

## 2 关系模式介绍

- 关系模型

- 一种数据模型（数据结构、数据操作、数据完整性）
- 关系模型三要素（关系、关系操作、关系完整性）
- 域：相同类型值的集合，域的基数就是值的个数
- 笛卡尔积：元组，元组数目——笛卡尔基数，可有重复  $D_1 \times D_2 \times \cdots \times D_n$
- 关系是笛卡尔积的子集（有意义的元组构成）
  - 关系是一个命名的二维表
  - 关系的一列是属性，属性必定有名字（属性名）
  - 关系的一行是元组
  - 分量是行和列的交叉，或者说，分量是某个元组的某个属性（值）
  - 关系的度：属性的数目
  - 关系的基数：元组的数目
  - 属性的域：属性可能取值的集合

- 性质

- 列是同质的。列中的分量类型必须相同，因为它们来自同一个域。
- 不同的列可来自同个域，但属性名不能相同
- 关系里的任意两行不能相同，因为关系是一个元组集合。数学上严格禁止，但是在实际数据库里面可以容忍重复
- 关系的行或列是无序的。即改变行或列的次序（例如交换两行或者两列），关系不会变
- 属性值(分量)必须是原子的(不可分)。满足该条件的关系称作第一范式

- 关系实例：一组特定的行

- 每个属性的允许值集合称为属性域

- 属性值（通常）需要是原子的；也就是不可分割
- 特殊值 null 是每个域的成员。表示该值为“未知”
- 空值会导致许多操作的定义变得复杂

- 码

- 超码：一个或多个属性的集合，这些属性的组合可以在关系中唯一的标识一个元组
- 候选码：最小的超码（他们的任意真子集都不能够成为超码）
- 主码：被设计者选中在一个关系中区分不同元组的候选码
- 外码：另一个关系模式的主码

- 数据库模式

**department (dept-no, dept-name)**  
**student (dept-no, student-no, name, sex)**

一个数据库的模式

department

dept-no	dept-name
1	Computer Science
1	Economics

student

dept-no	student-no	name	sex
1	2	Smith	Male
1	1	Jones	Male
2	3	Kate	Female

一个数据库的实例

- 数据库模式——是数据库的逻辑结构。
- 数据库实例——是给定时刻数据库中数据的快照。
  - 架构：讲师 (ID、姓名、部门名称、薪水)
  - 实例：表

#### 架构图

#### 关系查询语言

##### 关系代数 (过程化)

- 选择  $\sigma$ ：返回输入关系中满足谓词的行
  - 谓词中允许比较符,  $=$ ,  $<$ ,  $>$ ,  $\neq$ ,  $\leq$ ,  $\geq$
- 投影：对输入关系的所有行输出指定属性。从输出中去除重复元组
- 自然连接：从两个输入关系中输出这样的元组对：它们在具有相同名字的所有属性上取值相同

$$r \bowtie_{\theta} s = \sigma_{\theta} (r \times s)$$

Thus

$$\sigma_{instructor.id = teaches.id} (instructor \times teaches)$$

Can equivalently be written as

$$instructor \bowtie_{instructor.id = teaches.id} teaches.$$

- 笛卡尔积：输出所有的元组对

R			S			$R \times S$					
A	B	C	B	C	D	A	R.B	R.C	S.B	S.C	D
1	2	3	4	5	6	1	2	3	4	5	6

- 并：输出两个输入关系中元组的并
- 差
- 重命名：重命名的是关系

$$\rho_{boy} (\sigma_{sex="male"} (student))$$

boy



student-no	name	sex
1	Jones	Male
2	Smith	Male

$$\rho_{x(A_1, A_2, \dots, A_n)}(E)$$

下标表示将E的结果关系命名为x, 同时属性依次命名为 $A_1, A_2, \dots, A_n$

- 赋值操作 箭头"<="

Example: 查找“物理”和“音乐”的所有讲师。

$Physics \leftarrow \sigma_{dept\_name = 'Physics'}(instructor)$

$Music \leftarrow \sigma_{dept\_name = 'Music'}(instructor)$

$Physics \cup Music$

- SQL和关系演算是非过程化
- 属于总结
  - 关系数据模型：建立在表的集合的基础上。数据库用户可以对这些表进行查询，插入、删除、更新等。
  - 关系的模式是指它的逻辑设计，关系的实例指它在特定时刻的内容。
  - 关系的超码是一个或多个属性的集合，这些属性上的取值保证可以唯一识别出关系中的元组。候选码是最小的超码。关系的一个候选码被选作主码
  - 参照关系中的外码：对于参照关系中的每个元组来说，它在外码属性上的取值等于被参照关系中某元组主码的值。
  - 模式图是数据库中模式的图形化表示，显示数据库的关系，关系的属性，主码和外码
  - 关系查询语言定义了一组运算集，可以用于表上并输出表作为结果。这些运算可以组合成表达式
  - 关系代数提供了一组运算，以一个或多个关系为输入，返回一个关系为输出。

以上内容整理于 [幕布文档](#)