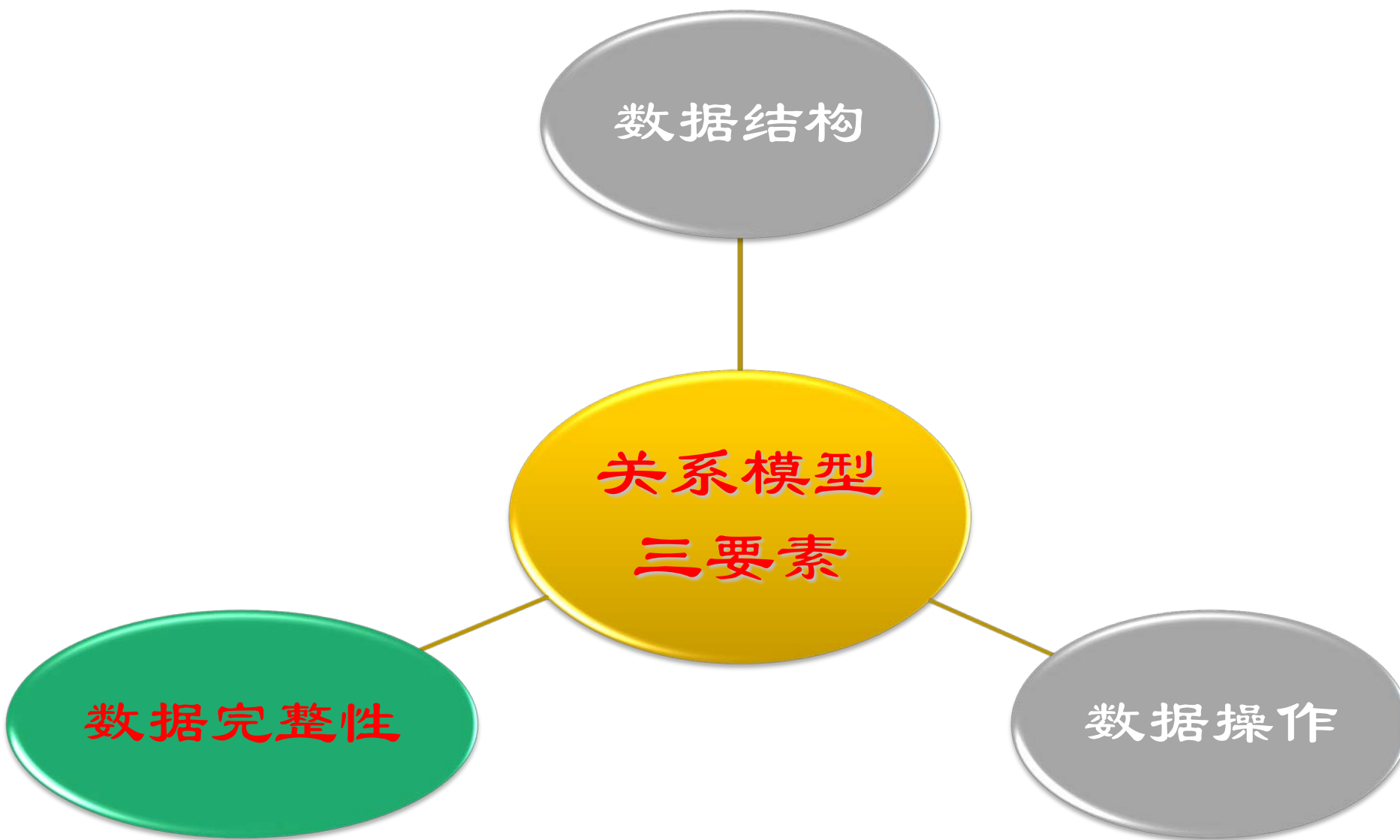
基于Codd提出的元组关系演算语言ALPHA

在INGRES上实现

QBE

基于域关系演算

由IBM公司研制



关系模型

完整性

实体  
完整性

参照  
完整性

用户定义  
完整性

# 实体完整性

关系的主码中的属性值不能为空值

空值：不知道或无意义

**意义：**关系对应到现实世界中的实体集，元组对应到实体，实体是相互可区分的，通过主码来唯一标识，若主码为空，则出现不可标识的实体，这是不容许的

DNA	NAME
null	张三
ATGC	张三

# 参照完整性

如果关系 $R_2$ 的外码 $F_k$ 与关系 $R_1$ 的主码 $P_k$ 相对应，则 $R_2$ 中每个元组的 $F_k$ 值或者等于 $R_1$ 中某个元组的 $P_k$ 值，或者为空值

如果关系 $R_2$ 的某个元组 $t_2$ 参照了关系 $R_1$ 的某个元组 $t_1$ ，则 $t_1$ 必须存在，也即必须与客观存在的实体发生联系

sno	sname	dno		dno	dname
s1	TOM	d1		d1	maths
s2	JERRY	null	允许为null		
s3	BOB	d3	违反参照完整性	d2	physics



# 用户定义的完整性

用户针对具体应用环境定义的完整性约束条件

- sno要求是8位整数，首位是0或1
- 飞行员的飞行里程与星级评定
- 选课人数不能少于10人，多于100人
- 在本地纳税记录超过5年才有购房资格
- 婚姻登记必须购买百年好合保险.....

实体完整性和参照完整性由系统自动支持

系统提供定义和检验用户定义的完整性的机制

# 课堂练习：关系模型完整性例子

供应商关系S (主码是“供应商号”)

供应商号	供应商名	所在城市
B01	红星	北京
S10	宇宙	上海
T20	黎明	天津
Z01	立新	重庆

零件关系P (主码是“零件号”，外码是“供应商号”)

零件号	颜色	供应商号
010	红	B01
312	白	S10
201	蓝	T20

今要向关系P中插入新行，新行的值分别列出如右图所示。哪些行能够插入？

1. (null, ‘黄’, T20)
2. (201, ‘红’, T20)
3. (037, ‘绿’, null)
4. (105, ‘蓝’, B01)
5. (101, ‘黄’, T11)